AI-Powered Data Pipeline for E-commerce Warehouse Demand Forecasting

# Implementation Steps

**Data Ingestion Using Terraform and Python**

Data Processing Using AWS Glue & Lambda

Machine Learning with Amazon SageMaker (for demand forecasting)

Model Integration & Optimisation

Deployment & Scaling

A diagram of a computer

AI-generated content may be incorrect.

Table of Contents

[Implementation Steps 1](#_Toc192929558)

[Step 1. Data Ingestion Using Terraform and Python 3](#_Toc192929559)

[Objective: 3](#_Toc192929560)

[Set Up AWS Cloud Shell 3](#_Toc192929561)

[Create Terraform Files 4](#_Toc192929562)

[Deploy Terraform Infrastructure 5](#_Toc192929563)

[Verify Setup 5](#_Toc192929564)

[Simulate Data Using Python Script 6](#_Toc192929565)

[Monitoring and Logging 9](#_Toc192929566)

[Issues Encountered 10](#_Toc192929567)

[**Problem:** 10](#_Toc192929568)

[**Solution:** 10](#_Toc192929569)

[Summary of Key Deliverables 10](#_Toc192929570)

# Step 1. Data Ingestion Using Terraform and Python

## Objective:

* Use Terraform to set up Amazon Kinesis and Amazon S3 for real-time data ingestion.
* Simulate sales and inventory data using a Python script to mimic multiple warehouse systems.

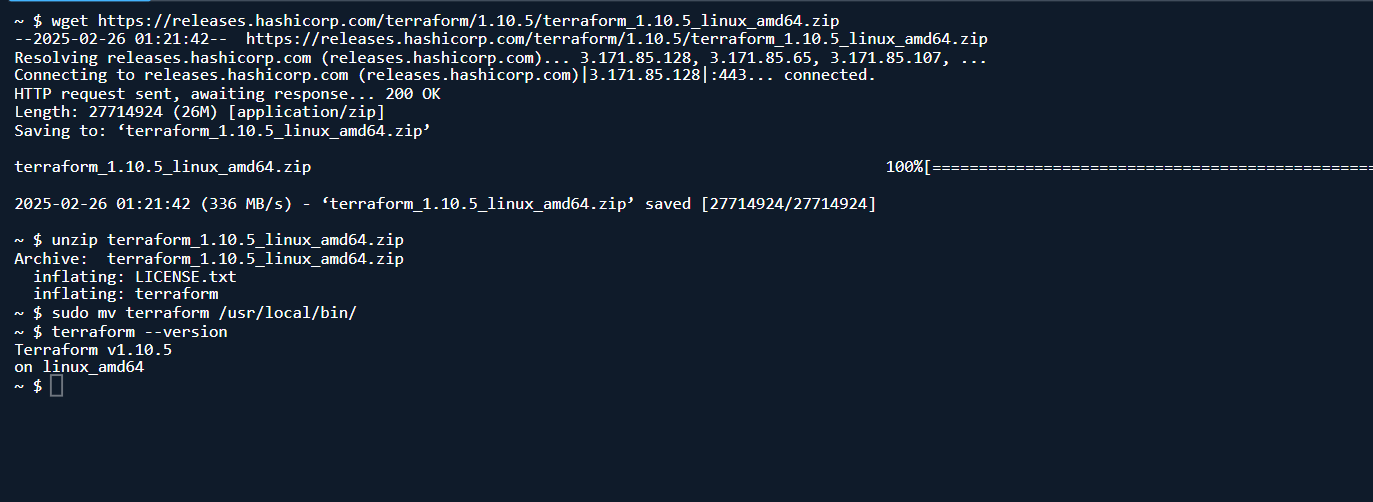
## **Set Up AWS Cloud Shell**

1. Verify terraform installation.

A screenshot of a computer program

AI-generated content may be incorrect.

1. If the Terraform version is out of date you can update it as shown below:



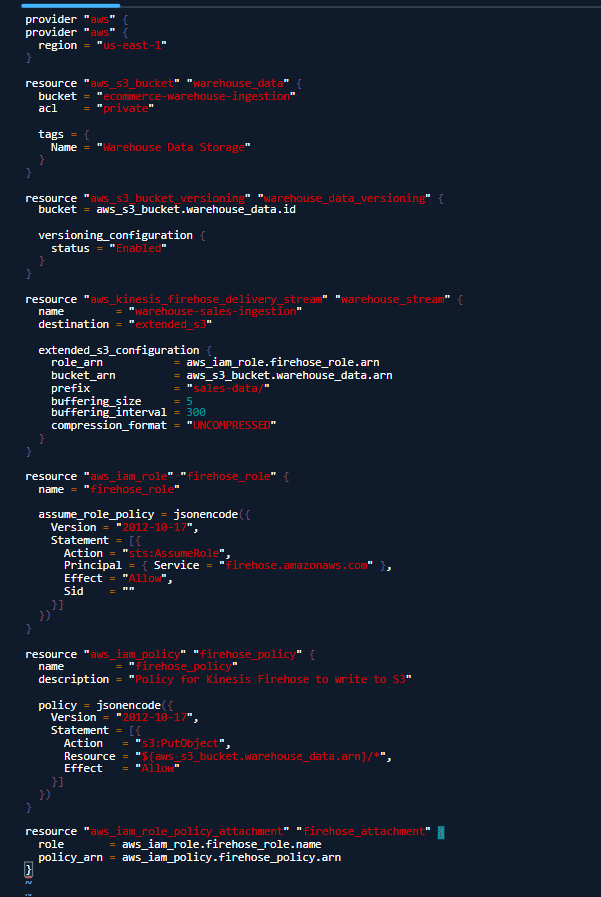
## Create Terraform Files

* 1. Create a directory and navigate into that directory

A blue screen with white text

AI-generated content may be incorrect.

* 1. Create a main.tf file which when applied will create an S3 bucket to store sales data, configure Amazon Kinesis Firehose to receive and send real-time data to this bucket, and establish IAM roles and policies that grant Firehose secure access to the bucket.



## Deploy Terraform Infrastructure

1. Terraform Init

**A screen shot of a computer

AI-generated content may be incorrect.**

1. Terraform Plan

**A computer screen shot of white text

AI-generated content may be incorrect.**

1. Terraform Apply

**A screenshot of a computer program

AI-generated content may be incorrect.**

## Verify Setup

1. Check S3 Bucket: Go to AWS Console → S3 → Ensure the bucket is created.

A screenshot of a computer

AI-generated content may be incorrect.

1. Check Kinesis Firehose: Go to AWS Console → Kinesis → Ensure the firehose stream is created.

A screenshot of a computer

AI-generated content may be incorrect.

## Simulate Data Using Python Script

1. **Upload Python Script to AWS Cloud Shell.**

**A computer code on a dark background

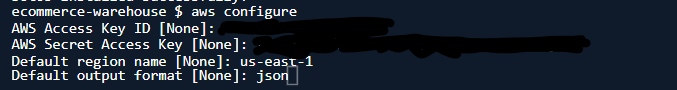
AI-generated content may be incorrect.**

1. Install required libraries

**A screen shot of a computer code

AI-generated content may be incorrect.**

1. Configure AWS CLI

****

1. Upload the code for the Simulation

**A computer screen shot of a program code

AI-generated content may be incorrect.**

1. Run the simulation

**A screen shot of a computer screen

AI-generated content may be incorrect.**

1. Verify data in Amazon S3

**A close up of a screen

AI-generated content may be incorrect.**

## Monitoring and Logging

1. Monitor Kinesis Firehose: AWS Console → Amazon Kinesis → Delivery Streams → Monitoring tab. Check data delivery success, throughput, and error logs.

A screenshot of a computer

AI-generated content may be incorrect.

## Issues Encountered

### **Problem:**

While running terraform plan, we encountered an *unsupported block type* error related to s3\_configuration. The error indicated that s3\_configuration was no longer supported for the aws\_kinesis\_firehose\_delivery\_stream resource.

### **Resolution:**

To resolve this, we replaced s3\_configuration with extended\_s3\_configuration, which is the correct format for newer versions of Terraform. Additionally, we addressed the deprecation warning for acl = "private" by removing it and relying on the default settings. After making these corrections, Terraform successfully created the necessary resources.

**A screenshot of a computer

AI-generated content may be incorrect.**

# Summary of Key Deliverables

1. **AWS Cloud Shell:** Configured to run Terraform and Python scripts.
2. **Terraform Deployment:** Kinesis Firehose and S3 bucket deployed using Terraform.
3. **Python Data Simulation:** Sales and inventory data simulated and sent to Kinesis Firehose.
4. **Real-Time Data Storage:** Data delivered to S3 and verified in the sales-data/ folder.
5. **Monitoring:** Kinesis Firehose and CloudWatch Logs configured for performance monitoring.