



## Faculty of Engineering & Applied Science

*Course #Title: Cloud Computing*

*Date: February 15rd, 2022*

*Project Milestone #2*

*Title: Data Ingestion Software-- Kafka Clusters*

*Group #2*

*Student names: Richard Said - 100659431, Kaamran Minhas - 100593277,  
Taha Hashmat - 100689792, Adam Wong Chew Onn - 100598499, Sarthak Sharma  
- 100604428*

### **Google Link for Final Videos:**

*<https://drive.google.com/drive/folders/1rLFE-SvmO0yf4jeGBDC3d4DBmjygxist?usp=sharing>*

### **GitHub Link for Individual and Final Submissions:**

*<https://github.com/tahahashmat/Group2-ProjectMilestone2>*

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

- EDA (Event driven architecture) is a software architecture used by many big software companies whose main goal when developing software applications is to produce, consume, react, and detect events. Even though big data involves large chunks of data being processed simultaneously, the more data you have the longer it will take to process it. EDA works around this drawback, and focuses on data being used to change its state (i.e. an event)
- **Advantages:** EDA tools have a number of advantages including:
  - Reducing time spent creating complicated ICs
  - Loosely coupled, which allows programmers to use multiple different technologies for different components (producers and consumers) while constructing an event driven architecture
  - Avoiding manufacturing mistakes and lowering manufacturing costs
  - EDA can be considered a distributed model. For this reason an Event Driven Architecture is considered to be highly scalable in accordance to the number of events being processed
  - Real time data processing which ensures that data is updated every millisecond
- **Disadvantages:**
  - The API provider is more complex, developer experience as well as governance and standardization becomes limited, and the analytics of APIs become more ambiguous.
  - Since each application that utilizes Event Driven Architecture consists of hundreds of producers and consumers, keeping track of each component and making sure it is functional and working error free is quite difficult, thus third party tools are required for error detection and handling.

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

- **Cluster:** One or more servers – or in this case referred to as ‘Kafka Brokers’ - running make up a Kafka cluster.
- **Broker:** A broker in kafka is basically synonymous with a kafka server and resides within a cluster.
- **Topic:** A topic is a collection of partitions that are handled as a whole.
- **Replica:** Kafka involves creating multiple copies of data for security purposes which is why they are distributed across multiple brokers. These copies are called replicas
- **Partition:** In kafka there are collections of data called “logs” and partitioning is essentially breaking those logs into multiple parts called partitions.
- **Zookeeper:** is a service synchronization and naming registry used in distributed systems.

- **Controller:** In a kafka cluster one broker is chosen to be the controller where replicas and partitions are managed by this controller as well as the re-assigning of such partitions.
- **Leader:** The followers passively duplicate the leader, while the leader handles all read and write requests for the partition. Load is evenly distributed throughout the cluster, each server functions as a leader for some of its partitions and a follower for others
- **Consumer:** Kafka is generally consumed by groups of people. When a large number of consumers subscribe to the same topic and belong to the same consumer group, each consumer receives messages from a subset of the subject's partitions.
- **Producer:** Each message is assigned to a topic partition by a producer partitioner, who then sends a produce request to the partition's leader.
- **Consumer Group:** Consumer groups are basically a culmination of multiple consumers subscribed to a particular topic containing partitions.