

Orquestación con Docker Swarm

Arturo Silvelo

Try New Roads

1. Clúster local: Crear entorno de pruebas

- ¿Qué es Docker Swarm? Motor de orquestación nativo de Docker para gestionar clústeres de contenedores.
- Permite crear un clúster de nodos (máquinas físicas o virtuales) y desplegar servicios distribuidos.
- Comando para inicializar un clúster en local:

```
docker swarm init
```

- Añadir nodos al clúster:

```
docker swarm join --token <token> <ip_manager>:2377
```

- Puedes simular varios nodos en una sola máquina usando VMs o Docker-in-Docker.

2. Despliegue: Gestionar stacks y rollback

- Un stack es un conjunto de servicios definidos en un archivo `docker-compose.yml` adaptado para Swarm.

- Desplegar un stack:

```
docker stack deploy -c docker-compose.yml mystack
```

- Listar servicios y stacks:

```
docker stack ls  
docker service ls
```

- Rollback de un servicio:

```
docker service update --rollback <service>
```

3. Escalado: Ajustar réplicas

- Swarm permite escalar servicios fácilmente:

```
docker service scale mystack_web=5
```

- El orquestador distribuye las réplicas entre los nodos disponibles.

4. Red overlay: Comunicación entre nodos

- Swarm crea redes overlay para que los servicios se comuniquen entre nodos, incluso en diferentes hosts.

- Crear una red overlay:

```
docker network create -d overlay mi_red
```

- Los servicios conectados a la misma red overlay pueden comunicarse por nombre de servicio.

5. Secrets: Gestionar datos sensibles

- Swarm permite gestionar secretos (contraseñas, claves, etc.) de forma segura.
- Crear un secreto:

```
echo "mi_password" | docker secret create db_password -
```

- Usar un secreto en un servicio:

```
services:
  db:
    image: mysql
    secrets:
      - db_password
secrets:
  db_password:
    external: true
```

- Los secretos solo están disponibles en el contenedor mientras el servicio se está ejecutando.

