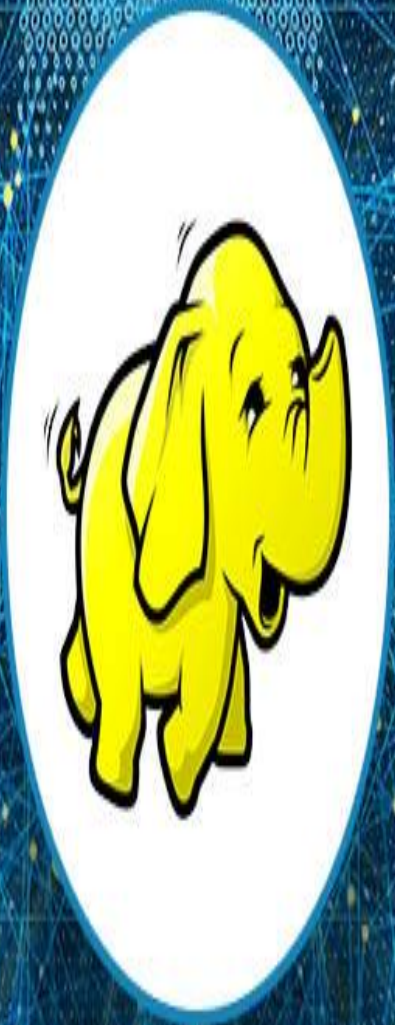
By Abdelrahman Omar



HDFS cluster installation guide

Table of content

Introduction

What is Hadoop?

Apache Hadoop is an open source framework that is used to efficiently store and process large datasets ranging in size from gigabytes to petabytes of data. Instead of using one large computer to store and process the data, Hadoop allows clustering multiple computers to analyse massive datasets in parallel more quickly.

In this document we will configure and install one node cluster, first we will show how to
Connect to EC2 instance through Putty, second we will update machine and install all
Required dependencies for installing a working node of Hadoop cluster,
Third, we will show how to configure ssh connection on EC2 instance and create client server
Relationship.
Finally, we will show how to download and configure Hadoop cluster and test hdfs commands
To assure working Hadoop cluster.

HDFS cluster installation guide

Section 1 installing required tools to connect to EC2 instance

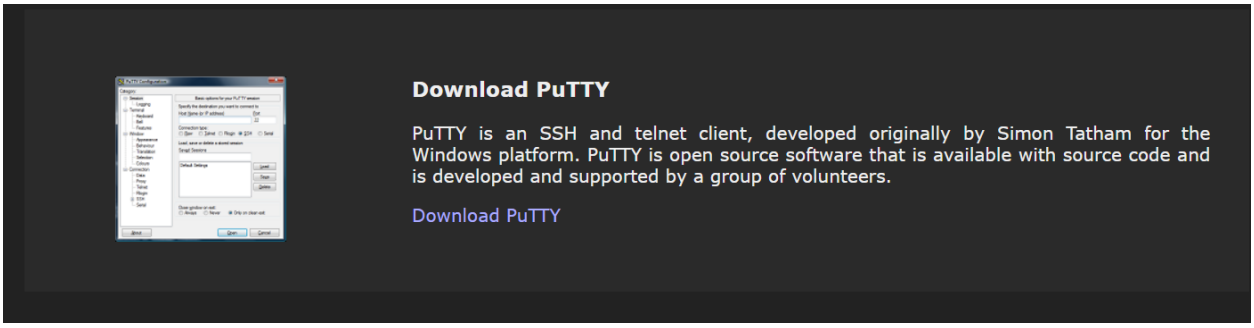
Overview

Downloading Putty and connect to EC2 instance.

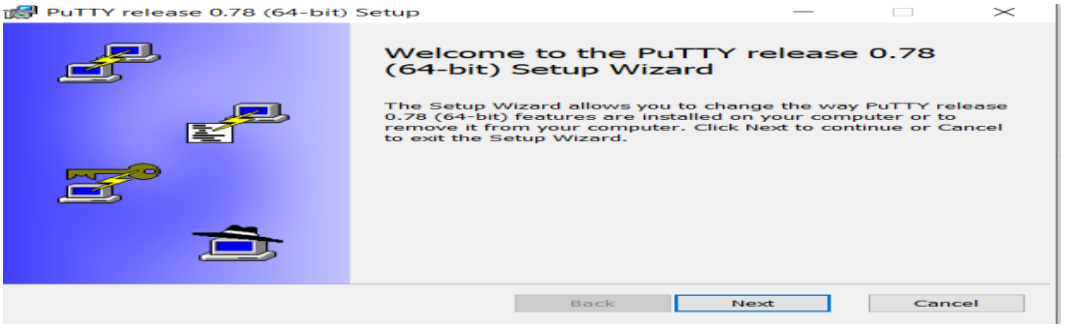
1- Download Putty

PuTTY is an SSH and telnet client to connect to Amazon EC2 instance

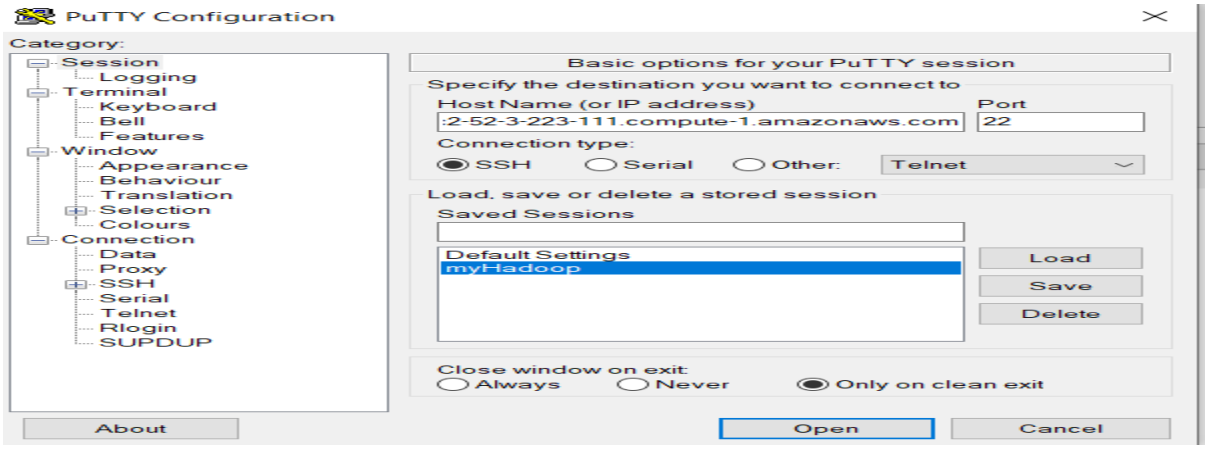
Download link



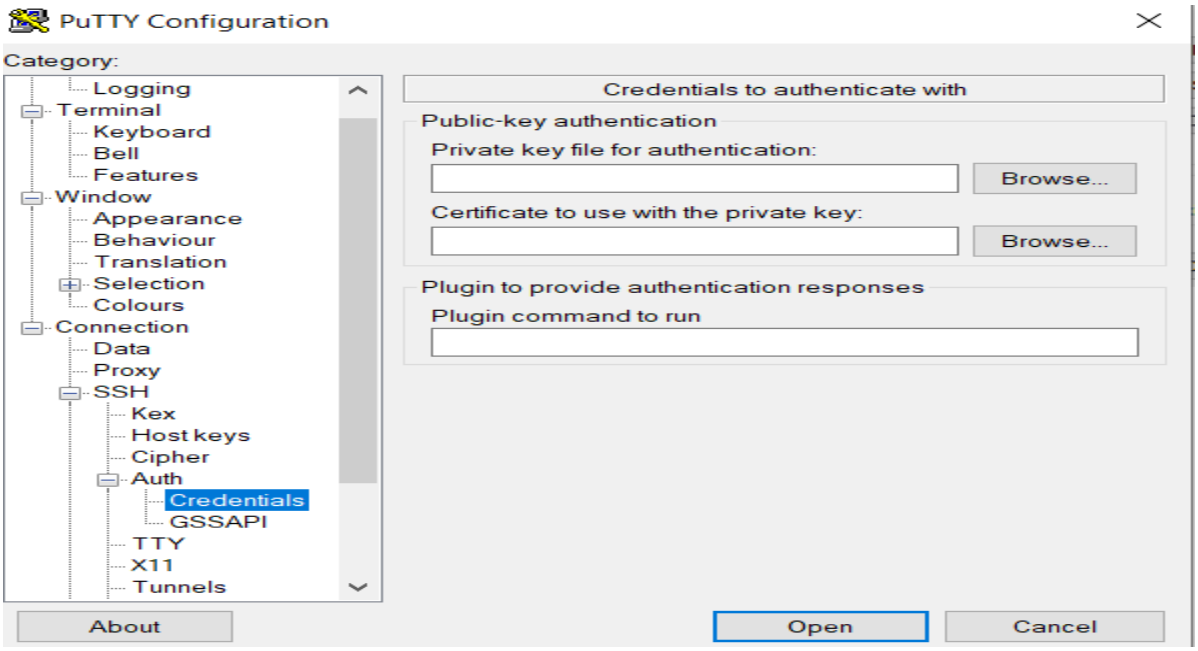
2- **Install Putty**



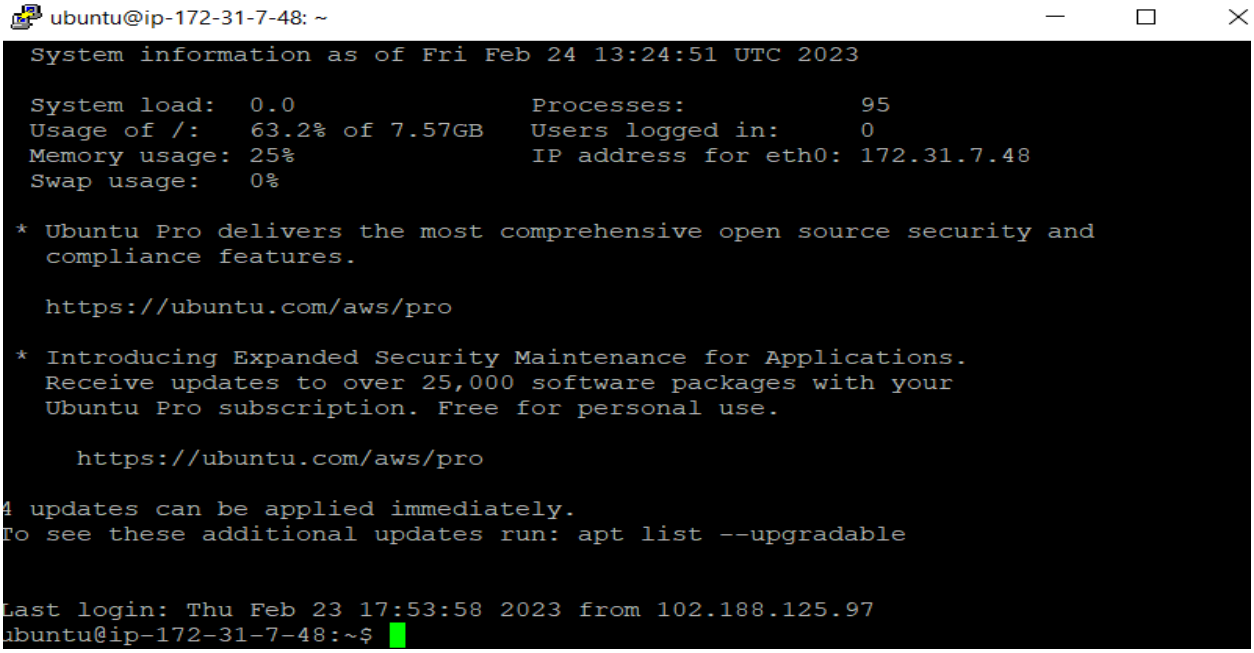
3- Open Putty and enter machine details



4- Select ssh->Auth->credentials to select key_pair for auth



5- Browse key_pair.ppk and select it and click open



After opening you are now connected to EC2 instance and can start writing commands

Section 2 install updates and dependencies

Overview

Install and updating the EC2 instance and installing JDK and JRE.

1- Execute command “sudo apt-get update”

```
ubuntu@ip-172-31-7-48:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [8
8.7 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease
[83.3 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 Packa
ges [8570 kB]
Get:6 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [256
9 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe Translation
-en [4941 kB]
Get:8 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [445
 kB]
Get:9 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Package
s [1113 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/multiverse amd64 Pa
ckages [151 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/multiverse Translat
ion-en [108 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64
Packages [2901 kB]
Get:13 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-
en [154 kB]
Get:14 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages
 [1270 kB]
Get:15 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en
 [296 kB]
Get:16 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packag
es [19.8 kB]
Get:17 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-
en [3928 B]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main Transl
ation-en [531 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted
amd64 Packages [1144 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted
Translation-en [159 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe am
d64 Packages [1882 kB]
```

2- Install java runtime environment (JRE) using command “sudo apt-get default-jre

```
ubuntu@ip-172-31-7-48:~$ sudo apt-get install default-jre
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  at-spi2-core ca-certificates-java default-jre-headless fontconfig-config
  fonts-dejavu-core fonts-dejavu-extra java-common libasound2 libasound2-data
  libatk-bridge2.0-0 libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0
  libatk1.0-data libatspi2.0-0 libavahi-client3 libavahi-common-data
  libavahi-common3 libcups2 libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau2
  libdrm-radeon1 libfontconfig1 libfontenc1 libgif7 libgl1 libgl1-mesa-dri
  libglapi-mesa libglvnd0 libglx-mesa0 libglx0 libgraphite2-3 libharfbuzz0b
  libice6 libjpeg-turbo8 libjpeg8 liblcms2-2 libllvm10 libnspr4 libnss3
  libpciaccess0 libpcsclite1 libsensors4 libsm6 libx11-xcb1 libxaw7
  libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-shape0
  libxcb-sync1 libxcomposite1 libxdamage1 libxfixes3 libxft2 libxi6
  libxinerama1 libxmu6 libxpm4 libxrandr2 libxrender1 libxshmfence1 libxt6
  libxtst6 libxv1 libxxf86dga1 libxxf86vm1 openjdk-11-jre
  openjdk-11-jre-headless x11-common x11-utils
Suggested packages:
  libasound2-plugins alsa-utils cups-common liblcms2-utils pcscd lm-sensors
  libnss-mdns fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei
  | fonts-wqy-zenhei fonts-indic mesa-utils
The following NEW packages will be installed:
  at-spi2-core ca-certificates-java default-jre default-jre-headless
  fontconfig-config fonts-dejavu-core fonts-dejavu-extra java-common
  libasound2 libasound2-data libatk-bridge2.0-0 libatk-wrapper-java
  libatk-wrapper-java-jni libatk1.0-0 libatk1.0-data libatspi2.0-0
  libavahi-client3 libavahi-common-data libavahi-common3 libcups2
  libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau2 libdrm-radeon1 libfontconfig1
  libfontenc1 libgif7 libgl1 libgl1-mesa-dri libglapi-mesa libglvnd0
  libglx-mesa0 libglx0 libgraphite2-3 libharfbuzz0b libice6 libjpeg-turbo8
  libjpeg8 liblcms2-2 libllvm10 libnspr4 libnss3 libpciaccess0 libpcsclite1
  libsensors4 libsm6 libx11-xcb1 libxaw7 libxcb-dri2-0 libxcb-dri3-0
  libxcb-glx0 libxcb-present0 libxcb-shape0 libxcb-sync1 libxcomposite1
  libxdamage1 libxfixes3 libxft2 libxi6 libxinerama1 libxmu6 libxpm4
  libxrandr2 libxrender1 libxshmfence1 libxt6 libxtst6 libxv1 libxxf86dga1
  libxxf86vm1 openjdk-11-jre openjdk-11-jre-headless x11-common x11-utils
0 upgraded, 74 newly installed, 0 to remove and 5 not upgraded.
Need to get 69.8 MB of archives.
```

3- Install JDK using command “sudo apt install openjdk-8-jdk”

```
ubuntu@ip-172-31-7-48:~$ sudo apt install openjdk-8-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  adwaita-icon-theme fontconfig gtk-update-icon-cache hicolor-icon-theme
  humanity-icon-theme libasyncns0 libcairo2 libcroco3 libdatrie1 libflac8
  libgail-common libgail18 libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-bin
  libgdk-pixbuf2.0-common libgtk2.0-0 libgtk2.0-bin libgtk2.0-common libjbig0
  libogg0 libpango-1.0-0 libpangocairo-1.0-0 libpangoft2-1.0-0 libpixman-1-0
  libpulse0 librsvg2-2 librsvg2-common libsndfile1 libthai-data libthai0
  libtiff5 libvorbis0a libvorbisenc2 libxcb-render0 libxcb-shm0 libxcursor1
  openjdk-8-jdk-headless openjdk-8-jre openjdk-8-jre-headless ubuntu-mono
Suggested packages:
  gvfs pulseaudio librsvg2-bin openjdk-8-demo openjdk-8-source visualvm
  libnss-mdns fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei
  fonts-wqy-zenhei fonts-indic
The following NEW packages will be installed:
  adwaita-icon-theme fontconfig gtk-update-icon-cache hicolor-icon-theme
  humanity-icon-theme libasyncns0 libcairo2 libcroco3 libdatrie1 libflac8
  libgail-common libgail18 libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-bin
  libgdk-pixbuf2.0-common libgtk2.0-0 libgtk2.0-bin libgtk2.0-common libjbig0
  libogg0 libpango-1.0-0 libpangocairo-1.0-0 libpangoft2-1.0-0 libpixman-1-0
  libpulse0 librsvg2-2 librsvg2-common libsndfile1 libthai-data libthai0
  libtiff5 libvorbis0a libvorbisenc2 libxcb-render0 libxcb-shm0 libxcursor1
```

Section 3 configure ssh connection

Overview

In this section we will configure ssh connection and create client server relationship.

1- **Cd .ssh/**
2- **Ssh localhost**

```
ubuntu@ip-172-31-7-48:~$ ls
ubuntu@ip-172-31-7-48:~$ cd .ssh/
ubuntu@ip-172-31-7-48:~/.ssh$ ls
authorized_keys  known_hosts
ubuntu@ip-172-31-7-48:~/.ssh$ ssh localhost
ubuntu@localhost: Permission denied (publickey).
ubuntu@ip-172-31-7-48:~/.ssh$ cat known_hosts
|1|eVtneFsgn57NyC9JIRrt5tk8+Sw=|E1ithaHFadOiGuV2MzASZH0gkXU= ecdsa-sha2-nistp256
 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBKB7KgQbaIplMNPOusvDGnxcXAo
NiqWBlgYxEuz+WJ/DfkeGtjhQrU774fwnw4Z+ZPoI6PEpCc+RUPp55q8wyVo=
ubuntu@ip-172-31-7-48:~/.ssh$ ssh 172.31.7.48
The authenticity of host '172.31.7.48 (172.31.7.48)' can't be established.
ECDSA key fingerprint is SHA256:N+1NMgbWx91Bw0vxHD/Q7w7NvdG1aKPn1qa66Seck9E.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.31.7.48' (ECDSA) to the list of known hosts.
ubuntu@172.31.7.48: Permission denied (publickey).
ubuntu@ip-172-31-7-48:~/.ssh$
```

It will return permission denied

3- **Use command “ls” to list all files and directories in ssh folder**

```
ubuntu@ip-172-31-83-145:~$ cd .ssh/
ubuntu@ip-172-31-83-145:~/.ssh$ ls
authorized_keys
ubuntu@ip-172-31-83-145:~/.ssh$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is fc:39:b2:8e:14:78:58:a3:69:03:6d:07:fb:3b:0a:31.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
Permission denied (publickey).
ubuntu@ip-172-31-83-145:~/.ssh$ ls
authorized_keys  known_hosts
ubuntu@ip-172-31-83-145:~/.ssh$ cat known_hosts
```

4- Use command “cat known_hosts”

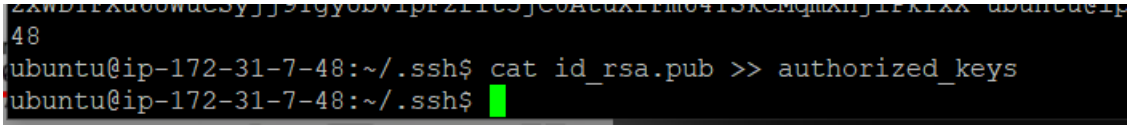
```
ubuntu@ip-172-31-83-145:~/.ssh$ cat known_hosts
|1|6n0dVXWPZAHEcyVH87VEnN8sTZM=|yzXqLWITu6ARBMDayOcdCDuBbMc= ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNT
YAAABBBNRjYe4z8m1ocW8JrP5DUmIgvqzIRlifKp6zokPAhPNq4FKnuho4OhhScyHjKaG+n0yFPeG8ED3YI6NSvLpRAcM=
ubuntu@ip-172-31-83-145:~/.ssh$
```

5-**use command “ssh-keygen “to create key to make client server relationship**

```
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.31.7.48' (ECDSA) to the list of known hosts
ubuntu@172.31.7.48: Permission denied (publickey).
ubuntu@ip-172-31-7-48:~/.ssh$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa.
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:7bqGUDKuFINJd/DWm1lMoP6b93oWKDUusxGmWNnZJY4 ubuntu@ip-172-31-7-48
The key's randomart image is:
+---[RSA 2048]----+
|    ..  ...      |
| . ...o o. .     |
|.o. .+o.=oo      |
|o o =o.E=*       |
|   +o=o+S +      |
|  ..o..= + .     |
| . . . o* . .    |
|       +o o      |
```

6-**use command “cat authorized_keys”**

7-use command “cat id_rsa.pub >> authorized_keys”



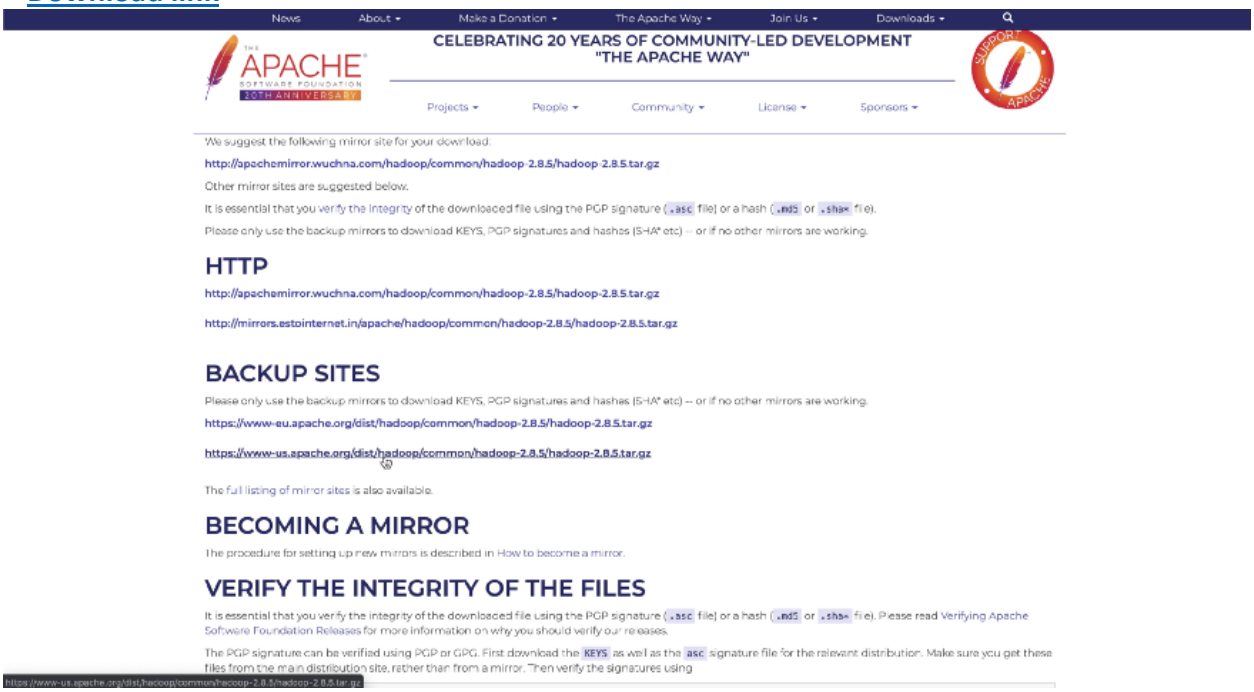
8- run command “ssh local – ssh <your machine ip> “ to test if client server created

Section 4 downloading and configuring Hadoop

Overview

In this section we will see how to download and configure Hadoop cluster
And test if hdfs commands works.

1- **Download Hadoop**
2- **Go to Hadoop -> Downloads page**
[Download link]



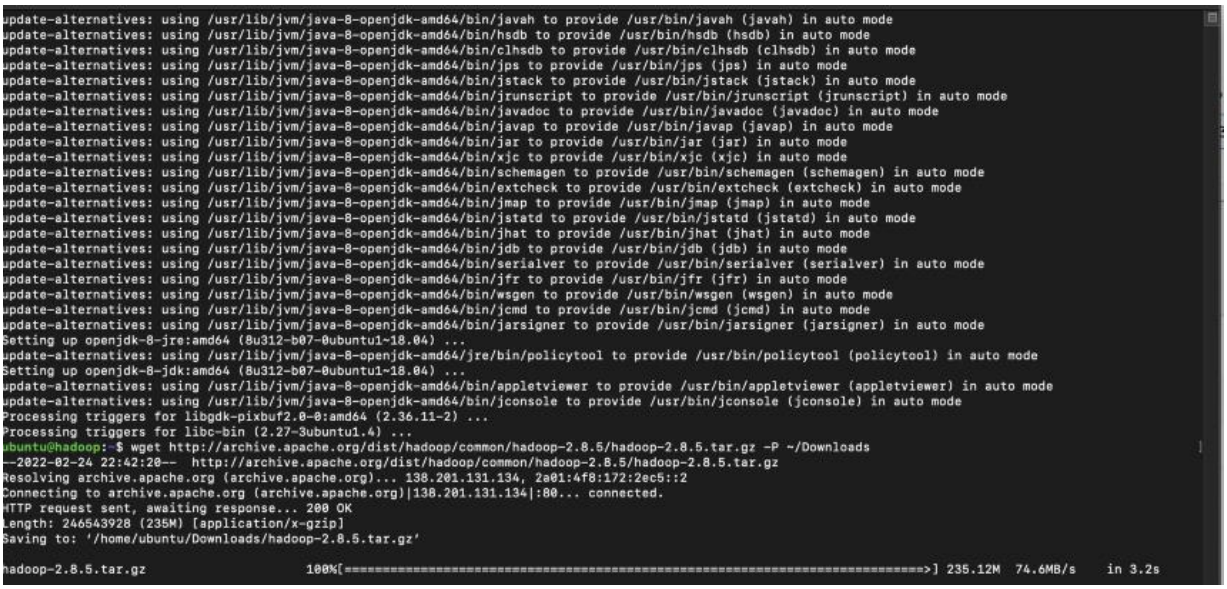
3- **Use command “wget <link> -P ~/Downloads” to download Hadoop**

```
ubuntu@ip-172-31-7-48:~$ wget https://dlcdn.apache.org/hadoop/common/hadoop-3.3.
4/hadoop-3.3.4.tar.gz
--2023-02-23 12:14:16--  https://dlcdn.apache.org/hadoop/common/hadoop-3.3.4/had
oop-3.3.4.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connecte
d.
HTTP request sent, awaiting response... 200 OK
Length: 695457782 (663M) [application/x-gzip]
Saving to: ‘hadoop-3.3.4.tar.gz’

hadoop-3.3.4.tar.gz 100%[===================>] 663.24M  58.2MB/s    in 7.9s

2023-02-23 12:14:24 (83.5 MB/s) - ‘hadoop-3.3.4.tar.gz’ saved [695457782/6954577
82]
```

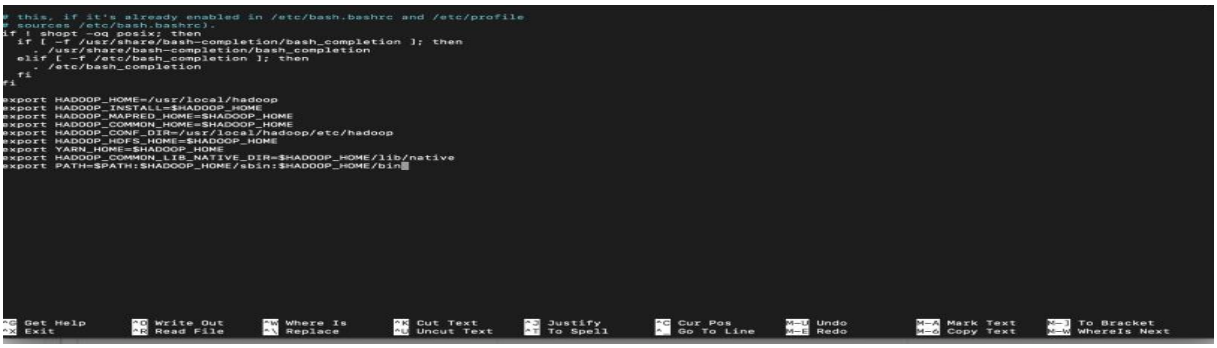
4- Use command “sudo tar zxvf ~/Downloads/hadoop-* -C /usr/local” to extract the package



5- Use command “nano ~/.bashrc” to edit the file
Write into it :

export HADOOP_HOME=/usr/local/hadoop
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_CONF_DIR=/usr/local/hadoop/etc/hadoop
export HADOOP_HDFS_HOME=$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

after writing to the file press ctrl+o->press enter -> press ctrl+x

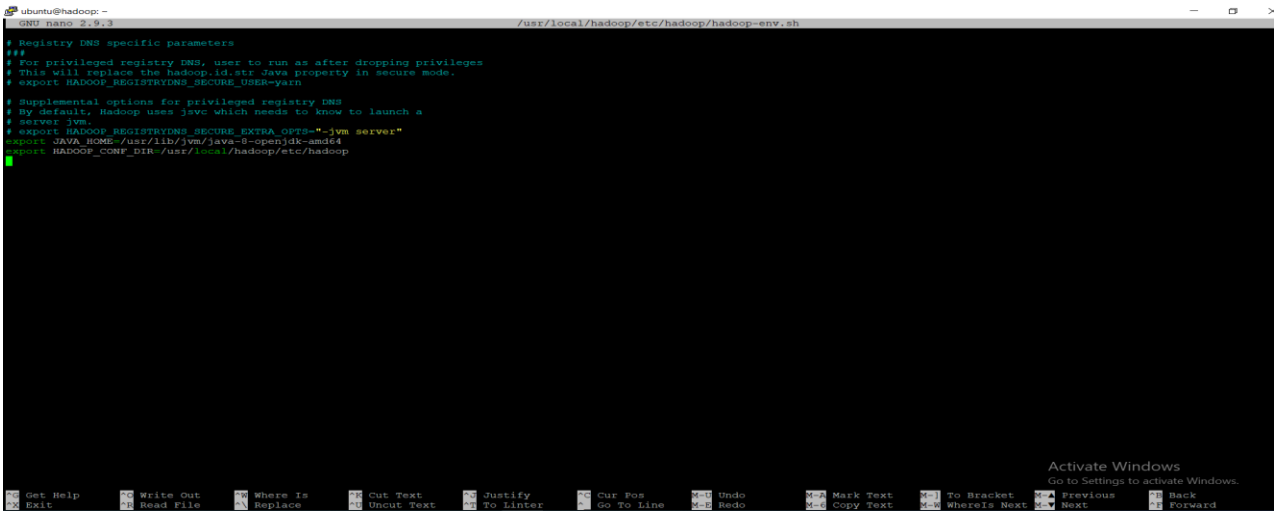


6- Use command “sudo nano $HADOOP_CONF_DIR/hadoop-env.sh” to Open hadoop-env.sh

Write into it to change hava_home and Hadoop config dir:

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_CONF_DIR=/usr/local/hadoop/etc/hadoop

after editing the file press ctrl+o->press enter->press ctrl+x



7- Use command “sudo nano $HADOOP_CONF_DIR/core-site.xml” to edit core-site.xml

Write into:

```
<configuration>
 <property>
  <name>fs.defaultFS</name>
  <value>hdfs://<your public dns name>:9000</value>
 </property>
</configuration>
```

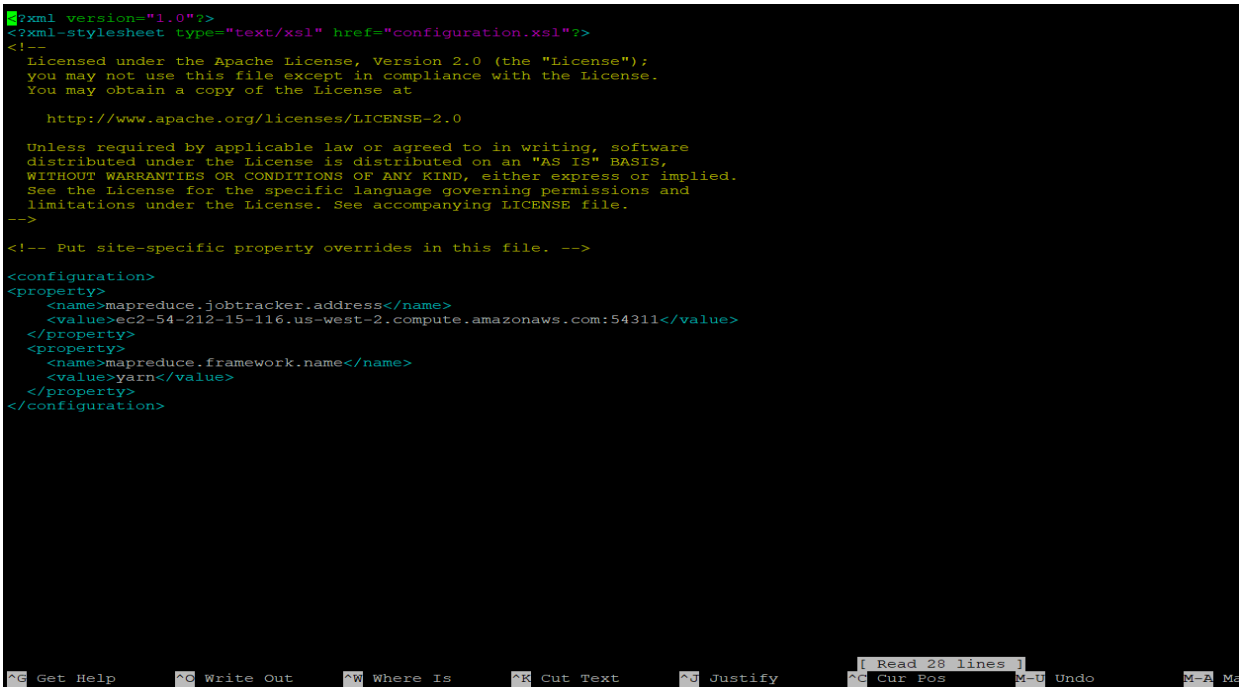
After write and edit press ctrl+o->press enter ->press ctrl+x

8- Use command "sudo nano $HADOOP_CONF_DIR/mapred-site.xml" to edit the file

Write into:

```
<configuration>
  <property>
    <name>mapreduce.jobtracker.address</name>
    <value><your public dns name>:54311</value>
  </property>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

After write and edit press ctrl+o->press enter ->press ctrl+x

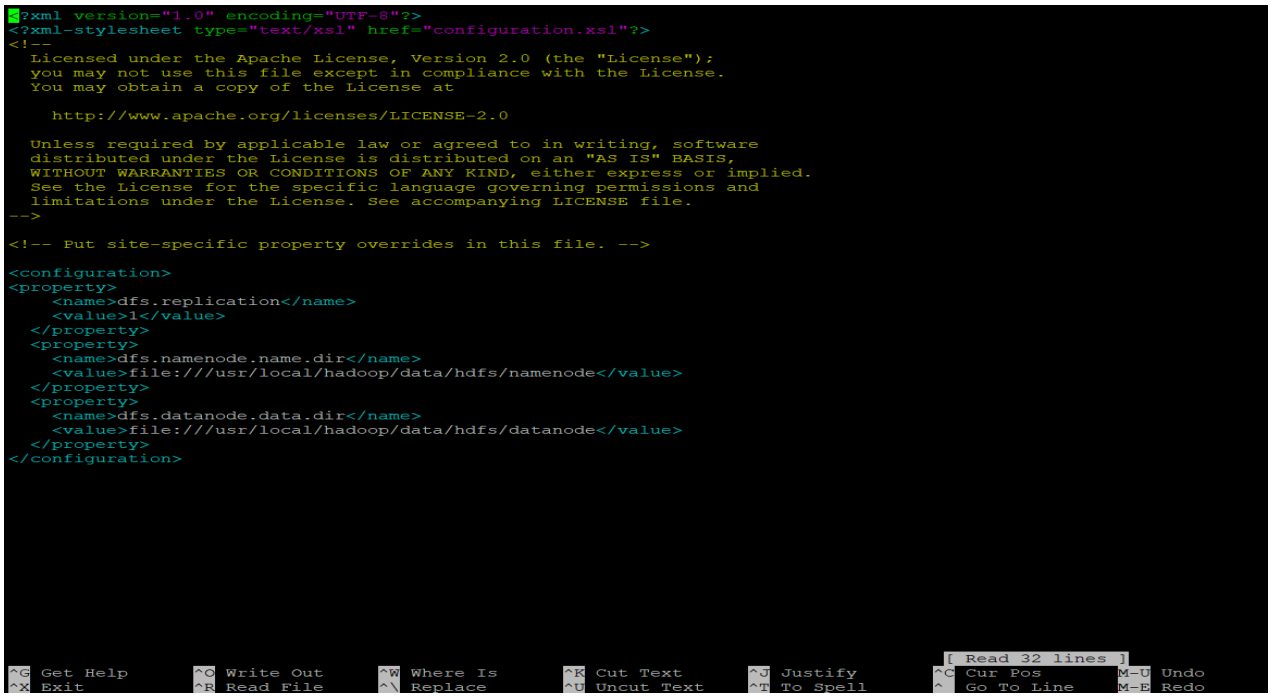


9- Use command "sudo nano $HADOOP_CONF_DIR/hdfs-site.xml"

Write into:

```
<configuration>
 <property>
  <name>dfs.replication</name>
  <value>1</value>
 </property>
 <property>
  <name>dfs.namenode.name.dir</name>
  <value>file:///usr/local/hadoop/data/hdfs/namenode</value>
 </property>
 <property>
  <name>dfs.datanode.data.dir</name>
  <value>file:///usr/local/hadoop/data/hdfs/datanode</value>
 </property>
</configuration>
```

After write and edit press ctrl+o->press enter ->press ctrl+x



10- Use command “sudo nano $HADOOP_CONF_DIR/yarn-site.xml” to edit yarn file

Write into:

```
<configuration>
 <property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
 </property>
 <property>
  <name>yarn.resourcemanager.hostname</name>
  <value><your public dns name></value>
 </property>
</configuration>
```

After write and edit press ctrl+o->press enter ->press ctrl+x

11- Use command “sudo mkdir -p $HADOOP_HOME/data/hdfs/namenode”
And command “ sudo mkdir -p $HADOOP_HOME/data/hdfs/datanode”

12- Use command “sudo chown -R ubuntu $HADOOP_HOME” to change the owner

13- Use command " sudo nano $HADOOP_HOME/masters" and command "sudo nano $HADOOP_HOME/slaves" to Change masters and slaves to Hadoop

14- Use commands " hdfs namenode -format" / "$HADOOP_HOME/sbin/start-dfs.sh" to lunch the Hadoop cluster

15- Use command "$HADOOP_HOME/sbin/start-yarn.sh" to start yarn

16- Use command "$HADOOP_HOME/sbin/mr-jobhistory-daemon.sh start historyserver" to Start the job history server

17- Use command "jps" to se java processes

Resources:

Click here for video instruction

Click here for medium article