Changing Replication Factor of a Topic in Apache Kafka

Published on: 2018-05-25 16:05:40

Posted By: Amit Kumar

This tutorial will provide you with steps to increase replication factor of a topic in Apache Kafka

Abstract

Replication factor is quite a useful concept to achieve reliability in Apache Kafka. It conveys information about number of copies to be maintained of messages for a topic.

E.g. if replication factor is set to two for a topic, every message sent to this topic will be stored on two brokers. However, at a time, only one broker (leader) serves client requests for a topic and remaining ones remain passive only to be used in case of leader broker is not available.

Apache Kafka ensures that you can't set replication factor to a number higher than available brokers in a cluster as it doesn't make sense to maintain multiple copies of a message on same broker. E.g. if you have two brokers running in a Kafka cluster, maximum value of replication factor can't be set to more than two.

Replication factor is set at the time of creation of a topic as shown in below command from Kafka home directory (assumming zookeeper is running on local machine with 2181 port) -

```
# Creates a topic with name 'demo-topic' with 2 partitions and 1 replication factor
./bin/kafka-topics.sh --create --zookeeper localhost:2181 --topic demo-topic --partitions 2 --replication-factor 1
```

You can verify replicatin factor by using ***--describe*** option of ***kafka-topics.sh*** as follows -

```
> ./bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic demo-topic
Topic:demo-topic        PartitionCount:2        ReplicationFactor:1     Configs:
        Topic: demo-topic       Partition: 0    Leader: 0       Replicas: 0     Isr: 0
        Topic: demo-topic       Partition: 1    Leader: 1       Replicas: 1     Isr: 1
```

However, you may want to increase replication factor of a topic later for either increased reliability or as part of deferred infrastructure rampification strategy.

Changing Replication Factor

We will now be increasing replication factor of our ***demo-topic*** to three as part of our deferred infrastructure rampification strategy.

First step is to create a JSON file named ***increase-replication-factor.json*** with reassignment plan to create two relicas (on brokers with id 0 and 1) for all messages of topic ***demo-topic*** as follows -

```
{
 "version":1,
 "partitions":[
      {"topic":"demo-topic","partition":0,"replicas":[0,1]},
      {"topic":"demo-topic","partition":1,"replicas":[1,0]}
 ]
}
```

Next step is to pass this JSON file to Kafka reassign partitions tool script with ***--execute*** option -

```
> ./bin/kafka-reassign-partitions.sh --zookeeper localhost:2181 --reassignment-json-file increase-replication-factor.json --execute
Current partition replica assignment
{
 "version":1,
 "partitions":[
      {"topic":"demo-topic","partition":0,"replicas":[0]},
      {"topic":"demo-topic","partition":1,"replicas":[1]}
 ]
}

Save this to use as the --reassignment-json-file option during rollback
Successfully started reassignment of partitions
{
 "version":1,
 "partitions":[
      {"topic":"demo-topic","partition":0,"replicas":[0,1]},
      {"topic":"demo-topic","partition":1,"replicas":[1,0]}
 ]
}
```

Finally, you can verify if replication factor has been changed for topic ***demo-topic*** using ***--describe*** option of ***kafka-topics.sh*** tool -

```
> ./bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic demo-topic
Topic:demo-topic        PartitionCount:2        ReplicationFactor:2     Configs:
        Topic: demo-topic       Partition: 0    Leader: 0       Replicas: 0,1     Isr: 0,1
        Topic: demo-topic       Partition: 1    Leader: 1       Replicas: 1,0     Isr: 1,0
```

We can also decrease replication factor of a topic by following same steps as above.

Thank you for reading through the tutorial. In case of any feedback/questions/concerns, you can communicate same to us through your comments and we shall get back to you as soon as possible.

Posted By: Amit Kumar
Published on: 2018-05-25 16:05:40