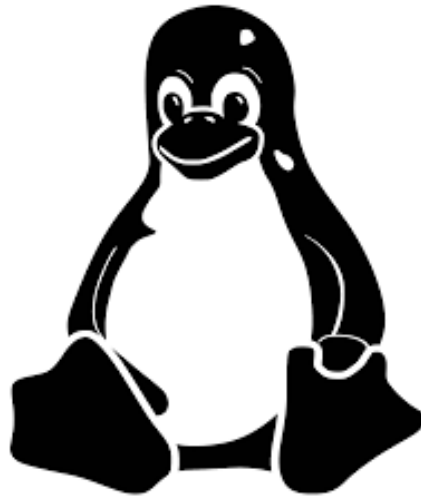


Computación Aplicada: Segundo Parcial - Parte asincrónica



Profesor: Ignacio Sanchez

**Alumno:
Moviglia Juan Manuel**

Fecha de entrega : 8 de Noviembre de 2021

Lugar de entrega y Materia: Universidad de Palermo - Computación Aplicada

Pasos Realizados del TP Integrador

Topologia de Red:

Se conformó una topología de red constituida por 3 redes:

Eth0 (NAT)

Esta red es la salida al “mundo exterior” de la topología en cuestión, conectada a un firewall el cual administra la conexión de todas las redes en simultáneo.

Eth1 (192.168.10.0)

Eth1 está conformada por un web-server (apache-tomcat) capaz de correr una aplicación en conjunto con la base de datos mySQL, y un cliente-dmz el cual facilito la instalación del web-server (JDK, apache-tomcat, webapps)

Eth2 (192.168.20.0)

La “red20” o “eth2”, fue compuesta por 5 máquinas “cliente”, un dhcp-server y un file-server. Dentro de las máquinas cliente, las ip de los cliente-05 y 06 están basadas en en el dhcp-server, el cual fue configurado en el archivo /etc/default/isc-dhcp-server y /etc/dhcp/dhcpd.conf para que el rango de ips por dhcp sea entre 192.168.20.101 y 110.

Servidor Web

Para la instalación del servidor web, se dispuso de una VM Debían 11 CLI, y otra VM pero con entorno gráfico (cliente-dmz).

En primer lugar se descargo el JDK-11 mediante el comando:

```
sudo apt install openjdk-11-jdk
```

Luego para setear Apache Tomcat, se obtuvo el archivo *.tar.gz desde el cliente-dmz y se lo transfirió a la carpeta /opt/tomcat desde la máquina “web-server” de la siguiente manera:

```
Sudo scp cliente-dmz@192.168.10.2:Desktop/apache-tomcat-9.0.54 /opt/tomcat
```

Consecuente a esto, se extrajo el .tar de la siguiente manera

```
sudo tar xzvf apache-tomcat-9.0.54tar.gz -C /opt/tomcat
--strip-components=1
```

Para poder poner en inicio el webserver con facilidad se creó un archivo Systemd en la dirección /etc/systemd/system/tomcat.service. El mismo contenía el siguiente código:

```
[Unit]
Description=Tomcat webs servlet container
After=network.target

[Service]
Type=forking

User=tomcat
Group=tomcat
RestartSec=10
Restart=always
Environment="JAVA_HOME=/usr/lib/jvm/java-1.11.0-openjdk
-amd64"
Environment="JAVA_OPTS=-Djava.awt.headless=true
-Djava.security.egd=file:/dev/./urandom"

Environment="CATALINA_BASE=/opt/tomcat"
Environment="CATALINA_HOME=/opt/tomcat"
Environment="CATALINA_PID=/opt/tomcat/temp/tomcat.pid"
Environment="CATALINA_OPTS=-Xms512M -Xmx1024M -server
-XX:+UseParallelGC"

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh

[Install]
WantedBy=multi-user.target
```

Finalizado esto, se crearon los usuarios debidos para administrar el web-server en el archivo /opt/tomcat/conf/tomcat-users.xml.

Por último, y no menos importante, se abrió el puerto 8080 para poder acceder al servidor desde navegadores conectados a la red

```
sudo ufw allow 8080
```

Y con esto el servidor está iniciado correctamente.

Pasos extra

Se desplegó la aplicación sample.war moviendo el archivo “sample.war” al directorio /opt/tomcat/webapps/. A su vez, se instaló MySQL (seteando la password de root en “password1234”), y se creó la base de datos con el código en cuestión:

```
CREATE DATABASE IF NOT EXISTS `userdb`;
USE `userdb`;

CREATE TABLE IF NOT EXISTS `users` (
  `id` int NOT NULL AUTO_INCREMENT,
  `firstname` varchar(45) DEFAULT NULL,
  `lastname` varchar(45) DEFAULT NULL,
  `dob` date DEFAULT NULL,
  `email` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8mb3;

DELETE FROM `users`;
INSERT INTO `users` (`id`, `firstname`, `lastname`, `dob`, `email`)
VALUES
(1, 'Maria', 'Fernandez', '1988-07-07', 'maria@fernandez.com');
```

Posterior a esto, se desplegó poc-jsp-mysql-crud.war en el directorio /opt/tomcat/webapps/ y se ejecutó la misma entrando al enlace <http://192.168.10.3:8080/poc-jsp-mysql-crud/index.jsp> desde el cliente-dmz

File Server

El file server se creó con 2 discos, uno donde está alojado el sistema operativo, y otro donde se guardarán los backups del cliente-03. A su vez, en el directorio /media/disco_backups se creó el script backup_home_cliente-03.sh capaz de realizar un backup de la carpeta home del cliente-03 cada 24 hrs.

Para agregar, se investigó la tecnología LVM y se la inculcó en el disco donde se alojan los backups. Esta tecnología permite reducir y modificar particiones en discos duros en tiempo real, sin la necesidad de desmontar el sistema de archivos. Para este servidor funciona perfectamente, ya que el mismo en una situación real debería de correr todo el tiempo, para hacer los backups de manera correcta.

Firewall

Se configuro las iptables del firewall para que admitan las siguientes reglas:

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

```
IPTABLES -P INPUT DROP
IPTABLES -P OUTPUT DROP
IPTABLES -P FORWARD DROP
```

El tráfico desde/hacia la interfaz loopback sea posible.

```
IPTABLES -A INPUT -i lo -j ACCEPT
IPTABLES -A OUTPUT -o lo -j ACCEPT
```

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

```
iptables -A INPUT -p tcp -s 192.168.20.2 -i eth2 --dport 22 -m
state --state NEW,ESTABLISHED -j ACCEPT iptables -A OUTPUT -p
tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
```

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

```
iptables -A FORWARD -i eth2 -o eth0 ACCEPT
iptables -A FORWARD -i eth0 -o eth2 ACCEPT
iptables -t nat -A POSTROUTING -s 192.168.20.3 -d 0.0.0.0/0 -j
MASQUERADE
```

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.

```
iptables -A FORWARD -s 192.168.20.4/32 -i eth2 -o eth1 -j
ACCEPT
iptables -A FORWARD -i eth1 -o eth2 -m state --state
RELATED,ESTABLISHED -j ACCEPT
```