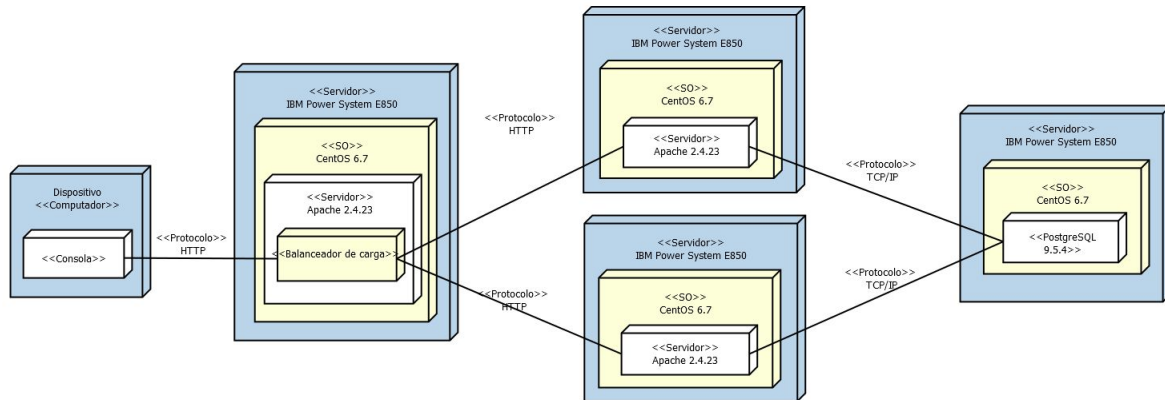


Taller 1

Integrantes: Jaime Andres Aristizabal 11104055

Diego Esteban Delgado 12107019

GitHub: https://diegoedelgado.github.io/distributed_system/



1. Escriba el archivo Vagrantfile para realizar el aprovisionamiento, teniendo en cuenta definir: máquinas a aprovisionar, interfaces solo anfitrión, interfaces tipo puente, declaración de cookbooks, variables necesarias para plantillas.

Para el desarrollo del aprovisionamiento requerido en el taller, se levantaron 4 máquinas virtuales. Una máquina que servirá de balanceador de carga de otras 2 máquinas que tendrán copias similares del servicio web, y una última máquina que se encarga de almacenar el servicio de la base de datos. El archivo de vagrantfile se adjuntará en la entrega, sin embargo se presentarán pantallazos de la configuración de cada máquina.

La configuración de la primera máquina, el balanceador de carga, es la presentada en la Imagen 1. Dentro de ella se carga un recipe que se llama proxy. Dentro de este recipe levantamos un apache que tiene habilitada la opción de balanceador de carga. Este apache no sirve como servicio WEB, sino como un proxy.

```
Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
  config.vm.define :centos_balancer do |node|
    node.vm.box = "centos6.4"
    node.vm.network :private_network, ip: "192.168.56.51"
    node.vm.network "public_network", :bridge => "eth2", ip: "192.168.131.51", :auto_config => "false", :netmask => "255.255.255.0"
    node.vm.provider :virtualbox do |vb|
      vb.customize ["modifyvm", :id, "--memory", "1024", "--cpus", "2", "--name", "centos_balancer" ]
    end
    config.vm.provision :chef_solo do |chef|
      chef.cookbooks_path = "cookbooks"
      chef.add_recipe "mirror"
      chef.add_recipe "proxy"
      chef.json = {"aptmirror" => {"server" => "192.168.131.254", "webip1" => "192.168.56.52", "webip2" => "192.168.56.53"}}
    end
  end
end
```

Imagen 1, Configuración balanceador de carga VagrantFile

La configuración de la segunda máquina es similar a la tercera, y es presentada en la Imagen 2. Dentro de ella se cargan los servicio WEB de apache con diferentes index, la maquina 2 o centos_web_1 tiene el index de facebook.com. y la máquina 3

o centos_web_2 tiene el index de google.com. Para cargar los dos index diferentes, se realizaron dos recipes, uno llamado apache y otro llamado apache2. Ambos recipes son idénticos pero cambian en su configuración de arranque y el index.

```

config.vm.define :centos_web_1 do |node|
  node.vm.box = "centos6.4"
  node.vm.network :private_network, ip: "192.168.56.52"
  node.vm.network "public_network", :bridge => "eth2", ip: "192.168.131.52", :auto_config => "false", :netmask => "255.255.255.0"
  node.vm.provider :virtualbox do |vb|
    vb.customize ["modifyvm", :id, "--memory", "1024", "--cpus", "2", "--name", "centos_web_1"]
  end
  config.vm.provision :chef_solo do |chef|
    chef.cookbooks_path = "cookbooks"
    chef.add_recipe "mirror"
    chef.add_recipe "apache"
    chef.json = {"aptmirror" => {"server" => "192.168.131.254"}}
  end
end

```

Imagen 2, Configuración servicio WEB máquina 2 y 3, VagrantFile

Por último en el VagrantFile se define la base de datos que estará sobre la plataforma PostgreSQL. La configuración del VagrantFile es la presentada en la imagen 3.

```

config.vm.define :centos_database do |db|
  db.vm.box = "centos6.4"
  db.vm.network :private_network, ip: "192.168.56.54"
  db.vm.network :public_network, :bridge => "eth2", ip:"192.168.131.54", :auto_config => "false", :netmask => "255.255.255.0"
  db.vm.provider :virtualbox do |vb|
    vb.customize ["modifyvm", :id, "--memory", "1024", "--cpus", "1", "--name", "centos_database"]
  end
end

config.vm.provision :chef_solo do |chef|
  chef.cookbooks_path = "cookbooks"
  chef.add_recipe "mirror"
  chef.add_recipe "postgres"
  chef.json = {"aptmirror" => {"server" => "192.168.131.254"}}
end
end

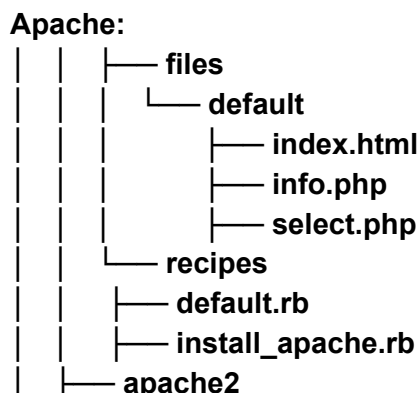
```

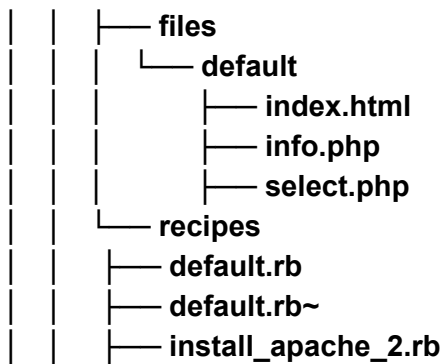
Imagen 3, Configuración base de datos PostgreSQL en la máquina 4. VagrantFile

2. Escriba los cookbooks necesarios para realizar la instalación de los servicios solicitados

A Continuación se describen los archivos utilizados en las recetas, partimos de las siguientes recetas:

Como se requería tener **dos** servidores Web, partimos de una máquina **centos_web_1** y **centos_web_2** respectivamente por lo que se tomó la decisión de partir este aprovisionamiento en dos recetas apache y apache2, respectivamente.





Partimos de la estructura anterior de archivos para esta recetas, para ambas recetas encontraremos las mismas instrucciones y difieren en el archivo index.html, este archivo para el caso de **centos_web_1** corresponde a la plantilla de www.Facebook.com y en el caso de **centos_web_2** a la pagina de www.google.com con el fin de diferenciar una máquina de la otra como se observa acontinuación.



Imagen 4, Máquina centos_web_1: Facebook.com

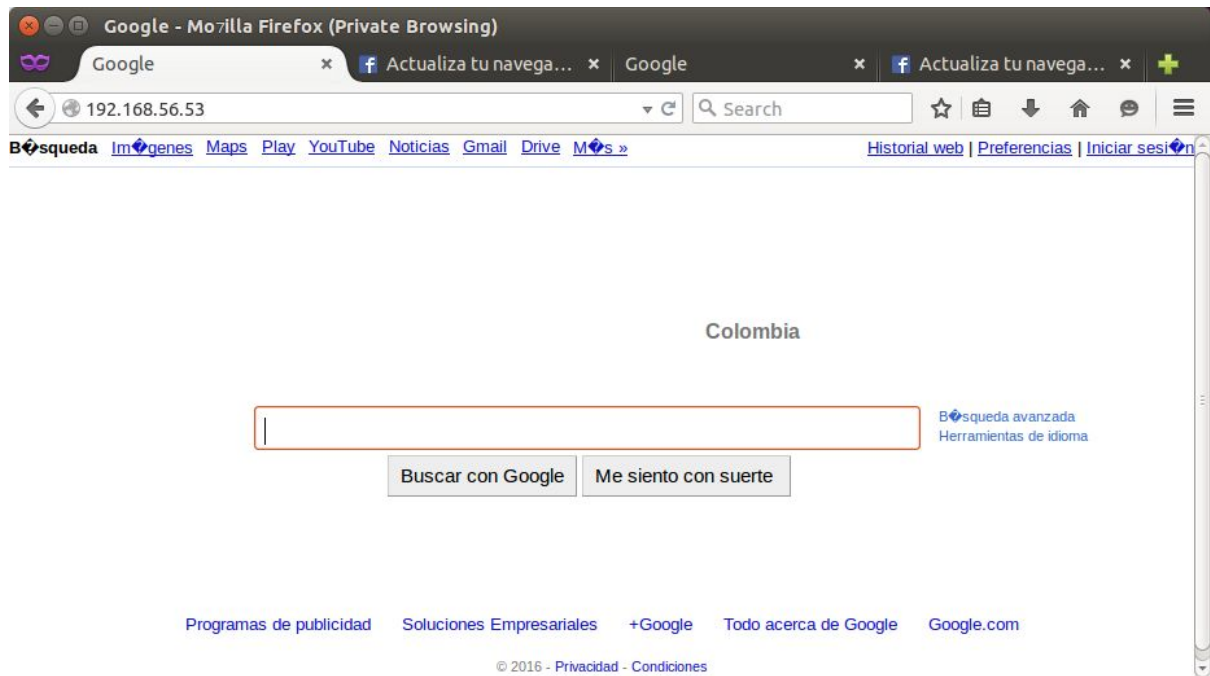


Imagen 5, Máquina centos_web_2 : Google.com

Posterior a esto contamos con la receta usada para el despliegue de estas máquinas virtuales:

```
config.vm.define :centos_web_1 do |node|
  node.vm.box = "centos6.4"
  node.vm.network :private_network, ip: "192.168.56.52"
  node.vm.network "public_network", :bridge => "eth2", ip:"192.168.131.52", :auto_config => "false", :netmask => "255.255.255.0"
  node.vm.provider :virtualbox do |vb|
    vb.customize ["modifyvm", :id, "--memory", "1024", "--cpus", "2", "--name", "centos_web_1" ]
  end
  config.vm.provision :chef_solo do |chef|
    chef.cookbooks_path = "cookbooks"
    chef.add_recipe "mirror"
    chef.add_recipe "apache"
    chef.json = {"aptmirror" => {"server" => "192.168.131.254"}}
  end
end

config.vm.define :centos_web_2 do |node|
  node.vm.box = "centos6.4"
  node.vm.network :private_network, ip: "192.168.56.53"
  node.vm.network "public_network", :bridge => "eth2", ip:"192.168.131.53", :auto_config => "false", :netmask => "255.255.255.0"
  node.vm.provider :virtualbox do |vb|
    vb.customize ["modifyvm", :id, "--memory", "1024", "--cpus", "2", "--name", "centos_web_2" ]
  end
  config.vm.provision :chef_solo do |chef|
    chef.cookbooks_path = "cookbooks"
    chef.add_recipe "mirror"
    chef.add_recipe "apache2"
    chef.json = {"aptmirror" => {"server" => "192.168.131.254"}}
  end
end
```

Imagen 6, Despliegue máquinas **centos_web_x**, archivo Vagrantfile

En cuanto al archivo de instalación de cada apache: **install_apache.rb** contamos con:

```

yum_package 'httpd' do
  action :install
end
yum_package 'php' do
  action :install
end
yum_package "php-pgsql" do
  action :install
end
execute 'pg' do
  command 'yum localinstall http://yum.postgresql.org/9.4/redhat/rhel-6-x86_64/pgdg-centos94-9.4-1.noarch.rpm -y'
end

yum_package 'postgresql94' do
  action :install
end

service 'httpd' do
  supports :status => true, :restart => true, :reload => true
  action [ :enable, :start ]
end

bash "open port" do
  user "root"
  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
  owner "root"
  group "wheel"
end

cookbook_file "/var/www/html/info.php" do
  source "info.php"
  mode 0644
  owner "root"
  group "wheel"
end

cookbook_file "/var/www/html/select.php" do
  source "select.php"
  mode 0644
  owner "root"
  group "wheel"
end
end

```

Imagen 7, Archivo **install_apache.rb**

Para realizar el proceso de instalación de paquetes realizamos un cookbook con el fin de usar la máquina del mirror situada en uno de los pc's de la sala, para este proceso usamos la siguiente receta:

```

1 |template "/etc/yum.repos.d/CentOS-Base.repo" do
2   source "CentOS-Base.repo.erb"
3   mode 0644
4   owner "root"
5   group "wheel"
6   variables(
7     :ip_mirror => "#{node[:aptmirror][:server]}"
8   )
9   end
10

```

Imagen 8, Archivo **updaterepo.rb**

Este script se encarga de reemplazar la ruta de los repositorios en las máquinas buscando que en lugar de ir a los repositorios oficiales y descargar los paquetes necesarios deba ir a la máquina local con estos paquetes agilizando así el proceso de aprovisionamiento. La máquina a la que se apunta es **centos64_updated4** la cual tiene la dirección de red 192.168.131.254.

Configuración CookBook PostgreSQL:

La configuración de la máquina que sostiene la base de datos, se basó en el cookbook pasado por el profesor. Sin embargo se realizaron algunas modificaciones las cuales se mencionan a continuación las cuales irán en orden de por carpeta. Cabe aclarar que cada uno de los archivos de configuración se anexaron en la entrega de este documento, para posterior validación. La distribución de carpetas con archivos es la presentada en la imagen X:

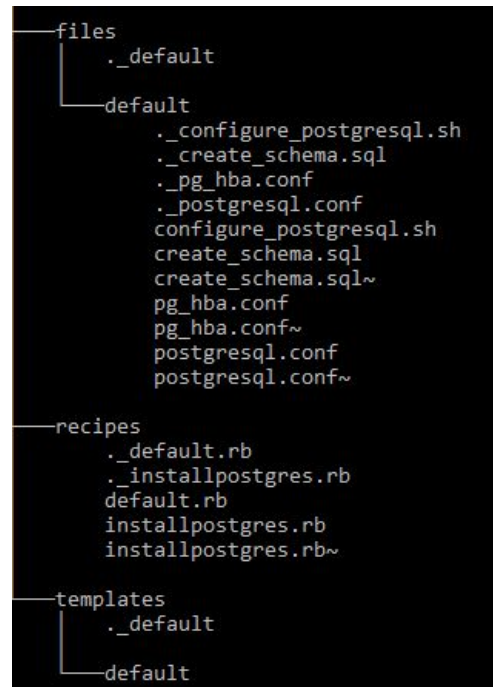


Imagen 9, Tree de carpeta postgresQL

Los archivos configure_postgresql.sh y create_schema.sql no tuvieron ningún cambio. Los cambios se dieron en pg_hba.conf, los cuales incluyeron la validación de las direcciones IP que podían acceder a la base de datos. La imagen XX representa el cambio realizado en las direcciones IP.

#	TYPE	DATABASE	USER	ADDRESS	METHOD
# "local" is for Unix domain socket connections only					
local	all		all		trust
# IPv4 local connections:					
host	all		all	127.0.0.1/32	ident
host	all		all	192.168.56.52/24	trust
host	all		all	192.168.56.53/24	trust
host	all		all	192.168.56.51/24	trust
host	all		all	192.168.131.53/24	trust
host	all		all	192.168.131.52/24	trust
host	all		all	192.168.131.51/24	trust
# IPv6 local connections:					
host	all		all	:::1/128	ident

Imagen 10, Cambios en pg_hba.conf

En el archivo postgresql.conf, se realizaron cambios al principio que indican cuál va a ser la dirección IP de entrada, el puerto que está escuchando y el número de conexiones. La imagen XX representan los cambios sobre el archivo.

```
# - Connection Settings -
listen_addresses = '*'      # what IP address(es) to listen on;
                             # comma-separated list of addresses;
                             # defaults to 'localhost'; use '*' for all
                             # (change requires restart)
port = 5432                 # (change requires restart)
max_connections = 100       # (change requires restart)
```

Imagen 11, Cambios en postgresql.conf

Dentro de la carpeta recipes se realizaron cambios en installpostgres.rb. Estos cambios corresponden al reinicio del servicio al final. Nos dimos cuenta que cuando terminaba de levantar las máquinas, el servicio de postgresql quedaba detenido por los cambios en los archivos de configuración, por lo tanto procedimos a reiniciar el servicio. Esa fue la línea que agregamos al final. La imagen XX representa los cambios realizados en el archivo installpostgres.rb.

```
bash "postgres restart" do
  user "root"
  code <<-EOH
  service postgresql-9.4 restart
  EOH
end
```

Imagen 12, Cambios en installpostgres.rb

Con esto se finaliza la configuración de la máquina que sostiene el servicio de la base de datos en el vagrantfile.

Configuración CookBook Proxy:

La configuración de la máquina para el balanceador se basó en el cookbook de apache, esto es porque el servicio de balanceo de carga según el esquema dado en el taller, debía estar basado en apache. Lo que se implementó en el apache fue un servicio proxy, con dos equipos balanceadores. El esquema de archivos que se utilizó es el presentado en la imagen XX.



Imagen 13, tree de carpeta Proxy.

Se realizaron modificaciones en el `install_apache.rb`. Como se debía pasar al `httpd.conf` las direcciones de las máquinas que iban a ser balanceo de carga, lo que se realizó fue un paso de variables a través de chef. Se cargo un json desde el vagrantfile con las ips de las máquinas que realizan el balanceo, la imagen XX presenta la variable json.

```
chef.json = {"aptmirror" => {"server" => "192.168.131.254", "webip1" => "192.168.56.52", "webip2" => "192.168.56.53"}}
```

Imagen 14, configuracion vagrantfile paso de ips.

Este json era también pasado por medio del `install_apache.rb` cuando se cargaba el `httpd.conf`. La imagen XX representa el paso de variable.

```
template "/etc/httpd/conf/httpd.conf" do
  source "httpd.conf"
  mode 0644
  owner "root"
  group "wheel"
  variables({
    :ip_web_1 => "#{node[:aptmirror][:webip1]}",
    :ip_web_2 => "#{node[:aptmirror][:webip2]}",
  })
end
```

Imagen 15, Configuración `install_apache.rb`

Estas variables a su vez llegaban al archivo de `httpd.conf` para ser usadas como las ips de los balanceadores de carga. Y como se había explicado, se utiliza el apache como servidor proxy y se configura para que reconozca dos servidores que harán el balanceo de la carga dentro del sistema. El esquema de configuración se presenta en la imagen XX.


```

ProxyRequests Off
<Proxy balancer://mycluster>
    # WebHead1
    BalancerMember http://<%= @ip_web_1 %>

    # WebHead2
    BalancerMember http://<%= @ip_web_2 %>

    ProxySet lbmethod=byrequests
</Proxy>

# Point of Balance
# This setting will allow to explicitly name the
# the location in the site that we want to be
# balanced, in this example we will balance "/"
# or everything in the site.
<Location /balancer-manager>

    SetHandler balancer-manager
</Location>

ProxyPass /balancer-manager !
ProxyPass / balancer://mycluster/

</VirtualHost>

```

Imagen 16, Configuración servidor proxy.

Estos fueron todos los cambios realizados en el recetas de proxy, para que apache funciona como balanceador de carga. Los resultados del balanceo fueron los presentados en las siguientes imágenes:

```

centos_database:
centos_database: Inserting generated public key within guest...
centos_database: Removing insecure key from the guest if its present...
centos_database: Key Inserted! Disconnecting and reconnecting using new SSH key...
==> centos_database: Machine booted and ready!
==> centos_database: Checking for guest additions in VM...
==> centos_database: Configuring and enabling network interfaces...
==> centos_database: Mounting shared folders...
centos_database: /vagrant => /home/distribuidos/Documentos/SisDistri/Taller1/act7
centos_database: /tmp/vagrant-chef/537715698930676b4eccc330a01084e2/cookbooks => /home/distribuidos/Documentos/SisDistri/Taller1/act7/cookbooks
==> centos_database: Running provisioner: chef-solo...
==> centos_database: Detected Chef (latest) is already installed
Generating chef JSON
and uploading...
==> centos_database: Running chef-solo...
==> centos_database: [2016-08-31T23:38:24+00:00] INFO: Forking chef instance to converge...
==> centos_database: [2016-08-31T23:38:24+00:00] INFO: *** Chef 11.8.0 ***
==> centos_database: [2016-08-31T23:38:24+00:00] INFO: Chef-client pid: 3467
==> centos_database: [2016-08-31T23:38:25+00:00] INFO: Setting the run list to ["recipe[mirror]", "recipe[postgres]"] from JSON
==> centos_database: [2016-08-31T23:38:25+00:00] INFO: Run List is [recipe[mirror], recipe[postgres]]
==> centos_database: [2016-08-31T23:38:25+00:00] INFO: Run List expands to [mirror, postgres]
==> centos_database: [2016-08-31T23:38:25+00:00] INFO: Starting Chef Run for localhost
==> centos_database: [2016-08-31T23:38:25+00:00] INFO: Running start handlers
==> centos_database: [2016-08-31T23:38:25+00:00] INFO: Start handlers complete.
==> centos_database: [2016-08-31T23:38:25+00:00] WARN: Cloning resource attributes for bash[open port] from prior resource (CHEF-3694)
==> centos_database: [2016-08-31T23:38:25+00:00] WARN: Previous bash[open port]: /tmp/vagrant-chef/537715698930676b4eccc330a01084e2/cookbooks/postgres/recipes/installpostgres.rb:29:in `from_file'
==> centos_database: [2016-08-31T23:38:25+00:00] WARN: Current bash[open port]: /tmp/vagrant-chef/537715698930676b4eccc330a01084e2/cookbooks/postgres/recipes/installpostgres.rb:37:in `from_file'
==> centos_database: [2016-08-31T23:38:25+00:00] INFO: template[/etc/yun.repos.d/CentOS-Base.repo] backed up to /var/chef/backup/etc/yun.repos.d/CentOS-Base.repo.chef-20160831233825.170000
==> centos_database: [2016-08-31T23:38:25+00:00] INFO: template[/etc/yun.repos.d/CentOS-Base.repo] updated file contents /etc/yun.repos.d/CentOS-Base.repo
==> centos_database: [2016-08-31T23:39:45+00:00] INFO: yum_package[postgres194-server] installing postgresql94-server-9.4.9-1PGDG.rhel6 from pgdg94 repository
==> centos_database: [2016-08-31T23:39:45+00:00] INFO: execute[initdb] ran successfully
==> centos_database: [2016-08-31T23:39:45+00:00] INFO: execute[nextinitdb] ran successfully
==> centos_database: [2016-08-31T23:39:47+00:00] INFO: execute[nextnextinitdb] ran successfully
==> centos_database: [2016-08-31T23:39:47+00:00] INFO: bash[open port] ran successfully
==> centos_database: [2016-08-31T23:39:47+00:00] INFO: bash[open port] ran successfully
==> centos_database: [2016-08-31T23:39:47+00:00] INFO: cookbook_file[/var/lib/pgsql/9.4/data/pg_hba.conf] backed up to /var/chef/backup/var/lib/pgsql/9.4/data/pg_hba.conf.chef-20160831233947.542416
==> centos_database: [2016-08-31T23:39:47+00:00] INFO: cookbook_file[/var/lib/pgsql/9.4/data/pg_hba.conf] updated file contents /var/lib/pgsql/9.4/data/pg_hba.conf
==> centos_database: [2016-08-31T23:39:47+00:00] INFO: cookbook_file[/var/lib/pgsql/9.4/data/pg_hba.conf] node changed to 711
==> centos_database: [2016-08-31T23:39:47+00:00] INFO: cookbook_file[/var/lib/pgsql/9.4/data/postgresql.conf] backed up to /var/chef/backup/var/lib/pgsql/9.4/data/postgresql.conf.chef-20160831233947.556
798
==> centos_database: [2016-08-31T23:39:47+00:00] INFO: cookbook_file[/var/lib/pgsql/9.4/data/postgresql.conf] updated file contents /var/lib/pgsql/9.4/data/postgresql.conf
==> centos_database: [2016-08-31T23:39:47+00:00] INFO: cookbook_file[/var/lib/pgsql/9.4/data/postgresql.conf] node changed to 711
==> centos_database: [2016-08-31T23:39:47+00:00] INFO: cookbook_file[/tmp/create_schema.sql] created file /tmp/create_schema.sql
==> centos_database: [2016-08-31T23:39:47+00:00] INFO: cookbook_file[/tmp/create_schema.sql] updated file contents /tmp/create_schema.sql
==> centos_database: [2016-08-31T23:39:47+00:00] INFO: cookbook_file[/tmp/create_schema.sql] owner changed to 0
==> centos_database: [2016-08-31T23:39:47+00:00] INFO: cookbook_file[/tmp/create_schema.sql] group changed to 10
==> centos_database: [2016-08-31T23:39:47+00:00] INFO: cookbook_file[/tmp/create_schema.sql] mode changed to 644
==> centos_database: [2016-08-31T23:39:47+00:00] INFO: bash[create schema] ran successfully
==> centos_database: [2016-08-31T23:39:50+00:00] INFO: bash[postgres restart] ran successfully
==> centos_database: [2016-08-31T23:39:50+00:00] INFO: Chef Run complete in 85.768692273 seconds
==> centos_database: [2016-08-31T23:39:50+00:00] INFO: Running report handlers
==> centos_database: [2016-08-31T23:39:50+00:00] INFO: Report handlers complete
==> centos_database: [2016-08-31T23:38:24+00:00] INFO: Forking chef instance to converge...
distribuidos@307C: ~/Documentos/SisDistri/Taller1/act7$

```

Imagen 17, Prueba al correr todo el vagrant



Imagen 18, Balanceo de carga, se ve la .52 desde el balanceador .51

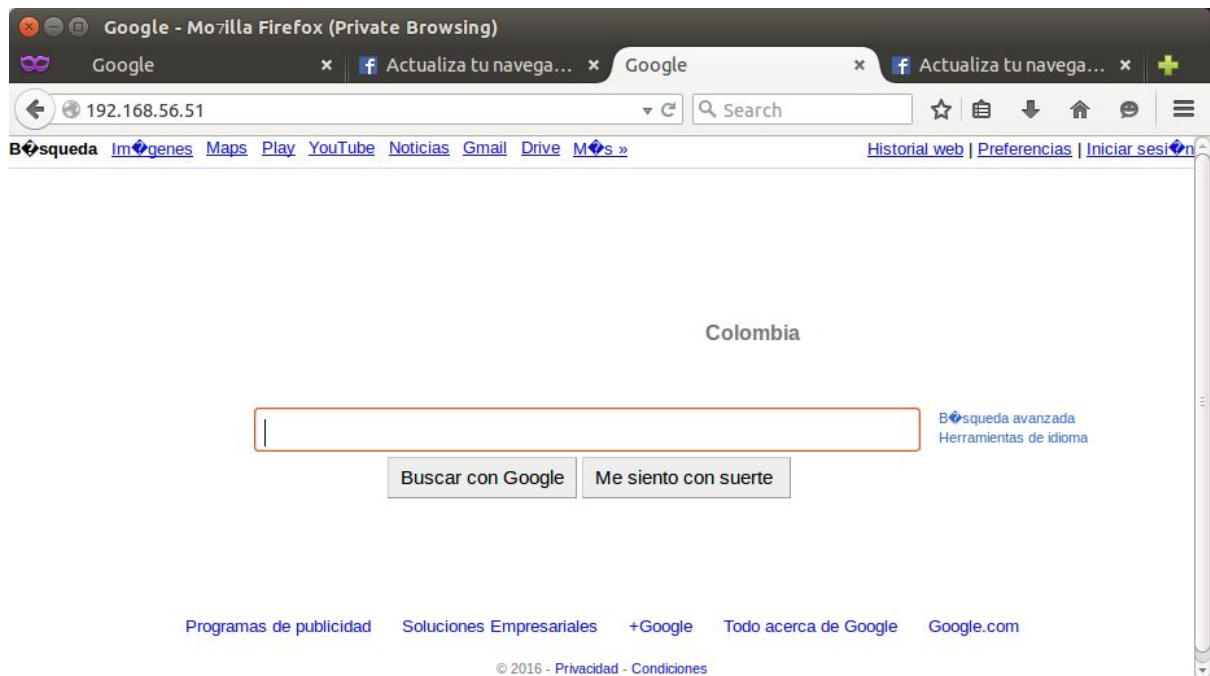


Imagen 19, Balanceo de carga, se ve la .53 desde el balanceador .51