Desafio #9

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

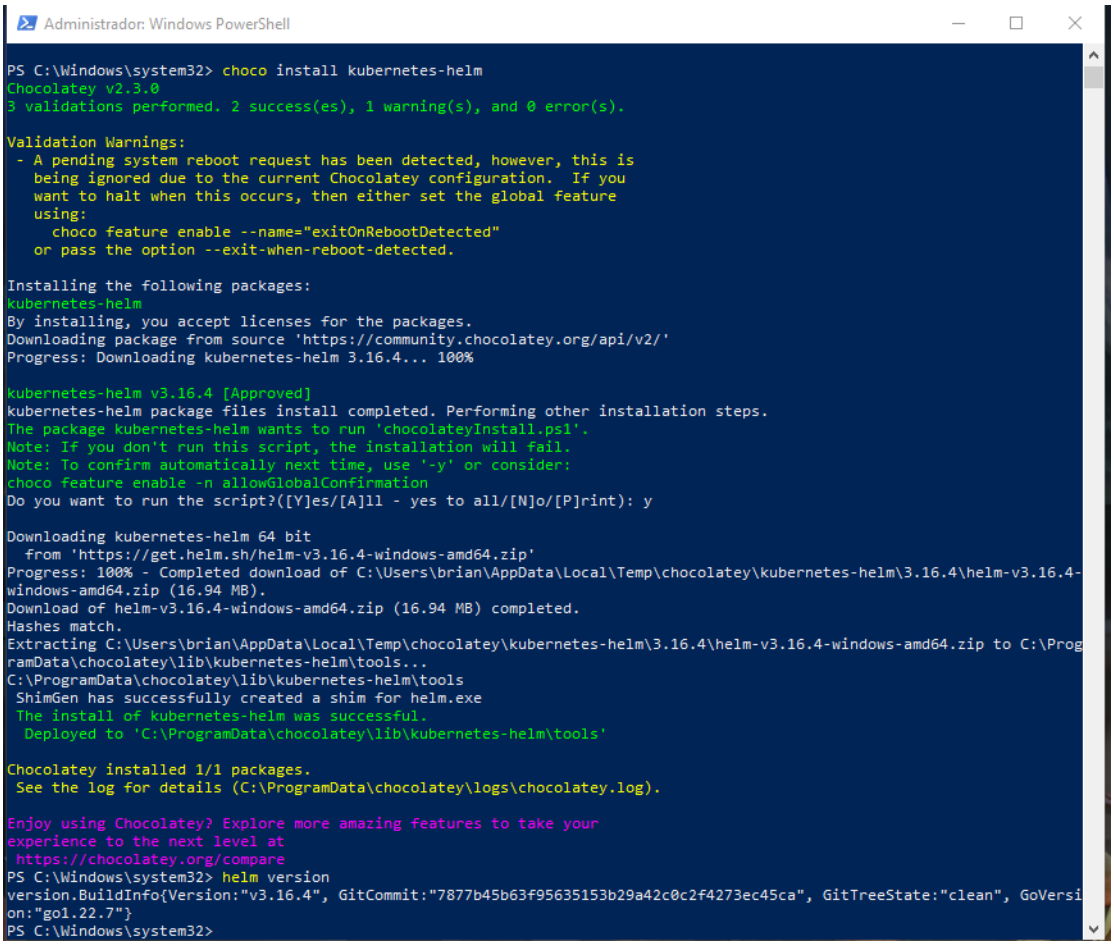
Requisitos:

1. Elaborar el Chart para realizar el deployment.
2. Redactar la documentación necesaria.

Instalación de Helm

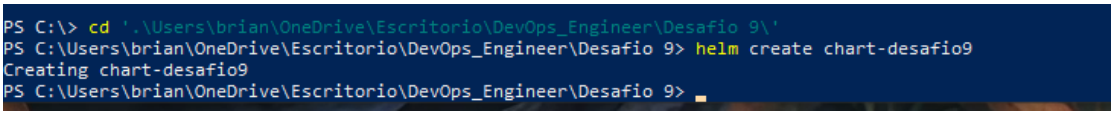
Utilizo comando;

$ choco install kubernetes-helm



Creamos el chart

$ helm create chart-desafio9



Eliminamos todos los .yaml del directorio templates

```
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 9\chart-desafio9> dir


    Directorio: C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 9\chart-desafio9


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          8/2/2025     01:30                charts
d-----          8/2/2025     01:30                templates
-a----          8/2/2025     01:30            349 .helmignore
-a----          8/2/2025     01:30           1150 Chart.yaml
-a----          8/2/2025     01:30           4300 values.yaml


PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 9\chart-desafio9> cd .\templates\
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 9\chart-desafio9\templates> rm *.yaml
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 9\chart-desafio9\templates> dir


    Directorio: C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 9\chart-desafio9\templates


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          8/2/2025     01:30                tests
-a----          8/2/2025     01:30           1772 NOTES.txt
-a----          8/2/2025     01:30           1852 _helpers.tpl


PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 9\chart-desafio9\templates>
```

Movemos todos los .yaml del Desafio 8 al directorio de trabajo

```
Administrador: Windows PowerShell
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 9> cd ..
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer> cp '.\Desafio 8\*.yaml' '.\Desafio 9\chart-desafio9\templates\'
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer> cd '.\Desafio 9\chart-desafio9\templates\'
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 9\chart-desafio9\templates> dir


    Directorio: C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 9\chart-desafio9\templates


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          8/2/2025     01:30                tests
-a----          7/2/2025     17:05            748 app-deployment.yaml
-a----          7/2/2025     17:05            350 app-service.yaml
-a----          7/2/2025     17:05            760 mongodb-deployment.yaml
-a----          7/2/2025     17:05            365 mongodb-service.yaml
-a----          8/2/2025     01:30           1772 NOTES.txt
-a----          8/2/2025     01:30           1852 _helpers.tpl


PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 9\chart-desafio9\templates>
```

Instalamos el chart;

$ helm install chart-desafio9 .\chart-desafio9

```
Administrador: Windows PowerShell
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 9> helm install chart-desafio9 .\chart-desafio9\
NAME: chart-desafio9
LAST DEPLOYED: Sat Feb  8 01:41:42 2025
NAMESPACE: default
STATUS: deployed
REVISION: 1
NOTES:
1. Get the application URL by running these commands:
  export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=chart-desafio9,app.kubernetes.io/ins
tance=chart-desafio9" -o jsonpath="{.items[0].metadata.name}")
  export CONTAINER_PORT=$(kubectl get pod --namespace default $POD_NAME -o jsonpath="{.spec.containers[0].ports[0].conta
inerPort}")
  echo "Visit http://127.0.0.1:8080 to use your application"
  kubectl --namespace default port-forward $POD_NAME 8080:$CONTAINER_PORT
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 9> kubectl get all
NAME                         READY   STATUS    RESTARTS   AGE
pod/app-666dcdbd45-6tngq     1/1     Running   0          16s
pod/mongodb-7b7ff49bcd-jl944   1/1     Running   0          16s

NAME                 TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)     AGE
service/app          ClusterIP   10.108.194.50   <none>        3000/TCP    16s
service/kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP     9h
service/mongodb      ClusterIP   10.99.75.132    <none>        27017/TCP   16s

NAME                     READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/app      1/1     1            1           16s
deployment.apps/mongodb   1/1     1            1           16s

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/app-666dcdbd45      1         1         1       16s
replicaset.apps/mongodb-7b7ff49bcd   1         1         1       16s
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 9>
```

**Error no documentado; tenía servicios y deployments del Desafio 8 con los mismos nombres que desplega el chart, que quedaron seteados e iniciaron junto a minikube, tuve que eliminarlos y volver a correr el helm install*

Una vez instalado, testeamos con petición HTTP;

```
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 9> kubectl get all
NAME                         READY   STATUS    RESTARTS   AGE
pod/app-666dcdbd45-lqhrz     1/1     Running   0          5s
pod/mongodb-7b7ff49bcd-n645q   1/1     Running   0          5s

NAME                 TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)     AGE
service/app          ClusterIP   10.96.187.182   <none>        3000/TCP    5s
service/kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP     9h
service/mongodb      ClusterIP   10.97.68.11     <none>        27017/TCP   5s

NAME                     READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/app      1/1     1            1           5s
deployment.apps/mongodb   1/1     1            1           5s

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/app-666dcdbd45      1         1         1       5s
replicaset.apps/mongodb-7b7ff49bcd   1         1         1       5s
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 9> curl 10.96.187.182:3000
curl : No es posible conectar con el servidor remoto
En línea: 1 Carácter: 1
+ curl 10.96.187.182:3000
+ ~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : InvalidOperation: (System.Net.HttpWebRequest:HttpWebRequest) [Invoke-WebRequest], WebException
    + FullyQualifiedErrorId : WebCmdletWebResponseException,Microsoft.PowerShell.Commands.InvokeWebRequestCommand

PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 9> minikube ip
172.24.96.4
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 9> curl 172.24.96.4:3000
curl : No es posible conectar con el servidor remoto
En línea: 1 Carácter: 1
+ curl 172.24.96.4:3000
+ ~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : InvalidOperation: (System.Net.HttpWebRequest:HttpWebRequest) [Invoke-WebRequest], WebException
    + FullyQualifiedErrorId : WebCmdletWebResponseException,Microsoft.PowerShell.Commands.InvokeWebRequestCommand
```

Error;

**Este error que se presentaba era motivo de que el servicio app quedaba con ClusterIP, y necesitaba configuración NodePort, para solucionar esto;*

```
Events:                  <none>
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 9> helm upgrade --install chart-desafio9 .\chart-desafio9\ --set service
.type=NodePort
Release "chart-desafio9" has been upgraded. Happy Helming!
NAME: chart-desafio9
LAST DEPLOYED: Sat Feb  8 02:04:27 2025
NAMESPACE: default
STATUS: deployed
REVISION: 3
NOTES:
1. Get the application URL by running these commands:
  export NODE_PORT=$(kubectl get --namespace default -o jsonpath="{.spec.ports[0].nodePort}" services chart-desafio9)
  export NODE_IP=$(kubectl get nodes --namespace default -o jsonpath="{.items[0].status.addresses[0].address}")
  echo http://$NODE_IP:$NODE_PORT
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 9> kubectl describe service app
Name:                     app
Namespace:                default
Labels:                   app.kubernetes.io/managed-by=Helm
                          io.kompose.service=app
Annotations:              kompose.cmd: C:\ProgramData\chocolatey\lib\kubernetes-kompose\tools\kompose.exe convert
                          kompose.version: 1.35.0 (9532ceef3)
                          meta.helm.sh/release-name: chart-desafio9
                          meta.helm.sh/release-namespace: default
Selector:                 io.kompose.service=app
Type:                     ClusterIP
IP Family Policy:         SingleStack
IP Families:              IPv4
IP:                       10.96.187.182
IPs:                      10.96.187.182
Port:                     3000  3000/TCP
TargetPort:               3000/TCP
Endpoints:                10.244.0.16:3000
Session Affinity:         None
Internal Traffic Policy:  Cluster
Events:                   <none>
PS C:\Users\brian\OneDrive\Escritorio\DevOps_Engineer\Desafio 9>
```

Configuración anterior de app-service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  annotations:
    kompose.cmd: C:\ProgramData\chocolatey\lib\kubernetes-kompose\tools\kompose.exe 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
```

Modificamos por;

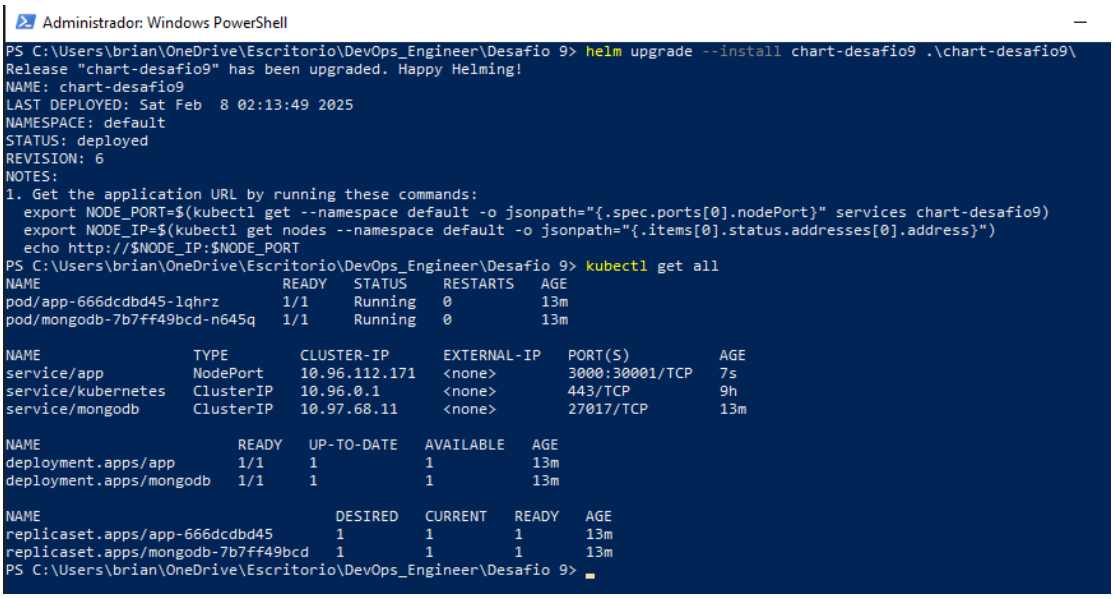
templates > ! app-service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: app
  labels:
    io.kompose.service: app
spec:
  type: NodePort
  ports:
    - name: "3000"
      port: 3000
      targetPort: 3000
      nodePort: 30001
  selector:
    io.kompose.service: app

```

Ejecutados comando para actualizar;

$ helm upgrade --install chart-desafio9 .\chart-desafio9

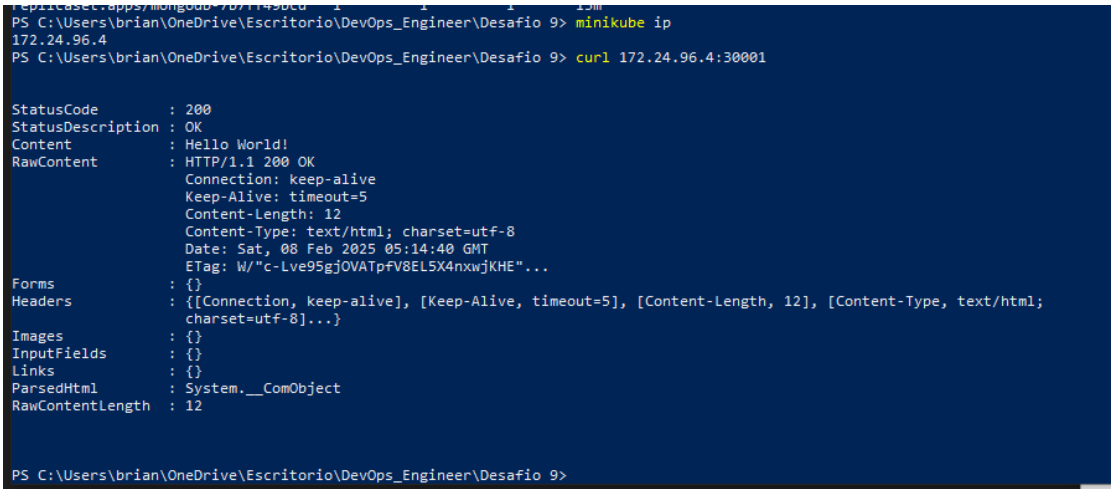


Obtenemos IP con comando;

$ minikube ip

Realizamos una petición HTTP;

$ curl 172.24.96.4:30001



Resultado; HTTP 200 OK