Kafka + Spark Streaming + PySpark on GCP Ubuntu

Step 1: Installing Spark

**Method 1: Using Instance VM

1) Download Spark Package and unpack it

```
ycao@cs570vmserver:~$ wget https://dlcdn.apache.org/spark/spark-3.3.1/spark-3.3.1-bin-hadoop3.tgz
--2022-12-06 06:10:07--  https://dlcdn.apache.org/spark/spark-3.3.1/spark-3.3.1-bin-hadoop3.tgz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 299350810 (285M) [application/x-gzip]
Saving to: 'spark-3.3.1-bin-hadoop3.tgz'

spark-3.3.1-bin-hadoop3.tgz 100%[=========================================>] 285.48M   137MB/s    in 2.1s

2022-12-06 06:10:09 (137 MB/s) - 'spark-3.3.1-bin-hadoop3.tgz' saved [299350810/299350810]
```

```
ycao@cs570vmserver:~$ tar -xvf spark-3.3.1-bin-hadoop3.tgz
spark-3.3.1-bin-hadoop3/
spark-3.3.1-bin-hadoop3/LICENSE
spark-3.3.1-bin-hadoop3/NOTICE
spark-3.3.1-bin-hadoop3/R/
spark-3.3.1-bin-hadoop3/R/lib/
spark-3.3.1-bin-hadoop3/R/lib/SparkR/
spark-3.3.1-bin-hadoop3/R/lib/SparkR/DESCRIPTION
spark-3.3.1-bin-hadoop3/R/lib/SparkR/INDEX
spark-3.3.1-bin-hadoop3/R/lib/SparkR/Meta/
spark-3.3.1-bin-hadoop3/R/lib/SparkR/Meta/Rd.rds
spark-3.3.1-bin-hadoop3/R/lib/SparkR/Meta/features.rds
spark-3.3.1-bin-hadoop3/R/lib/SparkR/Meta/hsearch.rds
spark-3.3.1-bin-hadoop3/R/lib/SparkR/Meta/links.rds
spark-3.3.1-bin-hadoop3/R/lib/SparkR/Meta/nsInfo.rds
spark-3.3.1-bin-hadoop3/R/lib/SparkR/Meta/package.rds
spark-3.3.1-bin-hadoop3/R/lib/SparkR/Meta/vignette.rds
spark-3.3.1-bin-hadoop3/R/lib/SparkR/NAMESPACE
```

2) Add environment variable and path to ~/.bashrc

```
ycao@cs570vmserver:~$ vi .bashrc
ycao@cs570vmserver:~$ source ~/.bashrc
```

```
#set spark related environment varibales
export SPARK_HOME=/home/ycao/spark-3.3.1-bin-hadoop3
export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin
```

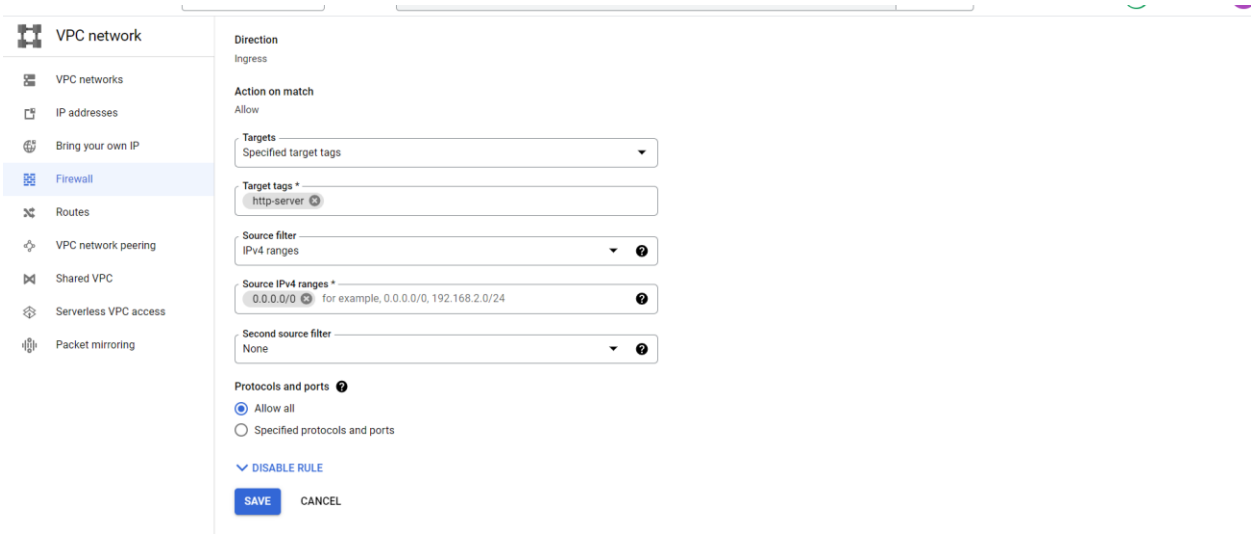
Note: Execute the file after changing the file (**$ source ~/.bashrc**)

3) Test installation of pyspark

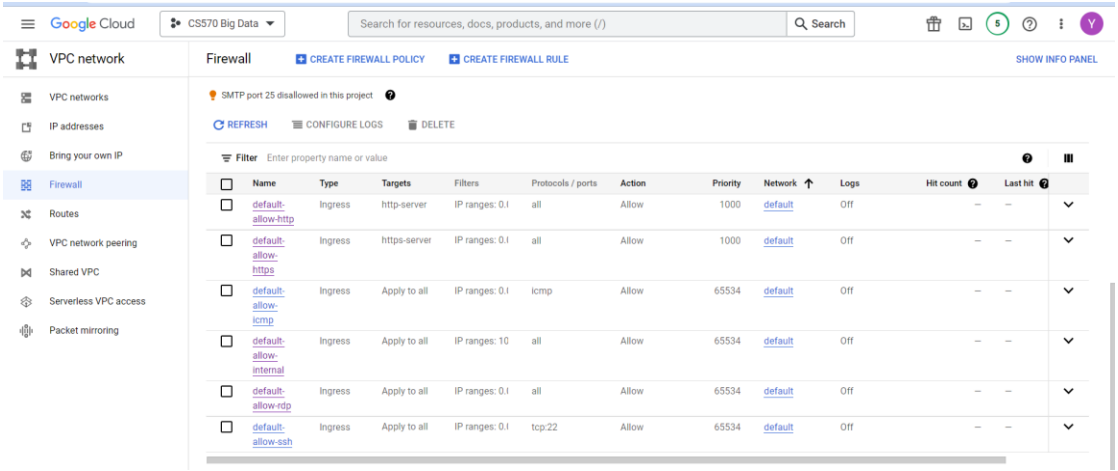
```
ycao@cs570vmserver:~$ pyspark
Python 3.8.10 (default, Jun 22 2022, 20:18:18)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/12/06 06:13:12 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builti
n-java classes where applicable
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.3.1
      /_/

Using Python version 3.8.10 (default, Jun 22 2022 20:18:18)
Spark context Web UI available at http://cs570vmserver.us-west2-a.c.cs570-big-data-363104.internal:4040
Spark context available as 'sc' (master = local[*], app id = local-1670307194502).
SparkSession available as 'spark'.
>>>
```

4) Change VM config on GCP

➔ VPC network



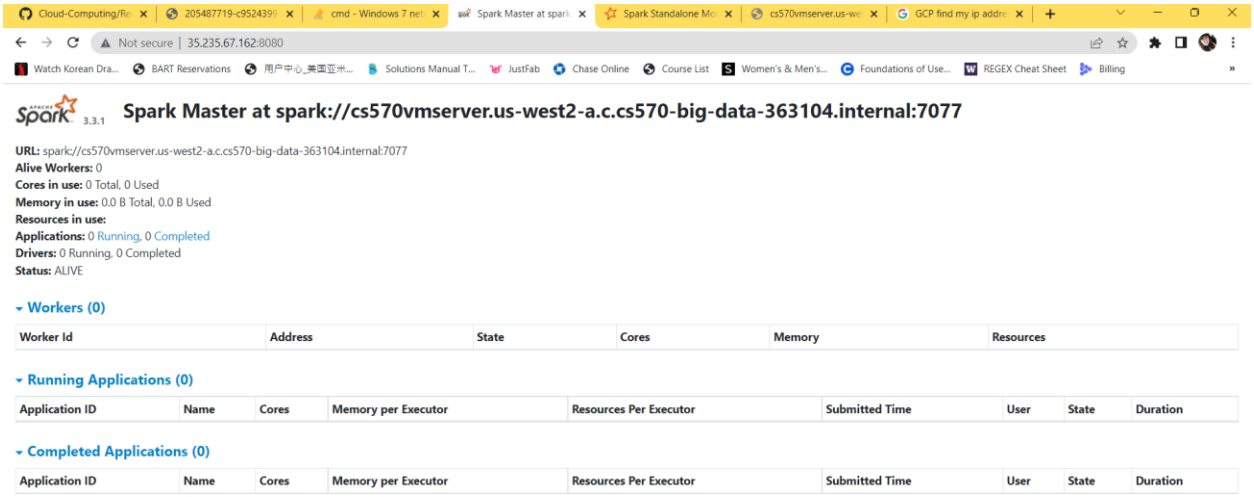
➔ Firewall



5) Start master

```
ycao@cs570vmserver:~$ start-master.sh
starting org.apache.spark.deploy.master.Master, logging to /home/ycao/spark-3.3.1-bin-hadoop3/logs/spark-ycao-o
rg.apache.spark.deploy.master.Master-1-cs570vmserver.out
ycao@cs570vmserver:~$
```

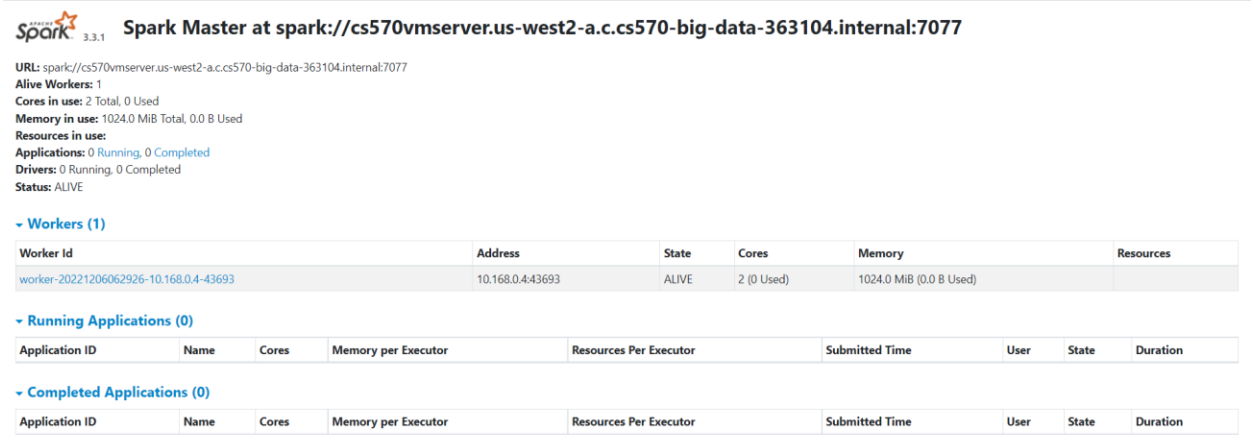
➔ Spark UI



6) Start slave

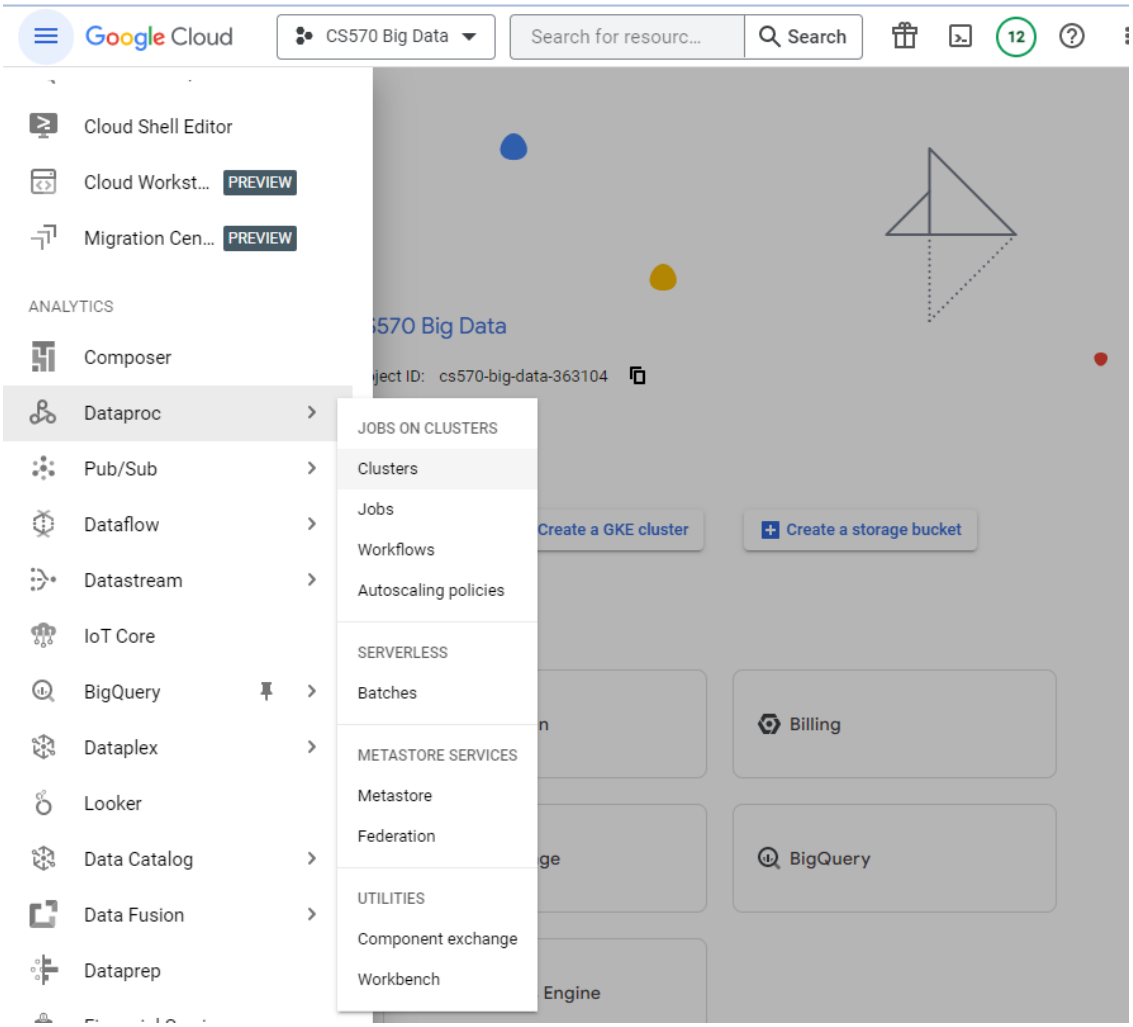
```
ycao@cs570vmserver:~$ start-slave.sh spark://35.235.67.162:7077
This script is deprecated, use start-worker.sh
starting org.apache.spark.deploy.worker.Worker, logging to /home/ycao/spark-3.3.1-bin-hadoop3/logs/spark-ycao-o
rg.apache.spark.deploy.worker.Worker-1-cs570vmserver.out
ycao@cs570vmserver:~$
```

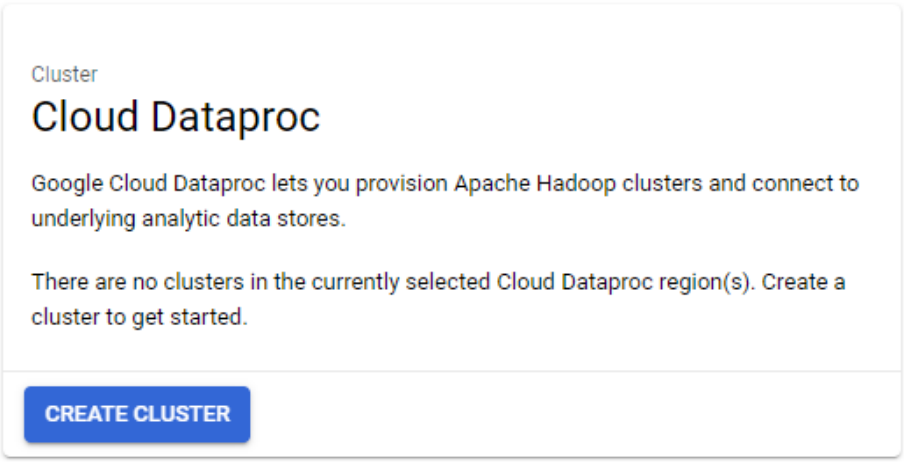
➔ Spark UI

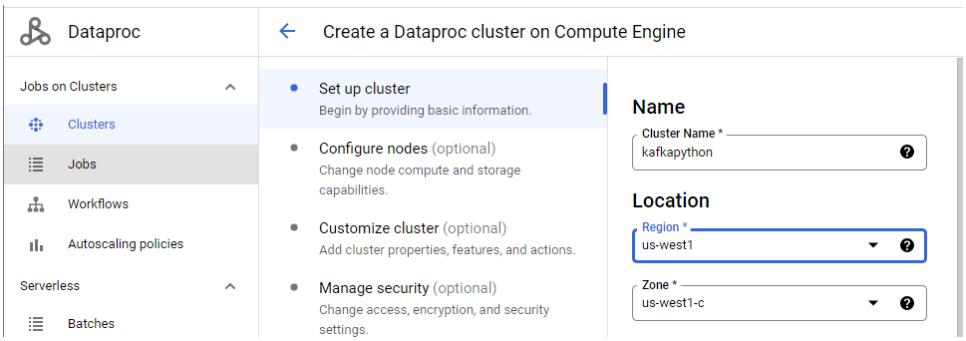


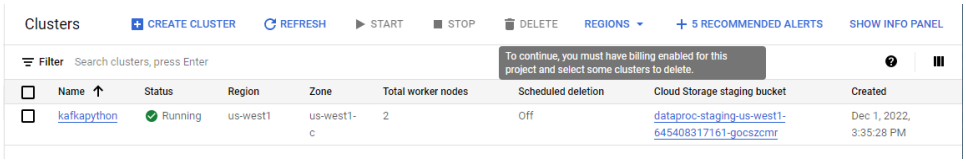
**Method 2: Using Cluster

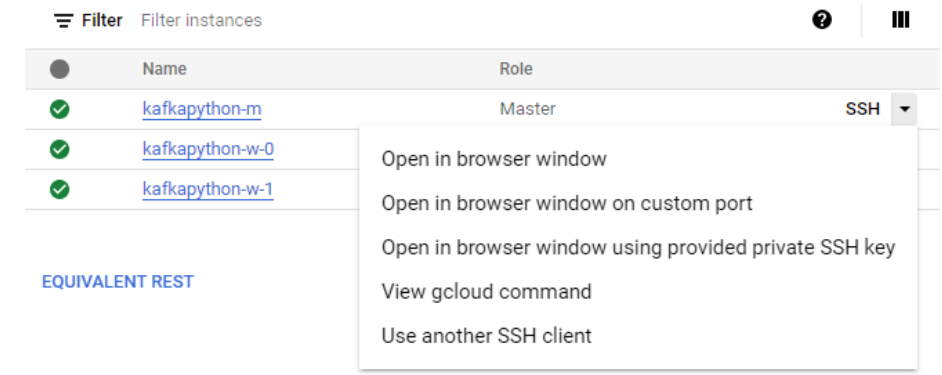
1) Create cluster











➔ Test pyspark

```
ycao@kafkapython-m:~$ pyspark
Python 3.8.13 | packaged by conda-forge | (default, Mar 25 2022, 06:04:10)
[GCC 10.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/12/02 00:14:46 INFO org.apache.spark.SparkEnv: Registering MapOutputTracker
22/12/02 00:14:46 INFO org.apache.spark.SparkEnv: Registering BlockManagerMaster
22/12/02 00:14:47 INFO org.apache.spark.SparkEnv: Registering BlockManagerMasterHeartbeat
22/12/02 00:14:47 INFO org.apache.spark.SparkEnv: Registering OutputCommitCoordinator
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.1.3
      /_/

Using Python version 3.8.13 (default, Mar 25 2022 06:04:10)
Spark context Web UI available at http://kafkapython-m.us-west1-c.c.cs570-big-data-363104.internal:44009
Spark context available as 'sc' (master = yarn, app id = application_1669937816581_0002).
SparkSession available as 'spark'.
>>>
```

Done!

Step 2: Install Kafka (Same with both methods)

1) Download and Unpack Kafka

```
ycao@kafkapython-m:~$ wget https://dlcdn.apache.org/kafka/3.3.1/kafka_2.13-3.3.1.tgz
--2022-12-01 23:45:04--  https://dlcdn.apache.org/kafka/3.3.1/kafka_2.13-3.3.1.tgz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 105053134 (100M) [application/x-gzip]
Saving to: 'kafka_2.13-3.3.1.tgz'

kafka_2.13-3.3.1.tgz          100%[=============================================>] 100.19M   226MB/s    in 0.4s

2022-12-01 23:45:05 (226 MB/s) - 'kafka_2.13-3.3.1.tgz' saved [105053134/105053134]

ycao@kafkapython-m:~$
```

```
ycao@kafkapython-m:~$ ls
kafka_2.13-3.3.1.tgz
ycao@kafkapython-m:~$ tar -xzf kafka_2.13-3.3.1.tgz
ycao@kafkapython-m:~$ ls
kafka_2.13-3.3.1  kafka_2.13-3.3.1.tgz
ycao@kafkapython-m:~$ cd kafka_2.13-3.3.1
ycao@kafkapython-m:~/kafka_2.13-3.3.1$
```

Step 3: Starting Kafka and Test with Example

1) Pre-requisites test

NOTE: Your local environment must have Java 8+ installed.

```
ycao@kafkapython-m:~/kafka_2.13-3.3.1$ java -version
openjdk version "1.8.0_352"
OpenJDK Runtime Environment (Temurin)(build 1.8.0_352-b08)
OpenJDK 64-Bit Server VM (Temurin)(build 25.352-b08, mixed mode)
ycao@kafkapython-m:~/kafka_2.13-3.3.1$
```

2) Start zookeeper (Terminal 1)

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

```
r.server.ZKDatabase)
[2022-12-01 23:47:51,404] INFO Snapshotting: 0x0 to /tmp/zookeeper/version-2/snapshot.0 (org.apache.zookeeper.server.persistence.FileTxnSnapLog)
[2022-12-01 23:47:51,404] INFO Snapshot taken in 1 ms (org.apache.zookeeper.server.ZooKeeperServer)
[2022-12-01 23:47:51,435] INFO zookeeper.request_throttler.shutdownTimeout = 10000 (org.apache.zookeeper.server.RequestThrottler)
[2022-12-01 23:47:51,436] INFO PrepRequestProcessor (sid:0) started, reconfigEnabled=false (org.apache.zookeeper.server.PrepRequestProcessor)
[2022-12-01 23:47:51,479] INFO Using checkIntervalMs=60000 maxPerMinute=10000 maxNeverUsedIntervalMs=0 (org.apache.zookeeper.server.ContainerManager)
[2022-12-01 23:47:51,480] INFO ZooKeeper audit is disabled. (org.apache.zookeeper.audit.ZKAuditProvider)
[2022-12-01 23:48:48,397] INFO Creating new log file: log.1 (org.apache.zookeeper.server.persistence.FileTxnLog)
```

3) Open another terminal session and start Kafka server(Terminal 2)

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

```
ka.network.SocketServer)
[2022-12-02 06:14:44,364] INFO Kafka version: 3.3.1 (org.apache.kafka.common.utils.AppInfoParser)
[2022-12-02 06:14:44,364] INFO Kafka commitId: e23c59d00e687ff5 (org.apache.kafka.common.utils.AppInfoParser)
[2022-12-02 06:14:44,365] INFO Kafka startTimeMs: 1669961684362 (org.apache.kafka.common.utils.AppInfoParser)
[2022-12-02 06:14:44,369] INFO [KafkaServer id=0] started (kafka.server.KafkaServer)
[2022-12-02 06:14:44,428] INFO [BrokerToControllerChannelManager broker=0 name=forwarding]: Recorded new control
ler, from now on will use broker kafkapython-m.us-west1-c.c.cs570-big-data-363104.internal:9092 (id: 0 rack: nul
l) (kafka.server.BrokerToControllerRequestThread)
[2022-12-02 06:14:44,500] INFO [BrokerToControllerChannelManager broker=0 name=alterPartition]: Recorded new con
troller, from now on will use broker kafkapython-m.us-west1-c.c.cs570-big-data-363104.internal:9092 (id: 0 rack:
 null) (kafka.server.BrokerToControllerRequestThread)
```

Once all services have successfully launched, you will have a basic Kafka environment running and ready to use.

4) Topics (Open another terminal: Terminal 3)

➔ Create topics

```
$ bin/kafka-topics.sh --create --topic input_recommend_product --bootstrap-server localhost:9092 –partition 3 –replication-factor 1
```

```
ycao@kafkapython-m:~/kafka_2.13-3.3.1$ bin/kafka-topics.sh --create --topic input_recommend_product --bootstrap-server
 localhost:9092 --partitions 3 --replication-factor 1
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could collide. To avoid is
sues it is best to use either, but not both.
Created topic input_recommend_product.
ycao@kafkapython-m:~/kafka_2.13-3.3.1$
```

Another example:

```
ycao@kafkapython-m:~$ ls
kafka_2.13-3.3.1  kafka_2.13-3.3.1.tgz
ycao@kafkapython-m:~$ cd kafka_2.13-3.3.1
ycao@kafkapython-m:~/kafka_2.13-3.3.1$ bin/kafka-topics.sh --create --topic quickstart-events --bootstrap-server localh
ost:9092
Created topic quickstart-events.
ycao@kafkapython-m:~/kafka_2.13-3.3.1$
```

➔ To check details of topic

```
$ bin/kafka-topics.sh --describe --topic quickstart-events --bootstrap-server localhost:9092
```

```
ycao@kafkapython-m:~/kafka_2.13-3.3.1$ bin/kafka-topics.sh --describe --topic quickstart-events --bootstrap-server loca
lhost:9092
Topic: quickstart-events        TopicId: iGzxKEnpTZ-ds3g5r-xUGQ PartitionCount: 1       ReplicationFactor: 1    Configs
:
        Topic: quickstart-events        Partition: 0    Leader: 0       Replicas: 0     Isr: 0
ycao@kafkapython-m:~/kafka_2.13-3.3.1$
```

➔ To check list of topic

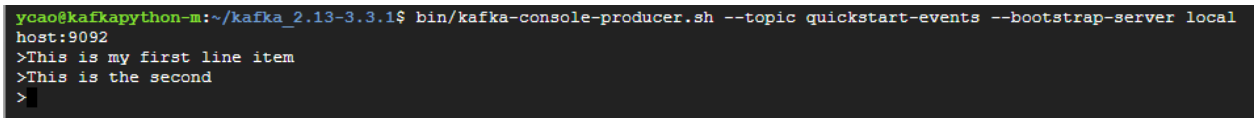
```
$ bin/kafka-topics.sh --list --bootstrap-server localhost:9092
```

5) Example Demo

Example 1: quickstart-evernts

➔ Write event

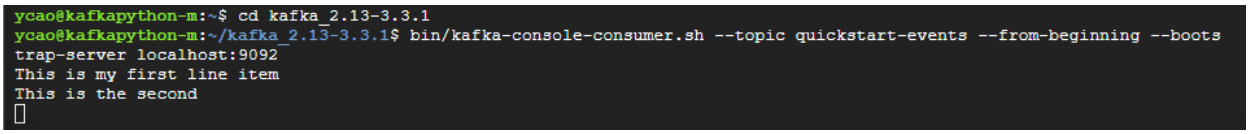
```
$ bin/kafka-console-producer.sh --topic quickstart-events --bootstrap-server localhost:9092
This is my first event
This is my second event
```



➔ Read event (Open another terminal: Terminal 4)

```
$ bin/kafka-console-consumer.sh --topic quickstart-events --from-beginning --bootstrap-server localhost:9092
This is my first event
This is my second event
```

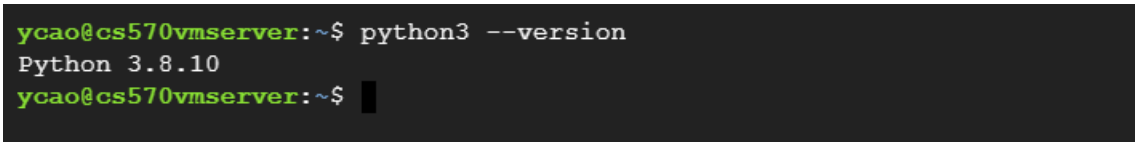
You can stop the consumer client with `Ctrl-C` at any time.



Example 2: input_recomment_product(Kafka-Python)

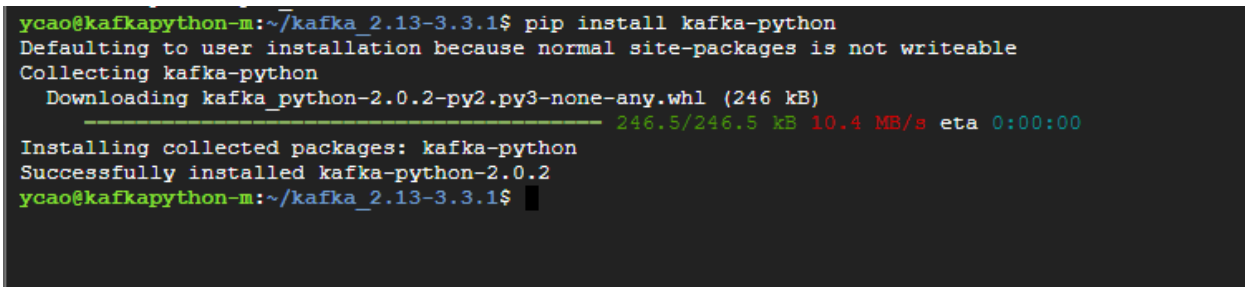
➔ With zookeeper and kafka terminal opened

➔ Check Ptyhon3 is installed



➔ Open another terminal (Terminal 3) and install kafka-python

$ pip install kafka-python



➔ Open python3 shell and start to type consumer.py

SSH-in-browser UPLOAD FILE DOWNLOAD FILE

```
from kafka import KafkaConsumer

consumer = KafkaConsumer('input_recommend_product', bootstrap_servers=['localhost:9092'])
for msg in consumer:
    print(msg.value)
~
~
~
~
~
~
~
```

```
ycao@cs570vmserver:~$ python3
Python 3.8.10 (default, Jun 22 2022, 20:18:18)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from kafka import KafkaConsumer
>>>
>>> consumer = KafkaConsumer('input_recommend_product', bootstrap_servers=['localhost:9092'])
>>> for msg in consumer:
...     print(msg.value)
...
b'(1, Main Menu), (2, Phone) , (3, Smart Phone), (4, iPhone)'
b'This is the end of testing for input_recommend_product topic.'
```

➔ Open another terminal (Terminal 4), then python3 shell and start to type producer.py

SSH-in-browser UPLOAD FILE DOWNLOAD FILE

```
from kafka import KafkaProducer

producer = KafkaProducer(bootstrap_servers='localhost:9092')

producer.send('input_recommend_product', b'(1, Main Menu), (2, Phone) , (3, Smart Phone), (4, iPhone)')

~
~
~
~
```

```
ycao@cs570vmserver:~$ python3
Python 3.8.10 (default, Jun 22 2022, 20:18:18)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from kafka import KafkaProducer
>>> producer = KafkaProducer(bootstrap_servers='localhost:9092')
>>> producer.send('input_recommend_product', b'(1, Main Menu), (2, Phone) , (3, Smart Phone), (4, iPhone)')
<kafka.producer.future.FutureRecordMetadata object at 0x7ff028cb3c70>
>>> producer.send('input_recommend_product', b'This is the end of testing for input_recommend_product topic.')
<kafka.producer.future.FutureRecordMetadata object at 0x7ff028ccd2e0>
>>>
```

Done!

Step 4: Spark Streaming

➔ Start NetCat

$nc -lk 9999

```
ycao@cs570vmserver:~$ nc -lk 9999
Hello world
What are you doing
Are you doing homework
Almost finish your homework
^Z
[1]+  Stopped                 nc -lk 9999
ycao@cs570vmserver:~$
```

➔ Open another terminal, start streaming in spark folder

$ ./bin/spark-submit examples/src/main/python/streaming/network_wordcount.py localhost 9999

```
22/12/09 03:21:54 INFO SparkContext: Running Spark version 3.3.1
22/12/09 03:21:54 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin
-java classes where applicable
22/12/09 03:21:54 INFO ResourceUtils: ==============================================================
22/12/09 03:21:54 INFO ResourceUtils: No custom resources configured for spark.driver.
22/12/09 03:21:54 INFO ResourceUtils: ==============================================================
22/12/09 03:21:54 INFO SparkContext: Submitted application: PythonStreamingNetworkWordCount
22/12/09 03:21:54 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name:
cores, amount: 1, script: , vendor: , memory -> name: memory, amount: 1024, script: , vendor: , offHeap -> name:
 offHeap, amount: 0, script: , vendor: ), task resources: Map(cpus -> name: cpus, amount: 1.0)
22/12/09 03:21:54 INFO ResourceProfile: Limiting resource is cpu
22/12/09 03:21:54 INFO ResourceProfileManager: Added ResourceProfile id: 0
22/12/09 03:21:54 INFO SecurityManager: Changing view acls to: ycao
22/12/09 03:21:54 INFO SecurityManager: Changing modify acls to: ycao
22/12/09 03:21:54 INFO SecurityManager: Changing view acls groups to:
22/12/09 03:21:54 INFO SecurityManager: Changing modify acls groups to:
22/12/09 03:21:54 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users  with
view permissions: Set(ycao); groups with view permissions: Set(); users  with modify permissions: Set(ycao); gro
ups with modify permissions: Set()
```

Step 5: Try Kafka-python streaming

1) Create the topics needed

```
ycao@kafkapython-m:~/kafka_2.13-3.3.1$ bin/kafka-topics.sh --create --topic input_event --bootstrap-server local
host:9092
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could collide. To av
oid issues it is best to use either, but not both.
Created topic input_event.
ycao@kafkapython-m:~/kafka_2.13-3.3.1$ bin/kafka-topics.sh --create --topic output_event --bootstrap-server loca
lhost:9092
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could collide. To av
oid issues it is best to use either, but not both.
Created topic output_event.
ycao@kafkapython-m:~/kafka_2.13-3.3.1$
```

```
ycao@kafkapython-m:~/kafka_2.13-3.3.1$ bin/kafka-topics.sh --list --bootstrap-server localhost:9092
input_event
output_event
ycao@kafkapython-m:~/kafka_2.13-3.3.1$
```

2) Create spark_script folder and download spark_streaming_kafka jar file

```
ycao@kafkapython-m:~$ cd spark_script
ycao@kafkapython-m:~/spark_script$ wget https://repo1.maven.org/maven2/org/apache/spark/spark-streaming-kafka-0-8_2.11/2.4.3/spark-streaming-kafka-0-8_2.11-2.4.3.jar
--2022-12-04 04:34:19--  https://repo1.maven.org/maven2/org/apache/spark/spark-streaming-kafka-0-8_2.11/2.4.3/spark-streaming-kafka-0-8_2.11-2.4.3.jar
Resolving repo1.maven.org (repo1.maven.org)... 199.232.192.209, 199.232.196.209
Connecting to repo1.maven.org (repo1.maven.org)|199.232.192.209|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 302964 (296K) [application/java-archive]
Saving to: 'spark-streaming-kafka-0-8_2.11-2.4.3.jar'

spark-streaming-kafka-0-8_2 100%[=========================================>] 295.86K  --.-KB/s    in 0.02s

2022-12-04 04:34:19 (14.1 MB/s) - 'spark-streaming-kafka-0-8_2.11-2.4.3.jar' saved [302964/302964]
```

3) Create spark_processor.py file

```
#create SC with the specified configuration
def spark_context_creator():
    conf = SparkConf()
    #set name for our app
    conf.setAppName("ConnectingDotsSparkKafkaStreaming")
    #The master URL to connect
    conf.setMaster('spark://abc.def.ghi.jkl:7077')
    sc = None
    try:
        sc.stop()
        sc = SparkContext(conf=conf)
    except:
        sc = SparkContext(conf=conf)
    return sc

sc = spark_context_creator()
#To avoid unncessary logs
sc.setLogLevel("WARN")

#batch duration, here i process for each second
ssc = StreamingContext(sc,1)

kafkaStream = KafkaUtils.createStream(ssc, 'abc.def.ghi.jkl:2181', 'test-consumer-group', {'input_event':1})

#processing each micro batch
def process_events(event):
    return (event[0], Counter(event[1].split(" ")).most_common(3))

lines = kafkaStream.map(lambda x : process_events(x))

producer = KafkaProducer(bootstrap_servers='abc.def.com:9092', value_serializer=str.encode, key_serializer=str.encode)

#push the processed event to Kafka
def push_back_to_kafka(processed_events):
    list_of_processed_events = processed_events.collect()
    producer.send('output_event', value = str(list_of_processed_events))

lines.foreachRDD(push_back_to_kafka)
~
```

4) Run spark_processor.py file

```
$ spark-submit --jars /spark_script/spark-streaming-kafka-0-8-assembly_2.11-2.4.8.jar --packages org.apache.spark:spark-streaming-kafka-0-8_2.11:2.1.2 org.apache.spark:spark-sql-kafka-0-10_2.12:3.3.1  --deploy-mode client spark_script/spark_processor.py
```

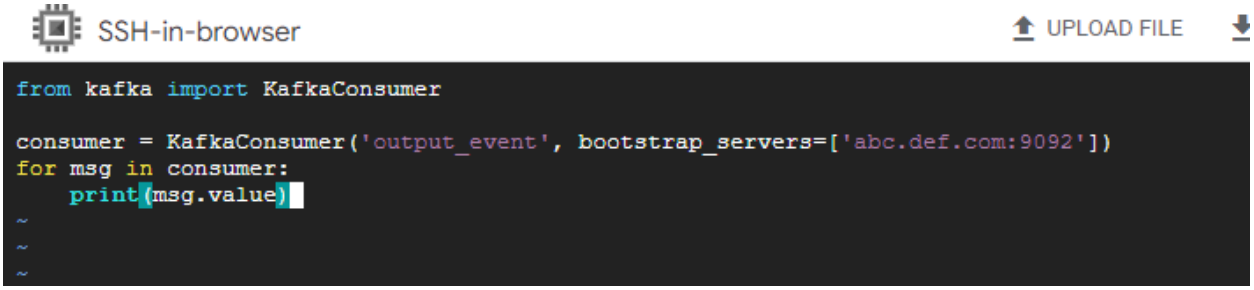
*Added some package to solve errors but still not able to fixed all

```
ycao@kafkapython-m:~$ spark-submit --jars /spark_script/spark-streaming-kafka-0-8-assembly_2.11-2.4.8.jar --packages o
rg.apache.spark:spark-streaming-kafka-0-8_2.11:2.1.2 org.apache.spark:spark-sql-kafka-0-10_2.12:3.3.1  --deploy-mode c
lient spark_script/spark_processor.py
:: loading settings :: url = jar:file:/usr/lib/spark/jars/ivy-2.4.0.jar!/org/apache/ivy/core/settings/ivysettings.xml
Ivy Default Cache set to: /home/ycao/.ivy2/cache
The jars for the packages stored in: /home/ycao/.ivy2/jars
org.apache.spark#spark-streaming-kafka-0-8_2.11 added as a dependency
:: resolving dependencies :: org.apache.spark#spark-submit-parent-a504dec0-10db-4767-8823-b329576d3974;1.0
        confs: [default]
        found org.apache.spark#spark-streaming-kafka-0-8_2.11;2.1.2 in central
        found org.apache.kafka#kafka_2.11;0.8.2.1 in central
        found org.scala-lang.modules#scala-xml_2.11;1.0.2 in central
        found com.yammer.metrics#metrics-core;2.2.0 in central
        found org.slf4j#slf4j-api;1.7.16 in central
        found org.scala-lang.modules#scala-parser-combinators_2.11;1.0.2 in central
        found com.101tec#zkclient;0.3 in central
        found log4j#log4j;1.2.17 in central
        found org.apache.kafka#kafka-clients;0.8.2.1 in central
        found net.jpountz.lz4#lz4;1.3.0 in central
        found org.xerial.snappy#snappy-java;1.1.2.6 in central
        found org.spark-project.spark#unused;1.0.0 in central
:: resolution report :: resolve 1999ms :: artifacts dl 26ms
        :: modules in use:
        com.101tec#zkclient;0.3 from central in [default]
        com.yammer.metrics#metrics-core;2.2.0 from central in [default]
        log4j#log4j;1.2.17 from central in [default]
        net.jpountz.lz4#lz4;1.3.0 from central in [default]
        org.apache.kafka#kafka-clients;0.8.2.1 from central in [default]
        org.apache.kafka#kafka_2.11;0.8.2.1 from central in [default]
        org.apache.spark#spark-streaming-kafka-0-8_2.11;2.1.2 from central in [default]
        org.scala-lang.modules#scala-parser-combinators_2.11;1.0.2 from central in [default]
        org.scala-lang.modules#scala-xml_2.11;1.0.2 from central in [default]
        org.slf4j#slf4j-api;1.7.16 from central in [default]
        org.spark-project.spark#unused;1.0.0 from central in [default]
        org.xerial.snappy#snappy-java;1.1.2.6 from central in [default]
        ---------------------------------------------------------------------
        |                  |            modules            ||   artifacts   |
        |       conf       | number| search|dwnlded|evicted|| number|dwnlded|
        ---------------------------------------------------------------------
        |      default     |   12  |   1   |   1   |   0   ||   12  |   0   |
        ---------------------------------------------------------------------
:: retrieving :: org.apache.spark#spark-submit-parent-a504dec0-10db-4767-8823-b329576d3974
        confs: [default]
        0 artifacts copied, 12 already retrieved (0kB/25ms)
Exception in thread "main" java.lang.NullPointerException
        at org.apache.hadoop.fs.Path.getName(Path.java:418)
        at org.apache.spark.deploy.DependencyUtils$.downloadFile(DependencyUtils.scala:136)
        at org.apache.spark.deploy.SparkSubmit.$anonfun$prepareSubmitEnvironment$8(SparkSubmit.scala:376)
        at scala.Option.map(Option.scala:230)
        at org.apache.spark.deploy.SparkSubmit.prepareSubmitEnvironment(SparkSubmit.scala:376)
        at org.apache.spark.deploy.SparkSubmit.org$apache$spark$deploy$SparkSubmit$$runMain(SparkSubmit.scala:894)
        at org.apache.spark.deploy.SparkSubmit.doRunMain$1(SparkSubmit.scala:180)
        at org.apache.spark.deploy.SparkSubmit.submit(SparkSubmit.scala:203)
        at org.apache.spark.deploy.SparkSubmit.doSubmit(SparkSubmit.scala:90)
        at org.apache.spark.deploy.SparkSubmit$$anon$2.doSubmit(SparkSubmit.scala:1039)
        at org.apache.spark.deploy.SparkSubmit$.main(SparkSubmit.scala:1048)
        at org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)
ycao@kafkapython-m:~$
```

➔ Producer.py

```
from kafka import KafkaProducer

producer = KafkaProducer(bootstrap_servers='abc.def.com:9092', value_serializer=str.encode, key_serializer=str.encode)
event_stream_key = 'product_list'
event_stream_value = 'product1 product2 product3 product1'
producer.send('input_event', key = event_stream_key, value = event_stream_value)
~
~
~
~
~
```

➔ Consumer.py


SSH-in-browser
UPLOAD FILE
from kafka import KafkaConsumer
consumer = KafkaConsumer('output_event', bootstrap_servers=['abc.def.com:9092'])
for msg in consumer:
print(msg.value)