

El proyecto consta de la creación de un servidor web utilizando Nginx.

Comenzamos con la creación de la estructura de las carpetas necesarias:

/conf

Contendrá conf con el archivo .conf

Lugar donde se colocan los archivos de configuración adicionales que permiten agregar configuraciones específicas de sitios o ajustes personalizados. En esta carpeta, los archivos suelen tener la extensión .conf.

El archivo default en sites-available es el archivo de configuración predeterminado de Nginx, contendrá:

- **listen 80 default_server;** Configura al servidor para escuchar en el puerto 80 como servidor predeterminado.
- **root /var/www/html;** Especifica la carpeta raíz desde la cual Nginx servirá los archivos.
- **index:** Define los archivos por defecto que se mostrarán si se accede al directorio (ej. index.html).
- **server_name _;** El guion bajo (_) indica que este es el servidor por defecto si no hay coincidencias específicas para el nombre de host.
- **location /:** Define las reglas para manejar las solicitudes en el directorio raíz.
- **try_files \$uri \$uri/ =404;** Intenta servir el archivo solicitado; si no existe, devuelve un error 404.

Copiaremos ambos en sus correspondientes carpetas.

La carpeta sites-availables debe incluir un archivo .conf por cada dominio que vayamos a crear, en este caso son: christian.com.conf, gregorioasensio.com.conf y seguro.net.conf

Los archivos se muestran de la siguiente manera:

christian.com.conf:

```

1  # Virtual Host configuration for example.com
2  #
3  # You can move that to a different file under sites-available/ and symlink that
4  # to sites-enabled/ to enable it.
5  #
6  server {
7      listen 80; # Puerto por defecto para HTTP
8      listen [::]:80; # Puerto por defecto para HTTP
9
10     server_name christian.com; # Nombre del dominio
11
12     root /var/www/html/christian.com; # Ruta de la carpeta raíz del dominio
13     index index.html; # Archivo por defecto
14
15     location / {
16         try_files $uri $uri/ =404; # Intenta servir el archivo solicitado, si no existe, muestra un error 404
17     }
18
19     #Definimos ERROR 404
20     error_page 404 /404.html;
21     location = /404.html {
22         root /var/www/html/christian.com/errors; # Archivos de errores
23         internal;
24     }
25 }

```

gregorioasensio.com.conf

```

1  # Virtual Host configuration for example.com
2  #
3  # You can move that to a different file under sites-available/ and symlink that
4  # to sites-enabled/ to enable it.
5  #
6  server {
7      listen 80; # Puerto por defecto para HTTP
8      listen [::]:80; # Puerto por defecto para HTTP
9
10     server_name gregorioasensio.com; # Nombre del dominio
11
12     root /var/www/html/gregorioasensio.com; # Ruta de la carpeta raíz del dominio
13     index index.html; # Archivo por defecto
14
15     location / {
16         try_files $uri $uri/ =404; # Intenta servir el archivo solicitado, si no existe, muestra un error 404
17     }
18
19     #Definimos ERROR 404
20     error_page 404 /404.html;
21     location = /404.html {
22         root /var/www/html/gregorioasensio.com/errors; # Archivos de errores
23         internal;
24     }
25 }

```

seguro.net.conf

```

1  server {
2      listen 80;
3      listen [::]:80;
4
5      server_name seguro.net;
6
7      # Redirigir todo el tráfico HTTP a HTTPS
8      return 301 https://$host$request_uri;
9  }
10
11
12  server {
13      listen 443 ssl;
14      server_name seguro.net www.seguro.net;
15
16      ssl_certificate /etc/nginx/certs/seguro.crt;
17      ssl_certificate_key /etc/nginx/certs/seguro.key;
18
19      # Configuraciones SSL adicionales
20      ssl_protocols TLSv1.2 TLSv1.3;
21      ssl_ciphers 'EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH';
22      ssl_prefer_server_ciphers on;
23
24      # Ruta al directorio raíz del sitio
25      root /var/www/html/seguro.net;
26      index index.html;
27
28      location / {
29          try_files $uri $uri/ =404; # Configuración adicional
30      }
31
32      error_page 401 /401.html;
33      location = /401.html {
34          root /var/www/html/seguro.net/errors; # Carpeta de archivos de error
35          internal;
36      }
37
38      error_page 403 /403.html;
39      location = /403.html {
40          root /var/www/html/seguro.net/errors;
41          internal;
42      }
43
44      error_page 404 /404.html;
45      location = /404.html {
46          root /var/www/html/seguro.net/errors;
47          internal;
48      }

```

```

50     error_page 500 /500.html;
51     location = /500.html {
52         root /var/www/html/seguro.net/errors;
53         internal;
54     }
55
56     location /privado {
57         auth_basic "Acceso Restringido";
58         auth_basic_user_file /etc/nginx/.htpasswd;
59     }
60 }

```

Nuestras páginas web deberán encontrarse dentro de la carpeta website, que a su vez contienen las carpetas de cada dominio.

Incluiremos un index.html por defecto en /website

Y cada web que queremos crear irá dentro de su dominio, incluyendo la carpeta /errors con los errores que queremos mostrar.

Dominio seguro SSL

A diferencia de los dominios normales, al definir el dominio privado SSL debemos tener en cuenta otros directorios dentro de él:

Privado: aquello que queremos mostrar solo una vez autenticados dentro del webhost.

Carpeta htpasswd (Se puede generar más tarde).

Cómo generarlo:

Instalación si no lo tenemos por defecto: `apt update && apt install apache2-utils`

Posteriormente lo generaremos con: `htpasswd -c /etc/nginx/.htpasswd nombre_usuario`

Carpeta certs (donde más tarde se añadirá el certificado y key al crearlo mediante openssl desde la terminal.)

Scripts

Definiremos los comandos que el contenedor hará automáticamente al ejecutarse. Se coloca en el directorio principal en la carpeta scripts, está definido como `entrypoint.sh` y contiene:

```

Users > christian > nginx-web > scripts > $ entrypoint.sh
1  #!/bin/bash
2
3  # Habilitar los sitios si no existen los enlaces simbólicos
4  if [ ! -L /etc/nginx/sites-enabled/dominio-one.com.conf ]; then
5  | ln -s /etc/nginx/sites-available/dominio-one.com.conf /etc/nginx/sites-enabled/
6  fi
7
8  if [ ! -L /etc/nginx/sites-enabled/dominio-two.com.conf ]; then
9  | ln -s /etc/nginx/sites-available/dominio-two.com.conf /etc/nginx/sites-enabled/
10 fi
11
12 # Recargar la configuración de Nginx
13 nginx -s reload
14
15 # Iniciar Nginx en primer plano
16 nginx -g 'daemon off;'

```

Creación del docker-compose.yml:

Tendrá la siguiente estructura y volúmenes:

```

Users > christian > nginx-web > 📄 docker-compose.yml
1  services:
2    web:
3      image: ubuntu/nginx
4      container_name: nginx_server
5      ports:
6        - "80:80"
7        - "443:443"
8      volumes:
9        - ./sites-available:/etc/nginx/sites-available
10       - ./website:/var/www/html/
11       - ./scripts:/docker-entrypoint.d/
12       - ./htpasswd/.htpasswd:/etc/nginx/.htpasswd
13       - ./certs:/etc/nginx/certs
14     restart: always

```

Generación de certificados:

Una vez instalado openssl en nuestro equipo, mediante el siguiente comando generaremos los certificados que más tarde añadiremos a certs:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout seguro.key -out seguro.crt
```

Hosts (en nuestro equipo):

Mediante la terminal añadiremos al archivo /etc/hosts las entradas:

```
127.0.0.1 christian.com
```

127.0.0.1 gregorioasensio.com
127.0.0.1 www.gregorioasensio.com
127.0.0.1 seguro.net
127.0.0.1 www.seguro.net

Accedemos con el comando:

```
sudo nano /etc/hosts
```

Y guardamos los cambios presionando Ctrl + O y Enter para confirmar.

Comprobamos los cambios con:

```
cat /etc/hosts
```

Docker: Ejecución del servidor definitivo con nginx.

Ejecución de nuestro Docker-compose mediante la terminal:

```
docker-compose --build -d
```

Una vez corriendo, podemos usar la terminal de nuestro contenedor mediante el comando:

```
docker exec -it nginx_server /bin/bash
```