June 7, 2021

Matias Garcia

Legajo: 90506

TP Integrador

Computación Aplicada

Ignacio Sanchez

Contents

[Topología 2](#_Toc73920930)

[Configuraciones generales de ambiente 3](#_Toc73920931)

[Configuraciones de Networking 4](#_Toc73920932)

[Firewall 6](#_Toc73920933)

[Instalación de JDK 10](#_Toc73920934)

[Apache Tomcat 12](#_Toc73920935)

[Servidor de archivos 13](#_Toc73920936)

[Implementación de LVM 15](#_Toc73920937)

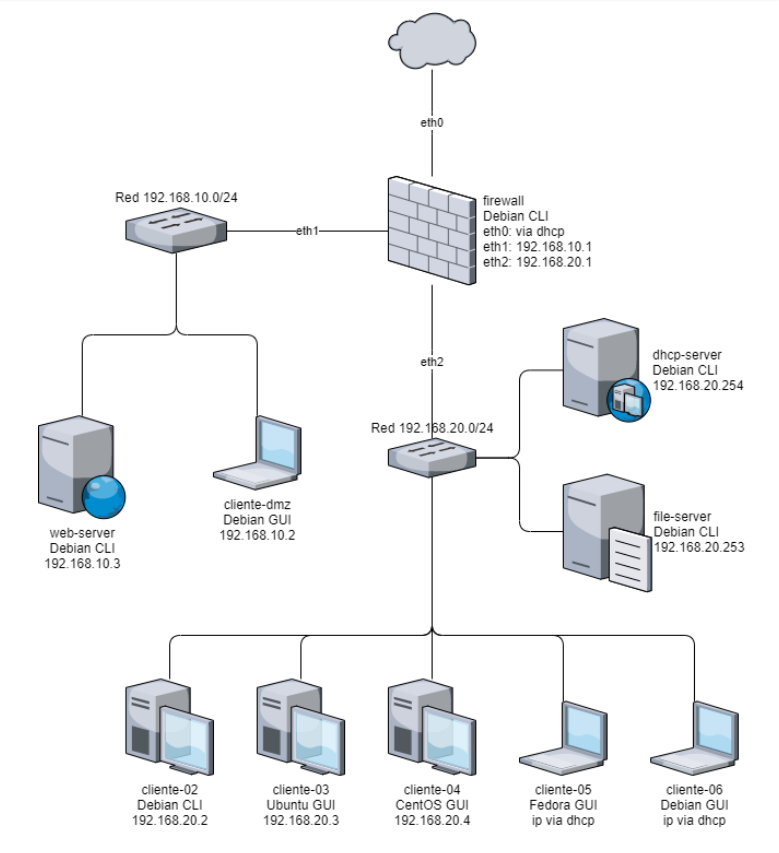
[Scripting 18](#_Toc73920938)

[Servidor DHCP 18](#_Toc73920939)

[Entrega 19](#_Toc73920940)

[Instalación y configuración en github 19](#_Toc73920941)

# Topología



**Usuarios:**

usu: mgarcia

pass: admin

pass root: admin

# Configuraciones generales de ambiente

Configuración ssh para acceso ssh desde mobaXterm hacia las virtuales.

Instalación de paquete ssh en todas las virtuales: *apt-get install ssh*

Cambiar nombre de hostname del so de las vms clonadas

Modificar archivo */etc/hostname*

Instalar paquete para el uso de herramientas de red, entre ellas ifconfig.

*apt install net-tools*

# Configuraciones de Networking

* Se configura en el firewall el archivo interfaces (/etc/network/interfaces), y se agrega la siguiente información para la configuración de una ip asignada por dhcp que tendrá salida a internet, y otras dos ip estáticas 192.168.10.1 y 20.1 para el funcionamiento interno de la infraestructura.

***# The primary network interface (DHCP - Salida a internet)***

*allow-hotplug eth0*

*iface eth0 inet dhcp*

***# Red 192.168.10.x (Red de Web-Server)***

*auto eth1*

*iface eth1 inet static*

*address 192.168.10.1*

*netmask 255.255.255.0*

***#Red 192.168.20.x (Red de file-server)***

*auto eth2*

*iface eth2 inet static*

*address 192.168.20.1*

*netmask 255.255.255.0*

Luego para que se efectúen los cambios se reinicia el servicio de networking:

*Service networking stop*

*Service networking start*

Se aplica el mismo concepto para configurar las ip estáticas en las VMs del web-server, cliente-dmz, dhcp-server, file-server, cliente-02, cliente-03 y cliente-04 definiendo el Gateway en cada caso (192.168.10.1 ó 192.168.20.1).

**Notas!!**

Para el **cliente-03** (Ubuntu 20), el archivo a modificar es */etc/netplan/01-network-manager-all.yaml* ya que a partir de Ubuntu 17 en adelante cambia el archivo de configuración que antes se heredaba de Debian.

Text

Description automatically generated

Luego mediante el comando *netplan apply* se aplican los cambios. Luego con *ip add* podremos ver que se ya quedo configurada la ip estática.

Para el **cliente-04** (Centos7), el archivo a configurar es:

*/etc/sysconfig/network-scripts/ifcfg-enp0s3*

Se marca en rojo las configuraciones aplicadas: Graphical user interface, text, application

Description automatically generated

Luego se reinicia el servicio: *systemctl restart network.service*

# Firewall

* El firewall deberá cargar la configuración de iptables al inicio.

Se genera el script en el firewall ubicado en /etc/network/if-up.d/ llamado *iptables* y se le asignaron permisos de ejecución *(chmod +x iptables).*

Este script carga la configuración de fw que establecimos con el comando

*iptables-save > /etc/iptables/rules.ipv4.conf* y las restablece con el comando *iptables-restore* mostrado debajo, ya que el contenido del directorio del deamon if-up.d es iniciado locate if-up.dal levantar la interface, es decir cuando inicia el sistema, por ejemplo.

Graphical user interface, text, application

Description automatically generated

A partir de ahora cualquier configuración de iptables cargada en el archivo rules.ipv4.conf será inicializado.

* Las políticas por defecto de las 3 cadenas de la tabla FILTER sea DROP.

Se definió por defecto que las cadenas de FILTER sean DROP mediante los siguientes comandos:

*iptables -P INPUT DROP*

*iptables -P OUTPUT DROP*

*iptables -P FORWARD DROP*

Text

Description automatically generated

Luego se cargo la configuración al archivo rules.ipv4.conf:

Text

Description automatically generated

(Se verifica que funciona reiniciando la vm).

* El tráfico desde/hacia la interfaz loopback sea posible.

Se habilita en iptables el tráfico hacia y desde la interfaz lo:

*iptables -A OUTPUT -o lo -j ACCEPT*

*iptables -A INPUT -i lo -j ACCEPT*

Se cargo en el archivo rules.ipv4.confg:

Text

Description automatically generated

* La única VM que pueda administrar el firewall vía ssh sea cliente-02.

Habilitamos el puerto ssh en iptables y luego lo cargamos al archivo rules.ipv4.conf

Habilitamos el puerto ssh default que es el 22 para la conexión entrante:

*iptables -A INPUT -p tcp –dport 22 -j ACCEPT*

Habilitamos el tráfico para las conexiones establecidas y relacionadas:

*iptables -A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT*

*iptables -A INPUT -p tcp -m multiport --dports 80,443 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT*

*iptables -A OUTPUT -p tcp -m multiport --dports 80,443 -m conntrack --ctstate ESTABLISHED -j ACCEPT*

Luego se configura el archivo /etc/ssh/sshd\_config. Y se agregan las siguientes dos líneas en el apartado de *Authentication*:

Graphical user interface

Description automatically generated with low confidence

Ya podemos conectarnos desde el cliente02:

*ssh -p 22 mgarcia@192.168.20.1*

* La única VM que pueda navegar por internet sea cliente-03.

Se habilita el fowarding en el archivo /etc/sysctl.conf

*net.ipv4.ip\_forward=1*

Permitimos que el fw enmascare, es decir, que otras VMs puedan salir a través del fw.

*echo 1 > /proc/sys/net/ipv4/ip\_forward*

Permitimos el tráfico de forward para el cliente-03 hacia internet hacia el web server:

*iptables -t nat -A POSTROUTING -s 192.168.20.3 -d 0.0.0.0/0 -j MASQUERADE*

Permitimos el tráfico entre las intefaces en ambos sentidos:

*iptables -A FORWARD -i eth2 -o eth0 -j ACCEPT*

*iptables -A FORWARD -i eth0 -o eth2 -j ACCEPT*

*Se guardan los cambios realizados en el archivo de arranque:*

*iptables-save > /etc/iptables/rules.ipv4.conf*

El s.o. no podía gestionar la conexión cableada y ni siquiera aparecía en la GUI para editarla. Esto se debe a que el parámetro *ifupdown* del archivo *NetworkManager.conf* mostrado debajo estaba en false. Se debe cambiar a true y reiniciar el sistema.

Text

Description automatically generated

Si bien sigue apareciendo como “desconexión cableada”, se consigue salir a internet:

Graphical user interface, application

Description automatically generated

Se verifica que el cliente 03 puede navegar correctamente.

* La única VM de la red 192.168.20.0/24 que pueda ingresar al web server de la red .10.0 sea cliente-04.

Permitimos el tráfico de solicitud desde el cliente-04 hacia el web-server, permitiéndole crear y establecer conexión:

*iptables -A FORWARD -i eth2 -o eth1 -p tcp -m state --state NEW,ESTABLISHED -s 192.168.20.4 -d 192.168.10.3 -j ACCEPT*

Permitimos el tráfico de vuelta desde el web-server hacia cliente-04, permitiéndole establecer conexión:

*iptables -A FORWARD -o eth1 -i eth2 -p tcp -m state --state ESTABLISHED -s 192.168.10.3 -d 192.168.20.4 -j ACCEPT*

Guardamos los cambios en el archivo:

*iptables-save > /etc/iptables/rules.ipv4.conf*

Si bien se puede acceder desde el dmz no se podía acceder desde el cliente-04. Se encontró que tanto el web-server como el cliente-04 no estaban tomando el Gateway configurado en el archivo interfaces. Se configuro a mano el Gateway en ambas VMs con el comando:

*ip route add default via 192.168.20.1* en el cliente-04 y *192.168.10.1* en el wev-server.

Se valida la ruta aplicada con el comando:

*route -n*

Observamos que desde el cliente-04 podemos hacer deploy de la app sample desde *192.168.10.3:8080/sample*

Graphical user interface, text, application, email

Description automatically generatedServidor WEB

## Instalación de JDK

* Obtener el archivo jdk-xxx.tar.gz desde el sitio de Oracle con cualquier maquina con entorno gráfico, via wget o scp desde otro equipo de la red.

Se descargó el archivo desde el cliente 06 y por scp se transfirió al web server:

*scp jdk-11.0.11\_linux-x64\_bin.tar.g mgarcia@192.168.0.49:/home/mgarcia/*

* Descomprimir el archivo jdk-xxx.tar.gz en el directorio /opt

Se descomprime el archivo en /opt:

*tar -xvf jdk-11.0.11\_linux-x64\_bin.tar.gz -C /opt/*

* Crear la variable de entorno JAVA\_HOME en ~/.bashrc y agregarla a la variable PATH.

Se definen las variables de entorno en el archivo .bashrc del usuario mgarcia:

*mgarcia@web-server:~$ nano ~/.bashrc*

*############## JAVA ###################*

*export JAVA\_HOME=/opt/jdk-16.0.1*

*export PATH=$PATH:$JAVA\_HOME/bin*

## Apache Tomcat

Instalación de Apache Tomcat y despliegue de la aplicación de ejemplo

* Obtener el archivo apache-tomcat-xxx.tar.gz desde el sitio de Apache desde cualquier VM con entorno gráfico.

Se copió del repo de CAaplicada por winscp

* Descomprimir el archivo apache-tomcat-xxx.tar.gz en el directorio /opt

*tar -xvf apache-tomcat-8.5.54.tar.gz -C /opt/*

* Obtener la aplicación de ej: #wget <https://tomcat.apache.org/tomcat-7.0-doc/appdev/sample/sample.war>

Se copió del repo de CAaplicada por winscp

* Desplegar la app de ejemplo.

Se copió a /opt/webapps el sample.war

RESTA ACCEDER desde el dmz o desde el cliente 04 a la web para ver el sample.war (“hola mundo”)

# Servidor de archivos

Se creó un nuevo hard disk en VirtualBox en la vm del file-server.

Con los siguientes comandos se entró en la gestión del disco sdb1 ya creado, se creó la nueva partición, se formateo y se montó en */media/disco\_backups* previamente creando la carpeta disco\_backups:

*fdisk /dev/sdb* 🡨 Se ingresa en este modo para particionarlo.

Con la opción n creamos una nueva partición. Definimos que es una partición primaria y luego definimos los sectores por predeterminado. Luego con la opción w guardamos los cambios aplicados.

Luego formateamos la nueva partición con:

*mkfs.ext4 /dev/sdb1* (respetamos el mismo formato que en el disco del so).

Ya quedó agregada y formateada la nueva partición del nuevo hard disk:

Text

Description automatically generated

Con *mkdir /media/disco\_backups* creamos la carpeta donde se montará el nuevo disco.

Montamos el disco: mount /dev/sdb1 /media/*disco\_backups*

Text

Description automatically generated

Editamos el archivo fstab para que el montaje del disco sea automático en cada inicio del sistema. (Previamente hacemos backup del archivo fstab en /etc/backupFiles/fstab por si es necesaria una restauración del mismo).

Text

Description automatically generated

Definimos el disco, el punto de montaje y el tipo de formato del disco.

La opción *defaults* incluye a las opciones:

* *rw*: Permitimos lectura-escritura en el disco.
* *suid*: Permite que a los programas que se efectúen sobre el sistema de archivos del disco se les pueda cambiar id de usuario o grupos de usuario.
* *dev*: Permitimos la utilización de nodos de dispositivo en el sistema de archivos.
* *exec*: Permite la ejecución de programas.
* *auto*: Automonta el disco.
* *nouser*: Solo permisos de montaje para root.
* *async*: Permitimos la asincronización de operaciones, para que no tengan que realizarse en el mismo momento.
* *noatime:* Desactiva la escritura de los tiempos de acceso a los archivos en la unidad.
* *dump: 0* (No se incluye la opción de backups por el sistema del disco)
* *pass: 2* (Permitimos al so la comprobación de consistencia de archivos)

## Implementación de LVM

Se creará el nuevo disco *disco\_backups.*

En primera instancia se agrega un disco nuevo en la máquina virtual de file-server:

Text

Description automatically generated

Instalamos las librerías para gestionar volumes groups:

*aptitude install lvm2*

!! No pude lograr instalar la librería para implementar lvm.

* Creamos el physical volume:

*pvcreate /dev/sdb*

* Creamos el volume group:

*vgcreate caplicada /dev/sdb*

* Creamos el logical volume:

*lvcreate caplicada -L 1.99G -n backups*

* Creamos el filesystem en el logical volumen:

*mkfs.ext3 /dev/mapper/testing-storage*

* Agregamos el nuevo volume a FSTAB

*/dev/mapper/testing-storage /storage ext3 defaults 0 2*

* Creamos la carpeta *disco\_backups* en */media*
* Montamos el disco: mount /dev/sdb1 /media/*disco\_backups*
* Editamos el archivo fstab para que el montaje del disco sea automático en cada inicio del sistema.
* Programar un script con el nombre backup\_home\_cliente-03.sh, que permita realizar un backup de su carpeta home de la maquina 192.168.20.3 y que cumpla con lo siguiente:

Se definió el archivo *config* en la carpeta *~/.ssh* con los datos del servidor ssh al que vamos a conectarnos cada vez que se realice el backup. De esta forma podemos conectarnos simplemente utilizando un nombre de host y no escribiendo todos los datos necesarios.

Datos cargados en el archivo *config*:

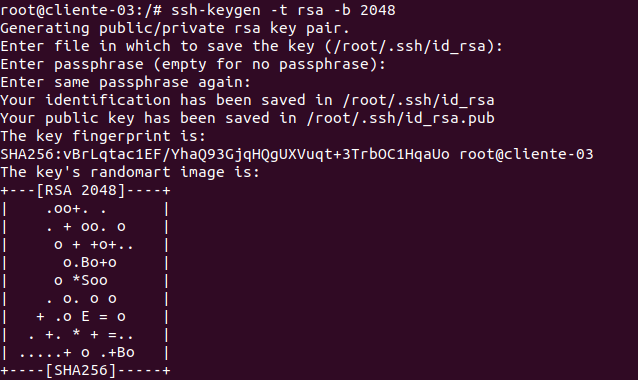
*Host file-server*

*HostName 192.168.20.253*

*User mgarcia*

*Port 22*

Se generan ambas claves (pública y privada) para la conexión ssh



Se hace un copy de la clave pública al servidor remoto al que nos conectaremos por ssh:

Text

Description automatically generated

Se observa que ya estamos utilizando el host definido en el archivo *config* pasos atrás. Y vemos como quedó agregada la clave al servidor remoto.

Se prueba el conexón ssh al servidor remoto utilizando el host del archivo config y ya sin solicitar una password:

Text

Description automatically generated

Se instaló la librería rsync en el cliente-03 para la transferencia de archivos:

*apt-get install rsync*

Parámetros usados de rsync en el script para la transferencia de archivos:

-a 🡪 Habilita el modo archivo.

-t 🡪 Evita la modificación de timestamp de los archivos transferidos.

-e 🡪 Define el protocolo a utilizar, en este caso ssh. (No usado)

*-h 🡪* Human-redeable (Muestra en formato legible).

*-v 🡪* Modo Verbose (muestra el progreso del proceso).

-r 🡪 Copia de forma recursiva, es decir si hay sub-directorios y archivos. (No usado)

-z 🡪 Comprime los datos durante la transferencia.

# Scripting

Se dejo el script en el cliente-03 (192.168.20.3) en la siguiente ubicación:

***/home/mgarcia/Scripting/scripting-tp-integrador.sh***

# Servidor DHCP

Se instaló el paquete en el dhcp-server:

*aptitude install isc-dhcp-server*

Se habilitó la “eth0” al dhcp-server en el archivo isc-dhcp-server

Se habilitó la siguiente subnet en el archivo de configuración /etc/dhcp/dhcpd.conf con el rango de asignación de IPs dinámicas:

*subnet 192.168.20.0 netmask 255.255.255.0 {*

*range 192.168.20.101 192.168.20.110;*

*option routers 192.168.20.1;*

*option domain-name-server 1.1.1.1,8.8.8.8;*

*option broadcast-address 192.168.20.255;*

*}*

Se verificó que este instalado el paquete en los clientes que reciben ip por dhcp y se instaló en los que hiciera falta con:

*aptitude install isc-dhcp-client*

Se reinició el servicio dhcp:

*/etc/init.d/isc-dhcp-server restart*

# Entrega

Para la entrega de este TP, deberá crear un repositorio en GitHub llamado “**up-ca-tpintegrador-<APELLIDO>-<NRO\_LEGAJO>**” (Por ej. https://github.com/sanchezih/up-ca-tpintegrador-sanchez-061234) y dentro del mismo deberá pushear un documento detallando los pasos realizados y los siguientes archivos. El dia de la entrega, a las 23:59 Hs. se clonarán los repositorios.

Los archivos a entregar serán los siguientes:

1. Un archivo llamado **rules.ipv4.conf**, en cual será testeado con el comando #iptables-restore < ./rules.ipv4.conf.
2. De la VM web-server, el archivo **.bashrc** que sobre el que tuvo que trabajar.
3. Los archivos de ISC DHCP sobre los que tuvo que trabajar: **/etc/default/isc-dhcp-server** y **/etc/dhcp/dhcpd.conf**.
4. El archivo **scripting-tp-integrador.sh**.
5. El archivo **backup\_home\_cliente-03.sh**.

# Instalación y configuración en github

Se creo una carpeta llamada 2PparcialFiles en /home/media del cliente-03 a la cua l e le realizara un push hacia git para clonar el repositorio en el examen.

**Pasos de instalación, configuración y carga de archivos en GIT:**

Se instaló git:

*apt-get install git-core*

Configuración de cuenta de git:



Se preparan los archivos que se subirán:

*git add .*

Se verifica el estado de archivos a sincronizar:

*git status*

Se realiza el commit:

*git commit -m “comentario de commit”*

Se agrega el Branch remoto al cual se hará el push de archivos:

*git remote add origin* [*https://github.com/mgarcia173/up-ca-tpintegrador-Garcia-90506.git*](https://github.com/mgarcia173/up-ca-tpintegrador-Garcia-90506.git)

*Se realiza el push:*

*git push -u origin master*

Nota: Nos pedirá el user y pass del repositorio.