**VM Cluster Login:**

To begin, log into the VM cluster at IP 128.110.155.147, with the username ubuntu, using the keyfile.

Once in the ubuntu directory, cd to Hadoop.

The files to run each task are in the file mini1.jar in the hadoop directory.

The .jar file is already in that directory, but if you wanted to compile you would run:

*bin/hadoop com.sun.tools.javac.Main \*.java*

*jar cf mini1.jar \*.class*

The enviromnetal variables should already be set, but if not here are the correct ones:

*export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64*

*export PATH=${JAVA\_HOME}/bin:${PATH}*

*export HADOOP\_CLASSPATH=${JAVA\_HOME}/lib/tools.jar*

**Part 3: Developing a Hadoop program**

the ngram file runs using the following format

bin/hadoop jar ngram.jar ngram inputfile outputfile n

To run the ngram file, put the file you want to analyze in the hdfs folder input2/

For example, I create a new file in the Hadoop directory “file1” using nano

First check if there is an input 2

*bin/hdfs dfs -ls /user/ubuntu/*

if there is, remove it

*bin/hdfs dfs -rmr input2*

then

*bin/hdfs dfs -mkdir input2*

then

*bin/hdfs dfs -put file1 input2*

check it copied

*bin/hdfs dfs -ls /user/ubuntu/input2*

then for a bigram (n=2) run:

*bin/hadoop jar mini1.jar ngram input2/ output/ 2*

and check output

*bin/hdfs dfs -cat output/\**

then delete the output

*bin/hdfs dfs -rmr output*

**Part 4: Developing a Hadoop program to analyze real logs**

The access logs are in the Hadoop file system in /user/ubuntu/input1

The relevant code files are in mini1.jar

**Question 1:**

To determine the hits made to website item “/assets/img/home-logo.png” run problem1 script

*bin/hadoop jar mini1.jar problem1 input1/ output/*

Then go to the output folder

*bin/hdfs dfs -cat output/\**

Which shows

/assets/img/home-logo.png 98744

Thus there are 98744 hits to the item “/assets/img/home-logo.png”

Then delete the output

*bin/hdfs dfs -rmr output*

**Question 2:**

To determine the hits to the ip 10.153.239.5 run problem2 script

*bin/hadoop jar mini1.jar problem2 input1/ output/*

Then go to the output folder

*bin/hdfs dfs -cat output/\**

Which shows

10.153.239.5 547

Thus there are 547 hits to the ip 10.153.239.5

Then delete the output

*bin/hdfs dfs -rmr output*

**Question 3:**

To determine the path with the most hits run problem3 script

*bin/hadoop jar mini1.jar problem3 input1/ output/*

Then go to the output folder

*bin/hdfs dfs -cat output/\**

Which shows

/assets/css/combined.css 117348

Thus “/assets/css/combined.css” was hit the most with 117348 hits

Then delete the output

*bin/hdfs dfs -rmr output*

**Question 4:**

To determine the ip with the most accesses run problem4 script

*bin/hadoop jar mini1.jar problem4 input1/ output/*

Then go to the output folder

*bin/hdfs dfs -cat output/\**

Which shows

10.216.113.172 158614

Thus 10.216.113.172 had the most accesses with 158614

Then delete the output

*bin/hdfs dfs -rmr output*

**Dockerfile**

To build the docker image locally you need to navigate to the directory with the following files:

Dockerfile

bootstrap.sh

core-site.xml

hdfs-site.xml

mapred-site.xml

yarn-site.xml

Then in the directory run where img\_name is whatever you want to call the image:

docker build -t img\_name .

Once it is built, you can start it running by

docker run -it img\_name which starts the yarn processes

The bootstrap script runs indefinitely, but you can access image through docker and run a few commands, for example wordcount:

*$HADOOP\_HOME/bin/hdfs dfs -put $HADOOP\_HOME/etc/hadoop/\*.xml /usr/local/input*

*$HADOOP\_HOME/bin/hdfs dfs -ls /usr/local/input*

*$HADOOP\_HOME/bin/hadoop jar $HADOOP\_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.1.jar wordcount /usr/local/input/ /usr/local/output/*

*$HADOOP\_HOME/bin/hdfs dfs -cat /usr/local/output/\**

*$HADOOP\_HOME/bin/hdfs dfs -rmr /usr/local/output*