

Fase 4 - Activitat 8.4: Volums i xarxes

0- Identificació del grup i activitat:

Curs: ASIX2

Projecte: PJ9 DevOps i Cloud Computing

Fase: 4

Activitat: 8.4

Grup/Individual: Individual

Membres/Alumne:

1- OJECTIUS DE L'ACTIVITAT

- a) Entendre el concepte de volum i de xarxa de contenidors.
- b) Aprendre a gestionar volums i xarxes de contenidors manualment
- c) Aprendre a gestionar volums i xarxes utilitzant docker compose
- d) Fer una activitat avaluable utilitzant els coneixements adquirits.

2 - APRENENT A TREBALLAR AMB VOLUMS I XARXES

2.1- Volums i xarxes en molt poques paraules

- a) Els **volums** de contenidors són un mecanisme per emmagatzemar dades de manera que:
- Puguin ser compartides entre contenidors.
 - Puguin ser compartides entre contenidors i la màquina host dels contenidors.
 - Les dades puguin emmagatzemar-se de manera permanent i no es perdin si aturem els contenidors.
 - Els volums s'utilitzen per:
 - Emmagatzemar bases de dades
 - Dades de l'aplicació: documents, fotografies, etc..
 - Backups
 - Compartició de dades amb equips remots
 - Els volums s'emmagatzemen dins del directori **/var/lib/docker/volumes**.
 - Es pot copiar o moure un volum d'un equip host de contenidors a un altre equip host de contenidors.

Per més informació llegeix [aquest link](#).

b) Una **xarxa de contenidors** permet comunicar contenidors entre ells, amb l'equip host de contenidors i amb el món exterior (o sigui, internet). Si, per exemple, una aplicació requereix dos o més contenidors que donen serveis diferent (per exemple Apache i MySQL), els 2 contenidors necessitaran una xarxa per poder comunicar entre ells i amb l'equip host de contenidors.

c) Existeixen diferents tipus de xarxes de contenidors que podem posar en marxa però normalment farem servir la de tipus **Bridge** que, és aquella que s'utilitza per defecte i que permet crear una xarxa dins d'un equip host individual. L'estructura de la xarxa de contenidors de tipus **Bridge** per defecte es pot veure a la pàgina 1 d'[aquest enllaç](#).

- d) Una **xarxa de contenidors** tipus **Bridge** té aquestes característiques:
- Tots els contenidors de la xarxa tenen la seva pròpia IP interna i el seu nom de contenidor intern.
 - Hi ha un DNS intern per passar noms interns de contenidors a IPs internes.
 - Els contenidors poden comunicar-se entre ells amb les seves IP internes o noms interns.
 - Els contenidors poden comunicar-se amb l'equip host amb les IP internes perquè l'equip host també té una IP interna.
 - L'equip host fa de router entre els contenidors i l'exterior.
 - Les aplicacions en els contenidors poden ser accessibles des del món exterior utilitzant la tècnica de Port Forwarding a on a cada IP i port intern es pot associar a una IP i port extern.

e) Podem definir la nostra pròpia xarxa de contenidors. En el cas de treballar amb una xarxa de tipus Bridge això ens permet:

- Utilitzar DNS i treballar amb noms de contenidors. Amb la xarxa Bridge per defecte no és possible.
- Es poden connectar i desconnectar containers de la xarxa sense necessitat d'aturar-la. Amb la xarxa Bridge per defecte no és possible.
- Altres avantatges que permeten fer més eficient l'us de contenidors i l'execució d'aplicacions.
- L'estructura d'una xarxa de contenidors de tipus Bridge definida per nosaltres es pot veure a la pàgina 2 d'[aquest enllaç](#).

2.2- Gestió de volums manual amb exemples

a) Creació d'un volum de nom **dades**: `docker volume create dades`

b) Llistat de volums disponibles: `docker volume ls`

d) Afegint el volum **dades** a un contenidor de nom **mysqldb2** que utilitza la imatge **mysql:8.0** i que exporta el port **3306** del contenidor al port **3306** del host. El volum **dades** estarà associat dins del contenidor al directori **/var/lib/mysql**. El password de **root** del servidor **MySQL** dins del contenidor serà **Fje24@**. Es crearà una base de dades de nom **pj9f4a84**.

```
ragrant@pj9f4a83-dacomo:~$ docker run --name mysqldb2 -i -t -d -e MYSQL_ROOT_PASSWORD=Fje24@ -e MYSQL_DATABASE=pj9f4a84 -v dades:/var/lib/mysql -p 3306:3306 mysql:8.0
```

e) Esborra el volum de nom **dades**: `docker volume rm dades` (el contenidor **mysqldb2** ha d'estar aturat)

2.3- Treballant amb volums utilitzant docker-compose.yml

Exemple de **docker-compose.yml** que pot crear un contenidor de nom **mysqldb2** que utilitza la imatge **mysql:8.0** i que exporta el port **3306** del contenidor al port **3306** del host. El volum **dades** estarà associat dins del contenidor al directori **/var/lib/mysql**. El password de **root** del servidor **MySQL** dins del contenidor serà **Fje24@**. Es crearà una base de dades de nom **pj9f4a84**.

```
services:
  mysql: # L'etiqueta identificadora d'aquest servei dins de docker-compose.yml
    image: mysql:8.0 # El nom que veurem quan fem docker images
    container_name: mysqldb2 # El nom que veurem quan fem docker ps -a
    environment:
      MYSQL_ROOT_PASSWORD: Fje24@
      MYSQL_DATABASE: pj9f4a84
    volumes:
      - vdades:/var/lib/mysql # Etiqueta identificadora del volum dins docker-compose.yml és vdades
    ports:
      - "3306:3306"

volumes:
  vdades:
    name: dades # El nom de volum que veurem quan executem docker volume ls
```

Aquest fitxer crearà el volum si no existeix.

2.4- Gestió de xarxes de contenidors manualment

a) Creació d'una xarxa de nom **pj9f4a84**: `docker network create pj9f4a84`

b) Comprova que a l'equip host ha aparegut una interfície de nom **br-<identificador>** que té l'adreça IP **172.18.0.1** amb l'ordre ip **a**. El valor **<identificador>** és una combinació de 12 números hexadecimals.

c) Llistat de xarxes disponibles: `docker network ls`

d) Esborra la xarxa de nom **dades**: `docker network rm pj9f4a84`

e) Comprova que a l'equip host ha desaparegut una interfície de nom **br-<identificador>** amb l'ordre ip **a**.

f) Unint dos contenidors a la xarxa **pj9f4a84**:

```
docker pull nginx:latest # Descarrega la darrera imatge del servidor web nginx
docker pull mysql:latest # Descarrega la darrera imatge del servidor mysql
docker network create pj9f4a84
docker run -name web -i -t -d -p 80:80 --network pj9f4a84 nginx:latest
docker run -name bd -i -t -d -e MYSQL_ROOT_PASSWORD=Fje24@ -p 3306:3306 --network pj9f4a84 mysql:latest
```

g) Comprova l'adreça IP dels contenidors:

```
docker inspect web | grep IPAddress | grep 172 # Troba l'adreça IP del contenidor web
docker inspect bd | grep IPAddress | grep 172 # Troba l'adreça IP del contenidor mysql
```

2.5- Treballant amb xarxes utilitzant docker-compose.yml

Exemple de **docker-compose.yml** que pot crear 2 contenidors de noms **web** i **mysql** que es connectaran a una xarxa de nom **pj9f4a84**. Si les xarxa no existeix es crearà. Descarregarà les imatges si cal.

```
services:
  web_pj9f4a84: # L'etiqueta identificadora d'aquest servei dins de docker-compose.yml
    image: nginx:latest
    container_name: web # El nom que veurem quan fem docker ps -a
    ports:
      - "80:80"
    networks:
      - xarxa_pj9f4a84 # L'etiqueta identificadora d'aquesta xarxa dins de docker-compose.yml

  mysql_pj9f4a84:
    image: mysql:latest # L'etiqueta identificadora d'aquest servei dins de docker-compose.yml
    container_name: mysql # El nom que veurem quan fem docker ps -a
    environment:
      MYSQL_ROOT_PASSWORD: Fje24@
    ports:
      - "3306:3306"
    networks:
      - xarxa_pj9f4a84 # L'etiqueta identificadora d'aquesta xarxa dins de docker-compose.yml

networks:
  xarxa_pj9f4a84: # L'etiqueta identificadora d'aquesta xarxa dins de docker-compose.yml
    name: pj9f4a84 # El nom que veurem quan fem docker network ls
    driver: bridge
```

3- ACTIVITAT AVALUABLE

NOTA: Recorda sempre que **xyyyz** són les 2 primeres lletres dels teus nom i cognoms.

a) Accedeix a la màquina **pj9f4a8.3** i dins del directori **/home/vagrant** crea una carpeta de nom **pj9f4a84**. A continuació, accedeix a la carpeta i crea un fitxer **docker-compose.yml** que faci les següent accions:

- Crea un servei de nom **Sweb_xyyyz** amb les següents característiques:
 - Utilitza la imatge **php versió 8.3-apach3**
 - Nom del contenidor: **Cweb_xyyyz**
 - El contenidor comparteix **/var/www/html** amb un volum en el host de nom **Vapl_xyyyz** etiquetat **Vdades_xyyyz** dins de **docker-compose.yml**.
 - Exporta el port **80** del contenidor al port **8080** del host.
 - El contenidor estarà connectat a una xarxa tipus **bridge** de nom **pj9f4a84_xyyyz** etiquetada **Xpj9f4a84_xyyyz** dins de **docker-compose.yml**.
- Crea un servei de nom **Smysql_xyyyz** amb les següents característiques:
 - Utilitza la imatge **mysql versió php versió 9.1**
 - Nom del contenidor: **Cmysql_xyyyz**
 - La contrasenya de **root** de **mysql** serà **Fje2425@**
 - Es crearà una base de dades de nom **BD_xyyyz**
 - El contenidor comparteix **/var/lib/mysql** amb un volum en el host de nom **dades_xyyyz** etiquetat **Vdades_xyyyz** dins de **docker-compose.yml**.
 - Exporta el port **3306** del contenidor al port **33306** del host.
 - El contenidor estarà connectat a una xarxa tipus **bridge** de nom **pj9f4a84_xyyyz** etiquetada **Xpj9f4a84_xyyyz** dins de **docker-compose.yml**.

b) A continuació, amb **docker compose**, descarrega les imatges, crea la xarxa i els volum i posa en marxa els contenidors. Quan acabi el procés, comprova:

- Les imatges existents
- Els contenidor existents
- Les xarxes existents
- Els volums existents

c) Comprova que dins de la màquina host s'han obert els ports **8080** i **33306** associats als ports **80** i **3306** del contenidors.

d) Comprova que s'han creat una interfície tipus **br** i dues interfícies tipus **veth**.

e) Comprova les adreces IP de les interfícies **eth0** del contenidors i la interfície tipus **br**.

f) A continuació, amb **docker compose** atura els contenidors i comprova que:

- Les interfícies **br** i **veth** han desaparegut
- Els ports **8080** i **33306** de la màquina host han desaparegut
- Els contenidors han desaparegut.
- La xarxa ha desaparegut

Lliurament de l'activitat

a) Comprovacions:

- Mostra que no hi ha cap imatge, xarxa, volum o contenidor a la màquina host.
- Amb **docker compose** crea imatges, contenidors, imatges i xarxes i mostra que s'han creat.
- Ports oberts a l'equip host.
- IPs de **br-<identificador>** del host i **eth0** dels contenidors.
- Amb **docker compose** atura els contenidors i demostra que ja no existeixen, i tampoc la xarxa, els ports del host i la interfície **br-<identificador>**.

b) Data límit per obtenir el **100%** de la nota: **dijous 20-11-24** a les **19.10**. Posteriorment la nota serà menor.