

Llançament amb docker run

Utilitzarem dos contenidors, un per al servidor Tomcat i l'altre per al proxy invers nginx.

1. La imatge que utilitzarem per al servidor Tomcat és tomcat:9.0.
2. La imatge que utilitzarem per al proxy invers és nginx.
3. Cal que els dos contenidors estiguen en la mateixa xarxa.

Creem una xarxa que serà la que utilitzaran els dos contenidors

```
osboxes@osboxes:~/Examen_PPS$ docker network create dockernetworkrun
08e163e8406bce3139df4dcce29d427eeb80acc3d9547347f3f33c6b5ed8a2c7
```

4. Cal ubicar el fitxer, sample.war, en el directori, /usr/local/tomcat/webapps, del servidor Tomcat utilitzant volums.

Amb la opció -v

/home/osboxes/Examen_PPS/sample.war:/usr/local/tomcat/webapps/sample.war

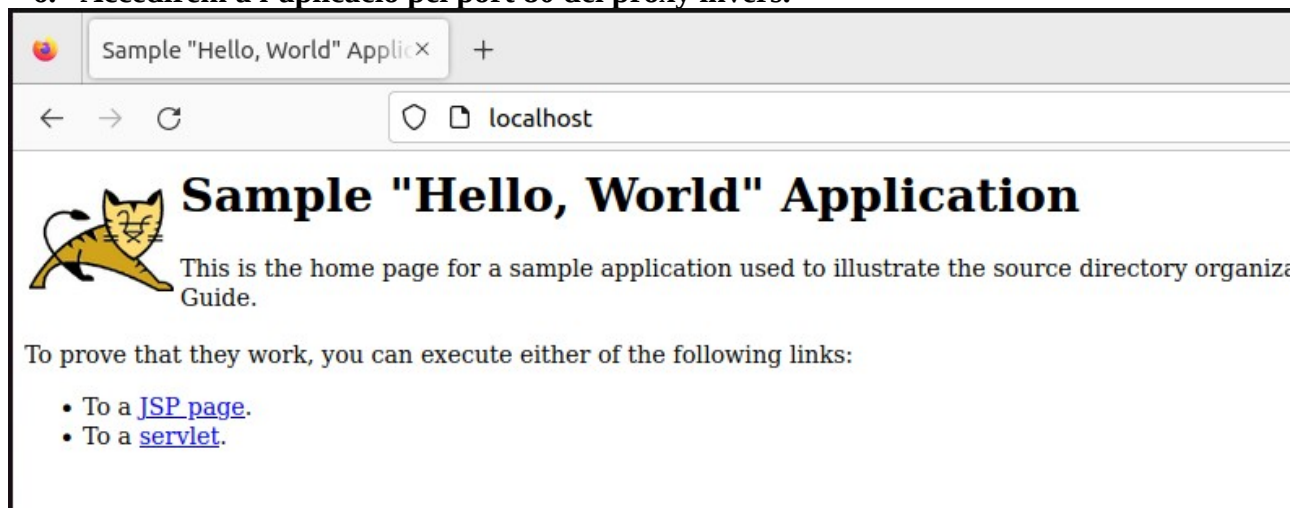
Anem a copiar el fitxer sample.war del nostre equip local a la carpeta del tomcat

```
osboxes@osboxes:~/Examen_PPS$ docker run -d --name aplicacionjava --network dockernetworkrun -v /home/osboxes/Examen_PPS/sample.war:/usr/local/tomcat/webapps/sample.war tomcat:9.0
50e9528ba3a11b9ded08cf5e6f5d7d11aead9f820b21940cbb40ffc94cf3466e
```

5. Cal ubicar el fitxer, default.conf, en el directori, /etc/nginx/conf.d, del proxy invers nginx. El mateix que el punt anterior però amb el default.conf

```
osboxes@osboxes:~/Examen_PPS$ docker run -d --name inversproxy --network dockernetworkrun -v /home/osboxes/Examen_PPS/default.conf:/etc/nginx/conf.d/default.conf -p 80:80 nginx
f8c98ea3861e6ccce7690d2e70c742417267f19a163ea87f2737676dc69a87f4
```

6. Accedirem a l'aplicació pel port 80 del proxy invers.



7. Els noms del contenidor proxy i de la xarxa es deixa a elecció de l'alumnat, el contenidor Tomcat
8. tindrà com a nom: aplicacionjava.
9. Els contenidors s'executaràn en segon pla. Això se aconsegueix amb la opció -d

Creació dels DockerFile

1. Creació del DockerFile de Tomcat

```
GNU nano 6.2 DockerFile *
# Utilitzem la imatge de tomcat:9.0
FROM tomcat:9.0

# Copiem el fitxer sample.war al directori corresponent
COPY sample.war /usr/local/tomcat/webapps
```

```
osboxes@osboxes:~/Examen_PPS$ docker build -t imatgetomcat .
Sending build context to Docker daemon 417.3kB
Step 1/2 : FROM tomcat:9.0
--> b07e16b11088
Step 2/2 : COPY sample.war /usr/local/tomcat/webapps
--> 801cfe0aa629
Successfully built 801cfe0aa629
Successfully tagged imatgetomcat:latest
```

Creem la imatge a partir del Dockerfile

2. Creació del DockerFile nginx

```
GNU nano 6.2 Dockerfile
# Utilitzem la
FROM nginx

# Copiem el fitxer default.conf dins del directori
COPY default.conf /etc/nginx/conf.d/
```

```
osboxes@osboxes:~/Examen_PPS$ docker build -t imatgenginx .
Sending build context to Docker daemon 417.3kB
Step 1/2 : FROM nginx
--> a99a39d070bf
Step 2/2 : COPY default.conf /etc/nginx/conf.d/
--> e17495da0a80
Successfully built e17495da0a80
Successfully tagged imatgenginx:latest
```

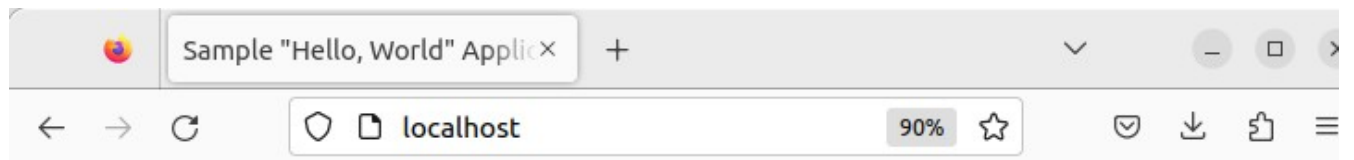
3. Llançament del contenidor de Tomcat

```
osboxes@osboxes:~/Examen_PPS$ docker run -d --name aplicacionjava --network dockernetworkrun -p 8080:8080 imatgetomcat
a4adda3f4009dfc5af9fb975c13bd5f17c7b286429190cda9a7d2a576bac9baf
```

4. Llançament del contenidor de nginx

```
osboxes@osboxes:~/Examen_PPS$ docker run -d --name inversa --network dockernetwo  
rkrun -p 80:80 imatgenginx  
29562c3e5fb782c11ac8b9980500b6139b7e2187a38b7b1f0580255e05b9a29c  
osboxes@osboxes:~/Examen_PPS$
```

5. Comprovació funcionament de l'aplicació



Sample "Hello, World" Application

This is the home page for a sample application used to illustrate the source directory organization of a web application utilizing the principles outlined in the Application Developer's Guide.

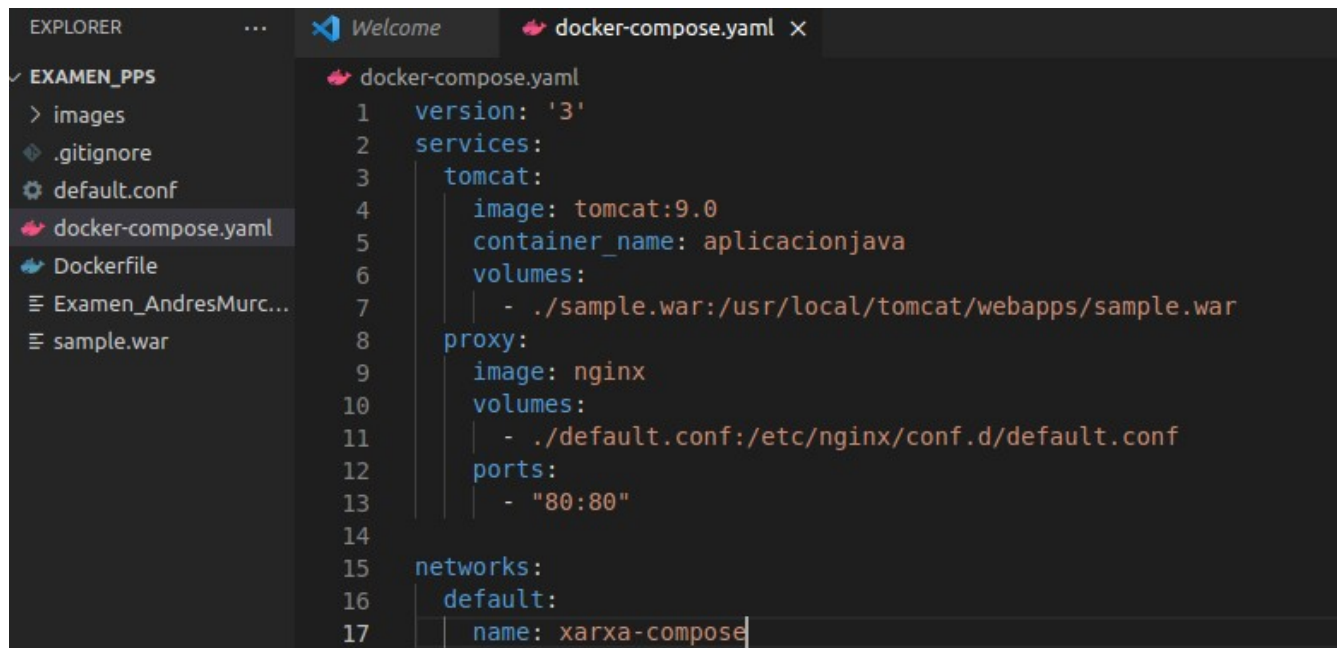
To prove that they work, you can execute either of the following links:

- To a [JSP page](#).
- To a [servlet](#).

--

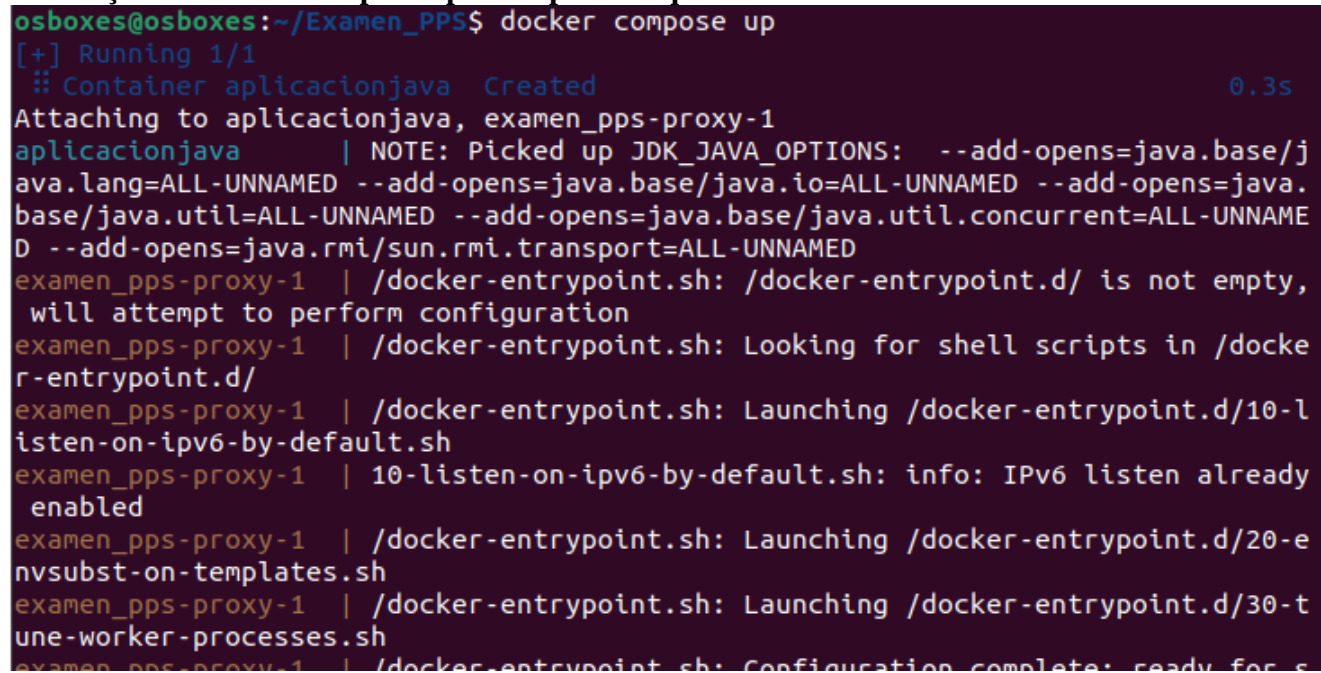
Creació del docker compose

1. Creem el docker-compose.yaml

A screenshot of a code editor (VS Code) showing the content of a file named 'docker-compose.yaml'. The Explorer sidebar on the left shows a project structure with folders like 'images', files like '.gitignore', 'default.conf', 'Dockerfile', and 'sample.war'. The main editor area displays the following YAML configuration:

```
1 version: '3'
2 services:
3   tomcat:
4     image: tomcat:9.0
5     container_name: aplicacionjava
6     volumes:
7       - ./sample.war:/usr/local/tomcat/webapps/sample.war
8   proxy:
9     image: nginx
10    volumes:
11      - ./default.conf:/etc/nginx/conf.d/default.conf
12    ports:
13      - "80:80"
14
15 networks:
16   default:
17     name: xarxa-compose
```

2. Llançem el docker-compose up I comprovem que funciona

A screenshot of a terminal window showing the output of the 'docker compose up' command. The output indicates that the 'aplicacionjava' container was created successfully in 0.3s and is now running. It also shows the configuration of the 'proxy' service, including the launch of various scripts like '10-listen-on-ipv6-by-default.sh' and '20-envsubst-on-templates.sh'.

```
osboxes@osboxes:~/Examen_PPS$ docker compose up
[+] Running 1/1
  :: Container aplicacionjava Created                                0.3s
Attaching to aplicacionjava, examen_pps-proxy-1
aplicacionjava | NOTE: Picked up JDK_JAVA_OPTIONS:  --add-opens=java.base/j
ava.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.
base/java.util=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAME
D --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED
examen_pps-proxy-1 | /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty,
will attempt to perform configuration
examen_pps-proxy-1 | /docker-entrypoint.sh: Looking for shell scripts in /docke
r-entrypoint.d/
examen_pps-proxy-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-l
isten-on-ipv6-by-default.sh
examen_pps-proxy-1 | 10-listen-on-ipv6-by-default.sh: info: IPv6 listen already
enabled
examen_pps-proxy-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-e
nvsubst-on-templates.sh
examen_pps-proxy-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/30-t
une-worker-processes.sh
examen_pps-proxy-1 | /docker-entrypoint.sh: Configuration complete; ready for s
```



Sample "Hello, World" Application

This is the home page for a sample application used to illustrate the source directory organization Guide.

To prove that they work, you can execute either of the following links:

- To a [JSP page](#).
- To a [servlet](#).