Contents

Environment 2
Install Hosts 2
Install JDK 5
Set non - password login 8
Hadoop Settings 11
    Set Hadoop configuration files 11
    Test Hadoop 20

Environment

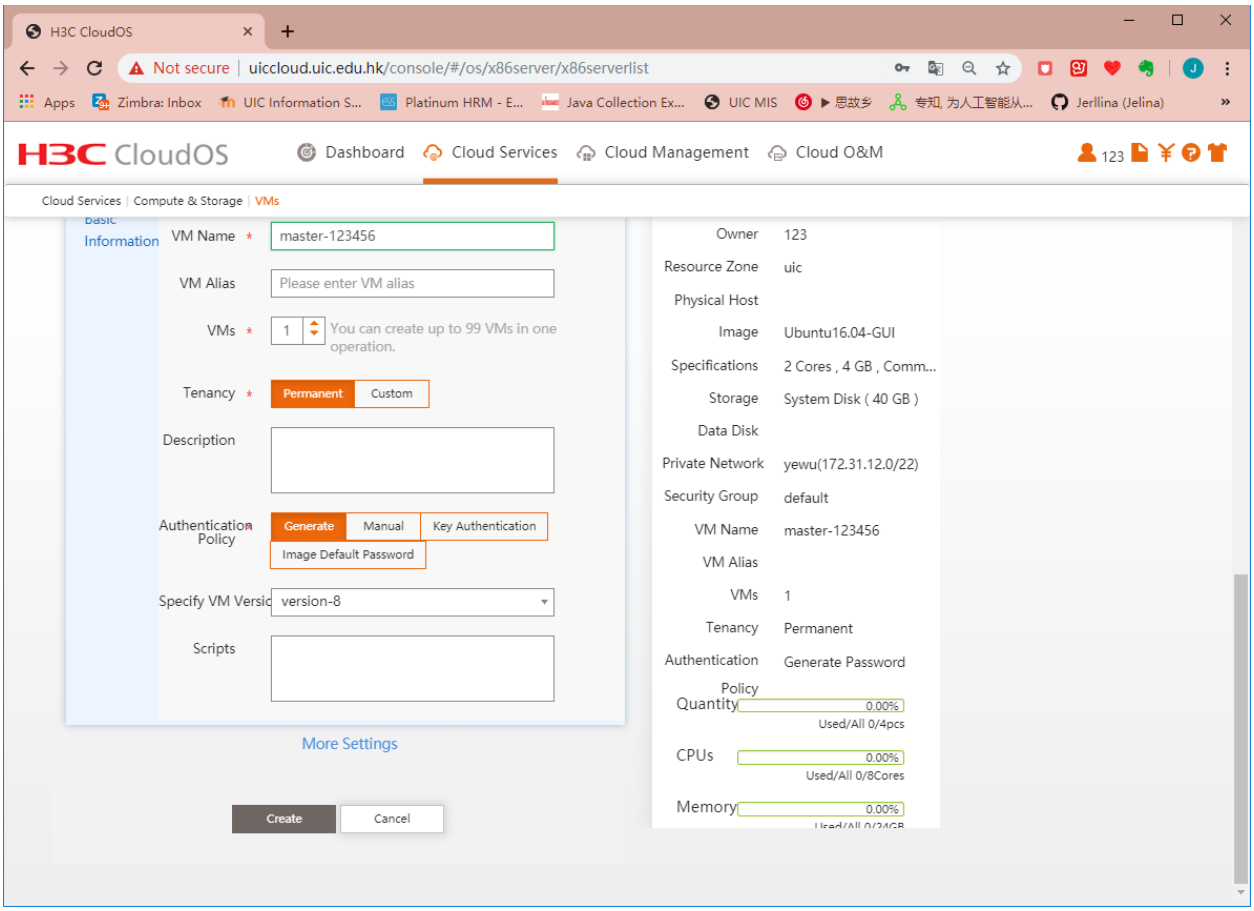
1. Under Cloud OS

2. 3 VM Ubuntu 16.04.1 GUI

3. 1 master and 2 slaves

Install Hosts

Create 3 virtual machines. As we need 1 master and 2 slaves, please name the computers as “master-id”, “slave1-id” and “slave2-id”, such as “master-123456”,” slave1-123456”and” slave2-123456”. The image is “Ubuntu 16.04-GUI”.



☐	master-123456	Running	Ubuntu16.04-GUI	2Cores4G40GCommon	yewu/172.31.13.9
☐	slave1-123456	Running	Ubuntu16.04-GUI	2Cores4G40GCommon	yewu/172.31.13.14
☐	slave2-123456	Running	Ubuntu16.04-GUI	2Cores4G40GCommon	yewu/172.31.13.11

Edit domain name files.

Check **your own IP and host name** on the Cloud OS, please.

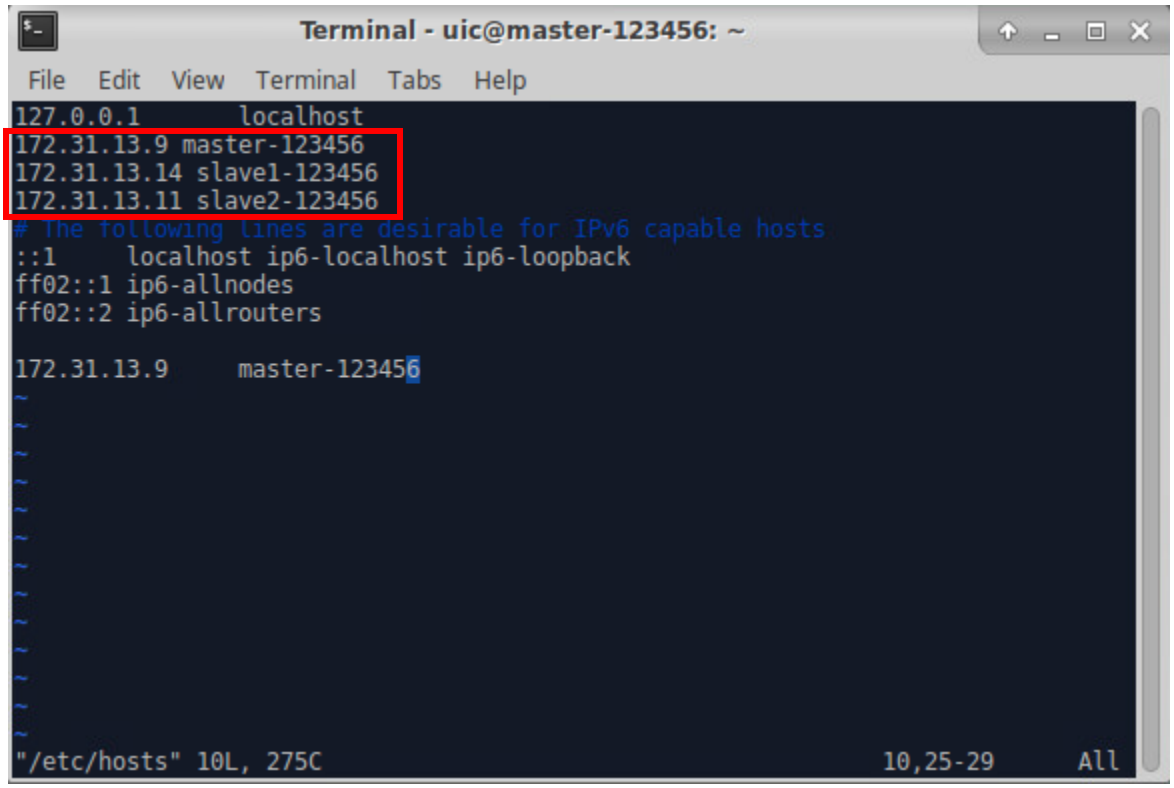
Open Terminal on the master host.

$ sudo vi /etc/hosts

Input password ‘uic’ and press ‘Enter’.

Then press the “**Insert**” button on your keyboard to edit the domain name file.

Modify the 3 IP addresses and **corresponding** names of your hosts into the file. Note that the names set in the hosts file should be **the same** as host names, otherwise unexpected errors will occur.

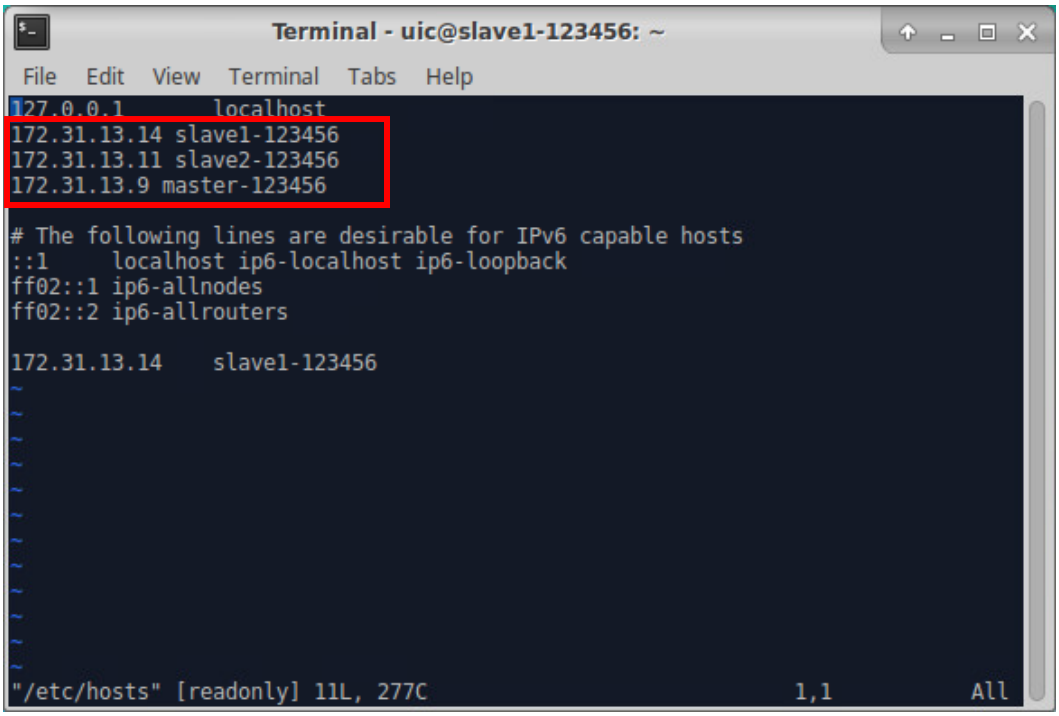


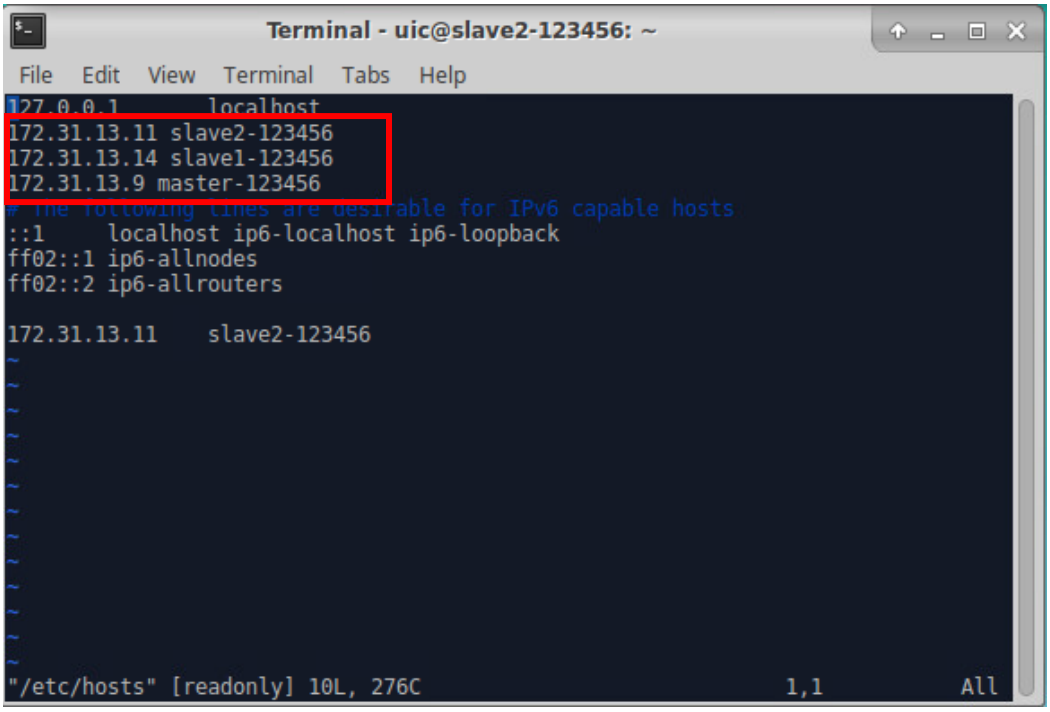
When finished, press on “**Esc**” on your keyboard.

Input **":wq!"** then "**Enter**" to save and exit.

Remember this operation for vim or vi editing. "w" means "write", "q" means "quit", "!" means " obligatory". If you just want to read and exit, just input ":q" then press "Enter".

Then do the same operations on two slave hosts.

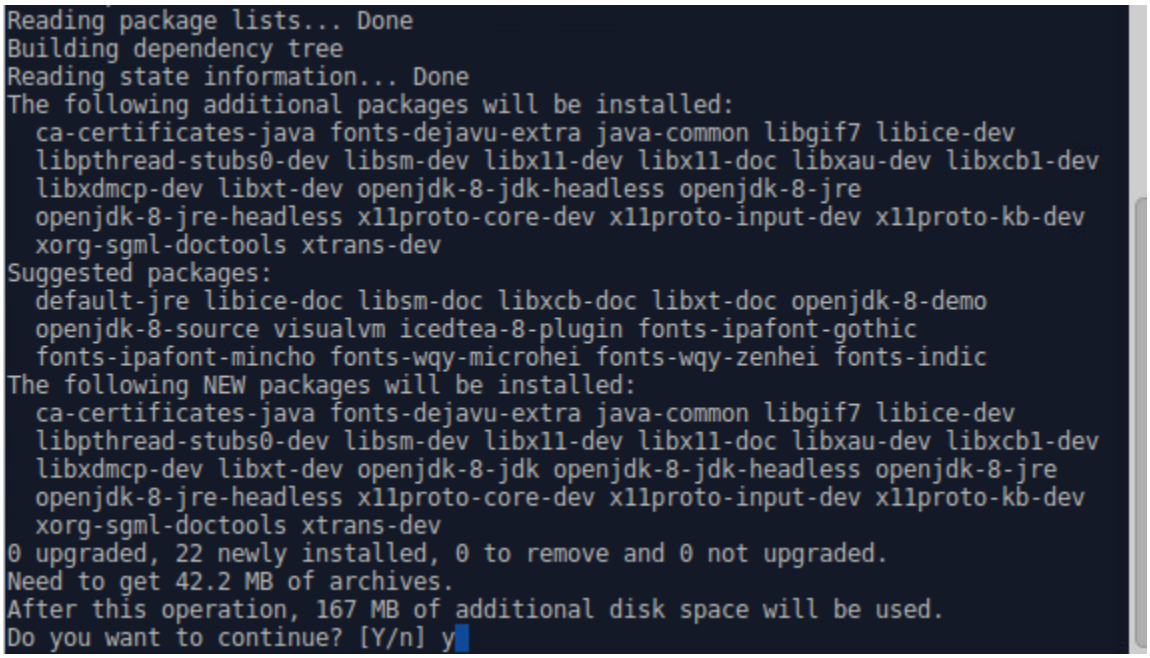




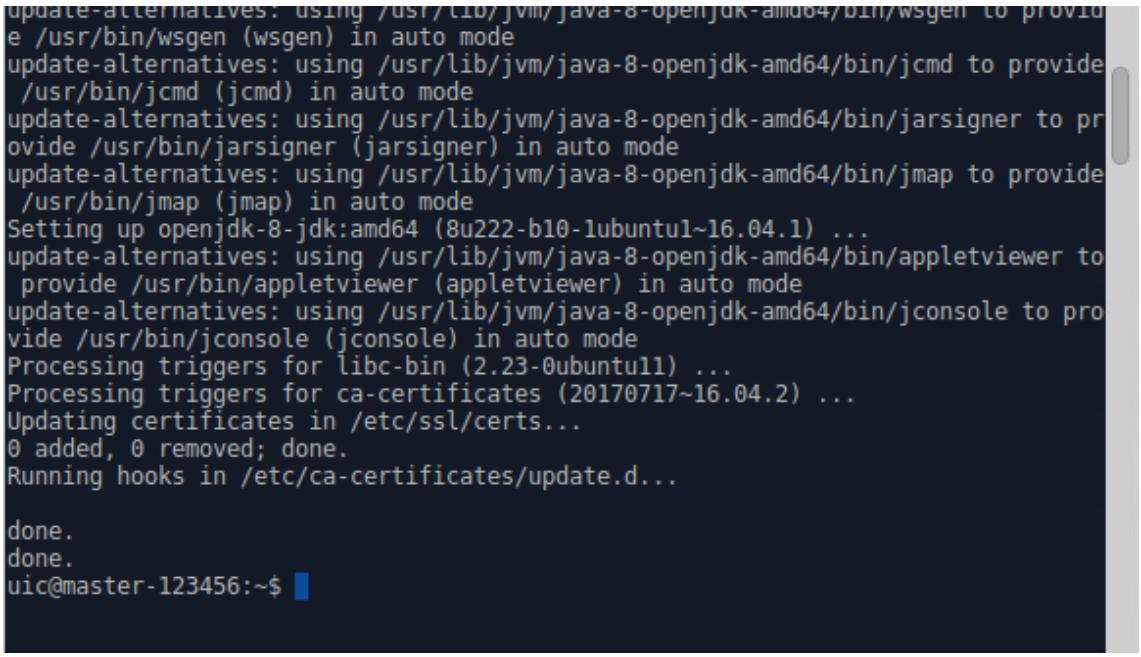
Install JDK

1. Install JDK

$sudo apt install openjdk-8-jdk



Input “Y” and press “Enter”.



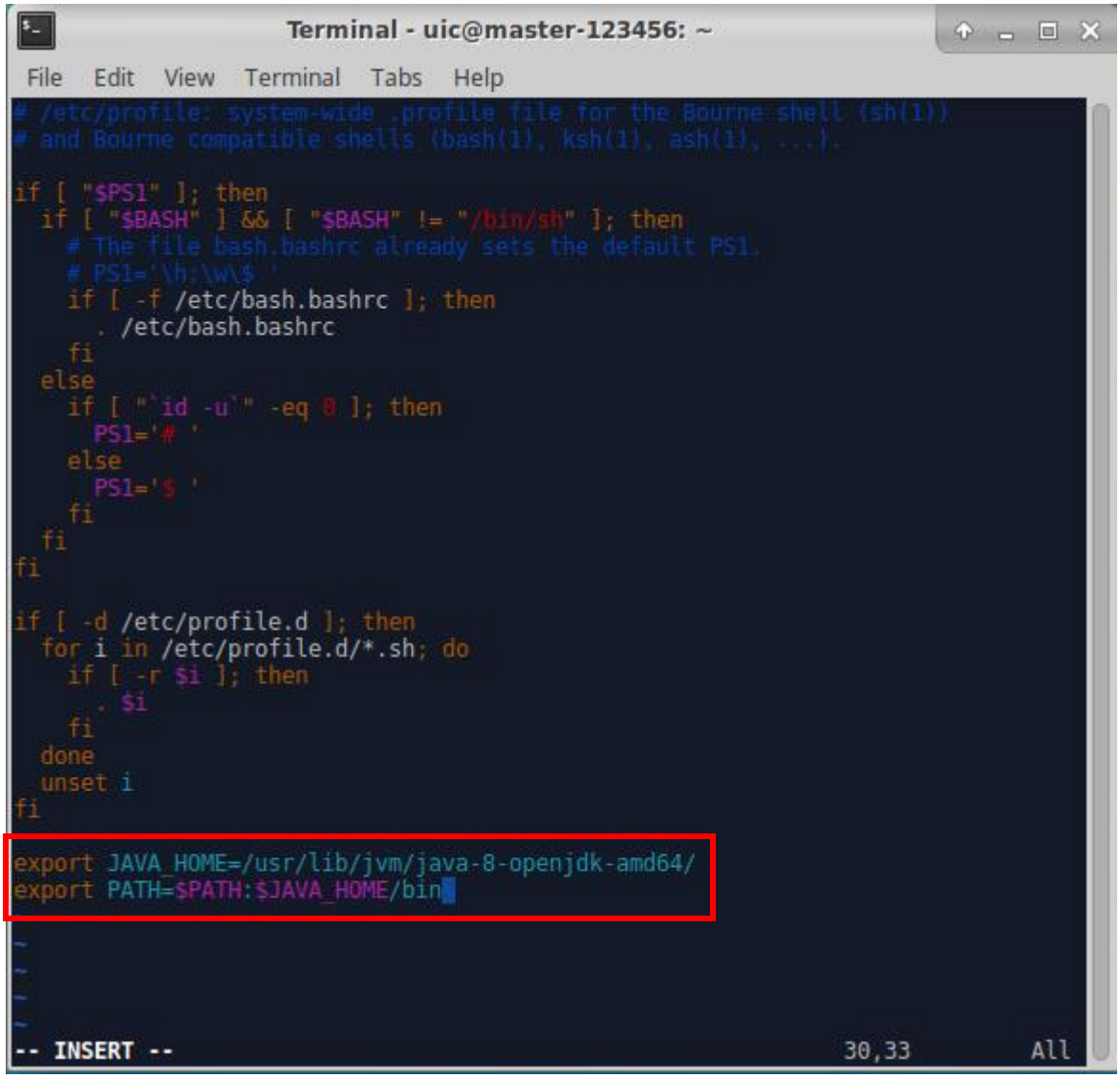
2. Modify the system path file

$ sudo vim /etc/profile

Set the environment variables (operation mentioned above -- Insert, Esc, :wq!, Enter)

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/

export PATH=$PATH:$JAVA_HOME/bin

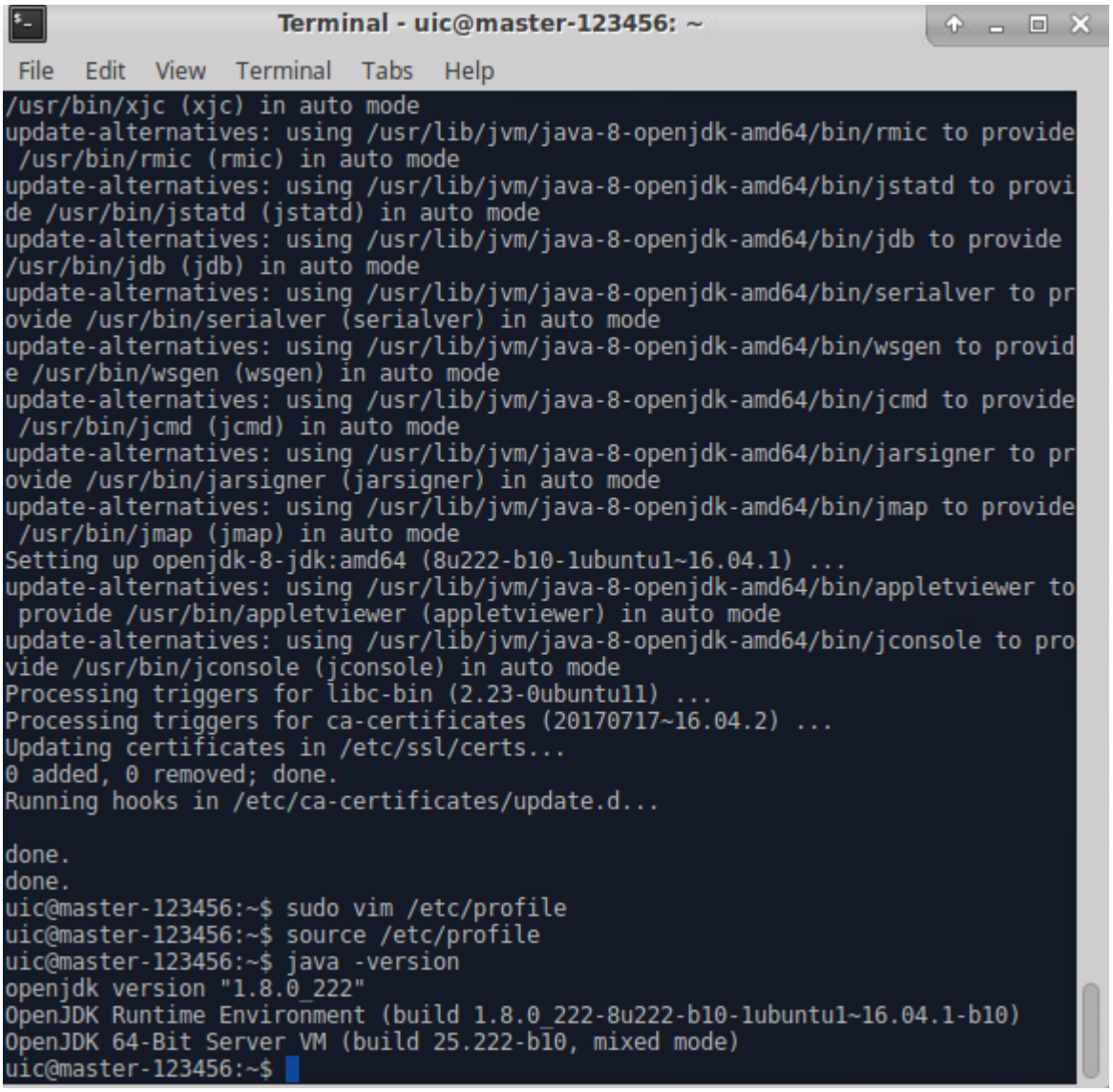


3. Put the environment variable into effect immediately

$ source /etc/profile

4. Verify

$ java -version



It is okay.

Then install JDK on the two slave hosts.

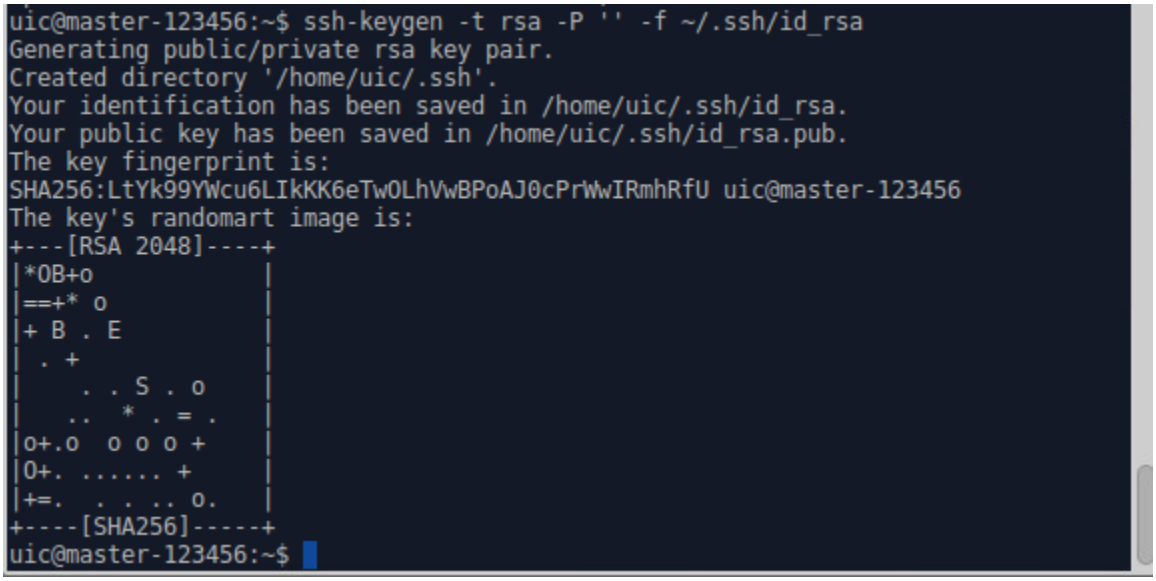
Set non - password login

Why and how to set ssh non-password login:

http://www.linuxidc.com/Linux/2015-03/114709.htm

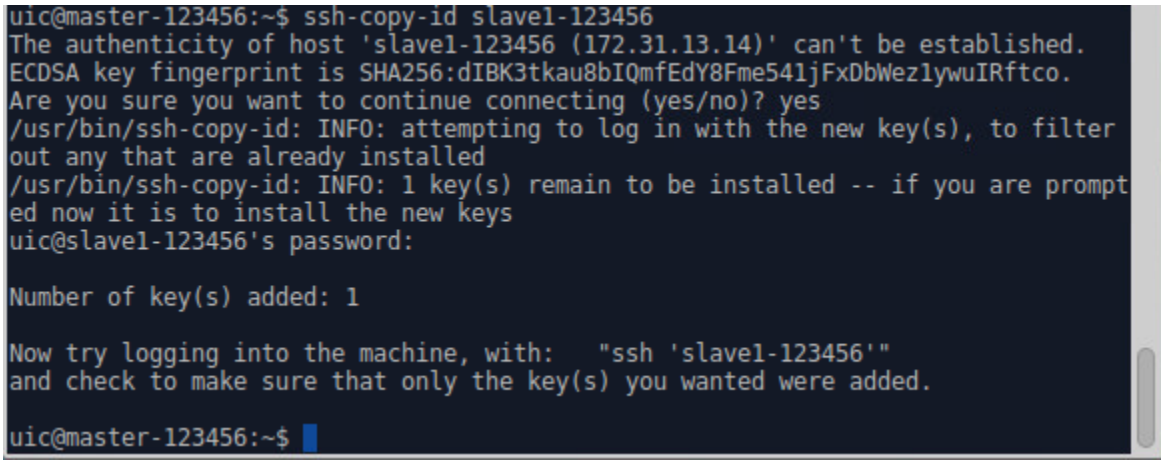
1. Generate the public key of SSH.

$ ssh-keygen –t rsa –P ‘’ –f ~/.ssh/id_rsa

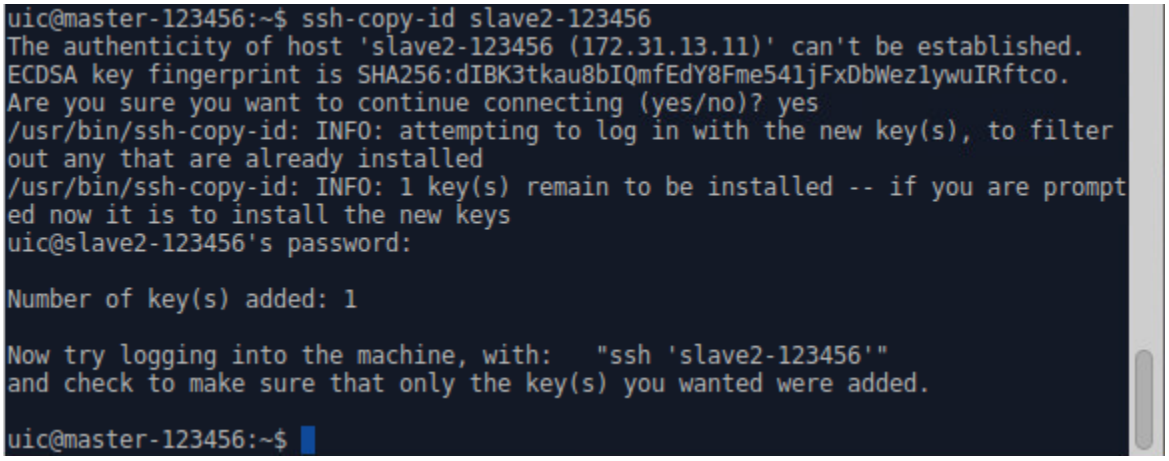


2. Copy and send files

$ ssh-copy-id slave1-123456

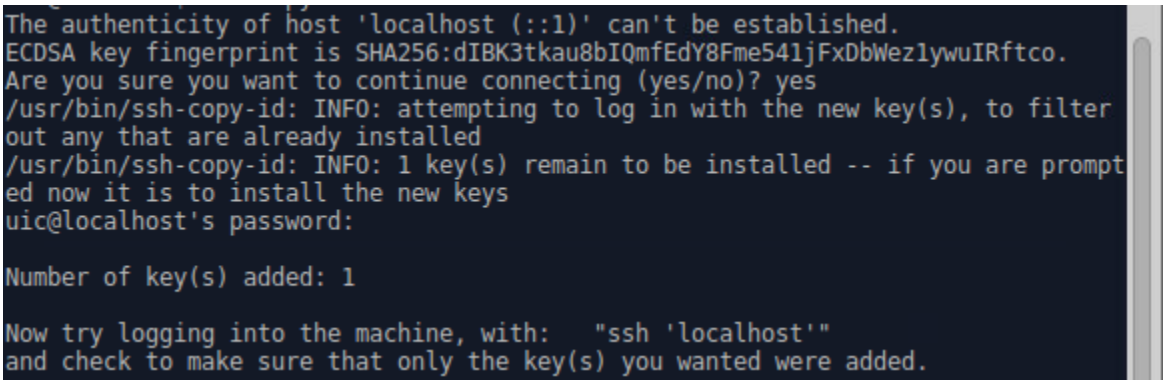


$ ssh-copy-id slave2-123456



Do not forget connect the master self

$ ssh-copy-id localhost



Then do the same operations on the two slave hosts.

On slave1-123456:

$ ssh-keygen –t rsa –P ‘’ –f ~/.ssh/id_rsa

$ ssh-copy-id master-123456

$ ssh-copy-id slave2-123456

On slave2-123456:

$ ssh-keygen –t rsa –P ‘’ –f ~/.ssh/id_rsa

$ ssh-copy-id master-123456

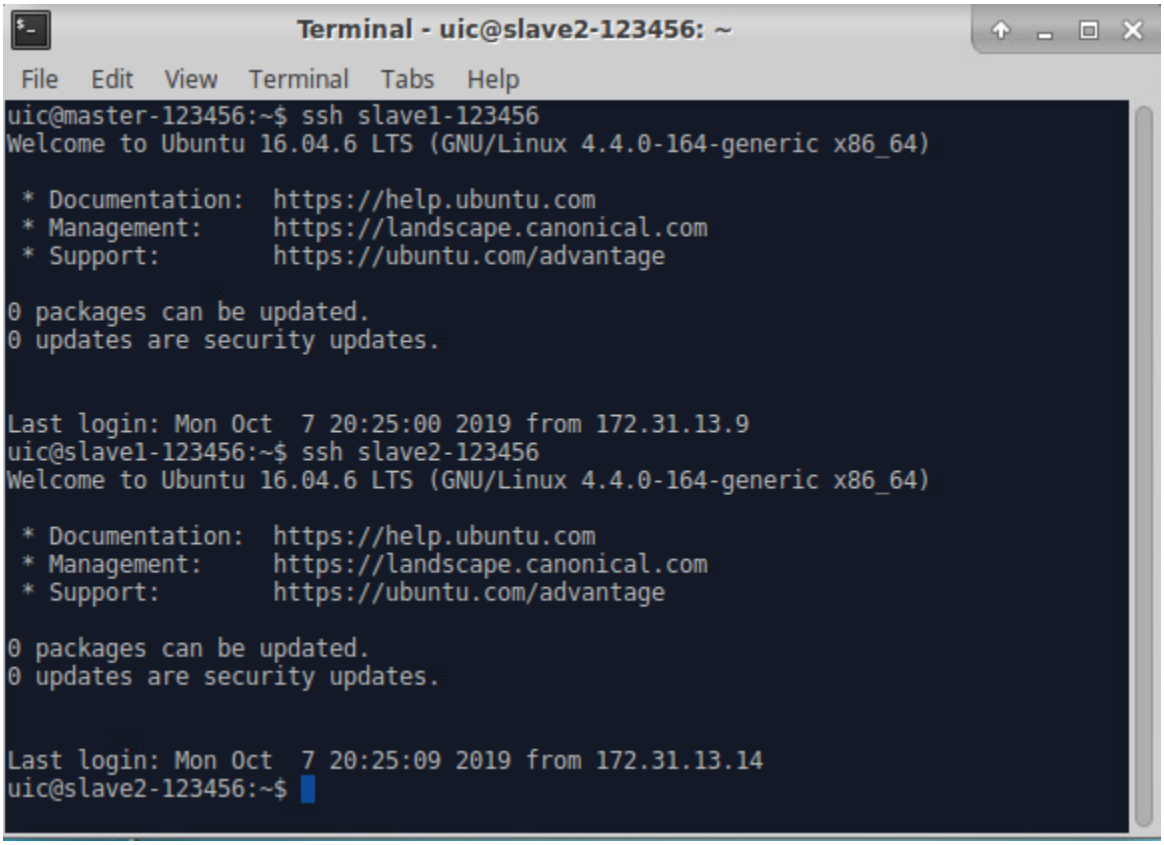
$ ssh-copy-id slave1-123456

3. Test the ssh connection from master to slaves.

On master:

$ ssh slave1-123456

$ ssh slave2-123456

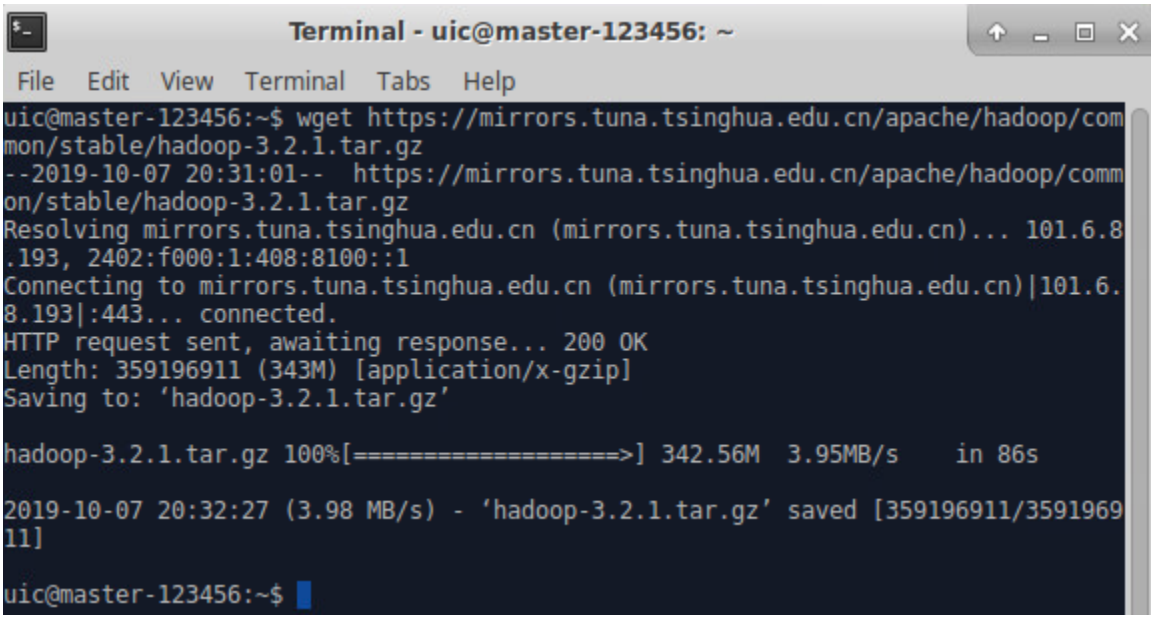


It is okay.

Hadoop Settings

1. Download

$ wget https://mirrors.tuna.tsinghua.edu.cn/apache/hadoop/common/stable/hadoop-3.2.1.tar.gz



2. Unzip the file

We put it in the default directory (/home/uic/hadoop-3.2.1)

$ tar -zxvf hadoop-3.2.1.tar.gz

Set Hadoop configuration files

Set the environment variables

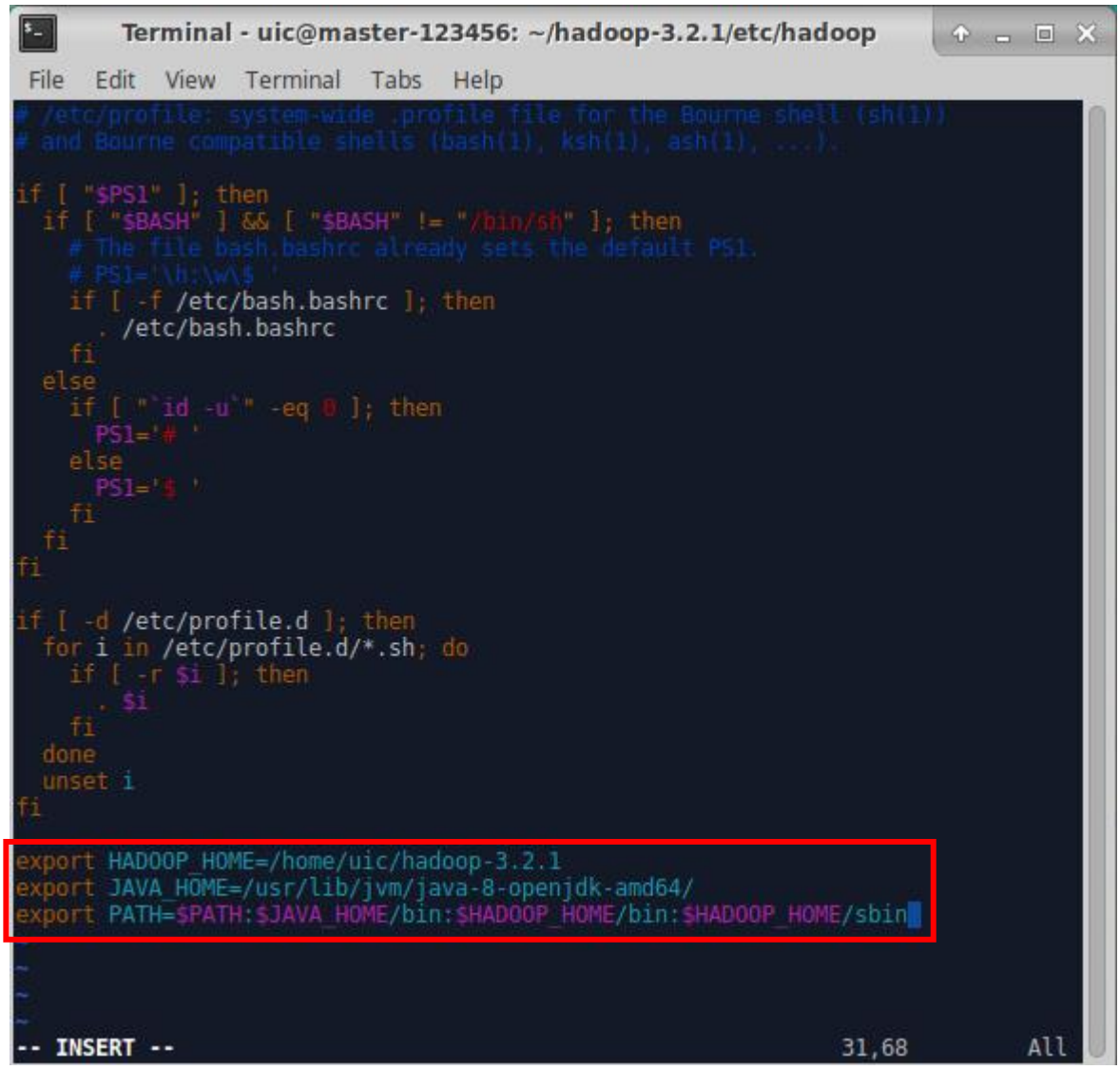
$ cd ~/hadoop-3.2.1/etc/hadoop

$ sudo vim /etc/profile

export HADOOP_HOME=/home/uic/hadoop-3.2.1

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/

export PATH=$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin

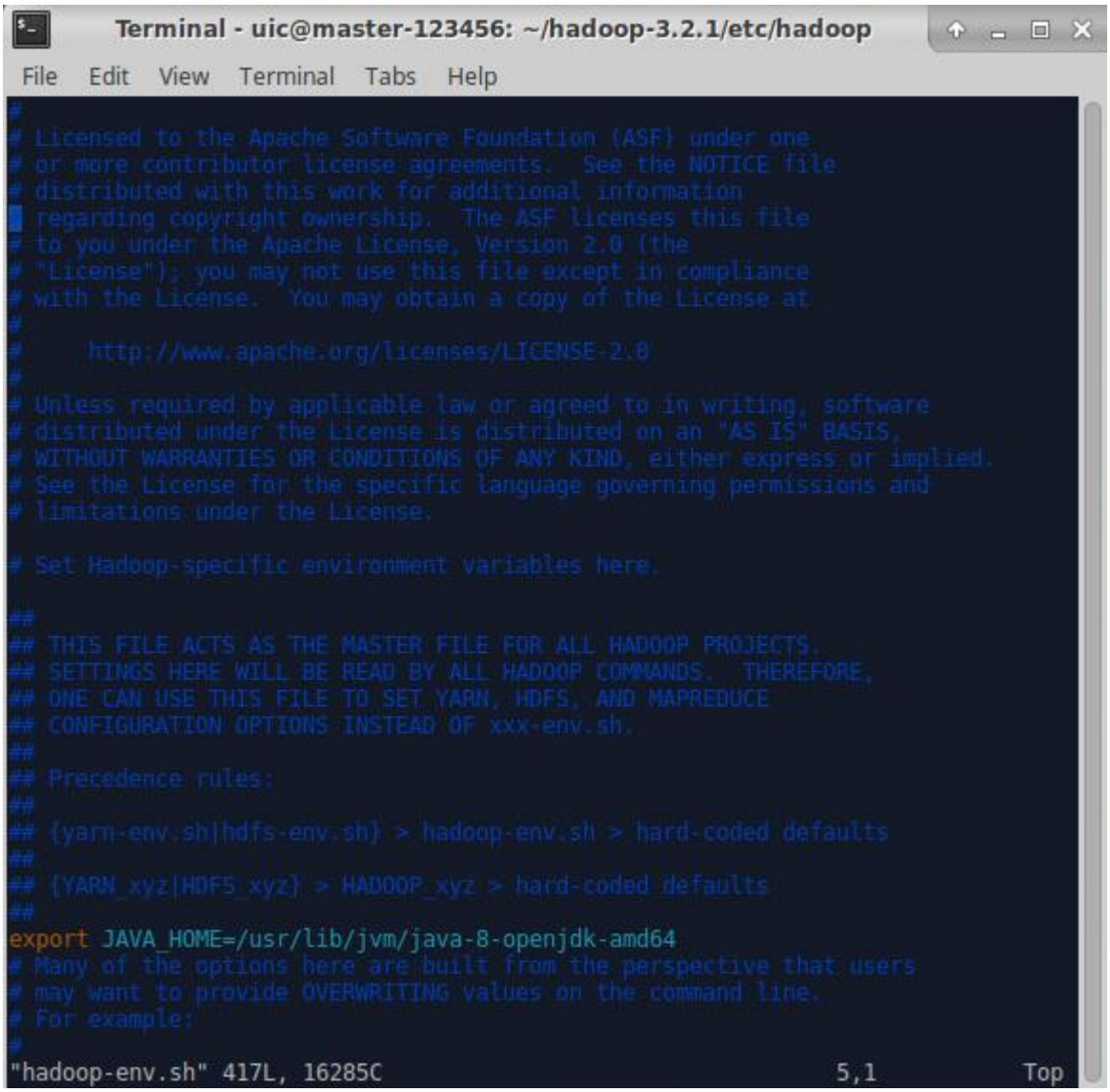


1. Configure hadoop-env.sh and yarn-env.sh

$ cd ~/hadoop-3.2.1/etc/hadoop

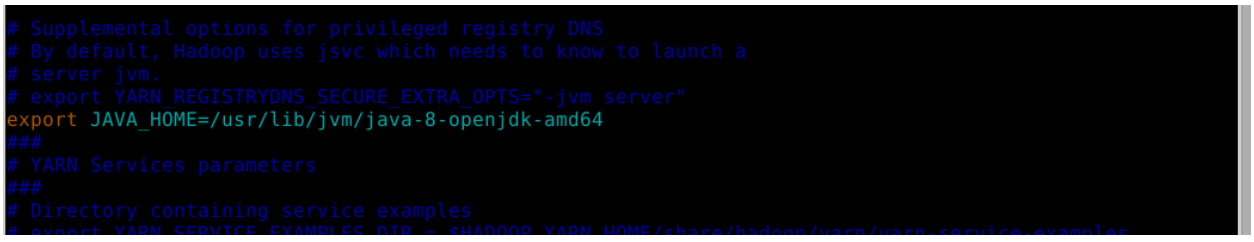
$ sudo vim hadoop-env.sh

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64



$ sudo vim yarn-env.sh

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64



Note: we need create the following folders for the following configuration:

~/hadoop-3.2.1/tmp

~/hadoop-3.2.1/hdfs/name

~/hadoop-3.2.1/hdfs/data

$ cd ~/hadoop-3.2.1

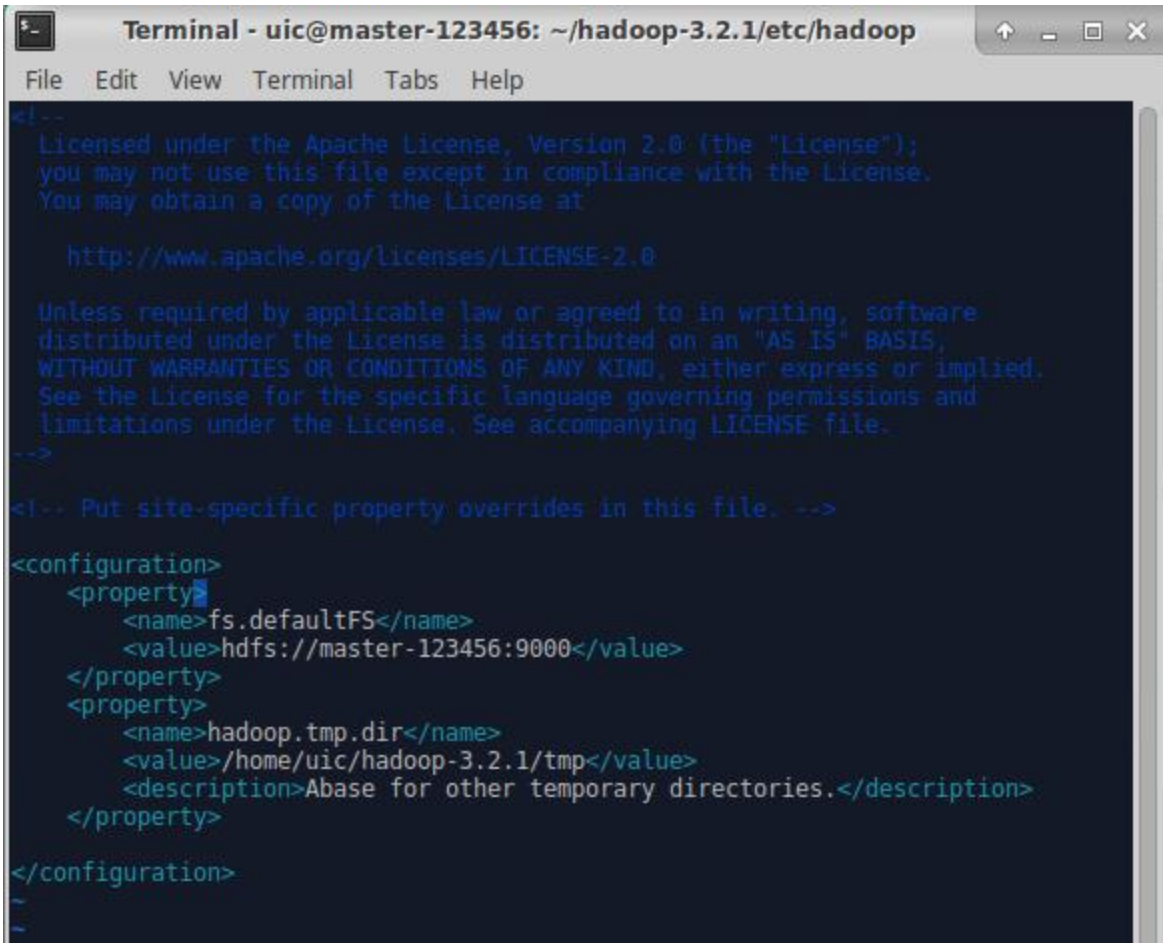
$ mkdir tmp

$ mkdir –p hdfs

2. core-site.xml

$ cd ~/hadoop-3.2.1/etc/hadoop
$ sudo vim core-site.xml

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://master-123456:9000</value>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/home/uic/hadoop-3.2.1/tmp</value>
    <description>Abase for other temporary directories.</description>
  </property>
</configuration>
```



3. mapred-site.xml

$ sudo vim mapred-site.xml

<configuration>

<property>

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

<value>yarn</value>

</property>

<property>

<name>mapred.job.tracker</name>

<value>master-123456:49001</value>

</property>

<property>

```
    <name>mapred.job.dir</name>
    <value>/home/uic/hadoop-3.2.1/var</value>
  </property>
  <property>
    <name>yarn.app.mapreduce.am.env</name>
    <value>HADOOP_MAPRED_HOME=/home/uic/hadoop-3.2.1</value>
  </property>
  <property>
    <name>mapreduce.map.env</name>
    <value>HADOOP_MAPRED_HOME=/home/uic/hadoop-3.2.1</value>
  </property>
  <property>
    <name>mapreduce.reduce.env</name>
    <value>HADOOP_MAPRED_HOME=/home/uic/hadoop-3.2.1</value>
  </property>
</configuration>
```

```
<configuration>
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
    <property>
        <name>mapred.job.tracker</name>
        <value>master-123456:49001</value>
   </property>
   <property>
        <name>mapred.job.dir</name>
        <value>/home/uic/hadoop-3.2.1/var</value>
   </property>

   <property>
        <name>yarn.app.mapreduce.am.env</name>
        <value>HADOOP_MAPRED_HOME=/home/uic/hadoop-3.2.1</value>
   </property>


   <property>
        <name>mapreduce.map.env</name>
        <value>HADOOP_MAPRED_HOME=/home/uic/hadoop-3.2.1</value>
   </property>

   <property>
        <name>mapreduce.reduce.env</name>
        <value>HADOOP_MAPRED_HOME=/home/uic/hadoop-3.2.1</value>
   </property>
</configuration>
```

4. hdfs-site.xml

$ sudo vim hdfs-site.xml

<configuration>

  <property>

    <name>dfs.replication</name>

    <value>2</value>

  </property>

  <property>

    <name>dfs.namenode.name.dir</name>

    <value>file:/home/uic/hadoop-3.2.1/hdfs/name</value>

  </property>

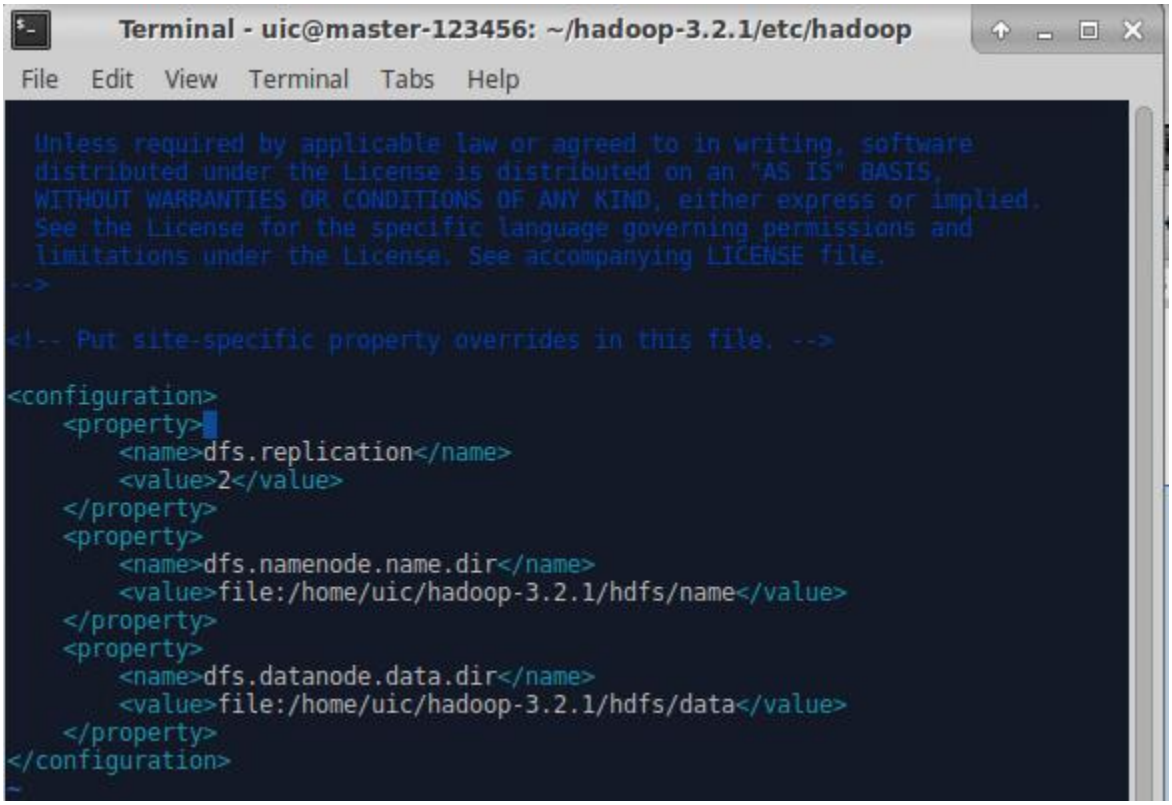
  <property>

    <name>dfs.datanode.data.dir</name>

<value>file:/home/uic/hadoop-3.2.1/hdfs/data</value>

</property>

</configuration>



5. yarn-site.xml

$ sudo vim yarn-site.xml

<configuration>

<property>

<name>yarn.resourcemanager.hostname</name>

<value>master-123456</value>

</property>

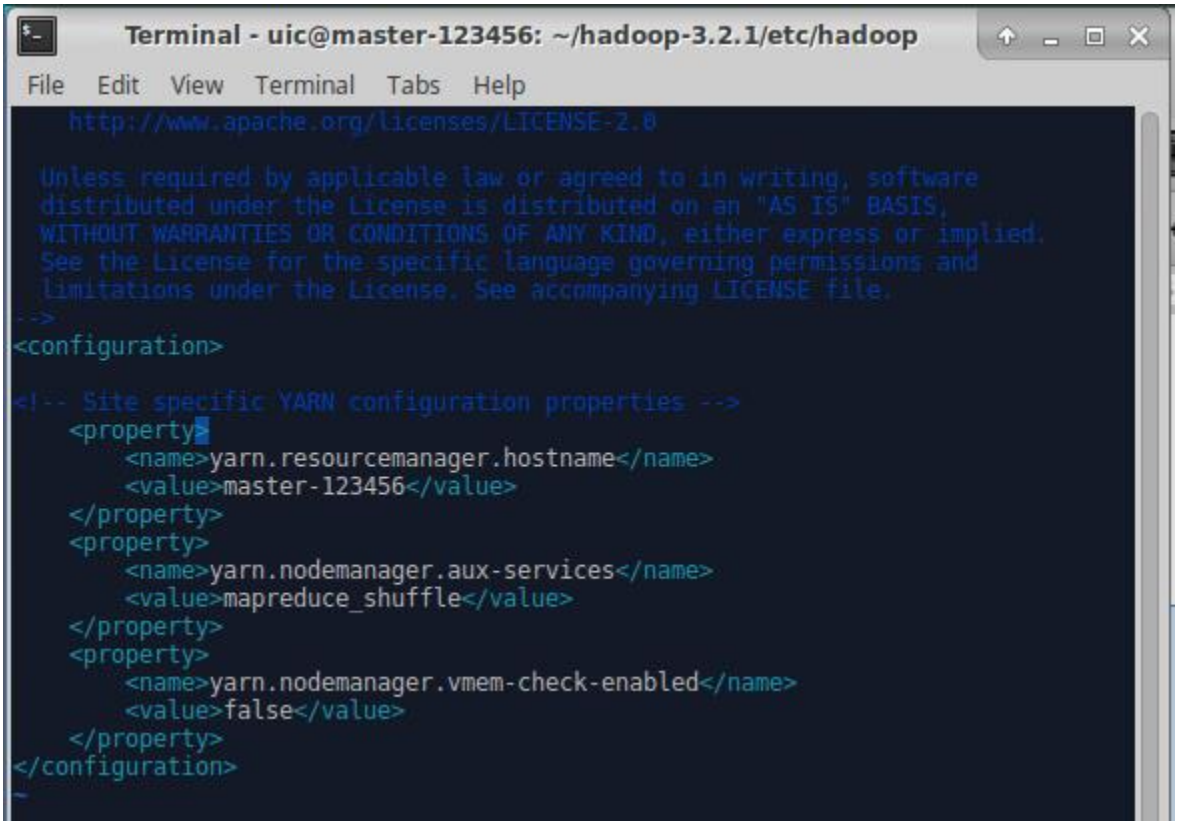
<property>

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

<value>mapreduce_shuffle</value>

</property>

```
  <property>
    <name>yarn.nodemanager.vmem-check-enabled</name>
    <value>false</value>
  </property>
</configuration>
```

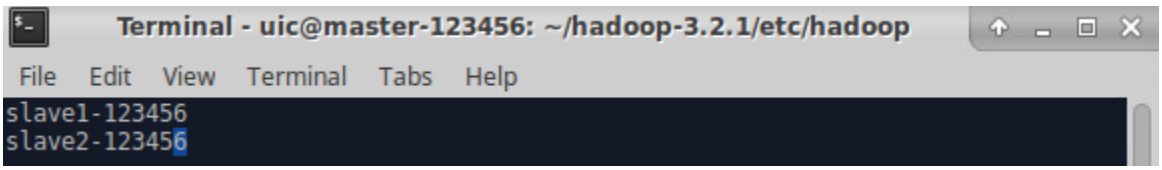


6. Set slave nodesw

$sudo vim workers

slave1-123456

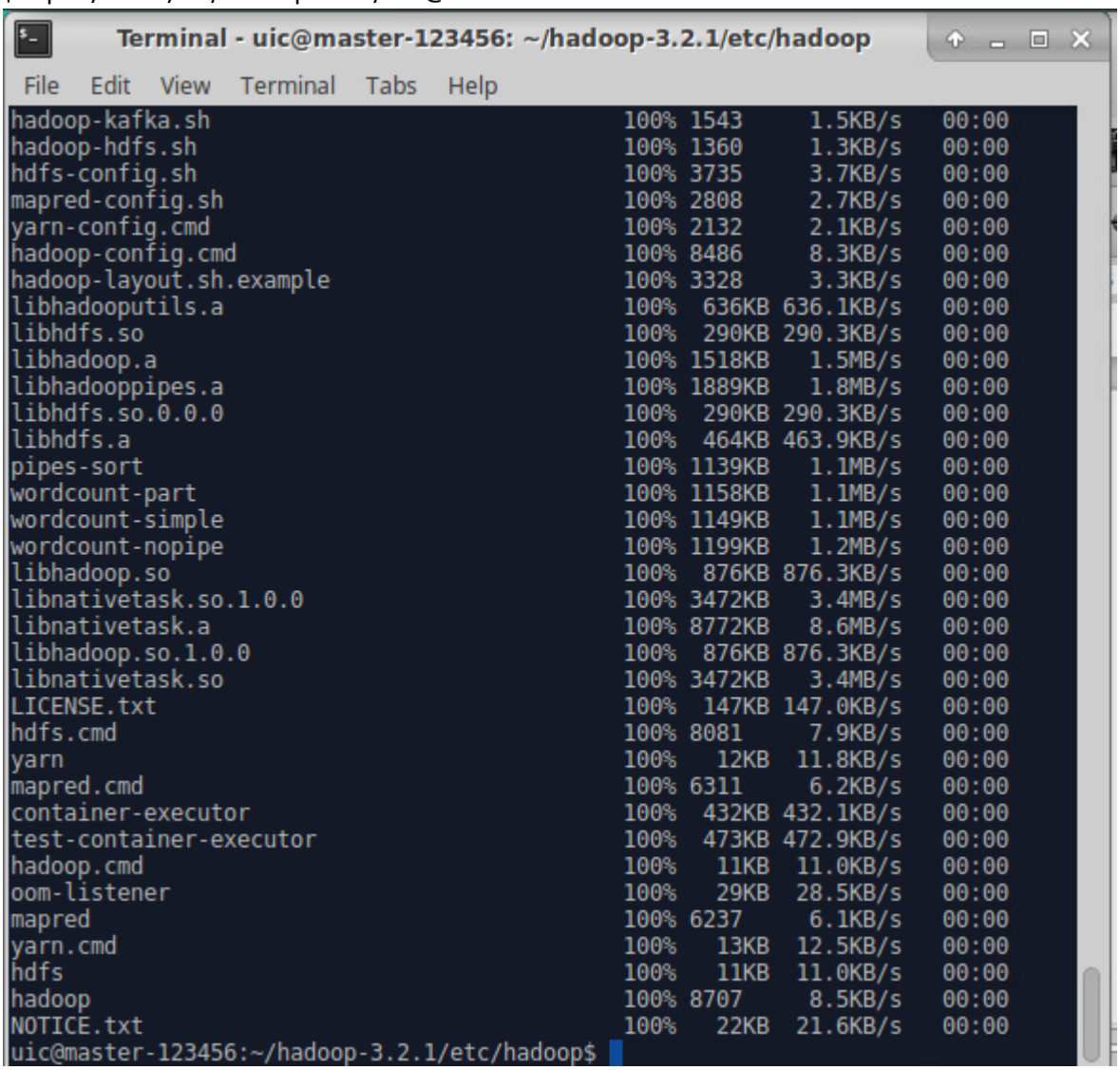
slave2-123456



7. Copy files to slaves

$ scp -r /home/uic/hadoop-3.2.1/ uic@slave1-123456:~

$ scp -r /home/uic/hadoop-3.2.1/ uic@slave2-123456:~

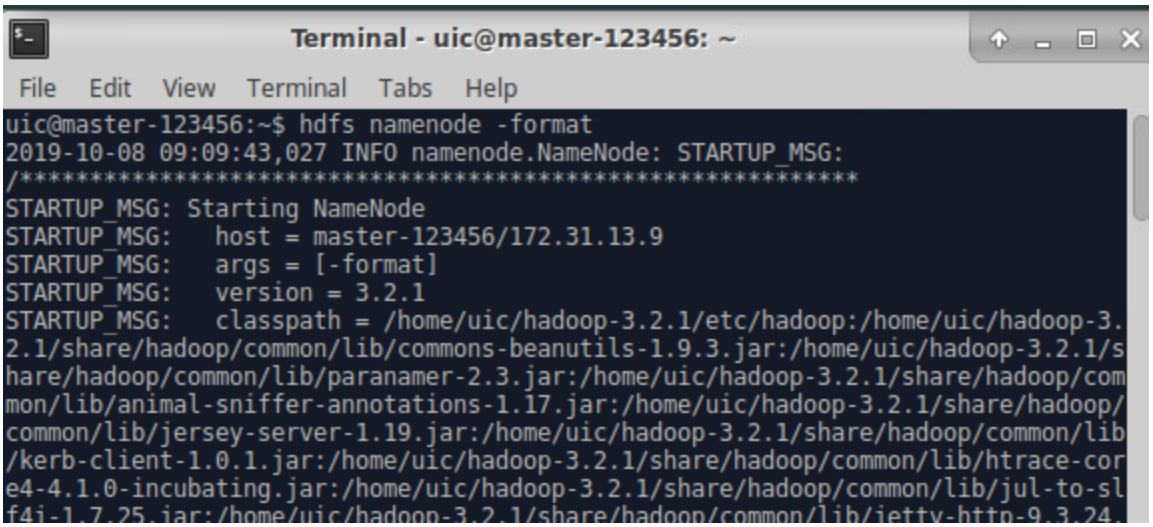


Test Hadoop

1. Format distributed file system

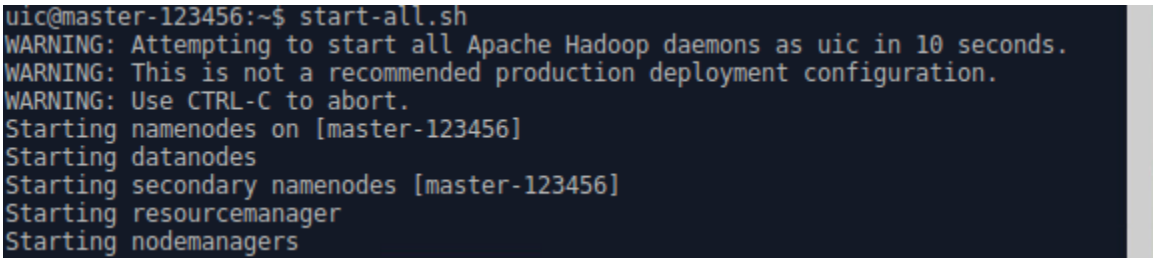
$ cd ~

$ hdfs namenode –format



2. Start

$ start-all.sh

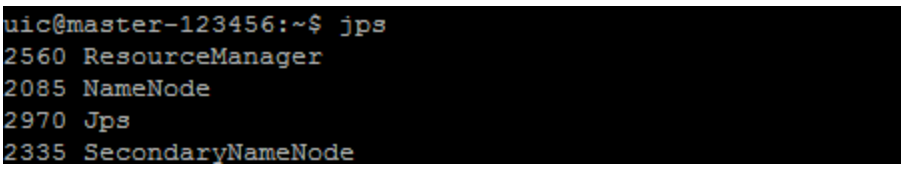


3. Verify if it is successful

On master host, there should be:

NameNode, SecondaryNameNode, Jps and ResourceManager.

$ jps



On slave hosts, there should be: Jps, DataNode and NodeManager.

$ jps

It is successful.

4. Stop

$ stop-all.sh

```
uic@master-123456:~$ stop-all.sh
WARNING: Stopping all Apache Hadoop daemons as uic in 10 seconds.
WARNING: Use CTRL-C to abort.
Stopping namenodes on [master-123456]
Stopping datanodes
Stopping secondary namenodes [master-123456]
Stopping nodemanagers
Stopping resourcemanager
uic@master-123456:~$
```

Note:

If perform "hdfs namenode –format" more than one time, you may encounter the problem that DataNode cannot start.

```
uic@slave1-123456:~$ jps
16529 Jps
16362 NodeManager
```

Try the following command: $ rm –rf /home/uic/hadoop-3.2.1/hdfs/* and then perform the format operation again.

5. test hdfs commands

$ hdfs dfs –mkdir –p /user/uic/hadoop/
Create the directories under HDFS

$ hdfs dfs -put ~/Desktop/test1.txt /user/uic/hadoop/test2.txt
Upload a file to a specified directory and rename it.

$ hdfs dfs –cat /user/uic/hadoop/test2.txt
View the contents of the in directory on HDFS.

$ hdfs dfs -ls /user/uic/hadoop/
List the files under HDFS.

$ hdfs dfs -get /user/uic/hadoop/test2.txt ~/Desktop
Get the file from HDFS
Like **put**, you can manipulate files and directories.

$ hdfs dfs -rm /user/uic/hadoop/test2.txt
Delete the specified file from HDFS.

$ hdfs dfsadmin –report
View basic statistics for HDFS.

6. Web interface

Once the Hadoop cluster is up and running check the web-ui of the components as described below:

To check the namenode information: http://master-123456:9870

To check the resource manager information: http://master-123456:8088