**Installing Hadoop on Ubuntu: A Step-by-Step Guide**

In this blog, discover how to set up and install Hadoop on an Ubuntu. Hadoop is a potent open-source framework for distributed data processing and storing. You may use the power of Hadoop, a large data processing platform, on your Ubuntu system by following the instructions in this guide.

**Step 1: Install Java Development Kit (JDK):**

***sudo apt update && sudo apt install openjdk-8-jdk***





There will be question asked in the process whether to continue, give “Y”.

**Step 2: Verify the Java version:**

***java -version***



**Step 3: Install SSH:**

Secure Shell (SSH) is essential to Hadoop’s operation since it secures communication between cluster nodes. In order to facilitate effective distributed data processing throughout the cluster, this step is essential for preserving data confidentiality and integrity. SSH can be installed by using:

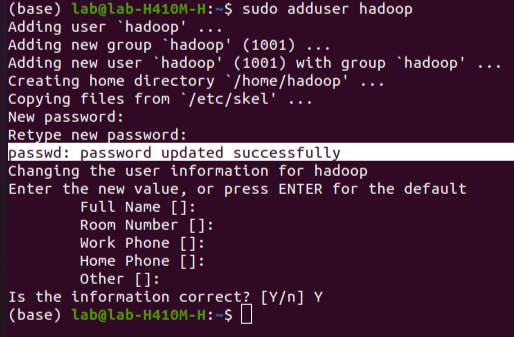
***sudo apt install ssh***

**Step 4: Create the Hadoop user:**

In order to manage Hadoop components efficiently, it’s recommended to create a dedicated user specifically for Hadoop operations.

This user will be responsible for running Hadoop services and accessing Hadoop’s web interfaces. Use the following command to create the ‘hadoop’ user and set up a password:

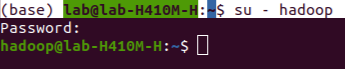
***sudo adduser hadoop***



**Step 5: Switch user:**

Once the ‘hadoop’ user is created, switch to this user to perform subsequent Hadoop installation and configuration tasks. Use the following command to switch users and operate within the ‘hadoop’ user context:

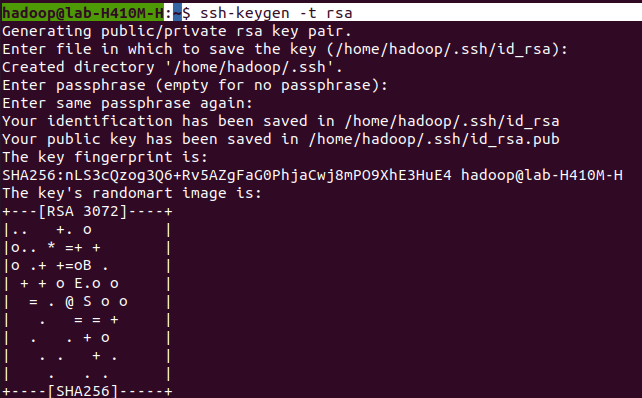
***su — hadoop***



**Step 6: Configure SSH:**

To enable password-less SSH access for the ‘hadoop’ user, you’ll need to generate an SSH keypair. Begin by generating an RSA keypair by executing the following command:

***ssh-keygen -t rsa***



**Step 7: Set permissions:**

Once the keypair is generated, you’ll need to set proper permissions and add the generated public key to the authorized keys file. This ensures secure access without the need for password authentication. Use these commands:

***cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys***

***chmod 640 ~/.ssh/authorized\_keys***

**Step 8: SSH to the localhost:**

Next, it’s important to test the SSH configuration by initiating an SSH connection to the localhost. This action allows the system to authenticate the localhost by adding RSA keys to known hosts. Execute the following command:

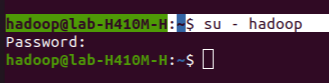
***ssh localhost***

Upon connection, you might be prompted to authenticate the localhost by typing ‘yes’ and pressing ‘Enter’.

**Step 9: Switch user:**

After configuring SSH and verifying the connection, switch back to the ‘hadoop’ user to continue the setup process using the following command:

***su — hadoop***



**Step 10: Install Hadoop:**

Start by downloading Hadoop version 3.3.6 from the Apache Hadoop repository using the wget command:

***wget***[***https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz***](https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz)

Once the download is complete, extract the contents from the downloaded file using the following command:

***tar -xvzf hadoop-3.3.6.tar.gz***

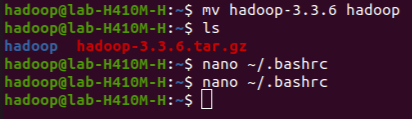


For easier reference, you might consider renaming the extracted folder to something simpler. This step is optional but can be useful for clarity:

***mv hadoop-3.3.6 hadoop***

Next, you’ll need to set up the necessary environment variables for Java and Hadoop. Open the ~/.bashrc file using a text editor. For instance, you can use ‘nano’ as the text editor:

***nano ~/.bashrc***



Add the following lines at the end of the file to set up the environment variables. to pasting the code we use ctrl+shift+v for saving the file ctrl+x and ctrl+y ,then hit enter:

***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***

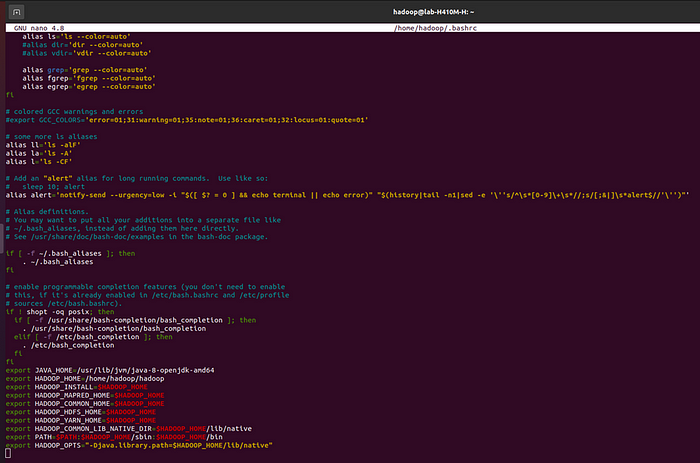
***export HADOOP\_HDFS\_HOME=$HADOOP\_HOME***

***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”***

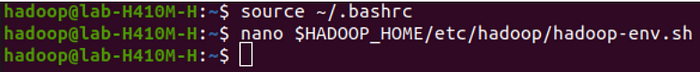


After updating the ~/.bashrc file, load the changes to the current environment using the following command:

***source ~/.bashrc***

Additionally, configure the JAVA\_HOME variable in the hadoop-env.sh file. Open the Hadoop environment configuration file with a text editor:

***nano $HADOOP\_HOME/etc/hadoop/hadoop-env.sh***



Locate the line that starts with “export JAVA\_HOME” within the file and set it to the appropriate Java path:

*export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64cmd*



**Step 11: Configuring Hadoop:**

First, navigate to the Hadoop directory to create the necessary directories for the NameNode and DataNode within the Hadoop user’s home directory:

*cd hadoop/*

*mkdir -p ~/hadoopdata/hdfs/{namenode,datanode}*

**Edit core-site.xml:**

*nano $HADOOP\_HOME/etc/hadoop/core-site.xml*



Edit the following section in the file to reflect your system’s hostname:

***<configuration>***

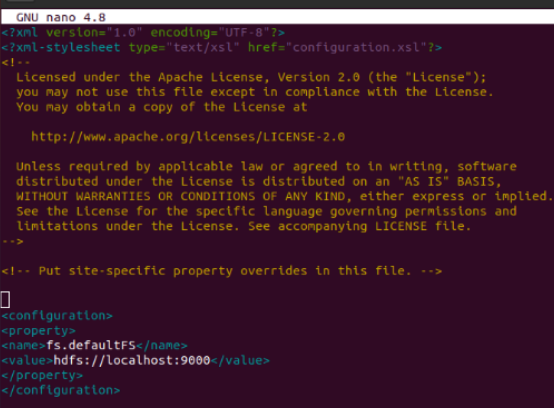
***<property>***

***<name>fs.defaultFS</name>***

***<value>hdfs://your\_system\_hostname:9000</value>***

***</property>***

***</configuration>***



**Edit hdfs-site.xml:**

Proceed to edit the hdfs-site.xml file to set the directory paths for the NameNode and DataNode:

*nano $HADOOP\_HOME/etc/hadoop/hdfs-site.xml*



**Update the settings in the file 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>***

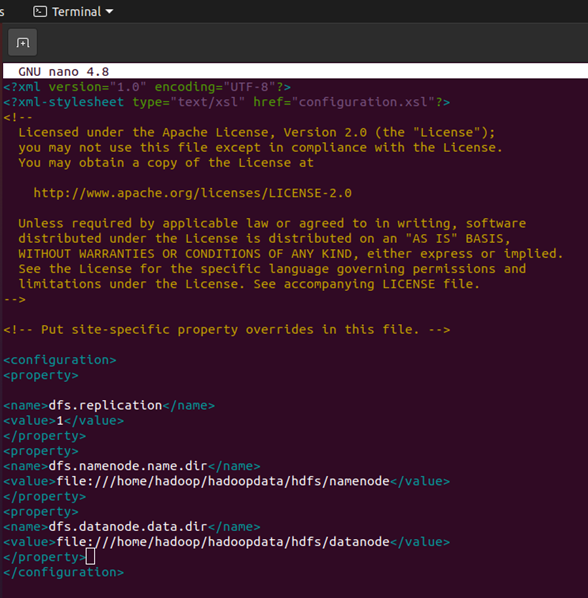
***<property>***

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

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

***</property>***

***</configuration>***



**Edit mapred-site.xml:**

***nano $HADOOP\_HOME/etc/hadoop/mapred-site.xml***

Alter the configuration within the file as follows:

***<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>***

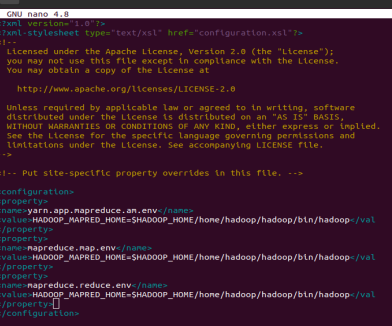
***<name>mapreduce.reduce.env</name>***

***<value>HADOOP\_MAPRED\_HOME=$HADOOP\_HOME/home/hadoop/hadoop/bin/hadoop</value>***

***</property>***

***</configuration>***





**Edit yarn-site.xml:**

Finally, modify the yarn-site.xml file to reflect the required changes:

*nano $HADOOP\_HOME/etc/hadoop/yarn-site.xml*



Adjust the settings as shown below:

***<configuration>***

***<property>***

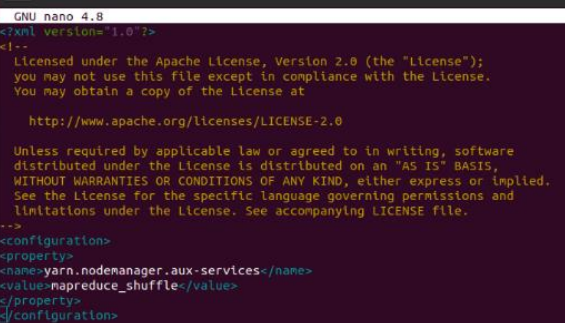
***<name>yarn.nodemanager.aux-services</name>***

***<value>mapreduce\_shuffle</value>***

***</property>***

***</configuration>***

Save the changes made in each file after editing.



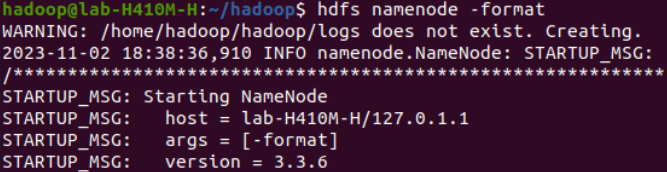
**Step 12: Start Hadoop Cluster**

Before initiating the Hadoop cluster, it’s necessary to format the Namenode as the ‘hadoop’ user.

**Format the Namenode:**

Format the Hadoop Namenode by running the following command:

*hdfs namenode -format*



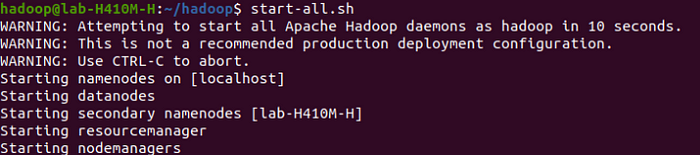
Upon successful completion of the formatting process, you should see a confirmation message similar to “Storage directory /home/hadoop/hadoopdata/hdfs/namenode has been successfully formatted”.

**Start the Hadoop Cluster:**

Now, start the Hadoop cluster by executing the following command:

*start-all.sh*

This command initializes all the required Hadoop daemons for the cluster to become operational.

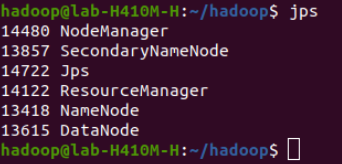


**Check the Status of Hadoop Services:**

To confirm that all necessary Hadoop services are up and running, you can check their status using the ‘jps’ command

*jps*

The ‘jps’ command displays the Java Virtual Machine (JVM) processes and should show a list of Hadoop services running, such as the NameNode, DataNode, ResourceManager, and NodeManager.

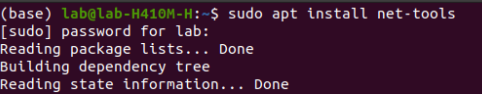


**Step 13: Accessing Hadoop Namenode and Resource Manager:**

Firstly, identify the IP address of your machine, if you’re using Ubuntu and do not have net-tools installed, you can use the following command to install it:

**Install Net-Tools:**

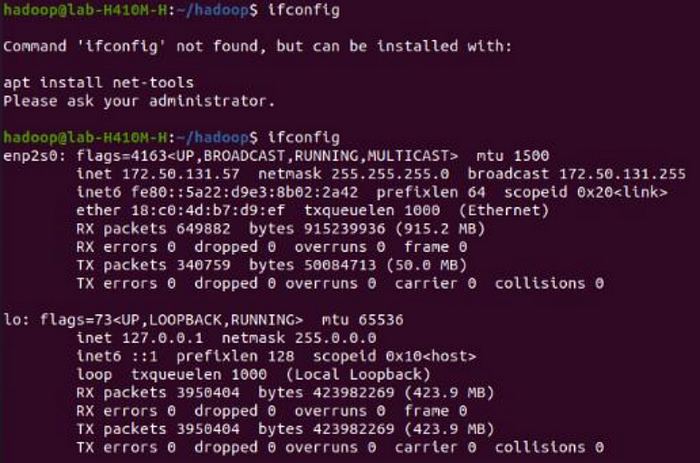
*sudo apt install net-tools*



Then, to find your IP address, run the following command:

**Determine IP Address:**

*ifconfig*

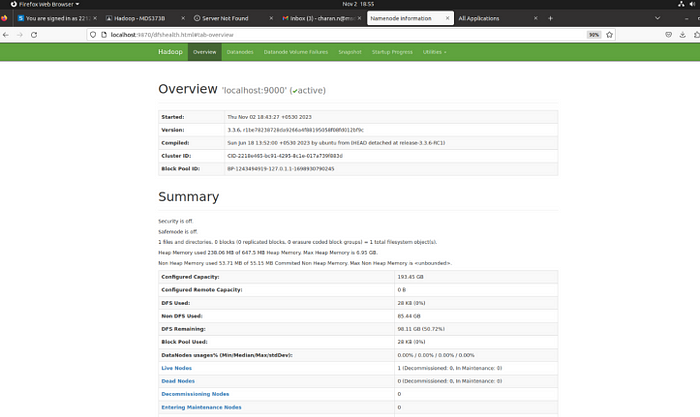


**Access the Hadoop Namenode:**

To access the Hadoop Namenode, open a web browser and enter the following URL:

[*http://192.168.1.6:9870*](http://192.168.1.6:9870/)

Replace ‘192.168.1.6’ with your machine’s IP address. Upon accessing this URL, you should be directed to the Hadoop Namenode page.

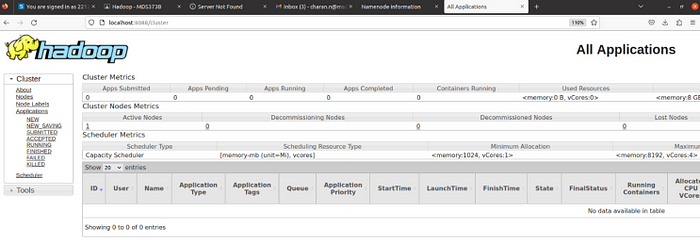


Access the Resource Manager:

Similarly, to access the Hadoop Resource Manager, open a web browser and enter the following URL:

[*http://192.168.1.6:8088*](http://192.168.1.6:8088/)

Again, replace ‘192.168.1.6’ with your actual IP address. This URL will lead you to the Hadoop Resource Manager page.



These steps allow you to access the Hadoop Namenode and Resource Manager via your web browser using the specific IP address and port numbers. By accessing these URLs, you’ll be able to monitor and manage your Hadoop cluster via the provided interfaces for Namenode and Resource Manager.

**Step 14: Verifying the Hadoop Cluster:**

Once the Hadoop cluster has been installed and configured, it’s essential to conduct some tests within the Hadoop Distributed File System (HDFS) to ensure its proper functioning.

**Create Directories in HDFS:**

To begin testing the Hadoop environment, let’s create a couple of directories within the HDFS. Run the following commands:

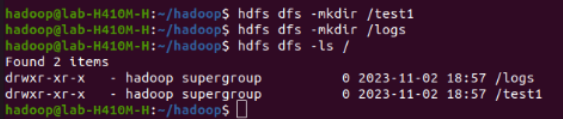
*hdfs dfs -mkdir /test1*

*hdfs dfs -mkdir /logs*

**List Directories in HDFS:**

To verify that the directories have been successfully created in the HDFS, execute the following command to list contents of the root directory (“/”):

*hdfs dfs -ls /*



You should see the created directories displayed in the output, indicating they were successfully created.

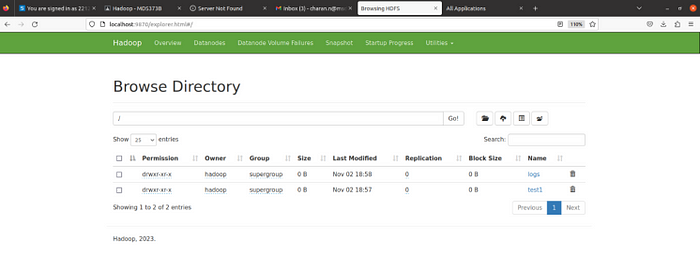
**Transfer Files to the Hadoop File System:**

As an example, you can move log files from the host machine to the Hadoop file system using the following command:

*hdfs dfs -put /var/log/\* /logs/*

**Verify Files and Directories:**

To further confirm the presence of the created directories and uploaded files, access the Hadoop web interface. Navigate to ‘Utilities’ => ‘Browse the file system’. Here, you should be able to visualise the directories created earlier and the files that were uploaded. This confirms their existence within the Hadoop file system.



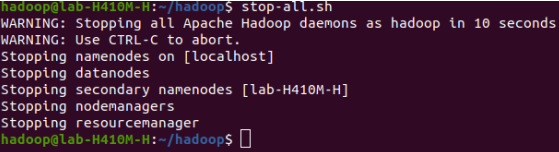
**Step 15: Stopping Hadoop Services:**

When you’re done with your Hadoop operations or need to stop the Hadoop services, follow these steps:

**Stop Hadoop Services:**

As the ‘hadoop’ user, execute the following command to halt the running Hadoop services:

*stop-all.sh*



This command stops all the essential Hadoop daemons that are currently running in your Hadoop cluster. After running this command, the Hadoop services will be shut down.

**Conclusion:**

In conclusion, you’ve learned the process of installing Hadoop on an Ubuntu system.

Now with the powerful Hadoop framework, you’re ready to explore and leverage the capabilities it offers for big data analytics.

Whether it’s data processing, storage, or complex analytics, Hadoop empowers you to manage and analyse vast volumes of data.

Remember that big data has the possibility of unleashing important insights and creativity as you begin on your Hadoop adventure inside the Ubuntu ecosystem.

The possibilities with Hadoop are unlimited, much like the infinite universe, and with each data point you analyse, you travel farther into the cosmos of knowledge.

Continue to explore, experiment, and strive for the stars. Your data-driven adventures with Hadoop on Ubuntu are only limited by your creativity.

So, dare to go where no dataset has gone before, and may your big data endeavours be as unlimited as the universe itself.