Alberto Plaza Montes
Ingeniería de Servidores
(UGR)

Configuración básica SSH para servidores

Alberto Plaza Montes
Ingeniería de Servidores
(UGR)

Antes de empezar

En este documento se resumirá mediante un ejemplo práctico las bases de configuración SSH para computadores, que normalmente se aplican a servidores linux.

No utilizaremos un dispositivo real para ello, si no un hipervisor (VirtualBox), pero todo lo explicado aquí es extrapolable a una situación real sustituyendo los pasos realizados en el hipervisor por la correspondiente configuración en los equipos físicos.

Esta información proviene de lo aprendido en las prácticas de la asignatura Ingeniería de Servidores, impartida en el grado en Ingeniería informática en la Universidad de Granada.

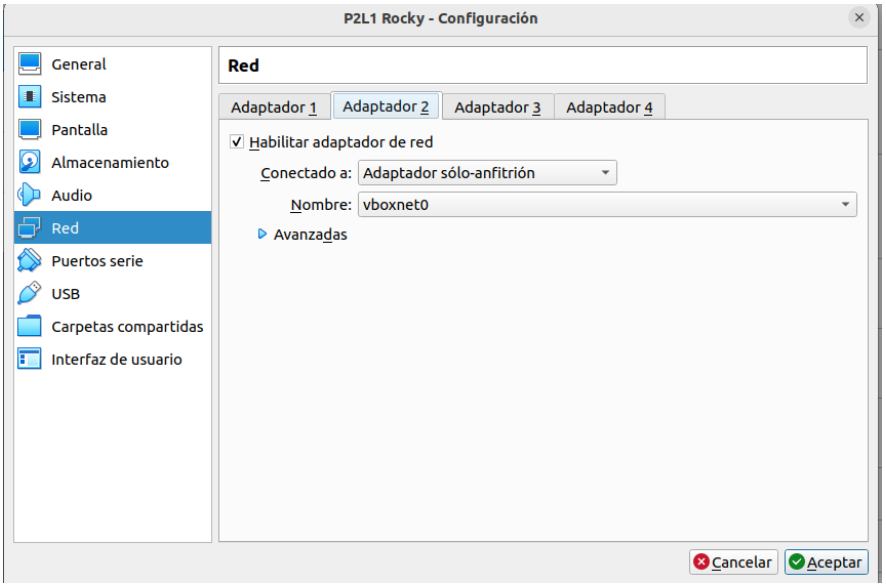
¿Qué es SSH?

SSH consiste de una forma muy básica, en la conexión remota entre servidores, principalmente nos permite controlar una terminal de un dispositivo desde otro dispositivo.

¿Qué realizaremos como ejemplo práctico?

Vamos a conectar dos servidores, uno en Rocky Server y otro en Ubuntu Server mediante conexión SSH, configurando los aspectos de seguridad básicos e imprescindibles para realizar la conexión de manera segura.

Configuración inicial y estructura

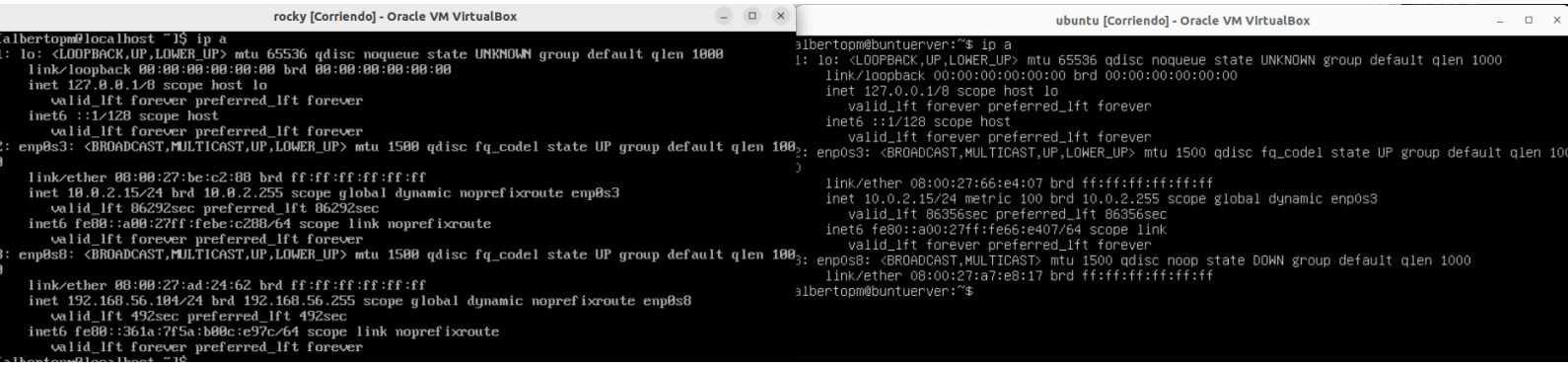


Partiremos con dos máquinas virtuales una en Rocky Server y otra en Ubuntu Server con sus respectivas instalaciones por defecto.

Respecto a la estructura de su conectividad, contarán con 2 interfaces de red. La NAT convencional en el adaptador 1, y una segunda conexión sólo anfitrión que añadiremos nosotros manualmente en el adaptador 2 mediante el hipervisor.

Nota: No olvidar confirmar que la casilla “cable conectado” está rellena en el apartado avanzadas.

Mediante el comando **ip a** podemos observar la actual estructura de redes



En esta imagen podemos ver las dos interfaces definidas en ambos servidores.

- Loopback: Esta interfaz siempre está.
- enp0s3: es la interfaz NAT del adaptador de red 1.
- enp0s8: Es la interfaz sólo anfitrión que añadimos en el adaptador de red 2.

De esta forma tenemos dos conexiones, una convencional, hacia internet, y otra como red interna, entre los dos computadores.

Asignación de IP

Como podemos ver en el comando ip a, la interfaz nos ha dado, en el caso de Rocky una dirección IP de forma automática que nosotros no queremos y el el caso de Ubuntu no ha especificado IP en las interfaces que hemos añadido, de modo que les asignaremos a cada uno una IP estática.

Para hacer las modificaciones puedes utilizar el editor de texto que quieras. En nuestro caso usaremos nano en ubuntu y vi en Rocky.

Alberto Plaza Montes
Ingeniería de Servidores
(UGR)

En Ubuntu Server:

Esto se hace desde el archivo de configuración que podemos encontrar en /etc/netplan/00-installer-config.yaml (después del netplan puedes hacer tabulador, es el único archivo en ese fichero).

De modo que para asignar la IP estática editaremos el fichero. Destacar que como es un .yaml, se rige por tabulaciones, por lo que estas son importantes.

Añadiremos la nueva interfaz, haremos que la ip sea estática (dhcp4: false) y le asignaremos la IP deseada (addresses: [IP/máscara]). El fichero quedaría tal que así:

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: false
      addresses: [192.168.56.105/24]
  version: 2
```

Importante introducir después el comando sudo netplan apply para que se realicen los cambios.

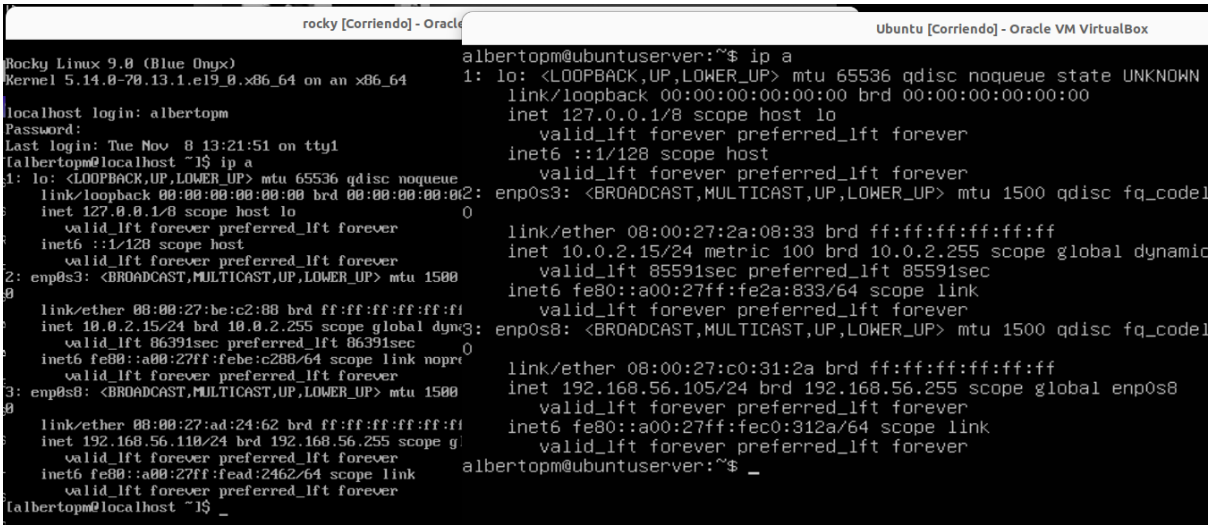
En Rocky Server:

En el caso de rocky no tiene un solo fichero para todas las interfaces, si no uno por interfaz. Podemos acceder a este mediante la siguiente ruta:
/etc/sysconfig/network-scripts/ifcfg-enp0s8

Aquí introduciremos la configuración que deseamos, entre ello lo principal que es la IP y la máscara.

TYPE=Ethernet
BOOTPROTO=none
NAME=enp0s8
DEVICE=enp0s8
ONBOOT=yes
IPADDR=192.168.56.110
NETMASK=255.255.255.0

Ahora podemos guardar el fichero y reiniciar.



De este modo tendríamos nuestras dos ips asignadas, y nuestras máquinas estarían conectadas mediante una red interna.

Podemos comprobar el correcto funcionamiento de la red lanzando pings mutuos a cada una de las IPs:

Instalación SSHD y activación automática

En ubuntu:

el SSH cliente viene instalado por defecto en ubuntu server, por lo que procederemos a instalar el SSH servidor:

sudo apt update

sudo apt install openssh-server

Ahora si tabulamos tras escribir ssh, podemos ver cómo aparece el demonio sshd, esto quiere decir que se ha instalado correctamente. Además, el servicio se ha iniciado de forma

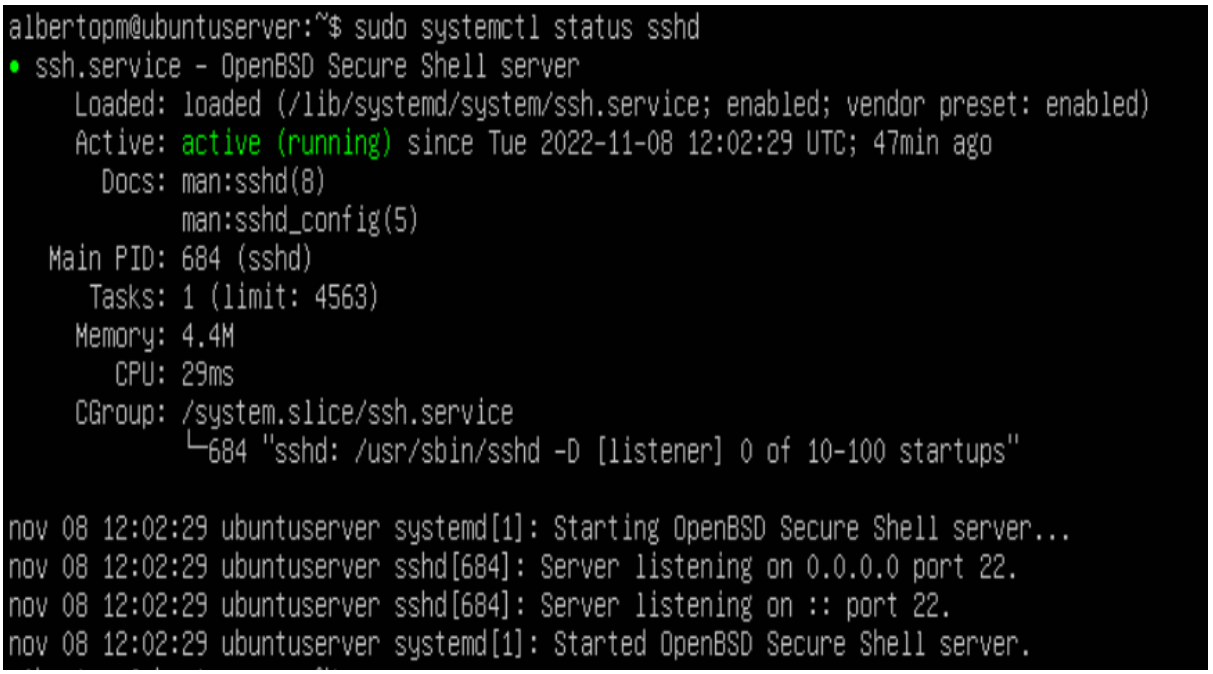
automática. Para que el servicio se inicie siempre de forma automática al arrancar el servidor:
sudo systemctl sshd start -permanent

En Rocky:

En Rocky Server ya está instalado sshd y su servicio está activo por defecto (algo muy a tener en cuenta al hacer un servidor en Rocky server).

Podemos comprobar el servicio en ambos servidores mediante el comando

sudo systemctl status sshd



Acceso SSH por defecto

Podemos comprobar el correcto funcionamiento del servicio utilizándolo entre los servidores. Para ello:

ssh <usuario>@<ip>

siendo usuario el usuario de la otra máquina al que quiero acceder, e ip su correspondiente IP.

Ejemplo de conexión:

En este caso nos hemos conectado a la terminal del servidor ubuntu server, de forma remota, desde el terminal de Rocky server.

Cuestiones de seguridad

Esta configuración básica presenta muchos inconvenientes de seguridad y hay ciertas modificaciones mínimas que son casi obligatorias para dotar al sistema de una mínima capa de seguridad, teniendo en cuenta lo peligroso que puede ser una conexión ssh no autorizada.

Puerto

SSH utiliza por defecto el puerto 22, cambiar el puerto que es necesario utilizar para la conexión es una buena medida de seguridad, pues así no es evidente el puerto que estás usando.

Esto se hace desde el archivo de configuración /etc/ssh/sshd_config. Para asignar un puerto descomentamos la linea #Port 22, y modificamos el 22 por el puerto que vayamos a usar. Es recomendable utilizar un puerto alto ya que estos suelen estar libres.

Si probamos la conexión de Rocky server a Ubuntu Server, esta funciona correctamente, sin embargo de Ubuntu server a Rocky Server no. Podría ser que el sistema operativo que estés utilizando tenga sus propias cuestiones exclusivas como Rocky, si no funciona correctamente el ssh, deberías revisarlas.

Cuestiones exclusivas Rocky Server

- ## SE linux

  Rocky Server tiene una capa extra de seguridad llamada SE Linux que nos está impidiendo realizar la conexión ssh. Para operar con Se Linux necesitamos el binario se manage, así que buscamos que paquete contiene:

  dnf provides semanage

  Y encontramos que el paquete policycoreutils-python-utils lo contiene, de modo que lo instalamos

  sudo dnf install policycoreutils-python-utils

  Con sudo semanage port -l obtenemos una lista de los puertos asignados, de modo que buscamos los asignados a ssh con un grep:

  sudo semanage port -l | grep “ssh”

  Y vemos como efectivamente solo está asignado el puerto 22. Para añadir el puerto que nosotros queremos asignar:

  sudo semanage port -a -t ssh_port_t -p tcp 22022

  -a: añadir
  -t <nombre puerto>: el puerto el concreto
  -p <modelo>: puerto

- Firewall

  A diferencia de Ubuntu Server, que tiene su firewall desactivado por defecto, Rocky Linux lo tiene activado.

  Para listar las reglas firewall:

  sudo firewall-cmd –list-all

  Y como podemos observar, nuestro puerto 22022 no es uno de los permitidos, así que lo añadimos:

  sudo firewall-cmd –add-port=22022/tcp --permanent

  Si queremos un extra de seguridad respecto al servidor en Ubuntu Server, podríamos activar el firewall con el UFW y añadirle el puerto 22022.

  sudo ufw enable

  sudo ufw allow 22022

Usuario root

La posibilidad del acceso al usuario root es también un claro punto de vulnerabilidad, pues una persona no autorizada al acceso, con algunos conocimientos podría saber de la existencia de este usuario en nuestro dispositivo. Además es el superusuario lo cual puede ser peligroso, por lo que desactivaremos la posibilidad de acceder al usuario root.

Para ello en el archivo /etc/ssh/sshd_config descomentamos la opción PermitRotLogin, y le pondremos no.

Nota: Recordar que es importante restaurar el servicio en los servidores ssh para que se realicen los cambios (sudo systemctl restart sshd).

Fingerprint ssh

Las contraseñas, en general, siguen siendo un punto vulnerable, pues estas pueden ser descubiertas de muchas formas. Para ello utilizamos las fingerprints.

Se trata de un método de acceso por autorización, es decir, sin contraseña. Lo que sucede es que el equipo 1 al intentar conectarse al equipo dos, cifra un paquete con la llave privada y lo envía. El equipo 2 puede descifrar este paquete ya que posee la llave pública del Equipo 1, y por tanto se autoriza el acceso del equipo 1 al equipo 2 sin necesidad de introducir la contraseña.

ssh utiliza un cifrado simétrico, mediante la fingerprint estableceremos un cifrado asimétrico, teniendo una llave privada (que tiene siempre la llave cliente) y una llave pública, que copiaremos en el servidor que queremos que pueda conectarse.

Para generar estos dos archivos:

ssh_keygen

- Dejamos el archivo por defecto.
- No le ponemos contraseña (aunque podría ser una capa extra de seguridad, si se nos extraviese la llave podría ser muy peligroso ya que podría accederse al servidor).

Podemos observar como se ha creado tanto la llave pública como la llave privada con ls -la .ssh/

Para envíar la llave tenemos un script predefinido que lo hace por nosotros:

ssh-copy-id -i .ssh/id_rsa.pub -p <puerto> <usuario>@<ip>

ssh-copy-id -i .ssh/id_rsa.pub -p 22022 albertopm@192.168.56.105

Tras esto podemos deshabilitar el acceso con contraseña para que así solo pueda acceder por ssh mediante la fingerprint.

En el fichero etc/ssh/sshd_config ponemos password authentication en no.

Finalmente podemos hacer conexiones sin introducir la contraseña de un equipo a otro.

Ejemplo de conexión automática:

ssh albertopm@192.168.56.105 -p 22022