

Práctica 2

Nota → Asegurar acceso a internet

Ubuntu Server

- `sudo vi /etc/netplan/00-installer-config.yaml`
- Encima de versión poner addresses: [192.168.56.105/24] a la altura de dhcp4
- `sudo netplan apply`

CentOs

- `sudo vi /etc/sysconfig/network-scripts/ifcfg-enp0s3` y poner ONBOOT=yes
- `reboot`

SSH

Ubuntu Server

Configuración de ssh e inhabilitación de root

`sudo apt-get install openssh-server`

`ps -Af | grep sshd` (Activar si no lo está)

`ssh localhost + yes`

`sudo vi /etc/ssh/sshd_config` → **PermitRootLogin no** (De serie solo está a 'no' con pass)

`sudo systemctl restart sshd`

`ssh root@192.168.56.105` (Desde nuestro pc, no debería dejarnos)

Cambiar los puertos

`sed s/'#Port 22'/'Port 22022'/ -i /etc/ssh/sshd_config` (En definitiva cambiar eso)

`sudo systemctl restart sshd`

`ssh usuario@192.168.56.105 -p 22022` para conectarnos a partir de ahora y en Ubuntu sí que podemos acceder desde fuera en este punto.

Firewall

En este caso debemos activar el firewall y añadirle 22022, no viene activado de serie.

`sudo ufw status`

`sudo ufw enable`

Ahora no podremos conectar desde 22022.

`sudo ufw allow 22022`

Ahora sí podemos acceder desde 22022.

CentOs

Configuración de ssh e inhabilitación de root

```
sudo yum install openssh-server (Ya debería venir)
ps -Af | grep sshd
ssh localhost + yes
sudo vi /etc/ssh/sshd_config → PermitRootLogin no (De serie sí se accede con pass)
sudo systemctl restart sshd
```

Cambiar los puertos

```
sudo vi /etc/ssh/sshd_config → Cambiamos #Port 22 por Port 22022
sudo systemctl restart sshd
sudo journalctl -xe (Nos dice qué sucede)
```

```
sudo yum install policycoreutils-python-utils.noarch
sudo semanage port -l | grep ssh
sudo semanage port -a -t ssh_port_t -p tcp 22022 (Tarda un poco)
sudo semanage port -l | grep ssh (Para comprobar)
sudo systemctl restart sshd
```

ssh usuario@192.168.56.105 -p 22022 para conectarnos a partir de ahora.

En este punto no podremos acceder desde otra maquina parece, miramos fw.

Firewall

```
sudo firewall-cmd - -add-port 22022/tcp - -permanent
sudo firewall-cmd - -add-port 22022/tcp
sudo firewall-cmd - -reload
```

Ya sí podemos conectarnos a través de 22022.

Fail2Ban

CentOs

dnf search fail2ban (No debería encontrarlo)

dnf search epel

sudo dnf install epel-release

dnf search fail2ban (Sí debería encontrarlo)

sudo dnf install fail2ban

systemctl status fail2ban

sudo systemctl enable fail2ban

sudo systemctl start fail2ban

systemctl status fail2ban

cat /etc/fail2ban/jail.conf (CUIDADO, NO modificar este)

sudo cp -a /etc/fail2ban/jail.conf /etc/fail2ban/jail.local

sudo vi /etc/fail2ban/jail.local → **Buscamos ssh después de JAILS**

- Ponemos enabled=true debajo de port=ssh

sudo fail2ban-client status sshd (No debería encontrar el jail)

sudo systemctl restart fail2ban.service

sudo fail2ban-client status sshd (Ahora sí)

Nota → Se puede configurar maxretry, bantime, etc en jail.local.

sudo vi /etc/fail2ban/jail.local → **Modificar donde antes port=22022**

systemctl restart fail2ban

sudo fail2ban-client set sshd unbanip xxx.xxx.xxx.x (Desbanear)

Pila Lamp

CentOs

```
sudo dnf install httpd (Apache)
systemctl status httpd
sudo systemctl enable httpd
sudo systemctl start httpd
systemctl status httpd
curl localhost (Para comprobar)
```

```
sudo dnf install php
php -a (Para comprobar)
```

```
sudo dnf install mariadb (mysql-server mysql-client en ubuntu)
systemctl status mariadb (Eso va a fallar porque solo hemos instalado el cliente)
sudo dnf install mariadb-server
systemctl status mariadb
sudo systemctl enable mariadb
sudo systemctl start mariadb
systemctl status mariadb
```

```
mysql_secure_installation → Enter, pass nueva, Y...Y
mysql -u root -p (Para conectarnos desde el propio CentOS)
```

```
sudo firewall-cmd - -add-port 80/tcp - -permanent (No conectaba desde fuera)
sudo firewall-cmd - -add-port 80/tcp
sudo firewall-cmd - -reload
```

Colocamos un archivo php en /var/www/html/ y lo llamamos index.php:

```
<?php
    $link=mysqli_connect("127.0.0.1", "root", "ISE", "mi_bd");

    if(!$link){
        die('No se ha podido conectar de forma exitosa a la DB');
    }

    echo 'Conexión exitosa a la DB';
    mysqli_close($link);

    phpinfo();
?>
```

mysql -u root -p

CREATE DATABASE mi_bd; (No lo hicimos en clase, quitar también del index)

sudo vi /etc/httpd/conf/httpd.conf → **añadimos *.php en <ifModule dir_module>**
comprobar navegador (Fallo)

sudo systemctl restart httpd
comprobar navegador (Fallo)

php /var/www/html/index.php (Debería dar error)

sudo dnf search php | grep mysql

sudo dnf install php-mysqldb

php /var/www/html/index.php (Ahora sí)
comprobar navegador (Fallo)

sudo getsebool -a | grep httpd → **poner httpd_can_network_connect_db de off a on:**

sudo setsebool -P httpd_can_network_connect_db=on (Tarda un poco)

sudo getsebool -a | grep httpd_can_network_connect_db

sudo systemctl restart httpd
comprobar navegador (Correcto)

Clave privada

ssh-keygen

ssh-copy-id 192.168.56.110 -p 22022 (Para enviarla al otro lado)

sudo vi /etc/ssh/sshd_config → Poner **Password Authentication no**

Git

git init → Toma instantánea

Editar ~/.gitconfig para añadir user

git log → Muestra log de commits (git log master → Muestra commits de master)

git status → Muestra los cambios que no están en stage y los archivos no seguidos

git add . → Sube archivos a stage

git commit -m "Mensaje" → Crea el commit

git branch -M main → Crea la rama main

git remote add origin https://github.com/ArturoAcf/Sistemas-Concurrentes-y-Distribuidos.git

git pull -rebase origin main → Actualiza el repositorio con el de github

git push origin main → Sube a github el repositorio con lo añadido

git diff → Mostrar diferencias entre directorio de trabajo y stage

git diff -staged → Mostrar diferencias entre stage y repositorio

git diff HEAD → Mostrar diferencias entre directorio de trabajo y repositorio

git reset 7_primeros_caracteres_del_commit_SHA → Devuelve los cambios indicados actualizando además HEAD

git restore * → Pone todo como esté el HEAD (Hacer si se hace reset)

git revert <commit> → Desaplica al directorio de trabajo los cambios hechos en commit creando uno nuevo (-n para evitar esto último)

git branch → Muestra la rama actual

git checkout <rama> → Nos mueve a esa rama (-b para crear e ir)

git stash → Salva el estado del directorio sin necesidad de hacer commit

git stash apply → Aplica el último stash creado

git stash apply <ns> → Aplica ns

git stash list → Muestra los disponibles

git merge → Une ramas

git init -bare → Útil para crear un repositorio en un servidor centralizado para compartir cambios que se vayan haciendo

Si es local: git clone -bare repo repo.git si es local + scp -P 22022 -r repo.git https:...
+ git remote add origin ssh://192...

git clone ubi_repo → Crea una copia local del repositorio indicado

Comandos

dd → Copiar bit a bit: **dd if=/dev/sda of=/dev/sdb** (input file – output file)
dd if=/dev/sda of=~/hdadisk.img (Imagen de sda)
dd if=hdadisk.img of=/dev/sdb (Recuperar imagen)

cpio → Copiar y empaquetar: **(a) cpio -o > archivo.cpio / (desde) cpio -idv < archivo.cpio**

tar → Compimir: **tar -cvzf directorio.tar.gz directorio**
tar -xvf directorio.tar

cp → Copiar: **cp (-a) algo algo**

rsync → Sincronizar origen con destino: **rsync origen destino**
rsync -avize “ssh -p 22022” prueba/* alcaa@192.168.56.110:/home/...