

Trabajo Práctico Computación Aplicada

Configuración del entorno:

Blanquear clave root: en GRUB, tocar la letra e, agregar al final de la línea de boot de Linux lo siguiente: “init=/bin/bash/” y tocar F10 para ejecutar. Ya como root, usar “`mount -o remount,rw /`”. Ahora sí, usar el comando “`passwd root`” y cambiar la clave a “palermo”.

Iniciar con el comando “`exec /sbin/init`”

Con el comando “`hostnamectl set-hostname TPServer`”, fijo el nombre del host a TPServer

Servicios:

SSH: como daba error al hacer apt update, y después de confirmar que no es un error de conectividad con internet o DNS (usando `ping 8.8.8.8` y `ping deb.debian.org`, me fijo en el `/etc/apt/sources.list`, donde los archivos estaban mal:

```
TPVMCA [Corriendo] - Oracle VM VirtualBox
1 deb http://cdn2-fastly.deb.debian.org/debian/ bullseye main contrib non-free
2 deb http://cdn2-fastly.deb.debian.org/debian/ bullseye-updates main contrib non-free
~ 3 deb http://cdn2-fastly.debian.org/ updates main contrib non-free
```

Los cambio a:

```
1 deb http://deb.debian.org/debian/ bullseye main contrib non-free_
2 deb http://deb.debian.org/debian/ bullseye-updates main contrib non-free_
3 deb http://security.debian.org/debian-security bullseye-security main contrib non-free
```

Instalo ssh con “`apt install openssh-server`”

edito el archivo de configuración con “`vi /etc/ssh/sshd_config`” las siguientes líneas:

PermitRootLogin yes

PasswordAuthentication yes (temporalmente, explico más adelante)

PubkeyAuthentication yes

reinicio ssh con “`systemctl restart ssh`”

A través de WinSCP, me conecto a través de usuario y contraseña del root (para esto deje en yes la opción de PasswordAuthentication) y subo la clave pública a

`/.ssh/authorized_keys`. Creo un archivo sin extensión que contiene la clave privada (lo

guardo como `id_rsa`) y, desde CMD de Windows, me conecto a través de la clave privada/pública usando:

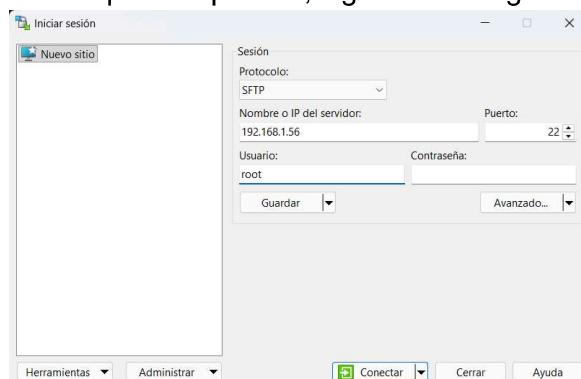
“`ssh -i “C:\Users\.....\id_rsa` root@ip” y logro la conexión.

WEB: hago “`apt update`” y “`apt upgrade`” para tener todos los paquetes actualizados para poder instalar apache2 y php con “`apt install apache2 php libapache2-mod-php -y`”.

Una vez descargados, verifico si están instalados haciendo “`systemctl status apache`” y “`php -v`”.

Vuelvo a usar WinSCP, a través de la clave privada/pública, siguiendo el siguiente procedimiento:

Avanzado, SSH, autenticación,
archivo de clave privada
(extensión .ppk), Aceptar



A través de la conexión, subo los archivos index.php y logo.png al directorio /root. De vuelta en la Máquina Virtual, creo el directorio /www_dir con “`mkdir -p /www_dir`” y muevo ambos archivos ahí. edito con vi el archivo de configuración

`/etc/apache2/sites-available/000-default.conf` y le pongo lo siguiente:

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /www_dir

    <Directory /www_dir>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Así configurando DocumentRoot a /www_dir

Ahora le doy los permisos necesarios al directorio y sus archivos

“`sudo chown -R www-data:www-data /www_dir`”

“`sudo chmod -R 755 /www_dir`”

“`chmod 644 /www_dir/index.php /www_dir/logo.png`”

y reinicio apache2 con “`systemctl restart apache2`” para actualizar los cambios

BASE DE DATOS: hago “`apt update`” para luego instalar la base de datos con “`apt install mariadb-server -y`”. Verifico si está corriendo con “`systemctl status mariadb`”

Usando WinSCP de la misma manera que especifiqué previamente, transfiero el archivo db.sql al servidor. Uso “`mysql </root/db.sql`” (que es donde guarde el archivo) para ejecutarlo.

Al abrir http://IP_SERVIDOR/index.php me aparece en blanco, por lo que agrego al principio del archivo .php lo siguiente:

```
error_reporting(E_ALL);
ini_set('display_errors', 1);
```

para que me aparezca cuál es el error. Al entrar nuevamente en el hhtp me aparece:

Fatal error: Uncaught Error: Class 'MySQLi' not found in /www_dir/index.php:13 Stack trace:
#0 {main} thrown in /www_dir/index.php on line 13

Instalo la extensión correspondiente con: “`sudo apt install php-mysql -y`” y reinicio apache: “`systemctl restart apache2`”.

Esto me permite ver los datos correspondientes a la base de datos en el http.

Configuración de red:

Para fijar la IP estática, hago lo siguiente: edito el archivo `/etc/network/interfaces` usando nano y agrego lo siguiente:

```
auto enp0s3
iface enp0s3 inet static
    address 192.168.1.56
    netmask 255.255.255.0
    gateway 192.168.1.1
```

reemplazando la configuración por dhcp (que le da la IP de manera automática al servidor) reinicio el servicios de network con “`systemctl restart networking`”

Almacenamiento:

En la pestaña de configuración de almacenamiento, creo un disco nuevo con 10GB de espacio y lo conecto a la máquina virtual. Ya en la terminal, verifico que está conectado con el comando “`lsblk`” y veo que está asignado bajo el nombre sdc. Usando “`fdisk /dev/sdc`” creo 2 particiones, una de 3GB y otra de 6GB. Verifico con “`lsblk`” que fueron creadas con éxito. Ahora le doy formato ext4 con “`mkfs.ext4 /dev/sdc1`” y “`mkfs.ext4 /dev/sdc2`”. Creo el directorio `/backup_dir` (`/www_dir` ya lo tengo creado) y los monto de forma permanente editando el archivo `/etc/fstab`. Obtengo los UUIDs de cada una de las particiones con “`blkid /dev/sdc1-2`” y las agrego, junto con las extensiones, directorios de montaje y más info en el archivo fstab. Guardo y cierro y reinicio la máquina para ver si se montaron correctamente. Esto lo compruebo usando “`df -h | grep dir`”.

Como no puedo crear archivos en el directorio `/proc`, lo que hago es crear una copia de `/proc/partitions` en `/root`, usando “`cat /proc/partitions > /root/particion`”.

Scripts:

creo el directorio `/opt/scripts` y el archivo `backup_full.sh` con el script pedido.

Ahora viene la parte de automatizar el código, para eso usamos el comando “`crontab -e`” y agregamos al final de archivo lo siguiente:

```
# Backup diario de /var/log a las 00:00
0 0 * * * /opt/scripts/backup_full.sh /var/log /backup_dir
# Backup de /www_dir lunes, miércoles y viernes a las 23:00
0 23 * * 1,3,5 /opt/scripts/backup_full.sh /www_dir /backup_dir
```

Para no tener que esperar a la hora de backup automático, uso “`run-parts --test /etc/cron.daily`” para realizar el backup y confirmo que crea el archivo .tar.gz

GTIHUB:

Creo una nueva clave ssh con

`"ssh-keygen -t ed25519 -C "tu_correo_de_github@example.com"`

Me pregunta donde quiero guardarla, la guardo en el directorio por defecto
`/root/.ssh/id_ed25519`

Ahora levanto el agente ssh con “`eval "$(ssh-agent -s)"`” y agrego la clave con
`“ssh-add ~/.ssh/id_ed25519”`

Hago un cat para ver la clave pública y poder copiarla “`cat ~/.ssh/id_ed25519.pub`”
Entro a Github, voy a Settings del usuario, SSH and GPG keys, New SHH key, le pongo de nombre “TP” y pego la clave que había copiado previamente.

Una vez generada la ssh, pruebo la conexión desde la terminal con “`ssh -T git@github.com`”. Me pregunta si quiero seguir con la conexión y pongo yes.

Logro ver que la conexión fue exitosa usando porque me aparece lo siguiente:
Hi tomassmusso! You've successfully authenticated, but GitHub does not provide shell access.

Entregables:

Una vez conectado a Github, paso a realizar los comprimidos de los directorios a entregar. Instalo tar, para eso hago un “`apt update`”, “`apt upgrade`” y después “`apt install tar -y`”.

Creo un directorio con “`mkdir /root/entregables`” y uso los siguientes comandos:

`tar -czf /root/entregables/root.tar.gz /root`

`tar -czf /root/entregables/etc.tar.gz /etc`

`tar -czf /root/entregables/opt.tar.gz /opt`

`tar -czf /root/entregables/www_dir.tar.gz /www_dir`

`tar -czf /root/entregables/backup_dir.tar.gz /backup_dir`

`tar -czf /root/entregable_tp/var.tar.gz /var`

Nota: el directorio `/proc` no se puede comprimir ya que es un sistema de archivos virtual, pero como el archivo `/root/particiones` está dentro del comprimido de `/root`, ya es suficiente para la entrega.

Como el directorio `/var` es muy grande para subir al repositorio, lo divido en partes de 100MB.

Subida de archivos:

para subir los archivos, primero clono el repositorio de github con “`git clone git@github.com:tomassmusso/tp_computacion.git`”, una vez clonado, uso cd para moverme a la copia del repositorio. Creo una carpeta `/entregables` con “`mkdir entregables`” y luego uso “`cp /root/entregables/* /root/tp_computacion/entregables`” para copiar todos los archivos. Uso “`git add .`”, “`git commit -m`” y “`git push origin main`” para pushear los archivos y confirmar los cambios. Hago el mismo procedimiento, subiendo el script `backup_full.sh` y una copia de lo modificado con crontab -e a una carpeta llamada `/adicionales`.