**Escalat i desplegament segur amb Docker Swarm**

En aquesta pràctica veurem els següents apartats:

1. *Desplegament d’un servei bàsic (NGINX)*

*Comencem creant un servei senzill amb NGINX que serveix una pàgina web estàtica. Aquest servei el despleguem dins d’un stack Swarm com a únic contenidor per iniciar la pràctica.*

1. *Escalat horitzontal controlat*

*Un cop desplegat el servei, practiquem com escalar-lo incrementant manualment el nombre de rèpliques (per exemple, de 1 a 3 rèpliques).*

1. *Simulació de desplegament segur (rolling update)*

*Fem un canvi al contingut de la pàgina (fitxer HTML) i reconstruïm la imatge. Després, actualitzem el servei utilitzant una estratègia de rolling update, és a dir, actualitzant les rèpliques de forma progressiva per minimitzar la interrupció del servei.*

1. *Aplicació de bones pràctiques de seguretat*

*A la segona part de la pràctica, configurem el servei per complir diversos principis de seguretat fonamentals:*

* *Utilització de secrets Docker: creem un secret simulat (web\_banner) que incorporem al contenidor a través de /run/secrets/, mostrant com protegir informació sensible.*
* *Ús de xarxes internes: configurem el servei dins d’una xarxa overlay privada per garantir que només serveis autoritzats hi puguin accedir.*

**Requisits previs**

* Docker en mode Swarm activat. Si no ho està, executar:

**docker swarm init --advertise-addr $(hostname -I | awk '{print $1}')**

El que fa aquesta comanda és iniciar swarm amb la IP local.

**1. Crear els fitxers**

mkdir escalat-docker

cd escalat-docker

Fitxer index.html:

**<!DOCTYPE html>**

**<html>**

 **<head>**

 **<meta charset="utf-8">**

 **<title>Webdemo</title>**

 **</head>**

 **<body>**

 **<h1>Hola! Sóc la rèplica: <span id="hostname">...</span></h1>**

 **<!--**

 **Aquest script fa una petició a /hostname i escriu la resposta dins del <span id="hostname">.**

 **Serveix per mostrar quin contenidor (hostname) està servint la pàgina en entorns amb múltiples rèpliques.**

 **Permet visualitzar l'escalat i el balanceig de càrrega en temps real.**

 **-->**

 **<script>**

 **fetch('/hostname')**

 **.then(r => r.text())**

 **.then(t => document.getElementById("hostname").innerText = t.trim());**

 **</script>**

 **</body>**

**</html>**

Fitxer nginx.conf

**worker\_processes 1;**

**events {**

 **worker\_connections 1024;**

**}**

**http {**

 **access\_log /dev/stdout;**

 **error\_log /dev/stderr;**

 **include mime.types;**

 **default\_type text/html;**

 **sendfile on;**

 **keepalive\_timeout 65;**

 **server {**

 **listen 80;**

 **server\_name localhost;**

 **location / {**

 **root /usr/share/nginx/html;**

 **index index.html;**

 **}**

 **location /hostname {**

 **default\_type text/plain;**

 **return 200 "$hostname\n";**

 **}**

 **location /banner {**

 **alias /run/secrets/web\_banner;**

 **default\_type text/plain;**

 **}**

 **}**

**}**

Fitxer Dockerfile:

**FROM alpine:3.19**

**RUN apk add --no-cache nginx**

**COPY index.html /usr/share/nginx/html/index.html**

**COPY nginx.conf /etc/nginx/nginx.conf**

**CMD ["nginx", "-g", "daemon off;"]**

**2. Crear un secret simulat**

**echo "BENVINGUT A LA INTRANET" | docker secret create web\_banner -**

*Un secret en Docker Swarm és una dada confidencial (com una contrasenya, clau d’API, token o certificat) que es gestiona de manera segura dins del clúster. Els secrets es guarden xifrats i només es fan accessibles als serveis que els necessiten, durant el temps d’execució i a través d’un fitxer de només lectura (normalment a /run/secrets/<nom> dins del contenidor).*

**3. Fitxer docker-compose.yml**

**version: '3.8'**

**services:**

 **web:**

 **image: webdemo\_web:latest**

 **ports:**

 **- "8080:80"**

 **secrets:**

 **- web\_banner**

 **networks:**

 **- internal**

 **deploy:**

 **replicas: 1**

 **update\_config:**

 **parallelism: 1**

 **delay: 5s**

 **restart\_policy:**

 **condition: on-failure**

**secrets:**

 **web\_banner:**

 **external: true**

**networks:**

 **internal:**

 **driver: overlay**

**Construïm la imatge amb**

**docker build -t webdemo\_web:latest .**

**4. Desplegar el servei**

**docker stack deploy -c docker-compose.yml webdemo**

**5. Validar el servei**

**docker service ls**

**docker service ps webdemo\_web**

Obrir el navegador i accedir a: [http://localhost:8080](http://localhost:8080/)

Recarregar diverses vegades per veure si canvia el nom del contenidor (hostname)

També podeu accedir a http://localhost:8080/banner per comprovar que es mostra el contingut del secret.

**6. Escalat manual**

**docker service scale webdemo\_web=3**

Validar amb:

**docker service ps webdemo\_web**

Torneu al navegador i aneu recarregant la pàgina clicant majúscules i la icona de recarregar:

Us contesta sempre la mateixa replica web?

**7. Simular una actualització**

Modificar el fitxer index.html afegint un canvi visible.

Per exemple:

**<h1>Hola! Sóc la rèplica: <span id="hostname"></span></h1>**

**<p style="color: green;">[Nova versió desplegada el 9 d’abril]</p>**

Reconstruir la imatge:

**docker build -t webdemo\_web:latest .**

Fer un rolling update:

**docker service update --force --image webdemo\_web:latest webdemo\_web**

 *docker service update
Comanda que actualitza un servei ja desplegat en mode Swarm.*

 *webdemo\_web
És el nom del servei que volem actualitzar dins de l’stack webdemo.
El format és nomStack\_nomServei.*

**8. Netejar l’entorn**

**docker stack rm webdemo**

**docker secret rm web\_banner**

**Objectius assolits:**

* Escalat horitzontal manual
* Actualització controlada (rolling update)
* Ús de secrets
* Execució amb privilegis mínims
* Exposició controlada a través d’un únic port i xarxa privada

**Per saber més:**

* <https://docs.docker.com/engine/swarm/how-swarm-mode-works/services/>
* <https://docs.docker.com/engine/swarm/secrets/>