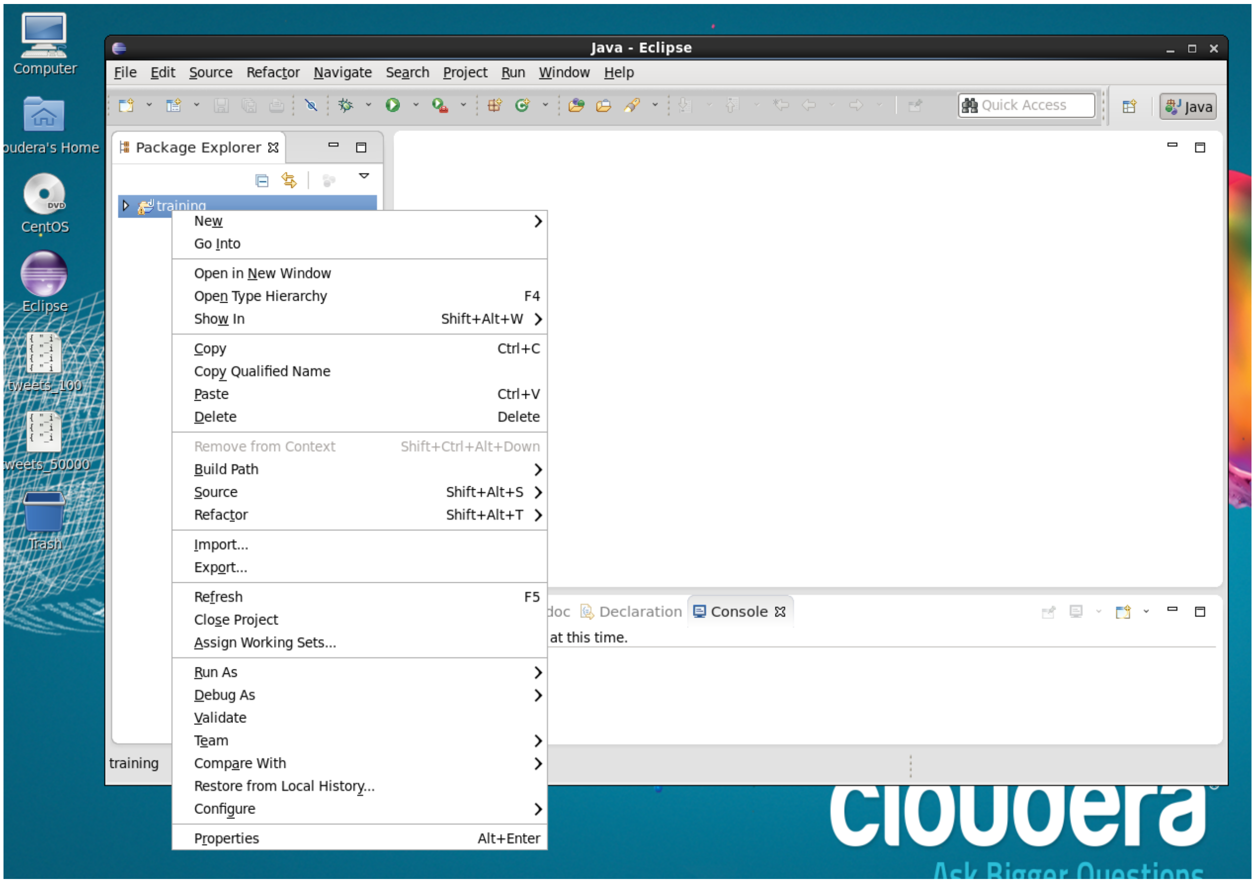
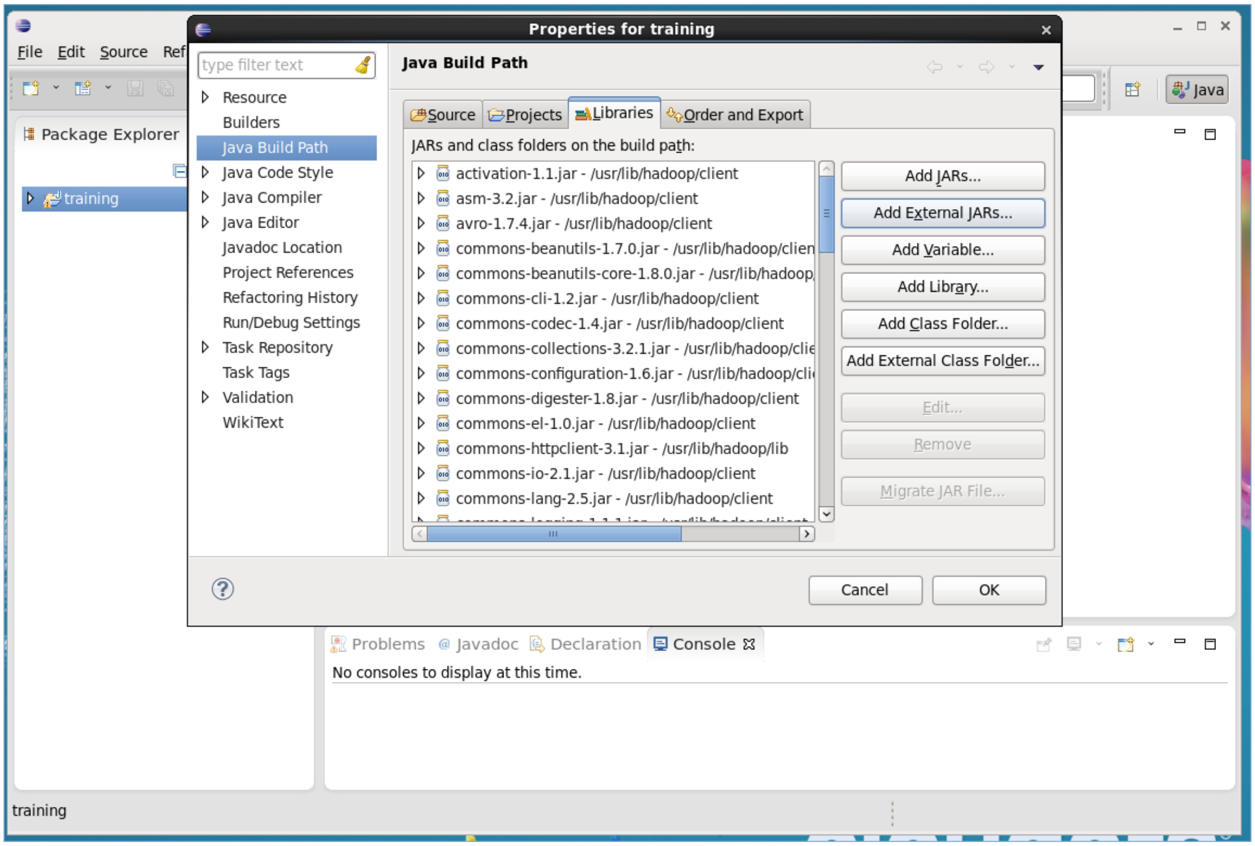
This document shows how to add twitter4j library to eclipse that comes with Cloudera VM machine and how to submit a job to the course cluster.

Adding twitter4j to Eclipse

Twitter4j jar file is needed to parse the Twitter JSON objects in the cluster

1. Open Cloudera VM image
2. Open Firefox inside the Cloudera VM image and download twitter4j-core.jar from http://tawassum.com/ksu
3. Double-click on eclipse icon on the desktop
4. Right-click on the predefined project "training" and choose "**Properties**"





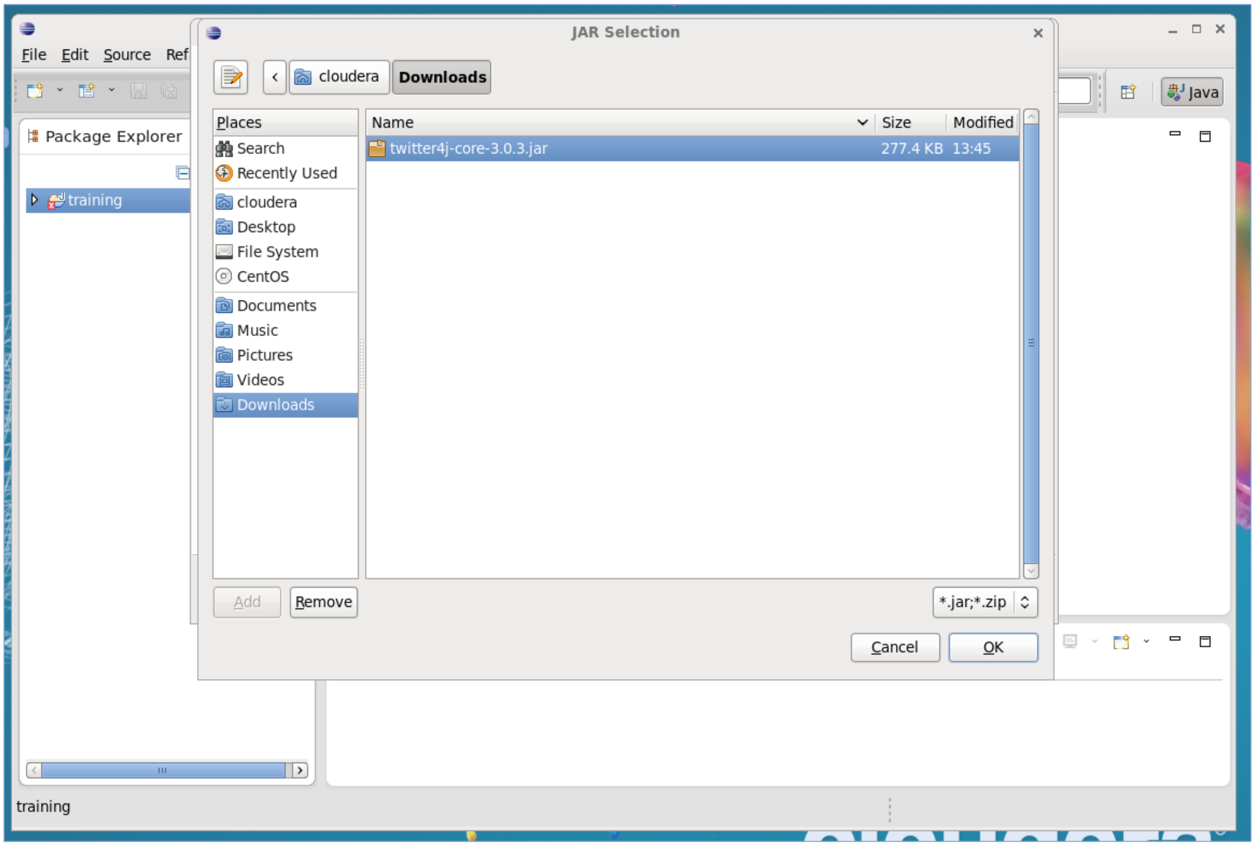
Java Build Path

JARs and class folders on the build path:

- activation-1.1.jar - /usr/lib/hadoop/client
- asm-3.2.jar - /usr/lib/hadoop/client
- avro-1.7.4.jar - /usr/lib/hadoop/client
- commons-beanutils-1.7.0.jar - /usr/lib/hadoop/clien
- commons-beanutils-core-1.8.0.jar - /usr/lib/hadoop/
- commons-cli-1.2.jar - /usr/lib/hadoop/client
- commons-codec-1.4.jar - /usr/lib/hadoop/client
- commons-collections-3.2.1.jar - /usr/lib/hadoop/clie
- commons-configuration-1.6.jar - /usr/lib/hadoop/cli
- commons-digester-1.8.jar - /usr/lib/hadoop/client
- commons-el-1.0.jar - /usr/lib/hadoop/client
- commons-httpclient-3.1.jar - /usr/lib/hadoop/lib
- commons-io-2.1.jar - /usr/lib/hadoop/client
- commons-lang-2.5.jar - /usr/lib/hadoop/client

No consoles to display at this time.

5. Click on “**Java Build Path**” from the left menu and choose the tab “**Libraries**”
6. Click “**Add External JARs**” and select twitter4j-core.jar you just downloaded and Click OK.

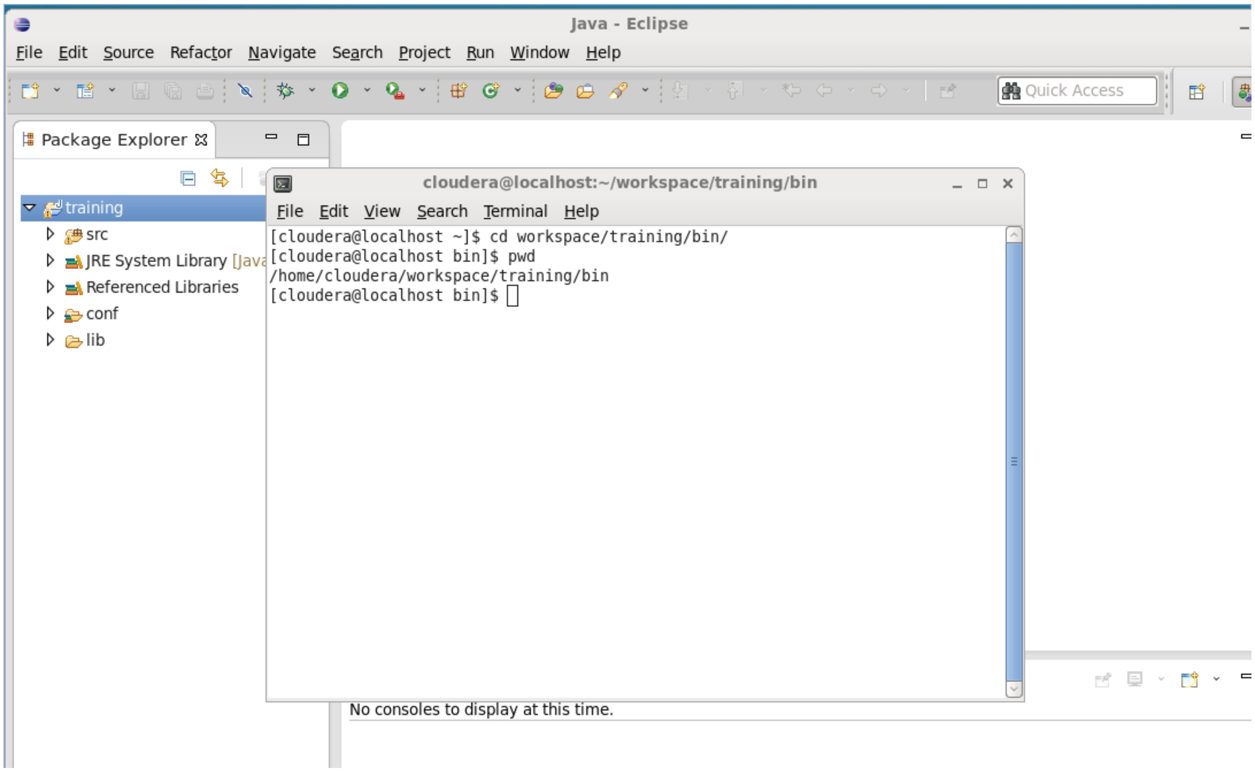


Name	Size	Modified
twitter4j-core-3.0.3.jar	277.4 KB	13:45

7. Now eclipse should not complain when you used libraries from twitter4j.

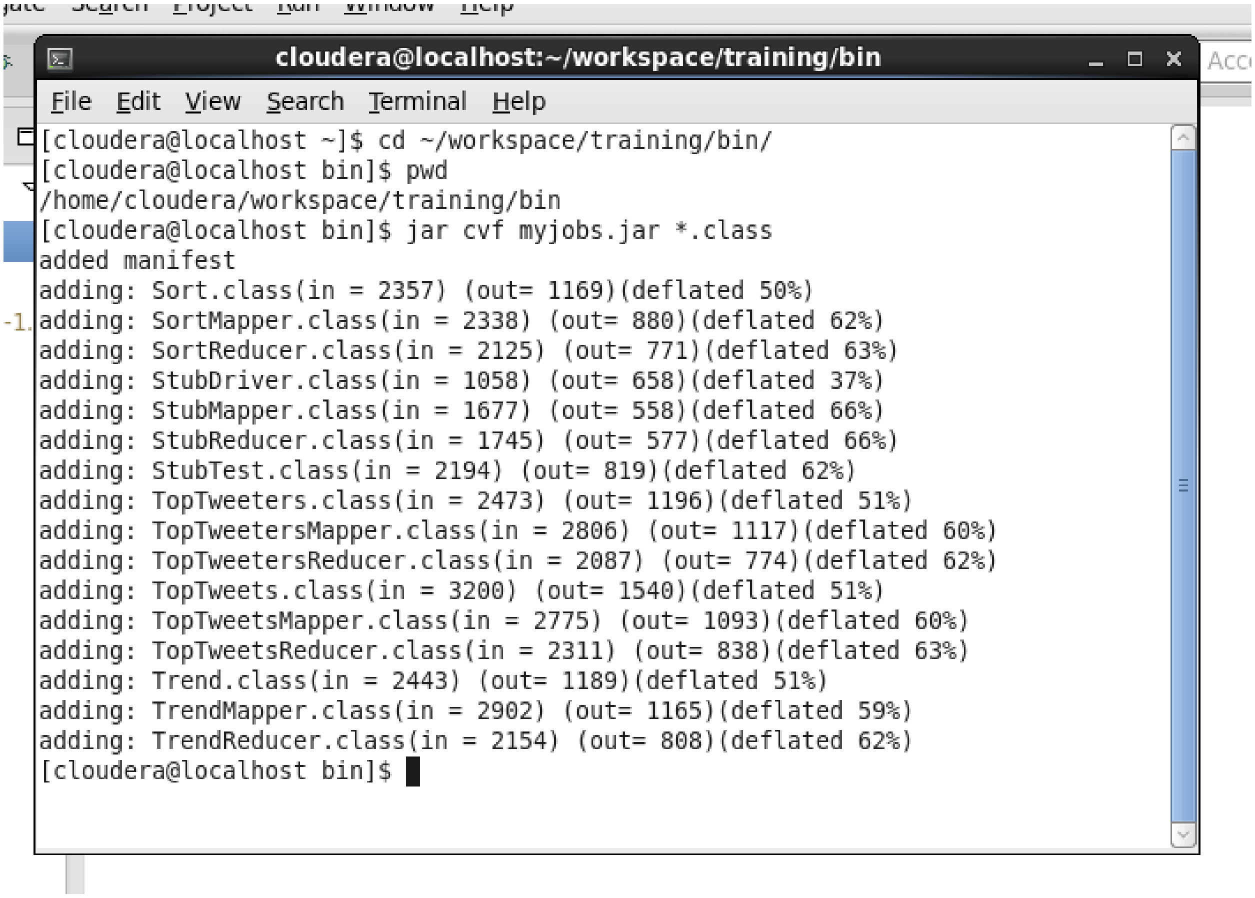
Packaging classes

1. Open a Terminal window on the VM machine and cd to ~/workspace/training/bin. Eclipse will be automatically compiling your classes in this directory


Java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
training
src
JRE System Library [Java
Referenced Libraries
conf
lib
cloudera@localhost:~/workspace/training/bin
File Edit View Search Terminal Help
No consoles to display at this time.

```
[cloudera@localhost ~]$ cd workspace/training/bin/
[cloudera@localhost bin]$ pwd
/home/cloudera/workspace/training/bin
[cloudera@localhost bin]$ 
```

2. Create a jar file of all your classes using the command: `jar cvf myjobs.jar *.class`


cloudera@localhost:~/workspace/training/bin
File Edit View Search Terminal Help
Acc

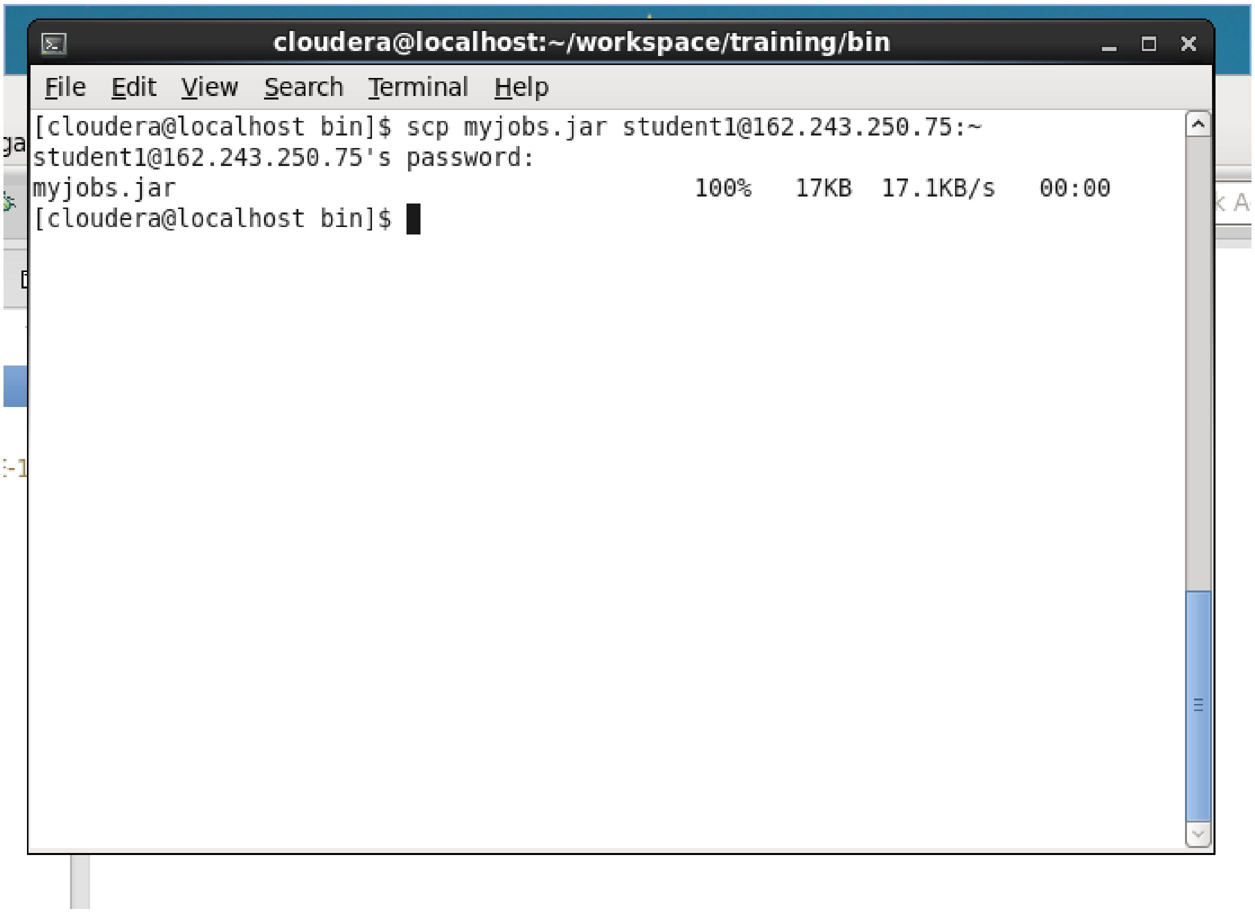
```
[cloudera@localhost ~]$ cd ~/workspace/training/bin/
[cloudera@localhost bin]$ pwd
/home/cloudera/workspace/training/bin
[cloudera@localhost bin]$ jar cvf myjobs.jar *.class
added manifest
adding: Sort.class(in = 2357) (out= 1169)(deflated 50%)
adding: SortMapper.class(in = 2338) (out= 880)(deflated 62%)
adding: SortReducer.class(in = 2125) (out= 771)(deflated 63%)
adding: StubDriver.class(in = 1058) (out= 658)(deflated 37%)
adding: StubMapper.class(in = 1677) (out= 558)(deflated 66%)
adding: StubReducer.class(in = 1745) (out= 577)(deflated 66%)
adding: StubTest.class(in = 2194) (out= 819)(deflated 62%)
adding: TopTweeters.class(in = 2473) (out= 1196)(deflated 51%)
adding: TopTweetersMapper.class(in = 2806) (out= 1117)(deflated 60%)
adding: TopTweetersReducer.class(in = 2087) (out= 774)(deflated 62%)
adding: TopTweets.class(in = 3200) (out= 1540)(deflated 51%)
adding: TopTweetsMapper.class(in = 2775) (out= 1093)(deflated 60%)
adding: TopTweetsReducer.class(in = 2311) (out= 838)(deflated 63%)
adding: Trend.class(in = 2443) (out= 1189)(deflated 51%)
adding: TrendMapper.class(in = 2902) (out= 1165)(deflated 59%)
adding: TrendReducer.class(in = 2154) (out= 808)(deflated 62%)
[cloudera@localhost bin]$ 
```

3. Upload this file to the cluster using the command:

   ```
   scp myjobs.jar student1@162.243.250.75:~
   ```

   Note: replace student1 with the user assigned to you. There are four users as follows:

User	Password
student1	student1
student2	student2
student3	student3
student4	student4



cloudera@localhost:~/workspace/training/bin
File Edit View Search Terminal Help

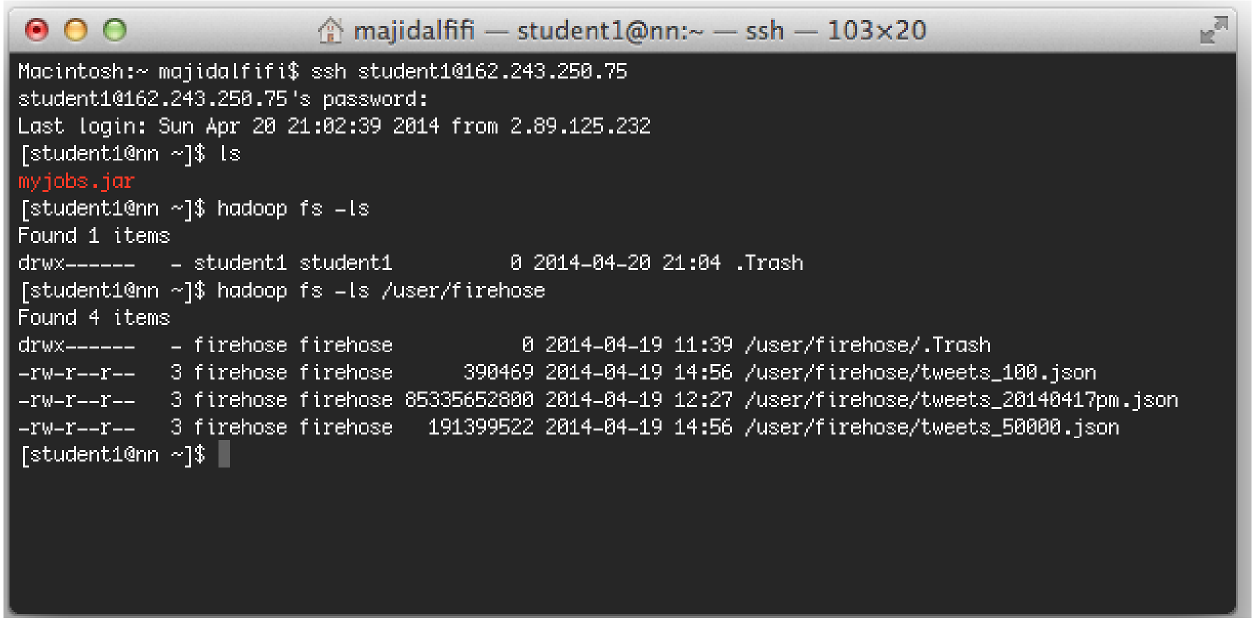
```
[cloudera@localhost bin]$ scp myjobs.jar student1@162.243.250.75:~
student1@162.243.250.75's password:
myjobs.jar                                    100%   17KB  17.1KB/s   00:00
[cloudera@localhost bin]$ 
```

4. By now your jobs jar is setting in your home directory on the cluster. you still need to submit it as explained below.

Job Submission

SSH to the server **162.243.250.75** either from the VM machine or directly from your machine using any SSH client. For example Putty on Windows or using the terminal that comes with Mac or Linux. Putty can be downloaded from http://www.putty.org/

```
ssh student1@162.243.250.75
```

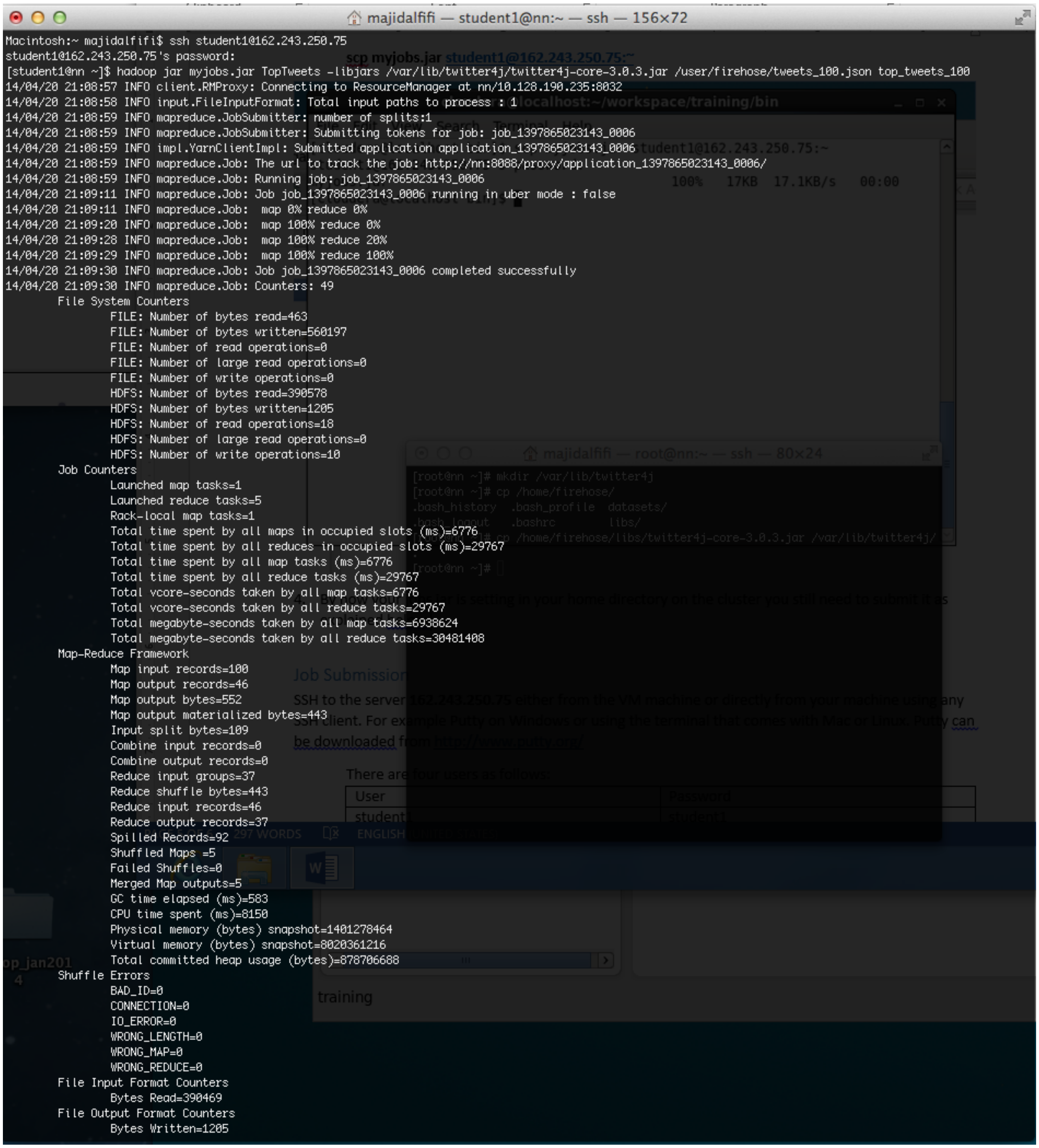


```
Macintosh:~ majidalfifi$ ssh student1@162.243.250.75
student1@162.243.250.75's password:
Last login: Sun Apr 20 21:02:39 2014 from 2.89.125.232
[student1@nn ~]$ ls
myjobs.jar
[student1@nn ~]$ hadoop fs -ls
Found 1 items
drwx------   - student1 student1          0 2014-04-20 21:04 .Trash
[student1@nn ~]$ hadoop fs -ls /user/firehose
Found 4 items
drwx------   - firehose firehose           0 2014-04-19 11:39 /user/firehose/.Trash
-rw-r--r--   3 firehose firehose      390469 2014-04-19 14:56 /user/firehose/tweets_100.json
-rw-r--r--   3 firehose firehose 85335652800 2014-04-19 12:27 /user/firehose/tweets_20140417pm.json
-rw-r--r--   3 firehose firehose   191399522 2014-04-19 14:56 /user/firehose/tweets_50000.json
[student1@nn ~]$ 
```

- You should see myjobs.jar file you just uploaded. From the above terminal, you can run HDFS and MapReduce commands for example "`hadoop fs -ls`" to list all file in your home directory on HDFS. Empty for now.
- Also note there is a user named firehose who has the datasets; you can access those datasets but you can't modify or delete them because they are owned by the user firehose.

Now to submit one of the jobs in the jar file do something like the following:

```
hadoop jar myjobs.jar TopTweets -libjars /var/lib/twitter4j/twitter4j-core-3.0.3.jar /user/firehose/tweets_100.json top_tweets_100
```



```
Macintosh:~ majidalfifi$ ssh student1@162.243.250.75
student1@162.243.250.75's password:
[student1@nn ~]$ hadoop jar myjobs.jar TopTweets -libjars /var/lib/twitter4j/twitter4j-core-3.0.3.jar /user/firehose/tweets_100.json top_tweets_100
14/04/20 21:08:57 INFO client.RMProxy: Connecting to ResourceManager at nn/10.128.190.235:8032
14/04/20 21:08:58 INFO input.FileInputFormat: Total input paths to process : 1
14/04/20 21:08:59 INFO mapreduce.JobSubmitter: number of splits:1
14/04/20 21:08:59 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1397865023143_0006
14/04/20 21:08:59 INFO impl.YarnClientImpl: Submitted application application_1397865023143_0006
14/04/20 21:08:59 INFO mapreduce.Job: The url to track the job: http://nn:8088/proxy/application_1397865023143_0006/
14/04/20 21:08:59 INFO mapreduce.Job: Running job: job_1397865023143_0006
14/04/20 21:09:11 INFO mapreduce.Job: Job job_1397865023143_0006 running in uber mode : false
14/04/20 21:09:11 INFO mapreduce.Job:  map 0% reduce 0%
14/04/20 21:09:20 INFO mapreduce.Job:  map 100% reduce 0%
14/04/20 21:09:28 INFO mapreduce.Job:  map 100% reduce 20%
14/04/20 21:09:29 INFO mapreduce.Job:  map 100% reduce 100%
14/04/20 21:09:30 INFO mapreduce.Job: Job job_1397865023143_0006 completed successfully
14/04/20 21:09:30 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=463
                FILE: Number of bytes written=560197
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=390578
                HDFS: Number of bytes written=1205
                HDFS: Number of read operations=18
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=10
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=5
                Rack-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=6776
                Total time spent by all reduces in occupied slots (ms)=29767
                Total time spent by all map tasks (ms)=6776
                Total time spent by all reduce tasks (ms)=29767
                Total vcore-seconds taken by all map tasks=6776
                Total vcore-seconds taken by all reduce tasks=29767
                Total megabyte-seconds taken by all map tasks=6938624
                Total megabyte-seconds taken by all reduce tasks=30481408
        Map-Reduce Framework
                Map input records=100
                Map output records=46
                Map output bytes=552
                Map output materialized bytes=443
                Input split bytes=109
                Combine input records=0
                Combine output records=0
                Reduce input groups=37
                Reduce shuffle bytes=443
                Reduce input records=46
                Reduce output records=37
                Spilled Records=92
                Shuffled Maps =5
                Failed Shuffles=0
                Merged Map outputs=5
                GC time elapsed (ms)=583
                CPU time spent (ms)=8150
                Physical memory (bytes) snapshot=1401278464
                Virtual memory (bytes) snapshot=8020361216
                Total committed heap usage (bytes)=878706688
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=390469
        File Output Format Counters
                Bytes Written=1205
```

You can now run `hadoop fs -ls` to explore the generated output.