Prácticas BigData

1. Montar un cluster real

- Paramos el cluster y el nodo1
- Clonar las máquinas tal y como se indican en el vídeo. En principio, en este curso probamos con 3 nodos, pero si tienes suficiente hardware puedes probar con más.
- En el nodo 1 modificamos el fichero "/opt/hadoop/etc/hadoop/slaves" para añadir los nodos que vamos a crear y que van a ser los esclavos. Debemos asegurarnos de que el nodo1 no está, porque va a ser el maestro. Por ejemplo
  - Nodo2
  - Nodo3
  - ….
- Vamos a generar de nuevo todo el cluster, por lo tanto vamos a gtener que limpiar todo.
- En cada nodo, debemos asegurarnos de:
  - Poner el nombre correcto a la máquina: nodo2, nodo3, etc…
  - En el caso de CENTOS 6 se hace con el comando

    ```
    host nodo2
    ```

    - Y modificando el nombre en el fichero /etc/sysconfig/network
  - En el caso de CENTOS 7 se hace con el comando

    ```
    hostnamectl set-hostname nodo2
    ```

  - Configurar correctamente la dirección IP, según se indica en el vídeo
  - El /etc/hosts debe ser igual en todos los nodos
  - Copiar el fichero "/home/hadoop/.bashrc" del nodo1 al resto de nodos para asegurarnos de que tenemos corresctamente las variables de arranque
  - Generar clave pública y privada con el comando "ssh-keygen" en cada nodo
  - Copiar la clave (/home/hadoop/.ssh/authorized_keys) del nodo1 al resto de nodos para poder tener conectividad ssh.
    - Si queremos poder trabajar desde todos los nodos debemos copiar la clave de todos
  - Copiar hadoop en cda nodo al /opt/hadoop y ponerle los permisos adecuados para el usuario "hadoop"

- Limpiamos el directorio “/datos” en cada nodo. Lo podemos hacer con el comando

```
rm -rf /datos/*
```

- Cambiamos en el hdfs-site.xml el valor de replicación a 3:

```
<property>
      <name>dfs.replication</name>
           <value>3</value>
      </property>
<property>
```

- Nos aseguramos de que en el core-site.xml esté identificado el nodo1 (No puede ser localhost, fallaría)
- **IMPORTANTE**: copiamos el directorio /opt/hadoop/etc/hadoop del nodo1 al resto de nodos, para asegurarnos de que los ficheros de configuración son iguales y por tanto son correctos. Cada vez que cambiemos la configuración debemos copiarlo al resto de nodos.

- Arrancamos las máquinas
- Antes de arrancar el cluster, volvemos a crear el HDFS

hdfs namenode -format

8/01/07 10:30:47 INFO util.GSet: capacity = 2^15 = 32768 entries

18/01/07 10:30:47 INFO namenode.FSImage: Allocated new BlockPoolId: BP-130915252-192.168.56.101-1515317447407

18/01/07 10:30:47 INFO common.Storage: Storage directory /datos/namenode has been successfully formatted.

18/01/07 10:30:47 INFO namenode.FSImageFormatProtobuf: Saving image file /datos/namenode/current/fsimage.ckpt_0000000000000000000 using no compression

18/01/07 10:30:47 INFO namenode.FSImageFormatProtobuf: Image file /datos/namenode/current/fsimage.ckpt_0000000000000000000 of size 323 bytes saved in 0 seconds.

18/01/07 10:30:47 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0

18/01/07 10:30:47 INFO namenode.NameNode: SHUTDOWN_MSG:

/**

SHUTDOWN_MSG: Shutting down NameNode at nodo1/192.168.56.101

**/

- Probamos a arrancar todo el cluster, primero la parte de HDFS. En este caso yo estoy haciendo el ejemplo con 7 nodos, 1 maestro y 6 esclavos

start-dfs.sh

```
Starting namenodes on [localhost]
localhost: starting namenode, logging to /opt/hadoop/logs/hadoop-hadoop-namenode-nodo1.out
nodo2: starting datanode, logging to /opt/hadoop/logs/hadoop-hadoop-datanode-nodo2.out
nodo6: starting datanode, logging to /opt/hadoop/logs/hadoop-hadoop-datanode-nodo6.out
nodo7: starting datanode, logging to /opt/hadoop/logs/hadoop-hadoop-datanode-nodo7.out
nodo5: starting datanode, logging to /opt/hadoop/logs/hadoop-hadoop-datanode-nodo5.out
nodo4: starting datanode, logging to /opt/hadoop/logs/hadoop-hadoop-datanode-nodo4.out
nodo3: starting datanode, logging to /opt/hadoop/logs/hadoop-hadoop-datanode-nodo3.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /opt/hadoop/logs/hadoop-hadoop-secondarynamenode-nodo1.out
```

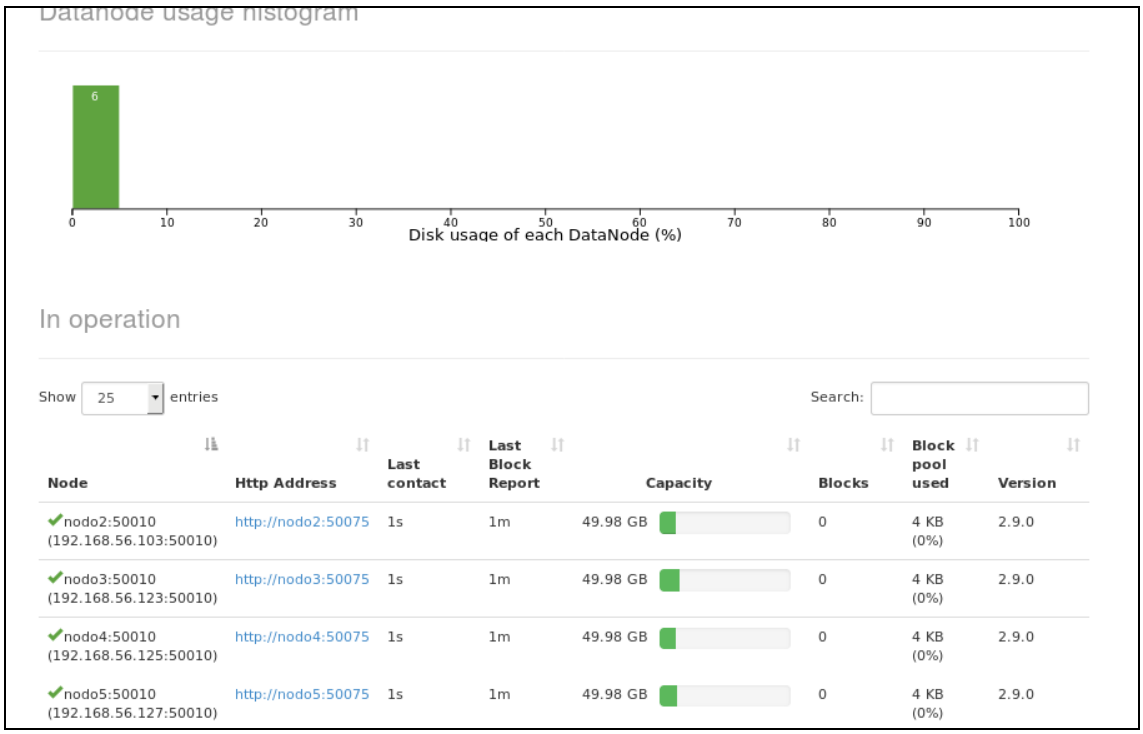
- En el maestro debemos tener estos procesos. Solo los procesos maestros

```
jps
21928 NameNode
22314 Jps
22157 SecondaryNameNode
```

- En cada uno de los esclavos solo debemos tener el datanode

```
jps
3834 Jps
3723 DataNode
```

- En el admin web deben aparecer todos los nodos que hemos creado



- Arrancamos ahora el YARN.

```
start-yarn.sh
```

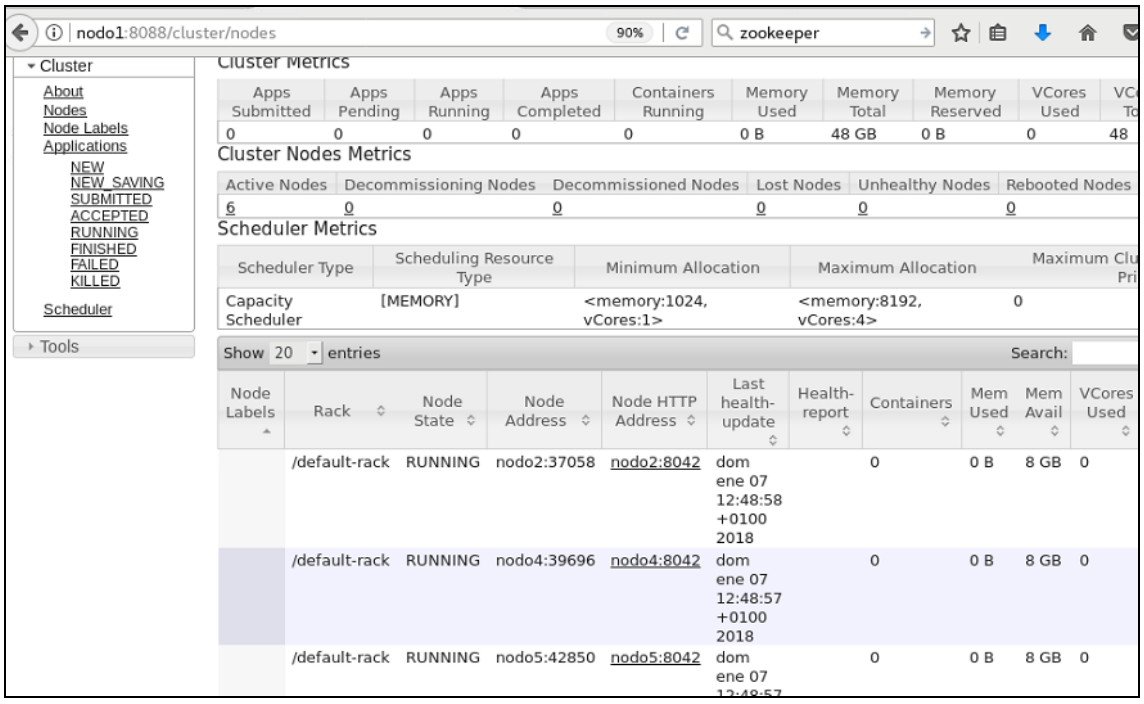
- En este caso, los procesos del maestro deben ser

```
jps
28833 NameNode
29538 Jps
29268 ResourceManager
29062 SecondaryNameNode
```

- Y los de los esclavos

```
jps
6373 Jps
6086 DataNode
6232 NodeManager
```

- En la Web de administración de YARN deben a parecer los nodos



- Por último, en el nodo1, arrancamos el History Server para poder controlar los procesos de Mapper y reducer
- En versión 2 de Haddop usamos el comando

mapred --daemon start historyserver

starting historyserver, logging to /opt/hadoop/logs/mapred-hadoop-historyserver-nodo1.out

jps

28833 NameNode

29907 ResourceManager

29062 SecondaryNameNode

2794 JobHistoryServer

2844 Jps

-

2. Trabajar con el cluster

- Podemos comprobar el cluster con dfsadmin -report
- Aparece información sobre cada nodo

hdfs dfsadmin -report

Configured Capacity: 321965260800 (299.85 GB)
Present Capacity: 281433378816 (262.11 GB)
DFS Remaining: 281433329664 (262.11 GB)
DFS Used: 49152 (48 KB)
DFS Used%: 0.00%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0
Missing blocks (with replication factor 1): 0
Pending deletion blocks: 0

Live datanodes (6):

Name: 192.168.56.103:50010 (nodo2)
Hostname: nodo2
Decommission Status : Normal
Configured Capacity: 53660876800 (49.98 GB)
DFS Used: 8192 (8 KB)
Non DFS Used: 6442340352 (6.00 GB)
DFS Remaining: 47218528256 (43.98 GB)
DFS Used%: 0.00%
DFS Remaining%: 87.99%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Sun Jan 07 12:53:28 CET 2018
Last Block Report: Sun Jan 07 12:41:43 CET 2018

Name: 192.168.56.123:50010 (nodo3)
Hostname: nodo3
Decommission Status : Normal
Configured Capacity: 53660876800 (49.98 GB)
DFS Used: 8192 (8 KB)
Non DFS Used: 7820476416 (7.28 GB)
DFS Remaining: 45840392192 (42.69 GB)
DFS Used%: 0.00%
DFS Remaining%: 85.43%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Sun Jan 07 12:53:28 CET 2018
Last Block Report: Sun Jan 07 12:41:43 CET 2018

Name: 192.168.56.125:50010 (nodo4)
Hostname: nodo4
Decommission Status : Normal
Configured Capacity: 53660876800 (49.98 GB)
DFS Used: 8192 (8 KB)
Non DFS Used: 6591770624 (6.14 GB)
DFS Remaining: 47069097984 (43.84 GB)
DFS Used%: 0.00%
DFS Remaining%: 87.72%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Sun Jan 07 12:53:28 CET 2018
Last Block Report: Sun Jan 07 12:41:43 CET 2018

3. Probar el rendimiento del cluster

- Podemos probar también el rendimiento del Cluster antes de empezar a trabajar con él.
- Existe un comando denominado TestDFIO en la libraría “hadoop-mapreduce-client-jobclient-2.9.0-tests.jar” que permite indicar el número de ficheros y el tamaño de cada uno par que nos haga un benchmark.
    - Se encuentra en el directorio /opt/hadoop/share/hadoop/mapreduce
- Por ejemplo, para probar 10 ficheros de 100MG cada uno

```
hadoop jar hadoop-mapreduce-client-jobclient-2.9.0-tests.jar TestDFSIO -write -nrFiles 10 -fileSize 100
```

- El resultado será similar al siguiente

```
8/01/07 13:14:02 INFO fs.TestDFSIO: ----- TestDFSIO ----- : write
18/01/07 13:14:02 INFO fs.TestDFSIO:            Date & time: Sun Jan 07 13:14:02 CET 2018
18/01/07 13:14:02 INFO fs.TestDFSIO:        Number of files: 10
18/01/07 13:14:02 INFO fs.TestDFSIO: Total MBytes processed: 1000
18/01/07 13:14:02 INFO fs.TestDFSIO:      Throughput mb/sec: 12,22
18/01/07 13:14:02 INFO fs.TestDFSIO: Average IO rate mb/sec: 12,71
18/01/07 13:14:02 INFO fs.TestDFSIO:  IO rate std deviation: 2,56
18/01/07 13:14:02 INFO fs.TestDFSIO:     Test exec time sec: 71,13
```

- También lo Podemos lanzar en modo “read” para ver el rendimiento en lectura