# Appendices and Deployment Guide

## Data Dictionary

### Finance Data

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Sample Data** |
| Stock | Stock Name of the Company | APPL |
| Date | Date when the stock price is gathered | 18/6/2024 |
| Open | Opening stock price | 217.59 |
| High | Highest stock price reached of the day | 219.63 |
| Low | Lowest stock price reached of the day | 213 |
| Close | Closing stock price | 214.29 |
| Adj Close | Closing Stock Price Adjusted for Dividends etc. | 214.29 |
| Volume | Volume traded | 7994300 |
| Close\_7\_Days | Closing 7-day price | 214.29 |

### News Data

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Sample Data** |
| Title | Title of article | Microsoft's AI chatbot will 'recall' everything you do on a PC |
| Date | Date of article | 20/5/2024 21:55 |
| Sources | Author of the article and last updated date of article | MATT O'BRIEN, MANUEL VALDESUpdated 20 May 2024 at 2:55 pmÂ·3-min read |
| Content | Full article text | ‘Content’ |
| URL | URL of the news article | https://sg.finance.yahoo.com/news/microsofts-ai-chatbot-recall-everything-185736872.html |
| Stock | Stock Name of the Company | APPL |

## Deployment Guide

### S3 Bucket

Create default bucket with any bucket name. Enable bucket versioning.

Take note of the bucket names as you will need them later. Recommended Setup is one bucket for Codes for EC2 to retrieve, The other bucket for the webscrape data.

If you need sample data, please put the news\_data\_overall.csv and stock\_price\_consolidated.csv file inside the webscrape bucket. Please also drop in inference\_finance\_consolidated.csv into the same bucket

### EC2 Instance

* + 1. Put the Two Codes below into the code-bucket-mle-cs611 S3 Bucket.
  1. Yahoo\_Finance\_Data\_Webscrape.py
  2. Yahoo\_Finance\_News\_Webscrape\_EC2.py
     1. Open EC2, Select Launch Instance, choose a Name, and leave all other settings as Default, then click launch instance

A screenshot of a computer

Description automatically generated

* + 1. Start the instance and run the commands in the EC2\_Install\_Webscrape.txt. Refer to the text file for more information.

### Lambda Functions

#### Role Creation

2 roles to be created. 1 role specific for web scraping the other role for the generic lambda functions

Webscrape Role

A screenshot of a computer

Description automatically generated

Role name: webscraper-role (or any name you choose)

Permissions:

1. AWSLambdaBasicExecutionRole (Customer Managed)
2. StartStopEC2 (Customer inline)

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "VisualEditor0",

"Effect": "Allow",

"Action": [

"ec2:StartInstances",

"ec2:StopInstances"

],

"Resource": "arn:aws:ec2:ap-southeast-1:357080086340:instance/i-0242760bdfc72d666"

},

{

"Sid": "VisualEditor1",

"Effect": "Allow",

"Action": [

"ssm:SendCommand",

"ec2:DescribeInstances",

"ssmmessages:\*",

"ssm:GetCommandInvocation"

],

"Resource": [

"\*",

"arn:aws:s3:::webscrape-bucket-mle611"

]

},

{

"Sid": "VisualEditor2",

"Effect": "Allow",

"Action": "ssm:SendCommand",

"Resource": "arn:aws:ec2:ap-southeast-1:357080086340:instance/i-0242760bdfc72d666"

}

]

}

Generic Lambda Role

Role Name: Any

Permissions:

A screenshot of a computer

Description automatically generated

putObjectS3 is optional as there is AmazonS3FullAccess

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"s3:PutObject",

"s3:PutObjectAcl"

],

"Resource": "arn:aws:s3:::webscrape-bucket-mle611/\*"

}

]

}

#### Functions

There should be a total of 8 Lambda functions to be created.

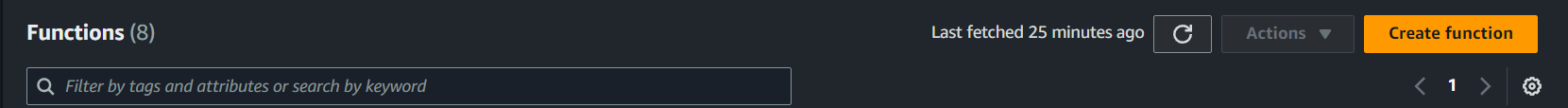
|  |  |
| --- | --- |
| **Name** | **Purpose** |
| Yahoo-News-Webscrape | News Webscraping |
| Yahoo-Finance-Webscrape | Finance Webscraping |
| Yahoo-Finance-S3 | Extraction of financial data from S3 |
| Yahoo-News-S3 | Extraction of news data from S3 |
| Yahoo-News-Update-S3 | Concatenation job for News data in S3 |
| Yahoo-Finance-Update-S3 | Concatenation job for Finance data in S3 |
| Finance-Inference-Update-S3 | Inference for finance price prediction |
| Yahoo-Sentiment-Extract | Extraction of sentiment analysis from S3 |

A screenshot of a computer

Description automatically generated

The setup for the above 8 is the same

1. Select Create Function.



1. Select author from scratch, type in the function name, Select runtime as Python 3.12, and change the default execution to the Generic Lambda Role set above, the click create function

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

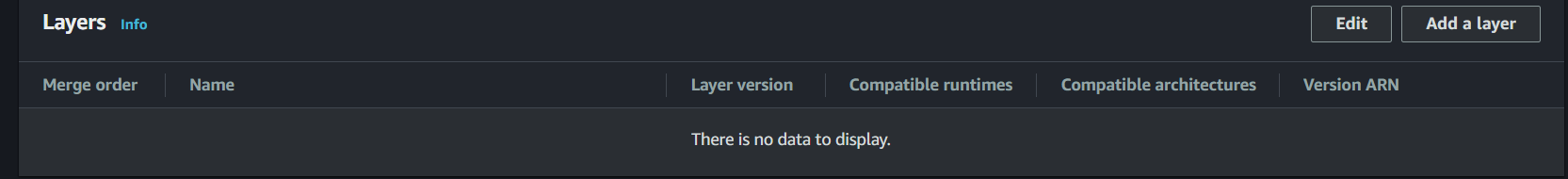
Description automatically generated

1. Once you created it, upload the respective zip file.

A screenshot of a computer

Description automatically generated

1. Go down to layers and click “Add a layer”



1. Select the layers as per below and click add.

A screenshot of a computer

Description automatically generated

1. There might be a need to increase the timeout, Memory and Storage. These can be found under the “Configurations” Tab.

A screenshot of a computer

Description automatically generated

### EventBridge Rules

To create 3 EventBridge Rules. Rules to be created instead of schedule as only rules can be set as the trigger for lLambda functions.

**Finance**

* + 1. Navigate to Rules in EventBridge and create rule

A screenshot of a computer

Description automatically generated

* + 1. Define rule detail as per the screenshot. Flexible time window set to 5min and Timeframe can be left blank.

A screenshot of a computer

Description automatically generated

* + 1. You can use any time to initiate the trigger. The example below shows the cron expression we used.

A screenshot of a schedule

Description automatically generated

* + 1. In the targets settings, select the finance data webscraping lambda function created previously.

A screenshot of a computer

Description automatically generated

* + 1. Proceed back to the Finance webscraping lambda function and add trigger

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**News**

Repeat the steps as in the finance rule. However, ensure the cron schedule is different from the finance rule and that the lambda function target is the news function. Also ensure that the target input is define as per the screenshot below. This is to select the stocks we want to ingest

A screenshot of a computer

Description automatically generated

**Update\_S3**

Repeat the steps as in the finance rule. However, ensure the cron schedule is different from the finance rule and that the lambda function target is the update lambda function.

Also ensure to include the payload for the following targets as per the screenshot below.

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

**Inference\_Update\_S3**

Repeat the steps as in the finance rule. However, ensure the cron schedule is different from the finance rule and that the lambda function target is the inference lambda function.

1. Follow the target settings below, with payload

{"sm\_finance\_endpoint\_name": "price-prediction-endpoint-2024-06-30-11-53-50",

"s3\_source\_bucket\_name": "webscrape-bucket-mle611",

"source\_file": "stock\_price\_consolidated.csv",

"s3\_dest\_bucket\_name": "webscrape-bucket-mle611",

"dest\_file": "Inference\_Finance\_Consoliated.csv"}

A screenshot of a computer program

Description automatically generated

### ML Pipeline

#### Stock\_Price\_Prediction

1. Ensure that the raw stock\_price\_consolidated.csv file is available in the bucket

A screen shot of a computer

Description automatically generated

1. Please Upload the “Stock\_Price\_Prediction\_Final.ipynb” file into sagemaker.
2. Use Kernel Data Science 3.0 to run the notebook. The endpoint will be deployed on sagemaker.

#### Sentiment\_Analysis

1. Run `pip install transformers` on AWS sagemaker terminal
2. Run `curl -fsSL https://ollama.com/install.sh | sh` on AWS sagemaker terminal
3. Run `ollama serve` on AWS sagemaker terminal
4. Please Upload ollama\_Final.ipynb into sagemaker and run it, the outputs will be found in S3

### API Gateway

There are 3 APIs here that needs to be set up

* Yahoo\_News\_API
* Yahoo\_Sentiment
* Yahoo-Finance-Data

All 3 APIs above follow the same steps for setup, if there is any difference it will be pointed out.

1. Navigate to API Gateway and Click Create API

A screenshot of a computer

Description automatically generated

1. Click on “Build” on Rest API

A screen shot of a computer

Description automatically generated

1. Key in the API name and Click Create API

A screenshot of a computer

Description automatically generated

1. Once Done, Click on Create Method

A screenshot of a computer

Description automatically generated

1. Select POST as Method Type, and Select the relevant lambda functions, then click create Method
   1. Yahoo\_News\_API – Yahoo\_News\_S3
   2. Yahoo\_Sentiment – Yahoo\_Sentiment\_Extract
   3. Yahoo-Finance-Data – Yahoo\_Finance\_S3

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A black rectangular object with a white line

Description automatically generated

1. Go to Enable CORS, and enable the below setting, then click save.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Click on Deploy API, select new Stage, choose a name for the new stage, then click deploy.

A screenshot of a computer

Description automatically generated

A screenshot of a stage setting

Description automatically generated

### Frontend Deployment

Requirements:

1. Node.js and npm must be installed.

To deploy the react application code to AWS Amplify, the process starts by building the application using the “npm run build” command, which creates a production-ready version of the app in the “build” folder. Navigate to the project directory at “CS611\_MLE\_Project\frontend” to run this command. Next, log in to the AWS Amplify Console and create a new Amplify app by choosing to deploy without a Git provider. Then, drag and drop the build folder into the provided area on the console. Once uploaded, Amplify automatically starts deploying the application, and the deployment process can be monitored through the console. After the deployment is complete, Amplify provides a URL where the live application can be accessed.

**Steps to Deploy React Code to AWS Amplify:**

1. Open the terminal and navigate to the React project directory:

cd CS611\_MLE\_Project\frontend

2. Run the following command to create a production build of the application

npm run build

This command generates a “build” folder containing the optimized production-ready files.

A screenshot of a computer screen

Description automatically generated

3. Open AWS console, and go to AWS Amplify service. Click “Create new app” and select “Deploy without Git” and click next.

A screenshot of a computer

Description automatically generated

4. Open the build folder and zip the contents.

A screenshot of a computer

Description automatically generated

5. Drag and drop the zipped folder into the designated area.

A screenshot of a computer

Description automatically generated

6. Click Save and deployed.

A screenshot of a computer

Description automatically generated

7. A domain URL is generated, allowing access to the web app.

A screenshot of a computer

Description automatically generated