**Sample Scenario**

*ACME company is a food delivery business operating in the United Kingdom. Recently, ACME’s customers have complained that certain food products are always out of stock in their area, leading to a poor customer experience.*

*ACME company want your team to develop a system that is able to forecast where certain products will be required in the country so they can proactively ship stock there.*

**Key Stakeholders**

**VP of Engineering**

The end user of this system will be the regional managers who will like to know what products their users are likely to purchase in the next 4 weeks. We would ideally like to expose this forecasting system as an API that integrates into our Enterprise Resource Planning software that is used day to day to organise company resources.

We have expertise in house to be able to tune and deploy our own models with a strong data engineering and data science function. We expect the system to be highly available, resilient as it will deal with about 50 requests per day from each of the 40 area managers around the country.

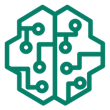
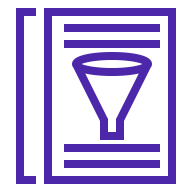
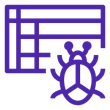
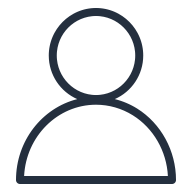
**Head of Data Science**

The data science team would be maintaining the system day to day. We are interested in ensuring data is catalogued correctly, easy to query to conduct exploratory analysis and should be integrated seamlessly into an ML training workflow that is easy to use for the team.

We are most comfortable with Python, it’s our language of choice, and we would like to understand how this model would be deployed using AWS.

**Chief Financial Officer**

This system needs to be elastic so costs should be next to nothing when it’s not being used. We are very keen to understand how we will monitor these costs and ensure that the data science team don’t exceed budgets aligned to them.



AWS Cloud

Area Managers



Data Scientists & Data Engineers

ETL and Analytics



Geofence Location Data



Job Management Data



Processed Data

AWS Glue - ETL

Crawler

AWS Glue  
Data Catalog

Amazon Athena

Amazon QuickSight

Machine Learning / Real-Time Inference

Amazon SageMaker – Machine Learning

Training

Model



Amazon Route 53



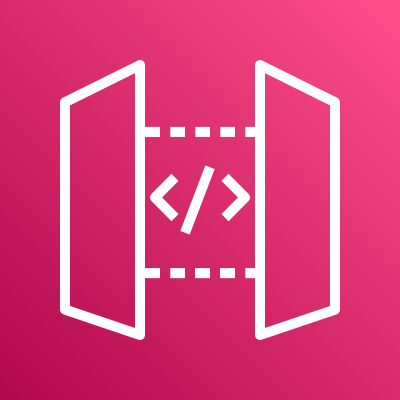
Amazon CloudFront



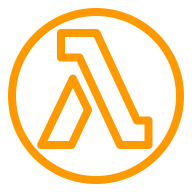
AWS WAF



Amazon Cognito



Amazon API Gateway



Get Predict Demand Location

AWS Lambda – API Layer



Amazon S3 Static Website

Web UIs

API Plane

Using the data collected using from the food delivery application, **AWS Glue, Amazon Athena**, and **Amazon QuickSight** will be used to transform and analyze the data for business analytics and for data exploration as part of a full-ML lifecycle with **Amazon SageMaker**.

* An **AWS Glue** workflow is started with an **Amazon EventBridge** event, in this case, when new objects are put into the Geofence Location and Job Management **Amazon S3** buckets
* An **AWS Glue** crawler is used to analyze and categorize data in the Geofence Location and Job Management **Amazon S3** buckets and infer schema to populate an **AWS Glue** Data Catalog; the AWS Glue crawler runs incremental crawls, it only processes folders in the **Amazon S3** buckets that were added since the last crawl
* The **AWS Glue** Data Catalog contains references to data that is used as sources and targets of extract, transform, and load (ETL) jobs in **AWS Glue;** the data, converted to Parquet format, is sent to a Processed Data **Amazon S3** bucket
* **Amazon Athena** uses SQL queries to query the Processed Data **Amazon S3** bucket, useful as part of a data exploration stage of a ML lifecycle
* Outputs from **Amazon Athena** queries are also used in **Amazon QuickSight** to analyze operational data of this location based service; data for **Amazon QuickSight** is refreshed incrementality within a look-back window of time
* Data from the Processed Data **Amazon S3** bucket is used to train a spatial-temporal machine learning (ML) model in **Amazon SageMaker**
* An **Amazon SageMaker** Serverless Inference endpoint is used to deploy the model trained in Stage 6, serverless endpoints automatically launch compute resources and scale them in and out depending on traffic
* The Predict Demand Location **AWS Lambda** function provides real time information to the taxi driver’s mobile application (via **Amazon** **API Gateway**) ensuring they can be in the right place at the right time to serve customers

****

AWS Cloud



Data Scientists

SageMaker Project

SageMaker Studio

SageMaker Pipelines

Model Registry

Model V1

Model V2

Model Vn

…

Source

Processing Job (Data Pre-Processing)

Training Job (Model Training)

Processing Job (Model Evaluation)

Register Model

Model Group

Model Hosting

Model Deployment



Manual Approval

Staging Endpoint

Production Endpoint

Model Monitoring

Trigger Deploy

Monitoring Schedule

Source

Build

Deploy Staging

Deploy Production



Data Engineers

The accuracy of the spatial-temporal machine learning model used for real time forecasting, may over time, decrease as new trends form (e.g. there may be increased demand in a new location or time period). This diagram outlines best practices around MLOps using **Amazon SageMaker Studio**. Using an **Amazon SageMaker Project**, model building and model deployment Pipelines, Experiments, Model groups, Endpoints and Repositories can be created automatically.

MLOps [Optional to add]

* An **Amazon SageMaker Pipeline** is started when a member of a data science team/data engineering team commit new code to a **AWS CodeCommit** repository
* The ML Model begins retraining which includes a data pre-processing Processing Job step and then onto a Training Job
* After the model is trained, the model is evaluated, against various metrics, using a Processing Job
* If model evaluation is successful, it is registered with **Amazon SageMaker Model Registry** under a Model Group, as a model is retrained, subsequent versions appear under the Model Group
* After the model is successfully registered and versioned, it triggers an **AWS CodePipeline** which takes starts the model deployment process with an **AWS CodeCommit** repository as a source
* After the Model Deployment deployment artifacts are built, the updated model is deployed to a staging serverless endpoint, from here, a team of data scientists can test the model’s inference
* If tests are successful, data scientists, can manually approve the deployment to production, the model is promoted to the production serverless endpoint; **Amazon SageMaker Serverless Inference** allows deployment of the model without selecting instance types or creating scaling policies
* Using an **Amazon SageMaker Model Monitor**, the model is continuously measured for prediction performance; Model Monitor also allows for model bias, model explainability and data quality monitoring