Lab 01: Introduction to Hadoop Ecosystem

Set up a Hadoop cluster

Name: Nguyễn Thanh Tuấn

Student ID: 22120405

TABLE OF CONTENTS

Content

I. Installation:

0. Prerequisite:

Running on Ubuntu 22.04 Operating System.

1. System preparation:

a. Update apt package manager to the newest version:

```
ubuntu@bigdata:~$ sudo apt update && sudo apt upgrade -y
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
15 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  landscape-common libnss-systemd libpam-systemd libplymouth5 libsystemd-shared libsystemd0
  libudev1 plymouth plymouth-theme-ubuntu-text systemd systemd-dev systemd-resolved
  systemd-sysv systemd-timesyncd udev
15 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 9241 kB of archives.
After this operation, 8192 B disk space will be freed.
Get:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libnss-systemd amd64 255.4-1ubu
ntu8.6 [159 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 systemd-dev all 255.4-1ubuntu8.
6 [104 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 systemd-timesyncd amd64 255.4-1
```

b. Install Java:

- Apache Hadoop 3.3 and upper supports Java 8 and Java 11 (runtime only). [1]
- Install java 11 of OpenJDK.

```
ubuntu@bigdata:~$ sudo apt install openjdk-11-jdk -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  alsa-topology-conf alsa-ucm-conf
  at-spi2-common at-spi2-core
  ca-certificates-java dconf-gsettings-backend
  dconf-service fonts-dejavu-extra
  gsettings-desktop-schemas java-common
  libasound2-data libasound2t64
  libatk-bridge2.0-0t64 libatk-wrapper-java
  libatk-wrapper-java-jni libatk1.0-0t64
  libatspi2.0-0t64 libavahi-client3
  libavahi-common-data libavahi-common3
  libcups2t64 libdconf1 libdrm-amdgpu1
  libdrm-intel1 libdrm-nouveau2 libdrm-radeon1
  libgbm1 libgif7 libgl1 libgl1-amber-dri
  libgl1-mesa-dri libglapi-mesa libglvnd0
  libglx-mesa0 libglx0 libgraphite2-3
  libharfbuzz0b libice-dev libice6 liblcms2-2
  libllvm19 libpciaccess0 libpcsclite1
  libpthread-stubs0-dev libsm-dev libsm6
  libvulkan1 libwayland-client0
  libwayland-server0 libx11-dev libx11-xcb1
```

JDK is installed into: "`/usr/lib/jvm/java-11-openjdk-amd64`".

c. <u>Install ssh and pdsh:</u>

```
ubuntu@bigdata:~$ sudo apt-get install ssh -y && sudo apt-get install pdsh -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  ssh
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 4658 B of archives.
After this operation, 57.3 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 ssh all 1:9.6p1-3ubuntu13.8 [46
58 B]
Fetched 4658 B in 0s (9628 B/s)
Selecting previously unselected package ssh.
(Reading database ... 77216 files and directories currently installed.)
Preparing to unpack .../ssh_1%3a9.6p1-3ubuntu13.8_all.deb ...
Unpacking ssh (1:9.6p1-3ubuntu13.8) ...
Setting up ssh (1:9.6p1-3ubuntu13.8) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Running kernel seems to be up-to-date.

Restarting services...
```

2. Install Hadoop:

a. <u>Install Hadoop package:</u>

- Stable version of Hadoop 3.x is Hadoop 3.4.1. [2]

- Install package using wget:

```
ubuntu@bigdata:~$ wget https://dlcdn.apache.org/hadoop/common/stable/hadoop-3.4.1.tar.gz
--2025-03-19 17:34:39--  https://dlcdn.apache.org/hadoop/common/stable/hadoop-3.4.1.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: 974002355 (929M) [application/x-gzip]
Saving to: 'hadoop-3.4.1.tar.gz'

hadoop-3.4.1.tar.gz     100%[===============================>] 928.88M  5.13MB/s    in 2m 44s

2025-03-19 17:38:23 (5.68 MB/s) - 'hadoop-3.4.1.tar.gz' saved [974002355/974002355]
```

b. <u>Install Hadoop signature:</u>

- Install Hadoop signature of hadoop-3.4.1.tar.gz package, which is stored in hadoop-3.4.1.tar.gz.asc.

```
ubuntu@bigdata:~$ wget https://dlcdn.apache.org/hadoop/common/stable/hadoop-3.4.1.tar.gz.asc
--2025-03-19 17:39:29--  https://dlcdn.apache.org/hadoop/common/stable/hadoop-3.4.1.tar.gz.asc
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: 833 [text/plain]
Saving to: 'hadoop-3.4.1.tar.gz.asc'

hadoop-3.4.1.tar.gz.asc 100%[===============================>]     833  --.-KB/s    in 0s

2025-03-19 17:39:29 (12.9 MB/s) - 'hadoop-3.4.1.tar.gz.asc' saved [833/833]
```

c. <u>Install Hadoop public key:</u>

- To check for the integrity of hadoop-3.4.1.tar.gz package, public keys need to be installed.

```
ubuntu@bigdata:~$ wget https://dlcdn.apache.org/hadoop/common/KEYS
--2025-03-19 17:39:52--  https://dlcdn.apache.org/hadoop/common/KEYS
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: 442675 (432K)
Saving to: 'KEYS'

KEYS                    100%[===============================>] 432.30K  2.08MB/s    in 0.2s

2025-03-19 17:39:53 (2.08 MB/s) - 'KEYS' saved [442675/442675]
```

d. <u>Import public keys:</u>

```
ubuntu@bigdata:~$ gpg --import KEYS
gpg: directory '/home/ubuntu/.gnupg' created
gpg: keybox '/home/ubuntu/.gnupg/pubring.kbx' created
gpg: key BE5AAA0BA210C095: 3 signatures not checked due to missing keys
gpg: /home/ubuntu/.gnupg/trustdb.gpg: trustdb created
gpg: key BE5AAA0BA210C095: public key "Arun C. Murthy <acmurthy@apache.org>" imported
gpg: key 220F69801F27E622: 8 signatures not checked due to missing keys
gpg: key 220F69801F27E622: public key "Konstantin I Boudnik (Cos) <cos@boudnik.org>" imported
gpg: key DBAF69BEA7239D59: 8 signatures not checked due to missing keys
gpg: key DBAF69BEA7239D59: public key "Doug Cutting (Lucene guy) <cutting@apache.org>" imported
gpg: key 08458C39E964B5FF: 1 signature not checked due to a missing key
gpg: key 08458C39E964B5FF: public key "Enis Soztutar (CODE SIGNING KEY) <enis@apache.org>" impo
rted
```

e. <u>Check for integrity:</u>

- Using imported public keys to check integrity with the signature files.

```
ubuntu@bigdata:~$ gpg --verify hadoop-3.4.1.tar.gz.asc hadoop-3.4.1.tar.gz
gpg: Signature made Thu Oct 10 00:10:30 2024 +07
gpg:                using RSA key 53931DAA708291409958BD474D22BB7D32882201
gpg: Good signature from "Mukund Thakur <mthakur@apache.org>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 5393 1DAA 7082 9140 9958  BD47 4D22 BB7D 3288 2201
ubuntu@bigdata:~$
```

"Good signature" means the package is ensured for integrity.

f. <u>Extract:</u>

- Extract Hadoop package.

```
ubuntu@bigdata:~$ sudo tar -xvzf hadoop-3.4.1.tar.gz
hadoop-3.4.1/
hadoop-3.4.1/include/
hadoop-3.4.1/include/SerialUtils.hh
hadoop-3.4.1/include/TemplateFactory.hh
hadoop-3.4.1/include/hdfs.h
hadoop-3.4.1/include/StringUtils.hh
hadoop-3.4.1/include/Pipes.hh
hadoop-3.4.1/share/
hadoop-3.4.1/share/doc/
hadoop-3.4.1/share/doc/hadoop/
hadoop-3.4.1/share/doc/hadoop/hadoop-kms/
hadoop-3.4.1/share/doc/hadoop/hadoop-kms/images/
hadoop-3.4.1/share/doc/hadoop/hadoop-kms/images/expanded.gif
hadoop-3.4.1/share/doc/hadoop/hadoop-kms/images/maven-logo-2.gif
hadoop-3.4.1/share/doc/hadoop/hadoop-kms/images/banner.jpg
hadoop-3.4.1/share/doc/hadoop/hadoop-kms/images/bg.jpg
hadoop-3.4.1/share/doc/hadoop/hadoop-kms/images/collapsed.gif
hadoop-3.4.1/share/doc/hadoop/hadoop-kms/images/icon_info_sml.gif
hadoop-3.4.1/share/doc/hadoop/hadoop-kms/images/logo_apache.jpg
hadoop-3.4.1/share/doc/hadoop/hadoop-kms/images/logos/
hadoop-3.4.1/share/doc/hadoop/hadoop-kms/images/logos/build-by-maven-white.png
hadoop-3.4.1/share/doc/hadoop/hadoop-kms/images/logos/maven-feather.png
hadoop-3.4.1/share/doc/hadoop/hadoop-kms/images/logos/build-by-maven-black.png
hadoop-3.4.1/share/doc/hadoop/hadoop-kms/images/icon_success_sml.gif
```

g. <u>Rename folder:</u>

- Folder is renamed as “hadoop”.

```
ubuntu@bigdata:~$ ls
KEYS  hadoop-3.4.1  hadoop-3.4.1.tar.gz  hadoop-3.4.1.tar.gz.asc
ubuntu@bigdata:~$ mv hadoop-3.4.1 hadoop
ubuntu@bigdata:~$ ls
KEYS  hadoop  hadoop-3.4.1.tar.gz  hadoop-3.4.1.tar.gz.asc
```

3. Set up SSH:

a. Check if the ssh is available:

```
ubuntu@bigdata: ~
ubuntu@bigdata:~$ ssh localhost
The authenticity of host 'localhost (::1)' can't be established.
ED25519 key fingerprint is SHA256:QHiThsyopt3HFiz6pXafv6P8N2eGoz6rA1HE8mW+YnI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Host key verification failed.
ubuntu@bigdata:~$
```

The ssh fails as evidence of ssh unavailability.

b. Create keys:

- Keys are created using RSA algorithm with an empty passphrase.
- The keys are stored in `~/.ssh/id_rsa`.

```
ubuntu@bigdata:~$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
Generating public/private rsa key pair.
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:/TKq44ov7PXLXY8sccUm12w4/+/m9R2sfqQXHG6ioP4 ubuntu@bigdata
The key's randomart image is:
+---[RSA 3072]----+
|                 |
|                 |
|            . +  |
|         .. B +. |
|        S .= +o .|
|        .... .o* |
| .  .   .o= o =+o|
|  oo o.o.+ * ..oB|
| .oooB=+Eo ooo+B|
+----[SHA256]-----+
```

c. Configure ssh:

- Put public key into `authorized_keys` file. This is the file consisting of public keys from the remote users, who are allowed to log in the server.

```
ubuntu@bigdata:~$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

- Change permission to 600 for `authorized_keys` file, which allows only the user to read and write.

chmod 0600 ~/.ssh/authorized_keys

```
ubuntu@bigdata:~$ chmod 0600 ~/.ssh/authorized_keys
```

4. Set up environments for Hadoop:

- Open file `~/.bashrc` to set up environment variables:
  - `JAVA_HOME`: Java installation location. Hadoop is built using Java.
  - `HADOOP_HOME`: home directory for Hadoop.
  - `HADOOP_INSTALL`: location where Hadoop is installed.
  - `HADOOP_MAPRED_HOME`: location of Hadoop MapReduce.
  - `HADOOP_COMMON_HOME`: location of common libraries used by Hadoop system.
  - `HADOOP_HDFS_HOME`: location of HDFS.
  - `HADOOP_YARN_HOME`: location of YARN.
  - `HADOOP_COMMON_LIB_NATIVE_DIR`: location of Hadoop's native libraries.
  - `HADOOP_OPTS`: addition options to Hadoop. In this case, the option is set for Java library path to include Hadoop's native libraries.
  - `PATH`: path of finding executable programs. In this case, Hadoop's `sbin` and `bin` directories are included.

```
  GNU nano 7.2                          /home/ubuntu/.bashrc *
# 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=~/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 HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
export PATH="$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin"

File Name to Write: /home/ubuntu/.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
```

- Verification:

```
ubuntu@bigdata:~$ source ~/.bashrc
ubuntu@bigdata:~$ echo $JAVA_HOME
/usr/lib/jvm/java-11-openjdk-amd64
ubuntu@bigdata:~$ echo $HADOOP_HOME
/home/ubuntu/hadoop
ubuntu@bigdata:~$ echo $HADOOP_INSTALL
/home/ubuntu/hadoop
ubuntu@bigdata:~$ echo $HADOOP_MAPRED_HOME
/home/ubuntu/hadoop
ubuntu@bigdata:~$ echo $HADOOP_COMMON_HOME
/home/ubuntu/hadoop
ubuntu@bigdata:~$ echo $HADOOP_HDFS_HOME
/home/ubuntu/hadoop
ubuntu@bigdata:~$ echo $HADOOP_YARN_HOME
/home/ubuntu/hadoop
ubuntu@bigdata:~$ echo $HADOOP_COMMON_LIB_NATIVE_DIR
/home/ubuntu/hadoop/lib/native
ubuntu@bigdata:~$ echo $HADOOP_OPTS
-Djava.library.path=/home/ubuntu/hadoop/lib/native
ubuntu@bigdata:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/
bin:/home/ubuntu/hadoop/sbin:/home/ubuntu/hadoop/bin
```

- Set up `JAVA_HOME` for Hadoop in `etc/hadoop/hadoop.env.sh`:

```
  GNU nano 7.2                    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-11-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
File Name to Write: etc/hadoop/hadoop-env.sh
^G Help          M-D DOS Format     M-A Append         M-B Backup File
^C Cancel        M-M Mac Format     M-P Prepend        ^T Browse
```

- Test Hadoop:

```
ubuntu@bigdata:~$ hadoop
Usage: hadoop [OPTIONS] SUBCOMMAND [SUBCOMMAND OPTIONS]
 or    hadoop [OPTIONS] CLASSNAME [CLASSNAME OPTIONS]
  where CLASSNAME is a user-provided Java class

  OPTIONS is none or any of:

buildpaths                       attempt to
                                 add class
                                 files from
                                 build tree
--config dir                     Hadoop
                                 config
                                 directory
--debug                          turn on
                                 shell
                                 script
                                 debug mode
--help                           usage
                                 information
hostnames list[,of,host,names]   hosts to
                                 use in
                                 worker mode
hosts filename                   list of
                                 hosts to
```

Hadoop runs successfully.

5. Pseudo-distributed configuration:

a. <u>Core configuration:</u>

- Open `etc/hadoop/core-site.xml` file. This file is used for core configuration for Hadoop systems like filesystem, I/O settings, etc.
- Add the following property:

```
  GNU nano 7.2                          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.defaultFS</name>
        <value>hdfs://localhost:9000</value>
    </property>
</configuration>


File Name to Write: etc/hadoop/core-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
```

Setting `fs.defaultFS=hdfs://localhost:9000` means to set default file system of Hadoop. The default file system in this case is HDFS and the NameNode runs on `localhost:9000`.

b. <u>HDFS configuration:</u>

- Open `etc/hadoop/hdfs-site.xml` file. This file controls the overall behaviour of HDFS.
- Add the following property:

```
  GNU nano 7.2                          hadoop/etc/hadoop/hdfs-site.xml *

<!-- 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/ubuntu/hadoop/hdfs/namenode</value>
        </property>

        <property>
                <name>dfs.data.dir</name>
                <value>file:///home/ubuntu/hadoop/hdfs/datanode</value>
        </property>
</configuration>

File Name to Write: hadoop/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
```

Setting `dfs.replication=1` means to set how many copies of each data block should be stored in HDFS. In this case, it is 1.

Setting `dfs.name.dir` means to set directory for NameNode to store necessary information.

Setting `dfs.data.dir` means to set directory for DataNode to store necessary information.

- Create indicated directory manually:

```
ubuntu@bigdata:~$ mkdir -p hadoop/hdfs/{datanode,namenode}
ubuntu@bigdata:~$ ls hadoop/hdfs/
datanode  namenode
```

c. <u>Format the filesystem:</u>

- Format the NameNode in HDFS.

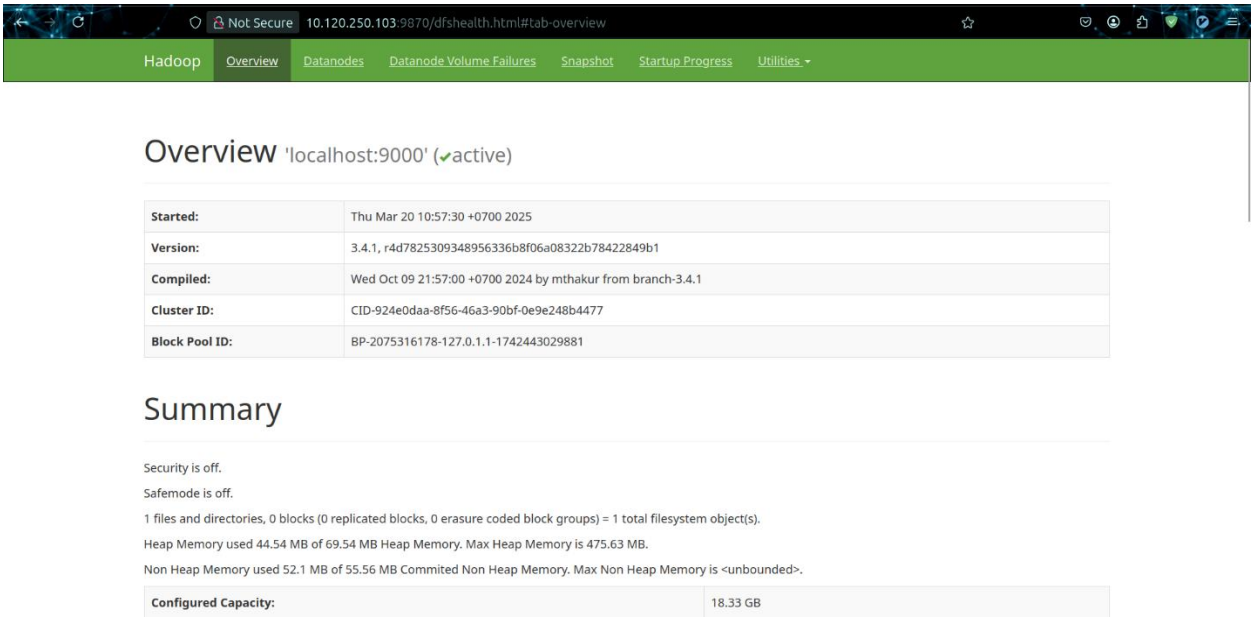
```
ubuntu@bigdata:~/hadoop$ hdfs namenode -format
2025-03-20 10:45:00,993 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = bigdata/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.4.1
STARTUP_MSG:   classpath = /home/ubuntu/hadoop/etc/hadoop:/home/ubuntu/hadoop/share/hadoop/comm
on/lib/commons-beanutils-1.9.4.jar:/home/ubuntu/hadoop/share/hadoop/common/lib/commons-math3-3.
6.1.jar:/home/ubuntu/hadoop/share/hadoop/common/lib/failureaccess-1.0.jar:/home/ubuntu/hadoop/s
hare/hadoop/common/lib/jetty-security-9.4.53.v20231009.jar:/home/ubuntu/hadoop/share/hadoop/com
mon/lib/j2objc-annotations-1.1.jar:/home/ubuntu/hadoop/share/hadoop/common/lib/jaxb-api-2.2.11.
jar:/home/ubuntu/hadoop/share/hadoop/common/lib/animal-sniffer-annotations-1.17.jar:/home/ubunt
u/hadoop/share/hadoop/common/lib/snappy-java-1.1.10.4.jar:/home/ubuntu/hadoop/share/hadoop/comm
on/lib/netty-codec-dns-4.1.100.Final.jar:/home/ubuntu/hadoop/share/hadoop/common/lib/commons-da
emon-1.0.13.jar:/home/ubuntu/hadoop/share/hadoop/common/lib/netty-transport-native-epoll-4.1.10
0.Final.jar:/home/ubuntu/hadoop/share/hadoop/common/lib/stax2-api-4.2.1.jar:/home/ubuntu/hadoop
/share/hadoop/common/lib/slf4j-api-1.7.36.jar:/home/ubuntu/hadoop/share/hadoop/common/lib/netty
-codec-http-4.1.100.Final.jar:/home/ubuntu/hadoop/share/hadoop/common/lib/kerb-core-2.0.3.jar:/
home/ubuntu/hadoop/share/hadoop/common/lib/jaxb-impl-2.2.3-1.jar:/home/ubuntu/hadoop/share/hado
op/common/lib/jetty-http-9.4.53.v20231009.jar:/home/ubuntu/hadoop/share/hadoop/common/lib/gson-
2.9.0.jar:/home/ubuntu/hadoop/share/hadoop/common/lib/netty-transport-classes-epoll-4.1.100.Fin
al.jar:/home/ubuntu/hadoop/share/hadoop/common/lib/kerby-util-2.0.3.jar:/home/ubuntu/hadoop/sha
re/hadoop/common/lib/netty-resolver-dns-native-macos-4.1.100.Final-osx-aarch_64.jar:/home/ubunt
u/hadoop/share/hadoop/common/lib/commons-configuration2-2.10.1.jar:/home/ubuntu/hadoop/share/ha
```

d. Start HDFS:

```
ubuntu@bigdata:~/hadoop$ start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [bigdata]
```

e. Test HDFS:

By now, NameNode web interface is available at `http://localhost:9870/`.



Web interface works normally.

6. **YARN Configuration:**

a. <u>MapReduce configuration:</u>

- Open `etc/hadoop/mapred-site.xml` file. This file is used for setting MapReduce in Hadoop.
- The following lines are added:

```
<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>
```

```
  GNU nano 7.2                    etc/hadoop/mapred-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.
-->

<!-- 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/m>
    </property>
</configuration>
File Name to Write: etc/hadoop/mapred-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
```

Set `mapreduce.framework.name=yarn` means to set the framework for MapReduce jobs YARN.

Set `mapreduce.application.classpath` means to include classpath for running MapReduce applications. The necessary libraries are assured to be available.

b. <u>YARN configuration:</u>

- Open etc/hadoop/yarn-site.xml file. This file is used for setting YARN in Hadoop.
- The following lines are added:

```
<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,HADOOP_HOME,PATH,LANG,TZ,HADOOP_MAPRED_HOME</value>

    </property>

</configuration>
```

```
  GNU nano 7.2                    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,CLASSPATH_PREPEND>
    </property>

</configuration>
File Name to Write: etc/hadoop/yarn-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
```

Set `yarn.nodemanager.aux-services=mapreduce-shuffle` means to enable NodeManagers, which is necessary to run MapReduce jobs in YARN.

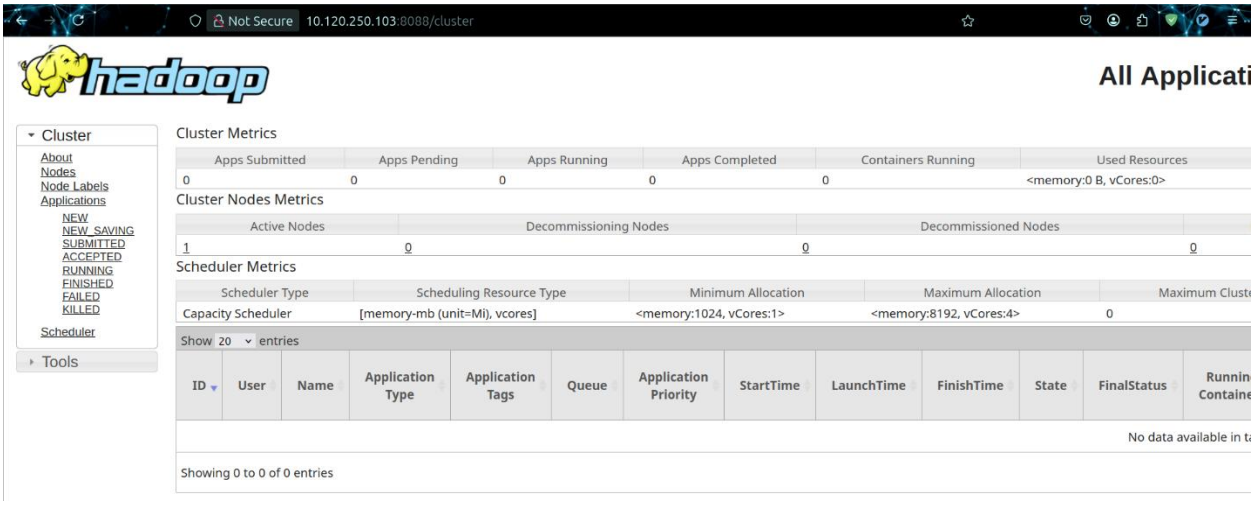
Set `yarn.nodemanager.env-whitelist` means to define environment variables that should be passed from the system to YARN.

c. <u>Start ResouceManager and NodeManager:</u>

- Start YARN.

```
ubuntu@hadoop:~/hadoop$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
```

- For verification: open the web interface for the ResourceManager, by default it is available at http://localhost:8088/.



7. LAB1 requirement:

a. <u>Create a folder in HDFS:</u>

- Create `/hcmus` folder.

```
ubuntu@bigdata:~/hadoop$ hdfs dfs -mkdir /hcmus
```

- Verification by listing all available directories in `/`.

```
ubuntu@bigdata:~/hadoop$ hdfs dfs -ls /
Found 2 items
drwxr-xr-x   - ubuntu supergroup          0 2025-03-20 13:15 /hcmus
drwxr-xr-x   - ubuntu supergroup          0 2025-03-20 10:59 /user
```

b. <u>Create a new user in Ubuntu:</u>

- Create khtn_22120405 and set up the user's password.

```
ubuntu@bigdata:~/hadoop$ sudo useradd -m khtn_22120405
ubuntu@bigdata:~/hadoop$ sudo passwd khtn_22120405
New password:
Retype new password:
passwd: password updated successfully
```

c. <u>Create a sub folder and add a file into it.</u>

- Create `/hcmus/22120405` folder.

```
ubuntu@bigdata:~/hadoop$ hdfs dfs -mkdir /hcmus/22120405
```

- Verification by listing all available directories in `/hcmus`.

```
ubuntu@bigdata:~/hadoop$ hdfs dfs -ls /hcmus
Found 1 items
drwxr-xr-x   - ubuntu supergroup          0 2025-03-20 13:47 /hcmus/22120405
```

- Create a txt file in Ubuntu.

```
ubuntu@bigdata:~/hadoop$ echo "Hello World, HADOOP!" > hello_world.txt
```

- Put the txt file from Ubuntu to HDFS.

```
ubuntu@bigdata:~/hadoop$ hdfs dfs -put hello_world.txt /hcmus/22120405
ubuntu@bigdata:~/hadoop$ hdfs dfs -ls /hcmus/22120405
Found 1 items
-rw-r--r--   1 ubuntu supergroup         21 2025-03-20 13:56 /hcmus/22120405/hello_world.txt
```

d. <u>Change permission and owner:</u>

- Change permission of all files in `/hcmus/22120405`, including the directory, to 744.

```
ubuntu@bigdata:~$ hdfs dfs -chmod -R 744 /hcmus/22120405
```

- Change owner of all files in `/hcmus/22120405`, including the directory, to khtn_22120405.

```
ubuntu@bigdata:~/hadoop$ hdfs dfs -chown -R khtn_22120405 /hcmus/22120405
```

- Verification:

```
ubuntu@bigdata:~/hadoop$ hdfs dfs -ls /hcmus
Found 1 items
drwxr--r--   - khtn_22120405 supergroup          0 2025-03-20 13:56 /hcmus/22120405
```

e. Run the hadoop-test.jar:

- Switch user:

```
ubuntu@hadoop:~/hadoop$ su khtn_22120405
Password:
$
```

- Run hadoop-test.jar:

```
khtn_22120405@bigdata:~$ java -jar hadoop-test.jar 9000 /hcmus/22120405
Trying to read /hcmus/22120405
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/22120405/hello_world.txt
Your student ID: 22120405 (ensure it matches your student ID)
The first method to get MAC address is failed: Could not get network interface
Trying the alternative method
The first method to get MAC address is failed: Could not get network interface
Trying the alternative method
File written at /home/khtn_22120405/22120405_verification.txt
```

- Result:

MAC=52-54-00-61-D4-7B
6ffddd767df79522e236e5e26fad68b005181f5152ed8f50046d000ef674affe

II. Word Count

1. WordCountDriver Class:

This class is the entrypoint for a Hadoop MapReduce job. This sets up the configuration for a job and submit it to the YARN. YARN will distribute the processing codes to the nodes to run MapReduce job.

```
-   public class WordCountDriver {
-       public static void main(String[] args) throws IOException,
    InterruptedException, ClassNotFoundException {
-           // Load default configuration in core-site.xml
-           Configuration conf = new Configuration();
-
-           // Create and set up a new job
-           Job job = Job.getInstance(conf);
-           job.setJarByClass(WordCountDriver.class); // Set the jar file
    that nodes will look up for necessary classes
-           job.setMapperClass(WordCountMapper.class);
-           job.setReducerClass(WordCountReducer.class);
-           job.setSortComparatorClass(WordCountComparator.class);
-
-           // Set up output
-           job.setOutputKeyClass(Text.class);
-           job.setOutputValueClass(IntWritable.class);
-
-           // Set input and output path based on the user's parameters
-           FileInputFormat.addInputPath(job, new Path(args[0]));
-           FileOutputFormat.setOutputPath(job, new Path(args[1]));
-
```

```
-           // Submit the job, the job is now distributed over the nodes to
    run
-           job.submit();
-           System.exit(job.waitForCompletion(true) ? 0 : 1); // Wait for
    the MapReduce task to end
-       }
-   }
-
```

2. **WordCountMapper Class:**

The input file is parsed using the FileInputFormat, which each line will be fed into the `map` method of WordCountMapper class. In particular, this method parses the line with non-alphabetic characters as the delimeters. The first character of every valid parsed words are used as keys and written for the next job.

```
-   public class WordCountMapper extends Mapper<Object, Text, Text,
    IntWritable> {
-       // Create in advance IntWritable to be sent
-       // Initialize as 1
-       private final static IntWritable valueSent = new IntWritable(1);
-
-       private final static Text keySent = new Text();
-
-       // Create a list of accepted starting character
-       private final static List<String> acceptedStartingChar =
    List.of("a", "f", "j", "g", "h", "c", "m", "u", "s");
-
-       @Override
-       protected void map(Object key, Text value, Context context) throws
    IOException, InterruptedException {
-           String line = value.toString(); // Extract to Java built-in's
    String
-
-           // Split the line based on non-alphabetic character
-           String[] words = line.split("[^A-Za-z]+");
-
-           for (String word : words) {
-               if (word.isEmpty())
-                   continue;
-
-               // Get the first character
```

```
-               // Make it lowercase as the task is counting case-
    insensitive line
-               String startingChar =
    String.valueOf(word.charAt(0)).toLowerCase();
-
-               // Eliminate words with unaccepted starting character
-               if (!acceptedStartingChar.contains(startingChar))
-                   continue;
-
-               // Set output for the next steps
-               keySent.set(startingChar);
-               context.write(keySent, valueSent);
-           }
-       }
-   }
-
```

3. WordCountComparator Class:

This class happens in the Shuffle & Sort phase. It creates the orders of data to be fed into the Reduce phase. In this case, the order is “a”, “f”, “j”, “g”, “h”, “c”, “m”, “u”, “s” respectively.

```
-   /// This class makes the keys that Receivers receive follow the order
    as: a, f, j, g, h, c, m, u, s
-   /// Hence, the output follows this order also.
-   public class WordCountComparator extends WritableComparator {
-       protected WordCountComparator() {
-           super(Text.class, true);
-       }
-
-       // Create a map of orders
-       private static final Map<String, Integer> mapOrder = Map.of(
-               "a", 1,
-               "f", 2,
-               "j", 3,
-               "g", 4,
-               "h", 5,
-               "c", 6,
-               "m", 7,
-               "u", 8,
-               "s", 9
```

```
-       );
-
-       @Override
-       public int compare(WritableComparable w1, WritableComparable w2) {
-           Integer order1 = mapOrder.get(w1.toString());
-           Integer order2 = mapOrder.get(w2.toString());
-           return Integer.compare(order1, order2);
-       }
-   }
```

4. WordCountReducer Class:

This class collects all the same keys and accumulates the final result. It then writes out the final result into a file as output indicated in WordCountDriver Class.

```
-   public class WordCountReducer extends Reducer<Text, IntWritable, Text,
    IntWritable> {
-       @Override
-       protected void reduce(Text key, Iterable<IntWritable> values,
    Context context) throws IOException, InterruptedException {
-           // Count number of the same keys
-           int sum = 0;
-           for (IntWritable value : values) {
-               sum += value.get();
-           }
-
-           // Set output for the next steps
-           context.write(key, new IntWritable(sum));
-       }
-   }
-
```

5. Result:

The result is:

a	32921
f	18793
j	4530
g	16002

```
h      20911
c      42817
m      27239
u      24301
s      59567
```

References:

[1]. Hadoop Java Versions - Hadoop - Apache Software Foundation

[2]. Index of /hadoop/common