

Parcial 1: Vagrant

Aprovisionamiento en vagrant: Balanceador de carga

Universidad Icesi

Facultad de ingeniería - Ingeniería telemática - Sistemas Distribuidos

Jaime Vélez Escandón

Código: A00268524

Enlace de git: <https://github.com/jaimevlz7/parcial1-sd-2017a>

Inicialmente se configura el archivo "Vagrantfile" con las 4 máquinas que se usaran (2 maquinas web, 1 de base de datos y 1 balanceadora de carga) de la siguiente manera (Cada uno con su IP y configuraciones adecuadas):

```
config.vm.define :centos_db do |db|
  db.vm.box = "centosparcial"
  db.vm.network :private_network, ip: "192.168.33.14"
  db.vm.network "public_network", bridge: "enp5s0", ip: "192.168.131.225"
  db.vm.provider :virtualbox do |vb|
    vb.customize ["modifyvm", :id, "--memory", "512", "--cpus", "1", "--name", "centos-db" ]
  end
  config.vm.provision :chef_solo do |chef|
    chef.cookbooks_path = "cookbooks"
    chef.add_recipe "db"
  end
end
```

Imagen 1. Configuración base de datos.

```
config.vm.define :centos_web1 do |web1|
  web1.vm.box = "centosparcial"
  web1.vm.network :private_network, ip: "192.168.33.15"
  web1.vm.network "public_network", bridge: "enp5s0", ip: "192.168.131.226"
  web1.vm.provider :virtualbox do |vb|
    vb.customize ["modifyvm", :id, "--memory", "512", "--cpus", "1", "--name", "centos-web1" ]
  end
  config.vm.provision :chef_solo do |chef|
    chef.cookbooks_path = "cookbooks"
    chef.add_recipe "web1"
  end
end
```

Imagen 2. Configuración web 1.

```
config.vm.define :centos_web2 do |web2|
  web2.vm.box = "centosparcial"
  web2.vm.network :private_network, ip: "192.168.33.16"
  web2.vm.network "public_network", bridge: "enp5s0", ip: "192.168.131.227"
  web2.vm.provider :virtualbox do |vb|
    vb.customize ["modifyvm", :id, "--memory", "512", "--cpus", "1", "--name", "centos-web2" ]
  end
  config.vm.provision :chef_solo do |chef|
    chef.cookbooks_path = "cookbooks"
    chef.add_recipe "web2"
  end
end
```

Imagen 3. Configuración web 2.

```

config.vm.define :centos_balancer do |balancer|
  balancer.vm.box = "centosparcial"
  balancer.vm.network :private_network, ip: "192.168.33.17"
  balancer.vm.network "public_network", bridge: "enp5s0", ip: "192.168.131.228"
  balancer.vm.provider :virtualbox do |vb|
    vb.customize ["modifyvm", :id, "--memory", "512", "--cpus", "1", "--name", "centos-balancer" ]
  end
  config.vm.provision :chef_solo do |chef|
    chef.cookbooks_path = "cookbooks"
    chef.add_recipe "balancer"
  end
end

```

Imagen 4. Configuración balanceador de carga.

Para el desarrollo de este examen, se debe tener en cuenta que se debe automatizar el despliegue, por lo que la instalación, configuración y manejo de las máquinas se debe realizar de manera automática; por ejemplo para las 2 máquinas web, se debe instalar apache ("httpd"), mysql (con sus configuraciones de IP y usuario para acceder a la base de datos) y php. Teniendo en cuenta los "Template" y los "cookbook" que se definen para dar solución (web1 y web2 tienen estructura igual, solo cambia en los atributos de la máquina).

```

package 'httpd'
package 'php'
package 'php-mysql'
package 'mysql'

|
service 'httpd' do
  action [:enable, :start]
end

bash 'open port' do
  code <<-EOH
  iptables -I INPUT 5 -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT
  service iptables save
EOH
end

cookbook_file 'var/www/html/index.html' do
  source 'index.html'
  mode 0644
end

cookbook_file 'var/www/html/info.php' do
  source 'info.php'
  mode 0644
end

template 'var/www/html/select.php' do
  source 'select.php.erb'
  mode 0777
  variables(
    usuarioweb_web: node[:web][:usuarioweb_web],
    ip_web: node[:web][:ip_web],
    passwordweb_web: node[:web][:passwordweb_web]
  )
end

```

Imagen 5. Declaración de configuración inicial de web1.

```

default[:web1][:usuarioweb_web]='icesi'
default[:web1][:ip_web]='192.168.131.225'
default[:web1][:passwordweb_web]='12345'

```

Imagen 6. Atributos web1

Para base de datos se muestra a continuación la configuración, con las librerías, templates y cookbooks necesarios, además de la plantilla de variables mostradas a continuación:

```

package 'mysql-server'
bash 'extract_module' do
  code <<-EOH
  iptables -I INPUT 5 -p tcp -m state --state NEW -m tcp --dport 3306 -j ACCEPT
  service iptables save
  EOH
end
service 'mysqld' do
  action [:enable, :start]
end
package 'expect'
template '/tmp/configure_mysql.sh' do
  source 'configure_mysql.sh.erb'
  mode 0777
  variables(
    password_db: node[:db][:password_db]
  )
end
bash 'configure mysql' do
  cwd '/tmp'
  code <<-EOH
  ./configure_mysql.sh
  EOH
end
template '/tmp/create_schema.sql' do
  source 'create_schema.sql.erb'
  mode 0777
  variables(
    usuarioweb_db: node[:db][:usuarioweb_db],
    ip_db: node[:db][:ip_db],
    ip_db1: node[:db][:ip_db1],
    passwordweb_db: node[:db][:passwordweb_db]
  )
end
bash 'create schema' do
  cwd '/tmp'
  code <<-EOH
  cat create_schema.sql | mysql -u root -pdistribuidos
  EOH
end

```

Imagen 9. Declaración de configuración inicial db

```

default[:db][:password_db]='distribuidos'
default[:db][:usuarioweb_db]='icesi'
default[:db][:ip_db]='192.168.131.226'
default[:db][:passwordweb_db]='12345'
default[:db][:ip_db1]='192.168.131.227'

```

Imagen 8. Atributos db

```

CREATE database database1;
USE database1;
CREATE TABLE example(
  id INT NOT NULL AUTO_INCREMENT,
  PRIMARY KEY(id),
  name VARCHAR(30),
  age INT);
INSERT INTO example (name,age) VALUES ('flanders',25);
-- http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO:_Ch34:_Basic_MySQL_Configuration
GRANT ALL PRIVILEGES ON *.* to '<%=@usuarioweb_db%>'@'<%=@ip_db%>' IDENTIFIED by '<%=@passwordweb_db%>';
GRANT ALL PRIVILEGES ON *.* to '<%=@usuarioweb_db%>'@'<%=@ip_db1%>' IDENTIFIED by '<%=@passwordweb_db%>';

```

Imagen 9. plantilla db

Para el balanceador (“balancer”); se encuentran las librerías, paquetes, templates y cookbooks necesarios para el funcionamiento. Adicionalmente está el documento donde se configura apache, como un proxy balancer.

```

package 'httpd'
package 'php'
package 'php-mysql'
package 'mysql'

service 'httpd' do
  action [:enable, :start]
end

cookbook_file 'etc/httpd/conf.d/lb.conf' do
  source 'lb.conf'
  mode 0644
end

bash 'open port' do
  code <<-EOH
  iptables -I INPUT 5 -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT
  service iptables save
  sudo service httpd graceful
  service httpd graceful
EOH
end

```

Imagen 10. Configuración inicial balancer.

```

ProxyRequests off
<Proxy balancer://192.168.131.228>
  BalancerMember http://192.168.131.226:80
  BalancerMember http://192.168.131.227:80
  ProxySet lbmethod=byrequests
</Proxy>

<Location /balancer-manager>
  SetHandler balancer-manager
</Location>
ProxyPass /balancer-manager !
ProxyPass / balancer://192.168.131.228/

```

Imagen 11. Plantilla de configuración balancer.

A continuación se encuentran las capturas que muestran el correcto funcionamiento del balanceador además de la interfaz de administración del mismo (“Balancer-Manager”)

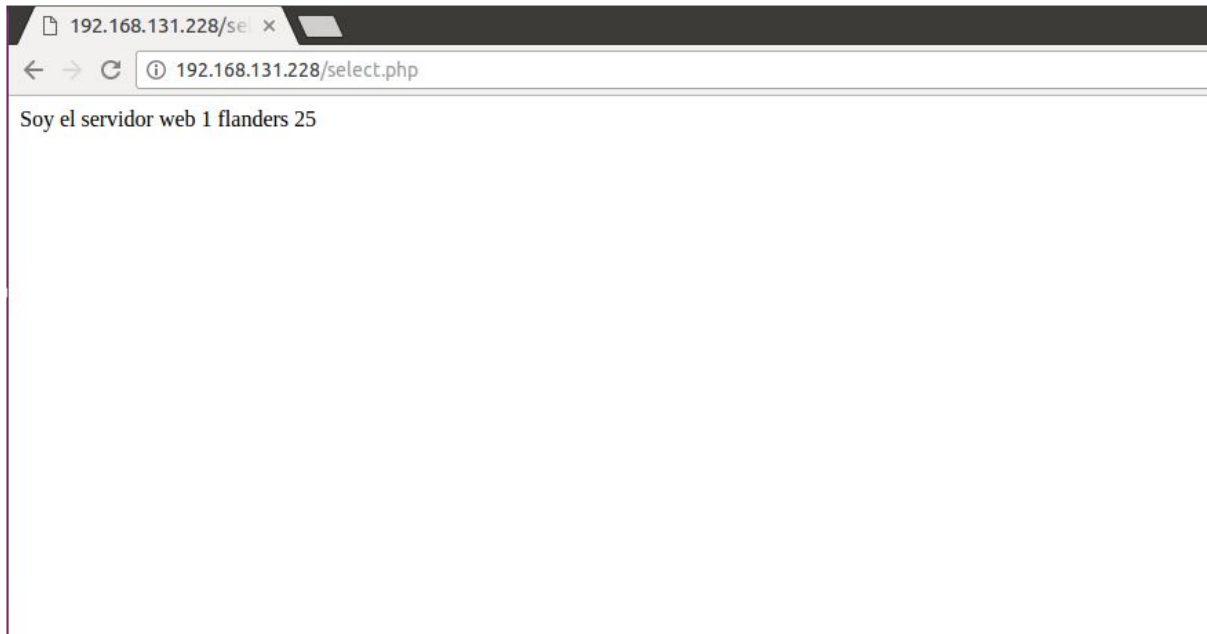


Imagen 12. Captura de balanceador con web 1.

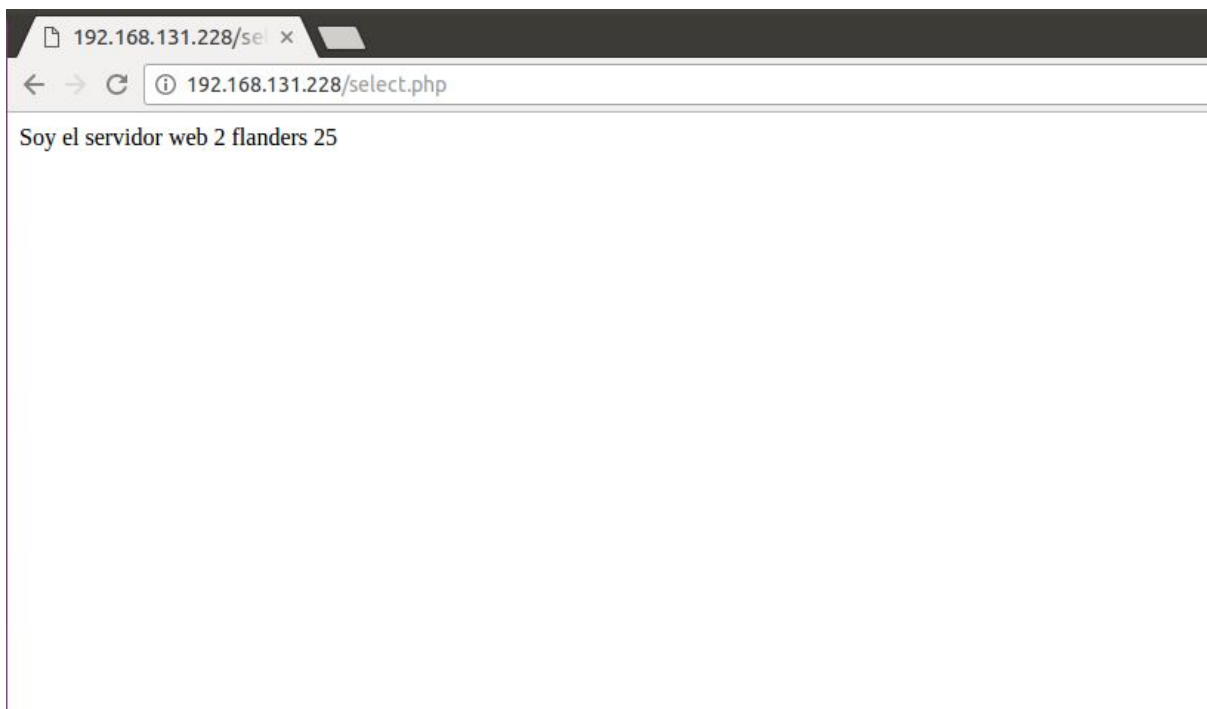


Imagen 13. Captura de balanceador con web 2.

Balancer Manager

192.168.131.228/balancer-manager

Load Balancer Manager for 192.168.131.228

Server Version: Apache/2.2.15 (Unix) DAV/2 PHP/5.3.3
Server Built: Aug 13 2013 17:27:11

LoadBalancer Status for balancer://webfarm

StickySession	Timeout	FailoverAttempts	Method
-	0	1	byrequests

Worker URL	Route	RouteRedir	Factor	Set	Status	Elected To	From
http://192.168.131.226		1	0	Init	Ok	16	8.0K 880
http://192.168.131.227		1	0	Init	Ok	15	7.5K 1.1K

Apache/2.2.15 (CentOS) Server at 192.168.131.228 Port 80

Imagen 13. Captura de balancer manager.