## Kafka Commands

**Start services required for kafka**

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

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

**Create Topic**

bin/kafka-topics.sh --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1 --topic testTopic

**List existing Topics**

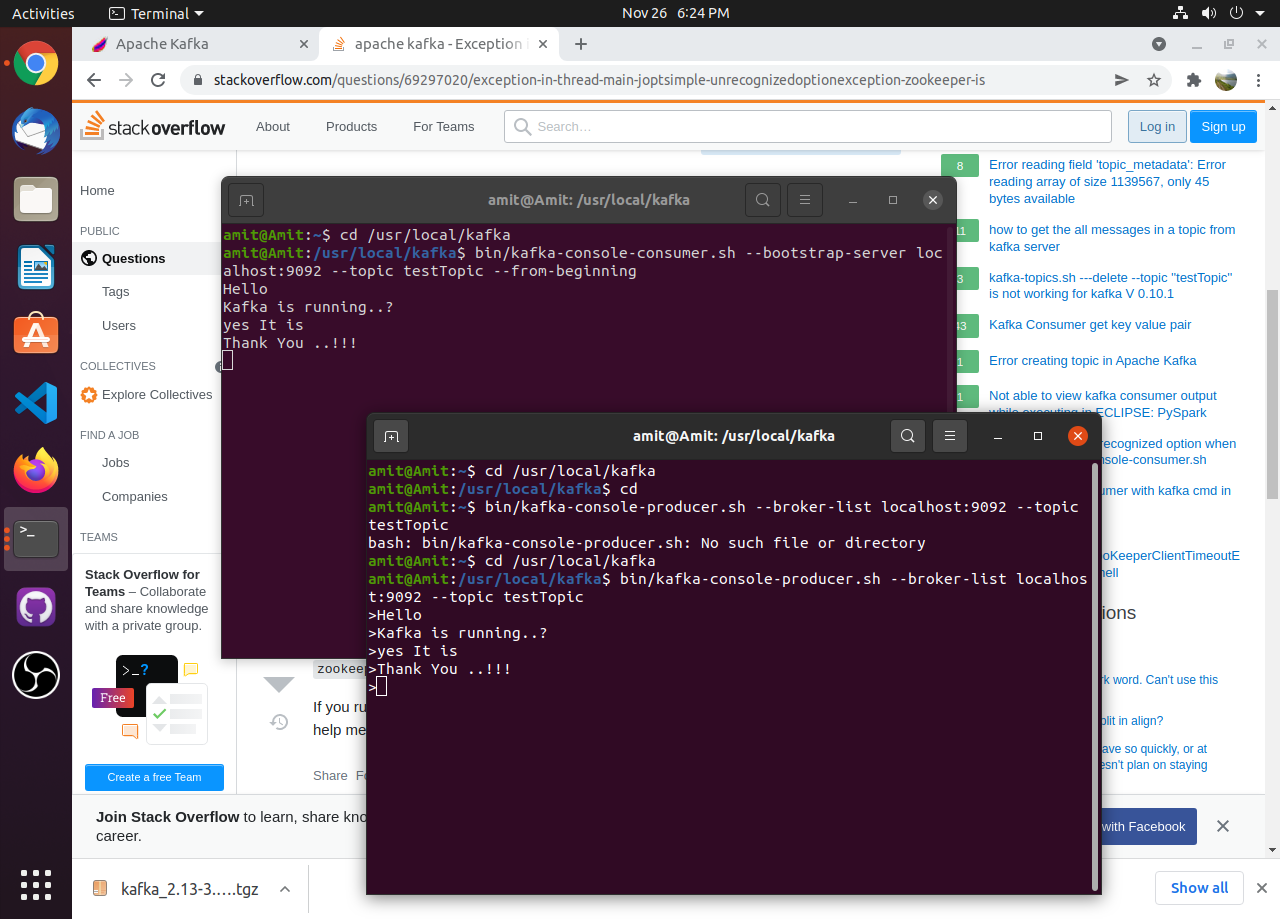
bin/kafka-topics.sh --list --bootstrap-server localhost:9092

**Start Producer**

bin/kafka-console-producer.sh --broker-list localhost:9092 --topic testTopic

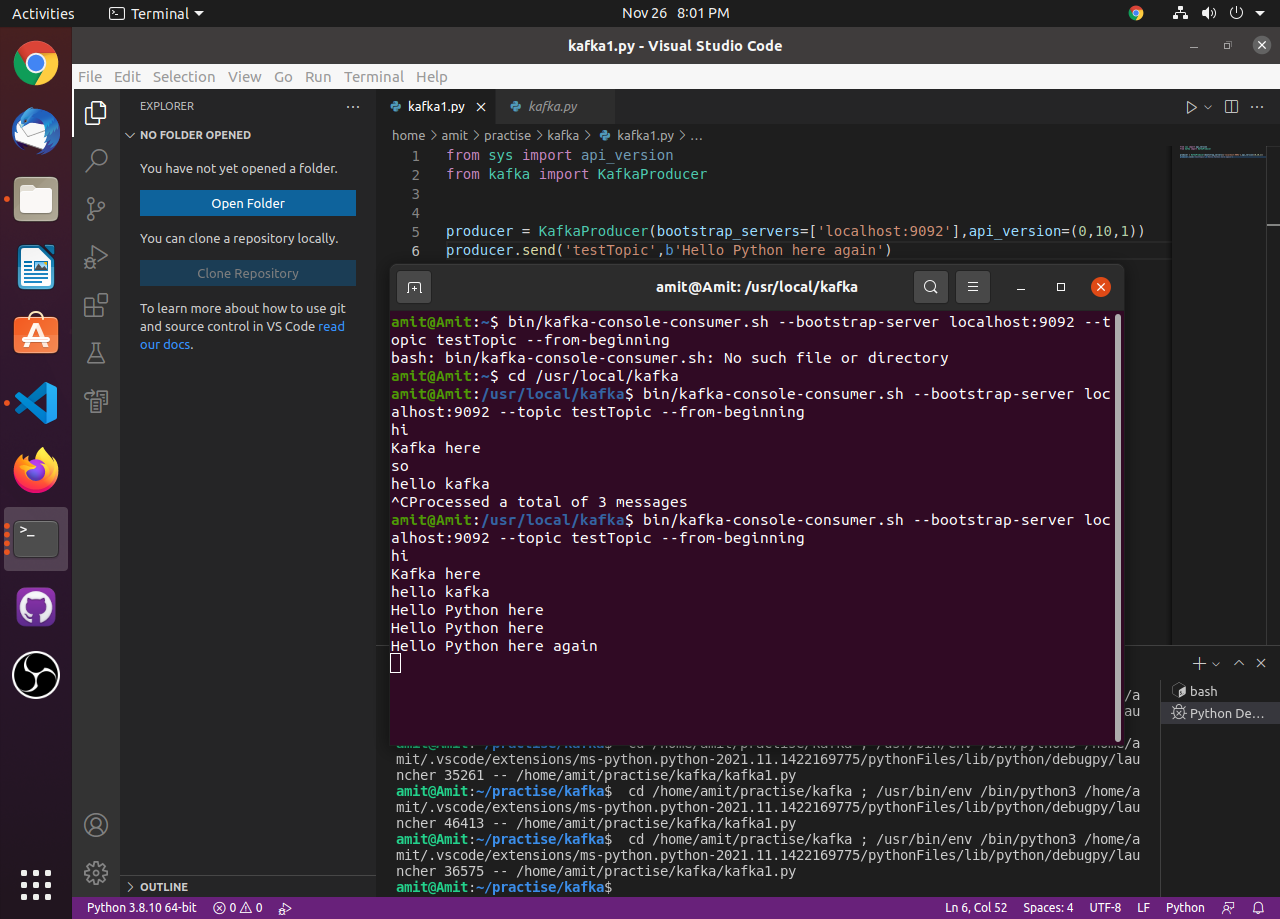
**Start Receiver**

bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic testTopic --from-beginning

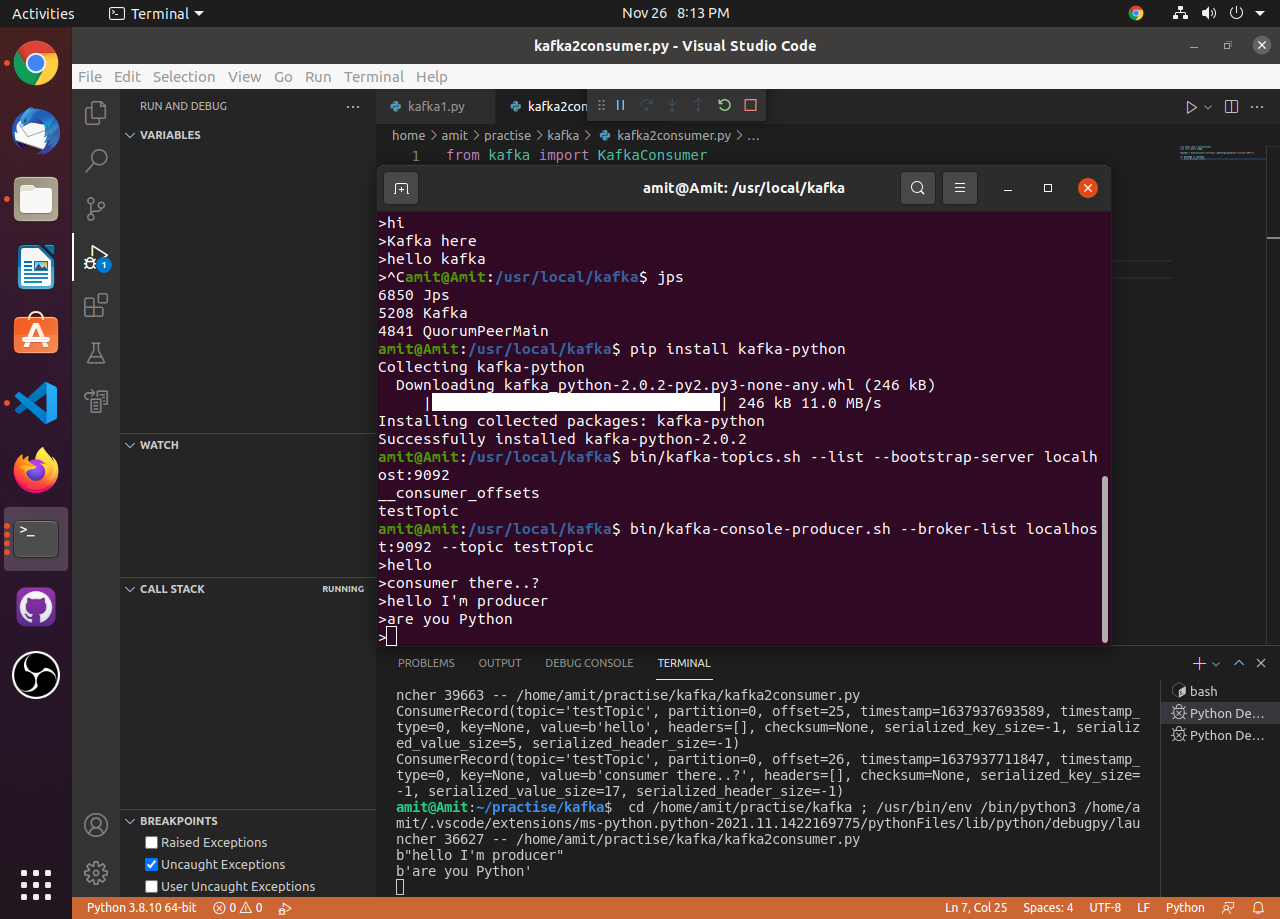
****

**Using Kafka with Python (using kafka library)**

Sending message as a producer using python



Receiving message as a consumer using python



**Multi-Broker Setup in Kafka**

Update the server.properties file

Make two copies of server.properties file and rename them to server1.properties and server2.properties

**Changes in server1.properties**

broker.id=1

listeners=PLAINTEXT://:9093

log.dirs=/tmp/kafka-logs1

**Changes in server2.properties**

broker.id=2

listeners=PLAINTEXT://:9094

log.dirs=/tmp/kafka-logs2

**Start zookeeper and kafka servers**

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

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

sh bin/kafka-server-start.sh config/server1.properties

sh bin/kafka-server-start.sh config/server2.properties

**Create Topic (use anyone port number)**

bin/kafka-topics.sh --create --bootstrap-server localhost:9093 --replication-factor 1 --partitions 1 --topic testTopic

bin/kafka-topics.sh --create --bootstrap-server localhost:9094 --replication-factor 1 --partitions 1 --topic testTopic2

**Alter Topic (use anyone port number)**

bin/kafka-topics.sh --bootstrap-server localhost:9093 --alter --topic testTopic --partitions 3

**List existing Topics (use anyone port number)**

bin/kafka-topics.sh --list --bootstrap-server localhost:9093

**Start Producer**

bin/kafka-console-producer.sh --broker-list localhost:9092,localhost:9093,localhost:9094 --topic testTopic

**Start Receiver**

bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic testTopic --from-beginning

**Kafka output to HDFS**

Producer

import csv

import requests

from kafka import KafkaProducer

from json import dumps

# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key

CSV\_URL = 'https://www.alphavantage.co/query?function=TIME\_SERIES\_INTRADAY\_EXTENDED&symbol=IBM&interval=15min&slice=year1month1&apikey=demo'

with requests.Session() as s:

download = s.get(CSV\_URL)

decoded\_content = download.content.decode('utf-8')

cr = csv.reader(decoded\_content.splitlines(), delimiter=',')

my\_list = list(cr)

producer = KafkaProducer(bootstrap\_servers=['localhost:9092'],value\_serializer=lambda K:dumps(K).encode('utf-8'))

for row in my\_list:

producer.send('testTopic',row)

Consumer

from kafka import KafkaConsumer

import pydoop.hdfs as hdfs

consumer = KafkaConsumer('testTopic',bootstrap\_servers=['localhost:9092'])

hdfs\_path = 'hdfs://localhost:9000/StockDatapydoop/stock\_file.txt'

for message in consumer:

values = message.value.decode('utf-8')

with hdfs.open(hdfs\_path, 'at') as f:

print(message.value)

f.write(f"{values}\n")

**Kafka TWITTER tweets to HDFS**

Producer

import tweepy

from textblob import TextBlob

from kafka import KafkaProducer

from json import dumps

with open('config.json') as json\_file:

data = json.load(json\_file)

CONSUMER\_KEY = data['CONSUMER\_KEY']

CONSUMER\_SECRET = data['CONSUMER\_SECRET']

ACCESS\_TOKEN = data['ACCESS\_TOKEN']

ACCESS\_TOKEN\_SECRET = data['ACCESS\_TOKEN\_SECRET']

auth = tweepy.OAuthHandler(CONSUMER\_KEY, CONSUMER\_SECRET)

auth.set\_access\_token(ACCESS\_TOKEN, ACCESS\_TOKEN\_SECRET)

producer = KafkaProducer(bootstrap\_servers=['localhost:9092'],value\_serializer=lambda K:dumps(K).encode('utf-8'))

api = tweepy.API(auth)

cursor = tweepy.Cursor(api.search\_tweets,q="crypto",tweet\_mode='extended').items(100)

for tweet in cursor:

producer.send('testTopic',tweet.full\_text)

Consumer

from kafka import KafkaConsumer

import pydoop.hdfs as hdfs

consumer = KafkaConsumer('testTopic',bootstrap\_servers=['localhost:9092'])

hdfs\_path = 'hdfs://localhost:9000/StockDatapydoop/stock\_file.txt'

for message in consumer:

values = message.value.decode('utf-8')

with hdfs.open(hdfs\_path, 'at') as f:

print(message.value)

f.write(f"{values}\n")