**Contents:**

* **Alter topic partitions.**
* **Partition re-assignment.**
* **Alter Topic Partitions:**

1. We can alter topic partitions by using alter attribute.
2. We can only increase the partitions and cannot decrease, because there will be chance of data loss if we will decrease the partitions.
3. Core principal of Kafka is there should not be any data loss.
4. We can alter partitions using below command:

**./kafka-topics.sh --bootstrap-server localhost:9092 --alter --topic Topic-1 --partitions=2**

**Before Partition:**



**After Partition:**



* **Partition Re-assignment:**

**Use Case:**

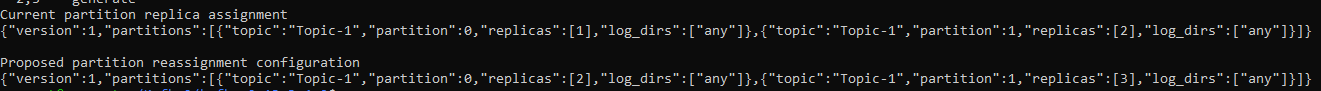
1. **Move partitions across brokers.**
2. **Selectively move replicas of a partition to a specific set of brokers.**
3. **Increase the replication factor.**
4. **Move partition across brokers:**

**There are three stages while moving partitions:**

1. **Generate:**

**bin/kafka-reassign-partitions.sh --bootstrap-server localhost:9092 --topics-to-move-json-file TopicsToMove.json --broker-list "2,3" –generate**

**Output:**



When we will run the generate script, we will get two json files, i.e.

1. Current partition replica assignment

2. Proposed partition reassignment configuration

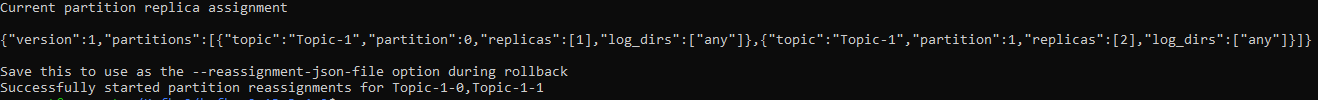
In Current partition replica assignment, we have a json file with current replica configurations and in Proposed partition reassignment configuration, we will get the partition reassignment json with latest brokers.

Now save the proposed json file in our system and use it for execute stage.

1. **Execute:**

**bin/kafka-reassign-partitions.sh --bootstrap-server localhost:9092 --reassignment-json-file SuggestedChange.json --execute**

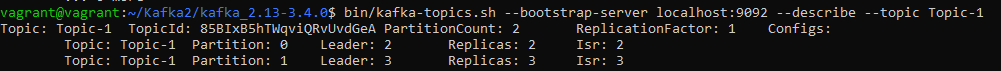
**We will get below output once run above script:**



**We will get one json which is previous configuration of partitions, we can save it for rollback as given in output.**

**Here we can check if partitions are changed or not using below command:**

**bin/kafka-topics.sh --bootstrap-server localhost:9092 --describe --topic Topic-1**



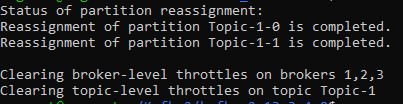
As we can see partition changed from B1,B2 to B2,B3

1. **Verify:**

Now we can verify that our partition reassignment is successful or not using below command:

**bin/kafka-reassign-partitions.sh --bootstrap-server localhost:9092 --reassignment-json-file SuggestedChange.json --verify**

We will get below output using above command:



From above output we can get to know our partition reassignment is successful.

* **Change Replication-factor:**

We can change replication factor from 1 to 2 using json file which we received while describing the partition:

**Before:**

|  |
| --- |
| **{“version”:1,"partitions":[{"topic":"Topic-1","partition":0,"replicas":[2]},{"topic":"Topic-1","partition":1,"replicas":[3]}]}** |

**After:**

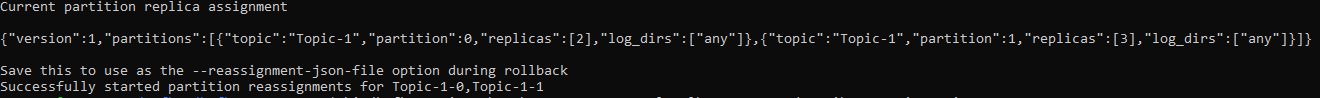
|  |
| --- |
| **{“version”:1,"partitions":[{"topic":"Topic-1","partition":0,"replicas":[1,2]},{"topic":"Topic-1","partition":1,"replicas":[3,2]}]}** |

Above is the json file which we have updated to increase replication factor.

We used below script:

**bin/kafka-reassign-partitions.sh --bootstrap-server localhost:9092 --reassignment-json-file SuggestedChange.json –execute**

**Output:**



We can verify the replication factor change using below command:

**bin/kafka-topics.sh --bootstrap-server localhost:9092 --describe --topic Topic-1**

**Output:**

**Before:**



**After:**



In above screenshots we can see the replication factor changed from 2 and 3 to 1,2 and 3,2.