STEPS TO SET UP MULTI NODE KAFKA AND ZOOKEEPER CLUSTER

To deploy Kafka in cluster, first there is need to set up Zookeeper cluster. This document mentions steps to setup both.

Apache Kafka:

Apache Kafka is a high-throughput distributed messaging system is designed to allow a single cluster to serve as the central data backbone for a large organization. It can be elastically and transparently expanded without downtime. Data streams are partitioned and spread over a cluster of machines to allow data streams larger than the capability of any single machine and to allow clusters of coordinated consumer.

Apache Zookeeper:

Zookeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. All of these kinds of services are used in some form or another by distributed applications. Each time they are implemented there is a lot of work that goes into fixing the bugs and race conditions that are inevitable. Because of the difficulty of implementing these kinds of services, applications initially usually skimp on them, which make them brittle in the presence of change and difficult to manage. Even when done correctly, different implementations of these services lead to management complexity when the applications are deployed.

Pre-requisite:

1. Install JDK: Java 1.7/1.8
2. Download Zookeeper as per platform requirement (<http://zookeeper.apache.org/releases.html#download>)

Document refers to version 3.4.8 for zookeeper.

1. Download Kafka as per platform requirement (<https://kafka.apache.org/downloads>)

Document conforms to 0.9.0.0 version of Kafka.

Set up zookeeper cluster:

Kafka uses Apache Zookeeper to maintain the list of brokers that are currently members of a cluster.

It is used to store the kafka cluster meta data. It also assists in leader election.

The cluster of the zookeeper is known as **ensemble**.

Every time a broker process starts, it registers itself with its ID in Zookeeper by creating an **ephemeral node**.

Different Kafka components subscribe to the **/brokers/ids** path in Zookeeper where brokers are registered so they get notified when brokers are added or removed.

***To summarize, Kafka uses Zookeeper’s ephemeral node feature to elect a controller and to notify the controller when nodes join and leave the cluster***.

Zookeeper cluster should have odd number of nodes e.g. 3, 5, 7 etc.

**Quorum** - Minimum number of zookeeper nodes that should be up & running in ensemble for whole cluster to work.

Q = (N/2)

Number of nodes Quorum

------------------------------------------------------------------------------

3 2

5 3

7 4

**Execute following commands on each machines**:

1. Extract the downloaded zookeeper package at appropriate location on each of the machines

As: e.g. /home/zookeeper/zk-server-1

1. Create data directory

As: mkdir /home/zookeeper/data

1. Create Log directory

As: mkdir /home/zookeeper/log

1. On each of the machine where zookeeper is to be installed in cluster, execute following command: echo “1” > /home/zookeeper/data/myid

Note: On first machine keep it as 1, second 2 and third machine 3 viz:

|  |  |
| --- | --- |
| **Machine** | **Command** |
| First machine in cluster | echo “1” > /home/zookeeper/data/myid |
| Second machine in cluster | echo “2” > /home/zookeeper/data/myid |
| Third machine in cluster | echo “3” > /home/zookeeper/data/myid |

And so on.

1. Now open the **zoo.cfg** file from /home/zookeeper/zk-server-1/conf/zoo.cfg and make following changes:
   1. Set up data directory in zoo.cfg

# the directory where the snapshot is stored.

# do not use /tmp for storage, /tmp here is just

# example sakes.

dataDir=/home/zookeeper/data

* 1. Set up log directory in zoo.cfg

dataLogDir=/home/zookeeper/log

* 1. Set up client port

# the port at which the clients will connect

clientPort=2184

* 1. Set up zookeeper server and its port details

# Zookeeper ensemble must know about every other machine in ensemble

# specify the server ids by creating ‘myid’ file in dataDir

# User hostname instead of IP addresses preferably for maintenance purpose

server.1=1X.1XX.1X2.101:5888:6888

server.2=1X.1XX.1X2.102:5888:6888

server.3=1X.1XX.1X2.103:5888:6888

***In case you don’t find zoo.cfg but zoo\_sample.cfg, rename the same to zoo.cfg***.

1. Optionally you can set up logging for zookeeper by altering contents of log4j.properties in conf directory.
2. Command to start/stop/check status of zookeeper on each machine:

cd /home/zookeeper/zk-server-1/bin ./zkServer.sh start ./zkServer.sh status ./zkServer.sh stop

Steps to set up Kafka cluster:

1. Extract the downloaded kafka install package in appropriate directory on each of the machines

e.g: /home/kafka/kafka\_2.11-0.9.0.0

Also create directory for log as: /home/kafka/logs

1. Now open the **server.properties** from /home/kafka/kafka\_2.11-0.9.0.0/config directory and add following details:
   1. Edit the server/broker id:

Change broker.id=1

<<*For other machines keep it as 2, 3 etc. so that each broker on different machine will have different id*>>

* 1. Edit server socket settings as:

listeners=PLAINTEXT://:9095

# The port the socket server listens on

port=9095

* 1. # Hostname the broker will bind to. If not set, the server will bind to all interfaces

# These are IPs of this machine (localhost machine)

host.name=1X.13X.1X1.19X

advertised.host.name=1X.13X.1X1.19X

* 1. Edit log directory information

# A comma seperated list of directories under which to store log files

log.dirs=/home/kafka/logs

* 1. Add zookeeper information

# Zookeeper connection string (see zookeeper docs for details).

# This is a comma separated host:port pairs, each corresponding to a zk

# server. e.g. "127.0.0.1:3000,127.0.0.1:3001,127.0.0.1:3002".

# You can also append an optional chroot string to the urls to specify the

# root directory for all kafka znodes.

zookeeper.connect=1X.1XX.1X2.101:2184, 1X.1XX.1X2.102:2184, 1X.1XX.1X2.103:2184

* 1. Add additional properties as :

delete.topic.enable=true

leader.imbalance.check.interval.seconds=10

1. Now save the **server.properties**. Ensure these changes are done on each machine.
2. Command to start kafka broker on each machine:

/home/kafka/kafka\_2.11-0.9.0.0/bin/kafka-server-start.sh -daemon /home/kafka/kafka\_2.11-0.9.0.0/config/server1.properties

Command to stop kafka broker on each machine:

/home/kafka/kafka\_2.11-0.9.0.0/bin/kafka-server-stop.sh

Note: Kafka deployment package comes with zookeeper. So there are two options, one to use this zookeeper that is zipped with kafka package or second is to use zookeeper separately (as explained above).

The commands to start up the zookeeper in that case will be: (On all 3 machines)

Command: nohup /home/kafka/kafka\_2.11-0.9.0.0/bin/zookeeper-server-start.sh config/zookeeper.properties &

References:

<https://myjeeva.com/zookeeper-cluster-setup.html>

<https://rklicksolutions.wordpress.com/2016/05/02/kafka-zookeeper/>

<http://armourbear.blogspot.in/2015/03/setting-up-multinode-kafka-cluster.html>

https://www.youtube.com/watch?v=ipbBfBw-STk