

## **>> ANEXO 2. CONFIGURACIÓN DE UN CLÚSTER EN HADOOP**

En este documento, se van a abordar los siguientes contenidos:

- Explicar los modos de funcionamiento de Hadoop.
- Parámetros de configuración más significativos.
- Características en clústers de distintos tamaños.
- Herramientas de monitorización: Ganglia y Nagios.
- Aprender a configurar un nodo principal y un nodo secundario.
- Realizar un ejercicio de puesta en marcha de un clúster de Hadoop.

### **Modos de funcionamiento**

Hadoop presenta los siguientes modos de funcionamiento:

- Standalone:
  - Mismo proceso Java.
  - Sin sistema de archivos distribuido.
  - Sistema de archivos local.
  - Depuración de aplicaciones.
- Pseudodistribuido:
  - Varios procesos Java para cada uno de los demonios (NameNode, DataNode, JobTracker, TaskTracker).
  - Todos los procesos sobre la misma máquina.
  - Sistema de archivos sobre HDFS.
- Distribuido:
  - Varios procesos Java.
  - Varias máquinas cada una con distintos demonios configurados.
  - Sistema de archivos distribuidos.
  - conf/hadoop-site.xml
  - conf/masters
  - conf/slaves

### Selección de las máquinas

Para llevar a cabo la selección de máquinas para nuestro clúster, debemos tener en cuenta algunas consideraciones:

- Hadoop está diseñado para trabajar con cualquier tipo de *hardware*.
- Lenguaje de programación y uso de RAM:
  - Normalmente los *jobs* de Hadoop escritos en Java consumen entre 1 y 2 GB de RAM.
  - Python aumenta el consumo de memoria.
- Número de discos por máquina:
  - Aumentar el número de discos no siempre es la mejor solución al desaprovechar el rendimiento en paralelo.
  - Ejemplo: tres máquinas con cuatro discos frente a una máquina de doce discos: se aprovecha el procesamiento en paralelo y la salida será más rápida.
- Red:
  - Conexiones Gigabit Ethernet mejoran el rendimiento frente a otras redes de peor rendimiento.

A continuación, vemos un benchmark realizado sobre dos máquinas diferentes para la obtención de métricas sobre el funcionamiento de Hadoop.

### Hardware<sup>1</sup>

Cluster name	CPU model	CPU freq	Cores	RAM	Disk size	Disk interface	Disk rpm	Disks	Network type	Number of machines	Number of racks
Herd1	Intel Xeon LV	2.0ghz	4	4gb	0.25tb	SATA	7200rpm	4	GigE	35	2
Herd2	Intel Xeon 5320	1.86ghz	8	8gb	0.75tb	SATA2	7200rpm	4	GigE	20	2

**Tabla 1.** Hardware.  
*Fuente:* Hadoop wiki.

### Benchmark

Cluster name	Version	Sort time s	Mappers	Reducers	Max map tasks / node	Max reduce tasks / node	Map speculative ex	Reduce speculative ex	Parallel copies	Sort mb	Sort factor
Herd1	0.14.3	3977 s	5600	175	?	?	Yes	Yes	20	200	10
Herd2	0.18.3	1715 s	1520	136	7	8	No	Yes	20	100	50

**Tabla 2.** Datos de la comparación.  
*Fuente:* Hadoop wiki.

Por último, se recomienda la consulta del siguiente enlace: Hadoop Wiki. “Machine Scaling”. [En línea] URL disponible en: <http://wiki.apache.org/hadoop/MachineScaling>

<sup>1</sup> Hadoop Wiki: “Cluster benchmark”. [En línea] URL disponible en: <http://wiki.apache.org/hadoop/HardwareBenchmarks>

A continuación, se detallan algunas consideraciones que se deben tener en cuenta a la hora de la implementación de clústers según su tamaño:

### Clústers pequeños: 2-10 nodos

- Clúster de 2 máquinas:
  - Primer nodo, corren todos los demonios: NameNode, JobTracker, DataNode y TaskTracker.
  - Segundo nodo tendrá solo los demonios de DataNode y TaskTracker.
- Clúster de 3 o más máquinas:
  - Primer nodo, corren los demonios de gestión: NameNode y JobTracker.
  - Segundo y tercer nodos: tendrán solo los demonios de DataNode y TaskTracker.
- Factor de replicación =3 para rangos de entre 8/10 nodos. Valores superiores a 3 no suelen ser necesarios.
- Cambios en configuración: conf/hadoop-site.xml
- Ficheros individuales grandes, es preferible ajustar sus niveles de replicación para no sobrecargar la red.

```
<configuration>
  <property>
    <name>mapred.job.tracker</name>
    <value>head.server.node.com:9001</value>
  </property>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://head.server.node.com:9000</value>
  </property>
  <property>
    <name>dfs.data.dir</name>
    <value>/home/hadoop/dfs/data</value>
    <final>true</final>
  </property>
  <property>
    <name>dfs.name.dir</name>
    <value>/home/hadoop/dfs/name</value>
    <final>true</final>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/tmp/hadoop</value>
    <final>true</final>
  </property>
  <property>
    <name>mapred.system.dir</name>
    <value>/hadoop/mapred/system</value>
    <final>true</final>
  </property>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
  </property>
</configuration>
```

### Clústers medianos: 10- 40 nodos

Características comunes:

- Suelen ocupar un único rack.
- Suele ser necesaria la alta disponibilidad.
- El NameNode es el único punto de fallo.

Creación de *backups* configurando `hadoop-site.xml`. Almacenaremos todos los metadatos en las rutas indicadas separadas por comas.

```
<property>
  <name>dfs.name.dir</name>
  <value>/home/hadoop/dfs/name,/mnt/namenode-backup</value>
  <final>true</final>
</property>
```

- La máquina de *backup* se suele emplear para que corra el demonio `SecondaryNameNode`.
  - Los metadatos del Namenode se almacenan en un fichero llamado `fsimage` + `editlog` (indicadores asociados a la imagen).
  - La mezcla de los dos archivos anteriores da lugar a un segundo fichero llamado `fsimage`.
  - Este proceso de mezcla consume mucha memoria.
- `conf/slaves`:
  - `DataNodes`.
  - `TaskTracker`.
- `conf/masters`: se deberá indicar el nodo secundario.
  - `NameNode`.
  - `JobTracker`.
- Tareas de mantenimiento del clúster: desinstalación de nodos.
- Modificaremos el fichero `conf/hadoop-site.xml` para crear un fichero de exclusiones en el que incluiremos en el futuro las máquinas del clúster que no

queremos que estén configuradas, tanto para el NameNode como del Job Tracker.

```
<property>
  <name>dfs.hosts.exclude</name>
  <value>/home/hadoop/excludes</value>
  <final>true</final>
</property>
<property>
  <name>mapred.hosts.exclude</name>
  <value>/home/hadoop/excludes</value>
  <final>true</final>
</property>
```

- HDFS no reserva espacio libre para el DataNode.

```
<property>
  <name>dfs.datanode.du.reserved</name>
  <value>1073741824</value>
  <final>true</final>
</property>
```

- Tamaño de pila asociado a cada tarea. Por defecto son 200 MB.

```
<property>
  <name>mapred.child.java.opts</name>
  <value>-Xmx512m</value>
</property>
```

- Discos por máquina.
- HDFS: dfs.data.dir

```
<property>
  <name>dfs.data.dir</name>
  <value>/d1/dfs/data,/d2/dfs/data,/d3/dfs/data,/d4/dfs/data</value>
  <final>true</final>
</property>
```

- MapReduce: mapred.local.dir

```
<property>
  <name>mapred.local.dir</name>
  <value>/d1/mapred/local,/d2/mapred/local,/d3/mapred/local,/d4/mapred/local</value>
  <final>true</final>
</property>
```

- NameNode: dfs.name.dir
  - Replicación de datos.
  - Problemas de I/O.

### Clústers grandes: múltiples racks

- Gestión de los metadatos en el NameNode:
  - Máquina con más memoria RAM para mantener los bloques de memoria de modo eficiente.
  - Por defecto el tamaño del bloque son 64 MB, pero podemos aumentarlo para disminuir el número de los mismos.

```
<property>
  <name>dfs.block.size</name>
  <value>134217728</value>
</property>
```

- Establecer el SecondaryNameNode en otro rack par evitar fallos de un rack individual.
- Gestión de peticiones de estado de salud de los nodos: tanto por el demonio NameNode como del JobTracker, se recomienda aumentar el número de hilos dedicados a esta tarea.

```
<property>
  <name>dfs.namenode.handler.count</name>
  <value>40</value>
</property>
<property>
  <name>mapred.job.tracker.handler.count</name>
  <value>40</value>
</property>
```

- Otras propiedades aplicables a clústers de entre 250 y 2.000 nodos

Property	Range	Description
io.file.buffer.size	32768-131072	Read/write buffer size used in SequenceFiles (should be in multiples of the hardware page size)
io.sort.factor	50-200	Number of streams to merge concurrently when sorting files during shuffling
io.sort.mb	50-200	Amount of memory to use while sorting data
mapred.reduce.parallel.copies	20-50	Number of concurrent connections a reducer should use when fetching its input from mappers
tasktracker.http.threads	40-50	Number of threads each TaskTracker uses to provide intermediate map output to reducers
mapred.tasktracker.map.tasks.maximum	$1/2 * (\text{cores/node})$ to $2 * (\text{cores/node})$	Number of map tasks to deploy on each machine.
mapred.tasktracker.reduce.tasks.maximum	$1/2 * (\text{cores/node})$ to $2 * (\text{cores/node})$	Number of reduce tasks to deploy on each machine.

**Tabla 3.** Propiedades de clústers grandes.

*Fuente:* Fundación Apache.

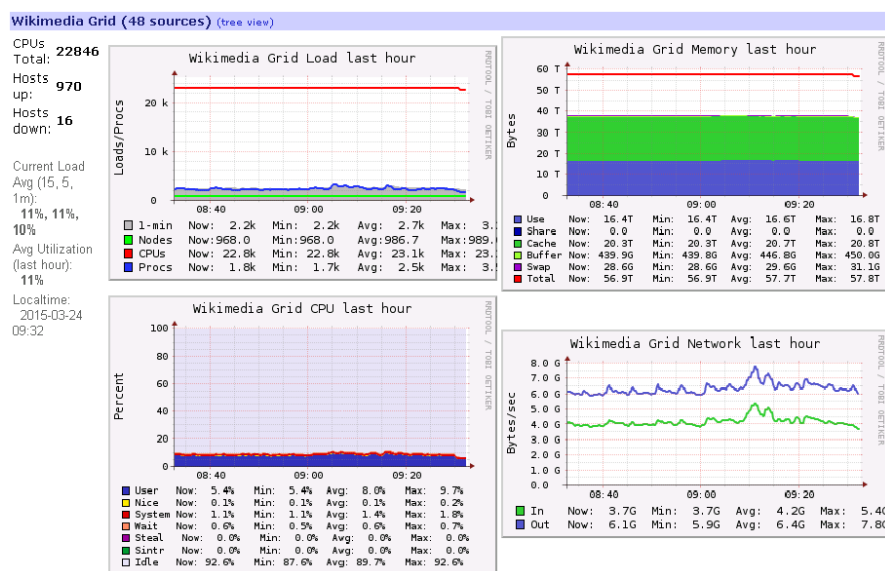


### Monitorización del rendimiento

Existen diversas herramientas para la monitorización de clústers en cuanto a su salud y funcionamiento; algunas incluso tienen métricas del propio ecosistema integradas dentro de la herramienta. Algunos ejemplos son:

#### Ganglia

<http://ganglia.sourceforge.net>  
<http://it-ebooks.info/book/1283/>



- *Framework* de monitorización para sistemas distribuidos.
- Recopila métricas de máquinas individuales y las envía a un sistema agregador para facilitar información global al administrador del clúster.
- Permite integración con otras aplicaciones como por ejemplo Hadoop.  
<http://wiki.apache.org/hadoop/GangliaMetrics>
- Instalar y configurar Ganglia teniendo el proceso gmond corriendo en cada máquina del clúster y el proceso gmetad.
- Crear un fichero de configuración en \$HADOOP\_HOME/conf/hadoop-metrics.properties.

```
dfs.class=org.apache.hadoop.metrics.ganglia.GangliaContext
dfs.period=10
dfs.servers=localhost:8649
```

```
mapred.class=org.apache.hadoop.metrics.ganglia.GangliaContext
mapred.period=10
mapred.servers=localhost:8649
```

## Nagios

<http://www.nagios.org>

<http://www.it-ebooks.info/book/3517/>



- Herramientas de diagnóstico del clúster como información de red, discos y uso de CPU.
- Varios *dashboards* personalizados.
- Visualización en APP.

### **Ejercicio práctico**

Con la información presentada en este anexo, se han de realizar las siguientes tareas:

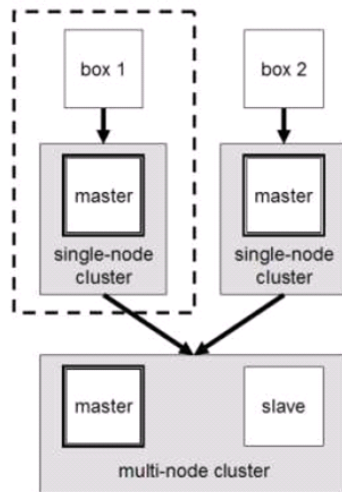
1. Formar grupos de cinco personas.
2. Cada persona montará una MV.
3. Identificar qué máquina tendrá el rol de máster y cuáles el rol de esclavo.
4. Configurar el clúster.
5. Descargar cinco libros de <https://www.gutenberg.org/>
6. Subir los cinco libros a HDFS en formato txt.
7. Ejecutar un wordcount dando como entrada el directorio donde hemos subido los libros.
8. Revisar las distintas interfaces para ver cómo se ejecutó el job.

Para la resolución de este ejercicio es necesario utilizar varias máquinas virtuales, por lo que es recomendable hacerla en grupo. Se trata de una actividad completamente voluntaria y extraordinaria al contenido del curso.

### Configuración del clúster

Separar configuración de ruta de instalación:

- Usar ruta por defecto.
- Crear variable de entorno HADOOP\_CONF\_DIR.
- Indicarlo en los arranques y paradas:
  - \$HADOOP\_INSTALL/hadoop/bin/start-all.sh -config /home/bigdata/hadoop-config
  - HADOOP\_INSTALL/hadoop/bin/stop-all.sh -config /home/bigdata/hadoop-config

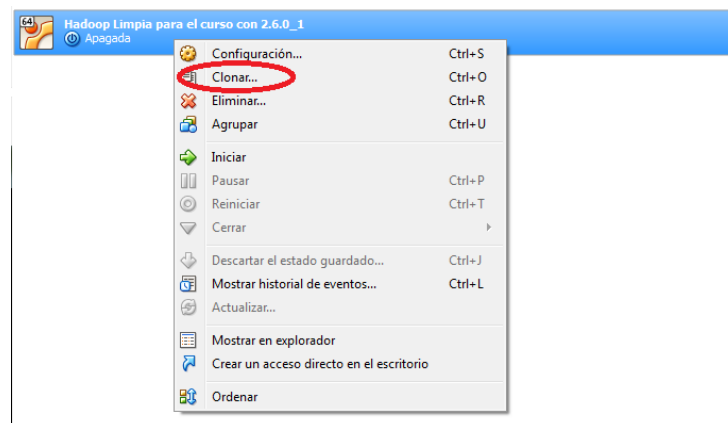


### Creación del clúster

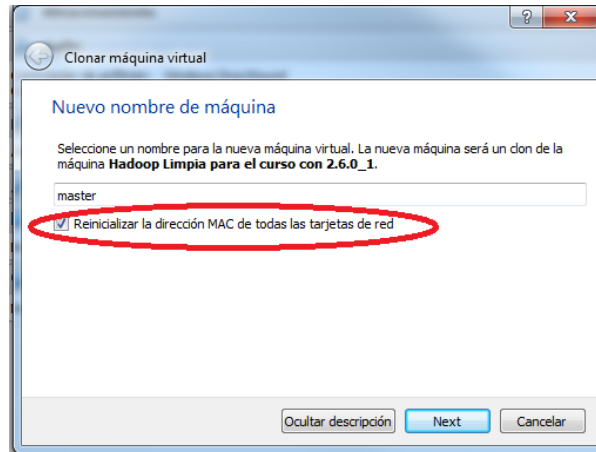
- Establecer comunicación entre las máquinas.
- Instalar certificados.
- Configurar las máquinas que actuarán como maestra y esclavo.
- Formatear el sistema de archivos.
- Arrancar el clúster.

### Creación de mi primer clúster

- Para empezar a configurar nuestro primer clúster de Hadoop, lo primero que debemos hacer es localizar la máquina virtual que hemos utilizado para las sesiones *online* de Hadoop.
- Una vez localizada la máquina virtual, pasaremos a crear dos copias de ella. Una para el nodo maestro y otra para el esclavo.

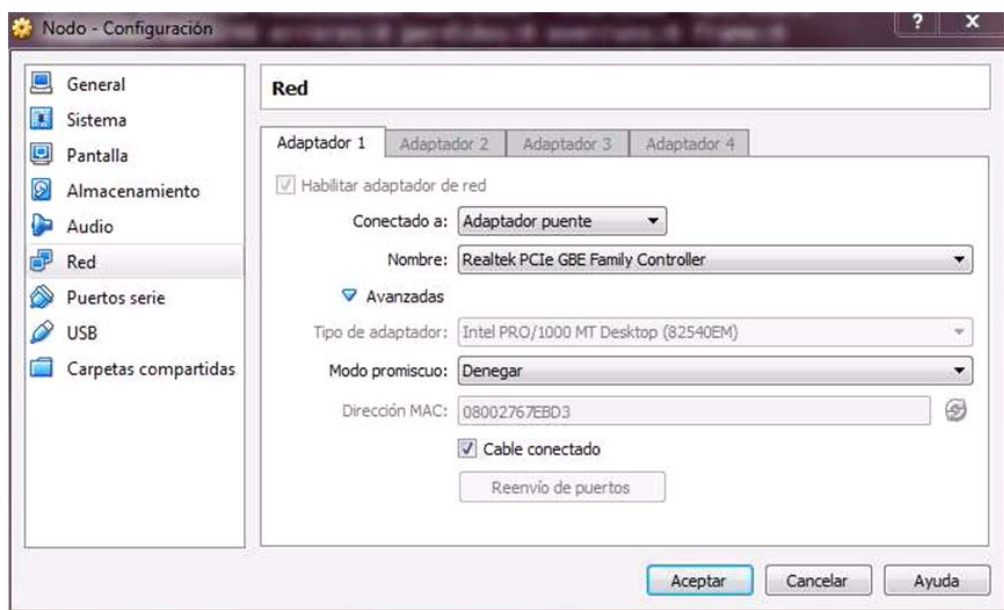


### Proceso de clonación



### Revisión de las interfaces de red

Una vez finalizado el proceso de clonación, será necesario revisar la configuración de las tarjetas de red de las máquinas con las que vayamos a trabajar.



### Hostname

Al trabajar en un entorno virtual y clonar la misma máquina, todas ellas tendrán el mismo nombre de host. Esto supone un problema para el funcionamiento del clúster.

Por ello, será necesario modificar este nombre en todas las máquinas a través del comando:

```
sudo nano /etc/hostname
```

```
bigdata@master:~$ sudo nano /etc/hostname
```

Bastará con establecer un criterio para nombrar las distintas máquinas, por ejemplo: master, slave1, slave2...

```
GNU nano 2.2.6 Archivo: /etc/hostname
master
```

### Revisión de las interfaces de red

Lo primero que debemos comprobar nada más encender la máquina será qué IP nos ha asignado el servidor DHCP del espacio en el que estemos.

```
bigdata@bigdata:~$ ifconfig
eth0      Link encap:Ethernet  direcciónHW 08:00:27:0b:7a:3e
          Dirección inet6: fe80::a00:27ff:fe0b:7a3e/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST  MTU:1500  Métrica:1
          Paquetes RX:111 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:121 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:1000
          Bytes RX:11077 (11.0 KB)  TX bytes:22679 (22.6 KB)

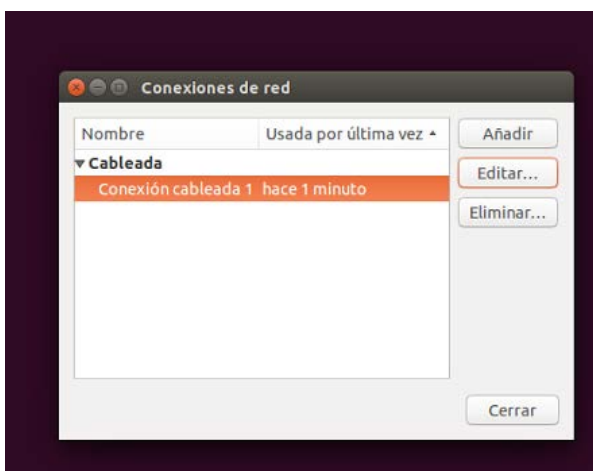
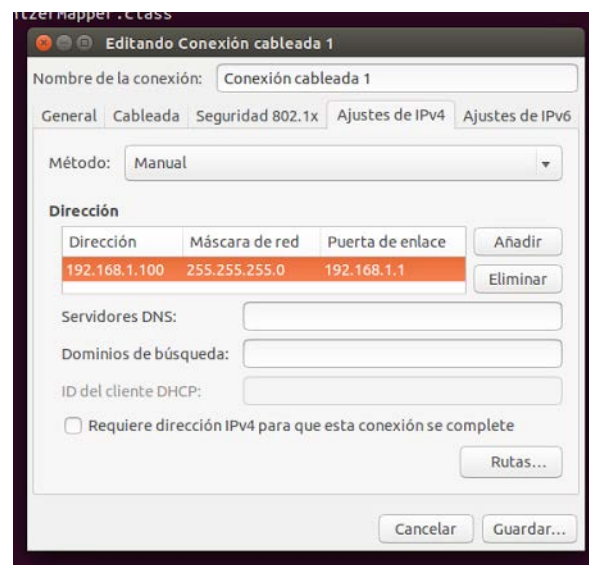
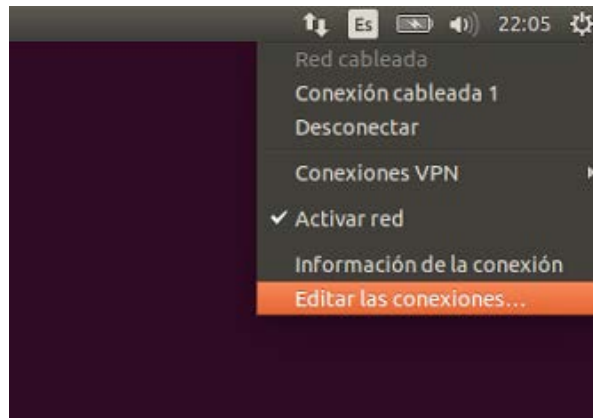
lo        Link encap:Bucle local
          Direc. inet:127.0.0.1  Másc:255.0.0.0
          Dirección inet6: ::1/128 Alcance:Anfitrión
          ACTIVO BUCLE FUNCIONANDO  MTU:65536  Métrica:1
          Paquetes RX:167 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:167 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:0
          Bytes RX:12081 (12.0 KB)  TX bytes:12081 (12.0 KB)
```

Una medida que conviene tener en cuenta es asignar a las máquinas una IP estática para tener localizado cada nodo del clúster.

### Asignación de IP estática

Para la asignación de la IP estática tenemos dos opciones:

- Opción 1: en la parte superior derecha del escritorio, pinchar en el icono de la “red”.



- Opción 2:
  - Accedemos al fichero de configuración de interfaces:

```
bigdata@bigdata:~$ sudo nano /etc/network/interfaces
```

- Incluir la configuración de red correspondiente:

```
GNU nano 2.2.6 Archivo: /etc/network/interfaces

# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
address 192.168.1.201
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.1
dns-nameservers 8.8.8.8
```

- Reiniciar las interfaces del ordenador para aplicar los cambios:

```
bigdata@bigdata:~$ sudo nano /etc/network/interfaces
[sudo] password for bigdata:
bigdata@bigdata:~$ sudo /etc/init.d/networking restart
```

- Si después del paso anterior ejecutamos “ifconfig” y aún no se nos ha actualizado la configuración de red:

```
bigdata@bigdata:~$ sudo init 6
```

### Conexión ssh

Para hacer algunos ejemplos puede que tengamos la necesidad de conectarnos a internet desde la máquina virtual. Para conseguir esto en nuestro entorno debemos configurar al menos un servidor DNS.

```
$ sudo nano /etc/resolv.conf
```



E incluimos la siguiente línea:

```
# Configuración de servidores DNS
nameserver 8.8.8.8
```

Reinicia las interfaces de red para aplicar los cambios. Esto puedes hacerlo de la siguiente manera:

```
$ sudo /etc/init.d/networking restart
```

Si tuvieras algún problema con el comando anterior, puedes probar a deshabilitar y habilitar de nuevo la interfaz de red que acabamos de configurar de la siguiente manera:

```
$ sudo ifconfig eth0 down
$ sudo ifconfig eth0 up
```

Comprueba que tienes conectividad con otros equipos de la red y que tienes conexión a internet.

```
$ ping 192.168.1.1 (para comprobar conectividad con tu puerta de enlace)
$ ping google.com (para comprobar conexión a Internet)
```

### Comprobación configuración de red

Con el comando “ifconfig” podemos confirmar que la configuración de red ha surtido efecto.

```
bigdata@bigdata:~$ ifconfig
eth0      Link encap:Ethernet  direcciónHW 08:00:27:0b:7a:3e
          Direc. inet:192.168.1.200  Difus.:192.168.1.255  Másc:255.255.255.0
          Dirección inet6: fe80::a00:27ff:fe0b:7a3e/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
          Paquetes RX:48 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:46 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:1000
          Bytes RX:4744 (4.7 KB)  TX bytes:7270 (7.2 KB)

lo        Link encap:Bucle local
          Direc. inet:127.0.0.1  Másc:255.0.0.0
          Dirección inet6: ::1/128 Alcance:Anfitrión
          ACTIVO BUCLE FUNCIONANDO MTU:65536 Métrica:1
          Paquetes RX:191 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:191 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:0
          Bytes RX:13665 (13.6 KB)  TX bytes:13665 (13.6 KB)
```

Una vez obtengamos el resultado esperado en el paso anterior, debemos pasar a probar la comunicación entre nuestros nodos. Para ello debemos emplear los comandos siguientes:

- *ping 192.168.1.201* desde el nodo maestro (en mi caso).
- *ping 192.168.1.200* desde el nodo esclavo (en mi caso).

```
bigdata@bigdata:~$ ping 192.168.1.201
PING 192.168.1.201 (192.168.1.201) 56(84) bytes of data:
64 bytes from 192.168.1.201: icmp_seq=1 ttl=64 time=2.54 ms
64 bytes from 192.168.1.201: icmp_seq=2 ttl=64 time=1.59 ms
64 bytes from 192.168.1.201: icmp_seq=3 ttl=64 time=0.828 ms
64 bytes from 192.168.1.201: icmp_seq=4 ttl=64 time=0.466 ms
64 bytes from 192.168.1.201: icmp_seq=5 ttl=64 time=0.730 ms
64 bytes from 192.168.1.201: icmp_seq=6 ttl=64 time=1.51 ms
64 bytes from 192.168.1.201: icmp_seq=7 ttl=64 time=0.550 ms
64 bytes from 192.168.1.201: icmp_seq=8 ttl=64 time=1.49 ms
^C
--- 192.168.1.201 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7018ms
rtt min/avg/max/mdev = 0.466/1.215/2.544/0.659 ms
```

### Asignación de nombres de host

Para aislar nuestro clúster de los cambios que podamos hacer en nuestra red es necesario que los nodos de nuestro clúster se comuniquen a través de nombres. Para ello debemos configurar el fichero `/etc/hosts` (en todos los nodos) con el comando:

*sudo nano /etc/hosts*

Este comando abrirá el siguiente archivo, debemos incluir las líneas marcadas:

```
GNU nano 2.2.6 Archivo: /etc/hosts
127.0.0.1    localhost
127.0.1.1    bigdata

192.168.1.200 master
192.168.1.201 slave

# 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
```

### Autenticación entre los nodos

Una vez conectadas las máquinas, debemos solucionar los temas de autenticación.

El usuario bigdata en el nodo maestro (bigdata@master) debe ser capaz de conectarse:

- A su misma cuenta de usuario en el master.
- A la cuenta del usuario bigdata en el esclavo (bigdata@slave) a través de un login SSH sin contraseña. Para ello solo es necesario añadir la contraseña pública del usuario bigdata@master (que debería estar en \$HOME/.ssh/id\_rsa.pub) a las claves autorizadas del bigdata@slave (en \$HOME/.ssh/authorized\_keys). Esto se puede realizar con el siguiente comando:

```
ssh-copy-id -i $HOME/.ssh/id_rsa.pub bigdata@slave
```

```
bigdata@bigdata:~$ ssh-copy-id -i $HOME/.ssh/id_rsa.pub bigdata@slave
The authenticity of host 'slave (192.168.1.201)' can't be established.
ECDSA key fingerprint is 86:ca:81:2d:e5:56:8b:d7:eb:75:a6:58:17:da:45:bb.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are
already installed

/usr/bin/ssh-copy-id: WARNING: All keys were skipped because they already exist on the remote sys
tem.
```

### Generación de credenciales ssh

Si no tuviéramos la clave generada, los pasos para hacerlo son:

#### *Paso 1. Generar la clave*

Mediante el siguiente comando se puede generar un par de claves públicas y privadas. El parámetro -t especifica el tipo de clave generada que en este caso empleará el algoritmo rsa.

```
$ ssh-keygen -t rsa -P ""
```

A todas las preguntas que se realicen pulsaremos intro o indicaremos yes. Como resultado se generarán dos ficheros id\_rsa and id\_rsa.pub en la carpeta .ssh en el directorio home o root.

### *Paso 2. Establecer autenticación*

Ahora se puede copiar la clave pública del nodo maestro a los distintos nodos esclavos mediante el siguiente comando:

```
$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys  
$ ssh-copy-id -i /root/.ssh/id_rsa.pub bigdata@slave
```

### Pruebas de autenticación

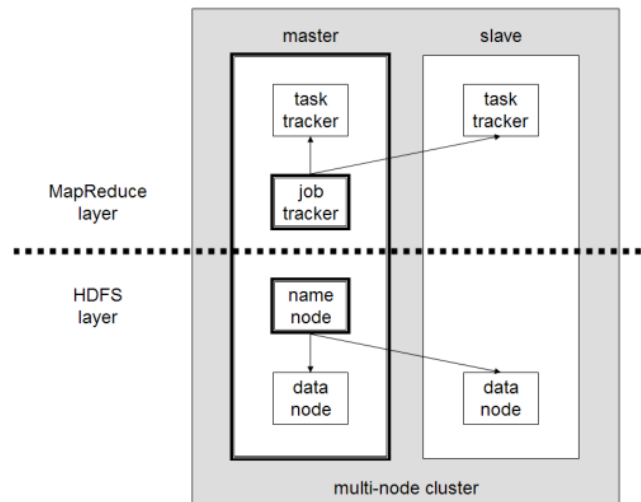
Una vez configurados los accesos, realizaremos pruebas de conexión como se observa en las pantallas siguientes:

```
bigdata@bigdata:~$ ssh slave  
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-32-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com/
```

```
bigdata@bigdata:~$ ssh master  
The authenticity of host 'master (192.168.1.200)' can't be established.  
ECDSA key fingerprint is 86:ca:81:2d:e5:56:8b:d7:eb:75:a6:58:17:da:45:bb.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'master,192.168.1.200' (ECDSA) to the list of known hosts.  
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-32-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com/
```

### Configuración del clúster

Una vez tenemos las dos máquinas comunicadas, vamos a explicar cómo configurar una máquina Ubuntu como nodo maestro y otra como nodo esclavo. El nodo maestro actuará también como nodo esclavo debido a que solo hay dos máquinas disponibles en el clúster pero necesitamos distribuir la información y el procesamiento entre múltiples máquinas.



**Figura 1.** Demonios clúster.  
*Fuente:* Fundación Apache.

El nodo maestro lanzará los demonios para cada capa:

- Namenode para la capa de almacenamiento en HDFS.
- Jobtracker para la capa de procesamiento de MapReduce.

Ambas máquinas ejecutarán demonios esclavos:

- Datanode para la capa de HDFS.
- TaskTracker para la capa de MapReduce.

*Paso 1. Crear ficheros master (solo en el nodo maestro)*

```
bigdata@bigdata:~/hadoop/etc/hadoop$ sudo nano masters
```

Incluimos la siguiente línea:

```
GNU nano 2.2.6 Archivo: masters
master
|
```

A pesar de su nombre, el archivo `conf/masters` define en qué máquinas Hadoop arrancará los `SecondaryNameNode` en el clúster. En nuestro caso, es solo la máquina master.

*Paso 2. Crear ficheros slaves (solo en el nodo maestro)*

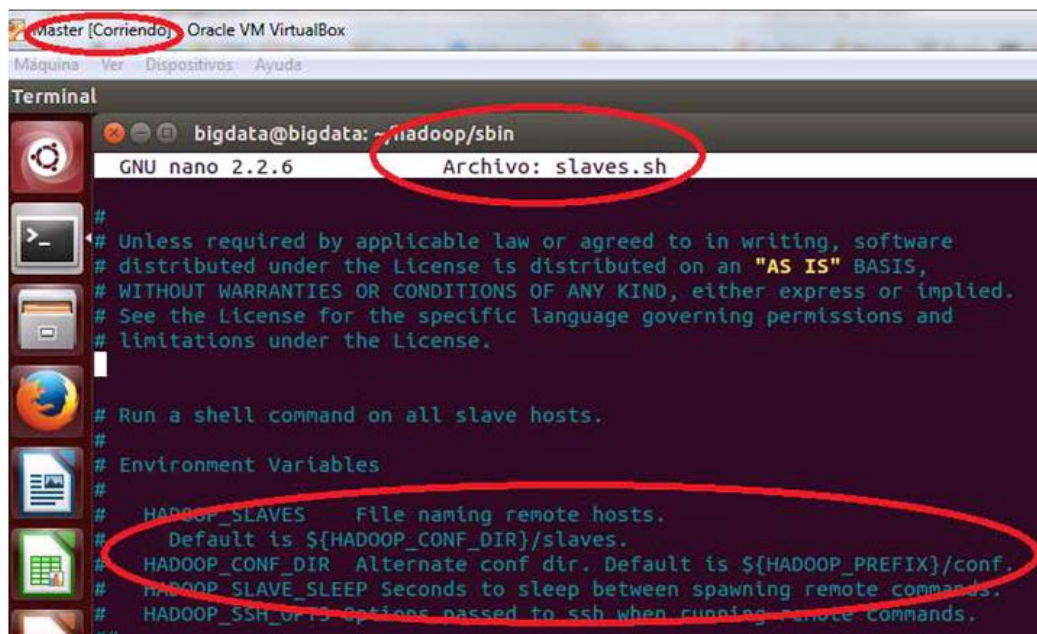
```
bigdata@bigdata:~/hadoop/etc/hadoop$ sudo nano slaves
```

Incluimos la siguiente línea:

```
GNU nano 2.2.6 Archivo: slaves
master
slave
```

Actualizar el fichero `etc/hadoop/slaves` y añadir un host esclavo por línea.

En el archivo `sbin/slaves.sh` ubicado en el directorio donde en su momento instalamos Hadoop se pueden consultar las rutas por defecto con la configuración de Hadoop. Una buena práctica es mover estos directorios para independizar la configuración de la instalación.



*Paso 3. Editar el fichero etc/hadoop/core-site.xml (todos los nodos)*

```
bigdata@bigdata:~/hadoop/etc/hadoop$ sudo nano core-site.xml
```

Modificamos el parámetro [fs.default.name](#) que especifica el NameNode (HDFS master) host y puerto.

```
<configuration>
<property>
  <name>fs.default.name</name>
  <value>hdfs://master:54310</value>
</property>
</configuration>
```

*Paso 4. Editar el fichero etc/hadoop/mapred-site.xml (todos los nodos)*

```
bigdata@bigdata:~/hadoop/etc/hadoop$ sudo nano mapred-site.xml
```

Este fichero especifica el puerto y host en el que corre el jobtracker. En el debemos añadir:

```
<property>
<name>mapred.job.tracker</name>
<value>master:54311</value>
<description>El host y puerto en el que el job tracker de MapReduce corre.
Si se indica "local", entonces los jobs correrán en una única tarea
de map-reduce.
</description>
</property>
```

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

<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>

<property>
<name>mapred.job.tracker</name>
<value>master:54311</value>
<description>El host y puerto en el que el job tracker de MapReduce corre.
Si se indica "local", entonces los jobs correrán en una única tarea
de map-reduce.
</description>
</property>
</configuration>
```



*Paso 5. Editar el fichero etc/hadoop/hdfs-site.xml (solo en el maestro)*

```
bigdata@bigdata:~/hadoop/etc/hadoop$ sudo nano hdfs-site.xml
```

Este fichero se emplea para especificar el grado de replicación. Define en cuántas máquinas un fichero individual debería replicarse antes de estar disponible.

El valor por defecto de replicación es 3, pero como en este caso solo trabajamos con 2 nodos indicaremos este valor.

```
<property>
<name>dfs.replication</name>
<value>2</value>
<description>Valor por defecto de bloque de replicación.
El número actual de replications puede especificarse al crear el fichero.
Este valor por defecto se emplea si no se indica valor en el momento de la
creación.
</description>
</property>
```

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

<configuration>
<property>
<name>dfs.replication</name>
<value>2</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>file:/home/bigdata/hadoop_store/hdfs/namenode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>file:/home/bigdata/hadoop_store/hdfs/datanode</value>
</property>
</configuration>
```


*Paso 6. Editar el fichero etc/hadoop/yarn-site.xml (todos los nodos)*

```
bigdata@master:~/hadoop/etc/hadoop$ sudo nano yarn-site.xml
```

En este fichero debes añadir la propiedad que especifica la máquina y el puerto de resource manager contra el que deberán registrarse todas las máquinas del clúster.



```
<property>
<name>yarn.resourcemanager.hostname</name>
<value>master</value>
</property>
```



```
GNU nano 2.2.6 Archivo: 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

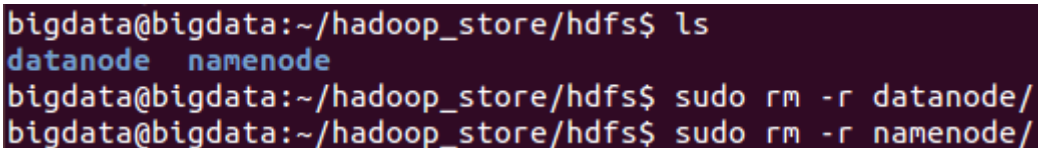
    http://www.apache.org/licenses/LICENSE-2.0

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.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>

<property>
  <name>yarn.resourcemanager.hostname</name>
  <value>master</value>
</property>
</configuration>
```

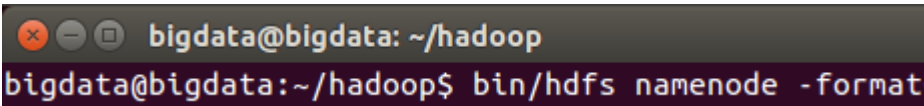
### Formatear HDFS

Antes de formatear el sistema de archivos HDFS, es necesario borrar las siguientes carpetas:



```
bigdata@bigdata:~/hadoop_store/hdfs$ ls
datanode namenode
bigdata@bigdata:~/hadoop_store/hdfs$ sudo rm -r datanode/
bigdata@bigdata:~/hadoop_store/hdfs$ sudo rm -r namenode/
```

Una vez hecho esto podemos pasar a realizar el formateo.



```
bigdata@bigdata: ~/hadoop
bigdata@bigdata:~/hadoop$ bin/hdfs namenode -format
```

```

15/05/12 18:22:31 INFO namenode.FSNamesystem: dfs.namenode.safemode.min.datanodes = 0
15/05/12 18:22:31 INFO namenode.FSNamesystem: dfs.namenode.safemode.extension = 30000
15/05/12 18:22:31 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
15/05/12 18:22:31 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry c
ache entry expiry time is 600000 millis
15/05/12 18:22:31 INFO util.GSet: Computing capacity for map NameNodeRetryCache
15/05/12 18:22:31 INFO util.GSet: VM type = 64-bit
15/05/12 18:22:31 INFO util.GSet: 0.029999999329447746% max memory 966.7 MB = 297.0 KB
15/05/12 18:22:31 INFO util.GSet: capacity = 2^15 = 32768 entries
15/05/12 18:22:31 INFO namenode.NNConf: ACLs enabled? false
15/05/12 18:22:31 INFO namenode.NNConf: XAttrs enabled? true
15/05/12 18:22:31 INFO namenode.NNConf: Maximum size of an xattr: 16384
15/05/12 18:22:31 INFO namenode.FSImage: Allocated new BlockPoolId: BP-476154643-127.0.1.1-143144
7751286
15/05/12 18:22:31 INFO common.Storage: Storage directory /home/bigdata/hadoop_store/hdfs/namenode
has been successfully formatted.
15/05/12 18:22:32 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >=
0
15/05/12 18:22:32 INFO util.ExitUtil: Exiting with status 0
15/05/12 18:22:32 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at bigdata/127.0.1.1
*****/

```

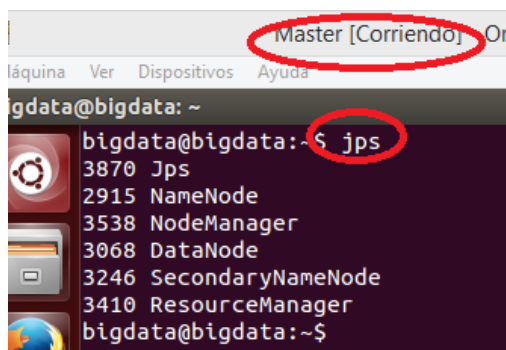
### Demonios: arranque desde master

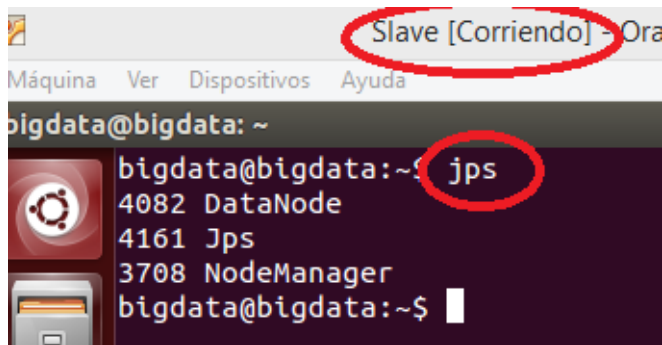
Una vez realizado el formateo solo no quedará arrancar los demonios en el nodo maestro. Si hemos configurado de manera correcta nuestro clúster, no será necesario arrancar nada en los nodos esclavos. De esta tarea se encargará el nodo maestro.

Para arrancar los demonios utilizaremos los dos siguientes comandos (en el mater):

```
start-dfs.sh
start-yarn.sh
```

Una vez ejecutados los comandos de arranque de los demonios, utilizaremos el comando *jps* para confirmar que todos los demonios se están ejecutando de manera correcta:



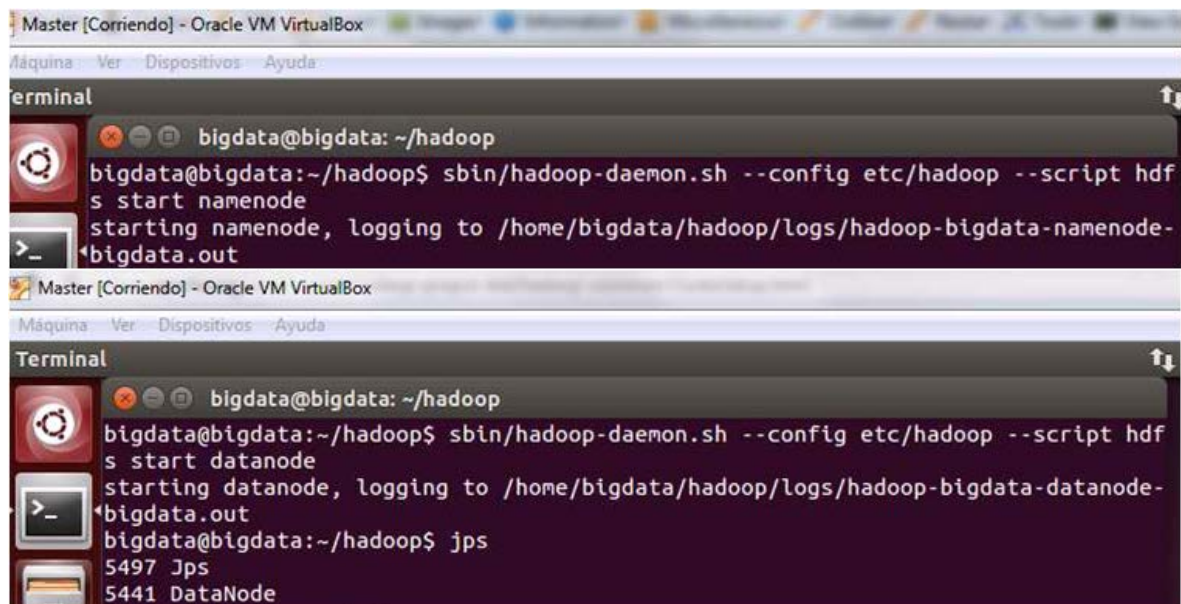


```
Slave [Corriendo] - Ora
Máquina Ver Dispositivos Ayuda
bigdata@bigdata: ~
bigdata@bigdata:~$ jps
4082 DataNode
4161 Jps
3708 NodeManager
bigdata@bigdata:~$
```

### Demonios: arranque individual

Arrancar los demonios del maestro individualmente:

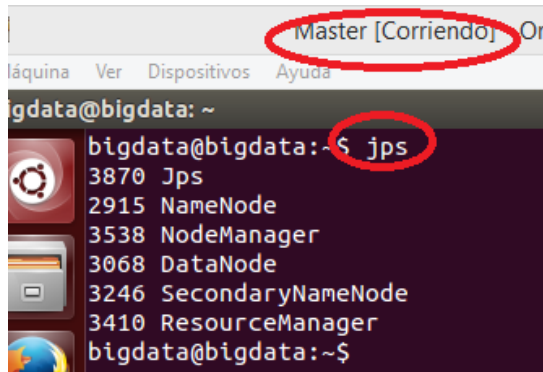
- `$HADOOP_HOME/sbin/hadoop-daemon.sh --config $HADOOP_HOME/etc/hadoop --script hdfs start namenode`
- `$HADOOP_HOME/sbin/hadoop-daemon.sh --config $HADOOP_HOME/etc/hadoop --script hdfs start datanode`
- `$HADOOP_HOME/sbin/yarn-daemon.sh start resourcemanager`
- `$HADOOP_HOME/sbin/yarn-daemon.sh start nodemanager`
- `$HADOOP_HOME/sbin/mr-jobhistory-daemon.sh start historyserver`



```
Master [Corriendo] - Oracle VM VirtualBox
Máquina Ver Dispositivos Ayuda
terminal
bigdata@bigdata: ~/hadoop
bigdata@bigdata:~/hadoop$ sbin/hadoop-daemon.sh --config etc/hadoop --script hdfs start namenode
starting namenode, logging to /home/bigdata/hadoop/logs/hadoop-bigdata-namenode-bigdata.out
bigdata@bigdata:~/hadoop$

Master [Corriendo] - Oracle VM VirtualBox
Máquina Ver Dispositivos Ayuda
Terminal
bigdata@bigdata: ~/hadoop
bigdata@bigdata:~/hadoop$ sbin/hadoop-daemon.sh --config etc/hadoop --script hdfs start datanode
starting datanode, logging to /home/bigdata/hadoop/logs/hadoop-bigdata-datanode-bigdata.out
bigdata@bigdata:~/hadoop$ jps
5497 Jps
5441 DataNode
```

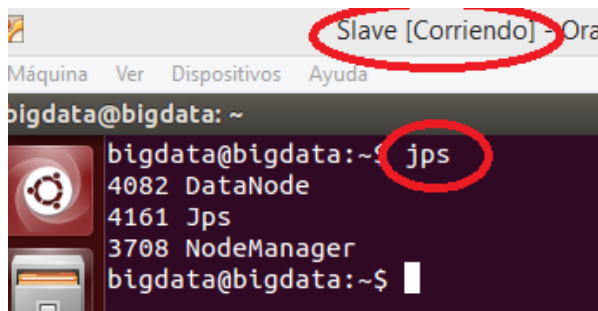
### Demonios: arranque individual



```
bigdata@bigdata:~$ jps
3870 Jps
2915 NameNode
3538 NodeManager
3068 DataNode
3246 SecondaryNameNode
3410 ResourceManager
bigdata@bigdata:~$
```

Arrancar los demonios del esclavo individualmente:

```
sudo rm -r /home/bigdata/hadoop_store/hdfs/datanode/
$HADOOP_HOME/sbin/hadoop-daemon.sh start datanode
$HADOOP_HOME/sbin/yarn-daemon.sh start nodemanager
```



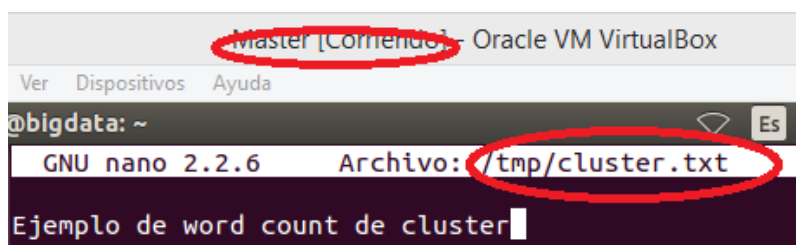
```
bigdata@bigdata:~$ jps
4082 DataNode
4161 Jps
3708 NodeManager
bigdata@bigdata:~$
```

### Comprobación del funcionamiento

Creamos un fichero en el master para realizar un wordcount.

1. Creamos el fichero:

```
sudo nano /tmp/cluster.txt
```



```
GNU nano 2.2.6 Archivo: /tmp/cluster.txt
Ejemplo de word count de cluster
```

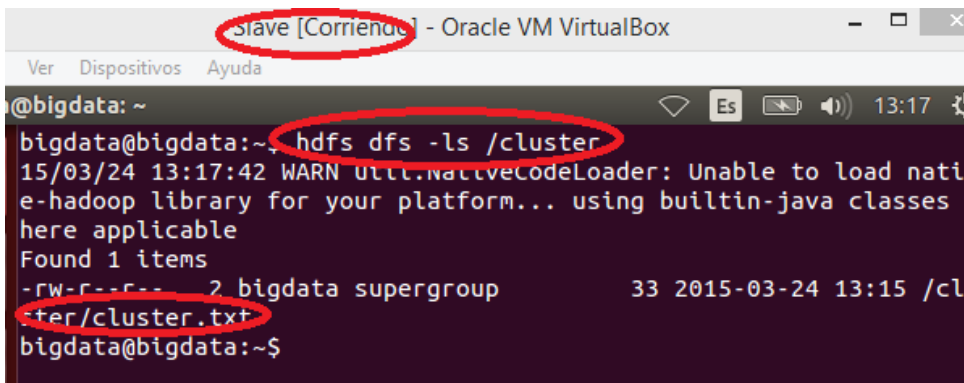
2. Subimos el fichero a HDFS a una carpeta llamada cluster:

```
hdfs dfs -mkdir /cluster
hdfs dfs -put /tmp/cluster.txt /cluster
```

```
bigdata@bigdata:~$ hdfs dfs -mkdir /cluster
bigdata@bigdata:~$ hdfs dfs -put /tmp/cluster.txt /cluster
15/03/24 13:15:17 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

3. Comprobamos desde la máquina esclava que el fichero es accesible:

```
hdfs dfs -ls /cluster
```



```
@bigdata: ~
bigdata@bigdata:~$ hdfs dfs -ls /cluster
15/03/24 13:17:42 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 1 items
-rw-r--r-- 2 bigdata supergroup 33 2015-03-24 13:15 /cluster/cluster.txt
bigdata@bigdata:~$
```

4. Ejecutamos la instrucción de wordcount desde la maestra:

```
hadoop jar /home/bigdata/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.8.0.jar wordcount /cluster /out
```

```
bigdata@bigdata:~$ hdfs dfs -put /tmp/cluster.txt /cluster
15/03/24 13:15:17 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
bigdata@bigdata:~$ hadoop jar /home/bigdata/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.0.jar wordcount /cluster /out
15/03/24 13:20:41 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
15/03/24 13:20:42 INFO client.RMProxy: Connecting to ResourceManager
```

5. Desde la máquina secundaria, ejecutamos un ls para ver el resultado si ha sido satisfactorio:

```
hdfs dfs -ls /out
```

```
bigdata@bigdata:~$ hdfs dfs -ls /out
15/03/24 13:23:44 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r--  2 bigdata supergroup          0 2015-03-24 13:21 /out/_SUCCESS
-rw-r--r--  2 bigdata supergroup       40 2015-03-24 13:21 /out/part-r-00000
bigdata@bigdata:~$ hdfs dfs cat /out/part-r-00000
```

6. Comprobamos el resultado:

```
hdfs dfs -cat /out/part-r-00000
```

```
bigdata@bigdata:~$ hdfs dfs cat /out/part-r-00000
cat: Unknown command
Did you mean -cat? This command begins with a dash.
bigdata@bigdata:~$ hdfs dfs -cat /out/part-r-00000
15/03/24 13:24:05 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Ejemplo 1
cluster 1
count   1
de      2
word    1
bigdata@bigdata:~$
```

Una vez ejecutado el job, podremos consultar las distintas interfaces web para consultar cómo se desarrolló el trabajo en nuestro clúster.


### ResourceManager

<http://master:8088>

En el ResourceManager podremos ver los nodos disponibles para el procesamiento paralelo.



## UD 2. HERRAMIENTAS DEL ECOSISTEMA HADOOP



Nodes of the cluster

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
5	0	0	5	0	0 B	40 GB	0 B	0	40	0	5	0	0	0	0

Show: 20 entries


Node Labels	Rack	Node State	Node Address	Node HTTP Address	Last health-update	Health-report	Containers	Mem Used	Mem Avail	VCores Used	VCores Avail	Version
/default-rack		RUNNING	slave03:33889	slave03:8042	14-may-2015 14:20:18		0	0 B	8 GB	0	8	2.6.0
/default-rack		RUNNING	slave:50053	slave:8042	14-may-2015 14:20:14		0	0 B	8 GB	0	8	2.6.0
/default-rack		RUNNING	master:39407	master:8042	14-may-2015 14:20:22		0	0 B	8 GB	0	8	2.6.0
/default-rack		RUNNING	slave04:59457	slave04:8042	14-may-2015 14:20:17		0	0 B	8 GB	0	8	2.6.0
/default-rack		RUNNING	slave02:41147	slave02:8042	14-may-2015 14:20:18		0	0 B	8 GB	0	8	2.6.0

Showing 1 to 5 of 5 entries

### Job History

<http://master:19888>

En el Job History podremos consultar el histórico de trabajos realizados, así como información asociada a estos trabajos como número de maps y de reduce.



MapReduce Job job\_1431602541054\_0005

Job Overview

Job Name:	word count
User Name:	bigdata
Queue:	default
State:	SUCCEEDED
Uberized:	false
Submitted:	Thu May 14 14:18:15 CEST 2015
Started:	Thu May 14 14:18:25 CEST 2015
Finished:	Thu May 14 14:19:18 CEST 2015
Elapsed:	53sec
Diagnostics:	
Average Map Time:	32sec
Average Shuffle Time:	23sec
Average Merge Time:	0sec
Average Reduce Time:	36sec


ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Thu May 14 14:18:17 CEST 2015	slave:8042	logs

Task Type	Total	Complete
Map	6	6
Reduce	1	1

Attempt Type	Failed	Killed	Successful
Maps	0	1	6
Reduces	0	0	1

También podremos consultar cada operación map y reduce por separado, ver en que nodo se realizó el trabajo, etc.



SUCCESSFUL MAP attempts in job\_1431602541054\_0005

Show: 20 entries

Attempt	State	Status	Node	Logs	Start Time	Finish Time	Elapsed Time	Note
attempt_1431602541054_0005_m_000000_0	SUCCEEDED	map	default: rack3/axe0d:8042	logs	Thu, 14 May 2015 12:18:28 GMT	Thu, 14 May 2015 12:18:53 GMT	25sec	
attempt_1431602541054_0005_m_000001_1	SUCCEEDED	map	default: rack3/axe0d:8042	logs	Thu, 14 May 2015 12:19:05 GMT	Thu, 14 May 2015 12:19:12 GMT	6sec	
attempt_1431602541054_0005_m_000002_0	SUCCEEDED	map	default: rack3/axe0d:8042	logs	Thu, 14 May 2015 12:18:27 GMT	Thu, 14 May 2015 12:19:14 GMT	47sec	
attempt_1431602541054_0005_m_000003_0	SUCCEEDED	map	default: rack3/axe0d:8042	logs	Thu, 14 May 2015 12:18:27 GMT	Thu, 14 May 2015 12:19:13 GMT	45sec	
attempt_1431602541054_0005_m_000004_0	SUCCEEDED	map	default: rack3/axe0d:8042	logs	Thu, 14 May 2015 12:18:27 GMT	Thu, 14 May 2015 12:19:13 GMT	45sec	
attempt_1431602541054_0005_m_000005_0	SUCCEEDED	map	default: rack3/axe0d:8042	logs	Thu, 14 May 2015 12:18:28 GMT	Thu, 14 May 2015 12:18:53 GMT	25sec	

Showing 1 to 6 of 6 entries



## SUCCESSFUL REDUCE attempts in job\_1431602541054\_0005

Logged in as: drago

Attempt	State	Status	Node	Logs	Start Time	Shuffle Finish Time	Merge Finish Time	Finish Time	Elapsed Time Shuffle	Elapsed Time Merge	Elapsed Time Reduce	Elapsed Time	Note
attempt_1431602541054_0005_r_000000_0	SUCCEEDED	reduce > reduce	/default: rackslave09-8042	logs	Thu, 14 May 2013 12:18:56 GMT	Thu, 14 May 2013 12:19:20 GMT	Thu, 14 May 2013 12:19:20 GMT	Thu, 14 May 2013 12:19:17 GMT	23sec	0sec	N/A	21sec	

## Interface Namenode

<http://localhost:50070>

En esta interfaz podremos consultar los nodos que participan en el almacenamiento de HDFS e información asociada a los nodos.

Hadoop	Overview	Datanodes	Snapshot	Startup Progress	Utilities
--------	----------	-----------	----------	------------------	-----------

## Datanode Information

In operation

Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
slave (192.168.1.201:50010)	1	In Service	28.42 GB	5.61 MB	6.47 GB	21.95 GB	16	5.61 MB (0.02%)	0	2.6.0
slave03 (192.168.1.203:50010)	0	In Service	28.42 GB	10.52 MB	6.43 GB	21.98 GB	21	10.52 MB (0.04%)	0	2.6.0
slave04 (192.168.1.204:50010)	0	In Service	28.42 GB	11.8 MB	5.78 GB	22.63 GB	27	11.8 MB (0.04%)	0	2.6.0
master (192.168.1.200:50010)	1	In Service	28.42 GB	3.65 MB	6.49 GB	21.92 GB	11	3.65 MB (0.01%)	0	2.6.0
slave02 (192.168.1.202:50010)	0	In Service	28.42 GB	7.95 MB	6.43 GB	21.98 GB	18	7.95 MB (0.03%)	0	2.6.0

También nos permitirá navegar por el sistema de archivos, consultar el factor de replicación o ver en qué nodos existe una copia de un archivo completo.

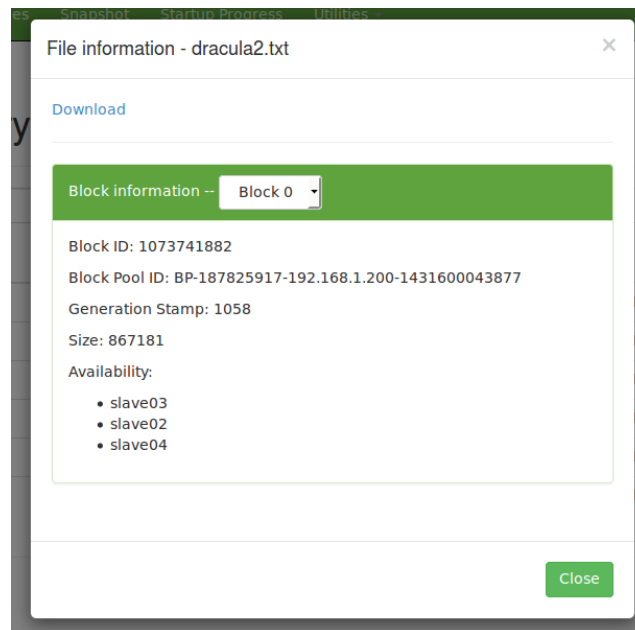
Hadoop	Overview	Datanodes	Snapshot	Startup Progress	Utilities
--------	----------	-----------	----------	------------------	-----------

## Browse Directory

/cluster2						Go
Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	bigdata	supergroup	846.86 KB	3	128 MB	<a href="#">dracula.txt</a>
-rw-r--r--	bigdata	supergroup	846.86 KB	3	128 MB	<a href="#">dracula2.txt</a>
-rw-r--r--	bigdata	supergroup	846.86 KB	3	128 MB	<a href="#">dracula3.txt</a>
-rw-r--r--	bigdata	supergroup	846.86 KB	3	128 MB	<a href="#">dracula4.txt</a>
-rw-r--r--	bigdata	supergroup	846.86 KB	3	128 MB	<a href="#">dracula5.txt</a>
-rw-r--r--	bigdata	supergroup	846.86 KB	3	128 MB	<a href="#">dracula6.txt</a>

Hadoop, 2014.





Por último, ¿qué ocurre si apagamos el nodo slave04 simulando un error?

Hadoop Overview Datanodes Snapshot Startup Progress Utilities

### Datanode Information

In operation

Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
slave (192.168.1.201:50010)	1	In Service	28.42 GB	8.96 MB	6.47 GB	21.94 GB	26	8.96 MB (0.03%)	0	2.6.0
slave03 (192.168.1.203:50010)	0	In Service	28.42 GB	13.18 MB	6.43 GB	21.98 GB	30	13.18 MB (0.05%)	0	2.6.0
master (192.168.1.200:50010)	1	In Service	28.42 GB	6.84 MB	6.49 GB	21.92 GB	20	6.84 MB (0.02%)	0	2.6.0
slave02 (192.168.1.202:50010)	2	In Service	28.42 GB	11.59 MB	6.43 GB	21.98 GB	26	11.59 MB (0.04%)	0	2.6.0
slave04 (192.168.1.204:50010)	Thu May 14 2015 16:15:30 GMT+0200 (CEST)	Dead	-	-	-	-	-	-	-	-

