Fase 4 - Activitat 5.4: Desplegament d'una aplicació utilitzant contenidors Docker.

0- Identificació del grup i activitat:

Curs: ASIX2
Projecte: GP2 DevOps i Cloud Computing
Fase: 4
Activitat: 5.4
Grup:
Membres:

1- Introducció i objectius de l'activitat 5.4

1.1- Objectius de l'activitat

- Fer una breu introducció a la tecnologia de contenidors i Docker
- Treballar amb contenidors Docker
- Desplegar i accedir a una aplicació PHP utilitzant contenidors Docker

1.2- Imatges i contenidors. Dockers

a) Una **Imatge de contenidor**:

- És una manera lleugera d'empaquetar una aplicació.
- Dins d'una imatge de contenidor empaquetarem únicament el codi, eines de sistema, configuracions, biblioteques i dependències necessàries per fer anar un aplicació.
- És fàcilment portable d'un entorn informàtic a un altre i per tant, amb indepèndencia del sistema operatiu i maquinari amb el qual treballem, l'aplicació funcionarà sempre.
- Ha de ser posada en marxa o executada per poder accedir a l'aplicació. Tenir la imatge descarregada no és suficient per accedir a l'aplicació.
- Pot ser executada multiples vegades sobre un mateix entorn informàtic.
- Ens permet distribuir una aplicació de manera fiable perquè s'executarà sempre de la mateixa manera sempre (no importa quantes vegades s'executi o l'entorn informàtic sobre el qual s'executi).

b) Un **contenidor**:

- És una imatge de contenidor en **execució**.
- Permet accedir a l'aplicació empaquetada dins de la imatge del contenidor.
- Només es pot crear a partir d'una imatge. Sense una imatge no podem crear un contenidor.
- Es poden crear multiples contenidors a partir d'una única imatge.
- Ailla el funcionament de l'aplicació del seu entorn informàtic (sistema operatiu, programari, maquinari, etc..). Això vol dir que l'aplicació dins del contenidor funciona sempre igual amb independència de l'entorn informàtic sobre el qual s'executa.
- Ailla una aplicació d'altres aplicacions perquè cadascuna té el seu propi codi, biblioteques, dependencies, eines i configuracions separades de les d'altres aplicacions.
- Es pot executar sobre equips físics, màquines virtuals o des del cloud (núvol) i sempre funciona de la mateixa manera.
- Necessita pel seu funcionament:
  - Codi, eines de sistema, configuracions, biblioteques i dependències empaquetades (imatge).
  - Els recursos de l'entorn informàtic (sistema operatiu, programari, maquinari, etc..) sobre el qual s'executa.
  - Una eina de gestió, administració i monitorització de contenidors instal·lada dins de l'entorn informàtic sobre el qual s'executa.

c) Per desplegar una aplicació també podem utiltzar **màquines virtual**. L'avantatge de treballar amb **màquines virtual** és que permet utilitzar sistemes operatius diversos però fa que el desplegament d'aplicacions sigui més lent i utilitza més recursos del sistema físic. En canvi un **contenidor** és més lleuger i

ràpid però no és tan flexible perquè dins d'un contenidor només pot 'executar-se un sistema que normalment és un Linux mínim.

d) Respecte de **Docker**:

- És una tecnologia dissenyada per ajudar els administradors i desenvolupador a fàcilment crear, empaquetar, testejar, compartir, desplegar i executar aplicacions per mitja de l'ús d'imatges de contenidors i contenidors.
- Per poder utilitzar aquesta tecnologia ens cal instal·lar sore el nostre sistema el programari Docker Engine. Aquest programari consta bàsicament de:
  - Un servidor **dockerd** que serà un procés que s'executarà en 2n terme de manera permanent des de l'arrancada del sistema excepte que ho aturem o deshabilitem
  - Un client de nom **docker** per interactuar amb el servidor i poder crear imatges, esborrar imatges, modificar imatges, visualitzar imatges, crear contenidors, esborrar contenidors, modificar contenidors, visualitzar contenidors, etc, etc….
- Docker Engine està disponible per Linux, Windows i Mac.
- Per les seves caràcterístiques permet:
  - Reduir dràsticament el temps de desenvolupament del codi.
  - Reduir dràsticament el temps d'implantanció del codi en producció.
  - Reduir dràsticament el retard entre la finalització del codi i la seva implantanció en producció.
  - Executar contenidors en multiples entorns informàtics: màquines físiques, virtuals, cloud,…
  - Compartir aplicacions de manera ràpida, portable, fiable i econòmica.
  - Tenir una alternativa ràpida i eficient a treballar amb màquines virtuals per desplegar aplicacions.

e) Per crear un **contenidor Docker** amb una aplicació dockeritzada - és a dir, una aplicació desplegada sobre el contenidor i a la qual es pot accedir quan posem en marxa el contenidor- cal seguir els següents passos en general:

- Creació d'un projecte amb l'aplicació
- Creació d'un fitxer Dockerfile amb la configuració de la imatge de contenidor que volem crear.
- Creació d'una imatge de contenidor a partir del fitxer Dockerfile.
- Creació i posada en marxa d'un contenidor a partir de la imatge.
- Accés a l'apliació desplegada sobre el contenidor

f) Un fitxer **Dockerfile**:

- Docker pot crear imatges automàticament llegint les instruccions d'un Dockerfile.
- Un Dockerfile és un document de text que conté les ordres necessàries per muntar (crear) una imatge de contenidor utilitzant Docker.
- Dins d'un fitxer Dockerfile trobarem indicacions sobre configuracions, dependències, biblioteques, sistema operatiu base, eines de sistema que li cal a l'apliacació per funcionar.
- Dins d'un fitxer Dockerfile trobarem indicacions sobre a on es troba l'aplicació i com està desplegada.

g) Què és un **Docker Registry** i **Docker Hub**?:

- Un registre de Docker Registry és un dipòsit que emmagatzema imatges de Docker i les fa disponibles per mitjà d'una xarxa local o per internet. Un dipòsit pot ser públic o privat. Si és públic tothom pot accedir a les imatges del dipòsit i si es privat només ho podem fer els usuaris amb permisos.
- **Docker Hub** és el repositori d'imatges (docker registry) per a contenidors públic més gran del món que allotja imatges oficials de grans empres (com Oracle), mitjanes i petites empreses, grans projectes tipus open source (MySQL,etc..) i de petits projectes personals que existeixen avui dia.
- Dins de **Docker Hub** també es poden crear dipòsits privats.
- **Docker Hub** és el repositori (docker registry) establert per defecte de la plataforma de **Docker Engine**.

g) Una nota final important. **Docker** pot ser instal·lat sobre:

- Una màquina física de tipus Windows, Linux o Mac.
- Una màquina virtual de tipus Windows, Linux o Mac.
- Una màquina en el núvol amb AWS, Azure, Red Hat, etc..

Per evitar problemes perquè cadascú fa servir una edició o versió de Windows diferents, o versions diferents de diverses distribucions de Linux, o versions de l'IOS de Mac diferents, instal·larem Dockers sobre màquines virtuals gestionades amb Vagrantfile

h) Documentació interessant:
- What is Docker? --> https://www.youtube.com/watch?v=YFl2mCHdv24 From 0:00 to 0:44
- Virtual Machine vs Docker --> https://www.youtube.com/watch?v=YFl2mCHdv24 From 0:45 to 1:56
- Container, instance, image, build and run --> https://www.youtube.com/watch?v=YFl2mCHdv24 From 1:57 to 2:16
- Dockerfile and image --> https://www.youtube.com/watch?v=YFl2mCHdv24 From 2:17 to 2:41
- DockerHub --> https://www.youtube.com/watch?v=YFl2mCHdv24 From 3:53 to 4:16
- Virtual Machine and Container in a nutshell --> https://deshanigeethika.medium.com/docker-tutorial-a6aa5b41e3ff

2- Creant una màquina virtual base amb el programari Docker instal·lat utilitzant Vagrant

a) Crea a la teva màquina física un directori de nom **gp1f4a5.4**. A continuació, dins de **gp1f4a5.4** s'hauran de crear 2 directoris. Un directori de nom **codi_projectes** i un altre de nom **vm**.

b) Dins del directori **vm** crearem tres fitxers:
- **Vagrantfile** → Per crear una màquina virtual amb Vagrant amb unes característiques que es donaran al següent apartat
- **README** amb:
  - Instruccions de com utilitzar el fitxer Vagrantfile per iniciar la màquina virtual, accedir-hi, sortir, aturar-la i esborrar-la.
  - Indicacions del lloc a on es troben els fitxers amb els codis php del projecte de manera que es pugin veure i modificar.
  - Indicacions de com accedir a l'aplicació web per veure el seu funcionament.
- **LICENSE** → Amb la llicència (de tipu MIT) posant en la part de Copyright el vostre nom i adreça de correu del grup i l'any actual.

c) El fitxer **Vagrantfile** ha de tenir les següents característiques:
- Ha d'utilitzar el Box **debian/bullseye64** (és la versió **11** de **debian**)
- El proveïdor ha de ser **VirtualBox**.
- El nom de sistema del host ha de ser **grupXX.fjeclot.net** a on **grupXX** a **XX** representa el vostre número de grup.
- El nom de la màquina vist des del gestor de VirtualBox serà a on **grupXX** a **XX** representa el vostre número de grup.
- Utilitza **2 CPUs** i **2048MiB** de memòria.
- Que la màquina s'afegeixi al grup **ASIX2** de **VirtualBox**.
- Afegeix una interfície de xarxa treballant amb **adaptador pont** que té una IP assignada via DHCP.
- Export el port **80/tcp** de la màquina virtual al **8000/tcp** de la màquina física.
- S'ha de compartir (sincronitzar) la carpeta **codi_projectes** de l'equip físic amb la carpeta **/home/vagrant/projectes** de la màquina virtual (que es crea automàticament).
- S'ha de fer un **update** del llistat de paquets de programari disponibles.
- Instal·la els paquets: **aptitude** i **net-tools**.

- Instal·la **Dockers**. Afegeix a la secció de **provision**:
  - **sudo apt-get install -y apt-transport-https ca-certificates curl gnupg2 software-properties-common**
  - **sudo curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add**
  - **sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian $(lsb_release -cs) stable"**
  - **sudo apt-get -y update**
  - **sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-compose**

d) Inicia la màquina virtual i accedeix al seu sistema. Comprova que tens el programari de Docker instal·lat. Executa:

docker -v

i el resultat hauria de ser:

```
vagrant@grup00:~$ docker -v
Docker version 24.0.6, build ed223bc
```

3- Creant un contenidor Docker amb amb l'aplicació desplegada

a) Dins de la màquina virtual, a la teva carpeta personal crea un directori de nom **projectes** si encara no existeix, tot i que ja hauria d'existir. Canvia al directori **projectes** i a continuació:

1. Assegura't que l'usuari vagrant és membre del grup docker. Executa:

   sudo gpasswd -a vagrant docker
   exit

   Torna a accedir a la màquina virtual. Comprova amb l'ordre **id** que **vagrant** és membre de **docker**.

2. Descarrega un paquet a on hi ha una aplicació web amb llenguatge PHP de nom **ppv**. Executa l'ordre:

   wget http://www.collados.org/asix2/m14/GP1/ppv.tar.gz

3. Descomprimeix i desempaqueta l'aplicació. Executa l'ordre:

   gzip -d ppv.tar.gz
   tar xf ppv.tar

4. Esborra (si existeix) **ppv.tar**. Executa l'ordre:

   rm ppv.tar

5. Comprova que s'ha creat un directori de nom ppv. Entra dins del directori i comprova que hi ha 2 directoris:
   - app
   - Dockerfile

6. Entra dins del directori **app**. Edita **index.html**. Canvia la cadena **xxyyzz** de la línia **34** per **grupXX** a on **XX** és el vostre número de grup.

7. Canvia al directori **ppv** novament. Edita el fitxer **Dockerfile**. Afegeix els següent continguts al fitxer:

```
FROM php:7.4-apache
COPY app /var/www/html
EXPOSE 80
```

8. Per muntar (és a dir, crear) una **imatge de contenidor** amb el codi de l'apliació web i totes les biblioteques, dependencies, eines de sistema, sistema bàsic i configuracions necessàries per fer anar l'apalicació, executa (recorda que XX és el teu número de grup):

docker build -t ppv_imatge_grupXX:1.0 .

NOTA: El caràcter **.** al final de l'ordre és important i forma part de l'ordre perquè indica el directori a es crea la imatge.

Aquesta ordre munta una imatge de contenidor que:
- S'anomena **ppv_image_grupXX**, a on **XX** és el vostre número de grup.
- Utilitza una imatge ja existent a **Docker Hub** de nom **php:7.4-apache** que té un sistema **Linux** mínim, **Apache2** i **PHP7.4** com a base per crear una imatge nova.
- Compartirà el directori **/var/www/html** del contenidor amb la carpeta **app** de la màquina virtual.
- Exposarà el port **80/tcp** del contenidor i per tant, es podrà accedir al port **80/tcp** del contenidor **Docker** des de la màquina virtual.
- La imatge serà desada com la versió **1.0**.

9. Executa en el teu sistema:

docker images

i comprova que la imatge s'ha creat correctament.

10. Executa ara l'ordre:

docker run --name ppv_container_grupXX -i -t -d -p 80:80 ppv_imatge_grupXX:1.0

Aquesta posa en marxa un contenidor:
- S'anomena **ppv_container_grupXX** (a on **XX** és el vostre número de grup).
- Creat a partir de la imatge **ppv_imatge_grupXX:1.0** (a on **XX** és el vostre número de grup).
- **Exposa** el port **80/tcp** del **contenidor** com a port **80/tcp** de la **màquina virtual** => Tota connexió al port **80/tcp** de la **màquina virtual** es rederigirà al port **80/tcp** del **contenidor**.
- Amb **-d** el contenidor s'executa en 2n terme.
- Les opcions **-i -t** serveixen bàsicament per poder accedir via bash al contenidor.

NOTA: Si el contenidor ja existeix i està aturat només cal executar la següent ordre per posar-lo en marxa: **docker start ppv_container_grupXX** (a on **XX** és el vostre número de grup).

11. Dins de la màquina virtual comprova que el procés **docker-proxy** ha obert el port **80/tcp** amb l'ordre: **sudo netstat -atupn**

b) Si tot s'ha fet correctament, ara existiran les següents relacions:

- Redireccions de ports:
  - Port **8000/tcp** de la màquina **física** → Port **80/tcp** de la màquina **virtual**
  - Port **80/tcp** de la màquina **virtual** → Port **80/tcp** del **contenidor**
  - Per tant **8000/tcp** de la màquina **física** → Port **80/tcp** del **contenidor**
- Compartició de carpetes:
  - **codis_projectes** de l'equip **físic** → **/home/vagrant/projectes** de l'equip **virtual**
  - **/home/vagrant/projectes/app** de l'equip **virtual** → **/var/www/html** del **contenidor**
  - Per tant, **codis_projectes/ppv/app** de l'equip físic → **/var/www/html** del **contenidor**.

c) Accedeix a l'aplicació i comprova que funciona. Obre un navegador de l'equip físic i accedeix a la següent URL: **http://localhost:8000**.

d) Comprova que pots aturar l'aplicació des de dins de la màquina virtual executant:

```
docker stop  ppv_container_grupXX
```

i tornar a iniciar-la amb:

```
docker start ppv_container_grupXX
```

a on a on **XX** és el vostre número de grup.

4- Creant un dipòsit a Github per pujar els fitxers de l'activitat gp1f4a5.4

a) Si tot funciona correctament, dins la carpeta **gp1f4a5.4** de la teva màquina físic crea un fitxer de nom **.gitignore** amb un editor de texte. Dins del fitxer escriu:

```
.gitignore
vm/.vagrant
```

Aquests és la llista dels fitxers que seran ignorats (i per tant no seran pujats al dipòsit de control de versions local) quan fem un **add** i un **commit** amb l'ordre **git**.

b) Dins de a carpeta **gp1f4a5.4** executa:

```
git init
git add *
git commit -m "Commit 1 - grupXX - Activitat gp1f4a5.4"
```

a on **XX** és el vostre número de grup.

Comprova que els fitxers pujats són: **Dockerfile**, **index.html**, **ppv.php**, **LICENSE**, **README** i **Vagrantfile**.

c) A continuació, crea un dins de **Github** un dipòsit **públic** i de nom **gp1f4a5.4** i puja els continguts del dipòsit local del teu projecte al dipòsit remot. Comprova que s'han pujat els continguts al dipòsit remot.

d) Convida als teus companys de grup per poder treballar com a col·laboradors del projecte. Comprova que els teus companys han acceptat la invitació i podem col·laborar amb tu en el teu projecte. La invitació es pot acceptar via el correu rebut amb la invitació o anant a Notificacions del teu compte de Github.

Lliurament de l'activitat

a) Envia un correu de **cf@collados.org** amb:
- Assumpte: **asix2_grupXX_gp1f1a5.4** (a on **XX** és el vostre número de grup).
- L'adreça URL del dipòsit de Github d'un membre del grup amb els fitxers del projecte.

b) Comprovació del funcionament de l'aplicació:
- Accés a la màquina virtual **grupXX** des de la màquina física d'un membre del grup.
- Comprovació que el funcionament del contenidor docker **ppv_container_grupXX** dins de la màquina virtual.
- Accés a l'aplicació executant-se en el docker des de la màquina física amb el navegador i utilitzant **http://localhost:8000**.
- Comprovació que l'aplicació funciona.
- Comprovació que s'ha pujat a Github els fitxers de l'aplicació i que s'han convidat els altres membres del grup a col·laborar-hi.

c) Data límit: **Dilluns 10-10-23** a les **20.05**. Fora d'aquest límit, la pràctica tindrà una menor valoració.