Pipelines + GitLab

A lo largo de la materia y en muchas de las prácticas que estuvimos haciendo usamos Jenkins como herramienta para automatizar el despliegue de software. ¡Pero no es el único recurso que tenemos disponible para hacerlo!

En la ejercitación de esta clase vamos a explorar, conocer y sumar una herramienta más de las que se usan en el mercado: ¡GitLab!

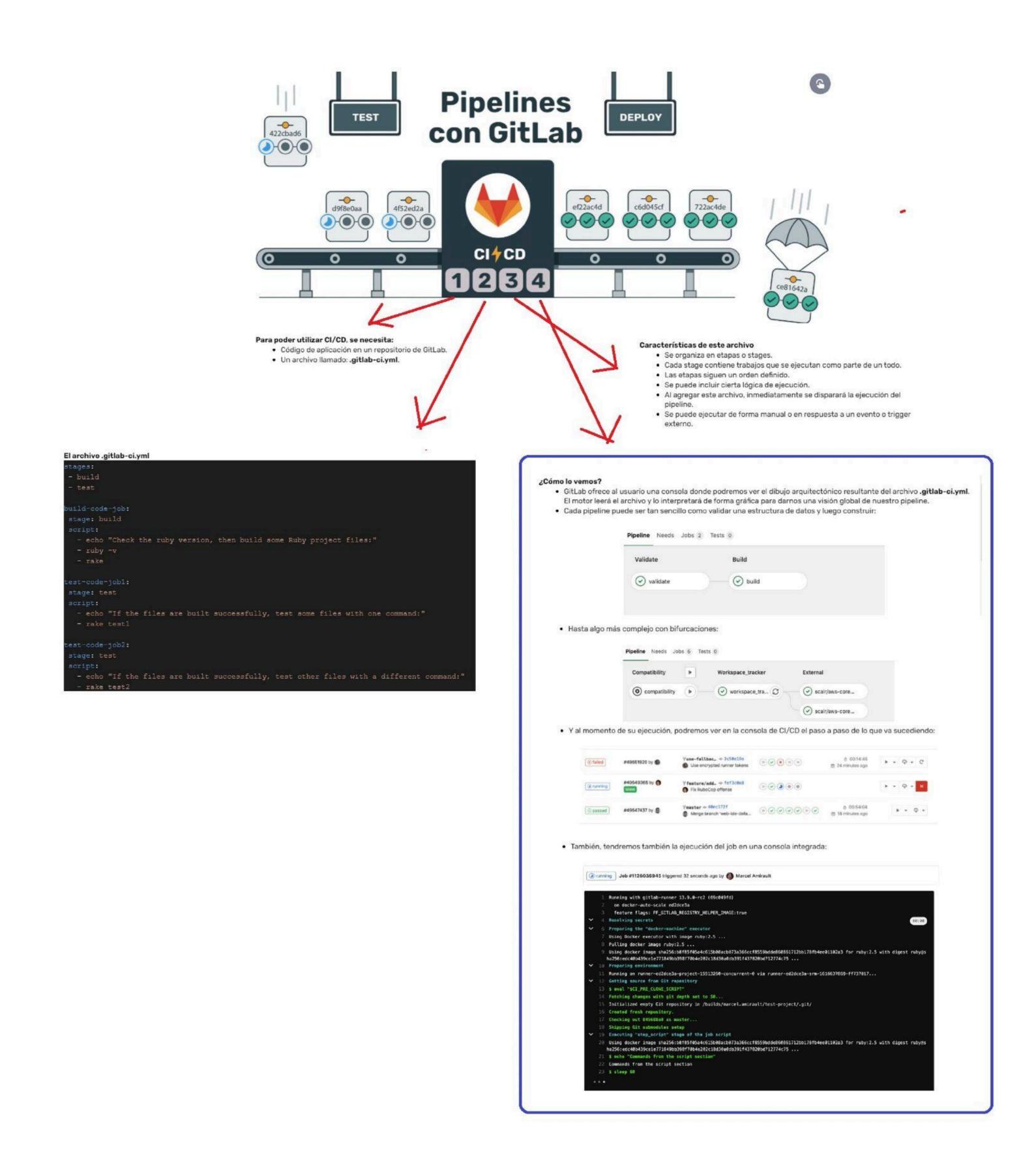
¿Qué es GitLab?

En el desarrollo de software los servicios de control de versiones son esenciales para administrar los repositorios GIT de cada proyecto.

GitLab está directamente integrado con GIT, sistema de control de versiones utilizado para rastrear cambios en archivos. Como servicio, GitLab nos ofrece almacenamiento de archivos en forma de repositorios con varias funciones:

- Hacer seguimiento de problemas.
- Efectuar versionando.
- Correr pipelines.
- Colaborar entre múltiples equipos.

¡Y muchas características extras! Para explorarlas y conocer en profundidad esta herramienta... ¡Sumergite en el mundo de GitLab!



¿GitHub o GitLab?

¡Pongámonos en tema!

Existen diferencias clave entre GitHub y GitLab, como:

- Integración continua: GitLab ofrece CI/CD sin costo alguno, de tal manera que no son necesarias herramientas de terceros, aunque también nos da la opción de integrarlo fácilmente.
- Autenticación: con GitLab podemos tener control granular de permisos en base a roles, mientras que con GitHub solo podemos dar acceso RW basado
- Importación y exportación: dada la alta capacidad de integración con otros productos, nos será realmente fácil importar datos desde distintos proveedores: Bitbucket, GitHub, Jenkins, etc. No obstante, GitHub requiere navegar a través de diversas documentaciones para poder lograr este mismo propósito. GitHub será a la vista más atractivo respecto de la

característica de importación de sus repositorios. Nota: Los proyectos de GitHub son tanto públicos como privados. Los códigos compartidos públicamente son abiertos y gratuitos. Por lo tanto, muchos desarrolladores usan repositorios públicos para compartir su software de código abierto en GitHub.

¡Amamos GitLab!

- GitLab es muy sencillo de configurar y administrar.
- Las revisiones de código y las solicitudes de "merge" son fáciles de usar y muy compactas. • Al ser una aplicación nativa de la nube, ofrece fuertes medidas de
- seguridad, como restricciones granulares y autenticación de usuario
- utilizando Kerberos. • Se integra fácilmente con diversas herramientas de terceros.
- Ofrece una organización e integración adecuada con Kubernetes. Permite una gestión de proyectos amplia y adaptable para acelerar el flujo de trabajo.

¡Las bondades de GitLab no terminan acá! Existen decenas de funciones, plugins para bajar, instalar y configurar muy útiles para el trabajo diario. El equipo de GitLab está constantemente creando nuevas características a un ritmo bastante superior en relación con otros productos, por lo que se aconseja suscribirse a reportes de actualización para estar al día con la herramienta.

El archivo .gitlab-ci.yml

```
stages:
 build
 - test
build-code-job:
stage: build
 script:
  - echo "Check the ruby version, then build some Ruby project files:"
  - ruby -v
  - rake
test-code-job1:
 stage: test
 script:
  - echo "If the files are built successfully, test some files with one command:"
  - rake test1
test-code-job2:
 stage: test
 script:
  - echo "If the files are built successfully, test other files with a different command:"
  - rake test2
```