**Titulo**

Diseño de arquitectura seleccionada respecto del Reto - Migración de backend Mobile desde Onprem a AWS.

**Estado**

Aceptada

**Fecha**

2023-10-16

**Contexto**

Nuestro cliente ha desarrollado una aplicación móvil, esta aplicación tiene un backend conformado por una arquitectura de monolito, donde tanto la aplicación de backend como la base de datos están alojadas en el mismo servidor dentro del datacenter corporativo del cliente. La empresa quiere modernizar su backend, planteando migrar a la nube para así poder aprovechar los beneficios que esta provee.

Para ello se requiere que la arquitectura resuelva los problemas y requisitos planteados a continuación:

• Mejora del rendimiento

• Que ofrezca alta disponibilidad

• Que sea escalable

• Que optimice los costos de infraestructura

• Que permita la evolución de los servicios y funcionalidades mediante despliegues parciales ya sea por regiones o bien por grupos de la cartera de clientes

• Que permita la adición de nuevas funciones en el futuro sin afectar los servicios en operación

• Que haga un uso eficiente de los recursos asignados

• Contar con mecanismos que mejoren los aspectos de seguridad, asegurando que solo los usuarios y aplicaciones autorizadas puedan utilizar los servicios web

**Decisión**

- Basándonos en los requerimientos solicitados, consideramos que una arquitectura totalmente serverless, se adapta perfectamente ya que proporciona.
    - **Escalabilidad automática y Alta disponibilidad**: La arquitectura serverless por diseño es altamente disponible y se escala automáticamente en función de la demanda, lo que significa que puede manejar picos de tráfico sin necesidad de intervención manual.
    - **Costos reducidos**: Solo se paga por los recursos que se utilizan, lo que puede resultar en un ahorro significativo en comparación con la infraestructura tradicional, haciendo un uso eficiente de los recursos y optimizando costos
    - **Desarrollo ágil**: Permite a los desarrolladores centrarse en escribir código y crear funcionalidades, en lugar de preocuparse por la gestión de servidores. Permitiendo además la evolución de los servicios mediante despliegues parciales añadiendo nuevas funcionalidades sin afectar a los servicios en operación
    - **Mantenimiento reducido**: Al no tener que administrar servidores, se reduce la carga de mantenimiento y actualizaciones de infraestructura.
- Nuestra aplicación monolito (onprem) deberá pasar por diversas fases antes de convertirse en una arquitectura totalmente serverless
    - Fase 1: Evaluación y movilización
    - Fase 2: Migración de la base de datos (DMS+CDC)
    - Fase 2.1 Migración backend APP (MGN)
    - Fase 3 Elección de patrones de migración
    - Fase 3.1 Romper el monolito en microservicios
    - Fase 4 CI/CD Pipeline| Mobile Backend Server | Disaster Recovery

**Consecuencias**

**Positivas:**

- **Escalabilidad:** La arquitectura serverless permite escalar automáticamente los recursos de la aplicación en función de la demanda, lo que puede mejorar el rendimiento y reducir los costos.
- **Agilidad:** La arquitectura serverless permite a los desarrolladores centrarse en la lógica empresarial en lugar de preocuparse por la infraestructura subyacente. Esto puede acelerar el tiempo de comercialización y mejorar la agilidad empresarial.
- **Costo:** A largo plazo, la arquitectura serverless puede ser más rentable que una aplicación monolítica, ya que solo se paga por los recursos utilizados.
- **Resiliencia:** La arquitectura serverless puede ser más resistente a las fallas que una aplicación monolítica, ya que los servicios se ejecutan en entornos aislados y pueden recuperarse automáticamente de las fallas.
- **Facilidad de mantenimiento**: La arquitectura serverless puede ser más fácil de mantener que una aplicación monolítica, ya que los servicios son independientes y se pueden actualizar o reemplazar sin afectar a otros servicios

**Negativas:**

- **Complejidad:** La arquitectura serverless puede ser más compleja que una aplicación monolítica, ya que se compone de múltiples servicios y componentes que deben trabajar juntos para proporcionar una solución completa.
- **Costo:** Aunque la arquitectura serverless puede ser más rentable a largo plazo, el costo inicial de migrar a una arquitectura serverless desde un monolito puede ser alto debido a que la complejidad de este tipo de migraciones, que en la mayoría de los casos requiere una refactorización del código, siendo imprescindible una buena capacitación del equipo de desarrollo para familiarizarse con los servicios serverless, además durante el periodo de migración habrán de coexistir ambas infraestructuras con el consiguiente coste. Además, el modelo de facturación basado en el uso puede ser difícil de predecir y controlar.

- **Latencia:** La arquitectura serverless puede tener una mayor latencia que una aplicación monolítica, ya que los servicios deben iniciarse y detenerse dinámicamente en función de la demanda, las llamadas entre servicios introducen una latencia adicional.
- **Desafíos de Base de datos:** En el caso de que un microservicio necesite información de varios microservicios donde cada uno tiene su propia base de datos, se produciría una latencia adicional ya que cada consulta añade un tiempo extra a la ejecución, en estos casos sería conveniente introducir un enfoque de mensajería asíncrona mediante el uso de colas (SQS), en pro de mantener la consistencia de las bases de datos y disminuir la complejidad de realizar múltiples consultas a múltiples bases de datos.
- **Dependencias:** Las aplicaciones serverless pueden tener dependencias complejas entre servicios, lo que puede dificultar la depuración y el mantenimiento.
- **Transacciones Distribuidas:** Coordinar transacciones distribuidas es complicado y puede afectar la consistencia y disponibilidad.
- **Gestión de Versiones:** Cada microservicio tiene su propio ciclo de vida. Actualizar y mantener múltiples versiones puede ser complicado. Se deben establecer prácticas sólidas de control de versiones y despliegue continuo.
- **Monitorización y Depuración:** En una arquitectura monolítica, la monitorización es más sencilla. En microservicios, se necesita una herramienta de monitorización más avanzada como Prometheus o Grafana que pueden ayudar a rastrear y depurar problemas en microservicios.
- **Seguridad:** La seguridad es un problema importante en cualquier arquitectura, pero la arquitectura serverless puede presentar desafíos únicos. Por ejemplo, el ritmo rápido de cambios en la aplicación , donde las actualizaciones frecuentes pueden introducir nuevas vulnerabilidades, o la gestión de identidad y el control de acceso, la cual debe ser sólida para evitar accesos no autorizados.

**Compliance**

- Nuestro cliente opera en Europa y Estados Unidos Por lo que ha de cumplir con los estándares GDPR e HIPAA)

**Notas**

- Author: Fran Díaz

- Version: 0.2
- Changelog:
    - 0.1: Versión Inicial propuesta.
    - 0.2: Se añade Compliance
    - 0.3: Cambios Propuestos