Mini project1 : Building a Data Ingestion Pipeline

1.Set Up the Environment

I have already these frameworks installed in my VM

2. Data Migration with Apache Sqoop and HDFS:

1. Create a local MySQL RDBMS database and populate it with sample data.

Connecting to MySQL:

1.Log in to MySQL using the following command:

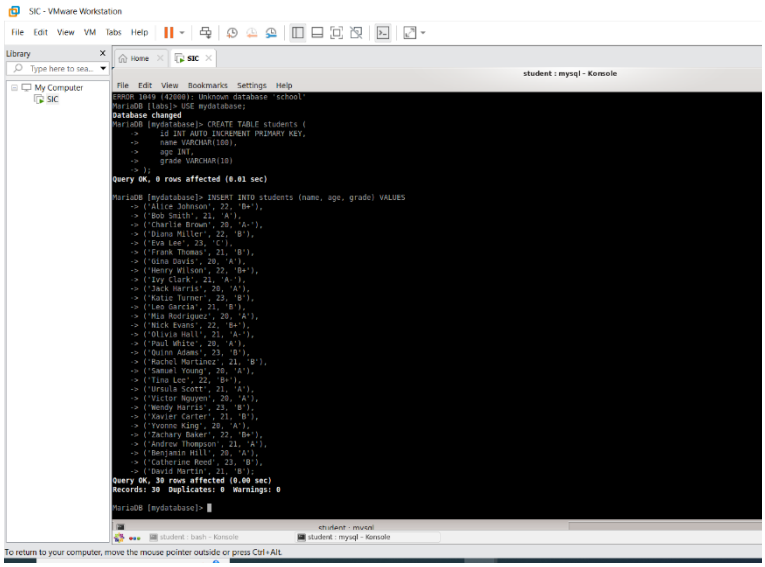
```
mysql --user=student --password=student labs
```

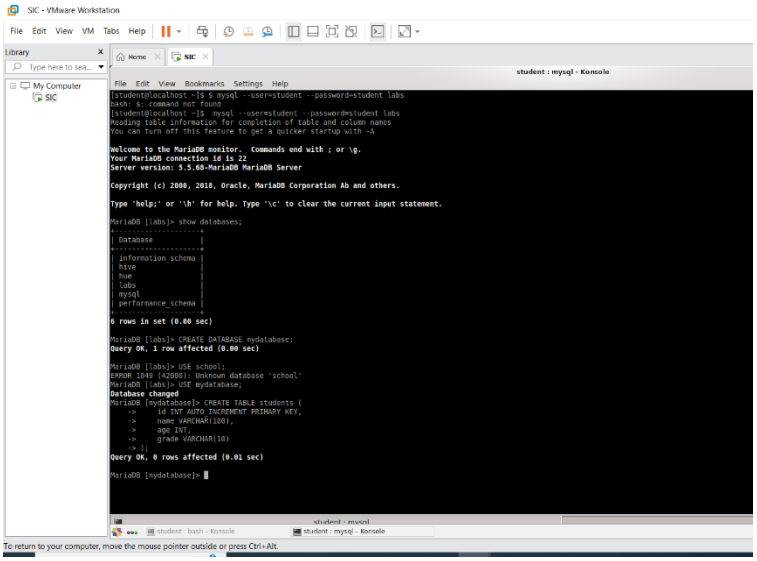
2. Creating a New Database in MariaDB:

```
CREATE DATABASE mydatabase;
```

3. Creating a Table and Adding Data:

```
USE mydatabase;

CREATE TABLE students (

  id INT AUTO_INCREMENT PRIMARY KEY,

  name VARCHAR(100),

  age INT,

  city VARCHAR(100)

);

INSERT INTO students (name, age, city) VALUES

  ('John Doe', 25, 'New York'),

  ('Jane Smith', 22, 'Los Angeles'),

  ('Ahmed Ali', 28, 'Cairo');
```



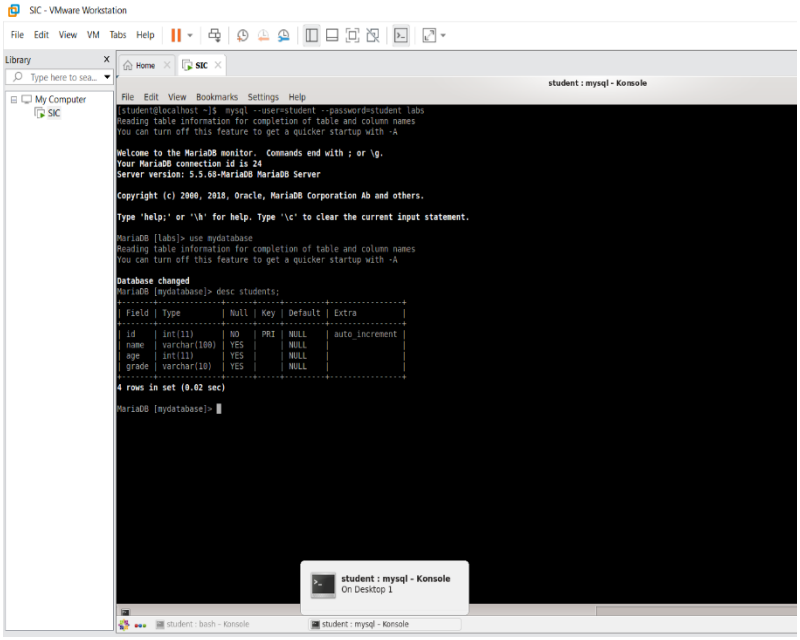


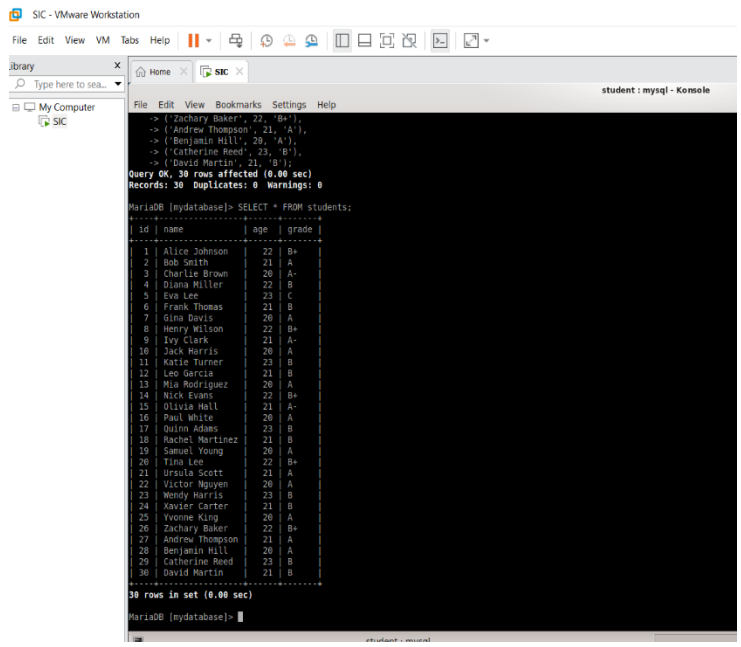
4. Viewing Data in the Table (SELECT Statement):

```
SELECT * FROM students;
```

5. Describing the Table

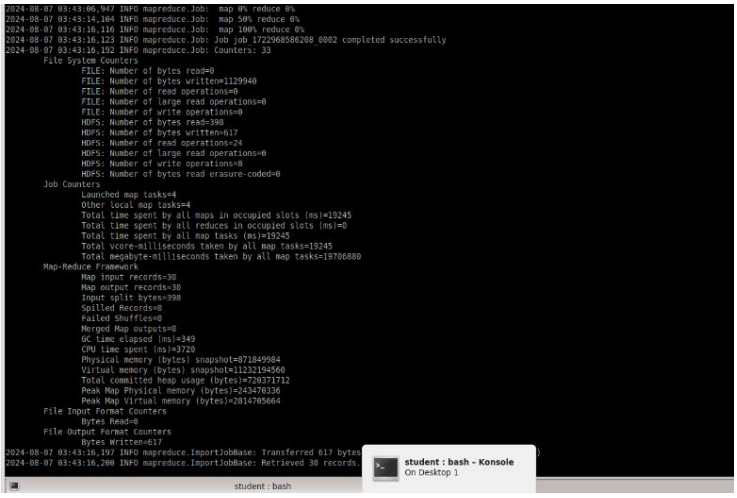
```
DESC students;
```

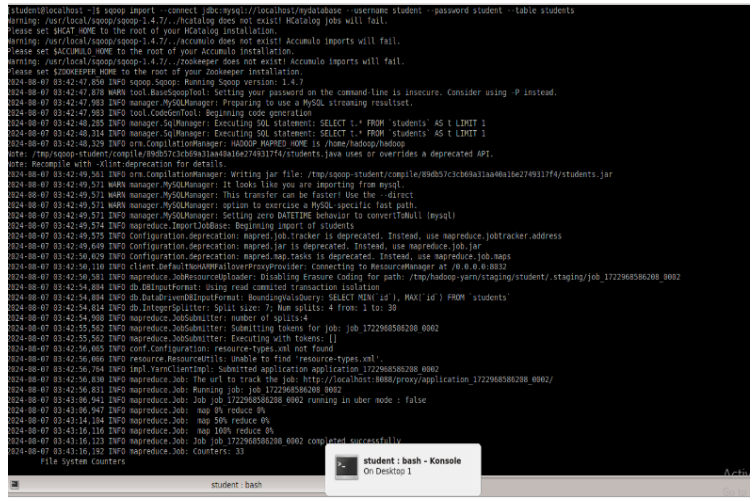




2. Writing Sqoop commands to import data from the local database to HDFS

```
$ sqoop import-connect jdbc:mysql://localhost/mydatabase --username student --password student --table students
```



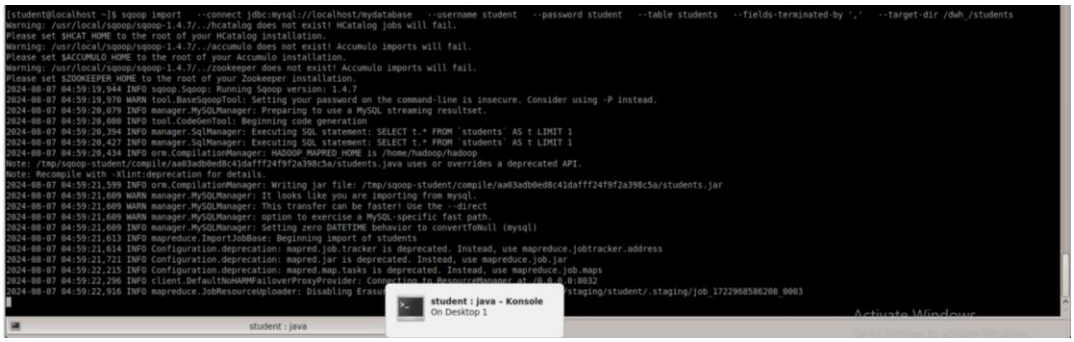


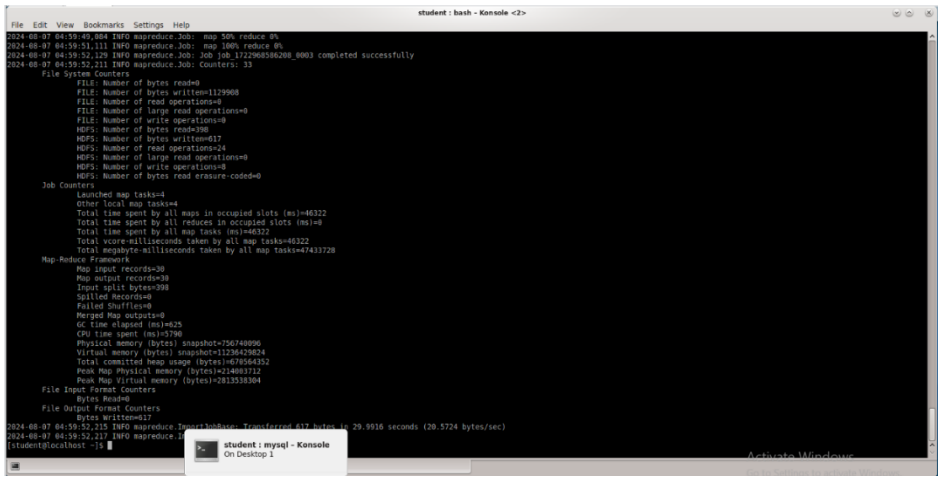
- Making target directory in Hadoop HDFS

```
$ hdfs dfs -Mkdir /dwh_
```

- Import database to hdfs target directory

```
$ sqoop import \
 --connect jdbc:mysql://localhost/mydatabase \
 --username student \
 --password student \
 --table students \
 --fields-terminated-by ',' \
 --target-dir /dwh_/students
```





- checking whether it was imported or not

hdfs dfs -ls /dwh_/students

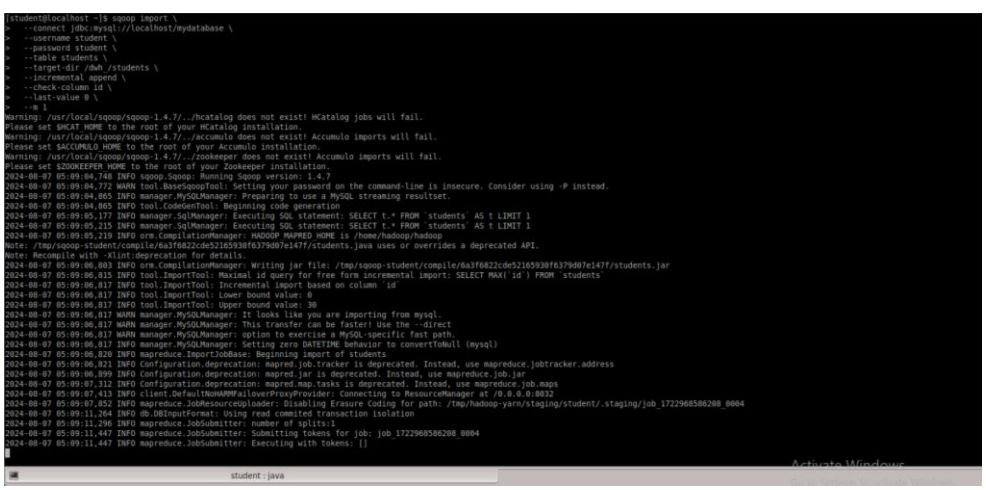
```
[student@localhost ~]$ hdfs dfs -ls /dwh_/students
Found 5 items
-rw-r--r--   1 student supergroup          0 2024-08-07 04:59 /dwh_/students/_SUCCESS
-rw-r--r--   1 student supergroup        155 2024-08-07 04:59 /dwh_/students/part-m-00000
-rw-r--r--   1 student supergroup        141 2024-08-07 04:59 /dwh_/students/part-m-00001
-rw-r--r--   1 student supergroup        145 2024-08-07 04:59 /dwh_/students/part-m-00002
-rw-r--r--   1 student supergroup        176 2024-08-07 04:59 /dwh_/students/part-m-00003
[student@localhost ~]$
```

- open any part and show the data

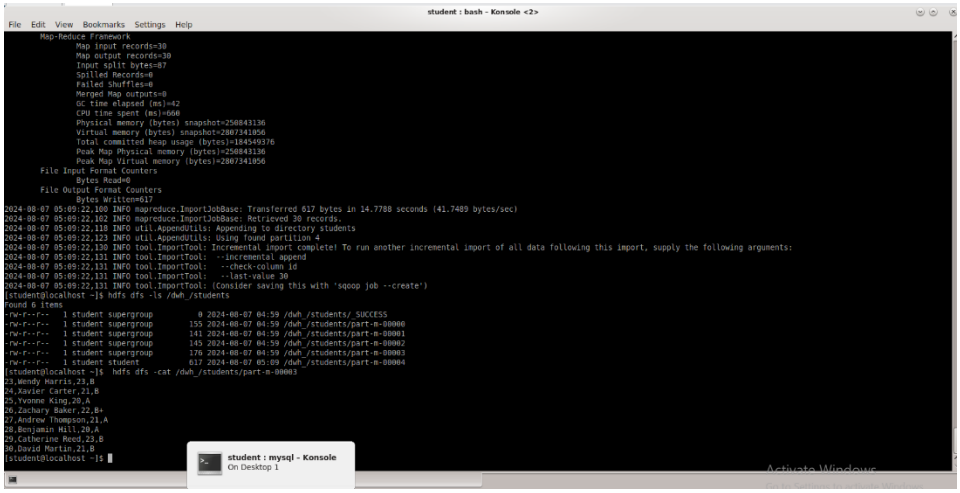
hdfs dfs -cat /dwh_/students/part-m-00001

```
[student@localhost ~]$  hdfs dfs -cat /dwh_/students/part-m-00001
9,Ivy Clark,21,A-
10,Jack Harris,20,A
11,Katie Turner,23,B
12,Leo Garcia,21,B
13,Mia Rodriguez,20,A
14,Nick Evans,22,B+
15,Olivia Hall,21,A-
[student@localhost ~]$
```

- Set up incremental import with Sqoop to handle new data entries.



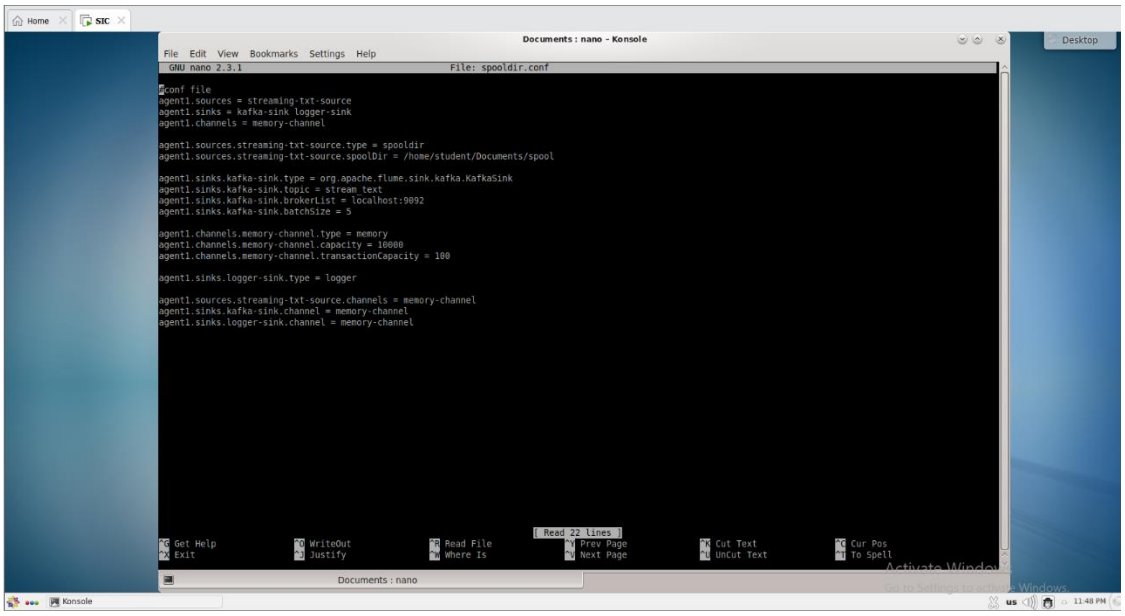
3. Real-Time Data Ingestion with Apache Flume and Apache Kafka:



setting up flume to send data to kafka topic

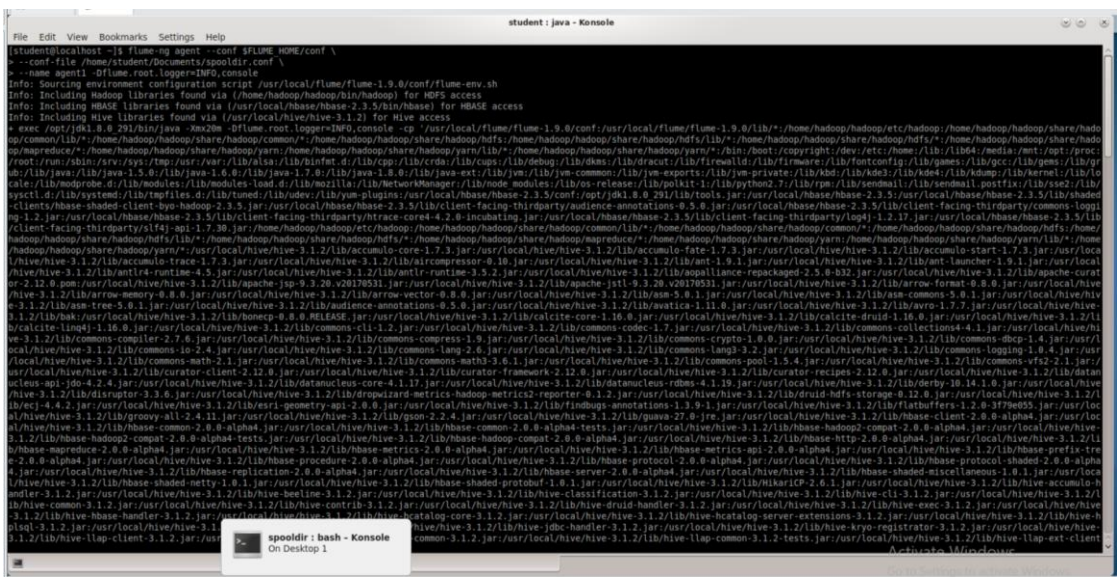
- first i have to configure the agent that is responsible for getting the data source

```
#conf file

agent1.sources = streaming-txt-source

agent1.sinks = kafka-sink logger-sink

agent1.channels = memory-channel

agent1.sources.streaming-txt-source.type = spooldir

agent1.sources.streaming-txt-source.spoolDir = /home/student/Documents/spool

agent1.sinks.kafka-sink.type = org.apache.flume.sink.kafka.KafkaSink

agent1.sinks.kafka-sink.topic = stream_text

agent1.sinks.kafka-sink.brokerList = localhost:9092

agent1.sinks.kafka-sink.batchSize = 5

agent1.channels.memory-channel.type = memory

agent1.channels.memory-channel.capacity = 10000

agent1.channels.memory-channel.transactionCapacity = 100

agent1.sinks.logger-sink.type = logger

agent1.sources.streaming-txt-source.channels = memory-channel

agent1.sinks.kafka-sink.channel = memory-channel

agent1.sinks.logger-sink.channel = memory-channel
```



i have used the following command to start the agent

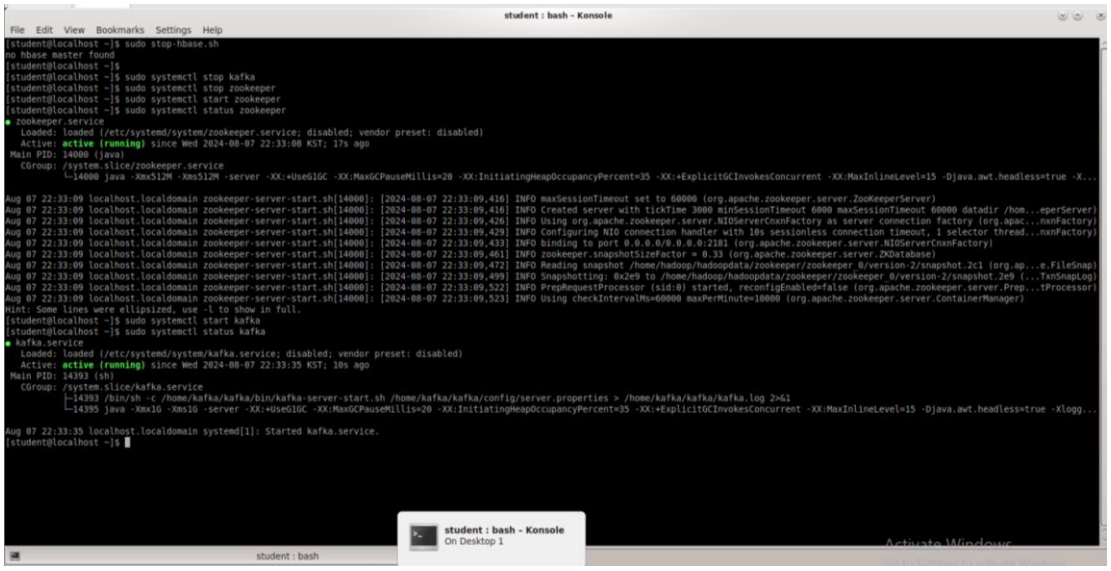
```
flume-ng agent \

--conf $FLUME_HOME/conf \

--conf-file /home/student/Documents/spooldir.conf \

--name agent1 -Dflume.root.logger=INFO,console
```



start the zookeepr service and Kafka service as well

```
sudo stop-hbase.sh

1.2. Restart Kafka and Zookeeper

sudo systemctl stop kafka
```

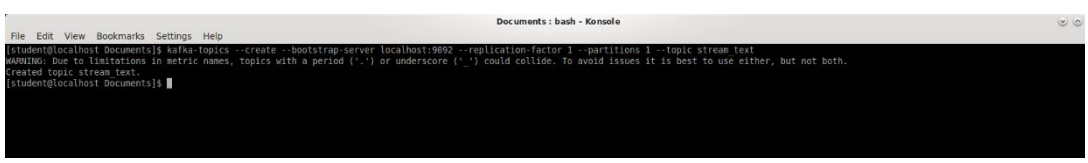
```
sudo systemctl stop zookeeper

sudo systemctl start zookeeper

sudo systemctl status zookeeper

sudo systemctl start kafka

sudo systemctl status kafka
```



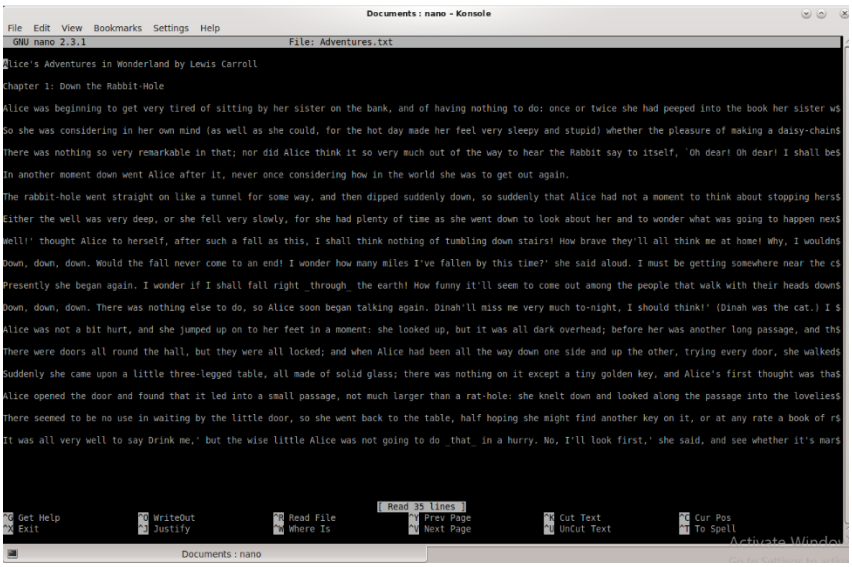
Create destination directory to collect streaming text

```
[student@localhost Documents]$ mkdir spool
[student@localhost Documents]$ ls
spool  spooldir.conf  spool_stream.py
[student@localhost Documents]$
```

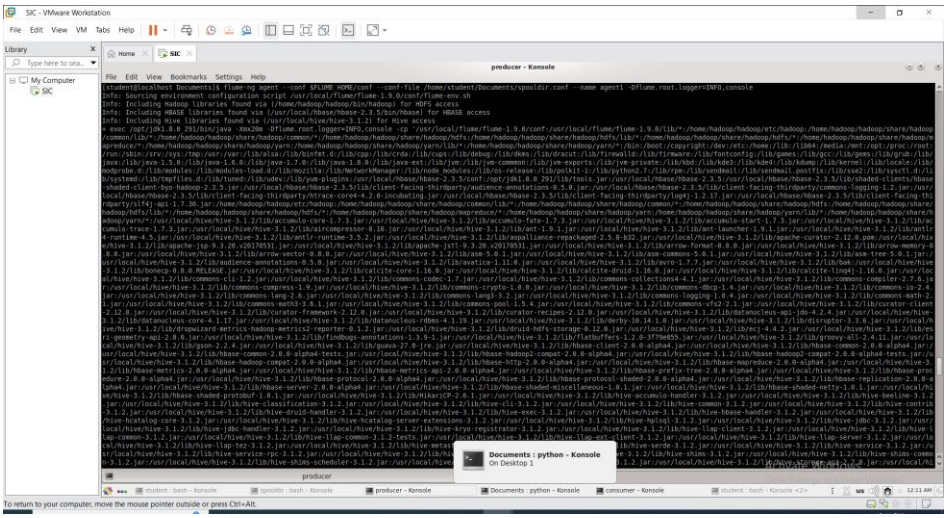
Let's create Kafka topic and name it **stream_text**

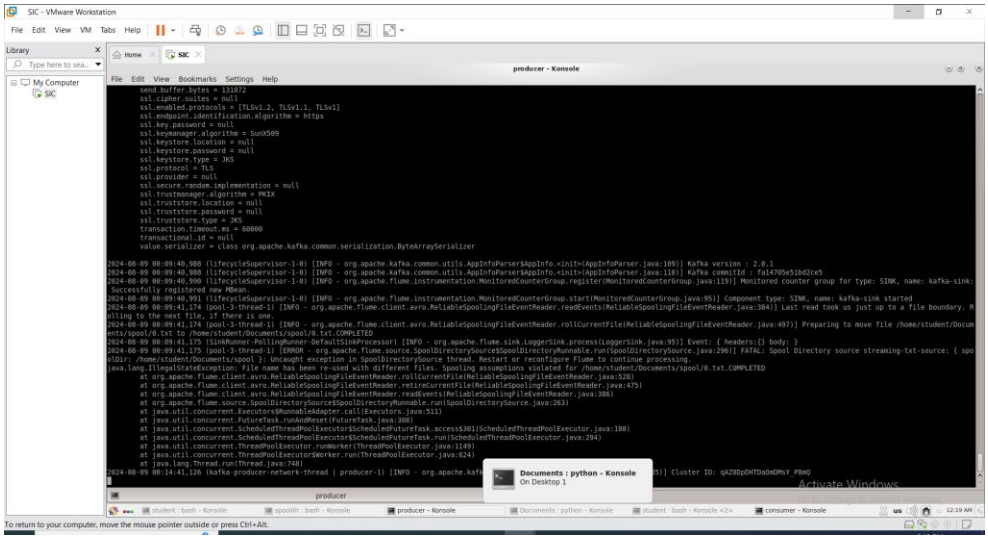


This is the data that I will send it to kafka

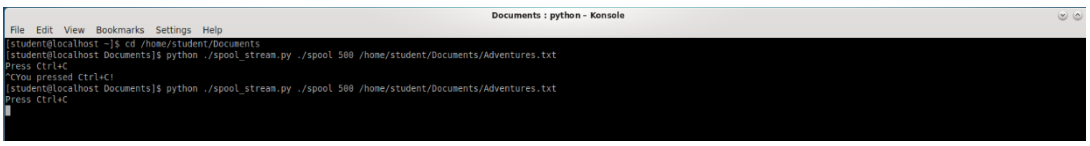


In another terminal We will navigate to /home/Documents/ to run producer





In another terminal We will navigate to /home/Documents to run python script to send our streaming data



In a new terminal I called it consumer, we will create Kafka consumer to receive messages from the server.

With this command

```
kafka-console-consumer --bootstrap-server localhost:9092 --topic stream_text --from-beginning
```

this is the data after run the pipeline