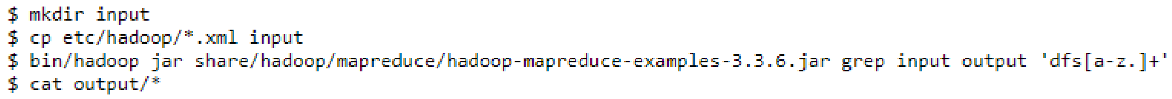
Project diary

11/10/2023 – first proof of concept program, Hadoop installed.

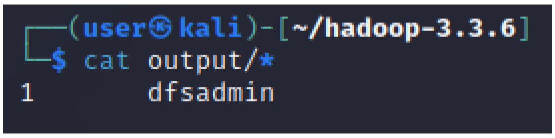
Proof of standalone operation, taken web pages from the Hadoop input, compiled with a regular expression to create an output.

Tried using the standalone operation given on Hadoop documentation website as shown:



```
$ mkdir input
$ cp etc/hadoop/*.xml input
$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.6.jar grep input output 'dfs[a-z.]+'
$ cat output/*
```

The following commands take a list of 8 webpages and searches for a regular expression in the pages, leading to a single result.



```
┌──(user㉿kali)-[~/hadoop-3.3.6]
└─$ cat output/*
1       dfsadmin
```

24/10/2023 – First functional YARN program, Hadoop resourcemanager first use

31/10/2023 – First functional wordcount program, untested

08/11/2023 – Found out JAVA_HOME is defined in /etc/environment and NOT .bashrc

Eclipse only works on java 17+

IF CONNECTION REFUSED PORT 22: sudo service ssh restart

cd Hadoop-3.3.6

bin/Hadoop jar Problem1.jar Problem1 /home/user/git/PROJECT/Code/HelloWorld/src/Input /home/user/git/PROJECT/Code/HelloWorld/src/Output

08/11/2023 – can upload to cluster on localhost

To launch hadoop cluster:

sudo service ssh restart

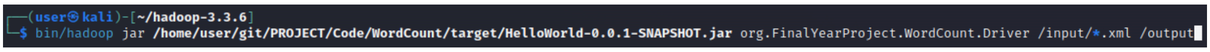
ssh localhost./

cd hadoop-3.3.6/

bin/hdfs namenode -format

sbin/start-dfs.sh

09/11/2023, 02:14:05 – Problem 1 runs on a single node setup. In its entirety.



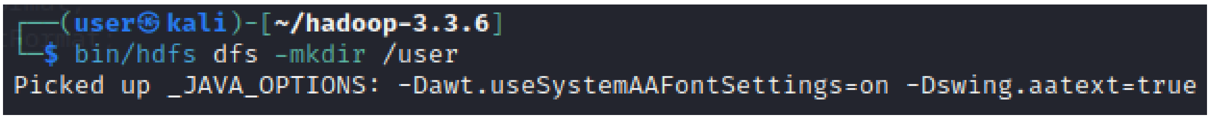
```
┌──(user㉿kali)-[~/hadoop-3.3.6]
└─$ bin/hadoop jar /home/user/git/PROJECT/Code/WordCount/target/HelloWorld-0.0.1-SNAPSHOT.jar org.FinalYearProject.WordCount.Driver /input/*.xml /output
```

STRUCTURE FOR EXECUTING HADOOP JARS

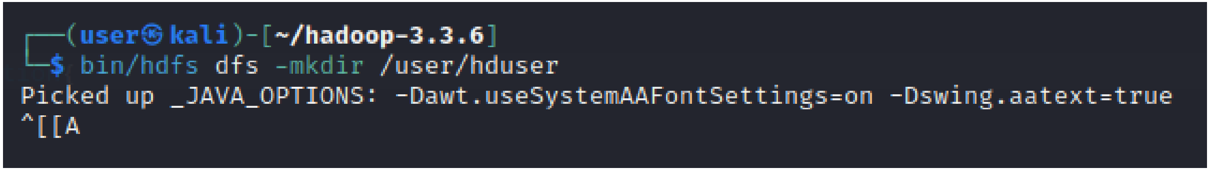
ASSUMING CD {HADOOP_HOME}

bin/hadoop jar <.jar file> <package.class> <input file(s)> <output folder (MUST NOT EXIST)>

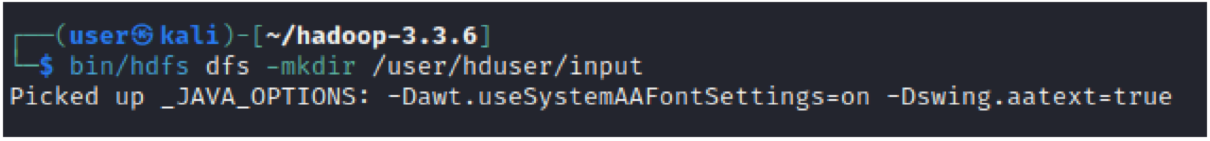
To add files to hadoop HDFS:



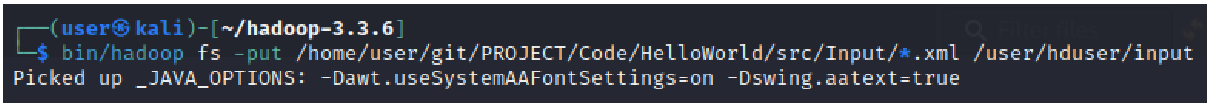
```
┌──(user㉿kali)-[~/hadoop-3.3.6]
└─$ bin/hdfs dfs -mkdir /user
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
```



```
┌──(user㉿kali)-[~/hadoop-3.3.6]
└─$ bin/hdfs dfs -mkdir /user/hduser
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
^[[A
```

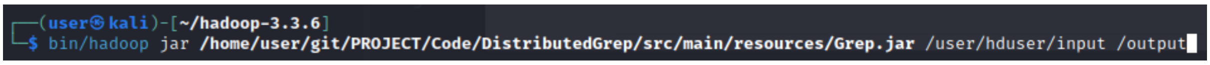


```
┌──(user㉿kali)-[~/hadoop-3.3.6]
└─$ bin/hdfs dfs -mkdir /user/hduser/input
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
```



```
┌──(user㉿kali)-[~/hadoop-3.3.6]
└─$ bin/hadoop fs -put /home/user/git/PROJECT/Code/HelloWorld/src/Input/*.xml /user/hduser/input
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
```

Command used to run problem 2.



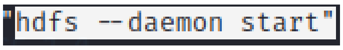
```
┌──(user㉿kali)-[~/hadoop-3.3.6]
└─$ bin/hadoop jar /home/user/git/PROJECT/Code/DistributedGrep/src/main/resources/Grep.jar /user/hduser/input /output
```

Hadoop mapred is DEPRECIATED

./WordCount.sh hdfs:///user/input/data/html/*.html hdfs:///user/output/wordcount

./RegexSearch.sh hdfs:///user/input/data/html/*.html hdfs:///user/output/distributedgrep

^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$



```
"hdfs --daemon start"
```

TO CONNECT NODES

SHARE KEYS WITH SSH:

Sudo service ssh restart

Ssh localhost

Ping IP of node to be added

THEN FOR EVERY NODE:

sudo nano /etc/hosts

paste eth0 -> inet address (192.168.56.xxx)

reboot every node

THEN copy ssh id to other nodes:

Ssh- copy-id user@<IPADDRESS>

Type y

AFTER ALL NODES CAN PING EACH OTHER

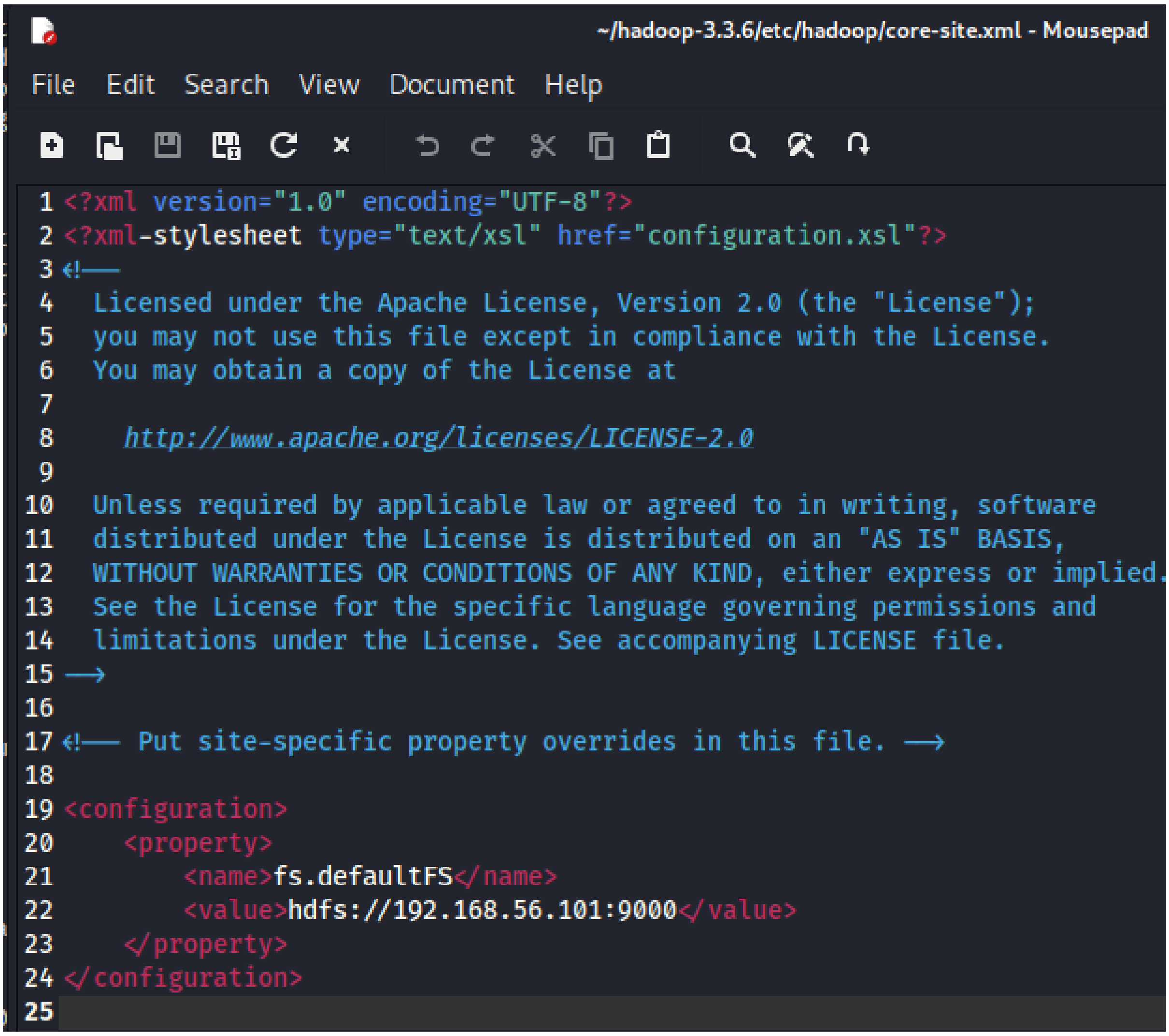
ON EVERY NODE DO THE FOLLOWING COMMANDS:

sbin/stop-all.sh

rm -Rf /tmp/

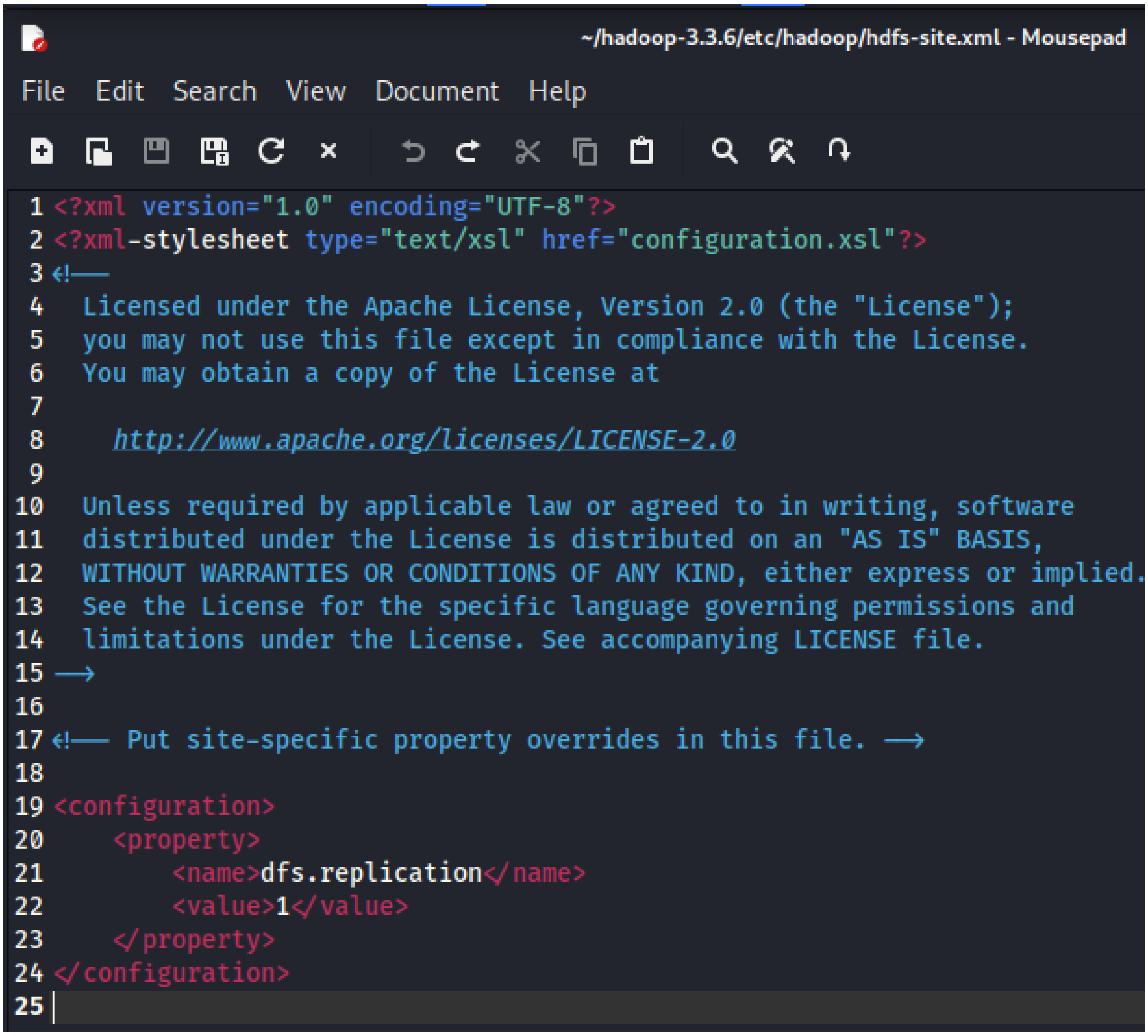
bin/hadoop namenode -format

IDEAL CORE-SITE.XML



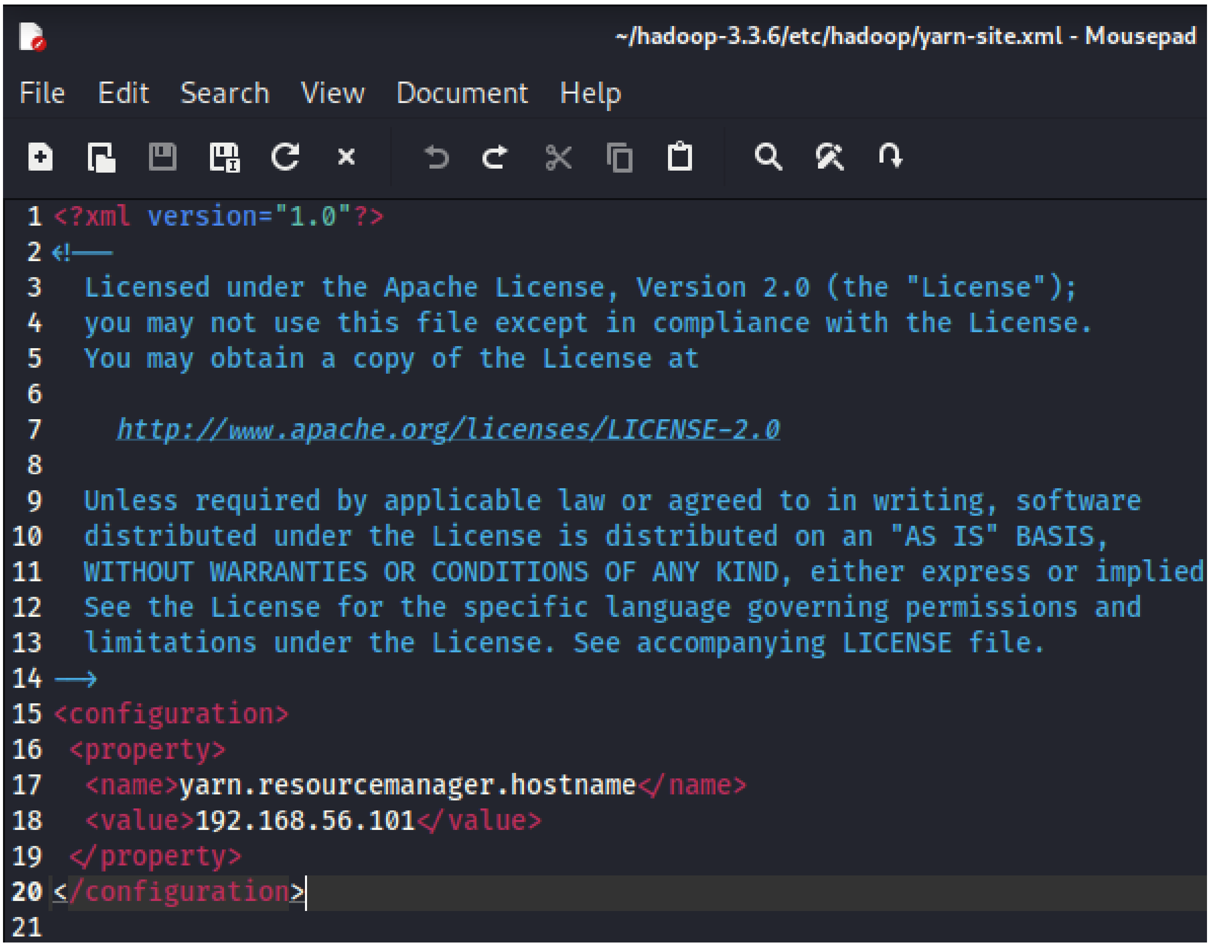
```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://192.168.56.101:9000</value>
    </property>
</configuration>

```

EXAMPLE hdfs-site.xml



```xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
</configuration>

```

EXAMPLE yarn-site.xml



```
<?xml version="1.0"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
<configuration>
 <property>
  <name>yarn.resourcemanager.hostname</name>
  <value>192.168.56.101</value>
 </property>
</configuration>
```

TO LAUNCH THE CLUSTER:

NAMENODE:

Sudo service ssh restart

Ssh localhost

Cd hadoop-3.3.6

rm -Rf /tmp

Bin/hdfs namenode -format

Sbin/start-dfs.sh

Sbin/start-yarn.sh

START DATANODE:

Sudo service ssh restart

Cd hadoop-3.3.6

Bin/hdfs namenode -format

Rm -rf /tmp

hdfs –daemon start datanode

USE “jps” TO CHECK EVERYTHING IS WORKING

TO CONFIGURE A NEW NODE:

```
hostnamectl set-hostname <New-Hostname>

hostnamectl status

sudo nano /etc/hosts

sudo service ssh restart

bin/hdfs namenode -format

hdfs –daemon stop datanode

hdfs –daemon start datanode
```

Frequency Distribution

```
bin/hadoop jar /home/user/git/PROJECT/Code/FrequencyDistribution/target/frequencydistribution-0.0.1-SNAPSHOT.jar main.java.org.finalyearproject.frequencydistribution.GraphBuilder /input /output

bin/hadoop jar /home/user/git/PROJECT/Code/FrequencyDistribution/target/frequencydistribution-0.0.1-SNAPSHOT.jar main.java.org.finalyearproject.frequencydistribution.GraphBuilder /FreqInput /output
```

KMEANS

```
bin/hadoop jar /home/user/git/PROJECT/Code/kmeans/target/kmeans-1.0.0.jar main.java.org.finalyearproject.kmeans.Driver /input/housing /output <Number of iterations>

hadoop fs -copyFromLocal /home/user/git/PROJECT/Code/kmeans/src/main/resources/Input/houseprices/housing.csv /input/housing

hadoop fs -rm -r /output

hadoop fs -copyFromLocal /home/user/git/PROJECT/Code/kmeans/src/main/resources/Input/houseprices/1553768847-housing.csv /input/housing
```

Using 1mb file capped out the 8.8gb filesystem with 60% of a single map. Had to trim it down.