

Continuous Delivery Continuous Deployment

¡Vamos a revisar y profundizar lo aprendido hasta acá de CI/CD! ¿Qué problemas viene a solucionar? ¿Cuáles son sus principales ventajas? ¿Por qué es tan importante para la cultura DevOps?

La CI/CD incorpora la automatización continua y el control permanente en todo el ciclo de vida de las aplicaciones, desde las etapas de integración y prueba hasta las de distribución e implementación. Este conjunto de prácticas es una solución para los problemas que puede generar la integración del código nuevo a los equipos de desarrollo y de operaciones.

Vamos ahora a detenernos en un momento en particular de este proceso. El Despliegue Continuo (CD) puede referirse a dos prácticas similares pero con impactos diversos:

- Continuous Delivery.
- Continuous Deployment.

¡Examinemos de cerca ambos procesos!

¡Una diferencia clave!

Los procesos de continuous delivery (o entrega continua) y continuous deployment (despliegue continuo) siguen las mismas etapas, ¡pero con una pequeña gran diferencia!

La **entrega** continua se focaliza en la automatización de los pasos para que nuestro software esté disponible para ser aplicado en los ambientes productivos en cualquier momento, ¡pero no hace la implementación automática en producción!

El despliegue va un paso más allá. En esta práctica, todo es automático en los ambientes, ¡también en producción! Buscamos que no exista intervención humana en ninguno de los procesos de trabajo.

Esta es la diferencia más importante con respecto a la entrega continua.

Para lograrlo, el pipeline de producción tiene una serie de pasos que deben ejecutarse en orden y forma satisfactoria ¡y de manera automática! Si alguno de esos pasos no finalizan de forma esperada, el proceso de despliegue no se llevará a cabo.

¡Veámoslo de manera gráfica!



¡Examinamos ambos procesos!

Atención:
Si bien ambos términos, “continuous deployment” y “continuous delivery”, manejan todo su ciclo de procesos de forma similar (test y release), la etapa final es distinta. ¡En “continuous deployment” el paso al ambiente productivo es automático!

Antes y después:
Dependiendo de las prácticas adoptadas por la organización, el despliegue puede percibirse como una etapa posterior a la entrega porque estamos impactando nuestro ambiente productivo de forma directa luego de hacer el delivery de nuestro producto.

Beneficios:

- Tiempos de entrega más rápidos.
- Mayor visibilidad al liberarse código con más frecuencia.
- Mejor calidad de software: hacemos cambios pequeños y podemos identificar fácilmente los problemas mediante múltiples testeos que podemos configurar antes que el producto sea lanzado en nuestro ambiente productivo.

Automatización:
Es el punto vital de estas prácticas porque sin ella no existirían desencadenantes o “triggers” que impacten sobre los procesos siguientes. ¡La automatización también nos facilita los procesos lógicos de toma de decisiones o diseñar bucles para tareas repetitivas!

Madurez y confianza:
Para que estos procesos corran de manera satisfactoria en nuestra organización necesitamos de un alto grado de madurez y confianza técnica de nuestro equipo de trabajo.

¡A jugar! ¿Entrega o despliegue? ¡Arrastrá cada ficha al proceso que mejor la caracteriza!

¿Entrega o despliegue?

Arrastrar cada ficha a la categoría que corresponda. Este juego viene con una pequeña trampa: ¡alguna de las fichas puede aplicar para ambos procesos! ¿Te animás?

Entrega continua

- Test unitario
- Pasaje manual

Compartido

- Nexus
- Versionado
- Artefactos
- Test de integración

Despliegue continuo

- Automatización End-to-End
- Etapas de release
- Madurez en los procesos
- Test de carga

Verificación

¡Felicitaciones!