Faculty of Engineering and Applied Science
Cloud Computing SOFE 4630U CRN: 74293
Lab 1

Due: February 15th 2022
Name: Iliya Karac
Student ID: 100703933

Github repository: https://github.com/cassidylinhares/cloud_lab2
Videos:
https://drive.google.com/drive/folders/12T3rEOcG5HhAQBWwqR7qzIgY_CCn_pbI?usp=sharing

Tasks:

1. Video watched
   Event driven architecture or EDA is software architecture that follows events. There are many different types of architectures like MVC, Restful, microservice, service oriented or message driven. In the EDA there are 3 main parts. Event producers, event consumers, and the data stream. The producers and consumers treat the data stream like a broker. They can publish and subscribe to a topic. The stream ingests events and holds them in chronological order like cache but it does not delete them when they become outdated.

2. Video seen
   Kafka is a hybrid between a controlled database and an event driven architecture with publishers and subscribers and a cluster that consists of brokers. The cluster tends to have a minimum of 3 brokers. The main server available for pub/sub, the zookeeper that controls ingestion of data and checks the status of all the servers to insure no issues while operating, and the last minimum requirement is the backup server that ingests the same data as the main broker and will take over if the main broker fails. The broker can hold several topics that are split into partitions that have a further subset for keeping track of message order. Lastly there are consumers/consumer groups that split consumption of messages from a topic based on their organization.

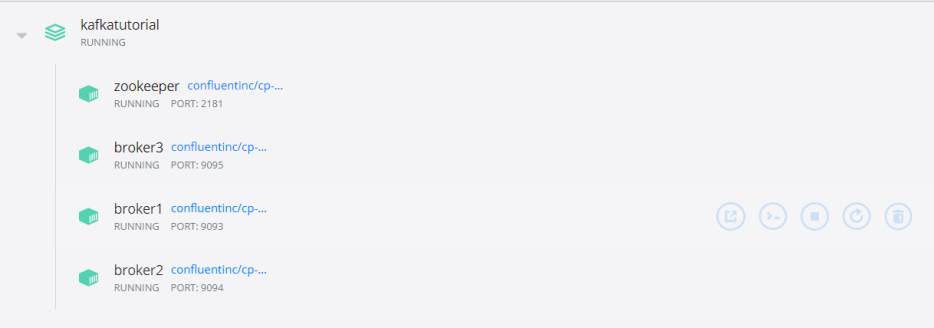
3. Answer the following questions

4. What is an EDA? This was answered in the first question. The advantages include a better alternative to a message based architecture assuming there is a feed of events from multiple producers. High throughput. Disk based storage allows for storage of events that number in the trillions. Scalable, fault tolerant, configurable, backpressure. The disadvantages are that EDA has an eventual limit to storage and past a certain amount of time old events have to be deleted due to storage constraints. Extremely complex to set up, sometimes overkill for certain applications. Does not work with realtime/low latency systems even with

only-once messaging.

5. **Cluster**- holds the structure of all the brokers and zookeeper
   Broker- a server in the cluster that typically has the job of retaining absorbing user information and allowing access to the consumers
   Topic-a title for a group of messages used to easily distinguish the data from the rest of the events in the broker. Producer specifies what topic it is publishing and consumer specifies what topic they are subscribed to.
   Replica- is a broker that mirrors another broker. They receive the same events. This is used as a backup in case of failure.
   Partition- topic is split up into different partitions to allow for chronological order of messages and a different subset for multiple consumers in the same group
   Zookeeper- in charge of the rest of the brokers, monitors their status and dictates data ingestion
   Controller- the server responsible for partitioning data and other administrative tasks
   Leader- main server for availability other servers depend on it for consistency
   Consumer- client that consumes data
   Producer- publishes events to a topic
   Consumer group- consists of 1 or more consumers, consume the data from a topic, consumers in a group tend to key into one partition to consume, if there is not enough consumers or one of them fails the rest will split up the remaining partitions amongst themselves

6. Video watched



C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.19042.1526]
(c) Microsoft Corporation. All rights reserved.

C:\Users\not_i\OneDrive\Desktop\kafka tutorial>docker exec broker1 kafka-topics --create --topic topic  --
rtitions 3 --replication-factor 3 --if-not-exists --bootstrap-server broker1:9092
Created topic topic.

C:\Users\not_i\OneDrive\Desktop\kafka tutorial>docker exec broker1 kafka-topics --create --topic topic2 --
rtitions 3 --replication-factor 2 --if-not-exists --bootstrap-server broker1:9092,broker2:9092,broker3:909
Created topic topic2.

C:\Users\not_i\OneDrive\Desktop\kafka tutorial>docker exec broker1 kafka-topics --describe --bootstrap-ser
r broker1:9092
Topic: topic    TopicId: hAIRX_DoSwSpJRf2WQu_pg PartitionCount: 3       ReplicationFactor: 3    Configs:
        Topic: topic    Partition: 0    Leader: 2       Replicas: 2,3,1 Isr: 2,3,1
        Topic: topic    Partition: 1    Leader: 3       Replicas: 3,1,2 Isr: 3,1,2
        Topic: topic    Partition: 2    Leader: 1       Replicas: 1,2,3 Isr: 1,2,3
Topic: topic2   TopicId: fMC6jMqcRki4FR_tYQc63g PartitionCount: 3       ReplicationFactor: 2    Configs:
        Topic: topic2   Partition: 0    Leader: 3       Replicas: 3,1   Isr: 3,1
        Topic: topic2   Partition: 1    Leader: 1       Replicas: 1,2   Isr: 1,2
        Topic: topic2   Partition: 2    Leader: 2       Replicas: 2,3   Isr: 2,3

C:\Users\not_i\OneDrive\Desktop\kafka tutorial>docker exec broker1 kafka-topics --list --bootstrap-server
oker1:9092
topic
topic2

C:\Users\not_i\OneDrive\Desktop\kafka tutorial>
```

C:\Windows\System32\cmd.exe - docker exec broker2 kafka-console-consumer --bootstrap-server broker1:9092 --to...

```
        Topic: topic    Partition: 2    Leader: 1       Replicas: 1,2,3 Isr: 1,2,3
Topic: topic2   TopicId: fMC6jMqcRki4FR_tYQc63g PartitionCount: 3       ReplicationFactor: 2    Configs:
        Topic: topic2   Partition: 0    Leader: 3       Replicas: 3,1   Isr: 3,1
        Topic: topic2   Partition: 1    Leader: 1       Replicas: 1,2   Isr: 1,2
        Topic: topic2   Partition: 2    Leader: 2       Replicas: 2,3   Isr: 2,3

C:\Users\not_i\OneDrive\Desktop\kafka tutorial>docker exec broker1 kafka-topics --list --bootstrap-server
oker1:9092
topic
topic2

C:\Users\not_i\OneDrive\Desktop\kafka tutorial>docker exec broker2 kafka-console-consumer --bootstrap-serv
 broker1:9092 --topic topic --from-beginning
value1
0.0
2.5
5.0
7.5
10.0
 val=6
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
```

C:\Windows\System32\cmd.exe - docker exec -it broker1 kafka-console-producer --broker-list br...

```
 - Container broker2      Started
                    10.3s
 - Container broker3      Started
                   9.4s
 - Container broker1      Started
                     8.9s

C:\Users\not_i\OneDrive\Desktop\kafka tutorial>
C:\Users\not_i\OneDrive\Desktop\kafka tutorial>docker exec broker2 bash -c "echo 'value1' | ka
fka-console-producer --request-required-acks 1 --broker-list broker2:9092 --topic topic"

C:\Users\not_i\OneDrive\Desktop\kafka tutorial>docker exec broker1 bash -c "seq 0 2.5 10 | kaf
ka-console-producer --request-required-acks 1 --broker-list broker3:9092,broker2:9092,broker1:
9092 --topic topic"

C:\Users\not_i\OneDrive\Desktop\kafka tutorial>docker exec broker1 bash -c "echo '5, val=6'| k
afka-console-producer --broker-list broker1:9092 --topic topic --property parse.key=true --pro
perty key.separator=,"

C:\Users\not_i\OneDrive\Desktop\kafka tutorial>docker exec broker1 bash -c "seq 0 2.5 20 | kaf
ka-console-producer --request-required-acks 1 --broker-list broker3:9092,broker2:9092,broker1:
9092 --topic topic"

C:\Users\not_i\OneDrive\Desktop\kafka tutorial>docker exec -it broker1 kafka-console-producer
--broker-list broker1:9092,broker2:9092,broker3:9092 --topic topic2
>yeet
>yeet2\
>yeet yeet yeet
>a=b =/3
>
```

C:\Windows\System32\cmd.exe - docker exec broker1 kafka-console-consumer --bootstrap-server broker1:9092,brok...

```
5.0
7.5
10.0
0.0
2.5
5.0
Processed a total of 10 messages

C:\Users\not_i\OneDrive\Desktop\kafka tutorial>docker exec broker1 kafka-console-consumer --bootstrap-serv
 broker1:9092,broker2:9092,broker3:9092 --topic topic2 --from-beginning
yeet
yeet2\
yeet yeet yeet
a=b =/3
```

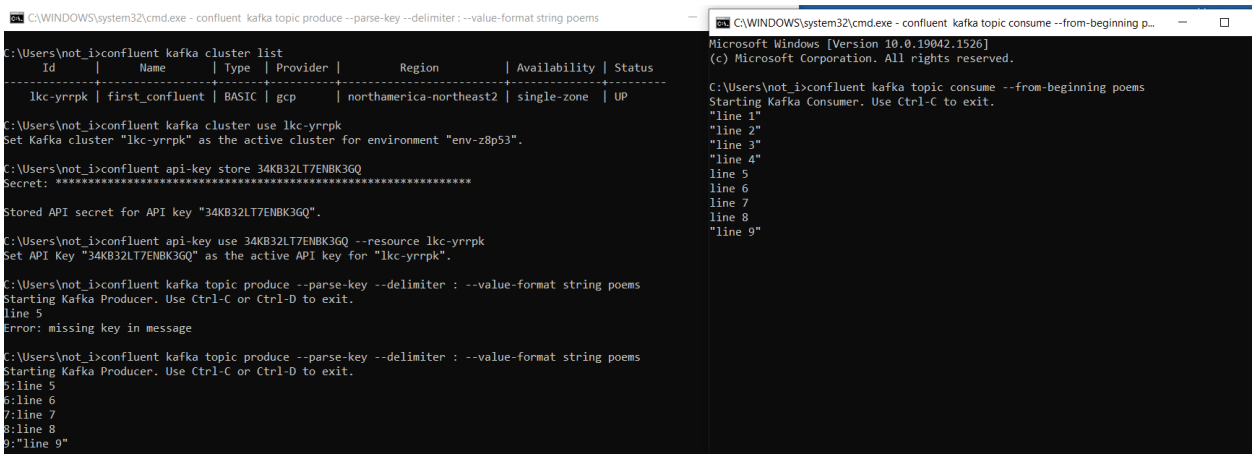
5. View video

6. See video

7. See video

8. Yml file updated

9.



Please watch video

10. done