Prueba Técnica – CloudOps AWS

Objetivo: Plantilla lista para usar para entregar la prueba técnica de una app web 3 capas en AWS (escalable, HA, costo‑eficiente). Incluye README, estructura sugerida, guía de despliegue y anexos.

Resumen

Arquitectura propuesta: ECS Fargate (frontend/back) + RDS/Aurora Serverless v2 o DynamoDB + S3 + CloudFront + ALB + WAF + Secrets Manager + VPC privada/publica + NAT + CloudWatch.

Escalabilidad: Auto Scaling en Fargate, caché en CloudFront y ElastiCache

Alta disponibilidad: Multi‑AZ (RDS), 2+ AZ en subnets para ECS y ALB

Costo: pagar por uso (Fargate/Lambda), RDS/Aurora Serverless v2, Fargate Spot opcional

Seguridad: IAM minimo privilegio, SG por capa, WAF, TLS, secretos en Secrets Manager, KMS

Diagrama de arquitectura

Archivo: docs/arquitectura.drawio.pdf

Flujos: usuario → CloudFront → ALB → ECS (frontend/backend) → DB; artefactos en S3; logs/metrics en CloudWatch; secretos en Secrets Manager.

Red: VPC /16, 2–3 AZ, subnets públicas (ALB/NAT) y privadas (ECS/DB), rutas separadas, NACL conservadoras.

Decisiones de diseño

**Compute:** ECS Fargate (sin gestionar EC2, buen fit para microservicios).

Datos: Aurora Serverless v2 si necesitamos SQL y transaccional.

Caché: CloudFront (estático) + ElastiCache Redis para sesiones/queries.

Red y seguridad: VPC aislada, SG mínimos necesarios, WAF managed rules, TLS 1.2+, secretos rotados.

Observabilidad: CloudWatch metrics/logs/alarms, X‑Ray tracing, dashboards por servicio, alarmas a SNS.

Pre‑requisitos

Cuenta AWS con permisos para: VPC, IAM, ECS, RDS/Dynamo, ALB, CloudWatch, WAF, ACM, S3, Secrets Manager.

Domain/SSL: Route53 + certificado en ACM en la región del ALB/CloudFront.

Artefactos app: imagen(es) en ECR (o empaquetado para Lambda). S3 para estáticos frontend.

Herramientas: Terraform ≥ 1.6, AWS CLI v2, Git

Despliegue (paso a paso)

1. Autenticación AWS

aws configure sso

1. Inicializar Terraform

cd iac/terraform/live/<env> #puede ser dev o prod

terraform init

1. Pre-plan

terraform validate && terraform fmt -recursive

terraform plan -out tfplan

1. Aplicar

terraform apply tfplan

1. Salidas de recursos

terraform output

Observabilidad

Logs: CloudWatch Log Groups por servicio (ECS task, ALB access logs en S3 si habilitas).

Métricas: CPU/Mem (ECS), Latencia/5xx (ALB), Conexiones/ACUs (RDS/Aurora), DynamoDB RCUs/WCUs.

Seguridad

IAM: roles por servicio (task roles), permisos mínimos.

Red: SG por capa, subnets privadas para ECS/DB, sólo ALB público.

Datos: cifrado en reposo (KMS) y en tránsito (TLS). Secretos en Secrets Manager.

WAF: managed rules (Common, SQLi/XSS), rate limiting.

Estimación de costos

Se adjunta calculadora.

Se estiman costos de Fargate, aurora rds con un cluster, costos de cloudwatch, elastic cache, salida de red a internet un aprox 1tb al mes en prd y 200gb en dev, cada uno de estos valores puede variar y ser mas costo eficiente si adquirimos compromisos de consumo, ya sea reservas de REDIS, o saving plan compute o una combinación de ambas.

<https://calculator.aws/#/estimate?id=d1b0103dfacccf0fb7bd652fd3e8af824fd5ec2a>

archivo: documents/ INFRAESTRUCTURA PRAGMA - Calculadora de precios de AWS.pdf

CI/CD

Archivo: .github/workflows/ci-cd.yaml

Contempla el despligue de un archivo dockerfile publicándolo en ECR y cambiando el task definition a la ultima imagen en el repositorio y se lanza en ramas como dev, preprod y prod.

Anexos

docs/arquitectura.drawio.pdf