Reto Técnico - Ingeniería Cloud

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

La nuble publica consiste en el uso de servicios de computación y almacenamiento provistos por una empresa como Azure o AWS, a la cual se puede acceder como pago por consumo sin la necesidad de tener infraestructura propia.

La nube privada consiste en el uso de infraestructura exclusiva para una organización, donde esta tiene control tanto sobre el software como sobre los datos, sin embargo, es más costoso, generalmente la usan organizaciones donde la sensibilidad de las aplicaciones y los datos es muy alta como por ejemplo entidades financieras

La nube hibrida presenta una solución que mezcla los dos conceptos donde se usa la nube publica para ciertas soluciones o despliegues que requieran escalamiento, pero manteniendo la información, datos sensibles o aplicaciones criticas en una nube privada. Por lo tanto, la nube hibrida permite usar las características de la nube publica en conjunto con un mejor control de la data sensible en la nube privada.

Describa tres prácticas de seguridad en la nube.

Dentro de las principales practicas se tiene el Control de Acceso, el cifrado y el monitoreo:

Con el control de acceso se puede definir a detalle quien puede ingresar a la nube y definir según el rol la capacidad de cada usuario para leer, crear o modificar los componentes o servicios, limitando así la capacidad de cada usuario para hacer cambios no autorizados y protegiendo información sensible

El cifrado permite proteger los datos ya sea en transito como en reposo, es decir cuando están siendo intercambiados por instancias o servicios o cuan do están almacenados estáticamente, esto significa que si alguien intercepta la información no van a poder ser leídos sin las claves correctas, protegiendo la privacidad de los datos

El monitoreo por otro lado permite visualizar actividad sospechosa mediante un seguimiento de las actividades o indicadores, esto permite identificar posibles amenazas o ataques y tomar las medidas correctivas específicas.

¿Qué es la IaC y cuáles son sus principales beneficios? Mencione 2 herramientas de IaC y sus principales características.

La infraestructura como código permite gestionar la infraestructura y aprovisionamiento de recurso mediante código de esta manera se puede definir toda la arquitectura en archivos de código, esto facilita la creación modificación y despliegue automatizado y replicable de recursos.

La IaC brida beneficios como:

Rapidez en el despliegue automatizando la creación de recursos

Consistencia entre entornos, al usar el mismo código se pueden replicar exactamente los entornos de desarrollo, pruebas y producción, esto evita problemas al mover aplicaciones de un entorno a otro y reduce errores de configuración

Escalabilidad, al estar automatizado el despliegue es más fácil escalar o replicar la infraestructura para nuevos requerimientos.

Herramientas IaC:

La mayoría de las nubes públicas y privadas ofrecen herramientas para aplicar la IaC como ejemplo se pueden mencionar:

Terraform: Permite la gestión de infraestructura en múltiples proveedores como AWS, Azure y GCP, para su aplicación utiliza lenguaje declarativo donde se describe las específicamente como se debe crear la infraestructura, además mantiene este estado facilitando la automatización

AWS Cloud Formation: Es exclusiva de AWS y permite definir todos los recursos en archivos plantilla en formato JSON o YAML, la plantilla es leída y AWS configura la infraestructura según las especificaciones, muy útil si se trabaja exclusivamente con AWS ya que se integra al 100% con sus servicios.

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

Las principales métricas que considerar en el monitoreo son:

1.- Disponibilidad: Permite evaluar el tiempo que los servicios son efectivamente accesibles por los usuarios, dependiendo del servicio y la criticidad se debe garantizar una alta disponibilidad para evitar posibles riesgos reputacionales y pérdida de confianza de los usuarios, lo que podría desencadenar en pérdidas económicas, el monitoreo activo permite identificar una interrupción y activar efectivamente planes de recuperación inmediata.

2.- Latencia: esta métrica se relaciona con la experiencia del usuario al usar los servicios y mide el tiempo que un servicio tarda en responder, por lo cual una respuesta con un tiempo muy elevado puede causar una percepción de lentitud y falta de rendimiento. El monitoreo permite identificar estos problemas y realizar los ajustes necesarios para optimizar la entrega de servicios.

3.- Uso de CPU y Memoria: El monitoreo efectivo delos recursos del servidor permite identificar como los servicios están haciendo uso de los recursos, de esta manera si existe un uso elevado de CPU y memoria, podría indicar que se necesita escalar la infraestructura o por el contrario un bajo uso de recursos puede indicar una sobreasignación de recursos, estas métricas permiten adecuar la infraestructura al costo beneficio establecido para que los servicios funcionen correctamente con los recursos requeridos.

4.- Tasa de Errores: El número de errores en el sistema es una métrica que permite identificar patrones de fallos ya sea en la configuración o la arquitectura, para tomar medidas correctivas de forma proactiva y evitar una posible afectación a los usuarios

¿Qué es Docker y cuáles son sus componentes principales?

Docker presenta una solución que consiste en una plataforma de contenedores que permiten empaquetar dentro de estas aplicaciones y sus dependencias. Uno o varios contenedores se ejecutan sobre el mismo sistema operativo del host y comparten el mismo kernel, una diferencia clave con las máquinas virtuales que requieren de un sistema operativo completo en cada instancia. Con Docker se tiene una ejecución consistente en cualquier entorno sin problemas de compatibilidad, ya que los contenedores por ser autocontenidos no dependen del entorno en el que se ejecutan, siempre que compartan el mismo sistema operativo. Esto permite un rápido despliegue y configuración de aplicaciones.

Componentes:

1.- Docker Engine: Es el núcleo de Docker, y hace posible la creación, ejecución y gestión de los contenedores, es responsable también de la interacción con el sistema operativo del host, funciona a través de un servidor y una APIRest para el control de contenedores

2.- Docker Images: Son plantillas inalterables que contienen todas las configuraciones, dependencias y archivos necesarias para la ejecución de una aplicación.

3.- Docker Containers: Un contenedor es una instancia en ejecución de una imagen Docker, estos proveen un entorno aislado y ligero y se pueden crear, reiniciar detener en cualquier máquina que tenga instalada Docker.

4.- Docker registry: Almacena y distribuye imágenes de contenedores, como Docker Hub en donde se almacena las imágenes en un registro centralizado donde los equipos pueden reutilizar, compartir y distribuir configuraciones de aplicaciones que ya están contenerizadas

Caso Práctico

Cree un diseño de arquitectura para una aplicación nativa de nube considerando los siguientes componentes:

- Frontend: Una aplicación web que los clientes utilizaran para navegación.
- Backend: Servicios que se comunican con la base de datos y el frontend.
- Base de datos: Un sistema de gestión de base 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) sustente su selección.
- Diseñe una arquitectura de nube. Incluya diagramas que representen la arquitectura y justifique sus decisiones de diseño (Utilice https://app.diagrams.net/).

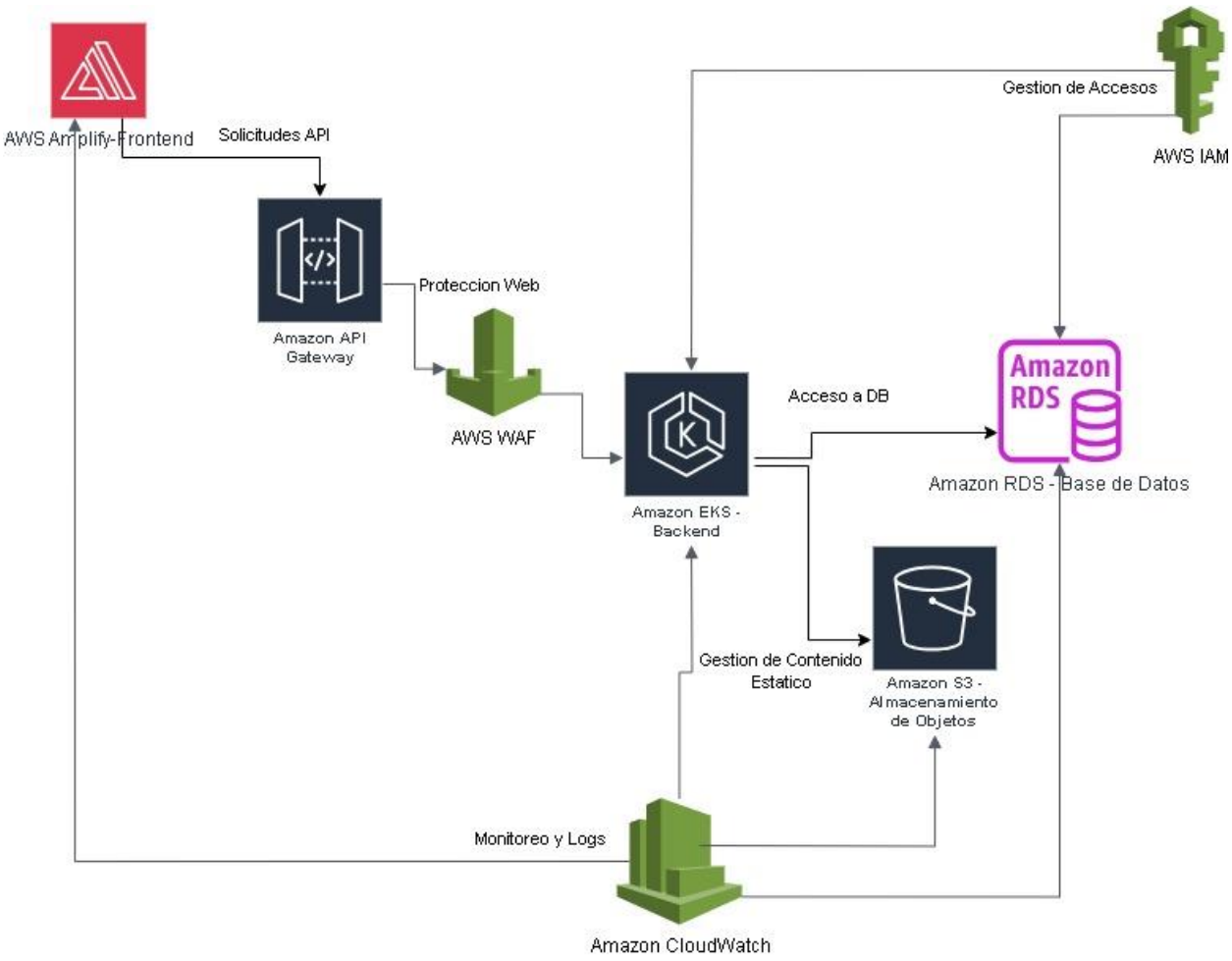
- Seleccione un proveedor de servicios de nube (Aws, Azure o GCP) sustente su selección.

Criterio	AWS	Azure	GCP
Servicios	Amplia variedad y especialización	Buena variedad, integración Microsoft	Amplia, menos especializados
Madurez	Líder de mercado	Menos maduro	Menos maduro
Escalabilidad	Excelente, Auto Scaling	Buena, menos flexible	Buena, menos opciones
Red Global	Más regiones y zonas	Amplia, pero menos que AWS	Sólida, menos regiones
Seguridad	Robusta, alto cumplimiento	Buena seguridad	Buena, menos certificaciones
Integración	Fluida con terceros	Buena con Microsoft	Buena con Google
Soporte	Extensa comunidad y soporte	Buena, menor que AWS	Buena, menor que AWS
Costos	Flexible y transparente	Competitivo, más complejo	Competitivo, menos opciones

Se opta por AWS debido a su amplia gama de servicios, madurez en el mercado, escalabilidad superior y robustez en seguridad, integración, además provee un soporte muy bueno y su estructura de costos es flexible.

- Diseñe una arquitectura de nube. Incluya diagramas que representen la arquitectura y justifique sus decisiones de diseño (Utilice https://app.diagrams.net/).

Diseño de Arquitectura



Componente	Servicio AWS	Justificación de la Elección
Frontend	AWS Amplify	Simplifica el despliegue automático y la integración continua para aplicaciones web.
Backend	Amazon EKS	Gestiona clústeres Kubernetes, proporcionando escalabilidad y resiliencia para microservicios.
Base de Datos	Amazon RDS	Ofrece una solución gestionada y escalable para bases de datos relacionales, reduciendo la carga administrativa.

Almacenamiento de Objetos	Amazon S3	Almacena y gestiona grandes volúmenes de datos estáticos con alta durabilidad y disponibilidad.
API Gateway	Amazon API Gateway	Facilita la gestión, seguridad y escalabilidad de las APIs entre frontend y backend.
Seguridad	AWS IAM, AWS WAF	IAM gestiona accesos y permisos de manera segura; WAF protege contra amenazas web comunes.
Monitoreo y Logging	Amazon CloudWatch	Monitorea el rendimiento y centraliza los logs para análisis y alertas en tiempo real.