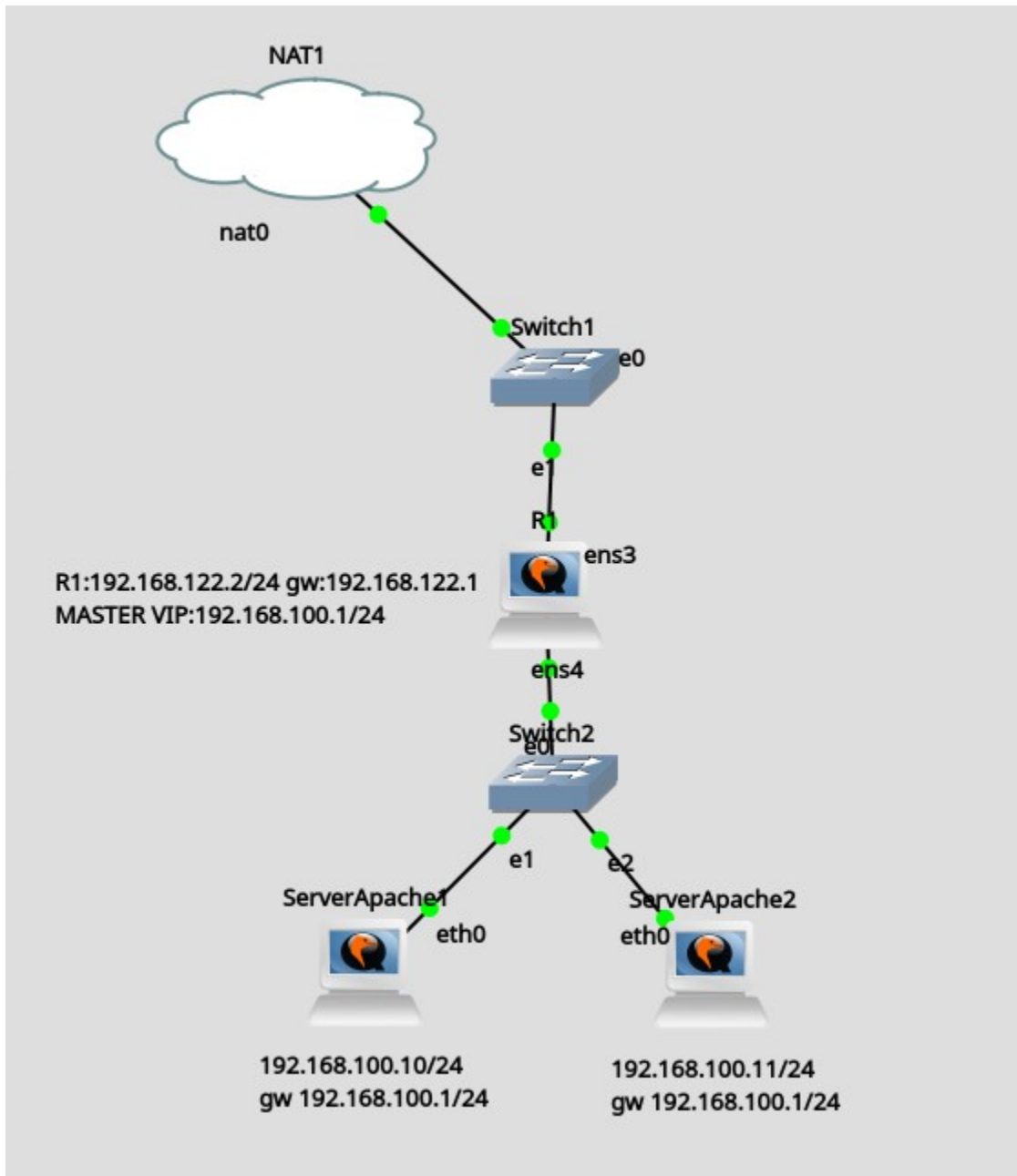


Servicio web en alta disponibilidad.

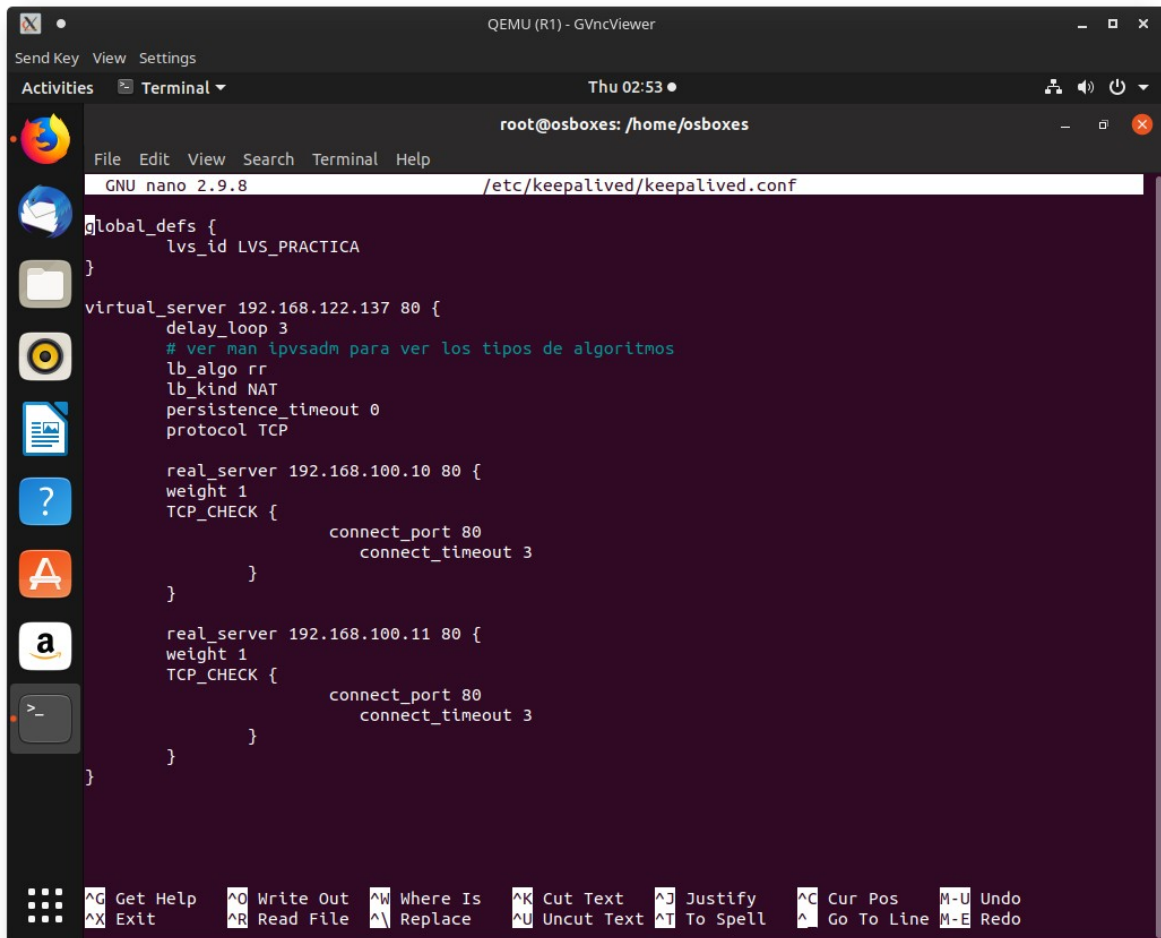
## Parte 1: Instalación y configuración de un balanceador con keepalived

Como anteriormente realice la practica T8P3 ya tenia un proyecto GNS3 con 2 maquinas con el NAT habilitado con lo cual aprovechare una de estas y metere 2 servers web mas quedando así.



## Servicio web en alta disponibilidad.

Ahora editaremos el archivo keepalived



```
QEMU (R1) - GVncViewer
Send Key View Settings
Activities Terminal Thu 02:53
root@osboxes: /home/osboxes
File Edit View Search Terminal Help
GNU nano 2.9.8 /etc/keepalived/keepalived.conf

global_defs {
    lvs_id LVS_PRACTICA
}

virtual_server 192.168.122.137 80 {
    delay_loop 3
    # ver man ipvsadm para ver los tipos de algoritmos
    lb_algo rr
    lb_kind NAT
    persistence_timeout 0
    protocol TCP

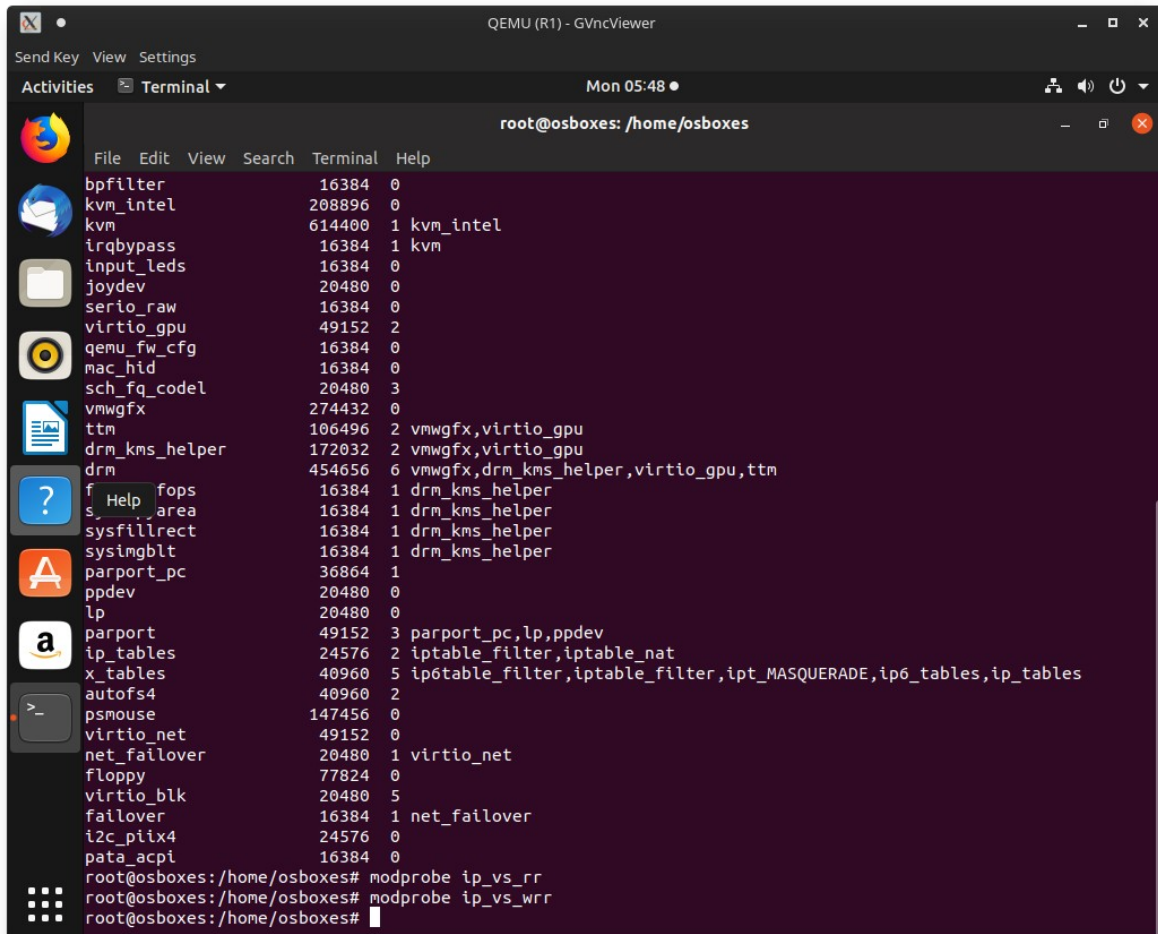
    real_server 192.168.100.10 80 {
        weight 1
        TCP_CHECK {
            connect_port 80
            connect_timeout 3
        }
    }

    real_server 192.168.100.11 80 {
        weight 1
        TCP_CHECK {
            connect_port 80
            connect_timeout 3
        }
    }
}
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo  
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^\_ Go To Line M-E Redo

## Servicio web en alta disponibilidad.

cargamos los modulos ip\_vs\_rr, ip\_vs\_wrr



```
QEMU (R1) - GVncViewer
Send Key View Settings
Activities Terminal Mon 05:48
root@osboxes: /home/osboxes

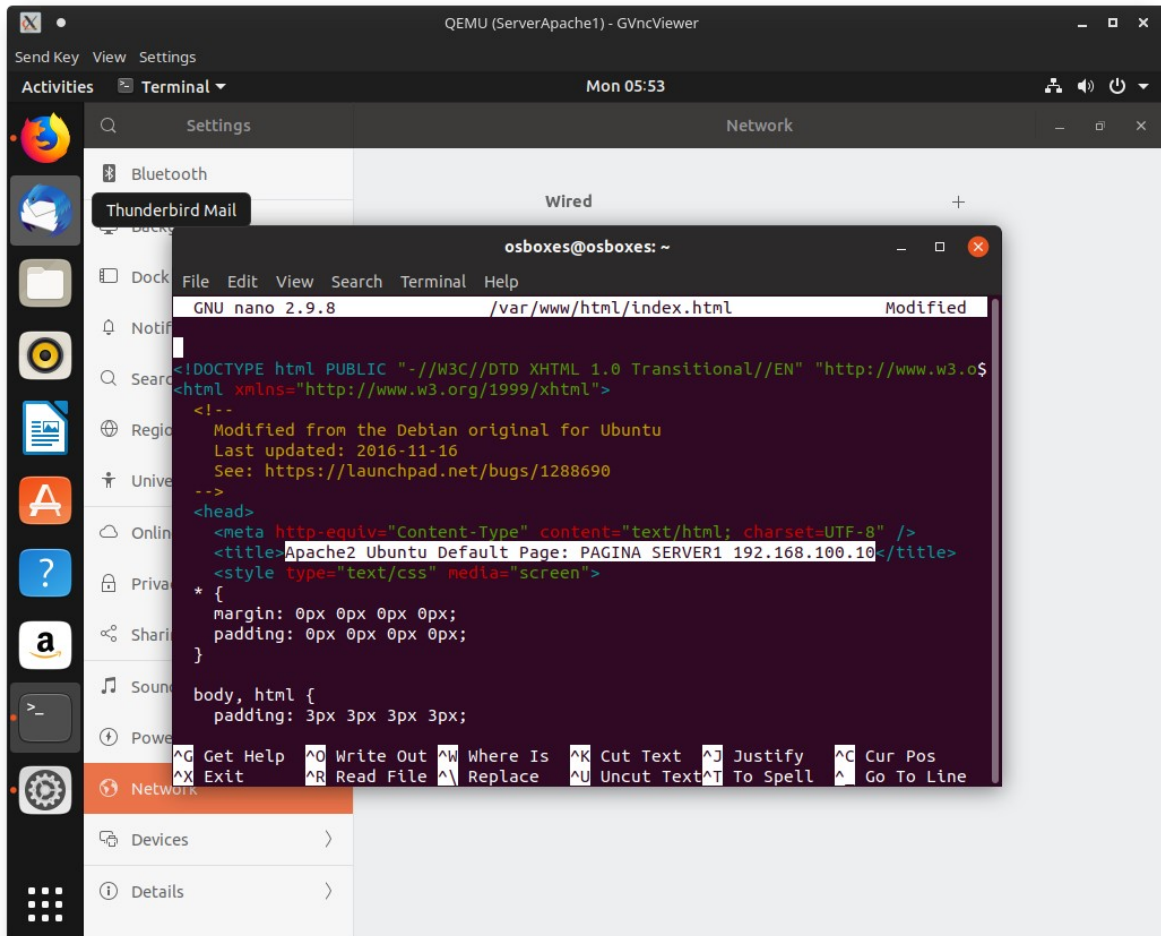
File Edit View Search Terminal Help

bpfilter 16384 0
kvm_intel 208896 0
kvm 614400 1 kvm_intel
irqbypass 16384 1 kvm
input_leds 16384 0
joydev 20480 0
serio_raw 16384 0
virtio_gpu 49152 2
qemu_fw_cfg 16384 0
mac_hid 16384 0
sch_fq_codel 20480 3
vmwgfx 274432 0
ttm 106496 2 vmwgfx,virtio_gpu
drm_kms_helper 172032 2 vmwgfx,virtio_gpu
drm 454656 6 vmwgfx,drm_kms_helper,virtio_gpu,ttm
fops 16384 1 drm_kms_helper
sysfs_area 16384 1 drm_kms_helper
sysfillrect 16384 1 drm_kms_helper
sysimgblt 16384 1 drm_kms_helper
parport_pc 36864 1
ppdev 20480 0
lp 20480 0
parport 49152 3 parport_pc,lp,ppdev
ip_tables 24576 2 iptable_filter,iptable_nat
x_tables 40960 5 ip6table_filter,iptable_filter,ipt_MASQUERADE,ip6_tables,ip_tables
autofs4 40960 2
psmouse 147456 0
virtio_net 49152 0
net_failover 20480 1 virtio_net
floppy 77824 0
virtio_blk 20480 5
failover 16384 1 net_failover
i2c_piix4 24576 0
pata_acpi 16384 0
root@osboxes:/home/osboxes# modprobe ip_vs_rr
root@osboxes:/home/osboxes# modprobe ip_vs_wrr
root@osboxes:/home/osboxes#
```

## Servicio web en alta disponibilidad.

Ahora prepararemos los servidores APACHE tanto el 1 como el 2

Instalamos el servicio apache2 y seguidamente editaremos las pagina para diferenciar los server



The screenshot shows a QEMU virtual machine window titled "QEMU (ServerApache1) - GVncViewer". Inside the VM, a terminal window is open, displaying the nano text editor editing the file `/var/www/html/index.html`. The terminal prompt is `osboxes@osboxes: ~`. The nano editor shows the following content:

```
GNU nano 2.9.8 /var/www/html/index.html Modified

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/1999/xhtml">
<html xmlns="http://www.w3.org/1999/xhtml">
<!--
  Modified from the Debian original for Ubuntu
  Last updated: 2016-11-16
  See: https://launchpad.net/bugs/1288690
-->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Apache2 Ubuntu Default Page: PAGINA SERVER1 192.168.100.10</title>
<style type="text/css" media="screen">
* {
  margin: 0px 0px 0px 0px;
  padding: 0px 0px 0px 0px;
}

body, html {
  padding: 3px 3px 3px 3px;
}
```

The nano editor's status bar at the bottom shows various keyboard shortcuts: `^G Get Help`, `^O Write Out`, `^W Where Is`, `^K Cut Text`, `^J Justify`, `^C Cur Pos`, `^X Exit`, `^R Read File`, `^_ Replace`, `^U Uncut Text`, `^T To Spell`, and `^_ Go To Line`.

## Servicio web en alta disponibilidad.



## Servicio web en alta disponibilidad.

Ahora comprobaremos desde R1 como se distribuye las peticiones entre las 2 webs

Primeramente se vera la web del server1



## Servicio web en alta disponibilidad.

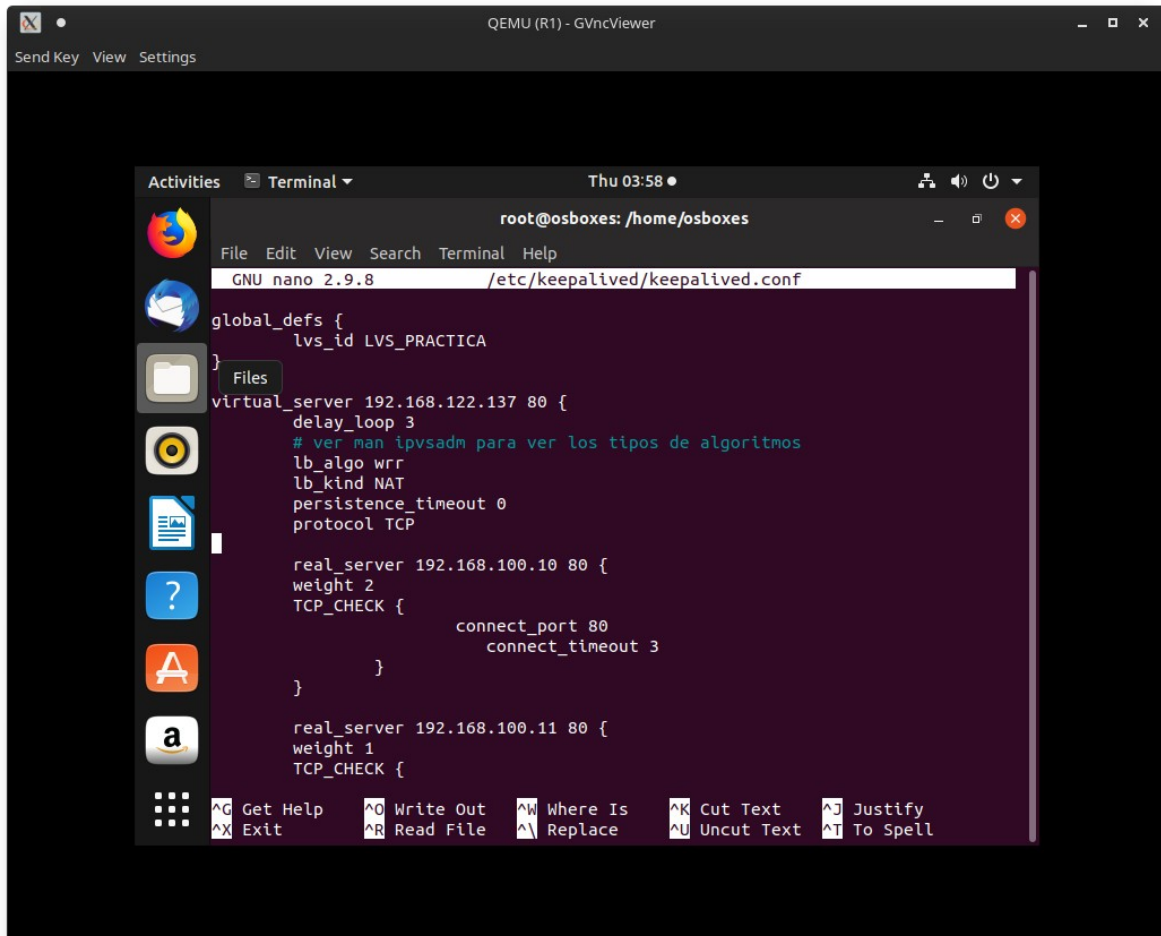
Se observa que tras actualizar unas cuantas veces la pagina esta a pasado a verse la del Server2



Con esta configuracion inicial, el peso es el mismo, algoritmo es rr (round-robin) y el delay loop 3 segundos. Con ello, cuando pasan tres segundos, la siguiente conexión cambia de servidor, y así sucesivamente.

## Servicio web en alta disponibilidad.

En la siguiente imagen, le doy peso 2 al server 1, y peso 1 al server 2. Lo que ocurre entonces, es que cada tres refrescos, dos se conecta al servidor 1 y el tercero se conecta al servidor 2. Se observa que para que se tenga en cuenta el peso, modifico el algoritmo a wrr (weighted round robin):



The image shows a terminal window titled "QEMU (R1) - GVncViewer" with a menu bar (Send Key, View, Settings). Inside the terminal, a nano editor is open at the file `/etc/keepalived/keepalived.conf`. The user is `root@osboxes` in the directory `/home/osboxes`. The configuration file content is as follows:

```
global_defs {
    lvs_id LVS_PRACTICA
}

virtual_server 192.168.122.137 80 {
    delay_loop 3
    # ver man ipvsadm para ver los tipos de algoritmos
    lb_algo wrr
    lb_kind NAT
    persistence_timeout 0
    protocol TCP

    real_server 192.168.100.10 80 {
        weight 2
        TCP_CHECK {
            connect_port 80
            connect_timeout 3
        }
    }

    real_server 192.168.100.11 80 {
        weight 1
        TCP_CHECK {
```

The terminal window also shows a sidebar with application icons (Firefox, Files, etc.) and a bottom status bar with keyboard shortcuts like `^G Get Help`, `^X Exit`, `^O Write Out`, `^R Read File`, `^W Where Is`, `^_ Replace`, `^K Cut Text`, `^U Uncut Text`, `^J Justify`, and `^T To Spell`.



## Servicio web en alta disponibilidad.

Voy haciendo refrescos cuando pasan tres segundos, y las dos primeras conexiones se hacen al server 1,



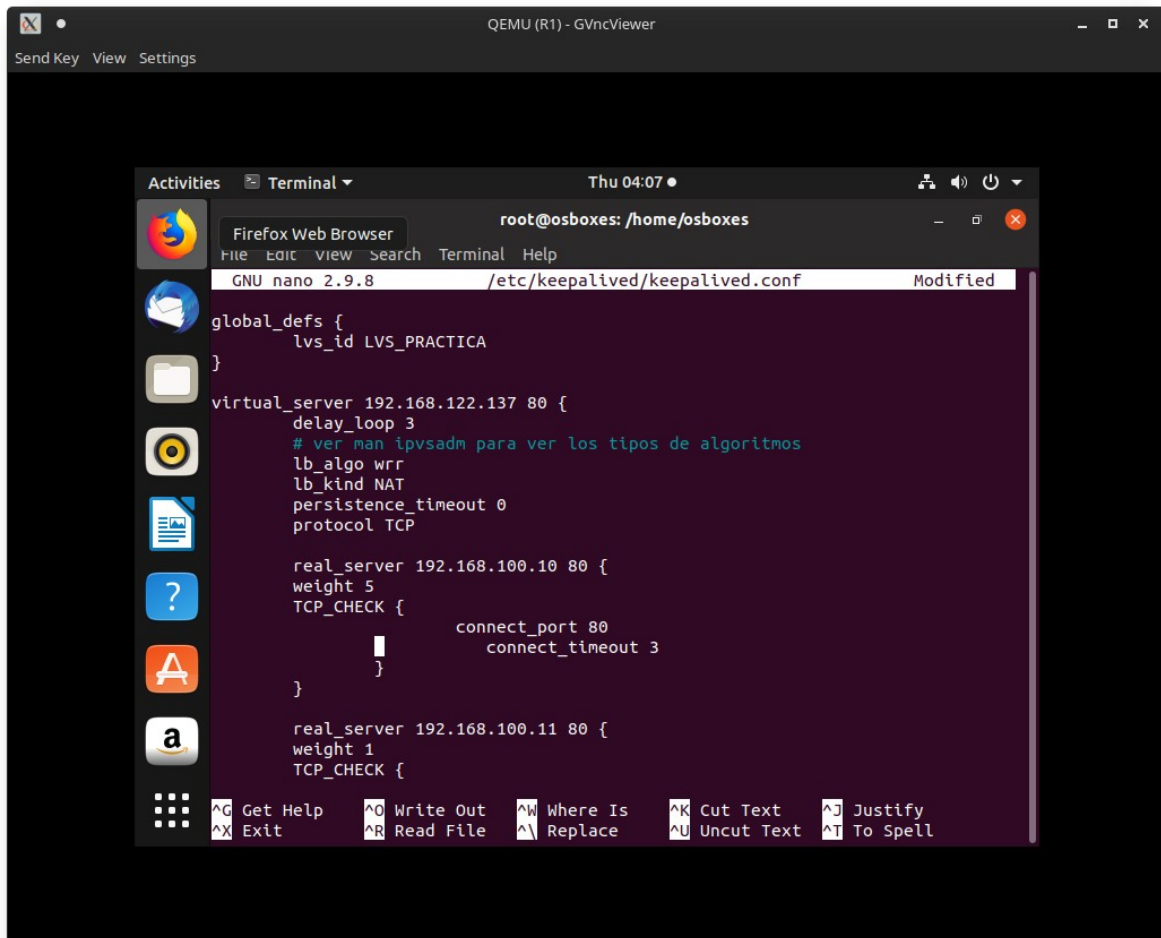
Servicio web en alta disponibilidad.

y la tercera se realiza al server 2



## Servicio web en alta disponibilidad.

Si configuro el peso a 5 en el servidor 1, 5 veces se conecta al servidor 1 y la sexta vez al servidor 2, y así sucesivamente



```
QEMU (R1) - GVncViewer
Send Key View Settings

Activities Terminal Thu 04:07
root@osboxes: /home/osboxes
GNU nano 2.9.8 /etc/keepalived/keepalived.conf Modified

global_defs {
    lvs_id LVS_PRACTICA
}

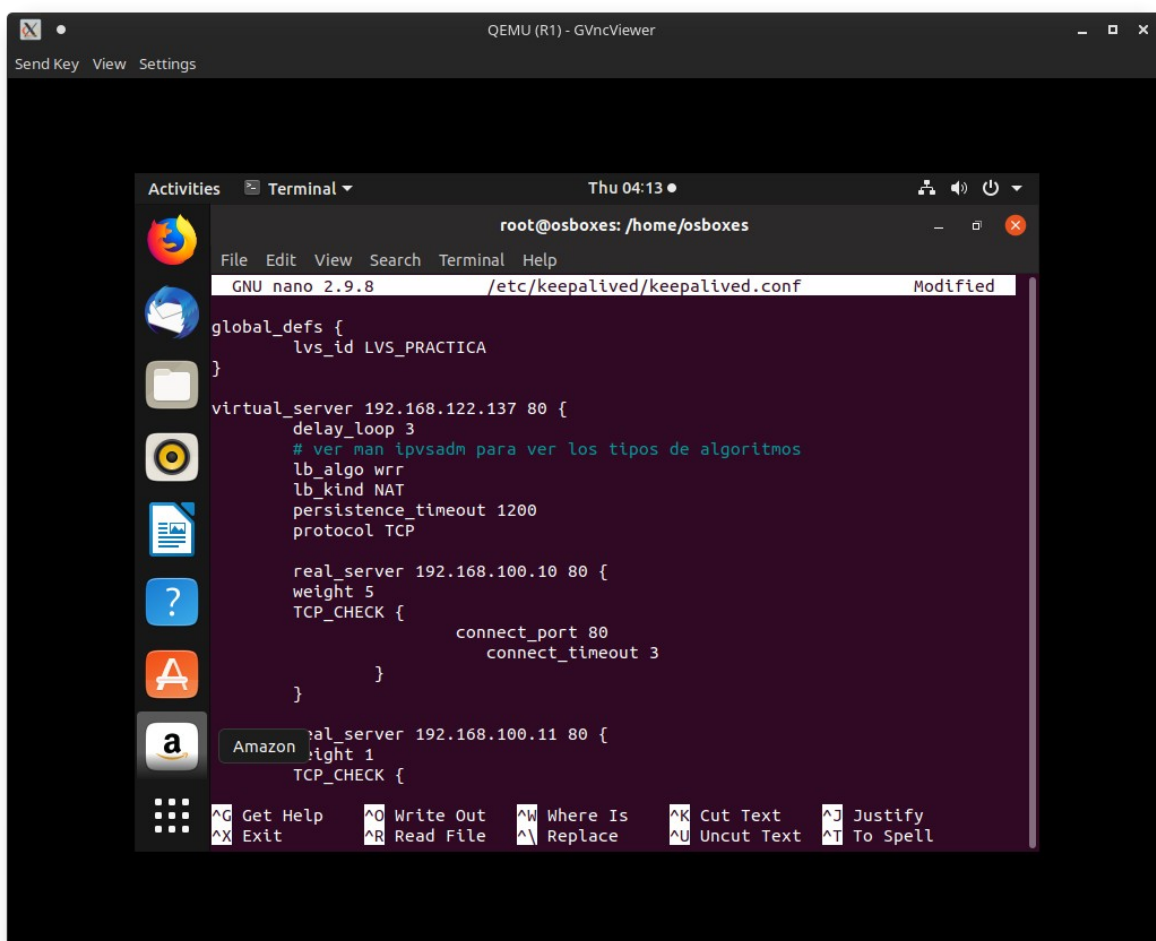
virtual_server 192.168.122.137 80 {
    delay_loop 3
    # ver man ipvsadm para ver los tipos de algoritmos
    lb_algo wrr
    lb_kind NAT
    persistence_timeout 0
    protocol TCP

    real_server 192.168.100.10 80 {
        weight 5
        TCP_CHECK {
            connect_port 80
            connect_timeout 3
        }
    }

    real_server 192.168.100.11 80 {
        weight 1
        TCP_CHECK {
```

## Servicio web en alta disponibilidad.

A continuación, tal como se solicita en la práctica, modifiqué el persistence timeout a 1200 segundos (20 minutos). De este modo, harán falta 20 minutos para que se realice el cambio de servidor. Esto es útil para mantener los id de sesiones en internet, en cualquier web donde hayan logins, tienda online etc... donde nosotros hayamos interactuado y guardado datos en el servidor, que en caso de cambio, nos desaparecerían.



```
QEMU (R1) - GVncViewer
Send Key View Settings

Activities Terminal Thu 04:13 root@osboxes: /home/osboxes
GNU nano 2.9.8 /etc/keepalived/keepalived.conf Modified
global_defs {
    lvs_id LVS_PRACTICA
}
virtual_server 192.168.122.137 80 {
    delay_loop 3
    # ver man ipvsadm para ver los tipos de algoritmos
    lb_algo wrr
    lb_kind NAT
    persistence_timeout 1200
    protocol TCP

    real_server 192.168.100.10 80 {
        weight 5
        TCP_CHECK {
            connect_port 80
            connect_timeout 3
        }
    }

    real_server 192.168.100.11 80 {
        weight 1
        TCP_CHECK {
```

Servicio web en alta disponibilidad.

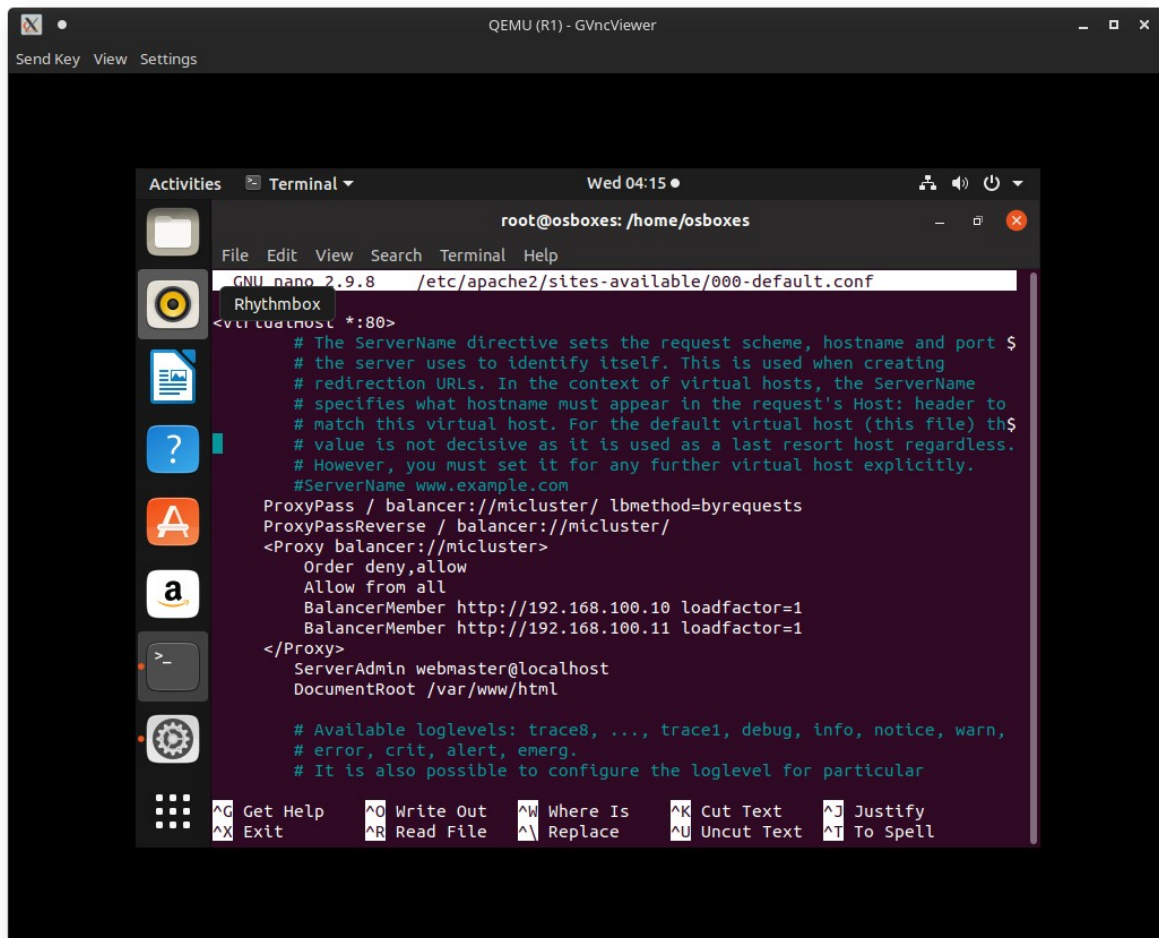
## Parte 2: Instalación y configuración de un proxy inverso con Apache

Esta parte de la practica duplicare el proyecto anterior GNS3 y así te entregare los 2.

Básicamente sera lo mismo pero instalamos apache en R1 y lo configuramos.

Aunque en la practica que indicas parece ser para configurarlo en CentOS yo te lo realizado en Ubuntu.

Básicamente es lo mismo pero ademas de editar o añadir el archivo `/etc/httpd/conf.d/proxy-inverso.conf` se edita el archivo `/etc/apache2/sites-available/000-default.conf`



```
QEMU (R1) - GVncViewer
Send Key View Settings

Activities Terminal Wed 04:15
root@osboxes: /home/osboxes

File Edit View Search Terminal Help
GNU nano 2.9.8 /etc/apache2/sites-available/000-default.conf
Rhythmbox
<vlt tuatnosL *:80>

# The ServerName directive sets the request scheme, hostname and port $
# 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) th$
# 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
ProxyPass / balancer://micluster/ lbmethod=byrequests
ProxyPassReverse / balancer://micluster/
<Proxy balancer://micluster>
    Order deny,allow
    Allow from all
    BalancerMember http://192.168.100.10 loadfactor=1
    BalancerMember http://192.168.100.11 loadfactor=1
</Proxy>
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

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell
```

conexión (habiendo un reinicio del mismo por ejemplo):

## Servicio web en alta disponibilidad.

Y así ya tenemos como la practica 1 que se balancean los 2 servidores  
Server 1



## Servicio web en alta disponibilidad.

Server 2

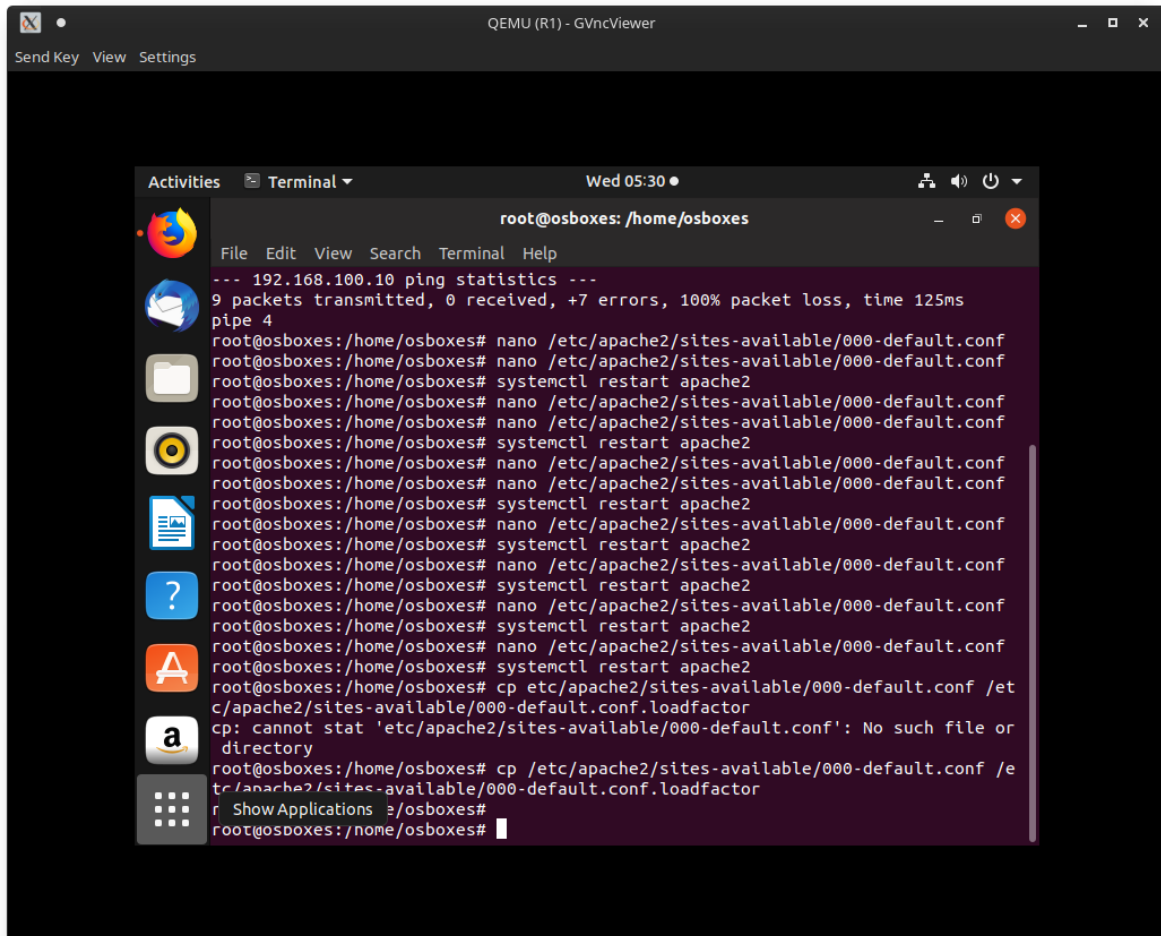


El factor loadfactor si se editan los valores que deben ser de 1 a 100 ya dependiendo valor introducido se repartiria el trabajo.

Si por ejemplo tenemos 2 servers y queremos balancear el trabajo a 1 mas potente que otro pues lo ideal seria 20 uno y 5 otro siendo este ultimo el menos potente o que menos peticiones reciba.

## Servicio web en alta disponibilidad.

Ahora vamos a editar el archivo `/etc/apache2/sites-available/000-default.conf` pero primeramente lo guardare como `/etc/apache2/sites-available/000-default.conf.loadfactor` para guardar la configuracion por si la quiero volver a cargar



```
QEMU (R1) - GVncViewer
Send Key View Settings

Activities Terminal Wed 05:30
root@osboxes: /home/osboxes
File Edit View Search Terminal Help
--- 192.168.100.10 ping statistics ---
9 packets transmitted, 0 received, +7 errors, 100% packet loss, time 125ms
pipe 4
root@osboxes: /home/osboxes# nano /etc/apache2/sites-available/000-default.conf
root@osboxes: /home/osboxes# nano /etc/apache2/sites-available/000-default.conf
root@osboxes: /home/osboxes# systemctl restart apache2
root@osboxes: /home/osboxes# nano /etc/apache2/sites-available/000-default.conf
root@osboxes: /home/osboxes# systemctl restart apache2
root@osboxes: /home/osboxes# nano /etc/apache2/sites-available/000-default.conf
root@osboxes: /home/osboxes# systemctl restart apache2
root@osboxes: /home/osboxes# nano /etc/apache2/sites-available/000-default.conf
root@osboxes: /home/osboxes# systemctl restart apache2
root@osboxes: /home/osboxes# nano /etc/apache2/sites-available/000-default.conf
root@osboxes: /home/osboxes# systemctl restart apache2
root@osboxes: /home/osboxes# nano /etc/apache2/sites-available/000-default.conf
root@osboxes: /home/osboxes# cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/000-default.conf.loadfactor
cp: cannot stat 'etc/apache2/sites-available/000-default.conf': No such file or directory
root@osboxes: /home/osboxes# cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/000-default.conf.loadfactor
r Show Applications @/osboxes#
root@osboxes: /home/osboxes#
```

Esta claro que para que funcione debera llamarse el archivo `/etc/apache2/sites-available/000-default.conf` pero a este guardare una copia llamada `/etc/apache2/sites-available/000-default.conf.persistencia`. Y asi a la hora de entregar el proyecto GNS3 tendras las 2 configuraciones.



## Servicio web en alta disponibilidad.

etc/apache2/sites-available/000-default.conf.persistencia sera la configuracion de esta parte del la practica

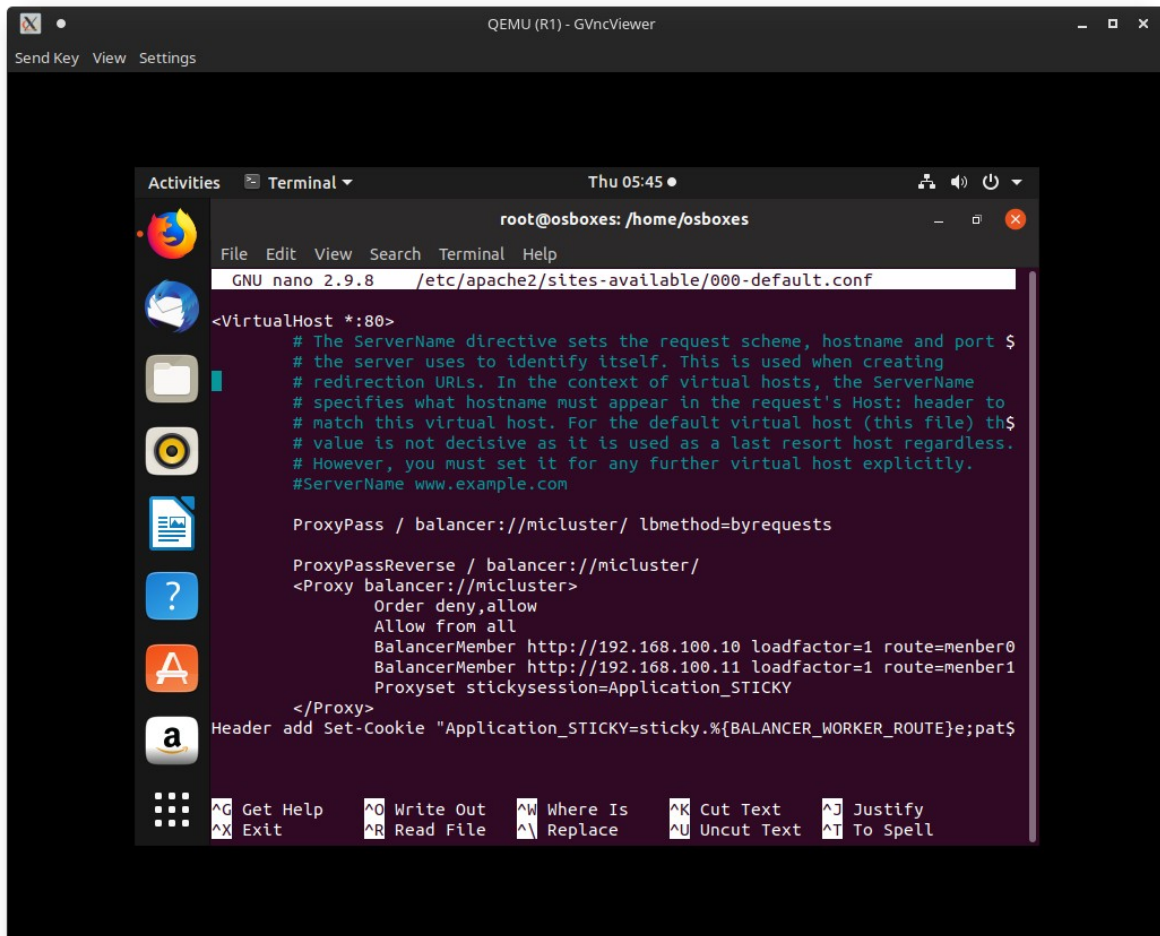
etc/apache2/sites-available/000-default.conf.loadfactor la configuracion anterior.

Primero habilitamos el modulo headers con “a2enmod headers”ya que anteriormente me arrojaba el siguiente error

```
root@osboxes:/home/osboxes# tail -f /var/log/syslog
Feb  5 06:18:06 osboxes apachectl[4881]: AH00526: Syntax error on line 21 of /etc/apache2/sites-enabled/000-default.conf:
Feb  5 06:18:06 osboxes apachectl[4881]: Invalid command 'Header', perhaps misspelled or defined by a module not included in the server configuration
Feb  5 06:18:06 osboxes apachectl[4881]: Action 'start' failed.
```

## Servicio web en alta disponibilidad.

Seguidamente editamos el archivo de configuracion quedando asi



```
QEMU (R1) - GVncViewer
Send Key View Settings

Activities Terminal Thu 05:45
root@osboxes: /home/osboxes
GNU nano 2.9.8 /etc/apache2/sites-available/000-default.conf

<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port $
    # 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) th$
    # 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

    ProxyPass / balancer://mcluster/ lbmethod=byrequests

    ProxyPassReverse / balancer://mcluster/
    <Proxy balancer://mcluster>
        Order deny,allow
        Allow from all
        BalancerMember http://192.168.100.10 loadfactor=1 route=member0
        BalancerMember http://192.168.100.11 loadfactor=1 route=member1
        ProxySet stickysession=Application_STICKY
    </Proxy>
    Header add Set-Cookie "Application_STICKY=sticky.%(BALANCER_WORKER_ROUTE)e;pat$

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell
```

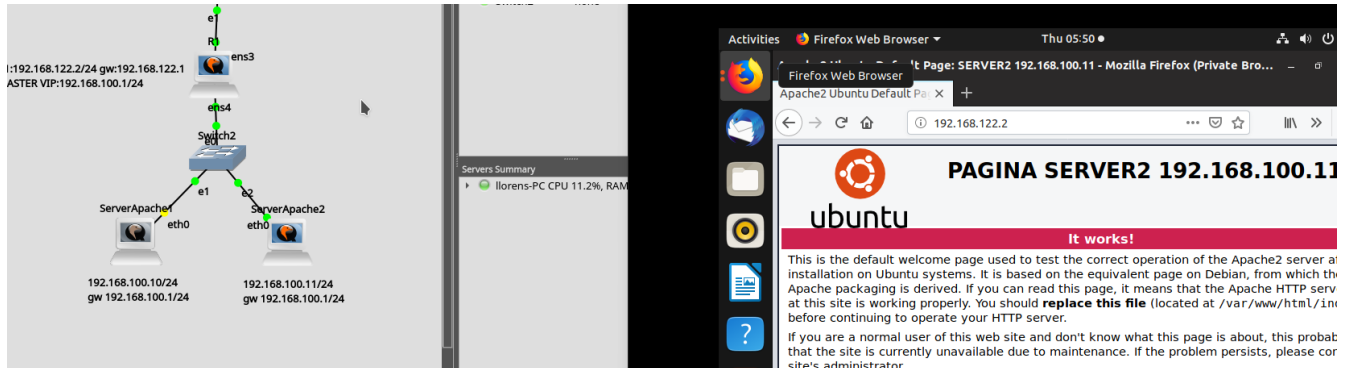
## Servicio web en alta disponibilidad.

Con esta ultima configuracion cada conexión de un cliente y mismo navegador ser realizara en el mismo servidor, aunque habra mas pestañas o aunque actualice las paginas ininidad de veces. Esta funcion es util para si el cliente inicia sesion en unos de los servers. El otro no le seria util para la misma sesion que tiene el cliente.



## Servicio web en alta disponibilidad.

Pero ahora supongamos que por motivos externos el servidor 1 fallase en mi caso vamos a congelar el server 1



Entonces la pagina aun seguiria operativa dado a que el server 2 aun seguiria operativo, supongo que deberia iniciar sesion nuevamente....