

Practica06

Creación de cuatro máquinas virtuales con Vagrant. Dos con apache, otra con Mysql y otra como balanceador de carga. E instalar Wordpress en las dos máquinas.

Vamos a crear una arquitectura con cuatro máquinas virtuales donde dos serán máquinas donde correrá apache con la web la cual queremos mostrar, otra donde residirán todas las BD en las cuales se apoyarán los sitios Web de las otras máquinas con el servidor web. Y una última la cual actuará como balanceador de carga a la cual nos conectaremos y automáticamente nos redigirá a uno de los dos servidores Web.

Paso 1: Creación del archivo Vagrantfile

Crearemos un archivo Vagrantfile en el cual debemos de declarar cuatro máquinas virtuales con ubuntu e IP's en red interna con cuatro IP's distintas. Además debemos de configurar tres provision diferentes para cada una de las máquinas. Un provision para el Mysql, otro para las dos máquinas de Apache y un último para el balanceador de carga.

Paso 2: Creación de los provision

Debemos de crear tres provision escritos en bash los cuales uno hará que se instale apache y un Wordpress para comprobar el correcto funcionamiento de la máquina con Mysql y un último el cuál será la máquina balanceador a la cual nos conectaremos.

Provision para apache:

```
#!/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

#INSTALAR wget
apt-get install -y wget
cd /tmp
rm -rf latest.zip
rm -rf wordpress
wget https://wordpress.org/latest.zip

#DESCOMPRIMIR EL ARCHIVO
apt-get install -y unzip
unzip latest.zip

#DEFINIMOS LAS VARIABLES
NOMBRE_BD=wordpress
NOMBRE_USU=word
PASSWD_BD=123456
SERVER=192.168.33.12
```

```
#NOS METEMOS DENTRO DE WORDPRESS Y CAMBIAMOS EL NOMBRE DE LAS VARIABLES POR
EL DE LAS NUESTRAS EN EL ARCHIVO wp-config.php EL CUAL HEMOS RENOMBRADO.
cd wordpress
mv wp-config-sample.php wp-config.php
sed -i "s/database_name_here/$NOMBRE_BD/" wp-config.php
sed -i "s/username_here/$NOMBRE_USU/" wp-config.php
sed -i "s/password_here/$PASSWD_BD/" wp-config.php
sed -i "s/localhost/$SERVER/" wp-config.php

#COPIAMOS LA CARPETA WORDPRESS Y LE DAMOS PERMISOS DE WWW-DATA
cp . /var/www/html -R
chown www-data:www-data /var/www/html/* -R

#BORRAMOS EL INDEX.HTML
rm -rf /var/www/html/index.html
```

En el otro provision debemos de poner las líneas para instalar y configurar Mysql.

Provision para NFS Server:

```
#!/bin/bash

# Incluyo el provision-for-apache
source /vagrant/provision/script-apache.sh

#Instalo nfs server

apt-get update
apt-get install -y nfs-kernel-server

# Compartimos la carpeta wp-content
chown nobody:nogroup /var/www/html/wp-content

#Modificamos el archivo para compartir
cp /vagrant/archivos/exports /etc/ -f
/etc/init.d/nfs-kernel-server restart
```

Provision para NFS Client:

```
#!/bin/bash

# Incluyo el provision-for-apache
source /vagrant/provision/script-apache.sh

#Instalo nfs cliente
```

```
apt-get update
apt-get install -y nfs-common

# Monto la carpeta
mount 192.168.33.10:/var/www/html/wp-content /var/www/html/wp-content
echo "192.168.33.10:/var/www/html/wp-content /var/www/html/wp-content nfs
auto,nofail,noatime,nolock,intr,tcp,actimeo=1800 0 0" >> /etc/fstab
```

Provision para Mysql:

```
#!/bin/bash
apt-get update
apt-get -y install debconf-utils

DB_ROOT_PASSWD=123456
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;"
mysql -uroot mysql -p$DB_ROOT_PASSWD <
/vagrant/provision/script/wordpress.sql

/etc/init.d/mysql restart
```

Y el script .sql para crear el usuario y la bd

```
DROP DATABASE IF EXISTS wordpress;
CREATE DATABASE wordpress DEFAULT CHARACTER SET utf8 COLLATE
utf8_general_ci;

USE mysql;
CREATE USER 'word'@'%' IDENTIFIED BY '123456';
GRANT ALL PRIVILEGES ON wordpress.* TO word@'%' IDENTIFIED BY '123456';
FLUSH PRIVILEGES;
```

Provision para el balanceador de carga

```
#!/bin/bash

apt-get update
apt-get install -y apache2

#ACTIVAMOS LOS MODULOS EN APACHE
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

#ELIMINAMOS 000-DEFAULT.CONF
rm -rf /etc/apache2/sites-enabled/000-default.conf
cp /vagrant/archivos/000-default.conf /etc/apache2/sites-enabled/

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

ARCHIVO 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
# AQUI PONEMOS LA IP DEL PRIMER SERVIDOR WEB
BalancerMember http://192.168.33.10

# Server 2
# AQUI PONEMOS LA IP DEL SEGUNDO SERVIDOR WEB
BalancerMember http://192.168.33.11
</Proxy>

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

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

Paso 3: Levantar las máquinas

El último paso es levantar las máquinas para ver que se realizan bien los scripts y comprobar que se comunican bien las dos máquinas entrando en una de la IP del balanceador de carga, instalar el WP y disfrutar.