

CURSO: DEVOPS SENIOR

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

Ejercicio Práctico 1

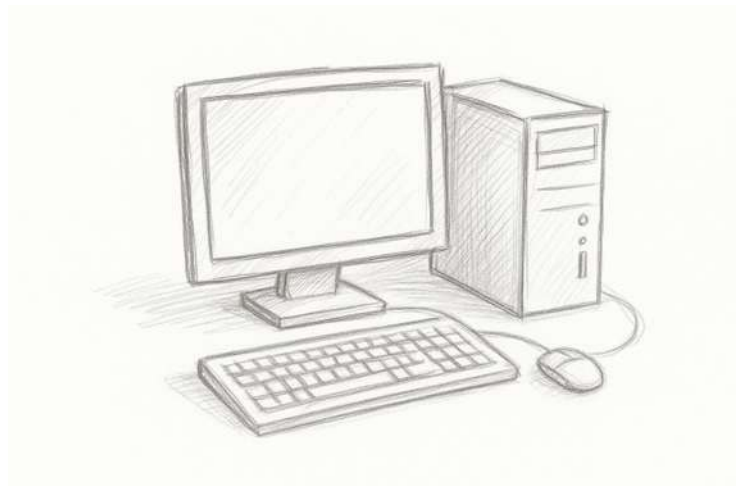
Registro e integración de un microservicio en Backstage con GitOps y CI/CD

Objetivo:

Instalar y personalizar una instancia de Backstage, conectar repositorios Git con despliegue GitOps, e integrar paneles de observabilidad (Prometheus/Grafana) para visualizar el estado de servicios en tiempo real.

Resultado esperado:

El participante entiende cómo construir un portal DevOps interno que centraliza CI/CD, estado de servicios y métricas observables, usando Git como fuente de verdad.



INSTRUCCIONES:

- **Entorno sugerido:**

- Local con Docker y Node.js.
- Cuenta gratuita en GitHub.
- Clúster con Kind o Minikube.
- kubectl, helm, kustomize, argocd, prometheus, grafana.

- **Actividad guiada paso a paso:**

Instalación local de Backstage:

- Sigue el quickstart oficial:

```
bash
npx @backstage/create-app
cd my-app
yarn dev
```

- **Catalogación de un repositorio GitOps:**

- Crea un archivo catalog-info.yaml en un repo Git que contenga manifiestos K8s versionados con Kustomize.
- Importa el repo en Backstage desde <http://localhost:3000/catalog-import>.

- **Integración con ArgoCD (simulada):**

- Instala ArgoCD en el clúster local.
- Usa el plugin backstage-plugin-argo-cd para mostrar estado de sincronización de aplicaciones en el portal.
- Enlace: <https://github.com/RoadieHQ/backstage-plugin-argo-cd>

- **Visualización de Observabilidad:**
 - Instala Prometheus + Grafana en el clúster.
 - Integra dashboards vía el plugin backstage-plugin-grafana.
- **Feedback técnico esperado:**
 - Los servicios aparecen sincronizados en ArgoCD.
 - El portal Backstage muestra el estado del deployment.
 - Los dashboards reflejan métricas reales (CPU, latencia, etc).