

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.

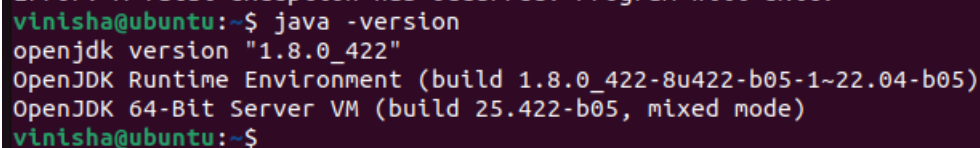
Procedure:**Step 1 : Install Java Development Kit**

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 with a dark background showing the output of the 'java -version' command. The text is as follows:
vinisha@ubuntu:~\$ java -version
openjdk version "1.8.0_422"
OpenJDK Runtime Environment (build 1.8.0_422-8u422-b05-1~22.04-b05)
OpenJDK 64-Bit Server VM (build 25.422-b05, mixed mode)
vinisha@ubuntu:~\$
The prompt 'vinisha@ubuntu:~\$' appears at the beginning and end of the output block.
ss**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

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/swathi/.ssh/id_rsa):
/home/swathi/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/swathi/.ssh/id_rsa
Your public key has been saved in /home/swathi/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:8YFvLMmpg8ekFMMP6kTWzwYKG8aA9UcbL9074MoHNTY swathi@swathi-VirtualBox
The key's randomart image is:
+---[RSA 3072]-----+
|o+.  o=              |
|.  =.o.=B  o  +      |
|.  *.o*** B          |
|.  +oEB=B  .          |
|  . . .              |
|  . . .              |
|  o.                 |
+---[SHA256]-----+
```

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:

```

vinisha@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.

82 updates can be applied immediately.
54 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

New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Sep 12 14:37:59 2024 from 127.0.0.1
vinisha@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

```

vinisha@ubuntu:~$ ls
apache-hive-3.1.2-bin      Downloads      pig
apache-hive-3.1.2-bin.tar.gz  hadoop-3.4.0  pig-0.16.0.tar.gz
DALab                    hadoop_data   Public
demo.csv                 hadoop-streaming-3.4.0.jar  snap
demo_pig.pig             metastore_db  Templates
derby.log                 Music         Videos
Desktop                  '~p'         word_count.txt
Documents                 Pictures
vinisha@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

```
vinisha@ubuntu: ~/hadoop-3.4.0/etc/hadoop
GNU nano 6.2 hadoop-env.sh
# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements.  See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership.  The ASF licenses this file
# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
# with the License.  You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
# Set Hadoop-specific environment variables here.
##
[ Read 434 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify  ^_ Go To Line
```

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

```
vinisha@ubuntu:/home$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as vinisha 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
Starting nodemanagers
vinisha@ubuntu:/home$
```

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

\$ jps

```
command jps not found, but there are 10 similar ones.
vinisha@ubuntu:~$ jps
3543 SecondaryNameNode
3816 ResourceManager
4298 Jps
3930 NodeManager
3372 DataNode
3260 NameNode
vinisha@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>

The screenshot shows the Hadoop Namenode information page. The browser address bar indicates the URL is `localhost:9870/dfshealth.html#tab-overview`. The page has a green navigation bar with tabs: Hadoop, Overview, Datanodes, Datanode Volume Failures, Snapshot, Startup Progress, and Utilities. The main content area is titled "Overview 'localhost:9000' (✓active)".

Started:	Sun Sep 22 21:19:07 +0530 2024
Version:	3.4.0, rbd8b77f398f626bb7791783192ee7a5dfaec760
Compiled:	Mon Mar 04 12:05:00 +0530 2024 by root from (HEAD detached at release-3.4.0-RC3)
Cluster ID:	CID-566541ac-8a79-422c-9225-9596f8bb8590
Block Pool ID:	BP-1713179684-127.0.1.1-1726242910442

Summary

Security is off.
Safemode is off.

86 files and directories, 50 blocks (50 replicated blocks, 0 erasure coded block groups) = 136 total filesystem object(s).

Heap Memory used 94.28 MB of 197 MB Heap Memory. Max Heap Memory is 437.5 MB.

Non Heap Memory used 54.2 MB of 55.38 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	23.94 GB
-----------------------------	----------

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 Resource Manager web interface. The browser address bar indicates the URL is `192.168.16:8088/cluster`. The page features the Hadoop logo and a sidebar with navigation links: Cluster, About, Nodes, Node Labels, Applications, NEW, NEW SAVING, SUBMITTED, ACCEPTED, RUNNING, FINISHED, FAILED, KILLED, Scheduler, and Tools.

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running
0	0	0	0	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes
1	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Min
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCore:

Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	Finis
Showing 0 to 0 of 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:

```
vinisha@ubuntu:~$ hdfs dfs -ls /
Found 6 items
drwxr-xr-x - vinisha supergroup 0 2024-09-15 21:17 /Weatherdata
drwxr-xr-x - vinisha supergroup 0 2024-09-19 21:16 /hadoop
drwxr-xr-x - vinisha supergroup 0 2024-09-20 19:28 /tmp
drwxrwxr-x - vinisha supergroup 0 2024-09-20 19:05 /tmp1
drwxr-xr-x - vinisha supergroup 0 2024-09-20 19:06 /user
drwxr-xr-x - vinisha supergroup 0 2024-09-15 21:18 /word_count_in_python
vinisha@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 web interface with the 'Browse Directory' view selected. The browser address bar shows 'localhost:9870/explorer.html#/'. The interface has a green navigation bar with tabs: Hadoop, Overview, Datanodes, Datanode Volume Failures, Snapshot, Startup Progress, and Utilities. Below the navigation bar, the 'Browse Directory' section is active. It shows a search bar with '/' and a 'Go!' button. Below the search bar, there are icons for file operations. A table lists the contents of the directory, showing 6 entries. The table has columns for Permission, Owner, Group, Size, Last Modified, Replication, Block Size, and Name. The entries are: Weatherdata, hadoop, tmp, tmp1, user, and word_count_in_python. At the bottom, it says 'Showing 1 to 6 of 6 entries' and has 'Previous', '1', and 'Next' buttons.

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	vinisha	supergroup	0 B	Sep 15 21:17	0	0 B	Weatherdata
drwxr-xr-x	vinisha	supergroup	0 B	Sep 19 21:16	0	0 B	hadoop
drwxr-xr-x	vinisha	supergroup	0 B	Sep 20 19:28	0	0 B	tmp
drwxrwxr-x	vinisha	supergroup	0 B	Sep 20 19:05	0	0 B	tmp1
drwxr-xr-x	vinisha	supergroup	0 B	Sep 20 19:06	0	0 B	user
drwxr-xr-x	vinisha	supergroup	0 B	Sep 15 21:18	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
```

```
vinisha@ubuntu:~$ stop-all.sh
WARNING: Stopping all Apache Hadoop daemons as vinisha in 10 seconds.
WARNING: Use CTRL-C to abort.
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [ubuntu]
Stopping nodemanagers
localhost: WARNING: nodemanager did not stop gracefully after 5 seconds: Trying
to kill with kill -9
Stopping resourcemanager
vinisha@ubuntu:~$
```

Result:

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