* Created AWS EC2 instance.
* Start the EC2 instance using the command after open the file location.
* ssh -i "kafka\_stock\_market.pem" ec2-user@ec2-52-15-147-25.us-east-2.compute.amazonaws.com
* Installed kafka server into it using the command.

wget <https://downloads.apache.org/kafka/3.6.2/kafka_2.12-3.6.2.tgz>

* Unzipped the files using

tar -xvf kafka\_2.12-3.6.2.tgz

* Installing Java
* Sudo yum install java
* Started the zooperkeerper using the command below.
* bin/zookeeper-server-start.sh config/zookeeper.properties
* Start the EC2 instance on the new terminal while keeping the Zookeeper running in one terminal. ( Follow same step as step 2).
* Adding more space for the kafka server.
* export KAFKA\_HEAP\_OPTS="-Xmx256M -Xms128M"
* Navigate to the location of the Kafka folder using ls command and the cd kafka\_2.12-3.6.2
* Start the kafka server using the below command.
* bin/kafka-server-start.sh config/server.properties
* Now we need to change the ip from private DNS to public DNS after closing both the zookeeper and the kafka server.
* We cannot access the private the DNS from our local so we need to make the EC2 instance to public
* Use the below command to view the config properties.
* sudo nano config/server.properties
* Change the your host name with the public ip as shown below.
* advertised.listeners=PLAINTEXT://your-host-name:9092
* advertised.listeners=PLAINTEXT://52.15.147.25:9092
* Press Ctrl +X and Yes to save it.
* Now start the zookeeper and start the kafka server again.
* If the zookeeper is not working and raising error that a process is already running. We need to first identify the process id of the event using the command. The default port no for zookeeper is 2181.
* sudo lsof -i :2181 to find the process ID.
* Now we have to kill the process using the PID( process ID).
* sudo kill -9 28079
* Now start the zookeeper again. Then the kakfa server again if it is refusing the connection kill the existing process in a similar way and start running both of them.
* We need to provide the security access form our local machine.
* Go to security in the EC2 instance and click edit in inbound rules and then we need to allow the request from our machine
* A screenshot of a computer

  Description automatically generated
* Click on add rule allow type to be All traffic and source to be Anywhere – Ipv4
* A screenshot of a computer

  Description automatically generated
* Now we need to start creation the topic. For this open a new terminal and connect to the EC2 and navigate to the kafka directory.
* Create topic using the below command by adding the topic name and changing the IP name.
* bin/kafka-topics.sh --create --topic demo\_test --bootstrap-server 3.21.244.255:9092 --replication-factor 1 --partitions 1
* Create a producer with the same topic name.
* bin/kafka-console-producer.sh --topic demo\_test --bootstrap-server 3.21.244.255:9092
* Open a new terminal for consumer and connect to EC2 and open kafka directory and create a consumer.
* bin/kafka-console-consumer.sh --topic demo\_test --bootstrap-server 3.21.244.255:9092
* A screenshot of a computer screen

  Description automatically generated
* Data moving from producer to consumer through the kafka server.
* Creating code in Google collab
* A screenshot of a computer

  Description automatically generated
* Sending the data from the producer to the consumer and getting the data in real time in the consumer.
* A screenshot of a computer program

  Description automatically generated
* When we are passing the data real time from twitter tweets. We won’t be saving it and sending it through the producer. We are directly sending the real-time data through the producer to the consumer.
* Now we need to create an AWS S3 bucket and write scripts to transfer the data from consumer to the S3 bucket.
* Use s3fs for this purpose for this and now we need to configure s3 bucket to access from our location machine. We need to got to the IAM, create a user with the administrative access and create a secret key.
* Dowload AWS CLI to use the command line interface for windows and open a new terminal and use the command
* aws configure and setup the access to the s3 bucket if you are using jupyter notebook or python script from your local machine.
* You need to install boto3 when you are using Google collab.
* Now start the consumer and and start it and then run the producer for few seconds and you can see the data being updated in real time in the S3 bucket.
* A screenshot of a computer

  Description automatically generated

Now create a crawler and choose our specific s3 bucket as the source. As below

s3://kafka-tweets-data/

* Now we create a new role using the IAM for the Glue crawler to access the data in s3 all other aws services for that we give administrative access by creating a new role using the IAM for the Glue to access all with a name glue-admin-access and keep it as the IAM role in the Glue crawler.
* Now we need to create a database using the add database and keep a name and add it in the AWS Glue crawler and then run the crawler.
* Now we can launch the AWS athena to query the results from the s3 for that first we can to add a location to store the query results by editing the settings unless we won’t be able to see it. You can I either create a new s3 bucket or just use the same bucket.
* A screenshot of a computer

  Description automatically generated
* Now we can query the results below and work on it.
* A screenshot of a computer

  Description automatically generated