

# Manual Avanzado de Docker (Edición Ampliada)

---

## 14. Balanceo de Carga con Docker (Nginx Load Balancer)

### Objetivo

Distribuir tráfico entre múltiples contenedores web para alta disponibilidad.

### docker-compose.yml

```
version: "3.9"
services:
  web1:
    image: nginx
  web2:
    image: nginx

  lb:
    image: nginx
    ports:
      - "8080:80"
    volumes:
      - ./nginx.conf:/etc/nginx/nginx.conf
```

### nginx.conf

```
events {}

http {
  upstream backend {
    server web1;
    server web2;
  }

  server {
    listen 80;
    location / {
      proxy_pass http://backend;
    }
  }
}
```

Ejecutar:

```
docker compose up -d
```

Verificar balanceo accediendo repetidamente a: <http://localhost:8080>

---

## 15. Proxy Inverso con Nginx

### Función

- Centraliza acceso.
- Oculta servicios internos.
- Permite SSL y balanceo.

Ejemplo:

```
server {
    listen 80;
    server_name app.local;

    location / {
        proxy_pass http://web:80;
    }
}
```

---

## 16. HTTPS y Certificados con Let's Encrypt

Generación de certificados (Certbot)

docker-compose.yml:

```
services:
  nginx:
    image: nginx
    ports:
      - "80:80"
      - "443:443"
  certbot:
    image: certbot/certbot
```

Certificado:

```
docker run --rm certbot/certbot certonly --standalone -d midominio.com
```

Certificado autofirmado (laboratorio)

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout server.key -  
out server.crt
```

## 17. Integración con Java en Docker

Aplicación Java simple

Main.java:

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Aplicación Java en Docker");  
    }  
}
```

Dockerfile:

```
FROM eclipse-temurin:21-jre  
WORKDIR /app  
COPY Main.class .  
CMD ["java","Main"]
```

Construcción:

```
javac Main.java  
docker build -t java-app .  
docker run java-app
```

Spring Boot

Dockerfile:

```
FROM eclipse-temurin:21-jre  
WORKDIR /app  
COPY target/app.jar app.jar  
EXPOSE 8080  
ENTRYPOINT ["java","-jar","app.jar"]
```

## 18. Linux en Docker (Administración)

## Acceso a contenedor Linux

```
docker run -it ubuntu bash
```

Comandos útiles:

```
apt update  
top  
df -h  
free -m  
ip a
```

## Gestión de usuarios

```
adduser alumno  
su alumno
```

---

# 19. Ejercicios Avanzados Resueltos

## Ejercicio 1

Crear un balanceador con 3 nodos web.

Solución:

```
docker compose up -d --scale web=3
```

---

## Ejercicio 2

Configurar HTTPS autofirmado.

Solución:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout key.pem -out cert.pem
```

---

## Ejercicio 3

Crear imagen Java personalizada.

Solución:

```
docker build -t java-asir .
docker run java-asir
```

---

## Ejercicio 4

Administrar un contenedor Linux.

Solución:

```
docker exec -it ubuntu bash
useradd admin
passwd admin
```

---

## Ejercicio 5

Simular caída de un nodo.

Solución:

```
docker stop web1
```

---

## 20. Prácticas Linux Propuestas

1. Crear un contenedor Ubuntu y:

- Crear usuarios.
- Instalar Apache.
- Configurar firewall interno.

2. Monitorizar consumo de recursos con:

```
htop
docker stats
```

3. Crear backup de un volumen:

```
docker run --rm -v datos:/data -v $(pwd):/backup ubuntu tar czf
/backup/copia.tar.gz /data
```

4. Restaurar backup.
  5. Simular ataque de carga con Apache Benchmark.
- 

## 21. Futuras Ampliaciones

- Kubernetes.
- CI/CD con Docker.
- Seguridad avanzada.
- Docker Swarm.