

# Desafio #8

## Objetivo:

El siguiente desafío tiene como objetivo poner en práctica como desplegar una aplicación y los servicios necesarios en un entorno de kubernetes, en este desafío vamos a conectar todo lo realizado en el desafío anterior con Kubernetes.

## Escenario:

Luego del trabajo realizado en el sprint anterior y por el aporte que realizamos para mejorar la experiencia de desarrollo nuestro equipo nos encargó, transformar los servicios de docker-compose en un manifiesto de kubernetes, para eso debemos contemplar realizar el deployment de la misma aplicación y hacer que esta se conecte con su propia base de datos.

La aplicación que va a ser manejada por este proceso se encuentra en el siguiente enlace:

<https://github.com/yosoyfunes/app-template-nestjs>

## Requisitos:

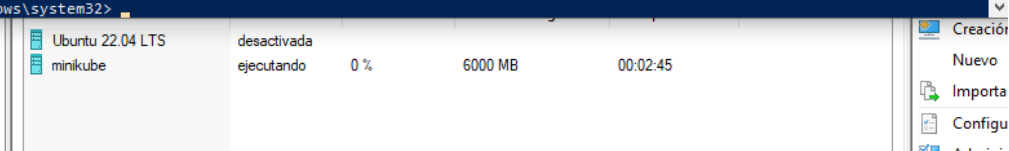
1. Elaborar los manifiestos necesarios para desplegar la aplicación vista en el desafío #5.
2. Elaborar toda la documentación necesaria.

En este desafío, en base a que tuve inconvenientes con WSL2 y Minikube, tuve que utilizar Windows y Docker Desktop,

Inicio minikube start lo cuál me genera una VM en HyperV

```
Chocolatey installed 1/2 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

Warnings:
- kubernetes-cli - kubernetes-cli v1.32.1 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.
PS C:\Windows\system32> minikube start
* minikube v1.35.0 en Microsoft Windows 10 Pro 10.0.19045.5371 Build 19045.5371
* Controlador hyperv seleccionado automáticamente. Otras opciones: virtualbox, ssh
* Descargando la imagen de arranque de la VM
  > minikube-v1.35.0-amd64.iso....: 65 B / 65 B [-----] 100.00% ? p/s 0s
  > minikube-v1.35.0-amd64.iso: 345.38 MiB / 345.38 MiB 100.00% 21.95 MiB p
* Starting "minikube" primary control-plane node in "minikube" cluster
* Descargando Kubernetes v1.32.0 ...
  > preloaded-images-k8s-v18-v1...: 333.57 MiB / 333.57 MiB 100.00% 21.66 M
* Creando hyperv VM (CPUs=2, Memory=6000MB, Disk=20000MB) ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube VM
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Preparando Kubernetes v1.32.0 en Docker 27.4.0...
  - Generando certificados y llaves
  - Iniciando plano de control
  - Configurando reglas RBAC...
* Configurando CNI bridge CNI ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Complementos habilitados: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
PS C:\Windows\system32> kubectl get all
NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP    53s
PS C:\Windows\system32>
```



Instalamos Kubernetes Kompose

*\$ choco install kubernetes-kompose*

```
PS C:\Windows\system32> choco install kubernetes-kompose
Chocolatey v2.3.0
3 validations performed. 2 success(es), 1 warning(s), and 0 error(s).

Validation Warnings:
- A pending system reboot request has been detected, however, this is
being ignored due to the current Chocolatey configuration. If you
want to halt when this occurs, then either set the global feature
using:
  choco feature enable --name="exitOnRebootDetected"
or pass the option --exit-when-reboot-detected.

Installing the following packages:
kubernetes-kompose
By installing, you accept licenses for the packages.
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading kubernetes-kompose 1.35.0... 100%

kubernetes-kompose v1.35.0 [Approved]
kubernetes-kompose package files install completed. Performing other installation steps.
ShimGen has successfully created a shim for kompose.exe
The install of kubernetes-kompose was successful.
  Deployed to 'C:\ProgramData\chocolatey\lib\kubernetes-kompose'

Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\Windows\system32>
```

Nos movemos al directorio de trabajo, una vez colocados los archivos del desafío 5, utilizamos el siguiente comando;

*\$ kompose convert*

```
Administrador: Windows PowerShell
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 8> kompose convert
INFO Kubernetes file "app-service.yaml" created
INFO Kubernetes file "mongodb-service.yaml" created
INFO Kubernetes file "app-deployment.yaml" created
INFO Kubernetes file "mongodb-deployment.yaml" created
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 8> 
```

Aplico manifiestos a minikube:

*\$ kubectl apply -f.*

```
Administrador: Windows PowerShell
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 8> kubectl apply -f .
deployment.apps/app created
service/app created
deployment.apps/mongodb created
service/mongodb created
error: error validating "docker-compose.yml": error validating data: [apiVersion not set, kind not set]; if you choose to ignore these errors, turn validation off with --validate=false
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 8> 
```

Error; Da error al subir docker-compose.yml

*\*Kubernetes no realiza automaticamente el proceso de Build, como si lo hace Docker Compose.*

Solución; Desplegamos buildamos a Docker Hub y configuramos docker-compose.yml

Comandos a utilizar;

*\$ docker tag app:latest briangirod/app:latest*

*\$ docker push briangirod/app:latest*

```
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 8> docker tag app:latest briangirod/app:latest
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 8> docker push briangirod/app:latest
The push refers to repository [docker.io/briangirod/app]
0ba67e019412: Pushed
37892ffbfc9a: Pushed
5650d6de56fd: Pushed
6504e29600c8: Pushed
a764067976b6: Pushed
e2468c591d72: Pushed
41713cf41c9e: Pushed
80a0a26b0712: Pushed
1f3e46996e29: Pushed
latest: digest: sha256:2e5cb3be5d47133dea19e571cbd9a23c6e3c9f88b401d3864afe06081ea4fc6b size: 856
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 8> 
```

Realizamos cambios en docker-compose.yml para apuntar al Docker Hub

```
docker-compose.yml X ! mongodb-service.yaml
docker-compose.yml
1  version: '3.8'
   ▶ Run All Services
2  services:
   ▶ Run Service
3      mongodb:
4          image: mongo:latest
5          container_name: mongodb
6          ports:
7              - "27017:27017"
   ▶ Run Service
8      app:
9          image: briangirod/app:latest
10         container_name: app
11         ports:
12             - "3000:3000"
```

Aplicamos los cambios nuevamente;

`$ kubectl apply -f.`

y ejecutamos el siguiente comando para verificar que esté corriendo;

`$ kubectl get all`

Funcionando;

```
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 8> kubectl get all
NAME                                READY   STATUS    RESTARTS   AGE
pod/app-666dcdbd45-96v96            1/1     Running   0           4m59s
pod/mongodb-7b7ff49bcd-x222d        1/1     Running   0           4m59s

NAME                                TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
service/app                         ClusterIP     10.101.94.100 <none>        3000/TCP    23m
service/kubernetes                  ClusterIP     10.96.0.1    <none>        443/TCP     32m
service/mongodb                     ClusterIP     10.107.229.240 <none>        27017/TCP   23m

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/app                 1/1     1             1           23m
deployment.apps/mongodb             1/1     1             1           23m

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/app-666dcdbd45      1         1         1       4m59s
replicaset.apps/app-696d566f86      0         0         0       23m
replicaset.apps/app-788c98bbcd       0         0         0       18m
replicaset.apps/mongodb-5b568ccc58   0         0         0       18m
replicaset.apps/mongodb-7b7ff49bcd   1         1         1       23m
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 8> kubectl port-forward service/app 3000:3000
Forwarding from 127.0.0.1:3000 -> 3000
Forwarding from [::1]:3000 -> 3000
Handling connection for 3000
Handling connection for 3000
```

Ingresamos a la base de datos con los siguientes comandos;

```
$ kubectl run -it --rm --image=mongo:latest mongo-client -- bash
```

y dentro de mongo-client;

```
$ mongosh mongodb://mongodb:27017
```

Mongodb

```
PS C:\Windows\system32> kubectl run -it --rm --image=mongo:latest mongo-client -- bash
If you don't see a command prompt, try pressing enter.
root@mongo-client:/# mongo mongodb://mongodb:27017
bash: mongo: command not found
root@mongo-client:/# mongosh mongodb://mongodb:27017
Current Mongosh Log ID: 67a668e16c0c38bcde544ca6
Connecting to:      mongodb://mongodb:27017/?directConnection=true&appName=mongosh+2.3.8
Using MongoDB:      8.0.4
Using Mongosh:      2.3.8

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

-----
The server generated these startup warnings when booting
2025-02-07T20:03:51.669+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2025-02-07T20:03:52.828+00:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2025-02-07T20:03:52.828+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
2025-02-07T20:03:52.828+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
2025-02-07T20:03:52.828+00:00: We suggest setting the contents of sysfsFile to 0.
2025-02-07T20:03:52.828+00:00: Your system has glibc support for rseq built in, which is not yet supported by tcmalloc-google and has critical performance implications. Please set the environment variable GLIBC_TUNABLES=glibc.pthread.rseq=0
2025-02-07T20:03:52.828+00:00: vm.max_map_count is too low
-----
test> _
```

Y por último, ingreso al pod y realizo una solicitud http a la DB para comprobar la comunicación con el servicio

```
$ kubectl exec -it pod/app-666dcdbd45-96v96 -- /bin/sh
```

y realizo una petición HTTP a la DB

Pruebo con curl pero la misma no estaba instalado, pruebo con wget;

```
$ wget -qO- 10.107.229.240:27017
```

Administrador: Windows PowerShell

```
PS C:\Windows\system32> kubectl get all
NAME                                READY    STATUS    RESTARTS   AGE
pod/app-666dcdbd45-96v96            1/1      Running   0           11m
pod/mongodb-7b7ff49bcd-x222d        1/1      Running   0           11m

NAME                                TYPE          CLUSTER-IP    EXTERNAL-IP   PORT(S)        AGE
service/app                          ClusterIP      10.101.94.100  <none>         3000/TCP       29m
service/kubernetes                   ClusterIP      10.96.0.1     <none>         443/TCP        38m
service/mongodb                      ClusterIP      10.107.229.240 <none>         27017/TCP      29m

NAME                                READY    UP-TO-DATE   AVAILABLE   AGE
deployment.apps/app                  1/1      1             1           29m
deployment.apps/mongodb              1/1      1             1           29m

NAME                                DESIRED    CURRENT   READY   AGE
replicaset.apps/app-666dcdbd45       1           1         1       11m
replicaset.apps/app-696d566f86       0           0         0       29m
replicaset.apps/app-788c98bbcd       0           0         0       24m
replicaset.apps/mongodb-5b568ccc58   0           0         0       24m
replicaset.apps/mongodb-7b7ff49bcd   1           1         1       29m

PS C:\Windows\system32> kubectl exec -it pod/app-666dcdbd45-96v96 -- /bin/sh
app # curl 10.107.229.240:27017
/bin/sh: curl: not found
app # wget -qO- 10.107.229.240:27017
it looks like you are trying to access MongoDB over HTTP on the native driver port.
app #
```