Pràctica 3: Treballant amb Apache2, Vagrant i contenidors Docker

1.- Objectiu de l'activitat

Els objectius objectius d'aquesta activitat són:

- Instal·lar **Docker** sobre una màquina virtual gestionada amb **Vagrant**
- Desplegar una aplicació PHP utilitzant Dockers. Això vol dir:
  - Crear una imatge
  - Crea un contenidor a partir de la imatge
  - Accedir a l'aplicació.

2.- Conceptes mínims sobre contenidors Docker

a) Avui dia hi ha 4 maneres típiques de desenvolupar i desplegar una aplicació i fer-la disponibles als usuaris finals. Podeu desplegar l'aplicació:

- Directament sobre l'equip físic.
- Sobre una màquina virtual.
- Utilitzant contenidors que s'executen sobre una màquina física
- Utilitzant contenidors que s'executen sobre una màquina virtual

Totes tenen avantatges i desavantatges però, la darrera opció ens proporciona una bona combinació de rapidesa de posada en marxa i execucio, portabilitat, eficiència en l'utilització dels recursos (CPU, RAM, espai de disc, xarxa,...), escalabilitat, aïllament (i per tant, menys risc de conflicte entre aplicacions) i seguretat.

b) Per poder treballar amb contenidors, ens cal un progamari gestor i motor de contenidors. Amb aquest programari és fàcil poder crear, posar en marxa, aturar, modificar, esborrar i utlitzar contenidors.

c) Un dels gestors i motors de contenidors més populars i més ampliament utilitzat del mercat és **Docker Engine** creat per la companyia **Docker, Inc**.

d) Alguns enllaços interessants sobre contenidors **Docker**:

- Avantatges d'un contenidor: https://www.youtube.com/watch?v=YFl2mCHdv24 (0:06 - 0:44)
- Què és un contenidor Docker?: https://www.youtube.com/watch?v=YFl2mCHdv24 (0:00 - 0:44)
- Conceptes bàsics: https://www.youtube.com/watch?v=YFl2mCHdv24 (1:57 - 2:16)
- Dockerfile i imàgens: https://www.youtube.com/watch?v=YFl2mCHdv24 (2:17 - 2:41)
- DockerHub --> https://www.youtube.com/watch?v=YFl2mCHdv24 (3:53 - 4:16)

e) Diferència entre una màquina virtual i un contenidor:

- Virtual Machine vs Docker --> https://www.youtube.com/watch?v=YFl2mCHdv24 (0:45 – 1:56)
- Virtual Machine and Container in a nutshell --> https://deshanigeethika.medium.com/docker-tutorial-a6aa5b41e3ff

3- Creació d'una màquina virtual amb Vagrant amb aprovisionament del programari per treballar amb contenidors Docker

a) Crea una carpeta de nom **sm8act03** dins de la teva màquina física . Dins de la carpeta **sm8act03** crea una altra carpeta de nom **projectes** i un altre de nom **vm**.

b) Dins de la carpeta de nom **vm** de la teva màquina física crea el següent fitxer **Vagrantfile**:

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

# VARIABLES

BOX_IMAGE = "xxxx"
PROVIDER = "xxxx"
NOM_MAQUINA_VIRTUAL = "xxxx"
NUM_CPUS = xxxx
MEMORIA_RAM = xxxx
HOSTNAME = "xxxx"
CARPETA_MAQ_FIS = "xxxx"
CARPETA_MAQ_VIR = "xxxx"
PORT_VIR1 = xxxx
PORT_FIS1 = xxxx
PORT_VIR2 = xxxx
PORT_FIS2 = xxxx
PROT = "xxxx"

# CONFIGURACIÓ DE LA MÀQUINA

Vagrant.configure("2") do |config|

    # BOX
    config.vm.box = BOX_IMAGE

    # CONFIGURACIÓ ESPECÍFICA DEL PROVIDER
    config.vm.provider PROVIDER do |provider|
        provider.name = NOM_MAQUINA_VIRTUAL
        provider.memory = MEMORIA_RAM
        provider.cpus = NUM_CPUS
        provider.customize ['modifyvm', :id, '--clipboard', 'bidirectional']
    end

    # CONFIGURACIÓ GENERAL
    config.vm.hostname = HOSTNAME
    config.vm.synced_folder CARPETA_MAQ_FIS,CARPETA_MAQ_VIR
    config.vm.network "forwarded_port", guest: PORT_VIR1, host: PORT_FIS1, protocol: PROT
    config.vm.network "forwarded_port", guest: PORT_VIR2, host: PORT_FIS2, protocol: PROT
    # PROGRAMARI A INSTAL·LAR I ORDRES A EXECUTAR DURANT LA CREACIÓ DE LA MÀQUINA
    config.vm.provision "shell", inline: <<-SHELL
        #
        # 1- Actualització de la llista de programari. Instal·lació del programari general
        #
        xxxxxxxxxxxxxxxxxxxxxx
        #
        # 2- Ordres d'instal·lació de Docker. Permís a l'usuari vagrant per utiitzar Docker sense sudo
        #
        xxxxxxxxxxxxxxxxxxxx
    SHELL

end
```

c) Fes les següents modificacions a les varibales del fitxer **Vagrantfile**:

- El Box utilitzat serà **debian/bookworm64**.
- El provider (software gestor de màquines virtuals) serà **virtualbox**.
- El nom de la màquina virtual dins del gestor serà a on **sm8act03_xxyyzz**. Recorda que **xxyyzz** són les 2 primeres lletres dels teus nom i cognoms.
- La quantitat de CPUs assignades a la màquina virtual serà de **3 CPUs**.
- La memòria RAM assignada a la màquina virtual serà de **3072MiB**.
- El nom de sistema de la màquina virtual serà **sm8act03-xxyyzz.fjeclot.local**.
- La carpeta de la màquina física compartida amb la màquina virtual serà **projectes**.
- La carpeta de la màquina virtual compartida amb la màquina física serà **/home/vagrant/projectes**.
- El port **80/tcp** de la màquina virtual s'exportarà al port **8080/tcp** de la màquina física.
- El port **443/tcp** de la màquina virtual s'exportarà al port **8443/tcp** de la màquina física.
- Abans de començar l'aprovisionament caldrà actualitzar la llista del programari disponible.
- El programari general aprovisionat serà: **aptitude, net-tools, whois i wget**
- Per poder aprovisionar el programari de contenidor **Docker** i permetre que l'usuari **vagrant** utilitzi **Docker** cal afegir les següents ordres dins del fitxer **Vagrantfile**:

```
#2.1- Programari necessari per poder instal·lar Docker
sudo apt-get -y install apt-transport-https ca-certificates curl gnupg
#
#2.2- Clau GPG (comprovació de signatura i integritat del programari descarregat)
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o \
/usr/share/keyrings/docker.gpg
#
#2.3- Afegint el dipòsit a sources.list
echo "deb [arch=$(dpkg -print-architecture) \
signed-by=/usr/share/keyrings/docker.gpg] \
https://download.docker.com/linux/debian bookworm stable" | sudo tee \
/etc/apt/sources.list.d/docker.list > /dev/null
#
#2.4- Descarregant el programari bàsic de Docker
sudo apt-get -y update
sudo apt-get install -y docker-ce docker-ce-cli containerd.io \
docker-compose-plugin
#
#2.5- Afegint l'usuari vagrant al grup docker
sudo gpasswd -a vagrant docker
```

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

a) Recorda que, per poder desplegar una aplicació i desplegar-la utilitzant contenidors Docker, cal seguir els següents passos:

1. Desenvolupament de l'aplicació (en PHP, Python, JavaScript, Java, C, Bash,.....).
2. Creació d'un fitxer Dockerfile.
3. Creació d'una imatge del contenidor amb el fitxer Dockerfile i l'aplicació.
4. Comprovació de que la imatge s'ha creat correctament.
5. Creació del contenidor a partir de la imatge creada.
6. Comprovació de que el contenidor s'ha creat correctament i està en execució.
7. Accés a l'aplicació per comprovar el seu funciomanent.

Ara seguirem aquests passos.

b) Dins de la **màquina física** accedeix a la carpeta **sm8act03/projectes** i crea una carpeta de nom **pizzeria**. Dins de **pizzeria** crear una carpet de nom **app**. Dins de la carpeta **app**, descarrega els 2 fitxers:

- **pizza.html** → https://raw.githubusercontent.com/dcolla2/pizzeria/refs/heads/main/projectes/pizzeria/app/pizza.html
- **pizza.php** → https://raw.githubusercontent.com/dcolla2/pizzeria/refs/heads/main/projectes/pizzeria/app/pizza.php

b) Dins de la carpeta **projectes** crea un fitxer de nom **Dockerfile** amb aquest continigut:

```
# Descarrega una imatge base de contenidor Docker amb Apache i PHP 8.4
FROM php:8.4-apache
# El directori de treball per defecte del contenidor serà /var/www/html
WORKDIR /var/www/html
# Copia el directori app dins del directori de treball per defecte
COPY app .
# Exposa el port 80/tcp a la màquina host
EXPOSE 80
```

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

- La carpeta **app** que té dins els fitxer **pizza.html** i **pizza.php**.
- El fitxer **Dockerfile** amb el contingut crear 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
```

e) 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** executant:

```
docker ps -a
```

f) Comprova que la màquina virtual té el port **80/tcp** obert pel contenidor **cpizza**. Executa des de la màquina virtual:

```
sudo netstat -atupn | grep docker
```

g) Comprova que la màquina física té el port **8080/tcp** obert pel programari **Vagrant**. Executa des de la màquina física i dins del directori a on es troba el fitxer **Vagrantfile**:

```
vagrant port
```

h) Accedix a l'aplicació utilitzant l'**adreça IP de la màquina física** i recordant que el port **80/tcp** de la màquina virtual està exportat al **8080/tcp** de la màquina física. Recorda també, que per aquesta aplicació, la seva pàgina inicial és **pizza.html**.

4- Control de l'aplicació

a) Dins de la màquina virtual, executa:

```
docker stop cpizza
```

i comprova que l'aplicació deixa de funcionar.

b) Dins de la màquina virtual, executa:

docker ps -a

i comprova que l'estat del contenidor és **`Exited`**.

c) Dins de la màquina virtual, executa:

docker start cpizza

i comprova que l'aplicació torna a funcionar.

d) Dins de la màquina virtual, executa:

docker ps -a

i comprova que l'estat del contenidor és **`Up`**.

Lliurament de l'activitat

a) Comprovacions:
- Mostra la imatge de contenidor creada dins de la màquina virtual.
- Posa en marxa el contenidor i comprova que el seu estat és **Up**.
- Accedeix a l'aplicació des del navegador de la màquina física i comprova que funciona.
- Atura el contenidor i comprova que el seu estat és **Exited**.
- Comprova ara que l'aplicació no funciona des del navegador de la màquina física.

b) Data de lliurament: A partir del **dijous 4-12-25** a les **16.50**.