Servidores Web de Altas Prestaciones

Práctica 4 - Asegurar la granja web

César Muñoz Reinoso

Curso 2022-2023

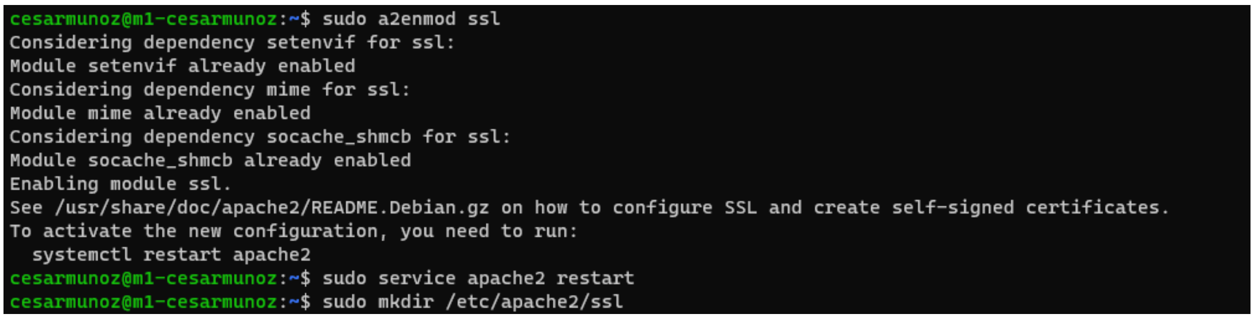
7 de mayo de 2023

Índice

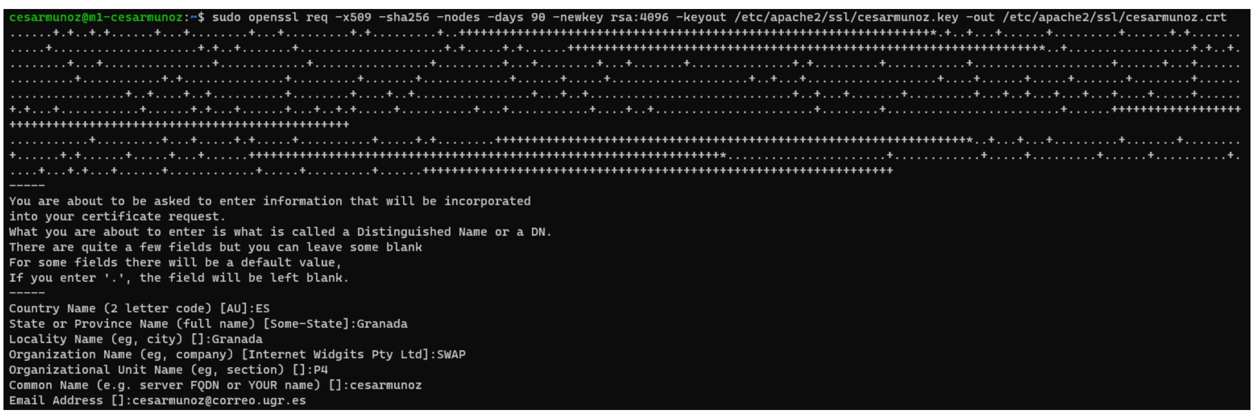
1. Certificado autofirmado SSL

1.1. Generar e instalar un certificado autofirmado

Para comenzar, utilizaremos un certificado propio SSL autofirmado usando openssl. Tendremos que habilitar el módulo SSL en Apache y crear una carpeta para almacenar los certificados SSL. Luego en la máquina M1 crearemos los archivos cesarmunoz.key y cesarmunoz.crt, que son la clave privada y el certificado firmado, respectivamente.



```
cesarmunoz@m1-cesarmunoz:~$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
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
cesarmunoz@m1-cesarmunoz:~$ sudo service apache2 restart
cesarmunoz@m1-cesarmunoz:~$ sudo mkdir /etc/apache2/ssl
```



```
cesarmunoz@m1-cesarmunoz:~$ sudo openssl req -x509 -sha256 -nodes -days 90 -newkey rsa:4096 -keyout /etc/apache2/ssl/cesarmunoz.key -out /etc/apache2/ssl/cesarmunoz.crt
-----
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) []:cesarmunoz
Email Address []:cesarmunoz@correo.ugr.es
```

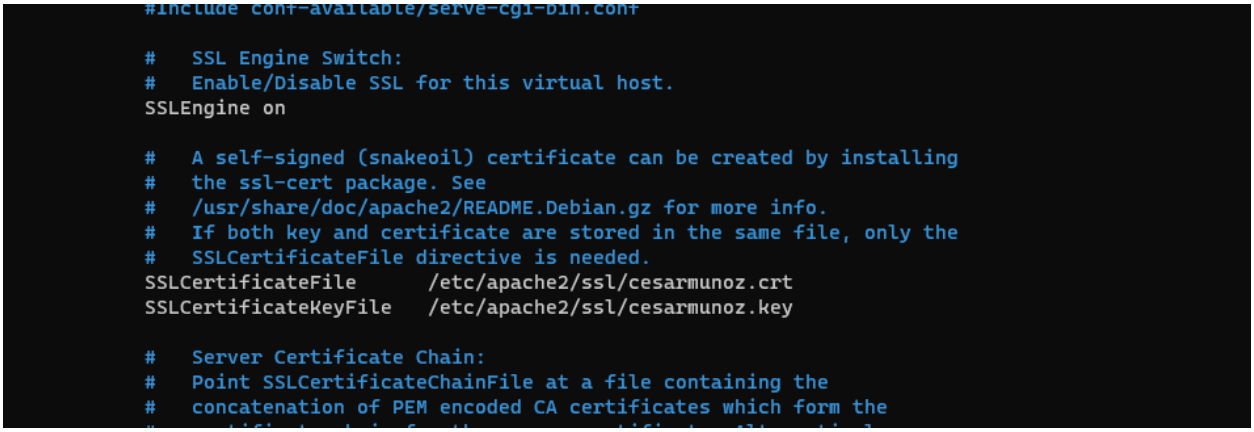
```
sudo openssl req -x509 -sha256 -nodes -days 90 -newkey rsa:4096 -keyout
/etc/apache2/ssl/cesarmunoz.key -out /etc/apache2/ssl/cesarmunoz.crt
```

Crearemos un certificado X509 que es formato estandarizado que se usa para los certificados de clave púbica. Una criptografia SHA256 que es la más segura por el momento, una validez de 90 días y una certificación de 4096-bit RSA. Las almacenamos en la carpeta /etc/apache2/ssl anteriormente creada.

Utilizamos la configuración de certificado de dominio:
- Nombre de país: ES
- Provincia: Granada
- Localidad: Granada
- Organización: SWAP
- Organización sección: P4
- Nombre: cesarmunoz
- Email: cesarmunoz@correo.ugr.es

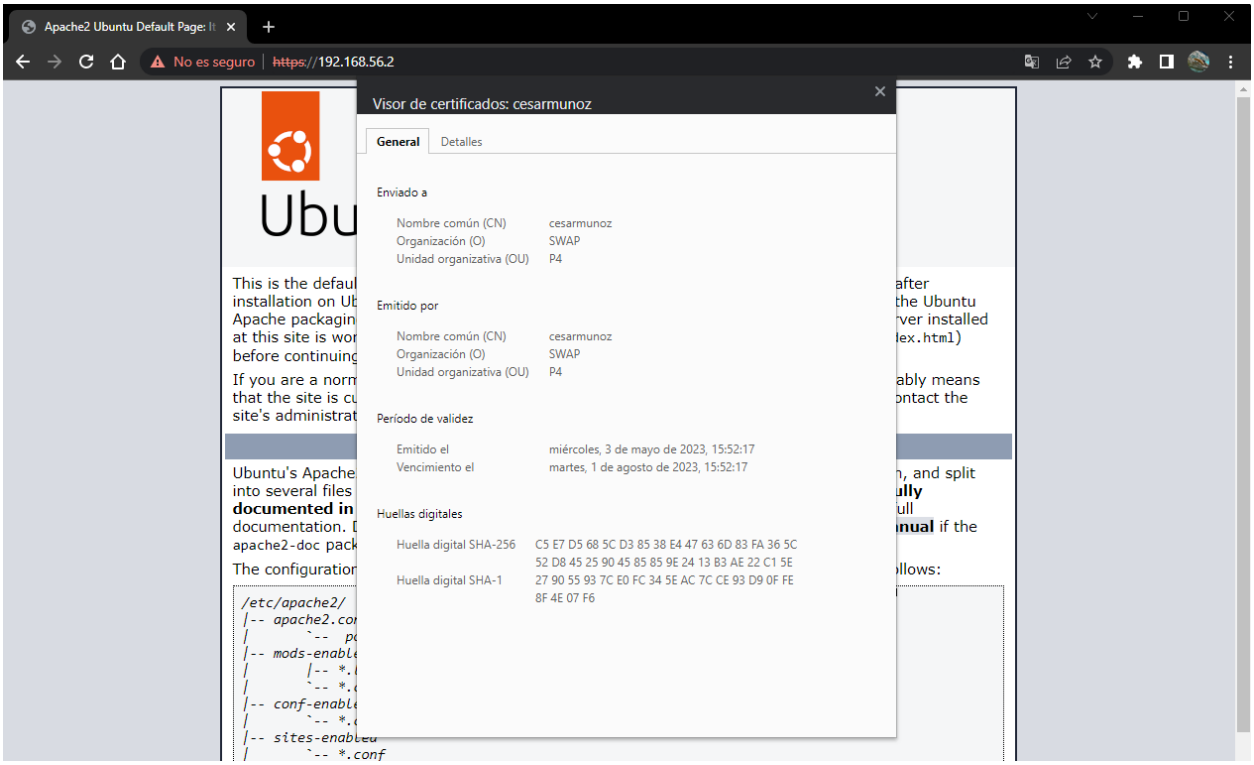
1.2. Apache con certificado SSL

Configuraremos Apache con la ruta a los certificados. Editaremos el archivo /etc/apache2/sites-available/default-ssl.conf, añadiendo tres lineas para localizar los archivos del certificado. Acto seguido, aplicamos cambios y reiniciamos apache.




```
cesarmunoz@m1-cesarmunoz:~$ sudo nano /etc/apache2/sites-available/default-ssl.conf
[sudo] password for cesarmunoz:
cesarmunoz@m1-cesarmunoz:~$ sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
cesarmunoz@m1-cesarmunoz:~$ sudo service apache2 reload
```

Comprobamos que efectivamente podemos acceder mediante HTTPS a la máquina M1 y que estamos utilizando el certificado autofirmado.


Apache2 Ubuntu Default Page: It
No es seguro | https://192.168.56.2
Visor de certificados: cesarmunoz
General
Detalles
Enviado a
Nombre común (CN) cesarmunoz
Organización (O) SWAP
Unidad organizativa (OU) P4
Emitido por
Nombre común (CN) cesarmunoz
Organización (O) SWAP
Unidad organizativa (OU) P4
Período de validez
Emitido el miércoles, 3 de mayo de 2023, 15:52:17
Vencimiento el martes, 1 de agosto de 2023, 15:52:17
Huellas digitales
Huella digital SHA-256 C5 E7 D5 68 5C D3 85 38 E4 47 63 6D 83 FA 36 5C 52 D8 45 25 90 45 85 85 9E 24 13 B3 AE 22 C1 5E
Huella digital SHA-1 27 90 55 93 7C E0 FC 34 5E AC 7C CE 93 D9 0F FE 8F 4E 07 F6

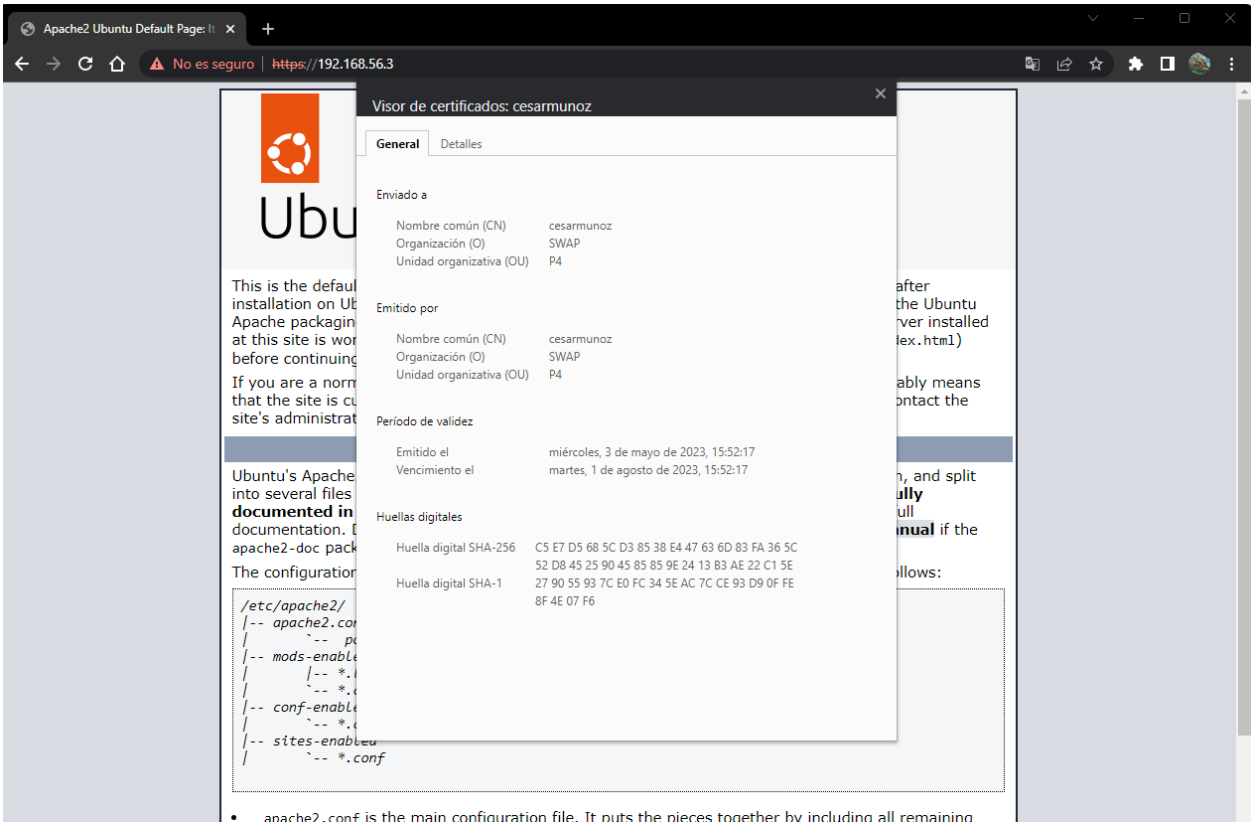
1.3. Configuración de máquina M2

Para que pueda utilizarse el mismo certificado con la máquina m2, copiamos el certificado con la directiva estudiada anteriormente scp.

```
cesarmunoz@m1-cesarmunoz:~$ sudo scp /etc/apache2/ssl/cesarmunoz.crt cesarmunoz@192.168.56.3:/home/cesarmunoz/cesarmunoz.crt
[sudo] password for cesarmunoz:
The authenticity of host '192.168.56.3 (192.168.56.3)' can't be established.
ED25519 key fingerprint is SHA256:0bDDV+3E86Qsp3tgCmpd8iyhV6jvYSnlKoR+XiCc/Zo.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.3' (ED25519) to the list of known hosts.
cesarmunoz@192.168.56.3's password:
cesarmunoz.crt                                                                100% 2130   962.1KB/s   00:00
cesarmunoz@m1-cesarmunoz:~$ sudo scp /etc/apache2/ssl/cesarmunoz.key cesarmunoz@192.168.56.3:/home/cesarmunoz/cesarmunoz.key
cesarmunoz@192.168.56.3's password:
cesarmunoz.key                                                                100% 3272     1.5MB/s   00:00
```

Seguimos como en M1, creando la carpeta /etc/apache2/ssl y moviendo ahi los certificados, activando el modulo SSL, configurando default-ssl, activando el sitio y reiniciando.

```
cesarmunoz@m2-cesarmunoz:~$ sudo mkdir /etc/apache2/ssl
cesarmunoz@m2-cesarmunoz:~$ sudo mv /home/cesarmunoz/cesarmunoz.* /etc/apache2/ssl
cesarmunoz@m2-cesarmunoz:~$ sudo a2enmod ssl
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.
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
cesarmunoz@m2-cesarmunoz:~$ sudo service apache2 restart
cesarmunoz@m2-cesarmunoz:~$ sudo nano /etc/apache2/sites-available/default-ssl.conf
cesarmunoz@m2-cesarmunoz:~$ sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
cesarmunoz@m2-cesarmunoz:~$ sudo service apache2 reload
```



1.4. Nginx como balanceador para peticiones HTTPS

Como en la máquina M2, copiamos los archivos del certificado al balanceador M3. Luego editamos el archivo /etc/nginx/conf.d/default.conf.

Incluimos el puerto de escucha 443, que corresponde con las peticiones HTTPS, mantenemos la escucha en el puerto 80 para que se puedan hacer también por HTTP. Habilitamos el SSL y especificamos donde están los archivos del certificado autofirmado. Las directivas ssl_protocols y ssl_ciphers se pueden utilizar para limitar las conexiones y permitir únicamente las versiones y cifrados de SSL que queramos. Comprobamos que admite una conexión mediante HTTPS.

```
cesarmunoz@m1-cesarmunoz:~$  sudo scp /etc/apache2/ssl/cesarmunoz.crt cesarmunoz@192.168.56.4:/home/cesarmunoz/cesarmunoz.crt
[sudo] password for cesarmunoz:
The authenticity of host '192.168.56.4 (192.168.56.4)' can't be established.
ED25519 key fingerprint is SHA256:Kd56b6JxVlkj7fGo2o0AJgN8LzXXPIh9YNUlSAXWDpE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.4' (ED25519) to the list of known hosts.
cesarmunoz@192.168.56.4's password:
cesarmunoz.crt                                                                100% 2130     1.2MB/s   00:00
cesarmunoz@m1-cesarmunoz:~$ sudo scp /etc/apache2/ssl/cesarmunoz.key cesarmunoz@192.168.56.4:/home/cesarmunoz/cesarmunoz.key
cesarmunoz@192.168.56.4's password:
cesarmunoz.key                                                                100% 3272     1.6MB/s   00:00
```

```
 GNU nano 6.2                                    /etc/nginx/conf.d/default.conf
upstream balanceo_cesarmunoz {
        server 192.168.56.2 weight=1 max_fails=3;
        server 192.168.56.3 weight=1;
        keepalive 3;
}

server{
        listen 80;
        listen 443 ssl;
        ssl on;
        ssl_certificate /home/cesarmunoz/ssl/cesarmunoz.crt;
        ssl_certificate_key /home/cesarmunoz/ssl/cesarmunoz.key;
        ssl_protocols       TLSv1 TLSv1.1 TLSv1.2 TLSv1.3;
        ssl_ciphers         HIGH:!aNULL:!MD5;
        server_name balanceador_cesarmunoz;

        access_log /var/log/nginx/balanceador_cesarmunoz.access.log;
        error_log /var/log/nginx/balanceador_cesarmunoz.error.log;
        root /var/www/;
        location /
        {
                proxy_pass http://balanceo_cesarmunoz;
                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 "";
        }
}
```

```
cesarmunoz@m2-cesarmunoz:~$ curl -k https://192.168.56.4/swap.html
<HTML>
<BODY>
MAQUINA M2
Web de ejemplo de cesarmunoz para SWAP
Email: cesarmunoz@correo.ugr.es
</BODY>
</HTML>
cesarmunoz@m2-cesarmunoz:~$ curl -k https://192.168.56.4/swap.html
<HTML>
<BODY>
MAQUINA M1
Web de ejemplo de cesarmunoz para SWAP
Email: cesarmunoz@correo.ugr.es
</BODY>
</HTML>
```

2. Certificado Cerbot

Para poder utilizar un certificado Cerbot necesitaremos tener un dominio y asociarlo a nuestra granja web. En esta práctica lo haremos de forma teórica. Instalamos en el balanceador de carga M3 y en los servidores M1 y M2 los paquetes necesarios de Cerbot.

En mi proyecto final de la asignatura explicaré como se debe hacer de manera correcta configurando un servidor con ip dinámica y un dominio con DuckDNS.

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

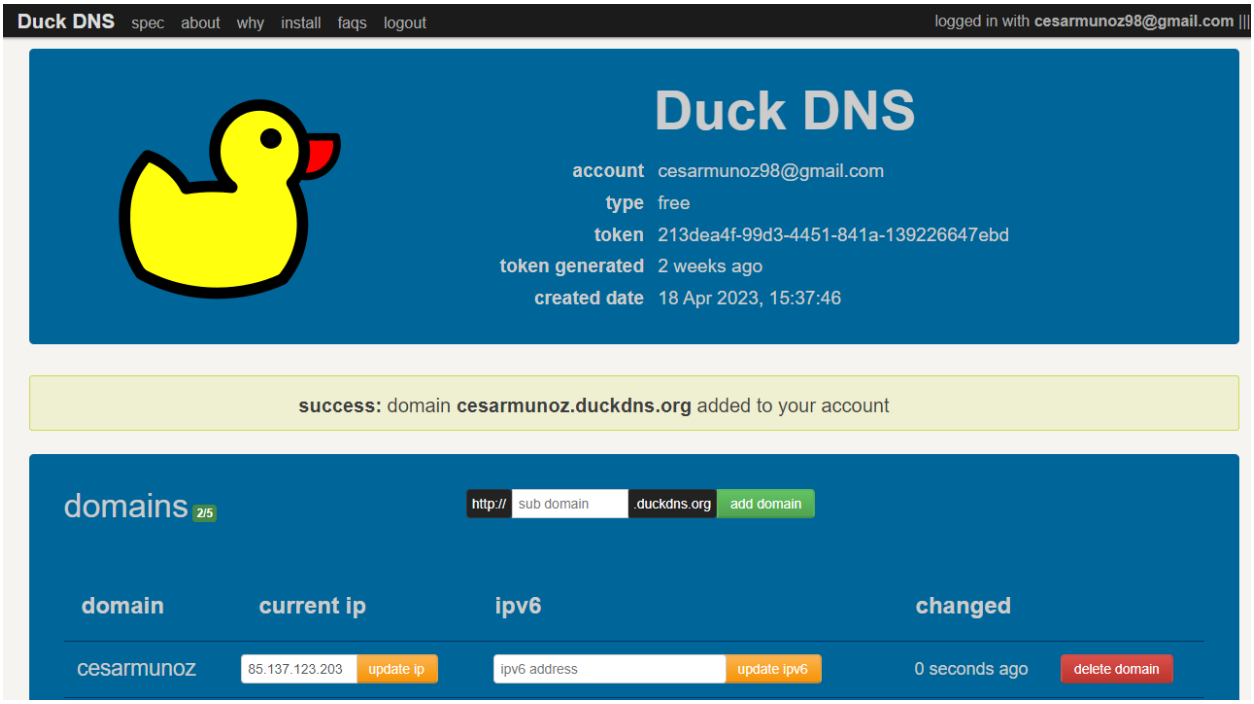
```
sudo apt install certbot python3-certbot-apache
```

Acto seguido certificaremos y modificaremos la configuración de Nginx y de Apache con

```
certbot --nginx -d cesarmunoz.duckdns.org
```

```
certbot --apache -d cesarmunoz.duckdns.org
```

Ya tendríamos configurado nuestra granja web con un certificado Cerbot.



3. Configuración del cortafuegos

Para que cada máquina funcione adecuadamente y no tengamos brechas de seguridad, necesitamos configurar el cortafuegos en cada una. Crearemos un script con todos los comandos de iptables necesarios, además como en cada reinicio de las máquinas la configuración de iptables se restablece, crearemos un daemon que se ejecutará en cada arranque del sistema.

3.1. Cortafuegos en balanceador de carga

En el balanceador de carga M3 hemos optado por eliminar todas las reglas anteriores y denegar todo trafico. Luego hemos habilitado HTTP/HTTPS, SSH y DNS y toda conexión desde localhost, asi como el protocolo icmp para permitir PING.

```
  GNU nano 6.2                                    ssl/iptables_script.sh
#!/bin/bash

#Eliminar al mismo tiempo todas sus reglas

iptables -F
iptables -X
iptables -Z
iptables -t nat -F

#Denegar todo el trafico

iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

#Permitimos la entrada de conexiones establecidas y relacionadas, la salida de nuevas, establecidas y relacionadas

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

#Permitir cualquier acceso desde localhost (interface lo):

iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

#Abrir el puerto 22 para permitir el acceso por SSH

iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT

#Abrir los puertos HTTP/HTTPS (80 y 443)

iptables -A INPUT -p tcp -m multiport --dports 80,443 -j ACCEPT
iptables -A OUTPUT -p tcp -m multiport --sports 80,443 -j ACCEPT

#Abrir el puerto 53 para permitir el acceso a DNS

iptables -A INPUT -m state --state NEW -p udp --dport 53 -j ACCEPT
iptables -A INPUT -m state --state NEW -p tcp --dport 53 -j ACCEPT

#Habilitar PING

iptables -A INPUT -p icmp -j ACCEPT
iptables -A OUTPUT -p icmp -j ACCEPT
```

3.2. Cortafuegos en Apache

En las máquinas M1 y M2 hemos vuelto a eliminar todas las reglas anteriores y denegar todo trafico y tan solo habilitado el trafico desde el balanceador de carga de HTTP/HTTPS y SSH. Esto se debe a que dichos servidores de la granja web no deben ser accesibles desde fuera de la red sin pasar por el balanceador. Comprobamos que no tenemos acceso a M2 desde la máquina afitrión, pero si podemos conectarnos mediante ssh a M2 desde M3.

```
  GNU nano 6.2                                          iptables_script.sh
#!/bin/bash

#Eliminar al mismo tiempo todas sus reglas

iptables -F
iptables -X
iptables -Z
iptables -t nat -F

#Denegar todo el trafico

iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

#Abrir los puertos SSH y HTTP/HTTPS (22, 80 y 443)
iptables -A INPUT -s 192.168.56.4 -p tcp -m multiport --dports 22,80,443 -j ACCEPT
iptables -A OUTPUT -d 192.168.56.4 -p tcp -m multiport --sports 22,80,443 -j ACCEPT
```

```
PS C:\Users\César> curl 192.168.56.3/swap.html
curl : No es posible conectar con el servidor remoto
En línea: 1 Carácter: 1
+ curl 192.168.56.3/swap.html
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : InvalidOperation: (System.Net.HttpWebRequest:HttpWebRequest) [Invoke-WebRequest], WebException
    + FullyQualifiedErrorId : WebCmdletWebResponseException,Microsoft.PowerShell.Commands.InvokeWebRequestCommand
```

```
cesarmunoz@m3-cesarmunoz:~$ ssh cesarmunoz@192.168.56.3
cesarmunoz@192.168.56.3's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-71-generic x86_64)

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

  System information as of vie 05 may 2023 16:40:18 UTC

  System load:  0.05615234375     Processes:               110
  Usage of /:   70.4% of 8.02GB   Users logged in:         1
  Memory usage: 58%               IPv4 address for enp0s3: 10.0.2.15
  Swap usage:   0%                IPv4 address for enp0s8: 192.168.56.3

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

     https://ubuntu.com/pro

El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 48 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»


Last login: Fri May  5 16:31:26 2023 from 192.168.56.4
cesarmunoz@m2-cesarmunoz:~$
```

3.3. Daemon para iptables persistente

Para hacer que los comandos iptables sean persistentes, creamos un daemon para ejecutar el script al inicio del arranque del sistema. En él se especifica que se debe ejecutar tras el login del sistema, la direccion del script sh y para todos los usuarios. Para que se ejecuten reiniciamos y habilitamos el daemon creado. Realizamos la misma tarea en todas las máquinas.

```
  GNU nano 6.2                          /etc/systemd/system/iptables_daemon.service
[Unit]
After=syslog.target

[Service]
ExecStart=/home/cesarmunoz/ssl/iptables_script.sh

[Install]
RequiredBy=multi-user.target
```

```
cesarmunoz@m2-cesarmunoz:~$ sudo nano /etc/systemd/system/iptables_daemon.service
cesarmunoz@m2-cesarmunoz:~$ systemctl daemon-reload
==== AUTHENTICATING FOR org.freedesktop.systemd1.reload-daemon ===
Authentication is required to reload the systemd state.
Authenticating as: César Muñoz Reinoso (cesarmunoz)
Password:
==== AUTHENTICATION COMPLETE ===
cesarmunoz@m2-cesarmunoz:~$ systemctl enable iptables_daemon.service
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-unit-files ===
Authentication is required to manage system service or unit files.
Authenticating as: César Muñoz Reinoso (cesarmunoz)
Password:
==== AUTHENTICATION COMPLETE ===
Created symlink /etc/systemd/system/multi-user.target.requires/iptables_daemon.service → /etc/systemd/system/iptables_daemon.serv
ice.
==== AUTHENTICATING FOR org.freedesktop.systemd1.reload-daemon ===
Authentication is required to reload the systemd state.
Authenticating as: César Muñoz Reinoso (cesarmunoz)
Password:
==== AUTHENTICATION COMPLETE ===
```