



Taller: Despliegue de Aplicaciones con Docker (4/5)

Milton Jesús Vera Contreras - miltonjesusvc@ufps.edu.co
Seminario Internacional de Investigación en Ingeniería del Software
Medellín, 2019/10/17

Parte 4: Ejemplo usando docker-compose (multi docker)

En esta cuarta guía aprenderá a desplegar una aplicación con varios docker, uno en cada capa web: base de datos, backend y frontend.

Fichero docke-compose.yml: Ubíquese en el directorio seiis2019/ejemploDockerCompose. Observe allí el fichero docker-compose.yml. Hay una guía de referencia y tutorial sobre este tipo de ficheros

```
cd /home/ubuntu/seiis2019/ejemploDockerCompose
more docker-compose.yml
```

```
32 lines (29 sloc) | 608 Bytes
Raw Blame History

1 version: '3'
2 services:
3   mydb:
4     container_name: mydb
5     image: mariadb
6     env_file:
7       - .env
8     volumes:
9       - /taller-docker/volumenes/mydb:/var/lib/mysql
10    ports:
11      - 3306:3306
12    restart: always
13
14    mybackend:
15      container_name: mybackend
16      image: ingsistemascloud/mytomcat
17      ports:
18        - 8080:8080
19      depends_on:
20        - mydb
21
22    myfrontend:
23      container_name: myfrontend
24      image: nginx
25      volumes:
26        - /taller-docker/volumenes/myfrontend:/var/www/html
27        - ./my-nginx.conf:/etc/nginx/conf.d/my-nginx.conf
28      ports:
29        - 80:80
30      depends_on:
31        - mybackend
```



Taller: Despliegue de Aplicaciones con Docker (4/5)

Milton Jesús Vera Contreras - miltonjesusvc@ufps.edu.co
Seminario Internacional de Investigación en Ingeniería del Software
Medellín, 2019/10/17

El fichero docker-compose.yml tiene un servicio por cada capa. Observe que hay unas reglas de dependencia “depends_on” entre los dockers. Observe que cada docker expone dos puertos, el primero corresponde al puerto que podrá usar el servidor y el segundo al puerto privado del docker. Los puertos pueden ser diferentes.

Ejecute el siguiente comando y automáticamente se crearán los tres docker

```
sudo docker-compose up -d
```

Cuando quiera recrear todos los dockers después de realizar cambios use el siguiente comando

```
sudo docker-compose up --build -d
```

Pruebe usar los comandos docker que aprendió en la primera y segunda parte del taller, para revisar las direcciones IP y detalles de las imágenes y los contenedores:

```
udo docker images
```

```
sudo docker container ps
```

```
sudo docker inspect mydb
```

```
sudo docker inspect backend
```

```
sudo docker inspect frontend
```

```
sudo docker inspect --format='{{range .NetworkSettings.Networks}}  
{{.IPAddress}}{{end}}' backend
```

```
sudo docker inspect --format='{{json .Config}}' backend
```

```
sudo ls /taller-docker/volumenes/
```

Ingresa al navegador y verifique que la aplicación está ejecutándose.