This tutorial was conducted on a Multipass instance.

**Step 1: Prepare the environment**

Ensure the system is up to date using command:

$ sudo apt update

A blue screen with white and red dots

AI-generated content may be incorrect.

Upgrade all system’s packages to the latest version, flag -y used to automatically approve the upgrade

$ sudo apt upgrade -y

A blue screen with white text

AI-generated content may be incorrect.

Install OpenJDK 11 – Java development toolkit for running Java applications (Java Runtime Environment) and developing Java applications (Java Development Kit)

$ sudo apt install openjdk-11-jdk -y

A screenshot of a computer program

AI-generated content may be incorrect.

To verify your installation, use command:

$ java -version

A blue screen with white text

AI-generated content may be incorrect.

**Step 2: Create user “khtn\_22120210”**

Create user “khtn\_22120210”, using command:

$ sudo adduser khtn\_22120210

A computer screen shot of white text

AI-generated content may be incorrect.

To grant superuser privileges to the new user, add them to the sudo group using the following command:

$ sudo usermod -aG sudo khtn\_22120210



As you can see, when using command $ groups khtn\_22120210, user “khtn\_22120210” have an alias “sudo”.

Upon completion, switch to the user “khtn\_22120210”using command:

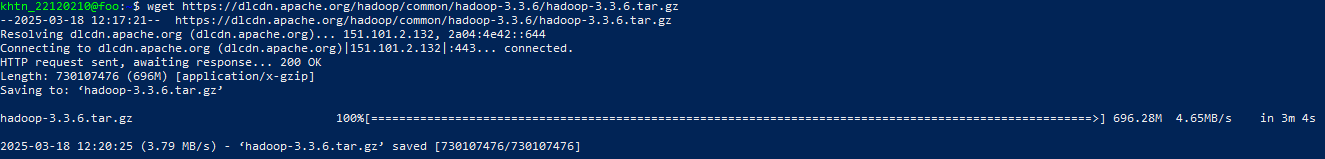
$ sudo su – khtn\_22120210



**Step 3: Download Hadoop**

Download Hadoop 3.3.6 from Hadoop’s official website using command:

$ wget https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz



Extract Hadoop using command:

$ tar -xvzf hadoop-3.3.6.tar.gz

A blue screen with white and red dots

AI-generated content may be incorrect.

Check whether we are currently in the same directory as hadoop-3.3.6 (default: ***/home/khtn\_22120210***). If so, move the files to ***/usr/local/hadoop*** using the following command:

$ sudo mv hadoop-3.3.6 /usr/local/hadoop



Create a directory to store logs by using:

$ sudo mkdir /usr/local/hadoop/logs





Alter the ownership of the ***/usr/local/hadoop*** directory to the user “khtn\_22120210” using following command:

$ sudo chown -R khtn\_22120210 /usr/local/hadoop



**Step 4: Create SSH key**

Install the OpenSSH server and client:

A blue screen with white text

AI-generated content may be incorrect.

Use the subsequent command to generate both private and public keys (it’s important not to set the passphrase):

$ ssh-keygen -t rsa

A computer screen shot of a blue screen

AI-generated content may be incorrect.

Add the public key to *authorized\_keys* using command:

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

Using the following command change the file permissions of authorized\_keys:

$ sudo chmod 640 ~/.ssh/authorized\_keys

Start the SSH service:

$ sudo service ssh start

Confirm the SSH configuration:

$ ssh localhost

A blue screen with white text

AI-generated content may be incorrect.

**Step 5: Configuring Hadoop’s environmental variables**

Open configuration file using command:

$ nano ~/.bashrc

A blue screen with white text

AI-generated content may be incorrect.

Move to the end of the file using Ctrl + / then Ctrl + V

A screenshot of a computer

AI-generated content may be incorrect.

Paste Hadoop’s configuration at the end of the file:

export JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64  
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 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”

A screenshot of a computer

AI-generated content may be incorrect.

Save the changes by pressing Ctrl+S and exit the *nano*text editor by pressing Ctrl+X

To enable the changes, source the *.bashrc*file using command:

$ source ~/.bashrc



Check for Hadoop version using command:

$ hadoop version

A blue screen with white text

AI-generated content may be incorrect.

**Step 6: Config Hadoop**

Define java environment variables in *hadoop-env.sh* file:

To open *hadoop-env.sh* file use the command:

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

A computer screen shot of a blue screen

AI-generated content may be incorrect.

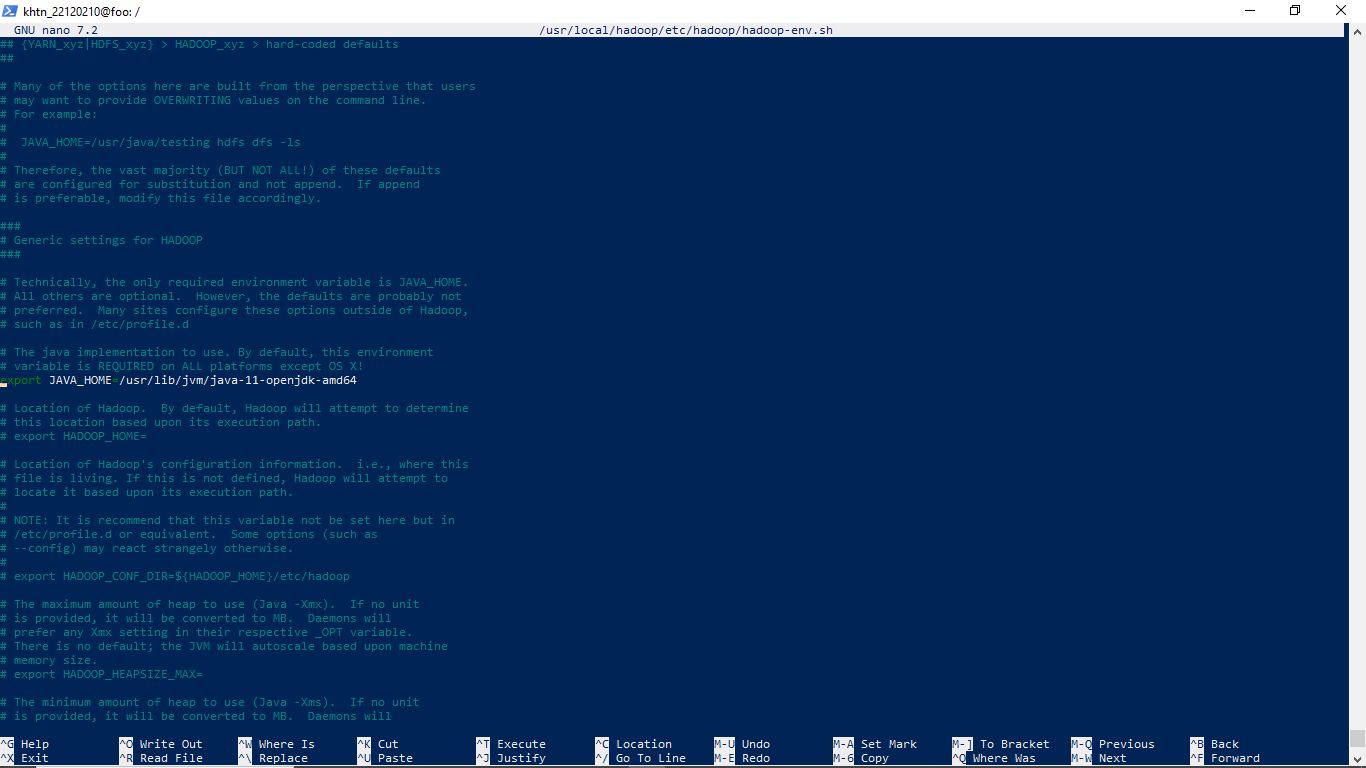
To search for the “export JAVA\_HOME” phrase use Ctrl + W

A screenshot of a computer

AI-generated content may be incorrect.

And change it as below:

export JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64



Configuring Hadoop to Pseudo-Distributed Mode:

Open file *core-site.xml* to configure HDFS using command:

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

A computer screen shot of a blue screen

AI-generated content may be incorrect.

Add the following lines between <configuration> </configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost:9000</value>

</property>

(To edit the file press Ctrl + Y, then Ctrl + V to go to the end of file and right-click to paste those above properties in place)

A computer screen shot of a blue screen

AI-generated content may be incorrect.

Then press Ctrl + S to save the file then Ctrl + X to exit

Create a directory to store node metadata using the following command:

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



Grant the ownership of the created directory to the “khtn\_22120210” user:

$ sudo chown -R khtn\_22120210 /home/hadoop/hdfs



Open file *hdfs-site.xml* to configure data storage folder, using command:

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

Add the following properties between <configuration> </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>

A screenshot of a computer program

AI-generated content may be incorrect.

Open file *mapred-site.xml* to configure MapReduce mode (default is Standalone Mode), using command:

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

Add the following lines between <configuration> </configuration>

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

A blue screen with white text

AI-generated content may be incorrect.

Open file yarn-site.xml to configure auxiliary service (default is Standalone Mode), using command:

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

Add the following lines between <configuration> </configuration>

<property>  
 <name>yarn.nodemanager.aux-services</name>  
 <value>mapreduce\_shuffle</value>  
</property>

A blue screen with white text

AI-generated content may be incorrect.

Format HDFS’ namenode, using command:

$ hdfs namenode -format

After starting YARN and DFS on “foo,” I realized that I hadn’t allocated enough RAM to the virtual machine. As a result, “foo” would not be able to handle other tasks needed to complete this lab. Therefore, I will create a new instance called “bigdata” and use it as an alternative. All setups are similar to the tutorial shown above.

**Step 7: Start Hadoop**

A screen shot of a computer program

AI-generated content may be incorrect.

To verify running services, using command:

$ jps

A screenshot of a computer program

AI-generated content may be incorrect.

Now you can use the provided Hadoop UI. **Note:** If you set up the environment on a virtual machine, use the VM's IPv4 address instead of **localhost** to access Hadoop’s web services. As shown below, my instance’s IPv4 address is **172.22.235.75**.



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Step 8: File manipulation with Hadoop**

Use the following commands to create a directory for hadoop-test.jar on HDFS:

$ hdfs dfs -mkdir /hcmus

$ hdfs dfs -mkdir /hcmus/22120210



Verify the owner of the directory:



khtn\_22120210 is the creator also owner of the directory so it is not necessary to change the owner

To transfer file between host and instance “bigdata”, open Windows Powershell with Administrator previllige and type the following command:

multipass transfer "E:\hadoop-test.jar" bigdata:/home/ubuntu

After that move the file to ***/home/khtn\_22120210***, using command:

$ sudo mv hadoop-test.jar /home/khtn\_22120210

Verify the transfer:



Upon completion, use the following command to upload file on HDFS:

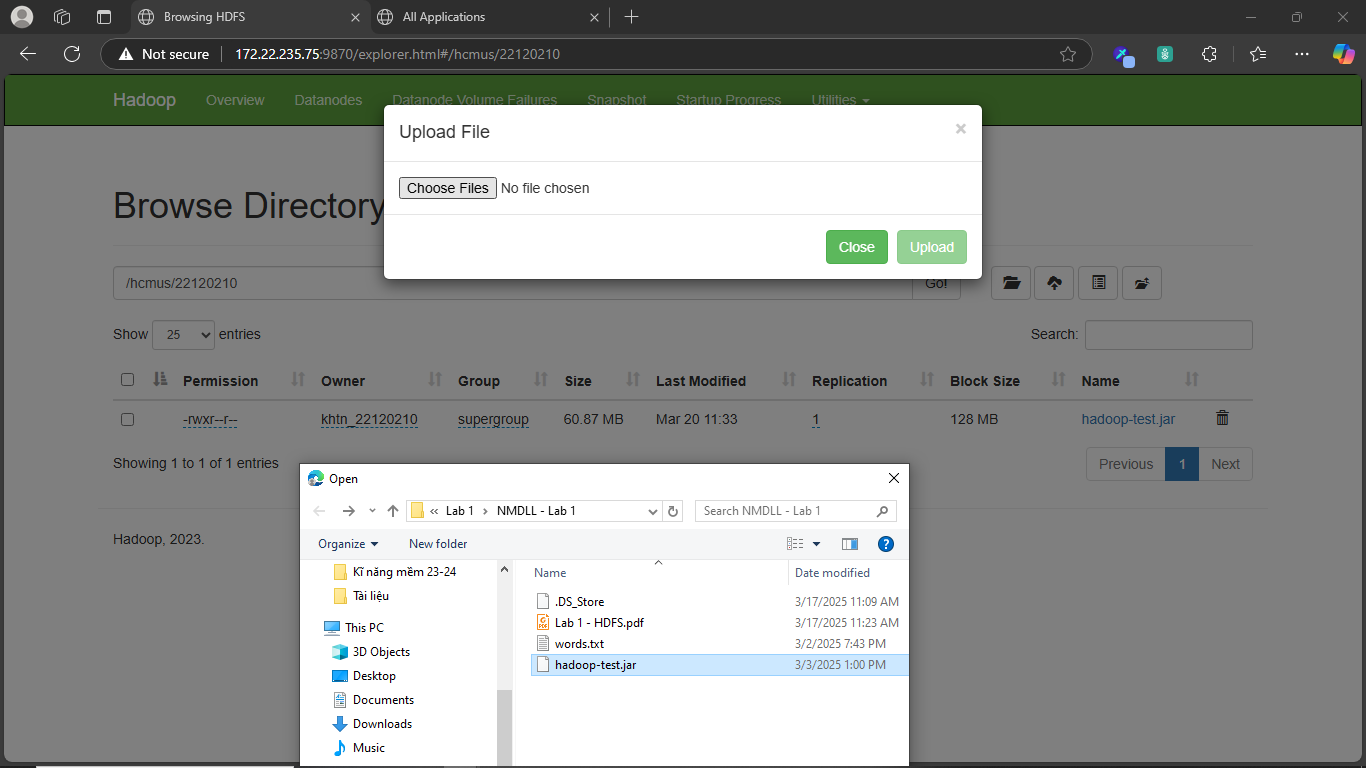
$ hdfs dfs -put hadoop-test.jar /hcmus/22120210

Verify if the file was uploaded successfully:

A blue screen with white text

AI-generated content may be incorrect.

You can also use the web service UI to upload the file:



Change the file permission to execute the file on hdfs:

$ hdfs dfs -chmod 744 /hcmus/22120210/hadoop-test.jar

**Step 9: Run the programme**

To execute the file, use command:

$ java -jar hadoop-test.jar 9000/hcmus/22120210

A blue screen with white text

AI-generated content may be incorrect.

Verify the result:

