BIG DATA ANALYTICS

LAB ASSIGNMENT 7

Mahesh Pachare

FINAL YEAR B.TECH
IT 191080054

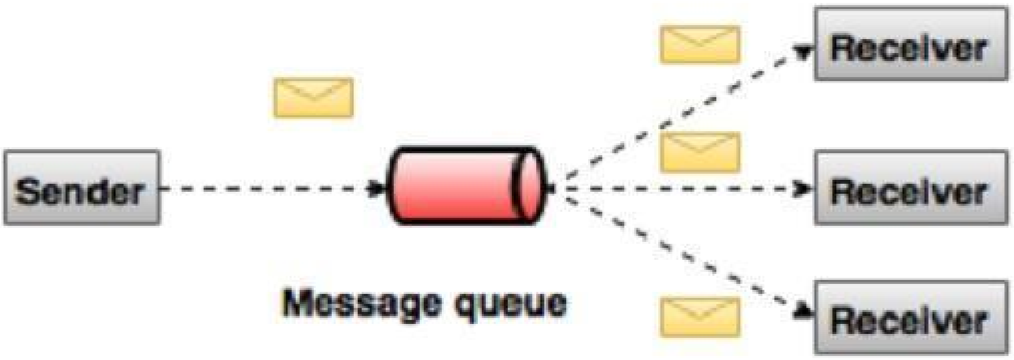
Aim: To Setup and install Apache Kafka and stream real time data from any social media website like Twitter, Facebook, instagram etc.

Theory:

Apache Kafka is an open-source distributed streaming system used for stream processing, real-time data pipelines, and data integration at scale. Originally created to handle real-time data feeds at LinkedIn in 2011, Kafka quickly evolved from messaging queue to a full-fledged event streaming platform capable of handling over 1 million messages per second, or trillions of messages per day.

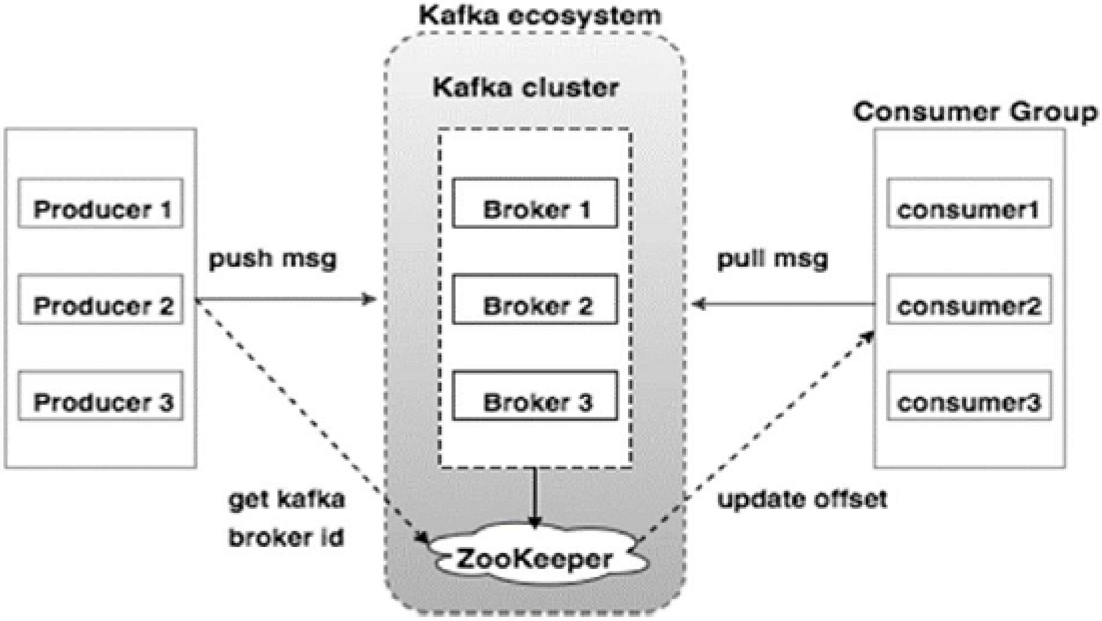
Publish-Subscribe Messaging System

In the publish-subscribe system, messages are persisted in a topic. Unlike point-to-point system, consumers can subscribe to one or more topic and consume all the messages in that topic. In the Publish-Subscribe system, message producers are called publishers and message consumers are called subscribers. A real-life example is Dish TV, which publishes different channels like sports, movies, music, etc., and anyone can subscribe to their own set of channels and get them whenever their subscribed channels are available.



Following are a few **benefits** of Kafka −

1. Reliability − Kafka is distributed, partitioned, replicated and fault tolerance.
2. Scalability − Kafka messaging system scales easily without down time..
3. Durability − Kafka uses "Distributed commit log" which means messages persists on disk as fast as possible, hence it is durable..
4. Performance − Kafka has high throughput for both publishing and subscribing messages. It maintains stable performance even though many TBs of messages are stored.



Commonly used to build real-time streaming data pipelines and real-time streaming applications, today, there are hundreds of Kafka use cases. Any company that relies on, or works with data can find numerous benefits.

1. **Data Pipelines**

   In the context of Apache Kafka, a streaming data pipeline means ingesting the data from sources into Kafka as it's created and then streaming that data from Kafka to one or more targets.

2. **Stream Processing**

   Stream processing includes operations like filters, joins, maps, aggregations, and other transformations which enterprises leverage to power many use-cases. Kafka Streams is a stream processing library built for Apache Kafka enabling enterprises to process data in real-time.Learn more

3. **Streaming Analytics**

   Kafka provides high throughput event delivery, and when combined with open-source technologies such as Druid can form a powerful Streaming Analytics Manager (SAM). Druid consumes streaming data from Kafka to enable analytical queries. Events are first loaded in Kafka, where they are buffered in Kafka brokers before they are consumed by Druid real-time workers.

4. **Streaming ETL**

Real-time ETL with Kafka combines different components and features such as Kafka Connect source and sink connectors to consume and produce data from/to any other database, application, or API, Single Message Transform (SMT) – an optional Kafka Connect feature, Kafka Streams for continuous data processing in real-time at scale.

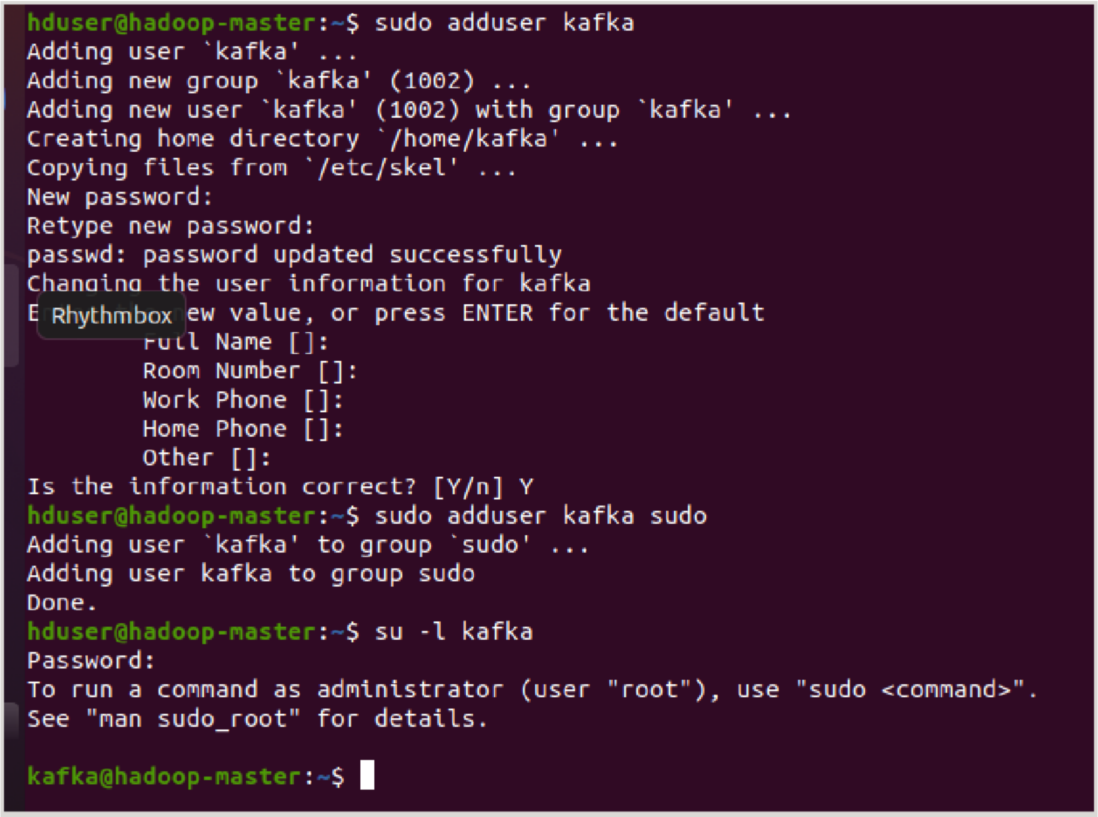
5. **Event-Driven Microservices**

Apache Kafka is the most popular tool for microservices because it solves many of the issues of microservices orchestration while enabling attributes that microservices aim to achieve, such as scalability, efficiency, and speed. It also facilitates inter-service communication while preserving ultra-low latency and fault tolerance.

Implementation:

A. Installing Apache Kafka:

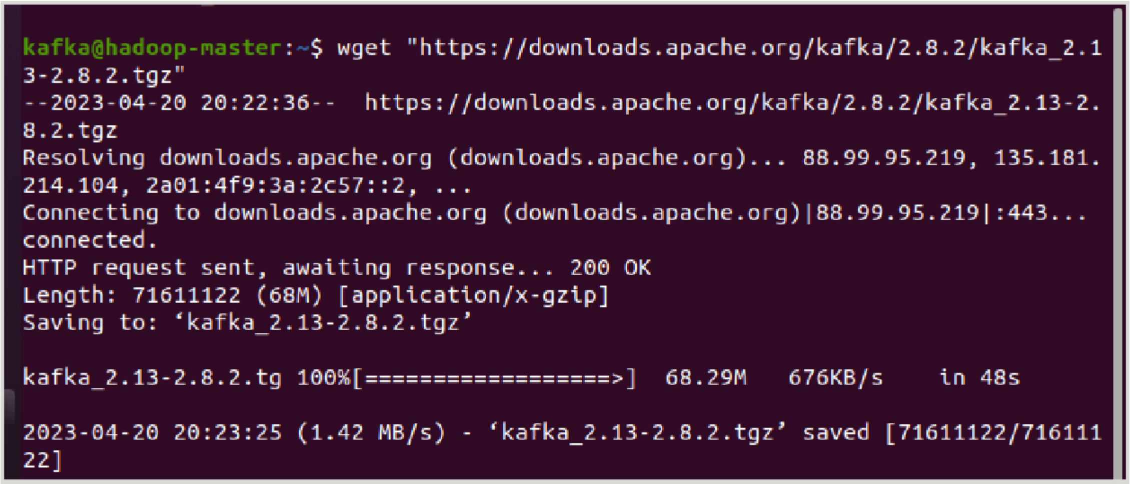
1. Because Kafka can handle requests over a network, your first step is to create a dedicated user for the service.
Log in to your server as your non-root sudo user, then create a user called kafka:



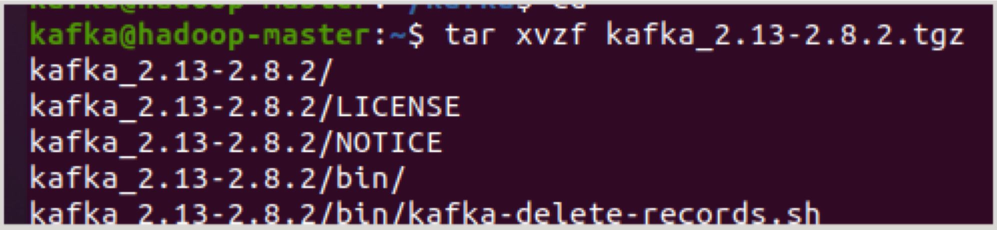
```
hduser@hadoop-master:~$ sudo adduser kafka
Adding user `kafka' ...
Adding new group `kafka' (1002) ...
Adding new user `kafka' (1002) with group `kafka' ...
Creating home directory `/home/kafka' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for kafka
E          ew value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
hduser@hadoop-master:~$ sudo adduser kafka sudo
Adding user `kafka' to group `sudo' ...
Adding user kafka to group sudo
Done.
hduser@hadoop-master:~$ su -l kafka
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

kafka@hadoop-master:~$
```

2. Downloading and Extracting the Kafka Binaries

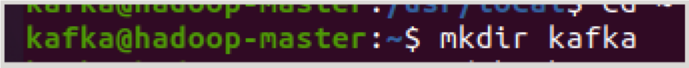
In this step, you'll download and extract the Kafka binaries into dedicated folders in your kafka user's home directory.



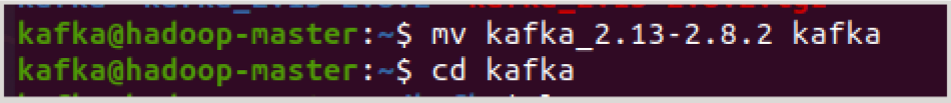
```
kafka@hadoop-master:~$ wget "https://downloads.apache.org/kafka/2.8.2/kafka_2.1
3-2.8.2.tgz"
--2023-04-20 20:22:36--  https://downloads.apache.org/kafka/2.8.2/kafka_2.13-2.
8.2.tgz
Resolving downloads.apache.org (downloads.apache.org)... 88.99.95.219, 135.181.
214.104, 2a01:4f9:3a:2c57::2, ...
Connecting to downloads.apache.org (downloads.apache.org)|88.99.95.219|:443...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 71611122 (68M) [application/x-gzip]
Saving to: ‘kafka_2.13-2.8.2.tgz’

kafka_2.13-2.8.2.tg 100%[===================>]  68.29M   676KB/s    in 48s

2023-04-20 20:23:25 (1.42 MB/s) - ‘kafka_2.13-2.8.2.tgz’ saved [71611122/716111
22]
```



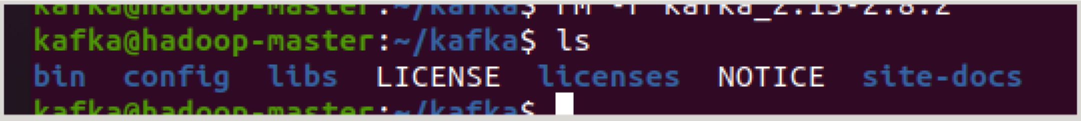
```
kafka@hadoop-master:~$ tar xvzf kafka_2.13-2.8.2.tgz
kafka_2.13-2.8.2/
kafka_2.13-2.8.2/LICENSE
kafka_2.13-2.8.2/NOTICE
kafka_2.13-2.8.2/bin/
kafka_2.13-2.8.2/bin/kafka-delete-records.sh
```



```
kafka@hadoop-master:~$ mkdir kafka
```



```
kafka@hadoop-master:~$ mv kafka_2.13-2.8.2 kafka
kafka@hadoop-master:~$ cd kafka
```



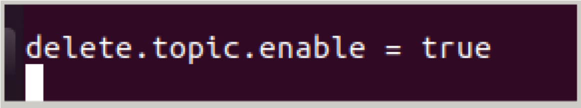
```
kafka@hadoop-master:~/kafka$ rm -r kafka_2.13-2.8.2
kafka@hadoop-master:~/kafka$ ls
bin  config  libs  LICENSE  licenses  NOTICE  site-docs
kafka@hadoop-master:~/kafka$
```

3. Configuring the Kafka Server

Kafka’s default behavior will not allow you to delete a topic. To modify this, you must edit the configuration file, which you will do in this step.

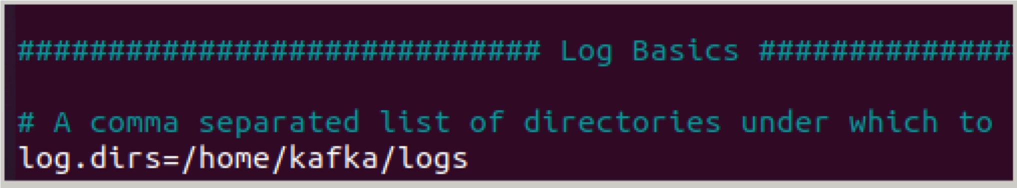
Kafka’s configuration options are specified in server.properties. Open this file with nano

First, add a setting that will allow you to delete Kafka topics. Add the following line to the bottom of the file:



```
delete.topic.enable = true
```

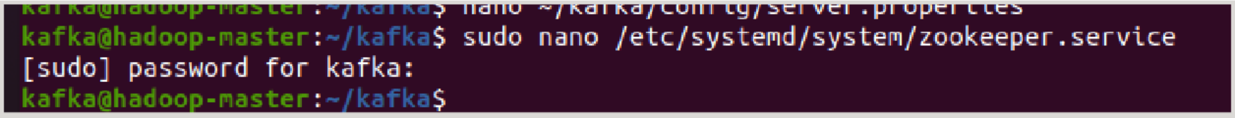
Second, you’ll change the directory where the Kafka logs are stored by modifying the log.dirs property. Find the log.dirs property and replace the existing route with the follwoing route:



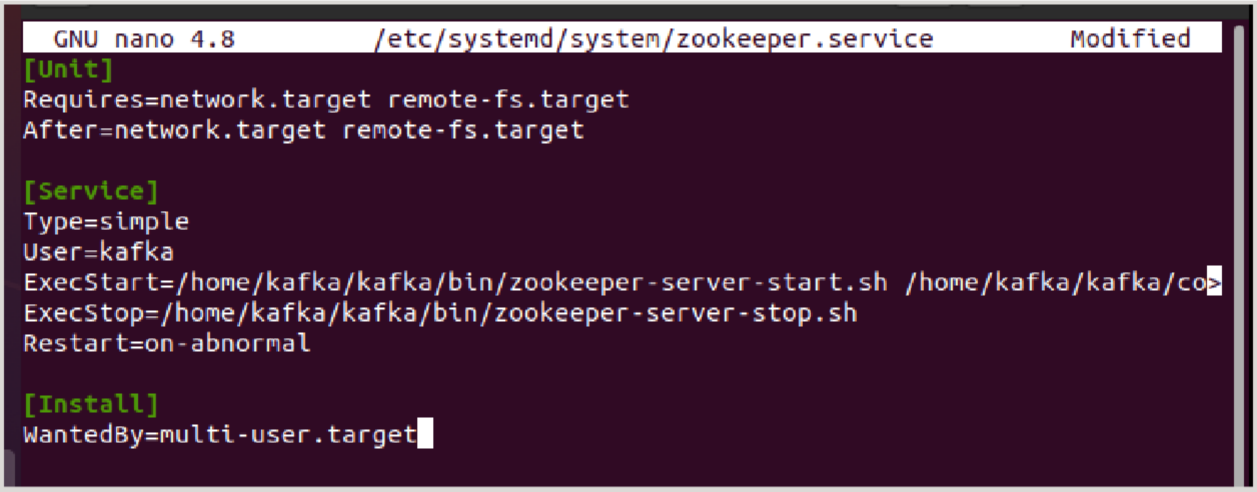
```
############################# Log Basics ##############

# A comma separated list of directories under which to
log.dirs=/home/kafka/logs
```

4.Creating systemd Unit Files and Starting the Kafka Server

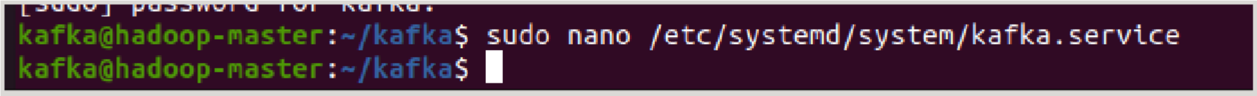
Kafka uses Zookeeper to manage its cluster state and configurations. It is used in many distributed systems



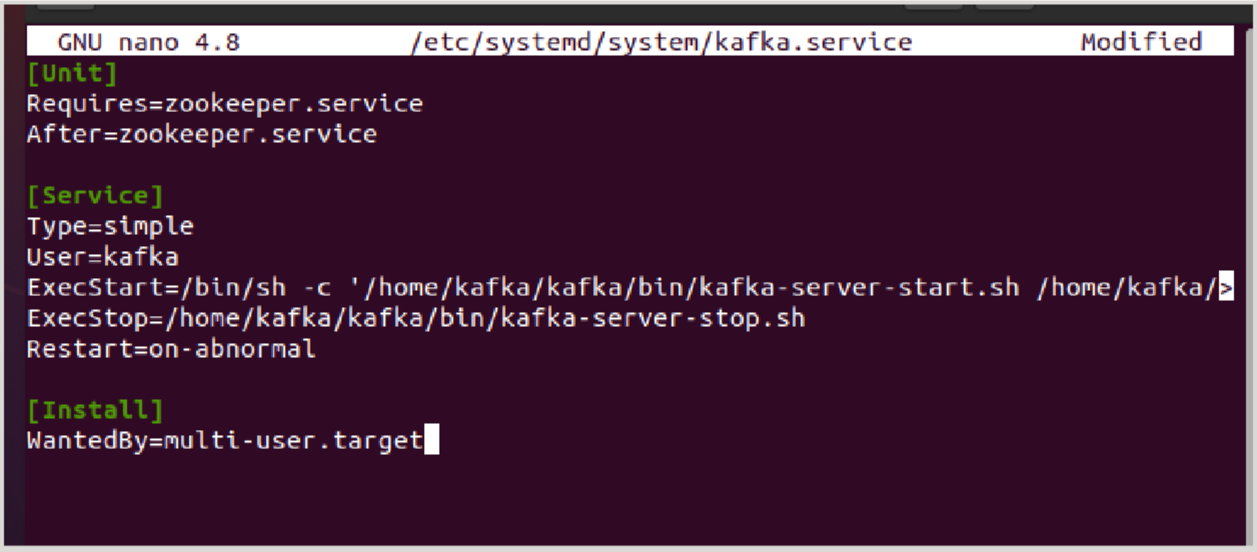
```
kafka@hadoop-master:~/kafka$ nano ~/kafka/config/server.properties
kafka@hadoop-master:~/kafka$ sudo nano /etc/systemd/system/zookeeper.service
[sudo] password for kafka:
kafka@hadoop-master:~/kafka$
```



```
  GNU nano 4.8          /etc/systemd/system/zookeeper.service          Modified
[Unit]
Requires=network.target remote-fs.target
After=network.target remote-fs.target

[Service]
Type=simple
User=kafka
ExecStart=/home/kafka/kafka/bin/zookeeper-server-start.sh /home/kafka/kafka/co>
ExecStop=/home/kafka/kafka/bin/zookeeper-server-stop.sh
Restart=on-abnormal

[Install]
WantedBy=multi-user.target
```

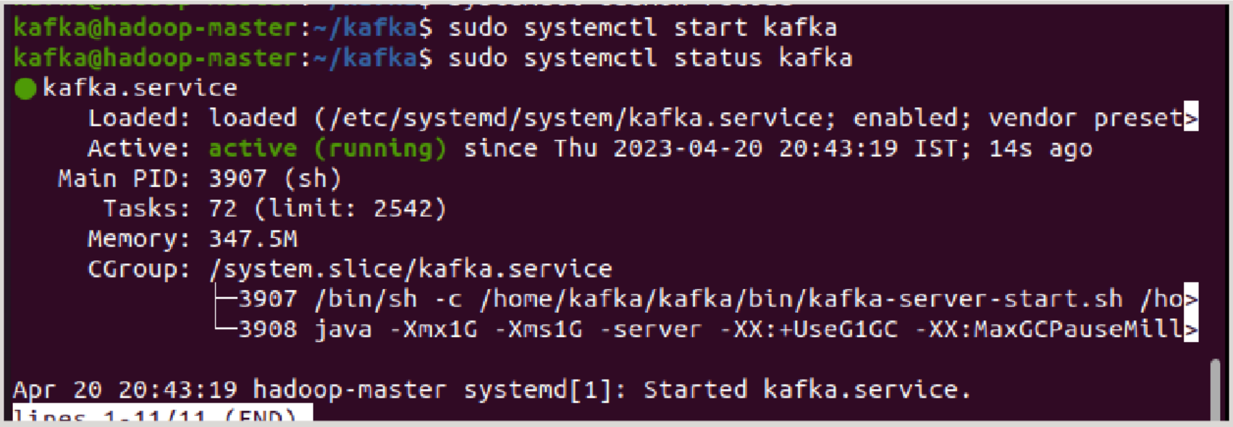


```
[sudo] password for kafka:
kafka@hadoop-master:~/kafka$ sudo nano /etc/systemd/system/kafka.service
kafka@hadoop-master:~/kafka$
```



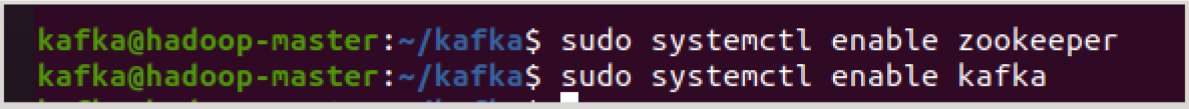
```
  GNU nano 4.8              /etc/systemd/system/kafka.service              Modified
[Unit]
Requires=zookeeper.service
After=zookeeper.service

[Service]
Type=simple
User=kafka
ExecStart=/bin/sh -c '/home/kafka/kafka/bin/kafka-server-start.sh /home/kafka/>
ExecStop=/home/kafka/kafka/bin/kafka-server-stop.sh
Restart=on-abnormal

[Install]
WantedBy=multi-user.target
```

Now that you have defined the units, start Kafka with the following command:



```
kafka@hadoop-master:~/kafka$ sudo systemctl start kafka
kafka@hadoop-master:~/kafka$ sudo systemctl status kafka
● kafka.service
     Loaded: loaded (/etc/systemd/system/kafka.service; enabled; vendor preset>
     Active: active (running) since Thu 2023-04-20 20:43:19 IST; 14s ago
   Main PID: 3907 (sh)
      Tasks: 72 (limit: 2542)
     Memory: 347.5M
     CGroup: /system.slice/kafka.service
             ├─3907 /bin/sh -c /home/kafka/kafka/bin/kafka-server-start.sh /ho>
             └─3908 java -Xmx1G -Xms1G -server -XX:+UseG1GC -XX:MaxGCPauseMill>

Apr 20 20:43:19 hadoop-master systemd[1]: Started kafka.service.
lines 1-11/11 (END)
```

To enable the kafka service on server boot, run the following command:



```
kafka@hadoop-master:~/kafka$ sudo systemctl enable zookeeper
kafka@hadoop-master:~/kafka$ sudo systemctl enable kafka
```

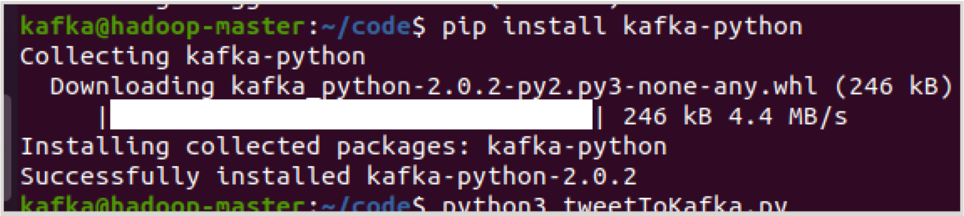
B. Sending real-time data from Twitter to Kafka

1. **Create a Python script:**

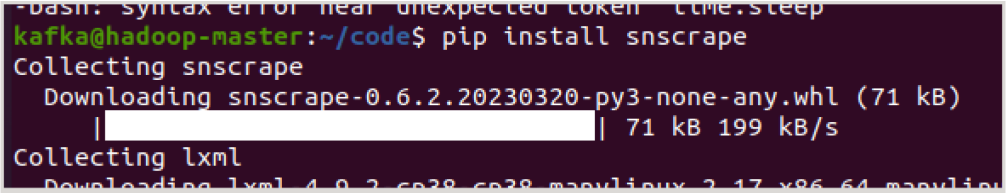
```
import time
from kafka import KafkaProducer
import snscrape.modules.twitter as sntwitter
from datetime import datetime
import json
# Set up Kafka producer
producer = KafkaProducer(bootstrap_servers=['localhost:9092'],
value_serializer=lambda x: json.dumps(x).encode('utf-8'))
# Set up the Twitter scraper
search_terms = ['covid', 'vaccine', 'lockdown'] # example search
terms
since_date = datetime.now().strftime('%Y-%m-%d') # start with
current date
while True:
    for term in search_terms:
        # Scrape tweets using snscrape
        for i, tweet in
enumerate(sntwitter.TwitterSearchScraper(f'{term}
since:{since_date}').get_items()):
            if i >= 10: # only scrape 10 tweets per search term
                break
            # Send tweet to Kafka
            producer.send('twitter', value={'term': term,
'tweet':tweet.content})
            producer.flush()
    # Wait for some time before scraping again
    time.sleep(60) # scrape every minute
```

2. **Running Python script to scrape data from Twitter and send it to the topic “Twitter” in Kafka**

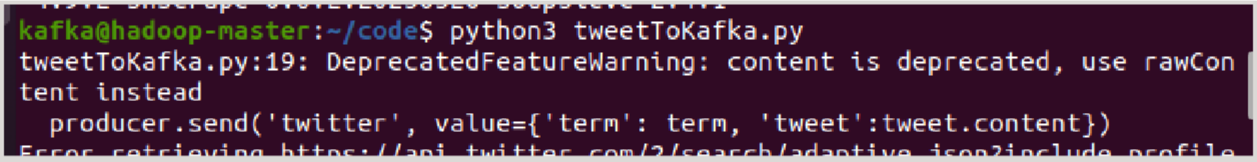
   Install the necessary modules using pip and execute the python script



```
kafka@hadoop-master:~/code$ pip install kafka-python
Collecting kafka-python
  Downloading kafka_python-2.0.2-py2.py3-none-any.whl (246 kB)
     |████████████████████████████████| 246 kB 4.4 MB/s
Installing collected packages: kafka-python
Successfully installed kafka-python-2.0.2
kafka@hadoop-master:~/code$ python3 tweetToKafka.py
```

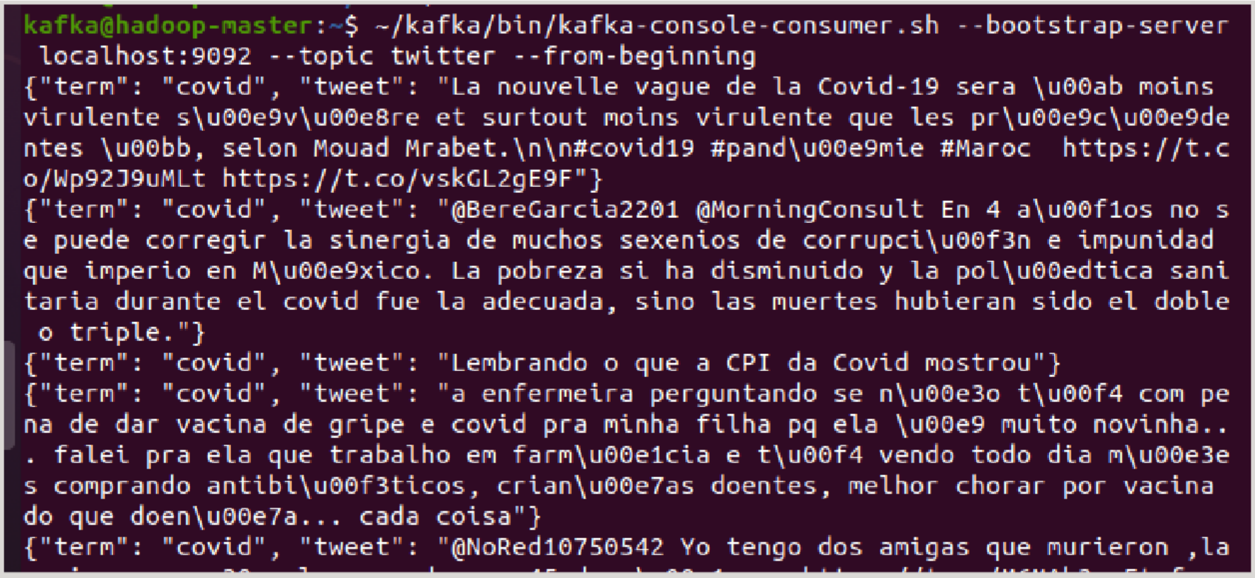


```
-bash: syntax error near unexpected token `time.sleep'
kafka@hadoop-master:~/code$ pip install snscrape
Collecting snscrape
  Downloading snscrape-0.6.2.20230320-py3-none-any.whl (71 kB)
     |████████████████████████████████| 71 kB 199 kB/s
Collecting lxml
  Downloading lxml-4.9.2-cp38-cp38-manylinux_2_17_x86_64.manylinu
```



```
4.9.2 snscrape-0.6.2.20230320 soupsieve-2.4.1
kafka@hadoop-master:~/code$ python3 tweetToKafka.py
tweetToKafka.py:19: DeprecatedFeatureWarning: content is deprecated, use rawCon
tent instead
  producer.send('twitter', value={'term': term, 'tweet':tweet.content})
Error retrieving https://api.twitter.com/2/search/adaptive.json?include_profile
```

3. Receiving data in real-time using Kafka consumer on the topic “Twitter”



```
kafka@hadoop-master:~$ ~/kafka/bin/kafka-console-consumer.sh --bootstrap-server
 localhost:9092 --topic twitter --from-beginning
{"term": "covid", "tweet": "La nouvelle vague de la Covid-19 sera \u00ab moins
virulente s\u00e9v\u00e8re et surtout moins virulente que les pr\u00e9c\u00e9de
ntes \u00bb, selon Mouad Mrabet.\n\n#covid19 #pand\u00e9mie #Maroc  https://t.c
o/Wp92J9uMLt https://t.co/vskGL2gE9F"}
{"term": "covid", "tweet": "@BereGarcia2201 @MorningConsult En 4 a\u00f1os no s
e puede corregir la sinergia de muchos sexenios de corrupci\u00f3n e impunidad
que imperio en M\u00e9xico. La pobreza si ha disminuido y la pol\u00edtica sani
taria durante el covid fue la adecuada, sino las muertes hubieran sido el doble
 o triple."}
{"term": "covid", "tweet": "Lembrando o que a CPI da Covid mostrou"}
{"term": "covid", "tweet": "a enfermeira perguntando se n\u00e3o t\u00f4 com pe
na de dar vacina de gripe e covid pra minha filha pq ela \u00e9 muito novinha..
. falei pra ela que trabalho em farm\u00e1cia e t\u00f4 vendo todo dia m\u00e3e
s comprando antibi\u00f3ticos, crian\u00e7as doentes, melhor chorar por vacina
do que doen\u00e7a... cada coisa"}
{"term": "covid", "tweet": "@NoRed10750542 Yo tengo dos amigas que murieron ,la
```

Conclusion:

Thus successfully installed apache kafka and streamed real-time data of social media website Twitter by scraping data using “Snscrape”.