

Lab 01: A Gentle Introduction to Hadoop

I. Team's result

ID	Name	Contribution
21120207	Nguyễn Thái Bình	25%
21120235	Trần Anh Duy	25%
21120240	Nguyễn Văn Hào	25%
21120257	Tôn Anh Huy	25%

II. Tasks

1. Setting up SNC - Single Node Cluster (4 points)

a. Nguyễn Thái Bình

Prerequisites

- Java

Install java

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

Checking Java installation

```
java -version
```

```
binhtn@DESKTOP-V21CHQU: ~
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/javap to provide /usr/bin/javap (javap) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jar to provide /usr/bin/jar (jar) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/xjc to provide /usr/bin/xjc (xjc) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/schemagen to provide /usr/bin/schemagen (schemagen) in
auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jps to provide /usr/bin/jps (jps) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/extcheck to provide /usr/bin/extcheck (extcheck) in aut
o mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jarsigner to provide /usr/bin/jarsigner (jarsigner) in
auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jmap to provide /usr/bin/jmap (jmap) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jstati to provide /usr/bin/jstati (jstati) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jdb to provide /usr/bin/jdb (jdb) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/serialver to provide /usr/bin/serialver (serialver) in
auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jfr to provide /usr/bin/jfr (jfr) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/wsgen to provide /usr/bin/wsgen (wsgen) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jcmd to provide /usr/bin/jcmd (jcmd) in auto mode
Setting up openjdk-8-jdk:amd64 (8u402+ga-2ubuntu1~20.04) ...
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/appletviewer to provide /usr/bin/appletviewer (appletvi
ewer) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jconsole to provide /usr/bin/jconsole (jconsole) in aut
o mode
Processing triggers for libgdk-pixbuf2.0-0:amd64 (2.40.0+dfsg-3ubuntu0.4) ...
Processing triggers for libc-bin (2.31-0ubuntu9.14) ...
[21120207] java -version
openjdk version "1.8.0_402"
OpenJDK Runtime Environment (build 1.8.0_402-8u402-ga-2ubuntu1~20.04-b06)
OpenJDK 64-Bit Server VM (build 25.402-b06, mixed mode)
[21120207]
```

- **SSH**

```
sudo apt-get install ssh
```

```
[21120207] sudo apt-get install ssh
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  ssh
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 5084 B of archives.
After this operation, 121 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 ssh all 1:8.2p1-4ubuntu0.11 [5084 B]
Fetched 5084 B in 1s (8299 B/s)
Selecting previously unselected package ssh.
(Reading database ... 47903 files and directories currently installed.)
Preparing to unpack .../ssh_1%3a8.2p1-4ubuntu0.11_all.deb ...
Unpacking ssh (1:8.2p1-4ubuntu0.11) ...
Setting up ssh (1:8.2p1-4ubuntu0.11) ...
[21120207]
```

Install Hadoop

Get the latest stable version from [Apache Download Mirrors](#)

Extract the downloaded file

```
tar xvzf hadoop-3.4.0.tar.gz
```

Configure Hadoop environment variables

Open .bashrc file

```
sudo nano ~/.bashrc
```

Add the following variables to the end of the file then save it

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=/home/binhtn/hadoop-3.4.0
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

```

binhtn@DESKTOP-V21CHQU: ~
GNU nano 6.2                               /home/binhtn/.bashrc *

f1

# 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

#Hadoop Related Options
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=/home/binhtn/hadoop-3.4.0
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"

File Name to Write: /home/binhtn/.bashrc
^G Help      M-D DOS Format      M-A Append      M-B Backup File
^C Cancel    M-M Mac Format      M-P Prepend    ^T Browse

```

Edit the file hadoop-env.sh

```
sudo nano hadoop-3.4.0/etc/hadoop/hadoop-env.sh
```

Find JAVA_HOME and change it as shown in the image below

```

binhtn@DESKTOP-V21CHQU: ~
GNU nano 6.2                               hadoop-3.4.0/etc/hadoop/hadoop-env.sh

# 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
#
# 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-8-openjdk-amd64
# The language environment in which Hadoop runs. Use the English
# environment to ensure that logs are printed as expected.
export LANG=en_US.UTF-8

[ Wrote 433 lines ]

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

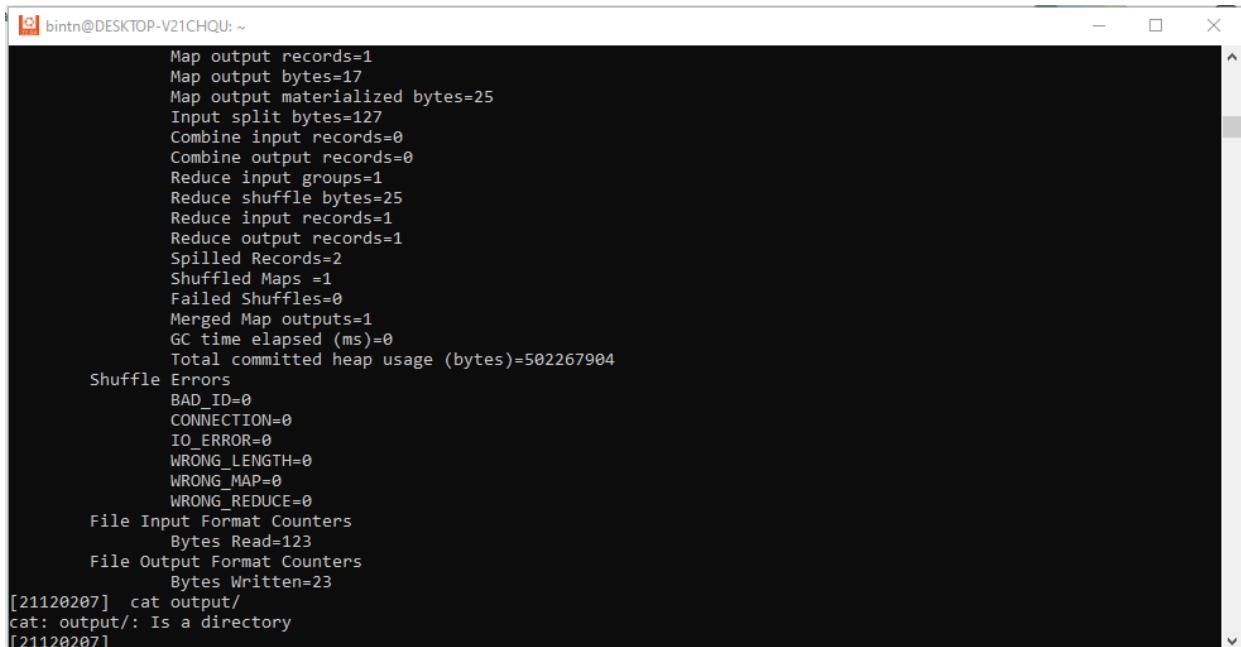
```

Standalone Operation

In `hadoop-3.4.0` folder, type these command

```
mkdir input
cp etc/hadoop/*.xml input
bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples.jar wordcount input output
```

The following example copies the unpacked conf directory to use as input and then finds and displays every match of the given regular expression. Output is written to the given output directory.



A screenshot of a terminal window titled "bintn@DESKTOP-V21CHQU: ~". The window contains the following text:

```
Map output records=1
Map output bytes=17
Map output materialized bytes=25
Input split bytes=127
Combine input records=0
Combine output records=0
Reduce input groups=1
Reduce shuffle bytes=25
Reduce input records=1
Reduce output records=1
Spilled Records=2
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=0
Total committed heap usage (bytes)=502267904
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=123
File Output Format Counters
  Bytes Written=23
[21120207] cat output/
cat: output/: Is a directory
[21120207]
```

Pseudo-Distributed Operation

Config `etc/hadoop/core-site.xml`

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

```

binhtn@DESKTOP-V21CHQU: ~                               etc/hadoop/core-site.xml
GNU nano 6.2
<?xml version="1.0" encoding="UTF-8"?>
<xmlelement 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>

[ Wrote 24 lines ]
^G Help      ^O Write Out   ^W Where Is   ^K Cut        ^T Execute   ^C Location   M-U Undo   M-A Set Mark
^X Exit      ^R Read File   ^\ Replace    ^U Paste     ^J Justify   ^I Go To Line M-F Redo   M-G Copy

```

Config [etc/hadoop/hdfs-site.xml](#)

```

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

```

```

binhtn@DESKTOP-V21CHQU: ~                               etc/hadoop/hdfs-site.xml *
GNU nano 6.2
<?xml version="1.0" encoding="UTF-8"?>
<xmlelement 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>
</configuration>

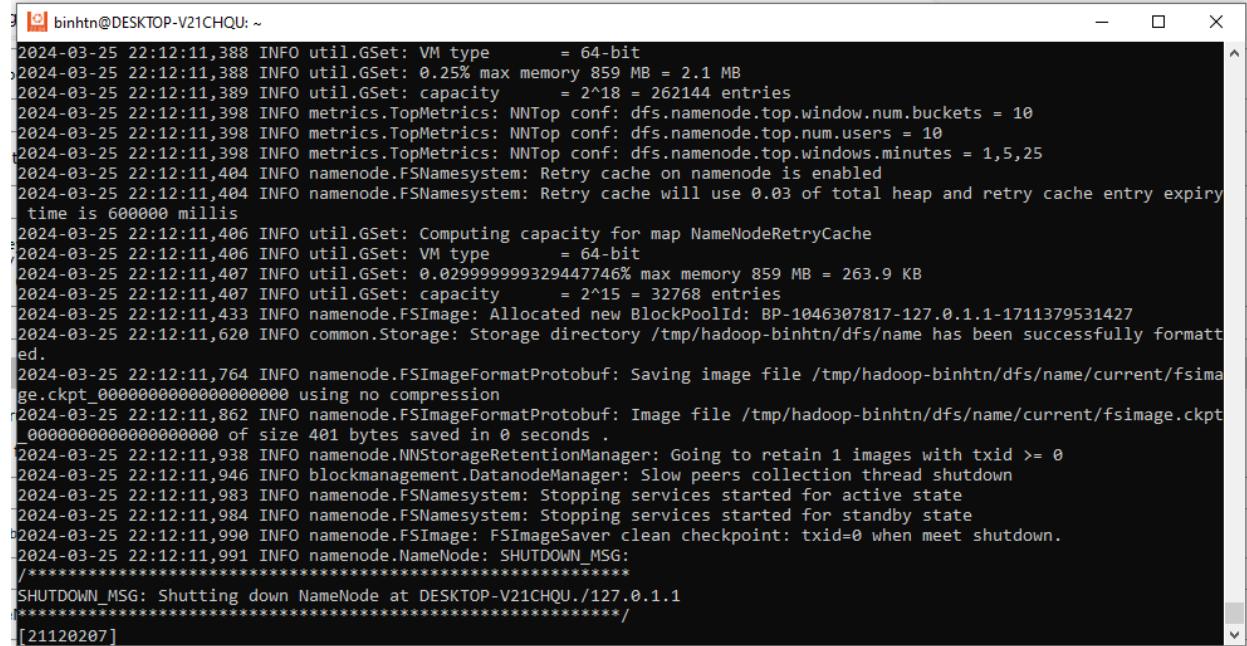
File Name to Write: etc/hadoop/hdfs-site.xml
^G Help      M-D DOS Format   M-A Append   M-B Backup File
^C Cancel    M-M Mac Format   M-P Prepend  ^T Browse

```

Execution

Format filesystem

```
bin/hdfs namenode -format
```



```
binhtn@DESKTOP-V21CHQU: ~
2024-03-25 22:12:11,388 INFO util.GSet: VM type      = 64-bit
2024-03-25 22:12:11,388 INFO util.GSet: 0.25% max memory 859 MB = 2.1 MB
2024-03-25 22:12:11,389 INFO util.GSet: capacity      = 2^18 = 262144 entries
2024-03-25 22:12:11,398 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
2024-03-25 22:12:11,398 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
2024-03-25 22:12:11,398 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
2024-03-25 22:12:11,404 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
2024-03-25 22:12:11,404 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 60000 millis
2024-03-25 22:12:11,406 INFO util.GSet: Computing capacity for map NameNodeRetryCache
2024-03-25 22:12:11,406 INFO util.GSet: VM type      = 64-bit
2024-03-25 22:12:11,407 INFO util.GSet: 0.029999999329447746% max memory 859 MB = 263.9 KB
2024-03-25 22:12:11,407 INFO util.GSet: capacity      = 2^15 = 32768 entries
2024-03-25 22:12:11,433 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1046307817-127.0.1.1-1711379531427
2024-03-25 22:12:11,620 INFO common.Storage: Storage directory /tmp/hadoop-binhtn/dfs/name has been successfully formatted.
2024-03-25 22:12:11,764 INFO namenode.FSImageFormatProtobuf: Saving image file /tmp/hadoop-binhtn/dfs/name/current/fsimage.ckpt_0000000000000000 using no compression
2024-03-25 22:12:11,862 INFO namenode.FSImageFormatProtobuf: Image file /tmp/hadoop-binhtn/dfs/name/current/fsimage.ckpt_0000000000000000 of size 401 bytes saved in 0 seconds .
2024-03-25 22:12:11,938 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2024-03-25 22:12:11,946 INFO blockmanagement.DatanodeManager: Slow peers collection thread shutdown
2024-03-25 22:12:11,983 INFO namenode.FSNamesystem: Stopping services started for active state
2024-03-25 22:12:11,984 INFO namenode.FSNamesystem: Stopping services started for standby state
2024-03-25 22:12:11,990 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2024-03-25 22:12:11,991 INFO namenode.NameNode: SHUTDOWN_MSG:
*****SHUTDOWN_MSG: Shutting down NameNode at DESKTOP-V21CHQU./127.0.1.1*****
[21120207]
```

Start NameNode daemon and DataNode daemon

```
sbin/start-dfs.sh
```

```

binhtn@DESKTOP-V21CHQU: ~
2024-03-25 22:12:11,398 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
2024-03-25 22:12:11,404 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
2024-03-25 22:12:11,404 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 millis
2024-03-25 22:12:11,406 INFO util.GSet: Computing capacity for map NameNodeRetryCache
2024-03-25 22:12:11,406 INFO util.GSet: VM type      = 64-bit
2024-03-25 22:12:11,407 INFO util.GSet: 0.02999999329447746% max memory 859 MB = 263.9 KB
2024-03-25 22:12:11,407 INFO util.GSet: capacity     = 2^15 = 32768 entries
2024-03-25 22:12:11,433 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1046307817-127.0.1.1-1711379531427
2024-03-25 22:12:11,620 INFO common.Storage: Storage directory /tmp/hadoop-binhtn/dfs/name has been successfully formatted.
2024-03-25 22:12:11,764 INFO namenode.FSImageFormatProtobuf: Saving image file /tmp/hadoop-binhtn/dfs/name/current/fsimage.ckpt_00000000000000000000 using no compression
2024-03-25 22:12:11,862 INFO namenode.FSImageFormatProtobuf: Image file /tmp/hadoop-binhtn/dfs/name/current/fsimage.ckpt_00000000000000000000 of size 401 bytes saved in 0 seconds .
2024-03-25 22:12:11,938 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2024-03-25 22:12:11,946 INFO blockmanagement.DatanodeManager: Slow peers collection thread shutdown
2024-03-25 22:12:11,983 INFO namenode.FSNamesystem: Stopping services started for active state
2024-03-25 22:12:11,984 INFO namenode.FSNamesystem: Stopping services started for standby state
2024-03-25 22:12:11,990 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2024-03-25 22:12:11,991 INFO namenode.NameNode: SHUTDOWN_MSG:
*****SHUTDOWN_MSG: Shutting down NameNode at DESKTOP-V21CHQU/127.0.1.1
*****[21120207] sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [DESKTOP-V21CHQU]
DESKTOP-V21CHQU: Warning: Permanently added 'desktop-v21chqu' (ED25519) to the list of known hosts.
[21120207]

```

Browse the web interface for the NameNode; by default it is available at localhost:9870

Started:	Mon Mar 25 22:12:59 +0700 2024
Version:	3.4.0, rbd8b77f398f626bb7791783192ee7a5dfaeecc760
Compiled:	Mon Mar 04 13:35:00 +0700 2024 by root from (HEAD detached at release-3.4.0-RC3)
Cluster ID:	CID-50ceef23-3b62-4408-8fcf-6b1d3f18675a
Block Pool ID:	BP-1046307817-127.0.1.1-1711379531427

Make the HDFS directories required to execute MapReduce jobs:

```
bin/hdfs dfs -mkdir -p /user/21120207
```

Config [etc/hadoop/mapred-site.xml](#)

```

<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
    <name>mapreduce.application.classpath</name>
    <value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HAI
  </property>
</configuration>

```

```

binhtn@DESKTOP-V21CHQU: ~                               etc/hadoop/mapred-site.xml *
GNU nano 6.2
<!--
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>
  <property>
    <name>mapreduce.application.classpath</name>
    <value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/lib/*</value>
  </property>
</configuration>

```

File menu: ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, M-U Undo, M-A Set Mark
 Edit menu: ^X Exit, ^R Read File, ^W Replace, ^U Paste, ^J Justify, ^Y Go To Line, M-E Redo, M-G Copy

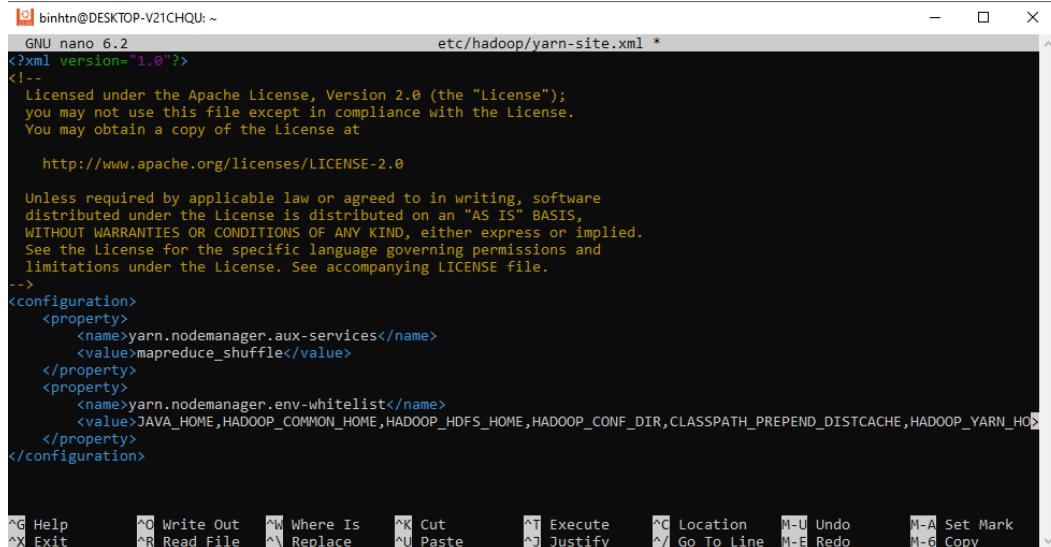
Config [etc/hadoop/yarn-site.xml](#)

```

<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.env-whitelist</name>
    <value>JAVA_HOME, HADOOP_COMMON_HOME, HADOOP_HDFS_HOME, HAI

```

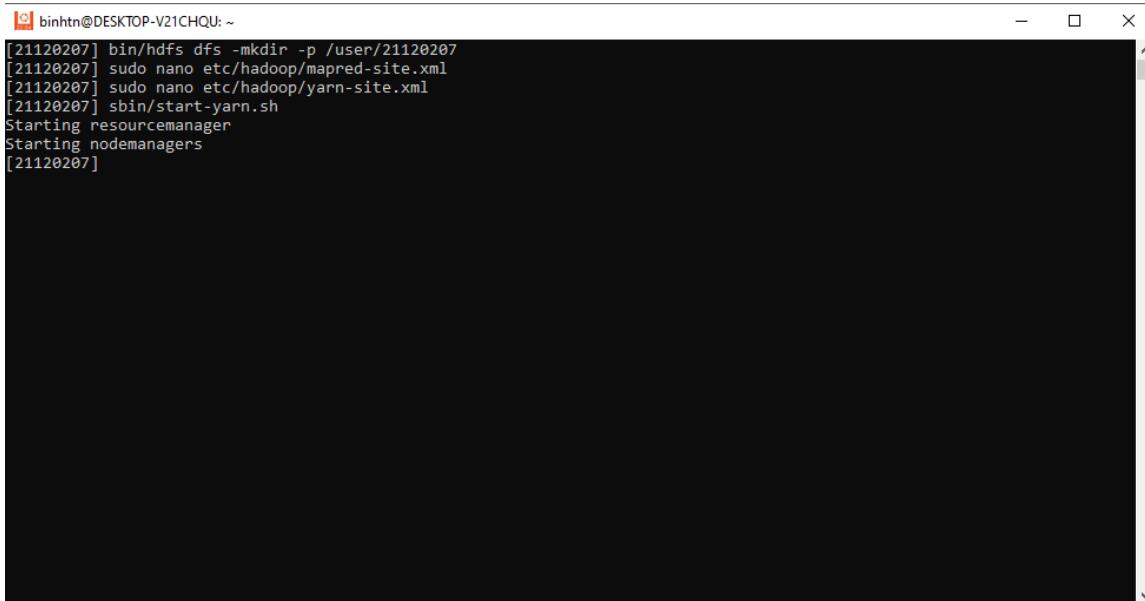
```
</property>  
</configuration>
```



```
binhtn@DESKTOP-V21CHQU: ~  
GNU nano 6.2          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>  
  <property>  
    <name>yarn.nodemanager.aux-services</name>  
    <value>mapreduce_shuffle</value>  
  </property>  
  <property>  
    <name>yarn.nodemanager.env-whitelist</name>  
    <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PREPEND_DISTCACHE,HADOOP_YARN_HOME_DIR</value>  
  </property>  
</configuration>  
  
^G Help      ^O Write Out   ^W Where Is   ^K Cut       ^T Execute   ^C Location   M-U Undo   M-A Set Mark  
^X Exit      ^R Read File   ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line M-B Redo   M-G Copy
```

Start ResourceManager daemon and NodeManager daemon

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



```
[21120207] bin/hdfs dfs -mkdir -p /user/21120207  
[21120207] sudo nano etc/hadoop/mapred-site.xml  
[21120207] sudo nano etc/hadoop/yarn-site.xml  
[21120207] sbin/start-yarn.sh  
Starting resourcemanager  
Starting nodemanagers  
[21120207]
```

Check web interface of ResourceManager at localhost:8088

The screenshot shows the Hadoop Web UI at localhost:8088/cluster. The top navigation bar includes back, forward, search, and refresh buttons. The title bar says "All Apps". The main content area has a sidebar with a tree view of cluster nodes, nodes, labels, applications, and a scheduler section. The main panel displays "Cluster Metrics" with counts for submitted, pending, running, completed apps, and running containers. It also shows "Cluster Nodes Metrics" with active, decommissioning, and decommissioned nodes. The "Scheduler Metrics" section details the Capacity Scheduler's configuration and current allocation. A table below lists application details like ID, User, Name, Application Type, Application Tags, Queue, Priority, Start Time, Launch Time, Finish Time, State, and Final Status. A message "No data" is shown at the bottom of the table.

Run `jps` to check

```
[binhnt@DESKTOP-V21CHQU ~]
[21120207] bin/hdfs dfs -mkdir -p /user/21120207
[21120207] bin/hdfs dfs -copyFromLocal /etc/hadoop/mapper-side.xml
[21120207] sudo nano etc/hadoop/yarn-site.xml
[21120207] sbin/start-yarn.sh
Starting resourcemanager
Starting datanodes
Starting resourcemangers
[21120207] jps
23522 DataNode
23412 ResourceManager
23412 NameNode
23717 SecondaryNameNode
24042 ResourceManager
24042 DataNode
[21120207]
```

b. Trần Anh Duy - 21120235

Prerequisites

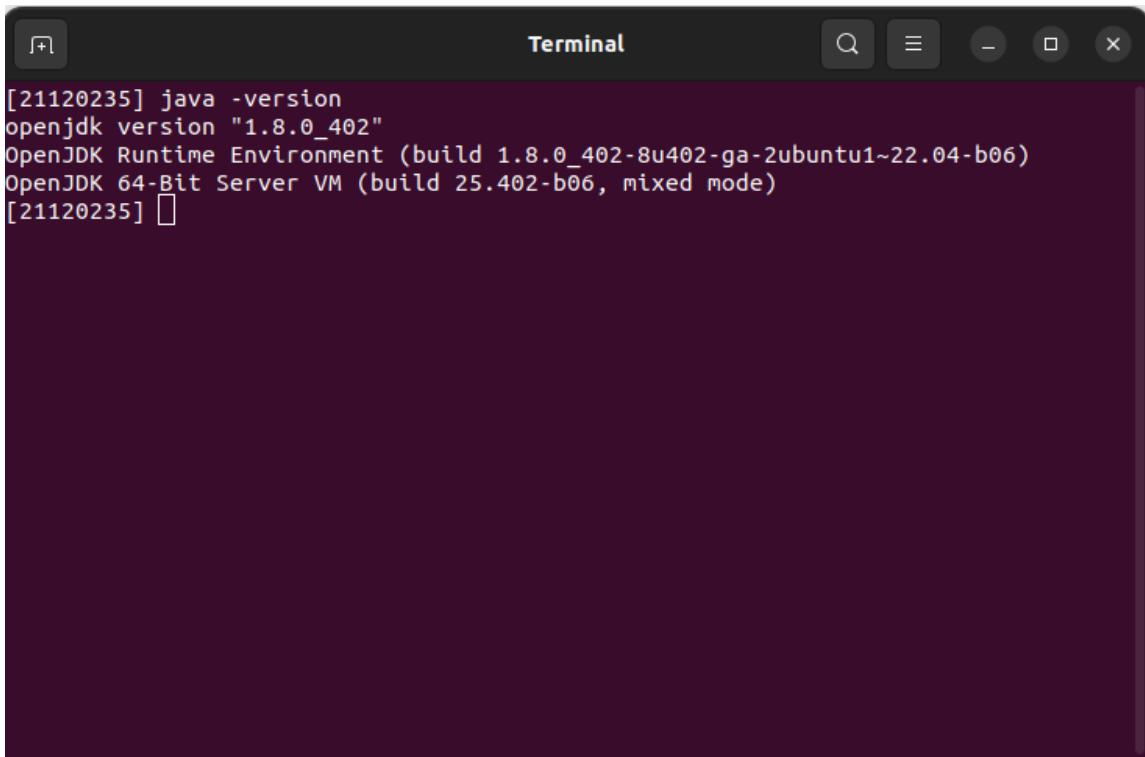
- Java

Install Java

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

Check if Java is already installed:

```
java -version
```

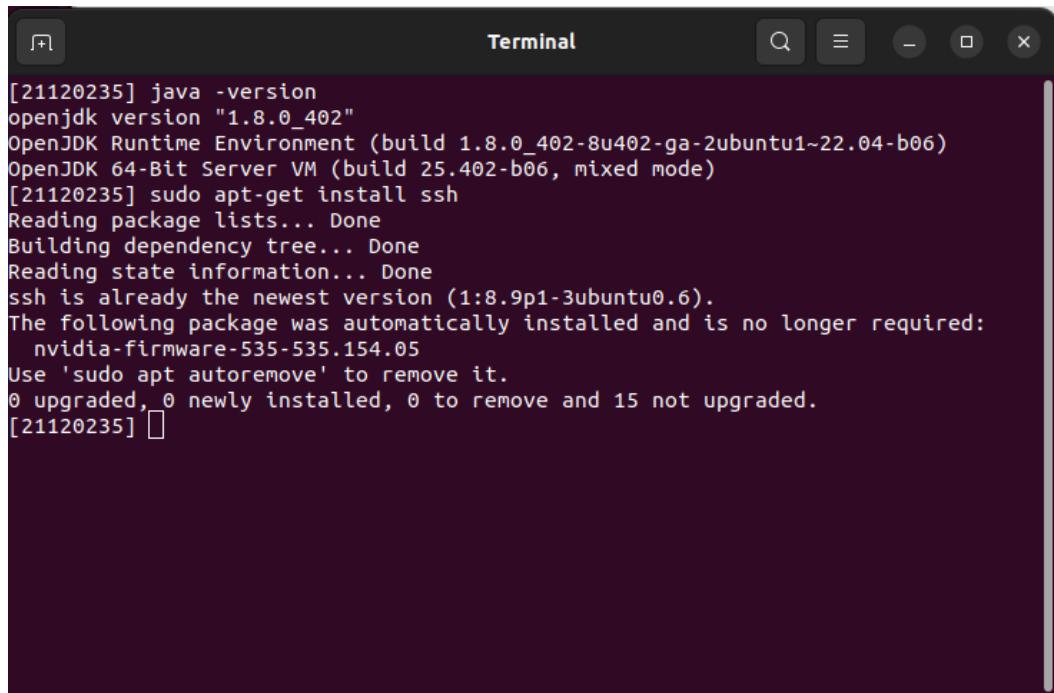


A screenshot of a dark-themed terminal window titled "Terminal". The window has standard Linux-style window controls at the top right. The terminal displays the output of the command "java -version". The output shows:

```
[21120235] java -version
openjdk version "1.8.0_402"
OpenJDK Runtime Environment (build 1.8.0_402-8u402-ga-2ubuntu1~22.04-b06)
OpenJDK 64-Bit Server VM (build 25.402-b06, mixed mode)
[21120235] 
```

- SSH

```
sudo apt-get install ssh
```



A screenshot of a dark-themed terminal window titled "Terminal". The window has standard Linux-style window controls at the top right. The terminal displays the output of the command "sudo apt-get install ssh". The output shows the package being installed and some dependency information:

```
[21120235] java -version
openjdk version "1.8.0_402"
OpenJDK Runtime Environment (build 1.8.0_402-8u402-ga-2ubuntu1~22.04-b06)
OpenJDK 64-Bit Server VM (build 25.402-b06, mixed mode)
[21120235] sudo apt-get install ssh
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ssh is already the newest version (1:8.9p1-3ubuntu0.6).
The following package was automatically installed and is no longer required:
  nvidia-firmware-535-535.154.05
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 15 not upgraded.
[21120235] 
```

Download Hadoop

Download the latest stable version Hadoop in this link:

<https://hadoop.apache.org/releases.html>

Extract file after download:

```
tar xvzf hadoop-3.4.0.tar.gz
```

After that, in the current directory will has a folder named `hadoop-3.4.0`

Set up environment variables for Hadoop (file .bashrc)

Open .bashrc file

```
sudo nano ~/.bashrc
```

Paste all lines to .bashrc then save it

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=/home/duy/hadoop-3.4.0
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

Terminal

```
GNU nano 6.2          /home/a21120235/.bashrc
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

PS1="[21120235] "

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=/home/a21120235/hadoop-3.4.0
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

Edit.hadoop-env `sudo nano hadoop-3.4.0/etc/hadoop/hadoop-env.sh`

Find `export JAVA_HOME` line to edit to `/usr/lib/jvm/java-8-openjdk-amd64`

Terminal

```
GNU nano 6.2          hadoop-3.4.0/etc/hadoop/hadoop-env.sh
#
# 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-8-openjdk-amd64

# The language environment in which Hadoop runs. Use the English
# environment to ensure that logs are printed as expected.
export LANG=en_US.UTF-8

# 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
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

Standalone Operation

In `hadoop-3.4.0` folder, type these command

```
mkdir input  
cp etc/hadoop/*.xml input  
bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples.jar  
cat output/
```

By default, Hadoop is configured to run in a non-distributed mode, as a single Java process. This is useful for debugging.

The following example copies the unpacked conf directory to use as input and then finds and displays every match of the given regular expression. Output is written to the given output directory.

```
Map-Reduce Framework  
  Map input records=1  
  Map output records=1  
  Map output bytes=17  
  Map output materialized bytes=25  
  Input split bytes=125  
  Combine input records=0  
  Combine output records=0  
  Reduce input groups=1  
  Reduce shuffle bytes=25  
  Reduce input records=1  
  Reduce output records=1  
  Spilled Records=2  
  Shuffled Maps =1  
  Failed Shuffles=0  
  Merged Map outputs=1  
  GC time elapsed (ms)=0  
  Total committed heap usage (bytes)=616562688  
Shuffle Errors  
  BAD_ID=0  
  CONNECTION=0  
  IO_ERROR=0  
  WRONG_LENGTH=0  
  WRONG_MAP=0  
  WRONG_REDUCE=0  
File Input Format Counters  
  Bytes Read=123  
File Output Format Counters  
  Bytes Written=23  
[21120235] cat output/■
```

Check the output content:

```
[21120235]cat output/part-r-00000  
1      dfsadmin
```

Pseudo-Distributed Operation

Hadoop can also be run on a single-node in a pseudo-distributed mode where each Hadoop daemon runs in a separate Java process.

Config `etc/hadoop/core-site.xml`

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

Config `etc/hadoop/hdfs-site.xml`

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

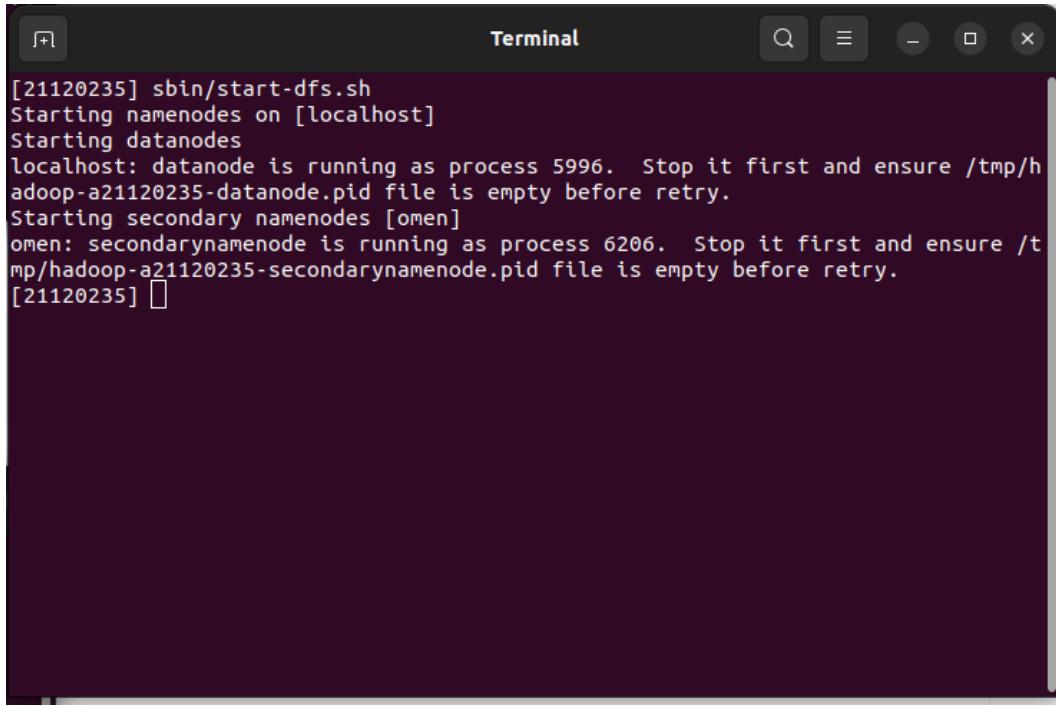
Format filesystem

```
bin/hdfs namenode -format
```

```
ation= null ? (Y or N) y
2024-03-27 19:14:04,531 INFO namenode.FSImage: Allocated new BlockPoolId: BP-
1277433387-127.0.1.1-1711541644523
2024-03-27 19:14:04,532 INFO common.Storage: Will remove files: [/tmp/hadoop-
duy/dfs/name/current/fsimage_00000000000000000000.md5, /tmp/hadoop-duy/dfs/nam-
e/current/VERSION, /tmp/hadoop-duy/dfs/name/current/seen_txid, /tmp/hadoop-du-
y/dfs/name/current/fsimage_0000000000000000]
2024-03-27 19:14:04,549 INFO common.Storage: Storage directory /tmp/hadoop-du-
y/dfs/name has been successfully formatted.
2024-03-27 19:14:04,581 INFO namenode.FSImageFormatProtobuf: Saving image fil-
e /tmp/hadoop-duy/dfs/name/current/fsimage.ckpt_0000000000000000 using no
compression
2024-03-27 19:14:04,702 INFO namenode.FSImageFormatProtobuf: Image file /tmp/
hadoop-duy/dfs/name/current/fsimage.ckpt_0000000000000000 of size 395 byte
s saved in 0 seconds .
2024-03-27 19:14:04,717 INFO namenode.NNStorageRetentionManager: Going to ret-
ain 1 images with txid >= 0
2024-03-27 19:14:04,724 INFO blockmanagement.DatanodeManager: Slow peers coll-
ection thread shutdown
2024-03-27 19:14:04,745 INFO namenode.FSNamesystem: Stopping services started
for active state
2024-03-27 19:14:04,746 INFO namenode.FSNamesystem: Stopping services started
for standby state
2024-03-27 19:14:04,749 INFO namenode.FSImage: FSImageSaver clean checkpoint:
txid=0 when meet shutdown.
2024-03-27 19:14:04,750 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at Omen/127.0.1.1
*****/
[21120235]
```

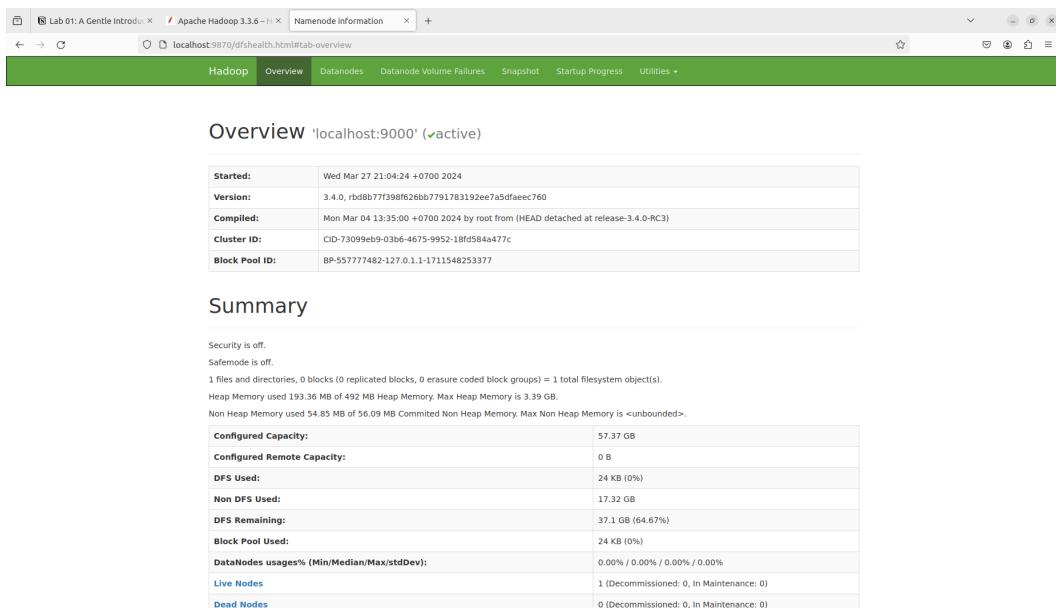
Start NameNode daemon and DataNode daemon

```
sbin/start-dfs.sh
```



```
[21120235] sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
localhost: datanode is running as process 5996. Stop it first and ensure /tmp/hadoop-a21120235-datanode.pid file is empty before retry.
Starting secondary namenodes [omen]
omen: secondarynamenode is running as process 6206. Stop it first and ensure /tmp/hadoop-a21120235-secondarynamenode.pid file is empty before retry.
[21120235]
```

Browse the web interface for the NameNode at <http://localhost:9870>



Overview 'localhost:9000' (active)

Started:	Wed Mar 27 21:04:24 +0700 2024
Version:	3.4.0, rdb8b7f398f626bb779178319zee7a5dfaeecc760
Compiled:	Mon Mar 04 13:35:00 +0700 2024 by root from (HEAD detached at release-3.4.0-RC3)
Cluster ID:	CID-73099eb9-03b6-4675-9952-18fd584a477c
Block Pool ID:	BP-557777482-127.0.1.1-1711548253377

Summary

Security is off.
Safemode is off.

1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).

Heap Memory used 193.36 MB of 492 MB Heap Memory. Max Heap Memory is 3.39 GB.

Non Heap Memory used 54.85 MB of 56.09 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

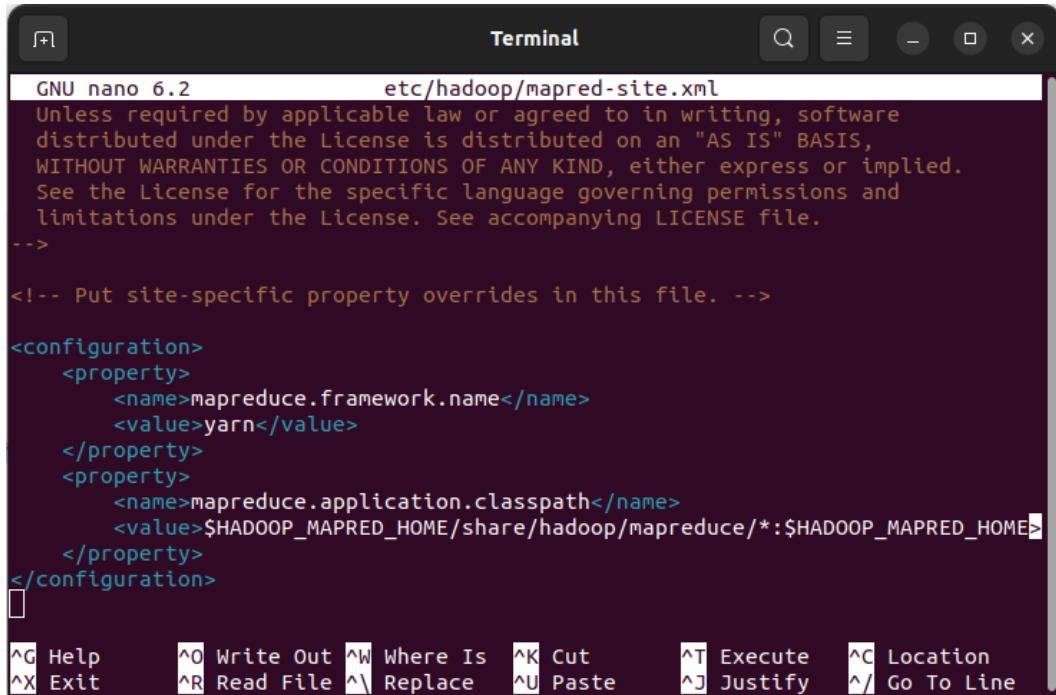
Configured Capacity:	57.37 GB
Configured Remote Capacity:	0 B
DFS Used:	24 KB (0%)
Non DFS Used:	17.32 GB
DFS Remaining:	37.1 GB (64.67%)
Block Pool Used:	24 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	1 (Decommissioned: 0, In Maintenance: 0)
Dead Nodes	0 (Decommissioned: 0, In Maintenance: 0)

Create directory required to execute MapReduce jobs:

```
bin/hdfs dfs -mkdir -p /user/21120235
```

Config [etc/hadoop/mapred-site.xml](#)

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
    <name>mapreduce.application.classpath</name>
    <value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/lib/*</value>
  </property>
</configuration>
```



The screenshot shows a terminal window titled "Terminal" with the file "etc/hadoop/mapred-site.xml" open in the nano editor. The terminal interface includes standard window controls (minimize, maximize, close) and a menu bar with "File", "Edit", "View", "Search", "Help", and "Terminal". The nano editor has its own set of key bindings at the bottom.

```
GNU nano 6.2          etc/hadoop/mapred-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>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
    <name>mapreduce.application.classpath</name>
    <value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/lib/*</value>
  </property>
</configuration>
```

Config [etc/hadoop/yarn-site.xml](#)

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
```

```
</property>
<property>
    <name>yarn.nodemanager.env-whitelist</name>
    <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HAI
</property>
</configuration>
```

The screenshot shows a terminal window titled "Terminal" with the command "GNU nano 6.2" at the top. The file path "etc/hadoop/yarn-site.xml" is displayed in the title bar, along with the URL "http://www.apache.org/licenses/LICENSE-2.0". The main content of the file is visible, including the XML configuration for the YARN NodeManager. The terminal window has a dark background with light-colored text. At the bottom, there is a menu bar with icons for Help, Exit, Write Out, Read File, Where Is, Replace, Cut, Paste, Execute, Justify, Location, and Go To Line, each associated with a specific keyboard shortcut.

```
GNU nano 6.2          etc/hadoop/yarn-site.xml
http://www.apache.org/licenses/LICENSE-2.0

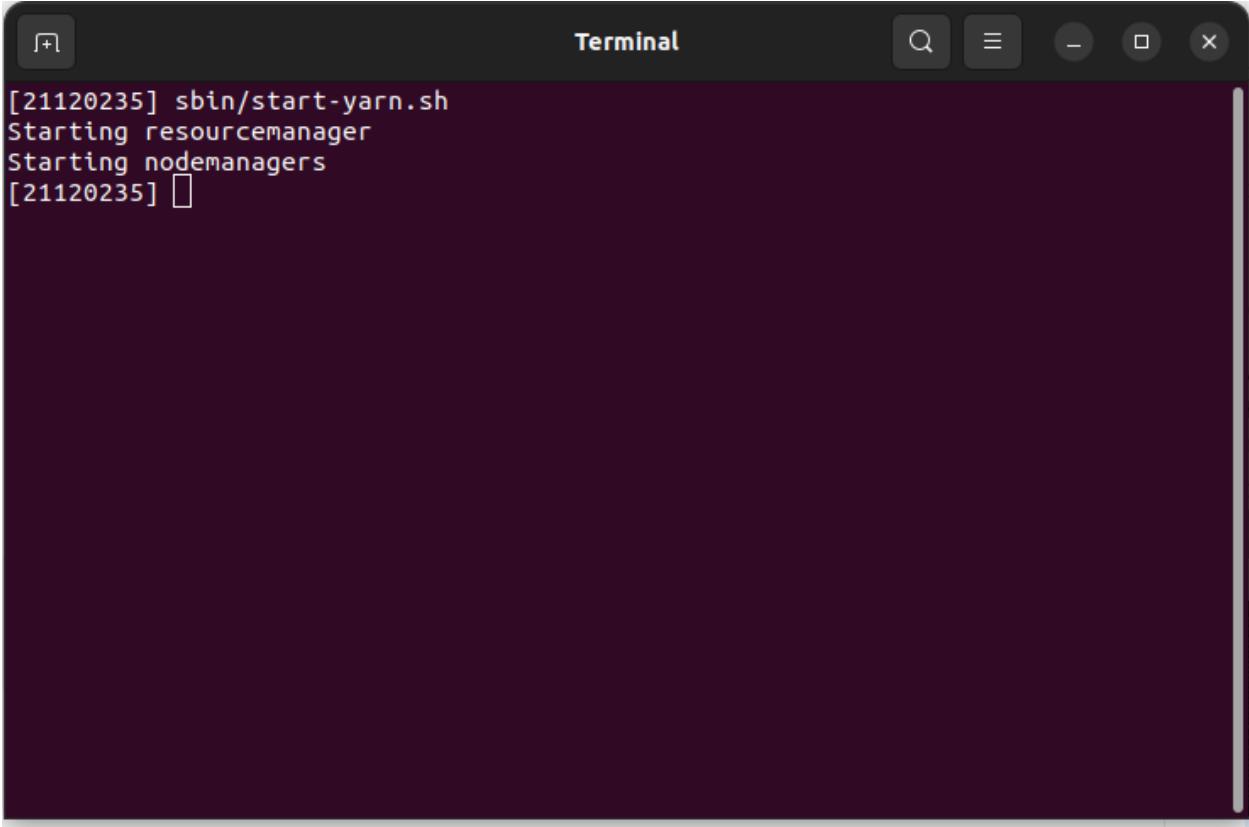
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->

<configuration>
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>
    <property>
        <name>yarn.nodemanager.env-whitelist</name>
        <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CL
    </property>
</configuration>
□

^G Help      ^O Write Out ^W Where Is  ^K Cut      ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

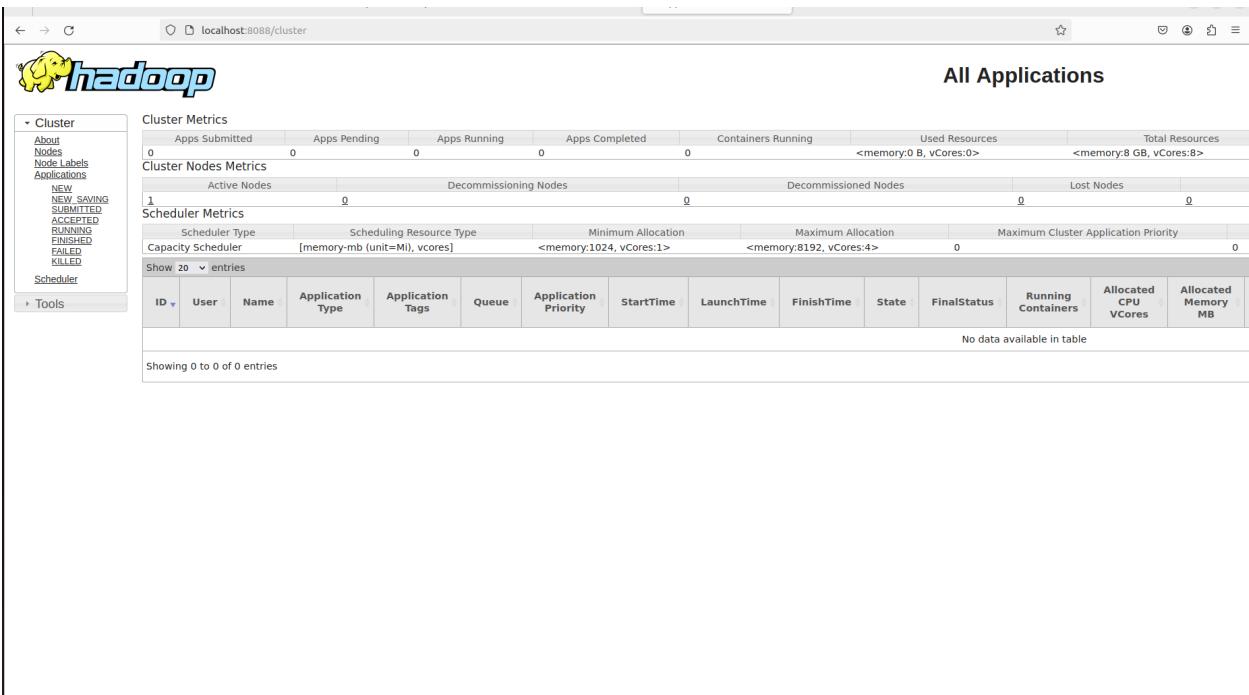
Start ResourceManager daemon and NodeManager daemon

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



```
[21120235] sbin/start-yarn.sh
Starting resourcemanager
Starting nodemanagers
[21120235] [
```

Check web interface of ResourceManager at localhost:8088

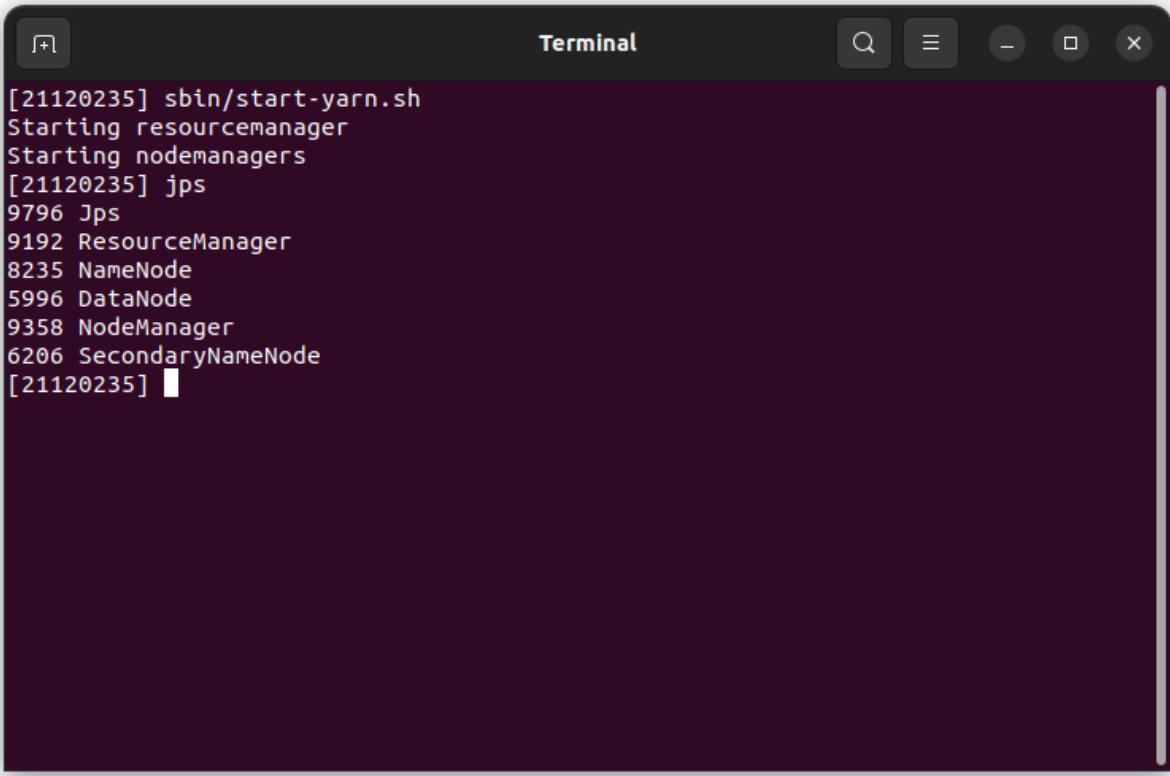


The screenshot shows the Hadoop ResourceManager's web interface at `localhost:8088/cluster`. The left sidebar has a tree view with "Cluster" expanded, showing "About", "Nodes", "Node Labels", and "Applications". Under "Applications", there is a dropdown menu with options: NEW, SAVING, SUBMITTED, ACCEPTED, RUNNING, FINISHED, FAILED, KILLED, and Scheduler. The "Scheduler" option is selected. Other items in the sidebar include "Tools". The main content area is titled "All Applications". It displays several metrics tables:

- Cluster Metrics**: Shows 0 Apps Submitted, 0 Apps Pending, 0 Apps Running, 0 Apps Completed, 0 Containers Running, <memory:0 B, vCores:0>, and <memory:8 GB, vCores:8>.
- Cluster Nodes Metrics**: Shows 1 Active Nodes, 0 Decommissioning Nodes, 0 Decommissioned Nodes, 0 Lost Nodes, and 0.
- Scheduler Metrics**: Shows Scheduler Type as Capacity Scheduler, Scheduling Resource Type as [memory-mb (unit=Mi), vcores], Minimum Allocation as <memory:1024, vCores:1>, Maximum Allocation as <memory:8192, vCores:4>, and Maximum Cluster Application Priority as 0.

Below these tables is a table header for "All Applications" with columns: ID, User, Name, Application Type, Application Tags, Queue, Application Priority, StartTime, LaunchTime, FinishTime, State, FinalStatus, Running Containers, Allocated CPU Vcores, and Allocated Memory MB. A note below the table says "No data available in table". At the bottom, it says "Showing 0 to 0 of 0 entries".

Run `jps` to check



The screenshot shows a terminal window titled "Terminal". The window contains the following text:

```
[21120235] sbin/start-yarn.sh
Starting resourcemanager
Starting nodemanagers
[21120235] jps
9796 Jps
9192 ResourceManager
8235 NameNode
5996 DataNode
9358 NodeManager
6206 SecondaryNameNode
[21120235]
```

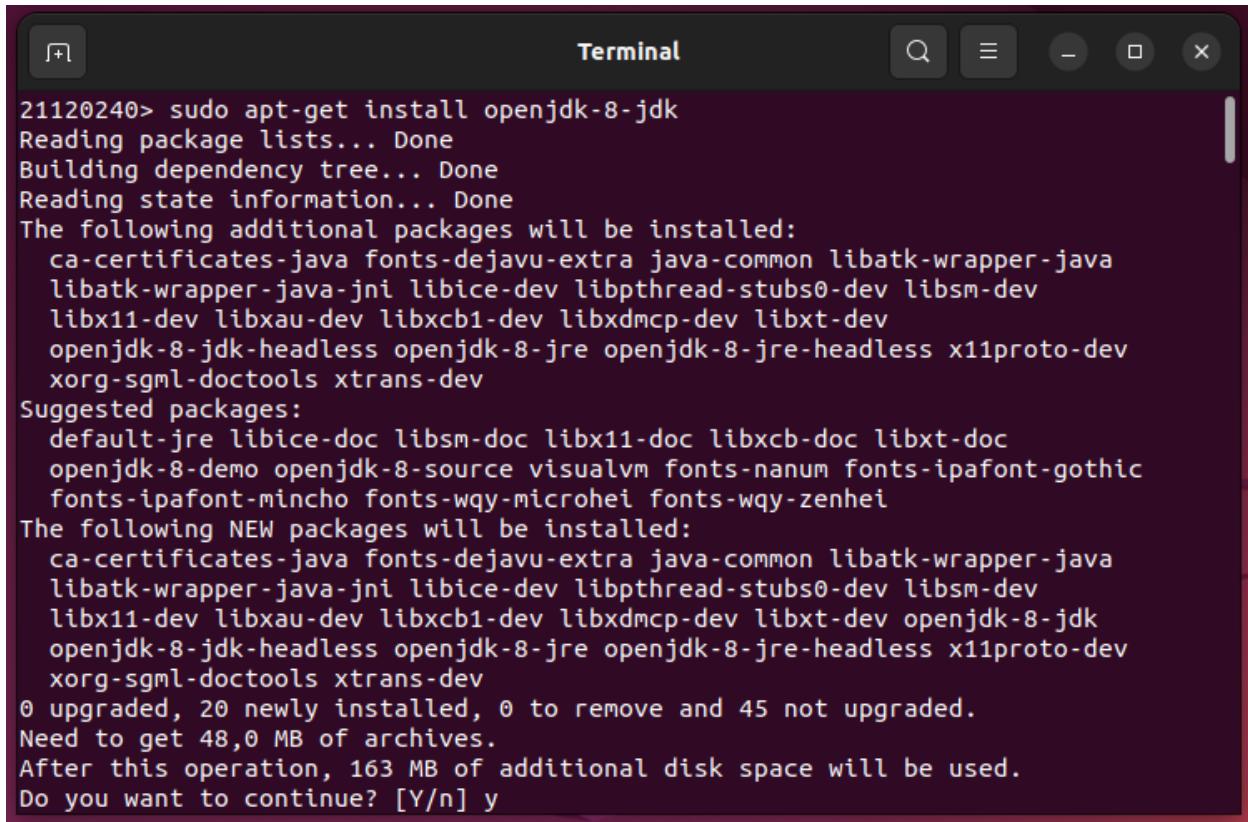
c. Nguyễn Văn Hào - 21120240

Prerequisites

- Java

Install Java

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

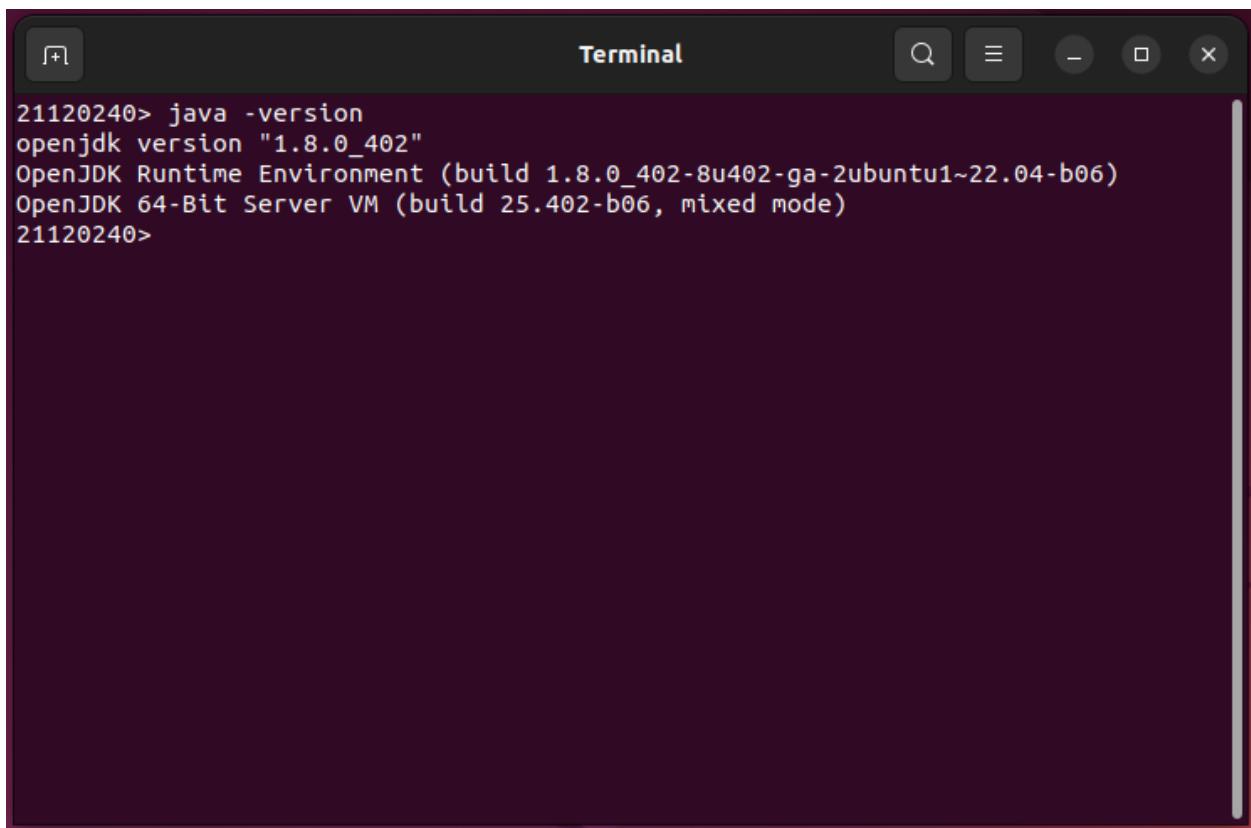


A screenshot of a terminal window titled "Terminal". The window has a dark theme with light-colored text. It displays the command "sudo apt-get install openjdk-8-jdk" followed by its execution output. The output shows the package list, dependency tree, state information, additional packages to be installed (including Java fonts and development libraries), suggested packages, new packages to be installed, and finally the summary of 0 upgraded, 20 newly installed packages, and 45 not upgraded. It also indicates a need for 48,0 MB of archives and 163 MB of additional disk space. The user is prompted with "Do you want to continue? [Y/n] y".

```
21120240> sudo apt-get install openjdk-8-jdk
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java
  libatk-wrapper-java-jni libice-dev libpthread-stubs0-dev libsm-dev
  libx11-dev libxau-dev libxcb1-dev libxdmcp-dev libxt-dev
  openjdk-8-jdk-headless openjdk-8-jre openjdk-8-jre-headless x11proto-dev
  xorg-sgml-doctools xtrans-dev
Suggested packages:
  default-jre libice-doc libsm-doc libxcb-doc libxt-doc
  openjdk-8-demo openjdk-8-source visualvm fonts-nanum fonts-ipafont-gothic
  fonts-ipafont-mincho fonts-wqy-microhei fonts-wqy-zenhei
The following NEW packages will be installed:
  ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java
  libatk-wrapper-java-jni libice-dev libpthread-stubs0-dev libsm-dev
  libx11-dev libxau-dev libxcb1-dev libxdmcp-dev libxt-dev openjdk-8-jdk
  openjdk-8-jdk-headless openjdk-8-jre openjdk-8-jre-headless x11proto-dev
  xorg-sgml-doctools xtrans-dev
0 upgraded, 20 newly installed, 0 to remove and 45 not upgraded.
Need to get 48,0 MB of archives.
After this operation, 163 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Check if Java was installed successfully

```
java -version
```

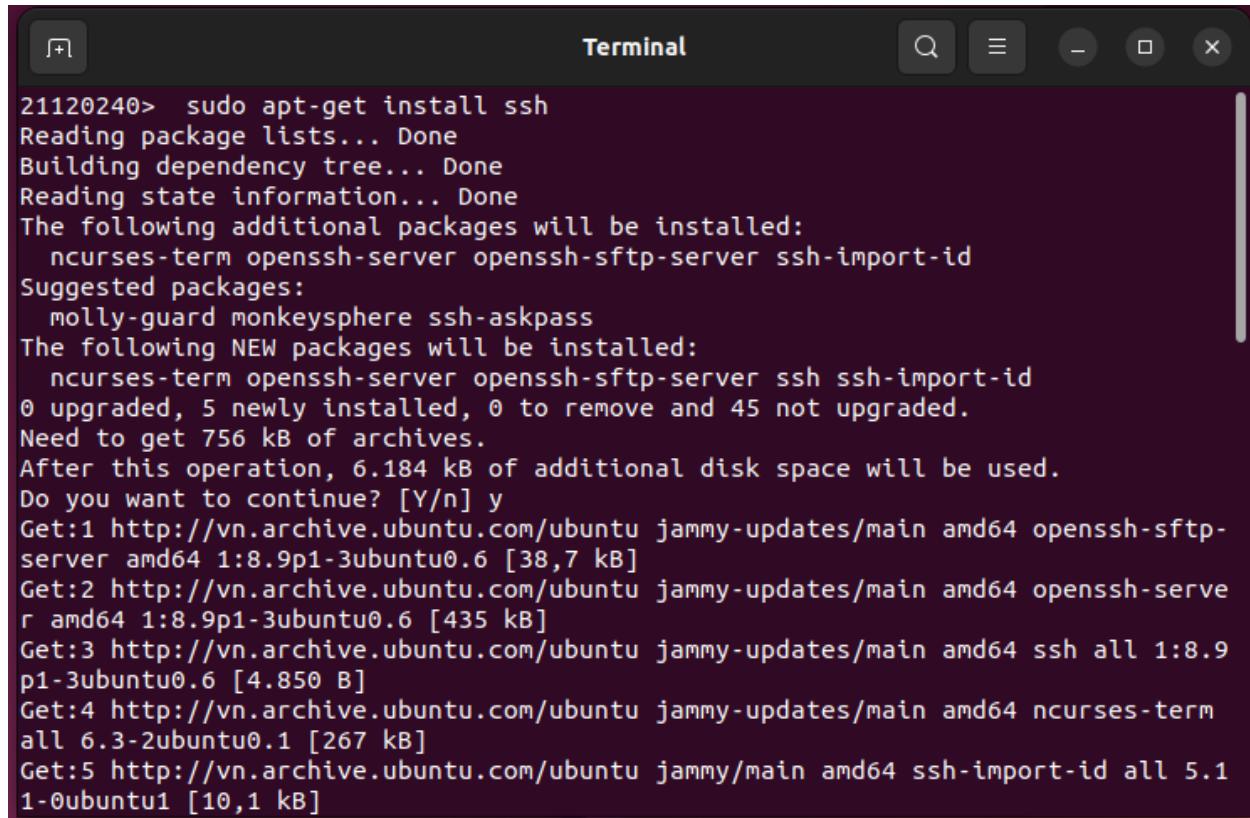


A screenshot of a terminal window titled "Terminal". The window has a dark theme with light-colored text. It displays the output of the command "java -version". The output shows:

```
21120240> java -version
openjdk version "1.8.0_402"
OpenJDK Runtime Environment (build 1.8.0_402-8u402-ga-2ubuntu1~22.04-b06)
OpenJDK 64-Bit Server VM (build 25.402-b06, mixed mode)
21120240>
```

- **SSH**

```
sudo apt-get install ssh
```



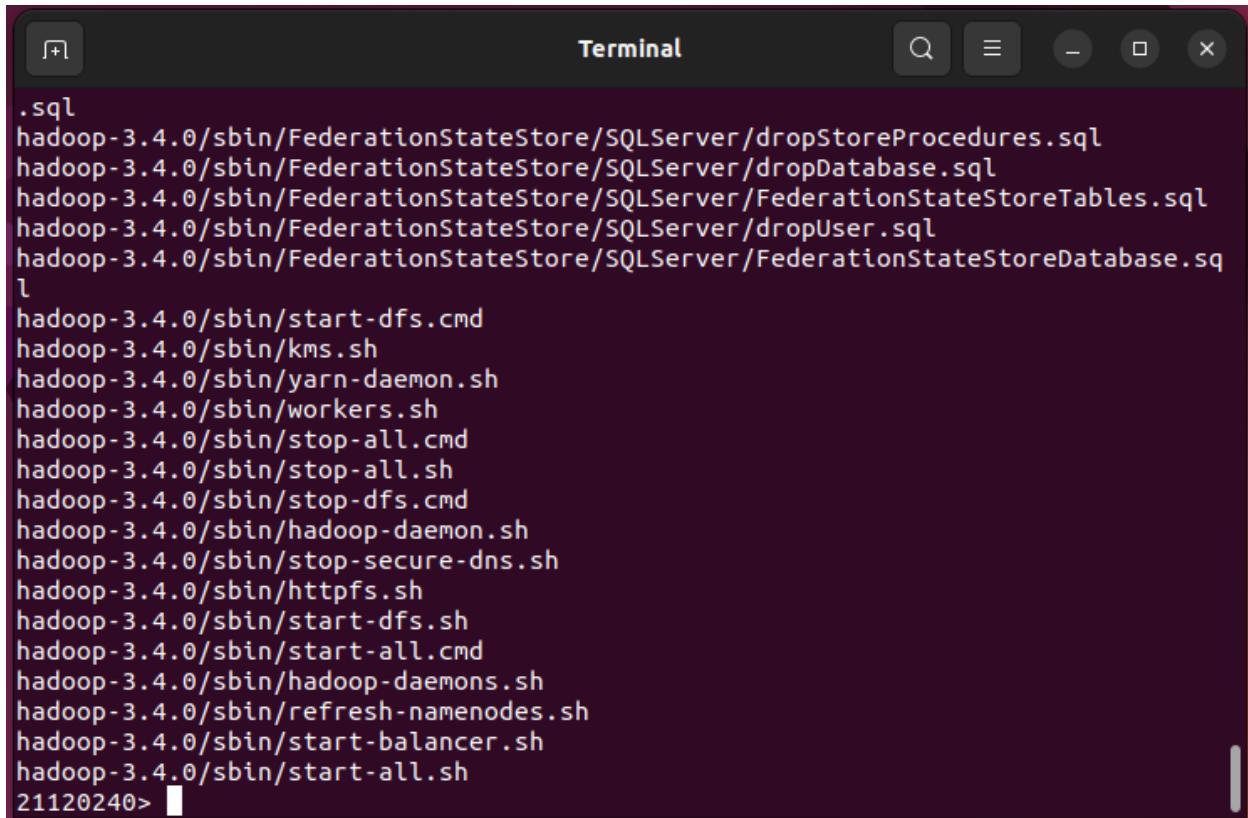
```
21120240> sudo apt-get install ssh
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere ssh-askpass
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh ssh-import-id
0 upgraded, 5 newly installed, 0 to remove and 45 not upgraded.
Need to get 756 kB of archives.
After this operation, 6.184 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://vn.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-sftp-server amd64 1:8.9p1-3ubuntu0.6 [38,7 kB]
Get:2 http://vn.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-server amd64 1:8.9p1-3ubuntu0.6 [435 kB]
Get:3 http://vn.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ssh all 1:8.9p1-3ubuntu0.6 [4.850 B]
Get:4 http://vn.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ncurses-term all 6.3-2ubuntu0.1 [267 kB]
Get:5 http://vn.archive.ubuntu.com/ubuntu jammy/main amd64 ssh-import-id all 5.1-0ubuntu1 [10,1 kB]
```

Download Hadoop

Download Hadoop in <https://dlcdn.apache.org/hadoop/common/>

Extract the file using this command:

```
tar xvzf hadoop-3.4.0.tar.gz
```



A screenshot of a terminal window titled "Terminal". The window shows a list of files and scripts located in the directory `/usr/lib/hadoop-3.4.0/sbin`. The files listed include various SQL scripts (e.g., `dropStoreProcedures.sql`, `dropDatabase.sql`, `FederationStateStoreTables.sql`, `dropUser.sql`, `FederationStateStoreDatabase.sql`) and shell scripts (e.g., `start-dfs.cmd`, `kms.sh`, `yarn-daemon.sh`, `workers.sh`, `stop-all.cmd`, `stop-all.sh`, `stop-dfs.cmd`, `hadoop-daemon.sh`, `stop-secure-dns.sh`, `httpfs.sh`, `start-dfs.sh`, `start-all.cmd`, `hadoop-daemons.sh`, `refresh-namenodes.sh`, `start-balancer.sh`, `start-all.sh`). The prompt at the bottom of the terminal is `21120240>`.

Set up environment variables for Hadoop (file .bashrc)

Open file `.bashrc` and paste these rows:

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=/home/hao/hadoop-3.4.0
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

```
GNU nano 6.2                               /home/hao/.bashrc *

# 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
PS1="21120240> "
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=/home/hao/hadoop-3.4.0
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

Key bindings for nano 6.2:

- ^G Help
- ^O Write Out
- ^W Where Is
- ^K Cut
- ^T Execute
- ^C Location
- ^X Exit
- ^R Read File
- ^\\ Replace
- ^U Paste
- ^J Justify
- ^/ Go To Line

Set up JAVA_HOME

Edit `export JAVA_HOME` in `hadoop-3.4.0/etc/hadoop/hadoop-env.sh`

The screenshot shows a terminal window with the title 'hadoop-env.sh' and the path '~/.hadoop-3.4.0/etc/hadoop'. The window contains the source code of the 'hadoop-env.sh' shell script. The code is mostly in blue font, with the 'export' command at line 54 highlighted in red. The script handles environment variables like JAVA_HOME, HADOOP_HOME, and LANG.

```
28 ## {yarn-env.sh|hdfs-env.sh} > hadoop-env.sh > hard-coded defaults
29 ##
30 ## {YARN_xyz|HDFS_xyz} > HADOOP_xyz > hard-coded defaults
31 ##
32
33 # Many of the options here are built from the perspective that users
34 # may want to provide OVERWRITING values on the command line.
35 # For example:
36 #
37 #   JAVA_HOME=/usr/java/testing hdfs dfs -ls
38 #
39 # Therefore, the vast majority (BUT NOT ALL!) of these defaults
40 # are configured for substitution and not append. If append
41 # is preferable, modify this file accordingly.
42
43 ###
44 # Generic settings for HADOOP
45 ###
46
47 # Technically, the only required environment variable is JAVA_HOME.
48 # All others are optional. However, the defaults are probably not
49 # preferred. Many sites configure these options outside of Hadoop,
50 # such as in /etc/profile.d
51
52 # The java implementation to use. By default, this environment
53 # variable is REQUIRED on ALL platforms except OS X!
54 export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64|
55
56 # The language environment in which Hadoop runs. Use the English
57 # environment to ensure that logs are printed as expected.
58 export LANG=en_US.UTF-8
59
60 # Location of Hadoop. By default, Hadoop will attempt to determine
61 # this location based upon its execution path.
62 # export HADOOP_HOME=
63
64 # Location of Hadoop's configuration information. i.e., where this
```

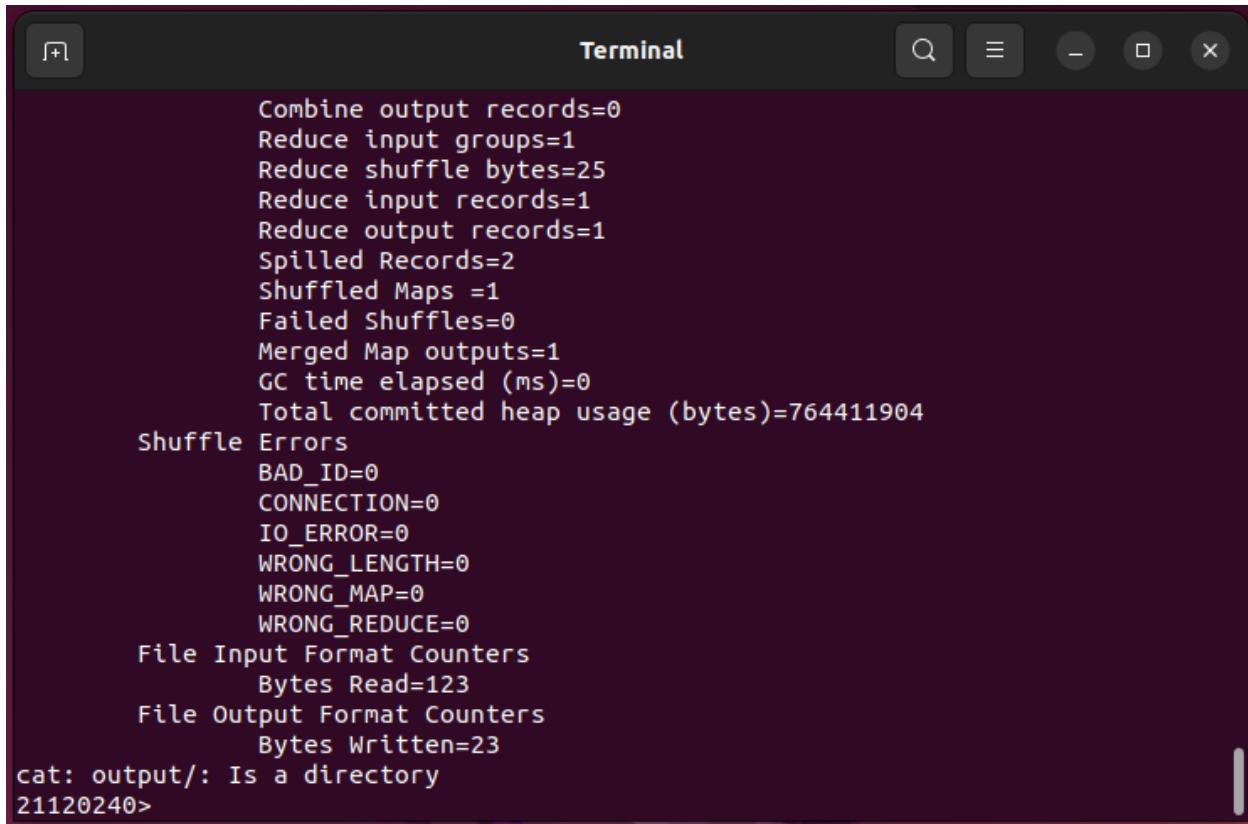
sh ▾ Tab Width: 8 ▾ Ln 54, Col 51 ▾ INS

Standalone Operation

Open terminal in `hadoop-3.4.0` and run this command:

```
mkdir input
cp etc/hadoop/*.xml input
bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples.jar grep "dfs[a-z.]+"
cat output/
```

It copies the directory to input folder and then finds and displays every match of the regex 'dfs[a-z.]+' then write to output folder.



A screenshot of a terminal window titled "Terminal". The window shows the output of a Hadoop job. The output includes various counters and error counts. At the bottom, it shows that "output/" is a directory.

```
Combine output records=0
Reduce input groups=1
Reduce shuffle bytes=25
Reduce input records=1
Reduce output records=1
Spilled Records=2
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=0
Total committed heap usage (bytes)=764411904
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=123
File Output Format Counters
Bytes Written=23
cat: output/: Is a directory
21120240>
```

Pseudo-Distributed Operation

Config [etc/hadoop/core-site.xml](#)

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

The screenshot shows a code editor window with the following details:

- Title Bar:** core-site.xml
- Path:** ~/hadoop-3.4.0/etc/hadoop
- Buttons:** Open, Save, Minimize, Maximize, Close.
- Code Content:**

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3 <!--
4 Licensed under the Apache License, Version 2.0 (the "License");
5 you may not use this file except in compliance with the License.
6 You may obtain a copy of the License at
7
8   http://www.apache.org/licenses/LICENSE-2.0
9
10 Unless required by applicable law or agreed to in writing, software
11 distributed under the License is distributed on an "AS IS" BASIS,
12 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 See the License for the specific language governing permissions and
14 limitations under the License. See accompanying LICENSE file.
15 -->
16
17 <!-- Put site-specific property overrides in this file. -->
18
19 <configuration>
20   <property>
21     <name>fs.defaultFS</name>
22     <value>hdfs://localhost:9000</value>
23   </property>
24 </configuration>
```
- Status Bar:** XML ▾ Tab Width: 8 ▾ Ln 24, Col 17 ▾ INS

Config [etc/hadoop/hdfs-site.xml](#)

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

The screenshot shows a code editor window with the following details:

- Title Bar:** hdfs-site.xml
- Path:** ~/hadoop-3.4.0/etc/hadoop
- Buttons:** Open, Save, Minimize, Maximize, Close.

The content of the file is as follows:

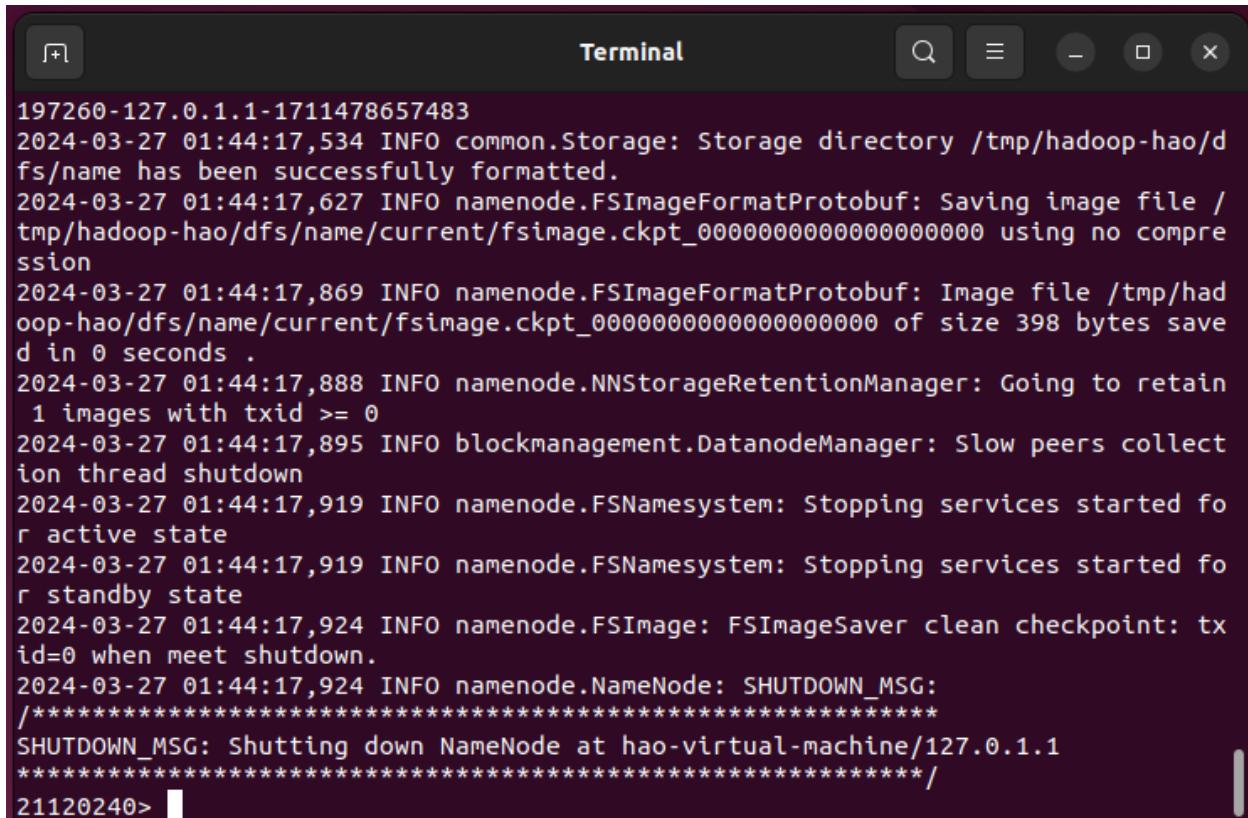
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3 <!--
4 Licensed under the Apache License, Version 2.0 (the "License");
5 you may not use this file except in compliance with the License.
6 You may obtain a copy of the License at
7
8   http://www.apache.org/licenses/LICENSE-2.0
9
10 Unless required by applicable law or agreed to in writing, software
11 distributed under the License is distributed on an "AS IS" BASIS,
12 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 See the License for the specific language governing permissions and
14 limitations under the License. See accompanying LICENSE file.
15 -->
16
17 <!-- Put site-specific property overrides in this file. -->
18
19 <configuration>
20   <property>
21     <name>dfs.replication</name>
22     <value>1</value>
23   </property>
24 </configuration>
```

At the bottom of the editor, the status bar displays:

- Saving file "/home/hao/hadoop-3.4.0/etc/hadoop/hdfs-site.xml"...
- XML
- Tab Width: 8
- Ln 24, Col 17
- INS

Format filesystem

```
bin/hdfs namenode -format
```

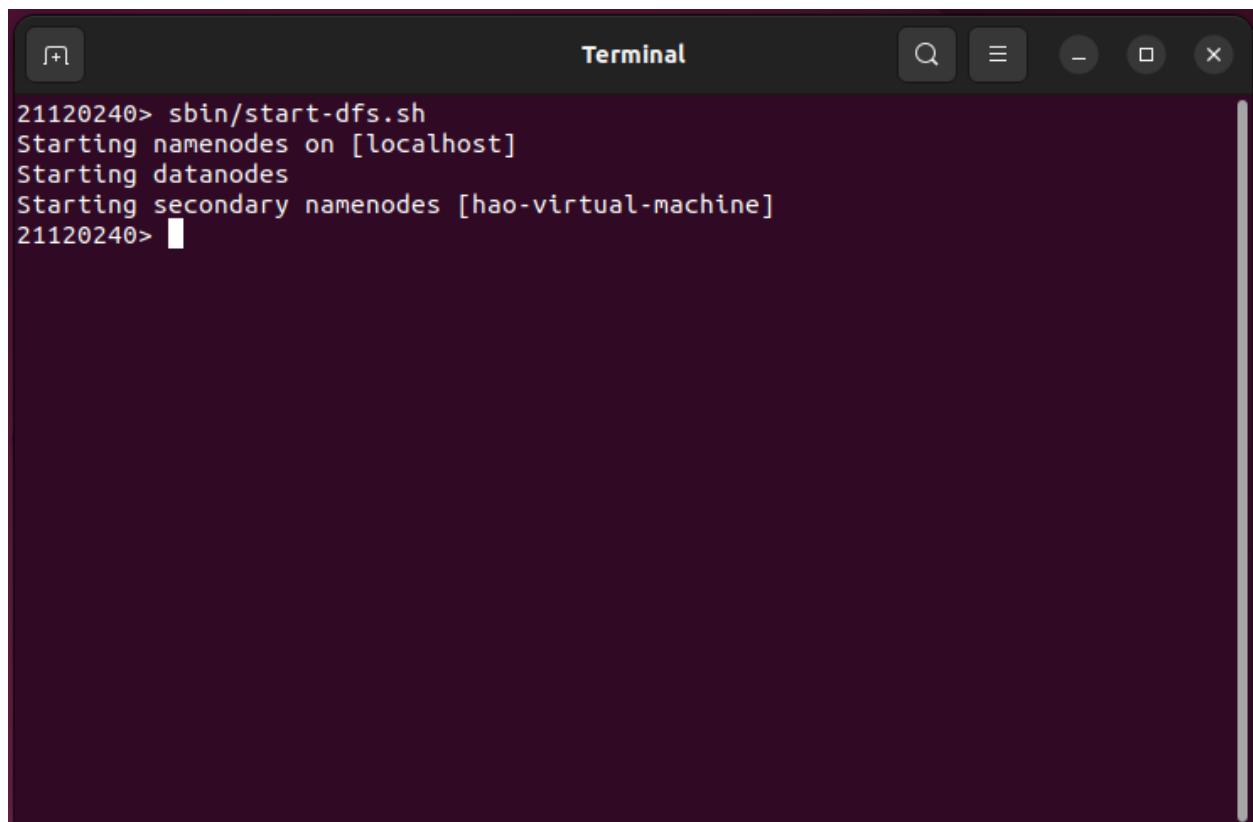


A screenshot of a terminal window titled "Terminal". The window shows a series of log messages from a Hadoop NameNode shutdown. The messages include:

```
197260-127.0.1.1-1711478657483
2024-03-27 01:44:17,534 INFO common.Storage: Storage directory /tmp/hadoop-hao/dfs/name has been successfully formatted.
2024-03-27 01:44:17,627 INFO namenode.FSImageFormatProtobuf: Saving image file /tmp/hadoop-hao/dfs/name/current/fsimage.ckpt_00000000000000000000 using no compression
2024-03-27 01:44:17,869 INFO namenode.FSImageFormatProtobuf: Image file /tmp/hadoop-hao/dfs/name/current/fsimage.ckpt_00000000000000000000 of size 398 bytes saved in 0 seconds .
2024-03-27 01:44:17,888 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2024-03-27 01:44:17,895 INFO blockmanagement.DatanodeManager: Slow peers collection thread shutdown
2024-03-27 01:44:17,919 INFO namenode.FSNamesystem: Stopping services started for active state
2024-03-27 01:44:17,919 INFO namenode.FSNamesystem: Stopping services started for standby state
2024-03-27 01:44:17,924 INFO namenode.FSImage: FSImageSaver clean checkpoint: tx id=0 when meet shutdown.
2024-03-27 01:44:17,924 INFO namenode.NameNode: SHUTDOWN_MSG:
*****SHUTDOWN_MSG: Shutting down NameNode at hao-virtual-machine/127.0.1.1*****
21120240>
```

Start NameNode daemon and DataNode daemon

```
sbin/start-dfs.sh
```



A screenshot of a terminal window titled "Terminal". The window has a dark theme with light-colored text. The terminal is displaying the output of a command-line session. The session starts with the command "sbin/start-dfs.sh" followed by several lines of text indicating the startup of various HDFS components: "Starting namenodes on [localhost]", "Starting datanodes", and "Starting secondary namenodes [hao-virtual-machine]". The session ends with the prompt "21120240>".

```
21120240> sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [hao-virtual-machine]
21120240>
```

Check by accessing localhost:9870

The screenshot shows a web browser window titled "Namenode information". The address bar displays "localhost:9870/dfshealth.html#tab-overview". The navigation bar includes links for "Hadoop", "Overview", "Datanodes", "Datanode Volume Failures", "Snapshot", "Startup Progress", and "Utilities". The main content area is titled "Overview 'localhost:9000' (✓active)". It contains a table with the following data:

Started:	Wed Mar 27 22:30:57 +0700 2024
Version:	3.4.0, rbd8b77f398f626bb7791783192ee7a5dfaeecc760
Compiled:	Mon Mar 04 13:35:00 +0700 2024 by root from (HEAD detached at release-3.4.0-RC3)
Cluster ID:	CID-3de75c82-9fe7-416a-a509-c4341ef3c6a8
Block Pool ID:	BP-350480455-127.0.1.1-1711553432974

Summary

Security is off.
Safemode is off.
1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).
Heap Memory used 139.37 MB of 233 MB Heap Memory. Max Heap Memory is 860.5 MB.
Non Heap Memory used 50.49 MB of 52.55 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	19.02 GB
-----------------------------	----------

Create directory required to execute MapReduce jobs:

```
bin/hdfs dfs -mkdir -p /user/21120240
```

Config [etc/hadoop/mapred-site.xml](#)

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

The screenshot shows a code editor window with the title "mapred-site.xml" and the path "~/hadoop-3.4.0/etc/hadoop". The file content is an XML configuration for Hadoop MapReduce. It includes a license notice and several property definitions. The file ends with a closing tag. The status bar at the bottom indicates "XML" mode, a tab width of 8, line 28, column 17, and an "INS" status.

```
1 <?xml version="1.0"?>
2 <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3 <!--
4   Licensed under the Apache License, Version 2.0 (the "License");
5   you may not use this file except in compliance with the License.
6   You may obtain a copy of the License at
7
8     http://www.apache.org/licenses/LICENSE-2.0
9
10 Unless required by applicable law or agreed to in writing, software
11 distributed under the License is distributed on an "AS IS" BASIS,
12 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 See the License for the specific language governing permissions and
14 limitations under the License. See accompanying LICENSE file.
15 -->
16
17 <!-- Put site-specific property overrides in this file. -->
18
19 <configuration>
20   <property>
21     <name>mapreduce.framework.name</name>
22     <value>yarn</value>
23   </property>
24   <property>
25     <name>mapreduce.application.classpath</name>
26     <value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/hadoop/
mapreduce/lib/*</value>
27   </property>
28 </configuration>
```

Config [etc/hadoop/yarn-site.xml](#)

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.env-whitelist</name>
    <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HAI
```

The screenshot shows a code editor window with the title "yarn-site.xml" and the path "~/hadoop-3.4.0/etc/hadoop". The file contains XML configuration for YARN. Lines 1 through 25 are shown, including the Apache License header and various property definitions for nodemanager aux-services and env-whitelist.

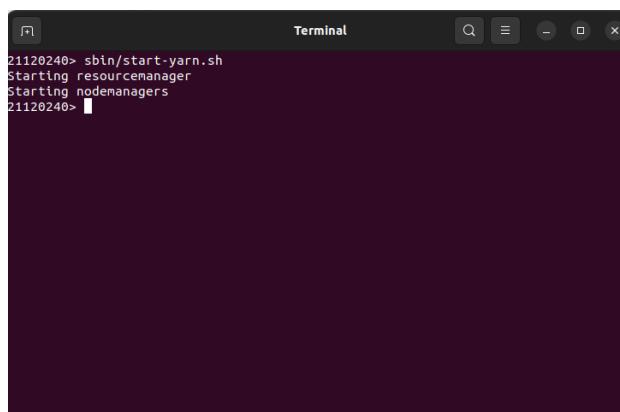
```
1 <?xml version="1.0"?>
2 <!--
3   Licensed under the Apache License, Version 2.0 (the "License");
4   you may not use this file except in compliance with the License.
5   You may obtain a copy of the License at
6
7     http://www.apache.org/licenses/LICENSE-2.0
8
9   Unless required by applicable law or agreed to in writing, software
10  distributed under the License is distributed on an "AS IS" BASIS,
11  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
12  See the License for the specific language governing permissions and
13  limitations under the License. See accompanying LICENSE file.
14 -->
15 <configuration>
16 <!-- Site specific YARN configuration properties -->
17   <property>
18     <name>yarn.nodemanager.aux-services</name>
19     <value>mapreduce_shuffle</value>
20   </property>
21   <property>
22     <name>yarn.nodemanager.env-whitelist</name>
23
24     <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PREPEND_DISTCACHE,</value>
25   </property>
26 </configuration>
```

Saving file "/home/hao/hadoop-3.4.0/etc/hadoop/yarn-site.xml"...

XML Tab Width: 8 Ln 25, Col 17 INS

Start ResourceManager daemon and NodeManager daemn

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



Check ResourceManager by accessing localhost:8088

The screenshot shows the Hadoop ResourceManager web interface at localhost:8088/cluster. The left sidebar has sections for Cluster (About Nodes, Node Labels, Applications), Scheduler (NEW, SUBMITTED, ACCEPTED, RUNNING, FINISHED, KILLED), and Tools. The main area displays Cluster Metrics (0 Apps Submitted, 0 Apps Pending, 0 Apps Running, 0 Apps Completed, 0 Containers Running), Cluster Node Metrics (1 Active Node, 0 Decommissioning Nodes, 0 Decommissioned), and Scheduler Metrics (Scheduler Type: Capacity Scheduler, Scheduling Resource Type: [memory:mb (unit=M), vcores], Minimum Allocation: <memory:1024, vCores:1>, Maximum Allocation: <memory:8192, vCores:1>). A table below shows application details with columns: ID, User, Name, Application Type, Application Tags, Queue, Application Priority, StartTime, LaunchTime, and Finish.

Run `jps` to check

The terminal window shows the output of the `jps` command. It lists several Java processes running on port 21120240:

```
21120240> jps
10693 Jps
9430 DataNode
10198 ResourceManager
10311 NodeManager
9611 SecondaryNameNode
9310 NameNode
21120240>
```

d. Tôn Anh Huy - 21120257

Prerequisites

- Java

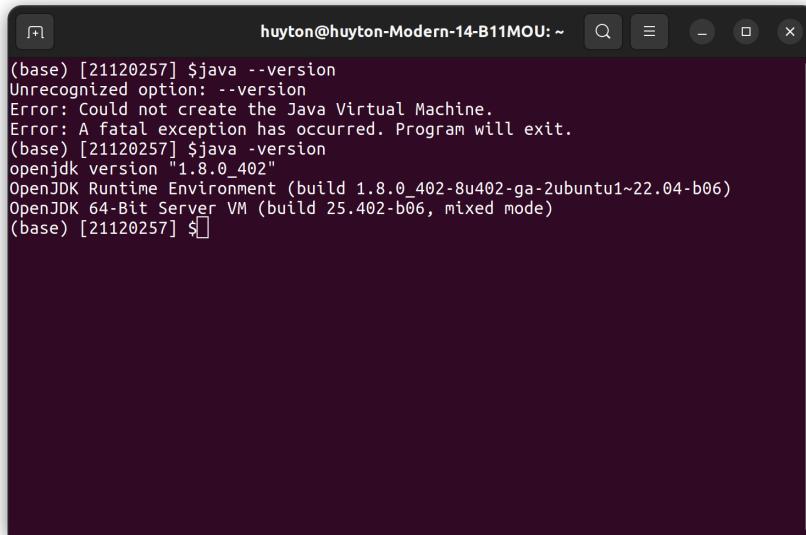
According to suggestions, we should install Java 8 (because Java 11 runtime only)

Type this in terminal to install Java

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

Check if we install successfully

```
java -version
```

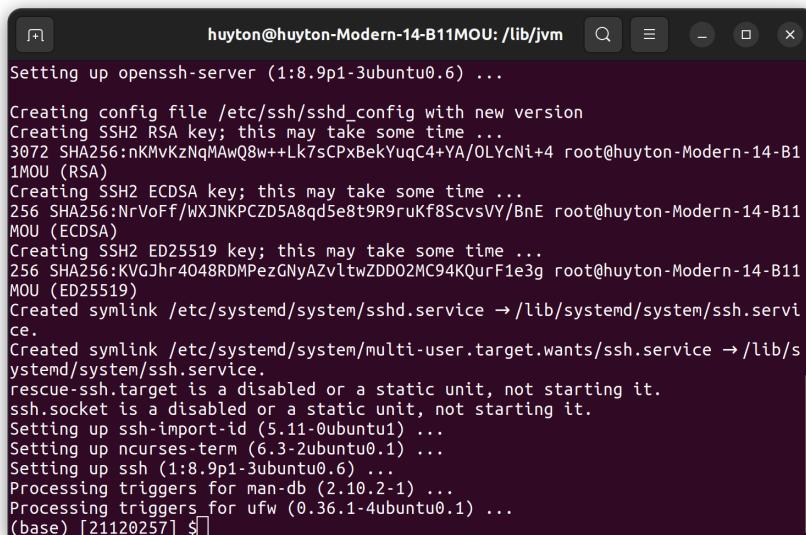


A terminal window titled "huyton@huyton-Modern-14-B11MOU: ~". The command "java -version" is run, resulting in the following output:

```
(base) [21120257] $java --version
Unrecognized option: --version
Error: Could not create the Java Virtual Machine.
Error: A fatal exception has occurred. Program will exit.
(base) [21120257] $java -version
openjdk version "1.8.0_402"
OpenJDK Runtime Environment (build 1.8.0_402-8u402-ga-2ubuntu1~22.04-b06)
OpenJDK 64-Bit Server VM (build 25.402-b06, mixed mode)
(base) [21120257] $
```

- **SSH**

```
sudo apt-get install ssh
```



A terminal window titled "huyton@huyton-Modern-14-B11MOU: /lib/jvm". The command "sudo apt-get install ssh" is run, resulting in the following log output:

```
Setting up openssh-server (1:8.9p1-3ubuntu0.6) ...
Creating config file /etc/ssh/sshd_config with new version
Creating SSH2 RSA key; this may take some time ...
3072 SHA256:nKMvKzNqMAwQ8w++Lk7sCPxBekYuqC4+YA/OLYcNi+4 root@huyton-Modern-14-B1
1MOU (RSA)
Creating SSH2 ECDSA key; this may take some time ...
256 SHA256:NrVoFF/WXJNKPCZD5A8qd5e8t9R9ruKf8ScvsVY/BnE root@huyton-Modern-14-B1
MOU (ECDSA)
Creating SSH2 ED25519 key; this may take some time ...
256 SHA256:KVGJhr4048RDMPezGNyAZvltwZDD02MC94KQurF1e3g root@huyton-Modern-14-B1
MOU (ED25519)
Created symlink /etc/systemd/system/sshd.service → /lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/s
ystemd/system/ssh.service.
rescue-ssh.target is a disabled or a static unit, not starting it.
ssh.socket is a disabled or a static unit, not starting it.
Setting up ssh-import-id (5.11-0ubuntu1) ...
Setting up ncurses-term (6.3-2ubuntu0.1) ...
Setting up ssh (1:8.9p1-3ubuntu0.6) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for ufw (0.36.1-4ubuntu0.1) ...
(base) [21120257] $
```

Download Hadoop

Go to this link: <https://hadoop.apache.org/releases.html> then download the latest stable version

In directory that containing downloaded file, type this line to extract

```
tar xvzf hadoop-3.4.0.tar.gz
```

After that, in the current directory will has a folder named `hadoop-3.4.0`

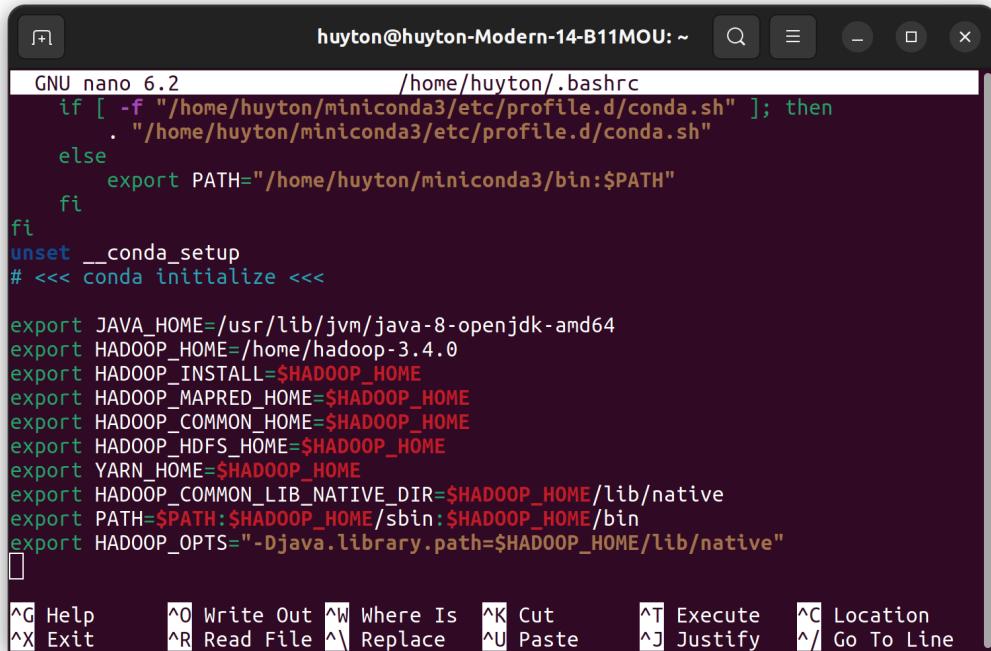
Set up environment variables for Hadoop (file .bashrc)

Open .bashrc file

```
sudo nano ~/.bashrc
```

Paste all lines to .bashrc then save it

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=/home/huyton/hadoop-3.4.0
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```



```
GNU nano 6.2 /home/huyton/.bashrc
if [ -f "/home/huyton/miniconda3/etc/profile.d/conda.sh" ]; then
    . "/home/huyton/miniconda3/etc/profile.d/conda.sh"
else
    export PATH="/home/huyton/miniconda3/bin:$PATH"
fi
unset __conda_setup
# <<< conda initialize <<<

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=/home/hadoop-3.4.0
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

Set up JAVA_HOME

In `hadoop-3.4.0/etc/hadoop/hadoop-env.sh`, find `export JAVA_HOME` line to edit

```
# 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-8-openjdk-amd64
```

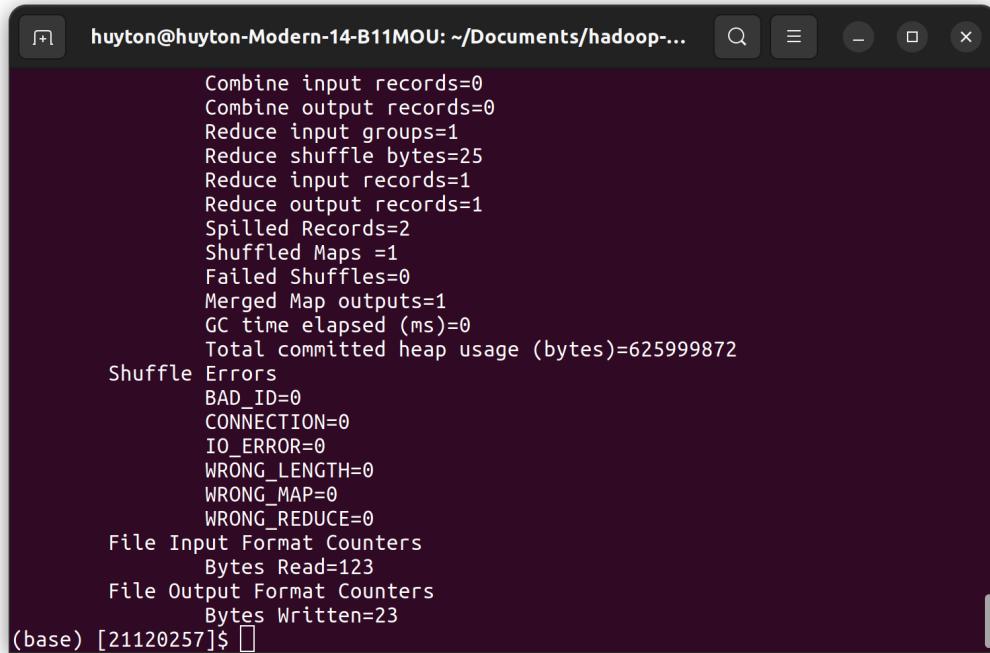
Standalone Operation

By default, Hadoop is configured to run in a non-distributed mode, as a single Java process.

In `hadoop-3.4.0` folder, type these command

```
mkdir input
cp etc/hadoop/*.xml input
bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples.jar wordcount input output/
```

It will copy the unpacked config directory to input folder and then finds and displays every match of the given regex. Output is written to the given output directory.



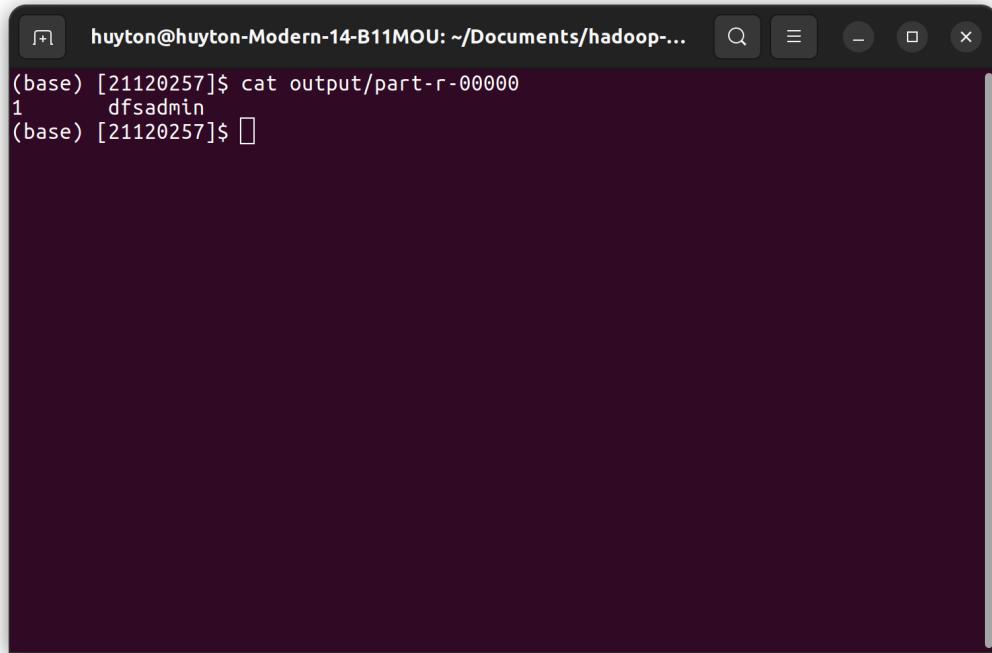
A screenshot of a terminal window titled "huyton@huyton-Modern-14-B11MOU: ~/Documents/hadoop-...". The window shows the following Hadoop command output:

```
Combine input records=0
Combine output records=0
Reduce input groups=1
Reduce shuffle bytes=25
Reduce input records=1
Reduce output records=1
Spilled Records=2
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=0
Total committed heap usage (bytes)=625999872
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=123
File Output Format Counters
Bytes Written=23
(base) [21120257]$
```

Check the output content by this command:

```
cat output/part-r-00000
```

Here are what we get:



huyton@huyton-Modern-14-B11MOU: ~/Documents/hadoop-...
(base) [21120257]\$ cat output/part-r-00000
1 dfsadmin
(base) [21120257]\$

Pseudo-Distributed Operation

Config `etc/hadoop/core-site.xml`

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

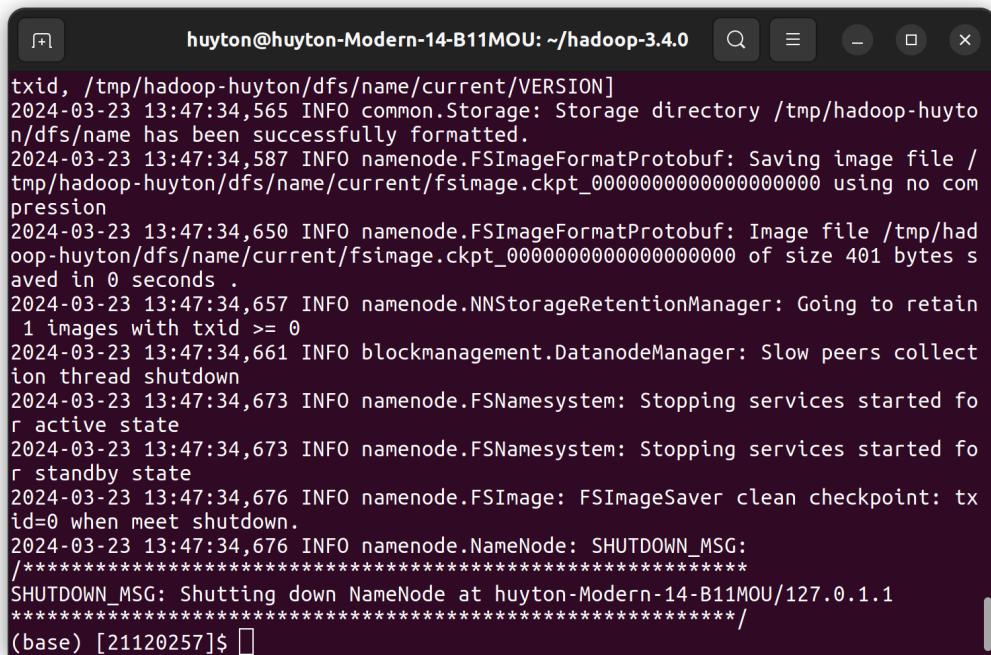
Config `etc/hadoop/hdfs-site.xml`

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

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

Format filesystem

```
bin/hdfs namenode -format
```

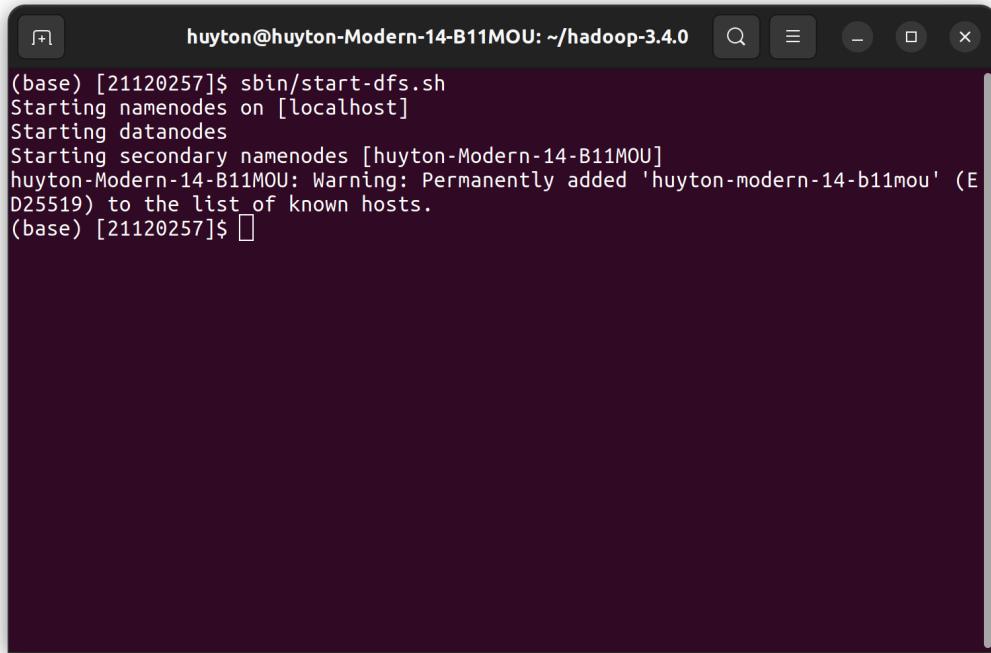


A terminal window titled "huyton@huyton-Modern-14-B11MOU: ~/hadoop-3.4.0". The window displays log output from the Hadoop NameNode format command. The log shows the process of creating a new fsimage and checkpoint files, retaining one image, and shutting down the NameNode.

```
txid, ./tmp/hadoop-huyton/dfs/name/current/VERSION]  
2024-03-23 13:47:34,565 INFO common.Storage: Storage directory /tmp/hadoop-huyton/dfs/name has been successfully formatted.  
2024-03-23 13:47:34,587 INFO namenode.FSImageFormatProtobuf: Saving image file /tmp/hadoop-huyton/dfs/name/current/fsimage.ckpt_00000000000000000000 using no compression  
2024-03-23 13:47:34,650 INFO namenode.FSImageFormatProtobuf: Image file /tmp/hadoop-huyton/dfs/name/current/fsimage.ckpt_00000000000000000000 of size 401 bytes saved in 0 seconds.  
2024-03-23 13:47:34,657 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0  
2024-03-23 13:47:34,661 INFO blockmanagement.DatanodeManager: Slow peers collection thread shutdown  
2024-03-23 13:47:34,673 INFO namenode.FSNamesystem: Stopping services started for active state  
2024-03-23 13:47:34,673 INFO namenode.FSNamesystem: Stopping services started for standby state  
2024-03-23 13:47:34,676 INFO namenode.FSImage: FSImageSaver clean checkpoint: tx id=0 when meet shutdown.  
2024-03-23 13:47:34,676 INFO namenode.NameNode: SHUTDOWN_MSG:  
*****  
SHUTDOWN_MSG: Shutting down NameNode at huyton-Modern-14-B11MOU/127.0.1.1  
*****  
(base) [21120257]$
```

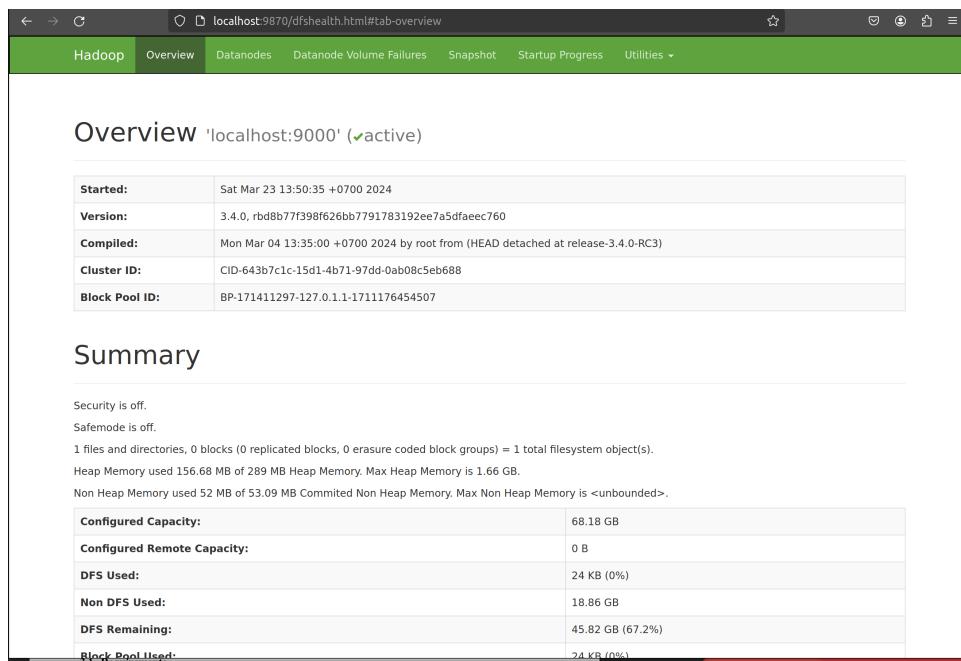
Start NameNode daemon and DataNode daemon

```
sbin/start-dfs.sh
```



```
(base) [21120257]$ sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [huyton-Modern-14-B11MOU]
huyton-Modern-14-B11MOU: Warning: Permanently added 'huyton-modern-14-b11mou' (E
D25519) to the list of known hosts.
(base) [21120257]$
```

Check web interface of NameNode at localhost:9870



localhost:9870/dfshealth.html#tab-overview

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities ▾

Overview 'localhost:9000' (✓active)

Started:	Sat Mar 23 13:50:35 +0700 2024
Version:	3.4.0, rbd8b77f398f626bb7791783192ee7a5dfaec760
Compiled:	Mon Mar 04 13:35:00 +0700 2024 by root from (HEAD detached at release-3.4.0-RC3)
Cluster ID:	CID-643b7c1c-15d1-4b71-97dd-0ab08c5eb688
Block Pool ID:	BP-171411297-127.0.1.1-1711176454507

Summary

Security is off.
Safemode is off.
1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).
Heap Memory used 156.68 MB of 289 MB Heap Memory. Max Heap Memory is 1.66 GB.
Non Heap Memory used 52 MB of 53.09 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	68.18 GB
Configured Remote Capacity:	0 B
DFS Used:	24 KB (0%)
Non DFS Used:	18.86 GB
DFS Remaining:	45.82 GB (67.2%)
Block Pool Used:	24 KB (0%)

Create directory required to execute MapReduce jobs:

```
bin/dfs dfs -mkdir -p /user/21120257
```

Config [etc/hadoop/mapred-site.xml](#)

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
    <name>mapreduce.application.classpath</name>
    <value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HAI
  </property>
</configuration>
```

Config [etc/hadoop/yarn-site.xml](#)

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.env-whitelist</name>
    <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HAI
  </property>
</configuration>
```

Start ResourceManager daemon and NodeManager daemn

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

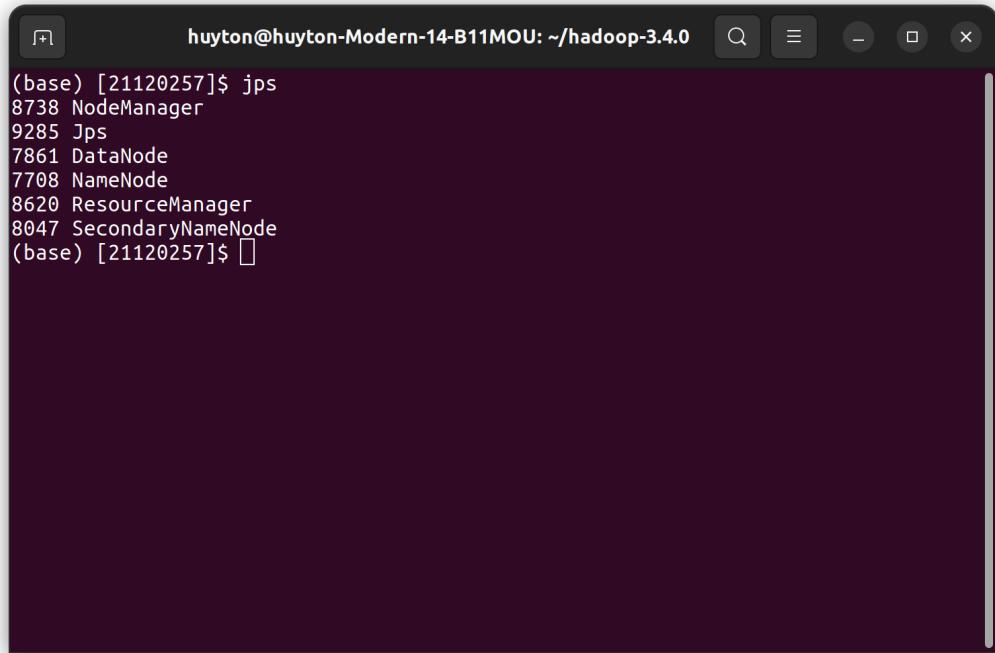
```
huyton@huyton-Modern-14-B11MOU: ~/hadoop-3.4.0
(base) [21120257]$ sbin/start-yarn.sh
Starting resourcemanager
Starting nodemanagers
(base) [21120257]$
```

Check web interface of ResourceManager at localhost:8088

The screenshot shows the Hadoop ResourceManager web interface. The URL is `localhost:8088/cluster`. The interface includes:

- Cluster Metrics:** Shows 0 Apps Submitted, 0 Apps Pending, 0 Apps Running, 0 Apps Completed, and 0 Containers Running.
- Cluster Nodes Metrics:** Shows 1 Active Node and 0 Decommissioning Nodes.
- Scheduler Metrics:** Shows the Scheduler Type as Capacity Scheduler, Scheduling Resource Type as [memory-mb (unit=Mi), vcores], Minimum Allocation as <memory:1024, vCores:1>, and Maximum Allocation as <memory:8192, vCores:1>. It also shows a table for Application Queues with columns: ID, User, Name, Application Type, Application Tags, Queue, Application Priority, Start Time, Launch Time, and Finish Time. The table is currently empty, showing 0 to 0 of 0 entries.

Run `jps` to check



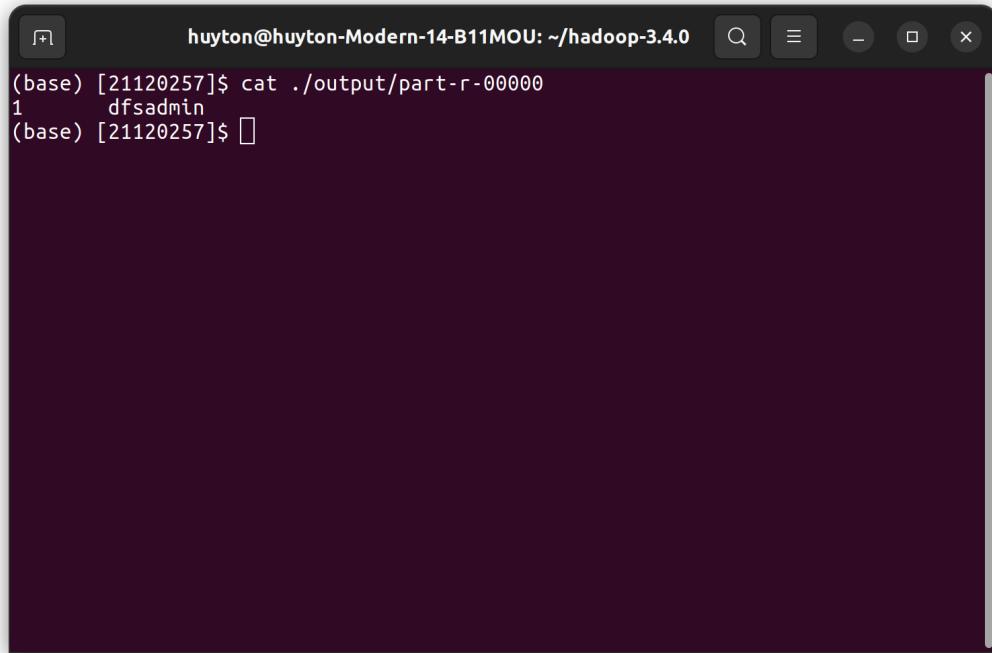
A screenshot of a terminal window titled "huyton@huyton-Modern-14-B11MOU: ~/hadoop-3.4.0". The window shows the output of the command "jps", which lists several Hadoop daemon processes:

```
(base) [21120257]$ jps
8738 NodeManager
9285 Jps
7861 DataNode
7708 NameNode
8620 ResourceManager
8047 SecondaryNameNode
(base) [21120257]$
```

Run mapreduce task like above by:

- Copy input files from local to hdfs
- Run mapreduce on hdfs
- Copy output to local

Result:



A screenshot of a terminal window titled "huyton@huyton-Modern-14-B11MOU: ~/hadoop-3.4.0". The window shows the command "cat ./output/part-r-00000" being run, which outputs the single line "dfsadmin".

```
(base) [21120257]$ cat ./output/part-r-00000
1    dfsadmin
(base) [21120257]$
```

2. Introduction to MapReduce (2 points)

How do the input keys-values, the intermediate keys-values, and the output keys-values relate?

- In Map function:
 - Map takes an input pair and produces a set of intermediate key-value pairs.
 - These intermediate pairs serve as temporary storage before reaching the reducer.
- In Reduce function:
 - The reducer receives an intermediate key "l" and a set of values associated with that key.
 - The reducer processes these grouped pairs and merges together the values for key "l" to form a possibly smaller set of values.
 - The final output key-value pairs are generated by the reducer.

In summary, the input keys-values represent the initial data, intermediate keys-values capture temporary results during computation, and output keys-values represent the final outcome of the MapReduce process

How does MapReduce deal with node failures?

The master pings workers (map nodes and reduce nodes) periodically to make sure that they are in a good condition. If master does not receive any responses from workers then it will mark them as failed.

- Map node

Completed map tasks must be re-executed whenever its node failed. Because their output (intermediate output) is stored in local machine that result in inaccessible problem.

Whenever a map node failed, it will transfer the current task to another node to execute again. Besides, all reduce nodes must be notified that there is a re-execution.

- Reduce node

Once reduce node complete its execution and produces output, that output will be stored in a global file system (such as HDFS). Therefore, there is no need to re-execute the task because output now is accessible.

- Master node

The master node write periodically checkpoints of state of nodes, location... . If it failed, the system can restore its functionality by starting a new copy from the last checkpoint.

What is the meaning and implication of locality? What does it use?

The “locality” term consider the location between input file and machine that execute map task.

There are some replicas of each data which stored in different machines.

The master node will manage the location information of input file and try to schedule a map task on a machine that already has data. If it meet failure, it will try

to schedule this task on other machines that are near input file.

This strategy minimizes data transfer over the network because data can be processed locally on the same machine where already has input data.

Moreover, it also help to reduce network bandwidth usage which lead to less execution time and enhance performance

Which problem is addressed by introducing a combiner function to the MapReduce model?

In map tasks, the repetition in intermediate keys usually exists

In WordCount example, we know that words frequencies tend to follow Zipf distribution (word has rank 1 will occur approximately twice as the word rank 2). Therefore, we produced a lot of same records which lead to sending a significant data across the network (because intermediate output from Map machines are sent to Reduce machines).

With combiner function, we reduce the amount of data above by performing local merging in each map machine before sending them across the network. The code used to implement combiner function and reduce function usually are the same.

The advantages of applying combiner function in MapReduce model:

- Network congestion problem: because the amount of data are significantly smaller
- Data transfer costs problem: because combiner function try to minimize the data by combining repetitions so that the size of data is smaller
- Speed up the MapReduce job: the faster data transfer and decrease network congestion help enhance the performance considerably

3. Running a warm-up problem: Word Count (2 points)

3.1 Create source code files

Firstly, we must create `WordCount.java` using the provided source code from the tutorial

Apache Hadoop 3.4.0 – MapReduce Tutorial

This document comprehensively describes all user-facing facets of the Hadoop MapReduce framework and serves as a tutorial.

🔗 <https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html#Purpose>

3.2 Compile `WordCount.java` and create a jar

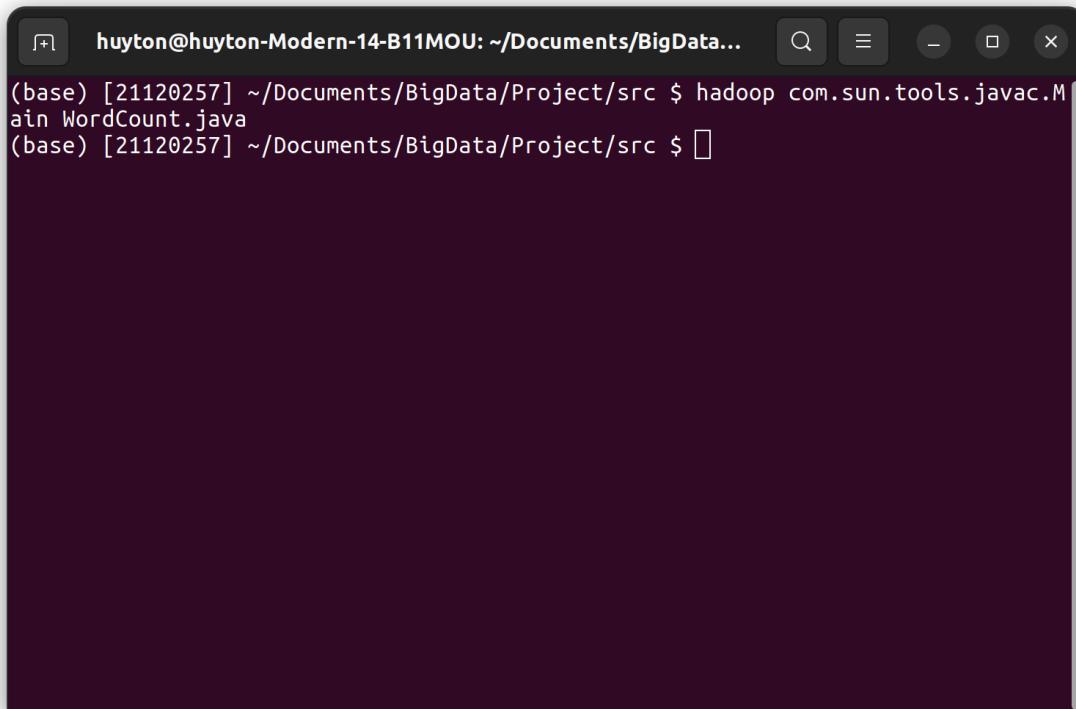
It may raise an error if we do not add the following line into `.bashrc`

```
export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar
```

This line used to add the `tools.jar` file from Java installation to the Hadoop classpath.

However, we can add it only for the current session instead of writing in `.bashrc` as well

```
$ bin/hadoop com.sun.tools.javac.Main WordCount.java
```



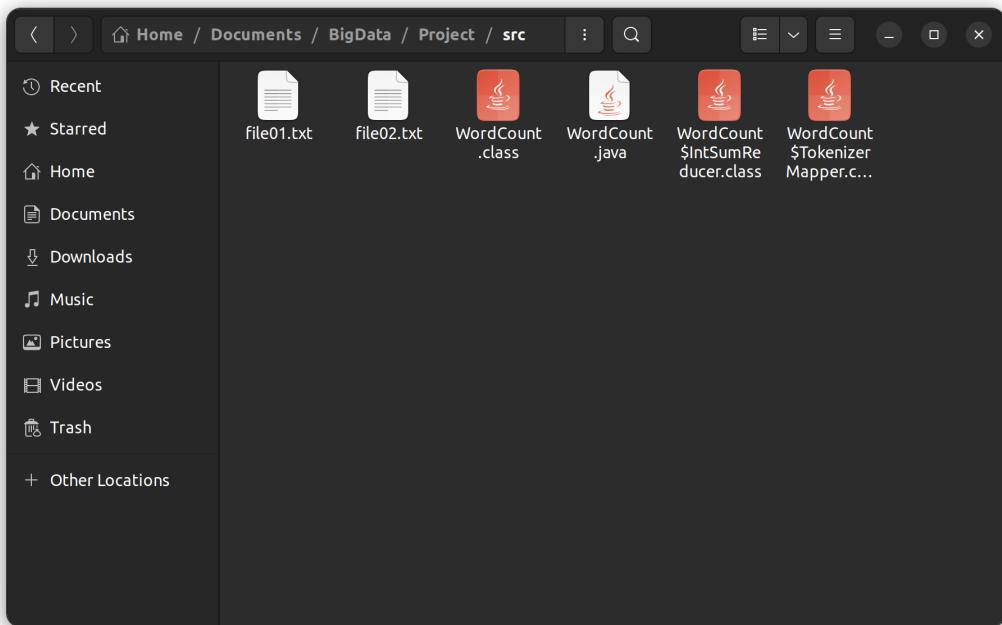
The screenshot shows a terminal window with a dark background. The title bar reads "huyton@huyton-Modern-14-B11MOU: ~/Documents/BigData...". The command entered is "bin/hadoop com.sun.tools.javac.Main WordCount.java". The output shows the command being run and then a prompt at the end of the line.

```
(base) [21120257] ~/Documents/BigData/Project/src $ hadoop com.sun.tools.javac.Main WordCount.java
(base) [21120257] ~/Documents/BigData/Project/src $ 
```

The above command will use Hadoop's bundled Java compilers to compile our file `WordCount.java`

`Com.sun.tools.javac.Main` is the main class in the compiler

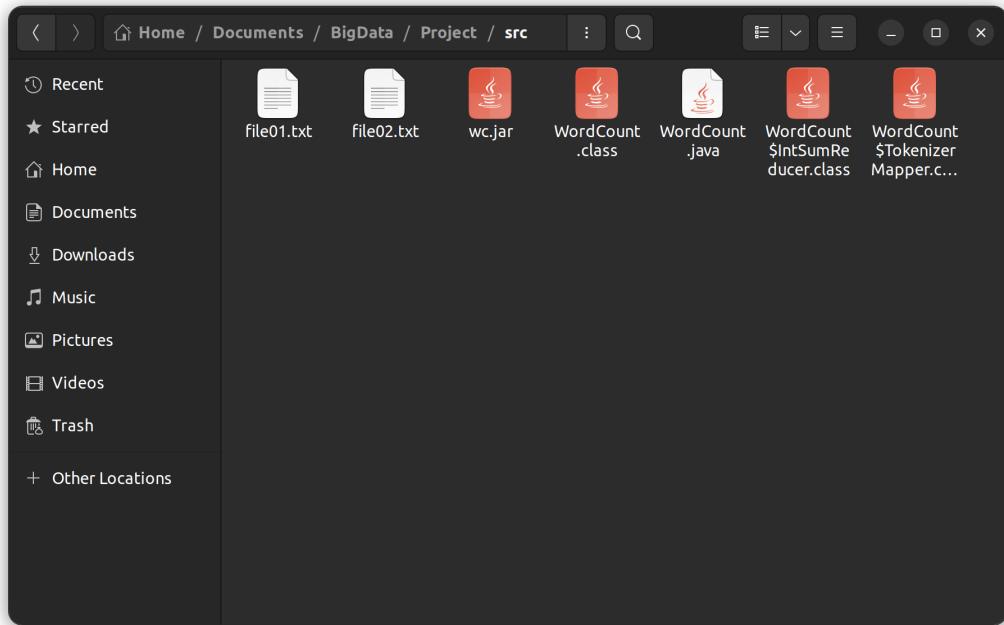
After the compiling process, we should receive `.class` files as below:



```
$ jar cf wc.jar WordCount*.class
```

This command will create a `.jar` (Java Archive) file then named it `wc.jar` in current folder by using all `.class` files whose name starts with `WordCount`

Once it finish, we must get a file `wc.jar` as below



3.3 Create input files and input folders

In the current working directory, we will create 2 files named `file01.txt` and `file02.txt` that are considered as input files

Data in `file01.txt`

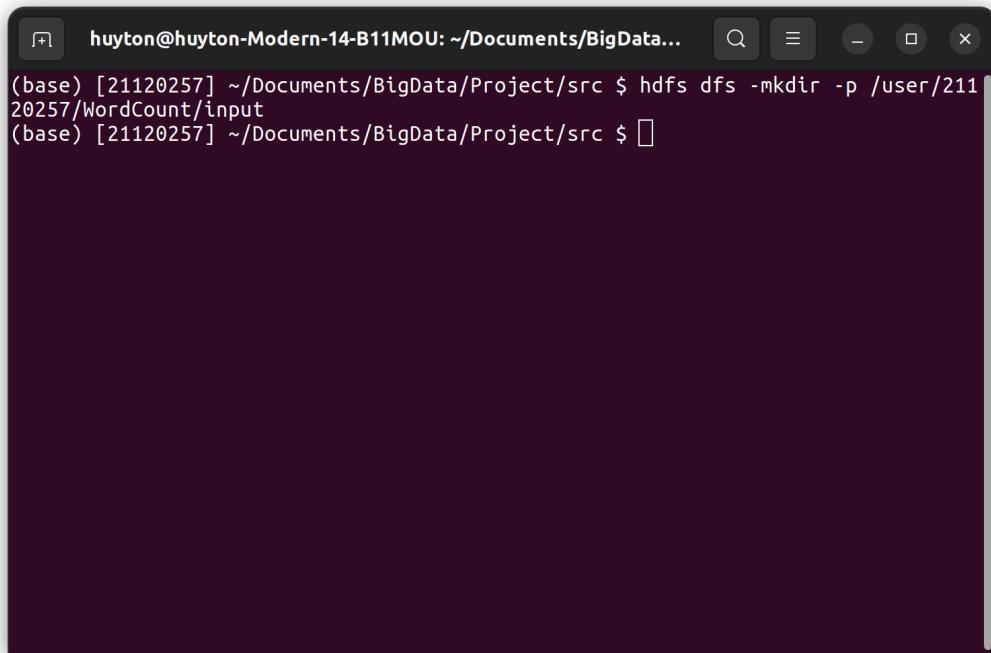
```
Hello World Bye World
```

Data in `file02.txt`

```
Hello Hadoop Goodbye Hadoop
```

Next, we will create input folder in hdfs

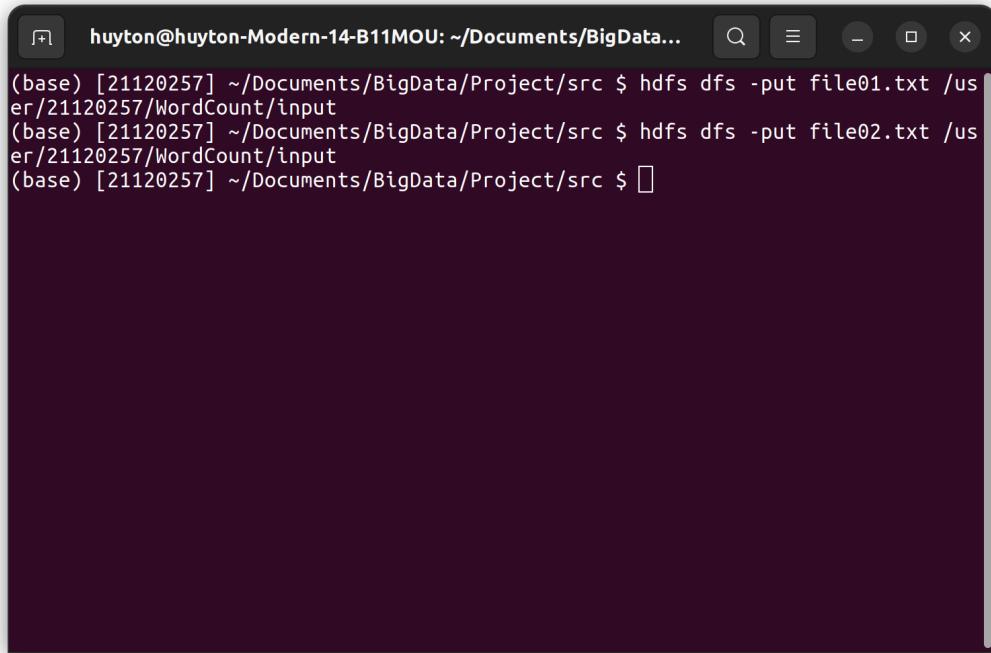
```
hdfs dfs -mkdir -p /user/21120257/WordCount/input
```



A screenshot of a terminal window titled "huyton@huyton-Modern-14-B11MOU: ~/Documents/BigData...". The window shows the command "hdfs dfs -mkdir -p /user/21120257/WordCount/input" being run, followed by a prompt "(base) [21120257] ~/Documents/BigData/Project/src \$".

Now, we will copy input file that we have just created before to hdfs

```
hdfs dfs -put file01.txt /user/21120257/WordCount/input  
hdfs dfs -put file02.txt /user/21120257/WordCount/input
```

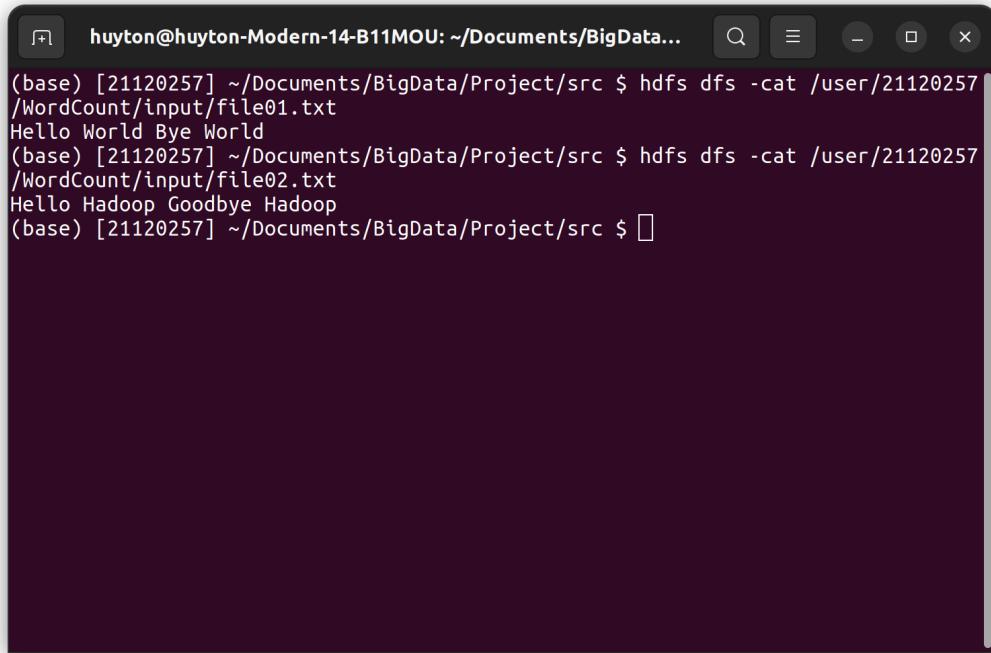


A screenshot of a terminal window titled "huyton@huyton-Modern-14-B11MOU: ~/Documents/BigData...". The window shows three commands being run:

```
(base) [21120257] ~/Documents/BigData/Project/src $ hdfs dfs -put file01.txt /user/21120257/WordCount/input  
(base) [21120257] ~/Documents/BigData/Project/src $ hdfs dfs -put file02.txt /user/21120257/WordCount/input  
(base) [21120257] ~/Documents/BigData/Project/src $
```

We will check to make sure that the input files are ready

```
hdfs dfs -cat /user/21120257/input/WordCount/file01.txt  
hdfs dfs -cat /user/21120257/input/WordCount/file02.txt
```



A screenshot of a terminal window titled "huyton@huyton-Modern-14-B11MOU: ~/Documents/BigData...". The window shows two commands being run: "hdfs dfs -cat /user/21120257/WordCount/input/file01.txt" and "hdfs dfs -cat /user/21120257/WordCount/input/file02.txt". The output of the first command is "Hello World Bye World" and the output of the second command is "Hello Hadoop Goodbye Hadoop". The terminal has a dark background and light-colored text.

```
(base) [21120257] ~/Documents/BigData/Project/src $ hdfs dfs -cat /user/21120257/WordCount/input/file01.txt
Hello World Bye World
(base) [21120257] ~/Documents/BigData/Project/src $ hdfs dfs -cat /user/21120257/WordCount/input/file02.txt
Hello Hadoop Goodbye Hadoop
(base) [21120257] ~/Documents/BigData/Project/src $
```

Now, the input files are ready for MapReduce

3.4 Run the application

To run the WordCount we run the following command

```
hadoop jar wc.jar WordCount /user/21120257/WordCount/input /user/21120257/WordCount/output
```

The command is used to run a Hadoop job using the `WordCount` class from the `wc.jar` file.

It uses all files in `/user/21120257/WordCount/input` as input and save the result to `/user/21120257/WordCount/output` folder

Note: the output folder should'nt be created before because Hadoop MapReduce will throw an error when output folder already exists. This is built to prevent accidental overwriting data in hdfs.

The result in terminal screen:

```
huyton@huyton-Modern-14-B11MOU: ~/Documents/BigData... 
    Shuffled Maps =2
    Failed Shuffles=0
    Merged Map outputs=2
    GC time elapsed (ms)=195
    CPU time spent (ms)=1300
    Physical memory (bytes) snapshot=927711232
    Virtual memory (bytes) snapshot=7732936704
    Total committed heap usage (bytes)=769130496
    Peak Map Physical memory (bytes)=348729344
    Peak Map Virtual memory (bytes)=2576433152
    Peak Reduce Physical memory (bytes)=245465088
    Peak Reduce Virtual memory (bytes)=2582196224
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=50
File Output Format Counters
Bytes Written=41
(base) [21120257] ~/Documents/BigData/Project/src $ 
```

Now we will check our output (the content in `part-r-00000`)

```
hdfs dfs -cat /user/21120257/WordCount/output/part-r-00000
```

```
huyton@huyton-Modern-14-B11MOU: ~/Documents/BigData... 
(base) [21120257] ~/Documents/BigData/Project/src $ hdfs dfs -cat /user/21120257/
/WordCount/output/part-r-00000
Bye      1
Goodbye 1
Hadoop   2
Hello    2
World    2
(base) [21120257] ~/Documents/BigData/Project/src $ 
```

4. Bonus (2 points)

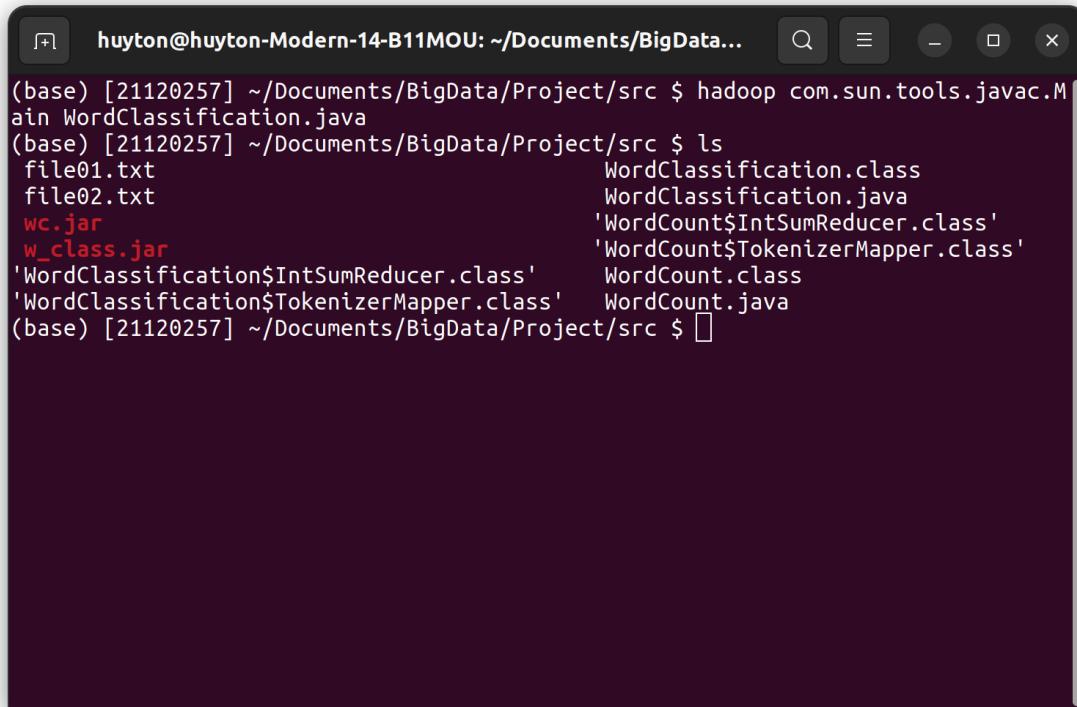
4.1 Word Length Count (0.5 points)

In `WordCount.java`, we will reuse and modify the map method in order to handle classification tasks.

The new files named `WordClassification.java`

Compile WordClassification

```
$ hadoop com.sun.tools.javac.Main WordClassification.java
```



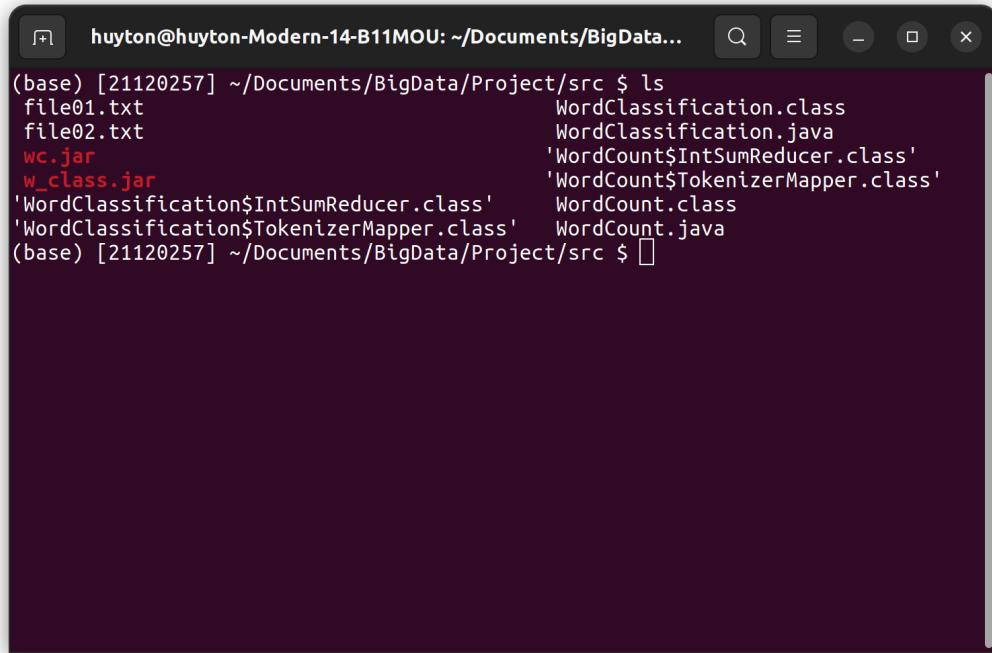
A terminal window showing the compilation of `WordClassification.java` and a subsequent `ls` command to list the contents of the directory. The output shows several class files and two text files.

```
huyton@huyton-Modern-14-B11MOU: ~/Documents/BigData...  
(base) [21120257] ~/Documents/BigData/Project/src $ hadoop com.sun.tools.javac.Main WordClassification.java  
(base) [21120257] ~/Documents/BigData/Project/src $ ls  
file01.txt           WordClassification.class  
file02.txt           WordClassification.java  
wc.jar             'WordCount$IntSumReducer.class'  
w_class.jar         'WordCount$TokenizerMapper.class'  
'WordClassification$IntSumReducer.class'  WordCount.class  
'WordClassification$TokenizerMapper.class' WordCount.java  
(base) [21120257] ~/Documents/BigData/Project/src $
```

After this step, we will get `.class` files

Create `.jar` file from `.class` files we have just created

```
$ hadoop jar cf w_class.jar WordClassification*.class
```



```
huyton@huyton-Modern-14-B11MOU: ~/Documents/BigData... (base) [21120257] ~/Documents/BigData/Project/src $ ls  
file01.txt           WordClassification.class  
file02.txt           WordClassification.java  
wc.jar             'WordCount$IntSumReducer.class'  
w_class.jar         'WordCount$TokenizerMapper.class'  
'WordClassification$IntSumReducer.class' WordCount.class  
'WordClassification$TokenizerMapper.class' WordCount.java  
(base) [21120257] ~/Documents/BigData/Project/src $ 
```

After this step, we should get `w_class.jar`

Run the application

Assume that we have input files in `/user/WordClassification/input` already

```
$ hadoop jar w_class.jar WordClassification /user/WordClassification/input
```

```
huyton@huyton-Modern-14-B11MOU: ~/Documents/BigData... 
    Shuffled Maps =2
    Failed Shuffles=0
    Merged Map outputs=2
    GC time elapsed (ms)=715
    CPU time spent (ms)=3930
    Physical memory (bytes) snapshot=949035008
    Virtual memory (bytes) snapshot=7734382592
    Total committed heap usage (bytes)=776994816
    Peak Map Physical memory (bytes)=353435648
    Peak Map Virtual memory (bytes)=2576269312
    Peak Reduce Physical memory (bytes)=245067776
    Peak Reduce Virtual memory (bytes)=2583867392
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=50
File Output Format Counters
Bytes Written=17
(base) [21120257] ~/Documents/BigData/Project/src $ 
```

Now we will check the output

```
$ hdfs dfs -cat /user/WordClassification/output/part-r-00000
```

```
huyton@huyton-Modern-14-B11MOU: ~/Documents/BigData... 
(base) [21120257] ~/Documents/BigData/Project/src $ hdfs dfs -cat /user/WordClassification/output/part-r-00000
Medium 7
Small 1
(base) [21120257] ~/Documents/BigData/Project/src $ 
```

III. Reflection

- Knowing the workflow of MapReduce job
- All members have installed Hadoop successfully
- Knowing how to create and run a program of MapReduce
- Knowing some basic command line in HDFS

IV. References

<https://research.google/pubs/mapreduce-simplified-data-processing-on-large-clusters/>