

1. Crear webapp.yaml con lo requerido

2. Iniciar el cluster de minikube:

```
minikube start
```

3. Aplicar el archivo

```
kubectl apply -f webapp.yaml
```

Salida en consola:

```
deployment.apps/webapp created
```

```
service/webapp-service created
```

4. Verificar recursos creados

```
kubectl get deployments
```

Salida en consola:

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
webapp	2/2	2	2	77s

```
kubectl get pods
```

Salida en consola:

NAME	READY	STATUS	RESTARTS	AGE
webapp-d5b7f5fb7-2sbg6	1/1	Running	0	2m6s
webapp-d5b7f5fb7-8bj7l	1/1	Running	0	2m6s

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:

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

```
kubect 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-756b59876-vxfgc -- printenv PASSWORD
```

Salida en consola:

```
secur0123
```

----- 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:

NAMESPACE	NAME	TARGET PORT	URL
default	webapp-volumen-service	80	http://192.168.49.2:30848
* Starting tunnel for service webapp-volumen-service.			
NAMESPACE	NAME	TARGET PORT	URL
default	webapp-volumen-service		http://127.0.0.1:57894
* Opening service default/webapp-volumen-service in default browser...			

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:

```
kubectrl 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

```
kubectrl apply -f db1.yaml
```

```
kubectrl apply -f db2.yaml
```

```
kubectrl apply -f pais-service.yaml
```

```
kubectrl apply -f auth-service.yaml
```

20. Aplicar

```
kubectl get pods
```

```
kubectl get svc
```

Salida en consola:

NAME	READY	STATUS	RESTARTS	AGE
auth-service-6b7fb87dc8-ljbnm	0/1	ContainerCreating	0	11s
db1-7d96bf4fc5-9srp2	0/1	ContainerCreating	0	12s
db2-pvc-54b76cdff5-6nmgp	0/1	ContainerCreating	0	12s
pais-service-67d545f5cc-4kzcx	0/1	ContainerCreating	0	12s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
auth-service	ClusterIP	10.100.82.214	<none>	8080/TCP	11s
db1	ClusterIP	10.111.109.178	<none>	5432/TCP	12s
db2-pvc	ClusterIP	10.110.113.34	<none>	5432/TCP	12s
pais-service	ClusterIP	10.103.7.52	<none>	8080/TCP	12s

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:

```
Forwarding from 127.0.0.1:8082 -> 8080
Forwarding from [::1]:8082 -> 8080
Handling connection for 8082
Handling connection for 8082
□
```