Crea un pod de forma declarativa con las siguientes especificaciones:

• Imagen: nginx

• Version: 1.19.4

• Label: app: nginx-server

• Limi

1. [Ingress Controller / Secrets] Crea los siguientes objetos de forma declarativa con las siguientes especificaciones:

• Imagen: nginx Version: 1.19.4

• 3 replicas

• Label: app: nginx-server

• Exponer el puerto 80 de los pods

• Limits:

CPU: 20 milicores

Memoria: 128Mi

• Requests:

CPU: 20 milicores

Memoria: 128Mi

Codigo Fuente:

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: nginx-server**

**labels:**

**app: nginx-server-deployment**

**spec:**

**replicas: 3**

**selector:**

**matchLabels:**

**app: nginx-server**

**template:**

**metadata:**

**labels:**

**app: nginx-server**

**spec:**

**containers:**

**- name: nginx-server**

**image: nginx:1.19.4**

**ports:**

**- containerPort: 80**

**resources:**

**requests:**

**memory: "128Mi"**

**cpu: "20m"**

**limits:**

**memory: "128Mi"**

**cpu: "20m"**

**---**

**apiVersion: v1**

**kind: Service**

**metadata:**

**name: service-nginx**

**spec:**

**selector:**

**app: nginx-server**

**ports:**

**- protocol: TCP**

**port: 80**

**targetPort: 80**

**---**

**apiVersion: networking.k8s.io/v1**

**kind: Ingress**

**metadata:**

**name: app-ingress**

**namespace: default**

**spec:**

**rules:**

**- host: ad.rodriguez.student.lasalle.com**

**http:**

**paths:**

**- path: "/"**

**pathType: Prefix**

**backend:**

**service:**

**name: service-nginx**

**port:**

**number: 80**

1. A continuación, tras haber expuesto el servicio en el puerto 80, se deberá acceder a la página principal de Nginx a través de la siguiente:

URL <http://ad.rodriguez.student.lasalle.com>

Resultado en el Navegador:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

* **Resultado ingresando por consola y ejecutando curl -k**

Se ingresa a uno de los pod creados en el replicaset del deployment a través del siguiente comando:

kubectl exec -it nginx-server-c8c84c768-dpp82 sh

y se ejecuta

curl -k <http://ad.rodriguez.student.lasalle.com>

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Una vez realizadas las pruebas con el protocolo HTTP, se pide acceder al servicio mediante la utilización del protocolo HTTPS, para ello:

* Crear un certificado mediante la herramienta OpenSSL u otra similar
* Crear un secret que contenga el certificado

Certificado generado

Texto

Descripción generada automáticamente

Secret generado

Texto

Descripción generada automáticamente

Se modofica el Ingress con el secret creado:

**apiVersion: networking.k8s.io/v1**

**kind: Ingress**

**metadata:**

**name: app-ingress**

**namespace: default**

**spec:**

**-tls:**

**- hosts:**

**- ad.rodriguez.student.lasalle.com**

**secretName: nginx-certs-keys**

**rules:**

**- host: ad.rodriguez.student.lasalle.com**

**http:**

**paths:**

**- path: "/"**

**pathType: Prefix**

**backend:**

**service:**

**name: service-nginx**

**port:**

**number: 80**

Validación:

curl -k https://ad.rodriguez.student.lasalle.com

Texto

Descripción generada automáticamente

2- [StatefulSet] Crear un StatefulSet con 3 instancias de MongoDB (ejemplo visto en clase) Se pide:

• Habilitar el clúster de MongoDB

Se crea el servicio headless para gestionar las Ip de los pod creados por el recurso:

apiVersion: v1

kind: Service

metadata:

name: mongodb-service

labels:

name: mongo

spec:

ports:

- port: 27017

targetPort: 27017

clusterIP: None

selector:

role: mongo

Se crea el recurso ReplicaSet con un StorageClass asociado:

apiVersion: apps/v1

kind: StatefulSet

metadata:

name: mongod

spec:

serviceName: mongodb-service

replicas: 3

selector:

matchLabels:

role: mongo

template:

metadata:

labels:

role: mongo

environment: test

replicaset: MainRepSet

spec:

terminationGracePeriodSeconds: 10

containers:

- name: mongod-container

image: mongo

command:

- "mongod"

- "--bind\_ip"

- "0.0.0.0"

- "--replSet"

- "MainRepSet"

resources:

requests:

cpu: 0.2

memory: 200Mi

ports:

- containerPort: 27017

volumeMounts:

- name: mongodb-persistent-storage-claim

mountPath: /data/db

volumeClaimTemplates:

- metadata:

name: mongodb-persistent-storage-claim

annotations:

volume.beta.kubernetes.io/storage-class: "standard" -- storageClass

spec:

accessModes: [ "ReadWriteOnce" ]

resources:

requests:

storage: 1Gi

Se empiezan a crear los contenedores por orden

Imagen que contiene Texto

Descripción generada automáticamente

Finalmente se crean las 3 replicas

Texto

Descripción generada automáticamente con confianza media

Se inicializa el cluster desde uno de los pod

kubectl exec -it mongod-0 mongo

Con el siguiente comando se configuran los pods para trabajar de manera conjunta

rs.initiate({\_id: "MainRepSet", version: 1, members: [ { \_id: 0, host: "mongod-0.mongodb-service.default.svc.cluster.local:27017" },

{ \_id: 1, host: "mongod-1.mongodb-service.default.svc.cluster.local:27017" },

{ \_id: 2, host: "mongod-2.mongodb-service.default.svc.cluster.local:27017" }]});

Texto

Descripción generada automáticamente

Stateful creados

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Volúmenes de persistencia creados

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

• Realizar una operación en una de las instancias a nivel de configuración y verificar que el cambio se ha aplicado al resto de instancias

Se actualiza la imagen de mongo de mongo a mongo:latest en el nodo mongod-0

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Se modifica la version de mongo a mongo:latest:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

Se realiza el replicaset dentro del cluster de uno de los nodos de mongoDB con el siguiente comando:

rs.initiate({\_id: "MainRepSet", version: 1, members: [ { \_id: 0, host: "mongod-0.mongodb-service.default.svc.cluster.local:27017" }, { \_id: 1, host: "mongod-1.mongodb-service.default.svc.cluster.local:27017" },{ \_id: 2, host: "mongod-2.mongodb-service.default.svc.cluster.local:27017" }]});

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

Se valida en uno de los nodos el cambio realizado:

Texto

Descripción generada automáticamente

• Diferencias que existiría si el montaje se hubiera realizado con el objeto de ReplicaSet

Los nombres de los pod tendrían un hash en vez de un estado creado de manera escalonada

Texto

Descripción generada automáticamente con confianza baja

Con el StateFulSet el estado se persiste por tanto se debe utilizar la instrucción volumeClaimTemplates la cual afirma en volúmenes persistentes para garantizar que puedan mantener el estado en los reinicios de los componentes.

3- [Horizontal Pod Autoscaler] Crea un objeto de kubernetes HPA, que escale a partir de las métricas CPU o memoria (a vuestra elección). Establece el umbral al 50% de CPU/memoria utilizada, cuando pase el umbral, automáticamente se deberá escalar al doble de replicas.

**Se crea el objeto HorizontalPodAutoscaler**

apiVersion: autoscaling/v2beta2

kind: HorizontalPodAutoscaler

metadata:

name: nginx

spec:

scaleTargetRef:

apiVersion: apps/v1

kind: Deployment

name: nginx

minReplicas: 1

maxReplicas: 10

metrics:

- type: Resource

resource:

name: cpu

target:

type: Utilization

averageUtilization: 50

- type: Resource

resource:

name: memory

target:

type: AverageValue

averageValue: 100Mi

Prueba de estres

kubectl run -i --tty load-generator --rm --image=busybox --restart=Never --

while true; do wget -q -O- http://10.103.188.178; done

Get svc

Texto

Descripción generada automáticamente

Gráfico, Texto

Descripción generada automáticamente con confianza media

Con cuatro replicas

Interfaz de usuario gráfica

Descripción generada automáticamente

Tabla

Descripción generada automáticamente con confianza baja

Imagen que contiene Gráfico

Descripción generada automáticamente