



**《Virtualization and Cloud Computing》**

**Experiment report**

School\_\_\_\_计算机学院\_\_\_\_\_\_\_\_

StuID \_\_\_\_3220200035\_\_\_\_\_

Name \_林飒 KARIM SALMA\_

**Guangdong University of Technology**

Experimental topic Hadoop2.0 deployment experiment

**1. Purpose of the experiment**

Learn to deploy and configure Hadoop in Linux (CentOS) using the traditional way of decompressing packages. Use VMware Workstation to install and configure three virtual machines, modify the software environment for configuring virtual machines, and install and configure the Hadoop software environment.

The purpose of this experiment is to enable students to better understand the deployment of Hadoop using the traditional decompression method in the Linux environment, master the basic operation of Linux system installation and environment configuration, and master the distributed installation and deployment of Hadoop.

1. **Experiment content and requirements**

**Textbook Example 5-2、3、4、5、6．**

1. [Virtual machine installation] Use VMware Workstation to virtualize three CentOS machines cMaster, cSlave0, cSlave1 (the machine name and password can be set to joe), and record the relevant information.

(P177 Example 5-2)

2. [Software Environment Configuration] Change the machine names of the above three CentOS virtual machines to cMaster, cSlave0, and cSlave1, add domain name mapping, close the firewall, install JDK, and record the relevant information.

(P177-179 Examples 5-3, 4)

3. [Install and deploy Hadoop] Install and deploy Hadoop in the virtual machine with the software environment set up above, where cMaster is the master node, cSlave0 and cSlave1 are the slave nodes, and relevant information is recorded.

(P179-181 Example 5-5)

4. Using the cluster you just created, complete the following requirements:

1) Use the Hadoop command to create a new folder "/in" in the cluster;

2) Upload all the files in the file "/home/joe/Hadoop-2.2.0/etc/hadoop/" on cMaster to the folder "/in" of the cluster;

3) Use the sample program WordCount to count the number of occurrences of each word under "/in", and store the result in the "/out" directory;

Record the relevant information.

(P181-182 Examples 5-6)

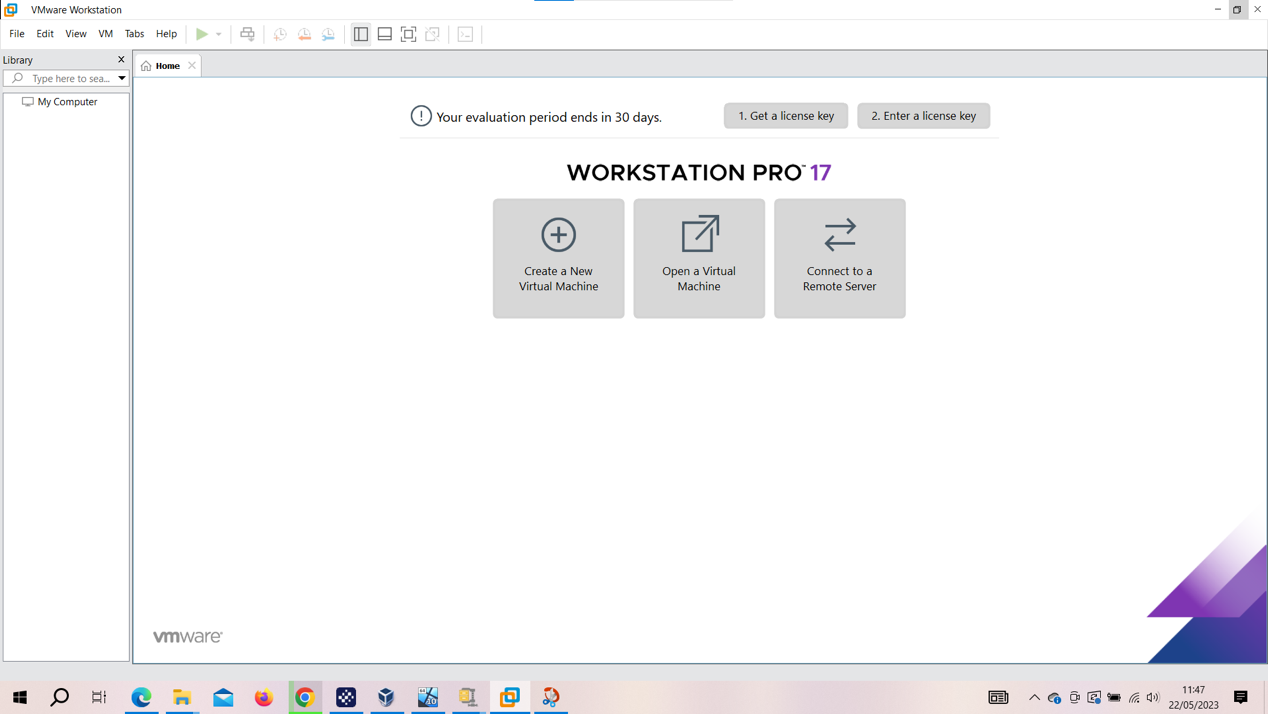
1. **Experimental results**
2. A\

【Example5-2】 Assuming that the configuration of machine A is 4G memory,

dual-core, 100G hard disk, and the system is 64-bit Windows7, it is now required to

virtualize this machine as three CentOS machines *cMaster, cSlave0, and cSlave1.*

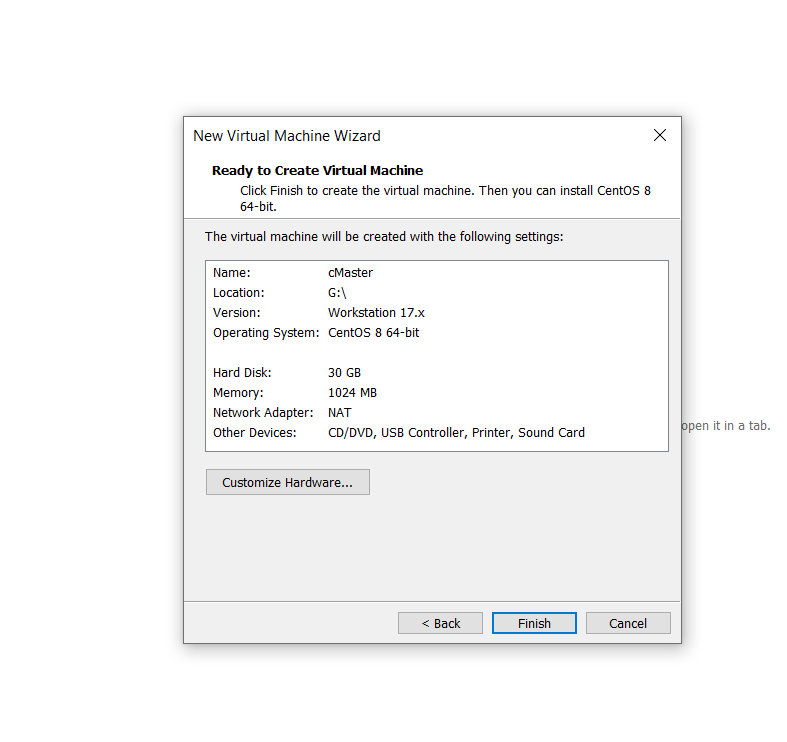
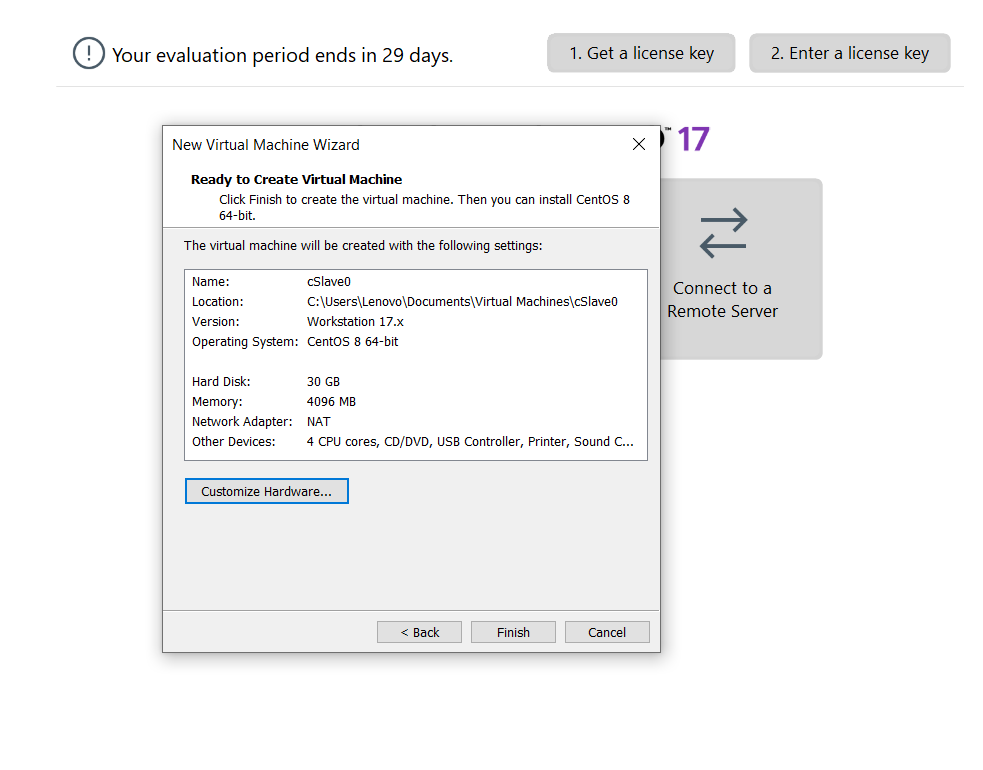
I download VMWare Workstation 17 PRO and I succesfully install it under my windows 10 .

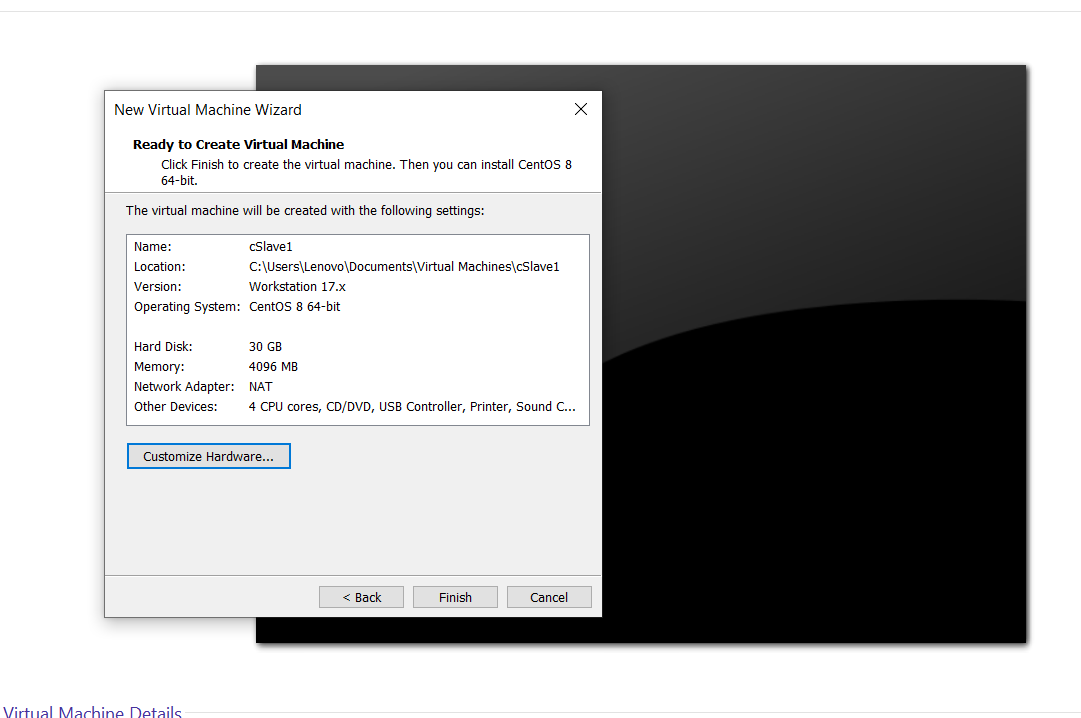


I went to CentOS official website to download the 64-bit

CentOS8.

Step 3: I Create a new CentOS virtual machine for the three machines: I open VMWare Workstation-> I click to New Virtual Wizard->Typical->Installer disc image file(iso)->fill in the username and I use joe like a password->I fill the name of each machines(cMaster, cSlave0 and cSlave1) in machine name >to Finish.





1. A\

【Example5-3】 There is a machine with CentOS system installed, the user name is

joe, and it is required to change this machine to cMaster, add domain name mapping,

close the firewall, and install JDK.

【Example5-4】Suppose there are three machines now, and the CentOS system is

installed. The user’s name of the installation system is joe. It is required to change the

names of the three machines to cMaster, cSlave0, and cSlave1, then add domain name

mapping, close the firewall, and install JDK

【Example5-5】There are three existing machines, and they are all installed

CentOS6.5. Each machine has completed the modification of the machine name, the

addition of domain name mapping, the closure of the firewall and the installation of

JDK according to the above method. Deploy Hadoop with cMaster as the master node

and cSlave0 and cSlave1 as the slave nodes.

【Example5-6】Use the cluster just created and complete the following requirements:

use the Hadoop command to create a new folder "/in" in the cluster; copy all files in

the folder "/home/joe/Hadoop-2.2.0/etc/hadoop" on cMaster Upload it to the cluster

folder "/in"; use the sample program WordCount to count the number of occurrences

of each word under "/in", and store the result in the "/out" directory

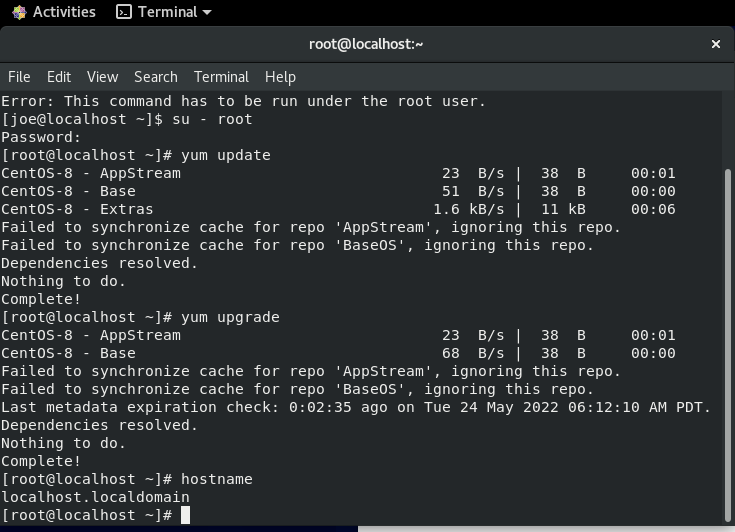
**[SOLUTION(5-3, 5-4, 5-5, 5-6)]**

To modify the machine’s name of each virtuals machines , I need to switch to the root user and edit the file that stores the machine name.

I will also check the hostname in cMaster cSlave0 and cSlave1 systems and rename it accordingly

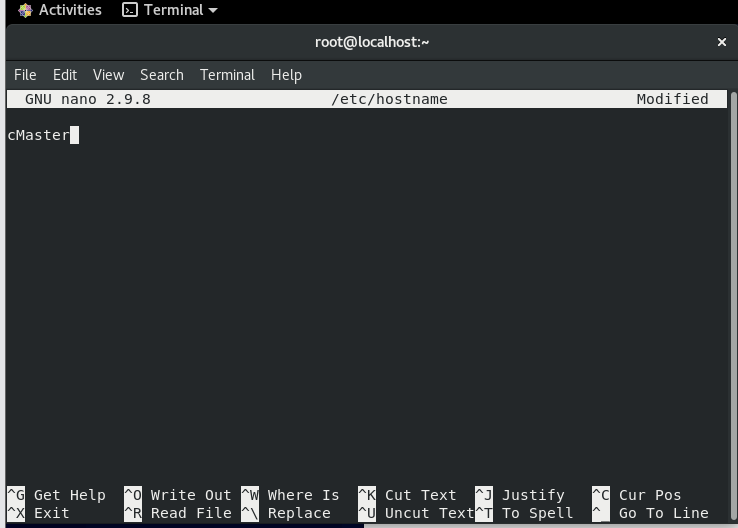
\*\*\* I check hosts name of cMaster and cSlaves(0and1) by using below command\*\*\*

[root@localhost ~]# hostname



**In Master machine**:

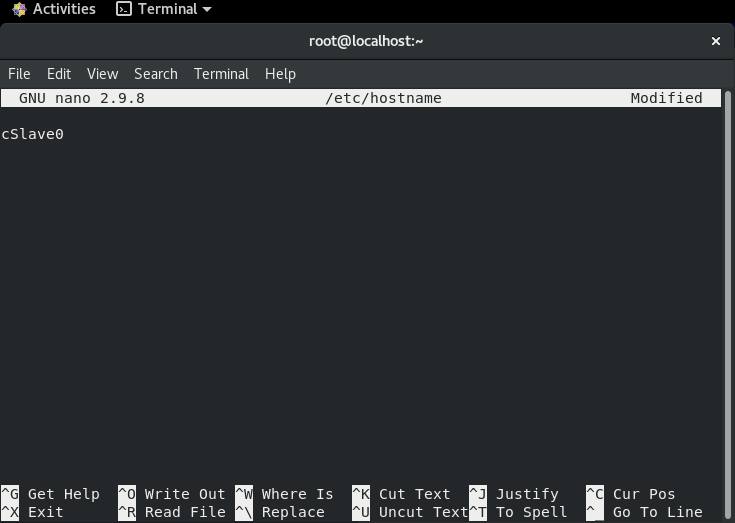
[root@localhost ~]# nano /etc/hostname



**In Slave machines**:

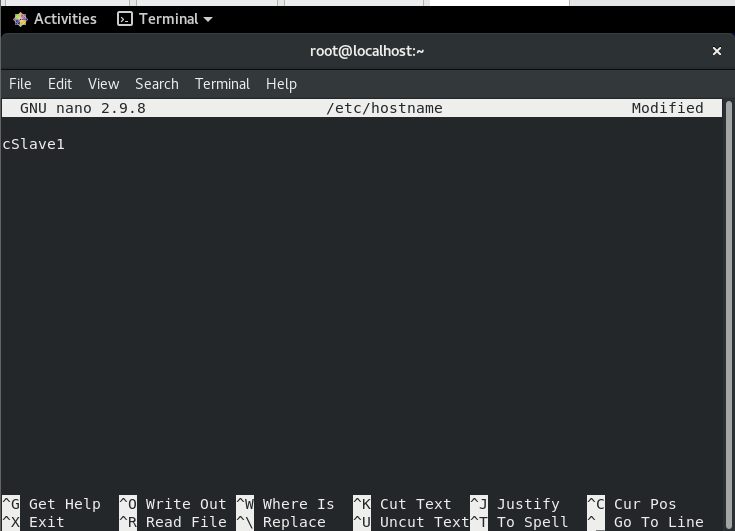
cSlave0 :

[root@localhost ~]# nano /etc/hostname



cSalve1:

[root@localhost ~]# nano /etc/hostname



**Edit /etc/hosts:**

Now lets ssh into cMaster,cSlave0,cSlave1 and change the /etc/hosts file, so that we can use hostname instead of IP everytime we wish to use or ping any of these machines:

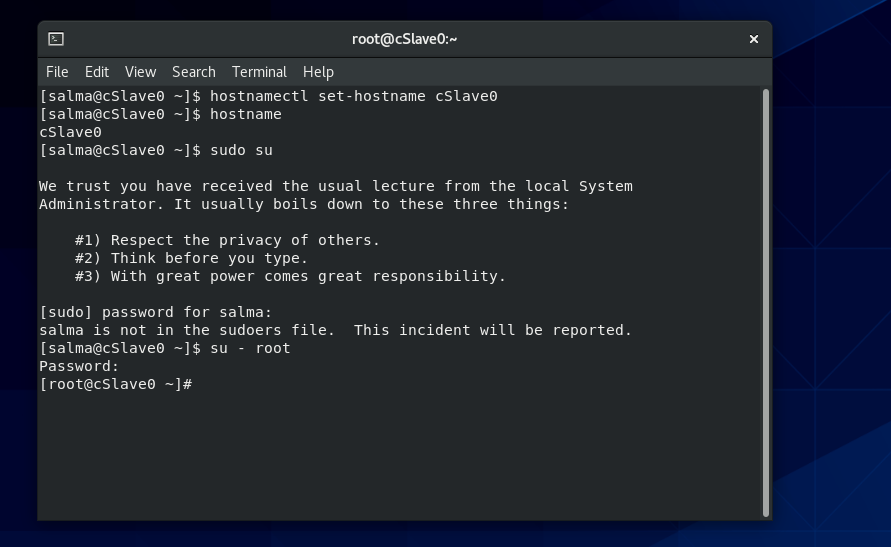
**cMaster**

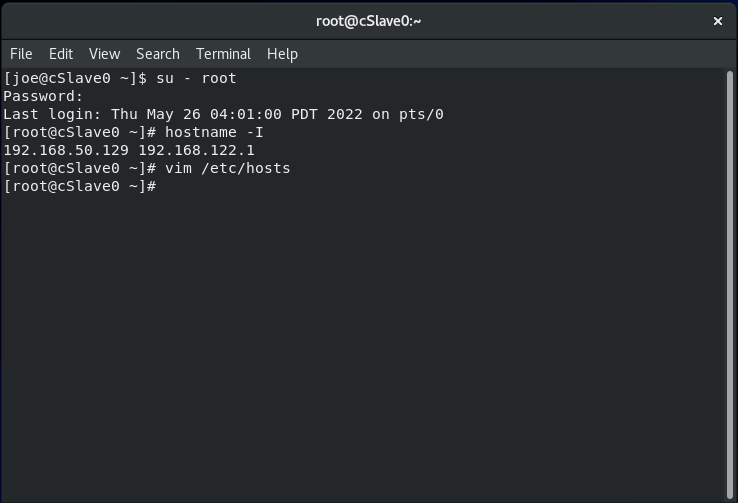
[root@cMaster ~]# vim /etc/hosts

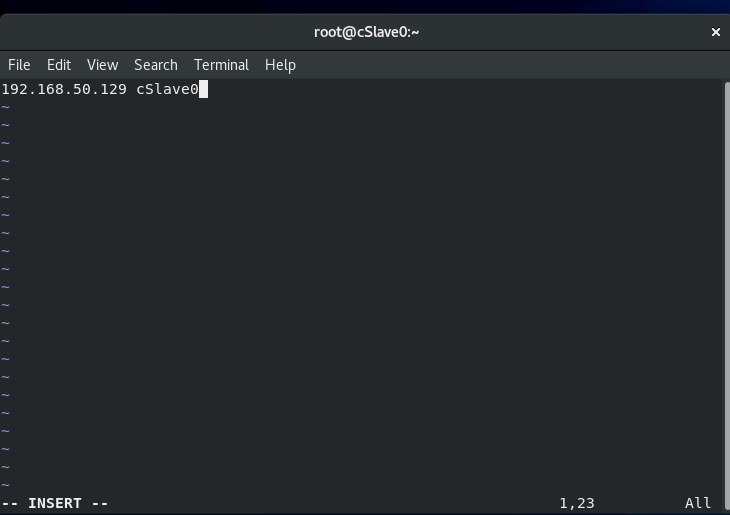


**cSlave0**

[root@cSlave0 ~]# vim /etc/hosts





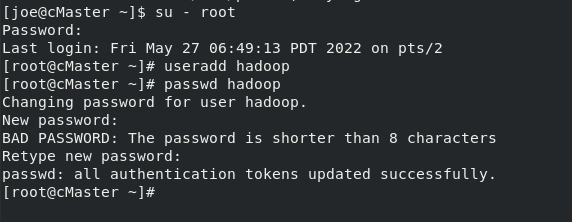


I created a new user using the command:

**cMaster**

[root@cMaster ~]# useradd hadoop

[root@cMaster ~]# passwd hadoop

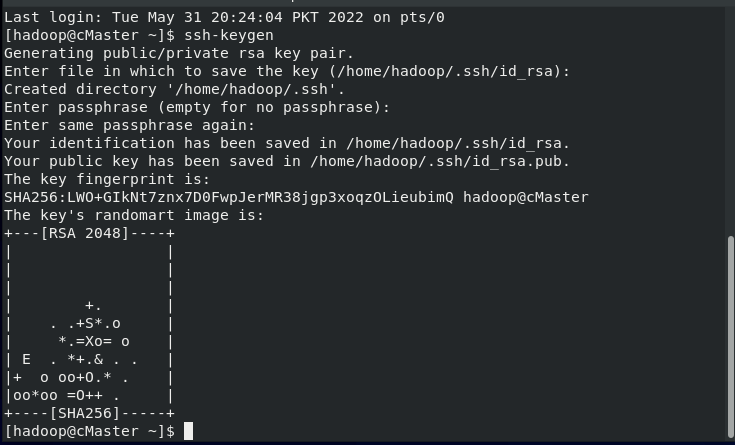


Set Up SSH Key-Pair Authentication

**cMaster**

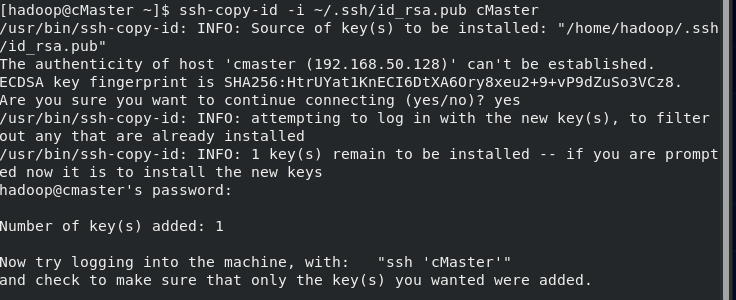
I login to my cMaster virtual machine as the hadoop user, and generate an SSH key as follow:

[hadoop@cMaster ~]$ ssh-keygen

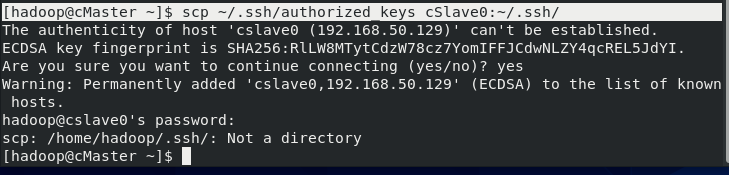


Now I copy id\_rsa.pub contents to authorized\_keys file and then transfer authorized\_keys to remote nodes:

[hadoop@cMaster ~]$ ssh-copy-id -i ~/.ssh/id\_rsa.pub cMaster

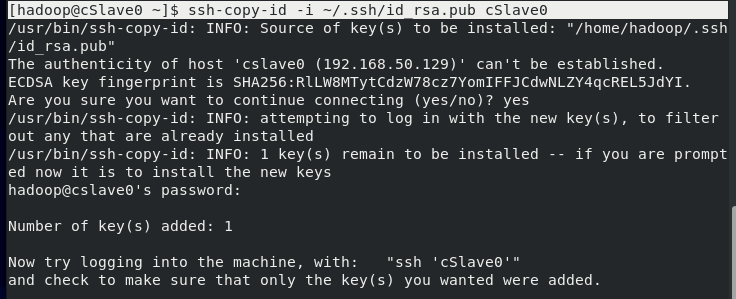


[hadoop@cMaster ~]$ scp ~/.ssh/authorized\_keys cSlave0:~/.ssh/

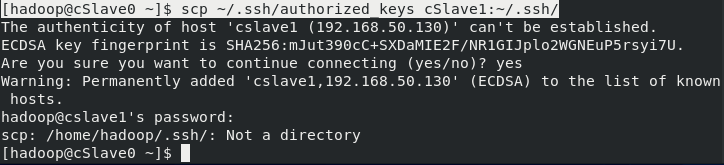


Next, login to cSlave0 as the hadoop user, copy id\_rsa.pub contents to authorized\_keys and then transfer to cSlave1 like below:

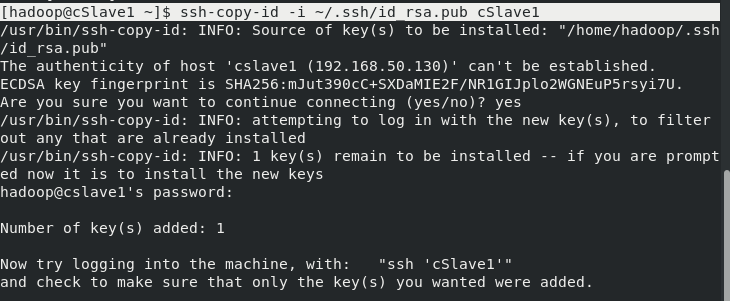
[hadoop@cSlave0 ~]$ ssh-copy-id -i ~/.ssh/id\_rsa.pub cSlave0

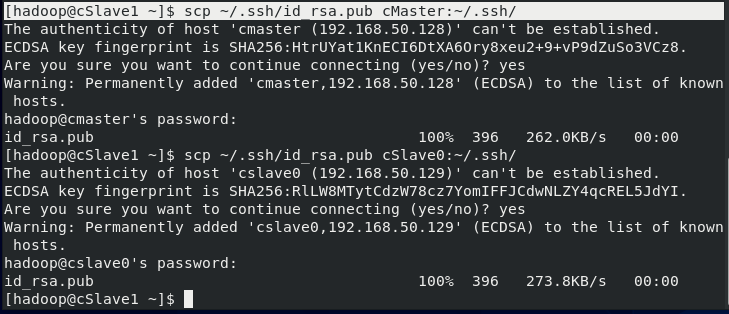


[hadoop@cSlave0 ~]$ scp ~/.ssh/authorized\_keys cSlave1:~/.ssh/



Next, I login to cSlave1, copy id\_rsa.pub contents to authorized\_keys and then transfer to cMaster, and cSlave0 like below:





Install Java

A screenshot of a computer

Description automatically generated

**Setting JAVA**

[root@cMaster ~]# echo "JAVA\_HOME=$(which java)" | sudo tee -a /etc/environment



[root@cSlave0 ~]# echo "JAVA\_HOME=$(which java)" | sudo tee -a /etc/environment

[root@cSlave1 ~]# echo "JAVA\_HOME=$(which java)" | sudo tee -a /etc/environment

[root@cMaster ~]# source /etc/environment

[root@cSlave0 ~]# source /etc/environment

[root@cSlave1 ~]# source /etc/environment

[root@cMaster ~]# echo $JAVA\_HOME

[root@cSlave0 ~]# echo $JAVA\_HOME

[root@cSlave1 ~]# echo $JAVA\_HOME

**Setting Hadoop :**

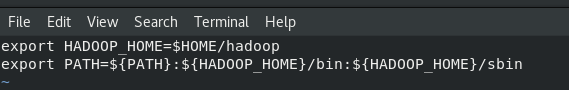
edit /home/hadoop/.bashrc like below\*\*\*

[root@cMaster ~]# vi /home/hadoop/.bashrc

[root@cSlave0 ~]# vi /home/hadoop/.bashrc

[root@cSlave1 ~]# vi /home/hadoop/.bashrc

I add following lines at the end of the file:



I save and close.

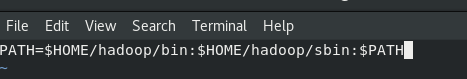
Next, edit /home/hadoop/.bash\_profile:

[root@cMaster ~]# vi ~/.bash\_profile

[root@cSlave0 ~]# vi ~/.bash\_profile

[root@cSlave1 ~]# vi ~/.bash\_profile

I add following line at the end of the file:

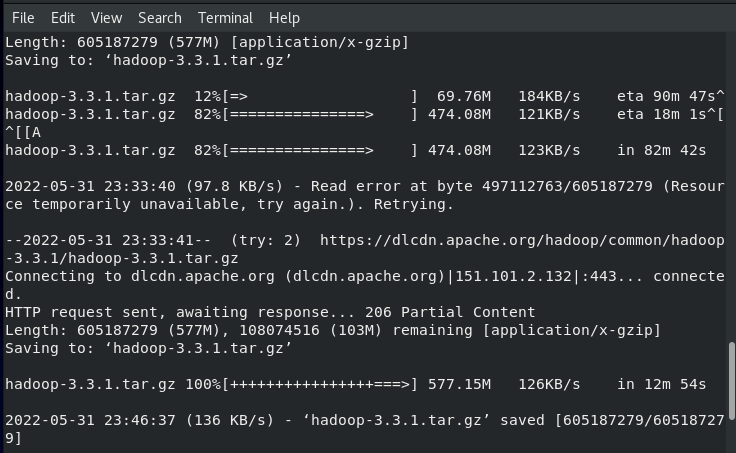


## 

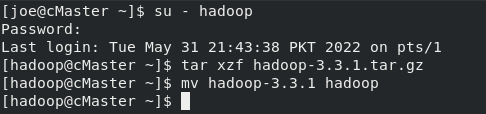
## Download Hadoop

I login to my cMaster as the hadoop user, and I download the latest available version of [Hadoop](http://apache.cs.utah.edu/hadoop/common/current/), and unzip it:

wget <https://dlcdn.apache.org/hadoop/common/hadoop-3.3.1/hadoop-3.3.1.tar.gz>



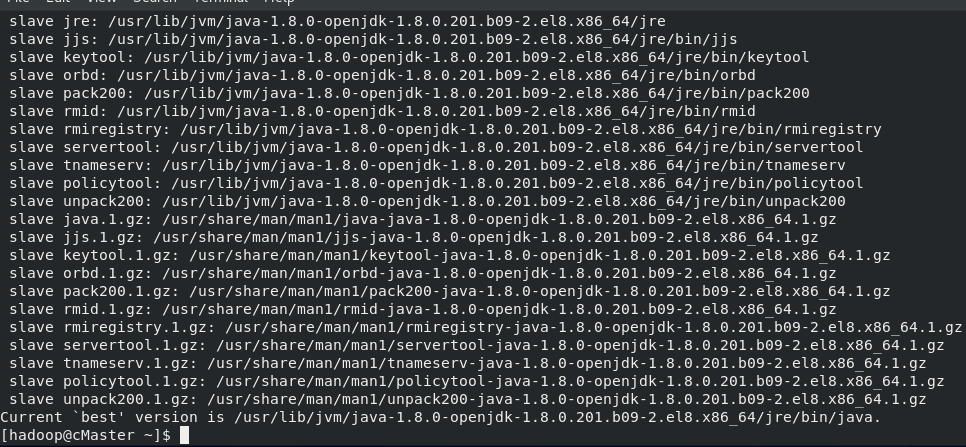
[hadoop@cMaster ~]$ tar xzf hadoop-3.3.1.tar.gz



## Configure Hadoop

At this stage, I will configure hadoop on cMaster first, then replicate the configuration to cSlaves(0 and 1) later.

[hadoop@cMaster ~]$ update-alternatives --display java



I take the value of the (link currently points to) and remove the trailing /bin/java.

/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.201.b09-2.el8.x86\_64/jre/



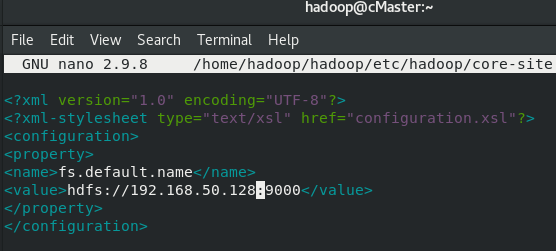
Edit hadoop-env.sh like below:

[hadoop@cMaster ~]$ nano ~/hadoop/etc/hadoop/hadoop-env.sh



Next, I edit core-site.xml file to set the NameNode location to master-node on port 9000:

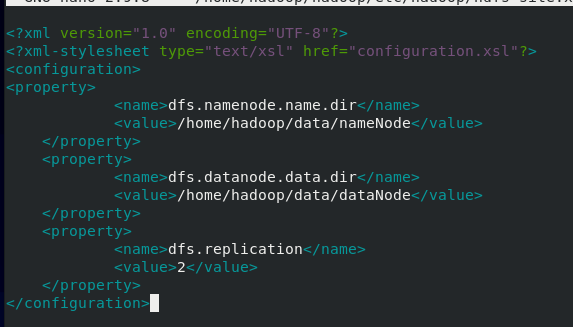
[hadoop@cMaster ~]$ nano ~/hadoop/etc/hadoop/core-site.xml



**Next, edit hdfs-site.conf to resemble the following configuration:**

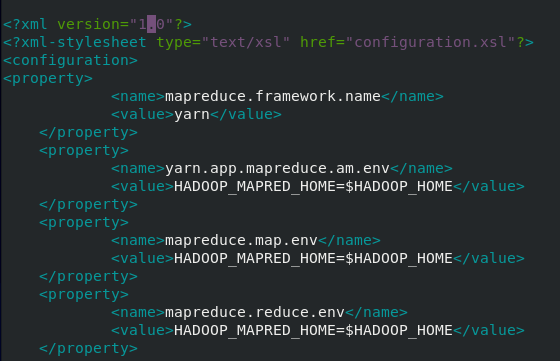
[hadoop@cMaster ~]$ nano ~/hadoop/etc/hadoop/hdfs-site.xml

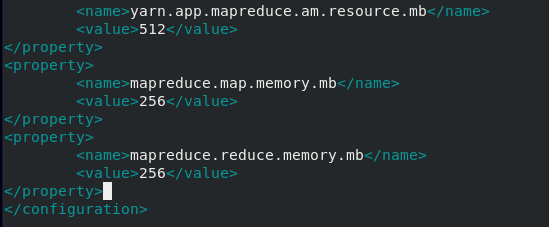
\*\*\*I add following code\*\*\*



**Next, edit the mapred-site.xml file, setting YARN as the default framework for MapReduce operations:**

[hadoop@cMaster ~]$ nano ~/hadoop/etc/hadoop/mapred-site.xml





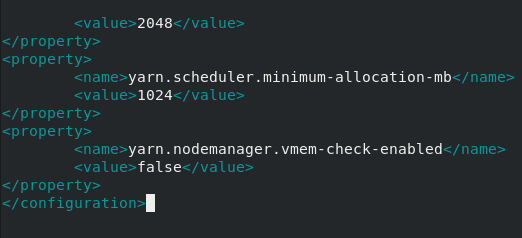
**Next, edit yarn-site.xml, which contains the configuration options for YARN.**

[hadoop@cMaster ~]$ nano ~/hadoop/etc/hadoop/yarn-site.xml

\*\*\*I add below code : with the my cMaster's ip address\*\*\*

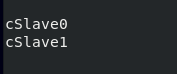
192.168.50.128





**Next, edit workers  file to include both of the cSlaves (cSlave0, cSlave1) :**

[hadoop@cMaster ~]$ nano ~/hadoop/etc/hadoop/workers



**At this stage, I have completed hadoop configuration on cMaster. In the next step I will duplicate hadoop configuration on cSlaves**.

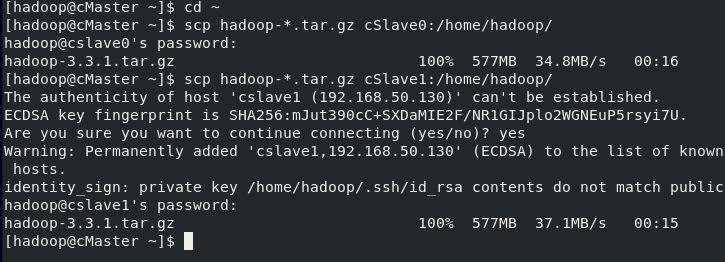
## Configure Worker Nodes

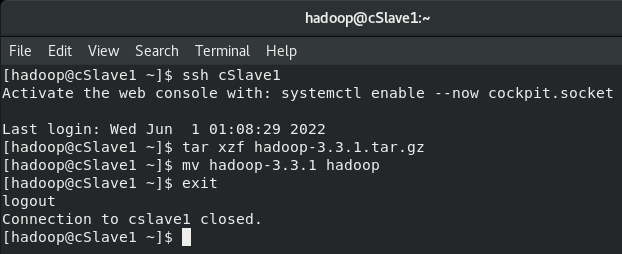
First copy the hadoop tarball file from cMaster to cSlaves machines like below:

[hadoop@cMaster ~]$ cd ~

[hadoop@cMaster ~]$ scp hadoop-\*.tar.gz cSlave0:/home/hadoop/

[hadoop@cMaster ~]$ scp hadoop-\*.tar.gz cSlave1:/home/hadoop/



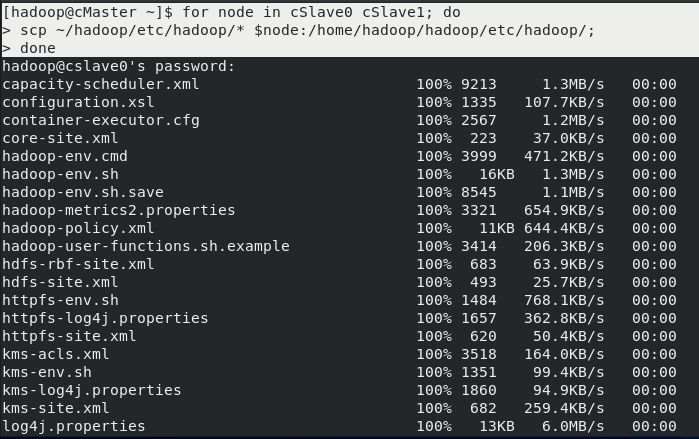


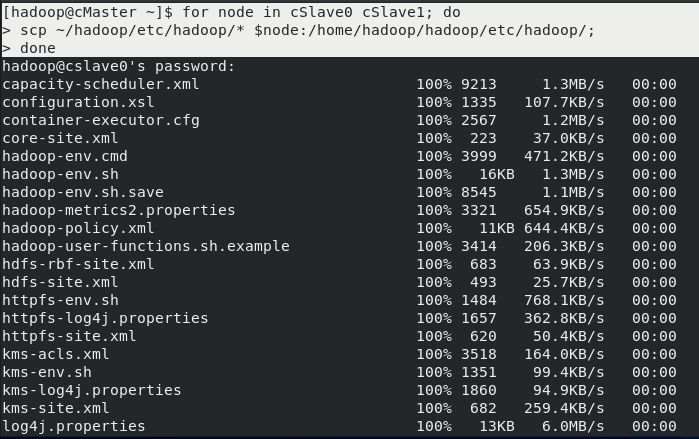
* From the cMaster, I duplicate the Hadoop configuration files to all worker nodes (cSlaves machine 0 and 1) by using command below:

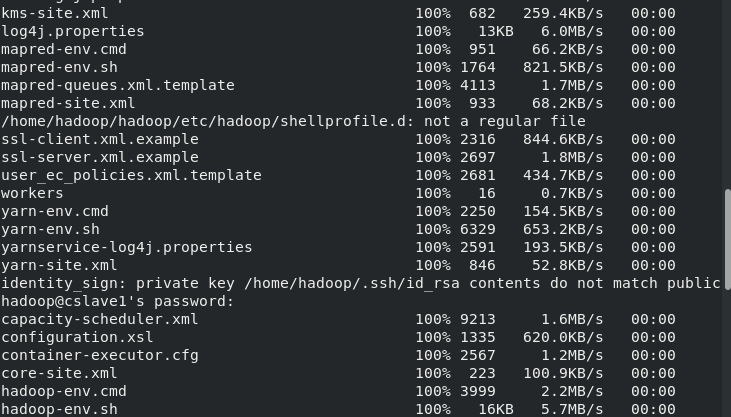
[hadoop@cMaster ~]$ for node in cSlave0 cSlave1; do

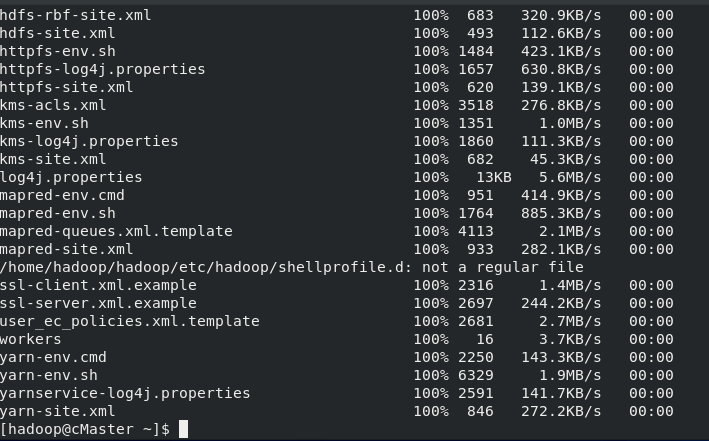
> scp ~/hadoop/etc/hadoop/\* $node:/home/hadoop/hadoop/etc/hadoop/;

> done







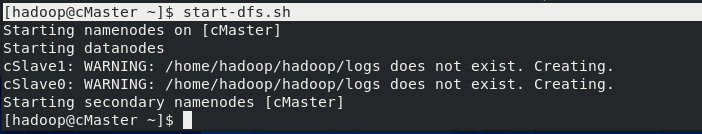


.

## Start Hadoop Clustering

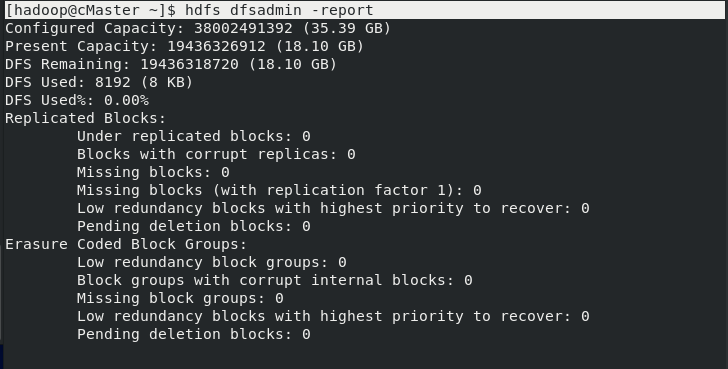
\*\*\*I login to cMaster as the hadoop user and I start the hadoop cluster by running the below command\*\*\*

[hadoop@cMaster ~]$ start-dfs.sh



I can get useful information about my hadoop cluster with the below command.

[hadoop@cMaster ~]$ hdfs dfsadmin -report



1. **Questions and Discussions**
2. Briefly describe the experience of this experiment?

A\ **What I can write about this experiment is it was very difficult and complicated to get through it without previous experiment, we struggled a lot to finally give something using Centos8 maybe using another software would be easier: So for Hadoop : I think it’s an open source framework that can use java to be implemented , it allows distributed processing of large datasets across clusters of servers using simple programming models.**

**Hadoop comes with code and configuration that references the JAVA\_HOME environment variable using java binary file, allowing it to run java code.**

**In this experiment I understood also how some components of Hadoop is working like Hadoop Distributed File (HDFS) and map reduce. I learned that Hadoop cluster consists of several nodes, each having its own disk space, memory, bandwidth, and processing. HDFS considers each disk drive and slave node in the cluster as inconsistent, and HDFS holds three copies of each data set throughout the cluster as a backup. The HDFS master node (NameNode) maintains each data block's information and copies.**

1. Consider how to configure and deploy a larger-scale cluster environment, such as how to deploy 100 servers?

A\ **As we can see below the last property string dfs.replication, indicates how much data time is needed being replicated in the cluster.**

<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>

<property>

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

<value>/home/hadoop/data/nameNode</value>

</property>

<property>

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

<value>/home/hadoop/data/dataNode</value>

</property>

<property>

<name>dfs.replication</name>

<value>100</value>

</property>

</configuration>