

**Ex 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:**

```
varunesh@varunesh:~$ java -version  
openjdk version "1.8.0_422"  
OpenJDK Runtime Environment (build 1.8.0_422-8u422-b05-1~22.04-b  
05)  
OpenJDK 64-Bit Server VM (build 25.422-b05, mixed mode)
```

**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
```

**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:

```
varunesh@varunesh:~$ ssh localhost
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.153.1-microsoft-standard-W
SL2 x86_64)

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

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how Mic
roK8s
   just raised the bar for easy, resilient and secure K8s cluster deploy
ment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge
New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat Sep 28 22:08:33 2024
```

**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
```

```
varunesh@varunesh:~$ :~/hadoop$ ls
LICENSE-binary  NOTICE.txt  etc          libexec      sbin
LICENSE.txt     README.txt  include      licenses-binary  share
NOTICE-binary   bin         lib          logs
```

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 PATH=$PATH:/usr/lib/jvm/java-8-openjdk-amd64/bin
export HADOOP_HOME=/home/subhikshaa/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
export HADOOP_STREAMING=$HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming
export HADOOP_LOG_DIR=$HADOOP_HOME/logs
export PDSH_RCMD_TYPE=ssh
```

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**

```
# Location of Hadoop. By default, Hadoop will attempt to determine
# this location based upon its execution path.
# export HADOOP_HOME=
JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

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
```

```
varunesh@varunesh:~$ :~/hadoop/etc/hadoop$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as subhikshaa 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 [Subhikshaa]
Starting resourcemanager
Starting nodemanagers
```

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

\$ jps

```
varunesh@varunesh:~$ :~/hadoop/etc/hadoop$ jps
4914 SecondaryNameNode
5190 ResourceManager
4698 DataNode
5340 NodeManager
5886 Jps
```

### 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)

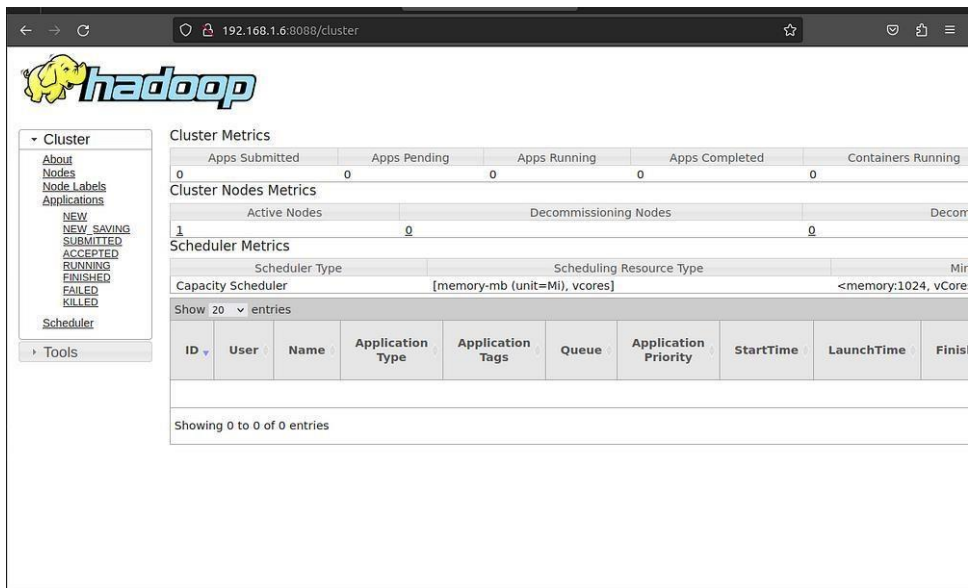
Started:	Mon Sep 02 10:43:55 +0530 2024
Version:	3.3.6, r1be78238728da9266a4f88195058f08fd012bf9c
Compiled:	Sun Jun 18 13:52:00 +0530 2023 by ubuntu from (HEAD detached at release-3.3.6-RC1)
Cluster ID:	CID-73012808-a614-4a4a-aa57-40b8fd6716fd
Block Pool ID:	BP-1797801860-127.0.1.1-1725252549180

### 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 Manage, 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 web interface at 192.168.1.6:8088/cluster. The interface includes a sidebar with navigation links (Cluster, About, Nodes, Node Labels, Applications, Scheduler, Tools) and a main content area. The main content area displays various metrics:

- Cluster Metrics:** A table showing App Status (Submitted, Pending, Running, Completed) and Containers Running, all with values of 0.
- Cluster Nodes Metrics:** A table showing Active Nodes (1) and Decommissioning Nodes (0).
- Scheduler Metrics:** A table showing Scheduler Type (Capacity Scheduler) and Scheduling Resource Type (memory-mb (unit=Mi), vcores).
- Applications:** A table showing application details (ID, User, Name, Application Type, Application Tags, Queue, Application Priority, StartTime, LaunchTime, FinishTime).

## 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:

```
$ hdfs dfs -ls /
```

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**

Search:

Show 25 entries

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
```

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

## Result:

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