

# Lab 01: Introduction to Hadoop Ecosystem

Student's Name: Lê Võ Nhật Minh

Student ID: 22120210

This tutorial was conducted on a Multipass instance (most operations are the same as installing with WSL<sup>[1]</sup>)

## Table of contents

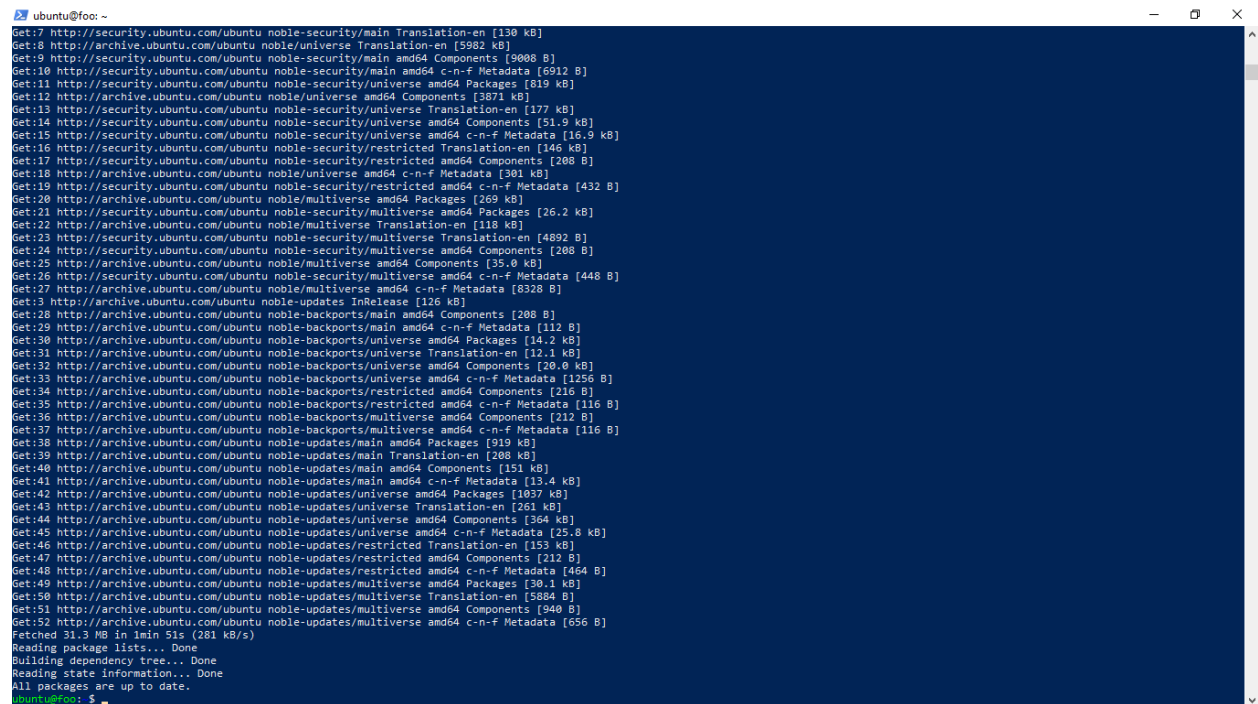
|  |    |
|--|----|
| I. Setup a Hadoop cluster .....                            | 3  |
| Step 1: Prepare the environment .....                      | 3  |
| Step 2: Create user “khtn_22120210” .....                  | 5  |
| Step 3: Download Hadoop .....                              | 6  |
| Step 4: Create SSH key .....                               | 9  |
| Step 5: Configuring Hadoop’s environmental variables ..... | 11 |
| Step 6: Config Hadoop .....                                | 14 |
| Step 7: Start Hadoop .....                                 | 22 |
| Step 8: File manipulation with Hadoop .....                | 24 |
| Step 9: Run the programme .....                            | 26 |
| II. Warm up with Word Count .....                          | 27 |
| 1. Word Count Mapper .....                                 | 27 |
| 2. Word Count Reducer .....                                | 28 |
| 3. Word Count Result .....                                 | 29 |
| III. References: .....                                     | 30 |

# I. Setup a Hadoop cluster

## Step 1: Prepare the environment

Ensure the system is up to date using command:

```
$ sudo apt update
```



```
ubuntu@foo: ~  
Get:7 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [130 kB]  
Get:8 http://archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]  
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [9088 B]  
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [6912 B]  
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [819 kB]  
Get:12 http://archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]  
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [177 kB]  
Get:14 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [51.9 kB]  
Get:15 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [16.9 kB]  
Get:16 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [146 kB]  
Get:17 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [208 B]  
Get:18 http://archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]  
Get:19 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [432 B]  
Get:20 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]  
Get:21 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [26.2 kB]  
Get:22 http://archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]  
Get:23 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [4892 B]  
Get:24 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]  
Get:25 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]  
Get:26 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [448 B]  
Get:27 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]  
Get:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]  
Get:28 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]  
Get:29 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]  
Get:30 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [14.2 kB]  
Get:31 http://archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [12.1 kB]  
Get:32 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [20.0 kB]  
Get:33 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1256 B]  
Get:34 http://archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]  
Get:35 http://archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]  
Get:36 http://archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]  
Get:37 http://archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]  
Get:38 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [919 kB]  
Get:39 http://archive.ubuntu.com/ubuntu noble-updates/main Translation-en [208 kB]  
Get:40 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [151 kB]  
Get:41 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [13.4 kB]  
Get:42 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1037 kB]  
Get:43 http://archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [261 kB]  
Get:44 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [364 kB]  
Get:45 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [25.8 kB]  
Get:46 http://archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [153 kB]  
Get:47 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]  
Get:48 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [464 B]  
Get:49 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [30.1 kB]  
Get:50 http://archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [5084 B]  
Get:51 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]  
Get:52 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [656 B]  
Fetched 31.3 MB in 1min 51s (281 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
All packages are up to date.  
ubuntu@foo: $
```

Figure 1.1: Check for the latest version of all packages.

Upgrade all system packages to the latest version. The flag `-y` used to automatically approve the upgrade

```
$ sudo apt upgrade -y
```

```
ubuntu@foo:~$ sudo apt upgrade -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

**Figure 1.2:** Upgrade all packages

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
```

```
Setting up openjdk-11-jre:amd64 (11.0.26+4-1ubuntu1~24.04) ...
Setting up openjdk-11-jdk-headless:amd64 (11.0.26+4-1ubuntu1~24.04) ...
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jar to provide /usr/bin/jar (jar) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jarsigner to provide /usr/bin/jarsigner (jarsigner) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/javac to provide /usr/bin/javac (javac) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/javadoc to provide /usr/bin/javadoc (javadoc) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/javap to provide /usr/bin/javap (javap) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jcmd to provide /usr/bin/jcmd (jcmd) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jdb to provide /usr/bin/jdb (jdb) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jdeprscan to provide /usr/bin/jdeprscan (jdeprscan) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jdeps to provide /usr/bin/jdeps (jdeps) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jfr to provide /usr/bin/jfr (jfr) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jimage to provide /usr/bin/jimage (jimage) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jinfo to provide /usr/bin/jinfo (jinfo) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jlink to provide /usr/bin/jlink (jlink) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jmap to provide /usr/bin/jmap (jmap) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jmod to provide /usr/bin/jmod (jmod) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jps to provide /usr/bin/jps (jps) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jrunscript to provide /usr/bin/jrunscript (jrunscript) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jshell to provide /usr/bin/jshell (jshell) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jstack to provide /usr/bin/jstack (jstack) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jstat to provide /usr/bin/jstat (jstat) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jstatd to provide /usr/bin/jstatd (jstatd) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/rmic to provide /usr/bin/rmic (rmic) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/serialver to provide /usr/bin/serialver (serialver) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jaotc to provide /usr/bin/jaotc (jaotc) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jhsdb to provide /usr/bin/jhsdb (jhsdb) in auto mode
Setting up openjdk-11-jdk:amd64 (11.0.26+4-1ubuntu1~24.04) ...
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jconsole to provide /usr/bin/jconsole (jconsole) in auto mode
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@foo:~$
```

**Figure 1.3:** Install Java development toolkit

To verify your installation, use command:

```
$ java -version
```

```
ubuntu@foo:~$ java -version
openjdk version "11.0.26" 2025-01-21
OpenJDK Runtime Environment (build 11.0.26+4-post-Ubuntu-1ubuntu124.04)
OpenJDK 64-Bit Server VM (build 11.0.26+4-post-Ubuntu-1ubuntu124.04, mixed mode, sharing)
```

**Figure 1.4:** Java version detail

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

Create user “khtn\_22120210”, using command:

```
$ sudo adduser khtn_22120210
```

```
ubuntu@foo:~$ sudo adduser khtn_22120210
info: Adding user `khtn_22120210' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `khtn_22120210' (1001) ...
info: Adding new user `khtn_22120210' (1001) with group `khtn_22120210 (1001)' ...
info: Creating home directory `/home/khtn_22120210' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for khtn_22120210
Enter the new value, or press ENTER for the default
  Full Name []: Ming3993
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] Y
info: Adding new user `khtn_22120210' to supplemental / extra groups `users' ...
info: Adding user `khtn_22120210' to group `users' ...
```

**Figure 2.1:** Add a new user to the Linux system

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

```
$ sudo usermod -aG sudo khtn_22120210
```

```
ubuntu@foo:/usr/local$ sudo usermod -aG sudo khtn_22120210
ubuntu@foo:/usr/local$ groups khtn_22120210
khtn_22120210 : khtn_22120210 sudo users
```

**Figure 2.2:** Granting administrative privileges to the new user

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
```

```
ubuntu@foo:~$ sudo su - khtn_22120210
khtn_22120210@foo:~$
```

**Figure 2.3:** Switch to the newly created user

### 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
```

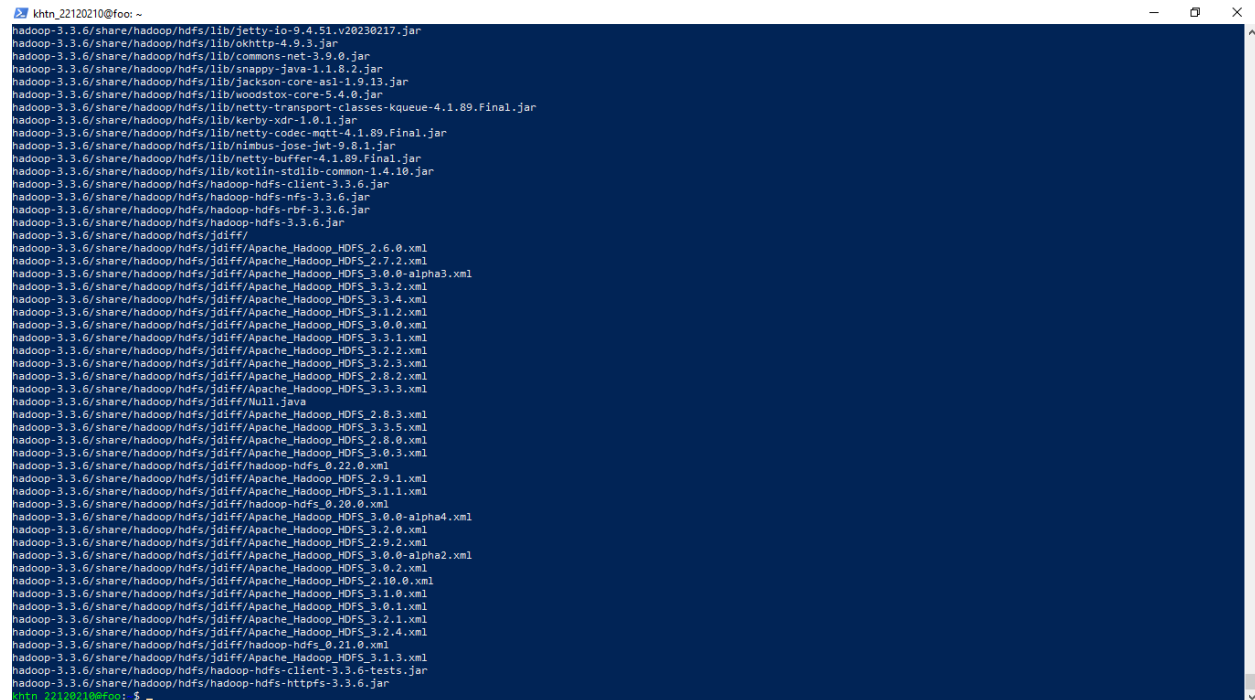
```
khtn_22120210@foo:~$ wget https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz
--2025-03-18 12:17:21-- https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 730107476 (696M) [application/x-gzip]
Saving to: 'hadoop-3.3.6.tar.gz'

hadoop-3.3.6.tar.gz           100%[=====] 696.28M  4.65MB/s  in 3m 4s
2025-03-18 12:20:25 (3.79 MB/s) - 'hadoop-3.3.6.tar.gz' saved [730107476/730107476]
```

**Figure 3.1:** Downloading Hadoop

Extract Hadoop using command:

```
$ tar -xvzf hadoop-3.3.6.tar.gz
```

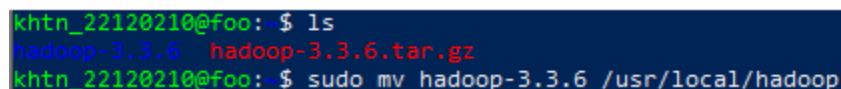


```
khtn_22120210@foo: ~  
hadoop-3.3.6/share/hadoop/hdfs/lib/jetty-io-9.4.51.v20230217.jar  
hadoop-3.3.6/share/hadoop/hdfs/lib/okhttp-4.9.3.jar  
hadoop-3.3.6/share/hadoop/hdfs/lib/commons-net-3.9.0.jar  
hadoop-3.3.6/share/hadoop/hdfs/lib/snappy-java-1.1.8.2.jar  
hadoop-3.3.6/share/hadoop/hdfs/lib/jackson-core-asl-1.9.13.jar  
hadoop-3.3.6/share/hadoop/hdfs/lib/woodstox-core-5.4.0.jar  
hadoop-3.3.6/share/hadoop/hdfs/lib/netty-transport-classes-kqueue-4.1.89.Final.jar  
hadoop-3.3.6/share/hadoop/hdfs/lib/kerby-xdr-1.0.1.jar  
hadoop-3.3.6/share/hadoop/hdfs/lib/netty-codec-mqtt-4.1.89.Final.jar  
hadoop-3.3.6/share/hadoop/hdfs/lib/nimbus-jose-jwt-9.8.1.jar  
hadoop-3.3.6/share/hadoop/hdfs/lib/netty-buffer-4.1.89.Final.jar  
hadoop-3.3.6/share/hadoop/hdfs/lib/kotlin-stdlib-common-1.4.10.jar  
hadoop-3.3.6/share/hadoop/hdfs/hadoop-hdfs-client-3.3.6.jar  
hadoop-3.3.6/share/hadoop/hdfs/hadoop-hdfs-nfs-3.3.6.jar  
hadoop-3.3.6/share/hadoop/hdfs/hadoop-hdfs-rbf-3.3.6.jar  
hadoop-3.3.6/share/hadoop/hdfs/hadoop-hdfs-3.3.6.jar  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_2.6.0.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_2.7.2.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.0.0-alpha3.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.2.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.3.4.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.1.2.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.0.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.3.1.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.2.2.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.2.3.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_2.8.2.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.3.3.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Null.java  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_2.8.3.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.3.5.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_2.8.0.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.0.3.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/hadoop-hdfs_0.22.0.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_2.9.1.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.1.1.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/hadoop-hdfs_0.20.0.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.0.0-alpha4.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.2.0.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_2.9.2.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.0.0-alpha2.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.0.2.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_2.10.0.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.1.0.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.0.1.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.2.1.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.2.4.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/hadoop-hdfs_0.21.0.xml  
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.1.3.xml  
hadoop-3.3.6/share/hadoop/hdfs/hadoop-hdfs-client-3.3.6-tests.jar  
hadoop-3.3.6/share/hadoop/hdfs/hadoop-hdfs-httpfs-3.3.6.jar  
khtn_22120210@foo: ~
```

Figure 3.2: Extracting Hadoop

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
```



```
khtn_22120210@foo: ~$ ls  
hadoop-3.3.6  hadoop-3.3.6.tar.gz  
khtn_22120210@foo: ~$ sudo mv hadoop-3.3.6 /usr/local/hadoop
```

Figure 3.3: Move Hadoop to the new folder

Create a directory to store logs by using:

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

```
khtn_22120210@foo:~$ sudo mkdir /usr/local/hadoop/logs
```

**Figure 3.4a:** Create directory for storing logs

```
khtn_22120210@foo:/usr/local/hadoop$ ls
LICENSE-binary  NOTICE-binary  README.txt  etc      lib      licenses-binary  sbin
LICENSE.txt     NOTICE.txt     bin         include  libexec  logs             share
```

**Figure 3.4b:** Verify successful creation

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
```

```
khtn_22120210@foo:~$ sudo chown -R khtn_22120210 /usr/local/hadoop
```

**Figure 3.5:** Alter directory’s ownership



## Step 4: Create SSH key

Install the OpenSSH server and client:

```
khtn_22120210@foo:~$ sudo apt install ssh
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  ssh
0 upgraded, 1 newly installed, 0 to remove and 15 not upgraded.
Need to get 4658 B of archives.
After this operation, 57.3 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 ssh all 1:9.6p1-3ubuntu13.8 [4658 B]
Fetched 4658 B in 1s (7838 B/s)
Selecting previously unselected package ssh.
(Reading database ... 77216 files and directories currently installed.)
Preparing to unpack .../ssh_1%3a9.6p1-3ubuntu13.8_all.deb ...
Unpacking ssh (1:9.6p1-3ubuntu13.8) ...
Setting up ssh (1:9.6p1-3ubuntu13.8) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

User sessions running outdated binaries:
  ubuntu @ session #3: java[2082]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

**Figure 4.1:** Instal OpenSSH Server package

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

```
$ ssh-keygen -t rsa
```

```
khtn_22120210@foo: $ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/khtn_22120210/.ssh/id_rsa):
Created directory '/home/khtn_22120210/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/khtn_22120210/.ssh/id_rsa
Your public key has been saved in /home/khtn_22120210/.ssh/id_rsa.pub
The key fingerprint is:
SHA256: [REDACTED] khtn_22120210@foo
The key's randomart image is:
+---[RSA 3072]---+
|
|
|
|
|
|
|
|
|
|
+-----[SHA256]-----+
```

**Figure 4.2:** Generate private and public keys

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
```

```
$ ssh localhost
```

```
khtn_22120210@foo:~$ ssh localhost
The authenticity of host 'localhost (::1)' can't be established.
ED25519 key fingerprint is SHA256:
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.
Enter passphrase for key '/home/khtn_22120210/.ssh/id_rsa':
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-55-generic x86_64)
```

### Figure 4.3: Localhost's information

## Step 5: Configuring Hadoop's environmental variables

Open configuration file using command:

```
$ nano ~/.bashrc
```

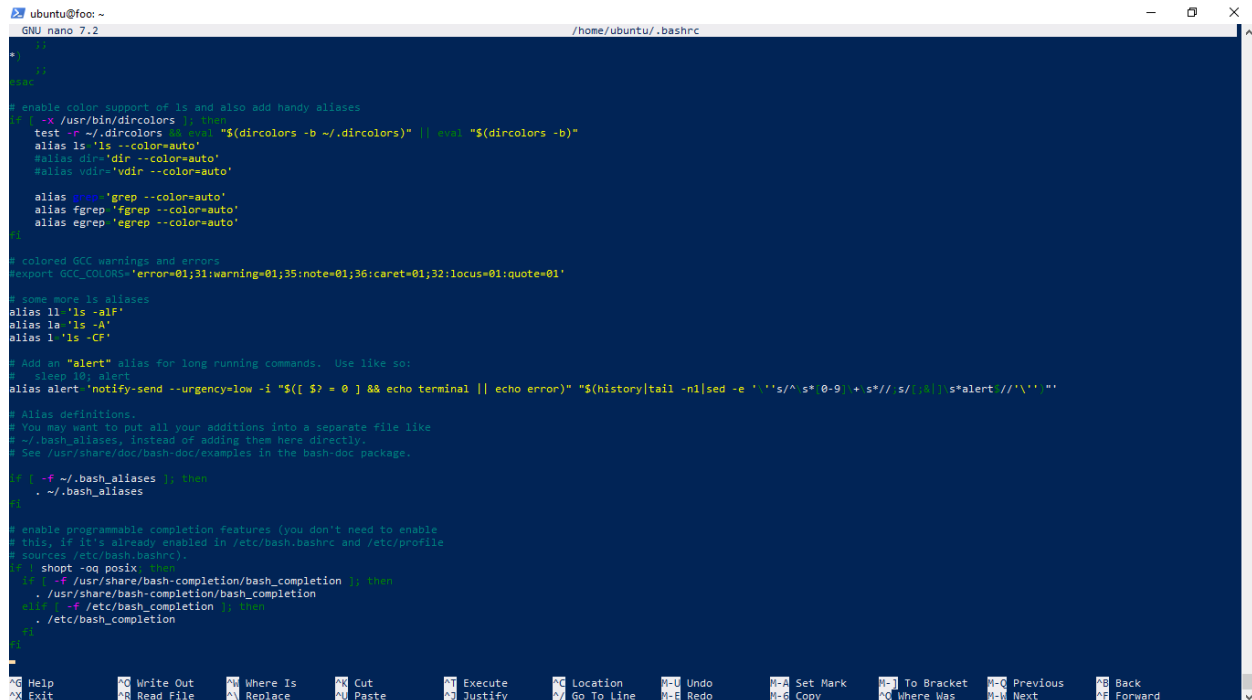
```
ubuntu@foo: ~  
GNU nano 7.2 /home/ubuntu/.bashrc  
# If not running interactively, don't do anything  
if [[ $- != *i* ]] ; then  
    return;  
fi  
  
# don't put duplicate lines or lines starting with space in the history.  
# See bash(1) for more options  
HISTCONTROL=ignoreboth  
  
# append to the history file, don't overwrite it  
shopt -s histappend  
  
# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)  
HISTSIZE=1000  
HISTFILESIZE=2000  
  
# check the window size after each command and, if necessary,  
# update the values of LINES and COLUMNS.  
shopt -s checkwinsize  
  
# If set, the pattern "***" used in a pathname expansion context will  
# match all files and zero or more directories and subdirectories.  
shopt -s globstar  
  
# make less more friendly for non-text input files; see lesspipe(1)  
:~X /usr/bin/lesspipe || ll eval: "$(SHELL=/bin/sh lesspipe)"  
  
# set variable identifying the chroot you work in (used in the prompt below)  
[ ! -z "${debian_chroot:-}" ] && { . /etc/debian_chroot ;} then  
    debian_chroot=$(cat /etc/debian_chroot)  
fi  
  
# set a fancy prompt (non-color, unless we know we "want" color)  
if [ "$TERM" ]; then  
    xterm-color *-256color | color_prompt yes;  
fi  
  
# uncomment for a colored prompt, if the terminal has the capability; turned  
# off by default to not distract the user: the focus in a terminal window  
# should be on the output of commands, not on the prompt  
force_color_prompt=yes  
  
if [ -n "${force_color_prompt}" ]; then  
    if [ -x /usr/bin/tput ] && tput setaf 1 &&/dev/null 2>> /dev/null  
then
```

Read 117 lines

|        |             |            |         |              |
|--------|-------------|------------|---------|--------------|
| ⌕ Help | ✎ Write Out | ↻ Where Is | ✂ Cut   | 🔍 Execute    |
| ⏹ Exit | 📄 Read File | 🔄 Replace  | 📋 Paste | 🔍 Justify    |
|        |             |            |         | ↶ Undo       |
|        |             |            |         | ↷ Redo       |
|        |             |            |         | 🔖 Set Mark   |
|        |             |            |         | 🗑 Copy       |
|        |             |            |         | 🔗 To Bracket |
|        |             |            |         | ⬅ Previous   |
|        |             |            |         | ➡ Next       |
|        |             |            |         | ⏪ Back       |
|        |             |            |         | ⏩ Forward    |

### Figure 5.1: Open `.bashrc` file

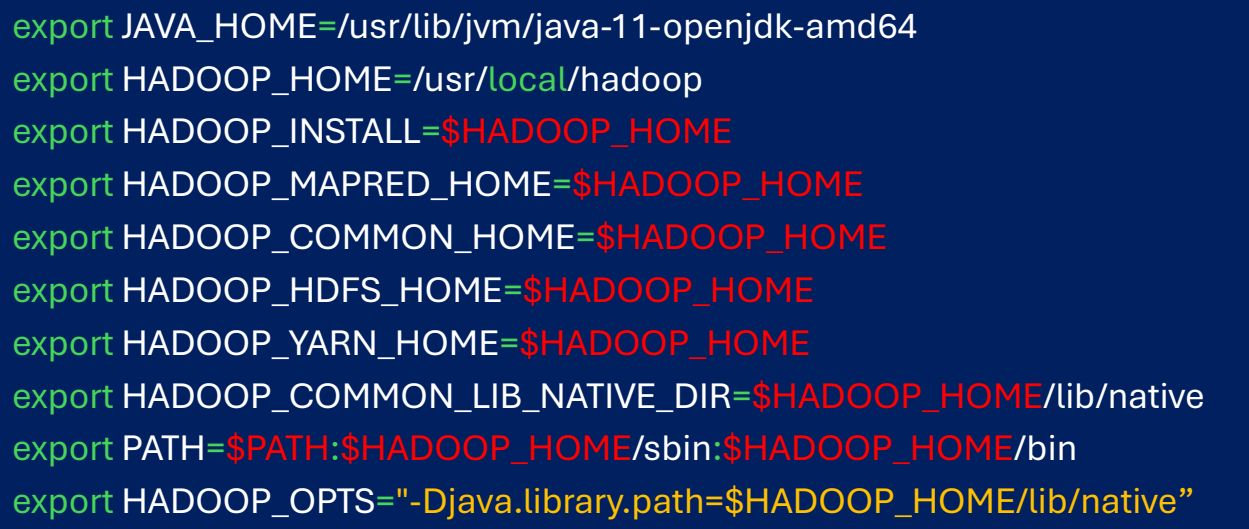
Move to the end of the file using **Ctrl + /** then **Ctrl + V**



The screenshot shows a terminal window with the nano text editor open to the file `/home/ubuntu/.bashrc`. The cursor is at the end of the file, after the last line of code. The code visible includes comments about enabling color support, defining aliases for `ls`, `grep`, and `ll`, and enabling programmable completion features. The bottom status bar of the nano editor shows various keyboard shortcuts like `Ctrl+H` for Help, `Ctrl+W` for Write Out, etc.

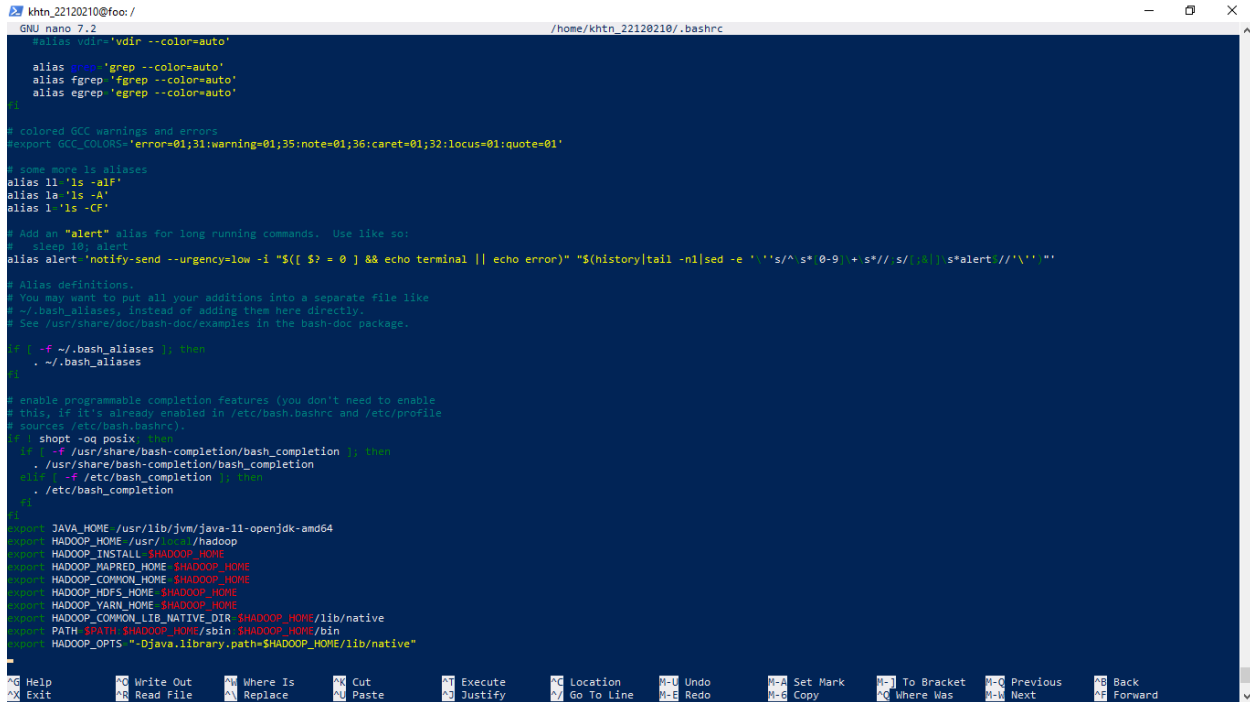
**Figure 5.2:** The end of `.bashrc` file

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



The screenshot shows a list of Hadoop configuration lines to be pasted into the `.bashrc` file. The lines are as follows:

```
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 terminal window showing the nano text editor editing the .bashrc file. The file contains various aliases for grep, some GCC color settings, an 'alert' alias, and Hadoop environment variables. The bottom of the window shows the nano editor's command palette with options like Help, Write Out, Where Is, Cut, Execute, Location, Undo, Set Mark, To Bracket, Previous, Back, Exit, Read File, Replace, Paste, Justify, Go To Line, Redo, Copy, Where Was, Next, and Forward.

```
kh tn_22120210@foo: /
GNU nano 7.2 /home/kh tn_22120210/.bashrc
#alias vdir='vdir --color=auto'

alias grep 'grep --color=auto'
alias fgrep 'fgrep --color=auto'
alias egrep 'egrep --color=auto'

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

# some more ls aliases
alias ll 'ls -alF'
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 "${?} = 0 ] && echo terminal || echo error)" "$(history|tail -n1|sed -e '\''s/^ s* 0-9\| + s*// s/|/|/ s*alert//'\''|)"'

# 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 [ -f /usr/share/bash-completion/bash_completion ] then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ] then
        . /etc/bash_completion
    fi
fi

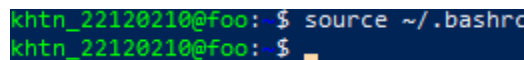
export JAVA_HOME /usr/lib/jvm/java-11-openjdk-amd64
export HADOOP_HOME /usr/local/hadoop
export HADOOP_INSTALL $HADOOP_HOME
export HADOOP_MAPRED_HOME $HADOOP_HOME
export HADOOP_COMMON_HOME $HADOOP_HOME
export HADOOP_HDFS_HOME $HADOOP_HOME
export HADOOP_YARN_HOME $HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR $HADOOP_HOME/lib/native
export PATH $PATH $HADOOP_HOME/sbin $HADOOP_HOME/bin
export HADOOP_OPTS "-Djava.library.path=$HADOOP_HOME/lib/native"
```

**Figure 5.3:** Adding configuration to `.bashrc` file

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
```

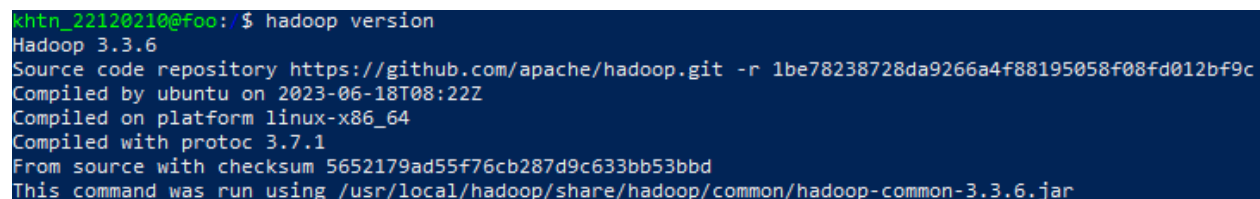
A terminal screenshot showing the command to source the .bashrc file.

```
kh tn_22120210@foo:~$ source ~/.bashrc
kh tn_22120210@foo:~$
```

**Figure 5.4:** Enable the changes

Check for Hadoop version using command:

```
$ hadoop version
```

A terminal screenshot showing the output of the 'hadoop version' command.

```
kh tn_22120210@foo: $ hadoop version
Hadoop 3.3.6
Source code repository https://github.com/apache/hadoop.git -r 1be78238728da9266a4f88195058f08fd012bf9c
Compiled by ubuntu on 2023-06-18T08:22Z
Compiled on platform linux-x86_64
Compiled with protoc 3.7.1
From source with checksum 5652179ad55f76cb287d9c633bb53bbd
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-3.3.6.jar
```

**Figure 5.5:** Hadoop version information

## 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
```

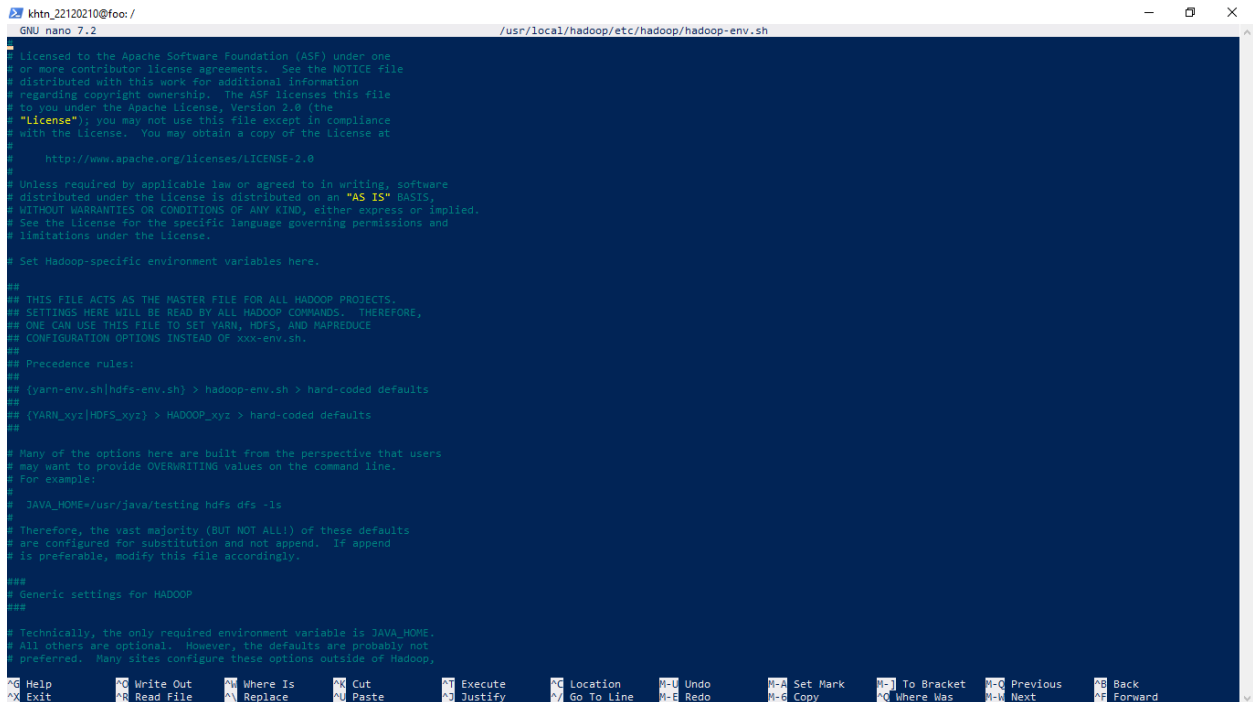
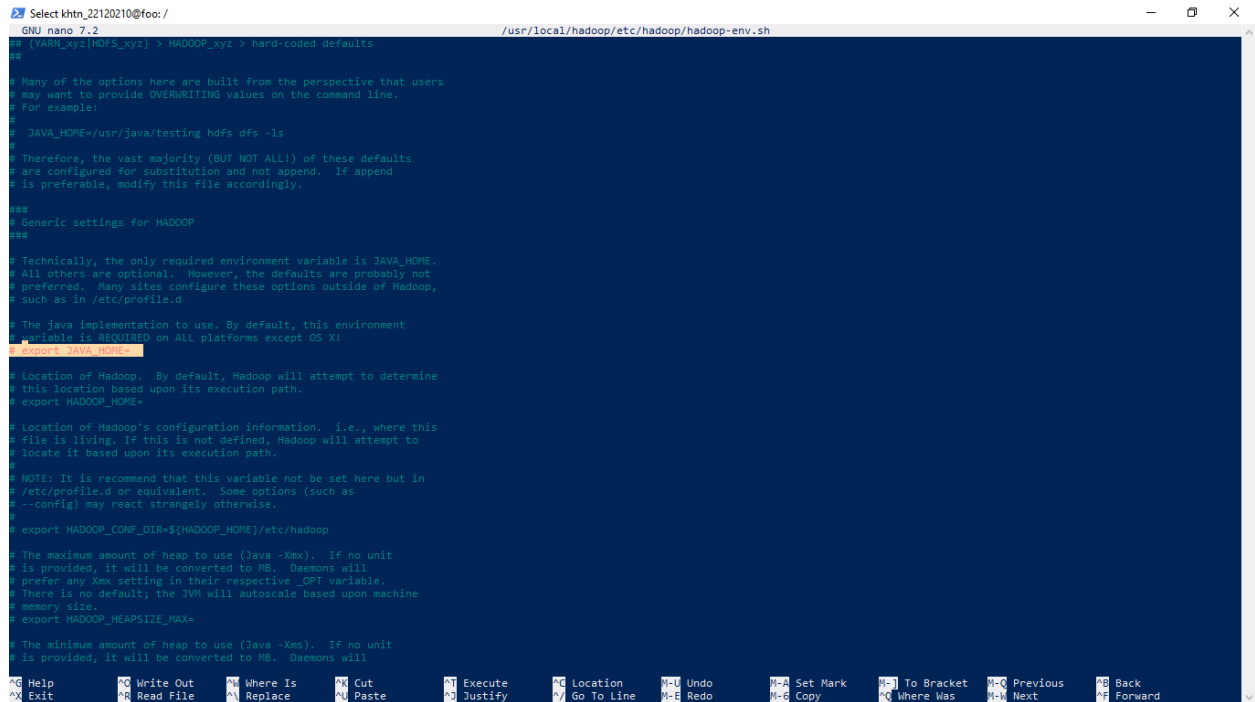
A screenshot of a terminal window showing the nano text editor editing the file /usr/local/hadoop/etc/hadoop/hadoop-env.sh. The terminal title bar shows 'khtn\_22120210@fooc: /usr/local/hadoop/etc/hadoop/hadoop-env.sh'. The nano editor interface includes a top status bar with 'GNU nano 7.2' and a bottom toolbar with various editing commands like Help, Write Out, Where Is, Cut, Execute, Location, Undo, Set Mark, To Bracket, Previous, Back, Exit, Read File, Replace, Paste, Justify, Go To Line, Redo, Copy, Where Was, Next, and Forward. The file content is a comment-heavy shell script for configuring Hadoop environment variables. It includes the Apache License header, precedence rules, and generic settings for HADOOP. The script is designed to be read by the HADOOP command-line programs, which will override any settings in this file with those in the command line or other configuration files. The script sets the JAVA\_HOME environment variable to /usr/java/testing and the HADOOP\_HOME environment variable to /usr/local/hadoop. It also sets the HADOOP\_CONF\_DIR environment variable to /usr/local/hadoop/etc/hadoop. The script is designed to be read by the HADOOP command-line programs, which will override any settings in this file with those in the command line or other configuration files. The script sets the JAVA\_HOME environment variable to /usr/java/testing and the HADOOP\_HOME environment variable to /usr/local/hadoop. It also sets the HADOOP\_CONF\_DIR environment variable to /usr/local/hadoop/etc/hadoop. The script is designed to be read by the HADOOP command-line programs, which will override any settings in this file with those in the command line or other configuration files.

Figure 6.1: Open *hadoop-env.sh* file

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

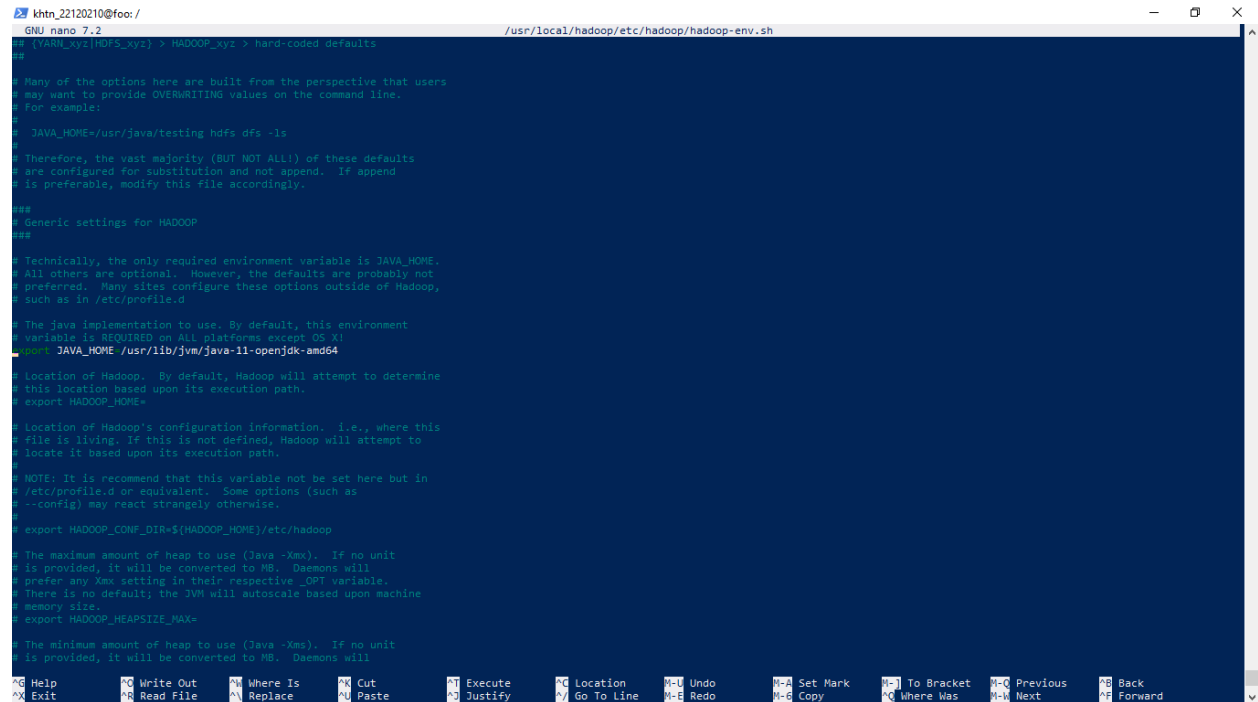


```
GNU nano 7.2 /usr/local/hadoop/etc/hadoop/hadoop-env.sh
## (YARN_KYz|HDFS_KYz) > HADOOP_KYz > hard-coded defaults
##
# Many of the options here are built from the perspective that users
# may want to provide OVERRIDING values on the command line.
# For example:
#
# JAVA_HOME=/usr/java/testing/hdfs/dfs -ls
#
# Therefore, the vast majority (BUT NOT ALL!) of these defaults
# are configured for substitution and not append. If append
# is preferable, modify this file accordingly.
###
# Generic settings for HADOOP
###
# Technically, the only required environment variable is JAVA_HOME.
# All others are optional. However, the defaults are probably not
# preferred. Many sites configure these options outside of Hadoop,
# such as in /etc/profile.d
#
# The java implementation to use. By default, this environment
# variable is REQUIRED on ALL platforms except OS X!
# export JAVA_HOME=
#
# Location of Hadoop. By default, Hadoop will attempt to determine
# this location based upon its execution path.
# export HADOOP_HOME=
#
# Location of Hadoop's configuration information. I.e., where this
# file is living. If this is not defined, Hadoop will attempt to
# locate it based upon its execution path.
#
# NOTE: It is recommend that this variable not be set here but in
# /etc/profile.d or equivalent. Some options (such as
# --config) may react strangely otherwise.
#
# export HADOOP_CONF_DIR=${HADOOP_HOME}/etc/hadoop
#
# The maximum amount of heap to use (Java -Xmx). If no unit
# is provided, it will be converted to MB. Daemons will
# prefer any Xmx setting in their respective _OPT variable.
# There is no default; the JVM will autoscale based upon machine
# memory size.
# export HADOOP_HEAPSIZE_MAX=
#
# The minimum amount of heap to use (Java -Xms). If no unit
# is provided, it will be converted to MB. Daemons will
```

**Figure 6.2:** Search for “export JAVA\_HOME” phrase

And change it as below:

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



```
khtn_22120210@foo: /usr/local/hadoop/etc/hadoop/hadoop-env.sh
GNU nano 7.2
## {HADOOP_OPTS} > HADOOP_OPTS > hard-coded defaults
##
# Many of the options here are built from the perspective that users
# may want to provide OVERRIDING values on the command line.
# For example:
#
# JAVA_HOME=/usr/java/testing/hdfs dfs -ls
#
# Therefore, the vast majority (BUT NOT ALL!) of these defaults
# are configured for substitution and not append. If append
# is preferable, modify this file accordingly.
###
# Generic settings for HADOOP
###
# Technically, the only required environment variable is JAVA_HOME.
# All others are optional. However, the defaults are probably not
# preferred. Many sites configure these options outside of Hadoop,
# such as in /etc/profile.d
#
# The java implementation to use. By default, this environment
# variable is REQUIRED on all platforms except OS X!
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
#
# Location of Hadoop. By default, Hadoop will attempt to determine
# this location based upon its execution path.
# export HADOOP_HOME=
#
# Location of Hadoop's configuration information. i.e., where this
# file is living. If this is not defined, Hadoop will attempt to
# locate it based upon its execution path.
#
# NOTE: It is recommend that this variable not be set here but in
# /etc/profile.d or equivalent. Some options (such as
# --config) may react strangely otherwise.
#
# export HADOOP_CONF_DIR=${HADOOP_HOME}/etc/hadoop
#
# The maximum amount of heap to use (Java -Xmx). If no unit
# is provided, it will be converted to MB. Daemons will
# prefer any Xmx setting in their respective _OPT variable.
# There is no default; the JVM will autoscale based upon machine
# memory size.
# export HADOOP_HEAPSIZE_MAX=
#
# The minimum amount of heap to use (Java -Xms). If no unit
# is provided, it will be converted to MB. Daemons will
```

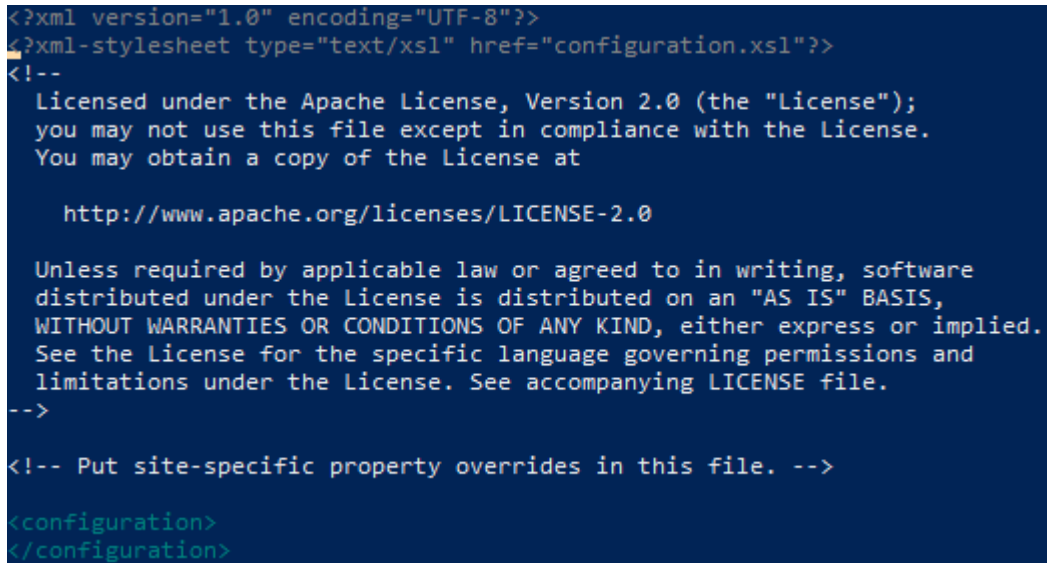
Figure 6.3: Change JAVA\_HOME directory path



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 screenshot of a terminal window showing the nano text editor editing the file \$HADOOP\_HOME/etc/hadoop/core-site.xml. The file content is XML, starting with an XML declaration and a license notice. The license notice states it is under the Apache License, Version 2.0, and provides a URL to the license. It also mentions that software is distributed on an "AS IS" BASIS without warranties. The file ends with a closing tag for the configuration section. The nano editor's status bar at the bottom shows the file path and the current line and column numbers.

```
<?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>
</configuration>
```

**Figure 6.4:** Open core-site.xml file

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

A screenshot of a terminal window showing the XML code to be added to the core-site.xml file. The code is enclosed in a blue box and shows the opening and closing tags for a property element, with the name 'fs.defaultFS' and the value 'hdfs://localhost:9000'.

```
<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)

```

<?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.defaultFS</name>
  <value>hdfs://localhost:9000</value>
</property>
</configuration>

```

**Figure 6.5:** Config *core-site.xml* file

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}
```

```
khtn_22120210@foo:~$ sudo mkdir -p /home/hadoop/hdfs/{namenode,datanode}
```

**Figure 6.6:** Create directory to store metadata

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

```
$ sudo chown -R khtn_22120210 /home/hadoop/hdfs
```

```
khtn_22120210@foo:~$ sudo chown -R khtn_22120210 /home/hadoop/hdfs
```

**Figure 6.7:** Grant the newly created directory to the “khtn\_22120210” user

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>
```

```

<?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>

```

**Figure 6.8:** Config *hdfs-site.xml* file

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>
```

```
<configuration>
<property>

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

  <value>yarn</value>

</property>
</configuration>
```

**Figure 6.9:** Config *mapred-site.xml* file

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>
```

```
<!-- Site specific YARN configuration properties -->
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>
</configuration>
```

**Figure 6.10:** Config *yarn-site.xml* file

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**

To start YARN and DFS, using command:

```
$ start-dfs.sh
```

```
$ start-yarn.sh
```

```
khtn_22120210@bigdata:~$ start-dfs.sh
Starting namenodes on [0.0.0.0]
Starting datanodes
Starting secondary namenodes [bigdata]
khtn_22120210@bigdata:~$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
```

**Figure 7.1:** Start YARN and DFS

To verify running services, using command:

```
$ jps
```

```
khtn_22120210@bigdata:~$ jps
10562 DataNode
11107 NodeManager
11463 Jps
10986 ResourceManager
10443 NameNode
10717 SecondaryNameNode
```

**Figure 7.2:** Running Java service

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 get access to Hadoop web services. As shown below, my instance's IPv4 address is **172.22.235.75**)

```
PS C:\Windows\system32> multipass list
Name      State      IPv4      Image
bigdata   Running    172.22.235.75  Ubuntu 24.04 LTS
```

**Figure 7.3:** Multipass instance's IPv4 address

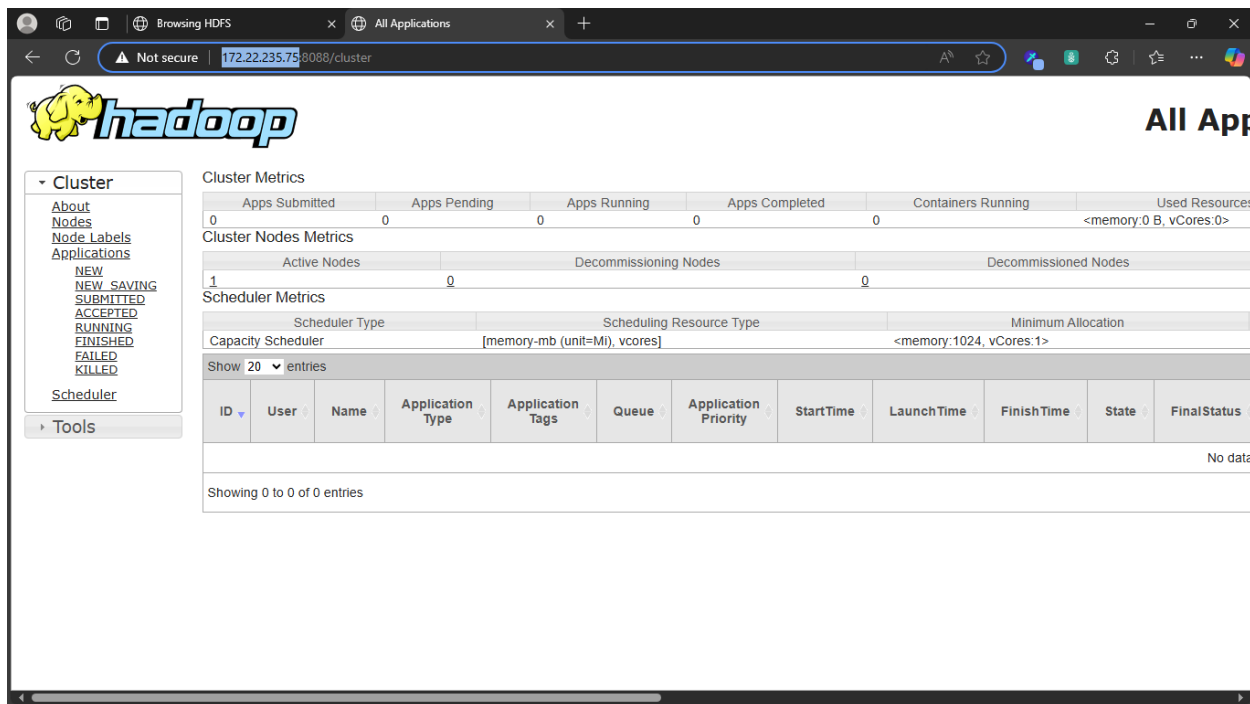
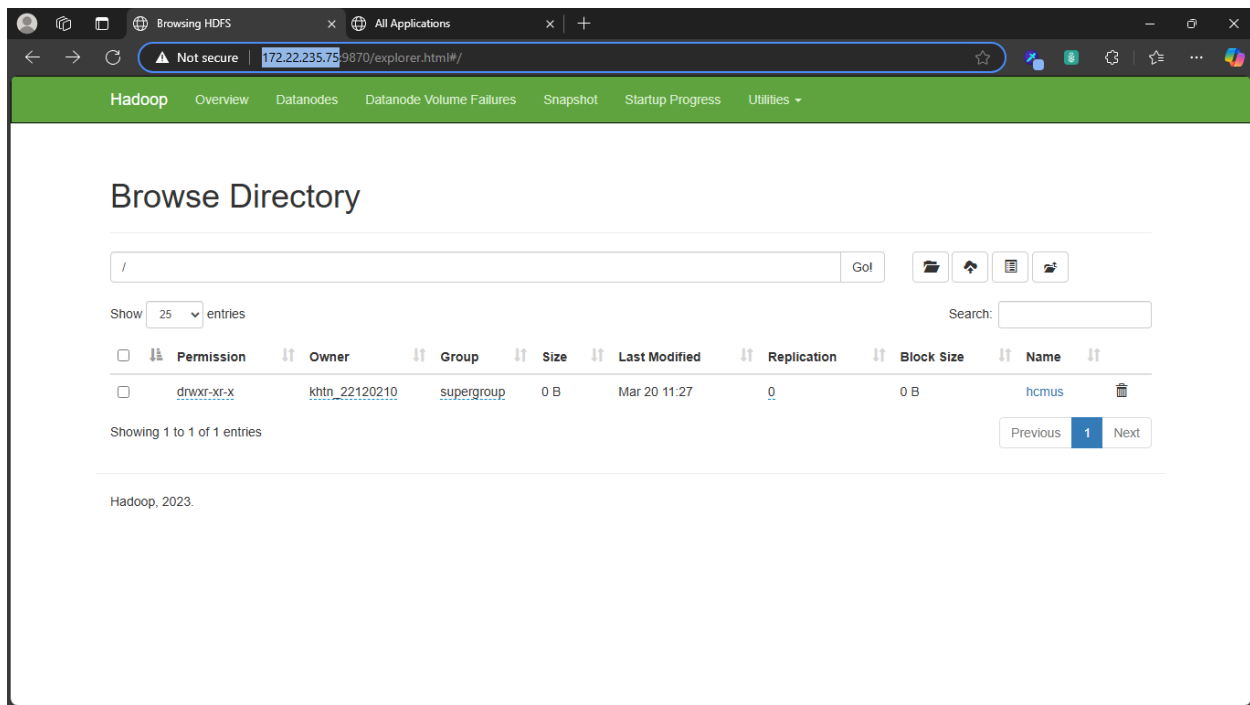


Figure 7.4a & 7.4b: Hadoop web services' UI

## 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
```

```
khtn_22120210@bigdata: $ hdfs dfs -mkdir /hcmus  
khtn_22120210@bigdata: $ hdfs dfs -mkdir /hcmus/22120210
```

**Figure 8.1:** Create directory on DFS

Verify the owner of the directory:

```
khtn_22120210@bigdata: $ hdfs dfs -ls -d /hcmus/22120210/  
drwxr-xr-x - khtn_22120210 supergroup 0 2025-03-20 11:33 /hcmus/22120210
```

**Figure 8.2:** Directory's information

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 privilege and type the following command (for ease of using, you can also use mount<sup>[2]</sup>):

```
multipass transfer "E:\hadoop-test.jar" bigdata:/home/ubuntu[3]
```

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

```
$ sudo mv hadoop-test.jar /home/khtn_22120210
```

Verifying the transfer:

```
khtn_22120210@bigdata:~$ ls  
hadoop-test.jar
```

**Figure 8.3:** Verifying the transfer

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

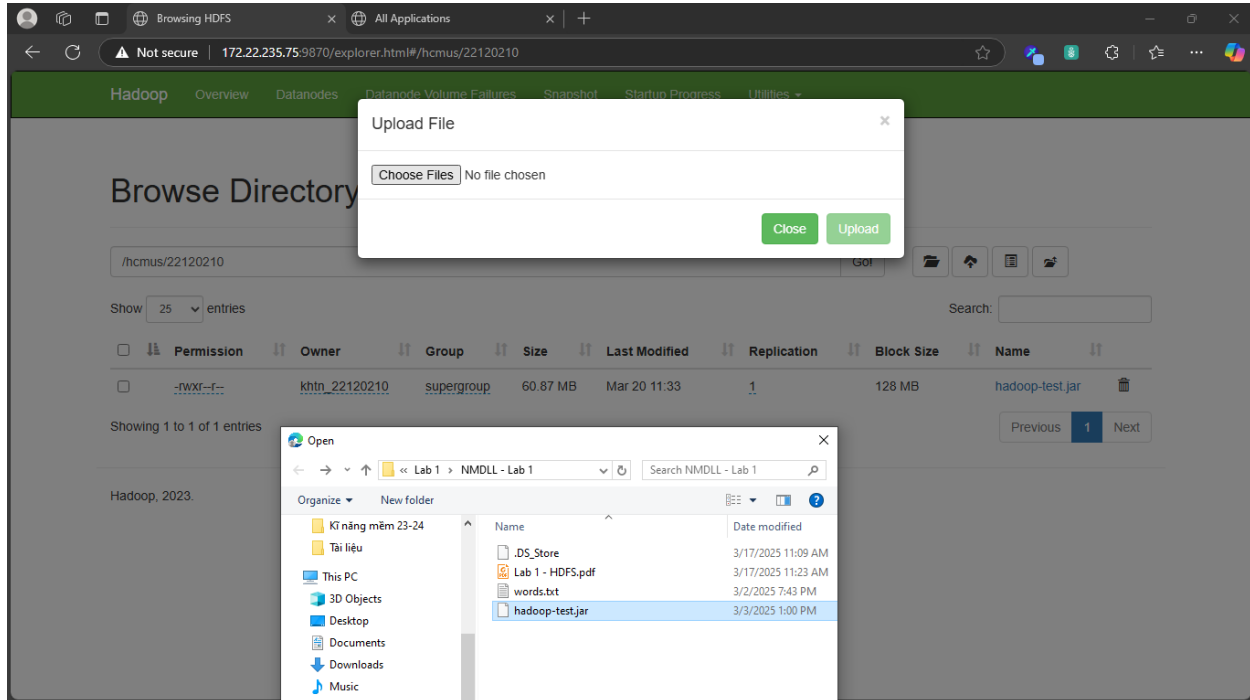
```
$ hdfs dfs -put hadoop-test.jar /hcmus/22120210
```

Verify if the file was uploaded successfully:

```
khtn_22120210@bigdata:~$ hdfs dfs -ls /hcmus/22120210  
Found 1 items  
-rw-r--r-- 1 khtn_22120210 supergroup 63827503 2025-03-20 11:05 /hcmus/22120210/hadoop-test.jar
```

**Figure 8.4:** Verifying the upload

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



**Figure 8.5:** Upload file using web service UI

Change the file permission to execute the file on hdfs:

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

## Step 9: Run the programme

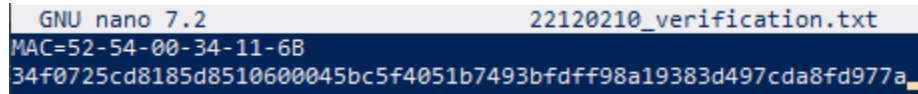
To execute the file, use command:

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

```
khtn_22120210@bigdata:~$ java -jar hadoop-test.jar 9000 /hcmus/22120210
Trying to read /hcmus/22120210
log4j:WARN No appenders could be found for logger (org.apache.hadoop.util.Shell).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Found hdfs://localhost:9000/hcmus/22120210/hadoop-test.jar
Your student ID: 22120210 (ensure it matches your student ID)
The first method to get MAC address is failed: Could not get network interface
Trying the alternative method
The first method to get MAC address is failed: Could not get network interface
Trying the alternative method
File written at /home/khtn_22120210/22120210_verification.txt
```

**Figure 9.1:** Output after executing the *hadoop-test.jar* file

Verifying the result:



```
GNU nano 7.2 22120210_verification.txt
MAC=52-54-00-34-11-6B
34f0725cd8185d8510600045bc5f4051b7493bfdff98a19383d497cda8fd977a_
```

**Figure 9.2:** The content of *22120210\_verification.txt* file

## II. Warm up with Word Count

### 1. Word Count Mapper.

```
letters_set = {'a', 'f', 'g', 'j', 'h', 'c', 'm', 'u', 's'}

for line in sys.stdin:
    words = re.split(r"[^a-zA-Z]+", line.strip())

    for word in words:
        if word and word[0].lower() in letters_set:
            print(f"{word[0].lower()}\t1")
```

First, we create a set containing the specified letters for later use.

Next, we read each line from the system input (**sys.stdin**) and process it.

The line **re.split(r"[^a-zA-Z]+", line.strip())** splits the input line into a list of words, using any non-alphabetical character (such as numbers, punctuation, or spaces) as a delimiter.

After splitting the words, we iterate through each one. If the word is not empty and its first letter (converted to lowercase) is in the predefined set, we print the first letter followed by the number 1, separated by a tab (**\t**).

## 2. Word Count Reducer.

```
desired_order = ['a', 'f', 'j', 'g', 'h', 'c', 'm', 'u', 's']
result = OrderedDict((key, 0) for key in desired_order)

for line in sys.stdin:
    letter, cnt = line.split('\t', 1)
    try:
        cnt = int(cnt)
    except ValueError:
        continue

    if letter in result:
        result[letter] += cnt

for letter in desired_order:
    if result[letter] > 0:
        print(f"{letter}\t{result[letter]}")
```

First, we create a list of letters in the desired order for output.

Next, we initialize an ordered dictionary where each key is a letter from the predefined list, and all values are initially set to zero.

Then, we iterate through each line of the system input, which comes from the mapper output. Each line contains a letter and a count, separated by a tab (`\t`). We attempt to convert the count into an integer, and if successful, we update the corresponding value in the dictionary.

Finally, we print the results in the predefined order, only including letters that have a count greater than zero.

### **3. Word Count Result.**

Here is the result:

|   |       |
|---|-------|
| a | 32921 |
| f | 18793 |
| j | 4530  |
| g | 16002 |
| h | 20911 |
| c | 42817 |
| m | 27239 |
| u | 24301 |
| s | 59567 |

### III. References:

- [1] Iqbal, M. (2024, January 9). Apache Hadoop 3.3.6 installation on Ubuntu 22.04.2 LTS WSL for Windows. Retrieved 2025, March 17<sup>th</sup>, from *Medium*. <https://medium.com/@madihaiqbal606/apache-hadoop-3-3-6-installation-on-ubuntu-22-04-2-lts-wsl-for-windows-bb57ed599bc6>
- [2] Abstract programmer. (2021, April 29). *Ubuntu Multipass tutorial for Linux* [Video]. Retrieved 2025, March 24<sup>th</sup>, from YouTube. <https://www.youtube.com/watch?v=Ky21G-ilMok>
- [3] Ltd, C. (n.d.-a). ``multipass transfer` command` | *Canonical*. Retrieved 2025, March 20<sup>th</sup>, from Canonical. <https://canonical.com/multipass/docs/transfer-command>