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

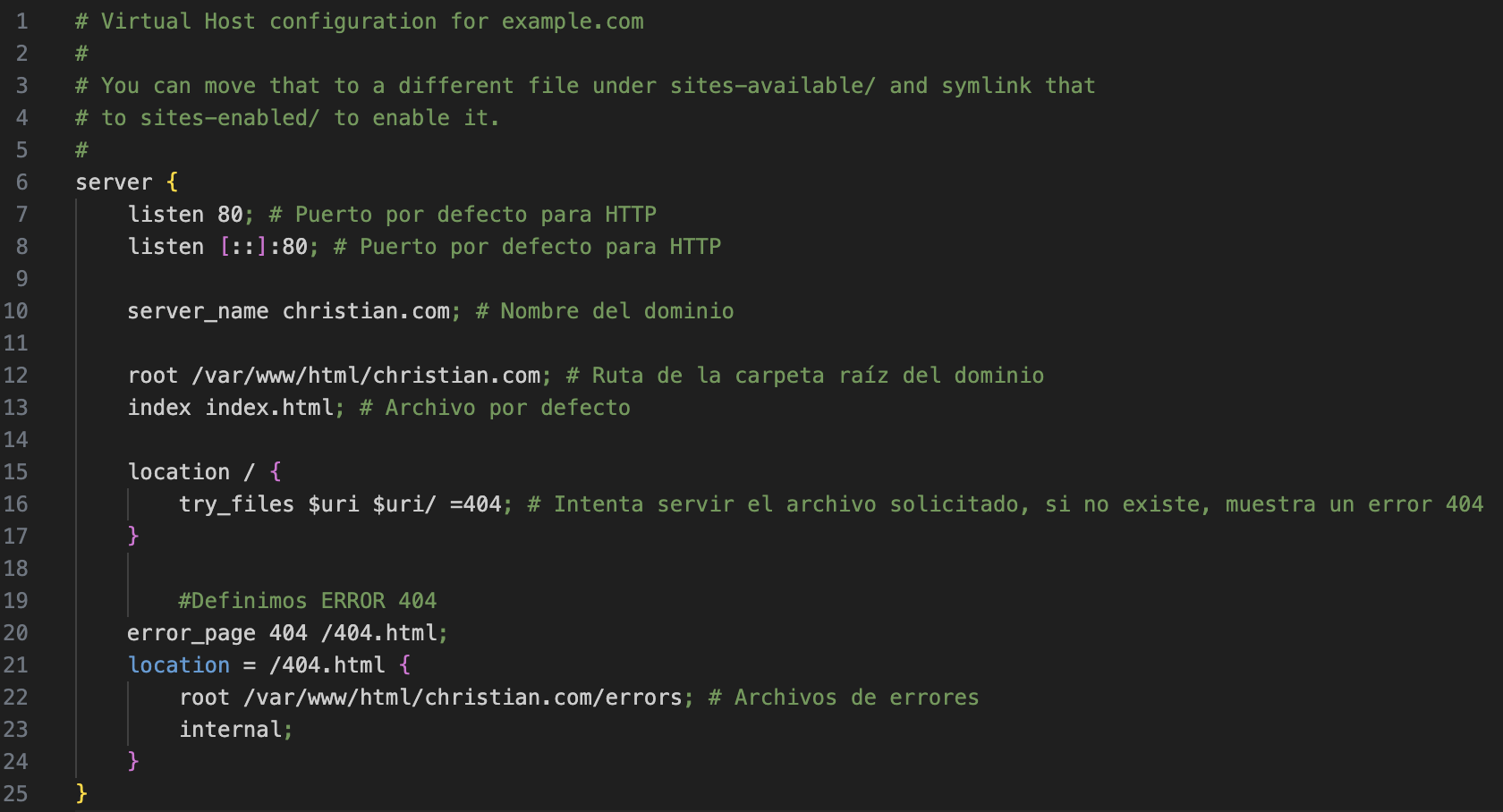
Comenzamos con la creación de la estructura de las carpetas necesarias:  
/conf  
Contendrá conf con el archivo .conf   
Lugar donde se colocan los archivos de configuración adicionales que permiten agregar configuraciones específicas de sitios o ajustes personalizados. En esta carpeta, los archivos 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:



gregorioasensio.com.conf

Texto

Descripción generada automáticamente

seguro.net.conf

Texto

Descripción generada automáticamente  
Texto

Descripción generada automáticamente

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 htcpasswd (Se puede generar más tarde).

Cómo generarlo:

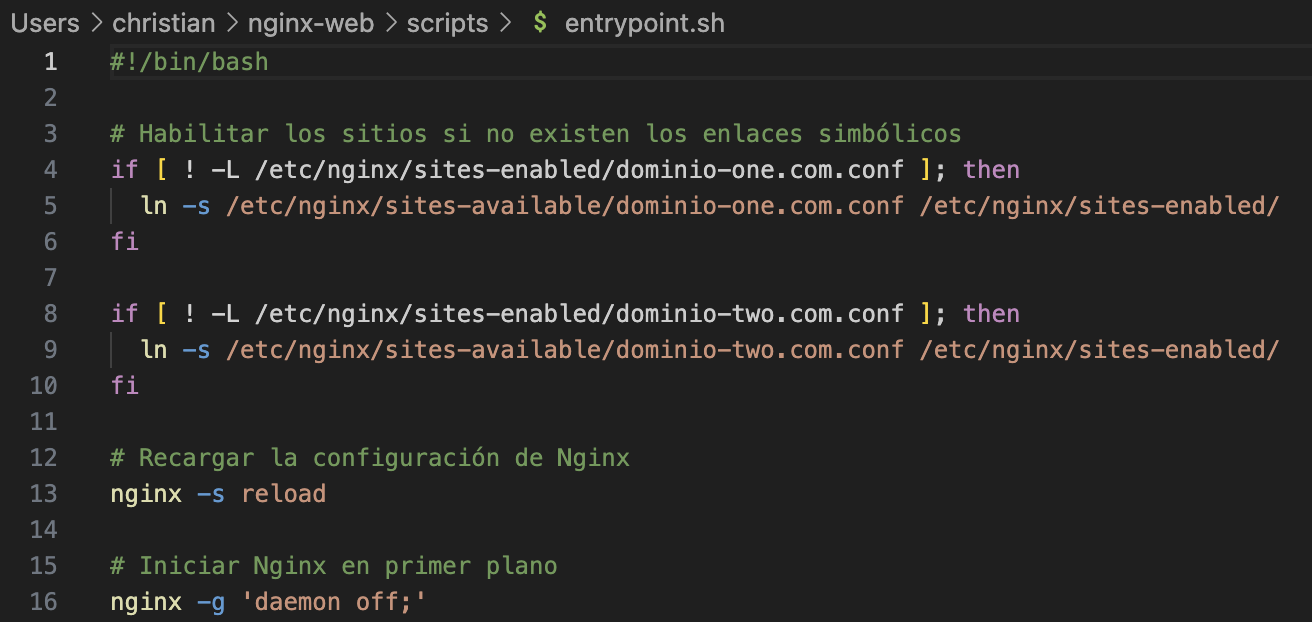
Instalación si no lo tenemos por defecto: apt update && apt install apache2-utils

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:



**Creación del docker-compose.yml:**

Tendrá la siguiente estructura y volúmenes:

Texto

Descripción generada automáticamente

**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](http://www.seguro.net)

Accedemos con el commando:

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