

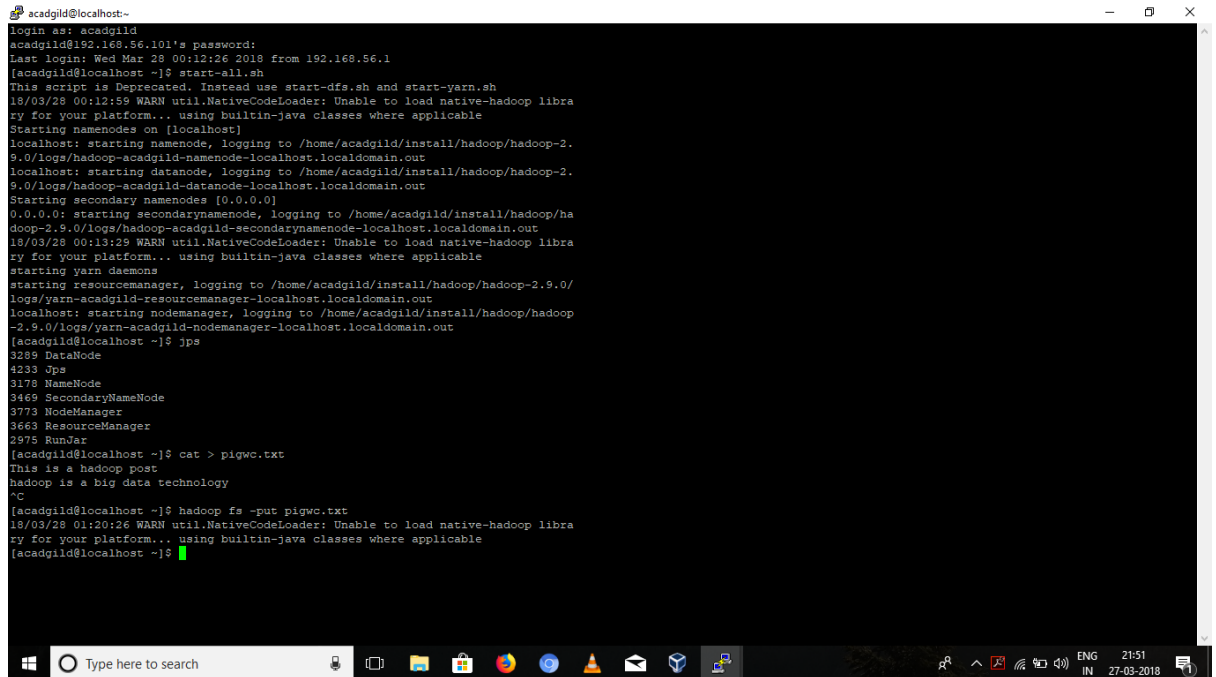
Assignment

Task 1

Word Count program in pig latin.

Command used:

- a) Create a txt file and load it to hdfs : `hadoop fs -put pigwc.txt`



```
acagild@localhost:~$ login as: acagild
acagild@192.168.56.101's password:
Last login: Wed Mar 28 00:12:26 2018 from 192.168.56.1
[acagild@localhost ~]$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
18/03/28 00:12:59 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
localhost: starting namenode, logging to /home/acagild/install/hadoop/hadoop-2.
9.0/logs/hadoop-acagild-namenode-localhost.localdomain.out
localhost: starting datanode, logging to /home/acagild/install/hadoop/hadoop-2.
9.0/logs/hadoop-acagild-datanode-localhost.localdomain.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /home/acagild/install/hadoop/ha
dooop-2.9.0/logs/hadoop-acagild-secondarynamenode-localhost.localdomain.out
18/03/28 00:13:29 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
Starting yarn daemons
starting resourcemanager, logging to /home/acagild/install/hadoop/hadoop-2.9.0/
logs/yarn-acagild-resourcemanager-localhost.localdomain.out
localhost: starting nodemanager, logging to /home/acagild/install/hadoop/hadoop
-2.9.0/logs/yarn-acagild-nodemanager-localhost.localdomain.out
[acagild@localhost ~]$ jps
3289 DataNode
4233 Jps
3178 NameNode
3469 SecondaryNameNode
3773 NodeManager
3663 ResourceManager
2975 RunJar
[acagild@localhost ~]$ cat > pigwc.txt
This is a hadoop post
hadoop is a big data technology
^C
[acagild@localhost ~]$ hadoop fs -put pigwc.txt
18/03/28 01:20:26 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
[acagild@localhost ~]$
```

- b) Write word count program :

`lines = LOAD 'pigwc.txt' AS (line:chararray);` --→ load the data in line

`words = FOREACH lines GENERATE FLATTEN(TOKENIZE(line)) as word;` -→ use
tokenize and segregate words in each line

`grouped = GROUP words BY word;` → group the tuple

`wordcount = FOREACH grouped GENERATE group, COUNT(words);` →count
each word.

`DUMP wordcount;` → execute the map reduce program

[illegible]