

# Desafio # 8

Fecha de entrega: 15/01/2025

## 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.

## Entregables:

Los entregables establecidos para este proyecto con:

1. Código fuente de todo lo producido.
2. Documentación.

3. Evidencia de las pruebas con resultado exitoso.
- 4.

## Evaluacion:

- Entrega en fecha.
- Redactar documentación legible y que sea comprendida por terceros..
- Añade material de soporte adicional.
  - Ejemplo: Diagrama de alto nivel.
- Cumple con las consignas solicitadas.
- El entregable es funcional.
  - Ejemplo: el script bash al ejecutarse funciona sin errores y realiza lo solicitado.

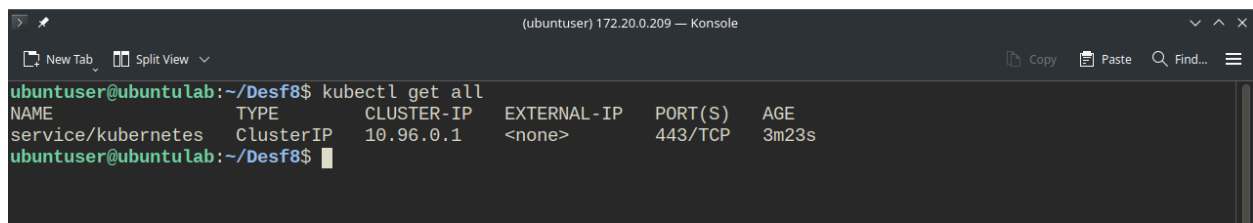
## Documentos de referencia:

- [Documentación NestJS.](#)
- [Manifiesto Kubernetes](#)
- [Kompose](#)

## Solución:

### Setup:

Voy a desplegar minikube en un server de laboratorio. siguiendo estas instrucciones <https://minikube.sigs.k8s.io/docs/start/?arch=%2Flinux%2Fx86-64%2Fstable%2Fbinary+download>



```
(ubuntuuser) 172.20.0.209 — Konsole
New Tab Split View
Copy Paste Find...
ubuntuuser@ubuntu:~/Desf8$ kubectl get all
NAME                TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
service/kubernetes  ClusterIP   10.96.0.1    <none>        443/TCP    3m23s
ubuntuuser@ubuntu:~/Desf8$
```

Instalo Kompose siguiendo las instrucciones en <https://kompose.io/installation/#installation> y verifico con `kompose version`

```

ubuntu@ubuntu:~/Desf8$ curl -L https://github.com/kubernetes/kompose/releases/download/v1.35.0/kompose-linux-amd64 -o kompose
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--    0
100 22.5M 100 22.5M    0     0  8373k      0  0:00:02  0:00:02 --:--:-- 14.0M
ubuntu@ubuntu:~/Desf8$ chmod +x kompose
sudo mv ./kompose /usr/local/bin/kompose
[sudo] password for ubuntu:
ubuntu@ubuntu:~/Desf8$ kompose version
1.35.0 (9532ceef3)
ubuntu@ubuntu:~/Desf8$

```

## Convertir docker-compose.yml a Kubernetes:

Luego de copiar los archivos al directorio de trabajo se usa `kompose convert` que generará los archivos YAML (deployment.yaml, service.yaml, etc.).

```

(ubuntu) 172.20.0.209 — Konsole
New Tab Split View Copy Paste Find...
ubuntu@ubuntu:~/Desf8$ ll
total 20
drwxrwxr-x 3 ubuntu docker 4096 Jan 30 14:21 ./
drwxr-x-- 16 ubuntu ubuntu 4096 Jan 30 14:11 ../
drwxr-xr-x 5 ubuntu ubuntu 4096 Jan 30 14:21 app-template-nestjs/
-rw-r--r-- 1 ubuntu ubuntu 197 Jan 30 14:21 docker-compose.yml
-rw-r--r-- 1 ubuntu ubuntu 667 Jan 30 14:21 Dockerfile
ubuntu@ubuntu:~/Desf8$ 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
ubuntu@ubuntu:~/Desf8$ ll
total 36
drwxrwxr-x 3 ubuntu docker 4096 Jan 30 14:23 ./
drwxr-x-- 16 ubuntu ubuntu 4096 Jan 30 14:11 ../
-rw-r--r-- 1 ubuntu docker 612 Jan 30 14:23 app-deployment.yaml
-rw-r--r-- 1 ubuntu docker 291 Jan 30 14:23 app-service.yaml
drwxr-xr-x 5 ubuntu ubuntu 4096 Jan 30 14:21 app-template-nestjs/
-rw-r--r-- 1 ubuntu ubuntu 197 Jan 30 14:21 docker-compose.yml
-rw-r--r-- 1 ubuntu ubuntu 667 Jan 30 14:21 Dockerfile
-rw-r--r-- 1 ubuntu docker 642 Jan 30 14:23 mongodb-deployment.yaml
-rw-r--r-- 1 ubuntu docker 306 Jan 30 14:23 mongodb-service.yaml
ubuntu@ubuntu:~/Desf8$

```

## Aplicar los manifiestos a Minikube:

Aplico con `kubectl apply -f .` (el `.` es para que aplique todos los manifiestos, desafortunadamente también intenta aplicar el docker-compose.yml, de ahí el error que sigue)

```

ubuntu@ubuntu:~/Desf8$ 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
ubuntu@ubuntu:~/Desf8$

```

## Verificación:

al hacer `kubectl get all` veo que los recursos se han aprovisionado, pero los pods no están en estado Ready.

```
(ubuntuuser) 172.20.0.209 — Konsole
New Tab Split View
ubuntuuser@ubuntulab:~/Desf8$ kubectl get all
NAME                                READY    STATUS             RESTARTS   AGE
pod/app-c5f856879-9nfv7             0/1     ErrImagePull       0          14s
pod/mongodb-6c8947597-5dn4c         0/1     Error              0          14s

NAME                                TYPE          CLUSTER-IP      EXTERNAL-IP    PORT(S)        AGE
service/app                         ClusterIP      10.103.103.163  <none>         3000/TCP       14s
service/kubernetes                  ClusterIP      10.96.0.1       <none>         443/TCP        31m
service/mongodb                     ClusterIP      10.111.47.47    <none>         27017/TCP      14s

NAME                                READY    UP-TO-DATE    AVAILABLE   AGE
deployment.apps/app                 0/1     1              0          14s
deployment.apps/mongodb             0/1     1              0          14s

NAME                                DESIRED    CURRENT    READY   AGE
replicaset.apps/app-c5f856879       1          1          0      14s
replicaset.apps/mongodb-6c8947597  1          1          0      14s
```

## Troubleshooting:

Si bien el proceso funcionó los pods no se estan desplegando correctamente. Vamos a revisarlos de a uno:

### 1.Pod app

Viendo el error “ErrImagePull” en el pod de app me di cuenta que voy a tener que hacer el build de la imagen y el push a dockerhub, Ya que kubernetes no hace el build (como si lo hacía docker-compose) así que, en el directorio donde estan los archivos de aplicación y el Dockerfile hago:

```
docker build -t app:latest .
docker tag app:latest jcavaiuolo/app:latest
docker push jcavaiuolo/app:latest
```

```
(ubuntuser) 172.20.0.209 — Konsole
New Tab Split View
Copy Paste Find...
ubuntuser@ubuntulab:~/Desf8/dockerfiles$ docker build -t app:latest .
[+] Building 121.8s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 706B
=> [internal] load metadata for docker.io/library/node:18-alpine
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/5] FROM docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd0 14.6s
=> => resolve docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd06 0.3s
=> => sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25 7.67kB / 7.67kB 0.0s
=> => sha256:6e804119c3884fc5782795bf0d2adc89201c63105aece8647b17a7bcebbbc385e 1.72kB / 1.72kB 0.0s
=> => sha256:dcbf7b337595be6f4d214e4eed84f230eefe0e4ac03a50380d573e289b9e5e40 6.18kB / 6.18kB 0.0s
=> => sha256:1f3e46996e2966e4faa5846e56e76e3748b7315e2ded61476c24403d592134f0 3.64MB / 3.64MB 1.1s
=> => sha256:37892ffbfcaa871a10f813803949d18c3015a482051d51b7e0da02525e63167c 40.01MB / 40.01MB 3.8s
=> => sha256:5650d6de56fd0bb419872b876ac1df28f577b39573c3b72fb0d15bf426d01bc1 1.26MB / 1.26MB 3.2s
=> => extracting sha256:1f3e46996e2966e4faa5846e56e76e3748b7315e2ded61476c24403d592134f0 1.7s
=> => sha256:6504e29600c8d5213b52cda800370abb3d12639802d06b46b6fce368990ca771 444B / 444B 2.7s
=> => extracting sha256:37892ffbfcaa871a10f813803949d18c3015a482051d51b7e0da02525e63167c 6.8s
=> => extracting sha256:5650d6de56fd0bb419872b876ac1df28f577b39573c3b72fb0d15bf426d01bc1 0.2s
=> => extracting sha256:6504e29600c8d5213b52cda800370abb3d12639802d06b46b6fce368990ca771 0.0s
=> [internal] load build context
=> => transferring context: 430.12kB
=> [2/5] WORKDIR /app
=> [3/5] COPY /app-template-nestjs/* .
=> [4/5] RUN npm install
=> [5/5] RUN npm run build
=> exporting to image
=> => exporting layers
=> => writing image sha256:c608d6212b027d1c74ef08083f5f60ac67c9c83e53dea2c47d637723b03b989e
=> => naming to docker.io/library/app:latest
ubuntuser@ubuntulab:~/Desf8/dockerfiles$ docker login

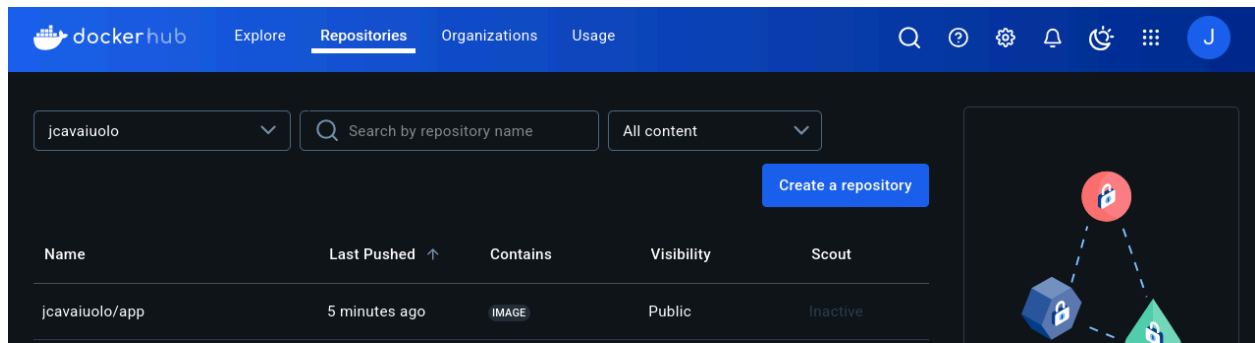
USING WEB-BASED LOGIN
To sign in with credentials on the command line, use 'docker login -u <username>'

Your one-time device confirmation code is: XNZK-6XTZ
Press ENTER to open your browser or submit your device code here: https://login.docker.com/activate

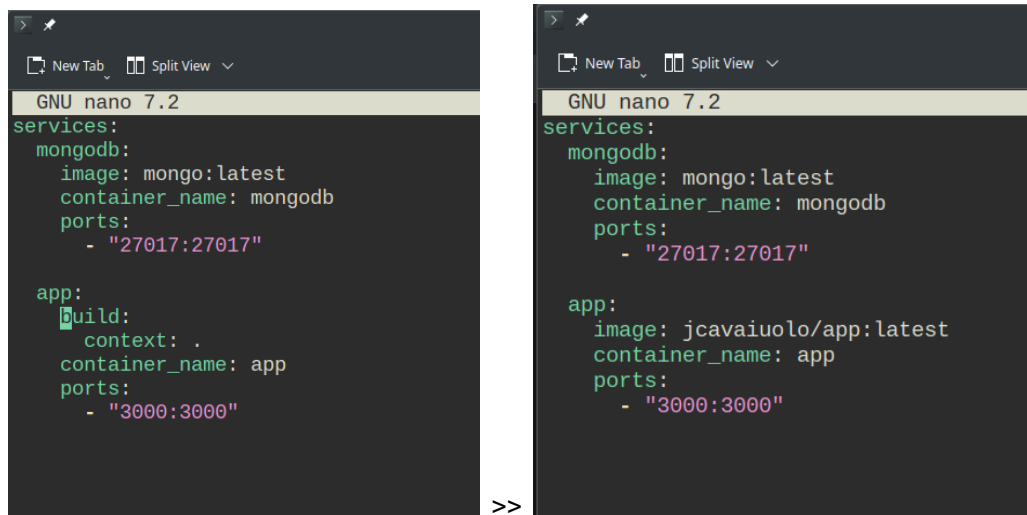
Waiting for authentication in the browser...
WARNING! Your password will be stored unencrypted in /home/ubuntuser/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credential-stores

Login Succeeded
ubuntuser@ubuntulab:~/Desf8/dockerfiles$ docker tag app:latest jcavaiuolo/app:latest
ubuntuser@ubuntulab:~/Desf8/dockerfiles$ docker push jcavaiuolo/app:latest
The push refers to repository [docker.io/jcavaiuolo/app]
92cae6f4e874: Pushed
88740bd7672d: Pushed
5d89d98defcf: Pushed
f4580807b6b7: Pushed
f3778221ab51: Mounted from library/node
8e0b3f8ca374: Mounted from library/node
d561411347c1: Mounted from library/node
a0904247e36a: Mounted from library/node
latest: digest: sha256:808e0fceed1aaad89365ebea04925e60e938a121fe74287c5f8a8e34922095dc size: 1995
ubuntuser@ubuntulab:~/Desf8/dockerfiles$
```

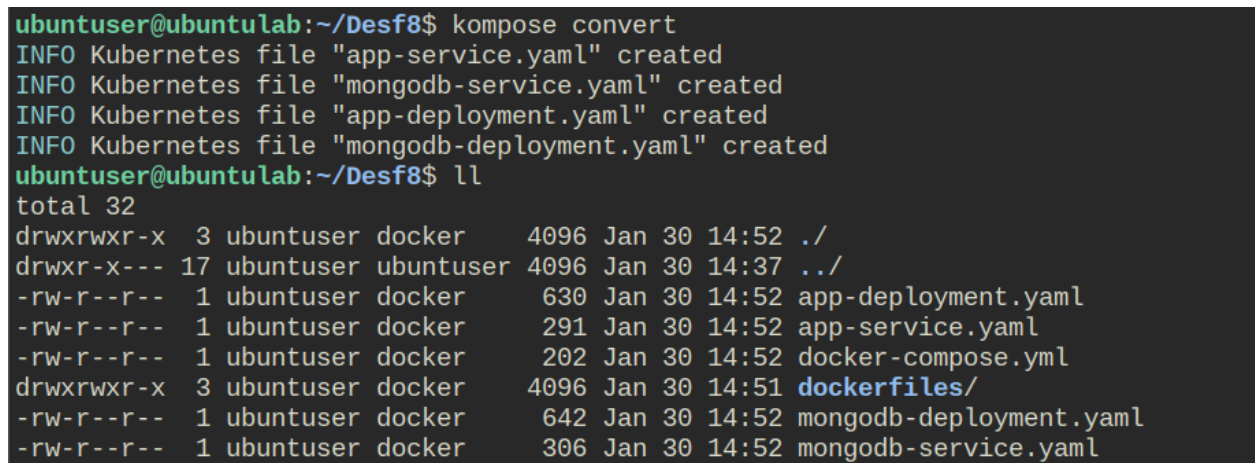
y verifico que la imagen se encuentre disponible en la interfaz web:



Modifico el archivo docker-compose para usar esta imagen y reconvertir con kompose:



Y lo convierto nuevamente con `kompose convert`



Aplico con `kubectl apply -f .`

```
(ubuntuuser) 172.20.0.209 — Konsole

New Tab Split View

ubuntuuser@ubuntu:~/Desf8$ kubectl apply -f .
deployment.apps/app created
service/app created
deployment.apps/mongodb created
service/mongodb created

ubuntuuser@ubuntu:~/Desf8$ kubectl get all
NAME                                READY    STATUS              RESTARTS   AGE
pod/app-546ccc4dd6-8fq55            1/1      Running             0           42s
pod/mongodb-6c8947597-r9lhx         0/1      CrashLoopBackOff    1 (14s ago) 42s

NAME                                TYPE               CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
service/app                        ClusterIP          10.104.117.6  <none>         3000/TCP         42s
service/kubernetes                 ClusterIP          10.96.0.1     <none>         443/TCP          44m
service/mongodb                    ClusterIP          10.103.172.5  <none>         27017/TCP        42s

NAME                                READY    UP-TO-DATE    AVAILABLE   AGE
deployment.apps/app                 1/1      1              1           42s
deployment.apps/mongodb             0/1      1              0           42s

NAME                                DESIRED    CURRENT    READY   AGE
replicaset.apps/app-546ccc4dd6      1          1          1       42s
replicaset.apps/mongodb-6c8947597  1          1          0       42s
ubuntuuser@ubuntu:~/Desf8$
```

## 2.Pod mongodb:

Al hacer `kubectl logs pod/mongodb-6c8947597-r9lhx` llego al siguiente output:

```
ubuntuuser@ubuntu:~/Desf8$ kubectl logs pod/mongodb-6c8947597-r9lhx
WARNING: MongoDB 5.0+ requires a CPU with AVX support, and your current system does not appear to have that!
see https://jira.mongodb.org/browse/SERVER-54407
see also https://www.mongodb.com/community/forums/t/mongodb-5-0-cpu-intel-g4650-compatibility/116610/2
see also https://github.com/docker-library/mongo/issues/485#issuecomment-891991814
```

El problema es que el CPU no tiene **soporte para AVX**, lo cual es un requisito para MongoDB 5.0 y versiones posteriores. Workaround: usar una versión anterior de MongoDB que no requiera AVX, como la 4.4. Así que actualizo una vez más el archivo `docker-compose`:



```
GNU nano 7.2
services:
  mongodb:
    image: mongo:latest
    container_name: mongodb
    ports:
      - "27017:27017"

  app:
    image: jcavaiuolo/app:latest
    container_name: app
    ports:
      - "3000:3000"

>>
```

Luego lo convierto nuevamente con `kompose convert` y aplico una vez mas con `kubectl apply -f .`

```
(ubuntuser) 172.20.0.209 — Konsole
ubuntuser@ubuntulab:~/Desf8$ kubectl get all
NAME                                READY   STATUS    RESTARTS   AGE
pod/app-546ccc4dd6-n2864            1/1     Running   0           7m13s
pod/mongodb-567b7775d4-ccdvm       1/1     Running   6 (3m58s ago)  7m12s

NAME                                TYPE               CLUSTER-IP    EXTERNAL-IP   PORT(S)    AGE
service/app                        ClusterIP          10.106.255.116 <none>        3000/TCP    7m13s
service/kubernetes                 ClusterIP          10.96.0.1     <none>        443/TCP    61m
service/mongodb                    ClusterIP          10.108.229.156 <none>        27017/TCP   7m12s

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/app                 1/1     1             1           7m13s
deployment.apps/mongodb             1/1     1             1           7m12s

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/app-546ccc4dd6      1         1         1       7m13s
replicaset.apps/mongodb-567b7775d4  1         1         1       7m12s
ubuntuser@ubuntulab:~/Desf8$
```

## 2.nodeport para app:

Acá me di cuenta de que el service app no estaba exponiendo el puerto hacia fuera del cluster asi que modifiko `app-service.yaml` para incluir el `type:nodeport`:

```
GNU nano 7.2
apiVersion: v1
kind: Service
metadata:
  annotations:
    kompose.cmd: kompose convert
    kompose.version: 1.35.0 (9532ceef3)
  labels:
    io.kompose.service: app
  name: app
spec:
  ports:
    - name: "3000"
      port: 3000
      targetPort: 3000
  selector:
    io.kompose.service: app
```

>>

```
GNU nano 7.2 app-service.y
apiVersion: v1
kind: Service
metadata:
  annotations:
    kompose.cmd: kompose convert
    kompose.version: 1.35.0 (9532ceef3)
  labels:
    io.kompose.service: app
  name: app
spec:
  type: NodePort
  ports:
    - name: "3000"
      port: 3000 # puerto "dentro" del cluster
      targetPort: 3000 # puerto que escucha el contenedor
      nodePort: 30000 # puerto en la máquina local
  selector:
    io.kompose.service: app
```

Y aplico con `kubectl apply -f .`.  
Éste es estado final del cluster:

```
(ubuntuuser) 172.20.0.209 — Konsole
New Tab Split View
Copy

ubuntuuser@ubuntu:lab:~/Desf8$ kubectl get all
NAME                                READY   STATUS    RESTARTS   AGE
pod/app-546ccc4dd6-n2864            1/1     Running   0           75m
pod/mongodb-567b7775d4-ccdvm       1/1     Running   6 (72m ago) 75m

NAME                                TYPE               CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
service/app                        NodePort           10.106.255.116 <none>        3000:30000/TCP   75m
service/kubernetes                 ClusterIP          10.96.0.1    <none>         443/TCP           129m
service/mongodb                    ClusterIP          10.108.229.156 <none>        27017/TCP        75m

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/app                 1/1     1             1           75m
deployment.apps/mongodb              1/1     1             1           75m

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/app-546ccc4dd6      1         1         1       75m
replicaset.apps/mongodb-567b7775d4 1         1         1       75m
ubuntuuser@ubuntu:lab:~/Desf8$
```

## Verificación #2:

Ahora si, al hacer un curl al nodeport en la ip de minikube recibo la respuesta de la aplicacion:

```
(ubuntuuser) 172.20.0.209 — Konsole
New Tab Split View
Copy Paste Find...

ubuntuuser@ubuntu:~/Desf8$ kubectl get all
NAME                                READY   STATUS    RESTARTS   AGE
pod/app-546ccc4dd6-n2864            1/1     Running   0           82m
pod/mongodb-567b7775d4-ccdvm        1/1     Running   6 (79m ago) 82m

NAME                                TYPE               CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
service/app                        NodePort           10.106.255.116 <none>        3000:30000/TCP   82m
service/kubernetes                  ClusterIP           10.96.0.1     <none>         443/TCP           136m
service/mongodb                     ClusterIP           10.108.229.156 <none>        27017/TCP        82m

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/app                 1/1     1             1           82m
deployment.apps/mongodb             1/1     1             1           82m

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/app-546ccc4dd6      1         1         1       82m
replicaset.apps/mongodb-567b7775d4  1         1         1       82m
ubuntuuser@ubuntu:~/Desf8$ minikube ip
192.168.67.2
ubuntuuser@ubuntu:~/Desf8$ curl 192.168.67.2
curl: (7) Failed to connect to 192.168.67.2 port 80 after 0 ms: Couldn't connect to server
ubuntuuser@ubuntu:~/Desf8$ curl 192.168.67.2:30000
Hello World!ubuntuuser@ubuntu:~/Desf8$
```

y para verificar que la app “vea” el contenedor de la base de datos me voy a conectar por ssh y hacer un curl a la ip del contenedor de mongo en el puerto 27017:

```
(ubuntuuser) 172.20.0.209 — Konsole
New Tab Split View
Copy Paste Find...

ubuntuuser@ubuntu:~/Desf8$ kubectl get all
NAME                                READY   STATUS    RESTARTS   AGE
pod/app-546ccc4dd6-n2864            1/1     Running   0           88m
pod/mongodb-567b7775d4-ccdvm        1/1     Running   6 (84m ago) 88m

NAME                                TYPE               CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
service/app                        NodePort           10.106.255.116 <none>        3000:30000/TCP   88m
service/kubernetes                  ClusterIP           10.96.0.1     <none>         443/TCP           142m
service/mongodb                     ClusterIP           10.108.229.156 <none>        27017/TCP        88m

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/app                 1/1     1             1           88m
deployment.apps/mongodb             1/1     1             1           88m

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/app-546ccc4dd6      1         1         1       88m
replicaset.apps/mongodb-567b7775d4  1         1         1       88m
ubuntuuser@ubuntu:~/Desf8$ kubectl exec -it pod/app-546ccc4dd6-n2864 -- /bin/sh
/app # curl 10.108.229.156:27017
It looks like you are trying to access MongoDB over HTTP on the native driver port.
/app #
```