

Fase 4 - Activitat 8.3: Eina docker compose i fitxer docker-compose.yml

0- Identificació del grup i activitat:

Curs: ASIX2

Projecte: PJ9 DevOps i Cloud Computing

Fase: 4

Activitat: 8.3

Grup/Individual: Individual

Membres/Alumne:

1- Introducció i objectius de l'activitat 8.3

- a) Comprendre el propòsit del fitxer **.dockerignore**
- b) Comprendre el funcionament i propòsit de l'ordre **docker compose**
- c) Comprendre el propòsit del fitxer **docker-compose.yml**
- d) Tenir una breu introducció al format YAML
- e) Realitzar una activitat avaluativa

2- Llegeix-me: host, .dockerignore, docker compose, docker-compose.yml i format YAML en poques paraules

a) Quan tractem el tema dels contenidors amb Docker, el **host** és l'equip informàtic sobre el qual es creen i s'executen els contenidors. En el nostre cas és normalment una màquina virtual creada amb Vagrant però també pot ser un equip físic.

a) Dins d'un fitxer **.dockerignore** escriurem la llista de fitxers i directories que NO volem que s'afegeixin a una imatge de contenidor quan fem un build. Això permet ajustar la mida de les imatges de contenidors, crear-les més ràpidament i augmentar la seguretat. Per més informació llegeix [aquest link](#).

b) L'eina **docker compose** permet llegir la configuració que es troba dins d'un arxiu de nom **docker-compose.yml** i a continuació amb una simple instrucció i de manera ràpida i automàtica:

- Crear totes les imatges (una o més) que siguin necessàries per desplegar una aplicació
- Crear i posar en marxa tots els contenidors (un o més) que siguin necessaris per desplegar una aplicació amb tots les seves opcions (noms, ports, imatges, etc...).
- Si cal, crear Volums per compartir dades entre contenidors i la màquina host dels contenidors, o també entre els contenidors amb uns altres.
- Crear una xarxa de contenidors perquè els contenidors es puguin comunicar entre ells d'una manera ràpida i eficient que si ho haguessin de fer a través de la màquina host dels contenidors.

Amb l'eina **docker compose** es molt fàcil desplegar des d'una aplicació sencilla que només té un contenidor fins a una aplicació complexa amb múltiples contenidors sense que calgui res més que el fitxer **docker-compose.yml** i executar una instrucció molt senzilla.

En comparació a fer la feina manualment, **docker compose** és més ràpid, més fàcil d'utilitzar, més fiable perquè redueix la probabilitat de cometre un error i més compartible. Per més informació llegeix [aquest link](#).

c) Allò que diferencia un tècnic qualificat d'un simple usuari de l'eina **docker compose** és la seva capacitat per crear un fitxer **docker-compose.yml**. Aquest fitxer utilitza el nou format **YAML** per definir els contenidors, imatges, volums, xarxes i dependències entre contenidors que són necessàries per desplegar una aplicació. El format **YAML** com més va més s'utilitza per crear arxius de configuració de programes i serveis, definir estructures de dades, etc..., i hauria de ser conegut per qualsevol administrador o desenvolupador. Per més informació llegeix [aquest link](#).

d) Respecte del format **YAML**:

- Els fitxer que utilitzen YAML són de text, llegible pels humans i tenen l'extensió **.yaml** o **.yml**.
- Igual que python les sagnies són obligatòries per indicar que un paràmetre està dins d'un altre.
- Les sagnies han de ser sempre iguals i només es poder utilitzar espais en blanc (mai tabuladors).
- Utilitza mapejat per associar un paràmetre amb el seu valor. Per exemple → **image: ipizza:1.0**
- Per veure altres característiques, llegeix [aquest link](#).

3- Creant la màquina virtual amb Vagrant

a) Dins de la màquina física:

- Accedeix a la carpeta **pj9** → **f4** → **a8** i crea una carpeta de nom **pj9f4a8.3**.
- Dins **pj9f4a8.3** copia el fitxer **Vagrantfile** de l'activitat **pj9f4a8.2** i fes els següents canvis:
 - **NOM_MAQUINA = pj9f4a8.3_xxyzz**
 - **HOSTNAME = pj9f4a83-xyzz.fjeclot.net**
Recorda que **xyzz** són les 2 primeres lletres del nom i cognoms.
 - Comenta les línies **13**, **14** i **39** per deshabilitar la compartició de la carpeta **codi** de la màquina física amb la carpeta **/home/vagrant/pizzeria** de la màquina virtual.
- Crea i inicia la màquina virtual. Comprova que la màquina es crea i s'inicia.

b) Dins de la màquina virtual comprova:

- Que el nom de sistema és correcte.
- L'adreça IP de la màquina virtual.
- Que **/vagrant** de la màquina virtual està compartida amb la carpeta **pj9f4a8.3** de la física.

4- Modificació de l'aplicació. Crea un dipòsit local i remot per pujar l'aplicació i el seu entorn de treball amb contenidors Docker i Vagrant.

a) Dins de la màquina física:

- Copia la carpeta **app** que vas crear a l'activitat **pj9f4a8.2** dins de la carpeta **pj9f4a8.3**.
- Modifica el fitxer **pizza.php**. Canvia així el contingut de les línies indicades:
 - Línia **2** → **\$pvp_inicial=10;**
 - Línia **4** → **echo "El preu de la pizza bàsica és \$pvp_inicial € (IVA inclòs)
";**
 - Línia **9** → **\$pvp=((100+\$IVA)/100)*(\$pvp_inicial+count(\$_GET['ingr']));**

b) Dins de la màquina física, i dins de la carpeta **pj9** → **f4** → **a8** → **pj9f4a8.3**:

- Crea dins **pj9f4a8.3** un fitxer de nom **.gitignore** amb el següent contingut:
.gitignore
.vagrant
- Crea un fitxer **Dockerfile** sense contingut.
- Crea un fitxer de nom **.dockerignore** sense contingut.
- Crea un fitxer de nom **docker-compose.yml** sense contingut.
- **Inicia** un dipòsit de **Git** dins de **pj9f4a8.3**.
- Fes un **add** del continguts del directori **pj9f4a8.3** incloent-hi el fitxer ocult **.dockerignore** (i exceptuant el indicats a **.gitignore**). Executa la següent ordre: **git add**.
- Fes un **commit** del continguts del directori **pj9f4a8.3** (excepte el indicats a **.gitignore**) dins del dipòsit local. El missatge del commit serà: **"Versió 1.0 del nou projecte pizzeria - Codi PHP i Vagrantfile"**.

c) Comprova que s'han pujat al dipòsit local els fitxers:

- **pizza.html** i **pizza.php**, de la carpeta **apl**.
- **Vagrantfile**
- **Dockerfile**
- **docker-compose.yml**

Executa: **git show --pretty="" --name-only**

- d) Crea dins del teu compte de **Github** un dipòsit de nom **pj9f4a8.3** de tipus **Privat**.
- e) Sincronitza el dipòsit local amb el dipòsit remot amb la informació que trobarà a la secció **...or push an existing repository from the command line**. Recorda que les instruccions s'han d'executar en el directori a on tens el dipòsit, o sigui, dins de **pj9f4a8.3**.
- f) Refresca la web del dipòsit per assegurar-te que els fitxers del projecte s'han pujat dins de les seves carpetes.

5- Creant el fitxer Dockerfile i afegint-lo al dipòsit del projecte

- a) Dins de la màquina física, i dins de la carpeta **pj9 → f4 → a8 → pj9f4a8.3**:
- Obre el fitxer **Dockerfile**.
 - Fes que el seu contingut sigui:

```
FROM php:8.2-apache
COPY app /var/www/html
EXPOSE 80
```
- b) Fes un **add** del continguts del directori **pj9f4a8.3**.
- c) Fes un nou **commit** del continguts del directori **pj9f4a8.3**. El missatge del commit serà: **"Versió 1.1 del nou projecte pizzeria - Dockerfile"**.
- d) Comprova que s'ha pujat al dipòsit local el fitxer **Dockerfile**.
- e) Puja la nova versió del projecte al dipòsit remot i refresca la web del dipòsit remot per assegurar-te que els fitxers del projecte s'han pujat.

6- Creant un fitxer .dockerignore per ignorar fitxers i directoris durant un build

- a) Dins de la màquina física, i dins de la carpeta **pj9 → f4 → a8 → pj9f4a8.3**:
- Obre el fitxer **.dockerignore**.
 - Fes que el seu contingut sigui:

```
.dockerignore
.gitignore
.git
.vagrant
Vagrantfile
docker-compose.yml
```
- b) Fes un **add** del continguts del directori **pj9f4a8.3**.
- c) Fes un nou **commit** del continguts del directori **pj9f4a8.3**. El missatge del commit serà: **"Versió 1.2 del nou projecte pizzeria - .dockerignore"**.
- d) Comprova que s'ha pujat al dipòsit local el fitxer **.dockerignore**.
- e) Puja la nova versió del projecte al dipòsit remot i refresca la web del dipòsit remot per assegurar-te que els fitxers del projecte s'han pujat.

7- Creant un fitxer docker-compose.yml mínim per iniciar l'aplicació

a) Dins de la màquina física, i dins de la carpeta **pj9** → **f4** → **a8** → **pj9f4a8.3** i crea un fitxer de nom **docker-compose.yml**.

b) Fes que el contingut de **docker-compose.yml** sigui:

```
services:
  webphp_pizzeria: # L'etiqueta identificadora d'aquest servei dins de docker-compose.yml
    image: ipizzeria:1.0 # El nom que veurem quan fem docker images
    build: .
    container_name: cpizzeria # El nom que veurem quan fem docker ps -a
    ports:
      - "80:80" # Primer el port del host i després el del contenidor
# Aquest docker-compose.yml:
# 1- Posa en marxa el servei webphp_pizzeria necessari per fer anar l'aplicació. Aquest
# servei utilitza un contenidor. Quan es faci un docker compose up es posarà en marxa
# aquest servei i els contenidor que necessita.
# 2- Crea una imatge de contenidor de nom ipizzeria:1.0
# 3- Crea un contenidor de nom cpizzeria que utilitzant la imatge ipizzeria:1.0
# 4- El contenidor exposa el seu port 80 al port 80 de la màquina virtual
```

b) Fes un **add** del continguts del directori **pj9f4a8.3**.

c) Fes un nou **commit** del continguts del directori **pj9f4a8.3**. El missatge del commit serà: *"Versió 1.3 del nou projecte pizzeria - - docker-compose.yml"*.

d) Comprova que s'ha pujat al dipòsit local el fitxer **docker-compose.yml**.

e) Puja la nova versió del projecte al dipòsit remot i refresca la web del dipòsit remot per assegurar-te que els fitxers del projecte s'han pujat.

8- Crea la imatge i posa en marxa el contenidor amb l'aplicació utilitzant docker-compose

a) Dins de la **màquina virtual**, accedeix a la carpeta **/vagrant** i comprova que tens els següents arxius:

```
vagrant@pj9f4a83-dacomo:~$ cd /vagrant
vagrant@pj9f4a83-dacomo:/vagrant$ ls -la
.  ..  .dockerignore  .git  .gitignore  .vagrant  Dockerfile  Vagrantfile  app  docker-compose.yml
vagrant@pj9f4a83-dacomo:/vagrant$
```

b) Ara, dins del directori **/vagrant** executa: **docker compose up -d**

c) Si tot ha anat bé, el sistema mostrarà el missatge: **✓ Container cpizzeria Started**

d) Comprova que s'ha creat:

- La imatge **ipizzeria:1.0**
- El contenidor **cpizzeria**

e) Utilitzant l'adreça IP de la teva màquina virtual i el port 80, accedeix des d'un navegador de la màquina física a l'aplicació i comprova que funciona.

9- Aturant l'aplicació utilitzant docker compose

a) Dins del directori **/vagrant** executa: **docker compose down**

b) Si tot ha anat bé, el sistema mostrarà el missatge: **✓ Container cpizzeria Removed**

c) Comprova que s'ha esborrat el contenidor **cpizzeria** però que la imatge **ipizzeria:1.0** encara existeix per poder ser utilitzada la propera vegada que vulguem iniciar el contenidor.

Lliurament de l'activitat

- a) Comprovacions dins de la màquina virtual creada amb Vagrant:
- Mostra els fitxers de la carpeta **/vagrant** de la màquina virtual creada amb Vagrant.
 - Comprova que no hi ha cap imatge ni cap contenidor
 - Amb **docker compose** posa en marxa l'aplicació i comprova que es crea la imatge i el contenidor.
- b) Des de la màquina física:
- Mostra dins del teu compte de **Github** el dipòsit **pj9f4a8.3** i el seu contingut.
 - Mostra el contingut de **docker-compose.yml**
 - Accedeix a l'aplicació i comprova que funciona.
- c) Des de la màquina virtual, atura l'aplicació i comprova que desapareix el contenidor.
- d) Data límit per obtenir el 100% de la nota: **dijous 20-11-24** a les **19.10**. Posteriorment la nota serà menor.