Assignment 1

BUAN 6346

Big Data Analytics

Dr. Jerry F. Perez

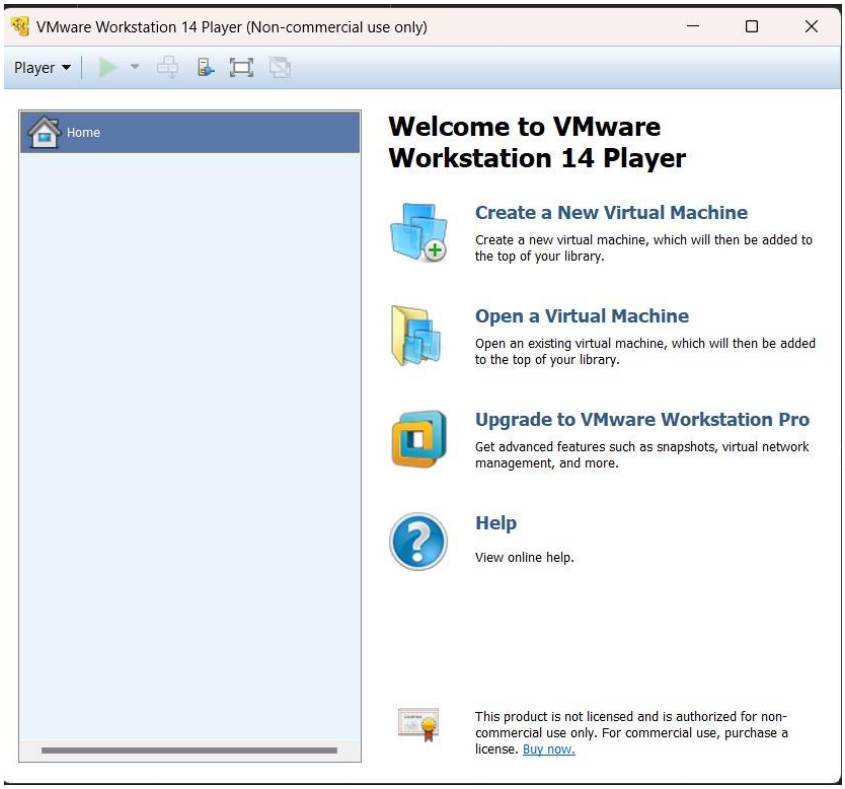
Yug Jyotirmay Singh
YJS220000

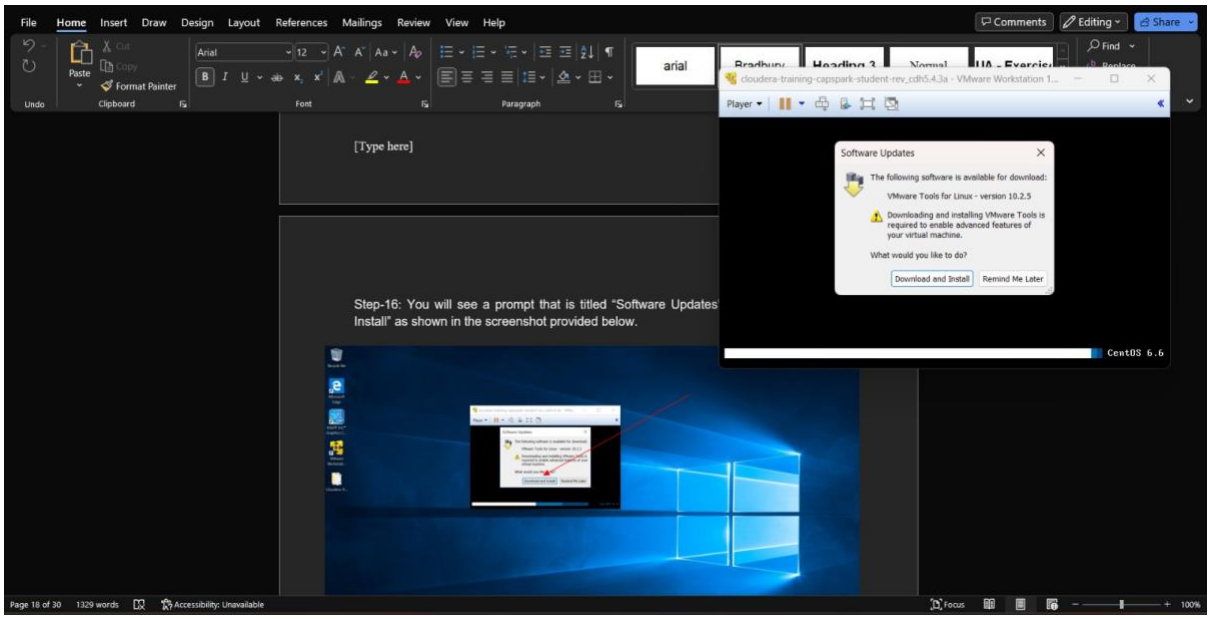
Table of Contents

Chapter 2

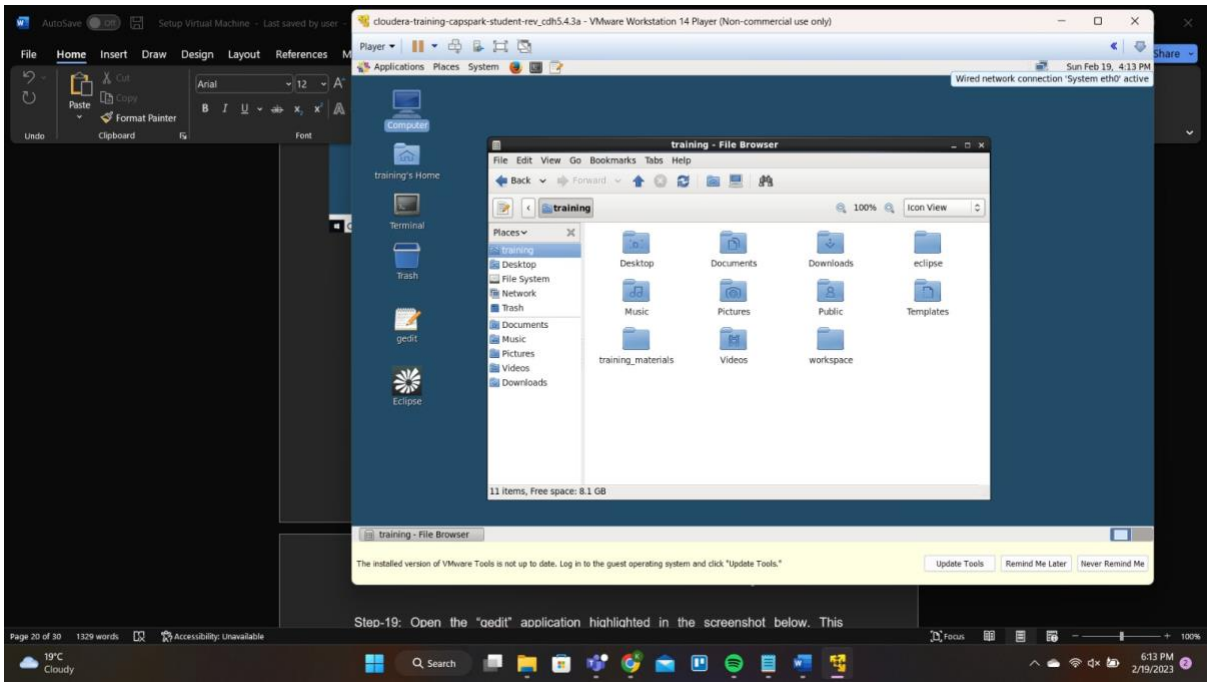
2.1 Getting Started

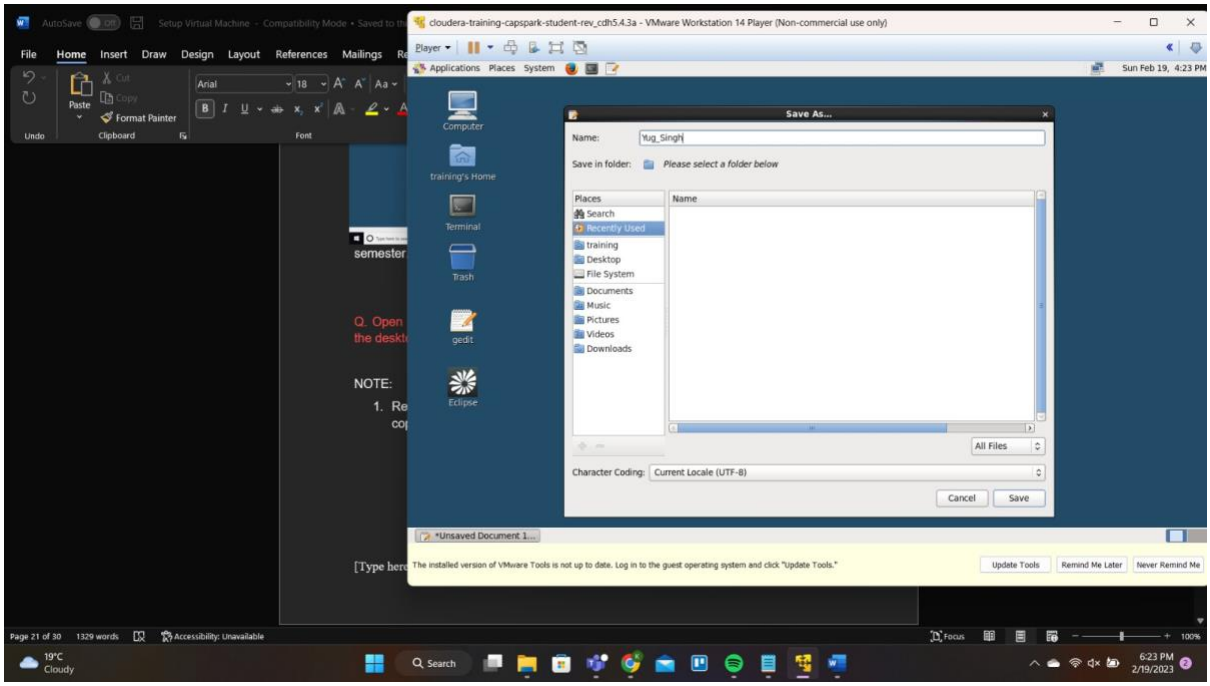
We are now setting up the VM and observing the working of the VM.

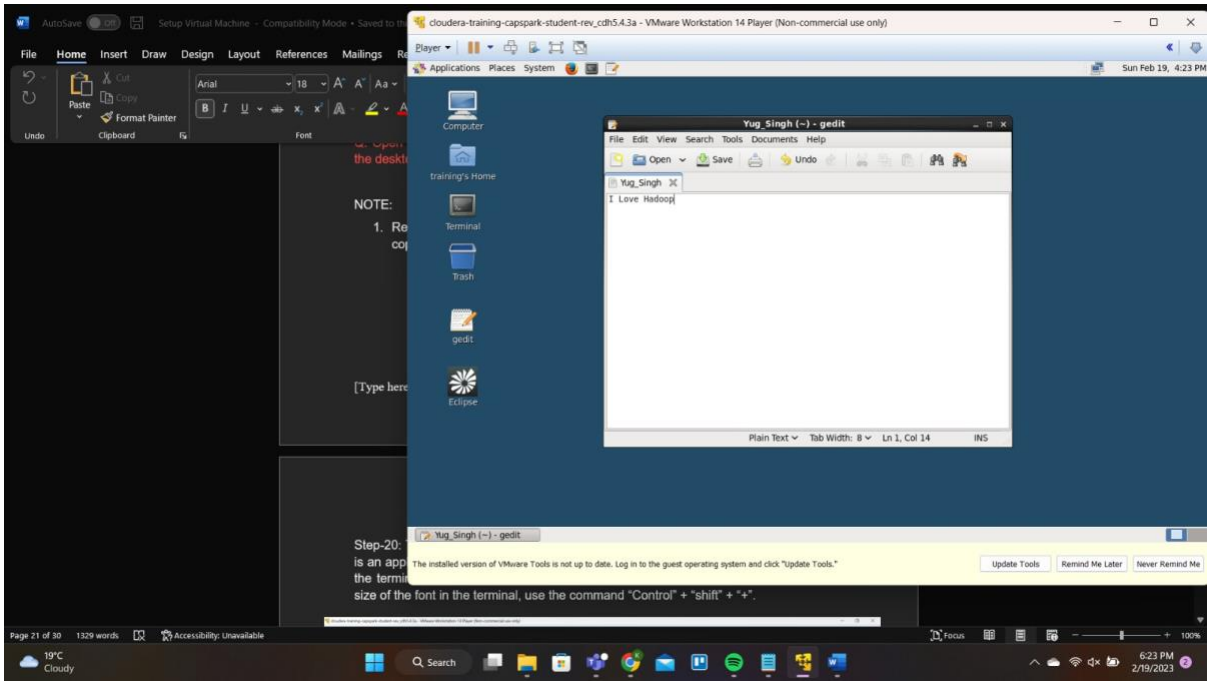


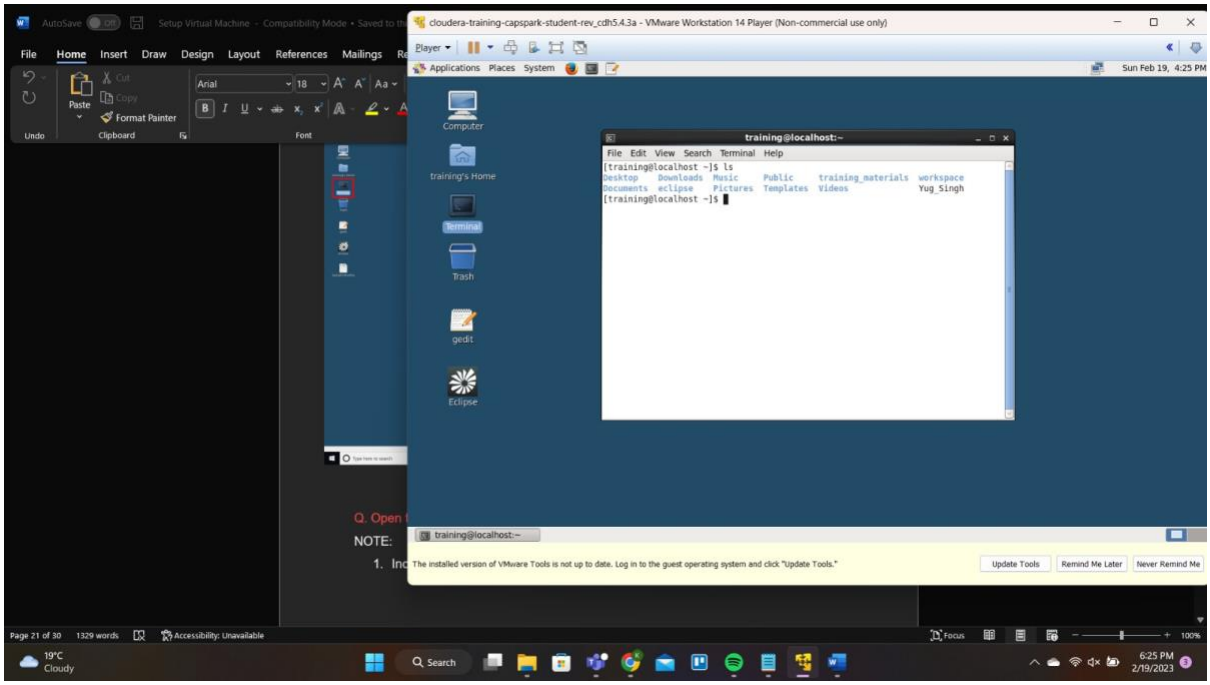


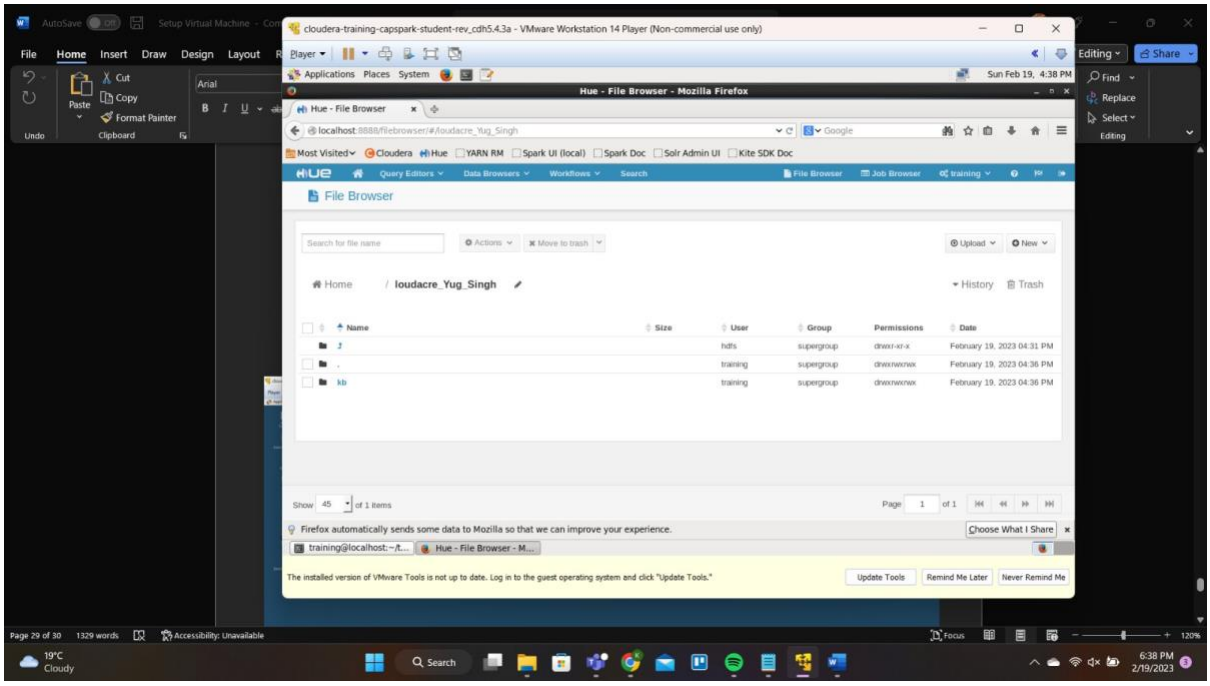
We will upload the files to HDFS before working with them.


cloudera-training-capspark-student-rev_cdh5.4.3a - VMware Workstation 14 Player (Non-commercial use only)
training - File Browser
Desktop
Documents
Downloads
eclipse
Music
Pictures
Public
Templates
training_materials
Videos
workspace
11 items, Free space: 8.1 GB
Wired network connection 'System eth0' active
Sun Feb 19, 4:13 PM
Update Tools
Remind Me Later
Never Remind Me
Step-19: Open the "gedit" application highlighted in the screenshot below. This


cloudera-training-capspark-student-rev_cdh5.4.3a - VMware Workstation 14 Player (Non-commercial use only)
Save As...
Name: Yug_Singh
Save in folder: Please select a folder below
Character Coding: Current Locale (UTF-8)
Cancel
Save
Sun Feb 19, 4:23 PM
Update Tools
Remind Me Later
Never Remind Me
NOTE:


cloudera-training-capspark-student-rev_cdh5.4.3a - VMware Workstation 14 Player (Non-commercial use only)
Yug_Singh (~) - gedit
I Love Hadoop
Step-20:
size of the font in the terminal, use the command "Control" + "shift" + "+".
NOTE:

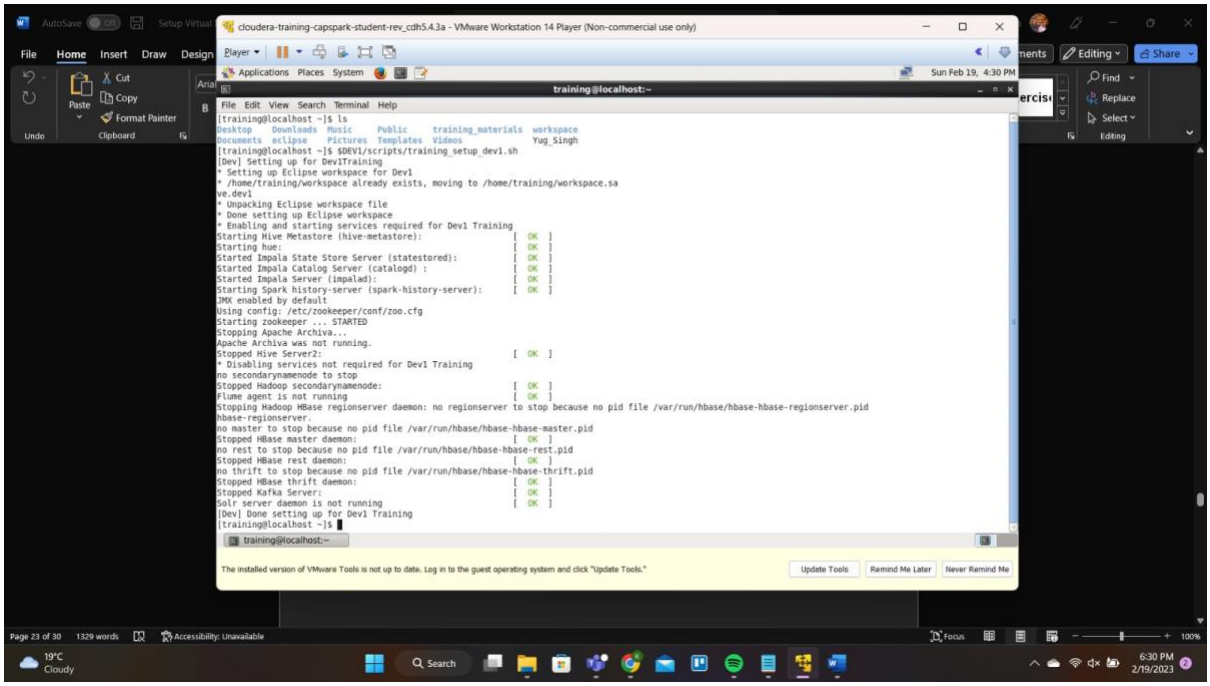

cloudera-training-capspark-student-rev_cdh5.4.3a - VMware Workstation 14 Player (Non-commercial use only)
training@localhost:~
[training@localhost ~]$ ls
Desktop Downloads Music Public training_materials workspace
Documents eclipse Pictures Templates Videos Yug_Singh
[training@localhost ~]$
NOTE:

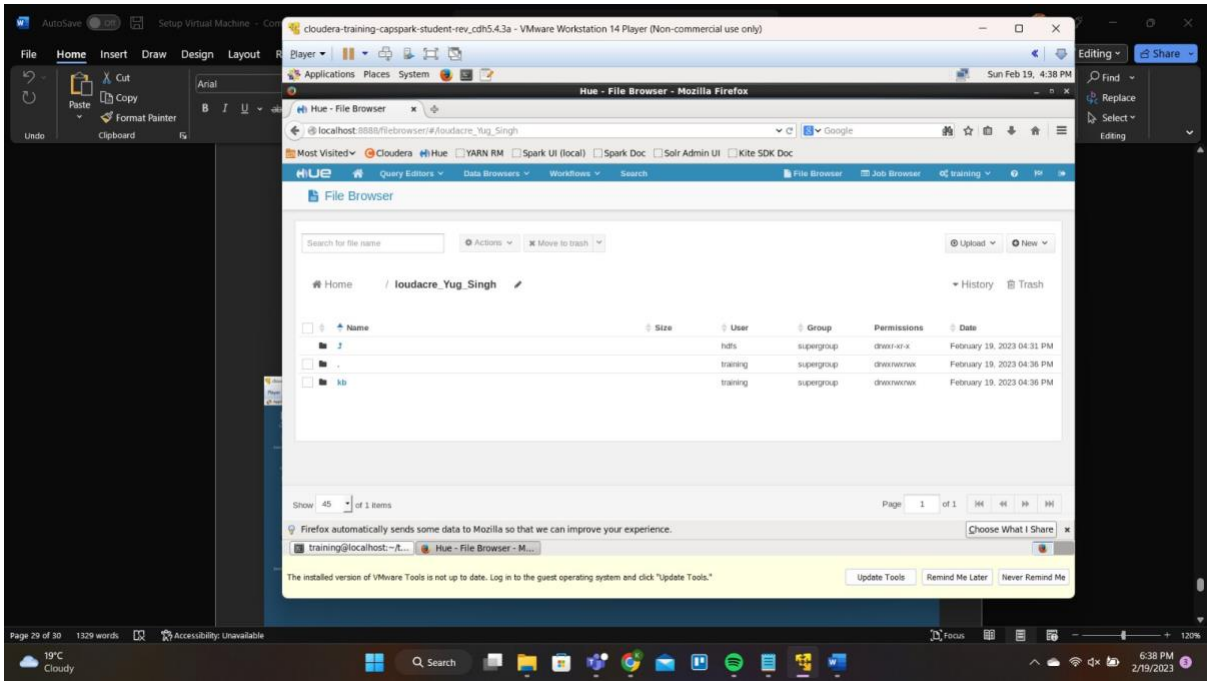


2.2 Run the setup script

The initial task to do here is to set up the environment, and we run the setup script.

$DEV1/scripts/training_setup_dev1.sh





Chapter 3

3.1 Explore the HDFS Command Line Interface

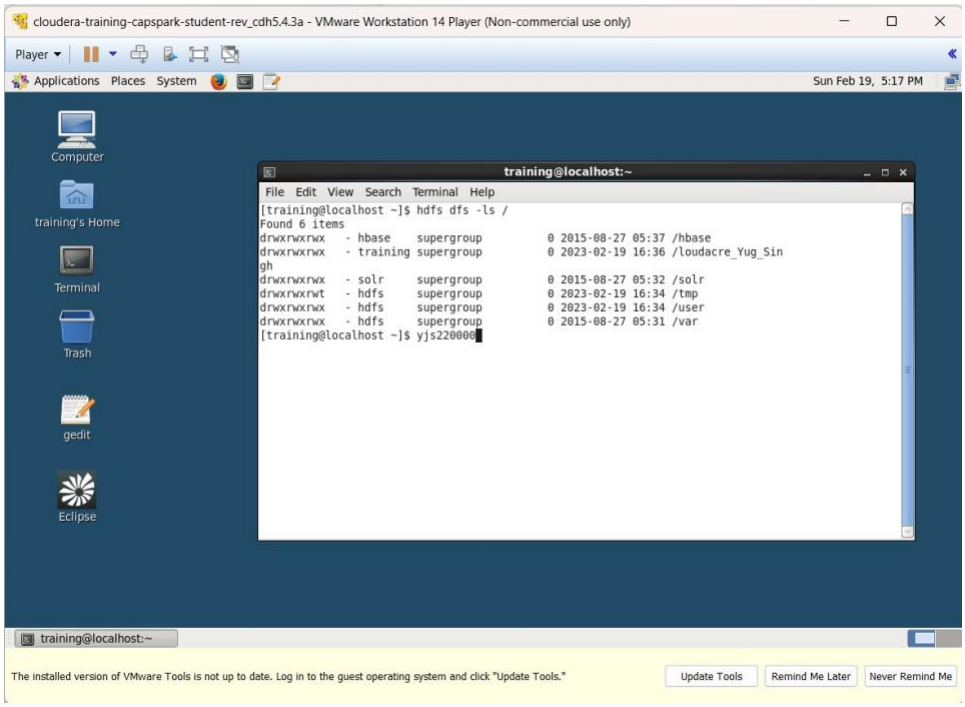
The easiest way to interact with HDFS is using the *hdfs* command.

To execute file system commands within HDFS, use the *hdfs dfs* command.

Use the following steps to explore HDFS CLI:

1.Execute the following command in terminal

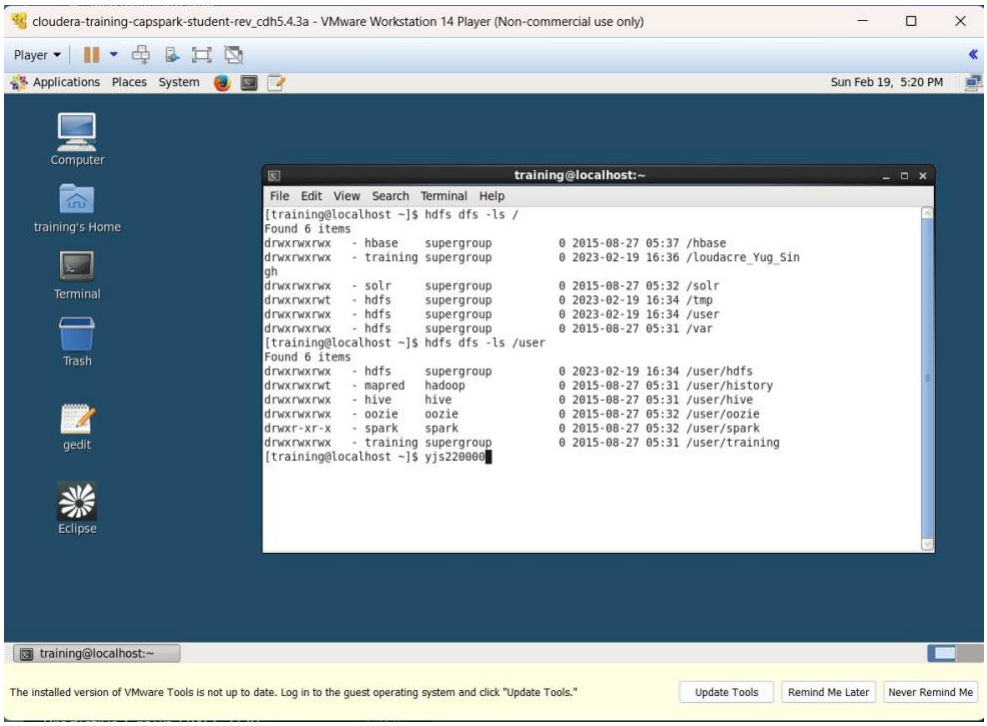
```
$hdfs dfs -ls /
```



We observe that the directory has 5 items.

2.Viewing the contents of the /user directory by running.

$ hdfs dfs -ls /user



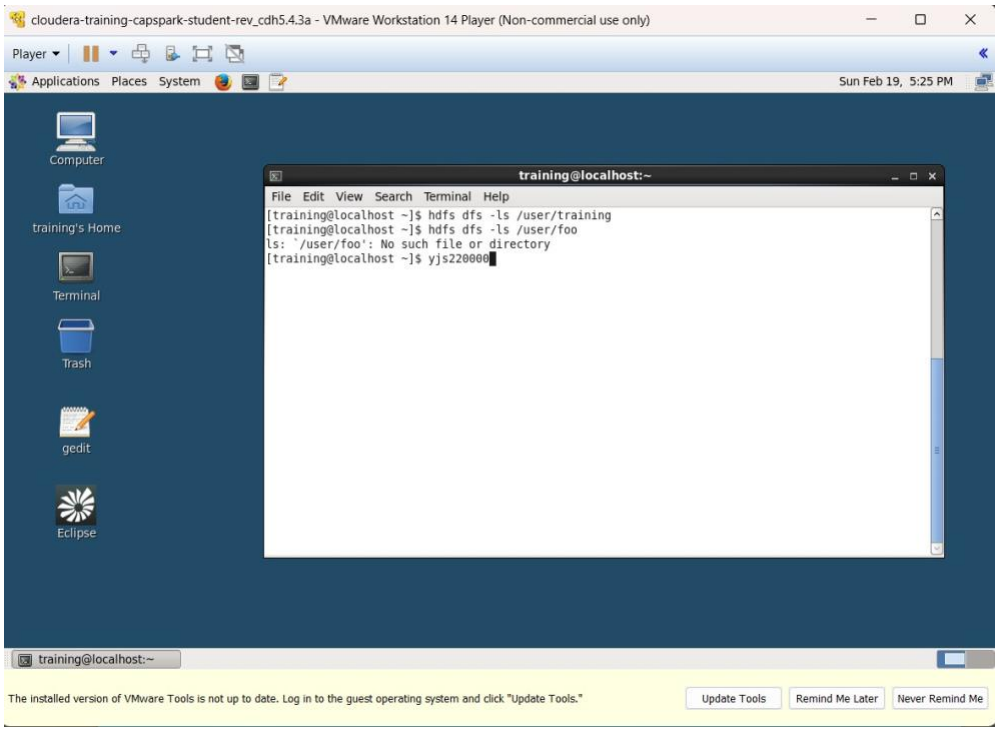
We observe that the user directory has 5 times.

3.Viewing the contents of the home directory by running:

$ hdfs dfs -ls /user/training

I also tried to look into a non-existent folder which displayed an error message.

$ hdfs dfs -ls /user/foo



3.2 Upload Files to HDFS

Besides browsing the existing filesystem, another essential thing you can do with the HDFS command-line interface is to upload new data into HDFS.

Use the following steps to achieve this:

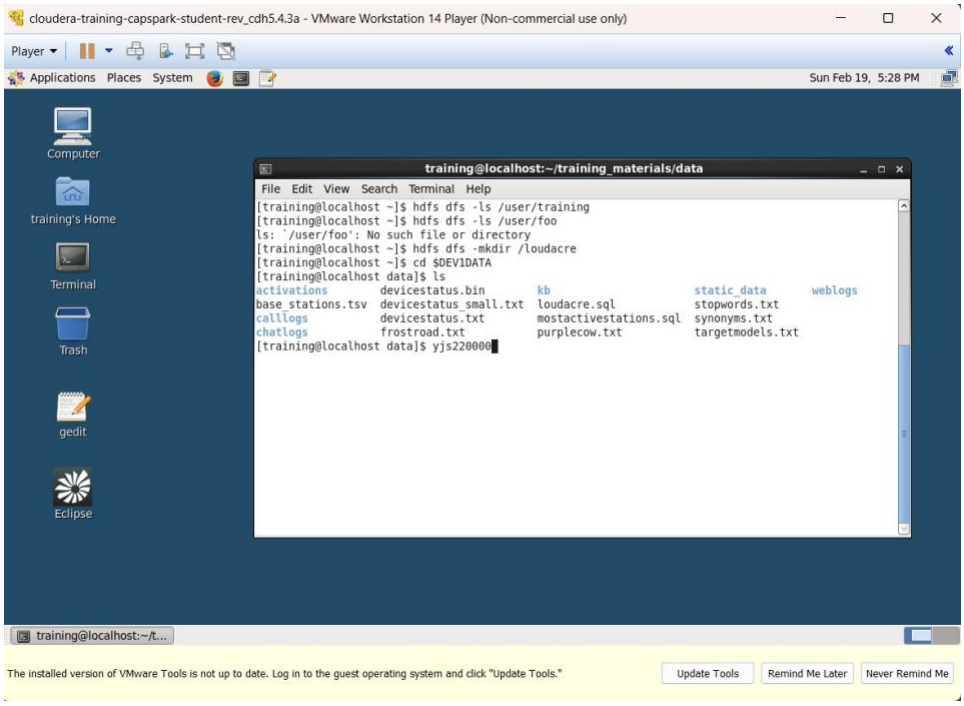
1. Start by creating a new top-level directory for homework assignments, will use this directory throughout the rest of the course.

$ hdfs dfs -mkdir /loudacre

2. Start by creating a new top-level directory for homework assignments, will use this directory throughout the rest of the course.

$cd $DEV1DATA

$ ls

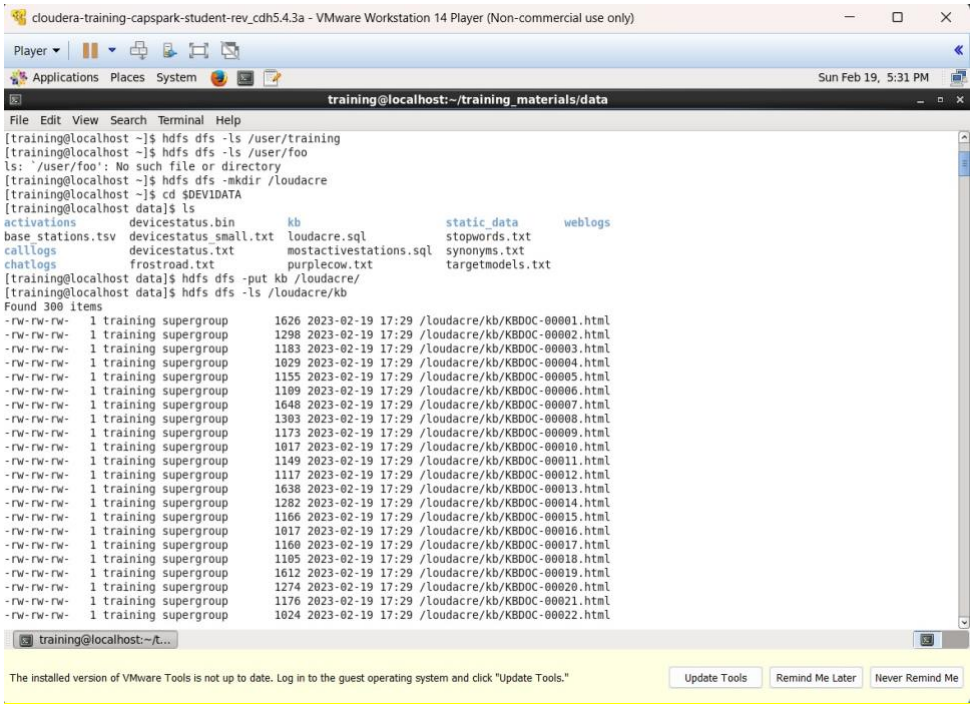


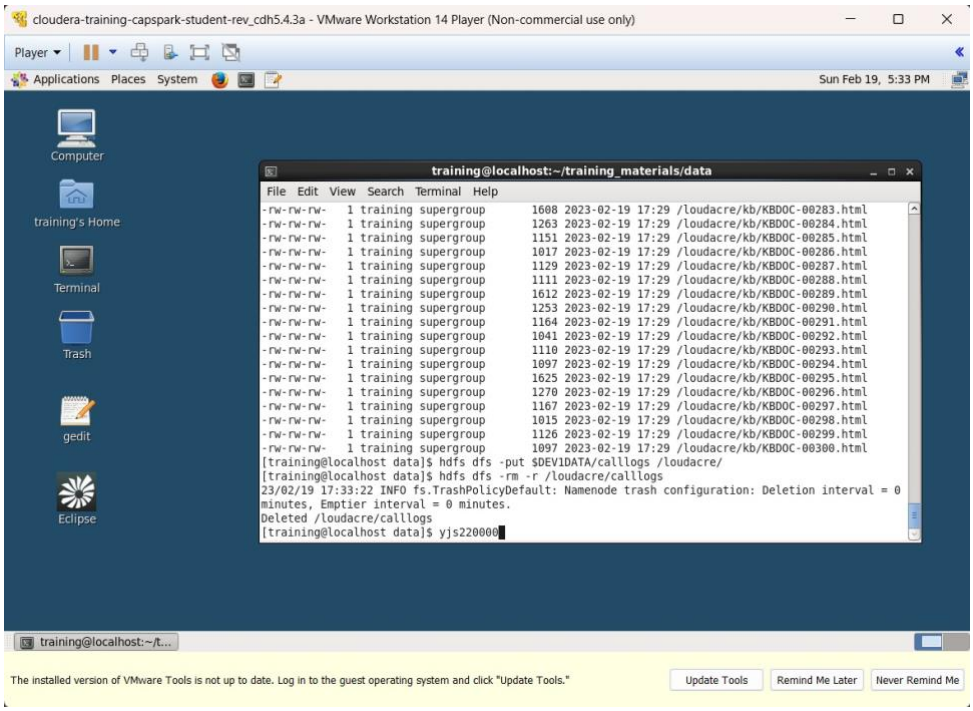
3. Insert this directory into HDFS. This copies the local kb directory and its contents into a remote HDFS directory named /loudacre/kb

$ hdfs dfs -put kb /loudacre/

4. Listing the files in the new HDFS directory.

$ hdfs dfs -ls /loudacre/kb

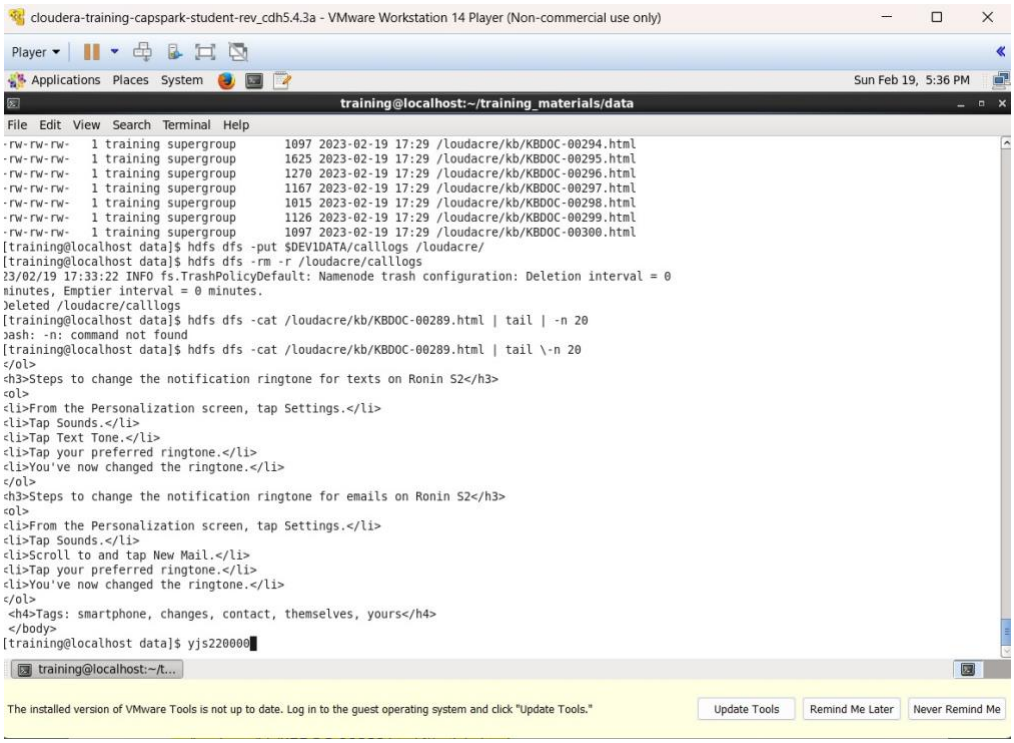




5. Uploading a directory, then removing it, as it is not needed for future exercises.

$ hdfs dfs -put $DEV1DATA/calllogs /loudacre/

$ hdfs dfs -rm -r /loudacre/calllogs

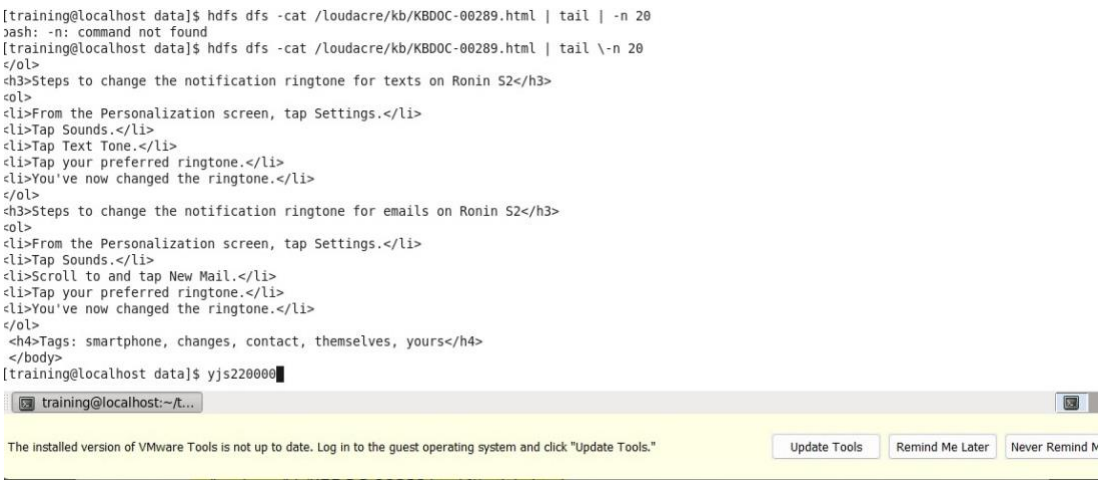


3.3 View HDFS files

Now, let's view some of the data we just copied into HDFS.

Use the following steps to achieve this:

1. The code is:
$ hdfs dfs -cat /loudacre/kb/KBDOC-00289.html | tail \ -n 20

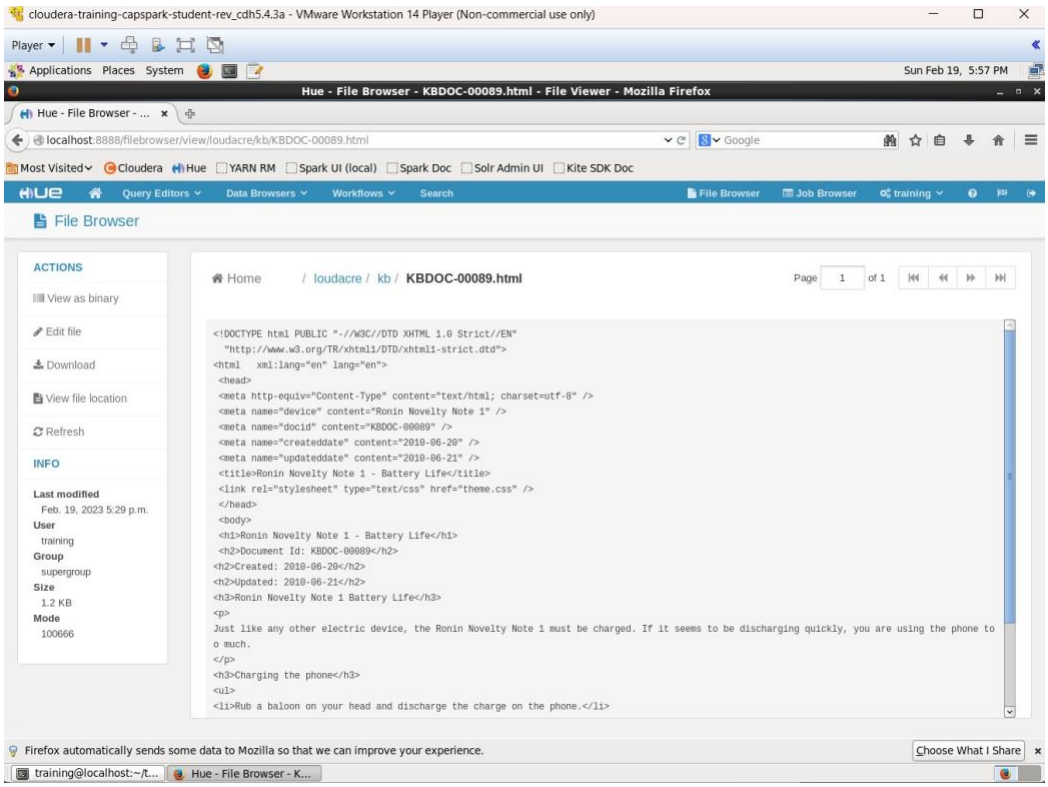


NOTE: This prints the last 20 lines of the article to your terminal. This command is handy for viewing HDFS data. An individual file is often huge, making it inconvenient to view the entire file in the terminal. For this reason,

it's often a good idea to pipe the output of the fs -cat command into the head, tail, more, or less.

2. You can use the hdfs dfs -get command to retrieve a local copy of a file or directory from HDFS. This command takes two arguments: an HDFS path and a local path. It copies the HDFS contents into the local filesystem:

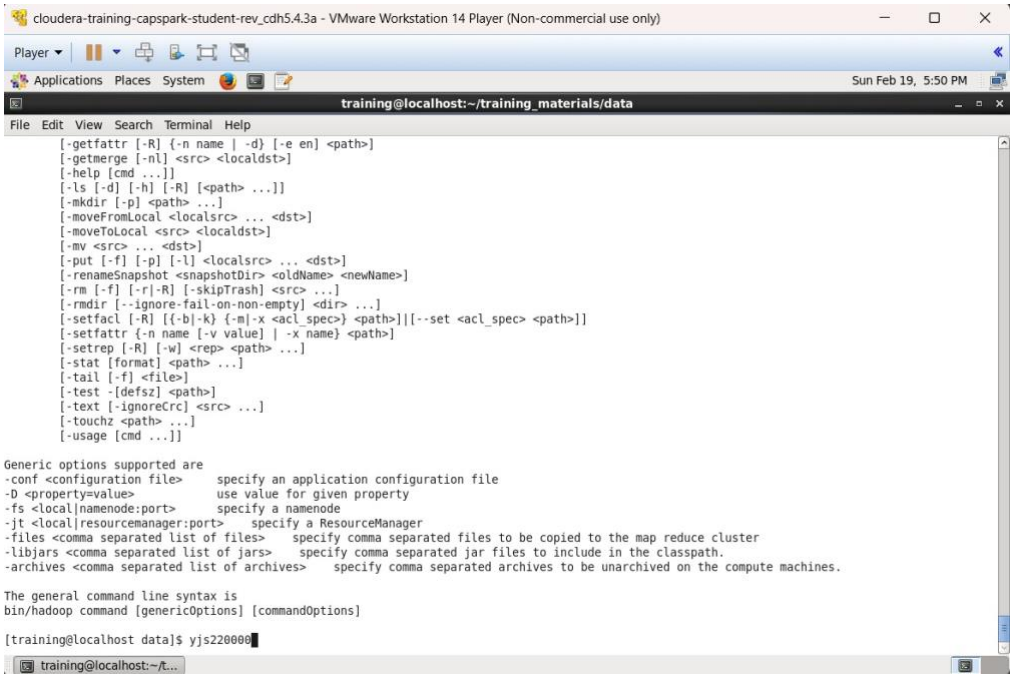
```
$ hdfs dfs -get \
/loudacre/kb/KBDOC-00289.html ~/article.html
$ less ~/article.html
```



3.Several other operations are available with the hdfs dfs command to perform the most common filesystem manipulations: mv, cp, mkdir, etc.

The following command provides help:

```
$hdfs dfs
```



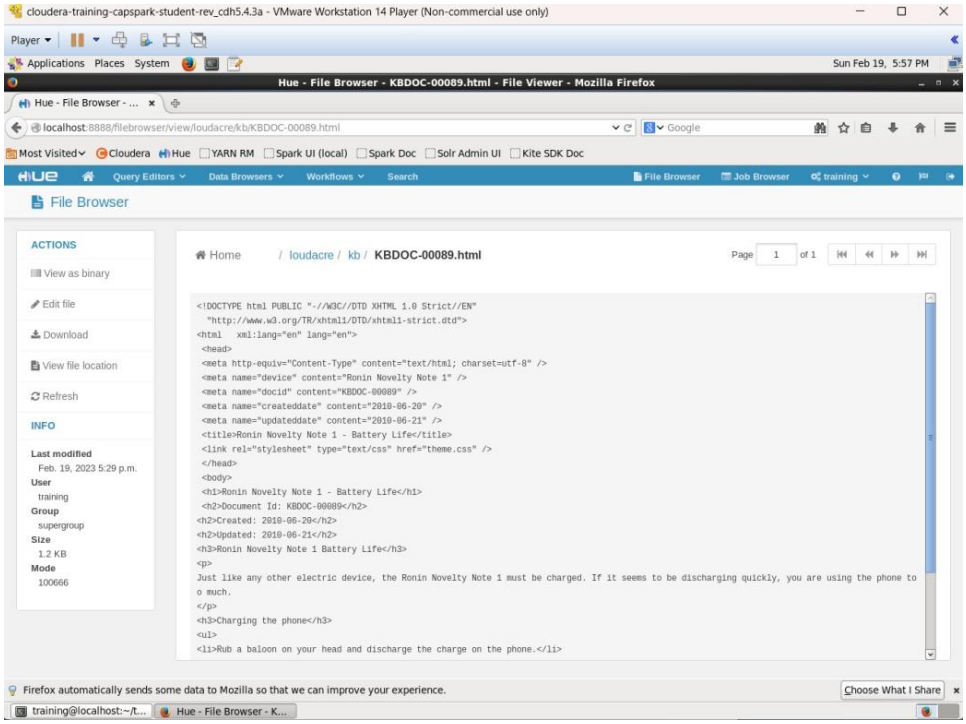
3.4 Use the Hue File Browser to browse, view and manage files

Start Firefox on the VM by clicking its icon on the main menu panel at the top of the screen. Click the Hue bookmark, or visit http://localhost:8888

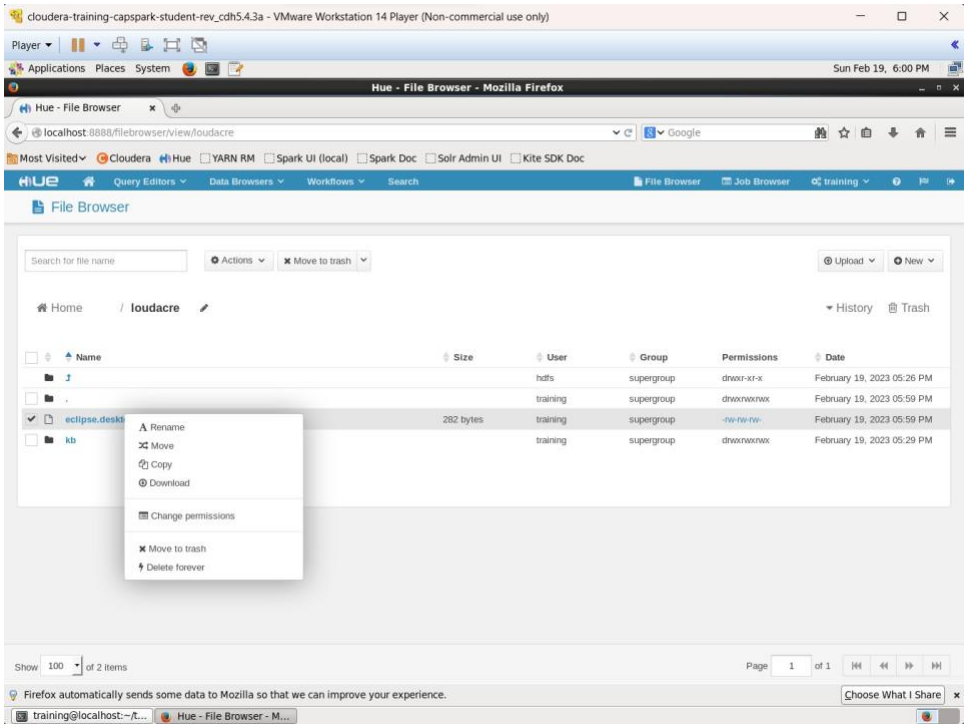
Because this is the first time anyone has logged into Hue on this server, you will be prompted to create a new user account. Enter username training and password training, then click Create Account. (If started, you may click "Remember Password")

• Note: You may see a misconfiguration warning when you first log in to Hue. This is because not all the services Hue depends on are running on the course VM. You can disregard the message.

-Hue has many useful features, many of which will be covered later in the course. To access HDFS, click File Browser in the Hue menu bar.



-We can see that the file is modified at a timestamp that matches to our command time.



- The file actions that we can see are shown as above.

Note: Using the upload option, we can upload the files into the file system, and I uploaded the eclipse.desktop file using the upload option.

Chapter 4

The focus of this exercise is not on what the application does, but on how YARN distributes tasks in a job across a cluster, and how to monitor an application and view its log files.

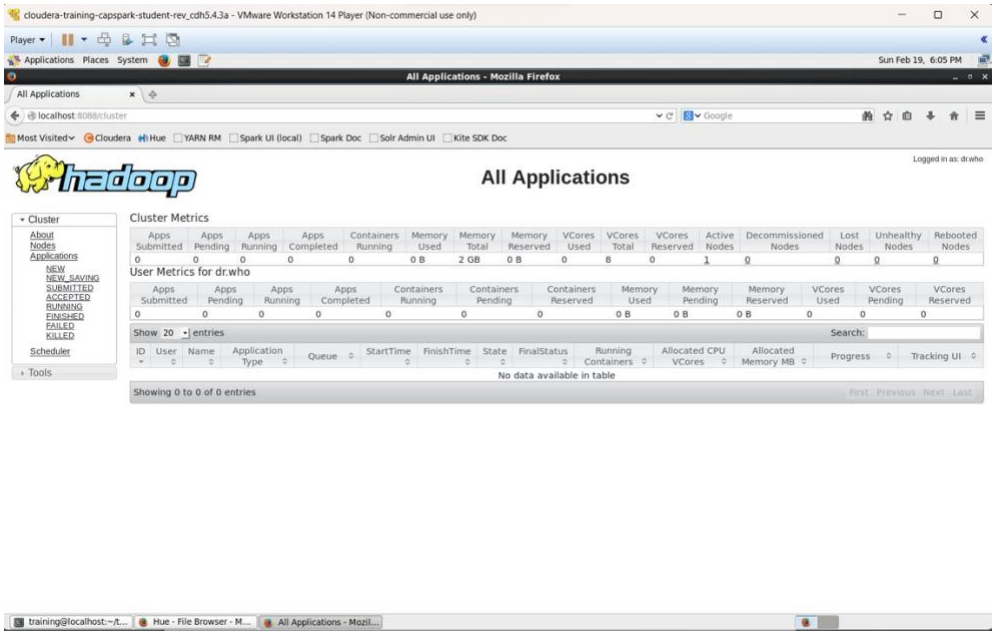
Files and Data Used in this Homework

Exercise directory:

$DEV1/exercises/yarn Dataset (HDFS): /loudacre/kb

4.1 Explore the YARD Cluster

1.Visit the YARN Resource Manager (RM) UI in Firefox using the provided bookmark, or by going to URL *http://localhost:8088*



-No jobs are currently running so the current view shows the cluster “at rest”.

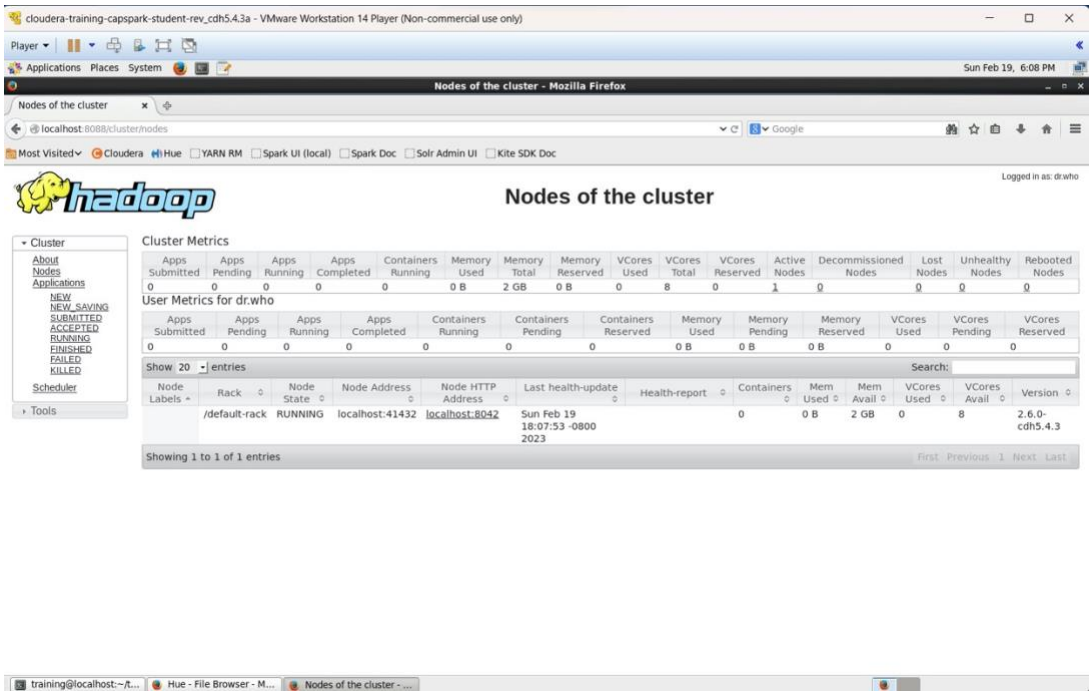
Who is Dr. Who?

You may notice that YARN says you are logged in as dr.who. This is what is displayed when user authentication is disabled for the cluster, as it is on

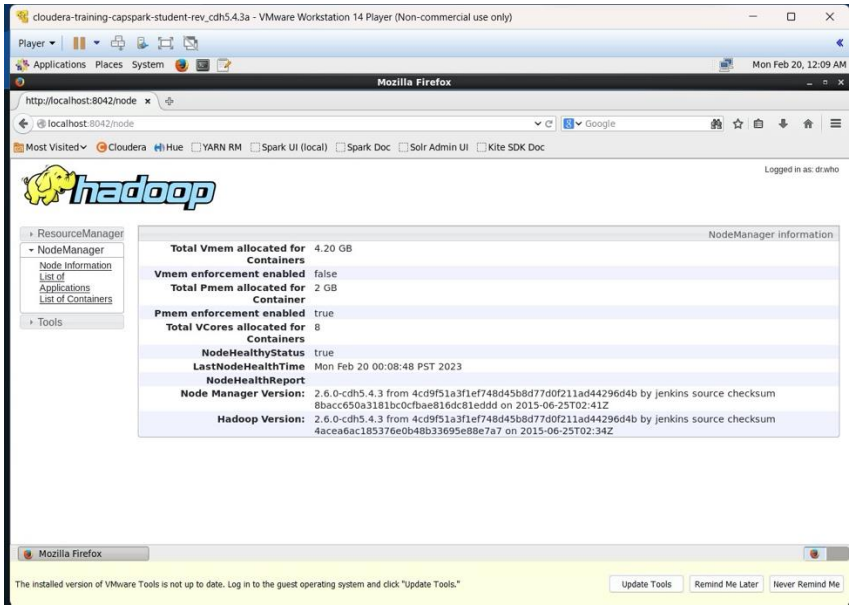
the training VM. If user authentication were enabled, you would have to log in as a valid user to view the YARN UI, and your actual user name would be displayed, together with user metrics such as how many applications you had run, how much system resources your applications used and so on.

We see that total memory is 2GB, vcores 8 and active nodes 1.
-Vcores are virtual cores, it is a share of host CPU that the YARN Node Manager allocates to available resources

2. Click on the Nodes link in the Cluster menu on the left. The bottom section will display a list of worker nodes in the cluster. The pseudo-distributed cluster used for training has only a single node, which is running on the local machine. In the real world, this list would show multiple worker nodes.



3.Click on the Node HTTP Address to open the Node Manager UI on that specific node. This displays statistics about the selected node, including amount of available memory, currently running applications (none, currently) and so on.



4.To return to the Resource Manager, expand ResourceManager on the left and press RM Home.

4.2 Submit an Application to the YARN Cluster

1. In a terminal window, change to the exercise directory:

```
$ cd $DEV1/exercises/yarn
```

2.Run the example wordcount.py program on the YARN cluster to count the frequency of words in the knowledge dataset:

```
$ spark-submit --master yarn-cluster \ wordcount.py /loudacre/kb/*
```

```
[training@localhost yarn]$ spark-submit --master yarn-cluster \ wordcount.py /loudacre/kb/*
Exception in thread "main" java.net.URISyntaxException: Illegal character in path at index 0:  wordco
        at java.net.URI$Parser.fail(URI.java:2829)
        at java.net.URI$Parser.checkChars(URI.java:3002)
        at java.net.URI$Parser.parseHierarchical(URI.java:3086)
        at java.net.URI$Parser.parse(URI.java:3044)
        at java.net.URI.<init>(URI.java:595)
        at org.apache.spark.util.Utils$.resolveURI(Utils.scala:1663)
        at org.apache.spark.deploy.SparkSubmitArguments.parse$1(SparkSubmitArguments.scala:432)
        at org.apache.spark.deploy.SparkSubmitArguments.parseOpts(SparkSubmitArguments.scala:288)
        at org.apache.spark.deploy.SparkSubmitArguments.<init>(SparkSubmitArguments.scala:87)
        at org.apache.spark.deploy.SparkSubmit$.main(SparkSubmit.scala:105)
        at org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/zookeeper/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/Stat
r.class]
SLF4J: Found binding in [jar:file:/usr/lib/flume-ng/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/Stati
.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
[training@localhost yarn]$ date
Mon Feb 20 09:58:12 PST 2023
[training@localhost yarn]$ yjs220000
training@localhost:~/t...
```

4.3 View the Application in the Hue Job Browser

- Go to Hue in Firefox, and select the Job Browser. (Depending on the width of your browser, you may see the whole label, or just the icon.)

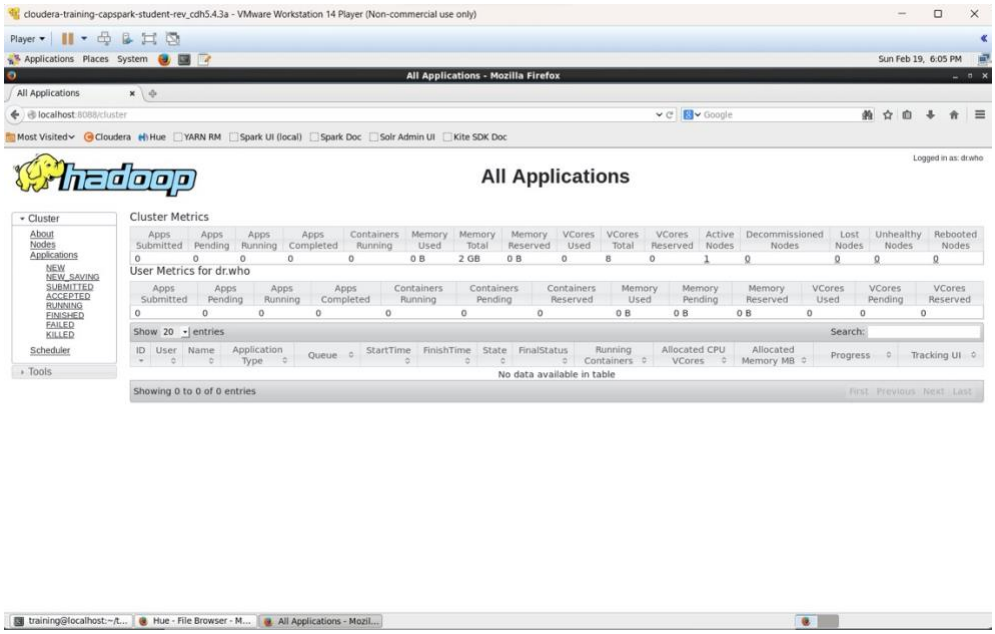
- The Job Browser displays a list of currently running and recently completed applications. (If you don't see the application you just started, wait a few seconds, the page will automatically reload; it can take some time for the application to be accepted and start running.)

- This page allows you to click the application ID to see details of the running application, or to kill a running job. (Don't do that now though!)

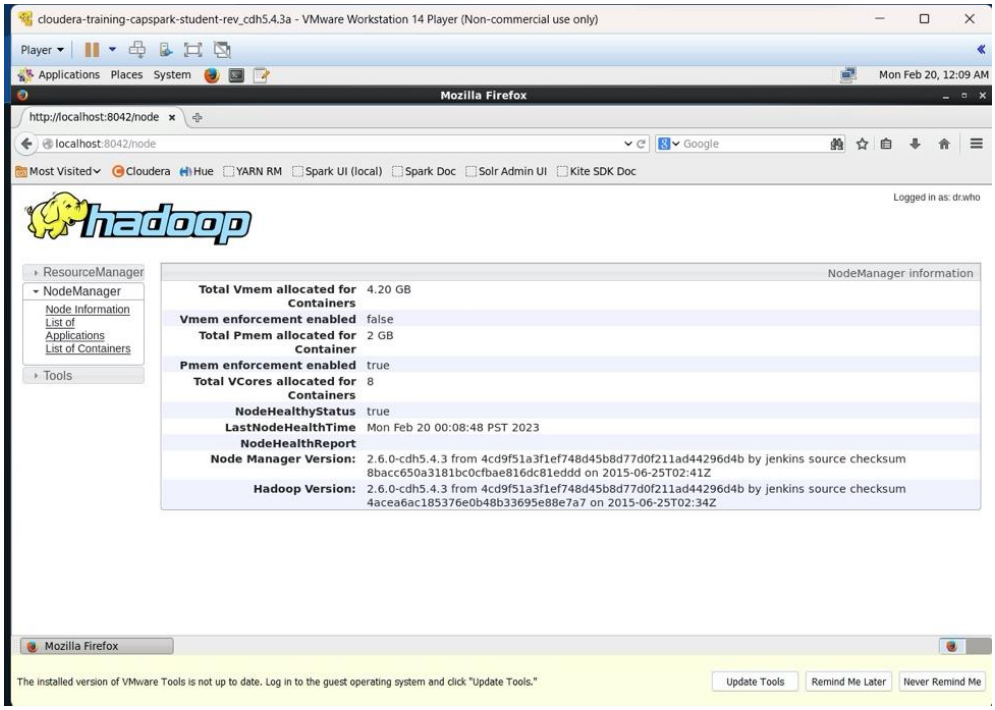
4.4 View the Application in the YARN UI

Note: To get a more detailed view of the cluster, use the YARN UI.

1.Reload the YARN RM page in Firefox. You will now see the application you just started in the bottom section of the RM home page.

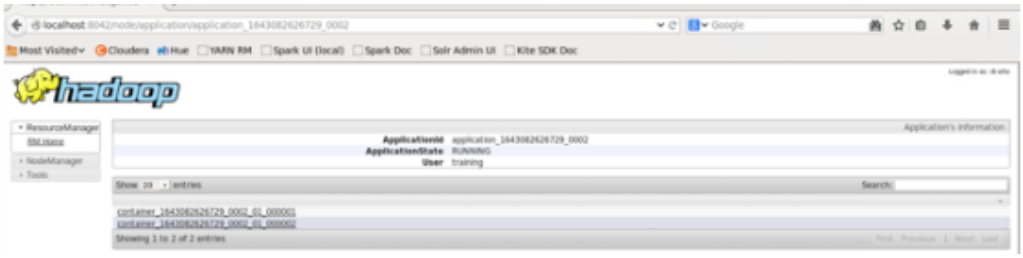


2. Select Nodes on the left menu, and Select the Node HTTP Address to open the Node Manager UI.



3. Now that an application is running, you can click on List of Applications to see the application you submitted. If it is still running, you will be able to see the containers as shown below.

- It doesn't appear when the whole application is executed, just run the command at 4.2.2 and refresh the browser.



Chapter 5

In this exercise, we will import tables from MySQL into HDFS with Sqoop

Files and Data Used in this Homework

Exercise directory: $DEV1/exercises/sqoop

MySQL Database: loudacre

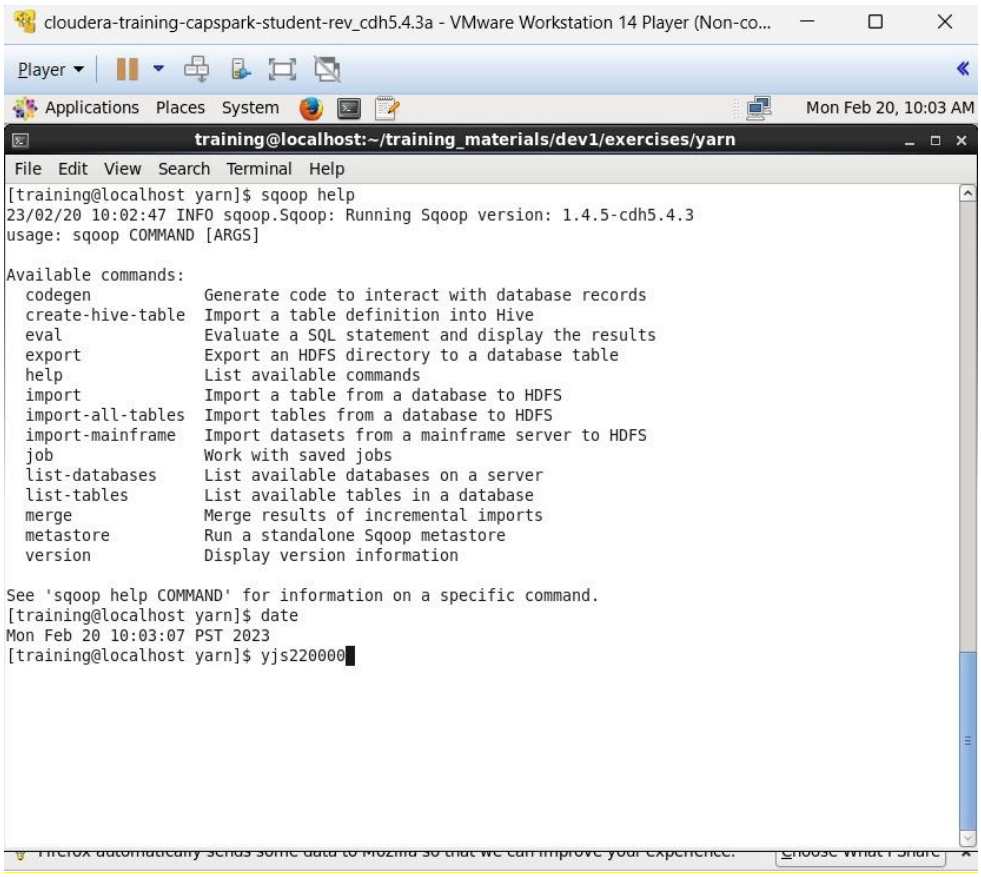
MySQL Tables: accounts, webpage

5.1 Import the Accounts table from MySQL

We can use Sqoop to look at the table layout in MySQL. With Sqoop, we can also import the table from MySQL to HDFS.

1. In a terminal window, Run the *sqoop help* command to familiarize ourselves with the options in Sqoop:

```
$ sqoop help
```



2. List the tables in the loudacre database:

```
$ sqoop list-tables \
--connect jdbc:mysql://localhost/loudacre \
--username training --password training
```

- We see the tables are in the loudacre database:

accountdevice

accounts

basestations

calllog

customerservicerep

device

knowledgebase

mostactivestations

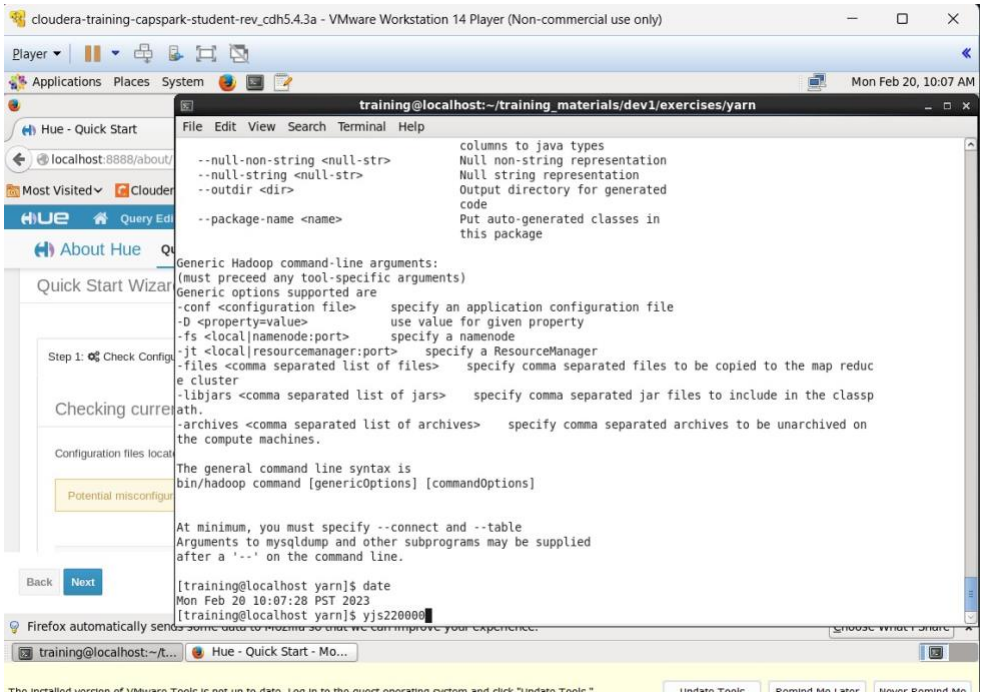
toparticles

webpage

websitehit

3. Run the sqoop import command to see its options:

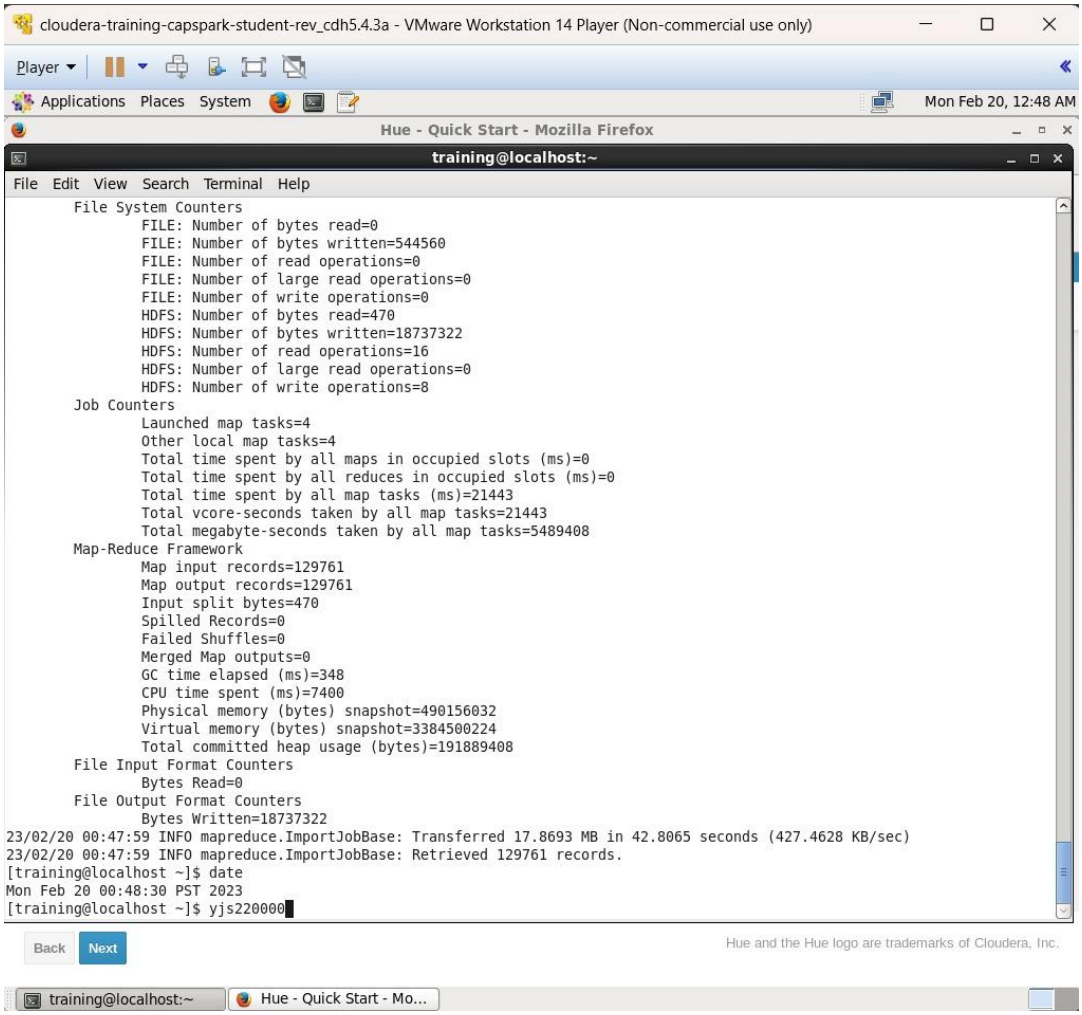
$ sqoop import --help



4.Use Sqoop to import the *accounts* table in the *loudacre* database and save it in HDFS under */loudacre*:

```
$ sqoop import \
--connect jdbc:mysql://localhost/loudacre \
--username training --password training \
--table accounts \
--target-dir /loudacre/accounts \
--null-non-string '\\N'
```

Note: The *--null-non-string* option tells Sqoop to represent null values as \N, which makes the imported data compatible with Hive and Impala.

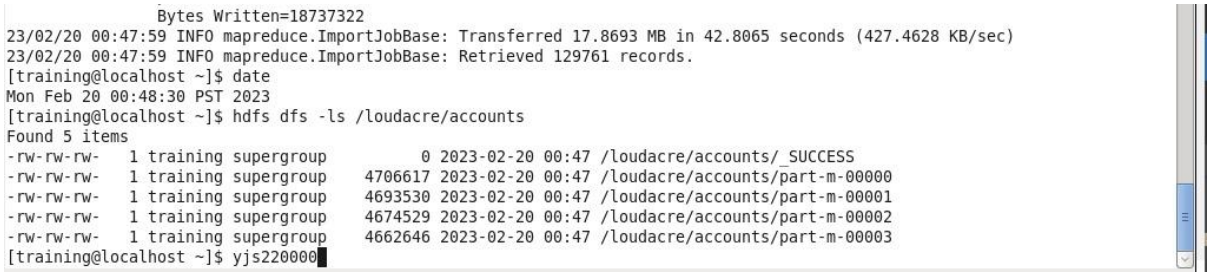


5.2 View the Imported Data

Sqoop imports the contents of the specified tables to HDFS. You can use the hdfs command line or the Hue File Browser to view the files and their contents.

1. List the contents of the *accounts* directory

$ hdfs dfs -ls /loudacre/accounts



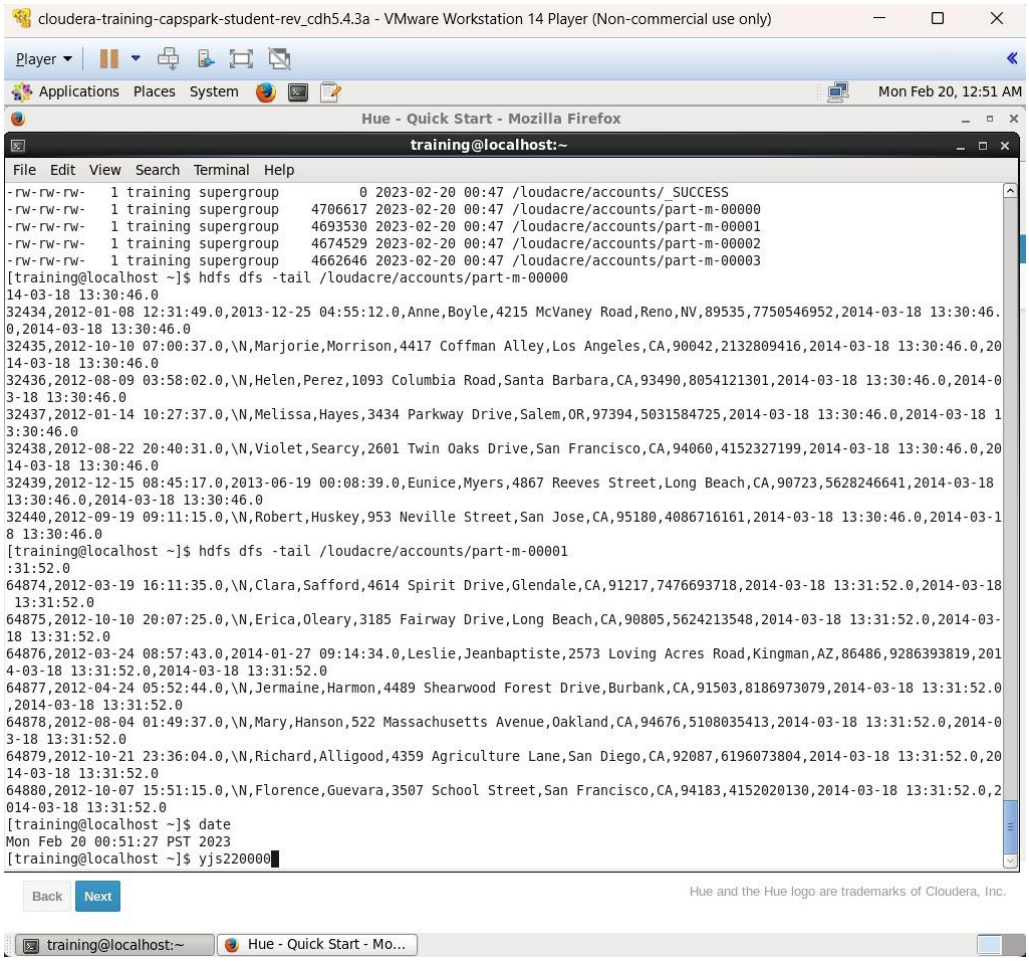
2. Use either the Hue File Browser or the HDFS *tail* command to view the last part of the file for each of the MapReduce partition files, e.g.:

$ hdfs dfs -tail /loudcare/accounts/part-m-00000

$ hdfs dfs -tail /loudcare/accounts/part-m-00001

$ hdfs dfs -tail /loudcare/accounts/part-m-00002

$ hdfs dfs -tail /loudcare/accounts/part-m-00003



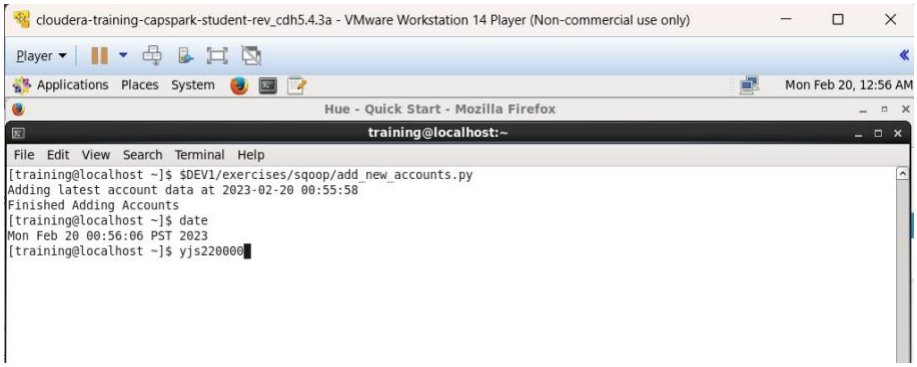
- I attached the screenshot of last command that is of part 3 file, and I see the greatest account number being *129761.*

5.3 Import Incremental Updates to Accounts

As Loudacre adds new accounts, the account data in HDFS must be updated as accounts are created. You can use Sqoop to append these new records.

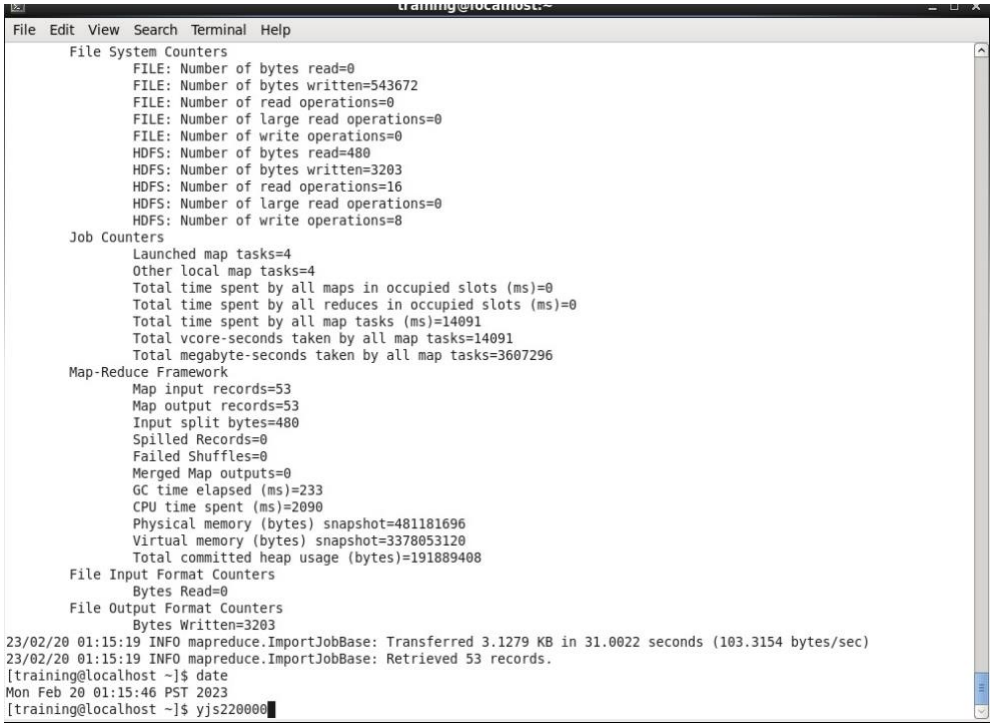
1. Run the *add_new_accounts.py* script to add the latest accounts to MySQL.

$ $DEV1/exercises/sqoop/add_new_accounts.py



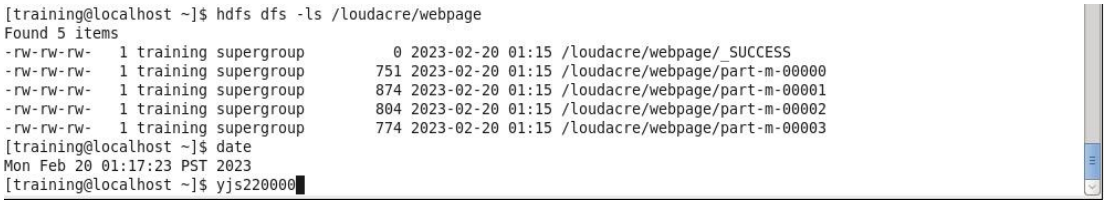
2. Incrementally import and append the newly added accounts to the accounts directory. Use Sqoop to import on the last value on the acct_num column largest account ID:

```
$ sqoop import \
--connect jdbc:mysql://localhost/loudacre \
--username training
--password training \
--incremental append \
--null-non-string '\\N' \
--table accounts \
--target-dir /loudacre/accounts \
--check-column acct_num \
--last-value 129761
```



3. List the contents of the *accounts* directory to verify the Sqoop import:

$ hdfs dfs -ls /loudacre/accounts



4. You should see three new files. Use Hadoop's *cat* command to view the entire contents of these files.

$ hdfs dfs -cat /loudacre/accounts/part-m-0000[456]

Note: I tried running this command line, but several times I was shown No such file or directory.

5.4 Import webpage data using an alternate find delimiter

1. We also want to import the webpage table to HDFS. But first look at a few records in that table using the sqoop *eval* command.

$ sqoop eval \

--query "SELECT * FROM webpage LIMIT 10" \

--connect jdbc:mysql://localhost/loudacre \

--username training --password training

```
[training@localhost ~]$ sqoop eval \
> --query "SELECT * FROM webpage LIMIT 10" \
> --connect jdbc:mysql://localhost/loudacre \
> --username training --password training
23/02/20 01:12:56 INFO sqoop.Sqoop: Running Sqoop version: 1.4.5-cdh5.4.3
23/02/20 01:12:56 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instea
d.
23/02/20 01:12:57 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
-------------------------------------------------------------
| web_page_num | web_page_file_name   | associated_files     |
-------------------------------------------------------------
| 1            | sorrento_f00l_sales.html | theme.css,code.js,sorrento_f00l.jpg |
| 2            | titanic_2100_sales.html | theme.css,code.js,titanic_2100.jpg |
| 3            | meetoo_3.0_sales.html | theme.css,code.js,meetoo_3.0.jpg |
| 4            | meetoo_3.1_sales.html | theme.css,code.js,meetoo_3.1.jpg |
| 5            | ifruit_1_sales.html  | theme.css,code.js,ifruit_1.jpg |
| 6            | ifruit_3_sales.html  | theme.css,code.js,ifruit_3.jpg |
| 7            | ifruit_2_sales.html  | theme.css,code.js,ifruit_2.jpg |
| 8            | ifruit_5_sales.html  | theme.css,code.js,ifruit_5.jpg |
| 9            | titanic_1000_sales.html | theme.css,code.js,titanic_1000.jpg |
| 10           | meetoo_1.0_sales.html | theme.css,code.js,meetoo_1.0.jpg |
-------------------------------------------------------------
[training@localhost ~]$ date
Mon Feb 20 01:13:48 PST 2023
[training@localhost ~]$ yjs220000
```

- We notice that the values in the last column contain commas. By default, sqoop uses commas as field separators, but because the data itself uses commas, we can't do that this time.

2. Use sqoop to import the webpage table, but use the tab character (\t) instead of the default (comma) as the field terminator.

$ sqoop import \

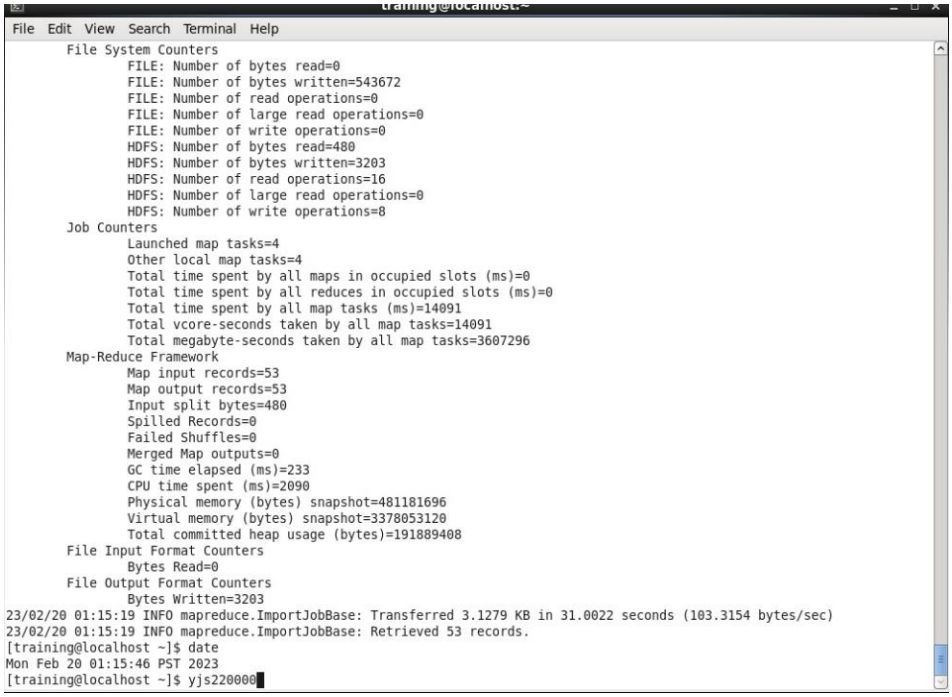
--connect jdbc:mysql://localhost/loudacre \

--username training --password training \

--table webpage \

--target-dir /loudacre/webpage \

--fields-terminated-by "\t"



3. Using Hue or the hdfs command line, review the data files imported to the */loudacre/webpage* directory. Take note of the structure of the data; you will use this data in the next exercises.

```
[training@localhost ~]$ hdfs dfs -ls /loudacre/webpage
Found 5 items
-rw-rw-rw-   1 training supergroup          0 2023-02-20 01:15 /loudacre/webpage/_SUCCESS
-rw-rw-rw-   1 training supergroup        751 2023-02-20 01:15 /loudacre/webpage/part-m-00000
-rw-rw-rw-   1 training supergroup        874 2023-02-20 01:15 /loudacre/webpage/part-m-00001
-rw-rw-rw-   1 training supergroup        804 2023-02-20 01:15 /loudacre/webpage/part-m-00002
-rw-rw-rw-   1 training supergroup        774 2023-02-20 01:15 /loudacre/webpage/part-m-00003
[training@localhost ~]$ date
Mon Feb 20 01:17:23 PST 2023
[training@localhost ~]$ yjs220000
```