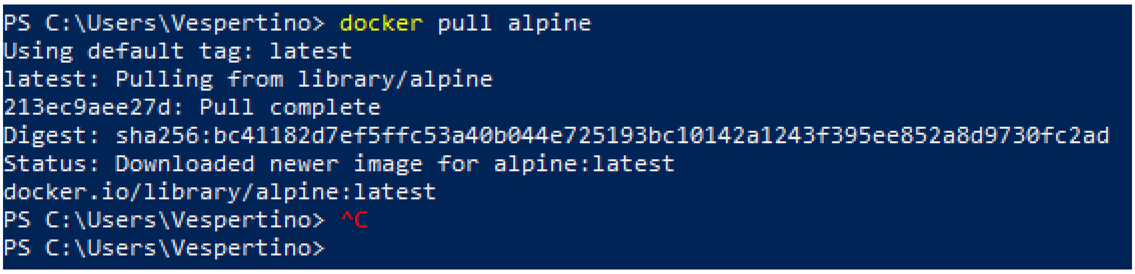
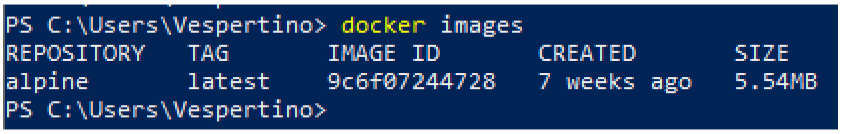


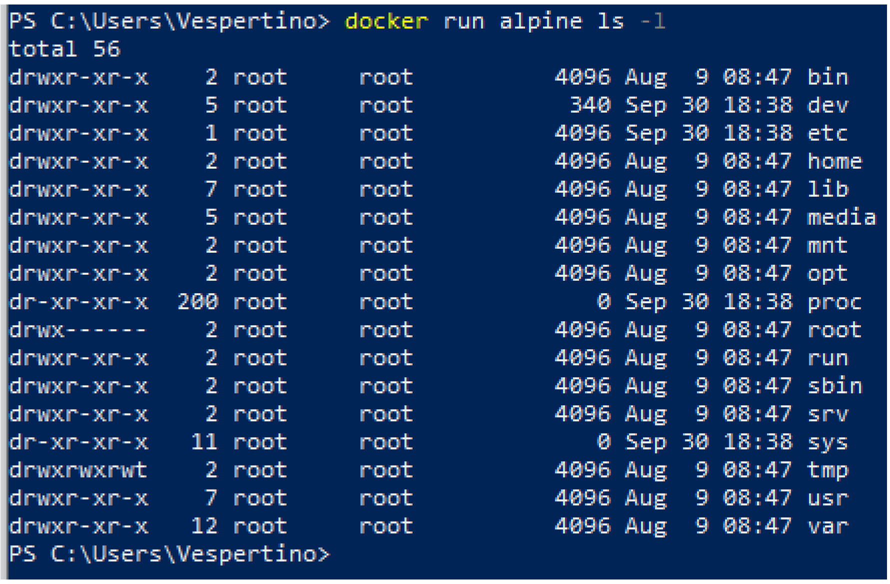
```
PS C:\Users\Vespertino> docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
213ec9aee27d: Pull complete
Digest: sha256:bc41182d7ef5ffc53a40b044e725193bc10142a1243f395ee852a8d9730fc2ad
Status: Downloaded newer image for alpine:latest
docker.io/library/alpine:latest
PS C:\Users\Vespertino> ^C
PS C:\Users\Vespertino>
```

Hemos iniciado Docker para preparar un SO Linux. Posteriormente hemos puesto este comando para obtener la imagen Alpine.



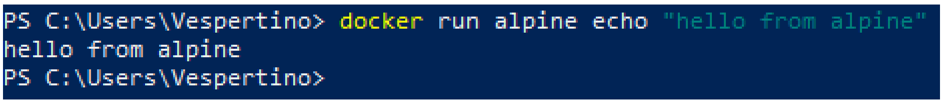
```
PS C:\Users\Vespertino> docker images
REPOSITORY   TAG       IMAGE ID       CREATED        SIZE
alpine       latest    9c6f07244728   7 weeks ago    5.54MB
PS C:\Users\Vespertino>
```

Este comando nos permite ver las imágenes que tenemos.



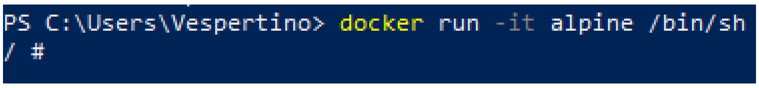
```
PS C:\Users\Vespertino> docker run alpine ls -l
total 56
drwxr-xr-x    2 root     root          4096 Aug  9 08:47 bin
drwxr-xr-x    5 root     root           340 Sep 30 18:38 dev
drwxr-xr-x    1 root     root          4096 Sep 30 18:38 etc
drwxr-xr-x    2 root     root          4096 Aug  9 08:47 home
drwxr-xr-x    7 root     root          4096 Aug  9 08:47 lib
drwxr-xr-x    5 root     root          4096 Aug  9 08:47 media
drwxr-xr-x    2 root     root          4096 Aug  9 08:47 mnt
drwxr-xr-x    2 root     root          4096 Aug  9 08:47 opt
dr-xr-xr-x  200 root     root             0 Sep 30 18:38 proc
drwx------    2 root     root          4096 Aug  9 08:47 root
drwxr-xr-x    2 root     root          4096 Aug  9 08:47 run
drwxr-xr-x    2 root     root          4096 Aug  9 08:47 sbin
drwxr-xr-x    2 root     root          4096 Aug  9 08:47 srv
dr-xr-xr-x   11 root     root             0 Sep 30 18:38 sys
drwxrwxrwt    2 root     root          4096 Aug  9 08:47 tmp
drwxr-xr-x    7 root     root          4096 Aug  9 08:47 usr
drwxr-xr-x   12 root     root          4096 Aug  9 08:47 var
PS C:\Users\Vespertino>
```

Lanzamos la imagen y vemos lo que hay dentro.



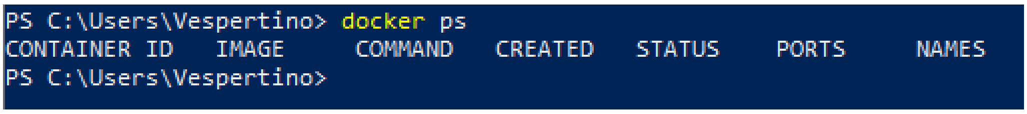
```
PS C:\Users\Vespertino> docker run alpine echo "hello from alpine"
hello from alpine
PS C:\Users\Vespertino>
```

Instanciamos un nuevo SO y ejecutamos el comando echo



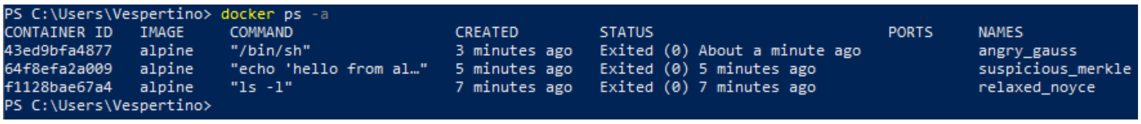
```
PS C:\Users\Vespertino> docker run -it alpine /bin/sh
/ #
```

Ejecutamos la consola del SO.



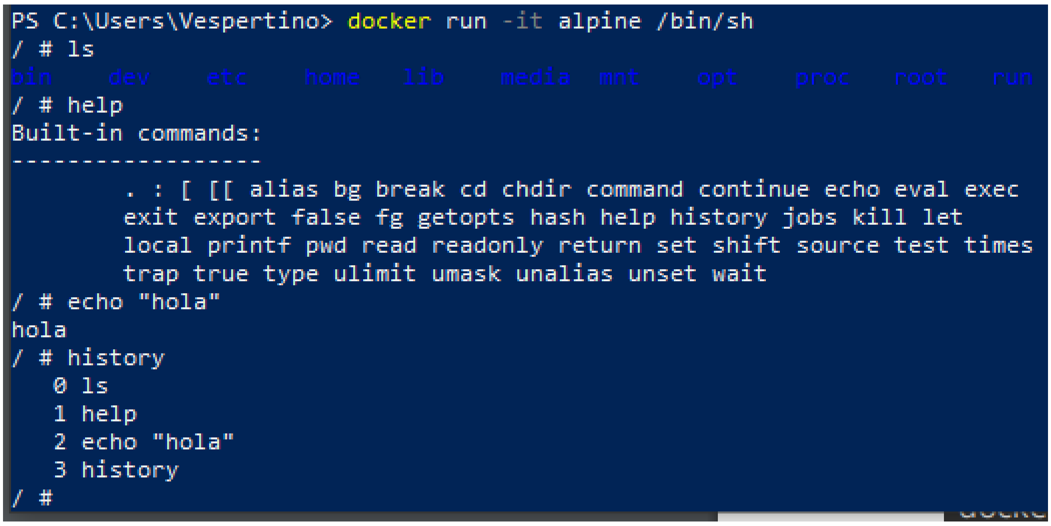
```
PS C:\Users\Vespertino> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
PS C:\Users\Vespertino>
```

Nos permite ver los contenedores que hay ejecutados



```
PS C:\Users\Vespertino> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS                          PORTS     NAMES
43ed9bfa4877   alpine    "/bin/sh"                3 minutes ago   Exited (0) About a minute ago             angry_gauss
64f8efa2a009   alpine    "echo 'hello from al…"   5 minutes ago   Exited (0) 5 minutes ago                  suspicious_merkle
f1128bae67a4   alpine    "ls -l"                  7 minutes ago   Exited (0) 7 minutes ago                  relaxed_noyce
PS C:\Users\Vespertino>
```

Vemos los contenedores que se han ejecutado, como un historial de los contenedores ejecutados.

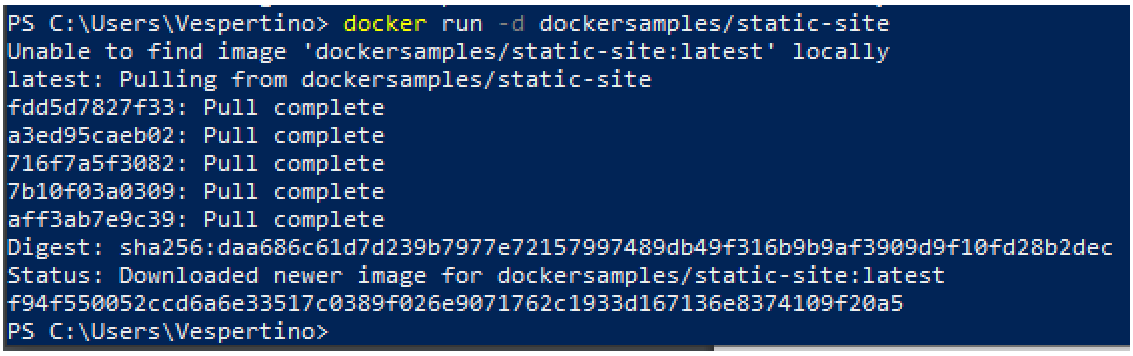


```
PS C:\Users\Vespertino> docker run -it alpine /bin/sh
/ # ls
bin    dev    etc    home   lib    media  mnt    opt    proc   root   run
/ # help
Built-in commands:
------------------
        . : [ [[ alias bg break cd chdir command continue echo eval exec
        exit export false fg getopts hash help history jobs kill let
        local printf pwd read readonly return set shift source test times
        trap true type ulimit umask unalias unset wait
/ # echo "hola"
hola
/ # history
   0 ls
   1 help
   2 echo "hola"
   3 history
/ #
```

Entramos en la consola y probamos a poner nuestros comandos favoritos.

Al usar Docker Ps ¿Aparece algún contender? ¿Por qué?

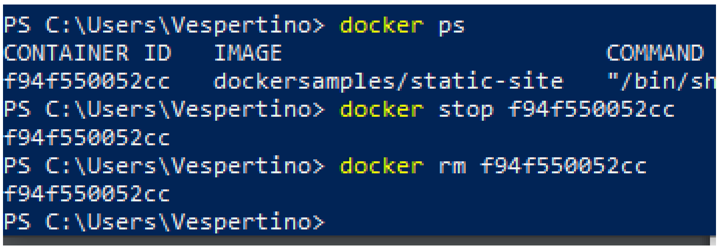
Si ya que estábamos ejecutándolo mientras lo pusimos.



```
PS C:\Users\Vespertino> docker run -d dockersamples/static-site
Unable to find image 'dockersamples/static-site:latest' locally
latest: Pulling from dockersamples/static-site
fdd5d7827f33: Pull complete
a3ed95caeb02: Pull complete
716f7a5f3082: Pull complete
7b10f03a0309: Pull complete
aff3ab7e9c39: Pull complete
Digest: sha256:daa686c61d7d239b7977e72157997489db49f316b9b9af3909d9f10fd28b2dec
Status: Downloaded newer image for dockersamples/static-site:latest
f94f550052ccd6a6e33517c0389f026e9071762c1933d167136e8374109f20a5
PS C:\Users\Vespertino>
```

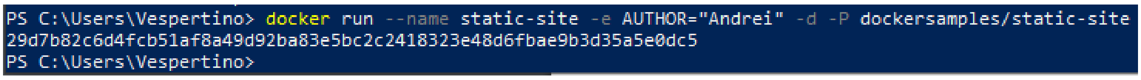
Entonces, ¿qué sucede cuando se ejecuta este comando?

Esta descargando el SO y el html donde está el sitio web.



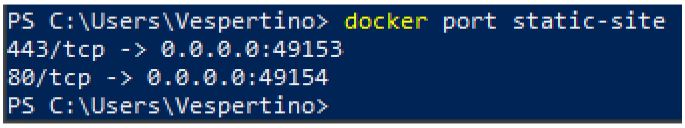
```
PS C:\Users\Vespertino> docker ps
CONTAINER ID   IMAGE                       COMMAND
f94f550052cc   dockersamples/static-site   "/bin/sh
PS C:\Users\Vespertino> docker stop f94f550052cc
f94f550052cc
PS C:\Users\Vespertino> docker rm f94f550052cc
f94f550052cc
PS C:\Users\Vespertino>
```

Paramos el Docker y lo eliminamos.



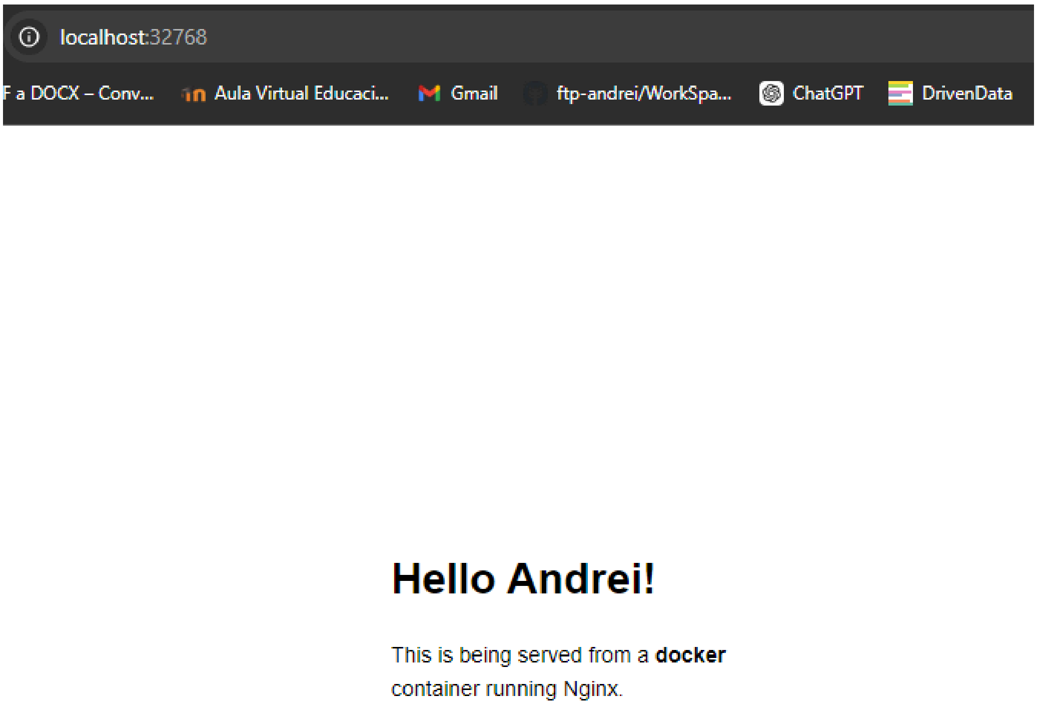
```
PS C:\Users\Vespertino> docker run --name static-site -e AUTHOR="Andrei" -d -P dockersamples/static-site
29d7b82c6d4fcb51af8a49d92ba83e5bc2c2418323e48d6fbae9b3d35a5e0dc5
PS C:\Users\Vespertino>
```

Lanzamos el contenedor



```
PS C:\Users\Vespertino> docker port static-site
443/tcp -> 0.0.0.0:49153
80/tcp -> 0.0.0.0:49154
PS C:\Users\Vespertino>
```

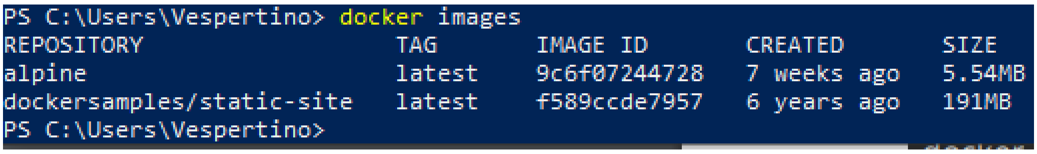
Miramos los puertos.



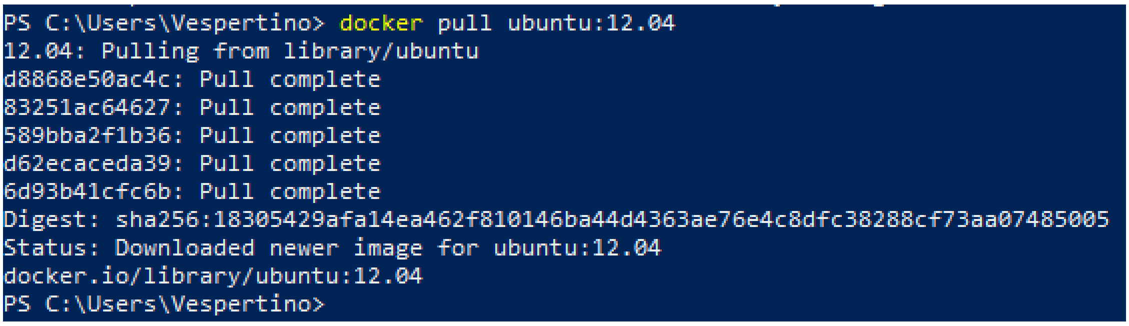
Hello Andrei!

This is being served from a **docker** container running Nginx.

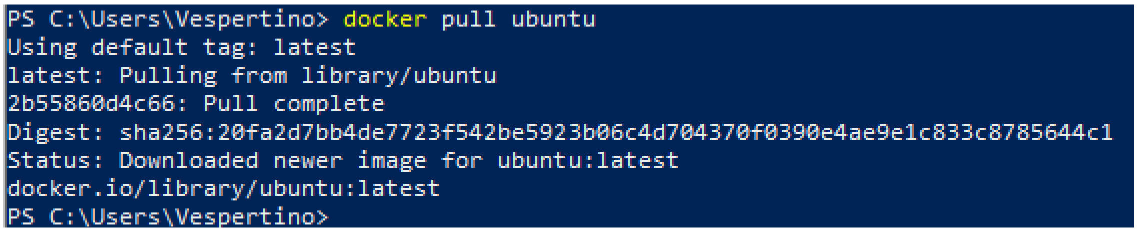
Entramos a la web a ver si funciona.



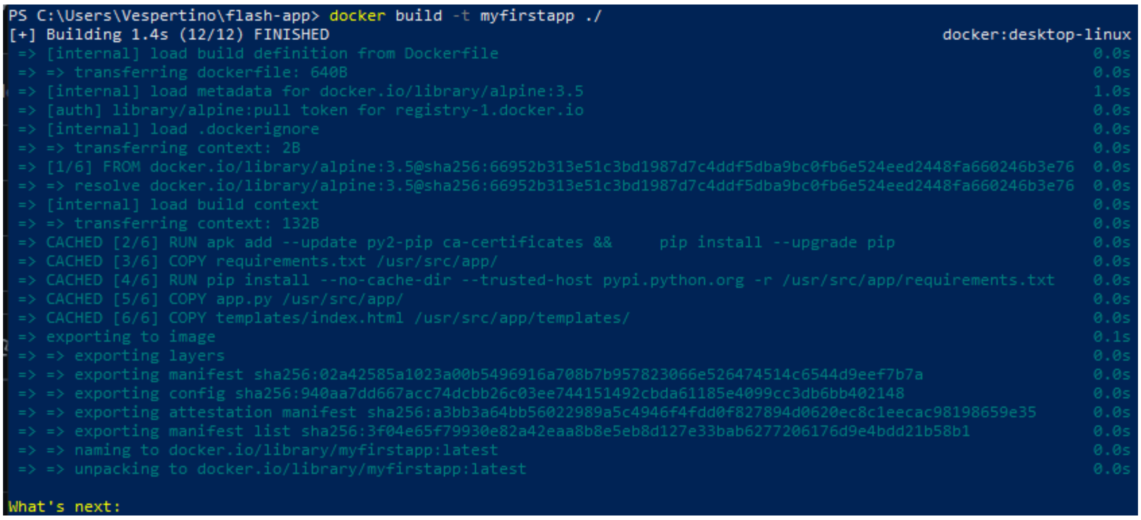
```
PS C:\Users\Vespertino> docker images
REPOSITORY                  TAG       IMAGE ID       CREATED       SIZE
alpine                      latest    9c6f07244728   7 weeks ago   5.54MB
dockersamples/static-site   latest    f589ccde7957   6 years ago   191MB
PS C:\Users\Vespertino>
```



```
PS C:\Users\Vespertino> docker pull ubuntu:12.04
12.04: Pulling from library/ubuntu
d8868e50ac4c: Pull complete
83251ac64627: Pull complete
589bba2f1b36: Pull complete
d62ecaceda39: Pull complete
6d93b41cfc6b: Pull complete
Digest: sha256:18305429afa14ea462f810146ba44d4363ae76e4c8dfc38288cf73aa07485005
Status: Downloaded newer image for ubuntu:12.04
docker.io/library/ubuntu:12.04
PS C:\Users\Vespertino>
```

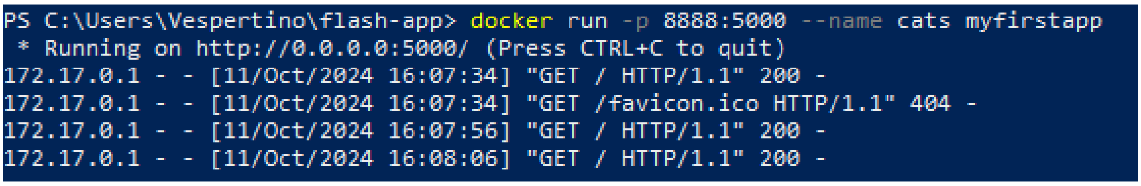


```
PS C:\Users\Vespertino> docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
2b55860d4c66: Pull complete
Digest: sha256:20fa2d7bb4de7723f542be5923b06c4d704370f0390e4ae9e1c833c8785644c1
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
PS C:\Users\Vespertino>
```



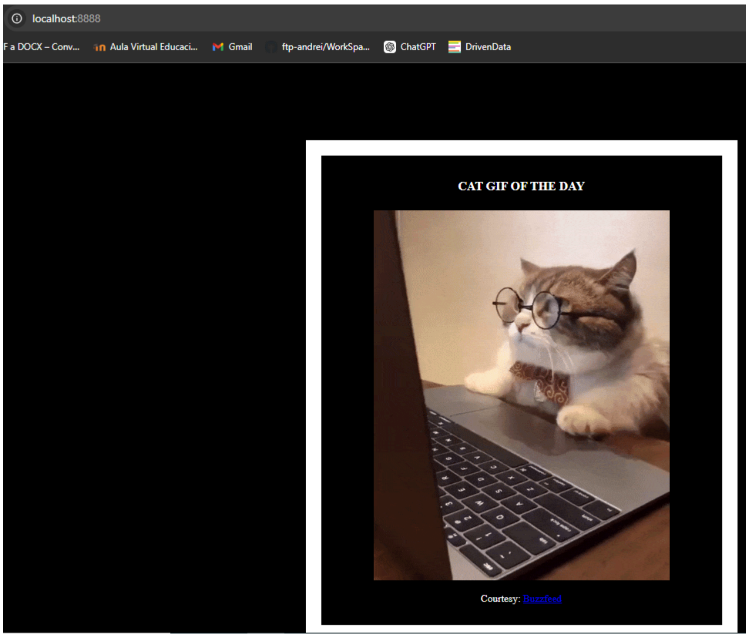
```
PS C:\Users\Vespertino\flash-app> docker build -t myfirstapp ./
[+] Building 1.4s (12/12) FINISHED                                                                   docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                              0.0s
 => => transferring dockerfile: 640B                                                                              0.0s
 => [internal] load metadata for docker.io/library/alpine:3.5                                                     1.0s
 => [auth] library/alpine:pull token for registry-1.docker.io                                                     0.0s
 => [internal] load .dockerignore                                                                                 0.0s
 => => transferring context: 2B                                                                                   0.0s
 => [1/6] FROM docker.io/library/alpine:3.5@sha256:66952b313e51c3bd1987d7c4ddf5dba9bc0fb6e524eed2448fa660246b3e76  0.0s
 => => resolve docker.io/library/alpine:3.5@sha256:66952b313e51c3bd1987d7c4ddf5dba9bc0fb6e524eed2448fa660246b3e76  0.0s
 => [internal] load build context                                                                                 0.0s
 => => transferring context: 132B                                                                                 0.0s
 => CACHED [2/6] RUN apk add --update py2-pip ca-certificates &&     pip install --upgrade pip                    0.0s
 => CACHED [3/6] COPY requirements.txt /usr/src/app/                                                              0.0s
 => CACHED [4/6] RUN pip install --no-cache-dir --trusted-host pypi.python.org -r /usr/src/app/requirements.txt   0.0s
 => CACHED [5/6] COPY app.py /usr/src/app/                                                                        0.0s
 => CACHED [6/6] COPY templates/index.html /usr/src/app/templates/                                                0.0s
 => exporting to image                                                                                            0.1s
 => => exporting layers                                                                                           0.0s
 => => exporting manifest sha256:02a42585a1023a00b5496916a708b7b957823066e526474514c6544d9eef7b7a                 0.0s
 => => exporting config sha256:940aa7dd667acc74dcbb26c03ee744151492cbda61185e4099cc3db6bb402148                   0.0s
 => => exporting attestation manifest sha256:a3bb3a64bb56022989a5c4946f4fdd0f827894d0620ec8c1eecac98198659e35     0.0s
 => => exporting manifest list sha256:3f04e65f79930e82a42eaa8b8e5eb8d127e33bab6277206176d9e4bdd21b58b1            0.0s
 => => naming to docker.io/library/myfirstapp:latest                                                              0.0s
 => => unpacking to docker.io/library/myfirstapp:latest                                                           0.0s

What's next:
```

Creamos un directorio, cogemos el repositorio de GitHub y ponemos los archivos dentro de la carpeta de flash-app. Una vez hecho eso usamos el comando que pone arriba para descomprimir y construirlo.



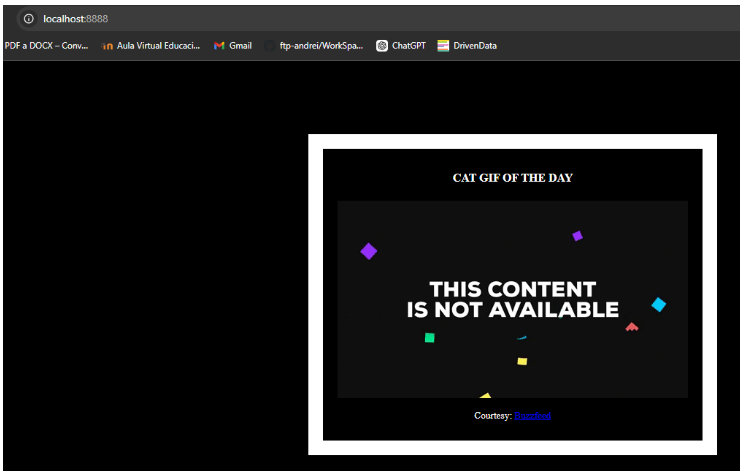
```
PS C:\Users\Vespertino\flash-app> docker run -p 8888:5000 --name cats myfirstapp
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
172.17.0.1 - - [11/Oct/2024 16:07:34] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [11/Oct/2024 16:07:34] "GET /favicon.ico HTTP/1.1" 404 -
172.17.0.1 - - [11/Oct/2024 16:07:56] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [11/Oct/2024 16:08:06] "GET / HTTP/1.1" 200 -
```

Iniciamos con este comando.



CAT GIF OF THE DAY

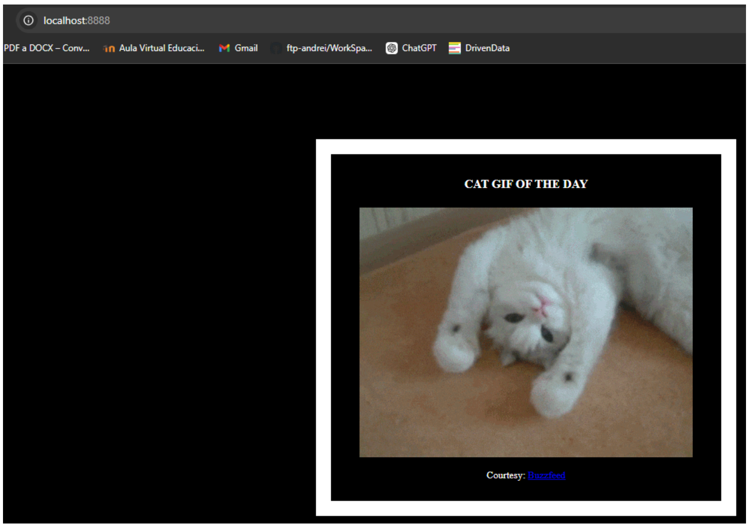
Courtesy: Buzzfeed



CAT GIF OF THE DAY

THIS CONTENT
IS NOT AVAILABLE

Courtesy: Buzzfeed



CAT GIF OF THE DAY

Courtesy: Buzzfeed

Iniciamos localhost con el puerto que se haya iniciado cats (8888) y nos saldrán imágenes random de gatos.