

Desafío 12

Bootcamp Devops Engineer

Desarrollo:

Para el desafío de crear un Helm chart para desplegar una aplicación NestJS junto con MongoDB, aquí se describen los pasos para estructurar el chart.

- **Estructura del Helm Chart**

Primero, crear la estructura básica del Helm chart. Se puede hacer usando el comando de Helm para crear un nuevo chart:

```
helm create desafio12-chart
```

```
(base) wilsonorlandoquesadamoncayo@mac-wquesad Desafio12 % helm create desafio12-chart
Creating desafio12-chart
```

- **Modificar y Crear los archivos yaml necesarios**

Edita el archivo values.yaml para definir los valores que pueden ser configurados.

```
desafio12-chart > values.yaml
1  replicaCount: 1
2
3  image:
4    repository: lwill1/desafio11
5    tag: "1"
6    pullPolicy: IfNotPresent
7
8  service:
9    type: NodePort
10   port: 3000
11   nodePort: 30001
12
13  mongo:
14    image:
15      repository: mongo
16      tag: latest
17    mongoUser: root
18    mongoPassword: password
19    mongoDatabase: app-desafio11
20    persistence:
21      enabled: true
22      storageClass: ""
23      size: 1Gi
24
```

Edita el archivo Chart.yaml para definir el chart.

```
desafio12-chart > Chart.yaml
1  apiVersion: v2
2  name: educacionit-chart
3  description: Helm chart para desplegar la aplicación EducacionIT y MongoDB
4  version: 0.1.0
5  appVersion: "1.0"
6
```

Crea el archivo deployment.yaml en la carpeta templates/:

```
desafio12-chart > templates > deployment.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: {{ .Release.Name }}-app
5  spec:
6    replicas: {{ .Values.replicaCount }}
7    selector:
8      matchLabels:
9        app: {{ .Release.Name }}-app
10   template:
11     metadata:
12       labels:
13         app: {{ .Release.Name }}-app
14     spec:
15       containers:
16         - name: desafio12
17           image: {{ .Values.image.repository }}:{{ .Values.image.tag }}
18           ports:
19             - containerPort: {{ .Values.service.port }}
20           env:
21             - name: MONGO_DB_URI
22               value: mongodb://mongodb:27017
23             - name: MONGO_DB_NAME
24               value: {{ .Values.mongo.mongoDatabase }}
25             - name: MONGO_DB_USER
26               value: {{ .Values.mongo.mongoUser }}
27             - name: MONGO_DB_PASS
28               value: {{ .Values.mongo.mongoPassword }}
29             - name: PORT
30               value: "{{ .Values.service.port }}"
```

Crea el archivo service.yaml en la carpeta templates/:

```
desafio12-chart > templates > service.yaml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: {{ include "desafio12-chart.fullname" . }}
5    labels:
6      {{- include "desafio12-chart.labels" . | nindent 4 }}
7  spec:
8    type: {{ .Values.service.type }}
9    ports:
10     - port: {{ .Values.service.port }}
11       targetPort: {{ .Values.service.port }}
12       nodePort: {{ .Values.service.nodePort }}
13       protocol: TCP
14       name: http
15    selector:
16      {{- include "desafio12-chart.selectorLabels" . | nindent 4 }}
```

mongo-service.yaml:

```
apiVersion: v1
kind: Service
metadata:
  name: {{ .Release.Name }}-mongodb
spec:
  type: ClusterIP
  ports:
    - port: 27017
      targetPort: 27017
      protocol: TCP
  selector:
    app: {{ .Release.Name }}-mongodb
```

- **Despliegue del Chart**

Iniciar Minikube:

minikube start

Instalar el Helm Chart:

helm install desafio12-release ./desafio12-chart

Verifica que los pods están corriendo

kubectl get all

NAME	READY	STATUS	RESTARTS	AGE
pod/desafio12-release-app-5b57b7c9c4-j65qb	1/1	Running	0	3m46s
pod/desafio12-release-mongodb-745478f795-zj59z	1/1	Running	0	33m
pod/prueba	0/1	ImagePullBackOff	0	15d

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/desafio12-release-educacionit-chart	NodePort	10.108.141.213	<none>	3000:30001/TCP	33m
service/desafio12-release-mongodb	ClusterIP	10.101.99.58	<none>	27017/TCP	3m46s
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	34d

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/desafio12-release-app	1/1	1	1	33m
deployment.apps/desafio12-release-mongodb	1/1	1	1	33m

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/desafio12-release-app-5b57b7c9c4	1	1	1	3m46s
replicaset.apps/desafio12-release-app-648d949d75	0	0	0	33m
replicaset.apps/desafio12-release-mongodb-745478f795	1	1	1	33m

Para acceder a el servicio de desde <http://localhost:30001>, se puede usar el comando `kubectl port-forward`.

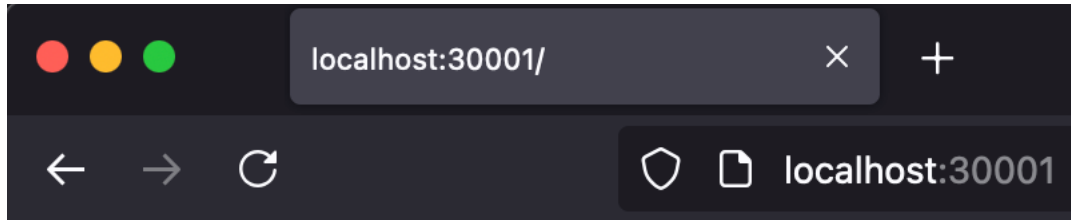
Este comando te permite redirigir el tráfico desde un puerto local a un puerto en un pod o servicio en el clúster de Kubernetes.

`kubectll port-forward desafio12-release-app-5b57b7c9c4-j65qb 30001:3000`

```
(base) wilsonorlandoquesadamoncayo@mac-wquesad Desafio12 % kubectl port-forward desafio12-release-app-5b57b7c9c4-j65qb 30001:3000
Forwarding from 127.0.0.1:30001 -> 3000
Forwarding from [::1]:30001 -> 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:30001`. Debería recibir una respuesta con el string 'Hello World!'.



Hello World!