DOCKER - E2

1. Download and run version 2.4.54-alpine of the web server to access the Apache server via port 8080.

```
docker run -d -p 8080:80 httpd:2.4.54-alpine
```

```
root@christianms13:~# docker run -d -p 8080:80 httpd:2.4.54-alpine
Unable to find image 'httpd:2.4.54-alpine' locally
2.4.54-alpine: Pulling from library/httpd
8921db27df28: Already exists
f2a34d2799ed: Pull complete
f404b93686e0: Pull complete
0fd3f8de080b: Pull complete
78172ca4b2d9: Pull complete
1eb322ae2606: Pull complete
Digest: sha256:3e28d0745563a7033c5d6db21a8681ecf64bdc765354d86d9181be613cd8bf32
Status: Downloaded newer image for httpd:2.4.54-alpine
1ba9e139f715b5867fa3ddbccdeded0fa3ece003de8dfb945491afc2269d366f
root@christianms13:~# docker images
REPOSITORY   TAG             IMAGE ID       CREATED       SIZE
nginx        latest          a99a39d070bf   12 days ago   142MB
node         19-alpine       17299c0421ee   2 weeks ago   176MB
httpd        2.4.54-alpine   ce10bb9ec184   2 weeks ago   56.9MB
ubuntu       18.04           e28a50f651f9   3 weeks ago   63.1MB
root@christianms13:~# docker ps
CONTAINER ID   IMAGE                COMMAND              CREATED          STATUS          PORTS                   NAMES
1ba9e139f715   httpd:2.4.54-alpine   "httpd-foreground"   16 seconds ago   Up 14 seconds   0.0.0.0:8080->80/tcp   pedantic_mcclintock
```

```
docker rename pdantic_mcclintock webapache
```

```
root@christianms13:~# docker rename pedantic_mcclintock  webapache
```

2. Access the "webapache" terminal. Locate the default directory where the welcome file is saved. Create a directory called "backupweb" and make a copy of the default page in this directory.

```
docker exec -it webapache sh
```

```
root@christianms13:~# docker exec -it webapache sh
/usr/local/apache2 #
```

```
cd /usr/local/apache2/htdocs
```

```
ls -la
```

```
/ # cd usr/local/apache2/htdocs/
/usr/local/apache2/htdocs # ls -la
total 16
drwxr-xr-x    2 root     root          4096 Jan  9 18:26 .
drwxr-xr-x    1 www-data www-data      4096 Jan  9 18:26 ..
-rw-r--r--    1 504      dialout        45 Jun 11  2007 index.html
```

```
mkdir backupweb
```

```
cp index.html backupweb/index_copy.html
```

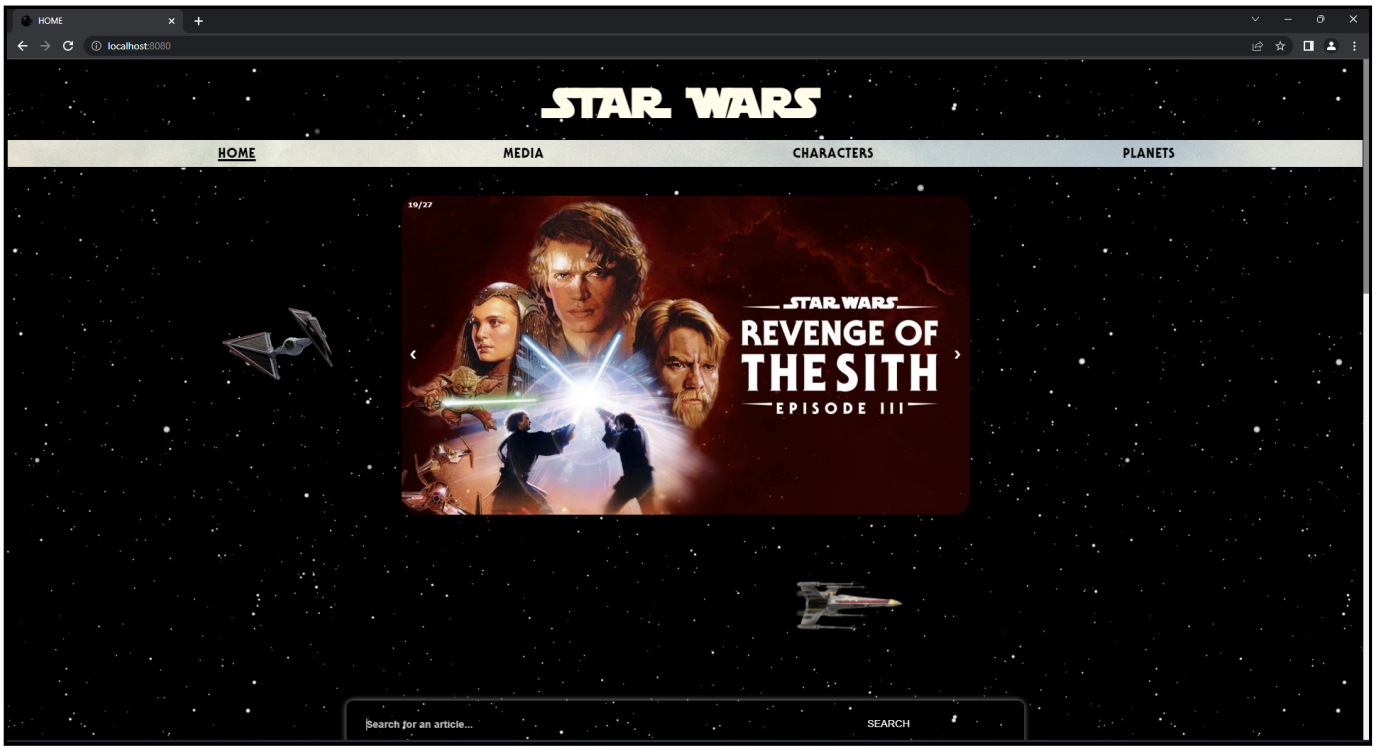
```
/usr/local/apache2/htdocs # mkdir backupweb
/usr/local/apache2/htdocs # cp index.html backupweb/index_copy.html
```

3. Exit the "webapache" docker terminal. Locate a complete static project where you have your html works, styles, etc... Copy all the content of this website in the "webapache" container so that we can view the content of our website on the apache server.

```
docker cp . webapache:/usr/local/apache2/htdocs
```

```
root@christianms13:/mnt/c/Users/chris/Documents/DAW/lenguajes_de_marcas/fin_1er_trimestre# docker cp . webapache:/usr/local/apache2/htdocs
```

4. To see if it works, stop the container and then start it again. Visit in a browser the address "localhost:8080".



5. Using the image pulled in exercise 1, create a new image using "Dockerfile" to create a container called "nuevaweb", which will be used as a web page, using port 8081.

Content in the "Dockerfile" file:

```
from httpd:2.4.54-alpine
copy . /usr/local/apache2/htdocs
expose 8081
```

```
root@christianms13:/mnt/c/Users/chris/Documents/DAW/lenguajes_de_marcas/fin_1er_trimestre# cat Dockerfile
from httpd:2.4.54-alpine
copy . /usr/local/apache2/htdocs
expose 8081
```

Building the image, this time I didn't specify the version:

```
docker build -t nuevaweb .
```

```
root@christianms13:/mnt/c/Users/chris/Documents/DAW/lenguajes_de_marcas/fin_1er_trimestre# docker build -t nuevaweb .
[+] Building 3.8s (7/7) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 113B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/httpd:2.4.54-alpine
 => [internal] load build context
 => => transferring context: 126.48MB
 => [1/2] FROM docker.io/library/httpd:2.4.54-alpine
 => [2/2] COPY . /usr/local/apache2/htdocs
 => exporting to image
 => => exporting layers
 => => writing image sha256:b38abece3c5141878c4da70e19e791edcff3d8751132ee16d41ca98db552035b
 => => naming to docker.io/library/nuevaweb

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```

```
docker images
```

```
root@christianms13:/mnt/c/Users/chris/Documents/DAW/lenguajes_de_marcas/fin_1er_trimestre# docker images
REPOSITORY      TAG             IMAGE ID       CREATED              SIZE
nuevaweb        latest          b38abece3c51   About a minute ago   183MB
web_node        v01             5b5dff62e488   11 hours ago         176MB
2smr/miwebdaw   v01             84750885cb61   19 hours ago         148MB
node            19-alpine       17299c0421ee   2 weeks ago          176MB
httpd           2.4.54-alpine   ce10bb9ec184   2 weeks ago          56.9MB
ubuntu          18.04           e28a50f651f9   3 weeks ago          63.1MB
```

An now creating the container from this image:

```
docker run -d --name nuevaweb -p 8081 nuevaweb
```

```
root@christianms13:/mnt/c/Users/chris/Documents/DAW/lenguajes_de_marcas/fin_1er_trimestre# docker run -d --name nuevaweb -p 8081 nuevaweb
e2fdfa7802a4ac2c1d34d7daa38a25796df390b045d72d465d964474400afbe5
```

And now it can be visited from a web browser.

6. Modify the web page content. What do you need to do?

To modify the web page, you must access the container where said page is hosted and modify the code from there.

7. To avoid having to rebuild the container on each page source code change web, use a volume that automatically synchronizes the web directory of our host with container web directory. Use the following statement: "docker run -d -p 80:80 -v host_directory:webapache_container_directory".

The command "docker run -d -p 80:80 -v host_directory:webapache_container_directory" will run a container in detached mode (-d) and map port 80 of the host to port 80 of the container (-p 80:80). It also creates a volume (-v host_directory:webapache_container_directory) that synchronizes the host directory host_directory with the container's webapache_container_directory directory. This means that any changes made to the files in host_directory will be reflected in the container's webapache_container_directory directory, and vice versa, eliminating the need to rebuild the container each time there is a change to the web source code.

8. Check that the previous step works. To do this, it modifies the content of the web page and test that when updating the changes are published by the server.

To check if the previous step works, you can modify the content of a web page in the host's host_directory and then test that the changes are visible on the web server running in the container. You can do this by accessing the web page through a web browser or by using a tool like curl to make an HTTP request to the container's IP address or hostname.