Service Name

Amazon Location Service

Service Overview

Amazon Location Service is a fully-managed location data platform that provides developers with geospatial capabilities including maps, places, routes, trackers, and geofences. It enables applications to incorporate location functionality using high-quality data from providers like Esri, HERE, and GrabMaps. The service offers pay-as-you-go pricing with no upfront commitments, provides data privacy and security through anonymization and encryption, and integrates seamlessly with other AWS services for comprehensive location-based solutions.

Key Use Cases

- Asset tracking and fleet management: Monitor vehicles, equipment, and personnel in real-time with geofencing capabilities for operational optimization
- Location-based marketing and customer engagement: Deliver personalized recommendations, proximity-based offers, and location-aware notifications to enhance user experience
- Delivery and logistics optimization: Calculate optimal routes, estimate travel times, and provide real-time tracking for enhanced supply chain visibility and customer satisfaction

Features

- **Maps**: Provides dynamic and static map visualization with multiple pre-designed styles, high-quality base map data, and support for embedding in applications
- **Places**: Offers forward and reverse geocoding, point-of-interest search, address validation, and autocomplete functionality with detailed business information
- **Routes**: Calculates optimal travel routes, estimates travel times, supports multi-point optimization, route matrices, and isoline calculations using real-time traffic data
- **Trackers**: Stores and retrieves current and historical location data for devices with filtering capabilities to reduce storage costs and visual noise
- **Geofences**: Detects when tracked devices enter or exit defined geographical boundaries and triggers automated actions through Amazon EventBridge integration

Value Proposition

Amazon Location Service offers superior data quality from leading providers like Esri, HERE, and GrabMaps, while maintaining full data ownership and privacy through anonymization.

The service provides seamless AWS integration with CloudFormation, CloudWatch, EventBridge, and IAM, enabling faster development and deployment:

- Companies like Halodoc and Geo.me have reduced geospatial costs by 88% and 90% respectively
- The pay-as-you-go model eliminates upfront commitments, and the fully-managed APIs reduce operational overhead while accelerating time-to-market for location-enabled applications

Service Integration

Amazon Location Service integrates natively with core AWS services including Amazon EventBridge for geofence event processing, AWS CloudFormation for infrastructure deployment, Amazon CloudWatch for monitoring and metrics, AWS CloudTrail for audit logging, AWS IAM for access control, and AWS KMS for encryption. The service also works with Amazon DynamoDB for data storage, Amazon S3 for backup storage, and supports AWS SDK integration across multiple programming languages. These integrations enable comprehensive monitoring, security compliance, and automated workflow orchestration.

Onboarding and Offboarding

Customers begin by setting up an AWS account and configuring authentication through API Keys, Amazon Cognito, or AWS IAM. The getting started process involves creating location resources like maps, place indexes, or trackers through the AWS Management Console or APIs:

- Amazon Location provides comprehensive developer tools including standard AWS SDKs, front-end mobile and web SDKs, sample applications, and solution guides
- The interactive Amazon Location Service Console offers visual learning tools, and detailed tutorials guide users through creating their first location-enabled applications using MapLibre GL JS

Getting Started Guide:
https://docs.aws.amazon.com/location/latest/developerguide/getting-started.html

Data Removal

Amazon Location Service automatically deletes tracker position data after 30 days of retention. Geofence geometries must be explicitly deleted by users. When offboarding, customers can delete resources through the AWS Management Console, APIs, or CLI. Deleting an AWS account removes all associated Amazon Location resources and data. The service provides APIs like BatchDeleteDevicePositionHistory for bulk data deletion, ensuring complete data removal during the offboarding process.

Backup/Restore and Disaster Recovery

Amazon Location Service operates as a fully-managed service with built-in high availability across multiple Availability Zones. The service maintains a 99.95% availability SLA with automatic failover capabilities. However, Amazon Location does not provide traditional backup and restore capabilities as it primarily processes real-time location requests rather than storing persistent application data, except for tracker positions which are automatically managed with 30-day retention.

Backup Capabilities

Amazon Location Service does not integrate with AWS Backup as it is primarily a request-response service for location operations. The service automatically manages tracker position data with built-in 30-day retention and deletion policies. For application-specific location data backup needs, customers should implement backup strategies using other AWS services like Amazon S3 or DynamoDB for storing processed location results.

Disaster Recovery

Amazon Location Service does not integrate with AWS Elastic Disaster Recovery as it is a fully-managed, stateless service with built-in multi-AZ redundancy. The service provides automatic failover and high availability through AWS's global infrastructure. For disaster recovery planning, customers should focus on backing up their application configurations, API keys, and any stored location processing results using appropriate AWS storage services.

Recovery Objectives

Amazon Location Service achieves low RPO and RTO through its fully-managed architecture with automatic failover across Availability Zones. The service's 99.95% availability SLA supports sub-minute recovery objectives for location requests. For applications requiring specific RPO/RTO targets, customers should implement caching strategies, configure appropriate retry logic, and design multi-region architectures using Amazon Location's global endpoint availability.

Service Constraints

Service Quotas: https://docs.aws.amazon.com/location/latest/developerguide/manage-quotas.html

FAQ: https://aws.amazon.com/location/faqs/

Technical Requirements

Service Documentation: https://docs.aws.amazon.com/location/latest/developerguide/index.html

User Guide: https://docs.aws.amazon.com/location/latest/developerguide/what-is.html

API Reference:
https://docs.aws.amazon.com/location/latest/APIReference/Welcome.html

Pricing Overview

Please see the AWS UK G Cloud 15 Pricing Document accompanying this service in the Digital Marketplace.

Public Pricing: https://aws.amazon.com/location/pricing/

Cost Optimization

Amazon Location Service offers multiple cost optimization strategies through tiered pricing buckets (Core, Advanced, Premium) allowing customers to select appropriate feature sets. The service provides filtering options for tracker positions to reduce storage costs by eliminating unnecessary location updates based on distance, time, or accuracy thresholds:

- Customers can optimize Maps API costs by using static maps for non-interactive scenarios, caching map tiles, and selecting appropriate data providers
- The Places API offers Label pricing for basic address text needs, reducing costs for simple geocoding operations

Savings Considerations

Key cost savings opportunities include leveraging the three-month free tier for evaluation and development, using batch APIs like BatchUpdateDevicePosition and BatchGetDevicePosition to reduce per-request costs, implementing local filtering for tracker updates to minimize stored positions, and selecting the most cost-effective pricing bucket for each use case. Volume discounts apply for monthly usage exceeding $5,000. Customers can achieve significant savings by optimizing API request frequency, using appropriate caching strategies, and selecting regional endpoints closest to their user base to reduce latency and associated costs.