



códigofacilito



CI/CD

Cami Gomez - Backend developer





## >\_ Temas

- Continuous Integration y Continuous Delivery.
- Beneficios.
- Cómo funciona.
- Ejercicio.





## >\_ **Continuous Integration.**

Es una práctica de desarrollo de software en la que los miembros de un equipo integran su trabajo de forma frecuente, generalmente varias veces al día, en un repositorio compartido. Cada integración se verifica automáticamente mediante la ejecución de pruebas automatizadas, lo que permite detectar errores de manera temprana.





## >\_ **Continuous Delivery.**

Es una práctica de desarrollo de software que se basa en la Integración Continua (CI) y se enfoca en automatizar el proceso de llevar código desde el repositorio hasta el entorno de producción de manera rápida y confiable.





# >\_ Beneficios





## >\_ Beneficios

- Detección temprana de errores.
- Mejora de la calidad del software.
- Feedback rápido.
- Reducción de riesgos.
- Automatización del despliegue.
- Estabilidad y confiabilidad.



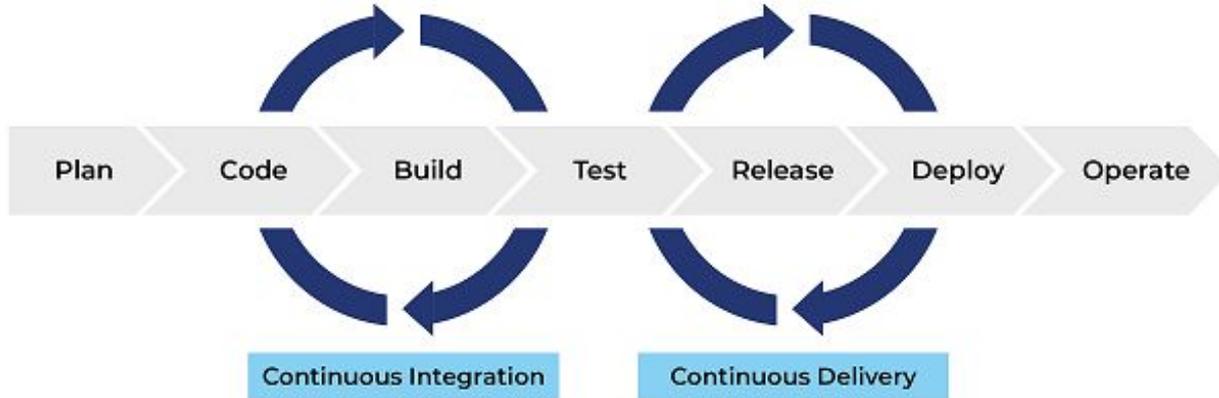


# >\_ **Cómo funciona el CI/CD.**





# CI/CD



# Collaborate

# Build

# Test

# Deploy

# Run

## Application Lifecycle Mgmt.



## SCM/VCS



## Testing



## Deployment



## Cloud / IaaS / PaaS



## Communication & ChatOps



## CI



## Config Mgmt. / Provisioning



## Orchestration & Scheduling



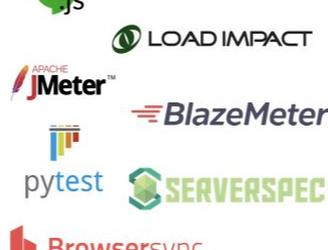
## Knowledge Sharing



## Build



## Testing



## Artefact Management



## BI / Monitoring / Logging



## Database Management



## Testing



## Artefact Management







# >\_ Flujo paso a paso.





## >\_ 1. Código

- **Desarrollo:** Los desarrolladores escriben código en sus entornos de desarrollo locales.
- **Revisión:** Cada cambio en el código es revisado por otros miembros del equipo, un proceso conocido como revisión de código o 'code review'.
- **Commit:** Tras la revisión, los cambios son consolidados ('commit') en el repositorio local del desarrollador.
- **Push:** Los cambios son enviados ('push') al repositorio central (como GitHub, GitLab, etc.), a menudo en una rama de características.





## >\_ 2. Build

- **Automatización del build:** Un servidor de CI (como Jenkins, GitHub Actions, etc.) detecta los cambios en el repositorio y desencadena un build automático.
- **Compilación:** El servidor de CI compila el código para verificar que se construye correctamente sin errores.
- **Preparación de Artefactos:** Se preparan artefactos (como binarios o paquetes) que serán usados en despliegues posteriores.





## >\_ 3. Pruebas

- **Pruebas Unitarias:** El servidor de CI ejecuta pruebas unitarias para validar la funcionalidad individual de componentes del código.
- **Pruebas de Integración:** Se ejecutan pruebas de integración para verificar que los diferentes módulos o servicios del sistema funcionan correctamente juntos.
- **Pruebas de Aceptación:** Pruebas automatizadas de alto nivel que evalúan la funcionalidad del sistema desde la perspectiva de un usuario final.
- **Análisis de Calidad:** Herramientas de análisis estático de código se pueden ejecutar para identificar problemas de calidad o seguridad.





## >\_ 4. Despliegue

- **Entorno de Pruebas:** Si las pruebas son exitosas, el código puede ser automáticamente desplegado a un entorno de pruebas para más validaciones.
- **Despliegue en Staging:** Después de las pruebas, el código se despliega en un entorno de 'staging' que simula la producción lo más cercanamente posible.
- **Aprobación para Producción:** Dependiendo de la política del equipo, un despliegue a producción puede requerir una aprobación manual.
- **Despliegue en Producción:** Finalmente, el código se despliega en el entorno de producción, haciéndolo accesible para los usuarios finales.





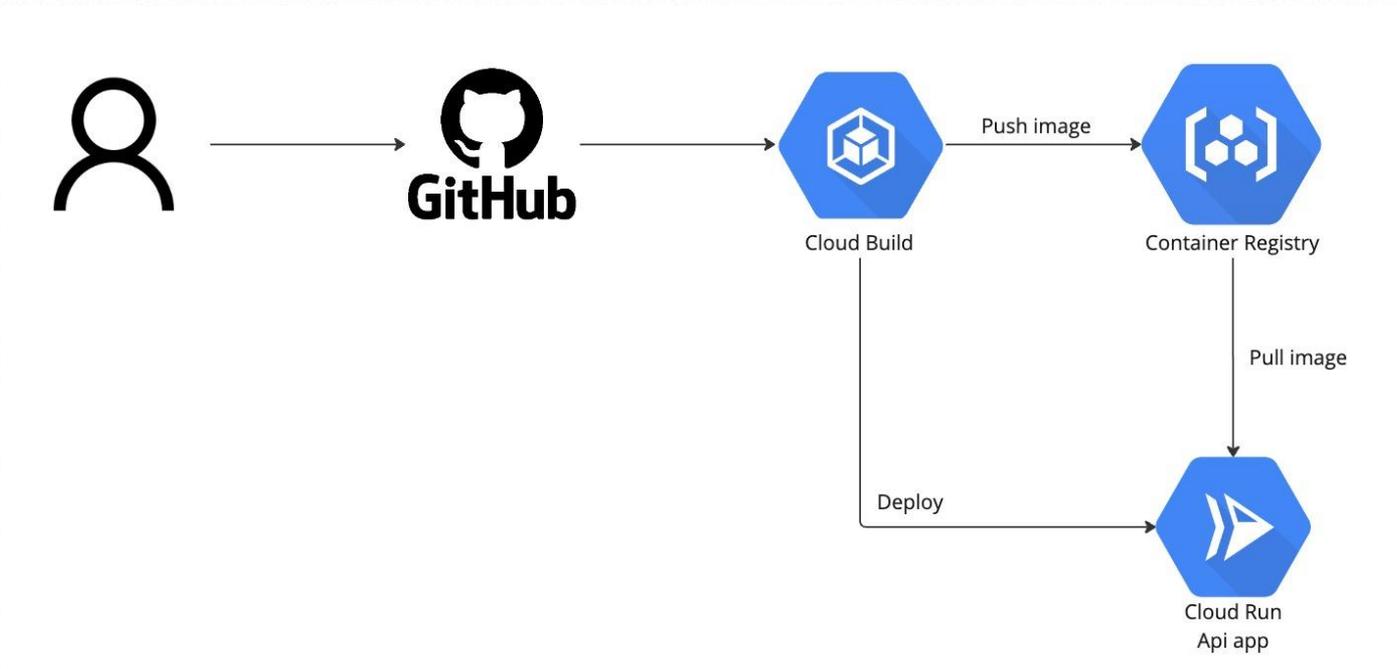
<b>Código</b>	<b>Artefacto</b>	<b>Deploy</b>
Github	Docker Image	Google Cloud Run, Amazon Elastic Container Service (ECS), Azure Container Instances.
GitLab	Archivo ZIP conteniendo el código fuente y las dependencias	Google Cloud Functions.
Bitbucket	Archivo JAR (Java ARchive)	(AWS) Elastic Beanstalk, Heroku, Tomcat.





# >\_ Ejercicio.





[https://github.com/camigomezdev/codigo\\_facilito\\_ci\\_cd](https://github.com/camigomezdev/codigo_facilito_ci_cd)



>\_ Gracias.

