Engineer

DevOps

∞ prácticas que unen desarrollo de software y operaciones

Qué es

DevOps es una forma de organizar el trabajo de software en la que quienes escriben un sistema también responden por cómo se compila, se lanza y se mantiene en funcionamiento, y los pasos entre escribir el código y ejecutarlo se automatizan en lugar de pasarse de un equipo a otro. El nombre mezcla development y operations, desarrollo y operaciones, y se acuñó para unir a esos dos grupos.

El NIST, Instituto Nacional de Estándares y Tecnología de Estados Unidos, lo define como prácticas para automatizar los procesos entre los equipos de desarrollo y operaciones «para que puedan compilar, probar y lanzar software más rápido y de forma más fiable». La definición es útil porque nombra la parte que importa: los procesos intermedios. Una empresa puede tener excelentes desarrolladores y excelentes operadores y aun así perder días en el punto en que unos entregan un lanzamiento a los otros.

En la práctica, DevOps se manifiesta en unos cuantos hábitos que se refuerzan entre sí:

- **Responsabilidad compartida sobre producción.** Quien cambia el código observa qué hace el cambio al ejecutarse, mediante monitorización, registros y alertas, y responde por ello.
- **Un camino automatizado hasta el lanzamiento.** La compilación, las pruebas y el despliegue se ejecutan como un flujo de procesamiento, igual cada vez, normalmente sobre CI.
- **Infraestructura escrita en archivos.** Servidores, redes y bases de datos se describen en plantillas o playbooks conservados junto al código, para poder reconstruir un entorno en lugar de recordarlo.

- **Una compilación, muchos entornos.** El artefacto que superó las pruebas en preproducción es el mismo que llega a producción; solo cambian sus ajustes.
- **Lanzamientos pequeños y frecuentes.** Cada uno lleva pocos cambios, por lo que un defecto es fácil de encontrar y de revertir.

DevOps no es un producto y solo de forma imprecisa es un nombre de puesto. Los proveedores venden herramientas bajo ese nombre y las empresas contratan «ingenieros DevOps», pero ninguna de esas cosas es la práctica en sí. Un ingeniero DevOps normalmente construye y cuida los flujos de procesamiento y la infraestructura que permiten que otros desarrolladores entreguen por su cuenta: una función dentro de la práctica, no un sustituto de ella.

Utilice el concepto en una decisión

Aplique «¿Qué es DevOps?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Cambios probados, publicados y reversibles, para que lleguen a sus clientes un martes por la tarde en vez de un viernes por la noche.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. Despliegue GIS SaaS con GitHub Actions — despliegue, datos e informes automatizados

Despliegue GIS SaaS con GitHub Actions Póngase en contacto →

Por qué importa

Cada traspaso entre equipos pierde algo. El desarrollador sabe por qué se hizo un cambio y de qué depende; el operador sabe cómo son los servidores y qué se rompió el mes pasado. Cuando un lanzamiento pasa de uno a otro como una incidencia y un paquete, ambas mitades de ese conocimiento llegan incompletas, y las carencias aparecen durante el despliegue, el momento más caro para descubrirlas.

Twelve-Factor App, una metodología sobre cómo se construyen las aplicaciones de software como servicio, identifica con precisión tres de esas carencias. Hay una separación temporal cuando el código escrito hoy llega a producción semanas después. Hay una separación de personas cuando un grupo escribe el código y otro lo despliega. Y hay una separación de herramientas cuando un portátil ejecuta una base de datos y producción otra. DevOps es, en gran medida, un esfuerzo por cerrar las tres: desplegar horas después de escribir, permitir que los autores desplieguen y mantener todos los entornos tan parecidos como sea posible.

El tamaño del lanzamiento es el segundo argumento. El libro de Google sobre ingeniería de fiabilidad de sitios lo expresa con claridad: los lanzamientos frecuentes significan menos cambios entre versiones. Un lanzamiento con tres cambios es fácil de analizar cuando falla. Uno con trescientos obliga a buscar.

El tercero es el trabajo operativo repetitivo, o toil. El mismo libro lo define como un trabajo ligado al funcionamiento de un servicio: es manual, repetitivo, automatizable, carece de valor duradero y crece al ritmo del servicio. Un equipo que reinicia un proceso a mano todos los martes, renueva certificados a partir de un recordatorio de calendario y amplía discos cuando salta una alerta está dedicando su tiempo a un trabajo que un script haría de forma más fiable. Automatizarlo es la manera de que un equipo pequeño siga el ritmo de un sistema que crece.

Las recomendaciones de seguridad han adoptado la misma estructura. La guía del NIST sobre DevSecOps para aplicaciones nativas de la nube parte de un flujo con etapas de compilación, pruebas, empaquetado, despliegue y operaciones, y cuenta la infraestructura, las políticas de ejecución y la monitorización entre los tipos de código que describe ese flujo.

Cómo se aplica en la práctica

Las páginas de este sitio muestran DevOps como un ámbito de responsabilidad más que como un cargo: la misma ingeniera diseña un sistema, lo lleva a producción y lo mantiene allí.

Para un producto cartográfico, el trabajo de la fundadora abarcó la infraestructura, el procesamiento de datos y la aplicación de mapas de cara al cliente durante dos años. Cubrió arquitectura, implementación y soporte en esas capas: diseñar, construir y mantener el entorno operativo, los flujos de procesamiento de datos y la aplicación cartográfica. Eso conecta desarrollo y operaciones en un mismo trabajo. El registro documenta el alcance y la duración de ese trabajo de ingeniería y mantenimiento; no establece mediciones de tiempo de actividad ni de tiempos de respuesta.

En un conjunto de 10 instancias PostgreSQL y MS SQL Server sobre VPS Ubuntu Linux, la parte operativa se automatizó pieza a pieza: copias de seguridad mediante `pg_dump` y tareas de SQL Server Agent, monitorización con Prometheus y Grafana, aplicación periódica de parches, cortafuegos UFW, control de acceso basado en roles y recuperación ante desastres documentada.

La página informa de un 99,9 % de tiempo de actividad en los diez servidores y de una reducción del 50 % en el mantenimiento manual gracias a la automatización: el argumento del trabajo operativo repetitivo medido en un conjunto concreto de servidores.

En una plataforma Azure con entornos de desarrollo, preproducción y producción, el despliegue se gestionó de principio a fin siguiendo una ruta fija. Ser responsable significaba responder por cada eslabón del camino: CI/CD compila las imágenes, las plantillas Bicep mantienen todas las etapas sobre infraestructuras equivalentes y los ajustes particulares de cada etapa se guardan separados de los secretos. Así, un artefacto compilado se promueve de etapa en etapa en lugar de recompilarse para cada una, y los cambios llegan a todos los entornos de forma predecible, sin un solo despliegue manual improvisado.

Dónde surgen los problemas

DevOps falla en silencio, normalmente conservando el vocabulario y perdiendo los hábitos.

Llega el título y permanece el muro. Se cambia el nombre de un equipo de operaciones o se contrata un «equipo DevOps» para situarlo entre los desarrolladores y producción. Los lanzamientos siguen pasando por un traspaso; solo ha cambiado el nombre de quien está al otro lado. La prueba es sencilla: pregunte quién puede entregar un cambio y a quién despiertan cuando falla. Si son personas distintas con información distinta, la práctica no ha llegado.

El despliegue sigue en manos de una persona que sigue una lista. El flujo compila y prueba, y después alguien inicia sesión en un servidor y ejecuta de memoria los cinco últimos pasos, de forma ligeramente distinta cada vez. Incluya también los pasos finales en el flujo, dé a los lanzamientos una sola ruta por los entornos y trate una publicación manual directa en producción como un incidente que hay que explicar.

Cada entorno recibe su propia compilación. Volver a compilar para preproducción y después para producción significa que lo probado no es lo entregado. Una dependencia que se resolvió de forma distinta en la segunda compilación aparece como un defecto que nadie puede reproducir. Compile una vez, etiquete el artefacto con el commit del que procede y promueva ese artefacto, con los ajustes proporcionados por el entorno:

```
tag="$(git rev-parse --short HEAD)"
docker build -t "registry.example.com/app:$tag" .
docker push "registry.example.com/app:$tag"
# staging and production run this same tag;
# only the settings they start it with differ
docker run --env-file staging.env "registry.example.com/app:$tag"
```

Preproducción se aleja de producción. Un ajuste cambiado a mano en un servidor, una base de datos actualizada primero en producción, una versión de biblioteca fijada en un lugar y sin fijar en otro: cada uno hace que preproducción sea algo menos útil como ensayo. Describa la infraestructura en plantillas, aplique las mismas en todas partes y ejecute en desarrollo los mismos servicios de apoyo que se ejecutan en producción.

El trabajo operativo repetitivo crece con el sistema. Las tareas manuales que llevaban diez minutos semanales para un servidor requieren horas para veinte. Sin una reserva deliberada de tiempo para automatizarlas, el equipo acaba dedicando sus días a mantener todo en marcha. Cuente el trabajo manual recurrente y limite la proporción de tiempo que puede consumir.

Lecturas adicionales

- DevOps — la definición del NIST, con las publicaciones especiales de las que procede.
- Compilación, lanzamiento y ejecución — por qué una compilación se hace una vez y se combina con la configuración para formar un lanzamiento.
- Paridad entre desarrollo y producción — las separaciones de tiempo, personas y herramientas entre desarrollo y producción y cómo cerrarlas.
- Configuración — mantener todo lo que varía entre despliegues fuera del código y en el entorno.
- Eliminar el trabajo operativo repetitivo — la definición de toil de Google y los argumentos para limitarlo.
- Ingeniería de lanzamientos — compilaciones herméticas, lanzamientos de autoservicio y políticas aplicadas por las herramientas de lanzamiento.



Glosario

Elija según sus restricciones reales

Describa la ruta de despliegue, los riesgos operativos y quiénes se encargan de las guardias. Podemos definir el alcance de un cambio con comprobaciones, una vía de recuperación y responsabilidades claras.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es DevOps?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/devops/