CURSO: DEVOPS SENIOR

Módulo 7: Infraestructura como Código avanzada

Ejercicio Práctico 2

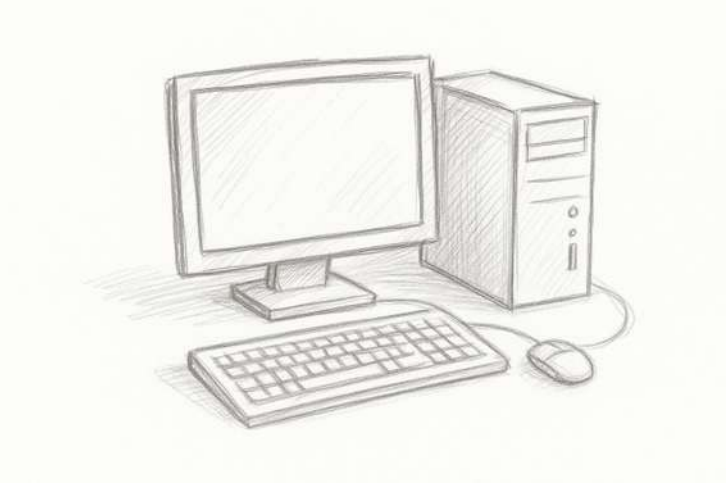
Políticas de Seguridad y Cumplimiento con Sentinel Open Source + Terraform Validate + Control de GitOps

Objetivo:

Aplicar políticas de control sobre IaC usando Sentinel en modo local, verificar su cumplimiento antes del despliegue, e integrarlo con un flujo GitOps simulado para control de cumplimiento en CI/CD.

Resultado esperado:

Aplicar principios de compliance-as-code, integrando validación estructural (terraform validate), lógica de negocio (sentinel) y control declarativo de cumplimiento en CI/GitOps.



INSTRUCCIONES:

- **Entorno sugerido:**
    - Local: Terraform CLI, Sentinel CLI, Git.
    - https://github.com/hashicorp/sentinel y el simulador oficial.
    - No se requiere Terraform Cloud (se usa modo local de Sentinel).

- **Actividad guiada paso a paso:**

  Infraestructura base:

    - Cree un main.tf que despliegue un bucket S3 (puedes simularlo con local_file si no deseas usar AWS).
    - Cree una variable versioning_enabled booleana.

- **Política Sentinel local:**
    - Defina una política (no_unversioned_buckets.sentinel) que prohíba crear buckets sin versionado activado.

- **Usa la CLI de Sentinel para ejecutar:**

```bash
sentinel test -policy no_unversioned_buckets.sentinel -run my_test.sentinel
```

- **Integración GitOps simulada:**

  Configure una acción en GitHub (.github/workflows/test.yml) que:

  - Corra terraform validate y sentinel test.
  - No permita hacer merge si la política falla.

- **Simulación de incumplimiento:**

  - Cambie el código para deshabilitar el versionado.
  - Haga commit y verifique que Sentinel bloquea el flujo.

- **Feedback técnico esperado:**

  - Logs del test Sentinel muestran si la política pasa o falla.
  - GitHub Actions bloquea merges si no se cumplen las reglas.
  - terraform validate debe funcionar antes de apply.