

2ª DAW – Despliegue de aplicaciones web



Práctica 3 – NGINX en AWS con IP elástica y SSL

Nombre y apellidos	Calificación
--------------------	--------------

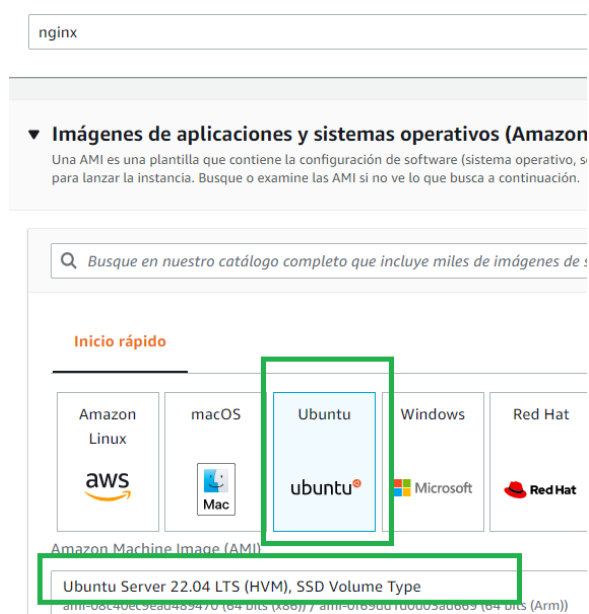
El **objetivo** es que el/la alumn@ aprenda a instalar y a configurar el servidor web de **Nginx** en un Ubuntu y aprenda las configuraciones mínimas necesarias para publicar sitios web en este servidor. Utilizaremos una máquina de **AWS** nueva para realizar esta configuración. Además:

- Aprenderemos a crear y asignar una estática (elastic IP) a nuestra instancia para que siempre que la levantemos, tengamos la misma IP y no varíe.
- Publicaréis en el subdominio que compró la profe la web.
- Securitizaremos el servidor instalando un certificado para vuestro dominio.
*Esto realmente lo haré yo porque vosotros, si no tenéis un dominio comprado (o bien podéis utilizar el de www.freenom.com)

NGINX es software libre y de código abierto. Es **servidor web/proxy** inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico. Un proxy inverso es un tipo de servidor proxy que recupera recursos en nombre de un cliente desde uno o más servidores. Básicamente hace de intermediario, es una capa más de seguridad en redes así las peticiones de recursos que realiza un cliente de la red interna realmente las realiza el proxy. Estos recursos se devuelven al cliente como si ese cliente fuera el que realmente los hubiera pedido. Este tipo de servidor no se van a configurar en este módulo, queda fuera de los contenidos (seguridad informática).

1ª Creación de la instancia en AWS

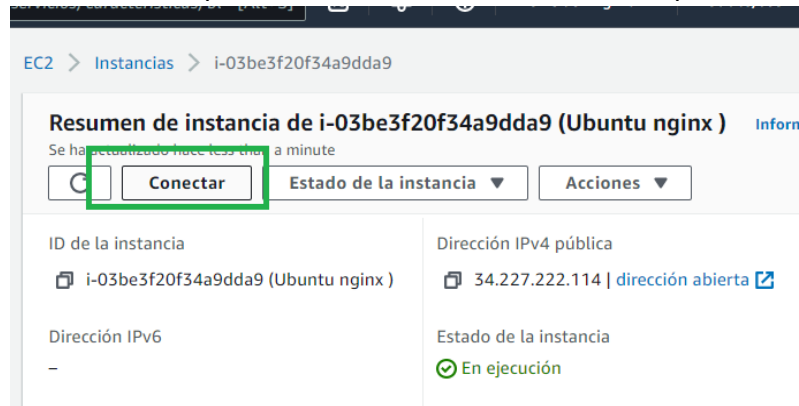
Lanzamos una nueva instancia en Amazon AWS “Ubuntu Server 22”:



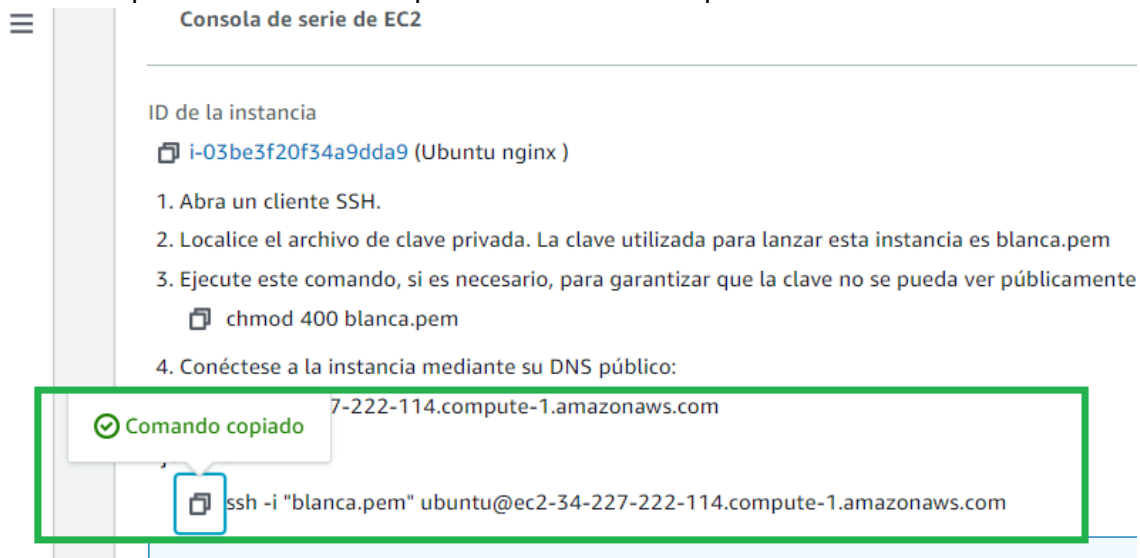
2ª DAW – Despliegue de aplicaciones web

Ponerle de nombre Ubuntu nginx, por ejemplo, para tener localizado lo que tenéis configurado en esta máquina.

Conectamos a la máquina a través de la consola online o por SSH:

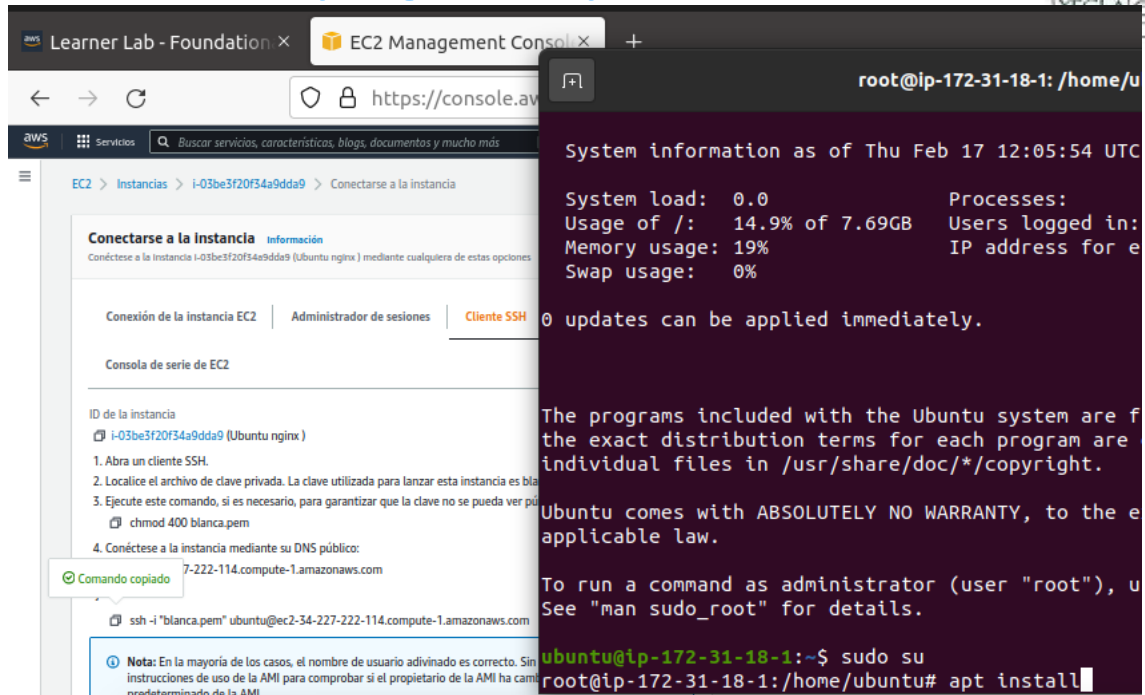


Copiáis el comando SSH para conectar a la máquina:



Debéis de tener la clave privada localizada para cuando accedáis lo hagáis desde el directorio dónde tenéis la **.pem**:

2ª DAW – Despliegue de aplicaciones web



The screenshot shows the AWS Management Console interface for an EC2 instance. On the left, the 'Conectar a la instancia' (Connect to instance) page is visible, showing the instance ID 'i-03be3f20f34a9dda9' and the public IP address '7-222-114.compute-1.amazonaws.com'. A terminal window is open on the right, displaying system information for the Ubuntu instance, including system load, memory usage, and swap usage. The terminal also shows the command 'sudo su' being executed, resulting in a root prompt.

2ª Instalación del servidor web en la instancia en AWS

Seguidamente, instalación del servidor web **NGINX**:

apt install nginx

```
/systemd/system/nginx.service.  
Setting up libjpeg-turbo8:amd64 (1.5.2-0ubuntu5.18.04.4) ...  
Setting up libnginx-mod-mail (1.14.0-0ubuntu1.9) ...  
Setting up libxpm4:amd64 (1:3.5.12-1) ...  
Setting up libnginx-mod-http-xslt-filter (1.14.0-0ubuntu1.9) ...  
Setting up libnginx-mod-http-geoip (1.14.0-0ubuntu1.9) ...  
Setting up libwebp6:amd64 (0.6.1-2ubuntu0.18.04.1) ...  
Setting up libjpeg8:amd64 (8c-2ubuntu8) ...  
Setting up fontconfig-config (2.12.6-0ubuntu2) ...  
Setting up libnginx-mod-stream (1.14.0-0ubuntu1.9) ...  
Setting up libtiff5:amd64 (4.0.9-5ubuntu0.4) ...  
Setting up libfontconfig1:amd64 (2.12.6-0ubuntu2) ...  
Setting up libgd3:amd64 (2.2.5-4ubuntu0.5) ...  
Setting up libnginx-mod-http-image-filter (1.14.0-0ubuntu1.9) ...  
Setting up nginx-core (1.14.0-0ubuntu1.9) ...  
Setting up nginx (1.14.0-0ubuntu1.9) ...  
Processing triggers for systemd (237-3ubuntu10.52) ...  
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...  
Processing triggers for ufw (0.36-0ubuntu0.18.04.2) ...  
Processing triggers for ureadahead (0.100.0-21) ...  
Processing triggers for libc-bin (2.27-3ubuntu1.4) ...  
root@ip-172-31-18-1:/home/ubuntu#
```

Comprobamos que el servidor se queda activo:

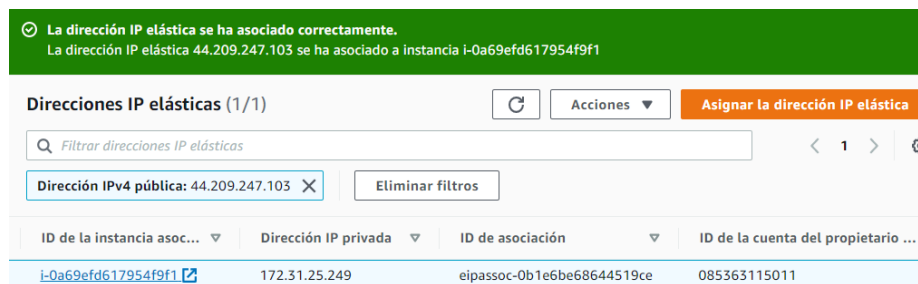
2ª DAW – Despliegue de aplicaciones web

```
Processing triggers for libc-bin (2.27-3ubuntu1.4) ...
root@ip-172-31-18-1:/home/ubuntu# systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: en
   Active: active (running) since Thu 2022-02-17 12:07:04 UTC; 1min 31s ago
     Docs: man:nginx(8)
    Main PID: 2309 (nginx)
      Tasks: 2 (limit: 1140)
    CGroup: /system.slice/nginx.service
            └─2309 nginx: master process /usr/sbin/nginx -g daemon on; master_pro
               └─2312 nginx: worker process

Feb 17 12:07:04 ip-172-31-18-1 systemd[1]: Starting A high performance web serve
Feb 17 12:07:04 ip-172-31-18-1 systemd[1]: nginx.service: Failed to parse PID fr
Feb 17 12:07:04 ip-172-31-18-1 systemd[1]: Started A high performance web server
lines 1-13/13 (END)
```

3º Creación IP elástica y asignación al servidor web en la instancia en AWS creada.

Creamos una IP elástica (ver en recursos extra en Aula Virtual) y se la asignamos a nuestra instancia creada (a la máquina Server Ubuntu con nginx instalado):



4º Modificación del DNS para publicar vuestro sitio web:

Opción 1

Configurarlo en vuestro dominio gratuito adquirido en www.freenom.com

Opción 2

Si no llegaste a comprar el dominio gratuito en www.nominalia.com o en www.freenom.com, me pasáis la IP elástica que habéis asignado a vuestro nginx al siguiente foro: <https://aulavirtual.murciaeduca.es/mod/forum/view.php?id=3328971>

Yo revisaré el foro y haré los cambios en el DNS de Nominalia para que podáis finalizar la práctica y poder probarlo todo bien:

2ª DAW – Despliegue de aplicaciones web



controlpanel.nominalia.com/domains/dns.html

Configuración actual de DNS

Soporte Técnico

¿Necesitas ayuda? ¡Solicita Soporte!

Crear / Modificar un registro A

Utiliza este formulario para crear o modificar un registro A (Address Record) para tu dominio.

Nombre: .despliegueweb.es

Añadir un registro A para la zona

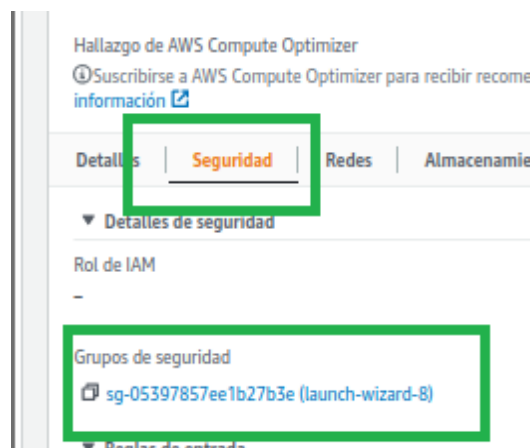
El registro A (Address Record) te permite asociar un "nombre" (Hostname) a una dirección IP. Cada dominio tiene un registro A primario que asocia tu dominio a una dirección IP.

Introduce la dirección

IP:

Uno o más registros A pueden estar asociados a la misma dirección IP, por ejemplo, porque esta misma dirección IP proporciona más servicios o bien porque cada registro A (prácticamente cada subdominio) puede asociarse a una IP distinta. Puedes crear todos los registros A que quieras y asociarlos a las direcciones IP que prefieras. Te recomendamos utilizar solo caracteres alfanuméricos.

Recordad que, si no os funciona, debéis de revisar y configurar las reglas de INPUT en “grupos de seguridad” para permitir todo el tráfico entrante a esta máquina:



Reglas de entrada

Ahora puede comprobar la conectividad de red con Reachability Analyzer

Ejecutar Reachability Analyzer

Reglas de entrada (1/1)

Administrar etiquetas Editar reglas de entrada

<input checked="" type="checkbox"/>	Name	ID de la regla del g...	Versión de IP	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción
<input checked="" type="checkbox"/>	-	sgr-0ef8c2d3ca7019a64	IPv4	SSH	TCP	22	0.0.0.0/0	-

Reglas de entrada

ID de la regla del grupo de seguridad

Tipo

Protocolo

Intervalo de puertos

Origen

Descripción: opcional

sg-0ef8c2d3ca7019a64

SSH

TCP

22

Personalizada

Q

Eliminar

-

Todo el tráfico

Todo

Anywhere-IP...

Q

0.0.0.0/0

Eliminar

Agregar regla

Cancelar Previsualizar los cambios Guardar reglas

5º Configuración de vuestro sitio web:

Probamos a resolver el nombre del servidor web dónde publicaremos nuestro sitio web. ¡Se debe mostrar el HTML por defecto de **nginx**!

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

En la imagen anterior, veis que resuelve el cliente sin problemas el nombre de la máquina y nos muestra la página de nginx por defecto. **Poned HTTP, no HTTPS.**

Pasamos a empezar a configurar **NGINX**. El directorio es `/etc/nginx/` lo listo para ver los ficheros que tenemos dentro:

```
root@blanca:/etc# cd nginx
root@blanca:/etc/nginx# ls
conf.d          koi-utf      modules-available  proxy_params  sites-enabled  win-utf
fastcgi.conf    koi-win      modules-enabled    scgi_params   snippets
fastcgi_params  mime.types   nginx.conf         sites-available uwsgi_params
root@blanca:/etc/nginx# _
```

- **sites-available**: tenemos los sitios web que vayamos creando, estén o no los sitios habilitados en ese momento.
- **sites-enabled**: sería la carpeta que contiene todos los sitios habilitados y que tienen enlaces simbólicos a los documentroot de los sitios publicados.

```
root@blanca:/etc/nginx# cd sites-enabled/
root@blanca:/etc/nginx/sites-enabled# ls
default
root@blanca:/etc/nginx/sites-enabled# ls -l
total 0
lrwxrwxrwx 1 root root 34 feb 16 18:58 default -> /etc/nginx/sites-available/default
root@blanca:/etc/nginx/sites-enabled# _
```

```
# available underneath a path with that package name, such as /etc/nginx
#
# Please see /usr/share/doc/nginx-doc/examples/ for more details
##

# Default server configuration
#
server {
    listen 80 default_server; PUERTO DE ESCUCHA, 80
    listen [::]:80 default_server;

    # SSL configuration
```

2ª DAW – Despliegue de aplicaciones web

```
# Don't use them in a production server!
#
# include snippets/snakeoil.conf;

root /var/www/html; DOCUMENTROOT

# Add index.php to the list if you are using PHP
index index.html index.htm index.nginx-debian.html;

server_name _; AQUÍ EL NOMBRE DEL DOMINIO

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ =404;
}

# pass PHP scripts to FastCGI server
#
```

Vamos a crear estructuras de directorio para el nuevo sitio web que vamos a crear en este servicio. Nos posicionamos en `/var/www` y creamos un nuevo directorio:

`mkdir despliegueweb`

```
root@ip-172-31-25-249:/var/www# mkdir despliegueweb
root@ip-172-31-25-249:/var/www# ls
despliegueweb  html
root@ip-172-31-25-249:/var/www#
```

Después, hay que añadir nuestro nuevo directorio de despliegue de la aplicación web en el fichero del directorio `sites-available`. Moveros al directorio:

`cd / etc/nginx`

Para ello, primero hacemos una copia del fichero `default` y con el fichero copiado empezamos a editar:

`cp default despliegueweb`

Editamos nuestro fichero de `despliegueweb`. Primero quitamos los valores por default asociados al puerto 80. Bajamos hasta el `documentroot` y ponemos la ruta de nuestro sitio, así como nuestro dominio:

2ª DAW – Despliegue de aplicaciones web



```
server {  
    listen 80; QUITAMOS LOS VALORES DEFAULT  
    listen [::]:80  
  
    # SSL configuration  
    #  
    # listen 443 ssl default_server;  
    #     en [::]:443 ssl default_server;  
    #  
    # Note: You should disable gzip for SSL traffic.  
    # See: https://bugs.debian.org/773332  
    #  
    # Read up on ssl_ciphers to ensure a secure configuration.  
    # See: https://bugs.debian.org/765782  
    #  
    # Self signed certs generated by the ssl-cert package  
    # Don't use them in a production server!  
    #  
    # include snippets/snakeoil.conf;  
  
    root /var/www/html/despliegueweb PONEMOS LA RUTA DE NUESTRO SITIO WEB  
  
    # Add index.php to the list if you are using PHP  
    index index.html index.htm index.nginx-debian.html;  
  
    server_name _ blanca.despliegueweb.es AÑADIMOS LOS DOMINIOS  
  
    location / {
```

Tenemos que crear ahora el enlace simbólico para que cualquier cambio que hagamos en el directorio de publicación de nuestro sitio se quede publicado.

ln -s /etc/nginx/sites-available/despliegueweb.

```
root@ip-172-31-18-1:/etc/nginx/sites-enabled# ln -s /etc/nginx/sites-available/serviciosenred  
root@ip-172-31-18-1:/etc/nginx/sites-enabled# ls -l  
total 0  
lrwxrwxrwx 1 root root 34 Feb 17 12:07 default -> /etc/nginx/sites-available/default  
lrwxrwxrwx 1 root root 41 Feb 17 17:21 serviciosenred -> /etc/nginx/sites-available/serviciosenred  
root@ip-172-31-18-1:/etc/nginx/sites-enabled#
```

ls -l

Una vez hecho esto, podemos borrar el sitio por defecto con rm default (tanto en sites-availables como en sites-enabled). Y testeamos los cambios con el comando:

nginx -t

```
root@ip-172-31-18-1:/etc/nginx/sites-available# nginx -t  
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok  
nginx: configuration file /etc/nginx/nginx.conf test is successful  
root@ip-172-31-18-1:/etc/nginx/sites-available#
```

Si no nos da errores, la revisión sintáctica de nuestros ficheros está ok.

Pasamos a crear nuestras páginas HTML:

Debéis hacer un restart o reload y comprobar que se cogen los cambios:

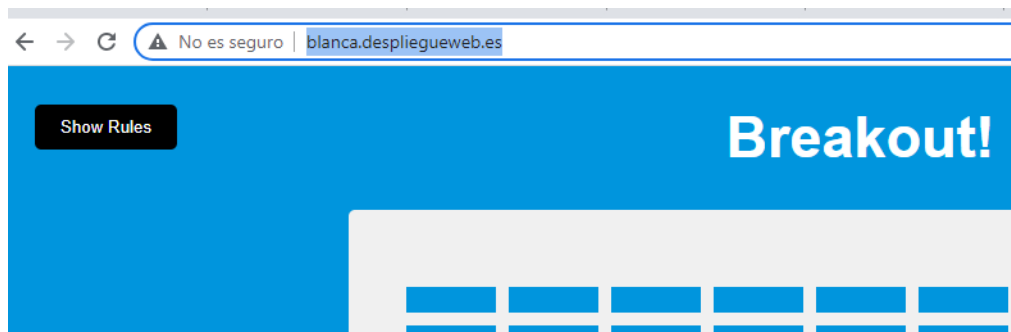
systemctl reload nginx

2ª DAW – Despliegue de aplicaciones web



```
oot@ip-172-31-18-1:/var/www/serviciosenred# systemctl reload nginx
oot@ip-172-31-18-1:/var/www/serviciosenred# systemctl status nginx
nginx.service - A high performance web server and a reverse proxy server
Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor pres
Active: active (running) since Thu 2022-02-17 17:09:14 UTC; 17min ago
Docs: man:nginx(8)
Process: 1845 ExecReload=/usr/sbin/nginx -g daemon on; master_process on
Main PID: 893 (nginx)
Tasks: 2 (limit: 1140)
```

Cargamos sitio web dónde se visualizará lo que hayáis desplegado vosotros:



6º Instalar certificado en servicio web:

Vamos a seguir los pasos de este sitio para dejar instalado un certificado en nuestro servidor web:

<https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-20-04-es>

1º Realizamos creación e instalación de los certificados:

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

```
update-initramfs: Generating /boot/initrd.img-5.4.0-100-aws
root@ip-172-31-18-1:/var/www/serviciosenred# apt install certbot python3-certbot-nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
certbot python3-certbot-nginx is already installed.
The following additional packages will be installed:
  libpython-stdlib libpython2.7-minimal libpython2.7-stdlib python python-minimal python-pyicu python2.7
  python2.7-minimal python3-acme python3-certbot python3-configargparse python3-future python3-josepy python3-lib
  python3-mock python3-parsedatetime python3-pbr python3-pyparsing python3-requests-toolbelt python3-rfc3339
  python3-tz python3-zope.component python3-zope.event python3-zope.hookable
Suggested packages:
  python3-certbot-apache python3-certbot-doc python-doc python-tk python2.7-doc binutils binfmt-support python-acm
  python-certbot-nginx-doc python-future-doc python-mock-doc python-pyparsing-doc
The following NEW packages will be installed:
  certbot libpython-stdlib libpython2.7-minimal libpython2.7-stdlib python python-minimal python-pyicu python2.7
  python2.7-minimal python3-acme python3-certbot python3-configargparse python3-future python3-josepy python3-lib
  python3-mock python3-parsedatetime python3-pbr python3-pyparsing python3-requests-toolbelt python3-rfc3339
  python3-tz python3-zope.component python3-zope.event python3-zope.hookable
```

2º Aplicamos y comprobamos que está levantado el servidor:

2ª DAW – Despliegue de aplicaciones web

```
reated symlink /etc/systemd/system/timers.target.wants/certbot.timer -> /etc/systemd/system/timers.target.wants/certbot.timer
Setting up python3-certbot-nginx (0.23.0-1) ...
Processing triggers for mime-support (3.60ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
root@ip-172-31-18-1:/var/www/serviciosenred# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@ip-172-31-18-1:/var/www/serviciosenred# systemctl status nginx
nginx.service - A high performance web server and a reverse proxy server
Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor prese
Active: active (running) since Thu 2022-02-17 17:09:14 UTC; 22min ago
Docs: man:nginx(8)
Main PID: 893 (nginx)
```

3º Aplicamos los certificados creados:

```
sudo certbot --nginx -d serviciosenred.website -d www.serviciosenred.website
```

```
Feb 17 17:09:14 ip-172-31-18-1 systemd[1]: Started A high performance web server and a reverse proxy server.
root@ip-172-31-18-1:/var/www/serviciosenred# sudo certbot --nginx -d serviciosenred.website -d www.serviciosenred.website
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator nginx, Installer nginx
Enter email address (used for urgent renewal and security notices) (Enter 'c' to
cancel): bpj154@hotmail.com

-----
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf. You must
agree in order to register with the ACME server at
https://acme-v01.api.letsencrypt.org/directory
-----
(A)gree/(C)ancel: a

-----
Would you be willing to share your email address with the Electronic Frontier
Foundation, a founding partner of the Let's Encrypt project and the non-profit
organization that develops Certbot? We'd like to send you email about our work
encrypting the web, EFF news, campaigns, and ways to support digital freedom.
-----
(Y)es/(N)o: n
Obtaining a new certificate
Performing the following challenges:
http-01 challenge for serviciosenred.website
http-01 challenge for www.serviciosenred.website
Waiting for verification...
Cleaning up challenges
Deploying Certificate to VirtualHost /etc/nginx/sites-enabled/serviciosenred
Deploying Certificate to VirtualHost /etc/nginx/sites-enabled/default
```

Si es la primera vez que ejecuta certbot, se le pedirá que ingrese una dirección de correo electrónico y que acepte las condiciones de servicio. Después de esto, certbot se comunicará con el servidor de Let's Encrypt y realizará una comprobación a fin de verificar que usted controle el dominio para el cual solicite un certificado. Si no tenemos el dominio realmente comprado, poned el email de alu.murciaeduca. Esto solo funciona con un dominio público real, si se realiza la práctica con uno ficticio, falla la generación de certificados.

Cuando termina con éxito, nos muestra los certificados alojados en nuestro servidor y su fecha de expiración (los certificados se deben de ir renovando):

2ª DAW – Despliegue de aplicaciones web

```

-----
Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 2
Redirecting all traffic on port 80 to ssl in /etc/nginx/sites-enabled/serviciosenred
Redirecting all traffic on port 80 to ssl in /etc/nginx/sites-enabled/default
-----

Congratulations! You have successfully enabled https://serviciosenred.website
and https://www.serviciosenred.website

You should test your configuration at:
https://www.ssllabs.com/ssltest/analyze.html?d=serviciosenred.website
https://www.ssllabs.com/ssltest/analyze.html?d=www.serviciosenred.website
-----

IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at:
   /etc/letsencrypt/live/serviciosenred.website/fullchain.pem

```

4º Verificar la renovación automática de Certbot:

sudo systemctl status certbot.timer


```

root@ip-172-31-18-1:/var/www/serviciosenred# systemctl status certbot.timer
● certbot.timer - Run certbot twice daily
   Loaded: loaded (/lib/systemd/system/certbot.timer; enabled; vendor preset: enabled)
   Active: active (waiting) since Thu 2022-02-17 17:30:25 UTC; 7min ago
   Trigger: Fri 2022-02-18 06:32:18 UTC; 12h left

Feb 17 17:30:25 ip-172-31-18-1 systemd[1]: Started Run certbot twice daily.
root@ip-172-31-18-1:/var/www/serviciosenred#

```

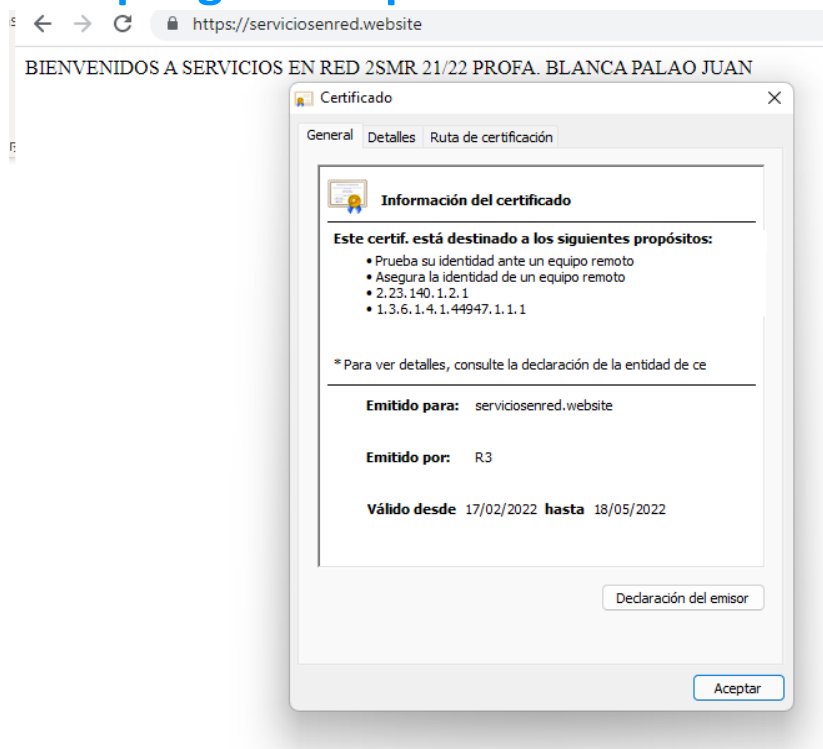
5º Navegamos con **HTTPS** y podemos comprobar que ya tenemos candado y navegación segura a nuestro servidor web:

← → ↻  https://serviciosenred.website

BIENVENIDOS A SERVICIOS EN RED 2SMR 21/22 PROFA. BLANCA PALAO JUAN

Podéis revisar el certificado:

2ª DAW – Despliegue de aplicaciones web



PRACTICA A REALIZAR en AWS

Tienes que configurar un servidor web sencillo con nginx creando una instancia de Ubuntu server 22 en AWS Academy con las siguientes características. Asígnale una IP elástica y realizad la configuración del DNS en vuestro dominio adquirido. Si no tenéis, me pasáis la IP elástica que habéis asignado al foro:

<https://aulavirtual.murciaeduca.es/mod/forum/view.php?id=3328971>

En el fichero de configuración de sitios que vas a crear nuevo, debes de poner el nombre de tu nombre FQDN de servidor web, por ejemplo www.nombreAlumno.despiliegueweb.es, el dominio que tengas yo creado en el DNS de Nominalia o en freenom.

1. DocumentRoot “/var/www/<tunombre>
2. DirectoryIndex index.html (debes crear tú un archivo de inicio que contenga tu nombre completo mínimo). Podéis desplegar otros componentes.
3. Comprueba que funciona utilizando cualquier navegador web con acceso a Internet por **HTTP**.
4. Crea e instala para tu sitio www.<tudominio>.es los certificados.
5. Accede al sitio ahora con **HTTPS** utilizando cualquier navegador web con acceso a Internet.

MATERIAL A ENTREGAR

Como siempre, se realiza la práctica y después se graba un vídeo explicativo.

2ª DAW – Despliegue de aplicaciones web

Criterios de calificación



No hecho (0)	Nivel bajo (2'5)	Nivel medio (6'5)	Nivel alto (hasta 10)
El alumno o alumna realiza un vídeo copiado sobre nginx o trabajo no entregado.	El vídeo no contiene toda la información y apartados nginx y/o no cumple con los requisitos de originalidad y contenido pedidos.	El vídeo contiene toda la información y cumple con los requisitos <u>mínimos</u> de originalidad y contenido pedidos.	El vídeo contiene toda la información y cumple ampliamente con los requisitos de originalidad y contenido pedidos. Además, organiza incluye información extra y la organización de la información está trabajada.