**Apache Kafka:**

Apache Kafka is a distributed event streaming platform. With an ability to allow applications to manage large amounts of data, Kafka is also fault-tolerant and built to scale. Apache Kafka’s framework is based on Java and the Publish-Subscribe Messaging system. The framework allows data streaming at an unprecedented rate, that too, from multiple sources Kafka is famous in the data community for data streaming services because it can handle Big Data with large input volumes. And, with minimum downtime and low latency, Kafka services are easy to scale up and down.

**Some of Kafka’s most valuable features are as follows:**

**High Scalability:** The partitioned log model allows Kafka services to scale beyond a single server’s capability.

**Low Latency:** Kafka services separate data streams, allowing low latency and high throughput.

Fault-Tolerant & Durable: In Kafka, partitions are segregated then duplicated across servers. The segregation and duplication process makes Kafka services fault-tolerant by protecting them against ad-hoc server failures like master and database failures.

**High Extensibility:** Kafka is highly accessible through various other applications, allowing developers to add more features.

**problem statement :-** to create a data pipeline, suppose we have to a pull a data from source and through Kafka topic we need to collect that data into target. kafka has a pub sub architecture, producer is responsible to publish a data into Kafka broker inside specific topic (specific area to store some related logical data) into specific partitions and consumer is responsible for to subscribe (read data from topic) data from those partitions. so here use case is like here in terms of source we are pulling data from excel sheet and dumping data into Kafka broker-topic and from there we are reading data and storing in output excels dividing data into different excels depends on different groups.

With 2nd consumer code we are trying do more exercise for practice , like we segregate data more into group wise pass and failed students into different different files

