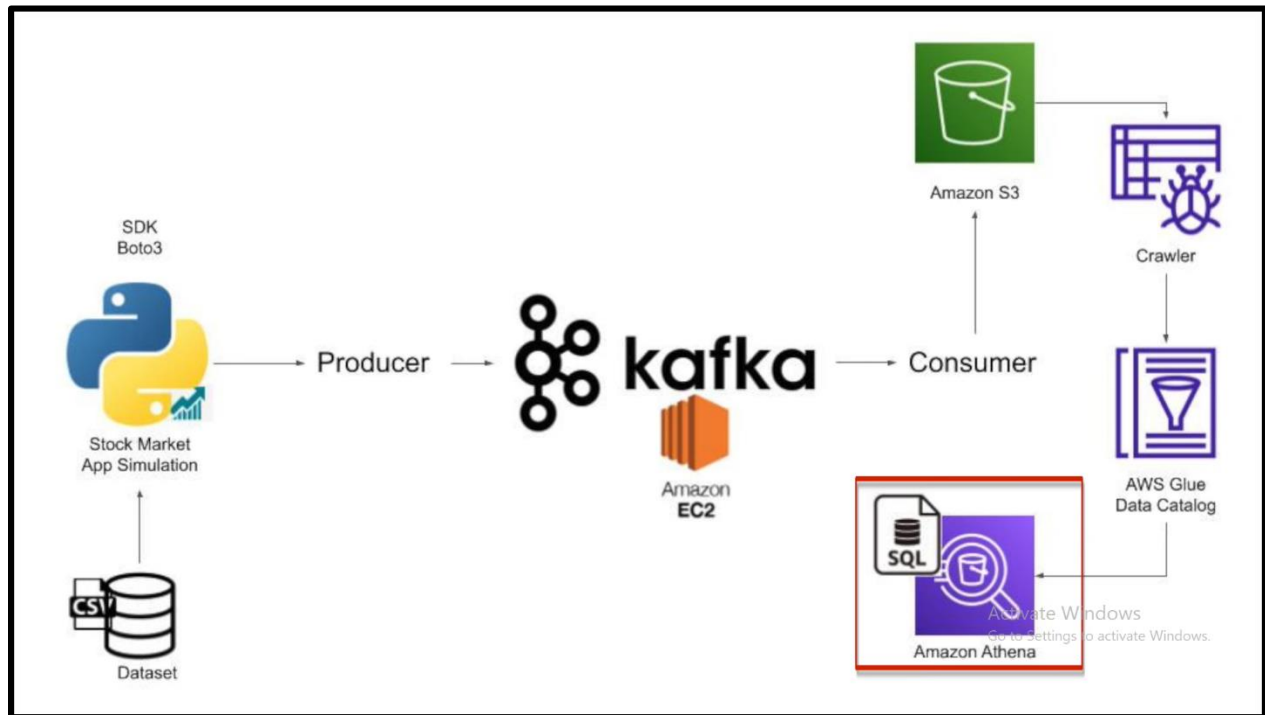


KAFKA STREAMING PROJECT



Step 1: Use python to produce stock data - - - - - by looping and giving to producer

Step 2: Use Kafka cluster to consume the data to S3 bucket of amazon- - - - - giving path to save the data to consumer in s3 bucket.

Step 3: Crawl that data to build a Glue catalog

Step 4: Analyze the data using Amazon Athena

Kafka real time:

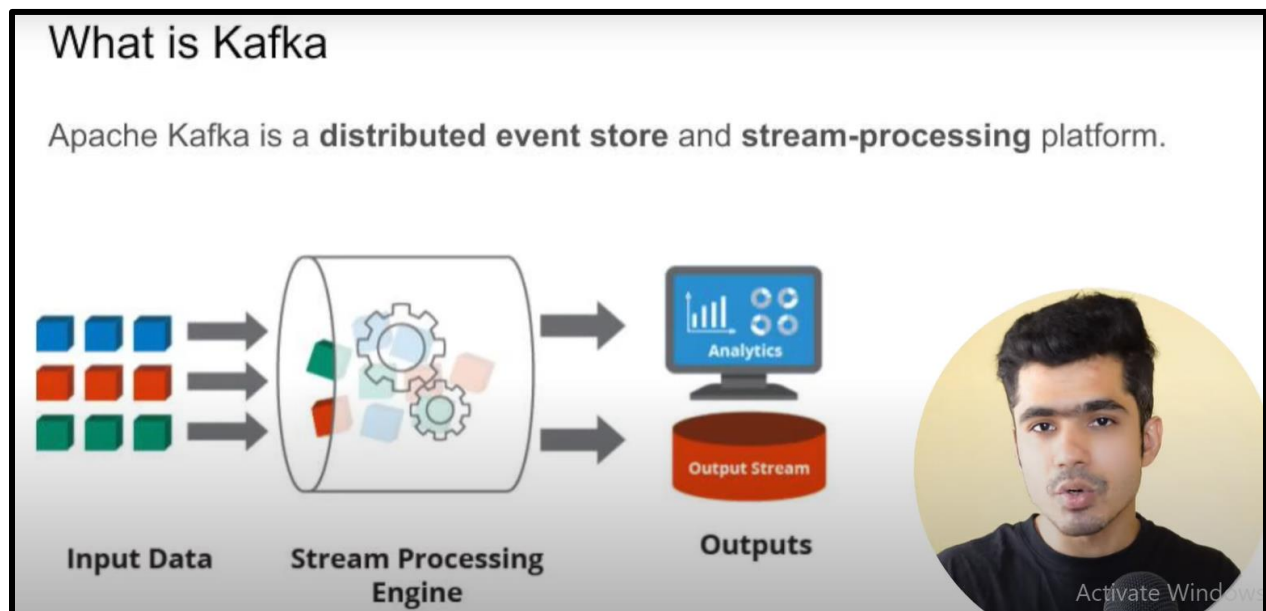
Introduction

In this project, you will execute an End-To-End Data Engineering Project on Real-Time Stock Market Data using Kafka.

We are going to use different technologies such as Python, Amazon Web Services (AWS), Apache Kafka, Glue, Athena, and SQL.

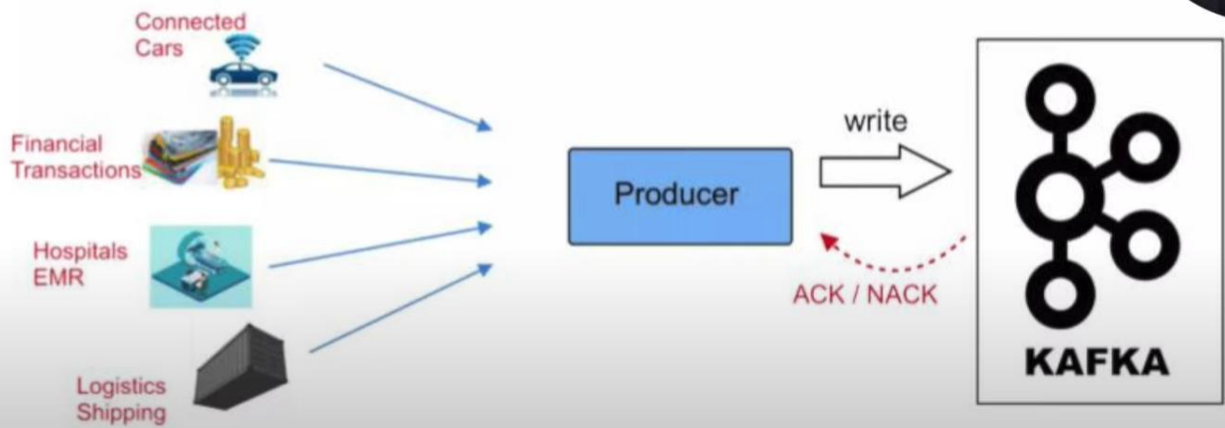
Technologies Used

- Programming Language - Python
- Amazon Web Service (AWS)
- S3 (Simple Storage Service)
- Athena
- Glue Crawler
- Glue Catalog
- EC2
- Apache Kafka

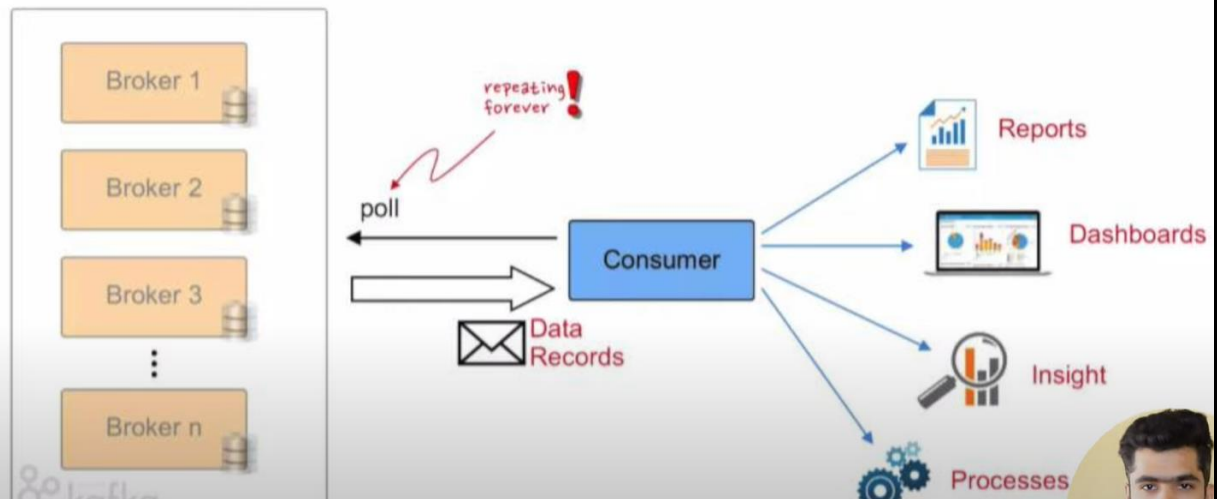


Multiple brokers inside one kafka cluster

Producers



Consumers



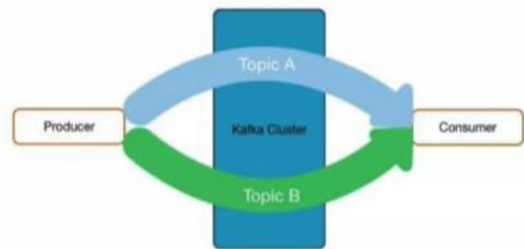
ZooKeeper Basics



- **Open Source** Apache Project
- Distributed **Key Value Store**
- Maintains **configuration information**
- Stores **ACLs** and **Secrets**
- Enables highly reliable **distributed coordination**
- Provides **distributed synchronization**

Topics

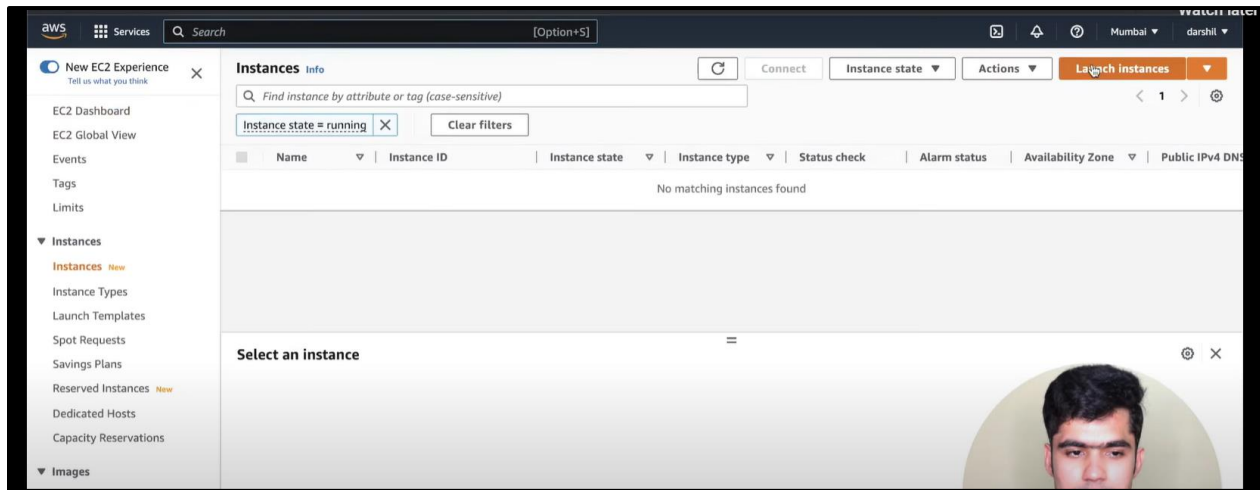
- **Topics:** Streams of “related” Messages in Kafka
 - Is a Logical Representation
 - Categorizes Messages into Groups
- Developers define Topics
- Producer \longleftrightarrow Topic: N to N Relation
- Unlimited Number of Topics



Topic is Basically, logical representation inside the kafka broker

Attach some photos here

Creating a EC2 instance:



Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ RSA
RSA encrypted private and public key pair

☐ ED25519
ED25519 encrypted private and public key pair

Private key file format


☒ .pem
For use with OpenSSH

☐ .ppk
For use with PuTTY

CancelCreate key pair

Launch the instance:

[EC2](#) > [Instances](#) > Launch an instance

 **Success**
Successfully initiated launch of instance ([i-08ddab2258427399a](#))

[▶ Launch log](#)

Next Steps

< 1 2 3 4 5 6 >

[EC2](#) > [Instances](#) > [i-08ddab2258427399a](#) > Connect to instance

Connect to instance [Info](#)


Connect to your instance [i-08ddab2258427399a](#) ([kafka-stock-market-ec2](#)) using any of these options

EC2 Instance Connect

Session Manager

SSH client


EC2 serial console

Instance ID
 [i-08ddab2258427399a](#) ([kafka-stock-market-ec2](#))


Connection Type

☒ **Connect using EC2 Instance Connect**
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

☐ **Connect using EC2 Instance Connect Endpoint**
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address
 16.16.58.181

User name
Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, `ec2-user`.

 **Note:** In most cases, the default user name `ec2-user` is correct. However, read your AMI usage instructions to

Here to connect from our local cmd we need ssh instance details: as given below

EC2 > Instances > i-08ddab2258427399a > Connect to instance

Connect to instance Info

Connect to your instance i-08ddab2258427399a (kafka-stock-market-ec2) using any of these options


EC2 Instance Connect

Session Manager


SSH client


EC2 serial console


Instance ID


 i-08ddab2258427399a (kafka-stock-market-ec2)


1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is kafka-stock-market-project-key.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.


 `chmod 400 kafka-stock-market-project-key.pem`
4. Connect to your instance using its Public DNS:

 `ec2-16-16-58-181.eu-north-1.compute.amazonaws.com`

 Command copied

 `ssh -i "kafka-stock-market-project-key.pem" ec2-user@ec2-16-16-58-181.eu-north-1.compute.amazonaws.com`

 **Note:** In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.



```
Microsoft Windows [Version 10.0.19045.3086]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Futurense>cd Documents/

C:\Users\Futurense\Documents>ls -ltr
'ls' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Futurense\Documents>ls
'ls' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Futurense\Documents>ssh -i "kafka-stock-market-project-key.pem" ec2-user@ec2-16-16-58-181.eu-north-1.compute.amazonaws.com
```

Get connected to our EC2 instance:

[illegible]

Download the kafka cluster inside our EC2 machine:

```
C:\Users\Futuretense\Documents>kafka>ssh -i "kafka-stock-market-project-key.pem" ec2-user@ec2-16-16-58-181.eu-north-1.compute.amazonaws.com
```

```
#  
##### Amazon Linux 2023  
#####  
#####|  
##/#/  
V# '->  
m/'
```

```
Last login: Sat Aug 5 15:14:40 2023 from 122.171.17.148  
[ec2-user@ip-172-31-22-150 ~]$ wget https://downloads.apache.org/kafka/3.5.1/kafka_2.12-3.5.1.tgz  
--2023-08-05 15:18:29-- https://downloads.apache.org/kafka/3.5.1/kafka_2.12-3.5.1.tgz  
Resolving downloads.apache.org (downloads.apache.org)... 135.181.214.104, 88.99.95.219, 2a01:4f8:10a:201a::2, ...  
Connecting to downloads.apache.org (downloads.apache.org)[135.181.214.104]:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 106956505 (102M) [application/x-gzip]  
Saving to: 'kafka_2.12-3.5.1.tgz'  
  
kafka_2.12-3.5.1.tgz      100%[=====] 102.00M   85.8MB/s    in 1.2s  
  
2023-08-05 15:18:30 (85.8 MB/s) - 'kafka_2.12-3.5.1.tgz' saved [106956505/106956505]  
  
[ec2-user@ip-172-31-22-150 ~]$
```

Now uncompress it: `tar -xvf kafka_2.12-3.3.1.tgz`

```
[ec2-user@ip-172-31-22-150 ~]$ ls
kafka_2.12-3.5.1  kafka_2.12-3.5.1.tgz
[ec2-user@ip-172-31-22-150 ~]$
```

Apache Kafka will run on the top of the JVM (Java Virtual Machine):

Install java

```
[ec2-user@ip-172-31-22-150 ~]$ sudo yum install java
Last metadata expiration check: 0:42:52 ago on Sat Aug 5 14:51:56 2023.
Dependencies resolved.
=====
Package                                Architecture      Version           Repository        Size
=====
Installing:
  java-17-amazon-corretto              x86_64            1:17.0.8+7-1.amzn2023.1  amazonlinux      188 k
Installing dependencies:
=====
```

```
[ec2-user@ip-172-31-22-150 ~]$ java --version
openjdk 17.0.8 2023-07-18 LTS
OpenJDK Runtime Environment Corretto-17.0.8.7.1 (build 17.0.8+7-LTS)
OpenJDK 64-Bit Server VM Corretto-17.0.8.7.1 (build 17.0.8+7-LTS, mixed mode, sharing)
[ec2-user@ip-172-31-22-150 ~]$
```


Now start the Zookeeper in kafka cluster:

```
ec2-user@ip-172-31-22-150:~/kafka_2.12-3.5.1
[ec2-user@ip-172-31-22-150 ~]$
[ec2-user@ip-172-31-22-150 ~]$ cd kafka_2.12-3.5.1/
[ec2-user@ip-172-31-22-150 kafka_2.12-3.5.1]$ bin/zookeeper-server-start.sh config/zookeeper.properties
```

ssh -i "kafka-stock-market-project-key.pem" ec2-user@ec2-16-16-58-181.eu-north-1.compute.amazonaws.com

Increase the memory of th kafka server:

```
export KAFKA_HEAP_OPTS="-Xmx256M -Xms128M"
```

Now run the kafka server after increasing the memory:

```
[ec2-user@ip-172-31-22-150 ~]$ cd kafka_2.12-3.5.1/
[ec2-user@ip-172-31-22-150 kafka_2.12-3.5.1]$ bin/kafka-server-start.sh config/server.properties
[2023-08-05 16:46:34,524] INFO Registered kafka:type=kafka.Log4jController MBean (kafka.utils.Log4jControllerRegistration$)
[2023-08-05 16:46:35,018] INFO Setting -Djdk.tls.rejectClientInitiatedRenegotiation=true to disable client-initiated TLS renegotiation (org.apache.zookeeper.common.X509Util)
[2023-08-05 16:46:35,136] INFO Registered signal handlers for TERM, INT, HUP (org.apache.kafka.common.utils.LoggingSignalHandler)
```

Here we can not access the kafka server as it is a private address

EC2 > Instances > i-08ddab2258427399a

Instance summary for i-08ddab2258427399a (kafka-stock-market-ec2) [Info](#)

Updated less than a minute ago

Instance ID i-08ddab2258427399a (kafka-stock-market-ec2)	Public IPv4 address 16.16.58.181 open address
IPv6 address -	Instance state Running
Hostname type IP name: ip-172-31-22-150.eu-north-1.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-22-150.eu-north-1.compute.internal
Answer private resource DNS name IPv4 (A)	Instance type t3.micro
Auto-assigned IP address 16.16.58.181 [Public IP]	VPC ID vpc-05ccf5e1b3ab03ec7
IAM Role	Subnet ID

To access the server we need to change that to public ip

So, first stop the servers both zookeeper and kafka server

Now in the ec2 instance: run the below command

sudo nano config/server.properties

Change the configuration according to below:

EC2 > Instances > i-08ddab2258427399a

Instance summary for i-08ddab2258427399a (kafka-stock-market-ec2) [Info](#)

Updated less than a minute ago

Instance ID i-08ddab2258427399a (kafka-stock-market-ec2)	Public IPv4 address 16.16.58.181 open address
IPv6 address -	Instance state Running

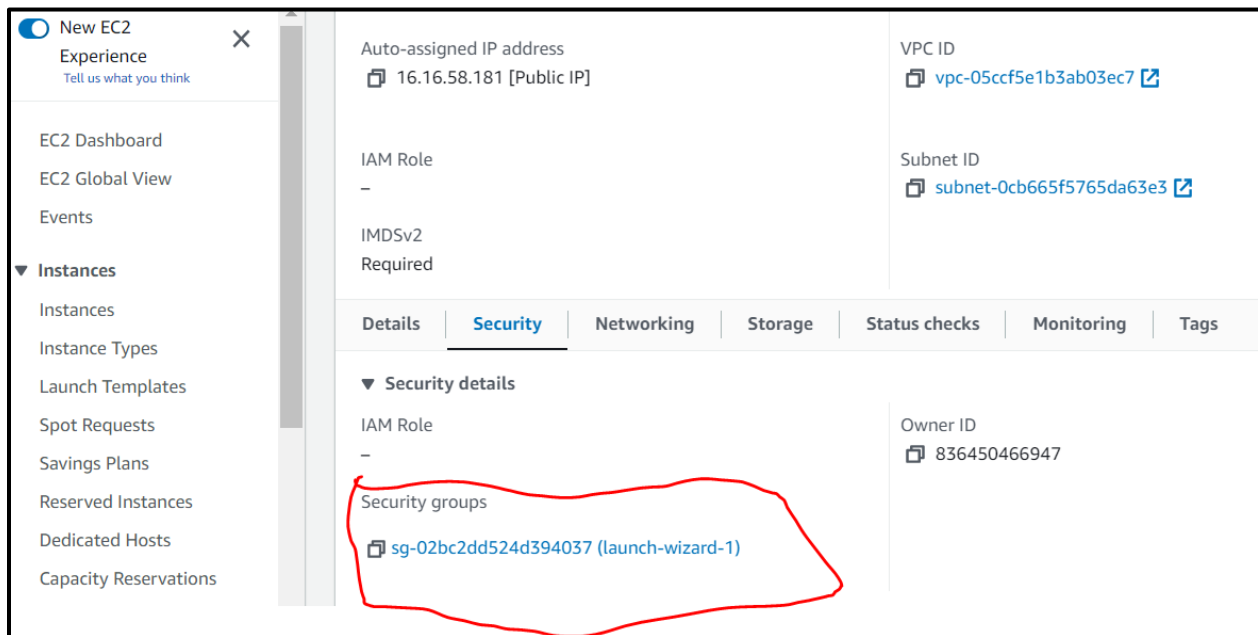
Enter the below the above address

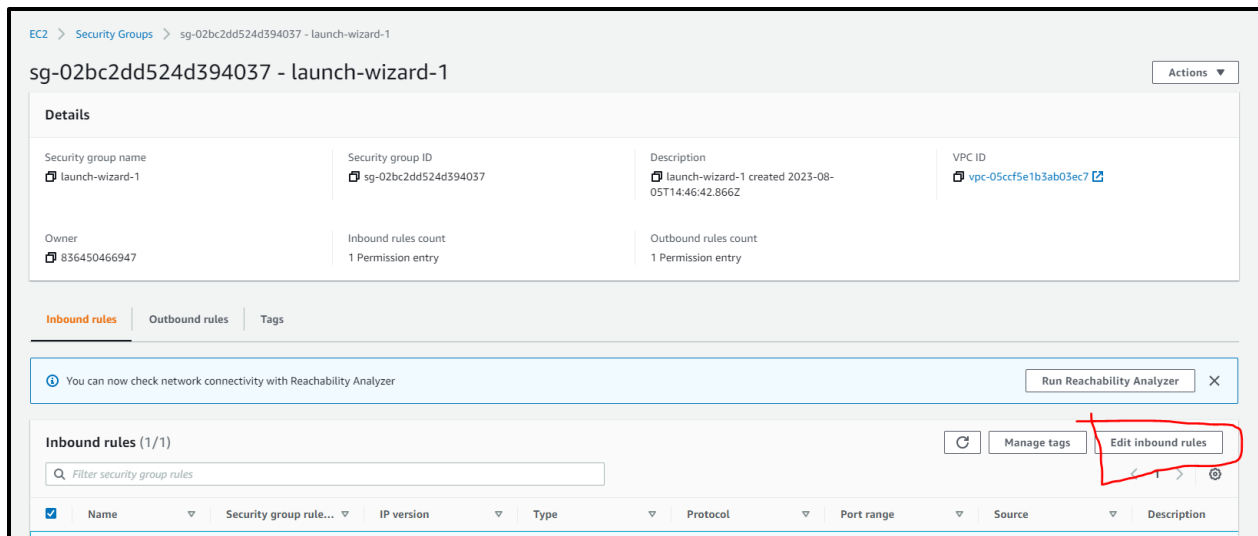
```
EC # Listener name, hostname and port the broker will advertise to clients.  
Ev # If not set, it uses the value for "listeners".  
advertised.listeners=PLAINTEXT://65.2.168.105:9092
```

Now again run the zookeeper server:

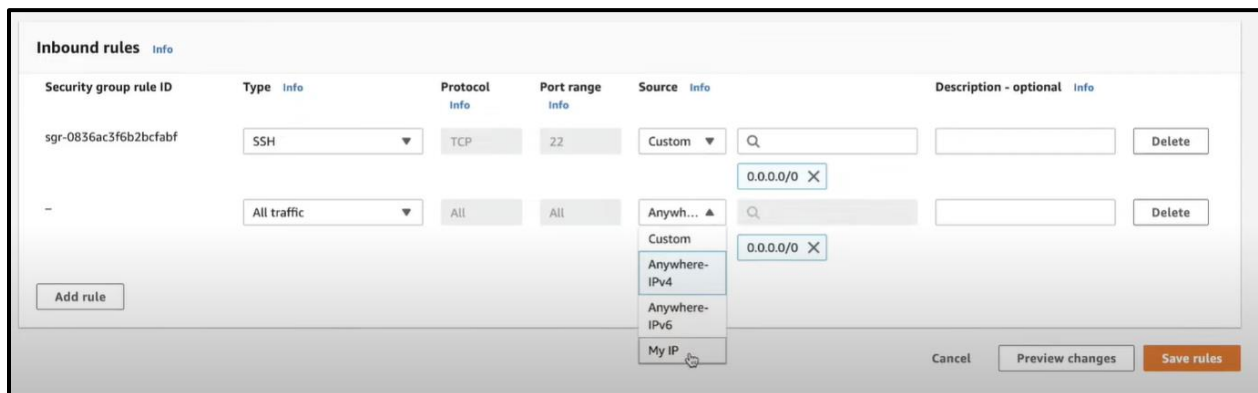
Run the kafka server as well

Now we have to give permission access to our local machine to the ec2 instance



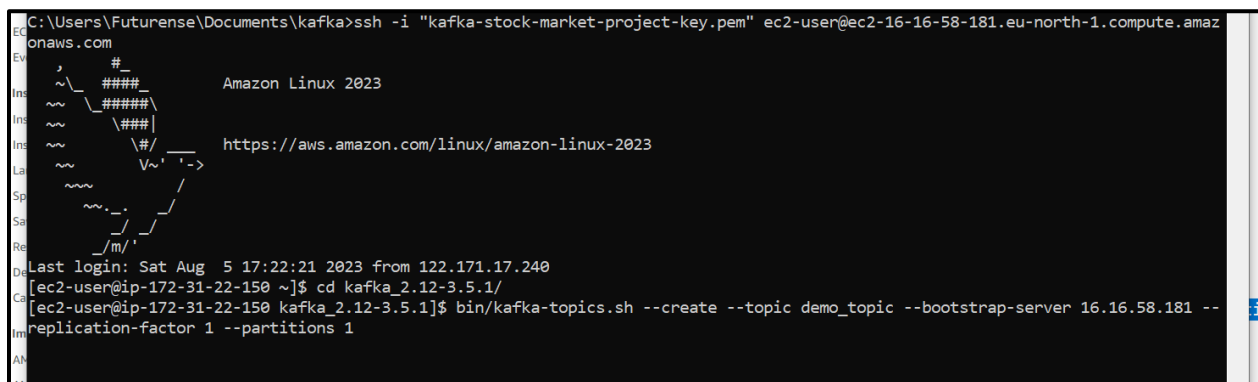


Here select 'Anywhere IPv4' is recommended for this project: as below

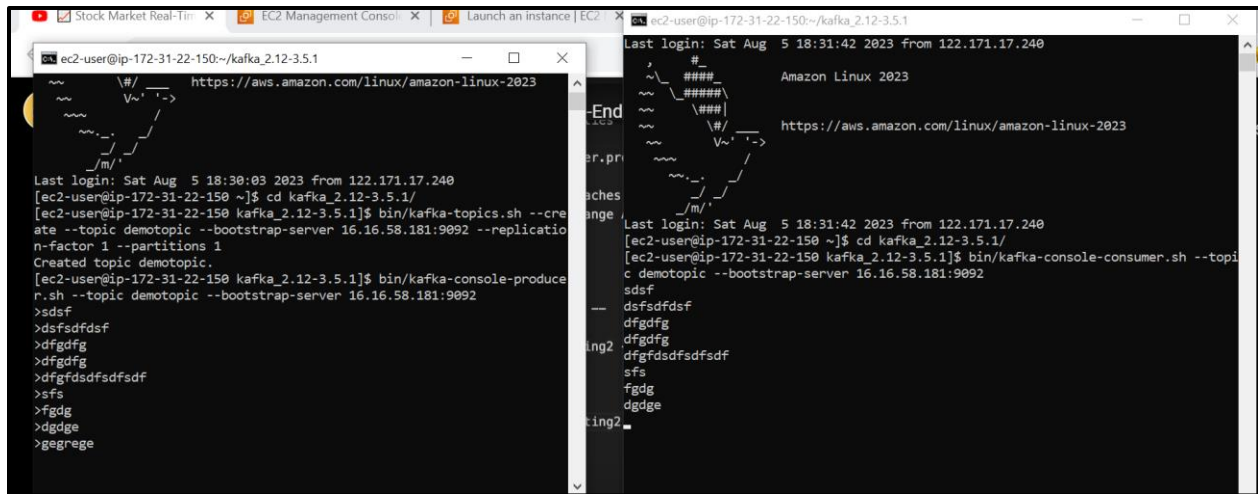


Now we have to make topic, producer and consumer

Create topic inside the kafka:



Create a producer in another terminal:

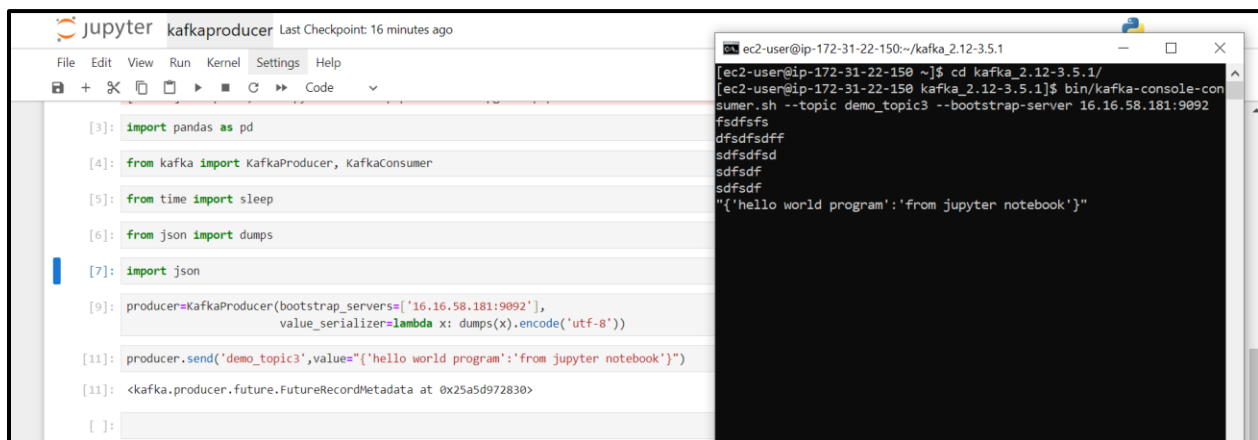


The image shows two terminal windows on an Amazon Linux 2023 EC2 instance. The left window shows the creation of a Kafka topic named 'demotopic' and the use of the 'kafka-console-producer.sh' script to send test messages. The right window shows the use of the 'kafka-console-consumer.sh' script to receive the messages.

```
ec2-user@ip-172-31-22-150:~/kafka_2.12-3.5.1
Last login: Sat Aug 5 18:30:03 2023 from 122.171.17.240
[ec2-user@ip-172-31-22-150 ~]$ cd kafka_2.12-3.5.1/
[ec2-user@ip-172-31-22-150 kafka_2.12-3.5.1]$ bin/kafka-topics.sh --create --topic demotopic --bootstrap-server 16.16.58.181:9092 --replication-factor 1 --partitions 1
Created topic demotopic.
[ec2-user@ip-172-31-22-150 kafka_2.12-3.5.1]$ bin/kafka-console-producer.sh --topic demotopic --bootstrap-server 16.16.58.181:9092
>sdsf
>sdsfdfsdf
>dfgdfg
>dfgdfg
>dfgdfdsfdfsdf
>sfs
>fgdg
>dgdge
>egege
```

```
ec2-user@ip-172-31-22-150:~/kafka_2.12-3.5.1
Last login: Sat Aug 5 18:31:42 2023 from 122.171.17.240
[ec2-user@ip-172-31-22-150 ~]$ cd kafka_2.12-3.5.1/
[ec2-user@ip-172-31-22-150 kafka_2.12-3.5.1]$ bin/kafka-console-consumer.sh --topic demotopic --bootstrap-server 16.16.58.181:9092
sdsf
sdsfdfsdf
dfgdfg
dfgdfg
dfgdfdsfdfsdf
sfs
fgdg
dgdge
```

Connecting Jupyter notebook with ec2 instance and sending data



The image shows a Jupyter Notebook interface with a code cell containing Python code to create a Kafka producer and send a message. The output of the code is displayed below the cell. A terminal window is also open, showing the Kafka console consumer receiving the message.

```
Jupyter | kafkaproducer | Last Checkpoint: 16 minutes ago
File Edit View Run Kernel Settings Help
[3]: import pandas as pd
[4]: from kafka import KafkaProducer, KafkaConsumer
[5]: from time import sleep
[6]: from json import dumps
[7]: import json
[9]: producer=KafkaProducer(bootstrap_servers=['16.16.58.181:9092'],
    value_serializer=lambda x: dumps(x).encode('utf-8'))
[11]: producer.send('demo_topic3',value={'hello world program':'from jupyter notebook'})
[11]: <kafka.producer.future.FutureRecordMetadata at 0x25a5d972830>
[ ]:
```

```
ec2-user@ip-172-31-22-150:~/kafka_2.12-3.5.1
[ec2-user@ip-172-31-22-150 ~]$ cd kafka_2.12-3.5.1/
[ec2-user@ip-172-31-22-150 kafka_2.12-3.5.1]$ bin/kafka-console-consumer.sh --topic demo_topic3 --bootstrap-server 16.16.58.181:9092
{"hello world program": "from jupyter notebook"}
```

```
File Edit View Run Kernel Settings Help  
+ % □ □ ▶ ■ ⌂ » Code ▾  
  
2 HSI 1987-01-05 2552.399902 2552.399902 2552.399902 2552.399902 2552.399902 0.0  
3 HSI 1987-01-06 2583.899902 2583.899902 2583.899902 2583.899902 2583.899902 0.0  
4 HSI 1987-01-07 2607.100098 2607.100098 2607.100098 2607.100098 2607.100098 0.0  
  
[17]: df.sample(1)  
  
[17]:
```

	Index	Date	Open	High	Low	Close	Adj Close	Volume
44375	N225	1979-02-02	6187.299805	6187.299805	6187.299805	6187.299805	6187.299805	

```
[*]: while True:  
    dict_for=df.sample(1).to_dict(orient='records')[0]  
    producer.send('demo_topics',value=dict_for)  
  
[]:  
  
[]:  
  
[]:
```

Because we are running kafka server in a small machine with one broker and one single partition

Step last:

Services

Search

[Alt+S]

Global

rajender7415

Successfully created bucket "kafka-project-bucket-rajender"

To upload files and folders, or to configure additional bucket settings choose [View details](#).

View details

Amazon S3

>

Buckets

▶ Account snapshot

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

View Storage Lens dashboard

Buckets (1) [Info](#)

Refresh

Copy content

Empty

Delete

Create bucket

Find buckets by name

< 1 > ⚙️

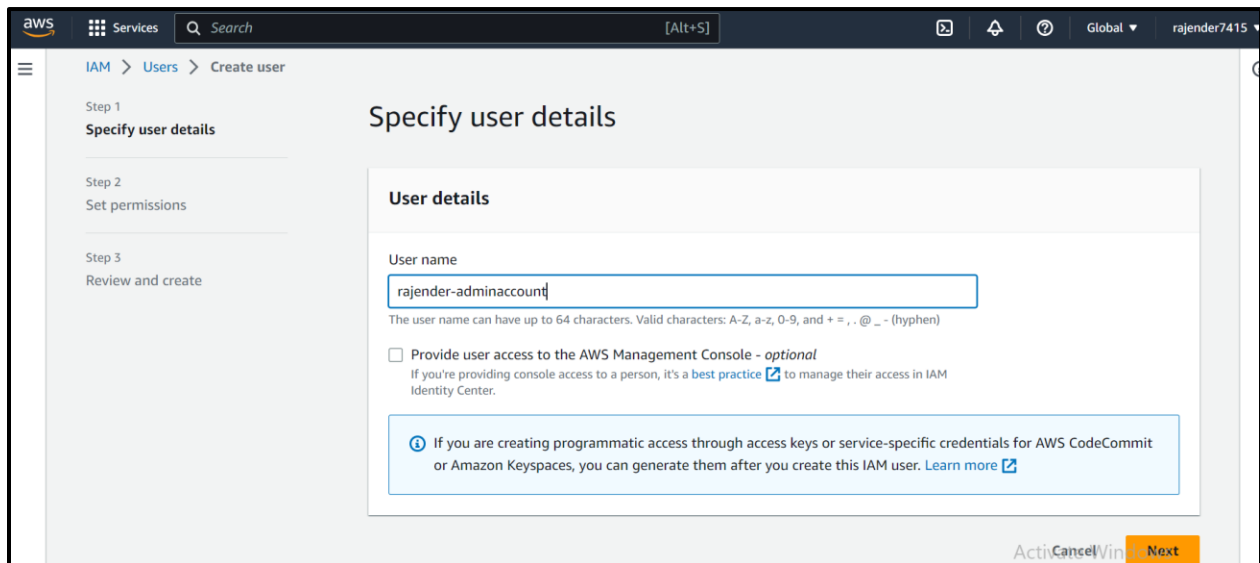
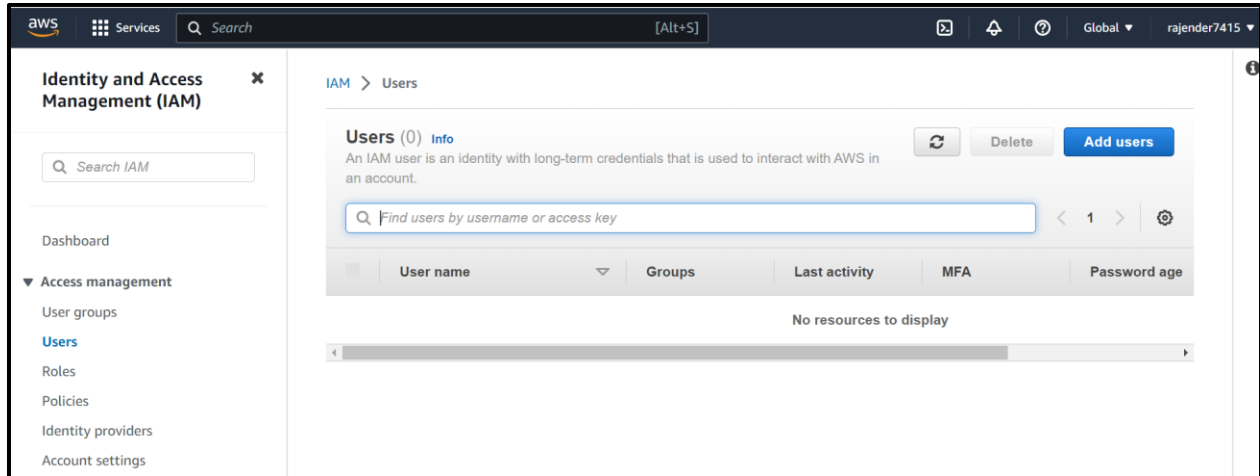
Name	AWS Region	Access	Creation date
<div>○</div> <div>kafka-project-bucket-rajender</div>	EU (Stockholm) eu-north-1	<u>Bucket and objects not public</u>	August 6, 2023, 12:18:34 (UTC+05:30)

And all options should be by default

Now upload csv file data to the S3 bucket from jupyter notebook:

But we need to access to that s3 bucket

Go to IAM > go to users



Give access administrator access and Amazons3FullAccess

Permissions policies (2/1116)

Choose one or more policies to attach to your new user.

Filter by Type

Search: s3 All types 11 matches

	Policy name	Type	At
<input type="checkbox"/>	AmazonDMSRedshiftS3Role	AWS managed	0
<input checked="" type="checkbox"/>	AmazonS3FullAccess	AWS managed	0
<input type="checkbox"/>	AmazonS3ObjectLambdaExecutionRolePolicy	AWS managed	0

Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details

User name kafka-user-project	Console password type None	Require password reset No
---------------------------------	-------------------------------	------------------------------

Permissions summary

Name	Type	Used as
AdministratorAccess	AWS managed - job function	Permissions policy
AmazonS3FullAccess	AWS managed	Permissions policy

Here two things: Access key ID, secret access key

Access key created

This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Retrieve access keys

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIA4FQC...VHSB7AZMT	***** Show

Access key best practices

Act ID

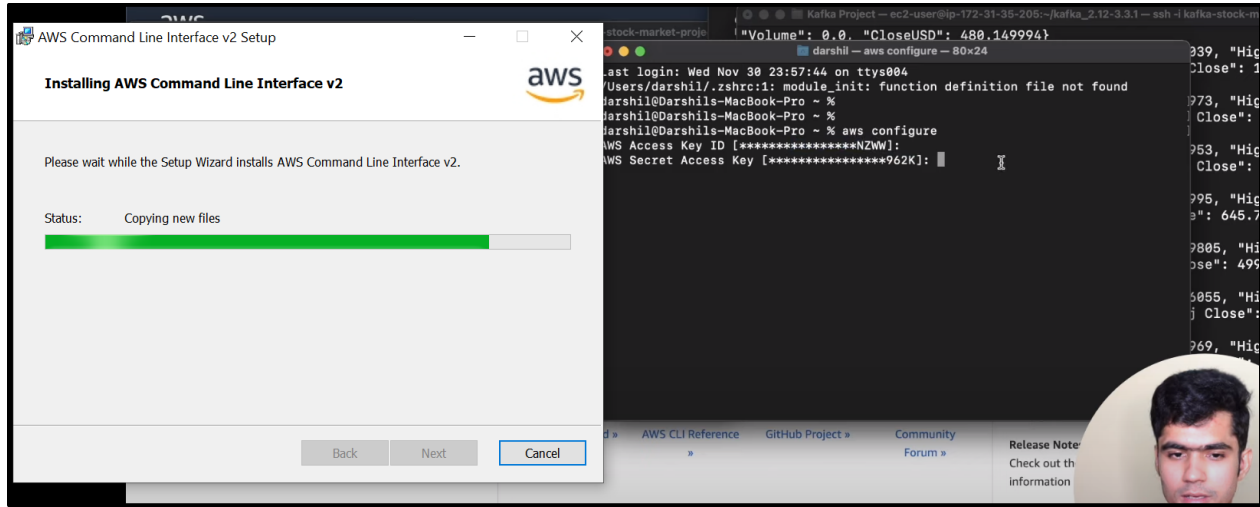
AKIA4FQCVHSBSFB7AZMT

key

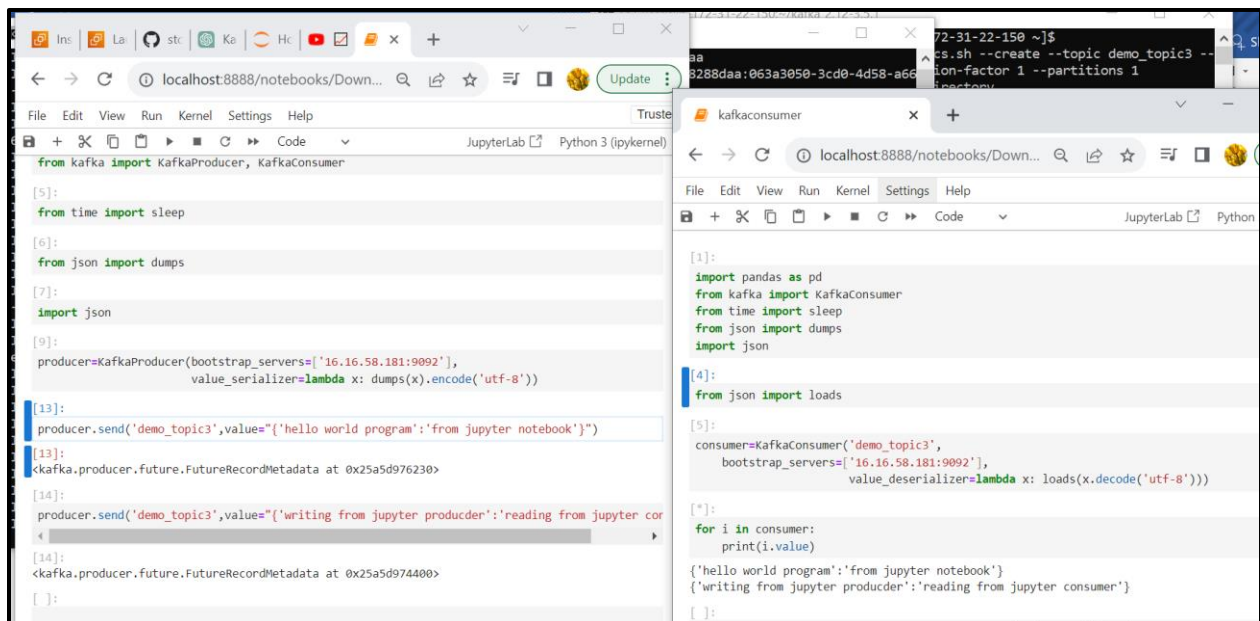
bXGUwPS34Bc4JnLdRjfvfQNNesBTvkIloOrsEwkn

Download the aws command line interface (CLI) from internet and install it:

Now run the cmd:



Doing it in jupyter notebooks (producer and consumer)



In producer:

```
4 HSI 1987-01-07 2607.100098 2607.100098 2607.100098 2607.100098 2607.100098 0.0 338.923013
```

```
In [14]: df.sample(1)
```

```
Out[14]:
```

	Index	Date	Open	High	Low	Close	Adj Close	Volume	CloseUSD
92125	SSMI	2005-11-29	7419.600098	7471.0	7409.600098	7440.700195	7440.700195	46744600.0	8259.177216

```
In [20]: while True:
          dict_for=df.sample(1).to_dict(orient='records')[0]
          producer.send('demotopic5',value=dict_for)
```

In consumer:

```
In [6]: from s3fs import S3FileSystem
        s3=S3FileSystem()
```

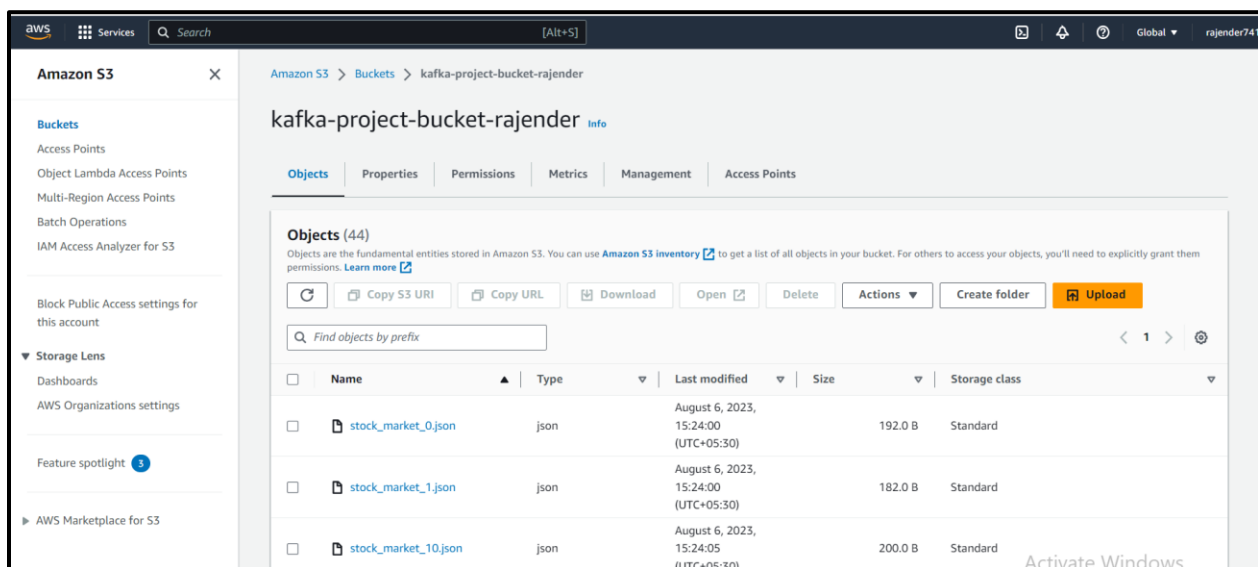
```
In [7]: # for count, i in enumerate(consumer):
        #     print(count)
        #     print(i.value)
```

```
In [*]: for count, i in enumerate(consumer):
        with s3.open("s3://kafka-project-bucket-rajender/stock_market_{}.json".format(count), "w") as file:
            json.dump(i.value, file)
```

```
In [ ]:
```

```
In [ ]:
```

See the output:



Now we have to create the crawler:

- The crawler will crawl the schema of the files
- The crawler helps to query the data using Athena
- It will take the data source so that it can be useful in Athena

The screenshot shows the AWS Glue console interface for creating a new crawler. The left sidebar contains navigation links for various AWS Glue features. The main content area is titled 'Choose data sources and classifiers' and includes a progress bar with five steps. Step 2, 'Choose data sources and classifiers', is the active step. It contains a 'Data source configuration' section with a radio button for 'Not yet' (selected) and a 'Yes' option. Below this is a 'Data sources (1)' table with columns for Type, Data source, and Parameters. The table shows one source: S3 with the path s3://kafka-project-bucket-rajender/ and the parameter Recrawl all. There is also a section for 'Custom classifiers - optional'.

AWS Glue **×**

AWS Glue > Crawlers > Add crawler

Step 1
Set crawler properties

Step 2
Choose data sources and classifiers

Step 3
Configure security settings

Step 4
Set output and scheduling

Step 5
Review and create

Choose data sources and classifiers

Data source configuration

Is your data already mapped to Glue tables?

☒ Not yet
Select one or more data sources to be crawled.

☐ Yes
Select existing tables from your Glue Data Catalog.

Data sources (1) [info](#)

The list of data sources to be scanned by the crawler.

Type	Data source	Parameters
<input type="radio"/> S3	s3://kafka-project-bucket-rajender/	Recrawl all

Custom classifiers - optional
A classifier checks whether a given file is in a format the crawler can handle. If it is, the classifier creates a schema in the form of a StructType object that matches that data format.

Cancel Previous **Next**

Now during the creation of the crawler, it will ask for the role - - - - which is glue/crawler wants to talk to s3 we need role

Create a role inside IAM:

The screenshot shows the AWS IAM console interface for creating a new role. The left sidebar contains navigation links for various AWS IAM features. The main content area is titled 'Create role' and includes a progress bar with three steps. Step 1, 'Select trusted entity', is the active step. It contains a 'Select trusted entity' section with a 'Trusted entity type' dropdown menu. The dropdown menu is open, showing options: AWS service (selected), AWS account, Web identity, SAML 2.0 federation, and Custom trust policy. Below the dropdown is a 'Use case' section with a 'Common use cases' dropdown menu. The dropdown menu is open, showing options: EC2, Lambda, and Glue (selected). There is also a 'Use cases for other AWS services' section with a dropdown menu.

Identity and Access Management (IAM) **×**

IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Select trusted entity [info](#)

Trusted entity type

☒ **AWS service**
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

☐ **AWS account**
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

☐ **Web identity**
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

☐ **SAML 2.0 federation**
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

☐ **Custom trust policy**
Create a custom trust policy to enable others to perform actions in this account.

Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Common use cases

☐ **EC2**
Allows EC2 instances to call AWS services on your behalf.

☐ **Lambda**
Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

Glue

☒ **Glue**
Allows Glue to call AWS services on your behalf.

Activate Windows
Go to Settings to activate Windows

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

Access reports

Access analyzer

Archive rules

Analizers

Settings

Credential report

Organization activity

Service control policies (SCPs)

Related consoles

IAM > Roles > Create role

Step 1

Select trusted entity

Step 2

Add permissions

Step 3

Name, review, and create

Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

aws-glue-role-access

Maximum 64 characters. Use alphanumeric and "+", "@", "_" characters.

Description

Add a short explanation for this role.

Allows Glue to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and "+", "@", "_" characters.

Step 1: Select trusted entities

Edit

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Principal": {
7         "Service": "glue.amazonaws.com"
8       },
9       "Action": "sts:AssumeRole"
10     }
11   ]
12 }
```

Create a database for the crawler:

AWS Glue

Getting started

ETL jobs

Visual ETL

Job run monitoring

Data Catalog tables

Data connections

Workflows (orchestration)

Data Catalog

Databases

Tables

Stream schema registries

Schemas

Connections

Crawlers

Classifiers

Catalog settings

Data Integration and ETL

Legacy pages

What's New

Documentation

AWS Marketplace

Enable compact mode

AWS Glue > Crawlers > Add crawler

Step 1

Set crawler properties

Step 2

Choose data sources and classifiers

Step 3

Configure security settings

Step 4

Set output and scheduling

Step 5

Review and create

Set output and scheduling

Output configuration

Target database

Choose a database

Clear selection

Add database

Target database is required

Table name prefix - optional

Type a prefix added to table names

Maximum table threshold - optional

This field sets the maximum number of tables the crawler is allowed to generate. In the event that this number is surpassed, the crawl will fail with an error. If not set, the crawler will automatically generate the number of tables depending on the data schema.

Type a number greater than 0

Advanced options

Crawler schedule

You can define a time-based schedule for your crawlers and jobs in AWS Glue. The definition of these schedules uses the Unix-like cron syntax. [Learn more](#)

Frequency

On demand

Cancel

Previous

Next

Crawler should be ready before querying:

AWS

Services

Search

[Alt+S]

Stockholm

rajender/7415

AWS Glue

Getting started

ETL jobs

Visual ETL

Job run monitoring

Data Catalog tables

Data connections

Workflows (orchestration)

Data Catalog

Databases

Tables

Stream schema registries

Schemas

Connections

Crawlers

Classifiers

One crawler successfully created

The following crawler is now created: "new-crawler-kafka-project"

AWS Glue > Crawlers

Crawlers

A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

Crawlers (1)

View and manage all available crawlers.

Last updated (UTC)

August 6, 2023 at 10:19:44

Action

Run

Create crawler

Filter crawlers

< 1 >

<input type="checkbox"/>	Name	State	Schedule	Last run	Last run timestamp	Log	Table changes from last r...
<input type="checkbox"/>	new-crawler-kafka-project	Ready	-	-	-	-	-

AWS Glue > Crawlers

Crawlers

A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

Crawlers (1/1) [Info](#)

View and manage all available crawlers.

Last updated (UTC)
August 6, 2023 at 10:22:17

[Refresh](#) [Action](#) [Run](#) [Create crawler](#)

<input checked="" type="checkbox"/>	Name	State	Schedule	Last run	Last run timestamp	Log	Table changes from last r...
<input checked="" type="checkbox"/>	new-crawler-kafka-project	Ready		Succeeded	August 6, 2023 at 10:20:17	View log	1 created

Now go to Athena- - - - here we can see the database and we can start querying

eu-north-1.console.aws.amazon.com/athena/home?region=eu-north-1#/query-editor

Amazon Athena

Query editor

Jobs
Workflows
Powered by Step Functions

Administration
Workgroups
Capacity reservations [New](#)
Data sources

Turn on compact mode

Before you run your first query, you need to set up a query result location in Amazon S3. [Edit settings](#)

Athena now supports typeahead code suggestions to speed up SQL query development
Typeahead suggestions are turned on by default. You can change this setting in query editor preferences. [Edit preferences](#)

Data

Data source: AwsDataCatalog

Database: stock_market_kafka

Tables and views

Tables (1)
kafka_project_bucket_rajender

Views (0)

Query 1

Run Query
Preview Table
Generate table DDL
Insert
Insert into editor
Manage
Delete table
View properties
View in Glue

SQL Ln 1, Col 1

Run Explain Cancel Clear Create

Reuse query results up to 60 minutes ago

Query results

Results

Search rows

Activate Windows
Go to Settings to activate Windows.

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

ENG 3:53 PM 8/6/2023

Creating a temp bucket for output storage:

Make sure the temp bucket in the same region

Amazon Athena > Query editor > Manage settings

Manage settings

Query result location and encryption

Location of query result - *optional*
Enter an S3 prefix in the current region where the query result will be saved as an object.

View

Browse S3

You can create and manage lifecycle rules for this bucket
Use Amazon S3 lifecycle rules to store your query results and metadata cost effectively or to delete them after a period of time.
[Learn more](#)

Lifecycle configuration

Expected bucket owner - *optional*

Specify the AWS account ID that you expect to be the owner of your query results output location bucket.

☐ Assign bucket owner full control over query results
Enabling this option grants the owner of the S3 query results bucket full control over the query results. This means that if your query result location is owned by another account, you grant full control over your query results to the other account.

☐ Encrypt query results

Cancel

Save

Amazon Athena

Amazon Athena > Query editor

Editor | Recent queries | Saved queries | **Settings**

Workgroup: primary

Query result and encryption settings

Manage

Query result location and encryption

Query result location	Encrypt query results	Expected bucket owner	Assign bucket owner full control over query results
s3://temp-bucket-for-output-athena/	-	-	Turned off

Amazon Athena ✕

Workgroup query engine upgrade complete
One or more workgroups have been upgraded to Athena engine version 3. To see the workgroups that have been upgraded, use the [Workgroup list page](#). For information about new features, see the [Athena Engine Version Reference](#).

Amazon Athena > Query editor

Editor | Recent queries | Saved queries | Settings

Workgroup: primary

Athena now supports typeahead code suggestions to speed up SQL query development
Typeahead suggestions are turned on by default. You can change this setting in query editor preferences. Edit preferences ✕

Data ↻ <

Data source: AwsDataCatalog

Database: stock_market_kafka

Tables and views Create @

Tables (1) < 1 >

+ kafka_project_bucket_rajender

Views (0) < 1 >

Query 1 ✕ | Query 2 ✕ + ▼

```
1 SELECT * FROM "stock_market_kafka"."kafka_project_bucket_rajender" limit 10;
```

SQL Ln 1, Col 1

Run again Explain Cancel Clear Create

☐ Reuse query results up to 60 minutes ago

Query editor

Jobs
Workflows
Powered by Step Functions

Administration
Workgroups
Capacity reservations [New](#)
Data sources

☐ Turn on compact mode

Run again Explain Cancel Clear Create

☐ Reuse query results up to 60 minutes ago

Query results | Query stats

Completed Time in queue: 151 ms Run time: 557 ms Data scanned: 3.55 KB

Results (10) Copy Download results

< 1 > @

#	index	date	open	high	low	close	adj close	volume	closeusd
1	GDAJI	1994-11-01	2052.219971	2070.030029	2047.609985	2066.179932	2066.179932	0.0	2520.7395170
2	IXIC	1977-05-12	96.959999	96.959999	96.959999	96.959999	96.959999	0.0	96.959999
3	N225	2009-06-15	10126.54981	10126.54981	10029.58984	10039.66992	10039.66992	1.582E8	100.3966992
4	IXIC	1987-06-18	428.100006	429.0	427.399994	429.0	429.0	1.585E8	429.0
5	J203.JO	2018-06-20	56253.30859	57174.55859	56253.30859	56651.66016	56651.66016	0.0	3965.6162112
6	000001.SS	1999-07-27	1605.371948	1607.046997	1580.817017	1590.713013	1590.713013	0.0	254.51408201
7	NVA	1975-12-10	490.619995	490.619995	490.619995	490.619995	490.619995	0.0	490.619995

Now add sleep to the producer:

```
In [*]: while True:
    dict_for=df.sample(1).to_dict(orient='records')[0]
    producer.send('demotopic5',value=dict_for)
    sleep(2)
```

Amazon Athena

Query editor

Jobs

Workflows
Powered by Step Functions

Administration

Workgroups

Capacity reservations [New](#)

Data sources

Turn on compact mode

Data

Data source: AwsDataCatalog

Database: stock_market_kafka

Tables and views

Filter tables and views

Tables (1)

kafka_project_bucket_rajender

Views (0)

Query 1 : X Query 2 : X

1 SELECT count(*) FROM "stock_market_kafka"."kafka_project_bucket_rajender";

SQL Ln 1, Col 74

Run again Explain Cancel Clear Create

Reuse query results up to 60 minutes ago

Query results Query stats

Completed Time in queue: 144 ms Run time: 996 ms Data scanned: 264.09 KB

Results (1)

Search rows

Copy Download results

_col0

1 1407

Activate Windows
Go to Settings to activate Windows

Amazon Athena

Query editor

Jobs

Workflows
Powered by Step Functions

Administration

Workgroups

Capacity reservations [New](#)

Data sources

Turn on compact mode

Data

Data source: AwsDataCatalog

Database: stock_market_kafka

Tables and views

Filter tables and views

Tables (1)

kafka_project_bucket_rajender

Views (0)

Query 1 : X Query 2 : X

1 SELECT count(*) FROM "stock_market_kafka"."kafka_project_bucket_rajender";

SQL Ln 1, Col 74

Run again Explain Cancel Clear Create

Reuse query results up to 60 minutes ago

Query results Query stats

Completed Time in queue: 160 ms Run time: 1.289 sec Data scanned: 271.18 KB

Results (1)

Search rows

Copy Download results

_col0

1 1445

Activate Windows
Go to Settings to activate Windows

See the number of files keep getting increased

