Instalación de paquetes

Para administrar nuestro servidor necesitaremos de algunos paquetes fundamentales. Se requieren estos paquetes para administrar remotamente el servidor y configurar la red de este. Los mismos serán descargados desde el repositorio oficial de Red Hat. Los paquetes serán:

NetworkManager-tui

Proporciona una interfaz de usuario para administrar la configuración de la red. Para ejecutar esta herramienta se utiliza el comando *nmtui*.

Comando para instalarlo:

sudo dnf install NetworkManager-tui

Cockpit

Desarrollada por Red Hat, permite a los usuarios administrar servidores Linux de forma remota a través de un navegador e interfaz gráfica.

Comando para instalarlo:

sudo dnf install cockpit

Openssh-server

Para que funcione el protocolo ssh es necesario instalar el servicio en el sistema operativo del servidor. Además es necesario instalar un cliente ssh en el dispositivo que quiere conectarse al servidor. El paquete openssh-server es fundamental para acceder remotamente a un servidor mediante CLI, proporcionando el servicio necesario para establecer la conexión.

Comando para instalarlo:

sudo dnf install openssh-server

Configuración

Paso 1

El primer paso para la configuración de ssh es instalarlo con el comando anterior.

Paso 2

Posteriormente, se comprueba su estado. En Red Hat y derivados, se utiliza el comando *"systemctl status sshd"* para ver el estado, en caso de que no esté activo el servicio, se puede iniciar con *"systemctl start sshd"*. Otra forma es *"systemctl enable sshd" lo que hará que el servicio quede activo siempre que arranque el servidor.*

Paso 3

Comprobar el firewall. A veces este software bloquea las comunicaciones entrantes y salientes a este y otros servicios. Para verificar si el firewall permite comunicación con ssh se introduce el comando "*firewall-cmd --list-services*" o para más información *"firewall-cmd --list-all".* En caso de no aparecer ssh, se utiliza "firewall-*cmd --permanet--add-services=ssh*". Se reinicia el firewall con *"firewall-cmd --reload"* y deberá quedar listo. Otras complicaciones es que en la lista de reglas enriquecidas exista algo que bloqué la conexión. Se comprueba con "*firewall-cmd --list-rich-rules*" o iptables -L.

Paso 4

Después de asegurar que no hay bloqueos en la conexión, existe un archivo principal de configuración para ssh. Este se aloja en /etc/ssh/sshd-config. Contiene datos importantes como el puerto a conectarse por ssh, tarjeta de red para los que funciona el servicio y métodos para acceder al servidor, entre otras opciones. Decidimos modificar este archivo, autenticándonos a través de una clave pública-privada y no con contraseña. Los cambios fueron:

PermitRootLogin prohibit-password (Indica que el usuario root no puede iniciar con contraseña, pero si usando autenticación mediante una clave pública)

PubkeyAuthentication yes (La autenticación se realiza con un archivo que contiene la clave necesaria. Es la opción más segura)

PasswordAuthentication no (Controla si se puede iniciar sesión en el servidor mediante contraseña para los usuarios que quieren autenticarse. Al estar "No" queda inactiva esta opción.

KbdInteractiveAuthentication no (desactiva la autenticación interactiva basada en teclado)

```
#LoginGraceTime 2m
PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile      .ssh/authorized_keys

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no

# Change to no to disable s/key passwords
KbdInteractiveAuthentication no
```

Es importante hacer *"systemctl restart sshd"* para que el servicio cargue la configuración.

Generar claves pública-privada

Cliente:

Para generar un par de claves pública-privada, estando en el directorio del usuario(preferentemente), siendo este el cliente, se utiliza el comando *ssh-keygen* en la shell. Aparecerá un mensaje indicando que las claves se crearán en la ruta "~/.ssh". Se introduce el comando, presiona "Enter" si quiere dejar esa ruta por defecto. Puede introducir una nueva. Después se podrá asegurar con una capa más de protección, la cual es poner una "contraseña" al cifrado. Es opcional. Con esto, estarían las claves creadas donde se indicaron, siendo la .pub copiada en el servidor.

Servidor:

Del lado del servidor se crea la carpeta ".ssh" en el home del usuario con el que se inicia sesión. Dentro de la carpeta debe existir el archivo "authorized_keys" con la clave pública generada en el cliente.

```
C:\Users\mesac>ssh cjx@192.168.1.15
Web console: https://bitmate:9090/ or https://192.168.1.15:9090/

Register this system with Red Hat Insights: rhc connect

Example:
# rhc connect --activation-key <key> --organization <org>

The rhc client and Red Hat Insights will enable analytics and additional
management capabilities on your system.
View your connected systems at https://console.redhat.com/insights

You can learn more about how to register your system
using rhc at https://red.ht/registration
Last login: Tue Jul  1 09:56:20 2025 from 192.168.1.9
cjx@bitmate:~$ 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 noprefixroute
       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 08:00:27:31:1e:0d brd ff:ff:ff:ff:ff:ff
    altname enx080027311e0d
    inet 192.168.1.15/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s3
       valid_lft 79256sec preferred_lft 79256sec
    inet6 fe80::a00:27ff:fe31:1e0d/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
cjx@bitmate:~$
```

En caso de no tener la clave correspondiente, mostrará un error de permisos.

```
C:\Users\mesac>ssh cjx@192.168.1.15
cjx@192.168.1.15: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).

C:\Users\mesac>
```