CSYE 7245 - Big Data Sys and Int Analytics

**Lab 3 - Apache Kafka**

**Information:**

|  |  |
| --- | --- |
| **Name** | **NUID** |
| Huỳnh Nhật Huy | 0829660378 |

**Lab Completion Date: 5th February’21**

# About

This lab demonstrates leveraging and implementing Kafka services for static data alongwith real-timeTwitter streaming.



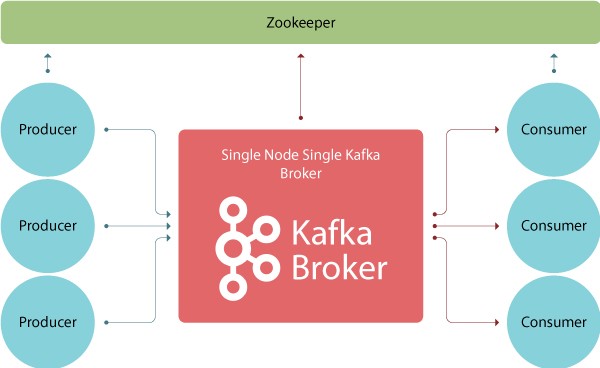
Apache Kafka is a streaming message platform. It is a publish-subscribe based durable messaging system. Kafka is designed to be high performance, highly available, and redundant. It is used to collect, process, store, and integrate data at scale. A messaging system sends messages between processes, applications, and servers.

It’s basic use cases includes: Stream Processing Messaging

Website Activity Tracking Log aggregation

Event Sourcing

Application health monitoring



These are four main parts in a Kafka system:

* **Broker**: Handles all requests from clients (producer, consumer and metadata) and keeps data replicated within the cluster. There can be one or more brokers in a cluster
* **Zookeeper**: Keeps track of status of the Kafka clusters (brokers, topics, users)
* **Producer**: Sends records to a broker
* **Consumer**: Consumes batches of records from the broker

# Experiment setup

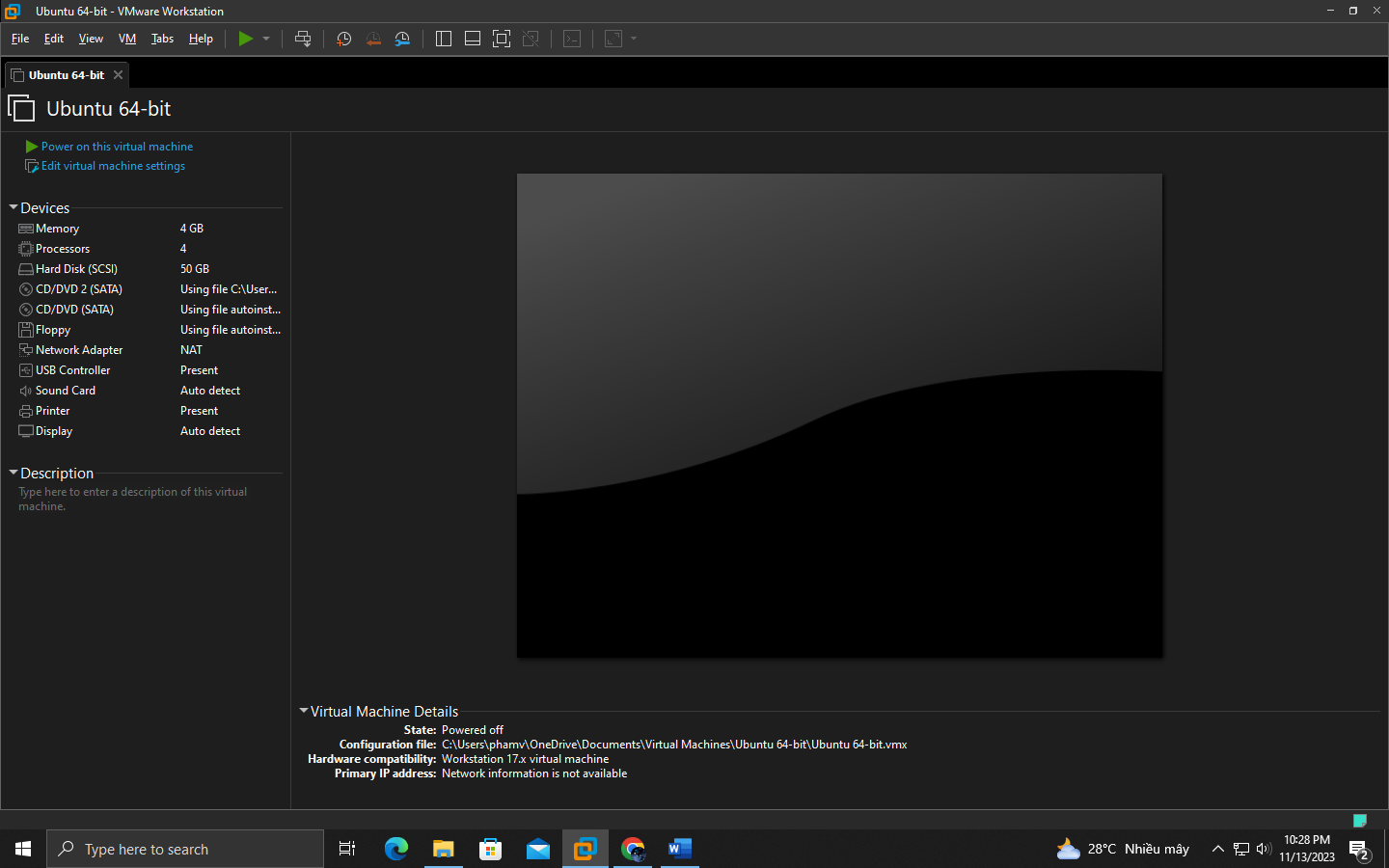
**Prerequisites:**

1. Installing Vmware Workstation Pro

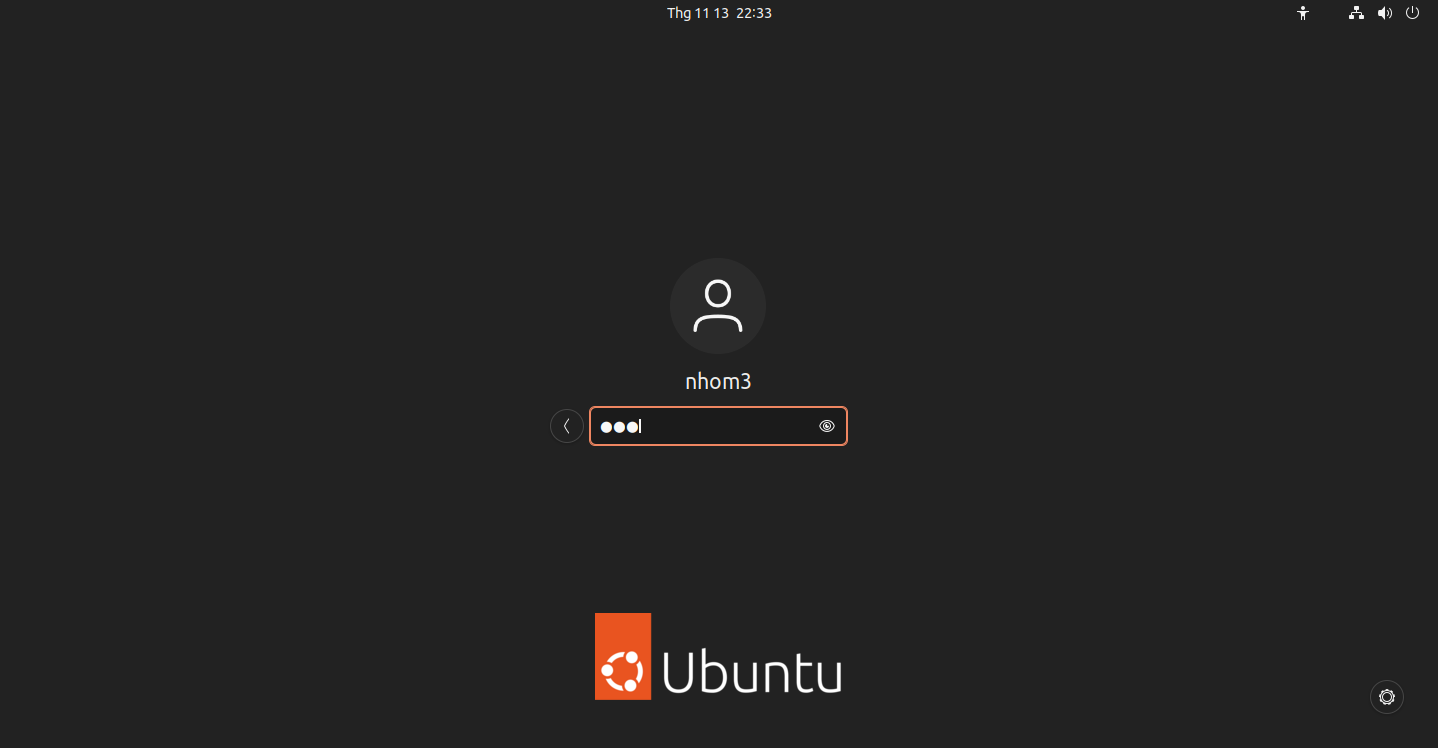
**Specifications:**

* 4 GB RAM
* 50 GB Hard Drive
* Downloading ubuntu iso file

**Vmware Workstation Pro Devices**



**Login Page**



**Requirements:**

## Installing Ubuntu Guest Edition

sudo apt install build-essential dkms linux-headers-$(uname -r)

Able to copy/paste the contents easily Full screen mode available

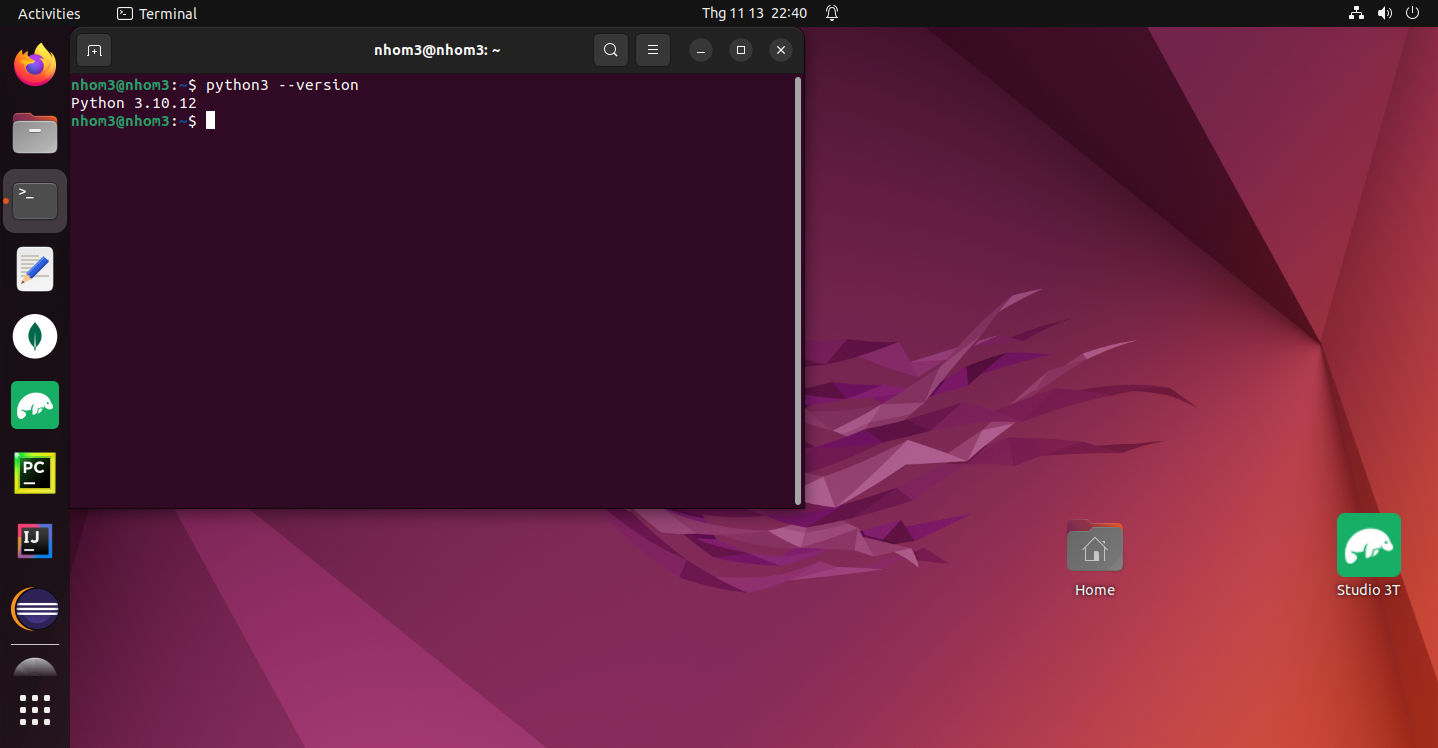
Certain in-built headers/packages available for additional functionalities

## Installing Python

Installing the latest version of Python

sudo apt install python3

sudo apt install python3-pip python3 --version



## Installing AWS CLI

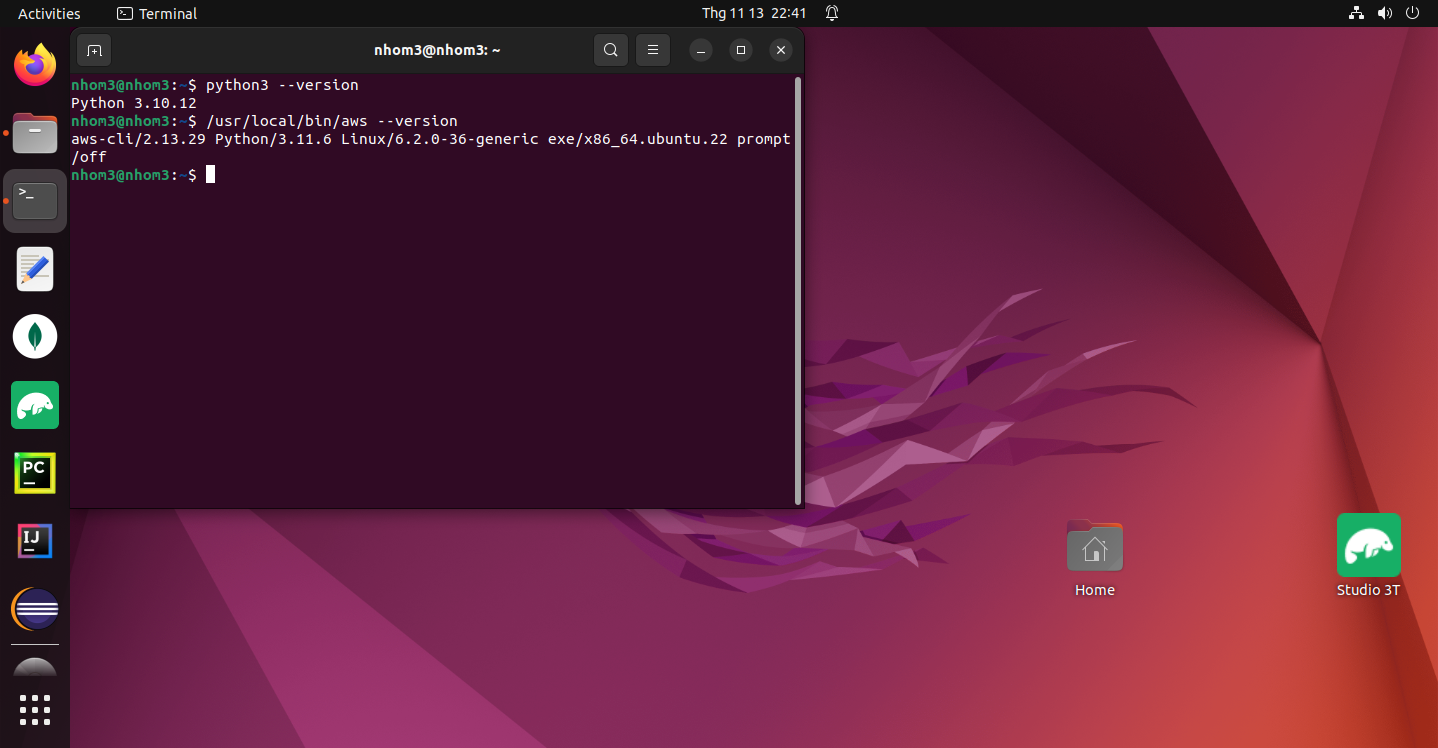
AWS CLI helps to access multiple AWS services and functionalities from the command line.

sudo apt install curl

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

unzip awscliv2.zip sudo ./aws/install

/usr/local/bin/aws –version



## Connecting with AWS

Connecting the server with AWS account by entering the Access and Secret keys

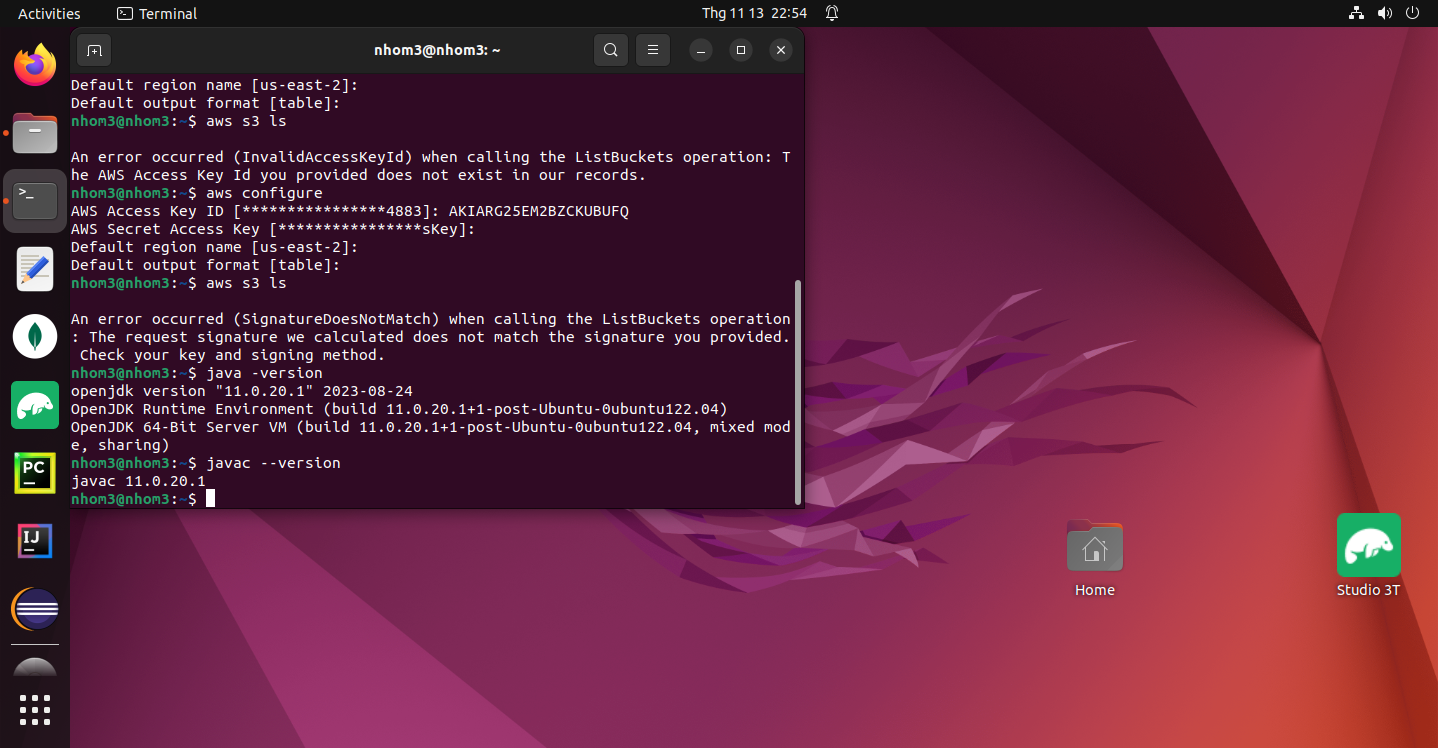
aws configure aws s3 ls

## Installing Java jdk

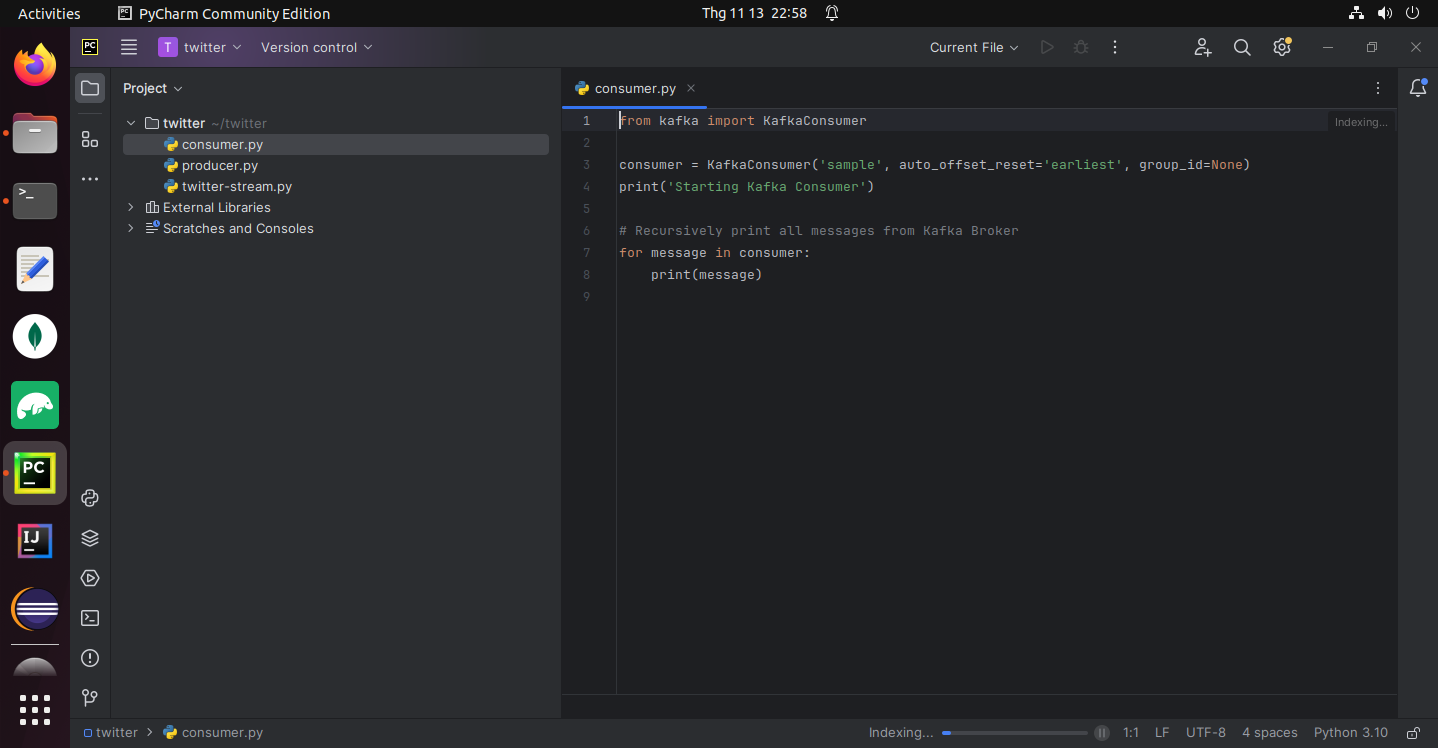
Java jdk is required for starting the Kafka broker and services

sudo apt update sudo apt list

sudo apt install default-jre sudo apt install default-jdk javac --version



* 1. **Installing Pycharm in Ubuntu**

****

# Test Results

## Installing Kafka

Download Apache Kafka from [here](https://kafka.apache.org/downloads)

Unzip Kafka binaries by using tar -xzvf pip3 install kafka-python

## Starting the Zookeeper service and Kafka broker

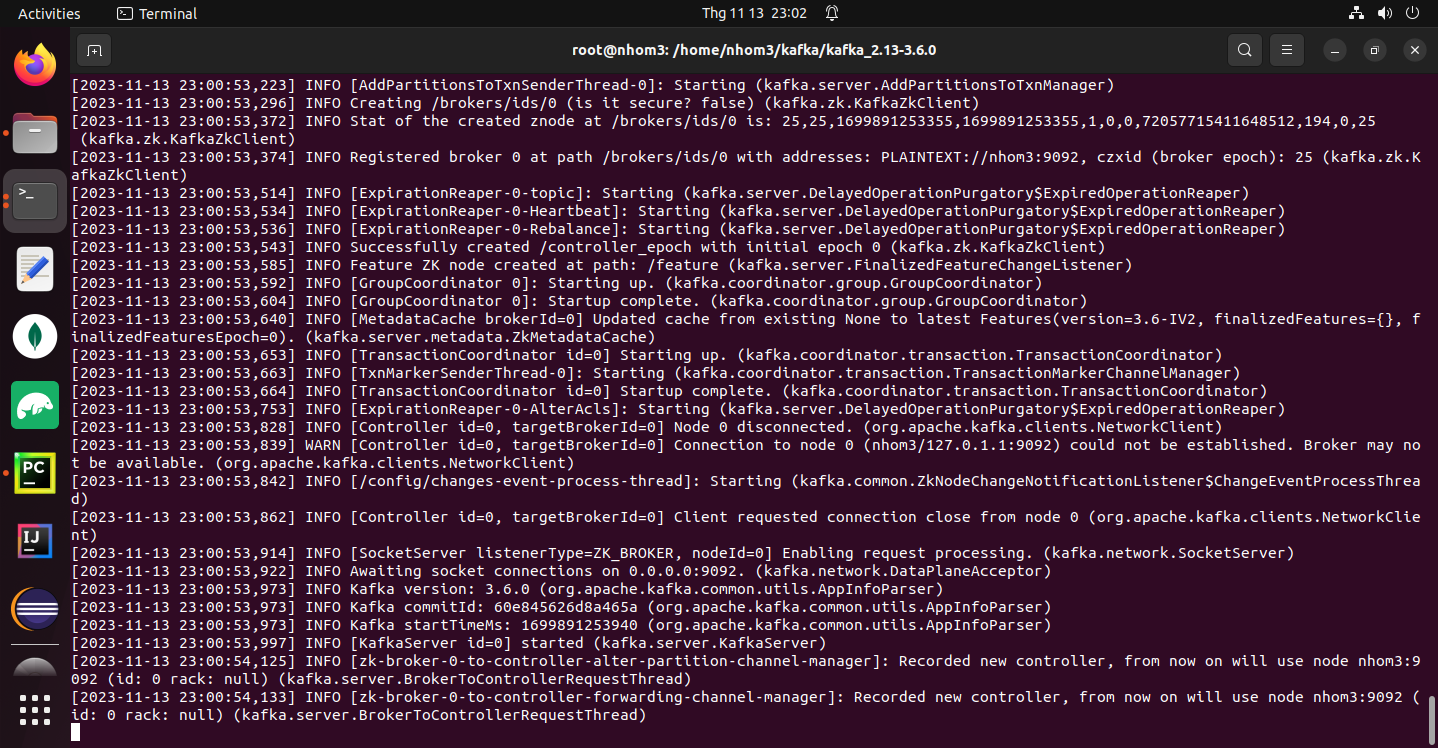
Navigate to the directory where the downloaded files are unzipped and start the Zookeeper service

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



Start the Kafka broker in a new terminal

bin/kafka-server-start.sh config/server.properties

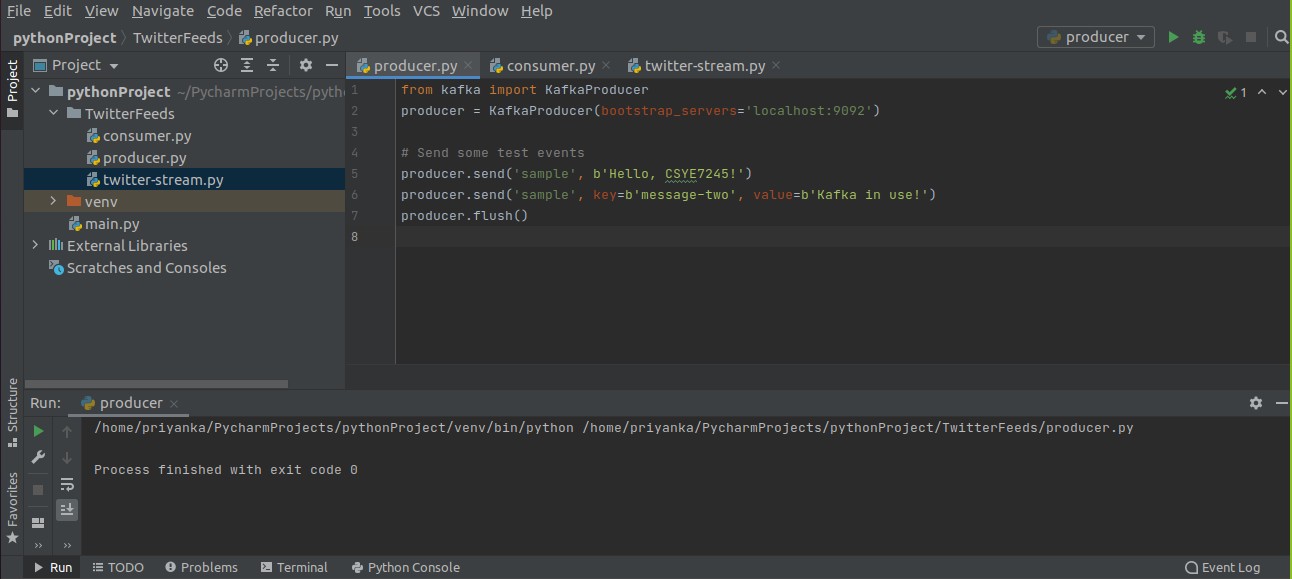


# Use Cases

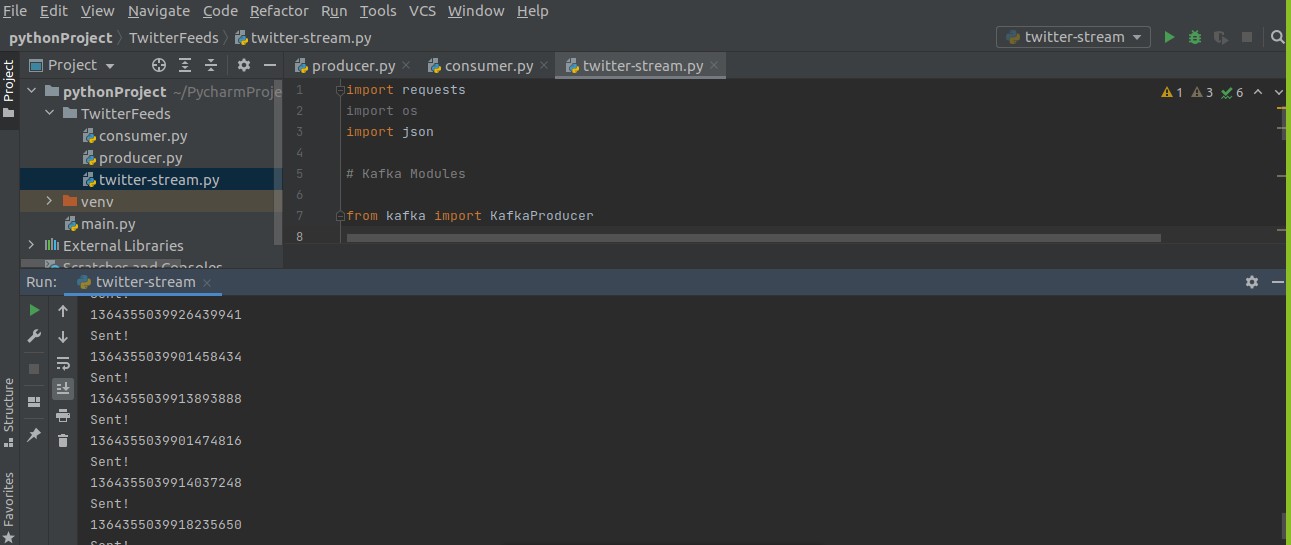
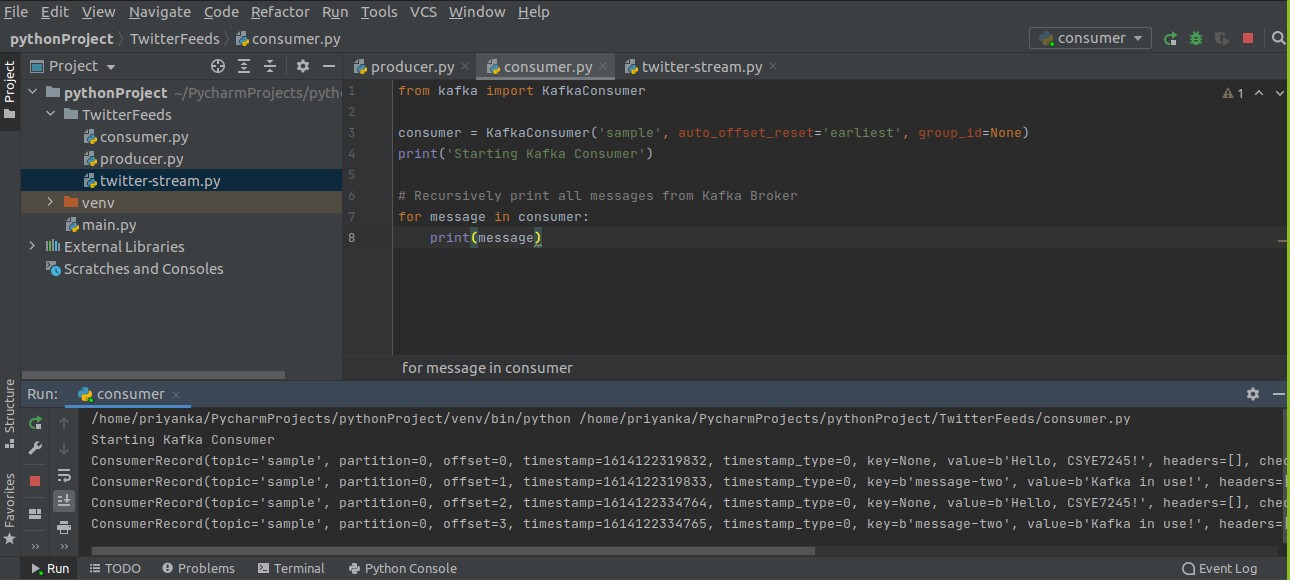
## Collecting real time sampled tweets from [Twitter](https://github.com/holladileep/CSYE7245-Spring2021-Labs/blob/main/kafka/twitter.com) and publishing them to our Kafka Broker

1. **producer.py**

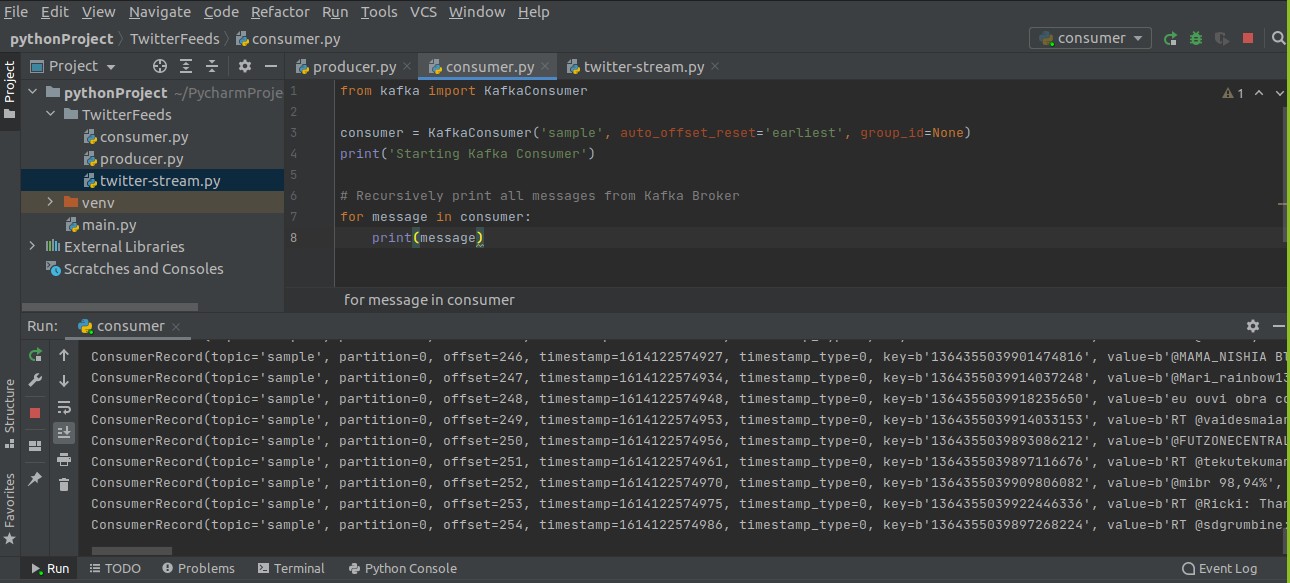
Running the script **producer.py** for generating events



## consumer.py

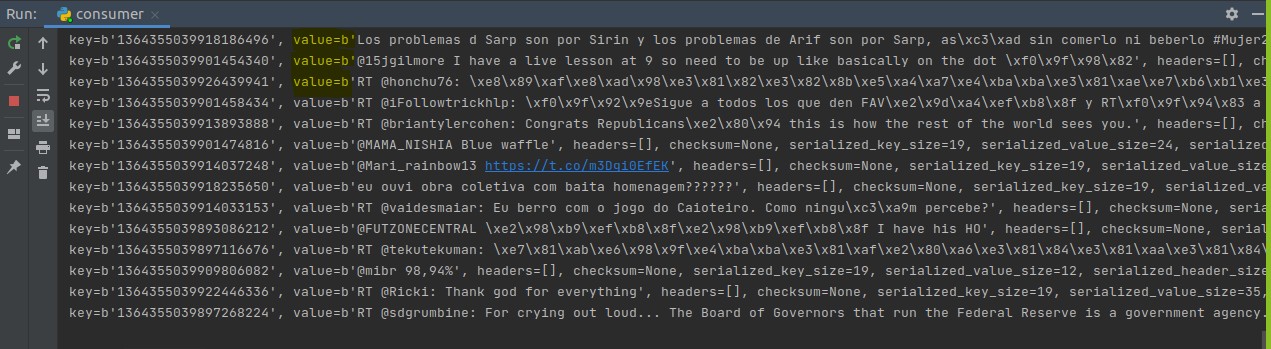
Running the script **consumer.py** to consume the events published by the producer.

## twitter-stream.py

Using the twitter-stream.py script to fetch tweets from Twitter's API in real-time.

Entering our bearer token in the twitter.py script under the **BEARER\_TOKEN** parameter. Tweets are published to the Kafka Broker.

On running **consumer.py** again, we can see all the published events that are collected by the consumer.



# Lessons learned

1. Learnt configuration of Oracle Virtual Box with Ubuntu operating system
2. Learnt the basic fundamentals of Apache Kafka
3. Implemented real-time data streaming using Twitter API in Apache Kafka

# References

<https://docs.cloudera.com/documentation/enterprise/6/6.1/PDF/cloudera-kafka.pdf>

[https://www.cloudkarafka.com/blog/2016-11-30-part1-kafka-for-beginners-what-is-apache-](https://www.cloudkarafka.com/blog/2016-11-30-part1-kafka-for-beginners-what-is-apache-kafka.html) [kafka.html](https://www.cloudkarafka.com/blog/2016-11-30-part1-kafka-for-beginners-what-is-apache-kafka.html)