**MirrorMaker** is a process in Apache Kafka to replicate or mirror data between Kafka Clusters.

One use case is to provide a replica of a complete Kafka cluster in another data center to cater to different use cases without impacting the original cluster.

In MirrorMaker, there is a consumer connector and producer connector. The consumer will read data from topics in source Kafka cluster and the producer connector will write those events or data to target Kafka Cluster. Source cluster and target cluster are independent of each other.

Let's understand this with a simple setup where both clusters exist on the same machine. We are using two Kafka Clusters; each with two Kafka nodes and one zookeeper node. All processes run on the same host. One Kafka Cluster is the source and the other is the target.

1. Create folders for zookeeper and Kafka logs.

cd /usr/local/kafka or in your kafka/chosen location

mkdir -p data/zookeeper1

mkdir -p data/zookeeper2

mkdir -p data/kafka-logs-1-1

mkdir -p data/kafka-logs-1-2

mkdir -p data/kafka-logs-2-1

mkdir -p data/kafka-logs-2-2

1. Configuration for zookeeper nodes.

vi config/zookeeper1.properties

dataDir=~/kafka\_2.13-2.4.1/data/zookeeper1

clientPort=2181

maxClientCnxns=0

---

vi config/zookeeper2.properties

dataDir=~/kafka\_2.13-2.4.1/data/zookeeper2

clientPort=2182

maxClientCnxns=0

1. Configuration for Kafka nodes. Total 4 Kafka nodes, 2 node connect to 2181 and other 2 to 2182.

cp config/server.properties config/server1-1.properties

cp config/server.properties config/server1-2.properties

cp config/server.properties config/server2-1.properties

cp config/server.properties config/server2-2.properties

vi ~/kafka\_2.13-2.4.1/config/server1-1.properties

broker.id=0

port=9093

zookeeper.connect=localhost:2181

advertised.host.name = localhost

log.dirs=~/kafka\_2.13-2.4.1/data/kafka-logs-1-1

-----

vi ~/kafka\_2.13-2.4.1/config/server1-2.properties

broker.id=1

port=9094

zookeeper.connect=localhost:2181

advertised.host.name = localhost

log.dirs=~/kafka\_2.13-2.4.1/data/kafka-logs-1-2

-----

vi ~/kafka\_2.13-2.4.1/config/server2-1.properties

broker.id=2

port=9095

zookeeper.connect=localhost:2182

advertised.host.name = localhost

log.dirs=~/kafka\_2.13-2.4.1/data/kafka-logs-2-1

-----

vi ~/kafka\_2.13-2.4.1/config/server2-2.properties

broker.id=4

port=9096

zookeeper.connect=localhost:2182

advertised.host.name = localhost

log.dirs=~/kafka\_2.13-2.4.1/data/kafka-logs-2-2

1. Start zookeeper nodes and Kafka nodes.

zookeeper-server-start.sh ../config/zookeeper1.properties

zookeeper-server-start.sh ../config/zookeeper2.properties

kafka-server-start.sh ../config/server1-1.properties

kafka-server-start.sh ../config/server1-2.properties

kafka-server-start.sh ../config/server2-1.properties

kafka-server-start.sh ../config/server2-2.properties

1. Create topic mirrormakerPOC on both Kafka clusters with same number of partition.

kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 2 --partitions 2 --topic mirrormakerPOC

kafka-topics.sh --create --zookeeper localhost:2182 --replication-factor 2 --partitions 2 --topic mirrormakerPOC

1. Create consumer and producer configuration file for mirror maker.

>cat sourceCluster1Consumer.config

bootstrap.servers=localhost:9093,localhost:9094

exclude.internal.topics=true

client.id=mirror\_maker\_consumer

group.id=mirror\_maker\_consumer

>cat targetClusterProducer.config

bootstrap.servers=localhost:9095,localhost:9096

acks=1

batch.size=50

client.id=mirror\_maker\_test\_producer

1. Now run MirrorMaker process based on consumer and producer configuration defined in last step.

kafka-mirror-maker.sh --consumer.config ../../sourceCluster1Consumer.config --num.streams 1 --producer.config ../../targetClusterProducer.config --whitelist= ’.\*’

kafka-mirror-maker.sh --consumer.config ../../sourceCluster1Consumer.config --num.streams 1 --producer.config ../../targetClusterProducer.config –whitelist Test1 –blacklist Test2

1. Start sending message to Kafka Cluster 1 listening on zookeeper port 2181.

kafka-console-producer.sh --broker-list localhost:9093,localhost:9094 --topic mirrormakerPOC

1. Start consuming on Kafka nodes of both Kafka Clusters.

kafka-console-consumer.sh --bootstrap-server localhost:9095,localhost:9096 --topic mirrormakerPOC --group topic\_group\_2

**Consume for Kafka nodes on 1st Cluster.**

kafka-console-consumer.sh --bootstrap-server localhost:9093,localhost:9094 --topic mirrormakerPOC --group topic\_group\_1

1. Monitor list of topics,  details of topic and offset for particular consumer-group.

kafka-topics.sh --list --zookeeper localhost:2182

kafka-topics.sh --list --zookeeper localhost:2181

kafka-topics.sh --describe --zookeeper localhost:2182 --topic mirrormakerPOC

kafka-topics.sh --describe --zookeeper localhost:2181 --topic mirrormakerPOC

kafka-consumer-groups.sh --bootstrap-server localhost:9095,localhost:9096 --group topic\_group\_2 –describe