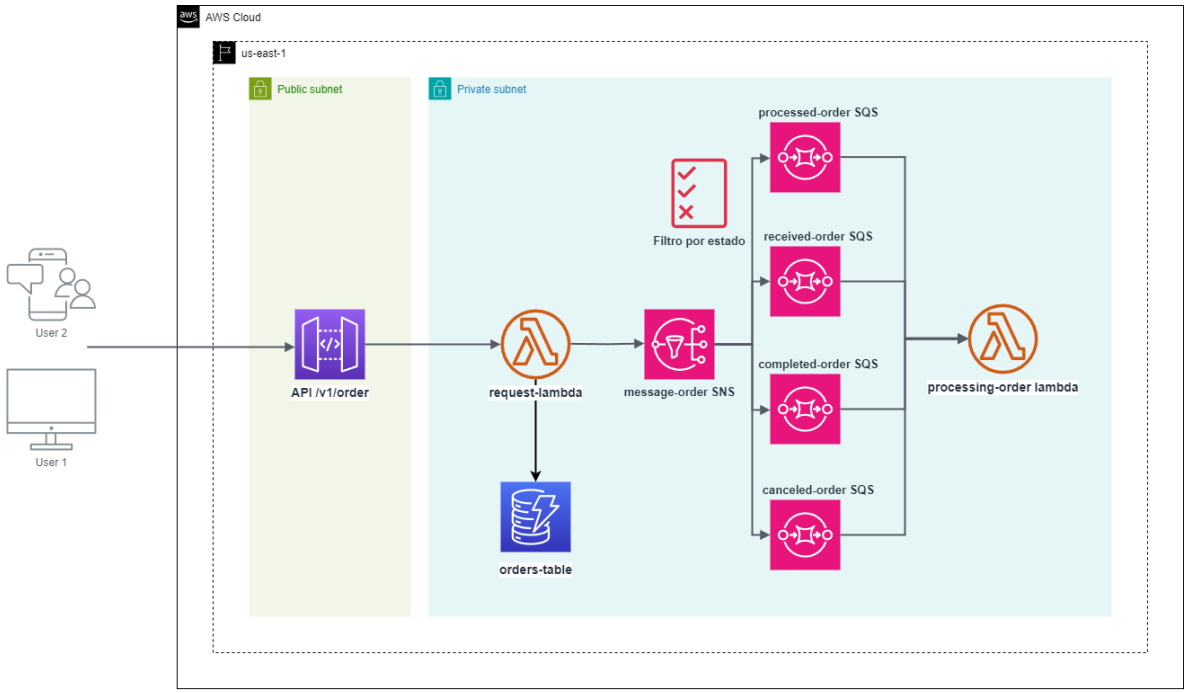
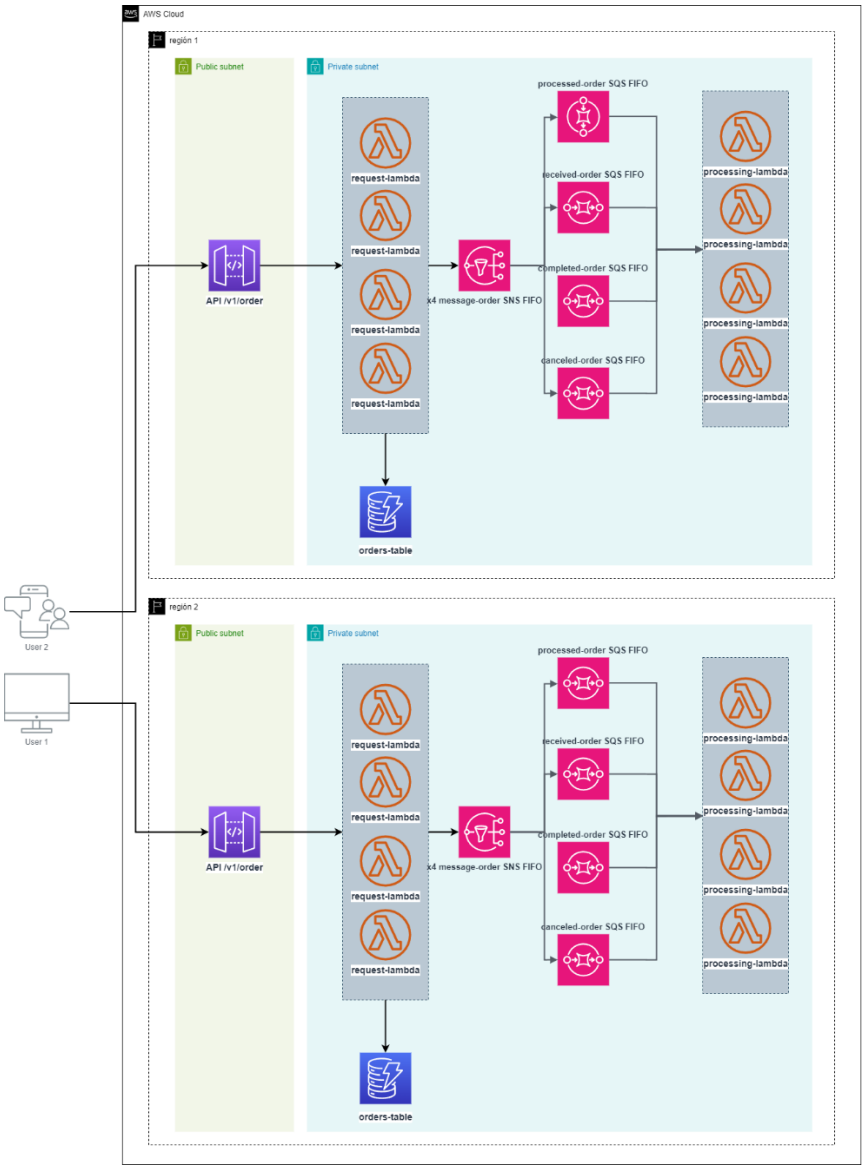
Arquitectura propuesta para la solución



- Se utilizó API Gateway para recibir las solicitudes del cliente.
- Se creó una lambda request para recibir las solicitudes del api, guardar en la base de datos y enviar un mensaje al sistema de colas, se usó lambda por su facilidad de instalación y altamente escalable para un proceso de bajo esfuerzo en procesamiento.
- Se utilizó base de datos dynamo para almacenar la información por su facilidad de instalación y altamente escalable, dado que no se tiene una fuerte relación entre las entidades de los datos es una buena opción.
- Para el sistema de colas, se implementó una integración entre sns para recibir los mensajes de las solicitudes y aplicar una política de filtros por atributos del mensaje y enviar una cola específica para cada estado de solicitud, generando desacople en desarrollo y generando escalabilidad de infraestructura en caso de requerirse más estados.
- Finalmente se creó la lambda processing para recibir los mensajes de cada cola y procesarlos.

Propuestas de escalamiento

Alternativa #1



En caso de que se incremente el volumen de las solicitudes que lleguen al límite de concurrencia de 1000 lambdas, se propone escalar a crear una lambda por cada estado y tener mayor capacidad de concurrencia. Si aún así se requiere más capacidad, se puede optar por el multi región y replicar la infraestructura en otra región cercana geográficamente y lograr una doble capacidad de la infraestructura inicial.

Para garantizar el orden de llegada de las solicitudes, se propone configurar los temas de sns y colas sqs de estándar a **FIFO.**

Alternativa #2

Como otra alternativa, se propone reemplazar las lambdas por kubernetes donde se puede alcanzar un límite hasta 50.000 pods aprovisionando la cpu y memoria necesaria para alcanzar la máxima transaccionalidad deseada.