https://www.digitalocean.com/community/tutorials/how-to-install-apache-kafka-on-ubuntu-14-04

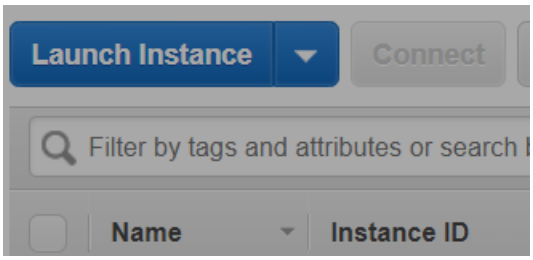
https://dzone.com/articles/installing-and-running-kafka-on-an-aws-instance

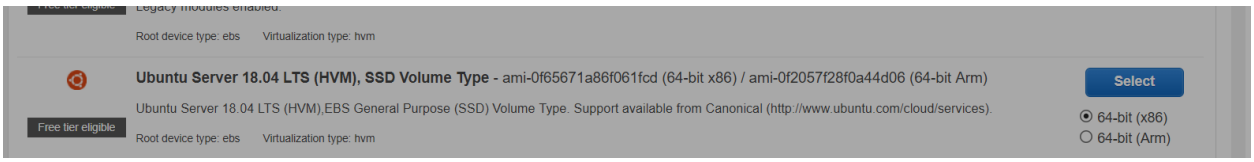
https://www.tutorialspoint.com/apache_kafka/apache_kafka_installation_steps.htm

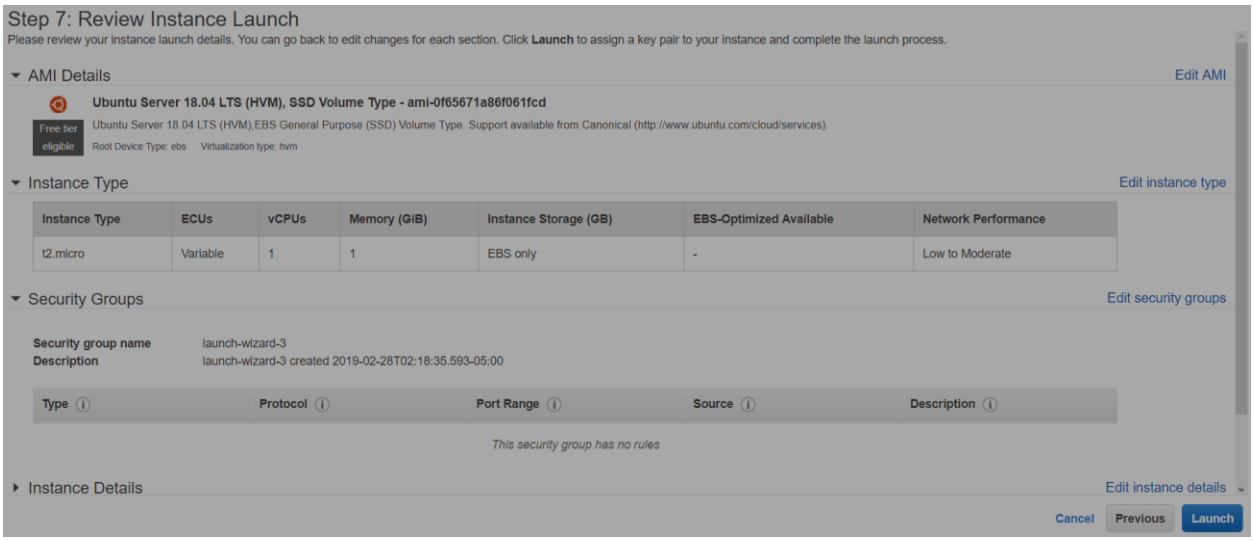
https://www.tutorialspoint.com/apache_kafka/apache_kafka_basic_operations.htm

open all links.

Create ubuntu instance on aws and launch it



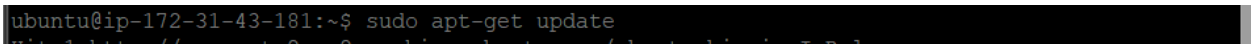




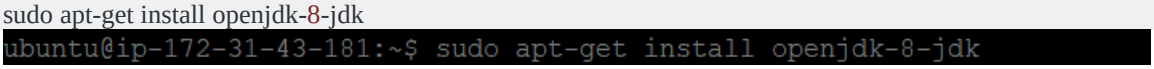
Open SSH client(or putty)

Install JAVA:

```
sudo apt-get update
```




```
sudo apt-get install openjdk-8-jdk
```



Step 3 — Install ZooKeeper

Apache ZooKeeper is an open source service built to coordinate and synchronize configuration information of nodes that belong to a distributed system. A Kafka cluster depends on ZooKeeper to perform—among other things—operations such as detecting failed nodes and electing leaders.

Since the ZooKeeper package is available in Ubuntu's default repositories, install it using apt-get.

```
sudo apt-get install zookeeperd
```

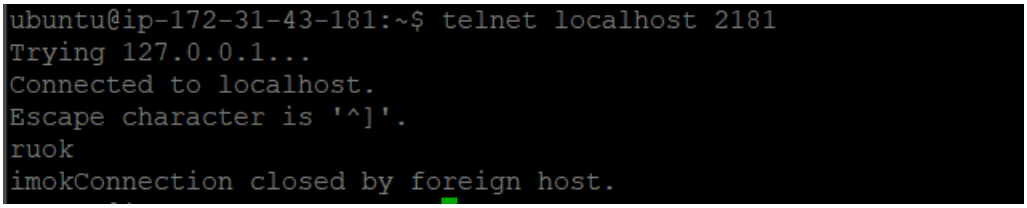
After the installation completes, ZooKeeper will be started as a daemon automatically. By default, it will listen on port **2181**.

To make sure that it is working, connect to it via Telnet:

```
telnet localhost 2181
```

At the Telnet prompt, type in `ruok` and press `ENTER`.

If everything's fine, ZooKeeper will say `imok` and end the Telnet session.

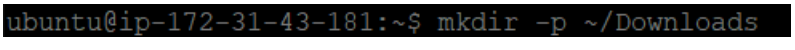


Step 4 — Download and Extract Kafka Binaries

Now that Java and ZooKeeper are installed, it is time to download and extract Kafka.

To start, create a directory called `Downloads` to store all your downloads.

```
mkdir -p ~/Downloads
```



From this link https://archive.apache.org/dist/kafka/0.8.2.1/ download
https://archive.apache.org/dist/kafka/0.8.2.1/kafka_2.11-0.8.2.1.tgz

```
wget "https://archive.apache.org/dist/kafka/0.8.2.1/kafka_2.11-0.8.2.1.tgz" -O
~/Downloads/kafka.tgz
```

```
ubuntu@ip-172-31-43-181:~$ wget "https://archive.apache.org/dist/kafka/0.8.2.1/kafka_2.11-0.8
.2.1.tgz" -O ~/Downloads/kafka.tgz
--2019-02-28 07:32:31--  https://archive.apache.org/dist/kafka/0.8.2.1/kafka_2.11-0.8.2.1.tgz
Resolving archive.apache.org (archive.apache.org)... 163.172.17.199
Connecting to archive.apache.org (archive.apache.org)|163.172.17.199|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 15771850 (15M) [application/x-gzip]
Saving to: '/home/ubuntu/Downloads/kafka.tgz'

/home/ubuntu/Downloads/ 100%[=========================>]  15.04M  8.01MB/s    in 1.9s

2019-02-28 07:32:33 (8.01 MB/s) - '/home/ubuntu/Downloads/kafka.tgz' saved [15771850/15771850
]
```

```
mkdir -p ~/kafka && cd ~/kafka
```

```
tar -xvzf ~/Downloads/kafka.tgz --strip 1
```

```
ubuntu@ip-172-31-43-181:~$ mkdir -p ~/kafka && cd ~/kafka
ubuntu@ip-172-31-43-181:~/kafka$ tar -xvzf ~/Downloads/kafka.tgz --strip 1
```

Step 5 — Configure the Kafka Server

The next step is to configure the Kakfa server.

```
nano ~/kafka/config/server.properties
```

```
ubuntu@ip-172-31-43-181:~/kafka$ nano ~/kafka/config/server.properties
```

By default, Kafka doesn't allow you to delete topics. To be able to delete topics, add the following line at the end of the file:

```
delete.topic.enable = true
```

Since Kafka uses Zookeeper, we need to first start a Zookeeper server. We can use the convenience script packaged with Kafka to start a single-node Zookeeper instance or we can start Zookeeper on a standalone instance and specify its configurations in **zookeeper.properties** configuration file,

we would be starting it using the convenience script that is packaged with Kafka. Since we have 1 GB RAM we would be
setting **KAFKA_HEAP_OPTS** environment variable in our **.bashrc** to 50% of total RAM ie 250 MB in our case.

```
nano .bashrc
```

```
ubuntu@ip-172-31-43-181:~/kafka$ cd ..
ubuntu@ip-172-31-43-181:~$ nano .bashrc
```

Insert the following environment variable.

```
export KAFKA_HEAP_OPTS="-Xmx250M -Xms250M"
```

```
export KAFKA_HEAP_OPTS="-Xmx250M -Xms250M"
```

After setting the variable, source your .baschrc.

```
source .bashrc
```

Step 6: Start ZooKeeper

bin/zookeeper-server-start.sh config/zookeeper.properties

```
ubuntu@ip-172-31-43-181:~$ ls
Downloads  kafka
ubuntu@ip-172-31-43-181:~$ cd kafka/
ubuntu@ip-172-31-43-181:~/kafka$ bin/zookeeper-server-start.sh config/zookeeper.properties
```

To start Kafka Broker, type the following command in new terminal –

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

```
ubuntu@ip-172-31-43-181:~$ cd kafka/
ubuntu@ip-172-31-43-181:~/kafka$ bin/kafka-server-start.sh config/server.properties
```