**Contents:**

* **Internals of Topics, Partitions and Replication**
* **What happens internally when we create a topic?**
* Partition State
* Replication State
* **Kafka Controller node.**
* **Internals of Topics, Partitions and Replication:**
* At first, we will create a topic with partitions=3 and replication-factor 2.
* Go inside /kafka\_2.13-3.4.0/

bin/kafka-topics.sh --bootstrap-server localhost:9092, localhost:9093 --create --topic E2EKafka --partitions=3 --replication-factor=2

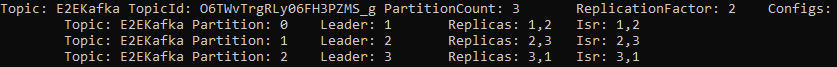
* We will get below output once created the topic:

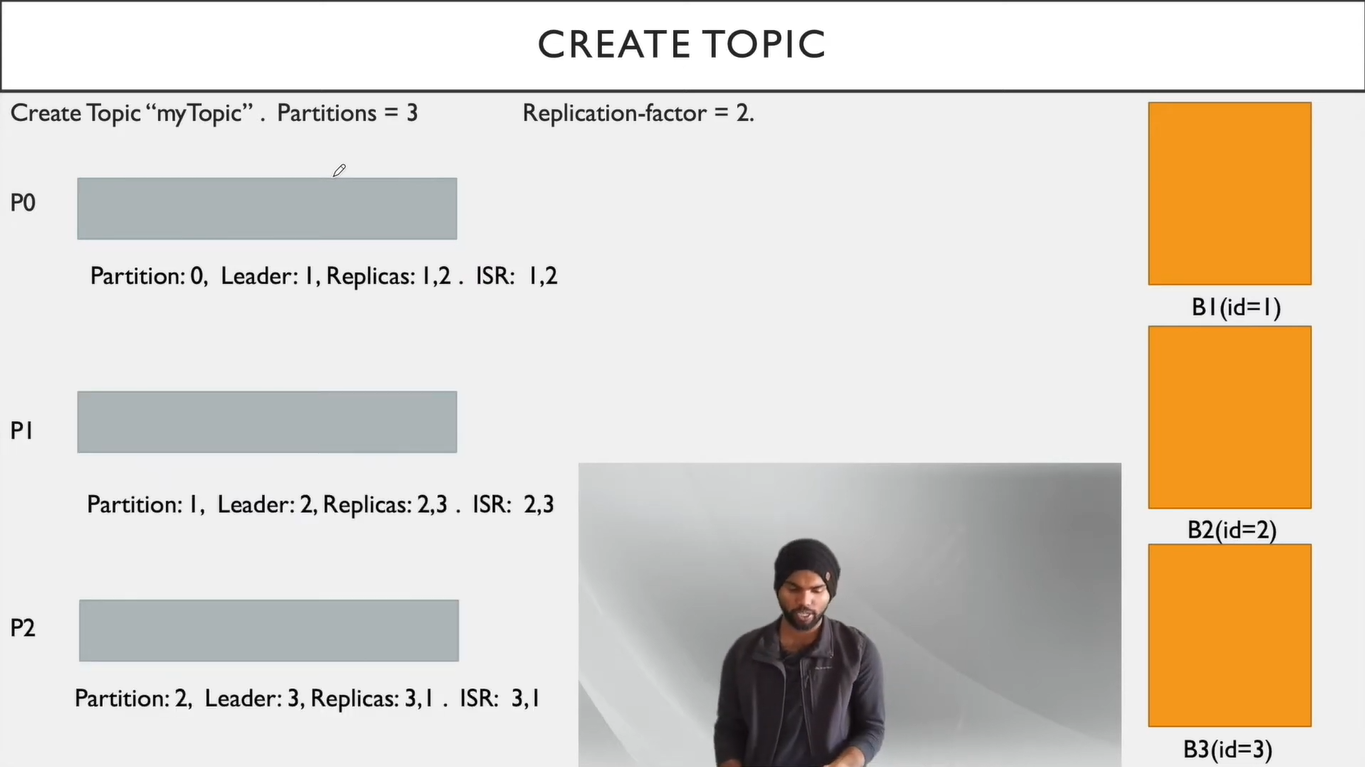


* Now we will check the description of topic, where we will get information about leader, partitions and replications.

./kafka-topics.sh --bootstrap-server localhost:9092, localhost:9093 --describe --topic E2EKafka

* Below is the output:





P0

P2’

P2

P1’

P1

P0’

Figure 1\_Ref. partitions and replicas are diff. than our example

1. In above image we can see there are 3 partitions and 2 replication factors.
2. Here we will go for P0 partition.
3. In P0 there is one leader (B1) and replicas are B1 and B2, ISR are B1 and B2.
4. As Leader is B1 all the read and write operations will be performed by B1.
5. Read operation is from consumers and write operation is from producer.
6. Now we will see replicas. In P0 there are 2 replicas B1 and B2. All the operations are performed by B1 as it is leader.
7. Replica from B1 will be main replica and replica from B2 will be follower replica.
8. If B1 will die or goes down B2 will act as main replica for read and write operations.
9. Now we will see ISR which is In Sync Replicas. When a producer produces a message, it will produce it on B1 that time B2 will copy the messages from B1 as it is a replica in P0.
10. When both B1 and B2 are in sync means all the brokers have same messages we can see both broker ids in ISR field when we describe a topic.
11. When all the replicas are not in sync, we can see only those which are in sync and not those which are not in sync.

* **What happens internally when we create a topic?**
* **Partition States:**

1. **Non-Existent Partition:**

This state indicates that the partition was either never created or deleted after creation.

1. **New Partition:**

After creation the partition is in new partition state. In this sate the partition should have replicas assigned but no leader/ISR yet.

1. **Online Partition:**

Once a leader is selected for a partition it will be in online partition state.

1. **Offline Partition:**

If Leader broker went down or dies that time the partition moves to offline partition state. And again, leader elections will take place.

|  |
| --- |
| **Note: In Kafka cluster we have only one controller node, and this manages states of all the partitions.** |

* **Replica States:**

1. **New Replica:**

When replicas are created during topic creation or partition reassignment the state of replica should be New Replica.

1. **Online Replica:**

Once a replica is started and a part of assigned replicas for its partition, it is I this state. I this state it can either become leader or follower state change request.

1. **Offline Replica:**

If a replica dies it moves to this state. This happens when the broker hosting the replica is down.

1. **Non-Existent Replica:**

If a replica is deleted it moves to this state.

* **Controller Node:**

1. In a Kafka cluster, one of the brokers serves as the controller.
2. The controller node is responsible for managing the states of partitions and replicas.
3. Controller node is also responsible for performing administrative tasks like reassigning the partitions.