

Exp.No: 1**Downloading and installing Hadoop, Understanding different Hadoop modes, Startup scripts, Configuration files.****AIM:**

To Download and install Hadoop, Understanding different Hadoop modes, Startup scripts, Configuration files.

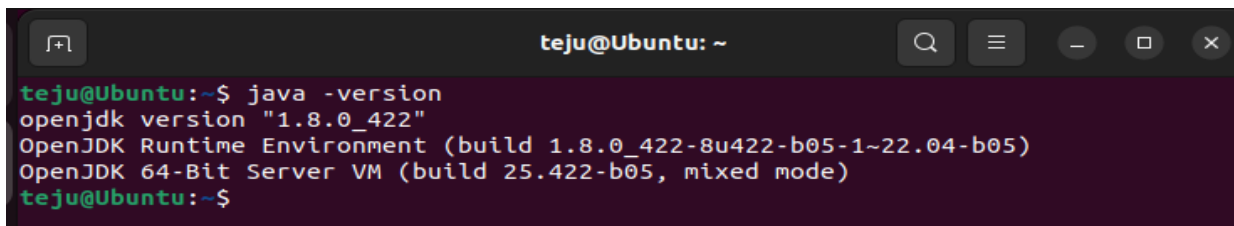
Procedure:**Step 1 : Install Java Development Kit**

The default Ubuntu repositories contain Java 8 and Java 11 both. But, Install Java 8 because hive only works on this version. Use the following command to install it.

```
$sudo apt update&&sudo apt install openjdk-8-jdk
```

Step 2 : Verify the Java version

Once installed, verify the installed version of Java with the following command: \$

java -version Output:A terminal window titled 'teju@Ubuntu: ~' with search, menu, and window control icons. The command 'java -version' has been executed, resulting in the following output: 'openjdk version "1.8.0_422"', 'OpenJDK Runtime Environment (build 1.8.0_422-8u422-b05-1~22.04-b05)', and 'OpenJDK 64-Bit Server VM (build 25.422-b05, mixed mode)'. The prompt returns to 'teju@Ubuntu:~\$'.**Step 3: Install SSH**

SSH (Secure Shell) installation is vital for Hadoop as it enables secure communication between nodes in the Hadoop cluster. This ensures data integrity, confidentiality, and allows for efficient distributed processing of data across the cluster. **\$sudo apt install ssh**

Step 4 : Create the hadoop user :

All the Hadoop components will run as the user that you create for Apache Hadoop, and the user will also be used for logging in to Hadoop's web interface. Run the command to create user and set password:

```
$ sudo adduser Hadoop
```

Step 5 : Switch user

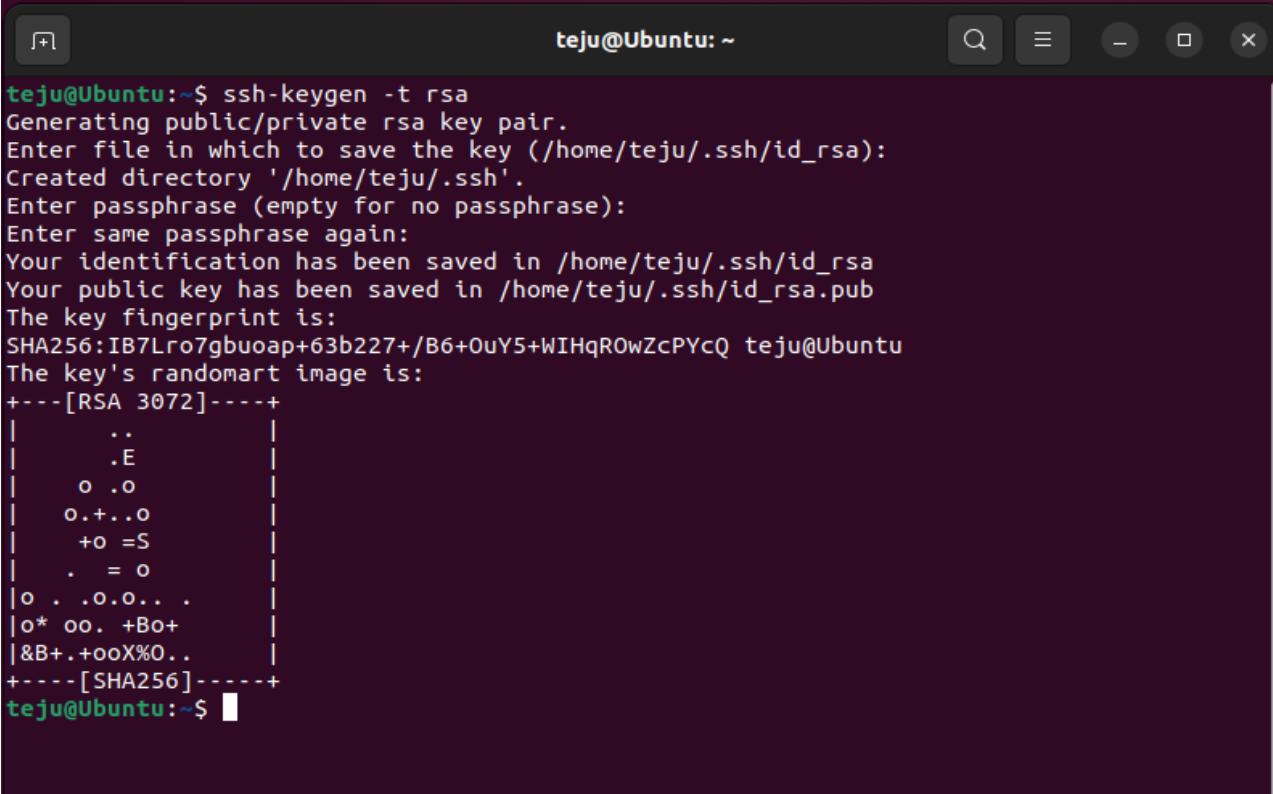
Switch to the newly created hadoop user:

```
$ su - hadoop
```

Step 6 : Configure SSH

Now configure password-less SSH access for the newly created hadoop user, so didn't enter the key to save file and passphrase. Generate an SSH keypair (generate Public and Private Key Pairs)first

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



```
teju@Ubuntu:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/teju/.ssh/id_rsa):
Created directory '/home/teju/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/teju/.ssh/id_rsa
Your public key has been saved in /home/teju/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:IB7Lro7gbuoap+63b227+/B6+0uY5+WIHqROWZcPYcQ teju@Ubuntu
The key's randomart image is:
+---[RSA 3072]---+
|      ..          |
|      .E          |
|     o .o         |
|    o+.o.o        |
|   +o =S          |
|   . = o          |
|o . .o.o.o.. .   |
|o* oo. +Bo+      |
|&B+.+ooX%O..    |
+---[SHA256]-----+
teju@Ubuntu:~$
```

Step 7 : Set permissions :

Next, append the generated public keys from id_rsa.pub to authorized_keys and set proper permission:

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

```
$ chmod 640 ~/.ssh/authorized_keys
```

Step 8 : SSH to the localhost

Next, verify the password less SSH authentication with the following command:

```
$ ssh localhost
```

You will be asked to authenticate hosts by adding RSA keys to known hosts. Type yes and hit Enter to authenticate the localhost:

```

teju@Ubuntu: ~
teju@Ubuntu:~$ ssh localhost
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.8.0-40-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

54 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

4 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Last login: Fri Aug 23 18:13:55 2024 from 127.0.0.1
teju@Ubuntu:~$

```

Step 9 : Switch user

Again switch to hadoop. So, First, change the user to hadoop with the following command: **\$ su-hadoop**

Step 10 : Install hadoop

Next, download the latest version of Hadoop using the wget command:

\$ wget <https://downloads.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz> Once downloaded, extract the downloaded file:

\$ tar -xvzf hadoop-3.3.6.tar.gz

Next, rename the extracted directory to hadoop:

\$ mv hadoop-3.3.6 hadoop

```

teju@Ubuntu: ~
teju@Ubuntu:~$ ls
Desktop    hadoop-3.4.0  pig        pseudo      snap        workspace
Documents  mapper.py     pig-0.17.0.tar.gz  Public      Templates
Downloads  Music        pig_1725957258136.log  R           Videos
emp.json   Pictures     process_data.py      reducer.py  wordcount
teju@Ubuntu:~$

```

Next, you will need to configure Hadoop and Java Environment Variables on your system. Open the `~/.bashrc` file in your favorite text editor. Use nano editor , to pasting the code we use `ctrl+shift+v` for saving the file `ctrl+x` and `ctrl+y` ,then hit enter:

Next, you will need to configure Hadoop and Java Environment Variables on your system.

Open the `~/.bashrc` file in your favorite text editor:

\$ nano ~/.bashrc

Append the below lines to file.

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=/home/hadoop/hadoop
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export HADOOP_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"
```

Save and close the file. Then, activate the environment variables with the following command:

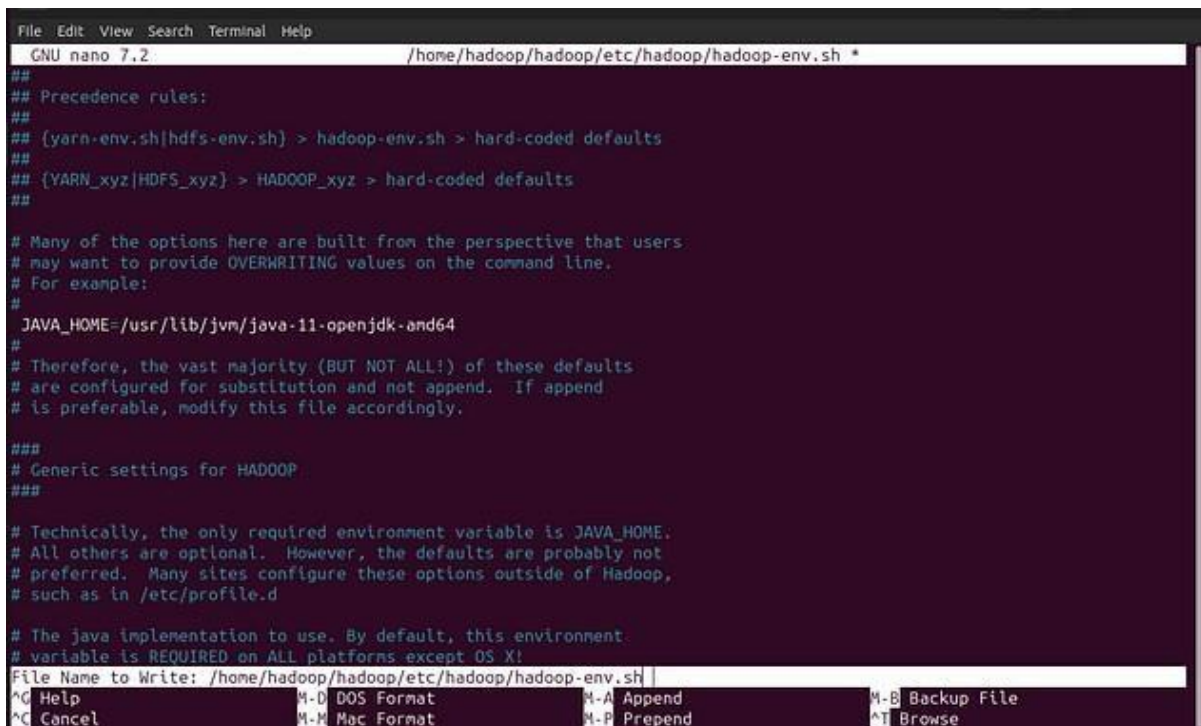
s\$ source ~/.bashrc

Next, open the Hadoop environment variable file: **\$ nano**

\$HADOOP_HOME/etc/hadoop/hadoop-env.sh

Search for the “export JAVA_HOME” and configure it.

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



```
File Edit View Search Terminal Help
GNU nano 7.2 /home/hadoop/hadoop/etc/hadoop/hadoop-env.sh *
##
## Precedence rules:
##
## (yarn-env.sh|hdfs-env.sh) > hadoop-env.sh > hard-coded defaults
##
## {YARN_xyz|HDFS_xyz} > HADOOP_xyz > hard-coded defaults
##
# Many of the options here are built from the perspective that users
# may want to provide OVERWRITING values on the command line.
# For example:
#
# JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
#
# Therefore, the vast majority (BUT NOT ALL!) of these defaults
# are configured for substitution and not append. If append
# is preferable, modify this file accordingly.
###
# Generic settings for HADOOP
###
# Technically, the only required environment variable is JAVA_HOME.
# All others are optional. However, the defaults are probably not
# preferred. Many sites configure these options outside of Hadoop,
# such as in /etc/profile.d
#
# The java implementation to use. By default, this environment
# variable is REQUIRED on ALL platforms except OS X!
File Name to Write: /home/hadoop/hadoop/etc/hadoop/hadoop-env.sh |
^C Help ^M-D DOS Format ^M-A Append ^M-B Backup File
^C Cancel ^M-M Mac Format ^M-P Prepend ^M-T Browse
```

Save and close the file when you are finished.

Step 11 : Configuring Hadoop :

First, you will need to create the namenode and datanode directories inside the Hadoop user home directory. Run the following command to create both directories:

```
$ cd hadoop/
```

```
$mkdir -p ~/hadoopdata/hdfs/{namenode,datanode}
```

- Next, edit the core-site.xml file and update with your system hostname:

```
$nano $HADOOP_HOME/etc/hadoop/core-site.xml
```

Change the following name as per your system hostname:

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

Save and close the file.

Then, edit the hdfs-site.xml file:

```
$nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml
```

- Change the NameNode and DataNode directory paths as shown below:

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>

  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:///home/hadoop/hadoopdata/hdfs/namenode</value>
  </property>

  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:///home/hadoop/hadoopdata/hdfs/datanode</value>
  </property>
</configuration>
```

- Then, edit the mapred-site.xml file:

```
$nano $HADOOP_HOME/etc/hadoop/mapred-site.xml
```

- Make the following changes:

```
<configuration>
  <property>
    <name>yarn.app.mapreduce.am.env</name>
    <value>HADOOP_MAPRED_HOME=$HADOOP_HOME/home/hadoop/hadoop/bin/hadoop</value>
  </property>
  <property>
    <name>mapreduce.map.env</name>
    <value>HADOOP_MAPRED_HOME=$HADOOP_HOME/home/hadoop/hadoop/bin/hadoop</value>
  </property>
  <property>
    <name>mapreduce.reduce.env</name>
    <value>HADOOP_MAPRED_HOME=$HADOOP_HOME/home/hadoop/hadoop/bin/hadoop</value>
  </property>
</configuration>
```

- Then, edit the yarn-site.xml file:
\$nano \$HADOOP_HOME/etc/hadoop/yarn-site.xml
- Make the following changes:

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
</configuration>
```

Save the file and close it .

Step 12 – Start Hadoop Cluster

Before starting the Hadoop cluster. You will need to format the Namenode as a hadoop user.

Run the following command to format the Hadoop Namenode:

```
$hdfs namenode -format
```

Once the namenode directory is successfully formatted with hdfs file system, you will see the message “Storage directory /home/hadoop/hadoopdata/hdfs/namenode has been successfully formatted “

Then start the Hadoop cluster with the following command.

\$ start-all.sh

```
teju@Ubuntu:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as teju in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
localhost: namenode is running as process 15163. Stop it first and ensure /tmp/hadoop
-teju-namenode.pid file is empty before retry.
Starting datanodes
localhost: datanode is running as process 15267. Stop it first and ensure /tmp/hadoop
-teju-datanode.pid file is empty before retry.
Starting secondary namenodes [Ubuntu]
Ubuntu: secondarynamenode is running as process 15435. Stop it first and ensure /tmp/
hadoop-teju-secondarynamenode.pid file is empty before retry.
2024-09-30 21:11:34,567 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
Starting resourcemanager
resourcemanager is running as process 17540. Stop it first and ensure /tmp/hadoop-tej
u-resourcemanager.pid file is empty before retry.
Starting nodemanagers
localhost: nodemanager is running as process 17647. Stop it first and ensure /tmp/had
oop-teju-nodemanager.pid file is empty before retry.
teju@Ubuntu:~$
```


You can now check the status of all Hadoop services using the jps command:

\$ jps

```
teju@Ubuntu:~$ jps
15267 DataNode
17066 GetConf
17083 Jps
15435 SecondaryNameNode
15163 NameNode
teju@Ubuntu:~$
```

Step 13 – Access Hadoop Namenode and Resource Manager

- First we need to know our ipaddress, In Ubuntu we need to install net-tools to run ipconfig command,

If you installing net-tools for the first time switch to default user:

\$sudo apt install net-tools

- Then run ifconfig command to know our ip address: **ifconfig**

Here my ip address is 192.168.1.6.

- To access the Namenode, open your web browser and visit the URL <http://your-serverip:9870>.
- You should see the following screen:
<http://192.168.1.6:9870>

Overview 'localhost:9000' (active)

Started:	Mon Sep 02 10:43:55 +0530 2024
Version:	3.3.6, r1be78238728da9266a4f88195058f08fd012bf9c
Compiled:	Sun Jun 18 13:52:00 +0530 2023 by ubuntu from (HEAD detached at release-3.3.6-RC1)
Cluster ID:	CID-73012808-a614-4a4a-aa57-40b8fd6716fd
Block Pool ID:	BP-1797801860-127.0.1.1-1725252549180

Summary

Security is off.
Safemode is off.
16 files and directories, 6 blocks (6 replicated blocks, 0 erasure coded block groups) = 22 total filesystem object(s).
Heap Memory used 77.73 MB of 221 MB Heap Memory. Max Heap Memory is 690 MB.
Non Heap Memory used 54.34 MB of 55.69 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	24.44 GB
Configured Remote Capacity:	0 B
DFS Used:	456 KB (0%)
Non DFS Used:	11.77 GB
DFS Remaining:	11.4 GB (46.66%)

To access Resource Manage, open your web browser and visit the URL <http://your-serverip:8088>. You should see the following screen: <http://192.168.16:8088>

Step 14 – Verify the Hadoop Cluster

At this point, the Hadoop cluster is installed and configured. Next, we will create some directories in the HDFS filesystem to test the Hadoop.

Let's create some directories in the HDFS filesystem using the following commands:

```
$ hdfsdfs -mkdir /test1
$ hdfsdfs -mkdir /logs
```

Next, run the following command to list the above directory:

```
teju@Ubuntu:~$ hdfs dfs -ls /
2024-09-30 21:21:10,339 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 9 items
drwxr-xr-x - user supergroup          0 2024-08-29 22:54 /WordCount
-rw-r--r-- 1 user supergroup      51647 2024-08-31 11:28 /input_txt
drwxr-xr-x - user supergroup          0 2024-09-10 14:34 /json
drwxr-xr-x - user supergroup          0 2024-09-01 22:45 /pig_output_data
drwxr-xr-x - user supergroup          0 2024-09-01 22:11 /piginput
drwxrwxrwx - user supergroup          0 2024-09-10 14:05 /tmp
drwxr-xr-x - user supergroup          0 2024-09-01 22:25 /udfs
drwxr-xr-x - user supergroup          0 2024-09-10 10:15 /user
drwxr-xr-x - user supergroup          0 2024-09-01 20:40 /weatherdata
teju@Ubuntu:~$
```

Also, put some files to hadoop file system. For the example, putting log files from host machine to hadoop file system.

```
$ hdfs dfs -put /var/log/* /logs/
```


You can also verify the above files and directory in the Hadoop Namenode web interface.

Go to the web interface, click on the Utilities => Browse the file system. You should see your directories which you have created earlier in the following screen:

The screenshot shows the Hadoop Namenode web interface. The top navigation bar includes links for Hadoop, Overview, Datanodes, Datanode Volume Failures, Snapshot, Startup Progress, and Utilities. The main section is titled 'Browse Directory' and shows a file explorer view. The address bar indicates the path is '/'. Below the address bar, there are icons for file operations and a search bar. A table lists the contents of the directory, showing columns for Permission, Owner, Group, Size, Last Modified, Replication, Block Size, and Name. The table contains five entries: 'home', 'tmp', 'user', 'weatherdata', and 'word_count_in_python'. At the bottom, there is a pagination control showing 'Showing 1 to 5 of 5 entries' and buttons for 'Previous', '1', and 'Next'.

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	hadoop	supergroup	0 B	Sep 02 12:12	0	0 B	home
drwxrwxr-x	hadoop	supergroup	0 B	Sep 02 13:29	0	0 B	tmp
drwxr-xr-x	hadoop	supergroup	0 B	Sep 02 13:26	0	0 B	user
drwxr-xr-x	hadoop	supergroup	0 B	Sep 02 11:38	0	0 B	weatherdata
drwxr-xr-x	hadoop	supergroup	0 B	Sep 03 20:04	0	0 B	word_count_in_python

Step 15 – Stop Hadoop Cluster

To stop the Hadoop all services, run the following command:

\$ stop-all.sh

```
teju@Ubuntu:~$ stop-all.sh
WARNING: Stopping all Apache Hadoop daemons as teju in 10 seconds.
WARNING: Use CTRL-C to abort.
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [Ubuntu]
2024-09-30 21:28:12,740 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Stopping nodemanagers
Stopping resourcemanager
teju@Ubuntu:~$
```

Result:

The step-by-step installation and configuration of Hadoop on Ubuntu linux system have been successfully completed.

