

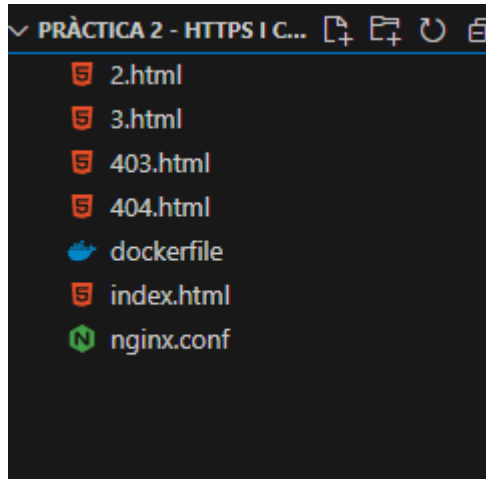
Pràctica 2 - HTTPS i codis d'estat HTTP



Carlos Jiménez Banús

Pas 1

Creació Fitxers HTML Creació fitxers HTML per cada resposta Crearem els fitxers HTML per la gestió dels estats d'HTTP. Voldrem gestionar els 403 i 404, ja que són els que bàsicament veurà el client. També crearem el fitxer index.html i dues pàgines més de la nostra web les quals mostrarà clarament quina temàtica hem escollit i l'usuari tindrà opció de canviar d'una de les pàgines a l'altra. Es valorarà l'estètica dels fitxers. A cada pàgina ha d'aparèixer el vostre nom.



Pas 2

Creació imatge nginx personalitzada Un cop tenim els fitxers HTML llestos, crearem un Dockerfile que contingui:

- Imatge del servidor.
- Còpia dels fitxers HTML.
- Instal·lació nano.
- Etiqueta de l'autor del contenidor.

```
FROM nginx:latest

LABEL maintainer="Carlos Jimenez"

COPY index.html /usr/share/nginx/html/
COPY 2.html /usr/share/nginx/html/
COPY 3.html /usr/share/nginx/html/
COPY 403.html /usr/share/nginx/html/
COPY 404.html /usr/share/nginx/html/

RUN apt-get update && \
    apt-get install -y nano && \
    rm -rf /var/lib/apt/lists/*

EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]
```

Creació de la imatge per terminal:

```
jfcar@DESKTOP-77JCNGS MINGW64 ~/OneDrive/Documentos/Carlos Jimenez/2wiaw/M08/Pràctica 2 - HTTPS i codis d'estat HTTP (main)
$ docker build -t server-nginx .
#0 building with "default" instance using docker driver

#1 [internal] load .dockerignore
#1 transferring context: 2B done
#1 DONE 0.0s

#2 [internal] load build definition from dockerfile
#2 transferring dockerfile: 428B done
#2 DONE 0.0s

#3 [internal] load metadata for docker.io/library/nginx:latest
#3 ...

#4 [auth] library/nginx:pull token for registry-1.docker.io
#4 DONE 0.0s

#3 [internal] load metadata for docker.io/library/nginx:latest
#3 DONE 1.4s

#5 [1/7] FROM docker.io/library/nginx:latest@sha256:10d1f5b58f74683ad34eb29287e07dab1e90f10af243f151bb50aa5dbb4d62ee
#5 DONE 0.0s

#6 [internal] load build context
#6 transferring context: 140B done
#6 DONE 0.0s

#7 [6/7] COPY 404.html /usr/share/nginx/html/
#7 CACHED

#8 [4/7] COPY 3.html /usr/share/nginx/html/
#8 CACHED

#9 [5/7] COPY 403.html /usr/share/nginx/html/
#9 CACHED

#10 [2/7] COPY index.html /usr/share/nginx/html/
#10 CACHED

#11 [3/7] COPY 2.html /usr/share/nginx/html/
#11 CACHED

#12 [7/7] RUN apt-get update && apt-get install -y nano && rm -rf /var/lib/apt/lists/*
#12 CACHED

#13 exporting to image
#13 exporting layers done
#13 writing image sha256:672c37df55444ecdac1d29637b9ee5817d0dd87aadf1df5fae80533aa8a6ed40 done
#13 naming to docker.io/library/server-nginx done
#13 DONE 0.0s

What's Next?
View a summary of image vulnerabilities and recommendations - docker scout quickview
```

Pas 3

Edició nginx.conf i creació certificats Iniciarem un contenidor de la imatge creada, crearem els certificats utilitzant un xifratge AES-256 o RSA-2048 (mostra com l'has creat) i configurarem el fitxer nginx.conf segons les nostres necessitats. El port 80 haurà de redirigir al 443. Tota l'edició del fitxer nginx.conf i la creació del certificat ho farem des del contenidor. Mostrarem com accedim, naveguem i fem els diferents apartats. Podem per exemple mostrar els diferents passos que anem fent per resoldre l'activitat.

Creació de docker amb la imatge anterior:

```
jfcar@DESKTOP-77JCNGS MINGW64 ~/OneDrive/Documentos/Carlos Jimenez/2wiaw/M08/Pràctica 2 - HTTPS i codis d'estat HTTP (main)
$ docker run -d --name nginx-server -p 80:80 -p 443:443 server-nginx:latest
863a9c90ff695a941af5Fe0ba1aa3058ebc4e6e7d44108400777490e15f8e879
```

Entrem a la terminal del docker:

```
jfcar@DESKTOP-77JCNGS MINGW64 ~/OneDrive/Documentos/Carlos Jimenez/2wiaw/M08/Pràctica 2 - HTTPS i codis d'estat HTTP (main)
$ docker exec -it nginx-server /bin/bash
```

Creació de directoris on es guarden els certificats amb el comandament mkdir /etc/nginx/certificados

```
root@a5a939d33b99:/# mkdir /etc/nginx/certificados
root@a5a939d33b99:/# ls /etc/nginx/
certificados  conf.d  fastcgi_params  mime.types  modules  nginx.conf  scgi_params  uwsgi_params
```

Comanda per genera la clau privada “openssl genrsa -out /etc/nginx/certificados/clave.key 2048”

```
root@a5a939d33b99:/# openssl genrsa -out /etc/nginx/certificados/clave.key 2048
```

Comanda per genera el certificat autofirmat “openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/nginx/certificados/clave.key -out /etc/nginx/certificados/certificado.crt”

[illegible]

Editem el fitxer nginx.conf

```
events {
    worker_connections 1024;
}

# Config server
http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    server {
        listen      80;
        server_name localhost;

        location / {
            return 301 https://$host$request_uri;
        }
    }

    # Server HTTPS
    server {
        listen      433 ssl;
        server_name localhost;

        ssl_certificate /etc/nginx/certificados/certificado.crt;
        ssl_certificate_key /etc/nginx/certificados/certificado.crt;

        location / {
            root      /usr/share/nginx/html;
            index      index.html index.htm;
        }

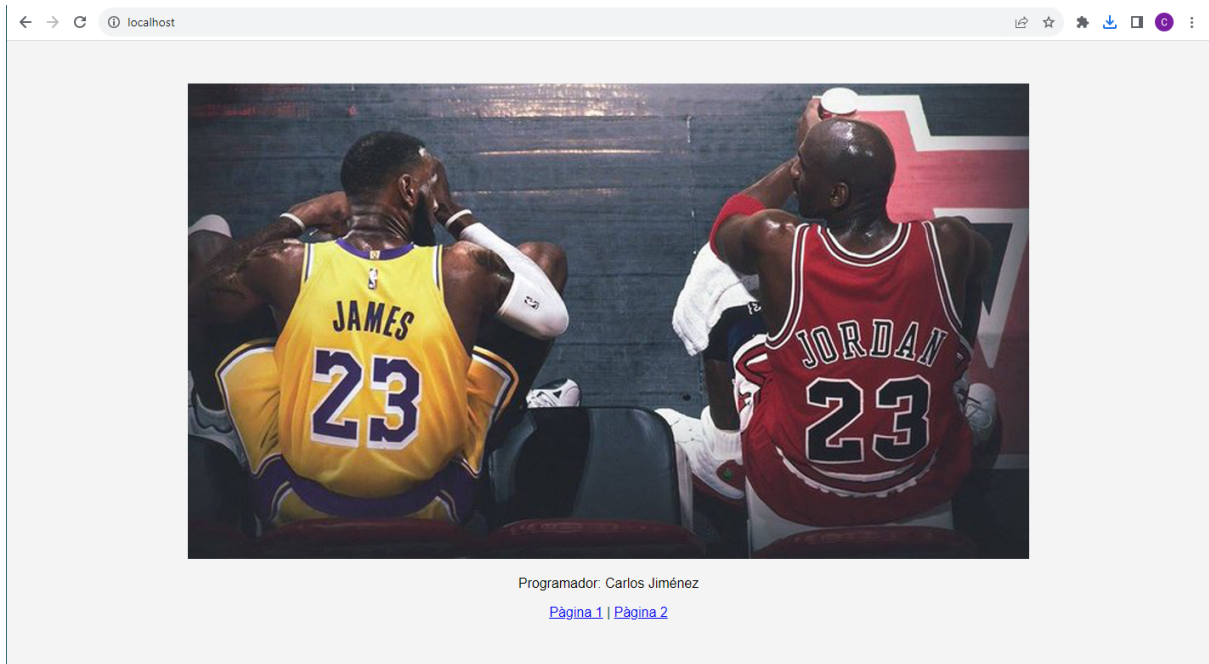
        # Config error 404
        error_page 404 /404.html
            location = /404.html {
                root /usr/share/nginx/html;
                internal;
            }

        # Config error 403
        error_page 403 /403.html
            location = /403.html {
                root /usr/share/nginx/html;
                internal;
            }
    }
}
```

Pas 4

Comprovacions codis d'estat Mostrarem captures de pantalla de com el servidor ens retorna els fitxers que hem creat en funció de l'error comés. Així com un vídeo de navegació pel lloc web.

```
jfcar@DESKTOP-77JCNGS MINGW64 ~/OneDrive/Documentos/Carlos Jimenez/2wiaw/M08/Pràctica 2 - HTTPS i codis d'estat HTTP (main)
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
863a9c90ff69   server-nginx:latest  "/docker-entrypoint..."  37 minutes ago  Up 37 minutes  0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp  nginx-server
```



docker image ls

```
jfcarr@DESKTOP-77JCN64 MINGW64 ~/OneDrive/Documentos/Carlos Jimenez/2w1aw/M08/Pràctica 2 - HTTPS i codis d'estat HTTP (main)
$ docker image ls
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
httpsserver          version1.0      672c37df5544   2 hours ago    189MB
server-nginx         latest         672c37df5544   2 hours ago    189MB
```

docker ps

```
jfcarr@DESKTOP-77JCN64 MINGW64 ~/OneDrive/Documentos/Carlos Jimenez/2w1aw/M08/Pràctica 2 - HTTPS i codis d'estat HTTP (main)
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
863a9c90ff69   server-nginx:latest   "/docker-entrypoint..." 57 minutes ago Up 57 minutes 0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp nginx-server
```

Conclusions del projecte:

Què has après?

Durant el desenvolupament d'aquest projecte, he après com crear un entorn Docker per a un servidor web utilitzant Nginx, com gestionar estats d'HTTP com el 403 i 404, i com configurar certificats SSL per habilitar la comunicació segura. També he après com escriure un Dockerfile per personalitzar una imatge Docker, copiar fitxers al seu interior, instal·lar paquets i gestionar les configuracions del servidor web.

Què t'ha costat més?

Una de les parts més desafiantes va ser l'ús de terminals en entorns dins del docker. També, l'ús de certificats SSL i la configuració d'Nginx per a una comunicació segura podrien resultar complexos, especialment per als principiants.

Com valors el treball?

Valoro positivament aquest treball ja que proporciona una bona introducció a les pràctiques de desenvolupament web i a la gestió de contenidors Docker. El projecte permet explorar aspectes rellevants com la configuració del servidor web, la gestió d'estats d'HTTP, l'ús de certificats SSL, i la creació de Dockerfiles. Tot i els desafiaments, aquesta experiència és valuosa per adquirir habilitats pràctiques en desenvolupament i desplegament de llocs web.