

Preguntas teóricas:

1.-

**Nube Pública:** Es como usar el transporte público donde varias empresas y personas pueden compartir el mismo sistema de recursos, como servidores y almacenamiento, ofrecidos por empresas como AWS o Google Cloud. No se necesita preocuparse por mantenimiento y solo se paga por lo que se usa.

**Nube Privada:** Es más exclusivo para la empresa, lo cual te da más seguridad y control sobre cómo lo usas.

**Nube Híbrida:** Pongamos en el caso que tenemos nuestro propio auto pero usamos el transporte público cuando te conviene. Con la nube híbrida, se puede mezclar ambos mundos y mantener datos y aplicaciones más críticos en tu nube privada y usar la nube pública para tareas menos delicadas.

2.-

**Gestión de Identidades y Accesos.** - Configura y controla quién puede acceder. Utiliza el principio de menor privilegio, lo cual otorga solo los permisos mínimos necesarios para cada usuario y función. También se puede implementar autenticación multifactor para añadir una capa de seguridad extra.

**Encriptación de Datos.** - La encriptación es una barrera que protege los datos incluso si alguien logra acceder a ellos.

**Monitoreo y Alertas en Tiempo Real.** - Configurar alertas en tiempo real es fundamental para responder rápidamente a posibles amenazas.

3.-

IaC representa una metodología para gestionar y aprovisionar la infraestructura mediante archivos de configuración en lugar de procesos manuales.

### **Beneficios:**

- **Versionamiento y Trazabilidad.** - Esto permite identificar cuándo y quién realizó un cambio específico.
- **Inmutabilidad de Infraestructura.** - Asegurar que no haya configuraciones obsoletas y reduce el riesgo de errores en cambios a gran escala.
- **Escalabilidad y Eficiencia.** - Facilita el escalamiento automático de la infraestructura en la nube según la demanda.

### **Herramientas:**

- **Terraform.** - Usa su propio lenguaje de configuración, HCL, que está diseñado para ser intuitivo y declarativo. Esto facilita la definición de infraestructuras complejas de manera legible y manejable.

- **AWS CloudFormation.** - Al ser una herramienta de IaC desarrollada por AWS, CloudFormation proporciona soporte completo para la mayoría de los servicios de AWS, incluyendo los últimos lanzamientos.

#### 4.- Considero las siguientes:

- **Disponibilidad y Rendimiento de los Recursos:** Estas métricas miden qué tan bien funcionan los recursos del sistema, como el uso de la CPU, la memoria y la red.
- **Base de Datos:** Estas métricas evalúan el rendimiento de las bases de datos, como el tiempo que tardan en responder y cuántas conexiones tienen.
- **Disponibilidad y Fiabilidad:** Miden cuánto tiempo está en línea el servicio y si ocurren errores frecuentes.
- **Aplicación:** Estas métricas muestran el rendimiento de la aplicación en sí, como cuánto tarda en responder y cuántas solicitudes recibe.
- **Seguridad:** Miden el nivel de seguridad, como intentos de acceso fallidos y posibles vulnerabilidades.

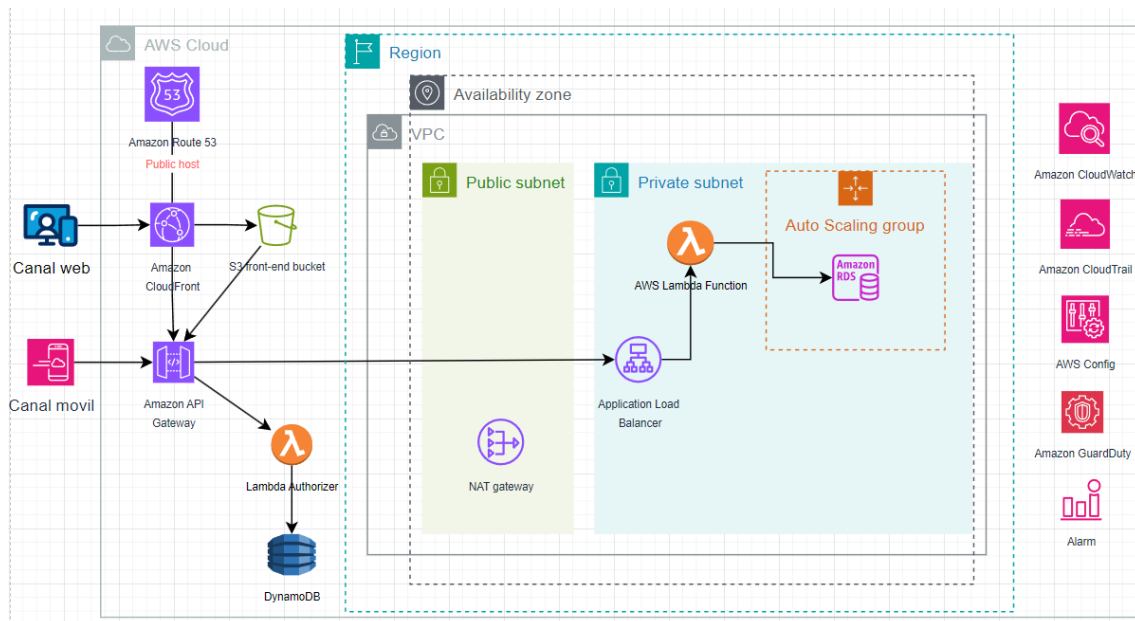
#### 5.-

Docker es una herramienta que permite crear contenedores, que son como pequeñas cajas donde puedes meter tu aplicación con todo lo que necesita para. Esto hace que la aplicación se ejecute de la misma forma en cualquier lugar, sin importar el entorno.

### Componentes de Docker:

1. **Docker Engine:** Es el motor de Docker, que permite crear y ejecutar los contenedores. Este tiene dos partes:
  - **Docker Daemon:** Un servicio que trabaja en segundo plano y gestiona los contenedores.
  - **Docker CLI:** La línea de comandos que se usa para comunicarte con el Daemon y decirle qué hacer.
2. **Imágenes:** Son como plantillas que contienen el sistema y los archivos necesarios para que la aplicación funcione.
3. **Contenedores:** Son instancias en funcionamiento de las imágenes, como si fueran copias en ejecución de una plantilla que están listas para usarse.
4. **Dockerfile:** Es un archivo donde se describe paso a paso cómo crear una imagen. Se indica qué instalar, qué copiar y cómo configurarlo.
5. **Docker Hub:** Es un repositorio en la nube donde se puede subir y compartir las imágenes de Docker, o descargar otras ya hechas.

#### 6.-



Para que esta aplicación funcione bien y sea rápida, escalable y segura, se ha diseñado un conjunto de servicios en la nube que se conectan y trabajan juntos de la siguiente manera:

- **Dominio y tráfico:**

- **Amazon Route 53** es como el directorio telefónico de tu aplicación en la nube. Dirige a los usuarios a la "ubicación" correcta dentro de Amazon, asegurándose de que encuentren tu aplicación fácilmente.

- **Distribución de contenido rápido (CDN):**

- **Amazon CloudFront** se encarga de que las imágenes, los videos y otros archivos estáticos que componen la interfaz de la aplicación frontend lleguen rápido a los usuarios. Funciona como una red global que almacena estos archivos en varias ubicaciones para que estén cerca de los usuarios, sin importar dónde se encuentren.

- **Almacenamiento de archivos del frontend:**

- **Amazon S3** es donde se guardan los archivos de la página web (HTML, CSS, JavaScript) que se ve en el navegador o en la app. Es como la "biblioteca" que contiene el contenido que el usuario verá.

- **Control de acceso y autenticación:**

- **API Gateway** recibe las solicitudes de los usuarios y se asegura de que lleguen al lugar correcto en el backend.
  - **Lambda Authorizer** trabaja con API Gateway para verificar que el usuario tenga los permisos adecuados para hacer lo que está intentando hacer. Para

esto, usa **DynamoDB**, que es una base de datos en la que se guarda la información de roles y permisos de los usuarios.

- **Backend escalable y flexible:**

- Cuando una solicitud de un usuario pasa el control de acceso, llega al **backend**, donde ocurren las operaciones principales:
  - **AWS Lambda** maneja tareas rápidas y de bajo consumo, como consultar o actualizar ciertos datos. Es perfecto para tareas que no necesitan mucha potencia o duración.
  - **Auto Scaling Group con EC2** maneja tareas más pesadas o de larga duración. Aquí, si muchas personas están usando la aplicación al mismo tiempo, el Auto Scaling Group agrega más servidores automáticamente para mantener el rendimiento, y luego los reduce cuando la demanda baja.
- Un **Load Balancer** se asegura de que las solicitudes se distribuyan de manera uniforme entre las diferentes instancias del backend, evitando que una sola se sobrecargue.

- **Base de datos:**

- **Amazon RDS** guarda los datos importantes de la aplicación, como información de los usuarios o registros de actividades.

- **Seguridad y monitoreo:**

- **CloudWatch** y **CloudTrail** funcionan como el sistema de vigilancia de la aplicación. CloudWatch detecta problemas de rendimiento y crea alertas si algo va mal, mientras que CloudTrail registra toda la actividad en los servicios, por si se necesita revisar algo o resolver problemas de seguridad.
- **GuardDuty** actúa como un guardia de seguridad que detecta amenazas, y **AWS Config** registra cualquier cambio que ocurra en la configuración de los servicios, manteniendo todo bajo control.