Bootcamp DevOps

Arquitectura de la solución tecnológica para el despliegue de una API REST

Fran Marín Pérez

# Introducción

La API REST consiste en una aplicación web que implementa tres métodos HTTP: GET, POST y DELETE. La API interactuará con una base de datos, gestionando el almacenamiento, recuperación y eliminación de la información.

# Descripción de la arquitectura del sistema

El diseño propuesto se basa en una arquitectura de microservicios mediante el uso de Kubernetes para asegurar una eficiente gestión de los diferentes componentes mediante el escalado horizontal, balanceo, alta disponibilidad y tolerancia a fallos.

## Componentes principales

### Capa de aplicación

Los siguientes servicios son desplegados en un clúster de Kubernetes:

#### API

* **Aplicación web en** **Flask** (Python).
* Contenerizacióncon Docker para garantizar flexibilidad y portabilidad.

#### Base de datos

* **PostgreSQL.**
* Despliegue de réplicas de lectura para una mayor disponibilidad.
* Implementación de un sistema de *backup* y recuperación.

#### Telemetría

* Elastic Stack (ELK) para *logging.*
* Stack Prometheus + Grafana para monitorización.

### Infraestructura de red

* Uso de **balanceador de carga** para distribuir las conexiones entre instancias.
* Terminación de **SSL** para encriptar las conexiones a la aplicación web.

## Seguridad

* Control de acceso basado en roles (**RBAC**).
* Gestión de secretos con **Vault.**

# Arquitectura Cloud

# Descripción del ciclo de vida

### GitHub

* *GitHub* para alojar el código fuente de la aplicación.
* Uso del *branching model*[Trunk-Based Development](https://trunkbaseddevelopment.com/).
* GitHub Actions para ejecutar test y verificar la validez de las nuevas *features* de la aplicación.

### Registro de artefactos

* Uso de un *container registry* para almacenar las nuevas versiones de la aplicación en una imagen Docker.

### Entornos de despliegue

* Development (dev): Para desarrolladores
* Production (pro): Para el usuario final