# Desafío #9

Fecha de entrega: 22/01/2025

# Objetivo:

El objetivo de este desafío es poner en práctica lo visto sobre helm y desarrollar nuestro propio helm chart, tomando como entrada lo visto en los desafíos 5 y 8.

### **Escenario:**

Nuestro equipo identificó que Kubernetes agrega mucho valor a la hora de mantener los deployments de nuestra aplicación y ha estado analizando la manera en que gestionamos el código de estos deployments.

Luego de varias reuniones, identificaron que los manifiestos duplican mucho código y que esto se puede resolver utilizando algún sistema de templates. Durante este sprint, se nos asignó la tarea de desarrollar un Helm chart para gestionar el deployment de la aplicación. Este chart debe desplegar la aplicación y el servicio de base de datos MongoDB para almacenar los datos de nuestra aplicación.

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 el Chart para realizar el deployment.

2. Redactar 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.

## Evaluación:

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

### Documentos de referencia:

- <u>Documentación NestJS.</u>
- Manifiesto Kubernetes
- Helm Chart Development Tips

#### Solución:

#### Setup:

Voy a instalar helm en mi server Ubuntu siguiendo las instrucciones de aqui: https://helm.sh/docs/intro/install/#from-script

## Implementación:

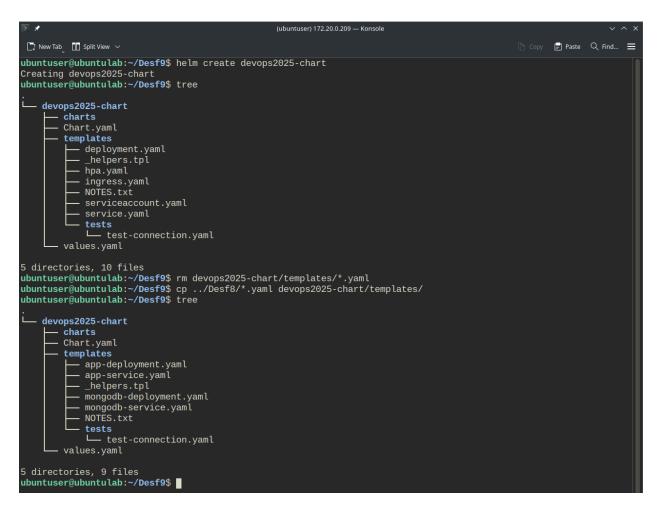
Voy a crear el directorio del chart con helm create devops2025-chart

```
ubuntuser@ubuntulab:~/Desf9$ helm create devops2025-chart
Creating devops2025-chart
ubuntuser@ubuntulab:~/Desf9$ tree
  devops2025-chart
      - charts
       Chart.yaml
        templates
           deployment.yaml
            _helpers.tpl
            hpa.yaml
           ingress.yaml
           NOTES.txt
            serviceaccount.yaml
            service.yaml
            tests

    test-connection.yaml

       - values.yaml
5 directories, 10 files
ubuntuser@ubuntulab:~/Desf9$
```

Voy a eliminar todos los .yaml en el directorio templates y reemplazarlos por los yaml del desafío anterior.



#### Validación:

En este punto ya se puede instalar el chart con: helm install devops2025-chart ./devops2025-chart

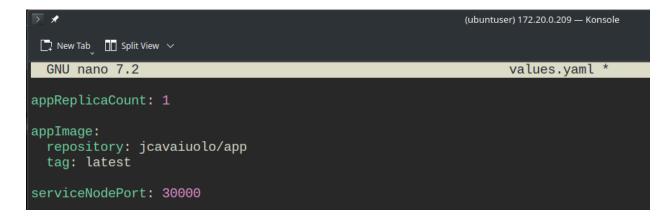
Verificando con kubectl get all se verifica que el estado del cluster coincide con el esperado (del desafío anterior)

```
(ubuntuser) 172.20.0.209 — Konsole
    New Tab Split View V
                                                                                                                                                                                                                                                                     Copy Paste Q Find... ≡
 ubuntuser@ubuntulab:~/Desf9$ helm install devops2025-chart ./devops2025-chart
 NAME: devops2025-chart
LAST DEPLOYED: Thu Jan 30 20:09:24 2025
  NAMESPACE: default
 STATUS: deployed
 REVISION: 1
NOTES:
1. Get the application URL by running these commands:
    export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=devops2025-chart,app.kubernet
    es.io/instance=devops2025-chart" -o jsonpath="{.items[0].metadata.name}")
    export CONTAINER_PORT=$(kubectl get pod --namespace default $POD_NAME -o jsonpath="{.spec.containers[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0].ports[0
  ].containerPort}")
      echo "Visit http://127.0.0.1:8080 to use your application"
      kubectl --namespace default port-forward $POD_NAME 8080:$CONTAINER_PORT
  ubuntuser@ubuntulab:~/Desf9$ kubectl get all
                                                                                                        STATUS
                                                                                       READY
                                                                                                                                          RESTARTS
                                                                                                                                                                        AGE
 pod/app-546ccc4dd6-zh8xl
                                                                                                              Running
 pod/mongodb-567b7775d4-wgnwl 1/1
                                                            TYPE
                                                                                             CLUSTER-IP
                                                                                                                                             EXTERNAL-IP
                                                                                                                                                                                     PORT(S)
                                                                                                                                                                                                                                    AGE
                                                            NodePort
                                                                                             10.97.239.93
                                                                                                                                                                                     3000:30000/TCP
 service/app
                                                                                                                                             <none>
 service/kubernetes
                                                           ClusterIP
                                                                                             10.96.0.1
                                                                                                                                                                                     443/TCP
                                                                                                                                                                                                                                     5h59m
                                                                                                                                             <none>
                                                                                             10.102.181.184
                                                                                                                                                                                    27017/TCP
 service/mongodb
                                                           ClusterIP
                                                                                                                                             <none>
                                                                         READY
                                                                                                UP-TO-DATE
                                                                                                                                    AVAILABLE
 deployment.apps/app
                                                                         1/1
 deployment.apps/mongodb
                                                                                                        DESIRED
                                                                                                                                    CURRENT
                                                                                                                                                                READY
                                                                                                                                                                                       AGE
 replicaset.apps/app-546ccc4dd6
                                                                                                                                                                                       24s
  replicaset.apps/mongodb-567b7775d4
                                                                                                                                                                                        24s
ubuntuser@ubuntulab:~/Desf9$
```

### Implementación de variables:

voy a pasar algunas variables al archivo de values.yaml. voy a modificar los manifiestos. Para la app voy a definir:

- nombre de la imagen
- número de réplicas
- puerto nodeport



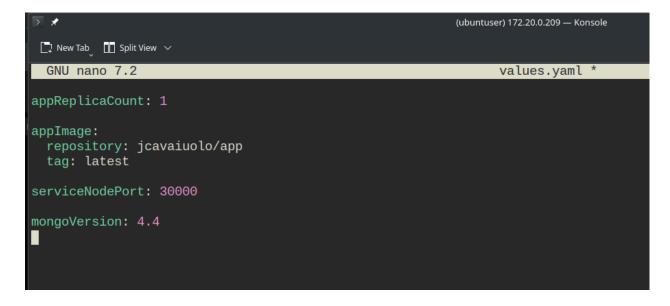
y ajusto los manifiestos de app-deployment y app-service:

```
apiVersion: apps/v1
kind: Deployment
   metadata
         annotations:
            kompose.cmd: kompose convert
kompose.version: 1.35.0 (9532ceef3)
       io.kompose.service: app
name: app
                                                                                                                                                                                                         replicas: {{ .Values.appReplicaCount }}
             matchLabels:
        io.kompose.service: app template:
              metadata:
               annotations:
kompose.cmd: kompose convert
kompose.version: 1.35.0 (9532ceef3)
            containers:
- image: jcavaiuolo/app:latest
name: app
                                                                                                                                                                                              containers:
- image: "{{ .Values.appImage.repository }}:{{ .Values.appImage.tag }}"
- name: app
- ports:
- restrictions for the containers and applications are containers.
                - containerPort: 3000
| protocol: TCP
restartPolicy: Always
                                                                                              ! app-service.yaml • ! app-service.yaml.bak -- app-service.yaml •
io.kompose.service: app
name: app
        spec:
type: NodePort  # Cambiar el tipo a NodePort

ports:
- name: "3000"
port: 3000  # El puerto dentro del clúster
targetPort: 3000  # El puerto que escucha el contenedor de la apr
nodePort: 30000  # El puerto asignado en la máquina local
                                                                                                                                                                                               10 spec:
11 type: NodePort  # Cambiar el tipo a NodePort
12 ports:
13 - name: "3000"
14 port: 3000  # El puerto dentro del clúster
15 targetPort: 3000  # El puerto que escucha el contenedor de la app
16+ nodePort: {{ .Values.serviceNodePort }} # El puerto asignado en ]
                            mpose.service: app
```

Y, dado que en el ejercicio anterior me dio problemas la versión de la imagen de mongodo voy a usar la versión como variable en el deployment para el pod de mongo:

agregando mongolmageTag: 4.4 al values.yaml queda así:



y actualizo el manifiesto del deployment para que use esta versión:

```
Deef) deverya2025 chart > templates > ! mongodb deployment.yaml

1 apiVersion: apps/v1
2 kl.nd: Deployment
3 metadata:
4 annotations:
5 kompose.cend: kompose convert
6 kompose.version: 1.35.0 (9532ceef3)
7 labels:
8 io. kompose service: mongodb
9 name: mongodb
10 spec:
11 replicas: 1
12 selector:
13 matchlabels:
14 io. kompose.service: mongodb
15 template:
16 kompose.version: 1.35.0 (9532ceef3)
17 labels:
18 kompose.cend: kompose convert
19 kompose.version: percentage of the percentage of the
```

### **Troubleshooting:**

Como borré el archivo values yaml, algunas de las funciones preconstruidas en el helm chart dejaron de funcionar. Así que tuve que hacer algunos ajustes a las variables. La versión final del archivo values quedó así:

```
(ubuntuser) 172.20.0.209 — Konsole
New Tab Split View V
ubuntuser@ubuntulab:~/Desf9$ helm lint ./devops2025-chart/
==> Linting ./devops2025-chart/
[INFO] Chart.yaml: icon is recommended
1 chart(s) linted, 0 chart(s) failed
ubuntuser@ubuntulab:~/Desf9$ nano devops2025-chart/values.yaml
ubuntuser@ubuntulab:~/Desf9$ cat devops2025-chart/values.yaml
appReplicaCount: 2
appImage:
 repository: jcavaiuolo/app
 tag: latest
serviceNodePort: 30002  # esta variable quedaria mejor en la seccion "service"
mongoVersion: 4.4
ingress:
 enabled: false
service:
 type: NodePort
ubuntuser@ubuntulab:~/Desf9$
```

Nota: cambie el número de réplicas a 2 y el nodeport a 30002 solo para poder ver el cambio en el estado del cluster en el paso de verificación.

#### Verificación:

Luego de hacer helm install devops2025-chart ./devops2025-chart y verificar el STATUS: deployed. Se puede ejecutar kubectl get all y se verifica el despliegue. Particularmente los 2 pods de app, y el nodeport seteado a 30002 en el paso anterior.

```
(ubuntuser) 172.20.0.209 — Konsole
 ☐ New Tab ☐ Split View ∨
                                                                                                                                                              🖹 Paste 🔍 Find... 🗏
 ubuntuser@ubuntulab:~/Desf9$ kubectl get all
NAME TYPE CLUSTER-IP service/kubernetes ClusterIP 10.96.0.1
                                                              EXTERNAL-IP
                                                                                  443/TCP
                                                             <none>
 ubuntuser@ubuntulab:~/Desf9$ helm install devops2025-chart ./devops2025-chart
NAME: devops2025-chart
 LAST DEPLOYED: Thu Jan 30 21:20:34 2025
NAMESPACE: default
STATUS: deployed
REVISION: 1
NOTES:
1. Get the application URL by running these commands:
export NODE_PORT=$(kubectl get --namespace default -o jsonpath="{.spec.ports[0].nodePort}" services devops2025-chart)
export NODE_IP=$(kubectl get nodes --namespace default -o jsonpath="{.items[0].status.addresses[0].address}")
echo http://$NODE_IP:$NODE_PORT
ubuntuser@ubuntulab:~/Desf9$ kubectl get all
                                          READY
                                                                  RESTARTS
pod/app-546ccc4dd6-8dlmd
                                                    Running
                                                                                  56s
pod/app-546ccc4dd6-w2chv
                                                     Running
pod/mongodb-567b7775d4-fktgm
                                             CLUSTER-IP
                                                                    EXTERNAL-IP
                                                                                       3000:30002/TCP
443/TCP
service/app
service/kubernetes
                            NodePort
                            ClusterIP
                                             10.96.0.1
                                                                    <none>
                                                                                                              7h10m
                                             10.110.151.166
 service/mongodb
                                              UP-TO-DATE
                                                               AVAILABLE
                                                                                AGE
NAME
deployment.apps/app
                                  2/2
1/1
deployment.apps/mongodb
                                                  DESIRED
                                                               CURRENT
                                                                             READY
                                                                                        57s
57s
replicaset.apps/app-546ccc4dd6
replicaset.apps/mongodb-567b77<u>75d4</u>
 ubuntuser@ubuntulab:~/Desf9$
```

al obtener la ip de minikube con minikube ip. se puede hacer un curl al puerto expuesto en el cluster y se valida la comunicación con la aplicación:

```
ubuntuser@ubuntulab:~/Desf9$ minikube ip
192.168.67.2
ubuntuser@ubuntulab:~/Desf9$ curl 192.168.67.2:30002
Hello World!ubuntuser@ubuntulab:~/Desf9$ ■
```