**COMPUTACION APLICADA**

**TRABAJO PRÁCTICO**

**Alumno: Blanco, Pablo**

**Cuatrimestre/Año: 1/2020**

**Docentes: Sanchez, Ignacio**

**Kainer, Claudio**

Índice

[Instalación Java SDK y Tomcat 3](#_Toc41929500)

[Configuración Firewall 5](#_Toc41929501)

[Configuración de servidor DHCP 7](#_Toc41929502)

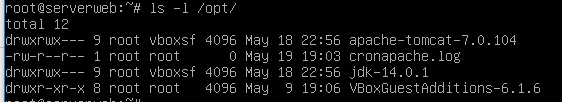
[Configuración LVM 8](#_Toc41929503)

[Agregando un disco a la configuración de LVM 9](#_Toc41929504)

[Diferencias con otras distribuciones 12](#_Toc41929505)

# Instalación Java SDK y Tomcat

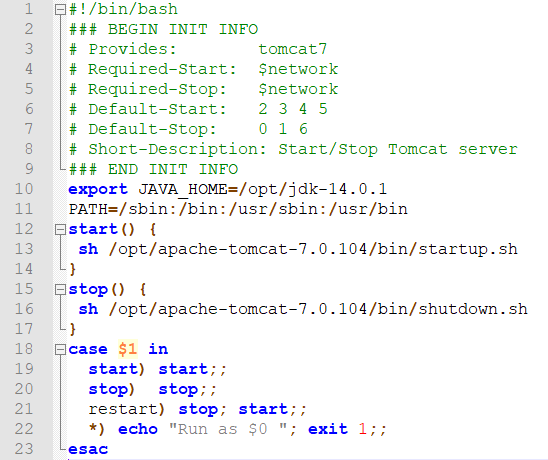
Luego de descargar las aplicaciones como lo mencionaba la consigna del trabajo se descomprimieron con el comando **tar xzvf archivo.tar.gz** y se copiaron al directorio /opt



Para que Java inicie automáticamente se modificó el archivo **/root/.bashrc** agregando las siguientes líneas:

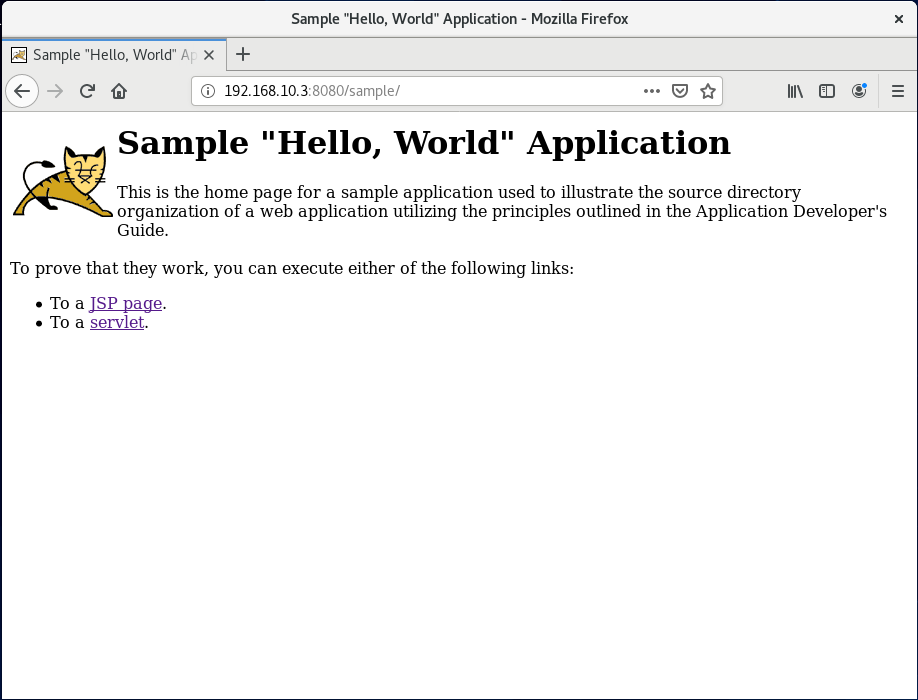


Para que tomcat 7 inicie automáticamente como servicio se creó en **/etc/init.d** el script **tomcat7.sh**



Luego se ejecutó el comando “**update-rc.d tomcat7 defaults”** para actualizar los servicios que se ejecutan al inicio.

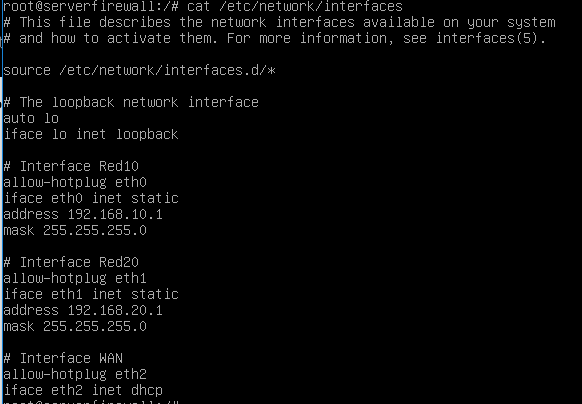
Una vez levantados los servicios de Java y Tomcat se copió la aplicación de ejemplo “simple.war” al directorio al directorio /webapps dentro de la carpeta de instalación de apache y se verifico el funcionamiento:



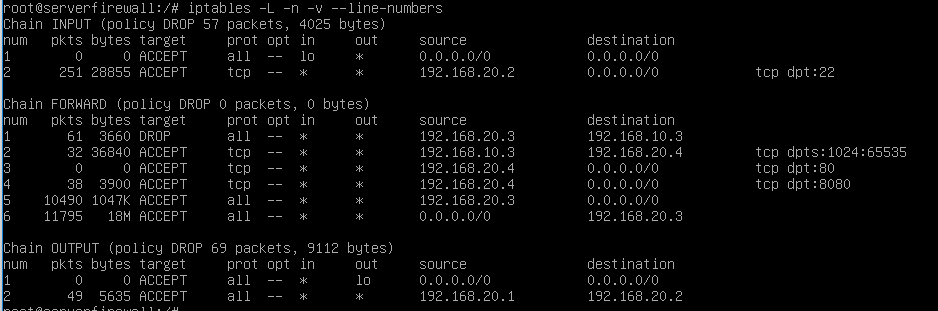
# Configuración Firewall

El primer paso fue instalar iptables, para ello se utilizó el comando **apt-get install iptables**

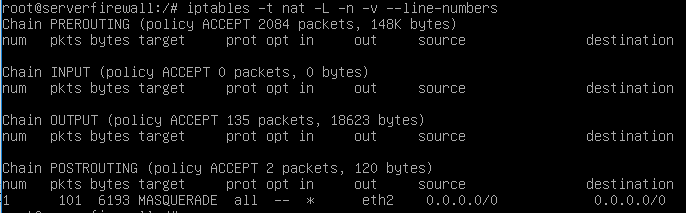
Luego se agregaron desde la configuración de Virtual Box 2 adaptadores de Red para cumplir con los requerimientos de la configuración pedida.

Una vez agregados los adaptadores de red se configuro el archivo **/etc/network/interfaces** de la siguiente manera:

En el mismo archivo se puede ver a qué red corresponde cada interfaz, junto con su configuración.

La configuración de iptables se realizó con los comandos correspondientes para llegar a la siguiente configuración de puertos: 

Y también se configuraron las reglas de NAT necesarias:



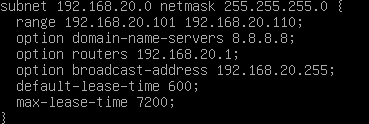
Por último se salvó el archivo en **/root/rules.ipv4.conf** y para hacer que la reglas se restauren al inicio (cuando levantan los adaptadores de red) se creó un script en **/etc/network/if-up.d** con el siguiente contenido:



# Configuración de servidor DHCP

Para iniciar se instaló ISC DHCP con el comando **apt-get instal isc-dhcp-server**

Luego se configuro el archivo **/etc/dhcp/dehcpd.conf** con las siguientes líneas para asignar el rango de direcciones pedido en las consignas a la LAN



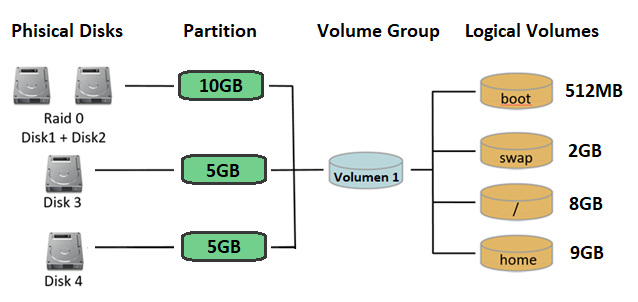
Y en el archivo **/etc/default/isc-dhcp-server** solo se agregó la interfaz de red correspondiente que responde las solicitudes dhcp:



# Configuración LVM

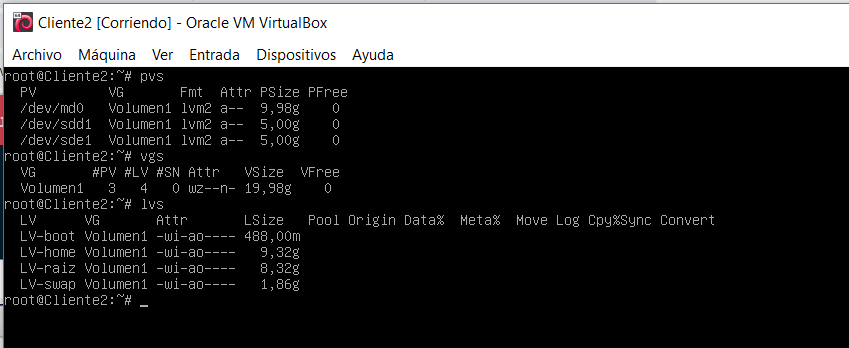
La configuración de LVM se realizó durante la instalación. Para ello en Virtual Box se crearon 4 discos virtuales de 5 GB cada uno.

Luego, con el fin de probar algunas funcionalidades de que otorga Debian se creó con 2 de los discos un RAID 0 para luego hacer una partición de 10GB y los otros 2 discos quedaron con una partición de 5GB cada uno de la siguiente manera:



Como se puede ver en la imagen, se creó un único volumen con todas las particiones disponibles y en dicho volumen se crearon 4 volúmenes lógicos asignados como se ve en la imagen anterior.

A continuación vemos la ejecución de los comandos **vgdisplay –v** , **pvdisplay –v** y **lvdisplay –v** para mostrar la configuración de LVM.



# Agregando un disco a la configuración de LVM

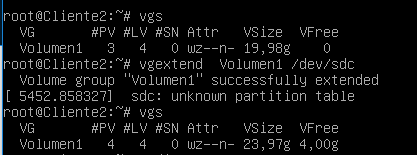
Para continuar con la pruebas de LVM, el objetivo fue extender los volumenes lógicos que corresponden a los directorios **/** y **/home.**

Para ello se agregó desde la configuración de Virtual Box un nuevo “disco físico” de 4 GB.

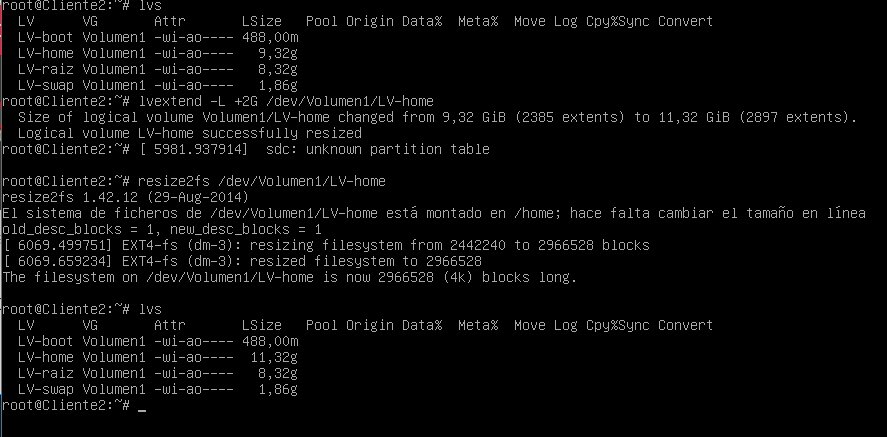
El mismo fue reconocido por el OS en **/dev/sdc**. Luego se ejecuto el comando **pvcreate /dev/sdc** para que el disco se agregue a la configuracion de LVM y por ultimo la ejecucion del comando **pvs** para ver todos los discos fisicos como se muestra en la siguiente imagen:



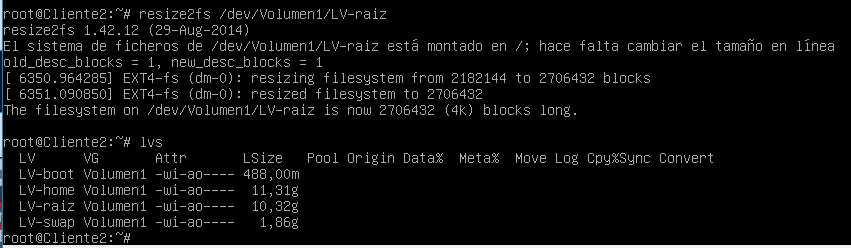
En la próxima imagen vemos la ejecución del comando **vgs** para mostrar el volumen creado y luego el comando “**vgextend Volumen1 /dev/sdc”** para justamente extender el Volumen uno con la capacidad del disco sdc anteriormente creado, por último la ejecución de **vgs** nuevamente para verificar:



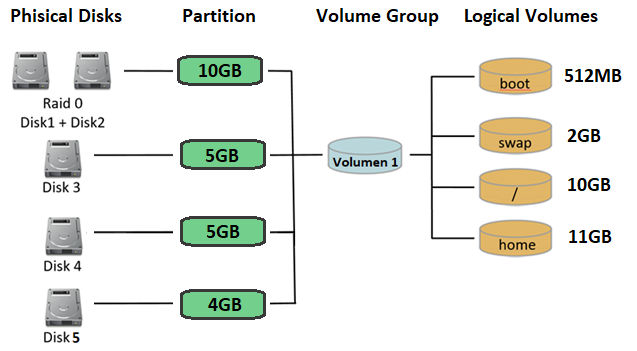
En la siguiente imágen vemos la ejecucion del comando **lvs** para mostrar los volumenes logicos y posteriormente la ejecucion del comando **“lvextend –L +2G /dev/Volumen1/LV-home”** para extender 2 GB el volume logico **/home.** Luego puede verse nuevamente la ejecucion de **lvs** para verificar el tamaño del volumen logico LV-home:



Los mismos pasos se realizaron esta vez para el volumen lógico **LV-raiz:**



Realizando esta configuración se pudo agregar un nuevo disco físico y esa capacidad utilizarla para extender 2 volúmenes lógicos distintos obteniendo así la siguiente estructura:



En conclusión se puedo verificar que es muy sencillo extender la capacidad de un volumen implementando la solución de LVM.

# Diferencias con otras distribuciones

Pudo verificarse lo visto en clase sobre las distintas distribuciones de GNU/Linux. Pongo como ejemplo la configuración de red en Ubuntu 20.04. Al querer realizar la ejecución de “ifconfig” note que no estaba disponible el comando y para ello fue necesario la instalación de un paquete llamado **net-tools.**



Y a su vez los archivos de configuracion de red se encuentran en otra ubicación distinta que en Debian 8 e incluso los archivos tienen formatos distintos:



