

ARQUITECTURA I TECNOLOGIES DEL SOFTWARE
PRÀCTICA - DEVOPS
Enginyeria Informàtica
Escola d'Enginyeria (EE)
Universitat Autònoma de Barcelona (UAB)

Lluís Echeverria

DEPT. CIÈNCIES DE LA COMPUTACIÓ (DCC)
v1.0

2025

Contents

1	Introducció	3
1.1	Distribució de la nota	3
1.2	Data límit d'entrega	3
2	Part 1. Docker	4
2.1	Configuració dels micro-serveis	4
2.1.1	El balancejador de càrrega	4
2.1.2	Frontend & Backend	5
2.1.3	Base de dades	5
2.1.4	Gestor de base de dades	6
2.1.5	Configuració de xarxa	6
2.1.6	Lògica en backend	6
2.2	Objectius	7
3	Part 2. Docker-compose	8
3.1	Objectius	8
4	Entrega	9
5	Avaluació	9
6	Recursos Auxiliars	9

1 Introducció

El present document descriu les activitats relacionades amb la part pràctica del **Tema 2 DevOps** de l'assignatura **Arquitectura i Tecnologies del Software**.

La pràctica està dividida en dos parts:

1. **Part 1.** Docker
2. **Part 2.** Docker-compose

A continuació, s'explica els detalls de cada una de les dues parts.

1.1 Distribució de la nota

Per calcular la nota global de la pràctica, cada una de les parts té el següent pes:

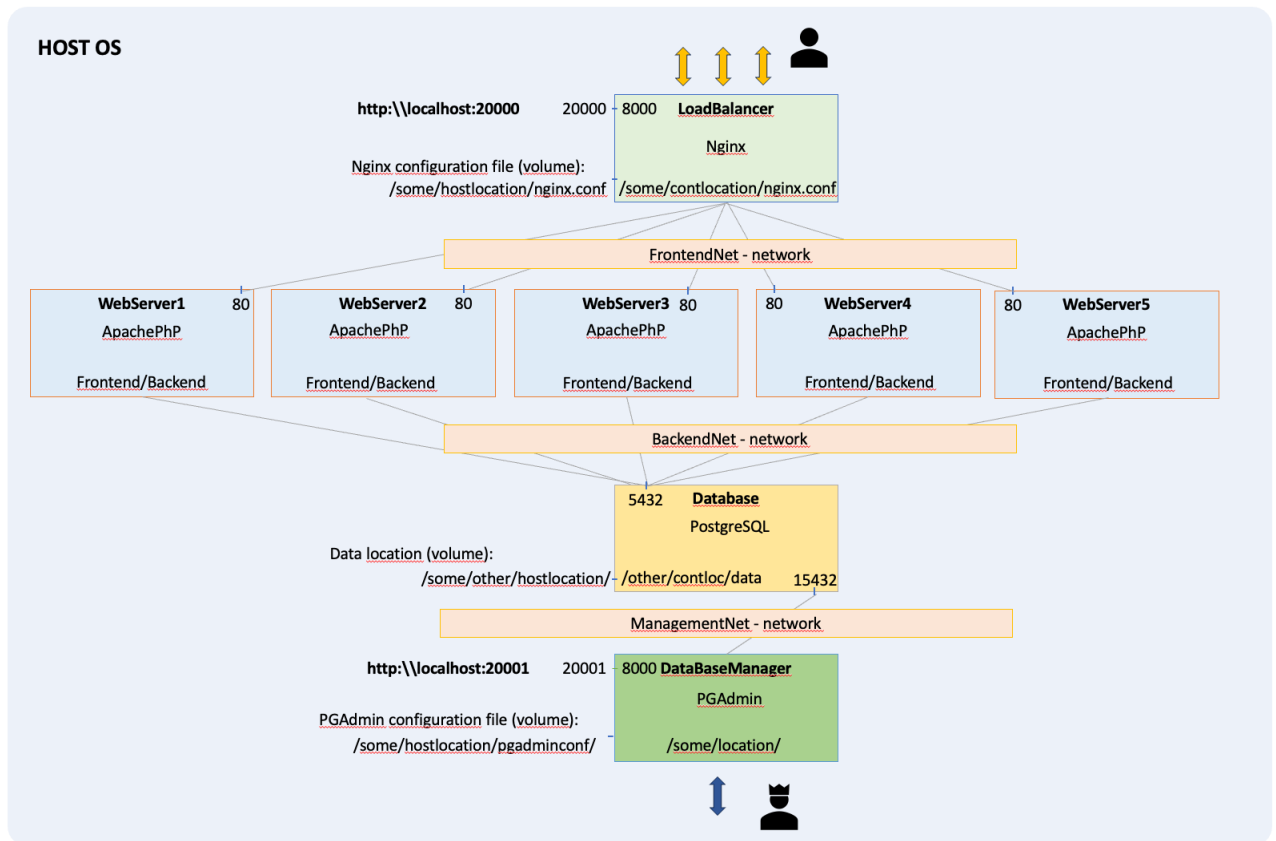
1. **Part 1.** 75%
2. **Part 2.** 25%

1.2 Data límit d'entrega

Les dos parts s'entregaran de forma conjunta abans del dia 12 de maig de 2025 (11 de maig 23:59).

2 Part 1. Docker

L'objectiu d'aquesta primera part de la pràctica és construir la següent topologia d'aplicació per microserveis mitjançant contenidors de Docker.



L'aplicació que s'ha de desenvolupar consisteix en una solució web clàssica on, inicialment, es troba un balancejador de càrrega que redirigeix les peticions http dels usuaris cap a la capa de frontend. A continuació, aquesta capa de frontend és l'encarregada de respondre amb un recurs o pàgina html adequadament generat per mostrar la informació. La generació d'aquesta pàgina es realitza en el backend, que accedeix a la base de dades per realitzar les accions necessàries i extreure la informació a mostrar a la pàgina web. Finalment, hi ha un servei web addicional que habilita la gestió de la base de dades.

2.1 Configuració dels micro-serveis

Per fer-ho, és necessari desplegar una arquitectura de micro-serveis, amb quatre tipus de serveis o funcionalitats diferenciades.

2.1.1 El balancejador de càrrega

El primer servei, el balancejador de càrrega, estarà implementat pel Software Nginx i format per un únic contenidor amb les següents característiques:

- Imatge oficial de Nginx

- Configurat amb una política de balanceig Round-Robin
- Nom del contenidor: LoadBalancer
- Hostname del contenidor: LoadBalancer
- Redirecció del port 20000 del host al 8000 del contenidor, on estarà escoltant el balancejador (s'haurà de configurar Nginx perquè escolti a aquest port)
- Volum del fitxer de configuració del balancejador entre host i contenidor

2.1.2 Frontend & Backend

El segon servei, representant tant el frontend com la lògica del backend, estarà format per cinc contenidors que aniran rebent les peticions per part del balancejador. Aquest servei estarà implementat mitjançant php i servit per un servidor d'aplicacions web Apache httpd, de manera que la pàgina primer es genera amb php en backend i després se serveix a frontend. La generació del codi php és molt senzilla i no és necessari tenir coneixements d'aquest framework. Al final d'aquest document trobareu enllaços amb la informació necessària per implementar-ho. Les característiques d'aquests contenidors són:

- Imatge oficial de php, variant amb apache httpd preinstal·lat
- S'haurà de crear un dockerfile que:
 - Parteixi de la imatge de php
 - Instal·li la extensió pgsql (per habilitar les funcionalitats de connexió a base de dades)
 - Habiliti la extensió pgsql al fitxer de configuració de php
 - Afegeixi a la imatge, a la ruta adequada, un fitxer index.php que prèviament haureu creat i preparat per realitzar les tasques de connexió a base de dades necessàries
- Nom dels contenidors: WebServer1, WebServer2, WebServer3, WebServer4 i WebServer5 respectivament
- Hostname dels contenidors: WebServer1, WebServer2, WebServer3, WebServer4 i WebServer5 respectivament

Apache httpd rebrà les peticions al port per defecte 80.

2.1.3 Base de dades

El tercer servei consisteix en una base de dades PostgreSQL. El primer cop que s'arranqui el servei, s'inicialitzarà de forma específica per crear una base de dades AppDB, amb una taula AppTable que disposarà de dues columnes, WebServer i Datetime, de tipus text i timestamp respectivament. Les característiques d'aquest contenidor són:

- Imatge oficial de PostgreSQL
- Es crearà el superusuari 'useradmin' amb password 'secure1234'
- S'haurà de crear un script .sql d'inicialització que creï la base de dades o la taula. Aquest s'haurà de col·locar allí on postgresql ho necessiti.

- Nom del contenidor: Database
- Hostname del contenidor: Database
- Volum per persistir la base de dades

PostgreSQL acceptarà connexions al port 15432.

2.1.4 Gestor de base de dades

L'últim servei serà un PGAdmin que estarà connectat a la base de dades i permetrà la seva gestió. Les característiques d'aquest contenidor són:

- Imatge oficial de PGAdmin
- Es crearà l'usuari 'admin@admin.com' amb password 'secretsecret'
- Nom del contenidor: DataBaseManager
- Hostname del contenidor: DataBaseManager
- Redirecció del port 20001 del host al 8000 del contenidor, on estarà escoltant el servei (s'haurà de configurar pgadmin perquè escolti a aquest port)
- Volum per persistir la configuració del servei, informació d'usuari...

Un cop arrancat, s'haurà de configurar l'accés a la base de dades i demostrar que s'hi pot accedir i veure la informació. La creació de la base de dades i la taula s'ha d'automatitzar en el moment d'inicialitzar el servei anterior, i no s'acceptarà que es realitzi aquest procés mitjançant PGAdmin o altres mecanismes manuals.

2.1.5 Configuració de xarxa

Serà necessari també definir la topologia de xarxa plantejada en l'esquema de la solució, creant les tres xarxes 'FrontendNet', 'BackendNet' i 'ManagementNet', de tipus Bridge, i connectant els contenidors amb direccionament estàtic i IP fixa (vosaltres escollu rang i IPs). Gràcies a la redirecció de ports proposada es podrà provar la solució des d'un navegador en la màquina host.

2.1.6 Lògica en backend

En el moment de generar la pàgina web en el backend del segon servei (els WebServers), serà necessari realitzar dues consultes a la base de dades AppDB, a la taula AppTable. La primera afegirà una instància a la taula amb el nom del Server que realitza la consulta (WebServer1, WebServer2, WebServer3, WebServer4 o WebServer5) a la columna WebServer, i a la columna Datetime el datetime en aquell instant. Un cop realitzat l'insert, la segona consulta recuperarà el nombre de instàncies en la taula corresponents a aquell WebServer. Amb aquesta informació es generarà el contingut de la pàgina web, que haurà de mostrar el codi del contenidor que la gestiona i el nombre de vegades que ha servit resposta a client. Trobareu una plantilla d'aquest codi php al fitxer index.php i s'hauran d'utilitzar funcions com pg_pconnect o pg_query per establir les connexions a base de dades i realitzar les consultes.

2.2 Objectius

1. Configuració contenidors Docker: Dockerfile + altres recursos + comandes d'execució amb els paràmetres indicats
2. Configuració xarxa
3. Configuració de volums
4. Configuració de fitxers addicionals per implementar la lògica explicada
5. Configuració PGAdmin mitjançant la seva UI per accedir a la base de dades
6. Informe final amb explicacions de com s'ha implementat la solució, les comandes utilitzades, i captures de pantalla per demostrar el correcte funcionament

3 Part 2. Docker-compose

L'objectiu d'aquesta segona part de la pràctica és construir la mateixa topologia d'aplicació per microserveis mitjançant un únic fitxer docker-compose.

3.1 Objectius

1. Configuració i gestió de serveis amb un únic fitxer docker-compose
 - a) Configuració serveis
 - b) Configuració xarxa
 - c) Configuració de volums
2. Configuració de fitxers addicionals per implementar la lògica explicada
3. Configuració PGAdmin mitjançant la seva UI per accedir a la base de dades
4. Informe final amb explicacions de com s'ha implementat la solució, les comandes utilitzades, i captures de pantalla per demostrar el correcte funcionament.

4 Entrega

Per fer l'entrega caldrà que genereu un zip: **ats_2025_CC_GX.zip** (on GX correspon al identificador del grup) amb el vostre treball, aquest zip l'haureu de pujar al campus. Dins el zip hi ha d'haver:

- informe.pdf : Un únic informe en format pdf amb les dues parts. Indiqueu els noms dels alumnes i l'identificador del grup.
- files: Directori amb tots els fitxers necessaris per executar la solució (Dockerfiles, compose.yml, altres fitxers auxiliars...)

5 Avaluació

Les imatges utilitzades han de ser les oficials, de Docker Hub, per cada tecnologia ja sigui per crear directament un contenidor o servei, o per definir el Dockerfile. Els dockerfiles creats es compilaran amb la comanda docker build. Es podrà entrar als contenidors per validar que les configuracions són correctes, però en cap cas s'acceptaran entregues on es modifiquin paràmetres entrant als contenidors per línia de comandes.

La validació serà mitjançant el navegador web de la màquina host, on s'accedirà a la url localhost:20000, i cada vegada que s'actualitzi mostrarà la pàgina corresponent gràcies al balanceig Round-Robin entre els servidors WebServer1, WebServer2, WebServer3, WebServer4 o WebServer5. Si tot funciona correctament s'hauria de mostrar el següent:

WebServer ID: WebServer1 - Num served requests: 1

I si es continua realitzant peticions, podriem veure:

WebServer ID: WebServer4 - Num served requests: 3

Finalment, si es navega a la url localhost:20001, s'accedirà a la UI de PGAdmin per gestionar la base de dades.

Finalment, s'haurà de realitzar un informe final que expliqui els passos seguits i comandes executades per implementar la solució proposada. Aquest haurà de contenir, més enllà de les explicacions i documentació necessàries, captures de pantalla del navegador per demostrar que la solució està funcionant de forma correcta.

Elements que es tindran en compte per avaluar són:

- Complir els objectius.
- La solució compleix amb els requeriments sense incorporar configuracions no sol·licitades.
- Es respecten els formats i noms de fitxers en l'entrega.
- La documentació està treballada.
- No es detecta plagi.

6 Recursos Auxiliars

Documentació oficial NGINX:

http://nginx.org/en/docs/http/load_balancing.html
<https://www.nginx.com/blog/load-balancing-with-nginx-plus/>
<https://www.nginx.com/blog/load-balancing-with-nginx-plus-part2/>
<https://www.nginx.com/resources/admin-guide/load-balancer/>

Anex configuració exemple nginx.conf NGINX y proxy com a balancejador de pàgines web

Utilitats de php per implementar la connexió a la base de dades:
<https://www.php.net/manual/en/function.pg-query.php>
<https://www.php.net/manual/en/function.pg-connect.php>

Docker documentació:
<https://docs.docker.com/compose/>
<https://docs.docker.com/engine/reference/commandline/build/>