

**1. Introducción**

**1.1. Descripción del proyecto**

El proyecto consiste en la implementación de una intranet alojada en la nube utilizando Kubernetes en instancias EC2 de AWS. En lugar de desarrollar la intranet desde cero, se utilizará un CMS basado en WordPress para facilitar la gestión de contenidos. La autenticación de usuarios se realizará mediante la API de GitHub, y se integrará ArgoCD para la automatización del despliegue continuo. Toda la infraestructura se definirá mediante **Terraform** y su código se alojará en **GitHub** para una gestión eficiente y automatizada.

**1.2. Objetivos del proyecto**

* Aprender y aplicar Kubernetes en un entorno cloud.
* Desplegar una intranet funcional utilizando WordPress.
* Implementar CI/CD con ArgoCD para la gestión automatizada de despliegues.
* Integrar una autenticación segura con OAuth de GitHub.
* Automatizar la infraestructura con Terraform y alojarla en GitHub.
* Optimizar el costo de infraestructura utilizando soluciones económicas y eficientes.

**2. Análisis del contexto y justificación de la propuesta**

Kubernetes no es la opción más ligera para una intranet, pero se elige por dos razones principales:

1. **Aprendizaje y experimentación**: Se busca adquirir habilidades en Kubernetes, una tecnología clave en DevOps y Cloud Computing.
2. **Migración futura**: Se plantea como base para la migración de aplicaciones de una empresa que actualmente opera en servidores físicos.
3. **Infraestructura como código**: Se usará Terraform para definir y gestionar toda la infraestructura en AWS, asegurando despliegues reproducibles y escalables.

**3. Estado del arte**

**Tecnologías existentes y referencias**

Para la implementación de una intranet, existen diversas soluciones en el mercado:

* **SharePoint**: Plataforma de Microsoft para la creación de intranets empresariales, con integración en el ecosistema de Office 365.
* **Google Workspace**: Solución basada en la nube que permite la creación de espacios colaborativos con Google Sites.
* **CMS populares (WordPress, Joomla, Drupal)**: Sistemas de gestión de contenidos utilizados para construir portales web, blogs e intranets.
* **Soluciones Serverless (AWS Lambda, DynamoDB, Amplify)**: Alternativas modernas para desplegar aplicaciones sin necesidad de gestionar servidores.
* **Infraestructura como código (IaC) con Terraform**: Permite definir y desplegar infraestructura en la nube de manera declarativa y automatizada.

En este proyecto, se elige **WordPress** debido a su facilidad de uso, amplia documentación y la posibilidad de integrarse con Kubernetes mediante contenedores. Además, **Terraform** se utilizará para la gestión automatizada de la infraestructura en AWS.

**4. Requisitos del proyecto**

**4.1.1. Requisitos funcionales**

* Implementación de una intranet con WordPress.
* Autenticación con GitHub OAuth.
* Panel de usuario autenticado.
* Gestión de contenidos y usuarios.
* Infraestructura desplegada con Terraform y almacenada en GitHub.

**4.1.2. Requisitos no funcionales**

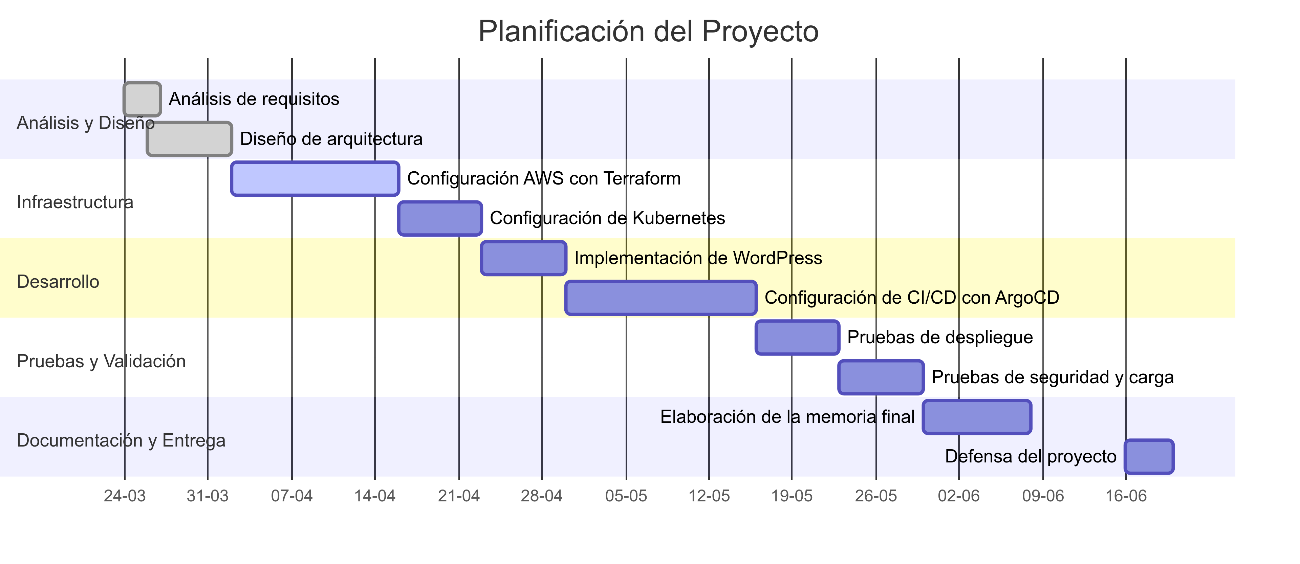
* Disponibilidad en la nube.
* Despliegue automático con CI/CD.
* Escalabilidad del sistema con Kubernetes.
* Definición de infraestructura como código (IaC) con Terraform.

**5. Planificación**

**5.1. Fases del proyecto**

1. Análisis y diseño.
2. Infraestructura.
3. Desarrollo.
4. Pruebas y validacion
5. Documentacion y entrega

**5.2. Cronograma de trabajo**



**5.3. Recursos necesarios**

* AWS EC2 para Kubernetes.
* DockerHub para almacenamiento de imágenes.
* GitHub para repositorios y control de versiones.
* Terraform para la gestión de infraestructura.
* Herramientas de monitoreo como Prometheus y Grafana.