

# Infraestructura Cloud y DevOps

## DevOps y Arquitectura Escalable

Diseño de plataforma de video bajo demanda con  
infraestructura cloud moderna y segura



# Contenido

01

## Diseño de Infraestructura Cloud

Modelos de implementación y servicios en la nube

02

## Arquitectura de Cómputo

Escalabilidad y balanceamiento de carga

03

## Redes y Seguridad

VPC, CDN y estrategias de protección

04

## Automatización CI/CD

Pipelines y infraestructura como código





# Contexto Proyecto



## Plataforma VOD

Desarrollar plataforma video bajo demanda similar a Netflix para usuarios globales

## Requisitos Técnicos

Infraestructura escalable, segura y eficiente soportando miles de usuarios simultáneamente



# Diseño Infraestructura Cloud

## Modelo Implementación

AWS: La nube pública (AWS) permite elasticidad, pago por uso, alta disponibilidad global y servicios gestionados, lo que reduce la complejidad operativa. Esto es ideal para una startup o empresa en crecimiento que busca escalar rápido sin altos costos iniciales.

## Servicios Cloud

IaaS (EC2): Para servicios como bases de datos en modo administrado (RDS).

PaaS (ECS/EKS): Para desplegar microservicios y manejar escalabilidad.

FaaS (Lambda): Para tareas como transcodificación o generación de thumbnails.

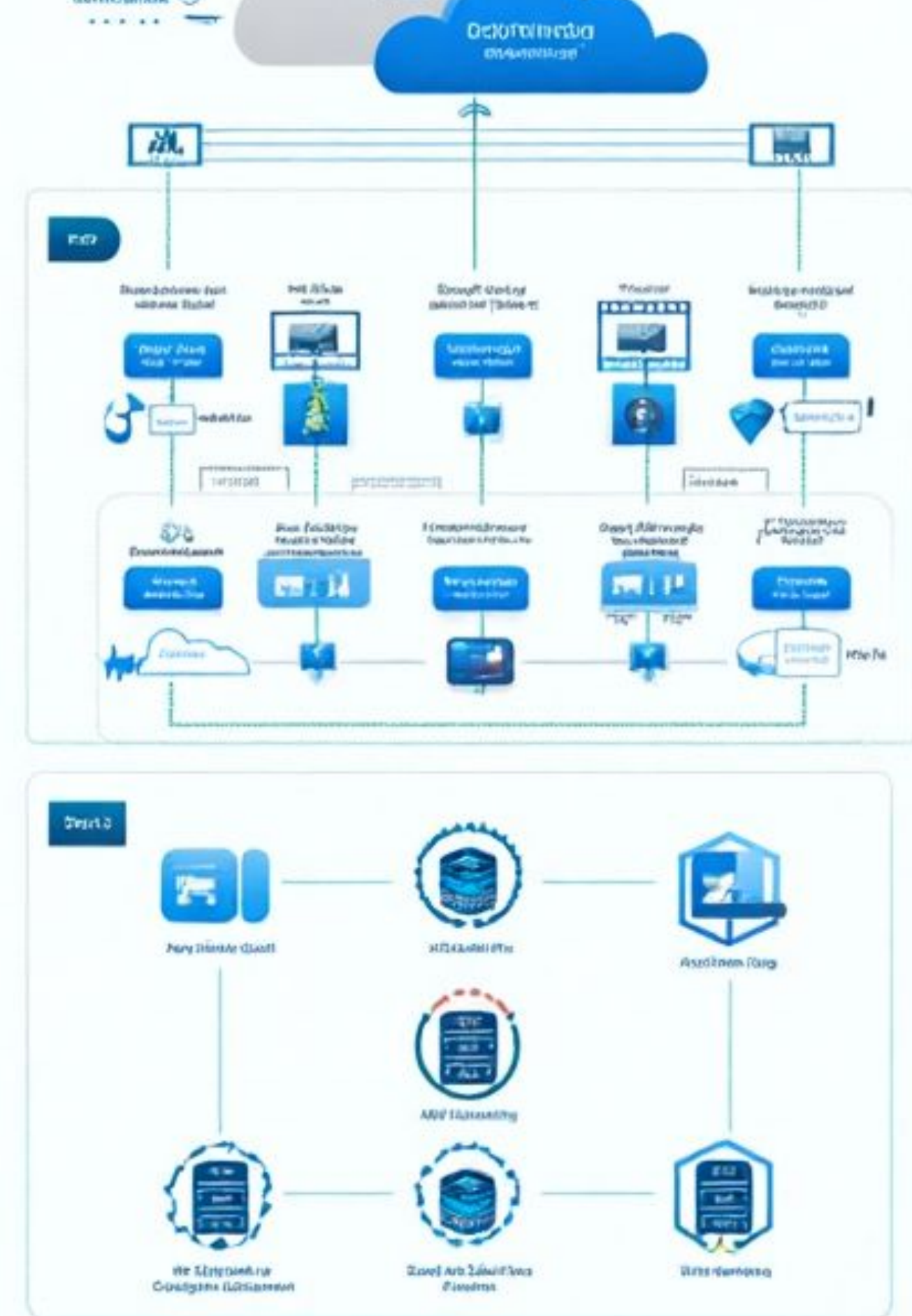
## Almacenamiento Distribuido

Amazon S3: Para almacenar archivos de video.

Amazon Glacier: Para archivos de respaldo o históricos.

Amazon RDS: Para metadatos de usuarios y videos (PostgreSQL).

ElastiCache (Redis): Para sesiones de usuario y mejora de tiempos de respuesta.





## Escalabilidad Horizontal

### Auto Scaling

HPA (Horizontal Pod Autoscaler): en EKS según uso de CPU o peticiones.

Auto Scaling Groups: en caso de usar EC2 como worker nodes.

### Load Balancing

Load Balancer (ELB): distribuye tráfico HTTP/S a múltiples pods.

## Orquestación Contenedores AWS EKS

### Kubernetes

Kubernetes ofrece portabilidad, escalabilidad automática, gestión avanzada de recursos y facilita CI/CD. Es ideal para un entorno con múltiples microservicios que deben escalar en demanda.

### Docker

Contenedorización de aplicaciones y servicios distribuidos

### Service Mesh

Comunicación segura entre microservicios y balanceadores

### Registry

Gestión centralizada de imágenes de contenedores

## Monitorización

### Métricas

CloudWatch y Prometheus para seguimiento de rendimiento

### Alertas

Sistema de notificaciones automáticas por umbrales

**99.9%**

Disponibilidad Sistema

**< 2s**

Latencia Streaming

**10K+**

Usuarios Concurrentes



## Redes Seguras

- VPC con subredes públicas (para Load Balancer y frontend) y subredes privadas (para backend y base de datos).
- NAT Gateway para que recursos privados accedan a Internet.



## Seguridad Avanzada

- IAM: acceso basado en roles.
- AWS Secrets Manager: para almacenar tokens/API Keys de forma segura.
- TLS/SSL para todo el tráfico de red.
- KMS: para cifrado en reposo (datos en RDS y S3).





# CDN

Amazon CloudFront: entrega de video con baja latencia a nivel global, cacheando contenido en edge locations.



Edge Locations



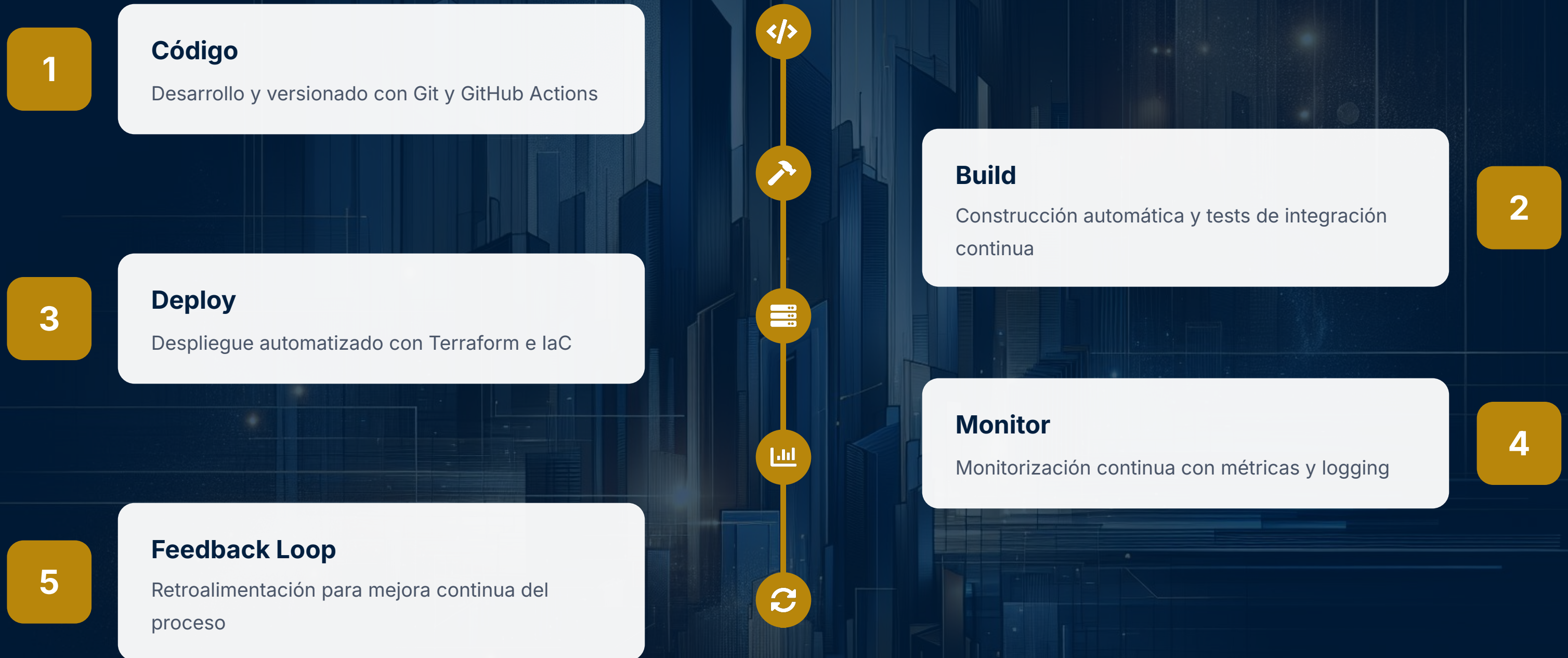
Cache  
Inteligente



Geolocalización



# Pipeline CI/CD





# Pipeline CI/CD Código

## Github actions

```
deploy.yml
1  name: CI/CD - VOD Platform
2
3  on:
4    push:
5      branches:
6        - main # Ejecuta solo en rama main
7    pull_request:
8      branches:
9        - main
10
11 jobs:
12   deploy:
13     runs-on: ubuntu-latest
14
15     steps:
16       # 1. Clona el repositorio
17       - name: Checkout code
18         uses: actions/checkout@v3
19
20       # 2. Configura credenciales de AWS (requiere secretos configurados)
21       - name: Configure AWS Credentials
22         uses: aws-actions/configure-aws-credentials@v2
23         with:
24           aws-access-key-id: ${ secrets.AWS_ACCESS_KEY_ID }
25           aws-secret-access-key: ${ secrets.AWS_SECRET_ACCESS_KEY }
26           aws-region: us-east-1
27
28       # 3. Configura Terraform CLI
29       - name: Set up Terraform
30         uses: hashicorp/setup-terraform@v2
31         with:
32           terraform_version: 1.5.5
33
34       # 4. Inicializa Terraform (descarga plugins, prepara estado)
35       - name: Terraform Init
36         run: terraform init
37
38       # 5. Planifica cambios en la infraestructura
39       - name: Terraform Plan
40         run: terraform plan
41
42       # 6. Aplica cambios automáticamente
43       - name: Terraform Apply
44         run: terraform apply -auto-approve
45
46       # 7. Login a Amazon ECR (repositorio de contenedores)
47       - name: Login to Amazon ECR
48         uses: aws-actions/amazon-ecr-login@v1
49
50       # 8. Construye y sube la imagen Docker del backend
51       - name: Build and Push Docker Image
52         run: |
53           IMAGE_URI="${ secrets.AWS_ACCOUNT_ID }.dkr.ecr.us-east-1.amazonaws.com/vod-backend:latest"
54           docker build -t vod-backend .
55           docker tag vod-backend:latest $IMAGE_URI
56           docker push $IMAGE_URI
57
58       # 9. Configura acceso al cluster EKS
59       - name: Configure kubeconfig
60         run: |
61           aws eks update-kubeconfig --name vod-cluster --region us-east-1
62
63       # 10. Despliega los manifiestos de Kubernetes (k8s/)
64       - name: Deploy to EKS
65         run: |
66           kubectl apply -f k8s/
67
```



# Pipeline CI/CD Código

## Terraform

```
main.tf x
main.tf > module "eks" > enable_irs
1 # ----- providers.tf -----
2 # Define el proveedor y región a utilizar
3 provider "aws" {
4     region = "us-east-1"
5 }
6
7 # ----- variables.tf -----
8 # Variables reutilizables para entorno
9 variable "region" {
10     default = "us-east-1"
11 }
12
13 variable "cluster_name" {
14     default = "vod-cluster"
15 }
16
17 variable "vpc_id" {
18     description = "ID de la VPC existente"
19 }
20
21 variable "subnet_ids" {
22     type = list(string)
23     description = "Lista de subredes para los nodos EKS"
24 }
25
26 # ----- s3.tf -----
27 # S3 para almacenamiento de videos
28 resource "aws_s3_bucket" "vod_videos" {
29     bucket = "vod-video-storage-bucket"
30
31     tags = {
32         Name = "VideoStorage"
33         Project = "VOD"
34     }
35 }
```

```
36
37 # Versionado para manejar múltiples versiones de videos
38 resource "aws_s3_bucket_versioning" "versioning" {
39     bucket = aws_s3_bucket.vod_videos.id
40
41     versioning_configuration {
42         status = "Enabled"
43     }
44 }
45
46 # Cifrado del bucket con AES256 (por defecto)
47 resource "aws_s3_bucket_server_side_encryption_configuration" "encryption" {
48     bucket = aws_s3_bucket.vod_videos.id
49
50     rule {
51         apply_server_side_encryption_by_default {
52             sse_algorithm = "AES256"
53         }
54     }
55 }
56
57 # ----- eks.tf -----
58 # Crea el cluster EKS utilizando un módulo oficial de Terraform
59 module "eks" {
60     source          = "terraform-aws-modules/eks/aws"
61     cluster_name    = var.cluster_name
62     version         = "1.27"
63     subnets        = var.subnet_ids
64     vpc_id          = var.vpc_id
65 }
```

```
66 # Habilita la autenticación IRSA
67 # (IAM Roles for Service Accounts)
68 enable_irs    = true
69
70 # Define un grupo de nodos (EC2) para ejecutar los pods
71 node_groups = {
72     default = {
73         desired_capacity = 2
74         max_capacity     = 4
75         min_capacity     = 1
76         instance_type    = "t3.medium"
77
78         labels = {
79             role = "general"
80         }
81     }
82 }
83
84 tags = {
85     Environment = "dev"
86     Project     = "VOD"
87 }
88 }
89 }
```



# Herramientas y Tecnologías Principales

## Cloud Providers

- AWS con EC2, S3, CloudFront
- Servicios serverless  
Lambda/Functions

## DevOps Tools

- Docker para contenedorización
- Kubernetes para orquestación
- Terraform para IaC
- GitHub Actions para CI/CD

## Monitorización

- CloudWatch para métricas AWS
- Prometheus y Grafana stack
- ELK Stack para logging
- Alertmanager para notificaciones





# Infraestructura Cloud VOD - Proyecto DevOps

Diseño completo de arquitectura escalable para plataforma de video bajo demanda empresarial

[https://github.com/CesarHerr/Ev\\_devOps\\_Adalid/tree/main/modulo5](https://github.com/CesarHerr/Ev_devOps_Adalid/tree/main/modulo5)