

CURSO: DEVOPS SENIOR

Módulo 8: Platform Engineering & Internal Developer Platforms (IDP)

Ejercicio Práctico 2

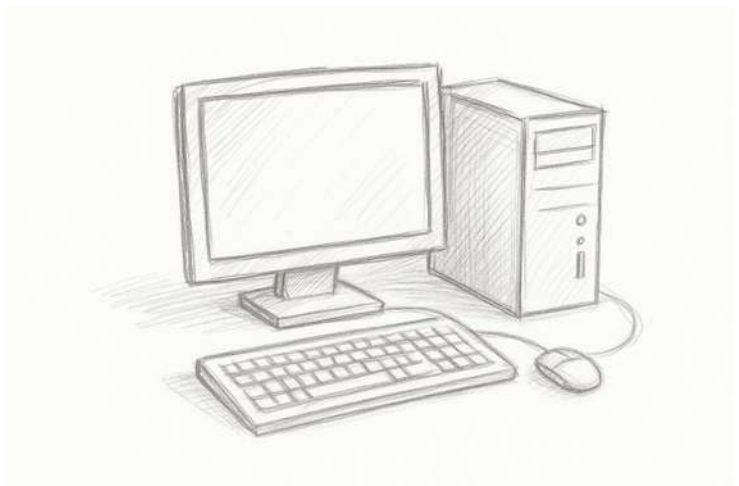
GitOps Stack Automático con Kustomize + ArgoCD + Backstage + CI/CD en GitHub

Objetivo:

Diseñar una estructura modular de GitOps Stacks, automatizar el despliegue de servicios versionados con Kustomize, integrar CI/CD en GitHub Actions y reflejar el estado en Backstage.

Resultado esperado:

El participante aplica los principios de GitOps en múltiples ambientes, integra observabilidad y CI/CD reales, y usa Backstage como centro de comando DevOps unificado.



INSTRUCCIONES:

- **Entorno sugerido:**
 - GitHub, Docker, Kustomize, ArgoCD, Backstage (local), clúster con Kind/Minikube.

- **Actividad guiada paso a paso:**

Estructura de repositorio GitOps:

- Cree un repo con la estructura:

```
csharp

gitops/
├── base/
│   ├── nginx/
│   └── overlays/
│       ├── dev/
│       └── prod/
```

- Use Kustomize para manejar las diferencias entre ambientes.

- **Integración CI/CD con GitHub Actions:**

Agrega un workflow que:

- Valide manifiestos (kubectl kustomize).
- Aplique cambios a dev automáticamente.
- Notifique en Pull Requests los cambios listos para prod.

- **Despliegue automático con ArgoCD:**
 - Crea aplicaciones declarativas por overlay (Application.yaml).
 - ArgoCD sincroniza los cambios desde el repositorio.
- **Visualización en Backstage:**
 - En cada carpeta de overlay, agrega un catalog-info.yaml.
 - Backstage detecta los cambios y actualiza el estado.
- **Feedback técnico esperado:**
 - CI/CD funcional: push en main genera validación y despliegue.
 - Backstage refleja estado y versión actual.
 - ArgoCD muestra sincronización completa.
 - Métricas de observabilidad pueden verse en Grafana (integrado o vía plugin).