

## Práctica 9. Almacenamiento (II)

Objetivo: En esta práctica continuamos evaluando distintos recursos de almacenamiento cliente/servidor utilizamos en CPD.

Presentar un documento pdf con la siguiente información:  
-Capturas personalizadas de cada apartado donde se ve la orden que se ha ejecutado.

Desarrollo:

Podemos utilizar el entorno de la práctica anterior

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. Copias de seguridad con Duplicity

Duplicity es una sencilla y potente herramienta para realizar copias de seguridad.

<http://duplicity.nongnu.org/>

En este apartado vamos a utilizar el nodo 1 como servidor SSH y el nodo 2 como cliente.

Creamos una carpeta `$HOME/test9` en el cliente (nodo 2). Esta es la carpeta de la que queremos hacer copias de seguridad.

Creamos una carpeta `$HOME/backups` en el servidor (nodo 1). En esa carpeta guardaremos las copias de seguridad.

Utilizando la documentación en <http://duplicity.nongnu.org/vers8/duplicity.1.html>

y mediante `scp://` (`scp` es el protocolo de comunicación de ficheros en SSH).

Hacer copias utilizando cifrado.

Una lista interesante de ejemplos de uso:

<http://futurile.net/resources/duplicity/#restore-from-a-specific-time>

### Lista de tareas a realizar:

- Hacer una copia de seguridad de la carpeta `test9`
- Añadir algunos ficheros más y modificar alguno de ellos.
- Hacer una nueva copia de seguridad de la carpeta `test9`.
- Recuperar en la carpeta `test9_A` sólo el fichero que hemos modificado. Así podemos comprobar como fácilmente podemos recuperar un fichero.
- Recuperar en la carpeta `test9_B` la primera versión del fichero que hemos modificado utilizando la fecha y hora.
- Hacer 3 copias incrementales, con modificaciones de 1 fichero.
- Borrar las copias incrementales y quedarse sólo con 1. ( `remove-all-inc-of-but-n-full` )
- Comprobar la lista de ficheros

## 2. Despliegue de NextCloud

Configuración servidor Nextcloud

Siguiendo los pasos de la práctica anterior, instalar docker en nuestro nodo1.

Siguiendo las indicaciones: <https://github.com/nextcloud/docker>

Utilizar el docker-compose y basándose en el ejemplo, desplegar el servidor version – apache con mariadb (mysql). No utilizar Base version – FPM.

En el puerto 8080 de nuestro nodo 1 tendremos instalado Nextcloud.

Accedemos a <http://192.168.12.11:8080> desde nuestro ordenador para continuar con la configuración, indicando que se utilizará la BBDD de mysql y las claves. Importante, como desplegamos desde docker-compose, el nombre del servidor de mysql es *db* (no localhost). Con docker-compose hay virtualmente dos nodos, uno el que ejecuta el nextcloud y el otro el que ejecuta la BBDD, por eso desde nextcloud piensa que la BBDD está en el nodo *db*

Una vez que estamos en el entorno de configuración podemos añadir usuarios.

### **Comprobación cliente Nextcloud.**

Como nuestra máquina ubuntu (servidor Nextcloud) estará visible con la ip:192.168.12.11

Instalamos el cliente cliente Nextcloud en nuestro ordenador.

<https://nextcloud.com/install/#install-clients>

E indicamos como dirección de nuestro servidor <http://192.168.12.11:8080>

Podemos utilizar nuestro nuevo usuario para crear y sincronizar nuestra carpeta nextcloud con nuestro servidor.

### **Apartados opcionales:**

#### **3. Duplicity con S3.**

En el caso del duplicity, utilizar el servidor S3 de la práctica anterior para realizar las copias de seguridad, utilizando s3:// (en lugar de scp://)

#### **4. Nextcloud accesible desde cualquier equipo en nuestra casa.**

Siguiendo las indicaciones de [https://www.vagrantup.com/docs/networking/public\\_network](https://www.vagrantup.com/docs/networking/public_network)

Podemos poner nuestro nodo1 para que sea visible en la red de nuestra casa (“public\_network”). Vagrant nos preguntará, o bien podemos fijar, que interfaz actúa de puente (bridge).

De esta forma podemos utilizar otro equipo o nuestro móvil para acceder a nuestro servidor Nextcloud.

Con

ip addr

podemos saber la ip que nos ha asignado el servidor DHCP de nuestro router. Podemos forzar al router para que nos asigne siempre la misma IP.