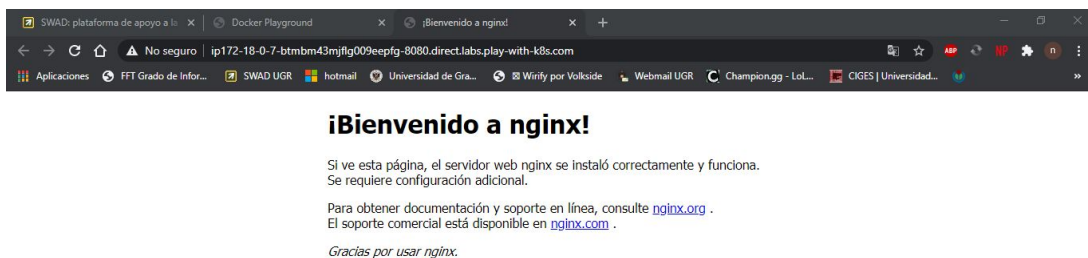


CPD_Practica_1. Docker

Del apartado VI, captura personalizada del acceso a NGINX con un fichero index.html modificado.

1. Descargamos la imagen NGINX con el comando:
`docker run --name nginx1 -p8080:80 -p8443:443 -d nginx`
2. Si pinchamos en 8080 veremos:



3. Entramos en nuestro contenedor:

```
DELETED

[node1 ~]$ docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED            STATUS
d18d0aa4eb58        nginx              "/docker-entrypoint..." 35 minutes ago    Up 35 minutes
[node1 ~]$ docker exec -it nginx1 /bin/bash
root@d18d0aa4eb58:/#
```

4. Hacemos Update y instalamos “vim” para poder modificar posteriormente el html

```
root@d18d0aa4eb58:/# apt-get update
Get:1 http://deb.debian.org/debian buster InRelease [122 kB]
Get:2 http://security.debian.org/debian-security buster/updates InRelease [65.4 kB]
Get:3 http://deb.debian.org/debian buster-updates InRelease [51.9 kB]
Get:4 http://security.debian.org/debian-security buster/updates/main amd64 Packages [231 kB]
Get:5 http://deb.debian.org/debian buster/main amd64 Packages [7906 kB]
Get:6 http://deb.debian.org/debian buster-updates/main amd64 Packages [7868 B]
Fetched 8385 kB in 2s (4749 kB/s)
Reading package lists... Done
root@d18d0aa4eb58:/# apt-get install vim
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libcm2 vim-common vim-runtime xxd
```

5. En la ruta siguiente modificamos el fichero html para que muestre lo que deseamos:

root@d18d0aa4eb58:/usr/share/nginx/html#

hacemos “vi index.html”

```
<!DOCTYPE html>
<html>
<head>
<title>P1 CPD ,MODIFICACION </title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>P1 CPD, Modificacion</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
```

Recargamos la pagina y vemos que se han producido los cambios.

ip172-18-0-7-btmbm43mjflg009eepfg-8080.direct.labs.play-with-k8s.com

SWAD UGR hotmail Universidad de Gra... Wirify por Volkside Webmail UGR

P1 CPD, Modificacion

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

Del apartado IX, creación interactiva de un contenedor docker.

1. Creamos el contenedor:

docker run -i -t ubuntu bash

```
[node2 ~]$ docker run -i -t ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
e6ca3592b144: Pull complete
534a5505201d: Pull complete
990916bd23bb: Pull complete
Digest: sha256:cbcf86d7781dbb3a6aa2bcea25403f6b0b443e20b9959165cf52d2cc9608e4b9
Status: Downloaded newer image for ubuntu:latest
root@ac3c84d0acbf:/#
```

Estaremos dentro del contenedor, actualizaremos.

apt update

```
root@811f68c7cd43:/# apt update
```

apt upgrade

```
root@811f68c7cd43:/# apt upgrade
Reading package lists... Done
```

Instalamos nginx:

apt install nginx

```
root@811f68c7cd43:/# apt install nginx
```

Salimos con: **exit**

Guardamos imagen **docker commit mi_imagen**

podemos consultar la ide de nuestro contenedor con **docker ps -a**

```
[node2 ~]$ docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS
PORTS         NAMES
ac3c84d0acbf   ubuntu    "bash"                  10 minutes ago Exited (0) 16 seconds ago
               clever benz
[node2 ~]$ docker commit ubuntu mi_imagen
Error response from daemon: No such container: ubuntu
[node2 ~]$
[node2 ~]$ docker commit ac3c84d0acbf mi_imagen
sha256:86b7a217cb35cbb813aaf035246f59f816f8ae314c6926c0bf814a114093f6b5
[node2 ~]$
```

Comprobamos que está disponible nuestra imagen

docker images

```
URL
ip172-18-0-5-btmbm43mjflg009eepfg.direct.labs.play-with-k
```

DELETE

```
[node2 ~]$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
mi_imagen     latest   86b7a217cb35   57 seconds ago 171MB
ubuntu        latest   bb0eaf4eee00   7 days ago    72.9MB
[node2 ~]$
```

Del anexo1, acceso por SSHFS.

Como esas máquinas virtuales son CentOS 7, hacemos una instalación con yum:

`yum -y install epel-release`

```
[node5 ~]$ yum -y install epel-release
```

`yum clean all`

```
[node5 ~]$ yum clean all
```

`yum -y install fuse-sshfs`

```
[node5 ~]$ yum -y install fuse-sshfs
```

Creamos un directorio y montamos remotamente nuestro home en él.

`mkdir mi_turing`

```
[node5 ~]$ mkdir mi_turing
```

```
[node5 ~]$ ls
```

```
anaconda-ks.cfg  mi_turing
```

```
[node5 ~]$ sshfs igmorillas@turing.ugr.es:. mi_turing
```

Aquí se ve lo que tengo en el interior de mi directorio

```
[node5 ~]$ ls
```

```
anaconda-ks.cfg  mi_turing
```

```
[node5 ~]$ ls ./mi_turing/
```

```
.Trash-2989/      MC/               ej2.c
.cache/           PDOO/             ej3
.ssh/             SCD/              ej3.c
4 Practica 4/     SO/               examen.txt
4 Practica 4.zip  SONachoElSalvador.zip  netflix
ACAP/             SOañopasado/      netflix.txt
archivos_imprimir/
```

desmontar el directorio

`umount mi_turing`

```
[node5 ~]$ umount mi_turing
```