VIET NAM NATIONAL UNIVERSITY HO CHI MINH CITY

UNIVERSITY OF SCIENCE

Lab 01: Introduction to Hadoop Ecosystem

Full Name: Nguyễn Hữu Nghĩa

Student ID: 22120227

Class: Nhập môn dữ liệu lớn - CQ2022/21

Instructor: Huỳnh Lâm Hải Đăng | hlhdang@fit.hcmus.edu.vn

Table of Contents

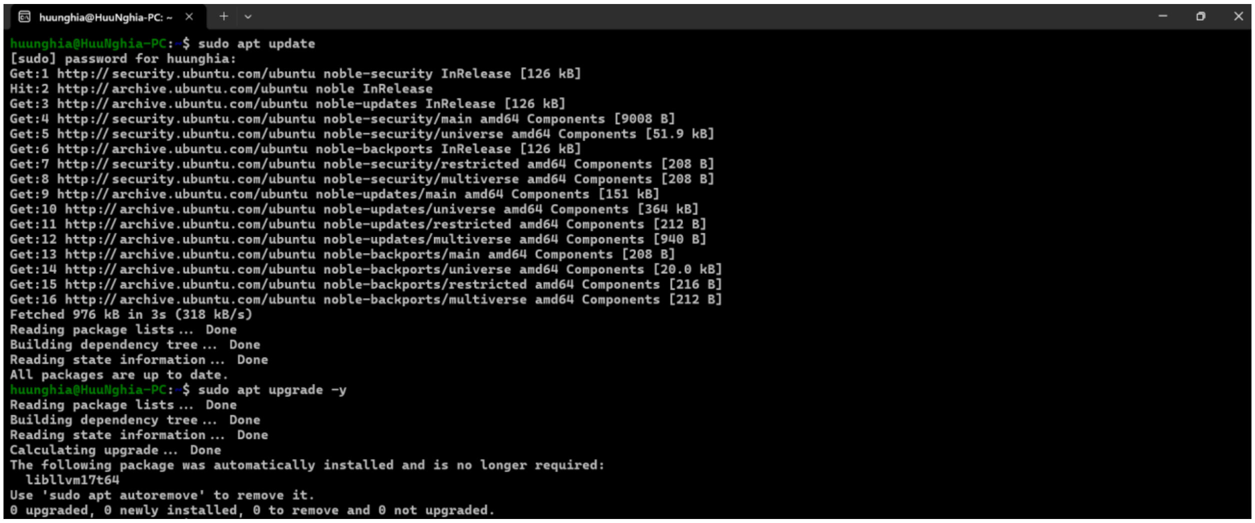
1/ Setup Hadoop Cluster Tutorial

a/ Install Hadoop (WSL[1])

Step 1: System Preparation

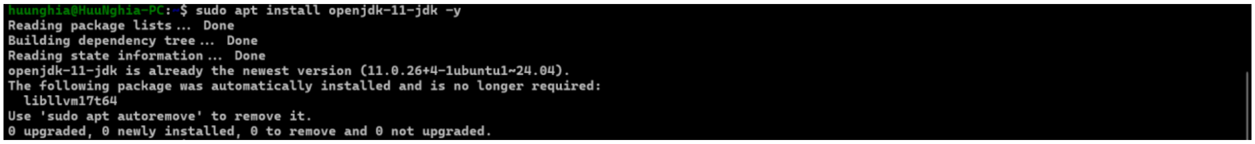
Ensure your system is updated and Java is installed.

- sudo apt update: This command fetches the latest package information from all configured sources.
- sudo apt upgrade -y: Installs the latest versions of all packages currently installed.



```
huunghia@HuuNghia-PC:~$ sudo apt update
[sudo] password for huunghia:
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:2 http://archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [9008 B]
Get:5 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [51.9 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [208 B]
Get:8 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:9 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [151 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [364 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:12 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:13 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:14 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [20.0 kB]
Get:15 http://archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:16 http://archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Fetched 976 kB in 3s (318 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
huunghia@HuuNghia-PC:~$ sudo apt upgrade -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
  libllvm17t64
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

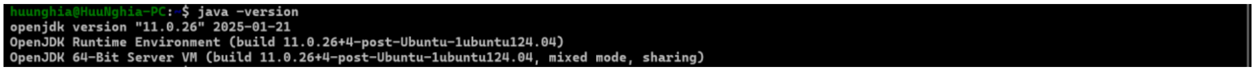
- sudo apt install openjdk-11-jdk -y: Installs OpenJDK 11, a requirement for Hadoop.



```
huunghia@HuuNghia-PC:~$ sudo apt install openjdk-11-jdk -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openjdk-11-jdk is already the newest version (11.0.26+4-1ubuntu1~24.04).
The following package was automatically installed and is no longer required:
  libllvm17t64
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Verify Java installation:

- java -version: Confirms Java is installed and displays the version.

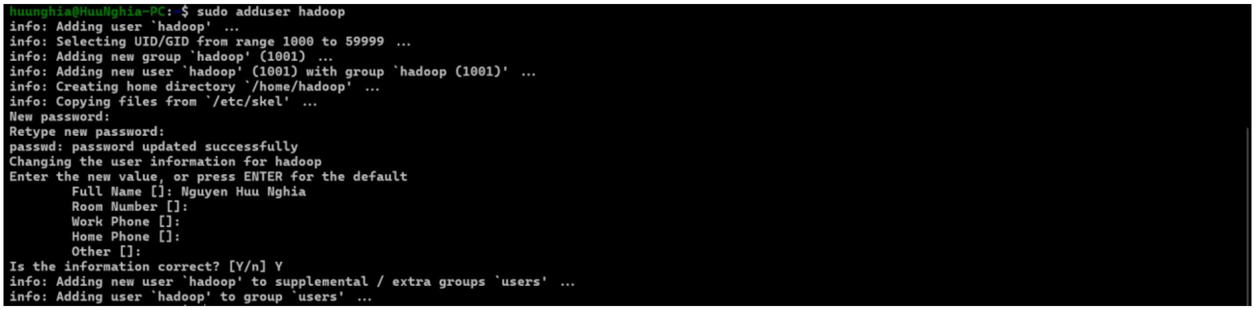


```
huunghia@HuuNghia-PC:~$ java -version
openjdk version "11.0.26" 2025-01-21
OpenJDK Runtime Environment (build 11.0.26+4-post-Ubuntu-1ubuntu124.04)
OpenJDK 64-Bit Server VM (build 11.0.26+4-post-Ubuntu-1ubuntu124.04, mixed mode, sharing)
```

Step 2: Create Hadoop User and Configure SSH

Create a dedicated Hadoop user:

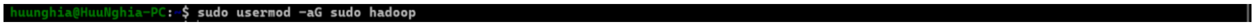
- sudo adduser hadoop: Creates a new user named "hadoop" with its own home directory.



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

Grant superuser privileges:

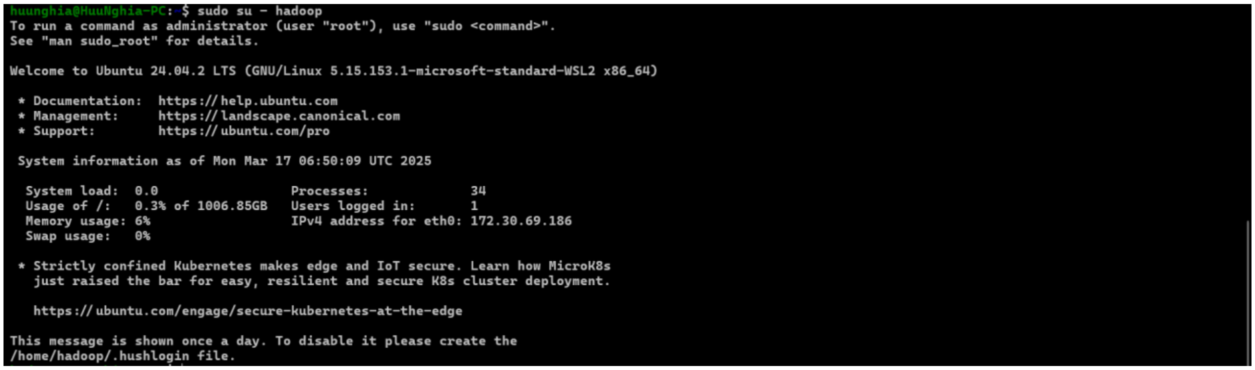
- sudo usermod -aG sudo hadoop: Adds the new user to the sudo group to allow administrative commands.



```
huunghia@HuuNghia-PC:~$ sudo usermod -aG sudo hadoop
```

Switch to the Hadoop user:

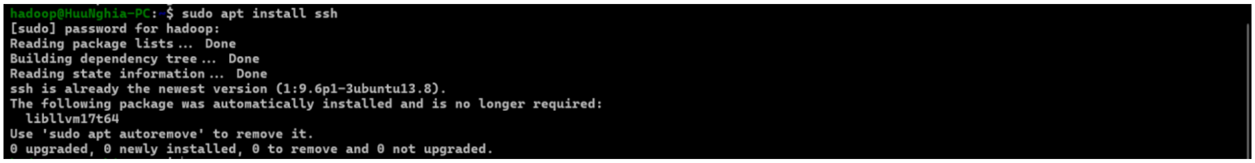
- sudo su - hadoop: Switches to the Hadoop user.



```
huunghia@HuuNghia-PC:~$ sudo su - hadoop
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 5.15.153.1-microsoft-standard-WSL2 x86_64)

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

 System information as of Mon Mar 17 06:50:09 UTC 2025

  System load:  0.0                 Processes:             34
  Usage of /:   0.3% of 1006.85GB   Users logged in:       1
  Memory usage: 6%                  IPv4 address for eth0: 172.30.69.186
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

This message is shown once a day. To disable it please create the
/home/hadoop/.hushlogin file.
```

Install OpenSSH:

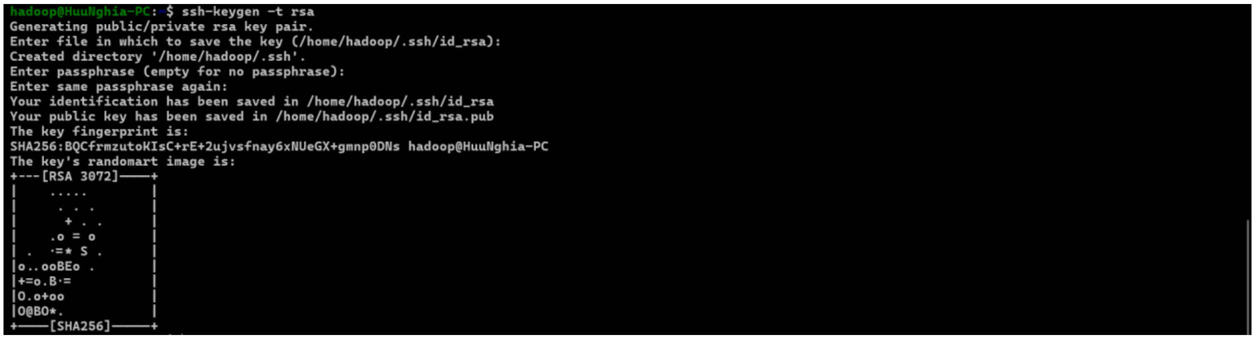
- sudo apt install ssh: Installs SSH client and server for remote and local communication.



```
hadoop@HuuNghia-PC:~$ sudo apt install ssh
[sudo] password for hadoop:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ssh is already the newest version (1:9.6p1-3ubuntu13.8).
The following package was automatically installed and is no longer required:
  libllvm17t64
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Generate SSH keys:

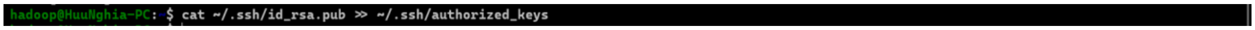
- ssh-keygen -t rsa: Generates a pair of RSA keys (private and public). Press Enter to save in the default location, and leave the passphrase empty for easier access.



```
hadoop@HuuNghia-PC:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Created directory '/home/hadoop/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
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:BQCfrmzutoKIsC+rE+2ujvsfnay6xNUeGX+gmnp0DNs hadoop@HuuNghia-PC
The key's randomart image is:
+---[RSA 3072]----+
|     .....       |
|      . . .      |
|       + . .     |
|     .o = o      |
| .   ·=* S .     |
|o..ooBEo .       |
|+=o.B·=          |
|O.o+oo           |
|O@BO*.           |
+----[SHA256]-----+
```

Add the public key to authorized keys:

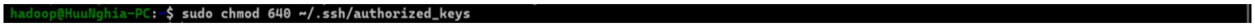
- cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys: Appends the public key to the list of authorized keys, allowing passwordless SSH.



```
hadoop@HuuNghia-PC:~$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

Set permissions on the key file:

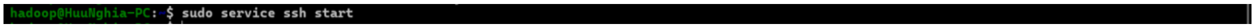
- sudo chmod 640 ~/.ssh/authorized_keys: Retricts access to the authorized keys file for security.



```
hadoop@HuuNghia-PC:~$ sudo chmod 640 ~/.ssh/authorized_keys
```

Start the SSH service:

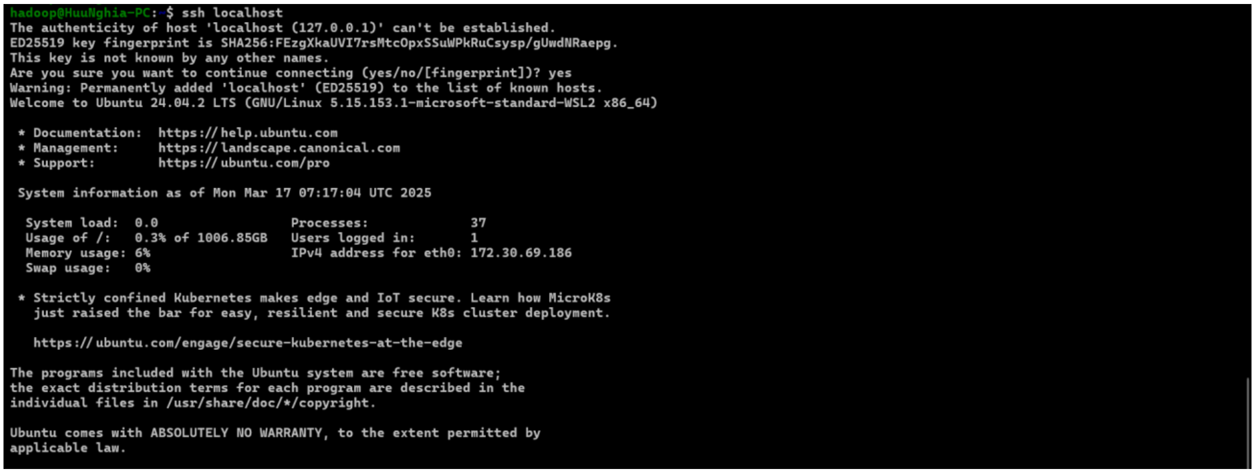
- sudo service ssh start: Starts the SSH service to allow connections.



```
hadoop@HuuNghia-PC:~$ sudo service ssh start
```

Confirm the SSH configuration:

- ssh localhost: Tests the SSH connection locally

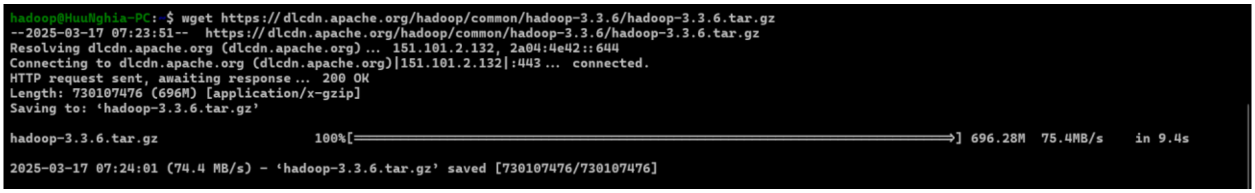


```
hadoop@HuuNghia-PC:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ED25519 key fingerprint is SHA256:FEzgXkaUVI7rsMtcOpxSSuWPkRuCsysp/gUwdNRaepg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 5.15.153.1-microsoft-standard-WSL2 x86_64)

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

 System information as of Mon Mar 17 07:17:04 UTC 2025

  System load:  0.0                Processes:             37
  Usage of /:   0.3% of 1006.85GB   Users logged in:       1
  Memory usage: 6%                 IPv4 address for eth0: 172.30.69.186
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

Step 3: Download and install Apache Hadoop

Download Hadoop:

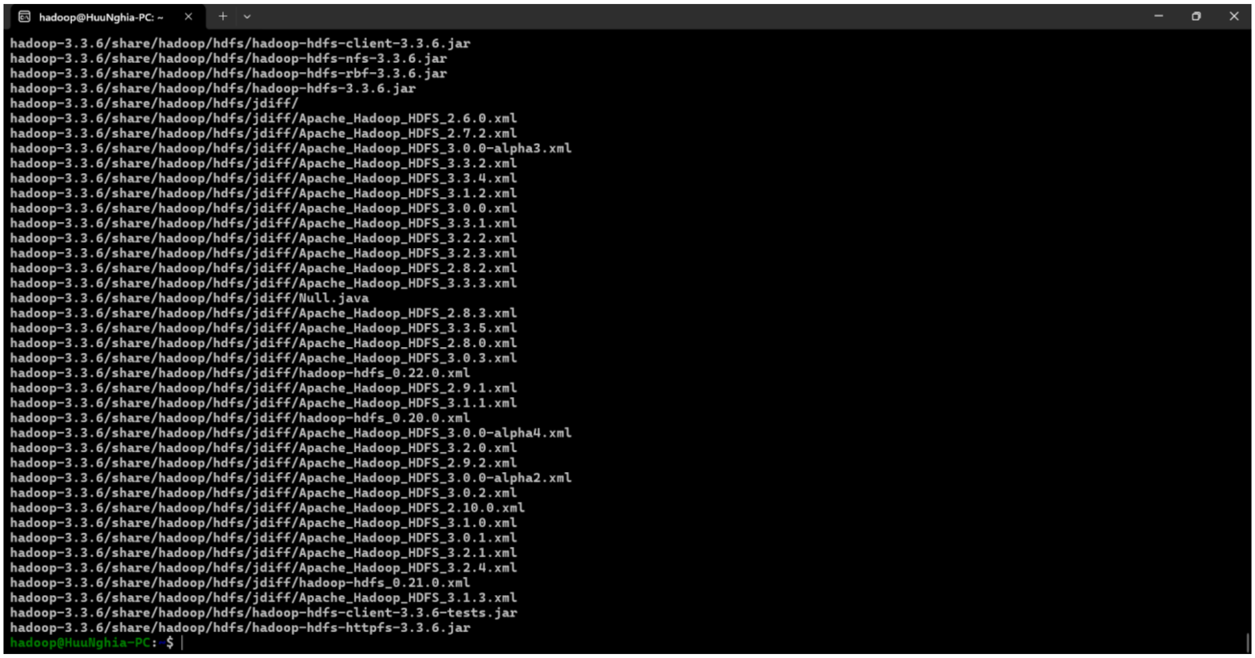
- wget https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz



```
hadoop@HuuNghia-PC:~$ wget https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz
--2025-03-17 07:23:51--  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  75.4MB/s    in 9.4s

2025-03-17 07:24:01 (74.4 MB/s) - ‘hadoop-3.3.6.tar.gz’ saved [730107476/730107476]
```

Extract the file:

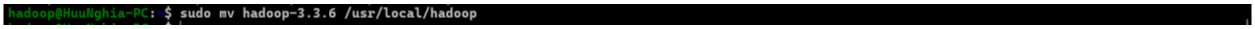
- sudo tar -xvzf hadoop-3.3.6.tar.gz: Extracts the downloaded tar.gz file.



```
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_3.3.2.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_3.0.0.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
hadoop@HuuNghia-PC:~$ |
```

Move the extracted files:

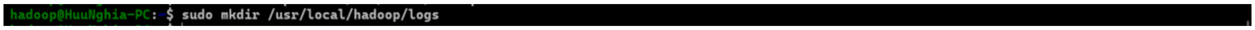
- sudo mv hadoop-3.3.6 /usr/local/hadoop: Moves and renames the Hadoop folder for easier access.



```
hadoop@HuuNghia-PC:~$ sudo mv hadoop-3.3.6 /usr/local/hadoop
```

Create a directory for logs:

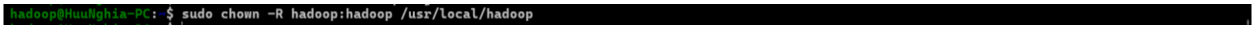
- sudo mkdir /usr/local/hadoop/logs: Creates a directory to store Hadoop logs.



```
hadoop@HuuNghia-PC:~$ sudo mkdir /usr/local/hadoop/logs
```

Change ownership:

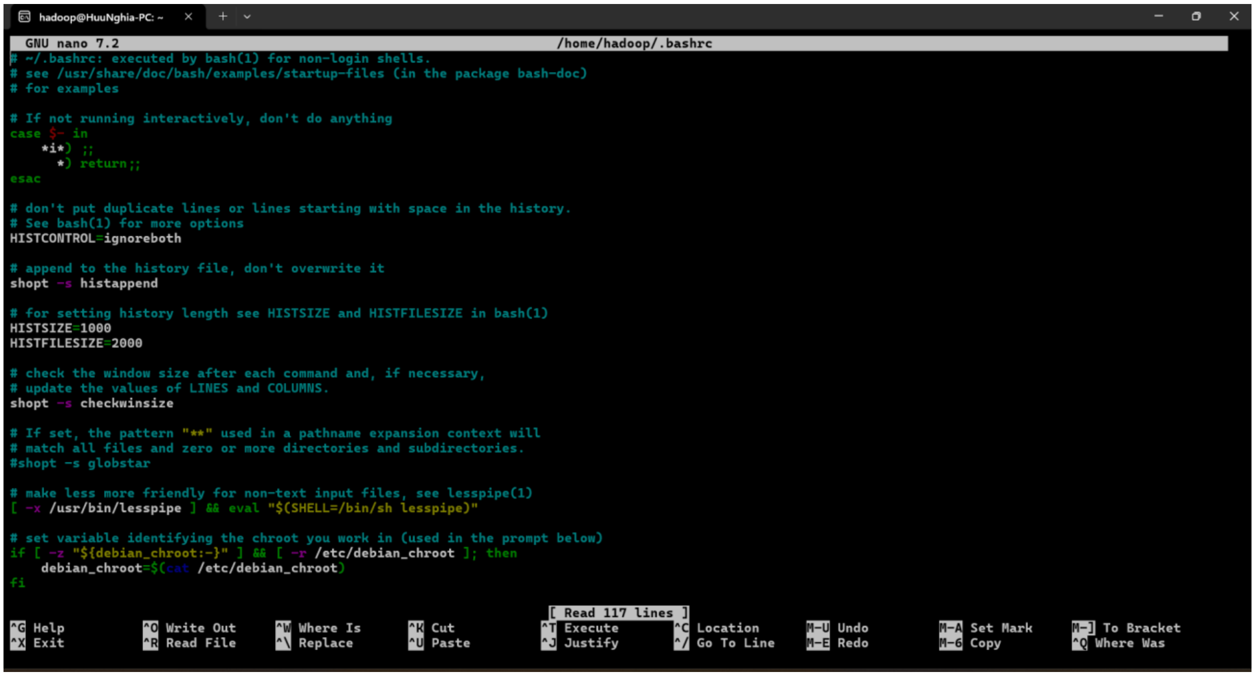
- sudo chown -R hadoop:hadoop /usr/local/hadoop: Alter the ownership of the */usr/local/hadoop* directory to the user *hadoop.*



```
hadoop@HuuNghia-PC:~$ sudo chown -R hadoop:hadoop /usr/local/hadoop
```

Configure Hadoop environment variables:

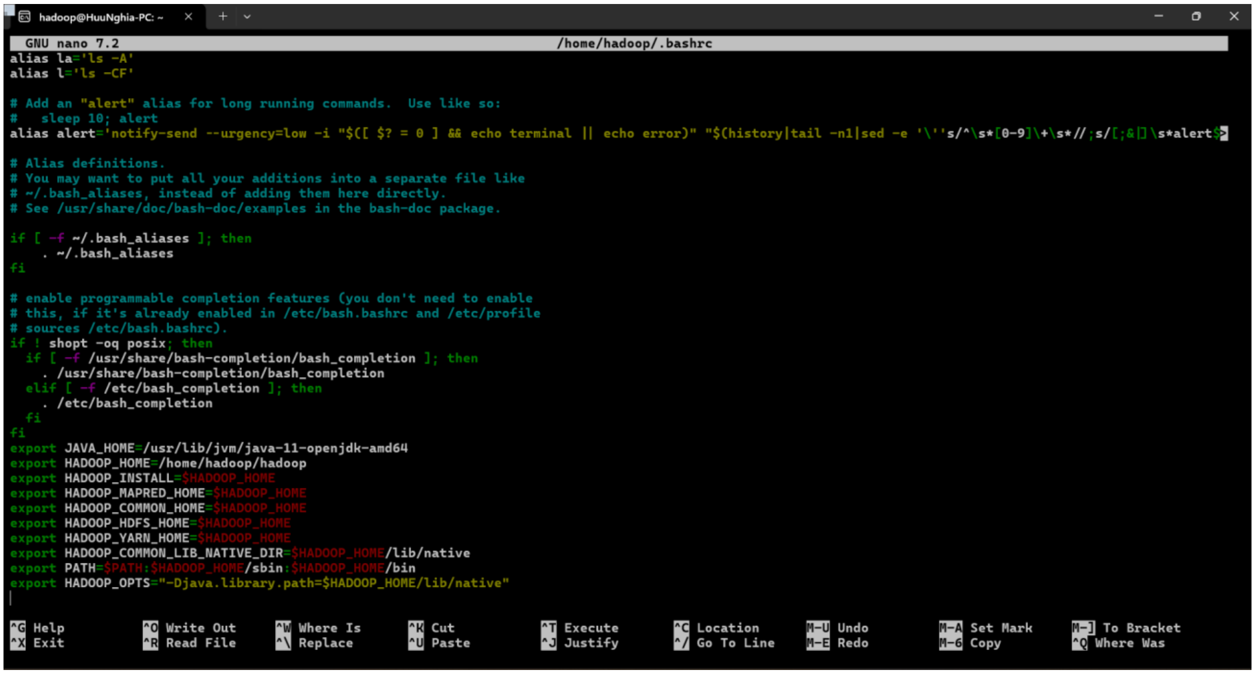
- sudo nano ~/.bashrc: Open the bashrc file.



```
  GNU nano 7.2                                    /home/hadoop/.bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *i*) ;;
      *) return;;
esac

# 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 ] && eval "$(SHELL=/bin/sh lesspipe)"

# set variable identifying the chroot you work in (used in the prompt below)
if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then
    debian_chroot=$(cat /etc/debian_chroot)
fi

                                    [ Read 117 lines ]
^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^Q Where Was
```

Navigate to the end of the file (Ctrl + /, then Ctrl + V), and add:

```
export JAVA_HOME=/usr/lib/jvm/java-11-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”
```

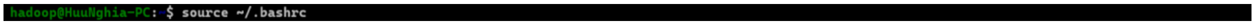


```
  GNU nano 7.2                                        /home/hadoop/.bashrc
alias la='ls -A'
alias l='ls -CF'

# 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=/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"


^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A Set Mark   M-] To Bracket
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo       M-6 Copy       ^Q Where Was
```

- Press **Ctrl+S** to save and press **Ctrl+X** to exit nano.

Enable the changes, source the *.bashrc* file:

source ~/.bashrc

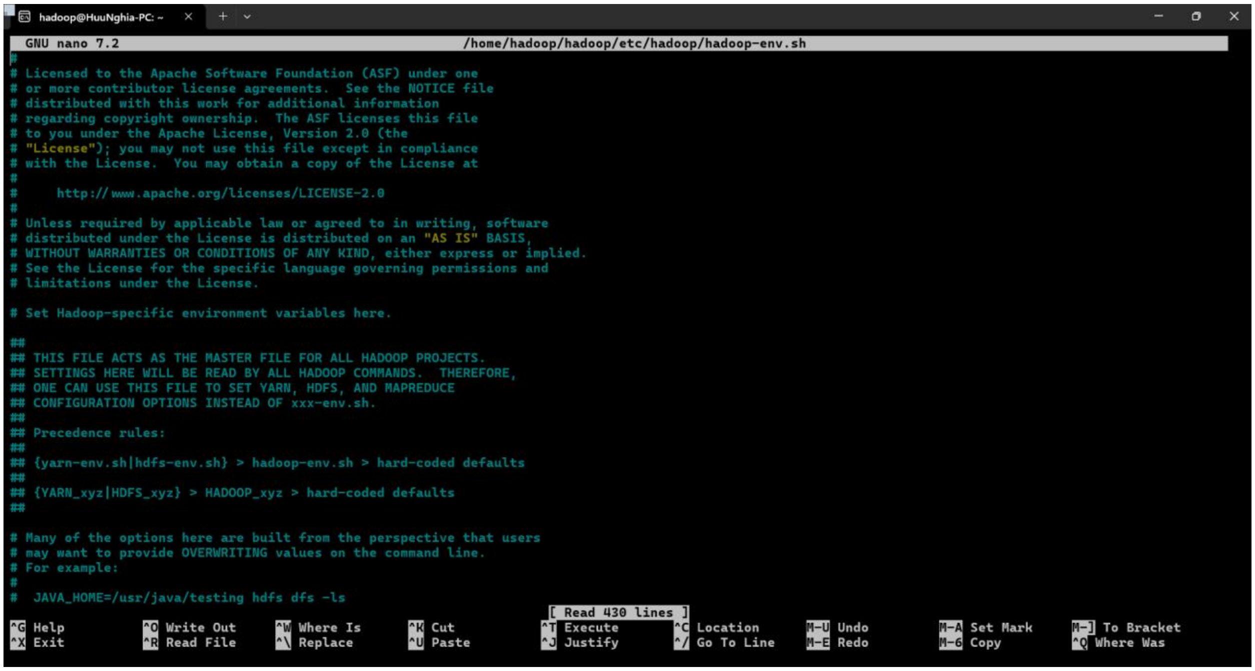


```
hadoop@HuuNghia-PC: $ source ~/.bashrc
```

Step 4: Configure java environment variables

Open the hadoop-env.sh file:

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

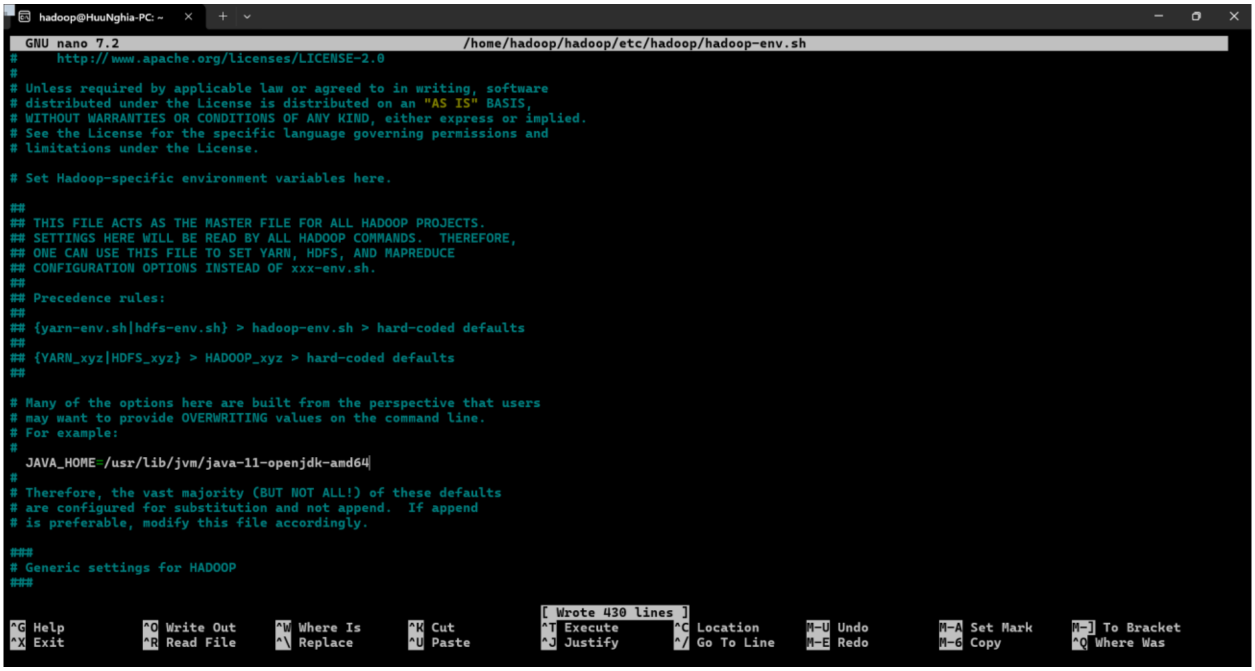


```
  GNU nano 7.2                                  /home/hadoop/hadoop/etc/hadoop/hadoop-env.sh
#
# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements.  See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership.  The ASF licenses this file
# to you 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.

# Set Hadoop-specific environment variables here.

##
## THIS FILE ACTS AS THE MASTER FILE FOR ALL HADOOP PROJECTS.
## SETTINGS HERE WILL BE READ BY ALL HADOOP COMMANDS.  THEREFORE,
## ONE CAN USE THIS FILE TO SET YARN, HDFS, AND MAPREDUCE
## CONFIGURATION OPTIONS INSTEAD OF xxx-env.sh.
##
## Precedence rules:
##
## {yarn-env.sh|hdfs-env.sh} > hadoop-env.sh > hard-coded defaults
##
## {YARN_xyz|HDFS_xyz} > HADOOP_xyz > hard-coded defaults
##

# Many of the options here are built from the perspective that users
# may want to provide OVERWRITING values on the command line.
# For example:
#
#  JAVA_HOME=/usr/java/testing hdfs dfs -ls
                                        [ Read 430 lines ]
^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^Q Where Was
```

Set the Java home path:

- Search for the line containing “export JAVA_HOME” and modify it (or add if missing):

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

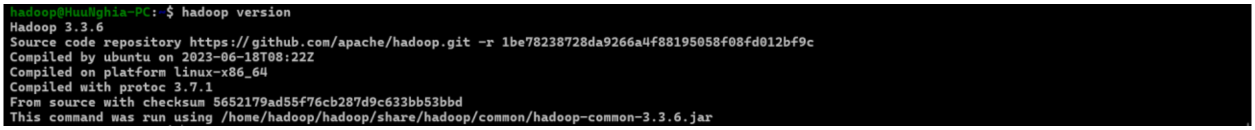


```
  GNU nano 7.2                                  /home/hadoop/hadoop/etc/hadoop/hadoop-env.sh
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# Set Hadoop-specific environment variables here.

##
## THIS FILE ACTS AS THE MASTER FILE FOR ALL HADOOP PROJECTS.
## SETTINGS HERE WILL BE READ BY ALL HADOOP COMMANDS.  THEREFORE,
## ONE CAN USE THIS FILE TO SET YARN, HDFS, AND MAPREDUCE
## CONFIGURATION OPTIONS INSTEAD OF xxx-env.sh.
##
## Precedence rules:
##
## {yarn-env.sh|hdfs-env.sh} > hadoop-env.sh > hard-coded defaults
##
## {YARN_xyz|HDFS_xyz} > HADOOP_xyz > hard-coded defaults
##

# Many of the options here are built from the perspective that users
# may want to provide OVERWRITING values on the command line.
# For example:
#
  JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
#
# 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
###
                                       [ Wrote 430 lines ]
^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^Q Where Was
```

Save and exit: Press **Ctrl+S** to save and **Ctrl+X** to exit.

Check the Hadoop version:

```
hadoop version
```

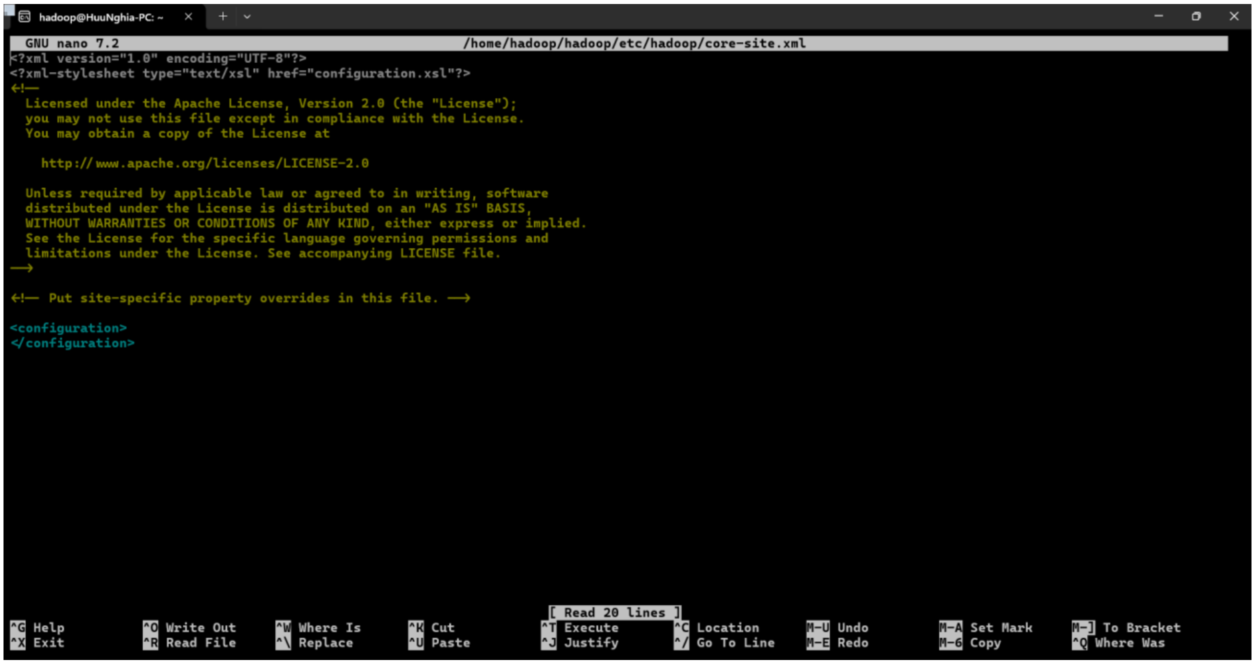


```
hadoop@HuuNghia-PC:~$ hadoop version
Hadoop 3.3.6
Source code repository https://github.com/apache/hadoop.git -r 1be78238728da9266a4f88195058f08fd012bf9c
Compiled by ubuntu on 2023-06-18T08:22Z
Compiled on platform linux-x86_64
Compiled with protoc 3.7.1
From source with checksum 5652179ad55f76cb287d9c633bb53bbd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

Step 5: Configure Hadoop

Open the core-site.xml file:

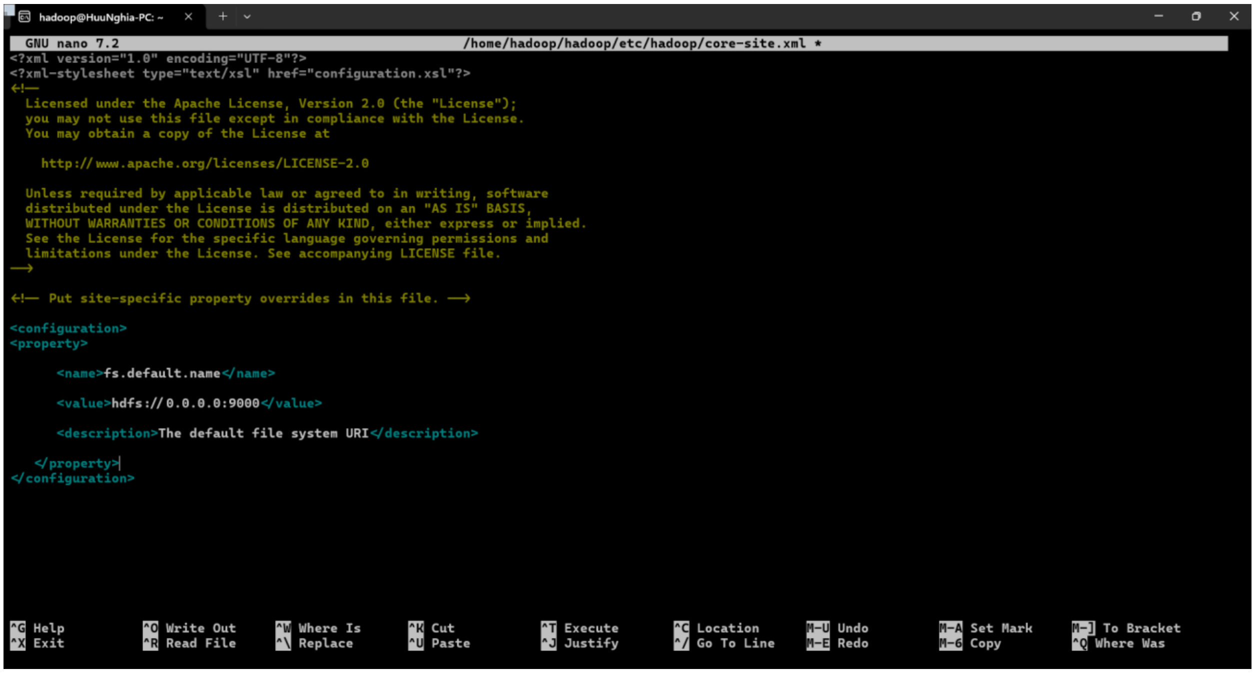
```
sudo nano $HADOOP_HOME/etc/hadoop/core-site.xml
```



```
  GNU nano 7.2                              /home/hadoop/hadoop/etc/hadoop/core-site.xml
<?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>
```

Add the following lines between *<Configuration> </Configuration>:*

```
<property>

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

      <value>hdfs://0.0.0.0:9000</value>

      <description>The default file system URI</description>

   </property>
```

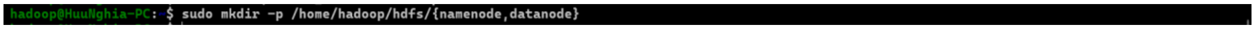


```
  GNU nano 7.2                                      /home/hadoop/hadoop/etc/hadoop/core-site.xml *
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file 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.default.name</name>

      <value>hdfs:// 0.0.0.0:9000</value>

      <description>The default file system URI</description>

   </property>
</configuration>
```

Save the changes and exit the editor.

Create directories for node metadata:

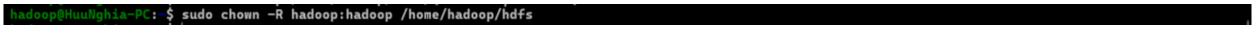
```
sudo mkdir -p /home/hadoop/hdfs/{namenode,datanode}
```



```
hadoop@HuuNghia-PC: $ sudo mkdir -p /home/hadoop/hdfs/{namenode,datanode}
```

Set ownership for Hadoop user:

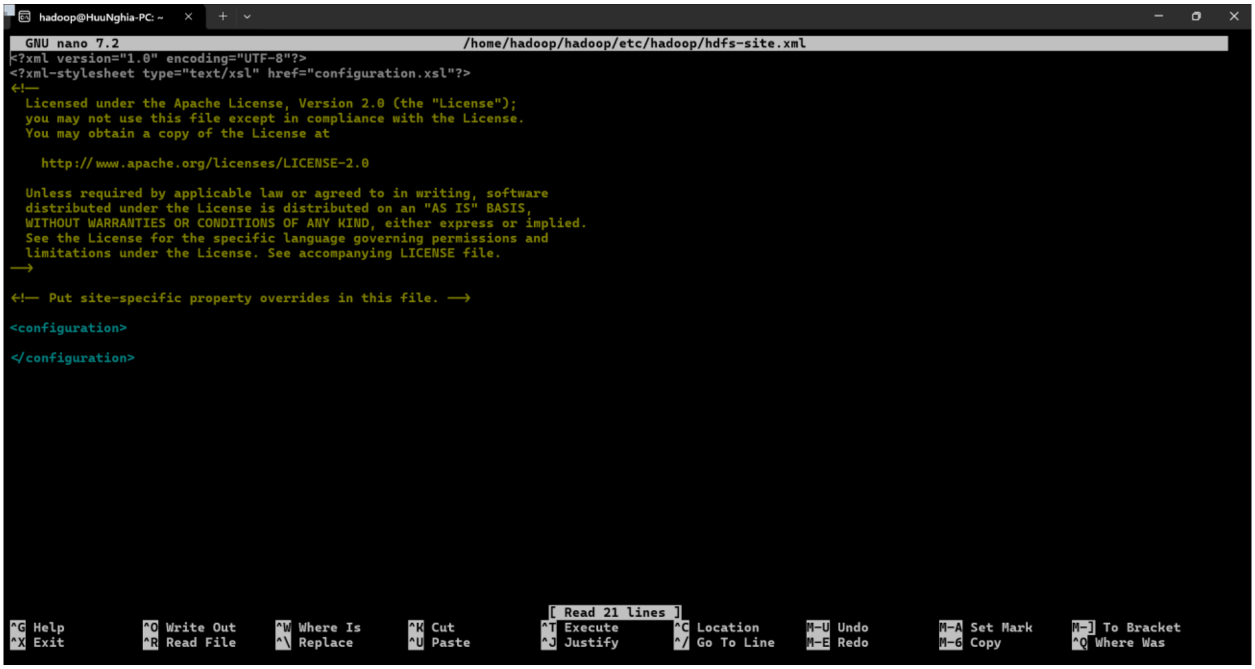
```
sudo chown -R hadoop:hadoop /home/hadoop/hdfs
```



```
hadoop@HuuNghia-PC: $ sudo chown -R hadoop:hadoop /home/hadoop/hdfs
```

Open the hdfs-site.xml file:

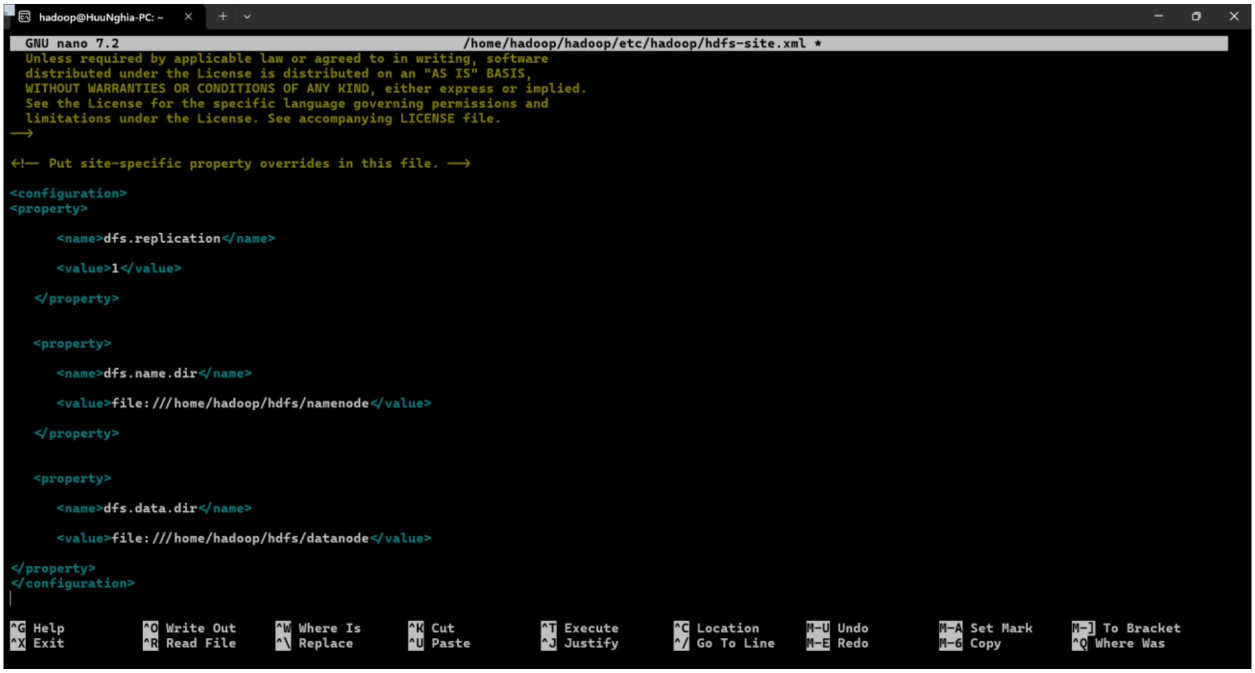
```
sudo nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml
```



```
  GNU nano 7.2                                  /home/hadoop/hadoop/etc/hadoop/hdfs-site.xml
<?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>

                                                               [ Read 21 lines ]
^G Help         ^O Write Out    ^W Where Is     ^K Cut          ^T Execute      ^C Location     M-U Undo        M-A Set Mark    M-] To Bracket
^X Exit         ^R Read File    ^\ Replace      ^U Paste        ^J Justify      ^/ Go To Line   M-E Redo        M-6 Copy        ^Q Where Was
```

Add the following lines 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>
```

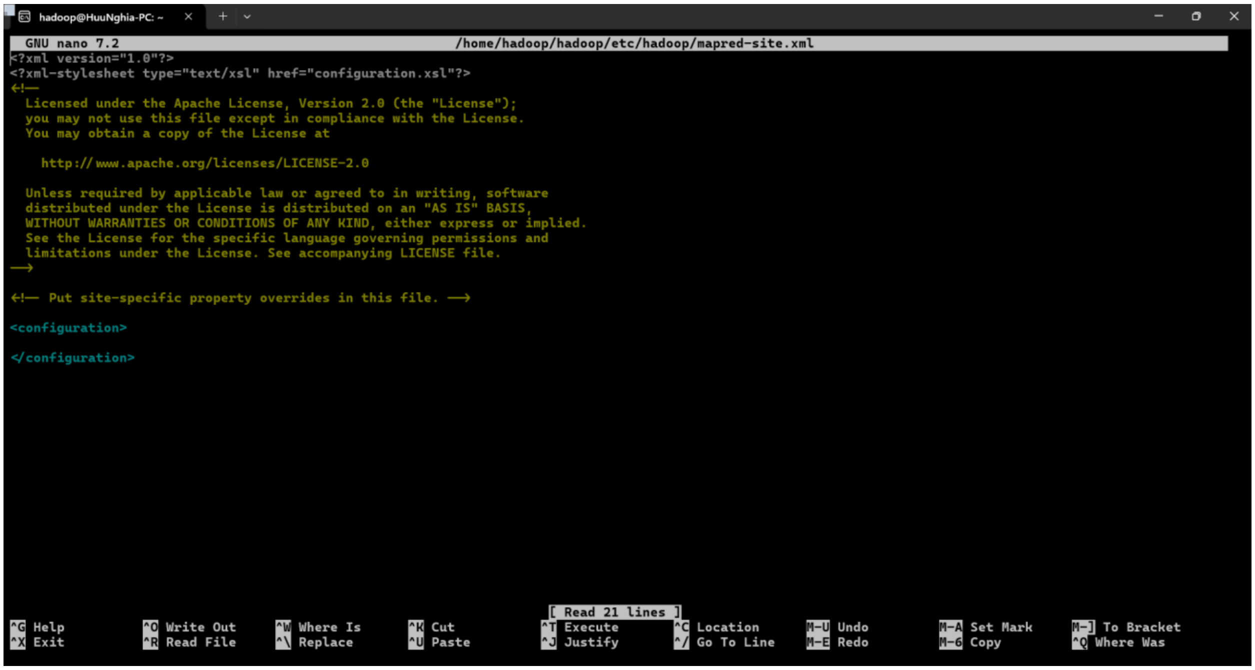


```
/home/hadoop/hadoop/etc/hadoop/hdfs-site.xml *
  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language 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>
```

Save the changes and exit the editor.

Open the *mapred-site.xml* file:

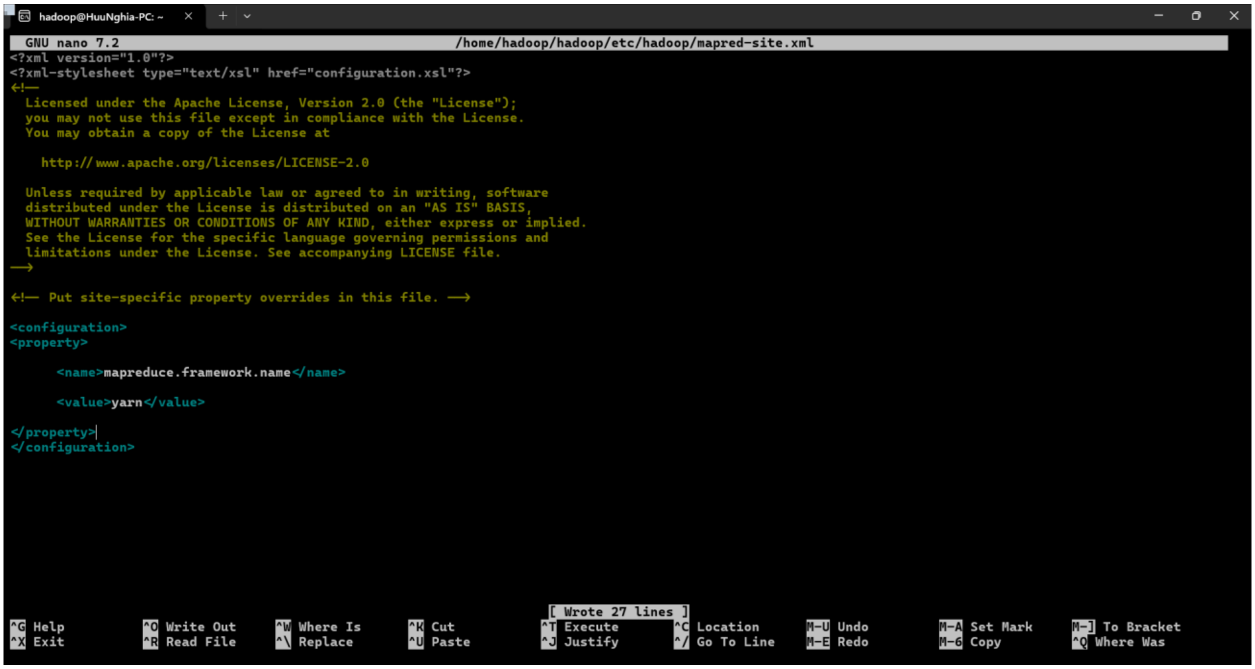
```
sudo nano $HADOOP_HOME/etc/hadoop/mapred-site.xml
```



```
/home/hadoop/hadoop/etc/hadoop/mapred-site.xml
<?xml version="1.0"?>
<?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>
```

And add the following lines between *<Configuration> </Configuration>:*

```
<property>

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

      <value>yarn</value>

</property>
```

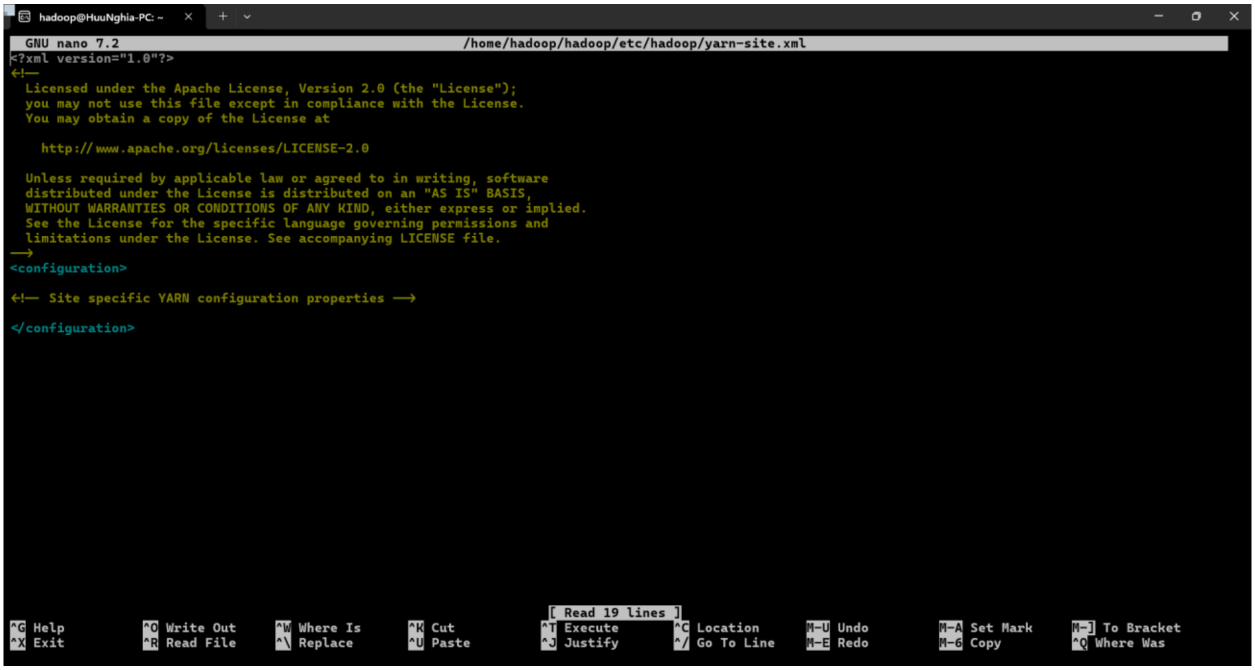


```
hadoop@HuuNghia-PC: ~
  GNU nano 7.2                              /home/hadoop/hadoop/etc/hadoop/mapred-site.xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except 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>mapreduce.framework.name</name>

      <value>yarn</value>

</property>
</configuration>

                                   [ Wrote 27 lines ]
^G Help         ^O Write Out    ^W Where Is     ^K Cut          ^T Execute      ^C Location     M-U Undo        M-A Set Mark    M-] To Bracket
^X Exit         ^R Read File    ^\ Replace      ^U Paste        ^J Justify      ^/ Go To Line   M-E Redo        M-6 Copy        ^Q Where Was
```

Save the changes and exit the editor.

Open the *yarn-site.xml* file:

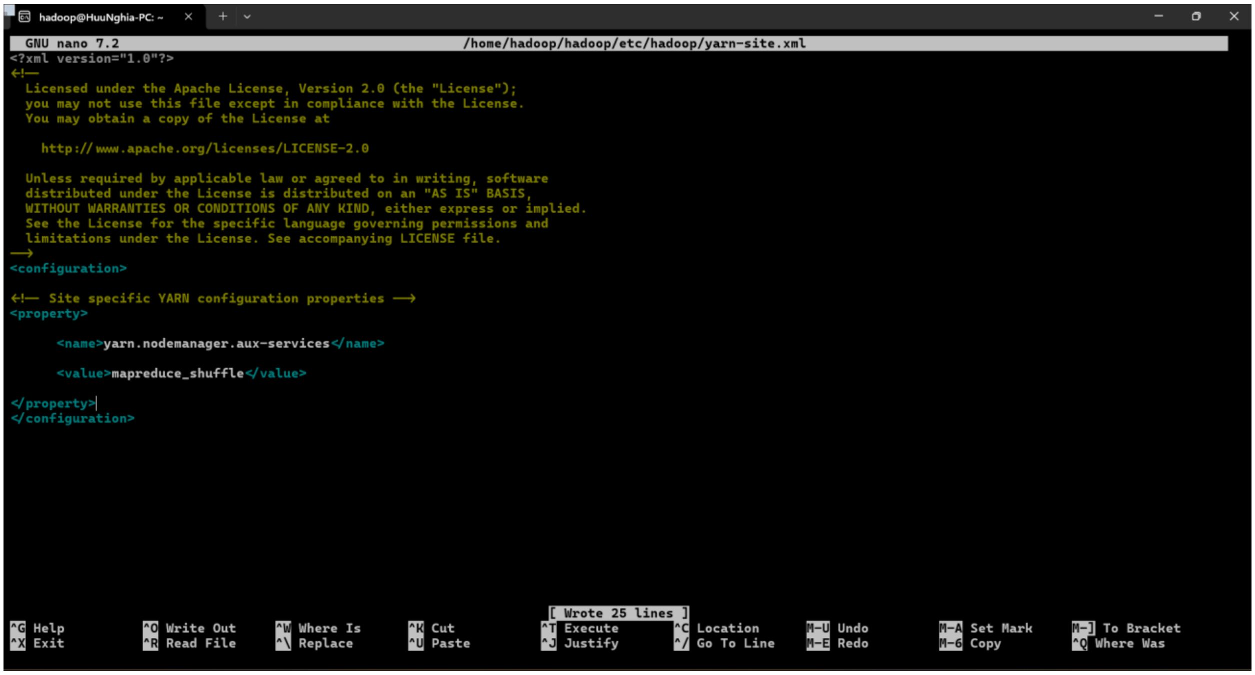
```
sudo nano $HADOOP_HOME/etc/hadoop/yarn-site.xml
```



```
  GNU nano 7.2                                        /home/hadoop/hadoop/etc/hadoop/yarn-site.xml
<?xml version="1.0"?>
<!--
  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.
-->
<configuration>

<!-- Site specific YARN configuration properties -->

</configuration>

                                                                   [ Read 19 lines ]
^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute       ^C Location      M-U Undo         M-A Set Mark     M-] To Bracket
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify       ^/ Go To Line    M-E Redo         M-6 Copy         ^Q Where Was
```

And add the following lines between <*Configuration*> </*Configuration*>

```
<property>

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

      <value>mapreduce_shuffle</value>

</property>
```

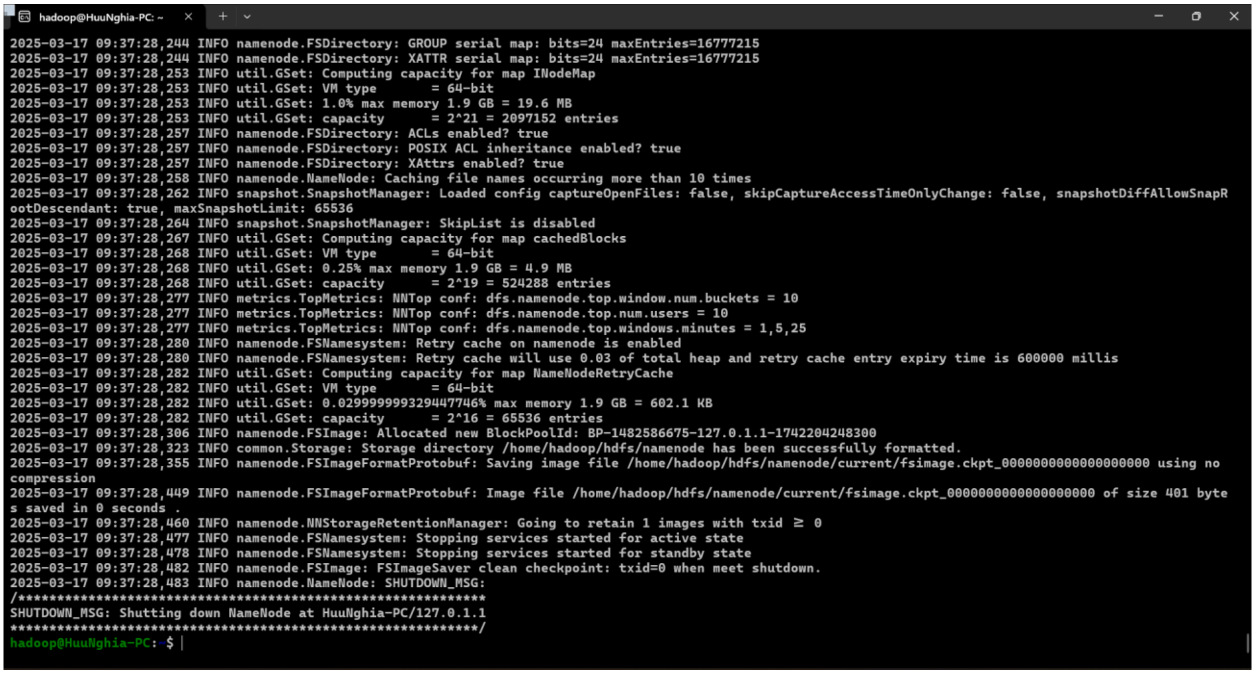


```
  GNU nano 7.2                                    /home/hadoop/hadoop/etc/hadoop/yarn-site.xml
<?xml version="1.0"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    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.
-->
<configuration>

<!-- Site specific YARN configuration properties -->
<property>

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

      <value>mapreduce_shuffle</value>

</property>
</configuration>

                                         [ Wrote 25 lines ]
^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^Q Where Was
```

Save the changes and exit the editor.

Format the HDFS NameNode:

hdfs namenode -format

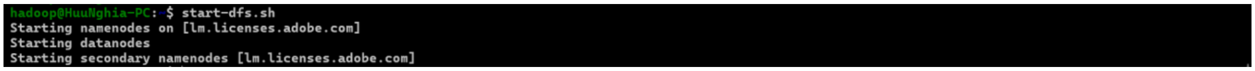


```
2025-03-17 09:37:28,244 INFO namenode.FSDirectory: GROUP serial map: bits=24 maxEntries=16777215
2025-03-17 09:37:28,244 INFO namenode.FSDirectory: XATTR serial map: bits=24 maxEntries=16777215
2025-03-17 09:37:28,253 INFO util.GSet: Computing capacity for map INodeMap
2025-03-17 09:37:28,253 INFO util.GSet: VM type       = 64-bit
2025-03-17 09:37:28,253 INFO util.GSet: 1.0% max memory 1.9 GB = 19.6 MB
2025-03-17 09:37:28,253 INFO util.GSet: capacity      = 2^21 = 2097152 entries
2025-03-17 09:37:28,257 INFO namenode.FSDirectory: ACLs enabled? true
2025-03-17 09:37:28,257 INFO namenode.FSDirectory: POSIX ACL inheritance enabled? true
2025-03-17 09:37:28,257 INFO namenode.FSDirectory: XAttrs enabled? true
2025-03-17 09:37:28,258 INFO namenode.NameNode: Caching file names occurring more than 10 times
2025-03-17 09:37:28,262 INFO snapshot.SnapshotManager: Loaded config captureOpenFiles: false, skipCaptureAccessTimeOnlyChange: false, snapshotDiffAllowSnapR
ootDescendant: true, maxSnapshotLimit: 65536
2025-03-17 09:37:28,264 INFO snapshot.SnapshotManager: SkipList is disabled
2025-03-17 09:37:28,267 INFO util.GSet: Computing capacity for map cachedBlocks
2025-03-17 09:37:28,268 INFO util.GSet: VM type       = 64-bit
2025-03-17 09:37:28,268 INFO util.GSet: 0.25% max memory 1.9 GB = 4.9 MB
2025-03-17 09:37:28,268 INFO util.GSet: capacity      = 2^19 = 524288 entries
2025-03-17 09:37:28,277 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
2025-03-17 09:37:28,277 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
2025-03-17 09:37:28,277 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
2025-03-17 09:37:28,280 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
2025-03-17 09:37:28,280 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 millis
2025-03-17 09:37:28,282 INFO util.GSet: Computing capacity for map NameNodeRetryCache
2025-03-17 09:37:28,282 INFO util.GSet: VM type       = 64-bit
2025-03-17 09:37:28,282 INFO util.GSet: 0.029999999329447746% max memory 1.9 GB = 602.1 KB
2025-03-17 09:37:28,282 INFO util.GSet: capacity      = 2^16 = 65536 entries
2025-03-17 09:37:28,306 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1482586675-127.0.1.1-1742204248300
2025-03-17 09:37:28,323 INFO common.Storage: Storage directory /home/hadoop/hdfs/namenode has been successfully formatted.
2025-03-17 09:37:28,355 INFO namenode.FSImageFormatProtobuf: Saving image file /home/hadoop/hdfs/namenode/current/fsimage.ckpt_0000000000000000000 using no
compression
2025-03-17 09:37:28,449 INFO namenode.FSImageFormatProtobuf: Image file /home/hadoop/hdfs/namenode/current/fsimage.ckpt_0000000000000000000 of size 401 byte
s saved in 0 seconds .
2025-03-17 09:37:28,460 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid ≥ 0
2025-03-17 09:37:28,477 INFO namenode.FSNamesystem: Stopping services started for active state
2025-03-17 09:37:28,478 INFO namenode.FSNamesystem: Stopping services started for standby state
2025-03-17 09:37:28,482 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2025-03-17 09:37:28,483 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at HuuNghia-PC/127.0.1.1
************************************************************/
hadoop@HuuNghia-PC:~$ |
```

Step 6: Start the Hadoop cluster

Start Hadoop services:

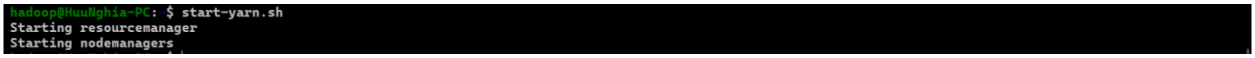
start-dfs.sh



```
hadoop@HuuNghia-PC:~$ start-dfs.sh
Starting namenodes on [lm.licenses.adobe.com]
Starting datanodes
Starting secondary namenodes [lm.licenses.adobe.com]
```

Start Node Manager and Resource Manager:

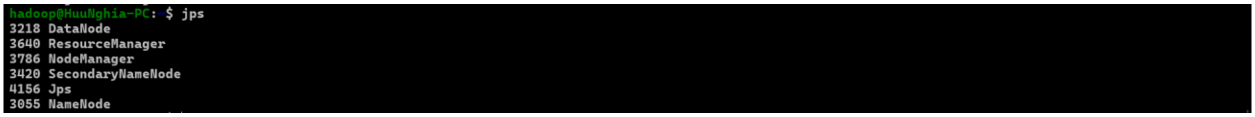
`start-yarn.sh`



```
hadoop@HuuNghia-PC:~$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
```

Verify running services:

`jps`



```
hadoop@HuuNghia-PC:~$ jps
3218 DataNode
3640 ResourceManager
3786 NodeManager
3420 SecondaryNameNode
4156 Jps
3055 NameNode
```

b/ Create a folder with path /hcmus on HDFS

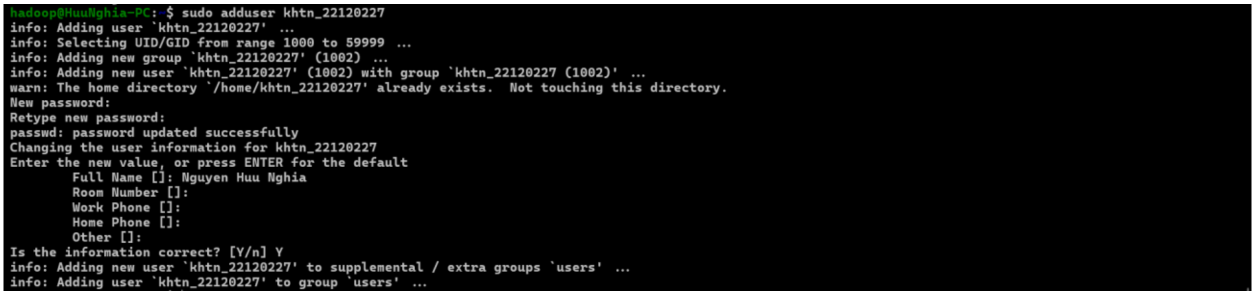
Using the following command:

`hdfs dfs -mkdir /hcmus`

c/ Create a user named khtn_<StudentID>

Using the following command:

`sudo adduser khtn_<StudentID>`

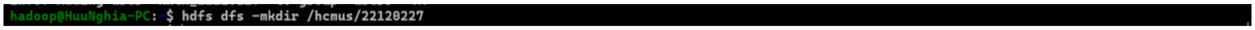


```
hadoop@HuuNghia-PC:~$ sudo adduser khtn_22120227
info: Adding user `khtn_22120227' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `khtn_22120227' (1002) ...
info: Adding new user `khtn_22120227' (1002) with group `khtn_22120227 (1002)' ...
warn: The home directory `/home/khtn_22120227' already exists.  Not touching this directory.
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for khtn_22120227
Enter the new value, or press ENTER for the default
        Full Name []: Nguyen Huu Nghia
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
info: Adding new user `khtn_22120227' to supplemental / extra groups `users' ...
info: Adding user `khtn_22120227' to group `users' ...
```

d/ Create a subfolder at /hcmus/<StudentID> and upload a file into it

Create a subfolder:

`hdfs dfs -mkdir /hcmus/<StudentID>`



```
hadoop@HuuNghia-PC:~$ hdfs dfs -mkdir /hcmus/22120227
```

Upload a file into it:

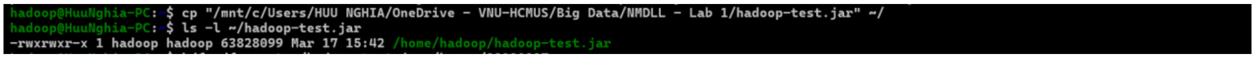
- Step 1: Copy the file to my home directory

  Using the **cp** command in Ubuntu to copy the file from the Windows mount to my home directory:

cp "/mnt/c/Users/HUU NGHIA/OneDrive - VNU-HCMUS/Big Data/NMDLL - Lab 1/hadoop-test.jar" ~/

Verify the file is copied, use the following command:

ls -l ~/hadoop-test.jar



```
hadoop@HuuNghia-PC:~$ cp "/mnt/c/Users/HUU NGHIA/OneDrive - VNU-HCMUS/Big Data/NMDLL - Lab 1/hadoop-test.jar" ~/
hadoop@HuuNghia-PC:~$ ls -l ~/hadoop-test.jar
-rwxrwxr-x 1 hadoop hadoop 63828099 Mar 17 15:42 /home/hadoop/hadoop-test.jar
```

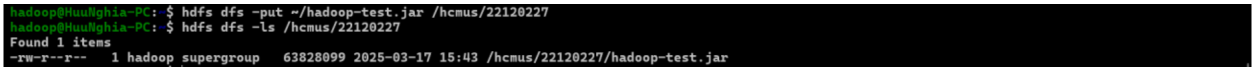
- Step 2: Upload the file from my home directory to HDFS

  Use the HDFS **put** command to upload the file:

  hdfs dfs -put ~/hadoop-test.jar /hcmus/<StudentID>

  Verify the upload by listing the directory in HDFS:

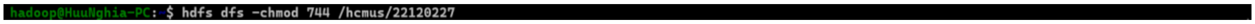
  hdfs dfs -ls /hcmus/<StudentID>



```
hadoop@HuuNghia-PC:~$ hdfs dfs -put ~/hadoop-test.jar /hcmus/22120227
hadoop@HuuNghia-PC:~$ hdfs dfs -ls /hcmus/22120227
Found 1 items
-rw-r--r--   1 hadoop supergroup   63828099 2025-03-17 15:43 /hcmus/22120227/hadoop-test.jar
```

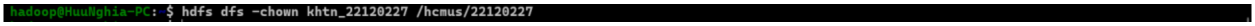
e/ Set permission and ownership

- hdfs dfs -chmod 744 /hcmus/<StudentID>**:** This sets permission for the folder. 744 means the owner has read, write, execute permissions, while others only have read access.



```
hadoop@HuuNghia-PC:~$ hdfs dfs -chmod 744 /hcmus/22120227
```

- hdfs dfs -chown khtn_<StudentID> /hcmus/<StudentID>**:** This changes owner of the folder to the user khtn_22120227, ensuring only that user can modify the folder.



```
hadoop@HuuNghia-PC:~$ hdfs dfs -chown khtn_22120227 /hcmus/22120227
```

f/ Run the attached JAR file named hadoop-test.jar

Note: The JAR file is already uploaded to HDFS, so I can run it directly from the local directory to ensure faster access and avoid redundant uploads.

- This command runs a pre-built Java JAR file, which processes data in my HDFS directory and generates a verification file to ensure everything is correctly configured.

- **<YOUR_HDFS_PORT>** : should be replaced with the port our Hadoop setup is running on (my port is 9000).

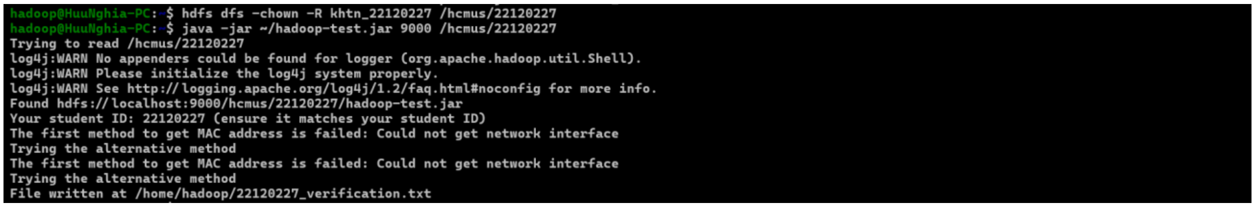
- **/hcmus/<StudentID>:** is the directory path I created in the previous steps – it tells the JAR file where to find my data.

Before running the JAR file, ensure the correct ownership is set for the directory. The JAR file checks both permissions and ownership. Without this step, it may fail with an ownership error:

hdfs dfs -chown -R khtn_<StudentID> /hcmus/<StudentID>

Then, run the JAR file:

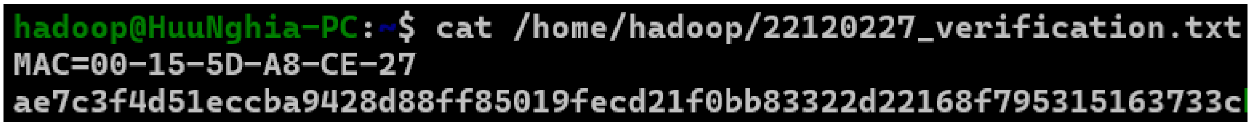
```
java -jar ~/hadoop-test.jar 9000 /hcmus/<StudentID>
```



```
hadoop@HuuNghia-PC:~$ hdfs dfs -chown -R khtn_22120227 /hcmus/22120227
hadoop@HuuNghia-PC:~$ java -jar ~/hadoop-test.jar 9000 /hcmus/22120227
Trying to read /hcmus/22120227
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/22120227/hadoop-test.jar
Your student ID: 22120227 (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/hadoop/22120227_verification.txt
```

Verify the result:

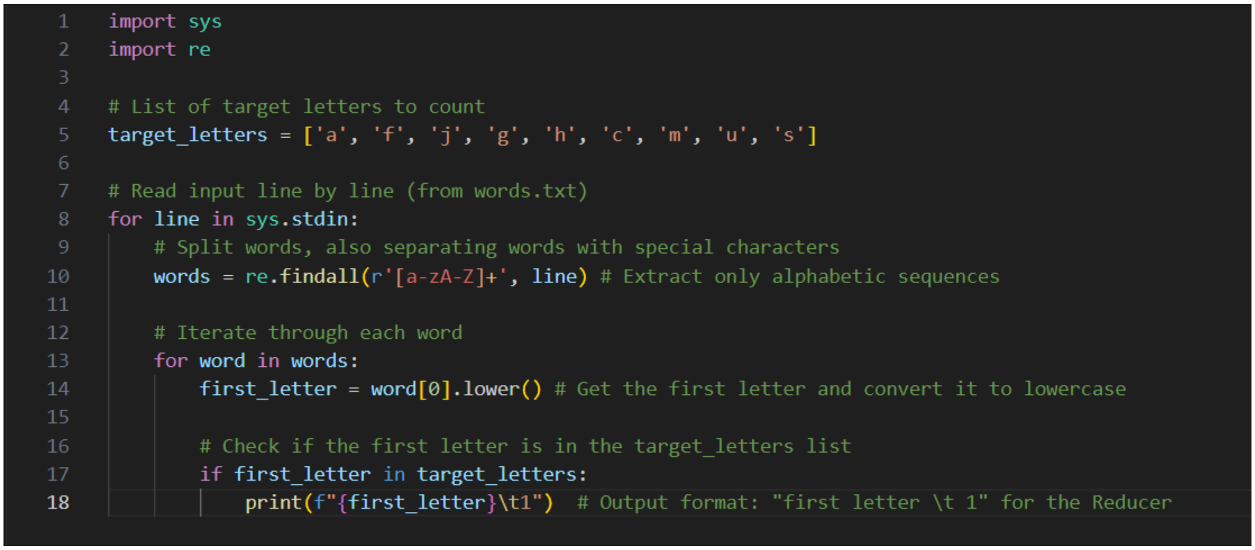
```
cat /home/hadoop/22120227_verification.txt
```



```
hadoop@HuuNghia-PC:~$ cat /home/hadoop/22120227_verification.txt
MAC=00-15-5D-A8-CE-27
ae7c3f4d51eccba9428d88ff85019fecd21f0bb83322d22168f795315163733c
```

2/ Word Count

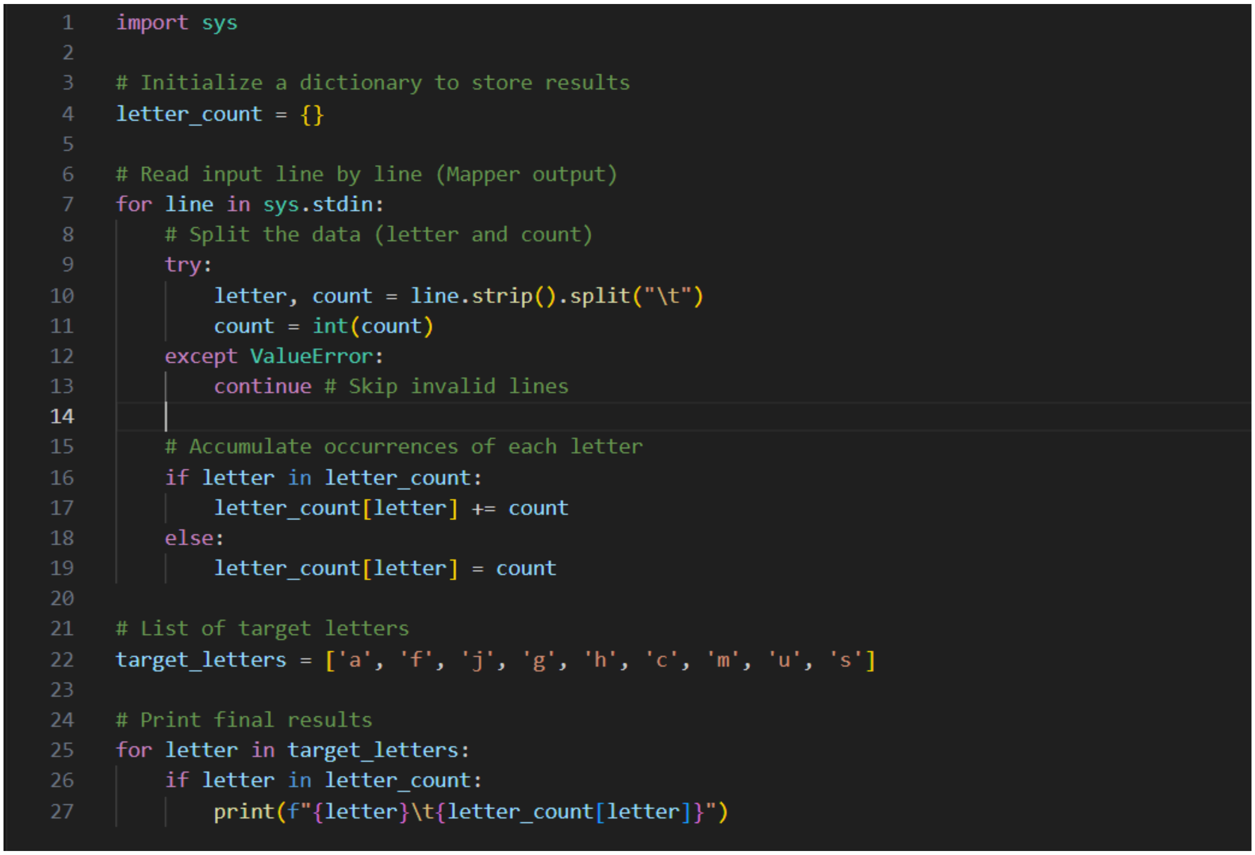
a/ Mapper[2]



```python
import sys
import re

# List of target letters to count
target_letters = ['a', 'f', 'j', 'g', 'h', 'c', 'm', 'u', 's']

# Read input line by line (from words.txt)
for line in sys.stdin:
    # Split words, also separating words with special characters
    words = re.findall(r'[a-zA-Z]+', line) # Extract only alphabetic sequences

    # Iterate through each word
    for word in words:
        first_letter = word[0].lower() # Get the first letter and convert it to lowercase

        # Check if the first letter is in the target_letters list
        if first_letter in target_letters:
            print(f"{first_letter}\t1")  # Output format: "first letter \t 1" for the Reducer
```

The **Mapper** reads input text line by line, extracts words while ignoring special characters, and checks if the first letter of each word matches a predefined list of target letters. If a match is found, the Mapper outputs a key-value pair, where the key is the first letter and the value is 1.

b/ Reducer[2]



```
import sys

# Initialize a dictionary to store results
letter_count = {}

# Read input line by line (Mapper output)
for line in sys.stdin:
    # Split the data (letter and count)
    try:
        letter, count = line.strip().split("\t")
        count = int(count)
    except ValueError:
        continue # Skip invalid lines

    # Accumulate occurrences of each letter
    if letter in letter_count:
        letter_count[letter] += count
    else:
        letter_count[letter] = count

# List of target letters
target_letters = ['a', 'f', 'j', 'g', 'h', 'c', 'm', 'u', 's']

# Print final results
for letter in target_letters:
    if letter in letter_count:
        print(f"{letter}\t{letter_count[letter]}")
```

The **Reducer** processes the output from the Mapper, aggregating the counts for each letter. It sums up all occurrences of each letter and prints the final count only for the predefined target letters.

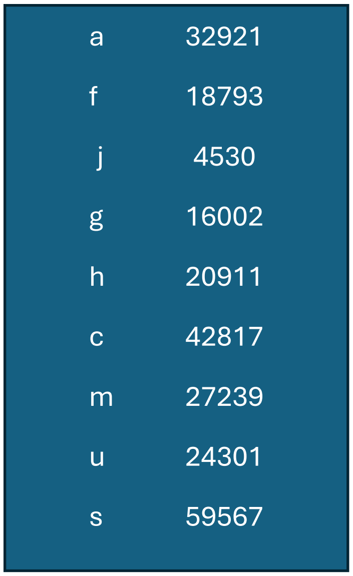
c/ Overall functional

This implementation follows the standard **MapReduce** approach:

- The **Mapper** filters and structures the data.
- The **Reducer** aggregates and summarizes the results.
- The output provides the total count of words starting with specific letters.

d/ Result

The result is:



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

References

[1]. Apache Hadoop 3.3.6 Installation on Ubuntu 22.04.2 LTS WSL for Windows | by Madiha Iqbal | Medium

[2]. Writing An Hadoop MapReduce Program In Python