

Real-Time Inventory Optimization Using AWS Lambda and Amazon Kinesis Configuration Manual

MSc Research
Cloud Computing

Abhishek Yadav
Student ID: 23153423

School of Computing
National College of Ireland

Supervisor: Dr. Shivani Jaswal

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name: Abhishek Yadav
Student ID: 23153423
Programme: Cloud Computing **Year:** 2024
Module: MSc Research Project
Lecturer: Dr. Shivani Jaswal
Submission Due Date: 12/12/2024
Project Title: Real-Time Inventory Optimization Using AWS Lambda and Amazon Kinesis
Word Count: **Page Count:** 11

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Abhishek Yadav.....
Date: 11/12/2024.....

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	<input type="checkbox"/>
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Abhishek Yadav
Student ID: 23153423

1 Introduction

The configuration manual provides a step-by-step instruction to how to setup the cloud services for the inventory management services for the user to perform the application work in local live environment based on the lit review performed by various paper and analysing the methodology of done by different researchers to make the prediction and forecast.

2 Steps for Configure the steup

Certation of account :

Visit AWS Registration to step up the account and provide all relevant information.

Now got to billing preferences to monitor costs chart.

Then setup of data Streaming with Kinesis

In the AWS Management Console search Kinesis and click on Kinesis Data Streams and follow the below mention steps.

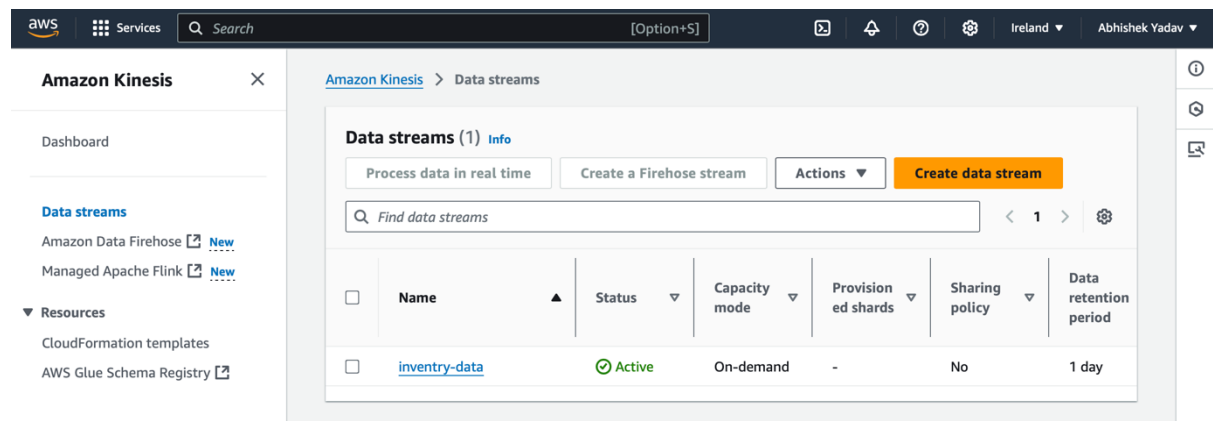


Figure1:Setup for Kinesis.

For creating the data stream:

- Click on create data stream.
- Give the name to your stream (RealTimeInventoryStream).
- In this we have to Specify the number of shards based.

Configure stream retention:

- Set the retention period (default is 24 hours).

Now moving to the intergradation of kinesis producers:

- Use the Kinesis SDK in your application to push inventory data into the stream.

Data stream settings

You can edit the settings after the data stream has been created and is in the active status.

Setting	Value	Editable after creation
Capacity mode	On-demand	✓ Yes
Data retention period	1 day	✓ Yes
Server-side encryption	Disabled	✓ Yes
Monitoring enhanced metrics	Disabled	✓ Yes
Data stream sharing policy	No policy	✓ Yes

Tags - optional [Info](#)

You can add tags to organize your AWS resources, track costs, and control access.

No tags associated with the Kinesis Data Stream.

Add new tag

You can add up to 50 more tags.

Cancel
Create data stream

Figure2: Kinesis data configuration.

Setting up AWS Lambda for for function trigger.

First we need to create a lambda function for that go to AWS Lambda and click on create a functions.

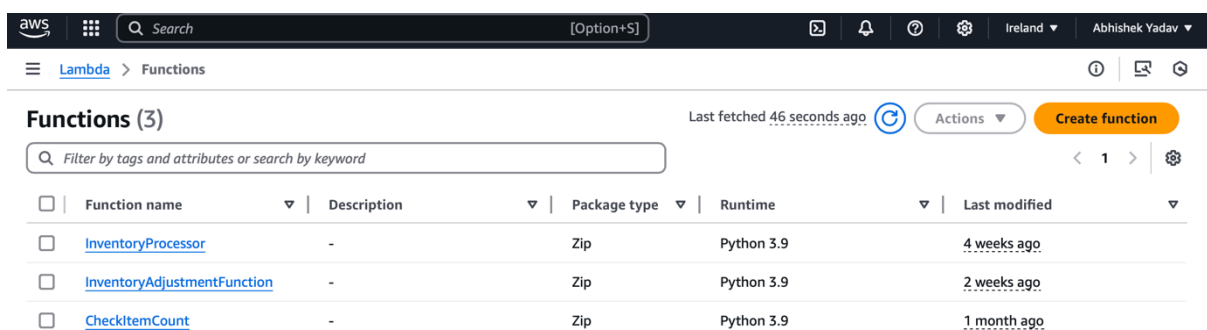


Figure3: Lambda function creation.

- Give the function name (Inventory process data).
- select the programming language as we choose (Python).
- Now we can create kinesis trigger:

- click to add triggers button and select the service as kinesis.

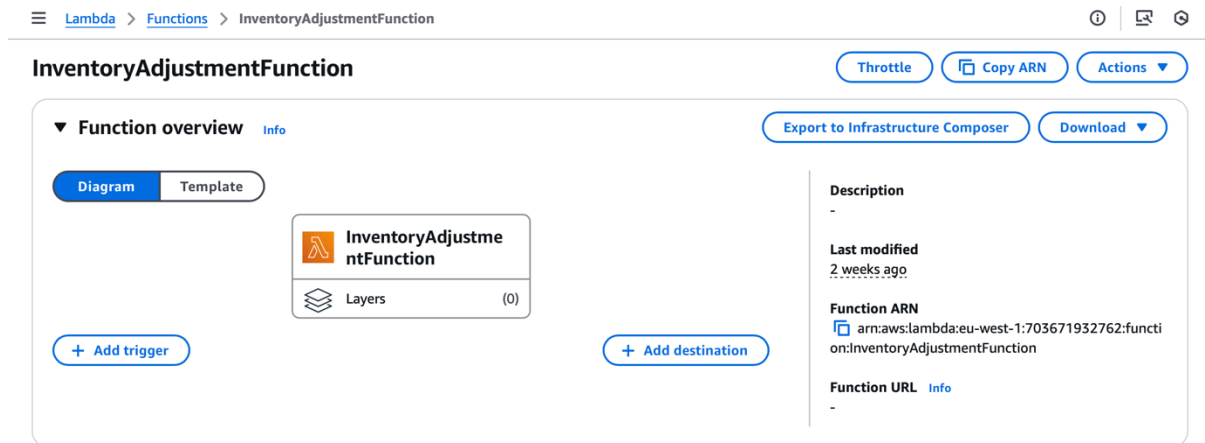


Figure4:Connection with trigger services.

- Here we can link the created which we have already created.

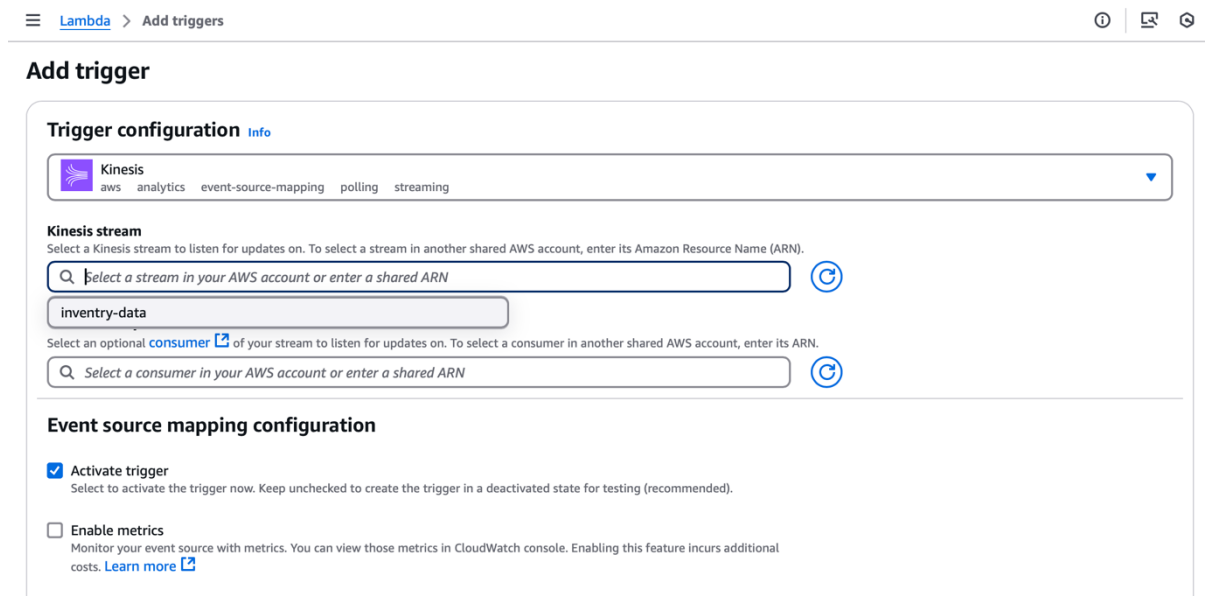


Figure5:Kinesis configuration with lambda function.

Now we have to write a lambda code which is the function to process incoming data and store results in DynamoDB.

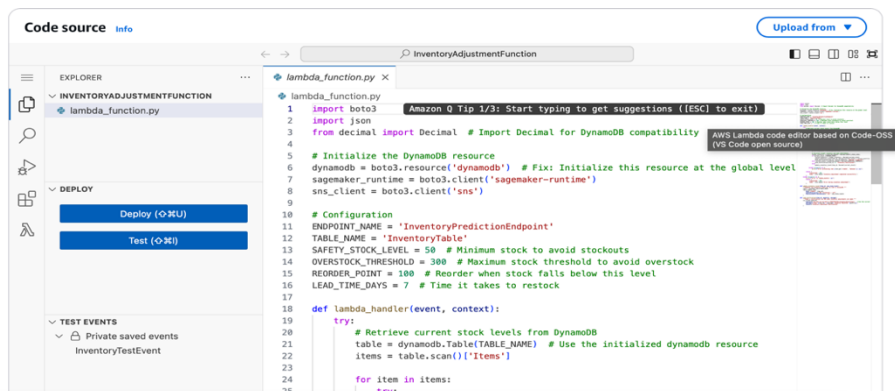


Figure6:Fuction process of code.

Moving Forward Data Storage on DynamoDB and S3.

- For setting up DynamoDB follow the below mention steps.
 - Go to amazon DynamoDB and create the table.
 - Give the name to the table (InventoryData).
 - Now we have to set a partition key (ItemID).

Create table

Table details [Info](#)

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name
This will be used to identify your table.

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

String

1 to 255 characters and case sensitive.

Sort key - optional
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

String

1 to 255 characters and case sensitive.

Figure7:Creation of DynamoDB Table

- For S3 Bucket setup:
 - Go to S3 and Create Bucket.
 - Give suitable name to the bucket (InventoryDataStorage).
 - After that set permissions as public access.

General purpose buckets (4) [Info](#) All AWS Regions

[Refresh](#) [Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

Buckets are containers for data stored in S3.

	Name	AWS Region	IAM Access Analyzer	Creation date
<input type="radio"/>	inventory-data-historical	Europe (Ireland) eu-west-1	View analyzer for eu-west-1	October 31, 2024, 08:26:23 (UTC+00:00)
<input type="radio"/>	inventorybucket	Europe (Ireland) eu-west-1	View analyzer for eu-west-1	November 10, 2024, 22:00:42 (UTC+00:00)
<input type="radio"/>	sagemaker-eu-west-1-703671932762	Europe (Ireland) eu-west-1	View analyzer for eu-west-1	October 31, 2024, 08:48:33 (UTC+00:00)
<input type="radio"/>	sagemaker-studio-703671932762-mgj7mbvn7k	Europe (Ireland) eu-west-1	View analyzer for eu-west-1	October 31, 2024, 08:48:27 (UTC+00:00)

Figure8:Setting and creation of S3 bucket.

- The Intergradation of S3 with Lambda:
 - Make the changes inside the Lambda function to save the processed data of inventory into the S3 bucket.

Machine Learning model integration with amazon SageMaker

- Go to Amazon SageMaker and click on create notebook instance.
- Name the instance “InventoryOptimizationModel”.
- Select the instance which show type ml.t3.medium.

The screenshot shows the Amazon SageMaker console interface. At the top, there's a breadcrumb trail: "Amazon SageMaker AI > Notebook instances > InventoryManagement". Below this, the title "InventoryManagement" is displayed with buttons for "Delete", "Stop", "Open Jupyter", and "Open JupyterLab". The main section is titled "Notebook instance settings" with an "Edit" button. It contains a table of instance details:

Name	InventoryManagement	Notebook instance type	ml.t3.medium
ARN	arn:aws:sagemaker:eu-west-1:703671932762:notebook-instance/InventoryManagement	Elastic Inference	-
Lifecycle configuration	-	Volume Size	5GB EBS
Status	✔ InService	Platform identifier	Amazon Linux 2, Jupyter Lab 3 (notebook-al2-v2)
Creation time	Oct 31, 2024 08:59 UTC	Minimum IMDS Version	2
Last updated	Dec 09, 2024 13:39 UTC		

Figure9:Stage Maker jupyter notebook configuration.

1. To Preparing and training dataset for prediction follow the given steps.
- Use SageMaker to preprocess the data inside the notebook.

The screenshot shows a Jupyter Notebook interface with the title "Untitled" and a last checkpoint of "31/10/2024 (autosaved)". The notebook is running on "conda_python3". The code in the cell is as follows:

```
In [2]: import pandas as pd
import boto3
from statsmodels.tsa.arima.model import ARIMA
import matplotlib.pyplot as plt

s3 = boto3.client('s3')
bucket_name = 'inventory-data-historical'
file_name = 'historical_inventory.csv'

s3.download_file(bucket_name, file_name, '/tmp/historical_inventory.csv')

data = pd.read_csv('/tmp/historical_inventory.csv', parse_dates=['date'], index_col='date')
print(data.head())

model = ARIMA(data['stockLevel'], order=(1,1,1))
model_fit = model.fit()

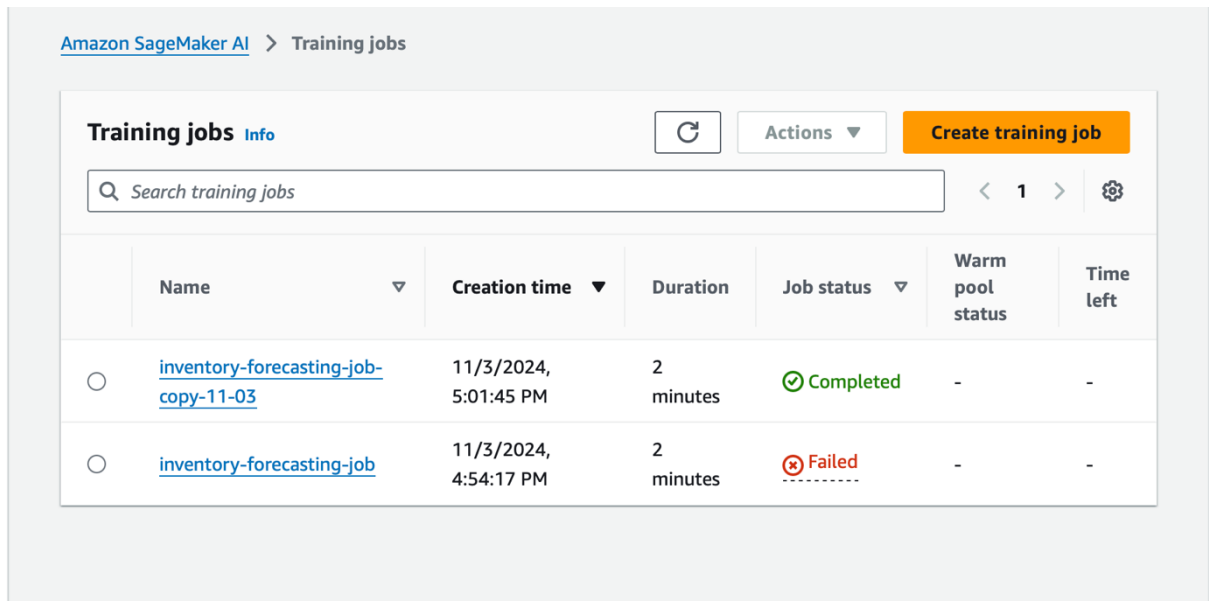
forecast = model_fit.forecast(steps=10)
print("Forecasted values:", forecast)

data['stockLevel'].plot(label='Historical Stock Level')
forecast.plot(label='Forecast', color='red')
plt.legend()
plt.show()
```

Figure10:Stage Maker jupyter notebook configuration with S3 bucket.

For Model training follow these steps.

- Use built-in algorithms Autoregressive Integrated Moving Average for train the model on inventory trends.

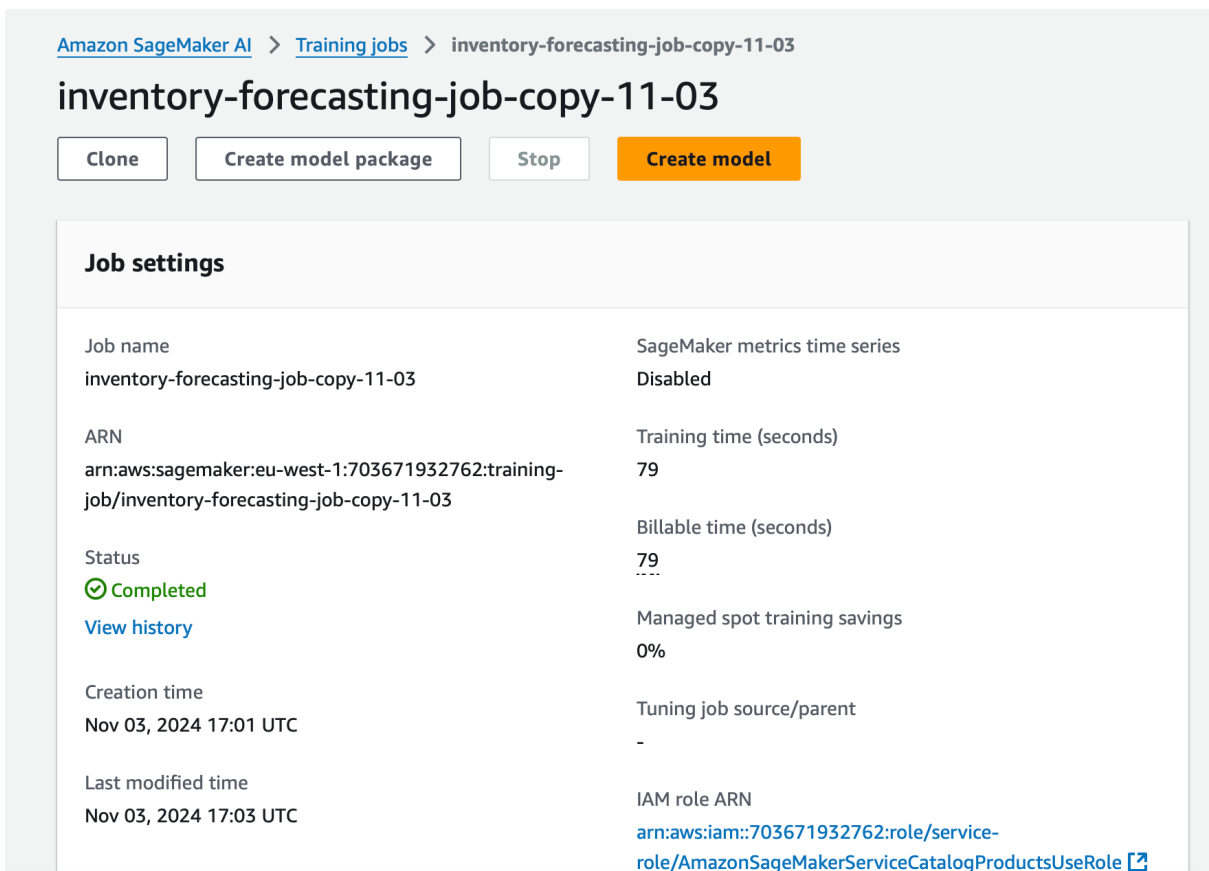


The screenshot shows the Amazon SageMaker AI Training jobs page. At the top, there's a breadcrumb trail: [Amazon SageMaker AI](#) > [Training jobs](#). Below this is a header section with 'Training jobs Info', a refresh button, an 'Actions' dropdown, and a 'Create training job' button. A search bar labeled 'Search training jobs' is also present. The main content is a table with columns: Name, Creation time, Duration, Job status, Warm pool status, and Time left. There are two rows of training jobs, both named 'inventory-forecasting-job-copy-11-03'. The first job is 'Completed' with a duration of 2 minutes, created on 11/3/2024 at 5:01:45 PM. The second job is 'Failed' with a duration of 2 minutes, created on 11/3/2024 at 4:54:17 PM.

	Name	Creation time	Duration	Job status	Warm pool status	Time left
<input type="radio"/>	inventory-forecasting-job-copy-11-03	11/3/2024, 5:01:45 PM	2 minutes	✓ Completed	-	-
<input type="radio"/>	inventory-forecasting-job-copy-11-03	11/3/2024, 4:54:17 PM	2 minutes	✗ Failed	-	-

Figure11:Stage Maker model training completion .

- Deploy the model train model as an endpoint for real-time prediction.



The screenshot shows the details page for a specific training job in Amazon SageMaker AI. The breadcrumb trail is: [Amazon SageMaker AI](#) > [Training jobs](#) > [inventory-forecasting-job-copy-11-03](#). The job name 'inventory-forecasting-job-copy-11-03' is prominently displayed. Below the name are buttons for 'Clone', 'Create model package', 'Stop', and 'Create model'. The 'Job settings' section is divided into two columns. The left column contains: Job name (inventory-forecasting-job-copy-11-03), ARN (arn:aws:sagemaker:eu-west-1:703671932762:training-job/inventory-forecasting-job-copy-11-03), Status (✓ Completed), View history link, Creation time (Nov 03, 2024 17:01 UTC), and Last modified time (Nov 03, 2024 17:03 UTC). The right column contains: SageMaker metrics time series (Disabled), Training time (seconds) (79), Billable time (seconds) (79), Managed spot training savings (0%), Tuning job source/parent (-), and IAM role ARN (arn:aws:iam::703671932762:role/service-role/AmazonSageMakerServiceCatalogProductsUseRole).

Job settings	
Job name inventory-forecasting-job-copy-11-03	SageMaker metrics time series Disabled
ARN arn:aws:sagemaker:eu-west-1:703671932762:training-job/inventory-forecasting-job-copy-11-03	Training time (seconds) 79
Status ✓ Completed View history	Billable time (seconds) 79
Creation time Nov 03, 2024 17:01 UTC	Managed spot training savings 0%
Last modified time Nov 03, 2024 17:03 UTC	Tuning job source/parent -
	IAM role ARN arn:aws:iam::703671932762:role/service-role/AmazonSageMakerServiceCatalogProductsUseRole

Figure12:Deploying the train model .

- we will use this endpoint to forecast inventory hits coming from the kinesis.

Amazon SageMaker AI > Endpoints > nventoryPredictionEndpoint

nventoryPredictionEndpoint Delete

Endpoint summary

Name nventoryPredictionEndpoint	Status ✔ InService	Type Real-time
ARN arn:aws:sagemaker:eu-west-1:703671932762:endpoint/nventoryPredictionEndpoint	Creation time Sun Dec 08 2024 10:43:34 GMT+0000 (Greenwich Mean Time)	Last updated Sun Dec 08 2024 10:46:35 GMT+0000 (Greenwich Mean Time)
URL https://runtime.sagemaker.eu-west-1.amazonaws.com/endpoints/nventoryPredictionEndpoint/invocations Learn more about the API	Model container logs /aws/sagemaker/endpoints/nventoryPredictionEndpoint	Alarms 0 alarms

Figure13:Endpoint to demonstration for prediction.

After the machine model integration we have to set up visualizing Data so that user can understand the trend therefore we will step up Amazon QuickSight

- For the QuickSightint setup go to Amazon QuickSight and Sign Up.
- Then choose the standard edition for a single user license.
- Connect to Data Sources and then add the DynamoDB and S3 as data sources.

Find analyses & more

- ★ Favorites
- 🕒 Recent
- 📁 My folders
- 📁 Shared folders
- 📊 Dashboards
- 📖 Data stories
- 📈 Analyses
- 📁 Datasets**

Datasets New dataset

Name		Owner	Last Modified	
results.csv	SPICE	Me	17 days ago	⋮
InventoryDataS3	SPICE	Me	22 days ago	⋮
People Overview	SPICE	Me	a month ago	⋮
Business Review	SPICE	Me	a month ago	⋮
Sales Pipeline	SPICE	Me	a month ago	⋮
Web and Social Media Analytics	SPICE	Me	a month ago	⋮

Figure14:QuickSight dataset demonstration setup.

- Now we will create a dashboards to visualize inventory trends and predictions.

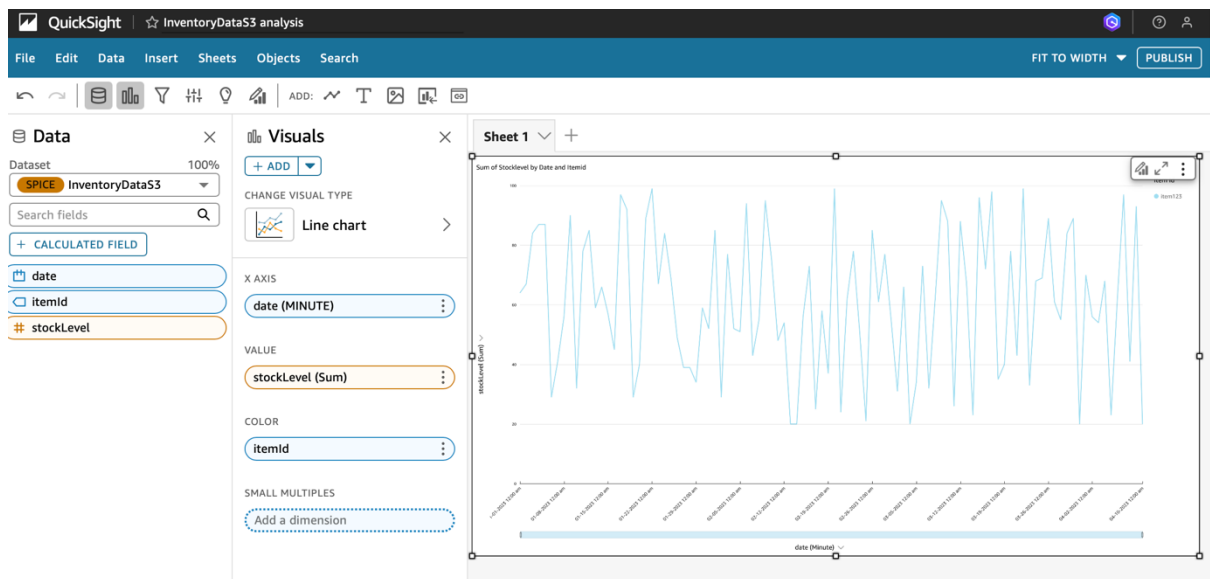


Figure15:QuickSight dashboard creation.

Once's the visual setup is done we move to setting up the Monitoring and Optimization in Amazon CloudWatch For that follow the below mention steps.

- Go to CloudWatch and click on create an alarm and select the custom Metrics .

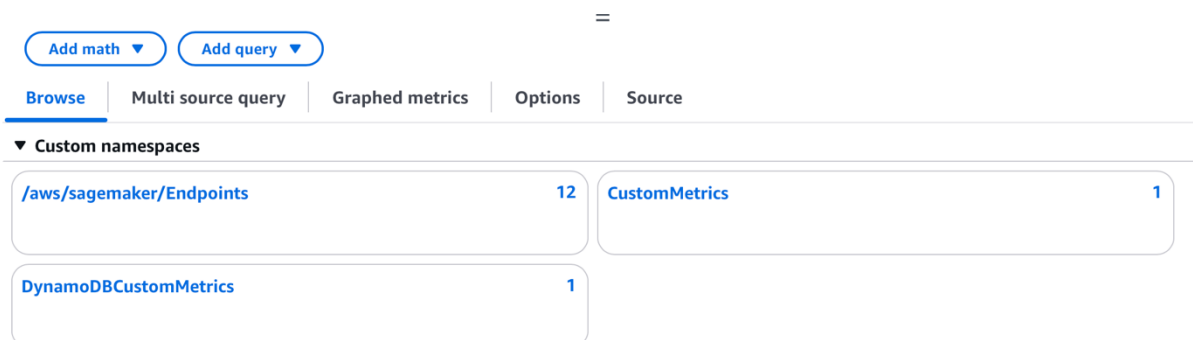


Figure16:Cloud watch alarm custom metrics selection.

- Click on the create alarms button and give name and defined the alarm type.
- Turn on the notifications via Amazon SNS to alert .

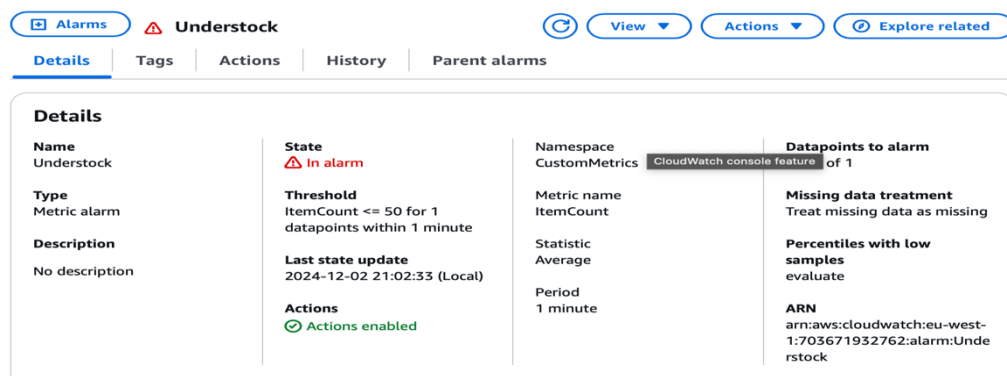


Figure17:Alarm configuration all settings.

1. Simulation Performances Steps

1. For Simulate Inventory first need to connect the EC2 instances with local terminal to perform update on inventory, so go to the connect button and click on it.

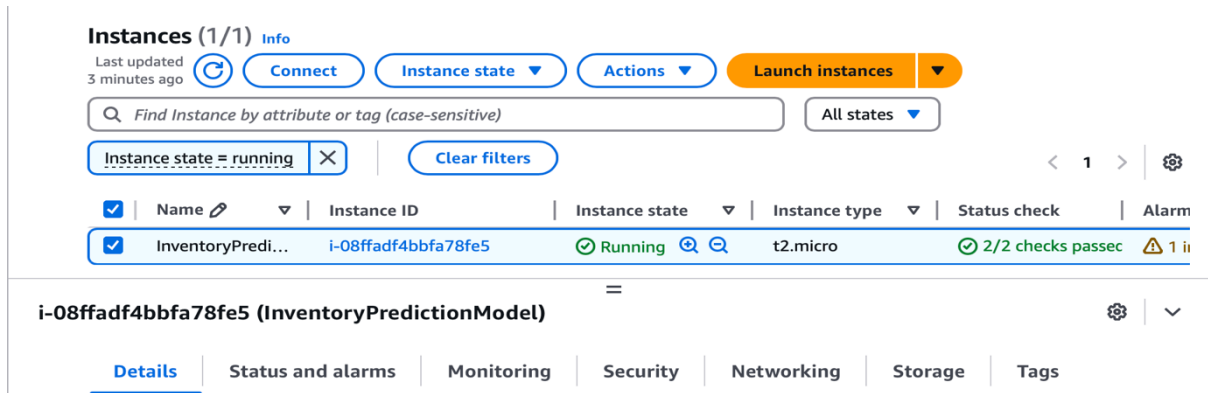


Figure18:EC2 instances connection.

2. After this it will open SSH client in which we have to copy the path of Example

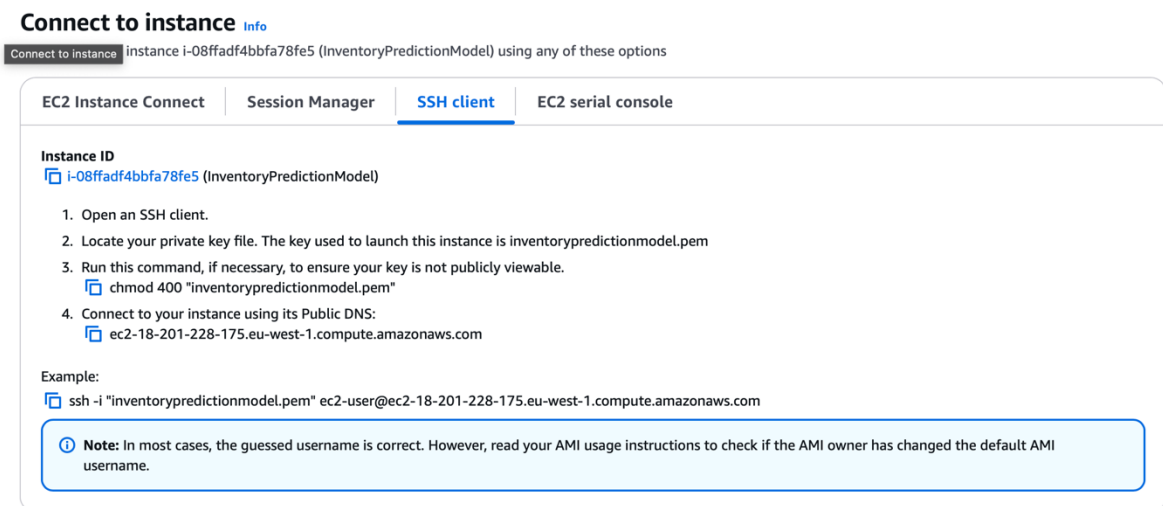


Figure19:SSH key for connection.

3. After this past this path into local terminal.

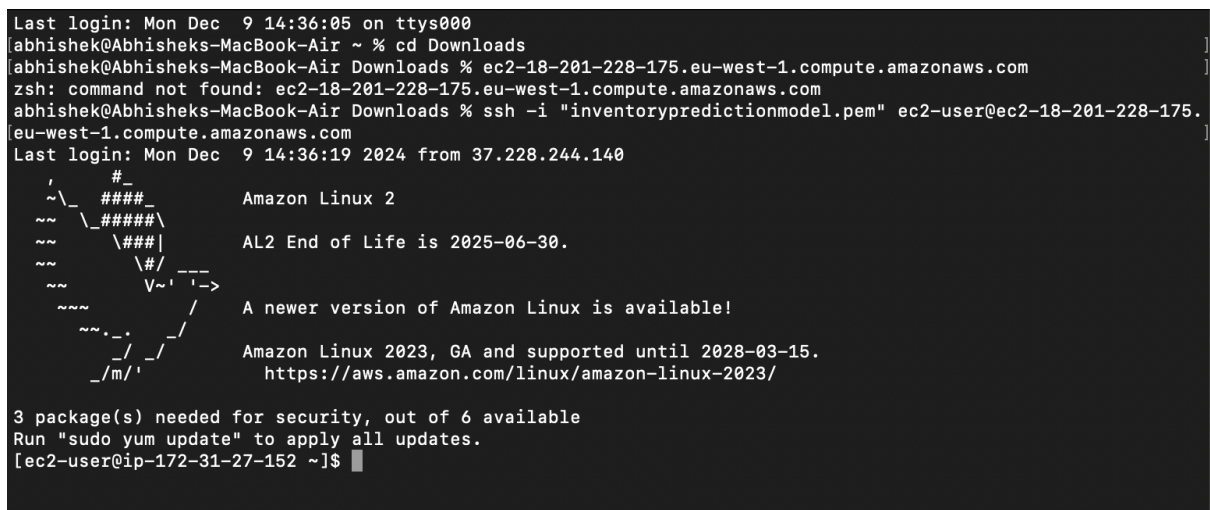


Figure20:Terminal demonstration of Connection.

4. The connection with the terminal and DynamoDB has been made now run the below mention code to check two parameter of Overstock and Understock situation which is defined to forece fully update the data into loop and check the situation of the inventory
 - Overstock

```
for i in {1..500}; do
  aws dynamodb put-item \
    --table-name InventoryTable \
    --item "{\"itemId\": {\"S\": \"SampleItemId$i\"}, \"attribute1\": {\"S\": \"Value$i\"}}"
done
```
 - Understock

```
for i in {1..500}; do
  aws dynamodb delete-item \
    --table-name InventoryTable \
    --key "{\"itemId\": {\"S\": \"SampleItemId$i\"}}"
done
```
5. After running above mention code user need to go on cloud watch and see the alarms which has been set for give an update on each threshold parameter and demonstrate the uses of CPU, lambda utilization of handling the situation and user get responses on email of each trigger and max and min limits of utilization.