

STEPS TO INSTALL HADOOP ON UBUNTU

1. Update and Upgrade Ubuntu

sudo apt-get update

sudo apt-get upgrade

2. Install the JDK

sudo apt install default-jdk

Check java version:

java -version

3. Add a user Hadoop

sudo adduser Hadoop

After creating the user, switch to the user.

su -hadoop

4. Install open-SSH and Generate SSH Key pair:

Do not enter the passphrase when generating keys.

apt install openssh-server openssh-client -y

ssh-keygen

Enter your directory as: `/your_home/.ssh/id_rsa`

**** Replace your home as : `/home/XYZ` where XYZ is your user**

5. Put the public keys to Authorized keys

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

6. Give necessary permissions to SSH folder and authorized key file.

chmod 700 ~/.ssh

chmod 600 ~/.ssh/authorized_keys

7. Allow firewall for SSH access.

sudo ufw allow ssh

8. Download Hadoop (Binary):

You can go to the [Apache Hadoop Site](https://hadoop.apache.org/), and download the latest stable binary version from site.

Copy the link to the binary file and use the command below:

wget https://dlcdn.apache.org/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz

9. Untar (Unzip) the downloaded file.

```
tar -xvzf hadoop-3.4.0.tar.gz
```

10. Now we move the extracted Hadoop directory to the installation folder.

```
sudo mv hadoop-3.4.0 /usr/local/Hadoop
```

11. Create a directory for storing the system logs for Hadoop

```
sudo mkdir /usr/local/hadoop/logs
```

12. Change the ownership of Hadoop Directory.

```
sudo chown -R hadoop:hadoop /usr/local/Hadoop
```

13. Configuring the environment variables for Hadoop.

```
sudo vi ~/.bashrc
```

ADD the following lines at the bottom of this file:

```
export HADOOP_HOME=/usr/local/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 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"
```

```
# edited GCC warnings and errors
#export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01;'

# some more ls aliases
alias ll='ls -lhr'
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands.  Use like so:
#   sleep 10; alert
alias alert='notify-send --urgency=low -i "${?} = 0" && echo terminal || echo error'

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
. ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
if [ -s /usr/share/bash-completion/bash_completion ]; then
. /usr/share/bash-completion/bash_completion
elif [ -f /etc/bash_completion ]; then
. /etc/bash_completion
fi
fi

export HADOOP_HOME=/usr/local/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 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.

14. Make the variables effective

source ~/.bashrc

15. Find path to Java Compiler

which javac

16. Determine the open JDK location by using the location from the output of above command.

readlink -f /usr/bin/javac

17. Copy the path found in the output of above command (upto open-jdk directory) and open the file:

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

and write the below lines at the bottom:

`export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64`

`export HADOOP_CLASSPATH+=" $HADOOP_HOME/lib/*.jar"`

Remember your JAVA_HOME variable should be the one you copied(upto open-jdk directory as shown above)

SAVE AND CLOSE THE FILE.

18. Navigate to Hadoop lib directory

cd /usr/local/hadoop/lib

19. Use the wget command to download the Javax activation file:

sudo wget <https://jcenter.bintray.com/javax/activation/javax.activation-api/1.2.0/javax.activation-api-1.2.0.jar>

20. Verify the Hadoop installation

hadoop version

You shall get output as:

```
hadoop@hadoop-iims:~$ hadoop version
Hadoop 3.4.0
Source code repository git@github.com:apache/hadoop.git -r bd8b77f398f626bb7791783192ee7a5dfaeec760
Compiled by root on 2024-03-04T06:35Z
Compiled on platform linux-x86_64
Compiled with protoc 3.21.12
From source with checksum f7fe694a3613358b38812ae9c31114e
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-3.4.0.jar
hadoop@hadoop-iims:~$
```

21. Make necessary configuration settings to specify the URL of Namenode.

```
sudo vi $HADOOP_HOME/etc/hadoop/core-site.xml
```

Add the following lines inside configuration part:

```
<configuration>

  <property>

    <name>fs.default.name</name>

    <value>hdfs://0.0.0.0:9000</value>

    <description>The default file system URI</description>

  </property>

</configuration>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed 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. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://0.0.0.0:9000</value>
    <description>The default file system URI</description>
  </property>
</configuration>
```

22. Create a directory for storing node metadata and change the ownership to Hadoop

```
sudo mkdir -p /home/hadoop/hdfs/{namenode,datanode}
```

```
sudo chown -R hadoop:hadoop /home/hadoop/hdfs
```

23. Edit the `hdfs-site.xml` configuration file to define the location for storing node metadata and the replication factor

sudo vi \$HADOOP_HOME/etc/hadoop/hdfs-site.xml and add the following lines:

Add the following lines inside configuration part:

```
<configuration>

  <property>

    <name>dfs.replication</name>

    <value>1</value>

  </property>

  <property>

    <name>dfs.name.dir</name>

    <value>file:///home/hadoop/hdfs/namenode</value>

  </property>

  <property>

    <name>dfs.data.dir</name>

    <value>file:///home/hadoop/hdfs/datanode</value>

  </property>

</configuration>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed 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. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

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

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

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

24. Edit the mapred-site.xml configuration file to define MapReduce values.

Add the following lines inside configuration part:

```
sudo vi $HADOOP_HOME/etc/hadoop/mapred-site.xml
```

And add the following lines:

```
<configuration>

  <property>

    <name>mapreduce.framework.name</name>

    <value>yarn</value>

  </property>

</configuration>
```

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed 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. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

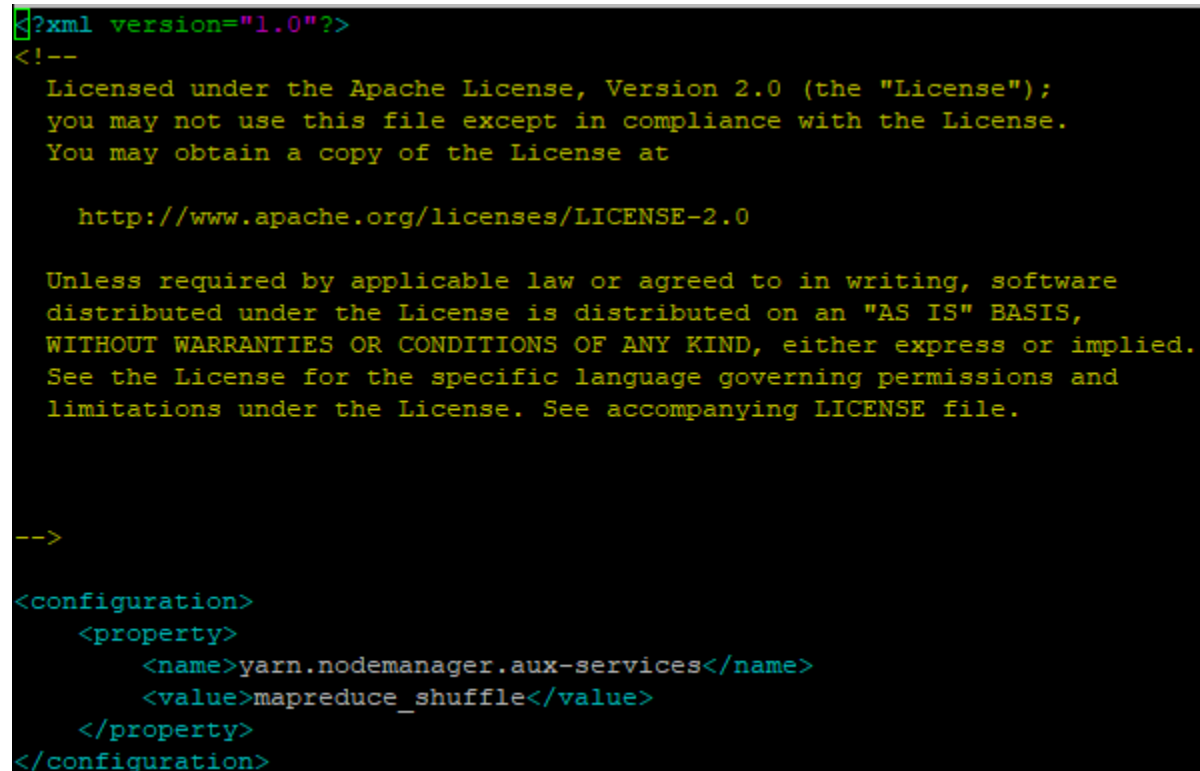
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

25. Edit the yarn-site.xml configuration file and define YARN-related settings.

sudo vi \$HADOOP_HOME/etc/hadoop/yarn-site.xml

And add the following lines:

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



```
<?xml version="1.0"?>
<!--
Licensed 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. See accompanying LICENSE file.
-->

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

26. Validate Hadoop Configuration. This command wipes out all existing data in the Namenode, effectively setting up a new HDFS file system. As a result, it also deletes the file system metadata, including files, directories, and block locations.

hdfs namenode -format

27. Start the Namenode and Datanode:

```
start-dfs.sh
```

THE OUTPUT SHOULD LOOK SOMEWHAT LIKE THIS:

```
Starting namenode on [namenode_host]... started
```

```
Starting secondarynamenode on [secondarynamenode_host]... started
```

```
Starting datanode on [datanode_host1]... started
```

```
... (similar messages for all datanodes)
```

28. Start YARN

```
start-yarn.sh
```

You should get something like:

```
starting yarn daemons
```

```
starting resourcemanager, logging to /path/to/yarn-resourcemanager.log ...  
started
```

```
starting nodemanager on [nodemanager_host1]... started
```

```
... (similar messages for all nodemanagers)
```

29. Verify all the running components

```
jps
```

You should get something like:

```
3214 SecondaryNameNode
```

```
4320 Jps
```

```
3854 Resourcemanager
```

```
3456 DataNode
```

```
4084 NodeManager
```

```
3274 NameNode
```


30. Access HDFS on command line

a. *ssh localhost*

b. Start by:

hdfs dfs -mkdir /xyz

31. You can open a Web browser to access Hadoop's NameNode (<http://localhost:9870>) and ResourceManager (<http://localhost:8088>) interfaces

The screenshot displays two web browser windows. The top window shows the Hadoop Overview page for the NameNode at 10.4.47.55:9870/dfshealth.html#tab-overview. The bottom window shows the Hadoop All App page for the ResourceManager at 10.4.47.55:8088/cluster.

Overview '0.0.0.0:9000' (✓active)

Started:	Mon Jun 10 10:01:55 +0545 2024
Version:	3.4.0, rbd8b77f398f626bb7791783192ee7a5dfaee760
Compiled:	Mon Mar 04 12:20:00 +0545 2024 by root from (HEAD detached at release-3.4.0-RC3)
Cluster ID:	CID-f0c071f2-402f-44e6-8815-3e3ba4a5f10e
Block Pool ID:	BP-862147918-127.0.1.1-1717993006969

Summary

Security is off.
 Safemode is off.
 3 files and directories, 1 blocks (1 replicated blocks, 0 erasure coded block groups) = 4 total filesystem object(s).
 Heap Memory used 97.68 MB of 137 MB Heap Memory. Max Heap Memory is 980 MB.

All App

Cluster
 About
 Nodes
 Node Labels
 Applications
 NEW
 NEW SAVING
 SUBMITTED
 ACCEPTED
 RUNNING
 FINISHED
 FAILED
 KILLED
 Scheduler
 Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Used Resources
0	0	0	0	0	<memory:0 B, vCores:0>

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes
1	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Max
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus
No data											

Showing 0 to 0 of 0 entries

****END OF MANUAL****

TO RUN MAPREDUCE JOBS:

edit the **mapred-site.xml** file:

Add following properties:

```
<property>
  <name>yarn.app.mapreduce.am.env</name>
  <value>HADOOP_MAPRED_HOME=/usr/local/hadoop</value>
</property>
```

```
<property>
  <name>mapreduce.map.env</name>
  <value>HADOOP_MAPRED_HOME=/usr/local/hadoop</value>
</property>
```

```
<property>
  <name>mapreduce.reduce.env</name>
  <value>HADOOP_MAPRED_HOME=/usr/local/hadoop</value>
</property>
```

Now for all the above 3 properties, replace the path `/usr/local/Hadoop` with your Hadoop Home Address.

Now add the 5th property :

```
5. <property> <name>mapreduce.application.classpath</name> <value></value> </property>
```

The **<value>** element must be filled with content after executing below commands (up to `yarn/*`) in your terminal:

1. `export HADOOP_CLASSPATH=$(hadoop classpath)`
2. `echo $HADOOP_CLASSPATH`

My output for above command is:

```
/usr/local/hadoop/etc/hadoop:/usr/local/hadoop/share/hadoop/common/lib/*:/usr/local/hadoop/share/hadoop/common/*:/usr/local/hadoop/share/hadoop/hdfs:/usr/local/hadoop/share/hadoop/hdfs/lib/*:/usr/local/hadoop/share/hadoop/hdfs/*:/usr/local/hadoop/share/hadoop/mapreduce/*:/usr/local/hadoop/share/hadoop/yarn:/usr/local/hadoop/share/hadoop/yarn/lib/*:/usr/local/hadoop/share/hadoop/yarn/*:/usr/local/hadoop/lib/javax.activation-api-1.2.0.jar
```

My final mapred-site.xml is:

```
<configuration>
```

```
  <property>
```

```
    <name>mapreduce.framework.name</name>
```

```
<value>yarn</value>

</property>

<property>

  <name>yarn.app.mapreduce.am.env</name>

  <value>HADOOP_MAPRED_HOME=/usr/local/hadoop</value>

</property>

<property>

  <name>mapreduce.map.env</name>

  <value>HADOOP_MAPRED_HOME=/usr/local/hadoop</value>

</property>

<property>

  <name>mapreduce.reduce.env</name>

  <value>HADOOP_MAPRED_HOME=/usr/local/hadoop</value>

</property>

<property>

  <name>mapreduce.application.classpath</name>

  <value>/usr/local/hadoop/etc/hadoop:/usr/local/hadoop/share/hadoop/common/lib/*:/usr/local/h
adoop/share/hadoop/common/*:/usr/local/hadoop/share/hadoop/hdfs:/usr/local/hadoop/share/h
adoop/hdfs/lib/*:/usr/local/hadoop/share/hadoop/hdfs/*:/usr/local/hadoop/share/hadoop/mapre
duce/*:/usr/local/hadoop/share/hadoop/yarn:/usr/local/hadoop/share/hadoop/yarn/lib/*:/usr/loc
al/hadoop/share/hadoop/yarn/*</value>

</property>

</configuration>
```

SAMPLE COMMAND:

```
hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.4.0.jar
wordcount /abhi/mysales.txt /output/part_1
```