**Nombre:** Dennys Alexander Pucha Carrera

**Paralelo:** 4to “A” **Fecha:** 02/07/2023

**Asignatura:** Sistemas Operativos

# Docente: Ing. Hernán Leonardo Torres Carrión M.Sc.

**ENSAYO Nº 7**

# Tema

Gestión de los Procesos en los Sistemas Operativos

# Antecedentes

En este ensayo, se explorarán los contenedores perimetrales desde dos perspectivas: en primer lugar, analizaré cómo se pueden desplegar servicios en estos contenedores para consumir usuarios del Edge, y en segundo lugar, compararé los contenedores perimetrales con los contenedores en la nube para comprender sus diferencias y semejanzas.

El despliegue de servicios en contenedores perimetrales es un tema de gran interés y relevancia en el mundo actual de la computación. Estos contenedores, ubicados cerca del usuario final, ofrecen la oportunidad de brindar servicios con una baja latencia, equilibrio de carga global y una mayor capacidad de escala.

El estudio de los contenedores perimetrales y su despliegue de servicios no solo es fascinante desde el punto de vista tecnológico, sino que también tiene una gran relevancia práctica en el campo de la ingeniería en computación. Estos temas nos permiten comprender cómo las organizaciones pueden mejorar la experiencia del usuario final al reducir la latencia, optimizar el uso del ancho de banda y brindar servicios más rápidos y eficientes.

El estudio de estos temas nos prepara para enfrentar los desafíos actuales y futuros en el ámbito de la computación y nos permite estar a la vanguardia de la innovación tecnológica.

# Descripción

Para determinar el despliegue de servicios en la computación en el borde es necesario primero conocer que son los contenedores perimetrales, a continuación, se presenta una definición de los mismos:

Los contenedores perimetrales son recursos informáticos descentralizados ubicados lo más cerca posible del usuario final para reducir la latencia, ahorrar ancho de banda y mejorar la experiencia digital general. [1]

En base a esta definición se puede decir que los contenedores perimetrales son una estrategia en el ámbito de la informática descentralizada que busca optimizar la distribución de recursos informáticos al situarlos en ubicaciones cercanas a los usuarios finales. Esta técnica se utiliza con el objetivo de mejorar la velocidad de respuesta, reducir la latencia, ahorrar ancho de banda y, en última instancia, ofrecer una mejor experiencia digital a los usuarios.

Algunas ventajas y desventajas de los contenedores perimetrales pueden ser las siguientes:

|  |  |
| --- | --- |
| Ventajas | Desventajas |
| **Baja latencia:** Los contenedores  perimetrales están ubicados cerca del usuario final, lo que reduce  significativamente la latencia y mejora la velocidad de respuesta de las  aplicaciones y servicios. | **Complejidad:** La gestión de múltiples contenedores distribuidos en diversas regiones requiere una planificación y monitorización cuidadosas. |
| **Equilibrio de carga global:** Los  contenedores perimetrales utilizan una única dirección IP Anycast para distribuir el tráfico globalmente hacia el  contenedor más cercano. Esto permite un equilibrio de carga eficiente y optimizado. | **Mayor superficie de ataque:** Al tener una red más amplia con múltiples puntos de presencia (PoP), la superficie de ataque se vuelve más extensa. Esto implica que se deben implementar  políticas de seguridad adecuadas para proteger la red y los contenedores  perimetrales de posibles amenazas. |
| **Escalabilidad:** La naturaleza  descentralizada de los contenedores  perimetrales permite su implementación en múltiples ubicaciones  simultáneamente. | **Cargos de red entre PoP:** Además de los cargos regulares de entrada y salida de datos, los contenedores perimetrales pueden incurrir en cargos adicionales por el tráfico entre los diferentes puntos de presencia (PoP). [1] |

A partir de las ventajas y desventajas de los contenedores perimetrales, se puede concluir que esta estrategia presenta beneficios significativos, pero también desafíos a considerar.

Por un lado, los contenedores perimetrales ofrecen baja latencia, equilibrio de carga global, escalabilidad, madurez en las tecnologías de contenedores y reducción del consumo de ancho de banda. Estas ventajas permiten mejorar la experiencia del usuario final, ofrecer respuestas más rápidas y eficientes, adaptarse a las demandas regionales y optimizar los recursos informáticos.

Sin embargo, también existen desventajas, como la complejidad de gestionar múltiples contenedores en diversas regiones, la necesidad de implementar políticas de seguridad

adecuadas debido a una mayor superficie de ataque y los posibles cargos adicionales por el tráfico entre los puntos de presencia. Estas desventajas requieren una planificación cuidadosa, monitorización constante y consideración de los costos asociados.

Desplegar servicios en contenedores perimetrales para atender a los usuarios del edge ofrece una solución efectiva para mejorar la experiencia de los usuarios finales al reducir la latencia y acercar los recursos informáticos a ellos. Sin embargo, este enfoque también implica la necesidad de una planificación cuidadosa, configuración de infraestructura, implementación y gestión adecuada.

La implementación exitosa de servicios en contenedores perimetrales requiere comprender las necesidades del servicio, seleccionar ubicaciones estratégicas, configurar la infraestructura apropiada y garantizar la seguridad de los contenedores y los datos que manejan. [2] Además, se deben realizar pruebas y optimizaciones continuas para asegurar el rendimiento óptimo y la satisfacción del usuario final.

A continuación, se presenta los pasos o consideraciones que se necesitan para desplegar servicios en los contenedores perimetrales:

*Ilustración 1: Despliegue de servicios en contenedores*

Para el despliegue de aplicaciones en Docker usando Edge computing se debe tener en cuenta lo siguiente:

1. Configurar tu entorno de edge computing
2. Preparar tu aplicación en Docker
3. Construir la imagen Docker
4. Etiquetar y empujar la imagen Docker
5. Configurar tus dispositivos de edge computing
6. Descargar la imagen Docker en los dispositivos de edge computing
7. Ejecutar el contenedor Docker en los dispositivos de edge computing
8. Monitorear y gestionar tus contenedores en los dispositivos de edge computing

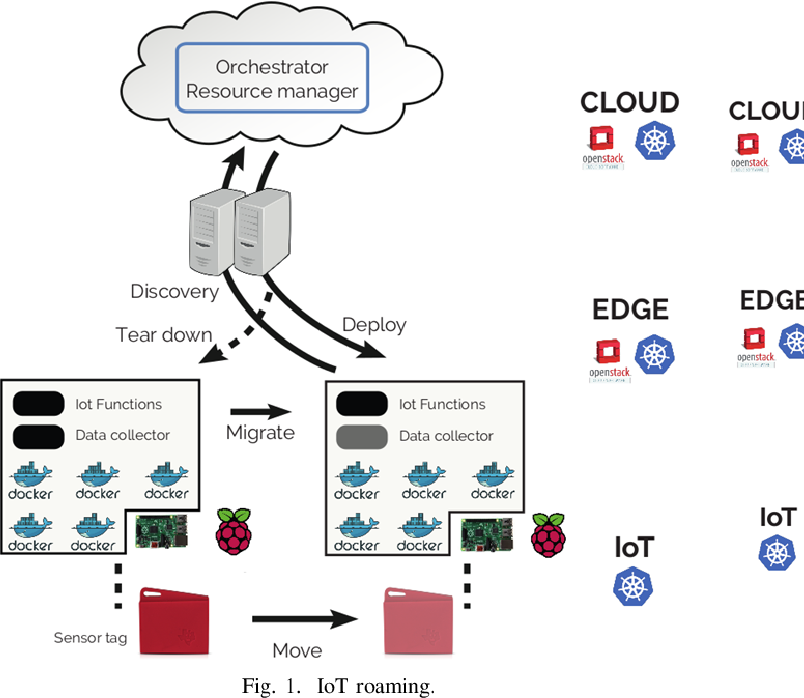


Ilustración 2: Representación del egde computing con Docker

Docker es una herramienta de contenedorización que se ha vuelto muy popular en el ámbito de DevOps. Proporciona una forma eficiente y fácil de empaquetar una aplicación y todas sus dependencias en un contenedor, lo que permite su ejecución en cualquier entorno compatible con Docker, independientemente de las diferencias en el sistema operativo o la infraestructura subyacente.

Al utilizar Docker, su aplicación se ejecuta dentro de un contenedor aislado, lo que significa que está encapsulada en un entorno virtualizado ligero y utiliza recursos mínimos del sistema. Esto tiene varias ventajas. En primer lugar, los contenedores de Docker son muy rápidos de iniciar y detener, lo que permite una implementación más rápida de la aplicación. Además, como los contenedores son portátiles, puede ejecutar su aplicación de manera consistente en diferentes entornos, lo que facilita el desarrollo, las pruebas y la implementación.

Cuando se trabaja en un entorno de producción en vivo, es común tener cientos o incluso miles de contenedores Docker en ejecución en múltiples clústeres. Administrar todos estos contenedores manualmente puede resultar complicado y propenso a errores. Aquí es donde entran en juego soluciones como Kubernetes y Docker Swarm.

Kubernetes es una plataforma de orquestación de contenedores que permite gestionar, escalar y coordinar automáticamente aplicaciones en contenedores a gran escala. Proporciona una arquitectura robusta y flexible para administrar clústeres de contenedores Docker. Con Kubernetes, puede definir cómo se deben ejecutar y escalar sus aplicaciones, y Kubernetes se encargará de orquestar los recursos necesarios para mantener el estado deseado de la aplicación.

Docker Swarm, por otro lado, es una solución de clúster nativa de Docker que permite crear y administrar clústeres de contenedores Docker. Proporciona una interfaz sencilla para definir y escalar servicios distribuidos en múltiples nodos. Aunque Docker Swarm es menos complejo que Kubernetes, aún ofrece características importantes para administrar clústeres de Docker a gran escala.

Si bien Kubernetes y Docker Swarm son soluciones poderosas para administrar grandes clústeres de contenedores Docker, es cierto que también introducen cierta complejidad adicional. Administrar estos sistemas requiere un conocimiento más profundo de las habilidades y conceptos relacionados con la orquestación de contenedores. Sin embargo, esta complejidad adicional se justifica por la capacidad de escalar rápidamente y administrar eficientemente aplicaciones distribuidas en contenedores en entornos de producción de gran escala.

A continuación, se presenta una tabla comparativa entre los contenedores perimetrales y los contenedores en la nube:

|  |  |  |
| --- | --- | --- |
| **Característica** | **Contenedores Perimetrales** | **Contenedores en la Nube** |
| Ubicación | En el borde de la red, cerca del usuario final | Centros de datos continentales o regionales lejanos |
| Ventajas principales | Baja latencia, cercanía al usuario final | Escalabilidad, recursos centralizados, flexibilidad |
| Herramientas utilizadas | Misma tecnología que contenedores en la nube | Misma tecnología que contenedores perimetrales |
| Monitoreo | Monitoreo con pruebas, métricas en tiempo real | Monitoreo con pruebas, métricas en tiempo real |
| Uso y análisis | Análisis del uso y métricas en tiempo real | Análisis del uso y métricas en tiempo real [3] |

A pesar de estas diferencias, ambos tipos de contenedores utilizan herramientas idénticas, como Docker, lo que permite a los desarrolladores aprovechar su experiencia existente en Docker sin tener que aprender nuevas tecnologías. [4] Además, tanto los contenedores perimetrales como los contenedores en la nube pueden ser administrados mediante interfaces de usuario web, herramientas de orquestación y API de administración.

# Conclusiones

* + El uso de los contenedores perimetrales es de gran relevancia debido a su ubicación cercana con el usuario final lo que permite una baja latencia y una mejor experiencia para los usuarios.
  + Los contenedores perimetrales implican la necesidad de una planificación cuidadosa, configuración de infraestructura, implementación y gestión adecuada, lo que nos dice que para una implementación exitosa de servicios en contenedores perimetrales se requiere comprender las necesidades del servicio por completo.
  + Para escoger entre los contenedores perimetrales y contenedores en la nube es muy necesario identificar los requisitos y objetivos específicos de la organización. Si la baja latencia y la proximidad al usuario final son prioritarias, los contenedores perimetrales son una opción adecuada. Por otro lado, si la escalabilidad y la centralización de recursos son más importantes, los contenedores en la nube son la opción preferida

.

1. Bibliografia
2. StackPath. “What are Edge Containers?”. StackPath Edge Academy. https://[www.stackpath.com/edge-academy/edge-containers/](http://www.stackpath.com/edge-academy/edge-containers/) (accedido el 19-06-2023).
3. IBM. “Simplificando el almacenamiento para containers, edge y nube híbrida”. IBM Systems Blog. https://[www.ibm.com/blogs/systems/mx-es/2021/04/simplificando-el-](http://www.ibm.com/blogs/systems/mx-es/2021/04/simplificando-el-) almacenamiento-para-containers-edge-y-nube-hibrida/ (accedido el 19-06-2023).
4. Supanet. “Here’s All You Need to Know About Edge Containers”. Supanet. https://[www.supanet.com/here-s-all-you-need-to-know-about-edge-containers-](http://www.supanet.com/here-s-all-you-need-to-know-about-edge-containers-) a22146.html (accedido el 19-06-2023).
5. Microsoft Azure. “Running Cognitive Services on IoT Edge”. Microsoft Azure Blog. https://azure.microsoft.com/en-us/blog/running-cognitive-services-on-iot-edge/ (fecha de acceso).