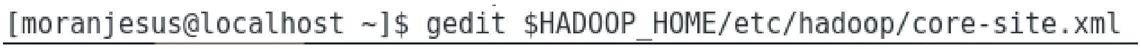
Instalación de Hadoop single node pseudo-distribuida

Partimos de que se tiene instalado Hadoop single node. Para la última parte también se necesita el programa que calcula la temperatura máxima

1. Modificamos la configuración de Hadoop:

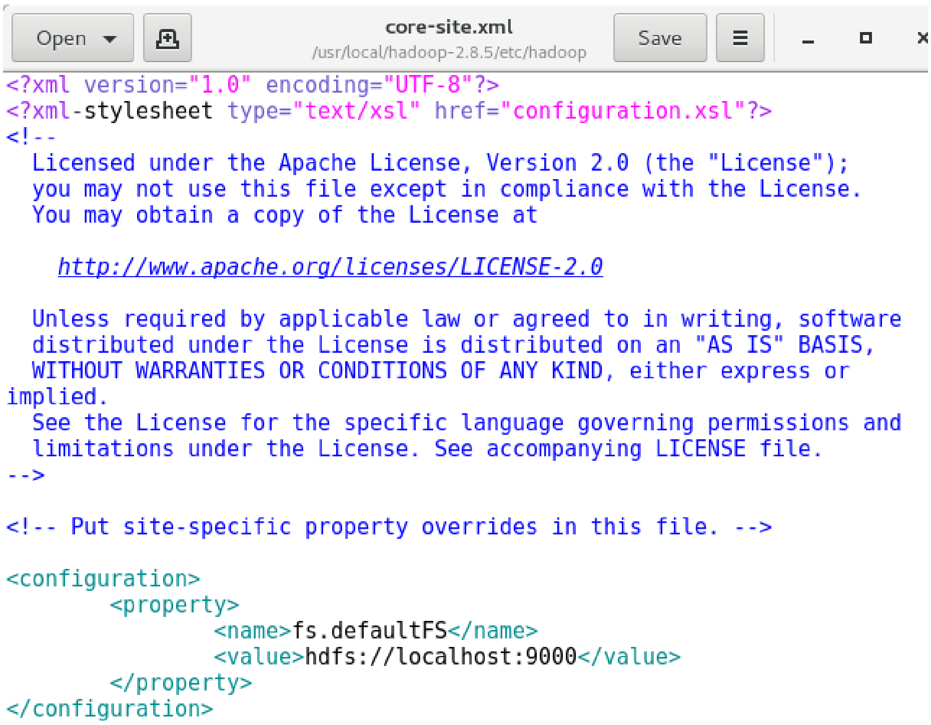
1.1. Ejecutamos: gedit $HADOOP_HOME/etc/hadoop/core-site.xml



```
[moranjesus@localhost ~]$ gedit $HADOOP_HOME/etc/hadoop/core-site.xml
```

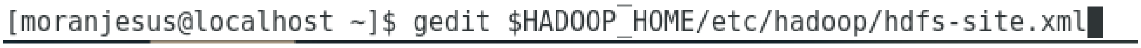
1.2. Añadimos dentro de configuración:

```
<property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
```



```
<?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://localhost:9000</value>
        </property>
</configuration>
```

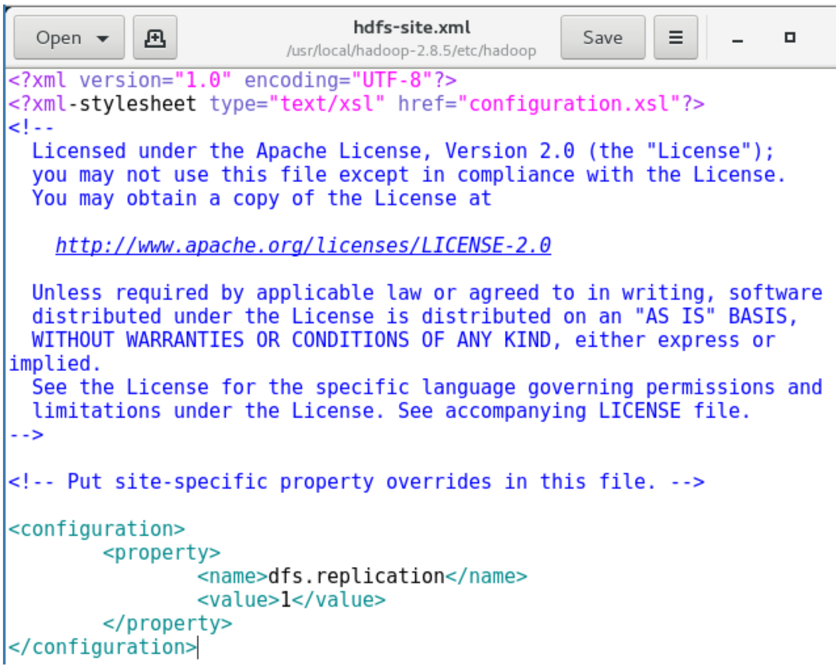
1.3. Ejecutamos: gedit $HADOOP_HOME/etc/hadoop/hdfs-site.xml



```
[moranjesus@localhost ~]$ gedit $HADOOP_HOME/etc/hadoop/hdfs-site.xml
```

1.4. Añadimos dentro de configuración:

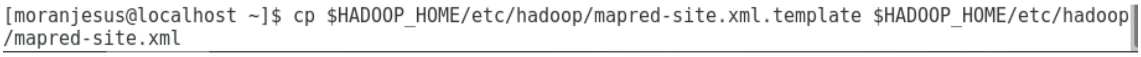
```
<property>
    <name>dfs.replication</name>
    <value>1</value>
</property>
```



```
<?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>
```

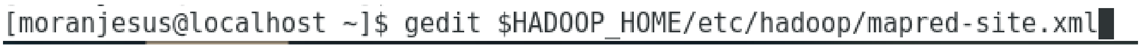
1.5. Ejecutamos: cp $HADOOP_HOME/etc/hadoop/mapred-site.xml.template $HADOOP_HOME/etc/hadoop/mapred-site.xml

Nota: puede que ya esté copiado y no se necesite copiar



```
[moranjesus@localhost ~]$ cp $HADOOP_HOME/etc/hadoop/mapred-site.xml.template $HADOOP_HOME/etc/hadoop
/mapred-site.xml
```

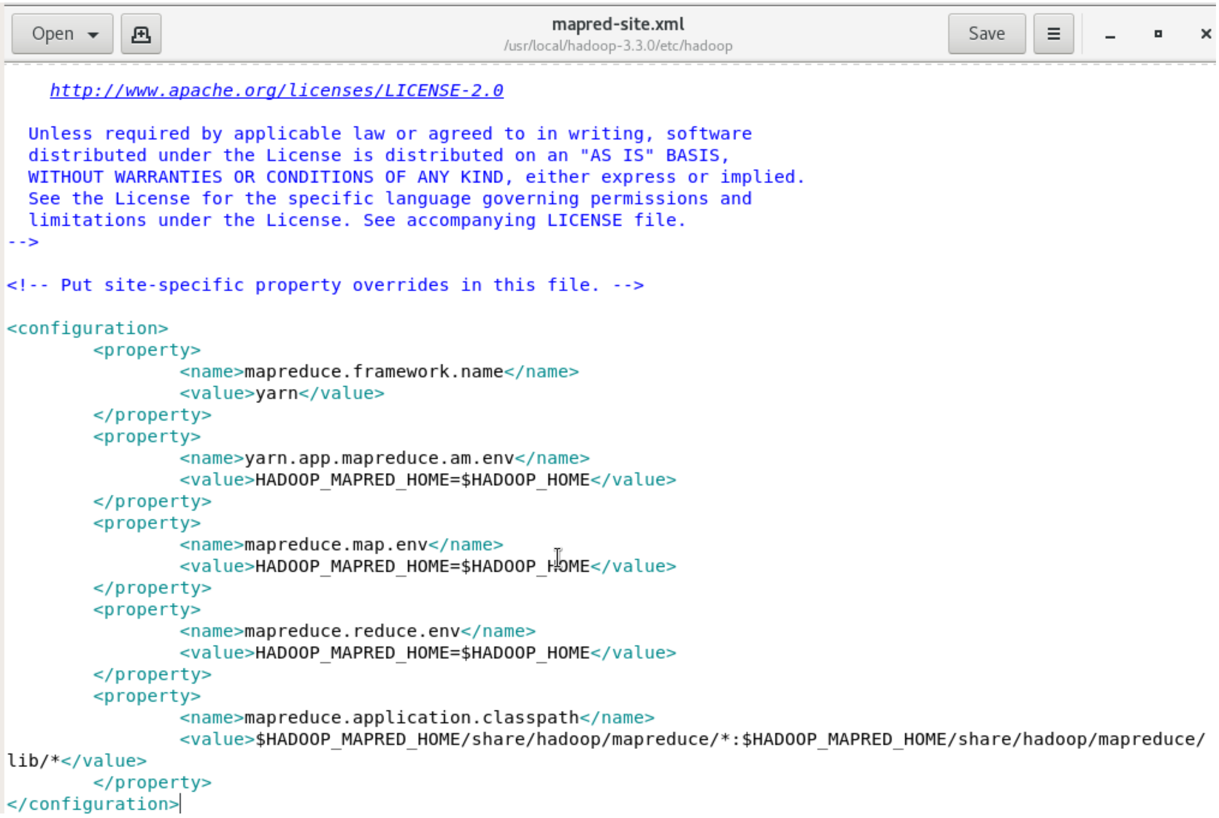
1.6. Ejecutamos: gedit $HADOOP_HOME/etc/hadoop/mapred-site.xml



```
[moranjesus@localhost ~]$ gedit $HADOOP_HOME/etc/hadoop/mapred-site.xml
```

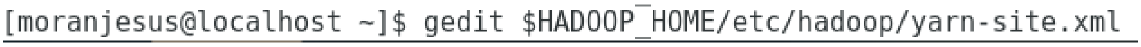
1.7. Añadimos dentro de configuración:

```
<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>mapreduce.application.classpath</name>

<value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/lib/*</value>
     </property>
```



```
    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>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=$HADOOP_HOME</value>
        </property>
        <property>
                <name>mapreduce.application.classpath</name>
                <value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/
lib/*</value>
        </property>
</configuration>
```

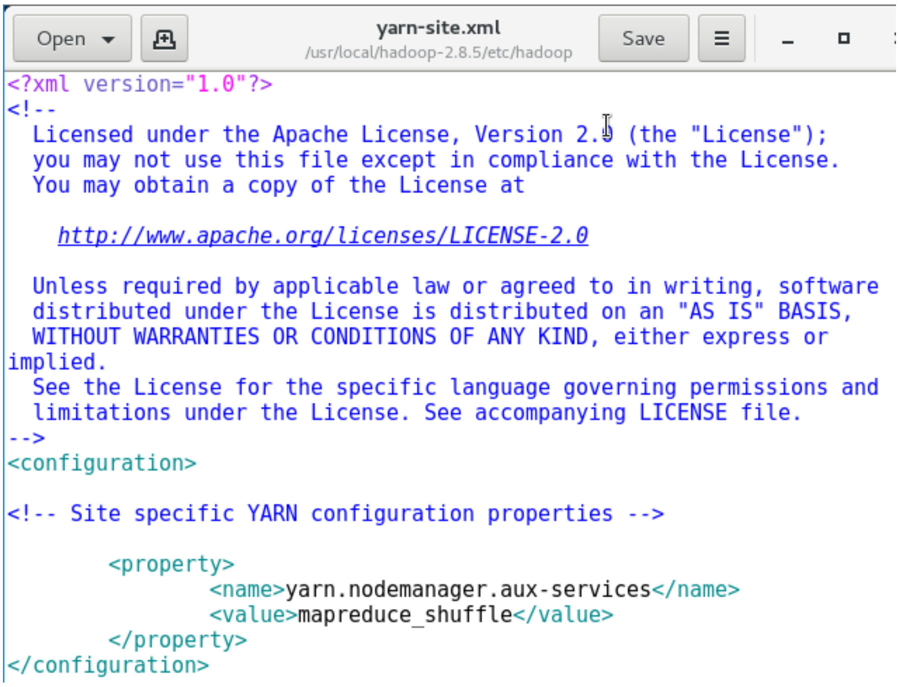
1.8. Ejecutamos: gedit $HADOOP_HOME/etc/hadoop/yarn-site.xml



```
[moranjesus@localhost ~]$ gedit $HADOOP_HOME/etc/hadoop/yarn-site.xml
```

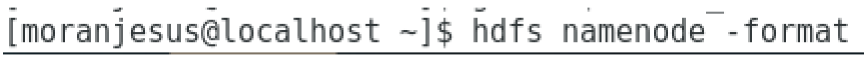
1.9. Añadimos dentro de configuración:

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



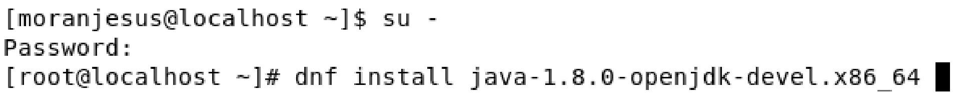
```
<?xml version="1.0"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    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>

<!-- Site specific YARN configuration properties -->

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

2. Formateamos el sistema de archivos: hdfs namenode -format



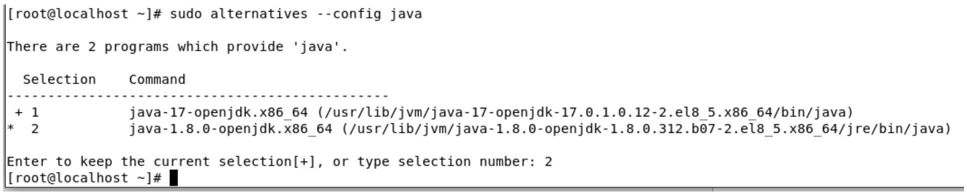
```
[moranjesus@localhost ~]$ hdfs namenode -format
```

3. Hadoop tiene un “problema” con las últimas versiones de java: https://issues.apache.org/jira/browse/HADOOP-10848. Hay varias formas de ejecutar Hadoop, lo recomendado es utilizar java 8. Para ello:
   3.1. Nos conectamos como sudo: su -
   3.2. Instalamos el openjdk 8, para ello: dnf install java-1.8.0-openjdk-devel.x86_64



```
[moranjesus@localhost ~]$ su -
Password:
[root@localhost ~]# dnf install java-1.8.0-openjdk-devel.x86_64
```

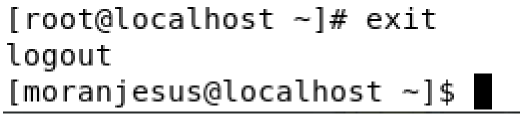
   3.3. Seleccionamos el nuevo java, para ello: sudo alternatives --config java



```
[root@localhost ~]# sudo alternatives --config java

There are 2 programs which provide 'java'.

  Selection    Command
-----------------------------------------------
 + 1           java-17-openjdk.x86_64 (/usr/lib/jvm/java-17-openjdk-17.0.1.0.12-2.el8_5.x86_64/bin/java)
*  2           java-1.8.0-openjdk.x86_64 (/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.312.b07-2.el8_5.x86_64/jre/bin/java)

Enter to keep the current selection[+], or type selection number: 2
[root@localhost ~]#
```

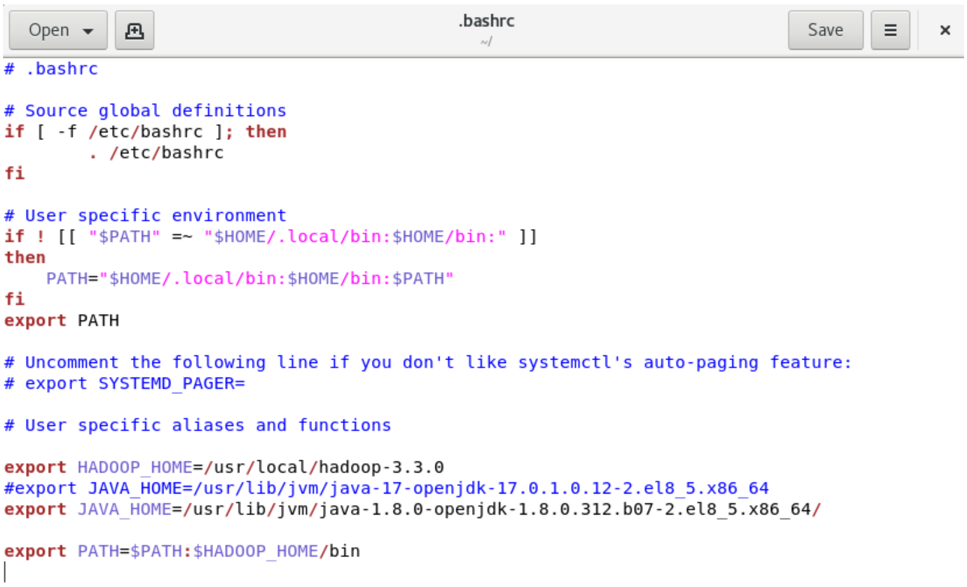
Importante seleccionar el java 8 que acabadmosd e descargar

   3.4. Salimos como usuario root, para ello: exit



```
[root@localhost ~]# exit
logout
[moranjesus@localhost ~]$
```

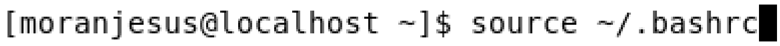
   3.5. Como nuestro usuario editamos la variable de entorno de JAVA_HOME para poner la nueva ubicación de java. Para ello: gedit ~/.bashrc



```
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# User specific environment
if ! [[ "$PATH" =~ "$HOME/.local/bin:$HOME/bin:" ]]
then
    PATH="$HOME/.local/bin:$HOME/bin:$PATH"
fi
export PATH

# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=

# User specific aliases and functions

export HADOOP_HOME=/usr/local/hadoop-3.3.0
#export JAVA_HOME=/usr/lib/jvm/java-17-openjdk-17.0.1.0.12-2.el8_5.x86_64
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.312.b07-2.el8_5.x86_64/

export PATH=$PATH:$HADOOP_HOME/bin
```

Notar que hemos comentado la anterior JAVA_HOME y puesto la nueva. Puede que los números del final sean diferentes para cada persona. Para comprobar cuáles son los números podéis hacer ls /usr/lib/jvm y copiar los números adecuados

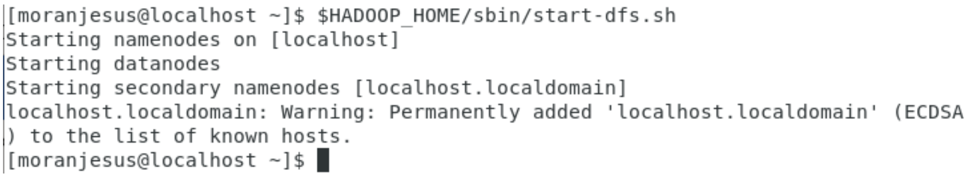
   3.6. Ahora hacemos un source para tener ya disponible la nueva variable de entorno, para ello: source ~/.bashrc



```
[moranjesus@localhost ~]$ source ~/.bashrc
```

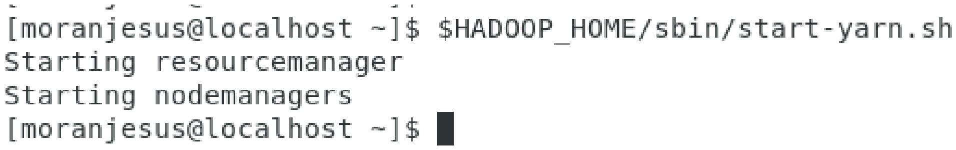
4. Inicializamos los servicios:

4.1. Inicializamos el sistema de archivos distribuido: $HADOOP_HOME/sbin/start-dfs.sh



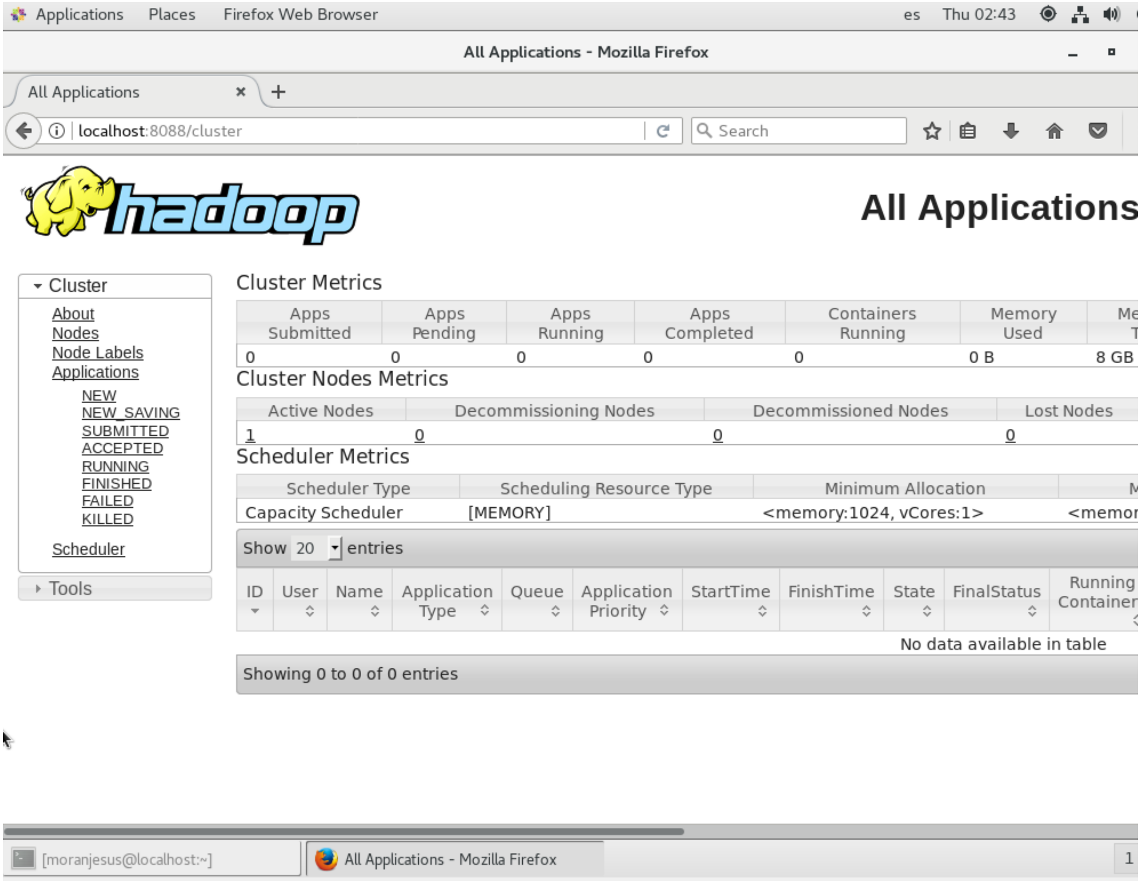
```
[moranjesus@localhost ~]$ $HADOOP_HOME/sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [localhost.localdomain]
localhost.localdomain: Warning: Permanently added 'localhost.localdomain' (ECDSA
) to the list of known hosts.
[moranjesus@localhost ~]$ 
```

4.2. Inicializamos yarn: $HADOOP_HOME/sbin/start-yarn.sh



```
[moranjesus@localhost ~]$ $HADOOP_HOME/sbin/start-yarn.sh
Starting resourcemanager
Starting nodemanagers
[moranjesus@localhost ~]$ 
```

4.3. Entramos en la página http://localhost:8088 (del equipo almaLinux)



All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Me T
0	0	0	0	0	0 B	8 GB

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes
1	0	0	0

Scheduler Metrics

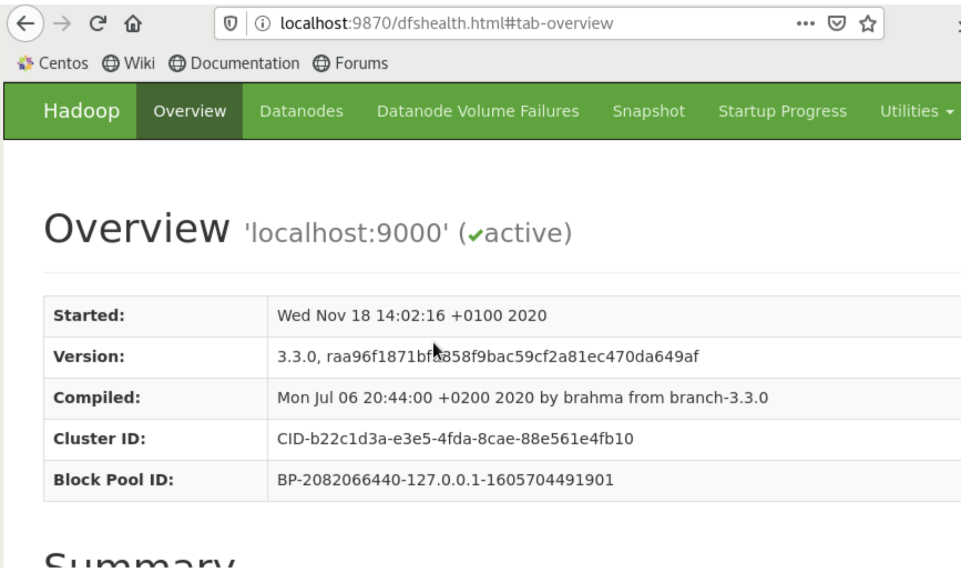
Scheduler Type	Scheduling Resource Type	Minimum Allocation	M
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memor

Show 20 entries

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Container
										No data available in table

Showing 0 to 0 of 0 entries

4.4. Entramos en la página http://localhost:9870 (del equipo almalinux)

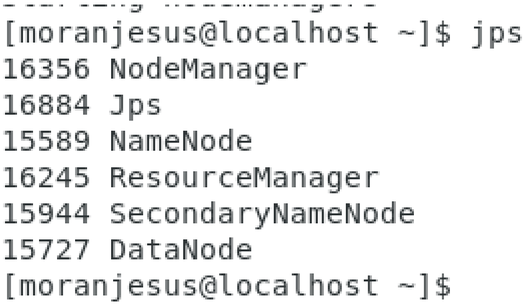


Overview 'localhost:9000' (✓active)

Started:	Wed Nov 18 14:02:16 +0100 2020
Version:	3.3.0, raa96f1871bfd858f9bac59cf2a81ec470da649af
Compiled:	Mon Jul 06 20:44:00 +0200 2020 by brahma from branch-3.3.0
Cluster ID:	CID-b22c1d3a-e3e5-4fda-8cae-88e561e4fb10
Block Pool ID:	BP-2082066440-127.0.0.1-1605704491901

Summary

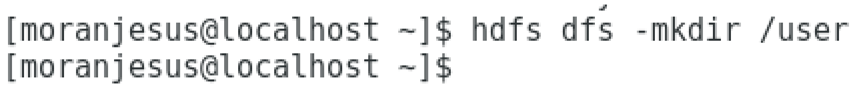
4.5. Comprobamos que están los daemons ejecutándose: jps



```
[moranjesus@localhost ~]$ jps
16356 NodeManager
16884 Jps
15589 NameNode
16245 ResourceManager
15944 SecondaryNameNode
15727 DataNode
[moranjesus@localhost ~]$
```

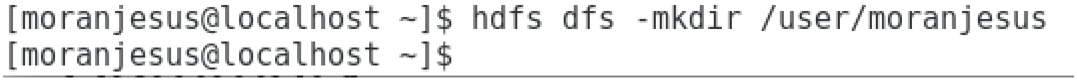
5. Creamos una estructura de carpetas en HDFS:

5.1. Creamos la carpeta user, para ello: hdfs dfs -mkdir /user



```
[moranjesus@localhost ~]$ hdfs dfs -mkdir /user
[moranjesus@localhost ~]$
```

5.2. Creamos la carpeta de nuestro usuario: hdfs dfs -mkdir /user/moranjesus

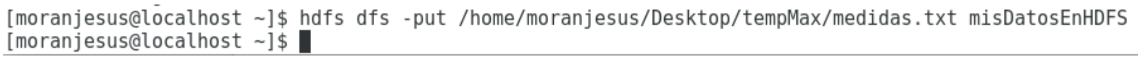


```
[moranjesus@localhost ~]$ hdfs dfs -mkdir /user/moranjesus
[moranjesus@localhost ~]$
```

6. Subimos datos a HDFS:

6.1. Ejecutamos la subida:

hdfs dfs -put /home/moranjesus/Desktop/tempMax/medidas.txt misDatosEnHDFS

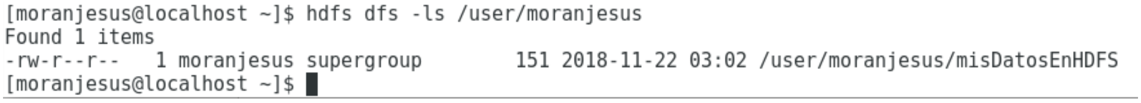


```
[moranjesus@localhost ~]$ hdfs dfs -put /home/moranjesus/Desktop/tempMax/medidas.txt misDatosEnHDFS
[moranjesus@localhost ~]$ 
```

6.2. Comprobamos que los datos se introdujeron en HDFS, para ello:

hdfs dfs -ls /user/moranjesus

nota: equivale a hdfs dfs -ls

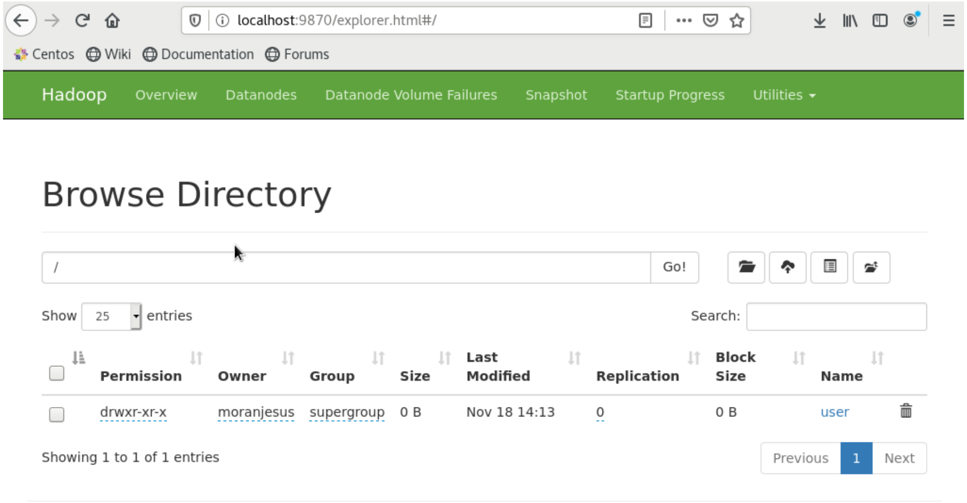


```
[moranjesus@localhost ~]$ hdfs dfs -ls /user/moranjesus
Found 1 items
-rw-r--r--   1 moranjesus supergroup        151 2018-11-22 03:02 /user/moranjesus/misDatosEnHDFS
[moranjesus@localhost ~]$ 
```

6.3. Comprobamos desde el servidor web HDFS que se introdujeron los datos, para ello:

Desde el navegador web: http://localhost:9870 (de la máquina almaLinux)

Luego, vamos a Utilities -> browse the file system:



Browse Directory

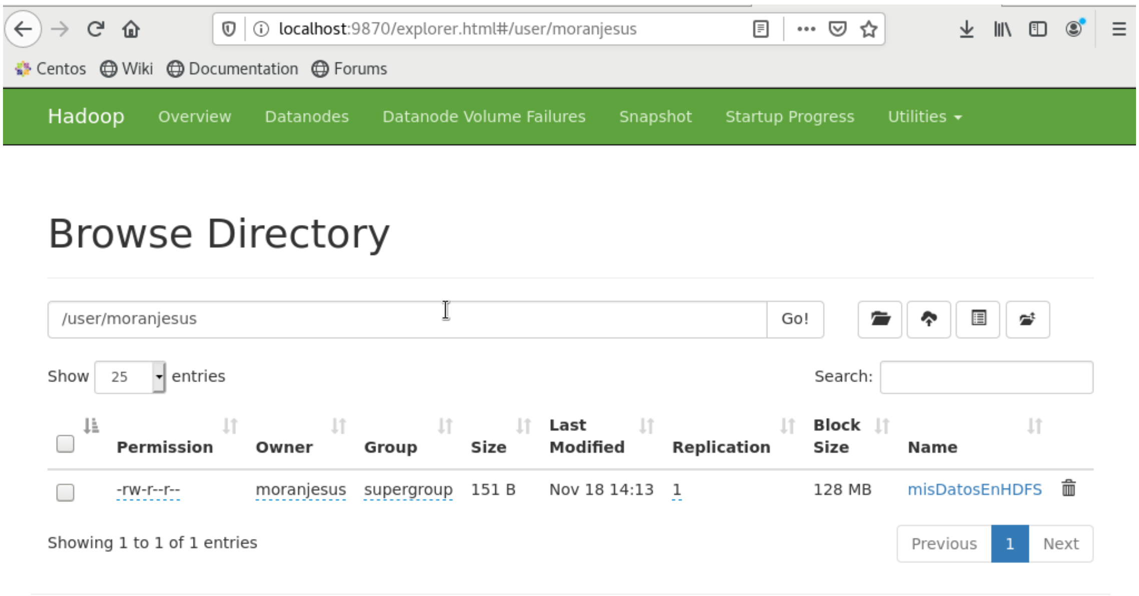
/

Show 25 entries　　Search:

	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
□	drwxr-xr-x	moranjesus	supergroup	0 B	Nov 18 14:13	0	0 B	user	🗑

Showing 1 to 1 of 1 entries

Entramos en /user/moranjesus



Browse Directory

/user/moranjesus

Show 25 entries　　Search:

	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
□	-rw-r--r--	moranjesus	supergroup	151 B	Nov 18 14:13	1	128 MB	misDatosEnHDFS	🗑

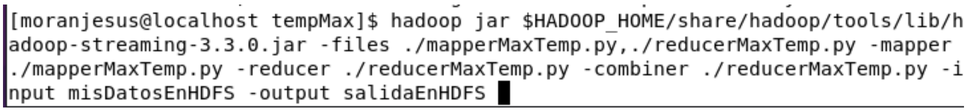
Showing 1 to 1 of 1 entries

7. Ejecutamos un programa sobre los datos de HDFS:
   7.1. Nos ubicamos en la carpeta que tenemos el programa maxTemp, para ello: cd /home/moranjesus/Desktop/tempMax

```
[moranjesus@localhost tempMax]$ 
```

   7.2. Ejecutamos el programa:

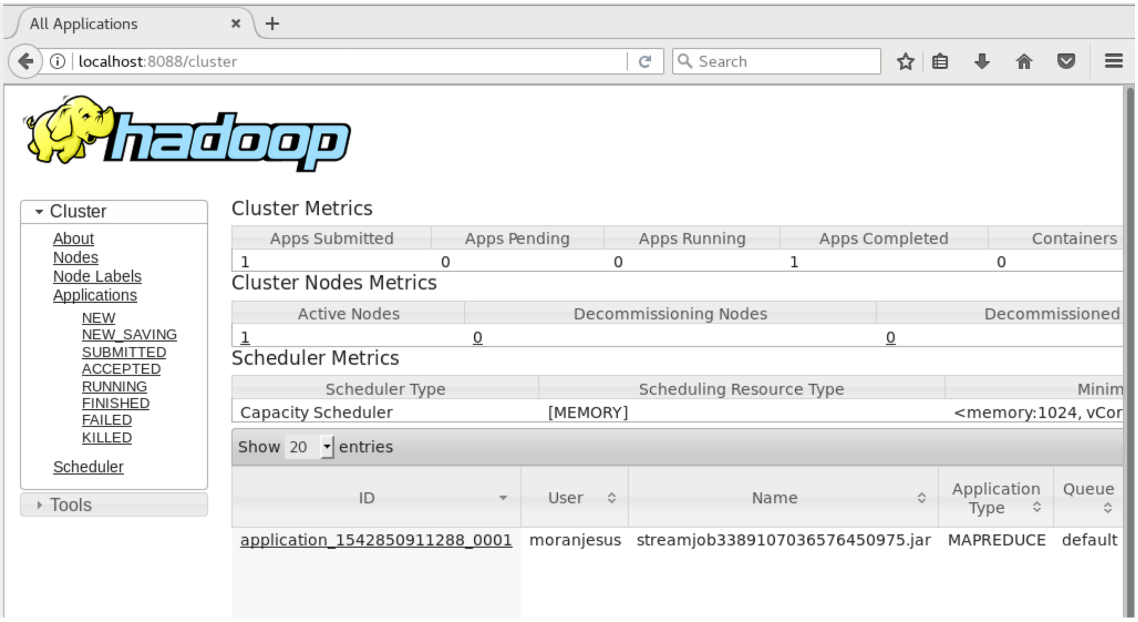
hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-3.3.0.jar -files ./mapperMaxTemp.py,./reducerMaxTemp.py -mapper ./mapperMaxTemp.py -reducer ./reducerMaxTemp.py -combiner ./reducerMaxTemp.py **-input misDatosEnHDFS -output salidaEnHDFS**



```
[moranjesus@localhost tempMax]$ hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/h
adoop-streaming-3.3.0.jar -files ./mapperMaxTemp.py,./reducerMaxTemp.py -mapper
./mapperMaxTemp.py -reducer ./reducerMaxTemp.py -combiner ./reducerMaxTemp.py -i
nput misDatosEnHDFS -output salidaEnHDFS 
```

```
	File System Counters
		FILE: Number of bytes read=39
		FILE: Number of bytes written=485885
		FILE: Number of read operations=0
		FILE: Number of large read operations=0
		FILE: Number of write operations=0
		HDFS: Number of bytes read=435
		HDFS: Number of bytes written=27
		HDFS: Number of read operations=9
		HDFS: Number of large read operations=0
		HDFS: Number of write operations=2
	Job Counters
		Killed map tasks=1
		Launched map tasks=2
		Launched reduce tasks=1
		Data-local map tasks=2
		Total time spent by all maps in occupied slots (ms)=22310
		Total time spent by all reduces in occupied slots (ms)=5428
		Total time spent by all map tasks (ms)=22310
		Total time spent by all reduce tasks (ms)=5428
		Total vcore-milliseconds taken by all map tasks=22310
		Total vcore-milliseconds taken by all reduce tasks=5428
		Total megabyte-milliseconds taken by all map tasks=22845440
		Total megabyte-milliseconds taken by all reduce tasks=5558272
	Map-Reduce Framework
		Map input records=11
		Map output records=11
		Map output bytes=77
		Map output materialized bytes=45
		Input split bytes=208
		Combine input records=11
		Combine output records=3
		Reduce input groups=3
```

7.3. Comprobamos desde la interfaz gráfica que se ejecutó el programa: http://localhost:8088



Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers
1	0	0	1	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned
1	0	0

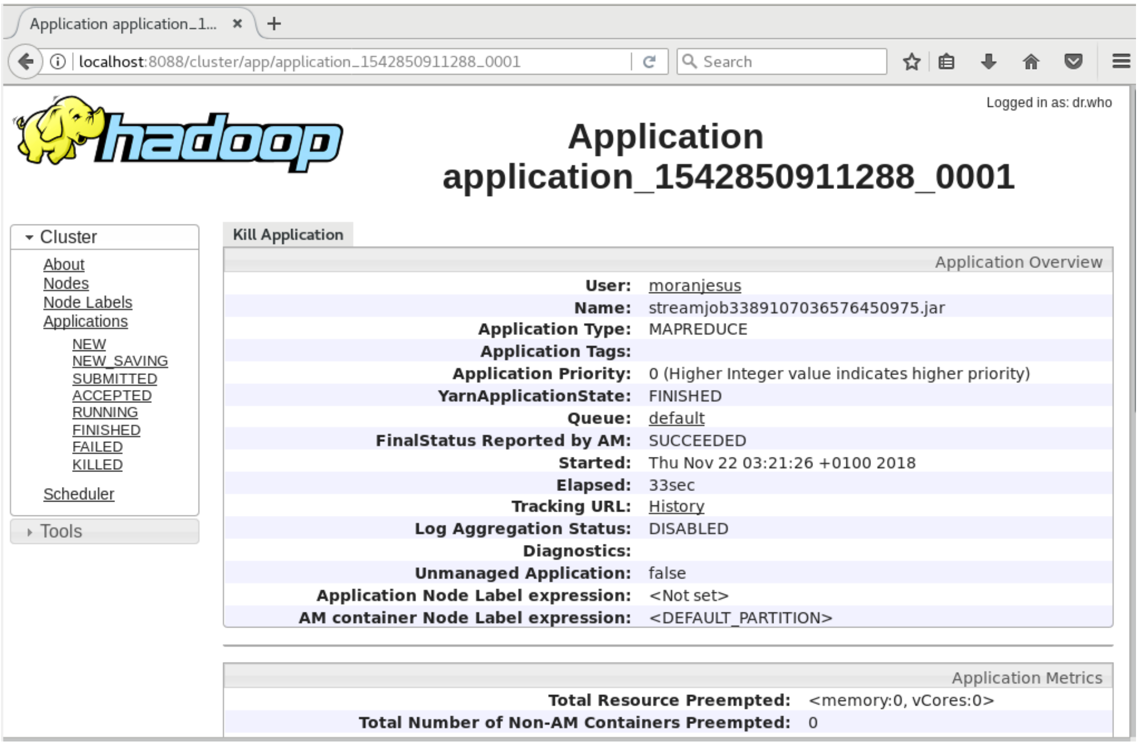
Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minim
Capacity Scheduler	[MEMORY]	<memory:1024, vCor

Show 20 entries

ID	User	Name	Application Type	Queue
application_1542850911288_0001	moranjesus	streamjob3389107036576450975.jar	MAPREDUCE	default

7.4. Si entramos dentro de la application_* podemos ver más información:



Application application_1542850911288_0001

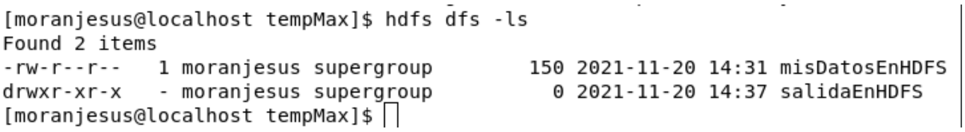
Application Overview

User:	moranjesus
Name:	streamjob3389107036576450975.jar
Application Type:	MAPREDUCE
Application Tags:	
Application Priority:	0 (Higher Integer value indicates higher priority)
YarnApplicationState:	FINISHED
Queue:	default
FinalStatus Reported by AM:	SUCCEEDED
Started:	Thu Nov 22 03:21:26 +0100 2018
Elapsed:	33sec
Tracking URL:	History
Log Aggregation Status:	DISABLED
Diagnostics:	
Unmanaged Application:	false
Application Node Label expression:	<Not set>
AM container Node Label expression:	<DEFAULT_PARTITION>

Application Metrics

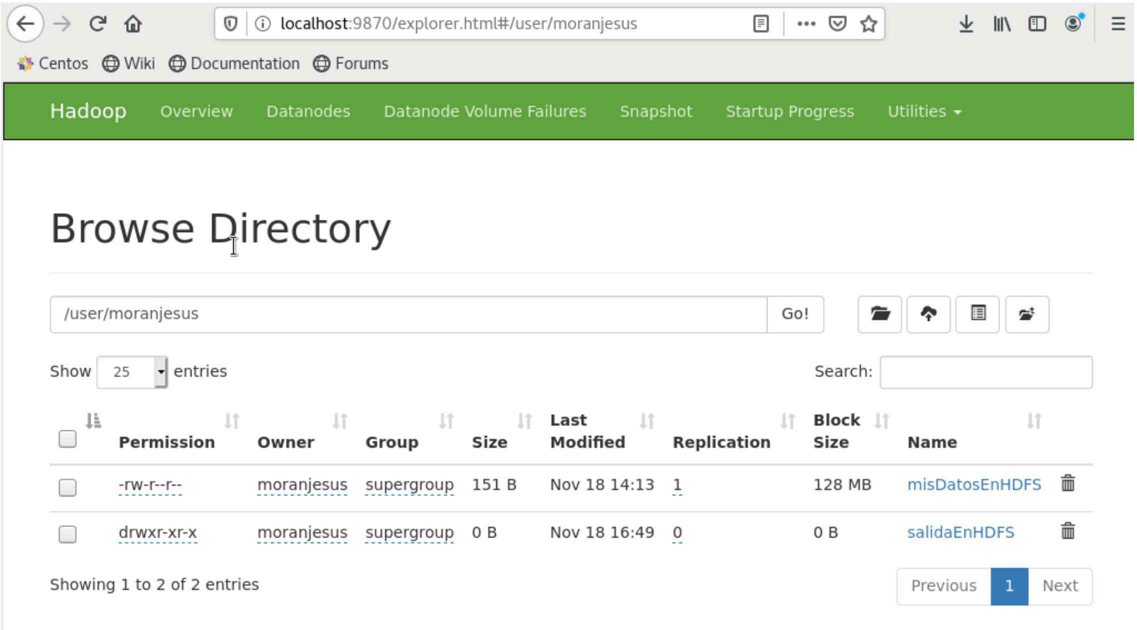
Total Resource Preempted:	<memory:0, vCores:0>
Total Number of Non-AM Containers Preempted:	0

7.5. Comprobamos que la salida se creó en HDFS: hdfs dfs -ls



```
[moranjesus@localhost tempMax]$ hdfs dfs -ls
Found 2 items
-rw-r--r--   1 moranjesus supergroup        150 2021-11-20 14:31 misDatosEnHDFS
drwxr-xr-x   - moranjesus supergroup          0 2021-11-20 14:37 salidaEnHDFS
[moranjesus@localhost tempMax]$ 
```

7.6. Comprobamos desde la interfaz gráfica que se creó: http://localhost:9870 luego ir a utilities -> browse the filesystem -> /user/moranjesus



Browse Directory

/user/moranjesus

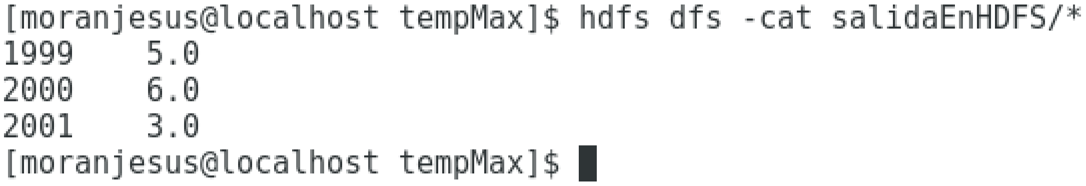
Show 25 entries

Search:

□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
□	-rw-r--r--	moranjesus	supergroup	151 B	Nov 18 14:13	1	128 MB	misDatosEnHDFS	🗑
□	drwxr-xr-x	moranjesus	supergroup	0 B	Nov 18 16:49	0	0 B	salidaEnHDFS	🗑

Showing 1 to 2 of 2 entries

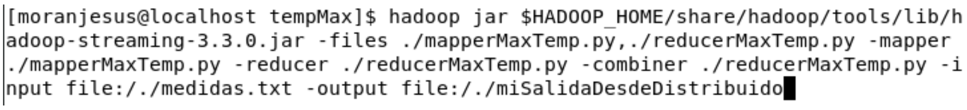
7.7. Si queremos ver los datos de salida podemos descargarlos desde la interfaz gráfica, o también mostrarlos por la terminal (en esto caso podemos porque son pocos, si fuesen GBs de salida, no podríamos): hdfs dfs -cat salidaEnHDFS/*



```
[moranjesus@localhost tempMax]$ hdfs dfs -cat salidaEnHDFS/*
1999    5.0
2000    6.0
2001    3.0
[moranjesus@localhost tempMax]$ 
```

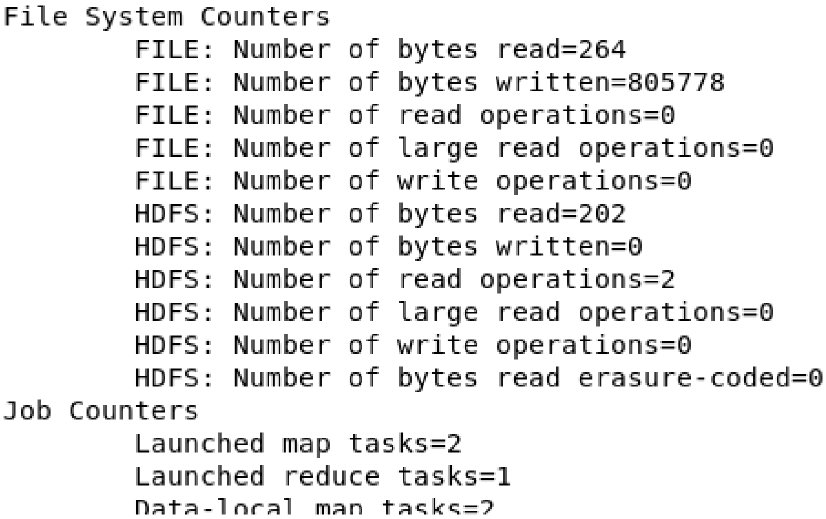
8. Cuando instalamos hadoop distribuido (o pseudodistribuido), el sistema de archivos por defecto deja de ser el local y pasa a ser hdfs. Además, todas las consultas irán relativas al usuario, es decir, si busco miEntrada, realmente estará buscando hdfs:/user/moranjesus/miEntrada (si es un archivo, el archivo, y si es una carpeta, los archivos de esa carpeta). Pero en ocasiones puede interesarnos ejecutar algo con datos del sistema de archivos local. Para ello podemos ejecutarlos utilizando el acrónimo file:/ Por ejemplo:

   hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-3.3.0.jar -files ./mapperMaxTemp.py,./reducerMaxTemp.py -mapper ./mapperMaxTemp.py -reducer ./reducerMaxTemp.py -combiner ./reducerMaxTemp.py -input **file:/**home/moranjesus/Desktop/tempMax/medidas.txt -output **file:/**home/moranjesus/Desktop/tempMax/miSalidaDesdeDistribuido



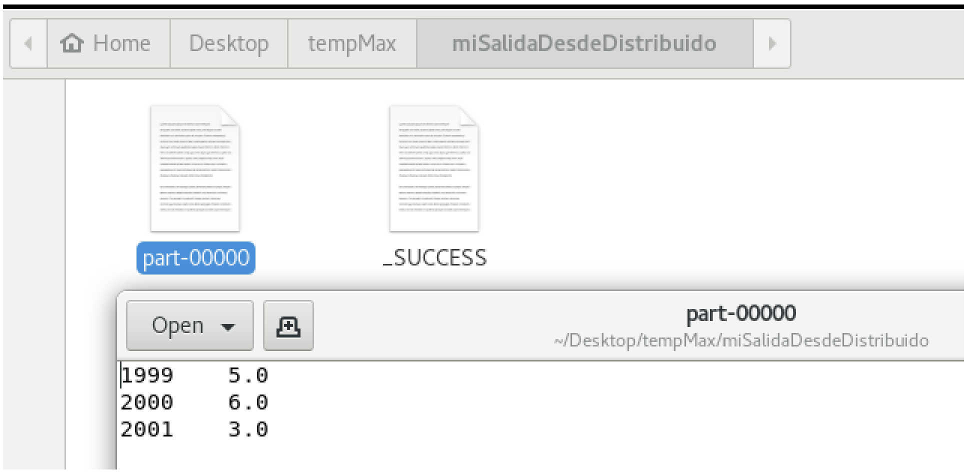
```
[moranjesus@localhost tempMax]$ hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/h
adoop-streaming-3.3.0.jar -files ./mapperMaxTemp.py,./reducerMaxTemp.py -mapper 
./mapperMaxTemp.py -reducer ./reducerMaxTemp.py -combiner ./reducerMaxTemp.py -i
nput file:/./medidas.txt -output file:/./miSalidaDesdeDistribuido
```

8.1. Tras ejecutar, tendremos:



```
File System Counters
        FILE: Number of bytes read=264
        FILE: Number of bytes written=805778
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=202
        HDFS: Number of bytes written=0
        HDFS: Number of read operations=2
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=0
        HDFS: Number of bytes read erasure-coded=0
Job Counters
        Launched map tasks=2
        Launched reduce tasks=1
        Data-local map tasks=2
```

8.2. Los datos se guardaron en nuestro sistema de archivos local (en una carpeta de nuestro ordenador):



```
1999	5.0
2000	6.0
2001	3.0
```

9. Si queremos apagar los servicios de Hadoop: $HADOOP_HOME/sbin/stop-all.sh

   (también se pueden apagar de uno en uno)