



5.- Orquestación de contenedores para DevOps

La orquestación de contenedores es una práctica esencial en DevOps que permite administrar eficientemente aplicaciones y asegurar su despliegue, escalado y funcionamiento adecuado.

- ¿Qué es la orquestación de contenedores?:** La orquestación de contenedores se refiere a la administración automatizada de contenedores en un entorno de desarrollo, pruebas y/o producción. Implica la creación, el despliegue, la escalabilidad y la monitorización de contenedores en una aplicación, todo ello de forma coordinada.
- Herramientas de orquestación de contenedores más relevantes son:**
 - Docker Swarm:** Es una herramienta de orquestación nativa de Docker que permite administrar un clúster de contenedores. Proporciona funcionalidades para el escalado automático, la distribución de carga y la recuperación ante fallos.
 - Kubernetes:** Es una plataforma de código abierto ampliamente adoptada para la orquestación de contenedores. Permite administrar y coordinar automáticamente la implementación y el escalamiento de contenedores en clústeres.
 - Apache Mesos:** Es una plataforma de orquestación de contenedores que proporciona capacidades avanzadas de administración y programación de recursos. Permite ejecutar varios frameworks de aplicaciones, como Marathon y Chronos, para gestionar contenedores.
- Ventajas de la orquestación de contenedores:**
 - Escalabilidad:** Las herramientas de orquestación permiten escalar automáticamente los contenedores en función de la demanda, lo que asegura que la aplicación pueda manejar un aumento en el tráfico sin problemas.
 - Alta disponibilidad:** La orquestación de contenedores garantiza que la aplicación se mantenga en funcionamiento incluso en caso de fallos en los contenedores individuales.
 - Gestión simplificada:** Las herramientas de orquestación proporcionan una interfaz unificada para administrar y monitorear todos los contenedores en un clúster, lo que simplifica las tareas de gestión y solución de problemas.



5.1.- ¿Qué es DevOps?

En el mundo de la computación en la nube, **DevOps es una metodología que combina** las prácticas de **desarrollo** de software (**Dev**) y las **operaciones de TI (Ops)** para lograr una colaboración estrecha y una entrega continua de aplicaciones de alta calidad. El objetivo principal de DevOps es agilizar el ciclo de vida del desarrollo de software, mejorando la eficiencia, la calidad y la confiabilidad de los productos y servicios ofrecidos.

Algunas de las principales alternativas en el mercado para implementar DevOps son:

- **AWS CodeStar:** Es un servicio de AWS que permite a los equipos de desarrollo comenzar rápidamente con proyectos de software, proporcionando plantillas predefinidas, herramientas de colaboración y una integración sencilla con otras herramientas de AWS.
- **Azure DevOps:** Es una plataforma integral de Microsoft que proporciona un conjunto completo de herramientas para la colaboración, la entrega continua, la administración de proyectos y el monitoreo en el desarrollo de software. Incluye características como control de versiones, seguimiento de problemas, compilación y despliegue automatizado.
- **Google Cloud Build:** Es una herramienta de Google Cloud que permite la creación, prueba y despliegue de aplicaciones de manera rápida y confiable. Proporciona integración con otras herramientas de Google Cloud, como Kubernetes Engine, para orquestar el despliegue de contenedores.

DevOps y las metodologías ágiles de desarrollo, como Scrum y Kanban, están estrechamente relacionados y se complementan entre sí para acelerar la entrega de software y mejorar la calidad del producto final.

Las metodologías ágiles se centran en la colaboración, la flexibilidad y la entrega incremental de funcionalidades. Estas metodologías promueven equipos multidisciplinarios que trabajan en ciclos cortos de desarrollo, llamados sprints, donde se priorizan y entregan constantemente las características más valiosas para el cliente. El objetivo principal es obtener retroalimentación temprana y frecuente para iterar y mejorar continuamente el producto.

DevOps se alinea perfectamente con las metodologías ágiles al abordar la colaboración y la entrega continua de software. DevOps enfatiza la integración y la colaboración entre los equipos de desarrollo y operaciones, eliminando las barreras entre ellos. En lugar de tener departamentos separados donde los equipos de desarrollo crean software y luego los equipos de operaciones lo despliegan y mantienen, **DevOps fomenta un enfoque de equipo único y colaborativo.**

El despliegue continuo es una práctica clave en DevOps y es una extensión natural de las metodologías ágiles. Consiste en automatizar y agilizar el proceso de entrega de software, lo que permite lanzamientos frecuentes y confiables de nuevas funcionalidades y correcciones de errores. En lugar de realizar grandes despliegues



infrecuentes, el despliegue continuo permite realizar cambios pequeños y frecuentes, lo que minimiza los riesgos y los tiempos de inactividad.

En resumen, DevOps es una metodología que promueve la colaboración y la entrega continua de software. **DevOps fomenta un enfoque de equipo único y colaborativo entre los desarrolladores y los técnicos de sistemas.** Las alternativas en el mercado, como AWS CodeStar, Azure DevOps y Google Cloud Build, brindan herramientas y servicios para implementar eficazmente DevOps en entornos de computación en la nube.

5.2.- Para qué necesitamos la orquestación de contenedores

La orquestación de contenedores es una tecnología que nos permite gestionar y coordinar eficientemente un conjunto de contenedores. Proporciona un conjunto de herramientas y servicios que facilitan la administración, escalabilidad y alta disponibilidad de las aplicaciones basadas en contenedores.

Algunos de los beneficios y usos principales de la orquestación de contenedores son:

- **Gestión y coordinación:** La orquestación permite gestionar de manera centralizada y coordinada todos los contenedores que forman parte de una aplicación. Esto incluye tareas como el despliegue, la configuración, la supervisión y la gestión de la comunicación entre los contenedores.
- **Escalabilidad automática:** Con la orquestación, podemos escalar automáticamente nuestros contenedores según la demanda de la aplicación. Esto significa que podemos aumentar o disminuir el número de contenedores en función del tráfico o la carga de trabajo, asegurando así un rendimiento óptimo.
- **Alta disponibilidad:** La orquestación garantiza la alta disponibilidad de nuestras aplicaciones al distribuir los contenedores en varios nodos o servidores. Si un nodo falla, los contenedores son automáticamente reubicados en otros nodos disponibles, evitando así la interrupción del servicio.
- **Balancedo de carga:** La orquestación de contenedores también ofrece funcionalidades de balanceo de carga. Distribuye el tráfico entrante de manera equilibrada entre los contenedores, asegurando una distribución uniforme de la carga y evitando que un contenedor se sobrecargue.



- **Gestión de la configuración:** La orquestación facilita la gestión de la configuración de los contenedores. Permite definir y mantener consistentes las variables de entorno, las configuraciones de red y otros parámetros relevantes para el correcto funcionamiento de la aplicación.
- **Despliegue declarativo:** La orquestación se basa en un enfoque declarativo, lo que significa que especificamos el estado deseado de nuestra aplicación en lugar de detallar los pasos específicos para llegar a ese estado. Esto simplifica el despliegue y la gestión de la aplicación, ya que la orquestación se encarga de implementar el estado declarado de manera automática.

En resumen, la orquestación de contenedores es esencial para gestionar y coordinar eficientemente los contenedores en un entorno de computación en la nube. Proporciona características como la gestión y coordinación centralizada, la escalabilidad automática, la alta disponibilidad, el balanceo de carga, la gestión de la configuración y el despliegue declarativo. Estas características son fundamentales para asegurar un despliegue eficiente, flexible y confiable de las aplicaciones basadas en contenedores.

5.3.- Control de versiones para el trabajo en equipo y el despliegue continuo de aplicaciones

El control de versiones es una parte fundamental del trabajo en equipo y el desarrollo de aplicaciones. **Permite gestionar los cambios realizados en el código fuente de manera colaborativa, mantener un historial de modificaciones** y facilitar la integración de nuevas funcionalidades. En el contexto de la computación en la nube, el control de versiones cobra aún más importancia, ya que es necesario coordinar el desarrollo y despliegue de contenedores en un entorno distribuido.

Una herramienta popular para el control de versiones en la nube es GitLab. Permite almacenar y gestionar el código fuente de manera centralizada, brindando un entorno colaborativo para el trabajo en equipo.

Algunas de las características clave de GitLab son:

- **Repositorios Git:** GitLab utiliza el sistema de control de versiones distribuido Git, lo que permite un seguimiento preciso de los cambios realizados en el código y la colaboración



entre desarrolladores.

- **Ramas y fusiones:** GitLab permite crear ramas para desarrollar nuevas funcionalidades o solucionar problemas sin afectar la rama principal. Posteriormente, se pueden fusionar las ramas para incorporar los cambios al código base.
- **Integración continua y despliegue continuo:** GitLab ofrece herramientas para implementar procesos de integración continua y despliegue continuo (**CI/CD**), lo que permite automatizar las pruebas y el despliegue de los contenedores a medida que se van realizando cambios en el código.

Por ejemplo, supongamos que tenemos un equipo de desarrollo que trabaja en una aplicación web basada en contenedores. Utilizando GitLab, cada desarrollador puede crear su propia rama para trabajar en nuevas características o solucionar errores. Una vez que los cambios se consideren listos, se fusionan en la rama principal (o también llamada master).

GitLab también puede integrarse con herramientas de automatización, como Docker, para construir y desplegar automáticamente los contenedores actualizados en un entorno de producción. Esto permite un despliegue continuo de la aplicación, asegurando que las últimas versiones del código estén disponibles para los usuarios finales de manera eficiente y confiable.

Revisión #1

Creado 2 mayo 2024 12:08:40 por Daniel Cano Verdú

Actualizado 2 mayo 2024 12:15:53 por Daniel Cano Verdú