Answer the following questions:

What is EDA? What are its advantages and disadvantages?

EDA stands for Event Driven Architecture. It composes of loosely coupled systems that communicates by publishing and subscribing events or by streaming events.

Advantages of EDA is that systems are loosely coupled which means they are more scalable since making changes to a system shouldn’t affect the other systems. Another advantage is that there is no single point of failure having multiple systems.

Disadvantages of EDA is that systems that follow EDA require a lot of error handling

* Reliability
* Consistency

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

Cluster – Contains usually 3 brokers with multiple topics in each broker. This is the central hub for producers and consumers to subscribe and publish messages.

Broker – Server within a cluster. Responsible for receiving messages, assigning offsets and committing messages to disk. Responsible for responding to consumers fetch request and sends them messages

Topic – Provide a way of categorizing data which can be broken down to a number of partitions. This is a way to organize data published by the producers.

Replica – Kafka replicas are copies of the data that are spread across multiple clusters and brokers.

Partition – Allows topics to work in parallel by splitting data into particular topic across multiple brokers

Zookeeper – keeps track of the brokers and topics in the cluster. This component is like the brain of the system as it oversees all. It can do some failure handling when a certain node breaks down.

Controller – is a broker in a cluster is assigned the controller which is responsible for managing other brokers in terms of the states of the data to achieve consistency or reassigning partitions

Leader – is a broker in a cluster that manages the read and write request of data while the other brokers simply follows/replicates the data of the leader.

Consumer – Applications consuming data from Kafka. Subscribe to a topic and retrieves the messages the order it was published.

Producer – Applications that produce data and sends it to the Kafka cluster. Creates new messages and sends to a specific topic

Consumer group – a group of consumers that consume data from a specific topic. Partitions in a topic is divided among the consumers so each consumer will consume from separate partitions.

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.

I created volumes to where the data will be persisted locally and I mounted the data directories of the zookeeper and the brokers to those volumes. Whenever the docker-compose is reset, it will retrieve the data from that folder and load it in to the brokers and zookeeper