CURSO: DEVOPS SENIOR

Módulo 3: Seguridad avanzada y DevSecOps

Ejercicio Práctico 2

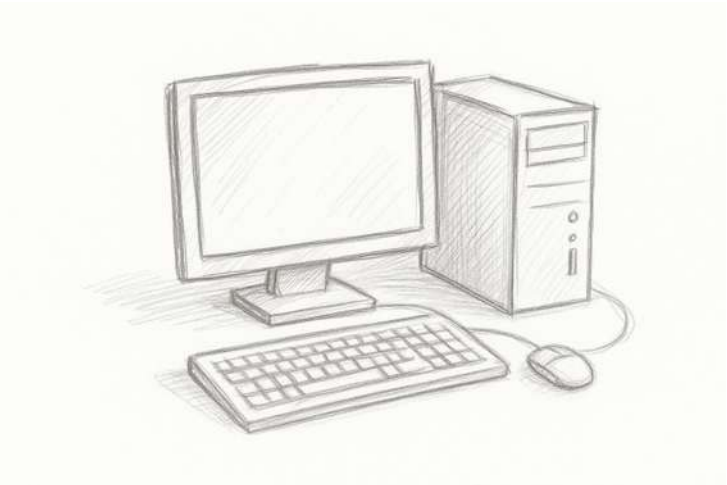
Seguridad en Kubernetes con HashiCorp Vault + Roles RBAC + Secretos Dinámicos

Objetivo:

Aplicar políticas de control de acceso y gestión segura de secretos en Kubernetes mediante Vault, integrando autenticación dinámica, uso de RBAC, y políticas finamente segmentadas.

Resultado esperado:

Comprender el uso de Vault como solución de gestión de secretos, el principio de menor privilegio y los riesgos de exposición en Kubernetes.



INSTRUCCIONES:

Entorno gratuito sugerido:

- Instale Minikube o usa Kind.
- Instale kubectl, helm y vault CLI.
- Siga esta guía oficial gratuita para instalar Vault en Kubernetes usando Helm: https://developer.hashicorp.com/vault/tutorials/kubernetes/kubernetes-minikube

Actividad práctica paso a paso:

- Implemente un despliegue Kubernetes que necesita acceder a una base de datos ficticia (ej: MySQL local).
- Cree un secreto dinámico en Vault para credenciales de MySQL.
- Configure autenticación mediante Kubernetes Service Accounts para permitir que solo ciertos pods puedan leer el secreto.

Aplica políticas RBAC para que:

- Un pod llamado frontend acceda a db-password.
- Un pod llamado logger no tenga permisos.

Verificación del aprendizaje:

- Desde dentro del pod frontend, use un Job o script para consumir el secreto vía Vault Agent.
- Intente hacer lo mismo desde el pod logger y demuestre que falla por falta de permisos.

Feedback técnico esperado:

- Validación de que los secretos no están en texto plano en los YAML.
- Logs de Vault donde se registra la autenticación y acceso.
- Evidencia de acceso autorizado/desautorizado según políticas RBAC.