

Fase 4 - Activitat 8.1: Desplegant una aplicació web amb contenidors Docker

0- Identificació del grup i activitat:

Curs: ASIX2

Projecte: PJ9 DevOps i Cloud Computing

Fase: 4

Activitat: 8.1

Grup/Individual: Individual

Membres/Alumne:

1.- Objectiu de l'activitat

Els objectius objectius d'aquesta activitat són:

- Aprendre noves ordres per treballar amb **Vagrant**
- Aprendre noves ordres per treballar amb **Git i Github**
- Ampliar els nostres coneixements sobre **Vagrant**.
- Ampliar els nostres coneixements sobre **Git i Github**.
- Tenir els equips de tots els membre de tots els grups del projecte preparats per treballar plenament amb **Vagrant, Git i Github**.

2.- Creació d'un fitxer Vagrantfile per crear una màquina que treballi amb Docker

a) Dins de la carpeta **pj9** → **f4** de la teva màquina física crea una carpeta de nom **a8**. Dins de la carpeta **a8** crea una altra carpeta de nom **pj9f4a8.1**. Dins de **pj9f4a8.1** crea 2 directoris: un directori de nom **codi** i un altre de nom **vm**.

b) Copia el fitxer **Vagrantfile** que vas crear dins de **pj9** → **f4** → **a7** → **gp1f4a7.7** → **vm** dins de **pj9** → **f4** → **a7** → **gp1f4a8.1** → **vm**.

c) Modifica el nou fitxer **Vagrantfile** de manera que:

- El nom de la màquina virtual dins del gestor serà a on **pj9f4a8.1_xxyyzz**. Recorda que **xxyyzz** són les 2 primeres lletres dels teus nom i cognoms.
- El nom de sistema de la màquina virtual serà **pj9f4a81-xxyyzz.fjeclot.net**. Recorda que **xxyyzz** són les 2 primeres lletres dels teus nom i cognoms.
- La carpeta compartida **codi** de la màquina física estarà associada a la carpeta **/home/vagrant/pizzeria** de la màquina virtual.

d) Crea i inicia la màquina, accedeix-hi i comprova que el programari de contenidors **Docker** està instal·lat.

3.- Desplegament d'una aplicació PHP per mitjà de contenidors Docker

a) Surt de la màquina virtual. Dins de la màquina física, accedeix a la carpeta **codi** i crea una carpeta de nom **app**. Dins de la carpeta **app**, descarrega els següents 2 fitxers:

- **pizza.html** → <https://raw.githubusercontent.com/asix2pj9/pizzeria/refs/heads/main/pizza.html>
- **pizza.php** → <https://raw.githubusercontent.com/asix2pj9/pizzeria/refs/heads/main/pizza.php>

b) Dins de la carpeta **codi** de la màquina física crea un fitxer de nom Dockerfile amb aquest contingut:

```
FROM php:7.4-apache
```

```
COPY app /var/www/html
```

```
EXPOSE 80
```

c) Accedeix novament a la màquina virtual i comprova que dins de la carpeta `/home/vagrant/pizzeria` s'ha creat:

- La carpeta `app` que té dins els fitxers `pizza.html` i `pizza.php`.
- El fitxer `Dockerfile` amb el contingut creat a l'apartat anterior

d) Crea una imatge del contenidor Docker de la teva aplicació executant:

```
docker build -t ipizza:1.0 .
```

i comprova que es crea la imatge `ipizza` versió `1.0` executant:

```
docker images
```

d) Crea un contenidor per accedir a l'aplicació executant:

```
docker run --name cpizza -i -t -d -p 80:80 ipizza:1.0
```

i comprova que es s'ha creat el contenidor `cpizza` i que els seu estat és `Up`. Executa:

```
docker ps -a
```

e) Comprova les adreces IP de la teva màquina física i de la teva màquina virtual.

f) Comprova que la màquina virtual té el port `80/tcp` obert pel programari de contenidors `Docker`.

g) Des de la màquina física, accedix a la pàgina inicial de l'aplicació `pizza.html`:

- Utilitzant l'adreça IP de la màquina virtual i accedint directament al port `80/tcp` de la màquina virtual.
- Utilitzant l'adreça IP de la màquina física i recordant que el port `80/tcp` de la màquina virtual està exportat al `8000/tcp` de la màquina física.

Lliurament de l'activitat

a) Comprovacions:

- Que es pot accedir a l'aplicació des de la màquina física utilitzant l'adreça IP de la màquina virtual i accedint al port `80/tcp`.
- Que es pot accedir a l'aplicació des de la màquina física utilitzant l'adreça IP de la màquina física i accedint al port `8000/tcp`.

b) Data límit per obtenir el 100% de la nota: **dilluns 30-10-24 i dilluns 11-11-24** fins a les **16.50**. Posteriorment la nota serà menor.

Annex - Altres ordres de gestió de contenidor i imatges docker

`docker stop <nom_contenidor>` → Atura un contenidor

`docker start <nom_contenidor>` → Inicia un contenidor

`docker rm <nom_contenidor>` → Esborra un contenidor

`docker exec -it <nom_contenidor> bash` → Accedir dins del contenidor amb un terminal

`docker rmi <nom_imatge>` → Esborra una imatge