Architecture Diagrams

# Location Services with Real-Time ML Forecasting

# Location Services with Real-Time ML Forecasting: Architecture Diagrams

# Table of Contents

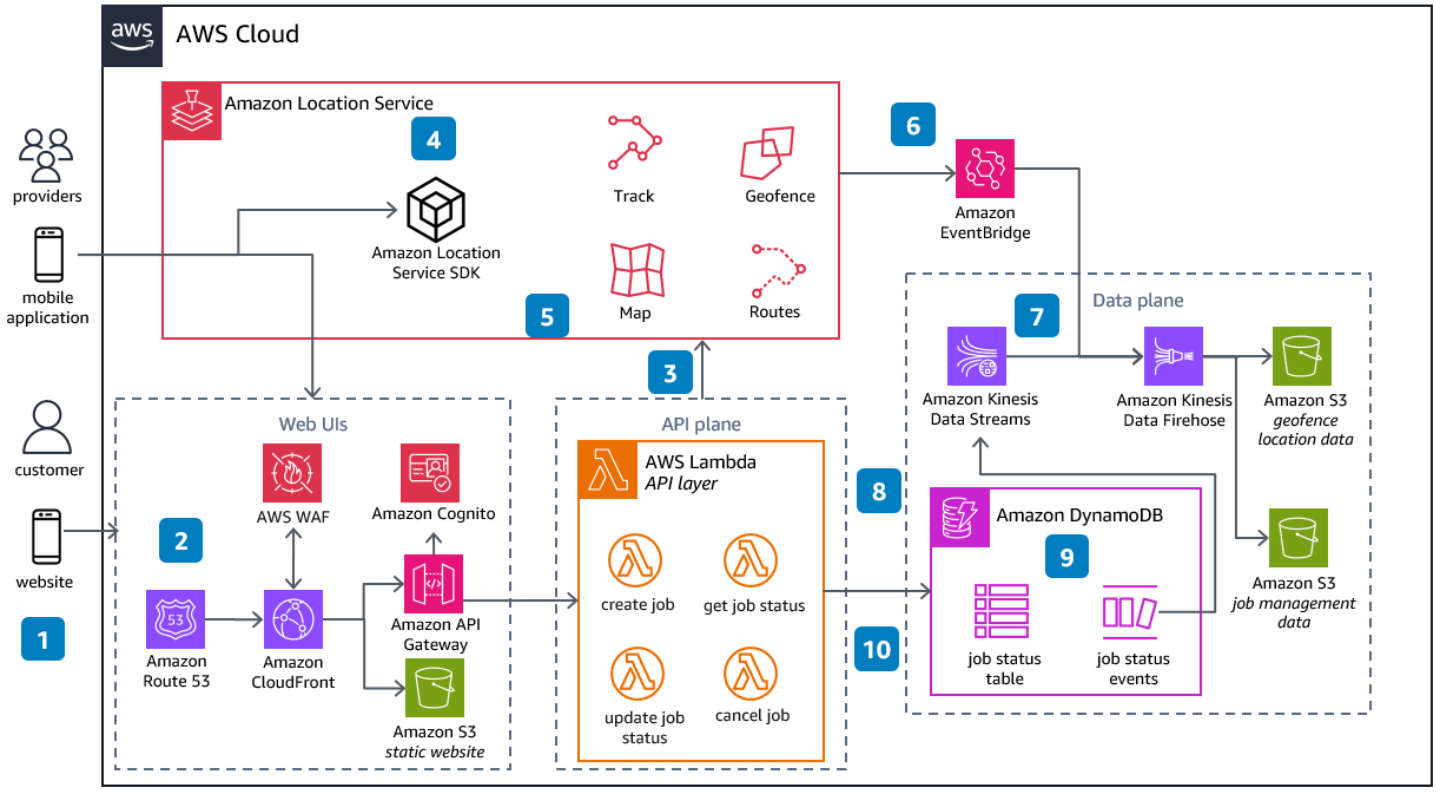
# Location Services with Real-Time ML Forecasting

Publication date: **September 19, 2023 (Diagram history)**

This reference architecture provides guidance on creating a provider-customer location-based platform to create a real-time inference pipeline to forecast demand.

## Sample Web Application Diagram

This reference architecture provides guidance on creating a provider-customer location-based platform (in this case, a taxi hailing service) using Amazon Location Service, Amazon Kinesis, and Amazon SageMaker AI to create a real-time inference pipeline to forecast demand.

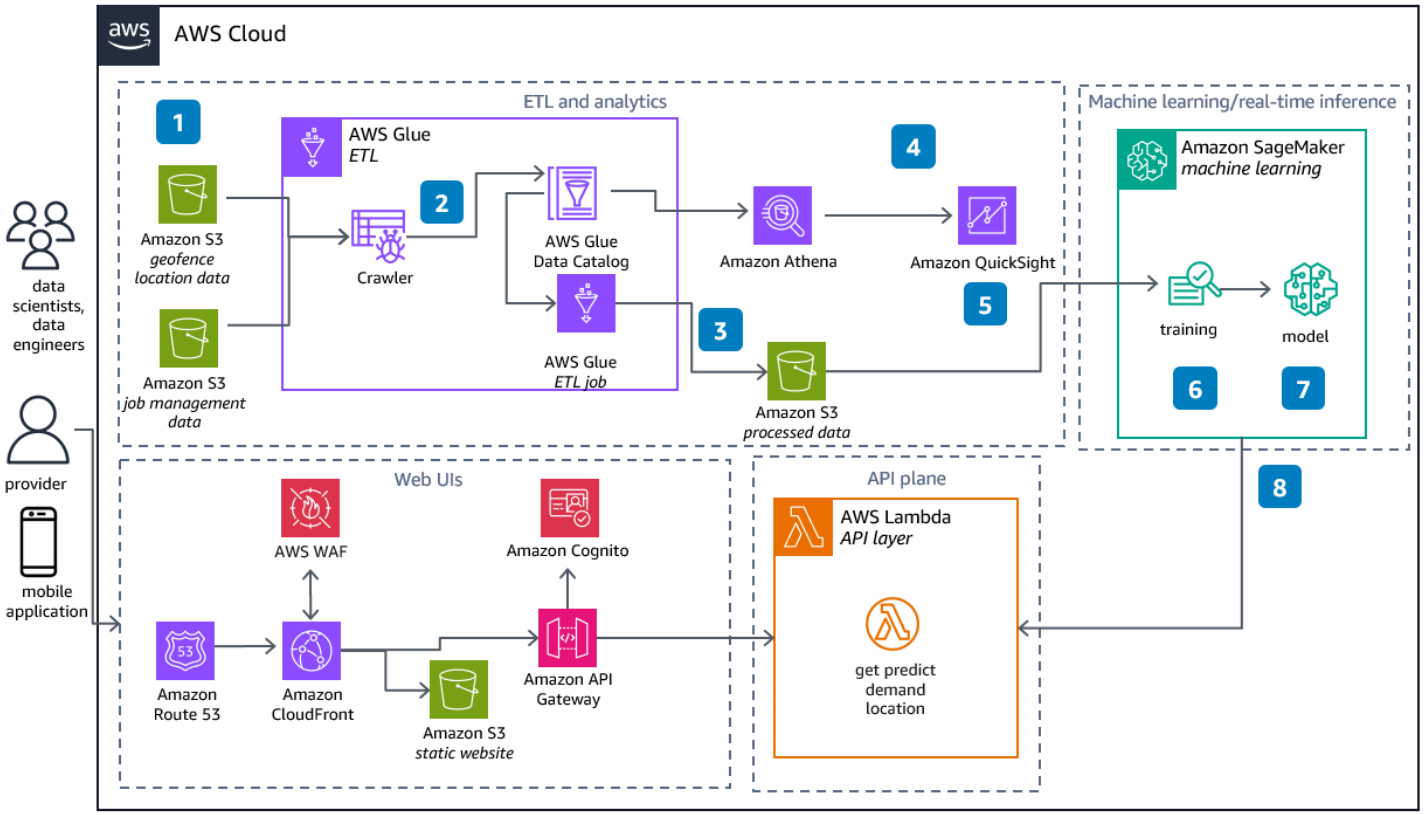



1. A customer hails a taxi by using the website and mobile app, which are hosted in an **Amazon Simple Storage Service** (Amazon S3) bucket.
2. The website is served using an **Amazon CloudFront** distribution, protected by **AWS WAF** with authentication provided by **Amazon Cognito**.
3. Requests are made to an **AWS Lambda** backed API, allowing the user to manage the request. An **Amazon Location Service** geofence is set around the customer using their location data.

4. Providers (such as taxi drivers) use a mobile application that uses the **Amazon Location Service** SDK to broadcast their location, and set statuses (available, offline, busy).
5. Using the **Amazon Location Service** route calculator, the nearest taxi driver to the user is found. A route is plotted between user and driver and shown to the customer on an **Amazon Location Service** map.
6. As the taxi driver nears the user's location and enters the user's geofence, an **Amazon EventBridge** rule reacts and sends a SMS or notification alert to the user.
7. Events such as location tracking and geofence creation are persisted in an **Amazon S3** bucket through **Amazon Data Firehose** for use in operational monitoring and analytics.
8. The API stores states into **Amazon DynamoDB** tables and job update activity generates **DynamoDB Streams** events.
9. **Amazon Kinesis Data Streams** captures these events for real-time analytics.
10. Job data in **DynamoDB** is also streamed into an **Amazon S3** bucket for persistence and analytics.

## Real-Time ML Forecasting Diagram

Using the data collected from the sample web application, AWS Glue, Amazon Athena, and Amazon Quick transform and analyze the data for business analytics and data exploration. This data is also used as part of a full-machine learning (ML) lifecycle with Amazon SageMaker AI.

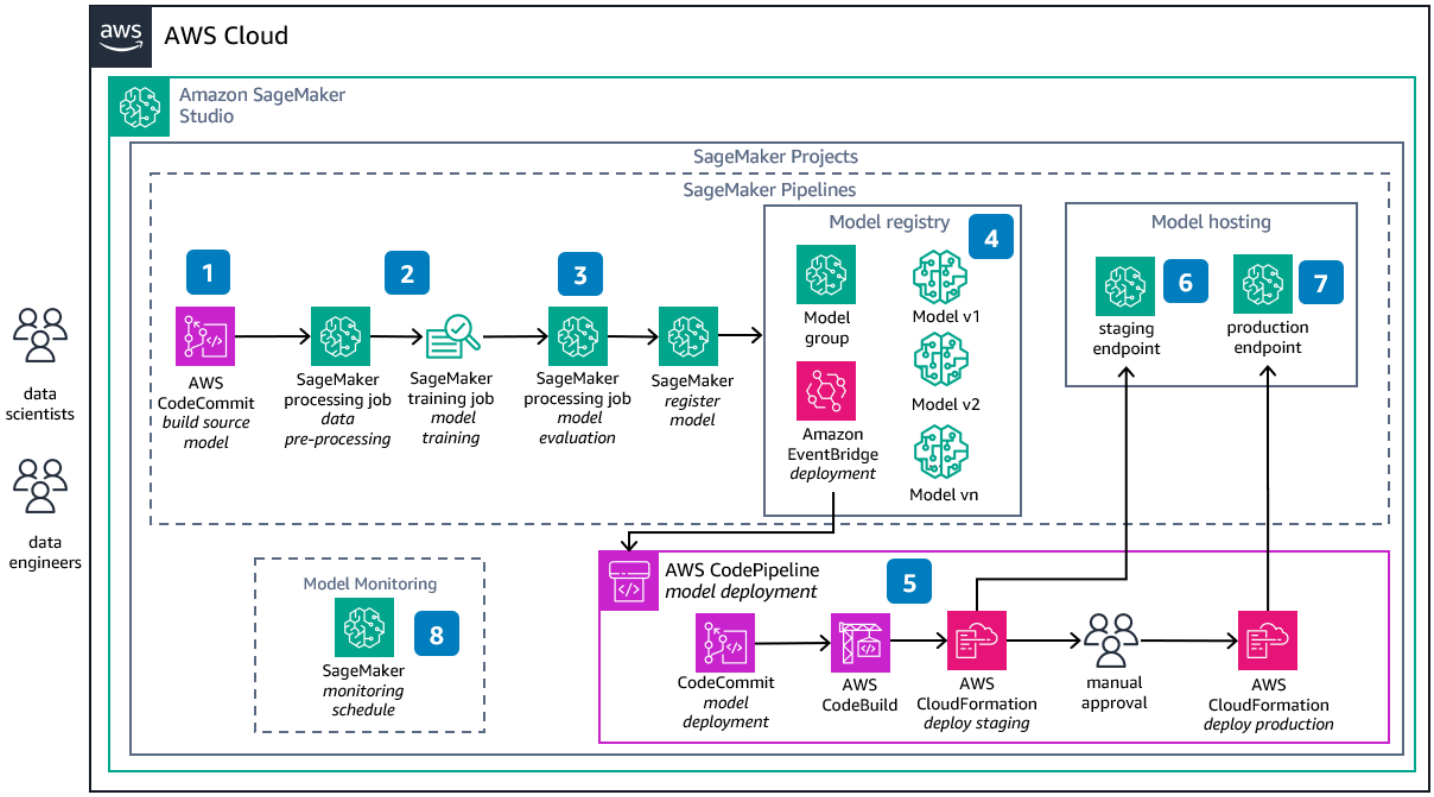



1. An **AWS Glue** workflow is started with an **Amazon EventBridge** event when new objects are put into the geofence location and job management **Amazon S3** buckets.
2. An **AWS Glue** crawler analyzes and categorizes data and infers schema to populate an **AWS Glue Data Catalog**. The **AWS Glue** crawler incrementally crawls the data, but only processes data in the **Amazon S3** buckets that were added since the last crawl.
3. An **AWS Glue** ETL job is run periodically to convert the source data to a Parquet columnar data format. The data is sent to a processed data **Amazon S3** bucket and uses the **Data Catalog** and a crawler for metadata.
4. Data scientists and engineers use **Athena** to query the processed data with SQL statements as part of the data exploration stage of a machine learning lifecycle.
5. **Quick** provides visualization for the processed data to show operational location-based insights. Data for **Quick** is refreshed incrementality within a look-back window of time as new data arrives.
6. Data from the processed data **Amazon S3** bucket is used to train a spatial-temporal machine learning model in **Amazon SageMaker AI**.
7. A **SageMaker AI** serverless inference endpoint is used to deploy the trained model. The serverless endpoint scales with prediction traffic changes.

8. A predict demand location **Lambda** function provides real-time information to the provider's mobile application by using **Amazon API Gateway**, ensuring they can be in the right place at the right time to serve customers.

## Full ML Lifecycle with a Spatial Temporal Model Diagram

The accuracy of the spatial-temporal machine learning model used for real-time forecasting might decrease over time as new trends form (for example, increased demand in a new location or time period). This diagram outlines best practices around MLOps using Amazon SageMaker AI Studio. Using an Amazon SageMaker AI Projects project, you can automatically create model building and model deployment pipelines, experiments, model groups, endpoints, and repositories.




1. An **Amazon SageMaker AI Pipelines** pipeline is started when a member of a data science team or data engineering team commits new code to an **AWS CodeCommit** repository to build the model.
2. The ML model begins retraining, including starting a data pre-processing processing job to create features from the data cleaned by the team. The model is then trained on this data.
3. After the model is trained, the model is evaluated against metrics such as Weighted Mean Absolute Percentage Error (WMAPE) by using a processing job.

4. If model evaluation is successful, it is registered with **Amazon SageMaker AI Model Registry** under a model group. As a model is continuously retrained, subsequent versions appear under the model group.
5. **AWS CodePipeline** is initiated when the model is successfully registered and versioned. This starts the model deployment process with a **CodeCommit** repository as a source.
6. After the model deploys, artifacts are built, the updated model is deployed to a staging serverless endpoint, and then a team of data scientists can test the model's inference.
7. Data scientists use **CodePipeline** to manually approve the deployment after testing and validation. The model is then promoted and deployed to the production serverless endpoint. **Amazon SageMaker AI Serverless Inference** allows deployment of the model without selecting instance types or creating scaling policies.
8. Using an **Amazon SageMaker AI Model Monitor**, you can continuously measure the model for prediction performance; **Model Monitor** also allows for model bias, model explainability, and data quality monitoring.

## Download editable diagram

To customize this reference architecture diagram based on your business needs, download the ZIP file which contains an editable PowerPoint.

## Create a free AWS account

Sign up now

Sign up for an AWS account. New accounts include 12 months of AWS Free Tier access, including the use of Amazon EC2, Amazon S3, and Amazon DynamoDB.

## Further reading

For additional information, refer to

- AWS Architecture Icons
- AWS Architecture Center
- AWS Well-Architected

# Contributors

Contributors to this reference architecture diagram include:

- Partha Dey, Solutions Architect, Amazon Web Services
- Adam Temple, Senior Solutions Architect, Amazon Web Services

# Diagram history

To be notified about updates to this reference architecture diagram, subscribe to the RSS feed.

| Change | Description | Date |
| --- | --- | --- |
| Initial publication | Reference architecture diagram first published. | September 19, 2023 |

**Note**

To subscribe to RSS updates, you must have an RSS plugin enabled for the browser you are using.