



UNIVERSIDAD DE
GUADALAJARA
Red Universitaria e Institución Benemérita de Jalisco

Centro Universitario de Ciencias Exactas e Ingenierías

Computación tolerante a fallas

Profesor: López Franco Michel

Emanuel

Sección: D06

Kubernetes

Alumno:

**Botello Martínez Nadia
Noemi**

28 de abril 2024

Contenido

Introducción	3
Desarrollo	3
Kubernetes	3
Ingress.....	4
LoadBalancer	5
Bibliografía	6

1. Introducción

Kubernetes proporciona un entorno robusto para administrar contenedores y aplicaciones, Ingress facilita la gestión del tráfico de red dentro de un clúster Kubernetes, y LoadBalancer distribuye eficientemente el tráfico entrante para mejorar el rendimiento y la disponibilidad de las aplicaciones. Juntos, estos componentes forman una infraestructura flexible y escalable para el desarrollo y despliegue de aplicaciones en la nube.

2. Desarrollo

Kubernetes:

Es una plataforma open source para la organización en contenedores que automatiza muchos de los procesos manuales involucrados en la implementación, la gestión y el ajuste de las aplicaciones que se alojan en ellos.

Kubernetes automatiza las tareas operativas de la administración de contenedores e incluye comandos integrados para implementar aplicaciones, actualizarlas, escalarlas a fin de que se ajusten a tus necesidades, supervisarlas y mucho más.

¿Para qué se utiliza?

- **Despliegue:** despliegue un número especificado de contenedores en un host especificado y manténgalos en ejecución en el estado deseado.
- **Implementaciones:** una implementación es un cambio en un despliegue. Kubernetes le permite iniciar, pausar, reanudar o revertir implementaciones.
- **Descubrimiento de servicios:** Kubernetes puede exponer automáticamente un contenedor a Internet o a otros contenedores utilizando un nombre DNS o una dirección IP.
- **Aprovisionamiento de almacenamiento:** configure Kubernetes para instalar almacenamiento persistente local o en la nube para sus contenedores según sea necesario.
- **Equilibrio de carga:** basado en el uso de la CPU o en métricas personalizadas, el equilibrio de carga de Kubernetes puede distribuir la carga de trabajo en toda la red para mantener el rendimiento y la estabilidad.

- **Autoescalado:** cuando el tráfico aumenta, el escalado automático de Kubernetes puede aumentar los nuevos clústeres según sea necesario para gestionar la carga de trabajo adicional.
- **Autocorrección para alta disponibilidad:** cuando se produce un error en un contenedor, Kubernetes puede reiniciarlo o reemplazarlo automáticamente para evitar el tiempo de inactividad. También puede retirar contenedores que no cumplan con los requisitos de comprobación de estado.

Ingress:

Ingress es un servicio que equilibra cargas de trabajo de tráfico de red en el clúster reenviando solicitudes públicas o privadas a sus apps. Puede utilizar Ingress para exponer varios servicios de app a la red privada o pública mediante un único dominio privado o público.

Componentes de Ingress:

- **Operador de Ingress**

El operador de Ingress de Red Hat OpenShift implementa reglas de direccionamiento que se aplican a todo el tráfico de entrada para las apps del clúster.

- **Controlador de Ingress**

Se crea un controlador de Ingress de Red Hat OpenShift basado en HAProxy para cada IngressController, y se crea un servicio de controlador de Ingress en cada zona donde tiene nodos de trabajador.

¿Cuándo usar Ingress?

Ingress es probablemente la solución más flexible a la hora de exponer servicios al exterior, pero también es la más complicada. Definitivamente es el método más útil si quieres exponer muchos servicios bajo la misma IP.

En cuanto a precio, pagas solo por un balanceador de carga en el caso de GCP, a diferencia que con servicios de tipo LoadBalancer, con los que tendríamos que pagar un balanceador de carga por cada servicio de este tipo que usemos en nuestro Cluster.

LoadBalancer:

Un Equilibrador de carga de aplicación actúa como la capa de aplicación, es decir, la séptima capa del modelo de interconexión de sistemas abiertos (OSI). Una vez que el equilibrador de carga ha recibido una solicitud, evalúa las reglas del oyente por orden de prioridad con el fin de determinar qué regla se debe aplicar. A continuación, selecciona un destino en el grupo de destino para la acción de la regla. Puede configurar las reglas del oyente de tal forma que las solicitudes se direccionen a diferentes grupos de destino en función del contenido del tráfico de aplicación.

¿Cómo funciona un load balancer?

El balanceo de carga se realiza mediante un algoritmo basado en el DNS (Domain Name System). El usuario accede a los sitios web a través de una URL, asociada a una dirección IP. Esta última contacta con el balanceador de carga, que transmite la petición al servidor. El balanceo dependerá del tipo de algoritmo utilizado. Los cuatro algoritmos más conocidos son Round Robin, Weighted Round Robin, Least Connections y Weighted Least Connections.

Ventajas e importancia del load balancer

La principal ventaja de un load balancer es que permite reducir los tiempos de respuesta de un sitio web tras las peticiones de los usuarios. La optimización de la carga de trabajo de los servidores reduce el riesgo de fallos provocados por una sobrecarga. Y es que, si una máquina deja de estar disponible, los usuarios serán redirigidos a otro servidor, de forma que siempre tendrán acceso a las páginas. De este modo, es posible asegurar una experiencia óptima gracias a una calidad de servicio constante: alojamiento flexible, alta disponibilidad, escalabilidad, etc. Además, OVHcloud no limita el ancho de banda ni el tráfico mensual.

3. Bibliografía

- IBM. (s.f.). Kubernetes. Recuperado el 28 de abril de 2024, de <https://www.ibm.com/mx-es/topics/kubernetes>
- OVHcloud. (s.f.). What is Load Balancing? Recuperado el 28 de abril de 2024, de <https://www.ovhcloud.com/es/public-cloud/what-load-balancing/>
- NGINX, Inc. (2023, 1 junio). What is load balancing? How load balancers work. NGINX. <https://www.nginx.com/resources/glossary/load-balancing/>