Hadoop HDFS & MapReduce

Part I

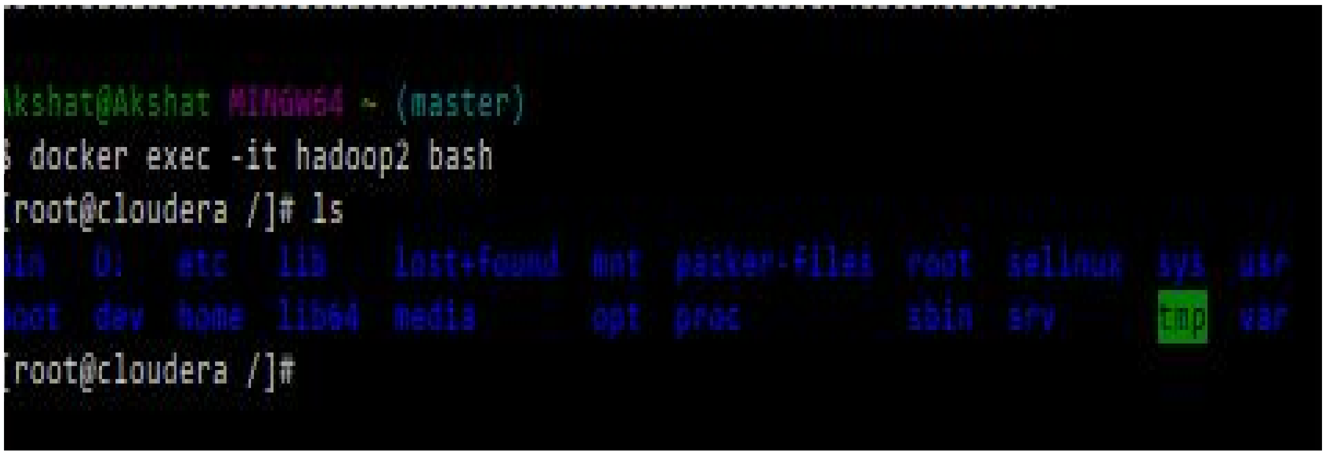
The wordcount program from the class was modified by created a tuple containing word pairs, and printing the pairs as the output of the mapper. The reducer then counts these tuples and aggregates them. The job was performed using the Hadoop streaming service. First, we run docker on Linux machine after setting it up. Then to start Hadoop, we first enter the following command to look for a hadoop image. If the image doesn't exist, then a new image is downloaded.

docker run --hostname=quickstart.cloudera --name=hadoop2 --privileged=true -t -d -p 80:80 -p 8080:8080 -p 8888:8888 -v /c/Users/:home cloudera/quickstart /usr/bin/docker-quickstart

We then enter the following command to execute hadoop-

docker exec -it hadoop2 bash

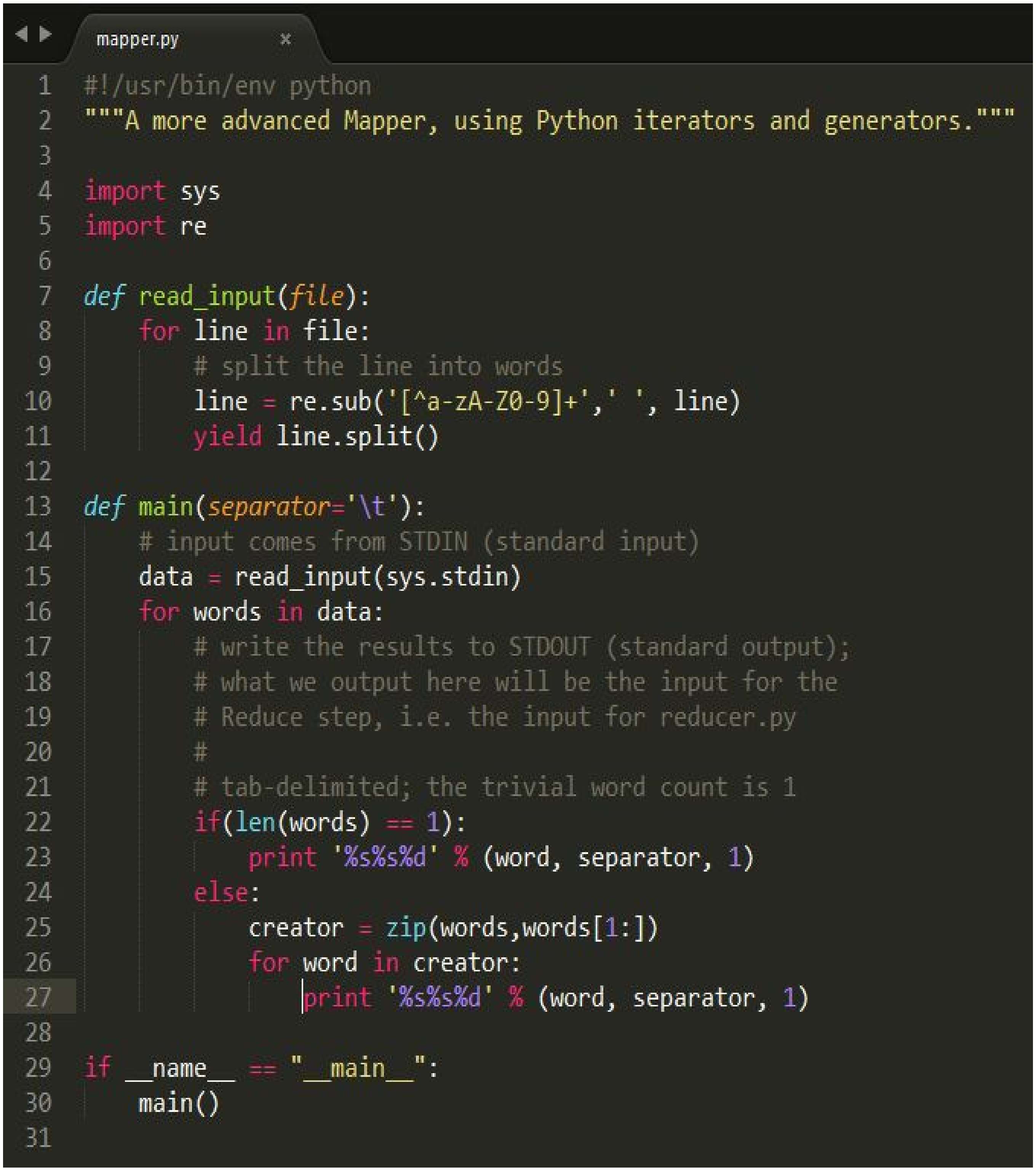
This gives the following output -



```
kshat@Akshat MINGW64 ~ (master)
$ docker exec -it hadoop2 bash
[root@cloudera /]# ls
bin  D:   etc   lib    lost+found  mnt  packer-files  root  selinux  sys  usr
boot dev  home  lib64  media       opt  proc          sbin  srv      tmp  var
[root@cloudera /]#
```

To run the file in Hadoop, I must iterate through to the directory containing my mapper and reducer, and execute the command for it to run. Then, I will check Hue to locate my output file.

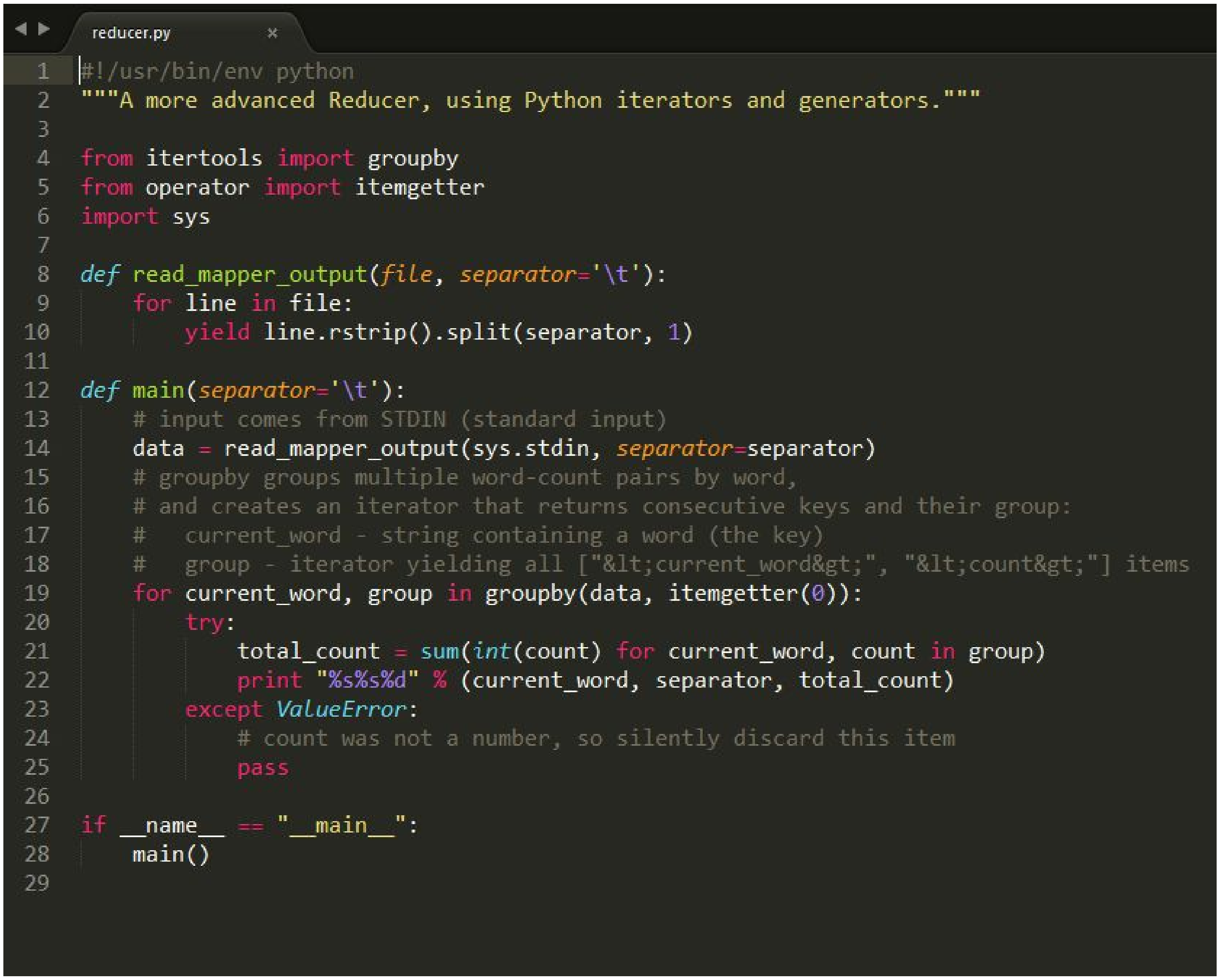
The mapping file, *mapper.py*, is as follows -



mapper.py

```python
#!/usr/bin/env python
"""A more advanced Mapper, using Python iterators and generators."""

import sys
import re

def read_input(file):
    for line in file:
        # split the line into words
        line = re.sub('[^a-zA-Z0-9]+',' ', line)
        yield line.split()

def main(separator='\t'):
    # input comes from STDIN (standard input)
    data = read_input(sys.stdin)
    for words in data:
        # write the results to STDOUT (standard output);
        # what we output here will be the input for the
        # Reduce step, i.e. the input for reducer.py
        #
        # tab-delimited; the trivial word count is 1
        if(len(words) == 1):
            print '%s%s%d' % (word, separator, 1)
        else:
            creator = zip(words,words[1:])
            for word in creator:
                print '%s%s%d' % (word, separator, 1)

if __name__ == "__main__":
    main()

```

The reducer file, *reduce.py*, is as follows -



reducer.py

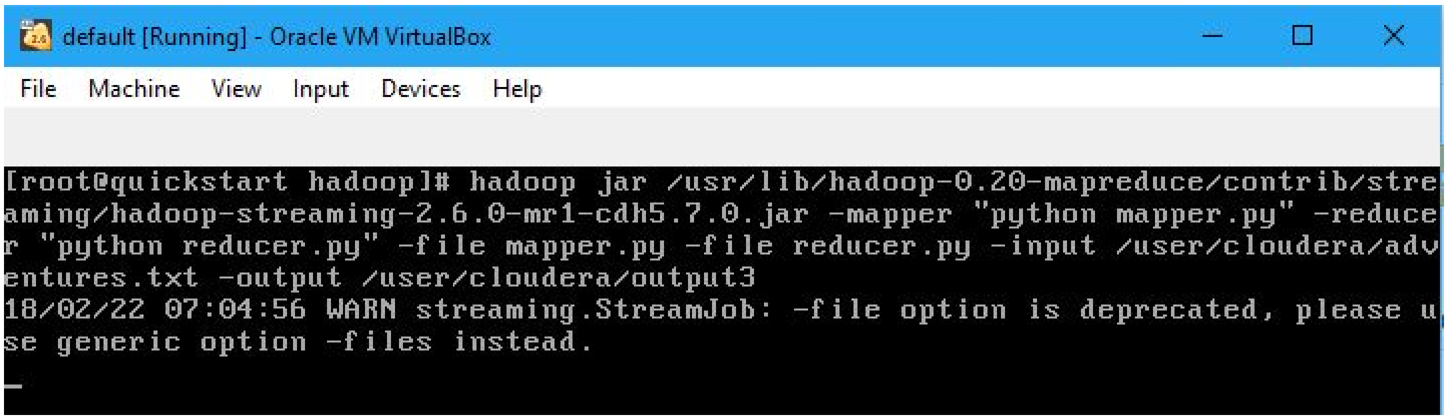
```python
#!/usr/bin/env python
"""A more advanced Reducer, using Python iterators and generators."""

from itertools import groupby
from operator import itemgetter
import sys

def read_mapper_output(file, separator='\t'):
    for line in file:
        yield line.rstrip().split(separator, 1)

def main(separator='\t'):
    # input comes from STDIN (standard input)
    data = read_mapper_output(sys.stdin, separator=separator)
    # groupby groups multiple word-count pairs by word,
    # and creates an iterator that returns consecutive keys and their group:
    #   current_word - string containing a word (the key)
    #   group - iterator yielding all ["&lt;current_word&gt;", "&lt;count&gt;"] items
    for current_word, group in groupby(data, itemgetter(0)):
        try:
            total_count = sum(int(count) for current_word, count in group)
            print "%s%s%d" % (current_word, separator, total_count)
        except ValueError:
            # count was not a number, so silently discard this item
            pass

if __name__ == "__main__":
    main()
```

I have setup my mapper and reducer in the folder hadoop, located in C:\Users\Akshat\Desktop\hadoop. After iterating through this path in hadoop, I must enter the following command to submit the job to hadoop -

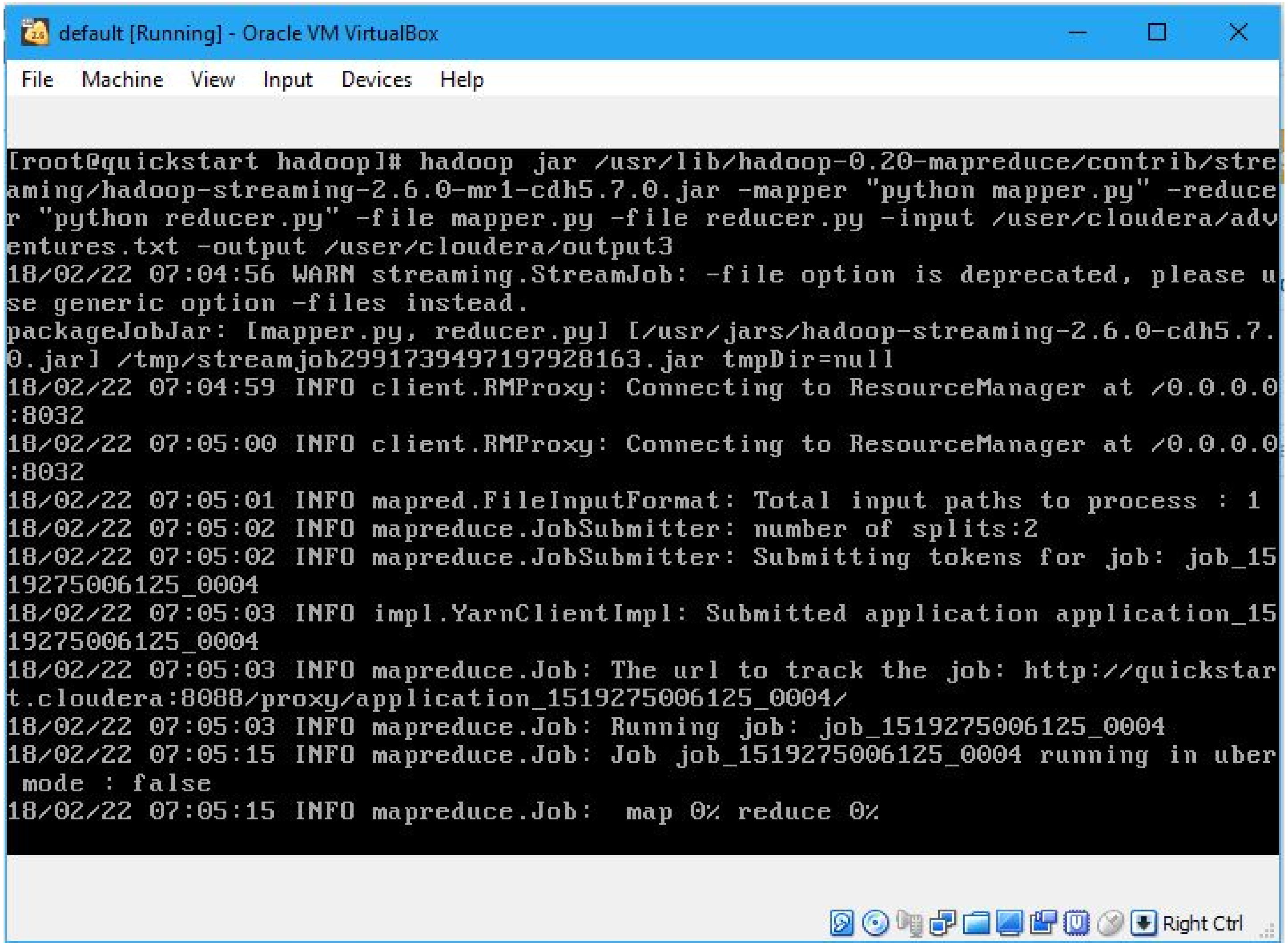
```
hadoop jar /usr/lib/hadoop-0.20-mapreduce/contrib/streaming/hadoop-streaming-2.6.0-mr1-cdh5.7.0.jar -mapper "python mapper.py" -reducer "python reducer.py" -file mapper.py -file reducer.py -input /user/cloudera/adventures.txt -output /user/cloudera/output8
```

The text file to use, *adventures.txt*, was placed in user/cloudera through Hue. This would run the job in Hadoop. The jar file would be created and the output would be positioned in the path specified in the command. In case of any errors in the code, the job would not complete and this would be intimated through the console and Hue's job tracker.

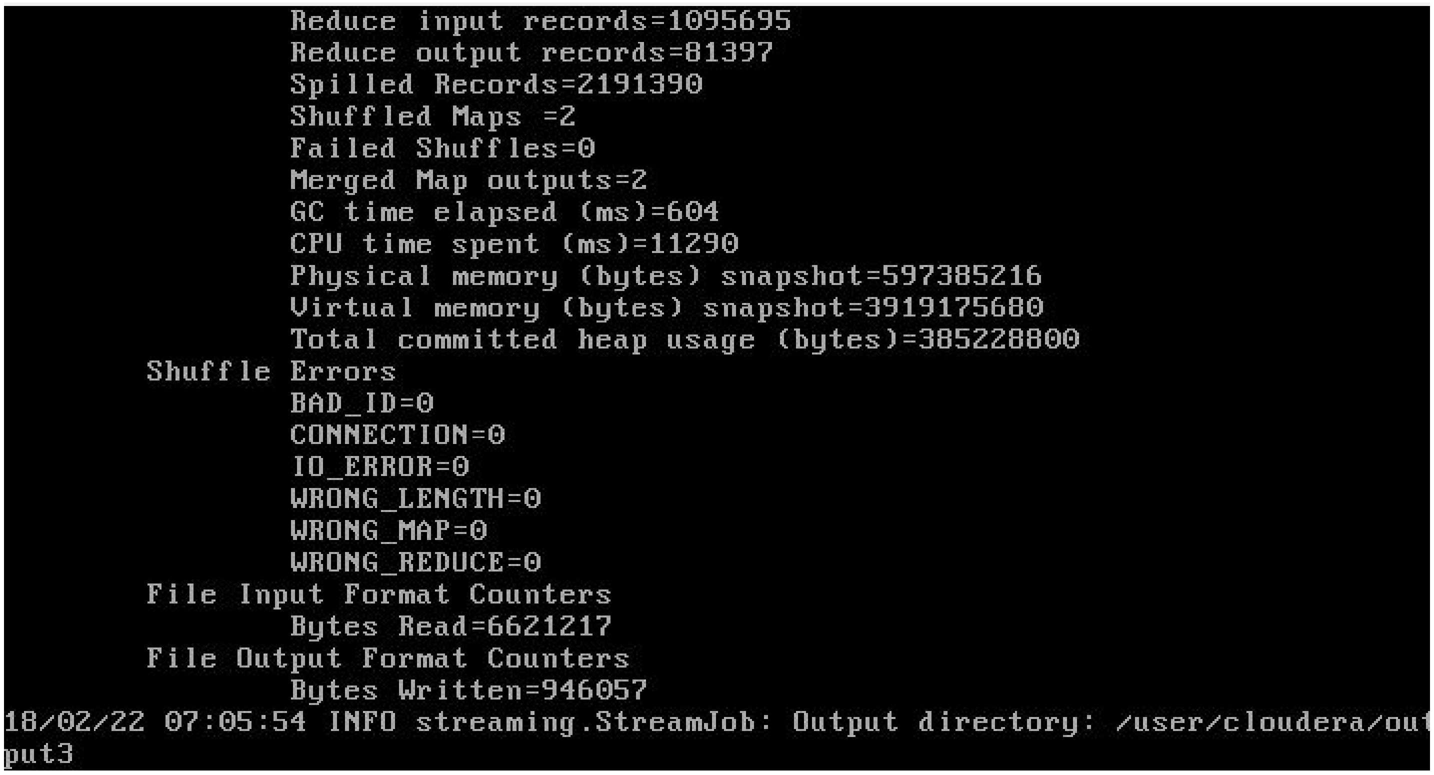
The output of the command above is as follows -


default [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

```
[root@quickstart hadoop]# hadoop jar /usr/lib/hadoop-0.20-mapreduce/contrib/stre
aming/hadoop-streaming-2.6.0-mr1-cdh5.7.0.jar -mapper "python mapper.py" -reduce
r "python reducer.py" -file mapper.py -file reducer.py -input /user/cloudera/adv
entures.txt -output /user/cloudera/output3
18/02/22 07:04:56 WARN streaming.StreamJob: -file option is deprecated, please u
se generic option -files instead.
_
```

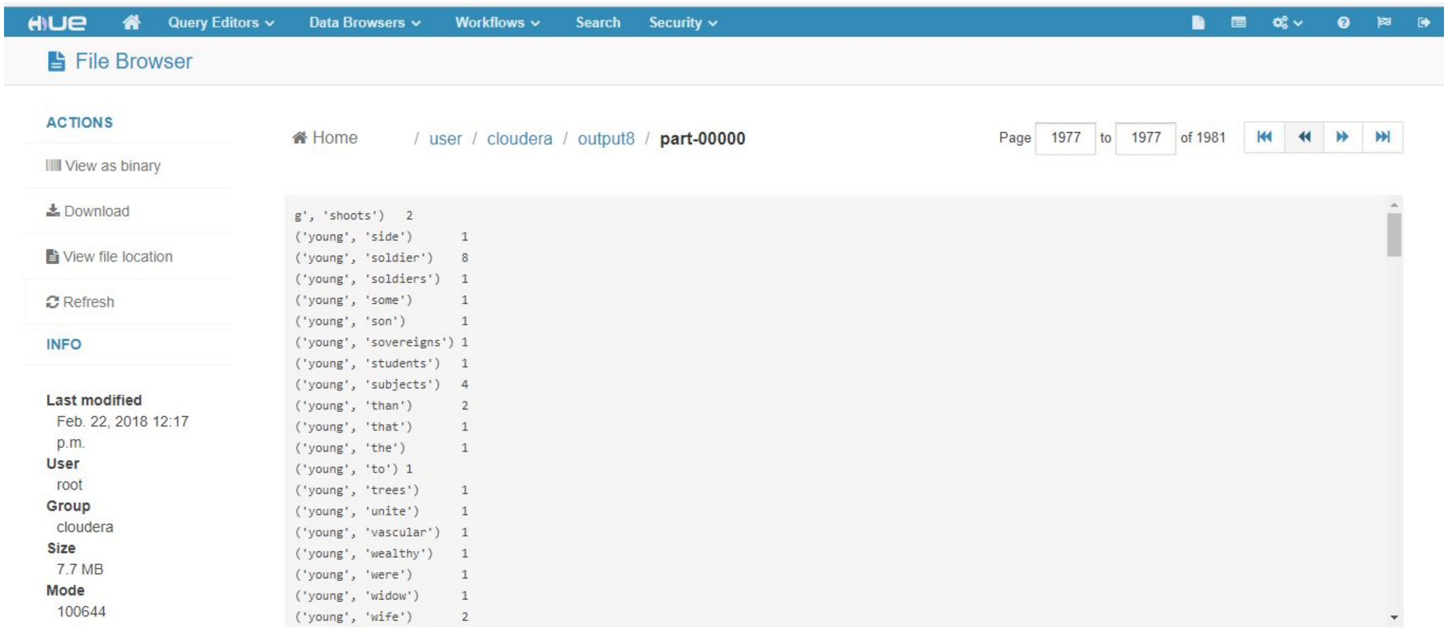

default [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

```
[root@quickstart hadoop]# hadoop jar /usr/lib/hadoop-0.20-mapreduce/contrib/stre
aming/hadoop-streaming-2.6.0-mr1-cdh5.7.0.jar -mapper "python mapper.py" -reduce
r "python reducer.py" -file mapper.py -file reducer.py -input /user/cloudera/adv
entures.txt -output /user/cloudera/output3
18/02/22 07:04:56 WARN streaming.StreamJob: -file option is deprecated, please u
se generic option -files instead.
packageJobJar: [mapper.py, reducer.py] [/usr/jars/hadoop-streaming-2.6.0-cdh5.7.
0.jar] /tmp/streamjob2991739497197928163.jar tmpDir=null
18/02/22 07:04:59 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0
:8032
18/02/22 07:05:00 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0
:8032
18/02/22 07:05:01 INFO mapred.FileInputFormat: Total input paths to process : 1
18/02/22 07:05:02 INFO mapreduce.JobSubmitter: number of splits:2
18/02/22 07:05:02 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_15
19275006125_0004
18/02/22 07:05:03 INFO impl.YarnClientImpl: Submitted application application_15
19275006125_0004
18/02/22 07:05:03 INFO mapreduce.Job: The url to track the job: http://quickstar
t.cloudera:8088/proxy/application_1519275006125_0004/
18/02/22 07:05:03 INFO mapreduce.Job: Running job: job_1519275006125_0004
18/02/22 07:05:15 INFO mapreduce.Job: Job job_1519275006125_0004 running in uber
 mode : false
18/02/22 07:05:15 INFO mapreduce.Job:  map 0% reduce 0%
```



```
                Reduce input records=1095695
                Reduce output records=81397
                Spilled Records=2191390
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=604
                CPU time spent (ms)=11290
                Physical memory (bytes) snapshot=597385216
                Virtual memory (bytes) snapshot=3919175680
                Total committed heap usage (bytes)=385228800
        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=6621217
        File Output Format Counters
                Bytes Written=946057
18/02/22 07:05:54 INFO streaming.StreamJob: Output directory: /user/cloudera/out
put3
```

Our program compiled and ran on hadoop successfully. On typing http://192.168.99.100:8888 in the web browser, Hue can be accessed. The output in Hue is as follows-



/ user / cloudera / output8 / part-00000

Last modified
Feb. 22, 2018 12:17 p.m.
User
root
Group
cloudera
Size
7.7 MB
Mode
100644

```
g', 'shoots')   2
('young', 'side')       1
('young', 'soldier')    8
('young', 'soldiers')   1
('young', 'some')       1
('young', 'son')        1
('young', 'sovereigns') 1
('young', 'students')   1
('young', 'subjects')   4
('young', 'than')       2
('young', 'that')       1
('young', 'the')        1
('young', 'to') 1
('young', 'trees')      1
('young', 'unite')      1
('young', 'vascular')   1
('young', 'wealthy')    1
('young', 'were')       1
('young', 'widow')      1
('young', 'wife')       2
```

Here, part-0000 is the output file, located in the folder output8. It contains the tuple containing pair of words, separated by their frequencies in the text file.

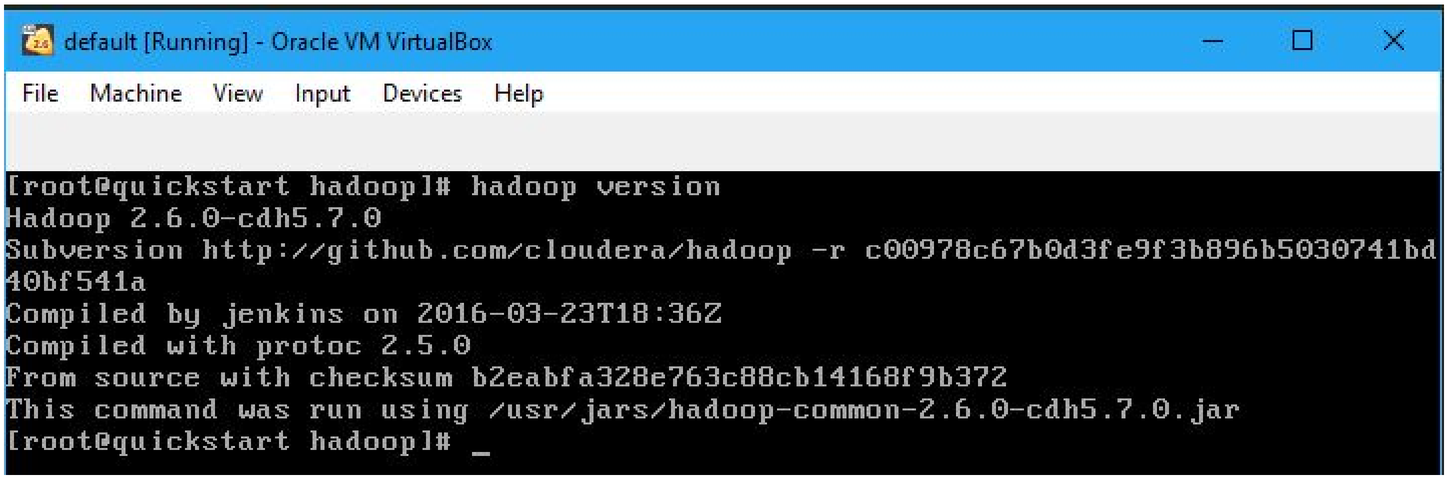
Part II

Running a version of Hadoop (docker or otherwise), create a directory on HDFS with your first name (e.g. mine will be 'juan'). Submit a screen grab of the output of a Hadoop fs listing showing your home directory and your new directory in it.

The version of Hadoop can be specified with the following command -

hadoop version

The output is as follows -


default [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

```
[root@quickstart hadoop]# hadoop version
Hadoop 2.6.0-cdh5.7.0
Subversion http://github.com/cloudera/hadoop -r c00978c67b0d3fe9f3b896b5030741bd
40bf541a
Compiled by jenkins on 2016-03-23T18:36Z
Compiled with protoc 2.5.0
From source with checksum b2eabfa328e763c88cb14168f9b372
This command was run using /usr/jars/hadoop-common-2.6.0-cdh5.7.0.jar
[root@quickstart hadoop]# _
```

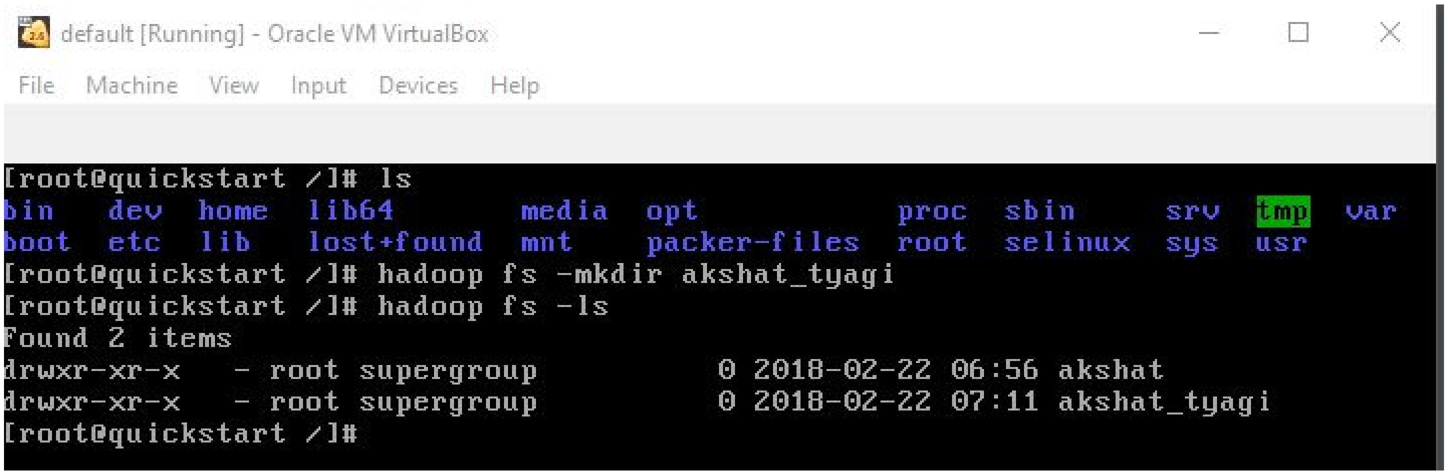
To create a new directory in Hadoop FS, the following command is executed -

hadoop fs -mkdir akshat_tyagi

This would create a directory named akshat_tyagi in the present working directory. To look at the files and directories in the current working directory, we may use the following command -

hadoop fs -ls

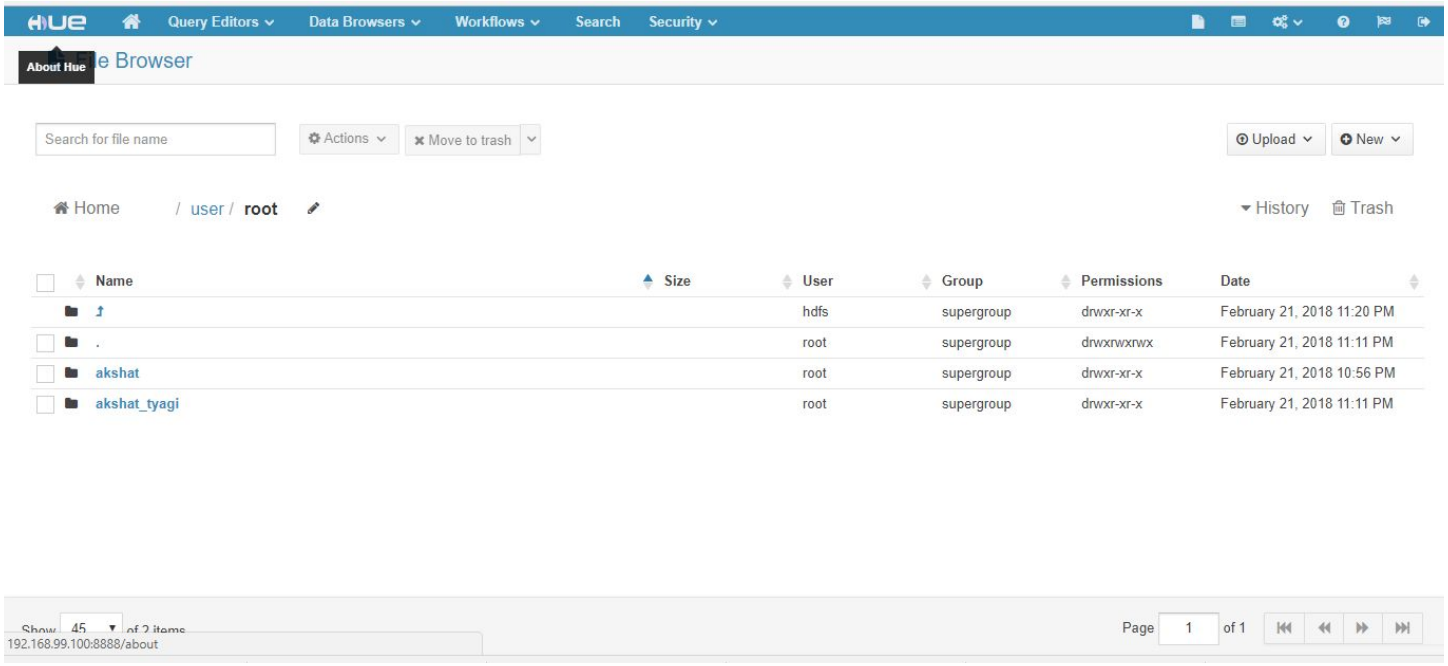
Basically, the command line for hadoop fs follows linux shell line, just that come of the important commands in linux end up as switches for hadoop fs. The outputs and the directories that we created have been shown in the screenshot of the terminal attached below -


default [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

```
[root@quickstart /]# ls
bin   dev   home  lib64       media  opt           proc  sbin     srv  tmp  var
boot  etc   lib   lost+found  mnt    packer-files  root  selinux  sys  usr
[root@quickstart /]# hadoop fs -mkdir akshat_tyagi
[root@quickstart /]# hadoop fs -ls
Found 2 items
drwxr-xr-x   - root supergroup          0 2018-02-22 06:56 akshat
drwxr-xr-x   - root supergroup          0 2018-02-22 07:11 akshat_tyagi
[root@quickstart /]#
```

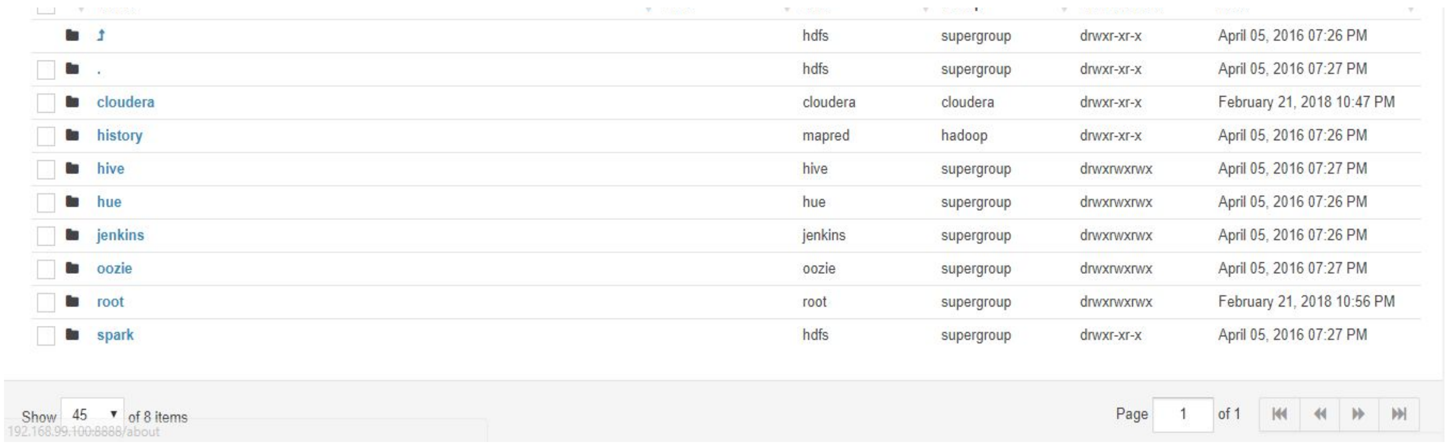
Submit a screen grab of Hue showing the same as (a)

The screenshots have been attached below -



Name	Size	User	Group	Permissions	Date
↥		hdfs	supergroup	drwxr-xr-x	February 21, 2018 11:20 PM
.		root	supergroup	drwxrwxrwx	February 21, 2018 11:11 PM
akshat		root	supergroup	drwxr-xr-x	February 21, 2018 10:56 PM
akshat_tyagi		root	supergroup	drwxr-xr-x	February 21, 2018 11:11 PM

As shown above, the two directories that we created are present in /user/root. The root directory is located as follows -



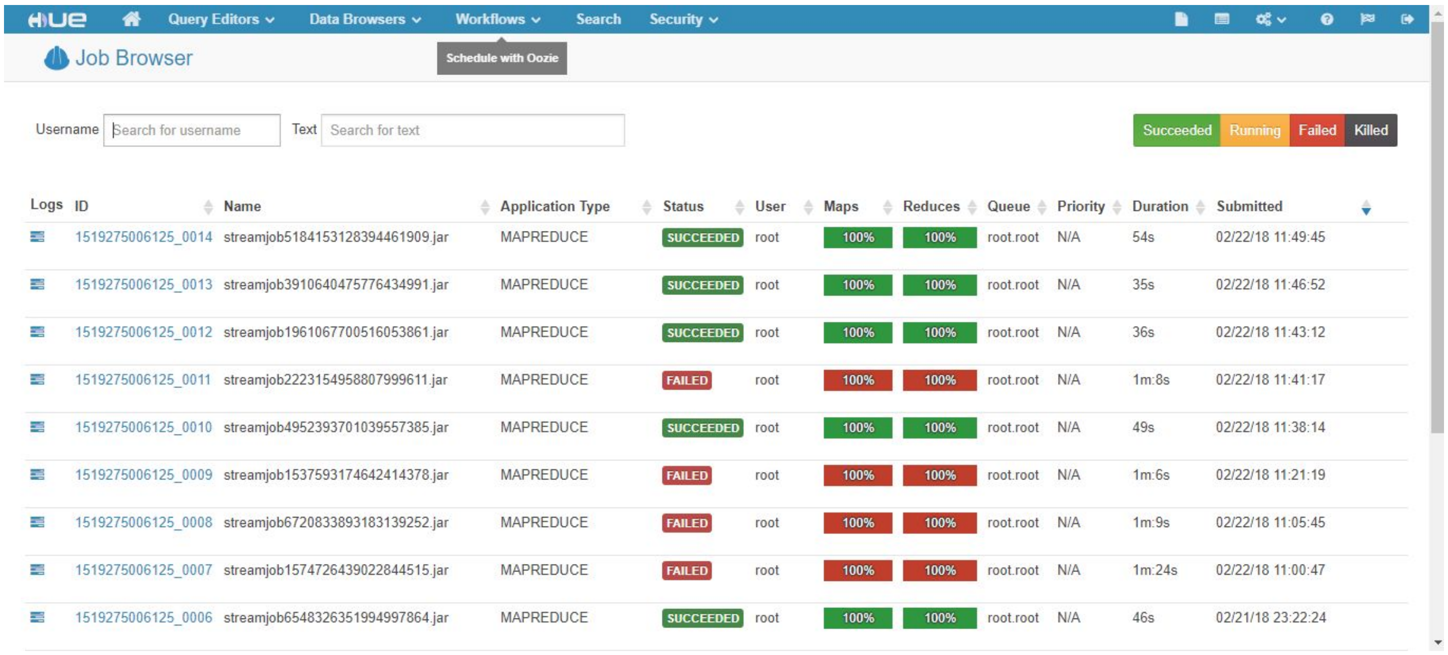
↥	hdfs	supergroup	drwxr-xr-x	April 05, 2016 07:26 PM
.	hdfs	supergroup	drwxr-xr-x	April 05, 2016 07:27 PM
cloudera	cloudera	cloudera	drwxr-xr-x	February 21, 2018 10:47 PM
history	mapred	hadoop	drwxr-xr-x	April 05, 2016 07:26 PM
hive	hive	supergroup	drwxrwxrwx	April 05, 2016 07:27 PM
hue	hue	supergroup	drwxrwxrwx	April 05, 2016 07:26 PM
jenkins	jenkins	supergroup	drwxrwxrwx	April 05, 2016 07:26 PM
oozie	oozie	supergroup	drwxrwxrwx	April 05, 2016 07:27 PM
root	root	supergroup	drwxrwxrwx	February 21, 2018 10:56 PM
spark	hdfs	supergroup	drwxr-xr-x	April 05, 2016 07:27 PM

Show 45 of 8 items — Page 1 of 1

192.168.99.100:8888/about

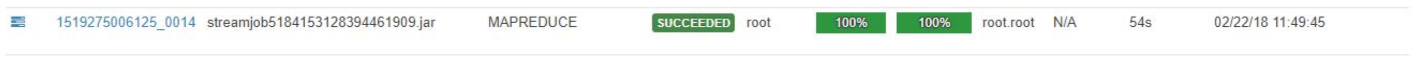
Submit a screen grab of your program running or completed in Hadoop. Hue has a jobs status page, use that one. Or use the command: 'hadoop jar' . See the documentation.

The screengrab has been attached below -



Logs	ID	Name	Application Type	Status	User	Maps	Reduces	Queue	Priority	Duration	Submitted
	1519275006125_0014	streamjob5184153128394461909.jar	MAPREDUCE	SUCCEEDED	root	100%	100%	root.root	N/A	54s	02/22/18 11:49:45
	1519275006125_0013	streamjob3910640475776434991.jar	MAPREDUCE	SUCCEEDED	root	100%	100%	root.root	N/A	35s	02/22/18 11:46:52
	1519275006125_0012	streamjob1961067700516053861.jar	MAPREDUCE	SUCCEEDED	root	100%	100%	root.root	N/A	36s	02/22/18 11:43:12
	1519275006125_0011	streamjob2223154958807999611.jar	MAPREDUCE	FAILED	root	100%	100%	root.root	N/A	1m:8s	02/22/18 11:41:17
	1519275006125_0010	streamjob4952393701039557385.jar	MAPREDUCE	SUCCEEDED	root	100%	100%	root.root	N/A	49s	02/22/18 11:38:14
	1519275006125_0009	streamjob1537593174642414378.jar	MAPREDUCE	FAILED	root	100%	100%	root.root	N/A	1m:6s	02/22/18 11:21:19
	1519275006125_0008	streamjob6720833893183139252.jar	MAPREDUCE	FAILED	root	100%	100%	root.root	N/A	1m:9s	02/22/18 11:05:45
	1519275006125_0007	streamjob1574726439022844515.jar	MAPREDUCE	FAILED	root	100%	100%	root.root	N/A	1m:24s	02/22/18 11:00:47
	1519275006125_0006	streamjob6548326351994997864.jar	MAPREDUCE	SUCCEEDED	root	100%	100%	root.root	N/A	46s	02/21/18 23:22:24

The status for the job for our wordcount program is as follows -



	1519275006125_0014	streamjob5184153128394461909.jar	MAPREDUCE	SUCCEEDED	root	100%	100%	root.root	N/A	54s	02/22/18 11:49:45