

Trabajo COS – Curso 2022/2023 – (2)

Instalación del sistema en el clúster de máquinas virtuales

1. Introducción.

Iniciar el programa *VirtualBox* en el ordenador del puesto de trabajo. Las máquinas virtuales deben estar creadas tal y como se indicó en la sesión 1.

Partimos de las máquinas con el disco duro vacío, sin particionado y sin sistema operativo. En esta sesión, particionaremos los discos e instalaremos el sistema operativo en todos los nodos del clúster. La distribución Linux elegida es la *CentOS-7-x86_64-Minimal-2009.iso*. Esta disponible en el directorio */mvirtuales/fuentes/ISOs/*. Se ha descargado de la siguiente URL http://isoredirect.centos.org/centos/7/isos/x86_64/.

La tabla siguiente resume los pasos que vamos a seguir para la instalación del sistema operativo en los distintos nodos. Los nodos Master y NFS se instalarán manualmente y los Servers se instalarán de forma autónoma.

Paso	Master	Servers	NFS
1			-Instalación local <i>CentOS 7</i> -Instalación <i>NFS</i> server -Ajustes <i>NFS</i>
2	- Instalación local <i>CentOS 7</i>		<i>Sistema CentOS 7</i>
3	<i>Sistema CentOS 7</i> -Instalación de <i>Cobbler</i> -Configuración de servicios PXE DHCP DNS	-Inicio desde PXE→ <i>Sistema CentOS desde PXE</i> -Instalación local automática de <i>CentOS7</i> y paquetes necesarios como el servidor <i>http Apache</i>	<i>Sistema CentOS 7</i>
4	<i>Sistema CentOS 7</i>	<i>Sistema CentOS 7</i>	<i>Sistema CentOS 7</i>
	Sistema arrancado		

En primer lugar, realizaremos la instalación completa del nodo de almacenamiento *NFS*, partiendo del DVD de instalación. Este nodo ofrece espacio de almacenamiento al resto de máquinas del cluster.

Seguidamente, procederemos a instalar el sistema en el nodo maestro y servidores. La estrategia es realizar una única instalación del master, después instalar los otros nodos con el uso del servidor de aprovisionamiento (*provisioning server*) *Cobbler*. El proceso que seguiremos será instalar el sistema operativo en el nodo *Master*, iniciamos y configuramos un servidor *Cobbler*, que servirá para que arranquen por red los nodos servidores, particionar sus discos duros locales, distribuir la imagen del sistema operativo, configurar cortafuegos *firewalld* y *SELinux*, instalar paquetes necesarios, etc, realizando una instalación local del sistema siguiendo un perfil de instalación (archivo *kickstart*). Una vez finalizado este proceso, los servidores arrancarán de forma automática el sistema operativo desde su disco duro local y estarán listos para incorporarse a la lista de servidores disponibles para servir las peticiones *http* de los usuarios. El uso de servidores de aprovisionamiento es tremendamente útil si el número de nodos del clúster es muy elevado.

2. Instalar el servidor de almacenamiento (“NFS”).

- Para la instalación, **conectaremos el interfaz de red al exterior** (*red-nat-1*), accediendo al menú de configuración, sección red, de la aplicación *VirtualBox*.
- Introduciremos el archivo de instalación *CentOS-7-x86_64-Minimal-2009.iso* en la unidad de DVD virtual. Arrancaremos la máquina virtual, iremos contestando a las preguntas relativas al idioma, etc y comenzaremos la instalación del sistema (*Instalar CentOS Linux 7*).
- Destino de la instalación → marcamos un disco, **“voy a configurar las particiones”** → Listo
Utilizaremos *LVM*, pulsamos “+” para añadir volúmenes y puntos de montaje

Se sugiere la siguiente configuración de los volúmenes.

- Arranque (punto de montaje: */boot*, tipo de dispositivo: *partición estándar*, Tamaño: 256 MiB, Sistema de archivos: *xfs*).
- Sistema (punto de montaje: */*, tipo de dispositivo: *LVM*, volume group: *centos*). Tamaño: 3 GiB. Sistema de archivos: *xfs*.
- Almacenamiento local (punto de montaje: */var*, tipo de dispositivo: *LVM*, volume group: *centos*). Tamaño: 6 GiB. Sistema de archivos: *xfs*.
- Área de intercambio. Punto de montaje: *swap*. Como tamaño, el espacio restante, tipo de dispositivo: *LVM*, volume group: *centos*, Sistema de archivos: *swap*

Al finalizar pulsamos listo→aceptar cambios

- Configuración de red → Interfaz desconectado, nombre del host: **nas**.
- **Empezar instalación**
- Creamos cuenta de *root*.
- Al finalizar la instalación pulsamos **“reiniciar”**. Se reinicia el sistema desde el disco duro.
- Iniciamos sesión como *root*.

Ahora realizaremos algunos ajustes en el sistema recién instalado:

Comprobar que servidor *ssh* ya esta instalado.

```
# rpm -qa | grep ssh
```

También podríamos haber comprobado si el servidor está en ejecución:

```
# systemctl status sshd
```

Configurar el servidor *ssh* para que acepte conexiones remotas a la cuenta *root*. Para ello, editaremos el archivo de configuración (puede utilizarse el editor **vi** o **nano**) y descomentamos la siguiente línea, pues por defecto no está permitida la conexión del usuario *root* por *ssh* :

```
■ /etc/ssh/sshd_config:
```

```
PermitRootLogin yes
```

Y lo reiniciamos para que los cambios se hagan efectivos:

```
# systemctl reload sshd
```

NFS (Network File System) es un protocolo que se utiliza para disponer de un sistema de archivos distribuido en una red de computadores. Lo vamos a usar para crear un directorio donde poder alojar las páginas web accesibles por los nodos servidores para poder responder a las peticiones de los usuarios. Al disponer de un almacenamiento compartido por todos los servidores, la caída de uno o varios de ellos no supondrá la parada del servicio. Lo que ocurrirá es que al disponer de menor número de servidores las peticiones tendrán un tiempo de respuesta mayor.

Crearemos además un segundo directorio compartido para almacenar algunos ficheros que *Cobbler* utilizará en el proceso de post instalación de los servidores.

Para poder proceder con la instalación de los paquetes necesarios, es preciso disponer de conexión con el mundo exterior, a continuación vamos a configurar la interfaz de red.

En primer lugar vamos a deshabilitar el NetworkManager pues no se va a usar.

```
# systemctl stop NetworkManager
```

```
# systemctl disable NetworkManager
```

Averiguamos el nombre que se le ha asignado a la tarjeta de red en el proceso de instalación.

```
# ip addr show
```

Supongamos que el nombre asignado es *enp0s3*.

Por defecto, en *CentOS* todas las interfaces de red están deshabilitadas, vamos a editar el fichero de configuración asociado a la interfaz de red, realizando los cambios necesarios y reiniciando el servicio para que los cambios se hagan visibles.

- `vi /etc/sysconfig/network-scripts/ifcfg-enp0s3:`

```
TYPE=Ethernet
BOOTPROTO=static
NAME=enp0s3
DEVICE=enp0s3
ONBOOT=yes
UUID= XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
IPADDR=192.168.1.5
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
DNS1=8.8.8.8
NM_CONTROLLED=no
```

Una vez efectuados los cambios reiniciamos el servicio

```
# systemctl restart network
```

Comprobamos que la dirección de *IP* se ha asignado correctamente:

```
# ip addr show
```

A continuación actualizaremos el sistema recién instalado:

```
# yum -y update
```

Instalamos el servidor NFS:

```
# yum -y install nfs-utils
```

Creamos dos directorios en los que se va a compartir la información y le damos permisos para que cualquiera pueda acceder. Si se quiere añadir un mayor grado de seguridad se pueden configurar para que sólo puedan acceder determinados usuarios o grupos.

```
# mkdir -p /var/web
# mkdir -p /var/post
# chown nfsnobody:nobody /var/web
# chown nfsnobody:nobody /var/post
```

Centos 7 viene con *SELinux* (seguridad mejorada) activado y por defecto no deja que *nfs* comparta directorios. Vamos a activar algunos flags o banderas para que no tengamos problemas a la hora de compartir los directorios.

```
# setsebool -P nfs_export_all_rw 1
# setsebool -P nfs_export_all_ro 1
```

Confirmar los cambios con:

```
# getsebool -a | grep nfs_export
```

Hay que modificar un atributo asociado a cada directorio para que SELinux no bloquee el acceso, para lo cual es necesario instalar el paquete:

```
# yum -y install policycoreutils-python
```

Y modificar la información de seguridad asociada a los directorios que vamos a compartir.

```
# semanage fcontext -a -t public_content_rw_t "/var/web(/.*)?"
# restorecon -Rv /var/web
# semanage fcontext -a -t public_content_rw_t "/var/post(/.*)?"
# restorecon -Rv /var/post
```

Necesitamos también añadir reglas al cortafuegos para que no bloquee el acceso desde los clientes:

```
# firewall-cmd --set-default-zone=internal
# firewall-cmd --permanent --add-service=nfs
# firewall-cmd --permanent --add-service=mountd
# firewall-cmd --permanent --add-service=rpc-bind
# firewall-cmd --reload
```

Una vez configurados el *firewall* y *SELinux* arrancamos los servicios y los habilitamos para que se inicien siempre en el proceso de arranque:

```
# systemctl start rpcbind
# systemctl start nfs-server
# systemctl enable rpcbind
# systemctl enable nfs-server
```

Hay que modificar el archivo */etc/exports* añadiendo los directorios que queremos exportar a través de *nfs*.

- */etc/exports*:

```
/var/web 10.0.100.0/24(rw,sync,no_root_squash)
/var/post 10.0.100.0/24(rw,sync,no_root_squash)
```

Para que los cambios surtan efecto reiniciamos el servidor:

```
# systemctl restart nfs-server
```

Comprobamos que se han exportado los dos directorios con la orden:

```
# exportfs -avr
```

o con la orden:

```
# showmount -e localhost
```

Es importante que todos los nodos del clúster tengan la misma fecha y la misma hora. Para este propósito disponemos de un protocolo llamado *NTP* (*Network Time Protocol*) que permite transferir la fecha y la hora a través de la red.

En primer lugar instalamos el paquete:

```
# yum -y install ntp
```

Para encontrar una lista de servidores activos hay que acceder a la página <http://www.pool.ntp.org>, visualizar los servidores activos seleccionando *Europe* y después *Spain*. Nos indica que para esta zona usemos los siguientes servidores:

```
server 0.es.pool.ntp.org
server 1.es.pool.ntp.org
server 2.es.pool.ntp.org
server 3.es.pool.ntp.org
```

Editamos el fichero de configuración `/etc/ntp.conf` añadiendo la lista de servidores y añadimos el parámetro *iburst* al final de cada línea para incrementar la velocidad de sincronización inicial.

Configuramos el cortafuegos para que permita la comunicación con el puerto *123/udp* usado por *ntp*.

```
# firewall-cmd --add-service=ntp --permanent
# firewall-cmd --reload
```

Una vez configurado el cortafuegos ejecutaremos los siguientes comandos para iniciar, habilitar en el arranque y ver el estado del servicio *ntp*.

```
# systemctl start ntpd
# systemctl enable ntpd
# systemctl status ntpd
```

Una vez finalizada la instalación de paquetes procedemos a modificar la configuración de red, pues a partir de este momento conectaremos el interfaz de red a la *red interna*.

- `vi /etc/sysconfig/network-scripts/ifcfg-enp0s3:`

```
TYPE=Ethernet
BOOTPROTO=static
NAME=enp0s3
DEVICE=enp0s3
ONBOOT=yes
UUID= xxxxxxxxxxxxxxxxxxxxxxxxxxxx
IPADDR=10.0.100.5
NETMASK=255.255.255.0
GATEWAY=10.0.100.1
DNS1=8.8.8.8
NM_CONTROLLED=no
```

Una vez efectuados los cambios reiniciamos el servicio

```
# systemctl restart network
```

Comprobamos que la dirección de *IP* se ha asignado correctamente:

```
# ip addr show
```

El nodo master con dirección IP 10.0.100.1 actuará de *router (gateway)* y *DNS* para todos los equipos conectados a la red interna 10.0.100.0/24.

Una vez completado, apagar la máquina con la orden *poweroff*, conectar el interfaz de red a la red interna, modificar el orden de arranque para que la primera opción sea HDD y arrancar de nuevo la máquina. En este punto sacaremos una **instantánea** del nodo de almacenamiento. Se sugiere que la instantánea se llame “**NFS-base**”.

3. Instalar el nodo maestro.

- Introduciremos el archivo de instalación “.iso” descargado previamente en la unidad de DVD virtual. Arrancaremos la máquina virtual, iremos contestando a las preguntas relativas al idioma, etc y comenzaremos la instalación del sistema (Instalar CentOS Linux 7).
- Destino de la instalación → marcamos un disco, “voy a configurar las particiones” →Listo
Utilizaremos LVM, pulsamos “+” para añadir volúmenes y puntos de montaje

Se sugiere la siguiente configuración de los volúmenes.

- Arranque (punto de montaje: /boot, tipo de dispositivo: partición estándar, Tamaño: 256 MiB, Sistema de archivos: xfs.
- Sistema (punto de montaje: /, tipo de dispositivo: LVM, volume group: centos). Tamaño: 4 GiB. Sistema de archivos: xfs.
- Almacenamiento local (punto de montaje /var, tipo de dispositivo: LVM, volume group: centos). Tamaño: 9 GiB. Sistema de archivos: xfs.
- Área de intercambio. Punto de montaje: swap. Como tamaño, el espacio restante, tipo de dispositivo: LVM, volume group: centos, Sistema de archivos: swap

Al finalizar pulsamos listo→aceptar cambios

- Configuración de red -> Interfaz desconectado, nombre del host: master.
- Empezar instalación
- Creamos cuenta de root. Usar siempre la misma contraseña elegida en la instalación del primer nodo.

Configurar las interfaces de red.

En primer lugar vamos a apagar y deshabilitar el NetworkManager pues no se va a usar:

```
# systemctl stop NetworkManager  
# systemctl disable NetworkManager
```

Los nombres de las interfaces se asignan automáticamente en el proceso de instalación. Es posible ver las interfaces de red disponibles con la orden:

```
# ip addr show
```

Vamos a suponer que los nombres asignados son:

Red-nat: enp0s3

Red interna: enp0s8

Por defecto en CentOS todas las interfaces de red están deshabilitadas, vamos a editar los ficheros de configuración asociados a las interfaces de red, realizando los cambios necesarios y reiniciando el servicio para que los cambios se hagan visibles.

Interfaz de *red nat*

- `vi /etc/sysconfig/network-scripts/ifcfg-enp0s3:`

```
TYPE=Ethernet
BOOTPROTO=static
NAME=enp0s3
DEVICE=enp0s3
ONBOOT=yes
UUID=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
IPADDR=192.168.1.5
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
DNS1=8.8.8.8
ZONE=external
NM_CONTROLLED=no
```

Interfaz de *red interna*

- `vi /etc/sysconfig/network-scripts/ifcfg-enp0s8:`

```
TYPE=Ethernet
BOOTPROTO=static
NAME=enp0s8
DEVICE=enp0s8
ONBOOT=yes
UUID=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
IPADDR=10.0.100.1
NETMASK=255.255.255.0
ZONE=internal
NM_CONTROLLED=no
```


Una vez efectuados los cambios reiniciamos el servicio

```
# systemctl restart network
```

Comprobamos que las dos interfaces de red tienen direcciones IP asignadas:

```
# ip addr show
```

Para finalizar comprobamos que tenemos conexión con el mundo exterior

```
# ping www.upv.es
```

Para permitir que el nodo master actúe como *router* para el resto de nodos de la red interna tenemos que habilitar *IP Forwarding*, que permitirá el paso de paquetes entre sus dos interfaces de red.

Habilitar IP Forwarding

```
# sysctl -w net.ipv4.ip_forward=1
```

para hacer los cambios permanentes tras reiniciar la máquina:

```
echo "net.ipv4.ip_forward=1" >> /etc/sysctl.conf
```

Configurar *NAT (Network Address Translation)* en el cortafuegos:

```
# firewall-cmd --set-default-zone=internal
# firewall-cmd --change-interface=enp0s3 --zone=external --permanent
# firewall-cmd --zone=external --add-masquerade --permanent
# firewall-cmd --permanent --direct --passthrough ipv4 -t nat -I POSTROUTING -o enp0s3 -j MASQUERADE -s 10.0.100.0/24
# firewall-cmd --complete-reload
# firewall-cmd --zone=internal --add-service=dns --permanent
# systemctl restart network && systemctl restart firewalld
```

Para comprobar que funciona correctamente desde el nodo *nfs*, podemos comprobar si tiene conexión con una máquina externa.

```
ping www.google.es
```

A continuación instalamos y configuramos el servicio *ntp* de forma análoga a cómo se hizo anteriormente en el nodo de almacenamiento.

Modificamos el archivo */etc/hosts*, añadiendo las parejas IP – nombre para la resolución de nombres. En nuestro clúster, adoptaremos el convenio de denominar a todas las máquinas con el mismo prefijo. Por ejemplo, si el prefijo es “cluster”, el nodo maestro será *cluster1*, y los nodos servidores irán de *cluster2* en adelante. Así el archivo quedaría:

- */etc/hosts*

```
127.0.0.1    localhost

10.0.100.5   nas nfserver nas.cluster nfserver.cluster

10.0.100.1   cluster1    cluster1.cluster    master
```

10.0.100.12	cluster2	cluster2.cluster
10.0.100.13	cluster3	cluster3.cluster
10.0.100.14	cluster4	cluster4.cluster

También aprovechamos para copiarlo al servidor NFS:

```
# scp /etc/hosts nas:/etc
```

Preparación para el acceso por ssh mediante clave pública. Copiaremos la clave pública del nodo maestro sobre el archivo que contiene las claves autorizadas:

```
# ssh-keygen
# cp /root/.ssh/id_rsa.pub /root/.ssh/authorized_keys
```

También aprovechamos para copiar las claves al servidor NFS:

```
# ssh nas "mkdir /root/.ssh"
# scp /root/.ssh/id_rsa.pub nas:/root/.ssh/authorized_keys
```

Instalar paquetes adicionales (cliente NFS):

```
# yum -y install nfs-utils
```

Comprobar los directorios exportados por el nodo de almacenamiento:

```
# showmount -e nas
```

Creamos dos directorios para realizar el montaje:

```
# mkdir -p /mnt/web
# mkdir -p /mnt/post
```

Editar el fichero `/etc/fstab` y añadir lo siguiente:

nas:/var/web	/mnt/web	nfs	defaults,_netdev,rw	0	0
nas:/var/post	/mnt/post	nfs	defaults,_netdev,rw	0	0

Ejecutamos la orden

```
# mount -a
```

comprobamos que tenemos montados ambos directorios:

```
# df -h
```

y que tenemos acceso a ellos:

```
# cd /mnt/web
# touch test1
# ls -al
```

3.1 Instalación y configuración de *Cobbler*.

Cuando hay que poner en marcha un elevado número de nodos físicos (bare metal) o de máquinas virtuales, los administradores de sistemas hacen uso de servidores de aprovisionamiento (provisioning servers). Estos servidores permiten instalar el sistema operativo en los discos duros locales de los nodos, instalar paquetes, configurar cortafuegos y *SELinux*, etc, de forma autónoma sin intervención del administrador. Existen varios servidores de aprovisionamiento gratuitos entre los que cabe destacar: *Cobbler*, *xCAT*, *Spacewalk*, *FAI*, etc. En el presente trabajo vamos a usar *Cobbler*, propiedad de *Red Hat*, por lo que no hay ningún problema al usarlo con *CentOS*.

Cobbler es un servidor de aprovisionamiento Linux que permite la configuración automática de nodos en entornos de red. Se puede utilizar para aprovisionar tanto máquinas físicas como máquinas virtuales de los principales hipervisores, *XEN*, *KVM*, *WMware*, *Virtualbox*, etc.

Une y automatiza tareas asociadas con diferentes comandos Linux utilizados a la hora de poner en marcha un nuevo sistema, simplificando las tareas de administración. Hace uso de servicios como PXE, DHCP, DNS, TFTP, etc.

Los nodos servidores arrancaran por red (PXE) y cada uno de ellos recibirá de forma automática la información necesaria para la instalación del sistema operativo, instalación de paquetes, configuración del adaptador de red, cortafuegos, configuración del servidor web, etc. Al finalizar este proceso el nodo esta listo para usarse sin intervención del administrador del sistema.

A continuación vamos a instalar y configurar *Cobbler*.

Para obtener los paquetes necesarios vamos a instalar y utilizar EPEL (Extra Packages for Enterprise Linux repository)

```
# yum install -y epel-release
```

Instalamos los paquetes necesarios

```
# yum install -y cobbler dnsmasq syslinux pykickstart xinetd httpd
```

Una vez instalados los paquetes hay que ponerlos en marcha y habilitarlos para que se inicien siempre que arranque el sistema:

```
# systemctl start cobblerd; systemctl enable cobblerd
# systemctl start httpd; systemctl enable httpd
# systemctl start tftp; systemctl enable tftp
# systemctl start xinetd; systemctl enable xinetd
```

Para habilitar el servicio TFTP accedemos al fichero */etc/xinetd.d/tftp* y cambiamos la opción:

```
disable = no
```

Reiniciamos el servicio:

```
# systemctl restart tftp
```

Es necesario generar un *hash* de la contraseña de *root* que *Cobbler* usara por defecto para nuevas instalaciones.

```
# openssl passwd -1 'vuestro_password_de_root' > /tmp/hash
# sed -i.bak "/default_password_crypted:/c\default_password_crypted: \"$(cat
/tmp/hash)\" /etc/cobbler/settings
```

Modificamos el fichero de configuración */etc/cobbler/setting* para actualizar esta cadena cifrada en el parámetro *'default_password_crypted'* y también habilitar la función *DHCP*, *DNS*, *PXE* y *TFTP* de *Cobbler* cambiando el valor del parámetro de 0 a 1.

Especificamos la dirección IP de su servidor TFTP en el parámetro *'next_server'* y la dirección IP del servidor *Cobbler* en el parámetro *'server'*. En nuestro caso en ambos parámetros colocamos la dirección IP del nodo master.

- # vi /etc/cobbler/settings

```
manage_dhcp: 1
manage_dns: 1
pxe_just_once: 1
next_server: 10.0.100.1
server: 10.0.100.1
```

Comprobar que en el parámetro *default_password_crypted*: se ha asignado correctamente el hash de vuestra contraseña de *root* (*/tmp/hash*).

El resto del fichero no se debe modificar.

El siguiente fichero que hay que modificar es */etc/cobbler/modules.conf*

- # vi /etc/cobbler/modules.conf

```
[dns]
module = manage_dnsmasq

[dhcp]
module = manage_dnsmasq
```

El resto del fichero no se debe modificar.

Indicándole que los servicios *DHCP* y *DNS* los gestionara el paquete *dnsmasq*, por defecto no es así. Si se opta por tener estos servicios por separado con sus propios demonios, estos se deben configurar manualmente.

Para establecer la configuración del servicio *DHCP* se debe editar el fichero */etc/cobbler/dhcp.template* y modificar lo siguiente:

- #vi /etc/cobbler/dhcp.template

```
subnet 10.0.100.0 netmask 255.255.255.0 {
```

```
option routers          10.0.100.1;
option domain-name-servers 10.0.100.1;
option subnet-mask      255.255.255.0;
range dynamic-bootp     10.0.100.10 10.0.100.253;
```

El resto del fichero no se debe modificar.

Y a continuación el fichero `/etc/cobbler/dnsmasq.template` y añadir los siguientes datos:

```
#vi /etc/cobbler/dnsmasq.template

interface=enp0s8
bind-interfaces
dhcp-range=10.0.100.10,10.0.100.253
```

El resto del fichero no se debe modificar.

A continuación se establecen las reglas necesarias en el cortafuegos para que no bloquee el tráfico a los puertos de escucha de los diferentes servicios:

```
# firewall-cmd --add-service=http --permanent
# firewall-cmd --add-service=https --permanent
# firewall-cmd --add-service=dhcp --permanent
# firewall-cmd --add-port=69/tcp --permanent
# firewall-cmd --add-port=69/udp --permanent
# firewall-cmd --add-port=4011/udp --permanent
# firewall-cmd --reload
```

Es muy importante que el nodo master que es el *front-end* del clúster tenga el módulo de seguridad mejorada (*SELinux*) activo y permitamos modificando algunas variables booleanas que deje trabajar a los diferentes servicios gestionados por *Cobbler* sin restricciones:

```
# setsebool -P cobbler_can_network_connect 1
# setsebool -P httpd_can_network_connect_cobbler 1
# setsebool -P httpd_serve_cobbler_files 1
# restorecon -R /var/lib/cobbler
# chcon -R -t cobbler_var_lib_t /etc/dnsmasq.conf
# touch /etc/ethers
# chcon -R -t cobbler_var_lib_t /etc/ethers
```

Descargamos los cargadores de arranque:

```
# yum install syslinux -y
# cp -r /usr/share/syslinux/* /var/lib/cobbler/loaders/
```

Reiniciamos todos los servicios, comprobamos la configuración y sincronizamos todo

```
# systemctl restart cobblerd xinetd
```

```
# cobbler check
# cobbler sync
```

Es necesario obtener el fichero “.iso” con la imagen del sistema operativo que queremos que *Cobbler* instale en los nodos servidores. Obtendremos la imagen con el comando *wget*.

```
# yum -y install wget
# wget http://isoredirect.centos.org/centos/7/isos/x86\_64/CentOS-7-x86\_64-Minimal-2009.iso
```

Importamos la imagen a *Cobbler*:

```
# mkdir -p /mnt/iso/centos7
# mount -o loop CentOS-7-x86_64-Minimal-2009.iso /mnt/iso/centos7
# cobbler import --arch=x86_64 --path=/mnt/iso/centos7 --name=CentOS7
# cobbler signature update
# cobbler distro list
    CentOS7-x86_64
# cobbler distro report --name=CentOS7-x86_64
```

A continuación hay que generar un fichero *kickstart* que le indicará los pasos a seguir en el proceso de instalación del sistema operativo. La ubicación de todas las plantillas *kickstart* es */var/lib/cobbler/kickstarts*. Crearemos en primer lugar una plantilla para el nodo *server1*, la llamaremos *server1.ks*.

- # vi /var/lib/cobbler/kickstarts/server1.ks

```
# Deshabilitamos el cortafuegos
firewall --disabled
# Instalar/actualizar sistema operativo
install
# Medio de arranque por red
url --url="http://10.0.100.1/cblr/links/CentOS7-x86\_64/"
# Hash del password de Root obtenido anteriormente
# Cambiar XXXXXXXX por el hash en la siguiente línea
rootpw --iscrypted XXXXXXXX
# Información red, toda la orden en una sola línea
network --bootproto=static --ip=10.0.100.12 --netmask=255.255.255.0
--gateway=10.0.100.1 --nameserver=10.0.100.1,8.8.8.8 --device=enp0s3
```

```
--hostname=cluster2 --onboot=on --noipv6 --activate
# Reboot despues de la instalación
reboot
# Tipo de encriptacion
auth --enableshadow --passalgo=sha512
# Usar instalación modo texto
text
firstboot --disable
# Teclado
keyboard es
# Lenguaje del sistema
lang en_US
# Configuración SELinux
selinux --disabled
# Configuración LOGS
logging level=info
# Zona Horaria
timezone Europe/Madrid
# Arranque del sistema
bootloader location=mbr
clearpart --all --initlabel --drives=sda
# Particionado del disco duro
part swap --asprimary --fstype="swap" --size=1024
part /boot --fstype xfs --size=256
part pv.01 --size=1 --grow
volgroup root_vg01 pv.01
logvol / --fstype xfs --name=lv_01 --vgname=root_vg01 --size=1 --grow
# Paquetes incluidos en la instalacion
%packages
@^minimal
@core
%end
# En post arranque se especifican comandos a ejecutar
%post
yum -y install epel-release
yum -y update
yum -y upgrade
```

```
yum -y install httpd ntp nfs-utils
yum -y install php wget net-tools

# Configuración NFS

# Creamos un punto de montaje para establecer conexión con el nodo nas.
mkdir -p /mnt/post
mkdir -p /mnt/web

# Creamos una entrada en el fichero /etc/fstab para montar el directorio
compartido del nodo de almacenamiento
echo "10.0.100.5:/var/post /mnt/post nfs defaults,_netdev,rw 0 0" >>
/etc/fstab
echo "10.0.100.5:/var/web /mnt/web nfs defaults,_netdev,rw 0 0 >>
/etc/fstab
mount -a

# Se copian por red todos los ficheros de configuración necesarios para
cada nodo
cat /mnt/post/hosts > /etc/hosts
systemctl disable NetworkManager

# acceso SSH por clave pública sin contraseña
sed -i 's/#PermitRootLogin no /PermitRootLogin yes/g' /etc/ssh/sshd_config
sed -i 's/#PermitRootLogin yes /PermitRootLogin yes/g' /etc/ssh/sshd_config
mkdir -p -mode=700 /root/.ssh

# Cambiar XXXXXXXX por la clave pública en la siguiente línea (incluyendo
la parte de ssh-rsa). Esta contenida en el fichero /root/.ssh/id_rsa.pub
cat >> /root/.ssh/authorized_keys << EOF
XXXXXXXX
EOF
chmod 600 /root/.ssh/authorized_keys
systemctl reload sshd
systemctl start httpd
systemctl enable httpd

# Configuración NTP
cp /mnt/post/ntp.conf /etc/ntp.conf
systemctl start ntpd
systemctl enable ntpd

%end
```


Durante el proceso de instalación de los nodos servidores el nodo de almacenamiento debe estar operativo pues en el proceso de post-arranque van a leer sus ficheros de configuración de uno de los directorios que exporta.

En el directorio compartido */var/post* del nodo de almacenamiento dejamos los ficheros necesarios para el proceso de post instalación.

- */var/post/hosts*

```
127.0.0.1 localhost
10.0.100.5  nas nfsserver nas.cluster nfsserver.cluster
10.0.100.1      cluster1      cluster1.cluster  master
10.0.100.12     cluster2      cluster2.cluster
10.0.100.13     cluster3      cluster3.cluster
10.0.100.14     cluster4      cluster4.cluster
```

- */var/post/ntp.conf*

```
server 10.0.100.1 prefer
```

Una vez tenemos el fichero *kickstart* preparado hay que crear un perfil de distribución en *Cobbler*:

```
# cobbler profile add --name CentOS7-Server1 --distro=CentOS7-x86_64
--kickstart=/var/lib/cobbler/kickstarts/server1.ks
```

Antes de iniciar la instalación es conveniente revisar el fichero *kickstart* para detectar posibles errores, sino se comprueba y se inicia la instalación, esta se detendría una vez encontrase el primer error:

```
# cobbler validateks
```

Seguidamente se pueden asignar perfiles a máquinas concretas. El parámetro “*interface*” indica la interfaz por donde *Cobbler* detectará la máquina y el parámetro “*mac*” indica la *mac* de la máquina a la que *Cobbler* instalará la distro elegida, siguiendo las indicaciones del fichero *kickstart*.

```
# cobbler system add --name=server1 --profile=CentOS7-Server1
--interface=enp0s8 --mac=08:00:27:01:01:F1
```

Para finalizar comprobamos que todo esta correcto con los siguientes comandos:

```
# cobbler check
# cobbler sync
```

Si no hay ningún problema, ponemos en marcha el nodo *server1*, indicando como primera opción de arranque el disco duro y después arranque por red y *Cobbler* se encarga del resto.

Habría que generar ficheros *kickstart* para el resto de servidores. Sólo cambiaría la configuración de red. Se podrían generar manualmente o automatizar el proceso con algún fichero de comandos. Por

ejemplo copiamos el fichero server1.ks sobre el fichero server2.ks, lo editamos y cambiamos lo siguiente:

```
network --bootproto=static --ip=10.0.100.13 --netmask=255.255.255.0
--gateway=10.0.100.1 --nameserver=10.0.100.1,8.8.8.8 --device=enp0s3
--hostname=cluster3 --onboot=on --noipv6 --activate
```

Habría que generar nuevos perfiles y asignarlos a al resto de servidores en función de su *mac*.

```
# cobbler profile add --name CentOS7-Server2 --distro=CentOS7-x86_64
--kickstart=/var/lib/cobbler/kickstarts/server2.ks
# cobbler system add --name=server2 --profile=CentOS7-Server2
--interface=enp0s8 --mac=08:00:27:01:01:F2
# cobbler chek
# cobbler sync
```