1. ¿Cuál es la diferencia entre nube pública, privada e hibrida?
La nube pública es más escalable, fácil de acceder y más fácil de mantener. Por ejemplo, AWS se encarga de mantener y reemplazar la infraestructura, dándole mantenimiento o cambiando el hardware por uno nuevo, y el usuario final no se preocupa por eso. Por otro lado, en la nube privada es menos escalable, pero con la ventaja de tener más control y seguridad en la infraestructura. Por último, la nube híbrida podría usarse en aplicaciones que necesiten escalar mucho, pero que parte de su infraestructura necesite mayor control. Por ejemplo, los bancos, en los cuales hay algunos países que no permiten que los datos estén en otro país, por ende, se necesitaría una nube privada para el backend o la base de datos y, de repente, una página web que sí escale en una nube pública.

2. Describa tres prácticas de seguridad en la nube.

- Principio de mínimos privilegios: Ya sea a desarrolladores o servidores, es necesario darle solo los mínimos privilegios, porque en caso de que alguien externo logre acceder a un servidor, podría borrar servicios como Base de datos, elasticsearch, etc o simplemente cambiar el firewall y abrir más vulnerabilidades. También un desarrollador podria crear o eliminar sin querer un servicio por tener muchos privilegios.
- Cifrado de datos: Es necesario ya sea en la comunicación entre servicios o datos estáticos. Por ejemplo en la comunicación entre elasticsearch o entre servidores podria existir alguien que pueda acceder al trafico y ver información sensible incluso en una nube privada, en cuanto a los datos estáticos un programador podria acceder a los datos de la db, pero si estan cifrados es mucho más complejo.
- Backups y plan de recuperación ante desastres: Es tener copias de seguridad en diferentes regiones en caso de una catastrofe, tambien es bueno tener la infraestructura como código para poder crearla en otra region de forma sencilla, otras personas usan réplicas de la base de datos en diferentes zonas para no tener downtime.

3. ¿Qué es la IaC, y cuales son sus principales beneficios?, mencione 2 herramientas de IaC y sus principales caracteristicas.
Es definir y manejar la infraestructura en código. De esta forma, podemos recrear nuestra infraestructura o replicarla de forma sencilla en diferentes regiones, reutilizar variables de entorno como grupos de seguridad y zonas, etc. Dentro de sus principales beneficios está la automatización, es decir, tener scripts para crear, eliminar y actualizar la infraestructura de manera automática. Otro beneficio es tener control de versiones de las configuraciones de la infraestructura.
Las 2 de las más usadas son:

- **Terraform**: Puedes usarlo con diferentes proveedores de la nube(aws,azure,etc.)
- **AWS CloudFormation**: Es una herramienta nativa de AWS.

4. ¿Qué metricas considera esenciales para el monitoreo de soluciones en la nube?
Las principales serian **CPU** y **RAM**, si las métricas están estables pero la aplicacion lenta seria bueno monitorear **Network** y **IOPS** del almacenamiento.

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

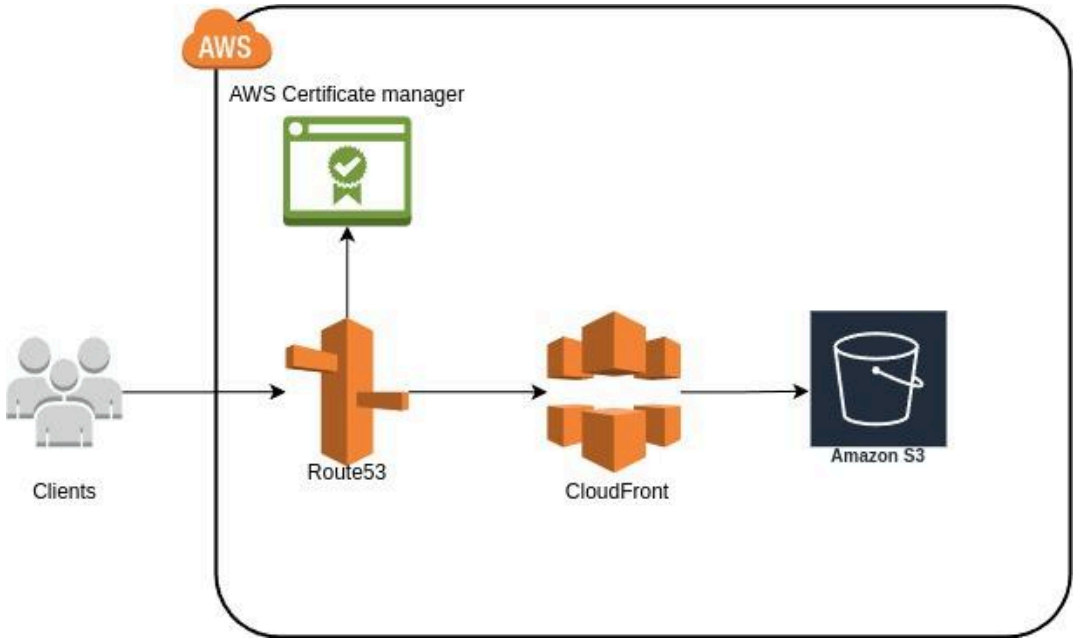
Es una herramienta que te permite crear y gestionar aplicaciones en contenedores, brindando portabilidad. Esto significa que tu aplicación puede funcionar de la misma manera en diferentes sistemas operativos y arquitecturas. Sus principales componentes son las **imágenes** y los **contenedores**. La aplicación se ejecuta en los contenedores, los cuales se crean a partir de las imágenes, esta característica nos permitirá tener muchos contenedores de forma sencilla siendo util en kubernetes y aws ECS.

6. Caso práctico.

El proveedor escogido es AWS.

Frontend

He escogido tener la aplicacion web en **s3 por su sencillez** y combinarlo con **cloudfront**, estos servicios de aws son muy robustos y en caso de miles o millones de requests no tendrían problemas. Si administramos bien los servicios el costo de esta infraestructura puede ser muy poco o nula. Por otro lado, en caso de muchas request(ataques) puede ser muy elevado el precio.



Backend

He escogido un **application load balance**r para poder distribuir la carga entre diferentes instancias de nuestra aplicación(api restful), y que sea fácilmente escalable usando un **autoscaling group**, pienso que tendriamos una instancia por defecto y agregar mas instancias y el uso de cpu es muy elevado. Para archivos estáticos podemos usar buckets de **S3.** Por último conectar nuestras Apis a una **RDS** con replica, opcionalmente podríamos agregar de forma intermedia un elasticCache(Redis). De forma opcional, si es una aplicación pequeña, podríamos cambiar RDS por una instancia pequeña de EC2 y administrar nuestra base de datos nosotros mismos.