Describiendo un Deployment

Podemos crear un Deployment de forma imperativa utilizando un comando como

el siguiente (se podrían indicar muchos más parámetros de configuración que

podemos consultar en la documentación):

kubectl create deployment nginx --image nginx

Nosotros, sin embargo, vamos a seguir describiendo los recursos en un fichero yaml. En este caso para describir un Deployment de nginx podemos escribir un

fichero nginx-deployment.yaml:

apiVersion: apps/v1

kind: Deployment

metadata:

name: deployment-nginx

labels:

app: nginx

spec:

revisionHistoryLimit: 2

strategy:

type: RollingUpdate

replicas: 2

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- image: nginx

name: contendor-nginx

ports:

- name: http

containerPort: 80

La creación de un Deployment crea un ReplicaSet y los Pods correspondientes. Por lo tanto en la definición de un Deployment se define también el ReplicaSet asociado (los parámetros replicas, selector y template). Los atributos relacionados con el Deployment que hemos indicado en la definición son:

- revisionHistoryLimit: Indicamos cuántos ReplicaSets antiguos deseamos conservar, para poder realizar rollback a estados anteriores. Por defecto, es 10.
- strategy: Indica el modo en que se realiza una actualización del Deployment. Es decir, cuando modificamos la versión de la imagen del Deployment, se crea un ReplicaSet nuevo y ¿qué hacemos con los pods?:
 - o Recreate: elimina los Pods antiguos y crea los nuevos.
 - RollingUpdate: va creando los nuevos Pods, comprueba que funcionan y se eliminan los antiguos; es la opción por defecto.

Además, hemos introducido un nuevo parámetro al definir el contenedor del pod: con el parámetro ports hemos indicado el puerto que expone el contenedor (containerPort) y le hemos asignado un nombre (name).