UNIVERSIDAD LATINA DE COSTA RICA

Proyecto, Marco Teorico

Estudiante:

Javier Chinchilla Lugo

Curso:

Apreciación de Arte

Profesor:

CARLOS ANDRES MENDEZ RODRIGUEZ

Diciembre, 2024

Sede San Pedro

Contenedores y Docker:

Mediante el uso de la herramienta Docker se ha logrado ir transformando el desarrollo, despliegue y gestión de aplicaciones mediante contenedores. Su adaptabilidad le permite sobresalir en varios roles: como un proyecto de código abierto y como un conjunto de herramientas simplificando la creación y administración de contenedores. Este enfoque completo ha sido clave para impulsar los contenedores como una opción rápida y eficaz en el desarrollo de aplicaciones.

El operatividad de Docker se fundamenta en el núcleo de Linux, utilizando propiedades como los nombres de espacio y los grupos de control para ejecutar aplicaciones y procesos de forma aislada. Esto permite a las organizaciones maximizar la utilización de su infraestructura sin afectar la seguridad. Gracias a su sistema de imágenes, Docker facilita el procedimiento de compartir aplicaciones y sus dependencias, automatizando su implementación en diversos entornos.

Entre los beneficios más sobresalientes de Docker se incluyen su modularidad, la gestión de versiones, la rápida puesta en marcha y la simplicidad para restaurar sistemas. Estas cualidades lo hacen una herramienta perfecta para contextos actuales que implementan metodologías como la integración y entrega continuas (CI/CD) o para aplicaciones en la nube que demandan escalabilidad y adaptabilidad. Sin embargo, también muestra algunas limitaciones, como el manejo de contenedores voluminosos o posibles falencias asociadas al demonio Docker y su relación con el núcleo del sistema operativo anfitrión.

En este proyecto, se utiliza Docker para llevar a cabo una solución de balanceo de carga para aplicaciones web. Este método no solo muestra su efectividad en la regulación del tráfico, sino que también constituye un fundamento para evaluar su influencia en el rendimiento y examinar posibles optimizaciones en situaciones más complejas.

Balanceo de carga en aplicaciones web:

Lo que se encarga el balanceo de carga es distribuir solicitudes de clientes entre varios servidores para mejorar la disponibilidad, el rendimiento y evitar los fallos. Hay dos categorías principales de balanceo de carga las cuales serian:

Hardware-based: Utiliza dispositivos dedicados, costosos pero muy eficientes.

Software-based: Implementaciones flexibles y económicas que pueden instalarse en servidores estándar, pero que consumen recursos de la máquina.

Los algoritmos comunes incluyen:

- Round Robin: Asigna solicitudes a los servidores en orden secuencial.
- Least Connections: Asigna al servidor con menos conexiones activas.
- IP Hash: Utiliza una función cifrado hash del IP del cliente para así garantizando que las solicitudes vayan al mismo servidor.

HAProxy

Es una solución de balanceo de carga de código abierto (gratis) que soporta balanceo tanto a nivel de red como a nivel de aplicación:

- NLB (Network Load Balancer): Basado en información de transporte (como TCP).
- ALB (Application Load Balancer): Analiza datos a nivel de aplicación, como encabezados HTTP.

Ventajas de HAProxy:

- Alta capacidad de procesamiento.
- Algoritmos avanzados como Weighted Round Robin, Least Connection y Source Hashing.
- Configuración ajustable para maximizar el rendimiento.

Aspectos de configuración y optimización

El artículo de HAProxy (2022) resalta la importancia de optimizar configuraciones como:

- Maxconn: Define el número máximo de conexiones simultáneas.
- Nbthread: Permite paralelizar tareas en múltiples hilos.
- Compresión y polling: Mejoran el uso de CPU y la eficiencia de red.

Impacto del balanceo de carga

Según los estudios:

1. El algoritmo Round Robin funciona bien en entornos homogéneos, pero en heterogéneos, Weighted Least Connections e IP Hash logran un mejor uso de recursos y tiempos de respuesta.
2. HAProxy puede ajustarse para escenarios específicos, como aplicaciones web de alto tráfico o sistemas con recursos limitados.

Uso de Docker para balanceo de carga:

Docker es sumamente flexible y portable lo cual causa que sea sumamente utilizado para implementar microservicios y aplicaciones en contenedores. Este genera archivos YAML, lo cual simplifica la orquestación de múltiples contenedores.

Implementación del balanceo de carga

1. Definición de servicios: Docker Compose da la oportunidad de crear múltiples contenedores en un archivo de docker-compose.yml básicamente los que crea el, cada uno con su imagen, puertos y volúmenes, así diferenciando uno del otro.
2. Configuración de balanceadores: A Compose se puede integrar un balanceador de carga como HAProxy que hablamos anteriormente, utilizándolo como servicio adicional.
3. Redes internas: Docker Compose permite crear redes internas que conectan los contenedores.

Beneficios

Facilidad de configuración: Gracias a Docker Compose al crear los entornos multi-contenedor se simplifica la configuración.

Escalabilidad: Permite escalar servicios con un solo comando (docker-compose up --scale servicio=n).

Portabilidad: Las configuraciones son reproducibles en cualquier sistema con Docker.

Integración con herramientas de CI/CD: Compatible con pipelines para automatizar despliegues.

Beneficios en entornos simples

- Simplicidad: Ideal para proyectos pequeños con pocos servicios.
- Eficiencia: Reduce la sobrecarga manual de configuración.
- Bajo costo: No requiere hardware especializado.
- Familiaridad: Herramientas como Docker son conocidas por la mayoría de los desarrolladores.

Limitaciones en entornos simples

Complejidad innecesaria: En aplicaciones monolíticas, puede agregar complejidad sin beneficios tangibles.

Dependencias adicionales: Necesita herramientas como Docker Compose y un balanceador.

Rendimiento limitado: En configuraciones básicas, el balanceador puede convertirse en un cuello de botella.

Bibliografía

DOCKER

Anderson, C. (2015). Docker [Software engineering]. *IEEE Software, 32*(3), 102–c3. https://doi.org/10.1109/MS.2015.62

Merkel, D. (2014). Docker: Lightweight Linux containers for consistent development and deployment. *Linux Journal, 2014*(239), 2. Recuperado de https://www.seltzer.com/margo/teaching/CS508.19/papers/merkel14.pdf

LOAD BALANCING

Rawls, C., & Salehi, M. A. (2022). Load Balancer Tuning: Comparative Analysis of HAProxy Load Balancing Methods. *High Performance Cloud Computing Lab, University of Louisiana*. Recuperado de https://arxiv.org/pdf/2212.14198

Ibrahim, I. M., Ameen, S. Y., Yasin, H. M., Omar, N., Kak, S. F., Rashid, Z. N., Salih, A. A., Salim, N. O. M., & Ahmed, D. M. (2021). Web Server Performance Improvement Using Dynamic Load Balancing Techniques: A Review. *Asian Journal of Research in Computer Science, 10*(1), 47–62. https://doi.org/10.9734/ajrcos/2021/v10i130234

DOCKER LOAD BALANCING

Reis, D., Piedade, B., Correia, F. F., Dias, J. P., & Aguiar, A. (2022). Developing Docker and Docker-Compose Specifications: A Developers' Survey. *IEEE Access, 10*, 2318–2329. https://doi.org/10.1109/ACCESS.2021.3137671

Piedade, B., Dias, J. P., & Correia, F. F. (2020). An Empirical Study on Visual Programming Docker Compose Configurations. *ACM/IEEE 23rd International Conference on Model Driven Engineering Languages and Systems (MODELS '20 Companion)*, October 18–23, 2020. https://doi.org/10.1145/3417990.3420194

CONTAINERS

Merkel, D. (2014). Docker: Lightweight Linux containers for consistent development and deployment. *Linux Journal, 2014*(239), 2. Recuperado de https://www.seltzer.com/margo/teaching/CS508.19/papers/merkel14.pdf

A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing

Publisher: IEEE

Cite This | PDF

Martin Randles ; David Lamb ; A. Taleb-Bendiab **All Authors**

Abstract:

The anticipated uptake of Cloud computing, built on well-established research in Web Services, networks, utility computing, distributed computing and virtualisation, will bring many advantages in cost, flexibility and availability for service users. These benefits are expected to further drive the demand