LAB 17: QUARKUS K8S

Autor: José Díaz

Github Repo: https://github.com/joedayz/quarkus-bcp-2025.git

Abre el proyecto **14-deploy-k8s-start**

Instrucciones para Podman

Esta demo contiene dos servicios:

- **expense-service:** Servicio REST de gastos
- **expense-client:** Cliente que consume expense-service.

1. Levantar y Validar el cluster.

   ./scripts/kind-up.sh

2. Construir y cargar las imágenes

   ./scripts/build-and-load-all.sh

3. Desplegar ambos componentes

   ./scripts/deploy-all-kind.sh

4. Verificar y probar

   kubectl get pods
   kubectl get svc expense-service expense-client
   curl http://localhost:8081/expenses

NOTA:

- expense-client se expone por NodePort 30081 y el kind-config lo publica en el host 8081.
- El ConfigMap expense-client-config inyecta EXPENSE_SVC=http://expense-service:8080 en el pod del cliente.

5. Luego de revisar la demo, realiza la limpieza de todo lo desplegado.

   ./scripts/undeploy-all-kind.sh

6. OPCIONAL: Si quieres probar la demo EXPENSE hacer un reinicio completo (borrar y recrear el cluster).

   expense/scripts/reset-and-redeploy.sh

Luego para desplegar sólo este servicio ‘expense’:

expense/scripts/kind-up.sh
expense/scripts/build-and-load.sh
expense/scripts/deploy-kind.sh

Luego probar en: http://localhost:8080

Instrucciones para Docker Desktop

Esta demo contiene dos servicios:

- **expense-service:** Servicio REST de gastos
- **expense-client:** Cliente que consume expense-service.

Los scripts se encuentran en el directorio **docker-desktop**.

1. Verificar que Docker Desktop Kubernetes esté disponible

scripts/cluster-up.sh

2. Construir las imágenes Docker

scripts/build-and-load-all.sh

3. Desplegar ambos componentes

scripts/deploy-all.sh

4. Verificar y probar:

kubectl get pods
kubectl get svc expense-service expense-client

Obtener la IP del servicio:
kubectl get svc expense-client

O usar port-forward:
kubectl port-forward svc/expense-client 8081:8080

Luego accede a: **http://localhost:8081**

Notas:

- expense-client se expone como LoadBalancer para facilitar el acceso desde Docker Desktop
- El ConfigMap expense-client-config inyecta EXPENSE_SVC=http://expense-service:8080 en el pod del cliente
- Docker Desktop Kubernetes puede usar imágenes locales directamente, sin necesidad de cargarlas manualmente

5. Eliminar los componentes de la demo:

   scripts/undeploy-all.sh

Enjoy!

Jose