

Practica06

Balanceador de carga con apache.

En esta práctica vamos a usar una máquina virtual con Apache HTTP Server como un proxy inverso para hacer de balanceador de carga. El objetivo de esta práctica es crear una arquitectura de alta disponibilidad que sea escalable y redundante, de modo que podamos balancear la carga entre todos los frontales web.

Elementos a tener en cuenta para la realización de la práctica:

La arquitectura estará formada por:

- Un balanceador de carga, implementado con un Apache HTTP Server configurado como proxy inverso.
- Una capa de front-end, formada por dos servidores web con Apache HTTP Server.
- Una capa de back-end, formada por un servidor MySQL.

Las direcciones IPs que tendrán las máquinas virtuales serán las siguientes:

- Balanceador: 192.168.33.10
- Frontal Web 1: 192.168.33.11
- Frontal Web 2: 192.168.33.12
- Servidor de bases de datos MySQL: 192.168.33.13

Activación de los módulos necesarios en Apache:

Habrà que activar los siguientes módulos:

```
a2enmod proxy
a2enmod proxy_http
a2enmod proxy_ajp
a2enmod rewrite
a2enmod deflate
a2enmod headers
a2enmod proxy_balancer
a2enmod proxy_connect
a2enmod proxy_html
a2enmod lbmethod_byrequests
```

Configuración de Apache para trabajar como balanceador de carga:

Editar el archivo `000-default.conf` que está en el directorio `/etc/apache2/sites-enabled` :

```
sudo nano /etc/apache2/sites-enabled/000-default.conf
```

Añadimos las directivas **Proxy** y **ProxyPass** dentro de VirtualHost.

```
<VirtualHost *:80>
  # Dejamos la configuración del VirtualHost como estaba
  # sólo hay que añadir las siguiente directivas: Proxy y ProxyPass

  <Proxy balancer://mycluster>
    # Server 1
    BalancerMember http://IP-HTTP-SERVER-1

    # Server 2
    BalancerMember http://IP-HTTP-SERVER-2
  </Proxy>

  ProxyPass / balancer://mycluster/
</VirtualHost>
```

Ahora tendremos que reemplazar **IP-HTTP-SERVER-1** y **IP-HTTP-SERVER-2** por las direcciones IPs de las dos máquinas que estamos utilizando con Front-End.

Reiniciamos el servicio Apache:

```
sudo /etc/init.d/apache2 restart
```

Vagrant File:

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

Vagrant.configure("2") do |config|

  config.vm.box = "ubuntu/xenial64"

  # Apache HTTP Server
  config.vm.define "web1" do |app|
    app.vm.hostname = "web1"
    app.vm.network "private_network", ip: "192.168.33.10"
    app.vm.provision "shell", path: "apache.sh"
  end

  # Apache HTTP Server
  config.vm.define "web2" do |app|
    app.vm.hostname = "web2"
    app.vm.network "private_network", ip: "192.168.33.11"
    app.vm.provision "shell", path: "apache.sh"
  end
end
```

```
# MySQL Server
config.vm.define "db" do |app|
  app.vm.hostname = "db"
  app.vm.network "private_network", ip: "192.168.33.12"
  app.vm.provision "shell", path: "mysql.sh"
end

# balanceador
config.vm.define "balance" do |app|
  app.vm.hostname = "banlace"
  app.vm.network "private_network", ip: "192.168.33.3"
  app.vm.provision "shell", path: "balance.sh"
end
end
```

Contenido del 000-Default.conf:

```
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    # ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    # LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    # Include conf-available/serve-cgi-bin.conf

    # AQUÍ ES DÓNDE REEMPLAZAMOS IP-HTTP-SERVER-1 y IP-HTTP-SERVER-2
```

```
<Proxy balancer://mycluster>
    # Server 1
    BalancerMember http://192.168.33.10

    # Server 2
    BalancerMember http://192.168.33.11
</Proxy>

ProxyPass / balancer://mycluster/

</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Guía de Scripts:

Balanceador:

```
#!/bin/bash

# Actualización
apt-get update

#Instalación de Apache
apt-get install -y apache2
apt-get install -y php libapache2-mod-php php-mysql
sudo /etc/init.d/apache2 restart

# modulos de activación
a2enmod proxy
a2enmod proxy_http
a2enmod proxy_ajp
a2enmod rewrite
a2enmod deflate
a2enmod headers
a2enmod proxy_balancer
a2enmod proxy_connect
a2enmod proxy_html
a2enmod lbmethod_byrequests

# editamos el archivo 000-default.conf
sudo rm -rf /etc/apache2/sites-enabled/000-default.conf
cp /vagrant/000-default.conf /etc/apache2/sites-enabled/

#Reiniciamos el servicio de Apache:
sudo /etc/init.d/apache2 restart
```

Script Apache:

```
#!/bin/bash

# Instalación de Apache
apt-get update
apt-get install -y apache2
apt-get install -y php libapache2-mod-php php-mysql
sudo /etc/init.d/apache2 restart

# Clonamos los repositorios
apt-get install -y git
cd /tmp
rm -rf iaw-practica-lamp
git clone https://github.com/josejuansanchez/iaw-practica-lamp.git

# Copiamos los repositorios
cd iaw-practica-lamp
cp src/* /var/www/html/

# Modificamos la base de datos que queremos usar:
sed -i 's/localhost/192.168.33.12/' /var/www/html/config.php
chown www-data:www-data /var/www/html/* -R

# Borramos el index:
rm -rf /var/www/html/index.html
```

Script de la base de datos:

```
#!/bin/bash

# Actualización e instalación de repositorios:
apt-get update
apt-get -y install debconf-utils

DB_ROOT_PASSWD=root
debconf-set-selections <<< "mysql-server mysql-server/root_password password
$DB_ROOT_PASSWD"
debconf-set-selections <<< "mysql-server mysql-server/root_password_again password
$DB_ROOT_PASSWD"

# Instalación de repositorios:
apt-get install -y mysql-server
sed -i -e 's/127.0.0.1/0.0.0.0/' /etc/mysql/mysql.conf.d/mysqld.cnf
#/etc/init.d/mysql restart

#mysql -uroot mysql -p$DB_ROOT_PASSWD <<< "GRANT ALL PRIVILEGES ON *.* TO root@'%'
IDENTIFIED BY '$DB_ROOT_PASSWD'; FLUSH PRIVILEGES;"

# Clonación de los repositorios:
apt-get install -y git
```

```
cd /tmp
rm -rf iaw-practica-lamp
git clone https://github.com/josejuansanchez/iaw-practica-lamp.git

# Creación de la base de datos
mysql -u root -p$DB_ROOT_PASSWD < /tmp/iaw-practica-lamp/db/database.sql

# Reiniciar el servicio
/etc/init.d/mysql restart
```