SOFE 4630U
Cloud Computing

Project Milestone #2
2022/02/15

Data Ingestion Software and Kafka Clusters

Sabesan Sivakumar 100701928

1. What is EDA? What are its advantages and disadvantages?

EDA which stands for event-driven architecture. Event driven architecture is event driven that follows the publish and subscriber model. This architecture is loosely coupled as each publisher(event) doesn't know which subscriber(consumer) they are listening to.

Advantages:

- Loosely Coupled: each publisher(event) doesn't know which subscriber(consumer) they are listening to.
- Fault tolerance: Since its event driven each event does not rely on each other meaning if a event were to fail it will only affect that event and not the entire system

Disadvantages:

- Error handling: An application will contain multiple producers and consumers which can increase the likelihood of an error to occur as they are multiple instances.
- Duplication of Events: Since in an EDA there is no set instructions on how each event is planed of aligned their could be a possibility where a single event may create multiple duplications of the same event across ever service the application holds which in result can take up storage.

2. In Kafka, what's meant by cluster, broker, topic, replica, partition, zookeeper, controller, leader, consumer, producer, and consumer group?

Cluster: This is where all the servers (Kafka brokers) that are running are being placed. Each topic in Kafka are split into partitions and the partitions are what the broker consist of.

Broker: The broker is essentially a server that runs through the cluster. A kafka cluster consist of multiple running Kafka brokers

Topic: The topic is used as the categories to organize all the messages the application gets. Each topic has its own unique name that is known throughout the

entire Kafka cluster. Each message that has been sent will contain the unique name to which that topic will only be able to be sent to and read from that message.

Replica: As you may assume by the name, it just means to have multiple copies of the same data and place it all across multiple servers so that if a server fails we can still retrieve the data from the application.

Partition: What partitioning does is that it takes a single log and breaks it into multiple logs that can be placed on separate nodes. This improves the overall redundancy as it can mange the logs on multiple separate nodes.

Zookeeper: The Zookeeper is what keeps track of the Kafka cluster, it is responsible for maintaining the information, naming and synchronization. For example if the leader node fails Zookeeper will be able to select a new leader node.
Controller: In the cluster that consists of all the brokers there will be one broker that is the controller. The controller in charge of assigning each partitions and replicas.

Leader: The leader is chosen via the zookeeper. Kafka will write to the leader and from the leader all the followers (All the other brokers) will get partitioned.

Consumer: The consumer is what subscribes to a topic. Consumers will be able to read and process the event they have subscribed too.

Producer: The producer is what produces(Writes) to an event in Kafka.

Consumer groups: This is a group of consumers that work with each other based off their group id. They will work with each other to split up the partition and retrieve different parts of the topic.

3. Prepare a video showing the codes that generated topics, produce messages, and consume them in both NodeJS and python. Your video should display the possible producer and consumer scenarios.

https://drive.google.com/file/d/1wKi0NNpZ2_HbL4EDAq2cGgJAOy7N5QHn/view?usp=sharing

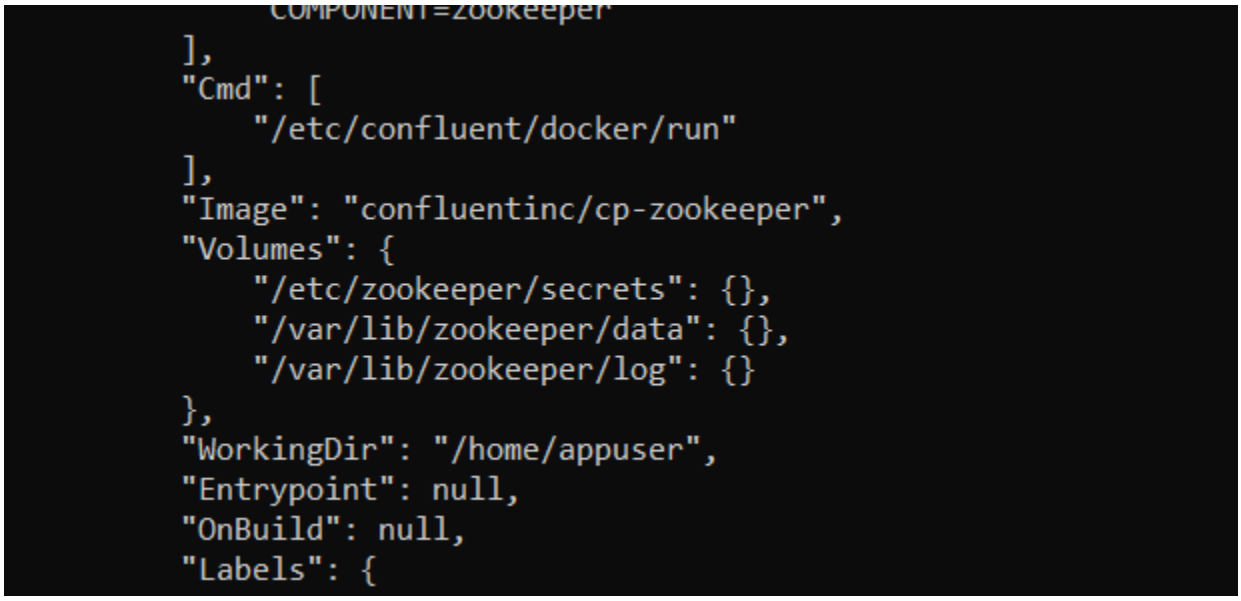
4. A problem in the used YAML file to create the docker images is that the data inside Kafka clusters are not persistent which means if the docker images are down, all its messages are lost. Update the YAML file for persistent data (hint: it's related to the volume options in Kafka brokers and zookeeper). Describe how this update solves the problem.

```yaml
services:
  zookeeper:
    image: confluentinc/cp-zookeeper
    hostname: zookeeper
    container_name: zookeeper
    networks:
      - kafka_Network
    ports:
      - 2181:2181
    volumes:
      -  /etc/zookeeper/secrets
      -  /var/lib/zookeeper/data
      -  var/lib/zookeeper/log
    environment:
      ZOOKEEPER_CLIENT_PORT: 2181
      ZOOKEEPER_TICK_TIME: 2000
  broker1:
    image: confluentinc/cp-kafka
    hostname: broker1
    container_name: broker1
    networks:
```

Following command below is what were used to obtain the volume path by getting the containerID.

```
C:\Users\sivak\Desktop\SOFE4630U-tut2-master\SOFE4630U-tut2-master\v1>docker ps
CONTAINER ID   IMAGE                          COMMAND                  CREATED          STATUS
       PORTS                                        NAMES
65e828a92c7d   confluentinc/cp-kafka          "/etc/confluent/dock…"   20 minutes ago   Up 20 mi
nutes   9092/tcp, 0.0.0.0:9095->9095/tcp             broker3
e10c0dbdb060   confluentinc/cp-kafka          "/etc/confluent/dock…"   20 minutes ago   Up 20 mi
nutes   9092/tcp, 0.0.0.0:9094->9094/tcp             broker2
826465fe6c9b   confluentinc/cp-kafka          "/etc/confluent/dock…"   20 minutes ago   Up 20 mi
nutes   9092/tcp, 0.0.0.0:9093->9093/tcp             broker1
4219a10595ce   confluentinc/cp-zookeeper      "/etc/confluent/dock…"   20 minutes ago   Up 20 mi
nutes   2888/tcp, 0.0.0.0:2181->2181/tcp, 3888/tcp   zookeeper

C:\Users\sivak\Desktop\SOFE4630U-tut2-master\SOFE4630U-tut2-master\v1>docker inspect 4219a105
95ce
```

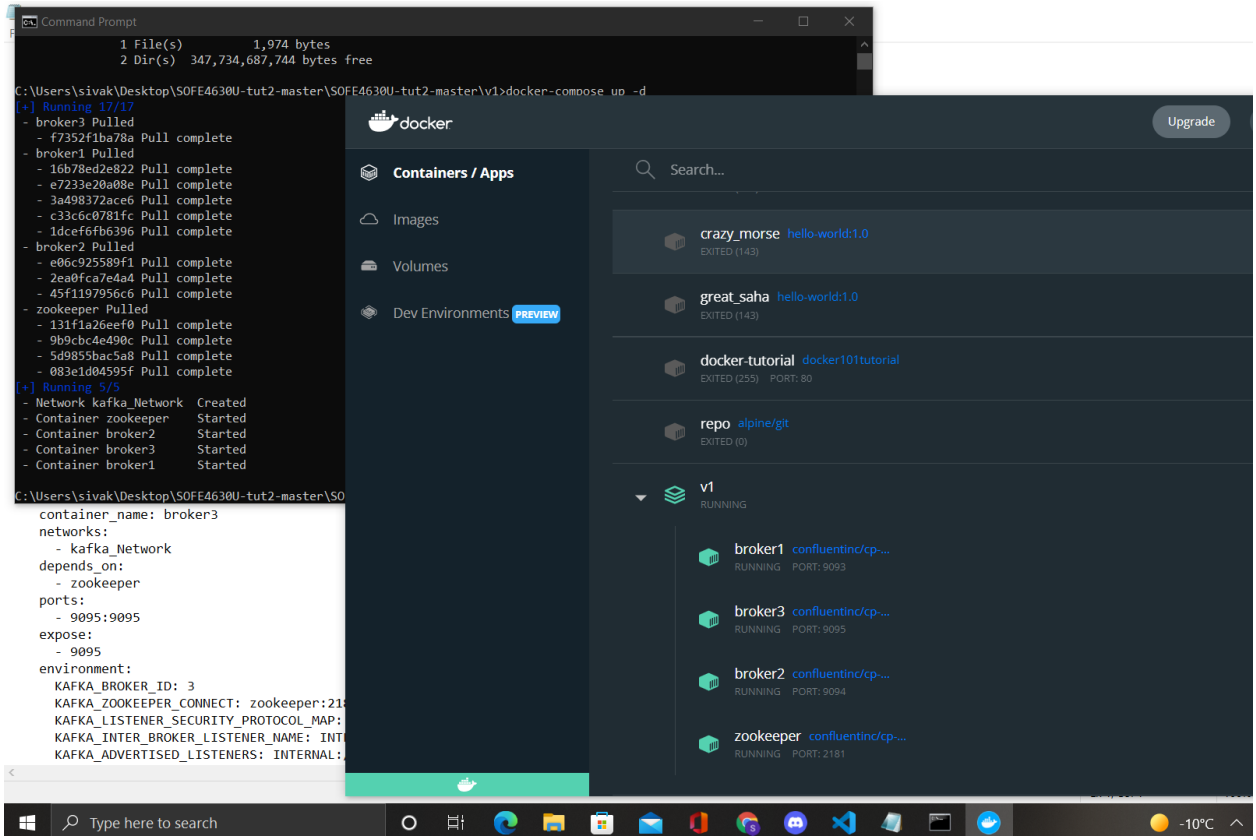


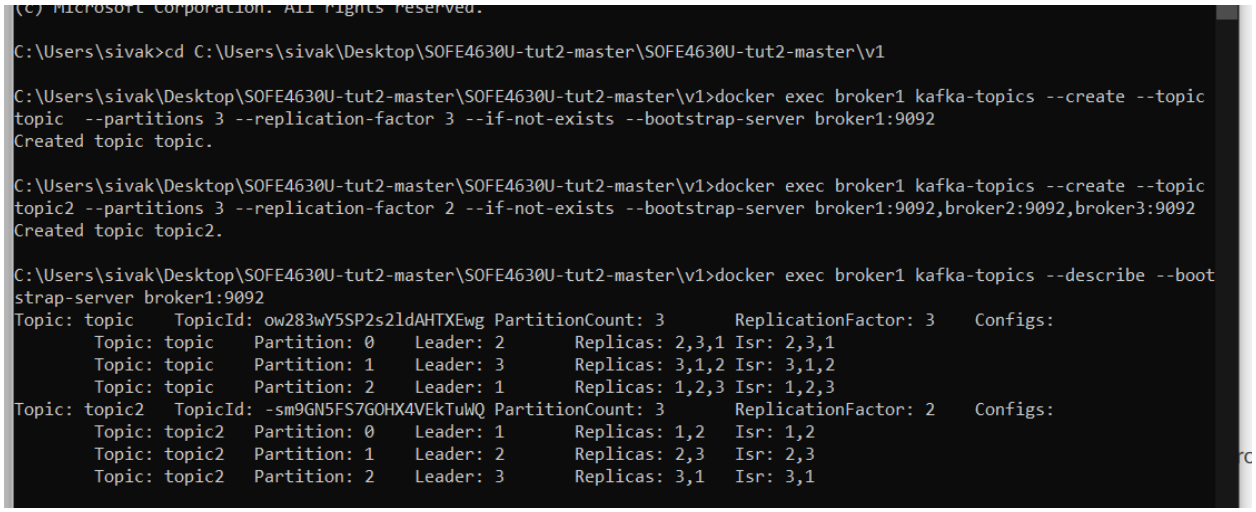
In the yml file we created a volume which consisted of the path of the zookeeper, now it can create a new volume and use the path specified to store data.

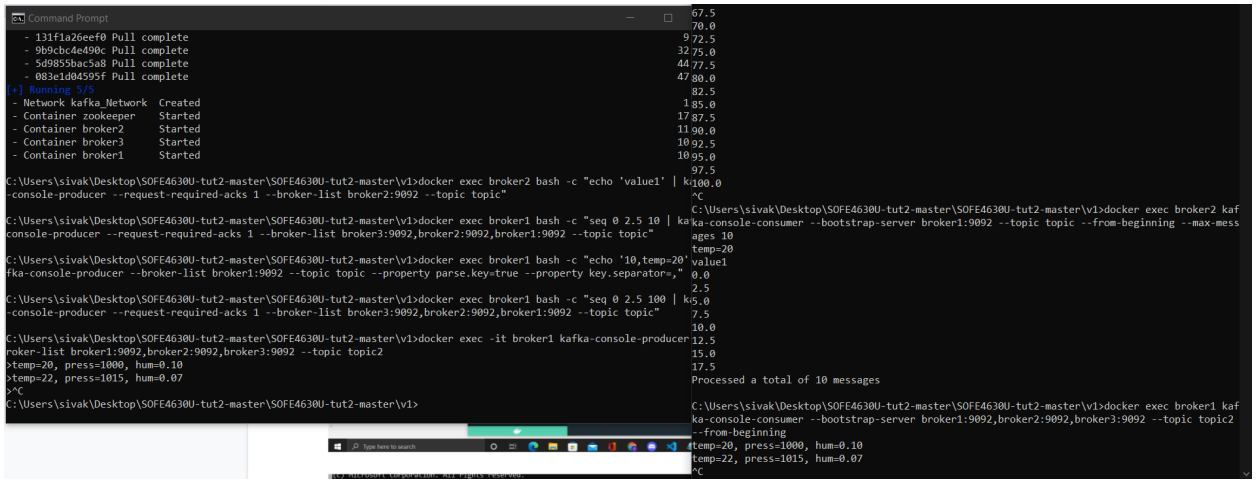
5. Follow the following video about Kafka in Confluent Cloud, and use the shown CLI to create a topic, consumer, and producer. Also update your python code to create a consumer, and producer using Kafka in Confluent Cloud (hint: only the connection information of Kafka Cluster has to be updated). Record a video illustrating those tools and showing them in action

https://drive.google.com/file/d/1wKi0NNpZ2_HbL4EDAq2cGgJAOy7N5QHn/view?usp=sharing

Screenshots:








```
C:\Users\sivak\Desktop\SOFE4630U-tut2-master\SOFE4630U-tut2-master\v1>docker ps
CONTAINER ID   IMAGE                          COMMAND                  CREATED          STATUS
        PORTS                                        NAMES
65e828a92c7d   confluentinc/cp-kafka          "/etc/confluent/dock…"   20 minutes ago   Up 20 mi
nutes   9092/tcp, 0.0.0.0:9095->9095/tcp             broker3
e10c0dbdb060   confluentinc/cp-kafka          "/etc/confluent/dock…"   20 minutes ago   Up 20 mi
nutes   9092/tcp, 0.0.0.0:9094->9094/tcp             broker2
826465fe6c9b   confluentinc/cp-kafka          "/etc/confluent/dock…"   20 minutes ago   Up 20 mi
nutes   9092/tcp, 0.0.0.0:9093->9093/tcp             broker1
4219a10595ce   confluentinc/cp-zookeeper      "/etc/confluent/dock…"   20 minutes ago   Up 20 mi
nutes   2888/tcp, 0.0.0.0:2181->2181/tcp, 3888/tcp   zookeeper

C:\Users\sivak\Desktop\SOFE4630U-tut2-master\SOFE4630U-tut2-master\v1>docker inspect 4219a105
95ce
```