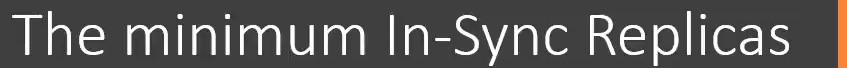
1. Graphical user interface, text

   Description automatically generated  
   
2. In the earlier lecture, we talked about committed and uncommitted msgs and their relationship with the ISR list.
3. **Agenda**:
   1. We will extend the idea further and learn about the minimum In-Sync-Replica (ISR)
4. 
5. Msgs are considered to be committed when it is written to all In-Sync replicas.
6. Now, let’s assume that we start with 3 replicas (B1, B2, B4) and all of them are healthy enough to be in the ISR.  
     
   Diagram

   Description automatically generated
7. However, after some time, two of them get failed (B2, B4).  
   Diagram

   Description automatically generated  
   In this case, even though we have configured 3 replicas for the topic but we are left with the single in-sync replica. That is the leader itself.  
   Now all the msgs will be considered to be committed when it is written to all the replicas even though we have just one replica in the ISR.   
     
   Diagram

   Description automatically generated  
   But that is risky scenario for Data consistency because Data could be lost if we lose the leader.  
   Kafka protects this scenario by setting the minimum number of In-Sync replicas to a topic.  
   If you want to be sure that the committed msgs to be written at least to replicas, then you need to set the minimum number of In-Sync replicas to two.  
     
   Diagram

   Description automatically generated  
   There is a side-effect of this setting (min-sync-replicas=2).  
   If a topic has three replicas and you set min-sync-replicas=2, then you can write to a partition only if two out of three replicas for that partition is in sync.  
   If at least two replicas are not in sync, the leader will not accept the msg and respond with **Not Enough Replica Exception**.   
   In order words, the Leader will become read-only partition and so we can read but can’t writ to until you bring up another replica and wait for it to catch up and get in-sync.  
   Diagram

   Description automatically generated
8. We have covered a lot in this and earlier lectures.   
   This might appear overwhelming, and it is.  
   But nothing to worry.   
   We will repeat this again and again in this course when we will start creating apps and examples.