

# Reto Técnico Ingeniero Cloud

## Preguntas teóricas:

1. ¿Cuál es la diferencia entre nube pública, privada e híbrida?

Pública: Ideal para organizaciones que buscan escalabilidad y costos eficientes, con menor control sobre los recursos.

Privada: Adecuada para organizaciones que necesitan mayor control y seguridad, dispuestas a asumir mayores costos.

2. Describa tres prácticas de seguridad en la nube.

- ✓ Gestión de Identidades y Accesos (IAM): definir roles y accesos de acuerdo con las políticas de la compañía.
- ✓ Cifrado de Datos: Cifrar en reposo, en tránsito y gestión de claves.
- ✓ Monitoreo y Registro Continuo: El monitoreo continuo permite detectar incidentes de seguridad de cualquier nivel de riesgo.

3. ¿Qué es la IaC, y cuáles son sus principales beneficios?, mencione 2 herramientas de IaC y sus principales características.

El enfoque es poder crear infraestructura desde scripts o archivos de configuración que describan la configuración, estos archivos se pueden versionar, compartir y reutilizar, lo que facilita la gestión de la infraestructura que conlleva a menos carga laboral. Principales beneficios son: Automatización, Escalabilidad, Auditoría y Control.

## **Herramientas:**

- ✓ **Administrador de recursos de Azure:** su característica principal es que permite crear, actualizar y eliminar recursos en su suscripción de Azure.
- ✓ **Ansible:** su característica es que utiliza lenguaje simple y legible para humanos, utiliza un mecanismo basado en push para implementar configuraciones y orquestar implementaciones complejas.

4. ¿Qué métricas considera esenciales para el monitoreo de soluciones en la nube?

Eventos sobre modificación en la infraestructura sobre los despliegues o cambios de réplicas.

Playbooks de Ansible.

5. ¿Qué es Docker y cuáles son sus componentes principales?

Docker es una plataforma revolucionaria de código abierto que está cambiando la forma en que construimos, desplegamos y gestionamos el software. La tecnología de contenedores Docker permite a los desarrolladores empaquetar aplicaciones en unidades estandarizadas para un despliegue sin problemas.

Principales características: Docker Engine, Docker Image, Docker Container, Docker Hub, Docker Compose y Docker Swarm, trabajan en conjunto para ofrecer un entorno de contenedorización robusto y flexible.

## 6. Caso práctico

### Elijo trabajar con los servicios de AWS por las siguientes razones:

- ✓ **Amplitud de Servicios:** AWS ofrece una amplia gama de servicios que cubren todas las necesidades de la arquitectura nativa en la nube.
- ✓ **Escalabilidad:** AWS proporciona escalabilidad automática a través de servicios como Auto Scaling y Elastic Load Balancing.
- ✓ **Seguridad:** AWS tiene características de seguridad avanzadas y cumple con muchos estándares de seguridad.
- ✓ **Ecosistema:** La integración con otros servicios es sencilla y potente, permitiendo una infraestructura cohesiva y eficaz.

### Arquitectura de la aplicación:

#### 1. Frontend:

- ✓ **Amazon CloudFront:** Para la entrega de contenido con baja latencia y alta velocidad.
- ✓ **Amazon S3:** Para alojar la aplicación web estática.

#### 2. Backend:

- ✓ **Amazon API Gateway:** Para gestionar las solicitudes HTTP entre el frontend y los servicios backend.
- ✓ **AWS Lambda:** Para ejecutar funciones sin servidor que gestionan la lógica del negocio.
- ✓ **Amazon ECS (Elastic Container Service):** Para ejecutar y escalar contenedores Docker en la nube.

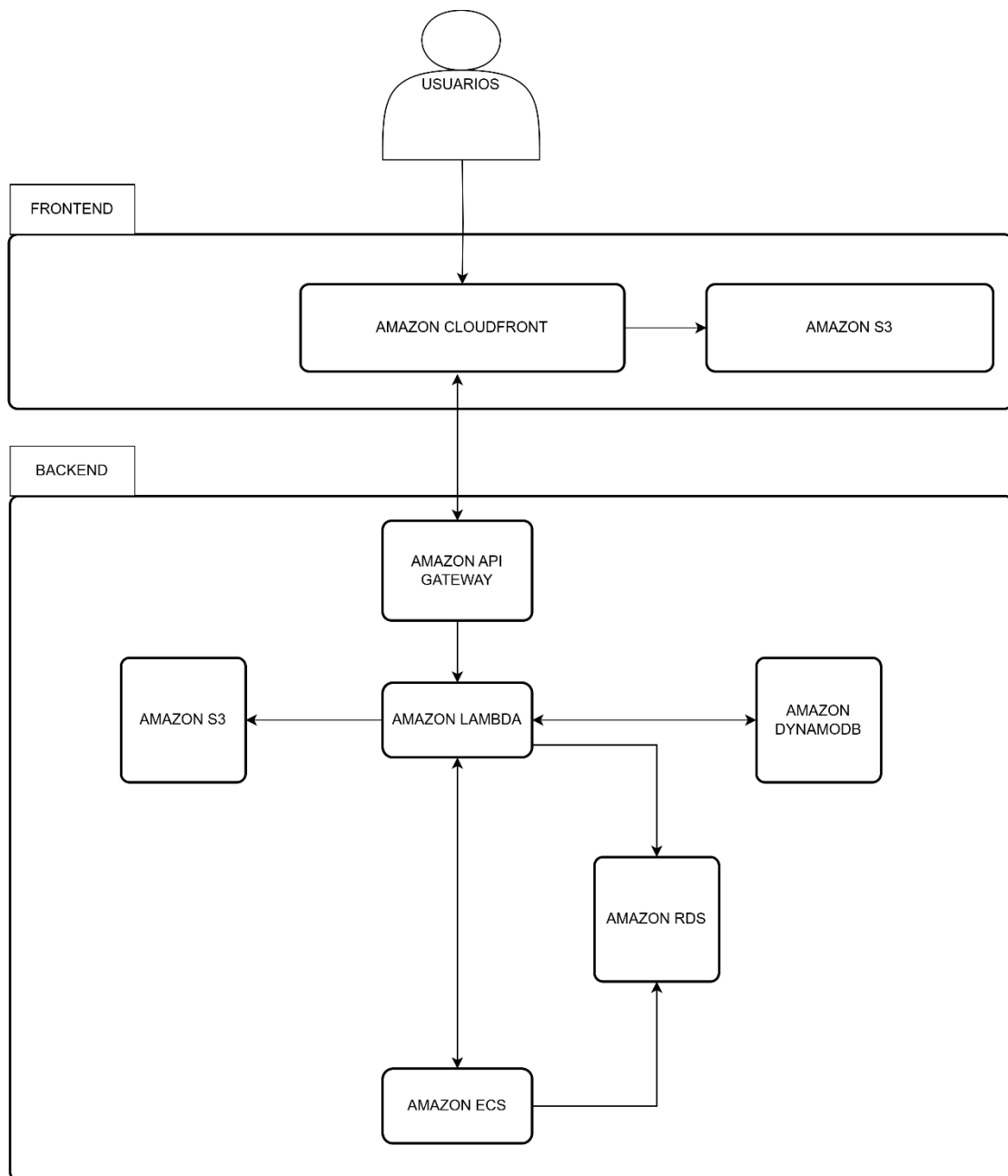
#### 3. Base de Datos:

- ✓ **Amazon RDS (Relational Database Service):** Para gestionar una base de datos relacional (por ejemplo, PostgreSQL).
- ✓ **Amazon DynamoDB:** Para una base de datos NoSQL altamente escalable.

#### 4. Almacenamiento de Objetos:

- ✓ **Amazon S3:** Para almacenar imágenes y contenido estático.

## Diseño de la arquitectura:



## Justificación del Diseño

- ✓ **Escalabilidad:** Los servicios seleccionados permiten escalar tanto horizontalmente (aumentar el número de instancias) como verticalmente (mejorar la capacidad de las instancias).
- ✓ **Costo-Eficiencia:** Utilizar servicios sin servidor como AWS Lambda reduce costos operativos al pagar solo por el uso real.
- ✓ **Flexibilidad:** La combinación de bases de datos relacionales y NoSQL proporciona flexibilidad en el manejo de diferentes tipos de datos.

- ✓ **Desempeño:** Servicios como CloudFront y S3 aseguran una rápida entrega de contenido.
- ✓ **Seguridad:** AWS proporciona servicios robustos y configurables para mantener la seguridad de los datos.

**Ruta GitHub:**

**<https://github.com/ciberguardian86/retotecnico>**