Task 2 Logbook

Setting Up Cluster w/o Kerberos

Nat adapter IP: 192.168.233.0

Set Hostname IP Address

hadoopmaster 192.168.233.3

hadoopworker1 192.168.233.4

Hadoopworker2 192.168.233.5

Created ssh keys

```
hadoop@hadoopnode1:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hadoop/.ssh/id_rsa
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:vBeo1RiS4nFA7o1Cic/GeYBgoRR5IHEBMa8XmlnLSj8 hadoop@hadoopnode1
The key's randomart image is:
+---[RSA 3072]----+
|B@X.             |
|BB.o .           |
|+oX + .          |
|.& @ . + o       |
|*.& o . S .      |
|.=..   o . .     |
|.  E  . . .      |
|    .    .       |
|                 |
+----[SHA256]-----+
```

```
hadoop@hadoopnode2:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hadoop/.ssh/id_rsa
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:5uwbcaxC83RgOSXCO3lVVjsDmoLc2+534KT9noa+5/g hadoop@hadoopnode2
The key's randomart image is:
+---[RSA 3072]----+
|      .. ooo     |
| . o  .++= .     |
|  o o.o+= +      |
|    .++. + o     |
|    ..+.S +      |
|     o Bo=       |
|      o=*o       |
|     ..o=o+.     |
|      .o*@E      |
+----[SHA256]-----+
```

```
hadoop@hadoopmaster:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hadoop/.ssh/id_rsa
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:WGWD8Vo6snIGVgFc8+fzymeehNPdhMy0DO5kK65aHb4 hadoop@hadoopmaster
The key's randomart image is:
+---[RSA 3072]----+
|   ..o+ .o+      |
|    .  +.+ .     |
|     . o + . .   |
|    . o * . * o  |
|   o o S + + * .|
|  . . o + X o o  |
|   . + . * * . .|
|    + . o *o.    |
|     ....E+o     |
+----[SHA256]-----+
```

Ssh-copy-id for each node to the other nodes and themselves

Hosts set for hadoopmaster and respectively the equivalent for the nodes

```
  GNU nano 4.8                                        /etc/hosts
127.0.0.1 localhost
192.168.233.3 hadoopmaster
192.168.233.4 hadoopnode1
192.168.233.5 hadoopnode2

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

/home/hadoop/hdfs/namenode

Config 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.default.name</name>
        <value>hdfs://hadoopmaster:9000</value>
    </property>
</configuration>
```

Hdfs-site.xml

```
  GNU nano 4.8                                hdfs-site.xml
<!--
  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.namenode.name.dir</name>
            <value>/home/hadoop/hdfs/namenode</value>
    </property>

    <property>
            <name>dfs.datanode.data.dir</name>
            <value>/home/hadoop/hdfs/datanode</value>
    </property>

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

Mapred-site config adapted for 2gb ram as well

```
  GNU nano 4.8                                mapred-site.xml
<!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
            <name>mapreduce.framework.name</name>
            <value>yarn</value>
    </property>
    <property>
            <name>yarn.app.mapreduce.am.env</name>
            <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
    </property>
    <property>
            <name>mapreduce.map.env</name>
            <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
    </property>
    <property>
            <name>mapreduce.reduce.env</name>
            <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
    </property>
    <property>
            <name>yarn.app.mapreduce.am.resource.mb</name>
            <value>512</value>
    </property>
    <property>
            <name>mapreduce.map.memory.mb</name>
            <value>256</value>
    </property>
    <property>
            <name>mapreduce.reduce.memory.mb</name>
            <value>256</value>
    </property>
</configuration>
```

Added workers removed local host as worker

```
  GNU nano 4.8                                                    workers
hadoopnode1
hadoopnode2
```

Config yarn with the public ip of the masternode and adapt it for 2gb nodes

```
  GNU nano 4.8                                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.acl.enable</name>
            <value>0</value>
    </property>

    <property>
            <name>yarn.resourcemanager.hostname</name>
            <value>192.168.233.3</value>
    </property>

    <property>
            <name>yarn.nodemanager.aux-services</name>
            <value>mapreduce_shuffle</value>
    </property>

    <property>
            <name>yarn.nodemanager.resource.memory-mb</name>
            <value>1536</value>
```

```
GNU nano 4.8                                    yarn-site.xml
    </property>

    <property>
            <name>yarn.resourcemanager.hostname</name>
            <value>192.168.233.3</value>
    </property>

    <property>
            <name>yarn.nodemanager.aux-services</name>
            <value>mapreduce_shuffle</value>
    </property>

    <property>
            <name>yarn.nodemanager.resource.memory-mb</name>
            <value>1536</value>
    </property>

    <property>
            <name>yarn.scheduler.maximum-allocation-mb</name>
            <value>1536</value>
    </property>

    <property>
            <name>yarn.scheduler.minimum-allocation-mb</name>
            <value>128</value>
    </property>

    <property>
            <name>yarn.nodemanager.vmem-check-enabled</name>
            <value>false</value>
    </property>
</configuration>
```

Created datanode folders for hadoopworker1+2

/home/hadoop/hdfs/datanode

First made a tar file using: tar –zcvf hadoop.tar.gz $HADOOP_HOME

Trasfered to other nodes using

Scp hadoop.tar.gz hadoopnode1:/home/hadoop

Scp hadoop.tar.gz hadoopnode2:/home/hadoop

Unziped using: tar –xzf hadoop

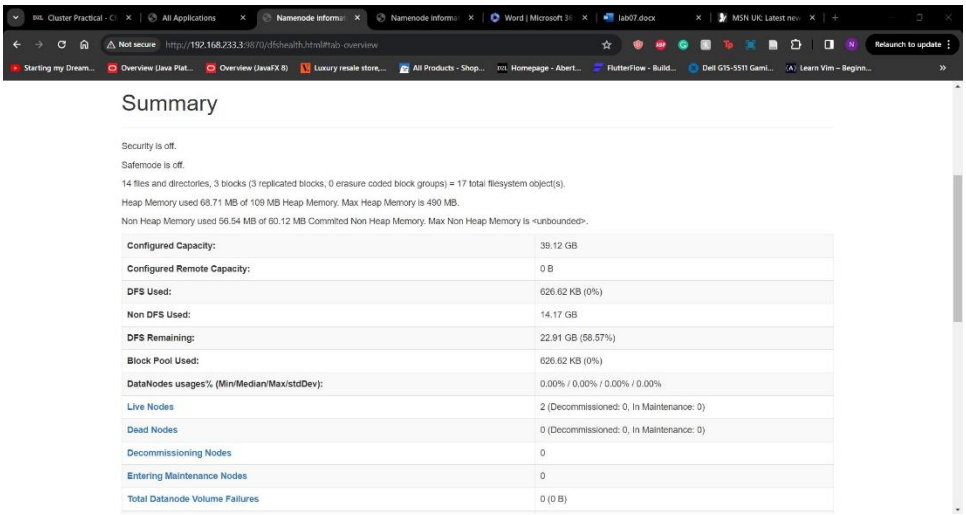
Also found scp –r hadoop-3.3.0/ hadoopnode1:/home/hadoop

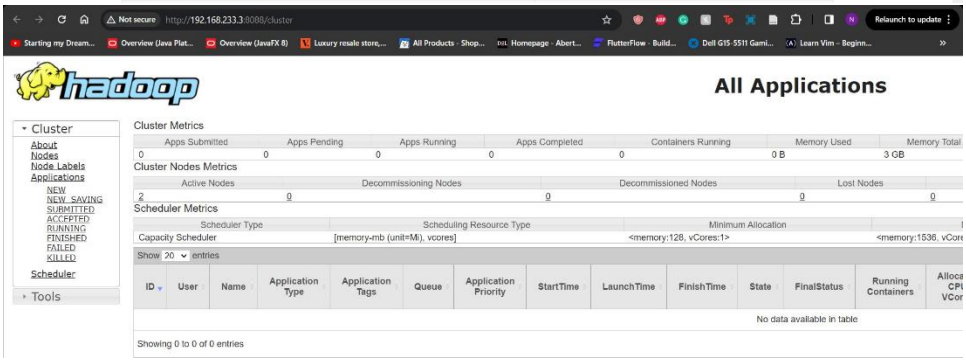
scp –r hadoop-3.3.0/ hadoopnode2:/home/hadoop

To recursively transfer the whole folder to the nodes

Permisions of datanode directory were set to root so there were no love nodes changed the permission and made the directory without using sudo

Now 2 live nodes therefore basic cluster opeational





Running a mapred job using

cd $HADOOP_HOME/share/hadoop/mapreduce

yarn jar hadoop-mapreduce-examples-3.3.0.jar pi 16 1000

Hadoop Master | Hadoop Worker 1 | Hadoop Worker 2

```
                                Map output bytes=288
                                Map output materialized bytes=448
                                Input split bytes=2406
                                Combine input records=0
                                Combine output records=0
                                Reduce input groups=2
                                Reduce shuffle bytes=448
                                Reduce input records=32
                                Reduce output records=0
                                Spilled Records=64
                                Shuffled Maps =16
                                Failed Shuffles=0
                                Merged Map outputs=16
                                GC time elapsed (ms)=4290
                                CPU time spent (ms)=22570
                                Physical memory (bytes) snapshot=4305768448
                                Virtual memory (bytes) snapshot=34833641472
                                Total committed heap usage (bytes)=3013607424
                                Peak Map Physical memory (bytes)=281616384
                                Peak Map Virtual memory (bytes)=2051584000
                                Peak Reduce Physical memory (bytes)=159313920
                                Peak Reduce Virtual memory (bytes)=2057175040
                        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=1888
                        File Output Format Counters
                                Bytes Written=97
                Job Finished in 54.435 seconds
                Estimated value of Pi is 3.14250000000000000000
                hadoop@hadoopmaster:~/hadoop-3.3.0/share/hadoop/mapreduce$
```

Setting Up Spark

```
nano ~/.bashrc
```

Then

```
export SPARK_HOME=/opt/spark
export PATH=$PATH:$SPARK_HOME/bin
```

Then

```
source ~/.bashrc
```

Running

```
hadoop@hadoopnode2:~$ spark-shell
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/home/hadoop/spark-3.0.
0/jars/spark-unsafe_2.12-3.0.0.jar) to constructor java.nio.DirectByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operation
s
WARNING: All illegal access operations will be denied in a future release
23/11/23 15:50:32 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... u
sing builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://hadoopnode2:4040
Spark context available as 'sc' (master = local[*], app id = local-1700754640648).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.0.0
      /_/

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 11.0.8)
Type in expressions to have them evaluated.
Type :help for more information.
```

Scala app operations

Installed sbt through the guide on mylearningspace

```
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed 29 Nov 2023 06:02:05 PM UTC

  System load:  0.0                Processes:              200
  Usage of /:   33.5% of 19.56GB   Users logged in:        1
  Memory usage: 13%                IPv4 address for ens33: 192.168.233.3
  Swap usage:   0%                 IPv4 address for ens33: 192.168.233.132


306 updates can be installed immediately.
193 of these updates are security updates.
To see these additional updates run: apt list --upgradable

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Wed Nov 29 17:55:56 2023
hadoop@hadoopmaster:~$ echo "deb https://repo.scala-sbt.org/scalasbt/debian all main" | sudo tee /etc/apt/sources.list.d/sbt.list
[sudo] password for hadoop:
deb https://repo.scala-sbt.org/scalasbt/debian all main
hadoop@hadoopmaster:~$ echo "deb https://repo.scala-sbt.org/scalasbt/debian /" | sudo tee /etc/apt/sources.list.d/sbt_old.list
deb https://repo.scala-sbt.org/scalasbt/debian /
hadoop@hadoopmaster:~$ curl -sL "https://keyserver.ubuntu.com/pks/lookup?op=get&search=0x2EE0EA64E40A89B84B2DF73499E82A75642AC823" | sudo -H gpg --no-defaul
t-keyring --keyring gnupg-ring:/etc/apt/trusted.gpg.d/scalasbt-release.gpg --import
gpg: keyring '/etc/apt/trusted.gpg.d/scalasbt-release.gpg' created
gpg: directory '/root/.gnupg' created
gpg: /root/.gnupg/trustdb.gpg: trustdb created
gpg: key 99E82A75642AC823: public key "sbt build tool <scalasbt@gmail.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1
hadoop@hadoopmaster:~$ sudo chmod 644 /etc/apt/trusted.gpg.d/scalasbt-release.gpg
```

```
gpg: Total number processed: 1
gpg:               imported: 1
hadoop@hadoopmaster:~$ sudo chmod 644 /etc/apt/trusted.gpg.d/scalasbt-release.gpg
hadoop@hadoopmaster:~$ sudo apt-get update
Hit:1 http://gb.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://gb.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Hit:3 http://gb.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:4 http://gb.archive.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:5 http://gb.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1,137 kB]
Get:8 http://gb.archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [272 kB]
Get:9 http://gb.archive.ubuntu.com/ubuntu focal-security/universe amd64 Packages [911 kB]
Get:10 http://gb.archive.ubuntu.com/ubuntu focal-security/universe Translation-en [191 kB]
Get:6 https://scala.jfrog.io/artifactory/debian all InRelease [3,558 B]
Ign:7 https://scala.jfrog.io/artifactory/debian  InRelease
Get:12 https://scala.jfrog.io/artifactory/debian all/main amd64 Packages [2,594 B]
Get:11 https://scala.jfrog.io/artifactory/debian  Release [815 B]
Get:13 https://scala.jfrog.io/artifactory/debian  Release.gpg [821 B]
Get:14 https://scala.jfrog.io/artifactory/debian  Packages [5,122 B]
Fetched 2,752 kB in 2s (1,136 kB/s)
Reading package lists... Done
hadoop@hadoopmaster:~$ sudo apt-get install sbt
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  sbt
0 upgraded, 1 newly installed, 0 to remove and 265 not upgraded.
Need to get 20.0 kB of archives.
After this operation, 50.2 kB of additional disk space will be used.
Get:1 https://scala.jfrog.io/artifactory/debian all/main amd64 sbt all 1.9.7 [20.0 kB]
Fetched 20.0 kB in 1s (15.1 kB/s)
Selecting previously unselected package sbt.
(Reading database ... 71591 files and directories currently installed.)
Preparing to unpack .../apt/archives/sbt_1.9.7_all.deb ...
Unpacking sbt (1.9.7) ...
Setting up sbt (1.9.7) ...
Creating system group: sbt
Creating system user: sbt in sbt with sbt daemon-user and shell /bin/false
Processing triggers for man-db (2.9.1-1) ...
```

Wrote the scala app in the src/main/scala/ directory and it is named LanguageDetectionApp.scala

```
hadoop@hadoopmaster:~$ cd src/main/scala/
hadoop@hadoopmaster:~/src/main/scala$ ls
LanguageDetectionApp.scala
hadoop@hadoopmaster:~/src/main/scala$
```

Putting books and top 1000 words for German English and French into hdfs using -put

```
hadoop@hadoopmaster:~/src/main/scala$ hdfs dfs -ls -R
drwxr-xr-x   - hadoop supergroup          0 2023-12-04 15:15 books
-rw-r--r--   1 hadoop supergroup      22657 2023-12-04 15:15 books/Bonhomme.txt
-rw-r--r--   1 hadoop supergroup     255648 2023-12-04 15:15 books/CoerDePantere.txt
-rw-r--r--   1 hadoop supergroup     130102 2023-12-04 15:15 books/Labancale.txt
-rw-r--r--   1 hadoop supergroup     304122 2023-12-04 15:15 books/LePoeteAssassine.txt
-rw-r--r--   1 hadoop supergroup     214461 2023-12-04 15:15 books/Lecrivain.txt
-rw-r--r--   1 hadoop supergroup     239018 2023-12-04 15:15 books/Vathek.txt
-rw-r--r--   1 hadoop supergroup     106237 2023-12-04 15:15 books/arabiannights.txt
-rw-r--r--   1 hadoop supergroup     342190 2023-12-04 15:15 books/artofwar.txt
-rw-r--r--   1 hadoop supergroup     176281 2023-12-04 15:15 books/battle.txt
-rw-r--r--   1 hadoop supergroup     523234 2023-12-04 15:15 books/goldenhour.txt
-rw-r--r--   1 hadoop supergroup     225727 2023-12-04 15:15 books/happyones.txt
-rw-r--r--   1 hadoop supergroup     254499 2023-12-04 15:15 books/kafir.txt
-rw-r--r--   1 hadoop supergroup     314769 2023-12-04 15:15 books/kak.txt
-rw-r--r--   1 hadoop supergroup     170837 2023-12-04 15:15 books/laffaire.txt
-rw-r--r--   1 hadoop supergroup      54698 2023-12-04 15:15 books/lastdays.txt
-rw-r--r--   1 hadoop supergroup     996324 2023-12-04 15:15 books/marquis.txt
-rw-r--r--   1 hadoop supergroup     905558 2023-12-04 15:15 books/nabob.txt
-rw-r--r--   1 hadoop supergroup     255809 2023-12-04 15:15 books/oz.txt
-rw-r--r--   1 hadoop supergroup     225727 2023-12-04 15:15 books/selfbibliography.txt
-rw-r--r--   1 hadoop supergroup     352992 2023-12-04 15:15 books/spellingsae.txt
-rw-r--r--   1 hadoop supergroup     714619 2023-12-04 15:15 books/wilhelm.txt
drwxr-xr-x   - hadoop supergroup          0 2023-12-04 15:16 languages
-rw-r--r--   1 hadoop supergroup       5845 2023-12-04 15:16 languages/eng.txt
-rw-r--r--   1 hadoop supergroup       7142 2023-12-04 15:16 languages/fr.txt
-rw-r--r--   1 hadoop supergroup       7028 2023-12-04 15:16 languages/ger.txt
```

Writing scala app code

```
  GNU nano 4.8                                                  LanguageDetectionApp.scal
import scala.io.Source
import org.apache.spark.SparkContext
import org.apache.spark.SparkContext._
import org.apache.spark.SparkConf
import scala.collection.immutable._

object LanguageDetectionApp {

    def main(args: Array[String]) {
        val conf = new SparkConf().setAppName("Language Detection App")
        val sc = new SparkContext(conf)
        val bookfiles = sc.textFile("hdfs://hadoopmaster:9000/user/hadoop/books/*.txt")
        val engdict = sc.textFile("hdfs://hadoopmaster:9000/user/hadoop/languages/eng.txt")
        val frdict = sc.textFile("hdfs://hadoopmaster:9000/user/hadoop/languages/fr.txt")
        val gerdict = sc.textFile("hdfs://hadoopmaster:9000/user/hadoop/languages/ger.txt")
        val bookwords = bookfiles.flatMap(x=> x.split(" "))
        val bookwordsPerFiles = bookwords.map(x=>(x,1))
        val reduce =  bookwordsPerFiles.reduceByKey(_+_)
        val engmap = engdict.flatMap(x=> x.split("\n"))
        val frmap = frdict.flatMap(x=> x.split("\n"))
        val germap = gerdict.flatMap(x=> x.split("\n"))
        val reducetupple = reduce.collect()
        //var finallist = ("Eng" -> "0","Fr" -> "0", "Ger" ->"0")
        var engnumber = 0
        var frnumber = 0
        var gernumber = 0
        for (e <- engmap.collect()){
        var checke = reduce.lookup(e)
        var existse = checke.isEmpty
        if(!existse)
        {
        engnumber = engnumber + checke(0)
        }
        }
        for (f <- frmap.collect()){
```

```
GNU nano 4.8
        for (f <- frmap.collect()){
        var checkf = reduce.lookup(f)
        var  existsf = checkf.isEmpty
        if(!existsf)
        {
        frnumber = frnumber + checkf(0)
        }

        }
        for (g <- germap.collect()){
        var checkg = reduce.lookup(g)
        var  existsg = checkg.isEmpty
        if(!existsg)
        {
        gernumber = gernumber + checkg(0)
        }
        }


        var toplang = ""
        var topwords = 0
        //bookwordsPerFiles.collect().foreach(println)
        //finallist.foreach(println)
        if ((engnumber>frnumber) && (engnumber>gernumber))
        {
        toplang = "English"
        topwords = engnumber
        }
        if ((frnumber>engnumber) && (frnumber>gernumber))
        {
        toplang = "French"
        topwords = frnumber
        }
        if ((gernumber>frnumber) && (gernumber>engnumber))
        {
        toplang = "German"
```

```
GNU nano 4.8                                                    Lang
        //bookwordsPerFiles.collect().foreach(println)
        //finallist.foreach(println)
        if ((engnumber>frnumber) && (engnumber>gernumber))
        {
        toplang = "English"
        topwords = engnumber
        }
        if ((frnumber>engnumber) && (frnumber>gernumber))
        {
        toplang = "French"
        topwords = frnumber
        }
        if ((gernumber>frnumber) && (gernumber>engnumber))
        {
        toplang = "German"
        topwords = gernumber
        }

        printf("There are %d", engnumber)
        println(" english words.")
        printf("There are %d", frnumber)
        println(" french words.")
        printf("There are %d", gernumber)
        println(" german words.")
        print("Therefore the most used language in the directory is ")
        print(toplang)
        print(" with a total of ")
        print(topwords)
        println(" words used.")
        //engmap.collect().foreach(println)
        //frmap.collect().foreach(println)
        //reduce.collect().foreach(println)

    }
}
```

Simple.sbt file

```
  GNU nano 4.8
name := "Language Detection App"
version := "1.0"
scalaVersion := "2.12.10"
libraryDependencies += "org.apache.spark" %% "spark-core" % "3.0.0"
```

Running scala app

```
spark-submit --class=LanguageDetectionApp target/scala-2.12/language-detection-app_2.12-1.0.jar
```

```
23/12/04 15:52:12 INFO DAGScheduler: Final stage: ResultStage 6010 (lookup at LanguageDetectionApp.scala:45)
23/12/04 15:52:12 INFO DAGScheduler: Parents of final stage: List(ShuffleMapStage 6009)
23/12/04 15:52:12 INFO DAGScheduler: Missing parents: List()
23/12/04 15:52:12 INFO DAGScheduler: Submitting ResultStage 6010 (ShuffledRDD[10] at reduceByKey at LanguageDet
ents
23/12/04 15:52:12 INFO MemoryStore: Block broadcast_3011 stored as values in memory (estimated size 4.2 KiB, fr
23/12/04 15:52:12 INFO MemoryStore: Block broadcast_3011_piece0 stored as bytes in memory (estimated size 2.5 K
23/12/04 15:52:12 INFO BlockManagerInfo: Added broadcast_3011_piece0 in memory on hadoopmaster:35071 (size: 2.5
23/12/04 15:52:12 INFO SparkContext: Created broadcast 3011 from broadcast at DAGScheduler.scala:1200
23/12/04 15:52:12 INFO DAGScheduler: Submitting 1 missing tasks from ResultStage 6010 (ShuffledRDD[10] at reduc
st 15 tasks are for partitions Vector(0))
23/12/04 15:52:12 INFO TaskSchedulerImpl: Adding task set 6010.0 with 1 tasks
23/12/04 15:52:12 INFO TaskSetManager: Starting task 0.0 in stage 6010.0 (TID 3050, hadoopmaster, executor driv
23/12/04 15:52:12 INFO Executor: Running task 0.0 in stage 6010.0 (TID 3050)
23/12/04 15:52:12 INFO ShuffleBlockFetcherIterator: Getting 21 (108.8 KiB) non-empty blocks including 21 (108.8
.0 B) remote blocks
23/12/04 15:52:12 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
23/12/04 15:52:12 INFO Executor: Finished task 0.0 in stage 6010.0 (TID 3050). 1432 bytes result sent to driver
23/12/04 15:52:12 INFO TaskSetManager: Finished task 0.0 in stage 6010.0 (TID 3050) in 12 ms on hadoopmaster (e
23/12/04 15:52:12 INFO TaskSchedulerImpl: Removed TaskSet 6010.0, whose tasks have all completed, from pool
23/12/04 15:52:12 INFO DAGScheduler: ResultStage 6010 (lookup at LanguageDetectionApp.scala:45) finished in 0.0
23/12/04 15:52:12 INFO DAGScheduler: Job 3006 is finished. Cancelling potential speculative or zombie tasks for
23/12/04 15:52:12 INFO TaskSchedulerImpl: Killing all running tasks in stage 6010: Stage finished
23/12/04 15:52:12 INFO DAGScheduler: Job 3006 finished: lookup at LanguageDetectionApp.scala:45, took 0.014760
There are 360439 english words.
There are 193629 french words.
There are 240882 german words.
Therefore the most used language in the directory is English with a total of 360439 words used.
23/12/04 15:52:12 INFO SparkContext: Invoking stop() from shutdown hook
23/12/04 15:52:12 INFO SparkUI: Stopped Spark web UI at http://hadoopmaster:4040
23/12/04 15:52:12 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
23/12/04 15:52:12 INFO MemoryStore: MemoryStore cleared
23/12/04 15:52:12 INFO BlockManager: BlockManager stopped
23/12/04 15:52:12 INFO BlockManagerMaster: BlockManagerMaster stopped
23/12/04 15:52:12 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped
23/12/04 15:52:12 INFO SparkContext: Successfully stopped SparkContext
23/12/04 15:52:12 INFO ShutdownHookManager: Shutdown hook called
23/12/04 15:52:12 INFO ShutdownHookManager: Deleting directory /tmp/spark-9672b925-641d-4fb0-a501-a6d6dc868542
23/12/04 15:52:12 INFO ShutdownHookManager: Deleting directory /tmp/spark-3ede9228-0dff-4320-ad40-031902f6b8e7
```

Setting Up Kerberos

Domain: nikosledakis.com

Setting hosts

```
  GNU nano 4.8                                  /etc/hosts
127.0.0.1 localhost
192.168.233.3 hadoopmaster hadoopmaster.nikosledakis.com
192.168.233.4 hadoopnode1 hadoopnode1.nikosledakis.com
192.168.233.5 hadoopnode2 hadoopnode2.nikosledakis.com

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Replicating to all nodes

Add yarn and hdfs users with password:hadoop

```
hadoop@hadoopmaster:~$ sudo adduser yarn
[sudo] password for hadoop:
Adding user `yarn' ...
Adding new group `yarn' (1001) ...
Adding new user `yarn' (1001) with group `yarn' ...
Creating home directory `/home/yarn' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for yarn
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
hadoop@hadoopmaster:~$ sudo adduser yarn hadoop
Adding user `yarn' to group `hadoop' ...
Adding user yarn to group hadoop
Done.
```

```
hadoop@hadoopmaster:~$ sudo adduser hdfs
Adding user `hdfs' ...
Adding new group `hdfs' (1002) ...
Adding new user `hdfs' (1002) with group `hdfs' ...
Creating home directory `/home/hdfs' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for hdfs
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
hadoop@hadoopmaster:~$ sudo adduser hdfs hadoop
Adding user `hdfs' to group `hadoop' ...
Adding user hdfs to group hadoop
Done.
```

Same for node1,2

Hadoop Master | **Hadoop Worker 1** | Hadoop Worker 2

```
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for yarn
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
hadoop@hadoopnode1:~$ sudo adduser yarn hadoop
Adding user `yarn' to group `hadoop' ...
Adding user yarn to group hadoop
Done.
hadoop@hadoopnode1:~$ sudo adduser hdfs
Adding user `hdfs' ...
Adding new group `hdfs' (1002) ...
Adding new user `hdfs' (1002) with group `hdfs' ...
Creating home directory `/home/hdfs' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for hdfs
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
hadoop@hadoopnode1:~$ sudo adduser hdfs hadoop
Adding user `hdfs' to group `hadoop' ...
Adding user hdfs to group hadoop
Done.
hadoop@hadoopnode1:~$ _
```

Hadoop Master | Hadoop Worker 1 | **Hadoop Worker 2**

```
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for yarn
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
hadoop@hadoopnode2:~$ sudo adduser yarn hadoop
Adding user `yarn' to group `hadoop' ...
Adding user yarn to group hadoop
Done.
hadoop@hadoopnode2:~$ sudo adduser hdfs
Adding user `hdfs' ...
Adding new group `hdfs' (1002) ...
Adding new user `hdfs' (1002) with group `hdfs' ...
Creating home directory `/home/hdfs' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for hdfs
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
hadoop@hadoopnode2:~$ sudo adduser hdfs hadoop
Adding user `hdfs' to group `hadoop' ...
Adding user hdfs to group hadoop
Done.
hadoop@hadoopnode2:~$
```

Installing Kerberos

sudo apt-get install krb5-kdc

Default realm: NIKOSLEDAKIS.COM

Servers for the realm: hadoopmaster.nikosledakis.com

Administrative server details: hadoopmaster.nikosledakis.com

sudo apt-get install krb5-admin-server

sudo krb5_newrealm

Password: hadoop

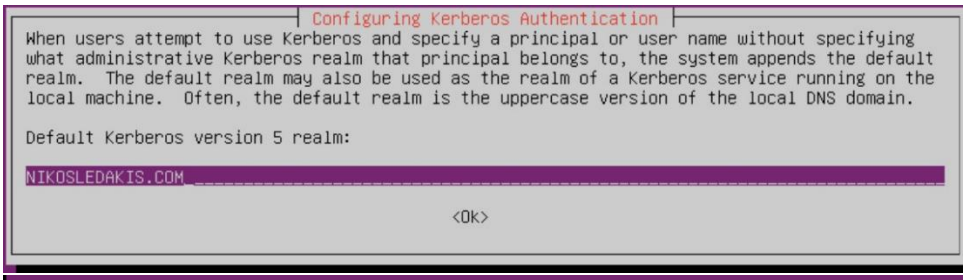
```
Preparing to unpack .../krb5-admin-server_1.17-6ubuntu4_amd64.deb ...
Unpacking krb5-admin-server (1.17-6ubuntu4) ...
Setting up krb5-admin-server (1.17-6ubuntu4) ...
Created symlink /etc/systemd/system/multi-user.target.wants/krb5-admin-server.service → /lib/systemd
/system/krb5-admin-server.service.
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for systemd (245.4-4ubuntu3.2) ...
hadoop@hadoopmaster:~$ sudo krb5_newrealm
This script should be run on the master KDC/admin server to initialize
a Kerberos realm.  It will ask you to type in a master key password.
This password will be used to generate a key that is stored in
/etc/krb5kdc/stash.  You should try to remember this password, but it
is much more important that it be a strong password than that it be
remembered.  However, if you lose the password and /etc/krb5kdc/stash,
you cannot decrypt your Kerberos database.
Loading random data
Initializing database '/var/lib/krb5kdc/principal' for realm 'NIKOSLEDAKIS.COM',
master key name 'K/M@NIKOSLEDAKIS.COM'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key:
Re-enter KDC database master key to verify:


Now that your realm is set up you may wish to create an administrative
principal using the addprinc subcommand of the kadmin.local program.
Then, this principal can be added to /etc/krb5kdc/kadm5.acl so that
you can use the kadmin program on other computers.  Kerberos admin
principals usually belong to a single user and end in /admin.  For
example, if jruser is a Kerberos administrator, then in addition to
the normal jruser principal, a jruser/admin principal should be
created.

Don't forget to set up DNS information so your clients can find your
KDC and admin servers.  Doing so is documented in the administration
guide.
hadoop@hadoopmaster:~$ _
```

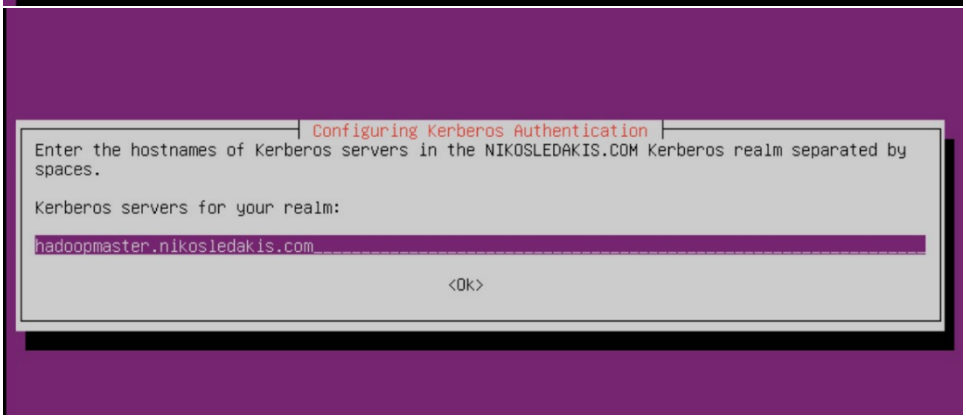
Kadmin add princ

```
kadmin.local:  addprinc hadoop/admin@NIKOSLEDAKIS.COM
WARNING: no policy specified for hadoop/admin@NIKOSLEDAKIS.COM; defaulting to no policy
Enter password for principal "hadoop/admin@NIKOSLEDAKIS.COM":
Re-enter password for principal "hadoop/admin@NIKOSLEDAKIS.COM":
Principal "hadoop/admin@NIKOSLEDAKIS.COM" created.
kadmin.local:  listprincs
K/M@NIKOSLEDAKIS.COM
hadoop/admin@NIKOSLEDAKIS.COM
kadmin/admin@NIKOSLEDAKIS.COM
kadmin/changepw@NIKOSLEDAKIS.COM
kadmin/hadoopmaster.nikosledakis.com@NIKOSLEDAKIS.COM
kiprop/hadoopmaster.nikosledakis.com@NIKOSLEDAKIS.COM
krbtgt/NIKOSLEDAKIS.COM@NIKOSLEDAKIS.COM
```

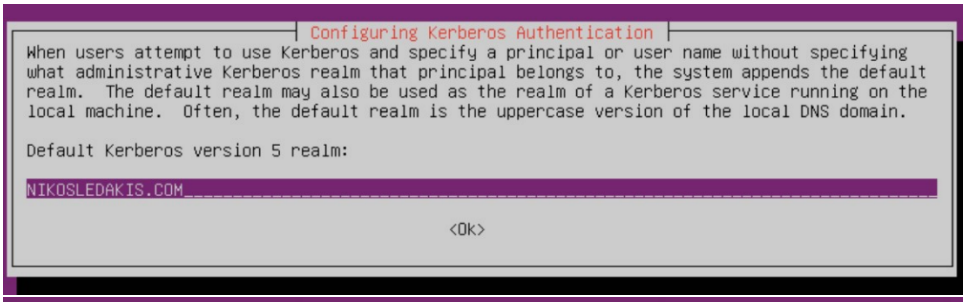
Setup on workers

node1





Node2



Principal creation

```
WARNING: no policy specified for HTTP/hadoopnode1.nikosledakis.com@NIKOSLEDAKIS.COM; defaulting to n
o policy
Principal "HTTP/hadoopnode1.nikosledakis.com@NIKOSLEDAKIS.COM" created.
kadmin.local:  addprinc -randkey HTTP/hadoopnode2.nikosledakis.com@NIKOSLEDAKIS.COM
WARNING: no policy specified for HTTP/hadoopnode2.nikosledakis.com@NIKOSLEDAKIS.COM; defaulting to n
o policy
Principal "HTTP/hadoopnode2.nikosledakis.com@NIKOSLEDAKIS.COM" created.
kadmin.local:  addprinc -randkey yarn/hadoopmaster.nikosledakis.com@NIKOSLEDAKIS.COM
WARNING: no policy specified for yarn/hadoopmaster.nikosledakis.com@NIKOSLEDAKIS.COM; defaulting to
no policy
Principal "yarn/hadoopmaster.nikosledakis.com@NIKOSLEDAKIS.COM" created.
kadmin.local:  addprinc -randkey yarn/hadoopnode1.nikosledakis.com@NIKOSLEDAKIS.COM
WARNING: no policy specified for yarn/hadoopnode1.nikosledakis.com@NIKOSLEDAKIS.COM; defaulting to n
o policy
Principal "yarn/hadoopnode1.nikosledakis.com@NIKOSLEDAKIS.COM" created.
kadmin.local:  addprinc -randkey yarn/hadoopnode2.nikosledakis.com@NIKOSLEDAKIS.COM
WARNING: no policy specified for yarn/hadoopnode2.nikosledakis.com@NIKOSLEDAKIS.COM; defaulting to n
o policy
Principal "yarn/hadoopnode2.nikosledakis.com@NIKOSLEDAKIS.COM" created.
kadmin.local:  listprincs
HTTP/hadoopmaster.nikosledakis.com@NIKOSLEDAKIS.COM
HTTP/hadoopnode1.nikosledakis.com@NIKOSLEDAKIS.COM
HTTP/hadoopnode2.nikosledakis.com@NIKOSLEDAKIS.COM
K/M@NIKOSLEDAKIS.COM
hadoop/admin@NIKOSLEDAKIS.COM
hdfs/hadoopmaster.nikosledakis.com@NIKOSLEDAKIS.COM
hdfs/hadoopnode1.nikosledakis.com@NIKOSLEDAKIS.COM
hdfs/hadoopnode2.nikosledakis.com@NIKOSLEDAKIS.COM
kadmin/admin@NIKOSLEDAKIS.COM
kadmin/changepw@NIKOSLEDAKIS.COM
kadmin/hadoopmaster.nikosledakis.com@NIKOSLEDAKIS.COM
kiprop/hadoopmaster.nikosledakis.com@NIKOSLEDAKIS.COM
krbtgt/NIKOSLEDAKIS.COM@NIKOSLEDAKIS.COM
yarn/hadoopmaster.nikosledakis.com@NIKOSLEDAKIS.COM
yarn/hadoopnode1.nikosledakis.com@NIKOSLEDAKIS.COM
yarn/hadoopnode2.nikosledakis.com@NIKOSLEDAKIS.COM
```

Keytabs created

Scp command Permission denied so Scped them to hadoopnode1 and hadoopnode2 home folder

```
hadoop@hadoopmaster:~$ sudo scp /opt/keytabs/hdfs-hadoopnode1.service.keytab hadoop@hadoopnode1:/hom
e/hadoop/hdfs.service.keytab
hadoop@hadoopnode1's password:
hdfs-hadoopnode1.service.keytab                                   100%  204    98.7KB/s   00:00
hadoop@hadoopmaster:~$ sudo scp /opt/keytabs/yarn-hadoopnode1.service.keytab hadoop@hadoopnode1:/hom
e/hadoop/yarn.service.keytab
hadoop@hadoopnode1's password:
yarn-hadoopnode1.service.keytab                                   100%  204   141.8KB/s   00:00
hadoop@hadoopmaster:~$ sudo scp /opt/keytabs/HTTP-hadoopnode1.service.keytab hadoop@hadoopnode1:/hom
e/hadoop/spnego.service.keytab
hadoop@hadoopnode1's password:
HTTP-hadoopnode1.service.keytab                                   100%  204   312.8KB/s   00:00
hadoop@hadoopmaster:~$ sudo scp /opt/keytabs/hdfs-hadoopnode2.service.keytab hadoop@hadoopnode2:/hom
e/hadoop/hdfs.service.keytab
hadoop@hadoopnode2's password:
hdfs-hadoopnode2.service.keytab                                   100%  204   293.2KB/s   00:00
hadoop@hadoopmaster:~$ sudo scp /opt/keytabs/HTTP-hadoopnode2.service.keytab hadoop@hadoopnode2:/hom
e/hadoop/spnego.service.keytab
hadoop@hadoopnode2's password:
HTTP-hadoopnode2.service.keytab                                   100%  204   185.5KB/s   00:00
hadoop@hadoopmaster:~$ sudo scp /opt/keytabs/yarn-hadoopnode2.service.keytab hadoop@hadoopnode2:/hom
e/hadoop/yarn.service.keytab
hadoop@hadoopnode2's password:
yarn-hadoopnode2.service.keytab                                   100%  204   226.1KB/s   00:00
hadoop@hadoopmaster:~$ _
```

then moved them to the correct directory and changed directory/renamed on hadoopmaster aswell

```
hadoop@hadoopmaster:/etc/security/keytabs$ ls
hdfs.service.keytab  spnego.service.keytab  yarn.service.keytab
hadoop@hadoopmaster:/etc/security/keytabs$ _
```

```
hadoop@hadoopnode1:/etc/security/keytabs$ ls
hdfs.service.keytab  spnego.service.keytab  yarn.service.keytab
hadoop@hadoopnode1:/etc/security/keytabs$ _
```

```
hadoop@hadoopnode2:/etc/security/keytabs$ ls
hdfs.service.keytab  spnego.service.keytab  yarn.service.keytab
hadoop@hadoopnode2:/etc/security/keytabs$
```

Set permisions on the nodes and hadoopmaster

```
hadoop@hadoopmaster:/etc/security/keytabs$ sudo chown hdfs:hadoop /etc/security/keytabs/hdfs.service
.keytab
hadoop@hadoopmaster:/etc/security/keytabs$ sudo chmod 400 /etc/security/keytabs/hdfs.service.keytab
hadoop@hadoopmaster:/etc/security/keytabs$ sudo chown yarn:hadoop /etc/security/keytabs/yarn.service
.keytab
hadoop@hadoopmaster:/etc/security/keytabs$ sudo chmod 400 /etc/security/keytabs/yarn.service.keytab
hadoop@hadoopmaster:/etc/security/keytabs$ sudo chown root:root /etc/security/keytabs/spnego.service
.keytab
hadoop@hadoopmaster:/etc/security/keytabs$ sudo chmod 400 /etc/security/keytabs/spnego.service.keyta
b
```

```
hadoop@hadoopnode1:/etc/security/keytabs$ ls
hdfs.service.keytab  spnego.service.keytab  yarn.service.keytab
hadoop@hadoopnode1:/etc/security/keytabs$ sudo chown hdfs:hadoop /etc/security/keytabs/hdfs.service.
keytab
hadoop@hadoopnode1:/etc/security/keytabs$ sudo chmod 400 /etc/security/keytabs/hdfs.service.keytab
hadoop@hadoopnode1:/etc/security/keytabs$ sudo chown hdfs:hadoop /etc/security/keytabs/yarn.service.
keytab
hadoop@hadoopnode1:/etc/security/keytabs$ sudo chmod 400 /etc/security/keytabs/yarn.service.keytab
hadoop@hadoopnode1:/etc/security/keytabs$ sudo chown yarn:hadoop /etc/security/keytabs/yarn.service.
keytab
hadoop@hadoopnode1:/etc/security/keytabs$ sudo chown root:root /etc/security/keytabs/spnego.service.
keytab.service.keytab
[sudo] password for hadoop:
chown: cannot access '/etc/security/keytabs/spnego.service.keytab.service.keytab': No such file or d
irectory
hadoop@hadoopnode1:/etc/security/keytabs$ sudo chmod 400 /etc/security/keytabs/spnego.service.keytab

hadoop@hadoopnode1:/etc/security/keytabs$
```

```
hadoop@hadoopnode2:/etc/security/keytabs$ sudo chown hdfs:hadoop /etc/security/keytabs/hdfs.service.
keytab
[sudo] password for hadoop:
hadoop@hadoopnode2:/etc/security/keytabs$ sudo chmod 400 hdfs:hadoop /etc/security/keytabs/hdfs.serv
ice.keytab
chmod: cannot access 'hdfs:hadoop': No such file or directory
hadoop@hadoopnode2:/etc/security/keytabs$ sudo chmod 400 /etc/security/keytabs/hdfs.service.keytab
hadoop@hadoopnode2:/etc/security/keytabs$ sudo chmod 400 /etc/security/keytabs/yarn.service.keytab
hadoop@hadoopnode2:/etc/security/keytabs$ sudo chown yarn:hadoop /etc/security/keytabs/yarn.service.
keytab
hadoop@hadoopnode2:/etc/security/keytabs$ sudo chown root:root /etc/security/keytabs/spnego.service.
keytab.service.keytab
chown: cannot access '/etc/security/keytabs/spnego.service.keytab.service.keytab': No such file or d
irectory
hadoop@hadoopnode2:/etc/security/keytabs$ sudo chown root:root /etc/security/keytabs/spnego.service.
keytab
hadoop@hadoopnode2:/etc/security/keytabs$ _
```

Installed jvc on all of them

```
hadoop@hadoopnode1:~$ sudo apt-get install jsvc
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libcommons-daemon-java
Suggested packages:
  java-virtual-machine
The following NEW packages will be installed:
  jsvc libcommons-daemon-java
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 72.4 kB of archives.
After this operation, 458 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://gb.archive.ubuntu.com/ubuntu focal/universe amd64 libcommons-daemon-java all 1.0.15-8 [
49.4 kB]
Get:2 http://gb.archive.ubuntu.com/ubuntu focal/universe amd64 jsvc amd64 1.0.15-8 [23.0 kB]
Fetched 72.4 kB in 0s (526 kB/s)
Selecting previously unselected package libcommons-daemon-java.
(Reading database ... 71577 files and directories currently installed.)
Preparing to unpack .../libcommons-daemon-java_1.0.15-8_all.deb ...
Unpacking libcommons-daemon-java (1.0.15-8) ...
Selecting previously unselected package jsvc.
Preparing to unpack .../jsvc_1.0.15-8_amd64.deb ...
Unpacking jsvc (1.0.15-8) ...
Setting up libcommons-daemon-java (1.0.15-8) ...
Setting up jsvc (1.0.15-8) ...
Processing triggers for man-db (2.9.1-1) ...
hadoop@hadoopnode1:~$ sudo mkdir /usr/l
lib/     lib32/   lib64/   libexec/ libx32/  local/
hadoop@hadoopnode1:~$ sudo mkdir /usr/lib/jvm/java-11-openjdk-amd64/lib/amd64
hadoop@hadoopnode1:~$ sudo ln -s /usr/lib/jvm/java-11-openjdk-amd64/lib/amd64/
hadoop@hadoopnode1:~$
```

```
hadoop@hadoopnode2:~$ sudo apt-get install jsvc
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libcommons-daemon-java
Suggested packages:
  java-virtual-machine
The following NEW packages will be installed:
  jsvc libcommons-daemon-java
0 upgraded, 2 newly installed, 0 to remove and 117 not upgraded.
Need to get 72.4 kB of archives.
After this operation, 458 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://gb.archive.ubuntu.com/ubuntu focal/universe amd64 libcommons-daemon-java all 1.0.15-8 [
49.4 kB]
Get:2 http://gb.archive.ubuntu.com/ubuntu focal/universe amd64 jsvc amd64 1.0.15-8 [23.0 kB]
Fetched 72.4 kB in 0s (195 kB/s)
Selecting previously unselected package libcommons-daemon-java.
(Reading database ... 108379 files and directories currently installed.)
Preparing to unpack .../libcommons-daemon-java_1.0.15-8_all.deb ...
Unpacking libcommons-daemon-java (1.0.15-8) ...
Selecting previously unselected package jsvc.
Preparing to unpack .../jsvc_1.0.15-8_amd64.deb ...
Unpacking jsvc (1.0.15-8) ...
Setting up libcommons-daemon-java (1.0.15-8) ...
Setting up jsvc (1.0.15-8) ...
Processing triggers for man-db (2.9.1-1) ...
hadoop@hadoopnode2:~$ sudo mkdir /
bin/        dev/        lib/        libx32/     mnt/        root/       srv/        tmp/
boot/       etc/        lib32/      lost+found/ opt/        run/        swap.img    usr/
cdrom/      home/       lib64/      media/      proc/       sbin/       sys/        var/
hadoop@hadoopnode2:~$ sudo mkdir /usr/lib/jvm/java-11-openjdk-amd64/lib/amd64
hadoop@hadoopnode2:~$ sudo ln -s /usr/lib/jvm/java-11-openjdk-amd64/lib/amd64
hadoop@hadoopnode2:~$
```

Setting up kerberos on hadoopmaster

Core-site.xml

```
  GNU nano 4.8                                    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.default.name</name>
        <value>hdfs://hadoopmaster:9000</value>
    </property>
    <property>
        <name>hadoop.security.authentication</name>
        <value>kerberos</value>
    </property>
    <property>
        <name>hadoop.security.authorization</name>
        <value>true</value>
    </property>
</configuration>
```

Hdfs-site

```
  GNU nano 4.8                                    hdfs-site.xml
    </property>
 <!-- Kerberos Setup -->
    <property>
            <name>dfs.block.access.token.enable</name>
            <value>true</value>
    </property>
    <property>
            <name>dfs.namenode.keytab.file</name>
            <value>/etc/security/keytabs/hdfs.service.keytab</value>
    </property>

    <property>
            <name>dfs.namenode.kerberos.principal</name>
            <value>hdfs/hadoopmaster.nikosledakis.com@NIKOSLEDAKIS.COM</value>
    </property>
    <property>
            <name>dfs.namenode.kerberos.http.principal</name>
            <value>HTTP/hadoopmaster.nikosledakis.com@NIKOSLEDAKIS.COM</value>
    </property>
    <property>
            <name>dfs.namenode.kerberos.internal.spnego.principal</name>
            <value>HTTP/hadoopmaster.nikosledakis.com@NIKOSLEDAKIS.COM</value>
    </property>
    <property>
            <name>dfs.datanode.data.dir.perm</name>
            <value>700</value>
    </property>
    <property>
            <name>dfs.datanode.address</name>
            <value>0.0.0.0:1004</value>
    </property>
    <property>
            <name>dfs.datanode.http.address</name>
```

```
  GNU nano 4.8                                    hdfs-site.xml
            <value>true</value>
    </property>
    <property>
            <name>dfs.namenode.keytab.file</name>
            <value>/etc/security/keytabs/hdfs.service.keytab</value>
    </property>

    <property>
            <name>dfs.namenode.kerberos.principal</name>
            <value>hdfs/hadoopmaster.nikosledakis.com@NIKOSLEDAKIS.COM</value>
    </property>
    <property>
            <name>dfs.namenode.kerberos.http.principal</name>
            <value>HTTP/hadoopmaster.nikosledakis.com@NIKOSLEDAKIS.COM</value>
    </property>
    <property>
            <name>dfs.namenode.kerberos.internal.spnego.principal</name>
            <value>HTTP/hadoopmaster.nikosledakis.com@NIKOSLEDAKIS.COM</value>
    </property>
    <property>
            <name>dfs.datanode.data.dir.perm</name>
            <value>700</value>
    </property>
    <property>
            <name>dfs.datanode.address</name>
            <value>0.0.0.0:1004</value>
    </property>
    <property>
            <name>dfs.datanode.http.address</name>
            <value>0.0.0.0:1006</value>
    </property>

</configuration>
```

Yarn-site

```
  GNU nano 4.8                                    yarn-site.xml

    <property>
            <name>yarn.nodemanager.vmem-check-enabled</name>
            <value>false</value>
    </property>
<!--Kerberos Setup-->
    <property>
            <name>yarn.resourcemanager.principal</name>
            <value>yarn/hadoopmaster.nikosledakis.com@NIKOSLEDAKIS.COM</value>
    </property>
    <property>
            <name>yarn.resourcemanager.keytab</name>
            <value>/etc/security/keytabs/yarn.service.keytab</value>
    </property>

    <property>
            <name>yarn.nodemanager.principal</name>
            <value>yarn/hadoopmaster.nikosledakis.com@NIKOSLEDAKIS.COM</value>
    </property>
    <property>
            <name>yarn.nodemanager.keytab</name>
            <value>/etc/security/keytabs/yarn.service.keytab</value>
    </property>
    <property>
            <name>yarn.nodemanager.container-executor.class</name>
            <value>org.apache.hadoop.yarn.server.nodemanager.LinuxContainerExecutor</value>
    </property>
    <property>
            <name>yarn.nodemanager.linux-container-executor.group</name>
            <value>yarn</value>
    </property>
</configuration>
```

Hadoop-env.sh

```
  GNU nano 4.8                                    hadoop-env.sh
# For example, it is common to use the same garbage collection settings
# for all the daemons.  So one could define:
#
# export HADOOP_GC_SETTINGS="-verbose:gc -XX:+PrintGCDetails -XX:+PrintGCTimeS
#
# .. and then use it as per the b option under the namenode.

###
# Secure/privileged execution
###

#
# Out of the box, Hadoop uses jsvc from Apache Commons to launch daemons
# on privileged ports.  This functionality can be replaced by providing
# custom functions.  See hadoop-functions.sh for more information.
#

# The jsvc implementation to use. Jsvc is required to run secure datanodes
# that bind to privileged ports to provide authentication of data transfer
# protocol.  Jsvc is not required if SASL is configured for authentication of
# data transfer protocol using non-privileged ports.
export JSVC_HOME=/usr/bin
```

```
  GNU nano 4.8                                    hadoop-env.sh
# defined if SASL is configured for authentication of data transfer protocol
# using non-privileged ports.
# This will replace the hadoop.id.str Java property in secure mode.
export HDFS_DATANODE_SECURE_USER=hdfs
export HDFS_SECURE_DATANODE_USER=hdfs_
```

Repeated the process for hadoopnode1 and2

```
  GNU nano 4.8                                    core-site.xml
<?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.default.name</name>
        <value>hdfs://hadoopmaster:9000</value>
    </property>
<!--Kerberos Setup-->
    <property>
        <name>hadoop.security.authentication</name>
        <value>kerberos</value>
    </property>
    <property>
        <name>hadoop.security.authorization</name>
        <value>true</value>
    </property>_
</configuration>
```

```
GNU nano 4.8                                    hdfs-site.xml
            <value>700</value>
    </property>

    <property>
            <name>dfs.datanode.address</name>
            <value>0.0.0.0:1004</value>
    </property>

    <property>
            <name>dfs.datanode.http.address</name>
            <value>0.0.0.0:1006</value>
    </property>

    <property>
            <name>dfs.datanode.kerberos.principal</name>
            <value>hdfs/_HOST@NIKOSLEDAKIS.COM</value>
    </property>

    <property>
            <name>dfs.datanode.kerberos.http.principal</name>
            <value>HTTP/_HOST@REALM</value>
    </property>


    <property>
            <name>dfs.datanode.keytab.file</name>
            <value>/etc/security/keytabs/hdfs.service.keytabs</value>
    </property>



</configuration>
```

```
GNU nano 4.8                                    hdfs-site.xml
<!--Kerberos Setup-->
    <property>
            <name>dfs.block.access.token.enable</name>
            <value>true /value>
    </property>

    <property>
            <name>dfs.datanode.data.dir.perm</name>
            <value>700</value>
    </property>

    <property>
            <name>dfs.datanode.address</name>
            <value>0.0.0.0:1004</value>
    </property>

    <property>
            <name>dfs.datanode.http.address</name>
            <value>0.0.0.0:1006</value>
    </property>

    <property>
            <name>dfs.datanode.kerberos.principal</name>
            <value>hdfs/_HOST@NIKOSLEDAKIS.COM</value>
    </property>

    <property>
            <name>dfs.datanode.kerberos.http.principal</name>
            <value>HTTP/_HOST@REALM</value>
    </property>


    <property>_
```

```
  GNU nano 4.8                                    yarn-site.xml
    </property>

    <property>
            <name>yarn.nodemanager.resource.memory-mb</name>
            <value>1536</value>
    </property>

    <property>
            <name>yarn.scheduler.maximum-allocation-mb</name>
            <value>1536</value>
    </property>

    <property>
            <name>yarn.scheduler.minimum-allocation-mb</name>
            <value>128</value>
    </property>

    <property>
            <name>yarn.nodemanager.vmem-check-enabled</name>
            <value>false</value>
    </property>
<!--Kerberos Setup-->

    <property>
            <name>yarn.resourcemanager.principal</name>
            <value>yarn/_HOST@NIKOSLEDAKIS.COM</value>
    </property>
    <property>
            <name>yarn.resourcemanager.keytab</name>
            <value>/etc/security/keytab/yarn.service.keytab</value>
    </property>
</configuration>
```

```
  GNU nano 4.8                                    hadoop-env.sh

# Default name for the service level authorization file
# Java property: hadoop.policy.file
# export HADOOP_POLICYFILE="hadoop-policy.xml"

#
# NOTE: this is not used by default!  <-----
# You can define variables right here and then re-use them later on.
# For example, it is common to use the same garbage collection settings
# for all the daemons.  So one could define:
#
# export HADOOP_GC_SETTINGS="-verbose:gc -XX:+PrintGCDetails -XX:+PrintGCTimeSt
#
# .. and then use it as per the b option under the namenode.

###
# Secure/privileged execution
###

#
# Out of the box, Hadoop uses jsvc from Apache Commons to launch daemons
# on privileged ports.  This functionality can be replaced by providing
# custom functions.  See hadoop-functions.sh for more information.
#

# The jsvc implementation to use. Jsvc is required to run secure datanodes
# that bind to privileged ports to provide authentication of data transfer
# protocol.  Jsvc is not required if SASL is configured for authentication of
# data transfer protocol using non-privileged ports.
export JSVC_HOME=/usr/bin
```

```
  GNU nano 4.8                                    hadoop-env.sh
# This **MUST** be uncommented to enable secure HDFS if using privileged ports
# to provide authentication of data transfer protocol.  This **MUST NOT** be
# defined if SASL is configured for authentication of data transfer protocol
# using non-privileged ports.
# This will replace the hadoop.id.str Java property in secure mode.
export HDFS_DATANODE_SECURE_USER=hdfs
export HADOOP_SECURE_DATANODE_USER=hdfs_
```

Started the hdfs cluster was not working changed the permission of the hdfs file to 777 and it works

However no live noded were appearing and no error logs showed up in the workers

Realized some things were wrong on the hdfs so

Node 1

```
<!--Kerberos Setup-->

    <property>
            <name>yarn.resourcemanager.principal</name>
            <value>yarn/hadoopnode1.nikosledakis.com@NIKOSLEDAKIS.COM</value>
    </property>
    <property>
            <name>yarn.resourcemanager.keytab</name>
            <value>/etc/security/keytabs/yarn.service.keytab</value>
    </property>
</configuration>
```

hdfs-site removed namenode configuration incase that was messing up the starting process of the datanode

```
  GNU nano 4.8                                    hdfs-site.xml
<!--Kerberos Setup-->
    <property>
            <name>dfs.block.access.token.enable</name>
            <value>true</value>
    </property>
<!--NameNode removed-->
    <property>
            <name>dfs.datanode.data.dir.perm</name>
            <value>700</value>
    </property>

    <property>
            <name>dfs.datanode.address</name>
            <value>0.0.0.0:1004</value>
    </property>

    <property>
            <name>dfs.datanode.http.address</name>
            <value>0.0.0.0:1006</value>
    </property>

    <property>
            <name>dfs.datanode.kerberos.principal</name>
            <value>hdfs/hadoopnode1.nikosledakis.com@NIKOSLEDAKIS.COM</value>
    </property>

    <property>
            <name>dfs.datanode.kerberos.http.principal</name>
            <value>HTTP/hadoopnode1.nikosledakis.com@NIKOSLEDAKIS.COM</value>
    </property>
    <property>
            <name>dfs.datanode.keytab.file</name>
            <value>/etc/security/keytabs/hdfs.service.keytab</value>
```

```
  GNU nano 4.8                                  hdfs-site.xml
            <value>true</value>
    </property>
<!--NameNode removed-->
    <property>
            <name>dfs.datanode.data.dir.perm</name>
            <value>700</value>
    </property>

    <property>
            <name>dfs.datanode.address</name>
            <value>0.0.0.0:1004</value>
    </property>

    <property>
            <name>dfs.datanode.http.address</name>
            <value>0.0.0.0:1006</value>
    </property>

    <property>
            <name>dfs.datanode.kerberos.principal</name>
            <value>hdfs/hadoopnode1.nikosledakis.com@NIKOSLEDAKIS.COM</value>
    </property>

    <property>
            <name>dfs.datanode.kerberos.http.principal</name>
            <value>HTTP/hadoopnode1.nikosledakis.com@NIKOSLEDAKIS.COM</value>
    </property>
    <property>
            <name>dfs.datanode.keytab.file</name>
            <value>/etc/security/keytabs/hdfs.service.keytab</value>
    </property>
</configuration>
```

```
<!--Kerberos Setup-->

    <property>
            <name>yarn.resourcemanager.principal</name>
            <value>yarn/hadoopnode1.nikosledakis.com@NIKOSLEDAKIS.COM</value>
    </property>
    <property>
            <name>yarn.resourcemanager.keytab</name>
            <value>/etc/security/keytabs/yarn.service.keytab</value>
    </property>
</configuration>
```

Node 2

```
  GNU nano 4.8                                  hdfs-site.xml
            <value>1</value>
    </property>
<!--Kerberos Setup-->
    <property>
            <name>dfs.block.access.token.enable</name>
            <value>true</value>
    </property>

    <property>
            <name>dfs.namenode.keytab.file</name>
            <value>/etc/security/keytabs/hdfs.service.keytab</value>
    </property>
    <property>
            <name>dfs.namenode.kerberos.principal</name>
            <value>hdfs/hadoopmaster.nikosledakis.com@NIKOSLEDAKIS.COM</value>
    </property>
    <property>
            <name>dfs.namenode.kerberos.http.principal</name>
            <value>HTTP/hadoopmaster.nikosledakis.com@NIKOSLEDAKIS.COM</value>
    </property>
    <property>
            <name>dfs.namenode.kerberos.internal.spnego.principal</name>
            <value>HTTP/hadoopmaster.nikosledakis.com@NIKOSLEDAKIS.COM</value>
    </property>
    <property>
            <name>dfs.datanode.data.dir.perm</name>
            <value>700</value>
    </property>
    <property>
            <name>dfs.datanode.address</name>
            <value>0.0.0.0:1004</value>
    </property>
```

```
GNU nano 4.8                                hdfs-site.xml
    <property>
            <name>dfs.namenode.kerberos.http.principal</name>
            <value>HTTP/hadoopmaster.nikosledakis.com@NIKOSLEDAKIS.COM</value>
    </property>
    <property>
            <name>dfs.namenode.kerberos.internal.spnego.principal</name>
            <value>HTTP/hadoopmaster.nikosledakis.com@NIKOSLEDAKIS.COM</value>
    </property>
    <property>
            <name>dfs.datanode.data.dir.perm</name>
            <value>700</value>
    </property>
    <property>
            <name>dfs.datanode.address</name>
            <value>0.0.0.0:1004</value>
    </property>
    <property>
            <name>dfs.datanode.http.address</name>
            <value>0.0.0.0:1006</value>
    </property>
    <property>
            <name>dfs.datanode.kerberos.principal</name>
            <value>hdfs/hadoopnode2.nikosledakis.com@NIKOSLEDAKIS.COM</value>
    </property>
    <property>
            <name>dfs.datanode.kerberos.http.principal</name>
            <value>HTTP/hadoopnode2.nikosledakis.com@NIKOSLEDAKIS.COM</value>
    </property>
    <property>
            <name>dfs.datanode.keytab.file</name>
            <value>/etc/security/keytabs/hdfs.service.keytab</value>
    </property>
</configuration>
```

```
  GNU nano 4.8                                yarn-site.xml
            <name>yarn.nodemanager.resource.memory-mb</name>
            <value>1536</value>
    </property>

    <property>
            <name>yarn.scheduler.maximum-allocation-mb</name>
            <value>1536</value>
    </property>

    <property>
            <name>yarn.scheduler.minimum-allocation-mb</name>
            <value>128</value>
    </property>

    <property>
            <name>yarn.nodemanager.vmem-check-enabled</name>
            <value>false</value>
    </property>
<!--Kerberos Setup -->
    <property>
            <name>yarn.resourcemanager.principal</name>
            <value>yarn/hadoopnode2.nikosledakis.com@NIKOSLEDAKIS.COM</value>
    </property>
    <property>
            <name>yarn.resourcemanager.keytab</name>
            <value>/etc/security/keytabs/yarn.service.keytab</value>
    </property>

</configuration>
```

Still no logs appeared and cannot start datanodes comes up with permission errors

Changed log path to kerb5.conf to add logging to a certain path and run restart the cluster for new logs however none were created

```
  GNU nano 4.8                                    krb5.conf
[libdefaults]
        default_realm = NIKOSLEDAKIS.COM

# The following krb5.conf variables are only for MIT Kerberos.
        kdc_timesync = 1
        ccache_type = 4
        forwardable = true
        proxiable = true

# The following encryption type specification will be used by MIT Kerberos
# if uncommented.  In general, the defaults in the MIT Kerberos code are
# correct and overriding these specifications only serves to disable new
# encryption types as they are added, creating interoperability problems.
#
# The only time when you might need to uncomment these lines and change
# the enctypes is if you have local software that will break on ticket
# caches containing ticket encryption types it doesn't know about (such as
# old versions of Sun Java).

#       default_tgs_enctypes = des3-hmac-sha1
#       default_tkt_enctypes = des3-hmac-sha1
#       permitted_enctypes = des3-hmac-sha1

# The following libdefaults parameters are only for Heimdal Kerberos.
        fcc-mit-ticketflags = true
[logging]
default = FILE:/var/log/krb5/krb5.log
[realms]
        NIKOSLEDAKIS.COM = {
                kdc = hadoopmaster.nikosledakis.com
                admin_server = hadoopmaster.nikosledakis.com
        }
        ATHENA.MIT.EDU = {
```

Additionally added these exports to Hadoop-env.sh to create a new logging folder

```
export HADOOP_SECURE_DN_USER=hdfs

export HADOOP_SECURE_DN_PID_DIR=/var/lib/hadoop-hdfs

export HADOOP_SECURE_DN_LOG_DIR=/var/log/hadoop-hdfs

export JSVC_HOME=/usr/lib/bigtop-utils/
```

```
# Specify the JVM options to be used when starting the NFS3 Gateway.
# These options will be appended to the options specified as HADOOP_OPTS
# and therefore may override any similar flags set in HADOOP_OPTS
#
# export HDFS_NFS3_OPTS=""

# Specify the JVM options to be used when starting the Hadoop portmapper.
# These options will be appended to the options specified as HADOOP_OPTS
# and therefore may override any similar flags set in HADOOP_OPTS
#
# export HDFS_PORTMAP_OPTS="-Xmx512m"

# Supplemental options for priviliged gateways


hadoop@hadoopnode1:~/hadoop-3.3.0/etc/hadoop$ sudo ./hdfs --config $HADOOP_HOME/etc/hadoop --daemon
start datanode
sudo: ./hdfs: command not found
hadoop@hadoopnode1:~/hadoop-3.3.0/etc/hadoop$ cd ../..
hadoop@hadoopnode1:~/hadoop-3.3.0$ cd bin
hadoop@hadoopnode1:~/hadoop-3.3.0/bin$ sudo ./hdfs --config $HADOOP_HOME/etc/hadoop --daemon start d
atanode
WARNING: HADOOP_SECURE_DN_USER has been replaced by HDFS_DATANODE_SECURE_USER. Using value of HADOOP
_SECURE_DN_USER.
WARNING: HADOOP_SECURE_DN_PID_DIR has been replaced by HADOOP_SECURE_PID_DIR. Using value of HADOOP_
SECURE_DN_PID_DIR.
WARNING: HADOOP_SECURE_DN_LOG_DIR has been replaced by HADOOP_SECURE_LOG_DIR. Using value of HADOOP_
SECURE_DN_LOG_DIR.
WARNING: /var/lib/hadoop-hdfs does not exist. Creating.
WARNING: /var/log/hadoop-hdfs does not exist. Creating.
ERROR: Cannot set priority of datanode process 2322
hadoop@hadoopnode1:~/hadoop-3.3.0/bin$ cd /var
hadoop@hadoopnode1:/var$ cd log/hadoop-hdfs/
hadoop@hadoopnode1:/var/log/hadoop-hdfs$ ls
hadoop-hdfs-root-datanode-hadoopnode1.out  privileged-root-datanode-hadoopnode1.out
privileged-root-datanode-hadoopnode1.err
hadoop@hadoopnode1:/var/log/hadoop-hdfs$ _
```

one empty .out file was created along with another empty .err file while priviledged-root-datanode-hadoopnode1.out file was created containing this error

```
  GNU nano 4.8                    privileged-root-datanode-hadoopnode1.out
Cannot find any VM in Java Home /usr/lib/jvm/java-11-openjdk-amd64
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
scheduling priority             (-e) 0
file size               (blocks, -f) unlimited
pending signals                 (-i) 7398
max locked memory       (kbytes, -l) 65536
max memory size         (kbytes, -m) unlimited
open files                      (-n) 1024
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
real-time priority              (-r) 0
stack size              (kbytes, -s) 8192
cpu time               (seconds, -t) unlimited
max user processes              (-u) 7398
virtual memory          (kbytes, -v) unlimited
file locks                      (-x) unlimited

                [ File 'privileged-root-datanode-hadoopnode1.out' is unwritable ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line M-E Redo
```

Starting the cluster

```
hadoop@hadoopmaster:~/hadoop-3.3.0/bin$ su hdfs
Password:
hdfs@hadoopmaster:/home/hadoop/hadoop-3.3.0/bin$ ./hdfs --config $HADOOP_HOME/etc/hadoop --daemon st
art namenode
hdfs@hadoopmaster:/home/hadoop/hadoop-3.3.0/bin$ exit
exit
hadoop@hadoopmaster:~/hadoop-3.3.0/bin$ su yarn
Password:
yarn@hadoopmaster:/home/hadoop/hadoop-3.3.0/bin$ ./yarn --config $HADOOP_HOME/etc/hadoop --daemon st
art resourcemanager
yarn@hadoopmaster:/home/hadoop/hadoop-3.3.0/bin$ exit
exit
hadoop@hadoopmaster:~/hadoop-3.3.0/bin$ _
```

```
hadoop@hadoopnode1:~/hadoop-3.3.0/etc/hadoop$ cd $HADOOP_HOME/bin
hadoop@hadoopnode1:~/hadoop-3.3.0/bin$ sudo ./hdfs --config $HADOOP_HOME/etc/hadoop --daemon start d
atanode
ERROR: Cannot set priority of datanode process 4638
hadoop@hadoopnode1:~/hadoop-3.3.0/bin$ sudo ./yarn --config $HADOOP_HOME/etc/hadoop --daemon start n
odemanager
hadoop@hadoopnode1:~/hadoop-3.3.0/bin$ _
```

```
hadoop@hadoopnode2:~/hadoop-3.3.0$ cd bin
hadoop@hadoopnode2:~/hadoop-3.3.0/bin$ sudo ./hdfs --config $HADOOP_HOME/etc/hadoop --daemon start d
atanode
ERROR: Cannot set priority of datanode process 4854
hadoop@hadoopnode2:~/hadoop-3.3.0/bin$ sudo ./yarn --config $HADOOP_HOME/etc/hadoop --daemon start n
odemanager
hadoop@hadoopnode2:~/hadoop-3.3.0/bin$
```

Hadoop | Overview | Datanodes | Datanode Volume Failures | Snapshot | Startup Progress | Utilities

Overview 'hadoopmaster:9000' (✔active)

Started:	Mon Dec 04 16:32:31 +0000 2023
Version:	3.3.0, raa96f1871bfd858f9bac59cf2a81ec470da649af
Compiled:	Mon Jul 06 19:44:00 +0100 2020 by brahma from branch-3.3.0
Cluster ID:	CID-7ce0be9a-8afa-4217-b327-88d75acd6ed6
Block Pool ID:	BP-224195785-192.168.233.3-1700482250304

Summary

Security is on.

Safe mode is ON. The reported blocks 0 needs additional 5 blocks to reach the threshold 0.9990 of total blocks 6. The minimum number of live datanodes is not required. Safe mode will be turned off automatically once the thresholds have been reached.

17 files and directories, 6 blocks (6 replicated blocks, 0 erasure coded block groups) = 23 total filesystem object(s).

Heap Memory used 43.47 MB of 64 MB Heap Memory. Max Heap Memory is 486 MB.

Non Heap Memory used 47.77 MB of 51.13 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

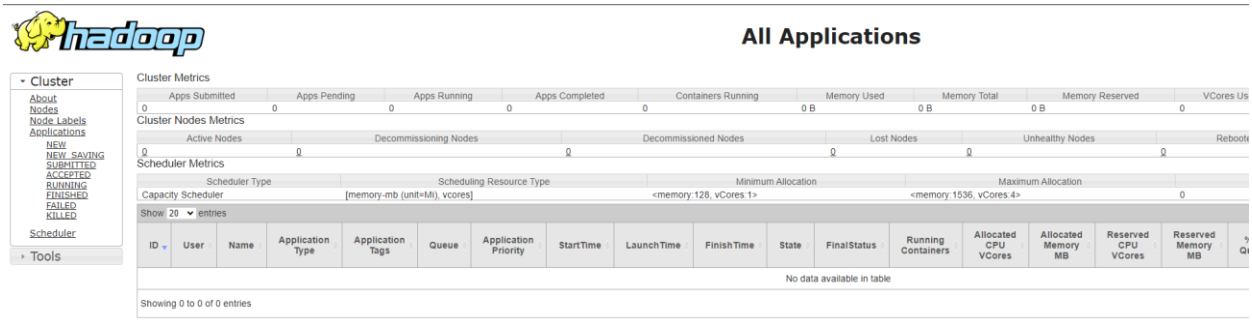
Configured Capacity:	0 B
Configured Remote Capacity:	0 B
DFS Used:	0 B (100%)
Non DFS Used:	0 B
DFS Remaining:	0 B (0%)
Block Pool Used:	0 B (100%)

DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	0 (Decommissioned: 0, In Maintenance: 0)
Dead Nodes	0 (Decommissioned: 0, In Maintenance: 0)
Decommissioning Nodes	0
Entering Maintenance Nodes	0
Total Datanode Volume Failures	0 (0 B)
Number of Under-Replicated Blocks	0
Number of Blocks Pending Deletion (including replicas)	0
Block Deletion Start Time	Mon Dec 04 16:32:31 +0000 2023
Last Checkpoint Time	Mon Dec 04 16:32:31 +0000 2023
Enabled Erasure Coding Policies	RS-6-3-1024k

NameNode Journal Status

Current transaction ID: 397

Journal Manager	State
FileJournalManager(root=/home/hadoop/hdfs/namenode)	EditLogFileOutputStream(/home/hadoop/hdfs/namenode/current/edits_inprogress_0000000000000000397)

NameNode Storage

Storage Directory	Type	State
/home/hadoop/hdfs/namenode	IMAGE_AND_EDITS	Active