

practica06

Balanceador de carga con Apache.

Arquitectura y direccionamiento IP de las maquinas.

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 son las siguientes:

- Balanceador. (IP: 192.168.33.10)
- Frontal Web 1. (IP: 192.168.33.11)
- Frontal Web 2. (IP: 192.168.33.12)
- Servidor de Base de Datos MySQL. (IP: 192.168.33.13)

Como activamos los modulos necesarios de apache.

Activamos 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
```

Estos modulos tendremos que meterlos en el script que creemos para el balance.

Configuración de Apache para trabajar como balanceador de carga.

Editamos 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>
```

Cambiaremos **IP-HTTP-SERVER-1** y **IP-HTTP-SERVER-2** por las direcciones IPs de las maquinas del Front-end.

Despues de aplicar los cambios reiniciaremos el servicio de apache con:

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

Script Balance

```
#!/bin/bash
apt-get update
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/
```

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

Script apache

```
#!/bin/bash

# instalacion 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

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

#copiar repositorio

cd iaw-practica-lamp
cp src/* /var/www/html/

#modificar 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 MySQL

```
#!/bin/bash
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"

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;"  
  
#clonar repositorios  
apt-get install -y git  
cd /tmp  
rm -rf iaw-practica-lamp  
git clone https://github.com/josejuansanchez/iaw-practica-lamp.git  
  
#crear base de datos  
mysql -u root -p$DB_ROOT_PASSWD < /tmp/iaw-practica-lamp/db/database.sql  
  
/etc/init.d/mysql restart
```

Contenido de 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  
  
    <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

```

Contenido de Vagrantfile

```

# -*- 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

  # 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

```