

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**

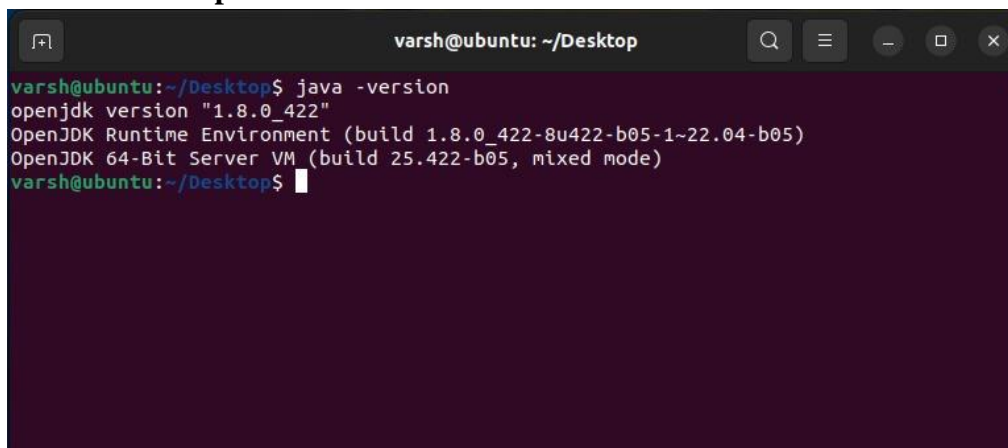
The default Ubuntu repositories contain Java 8 and Java 11 both. But, Install Java 8 because it only works on this version. Use the following command to install it.

```
$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 'varsh@ubuntu: ~/Desktop' with search, menu, and window control icons. The terminal shows the command 'java -version' being executed. The output is: '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 'varsh@ubuntu:~/Desktop\$'.**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

```

varsh@ubuntu: ~
hadoop-3.2.3/lib/native/libnativehadoop.so
hadoop-3.2.3/lib/native/libhdfspp.a
hadoop-3.2.3/LICENSE.txt
varsh@ubuntu:~$ cd hadoop-3.2.3/
varsh@ubuntu:~/hadoop-3.2.3$ ls
bin  etc  include  lib  libexec  LICENSE.txt  NOTICE.txt  README.txt  sbin  share
varsh@ubuntu:~/hadoop-3.2.3$ cd etc/hadoop/
varsh@ubuntu:~/hadoop-3.2.3/etc/hadoop$ ls
capacity-scheduler.xml      https-log4j.properties      mapred-site.xml
configuration.xml          https-signature.secret      shellprofile.d
container-executor.cfg     https-site.xml              ssl-client.xml.example
core-site.xml              kms-acls.xml                ssl-server.xml.example
hadoop-env.cmd             kms-env.sh                  user_ec_policies.xml.template
hadoop-env.sh              kms-log4j.properties        workers
hadoop-metrics2.properties kms-site.xml                yarn-env.cmd
hadoop-policy.xml          log4j.properties           yarn-env.sh
hadoop-user-functions.sh.example mapred-env.cmd              yarnservice-log4j.properties
hdfs-site.xml              mapred-env.sh              yarn-site.xml
https-env.sh               mapred-queues.xml.template
varsh@ubuntu:~/hadoop-3.2.3/etc/hadoop$ sudo nano hadoop-env.sh
[sudo] password for varsh:
varsh@ubuntu:~/hadoop-3.2.3/etc/hadoop$ sudo nano core-site.xml
varsh@ubuntu:~/hadoop-3.2.3/etc/hadoop$ sudo nano hdfs-site.xml
varsh@ubuntu:~/hadoop-3.2.3/etc/hadoop$ sudo nano mapred-site.xml
varsh@ubuntu:~/hadoop-3.2.3/etc/hadoop$ sudo nano yarn-site.xml
varsh@ubuntu:~/hadoop-3.2.3/etc/hadoop$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ED25519 key fingerprint is SHA256:pyydpR3wKFJgIoa0wnRP5WDyi8BayuqT5/GbGzLwCCJ0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.
varsh@localhost's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-41-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.

```

Now to 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

```

varsh@ubuntu: ~
* Management:  https://landscape.canonical.com
* Support:     https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

252 updates can be applied immediately.
195 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

varsh@ubuntu:~$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
Generating public/private rsa key pair.
Your identification has been saved in /home/varsh/.ssh/id_rsa
Your public key has been saved in /home/varsh/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:e7xxP6SVY7cFR0IPLNhqA6GQW09QnQt6wJNBqd2gTQ varsh@ubuntu
The key's randomart image is:
+---[RSA 3072]-----+
|  .oE=Bo.. |
| ..oO+oO+ . |
| o=O+oO.o + |
| .. o.O.. o o |
|      S = . o |
|      = . . o |
|      . + .o .. |
|      . ++.o o |
|      ...=++.. |
+---[SHA256]-----+
varsh@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:

```

varsh@ubuntu: ~
varsh@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.

128 updates can be applied immediately.
37 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Last login: Thu Sep 19 21:56:33 2024 from 127.0.0.1
varsh@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

```

varsh@ubuntu:~$ ls
apache-hive-3.1.3-bin.tar.gz      pig
app.py                           pig-0.16.0.tar.gz
demo_pig.pig                     pig_1726770462440.log
Desktop                          pig_1726770738890.log
Documents                        reducer1.py
Downloads                        reducer.py
hadoop-3.2.3                     snap
hadoop_data                     udf_example.pig
mapper1.py                      udf.txt
mapper.py                       uppercase_udf.py
mysql-apt-config_0.8.22-1_all.deb weather_output.txt
new_count                       weather.txt
output_wordcount.txt            word_count.txt
varsh@ubuntu:~$

```

\$ 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:

\$ 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/yarnsite.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

```
varsh@ubuntu:~$ /home/varsh/hadoop-3.2.3/sbin/start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as varsh in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [ubuntu]
Starting resourcemanager
resourcemanager is running as process 9810. Stop it first and ensure /tmp/hadoop-varsh-resourcemanager.pid file is empty before retry.
Starting nodemanagers
```

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

\$ jps

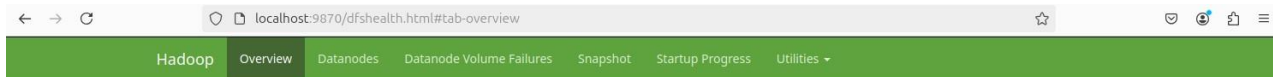
```
varsh@ubuntu:~$ jps
4000 SecondaryNameNode
4370 NodeManager
4251 ResourceManager
3724 NameNode
4700 Jps
3838 DataNode
varsh@ubuntu:~$
```

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://yourserverip: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 command:

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

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

```
varsh@ubuntu:~$ hdfs dfs -ls /
Found 4 items
drwxr-xr-x  - varsh supergroup          0 2024-09-19 23:47 /home
drwx-----  - varsh supergroup          0 2024-09-20 00:33 /tmp
drwxr-xr-x  - varsh supergroup          0 2024-09-19 23:22 /weatherdata
drwxr-xr-x  - varsh supergroup          0 2024-09-19 23:15 /word_count_in_pytho
n
varsh@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:

Browse Directory

/ Go!

Show 25 entries Search:

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

Showing 1 to 5 of 5 entries

Previous 1 Next

Hadoop, 2023.

Step 15 – Stop Hadoop Cluster

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

\$stop-all.sh

```
varsh@ubuntu:~$ /home/varsh/hadoop-3.2.3/sbin/stop-all.sh
WARNING: Stopping all Apache Hadoop daemons as varsh in 10 seconds.
WARNING: Use CTRL-C to abort.
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [ubuntu]
Stopping nodemanagers
Stopping resourcemanager
varsh@ubuntu:~$
```

Result:

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