

Laboratorio 7.- Kubernetes

Contenido:

1	PREPARAR EL ENTORNO	2
2	DESPLEGAR LA APLICACIÓN "LIBRO DE VISITAS"	2
3	MODIFICAR LA APLICACIÓN	4
4	ATRIBUCIÓN	5

Objetivo: Poner en práctica los conocimientos adquiridos sobre Kubernetes, creando una aplicación con diferentes tipos de objetos.

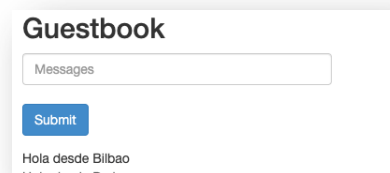
1 Preparar el entorno

Para realizar este laboratorio es necesario tener un cluster de Kubernetes funcional. Se recomienda utilizar un cluster gestionado de Google Kubernetes Engine en modo *Autopilot* e instalar las herramientas *gcloud* y *kubectl* en vuestro equipo local.

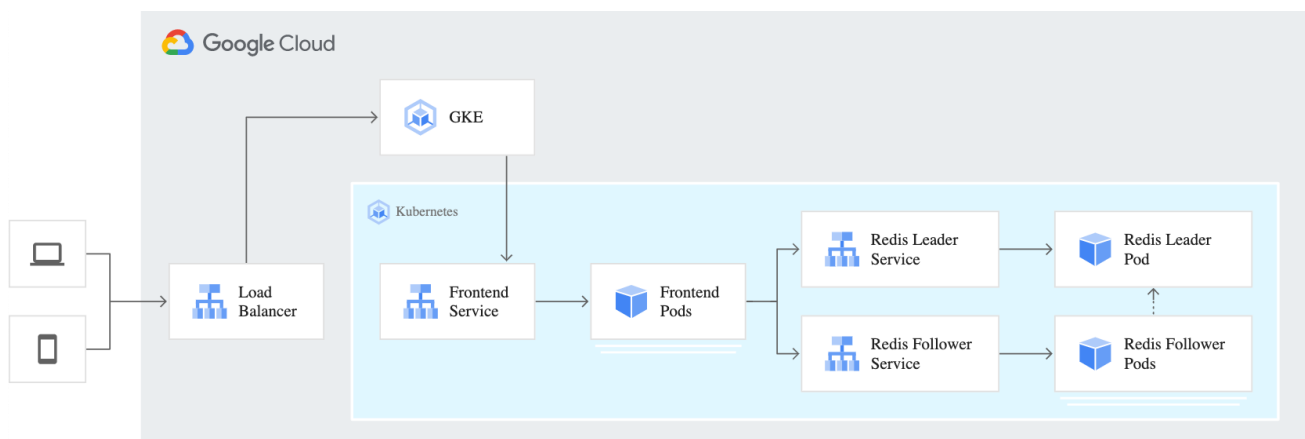
Se recomienda realizar la configuración utilizando *kubectl* desde línea de comandos y utilizar la interfaz Web de GCP sólo para revisar los cambios y monitorizar el estado del cluster.

2 Desplegar la aplicación "Libro de visitas"

En esta parte del laboratorio se va a desplegar una aplicación tipo "Libro de visitas", en la cual los usuarios podrán escribir mensajes de texto y serán visualizados en una interfaz Web. La interfaz Web será así:



La aplicación constará de una arquitectura de objetos Kubernetes como la siguiente¹:



Como se puede observar en el diagrama, hay diferentes tipos de elementos que interactúan entre sí.

- Las peticiones Web llegarán a un balanceador de carga que se encarga de enviarlas a GKE. Este balanceador estará provisto por Google Cloud y su conexión con el cluster se realizará mediante un servicio de tipo LoadBalancer.
- "Frontend Service", que será un objeto de tipo Service con la configuración de red para recibir y redirigir las peticiones Web al Pod que gestione la interfaz Web.

¹ Fuente: https://cloud.google.com/kubernetes-engine/images/guestbook_diagram.svg

Administración de Sistemas - Curso 2023 / 2024

- “Frontend Pods”, que será un objeto de tipo Deployment encargado de ejecutar Pods con la aplicación de la interfaz Web del libro de visitas. Esta aplicación está desarrollada en PHP y se comunicará con una BBDD Redis para leer y escribir los mensajes de los usuarios.
- “Redis Leader Service”, que será un objeto de tipo Service con la configuración de red para recibir las peticiones que le realicen los “Frontend Pods” y reenviarlas al Pod “Redis Leader”.
- “Redis Leader Pod”, que será un objeto Deployment que ejecute una instancia de la BBDD Redis en un Pod. En esta aplicación tendremos un despliegue distribuido de Redis para una mayor disponibilidad y tolerancia a fallos. Este Pod contiene la instancia principal de Redis que recibe las peticiones de escritura desde los “Frontend Pods”.
- “Redis Follower Service”, que será un objeto de tipo Service con la configuración de red para recibir peticiones desde los “Frontend Pods” y reenviarlas a los Pods “Redis Follower”.
- “Redis Follower Pods”, que será un objeto Deployment que controle la ejecución de 2 Pods con instancias de la BBDD Redis. Estas instancias replicarán los cambios que se realicen en el Pod “Redis Leader”.

El primer paso para comenzar el despliegue de la aplicación “Libro de visitas” es tener en marcha un cluster Kubernetes. Si se va a utilizar un cluster ya existente, es recomendable eliminar cualquier objeto que exista anteriormente para evitar problemas de configuración.

Una vez que se haya preparado el cluster, el primer paso es crear un objeto Deployment con la configuración del Pod que contendrá la instancia líder de la BBDD Redis. Su fichero de configuración se encuentra en: <https://github.com/GoogleCloudPlatform/kubernetes-engine-samples/blob/main/guestbook/redis-leader-deployment.yaml>

Tras crear el Deployment, se recomienda verificar que el Pod se crea y ejecuta correctamente (utilizando *kubectl get pods*). También es recomendable verificar que el proceso de Redis se ha iniciado sin problemas. Los logs de un Pod se pueden leer con el comando *kubectl logs*.

El siguiente paso es configurar un objeto Service que permita la conexión del Pod con la instancia líder de Redis con el resto del cluster. Su fichero de configuración es: <https://github.com/GoogleCloudPlatform/kubernetes-engine-samples/blob/main/guestbook/redis-leader-service.yaml>

Una vez que se haya configurado, se recomienda revisar su estado con el comando *kubectl get service*. Si se ha creado correctamente, habrá una línea con el nombre del Service, “redis-leader” y sus propiedades, entre ellas la dirección IP que permite la redirección de tráfico.

El siguiente paso es crear los Pods con las instancias de Redis “Follower”, que replicarán el contenido de la instancia “líder”. El fichero de configuración se encuentra en: <https://github.com/GoogleCloudPlatform/kubernetes-engine-samples/blob/main/guestbook/redis-follower-deployment.yaml>

Al igual que para el Deployment del Pod con la instancia líder de Redis, se recomienda verificar que los Pods de este Deployment se crean correctamente y que los procesos Redis se inician sin problemas dentro de cada Pod.

Una vez configurados los Pods para las instancias “Follower” de Redis, se debe crear un objeto ClusterIP que permita el acceso desde otros elementos del cluster. Utilizar el siguiente fichero de configuración: <https://github.com/GoogleCloudPlatform/kubernetes-engine-samples/blob/main/guestbook/redis-follower-service.yaml>

Al igual que para el Servicio de la instancia líder de Redis, se recomienda verificar que este objeto se ha creado correctamente mediante el comando *kubectl*.

Tras crear los 4 objetos, la configuración de la BBDD está completa. El siguiente paso es desplegar la interfaz Web de la aplicación. Su fichero de configuración es: <https://github.com/GoogleCloudPlatform/kubernetes-engine-samples/blob/main/guestbook/frontend-deployment.yaml>

Administración de Sistemas - Curso 2023 / 2024

Este fichero de configuración impone unas restricciones particulares sobre su ejecución: el nodo asignado debe dedicar al menos el 10% de su capacidad de cómputo y 100 MiB al contenedor. Se recomienda revisar el fichero para familiarizarse con las opciones y revisar la documentación asociada en:

<https://kubernetes.io/es/docs/concepts/configuration/manage-resources-containers/>

El último objeto Kubernetes a crear contiene la configuración de red para exponer la interfaz Web al exterior del cluster y hacerla accesible a los usuarios. Para ello se utiliza un objeto Service de tipo LoadBalancer, cuya configuración se encuentra en el siguiente fichero: <https://github.com/GoogleCloudPlatform/kubernetes-engine-samples/blob/main/guestbook/frontend-service.yaml>

Al igual que para los servicios anteriores, se recomienda revisar que el LoadBalancer se ha creado correctamente antes de continuar con el enunciado.

Una vez creado el LoadBalancer, el último paso de este apartado es revisar la IP pública que se le ha asignado a la aplicación. Esta información se puede obtener desde línea de comandos, visualizando las propiedades del objeto LoadBalancer con *kubect*l, en la columna "External-IP". Si todas las configuraciones son correctas, se puede utilizar esta dirección IP en un navegador para acceder a la aplicación "Libro de visitas".

3 Modificar la aplicación

En esta parte del laboratorio se proponen 3 modificaciones sobre la aplicación "Libro de visitas" para explorar otras capacidades de Kubernetes.

La primera modificación que se propone es cambiar el número de réplicas Pod en el Deployment "Frontend". Utilizar el comando *kubect*l scale para incrementar el valor de 3 a 5 y verificar con *kubect*l get que el cambio se ha realizado correctamente.

La segunda modificación que se propone es modificar los mensajes del libro de visitas, accediendo directamente a la BBDD de la aplicación. Utilizar el comando *kubect*l exec para abrir un terminal */bin/bash* dentro del contenedor líder de Redis. Una vez dentro, abrir una consola interactiva de la BBDD con el comando *redis-cli*.

Todos los mensajes del libro de visitas se encuentran dentro de una única clave de Redis. Buscar cuál es la clave que contiene los mensajes y añadir un nuevo mensaje. Abrir el libro de visitas en el navegador y verificar que el nuevo mensaje se visualiza en la interfaz Web.

La tercera modificación que se propone es reemplazar la interfaz Web del Libro de Visitas por una interfaz Web de administración de Redis, que permita explorar y modificar los contenidos de la BBDD. La interfaz de administración a utilizar será Redis Insight, una aplicación proporcionada por el equipo de Redis y que se distribuye en diferentes formas: <https://redis.com/es/redis-enterprise/redisinsight/>

Para realizar esta modificación, el primer paso es realizar los siguientes cambios en el objeto Deployment "Frontend" que controla la ejecución de los Pod con la interfaz Web:

- Número de réplicas: 1
- Imagen: redislabs/redisinsight:latest
- Puerto a exponer: 8001

Administración de Sistemas - Curso 2023 / 2024

La interfaz de Redis Insight se expone en el puerto 8001 (en lugar del 80 que utiliza la aplicación PHP del libro de visitas), por lo que es necesario realizar los siguientes cambios en el servicio LoadBalancer:

- Puerto a exponer al exterior del cluster: 80
- Puerto al que redirigir tráfico: 8001

Tras aplicar estos cambios, abrir en un navegador la IP externa del cluster. Si la configuración anterior es correcta, se mostrará la interfaz de bienvenida de Redis Insight, en la que podremos indicar que disponemos de una BBDD Redis y configurar la conexión a la misma. Para ello, elegir la primera opción del menú (*Connect to a Redis Database*) y proporcionar los siguientes datos:

- Anfitrión: redis-leader
- Puerto: 6379
- Nombre: redis-libro-visitas

Si se la configuración se ha realizado correctamente, en el apartado "Browser" de la sección "Browse" se podrá explorar la clave Redis que contiene los mensajes del libro de visitas.

Para concluir con este ejercicio, se recomienda eliminar todos los objetos (tanto Deployment como Service) creados en Cluster. Esta operación es especialmente importante si se ha utilizado GKE, los Pods en marcha que no se vayan a utilizar se facturan a la cuenta GCP.

4 Atribución

Este laboratorio está basado en el siguiente material:

- Documentación oficial Kubernetes, consultado en noviembre 2021:
<https://cloud.google.com/kubernetes-engine/docs/>
- Tutorial oficial Kubernetes, "Create a guestbook with Redis & PHP", consultado en noviembre 2021:
<https://cloud.google.com/kubernetes-engine/docs/tutorials/guestbook>