Ciclo de Vida del Software

Impacto de Docker en CI/CD

Josue Antonio Romero Ramon

↓ Usa las flechas o los botones para navegar ↓

Ciclo de Vida del Software

El SDLC (Software Development Life Cycle) es un proceso estructurado que guía el desarrollo de software desde su concepción hasta su retiro.

Planificación

Requisitos y alcance

Diseño

Arquitectura del sistema

Desarrollo

Codificación

Pruebas

QA y testing

Despliegue

Producción

Mantenimiento

Actualizaciones

Monitoreo

Métricas y logs

Iteración

Mejora continua

¿Qué es CI/CD?

CI - Integración Continua

Continuous Integration

Práctica de fusionar cambios de código frecuentemente en un repositorio compartido, donde se ejecutan builds y pruebas automatizadas.

- Commits frecuentes
- Build automático
- Tests automáticos

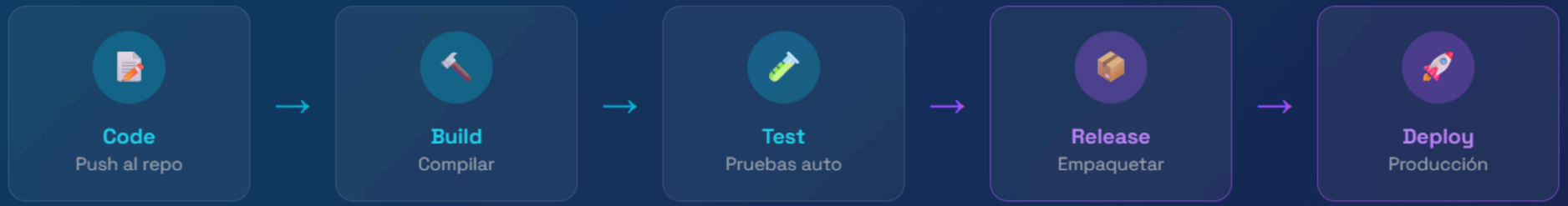
CD - Entrega/Despliegue Continuo

Continuous Delivery / Deployment

Automatización del proceso de release para que el código validado pueda desplegarse a producción en cualquier momento.

- Deploy automático
- Releases frecuentes
- Rollback fácil

Pipeline de CI/CD



CI - Integración Continua CD - Entrega Continua

¿Qué es Docker?

Plataforma de Contenedores

Docker permite empaquetar aplicaciones con todas sus dependencias en contenedores portátiles y ligeros.

Conceptos Clave:

- **Dockerfile**
  Instrucciones para construir imagen
- **Image**
  Plantilla inmutable del contenedor
- **Container**
  Instancia ejecutable de la imagen
- **Registry**
  Almacén de imágenes (Docker Hub)

```
docker build -t myapp . → docker push registry/myapp → docker run myapp
```

Impacto de Docker en CI/CD

Consistencia

"Works on my machine" ya no es problema. Mismo entorno en desarrollo, testing y producción.

Velocidad

Contenedores inician en segundos. Builds más rápidos con capas cacheadas.

Reproducibilidad

Cada build produce la misma imagen. Versionado de imágenes permite rollbacks.

Escalabilidad

Fácil escalar horizontal con orquestadores como Kubernetes.

Aislamiento

Cada contenedor aislado. Tests paralelos sin conflictos.

Portabilidad

Misma imagen funciona en cualquier infraestructura con Docker.

Workers / Runners

¿Qué son?

Son agentes o máquinas que ejecutan los jobs definidos en el pipeline de CI/CD. Escuchan instrucciones del servidor CI/CD y ejecutan las tareas asignadas.

GitLab Runners

- Shared, Group o Specific
- Executors: Shell, Docker, K8s
- Auto-scaling disponible

GitHub Actions Runners

- GitHub-hosted o Self-hosted
- Linux, Windows, macOS
- Matrices de ejecución

Jenkins Agents

- Master-Agent architecture
- Docker agents dinámicos
- Labels para targeting

Docker + Runners

- Runners en contenedores
- Docker-in-Docker (DinD)
- Entornos limpios por job

Conclusiones

- ✅ Docker revoluciona CI/CD al garantizar **consistencia** entre entornos
- ✅ Los **contenedores** aceleran builds y permiten paralelización
- ✅ Los **Runners/Workers** ejecutan pipelines de forma distribuida
- ✅ La combinación Docker + CI/CD es **estándar de la industria**

¡Gracias por su atención!