

DESAFÍO 12

Institución: Educación IT

Alumno: Becchero Mariano

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 10 y 11.

GUÍA DE PASOS

- 1- Como primer paso, vamos a crear los archivos Chart.yaml y values.yaml para definir las propiedades que luego vamos a utilizar en los templates.

Chart.yaml

```
1  apiVersion: v2
2  name: educacionit-app
3  description: A Helm chart for Kubernetes
4  type: application
5  version: 0.1.0
6  appVersion: "1.16.0"
```

values.yaml

```
1  replicaCount: 1
2
3  image:
4    repository: marianobecchero/educacionit-app
5    pullPolicy: IfNotPresent
6    tag: v1.0.0
7
8  imagePullSecrets: []
9
10 nameOverride: ""
11 fullnameOverride: ""
12
13 serviceAccount:
14   create: true
15   automount: true
16   annotations: {}
17   name: ""
18
19 podAnnotations: {}
20
21 podLabels: {}
22
23 podSecurityContext: {}
24
25 securityContext: {}
```

```
25   securityContext: {}
26
27   service:
28     type: ClusterIP
29     port: 3000
30
31   ingress:
32     enabled: false
33     className: ""
34     annotations: {}
35     hosts:
36       - host: chart-example.local
37         paths:
38           - path: /
39             pathType: ImplementationSpecific
40     tls: []
41
42   resources: {}
43
44   livenessProbe:
45     httpGet:
46       path: /
47       port: http
48   readinessProbe:
49     httpGet:
```

```

44   livenessProbe:
45     httpGet:
46       path: /
47       port: http
48   readinessProbe:
49     httpGet:
50       path: /
51       port: http
52
53   autoscaling:
54     enabled: false
55     minReplicas: 1
56     maxReplicas: 100
57     targetCPUUtilizationPercentage: 80
58
59   volumes: []
60
61   volumeMounts: []
62
63   nodeSelector: {}
64
65   tolerations: []
66
67   affinity: {}
68
69   mongoDB:
70     name: db-server
71
72   affinity: {}
73
74   mongoDB:
75     name: db-server
76     containerName: educacionit-app-db
77     volumeMountName: mongodb-data
78     portName: "27017"
79     port: 27017
80     env:
81       MONGO_INITDB_ROOT_USERNAME: "mariano"
82       MONGO_INITDB_ROOT_PASSWORD: "becchero"
83       MONGO_DB_NAME: "test"
84       MONGO_DB_URI: "mongodb://mariano:becchero@db-server-service:27017"

```

- 2- A partir de los manifiestos creados en el desafío 11, vamos a gestionar aún mejor el código creando templates que luego van a servir para nuestros Helm charts.

app-deployment.yaml

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: "{{ .Chart.Name }}" #educacionit-app
5  spec:
6    replicas: 1
7    selector:
8      matchLabels:
9        app: "{{ .Chart.Name }}" #educacionit-app
10   template:
11     metadata:
12       labels:
13         app: "{{ .Chart.Name }}" #educacionit-app
14     spec:
15       containers:
16         - name: "{{ .Chart.Name }}" #educacionit-app
17           image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}" #marianobecchero/educacionit-app:v1.0.0
18           env:
19             - name: MONGO_DB_NAME
20               value: "{{ .Values.mongodb.env.MONGO_DB_NAME }}" #test
21             - name: MONGO_INITDB_ROOT_USERNAME
22               value: "{{ .Values.mongodb.env.MONGO_INITDB_ROOT_USERNAME }}" #mariano
23             - name: MONGO_INITDB_ROOT_PASSWORD
24               value: "{{ .Values.mongodb.env.MONGO_INITDB_ROOT_PASSWORD }}" #becchero
25             - name: MONGO_DB_URI
26               value: "{{ .Values.mongodb.env.MONGO_DB_URI }}" #mongodb://mariano:becchero@db-server-service:27017
27       ports:
28         - containerPort: {{ .Values.service.port }} #3000
29         protocol: TCP
```

app-service.yaml

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: educacionit-app-service
5  spec:
6    ports:
7      - name: "3000"
8        port: {{ .Values.service.port }} #3000
9        targetPort: {{ .Values.service.port }} #3000
10   selector:
11     app: "{{ .Chart.Name }}" #educacionit-app
12   type: LoadBalancer
```

db-server-deployment.yaml

```

1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: "{{ .Values.mongodb.name }}" #db-server
5  spec:
6    replicas: {{ .Values.replicaCount }} #1
7    selector:
8      matchLabels:
9        app: "{{ .Values.mongodb.name }}" #db-server
10   template:
11     metadata:
12       labels:
13         app: "{{ .Values.mongodb.name }}" #db-server
14     spec:
15       containers:
16       - name: "{{ .Values.mongodb.name }}" #db-server
17         image: mongo:7.0
18         ports:
19         - containerPort: {{ .Values.mongodb.port }} #27017
20         env:
21         - name: MONGO_INITDB_ROOT_PASSWORD
22           value: "{{ .Values.mongodb.env.MONGO_INITDB_ROOT_PASSWORD }}" #becchero
23         - name: MONGO_INITDB_ROOT_USERNAME
24           value: "{{ .Values.mongodb.env.MONGO_INITDB_ROOT_USERNAME }}" #mariano

```

db-server-service.yaml

```

1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: db-server-service
5  spec:
6    ports:
7      - name: "27017"
8        port: {{ .Values.mongodb.port }} #27017
9        targetPort: {{ .Values.mongodb.port }} #27017
10   selector:
11     app: "{{ .Values.mongodb.name }}" #db-server

```

- 3- Una vez desarrollados los templates, desplegamos el chart con el siguiente comando:
helm install chart-desafio12 ./mychart

4- Vista de resultado exitoso

```
$ kubectl logs -f deployment.apps/educacionit-app
[Nest] 1 - 10/05/2024, 4:44:02 PM LOG [NestFactory] Starting Nest application...
[Nest] 1 - 10/05/2024, 4:44:03 PM LOG [InstanceLoader] MongooseModule dependencies initialized +287ms
[Nest] 1 - 10/05/2024, 4:44:03 PM LOG [InstanceLoader] ConfigHostModule dependencies initialized +1ms
[Nest] 1 - 10/05/2024, 4:44:03 PM LOG [InstanceLoader] AppModule dependencies initialized +0ms
[Nest] 1 - 10/05/2024, 4:44:03 PM LOG [InstanceLoader] ConfigModule dependencies initialized +1ms
[Nest] 1 - 10/05/2024, 4:44:02 PM LOG [NestFactory] Starting Nest application...
[Nest] 1 - 10/05/2024, 4:44:03 PM LOG [InstanceLoader] MongooseModule dependencies initialized +287ms
[Nest] 1 - 10/05/2024, 4:44:03 PM LOG [InstanceLoader] ConfigHostModule dependencies initialized +1ms
[Nest] 1 - 10/05/2024, 4:44:03 PM LOG [InstanceLoader] AppModule dependencies initialized +0ms
[Nest] 1 - 10/05/2024, 4:44:03 PM LOG [InstanceLoader] ConfigModule dependencies initialized +1ms
[Nest] 1 - 10/05/2024, 4:44:03 PM LOG [InstanceLoader] MongooseModule dependencies initialized +287ms
[Nest] 1 - 10/05/2024, 4:44:03 PM LOG [InstanceLoader] ConfigHostModule dependencies initialized +1ms
[Nest] 1 - 10/05/2024, 4:44:03 PM LOG [InstanceLoader] AppModule dependencies initialized +0ms
[Nest] 1 - 10/05/2024, 4:44:03 PM LOG [InstanceLoader] ConfigModule dependencies initialized +1ms
[Nest] 1 - 10/05/2024, 4:44:03 PM LOG [InstanceLoader] ConfigHostModule dependencies initialized +1ms
[Nest] 1 - 10/05/2024, 4:44:03 PM LOG [InstanceLoader] AppModule dependencies initialized +0ms
[Nest] 1 - 10/05/2024, 4:44:03 PM LOG [InstanceLoader] ConfigModule dependencies initialized +1ms
[Nest] 1 - 10/05/2024, 4:44:03 PM LOG [InstanceLoader] AppModule dependencies initialized +0ms
[Nest] 1 - 10/05/2024, 4:44:03 PM LOG [InstanceLoader] ConfigModule dependencies initialized +1ms
[Nest] 1 - 10/05/2024, 4:44:03 PM LOG [InstanceLoader] ConfigHostModule dependencies initialized +1ms
[Nest] 1 - 10/05/2024, 4:44:16 PM LOG [InstanceLoader] MongooseCoreModule dependencies initialized +13424ms
[Nest] 1 - 10/05/2024, 4:44:16 PM LOG [RoutesResolver] AppController {}: +14ms
[Nest] 1 - 10/05/2024, 4:44:16 PM LOG [RouterExplorer] Mapped {/, GET} route +6ms
[Nest] 1 - 10/05/2024, 4:44:16 PM LOG [RouterExplorer] Mapped {/, GET} route +6ms
[Nest] 1 - 10/05/2024, 4:44:16 PM LOG [NestApplication] Nest application successfully started +5ms
```