

Investigación sobre CI/CD, Vcpkg, CMake y GitHub Actions

CI/CD (Integración Continua / Entrega Continua)

La Integración Continua (CI) y la Entrega Continua (CD) son prácticas fundamentales en el desarrollo de software que automatizan el proceso de desarrollo, prueba y entrega de software.

- **CI (Integración Continua):** Se refiere a la práctica de integrar cambios de código en un repositorio compartido varias veces al día. Cada integración se verifica mediante la ejecución de pruebas automatizadas, lo que permite detectar y corregir errores rápidamente.
- **CD (Entrega Continua):** Amplía la CI al automatizar la entrega de software a un entorno de producción o preproducción después de pasar las pruebas de integración. Esto asegura que el software esté siempre en un estado potencialmente desplegable.

Historia y Origen

La Integración Continua (CI) y la Entrega Continua (CD) son prácticas que surgieron en respuesta a la necesidad de mejorar la eficiencia y la calidad del desarrollo de software. Aunque no hay una fecha específica de su creación, se pueden rastrear sus orígenes en la década de 1990 con los primeros enfoques de desarrollo ágil y las metodologías de desarrollo de software.

- **Integración Continua (CI):** Se popularizó a principios de la década de 2000 con la publicación del libro "Continuous Integration: Improving Software Quality and Reducing Risk" de Paul Duvall, Steve Matyas y Andrew Glover en 2007.
- **Entrega Continua (CD):** Surgió como una extensión natural de la CI, enfocándose en automatizar el proceso de entrega de software a los entornos de producción o preproducción de forma continua y confiable.

Componentes de CI/CD

El proceso de CI/CD se compone de varios elementos que trabajan juntos para automatizar el desarrollo, prueba y entrega de software. Algunos de estos componentes incluyen:

- **Integración Continua (CI):** Se utiliza para integrar y verificar cambios de código en un repositorio compartido varias veces al día mediante la ejecución de pruebas automatizadas.
- **Entrega Continua (CD):** Amplía la CI al automatizar la entrega de software a un entorno de producción o preproducción después de pasar las pruebas de integración.
- **Build (Construcción):** Proceso de compilación y empaquetado del software a partir del código fuente.
- **Test (Pruebas):** Ejecución de pruebas automatizadas para verificar la funcionalidad y la integridad del software.

- **Release (Publicación):** Preparación y empaquetado del software para su distribución a los usuarios finales.
- **Deploy (Despliegue):** Instalación y configuración del software en un entorno de producción o preproducción para su uso.

Ejemplos de Utilización

Algunos ejemplos de cómo se utiliza CI/CD en la práctica incluyen:

- **Desarrollo Web:** Automatización del proceso de construcción, prueba y despliegue de aplicaciones web para garantizar su funcionalidad y disponibilidad continua.
- **Desarrollo de Aplicaciones Móviles:** Integración continua de cambios de código, compilación de aplicaciones y ejecución de pruebas automatizadas en diferentes dispositivos móviles.
- **Desarrollo de Software Empotrado:** Automatización del proceso de compilación y prueba de firmware para dispositivos embebidos, como sistemas de control y dispositivos IoT.

Vcpkg

Vcpkg es un gestor de paquetes de código abierto desarrollado por Microsoft que simplifica la instalación y gestión de bibliotecas y dependencias externas en proyectos de C++.

- Permite a los desarrolladores de C++ utilizar fácilmente bibliotecas de terceros en sus proyectos sin tener que lidiar con la complejidad de la compilación y la configuración manual de las dependencias.

CMake

CMake es una herramienta de construcción multiplataforma que simplifica el proceso de configuración y compilación de proyectos de software.

- Proporciona un sistema de construcción declarativo basado en archivos de configuración (CMakeLists.txt) que permite a los desarrolladores definir las dependencias, opciones de compilación y otros aspectos de la compilación de forma sencilla y portátil.

GitHub Actions

GitHub Actions es un servicio de automatización integral que permite a los equipos de desarrollo automatizar, personalizar y ejecutar flujos de trabajo directamente desde sus repositorios de GitHub.

- Utiliza archivos de configuración YAML para definir flujos de trabajo, lo que facilita su escritura y mantenimiento.
- Ofrece integración nativa con GitHub, lo que permite ejecutar flujos de trabajo directamente desde el repositorio de código.
- Proporciona un Marketplace con una amplia variedad de acciones predefinidas y herramientas listas para usar, lo que facilita la configuración y personalización de flujos de trabajo de CI/CD.

Beneficios de Utilizar YAML Frente a XML

Algunos de los beneficios de utilizar YAML en lugar de XML en la definición de flujos de trabajo de CI/CD incluyen:

- **Legibilidad Mejorada:** YAML tiene una sintaxis más simple y legible en comparación con XML, lo que facilita la comprensión y edición de los archivos de configuración.
- **Menos Verbosidad:** YAML es menos verboso que XML, lo que significa que requiere menos caracteres para representar la misma información, lo que hace que los archivos de configuración sean más concisos y fáciles de mantener.
- **Facilidad de Uso:** YAML permite una estructura de datos más intuitiva y flexible, lo que facilita la definición de flujos de trabajo complejos con una sintaxis más natural.

Conclusión

La integración de prácticas de CI/CD con herramientas como Vcpkg, CMake y GitHub Actions proporciona una base sólida para mejorar la eficiencia, calidad y velocidad del desarrollo de software. Además, el uso de YAML en la definición de flujos de trabajo ofrece beneficios adicionales en términos de legibilidad, concisión y facilidad de uso en comparación con XML.

Webgrafía

- GitHub Actions documentation. Recuperado de <https://docs.github.com/es/actions>
- GitHub Marketplace: Actions. Recuperado de <https://github.com/marketplace?type=actions>
- CI/CD - Wikipedia. Recuperado de <https://en.wikipedia.org/wiki/CI/CD>
- "CI/CD with GitHub Actions" por Minhazul Haq. [Presentación en SlideShare](#)
- "Automating CI/CD with GitHub Actions" [Vídeo en YouTube](#)