Lab 10: Creating a Kafka Topic, Producer, and Consumer

In this lab, you run use kafka services on a command line to create a topic. We use Kafka to create producers and consumers and pass data through them.

1. Setup Apache Kafka

In order to reduce demand to our virtual machine, we shall stop HBase and run only Apache Kafka.

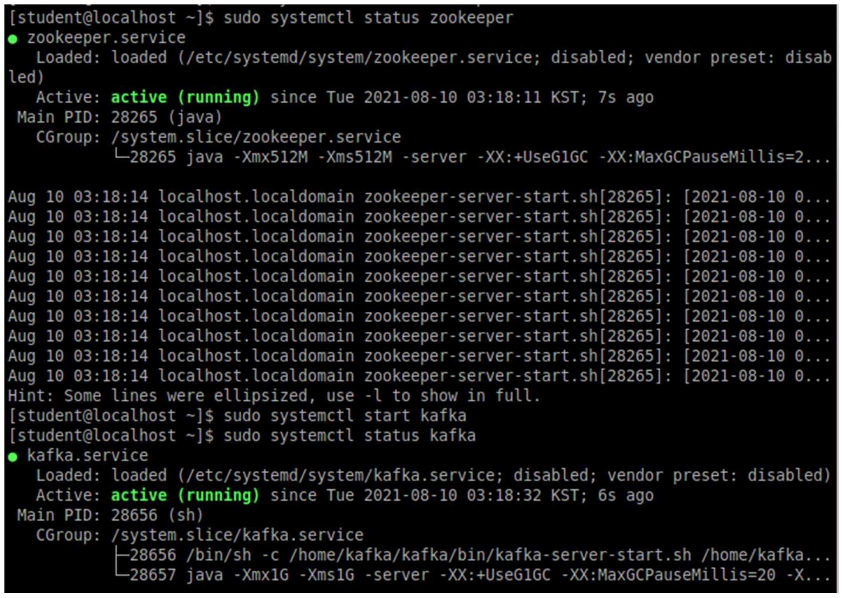
1.1. Stop HBase services

```
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
```

1.3. Make sure that both zookeeper and kaka is running. If not repeat above step.


```
[student@localhost ~]$ sudo systemctl status zookeeper
● zookeeper.service
   Loaded: loaded (/etc/systemd/system/zookeeper.service; disabled; vendor preset: disab
led)
   Active: active (running) since Tue 2021-08-10 03:18:11 KST; 7s ago
 Main PID: 28265 (java)
   CGroup: /system.slice/zookeeper.service
           └─28265 java -Xmx512M -Xms512M -server -XX:+UseG1GC -XX:MaxGCPauseMillis=2...

Aug 10 03:18:14 localhost.localdomain zookeeper-server-start.sh[28265]: [2021-08-10 0...
Aug 10 03:18:14 localhost.localdomain zookeeper-server-start.sh[28265]: [2021-08-10 0...
Aug 10 03:18:14 localhost.localdomain zookeeper-server-start.sh[28265]: [2021-08-10 0...
Aug 10 03:18:14 localhost.localdomain zookeeper-server-start.sh[28265]: [2021-08-10 0...
Aug 10 03:18:14 localhost.localdomain zookeeper-server-start.sh[28265]: [2021-08-10 0...
Aug 10 03:18:14 localhost.localdomain zookeeper-server-start.sh[28265]: [2021-08-10 0...
Aug 10 03:18:14 localhost.localdomain zookeeper-server-start.sh[28265]: [2021-08-10 0...
Aug 10 03:18:14 localhost.localdomain zookeeper-server-start.sh[28265]: [2021-08-10 0...
Aug 10 03:18:14 localhost.localdomain zookeeper-server-start.sh[28265]: [2021-08-10 0...
Aug 10 03:18:14 localhost.localdomain zookeeper-server-start.sh[28265]: [2021-08-10 0...
Hint: Some lines were ellipsized, use -l to show in full.
[student@localhost ~]$ sudo systemctl start kafka
[student@localhost ~]$ sudo systemctl status kafka
● kafka.service
   Loaded: loaded (/etc/systemd/system/kafka.service; disabled; vendor preset: disabled)
   Active: active (running) since Tue 2021-08-10 03:18:32 KST; 6s ago
 Main PID: 28656 (sh)
   CGroup: /system.slice/kafka.service
           ├─28656 /bin/sh -c /home/kafka/kafka/bin/kafka-server-start.sh /home/kafka...
           └─28657 java -Xmx1G -Xms1G -server -XX:+UseG1GC -XX:MaxGCPauseMillis=20 -X...
```

2. Creating a Kafka Topic

Create a Kafka topic named topic1_logs that will contain messages representing lines in log files.

2.1. Use kafka-topics to create a topic

2.1.1. Execute the following code from a terminal to create topic1_logs topic

```
$kafka-topics –create \
--bootstrap-server localhost:9092 \
--replication-factor 1\
--partitions 1\
--topic topic1_logs
```

You will see the message: Created topic “topic1_logs”.

Note:If you previously worked on an lab that used Kafka, you may get an error here indicating that this topic already exists. You may disregard the error.

2.1.2. Use the –list option to display all kafka topics and confirm that the new topic you just created is listed:

```
$ kafka-topics –list \
--bootstrap-server localhost:9092
```

2.2. Review the details of the topic1_logs.

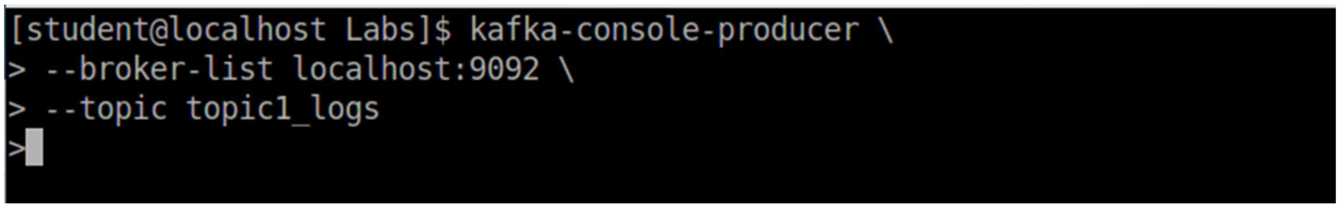
```
$kafka-topics –describe topic1_logs
--bootstrap-server localhost:9092
```

3. Create producers and consumers for a topic

   3.1. Open 2 terminals and create the producer on one and the consumer on the other.

      3.1.1. From the first terminal use kafka-console-producer command to start the producer.

```
$kafka-console-producer \
--broker-list localhost:9092 \
--topic topic1_logs
```

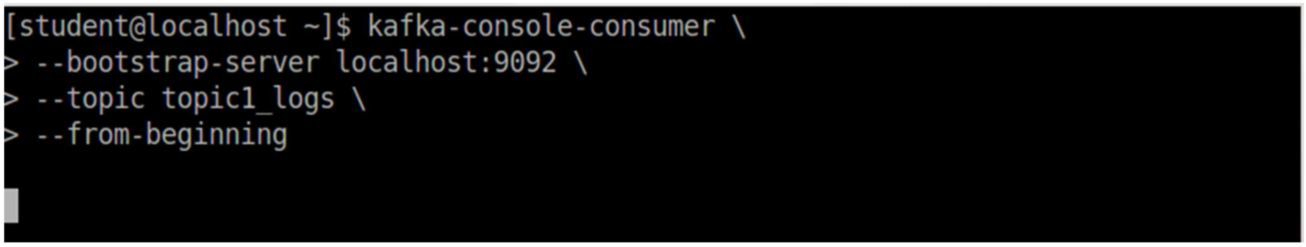


```
[student@localhost Labs]$ kafka-console-producer \
> --broker-list localhost:9092 \
> --topic topic1_logs
>
```

Notice that the kafka-console-producer is waiting for text to be typed in. Text that is type here will become a message in the Kafka topic topic1_logs.

      3.1.2. From the second terminal, use kafka-console-consumer to create a consumer for the topic

```
kafka-console-consumer \
--bootstrap-server localhost:9092 \
--topic topic1_logs \
--from-beginning
```



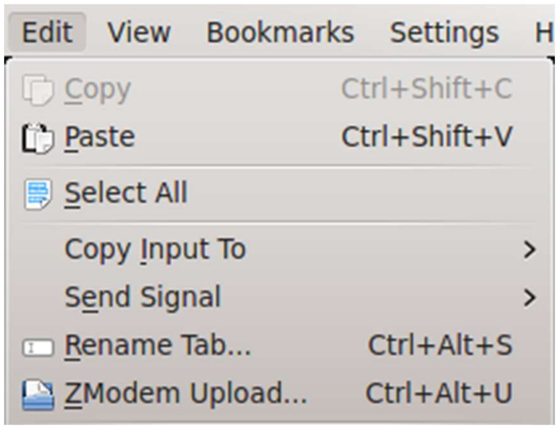
```
[student@localhost ~]$ kafka-console-consumer \
> --bootstrap-server localhost:9092 \
> --topic topic1_logs \
> --from-beginning
```

Notice that the kafka-console-consumer is waiting for messages to arrive at kafka topic topic1_logs.

3.2. Rename the terminals.

3.2.1. From the producer terminal, select Edit > Rename Tab and change the terminal tab Producer.

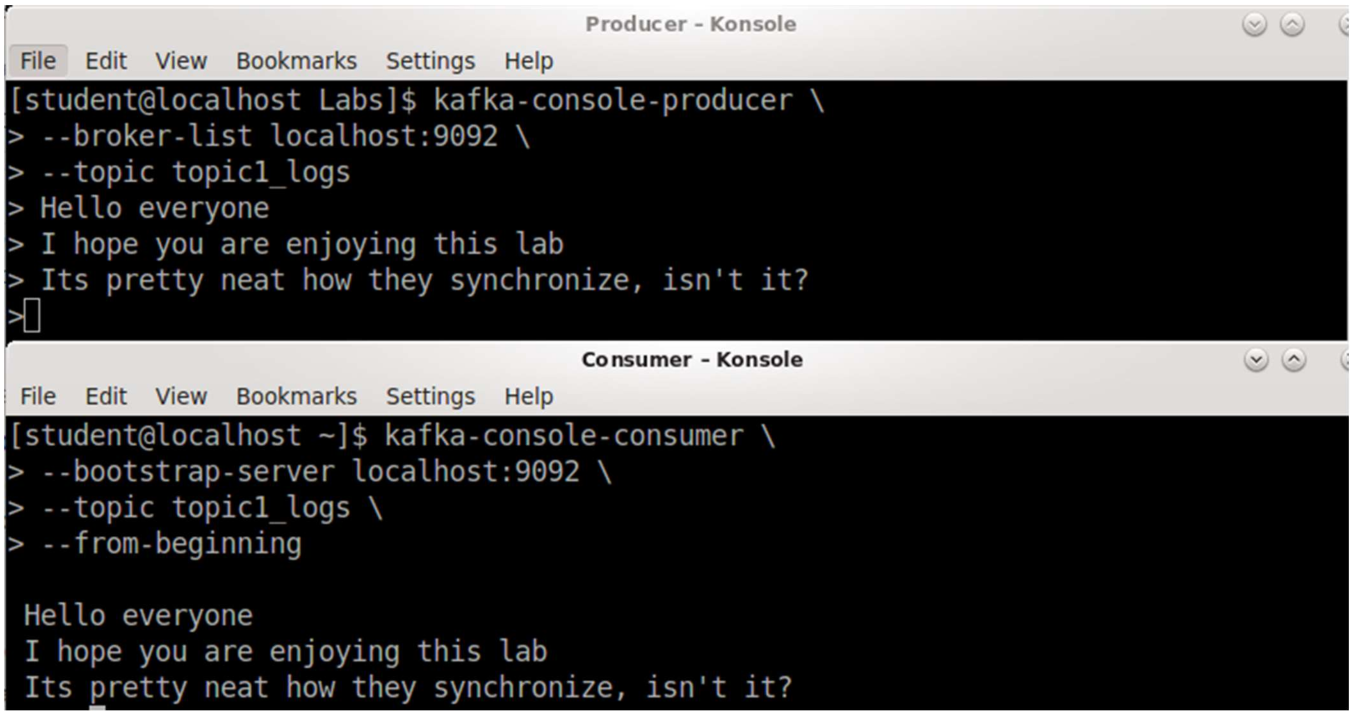
3.2.2. Do the same from the consumer terminal, naming it Consumer.



3.3. Produce messages for topic topic1_logs.

3.3.1. Begin typing something from the **Producer terminal** where the producer is running

3.3.2. Observe that in **Consumer terminal**, the consumer will pull messages that have been pushed by the producer.


Producer - Konsole
File Edit View Bookmarks Settings Help

```
[student@localhost Labs]$ kafka-console-producer \
> --broker-list localhost:9092 \
> --topic topic1_logs
> Hello everyone
> I hope you are enjoying this lab
> Its pretty neat how they synchronize, isn't it?
>
```

Consumer - Konsole
File Edit View Bookmarks Settings Help

```
[student@localhost ~]$ kafka-console-consumer \
> --bootstrap-server localhost:9092 \
> --topic topic1_logs \
> --from-beginning

 Hello everyone
 I hope you are enjoying this lab
 Its pretty neat how they synchronize, isn't it?
```

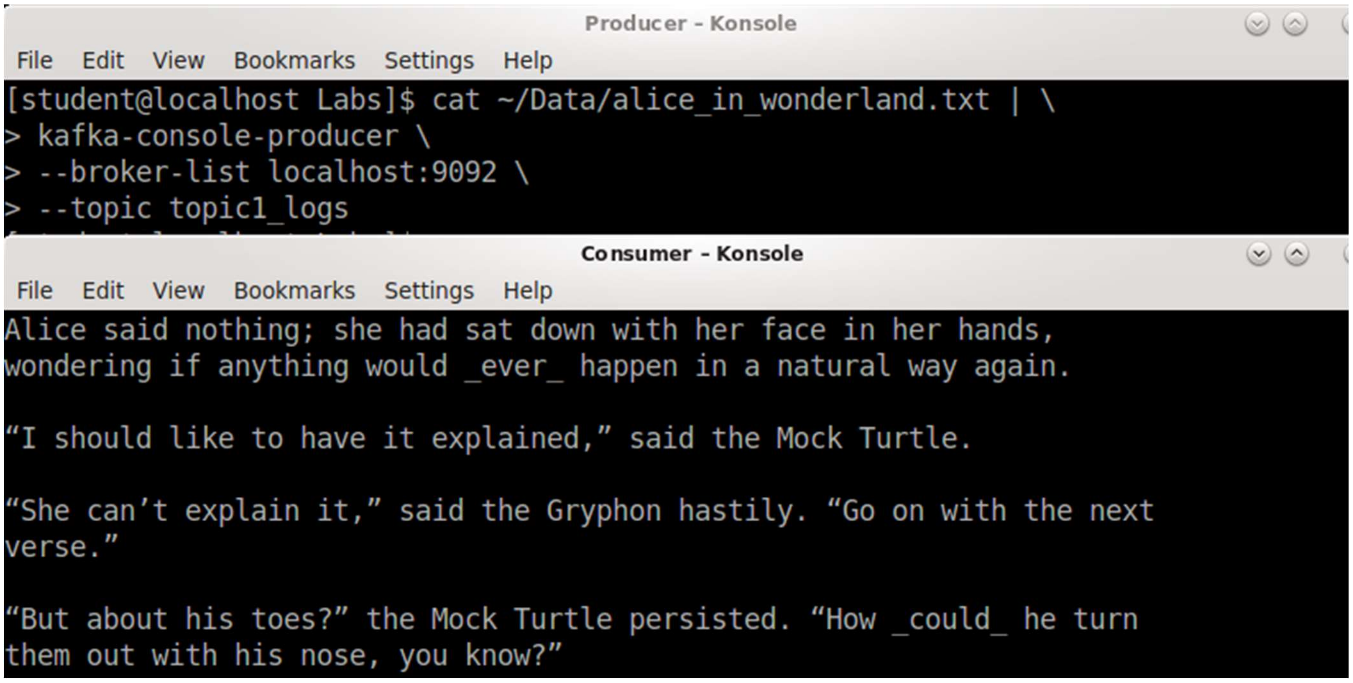
3.4. Create messages in batch mode.

3.4.1. From the Producer terminal, stop the Producer by sending a Ctlr-C KeyboardTerminate signal

3.4.2. Send the entire contents of Alice-in-Wonderland.txt file to the topic1_logs topic

```
cat ~/Data/alice_in_wonderland.txt | \
kafka-console-producer \
--broker-list localhost:9092 \
--topic topic1_logs
```

What happened? It went very fast. I hope you didn't blink. The entire content of the book "Alice in Wonderland" was passes as messages by the Producer and then picked up by the Consumer.


Producer - Konsole
File Edit View Bookmarks Settings Help

```
[student@localhost Labs]$ cat ~/Data/alice_in_wonderland.txt | \
> kafka-console-producer \
> --broker-list localhost:9092 \
> --topic topic1_logs
```

Consumer - Konsole
File Edit View Bookmarks Settings Help

```
Alice said nothing; she had sat down with her face in her hands,
wondering if anything would _ever_ happen in a natural way again.

“I should like to have it explained,” said the Mock Turtle.

“She can’t explain it,” said the Gryphon hastily. “Go on with the next
verse.”

“But about his toes?” the Mock Turtle persisted. “How _could_ he turn
them out with his nose, you know?”
```

3.5. Clean up

3.5.1. Stop producer and consumer as necessary using Ctrl-C kill signal.