

## Objetivos

Con este desafío, queremos evaluar tu capacidad para resolver problemas del mundo real, aplicando principios de diseño de sistemas, y buenas prácticas de arquitectura Cloud. Por lo tanto, te recomendamos que seas lo más detallado y explicativo posible con la solución que construyas, ya que esto nos ayudará a comprender tu manera de pensar y resolver problemas, así como también tu capacidad para aplicar conocimientos teóricos en la práctica.

En el caso de que los conceptos y tecnologías planteados en el desafío sean totalmente nuevos para vos, no te preocupes, ya que el resultado del desafío es evaluar tu capacidad para aprender nuevas tecnologías y herramientas; por lo tanto puedes hacer las preguntas que creas necesarias, nosotros vamos a intentar guiarte dentro de las posibilidades del desafío.

Cualquier duda que tengas con respecto al enunciado/objetivos del desafío, la puedes consultar por email, y la vamos a intentar responder lo antes posible.

Para el desarrollo de la solución, recomendamos poner foco en los siguientes aspectos:

- Tolerancia a fallos
- Escalabilidad
- Recuperación ante desastres

## Enunciado

Nuestro cliente, *Proyecto X*, quiere implementar una nueva infraestructura de 3 capas en AWS en su ambiente productivo. La misma consiste en una capa de aplicaciones de **Frontend**, otra capa de aplicaciones **Backend**, y una capa de **Database**. Previamente a implementar la solución, el cliente nos solicitó que realicemos el diseño de la arquitectura, incluyendo los componentes de infraestructura de red, aplicaciones, etc. El cliente hizo especial énfasis en la continuidad del negocio y la recuperación ante desastres, por lo tanto la solución debe incluir una estrategia de recuperación ante desastre, acorde con los tiempos requeridos por el cliente (detallado a continuación).

Para simplificar el propósito del desafío, vamos a definir como están construidas los distintos componentes de la infraestructura de 3 capas:

- **Frontend**: Hay 2 aplicaciones de Frontend, *home* y *checkout*, las cuales deben estar montadas sobre ECS, y las mismas deben estar en HA (High Availability).
- **Backend**: Hay 3 aplicaciones: *payments*, *products* y *shipping*. Todas ellas deben estar desplegadas en Lambda. Solo la app de *products* necesita acceso a la DB, las demás solo tienen conexión hacia APIs externas.
- **Database**: PostgreSQL, la cual está desplegada sobre RDS. Todas las aplicaciones usan el mismo cluster de RDS.
- **S3 buckets**: Hay 2 buckets de S3 que son usados por la aplicación de backend para guardar metadata de los pagos y ordenes de envío.

La nueva arquitectura debe cumplir los siguientes requisitos solicitados por el cliente:

- **High Availability**: La infraestructura debe estar configurada de manera que la falla de una zona de disponibilidad de AWS no afecte al funcionamiento

de la aplicación. Este punto se debe aplicar a cada uno de los componentes mencionados anteriormente (Front, Back y DB).

- La infraestructura debe soportar la caída de una region completa de AWS. El cliente esta dispuesto a tolerar unos minutos de downtime y de perdida de datos, los cuales no deben exceder los 10 minutos.
- El cambio de region, en caso de una falla, debe estar monitoreado y ser un proceso semi-automatizado, donde el cliente solamente tenga que "tocar un boton" para ejecutar el cambio a la region de recovery.
- Las redes deben estar segmentadas de acuerdo a sus funciones. Por ejemplo, las aplicaciones de Backend tienen que estar separados a nivel de red de las aplicaciones de Database.
- La comunicación entre aplicaciones tiene que estar protegida mediante el uso de firewalls.
- Desde las redes internas, solo se debe permitir conexiones salientes (outbound), pero no conexiones entrantes (inbound).

Los puertos en los que escucha cada aplicación son los siguientes:

- **Frontend:** TCP/80 y TCP/443
- **Backend:** No está expuesta a nivel de red, la función lambda es invocada mediante sus respectivas APIs.
- **Database:** TCP/5432

## Entregables

Los siguientes puntos son los entregables que se deben completar para el desafío. Es recomendable incluir todo el detalle que les sea posible, para qué la persona que lo evalúe pueda tener una mejor perspectiva de como desarrollaron la solución.

- Diagrama de arquitectura mostrando los diferentes componentes y como están conectados entre si. También se debe detallar las configuraciones de

los componentes clave, como tablas de ruteo, reglas de SecurityGroups, roles de IAM, conexiones entre regiones, etc.

- Documentación y explicación de la arquitectura. En este punto es recomendable poner foco en explicar y justificar las decisiones de diseño. Cuanto mas informacion nos brindes en este punto, mas rapido vamos a entender tu manera de razonar y resolver problemas.
- Recomendaciones sobre posibles puntos de mejora de la arquitectura a futuro.
- Agregar componentes de infraestructura faltantes: si detectas que el enunciado no especifica ciertos componentes de infraestructura necesarios, podes agregarlos y justificar la decision de diseño. Valoramos mucho la iniciativa y la proactividad. Tienes la libertad de agregar componentes que creas necesarios para la solucion, solamente recuerda explicar los motivos por los cuales decidiste agregarlos.