

Devops

TACS

CI/CD

TACS

Motivacion

— — —

- Releases lentos
- Errores manuales
- Procesos de deployment no estandarizados
- Baja calidad de software
- Problemas de seguridad

Continuous Integration

— — —

- Detección temprana de errores (linting, testing, building)
- Detección de problema de seguridad(dependency checks, static code analysis)
- Estandarización de buildings
- Feedback

CD

— — —

- Manejo de ambientes de despliegue.
- Deployment, con intervención.
- Delivery, sin intervención.

Git Branch Model

TACS

Ventajas

— — —

- Estandarización, aumento de gobernanza
- Mejora el flujo de trabajo
- Flujos específicos por tipos de branch

Gitflow

— — —

- Es un branching model
- Tipos de branch
 - Feature
 - Hotfix
 - Release
 - Estable
- Backports

laC

TACS

Infrastructure as code

— — —

- Abstraccion
- DSL (Domain specific language)
- Versionado
- Declarativo vs imperativo
- Vendor Lock in

Container Orchestration

TACS

Motivacion

— — —

- Facilitar la gestion de:
 - Networking
 - Escalamiento
 - Asignación de recursos
 - Rolling updates
 - Portabilidad

Kubernetes

TACS

Kubernetes

— — —

- Es una plataforma OpenSource
- Declarativo (Deseado vs Real)
- Container runtime

Componentes

— — —

- Nodos (master, workers)
- Kubernetes Api
- ETCD
- Scheduler
- Controllers
- Objects

GitOps

TACS

Componentes

— — —

- Git como fuente de verdad
- Versionado
- Automatización de cambios
- Separacion de aplicación y manifiestos