

SISTEMAS DISTRIBUIDOS

Práctica V

Kubernetes y Raft

Viernes B Mañana Pareja 2



14/12/2023

GUILLERMO BAJO LABORDA, 842748@unizar.es



Universidad
Zaragoza



Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

Índice

1. Introducción.....	3
2. Cambios realizados en el código.....	3
3. Despliegue con Kubernetes.....	3
Script lanzamiento.....	3
Arquitectura de Kubernetes.....	4
4. Testing.....	4

1. Introducción

En esta práctica hemos hecho uso de Kubernetes para probar el sistema “raft” elaborado en las prácticas 3 y 4.

Hemos adaptado el código de las prácticas para dar soporte a su uso con Kubernetes, algo distinto a ssh en el lanzamiento. Utilizando Kubernetes, hemos lanzado un clúster que comunica tres instancias de nodos raft entre sí y con las que un cliente interactúa para someter operaciones.

2. Cambios realizados en el código

Hemos realizado pequeños cambios en nuestro código raft para adaptarlo al despliegue con kubernetes. Principalmente hemos cambiado los archivos *clraft.go* y *main.go*, que inician el cliente y nodos raft, respectivamente.

clraft.go:

Obtenemos la url donde tiene que escuchar el servidor como argumento. Construimos el string a partir del nombre que damos al cliente, el DNS y el puerto en el que escuchamos.

Después, llamamos a *SometerOperacion* mientras no demos con un nodo líder, y comprobamos la escritura que hemos hecho.

main.go:

Únicamente hemos cambiado la forma en la que cada nodo raft recibe su nombre en la red y cómo establece la dirección en la que escuchar.

3. Despliegue con Kubernetes

Una vez modificado el código, construimos los dos elementos principales que utilizaremos para lanzar nuestro clúster.

Script lanzamiento

Comenzamos eliminando el cluster, si ya existía previamente para evitar errores. Después, creamos un nuevo cluster utilizando kind.

A continuación, compilamos y construimos las imágenes para el cliente y servidor. Las subimos al registro local para que sean accesibles.

Finalmente, el script se encarga de gestionar los recursos de Kubernetes, eliminando instancias previas de la aplicación y desplegando nuevos recursos utilizando un archivo de configuración específico.

Arquitectura de Kubernetes

Utilizaremos como base los ficheros `.yaml` proporcionados en el material de la práctica. Desplegamos tres réplicas de "raft" con un patrón de StatefulSet, asegurando la identidad única de cada instancia (réplica) y garantizando su estado persistente. Como plantilla, utilizaremos la imagen "servidor" construida y expuesta en el script anterior. Sobre esto, proporcionamos un servicio que actúa como una abstracción para acceder a las réplicas del servicio "raft" dentro del clúster de Kubernetes.

Además, ejecutamos un pod "cliente" para interactuar con el servicio "raft". Como con "raft", utilizaremos la imagen que hemos creado en el script.

En nuestro sistema, el cliente se comunica con el servicio "service-raft" a través de la red interna de Kubernetes utilizando el nombre de servicio DNS interno. El servicio "service-raft" enruta las solicitudes a una de las réplicas del "raft" StatefulSet en el puerto 6000.

4. Testing

Para la validación del correcto funcionamiento del sistema, se han comprobado los logs de los distintos servicios utilizando `kubectl logs client` y `kubectl logs raft-x` para ver la salida de los distintos procesos.