Installing Hadoop on Ubuntu

Subject: DISTRIBUTED COMPUTING

Reference Code: IT- 717

Prepared By: Kunal J Sahitya

What is Hadoop?

- Apache Hadoop is a collection of open-source software utilities that facilitates using a network of many computers to solve problems involving massive amounts of data and computation.

What is Hadoop?

- Apache Hadoop is a collection of open-source software utilities.
- Facilitates using a network of many computers to solve problems.

- Involving massive amounts of data and computation.
  - Initial release: April 1, 2006; 15 years ago[3]

What is Hadoop?

- Hadoop was created by Doug Cutting and Mike Cafarella, the creators of Apache Lucene (the widely used text search library).

- Hadoop has its origins in Apache Nutch (an open source web search engine).

- The name Hadoop is not an acronym; it’s a made-up name.

- Doug explains “It was a name my kid gave to a stuffed yellow elephant.”

Hadoop Ecosystem Overview

5

Installing Hadoop

1. Install OpenJDK on Ubuntu. (Preferably jdk-8)
2. Set Up a User for Hadoop Environment (Preferably non-root)
3. Download and Install/Extract Hadoop on Ubuntu
4. Single Node Hadoop Deployment (Pseudo-Distributed Mode)
5. Format HDFS NameNode

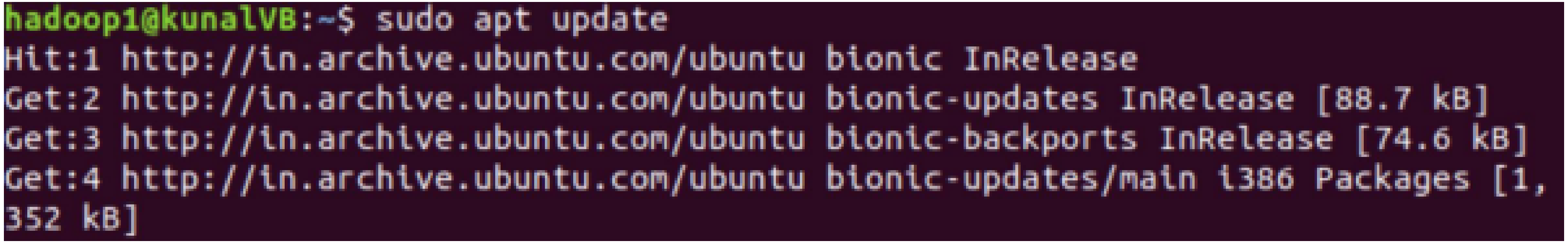
6. Start Hadoop Cluster & Access Hadoop UI from Browser 6

1. Install OpenJDK on Ubuntu. (Preferably jdk-8)

- Prerequisite for any Hadoop version to work on your system (Linux OR Windows) is java.
- Use the following command to update your system before initiating a new installation:

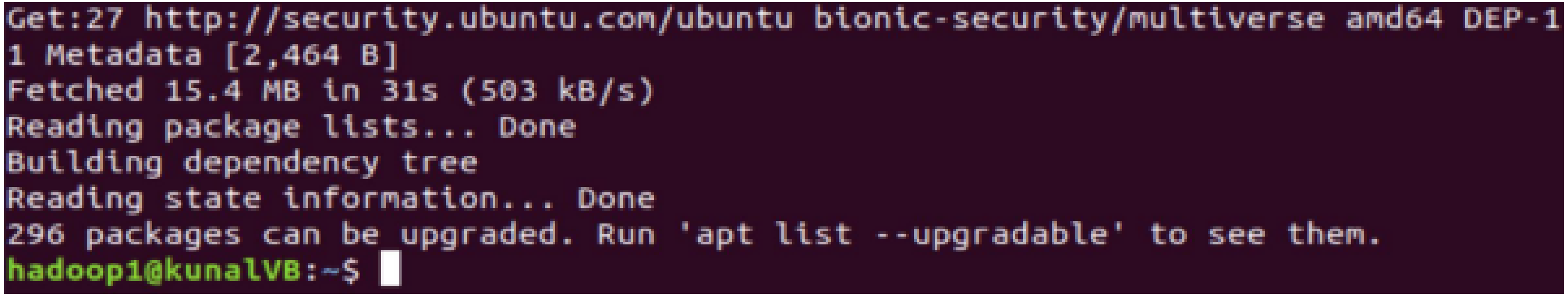
```
sudo apt update
```

Output: Update Your System



```
hadoop1@kunalVB:~$ sudo apt update
Hit:1 http://in.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://in.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu bionic-updates/main i386 Packages [1,
352 kB]
```

...



```
Get:27 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 DEP-1
1 Metadata [2,464 B]
Fetched 15.4 MB in 31s (503 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
296 packages can be upgraded. Run 'apt list --upgradable' to see them.
hadoop1@kunalVB:~$ 
```

1. Install OpenJDK on Ubuntu. (Preferably jdk-8)

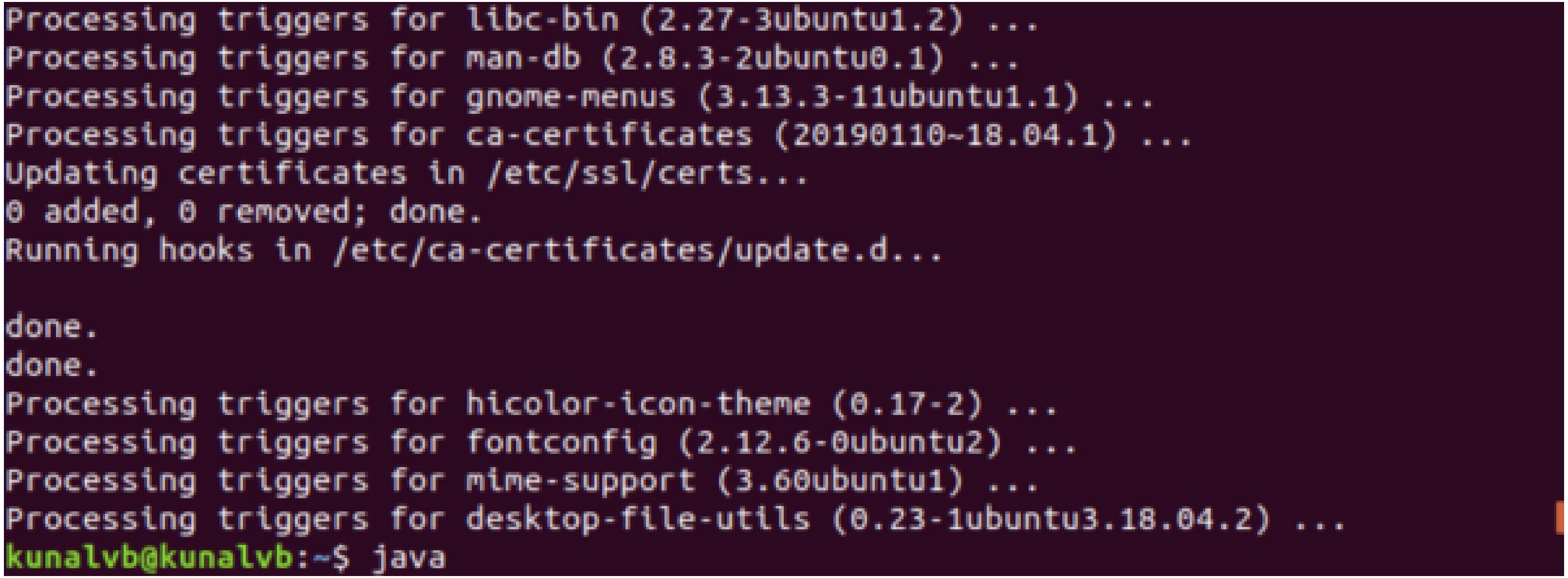
- Apache Hadoop 3.x fully supports Java 8. The OpenJDK 8 package in Ubuntu contains both the runtime environment and development kit.
- Type the following command in your terminal to install OpenJDK 8:

```
sudo apt install openjdk-8-jdk -y
```

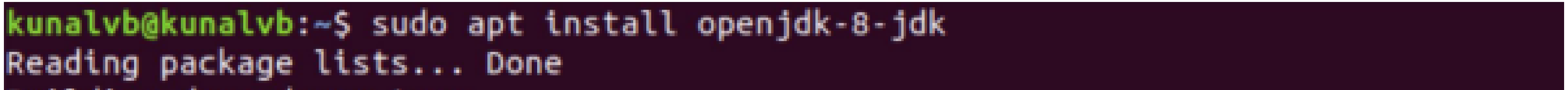
- The OpenJDK or Oracle Java version can affect how elements of a Hadoop ecosystem interact.
- Once the installation process is complete, verify the current Java version:

```
java -version; javac -version
```

Output: JDK Installation

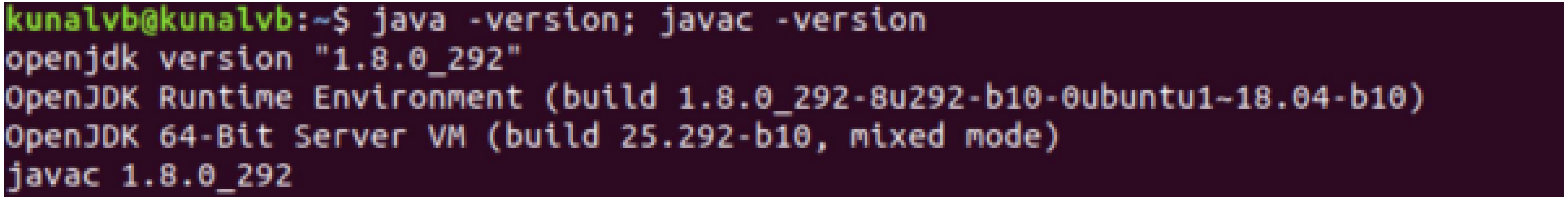


```
Processing triggers for libc-bin (2.27-3ubuntu1.2) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for gnome-menus (3.13.3-11ubuntu1.1) ...
Processing triggers for ca-certificates (20190110~18.04.1) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...

done.
done.
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for fontconfig (2.12.6-0ubuntu2) ...
Processing triggers for mime-support (3.60ubuntu1) ...
Processing triggers for desktop-file-utils (0.23-1ubuntu3.18.04.2) ...
kunalvb@kunalvb:~$ java
```



```
kunalvb@kunalvb:~$ sudo apt install openjdk-8-jdk
Reading package lists... Done
```

Output: Current JAVA Version



```
kunalvb@kunalvb:~$ java -version; javac -version
openjdk version "1.8.0_292"
OpenJDK Runtime Environment (build 1.8.0_292-8u292-b10-0ubuntu1~18.04-b10)
OpenJDK 64-Bit Server VM (build 25.292-b10, mixed mode)
javac 1.8.0_292
```

2. Create a User for Hadoop (non-root)

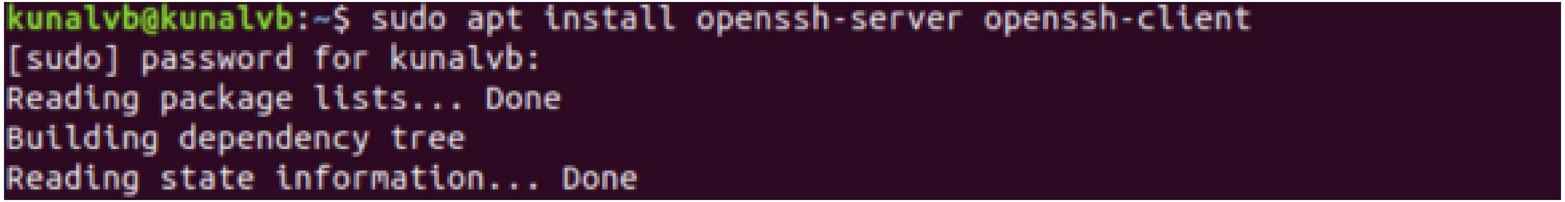
- It is advisable to create a non-root user, specifically for the Hadoop environment.
- A distinct user improves security and helps you manage your cluster more efficiently.
- To ensure the smooth functioning of Hadoop services, the user should have the ability to establish a passwordless SSH

connection with the localhost.

- Install the OpenSSH server and client using the following command:

sudo apt install openssh-server openssh-client

Output: Installing OPEN-SSH Server



```
kunalvb@kunalvb:~$ sudo apt install openssh-server openssh-client
[sudo] password for kunalvb:
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

...

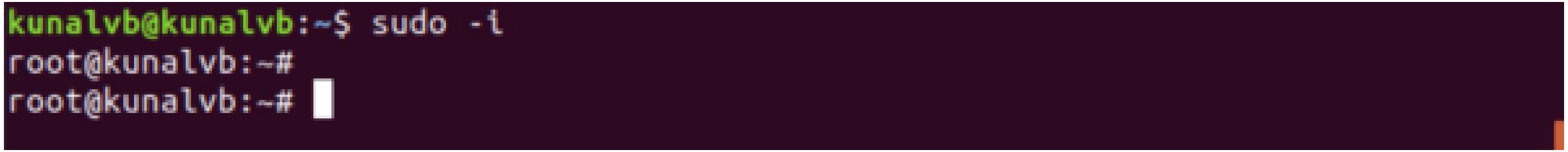
Create Hadoop User

- To add the user for Hadoop environment switch to the root user in your current terminal using following command: **sudo -i**
  - After that utilize the **adduser** command to create a new

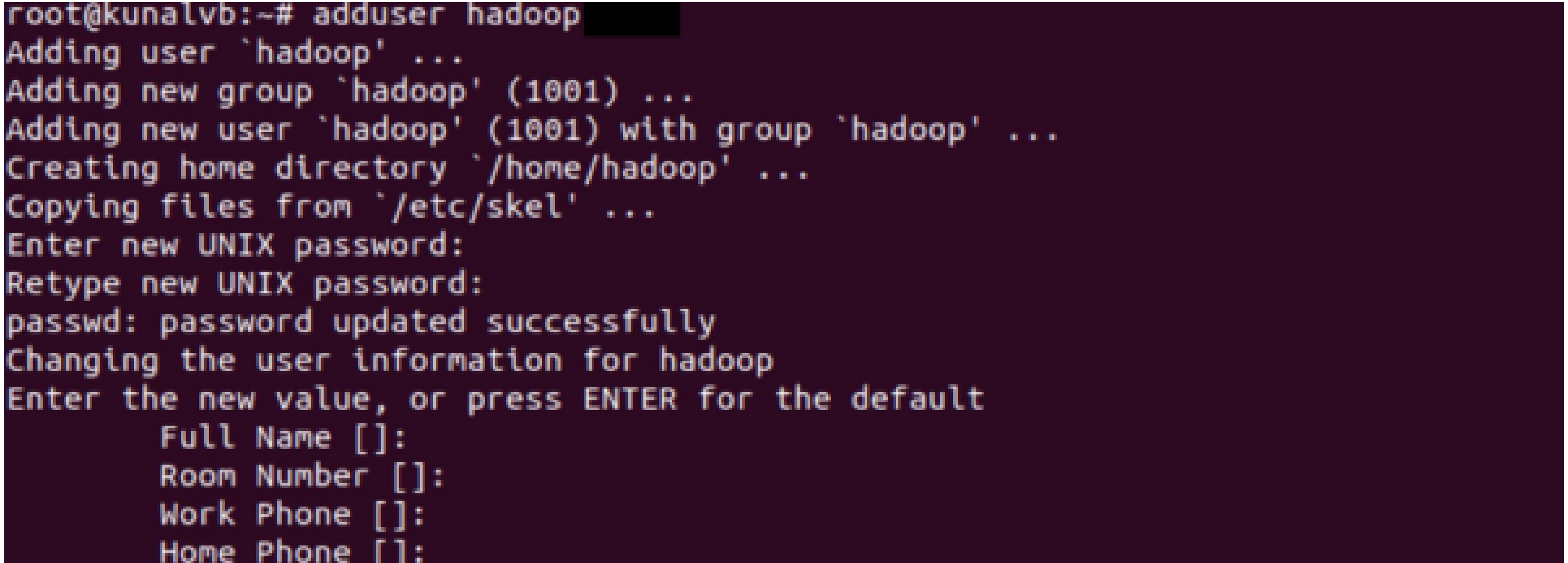
    Hadoop user:

sudo **adduser hadoop**

Output: Switching to root user



```
kunalvb@kunalvb:~$ sudo -i
root@kunalvb:~#
root@kunalvb:~# 
```



```
root@kunalvb:~# adduser hadoop [redacted]
Adding user `hadoop' ...
Adding new group `hadoop' (1001) ...
Adding new user `hadoop' (1001) with group `hadoop' ...
Creating home directory `/home/hadoop' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for hadoop
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
```

Add user into sudoers list

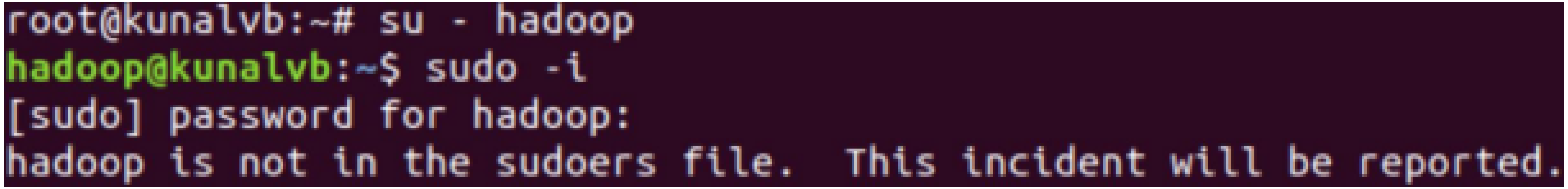
- Add the hadoop user in the sudoers list.

- Means we are adding hadoop environment user in the list of trusted users.
- Use following command to open the sudoers file kept inside

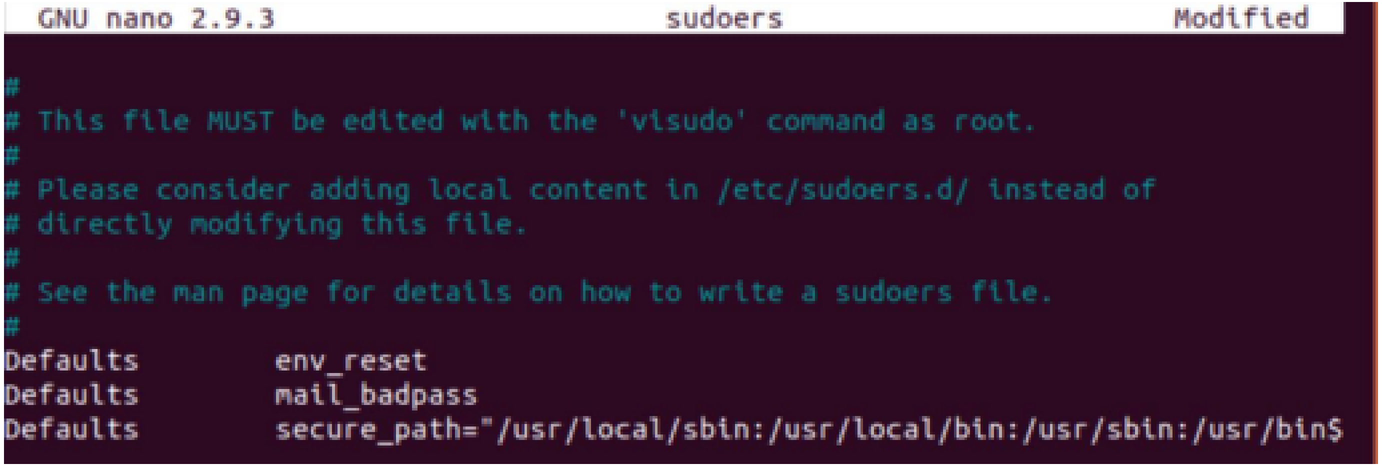
  etc directory:

```
nano /etc/sudoers
```

Output: Hadoop is not the sudoers

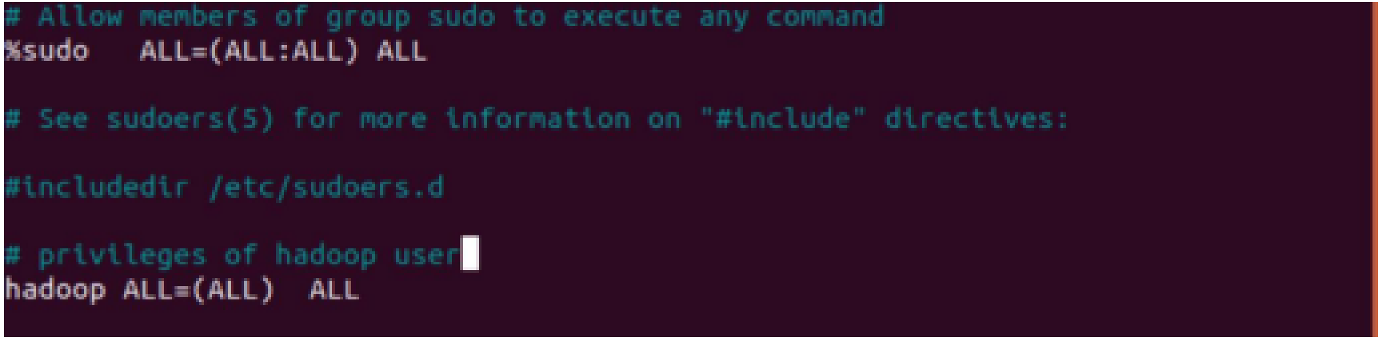


```
root@kunalvb:~# su - hadoop
hadoop@kunalvb:~$ sudo -i
[sudo] password for hadoop:
hadoop is not in the sudoers file.  This incident will be reported.
```



```
  GNU nano 2.9.3                      sudoers                      Modified

#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin$
```

...



```
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d

# privileges of hadoop user
hadoop ALL=(ALL)  ALL
```

Enable Passwordless SSH

- Generate an SSH key pair and define the location is is to be stored in:

  ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa

- Use the cat command to store the public key as authorized_keys in the ssh directory:

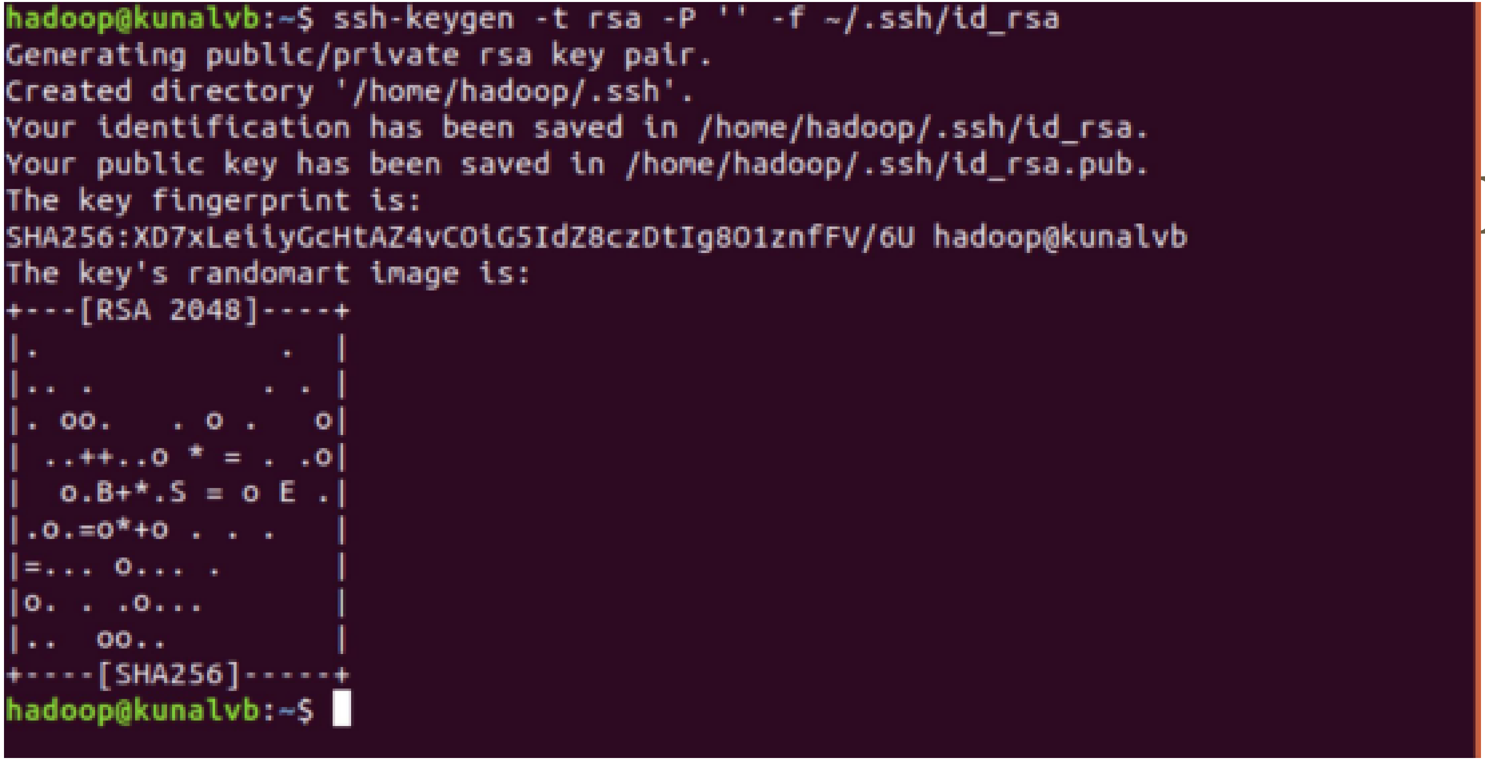
```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

- Set the permissions for your user with the chmod command:

```
chmod 0600 ~/.ssh/authorized_keys
```

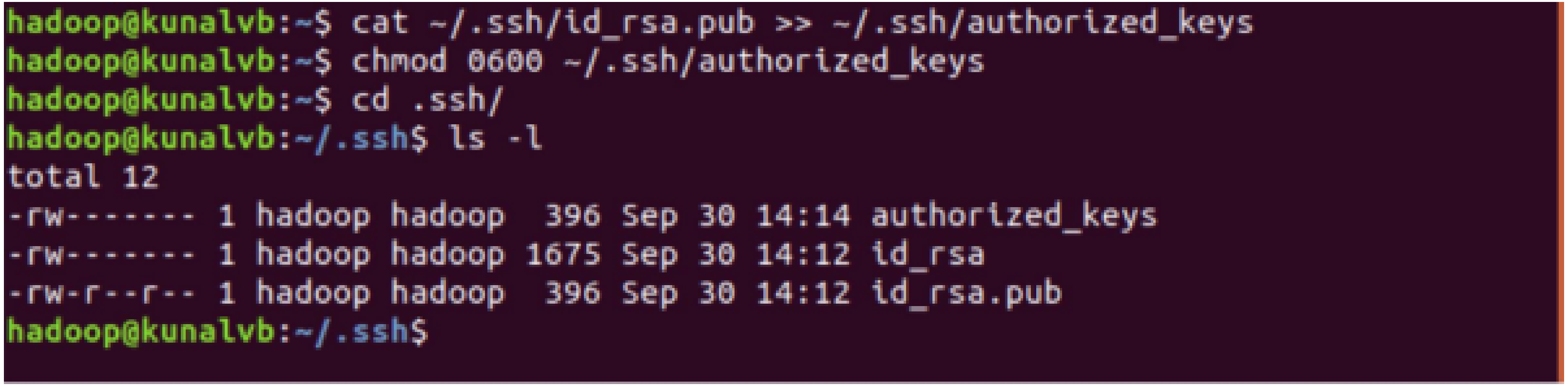
  - The new user is now able to SSH without needing to enter a password every time. Verify everything is set up correctly by using the hdoop user to SSH to localhost:

n



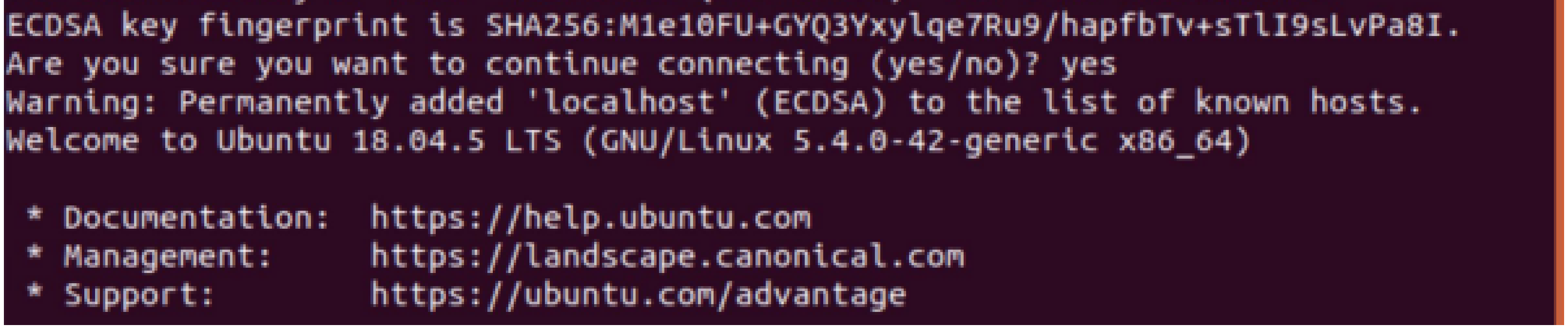
```
hadoop@kunalvb:~$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
Generating public/private rsa key pair.
Created directory '/home/hadoop/.ssh'.
Your identification has been saved in /home/hadoop/.ssh/id_rsa.
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:XD7xLeiiyGcHtAZ4vCOiG5IdZ8czDtIg8O1znfFV/6U hadoop@kunalvb
The key's randomart image is:
+---[RSA 2048]----+
|.             .  |
|.. .         . . |
|. oo.   . o .   o|
| ..++..o * = . .o|
|  o.B+*.S = o E .|
|.o.=o*+o . . .   |
|=... o... .      |
|o. . .o...       |
|..  oo..         |
+----[SHA256]-----+
hadoop@kunalvb:~$ 
```

Output1: Store the public key Output2: Set & Check the permission for user



```
hadoop@kunalvb:~$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
hadoop@kunalvb:~$ chmod 0600 ~/.ssh/authorized_keys
hadoop@kunalvb:~$ cd .ssh/
hadoop@kunalvb:~/.ssh$ ls -l
total 12
-rw------- 1 hadoop hadoop  396 Sep 30 14:14 authorized_keys
-rw------- 1 hadoop hadoop 1675 Sep 30 14:12 id_rsa
-rw-r--r-- 1 hadoop hadoop  396 Sep 30 14:12 id_rsa.pub
hadoop@kunalvb:~/.ssh$
```

Output: Connect to localhost using ssh



```
ECDSA key fingerprint is SHA256:M1e10FU+GYQ3Yxylqe7Ru9/hapfbTv+sTlI9sLvPa8I.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-42-generic x86_64)

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

...

3. Downloading Hadoop 3.2.1

- Download Hadoop from official apache site or using following link: https://hadoop.apache.org/release/3.2.1.html
- Once the download is complete, extract the files to initiate the Hadoop installation:

```
tar xzf hadoop-3.2.1.tar.gz
```

- The Hadoop binary files are now located within the hadoop 3.2.1 directory.

Output: Download Hadoop 3.2.1

r

U

D

D

|

T

I

,

S

J

K

©

4. Single Node Hadoop Deployment

- Hadoop **excels** when deployed in a **fully distributed mode** on a large cluster of networked servers.
- However, if you are new to Hadoop and want to **explore basic**

commands or test applications, you can **configure Hadoop on a single node**.

- This setup, also called **pseudo-distributed mode**, allows each Hadoop daemon to run as a single Java process.

Set of Configuration Files

- A Hadoop environment is configured by editing a set of configuration files:
- **bashrc**
- **hadoop-env.sh**

- **core-site.xml**

- **hdfs-site.xml**

- **mapred-site-xml**
- **yarn-site.xml**

Configure Hadoop EnvironmentVariables (bashrc)

- Edit the .bashrc shell configuration file using a text editor of your choice (we will be using nano):

```
sudo nano .bashrc
```

- Add the given content (in upcoming slide) to .bashrc file.
- Once you add the variables, save and exit the .bashrc file.
- It is vital to apply the changes to the current running

  environment by using the following command:

  source ~/.bashrc

Configuring .bashrc file

```
#Hadoop Related Options
export HADOOP_HOME=/home/hadoop/hadoop-3.2.1
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" 31
```

Output: Editing .bashrc file using ‘nano’

U

D

D

|

T

I

,

S

J

K

©

Output: Apply the changes to current running environment

Edit hadoop-env.sh File

• The hadoop-env.sh file serves as a master file to configure YARN, HDFS, MapReduce, and Hadoop-related project settings. • When setting up a single node Hadoop cluster, you need to define which Java implementation is to be utilized.

```
sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
```

- Uncomment the $JAVA_HOME variable and add the full path to the OpenJDK installation on your system.

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

Output: Setting JAVA_HOME Variable

Finding JAVA installation in your system

- If you need help to locate the correct Java path, run the following command in your terminal window:

```
which javac
```

- The resulting output provides the path to the Java binary directory.

- Use the provided path to find the OpenJDK directory with the following command:

```
readlink -f /usr/bin/javac
```

- The section of the path just before the /bin/javac directory needs to be assigned to the $JAVA_HOME variable.

Output: Locating Java

Edit core-site.xml File

- The core-site.xml file defines HDFS and Hadoop core properties.
- To set up Hadoop in a **pseudo-distributed mode**, you need to specify the **URL for your NameNode**, and the temporary directory Hadoop uses for the map and reduce process.

- Open the core-site.xml file in a text editor:

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

- Add the given configuration (Upcoming Slide) to override the default values for the temporary directory.

- Add your HDFS URL to replace the default local file system setting.

Edit core-site.xml File

```
<configuration>
<property>
```

```
  <name>hadoop.tmp.dir</name>
  <value>/home/hadoop/tmpdata</value>
</property>

<property>

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

  <value>hdfs://127.0.0.1:9000</value>
</property>
</configuration>
```

File

Edit hdfs-site.xml File

- The properties in the hdfs-site.xml file govern the location for storing node metadata, fsimage file, and edit log file.
- Configure the file by defining the NameNode and DataNode storage directories.

Edit hdfs-site.xml File

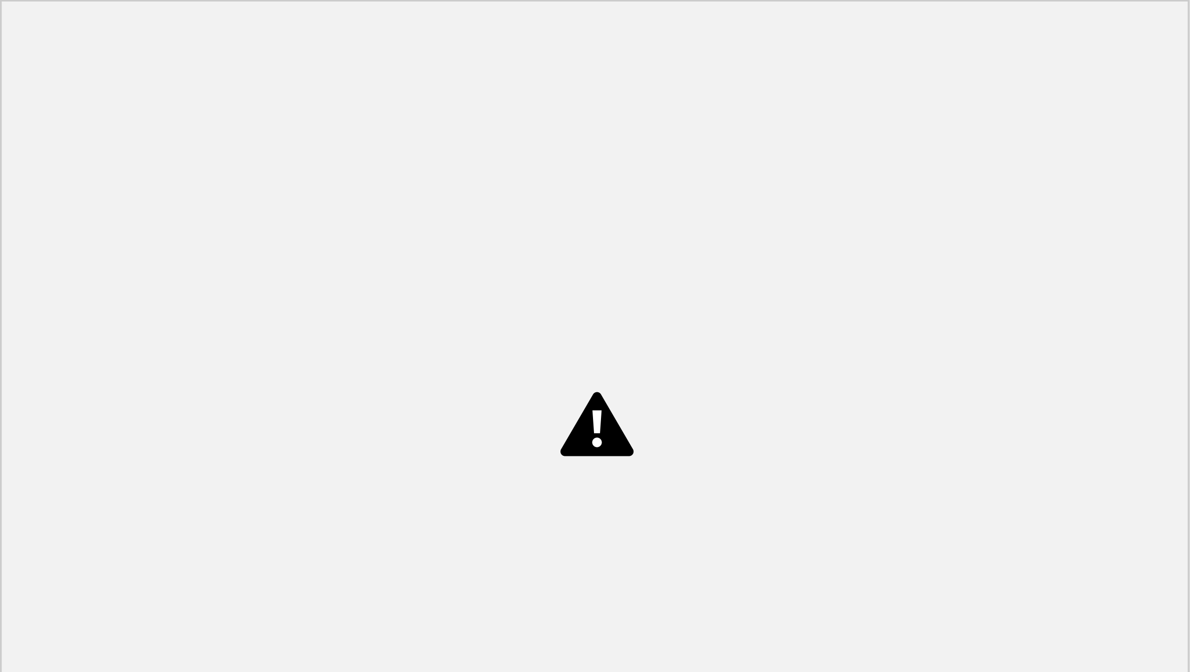
- Additionally, the default dfs.replication value of 3 needs to be changed to 1 to match the single node setup.
- Use the following command to open the hdfs-site.xml file for editing:

```
sudo nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml
```

```
<configuration>
    <property>
          <name>dfs.data.dir</name>
          <value>/home/hadoop/dfsdata/namenode</value
    > </property>

    <property>

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

          <value>/home/hadoop/dfsdata/datanode</value>

    </property>
```

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

File



Edit mapred-site.xml File

- Use the following command to access the mapred-site.xml file and define MapReduce values:

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

- Add the given configuration (Upcoming Slide) to change the default MapReduce framework name value to yarn.

Edit mapred-site.xml File

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


   </property>
```

```
</configuration>
```

ml File

Edit yarn-site.xml File

- The yarn-site.xml file is used to define settings relevant to YARN. It contains configurations for the Node Manager, Resource Manager, Containers, and Application Master.

- Open the yarn-site.xml file in a text editor:

sudo nano $HADOOP_HOME/etc/hadoop/yarn-site.xml

- Append the given configuration (Upcoming Slide) to the file.

```
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler
</value>
</property>
<property>
<name>yarn.resourcemanager.hostname</name>
<value>127.0.0.1</value>
</property>
<property>
<name>yarn.acl.enable</name>
<value>0</value>
```

Edit File yarn-site.xml

```
</property>
<property>
<name>yarn.nodemanager.env-whitelist</name>
```

```
<value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PERPEND
_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
```

l File

5. Format HDFS NameNode

- It is important to format the NameNode before starting Hadoop services for the first time:

```
hdfs namenode –format
```

- The shutdown notification signifies the end of the NameNode format process.

Output: Format Namenode

...

6. Start Hadoop Cluster

- Navigate to the hadoop-3.2.1/sbin directory and execute the following commands to start the NameNode and DataNode: **./start-dfs.sh**
  - The system takes a few moments to initiate the necessary nodes.

- Once the namenode, datanodes, and secondary namenode are up and running, start the YARN resource and nodemanagers by typing:

```
./start-yarn.sh
```

- As with the previous command, the output informs you that the processes are starting.

Start Hadoop Cluster

- Type this simple command to check if all the daemons are active and running as Java processes:

```
jps
```

- If everything is working as intended, the resulting list of

  running Java processes contains all the HDFS and YARN

  daemons.

fs daemon

Access Hadoop UI from Browser

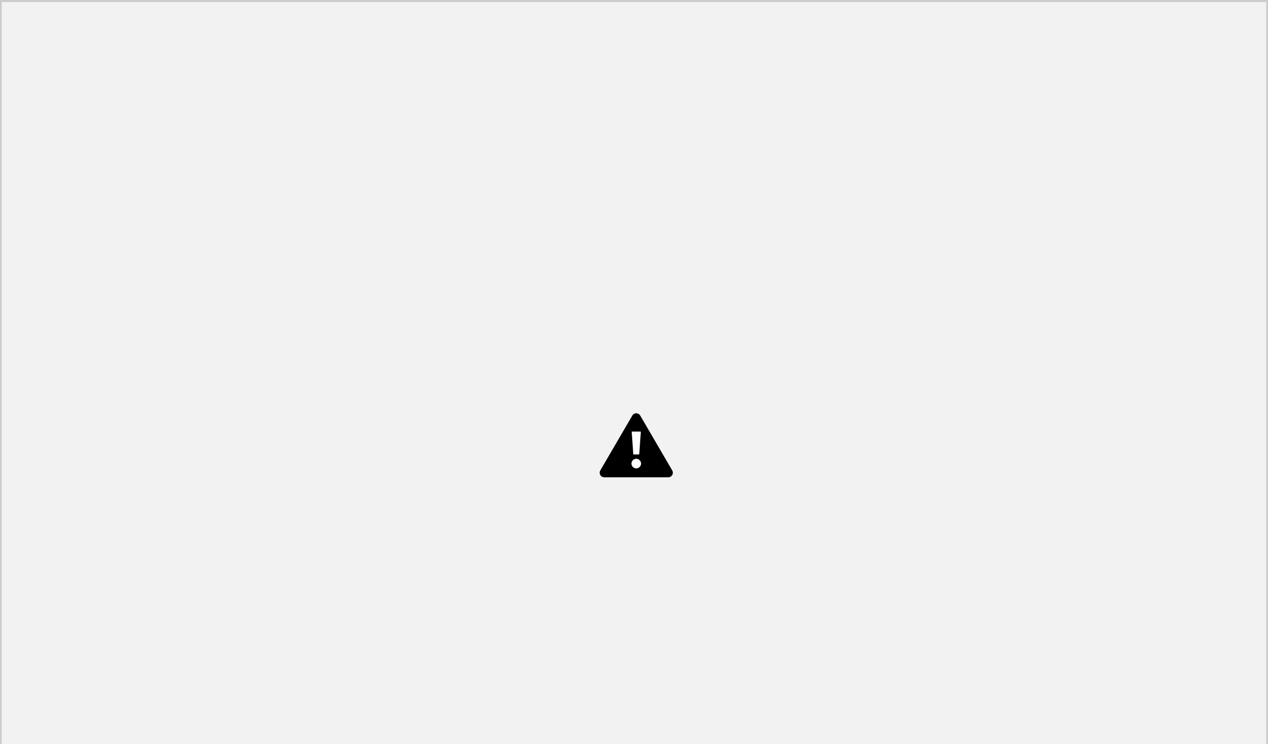
- Use your preferred browser and navigate to your localhost URL or IP.

- The default port number 9870 gives you access to the Hadoop NameNode UI: http://localhost:9870

- The default port 9864 is used to access individual DataNodes

  directly from your browser: http://localhost:9864

- The YARN Resource Manager is accessible on port 8088: http://localhost:8088 web



web UI

Output: YARN From web UI

s

U
D
D
T
I
,
S
J
K
©

Conclusion

- You have successfully installed Hadoop on Ubuntu and deployed it in a pseudo-distributed mode.
- A single node Hadoop deployment is an excellent starting point to explore basic HDFS commands

- You can also acquire the experience you need to design a fully distributed Hadoop cluster.

THANK YOU!