

¿Cómo hacer un pipeline de CI / CD exitoso?

Tenemos nuestro pipeline configurado. ¡Genial! Pero ahora... ¿Podemos asegurarnos que sea un éxito? ¿Cómo? ¿Qué podemos hacer para reducir al máximo cualquier inconveniente o retroceso que pueda aparecer?

Construir un pipeline completo de un extremo a otro no es tarea sencilla, ya que involucra multitud de procesos y herramientas. Por eso es bueno que podamos generar y tener siempre presente estas **buenas prácticas** para el trabajo en DevOps.

Checklist de buenas prácticas



Fallar rápido. Provocar que el pipeline falle ni bien se detecta un problema en la etapa, generando un feedback a los desarrolladores para que resuelvan el problema sin perder tiempo.



Tener etapas bien definidas. Cada etapa del pipeline debe representar de manera clara un proceso de CI/CD, para que sea más fácil medir y analizar su funcionamiento —y mejorar o detectar problemas—.



Hacer pruebas fuertes. Dedicar tiempo para que las pruebas automatizadas sean exhaustivas, ya que serán responsables de la calidad de nuestra entrega. ¡No queremos que un producto fallido llegue a producción!



Activar el Rollback. En caso de que llegue un producto fallido a producción, mediante el rollback podemos volver atrás los cambios y desplegar una versión anterior y sin fallas.



Bajar la dependencia. Eliminar del pipeline cualquier factor externo que no podamos controlar. Por ejemplo, si al hacer un test tenemos que hacer una consulta a una base de datos, es conveniente generar una base de datos efímera y no depender de una ajena. La mejor práctica, aunque difícil de conseguir, es levantar todo el entorno de prueba en cada corrida del pipeline y luego destruirlo.



Ejecutar fácilmente. Un pipeline no puede ser complejo de lanzar. El desarrollador tiene que poder ejecutarlo sin esfuerzo. Por ejemplo, es posible automatizar el lanzamiento al aprobar un nuevo código en un gestor de versiones.



Contar con credenciales seguras. Gestionar usuarios y contraseñas en depósitos que permitan resguardar estos datos sensibles. Ofuscar logs en donde puedan aparecer.

