

You can choose “VirtualBox” or docker to build a Hadoop/Spark clusters, and you should know that docker is more lightweight to build a cluster than using virtual machines. **You are not requested to use Docker**, but I think it is a good solution. This manual cannot cover all the details and you may find some problems with them, it is necessary for you to search for more materials.

We provide a pre-defined docker, you can pull it from [**https://hub.docker.com/**](https://hub.docker.com/).

The docker’s name is **nusbigdatacs4225/ubuntu-with-hadoop-spark**.

It is a ubuntu image included

1. jdk1.8.0_191(/usr/java)
2. Hadoop 2.8.5(/usr/local/hadoop)
3. Spark 2.2.0(usr/local/spark)

You can use this image to build your own clusters. (you need to check and change the configurations for your own environment)

Here is a simple example for setting a three nodes Hadoop cluster, you can write a script to do all this automatically.

1.download the docker and finish the configuration for the image.

“docker pull nusbigdatacs4225/ubuntu-with-hadoop-spark”

2.run the commands to create three containers.

“docker run -it -h master --name master nusbigdatacs4225/ubuntu-with-hadoop-spark”

“docker run -it -h slave01 --name slave01 nusbigdatacs4225/ubuntu-with-hadoop-spark”

“docker run -it -h slave02 --name slave02 nusbigdatacs4225/ubuntu-with-hadoop-spark”

3.run **“vim /etc/hosts”** to check IP address.

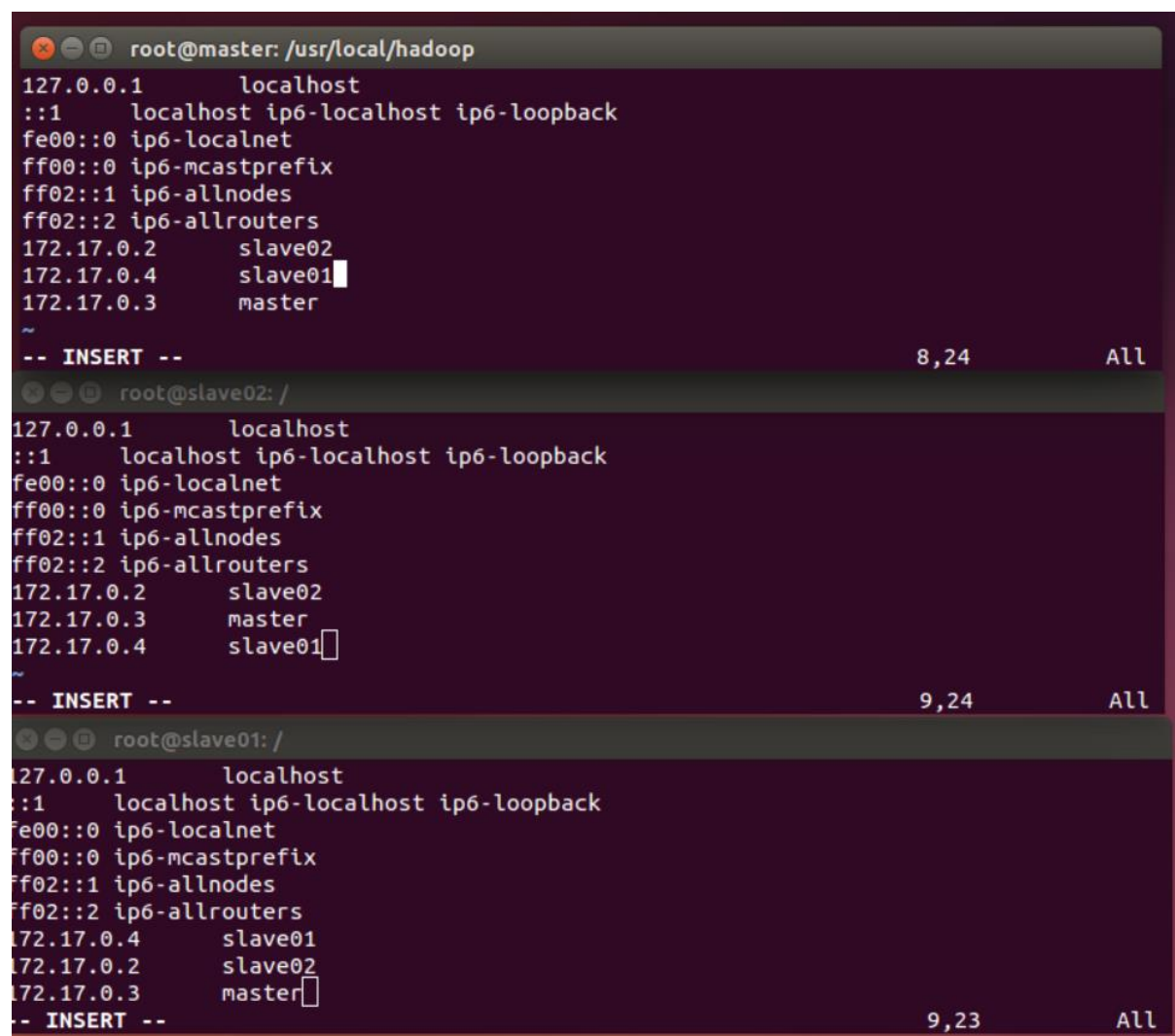
e.g.

master: 172.17.0.2

slave01:172.17.0.3

slave02:172.17.0.4

add this to all three containers. (you can use **“ssh slave01”** in master container to check this configuration)



```
root@master: /usr/local/hadoop
127.0.0.1      localhost
::1           localhost ip6-localhost ip6-loopback
fe00::0       ip6-localnet
ff00::0       ip6-mcastprefix
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
172.17.0.2     slave02
172.17.0.4     slave01
172.17.0.3     master
~
-- INSERT --                                     8,24      All

root@slave02: /
127.0.0.1      localhost
::1           localhost ip6-localhost ip6-loopback
fe00::0       ip6-localnet
ff00::0       ip6-mcastprefix
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
172.17.0.2     slave02
172.17.0.3     master
172.17.0.4     slave01
~
-- INSERT --                                     9,24      All

root@slave01: /
127.0.0.1      localhost
::1           localhost ip6-localhost ip6-loopback
fe00::0       ip6-localnet
ff00::0       ip6-mcastprefix
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
172.17.0.4     slave01
172.17.0.2     slave02
172.17.0.3     master
~
-- INSERT --                                     9,23      All
```

4.In master container, cd /usr/local/hadoop/etc/hadoop, run **“vim slaves”**, add **slave01 slave02** into this file.

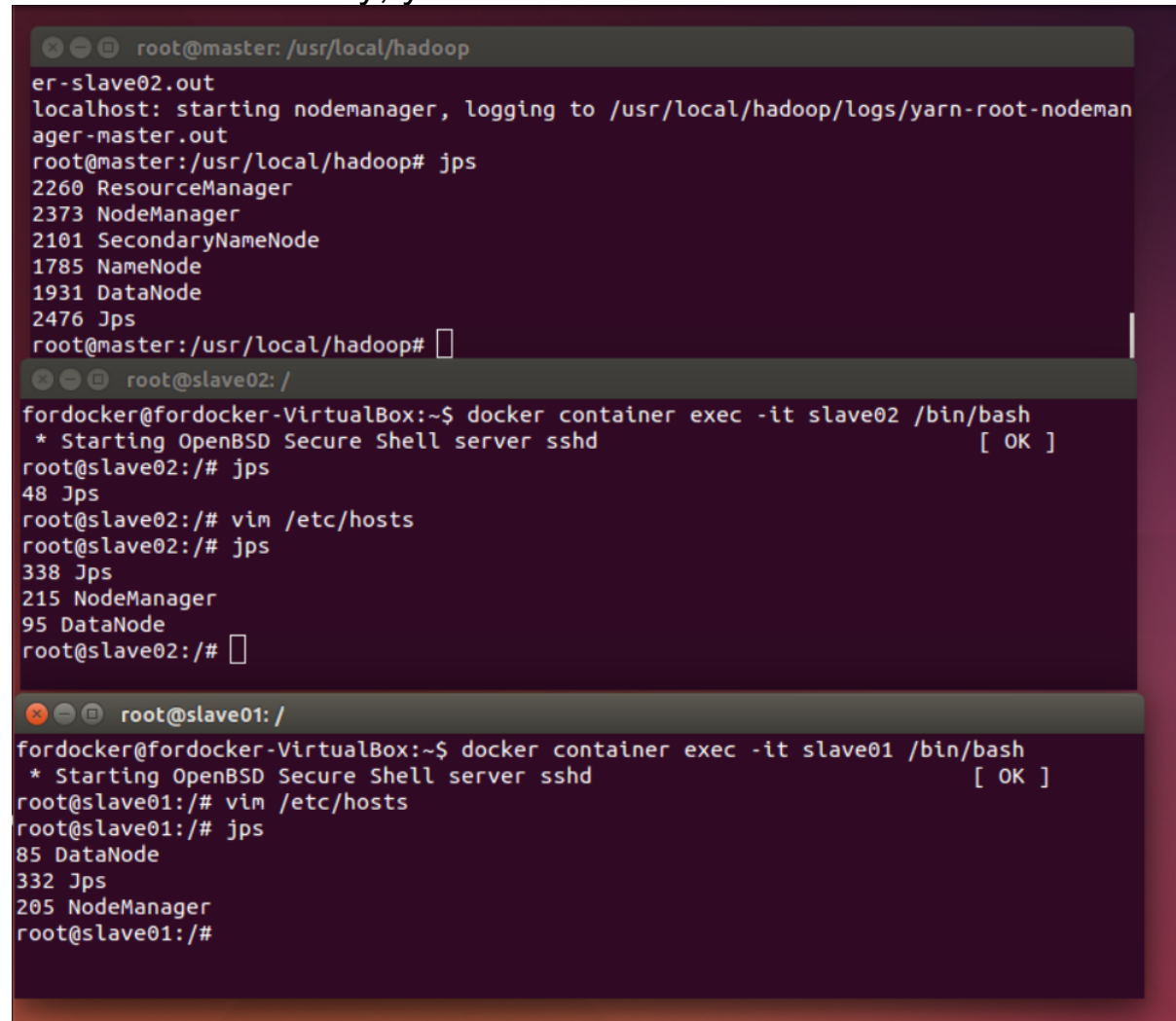
5. initialize the hdfs and run

“cd /usr/local/Hadoop”

“bin/hdfs namenode -format”

“sbin/start-all.sh”

6. if it work correctly, you will see



```
root@master: /usr/local/hadoop
er-slave02.out
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root-nodeman
ager-master.out
root@master:/usr/local/hadoop# jps
2260 ResourceManager
2373 NodeManager
2101 SecondaryNameNode
1785 NameNode
1931 DataNode
2476 Jps
root@master:/usr/local/hadoop#

root@slave02: /
fordocker@fordocker-VirtualBox:~$ docker container exec -it slave02 /bin/bash
* Starting OpenBSD Secure Shell server sshd [ OK ]
root@slave02:/# jps
48 Jps
root@slave02:/# vim /etc/hosts
root@slave02:/# jps
338 Jps
215 NodeManager
95 DataNode
root@slave02:/#

root@slave01: /
fordocker@fordocker-VirtualBox:~$ docker container exec -it slave01 /bin/bash
* Starting OpenBSD Secure Shell server sshd [ OK ]
root@slave01:/# vim /etc/hosts
root@slave01:/# jps
85 DataNode
332 Jps
205 NodeManager
root@slave01:/#
```

For spark environment you need to do it by yourselves.