1. Create two folders: ApacheZookeeper and ApacheKafka respectively to install zookeeper and Kafka in a directory of your choice. For e.g.:

**cd /home**

**mkdir ApacheZookeeper**

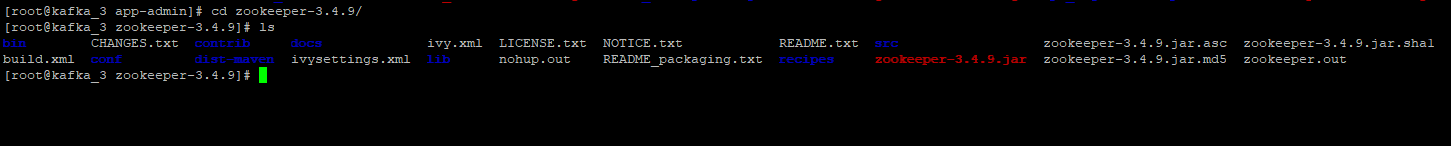
**mkdir ApacheKafka**

Now you should have two directories inside /home path

1. Navigate inside ApacheZookeeper: **cd /home/ApacheZookeeper** and unzip the zookeeper-3.4.9.tar.gz file here to setup Zookeeper.

**tar -zxf zookeeper-3.4.9.tar.gz**

Now you should have zookeeper-3.4.9 folder along with tar folder in this directory.



1. Navigate to **conf** folder and update the **zoo.cfg** file with the IP details and Port of your respective zookeeper. Attaching below one **zoo.cfg** file for reference.



Figure 1: zoo.cfg

Please note that **clientPort = zookeeper port** and **server.1, server.2** will be the IP address of zookeeper servers.

Also note **dataDir=/tmp/zookeeper** path which will contain various id files related to that particular zookeeper.

1. Navigate to **dataDir=/tmp/zookeeper** and make sure to assign a unique id to each zookeeper in each respective brokers where you are setting up zookeeper and kafka. In other words, each zookeeper server should have a unique number in the **myid** file. For example, server 1 will have value 1, server 2 will have value 2 and so on.

**cd /tmp/zookeeper**

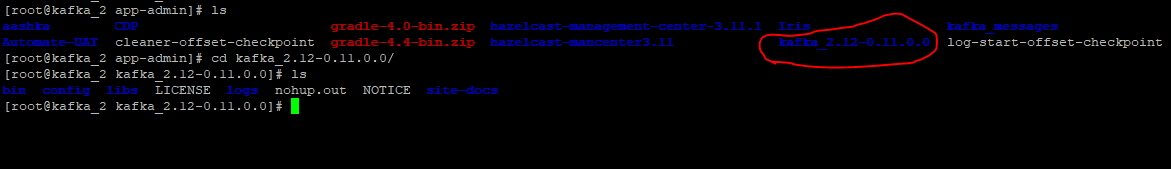
**echo ‘1’ > myid** (For zookeeper 1)

Similarly **echo ‘2’ > myid** and **echo ‘3’ > myid** for zookeeper in other servers as per your configuration.

1. Navigate Inside ApacheKafka : **cd /home/ApacheKafka** and unzip the kafka\_2.12-0.11.0.0.tgz file here to setup Kafka

**tar –zxf kafka\_2.12-0.11.0.0.tgz**

Now you should have kafka\_2.12-0.11.0.0 folder along with tar folder in this directory.



1. Navigate inside kafka folder to **config** folder :

**cd /home/app-admin/ApacheKafka/kafka\_2.12-0.11.0.0/config**

**vi zookeeper.properties**

Make sure **dataDir=/tmp/zookeeper** is set here. Attaching zookeeper.properties file for reference



Figure 2 : zookeeper.properties

1. Navigate inside kafka folder to **config** folder :

**cd /home/app-admin/ApacheKafka/kafka\_2.12-0.11.0.0/config/**

**vi server.properties**

Make sure below properties are updated as per your Broker configuration:

# The id of the broker. This must be set to a unique integer for each broker.

**broker.id=1** (Unique id for each broker)

**port=7701** (Broker port which will be different from Zookeeper port)

**listeners = PLAINTEXT://192.168.24.104:7701**  (IP of each broker)

**log.dirs=/home/kafka\_messages/**  (Unique directory to store kafka logs)

**num.partitions=1** (number of partitions as per your configuration need)

**zookeeper.connect=192.168.24.104:3701** (Give your zookeeper IP and port)

**offsets.topic.replication.factor = 3** (here 3 = no. of brokers. You can change this value to suit your configuration. Default value =1)

These are some primary properties. You can change any additional properties based on your requirement and configuration. Attaching a reference **server.properties** file.



Figure 3 : server.properties

Once all the Zookeeper and Kafka configuration is done as per above mentioned steps we can now proceed to start respective setup.

1. Start Zookeeper in all the machines wherever it is installed :

**cd zookeeper-3.4.9/**

**./bin/zkServer.sh start**

You should be able to see “**Starting zookeeper ...”** message.

Keep checking zookeeper.out file to track the progress of starting zookeeper.

Alternatively, you can fire below commands to see if zookeeper is started successfully:

**ps –ef | grep zookeeper**

**netstat –anp | grep 3701** (Here 3701 is zookeeper port. Make sure atleast one process is in LISTEN state on this port)

1. Once Zookeeper is successfully running on all machines, you can now proceed to start Kafka in all the machines using below commands :

**cd kafka\_2.12-0.11.0.0/**

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

Run this command on all machines where you have successfully started zookeeper and now aiming to start kafka. To check if Kafka is up and running fine keep checking nohup.out file to track the progress of starting Kafka.

Alternatively you can fire below commands to see if zookeeper is started successfully:

**ps –ef | grep kafka\_2.12-0.11.0.0**

**netstat –anp | grep 7701**  (Here 7701 is kafka port. Make sure atleast one process is in LISTEN state on this port)

1. Once Kafka is running fine with all specified configurations, you can now create topics from within Kafka folder.

**cd kafka\_2.12-0.11.0.0/**

Below is the command to create a Kafka topic :

**kafka-topics.sh --zookeeper $ZK\_HOSTS --create --topic $TOPIC\_NAME**

**--replication-factor 3 --partitions 1**

Example create topic “DropcopyEx”:

kafka-topics.sh --zookeeper 192.168.24.104:3701 --create --topic DropcopyEx

--replication-factor 3 --partitions 1

Keep replication-factor = number of brokers

Partitions = value specified in server.properties file as per your configuration.

Topic = name of topic that you want to create

DropcopyEx is the primary consumer topic in Sentinel. Below are the list of other topics that are used by Sentinel application. Feel free to create as per your need using above command only and changing the topic name in it:

4101

4102

4103

4104

4105

4106

4107

4108

4109

Transformed\_Transaction

Transformed\_Transaction\_partition