1. Diferencia entre nube pública, privada e híbrida.
    a. Nube Pública: Servicio de nube disponible para todos los usuarios, por ejemplo AWS, Azure, GCP. Los hardware y software son compartidos entre múltiples clientes pero cada quien teniendo independencia de sus recursos utilizados.
    b. Nube Privado: Servicio, cuyos componentes (hardware y software), está disponible exclusivamente para una organización. Proporciona mayor control a la organización, además de aislamiento y seguridad. Puede ser en las instalaciones privadas de la organización o en otro espacio compartido.
    c. Nube Híbrida: Es una combinación de las dos anteriores, permitiendo la comunicación entre recursos de ambas nubes.
2. Describa 3 prácticas de seguridad en la nube.
    a. Configurar la base de dato en una red privada
    b. Utilizar los grupos de seguridad (Security groups) para habilitar solo los puertos necesarios.
    c. Utilizar roles para la comunicación entre servicios de AWS.
3. Que es la IaC, y cuales son sus principales beneficios? Mencione 2 herramientas de IaC y sus principales características.
    a. Infraestructura como código es la herramienta utilizada para codificar la creación y configuración de infraestructura. Evita la creación manual de los componentes de infraestructura y automatiza procesos.
    b. Terraform: Herramienta multi nube, creación de módulos reutilizables.
    c. CloudFormation: Herramienta ofrecida por AWS.
4. ¿Qué métricas considera esenciales para el monitoreo de soluciones en la nube?
    a. Estados de los componentes (si está arriba o abajo)
    b. CPU
    c. Memoria
    d. Almacenamiento
    e. Tráfico de red
    f. Latencia
5. ¿Qué es Docker y cuales son sus componentes principales?
    a. Docker es una herramienta, la cual mediante la utilización de contenedores, permite la creación, configuración e implementación de aplicaciones de una manera más eficiente y portable independientemente del sistema operativo.
    b. Sus componentes principales son: DockerFile, Imágenes de Docker, Contenedores, Volúmenes, Redes.

Caso Práctico

Frontend

- Route53
- Cloudfront
- S3 (Bucket Policy)
- ACM

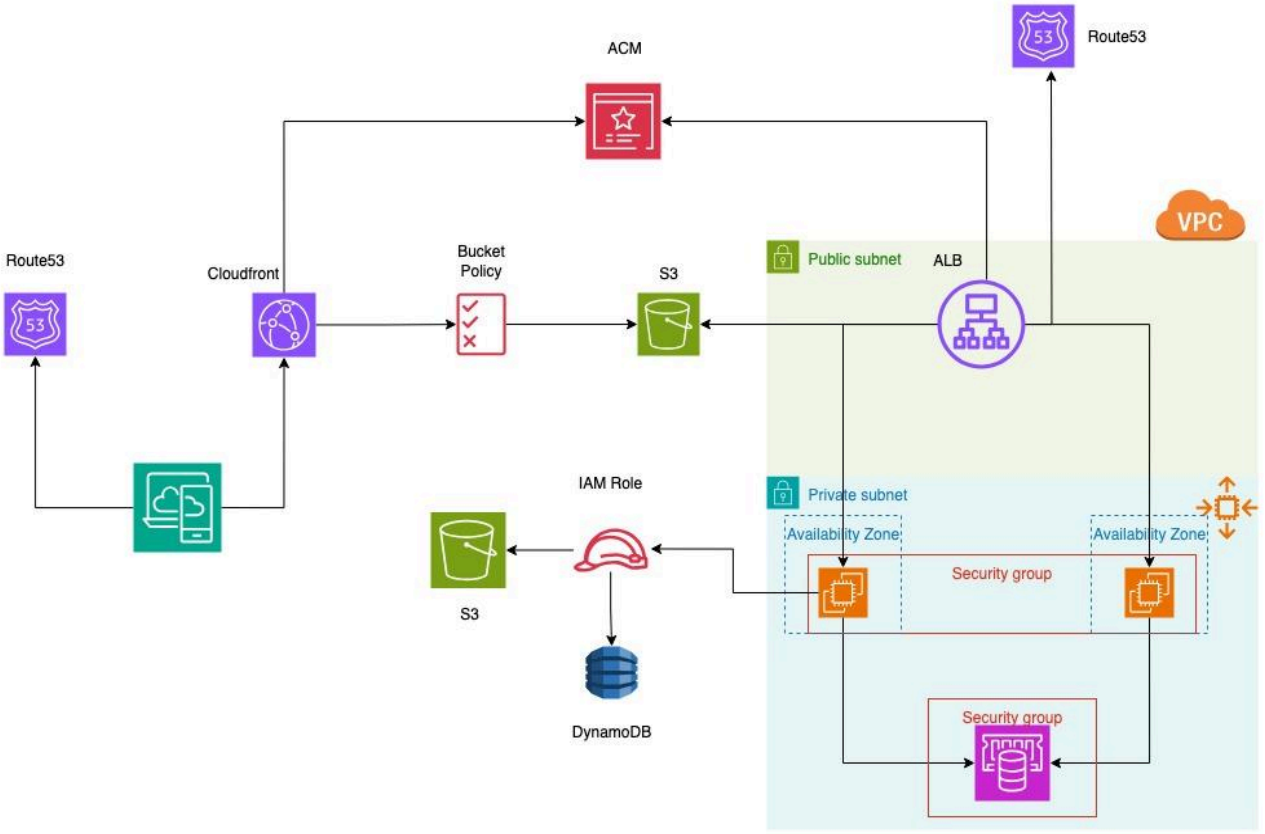
Solución serverless normalmente utilizada para contenido estático. Capacidad para manejar grandes cantidades de requests y en conjunto con Cloudfront para distribuir contenido lo más cerca posible a los usuarios. Igual en conjunto con Cloudfront se puede utilizar ACM para tener un sitio seguro con HTTPS. Route53 como DNS.

Backend

- VPC
- Application Load Balancer
- EC2
- AutoScaling
- RDS
- Security Groups
- DynamoDB o S3 para almacenamiento

Pudo haber utilizado igualmente una solución serverless con api gateway y lambdas, que muchas veces puede llegar a ser una solucion mas rapida de implementar a nivel de infraestructura y de bajo costo, pero he elegido un diseño con servidores EC2 para demostrar que igualmente me ha tocado manejar este tipo de proyectos donde usar lambdas no es suficiente, ya sea por capacidad de recursos o porque se ocupa que la aplicación o servicio esté siempre arriba.

Un diseño con escalabilidad y redundancia, un ALB para distribuir la carga a instancias EC2 distribuidas en diferentes zonas. Auto escalado configurado para colocar más máquinas de ser necesario cuando haya un crecimiento de carga. Instancias colocadas en redes privadas para mayor seguridad al igual que la base de datos. Para que la aplicación que está corriendo en las instancias pueda comunicarse de manera segura sin usar llaves con otros servicios de AWS como S3 o DynamoDB, se utilizan roles. Igual utilizar ACM en conjunto con ALB para tener un backend seguro con HTTPS. Route53 como DNS.


ACM
Route53
Route53
VPC
Public subnet
ALB
Cloudfront
Bucket
Policy
S3
IAM Role
Private subnet
Availability Zone
Availability Zone
Security group
S3
DynamoDB
Security group