Título

FASE 4.2: Phase 4.2 Mobile Backend Serverless con DR

Estado

Aceptada

Fecha

2023-11-2

Contexto

- Ya tenemos diseñada nuestra arquitectura 100% serverless, que aunque es altamente disponible y escalable a nivel de región gracias a su diseño, ¿Que ocurriría si la región donde tenemos alojada nuestra infraestructura se cae? ¿Cual sería nuestro RPO y RTO?.

Decisión

- Teniendo en cuenta que nuestra aplicación se basa en un modelo serverless, es posible tener un DR active-active por un pequeño coste, lo que nos permitirá tener una RPO maximo de 30 segundos y un RTO inferior a 1 minuto. Según compliance tenemos clientes en Europa y Estados unidos por lo que una arquitectura DR- Active-active encaja perfectamente para cumplir los requisitos SLA. Además, el uso de Cloudfront como ya vimos nos aportará mejor latencia y rendimiento.

  Como consideraciones a la hora de diseñar nuestro DR hemos identificado y dividido los servicios de AWS en 3 subgrupos amplios:

- **Servicios de datos persistentes**: en nuestro caso son serían los servicios de S3 y Aurora Serverless, los cuales deben ser implementados sin pérdida de datos.
- **Servicios basados en Código**: que serían servicios como Api Gateway y Lambda los cuales no almacenan ningún tipo de datos y están basado en configuraciones.
- **Aplicaciones Globales**: Amazon 53 y Cloudfront nos ayudaran a configurar nuestro DR.

En las arquitecturas Active-Active podemos usar Cloudfront o Global acelerator, debido a nuestro interés en hacer uso del caching a nivel de borde descartamos este último.

Consecuencias

Positivas:

- **Minimiza daños**: un plan de contingencia permite anticiparse a los problemas, lo que nos permitirá minimizar el impacto negativo en nuestros sistemas y servicios.
- **Garantiza la continuidad del negocio**: un plan de disaster recovery garantiza que la empresa pueda seguir funcionando, aunque hayan surgido problemas con los sistemas. La continuidad del negocio es fundamental para evitar pérdidas y garantizar un buen servicio a sus clientes.
- **Garantiza el acceso a los datos**: un plan de disaster recovery no solo se centra en recuperar los datos, sino que también en las funcionalidades necesarias para que la empresa pueda continuar trabajando2. Con un plan de disaster recovery las relaciones comerciales y los servicios que se ofrecen a los clientes no se verán afectados**.**

Negativas:

- **Requiere actualización constante**: un plan de disaster recovery debe incluir todos los sistemas críticos de la organización y adaptarse a las nuevas tecnologías y soluciones en la nube. Esto implica un esfuerzo de mantenimiento y revisión periódica del plan.
- **Requiere inversión económica**: un plan de disaster recovery implica un coste asociado a la implementación y el uso de las infraestructuras y servicios necesarios para garantizar la recuperación de los sistemas. Este coste puede variar según el nivel de disponibilidad y redundancia que se quiera alcanzar. En nuestro caso particular nuestro cliente solicitaba que la aplicación debía estar disponible en varias regiones por lo que dicho gasto está totalmente justificado.
- **Requiere pruebas y auditorías**: un plan de disaster recovery debe ser probado y auditado regularmente para comprobar su efectividad y detectar posibles fallos o mejoras. Estas pruebas pueden suponer una interrupción temporal del servicio o un riesgo de pérdida de datos si no se realizan correctamente.

Compliance

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

Procedimiento

Al implementar la configuración active-active en varias regiones, hay tres áreas clave que necesitan mayor atención:

- **Certificados Secure Sockets Layer/Transport Layer Security (SSL/TLS) para sus dominios personalizados que utilizan sus clientes para acceder a su aplicación.**
- **La lógica de enrutamiento basada en latencia para enrutar las solicitudes de los clientes a una región de AWS.**
- **Replicación de los Servicios persistentes.**

Para que el tráfico se cifre en tránsito en una configuración de varias regiones, deben estar disponibles certificados SSL/TLS coincidentes en cada región donde implementemos la aplicación. A tener en cuenta que Cloudfront es un servicio global por tanto deberemos emitir su certificado en la región del Norte de virginia.

Veamos el el flujo dns de nuestro diagrama:

1- Crearemos un registro de alias de Route 53 con un dominio personalizado myapp.com que apunte al nombre de dominio de CloudFront predeterminado para la distribución.
2- Configuremos nombres de dominio alternativos (CNAME) en CloudFront e importaremos el certificado desde ACM público de AWS en la región EE. UU. Este (Norte de Virginia).
3- Configuremos conjuntos de registros de Route 53 para los servicios regionales de AWS. Crearemos dos registros con el mismo nombre de dominio, api.myapp.com estableceremos la política de enrutamiento en latencia. En una configuración activo-activo de varias regiones, el tráfico basado en latencia enruta el tráfico a cada región que contiene la mejor latencia con un tiempo de ida y vuelta más bajo.
4- Solicitaremos un certificado público en ACM para cada región donde tenemos un servicio de AWS público.

Veamos ahora el flujo de la lógica de enrutamiento donde hay dos puntos de resolución dns antes de que API Gateway atienda la solicitud.

- En el navegador de cliente, resolviendo el nombre de dominio en la ip de Cloudfront.
- En la ubicación de borde de Cloudfront, resolviendo el nombre de dominio personalizado de Api Gateway.

1- El cliente realiza una solicitud HTTPS al dominio myapp.com. Route 53 resuelve el dominio

2- El cliente realiza la solicitud a la ubicación de puntos de presencia (PoP) de CloudFront más cercanos.

3- CloudFront acelera la distribución de su contenido al enrutar cada solicitud a través de la red global de AWS a la ubicación perimetral que mejor pueda servir su contenido. Normalmente, se trata de un servidor perimetral de CloudFront que proporciona la entrega más rápida al espectador.

4- La ubicación de CloudFront PoP realiza una solicitud DNS a Route 53 para resolver el nombre de dominio de origen myapp.com. Dado que este nombre de dominio tiene la política de enrutamiento basada en latencia configurada en la Ruta 53, las direcciones IP se devuelven del API Gateway en la región más cercana a la ubicación de CloudFront PoP.

5- La ubicación de CloudFront PoP realiza una solicitud HTTP a API Gateway en la región más cercana.

Por último, afrontamos el flujo de la replicación de los datos persistentes.

6- Nuestra base de datos está configurada a modo base de datos global con un clúster primario en la región de Europa y uno secundario en Estados unidos, lo que brinda da la capacidad de replicar los datos sin impacto en el rendimiento en la base de datos y permitiendo lecturas locales en cada región con baja latencia. La conmutación por error se puede completar en menos de 1 minuto. En este paso se implementa Write forwarding en nuestra base de datos Aurora Serverless, lo que reenviará las solicitudes de escritura al clúster primario para después replicarlo en el clúster secundario, esto permite también que la implementación de la aplicación sea independiente de los puntos finales.
7- Activamos la replicación Bidireccional de S3 los que nos permitirá replicar cualquier documento agregado al bucket de S3 en esa región, estando disponible de forma casi inmediata la otra. Se ha tenido en cuenta que los archivos que se van a almacenar en nuestro bucket de S3 son inferiores a 10MB, tomando una velocidad mínima de 10 Mbps la cual es bastante reducida, obtenemos que en ningún caso nuestro RPO será superior a 30 segundos.

Conclusión

El costo adicional de aplicar DR a nuestra infraestructura backend serverless es relativamente bajo si tenemos en cuenta que eliminamos el tiempo de inactividad en caso de interrupción regional, no olvidar nuestra capa lógica al ser serverless solo pagará el número de solicitudes procesadas suponiendo un gran ahorro.

Notas

- Author: Fran Díaz
- Version: 0.2
- Changelog:
    - 0.1: Versión Inicial propuesta.
    - 0.2: Modificaciones propuestas.