Arquitectura Funcional de alta disponibilidad en AWS

1. Descripción General

La solución se diseñó bajo los principios de alta disponibilidad, modularidad y automatización. Cada capa —red, aplicación, datos e integración— está separada lógicamente para facilitar despliegues independientes y futuras ampliaciones. La plantilla de CloudFormation parametriza las variables críticas, garantizando reproducibilidad y consistencia entre entornos.

2. Componentes Principales

La base de la arquitectura es una VPC con cuatro subredes: dos públicas y dos privadas, distribuidas en dos zonas de disponibilidad.

- Subredes públicas: alojan los Application Load Balancers (ALB) y el NAT Gateway.
- Subredes privadas: contienen instancias backend y la base de datos RDS.
- Internet Gateway (IGW) y rutas: permiten salida a Internet sólo para servicios públicos.
- NAT Gateway: habilita actualizaciones del sistema operativo en instancias privadas sin exponerlas públicamente.

Este diseño garantiza aislamiento y resiliencia ante fallos de zona.

2.1 Red (VPC y Subredes)

- **VPC** con bloque CIDR 10.0.0.0/16.
- **Subredes públicas**: 2 (una por AZ) para ALBs y NAT Gateway.
- **Subredes privadas**: 2 (una por AZ) para instancias de backend y base de datos.
- **Internet Gateway** y **Tabla de Rutas** para salida pública.
- **NAT Gateway** para permitir salida a Internet de las instancias privadas sin hacerlas públicas.

2.2 Seguridad

- **Grupos de Seguridad** diseñados por capa:
    - Frontend ALB: permite tráfico HTTP (puerto 80) desde Internet.
    - Backend ALB: acceso restringido al grupo del frontend.
    - Backend EC2: acceso solo desde el ALB del backend.
    - RDS: acceso únicamente desde el backend EC2.

2.3 Balanceadores de Carga (ALB)

- **Frontend ALB**: Público. Distribuye tráfico web hacia instancias frontend.
- **Backend ALB**: Interno. Distribuye tráfico desde frontend hacia backend.
- Cada ALB tiene:
    - Listener en el puerto 80.
    - Target Group asociado a instancias EC2 por capa.

2.4 Auto Scaling y Plantillas

- Se emplea **Launch Template** con perfil IAM LabRole.
- **Auto Scaling Groups (ASG)** para frontend y backend:
    - Capacidad deseada: 2
    - Mínimo: 2, Máximo: 3
    - Tags incluidos para identificación (Name: FrontendInstance, Name: BackendInstance).

2.5 Base de Datos

- **RDS MySQL 8.0.34**, con:
    - Clase db.t3.micro, almacenamiento 20 GiB.
    - Subnet Group con subredes privadas.
    - Sin backups, sin Multi-AZ.
    - Sin cifrado.

2.6 Almacenamiento en S3 + Lambda

- **Bucket S3** para almacenamiento de archivos.
- **Función Lambda** en Node.js 18 que recibe eventos de ObjectCreated desde S3.
- **Permisos Lambda** configurados para permitir invocación desde S3.

2.7 IAM y Seguridad

- Se reutiliza el rol existente LabRole para:
    - Instancias EC2 (vía Instance Profile).
    - Función Lambda.

2.8 Deletion Policy

- Todos los recursos incluyen DeletionPolicy: Delete para permitir borrado limpio del entorno desde CloudFormation.

3. Integración y Automatización

Para procesar objetos de S3 se ha definido una función Lambda que se dispara con cada evento de creación de objeto:

- Bucket de datos: con notificación configurada hacia Lambda.
- LambdaPermission: permite a S3 invocar la función.
- La función imprime el evento y retorna un status 200.

La infraestructura está completamente definida como código, lo que facilita pipelines de CI/CD.

4. Consideraciones y Exclusiones

- No se utiliza Cognito, CloudFront, SES, API Gateway ni Route 53.
- No se aplican configuraciones de cifrado ni backup.
- No se crean roles nuevos; se reutiliza únicamente LabRole.
- Toda la arquitectura es desplegable mediante un único archivo CloudFormation empaquetado.

5. Diagrama Arquitectónico

Se adjunta arquitectura en draw.io

6. Argumentación y Conclusiones

Este diseño ofrece una solución funcional modular con buenas prácticas de segmentación por capas, control de acceso mediante Security Groups y componentes reutilizables como Launch Templates y ALBs. Es ideal como base para ambientes de desarrollo o pruebas.

1. Modularidad con CloudFormation
   - Facilita despliegues repetibles y versionados.
   - Permite aislar fallos y escalar componentes independientemente.
2. Uso de ALB en dos capas
   - ALB público para exponer la aplicación.
   - ALB interno para aislar la lógica de negocio de la Internet pública.
3. Lambda + S3
   - Lambda es serverless, ideal para cargas esporádicas y procesamiento ligero.
4. RDS MySQL en pruebas
   - Minimiza costes y complejidad; para producción, se escalaría a Aurora o Multi-AZ.