

GUIA DE USUARIO

1. Para llevar a cabo dicho proyecto inicialmente se debe tener instalado un ambiente de máquina virtual para ello se tiene software como Vagrant y VirtualBox.
2. Instalando estas dos herramientas pasamos a implementar las máquinas virtuales para llevar a cabo un balanceador de carga simple, donde se tendrá una máquina virtual donde se instalará el balanceador de carga y dos máquinas virtuales que actuarán como backends. La siguiente figura muestra la información acerca de la implementación de las máquinas virtuales.

```
Vagrant.configure("2") do |config|
  if Vagrant.has_plugin? "vagrant-vbguest"
    config.vbguest.no_install = true
    config.vbguest.auto_update = false
    config.vbguest.no_remote = true
  end
  config.vm.define :nginx do |nginx|
    nginx.vm.box = "bento/centos-7.9"
    nginx.vm.network :private_network, ip: "192.168.100.2"
    nginx.vm.hostname = "nginx"
  end
  config.vm.define :backend1 do |backend1|
    backend1.vm.box = "bento/centos-7.9"
    backend1.vm.network :private_network, ip: "192.168.100.3"
    backend1.vm.hostname = "backend1"
  end
  config.vm.define :backend2 do |backend2|
    backend2.vm.box = "bento/centos-7.9"
    backend2.vm.network :private_network, ip: "192.168.50.4"
    backend2.vm.hostname = "backend2"
  end
end
```

Figura 1. Archivo de máquinas virtuales necesarias.

3. Posterior a implementar dicha configuración, levantamos las máquinas virtuales e iniciamos con la configuración del sistema.
4. Para este caso debemos instalar servicios como nginx y bind, para ello las líneas a ejecutar dentro de la máquina es el siguiente:
 - a. Para la máquina contenedora del balanceador de carga.

Iniciamos ingresando como superusuario a la máquina virtual y ejecutamos el siguiente comando para actualizar paquetes.

```
[root@nginx ~]# sudo yum update
```

Figura 2. Línea para actualizar paquetes.

Instalamos el paquete EPEL.

```
[root@nginx ~]# sudo yum install epel-release -y
```

Figura 3. Línea para instalar EPEL pack.

Procedemos a instalar nginx.

```
[root@nginx ~]# sudo yum install nginx
```

Figura 4. Línea para instalar nginx.

Una vez instalado nginx, realizamos la inicialización del demonio o servicio y además de esto, lo habilitamos para que al reiniciar la máquina esté activo. Se ejecutan las siguientes líneas:

```
[root@nginx ~]# systemctl start nginx
```

```
[root@nginx ~]# systemctl enable nginx
```

Figura 5. Inicializar y habilitar al inicio el servicio de nginx.

Otro servicio a instalar es el DNS el cual es llamado Berkeley Internet Name Domain (BIND), para ello debemos ejecutar la siguiente línea:

```
[root@nginx conf.d]# yum install bind-utils bind-libs bind-*
```

Figura 6. Línea instalación bind.

Para configurar dicho servicio debemos ir al archivo de configuración llamado named.conf, el cual está ubicado en la ruta **/etc/named.conf**, donde la configuración es la siguiente:

```
options {
    listen-on port 53 { 127.0.0.1; 192.168.100.2; };
    listen-on-v6 port 53 { ::1; };
    directory "/var/named";
    dump-file "/var/named/data/cache_dump.db";
    statistics-file "/var/named/data/named_stats.txt";
    memstatistics-file "/var/named/data/named_mem_stats.txt";
    recursing-file "/var/named/data/named.recursing";
    secroots-file "/var/named/data/named.secroots";
    allow-query { localhost; 192.168.100.0/24; };

    //Zona adelante
    zone "pjfinal.com" IN {
        type master;
        file "pjfinal.com.fwd";
    };

    //Zona reversa
    zone "100.168.192.in-addr.arpa" IN {
        type master;
        file "pjfinal.com.rev";
    };
};
```

Figura 7. Archivo de configuración para DNS.

Aparte de esta configuración debemos de crear dichas zonas donde estarán los récords del DNS. Estas zonas se deben crear en la ruta **/var/named**, donde asignamos los nombres como se muestra en la Figura 7 y damos permisos con 755 mediante la línea **chmod 755 pjfinal.com.fwd** y **chmod 755 pjfinal.com.rev**. A continuación se mostrará la configuración empleada tanto para la zona hacia adelante como para la zona reversa.

```
$ORIGIN pjfinal.com.
$TTL 3H
@ IN SOA server.pjfinal.com. root@pjfinal.com. (
    0 ; serial
    1D ; refresh
    1H ; retry
    1W ; expire
    3H ; minimum
)
@ IN NS server.pjfinal.com.
server IN A 192.168.100.2
www IN CNAME server
$ORIGIN 100.168.192.in-addr.arpa.
$TTL 3H
@ IN SOA server.pjfinal.com. root@pjfinal.com. (
    0 ; serial
    1D ; refresh
    1H ; retry
    1W ; expire
    3H ; minimum
)
@ IN NS server.pjfinal.com.
3 IN PTR server.pjfinal.com.
```

Figura 8. Archivo de configuración de zona adelante y reversa respectivamente.

Por último iniciamos el servicio de bind mediante la ejecución de **service named start**.

Teniendo esto se puede iniciar la configuración de del nginx cómo balanceador de carga, para ello debemos ir a una ruta específica, estando en está ruta debemos de crear un archivo el cual en nuestro caso es llamado load-balancing.conf, dentro de realizaremos la configuración pertinente del balanceador de carga, a continuación se mostrará mediante imágenes los pasos realizados y la configuración planteada..

```
/etc/nginx/conf.d

upstream backend {
    server 192.168.100.3;
    server 192.168.50.4;
}

server {
    listen 80;
    server_name server.pjfinal.com;
    server_name www.pjfinal.com;
    server_name 192.168.100.2;

    location / {
        proxy_redirect off;
        proxy_set_header    X-Real-IP $remote_addr;
        proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header    Host $http_host;
        proxy_pass http://backend;
    }
}
```

Figura 9.Ruta y archivo de configuración para nginx cómo balanceador de carga.

b. Para las máquinas backends.

Para el caso de las otras dos máquinas que actuarán cómo backends, instalamos nginx en ambas, pero en este caso no modificaremos ninguna directriz, ya que hará la funcionalidad de trabajar cómo un servicio web, donde serán lanzadas las páginas web. Repetimos los pasos realizados en la Figura 2, 3, 4 y 5.

Cómo se especificó anteriormente en estas dos máquinas el nginx actuará cómo soporte de paginas web, para ello debemos ir a la ruta `/usr/share/nginx/html` en donde encontraremos las páginas, en este caso editaremos la página de `index.html` donde pondremos mensajes diferenciadores para saber para qué servidor o que página está redirigiendo la petición. Para este caso el contenido que se colocó en las páginas fue el siguiente:

<pre></div> <div id="content"> <h1>Backend 1</h1> <h2>Pagina #1</h2> </div> </div></pre>	<pre></div> <div id="content"> <h1>Backend 2</h1> <h2>Pagina 2</h2> </div> </div></pre>
---	--

Figura 10.Contenido de las páginas de los backends.

5. Por último queda realizar las pruebas tanto con curl con la linea `curl server.pjfinal.com` o `curl www.pjfinal.com` , cómo desde el navegador web de preferencia.

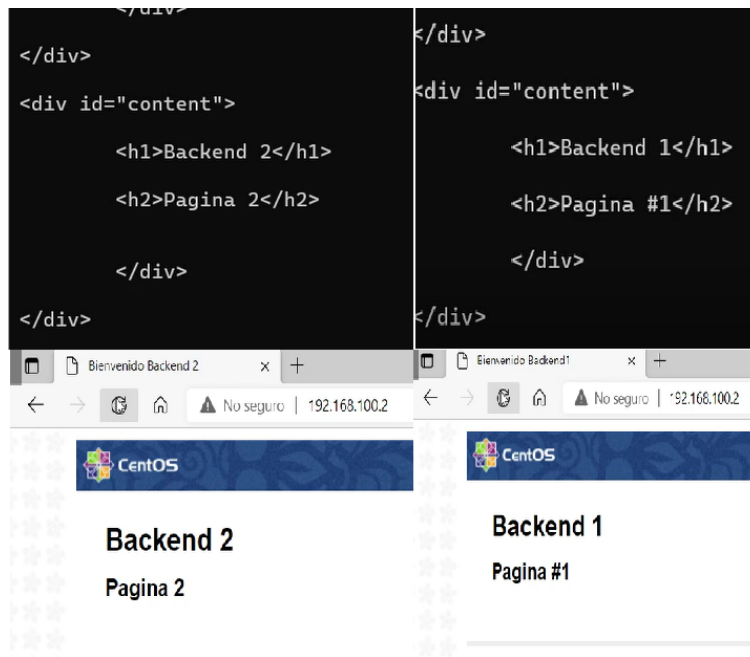


Figura 11. Pruebas realizadas con curl y desplegadas en el navegador web.