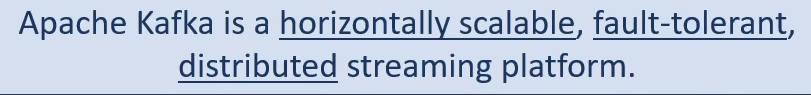
1. **Agenda**:
   1. In this section, we’re going to understand Kafka **internals and some architectural concepts**.  
      **NOTE**: These fundamental concepts are critical for the rest of the course.
2. Let’s get started.



1. You must be categorically (**सुनिश्‍चित रूप से, सुस्‍पष्‍ट रूप से**) sure about the answer.
2. Apache Kafka is **Messaging Broker**.
3. Everything else is an API, Library, Framework to interact with Kafka Broker or to work with Data in realtime.
4. Kafka Broker is a middleman b/w the Producers and Consumers playing three Primary Responsibilities.
   1. Graphical user interface, text, application

      Description automatically generated
   2. Text

      Description automatically generated with medium confidence  
      Storage is also critical to ensure that a consumer can consume it later.   
      So that a consumer need not necessarily read it in realtime.
   3. A picture containing text

      Description automatically generated
5. Till now, this is the simplest answer to “What is Apache Kafka”.  
   A more elaborated answer can be something like  
   Jatin: To understand the difference b/w Horizontal Scaling and Distributed System, visit  
   [https://stackoverflow.com/questions/61256131/what-is-the-difference-between-horizontal-scaling-and-distributed-systems#:~:text=\*Distributed%20systems%20have%20different%20components,and%20run%20on%20different%20machines](https://stackoverflow.com/questions/61256131/what-is-the-difference-between-horizontal-scaling-and-distributed-systems#:~:text=*Distributed%20systems%20have%20different%20components,and%20run%20on%20different%20machines).
6. 

It is consciously designed to build realtime **streaming data architecture**.  
This new definition has a lot to understand.  
Let’s break it into 3 parts.  
Text

Description automatically generated

* 1. In the first part, we will talk about **Kafka Message Storage Architecture**.  
     This discussion will help you to understand some core concepts such as **Topics, Logs, Partitions, replication Factors, segments, offset, offset-index**.
  2. Then we will move to **Kafka Cluster Architecture** which will help you to understand some concepts associated with **cluster formation, Zookeeper & Controller**.
  3. Finally, we will tie up these two architectures (Storage and Cluster) and try to understand how the work is **distributed** in the Kafka Cluster.  
     In this part, we will learn about **Leaders, Followers, In Sync Replica, Committed and Uncommitted Messages**.

1. A picture containing graphical user interface

   Description automatically generated
2. In the next lecture, we will learn about **Kafka Storage Architecture**.