

Desafío 12 - Bootcamp Devops Engineer

Alumno: Ignacio Peretti

Requisitos:

Elaborar un **HELM CHART** para el deployment de nuestra aplicación y redactar la documentación necesaria.

Requisitos para poder trabajar:

Tener instalado **Docker Desktop** y una cuenta en **Docker Hub** para alojar nuestra imagen en un repositorio.

En este caso se levantó un entorno de Kubernetes con **MINIKUBE** para realizar el trabajo.

Necesitaremos tener instalado **kubectl** para poder trabajar con la línea de comandos de kubernetes.

Comenzamos.

Una vez situados en nuestro espacio de trabajo, en este caso es la aplicación y la base de datos con la que trabajamos en los últimos dos desafíos.

Seguiremos con lo realizado en este último.

(<https://github.com/IgnacioPeretti/desafio11.git>)

1 - Crear un helm chart.

Para este paso utilizamos el comando

helm create educacionit-chart

```
PS C:\Users\CPU\Desktop\devops\devops-bootcamp\educacionitdevops\desafio10\chart-desafio12> helm create educacionit-chart
Creating educacionit-chart
PS C:\Users\CPU\Desktop\devops\devops-bootcamp\educacionitdevops\desafio10\chart-desafio12> |
```

Esto nos va a crear la estructura de nuestro chart y podemos adaptarlo a nuestras necesidades.

```
educacionit-chart/
├── charts
├── Chart.yaml
├── templates
│   ├── deployment.yaml
│   ├── _helpers.tpl
│   ├── hpa.yaml
│   ├── ingress.yaml
│   ├── NOTES.txt
│   ├── serviceaccount.yaml
│   └── service.yaml
└── values.yaml
```

Debemos agregar los archivos de deployment y servicio en la carpeta **templates** para nuestra base de datos.

Creada nuestra estructura debe contener los siguientes archivos.

Chart.yaml - Este archivo contiene metadatos sobre el Chart.

values.yaml - Este archivo define los valores configurables del chart, con él definimos los valores predeterminados para nuestra aplicación y la base de datos MongoDB.

deployment.yaml - Este archivo define el deployment para nuestra aplicación.

service.yaml - Este archivo define el servicio para nuestra aplicación.

mongodb-deployment.yaml - Este archivo define el deployment para MongoDB.

mongodb-service.yaml - Este archivo define el servicio para MongoDB

mongodb-data-persistentvolumeclaim.yml - define un PersistentVolumeClaim (PVC) en Kubernetes.

Chart.yaml

```
educacionit-chart > ! Chart.yaml
1  apiVersion: v2
2  name: educacionit-chart
3  description: A Helm chart for Kubernetes
4  type: application
5  version: 0.1.0
6  appVersion: "1.0"
7
```

values.yaml

```
educacionit-chart > ! values.yaml
1
2
3 replicaCount: 1
4
5 image:
6   repository: ignacioperetti/eduit-kubernetes
7   tag: latest
8   pullPolicy: IfNotPresent
9 service:
10   port: 3000
11   type: ClusterIP
12
13
14 mongodb:
15   enabled: true
16   image:
17     repository: mongo
18     tag: latest
19   service:
20     port: 27017
21     type: ClusterIP
22   auth:
23     enabled: true
24     username: root
25     password: s3cr3t!
26     database: desafio11-mongo
27
```

deployment.yaml

```
educacionit-chart > templates > ! deployment.yaml
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: educacionit-chart
5   labels:
6     app: educacionit-chart
7 spec:
8   replicas: {{ .Values.replicaCount }}
9   selector:
10     matchLabels:
11       app: educacionit-chart
12   template:
13     metadata:
14       labels:
15         app: educacionit-chart
16     spec:
17       containers:
18         - name: educacionit-chart
19           image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}"
20           ports:
21             - containerPort: 3000
22           env:
23             - name: MONGO_URI
24               value: "mongodb://{{ .Values.mongodb.auth.username }}:{{ .Values.mongodb.auth.password }}@mongodb:27017/{{ .Values.mongodb."
25
```

service.yaml

```
educacionit-chart > templates > ! service.yaml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: educacionit-chart
5  spec:
6    type: {{ .Values.service.type }}
7    ports:
8      - port: {{ .Values.service.port }}
9    selector:
10     app: educacionit-chart
11
12
```

mongodb-deployment.yaml

```
educacionit-chart > templates > ! mongodb-deployment.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: mongodb
5    labels:
6      app: mongodb
7  spec:
8    replicas: 1
9    selector:
10     matchLabels:
11       app: mongodb
12    template:
13     metadata:
14       labels:
15         app: mongodb
16     spec:
17       containers:
18         - name: mongodb
19           image: "{{ .Values.mongodb.image.repository }}:{{ .Values.mongodb.image.tag }}"
20           ports:
21             - containerPort: 27017
22           env:
23             - name: MONGO_INITDB_ROOT_USERNAME
24               value: "{{ .Values.mongodb.auth.username }}"
25             - name: MONGO_INITDB_ROOT_PASSWORD
26               value: "{{ .Values.mongodb.auth.password }}"
27             - name: MONGO_INITDB_DATABASE
28               value: "{{ .Values.mongodb.auth.database }}"
29           volumeMounts:
30             - name: mongodb-data
31               mountPath: /data/db
32
```

mongodb-service.yaml

```
educacionit-chart > templates > ! mongodb-service.yaml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: mongodb
5  spec:
6    ports:
7      - port: {{ .Values.mongodb.service.port }}
8        targetPort: 27017
9    selector:
10     app: mongodb
11     type: {{ .Values.mongodb.service.type }}
12
```

mongodb-data-persistentvolumeclaim.yml

```
educacionit-chart > templates > ! mongodb-data-persistentvolumeclaim.yaml
1  apiVersion: v1
2  kind: PersistentVolumeClaim
3  metadata:
4    name: mongodb-pvc
5  labels:
6    app: mongodb
7  spec:
8    accessModes:
9      - ReadWriteOnce
10   resources:
11     requests:
12       storage: 2Gi
```

Para poder hacer pruebas podemos ejecutar un lint de nuestro chart con el comando

helm lint educacionit-chart .

Finalmente podemos instalar nuestro chart.

helm install educacionit-chart educacionit-chart

Verifica el despliegue, asegúrate de que los recursos se han creado correctamente.

Comandos para comprobar el funcionamiento del chart.

```
PS C:\Users\CPU\Desktop\devops\devops-bootcamp\educacionitdevops\desafio10> helm install educacionit-chart educacionit-chart
NAME: educacionit-chart
LAST DEPLOYED: Tue Sep 17 00:07:33 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
```

helm list

```
PS C:\Users\CPU> helm ls
NAME                NAMESPACE    REVISION    UPDATED                               STATUS          CHART               APP VERSION
educacionit-chart   default       1           2024-09-17 00:07:33.3225569 -0300 -03  deployed       educacionit-chart-0.1.0 1.0
PS C:\Users\CPU>
```

kubectl get all

```
PS C:\Windows\system32> kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/educacionit-chart-545c5d7f9-5qfs5	1/1	Running	0	72s
pod/mongodb-595bb887cc-hjtgb	1/1	Running	0	72s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/educacionit-chart	ClusterIP	10.111.76.125	<none>	3000/TCP	72s
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	13d
service/mongodb	ClusterIP	10.96.183.241	<none>	27017/TCP	72s

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/educacionit-chart	1/1	1	1	72s
deployment.apps/mongodb	1/1	1	1	72s

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/educacionit-chart-545c5d7f9	1	1	1	72s
replicaset.apps/mongodb-595bb887cc	1	1	1	72s

```
PS C:\Windows\system32>
```

kubectl get deployments

```
PS C:\Windows\system32> kubectl get deployments
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
educacionit-chart	1/1	1	1	5m5s
mongodb	1/1	1	1	5m5s

```
PS C:\Windows\system32>
```