

Práctica 4

Asegurar la granja web

1.Tareas básicas:	2
1.1 Certificado autoafirmado SSL	2
Configurar apache con ruta de los certificados	3
1.2 Copiar certificado autoafirmado SSL a M1 y M2	4
Configurar balanceador nginx con nuevo server con los certificados SSL y parámetros correspondientes. Añadimos a los parámetros existentes:	5
2.Tareas avanzadas:	6
2.1. IPTABLES (scripts)	6
2.2. Configurar cortafuegos al arranque (hacer persistente reglas IPTABLES)	8
Vemos que funciona	9
HTTP	9
HTTPS	9
SSH	9
PING	10
2.3. Cerbot	10

1. Tareas básicas:

Crear e instalar en la máquina M1 un certificado SSL autofirmado para configurar el acceso HTTPS al servidor. Se debe comprobar que el servidor acepta tanto el tráfico HTTP como el HTTPS.

1.1. Certificado autofirmado SSL

Máquina 1 - Generar un certificado SSL autofirmado y configurar apache.

A. Activar módulo SSL de apache y crear directorio para certificados

- Activar módulo SSL

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

- Crear directorio SSL para los certificados

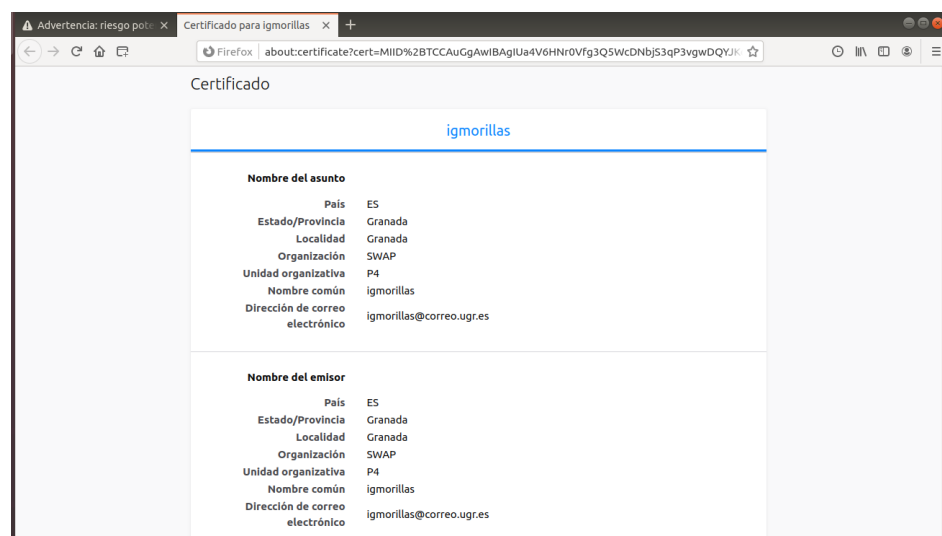
```
$ sudo mkdir /etc/apache2/ssl
```

B. Generar certificados (openssl)

```
$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048  
-keyout /etc/apache2/ssl/apache_igmorillas.key -out  
/etc/apache2/ssl/apache_igmorillas.crt
```

Nombre de país: ES

- Provincia: Granada
- Localidad: Granada
- Organización: SWAP
- Organización sección: P4
- Nombre: "igmorillas"
- Email: "igmorillas@correo.ugr.es"



C. Configurar apache con ruta de los certificados

- Configurar archivo default-ssl con los certificados SSL

```
$ sudo vi /etc/apache2/sites-available/default-ssl.conf
```

Y modificamos las líneas de SSLENGINE, SSLCertificateFile, SSLCertificateKeyFile.

```
# 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

#   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/apache2/ssl/apache_igmorillas.crt
SSLCertificateKeyFile   /etc/apache2/ssl/apache_igmorillas.key_

#   Server Certificate Chain:
#   Point SSLCertificateChainFile at a file containing the
#   concatenation of PEM encoded CA certificates which form the
```

- Activar el sitio default-ssl

```
$ sudo a2ensite default-ssl
```

```
$ sudo service apache2 reload
```

1.2 Copiar certificado autofirmado SSL a M1 y M2

Copiar al resto de máquinas servidoras (M2) y al balanceador de carga (M3) el certificado autofirmado creado en M1 (archivos .crt y .key) y configurarlas para que acepten tráfico HTTP y HTTPS.

Máquina 2 - No generar nuevos certificados

A. Copiar certificados de M1 en M2 en /etc/apache2/ssl

Máquina 1:

```
$ sudo scp apache_igmorillas.crt  
igmorillas@192.168.56.102:/home/igmorillas/apache_igmorillas.crt
```

```
$ sudo scp apache_igmorillas.key  
igmorillas@192.168.56.102:/home/igmorillas/apache_igmorillas.key
```

Máquina 2:

```
$ sudo a2enmod ssl
```

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

```
$ sudo mv /home/igmorillas/apache* /etc/apache2/ssl
```

Repetimos los pasos realizados en la máquina 1 ["Activar el sitio default-ssl"](#)

Máquina 3 - No generar nuevos certificados

A. Copiar certificados de M1 en M3 en /etc/apache2/ssl

Máquina 1:

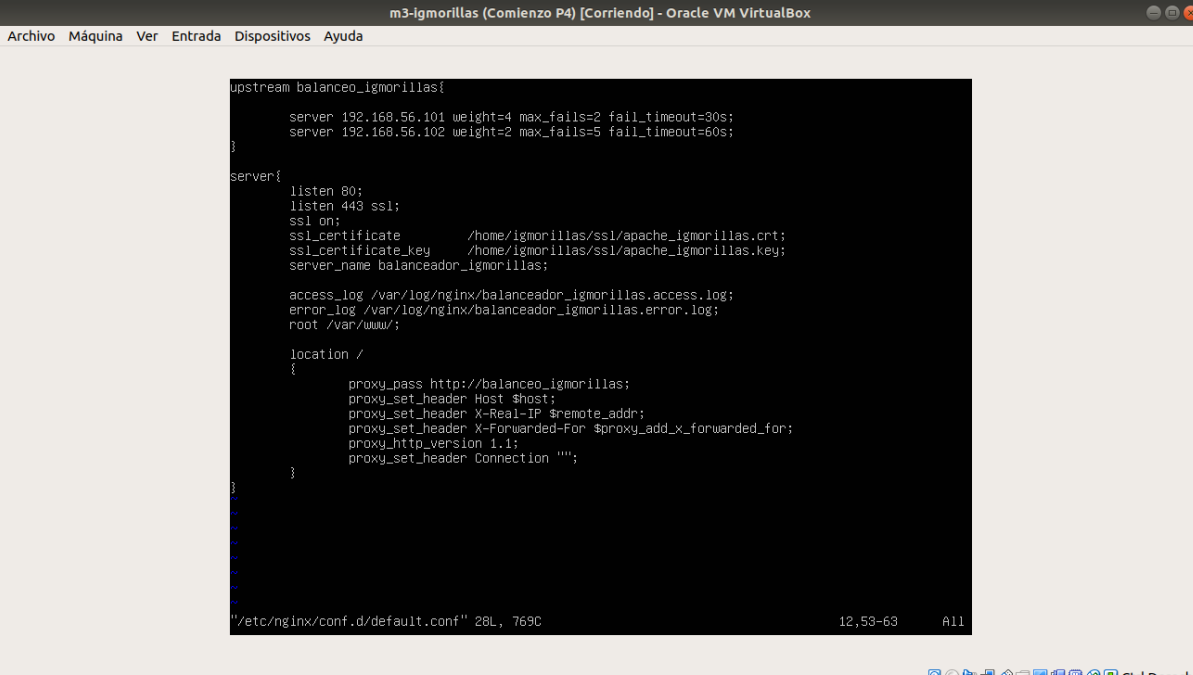
```
$ sudo scp apache_igmorillas.crt  
igmorillas@192.168.56.103:/home/igmorillas/apache_igmorillas.crt
```

```
$ sudo scp apache_igmorillas.key  
igmorillas@192.168.56.103:/home/igmorillas/apache_igmorillas.key
```

B. Configurar balanceador nginx con nuevo server con los certificados SSL y parámetros correspondientes. Añadimos a los parámetros existentes:

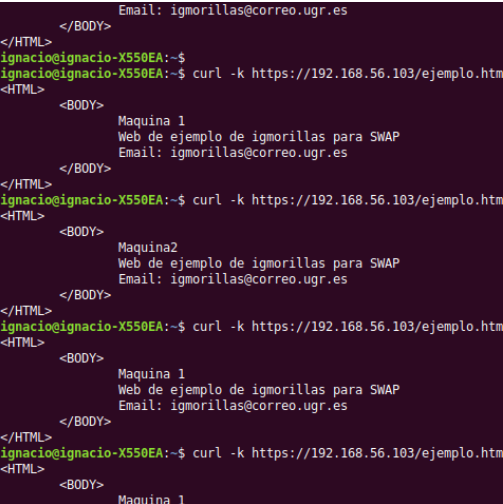
Máquina 3:

```
listen 443 ssl;  
ssl on;  
ssl_certificate /home/usuario/ssl/apache_usuarioUGR.crt;  
ssl_certificate_key /home/usuario/ssl/apache_usuarioUGR.key;
```



```
m3-igmorillas (Comienzo P4) [Corriendo] - Oracle VM VirtualBox  
Archivo Máquina Ver Entrada Dispositivos Ayuda  
  
upstream balanceo_igmorillas{  
    server 192.168.56.101 weight=4 max_fails=2 fail_timeout=90s;  
    server 192.168.56.102 weight=2 max_fails=5 fail_timeout=60s;  
}  
  
server{  
    listen 80;  
    listen 443 ssl;  
    ssl on;  
    ssl_certificate /home/igmorillas/ssl/apache_igmorillas.crt;  
    ssl_certificate_key /home/igmorillas/ssl/apache_igmorillas.key;  
    server_name balanceador_igmorillas;  
  
    access_log /var/log/nginx/balanceador_igmorillas.access.log;  
    error_log /var/log/nginx/balanceador_igmorillas.error.log;  
    root /var/www/;  
  
    location /  
    {  
        proxy_pass http://balanceo_igmorillas;  
        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 "";  
    }  
}
```

Comprobamos que funciona con http y https



```
Email: igmorillas@correo.ugr.es  
</BODY>  
</HTML>  
ignacio@ignacio-X550EA:~$  
ignacio@ignacio-X550EA:~$ curl -k https://192.168.56.103/ejemplo.html  
<HTML>  
  <BODY>  
    Maquina 1  
    Web de ejemplo de igmorillas para SWAP  
    Email: igmorillas@correo.ugr.es  
  </BODY>  
</HTML>  
ignacio@ignacio-X550EA:~$ curl -k https://192.168.56.103/ejemplo.html  
<HTML>  
  <BODY>  
    Maquina2  
    Web de ejemplo de igmorillas para SWAP  
    Email: igmorillas@correo.ugr.es  
  </BODY>  
</HTML>  
ignacio@ignacio-X550EA:~$ curl -k https://192.168.56.103/ejemplo.html  
<HTML>  
  <BODY>  
    Maquina 1  
    Web de ejemplo de igmorillas para SWAP  
    Email: igmorillas@correo.ugr.es  
  </BODY>  
</HTML>  
ignacio@ignacio-X550EA:~$ curl -k https://192.168.56.103/ejemplo.html  
<HTML>  
  <BODY>  
    Maquina 1  
    Web de ejemplo de igmorillas para SWAP  
    Email: igmorillas@correo.ugr.es  
  </BODY>  
</HTML>
```



```
Web de ejemplo de igmorillas para SWAP  
Email: igmorillas@correo.ugr.es  
</BODY>  
</HTML>  
ignacio@ignacio-X550EA:~$  
ignacio@ignacio-X550EA:~$ curl http://192.168.56.103/ejemplo.html  
<HTML>  
  <BODY>  
    Maquina 1  
    Web de ejemplo de igmorillas para SWAP  
    Email: igmorillas@correo.ugr.es  
  </BODY>  
</HTML>  
ignacio@ignacio-X550EA:~$ curl http://192.168.56.103/ejemplo.html  
<HTML>  
  <BODY>  
    Maquina2  
    Web de ejemplo de igmorillas para SWAP  
    Email: igmorillas@correo.ugr.es  
  </BODY>  
</HTML>  
ignacio@ignacio-X550EA:~$ curl http://192.168.56.103/ejemplo.html  
<HTML>  
  <BODY>  
    Maquina 1  
    Web de ejemplo de igmorillas para SWAP  
    Email: igmorillas@correo.ugr.es  
  </BODY>  
</HTML>  
ignacio@ignacio-X550EA:~$ curl http://192.168.56.103/ejemplo.html  
<HTML>  
  <BODY>  
    Maquina 1  
    Web de ejemplo de igmorillas para SWAP  
    Email: igmorillas@correo.ugr.es  
  </BODY>  
</HTML>
```

2. Tareas avanzadas:

1. Permitir SSH, PING y DNS a las máquinas M1, M2 y M3 así como el tráfico consigo misma (localhost). El resto de servicios y/o peticiones debe denegarse.
2. Configurar M3 estableciendo reglas de iptables para que sólo M3 sea quien acepte peticiones HTTP y HTTPS mientras que M1 y M2 no acepten peticiones a no ser que sean peticiones provenientes de M3.
3. Hacer que la configuración del cortafuegos se ejecute al arranque del sistema en todas las máquinas.
4. Crear, instalar y configurar un certificado SSL con Cerbot u otro.

2.1. IPTABLES (scripts)

Crearemos 3 scripts, uno para cada máquina, que será el mismo para el caso de la Máquina 1 y 2.

Máquina 1 y 2

- Permiten SSH, PING, DNS y localhost a la máquina 1 y 2 y deniegan las demás.
- M1 y M2 solo aceptan peticiones HTTP y HTTPS de la Máquina 3

```
m1-igmorillas (Instantánea 6) [Corriendo] - Oracle VM VirtualBox
Dispositivos  Ayuda

#!/bin/bash

# Eliminar todas las reglas
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X

# Denegar todo el trafico de informacion
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# permitir TCP
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

# permitir localhost
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# permitir DNS
iptables -A INPUT -p tcp --dport 53 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 53 -j ACCEPT
# permitir entrar SSH
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT

# permitir entrar ping
iptables -A INPUT -p icmp -j ACCEPT
iptables -A OUTPUT -p icmp -j ACCEPT

# permitir HTTP
iptables -A INPUT -s 192.168.56.103 -p tcp --dport 80 -j ACCEPT
iptables -A OUTPUT -s 192.168.56.103 -p tcp --sport 80 -j ACCEPT

# permitir HTTPS
iptables -A INPUT -s 192.168.56.103 -p tcp --dport 443 -j ACCEPT
iptables -A OUTPUT -s 192.168.56.103 -p tcp --sport 443 -j ACCEPT
```

Máquina 3

- Permiten SSH, PING, DNS, localhost, HTTP y HTTPS a la máquina 3 y deniegan las demás.

```
m3-igmorillas (fin iptables paso a paso) [Corriendo] - Oracle VM VirtualBox
Dispositivos Ayuda

#!/bin/bash

# Eliminar todas las reglas
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X

# Denegar todo el trafico de informacion
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# permitir TCP
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

# permitir localhost
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# permitir DNS
iptables -A INPUT -p tcp --dport 53 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 53 -j ACCEPT
# permitir entrar SSH
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT

# permitir entrar ping
iptables -A INPUT -p icmp -j ACCEPT
iptables -A OUTPUT -p icmp -j ACCEPT

# permitir HTTP
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 80 -j ACCEPT

# permitir HTTPS
iptables -A INPUT -p tcp --dport 443 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 443 -j ACCEPT
```

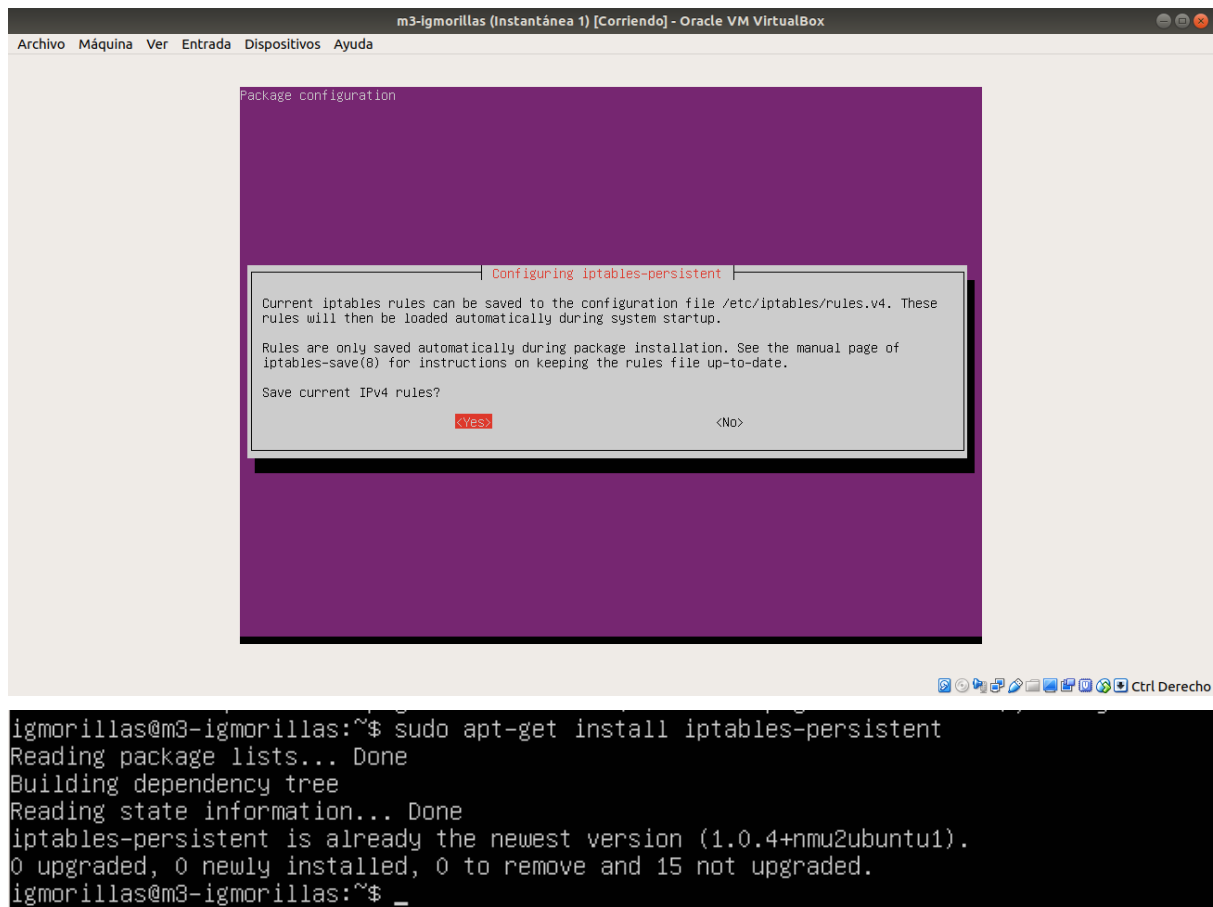
41,1 Bot

2.2. Configurar cortafuegos al arranque (hacer persistente reglas IPTABLES)

Hacer que la configuración del cortafuegos se ejecute al arranque del sistema en todas las máquinas.

Ejecutando el comando siguiente para hacerlas persistentes las reglas iptables entre arranques.

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

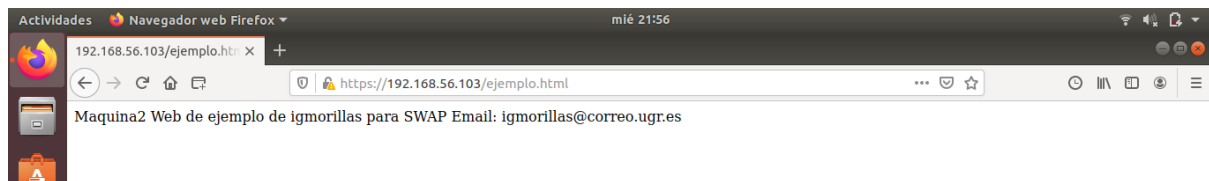


Vemos que funciona

HTTP



HTTPS



SSH

```
ignacio@ignacio-X550EA:~$ ssh igmorillas@192.168.56.101
igmorillas@192.168.56.101's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-142-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue May 11 18:06:49 UTC 2021

System load:  0.0               Processes:            93
Usage of /:   47.6% of 8.79GB   Users logged in:     1
Memory usage: 29%              IP address for enp0s3: 10.0.2.15
Swap usage:   0%               IP address for enp0s8: 192.168.56.101

 * Pure upstream Kubernetes 1.21, smallest, simplest cluster ops!

https://microk8s.io/

77 packages can be updated.
8 updates are security updates.

New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue May 11 17:46:48 2021
igmorillas@m1-igmorillas:~$
```

PING

```
ignacio@ignacio-X550EA:~$ ping 192.168.56.101
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.
64 bytes from 192.168.56.101: icmp_seq=1 ttl=64 time=0.709 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.748 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=0.868 ms
64 bytes from 192.168.56.101: icmp_seq=4 ttl=64 time=0.715 ms
^C
--- 192.168.56.101 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3034ms
rtt min/avg/max/mdev = 0.709/0.760/0.868/0.064 ms
```

2.3. Certbot

Primero de todo aclarar que no he podido finalizar la instalación porque he tenido problemas con la parte del DNS, pero igualmente describo los pasos a realizar.

1. Instalar Certbot y su complemento de Nginx.

```
$ sudo apt install certbot python3-certbot-nginx
```

2. Confirmar la configuración de Nginx
Lo hace buscando una directiva "server_name" que coincida con el dominio para el que está solicitando el certificado. Para ello debe estar en "/etc/nginx/sites-available/example.com"

```
$ sudo vi /etc/nginx/sites-available/example.com
```

Buscaremos la linea

```
...
server_name example.com www.example.com;
...
```

Y verificamos la sintaxis de las modificaciones de la configuración.

```
$ sudo nginx -t
```

Cargamos la nueva configuración de Nginx

```
$ sudo systemctl reload nginx
```

Para nuestros servidores apache

3. Instalar Certbot y su complemento de Nginx.

```
$ sudo apt install python-certbot-apache
```

4. Confirmar la configuración de Nginx
Lo hace buscando una directiva "server_name" que coincida con el dominio para el que está solicitando el certificado. Para ello debe estar en "/etc/nginx/sites-available/example.com"

```
$ sudo vi /etc/nginx/sites-available/example.com
```

Buscaremos la linea

```
...  
server_name example.com www.example.com;  
...
```

Y verificamos la sintaxis de las modificaciones de la configuración.

```
$ sudo apache2ctl configtest
```

Cargamos la nueva configuración de Nginx

```
$ sudo systemctl reload apache2
```

Siguiendo estos pasos debería obtener mi certificado SSL.