

1. Crear webapp.yaml

2. Iniciar el cluster de minikube:

minikube start

3. Aplicar el archivo

kubectl apply -f webapp.yaml

Salida en consola:

```
kubectl apply -f webapp.yaml

deployment.apps/webapp created
service/webapp-service created
```

4. Verificar recursos creados

```
kubectl get deployments

NAME      READY   UP-TO-DATE   AVAILABLE   AGE
webapp    0/2     2            0           42s
```

kubectl get pods

Salida en consola:

NAME READY STATUS RESTARTS AGE

NAME	READY	STATUS	RESTARTS	AGE
webapp-894cd5756-tpbp8	0/1	CreateContainerConfigError	0	70s
webapp-894cd5756-xdkm8	0/1	CreateContainerConfigError	0	70s

kubectl get services

Salida en consola:

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	27d
webapp-service	NodePort	10.98.58.128	<none>	80:30080/TCP	2m35s

5. Ir a http://localhost:30080

6. Crear configmap.yaml con lo requerido y aplicarlo

kubectl apply -f configmap.yaml

Salida en consola:

configmap/webapp-config created

7. Inyectar la variable de entorno SALUDO a webapp.yaml y reaplicar

kubectl apply -f webapp.yaml

Salida en consola:

```
kubectl exec -it webapp-894cd5756-tpbp8 -- printenv SALUDO
¡Hola desde Kubernetes!
```

deployment.apps/webapp configured

service/webapp-service unchanged

8. Verificar que el pod reciba la variable

kubectl get pods

Salida en consola:

NAME READY STATUS RESTARTS AGE

webapp-7b5cc74dfb-4xlqg 1/1 Running 0 37s

webapp-7b5cc74dfb-w9hbl 1/1 Running 0 35s

kubectl exec -it webapp-7b5cc74dfb-4xlqg -- printenv SALUDO

Salida en consola:

¡Hola desde Kubernetes!

----- MANEJAR INFORMACION SENSIBLE CON SECRETS -----

9. Crear secret.yaml con lo requerido y aplicarlo

kubectl apply -f secret.yaml

Salida en consola:

secret/webapp-secret created

10. Inyectar la variable de entorno PASSWORD a webapp.yaml y reaplicar

kubectl apply -f webapp.yaml

Salida en consola:

deployment.apps/webapp configured

service/webapp-service unchanged

kubectl get pods

Salida en consola:

NAME READY STATUS RESTARTS AGE

webapp-756b59876-vxfcg 1/1 Running 0 35s

webapp-756b59876-xhm5s 1/1 Running 0 38s

```
kubectl exec -it webapp-894cd5756-tpbp8 -- printenv PASSWORD
seculo123
```

----- MONTAR ARCHIVOS COMO VOLUMENES (CONFIG. POR ARCHIVO) -----

11. Crear config-volume.yaml con lo requerido y aplicar

kubectl apply -f config-volume.yaml

Salida en consola:

configmap/html-config created

11. Crear webapp-volumen.yaml con lo requerido y aplicar

kubectl apply -f webapp-volumen.yaml

Salida en consola:

deployment.apps/webapp-volumen created

kubectl expose deployment webapp-volumen --type=NodePort --port=80 --name=webapp-volumen-service

Salida en consola:

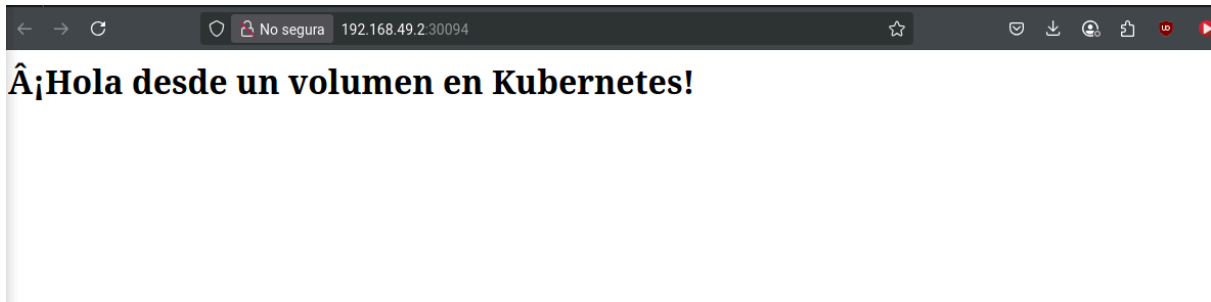
service/webapp-volumen-service exposed

12. minikube service webapp-volumen-service

Salida en consola:

```
minikube service webapp-volumen-service
-----|-----|-----|-----|
| NAMESPACE | NAME | TARGET PORT | URL |
|-----|-----|-----|-----|
| default | webapp-volumen-service | 80 | http://192.168.49.2:30094 |
|-----|-----|-----|-----|
```

Navegador:



----- LIMPIEZA DE LOS RECURSOS -----

13. Ejecutar:

```
kubectl delete -f webapp.yaml
kubectl delete -f configmap.yaml
kubectl delete -f secret.yaml
kubectl delete -f config-volume.yaml
kubectl delete -f webapp-volumen.yaml
kubectl delete service webapp-service
kubectl delete service webapp-volumen-service
```

----- DESPLIEGUE DE MULTIPLES SERVICIOS -----

14. Crear la carpeta k8s-auth-demo

15. Crear namespace.yml con lo requerido y aplicar

16. Ejecutar:

```
kubectl config set-context --current --namespace=auth-demo
```

Salida en consola:

Context "minikube" modified.

17. Definir las bases de datos en db1.yaml y db2.yaml

18. Definir los servicios país y auth en pais-service.yaml y auth-service.yaml

19. Aplicar todos los recursos

```
kubectl apply -f db1.yaml
kubectl apply -f db2.yaml
kubectl apply -f pais-service.yaml
kubectl apply -f auth-service.yaml
```

20. Aplicar

```
kubectl get pods
```

```
kubectl get svc
```

Salida en consola:

21. Acceso desde el navegador, en mi caso con Docker Desktop

```
kubectl port-forward svc/pais-service 8082:8080
```

```
kubectl port-forward svc/auth-service 8081:8080
```

Salida en consola:

```
• L> kubectl get pods
kubectl get svc
NAME                                READY   STATUS             RESTARTS   AGE
auth-service-7df446bdbb-c2bgx      0/1     ContainerCreating   0           1s
db1-dbd647-k99z9                   0/1     ContainerCreating   0           7s
db2-pvc-5cc595955-t69dd            0/1     ContainerCreating   0           5s
pais-service-796bb57fcc-dsfsp      0/1     ContainerCreating   0           4s
NAME      TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
auth-service  ClusterIP   10.102.138.231  <none>        8080/TCP   4s
db1          ClusterIP   10.107.166.102  <none>        5432/TCP   8s
db2-pvc      ClusterIP   10.107.118.83   <none>        5432/TCP   7s
pais-service  ClusterIP   10.106.39.136   <none>        8080/TCP   5s
```