



TECNOLÓGICO
NACIONAL DE MÉXICO®



ITES®
LOS CABOS

Instituto Tecnológico de Estudios

Superiores de Los Cabos

Ing. en Sistemas Computacionales

“Contenedores, Docker, Cluster, Balanceador”

Asignatura: Administración de servidores

Docente: Salvador Chavez Barajas

Grupo: 8IS-01M

Estudiantes:

Herrera Verdugo Jorge Luis

Número de control: 19380593

Los Cabos, B.C.S., Documento elaborado el 29/05/24



Contenido

Introducción.....	3
Contenedor	3
Características de los Contenedores	3
Docker	6
Componentes Principales de Docker	6
Cluster.....	7
Características de un Clúster	7
Balanceador	8
Bibliografía.....	11

Introducción

A continuación, se realizará una investigación sobre lo que son los contenedores, los Docker, el cluster y el balanceador.

Contenedor

Los contenedores son una tecnología de virtualización ligera que permite ejecutar aplicaciones de manera aislada y consistente en diferentes entornos. A diferencia de las máquinas virtuales (VM), que incluyen un sistema operativo completo y pueden ser bastante pesadas, los contenedores comparten el mismo núcleo del sistema operativo host y son mucho más eficientes en términos de recursos.

Características de los Contenedores

- **Aislamiento:**
 - Cada contenedor opera en un entorno aislado, garantizando que las aplicaciones y sus dependencias no interfieran entre sí.
 - El aislamiento mejora la seguridad y permite ejecutar múltiples aplicaciones en el mismo host sin conflictos.
- **Portabilidad:**
 - Los contenedores encapsulan todas las dependencias de una aplicación, incluyendo librerías y configuraciones necesarias, lo que asegura que la aplicación funcione de manera consistente en cualquier entorno.

- Las imágenes de contenedor se pueden mover fácilmente entre diferentes entornos, como desarrollo, pruebas y producción, sin necesidad de modificaciones.

- **Eficiencia en el Uso de Recursos:**

- A diferencia de las máquinas virtuales, los contenedores comparten el mismo kernel del sistema operativo, lo que reduce la sobrecarga y permite ejecutar más contenedores en el mismo hardware.
- El menor consumo de recursos se traduce en un mejor rendimiento y un uso más eficiente de la infraestructura.

- **Velocidad de Despliegue:**

- Los contenedores se inician rápidamente en comparación con las máquinas virtuales porque no necesitan arrancar un sistema operativo completo.
- Esta rapidez en el arranque facilita el escalado dinámico y la respuesta ágil a la demanda de aplicaciones.

- **Inmutabilidad:**

- Las imágenes de contenedor son inmutables, lo que significa que no cambian una vez que se crean. Cualquier actualización o cambio se realiza creando una nueva imagen.
- Esto garantiza consistencia y fiabilidad en los despliegues, ya que cada despliegue se basa en la misma imagen inmutable.

- **Escalabilidad:**

- Los contenedores son ideales para arquitecturas de microservicios, donde cada microservicio se puede desplegar y escalar de manera independiente.
- Herramientas de orquestación como Kubernetes permiten gestionar y escalar automáticamente los contenedores en función de la demanda.
- **Compatibilidad con DevOps:**
 - Los contenedores facilitan las prácticas de DevOps, como la integración continua y la entrega continua (CI/CD), al permitir la creación de entornos de desarrollo, prueba y producción consistentes y reproducibles.
 - Esto acelera los ciclos de desarrollo y despliegue, mejorando la colaboración entre los equipos de desarrollo y operaciones.
- **Manejo Simplificado de Dependencias:**
 - Los contenedores incluyen todas las dependencias necesarias para ejecutar una aplicación, eliminando problemas de compatibilidad y configuración en diferentes entornos.
- **Versionado:**
 - Las imágenes de contenedor pueden versionarse, permitiendo un fácil seguimiento y despliegue de diferentes versiones de una aplicación.
- **Infraestructura como Código (IaC):**
 - Los contenedores y sus configuraciones se pueden definir como código (por ejemplo, mediante Dockerfiles), facilitando la automatización y el manejo de infraestructura de manera programática.

Docker

Docker es una plataforma de contenedores que permite a los desarrolladores crear, desplegar y ejecutar aplicaciones en contenedores de manera sencilla y eficiente. Los contenedores son entornos ligeros y portátiles que agrupan una aplicación y sus dependencias, asegurando que la aplicación se ejecute de manera consistente en diferentes entornos.

Componentes Principales de Docker

- **Docker Engine:** Es el núcleo de Docker y se compone de:
 - **Docker Daemon (dockerd):** Es el servicio que ejecuta y gestiona contenedores en el host.
 - **Docker CLI (Command Line Interface):** Es la interfaz de línea de comandos que permite a los usuarios interactuar con Docker, enviando comandos al Docker Daemon.
 - **REST API:** Permite a otros programas interactuar con el Docker Daemon.
- **Docker Images:** Son plantillas de solo lectura que contienen el sistema de archivos y la configuración necesarios para ejecutar una aplicación. Las imágenes de Docker se crean a partir de archivos llamados Dockerfiles, que describen cómo construir la imagen.
- **Docker Containers:** Son instancias en ejecución de imágenes de Docker. Un contenedor incluye todo lo necesario para ejecutar una aplicación, incluyendo el código, las librerías, las dependencias y las configuraciones.

- **Dockerfile:** Es un archivo de texto que contiene una serie de instrucciones sobre cómo construir una imagen de Docker. Cada instrucción en el Dockerfile añade una capa a la imagen.
- **Docker Hub:** Es un registro público donde los usuarios pueden almacenar y compartir imágenes de Docker. Docker Hub permite a los desarrolladores buscar y descargar imágenes preconstruidas para usar en sus proyectos.
- **Docker Compose:** Es una herramienta que permite definir y gestionar aplicaciones de contenedores multi-contenedor. Utiliza un archivo YAML para describir los servicios, redes y volúmenes necesarios para una aplicación completa.

Cluster

Un clúster es un conjunto de computadoras (también llamadas nodos) que trabajan juntas de manera coordinada para actuar como un solo sistema. Los clústeres se utilizan para mejorar el rendimiento, la disponibilidad y la escalabilidad de aplicaciones y servicios. Aquí se detallan los aspectos clave de un clúster:

Características de un Clúster

- **Alta Disponibilidad (HA):**
 - Los clústeres aseguran que las aplicaciones y servicios estén disponibles incluso si algunos nodos fallan. Si un nodo se cae, otros nodos en el clúster pueden tomar su lugar automáticamente.
- **Escalabilidad:**

- Los clústeres permiten agregar más nodos según sea necesario para manejar un mayor volumen de trabajo. Esto se llama escalado horizontal.
- **Rendimiento Mejorado:**
 - Al distribuir las tareas entre varios nodos, los clústeres pueden realizar cálculos y procesamiento más rápidamente que una sola máquina.
- **Tolerancia a Fallos:**
 - Los clústeres pueden detectar fallos en los nodos y redistribuir las tareas automáticamente, asegurando que el sistema continúe funcionando sin interrupciones.
- **Distribución de la Carga:**
 - Los clústeres distribuyen la carga de trabajo entre los nodos para optimizar el uso de recursos y evitar que un solo nodo se sobrecargue.

Balanceador

Un balanceador de carga (load balancer) es un dispositivo o software que distribuye el tráfico de red o las solicitudes de servicio entre múltiples servidores o nodos en un clúster para asegurar que ningún solo servidor se sobrecargue. Esta distribución optimiza el uso de recursos, maximiza el rendimiento, minimiza el tiempo de respuesta y asegura la alta disponibilidad de aplicaciones y servicios.

Características del Balanceador de Carga

- **Distribución de Carga:**

- El balanceador de carga distribuye el tráfico entrante entre varios servidores según ciertos algoritmos o reglas, asegurando que cada servidor maneje una cantidad equilibrada de solicitudes.
- **Alta Disponibilidad:**
 - Al distribuir las solicitudes entre múltiples servidores, el balanceador de carga ayuda a asegurar que el servicio permanezca disponible incluso si uno o más servidores fallan.
- **Redundancia:**
 - Los balanceadores de carga pueden configurarse para detectar fallos en los servidores y redirigir el tráfico a servidores en buen estado, proporcionando una capa adicional de redundancia y recuperación ante fallos.
- **Escalabilidad:**
 - Permite escalar horizontalmente agregando más servidores al clúster para manejar un mayor volumen de tráfico sin interrumpir el servicio.
- **Optimización del Rendimiento:**
 - Mejora el rendimiento del sistema al asegurar que los recursos de cada servidor se utilicen de manera eficiente, evitando la sobrecarga de un solo servidor.
- **Seguridad:**

- Puede actuar como un punto de control centralizado para aplicar políticas de seguridad, como la limitación de la tasa de solicitudes y la autenticación de usuarios.

Bibliografía

Hightower, K., Burns, B., & Beda, J. (2017). *Kubernetes: Up and running: Dive into the future of infrastructure*. O'Reilly Media.

Alleyne, D., & Woolf, B. (2007). *The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise*. Addison-Wesley Professional.

Merkel, D. (2014). *Docker: Lightweight Linux containers for consistent development and deployment*. Linux Journal, (239), 2.

Alleyne, D., & Woolf, B. (2007). *The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise*. Addison-Wesley Professional.