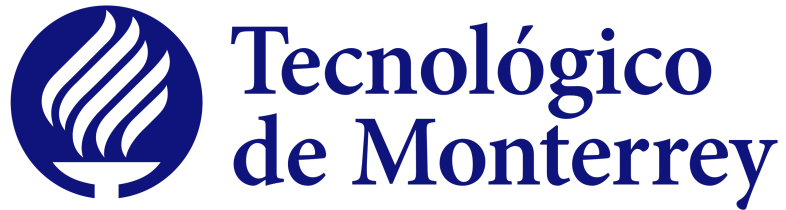


Instituto Tecnológico y de Estudios Superiores de Monterrey  
Campus Monterrey



Inteligencia artificial avanzada para la ciencia de datos II  
TC3007C, Grupo 101

Nombre del profesor: Félix Ricardo Botello Urrutia

## Cloud computing | Actividad 5 - Alta disponibilidad

Equipo 6 | Integrantes:

Oskar Arturo Gamboa Reyes	A01173648
Ricardo Salinas	A01284657
Oscar Gutiérrez Araiza	A00832992
Erika Martínez	A01028621
Marcelo de Luna	A00832239

Octubre 2024

Escenario:

Supongamos que somos una empresa de comercio electrónico que ha decidido migrar su plataforma a la nube para garantizar alta disponibilidad.

A. Diseñar una arquitectura de alta disponibilidad utilizando Azure, AWS o GCP.

1. Zona DNS y Distribución de Tráfico:

Amazon Route 53 (DNS Failover)

2. Balanceo de Carga y Autoescalado:

Elastic Load Balancer (ELB)

Amazon EC2 con Auto Scaling

3. Capa de Bases de Datos:

Amazon RDS (Multi-AZ) o Aurora con Global Database

4. Almacenamiento de Archivos:

Amazon S3

5. Distribución de Contenido:

Amazon CloudFront (CDN)

6. Red y Subnetting:

Amazon VPC con subnets en múltiples AZs

7. Monitorización y Backup:

Amazon CloudWatch

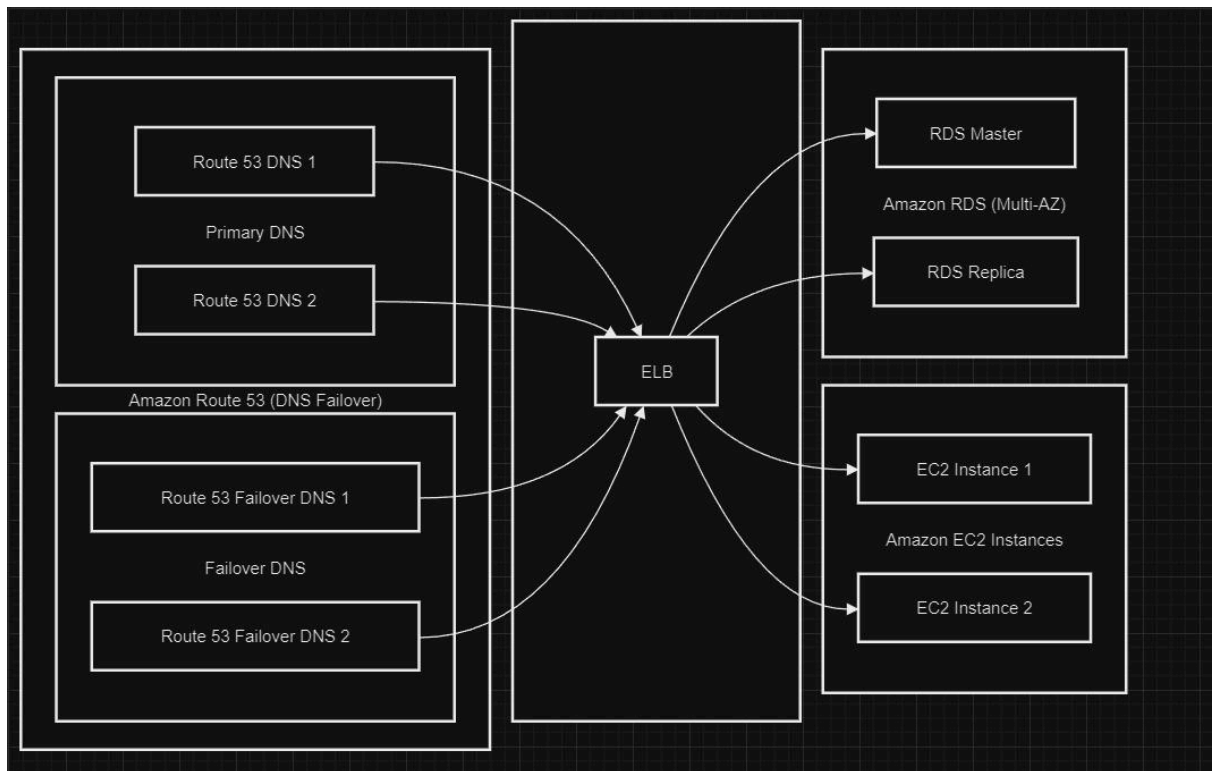
AWS Backup y Amazon S3 Glacier

B. Identificar los servicios o herramientas clave que se utilizarían para lograr la alta disponibilidad.

### **Componentes clave para garantizar la alta disponibilidad en AWS:**

- **Amazon Route 53 (DNS Failover)**
- **Elastic Load Balancer (ELB)**
- **Amazon EC2 (con Auto Scaling)**
- **Amazon RDS (Multi-AZ) o Amazon Aurora (con Global Database)**
- **Amazon S3 (con replicación entre regiones)**
- **Amazon CloudFront (CDN)**
- **Amazon VPC (Virtual Private Cloud) con múltiples subnets en diferentes AZs**
- **Amazon CloudWatch**
- **AWS Backup y Amazon S3 Glacier**
- **Auto Scaling Groups**

C. Presentar un diagrama de la arquitectura propuesta.



D. Explicar cómo cada componente contribuye a la alta disponibilidad.

### Explicación de los componentes:

- **Amazon Route 53 (DNS Failover):** Actúa como un servicio de DNS global que distribuye el tráfico entre múltiples regiones y realiza el failover automáticamente si una región no está disponible.
- **Elastic Load Balancer (ELB):** Balancea la carga entre varias instancias de EC2 distribuidas en diferentes zonas de disponibilidad (AZs) dentro de una misma región, asegurando que si una instancia falla, las otras sigan sirviendo el tráfico.
- **Amazon EC2 con Auto Scaling:** Las instancias se despliegan en diferentes AZs, y el grupo de autoescalado ajusta automáticamente el número de instancias según la carga, garantizando capacidad y disponibilidad continua.
- **Amazon RDS (Multi-AZ) o Aurora con Global Database:** Las bases de datos distribuidas aseguran que los datos estén siempre disponibles, incluso si una región falla, gracias a la replicación entre AZs o entre regiones.
- **Amazon S3 con replicación entre regiones:** Garantiza que los archivos críticos estén replicados y accesibles desde múltiples ubicaciones en caso de que una región no esté disponible.

- **Amazon CloudFront (CDN):** Distribuye contenido estático a usuarios cercanos geográficamente, reduciendo la latencia y mejorando la experiencia de usuario.
- **Amazon VPC con múltiples subnets:** Las instancias se despliegan en diferentes subnets y AZs, asegurando redundancia a nivel de red y mayor resiliencia.
- **Amazon CloudWatch:** Proporciona monitorización y alertas en tiempo real sobre la salud de la infraestructura, ayudando a detectar problemas antes de que afecten la disponibilidad.
- **AWS Backup y Amazon S3 Glacier:** Garantizan copias de seguridad automáticas y almacenamiento de datos a largo plazo, facilitando la recuperación en caso de desastres.

E. Identificar posibles desafíos y soluciones.

### Desafíos y Soluciones:

1. **Latencia entre regiones:** La replicación de datos entre regiones puede generar latencia. **Solución:** Usar Amazon CloudFront para distribuir contenido estático y estrategias de replicación optimizadas para minimizar los tiempos de respuesta.
2. **Costos elevados:** Mantener servicios en alta disponibilidad y multi-AZ aumenta los costos. **Solución:** Implementar políticas de autoescalado para ajustar dinámicamente los recursos según la demanda.
3. **Gestión de bases de datos distribuidas:** Asegurar la consistencia de los datos puede ser complejo cuando se replica entre regiones. **Solución:** Usar Amazon Aurora Global Database que gestiona automáticamente la replicación y la consistencia.