Prueba Técnica – Devsu

Link Repositório GitHub:

https://github.com/David-KtL/cloud-exercise

Preguntas Teóricas

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

- **Nube Pública:** Es un servicio ofrecido por terceros, accesible a través de internet y compartido entre múltiples clientes. Ejemplos incluyen AWS, Azure y Google Cloud. Como ventajas tiene escalabilidad, pago por uso y mantenimiento gestionado por el proveedor.
- **Nube Privada:** Infraestructura dedicada exclusivamente a una organización, ya sea on-premise o alojada por un tercero. Brinda mayor control, personalización y seguridad, pero implica costos operativos más altos.
- **Nube Híbrida:** Combinación de nube pública y privada, permitiendo la interoperabilidad entre ambas. Ofrece flexibilidad al aprovechar la escalabilidad de la nube pública mientras se mantienen datos sensibles en la nube privada.

2. Describa tres prácticas de seguridad en la nube

Para garantizar la seguridad en entornos cloud, se deben seguir prácticas clave:

1. **Gestión de identidades y accesos (IAM):** Implementar roles y políticas de acceso basadas en el principio de privilegio mínimo.
2. **Cifrado de datos:** Encriptar datos en tránsito y en reposo utilizando estándares como AES-256 y TLS 1.2+.
3. **Monitoreo y auditoría:** Configurar logs con AWS CloudTrail y Amazon GuardDuty para detectar actividades sospechosas y responder a incidentes.

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

Infraestructura como Código (IaC) es una metodología que permite gestionar y aprovisionar infraestructura mediante archivos de configuración en lugar de procesos manuales. Beneficios clave:

- **Automatización y consistencia:** Reduce errores humanos y asegura entornos idénticos.
- **Escalabilidad y rapidez:** Permite la creación y modificación rápida de infraestructura.
- **Versionado y control:** Se pueden rastrear cambios y revertirlos si es necesario.

Herramientas de IaC:

1. **Terraform:** Declarativo, soporta múltiples proveedores, permite reutilización con módulos y mantiene estado en archivos de backend.
2. **Ansible:** Basado en YAML, utiliza SSH para la configuración sin necesidad de agentes, permite automatizar tareas de configuración y despliegue en múltiples servidores.

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

El monitoreo es clave para la operación eficiente de aplicaciones en la nube, ya que se puede detectar incidentes, detectar accesos no autorizados, etc. Algunas de las metricas mas importantes son:

- **Disponibilidad y latencia:** Tiempos de respuesta y uptime del sistema.
- **Uso de CPU y memoria:** Para evitar sobrecarga y optimizar costos.
- **Tasa de errores:** Identificación de fallos en servicios y aplicaciones.
- **Tasa de solicitudes y tráfico:** Análisis de la demanda y detección de patrones de uso.
- **Seguridad:** Registros de acceso, intentos de autenticación fallidos y cambios en configuraciones críticas.

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

Docker es una plataforma que permite empaquetar aplicaciones en contenedores ligeros y portátiles. Sus componentes principales incluyen:

- **Docker Engine:** Servicio que ejecuta contenedores.
- **Dockerfile:** Archivo que define la imagen base y configuraciones del contenedor.
- **Docker Image:** Plantilla inmutable usada para crear contenedores.
- **Docker Container:** Instancia ejecutable de una imagen.
- **Docker Registry:** Almacén de imágenes como Docker Hub o Amazon ECR.

6. Caso Práctico: Diseño de Arquitectura para Aplicación Nativa de Nube

Requisitos

La aplicación debe considerar los siguientes componentes:

- **Frontend**: Una aplicación web que los clientes utilizarán para navegación.
- **Backend**: Servicios que se comunican con la base de datos y el frontend.
- **Base de datos**: Un sistema de gestión de base de datos que almacene información.
- **Almacenamiento de objetos**: Para gestionar imágenes y contenido estático.

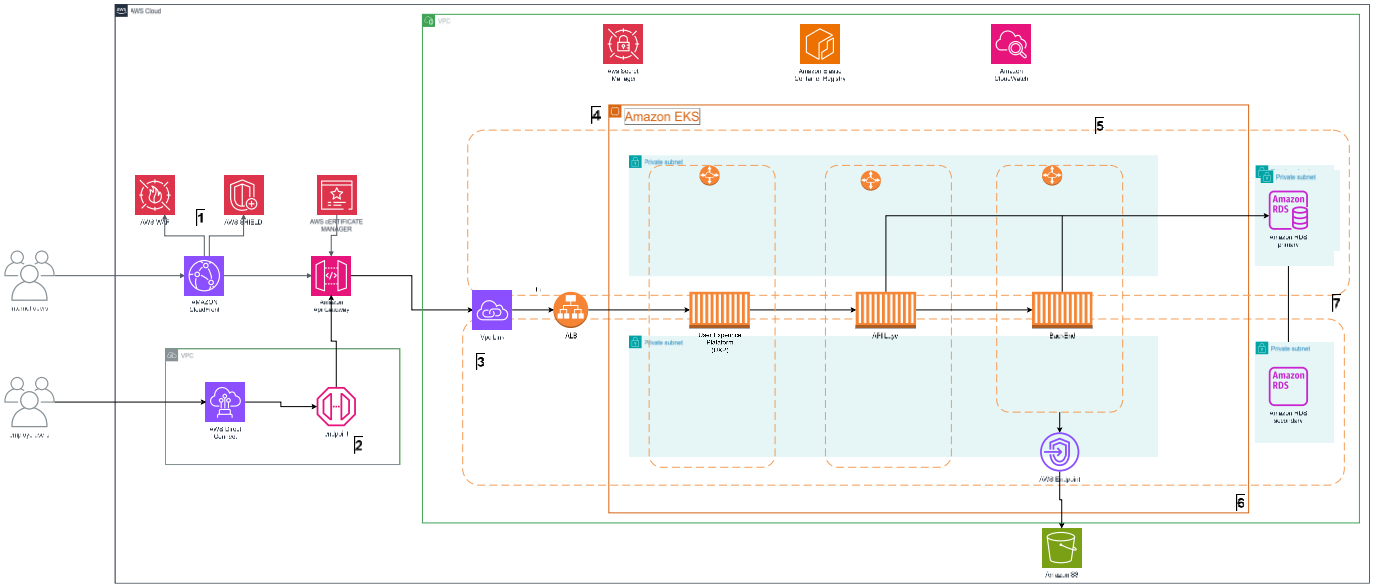
Objetivos del Diseño

1. *Seleccionar un proveedor de servicios de nube (AWS, Azure o GCP) y sustentar la selección.*

Justificación de Elección de AWS frente a Azure y GCP

En base el tipo de aplicación que se requiera se puede ir por cualquier cloud mencionado. Para este caso AWS representa la opción elegida para esta arquitectura en la nube debido a su enfoque específico en servicios modulares que facilitan la implementación de arquitecturas de microservicios.

2. Diseñar una arquitectura de nube.
3. Incluir diagramas que representen la arquitectura.



https://drive.google.com/file/d/1U6MCyyOOdXhiqva7hD1rU1oJGVoDrzu/view?usp=sharing

4. Justificar las decisiones de diseño.

La combinación de Amazon S3 para el almacenamiento de objetos estáticos del frontend con posibilidad de integración directa con CloudFront para distribución global, junto con la flexibilidad de Amazon EKS para la orquestación de contenedores del backend que permite patrones de despliegue avanzados como blue-green y canary, sumado a la capacidad de Amazon RDS para implementar bases de datos con alta disponibilidad mediante réplicas de lectura entre zonas con latencia mínima, conforman un ecosistema técnicamente superior en términos de integración nativa y rendimiento.

Justificación Técnica del Diseño de Arquitectura

Esta arquitectura implementa un diseño resiliente para aplicaciones nativas en la nube utilizando los servicios de AWS de manera estratégica:

1. La implementación de Amazon CloudFront como CDN global junto con AWS WAF y AWS Shield proporciona una primera capa de defensa crítica que protege la aplicación contra ataques DDoS y vulnerabilidades web comunes, mientras optimiza la entrega de contenido estático reduciendo la latencia para usuarios globales. Este enfoque permite filtrar tráfico malicioso antes de que alcance nuestra infraestructura principal.

2. La arquitectura utiliza Amazon API Gateway con endpoints privados conectados a través de AWS Direct Connect, garantizando que los usuarios corporativos accedan a los servicios mediante conexiones dedicadas de baja latencia sin exposición a internet. Esta configuración mantiene el tráfico sensible dentro de un perímetro controlado, cumpliendo con requisitos de seguridad empresarial.

3. El acceso a la VPC está estrictamente controlado a través del recurso VpcLink de API Gateway, creando un único punto de entrada auditado y administrado. Esta decisión arquitectónica minimiza la superficie de ataque y simplifica la aplicación de políticas de seguridad, permitiendo un control granular sobre qué tráfico puede acceder a los recursos internos.

4. Los contenedores de Amazon EKS se ejecutan sobre una infraestructura híbrida de AWS Fargate y Amazon EC2, proporcionando la flexibilidad de elegir el modelo de implementación óptimo para cada carga de trabajo específica. Fargate ofrece administración sin servidor para microservicios ligeros, mientras que EC2 permite configuraciones personalizadas para componentes que requieren optimización específica.

5. Aunque la arquitectura actual utiliza dos Zonas de Disponibilidad, su diseño modular permite la extensión a tres Zonas, incrementando significativamente la resiliencia ante fallos de infraestructura regionales. Esta capacidad de extensión demuestra un enfoque proactivo hacia la alta disponibilidad empresarial.

6. La implementación de endpoints de VPC para acceder a servicios AWS elimina la necesidad de conectividad a internet, reforzando significativamente la postura de seguridad. Esta configuración reduce la exposición a amenazas externas y mejora el rendimiento de la comunicación entre servicios al mantener el tráfico dentro de la red de AWS.

7. Amazon RDS en configuración Multi-AZ proporciona replicación síncrona de datos entre zonas independientes, con conmutación por error automática que garantiza continuidad operativa incluso ante la pérdida completa de una zona. Esta característica es fundamental para mantener la integridad y disponibilidad de los datos en aplicaciones empresariales críticas..

Esta arquitectura refleja las mejores prácticas de AWS para aplicaciones empresariales, priorizando seguridad, disponibilidad y rendimiento mientras mantiene la flexibilidad necesaria para evolucionar con los requisitos del negocio.