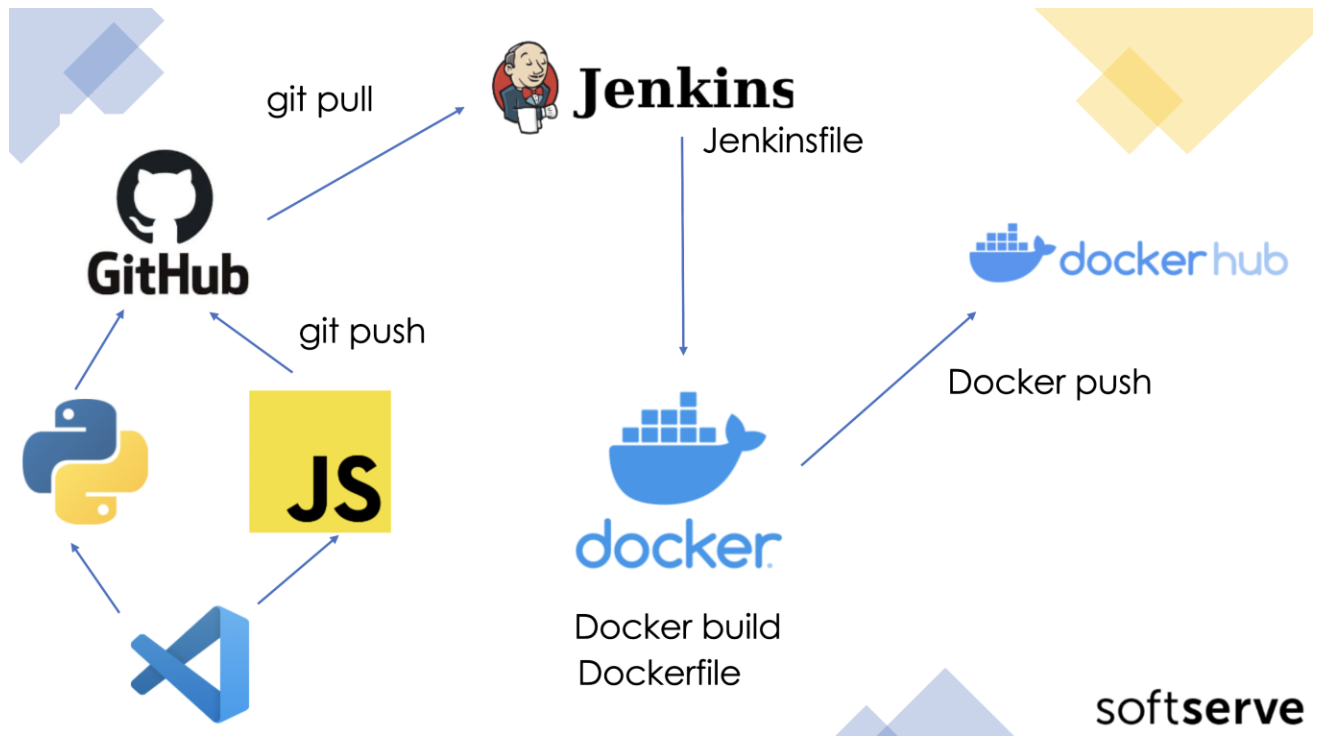


Laboratorio Continuous Integration & Continuous Delivery



1. Crear un proyecto en Visual Studio Code (o en el IDE de su preferencia) con algún desarrollo simple. Puede ser una página web estática (file.html)
2. Crear un repositorio público en github y subir allí el proyecto.
3. Descargar y correr localmente (Opcional: en un proveedor Cloud si lo desea) una imagen Docker de Jenkins en un container (<https://hub.docker.com/r/jenkins/jenkins>)
4. Instalar en el nuevo servidor de Jenkins los plugins recomendados por default. Adicional, instalar el plugin que permitirá hacer build y push de imágenes a Docker Hub (opcional: si desea puede hacerlo con otro plugin u otro método). Ver figura 1 (<https://plugins.jenkins.io/docker-build-publish/>)

CloudBees Docker Build and Publish plugin 1.3.3

This plugin enables building Dockerfile based projects, as well as publishing of the built images/repos to the docker registry.

[Report an issue with this plugin](#)

5. Crear una cuenta y un repositorio público en Docker Hub
6. Crear un nuevo Job/Pipeline en el servidor de Jenkins.
(<https://www.jenkins.io/doc/book/pipeline/jenkinsfile/>) El script del pipeline (jenkinsfile) deberá ser tomado desde el repositorio GitHub. Este Job deberá contar con dos Stages:

6.1 Stage 1: Tomará el dockerfile del proyecto y se ejecutará Docker Build

6.2. Stage 2: Tomará la imagen creada y se ejecutará un docker Push al repositorio de Docker Hub que previamente ya ha creado

Prueba de funcionamiento:

1. Al realizar un cambio en el proyecto y hacer commit & push a github, automáticamente se lanzará un trigger que ejecute el Job/pipeline de Jenkins de Manera automática.
2. La nueva Docker Image creada al finalizar la ejecución del pipeline, deberá visualizarse en el repositorio de Docker Hub.
- 3.

Entregable:

1. Archivo PDF donde se expliquen los procedimientos realizados y lo construido.
2. Link de repositorio GitHub: el cual debe incluir el proyecto desarrollado, jenkinsfile y Dockerfile.
3. Link de repositorio Docker Hub: donde se visualicen las imagenes creadas durante el proceso de pruebas.
4. Screenshots de una ejecución completa donde se visualice la hora, el commit &push realizado en github con algún cambio, la ejecución correcta del Job y la imagen creada en DockerHub. (agregar en PDF)
5. Cargar el entregable a: <https://forms.gle/JpnRPrt6z72Mf5DF7>

Deadline: Miércoles 20 de Julio 3:00pm

¡Happy Coding! Enjoy your life, and enjoy it more if it's coding with a good beer or an excellent coffee!

softserve