

Práctica 8. Almacenamiento

Objetivo: En esta práctica vamos a evaluar distintos recursos de almacenamiento cliente/servidor utilizamos en CPD.

Presentar un documento pdf con la siguiente información:
-Capturas personalizadas de cada apartado

Desarrollo:

Primero iniciamos una configuración con 2 máquinas ubuntu que puedan verse entre sí.

```
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/focal64"
  config.vm.provision "shell", inline: <<-SHELL
    sed -i 's/PasswordAuthentication no/PasswordAuthentication yes/' /etc/ssh/sshd_config
    systemctl restart sshd.service
    echo "192.168.12.11 nodo1" >> /etc/hosts
    echo "192.168.12.12 nodo2" >> /etc/hosts
  SHELL

  config.vm.define :nodo1 do |ub_config|
    ub_config.vm.hostname = "nodo1.vm"
    ub_config.vm.network "private_network", ip:"192.168.12.11"
    ub_config.vm.provider :virtualbox do |vb|
      vb.name = "nodo1"
      vb.customize ["modifyvm", :id, "--memory", "768"]
      vb.customize ["modifyvm", :id, "--cpus", "1"]
    end
  end

  config.vm.define :nodo2 do |ub_config|
    ub_config.vm.hostname = "nodo2.vm"
    ub_config.vm.network "private_network", ip:"192.168.12.12"
    ub_config.vm.provider :virtualbox do |vb|
      vb.name = "nodo2"
      vb.customize ["modifyvm", :id, "--memory", "768"]
      vb.customize ["modifyvm", :id, "--cpus", "1"]
    end
  end
end
```

Supongamos que *nodo1* es servidor y *nodo2* va a ser cliente.

Antes de levantar la máquina podemos comprobar si el box *ubuntu/focal64* está actualizado

```
vagrant box update
```

Levantamos la configuración

```
vagrant up
```

Activamos la conexión ssh para que se pueda acceder directamente de la máquina 2 a la máquina 1.

Creamos la pareja de claves en el nodo2

```
ssh-keygen -t rsa
```

copiamos las claves al nodo1

```
ssh-copy-id nodo1
```

1. Almacenamiento Rsync con SSH

Creamos una carpeta en el cliente (test1)

Sincronizamos la carpeta

Yo considero que test1 está en nodo1 y nodo2 es el cliente, al cual quiero copiar la cosas

```
rsync -avz test1 vagrant@nodo1:.
```

nodo2

Realizamos algún cambio

Resincronizamos. Comprobamos que los cambios se han actualizado en el servidor.

2. Servidor GIT

Crear repo local: *nodo2*

```
mkdir devel/test2  
cd devel/test2  
git init  
>README  
git add *  
git commit -m "mi 1er commit"
```

Crear repositorio remoto: *nodo1*

```
mkdir test2.git  
cd test2.git  
git init --bare
```

Conectar ambos repositorios: *nodo2*

```
git remote add origin vagrant@nodo1:/home/vagrant/test2.git  
git push -u origin master
```

2.b(Opcional) SparkeShare es una alternativa a dropbox que utiliza como cliente git.

Seguir el proceso de instalación en: <https://www.sparkleshare.org/>

3.Copias de seguridad con duplicity con ssh

creamos en el cliente el directorio test3

Hacemos una copia de seguridad con duplicity

añadimos ficheros

hacemos copia incremental

Recuperamos los ficheros en la carpeta test3b

4.Instalación del servidor Minio para objetos S3 (Bucket)

Instalamos el docker en nodo1

```
sudo apt update
sudo apt -y install apt-transport-https ca-certificates curl software-properties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal
stable"
sudo apt update
sudo apt -y install docker-ce
sudo usermod -aG docker vagrant
```

Salimos de la shell. Vovemos a entrar.

Lanzamos el minio con docker

```
docker run -p 9000:9000 \
--name minio1 \
-v /home/vagrant/minio_data:/data \
-e "MINIO_ACCESS_KEY=CpDRandom20" \
-e "MINIO_SECRET_KEY=scretkcPd20CY" \
minio/minio server /data &
```

4.b Probamos el cliente para acceder al servidores desde el nodo2 <https://github.com/minio/mc>

```
wget https://dl.min.io/client/mc/release/linux-amd64/mc
chmod +x mc
./mc -help
./mc alias set minio http://nodo1:9000 CpDRandom20 scretkcPd20CY
```

Probamos a acceder al servidor, Creamos bucket:

```
./mc mb minio/testx1
```

Crear subdirectorios es con la misma orden mb

```
./mc mb minio/testx1/d1
```

Copiamos ficheros:

```
./mc cp f1 minio/testx1
```

4.c) Podemos probar desde nuestro ordenador el cliente S3 (Ej: Dragondisk)

<http://www.s3-client.com/download-s3-compatible-cloud-client.html>

Podemos probar un cliente en modo gráfico para acceder desde nuestro ordenador.

4.d) (opcional) Copias de seguridad de duplicity en el servidor Minio

5. Configuración servidor Nextcloud

Lanzamos el contenedor en el nodo1

```
docker run -d -p 8080:80 nextcloud
```

Instalamos el cliente en nuestro ordenador