

Lets start with Installation first

Prerequisites :

1. Java

2. An Ubuntu 18.04 server with a non-root user with sudo privileges

**STEP 1:** To get started, we'll update our package list:

**sudo apt-get update**

**STEP 2:** Next, install OpenJDK, the default Java Development Kit on Ubuntu 18.04.

**sudo apt-get install default-jdk**

Once the installation is complete, let's check the version.

**java -version**

**STEP 3:** Installing Hadoop

With Java in place, we'll visit the [Apache Hadoop Releases page](http://hadoop.apache.org/releases.html) to find the most recent stable release.

<http://hadoop.apache.org/releases.html>

After visiting the above page, you'll find Apache Hadoop Releases. Click on Binary given under Tarball option against 2.8.1

[Top](#)[Wiki](#)

#### About

[Welcome](#)[Releases](#)[Download](#)[Release Notes](#)[Release Versioning](#)[Mailing Lists](#)[Issue Tracking](#)[Who We Are?](#)[Who Uses Hadoop?](#)[Buy Stuff](#)[Sponsorship](#)[Thanks](#)[Privacy Policy](#)[Bylaws](#)[Committer criteria](#)[License](#)[Documentation](#)[Related Projects](#)

## Apache Hadoop Releases

### Download

Hadoop is released as source code tarballs with corresponding binary tarballs for convenience. The downloads are tampered using GPG or SHA-256.

Version	Release Date	Tarball	GPG
<a href="#">3.0.0-alpha4</a>	07 July, 2017	<a href="#">source</a>	<a href="#">signature</a>
		<a href="#">binary</a>	<a href="#">signature</a>
<a href="#">2.8.1</a>	08 June, 2017	<a href="#">source</a>	<a href="#">signature</a>
		<a href="#">binary</a>	<a href="#">signature</a>
<a href="#">2.7.4</a>	04 August, 2017	<a href="#">source</a>	<a href="#">signature</a>
		<a href="#">binary</a>	<a href="#">signature</a>
<a href="#">2.6.5</a>	08 October, 2016	<a href="#">source</a>	<a href="#">signature</a>
		<a href="#">binary</a>	<a href="#">signature</a>

To verify Hadoop releases using GPG:

1. Download the release `hadoop-X.Y.Z-src.tar.gz` from a [mirror site](#).
2. Download the signature file `hadoop-X.Y.Z-src.tar.gz.asc` from [Apache](#).
3. Download the [Hadoop KEYS](#) file.
4. `gpg --import KEYS`
5. `gpg --verify hadoop-X.Y.Z-src.tar.gz.asc`

When you click on Binary, it will take you to next page.



We suggest the following mirror site for your download:

<http://www-us.apache.org/dist/hadoop/common/hadoop-2.8.1/hadoop-2.8.1.tar.gz>

Other mirror sites are suggested below. Please use the backup mirrors only to download PGP and MD5 signatures working.

## HTTP

<http://www-eu.apache.org/dist/hadoop/common/hadoop-2.8.1/hadoop-2.8.1.tar.gz>

<http://www-us.apache.org/dist/hadoop/common/hadoop-2.8.1/hadoop-2.8.1.tar.gz>

Circled link is desired download link for Hadoop.

STEP 4: Go to Terminal again and write the following command to start downloading Hadoop.

[REPORT THIS AD](#)

```
wget http://www-us.apache.org/dist/hadoop/common/hadoop-2.8.1/hadoop-2.8.1.tar.gz
```

It will take some time depending upon your internet connection speed.

**STEP 5: (OPTIONAL STEP)** After you download the above software, you can check its integrity that whether the downloaded software is modified or not.

On Apache Hadoop release page, you will find checksum link.

Version	Release Date	Tarball	GPG
<a href="#">3.0.0-alpha4</a>	07 July, 2017	<a href="#">source</a>	<a href="#">signature</a>
		<a href="#">binary</a>	<a href="#">signature</a>
<a href="#">2.8.1</a>	08 June, 2017	<a href="#">source</a>	<a href="#">signature</a>
		<a href="#">binary</a>	<a href="#">signature</a>
<a href="#">2.7.4</a>	04 August, 2017	<a href="#">source</a>	<a href="#">signature</a>
		<a href="#">binary</a>	<a href="#">signature</a>
<a href="#">2.6.5</a>	08 October, 2016	<a href="#">source</a>	<a href="#">signature</a>
		<a href="#">binary</a>	<a href="#">signature</a>

To verify Hadoop releases using GPG:

1. Download the release `hadoop-X.Y.Z-src.tar.gz` from a [mirror site](#).
2. Download the signature file `hadoop-X.Y.Z-src.tar.gz.asc` from [Apache](#).
3. Download the [Hadoop KEYS](#) file.
4. `gpg --import KEYS`
5. `gpg --verify hadoop-X.Y.Z-src.tar.gz.asc`

To perform a quick check using SHA-256:

1. Download the release `hadoop-X.Y.Z-src.tar.gz` from a [mirror site](#).
2. Download the checksum `hadoop-X.Y.Z-src.tar.gz.md5` from [Apache](#).
3. `shasum -a 256 hadoop-X.Y.Z-src.tar.gz`

All previous releases of Hadoop are available from the [Apache release archive](#) site.

Many third parties distribute products that include Apache Hadoop and related tools. Some of these are listed

## Release Notes

### 04 August, 2017: Release 2.7.4 available

This is the next release of Apache Hadoop 2.7 line.

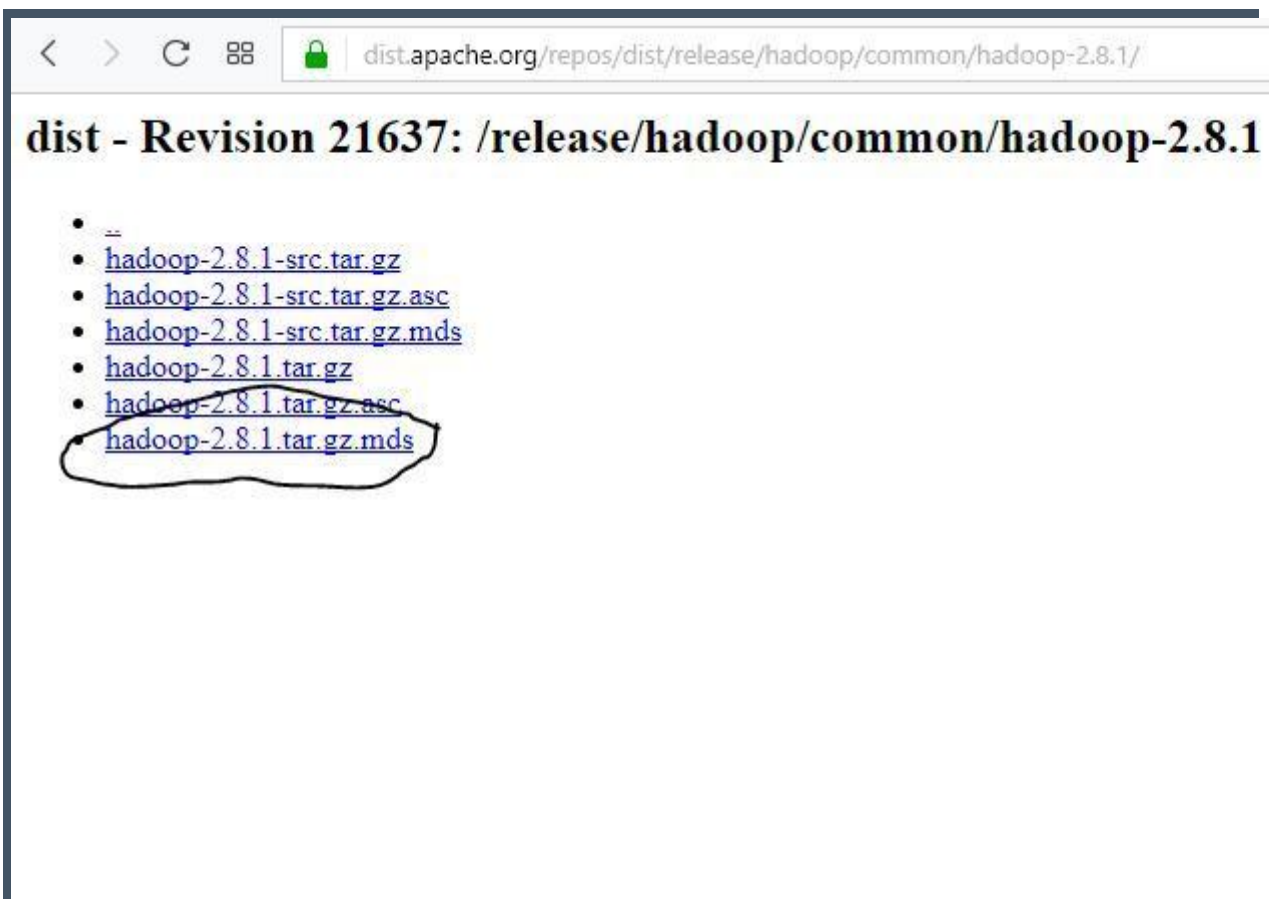
Please see the [Hadoop 2.7.4 Release Notes](#) for the list of 264 bugs fixes and optimizations since the previous r

In above image, link is encircled. When you click on this link, it will take to following page.

## dist - Revision 21637: /release/hadoop/common

- ..
- [KEYS](#)
- [current](#)
- [hadoop-1.2.1/](#)
- [hadoop-2.6.5/](#)
- [hadoop-2.7.1/](#)
- [hadoop-2.8.1/](#)
- [hadoop-3.0.0-alpha4/](#)
- [readme.txt](#)
- [stable](#)
- [stable1](#)
- [stable2](#)

After clicking the encircled link, it will further take you to following page :



Encircled link in above image is desired checksum file. Now will download this file by writing following command on terminal.

```
wget https://dist.apache.org/repos/dist/release/hadoop/common/hadoop-2.8.1/hadoop-2.8.1.tar.gz.mds
```

Then run the verification by executing following command on terminal :

```
shasum -a 256 hadoop-2.8.1.tar.gz
```

One will expect following output



Output

```
d489df3808244b906eb38f4d081ba49e50c4603db03efd5e594  
a1e98b09259c2 hadoop-2.8.1.tar.gz
```

Compare this value with the SHA-256 value in the .md5 file

Run following command on terminal :

```
cat hadoop-2.8.1.tar.gz.md5
```

```
hadoop-2.8.1.tar.gz: SHA256 = D489DF38 08244B90  
6EB38F4D 081BA49E 50C4603D B03EFD5E 594A1E98  
B09259C2
```

The output of the above command should match the value in the file we downloaded from apache.org.

**STEP 6 :** Now that we've verified that the file wasn't corrupted or changed, we'll use the tar command with the -x flag to extract, -z to uncompress, -v for verbose output, and -f to specify that we're extracting from a file.

```
tar -xzvf hadoop-2.8.1.tar.gz
```

Finally, we'll move the extracted files into /usr/local, the appropriate place for locally installed software.

```
sudo mv hadoop-2.8.1 /usr/local/Hadoop
```

**STEP 7:** Configuring JAVA HOME

To find the default Java path, make use of following command :

```
readlink -f /usr/bin/java | sed "s:bin/java::"
```

Output :

**/usr/lib/jvm/java-8-openjdk-amd64/jre/**

You can copy this output to set Hadoop's Java home to this specific version, which ensures that if the default Java changes, this value will not.

To begin, open `hadoop-env.sh`:

**sudo nano /usr/local/hadoop/etc/hadoop/hadoop-env.sh**

Edit the file with following modifications

```
#export JAVA_HOME=${JAVA_HOME}
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-
amd64/jre/
```

Put hashtag against default `JAVA_HOME` and add new `JAVA_HOME`.

**STEP 8 :** Check whether HADOOP got installed

Run following command to check.

**/usr/local/hadoop-3.1.2/bin/hadoop version**

If you get following output, it means everything is in right place.

Usage: `hadoop [-config confdir] [COMMAND | CLASSNAME]`  
`CLASSNAME` run the class named `CLASSNAME`

.....

**STEP 9: CONFIGURING HADOOP**



## INSTALLING SSH

For installing SSH you need to run following command :

```
apt-get install ssh
```

It will take sometime to install ssh. Once during installation user will be asked to continue. Press Y to proceed further.

User may face following error :

**Could not get lock /var/lib/dpkg/lock – Unable to lock the directory**

Error can be removed by running following commands :

```
sudo rm /var/lib/dpkg/lock
```

```
sudo apt-get update
```

After running these commands again try installing ssh.

**STEP 10:** Generate Public/Private RSA key pair by running following command

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

When you have been asked 'Enter file in which to save the key' during installation of key then simply press ENTER.

```

Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
23:42:08:de:26:bc:6e:4c:3e:3e:55:f7:a6:4d:8f:bb root@tutorials
The key's randomart image is:
+--[ RSA 2048 ]-----+
| .                      |
| o...                   |
| +.o.                   |
|  +. . .                |
| o  o o S               |
| = . . . =              |
| *.      = o            |
| o..      . o .         |
| ..      Eo             |
+-----+

```

## STEP 11: Making the generated public key authorized

You can make the generated public key authorized by running following command:

```
cat /home/hadoop/.ssh/id_rsa.pub >>
/home/hadoop/.ssh/authorized_keys
```

## STEP 12: Check if SSH working by running following command

[REPORT THIS AD](#)

## ssh localhost

During process, user will be asked to continue then simply type 'yes'.

You will get following message after installation

Welcome to Ubuntu 18.04 .....

.....

### **STEP 13:** Edit and Setup Configuration Files

To complete the setup of Hadoop, the following files will have to be modified:

```
~/bashrc
/usr/local/hadoop/etc/hadoop/hadoop-env.sh
/usr/local/hadoop/etc/hadoop/core-site.xml
/usr/local/hadoop/etc/hadoop/yarn-site.xml
/usr/local/hadoop/etc/hadoop/mapred-site.xml.template
/usr/local/hadoop/etc/hadoop/hdfs-site.xml
```

### **STEP 14:** Editing ~/bashrc

Open up ~/bashrc by using nano editor

**sudo nano ~/bashrc**

Go to the end of the file and paste/type the following content in it:

```
#HADOOP VARIABLES START
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-
amd64/jre/
export HADOOP_INSTALL=/usr/local/hadoop
export PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOP_INSTALL/sbin
export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_HOME=$HADOOP_INSTALL
export HADOOP_HDFS_HOME=$HADOOP_INSTALL
export YARN_HOME=$HADOOP_INSTALL
export
HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_INSTA
```

```
LL/lib/native
export HADOOP_OPTS="-
Djava.library.path=$HADOOP_INSTALL/lib"
#HADOOP VARIABLES END
```

Files opened and edited using nano can be saved using Ctrl + X. Upon the prompt to save changes, type Y. If you are asked for a filename, just press the enter key.

After saving and closing the .bashrc file, execute the following command so that your system recognizes the newly created environment variables:

```
source ~/.bashrc
```

### **STEP 15: Editing /usr/local/hadoop/etc/hadoop/hadoop-env.sh**

Already configured in Step 7.

### **STEP 16: Editing /usr/local/hadoop/etc/hadoop/core-site.xml**

The /usr/local/hadoop/etc/hadoop/core-site.xml file contains configuration properties that Hadoop uses when starting up. This file can be used to override the default settings that Hadoop starts with.

Open this file with nano using the following command:

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

In this file, enter the following content in between the <configuration></configuration> tag:

```
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
```

Save and close this file.

## **STEP 17: Editing /usr/local/hadoop/etc/hadoop/yarn-site.xml**

The /usr/local/hadoop/etc/hadoop/yarn-site.xml file contains configuration properties that MapReduce uses when starting up. This file can be used to override the default settings that MapReduce starts with.

Open this file with nano using the following command:

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

In this file, enter the following content in between the <configuration></configuration> tag:

```
<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</valu
e>
</property>
```

Save and close this file.

## **STEP 18: Creating and Editing /usr/local/hadoop/etc/hadoop/mapred-site.xml**

By default, the /usr/local/hadoop/etc/hadoop/ folder contains the /usr/local/hadoop/etc/hadoop/mapred-site.xml.template file which has to be renamed/copied with the name mapred-site.xml. This file is used to specify which framework is being used for MapReduce.

This can be done using the following command:

```
cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template /usr/local/hadoop/etc/hadoop/mapred-site.xml
```

(Don't think the above code is two step code, lower line is in continuation with first)

Once this is done, open the newly created file with nano using the following command:

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

In this file, enter the following content in between the <configuration></configuration> tag:

```
<property>
```

```
<name>mapred.job.tracker</name>
```

```
<value>localhost:54311</value>
```

```
<description>The host and port that the MapReduce job tracker runs
```

at. If "local", then jobs are run in-process as a single map

and reduce task.

```
</description>
```

```
</property>
```

Save and close this file.

## **STEP 19: Editing /usr/local/hadoop/etc/hadoop/hdfs-site.xml**

The /usr/local/hadoop/etc/hadoop/hdfs-site.xml has to be configured for each host in the cluster that is being used. It is used to specify the directories which will be used as the namenode and the datanode on that host.

Before editing this file, we need to create two directories which will contain the namenode and the datanode for this Hadoop installation.

This can be done using the following commands:

```
mkdir -p /usr/local/hadoop_store/hdfs/namenode  
mkdir -p /usr/local/hadoop_store/hdfs/datanode
```

Once this is done, open the /usr/local/hadoop/etc/hadoop/hdfs-site.xml file with nano using the following command:

```
sudo nano /usr/local/hadoop/etc/hadoop/hdfs-site.xml  
In this file, enter the following content in between the  
<configuration></configuration> tag:
```



```
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>file:/usr/local/hadoop_store/hdfs/namenode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>file:/usr/local/hadoop_store/hdfs/datanode</value>
</property>
```

Save and close this file.

## **STEP 20 : Format the New Hadoop Filesystem**

After completing all the configuration outlined in the above steps, the Hadoop filesystem needs to be formatted so that it can start being used.

This is done by executing the following command:

```
hdfs namenode -format
```

## **STEP 21: (FINAL STEP) STARTING HADOOP**

All that remains to be done is starting the newly installed single node cluster:

```
start-dfs.sh
```

While executing this command, you'll be prompted twice with a message similar to the following:

Are you sure you want to continue connecting (yes/no)?

Type in yes for both these prompts and press the enter key. Once this is done, execute the following command:

**start-yarn.sh**

Executing the above two commands will get Hadoop up and running. You can verify this by typing in the following command:

**jps**

If you see some IDs and Name and Data Node displayed alongside, it means that you now have a functional instance of Hadoop running.

## **REFERENCES :**

<https://www.digitalocean.com/community/tutorials/how-to-install-hadoop-on-ubuntu-13-10>

<http://hadoop.apache.org/releases.html>

<http://sourcedigit.com/21172-e-could-not-get-lock-varlibdpkglock-unable-to-lock-the-administration-directory/>

<https://www.tecmint.com/fix-unable-to-lock-the-administration-directory-var-lib-dpkg-lock/>