

B1 - *OS* y Servicios

Antonio Javier Rodríguez Romero



**UNIVERSIDAD
DE GRANADA**

Ingeniería de Servidores
3ºDGIIM

1. Instalación y configuración del servidor.

1.1. Ejercicio evaluable

El alumno/a debe ser capaz de presentar un MV con la configuración descrita en este apartado. La configuración debe ser permanente, es decir, en todo caso, tras reiniciar el equipo, la configuración será la esperada.

Para validar la configuración de red, el alumno/a debe ser capaz de:

- Hacer ping desde el equipo anfitrión a la MV y viceversa.
- Hacer ping desde la MV a cualquier equipo accesible públicamente en Internet por FQHN o IP.
- Conectar por ssh desde el equipo anfitrión a la MV.

Para realizar este apartado, primero creamos la MV en *Virtual Box* con la imagen de *RedHat* descargada y configuramos la hora y el idioma. En la configuración de la máquina habilitamos dos interfaces para esta máquina, una *NAT* y otra *Host-Only*.

Acto seguido, dentro de la máquina tendremos que:

- Crear un usuario nuevo y le asignamos una contraseña:
 - ***useradd antonio //passwd antonio*** (E introducimos nuestra contraseña).
- Para darle privilegios al usuario nuevo lo añadimos al grupo *wheel*:
 - ***usermod -aG wheel antonio***
- Para cambiar el prompt al buscado, editaremos el archivo *~/bashrc*, añadiendo al final del archivo la línea: ***PS1="[/u@/h-t /W]/\$"***. Una vez añadida la línea, ejecutamos:
 - ***source ~/.bashrc***

Deberíamos observar que ahora el formato de nuestro *prompt* es:

[<usuario>@<host>-<hora> <directorio>]\$

- Cambiamos el nombre del *host*:
 - ***hostnamectl set-hostname ajrrMV01***
- Para configurar la *IP* estática de la máquina utilizaremos la herramienta ***nmcli***. Por un lado, le cambiaré el nombre para tener claro cuál es la que conecta con nuestro host, y por otro le indicaremos que tiene que ser estática y la dirección que tendrá:
 - ***nmcli con modify "Wired connection 1" connection.id "Host-Only"***
 - ***nmcli con modify Host-Only ipv4.addresses 192.168.56.102*** (Le he asignado esta pq en VBox he visto que la del host es 192.168.56.101)
 - ***nmcli con modify Host-Only ipv4.method manual***
- Comprobamos que la configuración es correcta mirando la *IP*:
 - ***ip a***
- Una vez asignada la *IP* y conocida la de nuestro *Host*, podremos comprobar que la conexión es correcta.

```
[antonio@ajrrMV01-17:23:42 ~]$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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: enp8s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:08:d4:4b brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp8s3
        valid_lft 85150sec preferred_lft 85150sec
    inet6 fe80::a00:27ff:fe08:d44b/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: enp8s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:c1:3c:e5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.102/24 brd 192.168.56.255 scope global noprefixroute enp8s8
        valid_lft forever preferred_lft forever
    inet6 fe80::ab2b:2222:9154:2fcc/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

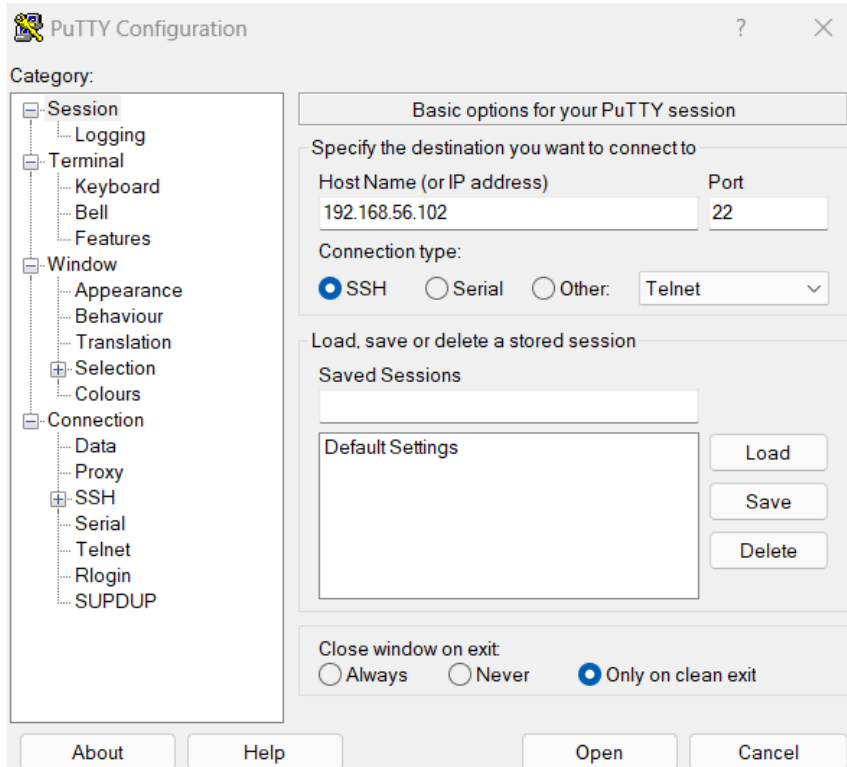
Adaptador de Ethernet Ethernet 2:

```
Sufijo DNS específico para la conexión. . . :
Vínculo: dirección IPv6 local. . . : fe80::698:acca:5be6:d7e0%8
Dirección IPv4. . . . . : 192.168.56.101
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . :
```

```
Haciendo ping a 192.168.56.102 con 32 bytes de datos:
Respuesta desde 192.168.56.102: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.56.102: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.56.102: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.56.102: bytes=32 tiempo<1m TTL=64
```

```
Estadísticas de ping para 192.168.56.102:
Paquetes: enviados = 4, recibidos = 4, perdidos = 0
(0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
Mínimo = 0ms, Máximo = 1ms, Media = 0ms
```

```
antonio@ajrrm01-17:23:47 ~]$ping google.com
PING google.com (172.217.17.14) 56(84) bytes of data.
64 bytes from mad07s09-in-f14.1e100.net (172.217.17.14): icmp_seq=1 ttl=115 time=19.9 ms
64 bytes from mad07s09-in-f14.1e100.net (172.217.17.14): icmp_seq=2 ttl=115 time=22.5 ms
64 bytes from mad07s09-in-f14.1e100.net (172.217.17.14): icmp_seq=3 ttl=115 time=22.7 ms
64 bytes from mad07s09-in-f14.1e100.net (172.217.17.14): icmp_seq=4 ttl=115 time=72.5 ms
64 bytes from mad07s09-in-f14.1e100.net (172.217.17.14): icmp_seq=5 ttl=115 time=27.2 ms
64 bytes from mad07s09-in-f14.1e100.net (172.217.17.14): icmp_seq=6 ttl=115 time=20.8 ms
64 bytes from mad07s09-in-f14.1e100.net (172.217.17.14): icmp_seq=7 ttl=115 time=39.9 ms
^C
--- google.com ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6014ms
rtt min/avg/max/mdev = 19.885/32.105/72.520/17.700 ms
antonio@ajrrm01-17:27:42 ~]$
```



```
antonio@ajrrMV01:~  
login as: antonio  
antonio@192.168.56.102's password:  
Last login: Fri Apr 5 17:03:04 2024  
[antonio@ajrrMV01-17:37:26 ~]$
```

2. Configuración de LVM y RAID.

2.1. Ejercicio evaluable

Partiendo de un servidor básico configurado de acuerdo al apartado 2, el alumno/a deberá afrontar el caso práctico descrito a continuación:

Se desea instalar un servicio de gestión documental en el servidor. Se espera que este servicio precise de una cantidad de espacio de almacenamiento creciente con el tiempo, pudiendo llegar a ser considerable.

Por otro lado, el contenido será crítico, por lo que se desea proporcionar algún mecanismo de respaldo ante fallos en el dispositivo de almacenamiento.

El alumno/a debe diseñar los cambios en el sistema de almacenamiento e implementarlo empleando prácticas adecuadas de administración que garanticen la conservación de la información en el sistema y procuren la máxima disponibilidad del servicio.

Primero, nos fijamos que se nos pide un servicio de gestión documental con respaldo frente a fallos y con una cantidad de espacio creciente con el tiempo, luego lo mejor será utilizar un *RAID 5* para ello, debido al respaldo frente a fallos que aporta y su flexibilidad en cuanto al almacenamiento.

Para esto, añadiremos en *VBox* 3 discos extra a nuestra máquina.

Comprobamos que son reconocidos por la MV con *lsblk*.

En nuestro caso son el *sdb*, *sdc* y *sdd*.

Para la configuración de la *raid* utilizaremos la herramienta *mdadm*. Si no la tenemos instalada, tendremos que hacerlo (*sudo dnf install mdadm*).

Ahora para la creación de la *raid* ejecutaremos:

sudo mdadm --create --verbose /dev/md0 --level=5 --raid-devices=3 /dev/sdb /dev/sdc /dev/sdd
Explicaremos lo que hace cada parámetro del comando:

--create: Esto indica que vamos a crear un nuevo RAID

--verbose: Es opcional, lo que hace básicamente es que te de feedback cuando se ejecuta el comando para ver que todo va bien.

/dev/md0: Nombre que le vamos a dar a la RAID

--level=5: Con esto le decimos que queremos hacer un RAID 1

--raid-devices=3: Indica el número de dispositivos que vamos a meter en el RAID, 2 en este caso.

Ahora para utilizar LVM sobre la raid, creamos un physical volumen sobre ella con

pvccreate /dev/md0

para ahora marcarlo como volumen group

vgcreate raid /dev/md0

y añadir una capa de abstracción para flexibilizar el añadir almacenamiento

lvcreate -L 3G -n nvar raid

Una vez creado el volumen lógico le asignamos un sistema de archivos entre ext4 o xfs. xfs es para tamaños de almacenamiento muy grandes, luego usaré ext4.

mkfs.ext4 /dev/raid/nvar

Montamos ahora este sistema de archivos en un directorio temporal:

mkdir /mnt/nvar

mount /dev/raid/nvar /mnt/nvar

Ahora ponemos el sistema en modo mantenimiento, es decir, en el target que solo permite la actividad del usuario root para que ningún usuario genere datos en /var:

systemctl isolate rescue.target

Copiamos toda la información de /var a /mnt/nvar:

cp -a /var/* /mnt/nvar/

y desmontamos:

umount /mnt/nvar

Cambiamos /var por /oldvar, por si algo sale mal que no lo hayamos eliminado:

mv /var /oldvar

Creamos la carpeta /var otra vez y montamos nuestro LV en ella:

mkdir /var

mount /dev/raid/nvar /var

y volvemos el sistema al estado normal

systemctl default

Con lsblk podemos comprobar si se ha realizado correctamente



```

NAME            MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINTS
sda              8:0    0   20G  0 disk
├─sda1           8:1    0    1G  0 part  /boot
├─sda2           8:2    0   19G  0 part
├─rl-root       253:0    0    17G  0 lvm    /
└─rl-swap       253:1    0    2G  0 lvm    [SWAP]
sdb              8:16    0    2G  0 disk
├─md0           9:0    0    4G  0 raid5
├─raid-nvar    253:2    0    3G  0 lvm    /var
sdc              8:32    0    2G  0 disk
├─md0           9:0    0    4G  0 raid5
├─raid-nvar    253:2    0    3G  0 lvm    /var
sdd              8:48    0    2G  0 disk
├─md0           9:0    0    4G  0 raid5
├─raid-nvar    253:2    0    3G  0 lvm    /var
sr0             11:0    1 1024M  0 rom
antonio@ajrrM01-11:55:59 ~$

```

3. Acceso seguro al servidor.

3.1. Ejercicio evaluable

Como caso práctico, partiendo de una MV con la configuración base descrita en el apartado 2, el alumno/a deberá ser capaz de instalar un servidor de HTTP, Apache o Nginx, y habilitar/deshabilitar su acceso por Firewall.

Para ello, instalará el servidor web de su elección y modificará la home page para mostrar un mensaje: “Bienvenidos a la web de <Nombre y Apellidos del alumno/a> en Prácticas ISE”.

El servicio web debe estar accesible en la servidor (MV) en el puerto por defecto (80) usando un navegador web convencional corriendo en el anfitrión (Host).

Un escaneo de puertos sobre el servidor solo debe mostrar como accesibles los puerto web y ssh.

Instalaré un servidor web apache, el paquete es httpd:

```
sudo dnf install httpd
```

El servicio de firewall se supone que tiene que estar instalado. Iniciamos el servicio httpd:

```
sudo systemctl start httpd
```

```
sudo systemctl enable httpd
```

Modificamos ya el archivo html donde pondremos la página web:

```
nano /var/www/html/index.html
```

y ponemos

```
<html>
```

```
<body>
```

```
<h1>Bienvenidos a la web de <nombre> en Prácticas ISE</h1>
```

```
</body>
```

```
</html>
```

ctrl+O para guardar y **ctrl+X** para salir.

Para que se apliquen los cambios:

```
sudo systemctl restart httpd
```

Ahora, con el servicio firewall-cmd añadimos el puerto 80 para que permita el acceso a la web:

```
sudo systemctl start firewalld
```

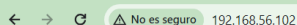
```
sudo systemctl enable firewalld
```

y permitimos el servicio http

```
sudo firewall-cmd --permanent --add-service=http
```

```
sudo firewall-cmd --reload
```

Si hemos hecho todo bien, accederemos a la página web poniendo en el buscador de nuestro host la ip de la MV.



Bienvenidos a la web de Antonio Javier Rodríguez Romero en Prácticas ISE

3.2. Ejercicio evaluable

Partiendo de un servidor base configurado siguiendo las indicaciones del apartado 2, el alumno/a modificará servicio SSHD para que, en lugar del puerto por defecto (22), se ejecute en un puerto alternativo de un valor mayor a 1024.

Se recomienda que consulte la lista de puertos reconocidos por el sistema en /etc/ports para evitar emplear un puerto que ya tenga una aplicación predefinida.

Se concederá acceso remoto por llave pública a un usuario de su elección.

El ejercicio se validará ejecutando un comando de forma remota sobre el servidor SSH con la nueva configuración. El comando presentará el contenido completo (incluido ficheros y directorios ocultos) con el directorio home del usuario remoto empleado en la conexión.

Para ello, desde el ordenador anfitrión (o una MV distinta a la que se va a acceder) se empleará ssh sin terminal remoto y sin contraseña, pasando como único como parámetro el comando a ejecutar.

Para comprobar los puertos en uso:

lsof -i -P -n

Para cambiar el puerto que usará nuestro servicio sshd, yo por ejemplo lo voy a poner en el 2222:

sudo semanage -a -t ssh_port_t -p tcp 2222

SI NO TIENES *semanage*, instalar paquete *policycoreutils-python-utils*.

Ahora buscamos en /etc/ssh/sshd_config donde pone #Port 22 y descomentamos y cambiamos por Port 2222.

Ahora con el servicio firewalld tenemos que permitir el tráfico por el puerto que hemos puesto:

sudo systemctl start firewalld

sudo systemctl enable firewalld

sudo firewall-cmd --permanent --add-port=2222/tcp

y reiniciamos el servicio sshd y firewalld

sudo systemctl restart sshd

sudo systemctl restart firewalld

Si ejecutamos lsof otra vez deberíamos ver el cambio.

Ahora probamos a conectar por ssh en el puerto nuevo con

ssh -p 2222 <ip>

Ahora vamos a la parte de la clave. Primero generamos un clave ssh en el host:

ssh-keygen

Le ponemos un nombre reconocible como “*ansible*” y no le ponemos contraseña. Encontraremos tanto la privada como la pública en ~/.ssh/ansible(.pub). Ahora le pasamos al servidor la clave:

ssh-copy-id -i ~/.ssh/ansible.pub <usuario>@<ip>

Con esto, cuando queramos conectarnos al servidor ya no nos pedirá la contraseña de este usuario si le indicamos la clave privada Y EL PUERTO (2222):

ssh -p 2222 -i ~/.ssh/ansible <usuario>@<ip>

Para la ejecución de un comando automáticamente se lo pondremos al final. El que nos pide en el enunciado será:

ssh -p 2222 -i ~/.ssh/ansible <usuario>@<ip> “ls -la”

4. Automatización con Ansible.

Tutoriales y ChatGPT