



Guide du développeur

Amazon Location Service



Amazon Location Service: Guide du développeur

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

Les marques et la présentation commerciale d'Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n'est pas d'Amazon, d'une manière susceptible de créer une confusion parmi les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

Table of Contents

Bienvenue	1
Qu'est-ce qu'Amazon Location Service ?	1
Fonctions principales	2
Services connexes	3
Démarrage rapide	5
Création d'une application Web	5
Créer des ressources	6
Configuration de l'authentification	7
Création de code HTML	9
Ajouter la carte	12
Ajouter une recherche	16
Candidature finale	21
Quelle est la prochaine étape	26
Création d'une application Android	26
Création de ressources Amazon Location pour votre application	27
Configuration de l'authentification	29
Créez l'application	32
Ajoutez la carte	32
Ajouter une recherche	36
Ajouter un suivi	46
Quelle est la prochaine étape	55
Création d'une application iOS	55
Créer des ressources	56
Configuration de l'authentification	58
Création de l'application	60
Code initial	61
Ajouter une carte	64
Ajouter une recherche	68
Ajouter un suivi	70
Quelle est la prochaine étape	82
Concepts de localisation d'Amazon	83
Présentation	84
Mappages	85
Styles de carte	86

Opinions politiques	86
Couches personnalisées	87
Rendu cartographique	87
Terminologie des cartes	89
Recherche de lieux	90
Concepts de géocodage	92
Résultats de recherche	92
Résultats multiples et pertinence	93
Résultats de l'adresse	93
Stockage des résultats de géocodage	96
Terminologie des lieux	96
Acheminements	97
Ressources pour le calculateur d'itinéraires	98
Calcul d'un itinéraire	98
Planification des itinéraires	100
Terminologie des itinéraires	101
Geofences et traceurs	102
Géofences	103
Traceurs	105
Terminologie Geofence	109
Terminologie des traceurs	110
Cas d'utilisation courants	112
Applications d'engagement des utilisateurs et de géomarketing	113
Applications de suivi des actifs	115
Applications de livraison	116
Fournisseurs de données	118
Couverture et fonctionnalités des fournisseurs de données	118
Styles de carte	120
Plus d'informations	120
Esri	120
GrabMaps	129
Technologies HERE	135
Données ouvertes	143
Fonctionnalités par fournisseur de données	152
Conditions d'utilisation et attribution des données	157
Régions et points de terminaison	158

Régions	158
Points de terminaison	160
Endpoints d'opération de l'API	161
Quotas de service	162
Gestion des quotas de votre service de localisation Amazon	176
Développement avec Amazon Location	179
Scénarios et cas d'utilisation	180
Kits SDK et outils	181
SDK par langue	181
MapLibre	185
SDK Amazon Location	190
API de localisation Amazon	215
Utilisation d'Amazon Location avec un AWS SDK	215
Mise à jour des messages d'erreur	216
Exemples de code	250
Site de démonstration Amazon Location	251
Tutoriel : démarrage rapide	252
Tutoriel : enrichissement des bases de données	252
Exemple : application Explore	253
Exemple : styler une carte	254
Exemple : dessiner des marqueurs	254
Exemple : dessiner des points groupés	255
Exemple : Dessiner un polygone	255
Exemple : modification de la langue de la carte	256
Blog : Notifications relatives aux délais de livraison estimés	257
Exemple : mises à jour de la position du stream	258
Exemple : application mobile de géofencing et de suivi	258
Comment utiliser Amazon Location	260
Conditions préalables relatives au compte	261
Inscrivez-vous pour un Compte AWS	261
Création d'un utilisateur doté d'un accès administratif	262
Accorder l'accès à Amazon Location Service	263
Utilisation de cartes	265
Prérequis	266
Afficher des cartes	269
Dessiner sur une carte	325

Étendue des paramètres d'une carte	325
Gestion des ressources cartographiques	327
Recherche de lieux	330
Prérequis	331
Géocodage	335
Géocodage inversé	343
Remplissage automatique	347
Utilisation des identifiants de lieu	353
Catégories et filtrage	355
Tutoriel : enrichissement des bases de données	360
Gestion des ressources de l'index des lieux	375
Calcul des itinéraires	378
Prérequis	379
Calculer l'itinéraire	383
Planification d'itinéraires	387
Positions non situées sur une route	394
Heure de départ	396
Mode de voyage	397
Gestion des ressources routières	398
Géofencing et suivi	402
Étape 1 : Ajouter des géofences	403
Étape 2 : Commencer le suivi	411
Étape 3 : associer un traceur à une collection de géofences	425
Étape 4 : Évaluer la position des appareils par rapport aux géofences	427
Vérifier la position des appareils	430
Réagir aux événements avec EventBridge	432
Suivi à l'aide AWS IoT de MQTT	438
Gestion des ressources de géofence	446
Gestion des ressources de suivi	454
Exemple d'application mobile de géofencing et de suivi	458
Balisage de vos ressources	477
Restrictions	478
Accorder l'autorisation de taguer	479
Ajouter un tag à une ressource	480
Suivez les coûts par tag	480
Contrôlez l'accès aux ressources à l'aide de balises	482

En savoir plus	482
Autoriser l'accès à Amazon Location	482
Utilisation de clés d'API	483
Utilisation d'Amazon Cognito	490
Surveillance d'Amazon Location Service	501
Surveillance avec CloudWatch	502
Utilisation CloudTrail avec Amazon Location	507
Utilisation AWS CloudFormation pour créer des ressources	512
Amazon Location et AWS CloudFormation modèles	512
En savoir plus sur AWS CloudFormation	513
Sécurité	514
Protection des données	515
Confidentialité des données	516
Conservation des données	516
Chiffrement des données au repos	517
Chiffrement des données en transit	530
Gestion de l'identité et des accès	530
Public ciblé	531
Authentification par des identités	532
Gestion des accès à l'aide de politiques	536
Comment Amazon Location Service fonctionne avec IAM	538
Comment Amazon Location Service fonctionne avec les utilisateurs non authentifiés	548
Exemples de politiques basées sur l'identité	548
Résolution des problèmes	561
Réponse aux incidents	563
Journalisation et surveillance	563
Validation de conformité	564
Résilience	566
Sécurité de l'infrastructure	566
Analyse de la configuration et des vulnérabilités	567
Prévention de l'adjoint confus	567
Bonnes pratiques de sécurité	567
Meilleures pratiques en matière de Detective	568
Bonnes pratiques en matière de prévention	568
Bonnes pratiques	569
Sécurité	570

Gestion des ressources	570
Facturation et gestion des coûts	571
Quotas et utilisation	571
Historique de la documentation	573
Glossaire AWS	584
.....	dlxxxv

Bienvenue sur Amazon Location Service

Bienvenue dans le guide du développeur d'Amazon Location Service.

Les rubriques suivantes peuvent vous aider à démarrer dans la documentation, en fonction de ce que vous essayez de faire.

Obtenez une vue d'ensemble d'Amazon Location

- Découvrez les [concepts d'Amazon Location](#).
- Approfondissez les fonctionnalités dans ce [Comment utiliser Amazon Location Service](#) chapitre.
- Consultez les applications de démonstration [sur le site de démonstration d'Amazon Location](#).
- Si vous en possédez déjà un Compte AWS, vous pouvez utiliser la [console Amazon Location Service](#) pour découvrir ses fonctionnalités par vous-même.

Utiliser Amazon Location en tant que développeur

- Créez votre première application avec le [Démarrage rapide](#).
- Découvrez comment fonctionnent les différentes fonctionnalités d'Amazon Location Service dans ce [Comment utiliser Amazon Location Service](#) chapitre.
- Consultez les SDK et les outils mis à votre disposition dans le [Développement avec Amazon Location](#) chapitre.
- Consultez [des exemples de code et des didacticiels](#) que vous pouvez utiliser dans vos propres applications. Vous pouvez également consulter la [page d'exemples](#) du site de démonstration Amazon Location pour trouver des exemples, filtrables par fonctionnalité, langue ou plateforme.
- Pour en savoir plus sur les API Amazon Location, consultez le [guide de référence des API](#).

Qu'est-ce qu'Amazon Location Service ?

Amazon Location Service vous permet d'ajouter des données de localisation et des fonctionnalités aux applications, notamment des fonctionnalités telles que les cartes, les points d'intérêt, le géocodage, le routage, les géofences et le suivi. Amazon Location fournit des services de géolocalisation (LBS) utilisant des données de haute qualité provenant de fournisseurs internationaux de confiance Esri, Grab et HERE. Grâce à des données abordables, à des fonctionnalités de suivi et de géofencing, ainsi qu'à des mesures intégrées pour la surveillance de l'état de santé, vous pouvez créer des applications géolocalisées sophistiquées.

Avec Amazon Location, vous gardez le contrôle des données de votre entreprise. Amazon Location anonymise toutes les requêtes envoyées aux fournisseurs de données en supprimant les métadonnées des clients et les informations de compte. En outre, les informations sensibles de suivi et de géolocalisation, telles que l'emplacement des installations, des actifs et du personnel, ne quittent pas du tout votre AWS compte. Cela vous permet de protéger les informations sensibles contre les tiers, de protéger la vie privée des utilisateurs et de réduire les risques de sécurité de votre application. Avec Amazon Location, Amazon et les tiers n'ont pas le droit de vendre vos données ou de les utiliser à des fins publicitaires.

Amazon Location est entièrement intégré à des services tels qu'Amazon AWS CloudTrail CloudWatch EventBridge, Amazon et AWS Identity and Access Management (IAM). Amazon Location simplifie votre flux de développement grâce à l'intégration des données et accélère la mise en production des applications grâce à des fonctionnalités intégrées de surveillance, de sécurité et de conformité.

Pour connaître les points forts, les détails du produit et les prix, consultez la page de service d'[Amazon Location Service](#).

Principales fonctionnalités d'Amazon Location

Amazon Location propose les fonctionnalités suivantes :

Mappages

Amazon Location Service Maps vous permet de visualiser les informations de localisation et constitue la base de nombreuses fonctionnalités de service basées sur la localisation. Amazon Location Service fournit des vignettes cartographiques de différents styles provenant des fournisseurs mondiaux de données de localisation Esri, Grab et HERE, ainsi que des cartes Open Data.

Endroits

Amazon Location Service Places vous permet d'intégrer des fonctionnalités de recherche dans votre application, de convertir des adresses en coordonnées géographiques en latitude et longitude (géocodage) et de convertir une coordonnée en adresse postale (géocodage inversé). Amazon Location Service fournit des données géospatiales de haute qualité auprès d'Esri, Grab et HERE pour prendre en charge les fonctions de Places.

Routage

Amazon Location Service Routes vous permet de trouver des itinéraires et d'estimer le temps de trajet en fonction des informations up-to-date routières et du trafic en temps réel. Créez des fonctionnalités qui permettent à votre application de demander le temps de trajet, la distance et l'itinéraire entre deux emplacements. Calculez le temps et la distance pour une matrice d'itinéraires à utiliser dans la planification des itinéraires.

Géofencing

Amazon Location Service Geofences vous permet de donner à votre application la capacité de détecter et d'agir lorsqu'un appareil entre ou sort d'une limite géographique définie connue sous le nom de géofence. Envoyez automatiquement un événement d'entrée ou de sortie à Amazon EventBridge lorsqu'une violation de géofence est détectée. Cela vous permet de lancer des actions en aval, telles que l'envoi d'une notification à une cible.

Traceurs

Amazon Location Service Trackers vous permet de récupérer la position actuelle et historique des appareils qui exécutent votre application de suivi. Vous pouvez également associer les traceurs aux géofences d'Amazon Location Service afin d'évaluer automatiquement les mises à jour de localisation de vos appareils par rapport à vos géofences. Les trackers peuvent vous aider à réduire les coûts en filtrant les mises à jour de position qui n'ont pas été déplacées avant de les stocker ou de les évaluer par rapport aux géofences.

Lorsque vous utilisez des traceurs, les informations de localisation sensibles sur vos appareils suivis ne quittent pas votre AWS compte. Cela permet de protéger les informations sensibles des tiers, de protéger la vie privée des utilisateurs et de réduire les risques de sécurité.

Services que vous pouvez utiliser avec Amazon Location

Utilisez les services suivants en plus d'Amazon Location Service.

Surveillance et gestion intégrées

Amazon Location Service est intégré à Amazon CloudWatch et Amazon EventBridge pour une surveillance et une gestion des données efficaces : AWS CloudTrail

- Amazon CloudWatch — Consultez les statistiques relatives à l'utilisation et à l'état des services, notamment les demandes, la latence, les défaillances et les journaux. Pour plus d'informations, consultez [the section called “Surveillance avec CloudWatch”](#).
- AWS CloudTrail— Enregistrez et surveillez vos appels d'API, qui incluent les actions effectuées par un utilisateur, un rôle ou un service AWS. Pour plus d'informations, consultez [the section called “Utilisation CloudTrail avec Amazon Location”](#).
- Amazon EventBridge — Activez une architecture d'application axée sur les événements afin de pouvoir utiliser des AWS Lambda fonctions pour activer d'autres parties de votre application et de vos flux de travail. Pour plus d'informations, consultez [the section called “Réagir aux événements avec EventBridge”](#).

Outils pour développeurs

Amazon Location Service propose une variété d'outils permettant aux développeurs de créer des applications géolocalisées. Il s'agit notamment AWS des SDK standard, des SDK mobiles et Web, ainsi que des exemples de code pour les combiner avec des bibliothèques open source telles que MapLibre. Utilisez la [console Amazon Location Service](#) pour en savoir plus sur les ressources et pour commencer à utiliser un outil d'apprentissage visuel et interactif.

Démarrage rapide avec Amazon Location Service

Le moyen le plus efficace de démarrer avec Amazon Location Service est d'utiliser la [console Amazon Location](#). Vous pouvez créer et gérer vos ressources et essayer la fonctionnalité Amazon Location à l'aide de [la page Explore](#).

Note

Pour utiliser la console Amazon Location Service ou suivre le reste de ce didacticiel, vous devez d'abord les terminer [Conditions préalables à l'utilisation d'Amazon Location Service](#), notamment en créant un AWS compte et en autorisant l'accès à Amazon Location.

Pour commencer à en savoir plus sur les API Amazon Location, utilisez le didacticiel suivant pour créer une application simple qui affiche une carte interactive et utilise la fonctionnalité de recherche. Il existe trois versions du didacticiel : l'une vous montre comment créer une page Web simple en utilisant JavaScript, la seconde montre la même chose pour une application Android utilisant Kotlin, et la troisième montre la même chose pour une application iOS utilisant Swift.

Rubriques

- [Création d'une application Web](#)
- [Création d'une application Android](#)
- [Création d'une application iOS](#)

Création d'une application Web

Dans cette section, vous allez créer une page Web statique avec une carte et la possibilité de rechercher un emplacement. Vous allez d'abord créer vos ressources Amazon Location et créer une clé d'API pour votre application.

Rubriques

- [Création de ressources Amazon Location pour votre application](#)
- [Configuration de l'authentification pour votre application](#)
- [Création du code HTML pour votre application](#)
- [Ajouter une carte interactive à votre application](#)

- [Ajouter un moteur de recherche à votre application](#)
- [Voir la demande finale](#)
- [Quelle est la prochaine étape](#)

Création de ressources Amazon Location pour votre application

Si vous ne les avez pas déjà, vous devez créer les ressources Amazon Location que votre application utilisera. Vous créez ici une ressource cartographique pour afficher des cartes dans votre application, ainsi qu'un index de lieux pour rechercher des emplacements sur la carte.

Pour ajouter des ressources de localisation à votre application

1. Choisissez le style de carte que vous souhaitez utiliser.
 - a. Dans la console Amazon Location, sur la page [Cartes](#), choisissez Create map pour prévisualiser les styles de carte.
 - b. Ajoutez un nom et une description pour la nouvelle ressource cartographique. Notez le nom que vous utilisez pour la ressource cartographique. Vous en aurez besoin lors de la création de votre fichier de script ultérieurement dans le didacticiel.
 - c. Choisissez une carte.

Note

Le choix d'un style de carte permet également de choisir le fournisseur de données cartographiques que vous utiliserez. Si votre application suit ou achemine des actifs que vous utilisez dans votre entreprise, tels que des véhicules de livraison ou des employés, vous ne pouvez utiliser HERE que comme fournisseur de géolocalisation. Pour plus d'informations, consultez la section 82 des [conditions de AWS service](#).

- d. Acceptez les conditions générales d'Amazon Location, puis choisissez Create map. Vous pouvez interagir avec la carte que vous avez choisie : zoomer, dézoomer ou vous déplacer dans n'importe quelle direction.
 - e. Notez le nom de ressource Amazon (ARN) affiché pour votre nouvelle ressource cartographique. Vous l'utiliserez pour créer l'authentification correcte plus loin dans ce didacticiel.
2. Choisissez l'index des lieux que vous souhaitez utiliser.

- a. Dans la console Amazon Location, sur la page [des index de lieux](#), choisissez Create place index.
- b. Ajoutez un nom et une description pour la nouvelle ressource d'index des lieux. Notez le nom que vous utilisez pour la ressource d'index des lieux. Vous en aurez besoin lors de la création de votre fichier de script ultérieurement dans le didacticiel.
- c. Choisissez un fournisseur de données.

 Note

Dans la plupart des cas, choisissez le fournisseur de données qui correspond au fournisseur de cartes que vous avez déjà choisi. Cela permet de garantir que les recherches correspondent aux cartes.

Si votre application suit ou achemine des actifs que vous utilisez dans votre entreprise, tels que des véhicules de livraison ou des employés, vous ne pouvez utiliser HERE que comme fournisseur de géolocalisation. Pour plus d'informations, consultez la section 82 des [conditions de AWS service](#).

- d. Choisissez l'option Stockage des données. Pour ce didacticiel, les résultats ne sont pas enregistrés. Vous pouvez donc choisir Non, à usage unique uniquement.
- e. Acceptez les conditions générales d'Amazon Location, puis choisissez Create place index.
- f. Notez l'ARN affiché pour votre nouvelle ressource d'index de localisation. Vous l'utiliserez pour créer l'authentification correcte dans la section suivante de ce didacticiel.

Configuration de l'authentification pour votre application

L'application que vous créez dans ce didacticiel est utilisée de manière anonyme, ce qui signifie que vos utilisateurs ne sont pas tenus de se AWS connecter pour utiliser l'application. Toutefois, par défaut, les API Amazon Location Service nécessitent une authentification pour être utilisées. Vous pouvez utiliser Amazon Cognito ou des clés d'API pour authentifier et autoriser les utilisateurs anonymes. Dans ce didacticiel, vous allez créer des clés d'API à utiliser dans l'exemple d'application.

 Note

Pour plus d'informations sur l'utilisation des clés d'API ou d'Amazon Cognito avec Amazon Location Service, consultez. [Octroi de l'accès à Amazon Location Service](#)

Pour configurer l'authentification pour votre application

1. Accédez à la [console Amazon Location](#) et choisissez les clés d'API dans le menu de gauche.
2. Choisissez Créer une clé d'API.

Important

La clé d'API que vous créez doit se trouver dans la même Compte AWS AWS région que les ressources Amazon Location Service que vous avez créées dans la section précédente.

3. Sur la page Créer une clé d'API, renseignez les informations suivantes.
 - Nom : nom de votre clé d'API, tel que `MyWebAppKey`.
 - Ressources — Choisissez les ressources Amazon Location Map et Place index que vous avez créées dans la section précédente. Vous pouvez ajouter plusieurs ressources en choisissant Ajouter une ressource. Cela permettra à la clé d'API d'être utilisée avec ces ressources.
 - Actions — Spécifiez les actions que vous souhaitez autoriser à l'aide de cette clé d'API. Vous devez sélectionner au moins `geo : GetMap *` et `geo : SearchPlaceIndexForPosition` pour que le didacticiel fonctionne comme prévu.
 - Vous pouvez éventuellement ajouter une description, une date d'expiration ou des balises à votre clé d'API. Vous pouvez également ajouter un référent (tel que `*.example.com`), pour limiter l'utilisation de la clé à partir d'un domaine particulier. Cela signifie que le didacticiel ne fonctionnera qu'à partir de ce domaine.

Note

Il est recommandé de protéger l'utilisation de votre clé d'API en définissant un délai d'expiration ou un référent, sinon les deux.

4. Choisissez Create API key pour créer la clé API.
5. Choisissez Afficher la clé d'API, puis copiez la valeur de la clé pour une utilisation ultérieure dans le didacticiel. Ce sera sous forme de formulaire `1.public.a1b2c3d4...`

⚠ Important

Vous aurez besoin de cette clé pour écrire le code de votre application ultérieurement dans ce didacticiel.

Création du code HTML pour votre application

Dans ce didacticiel, vous allez créer une page HTML statique qui intègre une carte et permet à l'utilisateur de trouver ce qui se trouve à un emplacement sur la carte. L'application sera composée de trois fichiers : un fichier HTML et un fichier CSS pour la page Web, et un fichier JavaScript (.js) pour le code qui crée la carte et répond aux interactions de l'utilisateur et aux événements cartographiques.

Créons d'abord le framework HTML et CSS qui sera utilisé pour l'application. Il s'agira d'une page simple avec un `<div>` élément pour contenir le conteneur de cartes et un `<pre>`élément pour afficher les réponses JSON à vos requêtes.

Pour créer le code HTML de votre application de démarrage rapide

1. Créez un nouveau fichier appelé `quickstart.html`.
2. Modifiez le fichier dans l'éditeur de texte ou dans l'environnement de votre choix. Ajoutez le code HTML suivant au fichier.

```
<!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>
```

Ce code HTML contient un pointeur vers le fichier CSS que vous allez créer à l'étape suivante, des éléments d'espace réservé pour l'application et du texte explicatif.

Il existe deux éléments d'espace réservé que vous utiliserez ultérieurement dans ce didacticiel. Le premier est l'<div id="map">élément qui détiendra le contrôle de la carte. Le second est l'<pre id="response">élément, qui affichera les résultats de la recherche sur la carte.

3. Enregistrez votre fichier.

Ajoutez maintenant le CSS pour la page Web. Cela définira le style du texte et des éléments d'espace réservé pour l'application.

Pour créer le CSS de votre application de démarrage rapide

1. Créez un nouveau fichier appelé `main.css`, dans le même dossier que le fichier `quickstart.html` créé lors de la procédure précédente.
2. Modifiez le fichier dans l'éditeur que vous souhaitez utiliser. Ajoutez le texte suivant au fichier.

```
* {
  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;
}

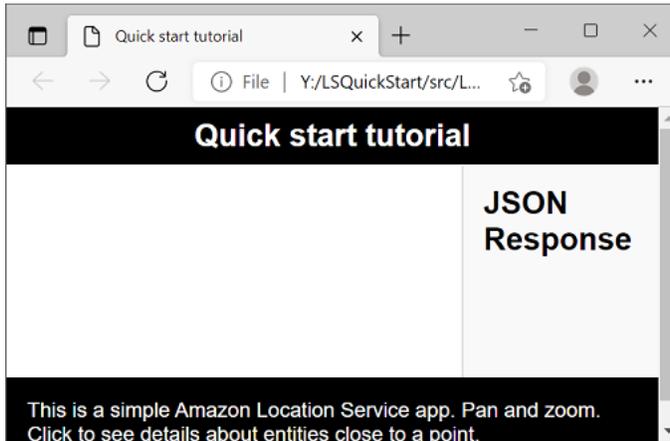
h2 {
  margin: 0;
}

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

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

Cela permet à la carte de remplir l'espace non utilisé par quoi que ce soit d'autre, définit la zone de nos réponses de telle sorte qu'elle occupe 30 % de la largeur de l'application et définit la couleur et les styles du titre et du texte explicatif.

3. Enregistrez le fichier.
4. Vous pouvez désormais consulter le `quickstart.html` fichier dans un navigateur pour voir la mise en page de l'application.



Vous allez ensuite ajouter le contrôle cartographique à l'application.

Ajouter une carte interactive à votre application

Maintenant que vous disposez d'un framework et d'un espace réservé à `div`, vous pouvez ajouter le contrôle cartographique à votre application. Ce didacticiel utilise [MapLibre GL JS](#) comme contrôle cartographique, pour obtenir des données depuis Amazon Location Service. Vous utiliserez également le [JavaScript Aide à l'authentification](#) pour faciliter la signature des appels vers les API Amazon Location à l'aide de votre clé d'API.

Pour ajouter une carte interactive à votre application

1. Ouvrez le `quickstart.html` fichier que vous avez créé dans la section précédente.
2. Ajoutez des références aux bibliothèques nécessaires et au fichier de script que vous allez créer. Les modifications que vous devez apporter sont indiquées dans **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>
```

Cela ajoute les dépendances suivantes à votre application :

- MapLibre ALLEZ-NOUS. Cette bibliothèque et cette feuille de style incluent un contrôle cartographique qui affiche des tuiles cartographiques et inclut des fonctionnalités interactives, telles que le panoramique et le zoom. Le contrôle permet également des extensions, telles que le dessin de vos propres entités sur la carte.
- Client Amazon Location. Cela fournit des interfaces pour la fonctionnalité Amazon Location nécessaire pour obtenir des données cartographiques et rechercher des lieux sur la carte. Le client Amazon Location est basé sur le AWS SDK pour la JavaScript version 3.
- Aide à l'authentification de localisation Amazon. Cela fournit des fonctions utiles pour authentifier Amazon Location Service à l'aide de clés d'API ou d'Amazon Cognito.

Cette étape ajoute également une référence à `main.js`, que vous allez créer ensuite.

3. Enregistrez le fichier `quickstart.html`.

4. Créez un nouveau fichier appelé `main.js` dans le même dossier que vos fichiers HTML et CSS, puis ouvrez-le pour le modifier.
5. Ajoutez le script suivant à votre fichier. Le texte en *rouge* doit être remplacé par la valeur de la clé API, le nom de la ressource cartographique et le nom de la ressource géographique que vous avez créés précédemment, ainsi que par l'identifiant de région de votre région (tel que `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();
```

Ce code définit les ressources Amazon Location, puis configure et initialise un contrôle MapLibre cartographique GL JS et le place dans votre `<div>` élément avec l'identifiant. `map`

Il est important de comprendre la `initializeMap()` fonction. Il crée un nouveau contrôle MapLibre cartographique (appelé `mglMap` localement, mais appelé `map` dans le reste du code) qui est utilisé pour afficher la carte dans votre application.

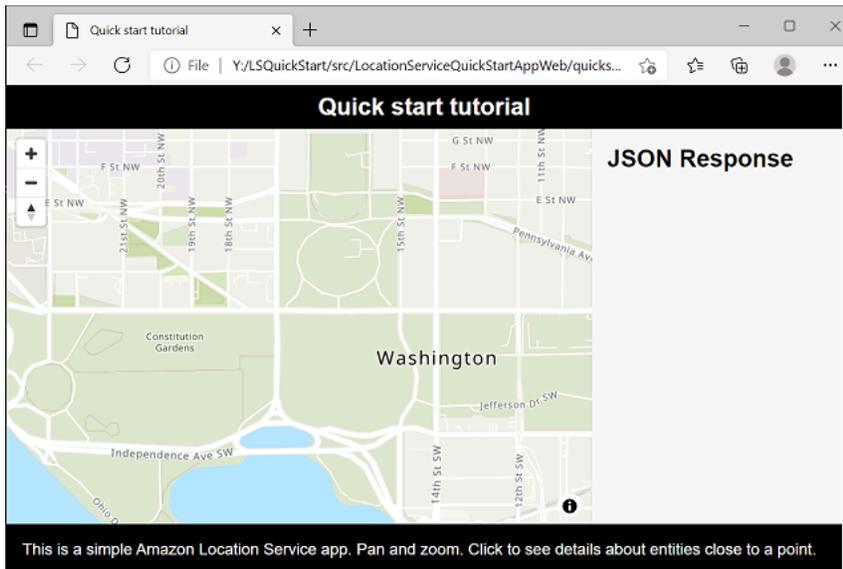
```
// Initialize the map
const mglMap = new maplibregl.Map({
  container: "map", // HTML element ID of map element
  center: [-77.03674, 38.891602], // Initial map centerpoint
  zoom: 16, // Initial map zoom
  style: 'https://maps.geo.${region}.amazonaws.com/maps/v0/maps/${mapName}/style-
descriptor?key=${apiKey}',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

Lorsque vous créez un nouveau contrôle MapLibre cartographique, les paramètres que vous transmettez indiquent l'état initial du contrôle cartographique. Ici, nous définissons les paramètres suivants.

- Conteneur HTML, qui utilise l'élément `map` `div` dans notre code HTML.
 - Le centre initial de la carte jusqu'à un point situé à Washington, DC.
 - Le niveau de zoom jusqu'à 16 (zoom sur le niveau d'un quartier ou d'un bloc).
 - Style à utiliser pour la carte, qui fournit MapLibre une URL à utiliser pour obtenir les tuiles de la carte et les autres informations nécessaires au rendu de la carte. Notez que cette URL inclut votre clé d'API pour l'authentification.
6. Enregistrez votre JavaScript fichier et ouvrez-le à l'aide d'un navigateur. Vous avez maintenant une carte sur votre page, sur laquelle vous pouvez utiliser les actions de panoramique et de zoom.

Note

Vous pouvez utiliser cette application pour voir comment se comporte le contrôle MapLibre cartographique. Vous pouvez essayer d'utiliser les touches `Ctrl` ou `Shift` lorsque vous faites glisser le pointeur pour découvrir d'autres manières d'interagir avec la carte. Toutes ces fonctionnalités sont personnalisables.



Votre application est presque terminée. Dans la section suivante, vous allez choisir un emplacement sur la carte et afficher l'adresse du lieu choisi. Vous allez également afficher le JSON obtenu sur la page, pour voir les résultats complets.

Ajouter un moteur de recherche à votre application

La dernière étape de votre application consiste à ajouter la recherche sur la carte. Dans ce cas, vous allez ajouter une recherche par géocodage inversé, dans laquelle vous trouverez les éléments à un emplacement donné.

Note

Amazon Location Service permet également de rechercher par nom ou adresse pour trouver les emplacements des lieux sur la carte.

Pour ajouter une fonctionnalité de recherche à votre application

1. Ouvrez le `main.js` fichier que vous avez créé dans la section précédente.
2. Modifiez la `main` fonction, comme indiqué. Les modifications que vous devez apporter sont indiquées dans **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.");
    }
});
}
```

Ce code commence par configurer l'assistant d'authentification Amazon Location pour qu'il utilise votre clé API.

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

Il utilise ensuite cet assistant d'authentification et la région que vous utilisez pour créer un nouveau client Amazon Location.

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

Ensuite, le code répond lorsque l'utilisateur choisit un endroit sur le contrôle cartographique. Pour ce faire, il capture un événement MapLibre fourni pour `click`.

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

L'événement `click` MapLibre fournit des paramètres qui incluent la latitude et la longitude choisies par l'utilisateur (`e.lngLat`). Dans l'événement `click`, le code crée le `searchPlaceIndexForPositionCommand` pour trouver les entités à la latitude et à la longitude données.

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

Ici, `IndexName` il s'agit du nom de la ressource Place Index que vous avez créée précédemment, de la latitude et de la longitude à rechercher, `Language` de la langue préférée pour les résultats et d'indiquer `MaxResults` à Amazon Location de ne renvoyer qu'un maximum de cinq résultats. `Position`

Le code restant vérifie l'absence d'erreur, puis affiche les résultats de la recherche dans l'élément appelé `response`, et affiche le premier résultat dans une boîte d'alerte.

3. (Facultatif) Si vous enregistrez et ouvrez le `quickstart.html` fichier dans un navigateur maintenant, le choix d'un emplacement sur la carte vous indiquera le nom ou l'adresse du lieu que vous avez choisi.
4. La dernière étape de l'application consiste à utiliser la MapLibre fonctionnalité pour ajouter un marqueur à l'endroit sélectionné par l'utilisateur. Modifiez la `main` fonction comme suit. Les modifications que vous devez apporter sont indiquées dans **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);

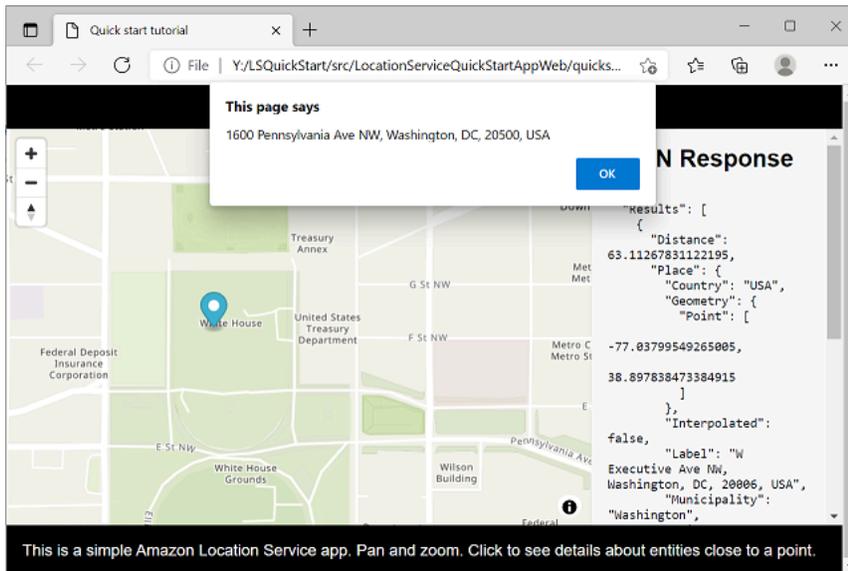
    // Set up parameters for search call
    let params = {
        IndexName: placesName,
        Position: [e.lngLat.lng, e.lngLat.lat],
        Language: "en",
        MaxResults: "5",
    };

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

    ...
```

Ce code déclare une `marker` variable, qui est renseignée chaque fois que l'utilisateur sélectionne un emplacement, indiquant où il l'a sélectionné. Le marqueur est automatiquement affiché par le contrôle cartographique, une fois qu'il est ajouté à la carte avec `.addTo(map);`. Le code vérifie également la présence d'un marqueur précédent et le supprime, de sorte qu'il n'y ait qu'un seul marqueur à la fois à l'écran.

5. Enregistrez le `main.js` fichier, puis ouvrez-le dans un navigateur. `quickstart.html` Vous pouvez vous déplacer et zoomer sur la carte, comme auparavant, mais désormais, si vous choisissez un emplacement, vous verrez des détails sur le lieu que vous avez choisi.



Votre demande de démarrage rapide est terminée. Ce didacticiel vous explique comment créer une application HTML statique qui :

- Crée une carte avec laquelle les utilisateurs peuvent interagir.
- Gère un événement cartographique (`click`).
- Appelle une API Amazon Location Service, spécifiquement pour effectuer une recherche sur la carte à un emplacement, en utilisant `searchPlaceIndexForPosition`.
- Utilise le contrôle MapLibre cartographique pour ajouter un marqueur.

Voir la demande finale

Le code source final de cette application est inclus dans cette section. Vous pouvez également trouver le projet final [sur GitHub](#).

[Vous pouvez également trouver une version de l'application qui utilise Amazon Cognito au lieu des clés d'API activées. GitHub](#)

Overview

Sélectionnez chaque onglet pour afficher le code source final des fichiers dans ce didacticiel de démarrage rapide.

Les fichiers sont les suivants :

- quickstart.html : le framework de votre application, y compris les supports d'éléments HTML pour la carte et les résultats de recherche.
- main.css : feuille de style de l'application.
- main.js : script de votre application qui authentifie l'utilisateur, crée la carte et effectue des recherches sur un `click` événement.

quickstart.html

Le framework HTML pour l'application de démarrage rapide.

```
<!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 feuille de style de l'application de démarrage rapide.

```
* {
  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

Le code de l'application de démarrage rapide. Le texte en *rouge* doit être remplacé par les noms d'objets Amazon Location appropriés.

```
// 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();
```

Quelle est la prochaine étape

Vous avez terminé le didacticiel de démarrage rapide et vous devriez avoir une idée de la façon dont Amazon Location Service est utilisé pour créer des applications. Pour tirer le meilleur parti d'Amazon Location, vous pouvez consulter les ressources suivantes :

- Approfondissez les [concepts d'Amazon Location Service](#)
- En savoir plus sur [l'utilisation des fonctionnalités d'Amazon Location](#)
- Découvrez comment développer cet exemple et créer des applications plus complexes en consultant des [exemples de code à l'aide d'Amazon Location](#)

Création d'une application Android

Dans cette section, vous allez créer une application Android avec une carte, la possibilité de rechercher un emplacement et un suivi au premier plan. Tout d'abord, vous allez créer vos ressources Amazon Location, une identité Amazon Cognito et une clé d'API pour votre application.

Rubriques

- [Création de ressources Amazon Location pour votre application](#)
- [Configuration de l'authentification pour votre application](#)
- [Création de l'application Android de base](#)

- [Ajouter une carte interactive à votre application](#)
- [Ajout de la recherche par géocodage inversé à votre application](#)
- [Ajouter le suivi à votre application](#)
- [Quelle est la prochaine étape](#)

Création de ressources Amazon Location pour votre application

Si vous ne les avez pas déjà, vous devez créer les ressources Amazon Location que votre application utilisera. Vous créez ici une ressource cartographique pour afficher des cartes dans votre application, un index des lieux pour rechercher des emplacements sur la carte et un traceur pour suivre un objet sur la carte.

Pour ajouter des ressources de localisation à votre application

1. Choisissez le style de carte que vous souhaitez utiliser.
 - a. Dans la console Amazon Location, sur la page [Cartes](#), choisissez Create map pour prévisualiser les styles de carte.
 - b. Ajoutez un nom et une description pour la nouvelle ressource cartographique. Notez le nom que vous utilisez pour la ressource cartographique. Vous en aurez besoin lors de la création de votre fichier de script ultérieurement dans le didacticiel.
 - c. Nous vous recommandons de choisir le style de carte HERE pour votre carte.

Note

Le choix d'un style de carte permet également de choisir le fournisseur de données cartographiques que vous utiliserez. Si votre application suit ou achemine des actifs que vous utilisez dans votre entreprise, tels que des véhicules de livraison ou des employés, vous ne pouvez utiliser HERE que comme fournisseur de géolocalisation. Pour plus d'informations, consultez la section 82 des [conditions de AWS service](#).

- d. Acceptez les conditions générales d'Amazon Location, puis choisissez Create map. Vous pouvez interagir avec la carte que vous avez choisie : zoomer, dézoomer ou vous déplacer dans n'importe quelle direction.
- e. Notez le nom de ressource Amazon (ARN) affiché pour votre nouvelle ressource cartographique. Vous l'utiliserez pour créer l'authentification correcte plus loin dans ce didacticiel.

2. Choisissez l'index des lieux que vous souhaitez utiliser.
 - a. Dans la console Amazon Location, sur la page [des index de lieux](#), choisissez Create place index.
 - b. Ajoutez un nom et une description pour la nouvelle ressource d'index des lieux. Notez le nom que vous utilisez pour la ressource d'index des lieux. Vous en aurez besoin lors de la création de votre fichier de script ultérieurement dans le didacticiel.
 - c. Choisissez un fournisseur de données.

 Note

Dans la plupart des cas, choisissez le fournisseur de données qui correspond au fournisseur de cartes que vous avez déjà choisi. Cela permet de garantir que les recherches correspondent aux cartes.

Si votre application suit ou achemine des actifs que vous utilisez dans votre entreprise, tels que des véhicules de livraison ou des employés, vous ne pouvez utiliser HERE que comme fournisseur de géolocalisation. Pour plus d'informations, consultez la section 82 des [conditions de AWS service](#).

- d. Choisissez l'option Stockage des données. Pour ce didacticiel, les résultats ne sont pas enregistrés. Vous pouvez donc choisir Non, à usage unique uniquement.
 - e. Acceptez les conditions générales d'Amazon Location, puis choisissez Create place index.
 - f. Notez l'ARN affiché pour votre nouvelle ressource d'index de localisation. Vous l'utiliserez pour créer l'authentification correcte dans la section suivante de ce didacticiel.
3. Pour créer un outil de suivi à l'aide de la console Amazon Location.
 - a. Ouvrez la [console Amazon Location Service](#).
 - b. Dans le volet de navigation de gauche, choisissez Trackers.
 - c. Choisissez Créer un tracker.
 - d. Remplissez tous les champs obligatoires.
 - e. Sous Filtrage des positions, nous vous recommandons d'utiliser le paramètre par défaut : TimeBased.
 - f. Choisissez Créer un tracker pour terminer.

Configuration de l'authentification pour votre application

L'application que vous créez dans ce didacticiel est anonyme, ce qui signifie que vos utilisateurs n'ont pas besoin de se connecter AWS pour utiliser l'application. Cependant, les API Amazon Location Service nécessitent une authentification pour être utilisées. Vous pouvez utiliser des clés d'API ou Amazon Cognito pour authentifier et autoriser les utilisateurs anonymes. Ce didacticiel utilisera Amazon Cognito et les clés d'API pour authentifier votre application.

Note

Pour plus d'informations sur l'utilisation d'Amazon Cognito ou des clés d'API avec Amazon Location Service, consultez. [Octroi de l'accès à Amazon Location Service](#)

Les didacticiels suivants vous montrent comment configurer l'authentification pour la carte, l'index des lieux et le tracker que vous avez créés, ainsi que comment configurer les autorisations pour Amazon Location.

Configurer l'authentification

1. Accédez à la [console Amazon Location](#) et sélectionnez les clés d'API dans le menu de gauche.
2. Cliquez sur « Créer une clé API ». N'oubliez pas que la clé d'API doit se trouver dans le même AWS compte et dans la même région que les ressources Amazon Location Service créées précédemment.
3. Renseignez les informations requises sur la page « Créer une clé API » :
 - Nom : Donnez un nom à votre clé d'API, par exemple MyAppKey.
 - Ressources : Choisissez les ressources d'index Amazon Location Service Map and Place créées précédemment. Vous pouvez ajouter plusieurs ressources en sélectionnant « Ajouter une ressource ». Cela permet d'utiliser la clé d'API avec des ressources spécifiées.
 - Actions : Spécifiez les actions autorisées pour cette clé d'API. Au minimum, sélectionnez `geo:GetMap` et assurez-vous `geo:SearchPlaceIndexForPosition` que le didacticiel fonctionne comme prévu.
 - Facultatif, vous pouvez ajouter une description, une date d'expiration, des balises ou un référent, par exemple `https://www.example.com` pour limiter l'utilisation de la clé à un domaine spécifique, permettant ainsi au didacticiel de fonctionner uniquement dans ce domaine.

4. Cliquez sur Créer une clé d'API pour générer la clé d'API.
5. Sélectionnez Afficher la clé d'API et copiez la valeur de la clé, par exemple `v1.public.a1b2c3d4` pour une utilisation ultérieure dans le didacticiel.

Création d'une politique IAM pour le suivi

1. Connectez-vous à la console IAM à l'adresse <https://console.aws.amazon.com/iam/> en tant qu'utilisateur disposant d'autorisations Administrateur.
2. Dans le panneau de navigation, sélectionnez Politiques (Politiques).
3. Dans le panneau de contenu, sélectionnez Créer une politique.
4. Choisissez l'option JSON, puis copiez et collez cette politique JSON dans la zone de texte 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}"
      ]
    }
  ]
}
```

Il s'agit d'un exemple de politique de suivi. Pour utiliser cet exemple pour votre propre politique, remplacez les TrackerNameespaces réservés RegionAccount, et.

Note

Bien que les pools d'identités non authentifiés soient destinés à être exposés sur des sites Internet non sécurisés, notez qu'ils seront échangés contre des informations d'identification AWS standard d'une durée limitée.

Il est important de définir de manière appropriée les rôles IAM associés aux pools d'identités non authentifiés. Pour plus d'informations sur l'utilisation et la définition appropriée des politiques dans Amazon Cognito avec Amazon Location Service, [consultez la section Accorder l'accès à Amazon Location Service](#).

5. Sur la page Révision et création, saisissez un nom pour le champ de nom de la politique. Passez en revue les autorisations accordées par votre politique, puis choisissez Créer une politique pour enregistrer votre travail.

La nouvelle politique s'affiche dans la liste des politiques gérées et est prête à être attachée.

Configurez l'authentification pour votre suivi

1. Configurez l'authentification pour votre application cartographique dans la console [Amazon Cognito](#).
2. Ouvrez la page Groupes d'identités.

Note

Le pool que vous créez doit se trouver dans le même AWS compte et dans la même AWS région que les ressources Amazon Location Service que vous avez créées dans la section précédente.

3. Choisissez Create Identity Pool.
4. En commençant par l'étape Configurer la confiance du pool d'identités. Pour l'authentification de l'accès utilisateur, sélectionnez Accès invité, puis appuyez sur Suivant.
5. Sur la page Configurer les autorisations, sélectionnez Utiliser un rôle IAM existant et entrez le nom du rôle IAM que vous avez créé à l'étape précédente. Lorsque vous êtes prêt, appuyez sur Suivant pour passer à l'étape suivante.
6. Sur la page Configurer les propriétés, attribuez un nom à votre pool d'identités. Appuyez ensuite sur Suivant.

7. Sur la page Réviser et créer, passez en revue toutes les informations présentes, puis appuyez sur Créer un pool d'identités.
8. Ouvrez la page Groupes d'identités et sélectionnez le pool d'identités que vous venez de créer. Copiez ou notez ensuite IdentityPoolId ce que vous utiliserez ultérieurement dans le script de votre navigateur.

Création de l'application Android de base

Dans ce didacticiel, vous allez créer une application Android qui intègre une carte et permet à l'utilisateur de trouver ce qui se trouve à un emplacement sur la carte.

Créez d'abord une application Kotlin vide à l'aide du nouvel assistant de projet d'Android Studio.

Pour créer une application vide (AndroidStudio)

1. Démarrer AndroidStudio. Ouvrez le menu et choisissez Fichier, Nouveau, Nouveau projet.
2. Dans l'onglet Téléphone et tablette, sélectionnez Activité vide, puis cliquez sur Suivant.
3. Choisissez un nom, un nom de package et un emplacement de sauvegarde pour votre application.
4. Dans la liste déroulante de la langue, sélectionnez Kotlin.
5. Choisissez Terminer pour créer votre application vierge.

Ajouter une carte interactive à votre application

Maintenant que vous avez créé une application de base, vous pouvez ajouter un contrôle cartographique à votre application. Ce didacticiel utilise des clés d'API pour gérer la vue cartographique. Le contrôle cartographique lui-même fait partie de la [bibliothèque MapLibre native](#), avec la clé API et MapLibre les données cartographiques proviennent d'Amazon Location.

Pour ajouter une carte à votre application, vous devez effectuer les actions suivantes :

- Ajoutez la MapLibre dépendance à votre projet.
- Configurez le code d'affichage de la carte avec Compose.
- Écrivez du code pour afficher la carte.

Pour ajouter la carte à votre application, procédez comme suit :

1. Ajoutez la MapLibre dépendance à votre projet

- a. Dans AndroidStudio, sélectionnez le menu Affichage, puis choisissez Outils, Fenêtres, Projet. Cela ouvrira la fenêtre Projet, qui vous permettra d'accéder à tous les fichiers de votre projet.
- b. Dans la fenêtre Projet, ouvrez Gradle puis ouvrez le `libs.versions.toml` fichier dans l'arborescence. Le `libs.versions.toml` fichier sera alors ouvert pour modification. Ajoutez maintenant les données de version et de bibliothèque ci-dessous dans le `libs.versions.toml` fichier.

```
[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. Une fois que vous avez terminé de modifier le `libs.versions.toml` fichier, vous AndroidStudio devez resynchroniser le projet. En haut de la fenêtre d'`libs.versions.toml` édition, vous AndroidStudio invite à effectuer la synchronisation. Sélectionnez « Synchroniser maintenant » pour synchroniser votre projet avant de continuer.
- d. Dans la fenêtre Projet, ouvrez Gradle Scripts dans l'arborescence et sélectionnez le `build.gradle` fichier pour votre module d'application. Le `build.gradle` fichier sera alors ouvert pour modification.
- e. Au bas du fichier, dans la section dépendances, ajoutez la dépendance suivante.

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

- f. Une fois que vous avez fini d'ajouter les dépendances Gradle, Android Studio doit resynchroniser le projet. En haut de la fenêtre d'édition de build.gradle, Android Studio, sélectionnez Synchroniser maintenant pour synchroniser votre projet avant de continuer.
2. Vous allez maintenant configurer le code de vue cartographique avec Compose. Procédez comme suit :
 - a. Dans la fenêtre Projet, ouvrez App, Java, le *nom de votre package* dans l'arborescence, et accédez au dossier ui. Dans le dossier ui, créez un répertoire de vue.
 - b. Dans le répertoire View, créez un MapLoadScreen.kt fichier.
 - c. Ajoutez le code suivant à votre MapLoadScreen.kt fichier.

```
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. Écrivez du code pour afficher la carte.

a. Ajoutez le code suivant à votre MainActivity.kt fichier.

```
// ...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. Enregistrez le fichier `MainActivity.kt`. Vous pouvez maintenant créer l'application. Pour l'exécuter, vous devrez peut-être configurer un appareil pour l'émuler ou utiliser l'application sur votre appareil. Android Studio Utilisez cette application pour voir comment se comporte le contrôle cartographique. Vous pouvez le déplacer en le faisant glisser sur la carte et en le pinçant pour zoomer.

Dans la section suivante, vous allez ajouter un marqueur sur la carte et afficher l'adresse de l'emplacement où se trouve le marqueur lorsque vous déplacez la carte.

Ajout de la recherche par géocodage inversé à votre application

Vous allez maintenant ajouter la recherche par géocodage inversé à votre application, dans laquelle vous trouverez les éléments à un emplacement donné. Pour simplifier l'utilisation d'une application Android, nous allons effectuer une recherche au centre de l'écran. Pour trouver un nouvel emplacement, déplacez la carte jusqu'à l'endroit où vous souhaitez effectuer la recherche. Nous placerons un marqueur au centre de la carte pour indiquer où nous cherchons.

L'ajout d'une recherche par géocodage inversé comportera deux parties.

- Ajoutez un marqueur au centre de l'écran pour indiquer à l'utilisateur où nous effectuons la recherche.
- Ajoutez une zone de texte pour les résultats, puis recherchez ce qui se trouve à l'emplacement du marqueur et affichez-le dans la zone de texte.

Pour ajouter un marqueur à votre application

1. Enregistrez cette image dans votre projet dans le `app/res/drawable` dossier sous le nom `red_marker.png` (vous pouvez également accéder à l'image depuis [GitHub](#)). Vous pouvez

également créer votre image. Vous pouvez également utiliser un fichier .png transparent pour les parties que vous ne souhaitez pas afficher.

2. Ajoutez le code suivant à votre fichier MapLoadScreen.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. Créez et exécutez votre application pour prévisualiser les fonctionnalités.

Votre application possède désormais un marqueur à l'écran. Dans ce cas, il s'agit d'une image statique qui ne bouge pas. Il est utilisé pour afficher le centre de la vue cartographique, où nous allons effectuer la recherche. Dans la procédure suivante, nous ajouterons la recherche à cet endroit.

Pour ajouter une recherche par géocodage inversé à un emplacement dans votre application

1. Dans la fenêtre Projet, ouvrez Gradle to `libs.versions.toml` file dans l'arborescence. Le `libs.versions.toml` fichier sera alors ouvert pour modification. Ajoutez maintenant les données de version et de bibliothèque ci-dessous dans le `libs.versions.toml` fichier.

```
[versions]
...
okhttp = "4.12.0"

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

[plugins]
...
```

2. Une fois que vous avez terminé de modifier le `libs.versions.toml` fichier, vous AndroidStudio devez resynchroniser le projet. En haut de la fenêtre d'édition de `libs.versions.toml`, vous AndroidStudio invite à effectuer la synchronisation. Sélectionnez « Synchroniser maintenant » pour synchroniser votre projet avant de continuer.
3. Dans la fenêtre Projet, ouvrez Gradle Scripts dans l'arborescence et sélectionnez le `build.gradle` fichier pour votre module d'application. Le `build.gradle` fichier sera alors ouvert pour modification.
4. Au bas du fichier, dans la section dépendances, ajoutez la dépendance suivante.

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

5. Une fois que vous avez terminé de modifier les dépendances Gradle, vous AndroidStudio devez resynchroniser le projet. En haut de la fenêtre d'`build.gradle` édition, vous AndroidStudio invite à effectuer la synchronisation. Sélectionnez cette option `SyncNow` pour synchroniser votre projet avant de continuer.
6. Maintenant, dans l'arborescence, ajoutez les données au répertoire des requêtes et créez la classe de `ReverseGeocodeRequest.kt` données. Ajoutez le code suivant à la classe.

```
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. Maintenant, dans l'arborescence, ajoutez les données au répertoire des réponses et créez la classe de `ReverseGeocodeResponse.kt` données. Ajoutez le code suivant à l'intérieur.

```
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. Maintenant, depuis la fenêtre Projet, ouvrez App, Java, le *nom de votre package* dans l'arborescence, et accédez au dossier ui, dans le dossier ui, créez le répertoire ViewModel.
9. Dans le répertoire ViewModel, créez un `MainViewModel.kt` fichier.
10. Ajoutez le code suivant à votre `MainViewModel.kt` fichier.

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

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

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

        val requestBody = ReverseGeocodeRequest(
            language = "en",
            maxResults = 1,
            position = listOf(latLng.longitude, latLng.latitude)
        )
    }
}
```

```

    val json = Gson().toJson(requestBody)

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

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

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

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

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

                if (responseLabel != null) {
                    label = responseLabel
                    isLabelAdded = true
                }
            }
        }
    })
}

```

11. S'il n'est pas déjà ouvert, `MapLoadScreen.kt` ouvrez-le comme dans la procédure précédente. Ajoutez le code suivant. Cela créera une vue de composition de texte dans laquelle vous verrez les résultats de recherche par géocodage inversé sur place.

```

// ...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. Dans l'application, sous Java, dans le dossier du nom du package AndroidStudio, ouvrez le MainActivity.kt fichier. Modifiez le code comme indiqué.

```

// ...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)
    setContentView {
        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
        )
      }
    }
  }
}
```

Ce code fonctionne avec la vue cartographique. La position de la caméra virtuelle définit la vue cartographique dans MapLibre. Le déplacement de la carte peut être considéré comme le déplacement de cette caméra virtuelle.

- ViewModel possède une variable d'étiquette : cette variable définit les données dans la vue de composition de texte.
- onMapReady: Cette fonction est mise à jour pour enregistrer deux nouveaux événements.
- L'onCameraMove événement se produit chaque fois que l'utilisateur déplace la carte. En général, lorsque vous déplacez la carte, nous voulons masquer la recherche jusqu'à ce que l'utilisateur ait fini de déplacer la carte.
- L'onCameraIdle événement se produit lorsque l'utilisateur interrompt le déplacement de la carte. Cet événement fait appel à notre fonction de géocodage inversé pour effectuer une recherche au centre de la carte.
- reverseGeocode(latLng: LatLng, apiKey: String): Cette fonction, appelée lors de l'événement onCameraIdle, recherche un emplacement au centre de la carte et met à jour l'étiquette pour afficher les résultats. Il utilise la cible de la caméra, qui définit le centre de la carte (là où la caméra regarde).

13. Enregistrez vos fichiers, créez et exécutez votre application pour voir si elle fonctionne.

Votre application de démarrage rapide avec fonctionnalité de recherche est terminée.

Ajouter le suivi à votre application

Pour ajouter le suivi à votre exemple d'application, procédez comme suit :

1. Ajoutez les dépendances du SDK de suivi et d'authentification à votre projet.
 2. Incluez les entrées d'autorisation et de service dans votre fichier `AndroidManifest.xml`.
 3. Configurez le code du bouton de démarrage/arrêt du suivi avec Compose.
 4. Ajoutez du code pour créer un `LocationTracker` objet et démarrer et arrêter le suivi.
 5. Créez un itinéraire de test avec l'émulateur Android.
1. Ajoutez les dépendances du SDK de suivi et d'authentification à votre projet.
 - a. Dans la fenêtre Projet, ouvrez Gradle puis ouvrez le `libs.versions.toml` fichier dans l'arborescence. Le `libs.versions.toml` fichier sera alors ouvert pour modification. Ajoutez maintenant les données de version et de bibliothèque ci-dessous dans le `libs.versions.toml` fichier.

```
[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. Une fois que vous avez terminé de modifier le `libs.versions.toml` fichier, vous AndroidStudio devez resynchroniser le projet. En haut de la fenêtre d'`libs.versions.toml` édition, vous AndroidStudio invite à effectuer la synchronisation. Sélectionnez « Synchroniser maintenant » pour synchroniser votre projet avant de continuer.

- c. Dans la fenêtre Projet, ouvrez Gradle Scripts dans l'arborescence et sélectionnez le `build.gradle` fichier pour votre module d'application. Le `build.gradle` fichier sera alors ouvert pour modification.
- d. Au bas du fichier, dans la section dépendances, ajoutez la dépendance suivante.

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

- e. Une fois que vous avez terminé de modifier les dépendances Gradle, vous AndroidStudio devez resynchroniser le projet. En haut de la fenêtre d'édition de `build.gradle`, AndroidStudio vous êtes invité à effectuer la synchronisation. Sélectionnez cette option SyncNow pour synchroniser votre projet avant de continuer.
2. Incluez les entrées d'autorisation et de service dans votre fichier `AndroidManifest.xml`.
 - Pour inclure les entrées d'autorisation et de service correctes dans votre fichier `AndroidManifest.xml` file, mettez à jour le fichier avec le code suivant :

```
<?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. Configurez le code du bouton de démarrage/arrêt du suivi avec Compose.
 - a. Ajoutez deux images de Play et Pause in res sous drawable nommées ic_pause et ic_play. Vous pouvez également accéder à l'image depuis [GitHub](#).
 - b. S'il n'est pas déjà ouvert, MapLoadScreen.kt ouvrez-le comme dans la procédure précédente. Ajoutez le code suivant. Cela créera une vue du bouton de composition où nous pourrons cliquer dessus pour démarrer et arrêter le suivi.

```
// ...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. Ajoutez du code pour créer un `LocationTracker` objet et démarrer et arrêter le suivi.

a. Ajoutez le code suivant dans le `MainViewModel.kt` fichier.

```

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

```

b. Ajoutez le code suivant à votre `MainActivity.kt` fichier.

```

// ...other imports
import software.amazon.location.auth.AuthHelper

```

```

import software.amazon.location.auth.LocationCredentialsProvider
import software.amazon.location.tracking.LocationTracker
import software.amazon.location.tracking.aws.LocationTrackingCallback
import software.amazon.location.tracking.config.LocationTrackerConfig
import software.amazon.location.tracking.database.LocationEntry
import software.amazon.location.tracking.filters.DistanceLocationFilter
import software.amazon.location.tracking.filters.TimeLocationFilter
import software.amazon.location.tracking.util.TrackingSdkLogLevel

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

    private val mainViewModel: MainViewModel by viewModels()
    private val poolId = "YOUR_AWS_IDENTITY_POOL_ID"
    private val trackerName = "YOUR_AWS_TRACKER_NAME"
    private val region = "YOUR_AWS_REGION"
    private val mapName = "YOUR_AWS_MAP_NAME"
    private val apiKey = "YOUR_AWS_API_KEY"
    private val coroutineScope = MainScope()
    private lateinit var locationCredentialsProvider:
LocationCredentialsProvider
    private lateinit var authHelper: AuthHelper

    override fun onCreate(savedInstanceState: Bundle?) {
        MapLibre.getInstance(this)
        super.onCreate(savedInstanceState)
        setContent {
            TestMapAppTheme {
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    MapLoadScreen(this, mainViewModel, onStartStopTrackingClick
= {
                        if (mainViewModel.isLocationTrackingForegroundActive) {
                            mainViewModel.isLocationTrackingForegroundActive =
false
                            mainViewModel.locationTracker?.stop()
                        } else {
                            if (checkLocationPermission(this))
                                mainViewModel.isLocationTrackingForegroundActive =
true
                        }
                    }
                }
            }
        }
    }
}

```

```

mainViewModel.locationTracker?.start(locationTrackingCallback = object :
    LocationTrackingCallback {
        override fun
onLocationAvailabilityChanged(locationAvailable: Boolean) {
    }

    override fun onLocationReceived(location:
LocationEntry) {
    }

    override fun onUploadSkipped(entries:
LocationEntry) {
    }

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

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

private fun authenticateUser() {
    coroutineScope.launch {
        authHelper = AuthHelper(applicationContext)
        locationCredentialsProvider =
authHelper.authenticateWithCognitoIdentityPool(
            poolId,
        )
        locationCredentialsProvider.let {
            val config = LocationTrackerConfig(
                trackerName = trackerName,
                logLevel = TrackingSdkLogLevel.DEBUG,
                latency = 1000,
            )
        }
    }
}

```

```
        frequency = 5000,
        waitForAccurateLocation = false,
        minUpdateIntervalMillis = 5000,
    )
    mainViewModel.locationTracker = LocationTracker(
        applicationContext,
        it,
        config,
    )

    mainViewModel.locationTracker?.enableFilter(TimeLocationFilter())

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

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

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

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

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

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

Le code ci-dessus montre comment créer un `LocationTracker` objet avec `AuthHelper` et comment démarrer et arrêter le suivi avec `LocationTracker`.

- `authenticateUser()`: Cette méthode crée `AuthHelper` des `LocationTracker` objets.
- `onStartStopTrackingClick`: Ce rappel est déclenché lorsque l'utilisateur clique sur le bouton démarrer/arrêter le suivi, qui démarrer/arrêter le suivi avec le SDK de suivi.

5. Créez un itinéraire de test avec l'émulateur Android.

- a. Ouvrez l'émulateur en lançant l'AVD à l'aide d'Android Studio.

- b. Ouvrez Extended Controls en cliquant sur l'icône Plus (trois points) dans la barre d'outils de l'émulateur.
- c. Ouvrez Emplacement en sélectionnant Emplacement dans la barre latérale.
- d. Créez un itinéraire avec les données GPX ou en cliquant sur la carte et en choisissant les données source et de destination.
- e. Lancez la simulation en cliquant sur PLAY ROUTE pour commencer à simuler l'itinéraire GPS.
- f. Testez l'application en exécutant votre application et en observant comment elle gère l'itinéraire simulé.

Voici la démo complète de l'application Android Quick Start.

Quelle est la prochaine étape

Le code source de cette application est disponible sur [GitHub](#).

Pour tirer le meilleur parti d'Amazon Location, vous pouvez consulter les ressources suivantes :

- Approfondissez les [concepts d'Amazon Location Service](#)
- En savoir plus sur [l'utilisation des fonctionnalités d'Amazon Location](#)
- Découvrez comment développer cet exemple et créer des applications plus complexes en consultant des [exemples de code à l'aide d'Amazon Location](#)

Création d'une application iOS

Dans cette section, vous allez créer une application iOS avec la possibilité de rechercher un emplacement et de suivre au premier plan. Tout d'abord, vous allez créer vos ressources Amazon Location et une identité Amazon Cognito pour votre application.

Rubriques

- [Création de ressources Amazon Location pour votre application](#)
- [Configuration de l'authentification pour votre application](#)
- [Création de l'application iOS de base](#)
- [Configuration du code initial](#)

- [Ajouter une carte interactive à votre application](#)
- [Ajouter un moteur de recherche à votre application](#)
- [Ajouter le suivi à votre application](#)
- [Quelle est la prochaine étape](#)

Création de ressources Amazon Location pour votre application

Si vous ne les avez pas déjà, vous devez créer les ressources Amazon Location que votre application utilisera. Vous allez créer une ressource cartographique pour afficher des cartes dans votre application, un index des lieux pour rechercher des emplacements sur la carte et un traceur pour suivre un objet sur la carte.

Pour ajouter des ressources de localisation à votre application

1. Choisissez le style de carte que vous souhaitez utiliser.
 - a. Dans la console Amazon Location, sur la page [Cartes](#), choisissez Create map pour prévisualiser les styles de carte.
 - b. Ajoutez un nom et une description pour la nouvelle ressource cartographique. Notez le nom que vous utilisez pour la ressource cartographique. Vous en aurez besoin lors de la création de votre fichier de script ultérieurement dans le didacticiel.
 - c. Choisissez une carte.

Note

Le choix d'un style de carte permet également de choisir le fournisseur de données cartographiques que vous utiliserez. Si votre application suit ou achemine des actifs que vous utilisez dans votre entreprise, tels que des véhicules de livraison ou des employés, vous ne pouvez utiliser HERE que comme fournisseur de géolocalisation. Pour plus d'informations, consultez la section 82 des [conditions de AWS service](#).

- d. Acceptez les conditions générales d'Amazon Location, puis choisissez Create map. Vous pouvez interagir avec la carte que vous avez choisie : zoomer, dézoomer ou vous déplacer dans n'importe quelle direction.
- e. Notez le nom de ressource Amazon (ARN) affiché pour votre nouvelle ressource cartographique. Vous l'utiliserez pour créer l'authentification correcte plus loin dans ce didacticiel.

2. Choisissez l'index des lieux que vous souhaitez utiliser.
 - a. Dans la console Amazon Location, sur la page [des index de lieux](#), choisissez Create place index.
 - b. Ajoutez un nom et une description pour la nouvelle ressource d'index des lieux. Notez le nom que vous utilisez pour la ressource d'index des lieux. Vous en aurez besoin lors de la création de votre fichier de script ultérieurement dans le didacticiel.
 - c. Choisissez un fournisseur de données.

 Note

Dans la plupart des cas, choisissez le fournisseur de données qui correspond au fournisseur de cartes que vous avez déjà choisi. Cela permet de garantir que les recherches correspondent aux cartes.

Si votre application suit ou achemine des actifs que vous utilisez dans votre entreprise, tels que des véhicules de livraison ou des employés, vous ne pouvez utiliser HERE que comme fournisseur de géolocalisation. Pour plus d'informations, consultez la section 82 des [conditions de AWS service](#).

- d. Choisissez l'option Stockage des données. Pour ce didacticiel, les résultats ne sont pas enregistrés. Vous pouvez donc choisir Non, à usage unique uniquement.
 - e. Acceptez les conditions générales d'Amazon Location, puis choisissez Create place index.
 - f. Notez l'ARN affiché pour votre nouvelle ressource d'index de localisation. Vous l'utiliserez pour créer l'authentification correcte dans la section suivante de ce didacticiel.
3. Pour créer un outil de suivi à l'aide de la console Amazon Location.
 - a. Ouvrez la [console Amazon Location Service](#).
 - b. Dans le volet de navigation de gauche, choisissez Trackers.
 - c. Choisissez Créer un tracker.
 - d. Remplissez tous les champs obligatoires.
 - e. Sous Filtrage des positions, nous vous recommandons d'utiliser le paramètre par défaut : TimeBased.
 - f. Choisissez Créer un tracker pour terminer.

Configuration de l'authentification pour votre application

L'application que vous créez dans ce didacticiel est utilisée de manière anonyme, ce qui signifie que vos utilisateurs ne sont pas tenus de se AWS connecter pour utiliser l'application. Cependant, les API Amazon Location Service nécessitent une authentification pour être utilisées. Vous utiliserez Amazon Cognito pour authentifier et autoriser les utilisateurs anonymes. Ce didacticiel utilisera Amazon Cognito pour authentifier votre application.

Note

Pour plus d'informations sur l'utilisation d'Amazon Cognito avec Amazon Location Service, consultez. [Octroi de l'accès à Amazon Location Service](#)

Les didacticiels suivants vous montrent comment configurer l'authentification pour la carte, l'index des lieux et le tracker que vous avez créés, ainsi que comment configurer les autorisations pour Amazon Location.

Création d'une politique IAM pour le suivi

1. Connectez-vous à la console IAM à l'adresse <https://console.aws.amazon.com/iam/> en tant qu'utilisateur disposant d'autorisations Administrateur.
2. Dans le panneau de navigation, sélectionnez Politiques (Politiques).
3. Dans le panneau de contenu, sélectionnez Créer une politique.
4. Choisissez l'option JSON, puis copiez et collez cette politique JSON dans la zone de texte 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}"
      ]
    }
  ]
}
```

Il s'agit d'un exemple de politique pour le suivi. Pour utiliser cet exemple pour votre propre politique, remplacez les TrackerNameespaces réservés Region Account IndexName,, MapName et.

Note

Bien que les pools d'identités non authentifiés soient destinés à être exposés sur des sites Internet non sécurisés, notez qu'ils seront échangés contre des informations d'identification standard limitées dans le temps. AWS

Il est important de définir de manière appropriée les rôles IAM associés aux pools d'identités non authentifiés. Pour plus d'informations sur l'utilisation et la définition appropriée des politiques dans Amazon Cognito avec Amazon Location Service, [consultez la section Accorder l'accès à Amazon Location Service](#).

5. Sur la page Révision et création, saisissez un nom pour le champ de nom de la politique. Vérifiez les autorisations accordées par votre politique, puis choisissez Créer une politique pour enregistrer votre travail.

La nouvelle politique s'affiche dans la liste des politiques gérées et est prête à être attachée.

Configurez l'authentification pour votre suivi

1. Configurez l'authentification pour votre application cartographique dans la console [Amazon Cognito](#).
2. Ouvrez la page Groupes d'identités.

Note

Le pool que vous créez doit se trouver dans le même AWS compte et dans la même AWS région que les ressources Amazon Location Service que vous avez créées dans la section précédente.

3. Choisissez Create Identity Pool.
4. En commençant par l'étape Configurer la confiance du pool d'identités. Pour l'authentification de l'accès utilisateur, sélectionnez Accès invité, puis appuyez sur Suivant.
5. Sur la page Configurer les autorisations, sélectionnez Utiliser un rôle IAM existant et entrez le nom du rôle IAM que vous avez créé à l'étape précédente. Lorsque vous êtes prêt, appuyez sur Suivant pour passer à l'étape suivante.
6. Sur la page Configurer les propriétés, attribuez un nom à votre pool d'identités. Appuyez ensuite sur Suivant.
7. Sur la page Réviser et créer, passez en revue toutes les informations présentes, puis appuyez sur Créer un pool d'identités.
8. Ouvrez la page Groupes d'identités et sélectionnez le pool d'identités que vous venez de créer. Copiez ou notez ensuite IdentityPoolId ce que vous utiliserez ultérieurement dans le script de votre navigateur.

Création de l'application iOS de base

Dans ce didacticiel, vous allez créer une application iOS qui intègre une carte et permet à l'utilisateur de trouver ce qui se trouve à un emplacement sur la carte.

Créons d'abord une application Swift à l'aide de l'assistant de projet de Xcode.

Pour créer une application vide (Xcode)

1. Ouvrez Xcode, puis dans le menu, choisissez Fichier, Nouveau, Nouveau projet.
2. Dans l'onglet iOS, sélectionnez App, puis Next.
3. Entrez un nom de produit, un identifiant d'organisation et dans le champ de saisie de l'interfaceSwiftUI. Choisissez Next pour finaliser la sélection.
4. Sélectionnez un emplacement où vous allez enregistrer votre projet et appuyez sur le bouton Créer pour créer l'application vide.

Une fois que vous avez créé l'application de base, vous devez installer les packages requis pour l'exemple d'application.

Installation des dépendances requises

1. Dans Xcode, cliquez avec le bouton droit sur le projet et choisissez Ajouter des packages... . Cela ouvrira la fenêtre Packages, dans laquelle vous pourrez ajouter des packages à votre projet.
2. Dans la fenêtre Packages, ajoutez les packages suivants :
 - [Pour le package natif de Maplibre, utilisez cette URL : https://github.com/maplibre/maplibre-gl-native-distribution](https://github.com/maplibre/maplibre-gl-native-distribution) À partir de l'URL, ajoutez les packages suivants :maplibre-gl-native-distribution, etMapbox.
 - Pour le SDK iOS d'authentification Amazon Location, utilisez cette URL : <https://github.com/aws-geospatial/amazon-location-mobile-auth-sdk-ios>. À partir de l'URL, ajoutez les packages suivants :amazon-location-mobile-auth-sdk-ios, etAmazonLocationi0SAuthSDK.
 - Pour le SDK iOS de suivi de la localisation Amazon, utilisez cette URL : <https://github.com/aws-geospatial/amazon-location-mobile-tracking-sdk-ios>. À partir de l'URL, ajoutez les packages suivants :amazon-location-mobile-tracking-sdk-ios, etAmazonLocationi0STrackingSDK.

Configuration du code initial

Activez les autorisations de localisation dans votre application

1. Ouvrez votre projet Xcode.
2. Localisez le Info.plist fichier du projet.
3. Ajoutez les clés nécessaires pour les autorisations de localisation en fonction des exigences de votre application. Voici les clés :
 - `NSLocationWhenInUseUsageDescription`: description des raisons pour lesquelles votre application a besoin d'un accès à la localisation lorsqu'elle est utilisée.
 - `NSLocationAlwaysAndWhenInUseUsageDescription`: description des raisons pour lesquelles votre application a besoin d'un accès permanent à la localisation.

Vous devez maintenant configurer les valeurs des ressources dans votre application. Ajoutez un nouveau fichier nommé `Config.xcconfig` et renseignez les valeurs que vous avez créées précédemment dans la console Amazon.

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

1. Dans la section du navigateur de gauche, sélectionnez le projet.
2. Dans la section des cibles, sélectionnez votre application et cliquez sur l'onglet Info.
3. Ajoutez des propriétés d'information avec des valeurs comme celles ci-dessous :
4. Ajoutez le `Config.swift` fichier avec le contenu ci-dessous, qui lira les valeurs de configuration à partir du fichier d'informations du bundle.

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

5. Créez un nouveau dossier portant ce nom `ViewModel` et ajoutez-y un `TrackingViewModel.swift` fichier.

```
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
}
}
```

Ajouter une carte interactive à votre application

Vous allez maintenant ajouter le contrôle cartographique à votre application. Ce didacticiel utilise MapLibre l' AWS API pour gérer la vue cartographique dans l'application. Le contrôle cartographique lui-même fait partie de la bibliothèque [MapLibre GL Native iOS](#).

1. Ajoutez `MapView.swift` le fichier dans le dossier Views avec le code suivant :

```
import SwiftUI
import MapLibre

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

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

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

```
context.coordinator.mlnMapView = mapView
mapView.delegate = context.coordinator

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

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

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

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

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

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

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

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

```

        mlnMapView.addAnnotation(marker)
        centerMarker = marker

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

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

```

2. Ajoutez `AWSSignatureV4Delegate` le fichier sous le `ViewModel` dossier. Ce fichier est utilisé pour signer toutes les requêtes `MapView` http pour afficher la carte :

```

import MapLibre
import AmazonLocationiOSAuthSDK

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

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

    func offlineStorage(_ storage: MLNOfflineStorage, urlForResourceOf kind:
MLNResourceKind, with url: URL) -> URL {
        if url.host?.contains("amazonaws.com") != true {
            return url
        }
        let signedURL = awsSigner.signURL(url: url, expires: .hours(1))
    }
}

```

```
        return signedURL
    }
}
```

3. Ajoutez `UserLocationView.swift` le fichier dans le dossier `Views`. Cela ajoute un bouton qui centre la carte sur l'emplacement de l'utilisateur

```
import SwiftUI

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

4. Ajoutez le `TrackingView.swift` fichier avec le code suivant :

```
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)
        }
    }
}
}
```

Vous pouvez maintenant créer l'application. Pour l'exécuter, vous devrez peut-être configurer un appareil pour l'émuler dans Xcode ou utiliser l'application sur votre appareil. Utilisez cette application pour voir comment se comporte le contrôle cartographique. Vous pouvez vous déplacer en faisant glisser le pointeur sur la carte et en pinçant pour zoomer. Vous pouvez vous-même modifier le mode de fonctionnement du contrôle cartographique pour le personnaliser en fonction des besoins de votre application.

Ajouter un moteur de recherche à votre application

Vous allez maintenant ajouter la recherche par géocodage inversé à l'application, où vous trouverez les éléments à un emplacement donné. Pour simplifier l'utilisation d'une application iOS, nous allons effectuer une recherche au centre de l'écran. Pour trouver un nouvel emplacement, déplacez la carte jusqu'à l'endroit où vous souhaitez effectuer la recherche. Nous placerons un marqueur au centre de la carte pour indiquer où nous cherchons.

1. Ajoutez le code suivant dans le fichier « TrackingViewModel .swift » qui est lié à la recherche par géocodage inversé

```
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. Mettez à jour le `TrackingView.swift` fichier avec le code suivant qui affichera l'adresse de l'emplacement centré de la vue cartographique

```

import SwiftUI

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

```

```
}  
    }  
  }  
}
```

Ajouter le suivi à votre application

La dernière étape de votre application consiste à ajouter une fonctionnalité de suivi à votre application. Dans ce cas, vous allez ajouter le suivi de départ, arrêter le suivi, récupérer et afficher les points de suivi sur votre application.

1. Ajoutez le `TrackingBottomView.swift` fichier dans votre projet. Il comporte un bouton qui démarre et arrête le suivi de la position des utilisateurs et affiche les points de suivi sur la carte.

```
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. Mettre à jour le TrackingView.swift fichier avec le code suivant

```
import SwiftUI

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



```
client.stopTracking()
trackingButtonText = NSLocalizedString("StartTrackingLabel", comment: "")
trackingButtonColor = .blue
trackingButtonIcon = "play.circle"
trackingActive = false
}
```

Note

Ils `startTracking` demanderont l'autorisation de localisation de l'utilisateur. L'application doit utiliser les autorisations `When In Use` ou `Only Once`. Dans le cas contraire, l'application affichera une erreur de refus d'autorisation.

Pour obtenir et afficher les emplacements de suivi, suivez cette procédure :

1. Pour obtenir les emplacements depuis l'appareil de l'utilisateur, vous devez fournir la date et l'heure de début et de fin. Un seul appel renvoie un maximum de 100 emplacements de suivi, mais s'il y en a plus de 100, il renverra une valeur `NextToken`. Vous devrez appeler les appels `getTrackerDevice`Location`` suivants avec `NextToken` pour charger plus de points de suivi pour les heures de début et de fin données.

```
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. Remplacez maintenant le code du `MapView.swift` fichier par le code suivant :

```

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]?)  
}
```

Pour localiser des valeurs de chaîne, procédez comme suit.

1. Créez et ajoutez un nouveau fichier appelé `Localizable.xcstrings`.
2. Cliquez avec le bouton droit sur le `Localizable.xcstrings` fichier et ouvrez-le en tant que code source.
3. Remplacez son contenu par le suivant :

```
{  
  "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. Enregistrez vos fichiers, puis créez et exécutez votre application pour prévisualiser les fonctionnalités.
5. Autorisez l'autorisation de localisation et appuyez sur le bouton de suivi. L'application commencera à télécharger les positions des utilisateurs et à les télécharger sur l'Amazon Location Tracker. Il affichera également les changements de localisation des utilisateurs, les points de suivi et l'adresse actuelle sur la carte.

Votre demande de démarrage rapide est terminée. Ce didacticiel vous explique comment créer une application iOS qui :

- Crée une carte avec laquelle les utilisateurs peuvent interagir.

- Gère plusieurs événements cartographiques associés à la modification de la vue cartographique par l'utilisateur.
- Appelle une API Amazon Location Service, spécifiquement pour effectuer une recherche sur la carte à un emplacement, à l'aide de l' `searchByPosition` API d'Amazon Location.

Quelle est la prochaine étape

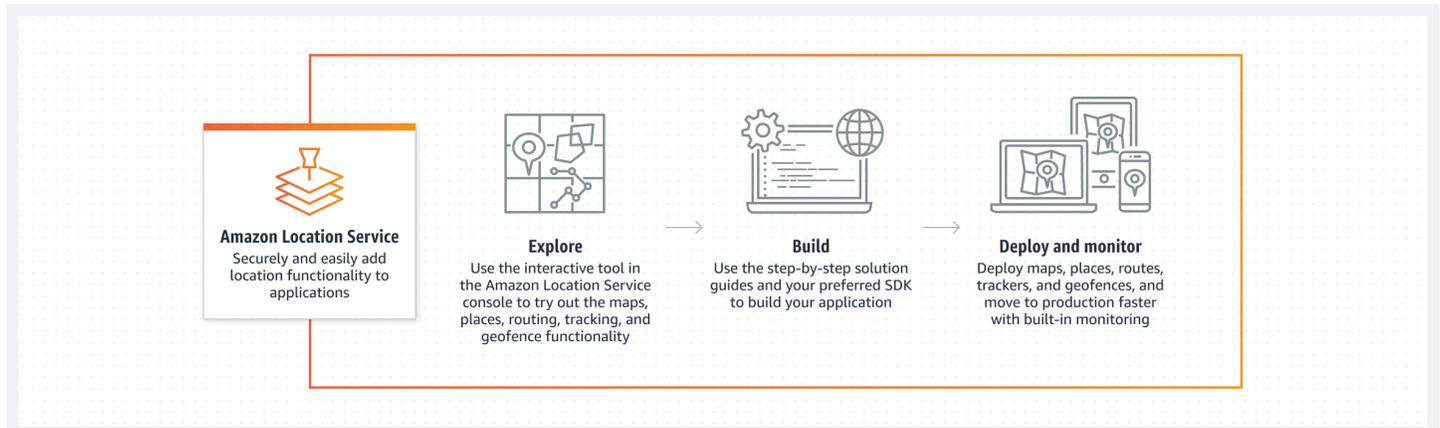
Le code source de cette application est disponible sur [GitHub](#).

Pour tirer le meilleur parti d'Amazon Location, vous pouvez consulter les ressources suivantes :

- Approfondissez les [concepts d'Amazon Location Service](#)
- En savoir plus sur [l'utilisation des fonctionnalités d'Amazon Location](#)
- Découvrez comment développer cet exemple et créer des applications plus complexes en consultant des [exemples de code à l'aide d'Amazon Location](#)

Concepts du service de location Amazon

Avec Amazon Location Service, vous pouvez ajouter des données de localisation à votre application en toute sécurité. Explorez certaines fonctionnalités à l'aide de l'[outil visuel et interactif](#), disponible sur la console Amazon Location. À l'aide de l'outil d'exploration, vous pouvez manipuler une carte par défaut, rechercher des points d'intérêt, tracer des barrières géographiques autour des zones d'intérêt et simuler l'envoi de positions d'appareils à un tracker.



Lorsque vous êtes prêt à créer, créez vos ressources et choisissez parmi une variété de styles de carte et de fournisseurs de données. Vous pouvez ensuite installer le SDK correspondant à votre environnement de développement et utiliser les API Amazon Location en suivant les instructions de ce guide. En outre, vous pouvez intégrer la surveillance à l'aide d'Amazon CloudWatch et AWS CloudTrail.

Les rubriques de cette section vous fournissent un aperçu des concepts fondamentaux d'Amazon Location et vous préparent à commencer à travailler avec la localisation dans vos propres applications.

Rubriques

- [Présentation d'Amazon Location](#)
- [Mappages](#)
- [Recherche de lieux](#)
- [Acheminements](#)
- [Geofences et traceurs](#)
- [Cas d'utilisation courants d'Amazon Location Service](#)
- [Qu'est-ce qu'un fournisseur de données ?](#)

- [Amazon Location, régions et points de terminaison](#)
- [Quotas d'Amazon Location Service](#)

Présentation d'Amazon Location

Amazon Location Service permet d'accéder à des fonctionnalités basées sur la localisation et à des fournisseurs de données par le biais AWS de ressources. Amazon Location propose cinq types de AWS ressources, en fonction du type de fonctionnalité dont vous avez besoin. Utilisez les différentes ressources ensemble pour créer une application complète basée sur la localisation. Vous pouvez créer une ou plusieurs de ces ressources à l'aide de la console Amazon Location, des API Amazon Location ou des SDK.

Chaque ressource définit le [fournisseur de données](#) sous-jacent à utiliser (le cas échéant) et donne accès aux fonctionnalités liées à son type.

Par exemple :

- [Amazon Location Service Maps](#) vous permet de choisir une carte auprès d'un fournisseur de cartes à utiliser sur votre application mobile ou Web.
- [Amazon Location Service Places](#) vous permet de choisir un fournisseur de données pour rechercher des points d'intérêt, compléter du texte partiel, effectuer le géocodage et le géocodage inversé.
- [Amazon Location Service Routes](#) vous permet de choisir un fournisseur de données, de rechercher des itinéraires et d'estimer le temps de trajet en fonction des informations up-to-date routières et du trafic en temps réel.
- Les [géofences Amazon Location Service](#) vous permettent de définir des zones d'intérêt sous forme de limites virtuelles. Vous pouvez ensuite évaluer les emplacements par rapport à eux et recevoir des notifications concernant les événements d'entrée et de sortie.
- Les [Amazon Location Service Trackers](#) reçoivent des mises à jour de localisation depuis vos appareils. Vous pouvez associer les traceurs aux collections de géofences afin que toutes les mises à jour de position soient automatiquement évaluées par rapport à vos géofences.

Vous pouvez utiliser les politiques IAM pour gérer et autoriser l'accès à vos ressources Amazon Location. Vous pouvez également organiser vos ressources en groupes de ressources afin de gérer et d'automatiser les tâches à mesure que le nombre de vos ressources augmente. Pour plus

d'informations sur la gestion AWS des ressources, consultez [What are AWS Resource Groups ?](#) Dans le guide de l'utilisateur d'AWS Resource Groups.

L'emplacement est défini à l'aide de coordonnées de latitude et de longitude conformes au [système géodésique mondial \(WGS 84\)](#), couramment utilisé comme système de référence de coordonnées standard pour les services du système de positionnement global (GPS).

Les sections suivantes décrivent le fonctionnement des composants d'Amazon Location.

Mappages

La ressource cartographique Amazon Location Service vous permet d'accéder aux données de fond de carte sous-jacentes d'une carte. Vous utilisez la ressource Map avec une bibliothèque de rendu cartographique pour ajouter une carte interactive à votre application. Vous pouvez ajouter d'autres fonctionnalités à votre carte, telles que des marqueurs (ou épingles), des itinéraires et des zones polygonales, selon les besoins de votre application.

Note

Pour plus d'informations sur l'utilisation pratique des ressources cartographiques, reportez-vous à [Utilisation d'Amazon Location Maps dans votre application](#).

Voici un aperçu de la création et de l'utilisation des ressources cartographiques :

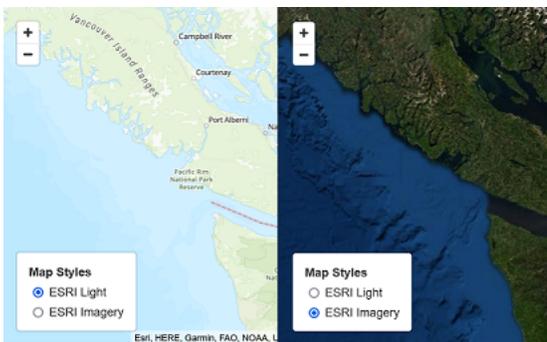


1. Vous créez une ressource cartographique dans votre AWS compte en sélectionnant un style de carte auprès d'un fournisseur de données.
2. Vous pouvez ensuite sélectionner et installer le SDK correspondant à votre environnement de développement et à vos applications. Pour plus d'informations sur les options disponibles, consultez la rubrique [Accès à Amazon Location](#).
3. Pour afficher une carte dans votre application, associez une ressource cartographique à une bibliothèque de rendu, telle qu'Amplify ou MapLibre Tangram. Pour plus d'informations, reportez-vous à la section [Utilisation de cartes](#) dans ce guide.

- Vous pouvez ensuite intégrer la surveillance en utilisant des services tels qu'Amazon CloudWatch et AWS CloudTrail Amazon Location. Pour plus d'informations, consultez [Surveillance d'Amazon Location Service avec Amazon CloudWatch](#) et [Enregistrement et surveillance avec AWS CloudTrail](#).

Styles de carte

Lorsque vous créez une ressource cartographique, vous devez choisir un style de carte pour cette ressource. Les styles de carte définissent l'apparence de la carte rendue. Par exemple, l'image suivante montre le même fournisseur de données avec deux styles différents issus de différentes ressources cartographiques dans Amazon Location. L'un des styles est un style de route typique, basé sur les données vectorielles de la carte. L'autre inclut des données matricielles présentant des images satellites. Le style peut changer lorsque vous zoomez ou dézoomez sur la carte, mais les styles ont généralement un thème cohérent. Il est possible de remplacer une partie ou la totalité des informations de style avant de les transmettre à la bibliothèque de rendu cartographique.



Opinions politiques

Certains styles de cartes d'Amazon Location Service prennent en compte d'autres points de vue politiques.

Note

L'opinion politique doit être utilisée conformément aux lois applicables, y compris les lois relatives à la cartographie du pays ou de la région où les cartes, images et autres données et contenus tiers auxquels vous accédez via Amazon Location Service sont mis à disposition.

Les styles de carte suivants reflètent le point de vue politique de l'Inde (IND).

- [Styles de carte Esri](#) :
 - Navigation Esri
 - Lampe Esri
 - Plan des rues Esri
 - Toile gris foncé Esri
 - Toile gris clair Esri
- [Styles de carte Open Data](#) :
 - Lumière standard pour les données ouvertes
 - Open Data Standard Dark
 - Lampe de visualisation des données ouvertes
 - Visualisation des données ouvertes dans le noir

Dans la console Amazon Location Service, vous pouvez filtrer les styles affichés pour afficher uniquement les styles qui reflètent le point de vue politique de l'Inde.

Couches personnalisées

Une couche personnalisée est une couche supplémentaire que vous pouvez activer pour un style de carte. Actuellement, seul le style de VectorEsriNavigation carte prend en charge la couche POI personnalisée.

Lorsque vous activez la couche POI personnalisée, elle ajoute un ensemble plus riche de lieux, tels que des boutiques, des services, des restaurants, des attractions et d'autres points d'intérêt à votre carte. Par défaut, la couche personnalisée est un `set`. Pour plus d'informations, voir [MapConfiguration](#) la référence de l'API de localisation.

Rendu cartographique

Pour afficher une carte dans votre application, vous utilisez généralement une bibliothèque de rendu de carte. Les bibliothèques peuvent utiliser plusieurs options courantes :

- MapLibre— MapLibre est une bibliothèque open source spécialement conçue pour le rendu de cartes interactives et constitue la méthode préférée pour le rendu de cartes à partir d'Amazon Location Service. MapLibre inclut la possibilité de restituer des données matricielles et vectorielles à partir d'une source de données (telle qu'une ressource cartographique Amazon Location). Vous pouvez étendre MapLibre pour dessiner vos propres données sur la carte.

- **Amplify** — Amplify est un framework open source permettant de créer des applications pour le Web, iOS, Android, etc. Si votre application utilise Amplify, vous pouvez l'étendre pour inclure la fonctionnalité Amazon Location. Amplify inclut des bibliothèques spécialement conçues pour créer des applications basées sur Amazon Location, y compris le rendu de cartes. Amplify est utilisé MapLibre pour afficher la carte, mais fournit des fonctionnalités supplémentaires spécifiques à Amazon Location Service pour une utilisation plus efficace, ainsi que des fonctionnalités de recherche et autres.
- **Tangram** — Tangram est une bibliothèque open source alternative qui produit des cartes interactives, similaires à. MapLibre

La bibliothèque de rendu cartographique extrait les données d'Amazon Location Service au moment de l'exécution, les rendant en fonction de la ressource cartographique que vous sélectionnez. La ressource cartographique définit le fournisseur de données et le style de carte qui seront utilisés.

L'image suivante montre comment la ressource cartographique est utilisée dans Amazon Location Service ainsi qu'une bibliothèque de rendu cartographique pour créer la carte finale.



1. Vous créez une ressource cartographique dans Amazon Location Service à l'aide du AWS Management Console ou AWS CLI. Cela définit le fournisseur de données et le style de carte que vous souhaitez utiliser.
2. Votre application inclut une bibliothèque de rendu cartographique. Vous donnez à la bibliothèque de rendu de carte le nom de la ressource cartographique à utiliser. La bibliothèque de rendu cartographique extrait les données et les informations de style relatives à cette ressource cartographique depuis Amazon Location et affiche la carte à l'écran.

Terminologie des cartes

Ressource cartographique

Vous permet d'accéder aux données cartographiques d'un fournisseur sélectionné. Utilisez la ressource cartographique pour récupérer des tuiles contenant des données cartographiques et un descripteur de style pour spécifier le mode de rendu des entités sur une carte.

Carte de base

Fournit un contexte géographique à votre carte, qui est stockée sous forme de couches de tuiles vectorielles. Les couches de tuiles incluent le contexte géographique tel que les noms de rues, les bâtiments et l'utilisation du sol à des fins de référence visuelle.

Vector

Les données vectorielles sont des données de forme composées de points, de lignes et de polygones. Il est souvent utilisé pour stocker et afficher des routes, des emplacements et des zones sur une carte. Une forme vectorielle peut également être utilisée comme icône pour les marqueurs sur une carte.

Raster

Les données matricielles sont des données d'image, constituées d'une grille, généralement de couleurs. Il est souvent utilisé pour stocker et afficher une représentation de données continues sur des cartes, telles que le terrain, l'imagerie satellite ou les cartes thermiques. Les images matricielles peuvent également être utilisées comme images ou icônes.

Style de carte

Les données vectorielles n'incluent pas intrinsèquement d'informations sur la manière de dessiner les couches de données pour créer la carte finale. Un style de carte définit la couleur et d'autres informations de style pour les données afin de définir leur apparence lors du rendu. Les ressources cartographiques incluent des informations de style pour la carte.

Amazon Location Service fournit des styles conformes à la [spécification de style Mapbox GL](#).

Tuile vectorielle

Format de mosaïque qui stocke les données cartographiques à l'aide de formes vectorielles. Ces données permettent d'obtenir une carte qui peut s'adapter à la résolution d'affichage et afficher les entités de différentes manières de manière sélective, tout en conservant une petite taille de fichier pour des performances optimales.

Format de fichier vectoriel pris en charge : Mapbox Vector Tiles (MVT).

Fichier de glyphes

Fichier binaire contenant des caractères Unicode codés. Utilisé par un moteur de rendu cartographique pour afficher les étiquettes.

Fichier Sprite

Fichier image PNG (Portable Network Graphic) qui contient de petites images matricielles, avec des descriptions de localisation dans un fichier JSON. Utilisé par un moteur de rendu de carte pour afficher des icônes ou des textures sur une carte.

Recherche de lieux

L'une des fonctions clés d'Amazon Location Service est la possibilité de rechercher les informations de géolocalisation. Amazon Location fournit cette fonctionnalité via la ressource d'index Place.

Note

Pour plus d'informations sur la manière d'utiliser les ressources de l'index des lieux pour effectuer des recherches dans la pratique, voir [Recherche de données de localisation et de géolocalisation à l'aide d'Amazon Location](#).

Vous pouvez utiliser les API d'index des lieux pour rechercher :

- Points d'intérêt, tels que les restaurants et les monuments. Effectuez une recherche par nom et par emplacement facultatif pour effectuer des recherches, et recevez une liste d'options classées par pertinence.

- Une adresse postale, recevant une latitude et une longitude pour cette adresse. C'est ce que l'on appelle le géocodage.
- Une position en latitude et en longitude, recevant l'adresse postale associée ou d'autres informations relatives à l'emplacement. C'est ce que l'on appelle le géocodage inversé.
- Une requête en texte libre partielle ou mal orthographiée, généralement lors de la saisie par l'utilisateur. C'est ce que l'on appelle la saisie semi-automatique, la suggestion automatique ou la correspondance floue.

L'index des lieux indique le fournisseur de données à utiliser pour la recherche.

Note

Les données cartographiques et autres informations de géolocalisation, y compris les emplacements exacts, peuvent varier selon les fournisseurs de données. Il est recommandé d'utiliser le même fournisseur de données pour votre index de lieux, votre carte et les autres ressources Amazon Location. Par exemple, si les lieux renvoyés par votre index des lieux ne correspondent pas à l'emplacement des mêmes lieux indiqué par votre ressource cartographique, vous pouvez placer un marqueur à ce qui semble être le mauvais endroit sur la carte.

Voici comment créer et utiliser des ressources d'index de lieux :



1. Tout d'abord, vous créez une ressource d'index des lieux dans votre AWS compte en sélectionnant un fournisseur de données.
2. Vous pouvez ensuite sélectionner et installer le SDK correspondant à votre environnement de développement et à vos applications. Pour plus d'informations sur les options disponibles, consultez la rubrique [Accès à Amazon Location](#).
3. Commencez à utiliser les API Amazon Location Places. Pour plus d'informations, consultez la rubrique sur l'utilisation de la [recherche par lieux](#).

4. Vous pouvez ensuite intégrer la surveillance à l'aide de services tels qu'Amazon CloudWatch et AWS CloudTrail. Pour plus d'informations, consultez [the section called “Surveillance avec CloudWatch”](#) et [the section called “Utilisation CloudTrail avec Amazon Location”](#).

Concepts de géocodage

Un index des lieux Amazon Location fournit une action appelée [SearchPlaceIndexForText](#) qui vous permet de spécifier le texte à rechercher. Par exemple, vous pouvez rechercher :

- Lieux : une recherche **Paris** peut renvoyer l'emplacement de la ville en France.
- Entreprises : une recherche de **coffee shop** pourrait renvoyer une liste de cafés, y compris leurs noms et adresses. Vous pouvez également spécifier un emplacement dans lequel effectuer la recherche ou un cadre dans lequel effectuer la recherche, afin de rendre les résultats plus pertinents. Dans ce cas, le fait de fournir un emplacement dans le centre-ville de Seattle, dans l'État de Washington, permettrait de rétablir les cafés de cette région.
- Adresses : une recherche **1600 Pennsylvania Ave, Washington D.C.** peut renvoyer l'emplacement de la Maison Blanche aux États-Unis (qui se trouve à cette adresse).

La recherche de texte de cette manière est généralement appelée géocodage, qui consiste à trouver un emplacement géographique pour l'adresse ou le lieu.

Amazon Location Service propose également une action de géocodage inverse appelée [SearchPlaceIndexForPosition](#). Cela prend un emplacement géographique et renvoie l'adresse, l'entreprise ou d'autres informations sur ce qui se trouve à cet endroit.

Résultats de recherche

Lorsque vous effectuez une demande de recherche réussie dans Amazon Location Service, un ou plusieurs résultats sont renvoyés. Chaque résultat inclut une étiquette, qui est le nom ou la description du résultat. Par exemple, une recherche sur **coffee shop**, peut renvoyer un résultat avec l'étiquette `Hometown Cafe` indiquant qu'un café appelé « Hometown Cafe » a été trouvé. Le résultat de la recherche inclut également généralement une adresse structurée (avec des propriétés telles que le numéro d'adresse, l'unité, la rue et le code postal). Selon le fournisseur de données, il inclura également d'autres métadonnées, telles que le pays et le fuseau horaire.

Pour une recherche sur le nom ou la catégorie d'une entreprise (par exemple **coffee shop**), vous souhaitez peut-être afficher tous les résultats renvoyés sur une carte. Pour une recherche

d'adresse, vous pouvez utiliser automatiquement le premier résultat. Consultez la rubrique suivante pour plus d'informations sur la pertinence.

Résultats multiples et pertinence

Lorsque vous effectuez une recherche par texte, Amazon Location Service trouve souvent plusieurs résultats. Par exemple, une recherche sur **Paris** peut renvoyer la ville en France, mais également la ville du Texas. Les résultats sont triés en fonction de leur pertinence, telle que déterminée par le fournisseur de données.

Note

Les résultats sont renvoyés par ordre de pertinence par tous les fournisseurs. Si vous choisissez Esri ou Grab comme fournisseur de données, les résultats incluent une valeur de pertinence que vous pouvez utiliser pour comprendre la pertinence relative entre les résultats d'une seule demande.

La spécification d'informations supplémentaires, telles que le nom d'un pays ou un lieu de recherche, peut modifier l'ordre des résultats, réduire le nombre de résultats ou même modifier l'ensemble des résultats renvoyés. Par exemple, une recherche **Paris** avec un lieu au Texas pour effectuer une recherche a plus de chances de revenir **Paris, Texas** comme premier résultat que **Paris, France**.

Dans une application interactive, vous pouvez utiliser la pertinence pour décider d'accepter le premier résultat ou de demander à un utilisateur de désambiguïser entre plusieurs résultats renvoyés. Si le premier résultat est très pertinent, vous pouvez simplement l'accepter comme la bonne réponse. S'il existe plusieurs résultats très pertinents, ou s'il n'y en a aucun, vous pouvez répertorier les résultats et laisser l'utilisateur sélectionner le meilleur résultat.

Résultats de l'adresse

Vous pouvez rechercher des adresses avec Amazon Location Service à l'aide de la même [SearchPlaceIndexForText](#) action. Plus vous fournissez d'informations, plus il est probable que l'adresse renvoyée corresponde à celle indiquée. Par exemple, **123 Main St** est moins susceptible de trouver un résultat correct que **123 Main St, Anytown, California, 90210**.

Les adresses ont plusieurs attributs, tels que le numéro de rue, la rue, la ville, la région, le code postal, etc. Ces attributs sont utilisés pour trouver une adresse dans l'index des lieux qui correspond

au plus grand nombre d'aspects possible. Plus le nombre d'attributs trouvés est élevé, plus la correspondance est considérée comme pertinente et plus elle est susceptible d'être renvoyée.

Note

La pertinence des résultats d'adresse dépend de la mesure dans laquelle le résultat correspond à l'entrée. Il peut s'agir du nombre d'attributs correspondants, mais également de la mesure dans laquelle les résultats correspondent à l'entrée. Par exemple, une entrée de **123 Main St** serait plus pertinente lorsqu'elle `Main St` est trouvée dans les données que si elle `Maine St` est le seul résultat. `Maine St` sera toujours renvoyé, mais probablement avec une valeur de pertinence inférieure.

Les résultats de recherche incluent une étiquette pour l'adresse complète (`123 Main St, Anytown, California, 90210`), mais également les attributs structurés individuels de l'adresse renvoyée. Cela est utile, car vous pouvez l'utiliser, par exemple, pour remplir les champs d'adresse d'une base de données ou pour examiner les résultats et trouver la ville, la région ou le code postal de l'emplacement trouvé.

Interpolation

Les données de l'index des adresses incluent les correspondances d'adresses exactes. Supposons, par exemple, qu'il y ait une rue `9th street` et qu'un bloc compte 2 maisons `240, 220` comme dans l'image suivante.



Le fournisseur de données crée les données de géolocalisation avec ces deux adresses connues. Vous pouvez rechercher ces deux adresses, et elles sont trouvées. Une fois que le fournisseur de données a créé les données cartographiques, supposons qu'une nouvelle maison soit ajoutée, entre les deux premières adresses. Cette nouvelle maison reçoit l'adresse 230. Si vous recherchez **230 S 9th St**, le fournisseur de données trouvera toujours un résultat. Au lieu d'utiliser une adresse connue, il effectue une interpolation entre les adresses déjà connues et estime la position de la nouvelle adresse à partir de celles-ci. Dans ce cas, il peut supposer que 230 se situe à mi-chemin entre 220 et 240 (et du même côté de la rue) et renvoyer une position approximative sur cette base.

Note

Les fournisseurs de données mettent régulièrement à jour leurs données de géolocalisation avec de nouvelles adresses. Dans ce cas, 230 S 9th St elle sera ajoutée aux données du fournisseur de données, mais il y aura généralement une période pendant laquelle une nouvelle adresse a été créée mais n'est pas encore ajoutée aux données.

Dans ce cas, le fournisseur de données ne peut pas dire si la nouvelle adresse existe dans le monde, car elle ne figure pas encore dans les données, mais fournit la meilleure réponse possible à partir des informations dont il dispose. Ce résultat est appelé interpolé et peut être renvoyé par le fournisseur de données dans les résultats. En cas de `interpolated` retour `false`, il s'agit d'une adresse

connue. S'il revient true, il s'agit d'une adresse approximative. S'il n'est pas renvoyé, le fournisseur de données n'a pas indiqué si le résultat provenait d'une interpolation.

Important

Le fournisseur de données peut également renvoyer des résultats interpolés pour des adresses qui n'existent pas du tout. Par exemple, dans ce cas, si vous l'avez saisie **232 S 9th St**, le fournisseur trouverait cette adresse inexistante et renverrait une localisation proche de 230, mais du côté 240. Les adresses interpolées sont utiles pour vous amener au bon endroit, mais il est bon de garder à l'esprit qu'il ne s'agit pas d'adresses connues.

Stockage des résultats de géocodage

Lorsque vous créez une ressource d'index géographique, vous devez spécifier une option de stockage de données (appelée `IntendedUse` dans l'API). Il peut s'agir de résultats à usage unique ou de résultats enregistrés. Il s'agit d'une question sur l'utilisation que vous comptez faire des résultats. Si vous souhaitez stocker les résultats (même à des fins de mise en cache), vous devez choisir l'option de stockage, et non l'option à usage unique.

Note

Lorsque vous avez choisi l'option stockée (intitulée `Oui`, les résultats seront stockés dans la console ou sélectionnée `storage` dans `CreatePlaceIndexAPI`), Amazon Location Service ne stocke pas les résultats pour vous. Cela indique que vous prévoyez de stocker les résultats.

Lorsque vous déterminez comment vous allez utiliser les résultats de vos requêtes adressées à Amazon Location Service, vous devez toujours connaître les [conditions de AWS service](#) applicables.

Terminologie des lieux

Placer une ressource d'index

Vous permet de choisir une source de données pour prendre en charge les requêtes de recherche. Par exemple, vous pouvez rechercher des points d'intérêt, des adresses ou des coordonnées. Lorsqu'une requête de recherche est envoyée à une ressource d'index géographique, elle est traitée à l'aide de la source de données configurée de la ressource.

Géocodage

Le géocodage est le processus qui consiste à saisir un texte, à le rechercher dans l'index des lieux et à renvoyer les résultats avec position.

Géocodage inversé

Le géocodage inversé est le processus qui consiste à prendre une position et à renvoyer des informations relatives à cette position à partir de l'index des lieux, telles que l'adresse, la ville ou le commerce de cette position.

Pertinence

La pertinence correspond à la mesure dans laquelle un résultat correspond à l'entrée. Ce n'est pas une mesure de l'exactitude.

Interpolation

L'interpolation est le processus qui consiste à trouver des adresses inconnues en utilisant des emplacements d'adresses connus comme points de repère.

Codes de pays ISO 3166

Amazon Location Service Places utilise [les codes de pays de l'Organisation internationale de normalisation \(ISO\) 3166](#) pour désigner des pays ou des régions.

Pour trouver le code d'un pays ou d'une région spécifique, utilisez la [plateforme de navigation en ligne ISO](#).

Acheminements

Cette section fournit un aperçu des concepts relatifs au routage à l'aide d'Amazon Location Service.

Note

Pour plus d'informations sur la façon d'utiliser les ressources d'itinéraire dans la pratique, voir [Calcul des itinéraires à l'aide d'Amazon Location Service](#).

Ressources pour le calculateur d'itinéraires

Les ressources du calculateur d'itinéraires vous permettent de trouver des itinéraires et d'estimer le temps de trajet en fonction du réseau up-to-date routier et des informations routières en temps réel fournies par le fournisseur de données que vous avez choisi.

Vous pouvez utiliser les API Routes pour créer des fonctionnalités qui permettent à votre application de demander le temps de trajet, la distance et la géométrie de l'itinéraire entre deux emplacements. Vous pouvez également utiliser l'API Routes pour demander le temps de trajet et la distance entre un ensemble de départs et de destinations en une seule demande afin de calculer une matrice.

Ce qui suit explique comment créer et utiliser une ressource de calcul d'itinéraire :



1. Tout d'abord, vous créez une ressource de calcul d'itinéraire dans votre AWS compte en sélectionnant un fournisseur de données.
2. Vous pouvez ensuite sélectionner et installer le SDK correspondant à votre environnement de développement et à vos applications.
3. Commencez à utiliser les API Amazon Location Routes. Pour plus d'informations sur l'utilisation des API de routage, consultez la rubrique sur [Calcul des itinéraires à l'aide d'Amazon Location Service](#).
4. Vous pouvez ensuite intégrer la surveillance à l'aide de services tels qu'Amazon CloudWatch et AWS CloudTrail. Pour plus d'informations, consultez [Surveillance d'Amazon Location Service avec Amazon CloudWatch](#) et [Enregistrement et surveillance avec AWS CloudTrail](#).

Calcul d'un itinéraire

Une ressource de calcul d'itinéraire Amazon Location fournit une action appelée `CalculateRoute` que vous pouvez utiliser pour créer un itinéraire entre deux emplacements géographiques (le point de départ et la destination). L'itinéraire calculé inclut la géométrie permettant de tracer l'itinéraire sur une carte, ainsi que le temps total et la distance parcourus par l'itinéraire.

Utiliser des waypoints

Lorsque vous créez votre demande d'itinéraire, vous pouvez ajouter des points de cheminement supplémentaires à l'itinéraire. Il s'agit de points situés entre le départ et la destination qui font office d'arrêts le long du parcours. L'itinéraire sera calculé à partir de chacun des points de cheminement spécifiés. L'itinéraire d'un point de la demande au suivant est appelé Leg a. Chaque étape inclut une distance, une durée et la géométrie de cette partie du parcours.

Note

Les points de cheminement sont routés dans l'ordre indiqué dans la demande. Ils ne sont pas réorganisés en fonction du chemin le plus court. Consultez la [Planification des itinéraires](#) section pour plus d'informations sur la recherche du chemin le plus court.

Vous pouvez inclure jusqu'à 25 points de cheminement dans une seule demande pour calculer un itinéraire.

Trafic et heure de départ

L'Amazon Location Service prend en compte le trafic lors du calcul d'un itinéraire. Le trafic qu'il prend en compte est basé sur l'heure que vous spécifiez. Vous pouvez choisir de partir maintenant ou indiquer une heure précise à laquelle vous souhaitez partir, ce qui aura une incidence sur le résultat de l'itinéraire en ajustant le trafic à l'heure spécifiée.

Note

Vous pouvez calculer l'heure d'arrivée en utilisant l'heure de départ et le temps de réponse de l'itinéraire, pour estimer l'arrivée d'un chauffeur, par exemple.

Si vous souhaitez qu'Amazon Location ne prenne pas en compte le trafic, ne spécifiez pas d'heure de départ et ne spécifiez pas d'heure de départ maintenant. Cela permettra de calculer un itinéraire qui suppose les meilleures conditions de circulation pour l'itinéraire.

Options du mode de voyage

Vous pouvez définir le mode de voyage lorsque vous calculez un itinéraire à l'aide d'Amazon Location Service. Le mode de déplacement par défaut est la voiture, mais vous pouvez également sélectionner le camion ou la marche à pied.

Si vous spécifiez le mode voiture ou camion, vous pouvez également définir des options supplémentaires.

Pour le mode voiture, vous pouvez spécifier que vous souhaitez éviter les routes à péage ou les ferries. Cela permettra d'éviter les ferries et les routes à péage, tout en continuant à les emprunter, s'ils constituent le seul moyen de se rendre à destination.

Pour le mode camion, vous pouvez également éviter les ferries et les routes à péage, mais vous pouvez également spécifier la taille et le poids du camion, afin d'éviter les itinéraires qui ne conviennent pas au camion.

Planification des itinéraires

Vous pouvez utiliser Amazon Location Service pour créer des entrées pour votre logiciel de planification et d'optimisation d'itinéraires. Vous pouvez créer des résultats d'itinéraire, y compris le temps de trajet et la distance parcourue, pour des itinéraires entre un ensemble de positions de départ et un ensemble de positions de destination. C'est ce qu'on appelle créer une matrice de route.

Note

Les logiciels de planification et d'optimisation d'itinéraires peuvent résoudre de nombreux scénarios. Par exemple, un logiciel de planification peut utiliser l'ensemble des temps et des distances entre les points pour calculer le chemin le plus court qui s'arrête à chaque point, fournissant ainsi un itinéraire efficace pour un seul conducteur. Un logiciel de planification peut également être utilisé pour répartir les arrêts entre plusieurs camions, ce qui permet d'améliorer l'efficacité d'une flotte ou de s'assurer que chaque client est visité dans les délais impartis. Amazon Location fournit les fonctions de routage de manière efficace pour permettre au logiciel de planification de mener à bien sa tâche.

Par exemple, compte tenu des positions de départ A et B et des positions de destination X et Y, Amazon Location Service indiquera le temps de trajet et la distance parcourue pour les itinéraires allant de A à X, A à Y, B à X et B à Y.

Comme pour le calcul d'un seul itinéraire, vous pouvez calculer les itinéraires avec différents modes de transport, évitements et conditions de circulation. Par exemple, vous pouvez spécifier que le véhicule est un camion de 35 pieds de long, et l'itinéraire calculé utilisera ces restrictions pour déterminer le temps et la distance parcourus. Vous ne pouvez pas inclure de points de cheminement dans le calcul d'une matrice d'itinéraires.

Le nombre de résultats renvoyés (et d'itinéraires calculés) est le nombre de positions de départ multiplié par le nombre de positions de destination. Vous êtes facturé pour chaque itinéraire calculé, et non pour chaque demande adressée au service. Ainsi, une matrice d'itinéraires comportant 10 départs et 10 destinations sera facturée comme 100 itinéraires.

Terminologie des itinéraires

Ressource de calcul d'itinéraire

Une AWS ressource qui vous permet d'estimer le temps de trajet, la distance et de tracer des itinéraires sur une carte à l'aide des données relatives au trafic et au réseau routier provenant du fournisseur de données de votre choix.

À l'aide des ressources du calculateur d'itinéraires, vous calculez des itinéraires pour différents modes de transport, détours et conditions de circulation.

Acheminement

Un itinéraire contient les détails utilisés lorsque vous empruntez un chemin à partir de la position de départ, des positions des points de cheminement et de la position de destination.

Voici des exemples de détails d'un itinéraire :

- Distance entre une position et une autre position.
- Le temps nécessaire pour passer d'une position à la suivante.
- LineString Géométrie représentant le tracé de l'itinéraire.

Pour plus d'informations sur les itinéraires, consultez la [syntaxe de réponse pour l' CalculateRoute opération](#) dans la référence de l'API Amazon Location Service Routes.

Matrice d'itinéraires

Une liste d'itinéraires, allant d'un ensemble de positions de départ à un ensemble de positions de destination. Utile comme entrée dans un logiciel de planification ou d'optimisation d'itinéraires.

Pour plus d'informations sur le calcul d'une matrice d'itinéraires, consultez la [syntaxe de l' CalculateRouteMatrix opération](#) dans la référence de l'API Amazon Location Service Routes.

LineStringgéométrie

Un itinéraire Amazon Location comprend une ou plusieurs étapes (un itinéraire allant d'un point de cheminement à un autre dans le cadre de l'itinéraire global). La géométrie de chaque jambe

est une polygline représentée par un `LineString`. A `LineString` est un tableau ordonné de positions qui peut être utilisé pour tracer un itinéraire sur une carte.

Voici un exemple de modèle `LineString` à trois points :

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

Waypoint

Les points de cheminement sont des positions intermédiaires qui font office d'arrêts le long d'un itinéraire entre la position de départ et la position de destination. L'ordre des escales sur l'itinéraire suit l'ordre dans lequel vous indiquez les positions des points de cheminement dans la demande.

Jambe

Une seule étape est le voyage d'une position à une autre. Si les positions ne sont pas situées sur une route, elles sont déplacées vers la route la plus proche. Le nombre d'étapes d'un parcours est inférieur d'un au nombre total de positions.

Un itinéraire sans point de passage consiste en une seule étape, du point de départ à la destination. Un itinéraire comportant un point de cheminement comprend deux étapes, de la position de départ au point de cheminement, puis du point de cheminement à la destination.

Étape

Une étape est une sous-section d'une jambe. Chaque étape fournit des informations récapitulatives pour cette étape de la jambe.

Geofences et traceurs

Cette section fournit un aperçu des concepts liés à l'utilisation des géofences et des traceurs Amazon Location Service. Les géofences sont des limites de polygones que vous pouvez utiliser pour être averti lorsque des appareils ou des positions entrent ou sortent des zones. Les ressources de suivi sont utilisées pour stocker et mettre à jour les positions des appareils lorsqu'ils se déplacent.

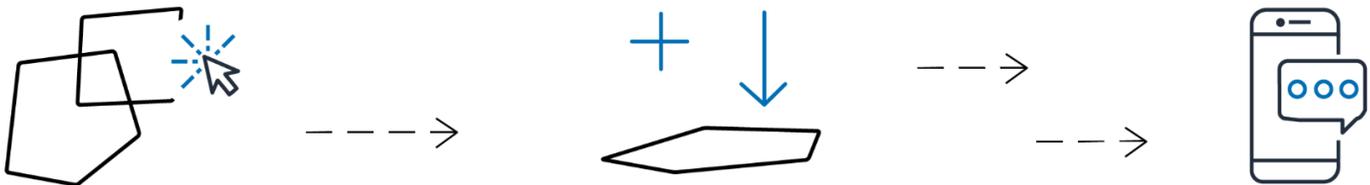
Note

Pour plus d'informations sur la façon d'utiliser les géofences et les traceurs dans la pratique, voir. [Géolocalisation d'une zone d'intérêt à l'aide d'Amazon Location](#)

Géofences

Les ressources de collecte de géofences vous permettent de stocker et de gérer des géofences, des limites virtuelles sur une carte. Vous pouvez évaluer les emplacements par rapport à une ressource de collecte de géofences et recevoir des notifications lorsque la mise à jour de localisation franchit les limites de l'une des géofences de la collection de géofences.

Ce qui suit explique comment créer et utiliser des ressources de collecte de géofences :



1. Créez une ressource de collecte de géofences dans votre AWS compte.
2. Ajoutez des géofences à cette collection. Vous pouvez le faire soit en utilisant l'outil de téléchargement de géofences sur la console Amazon Location, soit en utilisant l'API Amazon Location Geofences. Pour plus d'informations sur les options disponibles, consultez [Accès à l'emplacement Amazon](#).

Les géofences peuvent être définies par un polygone ou par un cercle. Utilisez un polygone pour savoir quand un appareil entre dans une zone spécifique. Utilisez un cercle pour savoir quand un appareil se trouve à une certaine distance (rayon) d'un point.

3. Vous pouvez commencer à évaluer les emplacements par rapport à toutes vos barrières géographiques. Lorsqu'une mise à jour de localisation franchit les limites d'une ou de plusieurs géofences, votre ressource de collecte de géofences émet l'un des types d'événements de géofences suivants sur Amazon : EventBridge
 - ENTER — Un événement est généré pour chaque géofence lorsque la mise à jour de localisation franchit ses limites en la saisissant.

- SORTIE — Un événement est généré pour chaque géofence lorsque la mise à jour de localisation franchit ses limites en la quittant.

Pour plus d'informations, consultez [the section called “Réagir aux événements avec EventBridge”](#). Vous pouvez également intégrer la surveillance à l'aide de services tels qu'Amazon CloudWatch et AWS CloudTrail. Pour plus d'informations, consultez [the section called “Surveillance avec CloudWatch”](#) et [the section called “Utilisation CloudTrail avec Amazon Location”](#).

Par exemple, si vous suivez une flotte de camions et que vous souhaitez être averti lorsqu'un camion arrive dans une certaine zone de l'un de vos entrepôts. Vous pouvez créer une géofence pour la zone autour de chaque entrepôt. Ensuite, lorsque les camions vous enverront des positions mises à jour, vous pouvez utiliser Amazon Location Service pour évaluer ces positions et voir si un camion est entré (ou sorti) de l'une des zones de géolocalisation.

Note

Vous êtes facturé en fonction du nombre de collections de géofences par rapport auxquelles vous évaluez. Votre facture n'est pas affectée par le nombre de géofences dans chaque collection. Étant donné que chaque collection de géofences peut contenir jusqu'à 50 000 géofences, vous souhaitez peut-être combiner vos géofences en un moins grand nombre de collections, dans la mesure du possible, afin de réduire le coût des évaluations de géofences. Les événements générés incluront l'ID de la géofence individuelle de la collection, ainsi que l'ID de la collection.

Événements Geofence

Les emplacements des positions que vous surveillez sont référencés par un identifiant appelé a DeviceId (et les positions sont appelées positions des appareils). Vous pouvez envoyer une liste des positions des appareils à évaluer directement à la ressource de collecte de géofences, ou vous pouvez utiliser un traceur. Consultez la section suivante pour plus d'informations sur les trackers.

Vous recevez des événements (via Amazon EventBridge) uniquement lorsqu'un appareil entre ou sort d'une géofence, et non pour chaque changement de position. Cela signifie que vous recevrez généralement des événements et que vous devrez y répondre beaucoup moins fréquemment qu'à chaque mise à jour de la position de l'appareil.

Note

Pour la première évaluation de l'emplacement d'un appareil spécifique `DeviceID`, il est supposé que l'appareil ne se trouvait auparavant dans aucune zone géographique. Ainsi, la première mise à jour générera un ENTER événement, s'il se trouve à l'intérieur d'une barrière géographique de la collection, et aucun événement dans le cas contraire.

Afin de calculer si un appareil est entré dans une géofence ou en est sorti, Amazon Location Service doit conserver l'état de position précédent de l'appareil. Cet état de position est conservé pendant 30 jours. Après 30 jours sans mise à jour pour un appareil, une nouvelle mise à jour de localisation sera traitée comme la première mise à jour de position.

Traceurs

Un tracker enregistre les mises à jour de position d'un ensemble d'appareils. Le tracker peut être utilisé pour demander la position actuelle ou l'historique des positions des appareils. Il stocke les mises à jour, mais réduit l'espace de stockage et le bruit visuel en filtrant les emplacements avant de les stocker.

Chaque mise à jour de position stockée dans les ressources de votre tracker peut inclure une mesure de la précision de la position et jusqu'à 3 champs de métadonnées concernant la position ou l'appareil que vous souhaitez stocker. Les métadonnées sont stockées sous forme de paires clé-valeur et peuvent stocker des informations telles que la vitesse, la direction, la pression des pneus ou la température du moteur.

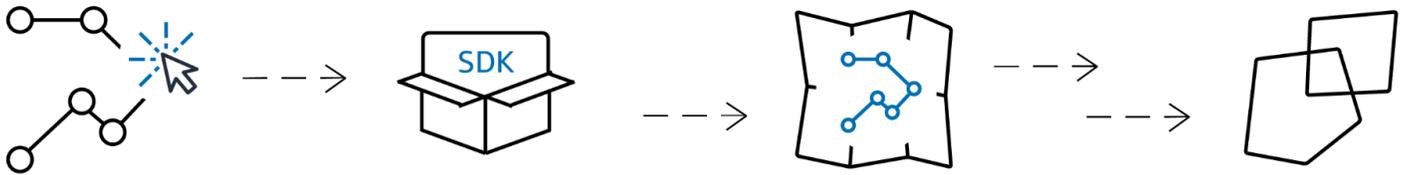
Note

Le stockage du tracker est automatiquement chiffré avec les clés AWS détenues. Vous pouvez ajouter une couche de chiffrement supplémentaire à l'aide des clés KMS que vous gérez, afin de garantir que vous êtes le seul à pouvoir accéder à vos données. Pour plus d'informations, consultez [Chiffrement des données au repos pour Amazon Location Service](#).

Le filtrage et le stockage de la position des traceurs sont utiles en eux-mêmes, mais les traceurs sont particulièrement utiles lorsqu'ils sont associés à des géofences. Vous pouvez associer les traceurs à une ou plusieurs de vos ressources de collecte de géofences, et les mises à jour de position sont

évaluées automatiquement par rapport aux géofences de ces collections. L'utilisation appropriée du filtrage peut également réduire considérablement les coûts de vos évaluations de géofences.

Le schéma suivant montre comment créer et utiliser des ressources de suivi :



1. Tout d'abord, vous créez une ressource de suivi dans votre AWS compte.
2. Décidez ensuite de la manière dont vous souhaitez envoyer les mises à jour de localisation à vos ressources de suivi. Utilisez [AWS les SDK](#) pour intégrer des fonctionnalités de suivi dans vos applications mobiles. Vous pouvez également utiliser le MQTT en suivant les step-by-step instructions de [suivi à l'aide](#) du MQTT.
3. Vous pouvez désormais utiliser votre ressource de suivi pour enregistrer l'historique des positions et le visualiser sur une carte.
4. Vous pouvez également lier votre ressource de suivi à une ou plusieurs collections de géofences afin que chaque mise à jour de position envoyée à votre ressource de suivi soit automatiquement évaluée par rapport à toutes les barrières géographiques de toutes les collections de géofences liées. Vous pouvez lier une ressource sur la page de détails des ressources de suivi de la console Amazon Location ou en utilisant l'API Amazon Location Trackers.
5. Vous pouvez ensuite intégrer la surveillance à l'aide de services tels qu'Amazon CloudWatch et AWS CloudTrail. Pour plus d'informations, consultez [the section called "Surveillance avec CloudWatch"](#) et [the section called "Utilisation CloudTrail avec Amazon Location"](#).

Utilisation de traceurs avec géofences

Les traceurs fournissent des fonctionnalités supplémentaires lorsqu'ils sont associés à des barrières géographiques. Vous associez un traceur à une collection de géofences, via la console Amazon Location ou l'API, pour évaluer automatiquement les emplacements des traceurs. Chaque fois que le traceur reçoit une position mise à jour, cette position est évaluée par rapport à chaque géofence de la collection, et les EXIT événements appropriés ENTER sont générés sur Amazon. EventBridge Vous pouvez également appliquer un filtrage au traceur et, en fonction du filtrage, vous pouvez réduire les coûts des évaluations de géofences en évaluant uniquement les mises à jour de localisation significatives.

Si vous associez le traceur à une collection de géofences alors qu'il a déjà reçu des mises à jour de position, la première mise à jour de position après l'association est traitée comme une mise à jour initiale pour les évaluations de géofences. S'il se trouve à l'intérieur d'une géofence, vous recevrez un ENTER événement. S'il ne se trouve dans aucune zone géographique, vous ne recevrez aucun EXIT événement, quel que soit l'état précédent.

Filtrage de position

Les trackers peuvent filtrer automatiquement les positions qui leur sont envoyées. Il existe plusieurs raisons pour lesquelles vous souhaitez peut-être filtrer certaines mises à jour de localisation de votre appareil. Si votre système n'envoie des rapports que toutes les minutes environ, vous pouvez filtrer les appareils par heure, en stockant et en évaluant les positions toutes les 30 secondes uniquement. Même si vous effectuez une surveillance plus fréquente, vous souhaitez peut-être filtrer les mises à jour de position pour atténuer le bruit du matériel GPS. Les positions GPS sont intrinsèquement bruyantes. Leur précision n'est pas parfaite à 100 %, de sorte que même un appareil immobile semble légèrement bouger. À basse vitesse, cette instabilité crée un encombrement visuel et peut provoquer de fausses entrées et sorties si l'appareil se trouve à proximité du bord d'une clôture géographique.

Le filtrage de position fonctionne lorsque les mises à jour de position sont reçues par un traceur, ce qui réduit le bruit visuel sur les trajectoires de votre appareil (instabilité), réduit le nombre de faux événements d'entrée et de sortie de barrières géographiques et aide à gérer les coûts en réduisant le nombre de mises à jour de position stockées et d'évaluations de géofences déclenchées.

Les trackers proposent trois options de filtrage des positions pour vous aider à gérer les coûts et à réduire l'agitation lors des mises à jour de votre position.

- Basé sur la précision — À utiliser avec n'importe quel appareil fournissant une mesure précise. La plupart des GPS et appareils mobiles fournissent ces informations. La précision de chaque mesure de position dépend de nombreux facteurs environnementaux, notamment la réception des satellites GPS, le paysage et la proximité des appareils Wi-Fi et Bluetooth. La plupart des appareils, y compris la plupart des appareils mobiles, peuvent fournir une estimation de la précision de la mesure en même temps que la mesure. Grâce au AccuracyBased filtrage, Amazon Location ignore les mises à jour de localisation si le déplacement de l'appareil est inférieur à la précision mesurée. Par exemple, si deux mises à jour consécutives d'un appareil ont une plage de précision de 5 m et 10 m, Amazon Location ignore la deuxième mise à jour si l'appareil s'est déplacé de moins de 15 m. Amazon Location n'évalue pas les mises à jour ignorées par rapport aux géofences et ne les stocke pas.

Lorsque la précision n'est pas fournie, elle est considérée comme nulle, et la mesure est considérée comme parfaitement précise, et aucun filtrage ne sera appliqué aux mises à jour.

Note

Vous pouvez utiliser le filtrage basé sur la précision pour supprimer tous les filtres. Si vous sélectionnez le filtrage basé sur la précision, mais que vous remplacez toutes les données de précision à zéro, ou que vous omettez complètement la précision, Amazon Location ne filtrera aucune mise à jour.

Dans la plupart des scénarios, le filtrage basé sur la précision est un bon choix pour filtrer les mises à jour de position, en fournissant un équilibre entre le suivi de la localisation et le filtrage des mises à jour inutiles, réduisant ainsi les coûts.

- Basé sur la distance : à utiliser lorsque vos appareils ne fournissent pas de mesure précise, mais que vous souhaitez tout de même tirer parti du filtrage pour réduire l'instabilité et gérer les coûts. `DistanceBased` le filtrage ignore les mises à jour de localisation dans lesquelles les appareils se sont déplacés à moins de 30 m (98,4 pieds). Lorsque vous utilisez le filtrage de `DistanceBased` position, Amazon Location n'évalue pas ces mises à jour ignorées par rapport aux géofences et ne stocke pas les mises à jour.

La précision de la plupart des appareils mobiles, y compris la précision moyenne des appareils iOS et Android, est inférieure à 15 m. Dans la plupart des applications, le `DistanceBased` filtrage permet de réduire l'effet des inexactitudes de localisation lors de l'affichage de la trajectoire des appareils sur une carte, ainsi que l'effet de rebond de plusieurs événements consécutifs d'entrée et de sortie lorsque les appareils se trouvent à proximité de la limite d'une géofence. Cela peut également contribuer à réduire le coût de votre application, en effectuant moins d'appels pour effectuer une évaluation par rapport à des géofences liées ou pour récupérer les positions des appareils.

Le filtrage basé sur la distance est utile si vous souhaitez filtrer, mais que votre appareil ne fournit pas de mesures de précision, ou si vous souhaitez filtrer un plus grand nombre de mises à jour que le filtrage basé sur la précision.

- Basé sur le temps — (par défaut) À utiliser lorsque vos appareils envoient des mises à jour de position très fréquemment (plus d'une fois toutes les 30 secondes) et que vous souhaitez réaliser des évaluations de géofences en temps quasi réel sans enregistrer chaque mise à jour. Dans le

cadre du TimeBased filtrage, chaque mise à jour de localisation est évaluée par rapport à des collections de géofences liées, mais toutes les mises à jour de localisation ne sont pas stockées. Si votre fréquence de mise à jour est supérieure à 30 secondes, une seule mise à jour toutes les 30 secondes est enregistrée pour chaque ID d'appareil unique.

Le filtrage basé sur le temps est particulièrement utile lorsque vous souhaitez stocker moins de positions, mais que vous souhaitez que chaque mise à jour des positions soit évaluée par rapport aux collections de géofences associées.

Note

Tenez compte des coûts de votre application de suivi lorsque vous décidez de votre méthode de filtrage et de la fréquence des mises à jour de position. Vous êtes facturé pour chaque mise à jour de position et une fois pour l'évaluation de la mise à jour de position par rapport à chaque collection de géofences liée. Par exemple, lorsque vous utilisez le filtrage basé sur le temps, si votre traceur est lié à deux collections de géofences, chaque mise à jour de position comptera comme une demande de mise à jour de localisation et deux évaluations de collections de géofences. Si vous signalez des mises à jour de position toutes les 5 secondes pour vos appareils et que vous utilisez un filtrage basé sur le temps, 720 mises à jour de localisation et 1 440 évaluations de géofences par heure vous seront facturées pour chaque appareil.

Terminologie Geofence

Collection Geofence

Ne contient aucune géofence ou plus. Il est capable de surveiller les géofences en émettant des événements d'entrée et de sortie, sur demande, pour évaluer la position d'un appareil par rapport à ses géofences.

Geofence

Géométrie polygonale ou circulaire qui définit une limite virtuelle sur une carte.

Géométrie des polygones

Une clôture géographique Amazon Location est la limite virtuelle d'une zone géographique et est représentée sous la forme d'une géométrie polygonale ou d'un cercle.

Un cercle est un point entouré d'une distance. Utilisez un cercle lorsque vous souhaitez être averti si un appareil se trouve à une certaine distance d'une position.

Un polygone est un réseau composé d'un ou de plusieurs anneaux linéaires. Utilisez un polygone lorsque vous souhaitez définir une limite spécifique pour les notifications de l'appareil. Un anneau linéaire est un ensemble de quatre sommets ou plus, le premier et le dernier sommet étant identiques pour former une limite fermée. Chaque sommet est un point bidimensionnel de la forme `[longitude, latitude]`, où les unités de longitude et de latitude sont les degrés. Les sommets doivent être listés dans le sens inverse des aiguilles d'une montre autour du polygone.

Note

Amazon Location Service ne prend pas en charge les polygones comportant plusieurs anneaux. Cela inclut les trous, les îles ou les multipolygones. Amazon Location ne prend pas non plus en charge les polygones enroulés dans le sens des aiguilles d'une montre ou qui traversent l'antiméridien.

Voici un exemple d'anneau externe linéaire unique :

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

Terminologie des traceurs

Ressource de suivi

AWS Ressource qui reçoit les mises à jour de localisation des appareils. La ressource de suivi fournit une assistance pour les requêtes de localisation, telles que la localisation actuelle et

historique de l'appareil. L'association d'une ressource de suivi à une collection de géofences évalue automatiquement les mises à jour de localisation par rapport à toutes les géofences de la collection de géofences liée.

Données de position suivies

Une ressource de suivi stocke des informations sur vos appareils au fil du temps. Les informations incluent une série de mises à jour de position, chaque mise à jour incluant le lieu, l'heure et des métadonnées facultatives. Les métadonnées peuvent inclure la précision d'une position et jusqu'à trois paires clé-valeur pour vous aider à suivre les informations clés relatives à chaque position, telles que la vitesse, la direction, la pression des pneus, le carburant restant ou la température du moteur du véhicule que vous suivez. Les trackers conservent l'historique de localisation de l'appareil pendant 30 jours.

Filtrage de position

Le filtrage des positions peut vous aider à contrôler les coûts et à améliorer la qualité de votre application de suivi en filtrant les mises à jour de position qui ne fournissent pas d'informations précieuses avant qu'elles ne soient stockées ou évaluées par rapport aux géofences.

Vous pouvez choisir `AccuracyBased` ou `TimeBased` filtrer. `DistanceBased` Par défaut, le filtrage des positions est défini sur `TimeBased`.

Vous pouvez configurer le filtrage des positions lorsque vous créez ou mettez à jour des ressources de suivi.

Format d'horodatage RFC 3339

Amazon Location Service Trackers utilise le format [RFC 3339](#), qui suit le format ISO [8601 de l'Organisation internationale de normalisation \(ISO\) pour](#) les dates et les heures.

Le format est « YYYY-MM-DDTHH:MM:SS.sssZ+ 00:00 » :

- YYYY-MM-DD— Représente le format de date.
- T— Indique que les valeurs temporelles suivront.
- hh:mm:ss.sss— Représente l'heure au format 24 heures.
- Z— Indique que le fuseau horaire utilisé est l'UTC, qui peut être suivi d'écarts par rapport au fuseau horaire UTC.
- +00:00— Indiquez éventuellement les écarts par rapport au fuseau horaire UTC. Par exemple, + 01:00 indique UTC + 1 heure.

Exemple

Le 2 juillet 2020, à 12 h 15 min 20 s, avec un ajustement d'une heure supplémentaire par rapport au fuseau horaire UTC.

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

Cas d'utilisation courants d'Amazon Location Service

Amazon Location Service vous permet de créer une gamme d'applications, allant du suivi des actifs au marketing géolocalisé. Les cas d'utilisation les plus courants sont les suivants :

Engagement des utilisateurs et géomarketing

Utilisez les données de localisation pour créer des solutions qui améliorent l'engagement des utilisateurs dans le marketing afin de cibler les clients. Par exemple, Amazon Location peut déclencher un événement qui déclenche une notification lorsqu'un client ayant commandé un café sur son application mobile se trouve à proximité. En outre, vous pouvez créer des fonctionnalités de ciblage géographique afin que les détaillants puissent envoyer des codes de réduction ou des dépliants numériques aux clients qui se trouvent à proximité des magasins cibles.

Suivi des actifs

Créez des fonctionnalités de suivi des actifs pour aider les entreprises à comprendre l'emplacement actuel et historique de leurs produits, de leur personnel et de leur infrastructure. Grâce aux fonctionnalités de suivi des actifs, vous pouvez créer un certain nombre de solutions qui optimisent le personnel à distance, sécurisent les expéditions en cours de route et maximisent l'efficacité des expéditions.

Livraison

Intégrez des fonctionnalités de localisation dans les applications de livraison pour stocker, suivre et coordonner le lieu de départ, les véhicules de livraison et leur destination. Par exemple, une application de livraison de nourriture intégrant les fonctionnalités Amazon Location dispose de fonctionnalités de géolocalisation et de géofencing qui peuvent automatiquement avertir un restaurant lorsqu'un chauffeur-livreur se trouve à proximité. Cela réduit le temps d'attente et contribue à maintenir la qualité des aliments livrés.

Cette rubrique présente l'architecture et les étapes relatives aux applications que vous pouvez créer avec Amazon Location.

Rubriques

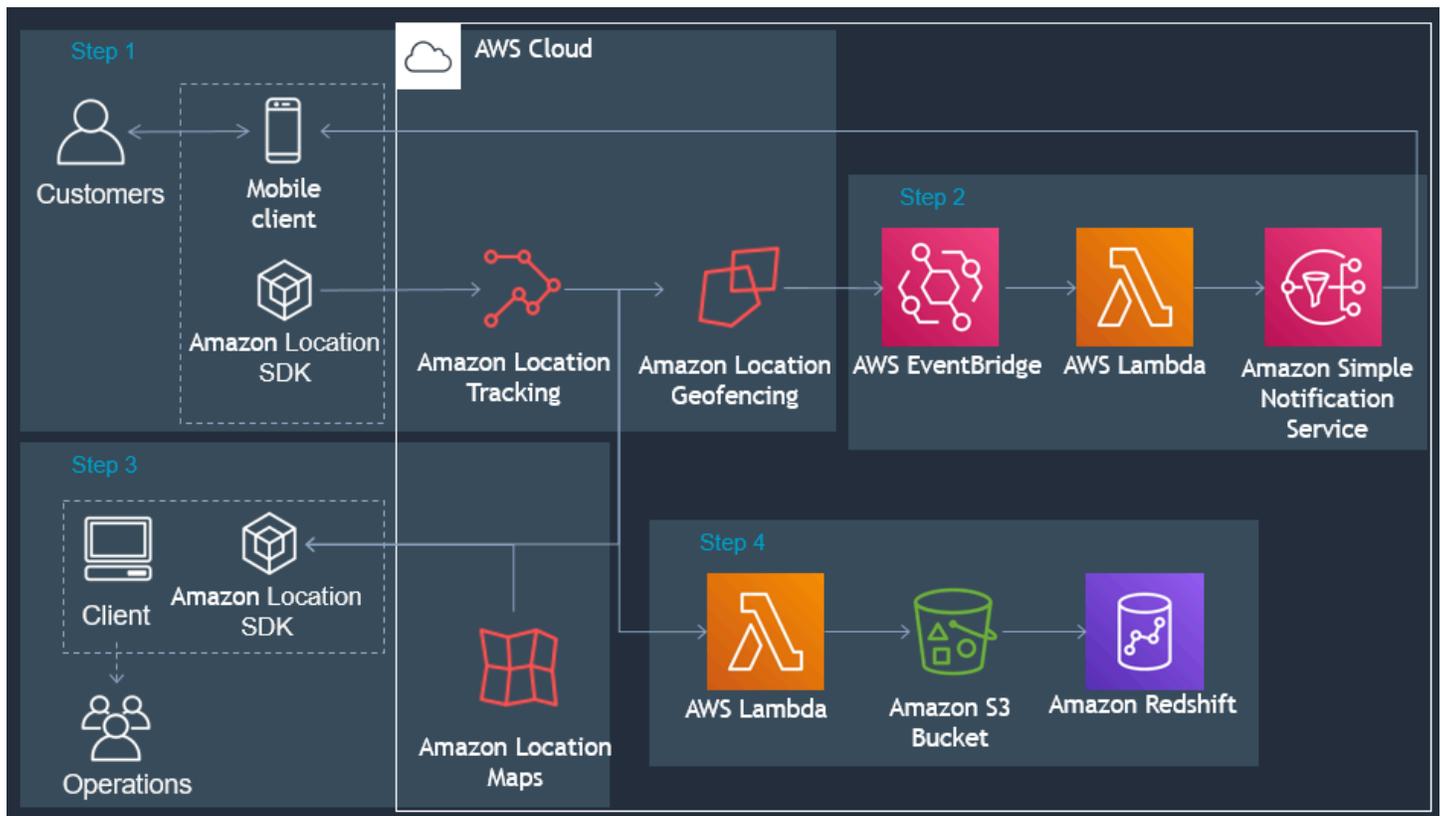
- [Applications d'engagement des utilisateurs et de géomarketing](#)
- [Applications de suivi des actifs](#)
- [Applications de livraison](#)

Applications d'engagement des utilisateurs et de géomarketing

Voici une illustration d'une architecture d'application d'engagement utilisateur et de géomarketing utilisant Amazon Location :

Grâce à cette architecture, vous pouvez :

- Lancez des événements en fonction de la proximité d'une cible afin de pouvoir envoyer des offres aux clients proches ou d'impliquer ceux qui ont récemment quitté votre établissement (ciblage géographique).
- Visualisez l'emplacement des appareils des clients sur une carte afin de suivre les tendances au fil du temps.
- Enregistrez les emplacements des appareils des clients que vous pouvez analyser au fil du temps.
- Analysez l'historique des positions pour identifier les tendances et les opportunités d'optimisation.



Voici un aperçu des étapes nécessaires à la création d'une application de géomarketing et d'engagement des utilisateurs :

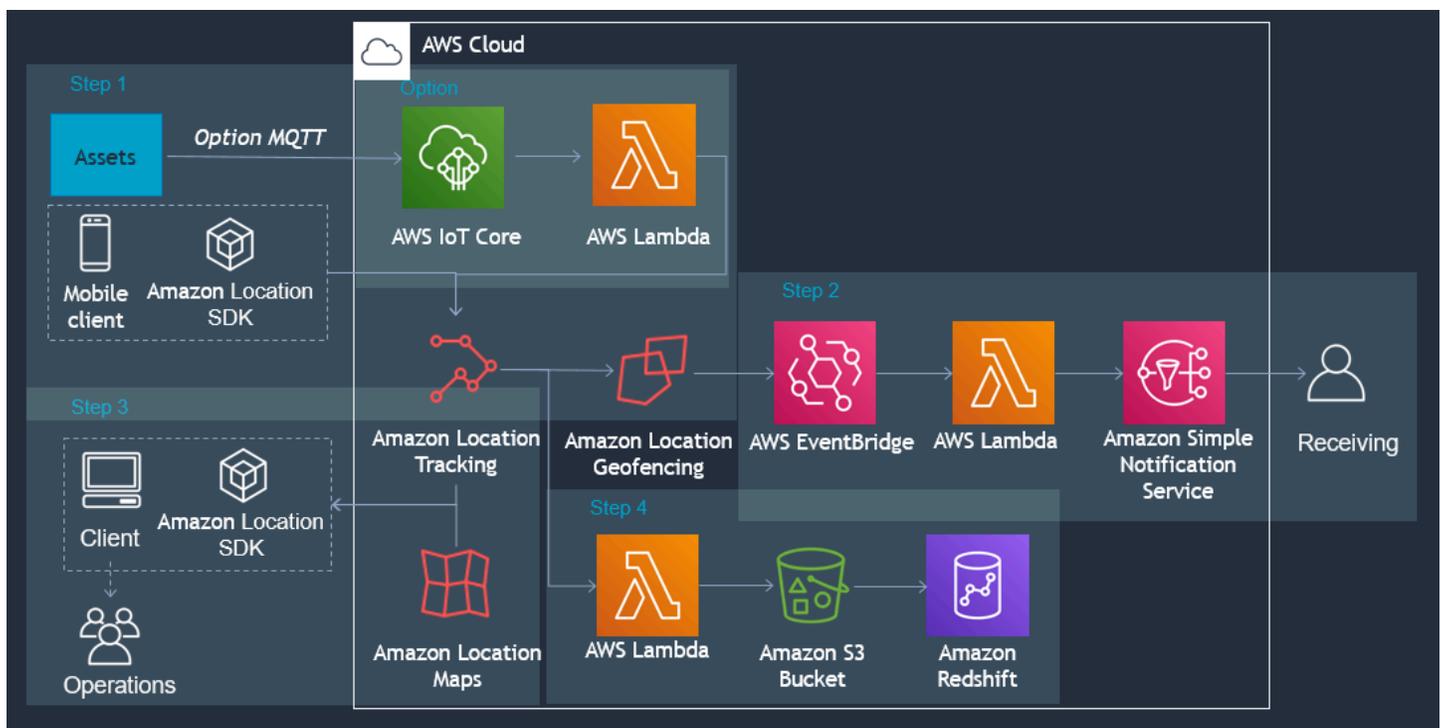
1. Créez vos géofences dans les collections Geofence et associez des traceurs à celles-ci. Pour plus d'informations, consultez [the section called "Géofencing et suivi"](#).
2. Configurez Amazon EventBridge pour envoyer une notification aux clients qui entrent ou sortent d'une zone d'intérêt géo-clôturée. Pour plus d'informations, consultez [the section called "Réagir aux événements avec EventBridge"](#).
3. Affichez les emplacements des clients et les géofences sur une carte. Pour plus d'informations, consultez la section [Utilisation de cartes](#).
4. Enregistrez les données de localisation dans un stockage à long terme pour une analyse plus approfondie.
5. Une fois que vous avez créé votre application, vous pouvez utiliser Amazon CloudWatch AWS CloudTrail pour la gérer. Pour plus d'informations, consultez [the section called "Surveillance avec CloudWatch"](#) et [the section called "Utilisation CloudTrail avec Amazon Location"](#).

Applications de suivi des actifs

Voici une illustration de l'architecture d'une application de suivi des actifs utilisant Amazon Location :

Grâce à cette architecture, vous pouvez :

- Affichez l'emplacement des actifs sur une carte pour avoir une vue d'ensemble. Par exemple, afficher une carte thermique utilisant des lieux ou des événements historiques pour aider une équipe chargée des opérations ou de la planification.
- Initiez des événements en fonction de la proximité des actifs pour informer le service destinataire afin de préparer l'arrivée d'un envoi et de réduire le temps de traitement.
- Stockez l'emplacement des actifs pour lancer des actions dans vos applications principales ou pour analyser les données au fil du temps.
- Analysez l'historique des positions pour identifier les tendances et les opportunités d'optimisation.



Voici un aperçu des étapes nécessaires à la création d'une application de suivi des actifs :

1. Créez vos géofences dans les collections Geofence et associez des traceurs à celles-ci. Pour plus d'informations, consultez [the section called "Géofencing et suivi"](#).

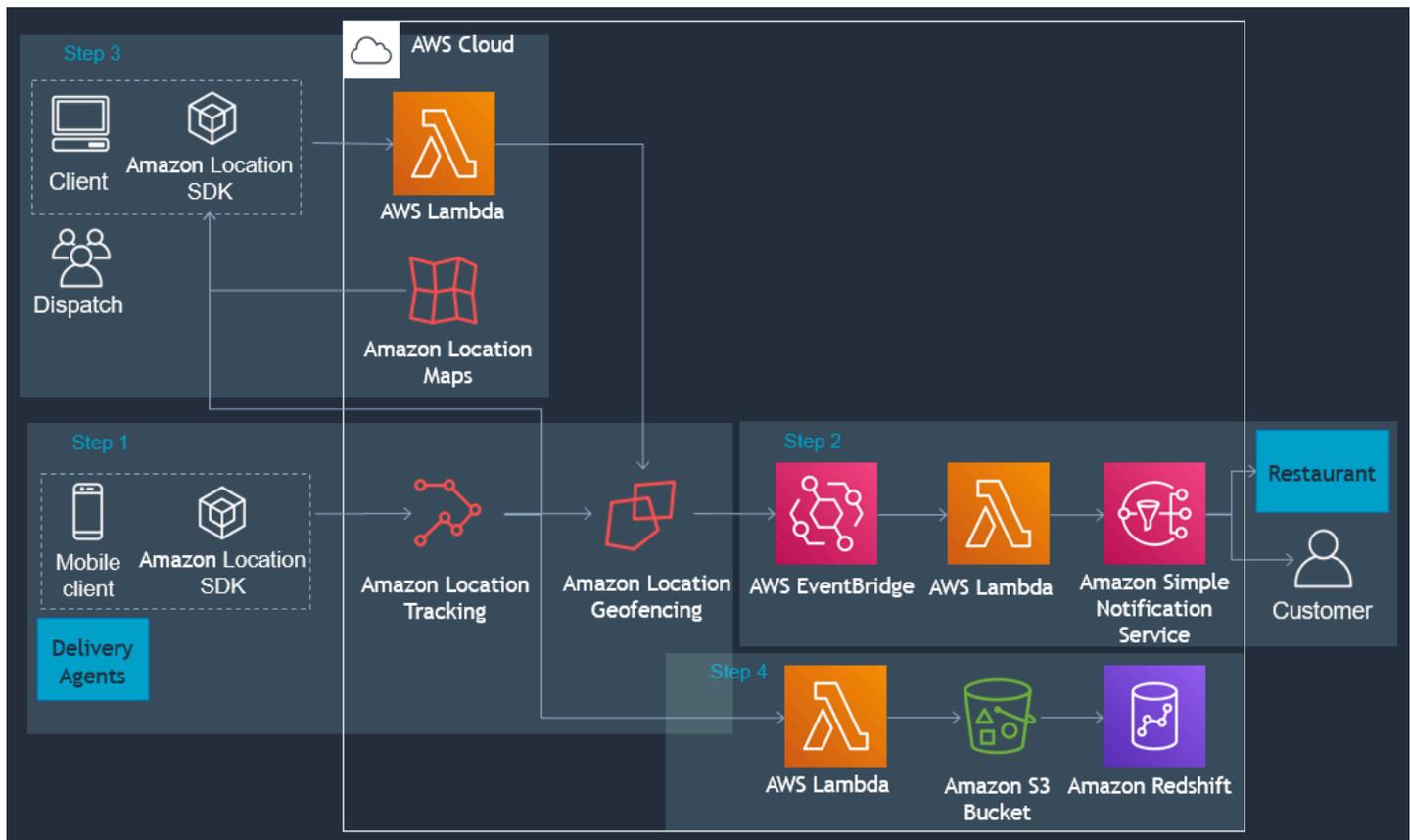
2. Configurez Amazon EventBridge pour envoyer une notification ou lancer un processus. Pour plus d'informations, consultez [the section called “Réagir aux événements avec EventBridge”](#).
3. Affichez vos actifs suivis et vos géofences actives sur une carte. Pour plus d'informations, consultez la section [Utilisation de cartes](#).
4. Enregistrez les données de localisation dans un stockage à long terme pour une analyse plus approfondie.
5. Une fois que vous avez créé votre application, vous pouvez utiliser Amazon CloudWatch AWS CloudTrail pour la gérer. Pour plus d'informations, consultez [the section called “Surveillance avec CloudWatch”](#) et [the section called “Utilisation CloudTrail avec Amazon Location”](#).

Applications de livraison

Voici une illustration d'une architecture d'application de livraison utilisant Amazon Location.

Grâce à cette architecture, vous pouvez :

- Organisez des événements en fonction de la proximité des agents de livraison afin que les ramassages soient prêts à temps et que les clients puissent être avertis de l'arrivée de leur livraison.
- Affichez les emplacements des chauffeurs, ainsi que les lieux de prise en charge et de dépose en temps quasi réel sur une carte pour donner aux équipes d'expédition une vue d'ensemble.
- Enregistrez les emplacements des agents de livraison afin de pouvoir agir en conséquence dans votre application principale ou de les analyser au fil du temps.
- Analysez l'historique des positions pour identifier les tendances et les opportunités d'optimisation.



Vous trouverez ci-dessous un aperçu des étapes nécessaires à la création d'une application de livraison :

1. Créez vos collections de géofences et associez les appareils suivis à la collection. Pour plus d'informations, veuillez consulter [the section called "Géofencing et suivi"](#).
2. Créez une AWS Lambda fonction pour ajouter et supprimer automatiquement des barrières géographiques lorsque vos commandes sont enregistrées.
3. Configurez Amazon EventBridge pour envoyer des notifications ou lancer un processus. Pour plus d'informations, consultez [the section called "Réagir aux événements avec EventBridge"](#).
4. Affichez les actifs suivis et les géofences actives sur une carte. Pour plus d'informations, consultez la section [Utilisation de cartes](#).
5. Enregistrez les données de localisation dans un stockage à long terme pour une analyse plus approfondie.
6. Une fois que vous avez créé votre application, vous pouvez utiliser Amazon CloudWatch AWS CloudTrail pour la gérer. Pour plus d'informations, consultez [the section called "Surveillance avec CloudWatch"](#) et [the section called "Utilisation CloudTrail avec Amazon Location"](#).

Qu'est-ce qu'un fournisseur de données ?

Utilisez Amazon Location Service pour accéder aux ressources de géolocalisation de plusieurs fournisseurs de données via votre AWS compte sans avoir besoin de contrats ou d'intégrations tiers. Cela peut vous aider à vous concentrer sur le développement de votre application, sans avoir à gérer les comptes, les informations d'identification, les licences et la facturation tiers.

Les services Amazon Location suivants utilisent des fournisseurs de données.

- **Cartes** — Choisissez des styles provenant de différents fournisseurs de cartes lorsque vous [créez une ressource cartographique](#). Vous pouvez utiliser les ressources cartographiques pour créer une carte interactive afin de visualiser les données.
- **Lieux** : choisissez un fournisseur de données lorsque vous [créez une ressource d'index de lieux](#) pour prendre en charge les requêtes de géocodage, de géocodage inversé et de recherches.
- **Itinéraires** — Choisissez un fournisseur de données pour prendre en charge les requêtes relatives au calcul d'itinéraires dans différentes zones géographiques et applications lorsque vous [créez une ressource de calcul d'itinéraires](#). Avec le fournisseur de données de votre choix, Amazon Location Service vous permet de calculer des itinéraires en fonction des données du réseau up-to-date routier, des données de trafic en temps réel, des fermetures planifiées et des modèles de trafic historiques.

Chaque fournisseur collecte et organise ses données par différents moyens. Ils peuvent également avoir des compétences différentes selon les régions du monde. Cette section fournit des informations sur nos fournisseurs de données. Vous pouvez sélectionner n'importe quel fournisseur de données en fonction de vos préférences.

Assurez-vous de lire les conditions générales lorsque vous utilisez les fournisseurs de données Amazon Location Service. Pour plus d'informations, consultez les [conditions de service AWS](#). Consultez également la [the section called “Confidentialité des données”](#) section pour plus d'informations sur la manière dont Amazon Location protège votre vie privée.

Couverture et fonctionnalités des fournisseurs de données

Le tableau suivant présente la couverture et les fonctionnalités de haut niveau pour chaque fournisseur de données.

Fournisseur de données	Couverture géographique	Couverture des fonctionnalités	Région AWS
Esri	Globale	Cartes, lieux, itinéraires	Toutes les régions dans lesquelles Amazon Location est disponible.
Saisir	Asie du sud-est	Cartes, lieux, itinéraires	Asie-Pacifique (Singapour)ap-southeast-1 , uniquement.
ICI	Globale	Cartes, lieux, itinéraires	Toutes les régions dans lesquelles Amazon Location est disponible.
Données ouvertes	Globale	Mappages	Toutes les régions dans lesquelles Amazon Location est disponible.

Pour plus d'informations sur les fonctionnalités spécifiques de chaque fournisseur de données, consultez [Fonctionnalités par fournisseur de données](#).

Chaque fournisseur de données collecte et produit des données de différentes manières. Pour en savoir plus sur leurs zones de couverture, consultez les rubriques suivantes :

- [Couverture : Esri](#)
- [Couverture : Grab](#)
- [Couverture : ICI](#)
- [Couverture : données ouvertes](#)

Si vous rencontrez un problème avec les données et souhaitez signaler une erreur au fournisseur de données, consultez les rubriques suivantes :

- [Signalement d'une erreur à Esri](#)
- [Signalement d'erreurs pour les GrabMaps données](#)
- [Erreur signalée à HERE](#)
- [Signalement des erreurs et contribution aux données ouvertes](#)

Styles de carte

Chaque fournisseur de données fournit un ensemble de styles de carte pour afficher les données cartographiques qu'il fournit. Par exemple, un style peut inclure des images satellites ou être optimisé pour montrer les routes à des fins de navigation. Vous trouverez la liste et des exemples de styles pour chaque fournisseur dans les rubriques suivantes.

- [Styles de carte Esri](#)
- [Saisissez les styles de carte](#)
- [Styles de carte HERE](#)
- [Styles de carte Open Data](#)

Plus d'informations sur chaque fournisseur de données

Les liens suivants fournissent des informations supplémentaires sur chaque fournisseur de données.

- [Esri](#)
- [GrabMaps](#)
- [Technologies HERE](#)
- [Données ouvertes](#)

Esri

Amazon Location Service utilise les services de localisation d'Esri pour aider les AWS clients à utiliser des cartes, à géocoder et à calculer des itinéraires de manière efficace. Les services de localisation d'Esri sont conçus à partir de données de ready-to-use localisation fiables et de haute qualité, organisées par des équipes d'experts de cartographes, de géographes et de démographes.

Pour plus d'informations sur les fonctionnalités, consultez les fournisseurs de données [Esri](#) on Amazon Location Service.

Rubriques

- [Styles de carte Esri](#)
- [Couverture : Esri](#)
- [Conditions d'utilisation et attribution des données : Esri](#)
- [Signalement d'une erreur à Esri](#)

Styles de carte Esri

Amazon Location Service prend en charge les styles de carte Esri suivants lors de la [création d'une ressource cartographique](#).

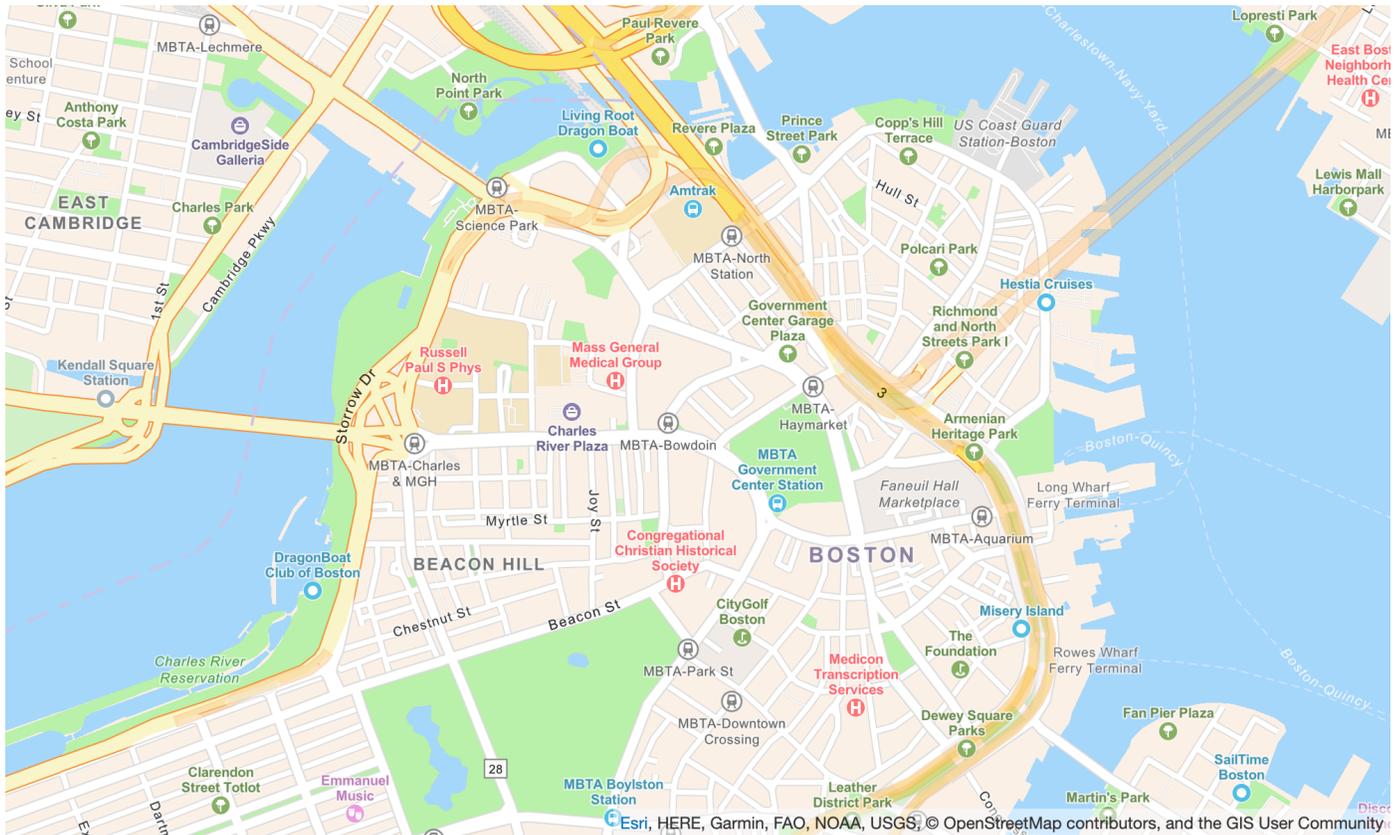
Note

Les styles de carte Esri qui ne sont pas répertoriés dans cette section ne sont pas pris en charge.

Les styles vectoriels Esri sont compatibles avec les styles alternatifs [Opinions politiques](#).

Esri Navigation

Navigation Esri



Nom du style de carte : VectorEsriNavigation

Cette carte fournit un fond de carte détaillé du monde symbolisé par un style de carte de navigation personnalisé conçu pour être utilisé pendant la journée sur des appareils mobiles.

Ce plan des rues complet comprend les autoroutes, les routes principales, les routes secondaires, les voies ferrées, les plans d'eau, les villes, les parcs, les monuments, les empreintes des bâtiments et les limites administratives. La couche de tuiles vectorielles de cette carte est créée à partir des mêmes sources de données que celles utilisées pour le World Street Map et les autres fonds de carte Esri. Activez la POI couche en la configurant [CustomLayers](#) pour tirer parti des données de lieux supplémentaires.

Pour plus d'informations, consultez [Esri World Navigation](#) sur le site Web d'Esri.

Note

La POI couche est activée sur la VectorEsriNavigation carte illustrée ci-dessus.

Fonts

Amazon Location diffuse des polices à l'aide de [GetMapGlyphs](#). Les piles de polices disponibles pour cette carte sont les suivantes :

- Arial en italique
- Arial Régulier
- Arial Bold
- Arial Unicode MS Bold
- Arial Unicode MS Régulier

Esri Imagery

Imagerie Esri

Nom du style de carte : `RasterEsriImagery`

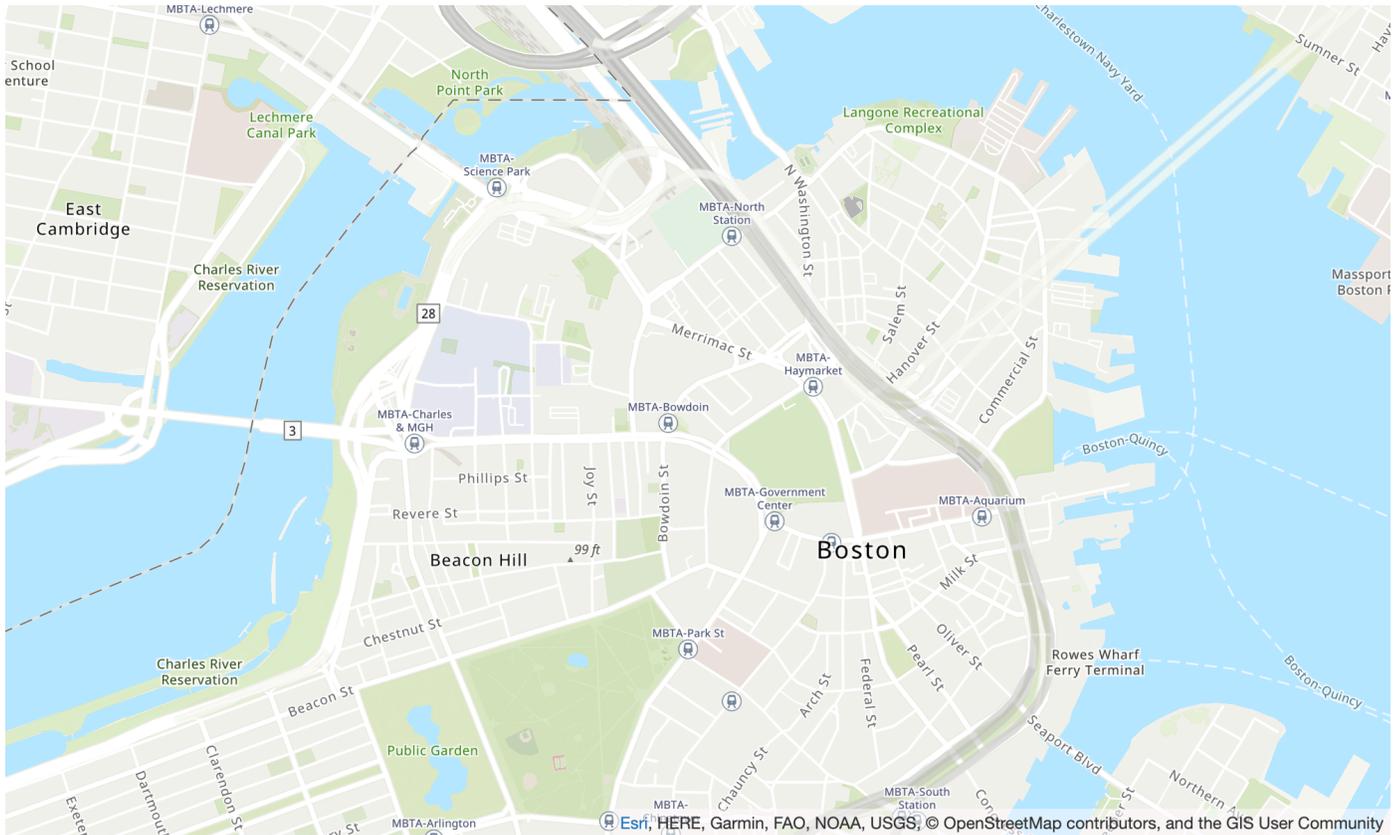
Cette carte fournit des images satellites et aériennes d'un mètre ou plus dans de nombreuses régions du monde, ainsi que des images satellites à faible résolution dans le monde entier.

La carte inclut 15 millions d'images à petite et moyenne échelle (environ 1:591 M à environ 1:72 k) et des images SPOT de 2,5 m (environ 1:288 k à environ 1:72 k) pour le monde entier. La carte présente des images d'une résolution de 0,5 m prises par Maxar sur la zone continentale des États-Unis et dans certaines parties de l'Europe occidentale. Cette carte présente des images submétriques supplémentaires de Maxar dans de nombreuses régions du monde. Dans d'autres parties du monde, la communauté des utilisateurs du SIG a fourni des images à différentes résolutions. Dans certaines communautés, des images à très haute résolution (jusqu'à 0,03 m) sont disponibles à une échelle d'environ 1:280.

Pour plus d'informations, consultez [Esri World Imagery](#) sur le site Web d'Esri.

Esri Light

Lampe Esri



Nom du style de carte : `VectorEsriTopographic`

Cela fournit un fond de carte détaillé pour le monde symbolisé par un style de carte Esri classique. Cela inclut les autoroutes, les routes principales, les routes secondaires, les voies ferrées, les plans d'eau, les villes, les parcs, les monuments, les empreintes des bâtiments et les limites administratives.

Ce fond de carte est compilé à partir de diverses sources officielles provenant de plusieurs fournisseurs de données, notamment le Service géologique des États-Unis (USGS), l'Agence américaine de protection de l'environnement (EPA), le Service des parcs nationaux des États-Unis (NPS), l'Organisation des Nations Unies pour l'alimentation et l'agriculture (FAO), le ministère des Ressources naturelles du Canada (RNCAN), HERE et Esri. Les données relatives à certaines zones proviennent de OpenStreetMap contributeurs. De plus, les données sont fournies par la communauté des SIG.

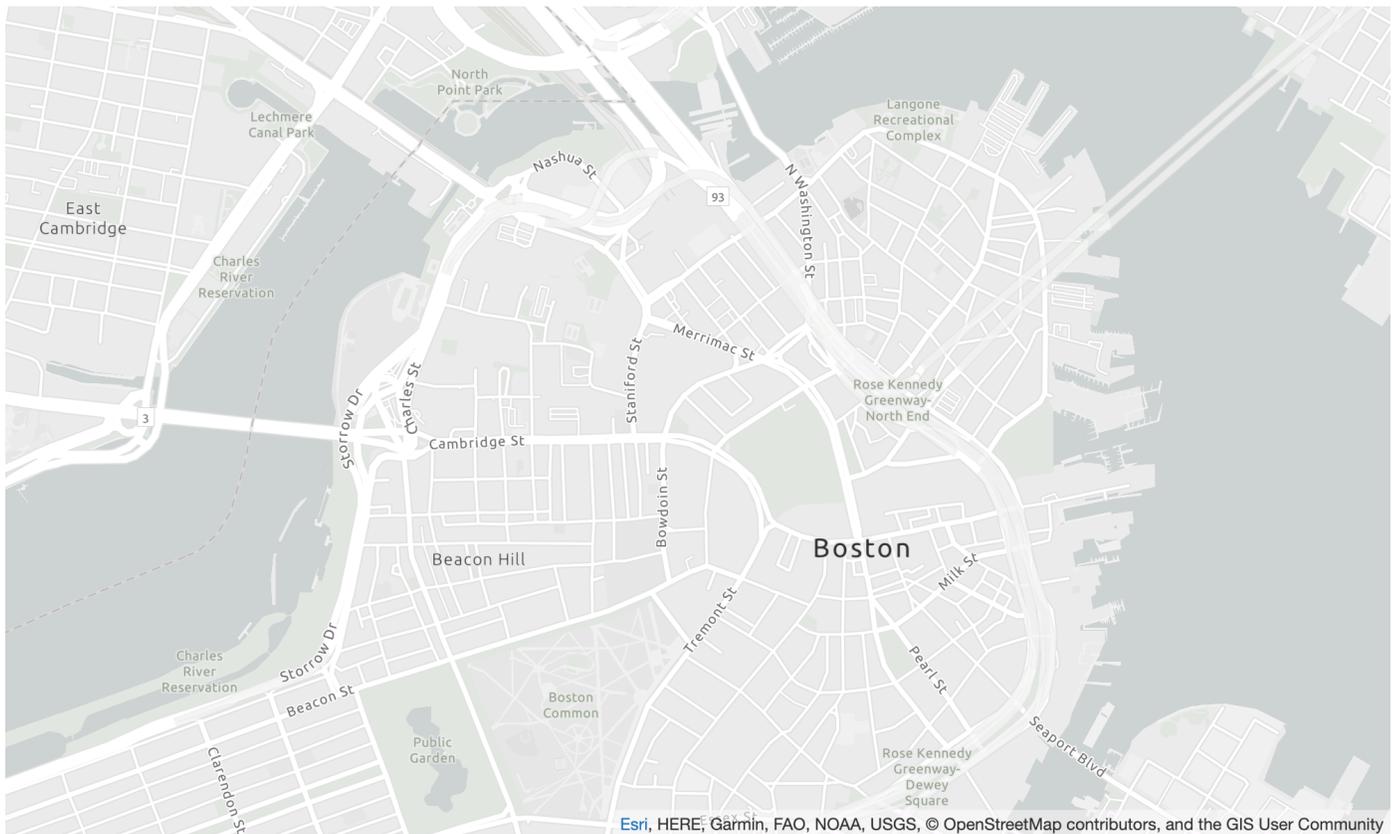
Fonts

Amazon Location diffuse des polices à l'aide de [GetMapGlyphs](#). Les piles de polices disponibles pour cette carte sont les suivantes :

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

Esri Light Gray Canvas

Toile gris clair Esri



Nom du style de carte : VectorEsriLightGrayCanvas

Cette carte fournit un fond de carte détaillé du monde symbolisé par un style de fond gris clair neutre avec un minimum de couleurs, d'étiquettes et de fonctionnalités, conçu pour attirer l'attention sur votre contenu thématique.

Cette couche de tuiles vectorielles est créée à partir des mêmes sources de données que celles utilisées pour le Light Gray Canvas et les autres fonds de carte Esri. La carte comprend les

autoroutes, les routes principales, les routes secondaires, les voies ferrées, les plans d'eau, les villes, les parcs, les monuments, les empreintes des bâtiments et les limites administratives.

Pour plus d'informations, consultez [Esri Light Gray Canvas](#) sur le site Web d'Esri.

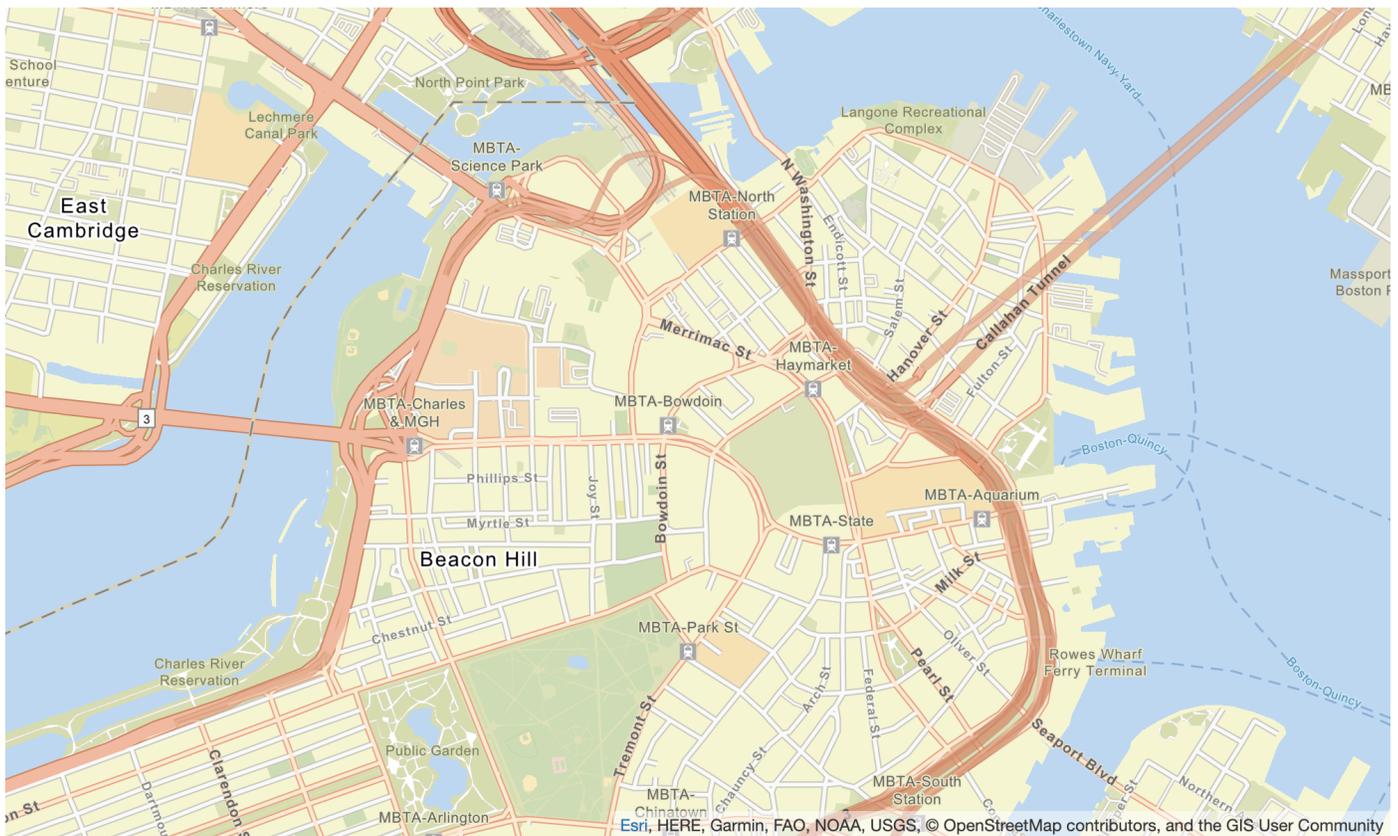
Fonts

Amazon Location diffuse des polices à l'aide de [GetMapGlyphs](#). Les piles de polices disponibles pour cette carte sont les suivantes :

- Ubuntu Italic
- Ubuntu Régulier
- Lampe Ubuntu
- Ubuntu Bold

Esri Street Map

Plan des rues Esri



Nom du style de carte : `VectorEsriStreets`

Cette carte fournit un fond de carte détaillé du monde symbolisé par un style de carte de navigation personnalisé conçu pour être utilisé pendant la journée sur des appareils mobiles.

Ce plan des rues complet comprend les autoroutes, les routes principales, les routes secondaires, les voies ferrées, les plans d'eau, les villes, les parcs, les monuments, les empreintes des bâtiments et les limites administratives. Il comprend également un ensemble plus riche de lieux, tels que des boutiques, des services, des restaurants, des attractions et d'autres points d'intérêt. La couche de tuiles vectorielles de cette carte est créée à partir des mêmes sources de données que celles utilisées pour le World Street Map et les autres fonds de carte Esri.

Pour plus d'informations, consultez [Esri World Street](#) sur le site Web d'Esri.

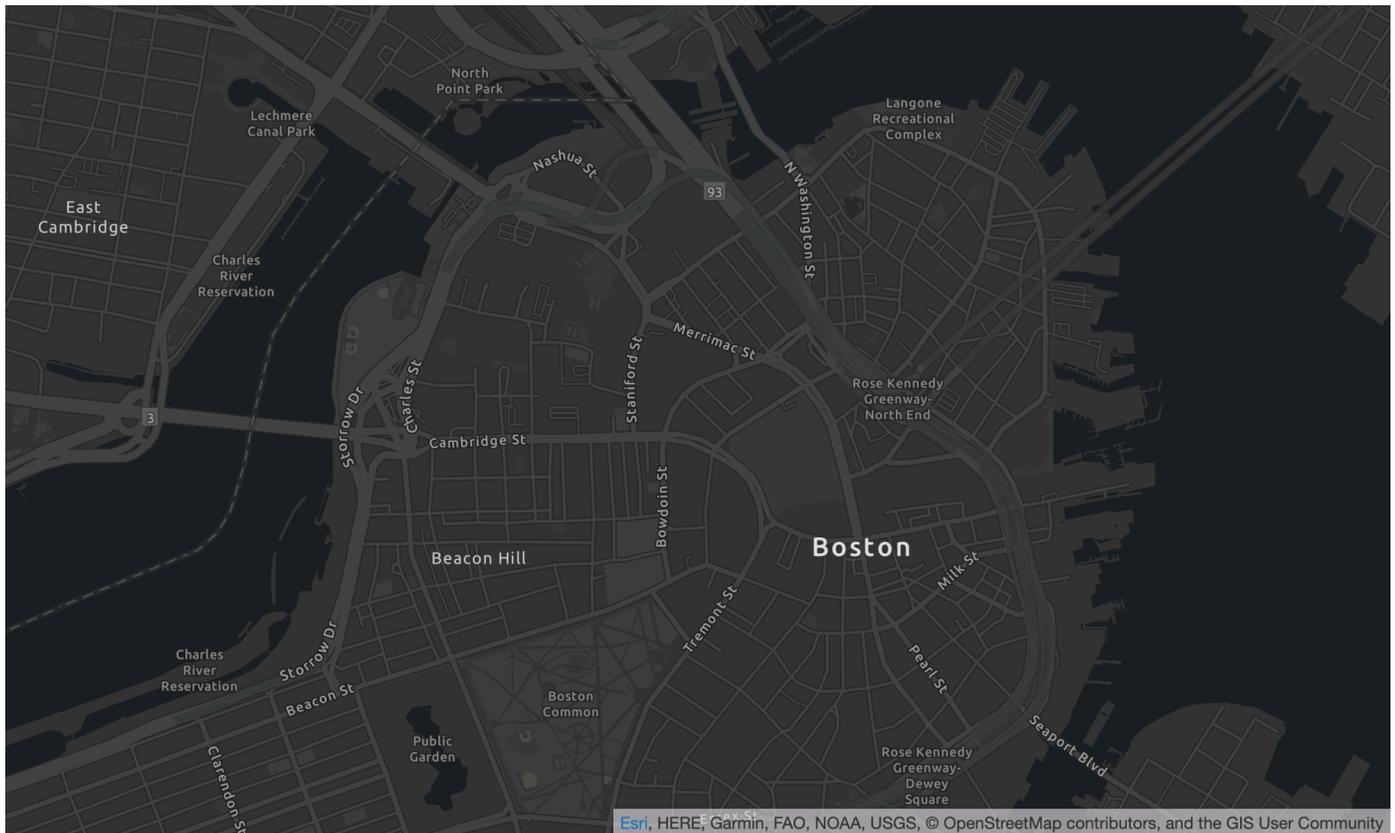
Fonts

Amazon Location diffuse des polices à l'aide de [GetMapGlyphs](#). Les piles de polices disponibles pour cette carte sont les suivantes :

- Arial en italique
- Arial Régulier
- Arial Bold
- Arial Unicode MS Bold
- Arial Unicode MS Régulier

Esri Dark Gray Canvas

Toile gris foncé Esri



Nom du style de carte : `VectorEsriDarkGrayCanvas`

Cette carte fournit un fond de carte vectoriel détaillé représentant le monde, symbolisé par un style de fond gris foncé neutre avec un minimum de couleurs, d'étiquettes et de fonctionnalités, conçu pour attirer l'attention sur votre contenu thématique.

Cette carte inclut les autoroutes, les routes principales, les routes secondaires, les voies ferrées, les plans d'eau, les villes, les parcs, les monuments, les empreintes des bâtiments et les limites administratives. Les couches de tuiles vectorielles de cette carte sont créées à partir des mêmes sources de données que celles utilisées pour la carte raster Dark Gray Canvas et les autres fonds de carte Esri.

Pour plus d'informations, consultez [Esri Dark Gray Canvas](#) sur le site Web d'Esri.

Fonts

Amazon Location diffuse des polices à l'aide de [GetMapGlyphs](#). Les piles de polices disponibles pour cette carte sont les suivantes :

- Ubuntu Medium Italic

- Ubuntu Medium
- Ubuntu Italic
- Ubuntu Régulier
- Ubuntu Bold

Couverture : Esri

Vous pouvez utiliser Esri en tant que fournisseur de données pour prendre en charge les requêtes de géocodage, de géocodage inversé et de recherches lorsque vous [créez une ressource d'index de lieux](#), ou pour prendre en charge les requêtes visant à calculer un itinéraire lorsque vous [créez une ressource de calculateur d'itinéraires](#).

Esri fournit différents niveaux de qualité des données dans différentes régions du monde. Pour plus d'informations sur la couverture dans la région qui vous intéresse, consultez :

- [Informations d'Esri sur la couverture du géocodage](#)
- [Informations d'Esri sur les réseaux routiers et la couverture du trafic](#)

Conditions d'utilisation et attribution des données : Esri

Avant d'utiliser les données d'Esri, assurez-vous de respecter toutes les exigences légales applicables, y compris les conditions de licence applicables à Esri et AWS.

Pour plus d'informations sur les AWS exigences, consultez les [conditions de service AWS](#).

Pour plus d'informations sur les directives d'attribution d'Esri, consultez les [attributions de données et les conditions d'utilisation d'Esri](#).

Signalement d'une erreur à Esri

Si vous rencontrez un problème avec les données et souhaitez signaler les erreurs et les incohérences à Esri, consultez l'article du support technique d'Esri intitulé [Comment : fournir des commentaires sur les fonds de carte et le géocodage](#).

GrabMaps

Grab est la plus grande organisation de livraison en Asie du Sud-Est, avec des millions de chauffeurs partenaires et de clients. Leur filiale crée [GrabMaps](#) des données up-to-date cartographiques dans

ces pays/régions pour leur propre usage et pour d'autres. Amazon Location Service utilise des GrabMaps « services de localisation » pour aider les AWS clients à utiliser des cartes, à géocoder et à calculer des itinéraires de manière efficace. GrabMaps« les services de localisation sont conçus pour fournir des données de ready-to-use localisation fiables et de haute qualité, en particulier pour les pays d'Asie du Sud-Est.

Pour plus d'informations sur les fonctionnalités supplémentaires, consultez [GrabMaps](#) les fournisseurs de données Amazon Location Service.

Important

Grab fournit des cartes uniquement pour l'Asie du Sud-Est et n'est disponible que dans la région Asie-Pacifique (Singapour) (ap-southeast-1). Pour plus d'informations, consultez [Pays/régions et zone couverts](#).

Rubriques

- [Saisissez les styles de carte](#)
- [Couverture : Grab](#)
- [Pays/régions et zone couverts](#)
- [Conditions d'utilisation et attribution des données : Grab](#)
- [Signalement d'erreurs pour les GrabMaps données](#)

Saisissez les styles de carte

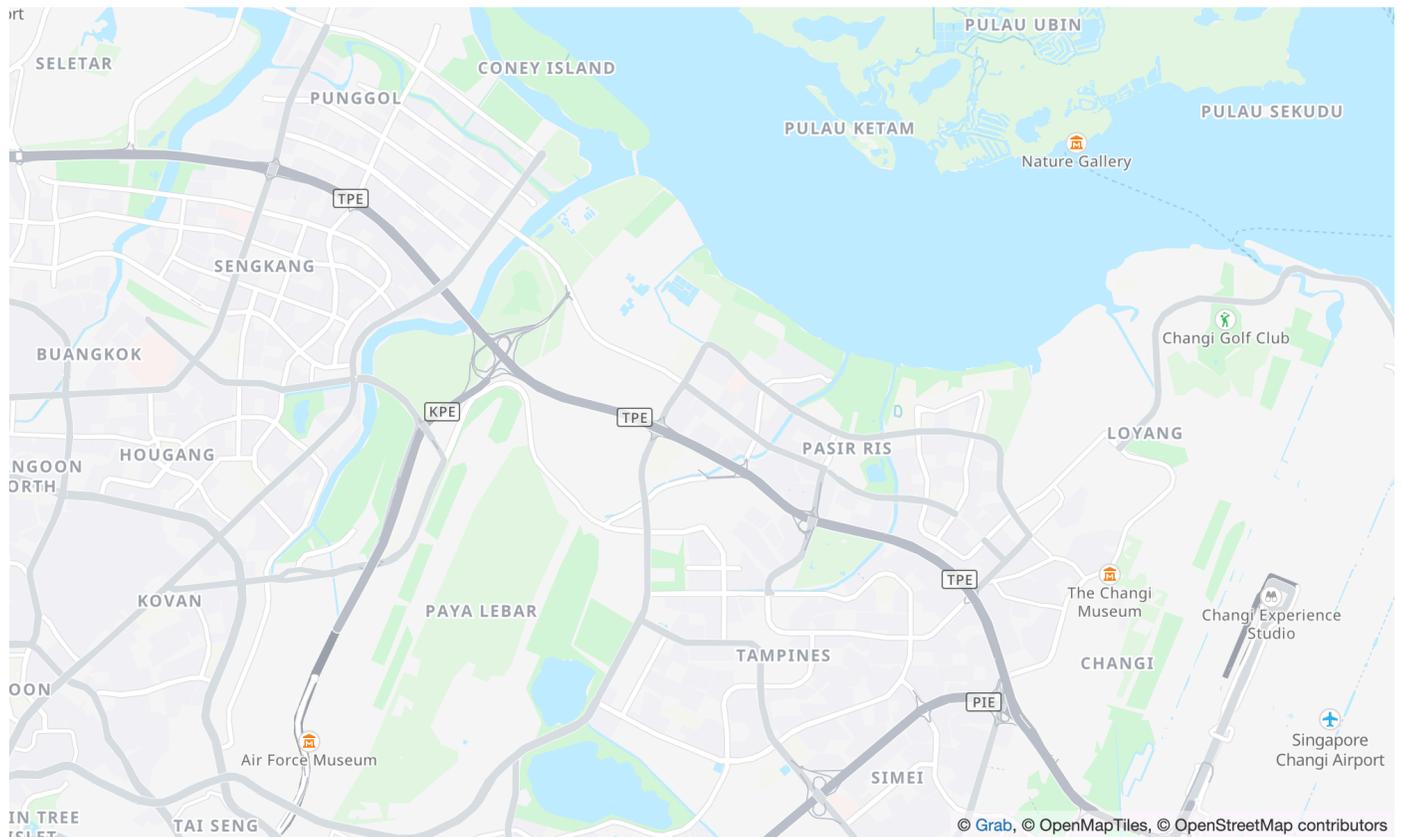
Amazon Location Service prend en charge les styles de carte Grab suivants lors de la [création d'une ressource cartographique](#) :

Note

Les styles de carte de capture qui ne sont pas répertoriés dans cette section ne sont actuellement pas pris en charge.

Grab Standard Light Map

Prenez une carte d'éclairage standard



Nom du style de carte : `VectorGrabStandardLight`

Le fond de carte standard de Grab avec des couleurs détaillées sur l'utilisation du sol, les noms des zones, les routes, les points de repère et les points d'intérêt couvrant l'Asie du Sud-Est.

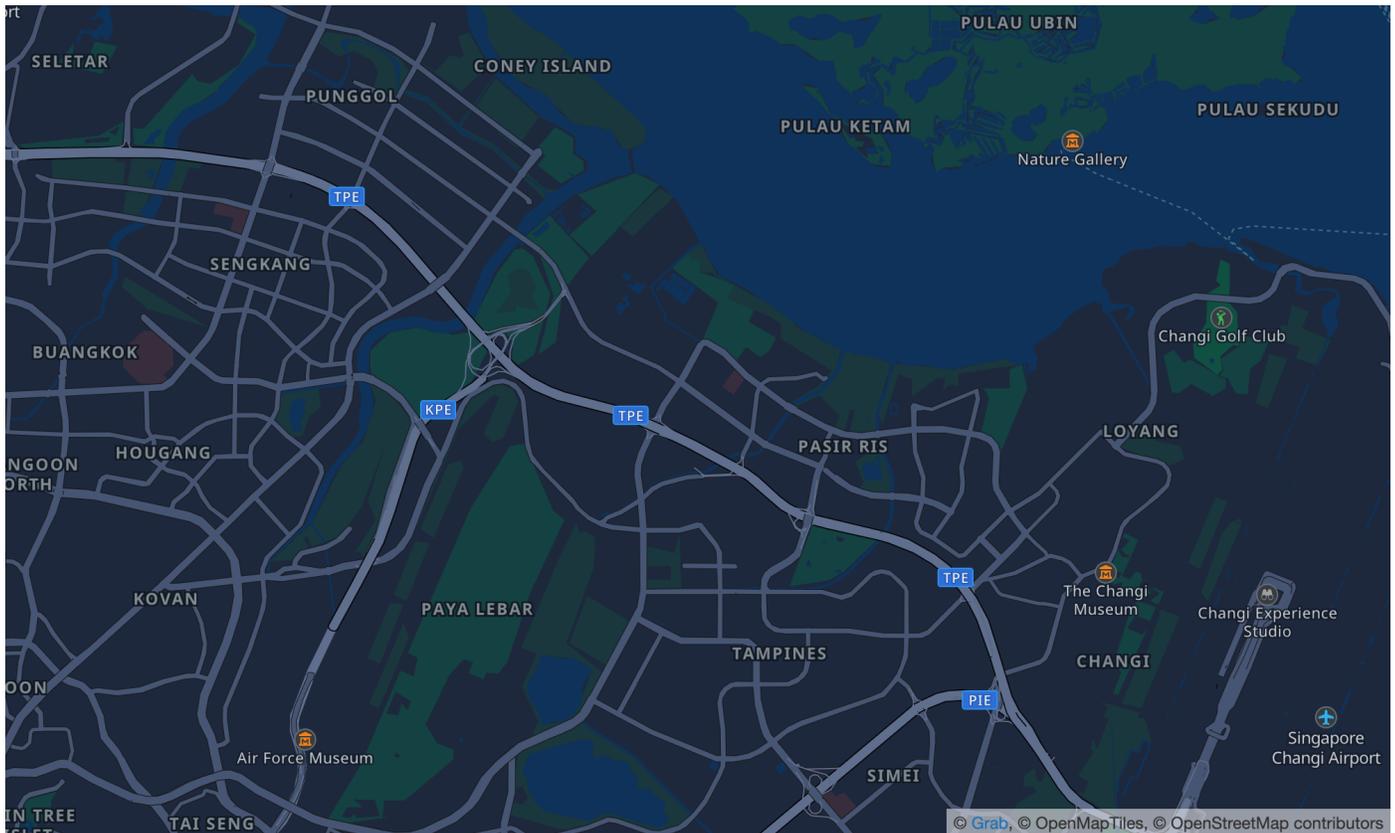
Fonts

Amazon Location diffuse des polices à l'aide de [GetMapGlyphs](#). Les piles de polices disponibles pour cette carte sont les suivantes :

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

Grab Standard Dark Map

Prenez une carte noire standard



Nom du style de carte : `VectorGrabStandardDark`

La variante foncée de Grab de leur carte de base standard, avec des couleurs détaillées sur l'utilisation du sol, les noms des zones, les routes, les points de repère et les points d'intérêt couvrant l'Asie du Sud-Est.

Fonts

Amazon Location diffuse des polices à l'aide de [GetMapGlyphs](#). Les piles de polices disponibles pour cette carte sont les suivantes :

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

Couverture : Grab

Vous pouvez utiliser Grab en tant que fournisseur de données pour prendre en charge les requêtes de géocodage, de géocodage inversé et de recherches lorsque vous [créez une ressource d'index de](#)

[lieux](#), ou pour prendre en charge les requêtes visant à calculer un itinéraire lorsque vous [créez une ressource de calculateur d'itinéraires](#).

Pays/régions et zone couverts

Grab fournit des cartes uniquement pour l'Asie du Sud-Est et n'est disponible que dans la région Asie-Pacifique (Singapour) (ap-southeast-1).

Grab fournit des données détaillées pour les pays/régions suivants :

- Malaisie
- Philippines
- Thaïlande
- Singapour
- Vietnam
- Indonésie
- Birmanie
- Cambodge

Note

En dehors de ces zones, les ressources Amazon Location Service créées avec Grab en tant que fournisseur de données ne fourniront aucun résultat. Cela inclut les résultats de recherche ou les itinéraires.

Les cartes de Grab se situent dans les limites suivantes :

- Sud — Latitude -21,943045533438166
- Ouest — Longitude 90,0
- Nord — Latitude 31,952162238024968
- Est — 146,25 de longitude

Pour les niveaux de zoom 1 à 4, Grab inclut une couverture mondiale. Pour les niveaux de zoom inférieurs ou égaux à 5, les vignettes cartographiques sont fournies uniquement dans ce cadre délimité.

Note

En dehors de cette zone délimitée, les ressources cartographiques Amazon Location Service créées avec Grab en tant que fournisseur de données ne renverront pas de vignettes cartographiques. Pour éviter de voir apparaître des erreurs 404 dans votre application, vous pouvez limiter la carte à l'aide d'un cadre, comme décrit dans [Définition de l'étendue d'une carte à l'aide de MapLibre](#).

Modes de voyage Grab Routing

Pour l'itinéraire, Grab propose des itinéraires en voiture et en moto pour tous les pays/régions listés précédemment.

Grab ne prend pas en charge l'acheminement des camions.

Pour les itinéraires cyclables et pédestres, Grab soutient les villes suivantes :

- Singapour
- Jakarta
- Manille
- Vallée de Klang
- Bangkok
- Hô-Chi-Minh-Ville
- Hanoi

Conditions d'utilisation et attribution des données : Grab

Lorsque vous utilisez les données de Grab, vous devez respecter toutes les exigences légales applicables, y compris les conditions de licence applicables à Grab et AWS.

Pour plus d'informations sur les AWS exigences, consultez les [conditions de service AWS](#).

Pour plus d'informations sur les « directives d'attribution GrabMaps », consultez la section 9.23 des Attributions de [données et conditions d'utilisation de](#) Grab.

Signalement d'erreurs pour les GrabMaps données

Si vous rencontrez un problème avec les données de GrabMaps, et si vous souhaitez signaler des erreurs ou des incohérences, [contactez le support AWS technique](#).

Technologies HERE

Amazon Location Service utilise les services de localisation de HERE Technologies pour aider les AWS clients à utiliser des cartes, à géocoder et à calculer des itinéraires de manière efficace. Les données de localisation de HERE constituent une plateforme centrée sur la localisation qui est ouverte, sécurisée et privée. En sélectionnant les données de localisation HERE, vous sélectionnez des données précises, actualisées et robustes qui sont déployées nativement sur le AWS Cloud.

Pour plus d'informations sur les fonctionnalités, consultez la [page ICI](#) sur les fournisseurs de données Amazon Location Service.

Rubriques

- [Styles de carte HERE](#)
- [Couverture : ICI](#)
- [Conditions d'utilisation et attribution des données : ICI](#)
- [Erreur signalée à HERE](#)

Styles de carte HERE

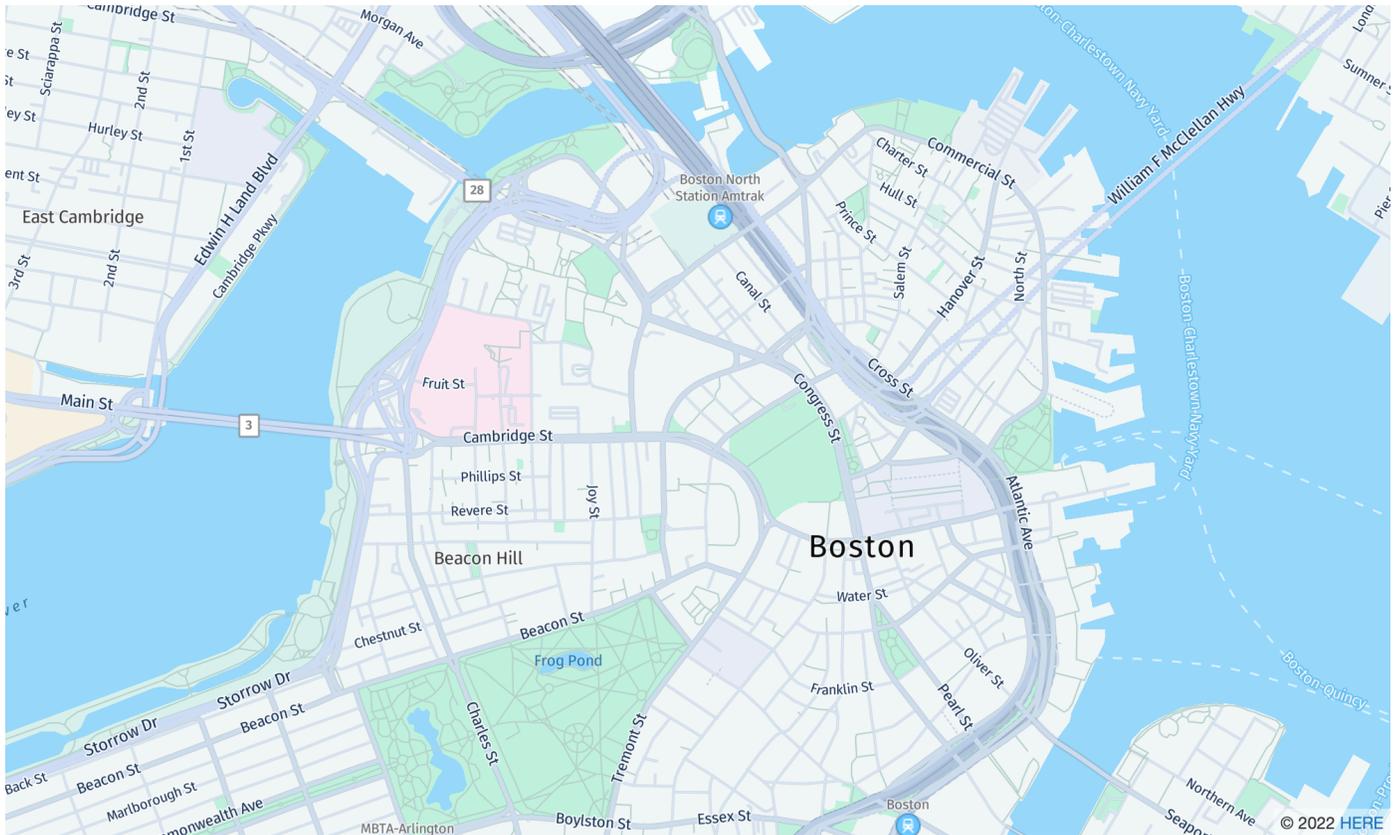
Amazon Location Service prend en charge les styles de carte HERE suivants lors de la [création d'une ressource cartographique](#) :

Note

Les styles de carte HERE qui ne sont pas répertoriés dans cette section ne sont actuellement pas pris en charge.

HERE Explore

ICI Découvrez



Nom du style de carte : VectorHereExplore

ICI Découvrez

Une carte de base neutre et détaillée du monde. Le plan des rues comprend les autoroutes, les routes principales, les routes secondaires, les voies ferrées, les plans d'eau, les villes, les parcs, les monuments, les empreintes des bâtiments et les limites administratives. Comprend une carte entièrement conçue du Japon.

Fonts

Amazon Location diffuse des polices à l'aide de [GetMapGlyphs](#). Les piles de polices disponibles pour cette carte sont les suivantes :

- Fira GO Italic
- Fira GO Régulier
- Fira GO Bold
- Lampe Noto Sans CJK JP
- Noto Sans CJK JP Régulier

- Photo Sans CJK JP Bold

HERE Imagery

Imagerie HERE



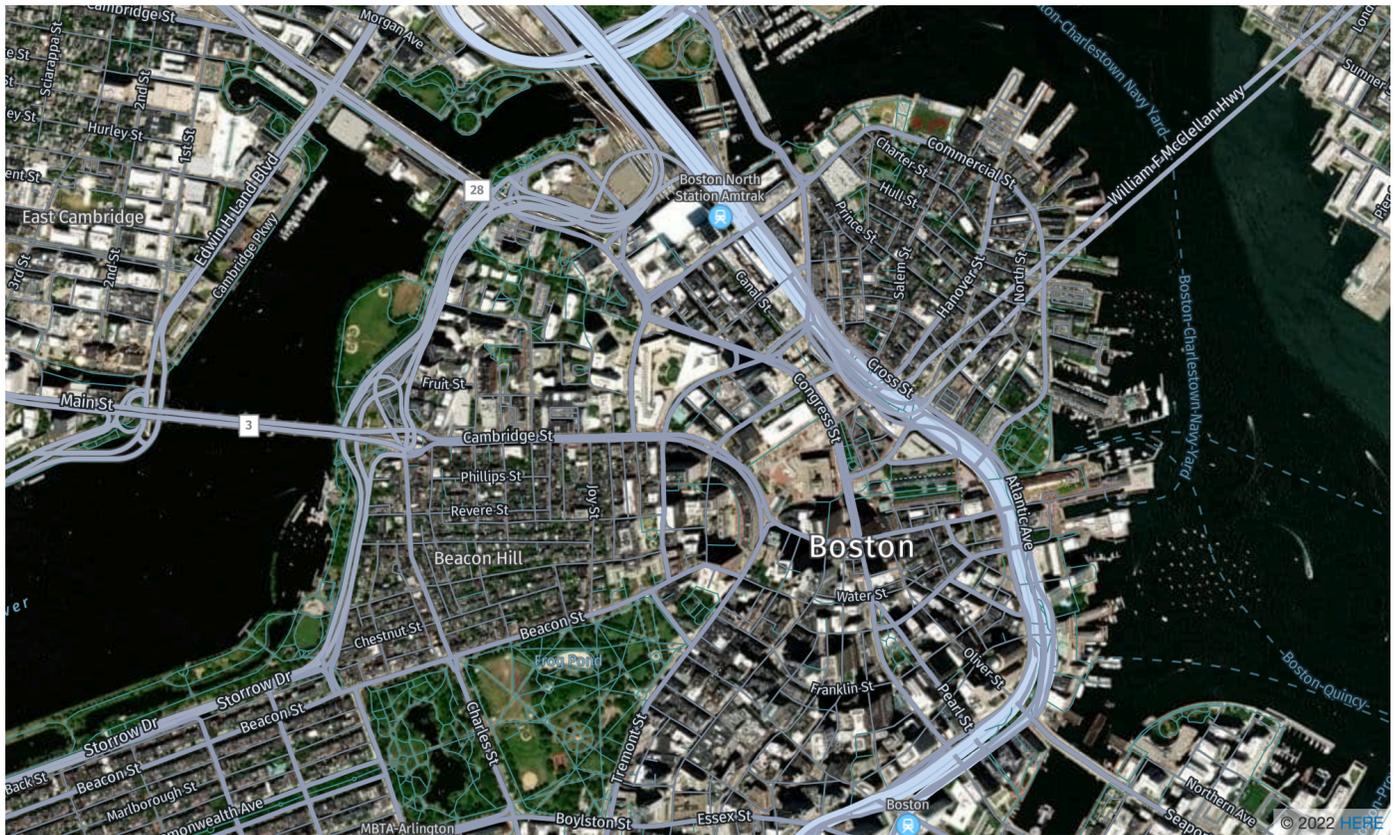
Nom du style de carte : RasterHereExploreSatellite

Imagerie HERE

HERE Imagery fournit des images satellites haute résolution avec une couverture mondiale.

HERE Hybrid

HERE Hybride



Nom du style de carte : HybridHereExploreSatellite

HERE Hybride

Le style HERE Hybrid affiche le réseau routier, les noms des rues et les étiquettes des villes sur des images satellites. Ce style superpose deux tuiles cartographiques : l'image satellite (tuile matricielle) en arrière-plan et le réseau routier et les étiquettes (tuile vectorielle) en haut. Ce style récupère automatiquement les tuiles matricielles et vectorielles nécessaires au rendu de la carte.

Note

Les styles hybrides utilisent à la fois des tuiles vectorielles et matricielles lors du rendu de la carte que vous voyez. Cela signifie que davantage de tuiles sont extraites qu'en utilisant uniquement des tuiles vectorielles ou matricielles. Vos frais incluront toutes les tuiles récupérées.

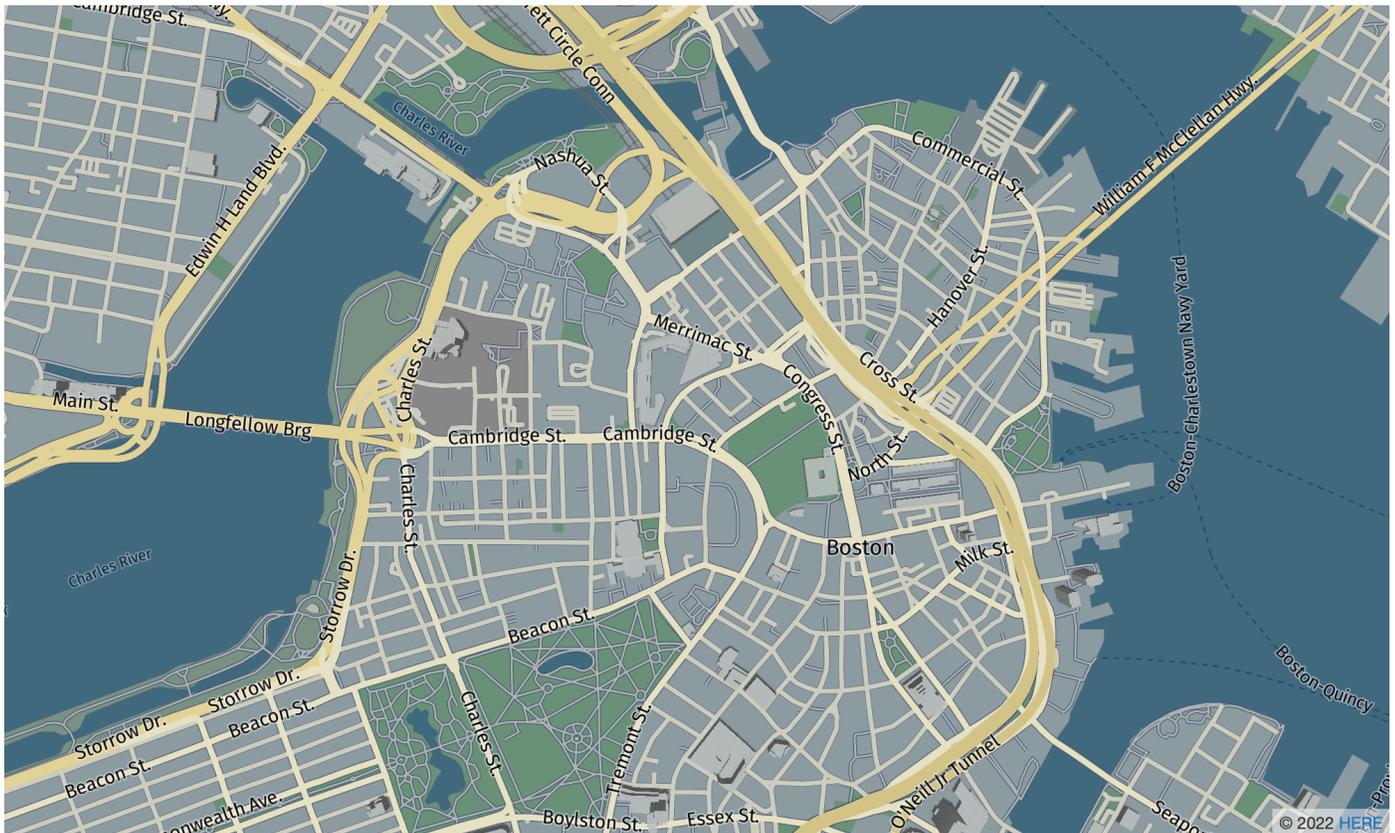
Fonts

Amazon Location diffuse des polices à l'aide de [GetMapGlyphs](#). Les piles de polices disponibles pour cette carte sont les suivantes :

- Fira GO Italic
- Fira GO Régulier
- Fira GO Bold
- Lampe Noto Sans CJK JP
- Noto Sans CJK JP Régulier
- Photo Sans CJK JP Bold

HERE Contrast (Berlin)

HERE Contrast (Berlin)



Nom du style de carte : VectorHereContrast

HERE Contrast (Berlin)

Une carte de base détaillée du monde qui combine les rendus 3D et 2D. Le plan des rues à contraste élevé inclut les autoroutes, les routes principales, les routes secondaires, les voies ferrées, les plans d'eau, les villes, les parcs, les monuments, les empreintes des bâtiments et les limites administratives.

Fonts

Amazon Location diffuse des polices à l'aide de [GetMapGlyphs](#). Les piles de polices disponibles pour cette carte sont les suivantes :

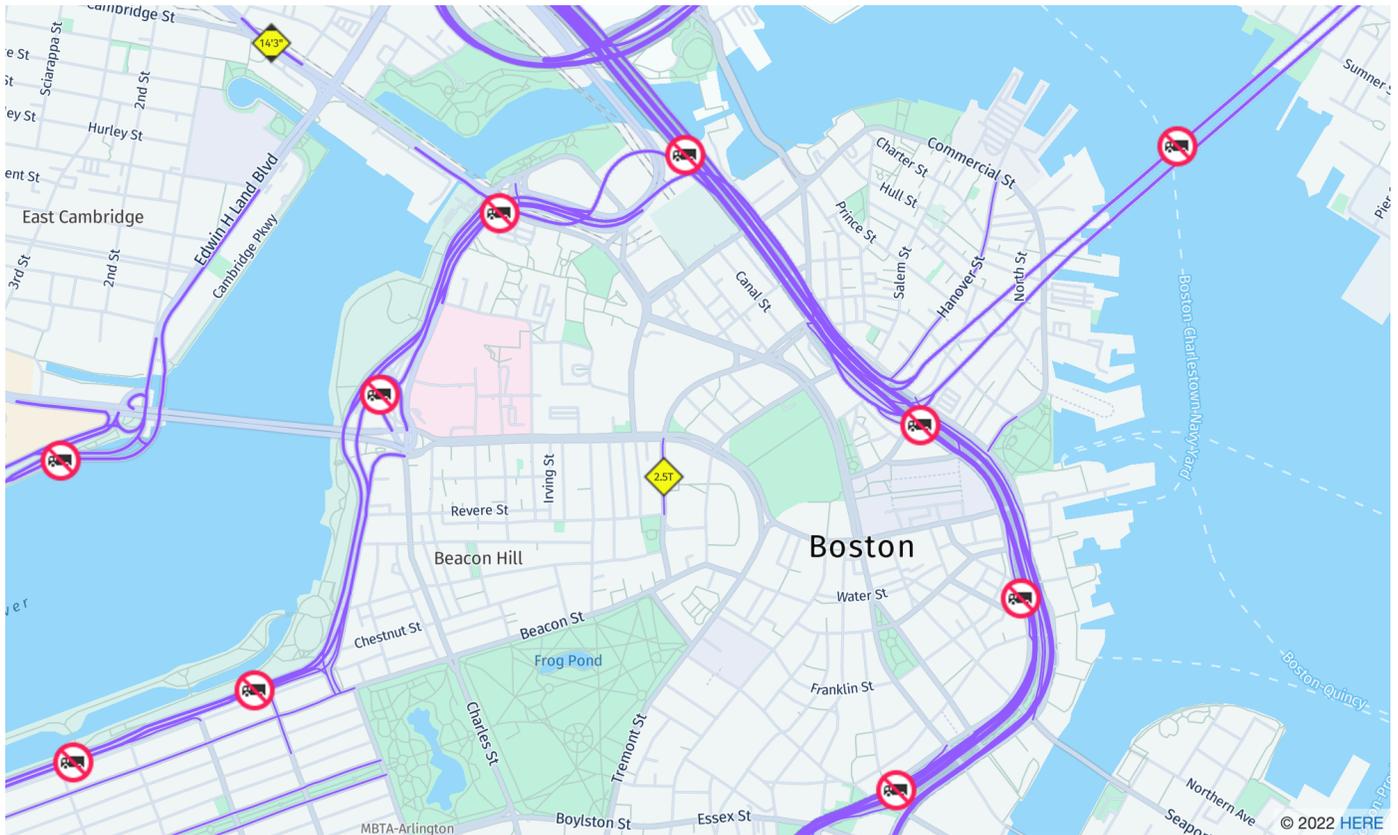
- Fira GO Régulier
- Fira GO Bold

Note

Ce style a été renommé à partir de `VectorHereBerlin` (HERE Berlin maps). `VectorHereBerlin` est obsolète, mais continuera de fonctionner dans les applications qui l'utilisent.

HERE Explore Truck

ICI Découvrez Truck



Nom du style de carte : `VectorHereExploreTruck`

ICI Découvrez Truck

Une carte de base neutre et détaillée du monde. Le plan des rues s'inspire du style HERE Explore et met en évidence les restrictions et les attributs des pistes (y compris la largeur, la hauteur et les matières dangereuses) à l'aide de symboles et d'icônes, afin de faciliter les cas d'utilisation dans les domaines du transport et de la logistique.

Fonts

Amazon Location diffuse des polices à l'aide de [GetMapGlyphs](#). Les piles de polices disponibles pour cette carte sont les suivantes :

- Fira GO Italic
- Fira GO Régulier
- Fira GO Bold
- Lampe Noto Sans CJK JP
- Noto Sans CJK JP Régulier

- Photo Sans CJK JP Bold

Pour plus d'informations sur la qualité des données cartographiques dans différentes régions du monde, voir [la couverture cartographique HERE](#).

Couverture : ICI

Vous pouvez utiliser HERE en tant que fournisseur de données pour prendre en charge les requêtes de géocodage, de géocodage inversé et de recherches lorsque vous [créez une ressource d'index de lieux](#), ou pour prendre en charge les requêtes visant à calculer un itinéraire lorsque vous [créez une ressource de calculateur d'itinéraires](#).

HERE fournit différents niveaux de qualité des données dans différentes régions du monde. Pour plus d'informations sur la couverture dans la région qui vous intéresse, consultez les pages suivantes :

- [Couverture de géocodage HERE](#)
- [Couverture des itinéraires automobiles HERE](#)
- [Couverture de l'itinéraire des camions HERE](#)

Conditions d'utilisation et attribution des données : ICI

Avant d'utiliser les données HERE, assurez-vous de respecter toutes les exigences légales applicables, y compris les termes de licence applicables à HERE et AWS. En raison des limites de licence, vous ne pouvez pas utiliser HERE pour stocker les résultats de géocodage pour des sites situés au Japon.

Pour plus d'informations sur les AWS exigences, consultez les [conditions de service AWS](#).

Pour plus d'informations sur les directives d'attribution de HERE, consultez la section 2 des [conditions des fournisseurs de HERE Technologies applicables à la localisation et aux autres contenus](#).

Erreur signalée à HERE

Pour signaler des erreurs et des incohérences cartographiques à HERE, rendez-vous sur Signaler une erreur <https://www.here.com/contact> cartographique et choisissez Signaler une erreur cartographique.

Données ouvertes

Amazon Location Service permet d'accéder aux données cartographiques open source via le fournisseur de données ouvertes. Open Data fournit des fonds de carte mondiaux créés à partir de la [distribution cartographique Daylight OpenStreetMap \(OSM\)](#), de [Natural Earth](#) et d'autres sources de données ouvertes. Les cartes fournies sont conçues pour prendre en charge différentes applications et différents cas d'utilisation, notamment la logistique et la livraison, ainsi que la visualisation des données dans les environnements Web et mobiles. Avec plus d'un million de cartographes, la communauté OSM met à jour des centaines de milliers de fonctionnalités par jour. Amazon Location Service intègre régulièrement ces modifications.

Pour plus d'informations sur les fonctionnalités, consultez [Open Data](#) sur les fournisseurs de données Amazon Location Service.

Rubriques

- [Styles de carte Open Data](#)
- [Couverture : données ouvertes](#)
- [Conditions d'utilisation et attribution des données : Open Data](#)
- [Signalement des erreurs et contribution aux données ouvertes](#)

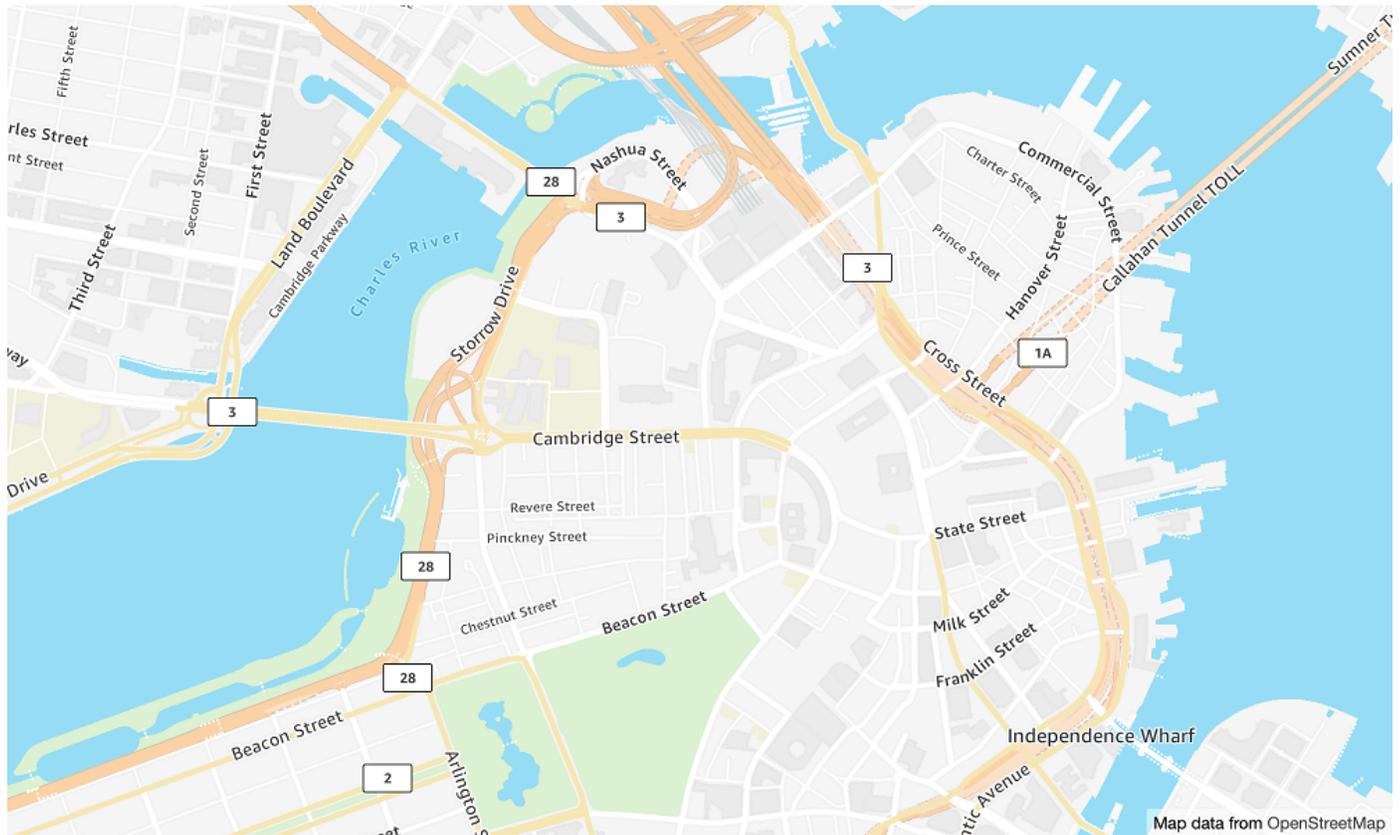
Styles de carte Open Data

Amazon Location Service prend en charge les styles de carte suivants lors de la [création d'une ressource cartographique](#) :

Les styles de carte Open Data sont compatibles avec les styles alternatifs [Opinions politiques](#).

Open Data Standard Light

Lumière standard pour les données ouvertes



Nom du style de carte : `VectorOpenDataStandardLight`

Cela fournit un fond de carte détaillé du monde dans un style de carte clair, adapté à l'utilisation de sites Web et d'applications mobiles. Cela inclut les autoroutes, les routes principales, les routes secondaires, les voies ferrées, les plans d'eau, les villes, les parcs, les monuments, les empreintes des bâtiments et les limites administratives.

Ce fond de carte est basé sur la [distribution cartographique OSM Daylight](#) compilée par les contributeurs OpenStreetMap (OSM). La communauté OSM compte plus de 1,8 million de contributeurs qui mettent à jour plus de 500 000 fonctionnalités par jour. Amazon Location Service intègre régulièrement ces modifications.

Fonts

Amazon Location diffuse des polices à l'aide de [GetMapGlyphs](#). Les piles de polices disponibles pour cette carte sont les suivantes :

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

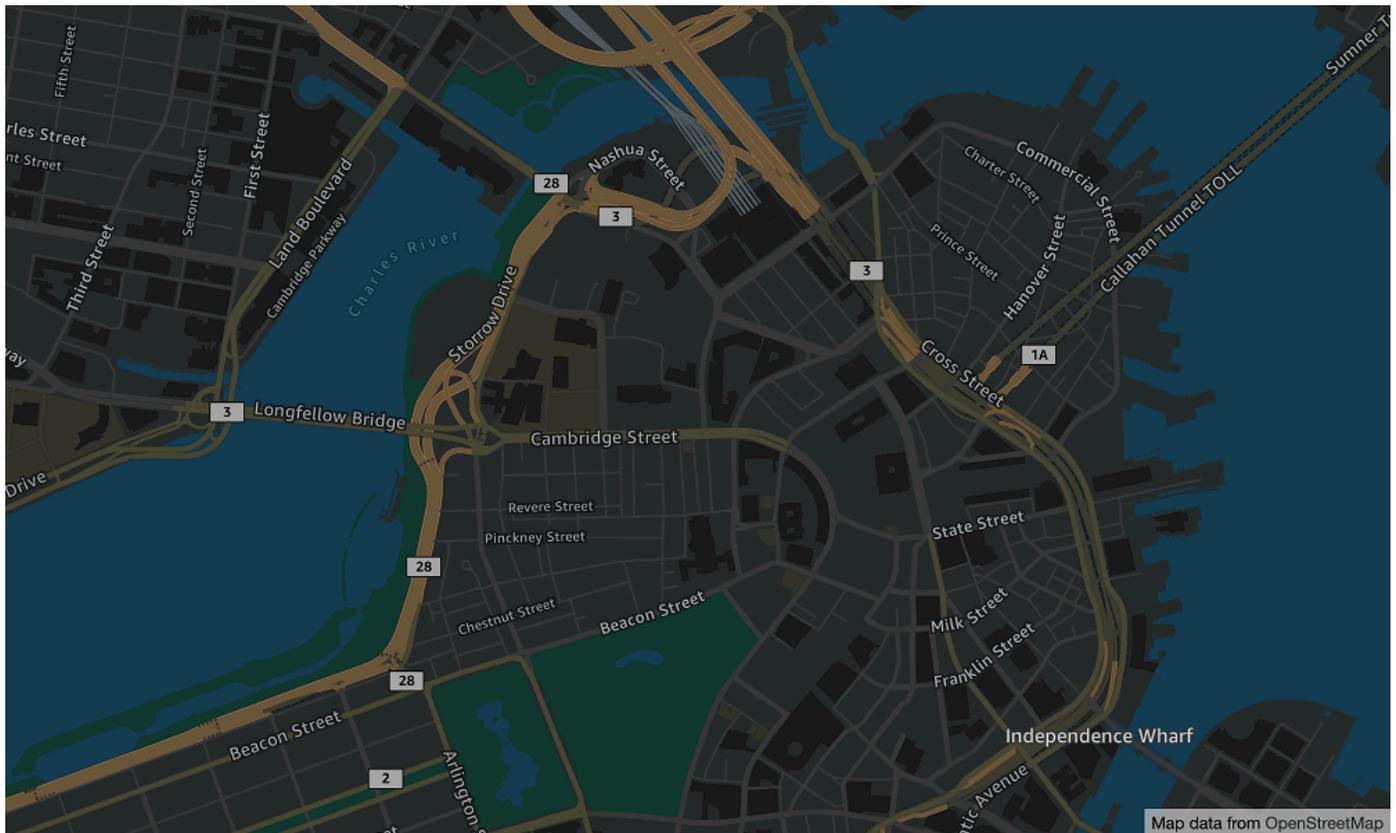
- Amazon Ember Medium, Noto Sans Medium
- Amazon Ember Regular Italic, Noto Sans Italic
- Amazon Ember Regular, Noto Sans Regular
- Amazon Ember Regular, Noto Sans Regular, Noto Sans Arabic Regular
- Amazon Ember Condensed RC Bold, Noto Sans Bold, Noto Sans Arabic Condensed Bold
- Amazon Ember Bold,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

Les polices utilisées par `VectorOpenDataStandardLight` sont des polices combinées utilisées Amazon Ember pour la plupart des glyphes, à l'exception des glyphes non Noto Sans pris en charge par Amazon Ember.

Open Data Standard Dark

Open Data Standard Dark



Nom du style de carte : `VectorOpenDataStandardDark`

Il s'agit d'un style de carte à thème sombre qui fournit un fond de carte détaillé pour le monde, adapté à l'utilisation de sites Web et d'applications mobiles. Cela inclut les autoroutes, les routes principales, les routes secondaires, les voies ferrées, les plans d'eau, les villes, les parcs, les monuments, les empreintes des bâtiments et les limites administratives.

Ce fond de carte est basé sur la [distribution cartographique OSM Daylight](#) compilée par les contributeurs OpenStreetMap (OSM). La communauté OSM compte plus de 1,8 million de contributeurs qui mettent à jour plus de 500 000 fonctionnalités par jour. Amazon Location Service intègre régulièrement ces modifications.

Fonts

Amazon Location diffuse des polices à l'aide de [GetMapGlyphs](#). Les piles de polices disponibles pour cette carte sont les suivantes :

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

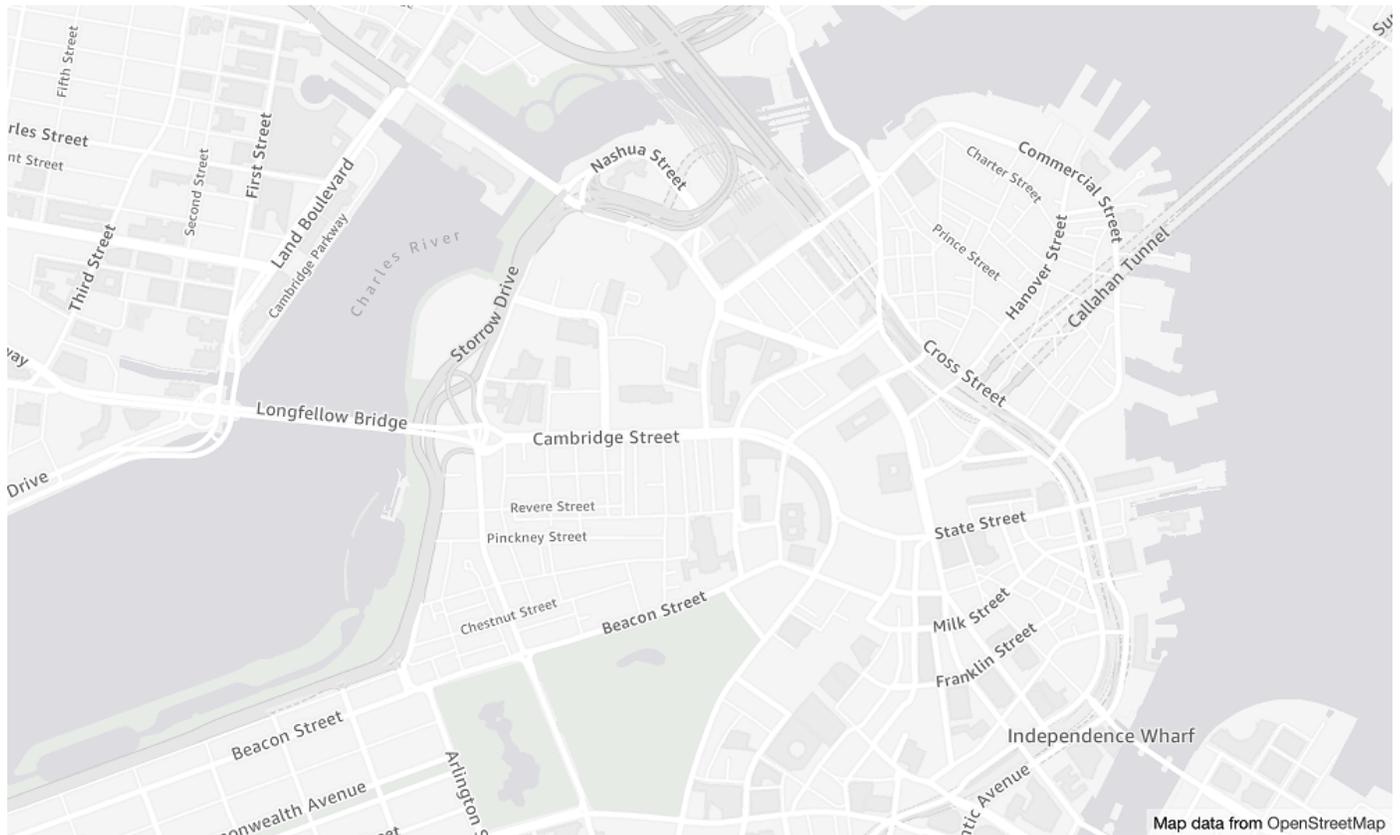
- Amazon Ember Medium, Noto Sans Medium
- Amazon Ember Regular Italic, Noto Sans Italic
- Amazon Ember Regular, Noto Sans Regular
- Amazon Ember Regular, Noto Sans Regular, Noto Sans Arabic Regular
- Amazon Ember Condensed RC Bold, Noto Sans Bold, Noto Sans Arabic Condensed Bold
- Amazon Ember Bold,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

Les polices utilisées par VectorOpenDataStandardDark sont des polices combinées utilisées Amazon Ember pour la plupart des glyphes, à l'exception des glyphes non Noto Sans pris en charge par Amazon Ember.

Open Data Visualization Light

Lampe de visualisation des données ouvertes



Nom du style de carte : `VectorOpenDataVisualizationLight`

Il s'agit d'un style sur le thème de la lumière, avec des couleurs atténuées et moins de fonctionnalités, ce qui facilite la compréhension des données superposées.

Ce fond de carte est basé sur la [distribution cartographique OSM Daylight](#) compilée par les contributeurs OpenStreetMap (OSM). La communauté OSM compte plus de 1,8 million de contributeurs qui mettent à jour plus de 500 000 fonctionnalités par jour. Amazon Location Service intègre régulièrement ces modifications.

Fonts

Amazon Location diffuse des polices à l'aide de [GetMapGlyphs](#). Les piles de polices disponibles pour cette carte sont les suivantes :

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

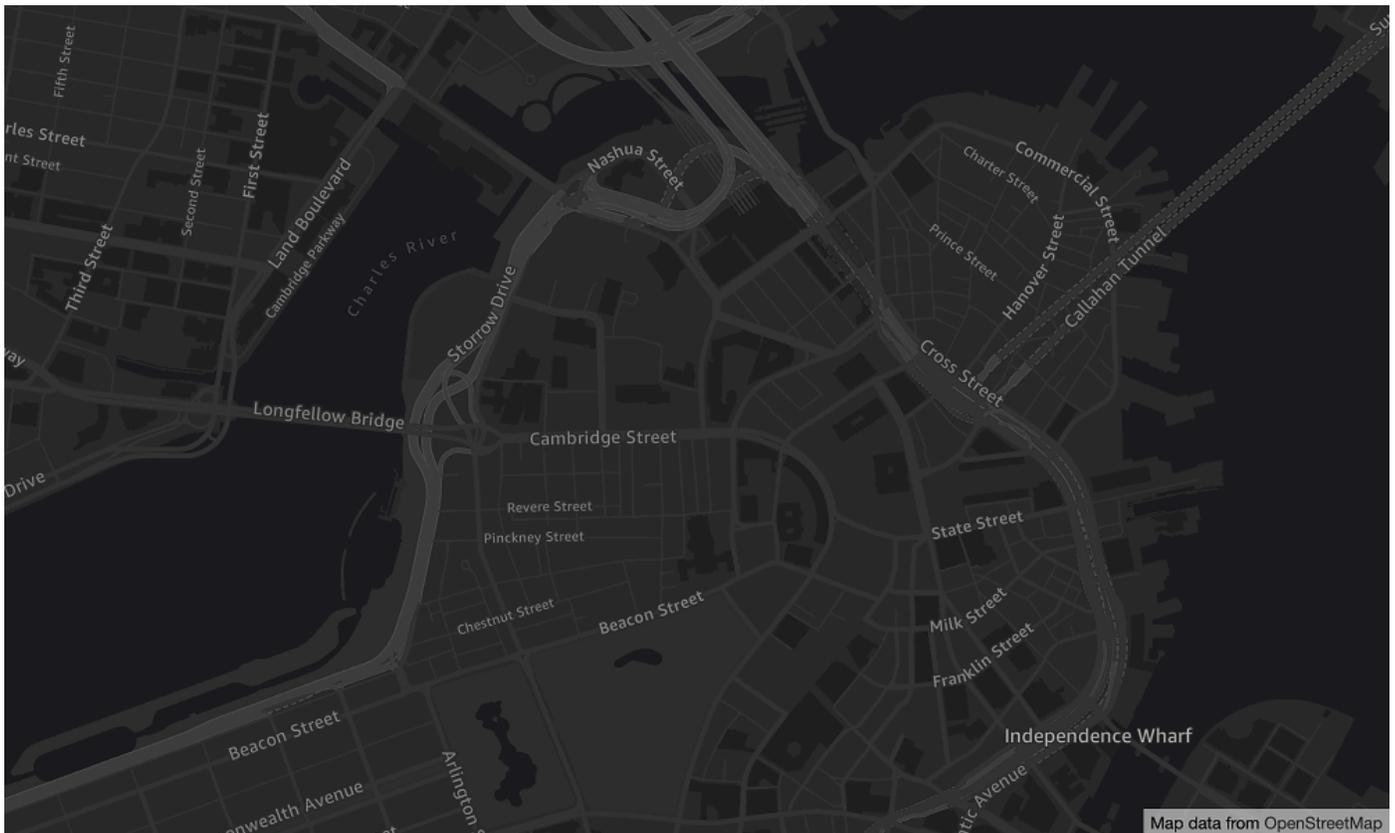
- Amazon Ember Regular Italic, Noto Sans Italic
- Amazon Ember Regular, Noto Sans Regular
- Amazon Ember Regular, Noto Sans Regular, Noto Sans Arabic Regular
- Amazon Ember Condensed RC Bold, Noto Sans Bold, Noto Sans Arabic Condensed Bold
- Amazon Ember Bold, Noto Sans Bold, Noto Sans Arabic Bold
-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

Les polices utilisées par `VectorOpenDataVisualizationLight` sont des polices combinées utilisées Amazon Ember pour la plupart des glyphes, à l'exception des glyphes non Noto Sans pris en charge par Amazon Ember

Open Data Visualization Dark

Visualisation des données ouvertes dans le noir



Nom du style de carte : `VectorOpenDataVisualizationDark`

Il s'agit d'un style à thème sombre avec des couleurs atténuées et moins de fonctionnalités, ce qui facilite la compréhension des données superposées.

Ce fond de carte est basé sur la [distribution cartographique OSM Daylight](#) compilée par les contributeurs OpenStreetMap (OSM). La communauté OSM compte plus de 1,8 million de contributeurs qui mettent à jour plus de 500 000 fonctionnalités par jour. Amazon Location Service intègre régulièrement ces modifications.

Fonts

Amazon Location diffuse des polices à l'aide de [GetMapGlyphs](#). Les piles de polices disponibles pour cette carte sont les suivantes :

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

Les polices utilisées par `VectorOpenDataVisualizationDark` sont des polices combinées utilisées Amazon Ember pour la plupart des glyphes, à l'exception des glyphes non Noto Sans pris en charge par Amazon Ember.

Couverture : données ouvertes

Open Data inclut des cartes couvrant le monde entier à afficher à l'aide d'une [ressource cartographique Amazon Location Service](#).

Note

Les données ouvertes sont destinées à être utilisées uniquement avec les ressources cartographiques d'Amazon Location Service. Vous ne pouvez pas utiliser Open Data en tant que fournisseur de données pour prendre en charge les requêtes de géocodage, de géocodage inversé et de recherches, ni pour prendre en charge les requêtes visant à calculer un itinéraire.

Conditions d'utilisation et attribution des données : Open Data

Avant d'utiliser Open Data, assurez-vous de respecter toutes les exigences légales applicables, y compris les termes de licence applicables aux données ouvertes et AWS.

Pour plus d'informations sur les AWS exigences, consultez les [conditions de service AWS](#).

Pour plus d'informations sur les directives d'attribution des données ouvertes, consultez les OpenStreetMap directives relatives au [droit d'auteur et aux licences](#) et aux directives OpenStreetMap d'[attribution des licences](#).

Signalement des erreurs et contribution aux données ouvertes

OpenStreetMap (OSM) et Natural Earth sont des projets de données ouvertes pilotés par la communauté. Si vous rencontrez un problème avec les données, vous pouvez signaler les erreurs ou apporter directement des corrections ou des suggestions.

- Pour signaler une erreur ou proposer une suggestion dans OSM, vous pouvez créer une note sur la carte. Il s'agit d'un commentaire sur la carte qui aide les contributeurs à apporter des corrections à la carte. Vous créez des notes via le [OpenStreetMap site Web](#). Pour plus d'informations sur les notes, consultez la section [Notes](#) sur le OpenStreetMap wiki.
- Pour plus d'informations sur la contribution directe aux données cartographiques OpenStreetMap, y compris l'ajout de lieux et la correction des erreurs, consultez la section [Contribuer aux données cartographiques](#) sur le OpenStreetMap wiki.
- Pour soumettre une demande de correction de données dans Natural Earth, vous pouvez soumettre un problème via le [site Web de Natural Earth](#).

Note

La correction des erreurs OpenStreetMap peut se faire rapidement, mais les corrections peuvent prendre un certain temps avant d'apparaître dans la distribution cartographique Daylight des données OSM utilisées par le fournisseur de données ouvertes. Le site Web [Daylight Map Distribution](#) fournit de plus amples informations sur le processus. En outre, Amazon Location Service met à jour les données cartographiques utilisées dans Amazon Location Service environ une fois par mois.

Fonctionnalités par fournisseur de données

Cette section décrit les fonctionnalités disponibles dans Amazon Location Service, classées par fournisseur de données.

Le tableau suivant fournit un aperçu général des fonctionnalités.

Fournisseur de données	Couverture géographique	Couverture des fonctionnalités	Région AWS
Esri	Globale	Cartes, lieux, itinéraires	Toutes les régions dans lesquelles Amazon Location est disponible.
Saisir	Asie du sud-est	Cartes, lieux, itinéraires	Asie-Pacifique (Singapour)ap-southeast-1 , uniquement.
ICI	Globale	Cartes, lieux, itinéraires	Toutes les régions dans lesquelles Amazon Location est disponible.
Données ouvertes	Globale	Mappages	Toutes les régions dans lesquelles Amazon Location est disponible.

Les onglets suivants présentent les détails de chaque zone fonctionnelle.

Map Features

Le tableau suivant présente les caractéristiques cartographiques par fournisseur de données. Pour plus d'informations sur les concepts cartographiques, consultez [Mappages](#).

Fournisseur de données	Types de cartes pris en charge	Niveaux de zoom vectoriel	Niveaux de zoom matriciels
Esri	Vector	0-15	0-23
	Raster (imagerie)		

Fournisseur de données	Types de cartes pris en charge	Niveaux de zoom vectoriel	Niveaux de zoom matriciels
	Pour plus d'informations, consultez Styles de carte Esri.		
Saisir	Vector (Asie du Sud-Est uniquement) Pour plus d'informations, consultez Saisissez les styles de carte.	0-14	none
ICI	Vector Raster (imagerie) Hybride Pour plus d'informations, consultez Styles de carte HERE.	1-17	0-19
Données ouvertes	Vector Pour plus d'informations, consultez Styles de carte Open Data.	0-15	none

 Note

Les niveaux de zoom représentent les paramètres maximum et minimum, tels que définis dans les API de chaque fournisseur. Différentes zones de la carte peuvent avoir des

maximums différents ; par exemple, les tuiles océaniques peuvent avoir des niveaux de zoom moins détaillés que les zones des grandes villes.

MapLibre (et d'autres moteurs de rendu cartographique) vous permettent de définir des niveaux de zoom minimum et maximum, et respecteront également les niveaux de zoom du fournisseur de données dans une zone. Vous n'avez donc pas à écrire de code pour gérer ces écarts.

Places and Search

Le tableau suivant présente les fonctionnalités de localisation et de recherche par fournisseur de données. Pour plus d'informations sur les concepts de lieux, consultez [Recherche de lieux](#).

Fournisseur de données	Géocodage	Géocodage inverse	Remplissage automatique	GetPlace
Esri	Toutes les fonctionnalités, sauf : PlaceId	Toutes les fonctionnalités, sauf : TimeZone PlaceId	Toutes les fonctionnalités	Toutes les fonctionnalités
Saisir	Toutes les fonctionnalités, sauf : type d'unité Catégories non prises en charge	Toutes les fonctionnalités	Toutes les fonctionnalités	Toutes les fonctionnalités, sauf : type d'unité SubMunicipality
ICI	Toutes les fonctionnalités, sauf : numéro d'unité	Toutes les fonctionnalités	Toutes les fonctionnalités	Toutes les fonctionnalités, sauf : numéro d'unité

Fournisseur de données	Géocodage	Géocodage inverse	Remplissage automatique	GetPlace
	type d'unité			type d'unité
	pertinence			SubMunicipality
	Limitations supplémentaires relatives au filtrage			
Données ouvertes	Non pris en charge	Non pris en charge	Non pris en charge	Supporte uniquement : SubMunicipality

Route features

Le tableau suivant présente les caractéristiques de l'itinéraire par fournisseur de données. Pour plus d'informations sur les concepts d'itinéraire, consultez [Acheminements](#). Pour une description plus détaillée des limites de la matrice de routes, voir [Restrictions relatives aux positions de départ et de destination](#).

Fournisseur de données	Modes de voyage	Calculer l'itinéraire	Matrice d'itinéraires
Esri	Voiture, camion, marche	Le départ et la destination doivent être situés à moins de 400 km l'un de l'autre. La durée totale du trajet ne peut pas dépasser 400 minutes.	Jusqu'à 10 positions de départ et de destination. Non pris en charge en Corée. Toutes les paires de départ et de destination doivent se trouver

Fournisseur de données	Modes de voyage	Calculer l'itinéraire	Matrice d'itinéraires
		ArrivalTime n'est pas pris en charge.	à moins de 400 km l'une de l'autre.
Saisir	Voiture, moto. Promenade à pied et à vélo dans certaines villes .	Aucune limite de distance.	Jusqu'à 350 positions de départ et de destination.
ICI	Voiture, camion, marche	Aucune limite de distance. Les itinéraires situés à plus de 10 km en dehors d'un cercle autour des positions de départ et de destination ne seront pas calculés.	Jusqu'à 350 positions de départ et de destination. Toutes les positions de départ et de destination doivent se situer dans un cercle de 180 km. Les itinéraires plus longs sont pris en charge, avec des restrictions supplémentaires .
Données ouvertes	Non pris en charge	Non pris en charge	Non pris en charge

Conditions d'utilisation et attribution des données pour les fournisseurs de données

Avant de faire appel à un fournisseur de données, assurez-vous de respecter toutes les exigences légales applicables, y compris les conditions de licence applicables à l'utilisation du fournisseur.

Pour plus d'informations sur les AWS exigences, consultez les [conditions de service AWS](#).

Lorsque vous utilisez un fournisseur de données avec vos ressources Amazon Location pour votre application ou votre documentation, assurez-vous de fournir des attributions pour chaque fournisseur de données que vous utilisez.

Pour plus d'informations sur la conformité et l'attribution pour chaque fournisseur de données, consultez les rubriques suivantes.

- Esri — [Conditions d'utilisation et attribution des données : Esri](#)
- Prenez — [Conditions d'utilisation et attribution des données : Grab](#)
- ICI — [Conditions d'utilisation et attribution des données : ICI](#)
- Données ouvertes — [Conditions d'utilisation et attribution des données : Open Data](#)

Amazon Location, régions et points de terminaison

Amazon Location est disponible dans les AWS régions suivantes :

Régions

Nom de la région	Région	Point de terminaison	Protocole
US East (Ohio)	us-east-2	geo.us-east-2.amazonaws.com	HTTPS
US East (N. Virginia)	us-east-1	geo.us-east-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	geo.us-west-2.amazonaws.com	HTTPS
Asie-Pacifique (Mumbai)	ap-south-1	geo.ap-south-1.amazonaws.com	HTTPS
Asie-Pacifique	ap-southeast-1	geo.ap-southeast-1.amazonaws.com	HTTPS

Nom de la région	Région	Point de terminaison	Protocole
(Singapour)			
Asia Pacific (Sydney)	ap-southeast-2	geo.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	geo.ap-northeast-1.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	geo.ca-central-1.amazonaws.com	HTTPS
Europe (Francfort)	eu-central-1	geo.eu-central-1.amazonaws.com	HTTPS
Europe (Irlande)	eu-west-1	geo.eu-west-1.amazonaws.com	HTTPS
Europe (Londres)	eu-west-2	geo.eu-west-2.amazonaws.com	HTTPS
Europe (Stockholm)	eu-north-1	geo.eu-north-1.amazonaws.com	HTTPS
Amérique du Sud (São Paulo)	sa-east-1	geo.sa-east-1.amazonaws.com	HTTPS

Nom de la région	Région	Point de terminaison	Protocole
AWS GovCloud (US-Ouest)	us-gov-west-1	geo.us-gov-west-1.amazonaws.com	HTTPS
		geo-fips.us-gov-west-1.amazonaws.com	HTTPS

Note

Pour plus d'informations sur l'utilisation des points de terminaison de ce tableau, consultez la section suivante.

Points de terminaison

La syntaxe générale d'un point de terminaison régional Amazon Location est la suivante :

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

Dans le cadre de cette syntaxe, Amazon Location utilise les codes de service suivants :

Service	Code de service
Cartes de localisation Amazon	cartes
Amazon Location Places	lieux
Géofences de localisation Amazon	géofencing
Traceurs de localisation Amazon	suivi
Itinéraires de localisation Amazon	itinéraires

Par exemple, le point de terminaison régional pour Amazon Location Maps pour l'est des États-Unis (Virginie du Nord) serait : `https://maps.geo.us-east-1.amazonaws.com`.

Endpoints d'opération de l'API

La syntaxe d'un point de terminaison du plan de contrôle Amazon Location Service est la suivante :

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

Les actions du plan de contrôle pour Amazon Location Service sont les suivantes :

Service	Point de terminaison	Opération API
Cartes de localisation Amazon	https://cp.maps.geo. <i>région</i> .amazonaws.com	CreateMap DeleteMap DescribeMap ListMaps UpdateMap
Amazon Location Places	https://cp.places.geo. <i>région</i> .amazonaws.com	CreatePlaceIndex DeletePlaceIndex DescribePlaceIndex ListPlaceIndexes UpdatePlaceIndex
Géofences de localisation Amazon	https://cp.geofencing.geo. <i>région</i> .amazonaws.com	CreateGeofenceCollection DeleteGeofenceCollection DescribeGeofenceCollection ListGeofenceCollections UpdateGeofenceCollection
Traceurs de localisation Amazon	https://cp.tracking.geo. <i>région</i> .amazonaws.com	CreateTracker

Service	Point de terminaison	Opération API
		DeleteTracker DescribeTracker UpdateTracker ListTrackers AssociateTrackerConsumer DisassociateTrackerConsumer ListTrackerConsumers
Itinéraires de localisation Amazon	<a href="https://cp.routes.geo.
région .amazonaws.com">https://cp.routes.geo. région .amazonaws.com	CreateRouteCalculator DeleteRouteCalculator DescribeRouteCalculator ListRouteCalculators UpdateRouteCalculator
Métadonnées de localisation Amazon	<a href="https://cp.metadata.geo.
région .amazonaws.com">https://cp.metadata.geo. région .amazonaws.com	CreateKey DeleteKey DescribeKey ListKeys UpdateKey

Quotas d'Amazon Location Service

Cette rubrique fournit un résumé des limites de débit et des quotas pour Amazon Location Service.

Note

Si vous avez besoin d'un quota plus élevé, vous pouvez utiliser la console Service Quotas pour [demander des augmentations de quota](#) pour des quotas ajustables. Lorsque vous demandez une augmentation de quota, sélectionnez la région dans laquelle vous souhaitez augmenter le quota, car la plupart des quotas sont spécifiques à la AWS région.

Les Quotas de Service sont le nombre maximum de ressources que vous pouvez avoir par AWS compte et par AWS région. Amazon Location Service refuse les demandes supplémentaires qui dépassent le quota de service.

Limites de taux (quotas commençant par un taux de...) sont le nombre maximal de demandes par seconde, avec un taux de rafale de 80 % de la limite pendant une partie de la seconde, défini pour chaque opération d'API. Amazon Location Service limite les demandes qui dépassent la limite de débit de l'opération.

Nom	Par défaut	Ajustable	Description
Ressources clés de l'API par compte	Chaque région prise en charge : 500	Non	Le nombre maximum de ressources clés d'API (actives ou expirées) que vous pouvez avoir par compte.
Ressources de collecte Geofence par compte	Chaque région prise en charge : 1 500	Oui	Le nombre maximum de ressources Geofence Collection que vous pouvez créer par compte.
Geofences par collection Geofence	Chaque région prise en charge : 50 000	Non	Le nombre maximum de géofences que vous pouvez créer par collection de géofences.
Cartographier les ressources par compte	Chaque Région prise en charge : 40	Oui	Le nombre maximum de ressources cartograp

Nom	Par défaut	Ajusté	Description
			hiques que vous pouvez créer par compte.
Placer les ressources de l'index par compte	Chaque Région prise en charge : 40	Oui	Le nombre maximum de ressources Place Index que vous pouvez créer par compte.
Taux de demandes AssociateTrackerConsumer d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de AssociateTrackerConsumer demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes BatchDeleteDevicePositionHistory d'API	Chaque région prise en charge : 50 par seconde	Oui	Le nombre maximum de BatchDeleteDevicePositionHistory demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes BatchDeleteGeofence d'API	Chaque région prise en charge : 50 par seconde	Oui	Le nombre maximum de BatchDeleteGeofence demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes BatchEvaluateGeofences d'API	Chaque région prise en charge : 50 par seconde	Oui	Le nombre maximum de BatchEvaluateGeofences demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.

Nom	Par défaut	Ajusté	Description
Taux de demandes BatchGetDevicePosition d'API	Chaque région prise en charge : 50 par seconde	Oui	Le nombre maximum de BatchGetDevicePosition demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes BatchPutGeofence d'API	Chaque région prise en charge : 50 par seconde	Oui	Le nombre maximum de BatchPutGeofence demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes BatchUpdateDevicePosition d'API	Chaque région prise en charge : 50 par seconde	Oui	Le nombre maximum de BatchUpdateDevicePosition demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes CalculateRoute d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de CalculateRoute demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes CalculateRouteMatrix d'API	Chaque région prise en charge : 5 par seconde	Oui	Le nombre maximum de CalculateRouteMatrix demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.

Nom	Par défaut	Ajusté	Description
Taux de demandes CreateGeofenceCollection d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de CreateGeofenceCollection demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes CreateKey d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de CreateKey demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes CreateMap d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de CreateMap demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes CreatePlaceIndex d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de CreatePlaceIndex demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes CreateRouteCalculator d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de CreateRouteCalculator demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.

Nom	Par défaut	Ajuste	Description
Taux de demandes CreateTracker d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de CreateTracker demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes DeleteGeofenceCollection d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de DeleteGeofenceCollection demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes DeleteKey d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de DeleteKey demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes DeleteMap d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de DeleteMap demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes DeletePlaceIndex d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de DeletePlaceIndex demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.

Nom	Par défaut	Ajusté	Description
Taux de demandes DeleteRouteCalculator d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de DeleteRouteCalculator demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes DeleteTracker d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de DeleteTracker demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes DescribeGeofenceCollection d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de DescribeGeofenceCollection demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes DescribeKey d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de DescribeKey demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes DescribeMap d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de DescribeMap demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.

Nom	Par défaut	Ajuste	Description
Taux de demandes DescribePlaceIndex d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de DescribePlaceIndex demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes DescribeRouteCalculator d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de DescribeRouteCalculator demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes DescribeTracker d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de DescribeTracker demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes DisassociateTrackerConsumer d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de DisassociateTrackerConsumer demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes ForecastGeofenceEvents d'API	Chaque région prise en charge : 50 par seconde	Oui	Le nombre maximum de ForecastGeofenceEvents demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.

Nom	Par défaut	Ajusté	Description
Taux de demandes GetDevicePosition d'API	Chaque région prise en charge : 50 par seconde	Oui	Le nombre maximum de GetDevicePosition demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes GetDevicePositionHistory d'API	Chaque région prise en charge : 50 par seconde	Oui	Le nombre maximum de GetDevicePositionHistory demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes GetGeofence d'API	Chaque région prise en charge : 50 par seconde	Oui	Le nombre maximum de GetGeofence demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes GetMapGlyphs d'API	Chaque région prise en charge : 50 par seconde	Oui	Le nombre maximum de GetMapGlyphs demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes GetMapSprites d'API	Chaque région prise en charge : 50 par seconde	Oui	Le nombre maximum de GetMapSprites demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.

Nom	Par défaut	Ajuste	Description
Taux de demandes GetMapStyleDescriptor d'API	Chaque région prise en charge : 50 par seconde	Oui	Le nombre maximum de GetMapStyleDescriptor demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes GetMapTile d'API	Chaque Région prise en charge : 500 par seconde	Oui	Le nombre maximum de GetMapTile demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes GetPlace d'API	Chaque région prise en charge : 50 par seconde	Oui	Le nombre maximum de GetPlace demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes ListDevicePositions d'API	Chaque région prise en charge : 50 par seconde	Oui	Le nombre maximum de ListDevicePositions demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes ListGeofenceCollections d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de ListGeofenceCollections demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.

Nom	Par défaut	Ajuste	Description
Taux de demandes ListGeofences d'API	Chaque région prise en charge : 50 par seconde	Oui	Le nombre maximum de ListGeofences demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes ListKeys d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de ListKeys demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes ListMaps d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de ListMaps demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes ListPlaceIndexes d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de ListPlaceIndexes demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes ListRouteCalculators d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de ListRouteCalculators demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.

Nom	Par défaut	Ajuste	Description
Taux de demandes ListTagsForResource d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de ListTagsForResource demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes ListTrackerConsumerConsumers d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de ListTrackerConsumerConsumers demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes ListTrackers d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de ListTrackers demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes PutGeofence d'API	Chaque région prise en charge : 50 par seconde	Oui	Le nombre maximum de PutGeofence demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes SearchPlaceIndexForPosition d'API	Chaque région prise en charge : 50 par seconde	Oui	Le nombre maximum de SearchPlaceIndexForPosition demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.

Nom	Par défaut	Ajuste	Description
Taux de demandes SearchPlaceIndexForSuggestions d'API	Chaque région prise en charge : 50 par seconde	Oui	Le nombre maximum de SearchPlaceIndexForSuggestions demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes SearchPlaceIndexForText d'API	Chaque région prise en charge : 50 par seconde	Oui	Le nombre maximum de SearchPlaceIndexForText demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes TagResource d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de TagResource demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes UntagResource d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de UntagResource demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes UpdateGeofenceCollection d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de UpdateGeofenceCollection demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.

Nom	Par défaut	Ajusté	Description
Taux de demandes UpdateKey d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de UpdateKey demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes UpdateMap d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de UpdateMap demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes UpdatePlaceIndex d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de UpdatePlaceIndex demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes UpdateRouteCalculator d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de UpdateRouteCalculator demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Taux de demandes UpdateTracker d'API	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum de UpdateTracker demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.

Nom	Par défaut	Ajuste	Description
Taux de demandes VerifyDevicePosition d'API	Chaque région prise en charge : 50 par seconde	Oui	Le nombre maximum de VerifyDevicePosition demandes que vous pouvez effectuer par seconde. Les autres demandes sont bloquées.
Ressources du calculateur d'itinéraires par compte	Chaque Région prise en charge : 40	Oui	Le nombre maximum de ressources Route Calculator que vous pouvez créer par compte.
Suivez les consommateurs par traceur	Chaque région prise en charge : 5	Non	Le nombre maximum de collections Geofence auxquelles la ressource Tracker peut être associée.
Ressources de suivi par compte	Chaque région prise en charge : 500	Oui	Le nombre maximum de ressources Tracker que vous pouvez créer par compte.

Note

Vous pouvez surveiller votre utilisation par rapport à vos quotas avec Cloudwatch. Pour plus d'informations, consultez [Utilisation CloudWatch pour surveiller l'utilisation par rapport aux quotas](#).

Gestion des quotas de votre service de localisation Amazon

Amazon Location Service est intégré à Service Quotas, un AWS service qui vous permet de consulter et de gérer vos quotas depuis un emplacement central. Pour plus d'informations, veuillez consulter [Qu'est-ce que Service Quotas?](#) dans le Guide de l'utilisateur Service Quotas.

Service Quotas permet de rechercher facilement la valeur de vos quotas de service Amazon Location.

AWS Management Console

Pour consulter les quotas du service Amazon Location à l'aide de la console

1. Ouvrez la console Service Quotas à l'adresse <https://console.aws.amazon.com/servicequotas/>.
2. Dans le panneau de navigation, choisissez Services AWS .
3. Dans la liste des AWS services, recherchez et sélectionnez Amazon Location.

Dans la liste Service quotas (Quotas de service), vous pouvez voir le nom du Service Quota, la valeur appliquée (le cas échéant), le quota AWS par défaut et si la valeur du quota est réglable.

4. Pour afficher des informations supplémentaires sur un quota de service, notamment la description, choisissez le nom du quota.
5. (Facultatif) Pour demander une augmentation de quota, sélectionnez le quota que vous souhaitez augmenter, sélectionnez Request quota increase (Demander une augmentation de quota), saisissez ou sélectionnez les informations requises, puis sélectionnez Request (Demander).

Pour utiliser davantage les quotas de service à l'aide de la console, consultez le [Guide de l'utilisateur Service Quotas](#). Pour demander une augmentation de quota, consultez [Demande d'augmentation de quota](#) dans le Guide de l'utilisateur Service Quotas.

AWS CLI

Pour consulter les quotas du service de localisation Amazon à l'aide du AWS CLI

Exécutez la commande suivante pour afficher les quotas Amazon Location par défaut.

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

Pour travailler davantage avec les quotas de service à l'aide du AWS CLI, consultez le Guide de [référence des AWS CLI commandes Service Quotas](#). Pour demander une augmentation de quota, consultez la commande [request-service-quota-increase](#) dans la [référence des commandes AWS CLI](#).

Commencer à utiliser Amazon Location Service en tant que développeur

Vous pouvez utiliser Amazon Location Service pour fournir des fonctionnalités liées à la géographie pour les applications dans de nombreux formats et systèmes différents, notamment les services Web principaux, les applications Web et les applications mobiles. De nombreux outils sont fournis pour vous aider à créer vos applications, notamment des SDK, des bibliothèques et des exemples de code.

Cette section fournit des informations et des liens pour vous aider à démarrer avec Amazon Location. En particulier, les rubriques suivantes fournissent des informations qui peuvent vous être très utiles :

- [Scénarios et cas d'utilisation](#)— Une liste de scénarios de développement et la manière dont Amazon Location Service peut vous aider à les réaliser.
- [SDK et outils Amazon Location](#) : kits de développement logiciel (SDK) et bibliothèques qui vous aideront à programmer avec Amazon Location.
- [Référence d'API Amazon Location Service](#) : référence aux principales API Amazon Location fournies avec le AWS SDK.
- [Exemples de code](#) — Cette section fournit des exemples qui vous aideront à démarrer ou à ajouter des fonctionnalités à votre application existante.
- [Tutoriel de démarrage rapide](#) — Ce didacticiel vous montre comment créer votre première application. Il existe des versions du didacticiel pour créer une application Web ou une application mobile basée sur Android.
- [Concepts du service de location Amazon](#)— Cette section de ce guide décrit les concepts de base d'Amazon Location, notamment les sections sur les cartes, la recherche de lieux, les itinéraires, les géofences et les traceurs.
- [Amplify](#) — Amplify est une solution complète qui intègre la plupart des fonctionnalités nécessaires à la création d'applications Web et mobiles à l'aide du. AWS Cloud Si vous utilisez déjà Amplify ou si vous choisissez d'utiliser Amplify, il dispose d'une bibliothèque géographique utilisant Amazon Location Service intégré que vous pouvez utiliser. [Pour commencer à utiliser Amplify Geo, consultez la documentation ici.](#)

Scénarios et cas d'utilisation

Amazon Location Service est un service qui s'exécute dans le AWS Cloud. Vous pouvez l'appeler depuis vos propres instances Amazon EC2 dans le cloud, mais de nombreuses applications de cartographie s'exécuteront sur des appareils ou sur une combinaison d'appareils et du cloud. Vous trouverez ci-dessous quelques scénarios typiques et la manière dont vous pourriez aborder leur développement.

- Une application principale qui vous aide à optimiser les itinéraires pour les conducteurs de votre flotte.

Vous pouvez écrire une application qui s'exécute sur [Amazon EC2](#) et AWS Cloud qui utilise Amazon Location Service pour [calculer les matrices d'itinéraires](#) en tant qu'entrée d'un optimiseur d'itinéraires pour votre flotte. Utilisez le [AWS SDK](#) pour passer des appels vers Amazon Location.

- Une application Web qui permet à vos clients de trouver les emplacements de votre entreprise.

Vous pouvez créer un site Web qui s'exécute sur des instances Amazon EC2, y compris une application basée sur la localisation. Utilisez le [AWS SDK pour](#) développer une application Web JavaScript permettant de rechercher des emplacements à l'aide de la [recherche de lieux](#) et d'afficher les résultats sur une [carte](#) à l'aide MapLibre de. Utilisez le SDK Amazon Location pour faciliter la programmation avec la localisation.

- Ajoutez des fonctionnalités de localisation à une application iOS ou Android existante.

Vous pouvez utiliser le AWS SDK pour Swift (iOS) ou [Kotlin](#) (Android) pour appeler Amazon Location afin d'ajouter des fonctionnalités de [recherche de lieux et de cartes](#) à votre application. MapLibre À utiliser pour afficher des cartes. Des [AWS SDK](#) supplémentaires sont disponibles pour d'autres langues.

- Suivez les actifs (appareils ou véhicules) et recevez des mises à jour lorsqu'ils entrent ou sortent des zones que vous définissez.

Une application de suivi des appareils se compose de plusieurs parties.

- [Une ressource de suivi doit être créée pour chaque appareil que vous suivez.](#) Il doit envoyer des mises à jour de position à Amazon Location Service, par exemple en utilisant [MQTT](#).
- Créez des [barrières géographiques pour définir les zones dans lesquelles vous souhaitez enregistrer les événements d'entrée et de sortie de vos actifs.](#)
- Vous pouvez utiliser [Amazon EC2](#) ou [AWS Lambda](#) pour répondre à vos événements lorsque des actifs entrent ou sortent des zones de géolocalisation.

- Vous pouvez développer cela pour créer des applications Web ou des applications pour appareils afin de suivre et d'afficher l'emplacement de vos actifs sur des cartes.

La section suivante fournit des informations détaillées sur les outils et les bibliothèques disponibles pour chaque aspect d'Amazon Location Service.

SDK et outils pour utiliser Amazon Location Service

Plusieurs outils vous aideront à utiliser Amazon Location Service.

- AWS SDK — Les kits de développement AWS logiciel (SDK) sont disponibles dans de nombreux langages de programmation courants et fournissent une API, des exemples de code et une documentation qui facilitent la création d'applications dans votre langue préférée. Les AWS SDK incluent les principales API et fonctionnalités d'Amazon Location, notamment l'accès à Maps, Places search, Routes, Geofence et Trackers. Pour en savoir plus sur les SDK disponibles pour être utilisés avec Amazon Location Service pour différentes applications et langues, consultez [SDK par langue](#).
- MapLibre— Amazon Location Service recommande de rendre les cartes à l'aide du moteur de [MapLibre](#) rendu. MapLibre est un moteur permettant d'afficher des cartes dans des applications Web ou mobiles. MapLibre dispose également d'un modèle de plugin et prend en charge l'interface utilisateur pour la recherche et les itinéraires dans certaines langues et plateformes. Pour en savoir plus sur MapLibre son utilisation et ses fonctionnalités, consultez [MapLibre](#).
- SDK Amazon Location — Le SDK Amazon Location est un ensemble de bibliothèques open source qui facilitent le développement d'applications avec Amazon Location Service. Les bibliothèques fournissent des fonctionnalités permettant l'authentification pour les applications mobiles et Web, le suivi de la localisation pour les applications mobiles, la conversion entre les types de données Amazon Location et [GeoJSON](#), ainsi qu'un package hébergé du client Amazon Location pour le AWS SDK v3. Pour en savoir plus sur le SDK Amazon Location, consultez [SDK Amazon Location](#).

SDK par langue

Les tableaux suivants fournissent des informations sur AWS les SDK et MapLibre les versions pour les langages et les frameworks, par type d'application : application Web, mobile ou principale.

Versions du SDK

Nous vous recommandons d'utiliser la version la plus récente du AWS SDK, ainsi que tout autre SDK, que vous utilisez dans vos projets, et de maintenir les SDK à jour. Le AWS SDK vous fournit les fonctionnalités les plus récentes, ainsi que des mises à jour de sécurité. Pour trouver la dernière version du AWS SDK pour JavaScript, par exemple, consultez la rubrique relative à l'[installation du navigateur](#) dans le AWS SDK pour obtenir JavaScript de la documentation.

Web frontend

Les AWS SDK et MapLibre versions suivants sont disponibles pour le développement d'applications Web frontales.

Langage/Framework	Kit SDK AWS	Cadre de rendu
Entièrement pris en charge		
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/
Partiellement pris en charge		
Flottement	https://docs.amplify.aws/start/q/integration/flutter/ Flutter n'est pas encore entièrement pris en charge par Amplify AWS, mais un support limité est proposé par le biais d'Amplify.	https://github.com/maplibre/flutter-maplibre-gl La bibliothèque MapLibre Flutter est considérée comme expérimentale.

Langage/Framework	Kit SDK AWS	Cadre de rendu
Node.js	https://aws.amazon.com/sdk-for-javascript/	Node.js n'est pas MapLibre pris en charge.
PHP	https://aws.amazon.com/sdk-for-php/	Il n'y a aucun MapLibre support pour PHP.

Mobile frontend

Les AWS SDK et MapLibre versions suivants sont disponibles pour le développement d'applications frontales mobiles.

Langage/Framework	AWS SDK	Cadre de rendu
Entièrement pris en charge		
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 d'authentification mobile Amazon Location Service pour Android : amazon-location-mobile-auth https://github.com/aws-geospatial/sdk-android</p> <p>SDK de suivi mobile Amazon Location Service pour Android : amazon-location-mobile-tracking https://github.com/aws-geospatial/sdk-android</p>	<p>https://maplibre.org/projects/maplibre-native/</p> <p>Nécessite des liaisons personnalisées, comme MapLibre c'est le cas en Java.</p>

Langage/Framework	AWS SDK	Cadre de rendu
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 d'authentification mobile Amazon Location Service pour iOS : amazon-location-mobile-auth https://github.com/aws-geospatial/sdk-ios</p> <p>SDK de suivi mobile Amazon Location Service pour iOS : amazon-location-mobile-tracking https://github.com/aws-geospatial/sdk-ios</p>	https://maplibre.org/projects/maplibre-native/
Partiellement pris en charge		
Flottement	<p>https://docs.amplify.aws/start/q/integration/flutter/</p> <p>Flutter n'est pas encore entièrement pris en charge par Amplify AWS, mais un support limité est proposé par le biais d'Amplify.</p>	<p>https://github.com/maplibre/flutter-maplibre-gl</p> <p>La bibliothèque MapLibre Flutter est considérée comme expérimentale.</p>

Backend application

Les AWS SDK suivants sont disponibles pour le développement d'applications principales. MapLibre n'est pas répertorié ici, car le rendu cartographique n'est généralement pas nécessaire pour les applications principales.

Langue	Kit SDK 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/
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/ Le AWS SDK pour Rust est en version préliminaire pour les développeurs.

Utilisation d' MapLibre outils et de bibliothèques avec Amazon Location

L'un des outils importants pour créer des applications interactives avec Amazon Location est MapLibre. [MapLibre](#) est principalement un moteur de rendu permettant d'afficher des cartes dans une application Web ou mobile. Cependant, il inclut également la prise en charge des plug-ins et

fournit des fonctionnalités permettant de travailler avec d'autres aspects d'Amazon Location. Les outils suivants décrivent les outils que vous pouvez utiliser, en fonction de la zone géographique avec laquelle vous souhaitez travailler.

Note

Pour utiliser n'importe quel aspect d'Amazon Location, installez le [AWS SDK correspondant à la langue que vous souhaitez utiliser](#).

• Cartes

Pour afficher des cartes dans votre application, vous avez besoin d'un moteur de rendu cartographique qui utilise les données fournies par Amazon Location et dessine à l'écran. Les moteurs de rendu de carte fournissent également des fonctionnalités permettant de déplacer et de zoomer la carte, ou d'ajouter des marqueurs, des punaises et d'autres annotations à la carte.

Amazon Location Service recommande de rendre les cartes à l'aide du moteur de [MapLibre](#) rendu. MapLibre GL JS est un moteur d'affichage de cartes JavaScript, tandis que MapLibre Native fournit des cartes pour iOS ou Android.

MapLibre dispose également d'un écosystème de plug-ins pour étendre les fonctionnalités de base. Pour plus d'informations, rendez-vous sur <https://maplibre.org/maplibre-gl-js-docs/plugins/>.

• Recherche de lieux

Pour simplifier la création d'une interface utilisateur de recherche, vous pouvez utiliser le [MapLibre géocodeur](#) pour le Web (les applications Android peuvent utiliser le [plug-in Android Places](#)).

Utilisez la [bibliothèque de géocodeur Amazon Location for Maplibre](#) pour simplifier le processus d'utilisation d'Amazon Location dans les `amazon-location-for-maplibre-gl-geocoder` applications. JavaScript

• Itinéraires

Pour afficher les itinéraires sur la carte, utilisez [MapLibre les directions](#).

• Geofences et traceurs

MapLibre ne dispose pas d'outils de rendu ou d'outils spécifiques pour les géofences et le suivi, mais vous pouvez utiliser la fonctionnalité de rendu et les [plug-ins](#) pour afficher les géofences et les appareils suivis sur la carte.

Les appareils suivis peuvent utiliser le [MQTT](#) ou envoyer manuellement des mises à jour à Amazon Location Service. Les événements Geofence peuvent être traités en utilisant [AWS Lambda](#)

De nombreuses bibliothèques open source sont disponibles pour fournir des fonctionnalités supplémentaires à Amazon Location Service, par exemple [Turf](#) qui fournit des fonctionnalités d'analyse spatiale.

De nombreuses bibliothèques utilisent les données au format [GeoJSON](#) standard ouvert. Amazon Location Service fournit une bibliothèque permettant d'utiliser GeoJSON dans JavaScript les applications. Pour plus d'informations, consultez la section suivante, [SDK et bibliothèques Amazon Location](#).

Plug-in Amazon Location MapLibre Geocoder

Le plugin de MapLibre géocodeur Amazon Location est conçu pour vous permettre d'intégrer plus facilement les fonctionnalités Amazon Location dans vos JavaScript applications lorsque vous travaillez sur le rendu de cartes et le géocodage à l'aide de la [maplibre-gl-geocoder](#) bibliothèque.

Installation

Vous pouvez installer le plugin de MapLibre géocodeur Amazon Location à partir de NPM pour une utilisation avec des modules, à l'aide de cette commande :

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

Vous pouvez l'importer dans un fichier HTML pour l'utiliser directement dans le navigateur, à l'aide d'un script :

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

Utilisation avec le module

Ce code permet de configurer une JavaScript carte Maplibre GL avec les fonctionnalités de géocodage d'Amazon Location. Il utilise l'authentification via le pool d'identités Amazon Cognito pour accéder aux ressources Amazon Location. La carte est rendue avec un style et des coordonnées centrales spécifiés, et permet de rechercher des lieux sur la carte.

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

Utilisation avec un navigateur

Cet exemple utilise le client de localisation Amazon pour effectuer une demande d'authentification à l'aide d'Amazon Cognito.

Note

Certains de ces exemples utilisent le client Amazon Location. Le client Amazon Location est basé sur le [AWS SDK pour la JavaScript version 3](#) et permet de passer des appels à Amazon Location via un script référencé dans un fichier HTML.

Incluez les éléments suivants dans un fichier 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>
```

Incluez les éléments suivants dans un JavaScript fichier :

```
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());
```

Les fonctions et commandes utilisées dans le plug-in de MapLibre géocodeur Amazon Location sont répertoriées ci-dessous :

- **buildAmazonLocationMaplibreGeocoder**

Cette classe crée une instance `AmazonLocationMaplibreGeocoder` qui est le point d'entrée de tous les autres appels :

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

- **getPlacesGeocoder**

Renvoie un objet `iControl` prêt à l'emploi qui peut être ajouté directement à une carte.

```
const geocoder = getPlacesGeocoder();

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

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

SDK et bibliothèques Amazon Location

Le SDK Amazon Location est un ensemble de bibliothèques open source qui fournissent des fonctionnalités utiles pour le développement d'applications Amazon Location. Les fonctionnalités suivantes sont incluses :

- Client Amazon Location : les objets Amazon Location du AWS SDK v3 sont regroupés et emballés pour faciliter leur utilisation dans le développement Web.
- Authentification : l'utilitaire d'authentification simplifie l'authentification (à l'aide d'Amazon Cognito ou de clés API) lors de la création d'une page Web ou d'une application [JavaScriptiOS](#) ou [Android pour](#) Amazon Location Service.
- Suivi — Les SDK de suivi mobile sont disponibles pour [iOS](#) et [Android](#). Ce SDK permet aux applications mobiles d'interagir plus facilement avec Amazon Location Trackers.
- Fonctions Amazon Location GeoJSON : les [utilitaires de conversion GeoJSON facilitent la conversion](#) entre les données au [format GeoJSON standard et les formats](#) Amazon Location API.

Rubriques

- [Comment commencer à utiliser le SDK Amazon Location](#)
- [Client Amazon Location](#)
- [JavaScript Aide à l'authentification](#)
- [Aides à la conversion GeoJSON](#)
- [SDK d'authentification mobile Android](#)
- [SDK d'authentification mobile iOS](#)
- [SDK de suivi mobile Android](#)
- [SDK de suivi mobile pour iOS](#)

Comment commencer à utiliser le SDK Amazon Location

Le SDK Amazon Location est un ensemble de fonctions qui peuvent simplifier l'utilisation d'Amazon Location Service dans une application. Vous pouvez installer et importer ces fonctions dans votre JavaScript application. Les sections suivantes décrivent le client Amazon Location, ainsi que les bibliothèques d'authentification et d'assistance GeoJSON.

Client Amazon Location

Avec le AWS SDK v3, le SDK est séparé par service. Vous pouvez installer uniquement les pièces dont vous avez besoin. Par exemple, pour installer le client Amazon Location et le fournisseur d'informations d'identification pour Amazon Cognito, utilisez les commandes suivantes.

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

Afin de faciliter l'utilisation d'Amazon Location Service dans les applications d'interface JavaScript Web, AWS fournit un ensemble hébergé comprenant la bibliothèque Amazon Location et le fournisseur d'informations d'identification. Pour utiliser le client groupé, ajoutez-le à votre code HTML dans une balise de script, comme suit :

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

Note

Le package est maintenu à jour et rétrocompatible pour faciliter son utilisation. L'utilisation de cette balise de script ou de l'installation de NPM permet toujours d'obtenir la dernière version.

JavaScript Aide à l'authentification

L'assistant JavaScript d'authentification Amazon Location simplifie l'authentification lorsque vous effectuez des appels à l'API Amazon Location depuis votre JavaScript application. Cet assistant d'authentification vous aide spécifiquement lorsque vous utilisez [Amazon Cognito](#) [ou des clés d'API](#) comme méthode d'authentification. Il s'agit d'une bibliothèque open source disponible sur GitHub <https://github.com/aws-geospatial/amazon-location-utilities-auth-helper-js>.

Note

La prise en charge d'Amazon Cognito dans l'assistant d'authentification ne prend pas en charge la fonctionnalité d'identités fédérées d'Amazon Cognito.

Montage

Vous pouvez utiliser les bibliothèques avec une installation locale, si vous utilisez un système de compilation tel que Webpack, ou en incluant des JavaScript bundles prédéfinis avec des `<script>` balises dans votre code HTML.

- Utilisez la commande suivante pour installer la bibliothèque à l'aide de NPM :

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

- Utilisez la commande suivante dans votre fichier HTML pour charger le script :

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

Importer

Pour utiliser une fonction spécifique dans votre JavaScript application, vous devez importer cette fonction. Le code suivant est utilisé pour importer la fonction `withIdentityPoolId` dans votre application.

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

Fonctions d'authentification

Les aides à l'authentification Amazon Location incluent les fonctions suivantes qui renvoient un `AuthHelper` objet :

- `async withIdentityPoolId(identityPoolId: string): AuthHelper`— Cette fonction renvoie un `AuthHelper` objet, initialisé pour fonctionner avec Amazon Cognito
- `async withAPIKey(API_KEY: string): AuthHelper`— Cette fonction renvoie un `AuthHelper` objet, initialisé pour fonctionner avec les clés d'API.

L'`AuthHelper` objet fournit les fonctions suivantes :

- `AuthHelper.getMapAuthenticationOptions()`— Cette fonction de l' `AuthHelper` objet renvoie un JavaScript objet avec le `transformRequest` qui peut être utilisé avec les options cartographiques dans MapLibre JS. Fourni uniquement lors de l'initialisation avec un pool d'identités.
- `AuthHelper.getLocationClientConfig()`— Cette fonction de l' `AuthHelper` objet renvoie un JavaScript objet avec le `credentials` qui peut être utilisé pour initialiser a. `LocationClient`
- `AuthHelper.getCredentials()`— Cette fonction de l' `AuthHelper` objet renvoie les informations d'identification internes d'Amazon Cognito. Fourni uniquement lors de l'initialisation avec un pool d'identités.

Exemple : Initialisation d'un objet MapLibre cartographique avec Amazon Cognito, à l'aide d'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
});
```

Exemple : Initialisation d'un objet MapLibre cartographique avec une clé d'API (**AuthHelper** inutile dans ce cas)

```
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',
});
```

Exemple : initialisez le client Location à partir du AWS SDK pour JS, à l'aide d'Amazon Cognito et AuthHelper

Cet exemple utilise le AWS SDK pour la JavaScript version 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
}));
```

Exemple : initialisez le client Location à partir du AWS SDK pour JS, à l'aide d'une clé d'API et AuthHelper

Cet exemple utilise le AWS SDK pour la JavaScript version 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
}));
```

Aides à la conversion GeoJSON

[Les aides à la conversion Amazon Location GeoJSON](#) fournissent des outils permettant de convertir les types de données Amazon Location Service vers et depuis le format GeoJSON standard.

GeoJSON est utilisé, par exemple, pour MapLibre afficher des données géographiques sur la carte. Il s'agit d'une bibliothèque open source disponible sur GitHub <https://github.com/aws-geospatial/amazon-location-utilities-datatypes-js>.

Montage

Vous pouvez utiliser les bibliothèques avec une installation locale, comme Webpack, ou en incluant des JavaScript bundles prédéfinis avec des `<script>` balises dans votre code HTML.

- Utilisez la commande suivante pour installer la bibliothèque à l'aide de NPM.

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

- Utilisez la commande suivante dans votre fichier HTML pour charger le script :

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

Importer

Pour utiliser une fonction spécifique dans votre JavaScript application, vous devez importer cette fonction. Le code suivant est utilisé pour importer la fonction `placeToFeatureCollection` dans votre application.

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

Fonctions de conversion GeoJSON

Les aides à la conversion Amazon Location GeoJSON incluent les fonctions suivantes :

- `placeToFeatureCollection(place: GetPlaceResponse | searchPlaceIndexForPositionResponse | searchPlaceIndexForTextResponse, keepNull: boolean): FeatureCollection`— Cette fonction convertit les réponses des fonctions de recherche de lieux en un fichier GeoJSON `FeatureCollection` avec une ou plusieurs entités ponctuelles.
- `devicePositionToFeatureCollection(devicePositions: GetDevicePositionResponse | BatchGetDevicePositionResponse | GetDevicePositionHistoryResponse | ListDevicePositionsResponse, keepNull: boolean)`— Cette fonction convertit les réponses des fonctions de position du dispositif de suivi en un fichier GeoJSON `FeatureCollection` avec une ou plusieurs entités ponctuelles.
- `routeToFeatureCollection(legs: CalculateRouteResponse): FeatureCollection`— Cette fonction convertit les réponses de la fonction de calcul d'itinéraire en un fichier GeoJSON `FeatureCollection` avec une seule `MultiStringLine` entité. Chaque étape de l'itinéraire est représentée par une `LineString` entrée dans le `MultiStringLine`.

- `geofenceToFeatureCollection(geofences: GetGeofenceResponse | PutGeofenceRequest | BatchPutGeofenceRequest | ListGeofencesResponse): FeatureCollection`— Cette fonction convertit la demande ou la réponse des fonctions de géofence en un fichier `FeatureCollection` GeoJSON avec des entités polygonales. Il peut convertir les géofences à la fois dans la réponse et dans la demande, ce qui vous permet d'afficher les géofences sur une carte avant de les télécharger avec `PutGeofence` ou `BatchPutGeofence`.

Cette fonction convertira une géofence circulaire en une entité dotée d'un polygone approximatif, mais comportera également des propriétés « centre » et « rayon » permettant de recréer la géofence circulaire, si nécessaire (voir la fonction suivante).

- `featureCollectionToGeofences(featureCollection: FeatureCollection): BatchPutGeofenceRequestEntry[]`— Cette fonction convertit un fichier GeoJSON `FeatureCollection` contenant des entités polygonales en un tableau d'objets `BatchPutGeofenceRequestEntry`, de sorte que le résultat puisse être utilisé pour créer une demande à `BatchPutGeofence`.

Si une entité `FeatureCollection` présente les propriétés « centre » et « rayon », elle sera convertie en une entrée de demande de géofence circulaire, sans tenir compte de la géométrie du polygone.

Exemple : conversion des résultats de recherche en couche de points dans MapLibre

Cet exemple utilise le AWS SDK pour la JavaScript version 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 d'authentification mobile Android

Ces utilitaires vous aident à vous authentifier lorsque vous passez des appels à l'API Amazon Location Service depuis vos applications Android. Cela est particulièrement utile lorsque vous utilisez [Amazon Cognito](#) ou des [clés d'API](#) comme méthode d'authentification.

Le SDK d'authentification mobile Android est disponible sur github : [Amazon Location Service Mobile Authentication SDK for Android](#). En outre, le SDK d'authentification mobile et le AWS SDK sont disponibles dans le référentiel [AWS Maven](#).

Installation

Pour utiliser le SDK d'authentification mobile, ajoutez les instructions d'importation suivantes à votre `build.gradle` fichier dans Android Studio.

```
implementation("software.amazon.location:auth:0.0.1")
implementation("com.amazonaws:aws-android-sdk-location:2.72.0")
```

Fonctions d'authentification

Le SDK d'assistance à l'authentification possède les fonctions suivantes :

- `authHelper.authenticateWithApiKey("My-Amazon-Location-API-Key")`: `LocationCredentialsProvider`: Cette fonction renvoie une valeur `LocationCredentialsProvider` initialisée pour fonctionner avec une clé d'API.
- `authHelper.authenticateWithCognitoIdentityPool("My-Cognito-Identity-Pool-Id")`: `LocationCredentialsProvider`: Cette fonction renvoie une valeur `LocationCredentialsProvider` initialisée pour fonctionner avec un pool d'identités Amazon Cognito.

Utilisation

Pour utiliser le SDK dans votre code, importez les classes suivantes :

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

Deux options s'offrent à vous lorsque vous créez l'assistant d'authentification et les instances du fournisseur de clients de localisation. Vous pouvez créer une instance à l'aide des [clés d'API Amazon Location](#) ou d'[Amazon Cognito](#).

- Pour créer une instance d'assistance à l'authentification à l'aide d'une clé d'API Amazon Location, déclarez la classe d'assistance comme suit :

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

- Pour créer une instance d'assistance à l'authentification à l'aide d'Amazon Cognito, déclarez la classe d'assistance comme suit :

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

Vous pouvez créer une instance client Amazon Location à l'aide du fournisseur d'informations de localisation et passer des appels au service Amazon Location. L'exemple suivant recherche des lieux proches d'une latitude et d'une longitude spécifiées.

```
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 d'authentification mobile iOS

Ces utilitaires vous aident à vous authentifier lorsque vous passez des appels à l'API Amazon Location Service depuis vos applications iOS. Cela est particulièrement utile lorsque vous utilisez [Amazon Cognito](#) ou des [clés d'API](#) comme méthode d'authentification.

Le SDK d'authentification mobile iOS est disponible sur github : [Amazon Location Service Mobile Authentication SDK for iOS](#).

Installation

Installez le SDK dans un projet Xcode :

1. Accédez à Fichier, puis sélectionnez Ajouter des dépendances de package dans votre projet XCode.
2. Tapez l'URL du package : <https://github.com/aws-geospatial/amazon-location-mobile-auth-sdk-ios/> dans la barre de recherche et appuyez sur la touche Entrée.
3. Sélectionnez le amazon-location-mobile-auth-sdk-ios package et appuyez sur Ajouter un package.
4. Sélectionnez le produit AmazonLocationiOSErrorSDK du package et appuyez sur Ajouter un package.

Fonctions d'authentification

Le SDK d'assistance à l'authentification possède les fonctions suivantes :

- `authHelper.authenticateWithApiKey("My-Amazon-Location-API-Key")`: `LocationCredentialsProvider`: Cette fonction renvoie une valeur `LocationCredentialsProvider` initialisée pour fonctionner avec une clé d'API.
- `authHelper.authenticateWithCognitoIdentityPool("My-Cognito-Identity-Pool-Id")`: `LocationCredentialsProvider`: Cette fonction renvoie une valeur `LocationCredentialsProvider` initialisée pour fonctionner avec un pool d'identités Amazon Cognito.

Utilisation

Pour utiliser le SDK d'authentification mobile, ajoutez les instructions suivantes à votre activité :

```
import AmazonLocationiOSErrorSDK
import AWSLocationXCF
```

Deux options s'offrent à vous lorsque vous créez l'assistant d'authentification et les instances du fournisseur de clients de localisation. Vous pouvez créer une instance à l'aide des [clés d'API Amazon Location](#) ou d'[Amazon Cognito](#).

- Pour créer une instance d'assistance à l'authentification à l'aide d'une clé d'API Amazon Location, déclarez la classe d'assistance comme suit :

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

- Pour créer une instance d'assistance à l'authentification à l'aide d'Amazon Cognito, déclarez la classe d'assistance comme suit :

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

Vous pouvez créer une instance client Amazon Location à l'aide du fournisseur d'informations de localisation et passer des appels au service Amazon Location. L'exemple suivant recherche des lieux proches d'une latitude et d'une longitude spécifiées.

```
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 suivi mobile Android

Le SDK de suivi mobile Amazon Location fournit des utilitaires qui permettent d'authentifier, de capturer les positions des appareils et d'envoyer des mises à jour de position à Amazon Location Trackers en toute simplicité. Le SDK prend en charge le filtrage local des mises à jour de localisation avec des intervalles de mise à jour configurables. Cela permet de réduire les coûts de données et d'optimiser la connectivité intermittente pour vos applications Android.

Le SDK de suivi Android est disponible sur GitHub : [Amazon Location Mobile Tracking SDK](#) [for](#) Android. En outre, le SDK d'authentification mobile et le AWS SDK sont disponibles dans le référentiel [AWS Maven](#). Le SDK de suivi Android est conçu pour fonctionner avec le AWS SDK général.

Cette section couvre les sujets suivants relatifs au SDK Android de suivi mobile Amazon Location :

Rubriques

- [Installation](#)
- [Utilisation](#)
- [Filtres](#)
- [Fonctions de suivi du SDK Android Mobile](#)
- [Exemples](#)

Installation

Pour installer le SDK, ajoutez les lignes suivantes à la section des dépendances de votre fichier build.gradle dans 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")
```

Utilisation

Cette procédure explique comment utiliser le SDK pour authentifier et créer l'LocationTracker objet :

Note

Cette procédure suppose que vous avez importé la bibliothèque mentionnée dans la [Installation](#) section.

1. Importez les classes suivantes dans votre code :

```
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. Créez ensuite un `AuthHelper`, puisque le `LocationCredentialsProvider` paramètre est obligatoire pour créer un `LocationTracker` objet :

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

3. Maintenant, utilisez le `LocationCredentialsProvider` et `LocationTrackerConfig` pour créer un `LocationTracker` objet :

```
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,
)
```

Filtres

Le SDK Android de suivi mobile Amazon Location intègre trois filtres de localisation.

- `TimeLocationFilter`: filtre l'emplacement actuel à télécharger en fonction d'un intervalle de temps défini.
- `DistanceLocationFilter`: filtre les mises à jour de localisation en fonction d'un seuil de distance spécifié.
- `AccuracyLocationFilter`: filtre les mises à jour de localisation en comparant la distance parcourue depuis la dernière mise à jour avec la précision de la position actuelle.

Cet exemple ajoute des filtres `LocationTracker` au moment de la création :

```
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,
)
```

Cet exemple active et désactive le filtre lors de l'exécution avec LocationTracker :

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

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

Fonctions de suivi du SDK Android Mobile

Le SDK de suivi mobile Amazon Location pour Android inclut les fonctions suivantes :

- Classe : LocationTracker

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

Il s'agit d'une fonction d'initialisation permettant de créer un LocationTracker objet. Il nécessite des instances de LocationCredentialsProvider, trackerName et éventuellement une instance de LocationTrackingConfig. Si la configuration n'est pas fournie, elle sera initialisée avec les valeurs par défaut.

- Classe : LocationTracker

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

Lance le processus d'accès à la position de l'utilisateur et de son envoi à un outil de suivi de localisation Amazon.

- Classe : `LocationTracker`

```
isTrackingInForeground()
```

Vérifie si le suivi de localisation est actuellement en cours.

- Classe : `LocationTracker`

```
stop()
```

Arrête le processus de suivi de la position de l'utilisateur.

- Classe : `LocationTracker`

```
startTracking()
```

Lance le processus d'accès à la position de l'utilisateur et d'envoi de celle-ci au AWS tracker.

- Classe : `LocationTracker`

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

Lance le processus d'accès à la position de l'utilisateur et de son envoi au AWS tracker alors que l'application est en arrière-plan. `BackgroundTrackingMode` propose les options suivantes :

- `ACTIVE_TRACKING`: Cette option suit activement les mises à jour de localisation d'un utilisateur.
 - `BATTERY_SAVER_TRACKING`: Cette option suit les mises à jour de localisation de l'utilisateur toutes les 15 minutes.
- Classe : `LocationTracker`

```
stopBackgroundService()
```

Arrête le processus d'accès à la position de l'utilisateur et d'envoi de celle-ci au AWS tracker alors que l'application est en arrière-plan.

- Classe : `LocationTracker`

```
getTrackerDeviceLocation()
```

Récupère l'emplacement de l'appareil auprès des services de localisation Amazon.

- Classe : `LocationTracker`

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

Récupère la position actuelle de l'appareil auprès du client du fournisseur de localisation fusionné et la télécharge sur Amazon Location Tracker.

- Classe : `LocationTracker`

```
uploadLocationUpdates(locationTrackingCallback:
LocationTrackingCallback?)
```

Télécharge la position de l'appareil vers les services de localisation Amazon après avoir filtré en fonction des filtres de localisation configurés.

- Classe : `LocationTracker`

```
enableFilter(filter: LocationFilter)
```

Active un filtre de localisation spécifique.

- Classe : `LocationTracker`

```
checkFilterIsExistsAndUpdateValue(filter: LocationFilter)
```

Désactivez un filtre de localisation particulier.

- Classe : `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())
```

Cela initialise le `LocationTrackerConfig` avec des valeurs de paramètres définies par l'utilisateur. Si aucune valeur de paramètre n'est fournie, elle sera définie sur une valeur par défaut.

- Classe : `LocationFilter`

```
shouldUpload(currentLocation: LocationEntry, previousLocation: LocationEntry?): Boolean
```

`LocationFilter` s'agit d'un protocole que les utilisateurs peuvent implémenter pour leur mise en œuvre de filtres personnalisés. Vous devez implémenter la `shouldUpload` fonction pour comparer l'emplacement précédent et actuel et indiquer si l'emplacement actuel doit être téléchargé.

Exemples

L'exemple de code suivant illustre les fonctionnalités du SDK de suivi mobile.

Cet exemple utilise le `LocationTracker` pour démarrer et arrêter le suivi en arrière-plan :

```
// 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 suivi mobile pour iOS

Le SDK de suivi mobile Amazon Location fournit des utilitaires qui permettent d'authentifier, de capturer les positions des appareils et d'envoyer des mises à jour de position à Amazon Location Trackers en toute simplicité. Le SDK prend en charge le filtrage local des mises à jour de localisation avec des intervalles de mise à jour configurables. Cela permet de réduire les coûts de données et d'optimiser la connectivité intermittente pour vos applications iOS.

Le SDK de suivi iOS est disponible sur GitHub : [Amazon Location Mobile Tracking SDK for iOS](#).

Cette section aborde les sujets suivants relatifs au SDK iOS de suivi mobile Amazon Location :

Rubriques

- [Installation](#)
- [Utilisation](#)
- [Filtres](#)
- [Fonctions de suivi du SDK iOS Mobile](#)
- [Exemples](#)

Installation

Pour installer le SDK de suivi mobile pour iOS, procédez comme suit :

1. Dans votre projet Xcode, accédez à Fichier et sélectionnez Ajouter des dépendances de package.
2. Tapez l'URL suivante : <https://github.com/aws-geospatial/amazon-location-mobile-tracking-sdk-ios/> dans la barre de recherche et appuyez sur la touche Entrée.
3. Sélectionnez le amazon-location-mobile-tracking-sdk-ios package et cliquez sur Ajouter un package.
4. Sélectionnez le produit AmazonLocationiOSTrackingSDK du package et cliquez sur Ajouter un package.

Utilisation

La procédure suivante explique comment créer un assistant d'authentification à l'aide des informations d'identification de Cognito.

1. Après avoir installé la bibliothèque, vous devez ajouter l'une des descriptions ou les deux dans votre `info.plist` fichier :

```
Privacy - Location When In Use Usage Description
Privacy - Location Always and When In Use Usage Description
```

2. Importez ensuite le AuthHelper dans votre classe :

```
import AmazonLocationiOSAuthSDKimport AmazonLocationiOSTrackingSDK
```

3. Vous allez ensuite créer un `AuthHelper` objet et l'utiliser avec le AWS SDK, en créant un assistant d'authentification à l'aide des informations d'identification d'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)
```

Filtres

Le SDK iOS de suivi mobile Amazon Location intègre trois filtres de localisation.

- `TimeLocationFilter`: filtre l'emplacement actuel à télécharger en fonction d'un intervalle de temps défini.
- `DistanceLocationFilter`: filtre les mises à jour de localisation en fonction d'un seuil de distance spécifié.
- `AccuracyLocationFilter`: filtre les mises à jour de localisation en comparant la distance parcourue depuis la dernière mise à jour avec la précision de la position actuelle.

Cet exemple ajoute des filtres `LocationTracker` au moment de la création :

```
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,
)
```

Cet exemple active et désactive le filtre lors de l'exécution avec `LocationTracker` :

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

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

Fonctions de suivi du SDK iOS Mobile

Le SDK de suivi mobile Amazon Location pour iOS inclut les fonctions suivantes :

- Classe : `LocationTracker`

```
init(provider: LocationCredentialsProvider, trackerName: String, config:
LocationTrackerConfig? = nil)
```

Il s'agit d'une fonction d'initialisation permettant de créer un `LocationTracker` objet. Il nécessite des instances de `LocationCredentialsProvider`, `trackerName` et éventuellement une instance de `LocationTrackingConfig`. Si la configuration n'est pas fournie, elle sera initialisée avec les valeurs par défaut.

- Classe : `LocationTracker`

```
setTrackerConfig(config: LocationTrackerConfig)
```

Cela permet à la configuration de Tracker de prendre effet à tout moment après l'initialisation du suivi de localisation

- Classe : `LocationTracker`

```
getTrackerConfig()
```

Cela permet d'obtenir la configuration de suivi de localisation à utiliser ou à modifier dans votre application.

Retours : `LocationTrackerConfig`

- Classe : `LocationTracker`

```
getDeviceId()
```

Obtient l'identifiant de l'appareil généré par le traqueur de localisation.

Retours : `String?`

- Classe : `LocationTracker`

```
startTracking()
```

Lance le processus d'accès à la position de l'utilisateur et d'envoi de celle-ci au AWS tracker.

- Classe : `LocationTracker`

```
resumeTracking()
```

Reprend le processus d'accès à la position de l'utilisateur et d'envoi de celle-ci au AWS traceur.

- Classe : `LocationTracker`

```
stopTracking()
```

Arrête le processus de suivi de la position de l'utilisateur.

- Classe : `LocationTracker`

```
startBackgroundTracking(mode: BackgroundTrackingMode)
```

Lance le processus d'accès à la position de l'utilisateur et de son envoi au AWS tracker alors que l'application est en arrière-plan. `BackgroundTrackingMode` propose les options suivantes :

- `Active` : Cette option n'interrompt pas automatiquement les mises à jour de localisation.
- `BatterySaving` : Cette option suspend automatiquement les mises à jour de localisation

- `None` : Cette option désactive globalement les mises à jour de localisation en arrière-plan

- Classe : `LocationTracker`

```
resumeBackgroundTracking(mode: BackgroundTrackingMode)
```

Reprend le processus d'accès à la position de l'utilisateur et d'envoi de celle-ci au AWS tracker alors que l'application est en arrière-plan.

- Classe : `LocationTracker`

```
stopBackgroundTracking()
```

Arrête le processus d'accès à la position de l'utilisateur et d'envoi de celle-ci au AWS tracker alors que l'application est en arrière-plan.

- Classe : `LocationTracker`

```
getTrackerDeviceLocation(nextToken: String?, startTime: Date? = nil, endTime: Date? = nil, completion: @escaping (Result<GetLocationResponse, Error>)
```

Récupère les emplacements de suivi téléchargés pour l'appareil de l'utilisateur entre la date et l'heure de début et de fin.

Retours : `Void`

- Classe : `LocationTrackerConfig`

```
init()
```

Cela initialise le `LocationTrackerConfig` avec les valeurs par défaut.

- Classe : `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)
```

Cela initialise le `LocationTrackerConfig` avec des valeurs de paramètres définies par l'utilisateur. Si aucune valeur de paramètre n'est fournie, elle sera définie sur une valeur par défaut.

- Classe : `LocationFilter`

```
shouldUpload(currentLocation: LocationEntity, previousLocation: LocationEntity?, trackerConfig: LocationTrackerConfig)
```

`LocationFilter` Il s'agit d'un protocole que les utilisateurs peuvent implémenter pour leur mise en œuvre de filtres personnalisés. Un utilisateur devra implémenter une `shouldUpload` fonction pour comparer l'emplacement précédent et actuel et revenir si l'emplacement actuel doit être téléchargé.

Exemples

Cette section présente des exemples d'utilisation du SDK de suivi mobile Amazon Location pour iOS.

Note

Assurez-vous que les autorisations nécessaires sont définies dans le `info.plist` fichier. Il s'agit des mêmes autorisations que celles répertoriées dans la [Utilisation](#) section.

L'exemple suivant illustre les fonctionnalités permettant de suivre la position des appareils et de récupérer les positions suivies :

```
Privacy - Location When In Use Usage Description
Privacy - Location Always and When In Use Usage Description
```

Commencez à suivre la position :

```
do {
    try locationTracker.startTracking()
    }
catch TrackingLocationError.permissionDenied {
    // Handle permissionDenied by showing the alert message or opening the app
    settings
    }
}
```

Reprenez le suivi de la localisation :

```
do {
    try locationTracker.resumeTracking()
    }
```

```
    }  
    catch TrackingLocationError.permissionDenied {  
        // Handle permissionDenied by showing the alert message or opening the app settings  
    }  
}
```

Arrêtez de suivre la position :

```
locationTracker.stopTracking()
```

Démarrez le suivi en arrière-plan :

```
do {  
  
    locationTracker.startBackgroundTracking(mode: .Active) // .Active, .BatterySaving, .None  
    }  
catch TrackingLocationError.permissionDenied {  
    // Handle permissionDenied by showing the alert message or opening the app settings  
}
```

Suivi des antécédents du CV :

```
do {  
    locationTracker.resumeBackgroundTracking(mode: .Active)  
    }  
catch TrackingLocationError.permissionDenied {  
    // Handle permissionDenied by showing the alert message or opening the app settings  
}
```

Pour arrêter le suivi en arrière-plan :

```
locationTracker.stopBackgroundTracking()
```

Récupérez les positions suivies de l'appareil à partir du tracker :

```
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 localisation Amazon

Amazon Location Service fournit des opérations d'API pour accéder par programmation à la fonctionnalité de localisation. Cela inclut les API pour les cartes, les lieux, les itinéraires, les traceurs, les géofences et le balisage de vos ressources. Pour plus d'informations sur les actions d'API disponibles, consultez la [référence de l'API Amazon Location Service](#).

Vous trouverez des exemples dans le [Exemples de code](#) chapitre de ce guide.

Utilisation d'Amazon Location avec un AWS SDK

AWS des kits de développement logiciel (SDK) sont disponibles pour de nombreux langages de programmation populaires. Chaque SDK fournit une API, des exemples de code et une documentation qui permettent aux développeurs de créer plus facilement AWS des applications dans leur langue préférée.

Pour plus d'informations sur les SDK disponibles pour une utilisation avec Amazon Location Service par langue, consultez [SDK par langue](#) ce guide.

Versions du SDK

Nous vous recommandons d'utiliser la version la plus récente du AWS SDK, ainsi que tout autre SDK, que vous utilisez dans vos projets, et de maintenir les SDK à jour. Le AWS SDK vous fournit les fonctionnalités les plus récentes, ainsi que des mises à jour de sécurité.

Pour trouver la dernière version du AWS SDK pour JavaScript, par exemple, consultez la rubrique relative à [l'installation du navigateur](#) dans le AWS SDK pour obtenir JavaScript de la documentation.

Mises à jour des messages d'erreur de l'API Amazon Location

À compter du 1er août 2023, l'équipe Amazon Location modifiera les messages d'erreur de l'API, comme décrit dans les tableaux suivants. Les codes d'erreur ne seront pas modifiés. Si vos applications dépendent de chaînes de messages d'erreur exactes, vous devez les mettre à jour avec les nouvelles chaînes. Pour obtenir de l'aide en cas de questions ou de problèmes, contactez AWS Support.

Rubriques

- [Lieux](#)
- [Mappages](#)
- [Traceurs](#)
- [Acheminements](#)
- [Metadonnées](#)
- [Géofences](#)

Lieux

Lieux

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
500	InternalServerErrorException	Internal Server Exception	Internal server error. Try again later.
404	ResourceNotFoundException	resource <PlaceIndexName> not found, reason: <Reason> Resource '<PlaceIndexName>' not found	Place index not found: <PlaceIndexName>.

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
		placeIdx<PlaceIndexName> not found, reason: <Reason> no place index with name '%s' found	
404	ResourceNotFoundException	place not found	Place not found: <PlaceId>.
400	ValidationException	PlaceIndex <PlaceIndexName> cannot be used for SearchPlaceIndexForSuggestions because it has IntendedUse <IntendedUse>	A place index with 'IntendedUse' set to Storage does not support 'SearchPlaceIndexForSuggestions' operation.
400	ValidationException	only one of 'BiasPosition' or 'FilterBBox' may be set	Only one of 'BiasPosition' or 'FilterBBox' may be set.
400	ValidationException	BiasPosition must have exactly 2 entries	'BiasPosition' must have exactly 2 entries.
400	ValidationException	BiasPosition[0] must be between -180 and 180	'BiasPosition[0]' must be between -180 and 180.
400	ValidationException	BiasPosition[1] must be between -90 and 90	'BiasPosition[1]' must be between -90 and 90.
400	ValidationException	FilterBBox must have exactly 4 entries	'FilterBBox' must have exactly 4 entries.

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
400	ValidationException	FilterBBox[0] must be between -180 and 180	'FilterBBox[0]' must be between -180 and 180.
400	ValidationException	FilterBBox[1] must be between -90 and 90	'FilterBBox[1]' must be between -90 and 90.
400	ValidationException	FilterBBox[2] must be between -180 and 180	'FilterBBox[2]' must be between -180 and 180.
400	ValidationException	FilterBBox[3] must be between -90 and 90	'FilterBBox[3]' must be between -90 and 90.
400	ValidationException	FilterBBox must have more southwesterly point before more northeasterly point	'FilterBBox' must have more southwesterly position before more northeasterly position.
400	ValidationException	Position must have exactly 2 entries	'Position' must have exactly 2 entries.
400	ValidationException	Position[0] must be between -180 and 180	'Position[0]' must be between -180 and 180.
400	ValidationException	Position[1] must be between -90 and 90	'Position[1]' must be between -90 and 90.

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
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.
400	ValidationException	Grab is only supported in the ap-southeast-1 region	'DataSource' Grab must only be used in following regions: ap-southeast-1.

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
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>.

Mappages

Mappages

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
500	InternalServerError	Internal Server Exception unable to find style template Error fetching style	Internal server error. Try again later.

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
		was not able to serialize the map style file	
404	ResourceNotFoundException	Map not found	Map not found: <MapName>.
404	ResourceNotFoundException	Sprites are not supported for this resource	Sprite not found: <SpriteName>.
400	ValidationException	Resource name should be set	'MapName' is a required field.
400	ValidationException	Must provide a valid number for start and end of Range	Font Unicode range start and end numbers must both be provided.
400	ValidationException	Start of range is an invalid number: <StartValue>	Start of font Unicode range must be a valid number.
400	ValidationException	End of range is an invalid number: <StartValue>	End of font Unicode range must be a valid number.
400	ValidationException	End of range must be exactly 255 higher from start of range, difference found: <Difference>	The difference between the start and end of the font Unicode range must be exactly 255. Difference found: <Difference>.

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
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 .
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 .

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
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 .
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 .

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
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'.
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.

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
400	ValidationException	'Z' must not be less than minimum zoom of style '<Style>' (<MinimumValue>)	'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>' (<MaximumValue>)	'Z' must not be greater than maximum zoom of style <Style> (<MaximumValue>).
400	ValidationException	'Z' value not supported	'Z' must be between 0 and 63.
400	ValidationException	tile resolution part of 'Y' must be omitted because '<Style>' is a vector style	Tile resolution part of 'Y' must be omitted for style <Style>.
400	ValidationException	tile resolution part of 'Y' must be at least 1	Tile resolution part of 'Y' must be at least 1.
400	ValidationException	tile resolution part of 'Y' must not be greater than max resolution of style '<Style>' (<MaximumResolution>)	Tile resolution part of 'Y' must not be greater than maximum resolution of style <Style> (max <MaxResolution>).

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
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.
400	ValidationException	Expired token	'NextToken' must not be expired.

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
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>.

Traceurs

Traceurs

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
500	InternalServerErrorException	Internal Server Exception	Internal server error. Try again later.

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
		<p>internal server error</p> <p>unable to retrieve point from the storage</p> <p>unable to verify tracker</p> <p>Error processing List request</p>	
404	ResourceNotFoundException	<p>tracker not found: <TrackerName></p> <p>Tracker with name <TrackerName> was not found</p>	Tracker not found: <TrackerName>.
404	ResourceNotFoundException	<p>association not found: TrackerName <TrackerName>; and ConsumerArn <ConsumerArn ></p>	Association between tracker <TrackerName> and consumer <ConsumerArn> is not found.
400	ValidationException	'ConsumerArn' must refer to a geofence collection resource	'ConsumerArn' must refer to a geofence collection resource.
400	ValidationException	'ConsumerArn' must refer to a resource in the same region as the tracker it is associated to	'ConsumerArn' must refer to a resource in the same region as the tracker it is associated with.

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
400	ValidationException	'ConsumerArn' must refer to a resource in the same AWS account as the tracker 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
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
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 .
400	ValidationException	key not found	Invalid KMS key. '<KmsKeyId>' <KmsKeyIdValue> not found.

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
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/ .
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/ .

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
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>.

Acheminements

Acheminements

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
500	InternalServerErrorException	Internal Server Exception	Internal server error. Try again later.
404	ResourceNotFoundException	Resource not found	Route calculator not found: <RouteCalculatorName>.
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.
400	ValidationException	'DataSource' must be one of: Here, Esri, Grab	'DataSource' must be one of Esri, Grab, Here.

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
400	ValidationException	<PricingPlan> pricing plan is not supported	'PricingPlan' must be set to RequestBasedUsage
400	ValidationException	unsupported price plan '<PricingPlan>'	'PricingPlan' must be set to RequestBasedUsage
400	ValidationException	Grab is only supported in the ap-southeast-1 region	'DataSource' <DataSourceName> must only be used in following regions: ap-southeast-1.
400	ValidationException	PricingPlan must be 'RequestBasedUsage'	'PricingPlan' must be set to RequestBasedUsage.
400	ValidationException	'DeparturePositions[0][0]' must be between -180 and 180	'DeparturePositions[0][0]' must be between -180 and 180.
400	ValidationException	'DeparturePositions[0][1]' must be between -90 and 90	'DeparturePositions[0][1]' must be between -90 and 90.
400	ValidationException	'DestinationPositions[0][0]' must be between -180 and 180	'DestinationPositions[0][0]' must be between -180 and 180.
400	ValidationException	'DestinationPositions[0][1]' must be between -90 and 90.	'DestinationPositions[0][1]' must be between -90 and 90

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
400	ValidationException	'DepartNow' may not be true if 'DepartureTime' is set	Only one of 'DepartNow' or 'DepartureTime' may be set.
400	ValidationException	'<TravelModeOption>' may not be set when 'TravelMode' has value <TravelModeOption>	'<TravelModeOption>' must not be set when 'TravelMode' has value <TravelModeOption>.
400	ValidationException	'CarModeOptions' may not be set when 'TravelMode' has value Walking	'CarModeOptions' must not be set when 'TravelMode' has value Walking.
400	ValidationException	'TruckModeOptions' may not be set when 'TravelMode' has value Walking	'TruckModeOptions' must not be set when 'TravelMode' has value Walking.
400	ValidationException	'TruckModeOptions' may not be set when 'TravelMode' has value Car	'TruckModeOptions' must not be set when 'TravelMode' has value Car.
400	ValidationException	'CarModeOptions' may not be set when 'TravelMode' has value Truck	'CarModeOptions' must not be set when 'TravelMode' has value Truck.

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
400	ValidationException	At least one of [Height, Length, Width] must be set in 'TruckModeOptions.Dimensions'	At least one of the following attribute must be set in TruckModeOptions.Dimensions: Height, Length, Width.
400	ValidationException	At least one of [Total] must be set in 'TruckModeOptions.Weight'	At least one of the following attribute must be set in TruckModeOptions.Weight: Total.
400	ValidationException	'DeparturePositions' count must be 10 or less with DataSource set to Esri	'DeparturePositions' must have length at most 10 for 'DataSource' Esri.
400	ValidationException	'DestinationPositions' count must be 10 or less with DataSource set to Esri	'DestinationPositions' must have length at most 10 for 'DataSource' Esri.
400	ValidationException	'DeparturePositions[0]' is more than 40km away from 'DestinationPositions[0]'	'DeparturePositions[0]' must not be more than 40 km away from 'DestinationPositions[0]'.
400	ValidationException	'DeparturePositions[0]' is more than 400km away from 'DestinationPositions[0]'	'DeparturePositions[0]' must not be more than 400 km away from 'DestinationPositions[0]'.

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
400	ValidationException	DeparturePositions [0] is contained within an unsupported region. Korea is not supported for CalculateRouteMatrix with the provider Esri.	DeparturePositions [0] is located in Korea, which is not supported when using CalculateRouteMatrix with data provider Esri.
400	ValidationException	'<HereTruckDimension>' must be between <Min> and <Max> <Unit>	'HereTruckDimension' must be between <Min> and <Max> <Unit>.
400	ValidationException	'WaypointPositions[0][0]' must be between -180 and 180	'WaypointPositions[0][0]' must be between -180 and 180.
400	ValidationException	'WaypointPositions[0][1]' must be between -90 and 90	'WaypointPositions[0][1]' must be between -90 and 90.
400	ValidationException	'WaypointPositions[1][0]' must be between -180 and 180	'WaypointPositions[1][0]' must be between -180 and 180.
400	ValidationException	'WaypointPositions[1][1]' must be between -90 and 90	'WaypointPositions[1][1]' must be between -90 and 90.
400	ValidationException	No road segment could be matched for one or more coordinates within a radius (1km)	One or more provided positions are more than 1 km from the nearest road segment.

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
400	ValidationException	Some positions in the request are unreachable	Some positions in the request are unreachable.
400	ValidationException	Total distance between all waypoints must be not be greater than 40km for DataSource Esri when using TravelMode Walking	Total distance between all route positions must not be greater than 40 km for 'DataSource' Esri and 'TravelMode' Walking.
400	ValidationException	Total distance between all waypoints must be not be greater than 400km for DataSource Esri	Total distance between all route positions must not be greater than 400 km for 'DataSource' Esri.
400	ValidationException	Following positions in the request are unreachable: <UnreachablePositions>	The following positions are unreachable: <UnreachablePositions>.
400	ValidationException	'DepartureTime' contains a badly-formatted timestamp	'DepartureTime' must follow the format YYYY-MM-DDThh:mm:ss.sssZ.
400	ValidationException	'TravelMode' <TravelMode> is not supported by <DataProvider>	'TravelMode' <TravelMode> not supported by data provider <DataProvider>.

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
400	ValidationException	'DeparturePositions' must be set	'DeparturePositions' must not be empty.
400	ValidationException	'DestinationPositions' must be set	'DestinationPositions' must not be empty.
400	ValidationException	Some inputs in the request are invalid	Some inputs in the request are invalid.
400	ValidationException	No route found between position <FirstPosition> and position <SecondPosition>	No route found between position <FirstPosition> and position <SecondPosition>.
400	ValidationException	No route found	No route found. For more information, see https://developer.amazon.com/documentation/routing-api/dev_guide/topics/no_route.html .
400	ValidationException	No route found	No route found.

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
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>.

Metadonnées

Metadonnées

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
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>.

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
400	ValidationException	Either ExpireTime or NoExpiry must be provided	At least one of the following fields must be set: 'ExpireTime', 'NoExpiry'.
400	ValidationException	NoExpiry cannot be set to false if no ExpireTime is provided	'ExpireTime' must be set when 'NoExpiry' has value false.
400	ValidationException	ExpireTime cannot be set if NoExpiry is true	'ExpireTime' must not be set when 'NoExpiry' has value true.
400	ValidationException	Expire time '<ExpireTimeValue>' is not a valid time format	'ExpireTime' must follow the format YYYY-MM-DDThh:mm:ss.sssZ.
400	ValidationException	Expire time '<ExpireTimeValue>' cannot be in the past when creating a key	'ExpireTime' must not be in the past.
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
400	ValidationException	The API Key %s has been recently used and the requested update may impact current usage. Specify ForceUpdate=true to update the API Key configuration.	This update may cause some users to lose API access. Because this API Key has been used in the last 7 days, you must set 'ForceUpdate' to true to confirm this change.
400	ValidationException	Expire time '<ExpireTimeValue>' must not be more than 1 minute in the past	'ExpireTime' must not be more than 1 minute in the past.
400	ValidationException	Description, ExpireTime, NoExpiry and Restrictions can't all be empty	At least one of the following fields must be set: 'Description', 'ExpireTime', 'NoExpiry', 'Restrictions'.
400	ValidationException	API Key expired	'ApiKeyId' must not be expired.
409	ConflictException	API key named <APIKeyName> already exists	Api key already exists: <APIKeyName>.

Géofences

Géofences

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
500	InternalServerErrorException	<p>internal server error</p> <p>Internal server error</p> <p>Unsupported geofence geometry encountered</p> <p>geometry marshal error</p> <p>geometry load error</p> <p>unable to get geofence collection</p> <p>unable to delete geofences</p> <p>unable to retrieve geofence</p> <p>Error processing List request</p>	<p>Internal server error. Try again later.</p>
404	ResourceNotFoundException	<p>collection not found: <GeofenceCollectionName></p> <p><GeofenceCollectionName> geofence collection not found</p>	<p>Geofence Collection not found: <GeofenceCollectionName>.</p>

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
		Resource not found error no geofence with given name found	
400	ValidationException	unsupported price plan '<PricingPlan>'	'PricingPlan' must be set to RequestBasedUsage.
400	ValidationException	KMS key must be a symmetric CMK. Invalid usage type: <UsageType>	KMS key must be a symmetric Customer Master Key (CMK). Invalid usage type <UsageType>. For how to create a symmetric CMK, refer to https://docs.aws.amazon.com/kms/latest/developerguide/create-keys.html#create-symmetric-cmk .
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.
400	ValidationException	PricingPlanDataSource cannot be updated without updating PricingPlan	'PricingPlan' must be provided to update 'PricingPlanDataSource'.

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
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.
400	ValidationException	access denied	Symmetric Customer Master Key (CMK) must allow Amazon Location to create grants to its KMS key.

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
400	ValidationException	duplicate geofence ID in batch	'GeofenceId' <DuplicatedGeofenceId> is duplicated in batch.
400	ValidationException	missing GeofenceId	'GeofenceId' must not be empty.
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.
400	ValidationException	Position[0] must be between -180 and 180	'Position[0]' must be between -180 and 180.
400	ValidationException	Position[1] must be between -90 and 90	'Position[1]' must be between -90 and 90.
400	ValidationException	radius must be less than or equal to 1000km	'Geometry.Circle.Radius' must be less than or equal to 1000km.
400	ValidationException	no geofence with given name found	Geofence not found: <CollectionName>.
400	ValidationException	Geometry must contain either a Circle or Polygon, not both	Only one of 'Circle' or 'Polygon' may be set within 'Geometry'.
400	ValidationException	Geometry must contain a Polygon or a Circle	One of 'Polygon' or 'Circle' must be set within 'Geometry'.

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
400	ValidationException	radius must be greater than 0m	'Geometry.Circle.Radius' must be greater than 0m.
400	ValidationException	empty polygon	'Geometry.Polygon' must not be empty.
400	ValidationException	empty polygon ring	'Geometry.Polygon' must not be empty.
400	ValidationException	circle can not cross antimeridian	'Geometry.Circle' must not cross antimeridian. Cut it in two such that neither part's representation crosses the antimeridian.
400	ValidationException	polygon can not cross antimeridian	'Geometry.Polygon' must not cross antimeridian. Cut it in two such that neither part's representation crosses the antimeridian.
400	ValidationException	polygon can not have interior rings (holes), remove holes	'Geometry.Polygon' must not have interior rings (holes). For more information about interior rings see https://www.rfc-editor.org/rfc/rfc7946.html#appendix-A.3 .

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
400	ValidationException	polygon ring is not closed	'Geometry.Polygon' contains an open ring. Close the ring by ensuring the first and last positions are equal.
400	ValidationException	polygon ring has more than 1000 vertices	'Geometry.Polygon' must not have more than 1000 vertices.
400	ValidationException	polygon ring has fewer than 4 positions	Number of vertices in 'Geometry.Polygon' must be greater or equal to 4.
400	ValidationException	invalid center	'Geometry.Circle.Center' must be a valid position (longitude/latitude pair).
400	ValidationException	radius must be greater than 0m	'Geometry.Circle.Radius' must be greater than 0 m.
400	ValidationException	longitude range should be between -180 and 180 degrees	Longitude must be between -180 and 180 degrees, but was set to <Provided Longitude>.

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
400	ValidationException	latitude range should be between -90 and 90 degrees	Latitude must be between -90 and 90 degrees, but was set to <Provided Longitude>.
400	ValidationException	polygon exterior ring is expected to be counter clockwise	'Geometry.Polygon' must be oriented counter-clockwise.
400	ValidationException	polygon interior ring should be clockwise oriented	'Geometry.Polygon' must be oriented clockwise.
400	ValidationException	radius must be less than or equal to 1000km	'Geometry.Circle.Radius' must be less than or equal to 1000 km.
400	ValidationException	timestamp.Parse() error	'SampleTime' must follow the format YYYY-MM-DDThh:mm:ss.sssZ.
400	ValidationException	invalid input	'SourceArn' must refer to a tracker resource.
400	ValidationException	arn: invalid prefix	'SourceArn' must be a valid ARN. For more information, see https://docs.aws.amazon.com/general/latest/gr/AWS-arns-and-namespaces.html .

Code d'erreur	Exception	Ancien message d'erreur	Nouveaux messages d'erreur
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/ .
409	ConflictException	collection already exists: <Geofence CollectionName>	Geofence Collection already exists: <GeofenceCollectionName>.
409	ConflictException	Resource conflict error	Geofence already exists: <Geofence Name>.

Exemples de code et didacticiels pour travailler avec Amazon Location Service

Cette rubrique contient une liste d'exemples de code, de didacticiels et d'articles de blog destinés à vous aider à en savoir plus sur Amazon Location Service. Chaque exemple de code inclut une description de son fonctionnement.

Vous trouverez des exemples supplémentaires sur la page [AWS Geospatial](#), la [GitHub page](#) d'[AWS exemples GitHub d'Amazon Location](#) et sur le [site du AWS blog](#).

Note

Il est bon de comprendre la différence entre la page AWS Géospatiale et la GitHub GitHub page AWS d'exemples.

- Géospatiale GitHub : la [GitHub page AWS Géospatiale](#) inclut des exemples créés et gérés par l'équipe Amazon Location Service.
- Exemples GitHub — La [GitHub page d'AWS échantillons d'Amazon Location](#) inclut des échantillons créés pour Amazon Location, mais qui peuvent être maintenus activement ou non.

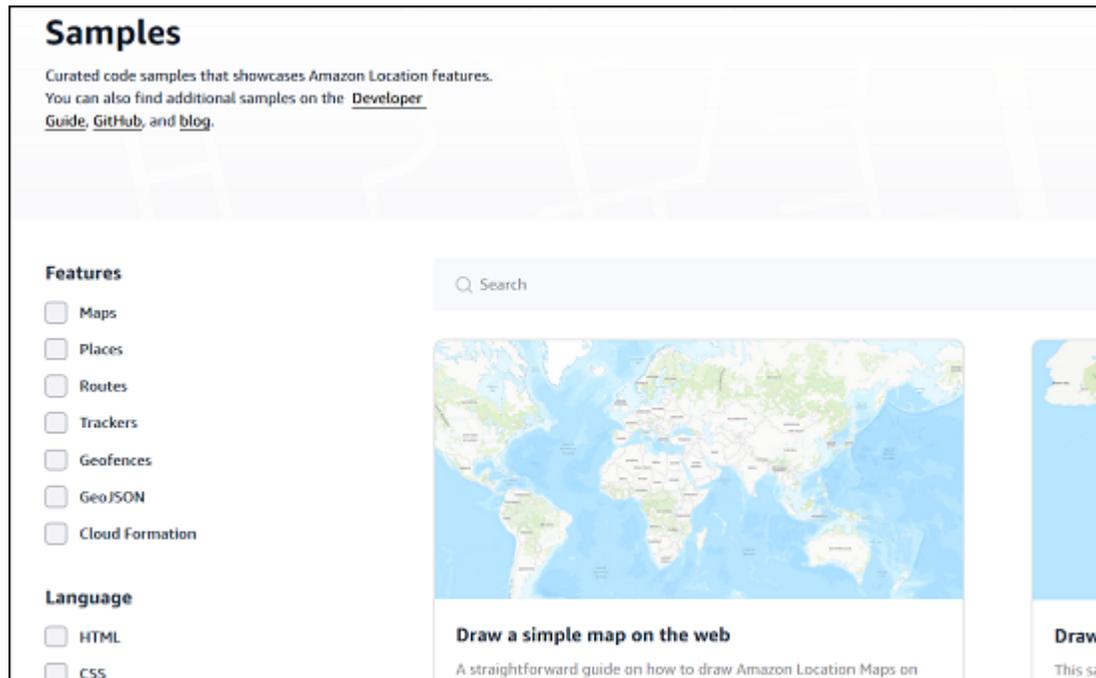
Le didacticiel de [démarrage rapide](#) est un bon point de départ avant d'utiliser d'autres exemples, car il montre comment remplir les prérequis utiles pour la plupart des exemples.

Rubriques

- [Site de démonstration Amazon Location](#)
- [Tutoriel : démarrage rapide](#)
- [Tutoriel : enrichissement des bases de données](#)
- [Exemple : application Explore](#)
- [Exemple : styler une carte](#)
- [Exemple : dessiner des marqueurs](#)
- [Exemple : dessiner des points groupés](#)
- [Exemple : Dessiner un polygone](#)
- [Exemple : modification de la langue de la carte](#)

- [Blog : Notifications relatives aux délais de livraison estimés](#)
- [Exemple : mises à jour de la position du stream](#)
- [Exemple : application mobile de géofencing et de suivi](#)

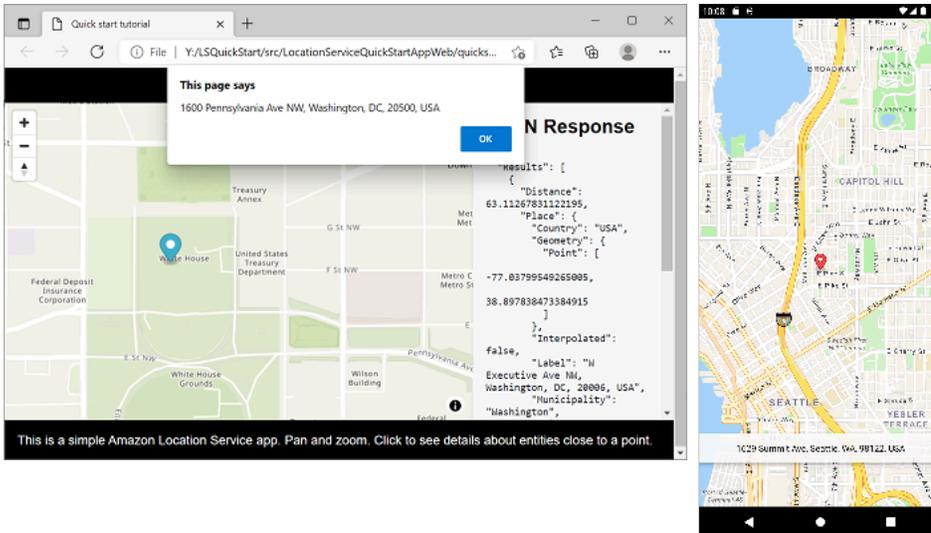
Site de démonstration Amazon Location



Vous pouvez voir des démos avec le code source d'Amazon Location Service en action sur le [site de démonstration d'Amazon Location](#). Ce site inclut une [démonstration Web hébergée](#), ainsi qu'une application de démonstration pour [Android](#).

Vous pouvez également trouver un large éventail d'exemples, filtrables par fonctionnalités, langue et plate-forme sur la page [Exemples](#) du site.

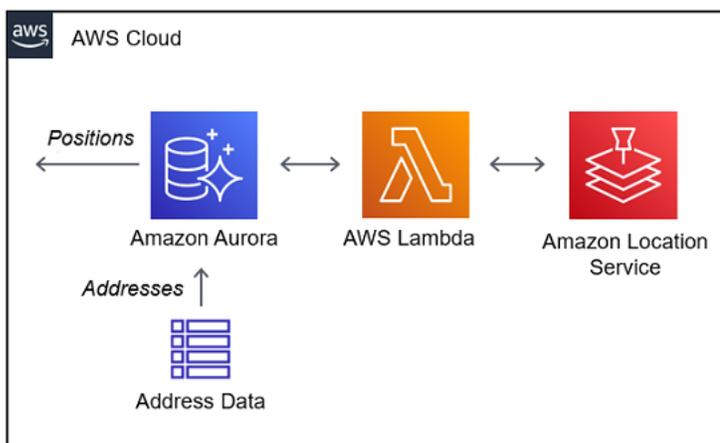
Tutoriel : démarrage rapide



Des didacticiels de démarrage rapide sont disponibles pour les appareils Web, iOS et Android. Pour chaque plateforme, le didacticiel explique comment ajouter une carte interactive à une application et comment appeler les API Amazon Location Service depuis votre application. Le didacticiel est disponible JavaScript dans une page Web statique, Kotlin pour une application téléphonique Android ou Swift pour une application iOS.

- JavaScript pour un lien de documentation de page Web statique : [Création d'une application Web](#)
- Lien de documentation de Kotlin pour une application Android : [Démarrage rapide avec Amazon Location Service](#)
- Lien vers la documentation de Swift pour une application iOS : [Création d'une application iOS](#)

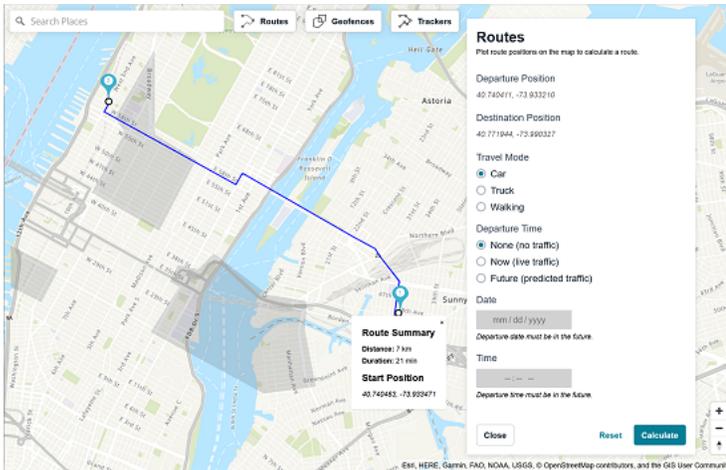
Tutoriel : enrichissement des bases de données



Ce didacticiel explique comment utiliser Amazon Location Service, appelé depuis AWS Lambda pour normaliser les adresses et ajouter la latitude et la longitude aux enregistrements d'une base de données Amazon Aurora. Utilisez Amazon Aurora et AWS Lambda.

Lien vers la documentation : [Amazon Aurora PostgreSQL fonctions définies par l'utilisateur pour Amazon Location Service](#)

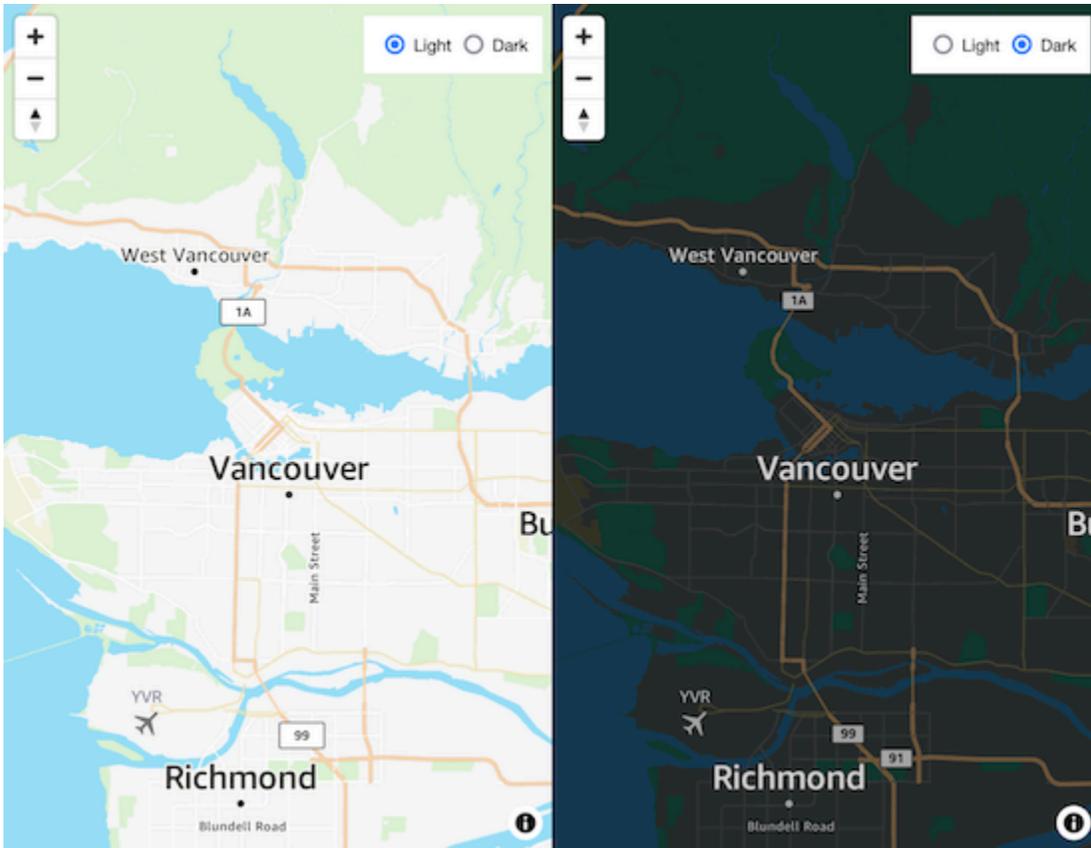
Exemple : application Explore



L'un des meilleurs moyens de découvrir les fonctionnalités d'Amazon Location Service consiste à utiliser la [fonctionnalité Explore](#) de la console Amazon Location. Cet exemple d'application Web complet imite les fonctionnalités des cartes, des lieux, des itinéraires, des barrières géographiques et des traceurs de la console pour vous montrer comment recréer ces fonctionnalités dans votre propre application. Utilisez Amplify, React et JavaScript.

GitHub Lien vers des exemples : [Découvrez un exemple d'application](#)

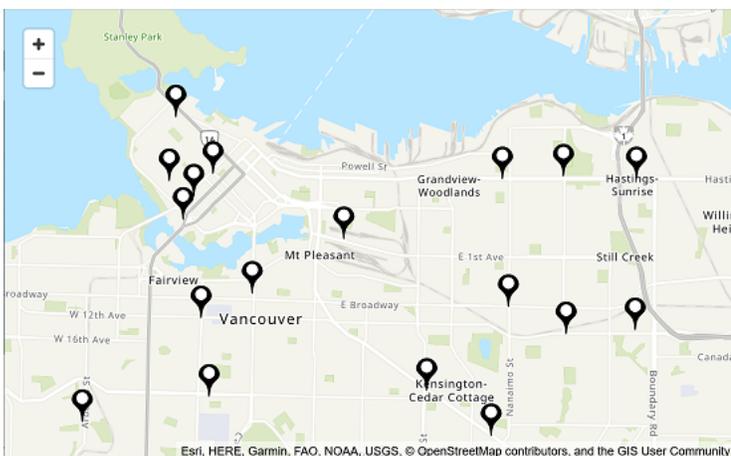
Exemple : styler une carte



Cet exemple de code montre comment passer d'une carte satellite à une carte routière vectorielle, MapLibre en utilisant in JavaScript. Utilisations MapLibre, l'assistant d'authentification Amazon Location, et JavaScript.

GitHub Lien géospatial : [carte interactive avec changement de style](#)

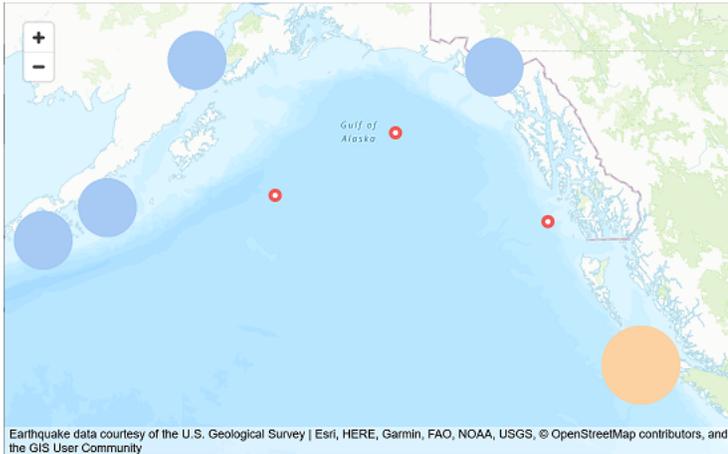
Exemple : dessiner des marqueurs



Cet exemple de code montre les sites Amazon Locker à Vancouver, en Colombie-Britannique, au Canada. Il montre comment dessiner des marqueurs à l'emplacement des points. Utilisez Node.js MapLibre, React, l'assistant d'authentification Amazon Location et JavaScript.

GitHub Lien géospatial : [carte interactive avec marqueurs aux points](#)

Exemple : dessiner des points groupés



À l'aide des données sismiques de l'USGS, cet exemple de code montre comment dessiner des points regroupés lorsqu'ils sont proches les uns des autres sur la carte. Utilisez Node.js MapLibre, React, Amplify et JavaScript.

Exemple de GitHub lien : [carte interactive avec groupes de points](#)

Exemple : Dessiner un polygone



Cet exemple de code montre comment dessiner un polygone sur la carte. Utilisez Node.js MapLibre, React, l'assistant d'authentification Amazon Location et JavaScript.

GitHub Lien géospatial : [carte interactive avec polygones](#)

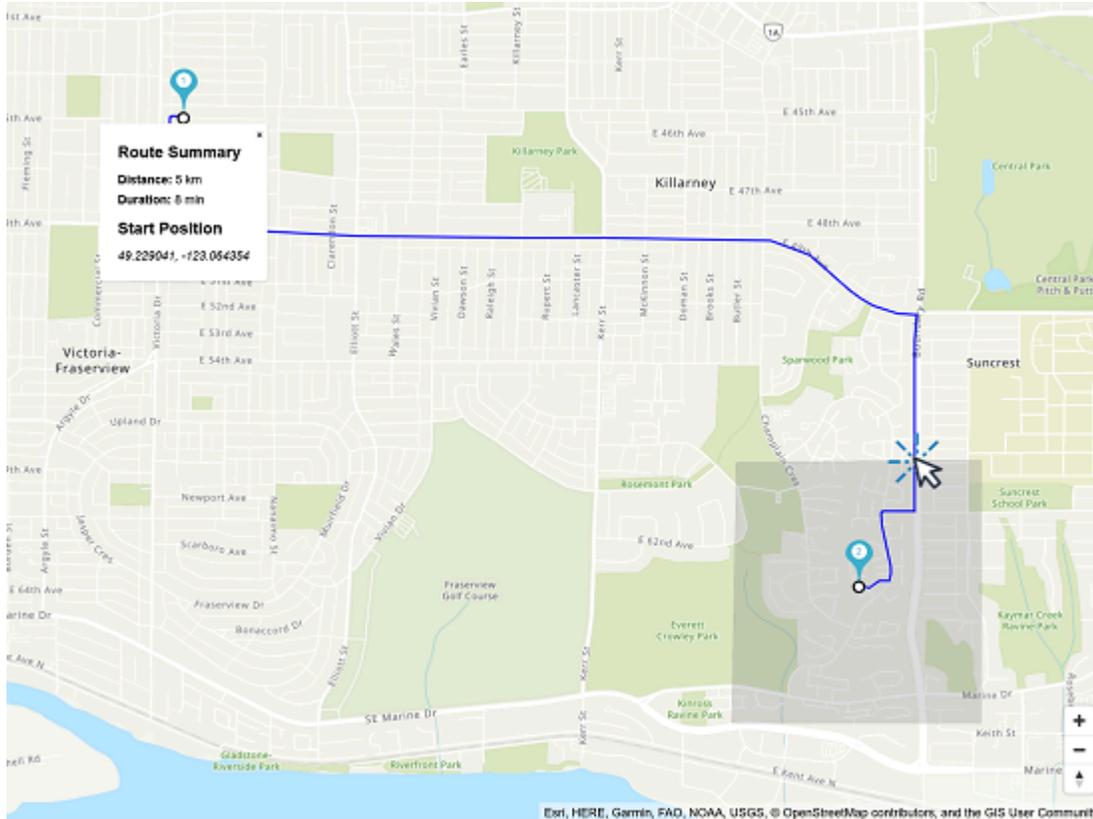
Exemple : modification de la langue de la carte



Cet exemple de code montre comment modifier la langue d'affichage des cartes dans Amazon Location. Utilisez Amplify, React et MapLibre

GitHub Lien vers des exemples : [Exemple de modification de la langue de la carte](#)

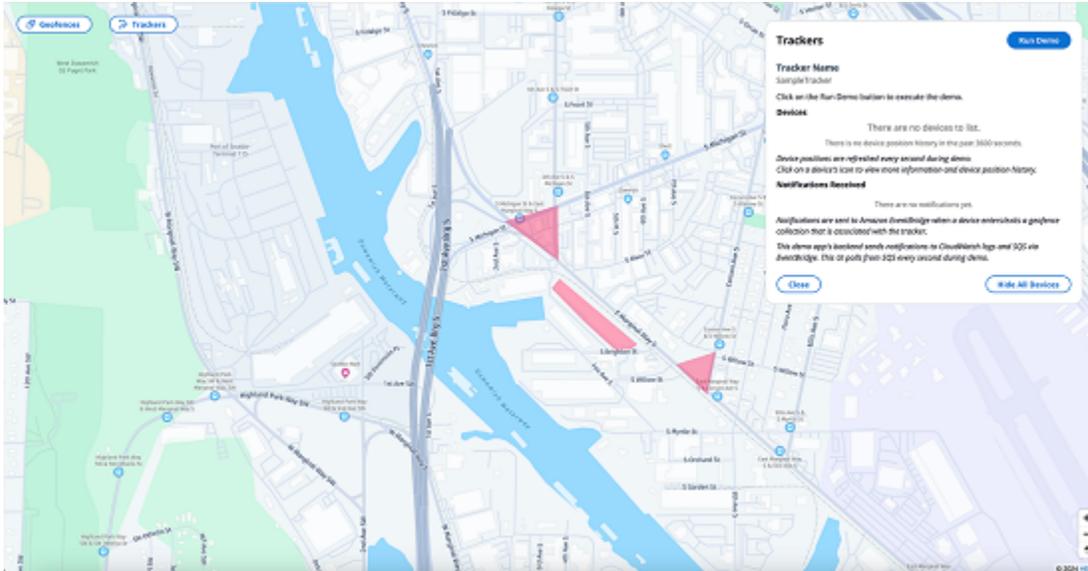
Blog : Notifications relatives aux délais de livraison estimés



Ce billet de blog présente différentes manières d'informer les clients des délais de livraison estimés. Il explique comment utiliser les itinéraires pour afficher le temps de conduite estimé, puis utiliser des traceurs et des géofences pour avertir lorsqu'un chauffeur se rapproche du client. Utilisez Amplify, React EventBridge, Amazon et Amazon Simple Notification Service (Amazon SNS).

Lien du blog : [Heure d'arrivée estimée et notifications de proximité](#)

Exemple : mises à jour de la position du stream



Application Kinesis Stream To Tracker : cet exemple montre comment utiliser Kinesis Data Stream pour publier des mises à jour de suivi avec Amazon Location Service. L'exemple est une application Lambda déployable écrite en python qui peut être intégrée à un flux de données Kinesis pour utiliser les événements Kinesis et mettre à jour par lots les positions des appareils.

Lien vers le dépôt : [Amazon Location Amazon Kinesis Data Streams Stream To Tracker App](#)

Pour plus d'informations sur le suivi et les géofences, consultez la documentation sur les [géofences](#) et les traceurs. [Les développeurs peuvent déployer l'application en suivant la documentation d'AWS Serverless Application Repository, ou directement depuis la console Lambda.](#)

Exemple d'application Device Position Streaming : Cet exemple de code montre comment diffuser les données de position de l'appareil vers un Kinesis Data Stream et comment fonctionnent les notifications de géofence. Cette application dépend de l'exemple d'application Kinesis Stream to Tracker, répertorié ci-dessus, pour que les positions du tracker diffusées soient mises à jour dans Amazon Location Service.

Lien vers le dépôt : [Amazon Location Device Position Streaming Sample App](#)

Exemple : application mobile de géofencing et de suivi

Cet exemple d'application montre comment un traceur et une clôture géographique interagissent à l'aide d'une combinaison des fonctionnalités Lambda et AWS IoT Amazon Location. Des tutoriels sont disponibles pour iOS et Android.

Lien vers le didacticiel : [exemple d'application mobile Geofence and Tracker](#)

Comment utiliser Amazon Location Service

Vous pouvez utiliser les fonctionnalités d'Amazon Location Service pour effectuer des tâches liées à la géographie et à la localisation. Vous pouvez ensuite combiner ces tâches pour traiter des cas d'utilisation plus complexes tels que le géomarketing, la livraison et le suivi des actifs.

Lorsque vous êtes prêt à intégrer des fonctionnalités de localisation dans votre application, utilisez les méthodes suivantes pour utiliser les fonctionnalités d'Amazon Location Service, en fonction de vos objectifs et de vos préférences :

- Outils d'exploration : si vous souhaitez tester les ressources Amazon Location, les outils suivants constituent le moyen le plus rapide d'accéder aux API et de les tester :
 - La [console Amazon Location](#) fournit une variété d'outils d'accès rapide. Vous pouvez créer et gérer vos ressources et essayer les API à l'aide [de la page Explorer](#). La console est également utile pour créer des ressources (généralement une tâche ponctuelle) en vue de l'utilisation de l'une des autres méthodes décrites plus loin.
 - L'[interface de ligne de AWS commande](#) (CLI) vous permet de créer des ressources et d'accéder aux API Amazon Location à l'aide d'un terminal. Il AWS CLI gère l'authentification lorsque vous la configurez avec vos informations d'identification.
 - Vous pouvez consulter [des exemples de code et des didacticiels](#) qui montrent comment effectuer des tâches à l'aide des API Amazon Location Service. Cela inclut [un exemple](#) qui imite la plupart des fonctionnalités de la page Explore de la console.
- SDK de plate-forme : si vous ne visualisez pas les données sur une carte, vous pouvez utiliser n'importe quel [outil AWS standard pour vous appuyer sur ces outils](#). AWS
 - Les SDK suivants sont disponibles : C++, Go, Java, .NET JavaScript, Node.js, PHP, Python et Ruby.
- SDK et bibliothèques frontaux : si vous souhaitez utiliser Amazon Location pour créer une application sur une plateforme mobile ou visualiser des données sur une carte sur n'importe quelle plateforme, les options suivantes s'offrent à vous :
 - Les AWS Amplify bibliothèques intègrent Amazon Location dans les applications [iOS](#), [Android](#) et [JavaScriptWeb](#).
 - Les MapLibre bibliothèques vous permettent de restituer des cartes côté client dans des applications [iOS](#), [Android](#) et [JavaScriptWeb](#) à l'aide d'Amazon Location.

- Les bibliothèques Tangram ES vous permettent de rendre des cartes 2D et 3D à partir de données vectorielles à l'aide d'OpenGL ES [dans](#) les applications Web [iOS](#) et Android. Il existe également Tangram pour les applications [JavaScriptWeb](#).
- Envoi de requêtes HTTPS directes : si vous travaillez avec un langage de programmation pour lequel aucun SDK n'est disponible, ou si vous souhaitez mieux contrôler la manière dont vous envoyez une demande AWS, vous pouvez accéder à Amazon Location en envoyant des demandes HTTPS directes authentifiées par le processus de signature Signature Version 4. Pour plus d'informations sur le [processus de signature de Signature version 4](#), consultez le [Références générales AWS](#).

Ce chapitre décrit de nombreuses tâches communes aux applications utilisant des données de localisation. La section consacrée aux [cas d'utilisation courants](#) décrit comment les combiner avec d'autres AWS services pour obtenir des cas d'utilisation plus complexes.

Rubriques

- [Conditions préalables à l'utilisation d'Amazon Location Service](#)
- [Utilisation d'Amazon Location Maps dans votre application](#)
- [Recherche de données de localisation et de géolocalisation à l'aide d'Amazon Location](#)
- [Calcul des itinéraires à l'aide d'Amazon Location Service](#)
- [Géolocalisation d'une zone d'intérêt à l'aide d'Amazon Location](#)
- [Marquage de vos ressources Amazon Location Service](#)
- [Octroi de l'accès à Amazon Location Service](#)
- [Surveillance d'Amazon Location Service](#)
- [Création de ressources Amazon Location Service avec AWS CloudFormation](#)

Conditions préalables à l'utilisation d'Amazon Location Service

Cette section décrit ce que vous devez faire pour utiliser Amazon Location Service. Vous devez disposer d'un Compte AWS et avoir configuré l'accès à Amazon Location pour les utilisateurs qui souhaitent l'utiliser.

Inscrivez-vous pour un Compte AWS

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

Pour vous inscrire à un Compte AWS

1. Ouvrez <https://portal.aws.amazon.com/billing/signup>.
2. Suivez les instructions en ligne.

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

Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWS est créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. Pour des raisons de sécurité, attribuez un accès administratif à un utilisateur et utilisez uniquement l'utilisateur root pour effectuer [les tâches nécessitant un accès utilisateur root](#).

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

Création d'un utilisateur doté d'un accès administratif

Une fois que vous vous êtes inscrit à un utilisateur administratif Compte AWS, que vous Utilisateur racine d'un compte AWS l'avez sécurisé AWS IAM Identity Center, que vous l'avez activé et que vous en avez créé un, afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisez votre Utilisateur racine d'un compte AWS

1. Connectez-vous en [AWS Management Console](#) tant que propriétaire du compte en choisissant Utilisateur root et en saisissant votre adresse Compte AWS e-mail. Sur la page suivante, saisissez votre mot de passe.

Pour obtenir de l'aide pour vous connecter en utilisant l'utilisateur racine, consultez [Connexion en tant qu'utilisateur racine](#) dans le Guide de l'utilisateur Connexion à AWS .

2. Activez l'authentification multifactorielle (MFA) pour votre utilisateur racine.

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

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

Pour obtenir des instructions, consultez [Activation d' AWS IAM Identity Center](#) dans le Guide de l'utilisateur AWS IAM Identity Center .

2. Dans IAM Identity Center, accordez un accès administratif à un utilisateur.

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

Connectez-vous en tant qu'utilisateur disposant d'un accès administratif

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

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

Attribuer l'accès à des utilisateurs supplémentaires

1. Dans IAM Identity Center, créez un ensemble d'autorisations conforme aux meilleures pratiques en matière d'application des autorisations du moindre privilège.

Pour obtenir des instructions, voir [Création d'un ensemble d'autorisations](#) dans le guide de AWS IAM Identity Center l'utilisateur.

2. Affectez des utilisateurs à un groupe, puis attribuez un accès d'authentification unique au groupe.

Pour obtenir des instructions, voir [Ajouter des groupes](#) dans le guide de AWS IAM Identity Center l'utilisateur.

Accorder l'accès à Amazon Location Service

Vos utilisateurs non administrateurs ne disposent d'aucune autorisation par défaut. Avant qu'ils puissent accéder à Amazon Location, vous devez leur accorder une autorisation en joignant une

politique IAM avec des autorisations spécifiques. Veillez à respecter le principe du moindre privilège lorsque vous accordez l'accès aux ressources.

Note

Pour plus d'informations sur l'accès des utilisateurs non authentifiés aux fonctionnalités d'Amazon Location Service (par exemple, dans une application Web), consultez [Octroi de l'accès à Amazon Location Service](#)

L'exemple de politique suivant donne à un utilisateur l'autorisation d'accéder à toutes les opérations Amazon Location. Pour obtenir plus d'exemples, consultez [Exemples de politiques basées sur l'identité pour Amazon Location Service](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "geo:*"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

Pour activer l'accès, ajoutez des autorisations à vos utilisateurs, groupes ou rôles :

- Utilisateurs et groupes dans AWS IAM Identity Center :

Créez un jeu d'autorisations. Suivez les instructions de la rubrique [Création d'un jeu d'autorisations](#) du Guide de l'utilisateur AWS IAM Identity Center .

- Utilisateurs gérés dans IAM par un fournisseur d'identité :

Créez un rôle pour la fédération d'identité. Pour plus d'informations, voir la rubrique [Création d'un rôle pour un fournisseur d'identité tiers \(fédération\)](#) du Guide de l'utilisateur IAM.

- Utilisateurs IAM :

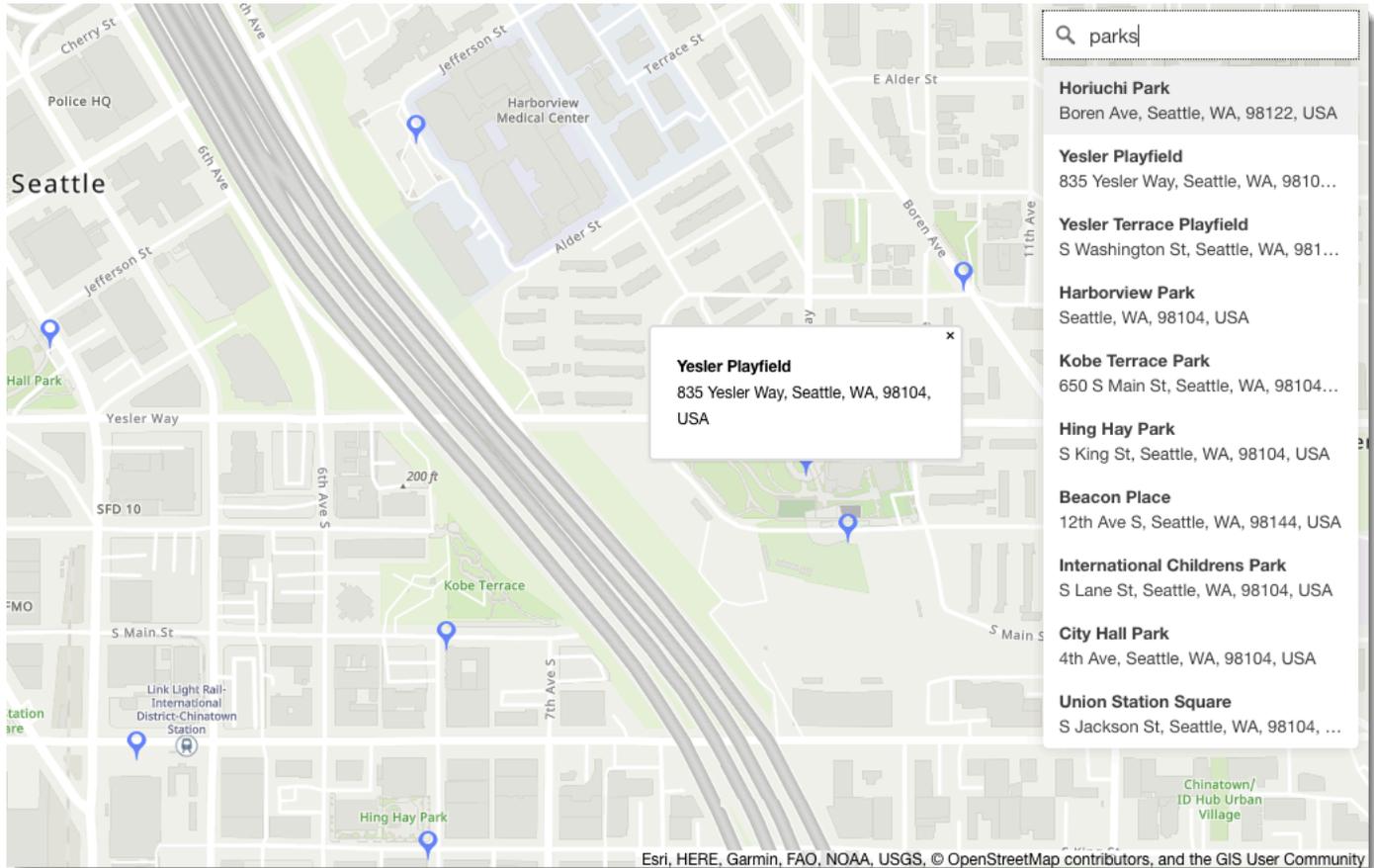
- Créez un rôle que votre utilisateur peut assumer. Suivez les instructions de la rubrique [Création d'un rôle pour un utilisateur IAM](#) du Guide de l'utilisateur IAM.

- (Non recommandé) Attachez une politique directement à un utilisateur ou ajoutez un utilisateur à un groupe d'utilisateurs. Suivez les instructions de la rubrique [Ajout d'autorisations à un utilisateur \(console\)](#) du Guide de l'utilisateur IAM.

Lorsque vous créez des applications qui utilisent Amazon Location Service, certains utilisateurs peuvent avoir besoin d'un accès non authentifié. Pour ces cas d'utilisation, consultez la section [Activation de l'accès non authentifié à l'aide d'Amazon Cognito](#).

Utilisation d'Amazon Location Maps dans votre application

Les cartes de localisation Amazon sont économiques et interactives. Vous pouvez remplacer une carte existante dans votre application pour économiser de l'argent ou en ajouter une nouvelle pour afficher visuellement les données de localisation, telles que l'emplacement de votre boutique.



Amazon Location Service vous permet de choisir un fournisseur de données pour les opérations cartographiques en créant et en configurant une ressource cartographique. La ressource cartographique configure le fournisseur de données et le style utilisé pour le rendu de la carte.

Après avoir créé votre ressource, vous pouvez envoyer des demandes en utilisant directement le AWS SDK ou en utilisant une bibliothèque spécialement conçue pour le rendu de cartes dans votre environnement.

Note

Pour un aperçu des concepts cartographiques, voir [Mappages](#).

Rubriques

- [Prérequis](#)
- [Afficher une carte dans votre application](#)
- [Dessiner des entités de données sur une carte](#)
- [Définition de l'étendue d'une carte à l'aide de MapLibre](#)
- [Gestion de vos ressources cartographiques](#)

Prérequis

Avant d'afficher une carte dans votre application, suivez les étapes préalables suivantes :

Rubriques

- [Création d'une ressource cartographique](#)
- [Authentification de vos demandes](#)

Création d'une ressource cartographique

Pour utiliser une carte dans votre application, vous devez disposer d'une ressource cartographique qui indique le style de carte et le fournisseur de données à utiliser dans vos cartes.

Note

Si votre application assure le suivi ou le routage des ressources que vous utilisez dans votre entreprise, comme les véhicules de livraison ou les employés, vous ne devez pas utiliser Esri comme fournisseur de géolocalisation. Consultez la section 82 des [Conditions de service AWS](#) pour plus de détails.

Vous pouvez créer une ressource cartographique à l'aide de la console Amazon Location Service, de l'AWS CLI, ou des API Amazon Location.

Console

Pour créer une ressource cartographique à l'aide de la console Amazon Location Service

1. Dans la console Amazon Location, sur la page [Cartes](#), choisissez Create map pour prévisualiser les styles de carte.
2. Ajoutez un nom et une description pour la nouvelle ressource cartographique.
3. Choisissez un style de carte.

Note

Si votre application assure le suivi ou le routage des ressources que vous utilisez dans votre entreprise, comme les véhicules de livraison ou les employés, vous ne devez pas utiliser Esri comme fournisseur de géolocalisation. Consultez la section 82 des [Conditions de service AWS](#) pour plus de détails.

4. Choisissez parmi les [Opinions politiques](#) à utiliser.
5. Acceptez les conditions générales d'Amazon Location, puis choisissez Create map. Vous pouvez interagir avec la carte que vous avez choisie : zoomer, dézoomer ou vous déplacer dans n'importe quelle direction.
6. Pour permettre à vos utilisateurs de changer de style (par exemple, pour leur permettre de passer de l'imagerie satellite au style vectoriel), vous devez créer une ressource cartographique pour chaque style.

Vous pouvez supprimer des ressources contenant des styles de carte que vous ne souhaitez pas utiliser sur la [page d'accueil de Maps](#) de la console.

API

Pour créer une ressource cartographique à l'aide des API Amazon Location

Utilisez l'[CreateMap](#) opération depuis les API Amazon Location.

L'exemple suivant est une demande d'API pour créer une ressource cartographique appelée à *ExempleMap* à l'aide du style de *VectorEsriStreets* carte.

```
POST /maps/v0/maps HTTP/1.1
Content-type: application/json

{
  "Configuration": {
    "Style": "VectorEsriStreets"
  },
  "MapName": "ExampleMap"
}
```

Note

Si votre application assure le suivi ou le routage des ressources que vous utilisez dans votre entreprise, comme les véhicules de livraison ou les employés, vous ne devez pas utiliser Esri comme fournisseur de géolocalisation. Consultez la section 82 des [Conditions de service AWS](#) pour plus de détails.

AWS CLI

Pour créer une ressource cartographique à l'aide de AWS CLI commandes

Utilisez la commande [create-map](#).

L'exemple suivant crée une ressource cartographique appelée *ExampleMap* using *VectorEsriStreets* comme style de carte.

```
aws location \
  create-map \
  --configuration Style="VectorEsriStreets" \
  --map-name "ExampleMap"
```

Note

Si votre application assure le suivi ou le routage des ressources que vous utilisez dans votre entreprise, comme les véhicules de livraison ou les employés, vous ne devez pas utiliser Esri comme fournisseur de géolocalisation. Consultez la section 82 des [Conditions de service AWS](#) pour plus de détails.

Authentification de vos demandes

Une fois que vous avez créé une ressource cartographique et que vous êtes prêt à intégrer des fonctionnalités de localisation dans votre application, vous devez choisir la manière dont vous souhaitez authentifier vos demandes.

Note

La plupart des applications frontales de cartes nécessitent un accès non authentifié aux cartes ou à d'autres fonctionnalités d'Amazon Location Service. En fonction de votre application, vous pouvez utiliser AWS Signature v4 pour authentifier les demandes, ou vous pouvez utiliser les clés d'API Amazon Cognito ou Amazon Location pour une utilisation non authentifiée. Pour en savoir plus sur toutes ces options, consultez [Octroi de l'accès à Amazon Location Service](#).

Afficher une carte dans votre application

Cette section propose des didacticiels sur l'utilisation des outils de rendu cartographique pour afficher une carte dans votre application mobile ou Web lorsque vous utilisez les API Amazon Location. Comme indiqué dans cette [Comment utiliser Amazon Location Service](#) rubrique, vous avez le choix entre plusieurs bibliothèques à utiliser lors du rendu de cartes avec Amazon Location, notamment Amplify et MapLibre Tangram.

Procédez de l'une des manières suivantes pour afficher une carte dans votre application :

- Le moyen le plus direct d'afficher une carte dans vos applications frontales Web et mobiles est de l'utiliser MapLibre. Vous pouvez suivre les [MapLibre tutoriels](#) ou même le [didacticiel de démarrage rapide](#) pour apprendre à les utiliser MapLibre.
- Si vous êtes un AWS Amplify développeur existant, vous souhaitez peut-être utiliser le SDK Amplify Geo. Pour en savoir plus, suivez le didacticiel [Amplify](#).
- Si vous utilisez déjà Tangram et que vous souhaitez continuer à l'utiliser pour afficher votre carte, tout en passant à Amazon Location Service, suivez le didacticiel [Tangram](#).

Rubriques

- [Utilisation de la MapLibre bibliothèque avec Amazon Location Service](#)
- [Utilisation de la bibliothèque Amplify avec Amazon Location Service](#)

- [Utilisation de Tangram avec Amazon Location Service](#)

Utilisation de la MapLibre bibliothèque avec Amazon Location Service

Les didacticiels suivants vous expliquent comment utiliser la MapLibre bibliothèque avec Amazon Location.

Rubriques

- [Utilisation de MapLibre GL JS avec Amazon Location Service](#)
- [Utilisation du SDK MapLibre natif pour Android avec Amazon Location Service](#)
- [Utilisation du SDK MapLibre natif pour iOS avec Amazon Location Service](#)

Utilisation de MapLibre GL JS avec Amazon Location Service

Utilisez [MapLibre GL JS](#) pour intégrer des cartes côté client dans des applications Web.

MapLibre GL JS est une JavaScript bibliothèque open source compatible avec les styles et les vignettes fournis par l'API Amazon Location Service Maps. Vous pouvez intégrer MapLibre GL JS dans un code HTML ou une JavaScript application de base pour intégrer des cartes côté client personnalisables et réactives.

Ce didacticiel explique comment intégrer MapLibre GL JS à Amazon Location dans un code HTML et une JavaScript application de base. Les mêmes bibliothèques et techniques présentées dans ce didacticiel s'appliquent également aux frameworks, tels que [React](#) et [Angular](#).

L'exemple d'application de ce didacticiel est disponible dans le référentiel d'exemples Amazon Location Service sur [GitHub](#).

Création de l'application : échafaudage

Ce didacticiel crée une application Web qui permet JavaScript de créer une carte sur une page HTML.

Commencez par créer une page HTML (`index.html`) qui inclut le conteneur de la carte :

- Entrez un `div` élément avec un `ou id map` pour appliquer les dimensions de la carte à la vue cartographique. Les dimensions sont héritées de la fenêtre d'affichage.

```
<html>
```

```
<head>
  <style>
    body {
      margin: 0;
    }

    #map {
      height: 100vh; /* 100% of viewport height */
    }
  </style>
</head>
<body>
  <!-- map container -->
  <div id="map" />
</body>
</html>
```

Création de l'application : ajout de dépendances

Ajoutez les dépendances suivantes à votre application :

- MapLibre GL JS (v3.x), et son CSS associé.
- L'emplacement d'Amazon [JavaScript Aide à l'authentification](#).

```
<!-- 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>
```

Cela crée une page vide contenant le conteneur de la carte.

Création de l'application : Configuration

Pour configurer votre application à l'aide de JavaScript :

1. Entrez les noms et les identifiants de vos ressources.

```
// Cognito Identity Pool ID
const identityPoolId = "us-east-1:54f2ba88-9390-498d-aaa5-0d97fb7ca3bd";
// Amazon Location Service Map name
const mapName = "ExampleMap";
```

2. Instanciez un fournisseur d'informations d'identification à l'aide du pool d'identités non authentifié que vous avez créé dans [Utilisation de cartes - Étape 2](#), Configuration de l'authentification. Nous allons le mettre dans une fonction appelée `initializeMap`, qui contiendra également un autre code d'initialisation de la carte, ajouté à l'étape suivante

```
// 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
}
```

Création de l'application : initialisation de la carte

Pour que la carte s'affiche une fois la page chargée, vous devez l'initialiser. Vous pouvez ajuster l'emplacement initial de la carte, ajouter des commandes supplémentaires et superposer des données.

```
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

Vous devez indiquer l'attribution d'un mot ou d'un texte pour chaque fournisseur de données que vous utilisez, que ce soit sur votre application ou dans votre documentation. Les chaînes d'attribution sont incluses dans la réponse du descripteur de style sous les sources `.grabmaptiles.attribution` touches `sources.esri.attributionsources.here.attribution`, et MapLibre GL JS fournira automatiquement une attribution. Lorsque vous utilisez les ressources Amazon Location avec [des fournisseurs de données](#), assurez-vous de lire [les conditions générales du service](#).

Exécution de l'application

Vous pouvez exécuter cet exemple d'application en l'utilisant sur un serveur Web local ou en l'ouvrant dans un navigateur.

Pour utiliser un serveur Web local, vous pouvez utiliser `npx`, car il est installé dans le cadre de Node.js. Vous pouvez utiliser `npx serve` depuis le même répertoire que `index.html`. Cela diffuse l'application sur `localhost:5000`.

Note

Si la politique que vous avez créée pour votre rôle Amazon Cognito non authentifié inclut `referer` une condition, il se peut que vous ne puissiez pas effectuer de tests avec des URL `localhost`: Dans ce cas, vous pouvez effectuer un test avec un serveur Web qui fournit une URL figurant dans votre politique.

Une fois le didacticiel terminé, l'application finale ressemble à l'exemple suivant.

```
<!-- 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(),
        });
      }
    </script>
  </body>
</html>
```

```
        map.addControl(new maplibregl.NavigationControl(), "top-left");
    }

    initializeMap();
</script>
</body>
</html>
```

L'exécution de cette application affiche une carte en plein écran utilisant le style de carte que vous avez choisi. Cet exemple est disponible dans le référentiel d'échantillons Amazon Location Service sur [GitHub](#).

Utilisation du SDK MapLibre natif pour Android avec Amazon Location Service

Utilisez le SDK [MapLibrenatif](#) pour intégrer des cartes interactives dans vos applications Android.

Le SDK MapLibre natif pour Android est une bibliothèque basée [sur Mapbox](#) Native, compatible avec les styles et les vignettes fournis par l'API Maps d'Amazon Location Service. Vous pouvez intégrer le SDK MapLibre natif pour Android afin d'intégrer des vues cartographiques interactives avec des cartes vectorielles évolutives et personnalisables dans vos applications Android.

Ce didacticiel explique comment intégrer le SDK MapLibre natif pour Android à Amazon Location. L'exemple d'application de ce didacticiel est disponible dans le référentiel d'exemples Amazon Location Service sur [GitHub](#).

Création de l'application : initialisation

Pour initialiser votre application :

1. Créez un nouveau projet Android Studio à partir du modèle Empty Activity.
2. Assurez-vous que Kotlin est sélectionné pour la langue du projet.
3. Sélectionnez un SDK minimum d'API 14 : Android 4.0 (Ice Cream Sandwich) ou version ultérieure.
4. Ouvrez la structure du projet, puis allez dans Fichier > Structure du projet... pour choisir la section Dépendances.
5. Lorsque cette option est <All Modules>sélectionnée, cliquez sur le bouton + pour ajouter une nouvelle dépendance de bibliothèque.
6. Ajoutez le SDK AWS Android version 2.20.0 ou ultérieure. Par exemple :
`com.amazonaws:aws-android-sdk-core:2.20.0`

7. Ajoutez le SDK MapLibre natif pour Android version 9.4.0 ou ultérieure. Par exemple :
`org.maplibre.gl:android-sdk:9.4.0`
8. Au niveau du projet de votre fichier build.gradle, ajoutez le référentiel maven suivant pour accéder aux packages pour Android : MapLibre

```
allprojects {
    repositories {
        // Retain your existing repositories
        google()
        jcenter()

        // Declare the repositories for MapLibre
        mavenCentral()
    }
}
```

Création de l'application : Configuration

Pour configurer votre application en fonction de vos ressources et de votre AWS région :

```
<?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>
```

Création de l'application : mise en page des activités

Modifier app/src/main/res/layout/activity_main.xml :

- Ajoutez un `MapView`, qui affiche la carte. Cela définira également le point central initial de la carte.
- Ajoutez un `TextView`, qui affiche l'attribution.

```
<?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

Vous devez indiquer l'attribution d'un mot ou d'un texte pour chaque fournisseur de données que vous utilisez, que ce soit sur votre application ou dans votre documentation. Les chaînes d'attribution sont incluses dans la réponse du descripteur de style sous les `source.grabmaptiles.attribution` touches `sources.esri.attributionsources`, `here.attribution`, et. Lorsque vous utilisez les ressources Amazon Location avec [des fournisseurs de données](#), assurez-vous de lire [les conditions générales du service](#).

Création de l'application : transformation des demandes

Créez une classe nommée `SigV4Interceptor` pour intercepter les demandes AWS et signez-les à l'aide de [Signature Version 4](#). Cela sera enregistré auprès du client HTTP utilisé pour récupérer les ressources cartographiques lors de la création de l'activité principale.

```
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()
    }
}
```

Création de l'application : activité principale

L'activité principale est chargée d'initialiser les vues qui seront affichées aux utilisateurs. Cela implique :

- Instanciation d'un Amazon Cognito. `CredentialsProvider`
- Enregistrement de l'intercepteur Signature version 4.
- Configuration de la carte en la pointant sur un descripteur de style de carte et en affichant l'attribution appropriée.

`MainActivity` est également chargé de transférer les événements du cycle de vie vers la vue cartographique, ce qui lui permet de conserver la fenêtre d'affichage active entre les invocations.

```
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()  
}  
}
```

L'exécution de cette application affiche une carte en plein écran dans le style de votre choix. Cet exemple est disponible dans le référentiel d'exemples Amazon Location Service sur [GitHub](#).

Utilisation du SDK MapLibre natif pour iOS avec Amazon Location Service

Utilisez le [SDK MapLibre natif pour iOS pour](#) intégrer des cartes côté client dans des applications iOS.

Le SDK MapLibre natif pour iOS est une bibliothèque basée [sur Mapbox GL Native](#), compatible avec les styles et les vignettes fournis par l'API Maps d'Amazon Location Service. Vous pouvez intégrer le SDK MapLibre natif pour iOS afin d'intégrer des vues cartographiques interactives avec des cartes vectorielles évolutives et personnalisables dans vos applications iOS.

Ce didacticiel explique comment intégrer le SDK MapLibre natif pour iOS à Amazon Location. L'exemple d'application de ce didacticiel est disponible dans le référentiel d'exemples Amazon Location Service sur [GitHub](#).

Création de l'application : initialisation

Pour initialiser votre application :

1. Créez un nouveau projet Xcode à partir du modèle d'application.
2. Sélectionnez SwiftUI pour son interface.
3. Sélectionnez l'application SwiftUI pour son cycle de vie.
4. Sélectionnez Swift pour sa langue.

Ajouter des MapLibre dépendances à l'aide de packages Swift

Pour ajouter une dépendance de package à votre projet Xcode :

1. Accédez à Fichier > Packages Swift > Ajouter une dépendance de package.
2. Entrez l'URL du référentiel : **<https://github.com/maplibre/maplibre-gl-native-distribution>**

Note

Pour plus d'informations sur les packages Swift, voir [Ajouter des dépendances de packages à votre application](#) sur Apple.com.

3. Dans votre terminal, installez CocoaPods :

```
sudo gem install cocoapods
```

4. Accédez au répertoire du projet de votre application et initialisez le Podfile avec le gestionnaire de CocoaPods packages :

```
pod init
```

5. Ouvrez le Podfile à ajouter en AWSCore tant que dépendance :

```
platform :ios, '12.0'  
  
target 'Amazon Location Service Demo' do  
  use_frameworks!  
  
  pod 'AWSCore'  
end
```

6. Téléchargez et installez les dépendances :

```
pod install --repo-update
```

7. Ouvrez l'espace de travail Xcode qui CocoaPods a créé :

```
xed .
```

Création de l'application : Configuration

Ajoutez les clés et valeurs suivantes à Info.plist pour configurer l'application :

Clé	Valeur
AWSRegion	us-east-1

Clé	Valeur
IdentityPoolId	us-east- 1:54 f2ba88-9390-498d-aaa5-0d97f b7ca3bd
MapName	ExampleMap

Création de l'application : ContentView mise en page

Pour afficher la carte, modifiez ContentView.swift :

- Ajoutez un MapView qui affiche la carte.
- Ajoutez un TextField qui affiche l'attribution.

Cela définit également le point central initial de la carte.

```
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

Vous devez indiquer l'attribution d'un mot ou d'un texte pour chaque fournisseur de données que vous utilisez, que ce soit sur votre application ou dans votre documentation. Les chaînes d'attribution sont incluses dans la réponse du descripteur de style sous les `source.grabmaptiles.attribution` touches `sources.esri.attribution`, `sources.here.attribution`, et. Lorsque vous utilisez les ressources Amazon Location avec [des fournisseurs de données](#), assurez-vous de lire [les conditions générales du service](#).

Création de l'application : transformation des demandes

Créez un nouveau fichier Swift nommé `AWSSignatureV4Delegate.swift` contenant la définition de classe suivante pour intercepter les demandes AWS et les signer à l'aide de [Signature Version 4](#). Une instance de cette classe sera affectée en tant que délégué au stockage hors ligne, qui est également responsable de la réécriture des URL, dans la vue cartographique.

```
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
}
}
```

Création de l'application : vue cartographique

La vue cartographique est chargée d'initialiser une instance du sous-jacent `AWSSignatureV4Delegate` et de le configurer `MGLMapView`, qui récupère les ressources et affiche la carte. Il gère également la propagation des chaînes d'attribution depuis la source du descripteur de style vers le `ContentView`.

Créez un nouveau fichier Swift nommé `MapView.swift` contenant la struct définition suivante :

```
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
}
}
```

L'exécution de cette application affiche une carte en plein écran dans le style de votre choix. Cet exemple est disponible dans le référentiel d'exemples Amazon Location Service sur [GitHub](#).

Utilisation de la bibliothèque Amplify avec Amazon Location Service

Le didacticiel suivant explique comment utiliser AWS Amplify Amazon Location. Amplify utilise MapLibre GL JS pour afficher des cartes dans votre application JavaScript basée.

Amplify est un ensemble de bibliothèques clientes open source qui fournissent des interfaces vers différentes catégories de services, notamment Amplify Geo, qui est alimenté par Amazon Location Service. [En savoir plus sur la JavaScript bibliothèque AWS Amplify géographique.](#)

Note

Ce didacticiel part du principe que vous avez déjà suivi les étapes décrites dans [Utilisation de cartes - Pour ajouter une carte à votre application](#).

Création de l'application : échafaudage

Ce didacticiel crée une application Web qui permet JavaScript de créer une carte sur une page HTML.

Commencez par créer une page HTML (`index.html`) qui inclut le conteneur de la carte :

- Entrez un `div` élément avec un `id` `map` pour appliquer les dimensions de la carte à la vue cartographique. Les dimensions sont héritées de la fenêtre d'affichage.

```
<html>
```

```

<head>
  <style>
    body { margin: 0; }
    #map { height: 100vh; } /* 100% of viewport height */
  </style>
</head>

<body>
  <!-- map container -->
  <div id="map" />
</body>
</html>

```

Création de l'application : ajout de dépendances

Ajoutez les dépendances suivantes à votre application :

- AWS Amplifybibliothèques de cartes et de géolocalisations.
- AWS Amplifybibliothèque de base.
- AWS Amplifybibliothèque d'authentification.
- AWS Amplifyfeuille de style.

```

<!-- CSS dependencies -->
  <link href="https://cdn.amplify.aws/packages/maplibre-
gl/1.15.2/maplibre-gl.css" rel="stylesheet" integrity="sha384-
DrPVD9GufrixGb7kWwRv0CywpXTmfvbK0Z5i5pN7urmIThew0zXKTME+gutUgtpeD"
  crossorigin="anonymous" referrerpolicy="no-referrer"></link>

<!-- JavaScript dependencies -->
  <script src="https://cdn.amplify.aws/packages/maplibre-gl/1.15.2/maplibre-gl.js"
  integrity="sha384-rwYfkmA0pciZS2bDuwZ/Xa/Gog6jXem8D/whm3wnsZSVFemDDlprcUXHnDDUcrNU"
  crossorigin="anonymous" referrerpolicy="no-referrer"></script>
  <script src="https://cdn.amplify.aws/packages/core/4.3.0/aws-amplify-core.min.js"
  integrity="sha384-70h+5w0l7XGyYvSqbkKi2Q7SA5K640V5nyW2/LEbevDQEV1HMJqJLA1A00z2hu8fJ"
  crossorigin="anonymous" referrerpolicy="no-referrer"></script>
  <script src="https://cdn.amplify.aws/packages/auth/4.3.8/aws-amplify-auth.min.js"
  integrity="sha384-jfkXCEfYyVmDXyKlgWNwv54xRaZgk14m7sje2jLVBtUXCD2p+WU8Y22mPZ9Xbdw"
  crossorigin="anonymous" referrerpolicy="no-referrer"></script>
  <script src="https://cdn.amplify.aws/packages/geo/1.1.0/aws-amplify-geo.min.js"
  integrity="sha384-TFMTyWuCbipXTzv0gzJbV8TPUupG1rA1AVrznAhCSpXTIdGw82bGd8RTk5rr3nP"
  crossorigin="anonymous" referrerpolicy="no-referrer"></script>

```

```
<script src="https://cdn.amplify.aws/packages/maplibre-gl-js-
amplify/1.1.0/maplibre-gl-js-amplify.umd.min.js" integrity="sha384-7/
RxWonKW1nM9zCKiwU9x6bkQTjldosg0D1vZYm0Zj+K/vUSnA3s0Mh1RRWAtHPi" crossorigin="anonymous"
referrerpolicy="no-referrer"></script>
<script>
  // application-specific code
</script>
```

Cela crée une page vide contenant le conteneur de la carte.

Création de l'application : Configuration

Pour configurer votre application à l'aide de JavaScript :

1. Entrez les identifiants du pool d'identités non authentifié que vous avez créé dans [Utilisation de cartes - Étape 2, Configuration](#) de l'authentification.

```
// 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. Configurez AWS Amplify pour utiliser les ressources que vous avez créées, y compris le pool d'identités et la ressource Map (illustrés ici avec le nom par défaut `deexplore.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,
    },
  }
});
```

Création de l'application : initialisation de la carte

Pour que la carte s'affiche une fois la page chargée, vous devez l'initialiser. Vous pouvez ajuster l'emplacement initial de la carte, ajouter des commandes supplémentaires et superposer des données.

```
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

Vous devez indiquer l'attribution d'un mot ou d'un texte pour chaque fournisseur de données que vous utilisez, que ce soit sur votre application ou dans votre documentation. Les chaînes d'attribution sont incluses dans la réponse du descripteur de style sous les sources `grabmaptiles.attribution` touches `sources.esri.attribution` `sources.here.attribution`, et. Amplify fournira automatiquement une attribution. Lorsque vous utilisez les ressources Amazon Location avec [des fournisseurs de données](#), assurez-vous de lire [les conditions générales du service](#).

Exécution de l'application

Vous pouvez exécuter cet exemple d'application en l'utilisant sur un serveur Web local ou en l'ouvrant dans un navigateur.

Pour utiliser un serveur Web local, vous pouvez utiliser `npx`, installé dans le cadre de Node.js, ou tout autre serveur Web de votre choix. Pour utiliser `npx`, tapez `npx serve` depuis le même répertoire que `index.html`. Cela diffuse l'application sur `localhost:5000`.

Note

Si la politique que vous avez créée pour votre rôle Amazon Cognito non authentifié inclut `referer` une condition, il se peut que vous ne puissiez pas effectuer de tests avec des URL `localhost`: Dans ce cas, vous pouvez effectuer un test avec un serveur Web qui fournit une URL figurant dans votre politique.

Une fois le didacticiel terminé, l'application finale ressemble à l'exemple suivant.

```
<html>
  <head>
    <!-- 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-TFMTyWuCbipXTzv0gzJbV8TPUupG1rA1AVrznAhCSpXTIdGw82bGd8RTk5rr3nP"
crossorigin="anonymous" referrerpolicy="no-referrer"></script>
    <script src="https://cdn.amplify.aws/packages/maplibre-gl-js-
amplify/1.1.0/maplibre-gl-js-amplify.umd.min.js" integrity="sha384-7/
```

```
RxWonKW1nM9zCKiwU9x6bkQTjldosg0D1vZYm0Zj+K/vUSnA3s0Mh1RRWAtHPi" crossorigin="anonymous"
referrerpolicy="no-referrer"></script>

<style>
  body { margin: 0; }
  #map { height: 100vh; }
</style>
</head>

<body>
  <div id="map" />
  <script type="module">
    // Cognito Identity Pool ID
    const identityPoolId = "region:identityPoolId"; // for example: us-
east-1:123example-1234-5678
    // extract the Region from the Identity Pool ID
    const region = identityPoolId.split(":")[0];

    // Configure Amplify
    const { Amplify } = aws_amplify_core;
    const { createMap } = AmplifyMapLibre;

    Amplify.configure({
      Auth: {
        identityPoolId,
        region,
      },
      geo: {
        AmazonLocationService: {
          maps: {
            items: {
              "explore.map": {
                style: "Default style"
              },
            },
            default: "explore.map",
          },
          region,
        },
      }
    });

    async function initializeMap() {
      const map = await createMap(
```

```
        {
            container: "map",
            center: [-123.1187, 49.2819],
            zoom: 10,
            hash: true,
        }
    );

    map.addControl(new maplibregl.NavigationControl(), "top-left");
}

initializeMap();
</script>
</body>
</html>
```

L'exécution de cette application affiche une carte en plein écran utilisant le style de carte que vous avez choisi. Cet exemple est également décrit dans l'onglet Intégrer une carte de n'importe quelle page de ressources cartographiques de la [console Amazon Location Service](#).

Une fois ce didacticiel terminé, consultez la rubrique [Afficher une carte](#) de la AWS Amplify documentation pour en savoir plus, notamment sur la façon d'afficher des marqueurs sur la carte.

Utilisation de Tangram avec Amazon Location Service

Cette section fournit les didacticiels suivants sur la façon d'intégrer Tangram à Amazon Location.

Important

Les styles Tangram présentés dans les didacticiels suivants sont uniquement compatibles avec les ressources cartographiques Amazon Location configurées avec ce `VectorHereContrast` style.

Voici un exemple de commande AWS CLI permettant de créer une nouvelle ressource cartographique appelée à *TangramExampleMap* l'aide du *VectorHereContrast* style :

```
aws --region us-east-1 \  
  location \  
  create-map \  
  --map-name "TangramExampleMap" \  
  --style "VectorHereContrast"
```

```
--configuration "Style=VectorHereContrast"
```

Note

La facturation est déterminée en fonction de votre utilisation. L'utilisation d'autres AWS services peut entraîner des frais. Pour plus d'informations, consultez les [tarifs d'Amazon Location Service](#).

Rubriques

- [Utilisation de Tangram avec Amazon Location Service](#)
- [Utilisation de Tangram ES pour Android avec Amazon Location Service](#)
- [Utilisation de Tangram ES pour iOS avec Amazon Location Service](#)

Utilisation de Tangram avec Amazon Location Service

[Tangram](#) est un moteur de cartographie flexible, conçu pour le rendu en temps réel de cartes 2D et 3D à partir de tuiles vectorielles. Il peut être utilisé avec les styles conçus par Mapzen et les vignettes HERE fournies par l'API Maps d'Amazon Location Service. Ce guide explique comment intégrer Tangram à Amazon Location dans une JavaScript application HTML/ de base, bien que les mêmes bibliothèques et techniques s'appliquent également lors de l'utilisation de frameworks tels que React et Angular.

Tangram est construit sur [Leaflet](#), une JavaScript bibliothèque open source de cartes interactives adaptées aux mobiles. Cela signifie que de nombreux plugins et contrôles compatibles avec Leaflet fonctionnent également avec Tangram.

Les styles Tangram conçus pour fonctionner avec le [schéma Tilezen](#) sont largement compatibles avec Amazon Location lorsque vous utilisez des cartes provenant de HERE. Il s'agit des licences suivantes :

- [Bubble Wrap](#) — Un style d'orientation complet avec des icônes utiles pour les points d'intérêt
- [Cinnabar](#) — Un look classique et incontournable pour les applications cartographiques générales
- [Refill](#) — Un style de carte minimaliste conçu pour les superpositions de visualisation de données, inspiré du style Toner emblématique de Stamen Design
- [Tron](#) — Une exploration des transformations d'échelle dans le langage visuel de TRON
- [Walkabout](#) — Un style axé sur le plein air, parfait pour la randonnée ou les sorties

[Ce guide explique comment intégrer Tangram à Amazon Location dans une JavaScript application HTML/ de base utilisant le style Tangram appelé Bubble Wrap.](#) Cet exemple est disponible dans le référentiel d'exemples Amazon Location Service sur [GitHub](#).

Bien que les autres styles Tangram soient mieux accompagnés de tuiles matricielles, qui encodent les informations de terrain, cette fonctionnalité n'est pas encore prise en charge par Amazon Location.

⚠ Important

Les styles Tangram présentés dans le didacticiel suivant sont uniquement compatibles avec les ressources cartographiques Amazon Location configurées avec ce `VectorHereContrast` style.

Création de l'application : échafaudage

L'application est une page HTML permettant JavaScript de créer la carte sur votre application Web. Créez une page HTML (`index.html`) et créez le conteneur de la carte :

- Entrez un `div` élément avec une carte `id` ou pour appliquer les dimensions de la carte à la vue cartographique.
- Les dimensions sont héritées de la fenêtre d'affichage.

```
<html>
  <head>
    <style>
      body {
        margin: 0;
      }

      #map {
        height: 100vh; /* 100% of viewport height */
      }
    </style>
  </head>
  <body>
    <!-- map container -->
    <div id="map" />
  </body>
```

```
</html>
```

Création de l'application : ajout de dépendances

Ajoutez les dépendances suivantes :

- Brochure et son code CSS associé.
- Tangram.
- SDK AWS pour JavaScript

```
<!-- CSS dependencies -->
<link
  rel="stylesheet"
  href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css"
  integrity="sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAsh0MAS6/keqq/
sMZM19scR4PsZChSR7A=="
  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>
```

Cela crée une page vide avec les prérequis nécessaires. L'étape suivante vous guide dans l'écriture du JavaScript code de votre application.

Création de l'application : Configuration

Pour configurer votre application avec vos ressources et vos informations d'identification :

1. Entrez les noms et les identifiants de vos ressources.

```
// 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. Instanciez un fournisseur d'informations d'identification à l'aide du pool d'identités non authentifié que vous avez créé dans [Utilisation de cartes - Étape 2](#), Configuration de l'authentification. Comme cela utilise des informations d'identification en dehors du flux de travail normal du SDK AWS, les sessions expirent au bout d'une heure.

```
// 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. Bien que Tangram vous permette de remplacer les URL utilisées pour récupérer les tuiles, il n'inclut pas la possibilité d'intercepter les demandes afin qu'elles puissent être signées.

Pour contourner ce problème, remplacez `sources.mapzen.url` pour pointer vers Amazon Location à l'aide d'un nom d'hôte synthétique `amazon.location`, qui sera géré par un technicien de [maintenance](#). Voici un exemple de configuration de scène à l'aide de [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,
  },
},
};
```

Création de l'application : transformation des demandes

Pour enregistrer et initialiser le service worker, créez une `registerServiceWorker` fonction à appeler avant l'initialisation de la carte. Cela enregistre le JavaScript code fourni dans un fichier séparé `sw.js` appelé `service worker controlling index.html`.

Les informations d'identification sont chargées depuis Amazon Cognito et sont transmises au technicien de service aux côtés de la région pour fournir les informations nécessaires à la signature des demandes de vignettes avec [Signature Version 4](#).

```
/**
 * Register a service worker that will rewrite and sign requests using Signature
 * Version 4.
 */
async function registerServiceWorker() {
  if ("serviceWorker" in navigator) {
    try {
      const reg = await navigator.serviceWorker.register("./sw.js");

      // refresh credentials from Amazon Cognito
      await credentials.refreshPromise();

      await reg.active.ready;

      if (navigator.serviceWorker.controller == null) {
        // trigger a navigate event to active the controller for this page
        window.location.reload();
      }

      // pass credentials to the service worker
      reg.active.postMessage({
        credentials: {
          accessKeyId: credentials.accessKeyId,
          secretAccessKey: credentials.secretAccessKey,
```

```
        sessionToken: credentials.sessionToken,
    },
    region: AWS.config.region,
  });
} catch (error) {
  console.error("Service worker registration failed:", error);
}
} else {
  console.warn("Service worker support is required for this example");
}
}
```

L'implémentation de Service Worker `sw.js` écoute les messages événements pour détecter les modifications des informations d'identification et de configuration de la région. Il agit également comme un serveur proxy en écoutant les `fetch` événements. Les événements ciblant le nom d'hôte `amazon.location` synthétique seront réécrits pour cibler l'API Amazon Location appropriée et signés à l'aide d'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));
});
```

Pour renouveler automatiquement les informations d'identification et les envoyer au service worker avant leur expiration, utilisez la fonction suivante dans `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());
}
```

Création de l'application : initialisation de la carte

Pour que la carte s'affiche une fois la page chargée, vous devez l'initialiser. Vous avez la possibilité d'ajuster l'emplacement initial de la carte, d'ajouter des contrôles supplémentaires et de superposer des données.

Note

Vous devez indiquer l'attribution d'un mot ou d'un texte pour chaque fournisseur de données que vous utilisez, que ce soit sur votre application ou dans votre documentation. Les chaînes d'attribution sont incluses dans la réponse du descripteur de style sous les `source.grabmaptiles.attribution` touches `sources.esri.attribution` `sources.here.attribution`, et. Étant donné que Tangram ne demande pas ces ressources et n'est compatible qu'avec les cartes d'HERE, utilisez « © 2020 HERE ». Lorsque vous utilisez les ressources Amazon Location avec [des fournisseurs de données](#), assurez-vous de lire [les conditions générales du service](#).

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

Exécution de l'application

Pour exécuter cet exemple, vous pouvez :

- Utilisez un hôte qui supporte le protocole HTTPS,
- Utilisez un serveur Web local pour respecter les restrictions de sécurité du personnel de maintenance.

Pour utiliser un serveur Web local, vous pouvez utiliser npx, car il est installé dans le cadre de Node.js. Vous pouvez utiliser npx serve depuis le même répertoire que index.html et sw.js. Cela sert l'application sur localhost:5000.

Voici le index.html fichier :

```
<!-- index.html -->
<html>
  <head>
    <link
      rel="stylesheet"
      href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css"
      integrity="sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmb1Ash0MAS6/
      keqq/sMZMZ19scR4PsZChSR7A=="
      crossorigin=""
```

```
    />
    <style>
      body {
        margin: 0;
      }

      #map {
        height: 100vh;
      }
    </style>
  </head>

  <body>
    <div id="map" />
    <script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"></script>
    <script src="https://unpkg.com/tangram"></script>
    <script src="https://sdk.amazonaws.com/js/aws-sdk-2.784.0.min.js"></script>
    <script>
      // configuration
      // Cognito Identity Pool ID
      const identityPoolId = "<Identity Pool ID>";
      // Amazon Location Service Map name; must be HERE-backed
      const mapName = "<Map name>";

      AWS.config.region = identityPoolId.split(":")[0];

      // instantiate a credential provider
      credentials = new AWS.CognitoIdentityCredentials({
        IdentityPoolId: identityPoolId,
      });

      const scene = {
        import: [
          // Bubble Wrap style
          "https://www.nextzen.org/carto/bubble-wrap-style/10/bubble-wrap-style.zip",
          "https://www.nextzen.org/carto/bubble-wrap-style/10/themes/label-7.zip",
          "https://www.nextzen.org/carto/bubble-wrap-style/10/themes/bubble-wrap-road-shields-usa.zip",
          "https://www.nextzen.org/carto/bubble-wrap-style/10/themes/bubble-wrap-road-shields-international.zip",
        ],
        // override values beneath the `sources` key in the style above
        sources: {
          mapzen: {
```

```
    // point at Amazon Location using a synthetic URL, which will be handled by
the service
    // worker
    url: `https://amazon.location/${mapName}/{z}/{x}/{y}`,
  },
  // effectively disable raster tiles containing encoded normals
  normals: {
    max_zoom: 0,
  },
  "normals-elevation": {
    max_zoom: 0,
  },
},
};

/**
 * Register a service worker that will rewrite and sign requests using Signature
Version 4.
 */
async function registerServiceWorker() {
  if ("serviceWorker" in navigator) {
    try {
      const reg = await navigator.serviceWorker.register("./sw.js");

      // refresh credentials from Amazon Cognito
      await credentials.refreshPromise();

      await reg.active.ready;

      if (navigator.serviceWorker.controller == null) {
        // trigger a navigate event to active the controller for this page
        window.location.reload();
      }

      // pass credentials to the service worker
      reg.active.postMessage({
        credentials: {
          accessKeyId: credentials.accessKeyId,
          secretAccessKey: credentials.secretAccessKey,
          sessionToken: credentials.sessionToken,
        },
        region: AWS.config.region,
      });
    } catch (error) {
```

```
        console.error("Service worker registration failed:", error);
    }
    } else {
        console.warn("Service Worker support is required for this example");
    }
}

/**
 * Initialize a map.
 */
async function initializeMap() {
    // register the service worker to handle requests to https://amazon.location
    await registerServiceWorker();

    // Initialize the map
    const map = L.map("map").setView([49.2819, -123.1187], 10);
    Tangram.leafletLayer({
        scene,
    }).addTo(map);
    map.attributionControl.setPrefix("");
    map.attributionControl.addAttribution("© 2020 HERE");
}

initializeMap();
</script>
</body>
</html>
```

Voici le sw.js fichier :

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

Cet exemple est disponible dans le référentiel d'exemples Amazon Location Service sur [GitHub](#).

Utilisation de Tangram ES pour Android avec Amazon Location Service

[Tangram ES](#) est une bibliothèque C++ pour le rendu de cartes 2D et 3D à partir de données vectorielles à l'aide d'OpenGL ES. C'est l'équivalent natif de [Tangram](#).

Les styles Tangram conçus pour fonctionner avec le [schéma Tilezen](#) sont largement compatibles avec Amazon Location lorsque vous utilisez des cartes provenant de HERE. Il s'agit des licences suivantes :

- [Bubble Wrap](#) — Un style d'orientation complet avec des icônes utiles pour les points d'intérêt.
- [Cinnabar](#) — Un look classique et incontournable pour les applications cartographiques générales.
- [Refill](#) — Un style de carte minimaliste conçu pour les superpositions de visualisation de données, inspiré du style Toner emblématique de Stamen Design.
- [Tron](#) — Une exploration des transformations d'échelle dans le langage visuel de TRON.
- [Walkabout](#) — Un style axé sur le plein air, parfait pour la randonnée ou les sorties.

Ce guide explique comment intégrer Tangram ES pour Android à Amazon Location en utilisant le style Tangram appelé Cinnabar. Cet exemple est disponible dans le référentiel d'exemples Amazon Location Service sur [GitHub](#).

Bien que les autres styles Tangram soient mieux accompagnés de tuiles matricielles, qui encodent les informations du terrain, cette fonctionnalité n'est pas encore prise en charge par Amazon Location.

Important

Les styles Tangram présentés dans le didacticiel suivant sont uniquement compatibles avec les ressources cartographiques Amazon Location configurées avec ce `VectorHereContrast` style.

Création de l'application : initialisation

Pour initialiser votre application :

1. Créez un nouveau projet Android Studio à partir du modèle Empty Activity.
2. Assurez-vous que Kotlin est sélectionné pour la langue du projet.
3. Sélectionnez un SDK minimum d'API 16 : Android 4.1 (Jelly Bean) ou version ultérieure.
4. Ouvrez la structure du projet pour sélectionner le fichier, la structure du projet... , puis choisissez la section Dépendances.
5. Lorsque cette option est <All Modules>sélectionnée, cliquez sur le bouton + pour ajouter une nouvelle dépendance de bibliothèque.
6. Ajoutez le SDK AWS Android version 2.19.1 ou ultérieure. Par exemple :
`com.amazonaws:aws-android-sdk-core:2.19.1`
7. Ajoutez la version 0.13.0 ou ultérieure de Tangram. Par exemple :
`com.mapzen.tangram:tangram:0.13.0`.

Note

Rechercher Tangram : `com.mapzen.tangram:tangram:0.13.0` générera un message indiquant qu'il est « introuvable », mais si vous cliquez sur OK, vous pourrez l'ajouter.

Création de l'application : Configuration

Pour configurer votre application en fonction de vos ressources et de votre AWS région :

1. Créer `app/src/main/res/values/configuration.xml`.
2. Entrez les noms et les identifiants de vos ressources, ainsi que la AWS région dans laquelle elles ont été créées :

```
<?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>
```

Création de l'application : mise en page des activités

Modifier `app/src/main/res/layout/activity_main.xml` :

- Ajoutez un `MapView`, qui affiche la carte. Cela définira également le point central initial de la carte.
- Ajoutez un `TextView`, qui affiche l'attribution.

Cela définira également le point central initial de la carte.

Note

Vous devez indiquer l'attribution d'un mot ou d'un texte pour chaque fournisseur de données que vous utilisez, que ce soit sur votre application ou dans votre documentation. Les chaînes d'attribution sont incluses dans la réponse du descripteur de style sous les `source.grabmaptiles.attribution` touches `sources.esri.attribution` `sources.here.attribution`, et. Étant donné que Tangram ne demande pas ces ressources et n'est compatible qu'avec les cartes d'HERE, utilisez « © 2020 HERE ». Lorsque vous utilisez les ressources Amazon Location avec [des fournisseurs de données](#), assurez-vous de lire [les conditions générales du service](#).

```
<?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>
```

Création de l'application : transformation des demandes

Créez une classe nommée `SigV4Interceptor` pour intercepter les AWS demandes et signez-les à l'aide de [Signature Version 4](#). Cela sera enregistré auprès du client HTTP utilisé pour récupérer les ressources cartographiques lors de la création de l'activité principale.

```
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()
    }
}
```

Création de l'application : activité principale

L'activité principale est chargée d'initialiser les vues qui seront affichées aux utilisateurs. Cela implique :

- Instanciation d'un Amazon Cognito. `CredentialsProvider`
- Enregistrement de l'intercepteur Signature version 4.
- Configuration de la carte en la pointant sur un style de carte, en remplaçant les URL des vignettes et en affichant l'attribution appropriée.

`MainActivity` est également chargé de transférer les événements du cycle de vie vers la vue cartographique.

```
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()
}
}
```

L'exécution de cette application affiche une carte en plein écran dans le style de votre choix. Cet exemple est disponible dans le référentiel d'exemples Amazon Location Service sur [GitHub](#).

Utilisation de Tangram ES pour iOS avec Amazon Location Service

[Tangram ES](#) est une bibliothèque C++ pour le rendu de cartes 2D et 3D à partir de données vectorielles à l'aide d'OpenGL ES. C'est l'équivalent natif de [Tangram](#).

Les styles Tangram conçus pour fonctionner avec le [schéma Tilezen](#) sont largement compatibles avec Amazon Location lorsque vous utilisez des cartes provenant de HERE. Il s'agit des licences suivantes :

- [Bubble Wrap](#) — Un style d'orientation complet avec des icônes utiles pour les points d'intérêt
- [Cinnabar](#) — Un look classique et incontournable pour les applications cartographiques générales
- [Refill](#) — Un style de carte minimaliste conçu pour les superpositions de visualisation de données, inspiré du style Toner emblématique de Stamen Design
- [Tron](#) — Une exploration des transformations d'échelle dans le langage visuel de TRON

- [Walkabout](#) — Un style axé sur le plein air, parfait pour la randonnée ou les sorties

Ce guide explique comment intégrer Tangram ES pour iOS à Amazon Location en utilisant le style Tangram appelé Cinnabar. Cet exemple est disponible dans le référentiel d'exemples Amazon Location Service sur [GitHub](#).

Bien que les autres styles Tangram soient mieux accompagnés de tuiles matricielles, qui encodent les informations du terrain, cette fonctionnalité n'est pas encore prise en charge par Amazon Location.

Important

Les styles Tangram présentés dans le didacticiel suivant sont uniquement compatibles avec les ressources cartographiques Amazon Location configurées avec ce `VectorHereContrast` style.

Création de l'application : initialisation

Pour initialiser l'application, procédez comme suit :

1. Créez un nouveau projet Xcode à partir du modèle d'application.
2. Sélectionnez SwiftUI pour son interface.
3. Sélectionnez l'application SwiftUI pour son cycle de vie.
4. Sélectionnez Swift pour sa langue.

Création de l'application : ajout de dépendances

Pour ajouter des dépendances, vous pouvez utiliser un gestionnaire de dépendances tel que [CocoaPods](#):

1. Dans votre terminal, installez CocoaPods :

```
sudo gem install cocoapods
```

2. Accédez au répertoire du projet de votre application et initialisez le Podfile avec le gestionnaire de CocoaPods packages :

```
pod init
```

- Ouvrez le Podfile pour ajouter AWSCore et en Tangram-es tant que dépendances :

```
platform :ios, '12.0'  
  
target 'Amazon Location Service Demo' do  
  use_frameworks!  
  
  pod 'AWSCore'  
  pod 'Tangram-es'  
end
```

- Téléchargez et installez les dépendances :

```
pod install --repo-update
```

- Ouvrez l'espace de travail Xcode qui CocoaPods a créé :

```
xed .
```

Création de l'application : Configuration

Ajoutez les clés et valeurs suivantes à Info.plist pour configurer l'application et désactiver la télémétrie :

Clé	Valeur
AWSRegion	us-east-1
IdentityPoolId	us-east- 1:54 f2ba88-9390-498d-aaa5-0d97f b7ca3bd
MapName	ExampleMap
URL de la scène	https://www.nextzen.org/carto/cinnabar-style/9/cinnabar-style.zip

Création de l'application : ContentView mise en page

Pour afficher la carte, modifiez `ContentView.swift` :

- Ajoutez un `MapView` qui affiche la carte.
- Ajoutez un `TextField` qui affiche l'attribution.

Cela définit également le point central initial de la carte.

Note

Vous devez indiquer l'attribution d'un mot ou d'un texte pour chaque fournisseur de données que vous utilisez, que ce soit sur votre application ou dans votre documentation. Les chaînes d'attribution sont incluses dans la réponse du descripteur de style sous les `source.grabmaptiles.attribution` touches `sources.esri.attribution` `sources.here.attribution`, et. Lorsque vous utilisez les ressources Amazon Location avec [des fournisseurs de données](#), assurez-vous de lire [les conditions générales du service](#).

```
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()
    }
}
```

Création de l'application : transformation des demandes

Créez un nouveau fichier Swift nommé `AWSSignatureV4URLHandler.swift` contenant la définition de classe suivante pour intercepter les AWS demandes et les signer à l'aide de [Signature Version 4](#). Cela sera enregistré en tant que gestionnaire d'URL dans le MapView Tangram.

```
import AWSCore
import TangramMap

class AWSSignatureV4URLHandler: TGDefaultURLHandler {
    private let region: AWSRegionType
    private let identityPoolId: String
    private let credentialsProvider: AWSCredentialsProvider

    init(region: AWSRegionType, identityPoolId: String) {
        self.region = region
        self.identityPoolId = identityPoolId
        self.credentialsProvider = AWSCognitoCredentialsProvider(regionType: region,
identityPoolId: identityPoolId)
        super.init()
    }

    override func downloadRequestAsync(_ url: URL, completionHandler: @escaping
TGDownloadCompletionHandler) -> UInt {
        if url.host?.contains("amazonaws.com") != true {
            // not an AWS URL
            return super.downloadRequestAsync(url, completionHandler:
completionHandler)
        }

        // URL-encode spaces, etc.
        let keyPath = String(url.path.dropFirst())
```

```
    guard let keyPathSafe =
keyPath.addingPercentEncoding(withAllowedCharacters: .urlPathAllowed) else {
        print("Invalid characters in path '\(keyPath)'; unsafe to sign")
        return super.downloadRequestAsync(url, completionHandler:
completionHandler)
    }

    // sign the URL
    let endpoint = AWSEndpoint(region: region, serviceName: "geo", url: url)
    let requestHeaders: [String: String] = ["host": endpoint!.hostName]
    let task = AWSSignatureV4Signer
        .generateQueryStringForSignatureV4(
            withCredentialProvider: credentialsProvider,
            httpMethod: .GET,
            expireDuration: 60,
            endpoint: endpoint!,
            keyPath: keyPathSafe,
            requestHeaders: requestHeaders,
            requestParameters: .none,
            signBody: true)
    task.waitUntilFinished()

    if let error = task.error as NSError? {
        print("Error occurred: \(error)")
    }

    if let result = task.result {
        // have Tangram fetch the signed URL
        return super.downloadRequestAsync(result as URL, completionHandler:
completionHandler)
    }

    // fall back to an unsigned URL
    return super.downloadRequestAsync(url, completionHandler: completionHandler)
}
}
```

Création de l'application : vue cartographique

La vue cartographique est chargée d'initialiser une instance du sous-jacent `AWSSignatureV4Delegate` et de le configurer `MGLMapView`, qui récupère les ressources et affiche la carte. Il gère également la propagation des chaînes d'attribution depuis la source du descripteur de style vers le `ContentView`.

Créez un nouveau fichier Swift nommé `MapView.swift` contenant la struct définition suivante :

```
import AWSCore
import TangramMap
import SwiftUI

struct MapView: UIViewRepresentable {
    private let mapView: TGMapView

    init() {
        let regionName = Bundle.main.object(forKey: "AWSRegion") as!
String
        let identityPoolId = Bundle.main.object(forKey: "IdentityPoolId")
as! String
        let mapName = Bundle.main.object(forKey: "MapName") as! String
        let sceneURL = URL(string: Bundle.main.object(forKey: "SceneURL")
as! String)!

        let region = (regionName as NSString).aws_regionTypeValue()

        // rewrite tile URLs to point at AWS resources
        let sceneUpdates = [
            TGSceneUpdate(path: "sources.mapzen.url",
                value: "https://maps.geo.\(regionName).amazonaws.com/maps/v0/
maps/\(mapName)/tiles/{z}/{x}/{y}")]

        // instantiate a TGURLHandler that will sign AWS requests
        let urlHandler = AWSSignatureV4URLHandler(region: region, identityPoolId:
identityPoolId)

        // instantiate the map view and attach the URL handler
        mapView = TGMapView(frame: .zero, urlHandler: urlHandler)

        // load the map style and apply scene updates (properties modified at runtime)
        mapView.loadScene(from: sceneURL, with: sceneUpdates)
    }

    func cameraPosition(_ cameraPosition: TGCameraPosition) -> MapView {
        mapView.cameraPosition = cameraPosition

        return self
    }

    // MARK: - UIViewRepresentable protocol
```

```
func makeUIView(context: Context) -> TGMMapView {
    return mapView
}

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

L'exécution de cette application affiche une carte en plein écran dans le style de votre choix. Cet exemple est disponible dans le référentiel d'exemples Amazon Location Service sur [GitHub](#).

Dessiner des entités de données sur une carte

Une fois que vous avez une application qui affiche une carte, en utilisant Amplify MapLibre ou Tangram pour afficher la carte, l'étape suivante consiste naturellement à dessiner des entités au-dessus de la carte. Par exemple, vous souhaitez peut-être afficher les emplacements de vos clients sous forme de marqueurs sur la carte.

En général, vous pouvez utiliser les [fonctions de recherche de lieux](#) pour trouver des emplacements à partir de vos données, puis utiliser les fonctionnalités d'Amplify ou de Tangram pour afficher les emplacements. MapLibre

Pour voir des exemples de rendu de différents types d'objets sur une carte, consultez les MapLibre exemples suivants :

- [Exemple : dessiner des marqueurs](#)
- [Exemple : dessiner des points groupés](#)
- [Exemple : Dessiner un polygone](#)

Pour plus d'exemples et de didacticiels, voir [Exemples de code et didacticiels pour travailler avec Amazon Location Service](#).

Définition de l'étendue d'une carte à l'aide de MapLibre

Parfois, vous ne voulez pas que vos utilisateurs puissent se déplacer ou zoomer sur le monde entier. Si vous utilisez MapLibre le contrôle cartographique, vous pouvez limiter l'étendue, ou les limites, du contrôle cartographique à l'aide de l'option `maxBounds`, et restreindre le zoom avec `minZoom` et les options `maxZoom`

L'exemple de code suivant montre comment initialiser le contrôle cartographique pour limiter le panoramique à une limite spécifique (dans ce cas, l'étendue de la source de données Grab).

Note

Ces exemples sont inclus JavaScript et fonctionnent dans le contexte du [Création d'une application Web](#) didacticiel.

```
// Set bounds to Grab data provider region
var bounds = [
    [90.0, -21.943045533438166], // Southwest coordinates
    [146.25, 31.952162238024968] // Northeast coordinates
];

var mlg1Map = new maplibregl.Map(
  {
    container: 'map',
    style: mapName,
    maxBounds: bounds // Sets bounds as max
    transformRequest,
  }
);
```

De même, vous pouvez définir un niveau de zoom minimal et maximal pour la carte. Les valeurs pour les deux peuvent être comprises entre 0 et 24, bien que les valeurs par défaut soient 0 pour le zoom minimum et 22 pour le maximum (les fournisseurs de données peuvent ne pas fournir de données à tous les niveaux de zoom). La plupart des bibliothèques de cartes gèrent cela automatiquement). L'exemple suivant initialise les `maxZoom` options `minZoom` et du contrôle MapLibre Map.

```
// Set the minimum and maximum zoom levels
var mlg1Map = new maplibregl.Map(
  {
    container: 'map',
    style: mapName,
    maxZoom: 12,
    minZoom: 5,
    transformRequest,
  }
);
```

i Tip

Le contrôle MapLibre Map permet également de définir ces options lors de l'exécution, plutôt que lors de l'initialisation, avec `get . . .` et des `set . . .` fonctions. Par exemple, utilisez `getMaxBounds` et `setMaxBounds` pour modifier les limites de la carte lors de l'exécution.

Gestion de vos ressources cartographiques

Vous pouvez gérer vos ressources cartographiques à l'aide de la console Amazon LocationAWS CLI, ou des API Amazon Location.

Lister les ressources cartographiques

Vous pouvez consulter la liste de vos ressources cartographiques à l'aide de la console Amazon LocationAWS CLI, ou des API Amazon Location.

Console

Pour consulter la liste des ressources cartographiques existantes à l'aide de la console Amazon Location

1. Ouvrez la console Amazon Location à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Choisissez Maps dans le volet de navigation de gauche.
3. Consultez la liste de vos ressources cartographiques sous Mes cartes.

API

Utilisez l'[ListMaps](#) opération depuis les API Amazon Location Maps.

L'exemple suivant est une demande d'API visant à obtenir la liste des ressources cartographiques du AWS compte.

```
POST /maps/v0/list-maps
```

Voici un exemple de réponse pour [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

Utilisez la commande [list-map](#).

L'exemple suivant permet d'obtenir une liste des ressources cartographiques du AWS compte.

```
aws location list-maps
```

Obtenir des informations détaillées sur les ressources cartographiques

Vous pouvez obtenir des informations sur n'importe quelle ressource cartographique de votre compte AWS à l'aide de la console Amazon Location, de l'AWS CLI, ou de l'API Amazon Location.

Console

Pour afficher les détails d'une ressource cartographique à l'aide de la console Amazon Location

1. Ouvrez la console Amazon Location à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Choisissez Maps dans le volet de navigation de gauche.
3. Sous Mes cartes, sélectionnez le lien du nom de la ressource cartographique cible.

API

Utilisez l'[DescribeMap](#) opération depuis les API Amazon Location Maps.

L'exemple suivant est une demande d'API pour obtenir les détails des ressources cartographiques pour *ExampleMap*.

```
GET /maps/v0/maps/ExampleMap
```

Voici un exemple de réponse pour [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

Utilisez la commande [describe-map](#).

L'exemple suivant permet d'AWS CLI d'obtenir les détails des ressources cartographiques pour *ExampleMap*.

```
aws location describe-map \
  --map-name "ExampleMap"
```

Supprimer une ressource cartographique

Vous pouvez supprimer une ressource cartographique de votre AWS compte à l'aide de la console Amazon Location, AWS CLI, ou des API Amazon Location.

Warning

Cette opération supprime définitivement la ressource.

Console

Pour supprimer une ressource cartographique existante à l'aide de la console Amazon Location

1. Ouvrez la console Amazon Location à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Choisissez Maps dans le volet de navigation de gauche.
3. Dans la liste Mes cartes, sélectionnez la carte cible dans la liste.
4. Choisissez Supprimer la carte.

API

Utilisez l'[DeleteMap](#) opération depuis les API Amazon Location Maps.

L'exemple suivant est une demande d'API pour supprimer la ressource cartographique *ExampleMap*.

```
DELETE /maps/v0/maps/ExampleMap
```

Voici un exemple de réponse positive pour [DeleteMap](#) :

```
HTTP/1.1 200
```

CLI

Utilisez la commande [delete-map](#).

L'exemple suivant est une AWS CLI commande permettant de supprimer la ressource cartographique *ExampleMap*.

```
aws location delete-map \  
  --map-name "ExampleMap"
```

Recherche de données de localisation et de géolocalisation à l'aide d'Amazon Location

Amazon Location inclut la possibilité de rechercher les données de géolocalisation, ou de lieu, du fournisseur que vous avez choisi. Plusieurs types de recherche sont disponibles.

- Géocodage — Le géocodage est le processus de recherche d'adresses, de régions, de noms commerciaux ou d'autres points d'intérêt, sur la base de la saisie de texte. Il renvoie les détails et l'emplacement (en latitude et longitude) des résultats trouvés.

- Géocodage inversé — Le géocodage inversé vous permet de trouver des lieux proches d'un emplacement donné.
- Complétion automatique — La saisie semi-automatique est le processus qui consiste à faire des suggestions automatiques au fur et à mesure que l'utilisateur saisit une requête. Par exemple, s'ils saisissent **Par** une suggestion, peut-être `Paris, France`.

Amazon Location vous permet de choisir un fournisseur de données pour les opérations de recherche de lieux en créant et en configurant une ressource d'index de lieux.

Une fois que vous avez créé votre ressource, vous pouvez envoyer des demandes à l'aide du AWS SDK correspondant à votre langue préférée, d'Amplify ou des points de terminaison de l'API REST. Vous pouvez utiliser les données de la réponse pour marquer des emplacements sur une carte, enrichir les données de position et convertir des positions en texte lisible par l'homme.

Note

Pour un aperçu des concepts relatifs aux lieux de recherche, voir [Recherche de lieux](#).

Rubriques

- [Prérequis](#)
- [Géocodage](#)
- [Géocodage inversé](#)
- [Remplissage automatique](#)
- [Utilisation des identifiants de lieu](#)
- [Catégories de lieux et filtrage des résultats](#)
- [Amazon Aurora PostgreSQL fonctions définies par l'utilisateur pour Amazon Location Service](#)
- [Gestion des ressources de votre index des places](#)

Prérequis

Avant de commencer le géocodage, le géocodage inversé ou la recherche de lieux, suivez les étapes préalables :

Rubriques

- [Création d'une ressource d'index des lieux](#)
- [Authentification de vos demandes](#)

Création d'une ressource d'index des lieux

Commencez par créer une ressource d'index des lieux dans votre AWS compte.

Lorsque vous créez une ressource d'index géographique, vous pouvez choisir parmi les fournisseurs de données disponibles pour prendre en charge les requêtes de géocodage, de géocodage inversé et de recherches :

1. Esri : pour plus d'informations sur la couverture d'Esri dans la région qui vous intéresse, consultez la section [Couverture du géocodage Esri](#) dans la documentation Esri.
2. Technologies HERE — Pour plus d'informations sur la couverture de HERE dans la région qui vous intéresse, consultez la section sur la [couverture de géocodage HERE](#) dans la documentation HERE.
3. Grab — Grab fournit des données uniquement pour l'Asie du Sud-Est. Pour plus d'informations sur la couverture de Grab, consultez [Pays/régions et zone couverts](#) ce guide.

Vous pouvez le faire à l'aide de la console Amazon Location Service AWS CLI, ou des API Amazon Location.

Console

Pour créer une ressource d'index de lieux à l'aide de la console Amazon Location Service

1. Ouvrez la console Amazon Location Service à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Dans le volet de navigation de gauche, choisissez Placer les index.
3. Choisissez Créer un index des lieux.
4. Remplissez les cases suivantes :
 - Nom — Entrez le nom de la ressource d'index des lieux. Par exemple, *ExemplePlaceIndex*. 100 caractères maximum. Les entrées valides incluent les caractères alphanumériques, les traits d'union, les points et les traits de soulignement.
 - Description — Entrez une description facultative.

5. Sous Fournisseurs de données, choisissez un [fournisseur de données](#) disponible à utiliser avec votre ressource d'index des lieux.

 Note

Si votre application assure le suivi ou le routage des ressources que vous utilisez dans votre entreprise, comme les véhicules de livraison ou les employés, vous ne devez pas utiliser Esri comme fournisseur de géolocalisation. Consultez la section 82 des [Conditions de serviceAWS](#) pour plus de détails.

6. Sous Options de stockage des données, indiquez si vous avez l'intention de stocker les résultats de recherche à partir de votre ressource d'index des lieux.
7. (Facultatif) Sous Tags (Balises), entrez des paramètres Key (Clé) et Value (Valeur) de balise. Cela ajoute une balise à votre nouvelle ressource d'index des lieux. Pour plus d'informations, veuillez consulter la rubrique [Balisage de vos ressources](#).
8. Choisissez Créer un index des lieux.

API

Pour créer une ressource d'index des lieux à l'aide des API Amazon Location

Utilisez l'[CreatePlaceIndex](#) opération depuis les API Amazon Location Places.

L'exemple suivant est une demande d'API pour créer une ressource d'index de lieux appelée à *ExemplePlaceIndex* l'aide du fournisseur de données *Esri*.

```
POST /places/v0/indexes
Content-type: application/json

{
  "DataSource": "Esri",
  "DataSourceConfiguration": {
    "IntendedUse": "SingleUse"
  },
  "Description": "string",
  "IndexName": "ExemplePlaceIndex",
  "Tags": {
    "Tag1" : "Value1"
  }
}
```

AWS CLI

Pour créer une ressource d'index de lieux à l'aide de AWS CLI commandes

Utilisez la commande [create-place-index](#).

L'exemple suivant crée une ressource d'index de lieux appelée *ExamplePlaceIndex* en utilisant *Esri* comme fournisseur de données.

```
aws location \  
  create-place-index \  
  --data-source "Esri" \  
  --description "Example place index" \  
  --index-name "ExamplePlaceIndex" \  
  --tags Tag1=Value1
```

Note

La facturation dépend de votre utilisation. L'utilisation d'autres AWS services peut entraîner des frais. Pour plus d'informations, consultez les [tarifs d'Amazon Location Service](#).

Authentification de vos demandes

Une fois que vous avez créé une ressource d'index des lieux et que vous êtes prêt à intégrer des fonctionnalités de localisation dans votre application, choisissez la manière dont vous souhaitez authentifier vos demandes :

- Pour découvrir comment accéder aux services, consultez la section [Accès à Amazon Location Service](#).
- Si vous avez un site Web avec des utilisateurs anonymes, vous pouvez utiliser des clés d'API ou Amazon Cognito.

Exemple

L'exemple suivant montre l'utilisation d'une clé d'API pour l'autorisation, à l'aide du [JavaScript SDK AWS v3](#) et de l'Amazon Location. [JavaScript Aide à l'authentification](#)

```
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);
```

Géocodage

Le géocodage est un processus qui convertit du texte, tel qu'une adresse, une région, le nom d'une entreprise ou un point d'intérêt, en un ensemble de coordonnées géographiques. Vous pouvez utiliser les ressources d'index de localisation pour envoyer des demandes de géocodage et intégrer les données extraites du géocodage afin d'afficher les données sur une carte pour votre application Web ou mobile.

Cette section explique comment envoyer une demande de géocodage simple et comment envoyer des demandes de géocodage avec des spécifications facultatives.

Géocodage

Vous pouvez envoyer une simple demande de géocodage à l'aide de l'[SearchPlaceIndexForText](#) opération de conversion d'une adresse en un ensemble de coordonnées. Une demande simple contient le paramètre obligatoire suivant :

- **Text**— Une adresse, un nom, une ville ou une région à convertir en un ensemble de coordonnées. Par exemple, la chaîne `Any Town`.

Pour spécifier un nombre maximum de résultats par page, utilisez le paramètre facultatif suivant :

- **MaxResults**— Limite le nombre maximum de résultats renvoyés dans la réponse à la requête.

Vous pouvez utiliser les AWS CLI ou les API Amazon Location.

API

L'exemple suivant est une [SearchPlaceIndexForText](#) demande visant à rechercher dans la ressource d'index des lieux une adresse, un nom, une ville ou une région appelée *Any Town*.

ExamplePlaceIndex

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json

{
  "Text": "Any Town",
  "MaxResults": 10
}
```

AWS CLI

L'exemple suivant est une [search-place-index-for-text](#) commande permettant de rechercher dans la ressource d'index des lieux une adresse, un nom, une ville ou une région appelée *Any Town*. *ExamplePlaceIndex*

```
aws location \
  search-place-index-for-text \
    --index-name ExamplePlaceIndex \
    --text "Any Town" \
    --max-results 10
```

Géocoder à proximité d'une position

Lors du géocodage, vous pouvez géocoder à proximité d'une position donnée avec le paramètre facultatif suivant :

- **BiasPosition**— La position que vous souhaitez rechercher à proximité. Cela permet d'affiner votre recherche en recherchant les résultats les plus proches de la position donnée. Défini comme [longitude, latitude]

L'exemple suivant est une [SearchPlaceIndexForText](#) demande visant à rechercher dans la ressource d'index des lieux une adresse, un nom, une ville ou une région intitulée *Any Town near the position* [-123.4567, 45.6789].

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json

{
  "Text": "Any Town",
  "BiasPosition": [-123.4567, 45.6789]
}
```

Géocodage dans un cadre de délimitation

Vous pouvez géocoder dans un cadre de délimitation pour affiner vos résultats aux coordonnées situées à l'intérieur d'une limite donnée à l'aide du paramètre facultatif suivant :

- **FilterBBox**— Un cadre de délimitation que vous spécifiez pour filtrer vos résultats en fonction des coordonnées situées dans les limites du cadre. Défini comme [LongitudeSW, LatitudeSW, LongitudeNE, LatitudeNE]

Note

Une demande ne peut pas contenir à la fois les **BiasPosition** paramètres **FilterBBox** et. La spécification des deux paramètres dans la demande renvoie une **ValidationException** erreur.

L'exemple suivant est une [SearchPlaceIndexForText](#) demande de recherche dans un cadre contenant une adresse, un nom, une ville ou une région appelée *Any Town*. Le cadre de délimitation indique que :

- La longitude du coin sud-ouest est *-124,1450*.

- La latitude du coin sud-ouest est **41.7045**.
- La longitude du coin nord-est est **-124.1387**.
- La latitude du coin nord-est est **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
  ]
}
```

Géocoder au sein d'un pays

Vous pouvez géocoder dans un ou plusieurs pays que vous spécifiez en utilisant le paramètre facultatif suivant :

- **FilterCountries**— Le pays ou la région dans lequel vous souhaitez géocoder. Vous pouvez définir jusqu'à 100 pays dans une seule demande à l'aide d'un code de pays à trois lettres [ISO 3166](#). Par exemple, utilisez AUS pour l'Australie.

L'exemple suivant est une [SearchPlaceIndexForText](#) demande de recherche d'une adresse, d'un nom, d'une ville ou d'une région intitulée « *N'importe quelle ville* » en Allemagne et en France.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json
```

```
{
  "Text": "Any Town",
  "FilterCountries": ["DEU","FRA"]
}
```

Filtrer par catégorie

Vous pouvez filtrer les catégories renvoyées dans votre demande de géocodage à l'aide du paramètre facultatif suivant :

- `FilterCategories`— Les catégories de résultats que vous souhaitez voir apparaître dans votre requête. Vous pouvez spécifier jusqu'à 5 catégories dans une seule demande. Vous trouverez la liste des catégories d'Amazon Location Service dans la section [Catégories](#). Par exemple, vous pouvez spécifier `Hotel` de ne spécifier que les hôtels récurrents dans votre requête.

L'exemple suivant est une [SearchPlaceIndexForText](#) demande de recherche d'un café appelé *Hometown Coffee* aux États-Unis.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json

{
  "Text": "Hometown Coffee",
  "FilterCategories": ["Coffee Shop"],
  "FilterCountries": ["USA"]
}
```

Pour plus de détails sur le filtrage par catégories, voir [Catégories de lieux et filtrage des résultats](#)

Géocodage dans la langue de votre choix

Vous pouvez définir une préférence linguistique pour les résultats de votre recherche à l'aide du `Language` paramètre facultatif. Par exemple, une recherche **100 Main St, Anytown, USA** peut être renvoyée `100 Main St, Any Town, USA` par défaut. Mais si vous le sélectionnez `Language, fr` les résultats peuvent être renvoyés à la `100 Rue Principale, Any Town, États-Unis place`.

- `Language`— Un code de langue à utiliser pour afficher les résultats de votre requête. La valeur doit être un code de langue [BCP 47](#) valide. Par exemple, `en` pour l'anglais.

Note

Si Language ce n'est pas spécifié ou si la langue spécifiée n'est pas prise en charge pour un résultat, la langue par défaut du partenaire pour ce résultat sera utilisée.

L'exemple suivant est une `SearchPlaceIndexForText` demande de recherche d'un lieu appelé **Any Town** avec la langue préférée spécifiée comme `de`.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json
{
  "Text": "Any Town",
  "Language": "de"
}
```

Exemple de réponse

Exemple

Voici un exemple de réponse lorsque vous appelez l'[SearchPlaceIndexForText](#) opération depuis les API Amazon Location Places. Les résultats incluent les [lieux](#) pertinents et le [résumé](#) de la demande. Deux réponses sont affichées, selon la sélection d'Esri ou HERE comme partenaire.

Exemple request

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json

{
  "Text": "Amazon",
  "MaxResults": 1,
  "FilterCountries": ["USA"],
  "BiasPosition": [-112.10, 46.32]
}
```

Exemple response (Esri)

```
{
  "Results": [
    {
```

```
    "Place": {
      "Country": "USA",
      "Geometry": {
        "Point": [
          -112.10667999999998,
          46.319090000000074
        ]
      },
      "Interpolated": false,
      "Label": "Amazon, MT, USA",
      "Municipality": "Amazon",
      "Region": "Montana",
      "SubRegion": "Jefferson County"
    },
    "Distance": 523.4619749879726,
    "Relevance": 1
  }
],
"Summary": {
  "BiasPosition": [
    -112.1,
    46.32
  ],
  "DataSource": "Esri",
  "FilterCountries": [
    "USA"
  ],
  "MaxResults": 1,
  "ResultBBox": [
    -112.10667999999998,
    46.319090000000074,
    -112.10667999999998,
    46.319090000000074
  ],
  "Text": "Amazon"
}
}
```

Example response (HERE)

```
{
  "Summary": {
    "Text": "Amazon",
```

```

    "BiasPosition": [
      -112.1,
      46.32
    ],
    "FilterCountries": [
      "USA"
    ],
    "MaxResults": 1,
    "ResultBBox": [
      -112.10668,
      46.31909,
      -112.10668,
      46.31909
    ],
    "DataSource": "Here"
  },
  "Results": [
    {
      "Place": {
        "Label": "Amazon, Jefferson City, MT, United States",
        "Geometry": {
          "Point": [
            -112.10668,
            46.31909
          ]
        },
        "Neighborhood": "Amazon",
        "Municipality": "Jefferson City",
        "SubRegion": "Jefferson",
        "Region": "Montana",
        "Country": "USA",
        "Interpolated": false,
        "TimeZone": {
          "Name": "America/Denver",
          "Offset": -25200
        }
      },
      "PlaceId": "AQAAAIAADsn2T3KdrRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqdlJJZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hV0_BUPgP7SFoWAI8BW2v7LvAjQ5NfUPy7a1v9a
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ",
      "Distance":
523.4619749905755
    }
  ]

```

```
}
```

Géocodage inversé

Le géocodage inversé est un processus qui convertit un ensemble de coordonnées en texte significatif, tel qu'une adresse, une région, le nom d'une entreprise ou un point d'intérêt. Vous pouvez utiliser les ressources d'index de localisation pour envoyer des demandes de géocodage inversé et intégrer les données extraites du géocodage inversé afin d'afficher les données sur une carte pour votre application Web ou mobile.

Cette section explique comment envoyer une simple demande de géocodage inversé.

Géocodage inversé

Vous pouvez envoyer une simple demande pour inverser le géocodage d'un ensemble de coordonnées et les convertir en une adresse significative, un point d'intérêt ou un emplacement général sans adresse à l'aide de cette [SearchPlaceIndexForPosition](#) opération. Une demande simple contient le paramètre obligatoire suivant :

- **Position**— Ensemble de coordonnées que vous souhaitez convertir en adresse, point d'intérêt ou emplacement général. Défini à l'aide du format `[longitude, latitude]`.

Pour spécifier un nombre maximum de résultats par page, ajoutez le paramètre facultatif suivant :

- **MaxResults**— Limite le nombre maximum de résultats renvoyés dans la réponse à la requête.

Si vous souhaitez spécifier une langue préférée pour les résultats de votre requête, utilisez le paramètre facultatif suivant :

- **Language**— Code de langue à utiliser pour le rendu des résultats. La valeur doit être un code de langue [BCP 47](#) valide. Par exemple, en pour l'anglais.

Note

Si **Language** ce n'est pas spécifié ou si la langue spécifiée n'est pas prise en charge pour un résultat, la langue par défaut du partenaire pour ce résultat sera utilisée.

Vous pouvez utiliser les AWS CLI ou les API Amazon Location.

API

L'exemple suivant est une [SearchPlaceIndexForPosition](#) demande visant à rechercher dans la ressource d'index des lieux une adresse `ExamplePlaceIndex`, un point d'intérêt ou un emplacement général significatif à proximité de la position `[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

L'exemple suivant est une [search-place-index-for-position](#) commande permettant de rechercher dans la ressource d'index des lieux une adresse `ExamplePlaceIndex`, un point d'intérêt ou un emplacement général significatif à proximité de la position `[122.3394, 47.6159]`.

```
aws location \
  search-place-index-for-position \
    --index-name ExamplePlaceIndex \
    --position -122.3394 47.6159 \
    --max-results 5 \
    --language de
```

Exemple de réponse

Exemple

Voici un exemple de réponse lorsque vous appelez l'[SearchPlaceIndexForPosition](#) opération depuis les API Amazon Location Places. Les résultats renvoient les [lieux](#) pertinents et le [résumé](#) de la demande. Deux réponses sont affichées, selon la sélection d'Esri ou Here comme partenaire.

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": "AQAIIAAdsn2T3KdRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqd1JZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hVO_BUPgP7SFoWAI8BW2v7LvAjQ5NfUPy7a1v9a
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ",
      "Distance": 1.868586125090601
    }
  ]
}
```

```
}
```

Remplissage automatique

La saisie automatique fournit des commentaires réactifs aux utilisateurs finaux lorsqu'ils saisissent leur requête de recherche. Il fournit des suggestions d'adresses et de points d'intérêt basées sur du texte libre partiel ou mal orthographié. Vous pouvez utiliser les ressources d'index de localisation pour demander des suggestions de saisie semi-automatique et afficher les suggestions qui en résultent dans votre application.

Amazon Location ne prend pas en charge le stockage des suggestions de saisie semi-automatique. Une erreur est renvoyée si l'index de localisation utilisé pour un appel de saisie semi-automatique est configuré pour être utilisé avec des géocodes stockés. Pour utiliser les géocodes enregistrés et demander des suggestions, créez et configurez plusieurs index de lieux.

Cette section décrit comment envoyer une demande de saisie semi-automatique. Il commence par la forme la plus élémentaire de la demande, puis présente les paramètres facultatifs que vous pouvez utiliser pour améliorer la pertinence des résultats de recherche à saisie semi-automatique.

Utilisation de la saisie semi-automatique

Vous pouvez envoyer une simple demande de suggestions de saisie automatique en utilisant cette [SearchPlaceIndexForSuggestions](#) opération. La forme la plus simple de la demande comporte un seul paramètre obligatoire, la requête `Text` :

- `Text`— Le texte partiel de forme libre à utiliser pour générer des suggestions de lieux. Par exemple, la chaîne `eiffel tow`.

Pour limiter le nombre de résultats renvoyés, ajoutez le `MaxResults` paramètre facultatif :

- `MaxResults`— Limite le nombre de résultats renvoyés dans la réponse à la requête.

Vous pouvez utiliser les API Amazon Location ou le AWS CLI.

API

L'exemple suivant est une [SearchPlaceIndexForSuggestions](#) demande visant à rechercher dans la ressource d'index des lieux *ExamplePlaceIndex*, jusqu'à 5 suggestions basées sur le nom de lieu partiel *kamp*.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/suggestions
Content-type: application/json

{
  "Text": "kamp",
  "MaxResults": 5
}
```

AWS CLI

L'exemple suivant est une [search-place-index-for-suggestions](#) commande permettant de rechercher dans la ressource d'index de lieux jusqu'à 5 suggestions basées sur le nom de lieu partiel *kamp*. *ExamplePlaceIndex*

```
aws location \
  search-place-index-for-suggestions \
  --index-name ExamplePlaceIndex \
  --text kamp \
  --max-results 5
```

L'appel à SearchPlaceIndexForSuggestions aboutit à une liste de lieux avec un nom et un identifiant pour chacun. Vous pouvez utiliser ces résultats pour suggérer ce que l'utilisateur est susceptible de rechercher pendant qu'il tape, par exemple en fournissant une liste déroulante de choix sous une zone de texte. Par exemple, voici les résultats des suggestions, basées sur la saisie par un utilisateur de *kamp*.

```
{
  "Summary": {
    "Text": "kamp",
    "MaxResults": 5,
    "DataSource": "Esri"
  },
  "Results": [
    {
```

```

        "Text": "Kampuchea",
        "PlaceId": "AQAAAIAADsn2T3KdrRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqd1JZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hV0_BUPgP7SFoWai8BW2v7LvAjQ5NfUPy7a1v9ajT3
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ"
    },
    {
        "Text": "Kampoul, Kabul, AFG",
        "PlaceId":
"AQAAAIAAA1mx1_-9ffzXD07rBgo9fh6E01Pd1YKvuT5rz2qBDxqBkhTlgkei0PR2s5sa3YBLxUqQI8bhymyYcu9R-
DkX3L9QSi3CB5LhNPu160iSFJo6H8S1CrX03QsJALhrr9mdbg0R4R4YDywkHkeBlbn7g5C5LI_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":
"AQAAAIAAA4HLQHdjUDcaaXLE9wtNIT1cjQYLgkbnMoG2eNN0AaQ8PJoWabLRXmmPUaAj8MAD6vT0i6zqaun5Mixyj7vnYX
    }
}

```

La section suivante explique comment utiliser les résultats PlaceID de ces résultats.

Utilisation des résultats de saisie semi-automatique

L'appel à `SearchPlaceIndexForSuggestions` aboutit à une liste de lieux avec un nom et un identifiant pour chacun. Vous pouvez utiliser ces résultats pour suggérer ce que l'utilisateur est susceptible de rechercher pendant qu'il tape, par exemple en fournissant une liste déroulante de choix sous une zone de texte. Lorsque l'utilisateur choisit l'un des résultats, vous pouvez appeler l'[GetPlace](#) opération avec l'identifiant de sa sélection pour renvoyer les détails de ce lieu, notamment son emplacement, son adresse ou d'autres informations.

Note

A n'PlaceId est valide que si tous les éléments suivants sont identiques dans la demande de recherche initiale et dans l'appel à `GetPlace`.

- Client Compte AWS
- Région AWS
- Fournisseur de données spécifié dans la ressource d'index des lieux

Généralement, vous les utilisez `GetPlace` avec les API Amazon Location. L'exemple suivant est une [GetPlace](#) demande visant à rechercher l'une des suggestions de la section précédente. Cet exemple est basé sur le nom de lieu partiel *kamp*.

```
POST /places/v0/indexes/ExamplePlaceIndex/  
places/AQAAAIAADsn2T3KdrRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-  
o3nqdlJZAdgcT2oWi1w9pS4wXX0k301vsKLGsPyHjV4EJxsu289i3hV0_BUPgP7SFoWAI8BW2v7LvAjQ5NfUPy7a1v9ajT3  
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ
```

Complétion automatique à proximité d'une position

Lorsque vous recherchez des suggestions de lieux à saisie automatique à l'aide de [SearchPlaceIndexForSuggestions](#), vous pouvez obtenir des suggestions plus pertinentes au niveau local en ajoutant le paramètre facultatif suivant :

- **BiasPosition**— La position que vous souhaitez rechercher à proximité. Défini comme `[longitude, latitude]`.

L'exemple suivant utilise une [SearchPlaceIndexForSuggestions](#) requête pour rechercher dans la ressource d'index des `ExamplePlaceIndex` lieux des suggestions de lieux correspondant à la requête partielle `kamp` proche de la position `[32.5827, 0.3169]`.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/suggestions  
Content-type: application/json  
  
{  
  "Text": "kamp",
```

```
"BiasPosition": [32.5827,0.3169]
}
```

Les suggestions renvoyées pour la même chose Text peuvent être différentes si une autre BiasPosition est choisie, par exemple [-96.7977, 32.7776].

Complétion automatique dans un cadre de sélection

Vous pouvez affiner votre recherche semi-automatique pour ne recevoir que des suggestions de lieux situés à l'intérieur d'une limite donnée en ajoutant le paramètre facultatif suivant :

- **FilterBBBox**— Un cadre de délimitation que vous spécifiez pour filtrer vos résultats en fonction des coordonnées situées dans les limites du cadre. Défini comme [LongitudeSW, LatitudeSW, LongitudeNE, LatitudeNE]

Note

Une demande ne peut pas contenir à la fois les BiasPosition paramètres FilterBBBox et. La spécification des deux paramètres dans la demande renvoie une ValidationException erreur.

L'exemple suivant utilise une [SearchPlaceIndexForSuggestions](#) requête pour rechercher dans la ressource d'index des *ExamplePlaceIndex* lieux des suggestions de lieux correspondant à la requête partielle *kamp*, et qui sont contenues dans le cadre de délimitation où :

- La longitude du coin sud-ouest du cadre de délimitation est de *32,5020*.
- La latitude du coin sud-ouest du cadre de délimitation est de *0,2678*.
- La longitude du coin nord-est du cadre de délimitation est de *32,6129*.
- La latitude du coin nord-est du cadre de délimitation est de *0,3502*.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/suggestions
Content-type: application/json

{
  "Text": "kamp",
  "FilterBBBox": [
    32.5020, 0.2678,
```

```

    32.6129, 0.3502
  ]
}

```

Les suggestions renvoyées pour la même chose `Text` sont différentes si une autre `FilterBBBox` est choisie, par exemple `[-97.9651, 32.0640, -95.1196, 34.0436]`.

Complétion automatique au sein d'un pays

Vous pouvez affiner votre recherche semi-automatique pour ne recevoir que des suggestions de lieux situés dans un pays donné, ou un ensemble de pays, en ajoutant le paramètre facultatif suivant :

- `FilterCountries`— Les pays dans lesquels vous souhaitez rechercher des suggestions de lieux. Vous pouvez spécifier jusqu'à 100 pays dans une seule demande en utilisant un code de pays à trois lettres [ISO 3166](#). Par exemple, utilisez `AUS` pour l'Australie.

L'exemple suivant utilise une [SearchPlaceIndexForSuggestions](#) requête pour rechercher dans la ressource d'index des `ExemplePlaceIndex` lieux des suggestions de lieux correspondant à la requête partielle `kamp` et qui sont contenues en Ouganda, au Kenya ou en Tanzanie :

```

POST /places/v0/indexes/ExamplePlaceIndex/search/suggestions
Content-type: application/json

{
  "Text": "kamp",
  "FilterCountries": ["UGA", "KEN", "TZA"]
}

```

Les suggestions renvoyées pour la même chose `Text` sont différentes si une autre `FilterCountries` liste est choisie, telle que `[» USA «]`.

Exemple de réponse

Voici un exemple de réponse contenant des suggestions de complétion automatique pour l'[SearchPlaceIndexForSuggestions](#) opération, à l'aide du texte `kamp`.

```

{
  "Summary": {

```

```

    "Text": "kamp",
    "MaxResults": 5,
    "DataSource": "Esri"
  },
  "Results": [
    {
      "Text": "Kampuchea",
      "PlaceId": "AQAAAAIAADsn2T3KdrRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqdl1JZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hV0_BUPgP7SFoWAI8BW2v7LvAjQ5NfUPy7a1v9ajT3
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ"
    },
    {
      "Text": "Kampoul, Kabul, AFG",
      "PlaceId":
"AQAAAAIAAA1mx1_-9ffzXD07rBgo9fh6E01Pd1YKvuT5rz2qBDxqBkhTlgkeiOPR2s5sa3YBLxUqQI8bhymSYcu9R-
DkX3L9QSi3CB5LhNPu160iSFJo6H8S1CrX03QsJALhrr9mdbg0R4R4YDywkHkeBlbnbn7g5C5LI_wYx873WeQZuilwtsGm8j
UeXcb_bg"
    },
    {
      "Text": "Kampala, UGA",
      "PlaceId":
"AQAAAAIAAzZfZt3qMrUkG0byhP6MM0pqy2L8SULLVWT7a3ertLBRS6Q5n7I4s9D7E0nRHADAj7mL7kvX1Q8HD-
mpuiATXNJ1Ix4_V_1B15zHe8j1YKMWvXbgb08cMpgR2fqYqZMR1x-
dfB0080oqujKZldvPIDK1kNe3GwcaqvMWWPMeaGd203brFynubAe-MmFF-Gjz-WBMfUy9og6MV7bkk6NGCA"
    },
    {
      "Text": "Kampar, Riau, IDN",
      "PlaceId": "AQAAAAIAAvbXXx-
sr0i111tH0kPdao0GF7WQ_KaZ444SEnevycp6Gtf_2JWgPfCE5bIQCYwya1uZQpX2a8YJoFm2K7Co14fLu7IK0yY0LhZx4k
",
    },
    {
      "Text": "Kampung Pasir Gudang Baru, Johor, MYS",
      "PlaceId":
"AQAAAAIAA4HLQHdjUDcaaXLE9wtNIT1cjQYLgkbnMoG2eNN0AaQ8PJoWabLRXmmPUaAj8MAD6vT0i6zqaun5Mixyj7vnYX
"
    }
  ]
}

```

Utilisation des identifiants de lieu

La recherche de lieux renvoie une liste de résultats. La plupart des résultats incluent un `PlaceId` pour ce résultat. Vous pouvez utiliser une `PlaceId` [GetPlace](#) opération pour renvoyer les informations relatives à ce lieu (y compris le nom, l'adresse, le lieu ou d'autres informations).

Note

L'utilisation [SearchPlaceIndexForSuggestions](#) renverra PlaceId des résultats pour tous les index de lieux créés avec n'importe quelle source de données. L'utilisation de [SearchPlaceIndexForText](#) PlaceId ne [SearchPlaceIndexForPosition](#) renverra a que si la source de données utilisée est ICI.

Chacun définit de PlaceId manière unique le lieu auquel il fait référence, mais un même lieu peut en avoir plusieurs PlaceId au fil du temps, en fonction du contexte. Les règles suivantes décrivent le caractère unique et la longévité d'unPlaceId.

- Les appels PlaceId renvoyés que vous passez sont spécifiques à vous Compte AWS, à la AWS région et au fournisseur de données de votre PlaceIndex ressource. GetPlace ne trouvera de résultats que lorsque ces trois attributs correspondent à l'appel d'origine qui a créé lePlaceId.
- Le PlaceId nom d'un lieu change lorsque les données relatives à ce lieu changent. Par exemple, lorsque l'entreprise à laquelle il fait référence déménage ou change de nom.
- Le PlaceId résultat d'un appel de recherche répété peut changer lorsque le service principal effectue une mise à jour. Le plus ancien PlaceId continuera d'être trouvé, mais les nouveaux appels à rechercher peuvent renvoyer un identifiant différent.

PlaceIdC'est une chaîne. Il n'y a pas de limite spécifique à la longueur d'unPlaceId. Voici un exemple de code validePlaceId.

```
AQAAAIAADsn2T3KdRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-  
o3nqd1JZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hV0_BUPgP7SFoWAI8BW2v7LvAjQ5NfUPy7a1v9ajT3  
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ
```

Le fait GetPlace d'appeler un PlaceId établissement dont les données ont changé (par exemple, un établissement qui a fermé ses portes) entraînera une 404 ResourceNotFound erreur. Un appel GetPlace avec un PlaceId nom non valide ou hors contexte, par exemple depuis un autre Compte AWS, renverra une 400 ValidationException erreur.

Bien que vous puissiez utiliser PlaceID dans les demandes suivantes, PlaceID n'est pas destiné à être un identifiant permanent et l'ID peut changer entre les appels d'API consécutifs. Veuillez consulter le comportement PlaceID suivant pour chaque fournisseur de données :

- Esri : Les identifiants de lieux changeront au moins tous les trimestres. La période typique pour ces changements serait mars, juin, septembre et décembre. Les identifiants de lieu peuvent également changer entre les changements trimestriels habituels, mais cela sera beaucoup moins fréquent.
- ICI : Nous vous recommandons de mettre les données en cache pendant une semaine au maximum afin de les maintenir à jour. Vous pouvez supposer que moins de 1 % des changements d'identifiant seront publiés au fil des versions, soit environ 1 à 2 fois par semaine.
- Grab : Les identifiants de lieu peuvent expirer ou devenir invalides dans les situations suivantes.
 - Opérations sur les données : les POI peuvent être supprimés de la base de données de Grab Map Ops sur la base de données de base, par exemple s'ils ont été fermés dans le monde réel, ont été détectés comme doublons de POI ou contiennent des informations incorrectes. Grab synchronisera les données avec l'environnement Waypoint chaque semaine.
 - POI interpolé : le POI interpolé est un POI temporaire généré en temps réel lors du traitement d'une demande, et il sera marqué comme dérivé dans le champ de la `place.result_type` réponse. Les informations des POI interpolés seront conservées pendant au moins 30 jours, ce qui signifie que dans les 30 jours, vous pourrez obtenir les détails des POI par ID de lieu à partir de l'API Place Details. Après 30 jours, les POI interpolés (identifiant de lieu et informations) peuvent expirer et être inaccessibles depuis l'API Places Details.

Catégories de lieux et filtrage des résultats

Les lieux sont classés par catégories. Si vous recherchez une entreprise, il peut s'agir d'une entreprise `Restaurant`, par exemple. Même les résultats d'une recherche d'adresse peuvent être classés selon qu'ils correspondent à une adresse, à une rue ou à une intersection.

De manière générale, Amazon Location Service classe les lieux en types de lieux. Les points d'intérêt sont ensuite classés en types de points d'intérêt.

Note

Tous les résultats ne comporteront pas de catégories.

Vous pouvez utiliser les catégories pour filtrer vos recherches de géocodage.

Filtrage des résultats

Lorsque vous utilisez `SearchPlaceIndexForText`, vous pouvez filtrer les résultats renvoyés en fonction des catégories que vous souhaitez utiliser. Par exemple :

- Si vous souhaitez rechercher un restaurant appelé « Hometown Coffee » et ne renvoyer que des résultats classés dans la catégorie des cafés, vous pouvez le faire en appelant `SearchPlaceIndexForText` et en incluant la catégorie Point d'intérêt Coffee Shop dans le `FilterCategories` paramètre.
- Lorsque vous recherchez « 123 Main St, Anytown, WA, 98123, États-Unis », vous pouvez filtrer les résultats uniquement en fonction des adresses, afin de ne pas obtenir de correspondance, par exemple, sur le code postal. Filtrez uniquement les adresses en incluant le type de lieu `AddressType` dans le `FilterCategories` paramètre.

Note

Tous les fournisseurs de données ne prennent pas en charge le filtrage, ou ne le prennent pas en charge de la même manière. Pour de plus amples informations, veuillez consulter [Limites de filtrage par fournisseur de données](#).

La section suivante répertorie les catégories sur lesquelles vous pouvez filtrer.

Catégories

Les listes suivantes indiquent les catégories utilisées par Amazon Location Service pour classer et filtrer. Ces catégories sont utilisées dans toutes les langues, indépendamment du paramètre de langue défini sur une langue différente.

Note

Amazon Location Service associe les catégories de fournisseurs de données à cet ensemble de catégories. Si un fournisseur de données place une place dans une catégorie qui ne figure pas dans la liste des catégories Amazon Location Service, la catégorie du fournisseur sera incluse dans les résultats en tant que catégorie supplémentaire.

Types de lieux — Ces types sont utilisés pour indiquer le type de correspondance utilisé pour trouver le résultat.

- `AddressType`— Renvoyé lorsque le résultat correspond à une adresse.
- `StreetType`— Renvoyé lorsque le résultat correspond à une rue.
- `IntersectionType`— Renvoie lorsque le résultat correspond à l'intersection de deux rues.
- `PointOfInterestType`— Renvoyé lorsque le résultat correspond à un point d'intérêt, tel qu'une entreprise ou un lieu municipal.
- `CountryType`— Renvoyé lorsque le résultat était apparié à un pays ou à une région majeure.
- `RegionType`— Renvoyé lorsque le résultat correspond à une région d'un pays, tel qu'un État ou une province.
- `SubRegionType`— Renvoyé lorsque le résultat était apparié à une sous-région d'un pays, telle qu'un comté ou une zone métropolitaine.
- `MunicipalityType`— Renvoie lorsque le résultat correspond à une ville ou un village.
- `NeighborhoodType`— Renvoie lorsque le résultat correspond à un quartier ou à une zone d'une ville.
- `PostalCodeType`— Renvoyé lorsque le résultat correspond à un code postal.

Catégories de points d'intérêt — Ces catégories sont utilisées pour indiquer le type d'entreprise ou d'emplacement pour les résultats relatifs aux points d'intérêt.

- `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

Limites de filtrage par fournisseur de données

Les fournisseurs ne disposent pas tous de la même fonctionnalité de filtrage. Le tableau suivant décrit les différences.

Fournisseur	API avec prise en charge des filtres	Catégories prises en charge pour le filtrage	Valeurs de retour
Esri	<code>SearchPlaceIndexForText</code> , <code>SearchPlaceIndexForSuggestions</code>	Filtrez par type de lieu et par catégorie de points d'intérêt.	Les catégories sont renvoyées par <code>SearchPlaceIndexForText</code> et <code>SearchPlaceIndexForSuggestions</code> .

Fournisseur	API avec prise en charge des filtres	Catégories prises en charge pour le filtrage	Valeurs de retour
			<code>rPosition</code> , et <code>GetPlace</code>
Ici	<code>SearchPlaceIndexForText</code> , <code>SearchPlaceIndexForSuggestions</code>	Filtrez uniquement par type de lieu.	Les catégories sont renvoyées par <code>SearchPlaceIndexForText</code> et <code>SearchPlaceIndexForSuggestions</code> <code>SearchPlaceIndexForPosition</code> , et <code>GetPlace</code>
Saisir	non pris en charge	non pris en charge	non pris en charge
Données ouvertes	n/a (la recherche de lieux n'est pas prise en charge)	N/A	s/o

Amazon Aurora PostgreSQL fonctions définies par l'utilisateur pour Amazon Location Service

Vous pouvez utiliser Amazon Location Service pour utiliser les coordonnées et les adresses stockées dans les tables de base de données afin de nettoyer et d'enrichir vos données géospatiales.

Par exemple :

- Vous pouvez utiliser le géocodage pour convertir les adresses en coordonnées afin de normaliser et de combler les lacunes des données relatives aux adresses stockées dans une table de base de données.

- Vous pouvez géocoder des adresses pour obtenir leur position et utiliser les coordonnées avec des fonctions spatiales de base de données, telles qu'une fonction qui affiche des lignes dans une zone spécifiée.
- Vous pouvez utiliser des données enrichies pour générer des rapports automatisés, tels que la génération d'un rapport automatique illustrant tous les appareils d'une zone donnée, ou un rapport automatisé pour l'apprentissage automatique qui illustre les zones présentant des taux d'échec plus élevés lors de l'envoi de mises à jour de localisation.

Ce didacticiel explique comment formater et enrichir les adresses stockées dans une table de Amazon Aurora PostgreSQL base de données à l'aide d'Amazon Location Service.

- Amazon Aurora PostgreSQL— Un moteur de base de données relationnelle entièrement géré, compatible avec MySQL et PostgreSQL, qui produit un débit jusqu'à cinq fois supérieur à celui de MySQL et jusqu'à trois fois supérieur à celui de PostgreSQL sans modifier la plupart de vos applications existantes. Pour plus d'informations, consultez [Qu'est-ce qu'Amazon Aurora ?](#) dans le guide de l'utilisateur Amazon Aurora.

Important

L'application qui en résulte dans ce didacticiel utilise un index de localisation qui stocke les résultats du géocodage. Pour plus d'informations sur les frais applicables au stockage des résultats de géocodage, consultez la [tarification d'Amazon Location Service](#).

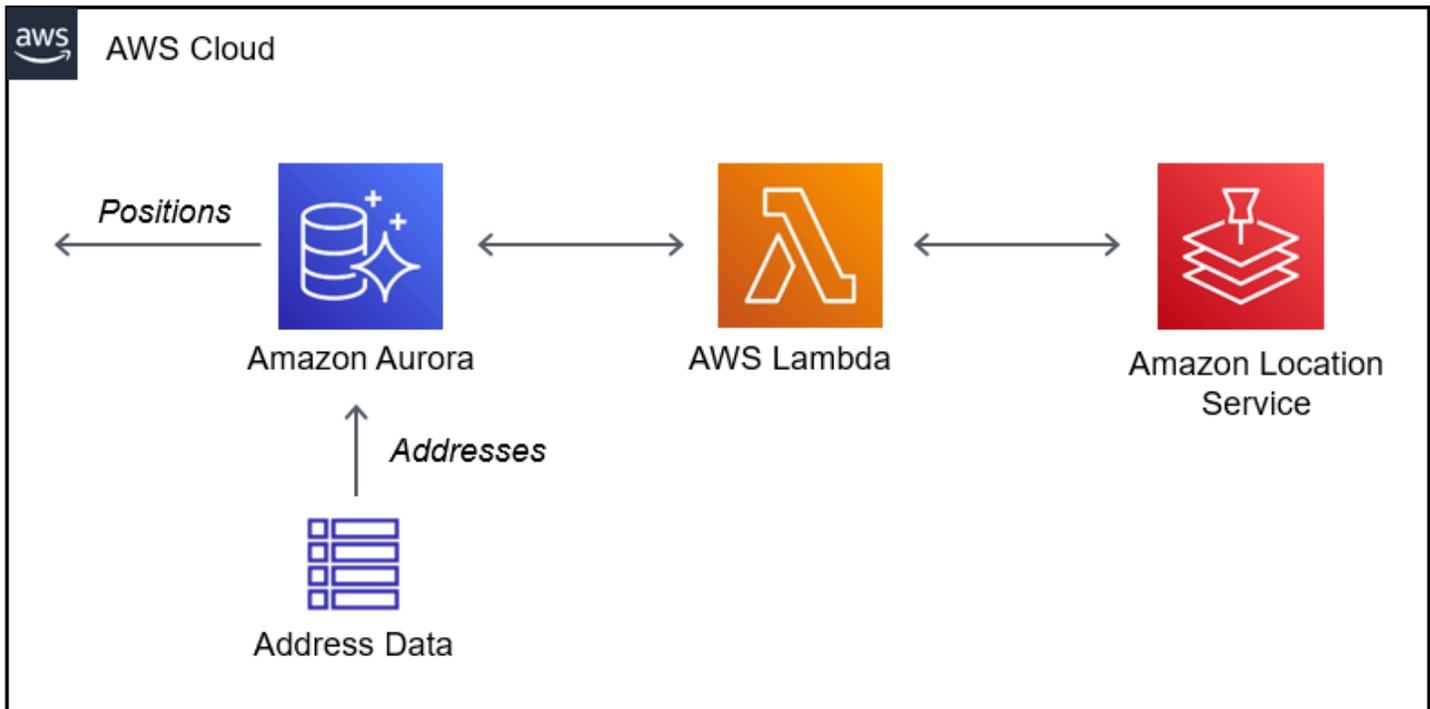
Un exemple de code est disponible dans le référentiel d'exemples d'Amazon Location Service sur [GitHub](#), qui inclut [un AWS CloudFormation modèle](#).

Rubriques

- [Présentation](#)
- [Prérequis](#)
- [Démarrage rapide](#)
- [Création d'une ressource d'index des lieux](#)
- [Création d'une AWS Lambda fonction pour le géocodage](#)
- [Accorder Amazon Aurora PostgreSQL l'accès à AWS Lambda](#)
- [Appeler la fonction AWS Lambda](#)

- [Enrichir une base de données contenant des données d'adresses](#)
- [Étapes suivantes](#)

Présentation



L'architecture implique les intégrations suivantes :

- Cette solution utilise une ressource d'index de localisation Amazon Location pour prendre en charge les requêtes de géocodage à l'aide de l'opération `SearchPlaceIndexForText`.
- AWS Lambda utilise un Lambda Python qui géocode les adresses lorsqu'une politique IAM autorise l'AWS Lambda à appeler l'opération de géocodage Amazon Location, `SearchPlaceIndexForText`.
- Accordez l'autorisation d'Amazon Aurora PostgreSQL d'invoquer la fonction Lambda de géocodage à l'aide d'une fonction SQL définie par l'utilisateur.

Prérequis

Avant de commencer, vous devez remplir les conditions préalables suivantes :

- Un Amazon Aurora PostgreSQL cluster. Pour plus d'informations sur la [création d'un cluster de base de données Amazon Aurora](#), consultez le guide de l'utilisateur Amazon Aurora.

Note

Si votre cluster Amazon Aurora n'est pas accessible au public, vous devez également configurer Amazon Aurora pour qu'il se connecte AWS Lambda à un cloud privé virtuel (VPC) sur votre AWS compte. Pour plus d'informations, consultez [Accorder Amazon Aurora PostgreSQL l'accès à AWS Lambda](#).

- Un outil de développement SQL pour se connecter au Amazon Aurora PostgreSQL cluster.

Démarrage rapide

Au lieu de suivre les étapes décrites dans ce didacticiel, vous pouvez lancer une pile rapide pour déployer une AWS Lambda fonction prenant en charge le fonctionnement d'Amazon Location [SearchPlaceIndexForText](#). Cela configure automatiquement votre AWS compte pour autoriser Amazon Aurora à appeler AWS Lambda.

Une fois que vous aurez configuré votre AWS compte, vous devrez :

- Ajoutez la fonctionnalité Lambda à Amazon Aurora. Consultez la section Ajouter le rôle IAM à un cluster de base de données Amazon Aurora dans [Accorder Amazon Aurora PostgreSQL l'accès à AWS Lambda](#).
- Chargez la fonction définie par l'utilisateur dans votre base de données. veuillez consulter [Appeler la fonction AWS Lambda](#).

Launch Stack 

Création d'une ressource d'index des lieux

Commencez par créer une ressource d'index de lieux pour prendre en charge les requêtes de géocodage.

1. Ouvrez la console Amazon Location Service à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Dans le volet de navigation de gauche, choisissez Placer les index.
3. Remplissez les cases suivantes :

- Nom — Entrez le nom de la ressource d'index des lieux. Par exemple, *AuroraPlaceIndex*. 100 caractères maximum. Les entrées valides incluent les caractères alphanumériques, les traits d'union, les points et les traits de soulignement.
 - Description — Entrez une description facultative. Par exemple, *index Place pour Amazon Aurora*.
4. Sous Fournisseurs de données, choisissez un [fournisseur de données](#) disponible à utiliser avec votre ressource d'index des lieux. Si vous n'avez aucune préférence, nous vous recommandons de commencer par *Esri*.
 5. Sous Options de stockage des données, spécifiez Oui, les résultats seront stockés. Cela indique que vous avez l'intention d'enregistrer les résultats du géocodage dans une base de données.
 6. (Facultatif) Sous Tags (Balises), entrez des paramètres Key (Clé) et Value (Valeur) de balise. Cela ajoute une balise à votre nouvelle ressource d'index des lieux. Pour plus d'informations, veuillez consulter la rubrique [Balisage de vos ressources](#).
 7. Choisissez Créer un index des lieux.

Création d'une AWS Lambda fonction pour le géocodage

Pour créer une connexion entre Amazon Location Service Amazon Aurora PostgreSQL et Amazon, vous avez besoin d'une AWS Lambda fonction permettant de gérer les demandes provenant du moteur de base de données. Cette fonction traduit l'événement de fonction Lambda défini par l'utilisateur et appelle l'opération Amazon Location. SearchPlaceIndexForText

Vous pouvez créer la fonction à l'aide de la AWS Lambda console AWS Command Line Interface, des API ou des AWS Lambda API.

Pour créer une fonction Lambda définie par l'utilisateur à l'aide de la console

1. Ouvrez la console AWS Lambda à l'adresse <https://console.aws.amazon.com/lambda/>.
2. Dans le menu de navigation de gauche, sélectionnez Fonctions.
3. Choisissez Créer une fonction et assurez-vous que l'option Auteur à partir de zéro est sélectionnée.
4. Remplissez les cases suivantes :
 - Nom de la fonction — Entrez un nom unique pour votre fonction. Les entrées valides incluent les caractères alphanumériques, les traits d'union et les traits de soulignement sans espaces. Par exemple, *AuroraGeocoder*.

- Runtime — Choisissez *Python 3.8*.
5. Choisissez Créer une fonction.
 6. Cliquez sur l'onglet Code pour ouvrir l'éditeur.
 7. Remplacez le code de l'espace réservé `lambda_function.py` par le code suivant :

```
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 vous avez donné à votre index de localisation un autre nom que *AuroraPlaceIndex*, créez une variable d'environnement nommée `PLACE_INDEX_NAME` pour attribuer le nom de la ressource à :
 - Dans l'onglet Configuration, sélectionnez Variables d'environnement.
 - Choisissez Modifier, puis Ajouter une variable d'environnement.
 - Pour Key : Enter `PLACE_INDEX_NAME`.
 - Pour la valeur : entrez le nom de votre ressource d'index géographique.
9. Choisissez Déployer pour enregistrer la fonction mise à jour.
10. Dans le menu déroulant Test, choisissez Configurer l'événement de test.
11. Choisissez Create new test event.
12. Entrez l'événement de test suivant :

```
{
  "text": "Baker Beach",
  "biasPosition": [-122.483, 37.790],
  "filterCountries": ["USA"]
}
```

13. Choisissez Test pour tester la fonction Lambda.
14. Cliquez sur l'onglet Configuration.
15. Sous Configuration générale : Choisissez Autorisations.
16. Sous Rôle d'exécution : choisissez le nom du rôle hyperlié pour accorder à Amazon Location Service des autorisations pour votre fonction Lambda.
17. Sous l'onglet Autorisations : sélectionnez le menu déroulant Ajouter des autorisations, puis choisissez Créer une politique intégrée.
18. Sélectionnez l'onglet JSON.
19. Ajoutez la politique IAM suivante :
 - La politique suivante autorise l'envoi `SearchPlaceIndexForText` à la ressource d'index de lieux *AuroraPlaceIndex*.

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

```
"Effect": "Allow",
  "Action": "geo:SearchPlaceIndexForText",
  "Resource": "arn:aws:geo:<Region>:<AccountId>:place-index/AuroraPlaceIndex"
}
]
}
```

20. Choisissez Examiner une politique.
21. Entrez un nom de stratégie. Par exemple, *AuroraPlaceIndexReadOnly*.
22. Choisissez Créer une politique.

Accorder Amazon Aurora PostgreSQL l'accès à AWS Lambda

Avant de Amazon Aurora PostgreSQL pouvoir invoquer une AWS Lambda fonction, vous devez accorder une autorisation d'accès.

Si votre Amazon Aurora PostgreSQL cluster n'est pas accessible au public, vous devez d'abord créer un point de terminaison VPC pour AWS Lambda qu'Amazon Aurora puisse appeler votre fonction Lambda.

Créez un point de terminaison VPC pour AWS Lambda

Note

Cette étape n'est requise que si votre Amazon Aurora PostgreSQL cluster n'est pas accessible au public.

1. Ouvrez la [Amazon Virtual Private Cloud Console](#).
2. Dans le volet de navigation de gauche, sélectionnez Endpoints.
3. Choisissez Créer un point de terminaison.
4. Dans le filtre Nom du service, entrez « lambda », puis choisissez `com.amazonaws.<region>.lambda`.
5. Choisissez le VPC contenant votre cluster Aurora.
6. Choisissez un sous-réseau pour chaque zone de disponibilité.
7. Dans le filtre Groupe de sécurité, entrez « par défaut » ou le nom du groupe de sécurité dont votre cluster Aurora est membre, puis choisissez le groupe de sécurité.

8. Choisissez Créer un point de terminaison.

Créez une politique IAM pour autoriser l'appel de votre fonction AWS Lambda

1. Ouvrez la [console IAM](#).
2. Dans le volet de navigation de gauche, développez la gestion des accès pour sélectionner Politiques.
3. Choisissez Créer une politique.
4. Dans l'onglet JSON, saisissez la politique suivante :
 - Voici un exemple de politique IAM qui Amazon Aurora PostgreSQL autorise l'appel de la AuroraGeocoder AWS Lambda fonction.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "lambda:InvokeFunction",
      "Resource": [
        "arn:aws:lambda:<Region>:<AccountId>:function:AuroraGeocoder"
      ]
    }
  ]
}
```

5. Choisissez Next : Tags pour ajouter des tags facultatifs.
6. Choisissez Suivant : vérification.
7. Passez en revue votre politique et entrez les informations suivantes pour celle-ci :
 - Nom — Utilisez des caractères alphanumériques et des caractères « +=, .@-_ ». 128 caractères maximum. Par exemple, *AuroraGeocoderInvoke*.
 - Description — Entrez une description facultative. Utilisez des caractères alphanumériques et des caractères « +=, .@-_ ». 1 000 caractères maximum.
8. Choisissez Créer une politique. Notez l'ARN de cette stratégie, que vous utilisez pour associer la stratégie à un rôle IAM.

Créez un rôle IAM pour autoriser Amazon Relational Database Service (Amazon RDS)

En créant un rôle IAM, Amazon Aurora PostgreSQL vous pouvez assumer le rôle en votre nom pour accéder à votre fonction Lambda. Pour plus d'informations, veuillez consulter [Création d'un rôle pour déléguer des autorisations à un utilisateur IAM](#) dans le Guide de l'utilisateur IAM.

L'exemple suivant est une AWS CLI commande qui crée un rôle nommé *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"
    }
  ]
}'
```

Associez votre politique IAM au rôle IAM

Lorsque vous avez un rôle IAM, attachez la politique IAM que vous avez créée.

L'exemple suivant est une AWS CLI commande qui associe la politique *AuroraGeocoderInvoke* au rôle *AuroraGeocoderInvokeRole*.

```
aws iam attach-role-policy --policy-arn AuroraGeocoderInvoke --role-
name AuroraGeocoderInvokeRole
```

Ajouter le rôle IAM à un cluster de base de données Amazon Aurora

L'exemple suivant est une AWS CLI commande permettant d'ajouter un rôle IAM à un Amazon Aurora PostgreSQL cluster de base de données nommé *MyAuroraCluster*.

```
aws rds add-role-to-db-cluster \
--db-cluster-identifiant MyAuroraCluster \
--feature-name Lambda \
--role-arn AuroraGeocoderInvokeRole \
--region your-region
```

Appeler la fonction AWS Lambda

Après avoir autorisé l'appel de Amazon Aurora PostgreSQL votre fonction Lambda de géocodage, vous pouvez créer Amazon Aurora PostgreSQL une fonction définie par l'utilisateur pour appeler la fonction de géocodage. AWS Lambda Pour plus d'informations, consultez la section [Invocation d'une AWS Lambda fonction depuis un Amazon Aurora PostgreSQL cluster](#) de bases de données dans le guide de l'utilisateur Amazon Aurora.

Installez les extensions PostgreSQL requises

Pour installer les `aws_lambda` extensions `aws_commons` et extensions PostgreSQL requises, [consultez la section Présentation de l'utilisation d'une fonction Lambda dans le guide de l'utilisateur Amazon Aurora](#).

```
CREATE EXTENSION IF NOT EXISTS aws_lambda CASCADE;
```

Installez les extensions PostGIS requises

PostGIS est une extension de PostgreSQL pour le stockage et la gestion des informations spatiales. Pour plus d'informations, consultez la section [Utilisation de l'extension PostGIS](#) dans le guide de l'utilisateur d'Amazon Relational Database Service.

```
CREATE EXTENSION IF NOT EXISTS postgis;
```

Créez une fonction SQL définie par l'utilisateur qui invoque la fonction Lambda

Dans un éditeur SQL, créez une nouvelle fonction définie par l'utilisateur `f_SearchPlaceIndexForText` pour appeler la fonction *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;
$function$;
```

Appelez la fonction SQL pour géocoder depuis Aurora

L'exécution de l'instruction SQL appelle la *AuroraGeocoder* fonction Lambda, qui prend les enregistrements d'adresses de la table de base de données de la base de données et les géocode à Amazon Aurora PostgreSQL l'aide d'une ressource d'index de localisation.

Note

Amazon Aurora PostgreSQL invoque la fonction Lambda pour chaque appel à la fonction SQL définie par l'utilisateur.

Si vous géocodez 50 lignes, Amazon Aurora PostgreSQL invoque la fonction Lambda 50 fois. Une invocation pour chaque ligne.

La fonction `f_SearchPlaceIndexForText` SQL suivante envoie des requêtes à l'[SearchPlaceIndexForText](#) API d'Amazon Location via la fonction *AuroraGeocoder* Lambda. La fonction renvoie une `geom` colonne qui est une géométrie PostGIS, qui est `ST_AsText(geom)` convertie en texte.

```
SELECT *, ST_AsText(geom)
FROM f_SearchPlaceIndexForText('Vancouver, BC');
```

Par défaut, le retour contiendra une ligne. Pour demander des lignes supplémentaires, dans la `MaxResults` limite autorisée, exécutez l'instruction SQL suivante tout en fournissant un `BiasPosition` et en limitant les résultats au Canada.

```
SELECT *
FROM f_SearchPlaceIndexForText('Mount Pleasant', ST_MakePoint(-123.113, 49.260), null,
'{"CAN"}', 5);
```

Pour filtrer les résultats à l'aide d'un cadre de sélection, transmettez un [Box2D](#) comme suit : `filter_bbox`

- [FilterBBox](#)— Filtre les résultats en renvoyant les places à l'intérieur d'un cadre de sélection. Ce paramètre est facultatif.

```
SELECT *
FROM f_SearchPlaceIndexForText('Mount Pleasant', null, 'BOX(-139.06 48.30, -114.03
60.00)>::box2d, '{"CAN"}', 5);
```

[Pour plus d'informations sur les types et les fonctions de PostGIS, consultez le manuel de référence PostGIS.](#)

Enrichir une base de données contenant des données d'adresses

Vous pouvez créer une adresse formatée, la normaliser et le géocoder simultanément à l'aide de l'opération Amazon Location à partir d'une table de base de `SearchPlaceIndexForText` données contenant les données suivantes réparties dans les colonnes suivantes :

- `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;

```

L'exemple suivant illustre une ligne de table de données résultante :

```

id |          formatted_address          |          normalized_address          |
latitude |          longitude          |
-----+-----+-----
+-----+-----+-----
42 | 123 Anytown Ave N, Seattle, WA | 123 Anytown Ave N, Seattle, WA, 12345, USA |
47.6223000127926 | -122.336745971039
(1 row)

```

Mettre à jour la table de base de données et remplir les colonnes

L'exemple suivant met à jour le tableau et remplit les colonnes avec les résultats des SearchPlaceIndexForText requêtes :

```

WITH source_data AS (
  -- select rows that have not been geocoded and created a formatted address for each
  SELECT
    id,
    address || ', ' || city || ', ' || state || ', ' || zip AS formatted_address
  FROM addresses
  WHERE label IS NULL
  -- limit the number of rows that will be geocoded; remove this to geocode the entire
table
  LIMIT 1
),
geocoded_data AS (
  -- geocode each row and keep it linked to the source's ID
  SELECT
    id,
    (f_SearchPlaceIndexForText(formatted_address)).*
  FROM source_data
)
UPDATE addresses
-- populate columns
SET

```

```
normalized_address = geocoded_data.label,  
latitude = ST_Y(geocoded_data.geom),  
longitude = ST_X(geocoded_data.geom)  
FROM geocoded_data  
-- ensure that rows match  
WHERE addresses.id = geocoded_data.id;
```

Étapes suivantes

Un exemple de code est disponible dans le référentiel d'exemples d'Amazon Location Service sur [GitHub](#), qui inclut [un AWS CloudFormation modèle](#).

Gestion des ressources de votre index des places

Vous pouvez gérer les ressources de votre index de localisation à l'aide de la console Amazon LocationAWS CLI, ou des API Amazon Location.

Répertoriez les ressources de votre index des lieux

Vous pouvez consulter la liste des ressources de votre index de localisation à l'aide de la console Amazon LocationAWS CLI, ou des API Amazon Location :

Console

Pour consulter la liste des ressources d'index des lieux à l'aide de la console Amazon Location

1. Ouvrez la console Amazon Location à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Choisissez Placer les index dans le volet de navigation de gauche.
3. Consultez la liste des ressources de votre index des lieux dans la section Mes index des lieux.

API

Utilisez l'[ListPlaceIndexes](#)opération depuis les API Amazon Location Places.

L'exemple suivant est une demande d'API visant à obtenir une liste des ressources d'index de lieux dans le AWS compte.

```
POST /places/v0/list-indexes
```

Voici un exemple de réponse pour [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

Utilisez la commande [list-place-indexes](#).

L'exemple suivant permet d'obtenir une liste des ressources de l'index des lieux dans le compte AWS.

```
aws location list-place-indexes
```

Obtenir les détails des ressources de l'index des lieux

Vous pouvez obtenir des informations sur n'importe quelle ressource d'index de localisation de votre compte AWS à l'aide de la console Amazon Location, de l'AWS CLI, des API Amazon Location ou des API Amazon Location :

Console

Pour consulter les détails d'une ressource d'index de lieux à l'aide de la console Amazon Location

1. Ouvrez la console Amazon Location à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Choisissez Placer les index dans le volet de navigation de gauche.
3. Sous Mes index de lieux, sélectionnez le lien du nom de la ressource d'index de lieux cible.

API

Utilisez l'[DescribePlaceIndex](#) opération depuis les API Amazon Location Place.

L'exemple suivant est une demande d'API pour obtenir les détails des ressources d'index de lieux pour *ExamplePlaceIndex*.

```
GET /places/v0/indexes/ExamplePlaceIndex
```

Voici un exemple de réponse pour [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

Utilisez la commande [describe-place-index](#).

L'exemple suivant permet d'AWS CLI d'obtenir les détails des ressources d'index de lieux pour *ExamplePlaceIndex*.

```
aws location describe-place-index \
  --index-name "ExamplePlaceIndex"
```

Supprimer une ressource d'index de lieux

Vous pouvez supprimer une ressource d'index de lieux de votre compte AWS à l'aide de la console Amazon Location, de l'AWS CLI, des API Amazon Location ou des API Amazon Location :

Console

Pour supprimer une ressource d'index de lieux à l'aide de la console Amazon Location

⚠ Warning

Cette opération supprime définitivement la ressource.

1. Ouvrez la console Amazon Location à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Choisissez Placer les index dans le volet de navigation de gauche.
3. Sous Index des lieux, sélectionnez la ressource d'index des lieux cible.
4. Choisissez Supprimer l'index des lieux.

API

Utilisez l'[DeletePlaceIndex](#) opération depuis les API Amazon Location Places.

L'exemple suivant est une demande d'API pour supprimer la ressource d'index de lieux *ExamplePlaceIndex*.

```
DELETE /places/v0/indexes/ExamplePlaceIndex
```

Voici un exemple de réponse positive pour [DeletePlaceIndex](#) :

```
HTTP/1.1 200
```

CLI

Utilisez la commande [delete-place-index](#).

L'exemple suivant est une AWS CLI commande permettant de supprimer la ressource d'index de lieux *ExamplePlaceIndex*.

```
aws location delete-place-index \  
  --index-name "ExamplePlaceIndex"
```

Calcul des itinéraires à l'aide d'Amazon Location Service

Amazon Location vous permet de sélectionner un fournisseur de données pour calculer un itinéraire en créant et en configurant une ressource de calcul d'itinéraire.

Vous pouvez utiliser la ressource de calcul d'itinéraire pour [calculer un itinéraire](#) en fonction de paramètres spécifiques à l'aide du AWS SDK ou des points de terminaison de l'API REST. Utilisez cette ressource de calcul d'itinéraire pour calculer les itinéraires entre une origine, une destination et jusqu'à 23 points de cheminement pour différents modes de transport, évitements et conditions de circulation.

Vous pouvez également utiliser la ressource de calcul d'itinéraires pour créer des entrées pour vos algorithmes ou produits de planification d'itinéraires en [calculant une matrice d'itinéraires](#). Calculez le temps de trajet et la distance parcourue entre un ensemble de positions de départ et un ensemble de positions de destination. Les logiciels de planification d'itinéraires peuvent utiliser ces données de temps et de distance pour optimiser un itinéraire ou un ensemble d'itinéraires, par exemple, si vous planifiez plusieurs itinéraires de livraison et que vous souhaitez trouver le meilleur itinéraire et la meilleure heure pour chaque arrêt. Vous pouvez calculer une matrice d'itinéraires pour différents modes de transport, évitements et conditions de circulation.

Note

Pour un aperçu des concepts de routage, voir [Acheminements](#).

Rubriques

- [Prérequis](#)
- [Calculer un itinéraire](#)
- [Planification d'itinéraires à l'aide d'une matrice d'itinéraires](#)
- [Positions non situées sur une route](#)
- [Heure de départ](#)
- [Mode de voyage](#)
- [Gestion des ressources de votre calculateur d'itinéraires](#)

Prérequis

Avant de commencer à calculer les itinéraires, suivez les étapes préalables :

Rubriques

- [Création d'une ressource de calcul d'itinéraire](#)
- [Authentification de vos demandes](#)

Création d'une ressource de calcul d'itinéraire

Avant de calculer un itinéraire, créez une ressource de calcul d'itinéraire dans votre AWS compte.

Lorsque vous créez une ressource de calcul d'itinéraire, vous pouvez choisir parmi les fournisseurs de données disponibles :

1. Esri — Pour plus d'informations sur la couverture d'Esri dans la région qui vous intéresse, consultez les informations d'[Esri sur les réseaux routiers et la couverture](#) du trafic.
2. Technologies HERE — Pour plus d'informations sur la couverture de HERE dans la région qui vous intéresse, consultez les rubriques Couverture des [itinéraires automobiles HERE et Couverture des itinéraires des camions HERE](#).
3. Grab — Pour plus d'informations sur la couverture de Grab, voir [Pays/régions et zone couverts](#).

Note

Si votre application assure le suivi ou le routage des ressources que vous utilisez dans votre entreprise, comme les véhicules de livraison ou les employés, vous ne devez pas utiliser Esri comme fournisseur de géolocalisation. Consultez la section 82 des [Conditions de service AWS](#) pour plus de détails.

Vous pouvez le faire à l'aide de la console Amazon Location ServiceAWS CLI, ou des API Amazon Location.

Console

Pour créer une ressource de calcul d'itinéraire à l'aide de la console Amazon Location

1. Ouvrez la console Amazon Location à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Dans le volet de navigation de gauche, choisissez Calculateurs d'itinéraires.
3. Choisissez Créer un calculateur d'itinéraire.
4. Remplissez les cases suivantes :
 - Nom — Entrez le nom de la ressource du calculateur d'itinéraire. Par exemple, *ExampleCalculator*. 100 caractères maximum. Les entrées valides incluent les caractères alphanumériques, les traits d'union, les points et les traits de soulignement.

- Description — Entrez une description facultative.
5. Pour les fournisseurs de données, choisissez un [fournisseur de données](#) à utiliser comme calculateur d'itinéraire.
 6. (Facultatif) Sous Tags (Balises), entrez des paramètres Key (Clé) et Value (Valeur) de balise. Cela ajoute une balise à votre nouvelle ressource de calcul d'itinéraire. Pour plus d'informations, veuillez consulter la rubrique [Balisage de vos ressources](#).
 7. Choisissez Créer un calculateur d'itinéraire.

API

Pour créer une ressource de calcul d'itinéraire à l'aide des API Amazon Location

Utilisez l'[CreateRouteCalculator](#) opération depuis les API Amazon Location Places.

L'exemple suivant est une demande d'API pour créer une ressource de calcul d'itinéraire appelée à *ExampleCalculator* l'aide du fournisseur de données *Esri*.

```
POST /routes/v0/calculators
Content-type: application/json

{
  "CalculatorName": "ExampleCalculator",
  "DataSource": "Esri",
  "Description": "string",
  "Tags": {
    "Tag1" : "Value1"
  }
}
```

AWS CLI

Pour créer une ressource de calcul d'itinéraire à l'aide de AWS CLI commandes

Utilisez la commande `create-route-calculator`.

L'exemple suivant crée une ressource de calcul d'itinéraire appelée *ExampleCalculator* en utilisant *Esri* comme fournisseur de données.

```
aws location \
  create-route-calculator \
```

```
--calculator-name "ExampleCalculator" \  
--data-source "Esri" \  
--tags Tag1=Value1
```

Note

La facturation dépend de votre utilisation. L'utilisation d'autres AWS services peut entraîner des frais. Pour plus d'informations, consultez les [tarifs d'Amazon Location Service](#).

Authentification de vos demandes

Une fois que vous avez créé une ressource de calcul d'itinéraire et que vous êtes prêt à intégrer des fonctionnalités de localisation dans votre application, choisissez la manière dont vous souhaitez authentifier vos demandes :

- Pour découvrir comment accéder aux services, consultez la section [Accès à Amazon Location Service](#).
- Si vous avez un site Web avec des utilisateurs anonymes, vous pouvez utiliser des clés d'API ou Amazon Cognito.

Exemple

L'exemple suivant montre l'utilisation d'une clé d'API pour l'autorisation, à l'aide du [JavaScriptSDK AWS v3](#) et de l'Amazon Location. [JavaScript Aide à l'authentification](#)

```
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);
```

Calculer un itinéraire

Vous pouvez utiliser Amazon Location Service pour calculer les itinéraires entre une origine et une destination, avec jusqu'à 23 points de cheminement le long de l'itinéraire, pour différents modes de transport, évitements et conditions de circulation.

Note

Vous devez d'abord créer une ressource de calcul d'itinéraire et configurer l'authentification pour vos demandes adressées à Amazon Location. Pour plus d'informations, consultez [Prérequis](#).

Commencez à calculer les itinéraires

Soumettez une demande simple en utilisant l'[CalculateRoute](#) opération. Une demande simple contient les champs obligatoires suivants :

- **DeparturePosition**— La position de départ à partir de laquelle calculer l'itinéraire. Défini comme [longitude, latitude]
- **DestinationPosition**— Position finale vers laquelle calculer l'itinéraire. Défini comme [longitude, latitude].

Note

Si vous spécifiez une position de départ ou de destination qui n'est pas située sur une route, Amazon Location [déplace la position vers la route la plus proche](#).

Vous pouvez éventuellement spécifier des [points de cheminement](#), une [heure de départ](#) et un [mode de voyage](#) dans votre demande.

Vous pouvez utiliser les AWS CLI ou les API Amazon Location.

API

L'exemple suivant est une CalculateRoute demande utilisant la ressource du calculateur d'itinéraire *ExampleCalculator*. La demande indique le calcul d'un itinéraire entre une position de départ `[-122.7565, 49.0021]` et une position de destination `[-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

L'exemple suivant est une calculate-route commande utilisant la ressource du calculateur d'itinéraire *ExampleCalculator*. La demande indique le calcul d'un itinéraire entre une position de départ `[-122.7565, 49.0021]` et une position de destination `[-122.3394, 47.6159]`.

```
aws location \
  calculate-route \
    --calculator-name ExampleCalculator \
    --departure-position -122.7565 49.0021 \
    --destination-position -122.3394 47.6159
```

Par défaut, la réponse est renvoyée Distance en kilomètres. Vous pouvez modifier l'unité de mesure en miles à l'aide du paramètre optionnel suivant :

- `DistanceUnit`— Spécifie le système d'unités à utiliser pour les résultats de distance.

Exemple

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route
```

```
Content-type: application/json
{
  "DeparturePosition": [-122.7565,49.0021],
  "DestinationPosition": [-122.3394, 47.6159],
  "DistanceUnit": "Miles"
}
```

Configuration de points de cheminement

Lorsque vous calculez un itinéraire, vous pouvez spécifier jusqu'à 23 points d'escale intermédiaires entre la position de départ et la position de destination à l'aide des positions des points de cheminement.

- **WaypointPositions**— Spécifie une liste ordonnée de positions intermédiaires à inclure le long d'un itinéraire entre la position de départ et la position de destination.

Note

Si vous spécifiez une position de point de cheminement qui n'est pas située sur une route, Amazon Location déplace la position vers la route la plus proche.

Exemple

La [CalculateRoute](#) requête suivante calcule un itinéraire avec 2 points de passage :

- La position de départ est [-122.7565, 49.0021] et la position de destination est [-122.3394, 47.6159].
- Pour le paramètre de requête **WaypointPositions** :
 - La position du premier arrêt est [-122.1884, 48.0936].
 - La deuxième position d'arrêt est [-122.3493, 47.6205].
- *Pour inclure la géométrie de la chaîne linéaire des jambes entre ces deux points de cheminement, définissez le paramètre facultatif suivant sur **true** :*
 - **IncludeLegGeometry**— Inclut la géométrie de chaque trajectoire entre deux positions dans la réponse.

```
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
}
```

Exemple de réponse

Voici un exemple de demande avec la réponse correspondante lorsque vous appelez l'[CalculateRoute](#) opération depuis l'API Amazon Location Routes avec la *valeur* `IncludeLegGeometry true`, qui inclut la géométrie linestring de chaque chemin entre une paire de positions dans la réponse.

Exemple request

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{
  "DeparturePosition": [-122.7565,49.0021],
  "DestinationPosition": [-122.3394, 47.6159],
  "IncludeLegGeometry": true
}
```

Exemple 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
  ]
}
}
```

Planification d'itinéraires à l'aide d'une matrice d'itinéraires

Vous pouvez utiliser Amazon Location Service pour créer des entrées pour votre logiciel de planification et d'optimisation d'itinéraires. Vous pouvez créer des résultats d'itinéraire, y compris le temps de trajet et la distance parcourue, pour des itinéraires entre un ensemble de positions de départ et un ensemble de positions de destination.

Par exemple, compte tenu des positions de départ A et B et des positions de destination X et Y, Amazon Location Service indiquera le temps de trajet et la distance parcourue pour les itinéraires allant de A à X, A à Y, B à X et B à Y.

Vous pouvez calculer les itinéraires en fonction des différents modes de transport, des évitements et des conditions de circulation. Par exemple, vous pouvez spécifier que le véhicule est un camion de 35 pieds de long, et l'itinéraire calculé utilisera ces restrictions pour déterminer le temps et la distance parcourus.

Le nombre de résultats renvoyés (et d'itinéraires calculés) est le nombre de positions de départ multiplié par le nombre de positions de destination. Vous êtes facturé pour chaque itinéraire calculé, et non pour chaque demande adressée au service. Ainsi, une matrice d'itinéraires comportant 10 départs et 10 destinations sera facturée comme 100 itinéraires.

Calcul d'une matrice d'itinéraires

Vous pouvez calculer une matrice d'itinéraires entre un ensemble de positions de départ et un ensemble de positions de destination. Les résultats de l'itinéraire incluront le temps de trajet et la distance parcourue.

Prérequis

- Vous devez d'abord créer une ressource de calcul d'itinéraire et configurer l'authentification pour vos demandes adressées à Amazon Location. Pour plus d'informations, consultez [Prérequis](#).

Soumettez une demande en utilisant l'[CalculateRouteMatrix](#) opération. Une demande minimale contient les champs obligatoires suivants :

- `DeparturePositions`— Ensemble de positions de départ pour lesquelles calculer les itinéraires. Défini comme un tableau de `[longitude, latitude]`
- `DestinationPositions`— Ensemble de positions finales pour lesquelles calculer les itinéraires. Défini comme un tableau de `[longitude, latitude]`.

Note

Si vous spécifiez une position de départ ou de destination qui n'est pas située sur une route, Amazon Location [déplace la position vers la route la plus proche](#).

Vous pouvez éventuellement spécifier une [heure de départ](#) et un [mode de voyage](#) dans votre demande.

Vous pouvez utiliser les AWS CLI ou les API Amazon Location.

API

L'exemple suivant est une `CalculateRouteMatrix` demande utilisant la ressource du calculateur d'itinéraire *ExampleCalculator*. La demande précise le calcul de la

matrice des itinéraires depuis les positions de départ [-122.7565, 49.0021] et [-122.2014, 47.6101] vers les positions de destination [-122.3394, 47.6159] et [-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

L'exemple suivant est une `calculate-route-matrix` commande utilisant la ressource du calculateur d'itinéraire *ExampleCalculator*. La demande précise le calcul de la matrice des itinéraires depuis les positions de départ [-122.7565, 49.0021] et [-122.2014, 47.6101] vers les positions de destination [-122.3394, 47.6159] et [-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]]"
```

Par défaut, la réponse est renvoyée Distance en kilomètres. Vous pouvez modifier l'unité de mesure en miles à l'aide du paramètre optionnel suivant :

- `DistanceUnit`— Spécifie le système d'unités à utiliser pour les résultats de distance.

Exemple

```
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"
}
```

Restrictions relatives aux positions de départ et de destination

Lors du calcul d'une matrice d'itinéraires, les positions de départ et de destination sont soumises à des restrictions. Ces restrictions varient en fonction du fournisseur utilisé par la RouteCalculator ressource.

Limitation	Esri	Saisir	ICI
Nombre de positions	Jusqu'à 10 positions de départ et 10 positions de destination.	Jusqu'à 350 positions de départ et 350 positions de destination.	Jusqu'à 350 positions de départ et 350 positions de destination. Pour les trajets plus longs, des restrictions supplémentaires s'appliquent. Consultez la section .
Distance entre les positions	Toutes les deux positions de départ et de destination doivent être situées à moins de 400 km l'une de l'autre (40 km pour les itinéraires de randonnée).		Toutes les positions de départ et de destination doivent se situer dans un cercle de 180 km de diamètre. Pour les trajets plus longs, des restricti

Limitation	Esri	Saisir	ICI
			ons supplémentaires s'appliquent. Consultez la section .
Longueur de l'itinéraire	Les itinéraires ne seront pas terminés si la durée totale du trajet est supérieure à 400 minutes.		Les itinéraires qui s'écartent de plus de 10 km en dehors d'un cercle autour des points de départ et de destination ne seront pas calculés. Pour les trajets plus longs, des restrictions supplémentaires s'appliquent. Consultez la section .
Régions	Le calcul d'une matrice d'itinéraires n'est pas pris en charge en Corée.	Disponible en Asie du Sud-Est. Pour obtenir la liste des pays/régions pris en charge et pour plus d'informations, consultez. Pays/régions et zone couverts	Aucune restriction supplémentaire.

Planification d'itinéraires à plus long terme

Le calcul d'une matrice des résultats d'itinéraires est utile pour une planification efficace des itinéraires, mais le calcul peut prendre un certain temps. Tous les fournisseurs de données Amazon Location Service limitent le nombre d'itinéraires ou la distance des itinéraires pouvant être calculés. Par exemple, HERE permet de créer des itinéraires entre 350 positions de départ et de destination, mais ces positions doivent se situer dans un cercle de 180 km. Et si vous souhaitez planifier des trajets plus longs ?

Vous pouvez calculer une matrice d'itinéraires avec des longueurs illimitées pour un plus petit nombre d'itinéraires, en utilisant `RouteCalculator` HERE comme fournisseur de données. Cela ne change pas la façon dont vous appelez l'[CalculateRouteMatrix](#) API, Amazon Location autorise simplement les itinéraires plus longs lorsque vous répondez aux exigences.

Les exigences pour le calcul d'itinéraires de plus grande longueur sont les suivantes :

- Ils `RouteCalculator` doivent utiliser le fournisseur de données HERE.
- Le nombre de positions de départ ne doit pas être supérieur à 15.
- Le nombre total d'itinéraires à calculer ne doit pas être supérieur à 100.
- L'itinéraire longue distance n'est pas autorisé pour les itinéraires de camions avec évitement de péage lorsque les itinéraires sont supérieurs à 1 000 km. Cette combinaison est plus lente à calculer et peut entraîner l'expiration de l'appel. Vous pouvez calculer ces itinéraires individuellement avec l'[CalculateRoute](#) opération.

Si votre appel ne répond pas à ces exigences (par exemple, si vous demandez 150 calculs d'itinéraire en un seul appel), vous `CalculateRouteMatrix` reviendrez à autoriser uniquement les règles relatives aux itinéraires les plus courts. Vous pouvez ensuite calculer les itinéraires, à condition que les positions se situent dans un cercle de 180 km.

Lorsque vous calculez des itinéraires plus longs, tenez compte des points suivants :

- Le calcul des itinéraires plus longs peut prendre plus de temps, même plus que le délai maximum pour les API Amazon Location. Si les délais d'attente sont fréquents pour des itinéraires spécifiques, vous pouvez essayer un plus petit nombre d'itinéraires pour chaque appel vers `CalculateRouteMatrix`.
- Si vous ajoutez d'autres positions de destination ou de départ à votre `CalculateRouteMatrix` demande, l'opération peut passer en mode plus restreint, et vous pouvez obtenir une erreur pour un itinéraire qui peut être calculé sans problème lorsqu'il y a moins d'itinéraires à créer. Dans ce cas, réduisez le nombre de positions de destination ou de départ et faites plusieurs demandes pour obtenir l'ensemble complet des calculs d'itinéraire dont vous avez besoin.

Exemple de réponse

Voici un exemple de demande avec la réponse correspondante lorsque vous appelez l'[CalculateRouteMatrix](#) opération depuis l'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"  
  }  
}
```

Positions non situées sur une route

Lorsque vous utilisez `CalculateRoute` ou `CalculateRouteMatrix` si vous spécifiez une position de départ, de destination ou de point de cheminement qui n'est pas située sur une route, Amazon Location déplace la position vers une route voisine.

La [CalculateRoute](#) demande suivante spécifie une position de départ et une position de destination qui ne sont pas situées sur une route :

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route  
Content-type: application/json  
{  
  "DeparturePosition": [-123.128014, 49.298472],  
  "DestinationPosition": [-123.134701, 49.294315]  
}
```

La réponse qui en résulte renvoie une position qui est capturée sur une route voisine :

```
{  
  "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": 8  
        }  
      ]  
    }  
  ]  
}
```

```
    "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"
}
}
```

Heure de départ

Par défaut, lorsque vous appelez `CalculateRoute` ou `CalculateRouteMatrix` si vous n'indiquez pas d'heure de départ dans la demande, les itinéraires calculés reflètent les conditions de circulation optimales.

Vous pouvez définir une heure de départ spécifique pour utiliser les conditions de trafic prévisionnelles et en temps réel fournies par le fournisseur de données de votre choix, en utilisant l'une des options suivantes :

- `DepartNow`— Lorsqu'il est défini sur `true`, il utilise les conditions de circulation en temps réel pour calculer le trajet le plus rapide.
- `DepartureTime`— Lorsqu'il est fourni, il utilise des conditions de circulation prévisibles et connues pour la durée demandée. Défini dans le [format](#) suivant : `:YYYY-MM-DDThh:mm:ss.sssZ`.

Exemple

La [CalculateRoute](#) demande suivante fixe l'heure de départ au 2 juillet 2024, à 12:15:20 UTC.

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{
  "DeparturePosition": [-122.7565,49.0021],
  "DestinationPosition": [-122.3394, 47.6159],
  "WaypointPositions":[
    [-122.1884,48.0936],
    [-122.3493,47.6205]
  ]
}
```

```
"IncludeLegGeometry": true,  
"DepartureTime": 2024-07-02T12:15:20.000Z,  
}
```

Mode de voyage

Vous pouvez définir un mode de déplacement lorsque vous utilisez `CalculateRoute` ou `CalculateRouteMatrix`. Le mode de déplacement influe sur la vitesse de déplacement et la compatibilité routière. Bien que le mode de déplacement par défaut soit la voiture, vous pouvez spécifier le mode de déplacement que vous utilisez lorsque vous parcourez un itinéraire à l'aide du paramètre facultatif suivant :

- `TravelMode`— Spécifie le mode de transport lors du calcul d'un itinéraire, tel que : *BicycleCar*, *Motorcycle*, *Truck*, ou *Walking*.

Limites

- Si vous spécifiez le mode `Walking` de voyage et que votre fournisseur de données est Esri, le point de départ et la destination doivent être situés dans un rayon de 40 km.
- `Bicycle` ou `Motorcycle` sont disponibles que lorsque vous utilisez Grab comme fournisseur de données.
- Grab propose uniquement `Bicycle` des `Walking` itinéraires dans certaines villes. Pour plus d'informations, consultez [Pays/régions et zone couverts](#).
- `Truck` n'est pas disponible lorsque vous utilisez Grab comme fournisseur de données.

Préférences supplémentaires

Si vous spécifiez un `TravelMode` de *Car*, vous pouvez définir des préférences de route supplémentaires à l'aide du paramètre facultatif suivant :

- `CarModeOptions`— Spécifie les préférences d'itinéraire lorsque vous voyagez en voiture, telles que *AvoidFerries* ou *AvoidTolls*.

Si vous spécifiez un `TravelMode` de *Truck*, vous pouvez définir des préférences de route supplémentaires à l'aide du paramètre facultatif suivant :

- **TruckModeOptions**— Spécifie les préférences d'itinéraire lorsque vous voyagez en camion, telles que *AvoidFerries* ou *AvoidTolls*, en plus de spécifier les itinéraires pouvant accueillir le *TruckDimensions* et *TruckWeight*.

Exemple

La [CalculateRoute](#) demande *Truck* suivante indique le mode de déplacement. Les restrictions d'itinéraire supplémentaires incluent : éviter les itinéraires empruntés par des ferries et éviter les routes qui ne peuvent pas accueillir les dimensions et le poids des camions.

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

Gestion des ressources de votre calculateur d'itinéraires

Vous pouvez gérer les ressources de votre calculateur d'itinéraires à l'aide de la console Amazon LocationAWS CLI, ou des API Amazon Location.

Listez les ressources de votre calculateur d'itinéraires

Vous pouvez consulter la liste de votre calculateur d'itinéraires à l'aide de la console Amazon LocationAWS CLI, ou des API Amazon Location :

Console

Pour consulter la liste des calculateurs d'itinéraires à l'aide de la console Amazon Location

1. Ouvrez la console Amazon Location à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Choisissez Calculateurs d'itinéraires dans le volet de navigation de gauche.
3. Consultez les détails du calculateur d'itinéraire sous Mes calculateurs d'itinéraires.

API

Utilisez l'[ListRouteCalculators](#) opération depuis les API Amazon Location Routes.

L'exemple suivant est une demande d'API visant à obtenir une liste de calculateurs d'itinéraires dans le AWS compte.

```
POST /routes/v0/list-calculators
```

Voici un exemple de réponse pour [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

Utilisez la commande `list-route-calculators`.

L'exemple suivant montre comment AWS CLI obtenir une liste de calculateurs d'itinéraires dans le AWS compte.

```
aws location list-route-calculators
```

Obtenez les détails du calculateur d'itinéraire

Vous pouvez obtenir des informations sur n'importe quelle ressource de calcul d'itinéraire de votre AWS compte à l'aide de la console Amazon Location, AWS CLI, ou des API Amazon Location :

Console

Pour consulter les détails d'un calculateur d'itinéraire à l'aide de la console Amazon Location

1. Ouvrez la console Amazon Location à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Choisissez Calculateurs d'itinéraires dans le volet de navigation de gauche.
3. Sous Mes calculateurs d'itinéraire, sélectionnez le lien du nom du calculateur d'itinéraire cible.

API

Utilisez l'[DescribeRouteCalculator](#) opération depuis les API Amazon Location Routes.

L'exemple suivant est une demande d'API pour obtenir les détails du calculateur d'itinéraire pour *ExampleCalculator*.

```
GET /routes/v0/calculators/ExampleCalculator
```

Voici un exemple de réponse pour [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

Utilisez la commande `describe-route-calculator`.

L'exemple suivant permet d'obtenir les détails du calculateur d'itinéraire pour *ExampleCalculator*.

```
aws location describe-route-calculator \  
  --calculator-name "ExampleCalculator"
```

Supprimer un calculateur d'itinéraire

Vous pouvez supprimer un calculateur d'itinéraire de votre AWS compte à l'aide de la console Amazon Location, de l'AWS CLI, ou des API Amazon Location :

Console

Pour supprimer un calculateur d'itinéraire à l'aide de la console Amazon Location

Warning

Cette opération supprime définitivement la ressource.

1. Ouvrez la console Amazon Location à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Choisissez Calculateurs d'itinéraires dans le volet de navigation de gauche.
3. Sous Mes calculateurs d'itinéraire, sélectionnez le calculateur d'itinéraire cible.
4. Choisissez Supprimer le calculateur d'itinéraire.

API

Utilisez l'[DeleteRouteCalculator](#) opération depuis les API Amazon Location Routes.

L'exemple suivant est une demande d'API pour supprimer la collection Geofence.

ExampleCalculator

```
DELETE /routes/v0/calculators/ExampleCalculator
```

Voici un exemple de réponse pour [DeleteRouteCalculator](#) :

```
HTTP/1.1 200
```

CLI

Utilisez la commande `delete-route-calculator`.

L'exemple suivant est une AWS CLI commande permettant de supprimer la collection Geofence.

ExampleCalculator

```
aws location delete-route-calculator \  
    --calculator-name "ExampleCalculator"
```

Géolocalisation d'une zone d'intérêt à l'aide d'Amazon Location

Une application de géofencing évalue la position d'un appareil suivi par rapport aux zones d'intérêt précédemment enregistrées. Cela permet de prendre des mesures en fonction des mises à jour de position. Par exemple, vous pouvez lancer un événement qui invite à recevoir une notification lorsqu'un client ayant commandé du café sur son application mobile se trouve à proximité d'un magasin.

Note

Pour un aperçu des concepts de géofencing et de traceur, voir [Geofences et traceurs](#)

Cette section du guide fournit des step-by-step instructions pour créer une application de géofencing à l'aide d'Amazon Location Service.

Présentation des étapes

1. Ajoutez des géofences autour des zones d'intérêt et stockez-les dans une ressource de collecte de géofences.
2. Commencez à suivre vos appareils cibles et stockez l'historique de localisation de l'appareil dans une ressource de suivi.
3. Associez votre ressource de suivi à votre ressource de collecte de géofences afin que les mises à jour de localisation de l'appareil soient automatiquement évaluées par rapport à toutes vos géofences.
4. Vous pouvez évaluer la position des appareils directement par rapport à vos ressources de collecte de géofences si vous ne souhaitez pas utiliser Amazon Location Trackers pour conserver l'historique de localisation de vos appareils.

Une fois que vous avez implémenté votre solution de géofencing, votre ressource de collecte de géofences émet les événements suivants :

- ENTER— Un appareil suivi entre dans une géofence au sein d'une collection de géofences.
- EXIT— Un appareil suivi sort d'une géofence au sein d'une collection de géofences.

Vous pouvez utiliser Amazon EventBridge pour réagir aux événements en les acheminant ailleurs.

Au lieu d'envoyer des mises à jour via les API Amazon Location Service depuis chaque appareil, vous pouvez utiliser MQTT pour envoyer des mises à jour de l'appareil.

Les rubriques suivantes décrivent en détail ces étapes et les alternatives.

Rubriques

- [Ajouter des géofences](#)
- [Commencer le suivi](#)
- [Associer un traceur à une collection de géofences](#)
- [Évaluez la position des appareils par rapport aux géofences](#)
- [Vérifier la position des appareils](#)
- [Réagir aux événements d'Amazon Location Service avec Amazon EventBridge](#)
- [Suivi à l'aide AWS IoT de MQTT avec Amazon Location Service](#)
- [Gestion de vos ressources de collecte de géofences](#)
- [Gestion de vos ressources de suivi](#)
- [Exemple d'application mobile de géofencing et de suivi](#)

Ajouter des géofences

Les géofences contiennent des points et des sommets qui forment une limite fermée, qui définit une zone d'intérêt. Les collections Geofence stockent et gèrent une ou plusieurs géofences.

[Les collections de géofences Amazon Location stockent les géofences définies à l'aide d'un format de données géospatiales standard appelé GeoJSON \(RFC 7946\)](#). Vous pouvez utiliser gratuitement des outils tels que geojson.io pour dessiner graphiquement vos géofences et enregistrer le fichier GeoJSON de sortie.

Note

Amazon Location ne prend pas en charge les polygones comportant des trous, les polygones multipolygones, les polygones dans le sens des aiguilles d'une montre et les géofences qui traversent l'antiméridien.

Création d'une collection de géofences

Créez une collection de géofences pour stocker et gérer les géofences à l'aide de la console Amazon Location, AWS CLI ou des API Amazon Location.

Console

Pour créer une collection de géofences à l'aide de la console Amazon Location

1. Ouvrez la console Amazon Location Service à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Dans le volet de navigation de gauche, choisissez Geofence collections.
3. Choisissez Créer une collection de géofences.
4. Remplissez les cases suivantes :
 - Nom — Entrez un nom unique. Par exemple, *ExampleGeofenceCollection*. 100 caractères maximum. Les entrées valides incluent les caractères alphanumériques, les traits d'union, les points et les traits de soulignement.
 - Description — Entrez une description facultative pour différencier vos ressources.
5. Dans le cadre d'une EventBridge règle avec CloudWatch comme cible, vous pouvez créer une EventBridge règle facultative pour commencer à [réagir aux événements de géofence](#). Cela permet à Amazon Location de publier des événements sur Amazon CloudWatch Logs.
6. (Facultatif) Sous Tags (Balises), entrez des paramètres Key (Clé) et Value (Valeur) de balise. Cela ajoute un tag à votre nouvelle collection de géofences. Pour plus d'informations, consultez [Marquage de vos ressources Amazon Location Service](#).
7. (Facultatif) Sous Chiffrement de clé géré par le client, vous pouvez choisir d'ajouter une clé gérée par le client. Cela ajoute une clé symétrique gérée par le client que vous créez, détenez et gérez par rapport au chiffrement AWS détenu par défaut. Pour plus d'informations, consultez la section [Chiffrement des données au repos](#).

8. Choisissez Créer une collection de géofences.

API

Pour créer une collection de géofences à l'aide des API Amazon Location

Utilisez l'[CreateGeofenceCollection](#) opération depuis les API Amazon Location Geofences.

L'exemple suivant utilise une demande d'API pour créer une collection de géofences appelée.

ExampleGeofenceCollection La collection de géofences est associée à une [AWS KMS clé gérée par le client pour chiffrer les données](#) du client.

```
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

Pour créer une collection de géofences à l'aide de commandes AWS CLI

Utilisez la commande [create-geofence-collection](#).

L'exemple suivant utilise un AWS CLI pour créer une collection de géofences appelée.

ExampleGeofenceCollection La collection de géofences est associée à une [AWS KMS clé gérée par le client pour chiffrer les données](#) du client.

```
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 facturation dépend de votre utilisation. L'utilisation d'autres AWS services peut entraîner des frais. Pour plus d'informations, consultez la [tarification d'Amazon Location Service](#).

Dessiner des géofences

Maintenant que vous avez créé votre collection de géofences, vous pouvez définir vos géofences. Les géofences sont définies sous la forme d'un polygone ou d'un cercle. [Pour dessiner une géofence polygonale, vous pouvez utiliser un outil d'édition GeoJSON, tel que `geojson.io`](#).

Pour créer une clôture géographique sous forme de cercle, vous devez définir le point central du cercle et le rayon. Par exemple, si vous souhaitez créer une clôture géographique pour être averti chaque fois qu'un appareil se trouve à moins de 50 mètres d'un emplacement spécifique, vous devez utiliser la latitude et la longitude de cet emplacement et spécifier un rayon de 50 mètres.

À l'aide des API Amazon Location Service, vous pouvez également ajouter des métadonnées à votre géofence, sous la forme de paires clé-valeur. Ils peuvent être utiles pour stocker des informations sur la géofence, telles que son type, ou d'autres informations spécifiques à votre application. Vous pouvez utiliser ces métadonnées lorsque [Réagir aux événements d'Amazon Location Service avec Amazon EventBridge](#).

Ajouter des géofences polygonales

Cette section décrit la création de géofences polygonales

Dessinez des géofences à l'aide d'un outil GeoJSON

[Maintenant que vous avez créé votre collection de géofences, vous pouvez définir vos géofences à l'aide d'un outil d'édition GeoJSON, tel que `geojson.io`](#).

Pour créer un fichier GeoJSON

1. Ouvrez un outil d'édition GeoJSON. Par exemple, `geojson.io`.
2. Choisissez l'icône Dessiner un polygone et dessinez la zone qui vous intéresse.
3. Choisissez Enregistrer, puis GeoJSON dans le menu déroulant.

Placer des géofences GeoJSON dans une collection de géofences

Vous pouvez utiliser le fichier GeoJSON obtenu pour charger vos géofences à l'aide de la console Amazon Location Service, de, AWS CLI ou des API Amazon Location :

Console

Pour ajouter une clôture géographique à une collection de géofences à l'aide de la console Amazon Location Service

1. Ouvrez la console Amazon Location Service à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Dans le volet de navigation de gauche, choisissez Geofence collections.
3. Dans la liste des collections Geofence, sélectionnez le lien du nom de la collection Geofence cible.
4. Sous Geofences, choisissez Create Geofences.
5. Dans la fenêtre Ajouter des géofences, faites glisser et déposez votre fichier GeoJSON dans la fenêtre.
6. Choisissez Ajouter des géofences.

API

Pour ajouter des barrières géographiques à l'aide des API Amazon Location

Utilisez l'[PutGeofence](#) opération depuis les API Amazon Location Geofences.

L'exemple suivant utilise une demande d'API pour ajouter une clôture géographique ayant l'ID *GEOFENCE-EXAMPLE1* à une collection de géofences appelée. *ExampleGeofenceCollection* Il spécifie également une propriété de métadonnées de géofence unique avec la clé Type et la valeur. 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]
        ]
      ]
    }
  }
}

```

Vous pouvez également ajouter plusieurs barrières géographiques à l'aide de cette opération.

[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

Pour ajouter une clôture géographique à une collection de barrières géographiques à l'aide de commandes AWS CLI

Utilisez la commande [put-geofence](#).

L'exemple suivant utilise un AWS CLI pour ajouter une clôture géographique à une collection de géofences appelée. *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"
}
```

Ajouter des géofences circulaires

Cette section décrit la création de géofences circulaires. Vous devez connaître la latitude et la longitude du point que vous souhaitez placer au centre du cercle, ainsi que le rayon en mètres du cercle. Vous pouvez créer des barrières géographiques circulaires à l'aide des API Amazon Location ou du AWS CLI

API

Pour ajouter des géofences circulaires à l'aide des API Amazon Location

Utilisez l'[PutGeofence](#) opération depuis les API Amazon Location Geofences.

L'exemple suivant utilise une demande d'API pour ajouter une clôture géographique ayant l'ID *GEOFENCE-EXAMPLE2* à une collection de géofences appelée : *ExampleGeofenceCollection*

```
PUT /geofencing/v0/collections/ExampleGeofenceCollection/geofence/GEOFENCE-EXAMPLE2
Content-type: application/json

{
  "Geometry": {
```

```
    "Circle": {
      "Center": [-5.716667, -15.933333],
      "Radius": 50
    }
  }
}
```

AWS CLI

Pour ajouter une géofence circulaire à une collection de géofences à l'aide de commandes AWS CLI

Utilisez la commande [put-geofence](#).

L'exemple suivant utilise un AWS CLI pour ajouter une clôture géographique à une collection de géofences appelée. *ExampleGeofenceCollection*

```
$ aws location \
  put-geofence \
    --collection-name ExampleGeofenceCollection \
    --geofence-id ExampleGeofenceCircle \
    --geometry 'Circle={Center=[-5.716667, -15.933333], Radius=50}'
```

Note

Vous pouvez également placer le JSON pour les géométries complexes dans son propre fichier, comme dans l'exemple suivant.

```
$ aws location \
  put-geofence \
    --collection-name ExampleGeofenceCollection \
    --geofence-id ExampleGeofenceCircle \
    --geometry file:circle.json
```

Dans l'exemple, le fichier circle.json inclut le JSON pour la géométrie du cercle.

```
{
  "Circle": {
    "Center": [-74.006975, 40.717127],
    "Radius": 287.7897969218057
  }
}
```

```
}
```

Commencer le suivi

Cette section vous explique comment créer une application de suivi qui capture la position des appareils.

Création d'un tracker

Créez une ressource de suivi pour stocker et traiter les mises à jour de position à partir de vos appareils. Vous pouvez utiliser la console Amazon Location Service AWS CLI, ou les API Amazon Location.

Chaque mise à jour de position stockée dans vos ressources de suivi peut inclure une mesure de la précision de la position et jusqu'à trois champs de métadonnées concernant la position ou l'appareil que vous souhaitez stocker. Les métadonnées sont stockées sous forme de paires clé-valeur et peuvent stocker des informations telles que la vitesse, la direction, la pression des pneus ou la température du moteur.

Les trackers filtrent les mises à jour de position au fur et à mesure qu'elles sont reçues. Cela réduit le bruit visuel sur les trajectoires de votre appareil (appelé instabilité) et le nombre de faux événements d'entrée et de sortie de barrières géographiques. Cela permet également de gérer les coûts en réduisant le nombre d'évaluations de géofences lancées.

Les trackers proposent trois options de filtrage des positions pour vous aider à gérer les coûts et à réduire l'agitation lors des mises à jour de votre position.

- Basé sur la précision — À utiliser avec n'importe quel appareil fournissant une mesure précise. La plupart des appareils mobiles fournissent ces informations. La précision de chaque mesure de position dépend de nombreux facteurs environnementaux, notamment la réception des satellites GPS, le paysage et la proximité des appareils Wi-Fi et Bluetooth. La plupart des appareils, y compris la plupart des appareils mobiles, peuvent fournir une estimation de la précision de la mesure en même temps que la mesure. Grâce au AccuracyBased filtrage, Amazon Location ignore les mises à jour de localisation si le déplacement de l'appareil est inférieur à la précision mesurée. Par exemple, si deux mises à jour consécutives provenant d'un appareil ont une plage de précision de 5 m et 10 m, Amazon Location ignore la deuxième mise à jour si l'appareil s'est déplacé de moins de 15 m. Amazon Location n'évalue pas les mises à jour ignorées par rapport aux géofences et ne les stocke pas.

Lorsque la précision n'est pas fournie, elle est considérée comme nulle et la mesure est considérée comme parfaitement précise.

 Note

Vous pouvez également utiliser le filtrage basé sur la précision pour supprimer tous les filtres. Si vous sélectionnez le filtrage basé sur la précision, mais que vous remplacez toutes les données de précision à zéro, ou que vous omettez complètement la précision, Amazon Location ne filtrera aucune mise à jour.

- Basé sur la distance : à utiliser lorsque vos appareils ne fournissent pas de mesure précise, mais que vous souhaitez tout de même tirer parti du filtrage pour réduire l'instabilité et gérer les coûts. DistanceBasedle filtrage ignore les mises à jour de localisation dans lesquelles les appareils se sont déplacés à moins de 30 m (98,4 pieds). Lorsque vous utilisez le filtrage de DistanceBased position, Amazon Location n'évalue pas ces mises à jour ignorées par rapport aux géofences et ne stocke pas les mises à jour.

La précision de la plupart des appareils mobiles, y compris la précision moyenne des appareils iOS et Android, est inférieure à 15 m. Dans la plupart des applications, le DistanceBased filtrage permet de réduire l'effet des inexactitudes de localisation lors de l'affichage de la trajectoire des appareils sur une carte, ainsi que l'effet de rebond de plusieurs événements consécutifs d'entrée et de sortie lorsque les appareils se trouvent à proximité de la limite d'une géofence. Cela peut également contribuer à réduire le coût de votre application, en effectuant moins d'appels pour effectuer une évaluation par rapport à des géofences liées ou pour récupérer les positions des appareils.

- Basé sur le temps — (par défaut) À utiliser lorsque vos appareils envoient des mises à jour de position très fréquemment (plus d'une fois toutes les 30 secondes) et que vous souhaitez réaliser des évaluations de géofences en temps quasi réel sans enregistrer chaque mise à jour. Dans le cadre du TimeBased filtrage, chaque mise à jour de localisation est évaluée par rapport à des collections de géofences liées, mais toutes les mises à jour de localisation ne sont pas stockées. Si votre fréquence de mise à jour est supérieure à 30 secondes, une seule mise à jour toutes les 30 secondes est enregistrée pour chaque ID d'appareil unique.

Note

Tenez compte des coûts de votre application de suivi lorsque vous décidez de votre méthode de filtrage et de la fréquence des mises à jour de position. Vous êtes facturé pour chaque mise à jour de position et une fois pour l'évaluation de la mise à jour de position par rapport à chaque collection de géofences liée. Par exemple, lorsque vous utilisez le filtrage basé sur le temps, si votre traceur est lié à deux collections de géofences, chaque mise à jour de position comptera comme une demande de mise à jour de localisation et deux évaluations de collections de géofences. Si vous signalez des mises à jour de position toutes les 5 secondes pour vos appareils et que vous utilisez un filtrage basé sur le temps, 720 mises à jour de localisation et 1 440 évaluations de géofences par heure vous seront facturées pour chaque appareil.

Votre facture n'est pas affectée par le nombre de géofences dans chaque collection. Étant donné que chaque collection de géofences peut contenir jusqu'à 50 000 géofences, vous souhaitez peut-être combiner vos géofences en un moins grand nombre de collections, dans la mesure du possible, afin de réduire le coût des évaluations de géofences.

Par défaut, vous recevrez EventBridge des événements chaque fois qu'un appareil suivi entre ou sort d'une géofence liée. Pour plus d'informations, consultez [Associer un traceur à une collection de géofences](#).

Vous pouvez activer des événements pour toutes les mises à jour de position filtrées pour une ressource de suivi. Pour plus d'informations, consultez [Activer les événements de mise à jour pour un tracker](#).

Note

Si vous souhaitez chiffrer vos données à l'aide de votre propre clé gérée par le AWS KMS client, la fonctionnalité Bounding Polygon Queries sera désactivée par défaut. En effet, en utilisant cette fonctionnalité de requêtes polygonales de délimitation, aucune représentation des positions de votre appareil ne sera cryptée à l'aide de votre clé AWS KMS gérée. Cependant, la position exacte de l'appareil est toujours cryptée à l'aide de votre clé gérée. Vous pouvez choisir d'activer la fonctionnalité Bounding Polygon Queries en définissant le `KmsKeyEnableGeospatialQueries` paramètre sur `true` lors de la création ou de la mise à jour d'un outil de suivi.

Console

Pour créer un outil de suivi à l'aide de la console Amazon Location

1. Ouvrez la console Amazon Location Service à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Dans le volet de navigation de gauche, choisissez Trackers.
3. Choisissez Créer un tracker.
4. Renseignez les champs suivants :
 - Nom — Entrez un nom unique. Par exemple, *ExampleTracker*. 100 caractères maximum. Les entrées valides incluent les caractères alphanumériques, les traits d'union, les points et les traits de soulignement.
 - Description — Entrez une description facultative.
5. Sous Filtrage des positions, choisissez l'option qui correspond le mieux à la manière dont vous avez l'intention d'utiliser votre ressource de suivi. Si vous ne définissez pas le filtrage de position, le paramètre par défaut est TimeBased. Pour plus d'informations, consultez [Traceurs](#) ce guide et [PositionFiltering](#) le manuel Amazon Location Service Trackers API Reference.
6. (Facultatif) Sous Tags (Balises), entrez des paramètres Key (Clé) et Value (Valeur) de balise. Cela ajoute un tag à votre nouvelle collection de géofences. Pour plus d'informations, veuillez consulter la rubrique [Balisage de vos ressources](#).
7. (Facultatif) Sous Chiffrement de clé géré par le client, vous pouvez choisir d'ajouter une clé gérée par le client. Cela ajoute une clé symétrique gérée par le client que vous créez, détenez et gérez par rapport au chiffrement AWS détenu par défaut. Pour plus d'informations, consultez la section [Chiffrement des données au repos](#).
8. (Facultatif) Sous KmsKeyEnableGeospatialQueries, vous pouvez choisir d'activer les requêtes géospatiales. Cela vous permet d'utiliser la fonctionnalité Bounding Polygon Queries, tout en chiffrant vos données à l'aide d'une clé gérée par AWS KMS du client.

Note

Lorsque vous utilisez la fonctionnalité Bounding Polygon Queries, la représentation des positions de votre appareil n'est pas cryptée à l'aide de votre clé AWS KMS gérée. Cependant, la position exacte de l'appareil est toujours cryptée à l'aide de votre clé gérée.

9. (Facultatif) Dans le cadre EventBridge de la configuration, vous pouvez choisir d'activer les EventBridge événements pour les mises à jour de position filtrées. Cela enverra un événement chaque fois qu'une mise à jour de position d'un appareil dans ce tracker répond à l'évaluation du filtrage de position.
10. Choisissez Créer un tracker.

API

Pour créer un outil de suivi à l'aide des API Amazon Location

Utilisez l'[CreateTracker](#) opération depuis les API Amazon Location Trackers.

L'exemple suivant utilise une demande d'API pour créer un outil de suivi appelé *ExampleTracker*. La ressource de suivi est associée à une [AWS KMS clé gérée par le client pour chiffrer les données du client](#) et ne [permet pas les mises à jour de position dans 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"
  }
}
```

Créez un tracker avec **KmsKeyEnableGeospatialQueries** activé

Dans l'exemple suivant, le paramètre est `KmsKeyEnableGeospatialQueries` défini sur `true`. Cela vous permet d'utiliser la fonctionnalité Bounding Polygon Queries, tout en chiffrant vos données à l'aide d'une clé gérée par le client AWS KMS .

Pour plus d'informations sur l'utilisation de la fonctionnalité Bounding Polygon Requêtes, voir [???](#)

Note

Lorsque vous utilisez la fonctionnalité Bounding Polygon Queries, la représentation des positions de votre appareil n'est pas cryptée à l'aide de votre clé AWS KMS gérée. Cependant, la position exacte de l'appareil est toujours cryptée à l'aide de votre clé gérée.

```
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

Pour créer un tracker à l'aide de AWS CLI commandes

Utilisez la commande [create-tracker](#).

L'exemple suivant utilise le AWS CLI pour créer un tracker appelé *ExampleTracker*. La ressource de suivi est associée à une [AWS KMS clé gérée par le client pour chiffrer les données du client](#) et ne [permet pas les mises à jour de position dans 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"
```

Créez un tracker avec `KmsKeyEnableGeospatialQueries` activé

Dans l'exemple suivant, le paramètre `KmsKeyEnableGeospatialQueries` est défini sur `true`. Cela vous permet d'utiliser la fonctionnalité Bounding Polygon Queries, tout en chiffrant vos données à l'aide d'une clé gérée par le client AWS KMS .

Pour plus d'informations sur l'utilisation de la fonctionnalité Bounding Polygon Requêtes, voir [???](#)

Note

Lorsque vous utilisez la fonctionnalité Bounding Polygon Queries, la représentation des positions de votre appareil n'est pas cryptée à l'aide de votre clé AWS KMS gérée. Cependant, la position exacte de l'appareil est toujours cryptée à l'aide de votre clé gérée.

```
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 facturation dépend de votre utilisation. L'utilisation d'autres AWS services peut entraîner des frais. Pour plus d'informations, consultez la [tarification d'Amazon Location Service](#).

Vous pouvez modifier la description, le filtrage des positions et la EventBridge configuration une fois le tracker créé en choisissant Modifier le tracker.

Authentification de vos demandes

Une fois que vous avez créé une ressource de suivi et que vous êtes prêt à commencer à évaluer les positions des appareils par rapport aux géofences, choisissez la manière dont vous souhaitez authentifier vos demandes :

- Pour découvrir comment accéder aux services, consultez la section [Accès à Amazon Location Service](#).
- Si vous souhaitez publier les positions des appareils à l'aide de demandes non authentifiées, vous pouvez utiliser Amazon Cognito.

Exemple

L'exemple suivant montre l'utilisation d'un pool d'identités Amazon Cognito à des fins d'autorisation, à l'aide du [JavaScript SDK AWS v3](#) et de l'emplacement Amazon. [JavaScript Aide à l'authentification](#)

```
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);
```

Mettez à jour votre traceur avec la position de l'appareil

Pour suivre vos appareils, vous pouvez publier des mises à jour de position des appareils sur votre traceur. Vous pouvez ensuite récupérer ces positions de l'appareil ou l'historique des positions de l'appareil à partir de votre ressource de suivi.

Chaque mise à jour de position doit inclure l'identifiant de l'appareil, un horodatage et une position. Vous pouvez éventuellement inclure d'autres métadonnées, notamment la précision et jusqu'à 3 paires clé-valeur pour votre propre usage.

Si votre traceur est lié à une ou plusieurs collections de géofences, les mises à jour seront évaluées par rapport à ces géofences (conformément aux règles de filtrage que vous avez spécifiées pour le traceur). Si un appareil franchit une zone géo-clôturée (en se déplaçant de l'intérieur de la zone vers l'extérieur, ou vice versa), vous recevrez des événements à l'intérieur. EventBridge Ces EXIT événements ENTER ou événements incluent les détails de mise à jour de la position, y compris l'identifiant de l'appareil, l'horodatage et toutes les métadonnées associées.

Note

Pour plus d'informations sur le filtrage des positions, consultez [Création d'un tracker](#).
Pour plus d'informations sur les événements de géofence, consultez [Réagir aux événements d'Amazon Location Service avec Amazon EventBridge](#)

Utilisez l'une des méthodes suivantes pour envoyer les mises à jour de l'appareil :

- [Envoyez des mises à jour MQTT](#) à une ressource AWS IoT Core et associez-la à votre ressource de suivi.
- Envoyez des mises à jour de localisation à l'aide de l'API Amazon Location Trackers AWS CLI, ou des API Amazon Location. Vous pouvez utiliser les [AWS SDK](#) pour appeler les API depuis votre application iOS ou Android.

API

Pour envoyer une mise à jour de position à l'aide des API Amazon Location

Utilisez l'[BatchUpdateDevicePosition](#) opération depuis les API Amazon Location Trackers.

L'exemple suivant utilise une demande d'API pour publier une mise à jour de la position de l'appareil *ExampleDevices* sur un tracker *ExampleTracker*.

```
POST /tracking/v0/trackers/ExampleTracker/positions
```

```
Content-type: application/json
```

```
{
  "Updates": [
    {
      "DeviceId": "1",
      "Position": [
        -123.12245146162303, 49.27521118043802
      ],
      "SampleTime": "2022-10-24T19:09:07.327Z",
      "PositionProperties": {
        "name" : "device1"
      },
      "Accuracy": {
        "Horizontal": 10
      }
    },
    {
      "DeviceId": "2",
      "Position": [
        -123.1230104928471, 49.27752402723152
      ],
      "SampleTime": "2022-10-02T19:09:07.327Z"
    },
    {
      "DeviceId": "3",
      "Position": [
        -123.12325592118916, 49.27340530543111
      ],
      "SampleTime": "2022-10-02T19:09:07.327Z"
    },
    {
      "DeviceId": "4",
      "Position": [
```

```

-123.11958813096311, 49.27774641063121
],
"SampleTime": "2022-10-02T19:09:07.327Z"
},
{
"DeviceId": "5",
"Position": [
-123.1277418058896, 49.2765989015285
],
"SampleTime": "2022-10-02T19:09:07.327Z"
},
{
"DeviceId": "6",
"Position": [
-123.11964267059481, 49.274188155916534
],
"SampleTime": "2022-10-02T19:09:07.327Z"
}
]
}

```

AWS CLI

Pour envoyer une mise à jour de position à l'aide de AWS CLI commandes

Utilisez la commande [batch-update-device-position](#).

L'exemple suivant utilise un AWS CLI pour publier une mise à jour de la position de l'appareil pour *ExampleDevice-1* et *ExampleDevice-2* sur un tracker *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"}}]'

```

Obtenir l'historique de localisation d'un appareil à partir d'un traceur

Votre ressource Amazon Location Tracker conserve l'historique de localisation de tous vos appareils suivis pendant une période de 30 jours. Vous pouvez récupérer l'historique de localisation de

l'appareil, y compris toutes les métadonnées associées, à partir de votre ressource de suivi. Les exemples suivants utilisent AWS CLI les ou les API Amazon Location.

API

Pour obtenir l'historique de localisation de l'appareil à partir d'un outil de suivi à l'aide des API Amazon Location

Utilisez l'[GetDevicePositionHistory](#) opération depuis les API Amazon Location Trackers.

L'exemple suivant utilise une demande d'URI d'API pour obtenir l'historique de localisation de l'appareil *ExampleDevice* à partir d'un tracker appelé en *ExampleTracker* commençant par 19:05:07 (inclus) et se terminant à 19:20:07 (exclusif) le 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

Pour obtenir l'historique de localisation de l'appareil à partir d'un traceur à l'aide de AWS CLI commandes

Utilisez la commande [get-device-position-history](#).

L'exemple suivant utilise an AWS CLI pour obtenir l'historique de localisation de l'appareil *ExampleDevice* à partir d'un tracker appelé en *ExampleTracker* commençant par 19:05:07 (inclus) et se terminant à 19:20:07 (exclusif) le 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"
```

Listez les positions de votre appareil

Vous pouvez consulter la liste des positions des appareils d'un outil de suivi à l' AWS CLI aide de l'API ou des ListDevicePositions API Amazon Location. Lorsque vous appelez l' ListDevicePositions API, une liste des dernières positions de tous les appareils associés à un tracker donné est renvoyée. Par défaut, cette API renvoie 100 des dernières positions des appareils par page de résultats pour un tracker donné. Pour renvoyer uniquement des appareils situés dans une région spécifique, utilisez le FilterGeometry paramètre pour créer une requête de polygone de délimitation. Ainsi, lorsque vous appelez ListDevicePositions, seuls les appareils situés à l'intérieur du polygone seront renvoyés.

Note

Si vous souhaitez chiffrer vos données à l'aide de votre propre clé gérée par le AWS KMS client, la fonctionnalité Bounding Polygon Queries sera désactivée par défaut. En effet, en utilisant cette fonctionnalité, aucune représentation des positions de votre appareil ne sera cryptée à l'aide de votre clé AWS KMS gérée. Cependant, la position exacte de l'appareil est toujours cryptée à l'aide de votre clé gérée.

Vous pouvez choisir d'activer la fonctionnalité Bounding Polygon Queries. Pour ce faire, définissez le KmsKeyEnableGeospatialQueries paramètre sur true lors de la création ou de la mise à jour d'un Tracker.

API

Utilisez l'[ListDevicePositions](#) opération depuis les API Amazon Location Trackers.

L'exemple suivant est une demande d'API pour obtenir une liste des positions des appareils dans une zone polygonale, à l'aide du paramètre facultatif. [FilterGeometry](#) L'exemple renvoie 3 emplacements de périphériques présents dans la zone définie par le Polygon tableau.

```
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"  
}
```

Voici un exemple de réponse pour [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

Utilisez la commande [list-trackers](#).

L'exemple suivant permet d' AWS CLI obtenir une liste de périphériques dans une zone polygonale.

```
aws location list-device-positions TODO: add arguments add props for filter geo
```

Associer un traceur à une collection de géofences

Maintenant que vous disposez d'une collection de géofences et d'un traceur, vous pouvez les relier afin que les mises à jour de localisation soient automatiquement évaluées par rapport à toutes vos géofences. Si vous ne souhaitez pas évaluer toutes les mises à jour de localisation, ou si vous ne stockez pas certaines de vos positions dans une ressource de suivi, vous pouvez [évaluer les positions des appareils par rapport aux géofences](#) à la demande.

Lorsque les positions des appareils sont évaluées par rapport aux géofences, des événements sont générés. Vous pouvez définir une action pour ces événements. Pour plus d'informations sur les actions que vous pouvez définir pour les événements de géofence, consultez [Réagir aux événements Amazon Location Service avec Amazon EventBridge](#).

Un événement Amazon Location inclut les attributs de la mise à jour de position de l'appareil qui le génère et certains attributs de la barrière géographique entrée ou sortie. Pour plus d'informations sur les données incluses dans un événement de géofence, consultez [Exemples EventBridge d'événements Amazon pour Amazon Location Service](#).

Les exemples suivants relient une ressource de suivi à une collection de géofences à l'aide de la console, des API Amazon Location ou des API Amazon Location. AWS CLI

Console

Pour associer une ressource de suivi à une collection de géofences à l'aide de la console Amazon Location Service

1. Ouvrez la console Amazon Location Service à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Dans le volet de navigation de gauche, choisissez Trackers.
3. Sous Device trackers, sélectionnez le lien du nom du tracker cible.
4. Sous Collections Geofence liées, choisissez Link Geofence Collection.
5. Dans la fenêtre Linked Geofence Collection, sélectionnez une collection de géofences dans le menu déroulant.
6. Choisissez Lier.

Une fois que vous aurez lié la ressource de suivi, le statut Actif lui sera attribué.

API

Pour associer une ressource de suivi à une collection de géofences à l'aide des API Amazon Location

Utilisez l'[AssociateTrackerConsumer](#) opération depuis les API Amazon Location Trackers.

L'exemple suivant utilise une demande d'API associée à une collection de géofences à l'aide de son [Amazon Resource Name](#) (ARN). *ExampleTracker*

```
POST /tracking/v0/trackers/ExampleTracker/consumers
Content-type: application/json

{
  "ConsumerArn": "arn:aws:geo:us-west-2:123456789012:geofence-
collection/ExampleGeofenceCollection"
}
```

AWS CLI

Pour lier une ressource de suivi à une collection de géofences à l'aide de commandes AWS CLI

Utilisez la commande [associate-tracker-consumer](#).

L'exemple suivant utilise un AWS CLI pour créer une collection de géofences appelée.

ExampleGeofenceCollection

```
aws location \
  associate-tracker-consumer \
    --consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-
collection/ExampleGeofenceCollection" \
    --tracker-name "ExampleTracker"
```

Évaluez la position des appareils par rapport aux géofences

Il existe deux méthodes pour évaluer les positions par rapport aux géofences afin de générer des événements de géofences :

- Vous pouvez lier les traceurs et les collections Geofence. Pour plus d'informations, consultez la section : [Associer un traceur à une collection de géofences](#)
- Vous pouvez envoyer une demande directe à la ressource de collecte de géofences pour évaluer une ou plusieurs positions à l'aide de l'[BatchEvaluateGeofences](#)API.

En outre, vous pouvez prévoir les événements de géofence entrants pour un appareil entrant, sortant ou restant inactif à l'intérieur d'une géofence. Utilisez l'[ForecastGeofenceEvents](#)API pour prévoir les événements.

Si vous souhaitez également suivre l'historique des positions de votre appareil ou afficher des positions sur une carte, associez le traceur à une collection de géofences. Il se peut également

que vous ne souhaitiez pas évaluer toutes les mises à jour de localisation ou que vous n'ayez pas l'intention de stocker les données de localisation dans une ressource de suivi. Dans l'un ou l'autre cas, vous pouvez adresser une demande directe à la collection de géofences et évaluer la position d'un ou de plusieurs appareils par rapport à ses géofences.

L'évaluation de la position des appareils par rapport aux géofences génère des événements. Vous pouvez réagir à ces événements et les rediriger vers d'autres AWS services. Pour plus d'informations sur les mesures que vous pouvez prendre lors de la réception d'événements de géofence, consultez [Réagir aux événements Amazon Location Service with Amazon EventBridge](#).

Un événement Amazon Location inclut les attributs de la mise à jour de la position de l'appareil qui le génère, notamment l'heure, la position, la précision et les métadonnées des valeurs clés, ainsi que certains attributs de la barrière géographique saisie ou sortie. Pour plus d'informations sur les données incluses dans un événement de géofence, consultez [Exemples EventBridge d'événements Amazon pour Amazon Location Service](#).

Les exemples suivants utilisent AWS CLI les ou les API Amazon Location.

API

Pour évaluer la position des appareils par rapport à la position des barrières géographiques à l'aide des API Amazon Location

Utilisez l'[BatchEvaluateGeofences](#) opération depuis les API Amazon Location Geofences.

L'exemple suivant utilise une demande d'API pour évaluer la position de l'appareil dans une *ExampleDevice* collection de géofences associée. *ExampleGeofenceCollection* Remplacez ces valeurs par vos propres géofences et identifiants d'appareils.

```
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

Pour évaluer la position des appareils par rapport à la position des barrières géographiques à l'aide de commandes AWS CLI

Utilisez la commande [batch-evaluate-geofences](#).

L'exemple suivant utilise un AWS CLI pour évaluer la position de par *ExampleDevice* rapport à une collection de géofences associée. *ExampleGeofenceCollection* Remplacez ces valeurs par vos propres géofences et identifiants d'appareils.

```
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"}} ]'
```

L'évaluation de la position des appareils par rapport aux géofences génère des événements. Traditionnellement, vous pouvez réagir aux événements en utilisant [Amazon EventBridge](#), mais ce processus ne vous permet de réagir aux événements qu'après qu'ils se soient produits. Si vous devez anticiper le moment où un appareil entre ou sort d'une barrière géographique, par exemple si un appareil franchit une frontière et sera donc soumis à une réglementation différente, vous pouvez utiliser l'[ForecastGeofenceEvents](#) API pour prévoir les futurs événements de géofence.

L'[ForecastGeofenceEvents](#) API utilise des critères tels que l'appareil, la proximité time-to-breach, la vitesse et la position pour prévoir les événements. L'API renverra un `ForecastedBreachTime`, qui indique l'heure estimée à laquelle l'événement de géofence se produira.

L'exemple suivant utilise les API Amazon Location.

API

Pour prévoir les événements de géofence à l'aide des API Amazon Location

Utilisez l' [ForecastGeofenceEvents](#) opération depuis les API Amazon Location Geofences.

L'exemple suivant utilise une demande d'API pour prévoir les événements de géofence d'un *ExampleDevice* parent à un. *ExampleGeofence* Remplacez ces valeurs par vos propres géofences et identifiants d'appareils.

```
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
}
```

Vérifier la position des appareils

Pour vérifier l'intégrité de la position d'un appareil, utilisez l'[VerifyDevicePosition](#) API. Cette API renvoie des informations sur l'intégrité de la position de l'appareil, en évaluant des propriétés telles que le signal cellulaire de l'appareil, le point d'accès Wi-Fi, l'adresse IPv4 et si un proxy est utilisé.

Prérequis

Avant de pouvoir utiliser les API répertoriées pour la vérification des appareils, assurez-vous de remplir les conditions préalables suivantes :

- Vous avez créé un outil de suivi pour le ou les appareils que vous souhaitez vérifier. Pour plus d'informations, consultez [Commencer le suivi](#).

L'exemple suivant montre une demande pour l'[VerifyDevicePosition](#) API Amazon Location.

API

Pour vérifier la position des appareils à l'aide des API Amazon Location

Utilisez l'[VerifyDevicePosition](#) opération depuis les API de suivi de localisation Amazon.

L'exemple suivant montre une demande d'API pour évaluer l'intégrité de la position d'un appareil. Remplacez ces valeurs par les identifiants de vos propres appareils.

```
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

Le SDK Location Integrity fournit des fonctionnalités améliorées liées à la vérification des appareils et peut être utilisé sur demande. Pour accéder au SDK, contactez le [Support commercial](#).

Réagir aux événements d'Amazon Location Service avec Amazon EventBridge

Amazon EventBridge est un bus d'événements sans serveur qui connecte efficacement les applications entre elles à l'aide de données provenant de AWS services tels qu'Amazon Location. EventBridge reçoit les événements d'Amazon Location et achemine ces données vers des cibles telles que AWS Lambda. Vous pouvez configurer des règles de routage pour déterminer où envoyer vos données afin de créer des architectures d'applications qui réagissent en temps réel.

Seuls les événements de géofence (ENTER et les EXIT événements lorsque les appareils entrent ou sortent des zones géo-clôturées) sont envoyés par défaut. EventBridge Vous pouvez également activer tous les événements de mise à jour de position filtrés pour une ressource de suivi. Pour plus d'informations, consultez [Activer les événements de mise à jour pour un tracker](#).

Pour plus d'informations, consultez [les événements et les modèles d'événements](#) du guide de EventBridge l'utilisateur Amazon.

Rubriques

- [Activer les événements de mise à jour pour un tracker](#)
- [Création de règles d'événement pour Amazon Location](#)
- [Exemples EventBridge d'événements Amazon pour Amazon Location Service](#)

Activer les événements de mise à jour pour un tracker

Par défaut, Amazon Location envoie uniquement des événements ENTER et EXIT géofence à EventBridge. Vous pouvez activer tous les UPDATE événements de position filtrés auxquels un tracker doit être envoyé EventBridge. Vous pouvez le faire lorsque vous [créez](#) ou [mettez à jour](#) un outil de suivi.

Par exemple, pour mettre à jour un outil de suivi existant à l'aide de AWS CLI, vous pouvez utiliser la commande suivante (utilisez le nom de votre ressource de suivi à la place de *MyTracker*).

```
aws location update-tracker --tracker-name MyTracker --event-bridge-enabled
```

Pour désactiver les événements de position pour un tracker, vous devez utiliser l'API ou la console Amazon Location Service.

Création de règles d'événement pour Amazon Location

Vous pouvez créer [jusqu'à 300 règles par bus d'événement](#) EventBridge pour configurer les actions entreprises en réponse à un événement Amazon Location.

Par exemple, vous pouvez créer une règle pour les événements de géolocalisation selon laquelle une notification push sera envoyée lorsqu'un téléphone est détecté à l'intérieur d'une limite de géolocalisation.

Pour créer une règle pour les événements Amazon Location

À l'aide des valeurs suivantes, [créez une EventBridge règle](#) basée sur les événements Amazon Location :

- Pour Type de règle, choisissez Règle avec un modèle d'événement.
- Dans la zone Modèle d'événement, ajoutez le modèle suivant :

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Geofence Event"]
}
```

Pour créer une règle pour les mises à jour de la position du tracker, vous pouvez plutôt utiliser le modèle suivant :

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Device Position Event"]
}
```

Vous pouvez éventuellement spécifier uniquement ENTER EXIT des événements en ajoutant un `detail` tag (si votre règle concerne les mises à jour de la position du tracker, il n'y en a qu'un seul `eventType`, il n'est donc pas nécessaire de filtrer sur celui-ci) :

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Geofence Event"],
  "detail": {
    "eventType": ["ENTER"]
  }
}
```

Vous pouvez également éventuellement filtrer sur les propriétés de la position ou de la géofence :

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Geofence Event"],
  "detail": {
    "eventType": ["ENTER"],
    "GeofenceProperties": {
      "Type": "LoadingDock"
    },
    "PositionProperties": {
      "VehicleType": "Truck"
    }
  }
}
```

- Pour `Select targets`, choisissez l'action cible à effectuer lorsqu'un événement est reçu d'Amazon Location Service.

Par exemple, utilisez une rubrique Amazon Simple Notification Service (SNS) pour envoyer un e-mail ou un SMS lorsqu'un événement se produit. Vous devez d'abord créer une rubrique Amazon SNS à l'aide de la console Amazon SNS. Pour plus d'informations, consultez la section [Utilisation d'Amazon SNS pour les notifications aux utilisateurs](#).

⚠ Warning

Il est recommandé de vérifier que la règle de l'événement a été correctement appliquée, faute de quoi votre action automatisée risque de ne pas démarrer comme prévu. Pour vérifier votre règle d'événement, initiez les conditions de la règle d'événement. Par exemple, simulez l'entrée d'un appareil dans une zone géo-clôturée.

Vous pouvez également capturer tous les événements depuis Amazon Location, en excluant simplement la `detail-type` section. Par exemple :

```
{
  "source": [
    "aws.geo"
  ]
}
```

ℹ Note

Le même événement peut être organisé plusieurs fois. Vous pouvez utiliser l'identifiant d'événement pour dédupliquer les événements que vous recevez.

Exemples EventBridge d'événements Amazon pour Amazon Location Service

Voici un exemple d'événement permettant de saisir une géofence initiée par un appel.

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"
  }
}
}
}

```

Voici un exemple d'événement de sortie d'une géofence initié par un appel.

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"
}
}
}

```

Voici un exemple d'événement pour une mise à jour de position, initiée par un appel `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"
  }
}
}
```

Suivi à l'aide AWS IoT de MQTT avec Amazon Location Service

Le [MQTT](#) est un protocole de messagerie léger et largement adopté conçu pour les appareils limités. AWS IoT Core prend en charge les connexions de périphériques utilisant le protocole MQTT et le protocole MQTT over WebSocket Secure (WSS).

[AWS IoT Core](#) connecte les appareils à eux AWS et vous permet d'envoyer et de recevoir des messages entre eux. Le moteur de AWS IoT Core règles stocke les requêtes concernant les sujets des messages de vos appareils et vous permet de définir des actions pour envoyer des messages à d'autres AWS services, tels qu'Amazon Location Service. Les appareils qui connaissent leur position sous forme de coordonnées peuvent voir leur position transmise à Amazon Location via le moteur de règles.

Note

Les appareils peuvent connaître leur propre position, par exemple via le GPS intégré. AWS IoT prend également en charge le suivi de la localisation des appareils tiers. Pour plus d'informations, consultez la section [Emplacement du périphérique AWS IoT principal](#) dans le Guide du développeur principal.

La procédure pas à pas suivante décrit le suivi à l'aide de AWS IoT Core règles. Vous pouvez également envoyer les informations de l'appareil à votre propre AWS Lambda fonction, si vous devez les traiter avant de les envoyer à Amazon Location. Pour plus de détails sur l'utilisation de Lambda pour le traitement de la localisation de vos appareils, consultez. [Utilisation AWS Lambda avec MQTT](#)

Rubriques

- [Prérequis](#)

- [Créer une règle de AWS IoT Core](#)
- [Testez votre AWS IoT Core règle dans la console](#)
- [Utilisation AWS Lambda avec MQTT](#)

Prérequis

Avant de commencer le suivi, vous devez remplir les conditions préalables suivantes :

- [Créez une ressource de suivi](#) à laquelle vous enverrez les données de localisation de l'appareil.
- [Créez un rôle IAM](#) pour accorder l'AWS IoT Core accès à votre tracker.

Lorsque vous suivez ces étapes, appliquez la politique suivante pour donner accès à votre traceur :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "WriteDevicePosition",
      "Effect": "Allow",
      "Action": "geo:BatchUpdateDevicePosition",
      "Resource": "arn:aws:geo:*:*:tracker/*"
    }
  ]
}
```

Créer une règle de AWS IoT Core

Créez ensuite une AWS IoT Core règle pour transmettre la télémétrie de position de vos appareils à Amazon Location Service. Pour plus d'informations sur la création de règles, consultez les rubriques suivantes du guide du AWS IoT Core développeur :

- [Création d'une AWS IoT règle](#) pour obtenir des informations sur la création d'une nouvelle règle.
- [Action de localisation](#) pour obtenir des informations spécifiques à la création d'une règle de publication sur Amazon Location

Testez votre AWS IoT Core règle dans la console

Si aucun appareil ne publie actuellement de télémétrie incluant la localisation, vous pouvez tester votre règle à l'aide de la AWS IoT Core console. La console dispose d'un client de test dans lequel vous pouvez publier un exemple de message pour vérifier les résultats de la solution.

1. Connectez-vous à la AWS IoT Core console à l'adresse <https://console.aws.amazon.com/iot/>.
2. Dans le volet de navigation de gauche, développez Test, puis choisissez le client de test MQTT.
3. Sous Publier dans un sujet, définissez le nom du sujet sur *iot/topic* (ou le nom du sujet que vous avez défini dans votre AWS IoT Core règle, s'il est différent) et fournissez les informations suivantes pour la charge utile du message. Remplacez l'horodatage *1604940328* par un horodatage valide au cours des 30 derniers jours (les horodatages antérieurs à 30 jours sont ignorés par les dispositifs de suivi Amazon Location Service).

```
{
  "payload": {
    "deviceid": "thing123",
    "timestamp": 1604940328,
    "location": { "lat": 49.2819, "long": -123.1187 },
    "accuracy": { "Horizontal": 20.5 },
    "positionProperties": { "field1": "value1", "field2": "value2" }
  }
}
```

4. Choisissez Publier dans le sujet pour envoyer le message de test.
5. Pour vérifier que le message a bien été reçu par Amazon Location Service, utilisez la AWS CLI commande suivante. Si vous l'avez modifié lors de la configuration, remplacez le nom du tracker par celui que vous avez utilisé.

```
aws location batch-get-device-position --tracker-name MyTracker --device-ids
thing123
```

Utilisation AWS Lambda avec MQTT

Bien que l'utilisation de AWS Lambda soit plus obligatoire lors de l'envoi des données de localisation de l'appareil à Amazon Location à des fins de suivi, vous souhaitez peut-être toujours utiliser Lambda dans certains cas. Par exemple, si vous souhaitez traiter vous-même les données de localisation de votre appareil avant de les envoyer à Amazon Location. Les rubriques suivantes

décrivent comment utiliser Lambda pour traiter les messages avant de les envoyer à votre tracker. Pour plus d'informations sur ce modèle, consultez l'[architecture de référence](#).

Rubriques

- [Prérequis](#)
- [Création d'une fonction Lambda](#)
- [Créer une règle de AWS IoT Core](#)
- [Testez votre AWS IoT Core règle dans la console](#)

Prérequis

Avant de commencer le suivi, vous devez [créer une ressource de suivi](#). Pour créer une ressource de suivi, vous pouvez utiliser la console Amazon Location, l'AWS CLI, ou les API Amazon Location.

L'exemple suivant utilise la console Amazon Location Service pour créer la ressource de suivi :

1. Ouvrez la console Amazon Location Service à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Dans le volet de navigation de gauche, choisissez Trackers.
3. Choisissez Créer un tracker.
4. Remplissez les cases suivantes :
 - Nom — Entrez un nom unique comportant un maximum de 100 caractères. Les entrées valides incluent les caractères alphanumériques, les traits d'union et les traits de soulignement. Par exemple, *MyTracker*.
 - Description — Entrez une description facultative. Par exemple, *Tracker pour enregistrer les positions des AWS IoT Core appareils*.
 - Filtrage des positions : sélectionnez le filtrage que vous souhaitez utiliser pour les mises à jour des positions. Par exemple, le filtrage basé sur la précision.
5. Choisissez Créer un tracker.

Création d'une fonction Lambda

Pour créer une connexion entre Amazon Location Service AWS IoT Core et Amazon, vous avez besoin d'une AWS Lambda fonction permettant de traiter les messages transférés par AWS IoT Core. Cette fonction permet d'extraire toutes les données de position, de les formater pour Amazon

Location Service et de les envoyer via l'API Amazon Location Tracker. Vous pouvez créer cette fonction via la AWS Lambda console, ou vous pouvez utiliser le AWS Command Line Interface (AWS CLI) ou les AWS Lambda API.

Pour créer une fonction Lambda qui publie des mises à jour de position sur Amazon Location à l'aide de la console :

1. Ouvrez la console AWS Lambda à l'adresse <https://console.aws.amazon.com/lambda/>.
2. Dans le menu de navigation de gauche, choisissez Fonctions.
3. Choisissez Créer une fonction et assurez-vous que l'option Auteur à partir de zéro est sélectionnée.
4. Remplissez les cases suivantes :
 - Nom de la fonction — Entrez un nom unique pour votre fonction. Les entrées valides incluent les caractères alphanumériques, les traits d'union et les traits de soulignement sans espaces. Par exemple, *MyLambda*.
 - Runtime — Choisissez *Python 3.8*.
5. Choisissez Créer une fonction.
6. Cliquez sur l'onglet Code pour ouvrir l'éditeur.
7. [Remplacez le code d'espace réservé `lambda_function.py` par le code suivant, en remplaçant la valeur attribuée `TRACKER_NAME` par le nom du tracker que vous avez créé comme condition préalable.](#)

```
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. Choisissez Déployer pour enregistrer la fonction mise à jour.
9. Cliquez sur l'onglet Configuration.
10. Dans la section Autorisations, choisissez le nom du rôle en lien hypertexte pour accorder à Amazon Location Service des autorisations pour votre fonction Lambda.
11. Sur la page Résumé de votre rôle, choisissez Ajouter des autorisations, puis dans la liste déroulante, sélectionnez Créer une politique intégrée.
12. Choisissez l'onglet JSON et remplacez la politique par le document suivant. Cela permet à votre fonction Lambda de mettre à jour les positions des appareils gérées par toutes les ressources de suivi dans toutes les régions.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "WriteDevicePosition",
      "Effect": "Allow",
      "Action": "geo:BatchUpdateDevicePosition",
      "Resource": "arn:aws:geo:*:*:tracker/*"
    }
  ]
}
```

13. Choisissez Examiner une politique.
14. Entrez un nom de stratégie. Par exemple, *AmazonLocationTrackerWriteOnly*.
15. Choisissez Créer une politique.

Vous pouvez modifier ce code de fonction, si nécessaire, pour l'adapter au schéma de message de votre appareil.

Créer une règle de AWS IoT Core

Créez ensuite une AWS IoT Core règle pour transmettre la télémétrie de position de vos appareils à la AWS Lambda fonction de transformation et de publication sur Amazon Location Service. L'exemple de règle fourni suppose que toute transformation nécessaire des charges utiles des appareils est gérée par votre fonction Lambda. Vous pouvez créer cette règle via la AWS IoT Core console, le AWS Command Line Interface (AWS CLI) ou les AWS IoT Core API.

Note

Bien que la AWS IoT console gère les autorisations nécessaires AWS IoT Core pour appeler votre fonction Lambda, si vous créez votre règle à partir du SDK AWS CLI ou du SDK, vous devez [configurer une politique à laquelle accorder](#) l'autorisation. AWS IoT

Pour créer un à AWS IoT Core l'aide de la console

1. Connectez-vous à la AWS IoT Core console à l'adresse <https://console.aws.amazon.com/iot/>.
2. Dans le volet de navigation de gauche, développez Act, puis sélectionnez Rules.
3. Choisissez Créer une règle pour démarrer le nouvel assistant de règles.
4. Saisissez un nom et une description pour la règle.
5. Pour l'instruction de requête Rule, mettez à jour l'FROMattribut pour faire référence à une rubrique dans laquelle au moins un appareil publie des données de télémétrie incluant la localisation. Si vous testez la solution, aucune modification n'est nécessaire.

```
SELECT * FROM 'iot/topic'
```

6. Sous Définir une ou plusieurs actions, choisissez Ajouter une action.
7. Sélectionnez Envoyer un message à une fonction Lambda.
8. Choisissez Configurer une action.
9. Recherchez et sélectionnez votre fonction Lambda dans la liste.
10. Choisissez Add action.
11. Choisissez Créer une règle.

Testez votre AWS IoT Core règle dans la console

Si aucun appareil ne publie actuellement de télémétrie incluant la localisation, vous pouvez tester votre règle et cette solution à l'aide de la AWS IoT Core console. La console dispose d'un client de test dans lequel vous pouvez publier un exemple de message pour vérifier les résultats de la solution.

1. Connectez-vous à la AWS IoT Core console à l'adresse <https://console.aws.amazon.com/iot/>.
2. Dans le volet de navigation de gauche, développez Test, puis choisissez le client de test MQTT.
3. Sous Publier dans un sujet, définissez le nom du sujet sur **iot/topic** (ou le nom du sujet que vous avez défini dans votre AWS IoT Core règle, s'il est différent) et fournissez les informations

suivantes pour la charge utile du message. Remplacez l'horodatage **1604940328** par un horodatage valide au cours des 30 derniers jours (tout horodatage de plus de 30 jours est ignoré).

```
{
  "payload": {
    "deviceid": "thing123",
    "timestamp": 1604940328,
    "location": { "lat": 49.2819, "long": -123.1187 },
    "accuracy": { "Horizontal": 20.5 },
    "positionProperties": { "field1": "value1", "field2": "value2" }
  }
}
```

4. Choisissez Publier dans le sujet pour envoyer le message de test.
5. Pour vérifier que le message a bien été reçu par Amazon Location Service, utilisez la AWS CLI commande suivante. Si vous les avez modifiés lors de la configuration, remplacez le nom du tracker et l'identifiant de l'appareil par ceux que vous avez utilisés.

```
aws location batch-get-device-position --tracker-name MyTracker --device-ids
thing123
```

Gestion de vos ressources de collecte de géofences

Gérez vos collections de géofences à l'aide de la console Amazon LocationAWS CLI, ou des API Amazon Location.

Répertoriez vos ressources de collecte de géofences

Vous pouvez consulter la liste de vos collections de géofences à l'aide de la console Amazon LocationAWS CLI, ou des API Amazon Location :

Console

Pour consulter la liste des collections de géofences à l'aide de la console Amazon Location

1. Ouvrez la console Amazon Location à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Choisissez Geofence collections dans le volet de navigation de gauche.
3. Consultez la liste de vos collections de géofences sous Mes collections de géofences.

API

Utilisez l'[ListGeofenceCollections](#) opération depuis les API Amazon Location Geofences.

L'exemple suivant est une demande d'API pour obtenir une liste des collections de géofences dans le AWS compte.

```
POST /geofencing/v0/list-collections
```

Voici un exemple de réponse pour 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

Utilisez la commande [list-geofence-collections](#).

L'exemple suivant permet d'AWS CLI d'obtenir une liste des collections de géofences présentes dans le AWS compte.

```
aws location list-geofence-collections
```

Obtenez les détails de la collection Geofence

Vous pouvez obtenir des informations sur n'importe quelle ressource de collecte de géofences de votre AWS compte à l'aide de la console Amazon Location AWS CLI, ou des API Amazon Location :

Console

Pour consulter les détails d'une collection de géofences à l'aide de la console Amazon Location

1. Ouvrez la console Amazon Location à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Choisissez Geofence collections dans le volet de navigation de gauche.
3. Sous Mes collections de géofences, sélectionnez le lien du nom de la collection de géofences cible.

API

Utilisez l'[DescribeGeofenceCollection](#) opération depuis les API Amazon Location Geofences.

L'exemple suivant est une demande d'API pour obtenir les détails de la collection Geofence pour *ExampleCollection*

```
GET /geofencing/v0/collections/ExampleCollection
```

Voici un exemple de réponse pour 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

Utilisez la commande [describe-geofence-collection](#).

L'exemple suivant permet d'AWS CLI d'obtenir les détails de la collecte de géofences pour *ExampleCollection*

```
aws location describe-geofence-collection \
  --collection-name "ExampleCollection"
```

Supprimer une collection de géofences

Vous pouvez supprimer une collection de géofences de votre AWS compte à l'aide de la console Amazon Location, AWS CLI, ou des API Amazon Location.

Console

Pour supprimer une collection de géofences à l'aide de la console Amazon Location

Warning

Cette opération supprime définitivement la ressource.

1. Ouvrez la console Amazon Location à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Choisissez Geofence collections dans le volet de navigation de gauche.
3. Sous Ma collection de géofences, sélectionnez la collection de géofences cible.
4. Choisissez Supprimer la collection de géofences.

API

Utilisez l'[DeleteGeofenceCollection](#) opération depuis les API Amazon Location.

L'exemple suivant est une demande d'API pour supprimer la collection Geofence.

ExampleCollection

```
DELETE /geofencing/v0/collections/ExampleCollection
```

Voici un exemple de réponse pour DeleteGeofenceCollection :

```
HTTP/1.1 200
```

CLI

Utilisez la commande [delete-geofence-collection](#).

L'exemple suivant est une AWS CLI commande permettant de supprimer la collection Geofence.

ExampleCollection

```
aws location delete-geofence-collection \  
  --collection-name "ExampleCollection"
```

Répertorier les géofences stockées

Vous pouvez répertorier les géofences stockées dans une collection de géofences spécifiée à l'aide de la console Amazon Location, AWS CLI ou des API Amazon Location.

Console

Pour consulter la liste des géofences à l'aide de la console Amazon Location

1. Ouvrez la console Amazon Location à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Choisissez Geofence collections dans le volet de navigation de gauche.
3. Sous Ma collection de géofences, sélectionnez le lien du nom de la collection de géofences cible.
4. Afficher les géofences dans la collection Geofences sous Geofences

API

Utilisez l'[ListGeofences](#) opération depuis les API Amazon Location Geofences.

L'exemple suivant est une demande d'API pour obtenir une liste des géofences stockées dans la collection de géofences. *ExampleCollection*

```
POST /geofencing/v0/collections/ExampleCollection/list-geofences
```

Voici un exemple de réponse pour 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

Utilisez la commande [list-geofences](#).

L'exemple suivant permet d'obtenir une liste des géofences stockées dans la collection de géofences. *ExampleCollection*

```
aws location list-geofences \
  --collection-name "ExampleCollection"
```

Obtenez des informations sur les géofences

Vous pouvez obtenir les détails d'une géofence spécifique, tels que l'heure de création, l'heure de mise à jour, la géométrie et le statut, à partir d'une collection de géofences à l'aide de la console Amazon Location AWS CLI ou des API Amazon Location.

Console

Pour consulter l'état d'une clôture géographique à l'aide de la console Amazon Location

1. Ouvrez la console Amazon Location à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Choisissez Geofence collections dans le volet de navigation de gauche.
3. Sous Ma collection de géofences, sélectionnez le lien du nom de la collection de géofences cible.
4. Sous Geofences, vous pourrez consulter l'état de vos géofences.

API

Utilisez l'[GetGeofence](#) opération depuis les API Amazon Location Geofences.

L'exemple suivant est une demande d'API pour obtenir les détails de la géofence à partir d'une collection de géofences. *ExampleCollection*

```
GET /geofencing/v0/collections/ExampleCollection/geofences/ExampleGeofence1
```

Voici un exemple de réponse pour 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

Utilisez la commande [get-geofence](#).

L'exemple suivant permet d'AWS CLI d'obtenir les détails de la collecte de géofences pour. *ExampleCollection*

```
aws location get-geofence \
  --collection-name "ExampleCollection" \
  --geofence-id "ExampleGeofence1"
```

Supprimer les géofences

Vous pouvez supprimer des géofences d'une collection de géofences à l'aide de la console Amazon Location, AWS CLI ou des API Amazon Location.

Console

Pour supprimer une barrière géographique à l'aide de la console Amazon Location

⚠ Warning

Cette opération supprime définitivement la ressource.

1. Ouvrez la console Amazon Location à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Choisissez Geofence collections dans le volet de navigation de gauche.
3. Sous Ma collection de géofences, sélectionnez le lien du nom de la collection de géofences cible.
4. Sous Geofences, sélectionnez la géofence cible.
5. Choisissez Supprimer la géofence.

API

Utilisez l'[BatchDeleteGeofence](#) opération depuis les API Amazon Location Geofences.

L'exemple suivant est une demande d'API pour supprimer les géofences de la collection de géofences. *ExampleCollection*

```
POST /geofencing/v0/collections/ExampleCollection/delete-geofences
Content-type: application/json

{
  "GeofenceIds": [ "ExampleGeofence11" ]
}
```

Voici un exemple de réponse positive pour [BatchDeleteGeofence](#).

```
HTTP/1.1 200
```

CLI

Utilisez la commande [batch-delete-geofence](#).

L'exemple suivant est une AWS CLI commande permettant de supprimer les géofences de la collection de géofences. *ExampleCollection*

```
aws location batch-delete-geofence \
```

```
--collection-name "ExampleCollection" \  
--geofence-ids "ExampleGeofence11"
```

Gestion de vos ressources de suivi

Vous pouvez gérer vos traceurs à l'aide de la console Amazon LocationAWS CLI, des API Amazon Location ou des API Amazon Location.

Répertoriez vos traceurs

Vous pouvez consulter votre liste de traceurs à l'aide de la console Amazon LocationAWS CLI, ou des API Amazon Location :

Console

Pour consulter la liste des traceurs existants à l'aide de la console Amazon Location

1. Ouvrez la console Amazon Location à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Choisissez Trackers dans le menu de navigation de gauche.
3. Consultez la liste de vos ressources de suivi sous Mes traceurs.

API

Utilisez l'[ListTrackers](#)opération depuis les API Amazon Location Trackers.

L'exemple suivant est une demande d'API pour obtenir une liste des traceurs de votre AWS compte.

```
POST /tracking/v0/list-trackers
```

Voici un exemple de réponse pour [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

Utilisez la commande [list-trackers](#).

L'exemple suivant permet d'obtenir une liste des traceurs de votre AWS compte.

```
aws location list-trackers
```

Déconnexion d'un traceur d'une collection de géofences

Vous pouvez déconnecter un traceur d'une collection de géofences à l'aide de la console Amazon Location, de l'AWS CLI, ou des API Amazon Location :

Console

Pour dissocier un traceur d'une collection de géofences associée à l'aide de la console Amazon Location

1. Ouvrez la console Amazon Location à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Choisissez Trackers dans le volet de navigation de gauche.
3. Sous Mes traceurs, sélectionnez le lien du nom du traceur cible.
4. Sous Collections de géofences liées, sélectionnez une collection de géofences avec le statut Liée.
5. Choisissez Dissocier.

API

Utilisez l'[DisassociateTrackerConsumer](#) opération depuis les API Amazon Location Trackers.

L'exemple suivant est une demande d'API visant à dissocier un traceur d'une collection de géofences associée.

```
DELETE /tracking/v0/trackers/ExampleTracker/consumers/arn:aws:geo:us-west-2:123456789012:geofence-collection/ExampleCollection
```

Voici un exemple de réponse pour [DisassociateTrackerConsumer](#) :

```
HTTP/1.1 200
```

CLI

Utilisez la commande [disassociate-tracker-consumer](#).

L'exemple suivant est une AWS CLI commande permettant de dissocier un traceur d'une collection de géofences associée.

```
aws location disassociate-tracker-consumer \  
  --consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-collection/  
ExampleCollection" \  
  --tracker-name "ExampleTracker"
```

Obtenir des informations sur le tracker

Vous pouvez obtenir des informations sur n'importe quel traceur de votre AWS compte en utilisant la console Amazon LocationAWS CLI, ou les API Amazon Location.

Console

Pour consulter les détails du traceur à l'aide de la console Amazon Location

1. Ouvrez la console Amazon Location à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Choisissez Trackers dans le menu de navigation de gauche.
3. Sous Mes traceurs, sélectionnez le lien du nom du traceur cible.
4. Consultez les détails du traceur sous Informations.

API

Utilisez l'[DescribeTracker](#) opération depuis les API Amazon Location Tracker.

L'exemple suivant est une demande d'API pour obtenir les détails du tracker *ExampleTracker*.

```
GET /tracking/v0 trackers/ExampleTracker
```

Voici un exemple de réponse pour [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

Utilisez la commande [describe-tracker](#).

L'exemple suivant est une AWS CLI commande permettant d'obtenir les détails du tracker pour *ExampleTracker*.

```
aws location describe-tracker \
  --tracker-name "ExampleTracker"
```

Supprimer un traceur

Vous pouvez supprimer un traceur de votre AWS compte à l'aide de la console Amazon LocationAWS CLI, ou des API Amazon Location :

Console

Pour supprimer une ressource cartographique existante à l'aide de la console Amazon Location

Warning

Cette opération supprime définitivement la ressource. Si la ressource de suivi est en cours d'utilisation, il se peut que vous rencontriez une erreur. Assurez-vous que la ressource cible n'est pas une dépendance pour vos applications.

1. Ouvrez la console Amazon Location à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Choisissez Trackers dans le volet de navigation de gauche.
3. Sous Mes traceurs, sélectionnez le traqueur cible.
4. Choisissez Supprimer le tracker.

API

Utilisez l'[DeleteTracker](#) opération depuis les API Amazon Location Tracker.

L'exemple suivant est une demande d'API pour supprimer le tracker *ExampleTracker*.

```
DELETE /tracking/v0/trackers/ExampleTracker
```

Voici un exemple de réponse pour [DeleteTracker](#) :

```
HTTP/1.1 200
```

CLI

Utilisez la commande [delete-tracker](#).

L'exemple suivant est une AWS CLI commande permettant de supprimer le tracker *ExampleTracker*.

```
aws location delete-tracker \  
  --tracker-name "ExampleTracker"
```

Exemple d'application mobile de géofencing et de suivi

Cette rubrique couvre des didacticiels conçus pour présenter les principales fonctionnalités de l'utilisation des géofences et des dispositifs de suivi Amazon Location dans une application mobile. Les applications montrent comment un traceur et une clôture géographique interagissent à l'aide d'une combinaison des fonctionnalités Lambda et AWS IoT Amazon Location. Deux didacticiels sont disponibles.

- [Exemple d'application de suivi et de géofencing pour Android](https://github.com/aws-geospatial/amazon-location-samples-android/tree/main/.tracking-with-geofence-notifications), et vous pouvez cloner les fichiers du projet depuis GitHub : <https://github.com/aws-geospatial/amazon-location-samples-android/tree/main/.tracking-with-geofence-notifications>

- [Exemple d'application de suivi et de géofencing pour iOS](https://github.com/aws-geospatial/amazon-location-samples-ios/tree/main/tracking-with-geofence-notifications), et vous pouvez cloner les fichiers du projet depuis GitHub : <https://github.com/aws-geospatial/amazon-location-samples-ios/tree/main/tracking-with-geofence-notifications>

Application de suivi des échantillons et de géofence pour Android

Cette rubrique couvre le didacticiel Android conçu pour présenter les principales fonctionnalités de l'utilisation des géofences et des dispositifs de suivi Amazon Location dans une application mobile. Les applications montrent comment un tracker et un geofence interagissent à l'aide d'une combinaison des fonctionnalités Lambda et AWS IoT Amazon Location.

Rubriques

- [Créez des ressources Amazon Location pour votre application](#)
- [Création d'une collection Geofence](#)
- [Associer un traceur à une collection de géofences](#)
- [Utilisation de AWS Lambda avec MQTT](#)
- [Configurer l'exemple de code d'application](#)
- [Utilisation de l'exemple d'application](#)

Créez des ressources Amazon Location pour votre application

Pour commencer, vous devez créer les ressources Amazon Location requises. Ces ressources seront essentielles au fonctionnement de l'application et à l'exécution des extraits de code fournis.

Note

Si vous n'avez pas créé de AWS compte, suivez les instructions du guide de l'utilisateur [de gestion des AWS comptes](#).

Pour commencer, vous devez créer un identifiant de pool d'identités Amazon Cognito, suivez la procédure suivante :

1. Ouvrez la [console Amazon Cognito](#) et sélectionnez Groupes d'identités dans le menu de gauche, puis sélectionnez Créer un pool d'identités.
2. Assurez-vous que l'option Accès invité est cochée, puis appuyez sur Suivant pour continuer.

3. Créez ensuite un nouveau rôle IAM ou utilisez un rôle IAM existant.
4. Entrez le nom du pool d'identités et assurez-vous que le pool d'identités a accès aux (geo) ressources Amazon Location pour la carte et le tracker que vous allez créer lors de la procédure suivante.

Ensuite, vous devez créer et mettre en forme une carte dans la console AWS Amazon Location, en suivant la procédure suivante :

1. Accédez à la [section Cartes](#) de la console Amazon Location et sélectionnez Create Map.
2. Donnez un nom et une description à la nouvelle carte. Enregistrez le nom que vous attribuez, tel qu'il sera utilisé ultérieurement dans le didacticiel.
3. Lorsque vous choisissez un style de carte, pensez au fournisseur de données cartographiques. Reportez-vous à la section 82 des [conditions de AWS service](#) pour plus de détails.
4. Acceptez les [conditions générales d'Amazon Location](#), puis sélectionnez Create Map (Créer une carte) pour terminer le processus de création de la carte.

Ensuite, vous devez créer un outil de suivi dans la console Amazon Location, en suivant la procédure suivante :

1. Ouvrez la [section Cartes](#) dans la console Amazon Location.
2. Choisissez Créer un tracker.
3. Renseignez les champs obligatoires. Notez le nom du traceur tel qu'il sera référencé tout au long de ce didacticiel.
4. Dans le champ Filtrage des positions, choisissez l'option qui correspond le mieux à la manière dont vous avez l'intention d'utiliser votre ressource de suivi. Si vous ne définissez pas le filtrage de position, le paramètre par défaut est `TimeBased`. Pour plus d'informations, consultez [Trackers](#) et [PositionFiltering](#) le Amazon Location API Reference.
5. Choisissez Create tracker pour terminer la création du tracker.

Création d'une collection Geofence

Vous allez maintenant créer une collection de géofences. Vous pouvez utiliser la console, l'API ou la CLI. Les procédures suivantes vous guident à travers chaque option.

- Créez une collection de géofences à l'aide de la console Amazon Location :

1. Ouvrez la section [Geofence Collections](#) de la console Amazon Location.
 2. Choisissez Créer une collection de géofences.
 3. Saisissez un nom et une description pour la collection.
 4. Dans le cadre de la EventBridge règle avec Amazon CloudWatch comme cible, vous pouvez créer une EventBridge règle facultative pour commencer à réagir aux événements de géofence. Cela permet à Amazon Location de publier des événements sur Amazon CloudWatch Logs.
 5. Appuyez sur la touche Créer une collection de géofences pour terminer la création de la collection.
- Créez une collection de géofences à l'aide de l'API Amazon Location :

Utilisez l'[CreateGeofenceCollection](#) opération depuis les API Amazon Location Geofences.

L'exemple suivant utilise une demande d'API pour créer une collection de géofences appelée.

GEOCOLLECTION_NAME

```
POST /geofencing/v0/collections
Content-type: application/json

{
  "CollectionName": "GEOCOLLECTION_NAME",
  "Description": "Geofence collection 1 for shopping center",
  "Tags": {
    "Tag1" : "Value1"
  }
}
```

- Créez une collection de géofences à l'aide des commandes de la AWS CLI :

Utilisez la commande `create-geofence-collection`. L'exemple suivant utilise une AWS CLI pour créer une collection de géofences appelée. *GEOCOLLECTION_NAME* Pour plus d'informations sur l'utilisation de la AWS CLI, consultez la [documentation de l'interface de ligne de AWS commande](#).

```
aws location \
  create-geofence-collection \
  --collection-name "ExampleGeofenceCollection" \
  --description "Shopping center geofence collection" \
  --tags Tag1=Value1
```

Associer un traceur à une collection de géofences

Pour lier un tracker à une collection de géofences, vous pouvez utiliser la console, l'API ou la CLI. Les procédures suivantes vous guident à travers chaque option.

Associez une ressource de suivi à une collection de géofences à l'aide de la console Amazon Location Service :

1. Ouvrez la console Amazon Location.
2. Dans le volet de navigation de gauche, choisissez Trackers.
3. Sous Device Trackers, sélectionnez le lien du nom du tracker cible.
4. Sous Collections Geofence liées, choisissez Link Geofence Collection.
5. Dans la fenêtre Linked Geofence Collection, sélectionnez une collection de géofences dans le menu déroulant.
6. Choisissez Lier.
7. Une fois que vous aurez lié la ressource de suivi, le statut Actif lui sera attribué.

Associez une ressource de suivi à une collection de géofences à l'aide des API Amazon Location :

Utilisez l'AssociateTrackerConsumer opération depuis les API Amazon Location Trackers. L'exemple suivant utilise une demande d'API qui ExampleTracker associe un à une collection de géofences en utilisant son Amazon Resource Name (ARN).

```
POST /tracking/v0/trackers/ExampleTracker/consumers
Content-type: application/json
{
  "ConsumerArn": "arn:aws:geo:us-west-2:123456789012:geofence-
collection/GOECOLLECTION_NAME"
}
```

Liez une ressource de suivi à une collection de géofences à l'aide des commandes AWS CLI :

Utilisez la commande `associate-tracker-consumer` . L'exemple suivant utilise une AWS CLI pour créer une collection de géofences appelée. *GOECOLLECTION_NAME*

```
aws location \  
associate-tracker-consumer \  
  --consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-  
collection/GOECOLLECTION_NAME" \  
  --tracker-name "ExampleTracker"
```

Utilisation de AWS Lambda avec MQTT

Pour créer une connexion entre Amazon Location AWS IoT et Amazon, vous avez besoin d'une fonction Lambda pour traiter les messages transmis par EventBridge CloudWatch des événements. Cette fonction permet d'extraire toutes les données de position, de les formater pour Amazon Location et de les envoyer via l'API Amazon Location Tracker.

La procédure suivante explique comment créer cette fonction via la console Lambda :

1. Ouvrez la [console](#).
2. Dans le menu de navigation de gauche, choisissez Fonctions.
3. Choisissez ensuite Créer une fonction et assurez-vous que l'option Auteur à partir de zéro est sélectionnée.
4. fournissez un nom de fonction, et pour l'option Runtime, choisissez Node.js 16.x.
5. Choisissez Créer une fonction.
6. Ouvrez l'onglet Code pour accéder à l'éditeur.
7. Remplacez le code de l'espace réservé dans le `index.js` fichier par le code suivant :

```
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. Appuyez sur le bouton Déployer pour enregistrer la fonction mise à jour.
9. Ouvrez ensuite l'onglet Configuration.
10. Dans la section Déclencheurs, appuyez sur le bouton Ajouter un déclencheur.
11. Sélectionnez EventBridge (CloudWatch Événements) dans le champ Source.
12. Sélectionnez l'option Règles existantes.

13. Entrez le nom de la règle, par exemple `AmazonLocationMonitor-GEOFENCECOLLECTION_NAME`.
14. Appuyez sur le bouton Ajouter.
15. Cela associera également les déclarations de politique basées sur les ressources dans l'onglet Autorisations

Vous allez maintenant configurer le client de test MQTT en utilisant AWS IoT la procédure suivante :

1. Ouvrez le [site https://console.aws.amazon.com/iot/](https://console.aws.amazon.com/iot/).
2. Dans le volet de navigation de gauche, sélectionnez le client de test MQTT.
3. Vous verrez une section intitulée Client de test MQTT dans laquelle vous pouvez configurer votre connexion MQTT.
4. Après avoir configuré les paramètres nécessaires, cliquez sur le bouton Connect pour établir une connexion avec le broker MQTT à l'aide des paramètres fournis.
5. Enregistrez le point final, tel qu'il sera utilisé ultérieurement dans le didacticiel.

Une fois connecté au client de test, vous pouvez vous abonner à des sujets MQTT ou publier des messages sur des sujets en utilisant les champs de saisie correspondants fournis dans l'interface du client de test MQTT. Vous allez ensuite créer une AWS IoT politique.

6. Dans le menu de gauche, sous Gérer, développez l'option de sécurité et cliquez sur Politiques.
7. Cliquez sur le bouton Créer une politique.
8. Entrez un nom de stratégie.
9. Dans le document de politique, sélectionnez l'onglet JSON.
10. Copiez-collez la politique ci-dessous, mais assurez-vous de mettre à jour tous les éléments avec votre *REGION* et *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. Cliquez sur le bouton Créer pour terminer.

Après avoir terminé la procédure précédente, vous allez maintenant mettre à jour les autorisations pour le rôle d'invité comme suit :

1. Accédez à Amazon Cognito et ouvrez votre pool d'identités. Passez ensuite à l'accès utilisateur et sélectionnez le rôle d'invité.
2. Cliquez sur les politiques d'autorisation pour activer la modification.

```

{
  '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. Avec les modifications de politique ci-dessus, toutes les AWS ressources nécessaires sont désormais configurées de manière appropriée pour l'application.

Configurer l'exemple de code d'application

1. Clonez ce dépôt : <https://github.com/aws-geospatial/amazon-location-samples-android/tree/main/tracking-with-geofence-notifications> sur votre machine locale.
2. Ouvrez le AmazonSampleSDKApp projet dans Android Studio.
3. Créez et exécutez l'application sur votre appareil ou émulateur Android.

Utilisation de l'exemple d'application

Pour utiliser l'exemple, suivez les procédures suivantes :

- Créez un **custom.properties** :

Pour configurer votre custom.properties fichier, procédez comme suit :

1. Ouvrez votre éditeur de texte ou IDE préféré.
2. Créez un nouveau fichier.
3. Enregistrez le fichier sous le nom custom.properties.
4. Mettez à jour le custom.properties avec l'exemple de code suivant et remplacez le MQTT_END_POINT POLICY_NAMEGEOFENCE_COLLECTION_NAME, et par les TOPIC_TRACKER noms de vos ressources :

```
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. Nettoyez et reconstruisez le projet. Ensuite, vous pouvez exécuter le projet.
- Connectez-vous :

Pour vous connecter à l'application, suivez les étapes ci-dessous :

1. Appuyez sur le bouton de connexion.
 2. Fournissez un identifiant de pool d'identités, un nom de tracker et un nom de carte.
 3. Appuyez à nouveau sur Se connecter pour terminer.
- Gérer les filtres :

Ouvrez l'écran de configuration et effectuez les opérations suivantes :

1. Activez ou désactivez les filtres à l'aide de l'interface utilisateur du commutateur.

2. Mettez à jour les filtres de temps et de distance si nécessaire.
- Opérations de suivi :

Ouvrez l'écran de suivi et effectuez les opérations suivantes :

- Vous pouvez démarrer et arrêter le suivi au premier plan, en arrière-plan ou en mode économie de batterie en appuyant sur les boutons correspondants.

Application de suivi d'échantillons et de géofencing pour iOS

Cette rubrique couvre le didacticiel iOS conçu pour présenter les principales fonctionnalités de l'utilisation des géofences et des dispositifs de suivi Amazon Location dans une application mobile. Les applications montrent comment un tracker et un geofence interagissent à l'aide d'une combinaison des fonctionnalités Lambda et AWS IoT Amazon Location.

Rubriques

- [Créez des ressources Amazon Location pour votre application](#)
- [Création d'une collection Geofence](#)
- [Associer un traceur à une collection de géofences](#)
- [Utilisation de AWS Lambda avec MQTT](#)
- [Configuration d'un exemple de code d'application](#)
- [Utilisation de l'exemple d'application](#)

Créez des ressources Amazon Location pour votre application

Pour commencer, vous devez créer les ressources Amazon Location requises. Ces ressources seront essentielles au fonctionnement de l'application et à l'exécution des extraits de code fournis.

Note

Si vous n'avez pas créé de AWS compte, suivez les instructions du guide de l'utilisateur [de gestion des AWS comptes](#).

Pour commencer, vous devez créer un identifiant de pool d'identités Amazon Cognito, suivez la procédure suivante :

1. Ouvrez la [console Amazon Cognito](#) et sélectionnez Groupes d'identités dans le menu de gauche, puis sélectionnez Créer un pool d'identités.
2. Assurez-vous que l'option Accès invité est cochée, puis appuyez sur Suivant pour continuer.
3. Créez ensuite un nouveau rôle IAM ou utilisez un rôle IAM existant.
4. Entrez le nom du pool d'identités et assurez-vous que le pool d'identités a accès aux (geo) ressources Amazon Location pour la carte et le tracker que vous allez créer lors de la procédure suivante.

Ensuite, vous devez créer et mettre en forme une carte dans la console AWS Amazon Location, en suivant la procédure suivante :

1. Accédez à la [section Cartes](#) de la console Amazon Location et sélectionnez Create Map.
2. Donnez un nom et une description à la nouvelle carte. Enregistrez le nom que vous attribuez, tel qu'il sera utilisé ultérieurement dans le didacticiel.
3. Lorsque vous choisissez un style de carte, pensez au fournisseur de données cartographiques. Reportez-vous à la section 82 des [conditions de AWS service](#) pour plus de détails.
4. Acceptez les [conditions générales d'Amazon Location](#), puis sélectionnez Create Map pour terminer le processus de création de la carte.

Ensuite, vous devez créer un tracker dans la console Amazon Location, en suivant la procédure suivante :

1. Ouvrez la [section Cartes](#) dans la console Amazon Location.
2. Choisissez Créer un traceur.
3. Renseignez les champs obligatoires. Notez le nom du tracker tel qu'il sera référencé tout au long de ce didacticiel.
4. Dans le champ Filtrage des positions, choisissez l'option qui correspond le mieux à la manière dont vous avez l'intention d'utiliser votre ressource de suivi. Si vous ne définissez pas le filtrage de position, le paramètre par défaut est TimeBased. Pour plus d'informations, consultez [Start tracking](#) et [PositionFiltering](#) dans le Amazon Location API Reference.
5. Choisissez Create tracker pour terminer la création du tracker.

Création d'une collection Geofence

Vous allez maintenant créer une collection de géofences. Vous pouvez utiliser la console, l'API ou la CLI. Les procédures suivantes vous guident à travers chacune des options.

- Créez une collection de géofences à l'aide de la console Amazon Location :
 1. Ouvrez la section [Geofence Collections](#) de la console Amazon Location.
 2. Choisissez Créer une collection de géofences.
 3. Saisissez un nom et une description pour la collection.
 4. Dans le cadre de la EventBridge règle avec Amazon CloudWatch comme cible, vous pouvez créer une EventBridge règle facultative pour commencer à réagir aux événements de géofence. Cela permet à Amazon Location de publier des événements sur Amazon CloudWatch Logs.
 5. Appuyez sur la touche Créer une collection de géofences pour terminer la création de la collection.

- Créez une collection de géofences à l'aide de l'API Amazon Location :

Utilisez l'[CreateGeofenceCollection](#) opération depuis les API Amazon Location Geofences.

L'exemple suivant utilise une demande d'API pour créer une collection de géofences appelée.

GEOCOLLECTION_NAME

```
POST /geofencing/v0/collections
Content-type: application/json

{
  "CollectionName": "GEOCOLLECTION_NAME",
  "Description": "Geofence collection 1 for shopping center",
  "Tags": {
    "Tag1" : "Value1"
  }
}
```

- Créez une collection de géofences à l'aide des commandes de la AWS CLI :

Utilisez la commande `create-geofence-collection`. L'exemple suivant utilise une AWS CLI pour créer une collection de géofences appelée. *GEOCOLLECTION_NAME* Pour plus d'informations sur l'utilisation de la AWS CLI, consultez la [documentation de l'interface de ligne de AWS commande](#).

```
aws location \  
  create-geofence-collection \  
  --collection-name "ExampleGeofenceCollection" \  
  --description "Shopping center geofence collection" \  
  --tags Tag1=Value1
```

Associer un traceur à une collection de géofences

Pour lier un tracker à une collection de géofences, vous pouvez utiliser la console, l'API ou la CLI. Les procédures suivantes vous guident à travers chacune des options.

Associez une ressource de suivi à une collection de géofences à l'aide de la console Amazon Location Service :

1. Ouvrez la console Amazon Location.
2. Dans le volet de navigation de gauche, choisissez Trackers.
3. Sous Device Trackers, sélectionnez le lien du nom du tracker cible.
4. Sous Collections Geofence liées, choisissez Link Geofence Collection.
5. Dans la fenêtre Linked Geofence Collection, sélectionnez une collection de géofences dans le menu déroulant.
6. Choisissez Lier.
7. Une fois que vous aurez lié la ressource de suivi, le statut Actif lui sera attribué.

Associez une ressource de suivi à une collection de géofences à l'aide des API Amazon Location :

Utilisez l'AssociateTrackerConsumer opération depuis les API Amazon Location Trackers. L'exemple suivant utilise une demande d'API associée à une collection de géofences à l'aide de son Amazon Resource Name (ARN). ExampleTracker

```
POST /tracking/v0/trackers/ExampleTracker/consumers  
Content-type: application/json  
{  
  "ConsumerArn": "arn:aws:geo:us-west-2:123456789012:geofence-  
collection/GEOCOLLECTION_NAME"  
}
```

Associez une ressource de suivi à une collection de géofences à l'aide des commandes de l'AWS CLI :

Utilisez la commande `associate-tracker-consumer` . L'exemple suivant utilise une CLI AWS pour créer une collection de géofences appelée. *GEOCOLLECTION_NAME*

```
aws location \  
  associate-tracker-consumer \  
    --consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-  
collection/GEOCOLLECTION_NAME" \  
    --tracker-name "ExampleTracker"
```

Utilisation de AWS Lambda avec MQTT

Pour créer une connexion entre Amazon Location AWS IoT et Amazon, vous avez besoin d'une fonction Lambda pour traiter les messages transmis par EventBridge CloudWatch des événements. Cette fonction permet d'extraire toutes les données de position, de les formater pour Amazon Location et de les envoyer via l'API Amazon Location Tracker.

La procédure suivante explique comment créer cette fonction via la console Lambda :

1. Ouvrez la [console](#).
2. Dans le menu de navigation de gauche, choisissez Fonctions.
3. Choisissez ensuite Créer une fonction et assurez-vous que l'option Auteur à partir de zéro est sélectionnée.
4. fournissez un nom de fonction, et pour l'option Runtime, choisissez Node.js 16.x.
5. Choisissez Créer une fonction.
6. Ouvrez l'onglet Code pour accéder à l'éditeur.
7. Remplacez le code de l'espace réservé dans le `index.js` fichier par le code suivant :

```
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. Appuyez sur le bouton Déployer pour enregistrer la fonction mise à jour.

9. Ouvrez ensuite l'onglet Configuration.
10. Dans la section Déclencheurs, appuyez sur le bouton Ajouter un déclencheur.
11. Sélectionnez EventBridge (CloudWatch Événements) dans le champ Source.
12. Sélectionnez l'option Règles existantes.
13. Entrez le nom de la règle, par exemple `AmazonLocationMonitor-GEOFENCECOLLECTION_NAME`.
14. Appuyez sur le bouton Ajouter.
15. Cela associera également les déclarations de politique basées sur les ressources dans l'onglet Autorisations

Vous allez maintenant configurer le client de test AWS IoT MQTT, suivez la procédure suivante :

1. Ouvrez le [site https://console.aws.amazon.com/iot/](https://console.aws.amazon.com/iot/).
2. Dans le volet de navigation de gauche, sélectionnez le client de test MQTT.
3. Vous verrez une section intitulée Client de test MQTT dans laquelle vous pouvez configurer votre connexion MQTT.
4. Après avoir configuré les paramètres nécessaires, cliquez sur le bouton Connect pour établir une connexion avec le broker MQTT à l'aide des paramètres fournis.
5. Enregistrez le point final, tel qu'il sera utilisé ultérieurement dans le didacticiel.

Une fois connecté au client de test, vous pouvez vous abonner à des sujets MQTT ou publier des messages sur des sujets en utilisant les champs de saisie correspondants fournis dans l'interface du client de test MQTT. Vous allez ensuite créer une AWS IoT politique.

6. Dans le menu de gauche, sous Gérer, développez l'option de sécurité et cliquez sur Politiques.
7. Cliquez sur le bouton Créer une politique.
8. Entrez un nom de stratégie.
9. Dans le document de politique, sélectionnez l'onglet JSON.
10. Copiez-collez la politique ci-dessous, mais assurez-vous de mettre à jour tous les éléments avec votre *REGION* et *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. Cliquez sur le bouton Créer pour terminer.

Configuration d'un exemple de code d'application

Pour configurer l'exemple de code, les outils suivants doivent être installés :

- Git
- XCode 15.3 ou version ultérieure
- iOS Simulator 16 ou version ultérieure

Utilisez cette procédure pour configurer l'exemple de code d'application :

1. Clonez le dépôt git à partir de cette URL : <https://github.com/aws-geospatial/amazon-location-samples-ios/tree/main/tracking-with-geofence-notifications>.
2. Ouvrez le fichier de projet AWSLocationSampleApp.xcodeproj.
3. Attendez que le processus de résolution du package soit terminé.
4. Dans le menu de navigation du projet, renommez ConfigTemplate.xcconfig Config.xcconfig et renseignez les valeurs suivantes :

```
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`
```

Utilisation de l'exemple d'application

Après avoir configuré l'exemple de code, vous pouvez désormais exécuter l'application sur un simulateur iOS ou un appareil physique.

1. Créez et exécutez l'application.
2. L'application vous demandera des autorisations de localisation et de notification. Tu dois les autoriser.
3. Appuyez sur le bouton de configuration de Cognito.
4. Enregistrez la configuration.
5. Vous pouvez désormais voir les options de filtre pour le temps, la distance et la précision. Utilisez-les selon vos besoins.
6. Accédez à l'onglet Suivi de l'application et vous verrez la carte et le bouton Démarrer le suivi.
7. Si vous avez installé l'application sur un simulateur, vous souhaitez peut-être simuler des changements de position. Cela peut être fait dans Fonctionnalités, sous l'option du menu Emplacement. Par exemple, sélectionnez Fonctionnalités, puis Emplacement, puis Freeway Drive.
8. Appuyez sur le bouton Démarrer le suivi. Vous devriez voir les points de suivi sur la carte.
9. L'application suit également les emplacements en arrière-plan. Ainsi, lorsque vous déplacez l'application en arrière-plan, elle vous demandera l'autorisation de poursuivre le suivi en mode arrière-plan.
10. Vous pouvez arrêter le suivi en appuyant sur le bouton Arrêter le suivi.

Marquage de vos ressources Amazon Location Service

Utilisez le balisage des ressources dans Amazon Location pour créer des balises afin de classer vos ressources par objectif, propriétaire, environnement ou critères. Le balisage de vos ressources vous permet de gérer, d'identifier, d'organiser, de rechercher et de filtrer vos ressources.

Par exemple, avec AWS Resource Groups, vous pouvez créer des groupes de AWS ressources basés sur une ou plusieurs balises ou parties de balises. Vous pouvez également créer des groupes en fonction de leur occurrence dans une pile AWS CloudFormation. À l'aide des Groupes de ressources et de l'Éditeur de balises, vous pouvez regrouper et afficher les données des applications composées de plusieurs services, ressources et régions en un seul endroit. Pour plus d'informations sur [les stratégies de balisage courantes](#), consultez le manuel de référence général AWS.

Chaque balise est une étiquette composée d'une clé et d'une valeur que vous définissez :

- Clé de balise : étiquette générale qui catégorise les valeurs des balises. Par exemple, `CostCenter`.
- Valeur du tag : description facultative de la catégorie clé du tag. Par exemple, `MobileAssetTrackingResourcesProd`.

Cette rubrique vous aide à démarrer avec le balisage en passant en revue les restrictions relatives au balisage. Il vous montre également comment créer des balises et utiliser des balises pour suivre le AWS coût de chaque étiquette active à l'aide de rapports de répartition des coûts.

Rubriques

- [Restrictions de balisage](#)
- [Accorder l'autorisation de baliser les ressources](#)
- [Ajouter un tag à une ressource Amazon Location Service](#)
- [Suivez le coût des ressources par tag](#)
- [Contrôlez l'accès aux ressources Amazon Location Service à l'aide de balises](#)
- [En savoir plus](#)

Restrictions de balisage

Les restrictions de base suivantes s'appliquent aux balises :

- Nombre maximum de balises par ressource : 50
- Pour chaque ressource, chaque clé d'identification doit être unique, et chaque clé d'identification peut avoir une seule valeur.

Note

Si vous ajoutez une nouvelles balise ayant la même clé de balise qu'une balise existante, la nouvelle balise remplace l'ancienne balise.

- Longueur de clé maximale : 128 caractères Unicode en UTF-8
- Longueur de valeur maximale : 256 caractères Unicode en UTF-8
- Les caractères autorisés pour les services sont les lettres, les chiffres et les espaces représentables en UTF-8, ainsi que les caractères suivants : + - = . _ : / @.
- Les clés et valeurs de balise sont sensibles à la casse.
- Le préfixe `aws:` est réservé à l'utilisation d'AWS. Lorsque la balise possède une clé de balise avec ce préfixe, vous ne pouvez pas modifier ou supprimer sa clé ou sa valeur. Les balises avec le `aws:` préfixe ne sont pas prises en compte dans le calcul de vos balises par limite de ressources.

Accorder l'autorisation de baliser les ressources

Vous pouvez utiliser les politiques IAM pour contrôler l'accès à vos ressources Amazon Location et accorder l'autorisation de baliser une ressource lors de sa création. Outre l'autorisation de créer des ressources, la politique peut inclure des `Action` autorisations pour autoriser les opérations de balisage :

- `geo:TagResource`— Permet à un utilisateur d'attribuer un ou plusieurs tags à une ressource Amazon Location spécifiée.
- `geo:UntagResource`— Permet à un utilisateur de supprimer une ou plusieurs balises d'une ressource Amazon Location spécifiée.
- `geo:ListTagsForResource`— Permet à un utilisateur de répertorier toutes les balises attribuées à une ressource Amazon Location.

Voici un exemple de politique permettant à un utilisateur de créer une collection de géofences et de baliser des ressources :

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

```
{
  "Sid": "AllowTaggingForGeofenceCollectionOnCreation",
  "Effect": "Allow",
  "Action": [
    "geo:CreateGeofenceCollection",
    "geo:TagResource"
  ],
  "Resource": "arn:aws:geo:region:accountID:geofence-collection/*"
}
```

Ajouter un tag à une ressource Amazon Location Service

Vous pouvez ajouter des balises lors de la création de vos ressources à l'aide de la console Amazon Location, de l'AWS CLI, des API Amazon Location ou des API Amazon Location :

- [Création d'une ressource cartographique](#)
- [Création d'une ressource d'index des lieux](#)
- [Création d'une ressource de calcul d'itinéraire](#)
- [Création d'une collection de géofences](#)
- [Création d'une ressource de suivi](#)

Pour étiqueter des ressources existantes, modifiez ou supprimez des balises

1. Ouvrez la console Amazon Location à l'[adresse https://console.aws.amazon.com/location/](https://console.aws.amazon.com/location/).
2. Dans le volet de navigation de gauche, choisissez la ressource que vous souhaitez baliser. Par exemple, Maps.
3. Choisissez une ressource dans la liste.
4. Choisissez Gérer les balises pour ajouter, modifier ou supprimer vos balises.

Suivez le coût des ressources par tag

Vous pouvez utiliser des balises pour la répartition des coûts afin de suivre vos AWS coûts en détail. Après avoir activé les balises de répartition des coûts, AWS utilise les balises de répartition des coûts pour organiser la facturation de vos ressources dans votre rapport de répartition des coûts. Cela vous permet de classer et de suivre vos coûts d'utilisation.

Il existe deux types de balises de répartition des coûts que vous pouvez activer :

- [AWS-generated](#) — Ces balises sont générées par AWS. Les balises utilisent le `aws :` préfixe, par exemple, `aws : createdBy`.
- [Défini par l'utilisateur](#) : il s'agit de balises personnalisées que vous créez. Les balises définies par l'utilisateur utilisent le `user :` préfixe, par exemple, `user : CostCenter`

Vous devez activer chaque type de tag individuellement. Une fois vos balises activées, vous pouvez [activer AWS Cost Explorer](#) ou consulter votre rapport mensuel de répartition des coûts.

AWS-generated tags

Pour activer les balises générées par AWS

1. Connectez-vous à la console de Gestion de la facturation et des coûts à l'adresse <https://console.aws.amazon.com/billing/>.
2. Dans le volet de navigation de gauche, choisissez Cost Allocation Tags.
3. Sous l'onglet Balises de répartition des coûts AWS générées, sélectionnez les clés de balise que vous souhaitez activer.
4. Choisissez Activer.

User-defined tags

Pour activer les balises définies par l'utilisateur

1. Connectez-vous à la console de Gestion de la facturation et des coûts à l'adresse <https://console.aws.amazon.com/billing/>.
2. Dans le volet de navigation de gauche, choisissez Cost Allocation Tags.
3. Dans l'onglet Balises de répartition des coûts définies par l'utilisateur, sélectionnez les clés de balise que vous souhaitez activer.
4. Choisissez Activer.

Une fois que vous avez activé vos balises, AWS génère un [rapport mensuel de répartition des coûts](#) pour l'utilisation et le coût de vos ressources. Ce rapport de répartition des coûts inclut tous vos AWS coûts pour chaque période de facturation, y compris les ressources étiquetées et non étiquetées.

Pour plus d'informations, consultez [Utilisation des balises d'allocation des coûts](#) dans le Guide de l'utilisateur AWS Billing and Cost Management.

Contrôlez l'accès aux ressources Amazon Location Service à l'aide de balises

AWS Identity and Access Management Les politiques (IAM) prennent en charge les conditions basées sur les balises, ce qui vous permet de gérer les autorisations pour vos ressources en fonction de balises, de clés et de valeurs spécifiques. Par exemple, une politique de rôle IAM peut inclure des conditions visant à limiter l'accès à des environnements spécifiques, tels que le développement, les tests ou la production, en fonction de balises.

Pour plus d'informations, consultez la rubrique sur le [contrôle de l'accès aux ressources en fonction des balises](#).

En savoir plus

Pour plus d'informations sur :

- Meilleures pratiques en matière de balisage, consultez la section [Balisage des ressources AWS](#) dans le manuel de référence général AWS.
- Utilisation de balises pour contrôler l'accès aux AWS ressources, voir [Contrôle de l'accès aux AWS ressources à l'aide de balises](#) dans le Guide de AWS Identity and Access Management l'utilisateur.

Octroi de l'accès à Amazon Location Service

Pour utiliser Amazon Location Service, un utilisateur doit avoir accès aux ressources et aux API qui composent Amazon Location. Il existe trois stratégies que vous pouvez utiliser pour accorder l'accès à vos ressources.

- Utiliser IAM : pour accorder l'accès aux utilisateurs authentifiés par AWS IAM Identity Center ou AWS Identity and Access Management (IAM), créez une politique IAM qui autorise l'accès aux ressources souhaitées. Pour plus d'informations sur IAM et Amazon Location, consultez [Identity and Access Management pour Amazon Location Service](#).
- Utiliser des clés d'API : pour accorder l'accès à des utilisateurs non authentifiés, vous pouvez créer des clés d'API qui donnent un accès en lecture seule à vos ressources Amazon Location Service. Cela est utile dans le cas où vous ne souhaitez pas authentifier tous les utilisateurs. Par exemple,

une application Web. Pour plus d'informations sur les clés d'API, consultez [Autoriser l'accès invité non authentifié à votre application à l'aide de clés d'API](#).

- Utiliser Amazon Cognito — Une alternative aux clés d'API consiste à utiliser Amazon Cognito pour accorder un accès anonyme. Amazon Cognito vous permet de créer une autorisation plus riche avec une politique afin de définir ce que peuvent faire les utilisateurs non authentifiés. Pour plus d'informations sur l'utilisation d'Amazon Cognito, consultez. [Autoriser l'accès invité non authentifié à votre application à l'aide d'Amazon Cognito](#)

Note

Vous pouvez également utiliser Amazon Cognito pour utiliser votre propre processus d'authentification ou pour combiner plusieurs méthodes d'authentification à l'aide d'Amazon Cognito Federated Identities. Pour plus d'informations, consultez [Getting Started with Federated Identities](#) dans le manuel Amazon Cognito Developer Guide.

Rubriques

- [Autoriser l'accès invité non authentifié à votre application à l'aide de clés d'API](#)
- [Autoriser l'accès invité non authentifié à votre application à l'aide d'Amazon Cognito](#)

Autoriser l'accès invité non authentifié à votre application à l'aide de clés d'API

Lorsque vous appelez les API Amazon Location Service dans vos applications, vous effectuez généralement cet appel en tant qu'utilisateur authentifié autorisé à effectuer les appels d'API. Toutefois, dans certains cas, vous ne souhaitez pas authentifier tous les utilisateurs de votre application. Par exemple, vous souhaitez peut-être qu'une application Web indiquant l'emplacement de votre entreprise soit accessible à tous ceux qui utilisent le site Web, qu'ils soient connectés ou non. Dans ce cas, une alternative consiste à utiliser des clés d'API pour effectuer les appels d'API.

Les clés d'API sont une valeur clé associée à des ressources Amazon Location Service spécifiques dans vos Compte AWS ressources et à des actions spécifiques que vous pouvez effectuer sur ces ressources. Vous pouvez utiliser une clé d'API dans votre application pour effectuer des appels non authentifiés vers les API Amazon Location pour ces ressources. Par exemple, si vous associez une clé d'API à la ressource cartographique MyMap et aux GetMap* actions, une application utilisant cette clé d'API pourra visualiser les cartes créées avec cette ressource, et votre compte sera débité

comme toute autre utilisation de votre compte. Cette même clé d'API ne permet pas de modifier ou de mettre à jour la ressource cartographique. Seule l'utilisation de cette ressource est autorisée.

Note

Les clés d'API ne peuvent être utilisées qu'avec les ressources cartographiques, géographiques et routières, et vous ne pouvez ni modifier ni créer ces ressources. Si votre application a besoin d'accéder à d'autres ressources ou actions pour les utilisateurs non authentifiés, vous pouvez utiliser Amazon Cognito pour fournir un accès avec ou à la place des clés d'API. Pour de plus amples informations, veuillez consulter [Autoriser l'accès invité non authentifié à votre application à l'aide d'Amazon Cognito](#).

Les clés d'API incluent une valeur en texte brut qui donne accès à une ou plusieurs ressources de votre Compte AWS. Si quelqu'un copie votre clé d'API, il peut accéder à ces mêmes ressources. Pour éviter cela, vous pouvez spécifier les domaines dans lesquels la clé d'API peut être utilisée lorsque vous créez la clé. Ces domaines sont appelés référents. Si nécessaire, vous pouvez également créer des clés d'API à court terme en définissant des délais d'expiration pour vos clés d'API.

Rubriques

- [Comparaison des clés d'API avec Amazon Cognito](#)
- [Création de clés d'API](#)
- [Utilisation d'une clé d'API pour appeler une API Amazon Location](#)
- [Utiliser une clé d'API pour afficher une carte](#)
- [Gestion de la durée de vie des clés d'API](#)

Comparaison des clés d'API avec Amazon Cognito

Les clés d'API et Amazon Cognito sont utilisés de la même manière pour des scénarios similaires, alors pourquoi utiliser l'un plutôt que l'autre ? La liste suivante met en évidence certaines des différences entre les deux.

- Les clés d'API ne sont disponibles que pour les ressources cartographiques, géographiques et routières, et uniquement pour certaines actions. Amazon Cognito peut être utilisé pour authentifier l'accès à la plupart des API Amazon Location Service.

- Les performances des requêtes cartographiques avec des clés d'API sont généralement plus rapides que dans des scénarios similaires avec Amazon Cognito. Une authentification simplifiée permet de réduire le nombre d'allers-retours vers le service et de demandes mises en cache lorsque vous obtenez à nouveau la même vignette cartographique en peu de temps.
- Avec Amazon Cognito, vous pouvez utiliser votre propre processus d'authentification ou combiner plusieurs méthodes d'authentification à l'aide des identités fédérées Amazon Cognito. Pour plus d'informations, consultez [Getting Started with Federated Identities](#) dans le manuel Amazon Cognito Developer Guide.

Création de clés d'API

Vous pouvez créer une clé d'API et l'associer à une ou plusieurs ressources de votre Compte AWS.

Vous pouvez créer une clé d'API à l'aide de la console Amazon Location Service AWS CLI, ou des API Amazon Location.

Console

Pour créer une clé d'API à l'aide de la console Amazon Location Service

1. Dans la [console Amazon Location](#), choisissez les clés d'API dans le menu de gauche.
2. Sur la page des clés d'API, choisissez Créer une clé d'API.
3. Sur la page Créer une clé d'API, renseignez les informations suivantes :
 - Nom : nom de votre clé d'API, tel que MyWebAppKey.
 - Description — Description facultative de votre clé d'API.
 - Ressources — Choisissez les ressources Amazon Location auxquelles vous souhaitez donner accès avec cette clé d'API dans la liste déroulante. Vous pouvez ajouter plusieurs ressources en choisissant Ajouter une ressource.
 - Actions — Spécifiez les actions que vous souhaitez autoriser avec cette clé d'API. Vous devez sélectionner au moins une action correspondant à chaque type de ressource sélectionné. Par exemple, si vous avez sélectionné une ressource géographique, vous devez sélectionner au moins l'un des choix sous Actions de localisation.
 - Heure d'expiration : ajoutez éventuellement une date et une heure d'expiration pour votre clé d'API. Pour de plus amples informations, veuillez consulter [Gestion de la durée de vie des clés d'API](#).

- Référents : ajoutez éventuellement un ou plusieurs domaines dans lesquels vous pouvez utiliser la clé API. Par exemple, si la clé d'API doit autoriser l'exécution d'une application sur le site `Webexample.com`, vous pouvez la `*.example.com/` définir comme référent autorisé.
 - Balises : ajoutez éventuellement des balises à la clé d'API.
4. Choisissez `Create API key` pour créer la clé API.
 5. Sur la page détaillée de la clé d'API, vous pouvez consulter les informations relatives à la clé d'API que vous avez créée. Choisissez `Afficher la clé d'API` pour voir la valeur de clé que vous utilisez lorsque vous appelez les API Amazon Location. La valeur clé aura le format `v1.public.a1b2c3d4...`. Pour plus d'informations sur l'utilisation de la clé API pour le rendu de cartes, consultez [Utiliser une clé d'API pour afficher une carte](#).

API

Pour créer une clé d'API à l'aide des API Amazon Location

Utilisez l'[CreateKey](#) opération depuis les API Amazon Location.

L'exemple suivant est une demande d'API visant à créer une clé *ExampleKey* d'API appelée sans date d'expiration et à accéder à une seule ressource cartographique.

```
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 réponse inclut la valeur de la clé d'API à utiliser lors de l'accès aux ressources de vos applications. La valeur clé aura le format `v1.public.a1b2c3d4...`. Pour en savoir plus sur l'utilisation de la clé API pour le rendu de cartes, consultez [Utiliser une clé d'API pour afficher une carte](#).

Vous pouvez également utiliser l'[DescribeKey](#) API pour trouver la valeur d'une clé ultérieurement.

AWS CLI

Pour créer une clé d'API à l'aide de AWS CLI commandes

Utilisez la commande [create-key](#).

L'exemple suivant crée une clé d'API appelée *ExampleKey* sans date d'expiration et donnant accès à une seule ressource cartographique.

```
aws location \  
  create-key \  
    --key-name ExampleKey \  
    --restrictions '{"AllowActions":["geo:GetMap*"],"AllowResources":  
["arn:aws:geo:region:map/mapname"]}' \  
    --no-expiry
```

La réponse inclut la valeur de la clé d'API à utiliser lors de l'accès aux ressources de vos applications. La valeur clé aura le format `v1.public.a1b2c3d4...`. Pour en savoir plus sur l'utilisation de la clé API pour le rendu de cartes, consultez [Utiliser une clé d'API pour afficher une carte](#). La réponse à `create-key` ressemble à ce qui suit.

```
{  
  "Key": "v1.public.a1b2c3d4...",  
  "KeyArn": "arn:aws:geo:region:accountId:api-key/ExampleKey",  
  "KeyName": "ExampleKey",  
  "CreateTime": "2023-02-06T22:33:15.693Z"  
}
```

Vous pouvez également l'`describe-key` utiliser pour rechercher la valeur clé ultérieurement. L'exemple suivant montre comment appeler `describe-key` une clé d'API nommée *ExampleKey*.

```
aws location describe-key \  

```

```
--key-name ExampleKey
```

Utilisation d'une clé d'API pour appeler une API Amazon Location

Après avoir créé une clé d'API, vous pouvez utiliser la valeur de la clé pour appeler les API Amazon Location dans votre application.

Les API qui prennent en charge les clés d'API possèdent un paramètre supplémentaire qui prend la valeur de la clé d'API. Par exemple, si vous appelez l'GetPlaceAPI, vous pouvez renseigner le paramètre [clé](#) comme suit

```
GET /places/v0/indexes/IndexName/places/PlaceId?key=KeyValue
```

Si vous renseignez cette valeur, vous n'avez pas besoin d'authentifier l'appel d'API avec AWS Sig v4 comme vous le feriez normalement.

Pour JavaScript les développeurs, vous pouvez utiliser Amazon Location [JavaScript Aide à l'authentification](#) pour authentifier les opérations d'API à l'aide de clés d'API.

Pour les développeurs mobiles, vous pouvez utiliser les kits SDK d'authentification mobile Amazon Location suivants :

- [SDK d'authentification mobile Amazon Location Service pour iOS](#)
- [SDK d'authentification mobile Amazon Location Service pour Android](#)

Pour AWS CLI les utilisateurs, lorsque vous utilisez le `--key` paramètre, vous devez également l'`--no-sign-request` utiliser, afin d'éviter de signer avec Sig v4.

Note

Si vous incluez à la fois une signature key et une signature AWS Sig v4 dans un appel à Amazon Location Service, seule la clé d'API est utilisée.

Utiliser une clé d'API pour afficher une carte

Vous pouvez utiliser la valeur de la clé d'API pour afficher une carte dans votre application à l'aide de MapLibre. Cela est un peu différent de l'utilisation des clés d'API dans d'autres API Amazon Location que vous appelez directement, car il MapLibre effectue ces appels pour vous.

L'exemple de code suivant montre comment utiliser la clé API pour afficher une carte sur une simple page Web à l'aide du contrôle cartographique MapLibre GL JS. Pour que ce code fonctionne correctement, remplacez le *v1.public.your-api-key-value*, *us-east-1*, et des *ExampleMap* chaînes dont les valeurs correspondent à votre. Compte 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>
```

Gestion de la durée de vie des clés d'API

Vous pouvez créer des clés d'API qui fonctionnent indéfiniment. Toutefois, si vous souhaitez créer une clé d'API temporaire, faire régulièrement pivoter les clés d'API ou révoquer une clé d'API existante, vous pouvez utiliser l'expiration de la clé d'API.

Lorsque vous créez une nouvelle clé d'API ou que vous mettez à jour une clé d'API existante, vous pouvez définir le délai d'expiration de cette clé d'API.

- Lorsqu'une clé d'API atteint son délai d'expiration, elle est automatiquement désactivée. Les clés inactives ne peuvent plus être utilisées pour effectuer des demandes de cartes.
- Vous pouvez supprimer une clé d'API 90 jours après l'avoir désactivée.
- Si vous avez une clé inactive que vous n'avez pas encore supprimée, vous pouvez la restaurer en reportant la date d'expiration à une date future.
- Pour créer une clé permanente, vous pouvez supprimer le délai d'expiration.
- Si vous tentez de désactiver une clé d'API qui a été utilisée au cours des 7 derniers jours, il vous sera demandé de confirmer que vous souhaitez apporter la modification. Si vous utilisez l'API Amazon Location Service, ou le AWS CLI, vous recevrez un message d'erreur, sauf si vous définissez le `ForceUpdate` paramètre sur `true`.

Autoriser l'accès invité non authentifié à votre application à l'aide d'Amazon Cognito

Vous pouvez utiliser l'authentification Amazon Cognito au lieu d'utiliser directement AWS Identity and Access Management (IAM) à la fois avec les SDK frontaux et les requêtes HTTPS directes.

Vous souhaitez peut-être utiliser cette forme d'authentification pour les raisons suivantes :

- Utilisateurs non authentifiés : si vous avez un site Web avec des utilisateurs anonymes, vous pouvez utiliser les groupes d'identités Amazon Cognito. Pour plus d'informations, consultez la section sur [the section called "Utilisation d'Amazon Cognito"](#).
- Votre propre authentification — Si vous souhaitez utiliser votre propre processus d'authentification ou combiner plusieurs méthodes d'authentification, vous pouvez utiliser les identités fédérées Amazon Cognito. Pour plus d'informations, consultez [Getting Started with Federated Identities](#) dans le manuel Amazon Cognito Developer Guide.

Amazon Cognito fournit des fonctionnalités d'authentification, d'autorisation et de gestion des utilisateurs pour les applications Web et mobiles. Vous pouvez utiliser les pools d'identités non authentifiés Amazon Cognito avec Amazon Location pour permettre aux applications de récupérer des informations d'identification temporaires et limitées. AWS

Pour plus d'informations, consultez [Getting Started with User Pools](#) dans le manuel Amazon Cognito Developer Guide.

Note

Pour les développeurs mobiles, Amazon Location fournit des kits SDK d'authentification mobile pour iOS et Android. Consultez les référentiels github suivants pour plus d'informations :

- [SDK d'authentification mobile Amazon Location Service pour iOS](#)
- [SDK d'authentification mobile Amazon Location Service pour Android](#)

Créer un groupe d'identités Amazon Cognito

Vous pouvez créer des groupes d'identités Amazon Cognito pour autoriser un accès invité non authentifié à votre application par le biais de la console Amazon Cognito, des API Amazon Cognito ou AWS CLI des API Amazon Cognito.

Important

Le pool que vous créez doit se trouver dans Compte AWS la même AWS région que les ressources Amazon Location Service que vous utilisez.

Vous pouvez utiliser les politiques IAM associées à des rôles d'identité non authentifiés avec les actions suivantes :

- `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`

L'inclusion d'autres actions Amazon Location n'aura aucun effet et les identités non authentifiées ne pourront pas les appeler.

Exemple

Pour créer un pool d'identités à l'aide de la console Amazon Cognito

1. Accédez à la [console Amazon Cognito](#).
2. Choisissez Manage Identity groupes (Gérer les groupes d'identité).
3. Choisissez Créer un nouveau groupe d'identités, puis entrez le nom de votre groupe d'identités.
4. Dans la section pliable Identités non authentifiées, choisissez Activer l'accès aux identités non authentifiées.
5. Sélectionnez Créer une réserve.
6. Choisissez les rôles IAM que vous souhaitez utiliser avec votre pool d'identités.
7. Agrandir Afficher les détails.
8. Sous Identités non authentifiées, entrez un nom de rôle.
9. Développez la section Afficher le document de politique, puis choisissez Modifier pour ajouter votre politique.
10. Ajoutez votre politique pour autoriser l'accès à vos ressources.

Vous trouverez ci-dessous des exemples de politiques pour les cartes, les lieux, les traceurs et les itinéraires. Pour utiliser les exemples pour votre propre politique, remplacez les espaces réservés à la *région* et à l'*AccountID* :

Maps policy example

La politique suivante accorde un accès en lecture seule à une ressource cartographique nommée. *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"
    }
  ]
}
```

L'ajout d'une [condition IAM](#) correspondante vous `aws:referer` permet de limiter l'accès du navigateur à vos ressources à une liste d'URL ou de préfixes d'URL. L'exemple suivant autorise l'accès à une ressource cartographique nommée uniquement `RasterEsriImagery` à partir du site `Web example.com` :

Warning

Bien qu'il `aws:referer` puisse limiter l'accès, il ne s'agit pas d'un mécanisme de sécurité. Il est dangereux d'inclure une valeur d'en-tête de référent connue publiquement. Les tiers non autorisés peuvent utiliser des navigateurs modifiés ou personnalisés pour fournir n'importe quelle valeur `aws:referer` de leur choix. Par conséquent, `aws:referer` il ne doit pas être utilisé pour empêcher des parties non autorisées de faire des AWS demandes directes. Cette clé est fournie uniquement pour permettre aux clients de protéger leur contenu numérique, stocké notamment dans Amazon S3, contre tout référencement sur des sites tiers non autorisés. [Pour plus d'informations, voir:referer.AWS](#)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "geo:GetMap*",
      "Resource": "arn:aws:geo:us-west-2:111122223333:map/
RasterEsriImagery",
      "Condition": {
        "StringLike": {
          "aws:referer": [
            "https://example.com/*",
            "https://www.example.com/*"
          ]
        }
      }
    }
  ]
}
```

Si vous [utilisez Tangram](#) pour afficher une carte, celui-ci n'utilise pas les descripteurs de style, les glyphes ou les sprites renvoyés par l'API Maps. Il est plutôt configuré en pointant vers un fichier .zip contenant les règles de style et les ressources nécessaires. La politique suivante accorde un accès en lecture seule à une ressource cartographique nommée d'après *ExempleMap* l'opération `GetMapTile`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "MapsReadOnly",
      "Effect": "Allow",
      "Action": [
        "geo:GetMapTile"
      ],
      "Resource": "arn:aws:geo:region:accountID:map/ExempleMap"
    }
  ]
}
```

Places policy example

La politique suivante accorde un accès en lecture seule à une ressource d'index de lieux nommée *ExamplePlaceIndex* pour rechercher des lieux par texte ou par position.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PlacesReadOnly",
      "Effect": "Allow",
      "Action": [
        "geo:SearchPlaceIndex*",
        "geo:GetPlace"
      ],
      "Resource": "arn:aws:geo:region:accountID:place-index/ExamplePlaceIndex"
    }
  ]
}
```

L'ajout d'une [condition IAM](#) correspondante vous `aws:referer` permet de limiter l'accès du navigateur à vos ressources à une liste d'URL ou de préfixes d'URL. L'exemple suivant refuse l'accès à une ressource d'index de lieux nommée à *ExamplePlaceIndex* partir de tous les sites Web de référence, à l'exception de `example.com`.

Warning

Bien qu'il `aws:referer` puisse limiter l'accès, il ne s'agit pas d'un mécanisme de sécurité. Il est dangereux d'inclure une valeur d'en-tête de référent connue publiquement. Les tiers non autorisés peuvent utiliser des navigateurs modifiés ou personnalisés pour fournir n'importe quelle valeur `aws:referer` de leur choix. Par conséquent, `aws:referer` il ne doit pas être utilisé pour empêcher des parties non autorisées de faire des AWS demandes directes. Cette clé est fournie uniquement pour permettre aux clients de protéger leur contenu numérique, stocké notamment dans Amazon S3, contre tout référencement sur des sites tiers non autorisés. [Pour plus d'informations, voir:referer.AWS](#)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "geo:*",
      "Resource": "arn:aws:geo:us-west-2:111122223333:place-
index/ExamplePlaceIndex",
      "Condition": {
        "StringLike": {
          "aws:referrer": [
            "https://example.com/*",
            "https://www.example.com/*"
          ]
        }
      }
    }
  ]
}
```

Trackers policy example

La politique suivante autorise l'accès à une ressource de suivi nommée *ExampleTracker* pour mettre à jour les positions des appareils.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UpdateDevicePosition",
      "Effect": "Allow",
      "Action": [
        "geo:BatchUpdateDevicePosition"
      ],
      "Resource": "arn:aws:geo:region:accountID:tracker/ExampleTracker"
    }
  ]
}
```

L'ajout d'une [condition IAM](#) correspondante vous `aws:referer` permet de limiter l'accès du navigateur à vos ressources à une liste d'URL ou de préfixes d'URL. L'exemple suivant refuse l'accès à une ressource de suivi nommée *ExampleTracker* depuis tous les sites Web de référence, à l'exception de `example.com`.

 Warning

Bien qu'il `aws:referer` puisse limiter l'accès, il ne s'agit pas d'un mécanisme de sécurité. Il est dangereux d'inclure une valeur d'en-tête de référent connue publiquement. Les tiers non autorisés peuvent utiliser des navigateurs modifiés ou personnalisés pour fournir n'importe quelle valeur `aws:referer` de leur choix. Par conséquent, `aws:referer` il ne doit pas être utilisé pour empêcher des parties non autorisées de faire des AWS demandes directes. Cette clé est fournie uniquement pour permettre aux clients de protéger leur contenu numérique, stocké notamment dans Amazon S3, contre tout référencement sur des sites tiers non autorisés. [Pour plus d'informations, voir:referer.AWS](#)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "geo:GetDevice*",
      "Resource": "arn:aws:geo:us-west-2:111122223333:tracker/ExampleTracker",
      "Condition": {
        "StringLike": {
          "aws:referer": [
            "https://example.com/*",
            "https://www.example.com/*"
          ]
        }
      }
    }
  ]
}
```

Routes policy example

La politique suivante autorise l'accès à une ressource de calcul d'itinéraire nommée *ExampleCalculator* pour calculer un itinéraire.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RoutesReadOnly",
      "Effect": "Allow",
      "Action": [
        "geo:CalculateRoute"
      ],
      "Resource": "arn:aws:geo:region:accountID:route-
calculator/ExampleCalculator"
    }
  ]
}
```

L'ajout d'une [condition IAM](#) correspondante vous `aws:referer` permet de limiter l'accès du navigateur à vos ressources à une liste d'URL ou de préfixes d'URL. L'exemple suivant refuse l'accès à un calculateur d'itinéraire nommé *ExampleCalculator* depuis tous les sites Web de référence, à l'exception de `example.com`.

Warning

Bien qu'il `aws:referer` puisse limiter l'accès, il ne s'agit pas d'un mécanisme de sécurité. Il est dangereux d'inclure une valeur d'en-tête de référent connue publiquement. Les tiers non autorisés peuvent utiliser des navigateurs modifiés ou personnalisés pour fournir n'importe quelle valeur `aws:referer` de leur choix. Par conséquent, `aws:referer` il ne doit pas être utilisé pour empêcher des parties non autorisées de faire des AWS demandes directes. Cette clé est fournie uniquement pour permettre aux clients de protéger leur contenu numérique, stocké notamment dans Amazon S3, contre tout référencement sur des sites tiers non autorisés. [Pour plus d'informations, voir:referer.AWS](#)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "geo:*",
      "Resource": "arn:aws:geo:us-west-2:111122223333:route-
calculator/ExampleCalculator",
      "Condition": {
        "StringLike": {
          "aws:referrer": [
            "https://example.com/*",
            "https://www.example.com/*"
          ]
        }
      }
    }
  ]
}
```

Note

Bien que les pools d'identités non authentifiés soient destinés à être exposés sur des sites Internet non sécurisés, notez qu'ils seront échangés contre des informations d'identification standard limitées dans le temps. AWS

Il est important de définir de manière appropriée les rôles IAM associés aux pools d'identités non authentifiés.

11. Choisissez Autoriser pour créer vos groupes d'identités.

Le pool d'identités qui en résulte suit la syntaxe `<region>:<GUID>`.

Par exemple :

```
us-east-1:1sample4-5678-90ef-aaaa-1234abcd56ef
```

Pour plus d'exemples de politiques spécifiques à Amazon Location, consultez [the section called "Exemples de politiques basées sur l'identité"](#).

Utilisation des groupes d'identités Amazon Cognito dans JavaScript

L'exemple suivant échange le pool d'identités non authentifié que vous avez créé contre des informations d'identification qui sont ensuite utilisées pour récupérer le descripteur de style de votre ressource cartographique. *ExampleMap*

```
const AWS = require("aws-sdk");

const credentials = new AWS.CognitoIdentityCredentials({
  IdentityPoolId: "<identity pool ID>" // for example, us-east-1:sample4-5678-90ef-
  aaaa-1234abcd56ef
});

const client = new AWS.Location({
  credentials,
  region: AWS.config.region || "<region>"
});

console.log(await client.getMapStyleDescriptor("ExampleMap").promise());
```

Note

Les informations d'identification récupérées à partir d'identités non authentifiées sont valides pendant une heure.

Voici un exemple de fonction qui renouvelle automatiquement les informations d'identification avant leur expiration.

```
async function refreshCredentials() {
  await credentials.refreshPromise();
  // schedule the next credential refresh when they're about to expire
  setTimeout(refreshCredentials, credentials.expireTime - new Date());
}
```

Pour simplifier ce travail, vous pouvez utiliser l'Amazon Location [JavaScript Aide à l'authentification](#). Cela permet à la fois d'obtenir les informations d'identification et de les actualiser. Cet exemple utilise le AWS SDK pour la JavaScript version 3.

```
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));
```

Étapes suivantes

- Pour modifier vos rôles, accédez à la [console IAM](#).
- Pour gérer vos groupes d'identités, accédez à la console [Amazon Cognito](#).

Surveillance d'Amazon Location Service

Lorsque vous utilisez Amazon Location Service, vous pouvez surveiller votre utilisation et vos ressources au fil du temps en utilisant :

- Amazon CloudWatch. Surveille les ressources de votre Amazon Location Service et fournit des métriques ainsi que des statistiques en temps quasi réel.
- AWS CloudTrail. Assure le suivi des événements de tous les appels vers les API Amazon Location Service.

Cette section fournit des informations sur l'utilisation de ces services.

Rubriques

- [Surveillance d'Amazon Location Service avec Amazon CloudWatch](#)
- [Enregistrement et surveillance avec AWS CloudTrail](#)

Surveillance d'Amazon Location Service avec Amazon CloudWatch

Amazon CloudWatch surveille vos AWS ressources et les applications que vous exécutez AWS en temps quasi réel. Vous pouvez surveiller les ressources Amazon Location à l'aide CloudWatch de ce service qui collecte des données brutes et transforme les métriques en statistiques significatives en temps quasi réel. Vous pouvez consulter des informations historiques sur une période allant jusqu'à 15 mois ou des statistiques de recherche à consulter dans la CloudWatch console Amazon pour avoir une meilleure idée de vos ressources Amazon Location. Vous pouvez également définir des alarmes en définissant des seuils, envoyer des notifications ou prendre des mesures lorsque ces seuils sont atteints.

Pour plus d'informations, consultez le [guide de CloudWatch l'utilisateur Amazon](#)

Rubriques

- [Métriques Amazon Location Service exportées vers Amazon CloudWatch](#)
- [Afficher les statistiques d'Amazon Location Service](#)
- [Créez des CloudWatch alarmes pour les métriques d'Amazon Location Service](#)
- [Utilisation CloudWatch pour surveiller l'utilisation par rapport aux quotas](#)
- [CloudWatch exemples de mesures pour Amazon Location Service](#)

Métriques Amazon Location Service exportées vers Amazon CloudWatch

Les métriques sont des points de données ordonnés dans le temps qui sont exportés vers CloudWatch. Une dimension est une paire nom/valeur qui identifie la métrique. Pour plus d'informations, consultez la section [Utilisation CloudWatch des métriques et des CloudWatch dimensions](#) dans le guide de CloudWatch l'utilisateur Amazon.

Voici les métriques vers lesquelles Amazon Location Service exporte CloudWatch dans l'espace de `AWS/Location` noms.

Métrique	Description
CallCount	<p>Le nombre d'appels effectués vers un point de terminaison d'API donné.</p> <p>Dimensions valides : noms des API Amazon Location Service</p> <p>Statistique valide : somme</p> <p>Unités : nombre</p>
ErrorCount	<p>Le nombre de réponses d'erreur suite à des appels effectués vers un point de terminaison d'API donné.</p> <p>Dimensions valides : noms des API Amazon Location Service</p> <p>Statistique valide : somme</p> <p>Unités : nombre</p>
SuccessCount	<p>Le nombre d'appels réussis effectués vers un point de terminaison d'API donné.</p> <p>Dimensions valides : noms des API Amazon Location Service</p> <p>Statistique valide : somme</p> <p>Unités : nombre</p>
CallLatency	<p>Le temps nécessaire à l'opération pour traiter et renvoyer une réponse lorsqu'un appel est effectué vers un point de terminaison d'API donné.</p> <p>Dimensions valides : noms des API Amazon Location Service</p> <p>Statistique valide : moyenne</p> <p>Unités : millisecondes</p>

Afficher les statistiques d'Amazon Location Service

Vous pouvez consulter les statistiques d'Amazon Location Service sur la Amazon CloudWatch console ou à l'aide de l'Amazon CloudWatchAPI.

Pour afficher les métriques à l'aide de la CloudWatch console

Exemple

1. Ouvrez la CloudWatch console à l'[adresse https://console.aws.amazon.com/cloudwatch/](https://console.aws.amazon.com/cloudwatch/).
2. Dans le panneau de navigation, sélectionnez Métriques.
3. Dans l'onglet Toutes les métriques, choisissez l'espace de noms Amazon Location.
4. Sélectionnez le type de métrique à afficher.
5. Sélectionnez la métrique et ajoutez-la au graphique.

Pour plus d'informations, consultez la section [Afficher les métriques disponibles](#) dans le guide de CloudWatch l'utilisateur Amazon.

Créez des CloudWatch alarmes pour les métriques d'Amazon Location Service

Vous pouvez l'utiliser CloudWatch pour définir des alarmes sur vos statistiques Amazon Location Service. Par exemple, vous pouvez créer une alarme CloudWatch pour envoyer un e-mail chaque fois qu'un pic de nombre d'erreurs survient.

Les rubriques suivantes vous offrent une présentation générale de la configuration des alarmes à l'aide de CloudWatch. Pour obtenir des instructions détaillées, consultez la section [Utilisation des alarmes](#) dans le guide de CloudWatch l'utilisateur Amazon.

Pour définir des alarmes à l'aide de la CloudWatch console

Exemple

1. Ouvrez la CloudWatch console à l'[adresse https://console.aws.amazon.com/cloudwatch/](https://console.aws.amazon.com/cloudwatch/).
2. Dans le volet de navigation, choisissez Alarm.
3. Sélectionnez Create Alarm (Créer une alerte).
4. Choisissez Sélectionner une métrique.
5. Dans l'onglet Toutes les métriques, sélectionnez l'espace de noms Amazon Location.
6. Sélectionnez une catégorie de mesures.
7. Recherchez la ligne contenant la métrique pour laquelle vous souhaitez créer une alarme, puis cochez la case à côté de cette ligne.

8. Choisissez Select metric (Sélectionner une métrique).
9. Sous Métrique, renseignez les valeurs.
10. Spécifiez les conditions de l'alarme.
11. Choisissez Suivant.
12. Si vous souhaitez envoyer une notification lorsque les conditions d'alarme sont remplies :
 - Sous Déclencheur d'état d'alarme, sélectionnez l'état d'alarme pour demander l'envoi d'une notification.
 - Sous Sélectionnez une rubrique SNS, choisissez Create new topic pour créer une nouvelle rubrique Amazon Simple Notification Service (Amazon SNS). Entrez le nom du sujet et l'adresse e-mail à laquelle envoyer la notification.
 - Sous Envoyer une notification, entrez des adresses e-mail supplémentaires auxquelles envoyer la notification.
 - Sélectionnez Ajouter une notification. La liste est enregistrée et s'affiche dans le champ des alarmes futures.
13. Choisissez Next une fois que vous avez terminé.
14. Entrez le nom et la description de l'alarme, puis choisissez Next.
15. Confirmez les détails de l'alarme, puis choisissez Next.

Note

Lorsque vous créez une nouvelle rubrique Amazon SNS, vous devez vérifier l'adresse e-mail avant de pouvoir envoyer une notification. Si l'e-mail n'est pas vérifié, la notification ne sera pas reçue lorsqu'une alarme est déclenchée par un changement d'état.

Pour plus d'informations sur la façon de configurer des alarmes à l'aide de la CloudWatch console, consultez la section [Créer une alarme qui envoie un e-mail](#) dans le guide de CloudWatch l'utilisateur Amazon.

Utilisation CloudWatch pour surveiller l'utilisation par rapport aux quotas

Vous pouvez créer des CloudWatch alarmes Amazon pour vous avertir lorsque l'utilisation d'un quota donné dépasse un seuil configurable. Cela vous permet de savoir quand vous êtes proche de vos limites de quota, et soit d'adapter votre utilisation pour éviter les dépassements de coûts, soit

de demander une augmentation de quota, si nécessaire. Pour plus d'informations sur la manière de CloudWatch surveiller les quotas, consultez [Visualiser vos quotas de service et configurer des alarmes](#) dans le guide de l' CloudWatch utilisateur Amazon.

CloudWatch exemples de mesures pour Amazon Location Service

Vous pouvez utiliser l'[GetMetricData](#) API pour récupérer des statistiques pour Amazon Location.

- Par exemple, vous pouvez surveiller `CallCount` et définir une alarme en cas de baisse du nombre de personnes.

Le suivi `CallCount` des indicateurs `SendDeviceLocation` peut vous aider à avoir une idée des actifs suivis. Si le `CallCount` taux baisse, cela signifie que les actifs suivis, tels qu'une flotte de camions, ont cessé d'indiquer leur position actuelle. Le réglage d'une alarme à cet effet peut vous aider à vous avertir qu'un problème s'est produit.

- Par exemple, vous pouvez surveiller `ErrorCount` et définir une alarme en cas de pic de nombre.

Les traceurs doivent être associés aux collections de géofences afin que l'emplacement des appareils puisse être évalué par rapport aux géofences. Si vous avez un parc d'appareils qui nécessite des mises à jour de localisation continues, le fait de voir le chiffre « `CallCount` pour » `BatchEvaluateGeofence` ou « `BatchPutDevicePosition` tomber à zéro » indique que les mises à jour ne sont plus diffusées.

Voici un exemple de sortie pour [GetMetricData](#) les métriques pour `CallCount` et `ErrorCount` pour la création de ressources cartographiques.

```
{
  "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"
  }
}
]
}
}

```

Enregistrement et surveillance avec AWS CloudTrail

AWS CloudTrail est un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service. CloudTrail enregistre tous les appels d'API en tant qu'événements. Vous pouvez utiliser Amazon Location Service CloudTrail pour surveiller vos appels d'API, notamment les appels depuis la console Amazon Location Service et les appels du AWS SDK vers les opérations de l'API Amazon Location Service.

Lorsque vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrail événements vers un compartiment S3, y compris des événements pour Amazon Location Service. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande envoyée à Amazon Location Service, l'adresse IP

à partir de laquelle la demande a été faite, l'auteur de la demande, la date à laquelle elle a été faite, ainsi que des informations supplémentaires.

Pour plus d'informations CloudTrail, consultez le [guide de AWS CloudTrail l'utilisateur](#).

Rubriques

- [Informations sur le service de location Amazon dans CloudTrail](#)
- [Comprendre les entrées du fichier journal Amazon Location Service](#)

Informations sur le service de location Amazon dans CloudTrail

CloudTrail est activé sur votre AWS compte lorsque vous le créez. Lorsqu'une activité a lieu dans Amazon Location Service, cette activité est enregistrée dans un CloudTrail événement avec les autres événements du AWS service dans l'historique des événements. Vous pouvez afficher, rechercher et télécharger les événements récents dans votre compte AWS. Pour plus d'informations, consultez la section [Affichage des événements à l'aide de l'historique des CloudTrail événements](#).

Pour un enregistrement continu des événements de votre AWS compte, y compris des événements pour Amazon Location Service, créez un historique. Un journal permet CloudTrail de fournir des fichiers journaux à un compartiment S3. Par défaut, lorsque vous créez un journal d'activité dans la console, il s'applique à toutes les régions AWS. Le journal d'activité consigne les événements de toutes les régions dans la partition AWS et livre les fichiers journaux dans le compartiment S3 de votre choix. En outre, vous pouvez configurer d'autres AWS services pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence.

Pour en savoir plus, consultez les ressources suivantes :

- [Présentation de la création d'un journal d'activité](#)
- [CloudTrail Services et intégrations pris en charge](#)
- [Configuration des notifications Amazon SNS pour CloudTrail](#)
- [Réception de fichiers CloudTrail journaux de plusieurs régions](#) et [réception de fichiers CloudTrail journaux de plusieurs comptes](#)

Toutes les actions d'Amazon Location Service sont enregistrées CloudTrail et documentées dans les [références de l'API Amazon Location Service](#). Par exemple, les appels au `CreateTracker` `UpdateTracker` et les `DescribeTracker` actions génèrent des entrées dans les fichiers CloudTrail journaux.

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations d'identité vous aident à déterminer si la demande a été effectuée :

- Avec les informations d'identification de l'utilisateur root ou AWS Identity and Access Management (IAM).
- avec des informations d'identification de sécurité temporaires correspondant à un rôle ou un utilisateur fédéré ;
- par un autre service AWS.

Pour plus d'informations, consultez la section [Élément userIdentity CloudTrail](#) .

Comprendre les entrées du fichier journal Amazon Location Service

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment S3 que vous spécifiez ou à Amazon CloudWatch Logs. Pour plus d'informations, consultez la section [Utilisation des fichiers CloudTrail journaux](#) dans le Guide de AWS CloudTrail l'utilisateur.

CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Une entrée de journal représente une demande individuelle à partir d'une source quelconque et comprend des informations sur l'opération demandée, y compris la date et l'heure de l'opération, les paramètres de la demande, etc.

Note

CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des appels d'API publics, ils n'apparaissent donc pas dans un ordre spécifique. Pour déterminer l'ordre des opérations, utilisez [eventTime](#).

L'exemple suivant montre une entrée de CloudTrail journal illustrant l'CreateTracker opération qui crée une ressource de suivi.

```
{
  "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",
}

```

Ce qui suit montre une entrée du journal de l'DescribeTracker opération, qui renvoie les détails d'une ressource de suivi.

```
{
  "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",
```

```
}
```

Création de ressources Amazon Location Service avec AWS CloudFormation

Amazon Location Service est intégré à AWS CloudFormation un service qui vous aide à modéliser et à configurer vos AWS ressources afin que vous puissiez passer moins de temps à créer et à gérer vos ressources et votre infrastructure. Vous créez un modèle qui décrit toutes les AWS ressources que vous souhaitez (telles que les ressources Amazon Location), puis vous AWS CloudFormation approvisionnez et configurez ces ressources pour vous.

Lorsque vous l'utilisez AWS CloudFormation, vous pouvez réutiliser votre modèle pour configurer vos ressources Amazon Location de manière cohérente et répétée. Décrivez vos ressources une seule fois, puis mettez-le en service autant de fois que vous le souhaitez dans plusieurs comptes et régions AWS.

Amazon Location et AWS CloudFormation modèles

Pour fournir et configurer des ressources pour Amazon Location et les services associés, vous devez comprendre les [AWS CloudFormation modèles](#). Les modèles sont des fichiers texte formatés en JSON ou YAML. Ces modèles décrivent les ressources que vous souhaitez allouer dans vos piles AWS CloudFormation. Si JSON ou YAML ne vous est pas familier, vous pouvez utiliser AWS CloudFormation Designer pour vous aider à démarrer avec des modèles AWS CloudFormation. Pour plus d'informations, consultez [Qu'est-ce que AWS CloudFormation Designer ?](#) dans le AWS CloudFormationGuide de l'utilisateur.

Amazon Location prend en charge la création des types de ressources suivants dans AWS CloudFormation :

- [AWS::Location::Map](#)
- [AWS::Location::PlaceIndex](#)
- [AWS::Location::RouteCalculator](#)
- [AWS::Location::Tracker](#)
- [AWS::Location::TrackerConsumer](#)
- [AWS::Location::GeofenceCollection](#)

Pour plus d'informations, notamment des exemples de modèles JSON et YAML pour les ressources Amazon Location Service, consultez la [référence du type de ressource Amazon Location Service](#) dans le guide de l'AWS CloudFormation utilisateur.

En savoir plus sur AWS CloudFormation

Pour en savoir plus sur AWS CloudFormation, consultez les ressources suivantes :

- [AWS CloudFormation](#)
- [Guide de l'utilisateur AWS CloudFormation](#)
- [Référence d'API AWS CloudFormation](#)
- [Guide de l'utilisateur de l'interface de ligne de commande AWS CloudFormation](#)

Sécurité dans Amazon Location Service

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez de centres de données et d'architectures réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le [modèle de responsabilité partagée](#) décrit cela comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS Cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de [AWS conformité Programmes](#) de de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à Amazon Location Service, consultez la section [AWS Services concernés par le programme de conformitéAWS](#) .
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris de la sensibilité de vos données, des exigences de votre entreprise, ainsi que de la législation et de la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation d'Amazon Location. Les rubriques suivantes expliquent comment configurer Amazon Location pour répondre à vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser vos ressources Amazon Location.

Rubriques

- [Protection des données dans Amazon Location Service](#)
- [Identity and Access Management pour Amazon Location Service](#)
- [Réponse aux incidents dans Amazon Location Service](#)
- [Validation de conformité pour Amazon Location Service](#)
- [Résilience dans Amazon Location Service](#)
- [Sécurité de l'infrastructure dans Amazon Location Service](#)
- [Analyse de configuration et de vulnérabilité dans Amazon Location](#)
- [Prévention du problème de l'adjoint confus entre services](#)
- [Bonnes pratiques de sécurité pour Amazon Location Service](#)

- [Bonnes pratiques pour Amazon Location Service](#)

Protection des données dans Amazon Location Service

Le [modèle de responsabilité AWS partagée](#) de s'applique à la protection des données dans Amazon Location Service. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez [Questions fréquentes \(FAQ\) sur la confidentialité des données](#). Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog [Modèle de responsabilité partagée AWS et RGPD \(Règlement général sur la protection des données\)](#) sur le Blog de sécuritéAWS .

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- Utilisez le protocole SSL/TLS pour communiquer avec les ressources. AWS Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-2 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS (Federal Information Processing Standard) disponibles, consultez [Federal Information Processing Standard \(FIPS\) 140-2](#) (Normes de traitement de l'information fédérale).

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels

que le champ Name (Nom). Cela inclut lorsque vous travaillez avec Amazon Location ou une autre entreprise Services AWS à l'aide de la console, de l'API ou AWS des SDK. AWS CLI Toutes les données que vous entrez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

Confidentialité des données

Avec Amazon Location Service, vous gardez le contrôle des données de votre entreprise. Amazon Location anonymise toutes les requêtes envoyées aux fournisseurs de données en supprimant les métadonnées des clients et les informations de compte.

Amazon Location n'utilise pas de fournisseurs de données pour le suivi et le géofencing. Cela signifie que vos données sensibles restent dans votre AWS compte. Cela permet de protéger les informations de localisation sensibles, telles que l'emplacement des installations, des actifs et du personnel, contre les tiers, de protéger la confidentialité des utilisateurs et de réduire les risques de sécurité de votre application.

Pour plus d'informations, consultez la [FAQ sur la confidentialitéAWS des données](#).

Conservation des données sur Amazon Location

Les caractéristiques suivantes concernent la manière dont Amazon Location collecte et stocke les données pour le service :

- Traceurs Amazon Location Service — Lorsque vous utilisez les API Trackers pour suivre la position des entités, leurs coordonnées peuvent être stockées. Les localisations des appareils sont conservées pendant 30 jours avant d'être supprimées par le service.
- Géofences d'Amazon Location Service : lorsque vous utilisez les API Geofences pour définir des zones d'intérêt, le service enregistre les géométries que vous avez fournies. Ils doivent être explicitement supprimés.

Note

La suppression de votre AWS compte entraîne la suppression de toutes les ressources qu'il contient. Pour plus d'informations, consultez la [FAQ sur la confidentialité AWS des données](#).

Chiffrement des données au repos pour Amazon Location Service

Amazon Location Service fournit un chiffrement par défaut pour protéger les données sensibles des clients au repos à l'aide de clés de chiffrement AWS détenues par nos soins.

- **AWS clés détenues** : Amazon Location utilise ces clés par défaut pour chiffrer automatiquement les données personnelles identifiables. Vous ne pouvez pas afficher, gérer ou utiliser les clés que vous AWS possédez, ni auditer leur utilisation. Toutefois, vous n'avez pas besoin de prendre de mesure ou de modifier les programmes pour protéger les clés qui chiffrent vos données. Pour plus d'informations, consultez [Clés détenues par AWS](#) dans le Guide du développeur AWS Key Management Service .

Le chiffrement des données au repos par défaut permet de réduire les frais opérationnels et la complexité liés à la protection des données sensibles. Dans le même temps, il vous permet de créer des applications sécurisées qui répondent aux exigences réglementaires et de conformité strictes en matière de chiffrement.

Bien que vous ne puissiez pas désactiver cette couche de chiffrement ou sélectionner un autre type de chiffrement, vous pouvez ajouter une deuxième couche de chiffrement aux clés de chiffrement AWS détenues existantes en choisissant une clé gérée par le client lorsque vous créez vos ressources de suivi et de collecte de géofences :

- **Clés gérées par le client** : Amazon Location prend en charge l'utilisation d'une clé symétrique gérée par le client que vous créez, détenez et gérez afin d'ajouter une deuxième couche de chiffrement par rapport au chiffrement AWS détenu existant. Étant donné que vous avez le contrôle total de cette couche de chiffrement, vous pouvez effectuer les tâches suivantes :
 - Établissement et gestion des stratégies de clé
 - Établissement et gestion des politiques IAM et des octrois
 - Activation et désactivation des stratégies de clé

- Rotation des matériaux de chiffrement de clé
- Ajout de balises
- Création d'alias de clé
- Planification des clés pour la suppression

Pour plus d'informations, consultez la section [clé gérée par le client](#) dans le guide du AWS Key Management Service développeur.

Le tableau suivant récapitule la manière dont Amazon Location chiffre les données personnelles identifiables.

Type de données	AWS chiffrement par clé possédée	Chiffrement par clé gérée par le client (facultatif)
Position Une géométrie ponctuelle contenant les détails de position de l'appareil .	Activées	Activées
PositionProperties Ensemble de paires clé-valeur associées à la mise à jour de position .	Activées	Activées
GeofenceGeometry Géométrie de géofence polygonale représentant la zone géofencée.	Activées	Activées
DeviceId L'identifiant de l'appareil spécifié lors du téléchargement d'une mise à jour de la	Activées	Non pris en charge

Type de données	AWS chiffrement par clé possédée	Chiffrement par clé gérée par le client (facultatif)
position de l'appareil sur une ressource de suivi.		
GeofenceId	Activées	Non pris en charge
Identifiant spécifié lors du stockage d'une géométrie de géofences ou d'un lot de géofences dans une collection de géofences donnée.		

Note

Amazon Location active automatiquement le chiffrement au repos à l'aide de clés AWS détenues afin de protéger gratuitement les données personnelles identifiables.

Toutefois, AWS KMS des frais s'appliquent pour l'utilisation d'une clé gérée par le client. Pour plus d'informations sur les tarifs, consultez les [AWS Key Management Service tarifs](#).

Pour plus d'informations AWS KMS, voir [Qu'est-ce que c'est AWS Key Management Service ?](#)

Comment Amazon Location Service utilise les subventions dans AWS KMS

Amazon Location nécessite une [autorisation](#) pour utiliser votre clé gérée par le client.

Lorsque vous créez une [ressource de suivi](#) ou une [collection de géofences](#) chiffrée à l'aide d'une clé gérée par le client, Amazon Location crée une subvention en votre nom en envoyant une [CreateGrant](#) demande à AWS KMS. Les subventions AWS KMS sont utilisées pour donner à Amazon Location l'accès à une clé KMS dans un compte client.

Amazon Location a besoin de l'autorisation d'utiliser votre clé gérée par le client pour les opérations internes suivantes :

- Envoyez [DescribeKey](#) des demandes AWS KMS à pour vérifier que l'ID de clé KMS symétrique géré par le client saisi lors de la création d'un tracker ou d'une collection de géofences est valide.

- Envoyez [GenerateDataKeyWithoutPlaintext](#) des demandes AWS KMS à pour générer des clés de données chiffrées par votre clé gérée par le client.
- Envoyez des demandes de [déchiffrement](#) AWS KMS à pour déchiffrer les clés de données chiffrées afin qu'elles puissent être utilisées pour chiffrer vos données.

Vous pouvez révoquer l'accès à l'octroi ou supprimer l'accès du service à la clé gérée par le client à tout moment. Dans ce cas, Amazon Location ne pourra accéder à aucune des données chiffrées par la clé gérée par le client, ce qui affectera les opérations qui dépendent de ces données. Par exemple, si vous tentez d'[obtenir les positions des appareils](#) à partir d'un outil de suivi crypté auquel Amazon Location ne peut pas accéder, l'opération renverra une `AccessDeniedException` erreur.

Création d'une clé gérée par le client

Vous pouvez créer une clé symétrique gérée par le client à l'aide de AWS Management Console, ou des AWS KMS API.

Pour créer une clé symétrique gérée par le client

Suivez les étapes de la rubrique [Création d'une clé symétrique gérée par le client](#) dans le Guide du développeur AWS Key Management Service .

Stratégie de clé

Les politiques de clés contrôlent l'accès à votre clé gérée par le client. Chaque clé gérée par le client doit avoir exactement une stratégie de clé, qui contient des instructions qui déterminent les personnes pouvant utiliser la clé et comment elles peuvent l'utiliser. Lorsque vous créez votre clé gérée par le client, vous pouvez spécifier une stratégie de clé. Pour plus d'informations, consultez [Gestion de l'accès aux clés gérées par le client](#) dans le Guide du développeur AWS Key Management Service .

Pour utiliser votre clé gérée par le client avec vos ressources Amazon Location, les opérations d'API suivantes doivent être autorisées dans la politique relative aux clés :

- [kms:CreateGrant](#) : ajoute une attribution à une clé gérée par le client. Accorde un accès de contrôle à une clé KMS spécifiée, ce qui permet d'accéder aux [opérations d'octroi](#) requises par Amazon Location. Pour plus d'informations sur [l'utilisation des subventions](#), consultez le guide du AWS Key Management Service développeur.

Cela permet à Amazon Location d'effectuer les opérations suivantes :

- Appelez `GenerateDataKeyWithoutPlainText` pour générer une clé de données chiffrée et la stocker, car la clé de données n'est pas immédiatement utilisée pour chiffrer.
- Appelez `Decrypt` pour utiliser la clé de données chiffrée stockée afin d'accéder aux données chiffrées.
- Configurez un directeur partant à la retraite pour permettre au service de `RetireGrant`.
- [kms:DescribeKey](#)— Fournit les informations clés gérées par le client pour permettre à Amazon Location de valider la clé.

Voici des exemples de déclarations de politique que vous pouvez ajouter pour 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" : "*"
}
]
```

Pour plus d'informations sur la [spécification d'autorisations dans une politique](#), consultez le Guide du développeur AWS Key Management Service .

Pour plus d'informations sur le [dépannage des clés d'accès](#), consultez le Guide du développeur AWS Key Management Service .

Spécifier une clé gérée par le client pour Amazon Location

Vous pouvez spécifier une clé gérée par le client en tant que seconde couche de chiffrement pour les ressources suivantes :

- [Ressource de suivi](#)
- [Collection Geofence](#)

Lorsque vous créez une ressource, vous pouvez spécifier la clé de données en saisissant un identifiant KMS, qu'Amazon Location utilise pour chiffrer les données personnelles identifiables stockées par la ressource.

- ID KMS : [identifiant de clé](#) pour une clé gérée par AWS KMS le client. Saisissez un ID de clé, un ARN de clé, un nom d'alias ou un ARN d'alias.

Contexte de chiffrement Amazon Location Service

Un [contexte de chiffrement](#) est un ensemble facultatif de paires clé-valeur qui contient des informations contextuelles supplémentaires sur les données.

AWS KMS utilise le contexte de chiffrement comme [données authentifiées supplémentaires](#) pour prendre en charge le chiffrement [authentifié](#). Lorsque vous incluez un contexte de chiffrement dans une demande de chiffrement de données, AWS KMS lie le contexte de chiffrement aux données chiffrées. Pour déchiffrer les données, vous devez inclure le même contexte de chiffrement dans la demande.

Contexte de chiffrement Amazon Location Service

Amazon Location utilise le même contexte de chiffrement dans toutes les opérations AWS KMS cryptographiques, où la clé `aws:geo:arn` et la valeur sont le [nom de ressource Amazon](#) (ARN) de la ressource.

Exemple

```
"encryptionContext": {
  "aws:geo:arn": "arn:aws:geo:us-west-2:111122223333:geofence-collection/SAMPLE-GeofenceCollection"
}
```

Utilisation du contexte de chiffrement pour la surveillance

Lorsque vous utilisez une clé symétrique gérée par le client pour chiffrer votre collection de traceurs ou de géofences, vous pouvez également utiliser le contexte de chiffrement dans les dossiers d'audit et les journaux pour identifier la manière dont la clé gérée par le client est utilisée. Le contexte de chiffrement apparaît également dans les [journaux générés par Amazon Logs AWS CloudTrail ou Amazon CloudWatch Logs](#).

Utilisation du contexte de chiffrement pour contrôler l'accès à votre clé gérée par le client

Vous pouvez utiliser le contexte de chiffrement dans les stratégies de clé et les politiques IAM comme conditions pour contrôler l'accès à votre clé symétrique gérée par le client. Vous pouvez également utiliser des contraintes de contexte de chiffrement dans un octroi.

Amazon Location utilise une contrainte de contexte de chiffrement dans les autorisations afin de contrôler l'accès à la clé gérée par le client dans votre compte ou votre région. La contrainte d'octroi exige que les opérations autorisées par l'octroi utilisent le contexte de chiffrement spécifié.

Exemple

Vous trouverez ci-dessous des exemples de déclarations de stratégie de clé permettant d'accorder l'accès à une clé gérée par le client dans un contexte de chiffrement spécifique. La condition énoncée

dans cette déclaration de stratégie exige que les octrois comportent une contrainte de contexte de chiffrement qui spécifie le contexte de chiffrement.

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

Surveillance de vos clés de chiffrement pour Amazon Location Service

Lorsque vous utilisez une clé gérée par le AWS KMS client avec vos ressources Amazon Location Service, vous pouvez utiliser [AWS CloudTrail Amazon CloudWatch Logs](#) pour suivre les demandes auxquelles Amazon Location envoie AWS KMS.

Les exemples suivants sont AWS CloudTrail des événements pour `CreateGrant`, `GenerateDataKeyWithoutPlainTextDecrypt`, et `DescribeKey` pour surveiller les opérations KMS appelées par Amazon Location afin d'accéder aux données chiffrées par votre clé gérée par le client :

CreateGrant

Lorsque vous utilisez une clé gérée par le AWS KMS client pour chiffrer vos ressources de collecte de données de suivi ou de géofence, Amazon Location envoie une `CreateGrant` demande en votre nom pour accéder à la clé KMS de votre compte. AWS Les subventions créées par Amazon Location sont spécifiques à la ressource associée à la clé gérée par le AWS KMS client. En outre, Amazon Location utilise cette `RetireGrant` opération pour supprimer une subvention lorsque vous supprimez une ressource.

L'exemple d'événement suivant enregistre l'opération `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

Lorsque vous activez une clé gérée par AWS KMS le client pour votre ressource de collecte de données de suivi ou de géofence, Amazon Location crée une clé de table unique. Il envoie une `GenerateDataKeyWithoutPlainText` demande AWS KMS qui spécifie la clé gérée par le AWS KMS client pour la ressource.

L'exemple d'événement suivant enregistre l'opération `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

Lorsque vous accédez à un tracker chiffré ou à une collection de géofences, Amazon Location appelle l'opération `Decrypt` pour utiliser la clé de données cryptée stockée afin d'accéder aux données cryptées.

L'exemple d'événement suivant enregistre l'opération `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 utilise cette `DescribeKey` opération pour vérifier si la clé gérée par le AWS KMS client associée à votre collection de traceurs ou de géofences existe dans le compte et dans la région.

L'exemple d'événement suivant enregistre l'opération `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"
}
```

En savoir plus

Les ressources suivantes fournissent plus d'informations sur le chiffrement des données au repos.

- Pour plus d'informations sur les [concepts de base AWS Key Management Service](#), consultez le Guide du développeur AWS Key Management Service .
- Pour plus d'informations sur les [meilleures pratiques de sécurité pour AWS Key Management Service](#), consultez le guide du AWS Key Management Service développeur.

Chiffrement des données en transit pour Amazon Location Service

Amazon Location protège les données en transit, lorsqu'elles sont acheminées vers et depuis le service, en chiffrant automatiquement toutes les données interréseaux à l'aide du protocole de chiffrement TLS (Transport Layer Security) 1.2. Les demandes HTTPS directes envoyées aux API Amazon Location Service sont signées à l'aide de l'[algorithme AWS Signature Version 4](#) pour établir une connexion sécurisée.

Identity and Access Management pour Amazon Location Service

AWS Identity and Access Management (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM

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

Rubriques

- [Public ciblé](#)
- [Authentification par des identités](#)
- [Gestion des accès à l'aide de politiques](#)
- [Comment Amazon Location Service fonctionne avec IAM](#)
- [Comment Amazon Location Service fonctionne avec les utilisateurs non authentifiés](#)
- [Exemples de politiques basées sur l'identité pour Amazon Location Service](#)
- [Résolution des problèmes d'identité et d'accès à Amazon Location Service](#)

Public ciblé

La façon dont vous utilisez AWS Identity and Access Management (IAM) varie en fonction du travail que vous effectuez sur Amazon Location.

Utilisateur du service : si vous utilisez le service Amazon Location pour effectuer votre travail, votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin. Au fur et à mesure que vous utilisez de plus en plus les fonctionnalités d'Amazon Location dans le cadre de votre travail, vous aurez peut-être besoin d'autorisations supplémentaires. En comprenant bien la gestion des accès, vous saurez demander les autorisations appropriées à votre administrateur. Si vous ne parvenez pas à accéder à une fonctionnalité dans Amazon Location, consultez [Résolution des problèmes d'identité et d'accès à Amazon Location Service](#).

Administrateur du service — Si vous êtes responsable des ressources Amazon Location au sein de votre entreprise, vous avez probablement un accès complet à Amazon Location. C'est à vous de déterminer les fonctionnalités et les ressources d'Amazon Location auxquelles les utilisateurs de votre service doivent accéder. Vous devez ensuite soumettre les demandes à votre administrateur IAM pour modifier les autorisations des utilisateurs de votre service. Consultez les informations sur cette page pour comprendre les concepts de base d'IAM. Pour en savoir plus sur la manière dont votre entreprise peut utiliser IAM avec Amazon Location, consultez [Comment Amazon Location Service fonctionne avec IAM](#).

Administrateur IAM : si vous êtes administrateur IAM, vous souhaitez peut-être en savoir plus sur la manière dont vous pouvez rédiger des politiques pour gérer l'accès à Amazon Location. Pour

consulter des exemples de politiques basées sur l'identité Amazon Location que vous pouvez utiliser dans IAM, consultez. [Exemples de politiques basées sur l'identité pour Amazon Location Service](#)

Authentification par des identités

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

Vous pouvez vous connecter en AWS tant qu'identité fédérée en utilisant les informations d'identification fournies par le biais d'une source d'identité. AWS IAM Identity Center Les utilisateurs (IAM Identity Center), l'authentification unique de votre entreprise et vos informations d'identification Google ou Facebook sont des exemples d'identités fédérées. Lorsque vous vous connectez avec une identité fédérée, votre administrateur aura précédemment configuré une fédération d'identités avec des rôles IAM. Lorsque vous accédez à AWS l'aide de la fédération, vous assumez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter au portail AWS Management Console ou au portail AWS d'accès. Pour plus d'informations sur la connexion à AWS, consultez la section [Comment vous connecter à votre compte Compte AWS dans](#) le guide de Connexion à AWS l'utilisateur.

Si vous y accédez AWS par programmation, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos demandes à l'aide de vos informations d'identification. Si vous n'utilisez pas d' AWS outils, vous devez signer vous-même les demandes. Pour plus d'informations sur l'utilisation de la méthode recommandée pour signer vous-même les demandes, consultez la section [Signature des demandes AWS d'API](#) dans le guide de l'utilisateur IAM.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, il vous AWS recommande d'utiliser l'authentification multifactorielle (MFA) pour renforcer la sécurité de votre compte. Pour en savoir plus, consultez [Authentification multifactorielle](#) dans le Guide de l'utilisateur AWS IAM Identity Center et [Utilisation de l'authentification multifactorielle \(MFA\) dans l'interface AWS](#) dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une identité de connexion unique qui donne un accès complet à toutes Services AWS les ressources du compte. Cette identité est

appelée utilisateur Compte AWS root et est accessible en vous connectant avec l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur racine et utilisez-les pour effectuer les tâches que seul l'utilisateur racine peut effectuer. Pour obtenir la liste complète des tâches qui vous imposent de vous connecter en tant qu'utilisateur root, consultez [Tâches nécessitant des informations d'identification d'utilisateur root](#) dans le Guide de l'utilisateur IAM.

Identité fédérée

La meilleure pratique consiste à obliger les utilisateurs humains, y compris les utilisateurs nécessitant un accès administrateur, à utiliser la fédération avec un fournisseur d'identité pour accéder à l'aide Services AWS d'informations d'identification temporaires.

Une identité fédérée est un utilisateur de l'annuaire des utilisateurs de votre entreprise, d'un fournisseur d'identité Web AWS Directory Service, du répertoire Identity Center ou de tout utilisateur qui y accède à l'aide des informations d'identification fournies Services AWS par le biais d'une source d'identité. Lorsque des identités fédérées y accèdent Comptes AWS, elles assument des rôles, qui fournissent des informations d'identification temporaires.

Pour une gestion des accès centralisée, nous vous recommandons d'utiliser AWS IAM Identity Center. Vous pouvez créer des utilisateurs et des groupes dans IAM Identity Center, ou vous pouvez vous connecter et synchroniser avec un ensemble d'utilisateurs et de groupes dans votre propre source d'identité afin de les utiliser dans toutes vos applications Comptes AWS et applications. Pour obtenir des informations sur IAM Identity Center, consultez [Qu'est-ce que IAM Identity Center ?](#) dans le Guide de l'utilisateur AWS IAM Identity Center .

Utilisateurs et groupes IAM

Un [utilisateur IAM](#) est une identité au sein de votre Compte AWS qui possède des autorisations spécifiques pour une seule personne ou application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des utilisateurs IAM ayant des informations d'identification à long terme tels que les clés d'accès. Toutefois, si certains cas d'utilisation spécifiques nécessitent des informations d'identification à long terme avec les utilisateurs IAM, nous vous recommandons de faire pivoter les clés d'accès. Pour plus d'informations, consultez [Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification](#) dans le Guide de l'utilisateur IAM.

Un [groupe IAM](#) est une identité qui concerne un ensemble d'utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez avoir un groupe nommé IAMAdmins et accorder à ce groupe les autorisations d'administrer des ressources IAM.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour en savoir plus, consultez [Quand créer un utilisateur IAM \(au lieu d'un rôle\)](#) dans le Guide de l'utilisateur IAM.

Rôles IAM

Un [rôle IAM](#) est une identité au sein de votre Compte AWS dotée d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais le rôle IAM n'est pas associé à une personne en particulier. Vous pouvez assumer temporairement un rôle IAM dans le en AWS Management Console [changeant de rôle](#). Vous pouvez assumer un rôle en appelant une opération d' AWS API AWS CLI ou en utilisant une URL personnalisée. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez [Utilisation de rôles IAM](#) dans le Guide de l'utilisateur IAM.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- Accès utilisateur fédéré – Pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour obtenir des informations sur les rôles pour la fédération, consultez [Création d'un rôle pour un fournisseur d'identité tiers \(fédération\)](#) dans le Guide de l'utilisateur IAM. Si vous utilisez IAM Identity Center, vous configurez un jeu d'autorisations. IAM Identity Center met en corrélation le jeu d'autorisations avec un rôle dans IAM afin de contrôler à quoi vos identités peuvent accéder après leur authentification. Pour plus d'informations sur les jeux d'autorisations, consultez la rubrique [Jeux d'autorisations](#) dans le Guide de l'utilisateur AWS IAM Identity Center .
- Autorisations d'utilisateur IAM temporaires : un rôle ou un utilisateur IAM peut endosser un rôle IAM pour profiter temporairement d'autorisations différentes pour une tâche spécifique.
- Accès intercompte : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (principal de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, dans certains Services AWS cas, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme

proxy). Pour connaître la différence entre les rôles et les politiques basées sur les ressources pour l'accès entre comptes, consultez la section Accès aux [ressources entre comptes dans IAM dans le guide de l'utilisateur IAM](#).

- Accès multiservices — Certains Services AWS utilisent des fonctionnalités dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, un rôle de service ou un rôle lié au service.
- Sessions d'accès direct (FAS) : lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur la politique relative à la transmission de demandes FAS, consultez [Sessions de transmission d'accès](#).
- Rôle de service : il s'agit d'un [rôle IAM](#) attribué à un service afin de réaliser des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d'IAM. Pour plus d'informations, consultez [Création d'un rôle pour la délégation d'autorisations à un Service AWS](#) dans le Guide de l'utilisateur IAM.
- Rôle lié à un service — Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.
- Applications exécutées sur Amazon EC2 : vous pouvez utiliser un rôle IAM pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une instance EC2 et qui envoient des demandes d'API. AWS CLI AWS Cette solution est préférable au stockage des clés d'accès au sein de l'instance EC2. Pour attribuer un AWS rôle à une instance EC2 et le mettre à la disposition de toutes ses applications, vous devez créer un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle et permet aux programmes qui s'exécutent sur l'instance EC2 d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez [Utilisation d'un rôle IAM pour accorder des autorisations à des applications s'exécutant sur des instances Amazon EC2](#) dans le Guide de l'utilisateur IAM.

Pour savoir dans quel cas utiliser des rôles ou des utilisateurs IAM, consultez [Quand créer un rôle IAM \(au lieu d'un utilisateur\)](#) dans le Guide de l'utilisateur IAM.

Gestion des accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique est un objet AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit leurs autorisations. AWS évalue ces politiques lorsqu'un principal (utilisateur, utilisateur root ou session de rôle) fait une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations sur la structure et le contenu des documents de politique JSON, consultez [Vue d'ensemble des politiques JSON](#) dans le Guide de l'utilisateur IAM.

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

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Les politiques IAM définissent les autorisations d'une action, quelle que soit la méthode que vous utilisez pour exécuter l'opération. Par exemple, supposons que vous disposiez d'une politique qui autorise l'action `iam:GetRole`. Un utilisateur appliquant cette politique peut obtenir des informations sur le rôle à partir de AWS Management Console AWS CLI, de ou de l' AWS API.

Politiques basées sur l'identité

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez [Création de politiques IAM](#) dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez associer à plusieurs

utilisateurs, groupes et rôles au sein de votre Compte AWS. Les politiques gérées incluent les politiques AWS gérées et les politiques gérées par le client. Pour découvrir comment choisir entre une politique gérée et une politique en ligne, consultez [Choix entre les politiques gérées et les politiques en ligne](#) dans le Guide de l'utilisateur IAM.

politiques basées sur les ressources

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

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

Listes de contrôle d'accès (ACL)

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

Amazon S3 et Amazon VPC sont des exemples de services qui prennent en charge les ACL. AWS WAF Pour en savoir plus sur les listes de contrôle d'accès, consultez [Vue d'ensemble des listes de contrôle d'accès \(ACL\)](#) dans le Guide du développeur Amazon Simple Storage Service.

Autres types de politique

AWS prend en charge d'autres types de politiques moins courants. Ces types de politiques peuvent définir le nombre maximum d'autorisations qui vous sont accordées par des types de politiques plus courants.

- **Limite d'autorisations** : une limite d'autorisations est une fonctionnalité avancée dans laquelle vous définissez le nombre maximal d'autorisations qu'une politique basée sur l'identité peut accorder

à une entité IAM (utilisateur ou rôle IAM). Vous pouvez définir une limite d'autorisations pour une entité. Les autorisations en résultant représentent la combinaison des politiques basées sur l'identité d'une entité et de ses limites d'autorisation. Les politiques basées sur les ressources qui spécifient l'utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces politiques remplace l'autorisation. Pour plus d'informations sur les limites d'autorisations, consultez [Limites d'autorisations pour des entités IAM](#) dans le Guide de l'utilisateur IAM.

- **Politiques de contrôle des services (SCP)** — Les SCP sont des politiques JSON qui spécifient les autorisations maximales pour une organisation ou une unité organisationnelle (UO) dans AWS Organizations. AWS Organizations est un service permettant de regrouper et de gérer de manière centralisée les multiples comptes AWS de votre entreprise. Si vous activez toutes les fonctionnalités d'une organisation, vous pouvez appliquer les politiques de contrôle des services (SCP) à l'un ou à l'ensemble de vos comptes. Le SCP limite les autorisations pour les entités figurant dans les comptes des membres, y compris chaque utilisateur racine d'un compte AWS d'entre elles. Pour plus d'informations sur les organisations et les SCP, consultez [Fonctionnement des SCP](#) dans le Guide de l'utilisateur AWS Organizations.
- **Politiques de séance** : les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de séance en résultant sont une combinaison des politiques basées sur l'identité de l'utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d'une politique basée sur les ressources. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour plus d'informations, consultez [politiques de séance](#) dans le Guide de l'utilisateur IAM.

Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS détermine s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section [Logique d'évaluation des politiques](#) dans le guide de l'utilisateur IAM.

Comment Amazon Location Service fonctionne avec IAM

Avant d'utiliser IAM pour gérer l'accès à Amazon Location, découvrez quelles fonctionnalités IAM peuvent être utilisées avec Amazon Location.

Fonctionnalités IAM que vous pouvez utiliser avec Amazon Location Service

Fonction IAM	Assistance Amazon Location
Politiques basées sur l'identité	Oui
Politiques basées sur les ressources	Non
Actions de politique	Oui
Ressources de politique	Oui
Clés de condition de politique (spécifiques au service)	Oui
ACL	Non
ABAC (étiquettes dans les politiques)	Oui
Informations d'identification temporaires	Oui
Autorisations de principaux	Non
Fonctions du service	Non
Rôles liés à un service	Non

Pour obtenir une vue d'ensemble de la façon dont Amazon Location et les autres AWS services fonctionnent avec la plupart des fonctionnalités IAM, consultez les [AWS services compatibles avec IAM](#) dans le guide de l'utilisateur IAM.

Politiques basées sur l'identité pour Amazon Location

Prend en charge les politiques basées sur l'identité	Oui
--	-----

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles

ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez [Création de politiques IAM](#) dans le Guide de l'utilisateur IAM.

Avec les politiques IAM basées sur l'identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Vous ne pouvez pas spécifier le principal dans une politique basée sur une identité car celle-ci s'applique à l'utilisateur ou au rôle auquel elle est attachée. Pour découvrir tous les éléments que vous utilisez dans une politique JSON, consultez [Références des éléments de politique JSON IAM](#) dans le Guide de l'utilisateur IAM.

Exemples de politiques basées sur l'identité pour Amazon Location

Pour consulter des exemples de politiques basées sur l'identité d'Amazon Location, consultez [Exemples de politiques basées sur l'identité pour Amazon Location Service](#)

Politiques basées sur les ressources au sein d'Amazon Location

Prend en charge les politiques basées sur les ressources	Non
--	-----

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

Pour permettre un accès intercompte, vous pouvez spécifier un compte entier ou des entités IAM dans un autre compte en tant que principal dans une politique basée sur les ressources. L'ajout d'un principal entre comptes à une politique basée sur les ressources ne représente qu'une partie de l'instauration de la relation d'approbation. Lorsque le principal et la ressource sont différents Comptes AWS, un administrateur IAM du compte sécurisé doit également accorder à l'entité principale (utilisateur ou rôle) l'autorisation d'accéder à la ressource. Pour ce faire, il attache une politique basée sur une identité à l'entité. Toutefois, si une politique basée sur des ressources

accorde l'accès à un principal dans le même compte, aucune autre politique basée sur l'identité n'est requise. Pour plus d'informations, consultez [la section Accès aux ressources entre comptes dans IAM](#) dans le guide de l'utilisateur d'IAM.

Actions politiques pour Amazon Location

Prend en charge les actions de politique	Oui
--	-----

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

L'élément `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Les actions de stratégie portent généralement le même nom que l'opération AWS d'API associée. Il existe quelques exceptions, telles que les actions avec autorisations uniquement qui n'ont pas d'opération API correspondante. Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une stratégie afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des actions Amazon Location, consultez la section [Actions définies par Amazon Location Service](#) dans la référence d'autorisation du service.

Les actions politiques dans Amazon Location utilisent le préfixe suivant avant l'action :

```
geo
```

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

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

Vous pouvez aussi spécifier plusieurs actions à l'aide de caractères génériques (*). Par exemple, pour spécifier toutes les actions qui commencent par le mot Get, incluez l'action suivante :

```
"Action": "geo:Get*"
```

Pour consulter des exemples de politiques basées sur l'identité d'Amazon Location, consultez [Exemples de politiques basées sur l'identité pour Amazon Location Service](#)

Ressources relatives aux politiques pour Amazon Location

Prend en charge les ressources de politique Oui

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

L'élément de politique JSON `Resource` indique le ou les objets auxquels l'action s'applique. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de définir une ressource à l'aide de son [Amazon Resource Name \(ARN\)](#). Vous pouvez le faire pour des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

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

Pour consulter la liste des types de ressources Amazon Location et de leurs ARN, consultez la section [Ressources définies par Amazon Location Service](#) dans la référence d'autorisation du service. Pour savoir avec quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez [Actions définies par Amazon Location Service](#).

Pour consulter des exemples de politiques basées sur l'identité d'Amazon Location, consultez [Exemples de politiques basées sur l'identité pour Amazon Location Service](#)

Clés relatives aux conditions de politique pour Amazon Location

Prend en charge les clés de condition de politique spécifiques au service	Oui
---	-----

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

L'élément `Condition` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. Vous pouvez créer des expressions conditionnelles qui utilisent des [opérateurs de condition](#), tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande.

Si vous spécifiez plusieurs éléments `Condition` dans une instruction, ou plusieurs clés dans un seul élément `Condition`, AWS les évalue à l'aide d'une opération AND logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une OR opération logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez accorder à un utilisateur IAM l'autorisation d'accéder à une ressource uniquement si elle est balisée avec son nom d'utilisateur IAM. Pour plus d'informations, consultez [Éléments d'une politique IAM : variables et identifications](#) dans le Guide de l'utilisateur IAM.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques au service. Pour voir toutes les clés de condition AWS globales, voir les clés de [contexte de condition AWS globales](#) dans le guide de l'utilisateur IAM.

Pour consulter la liste des clés de condition Amazon Location, consultez la section [Clés de condition pour Amazon Location Service](#) dans la référence d'autorisation du service. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, consultez [Actions définies par Amazon Location Service](#).

Amazon Location prend en charge les clés de condition pour vous permettre d'autoriser ou de refuser l'accès à des barrières géographiques ou à des appareils spécifiques dans vos déclarations de politique. Les clés de condition suivantes sont disponibles :

- `geo:GeofenceIds` à utiliser avec les actions `Geofence`. Le type est `ArrayOfString`.
- `geo:DeviceIds` à utiliser avec les actions `Tracker`. Le type est `ArrayOfString`.

Les actions suivantes peuvent être utilisées `geo:GeofenceIds` dans votre politique IAM :

- `BatchDeleteGeofences`
- `BatchPutGeofences`
- `GetGeofence`
- `PutGeofence`

Les actions suivantes peuvent être utilisées `geo:DeviceIds` dans votre politique IAM :

- `BatchDeleteDevicePositionHistory`
- `BatchGetDevicePosition`
- `BatchUpdateDevicePosition`
- `GetDevicePosition`
- `GetDevicePositionHistory`

Note

Vous ne pouvez pas utiliser ces clés de condition avec les `ListDevicePosition` actions `BatchEvaluateGeofencesListGeofences`, ou.

Pour consulter des exemples de politiques basées sur l'identité d'Amazon Location, consultez.

[Exemples de politiques basées sur l'identité pour Amazon Location Service](#)

ACL sur Amazon Location

Prend en charge les listes ACL

Non

Les listes de contrôle d'accès (ACL) vérifient quels principaux (membres de compte, utilisateurs ou rôles) ont l'autorisation d'accéder à une ressource. Les listes de contrôle d'accès sont similaires aux

politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

ABAC avec Amazon Location

Prend en charge ABAC (étiquettes dans les politiques)	Oui
---	-----

Le contrôle d'accès par attributs (ABAC) est une stratégie d'autorisation qui définit des autorisations en fonction des attributs. Dans AWS, ces attributs sont appelés balises. Vous pouvez associer des balises aux entités IAM (utilisateurs ou rôles) et à de nombreuses AWS ressources. L'étiquetage des entités et des ressources est la première étape d'ABAC. Vous concevez ensuite des politiques ABAC pour autoriser des opérations quand l'identification du principal correspond à celle de la ressource à laquelle il tente d'accéder.

L'ABAC est utile dans les environnements qui connaissent une croissance rapide et pour les cas où la gestion des politiques devient fastidieuse.

Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans [l'élément de condition](#) d'une politique utilisant les clés de condition `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

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

Pour plus d'informations sur l'ABAC, consultez [Qu'est-ce que le contrôle d'accès basé sur les attributs \(ABAC\) ?](#) dans le Guide de l'utilisateur IAM. Pour accéder à un didacticiel décrivant les étapes de configuration de l'ABAC, consultez [Utilisation du contrôle d'accès par attributs \(ABAC\)](#) dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur le balisage des ressources Amazon Location, consultez [Marquage de vos ressources Amazon Location Service](#).

Pour visualiser un exemple de politique basée sur l'identité permettant de limiter l'accès à une ressource en fonction des balises de cette ressource, consultez [Contrôlez l'accès aux ressources en fonction des balises](#).

Utilisation d'informations d'identification temporaires avec Amazon Location

Prend en charge les informations d'identification temporaires	Oui
---	-----

Certains Services AWS ne fonctionnent pas lorsque vous vous connectez à l'aide d'informations d'identification temporaires. Pour plus d'informations, y compris celles qui Services AWS fonctionnent avec des informations d'identification temporaires, consultez Services AWS la section relative à l'utilisation [d'IAM](#) dans le guide de l'utilisateur d'IAM.

Vous utilisez des informations d'identification temporaires si vous vous connectez à l' AWS Management Console aide d'une méthode autre qu'un nom d'utilisateur et un mot de passe. Par exemple, lorsque vous accédez à AWS l'aide du lien d'authentification unique (SSO) de votre entreprise, ce processus crée automatiquement des informations d'identification temporaires. Vous créez également automatiquement des informations d'identification temporaires lorsque vous vous connectez à la console en tant qu'utilisateur, puis changez de rôle. Pour plus d'informations sur le changement de rôle, consultez [Changement de rôle \(console\)](#) dans le Guide de l'utilisateur IAM.

Vous pouvez créer manuellement des informations d'identification temporaires à l'aide de l' AWS API AWS CLI or. Vous pouvez ensuite utiliser ces informations d'identification temporaires pour y accéder AWS. AWS recommande de générer dynamiquement des informations d'identification temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez [Informations d'identification de sécurité temporaires dans IAM](#).

Autorisations principales interservices pour Amazon Location

Prend en charge les sessions d'accès direct (FAS)	Non
---	-----

Lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour

être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur une politique lors de la formulation de demandes FAS, consultez [Transmission des sessions d'accès](#).

Rôles de service pour Amazon Location

Prend en charge les fonctions de service	Non
--	-----

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

Warning

La modification des autorisations associées à un rôle de service peut perturber les fonctionnalités d'Amazon Location. Modifiez les rôles de service uniquement lorsque Amazon Location fournit des instructions à cet effet.

Rôles liés à un service pour Amazon Location

Prend en charge les rôles liés à un service	Non
---	-----

Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

Pour plus d'informations sur la création ou la gestion des rôles liés à un service, consultez [Services AWS qui fonctionnent avec IAM](#). Recherchez un service dans le tableau qui inclut un Yes dans la colonne Rôle lié à un service. Choisissez le lien Oui pour consulter la documentation du rôle lié à ce service.

Comment Amazon Location Service fonctionne avec les utilisateurs non authentifiés

De nombreux scénarios d'utilisation d'Amazon Location Service, notamment l'affichage de cartes sur le Web ou dans une application mobile, nécessitent d'autoriser l'accès aux utilisateurs qui ne se sont pas connectés avec IAM. Pour ces scénarios non authentifiés, deux options s'offrent à vous.

- Utiliser des clés d'API : pour accorder l'accès à des utilisateurs non authentifiés, vous pouvez créer des clés d'API qui donnent un accès en lecture seule à vos ressources Amazon Location Service. Cela est utile dans le cas où vous ne souhaitez pas authentifier tous les utilisateurs. Par exemple, une application Web. Pour plus d'informations sur les clés d'API, consultez [Autoriser l'accès invité non authentifié à votre application à l'aide de clés d'API](#).
- Utiliser Amazon Cognito — Une alternative aux clés d'API consiste à utiliser Amazon Cognito pour accorder un accès anonyme. Amazon Cognito vous permet de créer une autorisation plus riche avec une politique IAM afin de définir ce que peuvent faire les utilisateurs non authentifiés. Pour plus d'informations sur l'utilisation d'Amazon Cognito, consultez [Autoriser l'accès invité non authentifié à votre application à l'aide d'Amazon Cognito](#)

Pour une vue d'ensemble de la fourniture d'accès aux utilisateurs non authentifiés, voir [Octroi de l'accès à Amazon Location Service](#)

Exemples de politiques basées sur l'identité pour Amazon Location Service

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou à modifier les ressources Amazon Location. Ils ne peuvent pas non plus effectuer de tâches à l'aide de l'API AWS Management Console, AWS Command Line Interface (AWS CLI) ou de AWS l'API. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

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

Pour plus de détails sur les actions et les types de ressources définis par Amazon Location, y compris le format des ARN pour chacun des types de ressources, consultez la section [Actions, ressources et clés de condition pour Amazon Location Service](#) dans la référence d'autorisation du service.

Rubriques

- [Bonnes pratiques en matière de politiques](#)
- [Utilisation de la console Amazon Location](#)
- [Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations](#)
- [Utilisation des ressources Amazon Location Service dans le cadre de la politique](#)
- [Autorisations pour mettre à jour les positions des appareils](#)
- [Politique de lecture seule pour les ressources de suivi](#)
- [Politique de création de géofences](#)
- [Politique de lecture seule pour les géofences](#)
- [Autorisations pour le rendu d'une ressource cartographique](#)
- [Autorisations pour autoriser les opérations de recherche](#)
- [Politique de lecture seule pour les calculateurs d'itinéraires](#)
- [Contrôlez l'accès aux ressources en fonction des clés de condition](#)
- [Contrôlez l'accès aux ressources en fonction des balises](#)

Bonnes pratiques en matière de politiques

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

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez [politiques gérées par AWS](#) ou [politiques gérées par AWS pour les activités professionnelles](#) dans le Guide de l'utilisateur IAM.
- Accorder les autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de

moindre privilège. Pour plus d'informations sur l'utilisation de IAM pour appliquer des autorisations, consultez [politiques et autorisations dans IAM](#) dans le Guide de l'utilisateur IAM.

- Utiliser des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que AWS CloudFormation. Pour plus d'informations, consultez [Conditions pour éléments de politique JSON IAM](#) dans le Guide de l'utilisateur IAM.
- Utilisez IAM Access Analyzer pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : IAM Access Analyzer valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez [Validation de politique IAM Access Analyzer](#) dans le Guide de l'utilisateur IAM.
- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue. Compte AWS Pour exiger le MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez [Configuration de l'accès aux API protégé par MFA](#) dans le Guide de l'utilisateur IAM.

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

Utilisation de la console Amazon Location

Pour accéder à la console Amazon Location Service, vous devez disposer d'un ensemble minimal d'autorisations. Ces autorisations doivent vous permettre de répertorier et de consulter les informations relatives aux ressources Amazon Location présentes dans votre Compte AWS. Si vous créez une stratégie basée sur l'identité qui est plus restrictive que l'ensemble minimum d'autorisations requis, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs ou rôles) tributaires de cette stratégie.

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

Pour garantir que les utilisateurs et les rôles peuvent utiliser la console Amazon Location, associez la politique suivante aux entités. Pour plus d'informations, consultez [Ajout d'autorisations à un utilisateur](#) dans le Guide de l'utilisateur IAM.

La politique suivante donne accès à la console Amazon Location Service, afin de pouvoir créer, supprimer, répertorier et consulter les informations relatives aux ressources Amazon Location dans votre AWS compte.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GeoPowerUser",
      "Effect": "Allow",
      "Action": [
        "geo:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Vous pouvez également accorder des autorisations en lecture seule pour faciliter l'accès en lecture seule. Avec les autorisations en lecture seule, un message d'erreur apparaît si l'utilisateur tente d'effectuer des actions d'écriture telles que la création ou la suppression de ressources. À titre d'exemple, voir [the section called "Politique de lecture seule pour les traceurs"](#)

Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

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

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
```

```

        "iam:GetUserPolicy",
        "iam:ListGroupsForUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
    ],
    "Resource": ["arn:aws:iam::*:user/${aws:username}"]
},
{
    "Sid": "NavigateInConsole",
    "Effect": "Allow",
    "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
    ],
    "Resource": "*"
}
]
}

```

Utilisation des ressources Amazon Location Service dans le cadre de la politique

Amazon Location Service utilise les préfixes suivants pour les ressources :

Préfixe de ressource Amazon Location

Ressource	Préfixe de ressource
Ressources cartographiques	map
Placer des ressources	place-index
Ressources relatives aux itinéraires	route-calculator
Ressources de suivi	tracker
Ressources de la collection Geofence	geofence-collection

Utilisez la syntaxe ARN suivante :

```
arn:Partition:geo:Region:Account:ResourcePrefix/ResourceName
```

Pour plus d'informations sur le format des ARN, consultez [Amazon Resource Names \(ARN\) et AWS Service Namespaces](#).

Exemples

- Utilisez l'ARN suivant pour autoriser l'accès à une ressource cartographique spécifiée.

```
"Resource": "arn:aws:geo:us-west-2:account-id:map/map-resource-name"
```

- Pour spécifier l'accès à toutes les map ressources appartenant à un compte spécifique, utilisez le caractère générique (*) :

```
"Resource": "arn:aws:geo:us-west-2:account-id:map/*"
```

- Certaines actions Amazon Location, telles que celles relatives à la création de ressources, ne peuvent pas être effectuées sur une ressource spécifique. Dans ces cas-là, vous devez utiliser le caractère générique (*).

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

Pour consulter la liste des types de ressources Amazon Location et de leurs ARN, consultez la section [Ressources définies par Amazon Location Service](#) dans la référence d'autorisation du service. Pour savoir avec quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez [Actions définies par Amazon Location Service](#).

Autorisations pour mettre à jour les positions des appareils

Pour mettre à jour les positions de plusieurs appareils de suivi, vous devez autoriser un utilisateur à accéder à une ou plusieurs de vos ressources de suivi. Vous souhaitez également autoriser l'utilisateur à mettre à jour un lot de positions d'appareils.

Dans cet exemple, en plus d'accorder l'accès aux ressources Tracker1 et Tracker2, la politique suivante autorise l'utilisation de l'geo:BatchUpdateDevicePositionaction sur les ressources Tracker1 et 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 vous souhaitez limiter l'utilisateur afin qu'il puisse uniquement mettre à jour les positions de l'appareil pour un appareil spécifique, vous pouvez ajouter une clé de condition pour cet identifiant d'appareil.

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

```
]
}
```

Politique de lecture seule pour les ressources de suivi

Pour créer une politique de lecture seule pour toutes les ressources de suivi de votre AWS compte, vous devez autoriser l'accès à toutes les ressources de suivi. Vous devez également autoriser un utilisateur à accéder aux actions qui lui permettent d'obtenir la position de plusieurs appareils, d'obtenir la position de l'appareil à partir d'un seul appareil et d'obtenir l'historique des positions.

Dans cet exemple, la politique suivante autorise les actions suivantes :

- `geo:BatchGetDevicePosition` pour récupérer la position de plusieurs appareils.
- `geo:GetDevicePosition` pour récupérer la position d'un seul appareil.
- `geo:GetDevicePositionHistory` pour récupérer l'historique des positions d'un appareil.

```
{
  "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/*"
    }
  ]
}
```

Politique de création de géofences

Pour créer une politique autorisant un utilisateur à créer des géofences, vous devez autoriser l'accès à des actions spécifiques qui permettent aux utilisateurs de créer une ou plusieurs géofences sur une collection de géofences.

La politique ci-dessous autorise les actions suivantes sur la *collecte* :

- `geo:BatchPutGeofence` pour créer plusieurs géofences.
- `geo:PutGeofence` pour créer une géofence unique.

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

Politique de lecture seule pour les géofences

Pour créer une politique de lecture seule pour les géofences stockées dans une collection de géofences de votre AWS compte, vous devez autoriser l'accès aux actions qui lisent à partir de la collection de géofences stockant les géofences.

La politique ci-dessous autorise les actions suivantes sur la *collecte* :

- `geo:ListGeofences` pour répertorier les géofences dans la collection de géofences spécifiée.
- `geo:GetGeofence` pour récupérer une géofence dans la collection de géofences.

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

Autorisations pour le rendu d'une ressource cartographique

Pour accorder des autorisations suffisantes pour afficher des cartes, vous devez autoriser l'accès aux vignettes de carte, aux sprites, aux glyphes et au descripteur de style :

- `geo:GetMapTile` récupère les tuiles de carte utilisées pour afficher de manière sélective les entités d'une carte.
- `geo:GetMapSprites` récupère la feuille de sprites PNG et le document JSON correspondant décrivant les décalages qu'elle contient.
- `geo:GetMapGlyphs` récupère les glyphes utilisés pour afficher le texte.
- `geo:GetMapStyleDescriptor` récupère le descripteur de style de la carte, qui contient les règles de rendu.

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

Autorisations pour autoriser les opérations de recherche

Pour créer une politique autorisant les opérations de recherche, vous devez d'abord autoriser l'accès à la ressource d'index des lieux de votre AWS compte. Vous souhaitez également autoriser l'accès

aux actions qui permettent à l'utilisateur d'effectuer une recherche à l'aide de texte par géocodage et de rechercher à l'aide d'une position par géocodage inversé.

Dans cet exemple, en plus d'accorder l'accès à *PlaceIndex*, la politique suivante autorise également les actions suivantes :

- `geo:SearchPlaceIndexForPosition` vous permet de rechercher des lieux ou des points d'intérêt à proximité d'une position donnée.
- `geo:SearchPlaceIndexForText` vous permet de rechercher une adresse, un nom, une ville ou une région à l'aide de texte 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"
    }
  ]
}
```

Politique de lecture seule pour les calculateurs d'itinéraires

Vous pouvez créer une politique en lecture seule pour permettre à un utilisateur d'accéder à une ressource de calcul d'itinéraire pour calculer un itinéraire.

Dans cet exemple, en plus d'accorder l'accès à *ExampleCalculator*, la politique suivante autorise l'opération suivante :

- `geo:CalculateRoute` calcule un itinéraire en fonction d'une position de départ, d'une position de destination et d'une liste de positions de points de cheminement.

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

```

"Statement": [
  {
    "Sid": "RoutesReadOnly",
    "Effect": "Allow",
    "Action": [
      "geo:CalculateRoute"
    ],
    "Resource": "arn:aws:geo:us-west-2:accountID:route-calculator/ExampleCalculator"
  }
]
}

```

Contrôlez l'accès aux ressources en fonction des clés de condition

Lorsque vous créez une politique IAM pour autoriser l'accès à l'utilisation de géofences ou de positions d'appareils, vous pouvez utiliser [les opérateurs de condition](#) pour contrôler plus précisément les géofences ou les appareils auxquels un utilisateur peut accéder. Vous pouvez le faire en incluant l'identifiant de géofence ou l'identifiant de l'appareil dans l'Conditionélément de votre politique.

L'exemple de politique suivant montre comment créer une politique qui permet à un utilisateur de mettre à jour les positions des appareils pour un appareil spécifique.

```

{
  "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"
          ]
        }
      }
    }
  ]
}

```

```
}
```

Contrôlez l'accès aux ressources en fonction des balises

Lorsque vous créez une politique IAM pour autoriser l'accès à vos ressources Amazon Location, vous pouvez utiliser le [contrôle d'accès basé sur les attributs](#) pour mieux contrôler les ressources qu'un utilisateur peut modifier, utiliser ou supprimer. Vous pouvez le faire en incluant les informations relatives aux balises dans l'Conditionnement de votre politique afin de contrôler l'accès en fonction de vos [balises](#) de ressources.

L'exemple de politique suivant montre comment vous pouvez créer une politique permettant à un utilisateur de créer des barrières géographiques. *Cela autorise les actions suivantes pour créer une ou plusieurs géofences sur une collection de géofences appelée `Collection` :*

- `geo:BatchPutGeofence` pour créer plusieurs géofences.
- `geo:PutGeofence` pour créer une géofence unique.

Toutefois, cette politique utilise l'Conditionnement pour accorder l'autorisation uniquement si la balise `Collection` a la valeur du nom d'utilisateur de cet utilisateur. `Owner`

- Par exemple, si un utilisateur nommé `richard-roe` tente de consulter une `collection` de locations Amazon, la `collection` doit être étiquetée `Owner=richard-roe` ou `owner=richard-roe`. Dans le cas contraire, l'accès est refusé à l'utilisateur.

Note

La clé de condition d'étiquette `Owner` correspond à la fois à `Owner` et à `owner`, car les noms de clé de condition ne sont pas sensibles à la casse. Pour plus d'informations, veuillez consulter la rubrique [Éléments de stratégie JSON IAM : Condition](#) dans le Guide de l'utilisateur 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}"}
}
]
```

Pour un didacticiel expliquant [comment définir les autorisations d'accès aux ressources AWS en fonction de balises](#), consultez le guide de AWS Identity and Access Management l'utilisateur.

Résolution des problèmes d'identité et d'accès à Amazon Location Service

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

Rubriques

- [Je ne suis pas autorisé à effectuer une action sur Amazon Location](#)
- [Je ne suis pas autorisé à effectuer iam : PassRole](#)
- [Je souhaite autoriser des personnes extérieures Compte AWS à moi à accéder à mes ressources Amazon Location](#)

Je ne suis pas autorisé à effectuer une action sur Amazon Location

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

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

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

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

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

Je ne suis pas autorisé à effectuer `iam:PassRole`

Si vous recevez un message d'erreur indiquant que vous n'êtes pas autorisé à effectuer l'action `iam:PassRole`, vos politiques doivent être mises à jour pour vous permettre de transmettre un rôle à Amazon Location.

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

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour effectuer une action dans Amazon Location. Toutefois, l'action nécessite que le service ait des autorisations accordées par un rôle de service. Mary ne dispose pas des autorisations nécessaires pour transférer le rôle au service.

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

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

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

Je souhaite autoriser des personnes extérieures à mon compte AWS à accéder à mes ressources Amazon Location

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACL), vous pouvez utiliser ces politiques pour donner l'accès à vos ressources.

Pour en savoir plus, consultez les éléments suivants :

- Pour savoir si Amazon Location prend en charge ces fonctionnalités, consultez [Comment Amazon Location Service fonctionne avec IAM](#).
- Pour savoir comment fournir l'accès à vos ressources sur celles Comptes AWS que vous possédez, consultez la section [Fournir l'accès à un utilisateur IAM dans un autre utilisateur Compte AWS que vous possédez](#) dans le Guide de l'utilisateur IAM.
- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section [Fournir un accès à des ressources Comptes AWS détenues par des tiers](#) dans le guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, consultez [Fournir un accès à des utilisateurs authentifiés en externe \(fédération d'identité\)](#) dans le Guide de l'utilisateur IAM.
- Pour connaître la différence entre l'utilisation de rôles et de politiques basées sur les ressources pour l'accès entre comptes, consultez la section Accès aux [ressources entre comptes dans IAM dans le guide de l'utilisateur d'IAM](#).

Réponse aux incidents dans Amazon Location Service

La sécurité est la priorité absolue chez AWS. Dans le cadre du [modèle de responsabilité partagée](#) du AWS cloud, AWS gère un centre de données et une architecture réseau qui répondent aux exigences des organisations les plus sensibles en matière de sécurité. En tant que AWS client, vous partagez la responsabilité du maintien de la sécurité dans le cloud. Cela signifie que vous contrôlez la sécurité que vous choisissez de mettre en œuvre à partir AWS des outils et fonctionnalités auxquels vous avez accès.

En établissant une base de sécurité répondant aux objectifs de vos applications exécutées dans le cloud, vous êtes en mesure de détecter les écarts auxquels vous pouvez réagir. La réponse aux incidents de sécurité pouvant être un sujet complexe, nous vous encourageons à consulter les ressources suivantes afin de mieux comprendre l'impact de la réponse aux incidents (IR) et de vos choix sur les objectifs de votre entreprise : le [guide de réponse aux incidents de sécurité AWS](#), le livre blanc sur les [meilleures pratiques de sécurité AWS](#) et le [cadre d'adoption du cloud AWS \(AWS CAF\)](#).

Journalisation et surveillance dans Amazon Location Service

La journalisation et la surveillance jouent un rôle important dans la réponse aux incidents. Il vous permet d'établir une base de sécurité pour détecter les écarts que vous pouvez étudier et corriger.

En mettant en œuvre la journalisation et la surveillance pour Amazon Location Service, vous êtes en mesure de garantir la fiabilité, la disponibilité et les performances de vos projets et de vos ressources.

AWS fournit plusieurs outils qui peuvent vous aider à enregistrer et à collecter des données pour la réponse aux incidents :

AWS CloudTrail

Amazon Location Service s'intègre AWS CloudTrail à un service qui fournit un enregistrement des actions effectuées par un utilisateur, un rôle ou un AWS service. Cela inclut les actions depuis la console Amazon Location Service et les appels programmatiques aux opérations de l'API Amazon Location. Ces comptes rendus d'actions sont appelés événements. Pour plus d'informations, consultez la section [Enregistrement et surveillance d'Amazon Location Service avec AWS CloudTrail](#).

Amazon CloudWatch

Vous pouvez utiliser Amazon CloudWatch pour collecter et analyser les statistiques relatives à votre compte Amazon Location Service. Vous pouvez activer les CloudWatch alarmes pour vous avertir si une métrique répond à certaines conditions et a atteint un seuil spécifié. Lorsque vous créez une alarme, CloudWatch envoie une notification à un Amazon Simple Notification Service que vous définissez. Pour plus d'informations, consultez le [Monitoring Amazon Location Service with Amazon CloudWatch](#).

AWS Health Tableaux de bord

À l'aide [AWS Health des tableaux](#) de bord, vous pouvez vérifier l'état du service Amazon Location Service. Vous pouvez également surveiller et consulter les données historiques relatives à tout événement ou problème susceptible d'affecter votre AWS environnement. Pour plus d'informations, consultez le [Guide de l'utilisateur AWS Health](#).

Validation de conformité pour Amazon Location Service

Pour savoir si un [programme Services AWS de conformité Service AWS s'inscrit dans le champ d'application de programmes de conformité](#) spécifiques, consultez Services AWS la section de conformité et sélectionnez le programme de conformité qui vous intéresse. Pour des informations générales, voir Programmes de [AWS conformité Programmes AWS](#) de .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir [Téléchargement de rapports dans AWS Artifact](#) .

Votre responsabilité en matière de conformité lors de l'utilisation Services AWS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. AWS fournit les ressources suivantes pour faciliter la mise en conformité :

- [Guides de démarrage rapide sur la sécurité et la conformité](#) : ces guides de déploiement abordent les considérations architecturales et indiquent les étapes à suivre pour déployer des environnements de base axés sur AWS la sécurité et la conformité.
- [Architecture axée sur la sécurité et la conformité HIPAA sur Amazon Web Services](#) : ce livre blanc décrit comment les entreprises peuvent créer des applications AWS conformes à la loi HIPAA.

 Note

Tous ne Services AWS sont pas éligibles à la loi HIPAA. Pour plus d'informations, consultez le [HIPAA Eligible Services Reference](#).

- AWS Ressources de <https://aws.amazon.com/compliance/resources/> de conformité — Cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- [AWS Guides de conformité destinés aux clients](#) — Comprenez le modèle de responsabilité partagée sous l'angle de la conformité. Les guides résument les meilleures pratiques en matière de sécurisation Services AWS et décrivent les directives relatives aux contrôles de sécurité dans de nombreux cadres (notamment le National Institute of Standards and Technology (NIST), le Payment Card Industry Security Standards Council (PCI) et l'Organisation internationale de normalisation (ISO)).
- [Évaluation des ressources à l'aide des règles](#) du guide du AWS Config développeur : le AWS Config service évalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.
- [AWS Security Hub](#)— Cela Service AWS fournit une vue complète de votre état de sécurité interne AWS. Security Hub utilise des contrôles de sécurité pour évaluer vos ressources AWS et vérifier votre conformité par rapport aux normes et aux bonnes pratiques du secteur de la sécurité. Pour obtenir la liste des services et des contrôles pris en charge, consultez [Référence des contrôles Security Hub](#).
- [Amazon GuardDuty](#) — Cela Service AWS détecte les menaces potentielles qui pèsent sur vos charges de travail Comptes AWS, vos conteneurs et vos données en surveillant votre environnement pour détecter toute activité suspecte et malveillante. GuardDuty peut vous aider

à répondre à diverses exigences de conformité, telles que la norme PCI DSS, en répondant aux exigences de détection des intrusions imposées par certains cadres de conformité.

- [AWS Audit Manager](#)— Cela vous Service AWS permet d'auditer en permanence votre AWS utilisation afin de simplifier la gestion des risques et la conformité aux réglementations et aux normes du secteur.

Résilience dans Amazon Location Service

L'infrastructure AWS mondiale est construite autour Régions AWS de zones de disponibilité. Régions AWS fournissent plusieurs zones de disponibilité physiquement séparées et isolées, connectées par un réseau à faible latence, à haut débit et hautement redondant. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone à l'autre sans interruption. Les zones de disponibilité sont davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l'échelle que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur les zones de disponibilité Régions AWS et les zones de disponibilité, consultez la section [Infrastructure AWS globale](#).

Outre l'infrastructure AWS mondiale, Amazon Location propose plusieurs fonctionnalités pour répondre à vos besoins en matière de résilience et de sauvegarde des données.

Sécurité de l'infrastructure dans Amazon Location Service

En tant que service géré, Amazon Location Service est protégé par la sécurité du réseau AWS mondial. Pour plus d'informations sur les services AWS de sécurité et sur la manière dont AWS l'infrastructure est protégée, consultez la section [Sécurité du AWS cloud](#). Pour concevoir votre AWS environnement en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section [Protection de l'infrastructure](#) dans le cadre AWS bien architecturé du pilier de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder à Amazon Location via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser [AWS Security Token Service](#) (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Analyse de configuration et de vulnérabilité dans Amazon Location

La configuration et les contrôles informatiques sont une responsabilité partagée entre vous AWS et vous, notre client. Pour plus d'informations, consultez le [modèle de responsabilité AWS partagée](#).

Prévention du problème de l'adjoint confus entre services

Le problème de député confus est un problème de sécurité dans lequel une entité qui n'est pas autorisée à effectuer une action peut contraindre une entité plus privilégiée à le faire. En AWS, l'usurpation d'identité interservices peut entraîner la confusion des adjoints. L'usurpation d'identité entre services peut se produire lorsqu'un service (le service appelant) appelle un autre service (le service appelé). Le service appelant peut être manipulé et ses autorisations utilisées pour agir sur les ressources d'un autre client auxquelles on ne serait pas autorisé d'accéder autrement. Pour éviter cela, AWS fournit des outils qui vous aident à protéger vos données pour tous les services avec des principaux de service qui ont eu accès aux ressources de votre compte.

Amazon Location Service n'agit pas en tant que service d'appel en votre nom vers d'autres AWS services. Vous n'avez donc pas besoin d'ajouter ces protections dans ce cas. Pour en savoir plus sur le problème du député confus, voir [Le problème du député confus](#) dans le guide de AWS Identity and Access Management l'utilisateur.

Bonnes pratiques de sécurité pour Amazon Location Service

Amazon Location Service fournit un certain nombre de fonctionnalités de sécurité à prendre en compte lors de l'élaboration et de la mise en œuvre de vos propres politiques de sécurité. Les bonnes pratiques suivantes doivent être considérées comme des instructions générales et ne représentent pas une solution de sécurité complète. Étant donné que ces bonnes pratiques peuvent ne pas être appropriées ou suffisantes pour votre environnement, considérez-les comme des remarques utiles plutôt que comme des recommandations.

Les meilleures pratiques de Detective en matière de sécurité pour Amazon Location Service

Les bonnes pratiques suivantes pour Amazon Location Service peuvent aider à détecter les incidents de sécurité :

Mettre en œuvre les outils de surveillance AWS

La surveillance est essentielle à la réponse aux incidents et garantit la fiabilité et la sécurité des ressources d'Amazon Location Service et de vos solutions. Vous pouvez implémenter des outils de surveillance à partir des différents outils et services disponibles via AWS pour surveiller vos ressources et vos autres services AWS.

Par exemple, Amazon CloudWatch permet de surveiller les métriques pour Amazon Location Service et de configurer des alarmes pour vous avertir si une métrique répond à certaines conditions que vous avez définies et a atteint un seuil que vous avez défini. Lorsque vous créez une alarme, vous pouvez configurer CloudWatch l'envoi d'une notification à l'aide d'Amazon Simple Notification Service. Pour plus d'informations, consultez [the section called “Journalisation et surveillance”](#).

Activer les outils de journalisation AWS

La journalisation fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un service AWS dans Amazon Location Service. Vous pouvez implémenter des outils de journalisation, par exemple AWS CloudTrail pour collecter des données sur les actions visant à détecter les activités inhabituelles des API.

Lorsque vous créez un suivi, vous pouvez le configurer CloudTrail pour consigner les événements. Les événements sont des enregistrements des opérations de ressources effectuées sur ou au sein d'une ressource, telles que la demande envoyée à Amazon Location, l'adresse IP à partir de laquelle la demande a été faite, l'auteur de la demande, le moment où la demande a été faite, ainsi que des données supplémentaires. Pour plus d'informations, consultez la section [Enregistrement des événements liés aux données pour les sentiers](#) dans le guide de AWS CloudTrail l'utilisateur.

Bonnes pratiques de sécurité préventive pour Amazon Location Service

Les bonnes pratiques suivantes pour Amazon Location Service peuvent vous aider à prévenir les incidents de sécurité :

Utilisez des connexions sécurisées

Utilisez toujours des connexions cryptées, telles que celles qui commencent par `https://` pour protéger les informations sensibles pendant le transport.

Implémenter l'accès aux ressources avec le moindre privilège

Lorsque vous créez des politiques personnalisées pour les ressources Amazon Location, accordez uniquement les autorisations requises pour effectuer une tâche. Il est recommandé de commencer par un ensemble minimal d'autorisations et d'accorder des autorisations supplémentaires si nécessaire. La mise en œuvre de l'accès avec le moindre privilège est essentielle pour réduire le risque et l'impact susceptibles de résulter d'erreurs ou d'attaques malveillantes. Pour plus d'informations, consultez [the section called "Gestion de l'identité et des accès"](#).

Utiliser des identifiants uniques au monde comme identifiants d'appareils

Utilisez les conventions suivantes pour les identifiants des appareils.

- Les identifiants des appareils doivent être uniques.
- Les identifiants des appareils ne doivent pas être secrets, car ils peuvent être utilisés comme clés étrangères pour d'autres systèmes.
- Les identifiants d'appareils ne doivent pas contenir d'informations personnelles identifiables (PII), telles que les identifiants des appareils téléphoniques ou les adresses e-mail.
- Les identifiants des appareils ne doivent pas être prévisibles. Les identificateurs opaques tels que les UUID sont recommandés.

Ne pas inclure les informations personnelles dans les propriétés de position de l'appareil

Lorsque vous envoyez des mises à jour d'appareils (par exemple, lors de [DevicePositionUpdate](#) leur utilisation), n'incluez pas d'informations personnellement identifiables (PII) telles que le numéro de téléphone ou l'adresse e-mail dans le `PositionProperties`

Bonnes pratiques pour Amazon Location Service

Cette rubrique présente les meilleures pratiques pour vous aider à utiliser Amazon Location Service. Bien que ces bonnes pratiques puissent vous aider à tirer pleinement parti d'Amazon Location Service, elles ne constituent pas une solution complète. Vous devez suivre uniquement les recommandations applicables à votre environnement.

Rubriques

- [Sécurité](#)
- [Gestion des ressources](#)
- [Facturation et gestion des coûts](#)
- [Quotas et utilisation](#)

Sécurité

Pour aider à gérer ou même à éviter les risques de sécurité, tenez compte des meilleures pratiques suivantes :

- Utilisez la fédération d'identité et les rôles IAM pour gérer, contrôler ou limiter l'accès à vos ressources Amazon Location. Pour plus d'informations, consultez [Bonnes pratiques IAM](#) dans le Guide de l'utilisateur IAM.
- Respectez le principe du moindre privilège pour n'accorder que l'accès minimum requis à vos ressources Amazon Location Service. Pour plus d'informations, consultez [the section called "Gestion des accès à l'aide de politiques"](#).
- Pour les ressources Amazon Location Service utilisées dans les applications Web, limitez l'accès à l'aide d'une condition `aws:ref:iam` IAM, en limitant l'utilisation par des sites autres que ceux inclus dans la liste d'autorisation.
- Utilisez des outils de surveillance et de journalisation pour suivre l'accès aux ressources et leur utilisation. Pour plus d'informations, consultez la section « [Enregistrement the section called "Journalisation et surveillance" des événements liés aux données pour les sentiers](#) » dans le guide de AWS CloudTrail l'utilisateur.
- Utilisez des connexions sécurisées, telles que celles qui visent `https://` à renforcer la sécurité et à protéger les utilisateurs contre les attaques lors de la transmission de données entre le serveur et le navigateur.

Pour plus d'informations sur les meilleures pratiques de sécurité en matière de détection et de prévention, consultez la rubrique sur [the section called "Bonnes pratiques de sécurité"](#).

Gestion des ressources

Pour gérer efficacement vos ressources de localisation dans Amazon Location Service, prenez en compte les meilleures pratiques suivantes :

- Utilisez des points de terminaison régionaux qui sont au cœur de votre base d'utilisateurs attendue afin d'améliorer leur expérience. Pour plus d'informations sur les points de terminaison régionaux, consultez [Amazon Location, régions et points de terminaison](#).
- Pour les ressources qui utilisent des fournisseurs de données, telles que les ressources cartographiques et les ressources d'index de lieux, assurez-vous de respecter les conditions d'utilisation du fournisseur de données concerné. Pour plus d'informations, consultez la section [Fournisseurs de données](#).
- Minimisez la création de ressources en disposant d'une ressource pour chaque configuration de carte, d'index de lieux ou d'itinéraires. Au sein d'une région, vous n'avez généralement besoin que d'une seule ressource par fournisseur de données ou style de carte. La plupart des applications utilisent les ressources existantes et ne créent pas de ressources au moment de l'exécution.
- Lorsque vous utilisez différentes ressources dans une seule application, telles qu'une ressource cartographique et un calculateur d'itinéraire, utilisez le même fournisseur de données pour chaque ressource afin de garantir la concordance des données. Par exemple, une géométrie d'itinéraire que vous créez à l'aide de votre calculateur d'itinéraire s'aligne sur les rues de la carte dessinée à l'aide de la ressource cartographique.

Facturation et gestion des coûts

Pour vous aider à gérer vos coûts et votre facturation, prenez en compte les meilleures pratiques suivantes :

- Utilisez des outils de surveillance, tels qu'Amazon CloudWatch, pour suivre l'utilisation de vos ressources. Vous pouvez définir des alertes qui vous avertissent lorsque l'utilisation est sur le point de dépasser les limites spécifiées. Pour plus d'informations, consultez la section [Création d'une alarme de facturation pour surveiller vos frais AWS estimés](#) dans le guide de CloudWatch l'utilisateur Amazon.

Quotas et utilisation

Vous Compte AWS incluez des quotas qui fixent une limite par défaut à votre consommation. Vous pouvez configurer des alarmes pour vous avertir lorsque votre consommation se rapproche de votre limite, et vous pouvez demander une augmentation du quota lorsque vous en avez besoin. Pour plus d'informations sur l'utilisation des quotas, consultez les rubriques suivantes.

- [Quotas d'Amazon Location Service](#)

- [Utilisation CloudWatch pour surveiller l'utilisation par rapport aux quotas](#)
- [Visualisez vos quotas de service et définissez des alarmes](#) dans le guide de l' CloudWatch utilisateur Amazon.

Vous pouvez créer des alarmes pour vous avertir à l'avance lorsque vous êtes sur le point de dépasser vos limites. Nous vous recommandons de définir des alarmes pour chaque quota dans chaque Région AWS endroit où vous utilisez Amazon Location. Par exemple, vous pouvez surveiller votre utilisation de l'`SearchPlaceIndexForText` opération et créer une alarme lorsque vous dépassez 80 % de votre quota actuel.

Lorsque vous recevez un avertissement concernant votre quota, vous devez décider de la marche à suivre. Il se peut que vous utilisiez des ressources supplémentaires en raison de l'augmentation de votre clientèle. Dans ce cas, vous souhaitez peut-être demander une augmentation de votre quota, par exemple une augmentation de 50 % du quota pour un appel d'API dans cette région. Ou peut-être qu'une erreur de votre service vous oblige à passer d'autres appels inutiles à Amazon Location. Dans ce cas, vous souhaitez résoudre le problème dans votre service.

Historique du document

Le tableau suivant décrit la documentation d'Amazon Location Service. Pour être informé des mises à jour, vous pouvez vous abonner à un flux RSS.

Modification	Description	Date
Amazon Location Service publie un nouveau SDK pour JavaScript	Pour faciliter le développement d'applications Amazon Location depuis des interfaces Web, Amazon Location ajoute un nouveau SDK open source qui prend en charge le SDK AWS pour JavaScript v3, simplifie l'authentification et utilise GeoJSON. Pour plus d'informations, consultez le SDK Amazon Location .	6 juillet 2023
Amazon Location Service publie des clés d'API pour une disponibilité générale	Amazon Location ajoute la prise en charge du lieu et de l'itinéraire et annonce la disponibilité générale de la fonctionnalité des clés d'API. Pour plus d'informations, consultez la section Utilisation des clés d'API .	6 juillet 2023
Amazon Location Service ajoute des EventBridge événements Amazon pour les mises à jour de position	Amazon Location ajoute la prise en charge de l'envoi d'événements de mise à jour de la position du tracker à EventBridge. Pour plus d'informations, notamment sur la manière d'activer les événements pour un outil de suivi, voir Réagir aux	6 juillet 2023

Amazon Location ajoute des événements avec EventBridge.		
Amazon Location ajoute des métadonnées aux géofences	À l'aide de l'API Amazon Location, vous pouvez désormais ajouter des propriétés de métadonnées à vos géofences. Ils sont stockés avec votre géofence et inclus dans les événements liés à la géofence sur Amazon. EventBridge Pour plus d'informations, voir Dessiner des géofences et Créer des règles d'événement .	15 juin 2023
Amazon Location ajoute des catégories pour les lieux	Amazon Location ajoute des catégories dans les résultats de recherche sur place et filtre les résultats par catégorie . Pour plus d'informations, consultez la section Catégories et filtrage .	15 juin 2023
Amazon Location présente des opinions politiques	Amazon Location ajoute des points de vue politiques à certains styles de carte. Pour plus d'informations, voir Opinions politiques .	23 mai 2023

[Amazon Location présente un nouveau site de démonstration et d'échantillons](#)

Amazon Location annonce la création d'un nouveau site Web qui vous donne accès à des démonstrations et à des exemples d'Amazon Location. Pour plus d'informations, consultez le [site de démonstration d'Amazon Location](#).

3 mai 2023

[Amazon Location introduit des itinéraires plus longs dans CalculateRouteMatrix](#)

Amazon Location autorise désormais les itinéraires de longueur illimitée pour les itinéraires matriciels créés avec le fournisseur de données HERE. Pour plus d'informations, consultez la section [Planification d'itinéraires à plus long terme](#).

24 avril 2023

[La documentation Amazon Location ajoute des différences entre les fonctionnalités selon le fournisseur de données](#)

La documentation Amazon Location a été mise à jour avec des informations concernant les différences entre chaque fournisseur de données dans Maps, Places search et routage. Pour plus d'informations, voir [Fonctionnalités par fournisseur de données](#).

30 mars 2023

[Disponibilité générale des cartes Amazon Location Open Data](#)

Disponibilité générale du fournisseur de données et du style d'Amazon Location Service, sur la OpenStreetMap base des cartes Daylight. Pour plus d'informations, consultez [Open Data](#).

7 mars 2023

Amazon Location ajoute une nouvelle méthode d'autorisation en version préliminaire	Amazon Location Service ajoute des clés d'API en tant que nouvelle méthode d'autorisation pour les utilisateurs anonymes, en mode aperçu. Pour plus d'informations, consultez Autoriser l'accès invité non authentifié à votre application à l'aide de clés d'API .	23 février 2023
La documentation Amazon Location a été mise à jour avec les dernières bonnes pratiques en matière d'IAM	La documentation d'Amazon Location Service a été mise à jour afin de respecter les AWS Identity and Access Management meilleures pratiques les plus récentes. Pour plus d'informations, consultez la section Sécurité dans Amazon Location Service .	26 janvier 2023
Amazon Location Service s'ajoute GrabMaps en tant que fournisseur de données en Asie du Sud-Est	Amazon Location se présente GrabMaps en tant que fournisseur de données en Asie du Sud-Est. Pour plus d'informations, consultez GrabMaps .	10 janvier 2023
Cartes de données ouvertes d'Amazon Location Service en avant-première	Nouveau fournisseur et style de données de localisation Amazon ajoutés en version préliminaire publique, sur la OpenStreetMap base des cartes Daylight. Pour plus d'informations, voir Open Data (version préliminaire) .	15 décembre 2022

[Nouveaux styles d'imagerie satellite HERE](#)

Deux nouveaux styles de carte ont été ajoutés pour les cartes utilisant HERE comme fournisseur de données, HERE Satellite Imagery et HERE Hybrid. Pour plus d'informations, consultez la section [Styles de carte HERE](#).

25 octobre 2022

[Unités dans les adresses](#)

Amazon Location Service prend désormais en charge les unités dans les adresses, par exemple « 123 Main St, Apartment 3B, Anytown, États-Unis ».

20 septembre 2022

[Obtenir des places par identifiant](#)

Amazon Location Service inclut désormais une assistance pour trouver l'emplacement exact suggéré par l'opération `SearchPlaceIndexForSuggestions` utilisant l'opération `GetPlace`. Consultez la section [Utilisation de la saisie semi-automatique](#).

20 septembre 2022

[Clés de condition supplémentaires pour la politique IAM](#)

Amazon Location Service prend désormais en charge des clés de condition supplémentaires qui vous permettent de définir l'accès à des barrières géographiques ou à des appareils spécifiques dans le cadre de la politique IAM. Voir [Clés de condition](#).

23 août 2022

Géofences circulaires	Amazon Location Service prend désormais en charge les géofences définies comme un cercle avec un point central et un rayon, afin de détecter les événements lorsque des appareils se trouvent à une certaine distance d'un lieu. Voir Ajouter des géofences circulaires .	11 août 2022
Référence d'API combinée	Amazon Location Service dispose désormais d'un guide de référence d'API unique, plutôt que de guides distincts pour chaque sous-service. Pour plus d'informations sur les API, consultez Amazon Location APIs .	7 juillet 2022
Intégration des Quotas de Service	Amazon Location est désormais intégré à Service Quotas , ce qui vous permet de consulter et de gérer vos quotas via AWS Management Console ou à l'aide du AWS CLI.	6 juillet 2022
Chapitre de documentation sur les concepts mis à jour	Le chapitre sur les concepts d'Amazon Location a été mis à jour avec plus d'informations destinées aux utilisateurs d'Amazon Location.	22 avril 2022

Nouveau didacticiel de démarrage rapide pour Android	Un nouveau didacticiel de démarrage rapide pour le développement Android à l'aide de Kotlin a été ajouté pour permettre aux développeurs d'être rapidement opérationnels.	15 avril 2022
Nouveaux styles de carte HERE	Deux nouveaux styles de carte ont été ajoutés pour les cartes utilisant HERE comme fournisseur de données. Pour plus d'informations, consultez la section Styles de carte HERE .	15 mars 2022
Restructurez la documentation avec des exemples de code et des didacticiels ajoutés	Ce guide du développeur a été restructuré pour trouver plus facilement des sujets, notamment de nouveaux chapitres sur le démarrage rapide et les exemples de code .	25 février 2022
Filtrage de position basé sur la précision pour les trackers	Vous pouvez désormais utiliser des filtres basés sur la précision lors de la création d'une ressource de suivi.	7 décembre 2021
Complétion automatique pour les index de lieux	Vous pouvez désormais utiliser la saisie semi-automatique lorsque vous recherchez des index de lieux.	6 décembre 2021

[Nouveau didacticiel Amplify pour l'utilisation de cartes](#)

Un nouveau didacticiel est disponible et montre comment AWS Amplify afficher des cartes dans une application Web. Le didacticiel est disponible sur [Utilisation de la bibliothèque Amplify avec Amazon Location Service.](#)

24 novembre 2021

[Placer des extensions de requête](#)

Amazon Location Service prend désormais en charge la définition d'une langue préférée pour les résultats lors du géocodage, ou du géocodage inversé, et ajoute le fuseau horaire et d'autres informations aux résultats. Pour plus d'informations sur le géocodage et le géocodage inversé, consultez les rubriques [Géocodage](#), [géocodage inversé et recherche.](#)

16 novembre 2021

[Filtrage de la position du tracker](#)

Amazon Location Service ajoute une nouvelle fonctionnalité de filtrage des positions aux dispositifs de suivi afin de vous aider à contrôler les coûts. Cette fonctionnalité filtre certaines mises à jour de position sur les appareils avant que les mises à jour ne soient stockées ou évaluées par rapport aux géofences. Pour plus d'informations sur le filtrage de position, consultez la section [Trackers](#).

5 octobre 2021

[opérations de mise à jour](#)

Les opérations suivantes ont été ajoutées aux références d'API Amazon Location Service : [UpdateMap](#), [UpdatePlaceIndex](#), [UpdateRouteCalculator](#), [UpdateGeofenceCollection](#), et [UpdateTracker](#).

19 juillet 2021

[Mise à jour du didacticiel : fonctions Amazon Aurora PostgreSQL définies par l'utilisateur](#)

Un nouveau didacticiel a été ajouté pour expliquer comment utiliser les [fonctions Amazon Aurora PostgreSQL définies par l'utilisateur avec Amazon Location](#) pour valider, nettoyer et enrichir les données géospatiales.

19 juillet 2021

[CloudFormation Ressources AWS](#)

Amazon Location prend désormais en charge [la création des types de ressources suivants dans CloudFormation les ressources AWS](#) :

```
AWS::Location::Map
  AWS::Location::PlaceIndex
  AWS::Location::RouteCalculator
  ,AWS::Location::Tracker
  ,,AWS::Location::TrackerConsumer
  ,etAWS::Location::GeofenceCollection .
```

7 juin 2021

[Balisage des ressources](#)

Vous pouvez désormais [ajouter des balises à vos ressources Amazon Location](#) pour faciliter la gestion, l'identification, l'organisation, la recherche et le filtrage de vos ressources.

1er juin 2021

[Disponibilité générale](#)

Version de disponibilité générale de la documentation destinée aux développeurs d'Amazon Location Service : [région, points de terminaison et quotas de service](#) mis à jour.

1er juin 2021

Imagerie Esri	Amazon Location prend désormais en charge l'utilisation du style de carte Esri : Esri Imagery . Pour plus d'informations, consultez Esri World Imagery sur le site Web d'Esri.	1er juin 2021
Calcul des itinéraires	Vous pouvez désormais utiliser les calculateurs d'itinéraires Amazon Location pour calculer les itinéraires et estimer le temps de trajet en fonction du réseau up-to-date routier et des informations de trafic en temps réel fournies par le fournisseur de données que vous avez choisi.	1er juin 2021
AWS KMSChiffrement par clé géré par le client pour les données au repos	Amazon Location prend désormais en charge l'utilisation d'une clé symétrique gérée par le client que vous créez, détenez et gérez pour ajouter une deuxième couche de chiffrement par rapport au chiffrement que vous AWS possédez déjà .	1er juin 2021
Version préliminaire publique	Publication initiale de la documentation préliminaire publique.	16 décembre 2020
Mise à jour du didacticiel : affichage de cartes	Les didacticiels d'affichage de cartes MapLibre pour Android et iOS ont été mis à jour pour utiliser le SDK MapLibre natif.	17 mars 2020

Glossaire AWS

Pour connaître la terminologie la plus récente d'AWS, consultez le [Glossaire AWS](#) dans la Référence Glossaire AWS.

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.