



¿CÓMO IMPLEMENTAR LA ENTREGA CONTINUA?

INTRODUCCIÓN

El objetivo de la entrega continua (CD) es asegurar que el software esté siempre en un estado de desplegarse de forma segura en producción. Este proceso está automatizado y es una extensión de la integración continua.

El propósito es mantener el entorno de producción actualizado integrando el código más reciente disponible en el control de versiones



IMPLEMENTACIÓN

Seguir un enfoque iterativo, centrado en la mejora continua.

1

AUTOMATIZACIÓN DE COMPILACIÓN (CI)

Configura la CI para automatizar la compilación y ejecución del código tras cada cambio.

2

IMPLEMENTAR PRUEBAS AUTOMATIZADAS

Incorpora pruebas unitarias, de integración y de regresión para garantizar la calidad del software.

3

CONFIGURAR EL DESPLIEGUE AUTOMATIZADO

Automatiza el despliegue para realizar implementaciones rápidas y seguras en producción.

4

ESTABLECER MONITOREO Y RETROALIMENTACIÓN

Implementa monitoreo continuo y utiliza la retroalimentación para optimizar futuros ciclos de entrega.

PRÁCTICAS

AUTOMATIZACIÓN DEL PIPELINE

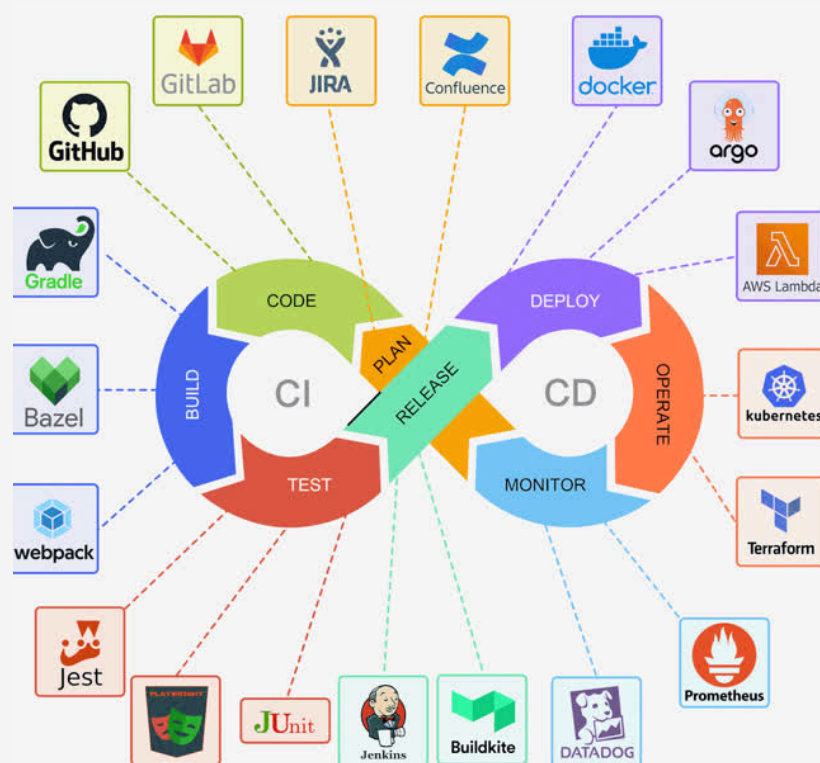
La implementación de un pipeline que permita construir, probar y desplegar código de manera automatizada.

FEEDBACK CONTINUO

Incorporar mecanismos de retroalimentación continua para detectar problemas rápidamente y ajustar el código de forma iterativa.

VERSIONAMIENTO

Adopción de metodologías que permitan un control que asegure que se trabaje de manera estructurada y ordenada.



BENEFICIOS

MENOS RIESGOS

Despliegues en pequeñas partes y frecuentes, reducen los riesgos asociados con lanzamientos complejos.

MEJOR CALIDAD

Las pruebas automáticas aseguran la validación continua del código.

COLABORACIÓN EFICIENTE

Promueve el trabajo conjunto y la sincronización entre equipos.

DESPLIEGUE RÁPIDO

Facilita la entrega de nuevas funciones de manera más frecuente

RETROALIMENTACIÓN INMEDIATA

Permite ajustes rápidos según las necesidades de los usuarios.



CONCLUSIÓN

Implementar prácticas de entrega continua en una organización permiten ciclos de desarrollo más rápidos y confiables, mejora la calidad del software y reduce el tiempo de respuesta a errores, ayudando a las organizaciones a mantener una ventaja competitiva en un mercado dinámico.



REFERENCIAS

- Shahin, M., Ali Babar, M., & Zhu, L. (2017). Continuous integration, delivery, and deployment: A systematic review on approaches, tools, challenges, and practices. IEEE Access, 5, 3909–3943.
- GeeksforGeeks. CI/CD. <https://www.geeksforgeeks.org/ci-cd-continuous-integration-and-continuous-delivery/>

