1. Text

   Description automatically generated  
   
2. In earlier lectures, we explored two dimensions of Apache Kafka.
   1. Partitions - Log Files.
   2. Cluster Formation.
3. The next step is to tie up the relationship b/w these two dimensions and understand how the work is distributed among the Brokers in the Kafka Cluster.  
   Primarily, we want to understand what makes Kafka to be scalable and fault tolerance system.  
   
4. Diagram

   Description automatically generated If we look at the Kafka Topic organization, it is broken down into **independent partitions**.
5. Each Partition is **self-contained**.  
   It means all the information about the partition such as segments files(log files), index file (maintaining offsets), timeindex (maintaining timestamps) are contained in the same directory at a place which we call **Partition Directory** (**Data Directory**).
6. This structure is excellent because it allows us to distribute the work among the Kafka Brokers in a Cluster efficiently as partitions can be distributed over the brokers running on different machines inside a cluster. All we need to do is to spread the responsibility of partitions in the Kafka Cluster.  
   The point is straightforward. When you create a topic, the responsibility of creating, storing and managing the partitions is distributed among the available brokers in the cluster. That means every Broker is responsible for managing one or more partitions that are assigned to that Broker and that is how work is shared by the brokers.
7. Having said that, let’s try to understand with an example.
   1. **NOTE**: We will use the term Assignment of partition to a broker which means Kafka is asking a broker to **create** that partition.
   2. A screenshot of a computer screen

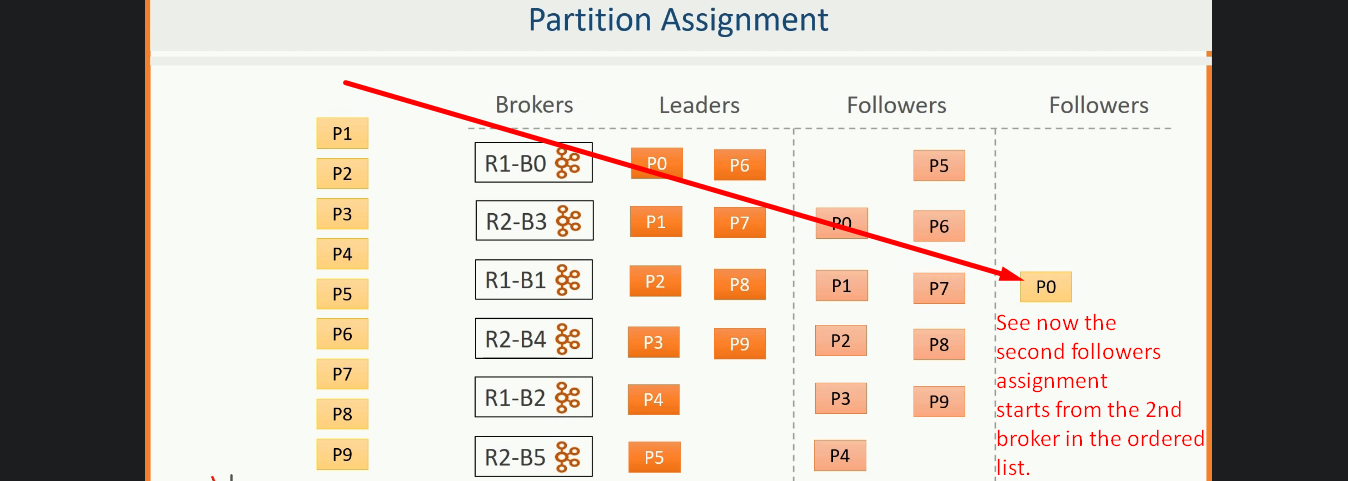
      Description automatically generated
   3. To distribute the 30 replicas (10 Leaders + 20 Followers), Kafka applies the following steps.
   4. 
      1. Makes an ordered list of available brokers.
      2. Assign Leaders and followers to the list in order.
   5. Let’s apply the first rule 🡪 **Ordered List of Brokers.**Diagram

      Description automatically generated
   6. **Let’s apply the 2nd rule 🡪 Leaders & Followers assignment.**Table

      Description automatically generated
   7. Now we have 30 partitions (10 Leaders & 20 Followers) which are to be assigned to that ordered list of 6 Brokers.
   8. Ideally, Kafka should place five partitions (30 replicas/6brokers = 5Partitions/Broker) on each broker to achieve the **even distribution** of the partitions.  
      Total Partitions = 30  
      Total Brokers = 6, where each broker will be assigned 5 partitions.
   9. However, we have to achieve other goal which is **fault tolerance.**
      1. It means if one broker fails, other broker has the same copy.
      2. Further, making sure if an entire rack fails, we will have a copy on a different rack 😊.  
         For example, we have 3 copies of partition p0  which should not be assigned to the same Broker and all of them to the same rack.
   10. Let’s see how kafka does that (Partitions assignment to brokers achieving two above goals).
   11. As soon as we have ordered list of Brokers, assigning partitions is as simple as assigning one partition to each broker in **round robin fashion**.
   12. Kafka starts with leader partitions and finishes assignment of leaders first.
   13. Table

       Description automatically generated
   14. A picture containing PowerPoint

       Description automatically generated
   15. Let’s now assign 20 followers.
   16. Followers assignment starts from the second broker in the ordered list following the **round-robin fashion**.
   17. A picture containing calendar

       Description automatically generated  
       Above we assigned P0 replica in such a way that we achieved fault tolerance like if entire Rack R1 along with Leader P0 goes down, P0 replica is still on other broker on other rack R2-B3.  
         
       A picture containing calendar

       Description automatically generated  
         
       PowerPoint

       Description automatically generated  
         
       PowerPoint

       Description automatically generated with low confidence
   18. You can compare other Partitions as well.  
       They are well arranged in such a way that at least two copies are placed on two different racks. (We have 2 racks and 3 replicas so one rack will get two replicas)
   19. This arrangement is ideal for fault tolerance.
8. Great!!! We learnt how the replicas are distributed among the brokers in a cluster making the system fault tolerance.
9. However, distributing the replicas among the Brokers is the 1st half of the work distribution.  
   The 2nd half of the problem is to define the responsibility of an individual broker.  
   We will discuss it in next lecture.