Below is the high level procedure for creation of 3 node hadoop cluster on a single Ubuntu machine.

$ mkdir demo

$ cd demo

$ tar zxvf ~/hadoop-2.6.0.tar.gz

$ cp -r hadoop-2.6.0 node1

$ cp -r hadoop-2.6.0 node2

$ cp -r hadoop-2.6.0 node3

$ mkdir storage

#### Storage folder is for storing all the nodes’ files. This folder is just for convenience, as it’s easier to restart from scratch, just be deleting the storage folder content.

Node1 will hosts:

HDFS namenode HDFS datanode Resource Manager NodeManager

node1/etc/hadoop/core-site.xml

So, the respectively.

file contains the location of the namenode. Following are core-site.xml, hdfs-site.xml, mapred-site.xml and yarn-site.xml files

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost</value>

</property>

<configuration>

<configuration>

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>/home/user/demo/storage/node1/namenode</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>/home/user/demo/storage/node1/datanode</value>

</property>

</configuration>

<configuration>

<property>

<name>mapreduce.resourcemanager.address</name>

<value>localhost:8032</value>

</property>

</configuration>

<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

<property>

<name>yarn.resourcemanager.scheduler.class</name>

<value>org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.Ca

</property>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

<property>

<name>yarn.nodemanager.webapp.address</name>

<value>localhost:8042</value>

</property>

</configuration>

#### Node2 will hosts:

HDFS datanode Yarn Nodemanager

Below are its configuration properties that needs to be edited respectively for all the above \*-site.xml files

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost</value>

</property>

<configuration>

<configuration>

<property>

<name>dfs.datanode.data.dir</name>

<value>/home/user/demo/storage/node2/datanode</value>

</property>

<property>

<name>dfs.datanode.address</name>

<value>localhost:50110</value>

</property>

<property>

<name>dfs.datanode.ipc.address</name>

<value>localhost:50120</value>

</property>

<property>

<name>dfs.datanode.http.address</name>

<value>localhost:50175</value>

</property>

</configuration>

mapreduce.resourcemanager.address localhost:8032

mapreduce.shuffle.port 13662

mapreduce.framework.name

yarn

yarn.resourcemanager.scheduler.class

org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.CapacityS

yarn.nodemanager.aux-services

mapreduce\_shuffle

yarn.nodemanager.aux-services.mapreduce.shuffle.class

org.apache.hadoop.mapred.ShuffleHandler

yarn.app.mapreduce.am.staging-dir

/tmp/hadoop-yarn/staging

yarn.nodemanager.webapp.address

localhost:8142

yarn.nodemanager.localizer.address

localhost:8140

#### Node3 will hosts:

HDFS datanode Yarn Nodemanager

Similar to node2, node3 has its own below configuration properties that needs to be edited respectively for all the above \*-site.xml files

fs.defaultFS hdfs://localhost

dfs.datanode.data.dir

/home/user/demo/storage/node3/datanode

dfs.datanode.address localhost:50210

dfs.datanode.ipc.address localhost:50220

dfs.datanode.http.address localhost:50275

mapreduce.jobhistory.address localhost:10020

mapreduce.jobhistory.webapp.address localhost:19888

mapreduce.resourcemanager.address localhost:8032

mapreduce.shuffle.port 13762

mapreduce.framework.name

yarn

yarn.resourcemanager.scheduler.class

org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.CapacityS

yarn.nodemanager.aux-services

mapreduce\_shuffle

yarn.nodemanager.aux-services.mapreduce.shuffle.class

org.apache.hadoop.mapred.ShuffleHandler

yarn.app.mapreduce.am.staging-dir

/tmp/hadoop-yarn/staging

yarn.nodemanager.webapp.address

localhost:8242

yarn.nodemanager.localizer.address localhost:8240

#### After making all the above changes, run the below commands from respective directories.

Format the Namenode:

$ cd node1/bin

$ ./hadoop namenode -format

#### We are now ready to start the namenode, datanode, resourcemanager and nodemanager on node1:

$ cd node1/bin

$ ./hadoop namenode &

$ ./hadoop datanode &

$ ./yarn resourcemanager &

$ ./yarn nodemanager &

#### We start a datanode and a nodemanager on node2:

$ cd node2/bin

$ ./hadoop datanode &

$ ./yarn nodemanager &

#### We start a datanode and a nodemanager on node3:

$ cd node3/bin

$ ./hadoop datanode &

$ ./yarn nodemanager &

#### We access to the HDFS web console (http://localhost:50070) to verify that the namenode is able to see the 3 live datanodes and resource manager at http://localhost:8088/cluster .

we can also verify the services running on single machine.

user@ubuntu-1:~$ jps 32738 DataNode

32587 NodeManager

32228 DataNode

349 NodeManager 31716 NameNode 841 Jps

31792 DataNode

32103 NodeManager

31880 ResourceManager user@ubuntu-1:~$