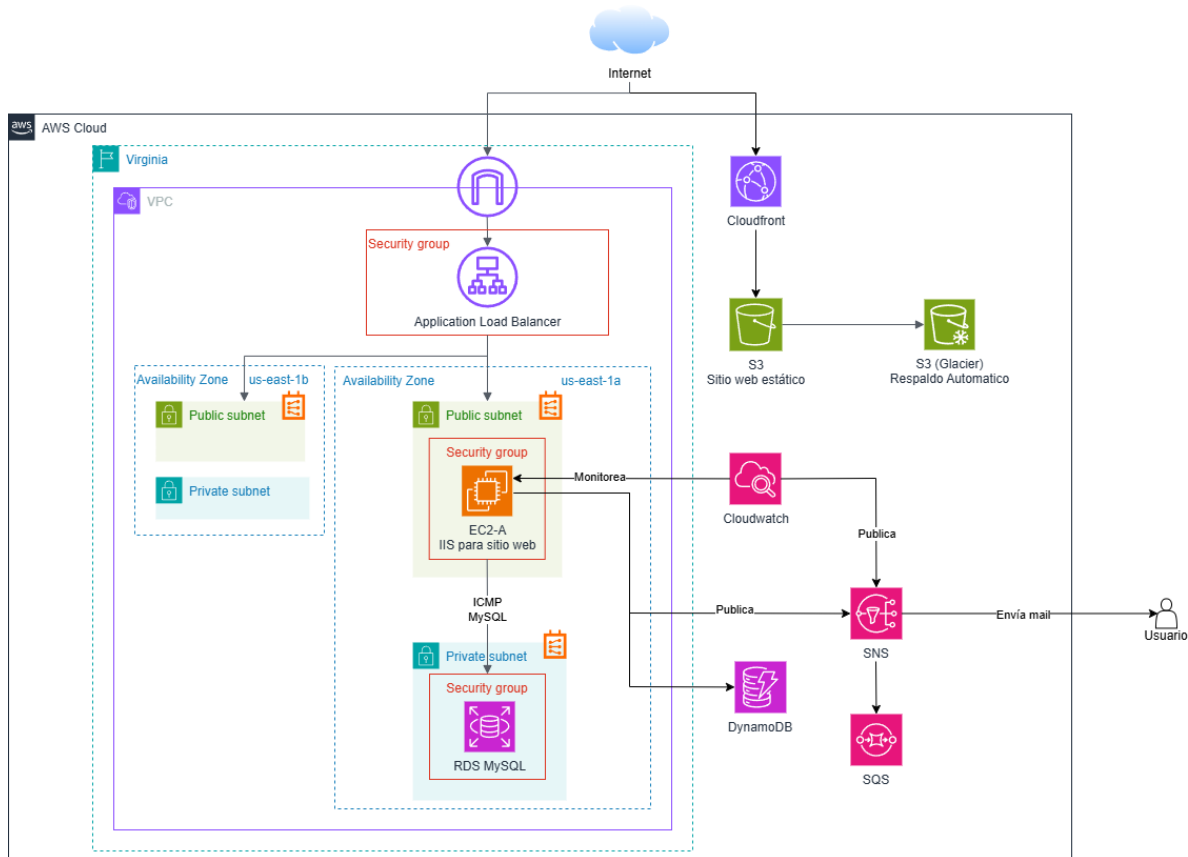


## Resumen del proyecto

La solución Infraestructura Viva fue diseñada con el objetivo de migrar un entorno on-premise a la nube, modernizando la infraestructura de la empresa "Soluciones Digitales ACME". Se implementó una arquitectura escalable, segura y resiliente, utilizando servicios dentro del nivel gratuito de AWS y laboratorios de AWS Academy.



### 1. Almacenamiento de archivos estáticos: Amazon S3 + CloudFront

- **S3** aloja una página web estática, permitiendo un almacenamiento de alta disponibilidad y durabilidad para archivos de uso frecuente.
- Se implementó una **regla de replicación hacia otro bucket con clase de almacenamiento Glacier**, permitiendo cumplir con políticas de archivado y respaldo de datos a largo plazo.
- **CloudFront** se utiliza como CDN para distribuir los contenidos del sitio web de forma global, mejorando tiempos de carga y reduciendo la latencia, además de añadir una capa adicional de seguridad.

Justificación: Uso ideal para sitios estáticos, separación entre datos activos y archivados, y entrega rápida de contenido con bajo costo.

## 2. Cómputo: Amazon EC2 con IIS

- Se desplegó una **instancia EC2 con Windows Server e IIS**, alojando una aplicación web dinámica.
- La instancia reside en una **subred pública dentro de una VPC**, protegida por grupos de seguridad.
- Se utilizó **Apache Benchmark** para realizar pruebas de carga y evaluar el rendimiento del servidor bajo concurrencia.

Justificación: Permite el control total del entorno de ejecución, ideal para entornos web tradicionales. Compatible con herramientas como SQLTools y fácil de monitorear con CloudWatch

## 3. Bases de datos: Amazon RDS y DynamoDB

- **Amazon RDS (MySQL)** fue implementado en una **subred privada**, permitiendo conectividad solo desde la instancia EC2, reforzando la seguridad de los datos relacionales.
- Se configuró un **plan de backup automático** para protección ante fallos.
- **Amazon DynamoDB** fue definido teóricamente como base NoSQL para futuras integraciones, simulando operaciones CRUD mediante el SDK boto3 en Python.

Justificación: Se cumple con la integración de base de datos relacional y NoSQL, sin dependencia de hardware, asegurando escalabilidad y facilidad de mantenimiento.

## 4. Red: Amazon VPC

- Se creó una **VPC personalizada** con subredes públicas y privadas distribuidas en **dos zonas de disponibilidad** (us-east-1a y us-east-1b), cumpliendo requisitos de alta disponibilidad.
- **Tablas de enrutamiento** separan tráfico interno y externo.
- Un **Application Load Balancer (ALB)** enruta tráfico a la instancia EC2 y mejora la tolerancia a fallos.
- **Internet Gateway** permite acceso externo desde las subredes públicas.

Justificación: Se establece un entorno aislado, seguro y escalable para los servicios, con alta disponibilidad y control de tráfico.

## 5. Notificaciones y reintentos: SNS + SQS + DLQ

- Se creó un tópico SNS **“Notificaciones-Inventario”** al que la instancia EC2 puede publicar mensajes.
- **Subscripciones por correo electrónico** permiten recibir alertas en tiempo real.

- SNS también enruta mensajes a una **cola SQS**, permitiendo desacoplar el procesamiento de eventos.
- Se configuró una **Dead Letter Queue (DLQ)** para manejar errores y aplicar una **política de reintentos** automática.

✓ Justificación: Implementa comunicación asíncrona, resiliente y desacoplada, ideal para manejar eventos con tolerancia a fallos.

## 6. Monitoreo y alertas: Amazon CloudWatch

- Se monitorean tres métricas clave de la instancia EC2:
  - **CPUUtilization**: Uso del procesador.
  - **NetworkIn**: Tráfico de entrada a la red.
  - **StatusCheckFailed**: Fallos detectados en la instancia.
- Se definieron dos alarmas:
  - **Fallo de instancia**: reinicia la instancia ante fallos.
  - **Alarma de CPU**: alerta si la CPU supera el 70% de uso.
- Ambas alarmas publican mensajes al tópico SNS **“Notificaciones-Métricas”**.

Justificación: Permite actuar proactivamente ante incidentes, mejorar la observabilidad y automatizar respuestas ante fallos.

## Guía de Buenas Prácticas

### Seguridad

- Se implementaron subredes privadas para los recursos sensibles como la base de datos RDS, asegurando que no sean accesibles directamente desde internet.
- Las instancias EC2 y otros servicios están protegidos mediante grupos de seguridad que permiten únicamente el tráfico necesario (por ejemplo, acceso restringido a puertos específicos desde direcciones IP conocidas).
- El acceso a servicios como RDS se realiza únicamente desde la EC2 en la misma red privada, evitando exposición innecesaria.

### Escalabilidad

- La arquitectura contempla el uso de un Application Load Balancer (ALB), que permite distribuir el tráfico entrante entre múltiples instancias EC2 en caso de que se necesite escalar horizontalmente.

- El almacenamiento en S3 se utiliza para servir archivos estáticos, aprovechando su capacidad elástica y replicación automática.

### **Administración de Red**

- Se utilizó una VPC bien segmentada, con subredes públicas y privadas, lo que permite aislar componentes internos y ofrecer servicios públicos de forma segura.
- Cada subred cuenta con su propia tabla de ruteo configurada para controlar el tráfico de red interno y externo.
- La conexión a internet se gestiona a través de un Internet Gateway para las subredes públicas, manteniendo aislados los recursos privados.