



Practica 4

Asegurar la granja web

Carlos Garcia Segura
SWAP

Índice

1. [Instalar certificado SSL auto firmado](#)
2. [Configuración del cortafuegos](#)

1. Instalar certificado SSL auto firmado

Lo primero que debemos hacer es activar el module ssl en apache2, crear un fichero para los certificados y generar los certificados. Para ello usamos los siguientes comandos

```
sudo a2enmod ssl
```

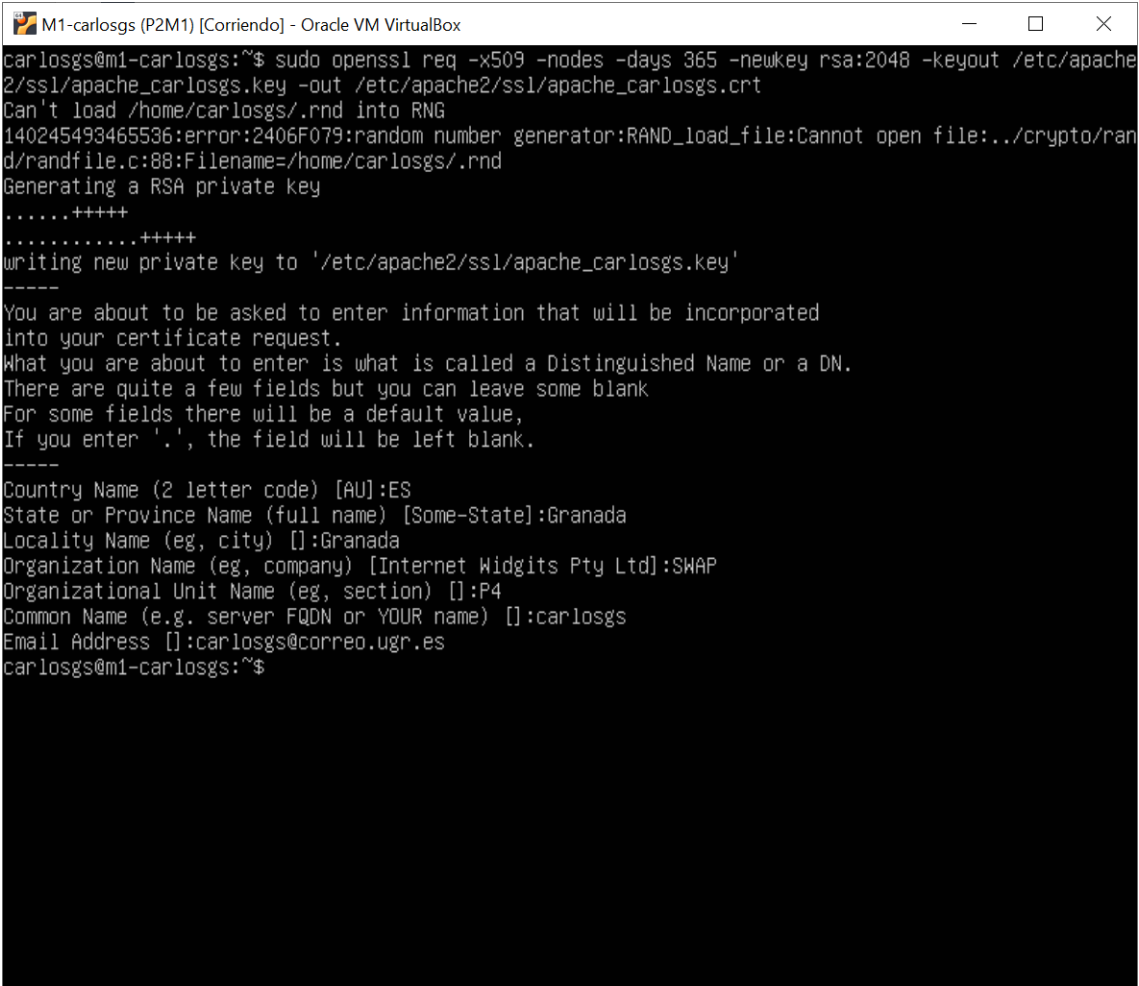
```
sudo service apache2 restart
```

```
sudo mkdir /etc/apache2/ssl
```

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
```

```
/etc/apache2/ssl/apache_usuarioUGR.key -out
```

```
/etc/apache2/ssl/apache_usuarioUGR.crt
```



```
M1-carlosgs (P2M1) [Corriendo] - Oracle VM VirtualBox
carlosgs@m1-carlosgs:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache_carlosgs.key -out /etc/apache2/ssl/apache_carlosgs.crt
Can't load /home/carlosgs/.rnd into RNG
140245493465536:error:2406F079:random number generator:RAND_load_file:Cannot open file:../crypto/rand/randfile.c:88:Filename=/home/carlosgs/.rnd
Generating a RSA private key
.....+++++
.....+++++
writing new private key to '/etc/apache2/ssl/apache_carlosgs.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Granada
Locality Name (eg, city) []:Granada
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SWAP
Organizational Unit Name (eg, section) []:P4
Common Name (e.g. server FQDN or YOUR name) []:carlosgs
Email Address []:carlosgs@correo.ugr.es
carlosgs@m1-carlosgs:~$
```

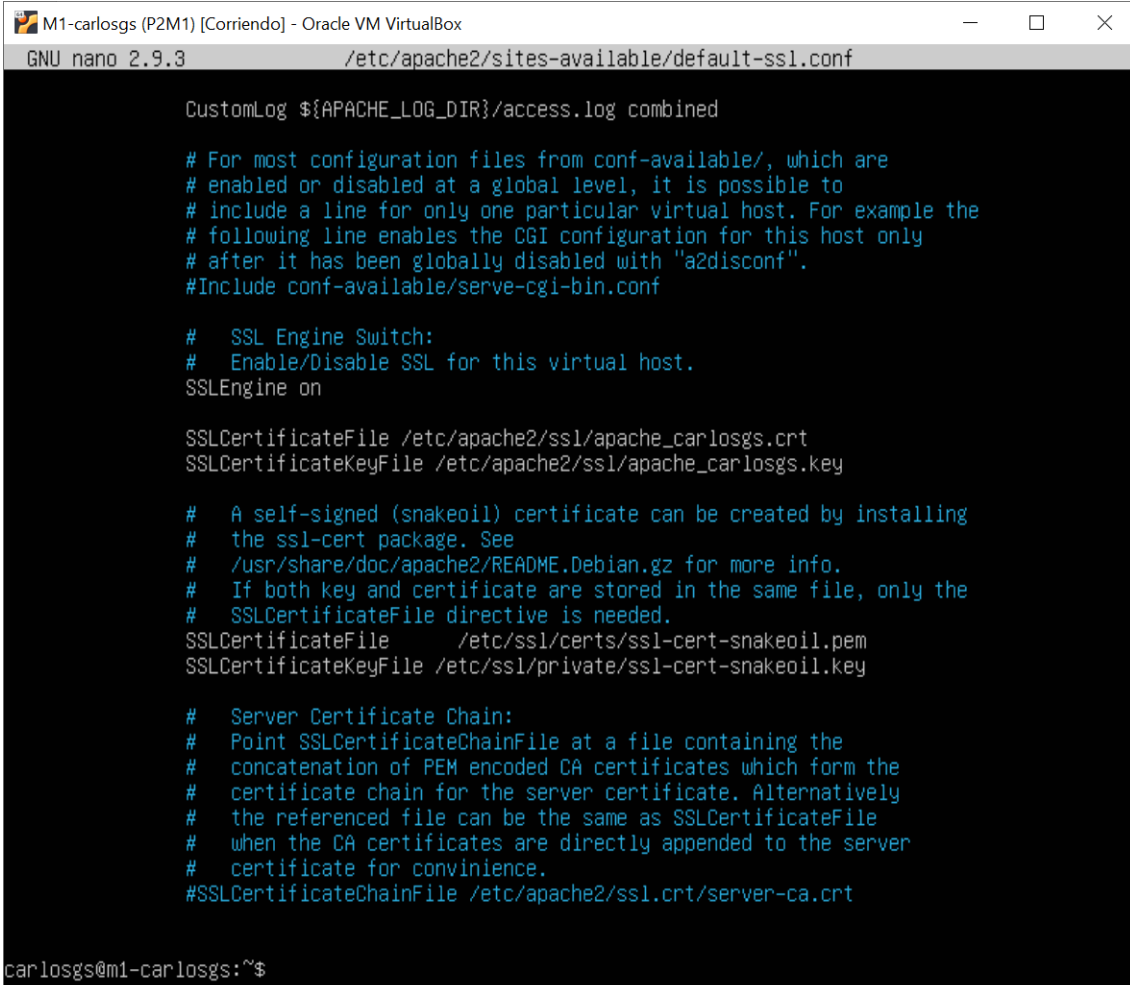
Lo siguiente que vamos a hacer es agregar las rutas de los certificados en el archivo de configuración

```
nano /etc/apache2/sites-available/default-ssl.conf
```

Y agregamos la ruta de los certificados debajo del parámetro **SSLEngine on**:

```
SSLCertificateFile /etc/apache2/ssl/apache_usuarioUGR.crt
```

```
SSLCertificateKeyFile /etc/apache2/ssl/apache_usuarioUGR.key
```



The screenshot shows a terminal window titled "M1-carlosgs (P2M1) [Corriendo] - Oracle VM VirtualBox". The terminal is running the GNU nano 2.9.3 editor, editing the file /etc/apache2/sites-available/default-ssl.conf. The configuration file content is as follows:

```
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf

# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on

SSLCertificateFile /etc/apache2/ssl/apache_carlosgs.crt
SSLCertificateKeyFile /etc/apache2/ssl/apache_carlosgs.key

# A self-signed (snakeoil) certificate can be created by installing
# the ssl-cert package. See
# /usr/share/doc/apache2/README.Debian.gz for more info.
# If both key and certificate are stored in the same file, only the
# SSLCertificateFile directive is needed.
SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key

# Server Certificate Chain:
# Point SSLCertificateChainFile at a file containing the
# concatenation of PEM encoded CA certificates which form the
# certificate chain for the server certificate. Alternatively
# the referenced file can be the same as SSLCertificateFile
# when the CA certificates are directly appended to the server
# certificate for convenience.
#SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt
```

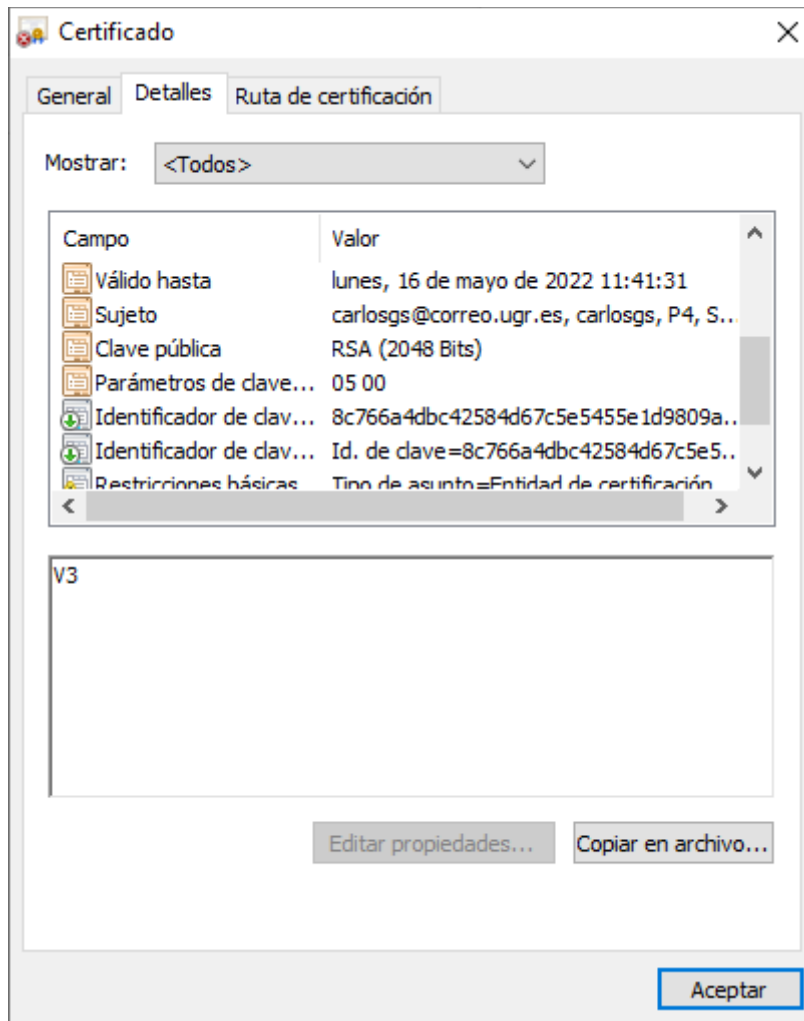
The terminal prompt at the bottom is carlosgs@m1-carlosgs:~\$.

Activamos el sitio default-ssl y reiniciamos apache:

```
a2ensite default-ssl
```

```
service apache2 reload
```

Nos metemos en <https://192.168.56.101/> y comprobamos el certificado



Lo siguiente será copiar los certificados en la M2

```
sudo scp apache.crt usuario@ipm2:/home/usuario/apache_usuarioUGR.crt
```

```
sudo scp apache.key usuario@ipm2:/home/usuario/apache_usuarioUGR.key
```

Una vez hecho esto repetimos lo que hicimos en la maquina anterior para generar los certificados

```
sudo mkdir /etc/apache2/ssl
```

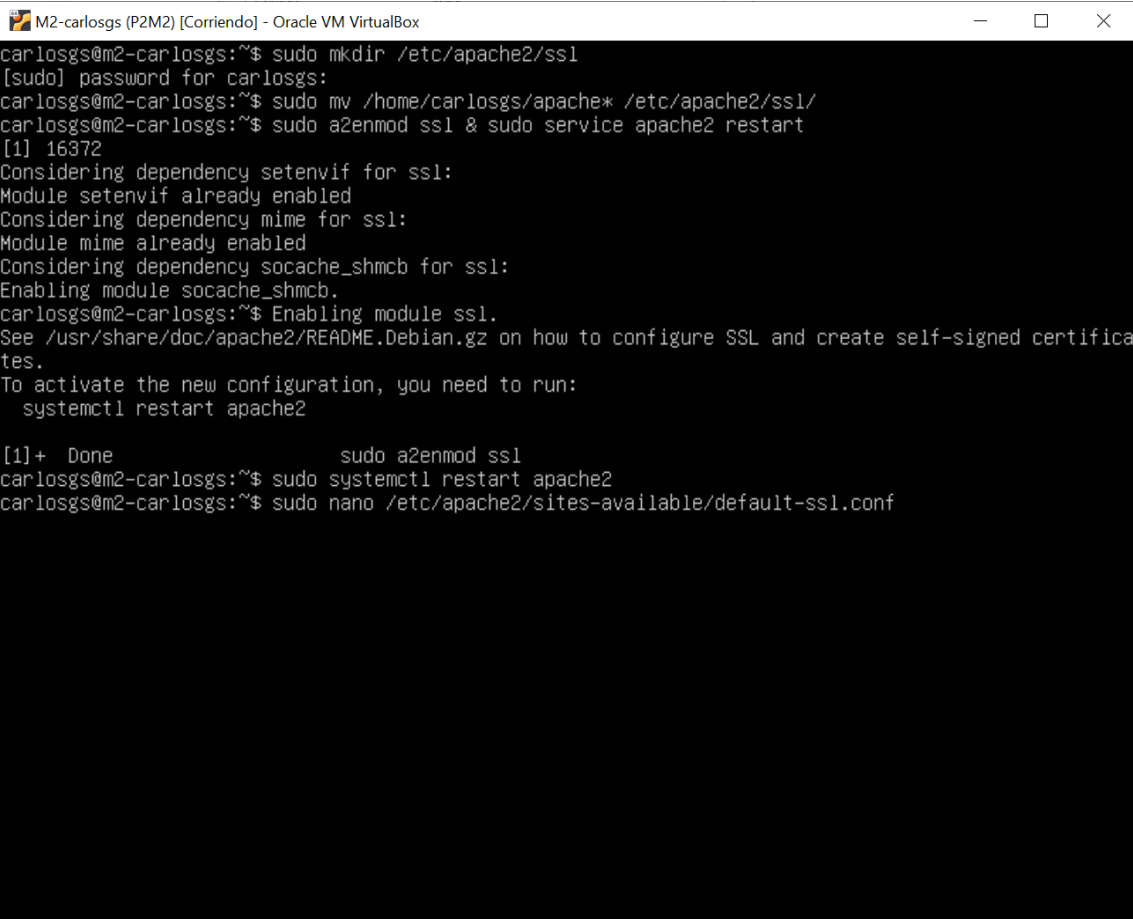
```
sudo mv /home/usuario/apache* /etc/apache2/ssl
```

```
sudo a2enmod ssl & sudo service apache2 restart
```

```
sudo nano /etc/apache2/sites-available/default-ssl.conf "SSLCertificateFile  
    /etc/apache2/ssl/apache_usuarioUGR.crt SSLCertificateKeyFile  
    /etc/apache2/ssl/apache_usuarioUGR.key"
```

```
sudo a2ensite default-ssl
```

```
sudo service apache2 reload
```



```
M2-carlosgs (P2M2) [Corriendo] - Oracle VM VirtualBox
carlosgs@m2-carlosgs:~$ sudo mkdir /etc/apache2/ssl
[sudo] password for carlosgs:
carlosgs@m2-carlosgs:~$ sudo mv /home/carlosgs/apache* /etc/apache2/ssl/
carlosgs@m2-carlosgs:~$ sudo a2enmod ssl & sudo service apache2 restart
[1] 16372
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
carlosgs@m2-carlosgs:~$ Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
    systemctl restart apache2

[1]+  Done                  sudo a2enmod ssl
carlosgs@m2-carlosgs:~$ sudo systemctl restart apache2
carlosgs@m2-carlosgs:~$ sudo nano /etc/apache2/sites-available/default-ssl.conf
```

```
M2-carlosgs (P2M2) [Corriendo] - Oracle VM VirtualBox
GNU nano 2.9.3 /etc/apache2/sites-available/default-ssl.conf

# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf

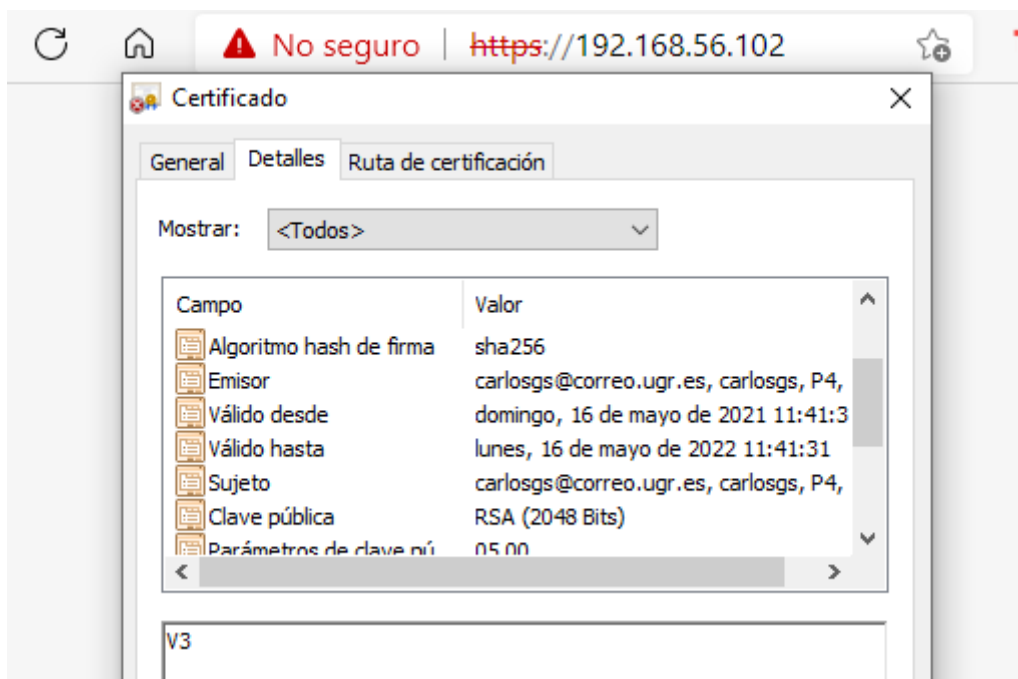
# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on

SSLCertificateFile /etc/apache2/ssl/apache_carlosgs.crt
SSLCertificateKeyFile /etc/apache2/ssl/apache_carlosgs.key
# A self-signed (snakeoil) certificate can be created by installing
# the ssl-cert package. See
# /usr/share/doc/apache2/README.Debian.gz for more info.
# If both key and certificate are stored in the same file, only the
# SSLCertificateFile directive is needed.
#
# SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
#
# SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key

# Server Certificate Chain:
# Point SSLCertificateChainFile at a file containing the
# concatenation of PEM encoded CA certificates which form the
# certificate chain for the server certificate. Alternatively
# the referenced file can be the same as SSLCertificateFile
# when the CA certificates are directly appended to the server
# certificate for convinience.
#SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

# Certificate Authority (CA):
# Set the CA certificate verification path where to find CA
# certificates for client authentication or alternatively one

carlosgs@m2-carlosgs:~$ _
```



Por último, tenemos que configurar el balanceador. Para ellos copiamos los certificados a esta máquina y lo movemos,

```
sudo scp apache.crt usuario@ipm2:/home/usuario/apache_usuarioUGR.crt
```

```
sudo scp apache.key usuario@ipm2:/home/usuario/apache_usuarioUGR.key
```

```
sudo mkdir /home/carlosgs/ssl
```

```
sudo mv /home/carlosgs/apache* /home/carlosgs/ssl
```

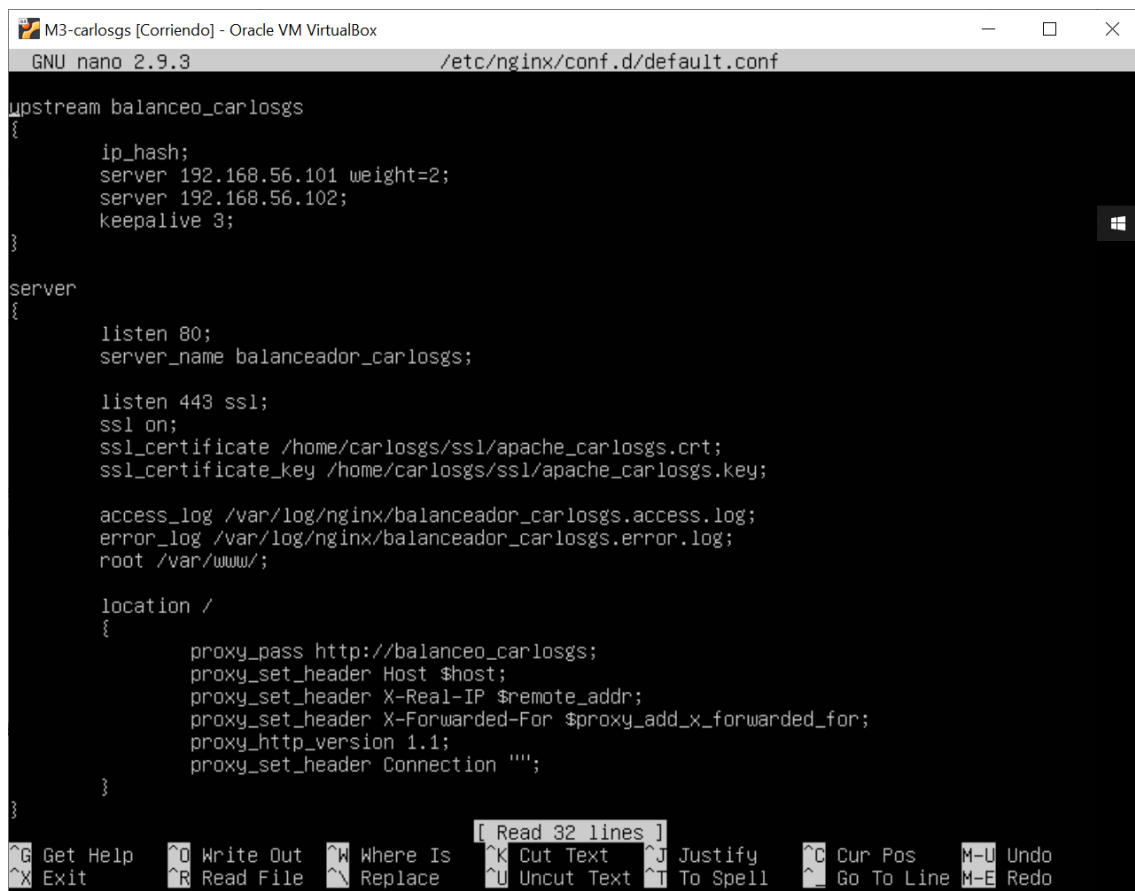
Una vez hecho esto editamos el archivo de configuración de nginx para que acepte peticiones https.

```
listen 443 ssl;
```

```
ssl on;
```

```
ssl_certificate /home/usuario/ssl/apache_usuarioUGR.crt;
```

```
ssl_certificate_key /home/usuario/ssl/apache_usuarioUGR.key;
```



The screenshot shows a terminal window titled "M3-carlosgs [Corriendo] - Oracle VM VirtualBox". The terminal is running the nano text editor, editing the file "/etc/nginx/conf.d/default.conf". The configuration file content is as follows:

```
upstream balanceo_carlosgs
{
    ip_hash;
    server 192.168.56.101 weight=2;
    server 192.168.56.102;
    keepalive 3;
}

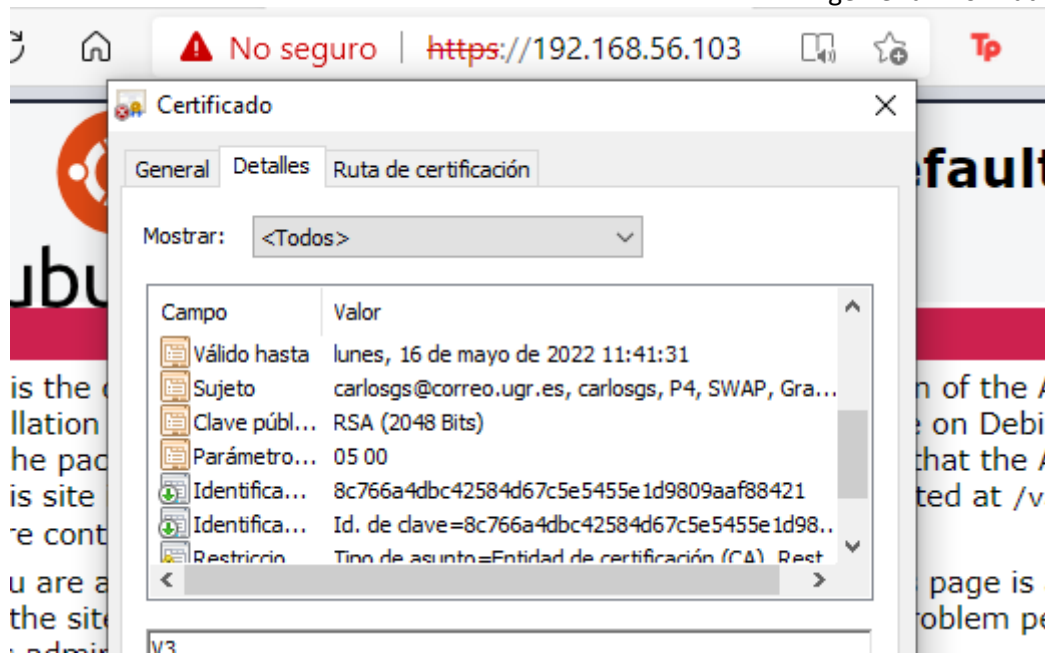
server
{
    listen 80;
    server_name balanceador_carlosgs;

    listen 443 ssl;
    ssl on;
    ssl_certificate /home/carlosgs/ssl/apache_carlosgs.crt;
    ssl_certificate_key /home/carlosgs/ssl/apache_carlosgs.key;

    access_log /var/log/nginx/balanceador_carlosgs.access.log;
    error_log /var/log/nginx/balanceador_carlosgs.error.log;
    root /var/www/;

    location /
    {
        proxy_pass http://balanceo_carlosgs;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
    }
}
```

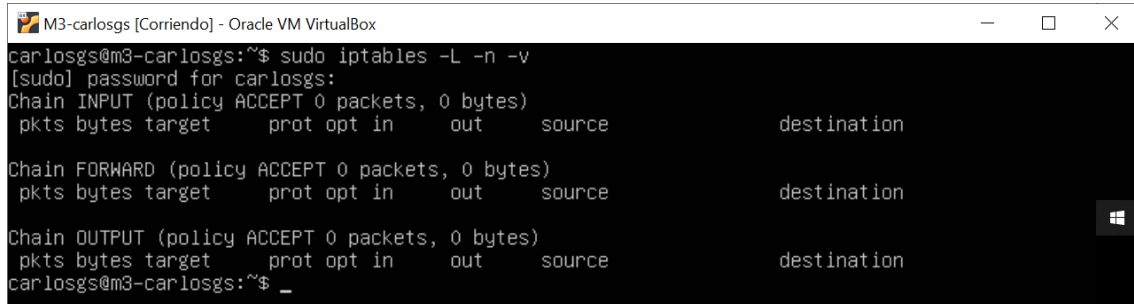
The terminal window also shows the nano editor's status bar at the bottom, indicating "Read 32 lines" and providing various keyboard shortcuts for navigation and editing.



2. Configuración del cortafuegos

Lo primero es comprobar el estado inicial del cortafuegos con

`Iptables -L -n -v`



```
M3-carlosgs [Corriendo] - Oracle VM VirtualBox
carlosgs@m3-carlosgs:~$ sudo iptables -L -n -v
[sudo] password for carlosgs:
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source                   destination
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source                   destination
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source                   destination
carlosgs@m3-carlosgs:~$ _
```

- Ahora para denegar todo el tráfico entrante que no sea HTTP y HTTPS en todas las maquinas deberemos aplicar los siguientes comandos en todas ellas.

- Lo primero será eliminar las reglas que existan (por defecto no debería haber ninguna) para comenzar desde 0.

`iptables -F`

`iptables -X`

`iptables -Z`

- Lo siguiente será denegar todo el tráfico entrante.

`iptables -P INPUT DROP`

`iptables -P OUTPUT ACCEPT`

`iptables -P FORWARD DROP`

`iptables -A INPUT -m state --state NEW,ESTABLISHED -j ACCEPT`

- Por último, añadimos reglas para aceptar los puertos 80 y 443 (HTTP y HTTPS)

`Iptables -A INPUT -m state --state NEW -p tcp --dport 80 -j ACCEPT`

`Iptables -A INPUT -m state --state NEW -p tcp --dport 443 -j ACCEPT`

```
M3-carlosgs [Corriendo] - Oracle VM VirtualBox
carlosgs@m3-carlosgs:~$ sudo iptables -L -n -v
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source         destination     state NEW,E
STABLISHED
 83 24153 ACCEPT     all  --  *      *        0.0.0.0/0      0.0.0.0/0
 0 0 ACCEPT     tcp  --  *      *        0.0.0.0/0      0.0.0.0/0      state NEW t
cp dpt:80
 0 0 ACCEPT     tcp  --  *      *        0.0.0.0/0      0.0.0.0/0      state NEW t
cp dpt:443

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source         destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source         destination
carlosgs@m3-carlosgs:~$ _
```

También creamos un script que hace lo mismo que hemos hecho anteriormente.

```
M1-carlosgs (P2M1) [Corriendo] - Oracle VM VirtualBox
GNU nano 2.9.3 iptables

#!/bin/bash

# (1) se eliminan todas las reglas que hubiera
#para hacer la configuracion limpia:
iptables -F
iptables -X
iptables -Z

# (2) establecer las politicas por defecto (denegar todo el trafico entrante):
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
iptables -A INPUT -m state --state NEW,ESTABLISHED -j ACCEPT

# (3) añadimos las reglas para aceptar los puertos 80 y 443 (HTTP y HTTPS)
iptables -A INPUT -m state --state NEW -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -m state --state NEW -p tcp --dport 443 -j ACCEPT
```

- Si queremos que M3 sea la única que acepta peticiones HTTPS y HTTP y que M1 y M2 solo acepten peticiones provenientes de M3 haremos lo siguiente.

- Lo primero será eliminar las reglas que existan (por defecto no debería haber ninguna) para comenzar desde 0.

```
iptables -F
```

```
iptables -X
```

```
iptables -Z
```

- Lo siguiente será denegar todo el tráfico entrante.

```
iptables -P INPUT DROP
```

```
iptables -P OUTPUT ACCEPT
```

```
iptables -P FORWARD DROP
```

```
iptables -A INPUT -m state --state NEW,ESTABLISHED -j ACCEPT
```

- En M3 aceptamos el tráfico HTTP y HTTPS

```
Iptables -A INPUT -m state --state NEW -p tcp --dport 80 -j ACCEPT
```

```
Iptables -A INPUT -m state --state NEW -p tcp --dport 443 -j ACCEPT
```

- En M1 y M2 aceptaremos el tráfico que provenga de M3

```
Iptables -I INPUT -s 192.168.56.103 -j ACCEPT
```

```
M1-carlosgs (P2M1) [Corriendo] - Oracle VM VirtualBox
carlosgs@m1-carlosgs:~$ sudo iptables -L -n -v
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source               destination
    0     0 ACCEPT    all  --  *      *       192.168.56.103       0.0.0.0/0
  253 73623 ACCEPT    all  --  *      *       0.0.0.0/0            0.0.0.0/0          state NEW,ESTABLISHED

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source               destination
carlosgs@m1-carlosgs:~$
```

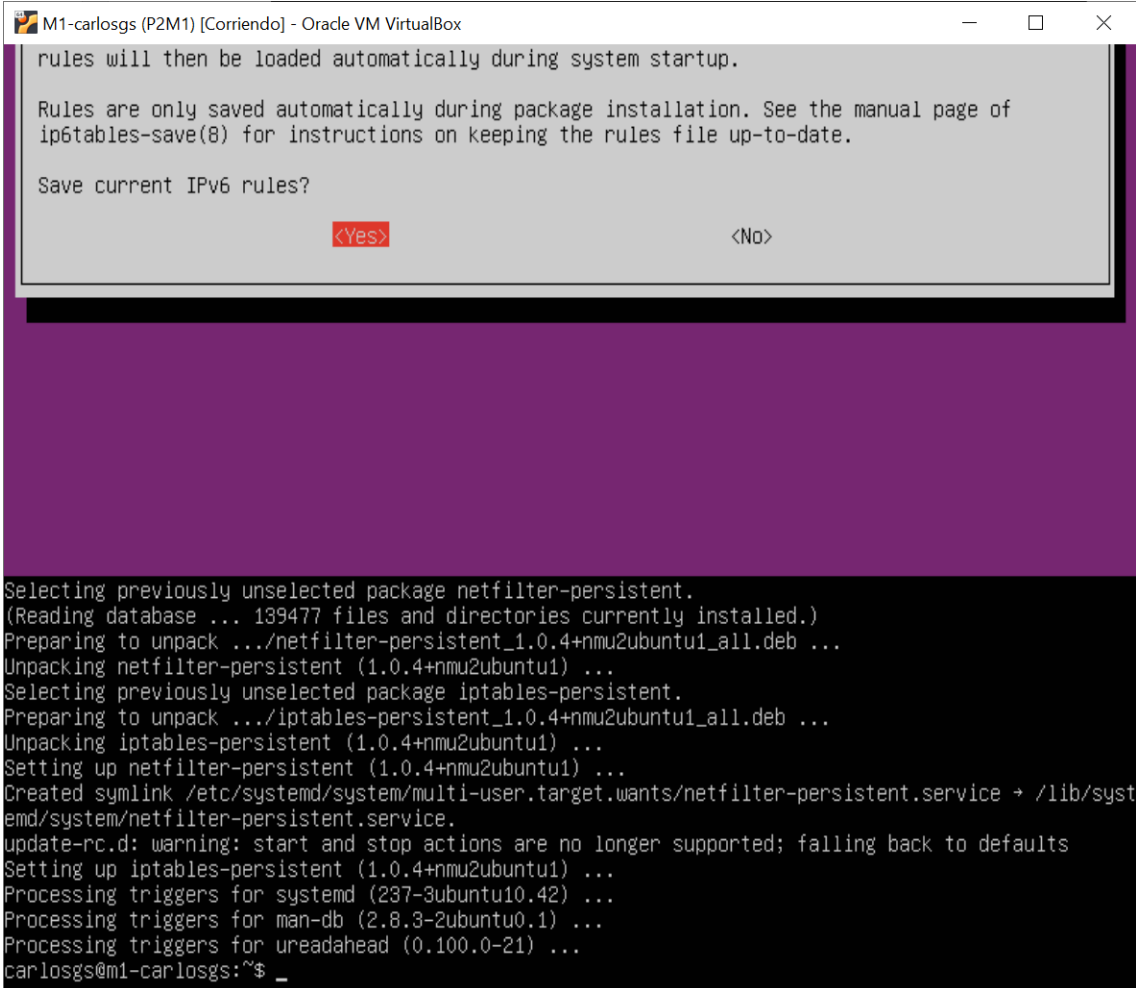
```
M2-carlosgs (P2M2) [Corriendo] - Oracle VM VirtualBox
carlosgs@m2-carlosgs:~$ sudo iptables -L -n -v
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source               destination
    0     0 ACCEPT    all  --  *      *       192.168.56.103       0.0.0.0/0
   91 26551 ACCEPT    all  --  *      *       0.0.0.0/0            0.0.0.0/0          state NEW,ESTABLISHED

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 2 packets, 397 bytes)
 pkts bytes target    prot opt in     out     source               destination
carlosgs@m2-carlosgs:~$
```

- Si queremos que la configuración del cortafuegos se ejecute al arranque del sistema en todas las máquinas, deberemos instalar en todas las máquinas el paquete `iptables-persistent` con:

```
sudo apt-get install iptables-persistent
```



```
M1-carlosgs (P2M1) [Corriendo] - Oracle VM VirtualBox
rules will then be loaded automatically during system startup.

Rules are only saved automatically during package installation. See the manual page of
iptables-save(8) for instructions on keeping the rules file up-to-date.

Save current IPv6 rules?
<Yes> <No>

Selecting previously unselected package netfilter-persistent.
(Reading database ... 139477 files and directories currently installed.)
Preparing to unpack .../netfilter-persistent_1.0.4+nmu2ubuntu1_all.deb ...
Unpacking netfilter-persistent (1.0.4+nmu2ubuntu1) ...
Selecting previously unselected package iptables-persistent.
Preparing to unpack .../iptables-persistent_1.0.4+nmu2ubuntu1_all.deb ...
Unpacking iptables-persistent (1.0.4+nmu2ubuntu1) ...
Setting up netfilter-persistent (1.0.4+nmu2ubuntu1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/netfilter-persistent.service → /lib/systemd/system/netfilter-persistent.service.
update-rc.d: warning: start and stop actions are no longer supported; falling back to defaults
Setting up iptables-persistent (1.0.4+nmu2ubuntu1) ...
Processing triggers for systemd (237-3ubuntu10.42) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
carlosgs@m1-carlosgs:~$ _
```

Con el comando

```
sudo iptables-save
```

las reglas se guardan y al iniciar el sistema seguirán estando