

## Reto Técnico Ingeniero Cloud

Consideraciones para el entregable:

- Cree un documento PDF donde se detallen las respuestas teóricas y el caso práctico.
- El documento debe estar bien organizado y ser fácil de leer. Asegúrese de añadir cualquier texto explicativo que considere necesario.
- Suba el documento como respuesta a este ejercicio. Asegúrese de subirlo dentro del tiempo de resolución.
- Adicional, crear un repositorio público en Github y subir el PDF a ese repositorio. Colocar la URL al repositorio en los comentarios de este ejercicio.

Preguntas teóricas:

1. ¿Cuál es la diferencia entre nube pública, privada e híbrida?  
La nube pública es un servicio en línea donde los clientes comparten recursos, lo que hace que esta sea escalable y económica. La nube privada en cambio es exclusiva por organización, lo cual nos permite tener mayor control y seguridad. Por otra parte la nube híbrida combina ambas bondades, y nos permite almacenar los datos sensibles en la nube privada mientras podemos utilizar los recursos de la nube pública para tareas menos sensibles.
2. Describa tres prácticas de seguridad en la nube.
  - Cifrado de datos, para la protección de los datos en tránsito o reposo.
  - Principio de menor privilegio, garantizar que los usuarios o aplicativos solo cuenten con el permiso requerido para la realización de su función y reducir riesgos de acceso no autorizados.
  - Multi Factor Authentication (MFA), siempre se debe tener mínimo una capa más de seguridad para la autenticación.
3. ¿Qué es la IaC, y cuáles son sus principales beneficios?, mencione 2 herramientas de IaC y sus principales características.

La IaC, bien conocida como “Infrastructure as Code” nos permite aprovisionar y gestionar infraestructura de manera automatizada.

Como beneficio podemos mencionar:

- Versionamiento y control de cambios
- Despliegue mediante código asegurando rapidez.
- Reducción de errores manuales.
- Recuperación rápida.

Herramientas:

- Terraform: es agnóstico al proveedor y gracias a sus diferentes plugins nos permite ser utilizado para desplegar arquitectura en cualquier nube.
- Azure ARM: Nativa de Azure y permite desplegar recursos.
- AWS CloudFormation: Nativa de AWS y nos ayuda en el despliegue de la infraestructura.

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

- Disponibilidad.
- Latencia.
- Uso del CPU y memoria.
- Almacenamiento.
- Tasa de errores.
- Tasa de peticiones.

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

Docker es una plataforma agnóstica al sistema operativo que permite empaquetar aplicaciones y sus dependencias en contenedores ligeros y portables.

Componentes principales:

- Dockerfile

Sería el punto de partida para la configuración de un aplicativo, de ahí podríamos tener ejecución de comandos como:

- `docker build -t mi-app .`
- `docker run -d mi-app`
- `.dockerignore`

Evitamos que se copien archivos innecesarios al contenedor.

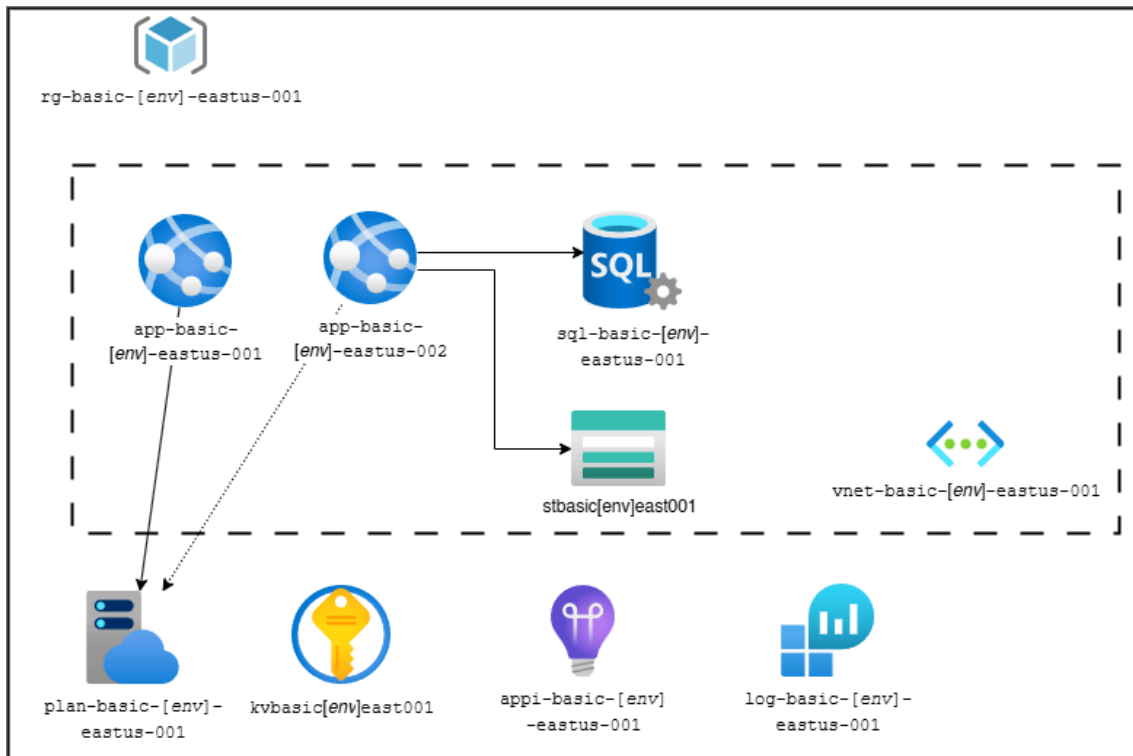
- `docker-compose.yml`

Se utiliza para definir los servicios y poder orquestar múltiples contenedores.

Caso práctico:

Diseño de la arquitectura: basic.drawio.png

Proveedor de servicio seleccionado: Azure porque nos ofrece mayor integración con respecto a los servicios empresariales, además también contamos con seguridad avanzada y escalabilidad.



He considerado los siguientes componentes para diseñar la arquitectura de la aplicación, aplicando la convención de nombres recomendada por Microsoft.

#### Administración y seguridad

- Resource Group; agrupa los recursos para facilitar su administración
- Key vault; almacena claves y secretos de manera segura, evitando la exposición de datos sensibles.
  - almacenar el string connection
  - variables de aplicativo
  - usuarios y contraseñas

#### Redes y comunicación

- Virtual network; nos ayuda con la intercomunicación de los distintos componentes y permite aislar accesos no autorizados.
  - redes privadas tipo 10.10.7.1/23

#### Aplicación y procesamiento

- App Service plan; define la capacidad de cómputo para las aplicaciones web, también permite escalabilidad y optimización de costos.
- Web App (Web Client) - [frontend]; aloja el aplicativo web expuesto para el usuario final.
- Web App (Web Api) - [backend]; funciones y logica del negocio para interactuar con la base de datos y otros sistemas.

#### Datos y Almacenamiento

- Azure Sql Server; gestiona la infraestructura de la base de datos, con su respectiva autenticación y seguridad.
- Azure Sql Server Database; base de datos relacional para almacenar información estructurada.
- Storage Account; almacenar imágenes en el contenedor que sea definido, archivos estáticos u otros que sean requeridos por la aplicación.

#### Monitoreo y Log

- Log Analytics Workspace; Centraliza los registros y métricas de la aplicación que nos ayuda con el monitoreo y análisis de datos.
- Application Insights; proporciona telemetría en tiempo real sobre la aplicación incluyendo errores y tiempos de respuesta.

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 (Utilice <https://app.diagrams.net/> ).