

Gestionando los ReplicaSet

En esta unidad vamos a crear un recurso ReplicaSet que controlará un conjunto de recursos Pods. Para ello vamos a utilizar el fichero que estudiamos en la unidad anterior: [nginx-rs.yaml](#).

Creación del ReplicaSet

Aunque en la unidad anterior usamos `kubectl create` para crear los recursos de nuestro cluster, es recomendable usar `kubectl apply`. La diferencia es la forma en la que actuamos sobre el cluster:

- **Configuración imperativa de objetos:** La definición del objeto está guardada en un fichero Yaml y ejecutamos un comando imperativo. Posteriormente no podremos modificar el objeto, habrá que borrarlo y crearlo de nuevo. Ejemplos:
 - `kubectl create -f recurso.yaml`
 - `kubectl delete -f recurso.yaml`
- **Configuración declarativa de objetos:** No se definen las acciones a realizar. Cuando se aplica la configuración del objeto estamos indicando un estado deseado al que queremos llegar. Posteriormente si la definición cambia, podremos cambiar el objeto. Recomendado en producción. Ejemplo:
 - `kubectl apply -f recurso.yaml`

Por lo tanto para crear nuestro ReplicaSet, ejecutamos:

`kubectl apply -f nginx-rs.yaml`

Y podemos ver los recursos que se han creado con:

`kubectl get rs,pods`

Observamos que queríamos crear 2 replicas del Pod, y efectivamente se han creado.

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

`kubectl describe rs replicaset-nginx`

Tolerancia a fallos

Y ahora comenzamos con las funcionalidades llamativas de Kubernetes. ¿Qué pasaría si borro uno de los Pods que se han creado? Inmediatamente se creará uno nuevo para que siempre estén ejecutándose los Pods deseados, en este caso 2:

kubectrl delete pod <nombre_del_pod>

kubectrl get pods

Escalabilidad

Para escalar el número de pods:

kubectrl scale rs replicaset-nginx --replicas=5

kubectrl get pods

Otra forma de hacerlo sería cambiando el parámetro replicas de fichero yaml, y volviendo a ejecutar:

kubectrl apply -f nginx-rs.yaml

La escalabilidad puede ser para aumentar el número de Pods o para reducirla:

kubectrl scale rs replicaset-nginx --replicas=1

Eliminando el ReplicaSet

Por último, si borramos un ReplicaSet se borrarán todos los Pods asociados:

kubectrl delete rs replicaset-nginx

Otra forma de borrar el recurso, es utilizar el fichero yaml:

kubectrl delete -f nginx-rs.yaml