Ubuntu Server

Configuraciones Iniciales

Requisitos del sistema

Ubuntu 22.04 Server (Requisitos mínimos)

CPU	RAM	Disco Duro
1 GHz	1 GB	2.5 GB

ACTUALIZACIÓN DE PAQUETES

1) Actualizar índice de paquetes:

    $ **sudo apt update**

2) Actualizar los paquetes instalados:

    $ **sudo apt upgrade**

3) Actualizar automáticamente paquetes críticos:

    $ **sudo dpkg-reconfigure -plow unattended-upgrades**

REPOSITORIOS

$ **sudo nano /etc/apt/sources.list**

main: Software libre mantenido por Canonical (Repositorio principal).

universe: Software libre mantenido por la comunidad.

restricted: Drivers privativos mantenido por Canonical.

multiverse: Software restringido por copyright mantenido por la comunidad.

Ejemplo de contenido:

```
deb http://es.archive.ubuntu.com/ubuntu focal universe
```

nota: Para desactivar repositorio añadir el símbolo de comentario “#”.

RED: CONFIGURACIÓN TEMPORAL

Identificar tarjetas de red ethernet:

```
$ ip a
```

Asignación de dirección IP temporal:

```
$ sudo ip addr add 192.168.1.10/24 dev eth0
$ ip address show dev eth0
```

Activar o desactivar en enlace:

```
$ ip link set dev eth0 up
$ ip link set dev eth0 down
```

RED: CONFIGURACIÓN TEMPORAL (II)

Puerta de enlace temporal:

```
$ sudo ip route add default via 10.0.2.2
$ ip route show
```

RED: CONFIGURACIÓN PERMANENTE

$ **sudo nano /etc/netplan/00-installer-config.yaml**

```
# Ejemplo I de configuración
network:
  ethernets:
    eth0:
      addresses: [192.168.1.100/24]
      gateway4: 10.0.2.2
      nameservers:
        addresses: [8.8.8.8,8.8.4.4]
      dhcp4: no
  version: 2
```

```
# Ejemplo II de configuración
network:
  ethernets:
    eth0:
      dhcp4: true
    eth1:
      addresses: [192.168.1.100/24]
      dhcp4: no
  version: 2
```

Aplicar cambios con la orden:
sudo netplan apply

NOMBRE DE EQUIPO Y DE HOSTS

1) Cambiar el nombre del servidor:

```
$ sudo hostnamectl set-hostname u-server
```

2) Modificar el fichero cloud.cfg para no perder el nombre al reiniciar:

```
$ sudo nano /etc/cloud/cloud.cfg

    preserve_hostname: true   # Modificar propiedad de false a true.
```

3) Modificar el fichero de hosts:

```
$ sudo nano /etc/hosts

    127.0.0.1    localhost
    127.0.1.1    u-server
```

Fecha y hora: NTP (Network Time Protocol)

1) Listar las zonas horarias

```
$ timedatectl list-timezones
```

2) Establecer la zona horaria

```
$ sudo timedatectl set-timezone Europe/Madrid
```

3) Activamos sincronización NTP

```
$ sudo timedatectl set-ntp on
```

OPENSSH (SECURE SHELL): ACCESO REMOTO

Instalación:

```
$ sudo apt install openssh-client
$ sudo apt install openssh-server
```

Configuración:

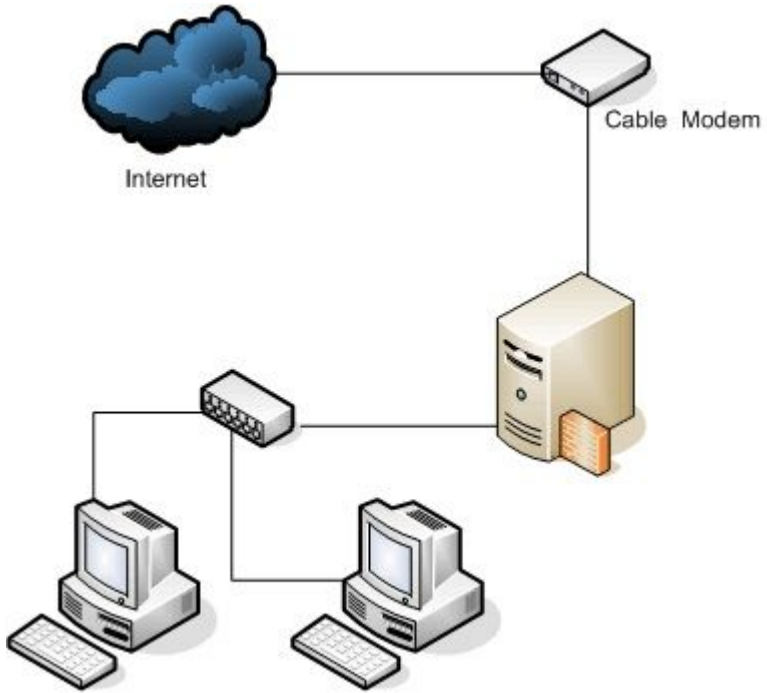
```
$ sudo nano /etc/ssh/sshd_config

    Port 22
    AllowUsers ana juan
    PermitRootLogin no
    Banner /etc/issue.net

$ sudo systemctl restart ssh
```

ssh -p 22 usuario@192.168.0.1
(**-p** es opcional si el puerto es el 22)

Acceso sin contraseña al servidor:
1) Si no la tenemos (*id_rsa*), generamos la clave RSA en el equipo cliente:
```
cd ~/.ssh
ssh-keygen -t rsa
```
2) Desde el cliente copiamos la clave pública al servidor:
```
ssh-copy-id -i .ssh/id_rsa.pub
usuario@192.168.1.100
```

EJERCICIO (I)

1. Realiza la instalación de Ubuntu Server sobre una máquina virtual con 2GB de memoria y 20GB de disco duro.
2. El nombre del servidor será **u-server**.
3. Instala, de forma manual, todas las actualizaciones pendientes del sistema.
4. Configura el servidor con la dirección IP fija **192.168.1.100/24** para la conexión con la red local.
5. Configura el servidor con una dirección IP obtenida por **DHCP** para la conexión a Internet.
6. Configura el protocolo de sincronización de la fecha y hora.
7. Realiza los ajustes adecuados para administrar el servidor desde un ordenador remoto.



USUARIO ROOT

La cuenta root está desactivada de forma predeterminada por seguridad. Para activar la cuenta de root de forma temporal:

```
$ sudo su
```

Aunque no es recomendable, podemos también activar la cuenta de root de forma permanente si le damos una clave:

```
$ sudo passwd root
```

Para volver a desactivar la cuenta de root:

```
$ sudo passwd -l root
```

Prompt:

root@server:~#

usuario@server:~$

GESTIÓN DE USUARIOS

Crear usuarios:

```
$ sudo adduser juan --ingroup users
```

Eliminar usuarios:

```
$ sudo deluser luis
```

Bloquear una cuenta (**l**ock) temporalmente:

```
$ sudo passwd -l juan
```

Desbloquear una cuenta (**u**nlock):

```
$ sudo passwd -u juan
```

GESTIÓN DE GRUPOS

Crear un grupo:

```
$ sudo addgroup ventas
```

Borrar un grupo:

```
$ sudo delgroup ventas
```

Asignar un usuario a un grupo:

```
$ sudo adduser juan ventas
```

Permisos adminstrativos:

```
$ sudo adduser juan sudo
```

PERMISOS

Cambiar los permisos de acceso a un fichero o directorio:

```
$ chmod 764 file.txt
```

Cambia el propietario de un fichero o directorio:

```
$ chown -R juan:users /directorio
```

Número	Binario	Lectura (*r*)	Escritura (*w*)	Ejecución (*x*)
0	000	✗	✗	✗
1	001	✗	✗	✓
2	010	✗	✓	✗
3	011	✗	✓	✓
4	100	✓	✗	✗
5	101	✓	✗	✓
6	110	✓	✓	✗
7	111	✓	✓	✓

EJERCICIO (II)

1. Crea los grupos **profesores** y **alumnos**.
2. Crea los usuarios **paula** y **pablo** pertenecientes al grupo de profesores.
3. Crea los usuarios **ana** y **alberto** pertenecientes al grupo de alumnos.
4. En srv crea la carpeta **profesores** para compartir documentos. Solo los profesores podrán acceder y modificar el contenido de este directorio.
5. En srv crea la carpeta **alumnos** para compartir documentos. Además de los alumnos, los profesores también podrán acceder pero solo con permiso de lectura.

PROGRAMAR ACTUALIZACIONES

Crear una tarea programada:

```
$ sudo crontab -e
```

```
00 3 * * * root apt update -y
```

Todos los días a las 3:00 AM

Asume YES a cualquier pregunta

Línea en el fichero crontab:

```
#  Minuto   Hora   Dia-Mes   Mes   Dia-Semana   Usuario   Comando
#  0-59     0-23   1-31      1-12  0-6 (0 Domingo)
```

AÑADIR UN DISCO DURO

1) Obtener información y localizar el nuevo disco duro:

```
# lshw -C disk
```

2) inicializar y particionar:

```
# fdisk /dev/sdb
```

n (add a new partition)

p (primary)

1 (partition number)

w (write changes)

3) Formatear la partición:

```
# mkfs -t ext4 /dev/sdb1
```

4) Crear el punto de montaje y montar la nueva partición:

```
# mkdir /mnt/backup
```

```
# mount /dev/sdb1 /mnt/backup
```

5) Averiguar UUID:

```
# blkid
```

6) Configurar el sistema para montar el volumen al inicio:

```
# nano /etc/fstab
```

Añadimos la línea usando el UUID obtenido

```
UUID=n...n /mnt/backup ext4 defaults 0 2
```

COPIAS DE SEGURIDAD CON SHELL SCRIPTS

1) Crear el script **backup.sh**:

```
#!/bin/bash

# Directorio a salvar y destino.
orig="/home"
dest="/mnt/backup"

# Nombre del fichero.
dia=$(date +%A)
fichero="Copia-$dia.tgz"

# Realizamos la copia de seguridad.
echo "Realizando backup de $orig en $dest/$fichero"
tar czf $dest/$fichero $orig
echo "Copia finalizada"

# Listamos los ficheros en $dest y comprobamos
# el tamaño de los ficheros.
ls -lh $dest
```

2) Dar permisos de ejecución:

```
$ chmod u+x backup.sh
```

3) Ejecutar el script:

```
$ sudo ./backup.sh
```

4) Programar el scripts:

```
$ sudo crontab -e
```

Añadir la línea:

```
0 0 * * * bash /usr/local/bin/backup.sh
```

COPIAS DE SEGURIDAD CON DUPLICITY (I)

Instalación:

```
$ sudo apt install duplicity
```

Crear una copia. En la primera ejecución se realizará una copia completa, el resto serán incrementales:

```
$ duplicity /datos/ file:///copia/
```

Forzar una copia completa:

```
$ duplicity full /datos/ file:///copia/
```

Lista el contenido de los backup:

```
$ duplicity list-current-files file:///copia/
```

COPIAS DE SEGURIDAD CON DUPLICITY (II)

Registros de todos los backup realizados:

```
$ duplicity collection-status file:///copia/
```

Recuperación de ficheros:

```
$ duplicity restore file:///copia/ /datos/
```

Notas:

- Se puede sustituir **file** por otros protocolos como **ssh** o **ftp** para hacer las copias en un servidor externo.

EJERCICIO (III)

1. Añade un **nuevo disco duro** a nuestro servidor para realizar copias de seguridad.
2. Realiza una copia de seguridad completa de los directorios **alumnos** y **profesores**.
3. Programa las copias de seguridad para que se realice una copia incremental diariamente.

CASO PRÁCTICO CON DUPLICITY

1. Realiza una copia de seguridad completa de los directorios **alumnos** y **profesores** en un servidor externo utilizando el protocolo **ftp**.
2. Automatiza la copia de seguridad para que se ejecute todos los días a las 12:00h

CASO PRÁCTICO CON DUPLICITY (SOLUCIÓN)

1) Instalamos los paquetes necesarios:

```
$ sudo apt install duplicity ncftp
```

2)Creamos el fichero y asignamos permisos:

```
$ sudo chmod 700 backup.sh
```

3)Contenido del fichero:

```
export PASSPHRASE=Cadena-de-cifrado
export FTP_PASSWORD=Clave-del-servidor-FTP
duplicity /srv/alumnos ftp://usuarioFtp@ftp.dominio.com/backup/alum
unset PASSPHRASE
unset FTP_PASSWORD
```

4) Automatizar añadiendo en el root crontab:

```
0 0 * * * /root/scripts/backup.sh >>/var/log/duplicity/alum.log
```