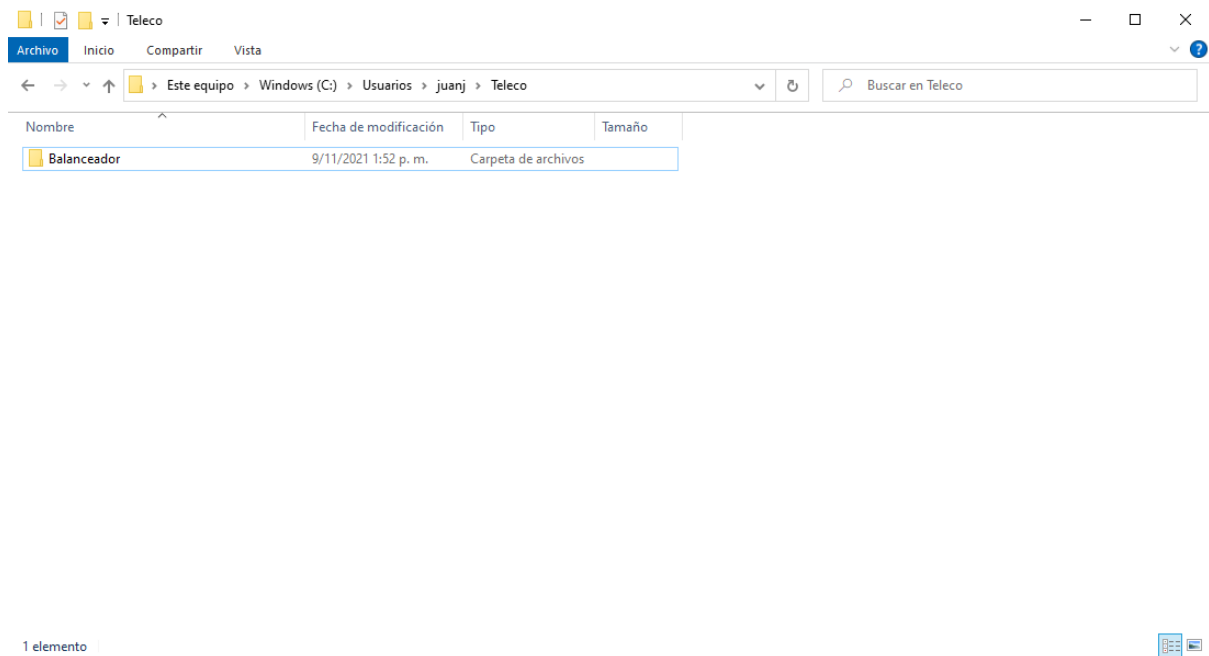


Guia Desarrollo balanceador de carga con Mod_proxy_balancer.

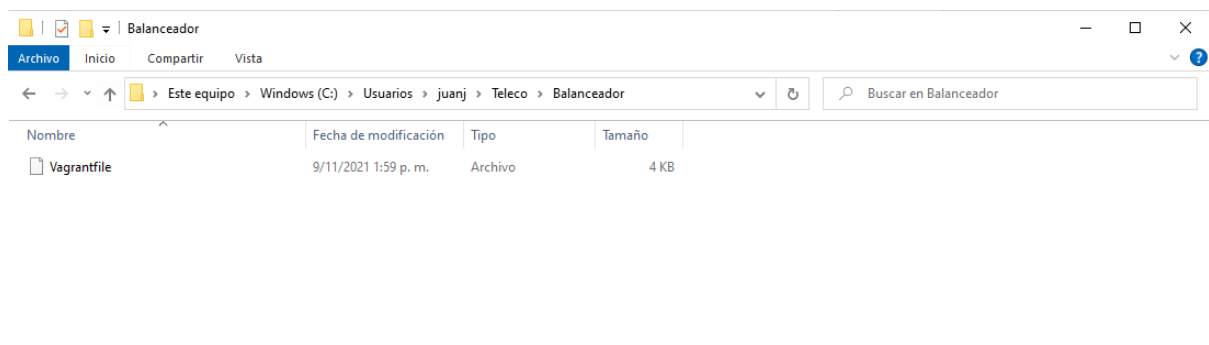
Para empezar creamos una carpeta en la cual alojaremos las máquinas virtuales.



Una vez creada la carpeta, accedemos a ella desde el CMD de windows o desde windows powershell. y ejecutamos el comando “vagrant init” y presionamos la tecla “enter”

```
PS C:\Users\juanj\Documents\Balanceador> vagrant init|
```

Una vez hecho esto, verificamos que en la carpeta creada aparece un archivo de nombre “Vagrantfile”



Modificamos el texto presente en el archivo “Vagrantfile” por un código como el siguiente.

```
1 $script = <<-'SCRIPT'
2 sudo -i
3 yum install -y httpd
4 SCRIPT
5
6 Vagrant.configure("2") do |config|
7
8     config.vm.define :balanceador do |balanceador|
9         balanceador.vm.box = "bento/centos-7.9"
10        balanceador.vm.network :private_network, ip: "192.168.50.20"
11        balanceador.vm.hostname = "balanceador"
12        balanceador.vm.provision "shell", inline: $script
13    end
14
15    config.vm.define :web1 do |web1|
16        web1.vm.box = "bento/centos-7.9"
17        web1.vm.network :private_network, ip: "192.168.50.21"
18        web1.vm.hostname = "web1"
19        web1.vm.provision "shell", inline: $script
20    end
21
22    config.vm.define :web2 do |web2|
23        web2.vm.box = "bento/centos-7.9"
24        web2.vm.network :private_network, ip: "192.168.50.22"
25        web2.vm.hostname = "web2"
26        web2.vm.provision "shell", inline: $script
27    end
28 end
```

En el código se configura las máquinas virtuales que se utilizarán.

Una vez modificado el contenido del Vagrantfile, guardamos los cambios y nos dirigimos al cmd o al windows powershell y una vez ahí ejecutamos el comando “vagrant up” para crear y encender las máquinas virtuales.

```
PS C:\Users\juanj\Documents\Balanceador> vagrant up|
```

Una vez terminado el proceso de creación de las máquinas virtuales verificamos que las máquinas estén encendidas, para ello utilizamos el comando “vagrant status” y se debería ver algo como lo siguiente.

```
PS C:\Users\juanj\Documents\Balanceador> vagrant status
Current machine states:

balanceador          running (virtualbox)
web1                 running (virtualbox)
web2                 running (virtualbox)

This environment represents multiple VMs. The VMs are all listed
above with their current state. For more information about a specific
VM, run `vagrant status NAME`.
PS C:\Users\juanj\Documents\Balanceador> |
```

En caso de que las máquinas no estén encendidas volver a utilizar el comando “vagrant up”.

Una vez encendidas las máquinas virtuales, procedemos a configurar los servidores web, para ello entramos a la máquina web1 con el comando “vagrant ssh web1”.

```
PS C:\Users\juanj\Documents\Balanceador> vagrant ssh web1|
```

Una vez dentro, verificamos que el servicio httpd esté instalado. para ello utilizamos el comando “service httpd status”

```
[vagrant@web1 ~]$ service httpd status|
```

Debería aparecer de la siguiente manera.

```
[vagrant@web1 ~]$ service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2021-11-09 20:34:52 UTC; 29min ago
     Docs: man:httpd(8)
           man:apachectl(8)
  Main PID: 840 (httpd)
   Status: "Total requests: 870; Current requests/sec: 0.5; Current traffic: 204 B/sec"
    CGroup: /system.slice/httpd.service
            └─ 840 /usr/sbin/httpd -DFOREGROUND
               873 /usr/sbin/httpd -DFOREGROUND
               874 /usr/sbin/httpd -DFOREGROUND
               875 /usr/sbin/httpd -DFOREGROUND
               876 /usr/sbin/httpd -DFOREGROUND
               878 /usr/sbin/httpd -DFOREGROUND
              3058 /usr/sbin/httpd -DFOREGROUND
[vagrant@web1 ~]$
```

Si el servicio no está activo lo encendemos de la siguiente manera.
en primer lugar nos logueamos como root con el siguiente comando

```
[vagrant@web1 ~]$ sudo -i
```

y después usamos el comando “service httpd start”

Una vez verificamos que el servicio está instalado y encendido, nos dirigimos al archivo de configuración “httpd.conf” usando el siguiente comando.

```
[root@web1 ~]# vim /etc/httpd/conf/httpd.conf
```

Una vez en el archivo, presionamos la tecla “A” para habilitar su edición y lo modificamos de tal manera que quede como el que está en el repositorio.

Una vez modificado el archivo de configuración presionamos la tecla “ESC” y después “:wq” para salir guardando cambios.

Una vez afuera reiniciamos el servicio con el comando “service httpd restart”

```
[root@web1 ~]# service httpd restart
```

Lo siguiente es crear el archivo index.html, para ello usamos el siguiente comando.

```
[root@web1 html]# vim /var/www/html/index.html
```

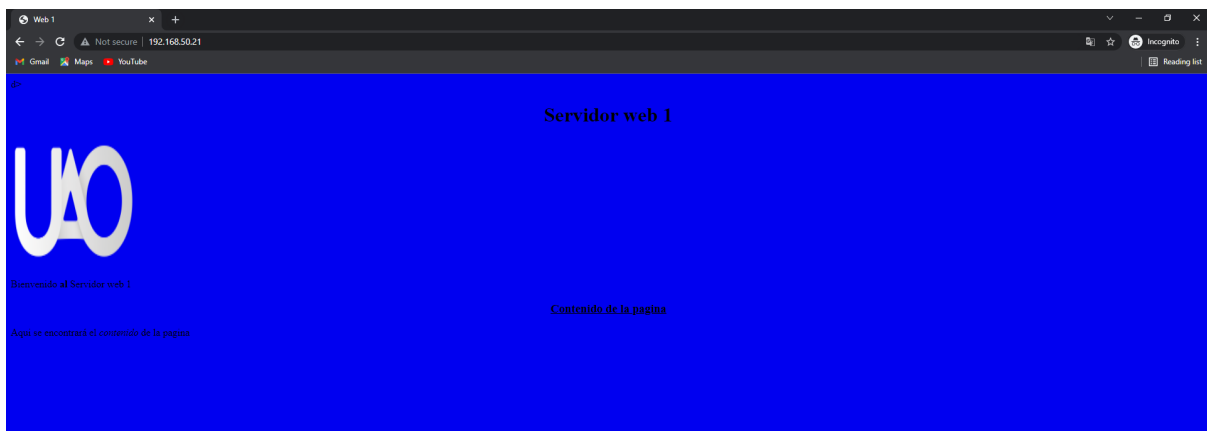
al presionar enter se abrirá el archivo en blanco; presionamos la tecla “A” para habilitar la edición y se modifica el archivo para que quede como el del repositorio.

Una vez hecho esto presionamos la tecla “ESC” y posteriormente “:wq” para salir guardando cambios.

Lo siguiente es darle permisos al archivo index.html, para ello usamos el siguiente comando.

```
[root@web1 html]# chmod 755 /var/www/html/index.html
```

Después de lo anterior solo queda probar, para ello abrimos el navegador y en el buscador escribimos la ip del servidor web 1 y deberá aparecer la página descrita en el archivo index.html.



Para configurar el servidor web 2 seguir los mismos pasos que para el servidor web 1, usando el archivo index.html para el servidor web 2.

Una vez configurados los servidores http, vamos a configurar el balanceador, para ello en la máquina balanceador nos logueamos como root usando el comando “sudo -i”

```
[vagrant@balanceador ~]$ sudo -i|
```

Ya como root vamos a verificar que el servicio httpd esté instalado y corriendo; para ello usamos el comando “service httpd status”.

```
[root@balanceador ~]# service httpd status|
```

Debería aparecer el servicio corriendo de la siguiente manera.

```
[root@balanceador ~]# service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2021-11-09 20:11:41 UTC; 6h ago
     Docs: man:httpd(8)
           man:apachectl(8)
  Main PID: 849 (httpd)
    Status: "Total requests: 0; Current requests/sec: 0; Current traffic:  0 B/sec"
    CGroup: /system.slice/httpd.service
            └─849 /usr/sbin/httpd -DFOREGROUND
              └─878 /usr/sbin/httpd -DFOREGROUND
                └─879 /usr/sbin/httpd -DFOREGROUND
                  └─880 /usr/sbin/httpd -DFOREGROUND
                    └─881 /usr/sbin/httpd -DFOREGROUND
                      └─882 /usr/sbin/httpd -DFOREGROUND

Nov 09 20:11:41 balanceador systemd[1]: Starting The Apache HTTP Server...
Nov 09 20:11:41 balanceador httpd[849]: AH00558: httpd: Could not reliably determine the server's ful...sage
Nov 09 20:11:41 balanceador systemd[1]: Started The Apache HTTP Server.
Hint: Some lines were ellipsized, use -l to show in full.
[root@balanceador ~]# |
```

Si el servicio no está activo lo encendemos con el comando “service httpd start”.

Por defecto los módulos necesarios vienen con la instalación de httpd, para verificarlos usamos el siguiente comando.

```
[root@balanceador ~]# httpd -M | grep proxy|
```

Aparecen los módulos de proxy necesarios para configurar la máquina como balanceador.

```
[root@balanceador ~]# httpd -M | grep proxy
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set
the 'ServerName' directive globally to suppress this message
proxy_module (shared)
proxy_ajp_module (shared)
proxy_balancer_module (shared)
proxy_connect_module (shared)
proxy_express_module (shared)
proxy_fcgi_module (shared)
proxy_fdpass_module (shared)
proxy_ftp_module (shared)
proxy_http_module (shared)
proxy_scgi_module (shared)
proxy_wstunnel_module (shared)
[root@balanceador ~]#
```

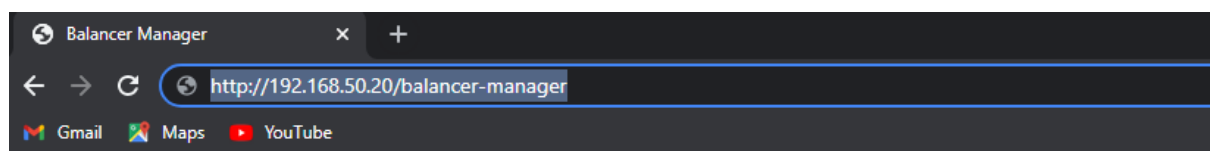
Ahora creamos el archivo de configuración “vhost_default_00.conf” en la dirección “/etc/httpd/conf.d” con el siguiente comando.

```
[root@balanceador ~]# vi /etc/httpd/conf.d/vhost_default_00.conf
```

Una vez abierto el archivo, presionamos la tecla “A” para activar la edición y lo modificamos como está en el repositorio.

Una vez modificado el archivo presionamos la tecla “ESC” y posteriormente “:wq” para salir guardando cambios.

Para probar que el balanceador está funcionando correctamente, en el navegador ponemos el siguiente enlace. <http://192.168.50.20/balancer-manager> y cargará la página donde se monitorea el estado del balanceador.



Load Balancer Manager for 192.168.50.20

Server Version: Apache/2.4.6 (CentOS)
Server Built: Oct 19 2021 13:53:40

LoadBalancer Status for [balancer://cluster1](#) [p5bfbd7e_cluster1]

MaxMembers	StickySession	DisableFailover	Timeout	FailoverAttempts	Method	Path	Active
2 [2 Used]	(None)	Off	0	1	byrequests	/	Yes

Worker URL	Route	RouteRedir	Factor	Set	Status	Elected	Busy	Load	To	From
http://192.168.50.21			1	0	Init Ok	0	0	0	0	0
http://192.168.50.22			1	0	Init Ok	0	0	0	0	0

