

Prácticas y herramientas

Ejercicio 1

*¿Podrías contar qué prácticas y herramientas DevOps usas en tu trabajo?
¿Cuáles te gustaría introducir? ¿Por qué?*

Siguiendo un poco la imagen del loop, diría:

- Build: utilizamos git, junto con github (hasta ahora self-hosted, pero se está migrando a la nube). Trabajamos mucho con docker y kubernetes, que nos permiten poder ejecutar la aplicación en local. Tenemos también una herramienta que se llama Xingbox que, en pocas palabras, nos permite ejecutar las aplicaciones (más bien los pods que queremos) en máquinas virtuales, de esta forma no es necesario tener todo el entorno montado en local.
- Plan: de esto se encarga principalmente nuestra PO y luego contribuimos los desarrolladores en los refinements. Hasta donde conozco, utilizamos
- Deploy: la verdad es que no tengo mucho conocimiento sobre el tema, pero sé que tienen su propio tool para los deploys llamado olympus, que nos permite ejecutar comandos a los pods de kubernetes en cuestión.
- Operate: el mismo olympus contiene comandos para gestión post-deploy, por así decirlo. Para control, logs y similares, utilizamos herramientas como: XING Cube (una herramienta desarrollada por la misma empresa que permite analizar si los workers - de los microservicios - funcionan correctamente), Grafana, Logjam, Graylog y alguna otra herramienta hay.
- Continuous integration: aquí es un poco más complejo porque cada equipo y cada tecnología utilizan lo suyo. Por ejemplo, en el repositorio donde tenemos nuestra parte de front-end, tenemos configurado el CI con Jenkins, SonarQube y algunas partes con Act (un CI tool que está en beta diseñado por la misma empresa). Los desarrolladores native de mi equipo trabajan en un monorepo y sé que utilizan herramientas completamente diferentes (por ejemplo Percy).
- Continuous feedback: tenemos nuestro propio tracking y además se utiliza Adobe Analytics, pero la intención es eliminarlo. La comunicación

a nivel de empresa se hacía vía Slack y Zoom, aunque ahora nos han hecho pasar a Office 365.

En general considero que tenemos un montón de herramientas, algunas más utilizadas que otras, y hasta ahora no me he encontrado con ningún problema que no pueda solucionar con alguna de las anteriores (o las que quizás se me escapan) herramientas.

Ejercicio 2

¿Podrías contar qué es Platform Engineering? ¿Cuáles son las diferencias entre Platform Engineering y DevOps?

Se entiende por *platform engineering* como la disciplina que se encarga de diseñar y desarrollar herramientas y workflows que permiten operar durante todo el lifecycle de una aplicación en la nube.

La principal diferencia se encuentra en que: devops se encarga principalmente de proporcionar partes (pipelines) para el CI/CD de los diferentes equipos, mientras que el platform engineering se encarga de proporcionar una herramienta/aplicación que permite a los equipos poder un ecosistema donde se encuentre todo integrado.