

# CDH installation & Application Test Report

He Shouchun (SCUID: 00001008350, Email: she@scu.edu)

Chapter 1. Prepare the virtual machine .....	2
1.1 Download virtual machine software .....	2
1.2 Plan the guest operation system .....	2
1.3 Post-install configurations.....	3
1.4 Network setting .....	4
1.5 Install the sshd service .....	5
1.6 Database installation choice.....	6
Chapter 2. Install Cloudera Manager and CDH.....	7
2.1 Preparation .....	7
2.2 Download and install the Cloudera Manager.....	7
2.3 Start the Cloudera Manager Admin Console .....	8
2.4 Install cluster nodes.....	10
2.5 Install the parcels .....	12
2.6 Install Hadoop services .....	13
2.7 Try to tune the performance of the server.....	15
Chapter 3. Run a testing program on the MapReduce .....	17
3.1 Preparation .....	17
3.2 Download testing source code .....	17
3.3 Prepare the test data and input/output folder in HDFS .....	18
3.4 Run test case t21.dat (successful) .....	19
3.5 Run test case t20.dat (no output).....	21

Read Cloudera installation guide: [http://www.cloudera.com/content/cloudera-content/cloudera-docs/CM4Ent/latest/Cloudera-Manager-Installation-Guide/cmig\\_intro\\_to\\_cm\\_install.html](http://www.cloudera.com/content/cloudera-content/cloudera-docs/CM4Ent/latest/Cloudera-Manager-Installation-Guide/cmig_intro_to_cm_install.html)

## Chapter 1. Prepare the virtual machine

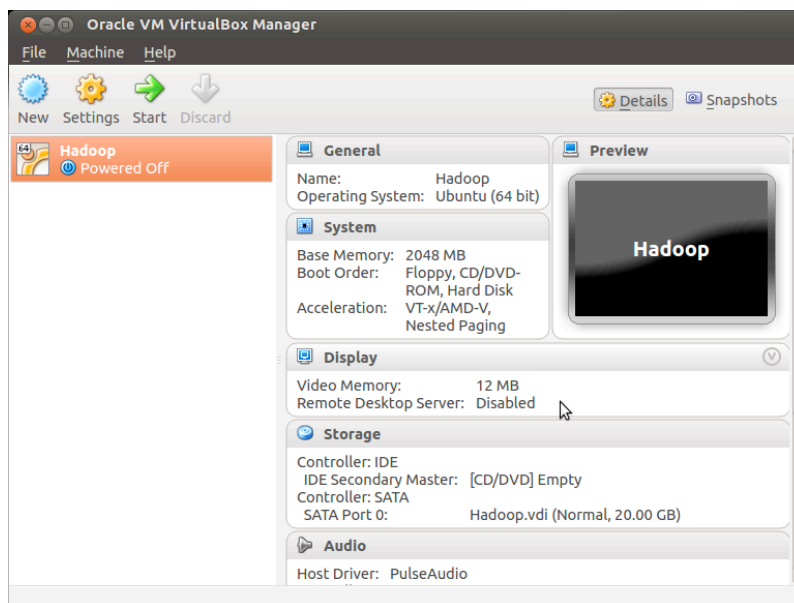
### 1.1 Download virtual machine software

Download VirtualBox from oracle website: <http://www.virtualbox.org/>, and install it on host machine. Then create a new virtual machine:

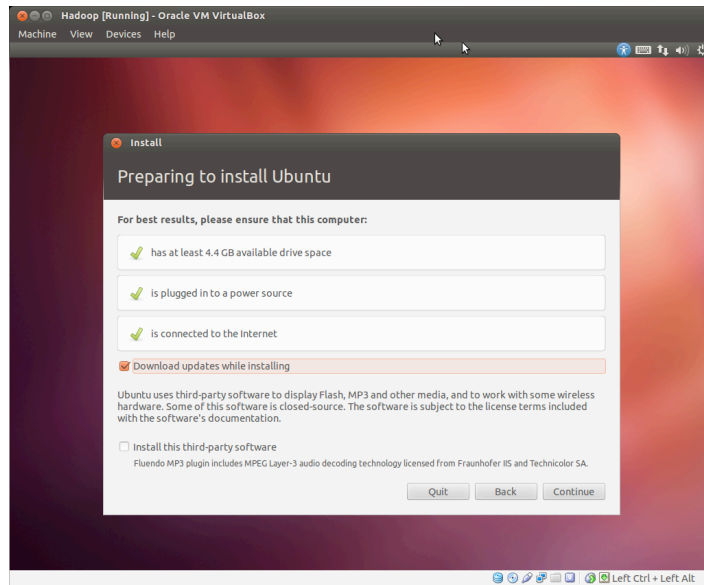


### 1.2 Plan the guest operation system

Selected the Ubuntu 12.04 (64bit). According to the Cloudera installation guide, plan 20GB virtual HD space and 2GB RAM for the guest OS. Just follow the Linux installation steps to install it in the VM.



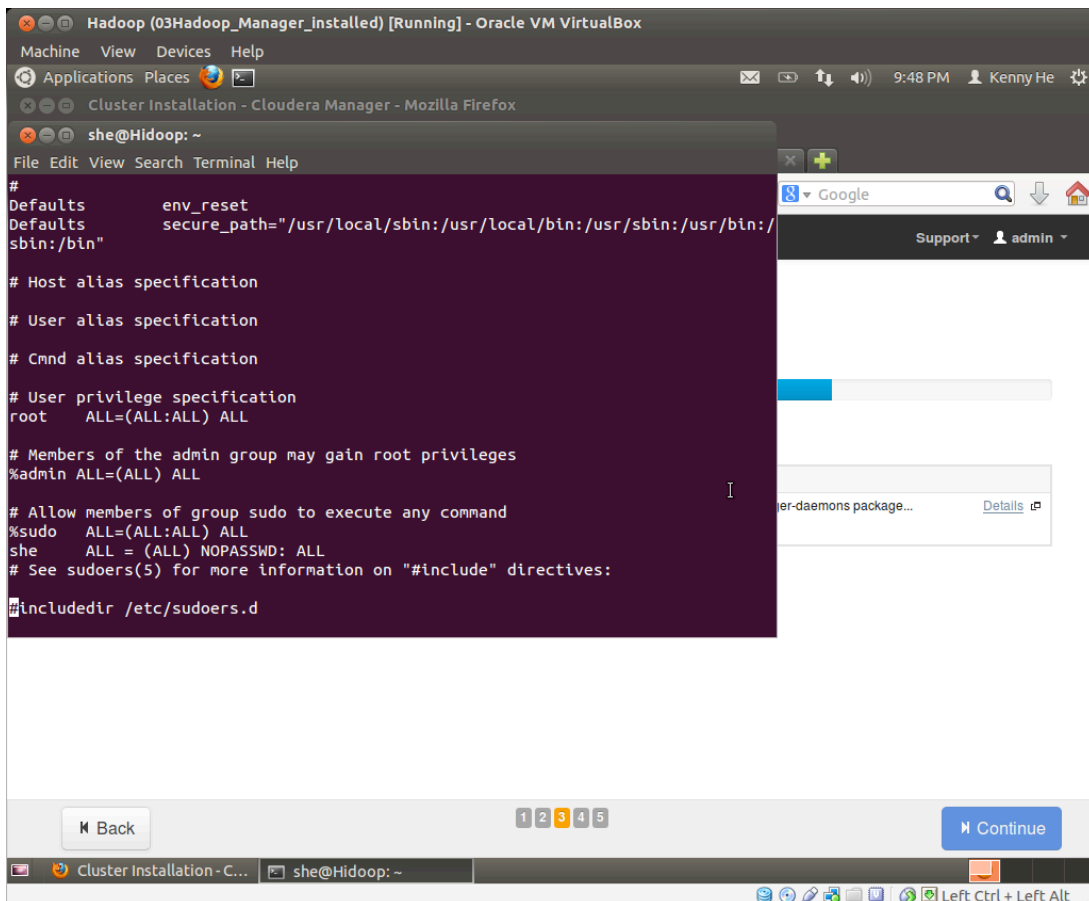
Ubuntu 12.04 installation:



### 1.3 Post-install configurations

Change the root password: `sudo -i`, input password and then switch to root mode, and `passwd root` to change the password of root.

Create a new account “she”, and make it a password-less sudo user by edit `/etc/sudoers`:



In the installation of CDH the system needs a account which has sudo privileges without input a password. So this step is very important.

Run software update manager to install all the patches: the English languages , the security patches, bug fixes, etc. After finished, save a snapshot called “Ubuntu-Install”.

## 1.4 Network setting

The default virtual network adapter for VirtualBox guest OS is NAT, which allows the guest OS users to access outside network but it does not allow the host OS users and other outside users to access the guest OS. To enable the host OS users to use/test the service on the guest OS, we need to add another network adapter whose type is “Bridged Adapter” and make this virtual network adapter on guest OS to be bridged to the Ethernet adapter “eth0”. However there is a limitation that only when the network cable is connected can both the guest virtual network adapter and the host Ethernet adapter work well. If the cable is pulled out, both adapters will be “ifdown” and can not connect to each other.

Setup static IP address for the Ethernet adapters of both the host and the guest OS:

Host: 192.168.1.100/24

Guest: 192.168.1.101/24

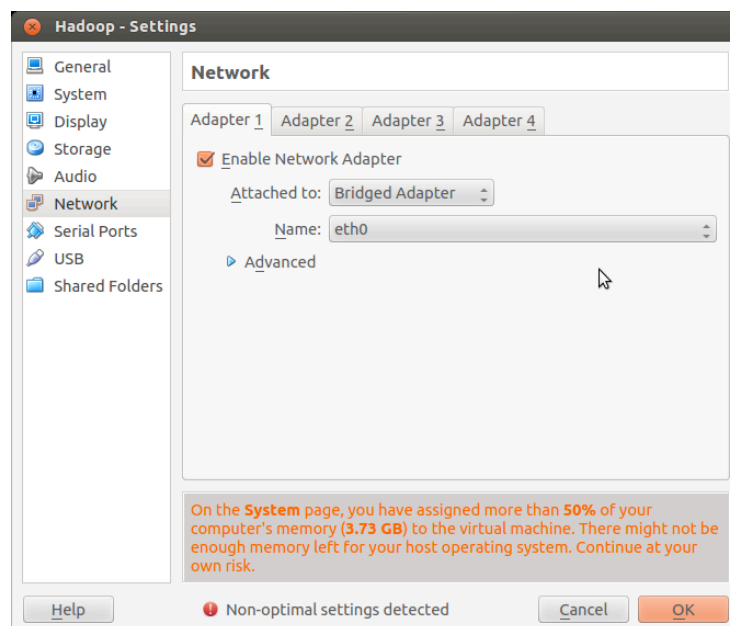
And must set a host name for the guest OS. Or we will meet problems during the installation.

I also tried to bridge the guest virtual network adapter to the wireless adapter. The VirtualBox official website shows this solution can work but on my laptop it does not work at all (the possible reason is that my host OS is Ubuntu and there are some limitations). I know it works on VMWare but I prefer using free VirtualBox VM.

So there are two adapters for the guest OS:

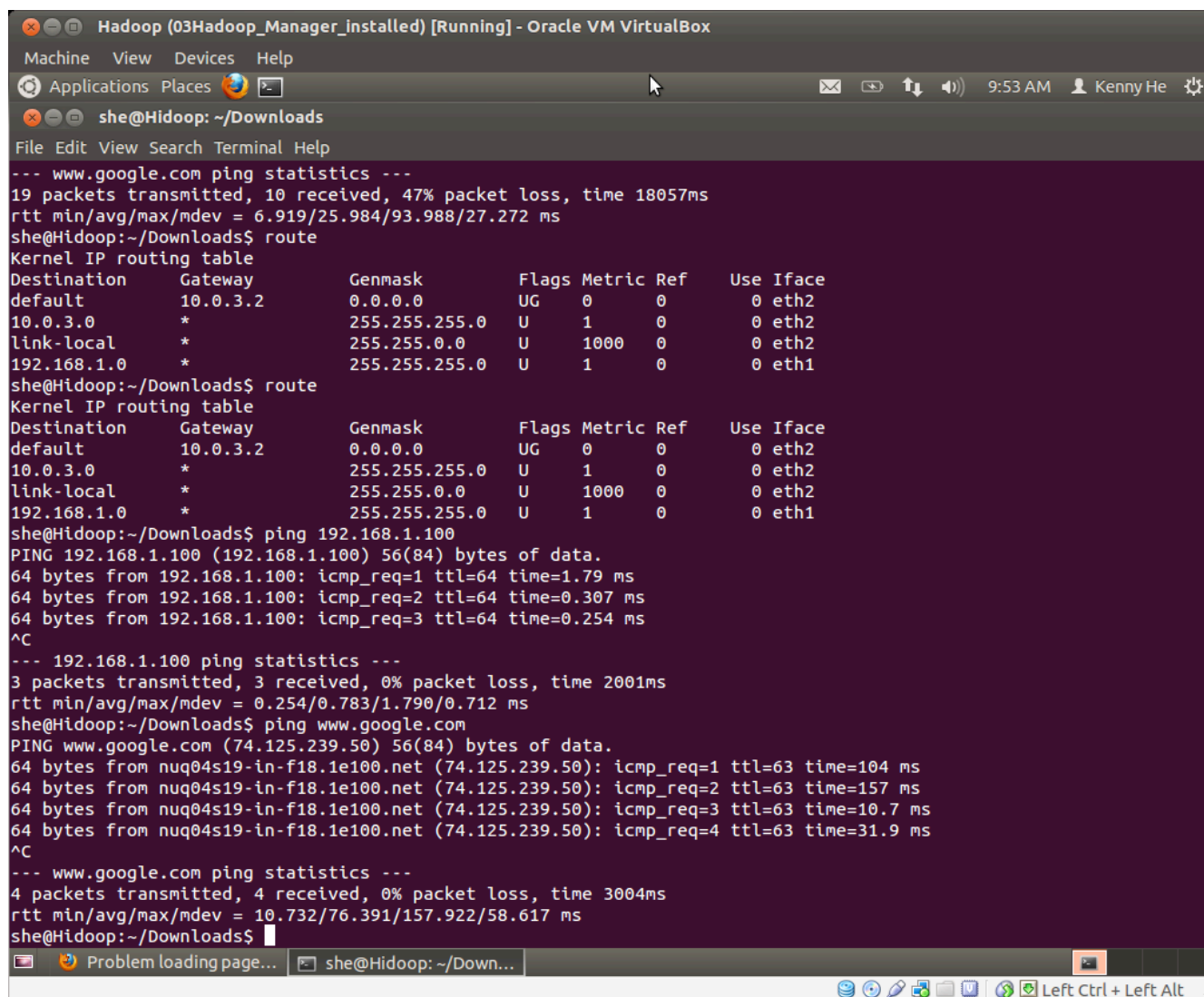
Adapter 1: “Bridged Adapter”, bridge to host Ethernet adapter eth0 for connecting to the host when Ethernet cable is connected;

Adapter 2: NAT, for Internet access via any Internet connection (e.g., through wlan0) on the host.



And then configure the guest OS route table to make sure that the route to Internet will go through Adapter 2 only and the Adapter 1 interface is for routing to local network 192.168.1.0/24 only.

If the configuration is correct, we can ping between the guest OS and host OS. And we can also open an Internet website (e.g. [www.google.com](http://www.google.com), [www.cloudera.com](http://www.cloudera.com)) from the Firefox web browser on the guest OS.



```
Hadoop (03Hadoop_Manager_Installed) [Running] - Oracle VM VirtualBox
Machine View Devices Help
Applications Places
she@Hadoop: ~/Downloads
File Edit View Search Terminal Help
--- www.google.com ping statistics ---
19 packets transmitted, 10 received, 47% packet loss, time 18057ms
rtt min/avg/max/mdev = 6.919/25.984/93.988/27.272 ms
she@Hadoop:~/Downloads$ route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default 10.0.3.2 0.0.0.0 UG 0 0 0 eth2
10.0.3.0 * 255.255.255.0 U 1 0 0 eth2
link-local * 255.255.0.0 U 1000 0 0 eth2
192.168.1.0 * 255.255.255.0 U 1 0 0 eth1
she@Hadoop:~/Downloads$ route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default 10.0.3.2 0.0.0.0 UG 0 0 0 eth2
10.0.3.0 * 255.255.255.0 U 1 0 0 eth2
link-local * 255.255.0.0 U 1000 0 0 eth2
192.168.1.0 * 255.255.255.0 U 1 0 0 eth1
she@Hadoop:~/Downloads$ ping 192.168.1.100
PING 192.168.1.100 (192.168.1.100) 56(84) bytes of data.
64 bytes from 192.168.1.100: icmp_req=1 ttl=64 time=1.79 ms
64 bytes from 192.168.1.100: icmp_req=2 ttl=64 time=0.307 ms
64 bytes from 192.168.1.100: icmp_req=3 ttl=64 time=0.254 ms
^C
--- 192.168.1.100 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.254/0.783/1.790/0.712 ms
she@Hadoop:~/Downloads$ ping www.google.com
PING www.google.com (74.125.239.50) 56(84) bytes of data.
64 bytes from nuq04s19-in-f18.1e100.net (74.125.239.50): icmp_req=1 ttl=63 time=104 ms
64 bytes from nuq04s19-in-f18.1e100.net (74.125.239.50): icmp_req=2 ttl=63 time=157 ms
64 bytes from nuq04s19-in-f18.1e100.net (74.125.239.50): icmp_req=3 ttl=63 time=10.7 ms
64 bytes from nuq04s19-in-f18.1e100.net (74.125.239.50): icmp_req=4 ttl=63 time=31.9 ms
^C
--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 10.732/76.391/157.922/58.617 ms
she@Hadoop:~/Downloads$
```

## 1.5 Install the sshd service

To maintain the server running in the guest, it is necessary to install SSH service:

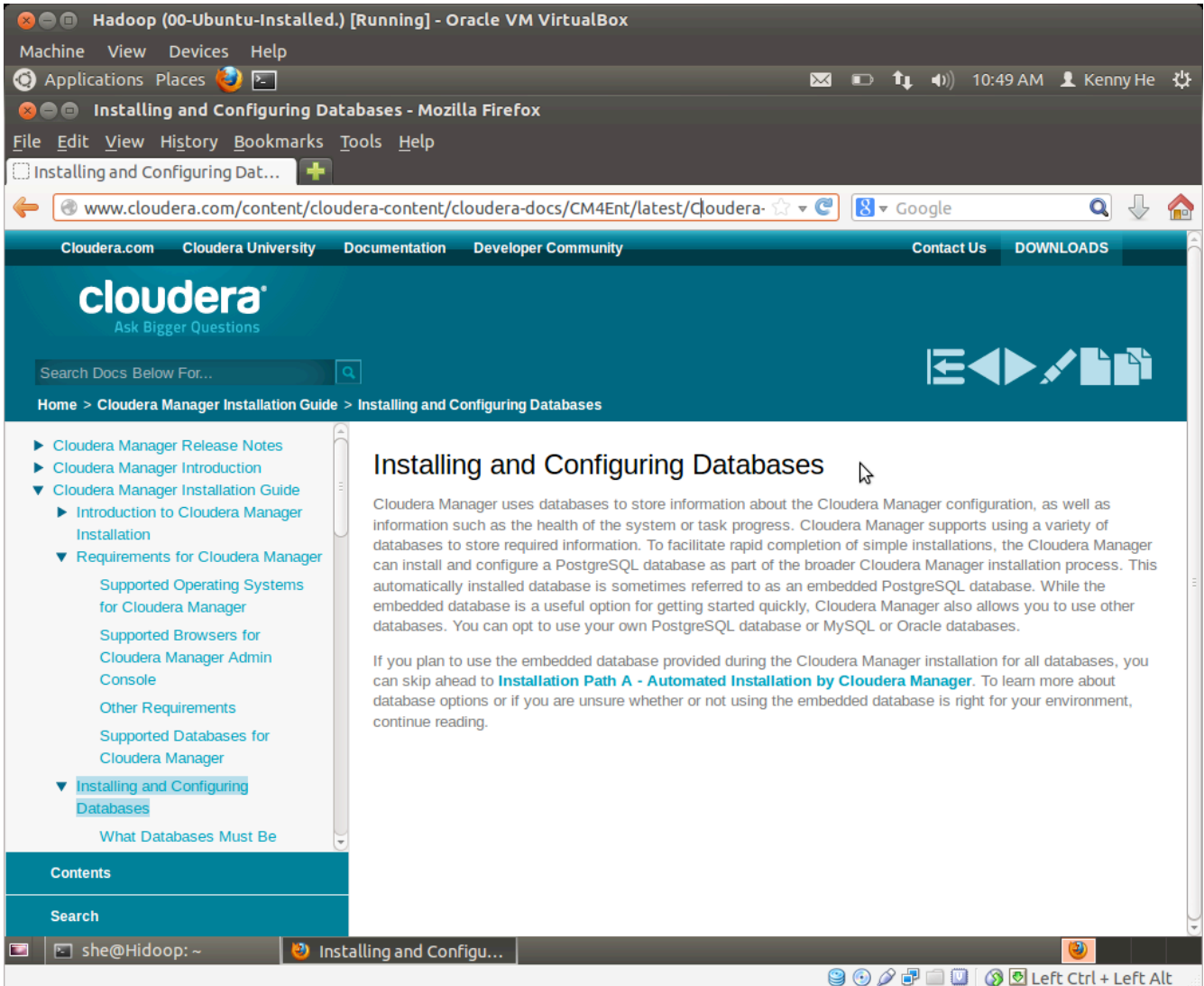
```
sudo apt-get install openssh-server
```

Then change the firewall setting on the guest OS. To make life easier, I simply disabled the firewall service: `sudo ufw disable`. The firewall setting may cause not only the SSH but many other services are not accessible from the host OS and the outside.

After the installation finished, try to connect to the guest OS with putty by the IP address of the bridged virtual Ethernet adapter.

## 1.6 Database installation choice

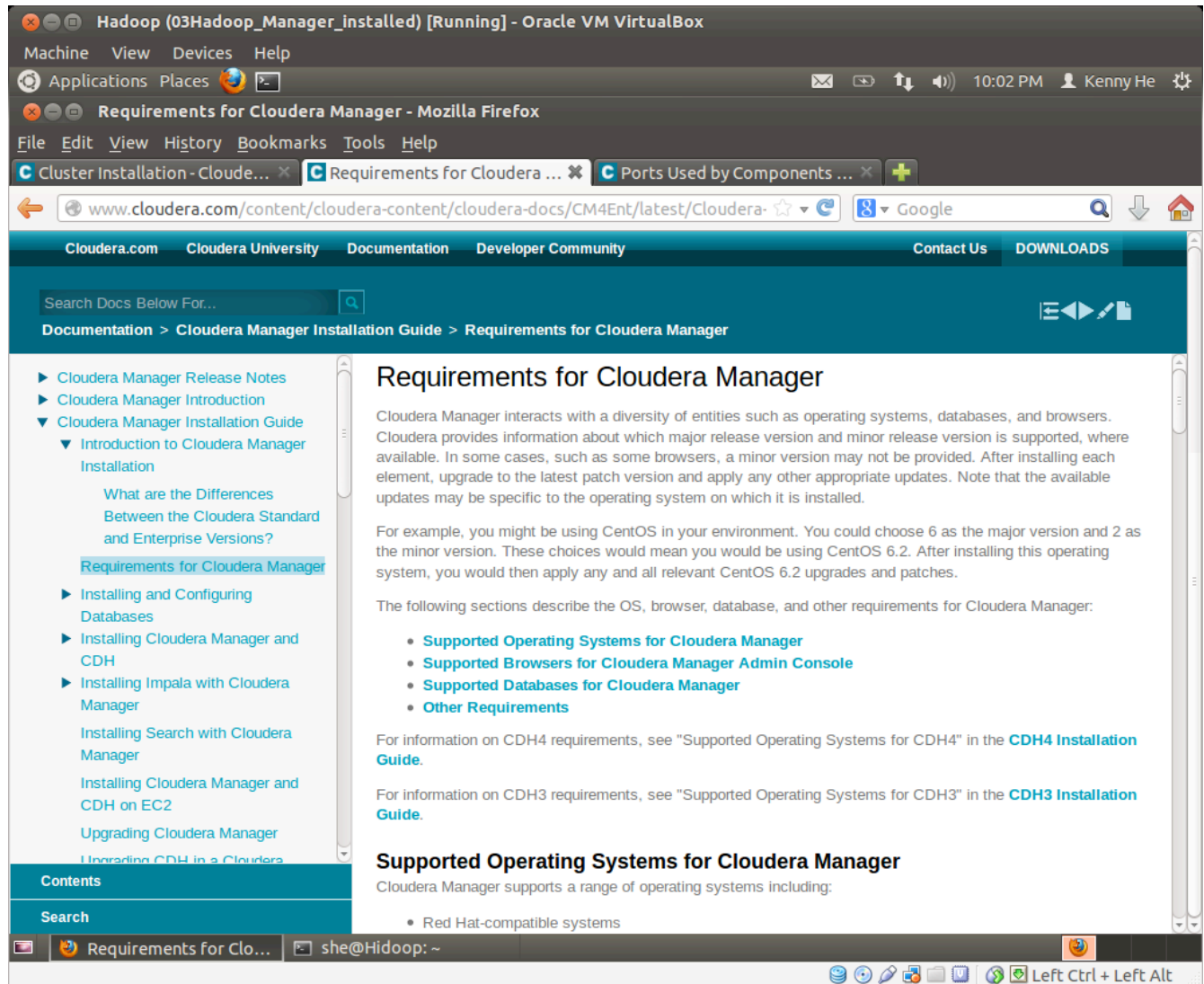
Following the instructions on [http://www.cloudera.com/content/cloudera-content/cloudera-docs/CM4Ent/latest/Cloudera-Manager-Installation-Guide/cmig\\_installing\\_configuring\\_dbs.html](http://www.cloudera.com/content/cloudera-content/cloudera-docs/CM4Ent/latest/Cloudera-Manager-Installation-Guide/cmig_installing_configuring_dbs.html). To make life easier, and get an opportunity to try and learn PostgreSQL DB (I am quite familiar with ORACLE, Sybase and MS SQL Server and MySQL), I prefer “Path A” automatically installs embedded PostgreSQL database.



# Chapter 2. Install Cloudera Manager and CDH

## 2.1 Preparation

Before start, I read the requirements of the installation carefully. All the details need to be read thoroughly and strictly followed. Or, the installation may fail.



To make life easier, I chose Path A in the installation guide at the first installation. Follow the steps in this page:

[http://www.cloudera.com/content/cloudera-content/cloudera-docs/CM4Ent/latest/Cloudera-Manager-Installation-Guide/cmig\\_install\\_path\\_A.html?scroll=cmig\\_topic\\_6\\_5](http://www.cloudera.com/content/cloudera-content/cloudera-docs/CM4Ent/latest/Cloudera-Manager-Installation-Guide/cmig_install_path_A.html?scroll=cmig_topic_6_5)

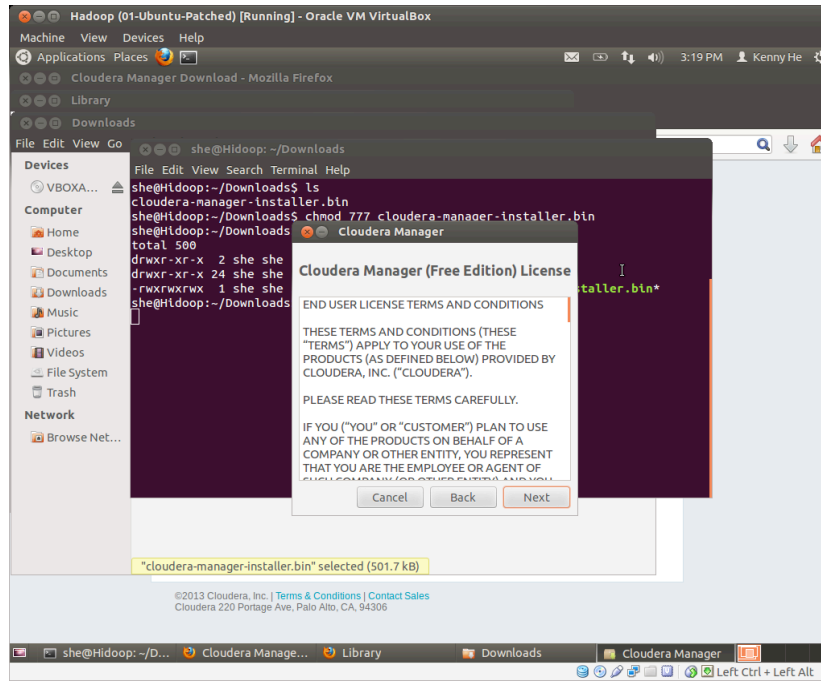
## 2.2 Download and install the Cloudera Manager

Download the installer “cloudera-manager-installer.bin”, change its attributes to “500”, and then run it

with root privilege:

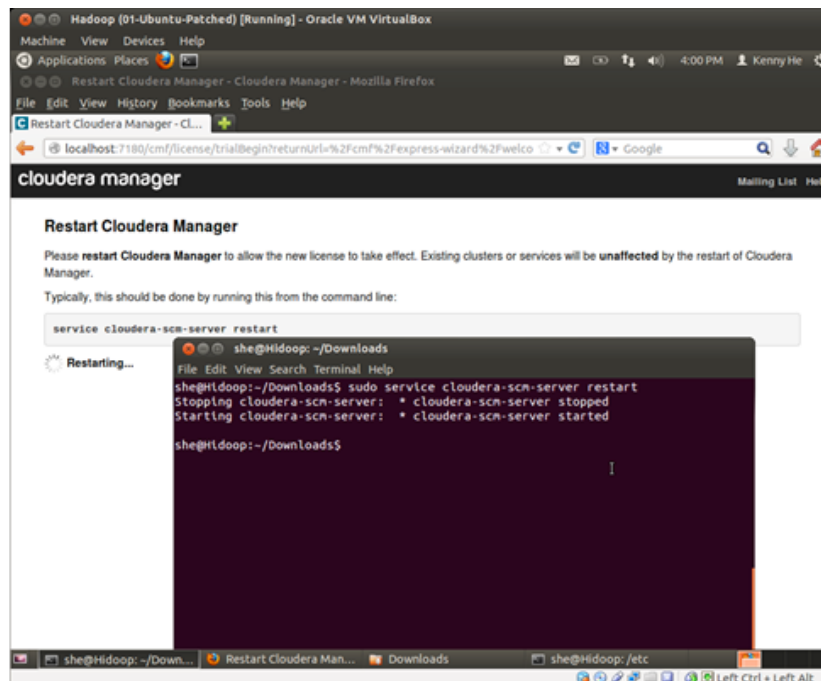
```
sudo cloudera-manager-installer.bin
```

The installation application will download the application from <http://www.cloudera.com/> and install. Depends on the network speed, the installation may take very long time. After successfully install it, create a VM snapshot.



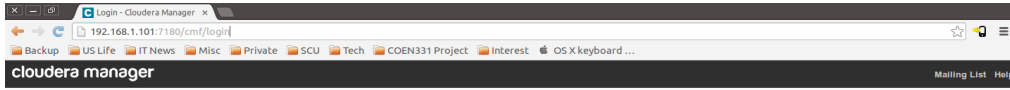
## 2.3 Start the Cloudera Manager Admin Console

Run the admin console by opening the URL <http://localhost:7180/> in the web browser of the guest OS.





If the network connection between the host and guest OS has been setup correctly, the admin console on the host OS by opening the URL <http://192.168.1.101:7180/> (the following screen shot is got from my host OS):



**Login**

Username:

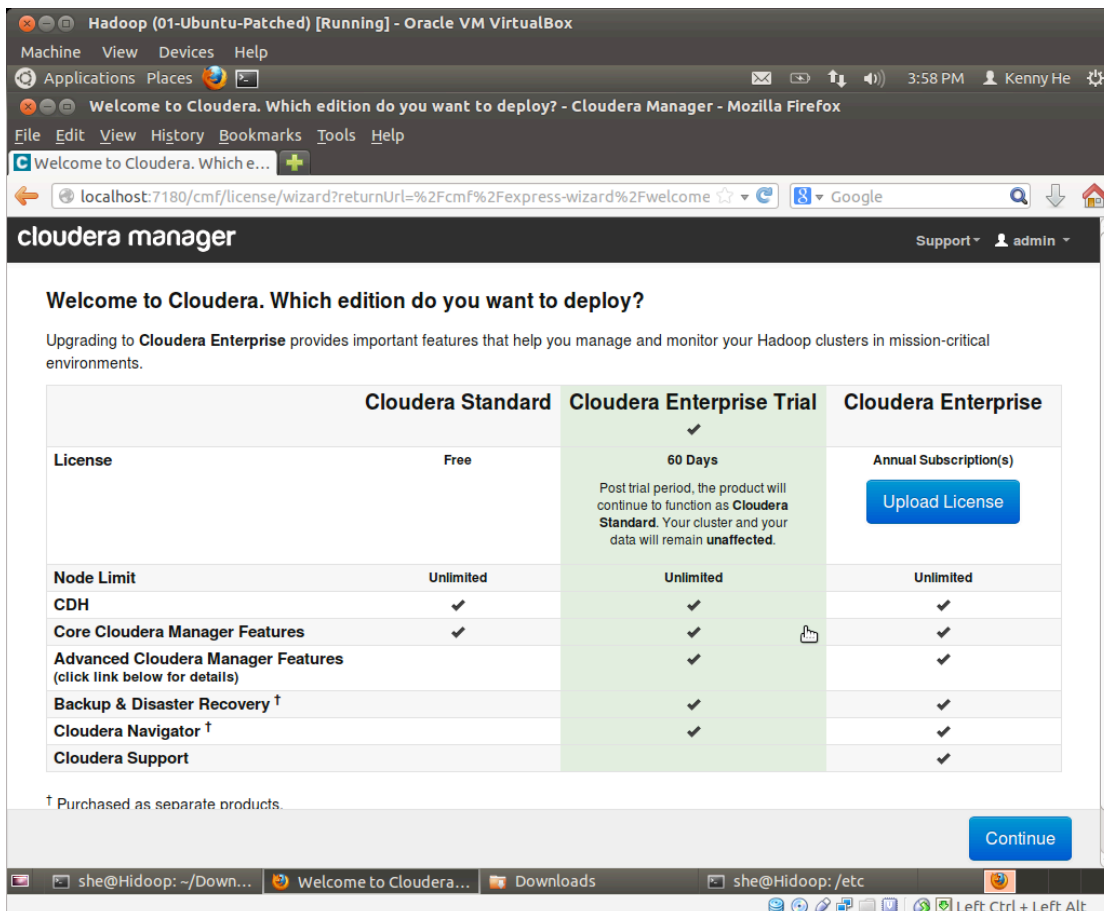
Password:

Remember me on this computer.

[Login](#)

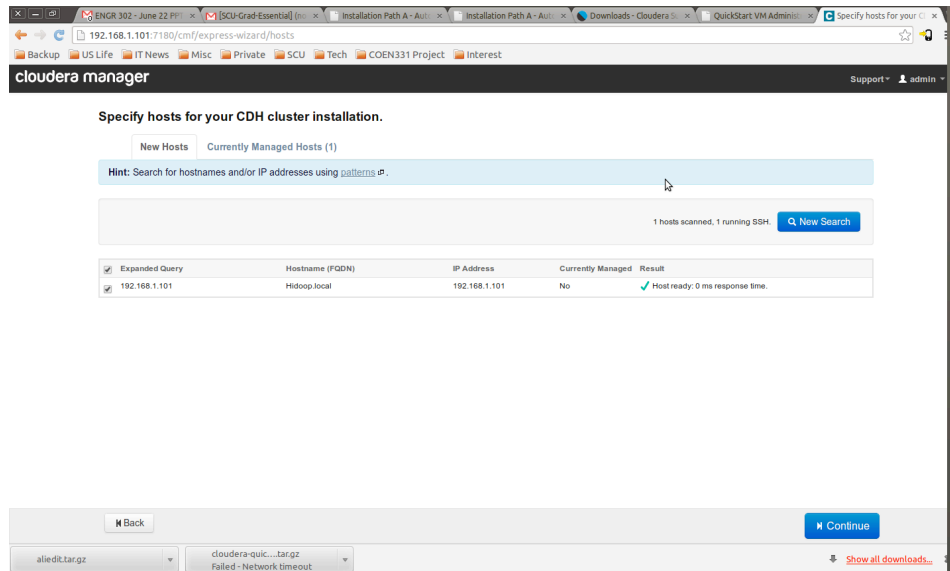
Then we can continue and restart the cloudera-scm-server service according to the instruction on the screen.

In the next page, the installation program will ask which edition to choose. Of course we can only choose the “Cloudera Standard” edition;



## 2.4 Install cluster nodes

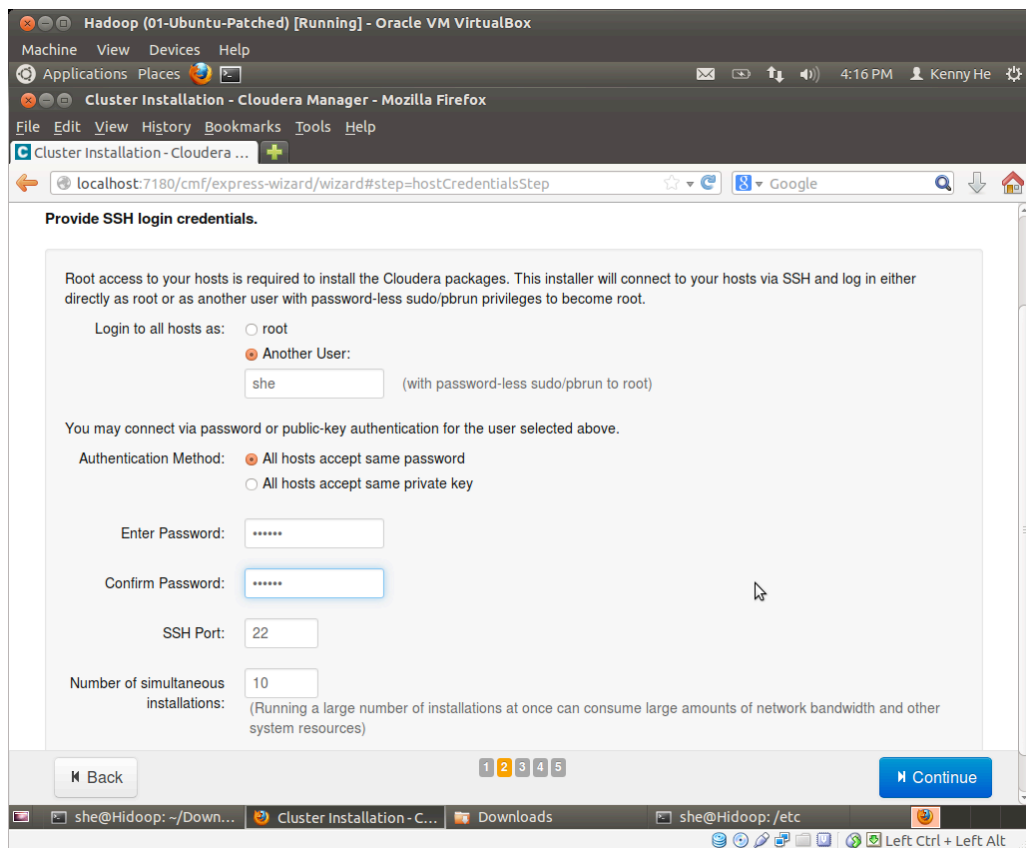
Click “Continue” and we can see a screen to ask for searching the nodes which will be installed as clusters. Input the address pattern 192.168.1.[100-101] (Refer to the IP address I set in chapter 1 when I was preparing the virtual machine network setup). And then go forward.



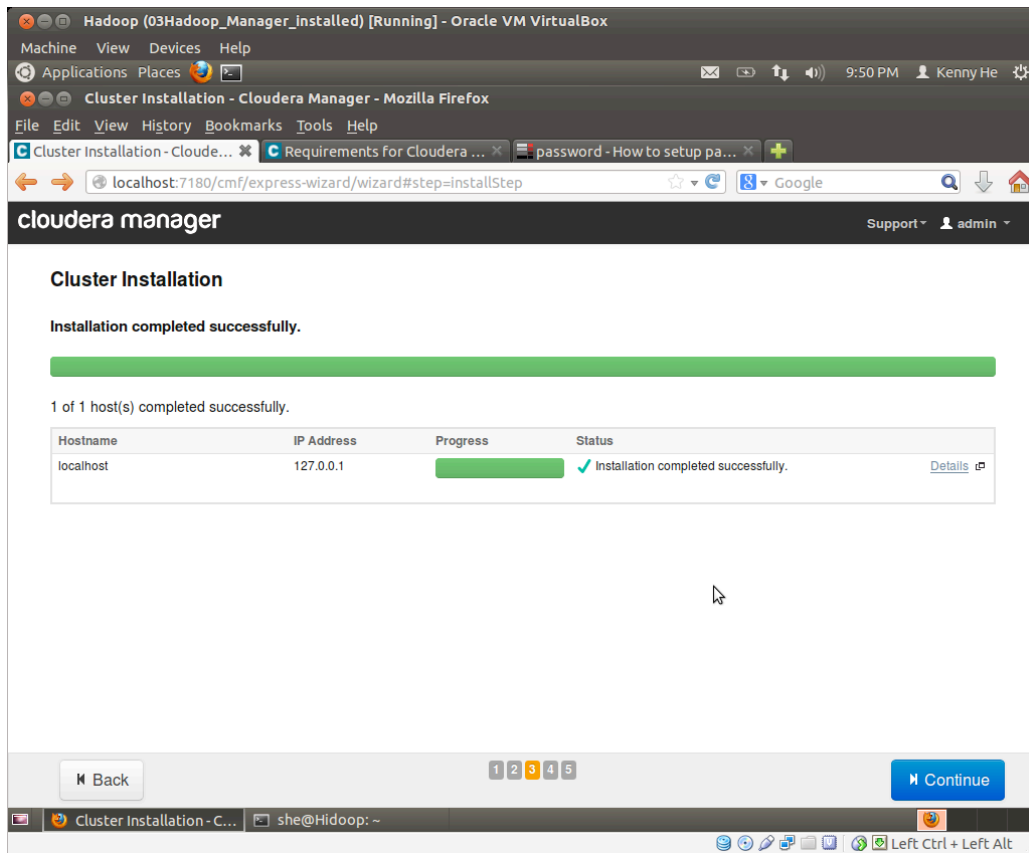
A valid host name and IP address should be provided in this step. We can also use “localhost” or “127.0.0.1”, but that is not recommended since the loop back IP address can be used for testing only and should never use them in real business network.

Fill in the installation options in the next 5 steps.

If the system cannot continue due to “failed to sudo”, that means you have not created an account which has password-less sudo privilege. Please go back to see how to create such account in “3. Post-install configurations” of chapter 1.

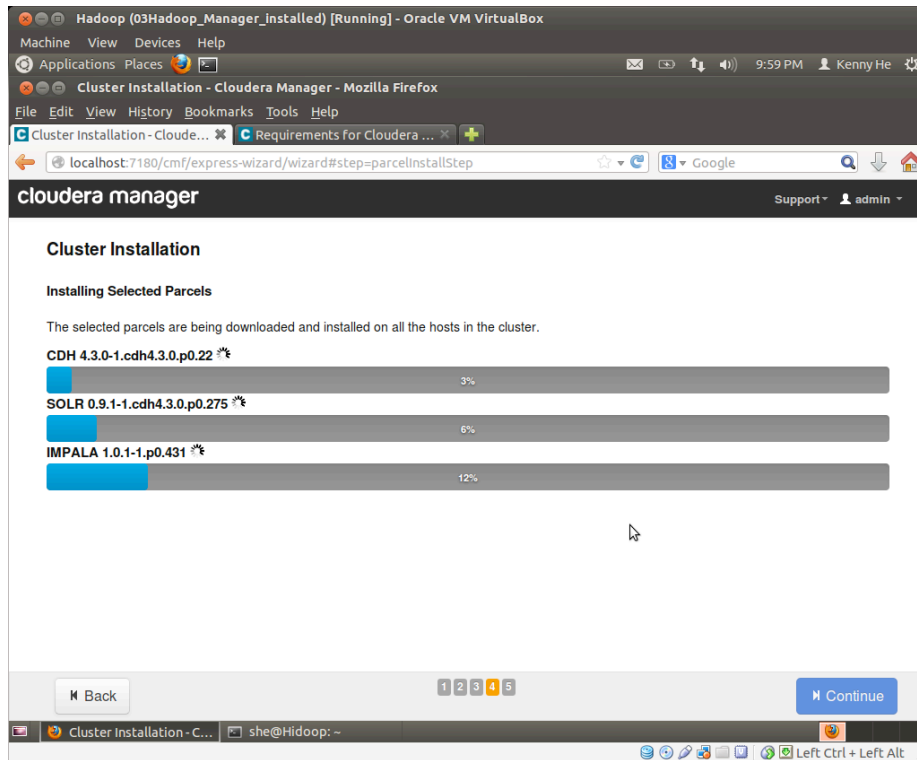


Then you can see the “Installation complete successfully” screen as below. Create a VM snapshot.



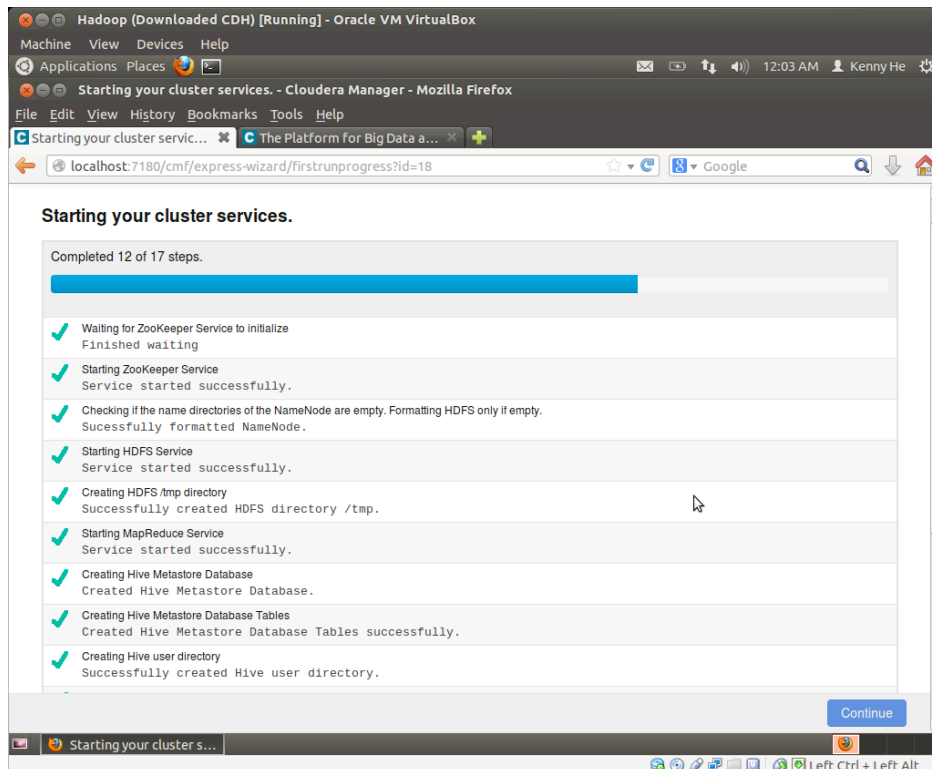
## 2.5 Install the parcels

Move forward by clicking the “Continue” button and install the parcels. The installation of Parcels may take a long time (about one hour on my computer). Create another snapshot of the VM once parcels installation is finished.

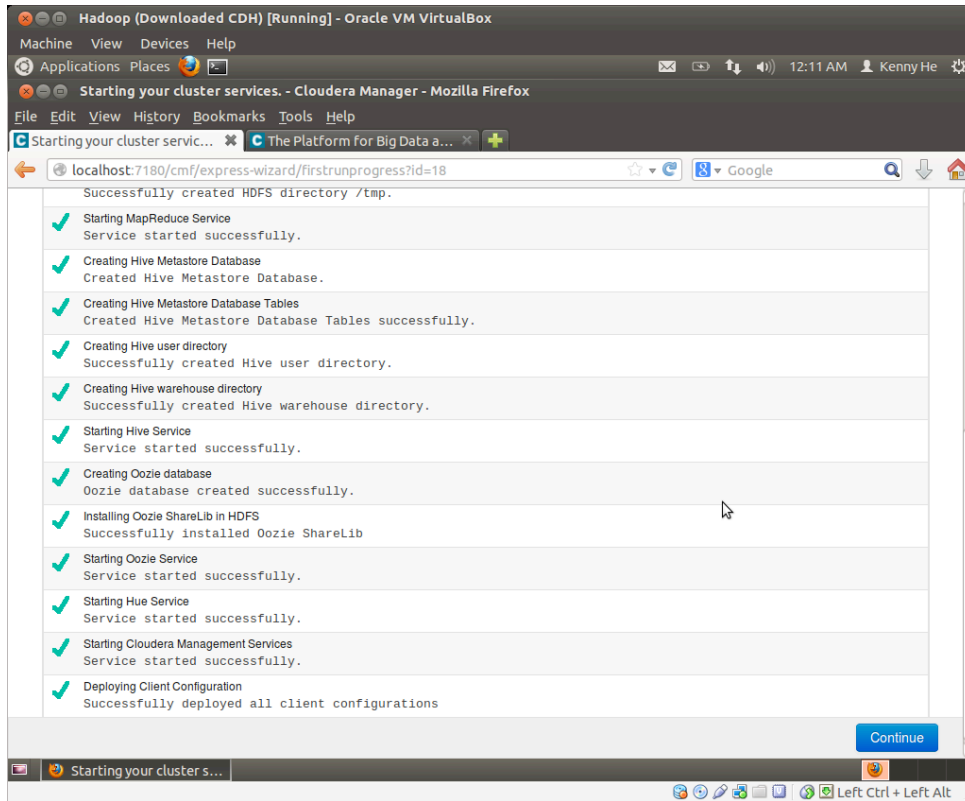


## 2.6 Install Hadoop services

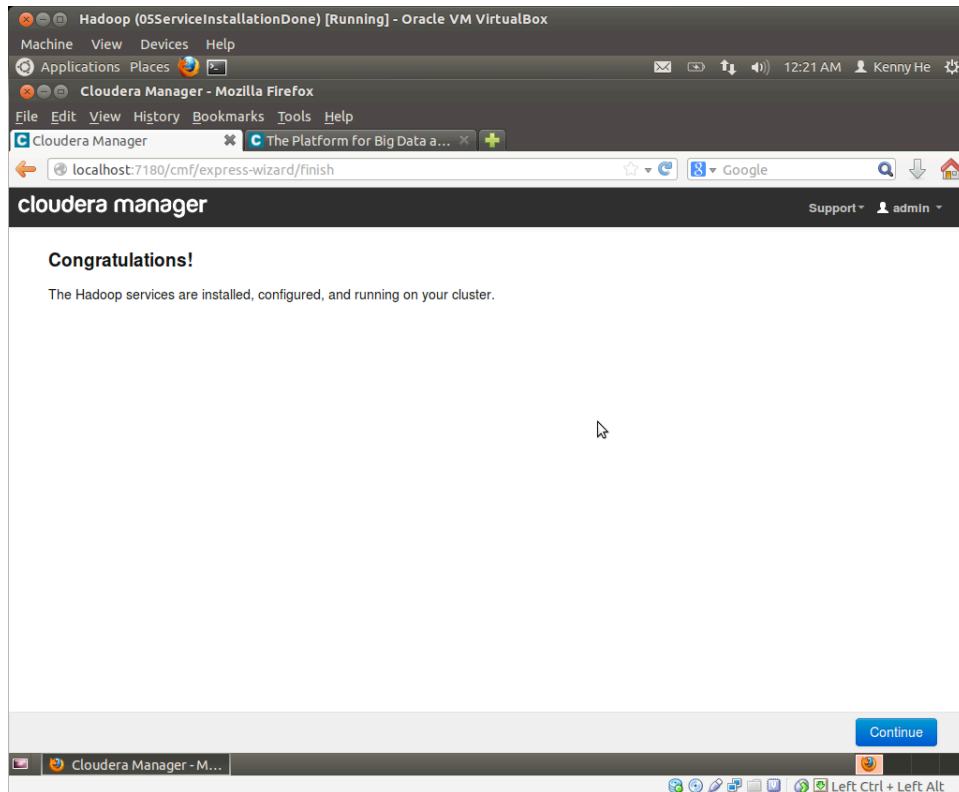
After installation finished, select “Core Hadoop services”, test the database connection, and then continue. The system will start the Hadoop cluster services:



Once all the services have been successfully started, that means the system installation finished. Click the “Continue” button. Create a snapshot of the virtual machine.

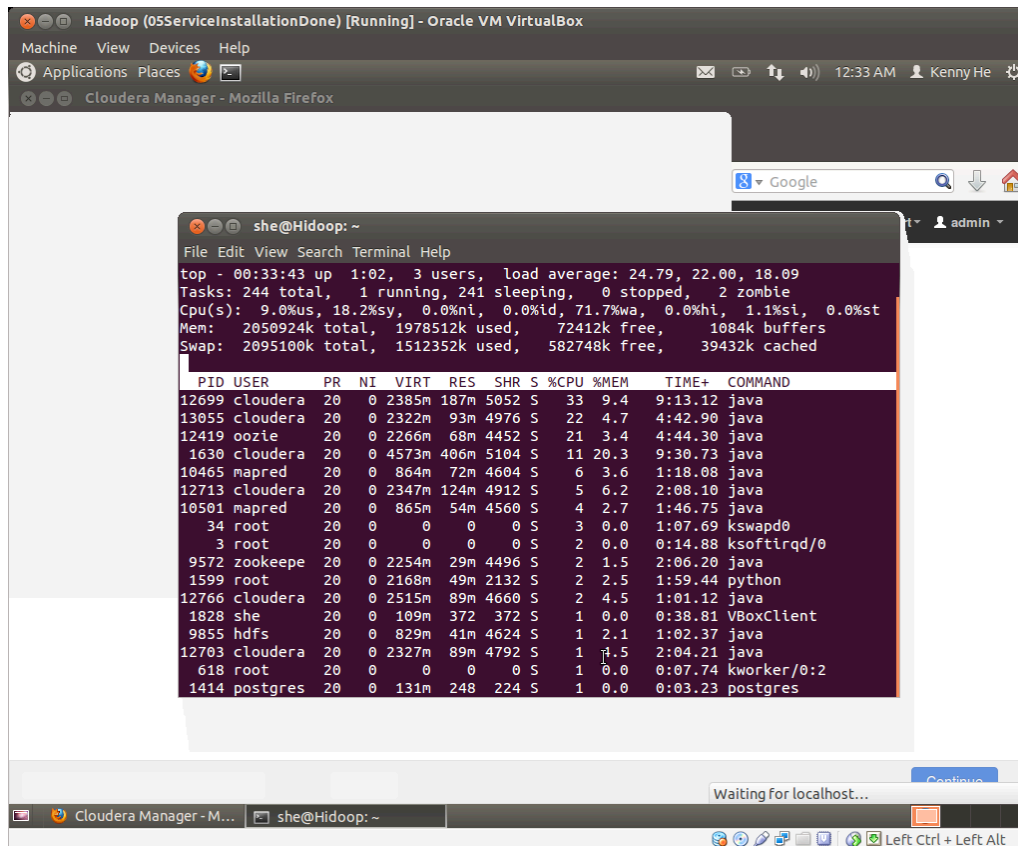


Click the “Continue”, wait for a few minutes, the page will be redirected to the “Congratulations”:



## 2.7 Try to tune the performance of the server

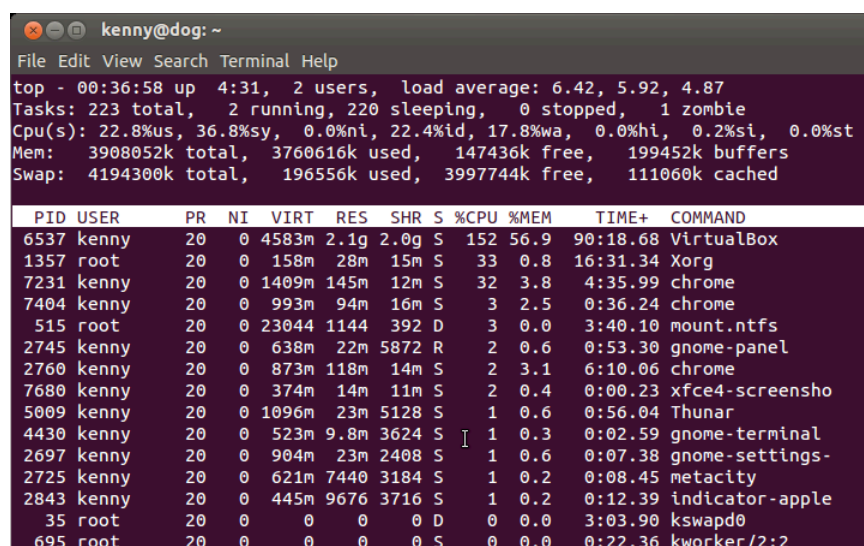
The virtual machine is running Hadoop now and it becomes very busy and almost freezes. Open a terminal and run “top” command and see that how busy the cluster runs:



```
she@Hadoop: ~
File Edit View Search Terminal Help
top - 00:33:43 up 1:02, 3 users, load average: 24.79, 22.00, 18.09
Tasks: 244 total, 1 running, 241 sleeping, 0 stopped, 2 zombie
Cpu(s): 9.0%us, 18.2%sy, 0.0%ni, 0.0%id, 71.7%wa, 0.0%hi, 1.1%st, 0.0%st
Mem: 2050924k total, 1978512k used, 72412k free, 1084k buffers
Swap: 2095100k total, 1512352k used, 582748k free, 39432k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 12699 cloudera  20   0 2385m 187m 5052 S   33   9.4   9:13.12 java
 13055 cloudera  20   0 2322m  93m 4976 S   22   4.7   4:42.90 java
 12419 oozie     20   0 2266m  68m 4452 S   21   3.4   4:44.30 java
 1630 cloudera  20   0 4573m 406m 5104 S   11  20.3   9:30.73 java
10465 mapred  20   0  864m  72m 4604 S    6   3.6   1:18.08 java
12713 cloudera  20   0 2347m 124m 4912 S    5   6.2   2:08.10 java
10591 mapred  20   0  865m  54m 4560 S    4   2.7   1:46.75 java
   34 root      20   0    0    0    0 S    3   0.0   1:07.69 kswapd0
    3 root      20   0    0    0    0 S    2   0.0   0:14.88 ksoftirqd/0
  9572 zookeepe  20   0 2254m  29m 4496 S    2   1.5   2:06.20 java
 1599 root      20   0 2168m  49m 2132 S    2   2.5   1:59.44 python
12766 cloudera  20   0 2515m  89m 4660 S    2   4.5   1:01.12 java
 1828 she      20   0  109m  372  372 S    1   0.0   0:38.81 VBoxClient
 9855 hdfs     20   0  829m  41m 4624 S    1   2.1   1:02.37 java
12703 cloudera  20   0 2327m  89m 4792 S    1   1.5   2:04.21 java
   618 root      20   0    0    0    0 S    1   0.0   0:07.74 kworker/0:2
 1414 postgres  20   0  131m  248  224 S    1   0.0   0:03.23 postgres
```

And I ran “top” command on my host OS, the VirtualBox consumes almost 150%-300% CPU time (my CPU has 4 cores):



```
kenny@dog: ~
File Edit View Search Terminal Help
top - 00:36:58 up 4:31, 2 users, load average: 6.42, 5.92, 4.87
Tasks: 223 total, 2 running, 220 sleeping, 0 stopped, 1 zombie
Cpu(s): 22.8%us, 36.8%sy, 0.0%ni, 22.4%id, 17.8%wa, 0.0%hi, 0.2%si, 0.0%st
Mem: 3908052k total, 3760616k used, 147436k free, 199452k buffers
Swap: 4194300k total, 196556k used, 3997744k free, 111060k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
  6537 kenny     20   0 4583m 2.1g 2.0g S  152  56.9  90:18.68 VirtualBox
 1357 root      20   0  158m  28m  15m S   33   0.8  16:31.34 Xorg
  7231 kenny     20   0 1409m 145m  12m S   32   3.8   4:35.99 chrome
  7404 kenny     20   0  993m  94m  16m S    3   2.5   0:36.24 chrome
   515 root      20   0 23044 1144  392 D    3   0.0   3:40.10 mount.ntfs
  2745 kenny     20   0  638m  22m 5872 R    2   0.6   0:53.30 gnome-panel
  2760 kenny     20   0  873m 118m  14m S    2   3.1   6:10.06 chrome
  7680 kenny     20   0  374m  14m  11m S    2   0.4   0:00.23 xfce4-screensho
  5009 kenny     20   0 1096m  23m 5128 S    1   0.6   0:56.04 Thunar
  4430 kenny     20   0  523m  9.8m 3624 S    1   0.3   0:02.59 gnome-terminal
  2697 kenny     20   0  904m  23m 2408 S    1   0.6   0:07.38 gnome-settings-
  2725 kenny     20   0  621m 7440 3184 S    1   0.2   0:08.45 metacity
  2843 kenny     20   0  445m 9676 3716 S    1   0.2   0:12.39 indicator-apple
    35 root      20   0    0    0    0 D    0   0.0   3:03.90 kswapd0
   695 root      20   0    0    0    0 S    0   0.0   0:22.36 kworker/2:2
```

The virtual machine is almost freezes so I can only SSH from my host machine to it and do some tuning.

First, change the init mode so the VM starts only in character UI and disable the GUI.

Secondly, disable all the services.

Thirdly, shut down the VM, disable the USB, sound card, and other useless HW from the VM, and give it more resources such as CPU cores and RAM.

Finally, copy the virtual machine to a faster computer.



## Chapter 3. Run a testing program on the MapReduce

### 3.1 Preparation

Just follow the guide of “Hadoop Tutorial”: <http://www.cloudera.com/content/cloudera-content/cloudera-docs/HadoopTutorial/CDH4/Hadoop-Tutorial.html>

Since I cannot get enough hardware resource to run a cluster with three or more virtual machines with Hadoop installed, I can only try to run the test program on a single node cluster. Then I need to make some configurations to the Hadoop virtual machine I installed to single node mode.

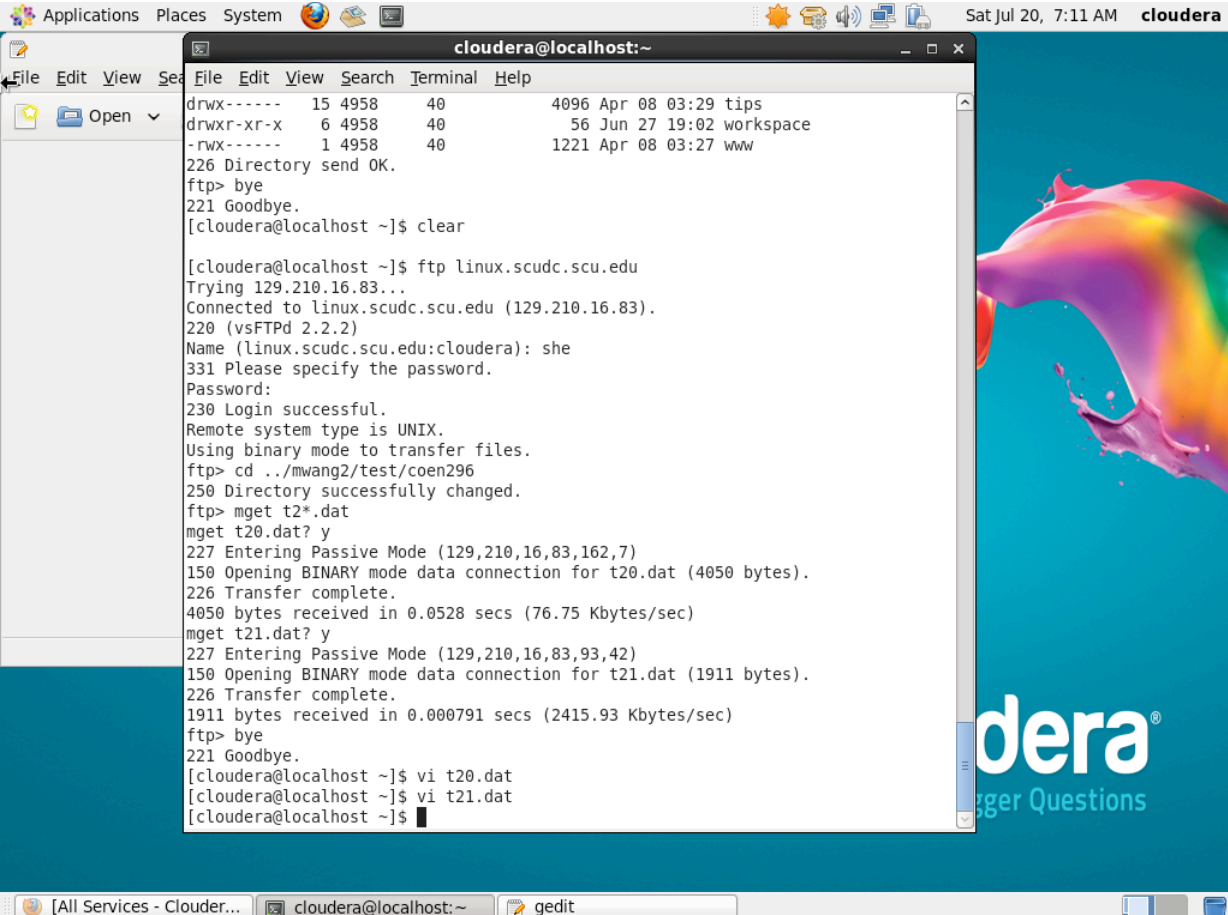
To make the life easier, I simply use the preinstalled quick VM provided by cloudera: <https://ccp.cloudera.com/display/SUPPORT/Cloudera+QuickStart+VM>

Download the pre-build VM for VirtualBox and import.

Download the latest stable (version 1.1.2) Hadoop MapReduce library (for testing Java application) from Apache:

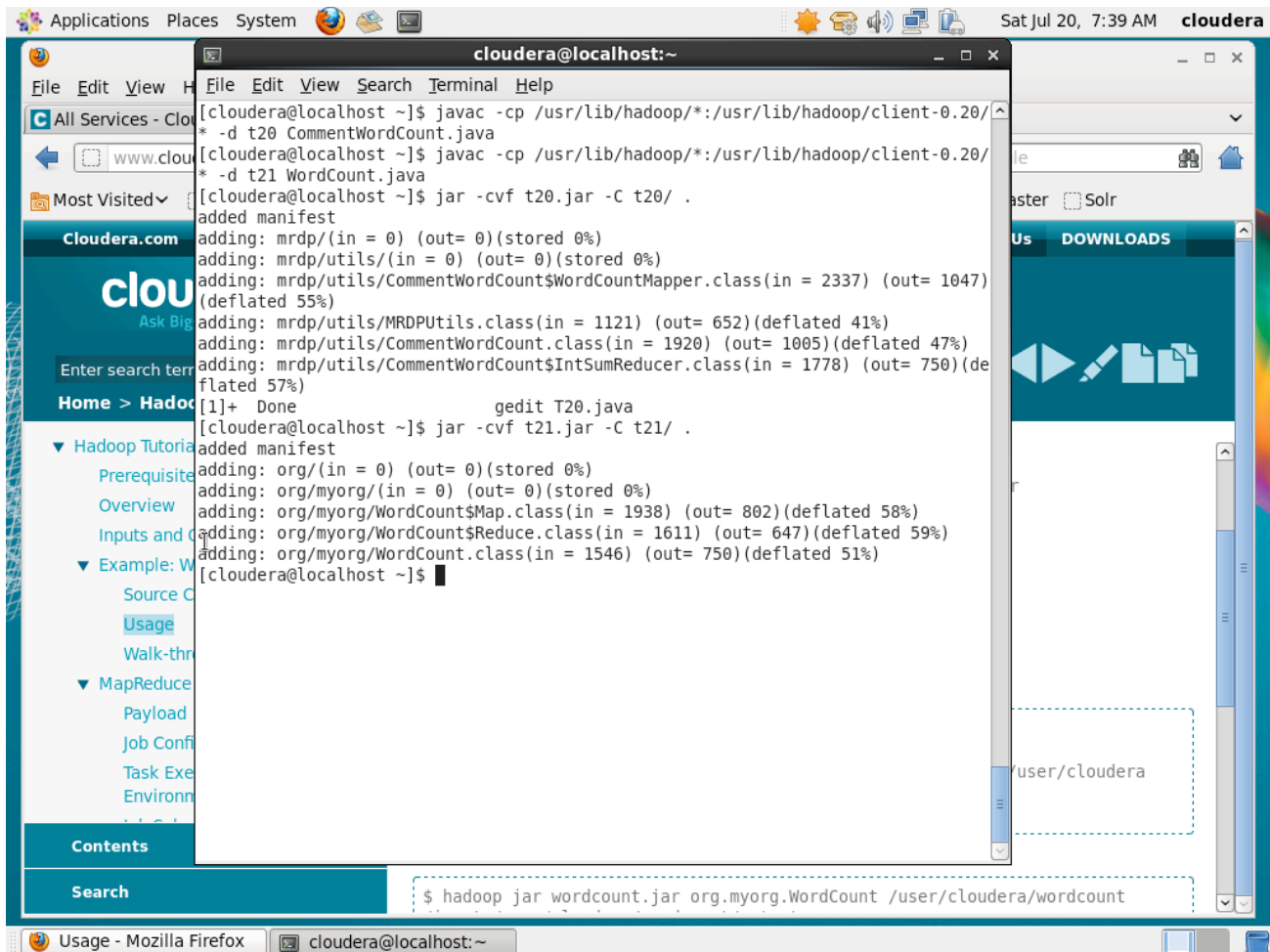
### 3.2 Download testing source code

Ftp to linux.scudc.scu.edu, and download source code from /users/mwang2/test/coen296/:



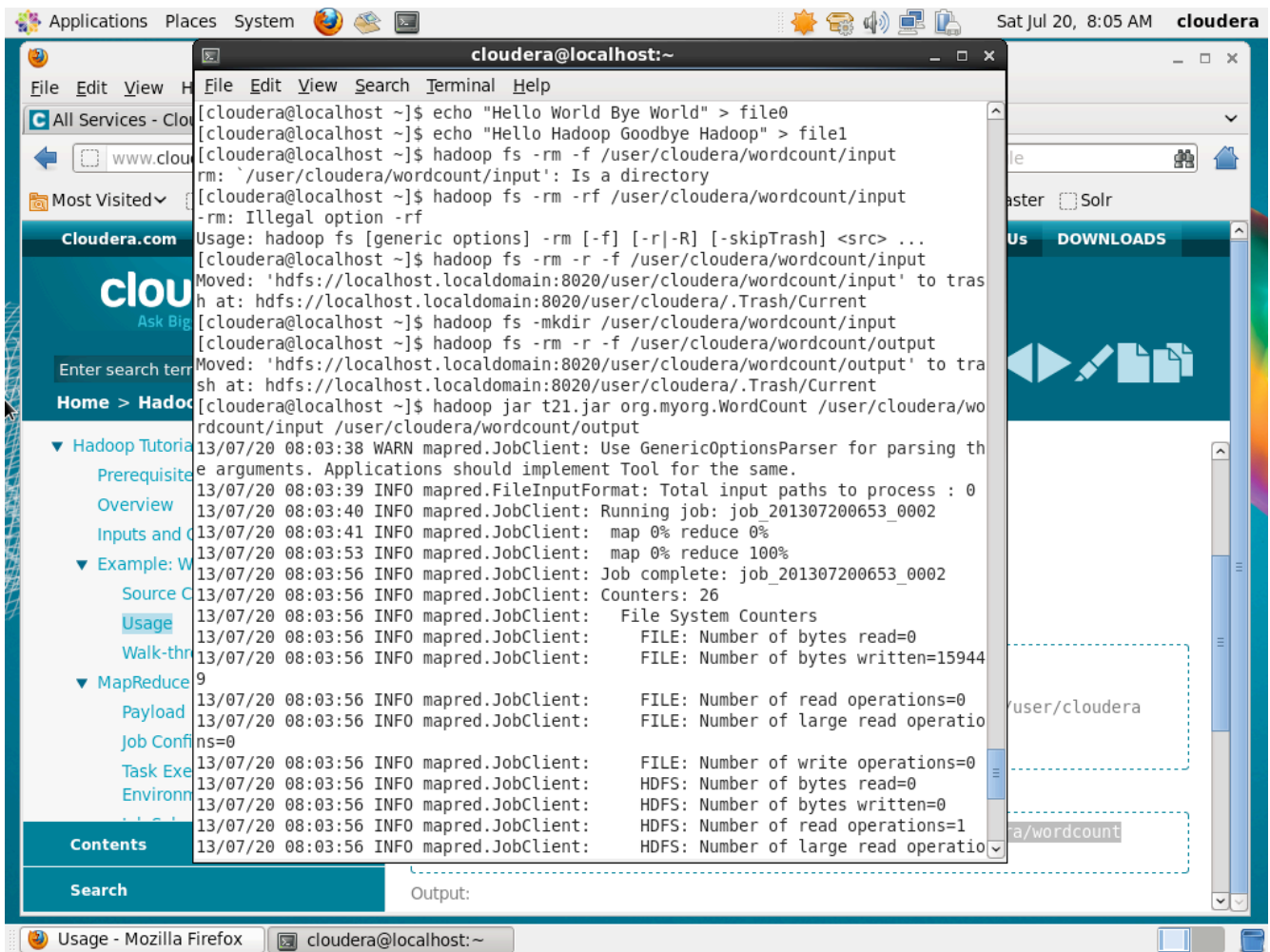
```
cloudera@localhost:~$ clear
[cloudera@localhost ~]$ ftp linux.scudc.scu.edu
Trying 129.210.16.83...
Connected to linux.scudc.scu.edu (129.210.16.83).
220 (vsFTPd 2.2.2)
Name (linux.scudc.scu.edu:cloudera): she
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd ../mwang2/test/coen296
250 Directory successfully changed.
ftp> mget t2*.dat
mget t20.dat? y
227 Entering Passive Mode (129,210,16,83,162,7)
150 Opening BINARY mode data connection for t20.dat (4050 bytes).
226 Transfer complete.
4050 bytes received in 0.0528 secs (76.75 Kbytes/sec)
mget t21.dat? y
227 Entering Passive Mode (129,210,16,83,93,42)
150 Opening BINARY mode data connection for t21.dat (1911 bytes).
226 Transfer complete.
1911 bytes received in 0.000791 secs (2415.93 Kbytes/sec)
ftp> bye
221 Goodbye.
[cloudera@localhost ~]$ vi t20.dat
[cloudera@localhost ~]$ vi t21.dat
[cloudera@localhost ~]$
```

Read the source code, and correct the errors in the source code. And rename the files to major class names. Then compile the test java applications and create the .jar packages:



### 3.3 Prepare the test data and input/output folder in HDFS

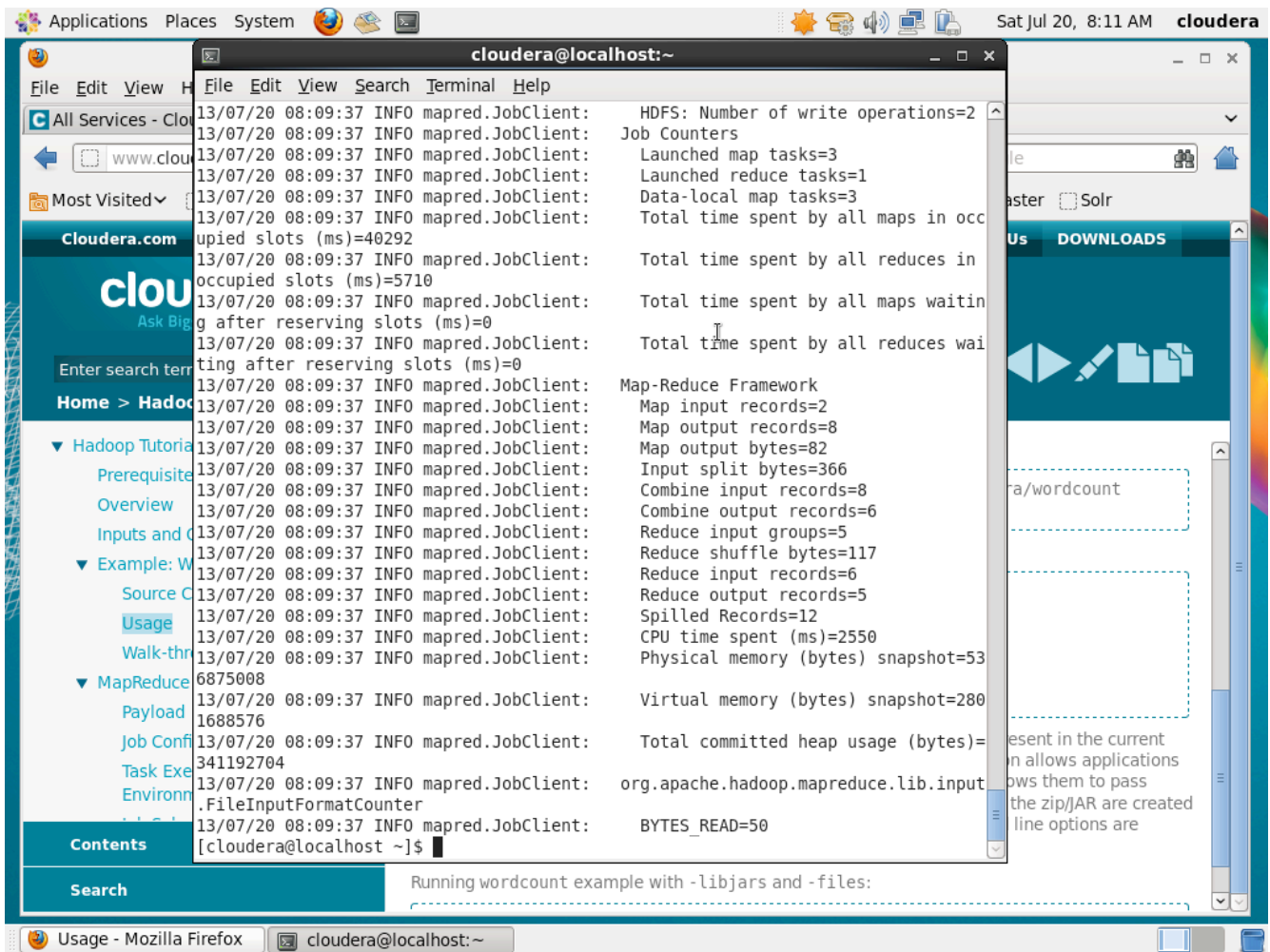
Follow the tutorial, create two input text files, file0 and file1, and upload them into the Hadoop HDFS file system. Since I had run the test before, I need to remove the existing input and output folders first.



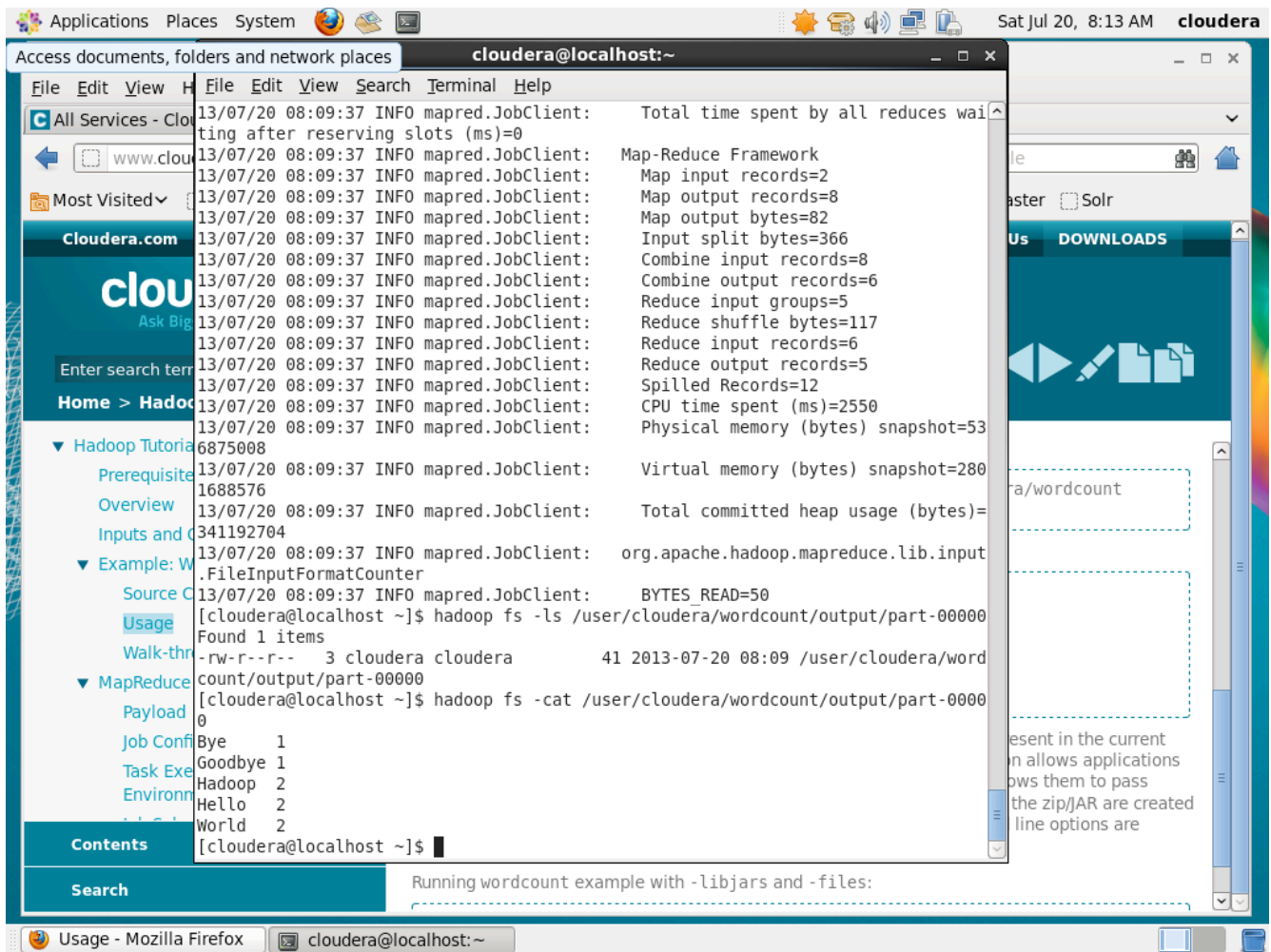
(In the picture above, I forgot upload the test files to HDFS before running the test. You can see the read=0 and definitely the output is nothing. I corrected the issue in the following operations and please refer to the following screen shots.)

### 3.4 Run test case t21.dat (successful)

Run the test application t21.dat (the sample Java test code from the Hadoop Tutorial, the famous WordCount.java) on Hadoop in the terminal. The Hadoop platform prints out the verbose information about how the job is divided and dispatched. Since I run Hadoop on a single node cluster, the system only emulate the task divide and dispatch and actually all tasks are run on the same node.

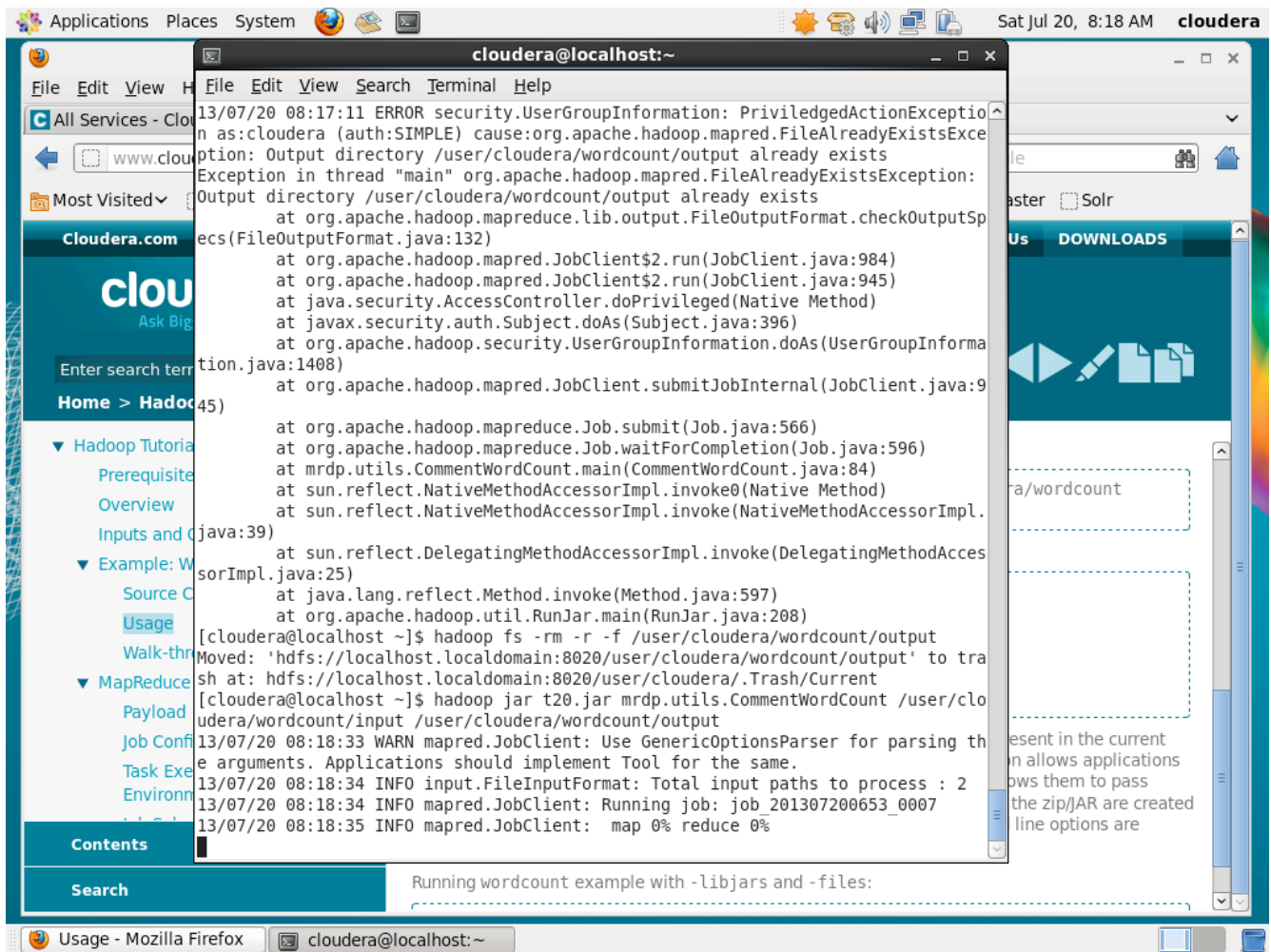


After finished, check the contents in the output files and found they are correct:

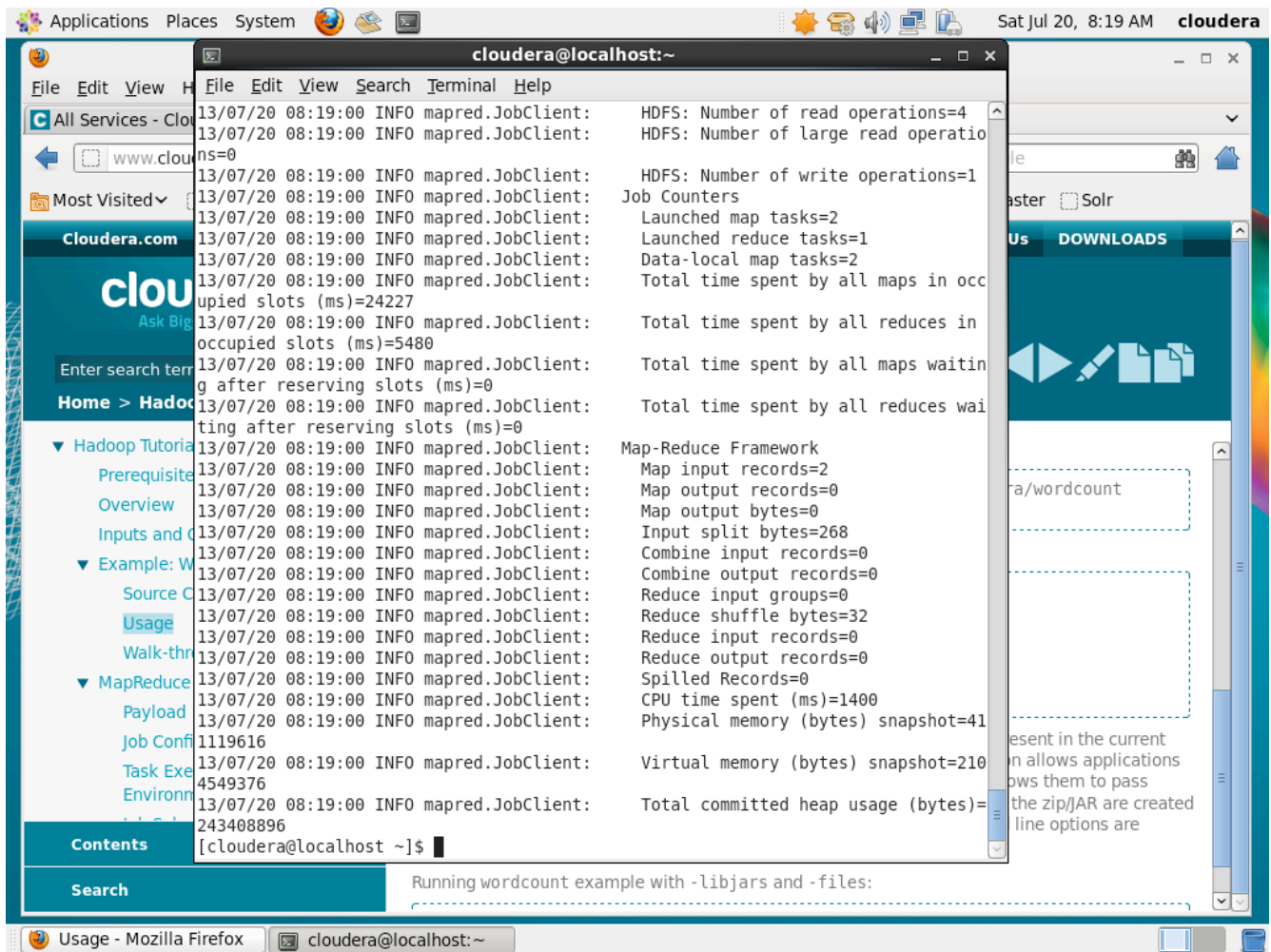


### 3.5 Run test case t20.dat (no output)

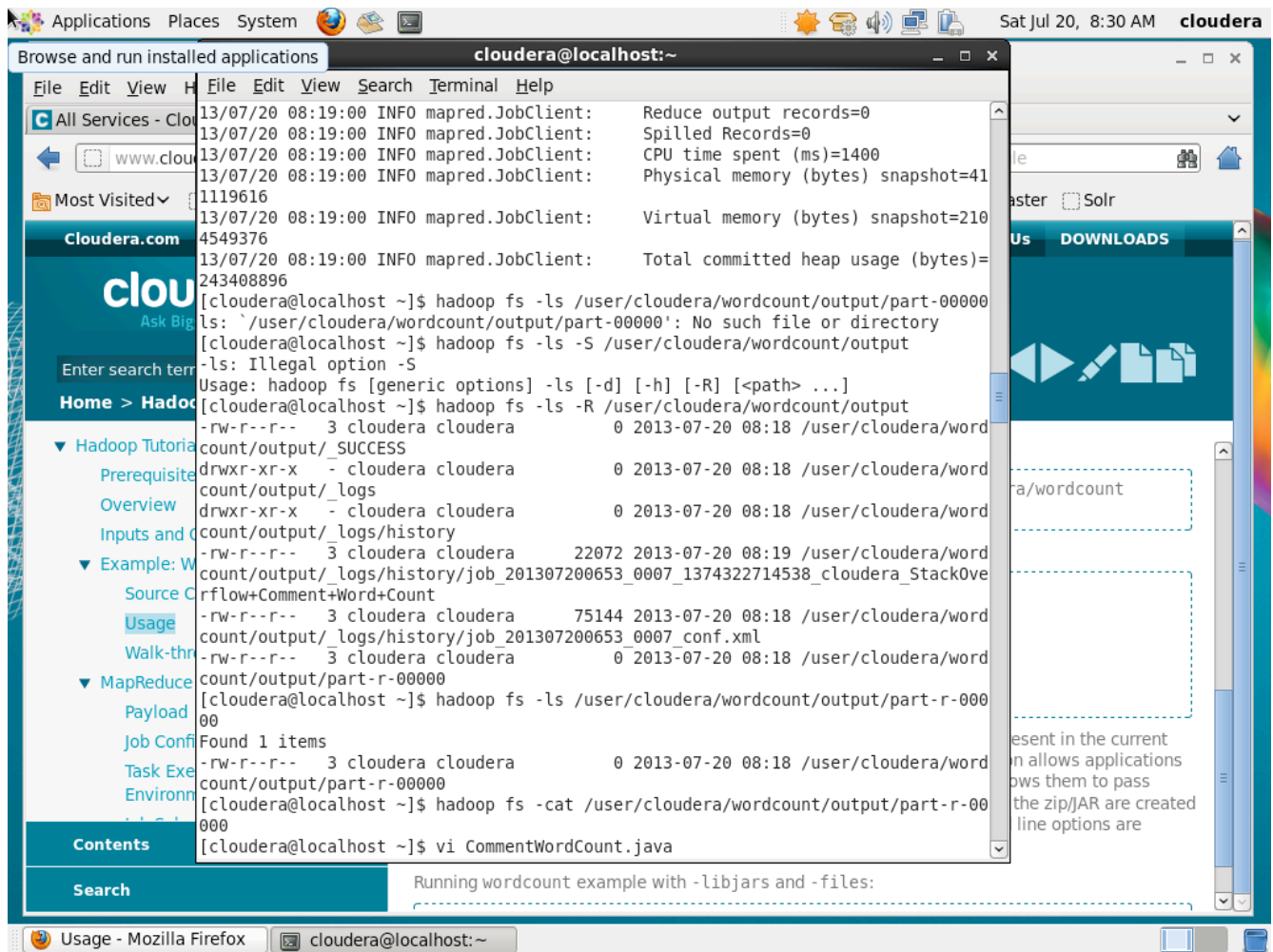
Run the test application t20.dat (The CommentWordCount.java provided by the professor) in the same way. Need to clear the output folder before running the application. Or, will see the “folder exist” exception.



The application ran successfully. But there is no output. Am I wrong in any stage?

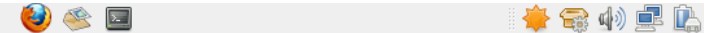


I tried to print the output file, but there is nothing inside. Am I wrong?



Checked the application source code and the Hadoop logs, and could not find out any clue about the issue.



Applications Places System  Sat Jul 20, 8:31 AM cloudera

File Edit View H

All Services - Clo

Most Visited

Cloudera.com

Home > Hadoop

- ▼ Hadoop Tutorial
  - Prerequisite
  - Overview
  - Inputs and Outputs
  - ▼ Example: WordCount
    - Source Code
    - Usage
    - Walk-through
  - ▼ MapReduce
    - Payload
    - Job Configuration
    - Task Execution
    - Environment

Contents

Search

cloudera@localhost:~

File Edit View Search Terminal Help

```
[cloudera@localhost ~]$ vi CommentWordCount.java
[cloudera@localhost ~]$ hadoop fs -cat /user/cloudera/wordcount/output/_logs/history/job_201307200653_0007_1374322714538_cloudera_StackOverflow+Comment+Word+Count
Meta VERSION="1" .
Job JOBID="job_201307200653_0007" JOBNAME="StackOverflow Comment Word Count" USER="cloudera" SUBMIT_TIME="1374322714538" JOBCONF="hdfs://localhost.localdomain:8020/user/cloudera/.staging/job_201307200653_0007/job.xml" VIEW_JOB="*" MODIFY_JOB="*" JOB_QUEUE="default" .
Job JOBID="job_201307200653_0007" JOB_PRIORITY="NORMAL" .
Job JOBID="job_201307200653_0007" LAUNCH_TIME="1374322714767" TOTAL_MAPS="2" TOTAL_REDUCES="1" JOB_STATUS="PREP" .
Task TASKID="task_201307200653_0007_m_000003" TASK_TYPE="SETUP" START_TIME="1374322715071" SPLITS="" .
MapAttempt TASK TYPE="SETUP" TASKID="task_201307200653_0007_m_000003" TASK Attempt ID="attempt_201307200653_0007_m_000003_0" START_TIME="1374322715186" TRACKER_NAME="tracker_localhost.localdomain:localhost.localdomain/127.0.0.1:59760" HTTP_PORT="50060" .
MapAttempt TASK TYPE="SETUP" TASKID="task_201307200653_0007_m_000003" TASK Attempt ID="attempt_201307200653_0007_m_000003_0" TASK STATUS="SUCCESS" FINISH_TIME="1374322718613" HOSTNAME="/default/localhost.localdomain" STATE STRING="setup" COUNTERS="{(org.apache.hadoop.mapreduce.FileSystemCounter)(File System Counter)[(FILE_BYTES_READ)(FILE: Number of bytes read)(0)][(FILE_BYTES_WRITTEN)(FILE: Number of bytes written)(160127)][(FILE_READ_OPS)(FILE: Number of read operations)(0)][(FILE_LARGE_READ_OPS)(FILE: Number of large read operations)(0)][(FILE_WRITE_OPS)(FILE: Number of write operations)(0)][(HDFS_BYTES_READ)(HDFS: Number of bytes read)(0)][(HDFS_BYTES_WRITTEN)(HDFS: Number of bytes written)(0)][(HDFS_READ_OPS)(HDFS: Number of read operations)(0)][(HDFS_LARGE_READ_OPS)(HDFS: Number of large read operations)(0)][(HDFS_WRITE_OPS)(HDFS: Number of write operations)(1)]}{(org.apache.hadoop.mapreduce.TaskCounter)(Map-Reduce Framework)[(SPILLED_RECORDS)(Spilled Records)(0)][(CPU_MILLISECONDS)(CPU time spent \\\(ms\\))(100)][(PHYSICAL_MEMORY_BYTES)(Physical memory \\\(bytes\\) snapshot)(88887296)][(VIRTUAL_MEMORY_BYTES)(Virtual memory \\\(bytes\\) snapshot)(688623616)][(COMMITTED_HEAP_BYTES)(Total committed heap usage \\\(bytes\\))(47841280)]}nullnullnullnullnullnullnullnullnullnull" .
Task TASKID="task_201307200653_0007_m_000003" TASK_TYPE="SETUP" TASK_STATUS="SUC
```

www.cloudera.com

Downloads

ra/wordcount

present in the current  
n allows applications  
ows them to pass  
the zip/JAR are created  
line options are

Running wordcount example with -libjars and -files: