

# Exposición de capacidades contenerizadas en EKS a sistemas externos

Last updated by | svchca07 | 18 de sept de 2023 at 5:54 p. m. COT

 **Importante:** Esta página está en construcción

## Caso5: Exposición de capacidades contenerizadas en EKS a sistemas externos:

Desde la capa applicativa se quiere tener estandarizado y controlado el acceso a los servicios expuestos en los EKS corporativos.

Cuando se habla de consumo de servicios en EKS tenemos identificados 12 escenarios posibles, los diagramas lógicos y la documentación de cada escenario se encuentran en el [siguiente link](#).

Los escenarios documentados en dicha guía son:

- **Escenario1 Consumidor es Externo a la organización y requiere acceso a capacidades contenerizadas en EKS a través del API Connect:**

En este caso se habla de aquellas APIs gestionadas que tienen como capa proveedora un servicio expuesto en un EKS en AWS.

- **Escenario2 Consumidor es Externo a la organización y requiere acceso a capacidades contenerizadas en EKS:**

Este escenario solo aplica para servicios que solo tienen un consumidor o que en su evaluación no amerita ser incluidos en la capa de APIs Gestionadas. Este escenario esta diseñado para dos tipos de consumidores: Consumidores de usuario final como aplicaciones móviles o web, y para aplicaciones que estén por fuera de la red interna de la organización como pueden ser aplicaciones de terceros o aplicaciones de otra empresas del Grupo Bancolombia.

Para este caso, la exposición de estas capacidades se puede realizar a través del componente API Gateway de AWS, tal y como se tiene documentado en la Wiki [INTEGRACION DE TERCEROS CON APLICACION BANCO USANDO mTLS Y FLUJO OAUTH 2.0 CLIENT CREDENTIALS](#)

- **Escenario3 Consumidor está interno on-premise y requiere acceso a capacidades contenerizadas en EKS a través del API Connect:**

Si una API puede ser reusable por múltiples consumidores se debe validar si es candidata a ser gestionada en el API Connect, cumpliendo los lineamientos de integración empresarial de la organización. En este caso son aquellos servicios que están expuestos en el EKS en AWS y por lo tanto deben ser expuestos en el API GW de API Connect que se encuentra en AWS.


Existe un caso de uso documentado en la wiki llamado [INTEGRACION DE API CONNECT CON APLICACIONES INTERNAS AWS UTILIZANDO ISTIO COMO AUTORIZADOR](#)

- **Escenario4 Consumidor está interno on-premise y requiere acceso a capacidades contenerizadas en EKS:**

Este escenario solo aplica para servicios que solo tienen un consumidor o que en su evaluación no amerita ser incluidos en la capa de APIs Gestionadas.

- **Escenario5 Consumidor es Interno en una cuenta de AWS y requiere acceso a capacidades contenerizadas en EKS a través del API Connect:**


Si una API puede ser reusable por múltiples consumidores se debe validar si es candidata a ser gestionada en el API Connect, cumpliendo los lineamientos de integración empresarial de la organización. En este caso son aquellos servicios que están expuestos en el EKS en AWS y por lo tanto deben ser expuestos en el API GW de API Connect que se encuentra en AWS.

 **Nota:** Este escenario es prácticamente igual al escenario3. Este escenario se documenta de manera independiente para guardar congruencia con la documentación oficial del equipo de integración del banco.

- **Escenario6 Consumidor es Interno en una cuenta de AWS y requiere acceso a capacidades contenerizadas en EKS:**

Este escenario solo aplica para servicios que solo tienen un consumidor o que en su evaluación no amerita ser incluidos en la capa de APIs Gestionadas. En este caso la capa consumidora se encuentra en AWS y la capa proveedora se encuentra en el EKS.

Existe un caso de uso documentado en la wiki llamado [INTEGRACION DE APLICACIONES INTERNAS EN CUENTAS AWS DIFERENTES UTILIZANDO ISTIO COMO AUTORIZADOR](#)

 **Nota:** Este escenario es prácticamente igual al escenario4. Este escenario se documenta de manera independiente para guardar congruencia con la documentación oficial del equipo de integración del banco.


- **Escenario7 Consumidor es un contenedor de un Namespace de un EKS y requiere acceso a otro Namespace en el mismo EKS a través del API Connect:**

Este escenario aplica cuando se tiene una aplicación proveedora que expone sus APIs a través de la capa de APIs Gestionadas y hay un consumidor que se encuentra en el mismo EKS. Siguiendo las estrategia de Integración Empresarial y garantizando el desacople de los servicios, así estén en el mismo EKS se debe salir del contexto del EKS a consumir las APIs en el API Connect.

- **Escenario8 Consumidor es un contenedor de un Namespace de un EKS y requiere acceso a otro Namespace en el mismo EKS:**

Este escenario solo aplica cuando la capa proveedora ofrece servicios que ante una evaluación con el área de Integración se define que no son APIs reusables y por lo tanto no son gestionadas. En este caso el consumo se puede hacer directo entre las aplicaciones dentro del mismo EKS.


Existe un caso de uso documentado en la wiki llamado [INTEGRACION DE APLICACIONES QUE ESTAN EN LA MISMA CUENTA Y EL MISMO CLUSTER EKS](#)

 **Nota:** Este caso es excepcional y se debe revisar en detalle con el área de integración del banco antes de realizar cualquier implementación, dado de la conexión directa entre Pods genera una interdependencia entre aplicaciones, y esto es un caso NO deseado.

- **Escenario9 Consumidor es un contenedor de un Namespace de un EKS y requiere acceso a otro Namespace de otro EKS a través del API Connect:**

Este escenario aplica cuando se tiene una aplicación proveedora que expone sus APIs a través de la capa de APIs Gestionadas y hay un consumidor que se encuentra en un EKS diferente. Siguiendo las estrategia

de Integración Empresarial y garantizando el desacople de los servicios, se debe realizar el consumo de las APIs a través del API Connect

 **Nota:** Este escenario es prácticamente igual al escenario7. Este escenario se documenta de manera independiente para guardar congruencia con la documentación oficial del equipo de integración del banco.

- **Escenario10 Consumidor es un contenedor de un Namespace de un EKS y requiere acceso a otro Namespace de otro EKS:**

Este escenario solo aplica cuando la capa proveedora ofrece servicios que ante una evaluación con el área de Integración se define que no son APIs reusables y por lo tanto no son gestionadas. En este caso el consumo se puede hacer directo pasando por los egress e ingress de cada EKS.

- **Escenario11 Consumidor es un Contenedor de un namespace y requiere acceso a un Contenedor del mismo namespace:**

Cuando son contenedores del mismo dominio o namespace, se puede hacer el consumo directo sin pasar por las capas de integración.

- **Escenario12 Escenario temporal de autenticación de conexiones entre API Connect y los servicios expuestos en EKS:**

Este escenario es temporal mientras se realizan pruebas de integración entre el API Connect de IBM y el Ingress de Istio mediante Open ID Connect (OIDC).