Kafka Stock Market Project - README

# Step 1: Create EC2 Machine

1. Launch an EC2 Linux instance (Ubuntu preferred).  
2. Connect to the instance using SSH.

# Step 2: Install Kafka on Ubuntu

Run the following commands:  
  
```bash  
wget https://downloads.apache.org/kafka/3.3.1/kafka\_2.12-3.3.1.tgz  
tar -xvf kafka\_2.12-3.3.1.tgz  
java -version  
sudo apt-get update  
sudo apt-get install openjdk-8-jdk -y  
java -version  
cd kafka\_2.12-3.3.1  
```

# Step 3: Start ZooKeeper

```bash  
bin/zookeeper-server-start.sh config/zookeeper.properties  
```

# Step 4: Start Kafka Server

Open another SSH session and run:  
  
```bash  
export KAFKA\_HEAP\_OPTS="-Xmx256M -Xms128M"  
cd kafka\_2.12-3.3.1  
bin/kafka-server-start.sh config/server.properties  
```  
  
Update `config/server.properties`:  
```bash  
advertised.listeners=PLAINTEXT://<PUBLIC\_IP>:9092  
```

# Step 5: Create a Kafka Topic

```bash  
cd kafka\_2.12-3.3.1  
bin/kafka-topics.sh --create --topic demo\_test --bootstrap-server <PUBLIC\_IP>:9092 --replication-factor 1 --partitions 1  
```

# Step 6: Start Kafka Producer

```bash  
bin/kafka-console-producer.sh --topic demo\_test --bootstrap-server <PUBLIC\_IP>:9092  
```

# Step 7: Start Kafka Consumer

```bash  
bin/kafka-console-consumer.sh --topic demo\_test --bootstrap-server <PUBLIC\_IP>:9092  
```

--------------------------------------------------------------------------------------------------------

# Step 8: Security Group Configuration

Update EC2 Security Group inbound rules:  
- Allow All Traffic from Anywhere (0.0.0.0/0) or MyIP  
- Save rules

# Step 9: Jupyter Notebook - Kafka Producer & Consumer

Install kafka-python:  
```bash  
pip install kafka-python  
```  
  
Producer Notebook:  
- Generate dummy stock data  
- Send to Kafka topic  
  
Consumer Notebook:  
- Continuously read messages from the topic  
  
Use `<PUBLIC\_IP>` in connection settings.

----------------------------------------------------------------------------------------------------

# Step 10: Send Data to S3

1. Create an S3 bucket  
2. Install and configure AWS CLI locally:  
```bash  
aws configure  
```  
3. Update producer code to push JSON messages to S3.

# Step 11: Glue Crawler Setup

1. Go to AWS Glue  
2. Create a new Database (e.g., stock-kafka-db)  
3. Create a Glue Crawler:  
- Select S3 bucket as data source  
- Choose IAM Role with Glue + S3 access + admin access  
- Run crawler → Creates table in Glue DB

# Step 12: Query Data in Athena

1. Open Athena  
2. Select database (stock-kafka-db)  
3. Run SQL queries (e.g., SELECT COUNT(\*) FROM table\_name;)  
4. Store query results in S3 output bucket

# Step 13: Real-Time Streaming

1. Add delay in producer code to simulate real-time streaming  
2. Keep consumer active in Jupyter  
3. Monitor Athena query results to see live data ingestion

# Project Completed

We successfully implemented a Kafka-based Stock Market streaming project:  
- Kafka Producer & Consumer on EC2  
- Real-time data pushed to S3  
- Glue crawler & Athena for querying and analysis

ss :