

DevOps

¿Qué es DevOps?

DevOps es la mezcla de tareas realizadas por los equipos de desarrollo de aplicaciones y sistemas de una empresa. El término DevOps se utiliza de varias maneras. En su sentido más amplio, DevOps es una filosofía operacional que promueve una mejor comunicación entre desarrollo y operaciones a medida que más elementos de operaciones se vuelven programables. En su interpretación más estrecha, DevOps describe la parte del equipo de tecnología de la información (TI) de una organización que crea y mantiene la infraestructura. El término también puede ser usado para describir una cultura que mira estratégicamente a toda la cadena de suministro de software, supervisando servicios compartidos y promoviendo el uso de nuevas herramientas de desarrollo y mejores prácticas.

¿Para que se usa?

El enfoque de DevOps busca fusionar el desarrollo y el despliegue de aplicaciones en un proceso más racionalizado que alinea el desarrollo, la garantía de calidad (QA) y los esfuerzos del equipo de operaciones. Este enfoque también transfiere algunas de las responsabilidades del equipo de operación al equipo de desarrollo para facilitar el desarrollo continuo, la integración continua, la entrega continua y los procesos de monitoreo continuo. La necesidad de dismantelar los silos entre el desarrollo y las operaciones ha sido acelerada por la necesidad de liberar el código más rápido y con mayor frecuencia para ayudar a la organización a responder de una manera más ágil a las necesidades cambiantes del negocio. Otros factores para romper los silos incluyen el creciente uso del cómputo de nube y los avances en infraestructuras definidas por software, microservicios, contenedores y automatización.

Ejemplos:

Por ejemplo, imaginemos que ya tenemos el código. Ahora hay que construir los ejecutables, básicamente compilar y enlazar. Ésta es una tarea que normalmente se ejecuta desde el entorno de desarrollo -típicamente Microsoft Visual Studio para entornos Windows y Eclipse para entornos Linux-, aunque han aparecido herramientas como Maven o Ant (ambas de la fundación Apache) que, con lenguaje Java, complementan estas actividades y resuelven las dependencias. La buena noticia es que la integración de Eclipse con Maven o Ant, al igual que con GIT o Subversion es muy sencilla.

La automatización de las pruebas, sean de código, de sistema, de servicio, regresivas, etc. va a permitir que los cambios que se realicen en el software sean verificados, lo que garantiza la calidad. Incluso permiten la ejecución de juegos de pruebas diarios y/o

semanales durante la noche que permiten tener un informe de resultados a primera hora de la mañana. Aquí la variedad de herramientas es casi infinita en lo que se refiere a proveedores convencionales. En el mundo open source hay productos muy interesantes como Selenium o Windmill para funcionalidad web, JMeter para pruebas de carga e incluso Taurus que funciona como framework por encima de las demás.

En la parte del despliegue también hay una legión de herramientas disponibles. ¿Queremos manejar las aplicaciones (y sistemas) como lo hemos hecho toda la vida, sea en físico o en virtual, o estamos dispuestos a transformar esa forma de hacerlo por una serie de capas de cebolla que se pueden desplegar con relativa independencia? Si optamos por lo primero, están Puppet o Chef (aunque también hay otras). Pero si nos atrevemos con la posibilidad de que las pequeñas piezas de código no solo se puedan desarrollar y probar con independencia del resto, sino que su despliegue se haga como una “pieza de lego” separada, entonces, queridos amigos, tenemos todo lo necesario para empezar a trabajar en Linux con Docker.