

**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:**

```
srinathi@srinathi-VirtualBox:~$ java -version
openjdk version "11.0.24" 2024-07-16
OpenJDK Runtime Environment (build 11.0.24+8-post-Ubuntu-1ubuntu324.04.1)
OpenJDK 64-Bit Server VM (build 11.0.24+8-post-Ubuntu-1ubuntu324.04.1, mixed mode, sharing)
srinathi@srinathi-VirtualBox:~$
```

**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

```
srinathi@srinathi-VirtualBox:~/hadoop/etc/hadoop$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/srinathi/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/srinathi/.ssh/id_rsa
Your public key has been saved in /home/srinathi/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:VqhvZqy32kQpmeQlsYy2g0iSTty0SThT2rt3aG19w3M srinathi@srinathi-VirtualBox
The key's randomart image is:
+---[RSA 3072]-----+
|
|. +
|=.0 . .
|o*++ o + o
|+B=0. * S
|O .=.o *
|o++ . . B
| ..o+oEB.
| .+ooo.
+----[SHA256]-----+
srinathi@srinathi-VirtualBox:~/hadoop/etc/hadoop$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized
```

### 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:

```
hadoop@localhost's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-41-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.

154 updates can be applied immediately.
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

*** System restart required ***
Last login: Mon Sep  2 10:14:41 2024 from 127.0.0.1
```

**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>**

```

srinath@srinath-VirtualBox:~$ wget https://d1cdn.apache.org/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz
--2024-09-11 22:08:49-- https://d1cdn.apache.org/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz
Resolving d1cdn.apache.org (d1cdn.apache.org)... 151.101.2.132, 2084:4e42::644
Connecting to d1cdn.apache.org (d1cdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 965537117 (921M) [application/x-gzip]
Saving to: 'hadoop-3.4.0.tar.gz'

hadoop-3.4.0.tar.gz           12K[=====] 117.97M 365KB/s eta 49
hadoop-3.4.0.tar.gz           17K[=====] 160.92M 127KB/s eta 60m 51s
hadoop-3.4.0.tar.gz           32K[=====] 302.21M 126KB/s eta 63m 4s
hadoop-3.4.0.tar.gz           35K[=====] 326.14M 126KB/s eta 61m 34s
hadoop-3.4.0.tar.gz           42K[=====] 387.49M 132KB/s eta 57m 30s
hadoop-3.4.0.tar.gz           55K[=====] 509.52M 127KB/s eta 5
hadoop-3.4.0.tar.gz           55K[=====] hadoop-3.4.0.tar.gz 83K[=====] 7
hadoop-3.4.0.tar.gz           87K[=====] 809.89M 127KB/s eta 11m 15s
hadoop-3.4.0.tar.gz           88K[=====] 813.78M 127KB/s eta 10m 52s
hadoop-3.4.0.tar.gz           95K[=====] 882.13M 126KB/s eta 4m
hadoop-3.4.0.tar.gz          100K[=====] 920.81M 131KB/s ln 95m 3s

2024-09-11 23:43:55 (165 KB/s) - 'hadoop-3.4.0.tar.gz' saved [965537117/965537117]

```

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**

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
^G Help ^O-D DOS Format ^M-A Append ^M-B Backup File
^C Cancel ^M-M Mac Format ^M-P Prepend ^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**

```
srinathi@srinathi-VirtualBox:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as srinathi 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 [srinathi-VirtualBox]
Starting resourcemanager
Starting nodemanagers
```

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

\$ jps

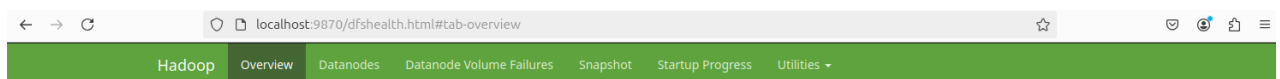
```
srinathi@srinathi-VirtualBox:~$ jps
4992 NameNode
6225 Jps
5447 SecondaryNameNode
5803 NodeManager
5660 ResourceManager
5167 DataNode
```

### 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 (46.65%)

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

The screenshot shows the Hadoop Resource Manager web interface. The browser address bar displays [192.168.16:8088/cluster](http://192.168.16:8088/cluster). The Hadoop logo is at the top left. A sidebar menu on the left includes 'Cluster' (selected), 'About', 'Nodes', 'Node Labels', 'Applications', and 'Tools'. The main content area displays several tables:

- Cluster Metrics:** A table with columns: Apps Submitted (0), Apps Pending (0), Apps Running (0), Apps Completed (0), and Containers Running (0).
- Cluster Nodes Metrics:** A table with columns: Active Nodes (1), Decommissioning Nodes (0), and Decommissioned Nodes (0).
- Scheduler Metrics:** A table with columns: Scheduler Type (Capacity Scheduler), Scheduling Resource Type (memory-mb (unit=Mb), vcores), and Minimum Resource (memory:1024, vCore:1).
- Applications:** A table with columns: ID, User, Name, Application Type, Application Tags, Queue, Application Priority, StartTime, LaunchTime, and FinishTime. It shows 0 entries.

### 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:

```
srinathi@srinathi-VirtualBox:~$ hdfs dfs -ls /
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Found 5 items
drwxr-xr-x - srinathi supergroup 0 2024-09-19 21:10 /home
drwxrwxr-x - srinathi supergroup 0 2024-09-19 21:21 /tmp
drwxrwxr-x - srinathi supergroup 0 2024-09-20 19:18 /user
drwxr-xr-x - srinathi supergroup 0 2024-09-19 20:46 /weatherdata
drwxr-xr-x - srinathi supergroup 0 2024-09-18 10:27 /word_count_in_python
srinathi@srinathi-VirtualBox:~$
```

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 at localhost:9870/explorer.html. The 'Browse Directory' page displays a table of files and directories in the root directory. The table columns include Permission, Owner, Group, Size, Last Modified, Replication, Block Size, and Name. The files listed are home, tmp, user, weatherdata, and word\_count\_in\_python. The page also shows a search bar, a 'Go!' button, and a 'Showing 1 to 5 of 5 entries' message.

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	srinathi	supergroup	0 B	Sep 21 00:13	0	0 B	home
drwxr-xr-x	srinathi	supergroup	0 B	Sep 20 23:24	0	0 B	tmp
drwxr-xr-x	srinathi	supergroup	0 B	Sep 20 23:13	0	0 B	user
drwxr-xr-x	srinathi	supergroup	0 B	Sep 19 20:46	0	0 B	weatherdata
drwxr-xr-x	srinathi	supergroup	0 B	Sep 18 10:27	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
```

```
srinathi@srinathi-VirtualBox:~$ stop-all.sh
WARNING: Stopping all Apache Hadoop daemons as srinathi in 10 seconds.
WARNING: Use CTRL-C to abort.
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [srinathi-VirtualBox]
Stopping nodemanagers
Stopping resourcemanager
srinathi@srinathi-VirtualBox:~$
```

## Result:

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