

---

# **BIG DATA**

## **MONTAR UN CLUSTER REAL**

**EDUARD LARA**

# 1. CLONAR EL NODO HADOOP

---

- Montaremos un clúster real. Simularemos como mínimo 3 nodos: un maestro y dos esclavos.
- Tenemos dos opciones:
  - Contar con una máquina muy gorda en la cual disponer de varias máquinas virtuales
  - Disponer de varios PC's sencillos en los que ir poniendo distintas máquinas virtuales.
- Evidentemente cuanto más nodos montemos más real será nuestro cluster hadoop
- El resto de nodos los sacaremos haciendo una copia de la máquina que hemos estado utilizando y que hemos llamado Nodo1.
- Crearemos al menos el nodo2 y el nodo3.

```
hadoop@nodo1:~$ uname -a
Linux nodo1 5.15.0-60-generic #66-Ubuntu SMP Fri Jan 20 14:29:49 UTC 2023 x86_64
x86_64 x86_64 GNU/Linux
hadoop@nodo1:~$ █
```

# 1. CLONAR EL NODO HADOOP

---

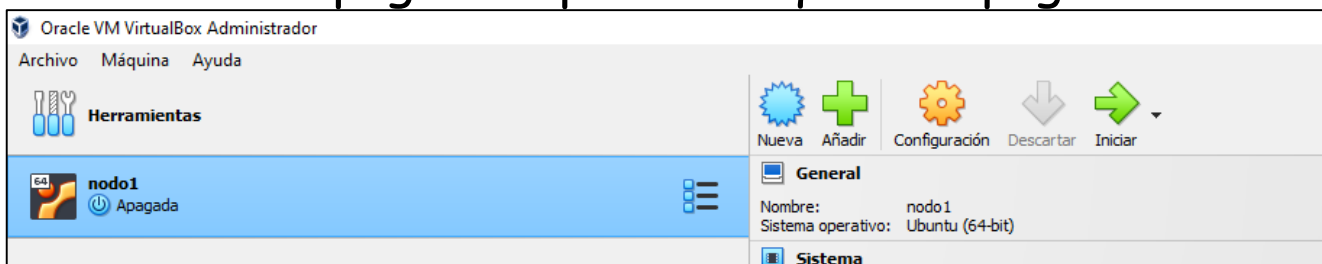
Paso 1. Antes de apagar la máquina nodo1 pararemos YARN y HDFS con **stop-yarn.sh** y **stop-dfs.sh**

```
hadoop@nodo1:~$ stop-yarn.sh
Stopping nodemanagers
Stopping resourcemanager
hadoop@nodo1:~$ stop-dfs.sh
Stopping namenodes on [nodo1]
Stopping datanodes
Stopping secondary namenodes [nodo1]
hadoop@nodo1:~$
```

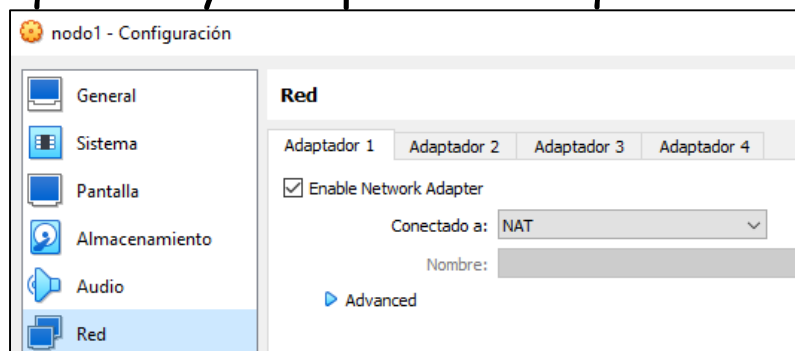
Paso 2. Instalamos la aplicacion ifconfig  
**sudo apt install net-tools**

# 1. CLONAR EL NODO HADOOP

**Paso 2.** Para apagar el nodo1 podemos hacerlo mediante los comandos **su root** y **init 0**. O en la administración del sistema Ubuntu, haremos click en el botón Apagar. Esperamos que se apague



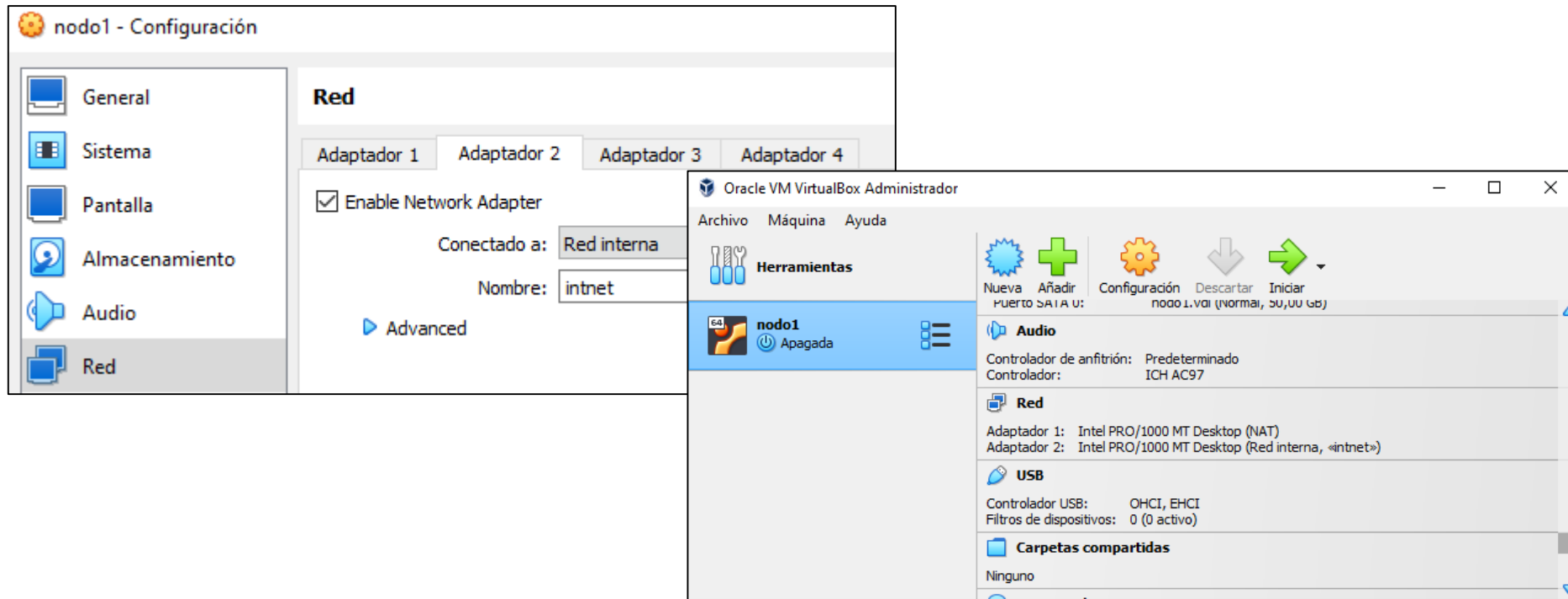
**Paso 3.** Una vez apagado nuestra máquina nodo1, la configuraremos para poder clonarla. Vamos a Configuración de Red donde vemos que Adaptador 1 esta conectado a la red de Internet mediante NAT. Lo dejamos así ya que esta interface es la que vamos a utilizar para descargar los paquetes y componentes que necesitamos de Internet



# 1. CLONAR EL NODO HADOOP

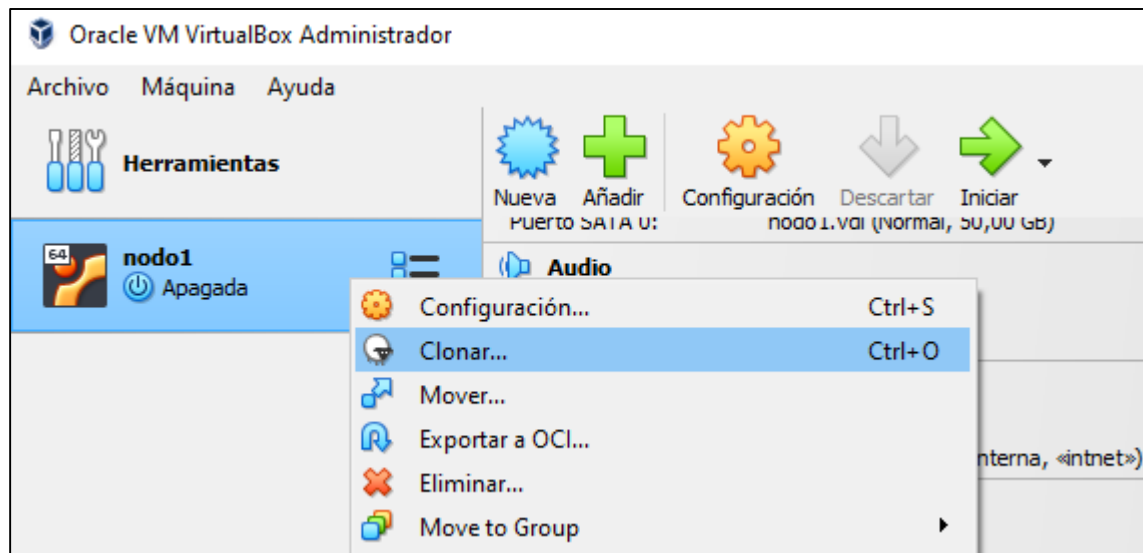
**Paso 4.** Crearemos una segunda tarjeta de red. Habilitaremos el Adaptador 2 para que sea de tipo red interna. Esta es la red que vamos a utilizar para las conexiones entre los nodos Hadoop. Esta será la interfaz que configuraremos exclusivamente a partir de ahora.

Hacemos click en aceptar y podemos comprobar que la maquina virtual nodo1 ha quedado configurado con dos tarjetas de red.



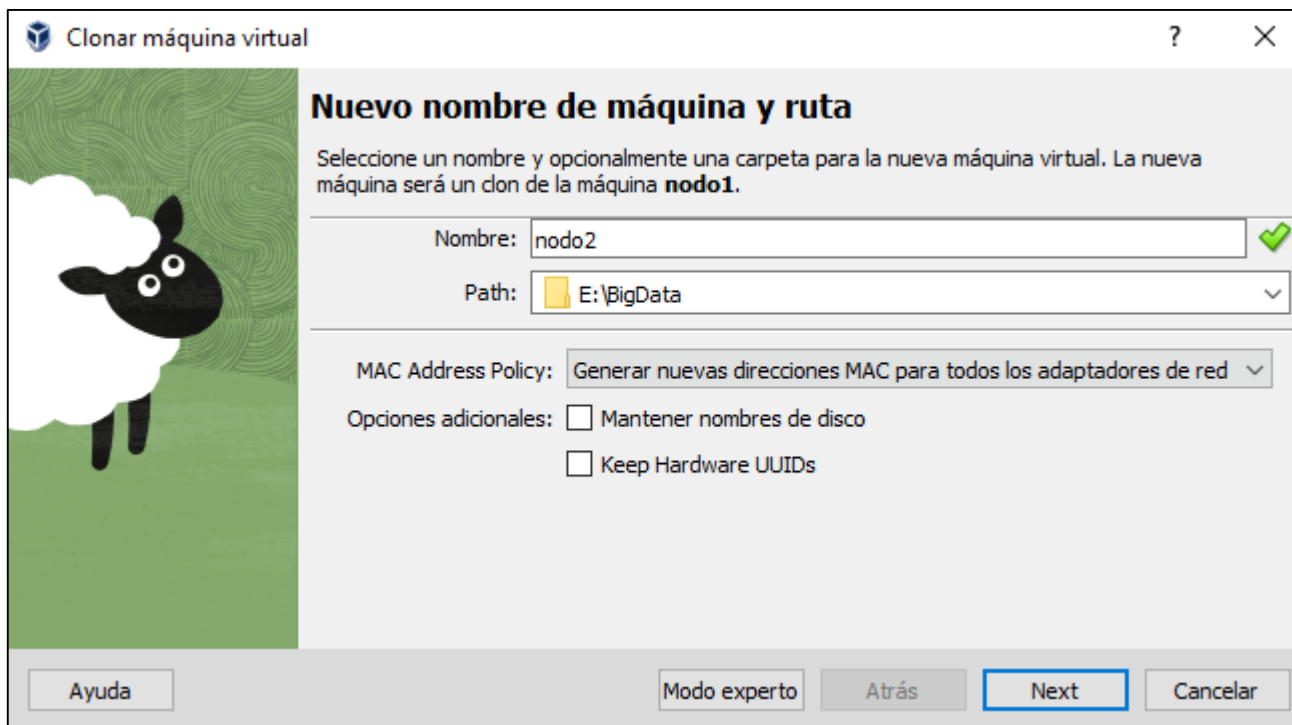
# 1. CLONAR EL NODO HADOOP

**Paso 5.** Para clonar el nodo1 hacemos click botón derecho sobre la máquina virtual y luego hacemos click en la opción clonar



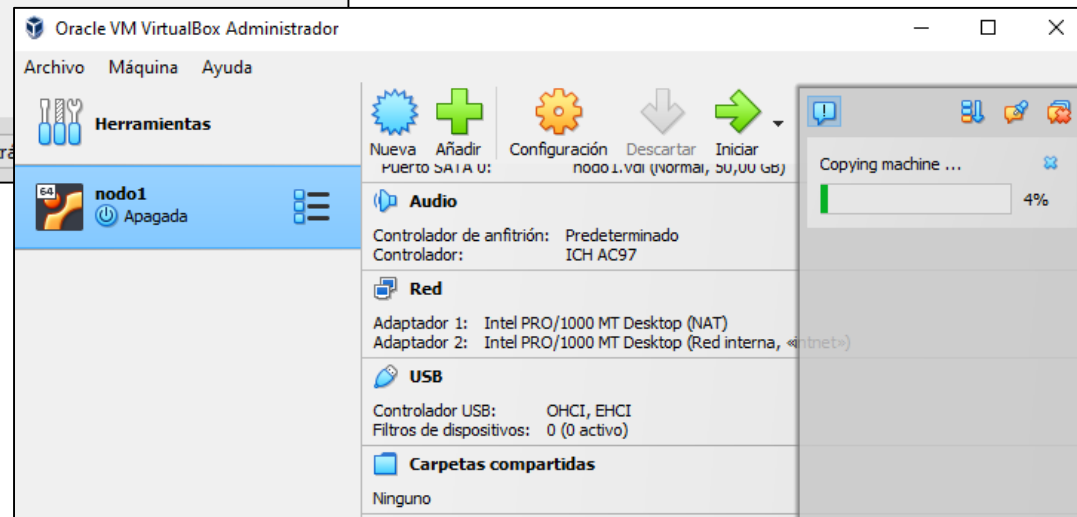
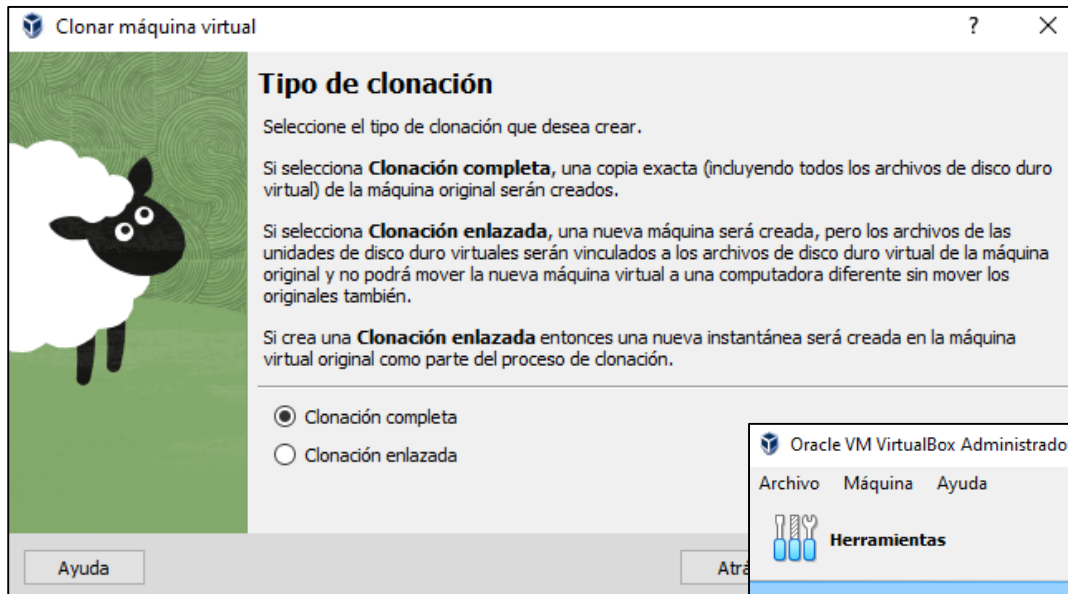
# 1. CLONAR EL NODO HADOOP

**Paso 6.** En el Nombre de la máquina pondremos **nodo2**, en el Path indicaremos su ubicación del disco duro, y en Política de direcciones MAC seleccionaremos que **Genere nuevas direcciones MAC para todos los adaptadores de red** (de lo contrario las @ MAC de las tarjetas serían las mismas e intentaría poner las mismas IPs)



# 1. CLONAR EL NODO HADOOP

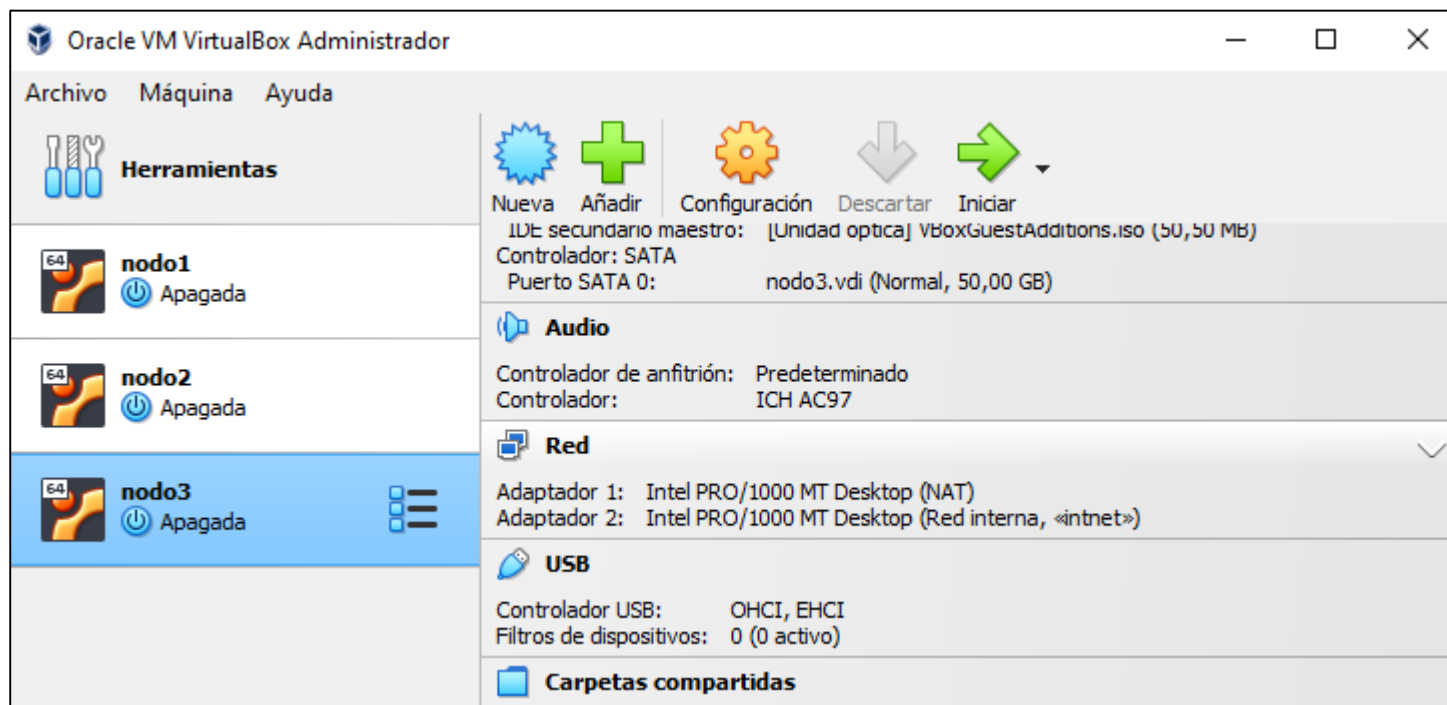
**Paso 7.** Hacemos click en Next y en Tipo de clonación indicamos **Clonación Completa** para que cree una máquina virtual completamente nueva. Le damos a Terminar y empieza a clonar el nodo1 a nodo2





# 1. CLONAR EL NODO HADOOP

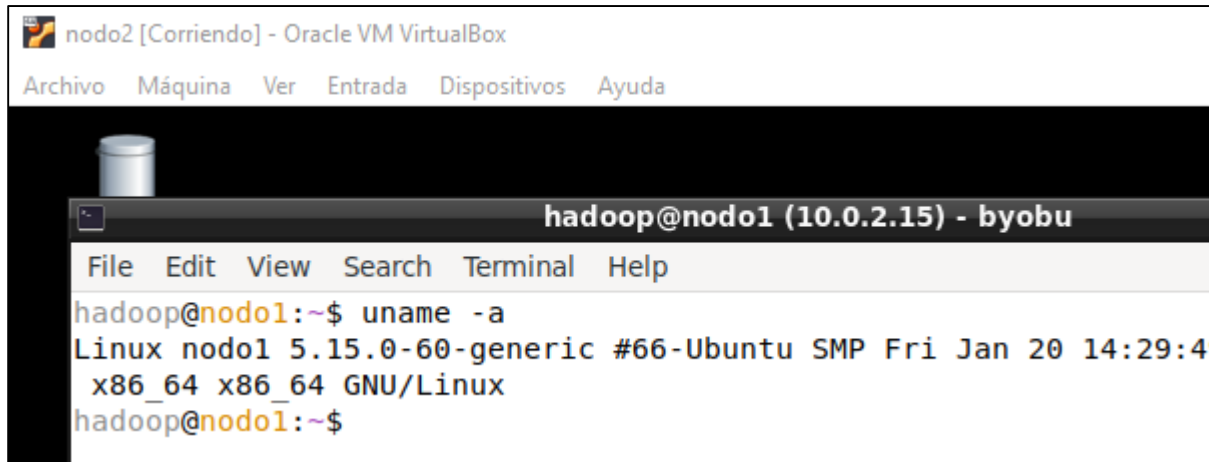
**Paso 8.** Cuando termine tendremos dos máquinas virtuales. Podemos hacer lo mismo para crear el nodo3. Si hay maquina suficiente se pueden crear mas nodos.



## 2. CONFIGURAR LA RED DE LOS NODOS

---

**Paso 1.** Arrancamos el nodo2 clonado. Si hacemos `uname -a` sobre esta maquina clonada vemos que todavía pone `nodo1`



```
nodo2 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

hadoop@nodo1 (10.0.2.15) - byobu
File Edit View Search Terminal Help
hadoop@nodo1:~$ uname -a
Linux nodo1 5.15.0-60-generic #66-Ubuntu SMP Fri Jan 20 14:29:4
x86_64 x86_64 GNU/Linux
hadoop@nodo1:~$
```

**Paso 2.** Se puede cambiar el nombre de la maquina con  
**`sudo hostname nodo2`**

```
hadoop@nodo1:~$ sudo hostname nodo2
[sudo] password for hadoop:
hadoop@nodo1:~$ uname -a
Linux nodo2 5.15.0-60-generic #66-Ubuntu SMP Fri Jan 20 14:29:49 UTC 2023 x86_64
x86_64 x86_64 GNU/Linux
hadoop@nodo1:~$
```

## 2. CONFIGURAR LA RED DE LOS NODOS

---

**Paso 3.** Para que se quede reflejado correctamente es necesario modificar el fichero `/etc/hostname`

```
hadoop@nodo1 (10.0.2.15) - byobu
File Edit View Search Terminal Help
GNU nano 6.2 /etc/hostname
nodo2
```

**NOTA:** Una vez realizado esto se debe de reiniciar la maquina

**Paso 4.** En el fichero `/etc/hosts` donde se definen los nombres de las máquinas, indicaremos las IPs fijas que van a tener cada uno de los nodos. Haremos el siguiente diseño:

```
hadoop@nodo2 (10.0.2.15) - byobu
File Edit View Search Terminal Help
GNU nano 6.2 /etc/hosts *
127.0.0.1 localhost
#127.0.0.1 nodo1
192.168.0.101 nodo1
192.168.0.102 nodo2
192.168.0.103 nodo3
```

Hay que ponerlo en los 3 nodos para que los 3 puedan verse entre sí

## 2. CONFIGURAR LA RED DE LOS NODOS

**Paso 5.** Para ver las interfaces de que dispone el sistema operativo tenemos 2 opciones, mediante los comandos **ip a** o **ifconfig**

```
hadoop@nodo2:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe05:489b prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:05:48:9b txqueuelen 1000 (Ethernet)
    RX packets 15 bytes 2701 (2.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 116 bytes 16292 (16.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::a00:27ff:fe8c:c267 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:8c:c2:67 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 15 bytes 2145 (2.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 58 bytes 5845 (5.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
```

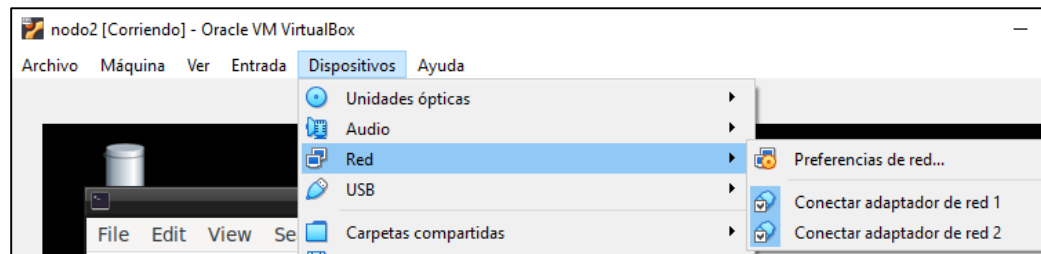
```
hadoop@nodo2:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:05:48:9b brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 85941sec preferred_lft 85941sec
    inet6 fe80::a00:27ff:fe05:489b/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:8c:c2:67 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::a00:27ff:fe8c:c267/64 scope link
        valid_lft forever preferred_lft forever
hadoop@nodo2:~$
```

## 2. CONFIGURAR LA RED DE LOS NODOS

Paso 6. Con ifconfig deberían aparecer las dos tarjetas:

Interfaz	Tipo	
enp0s3	NAT	Para conexión con Internet
enp0s8	Red Interna	para conectarnos con el resto de los nodos.

Si no se viera la interface **enp0s8**, primero comprobar que desde el menú de virtual box Dispositivos/Red se encuentra habilitada



## 2. CONFIGURAR LA RED DE LOS NODOS

---

Paso 7. También se puede probar de habilitar la interficie enp0s8 mediante el comando:

**sudo ifconfig enp0s8 up**

```
hadoop@nodo2:~$ ifconfig
enp0s3: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet6 fe80::a00:27ff:fe05:489b prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:05:48:9b txqueuelen 1000 (Ethernet)
    RX packets 14  bytes 2111 (2.1 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 97  bytes 13074 (13.0 KB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 58  bytes 5845 (5.8 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 58  bytes 5845 (5.8 KB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

hadoop@nodo2:~$ ifconfig enp0s8 up
SIOCSIFFLAGS: Operation not permitted
255 hadoop@nodo2:~$ sudo ifconfig enp0s8 up
```

## 2. CONFIGURAR LA RED DE LOS NODOS

---

**Paso 8.** Una interfaz se puede configurar mediante el comando `ifconfig`, pero esta forma de configuración es volátil, si se reinicia la maquina, la IP se pierde:

**`sudo ifconfig enp0s8 192.168.0.102 netmask 255.255.255.0`**

```
hadoop@nodo2:~$ sudo ifconfig enp0s8 192.168.0.102 netmask 255.255.255.0
hadoop@nodo2:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe05:489b prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:05:48:9b txqueuelen 1000 (Ethernet)
    RX packets 15 bytes 2701 (2.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 142 bytes 20515 (20.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.102 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::a00:27ff:fe8c:c267 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:8c:c2:67 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 108 bytes 18423 (18.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

## 2. CONFIGURAR LA RED DE LOS NODOS

---

**Paso 9.** Para que el nodo2 tenga una IP fija al arrancar, se debe de indicar en un determinado fichero de red. En el fichero yaml del directorio /etc/netplan se configura la IP fija del nodo2.

**sudo nano /etc/netplan/00-installer-config-all.yaml**

```
GNU nano 6.2                                00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
  version: 2
```

```
GNU nano 6.2                                00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: no
      addresses:
        - 192.168.0.102/24
      gateway4: 192.168.0.1
      nameservers:
        addresses: [8.8.8.8, 1.1.1.1]
```



## 2. CONFIGURAR LA RED DE LOS NODOS

---

**Paso 10.** Para que la nueva configuración tome efecto, 2 opciones:

- Reiniciando la maquina.
- Mediante el comando **sudo netplan apply** se aplican las instrucciones de red sin necesidad de reiniciar la maquina

Con `ifconfig` vemos las dos interfaces: la tarjeta de NAT que utilizaremos para conectarnos a Internet y la tarjeta interna que es la que vamos a utilizar para conectarnos entre los distintos nodos

```
hadoop@nodo2:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe05:489b  prefixlen 64  scopeid 0x20<link>
    ether 08:00:27:05:48:9b  txqueuelen 1000  (Ethernet)
    RX packets 5  bytes 1533 (1.5 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 53  bytes 7777 (7.7 KB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.0.102  netmask 255.255.255.0  broadcast 192.168.0.255
    inet6 fe80::a00:27ff:fe8c:c267  prefixlen 64  scopeid 0x20<link>
    ether 08:00:27:8c:c2:67  txqueuelen 1000  (Ethernet)
    RX packets 0  bytes 0 (0.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 48  bytes 6892 (6.8 KB)
```

## 2. CONFIGURAR LA RED DE LOS NODOS

---

**Paso 11.** Abrimos ahora el nodo1 y comprobamos los 3 ficheros:

/etc/hostname → ya esta ok

/etc/hosts

/etc/netplan/00-installer-config.yaml

```
GNU nano 6.2 /etc/hostname
nodo1
```

```
hadoop@nodo1 (10.0.2.15) - byobu
File Edit View Search Terminal Help
GNU nano 6.2 /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  version: 2
  ethernet:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: no
      addresses:
        - 192.168.0.101/24
      gateway4: 192.168.0.1
      nameservers:
        addresses: [8.8.8.8, 1.1.1.1]
```

```
hadoop@nodo1 (10.0.2.15) - byobu
File Edit View Search Terminal Help
GNU nano 6.2 /etc/hosts
127.0.0.1 localhost
#127.0.0.1 nodo1
192.168.0.101 nodo1
192.168.0.102 nodo2
192.168.0.103 nodo3
```

## 2. CONFIGURAR LA RED DE LOS NODOS

**Paso 12.** Inicialmente si hacemos ifconfig la interfaz de Red Interna no sale. Una vez configurados los ficheros, reiniciamos el equipo y ya aparece la interfaz correctamente configurada

```
hadoop@nodo1:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe81:a32d prefixlen 64
    ether 08:00:27:81:a3:2d txqueuelen 1000 (Ethernet)
    RX packets 26531 bytes 38848551 (38.8 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1670 bytes 126475 (126.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 54 bytes 5805 (5.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 54 bytes 5805 (5.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0

hadoop@nodo1:~$
```

```
hadoop@nodo1:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe81:a32d prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:81:a3:2d txqueuelen 1000 (Ethernet)
    RX packets 11 bytes 2038 (2.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 64 bytes 9367 (9.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.101 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::a00:27ff:fefa:3e67 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:fa:3e:67 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 88 bytes 10636 (10.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 79 bytes 8014 (8.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 79 bytes 8014 (8.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

## 2. CONFIGURAR LA RED DE LOS NODOS

---

**Paso 13.** Abrimos ahora el nodo3 y comprobamos los 3 ficheros:

/etc/hostname

/etc/hosts

/etc/netplan/00-installer-config.yaml

```
GNU nano 6.2 /etc/hostname
nodo3
```

```
hadoop@nodo1 (10.0.2.15) - byobu
File Edit View Search Terminal Help
GNU nano 6.2 /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: no
      addresses:
        - 192.168.0.103/24
      gateway4: 192.168.0.1
      nameservers:
        addresses: [8.8.8.8, 1.1.1.1]
```

```
hadoop@nodo1 (10.0.2.15) - byobu
File Edit View Search Terminal Help
GNU nano 6.2 /etc/hosts
127.0.0.1 localhost
#127.0.0.1 nodo1
192.168.0.101 nodo1
192.168.0.102 nodo2
192.168.0.103 nodo3
```

### 3. CONFIGURACION SSH ENTRE NODOS

---

**Paso 1.** Configuraremos la conexión segura entre las 3 máquinas. Debemos abrir los 3 nodos: nodo1, nodo2 y nodo3.

En el nodo1, en el home del usuario hadoop teníamos un directorio .ssh con la clave pública id\_rsa.pub, la clave privada y el fichero authorized\_keys que permitía la autenticación entre máquinas.

```
hadoop@nodo2:~$ cd
hadoop@nodo2:~$ cd .ssh/
hadoop@nodo2:~/.ssh$ ls -l
total 20
-rw----- 1 hadoop hadoop 566 ene  4 13:27 authorized_keys
-rw----- 1 hadoop hadoop 2602 ene  4 13:12 id_rsa
-rw-r--r-- 1 hadoop hadoop 566 ene  4 13:12 id_rsa.pub
-rw----- 1 hadoop hadoop 1262 ene 14 14:39 known_hosts
-rw-r--r-- 1 hadoop hadoop 142 ene  4 13:29 known_hosts.old
hadoop@nodo2:~/.ssh$
```

**Paso 2.** Lo borramos todo porque hay que volver a generarlo

```
hadoop@nodo2:~/.ssh$ rm *
hadoop@nodo2:~/.ssh$ ls
hadoop@nodo2:~/.ssh$ █
```

### 3. CONFIGURACION SSH ENTRE NODOS

---

**Paso 3.** Abrimos una segunda pestaña en el terminal y hacemos un ssh desde el nodo1 contra el nodo2. y vemos que nos va a pedir contraseña (esto es lo que queremos evitar con las claves publicas)

```
hadoop@nodo1:~/.ssh$ ssh nodo1
The authenticity of host 'nodo1 (192.168.0.101)' can't be established.
ED25519 key fingerprint is SHA256:hVU04B988kL7av27mS8wa6TURtu3DKrXPsoWhAowUD8.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'nodo1' (ED25519) to the list of known hosts.
hadoop@nodo1's password:
```

**Paso 4.** Una vez entrada la contraseña, ya vemos el nombre de la maquina nodo2 en el prompt del ssh. Desde dentro de la conexión, vamos al directorio .ssh y borramos todo

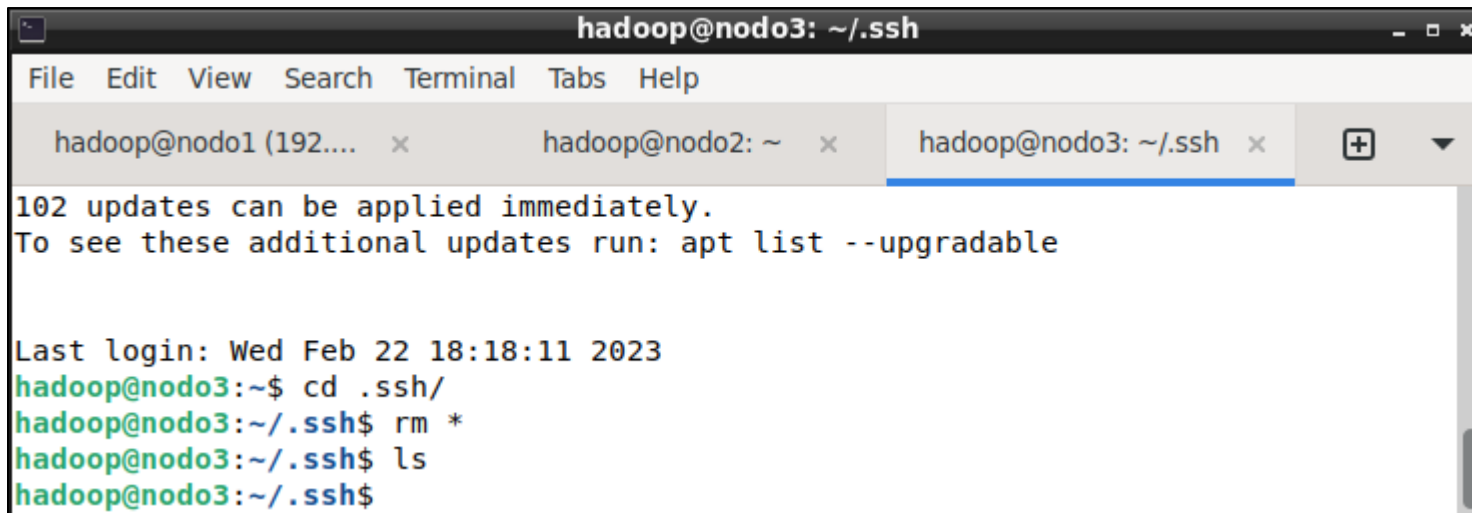
```
Last login: Wed Mar  1 07:34:31 2023 from 192.168.0.101
hadoop@nodo2:~$ pwd
/home/hadoop
hadoop@nodo2:~$ cd .ssh
hadoop@nodo2:~/.ssh$ rm *
hadoop@nodo2:~/.ssh$ ls
hadoop@nodo2:~/.ssh$ exit
logout
Connection to nodo2 closed.
hadoop@nodo1:~/.ssh$ █
```

Si salimos  
volvemos a ver el  
nombre del nodo1  
en el prompt

### 3. CONFIGURACION SSH ENTRE NODOS

---

**Paso 5.** Abrimos una tercera solapa y hacemos un SSH al nodo3. De esta manera accedemos a las 3 maquinas desde las pestañas del terminal del nodo1, sin tener que ir cambiando de una máquina a otra. Ponemos la contraseña y eliminamos el directorio .ssh



```
hadoop@nodo3: ~/.ssh
File Edit View Search Terminal Tabs Help
hadoop@nodo1 (192.... x hadoop@nodo2: ~ x hadoop@nodo3: ~/.ssh x
102 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Last login: Wed Feb 22 18:18:11 2023
hadoop@nodo3:~$ cd .ssh/
hadoop@nodo3:~/ssh$ rm *
hadoop@nodo3:~/ssh$ ls
hadoop@nodo3:~/ssh$
```



### 3. CONFIGURACION SSH ENTRE NODOS

---

**Paso 6.** Volvemos al nodo1 y vamos a volver a generar las clave con el comando `ssh-keygen`. Generamos una clave nueva haciendo enter en las contraseñas pedidas

```
hadoop@nodo1 (192.168.0.101) - byobu
File Edit View Search Terminal Tabs Help
hadoop@nodo1 (192.... x hadoop@nodo2: ~ x hadoop@nodo3: ~/.ssh x
hadoop@nodo1:~/.ssh$ 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:vpAQ2z9lGWetZJftJpbSS5LC1JUjrJH6DwFwM7EuS/0 hadoop@nodo1
The key's randomart image is:
+---[RSA 3072]---+
|      ..=. 0  .. |
|      ..+0.0000 |
|      .   .000*.+..|
|      + 0.00B = 0 |
|      0 + S.*.= * 0|
|      0 * +0. = + |
|      + + Eo  .   |
|      . 0  .     |
|      .           |
+-----[SHA256]-----+
hadoop@nodo1:~/.ssh$
```



### 3. CONFIGURACION SSH ENTRE NODOS

**Paso 7.** Copiamos la clave publica en el fichero `authorized_keys`. Si hacemos un `cat` vemos al final el usuario `hadoop` y el nombre de la máquina `nodo1`

```
hadoop@nodo1:~/.ssh$ ls -l
total 16
-rw----- 1 hadoop hadoop 2602 mar  1 08:47 id_rsa
-rw-r--r-- 1 hadoop hadoop  566 mar  1 08:47 id_rsa.pub
-rw----- 1 hadoop hadoop 1262 mar  1 08:37 known_hosts
-rw-r--r-- 1 hadoop hadoop  142 mar  1 07:34 known_hosts.old
hadoop@nodo1:~/.ssh$ cp id_rsa.pub authorized_keys
hadoop@nodo1:~/.ssh$ ls -l
total 20
-rw-r--r-- 1 hadoop hadoop  566 mar  1 09:52 authorized_keys
-rw----- 1 hadoop hadoop 2602 mar  1 08:47 id_rsa
-rw-r--r-- 1 hadoop hadoop  566 mar  1 08:47 id_rsa.pub
-rw----- 1 hadoop hadoop 1262 mar  1 08:37 known_hosts
-rw-r--r-- 1 hadoop hadoop  142 mar  1 07:34 known_hosts.old
hadoop@nodo1:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGC5gnYGJpHeffgHT+7lgCUYZ+u62QZfb/EZifIlEiqVy6Sv4dm
5Bk9P75xc52IvTFfDINC6+Vklqnajl0n8CAM1bZfC0pQ/A2M0vNx8MTLHTIpKjn07a2sRCm0tTQHQAiolNFY90
G2HH73L1a37mbdMQ0myPBZ9pd1rRHSWB7JKX7LvyikkgoK7mltM1J0YdDog/THmYXG80qFZy/cngVa++KusaDgq
+9qZL0CyQ0L951hH6KpFYte9splvJHsIDyy+P+55bprqK4vM2l+zkz/BqyWqrBydRXnrUchJF6rGNqsaQ9qdLGM
PbBH0BK67VasyLYKFIdrcAl/NLCgP0VaI4ZxBzSIAhqMGNUdLchQzceqgddV1pjaHtOKiL7TQuC52p669uXQ/v3
xA6zTwy43VATR8cUHYj3QzE9bNChh7KF0Uu8vZbYW0TKfIQZ0kGtq35JG8bre0KVjBXuVw793NYi6LsM06/v/uU
5YKXNItxZdwV0SYvrX8YyVuinpgpE= hadoop@nodo1
hadoop@nodo1:~/.ssh$
```

### 3. CONFIGURACION SSH ENTRE NODOS

**Paso 8.** Mediante el comando scp pasaremos `authorized_keys` a las otras maquinas `nodo2` y `nodo3`:

**scp `authorized_keys` `nodo2:/home/hadoop/.ssh`**

```
hadoop@nodo1 (192.168... x  hadoop@nodo2: ~ x  hadoop@nodo3: ~/.ssh x  +
hadoop@nodo1:~/.ssh$ scp authorized_keys nodo2:/home/hadoop/.ssh
hadoop@nodo2's password:
authorized_keys                                100% 566   386.6KB/s   00:00
hadoop@nodo1:~/.ssh$
```

**Paso 9.** Vamos al `nodo2` y dentro de `.ssh` ya tenemos el fichero `authorized_keys` del `nodo1`.

```
hadoop@nodo1 (192.168... x  hadoop@nodo2: ~/.ssh x  hadoop@nodo3: ~/.ssh x  +
hadoop@nodo2:~/.ssh$ ls -l
total 4
-rw-r--r-- 1 hadoop hadoop 566 mar  1 10:01 authorized_keys
hadoop@nodo2:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGC5gnYGJpHeffgHT+7lgCUYZ+u62QZfb/EZifIlEiqVy6Sv4dm
5Bk9P75xc52IvTFfDINC6+Vk1qnaJl0n8CAM1bZfC0pQ/A2M0vNx8MTLHTIpKjn07a2sRCm0tTQHQQAi0lNFY90
G2HH73L1a37mbdMQ0myPBZ9pd1rRHSWb7JKX7LvyikkgoK7mltM1J0YdDog/THmYXG80qFZy/cngVa++KusaDgq
+9qZL0CyQ0L951hH6KpFYte9sp1vJHsIDyy+P+55bprqK4vM2l+zKz/BqyWqrBydRXnrUchJF6rGNqsaQ9qdLgm
PbBH0BKg7VasyLYKFIdrcAl/NLCgPOVaI4ZxBzSIAhqMGNUdLchQzceqgddV1pjaHt0KiL7TQuC52p669uXQ/v3
xA6zTwy43VATR8cUHYj3QzE9bNChh7KF0Uu8vZbYW0TKfIQZ0kGtq35JG8bre0KVjBXuVw793NYi6LsM06/v/uU
5YKXNItxZdwV0SYvrX8YyVuinpge= hadoop@nodo1
hadoop@nodo2:~/.ssh$
```

### 3. CONFIGURACION SSH ENTRE NODOS

---

**Paso 10.** Generamos las claves en el nodo2 con ssh-keygen (pulsamos enter enter)

```
hadoop@nodo1 (192.168... x  hadoop@nodo2: ~/.ssh x  hadoop@nodo3: ~/.ssh x  +
hadoop@nodo2:~/.ssh$ 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:
SHA256:9ACtUNFIEemzinpuIDMRSoUYe6/2jihELD/krdozPYE hadoop@nodo2
The key's randomart image is:
+---[RSA 3072]-----+
|o.o. oB0          |
|o+  . o.o         |
|=..  o .o         |
|++..  +. o        |
|o= o.  oS .       |
|=.E.o .           |
|o+o= o            |
|. +Bo+            |
|=***+o.           |
+----[SHA256]-----+
hadoop@nodo2:~/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
hadoop@nodo2:~/.ssh$
```

### 3. CONFIGURACION SSH ENTRE NODOS

**Paso 11.** Añadimos el id\_rsa.pub acabado de generar en el nodo2 al authorized\_keys copiado del nodo1.

**cat id\_rsa.pub >> authorized\_keys**

```
hadoop@nodo1 (192.168... x  hadoop@nodo2: ~/.ssh x  hadoop@nodo3: ~/.ssh x  +  ▼
hadoop@nodo2:~/.ssh$ cat id_rsa.pub >> authorized_keys
hadoop@nodo2:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQC5gnYGJpHeffgHT+7lgCUYZ+u62QZfb/EZifIIEiqVy6Sv4dm
5Bk9P75xc52IvTFdINC6+Vklqnajl0n8CAM1bZfC0pQ/A2M0vNx8MTLHTIpKjn07a2sRCm0tTQHQQAi0lNFY90
G2HH73L1a37mbdMQ0myPBZ9pd1rRHSWB7JKX7LvyikkgoK7mltM1J0YdDog/THmYXG80qFZy/cngVa++KusaDgq
+9qZL0CyQ0L951hH6KpFYte9sp1vJHsIDyy+P+55bprqK4vM2l+zKz/BqyWqrBydRXnrUchJF6rGNqsaQ9qdLgm
PbBH0BKg7VasyLYKFIdrcAl/NLCgP0VaI4ZxBzSIAhqMGNUdLchQzceggddVlpjaHt0KiL7TQuC52p669uXQ/v3
xA6zTwy43VATR8cUHYj3QzE9bNChh7KF0Uu8vZbYW0TKfIQZ0kGtq35JG8bre0KVjBXuVw793NYi6LsM06/v/uU
5YKXNItxZdwV0SYvrX8YyVuinpgpE= hadoop@nodo1
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQCfovF5ojKb5lJjufXZJc4X0FZBiT8w8BU5kgFqDFm/+ZFNmZx
2S6mQRNlJeRfQ1xLMTR7ui2VX+BRRv9KK2bGBaR6t4c87fjWb5b9VYqslItjPEmrCRc08iGJVRcLN07rc7HAMhm
RHcUVl35MM/MTzorx15V1PFaBPZzrdMQaIpeI8r7W9yyGi7hlGXivrX5c66hLg8M6QRvNq7RWwpLj9h8Fl9BEaj
CluHhZv7D6aN3D+Pf75zBppSBzmZbcWSqKP0Af9hGd9AEZ8EhJTS6MjMKTnbXhmQL2pPp6Qx6xt2i36t4lnhLJR
yj6QXvvwEEky5Xnkla7voXyYGSA4l8sGSfoQMg3/aFlX79cHebuiE45hdKhTi0KnEZayMgNHPiYRwZ0j6yiYvUy
zBcoZ5Ng56//QwVH0d/3x8l0/JdXhJ4LyeIrlx36phN3RuWST9PkhKSUB2feR4MI4XwkZgmj1jXclCiaoemErA8
3Rf86keQYCSEbSVEgXGTZrndQE8V8= hadoop@nodo2
hadoop@nodo2:~/.ssh$
```

authorized\_keys contendrá las credenciales del nodo1 y del nodo2

### 3. CONFIGURACION SSH ENTRE NODOS

---

**Paso 12.** Hacemos el mismo proceso con el nodo3. Ahora desde el nodo2 copiamos el authorized keys al nodo3 mediante scp. Como no hay todavía relación de confianza nos vuelve a pedir la contraseña.

```
hadoop@nodo1 (192.168... x  hadoop@nodo2: ~/.ssh x  hadoop@nodo3: ~/.ssh x  +
hadoop@nodo2:~/.ssh$ scp authorized_keys nodo3:/home/hadoop/.ssh
The authenticity of host 'nodo3 (192.168.0.103)' can't be established.
ED25519 key fingerprint is SHA256:hVU04B988kL7av27mS8wa6TURtu3DKrXPsoWhAowUD8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'nodo3' (ED25519) to the list of known hosts.
hadoop@nodo3's password:
authorized_keys                                100% 1132    705.9KB/s   00:00
hadoop@nodo2:~/.ssh$
```



### 3. CONFIGURACION SSH ENTRE NODOS

**Paso 13.** Vamos a la pestaña del terminal del nodo3 y comprobamos que nos ha llegado el fichero `authorized_keys`. Si hacemos un `cat` vemos que tenemos las claves de `nodo1` y `nodo2`

```
hadoop@nodo1 (192.168... x   hadoop@nodo2: ~/.ssh x   hadoop@nodo3: ~/.ssh x   [+ v]
hadoop@nodo3:~/.ssh$ ls
authorized_keys
hadoop@nodo3:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGC5gnYGJpHeffgHT+7lgCUYZ+u62QZfb/EZifIlEiqVy6Sv4dm
5Bk9P75xc52IvTFdINC6+Vklqnajl0n8CAM1bZfCOpQ/A2M0vNx8MTLHTIpKjn07a2sRCm0tTQHQQAi0lNFY90
G2HH73L1a37mbdMQ0myPBZ9pd1rRHSWB7JKX7LvyikkgoK7mltM1J0YdDog/THmYXG80qFZy/cngVa++KusaDgq
+9qZL0CyQ0L951hH6KpFYte9sp1vJHsIDyy+P+55bprqK4vM2l+zKz/BqyWqrBydRXnrUchJF6rGNqsaQ9qdLGM
PbBH0BKG7VasyLYKFIdrcAl/NLCgP0VaI4ZxBzSIAhqMGNUdLchQzceggddV1pjaHt0KiL7TQuC52p669uXQ/v3
xA6zTwY43VATR8cUHYj3QzE9bNChh7KF0Uu8vZbYW0TKfIQZ0kGtq35JG8bre0KVjBXuVw793NYi6LsM06/v/uU
5YKXNItxZdwV0SYvrX8YyVuinpge= hadoop@nodo1
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGCfovF5ojKb5LjJufXZJc4X0FZBiT8w8BU5kgFqDFm/+ZFNmZx
2S6mQRNLJeRfQ1xLMTR7ui2VX+BRrv9KK2bGBaR6t4c87fjWb5b9VYqslItjPEmrCRc08iGJVRcLN07rc7HAMhm
RHCUVl35MM/MTzox15V1PFaBPZzrdMQaIpeI8r7W9yyGi7hLGXivrX5c66hLg8M6QRvNQ7RWwpLj9h8F19BEaj
CluHhZv7D6aAN3D+Pf75zBppSBzmZbcWSqKP0Af9hGd9AEZ8EhJTS6MjMKTnbXhmQL2pPp6Qx6xt2i36t4lnhLJR
yj6QXvvwEEky5Xnkla7voXyYGSa4l8sGSfoQMg3/aFlX79cHebuiE45hdKhTi0KnEZayMgNHPiYRwZ0j6yiYvUy
zBcoZ5Ng56//QwVH0d/3x8l0/JdXhJ4LyeIrlx36phN3RuwST9PkhKSub2feR4MI4XwkZgmj1jXc1CiaoemErA8
3Rf86keQYCSEbSVEgXGTZrndQE8V8= hadoop@nodo2
hadoop@nodo3:~/.ssh$
```

## 3. CONFIGURACION SSH ENTRE NODOS

**Paso 14.** Generamos las claves en nodo3 mediante ssh-keygen (hacemos enter + enter)

```
hadoop@nodo1 (192.168... x  hadoop@nodo2: ~/.ssh x  hadoop@nodo3: ~/.ssh x  +
hadoop@nodo3:~/.ssh$ 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:x817yhVFQrA+Lv9zI3QZJm1Y9krTpgWFgmPXew8BgXo hadoop@nodo3
The key's randomart image is:
+---[RSA 3072]---+
|                o+**=o|
|               = o.=B |
|              o o..=++|
|             ..E+ oo*=|
|            S.o =.*o|
|           . . +o+ |
|          . + + |
|         + =...|
|        +.oo.|
+-----[SHA256]-----+
hadoop@nodo3:~/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
hadoop@nodo3:~/.ssh$
```

### 3. CONFIGURACION SSH ENTRE NODOS

**Paso 15.** Añadimos el id\_rsa.pub recién generado en el nodo3 al fichero authorized\_keys del nodo2

```
cat id_rsa.pub >> authorized_keys
```

El nodo3 ya tiene el authorized\_keys con los tres nodos

```
hadoop@nodo1 (192.168... x hadoop@nodo2: ~/.ssh x hadoop@nodo3: ~/.ssh x + ▾
hadoop@nodo3:~/.ssh$ cat id_rsa.pub >> authorized_keys
hadoop@nodo3:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQC5gnYGJpHeffgHT+7lgCUYZ+u62QZfb/EZiflEiqVy6Sv4dm
5Bk9P75xc52IvTFfDINC6+Vk1qnajl0n8CAM1bZfC0pQ/A2M0vNx8MTLHTIpKjn07a2sRCm0tTQHQAiolNFY90
G2HH73L1a37mbdMQ0myPBZ9pd1rHRSWB7JKX7LvyikkgoK7mltM1J0YdDog/THmYXG80qFZy/cngVa++KusaDgq
+9qZL0CyQ0L951hH6KpFYte9sp1vJHsIDyy+P+55bprqK4vM2l+zkz/BqyWqrBydRXnrUchJF6rGNqsaQ9qdLGm
PbBH0BKG7VasyLYKFIdrcAl/NLCgP0VaI4ZxBzSIAhQMGNUdLchQzceqgddV1pjaHt0KiL7TQuC52p669uXQ/v3
xA6zTwy43VATR8cUHYj3QzE9bNChh7KF0Uu8vZbYW0TKfIQZ0kGtq35JG8bre0KVjBXuVw793NYi6LsM06/v/uU
5YKXNItxZdwV0SYvrX8YyVuinpgpE= hadoop@nodo1
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQCfovF5ojKb5LjufXZJc4X0FZBiT8w8BU5kgFqDFm/+ZFNmZx
2S6mQRNLJJeRfQ1xLMTR7ui2VX+BRRv9KK2bGBaR6t4c87fjWb5b9VYqslItjPEmrCRc08iGJVRcLN07rc7HAMhm
RHcUVl35MM/MTzorx15V1PFaBPZzrdMQaIpeI8r7W9yyGi7hlgXivR5c66hLg8M6QRvNQ7RWwpLj9h8F19BEaj
CluHhZv7D6aN3D+Pf75zBppSBzmZbcWsQkP0Af9hGd9AEZ8EhJTS6MjMKTnbXhmQL2pPp6Qx6xt2i36t4lnhLJR
yj6QXvvwEEky5XnklA7voXyYGSa4l8sGSfoQMg3/aFLX79cHebuiE45hdKhTi0KnEZayMgNHPiYRwZ0j6yiYvUy
zBcoZ5Ng56//QwVH0d/3x8l0/JdXhJ4LyeIrlx36phN3RuWST9PkhKSUB2fer4MI4XwkZgmj1jXc1CiaoemErA8
3Rf86keQYCEbSVegXGTZrndQE8V8= hadoop@nodo2
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQDo7kRneJ4h3NVKkvQuPZbbIKBz0zoMnQAlQ+NtxdnPSx7Uq5u
D4Hx1U9TIGp42gnmMzxrBtkhMU6lQS9qUdQYIiY6I3KTDRDvYBjBkckzgLjE60gar5ZjfmkrUh7NvjehdgIdIcH
b6p5CaGm7k+S96cPjMefB/Bj7nN0qS0fPxZGFWci00FW5CgNam5LNHBVrRg5af+sIiYbLnu8Q9lv6iXxZi+Q1WR
kUwCmtrNDAB8nlqu2xQGKsiNyMn+jxRaday/7XiGnU52oTZKJaeoHjDLjukYzVn3Fsn0Ax2486o8zx73iGM9zY
+C+mDIa7pkyneYUhlEMZb7N0kRnQeyWv+TedxRQiz0Zj3v+mLyhXucNvvDh8g8AYDPtw+QzKxWImv3C1IoM4tNp
EjWloZgtBTcjV6ZHjtXkzUqj77w7hQLG0oqeYsZW8powbqMN7rAwJx/BmVIwtquooofKnkPmMNlu2Vg5aLXSW57w
Ib1fDRf01ApEM/3QoDHCF2nrY2dAs= hadoop@nodo3
hadoop@nodo3:~/.ssh$
```



### 3. CONFIGURACION SSH ENTRE NODOS

---

**Paso 16.** Por ultimo pasaremos el fichero `authorized_keys` del nodo3 con las claves de los 3 nodos, al nodo1 y nodo 2.

```
hadoop@nodo1 (192.168... x   hadoop@nodo2: ~/.ssh x   hadoop@nodo3: ~/.ssh x   [+]  
hadoop@nodo3:~/.ssh$  
hadoop@nodo3:~/.ssh$ scp authorized_keys nodo2:/home/hadoop/.ssh  
hadoop@nodo2's password:  
authorized_keys                                100% 1698    828.5KB/s   00:00  
hadoop@nodo3:~/.ssh$ scp authorized_keys nodo1:/home/hadoop/.ssh  
hadoop@nodo1's password:  
authorized_keys                                100% 1698    1.1MB/s    00:00  
hadoop@nodo3:~/.ssh$
```

# 3. CONFIGURACION SSH ENTRE NODOS

**Paso 17.** Si visualizamos `authorized_keys` en `nodo1` y en `nodo2`, vemos que ya tenemos los 3 nodos dentro de nuestro Cluster.

hadoop@nodo1 (192.168... x
hadoop@nodo2: ~/.ssh x
hadoop@nodo3: ~/.ssh x
+

```

hadoop@nodo2:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGC5gnYGJpHeffgHT+7lgCUYZ+u62QZfb/EZifIleiqVy6Sv4dm
5Bk9P75xc52IvTFdINC6+VklqnaJl0n8CAMLbZfC0pQ/A2M0vNx8MTLHTIpKjn07a2sRCm0tTQHQAiolNFY90
G2HH73L1a37mbdMQ0myPBZ9pd1rRHSWB7JKX7LvyikkgoK7mltM1J0YdDog/THmYXG80qFZy/cngVa++KusaDgq
+9qZL0CyQ0L951hH6KpFYte9sp1vJHsIDyy+P+55bprqK4vM2l+zKz/BqyWqrBydRXnrUchJF6rGNqsaQ9qdLGM
PbBH0BK6G7VasyLYKFIdrcAl/NLCgPOVaI4Zx8zSIAhqMGNUdLchQzceggddV1pjaHt0KiL7TQuC52p669uXQ/v3
xA6zTwY43VATR8cUHYj3QzE9bNChh7KF0Uu8vZbYW0TKFIQZ0Kgtq35JG8bre0KVjBXuVw793NYi6LsM06/v/uU
5YKXNItxZdwV0SYvrx8YyVuinpgpE= hadoop@nodo1
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGCfovF5ojKb5LJjufXZJc4X0FZBiT8w8BU5kgFqDFm/+ZFNmZx
2S6mQRNLJeRfQ1xLMTR7ui2VX+BRrv9KK2bGBaR6t4c87fjWb5b9
RHCUVl35MM/MTz0rx15V1PFaBPZzrdMQaIpeI8r7W9yyGi7hLGXi
CluHhZv7D6aN3D+Pf75zBppSBzmZbcWSqKPOAf9hGd9AEZ8EhJTS
y6QXvvwEEky5Xnkla7voXyYGSa4l8sGSfoQMg3/aFLX79cHebui
zBcoZ5Ng56//QwVH0d/3x8l0/JdXhJ4LyeIrlx36phN3RuWST9PK
3Rf86keQYCSEbSVEgXGTZrndQE8V8= hadoop@nodo2
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGCDo7kRneJ4h3NVK
D4Hx1U9TIGp42gnmMzxrBtkhMU6lQS9qUdQYIiY6I3KTDrdvYBjB
b6p5CaGM7k+S96cPjMefB/Bj7nN0qS0fPXzGFwci00FW5CgNam5L
kUwCmtrNDAB8nlqu2xQGKsiNyMn+jxRaday/7XiGnU52oTqZKjae
+C+mDIa7pkynYUhlEMZb7N0kRnQeyWv+TEdXRQiz0Zj3v+mLyhX
EjWloZgtBTcjV6ZhjtXkzUqj77w7hQlG0oqeYsZW8powbqMN7rAw
Ib1fDRf01ApEM/3QoDHCF2nrY2dAs= hadoop@nodo3
hadoop@nodo2:~/.ssh$

```

hadoop@nodo1 (192.168... x
hadoop@nodo2: ~/.ssh x
hadoop@nodo3: ~/.ssh x
+

```

hadoop@nodo1:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGC5gnYGJpHeffgHT+7lgCUYZ+u62QZfb/EZifIleiqVy6Sv4dm
5Bk9P75xc52IvTFdINC6+VklqnaJl0n8CAMLbZfC0pQ/A2M0vNx8MTLHTIpKjn07a2sRCm0tTQHQAiolNFY90
G2HH73L1a37mbdMQ0myPBZ9pd1rRHSWB7JKX7LvyikkgoK7mltM1J0YdDog/THmYXG80qFZy/cngVa++KusaDgq
+9qZL0CyQ0L951hH6KpFYte9sp1vJHsIDyy+P+55bprqK4vM2l+zKz/BqyWqrBydRXnrUchJF6rGNqsaQ9qdLGM
PbBH0BK6G7VasyLYKFIdrcAl/NLCgPOVaI4Zx8zSIAhqMGNUdLchQzceggddV1pjaHt0KiL7TQuC52p669uXQ/v3
xA6zTwY43VATR8cUHYj3QzE9bNChh7KF0Uu8vZbYW0TKFIQZ0Kgtq35JG8bre0KVjBXuVw793NYi6LsM06/v/uU
5YKXNItxZdwV0SYvrx8YyVuinpgpE= hadoop@nodo1
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGCfovF5ojKb5LJjufXZJc4X0FZBiT8w8BU5kgFqDFm/+ZFNmZx
2S6mQRNLJeRfQ1xLMTR7ui2VX+BRrv9KK2bGBaR6t4c87fjWb5b9VYqsLItjPEmrCRc08iGJVRcLN07rc7HAMhm
RHCUVl35MM/MTz0rx15V1PFaBPZzrdMQaIpeI8r7W9yyGi7hLGXivrX5c66hLg8M6QRvN07RwWpLj9h8FL9BEaj
CluHhZv7D6aN3D+Pf75zBppSBzmZbcWSqKPOAf9hGd9AEZ8EhJTS6MjMKTnbXhmQl2pPp6Qx6xt2i36t4lnhLJR
y6QXvvwEEky5Xnkla7voXyYGSa4l8sGSfoQMg3/aFLX79cHebuiE45hdKhTi0KnEZayMgNHPiYrWZ0j6yiYvUy
zBcoZ5Ng56//QwVH0d/3x8l0/JdXhJ4LyeIrlx36phN3RuWST9PKhKSub2fer4MI4XwkZgmj1jXc1CiaoemErA8
3Rf86keQYCSEbSVEgXGTZrndQE8V8= hadoop@nodo2
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGCDo7kRneJ4h3NVKkvQuPZbbIKBz0zoMnQALQ+NtxdnPSx7Uq5u
D4Hx1U9TIGp42gnmMzxrBtkhMU6lQS9qUdQYIiY6I3KTDrdvYBjBkckzgLjE60gaR5ZjfmkrUh7NvjehdgIdIch
b6p5CaGM7k+S96cPjMefB/Bj7nN0qS0fPXzGFwci00FW5CgNam5LNHBVrRg5af+sIiYbLnu8Q9lv6iXxZi+Q1WR
kUwCmtrNDAB8nlqu2xQGKsiNyMn+jxRaday/7XiGnU52oTqZKjaeHjDLjYkYzVn3Fsn0Ax2486o8zx73iGM9zY
+C+mDIa7pkynYUhlEMZb7N0kRnQeyWv+TEdXRQiz0Zj3v+mLyhXUCNVvDh8g8AYDPtw+QzKxWImv3C1IoM4tNp
EjWloZgtBTcjV6ZhjtXkzUqj77w7hQlG0oqeYsZW8powbqMN7rAwJx/BmVIwtquoofKnKpMmNLu2Vg5aLXSW57W
Ib1fDRf01ApEM/3QoDHCF2nrY2dAs= hadoop@nodo3
hadoop@nodo1:~/.ssh$

```

### 3. CONFIGURACION SSH ENTRE NODOS

**Paso 18.** Para comprobar que ssh funciona sencillamente hacemos desde nodo1 ssh al resto de los nodos. No nos debe pedir password y debe entrar directamente.

```
hadoop@nodo1:~$ ssh nodo2
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-60-generic x86_64)

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

System information as of mié 01 mar 2023 11:38:16 UTC

System load: 0.13623046875      Processes:              175
Usage of /:  60.1% of 23.45GB   Users logged in:       1
Memory usage: 13%              IPv4 address for enp0s3: 10.0.2.15
Swap usage:  0%                IPv4 address for enp0s8: 192.168.0.103

102 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Wed Mar  1 08:04:27 2023 from 192.168.0.101
hadoop@nodo2:~$

hadoop@nodo2:~$ exit
logout
Connection to nodo2 closed.
hadoop@nodo1:~$ ssh nodo3
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-60-generic x86_64)

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

System information as of mié 01 mar 2023 11:38:16 UTC

System load: 0.0                Processes:              175
Usage of /:  60.2% of 23.45GB   Users logged in:       1
Memory usage: 11%              IPv4 address for enp0s3: 10.0.2.15
Swap usage:  0%                IPv4 address for enp0s8: 192.168.0.103

102 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Wed Mar  1 08:37:17 2023 from 192.168.0.101
hadoop@nodo3:~$
```

### 3. CONFIGURACION SSH ENTRE NODOS

---

**Paso 19.** Si hubiera algún problema con el protocolo SSH puede deberse a los permisos de Linux. Se debería comprobar que el fichero `authorized keys` solo tiene Lectura y escritura para el usuario porque de lo contrario el proceso SSH no funciona correctamente. Poniendo el siguiente comando en los 3 nodos, se solventaría ese posible problema:

**`chmod 600 authorized_keys`**

```
hadoop@nodo2:~/.ssh$ ls -l
total 20
-rw-r--r-- 1 hadoop hadoop 1698 mar  1 11:23 authorized_keys
-rw----- 1 hadoop hadoop 2602 mar  1 10:07 id_rsa
-rw-r--r-- 1 hadoop hadoop  566 mar  1 10:07 id_rsa.pub
-rw----- 1 hadoop hadoop  978 mar  1 11:01 known_hosts
-rw-r--r-- 1 hadoop hadoop  142 mar  1 11:01 known_hosts.old
hadoop@nodo2:~/.ssh$
```

## 4. MODIFICAR FICHEROS DEL CLUSTER

---

**Paso 1.** Antes de arrancar nuestro cluster debemos de reconfigurar los ficheros de configuración ubicados en `/opt/hadoop/etc/hadoop`.

### Requisitos necesarios

- a) Tenemos que tener el mismo usuario en todos los nodos del clúster (en este caso no hay ningún problema porque los hemos clonados)
- b) Que sean accesibles todos los nodos a través de SSH sin contraseña. Es la forma en que los distintos nodos de hadoop se entienden
- c) Lo lógico es tener los mismos directorios de Hadoop en cada nodo (toda la información del cluster en los mismos directorios)
- d) Haber copiado el software de hadoop en todos los nodos y en el mismo sitio. Esto hace que tengamos máquinas clónicas y que puedan ser sustituidas de manera muy rápida entre ellas

## 4. MODIFICAR FICHEROS DEL CLUSTER

---

- e) Haber creado el mismo directorio de datos en todos los nodos y con los mismos permisos (en nuestro caso /datos/datanode). Esto evita problemas de permisos, de configuración
- f) Es decir los servidores tienen que ser al final idénticos, clones a nivel de hadoop para poderle sacar el mayor rendimiento a los recursos de hadoop. Y no sólo a nivel de software sino a nivel de hardware

## 4. MODIFICAR FICHEROS DEL CLUSTER

---

**Paso 2.** En nodo2 borraremos el directorio namenode y vaciaremos el contenido de datanode. Para ellos nos conectamos al nodo2 desde nodo1 mediante ssh

```
hadoop@nodo1 (192.168.0.101) - byobu
File Edit View Search Terminal Help

hadoop@nodo1:~$ ssh nodo2
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-60-generic x86_64)
```

- En el directorio /datos eliminamos el directorio namenode, el cual solo debe de estar en el maestro.
- También eliminamos el directorio current de datanode, el cual viene de la maquina clonada. Volveremos a generar todos estos procesos

```
hadoop@nodo2:~$ cd /datos
hadoop@nodo2:/datos$ ls
datanode  namenode
hadoop@nodo2:/datos$ rm -r namenode
hadoop@nodo2:/datos$ ls
datanode
hadoop@nodo2:/datos$ cd datanode/
hadoop@nodo2:/datos/datanode$ ls
current
hadoop@nodo2:/datos/datanode$ rm -r current
hadoop@nodo2:/datos/datanode$
```



## 4. MODIFICAR FICHEROS DEL CLUSTER

---

**Paso 3.** Hacemos igual en el nodo3. En /datos eliminamos el directorio namenode (sólo debe de estar en el maestro) y tambien eliminamos el contenido del directorio datanode

```
hadoop@nodo2:~$ exit
logout
Connection to nodo2 closed.
hadoop@nodo1:~$ ssh nodo3
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-60-generic x86_64)
```

```
hadoop@nodo3:~$ cd /datos
hadoop@nodo3:/datos$ ls
datanode  namenode
hadoop@nodo3:/datos$ rm -r namenode/
hadoop@nodo3:/datos$ cd datanode/
hadoop@nodo3:/datos/datanode$ ls
current
hadoop@nodo3:/datos/datanode$ rm -r current/
hadoop@nodo3:/datos/datanode$
```



## 4. MODIFICAR FICHEROS DEL CLUSTER

---

**Paso 4.** En el nodo1 hacemos lo contrario que en nodo2 y nodo3.

- Nos cargamos el directorio de datanode, ya que en el maestro no tendremos datos, solo metadatos.
- También podemos eliminar el contenido de namenode. Pero se puede obviar ya que tenemos que volver a formatear el sistema de ficheros de nuevo y este proceso también lo borra

```
hadoop@nodo3:/datos/datanode$ exit
logout
Connection to nodo3 closed.
hadoop@nodo1:~$ cd /datos
hadoop@nodo1:/datos$ rm -r datanode/
hadoop@nodo1:/datos$ ls
namenode
hadoop@nodo1:/datos$ cd namenode/
hadoop@nodo1:/datos/namenode$ ls
current
hadoop@nodo1:/datos/namenode$ rm -r current/
hadoop@nodo1:/datos/namenode$
```

### RESUMEN:

/datos/namenode → Debe de estar en el maestro

/datos/datanode → debe de estar en cada uno de los nodos

## 4. MODIFICAR FICHEROS DEL CLUSTER

---

**Paso 5.** Ahora debemos reconfigurar los ficheros de configuración ubicados en el directorio `/opt/hadoop/etc/hadoop`

El primer fichero **`core-site.xml`** no se tiene que tocar, ya que el maestro de hdfs sigue siendo `nodo1`.

```
GNU nano 6.2                                core-site.xml

<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://nodo1:9000</value>
  </property>
</configuration>
```

## 4. MODIFICAR FICHEROS DEL CLUSTER

---

**Paso 6.** El segundo fichero `hdfs-site.xml` se debe de modificar. El parámetro `dfs.replication` pasa a valer 2, ya que pasamos a tener dos esclavos (el numero de bloques que queremos que replique en nuestro sistema). El directorio para el maestro y para los datos queda igual.

```
GNU nano 6.2                                hdfs-site.xml
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>/datos/namenode</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>/datos/datanode</value>
  </property>
</configuration>
```

## 4. MODIFICAR FICHEROS DEL CLUSTER

---

**Paso 7.** Copiaremos `hdfs-site.xml` en el `nodo2` y en el `nodo3`, para que sea el mismo en todos los nodos. Usaremos el comando `scp`:

```
hadoop@nodo1:/opt/hadoop/etc/hadoop$ scp hdfs-site.xml nodo2:/opt/hadoop/etc/hadoop/  
hdfs-site.xml                                100% 1048    54.9KB/s   00:00  
hadoop@nodo1:/opt/hadoop/etc/hadoop$ scp hdfs-site.xml nodo3:/opt/hadoop/etc/hadoop/  
hdfs-site.xml                                100% 1048    58.1KB/s   00:00  
hadoop@nodo1:/opt/hadoop/etc/hadoop$
```

**Paso 8.** El tercer fichero `mapred-site.xml` no hay que tocarlo, el framework de mapreduce continua siendo yarn

```
GNU nano 6.2                                mapred-site.xml *  
  
<configuration>  
  <property>  
    <name>mapreduce.framework.name</name>  
    <value>yarn</value>  
  </property>  
</configuration>
```

## 4. MODIFICAR FICHEROS DEL CLUSTER

---

**Paso 9.** El cuarto fichero yarn-site.xml lo dejamos también igual porque el maestro de procesos va a seguir siendo nodo1

```
GNU nano 6.2 /opt/hadoop/etc/hadoop/yarn-site.xml
<configuration>

<!-- Site specific YARN configuration properties -->
    <property>
        <name>yarn.resourcemanager.hostname</name>
        <value>nodo1</value>
    </property>
    <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.application.classpath</name>
        <value>/opt/hadoop/etc/hadoop:/opt/hadoop/share/hadoop/common/lib/*:/o>
    </property>
</configuration>
```

## 4. MODIFICAR FICHEROS DEL CLUSTER

---

**Paso 10.** Por ultimo existe un fichero donde debemos poner cuáles son los nodos esclavos: nodo2 y nodo3

- En la versión 2 → slaves
- En la versión 3 → workers

De esta manera no arrancará procesos de nodos esclavos en el nodo1.

```
GNU nano 6.2 workers
nodo2
nodo3
```

## 5. ARRANCAR EL CLUSTER

**Paso 1.** Antes de arrancar el cluster, tenemos que formatear el namenode

**hdfs namenode -format**

```
hadoop@nodo1:~$ hdfs namenode -format
```

```
2023-03-03 21:02:58,227 INFO util.GSet: 0.029999999329447746% max memory 873 MB = 268.2 KB
2023-03-03 21:02:58,227 INFO util.GSet: capacity      = 2^15 = 32768 entries
Re-format filesystem in Storage Directory root= /datos/namenode; location= null ? (Y or N) Y
```

Si no hemos vaciado el directorio namenode, nos pregunta si queremos borrar el directorio current existente. El formateo nos deja el directorio namenode limpio, con un directorio current nuevo.

```
en meet shutdown.
2023-03-03 21:04:11,680 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at nodo1/192.168.0.101
*****/
hadoop@nodo1:~$ ls /datos/namenode
current
hadoop@nodo1:~$
```



## 5. ARRANCAR EL CLUSTER

---

Paso 2. Arrancamos el nuevo sistema de ficheros HDFS

**start-dfs.sh**

```
hadoop@nodo1:/opt/hadoop/etc/hadoop$ start-dfs.sh
Starting namenodes on [nodo1]
Starting datanodes
Starting secondary namenodes [nodo1]
hadoop@nodo1:/opt/hadoop/etc/hadoop$ jps
224851 NameNode
225282 SecondaryNameNode
225666 Jps
hadoop@nodo1:/opt/hadoop/etc/hadoop$
```

Arranca:

- el namenode (en el nodo1 el maestro)
- los datanodes (teóricamente en los nodos 2 y 3 que hemos puesto en el fichero workers)
- el Secondary namenode que también arranca en el nodo1.

Pero si hacemos un jps en el maestro vemos que sólo tenemos el namenode y el secondarynamenode, el datanode no está aquí

## 5. ARRANCAR EL CLUSTER

---

**Paso 3.** Abrimos una solapa y nos vamos con ssh al nodo2 y con jps tenemos el datanode.

```
hadoop@nodo1 (192.168.0.101) - byobu x  hadoop@nodo2: ~  
hadoop@nodo2:~$ jps  
251907 Jps  
238926 DataNode  
hadoop@nodo2:~$
```

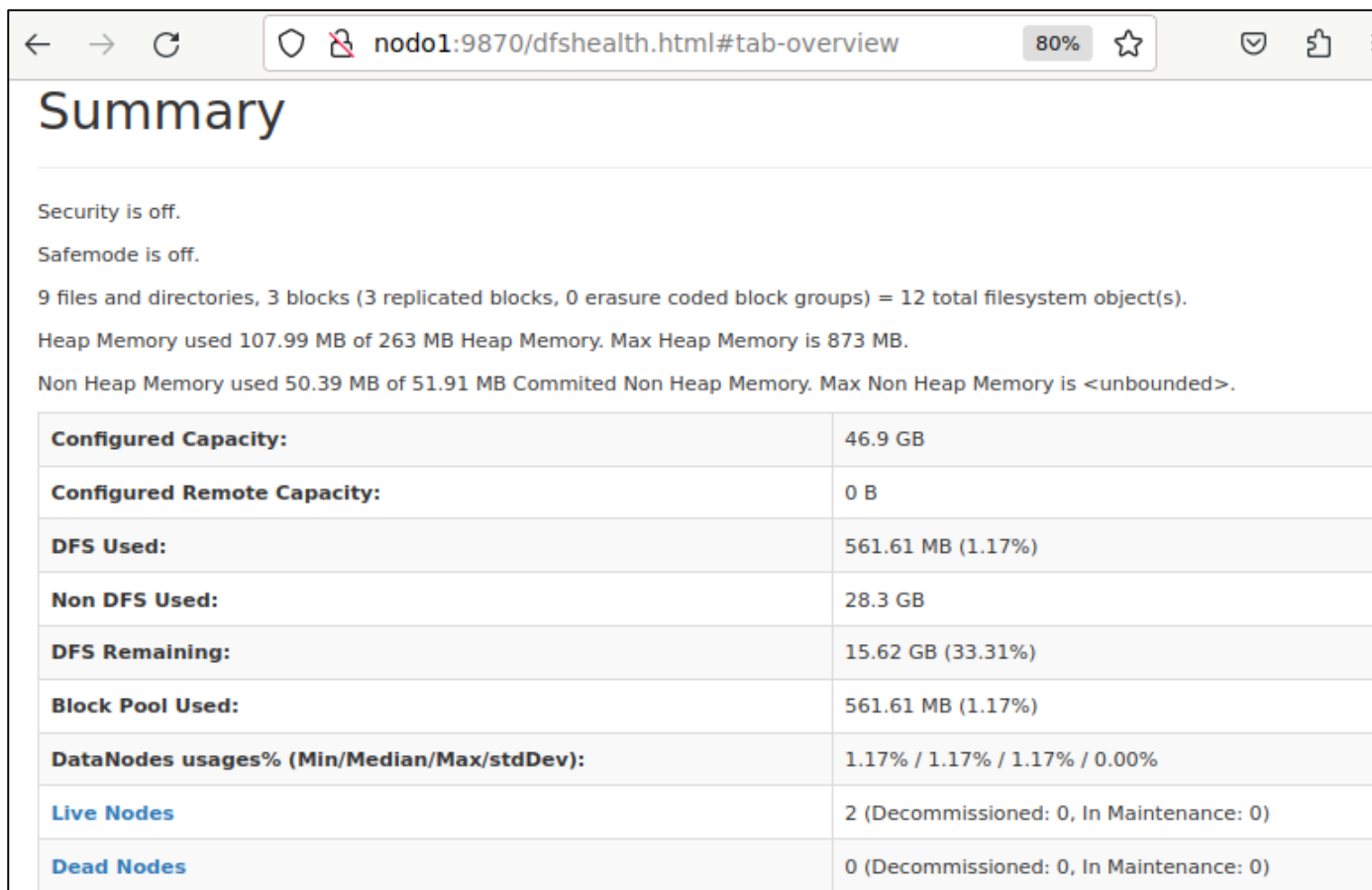
**Paso 4.** Ahora abrimos una tercera solapa y nos vamos al nodo3. Mediante jps tenemos al datanode.

```
hadoop@nodo1 (192.168... x  hadoop@nodo2: ~ x  hadoop@nodo3: ~ x  
hadoop@nodo3:~$ jps  
3848 Jps  
3708 DataNode  
hadoop@nodo3:~$
```

**Resumen:** el namenode sólo está en el maestro  
los datanodes sólo están en los esclavos

## 5. ARRANCAR EL CLUSTER

**Paso 5.** Entramos en la web de administración hadoop mediante nodo1:50070 (v2) o nodo1:9870 (v3). En Summary vemos que tenemos 2 nodos activos, que son los nodos esclavos (no esta el maestro)

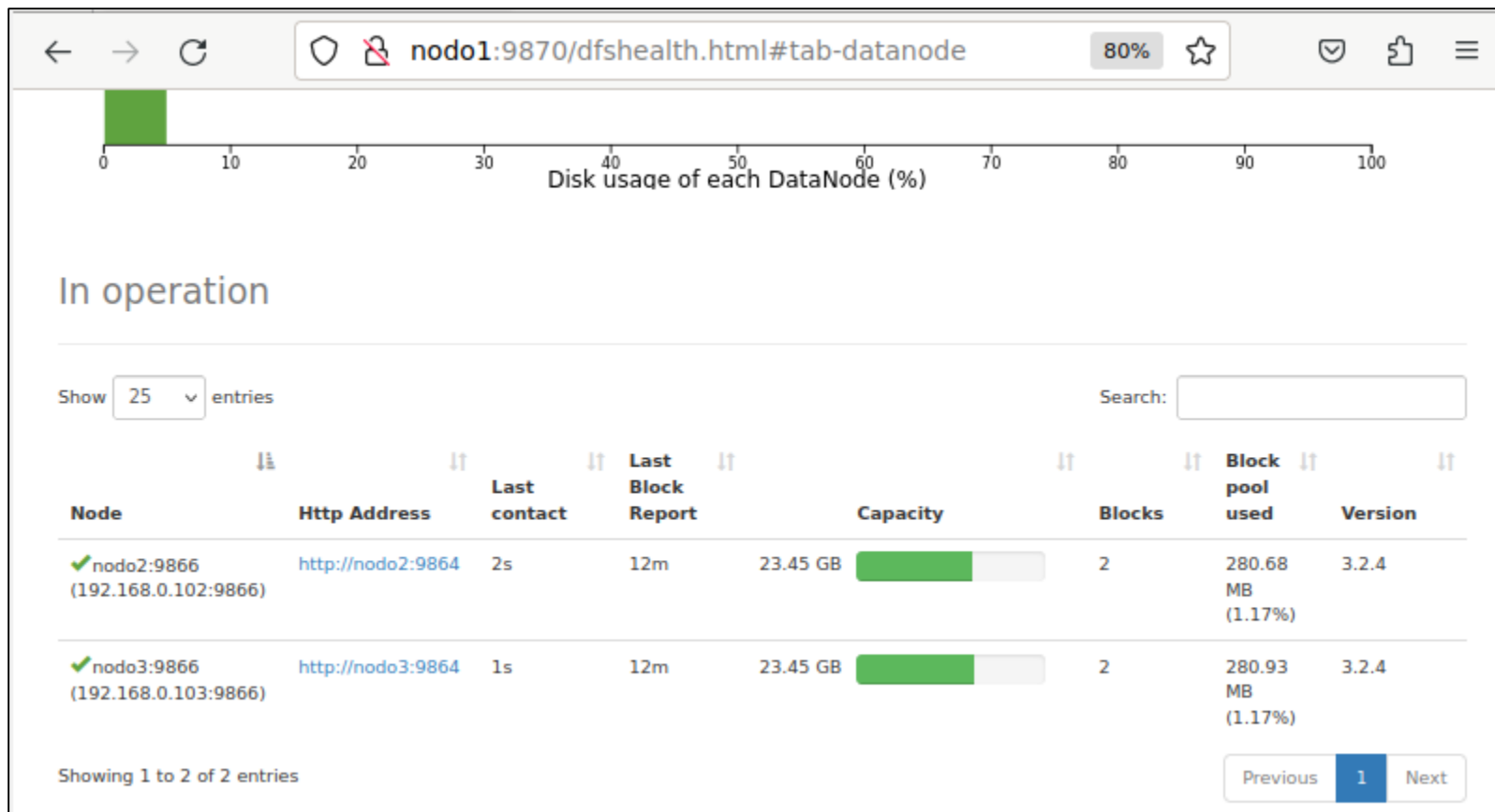


The screenshot shows a web browser window with the address bar displaying 'nodo1:9870/dfshealth.html#tab-overview'. The page title is 'Summary'. Below the title, there are several status messages: 'Security is off.', 'Safemode is off.', '9 files and directories, 3 blocks (3 replicated blocks, 0 erasure coded block groups) = 12 total filesystem object(s).', 'Heap Memory used 107.99 MB of 263 MB Heap Memory. Max Heap Memory is 873 MB.', and 'Non Heap Memory used 50.39 MB of 51.91 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.' Below these messages is a table with various metrics.

<b>Configured Capacity:</b>	46.9 GB
<b>Configured Remote Capacity:</b>	0 B
<b>DFS Used:</b>	561.61 MB (1.17%)
<b>Non DFS Used:</b>	28.3 GB
<b>DFS Remaining:</b>	15.62 GB (33.31%)
<b>Block Pool Used:</b>	561.61 MB (1.17%)
<b>DataNodes usages% (Min/Median/Max/stdDev):</b>	1.17% / 1.17% / 1.17% / 0.00%
<b>Live Nodes</b>	2 (Decommissioned: 0, In Maintenance: 0)
<b>Dead Nodes</b>	0 (Decommissioned: 0, In Maintenance: 0)

## 5. ARRANCAR EL CLUSTER

**Paso 6.** Si vamos a la pestaña Datanodes/Datanode Information vemos los nodos 1 y el 2. No se ve el maestro ya que no figura como nodo esclavo.



## 5. ARRANCAR EL CLUSTER

**Paso 7.** Si hacemos un Browse vemos que no tenemos absolutamente nada, ya que hemos hecho un format del sistema

### Browse Directory

Show

25

▼

entries

Search:

☐

⌵

Permission ⌵

Owner ⌵

Group ⌵

Size ⌵

Last Modified ⌵

Replication ⌵

Block Size ⌵

Name ⌵

No data available in table

Showing 0 to 0 of 0 entries

Previous

Next

## 5. ARRANCAR EL CLUSTER

---

Paso 8. Hacemos un **start-yarn.sh** para arrancar los procesos

```
hadoop@nodo1:~$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
hadoop@nodo1:~$ jps
37796 Jps
10889 ResourceManager
5530 SecondaryNameNode
4959 NameNode
hadoop@nodo1:~$ █
```

Arranca el resourcemanager en el nodo1 (el proceso maestro de esclavos) y los nodemanagers, los procesos en cada uno de los esclavos. Si hacemos jps en el maestro vemos los procesos iniciados en el nodo1:

- Namenode → hdfs
- SecondaryNamenode → hdfs
- ResourceManager → yarn

## 5. ARRANCAR EL CLUSTER

---

**Paso 9.** En cada uno de los nodos, con `jps` vemos que tenemos ahora el `nodemanager`

```
hadoop@nodo1 (192.... x  hadoop@nodo2: ~ x  hadoop@nodo3: ~ x
hadoop@nodo2:~$ jps
2694 Jps
2583 NodeManager
2078 DataNode
hadoop@nodo2:~$
```

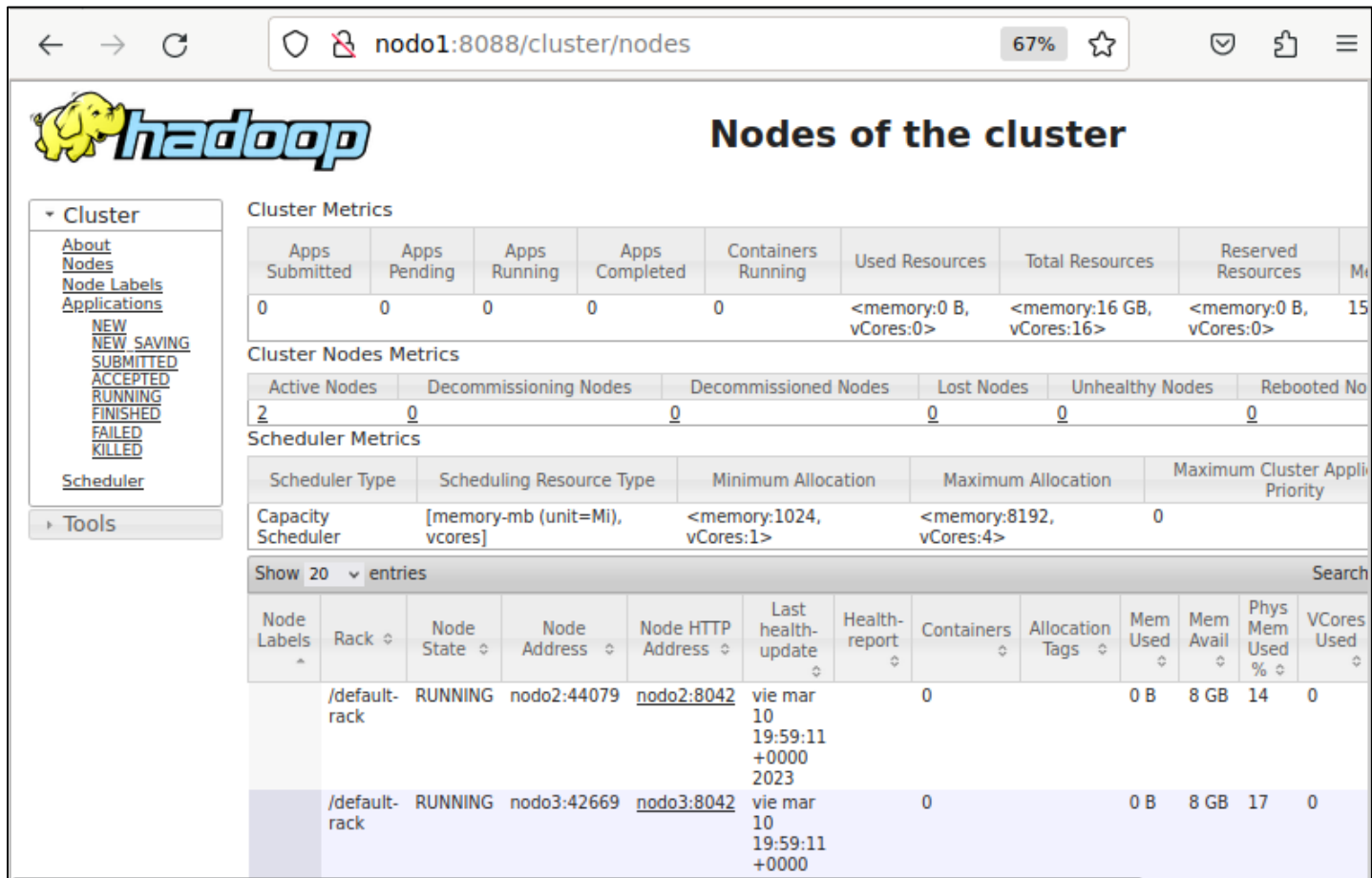
```
hadoop@nodo1 (192.... x  hadoop@nodo2: ~ x  hadoop@nodo3: ~ x
hadoop@nodo3:~$ jps
2637 NodeManager
2174 DataNode
2750 Jps
hadoop@nodo3:~$
```

Todos los procesos maestros en el `nodo1` y los dos procesos esclavos `nodemanager` y `datanode` en los `nodos 2 y 3`.



## 5. ARRANCAR EL CLUSTER

**Paso 10.** Si abrimos la web de administración del cluster nodo1:8088 en Nodes, sólo veremos los nodos esclavos: nodo2 y nodo3. Pero nunca veremos nodo1, ya que el nodo maestro no figura como esclavo.



The screenshot shows the Hadoop cluster administration web interface. The top navigation bar includes the Hadoop logo and the title "Nodes of the cluster". The sidebar on the left contains a "Cluster" section with links for "About", "Nodes", "Node Labels", "Applications", and "Scheduler". The main content area displays several tables:

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Used Resources	Total Resources	Reserved Resources	Memory
0	0	0	0	0	<memory:0 B, vCores:0>	<memory:16 GB, vCores:16>	<memory:0 B, vCores:0>	15

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
2	0	0	0	0	0

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[memory-mb (unit=M), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

Node Labels	Rack	Node State	Node Address	Node HTTP Address	Last health-update	Health-report	Containers	Allocation Tags	Mem Used	Mem Avail	Phys Mem Used %	VCores Used
/default-rack	RUNNING	nodo2:44079	nodo2:8042	vie mar 10 19:59:11 +0000 2023	0	0 B	8 GB	14	0			
/default-rack	RUNNING	nodo3:42669	nodo3:8042	vie mar 10 19:59:11 +0000	0	0 B	8 GB	17	0			