Título

FASE 2: DMS – Database Migration

Estado

Aceptada

Fecha

2023-10-19

Contexto

- Como indicábamos en la primera fase, migraremos base de datos alojada en el datacenter de nuestro cliente hacia el cloud, la base de datos origen utiliza un motor Postgre SQL que migraremos a una base de datos Aurora serverless (postgre SQL).

Decisión

- Durante el proceso de diseño, se consideraron varias alternativas para migrar los datos, incluida la replicación heterogénea y la migración manual. La migración manual se descartó a fin de evitar errores humanos, ya que la integridad de la base de datos es crítica para la empresa, por otro lado, no era necesario el uso de una estrategia heterogénea, ni la conversión de su esquema con SCT, ya que el motor de la base de datos origen era el mismo al de la base de datos destino. Por lo tanto, después de evaluar cuidadosamente cada opción, se decidió utilizar DMS - Migración homogénea + (CDC) replicación continua debido a su capacidad para migrar grandes volúmenes de datos con poco o ningún tiempo de inactividad y a que la base de datos origen de nuestro cliente tiene un motor compatible con nuestra base de datos destino (Aurora Serverless (potsgre)). Finalizada la migración podremos apuntar nuestra backend APP hacia el cloud conectando de esta forma nuestra aplicación a la nube, y eliminando la base de datos onprem una vez realizadas las pruebas pertinentes. Esto nos permitirá pasar a la siguiente fase.

Consecuencias

Positivas:

- La implementación exitosa de esta solución permitirá a los usuarios finales acceder a los datos migrados en una base de datos Amazon Aurora altamente disponible y escalable.
- Esta migración a cloud permitirá a nuestro equipo de base de datos realizar pruebas de rendimiento y optimización, pudiendo de esta forma descartar o corregir que los problemas de rendimiento (cuello de botella) indicados por el cliente no están en la base de datos.
- Nuestra migración al ser homogénea prescinde de usar Schema Conversion Tool (SCT), haciendo que la migración sea menos compleja.

Negativas:

- **Tiempo y recursos adicionales:** Realizar un replatforming de la base de datos antes del rehosting puede requerir tiempo y recursos adicionales, ya que necesita planificar, migrar y probar la base de datos antes de mover la aplicación. Esto podría retrasar el proceso de migración.
- **Posible interrupción durante la migración:** Si no se planifica cuidadosamente, el replatforming de la base de datos puede resultar en tiempo de inactividad durante la migración, lo que podría afectar a la disponibilidad de la aplicación.
- **Costos iniciales más altos:** Aunque a largo plazo se esperan ahorros de costos, la inversión inicial en la optimización de la base de datos y la migración puede ser más alta en comparación con un rehosting más simple. Por otro lado, hasta no haber finalizado la migración y poder realizar el cutover, habrá que costear ambas infraestructuras tanto la de origen onprem, como la de destino en AWS.

Compliance

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

Procedimiento

- https://catalog.workshops.aws/well-architected-migration/en-US/2-migration/1-initial-credentials-setup/2-quotas
- Para que nuestra migración se realice de forma privada y segura realizaremos una conexión mediante vpn entre nuestro data center onprem y nuestra vpc.

- Crearemos la instancia de replicación
- Crearemos los endpoint de origen (nuestra base de datos onprem) y destino (Aurora Serverless (PostgreSQL)).
- Identificaremos el numero disponible de tablas y objetos para ser migrados
- Nuestro motor de base de datos onprem es homogéneo y compatible con la versión postgre que seleccionamos en Aurora, por lo que realizamos un Full-Load task y activamos Change data capture (CDC) para tener un mínimo downtime.

Notas

- Autor: Fran Díaz
- Version: 0.2
- Changelog:
    - 0.1: Versión Inicial propuesta.
    - 0.2 Cambios propuestos