Gestionando los Services

Service de tipo NodePort

Para aprender cómo gestionamos los Services, vamos a trabajar con el Deployment de nginx (nginx-deployment.yaml) y el Service NodePort (nginx-srv.yaml) para acceder a los Pods de este despliegue desde el exterior.

Creamos el Deployment

El primer paso sería crear el Deployment de nginx:

kubectl apply -f nginx-deployment.yaml

Creamos el Service

A continuación vamos a crear el Service de tipo NodePort que nos permitirá acceder al servidor nginx.

kubectl apply -f nginx-srv.yaml

Para ver los Services que tenemos creado:

kubectl get services

Recuerda que si usamos kubectl get all también se mostrarán los Services.

Antes de acceder a la aplicación podemos ver la información más detallada del Service que acabamos de crear:

kubectl describe service/nginx

Name: nginx

...

Selector: app=nginx

Type: NodePort

...

IP: 10.110.81.74

Port: service-http 80/TCP

TargetPort: http/TCP

NodePort: service-http 32717/TCP

Endpoints: 172.17.0.3:80,172.17.0.4:80

...

Podemos ver la etiqueta de los Pods a los que accede (Selector). El tipo de Service (Type). La IP virtual que ha tomado (CLUSTER-IP) y que es accesible desde el cluster (IP). El puerto por el que ofrece el Service (Port). El puerto de los Pods a los que redirige el tráfico (TargetPort). Al ser un service de tipo NodePort nos da información del puerto que se asignado para acceder a la aplicación (NodePort). Y por último, podemos ver las IPs de los Pods que ha seleccionado y sobre los que balanceará la carga (Endpoints).

Accediendo a la aplicación

Vemos el Service que hemos creado:

kubectl get services

...

nginx NodePort 10.110.81.74 <none> 80:32717/TCP 32s

Observamos que se ha asignado el puerto 32717 para el acceso, por lo tanto si desde un navegador accedemos a la IP del nodo master y a este puerto podremos ver la aplicación.

¿Cómo sé la dirección ip del nodo master del cluster minikube? Podemos ejecutar:

minikube ip

192.168.39.222

Y ya podemos acceder desde un navegador web:

Service ClusterIP

En esta ocasión vamos a desplegar una base de datos MariaDB. En este caso no vamos a necesitar acceder a la base de datos desde el exterior, pero necesitamos

que los Pods de otro despliegue puedan acceder a ella. Por lo tanto vamos a crear un Service de tipo ClusterIP.

Para el despliegue de MariaDB vamos a usar el fichero mariadb-deployment.yaml. Puedes comprobar que en la definición del contenedor hemos añadido la sección env que nos permite establecer variables de entorno para configurar el contenedor (los estudiaremos en el siguiente módulo).

Para la creación del Service utilizamos el fichero mariadb-srv.yaml.

Para la creación del Deployment y el Service vamos ejecutando las siguientes instrucciones:

kubectl apply -f mariadb-deployment.yaml

kubectl apply -f mariadb-srv.yaml

Comprobamos el Service creado:

kubectl get services

mariadb ClusterIP 10.106.60.233 <none> 3306/TCP 2m22s

kubectl describe service/mariadb

Name: mariadb

...

Selector: app=mariadb

Type: ClusterIP

...

IP: 10.106.60.233

Port: service-bd 3306/TCP

TargetPort: db-port/TCP

Endpoints: 172.17.0.5:3306

...

Podemos comprobar que no se ha mapeado un puerto aleatorio para que accedamos usando la IP del nodo master. Los Pods que accedan a la IP 10.106.60.233 o al nombre mariadb y al puerto 3306 estarán accediendo al Pod (172.17.0.5:3306) del despliegue de mariadb.

Eliminando los servicios

Por ejemplo para borrar el servicio mariadb, ejecutaríamos:

kubectl delete service mariadb