

---

# Práctica

IAW - Implantación de Aplicaciones Web

Curso 2025/2026

# Índice general

<b>1</b>	<b>Balanceo de carga con HAProxy</b>	<b>1</b>
1.1	Cómo escalar los servicios definidos en un archivo <code>docker-compose.yml</code> . . . . .	1
1.2	Ejemplo de un archivo <code>docker-compose.yml</code> con un balanceador de carga . . . . .	1
1.3	Archivo de configuración <code>haproxy.cfg</code> . . . . .	2
1.4	Referencias . . . . .	2
<b>2</b>	<b>Licencia</b>	<b>3</b>

# 1 Balanceo de carga con HAProxy

En esta práctica vamos a modificar los archivos `docker-compose.yml` que hemos creado en las prácticas 15 y 16, y vamos a incluir un nuevo contenedor Docker con HAProxy para balancear la carga de los contenedores que ejecutan la aplicación web.

Posteriormente deberá realizar la implantación de ambos sitios web en Amazon Web Services (AWS) haciendo uso de contenedores Docker y de la herramienta Docker Compose.

## 1.1. Cómo escalar los servicios definidos en un archivo `docker-compose.yml`

Cuando ejecutamos `docker-compose` tenemos la posibilidad de indicar el número de instancias que queremos tener de cada uno de los servicios que vamos a crear.

El comando sería el siguiente:

```
1 docker-compose up --scale SERVICE=NUM
```

Donde:

- `SERVICE` es el nombre del servicio que queremos escalar
- `NUM` es el número de instancias que queremos tener de ese servicio.

**Ejemplo:**

En el siguiente ejemplo estaríamos iniciando todos los servicios que están definidos en el archivo `docker-compose.yml` y para el servicio de `wordpress` estaríamos creando 4 instancias.

```
1 docker-compose up --scale wordpress=4
```

## 1.2. Ejemplo de un archivo `docker-compose.yml` con un balanceador de carga

A continuación se muestra **un fragmento** de un archivo `docker-compose.yml` que incluye un servicio de balanceo de carga con HAProxy que nos puede servir de ejemplo:

```
1 services:
2   lb:
3     image: dockercloud/haproxy <1>
4     ports:
5       - 80:80 <2>
6       - 1936:1936 <3>
7     links:
8       - apache <4>
9     volumes:
10      - /var/run/docker.sock:/var/run/docker.sock <5>
11
12  apache:
13    ...
```

<1> Utilizaremos la imagen [dockercloud/haproxy](#) que está disponible en [Docker Hub](#).

<2> El puerto 80 será el puerto del servicio que queremos balancear.

<3> El puerto 1936 nos permite acceder a una página web con información estadística del balanceador.

<4> Creamos un enlace con el servicio que queremos balancear. Los enlaces permiten que los contenedores se descubran entre sí y transfieran de manera segura información sobre un contenedor a otro contenedor.

<5> Es necesario montar el socket UNIX del Docker daemon (/var/run/docker.sock) para que el contenedor `lb` pueda comunicarse con el Docker daemon y obtener información del resto de contenedores.

### 1.3. Archivo de configuración haproxy.cfg

El archivo de configuración que utiliza [HAProxy](#) es `haproxy.cfg`.

Localice el archivo `haproxy.cfg` dentro del contenedor que ejecuta el balanceo de carga y estudie los parámetros de configuración que está utilizando.

### 1.4. Referencias

- [Ejemplo de balanceo de carga con HAProxy de una aplicación web con Redis y Flask.](#)
- [An Introduction to HAProxy and Load Balancing Concepts.](#)
- [Balancear servicio web con HAProxy en Ubuntu 18.04.](#)
- [HAProxy. Starter guide.](#)
- [HAProxy. Configuration manual.](#)
- [ab - Apache HTTP server benchmarking tool.](#)
- [Docker](#)
- [Docker Compose](#)

## 2 Licencia

Esta página forma parte del curso Implantación de Aplicaciones Web de José Juan Sánchez y su contenido se distribuye bajo una licencia Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.