How To Set Up a Hadoop 3.2.1 Multi-Node Cluster on Ubuntu 18.04 (4 Nodes Real Machines)

By:

Muhammad Shahid Bhatti

Pr-requirements

- ➓ Ubuntu 18.04 installed on all machines.
- ➓ All machines are networked.

What are we going to install in order to create the Hadoop Multi-Node Cluster?

- ➓ Java 8;
- ➓ SSH;
- ➓ hadoop-3.2.1

Step 1:

Install SSH using the following command:

```
sudo apt install ssh
```

It will ask you for the password. When it asks for confirmation, just give it.

Step 3:

Open the **.bashrc** file with the following command:

```
nano .bashrc
```

At the end of the file just write the following line:

```
export PDSH_RCMD_TYPE=ssh
```

Step 4:

Now let’s configure SSH. Let’s create a new key using the following command:

```
ssh-keygen -t rsa -P ""
```

Just press **Enter** everytime that is needed.

Step 5:

Now we need to copy the public key to the authorized_keys file with the following command:

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

Step 6:

Now we can verify the SSH configuration by connecting to the localhost:

```
ssh localhost
```

Just type **“yes”** and press **Enter** when needed.

Use the following command to exit.

exit

Step 7:

This is the step where we install Java 8. We use this command:

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

Just as previously, give confirmation when needed.

Step 8:

This step isn’t really a step, it’s just to check if Java is now correctly installed:

```
java -version
```

Step 9:

Download Hadoop using the following command(if not already downloaded):

```
sudo wget -P ~ https://mirrors.sonic.net/apache/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz
```

Step 10:

We need to unzip the **hadoop-3.2.1.tar.gz** file with the following command:

```
tar xzf hadoop-3.2.1.tar.gz
```

Step 11:

Change the **hadoop-3.2.1** folder name to **hadoop** (this maked it easier to use). Use this command:

```
mv hadoop-3.2.1 hadoop
```

Step 12:

"/usr/lib/jvm/java-8-openjdk-amd64"

Open the **hadoop-env.sh** file in the nano editor to edit **JAVA_HOME:**

```
nano ~/hadoop/etc/hadoop/hadoop-env.sh
```

Paste this line to JAVA_HOME:

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/
```

Step 13:

Change the **hadoop** folder directory to **/usr/local/hadoop.** This is the command:

```
sudo mv hadoop /usr/local/hadoop
```

Provide the password when needed.

Step 14:

Open the **environment** file on nano with this command:

```
sudo nano /etc/environment
```

Then, add the following configurations:

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/
usr/local/hadoop/bin:/usr/local/hadoop/sbin"JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64/jre"
```

Step 15:

Now we will add a user called **hadoopuser**, and we will set up it's configurations:

```
sudo adduser hadoopuser
```

Provide the password and you can leave the rest blank, just press **Enter.**

Now type these commands:

```
sudo usermod -aG hadoopuser hadoopuser
sudo chown hadoopuser:root -R /usr/local/hadoop/
sudo chmod g+rwx -R /usr/local/hadoop/
sudo adduser hadoopuser sudo
```

Step 16:

Now we need to verify the machine ip address:

```
ip addr
```

Now, as you can see, IP is 172.16.16.174, just remember this will be different for you, you need to act accordingly when the IP addresses are used later.

My network will be as follows:

master: **172.16.16.174**

slave1: **172.16.16.171**

slave2: **172.16.16.172**

slave3: **172.16.16.173**

In your case, just keep adding 1 to the last number of the IP you get on your machine, just as I did for mine.

Step 17:

Open the **hosts** file and insert your Network configurations:

```
sudo nano /etc/hosts
```

172.16.16.174 hadoop-master

172.16.16.171 hadoop-slave1

172.16.16.172 hadoop-slave2

172.16.16.173 hadoopslave3

Step 18:

On the master VM, open the **hostname** file on nano:

```
sudo nano /etc/hostname
```

Insert the name of your master machine. (note, it's the same name you entered previously on the hosts file)

Now do the same on the slaves:

```
sudo nano /etc/hostname
```

Step 19:

Also, you should reboot all of them so this configuration taked effect:

```
sudo reboot
```

Step 20:

Configure the SSH on **hadoop-master**, with the hadoopuser. This is the command:

```
su - hadoopuser
```

Step 21:

Create an SSH key:

```
ssh-keygen -t rsa
```

Step 22:

Now we need to copy the SSH key to all the users. Use this command on hadoop-master:

```
ssh-copy-id hadoopuser@hadoop-master
ssh-copy-id hadoopuser@hadoop-slave1
ssh-copy-id hadoopuser@hadoop-slave2
ssh-copy-id hadoopuser@hadoop-slave4
```

Step 23:

On hadoop-master, open **core-site.xml** file on nano:

```
sudo nano /usr/local/hadoop/etc/hadoop/core-site.xml
```

Then add the following configurations:

```
<configuration>
<property>
<name>fs.defaultFS</name>
<value>hdfs://hadoop-master:9000</value>
</property>
</configuration>
```

Step 24:

Still on hadoop-master, open the **hdfs-site.xml** file.

```
sudo nano /usr/local/hadoop/etc/hadoop/hdfs-site.xml
```

Add the following configurations:

```
<configuration>
<property>
<name>dfs.namenode.name.dir</name><value>/usr/local/hadoop/data/nameNode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name><value>/usr/local/hadoop/data/dataNode</value>
</property>
<property>
<name>dfs.replication</name>
<value>2</value>
</property>
</configuration>
```

Step 25:

We're still on hadoop-master, let's open the **workers** file:

```
sudo nano /usr/local/hadoop/etc/hadoop/workers
```

Add these two lines: (the slave names, remember the hosts file?)

```
hadoop-slave1
hadoop-slave2
hadoop-slave3
```

Step 26:

We need to copy the Hadoop Master configurations to the slaves, to do that we use these commands:

```
scp /usr/local/hadoop/etc/hadoop/* hadoop-slave1:/usr/local/hadoop/etc/hadoop/
scp /usr/local/hadoop/etc/hadoop/* hadoop-slave2:/usr/local/hadoop/etc/hadoop/
scp /usr/local/hadoop/etc/hadoop/* hadoop-slave3:/usr/local/hadoop/etc/hadoop/
```

Step 27:

Now we need to format the HDFS file system. Run these commands hadpoop-master:

```
source /etc/environment
hdfs namenode -format
```

Step 28:

Start HDFS with this command:

```
start-dfs.sh
```

To check if this worked, run the follwing command. This will tell you what resources have been initialized:

```
jps
```

```
Now we need to do the same in the slaves:
```

Step 29:

Let’s see if this worked:

Open your browser and type hadoop-master:9870.

This is what mine shows, hopefully yours is showing the same thing!

```
As you can see, all three nodes are operational!
```

Step 30:

Let’s configure **yarn**, just execute the following commands:

```
export HADOOP_HOME="/usr/local/hadoop"

export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export HADOOP_HDFS_HOME=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_YARN_HOME=$HADOOP_HOME
```

######

```
export HADOOP_OPTS="$HADOOP_OPTS -Djava.library.path=$HADOOP_HOME/lib/native"
```

Step 31:

In both slaves, open **yarn-site.xml** on nano:

```
sudo nano /usr/local/hadoop/etc/hadoop/yarn-site.xml
```

You have to add the following configurations on both slaves:

```
<property>
<name>yarn.resourcemanager.hostname</name>
<value>hadoop-master</value>
</property>
```

Step 32:

On the master, let's start yarn. Use this command:

```
start-yarn.sh
```

Step 33:

Open your browser. Now you will type http://hadoop-master:8088/cluster

As you can see, the cluster shows 3 active nodes!

//

```
sudo add-apt-repository ppa:obsproject/obs-studio

sudo apt update
sudo apt install obs-studio
```

¥¥