**Kafka and zookeeper installation**

<https://www.agiratech.com/kafka-zookeeper-multi-node-cluster-setup/>

**Kafka commands**

*Create topic:*

kafka-topics.sh --zookeeper $ZK\_HOSTS --create --topic $TOPIC\_NAME --replication-factor 3 --partitions 1

kafka-topics.sh --zookeeper 192.168.24.104:3701 --create --topic DropcopyEx --replication-factor 2 --partitions 1

*Describe topic:*

kafka-topics.sh --describe --zookeeper <zkhost>:<zkPort> --topic <topic name>

bin/kafka-topics.sh --describe --zookeeper 192.168.26.133:3701 --topic Dropcopy-Ex

This will print the topic details like, number of partitions, replication factor, leader, replicas and in-sync replicas

*Change topic retention time: (Server level Default is 7 days)*

*(All topics used in Sentinel have 48hrs retention)*

kafka-topics.sh --zookeeper <zkhost>:<zkPort> --alter --topic <topic name> --config retention.ms= <desired time in milliseconds>

*Purge topic:*

Temporarily update the retention time on the topic to say 12hrs

kafka-topics.sh --zookeeper <zkhost>:<zkPort> --alter --topic <topic name> --config retention.ms= 43200000

Then wait for the purge to take effect. Once purged reset the retention.ms property to the old value of 48hrs

kafka-topics.sh --zookeeper <zkhost>:<zkPort> --alter --topic <topic name> --config retention.ms= 172800000

*Get current offset of any topic:*

kafka-run-class.sh kafka.tools.GetOffsetShell --broker-list <kafka-ip>:<kafka-port> --topic <topic name> --time -1

*Check lag of all topics in a group (lag means the number of message which are published but have not been consumed by*

kafka-consumer-groups.sh --bootstrap-server <kafka-ip>:<kafka-port> --describe --group <group name>

*List all topics inside a broker :*

kafka-topics.sh --list –zookeeper <zkhost>:<zkPort>

bin/kafka-topics.sh --list --zookeeper 192.168.26.133:3701

(Here 3701 is zookeeper port which is defined in zookeeper.connect property in server.properties)

*Produce a message:*

bin/kafka-console-producer.sh --broker-list broker\_1\_ip:port broker\_2\_ip:port broker\_n\_ip:port --topic topic\_name

bin/kafka-console-producer.sh --broker-list 192.168.26.133:7701 --topic AKRM

(Here port is 7701 for broker which is mentioned in Listener property in server.properties)

*Consume a message: (from any broker)*

bin/kafka-console-consumer.sh --zookeeper zookeeper\_ip:port --from-beginning --topic topic\_name

bin/kafka-console-consumer.sh --bootstrap-server 192.168.26.133:7701 --from-beginning --topic AKRM

*Alter the Replication factor of an already created Topic:*

1. Increase the partition to match the replication factor

bin/kafka-topics.sh --zookeeper 192.168.26.133:3701 --alter --topic AK\_TEST --partitions 3

* Then do a describe to check that the number of partitions have now increased from 1 to 3
* bin/kafka-topics.sh --describe --zookeeper 192.168.26.133:3701 --topic AK\_TEST

1. Now create a json file with custom configuration details for the new replication factor details

using below format :

{"version":1,

"partitions":[

{"topic":"AK\_TEST","partition":0, "replicas":[0,1,2]},

{"topic":"AK\_TEST","partition":1, "replicas":[0,1,2]},

{"topic":"AK\_TEST","partition":2, "replicas":[0,1,2]}

]}

1. Now use the above json to replicate the topics across all brokers

bin/kafka-reassign-partitions.sh --zookeeper 192.168.26.133:3701 --reassignment-json-file <<FILENAME.json> --execute

bin/kafka-reassign-partitions.sh --zookeeper 192.168.26.133:3701 --reassignment-json-file bin/increase-replication-factor.json --execute

* Then do a describe to check that the number of partitions have now increased from 1 to 3
* bin/kafka-topics.sh --describe --zookeeper 192.168.26.133:3701 --topic AK\_TEST

*Delete a topic:*

1. Stop Kafka server
2. Make sure that delete.topic.enable = true is set in properties file
3. Use below command :

bin/kafka-topics.sh --zookeeper 192.168.26.133:3701 --delete --topic AK-RETEST

1. Check under kafka-logs that the corresponding topic is now deleted Or check under /brokers/topics folder (based on your configurations) if topic is deleted or not
2. Try recreating topic with same name and config to make sure it is deleted.

*To get the latest offset number of a message on a given topic:*

/home/mcxuat/kafka\_2.12-0.11.0.0/bin/kafka-run-class.sh kafka.tools.GetOffsetShell --broker-list 192.168.26.133:7701 --topic DropcopyEx --time -1

*To start kafka:*

nohup ./bin/kafka-server-start.sh config/server.properties &

nohup ./bin/kafka-server-start.sh config/server-1.properties &

*To start zookeeper:*

cd zookeeper-3.4.9/

./bin/zkServer.sh start

*Kill Kafka and Zookeeper:*

Ps-ef | grep kafka -> kill kafka pid

Ps –ef | grep zookeeper -> kill zookeeper pid

First bring up zookeeper then restart kafka

*Default properties while creating kafka topic from java code :*

offsets.topic.replication.factor=1

transaction.state.log.replication.factor=1

transaction.state.log.min.isr=1

num.partitions=1

auto.create.topics.enable = true

*Meta Properties path (defined in server.properties file as logs.dir)* : /home/app-admin/kafka\_messages/meta.properties

*To identify underlying brokers:*

*Inside kafka folder:*

./bin/zookeeper-shell.sh 192.168.24.105:3701 ls /brokers/ids

You would see something like below: WATCHER::  WatchedEvent state:SyncConnected type:None path:null [0, 1, 2]

IF no broker is attached to the zookeeper the below error will occur during topic creation:

**Error while executing topic command : replication factor: 1 larger than available brokers: 0**

And ls /brokers/ids will show empty array : type:None path:null []

**Solution:** created fresh server-1.properties with new broker.id. Updated the same broker.id in /home/app-admin/aashka folder and used this folder as *logs.dir* value in server-1.properties file. Then started kafka using server-1.properties as parameter :

**nohup ./bin/kafka-server-start.sh config/server-1.properties &**

*Steps to hard delete a topic :*

- cd zookeeper-3.4.9/

- bin/zkCli.sh -server 192.168.24.105:3701

- ls /brokers/topics -> rmr /brokers/topics/TOPICNAME

- ls /config/topics -> rmr /config/topics/TOPICNAME

- ls /admin/delete\_topics -> rmr /admin/delete\_topics

* Delete the topicName from kafka\_messages folder too (once the log file is empty).

To check if zookeeper and kafka are up:

* netstat -anp | grep 3701 //zookeeper
* netstat -anp | grep 7701 //kafka