

Gestionando los Deployment

En esta unidad vamos a trabajar con el recurso Deployment, vamos a crear un Deployment de un servidor nginx, usando el fichero yaml que hemos visto en la unidad anterior: [nginx-deployment.yaml](#).

Creación del Deployment

Cuando creamos un Deployment, se creará un ReplicaSet asociado, que creará y controlará los Pods que hayamos indicado.

kubectl apply -f nginx-deployment.yaml

kubectl get deploy,rs,pod

Para ver los recursos que hemos creado también podemos utilizar la instrucción:

kubectl get all

Esta orden muestra los Deployments, ReplicaSets, Pods y Services que tenemos creados en el cluster. Los Services lo estudiaremos en el siguiente módulo.

Escalado de los Deployments

Como ocurría con los ReplicaSets los Deployment también se pueden escalar, aumentando o disminuyendo el número de Pods asociados. Al escalar un Deployment estamos escalando el ReplicaSet asociado en ese momento:

kubectl scale deployment deployment-nginx --replicas=4

Otras operaciones

Si queremos acceder a la aplicación, podemos utilizar la opción de port-forward sobre el despliegue (de nuevo recordamos que no es la forma adecuada para acceder a un servicio que se ejecuta en un Pod, pero de momento no tenemos otra). En este caso si tenemos asociados más de un Pod, la redirección de puertos se hará sobre un solo Pod (no habrá balanceo de carga):

kubectl port-forward deployment/deployment-nginx 8080:80

Si queremos ver los logs generados en los Pods de un Deployment:

kubectl logs deployment/deployment-nginx

Si queremos obtener información detallada del recurso Deployment que hemos creado:

kubectl describe deployment/deployment-nginx

Eliminando el Deployment

Si eliminamos el Deployment se eliminarán el ReplicaSet asociado y los Pods que se estaban gestionando.

kubectl delete deployment deployment-nginx