

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

- Nube Pública, infraestructura gestionada por un proveedor externo (AWS, Azure, GCP). Entre sus ventajas: escalabilidad ilimitada, costos son pago por uso, sin mantenimiento de hardware.
- Nube Privada, corresponde a la infraestructura y servicios internos gestionados por una empresa. Como desventajas: costos de hardware altos, escalabilidad limitada.
- Nube Híbrida, uso compartido de nubes privadas y privadas, permitiendo flexibilidad teniendo sistemas críticos en nube privada y escalamiento de servicios en nube pública.

2. Describa 3 prácticas de seguridad en nube

- Gestión de Identidad y Accesos (IAM), principio de mínimo privilegio, autenticación multifactor (MFA), roles y grupos
- Cifrado de datos, en tránsito (TLS, HTTPS), en reposo
- Monitoreo, permite garantizar el rendimiento y la disponibilidad de: infraestructura, aplicaciones. Optimizar el uso de recursos para evitar gastos innecesarios

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

Permite gestionar y aprovisionar infraestructura (servidores, redes, base datos) mediante código y scripts, en lugar de configuraciones manuales que pueden llevar a errores manuales de no existir un procedimientos o directrices. Permite desplegar infraestructuras completas con un solo comando.

Entre las herramientas de IaC más usadas tenemos:

- Terraform, es agnóstico puede ser usado entre los proveedores públicos de nube (AWS, Azure, GCP), el estado de la infraestructura es almacenado en archivo '.tfstate'
- AWS CDK, usa lenguajes de programación (typescript, python) para definir la infraestructura. Esta herramienta es exclusiva de AWS. La infraestructura creada queda registrada en AWS CloudFormation

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

- Porcentaje uso CPU, para identificar tiempos prolongados de procesamiento
- Uso RAM, identificar uso de recursos por las aplicaciones
- Latencia, para identificar el tiempos altos / bajos entre la petición y la respuesta

5. Qué es docker y cuáles son sus componentes principales?

Plataforma de virtualización ligera que permite empaquetar, distribuir y ejecutar aplicaciones dentro de contenedores, sus componentes principales:

- Docker engine, se encarga de la gestión de contenedores
- Images, son plantillas inmutables que contienen el sistema base, dependencias y la aplicación
- Contenedores, son instancias en ejecución de las imágenes. Son efímeros
- Volúmenes, mecanismo para persistir datos fuera del ciclo de vida de un contenedor

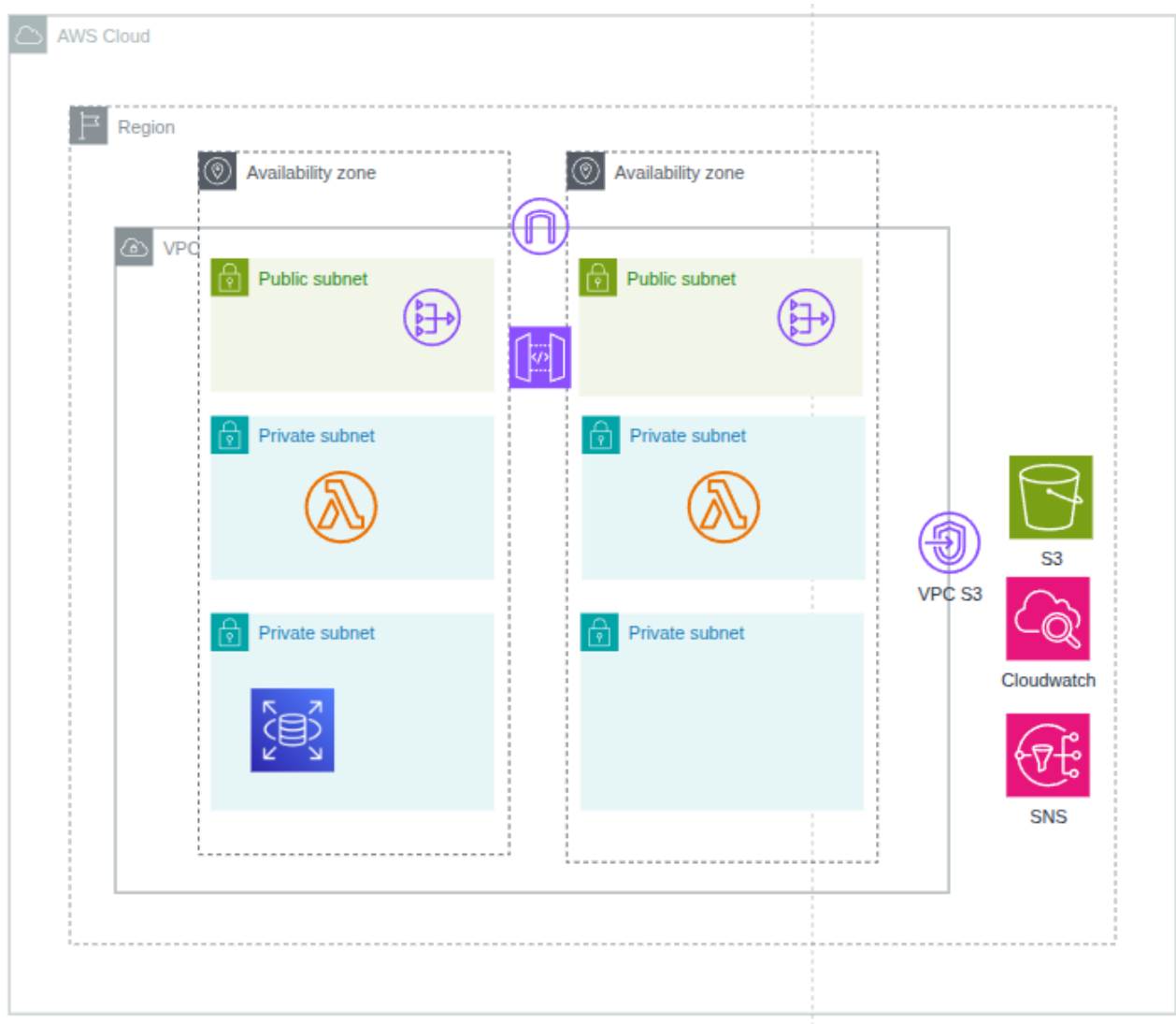
6. Caso práctico: Cree un diseño de arquitectura para una aplicación nativa de nube, considerando los siguientes componentes:

- 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

Diseño

- Seleccione un proveedor de servicios de nube (AWS, Azure o GCP) y sustente su selección
- Diseñe una arquitectura de nube. Incluya diagramas que representen la arquitectura y justifique sus decisiones de diseño.



Para esta arquitectura se eligió AWS como proveedor cloud por los siguientes motivos:

- Proveedor cloud fuertemente posicionado en el mercado
- Modelo de costos eficiente
- Ecosistema serverless maduro
- Servicios gestionados

La arquitectura seleccionada es una de 3 capas: Web, Aplicación y Persistencia.

- Capa Web, es el punto de entrada a la aplicación por parte de los usuarios desde internet.
- Capa Aplicación, lógica de negocio de la aplicaciones. Es privada no tiene acceso directo desde internet. Únicamente recibe peticiones de la capa web.
- Capa Persistencia, para almacenar datos de la aplicación. El acceso exterior a esta capa esta bloqueado, solo procesa solicitudes de capa aplicación.

Asumiendo que se trata de una aplicación en React, para esta arquitectura se ha elegido los siguientes servicios:

Frontend (Capa Web)

- Aplicación React se despliega en un bucket S3 configurado para hosting estático
- API Gateway, para enrutamiento de peticiones hacia el backend, exponiendo endpoints tipo REST

Backend (Capa Aplicación)

- Recibe las peticiones del API Gateway que son procesadas por las lambdas
- Cada lambda representa un caso de uso específico
- Esta capa se encarga de comunicarse con la base datos y gestionar objetos (archivos) en S3
- Para interactuar con S3 se usa un VPC endpoint para tener una comunicación directa

Persistencia

- Para este ejemplo se considera usar Amazon RDS para persistir datos relacionales

Adicionalmente, la arquitectura incluye AWS CloudWatch para la trazabilidad de los logs.