**Zookeeper configuration:**

* **We created data directories for all the zookeeper nodes to store data**

mkdir -p /usr/local/kafka/data/zookeeper1

mkdir -p /usr/local/kafka/data/zookeeper2

mkdir -p /usr/local/kafka/data/zookeeper3

* **We created zookeeper properties files**

sudo nano /usr/local/zookeeper/config/zookeeper1.properties

sudo nano /usr/local/zookeeper/config/zookeeper2.properties

sudo nano /usr/local/zookeeper/config/zookeeper3.properties

|  |  |  |
| --- | --- | --- |
| Zookeeper-1:  zookeeper1.properties:  **dataDir=** **/usr/local/kafka/data/zookeeper1**  **clientPort=2181**  **tickTime=2000**  **initLimit=5**  **syncLimit=2**  **server.1=localhost:2666:3666**  **server.2=localhost:2667:3667**  **server.3=localhost:2668:3668**  **maxClientCnxns=0** | Zookeeper -2:  zookeeper2.properties:  **dataDir=** **/usr/local/kafka/data/zookeeper2**  **clientPort=2182**  **tickTime=2000**  **initLimit=5**  **syncLimit=2**  **server.1=localhost:2666:3666**  **server.2=localhost:2667:3667**  **server.3=localhost:2668:3668**  **maxClientCnxns=0** | Zookeeper -3:  Zookeeper3.properties:  **dataDir=** **/usr/local/kafka/data/zookeeper3**  **clientPort=2183**  **tickTime=2000**  **initLimit=5**  **syncLimit=2**  **server.1=localhost:2666:3666**  **server.2=localhost:2667:3667**  **server.3=localhost:2668:3668**  **maxClientCnxns=0** |

**dataDir –** the location to store the in-memory database snapshots and, unless specified otherwise, the transaction log of updates to the database

**clientPort -** the port to listen for client connections

**ticktime -** the basic time unit in milliseconds used by ZooKeeper.

**initLimit -** is timeouts ZooKeeper uses to limit the length of time the ZooKeeper servers in quorum have to connect to a leader

(In this case the timeout for initLimit is 5 ticks at 2000 milliseconds a tick, or 10 seconds)

**syncLimit** **-** Amount of time, in ticks, to allow followers to sync with ZooKeeper

**maxClientCnxns** **-** Maximum number of concurrent connections that a single client can make to a single member of the ZooKeeper ensemble

(As this vale is set to 0, there is no limit of connections)

* **Opened the in-bound ports 2181, 2182 and 2183 in the AWS EC2 console for the zookeeper nodes.**
* **Opened the in-bound ports 2666, 2667, 2668 in the AWS EC2 console for followers to connect to the leaders**
* **Opened the in-bound ports 3666, 3667, 3668 in the AWS EC2 console for leader election**
* **We created the unique id for each zookeeper instance**

echo 1 | sudo tee /usr/local/kafka/data/zookeeper1/myid

echo 2 | sudo tee /usr/local/kafka/data/zookeeper2/myid

echo 3 | sudo tee /usr/local/kafka/data/zookeeper3/myid

**Kafka configuration:**

* **We created 3 brokers**

cp /usr/local/kafka/config/server.properties /usr/local/kafka/config/server-1.properties

cp /usr/local/kafka/config/server.properties /usr/local/kafka/config/server-2.properties

cp /usr/local/kafka/config/server.properties /usr/local/kafka/config/server-3.properties

sudo nano /usr/local/kafka/config/server-1.properties

sudo nano /usr/local/kafka/config/server-2.properties

sudo nano /usr/local/kafka/config/server-3.properties

|  |  |  |
| --- | --- | --- |
| Broker-1:  config/server-1.properties:  **broker.id=0**  **listeners=PLAINTEXT://*<Public DNS>*:9093**  **offsets.topic.replication.factor=3**  **transaction.state.log.replication.factor=3**  **transaction.state.log.min.isr=2**  **log.dir=/tmp/kafka-logs-0**  **log.retention.hours = 48**  **zookeeper.connect=localhost:2181,localhost:2182,localhost:2183** | Broker-2:  config/server-2.properties:  **broker.id=1**  **listeners=PLAINTEXT://*<Public DNS>*:9094**  **offsets.topic.replication.factor=3**  **transaction.state.log.replication.factor=3**  **transaction.state.log.min.isr=2**  **log.dir=/tmp/kafka-logs-1**  **log.retention.hours = 48**  **zookeeper.connect=localhost:2181,localhost:2182,localhost:2183** | Broker-3:  config/server-3.properties:  **broker.id=2**  **listeners=PLAINTEXT://*<Public DNS>*:9095**  **offsets.topic.replication.factor=3**  **transaction.state.log.replication.factor=3**  **transaction.state.log.min.isr=2**  **log.dir=/tmp/kafka-logs-2**  **log.retention.hours = 48**  **zookeeper.connect=localhost:2181,localhost:2182,localhost:2183** |

**(The parameters not mentioned here remained with the default values)**

**broker.id –** broker id

**listeners –** the address the broker socket listens on

**offsets.topic.replication.factor -** specify the replication factor for the \_\_consumer\_offsets topic. This topic stores information about committed offsets for each topic:partition per group of consumers

(We’ve set this value to 3 to take advantage of having 3 brokers, providing more redundancy for this information)

**transaction.state.log.replication.factor -** the replication factor for the transaction topic. Internal topic creation will fail until the cluster size meets this replication factor requirement.

(We’ve set this value to 3 to take advantage of having 3 brokers, providing more redundancy for this information)

**transaction.state.log.min.isr -** minimum ISR for this topic

(All the topics will have at least the leader and one replica in sync to continue to provide service)

**log.dir** **–** the directory in which the log data is kept

**log.retention.hours** **-** the number of hours to keep a log file before deleting it

(As mentioned before we’ve set it to 48, the messages will be kept for 48 hours before they are deleted)

**zookeeper.connect -** ZooKeeper connection string

(Contains the addresses for the zookeeper nodes)

* **Opened the in-bound ports 9093, 9094 and 9095 in the AWS EC2 console for the Kafka brokers.**