**Introduction**

Every major industry is implementing [Apache Hadoop](https://phoenixnap.com/kb/apache-hadoop-architecture-explained) as the standard framework for processing and storing big data. Hadoop is designed to be deployed across a network of hundreds or even thousands of [dedicated servers](https://phoenixnap.com/servers/dedicated). All these machines work together to deal with the massive volume and variety of incoming datasets.

Deploying Hadoop services on a single node is a great way to get yourself acquainted with basic Hadoop commands and concepts.

**This easy-to-follow guide helps you install Hadoop on Ubuntu 18.04 or Ubuntu 20.04.**

**Prerequisites**

* Access to a terminal window/command line
* **Sudo** or **root** privileges on local /remote machines

**Install OpenJDK on Ubuntu**

The [Hadoop framework](https://phoenixnap.com/blog/what-is-hadoop) is written in Java, and its services require a compatible Java Runtime Environment (JRE) and Java Development Kit (JDK). Use the following command to update your system before initiating a new installation:

sudo apt update

At the moment, **Apache Hadoop 3.x fully supports Java 8**. The OpenJDK 8 package in Ubuntu contains both the runtime environment and development kit.

Type the following command in your terminal to install OpenJDK 8:

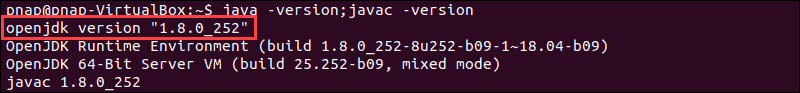
sudo apt install openjdk-8-jdk -y

The OpenJDK or Oracle Java version can affect how elements of a Hadoop ecosystem interact. To install a specific Java version, check out our detailed guide on [how to install Java on Ubuntu](https://phoenixnap.com/kb/how-to-install-java-ubuntu).

Once the installation process is complete, [verify the current Java version](https://phoenixnap.com/kb/check-java-version-linux):

java -version; javac -version

The output informs you which Java edition is in use.



## Set Up a Non-Root User for Hadoop Environment

It is advisable to create a non-root user, specifically for the Hadoop environment. A distinct user improves security and helps you manage your cluster more efficiently. To ensure the smooth functioning of Hadoop services, the user should have the ability to establish a [passwordless SSH connection](https://phoenixnap.com/kb/setup-passwordless-ssh) with the localhost.

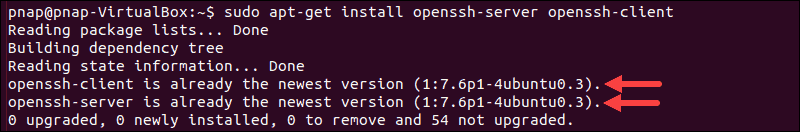
### Install OpenSSH on Ubuntu

Install the OpenSSH server and client using the following command:

sudo apt install openssh-server openssh-client -y

sudo apt-get install vsftpd

In the example below, the output confirms that the latest version is already installed.



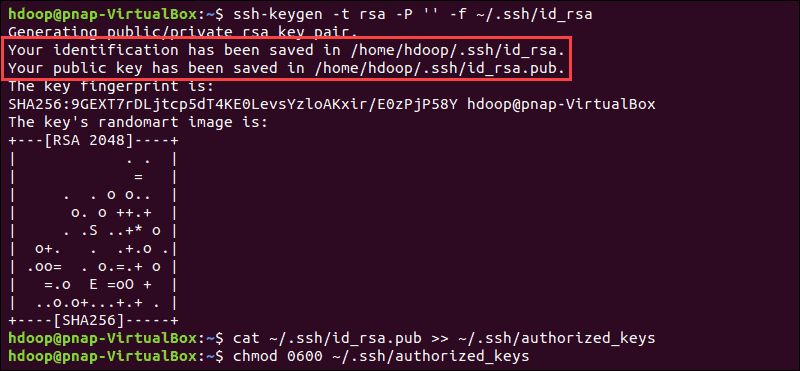
If you have installed OpenSSH for the first time, use this opportunity to implement these vital [SSH security recommendations.](https://phoenixnap.com/kb/linux-ssh-security)

### Enable Passwordless SSH for Hadoop User

[Generate an SSH key pair](https://phoenixnap.com/kb/generate-setup-ssh-key-ubuntu)and define the location is is to be stored in:

ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa

The system proceeds to generate and save the SSH key pair.



Use the **cat** command to store the public key as **authorized\_keys**in the ssh directory:

cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

Set the permissions for your user with the **chmod** command:

chmod 0600 ~/.ssh/authorized\_keys

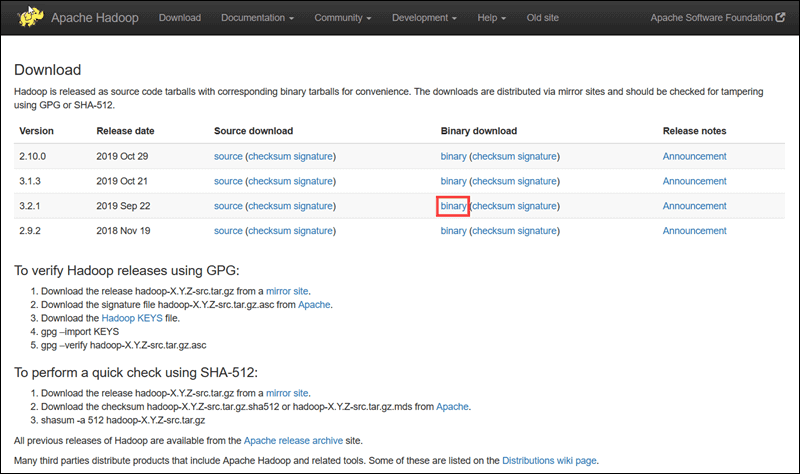
The new user is now able to SSH without needing to enter a password every time. Verify everything is set up correctly by using the **hdoop** user to SSH to localhost:

ssh localhost

After an initial prompt, the Hadoop user is now able to establish an SSH connection to the localhost seamlessly.

## Download and Install Hadoop on Ubuntu

Visit the [official Apache Hadoop project page](https://hadoop.apache.org/releases.html), and select the version of Hadoop you want to implement.



The steps outlined in this tutorial use the Binary download for **Hadoop Version 3.3.4**.

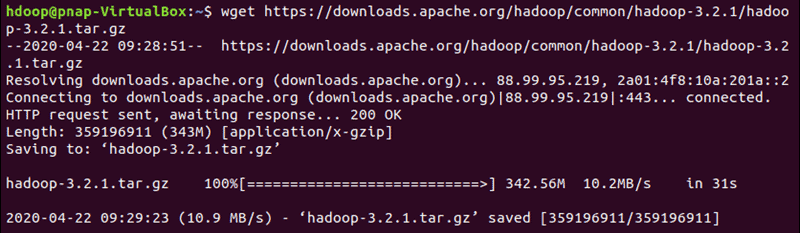
Select your preferred option, and you are presented with a mirror link that allows you to download the **Hadoop tar package**.



**Note:** It is sound practice to verify Hadoop downloads originating from mirror sites. The instructions for using GPG or SHA-512 for verification are provided on the official download page.

Use the provided mirror link and download the Hadoop package with the **wget** command:

wget https://downloads.apache.org/hadoop/common/hadoop-3.3.4/hadoop-3.3.4.tar.gz



Once the download is complete, extract the files to initiate the Hadoop installation:

tar xzf hadoop-3.3.4.tar.gz

The Hadoop binary files are now located within the hadoop-3.3.4 directory.

## Single Node Hadoop Deployment (Pseudo-Distributed Mode)

Hadoop excels when deployed in a **fully distributed mode** on a large cluster of networked servers. However, if you are new to Hadoop and want to explore basic commands or test applications, you can configure Hadoop on a single node.

This setup, also called **pseudo-distributed mode**, allows each Hadoop daemon to run as a single Java process. A Hadoop environment is configured by editing a set of configuration files:

* bashrc
* hadoop-env.sh
* core-site.xml
* hdfs-site.xml
* mapred-site-xml
* yarn-site.xml

### Configure Hadoop Environment Variables (bashrc)

Edit the .bashrc shell configuration file using a text editor of your choice (we will be using nano):

sudo nano .bashrc

Define the Hadoop environment variables by adding the following content to the end of the file:

#Hadoop Related Options

export HADOOP\_HOME=/home/hdoop/hadoop-3.3.4 – should be your user

export HADOOP\_INSTALL=$HADOOP\_HOME

export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_HOME=$HADOOP\_HOME

export HADOOP\_HDFS\_HOME=$HADOOP\_HOME

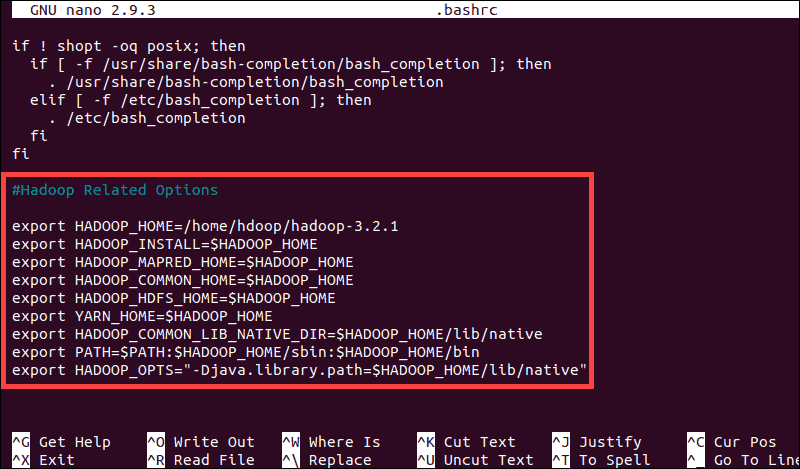
export YARN\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native

export PATH=$PATH:$HADOOP\_HOME/sbin:$HADOOP\_HOME/bin

export HADOOP\_OPTS="-Djava.library.path=$HADOOP\_HOME/lib/native"

Once you add the variables, save and exit the .bashrc file.



It is vital to apply the changes to the current running environment by using the following command:

source ~/.bashrc

### Edit hadoop-env.sh File

The hadoop-env.sh file serves as a master file to configure YARN, [HDFS](https://phoenixnap.com/kb/what-is-hdfs), [MapReduce](https://phoenixnap.com/kb/hadoop-mapreduce), and Hadoop-related project settings.

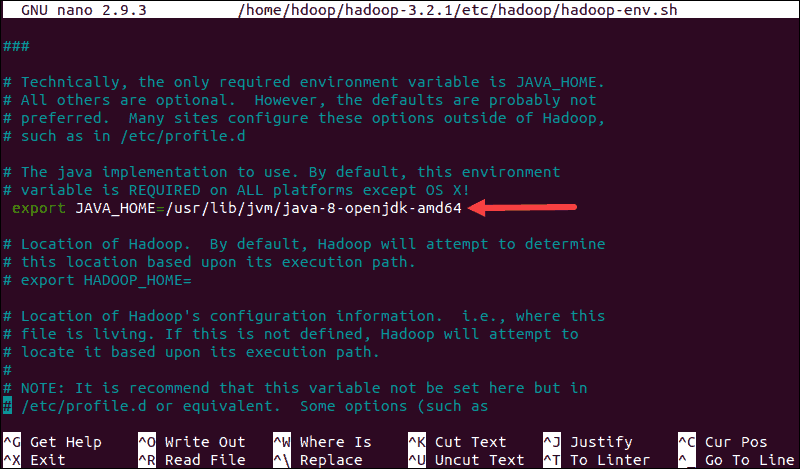
When setting up a **single node Hadoop cluster**, you need to define which Java implementation is to be utilized. Use the previously created **$HADOOP\_HOME** variable to access the hadoop-env.sh file:

sudo nano $HADOOP\_HOME/etc/hadoop/hadoop-env.sh

Uncomment the **$JAVA\_HOME** variable (i.e., remove the **#** sign) and add the full path to the OpenJDK installation on your system. If you have installed the same version as presented in the first part of this tutorial, add the following line:

export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

The path needs to match the location of the Java installation on your system.



If you need help to locate the correct Java path, run the following command in your terminal window:

which javac

The resulting output provides the path to the Java binary directory.

Java binary directory location with which command.

Use the provided path to find the OpenJDK directory with the following command:

readlink -f /usr/bin/javac

The section of the path just before the /bin/javac directory needs to be assigned to the **$JAVA\_HOME** variable.

The location of the openjdk binary directory.

### Edit core-site.xml File

The core-site.xml file defines HDFS and Hadoop core properties.

To set up Hadoop in a pseudo-distributed mode, you need to **specify the URL** for your NameNode, and the temporary directory Hadoop uses for the map and reduce process.

Open the core-site.xml file in a text editor:

sudo nano $HADOOP\_HOME/etc/hadoop/core-site.xml

Add the following configuration to override the default values for the temporary directory and add your HDFS URL to replace the default local file system setting:

<configuration>

<property>

<name>hadoop.tmp.dir</name>

<value>/home/hdoop/tmpdata</value> should be your user

</property>

<property>

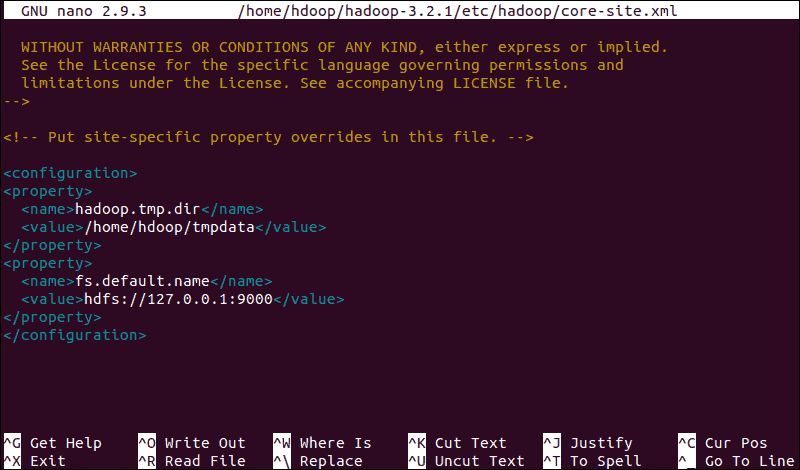
<name>fs.default.name</name>

<value>hdfs://127.0.0.1:9000</value>

</property>

</configuration>

This example uses values specific to the local system. You should use values that match your systems requirements. The data needs to be consistent throughout the configuration process.



Do not forget to [create a Linux directory](https://phoenixnap.com/kb/create-directory-linux-mkdir-command) in the location you specified for your temporary data.

### Edit hdfs-site.xml File

The properties in the hdfs-site.xml file govern the location for storing node metadata, fsimage file, and edit log file. Configure the file by defining the **NameNode**and**DataNode storage directories**.

Additionally, the default **dfs.replication** value of **3** needs to be changed to **1** to match the single node setup.

Use the following command to open the hdfs-site.xml file for editing:

sudo nano $HADOOP\_HOME/etc/hadoop/hdfs-site.xml

Add the following configuration to the file and, if needed, adjust the NameNode and DataNode directories to your custom locations:

<configuration>

<property>

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

<value>/home/hdoop/dfsdata/namenode</value> should be your user

</property>

<property>

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

<value>/home/hdoop/dfsdata/datanode</value> should be your user

</property>

<property>

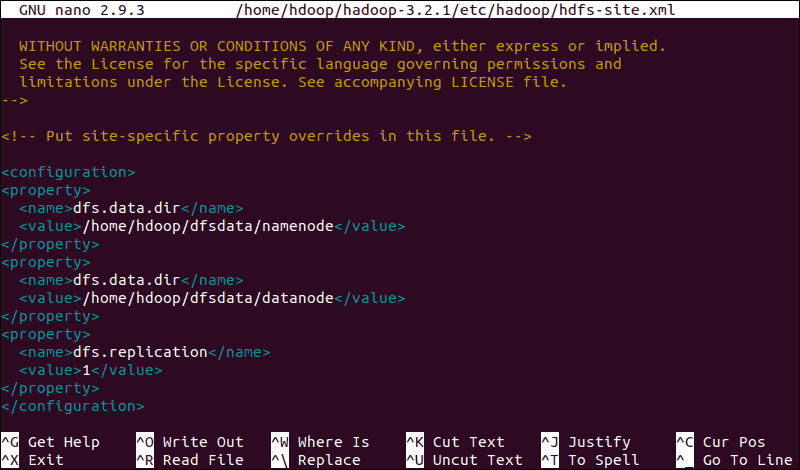
<name>dfs.replication</name>

<value>1</value>

</property>

</configuration>

If necessary, create the specific directories you defined for the **dfs.data.dir** value.



### Edit mapred-site.xml File

Use the following command to access the mapred-site.xml file and **define MapReduce values**:

sudo nano $HADOOP\_HOME/etc/hadoop/mapred-site.xml

Add the following configuration to change the default MapReduce framework name value to **yarn**:

<configuration>

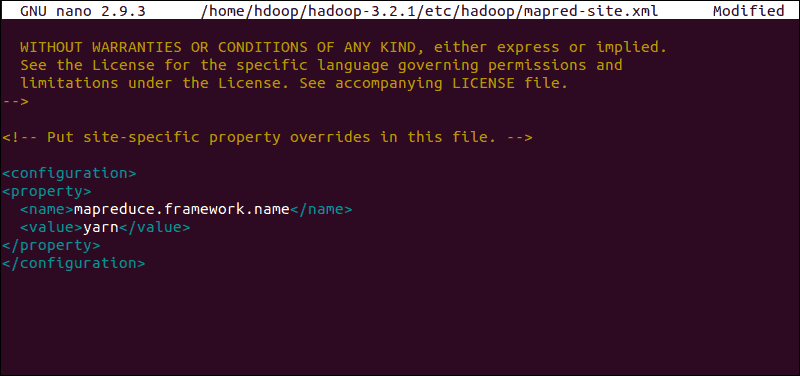
<property>

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

<value>yarn</value>

</property>

</configuration>



### Edit yarn-site.xml File

The yarn-site.xml file is used to define settings relevant to **YARN**. It contains configurations for the **Node Manager, Resource Manager, Containers,** and **Application Master**.

Open the yarn-site.xml file in a text editor:

sudo nano $HADOOP\_HOME/etc/hadoop/yarn-site.xml

Append the following configuration to the file:

<configuration>

<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.resourcemanager.hostname</name>

<value>127.0.0.1</value>

</property>

<property>

<name>yarn.acl.enable</name>

<value>0</value>

</property>

<property>

<name>yarn.nodemanager.env-whitelist</name>

<value>JAVA\_HOME,HADOOP\_COMMON\_HOME,HADOOP\_HDFS\_HOME,HADOOP\_CONF\_DIR,CLASSPATH\_PERPEND\_DISTCACHE,HADOOP\_YARN\_HOME,HADOOP\_MAPRED\_HOME</value>

</property>

</configuration>

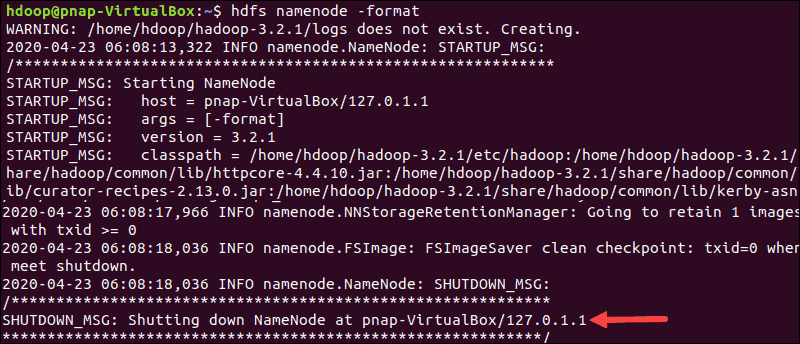


## Format HDFS NameNode

It is important to **format the NameNode** before starting Hadoop services for the first time:

hdfs namenode -format

The shutdown notification signifies the end of the NameNode format process.

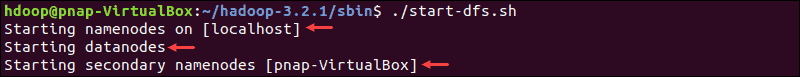


## Start Hadoop Cluster

Navigate to the hadoop-3.3.4/sbin directory and execute the following commands to start the NameNode and DataNode:

./start-dfs.sh

The system takes a few moments to initiate the necessary nodes.



Once the namenode, datanodes, and secondary namenode are up and running, start the YARN resource and nodemanagers by typing:

./start-yarn.sh

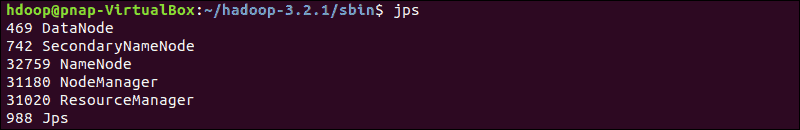
As with the previous command, the output informs you that the processes are starting.

The system initiates the resource and node manager daemons.

Type this simple command to check if all the daemons are active and running as Java processes:

jps

If everything is working as intended, the resulting list of running Java processes contains all the HDFS and YARN daemons.

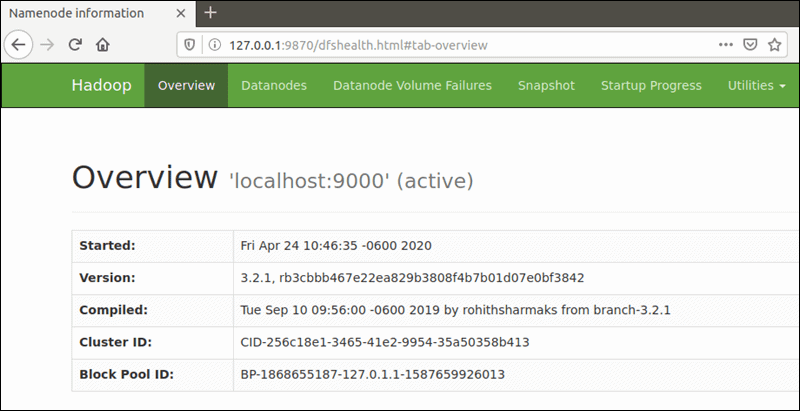


## Access Hadoop UI from Browser

[Use your preferred browser](https://phoenixnap.com/blog/best-google-chrome-security-extensions) and navigate to your localhost URL or IP. The default port number **9870** gives you access to the Hadoop NameNode UI:

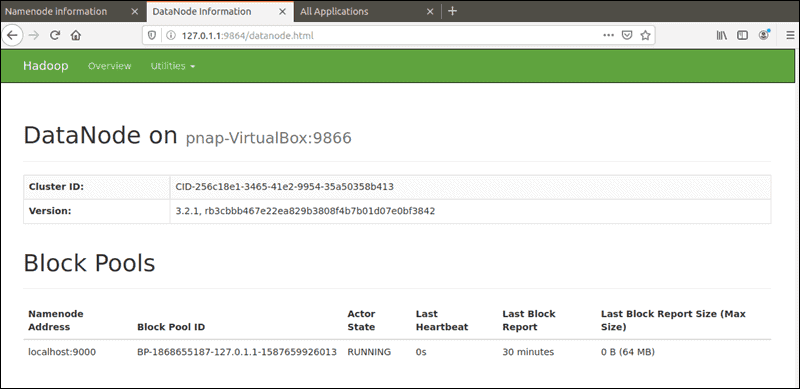
http://localhost:9870

The NameNode user interface provides a comprehensive overview of the entire cluster.



The default port **9864** is used to access individual DataNodes directly from your browser:

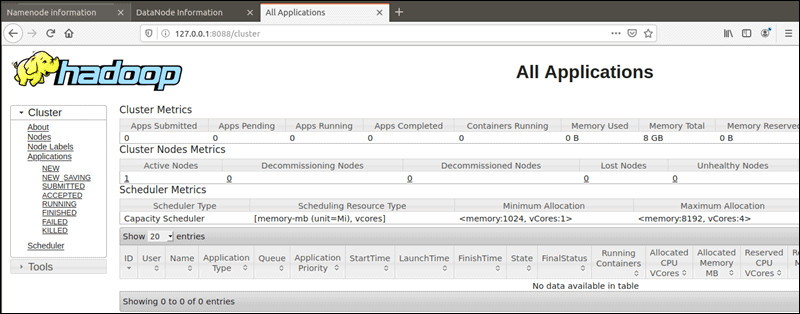
http://localhost:9864



The YARN Resource Manager is accessible on port **8088**:

http://localhost:8088

The Resource Manager is an invaluable tool that allows you to monitor all running processes in your Hadoop cluster.



**Conclusion**

You have successfully installed Hadoop on Ubuntu and deployed it in a pseudo-distributed mode. A single node Hadoop deployment is an excellent starting point to explore basic HDFS commands and acquire the experience you need to design a fully distributed Hadoop cluster

Finally all nodes should be stopped...so use command

./stop-all.sh