

Desafío 11

Bootcamp Devops Engineer

Desarrollo:

Vamos a crear los manifiestos de Kubernetes necesarios para desplegar la aplicación NestJS y su base de datos MongoDB, basándonos en lo el desafío anterior.

- **Manifiestos de Kubernetes**

Se crean los siguientes archivos YAML en una carpeta en la raíz del proyecto llamada ks8.

1. Deployment de la Aplicación (app-deployment.yaml).

Crea un archivo llamado app-deployment.yaml

2. Deployment para MongoDB

Crea un archivo llamado mongo-deployment.yaml.

3. Persistent Volume Claim para MongoDB


Crea un archivo llamado mongo-pvc.yaml:

- **Construir la imagen Docker y subir a Dockerhub**



Para asegurarnos de que la imagen esté disponible en un registro accesible:



```
docker build -t <usuario_dockerhub>/app:tag .
```

```
docker push < usuario_dockerhub>/app:tag
```

1wil1/desafio11 



Last pushed 3 minutes ago

This repository does not have a description   INCOMPLETE

This repository does not have a category   INCOMPLETE

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
 1		Image	2 minutes ago	3 minutes ago

[See all](#)

- **Aplicar los manifiestos**

Se deben ejecutar los siguientes comandos:

```
kubectl apply -f mongo-pvc.yaml
```

```
kubectl apply -f mongo-deployment.yaml
```

```
kubectl apply -f app-deployment.yaml
```

Verifica que los pods están corriendo

```
kubectl get pods
```

```
(base) wilsonorlandoquesadamoncayo@MAC-WQUESAD ks8 % kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
desafio11-app-5df875f6ff-v7tv4	0/1	ContainerCreating	0	70s
desafio11-app-5ff8ff5f4c-7j8bk	0/1	ImagePullBackOff	0	3m53s
mongodb-5cd4dbd46b-2lgct	1/1	Running	0	4m

```
kubectl get all
```

```
(base) wilsonorlandoquesadamoncayo@MAC-WQUESAD ks8 % kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/desafio11-app-5df875f6ff-v7tv4	1/1	Running	0	9m26s
pod/mongodb-5cd4dbd46b-2lgct	1/1	Running	0	12m
pod/prueba	0/1	ImagePullBackOff	0	14d

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/desafio11-service	NodePort	10.103.62.56	<none>	3000:30001/TCP	12m
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	32d
service/mongodb	ClusterIP	10.104.28.162	<none>	27017/TCP	12m

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/desafio11-app	1/1	1	1	12m
deployment.apps/mongodb	1/1	1	1	12m

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/desafio11-app-5df875f6ff	1	1	1	9m27s
replicaset.apps/desafio11-app-5ff8ff5f4c	0	0	0	12m
replicaset.apps/mongodb-5cd4dbd46b	1	1	1	12m

Para acceder a el servicio de Kubernetes desde <http://localhost:30002>, se puede usar el comando `kubectl port-forward`.

Este comando te permite redirigir el tráfico desde un puerto local a un puerto en un pod o servicio en el clúster de Kubernetes.

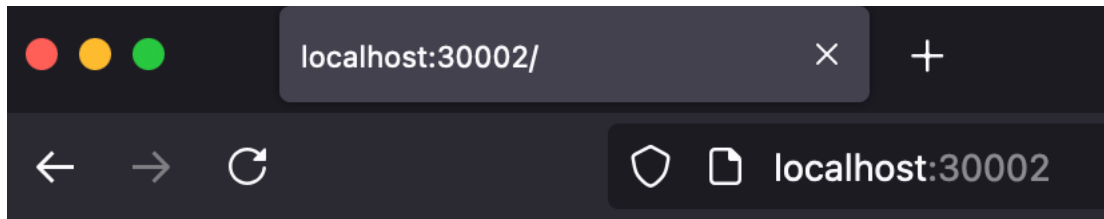
```
kubectl port-forward service/desafio11-service 30002:3000
```

```
(base) wilsonorlandoquesadamoncayo@MAC-WQUESAD ks8 % kubectl port-forward service/desafio11-service 30002:3000
Forwarding from 127.0.0.1:30002 -> 3000
Forwarding from [::1]:30002 -> 3000
```

- **Verifica Funcionamiento y Resultados exitosos**

Para probar la aplicación se puede en un navegador web o con una herramienta como curl o Postman para hacer una solicitud GET a `http://localhost:30002`.

Debería recibir una respuesta con el string 'Hello World!'.



Hello World!