Reto Técnico

Diego Vinicio Chicaiza Herrera

Preguntas

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

- Nube pública: Los recursos informáticos (como servidores, almacenamiento y bases de datos) son provistos y gestionados por un proveedor externo (por ejemplo, AWS, Azure, Google Cloud) y son compartidos entre múltiples clientes. El usuario accede a estos recursos a través de Internet y solo paga por el consumo realizado.
- Nube privada: La infraestructura está dedicada exclusivamente a una sola organización. Puede estar ubicada en las instalaciones del cliente o ser hospedada por un proveedor externo. Brinda mayor control, seguridad y personalización.
- Nube híbrida: Combina la nube pública y la privada, permitiendo que datos y aplicaciones se compartan entre ambas. Esto ofrece mayor flexibilidad, permitiendo, por ejemplo, mantener cargas de trabajo sensibles en la nube privada y aprovechar la nube pública para cargas variables o de alta demanda.

2. Describa tres prácticas de seguridad en la nube.

- Gestión de identidades y accesos (IAM): Establecer controles estrictos para definir quién puede acceder a qué recursos, siguiendo los principios de privilegio mínimo y autenticación multifactor.
- Encriptación de datos: Proteger los datos sensibles cifrándolos tanto en tránsito como en reposo, evitando accesos no autorizados.
- Monitoreo y auditoría de actividades: Implementar sistemas de monitoreo para detectar actividades sospechosas y mantener registros de auditoría de accesos y cambios en la infraestructura.

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

La Infraestructura como Código es la práctica de definir la infraestructura o recursos como redes, servidores, bases de datos, políticas de seguridad, etc. Se utilizan archivos de código para expresar lo que se desea construir. Esto permite gestionar y aprovisionar recursos de manera repetible y automática, eliminando tareas manuales.

- Terraform
  Es multiplataforma y declarativa.
  Permite diseñar, administrar y versionar la infraestructura en distintos proveedores de nube desde un solo archivo de configuración.
- AWS CloudFormation
  Especializada en AWS.
  Utiliza plantillas en JSON o YAML para definir recursos; soporta integración con otros servicios de AWS y automatiza el despliegue completo de la infraestructura.

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

- **Latencia**: Tiempo de respuesta de los servicios o aplicaciones.
- **Disponibilidad**: Porcentaje de tiempo en que el servicio está operativo.
- **Uso de recursos**: Consumo de CPU, memoria, almacenamiento y ancho de banda.
- **Errores y logs**: Tasa de errores, análisis de logs para detectar fallos, intentos de acceso fallidos y patrones inusuales.
- **Tráfico de red**: Monitoreo del tráfico entrante y saliente para identificar posibles cuellos de botella o ataques.

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

Docker es una plataforma de código abierto diseñada para desarrollar, enviar y ejecutar aplicaciones dentro de contenedores. Los contenedores permiten empaquetar una aplicación junto con todas sus dependencias, facilitando su ejecución consistente en diferentes entornos.

Componentes principales de Docker:

- **Docker Engine:** Es el motor principal encargado de crear, administrar y ejecutar los contenedores en el sistema operativo anfitrión.
- **Docker Daemon (dockerd):** Es el proceso que se ejecuta en segundo plano y gestiona los objetos de Docker como imágenes, contenedores, redes y volúmenes. Responde a las solicitudes de la API de Docker.
- **Docker Registry:** Es el sistema de almacenamiento y distribución para imágenes de Docker, permitiendo que los usuarios suban y descarguen imágenes. Ejemplo: Docker Hub, que es un registro público de imágenes.

Caso Práctico

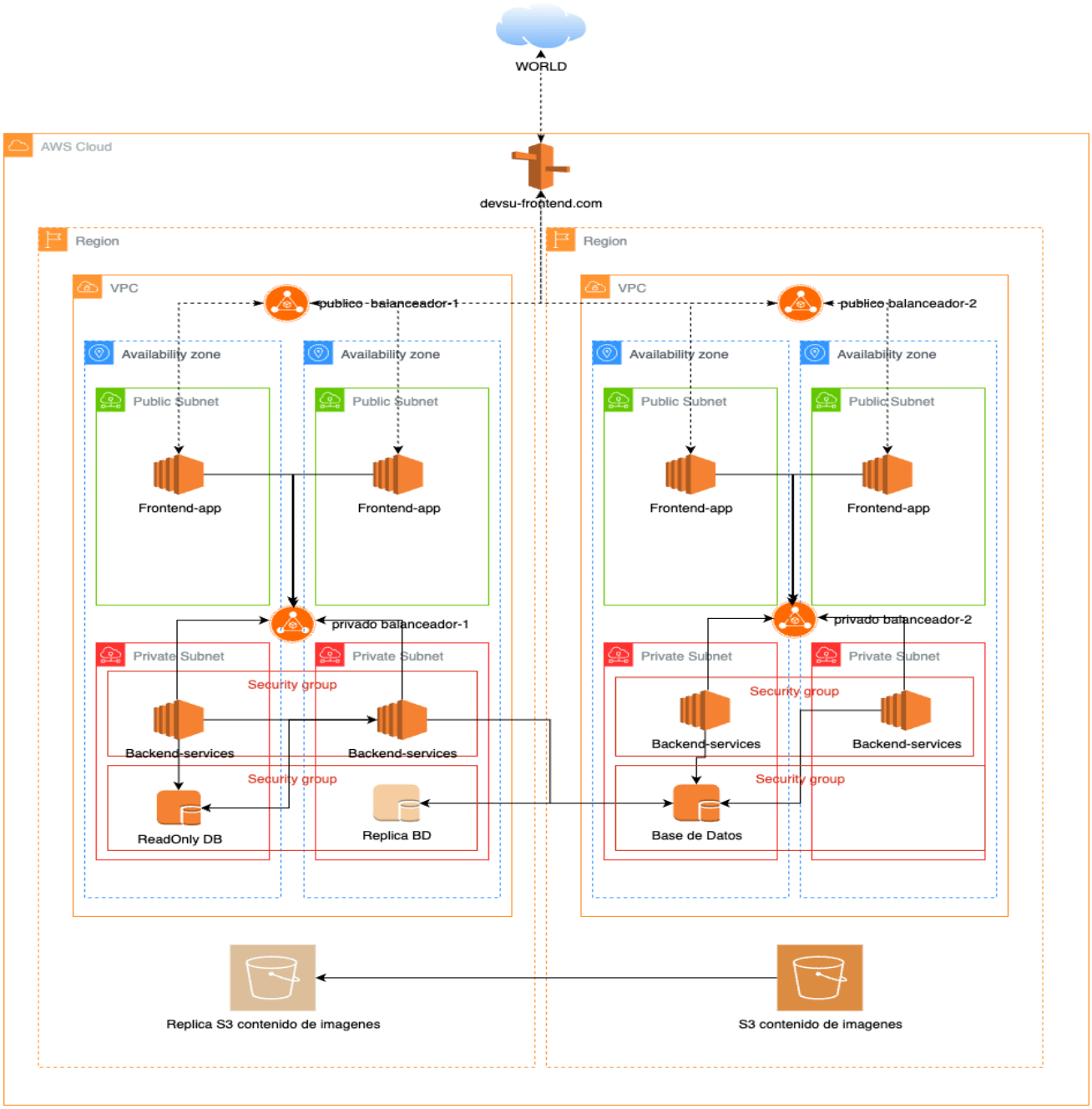
Proveedor de Nube: AWS (Amazon Web Services)

Justificación

He elegido AWS para este caso práctico porque la versatilidad y madurez de sus servicios me permite cumplir con lo propuesto en el ejercicio de forma equilibrada en disponibilidad, costos, rendimiento y seguridad, manteniendo además una arquitectura escalable y fácil de operar. AWS ofrece múltiples opciones para diseñar una solución con alta tolerancia a fallos (por ejemplo, distribuyendo componentes en varias *Availability Zones*), lo que reduce el riesgo de interrupciones y ayuda a mantener los niveles de servicio esperados.
Desde el punto de vista económico, AWS facilita optimizar el gasto mediante modelos de pago por uso y la posibilidad de ajustar recursos según la demanda, evitando sobreaprovisionamiento. A nivel técnico, su ecosistema de servicios administrados permite implementar componentes clave (cómputo, almacenamiento, bases de datos, redes y monitoreo) sin asumir una carga excesiva de mantenimiento, lo que acelera los tiempos de entrega y mejora la estabilidad del entorno.

Finalmente, AWS también permite definir desde el inicio un plan sólido de Disaster Recovery, con estrategias que van desde copias de seguridad y replicación, hasta despliegues en regiones alternativas asegurando continuidad del negocio ante fallos mayores. En conjunto, estas capacidades hacen que AWS sea una opción adecuada para implementar una solución confiable, eficiente y preparada para crecimiento y contingencias.

Propuesta Diagrama Arquitectónico:



Justificación

- Route 53 (routing por geolocalización)

  Lo elegí para dirigir a los usuarios a la región más adecuada según su ubicación,

reduciendo la latencia y mejorando la experiencia. Además aporta alta disponibilidad con DNS administrado y opciones de failover.

- Application Load Balancer (2 regiones)

  Lo elegí para distribuir tráfico HTTP/HTTPS de forma inteligente y soportar alta disponibilidad multi-región.

  1. ALB Internet-facing (público): permite exponer la aplicación a Internet de manera controlada y balancear tráfico hacia el frontend.
  2. ALB Internal (privado): permite aislar servicios internos (solo accesibles dentro de la VPC), mejorando seguridad y separación de capas.

- 4 Availability Zones

  Las elegí para tolerancia a fallos: si una AZ cae, el servicio puede seguir operando en las otras, aumentando resiliencia y disponibilidad.

- 4 subredes públicas + 4 subredes privadas

  Las elegí para segmentar la red por seguridad y buenas prácticas:

  - Públicas: para componentes que deben recibir tráfico externo.

  - Privadas: para bases de datos y los servicios backend internos, evitando exposición directa a Internet.

- Amazon RDS (replicación + 1 nodo Read Replica / read-only)

  Lo elegí para tener una base de datos administrada, con alta disponibilidad y mejor rendimiento en lecturas al descargar consultas al nodo read-only. También mejora la capacidad de recuperación ante incidentes.

- Security Groups en redes privadas

  Los elegí porque funcionan como firewall a nivel de instancia/servicio, permitiendo controlar el tráfico por puertos y orígenes y reducir la superficie de ataque al mantener reglas estrictas en la capa privada.

- Amazon S3 (replicación a otro bucket en otra región)

  Lo elegí por su durabilidad y bajo costo para almacenamiento de objetos, y la replicación cross-region para Disaster Recovery, asegurando continuidad y disponibilidad de datos incluso si una región falla.