

# Diseño Teórico de una Arquitectura Web Segura y Escalable en AWS

## Introducción

Este documento presenta una propuesta de solución para diseñar una arquitectura web segura y escalable utilizando servicios de AWS. Incluye la configuración de infraestructura en la nube, políticas de IAM, estrategias de gestión de recursos y una implementación teórica. Este diseño busca asegurar alta disponibilidad, escalabilidad y cumplimiento de principios de seguridad.

## Objetivos

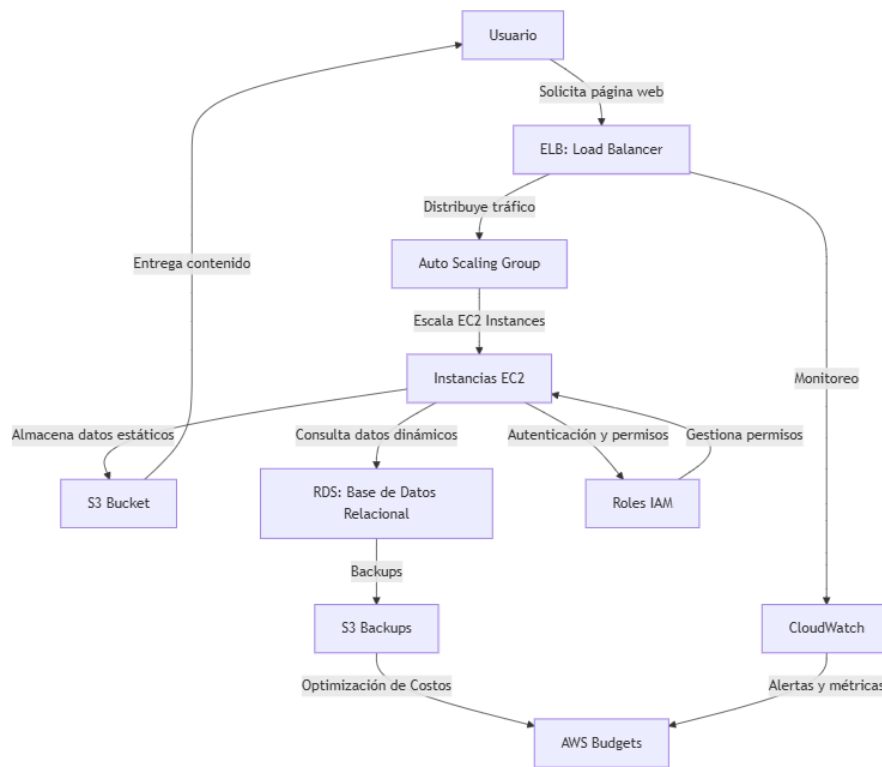
1. Diseñar una arquitectura en la nube utilizando servicios principales de AWS como EC2, S3 y VPC.
2. Configurar roles y políticas de IAM que cumplan con el principio de menor privilegio.
3. Desarrollar una estrategia de gestión de recursos que optimice rendimiento y costos.

## Parte 1: Diseño de la Infraestructura en la Nube

Para una aplicación web escalable, la infraestructura se diseñará con los siguientes componentes:

- **AWS EC2**: Servidores virtuales para alojar la aplicación web.
- **AWS S3**: Almacenamiento de objetos para contenido estático (imágenes, archivos CSS/JS).
- **AWS VPC**: Red privada virtual para aislar recursos y definir subredes públicas y privadas.
- **RDS (opcional)**: Base de datos relacional para la persistencia de datos.

*Diagrama arquitectónico propuesto:*



## Parte 2: Configuración de IAM

### 1. \*\*Roles de IAM\*\*:

- Rol para desarrolladores: Acceso limitado a S3 y CloudWatch.
- Rol para administradores: Acceso completo a todos los servicios necesarios.
- Rol para servidores de aplicación: Acceso restringido a S3 para recuperación de contenido estático.

### 2. \*\*Políticas de IAM\*\*:

- Política de menor privilegio aplicada a cada rol para cumplir con los principios de seguridad.
- Uso de etiquetas para organizar y gestionar permisos según el entorno (desarrollo, producción).

## Parte 3: Estrategia de Gestión de Recursos

Para garantizar la escalabilidad y el control de costos:

- **\*\*Auto Scaling\*\***: Configurar reglas para ajustar automáticamente el número de instancias EC2 según la demanda.
- **\*\*Elastic Load Balancing (ELB)\*\***: Para distribuir el tráfico de manera uniforme entre las instancias EC2.
- **\*\*AWS Budgets\*\***: Definir alertas de costos para supervisar y optimizar el uso de recursos.

## Parte 4: Implementación Teórica

En esta arquitectura teórica:

- Las solicitudes del usuario son manejadas por el ELB y dirigidas a las instancias EC2.
- El contenido estático es servido desde S3, reduciendo la carga en los servidores de aplicación.
- La VPC asegura el aislamiento de la red, con subredes privadas para mayor seguridad.
- CloudWatch monitorea métricas clave y proporciona alertas para garantizar la estabilidad.

## Parte 5: Discusión y Evaluación

1. **\*\*Elección de Servicios\*\***:

- EC2 y S3 son ideales para manejar aplicaciones web dinámicas con contenido estático.
- La VPC mejora la seguridad al segregar los recursos.

2. **\*\*IAM y Seguridad\*\***:

- Las políticas de IAM aseguran que solo los usuarios y servicios autorizados tengan acceso.

3. **\*\*Gestión de Recursos\*\***:

- Auto Scaling y ELB garantizan alta disponibilidad.
- AWS Budgets mantiene los costos bajo control, haciendo el diseño sostenible.