

Pràctica 3a: Docker containers i Apache Tomcat 9

1- Documentació

1.1- Servidors d'aplicacions, contenidors web, Tomcat i paquets war

- a) [Diferència ràpida entre servidor web, contenidor web i servidor d'aplicacions](#)
- b) [Altres diferències entre servidor web, contenidor web i servidor d'aplicacions](#)
- c) [Tomcat vs WildFly vs Glassfish](#)
- c) [Diferència entre servidor web, contenidor web i servidor d'aplicacions. Casos d'utilització](#)
- d) Conceptes complementaris:
 - [Servlets](#)
 - [JSP](#)
 - [JEE\(J2EE\)](#)
- e) [Introducció a Apache Tomcat](#)
- f) [Fitxers \(paquets\) WAR](#)

1.2- Contenedors. Docker containers

- a) [Introducció al concepte de contenidor](#)
- b) [Contenedors Docker](#)
- c) Mira d'aquest [video de youtube](#) des de 0:00 a 2:42
- d) Documentació oficial de Docker: <https://docs.docker.com/get-started/>
- e) Sobre Docker Hub: <https://docs.docker.com/docker-hub/>
- f) CLI de Docker: <https://docs.docker.com/engine/reference/commandline/run/>
- g) Docker - Child commands: <https://docs.docker.com/engine/reference/commandline/docker/>
- h) Llegeix d'aquest [enllaç](#) la secció Comparing Docker containers with virtual machines.
- i) Imatge oficial Docker de Tomcat: https://hub.docker.com/_/tomcat

2- Instal·lació de Docker

2.1- Instal·lació de Docker sobre Debian Linux. Executa com a root o amb sudo:

apt-get update

apt-get install apt-transport-https ca-certificates curl gnupg2 software-properties-common

curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -

add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian \$(lsb_release -cs) stable"

apt-get update

apt-get install docker-ce docker-ce-cli containerd.io docker-compose

gpsswd -a \$USER docker (Fes log out i log in perquè l'ordre tingui efecte)

Per més informació sobre la instal·lació de Docker sobre Debian Linux:

<https://docs.docker.com/install/linux/docker-ce/debian/>

<https://docs.docker.com/install/linux/linux-postinstall/#manage-docker-as-a-non-root-user>

2.2- Instal·lació de Docker sobre Windows

* Informació: <https://docs.docker.com/docker-for-windows/install/>

* Video de youtube: <https://www.youtube.com/watch?v=GIMExUnjzMw>

* Mira la secció System Requirements a l'apartat What to know before you install.

* Crea un compte de Docker Hub a <https://hub.docker.com/>

* Descarrega Docker for Windows Installer.exe des d'[aquí](#)

* Executa **Docker for Windows Installer.exe** i segueix les instruccions. Es generarà una icona a l'escriptori i una icona de Docker a la barra d'estat quan s'executa.

3- Descarregant i instanciant una imatge de Docker amb Apache Tomcat 9

a) Descarrega la imatge d'Apache Tomcat 9 dels dipòsits oficials de Docker Hub. Executa:

```
docker pull tomcat:9.0
```

b) Comprova que tens la imatge descarregada. Executa:

```
docker images
```

c) Inspecciona les característiques de la imatge i comprova que exposa el port 8080/tcp. Executa:

```
docker inspect --format="{{json .Config.ExposedPorts }}" tomcat:9.0
```

d) Instancia la imatge tomcat:9.0 i crea un contenidor de nom m08uf1pr3a. Amb l'opció -p exposa el port 8080/tcp del contenidor al port 8888/tcp del teu equip host. Amb l'opció -d indica que s'executi en 2n terme. Executa:

```
docker run --name m08uf1pr3a -d -p 8888:8080 tomcat:9.0
```

e) Comprova que el contenidor s'ha instanciat i està actiu (up), que exposa el seu port 8080/tcp per mitjà del port 8888/tcp de la màquina host i que el seu nom és correcte. Executa:

```
docker ps -a
```

Comprova també el valor del CONTAINER ID.

f) Comprova dins de la màquina host que el port 8888/tcp està obert i associat a l'aplicació docker-proxy.

g) Comprova que pots accedir a l'interpret d'ordres bash del contenidor via terminal executant:

```
docker exec -it m08uf1pr3a bash
```

i per sortir del terminal del contenidor executa:

```
exit
```

4- Configuració inicial d'Apache Tomcat 9 per accedir al gestor d'aplicacions

a) Torna a accedir al contenidor i instal·la nano. Executa des de dins del contenidor:

```
apt-get update  
apt-get install nano
```

b) Crea un usuari d'Apache Tomcat 9 de nom tomcat amb permis d'accés al gestor d'aplicacions, amb contrasenya FjeClot20# i que tingui el role de manager-gui per poder administrar aplicacions web desplegades amb Tomcat. Realitza els següents passos:

* Obre el fitxer /usr/local/tomcat/conf/tomcat-users.xml

* A la secció <tomcat-users> afegeix les etiquetes:

```
<role rolename="manager-gui"/>
```

```
<role rolename="manager-script"/>
```

```
<user username="tomcat" password="FjeClot20#" roles="manager-gui,manager-script"/>
```

c) Per poder fer accessible el gestor d'aplicacions de Tomcat9, mou tots els fitxers i directoris de /usr/local/tomcat/webapps.dist a usr/local/tomcat/webapps. Executa:

```
cd usr/local/tomcat/webapps.dist  
mv * ../webapps
```

d) Configura el gestor d'aplicacions (que és una aplicació web) de manera que accepti connexions des de fora de la màquina o contenidor en la qual s'executa. Executa:

```
cd usr/local/tomcat/webapps/manager/META-INF/
```

i amb **nano** obre el fitxer **context.xml** i assegurat que l'etiqueta **Valve** està comentada:

```
<!--  
<Valve className="org.apache.catalina.valves.RemoteAddrValve"  
    allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" />  
-->
```

e) Surt del contenidor i reinicia'l. Executa:

Des del contenidor → **exit**

Des del host → **docker restart m08uf1pr3a**

f) Des de la màquina host accedeix al **gestor d'aplicacions web de Tomcat9** (en anglès Tomcat Web Application Manager) amb un navegador utilitzant l'adreça **http://localhost:8888/manager/html**. Et demanarà l'usuari i la contrasenya de l'apartat b).

5- Gestió bàsica d'aplicacions web amb Apache Tomcat 9

a) Accedeix al gestor d'aplicacions web de Tomcat d'**Apache Tomcat 9** des de la teva màquina host.

b) Desplegament d'una aplicació:

a) Des de la teva màquina física descarrega [m08uf1ex1.war](#)

b) A la secció **Deploy** → **WAR file to deploy** selecciona l'aplicació **m08uf1ex1.war** dins de la teva màquina física, i a continuació desplega l'aplicació dins del contenidor web **Apache Tomcat 9**.

c) Comprova que **m08uf1ex1** es troba a la llista **Applications d'Apache Tomcat 9**.

c) Des de l'eina web de gestió d'aplicacions accedeix a l'aplicació desplegada. Comprova que s'hi pot accedir i comprova la seva URI.

d) Des de la màquina física, connectat directament a l'adreça URI trobada a l'apartat anterior. Comprova que s'hi pot accedir. Accedeix a la part protegida de l'aplicació amb el nom d'usuari **daw2** i la contrasenya **fjeclot**.

e) Des de l'eina d'administració d'apliacions atura l'aplicació **m08uf1ex1**. Recarrega la pàgina i comprova que passa.

f) Des de l'eina d'administració d'apliacions torna a posar en marca l'aplicació **m08uf1ex1**. Recarrega la pàgina i comprova que passa.

g) Des de l'eina d'administració d'apliacions elimina (undeploy) l'aplicació web **m08uf1ex1**. Comprova que **m08uf1ex1** ha desaparegut de la llista Applications de la web de gestió d'aplicacions.

6- Creació d'un fitxer .war a partir del codi font

a) Dins del contenidor crear **HelloWorld** dins de **/usr/local**. Dins de la carpeta, descarrega el fitxer [HelloWorld.targ.gz](#). Descomprimeix el fitxer descarregat i trobaràs els fitxers **HelloWorld.java**, **web.xml** i **index.jsp**. Esborra el fitxer descarregat. Modifica el fitxer **index.jsp** perquè a la secció **Autor** surti el teu nom i cognoms.

b) Compila **HelloWorld.java** amb l'ordre:

```
javac -classpath /usr/local/tomcat/lib/servlet-api.jar HelloWorld.java
```

Comprova que s'ha creat el fitxer **HelloWorld.class**.

c) Crea un directori de nom **WEB-INF**. Dins de **WEB-INF** crea una carpeta de nom **classes**. Mou **HelloWorld.class** a la carpeta **classes**. Mou **web.xml** dins de **WEB-INF**. Us ha de quedar un directori així:

```
dacomo@daw2m08:~/temp/HelloWorld$ ls
HelloWorld.java index.jsp WEB-INF
dacomo@daw2m08:~/temp/HelloWorld$ cd WEB-INF/
dacomo@daw2m08:~/temp/HelloWorld/WEB-INF$ ls
classes web.xml
dacomo@daw2m08:~/temp/HelloWorld/WEB-INF$ cd classes/
dacomo@daw2m08:~/temp/HelloWorld/WEB-INF/classes$ ls
HelloWorld.class
dacomo@daw2m08:~/temp/HelloWorld/WEB-INF/classes$ █
```

d) Dins del directori **HelloWorld** executa:

```
jar -cvf HelloWorld.war index.jsp WEB-INF
```

i comprova que es crea el paquet **HelloWorld.war**.

e) Desplega **HelloWorld.war** sobre **Apache Tomcat 9**. Simplement copia **HelloWorld.war** dins de **/usr/local/tomcat/webapps**. Refresca el gestor d'aplicacions. Comprova que pots veure i controlar l'aplicació **HelloWorld** des d'**Apache Tomcat 9**, que pots accedir a l'aplicació i que funciona.

Forma de lliurament de la pràctica

1- **Data de comprovació de la pràctica els dies 24-11-20**

2- **Comprovació de la pràctica:**

a) Comprovació que el contenidor Docker **m08uf1pr3a** està en execució i que el seu port **8080/tcp** està associat al port **8888/tcp** de la màquina host.

b) Comprovació d'accés al **gestor d'aplicacions de Tomcat 9** des de la màquina **host** amb un navegador.

c) Comprovació que l'aplicació **m08uf1ex1** s'ha desplegat sobre **Apache Tomcat 9**. Comprovació que l'aplicació funciona correctament des del navegador de la màquina host.

d) Comprovació que el paquet **HelloWorld.war** s'ha creat i s'ha desplegat sobre **Apache Tomcat 9**. Comprovació que l'aplicació funciona correctament des del navegador utilitzant la seva URI.

ANNEX

a) Un cop lliurada la pràctica pots aturar el docker, esborrar la instància i la imatge executant:

```
docker stop m08uf1pr3a
docker rm m08uf1pr3a
docker rmi tomcat:9.0
```

b) Per assegurar-te de netejar completament qualsevol objecte remanent que hagi pogut quedar dels contenidors i imatges creats pots executar: **docker system prune -a**