

# Hadoop 2.6.5 Installing on Ubuntu 16.04 and 18.04 (Single-Node Cluster)

## **Note:**

*1. Jdk 8 is recommended*

*2. copy java path from step4, which required for step 15 and 18.*

*3. Make sure step15 and step18 having same java home path.*

---

## **Step 1:**

updates the package lists for upgrades for packages that need upgrading, as well as new packages that have just come to the repositories.

```
bhaskar@D:~$sudo apt-get update
```

## **Step 2: Installing Java**

```
bhaskar@D:~$sudo apt-get install default-jdk
```

Note: if not install login as root user (sudo -i)

**Note:** sudo apt-get install openjdk-8-jdk (recommended)

## **Step 3: Find version of Java installed**

```
bhaskar@D:~$java -version
```

## **Step 4: To know the java path**

```
sudo update-alternatives --config java
```

```
sudo update-alternatives --config javac
```

## **Step 5: Adding a dedicated Hadoop user**

The next step is to create a dedicated user and group for our Hadoop installation. This allows all of the installation to be insulated from the rest of the environment, as well as enable tighter security measures to be enforced (in case you have a production environment). We will create a user hduser and a group hadoop, and add the user to the group. This can be done using the following commands.

```
bhaskar@D:~$sudo addgroup hadoop
```

#### **Step 6:**

```
bhaskar@D:~$ sudo adduser --ingroup hadoop hduser
```

#### **Step 7: We can check if we create the hadoop group and hduser user**

```
bhaskar@D:~$sudo adduser hduser sudo
```

```
bhaskar@D:~$groups hduser
```

You supposed to get following in terminal

```
hduser : hadoop sudo
```

#### **Step 8:Installing SSH**

The hadoop control scripts rely on SSH to perform cluster-wide operations. For example, there is a script for stopping and starting all the daemons in the clusters. To work seamlessly, SSH needs to be setup to allow password-less login for the hadoop user from machines in the cluster. The simplest way to achieve this is to generate a public/private key pair, and it will be shared across the cluster.

Hadoop requires SSH access to manage its nodes, i.e. remote machines plus your local machine. For our single-node setup of Hadoop, we therefore need to configure SSH access to localhost for the hduser user we created in the earlier.

We have to generate an SSH key for the hduser user.

```
bhaskar@D:~$sudo apt-get install ssh
```

#### **Step 9: Verify installation**

```
bhaskar@D:~$which ssh
```

```
bhaskar@D:~$which sshd
```

#### **Step 10:**

Hadoop uses SSH (to access its nodes) which would normally require the user to enter a password. However, this requirement can be eliminated by creating and setting up SSH certificates using the following commands. If asked for a filename just leave it blank and press the enter key to continue.

```
bhaskar@D:~$ su hduser
```

```
hduser@D:/home/bhaskar$ ssh-keygen -t rsa -P ""
```

The following command adds the newly created key to the list of authorized keys so that Hadoop can use ssh without prompting for a password.

```
hduser@D:/home/bhaskar$ cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
```

#### **Step 11: to check ssh**

```
hduser@D:/home/bhaskar$ ssh localhost
```

#### **Step 12: move the Hadoop installation to the /usr/local/hadoop directory. So, we should create the directory first:**

```
hduser@D:/home/bhaskar$ sudo mkdir -p /usr/local/hadoop
```

#### **Step 13: Install Hadoop**

*Note: based on your Hadoop version please modify below commands*

---

#### **If File downloaded in local drive, follow this**

```
hduser@D:/home/bhaskar$ cd /home/bhaskar/Downloads/hadoop-2.6.5/
```

#### **move the Hadoop installation to the /usr/local/hadoop directory**

```
hduser@D:/home/bhaskar/home/bhaskar/Downloads/hadoop-2.6.5$ sudo mv * /usr/local/hadoop/
```

(or)

#### **If you directly downloaded from mirrors, follow this**

```
hduser@D:/home/bhaskar$
```

```
wget http://mirrors.sonic.net/apache/hadoop/common/hadoop-2.6.5/hadoop-2.6.5.tar.gz  
(check proper mirror)
```

```
hduser@D:/home/bhaskar$ tar xvfz hadoop-2.6.5.tar.gz
```

Move to the folder, where your hadoop download is available and execute the following

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

#### **Step 14: set read/write permission**

```
hduser@D:/home/bhaskar$ sudo chown -R hduser:hadoop /usr/local/hadoop
```

#### **Setup Configuration Files**

#### **Step 15:**

Before editing the .bashrc file in hduser's home directory, we need to find the path where Java has been installed to set the JAVA\_HOME environment variable using **Step 4**

**hduser@D:/home/bhaskar\$ vim ~/.bashrc**

**(or)**

**hduser@D:/home/bhaskar\$ sudo gedit ~/.bashrc**

**Add the following @ end**

```
#HADOOP VARIABLES START
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
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=$HADOOP_INSTALL/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_INSTALL/lib"
export HADOOP_HOME_WARN_SUPPRESS=1
export HADOOP_ROOT_LOGGER="WARN,DRFA"
```

**#HADOOP VARIABLES END**

**Step 16:**

**hduser@D:/home/bhaskar\$ source ~/.bashrc**

**Step 17:**

**hduser@D:/home/bhaskar\$ vi /usr/local/hadoop/etc/hadoop/hadoop-env.sh**

**Add the following**

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

**Step 18:**

**hduser@D:/home/bhaskar\$ sudo mkdir -p /app/hadoop/tmp**

**hduser@D:/home/bhaskar\$ sudo chown hduser:hadoop /app/hadoop/tmp**

**Step 19:**

**hduser@D:/home/bhaskar\$ vi /usr/local/hadoop/etc/hadoop/core-site.xml**

**Open the file and enter the following in between the <configuration></configuration> tag:**

```
<property>
  <name>hadoop.tmp.dir</name>
  <value>/app/hadoop/tmp</value>
  <description>A base for other temporary directories.</description>
```

```

</property>

<property>
  <name>fs.default.name</name>
  <value>hdfs://localhost:54310</value>
  <description>The name of the default file system. A URI whose
  scheme and authority determine the FileSystem implementation. The
  uri's scheme determines the config property (fs.SCHEME.impl) naming
  the FileSystem implementation class. The uri's authority is used to
  determine the host, port, etc. for a filesystem.</description>
</property>

```

### **Step 20:**

```
hduser@D:/home/bhaskar$ cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template
/usr/local/hadoop/etc/hadoop/mapred-site.xml
```

**Note: Higher version of hadoop the step 20 not required.**

### **Step 21:**

```
hduser@D:/home/bhaskar$ vim /usr/local/hadoop/etc/hadoop/mapred-site.xml
```

Open the file and enter the following in between the <configuration></configuration> tag:

### **Add the following**

```

<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>

```

### **Step 22:**

```
hduser@D:/home/bhaskar$ sudo mkdir -p /usr/local/hadoop_store/hdfs/namenode
```

```
hduser@D:/home/bhaskar$ sudo mkdir -p /usr/local/hadoop_store/hdfs/datanode
```

```
hduser@D:/home/bhaskar$ sudo chown -R hduser:hadoop /usr/local/hadoop_store
```

### **Step 23:**

```
hduser@D:/home/bhaskar$ vim /usr/local/hadoop/etc/hadoop/hdfs-site.xml
```

Open the file and enter the following in between the <configuration></configuration> tag:

### **And add the following**

```

<property>
  <name>dfs.replication</name>
  <value>1</value>
  <description>Default block replication.
  The actual number of replications can be specified when the file is created.
  The default is used if replication is not specified in create time.
</description>
</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>

```

#### **Step 24:**

**hduser@D:/home/bhaskar\$ hadoop namenode -format**

Important Note :

- Note that hadoop namenode -format command should be executed **once** before we start using Hadoop.
- If this command is executed again after Hadoop has been used, it'll destroy all the data on the Hadoop file system.

#### **Step 25: Starting Hadoop**

**hduser@D:/home/bhaskar\$ start-all.sh**

#### **Step 26:**

**hduser@D:/home/bhaskar\$ jps**

For checking running process in our Hadoop cluster we use JPS command. **JPS** stands for **Java Virtual Machine Process Status Tool**.

After running jps command the following daemons should start.

```

7040 NameNode
7956 Jps
7156 DataNode
7525 ResourceManager
7367 SecondaryNameNode
7834 NodeManager

```

**Note: Your Hadoop installation is successful only if above daemons should start**

**Step 27: Stop Hadoop**

**hduser@D:/home/bhaskar\$ stop-all.sh**