

How to install Apache Hadoop 2.6.0 in Ubuntu (Multi node/Cluster setup)

[Home \(https://pingax.com\)](https://pingax.com/) / [How to install Apache Hadoop 2.6.0 in Ubuntu \(Multi node/Cluster setup\)](#)

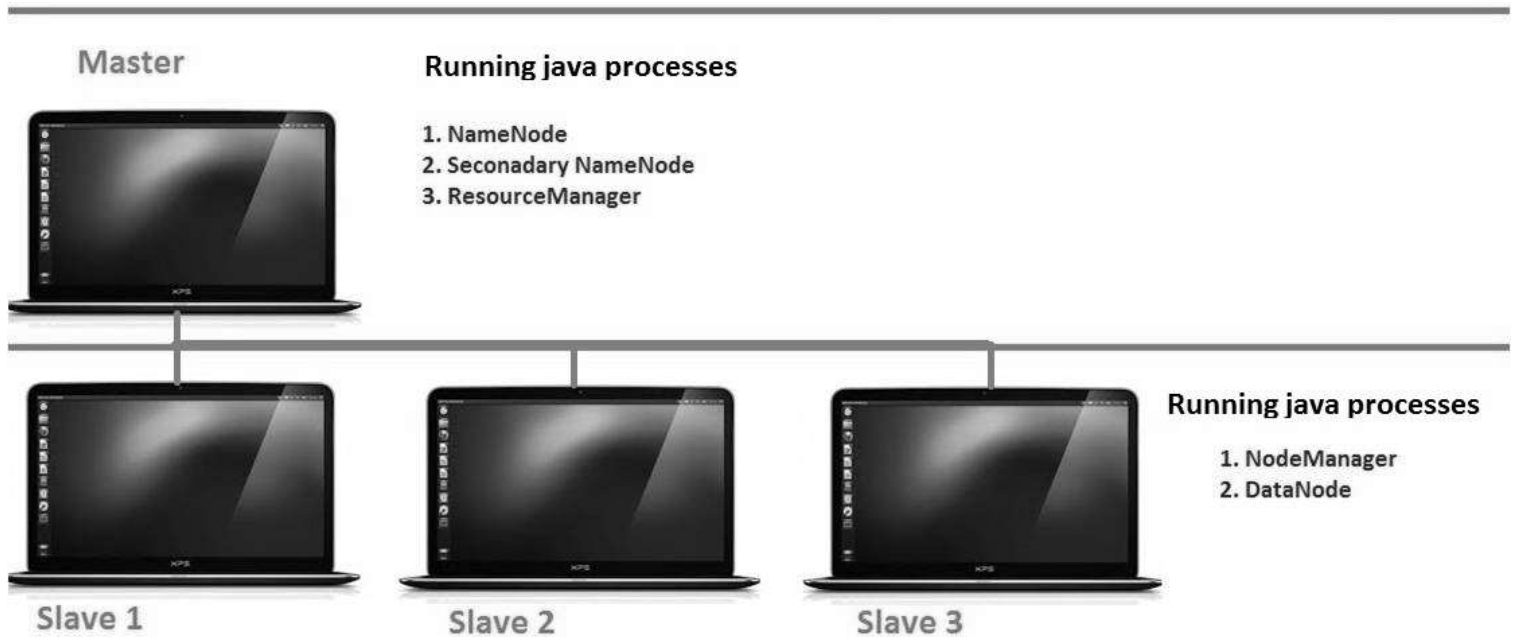
🕒 APR 20, 2015 ([HTTPS://PINGAX.COM/2022/09/08/](https://pingax.com/2022/09/08/)) 👤 VIGNESH PRAJAPATI

📁 [HADOOP \(HTTPS://PINGAX.COM/CATEGORY/HADOOP/\)](https://pingax.com/category/hadoop/)

💬 2,479 COMMENTS ([HTTPS://PINGAX.COM/INSTALL-APACHE-HADOOP-UBUNTU-CLUSTER-SETUP/#COMMENTS](https://pingax.com/install-apache-hadoop-ubuntu-cluster-setup/#comments))

HOW TO INSTALL APACHE HADOOP 2.6.0 IN UBUNTU (MULTI NODE/CLUSTER SETUP)

As you have reached on this blogpost of Setting up Multinode Hadoop cluster, I may believe that you have already read and experimented with my previous blogpost on HOW TO INSTALL APACHE HADOOP 2.6.0 IN UBUNTU (SINGLE NODE SETUP) (<https://pingax.com/install-hadoop2-6-0-on-ubuntu/>). If not then first I would like to recommend you to read it before proceeding here. Since we are interested to setting up Multinode Hadoop cluster, we must have multiple machines to be fit with in Master- Slave architecture.



Here, Multinode Hadoop cluster as composed of Master-Slave Architecture to accomplishment of BigData processing which contains multiple nodes. So, in this post I am going to consider three machines (One as MasterNode and rest two are as SlaveNodes) for setting up Hadoop Cluster. Here there is a correlation between number of computer in cluster with size of data and data processing technique. Hence heavier the dataset (as well as heavier the data processing technique) require larger number of computer/nodes in Hadoop cluster.

Let's get started towards setting up a fresh Multinode Hadoop (2.6.0) cluster. Follow the given steps,

Prerequisites

1. Installation and Configuration of Single node Hadoop :

Install and Configure Single node Hadoop which will be our Master node. To get instructions over How to setup Hadoop Single node, visit – previous blog <https://pingax.com/install-hadoop2-6-0-on-ubuntu/> (<https://pingax.com/install-hadoop2-6-0-on-ubuntu/>).

2. Prepare your computer network (Decide no of nodes to set up cluster) :

Based on the helping parameters like *Purpose of Hadoop Multinode cluster*, *Size of the dataset to be processed* and *Availability of Machines*, you need to define no of Master nodes and no of Slave nodes to be configured for Hadoop Cluster setup.

3. Basic installation and configuration :

Step 3A : Hostname identification of your nodes to be configured in the further steps. To Masternode, we will name it as HadoopMaster and to 2 different Slave nodes, we will name them as HadoopSlave1, HadoopSlave2 respectively in /etc/hosts directory. After deciding a hostname of all nodes, assign their names by updating hostnames (You can ignore this step if you do not want to setup names.) Add all host names to /etc/hosts directory in all Machines (Master and Slave nodes).

```
# Edit the /etc/hosts file with following command
sudo gedit /etc/hosts

# Add following hostname and their ip in host table
192.168.2.14    HadoopMaster
192.168.2.15    HadoopSlave1
192.168.2.16    HadoopSlave2
```

Step 3B : Create hadoop as group and hduser as user in all Machines (if not created !!).

```
vignesh@HadoopMaster:~$ sudo addgroup hadoop
vignesh@HadoopMaster:~$ sudo adduser --ingroup hadoop hduser
```

If you require to add hdusers to sudoers, then fire following command

```
sudo usermod -a -G sudo hduser
```

OR

Add following line in /etc/sudoers/

```
hduser    ALL=(ALL:ALL) ALL
```

Step 3C : Install rsync for sharing hadoop source with rest all Machines,

```
sudo apt-get install rsync
```

Step 3D : To make above changes reflected, we need to reboot all of the Machines.

```
sudo reboot
```

Hadoop configuration steps

1. Applying Common Hadoop Configuration :

However, we will be configuring Master-Slave architecture we need to apply the common changes in Hadoop config files (i.e. common for both type of Mater and Slave nodes) before we distribute these Hadoop files over the rest of the machines/nodes. Hence, these changes will be reflected over your single node Hadoop setup. And from the step 6, we will make changes specifically for Master and Slave nodes respectively.

Changes:

1. Update core-site.xml

Update this file by changing hostname from localhost to HadoopMaster

```
## To edit file, fire the below given command
hduser@HadoopMaster:/usr/local/hadoop/etc/hadoop$ sudo gedit core-site.xml

## Paste these lines into <configuration> tag OR Just update it by replacing localhost w
ith master
<property>
  <name>fs.default.name</name>
  <value>hdfs://HadoopMaster:9000</value>
</property>
```

2. Update hdfs-site.xml

Update this file by updating replication factor from 1 to 3.

```
## To edit file, fire the below given command
hduser@HadoopMaster:/usr/local/hadoop/etc/hadoop$ sudo gedit hdfs-site.xml

## Paste/Update these lines into <configuration> tag
<property>
  <name>dfs.replication</name>
  <value>3</value>
</property>
```

3. Update yarn-site.xml

Update this file by updating the following three properties by updating hostname from localhost to HadoopMaster,

```
## To edit file, fire the below given command
hduser@HadoopMaster:/usr/local/hadoop/etc/hadoop$ sudo gedit yarn-site.xml

## Paste/Update these lines into <configuration> tag
<property>
  <name>yarn.resourcemanager.resource-tracker.address</name>
  <value>HadoopMaster:8025</value>
</property>
<property>
  <name>yarn.resourcemanager.scheduler.address</name>
  <value>HadoopMaster:8035</value>
</property>
<property>
  <name>yarn.resourcemanager.address</name>
  <value>HadoopMaster:8050</value>
</property>
```

4. Update Mapred-site.xml

Update this file by updating and adding following properties,

```
## To edit file, fire the below given command
hduser@HadoopMaster:/usr/local/hadoop/etc/hadoop$ sudo gedit mapred-site.xml

## Paste/Update these lines into <configuration> tag
<property>
    <name>mapreduce.job.tracker</name>
    <value>HadoopMaster:5431</value>
</property>
<property>
    <name>mapred.framework.name</name>
    <value>yarn</value>
</property>
```

5. Update masters

Update the directory of master nodes of Hadoop cluster

```
## To edit file, fire the below given command
hduser@HadoopMaster:/usr/local/hadoop/etc/hadoop$ sudo gedit masters

## Add name of master nodes
HadoopMaster
```

6. Update slaves

Update the directory of slave nodes of Hadoop cluster

```
## To edit file, fire the below given command
hduser@HadoopMaster:/usr/local/hadoop/etc/hadoop$ sudo gedit slaves

## Add name of slave nodes
HadoopSlave1
HadoopSlave2
```

2. Copying/Sharing/Distributing Hadoop config files to rest all nodes – master/slaves

Use rsync for distributing configured Hadoop source among rest of nodes via network.

```
# In HadoopSlave1 machine
sudo rsync -avxP /usr/local/hadoop/ hduser@HadoopSlave1:/usr/local/hadoop/

# In HadoopSlave2 machine
sudo rsync -avxP /usr/local/hadoop/ hduser@HadoopSlave2:/usr/local/hadoop/
```

The above command will share the files stored within hadoop folder to Slave nodes with location – /usr/local/hadoop. So, you don't need to again download as well as setup the above configuration in rest of all nodes. You just need Java and rsync to be installed over all nodes. And this JAVA_HOME path need to be matched with \$HADOOP_HOME/etc/hadoop/hadoop-env.sh file of your Hadoop distribution which we had already configured in Single node Hadoop configuration.

3. Applying Master node specific Hadoop configuration: (Only for master nodes)

These are some configuration to be applied over Hadoop MasterNodes (Since we have only one master node it will be applied to only one master node.)

Step 6A : Remove existing Hadoop_data folder (which was created while single node hadoop setup.)

```
sudo rm -rf /usr/local/hadoop_tmp/
```

Step 6B : Make same (/usr/local/hadoop_tmp/hdfs) directory and create NameNode (/usr/local/hadoop_tmp/hdfs/namenode) directory

```
sudo mkdir -p /usr/local/hadoop_tmp/
sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode
```

Step 6C : Make hduser as owner of that directory.

```
sudo chown hduser:hadoop -R /usr/local/hadoop_tmp/
```

4. Applying Slave node specific Hadoop configuration : (Only for slave nodes)

Since we have three slave nodes, we will be applying the following changes over HadoopSlave1, HadoopSlave2 and HadoopSlave3 nodes.

Step 7A : Remove existing Hadoop_data folder (which was created while single node hadoop setup)

```
sudo rm -rf /usr/local/hadoop_tmp/hdfs/
```

Step 7B : Creates same (/usr/local/hadoop_tmp/) directory/folder,
an inside this folder again Create DataNode (/usr/local/hadoop_tmp/hdfs/namenode) directory/folder

```
sudo mkdir -p /usr/local/hadoop_tmp/  
sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode
```

Step 7C : Make hduser as owner of that directory

```
sudo chown hduser:hadoop -R /usr/local/hadoop_tmp/
```

5. Copying ssh key for Setting up passwordless ssh access from Master to Slave node :

To manage (start/stop) all nodes of Master-Slave architecture, hduser (hadoop user of Master node) need to be login on all Slave as well as all Master nodes which can be possible through setting up passwordless SSH login. (If you are not setting this then you need to provide password while starting and stopping daemons on Slave nodes from Master node).

Fire the following command for sharing public SSH key – \$HOME/.ssh/id_rsa.pub file (of HadoopMaster node) to authorized_keys file of hduser@HadoopSlave1 and also on hduser@HadoopSlave1 (in \$HOME/.ssh/authorized_keys)

```
hduser@HadoopMaster: ~$ ssh-copy-id -i $HOME/.ssh/id_rsa.pub hduser@HadoopSlave1  
hduser@HadoopMaster: ~$ ssh-copy-id -i $HOME/.ssh/id_rsa.pub hduser@HadoopSlave2
```

6. Format Namenode (Run on MasterNode) :


```
# Run this command from Masternode
hduser@HadoopMaster: /usr/local/hadoop$ hdfs namenode -format
```

7. Starting up Hadoop cluster daemons : (Run on MasterNode)

Start HDFS daemons:

```
hduser@HadoopMaster: /usr/local/hadoop$ start-dfs.sh
```

Start MapReduce daemons:

```
hduser@HadoopMaster: /usr/local/hadoop$ start-yarn.sh
```

Instead both of these above command you can also use start-all.sh, but its now deprecated so its not recommended to be used for better Hadoop operations.

8. Track/Monitor/Verify Hadoop cluster : (Run on any Node)

Verify Hadoop daemons on Master :

```
hduser@HadoopMaster: jps
```



([https://pingax.com/wp-](https://pingax.com/wp-content/uploads/2015/04/master.png)

[content/uploads/2015/04/master.png](https://pingax.com/wp-content/uploads/2015/04/master.png))

Verify Hadoop daemons on all slave nodes :

```
hduser@HadoopSlave1: jps
```

```
hduser@HadoopSlave2: jps
```

(<https://pingax.com/wp-content/uploads/2015/04/jpsSLave11.png>)

(As shown in above snap- The running services of HadoopSlave1 will be the same for all Slave nodes configured in Hadoop Cluster.)

Monitor Hadoop ResourceManage and Hadoop NameNode via web-version,

If you wish to track Hadoop MapReduce as well as HDFS, you can also try exploring Hadoop web view of ResourceManager and NameNode which are usually used by hadoop administrators. Open your default browser and visit to the following links from any of the node.

For ResourceManager – [Http://HadoopMaster:8088](http://HadoopMaster:8088) (<http://HadoopMaster:8088>)

(<https://pingax.com/wp-content/uploads/2015/04/master80881.png>)

For NameNode – [Http://HadoopMaster:50070](http://HadoopMaster:50070) (<http://HadoopMaster:50070>)

(<https://pingax.com/wp-content/uploads/2015/04/master500701.png>)

If you are getting the similar output as shown in the above snapshot for Master and Slave nodes then Congratulations! You have successfully installed Apache Hadoop in your Cluster and if not then post your error messages in comments. We will be happy to help you. Happy Hadooping!! Also you can request me (vignesh@pingax.com) for blog title if you want me to write over it.



(</install-apache-hadoop-ubuntu-cluster-setup/?format=pdf>)

Google+ Comments

Powered by Google+ Comments (<http://3doordigital.com/wordpress/plugins/google-plus-comments/>)


2,479 comments to “How to install Apache Hadoop 2.6.0 in Ubuntu (Multi node/Cluster setup)”

You can leave a reply or Trackback (<https://pingax.com/install-apache-hadoop-ubuntu-cluster-setup/trackback/>) this post.

EMAIL SUBSCRIPTION

Your email here

Subscribe

 [Posts RSS \(Really Simple Syndication\) \(https://pingax.com/feed/\)](https://pingax.com/feed/)

 [Comments RSS \(Really Simple Syndication\) \(https://pingax.com/comments/feed/\)](https://pingax.com/comments/feed/)

Download
The
Whitepaper

CrowdStrike®

Open





Learn data science by doing



Learn More

(<https://datacamp.pxf.io/c/1170627/1091987/13294>)



NATURE
RESEARCH
EDITING

Get 10% Off Your First Order

A Nature Research
service. Expert scientific
& english editing.

Open

RECENT POSTS

SparkR with Rstudio in Ubuntu 12.04 (<https://pingax.com/sparkr-with-rstudio-ubuntu-12-04/>)

How to install Apache Hadoop 2.6.0 in Ubuntu (Multi node/Cluster setup) (<https://pingax.com/install-apache-hadoop-ubuntu-cluster-setup/>)

Predictive analysis in eCommerce part-3 (<https://pingax.com/predictive-analysis-ecommerce-part-3/>)

How to install Apache Hadoop 2.6.0 in Ubuntu (Single node setup) (<https://pingax.com/install-hadoop2-6-0-on-ubuntu/>)

Build Predictive Model on Big data: Using R and MySQL Part-3 (<https://pingax.com/build-predictive-model-big-data-using-r-mysql-part-3/>)

CATEGORIES

Hadoop (<https://pingax.com/category/hadoop/>)

Machine Learning (<https://pingax.com/category/machine-learning/>)

MongoDB (<https://pingax.com/category/mongodb/>)

R (<https://pingax.com/category/r/>)

Spark (<https://pingax.com/category/spark/>)

