

Apache Kafka

Zookeeper:

First thing to consider for running kafka is to install and run zookeeper. You can download it from <http://www-us.apache.org/dist/zookeeper/>. After downloading you should extract the downloaded file to /opt/ path by first changing directory to /opt/ and then using the following command

```
tar -xzf /path/to/download/zookeeper-3.4.13.tar.gz
```

Then you should make a folder called data in zookeeper folder

```
cd zookeeper-3.4.13  
mkdir data
```

After that you need to create the config file in conf folder and set its values

```
vim conf/zoo.cfg
```

Insert following details in the config file

```
tickTime=2000  
dataDir=/opt/zookeeper-3.4.13/data  
clientPort=2181  
initLimit=5  
syncLimit=2
```

Now everything is ready for zookeeper. For running it you should run the following command from zookeeper directory

```
bin/zkServer.sh start
```

And to stop zookeeper use this command

```
bin/zkServer.sh stop
```

You can check 2181 port to see if zookeeper is running or not using

```
netstat -ant | grep :2181
```

Please note that zookeeper is already up and running (unless someone has stopped it!) on server1 (IP : 96...)

Kafka Installation:

To install kafka simply download it via

http://www-eu.apache.org/dist/kafka/1.1.0/kafka_2.11-1.1.0.tgz. Then extract it to /opt/ directory like zookeeper using

```
tar -xzf /path/to/download/kafka_2.11-1.1.0.tgz
```

You can change config for a broker in config/server.properties file. Remember that you need to set zookeeper ip:port and broker id in the config file. Each broker needs a different broker id so if you want to use multiple brokers with same zookeeper you need to assign them different ids. Set zookeeper address and broker id using following syntax in config/server.properties

```
broker.id=0  
zookeeper.connect=zookeeper_ip:2181
```

Please note that Kafka is already installed on both servers in /opt/ directory.

Running Kafka Brokers and Creating Topics:

Default value of broker id in server.properties is 0. To run a broker using this config you should enter this command

```
bin/kafka-server-start.sh config/server.properties
```

In this project three brokers are going to be a good idea which two of them must be run on the same server. So on one server we are going to have server0.properties and server1.properties with ids 0 and 1 and on the other server we are going to have server2.properties with id 2. To stop a broker you can run

```
bin/kafka-server-stop.sh config/server.properties
```

Next command shows how to create a topic

```
bin/kafka-topic.sh --create --zookeeper localhost:2181  
--replication-factor 1 --partitions 1 --topic testing
```

Change localhost with zookeeper ip, replication factor and partitions with number of brokers we are going to use (in this case 3) and testing with real topic name. To see list of all topics you can use

```
bin/kafka-topics.sh --list --zookeeper localhost:2181
```

Kafka Producer/Consumer:

To run a kafka producer you should enter

```
bin/kafka-console-producer.sh --broker-list localhost:9092  
--topic testing
```

The --broker-list argument is a list of all brokers ip:port that you want to connect to. To check a broker's listening port you can see the listeners value in config/server.properties file. After sending all messages you can finish kafka producer using ctrl+D.

To run a kafka consumer there are two version of commands. The old one connects to zookeeper but the new one connects to broker itself.

The old one:

```
bin/kafka-console-consumer.sh --zookeeper localhost:2181  
--topic testing
```

The new one:

```
bin/kafka-console-consumer.sh --bootstrap-server  
localhost:9092 --topic testing
```

However the new command doesn't work on my computer despite downloading newest version of kafka. I hope you can handle it in some way. By adding `--from-beginning` to these commands you can see all previous messages of the topic.

For producer/consumer Java APIs you can check its [producer](#) and [consumer](#) javadocs. Required dependencies are already added to pom.xml and these javadocs contain sample codes which I couldn't get a result from but I hope you can!