Istio mTLS eks

Last updated by | Bibiana Alejandra Arismendi Mallama | 24 de jul de 2024 at 2:59 p. m. COT

⚠️ **Importante:**
Todos los nameSpaces en el entorno de EKS de Bancolombia deben tener habilitado el sidecard de istio (istio-injection=enabled). Existen casos muy puntuales para productos de caja negra que no soportan integración con un serviceMesh, en ese caso se debe seguir el siguiente proceso para solicitar la excepción.

Contents

Istio

GENERALIDADES

Introducción a Istio

¿Qué es un service mesh?

Un service mesh es una capa de infraestructura dedicada que puede agregar a sus aplicaciones. Le permite agregar de forma transparente capacidades como la observabilidad, la gestión del tráfico y la seguridad, sin agregarlas a su propio código. El término "malla de servicio" describe tanto el tipo de software que utiliza para implementar este patrón como el dominio de red o seguridad que se crea cuando utiliza ese software.

¿Qué es istio?

Istio es una red de servicios de código abierto que se superpone de forma transparente a las aplicaciones distribuidas existentes. Las potentes funciones de Istio brindan una forma uniforme y más eficiente de proteger, conectar y monitorear los servicios. Istio es el camino hacia el equilibrio de carga, la autenticación de servicio a servicio y la supervisión, con pocos o ningún cambio de código de servicio. Su poderoso plano de control trae características vitales, que incluyen:

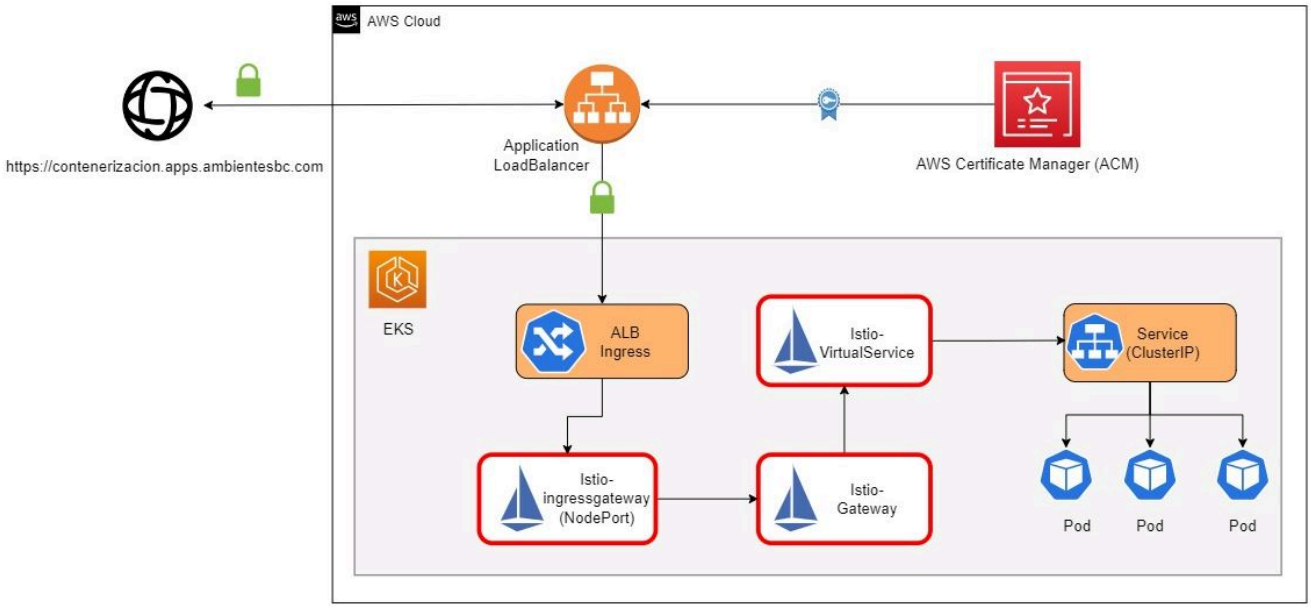
Fuente de consulta

LINEAMIENTOS Y MEJORES PRÁCTICAS

- Todos los recursos en istio deben estar etiquetados de acuerdo con Estándar de etiquetas de recursos AWS
- Un Virtual Service siempre va asociado a un Gateway
- La definición del paths del proyecto en el Gateway queda a disposición del proyecto si se hace de manera absoluta o relativa

CASOS DE USO

Arquitectura de istio en Bancolombia



¿CÓMO USAR EL SERVICIO EN BANCOLOMBIA?

Para el uso de istio se deben desplegar los recursos gateway, virtual service, destination rule (es opcional) puede hacer uso de istio y sus componentes para canary.

```
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
  name: #{app-gateway}#
  namespace: #{namespace}#
  labels:
    app.bancolombia.com.co/env: #{env}#
    app.bancolombia.com.co/cost-center: #{cost-center}#
    app.bancolombia.com.co/application-code: #{application-code}#
    app.bancolombia.com.co/project: #{project-name}#
    app.bancolombia.com.co/pmo: #{pmo}#
spec:
  selector:
    istio: ingressgateway # use istio default controller
  servers:
    - port:
        number: #{service-port}#
        name: #{service}#-port
        protocol: #{app-gateway-protocol}#
      hosts:
        - '#{gateway-internal-host}#'
---
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: #{service}#-vs
  namespace: #{namespace}#
  labels:
    app.bancolombia.com.co/env: #{env}#
    app.bancolombia.com.co/cost-center: #{cost-center}#
    app.bancolombia.com.co/application-code: #{application-code}#
    app.bancolombia.com.co/project: #{project-name}#
    app.bancolombia.com.co/pmo: #{pmo}#
spec:
  hosts:
    - '#{gateway-internal-host}#'
  gateways:
    - #{app-gateway}#
  http:
    - match:
        - uri:
            prefix: '#{path-prefix}#'
      rewrite:
        uri: '#{path-rewrite}#'
      route:
        - destination:
            host: #{service}#
            port:
              number: #{service-port}#
```

Sidecar Istio

El sidecar de istio es un contenedor adicional que sirve para abstraer métricas de su contenedor principal y aprovechar otras funcionalidades sin afectar su aplicación.

Para hacer uso de este sidecar debe cumplir una única regla:

- El namespace debe tener asignado el siguiente label `istio-injection=enabled`

Verá reflejado en el campo de READY 2/2 esto quiere decir que existen dos contenedores dentro de un pod, véalo gráficamente.

```
neytor@MacBook-Pro-de-Yonier git-bco % kubectl describe ns default
Name:         default
Labels:       istio-injection=enabled
              kubernetes.io/metadata.name=default
Annotations:  <none>
Status:       Active

No resource quota.

No LimitRange resource.
neytor@MacBook-Pro-de-Yonier git-bco % kubectl get pods
NAME                                        READY   STATUS    RESTARTS   AGE
skillfullers-validator-dp-59bcb4479c-j246b   2/2     Running   0          13h
neytor@MacBook-Pro-de-Yonier git-bco % kubectl get pods -o=jsonpath="{.items[*].spec.containers[*].name}"
skillfullers-validator istio-proxy%
neytor@MacBook-Pro-de-Yonier git-bco %
```

> **Nota:** Existe un pipeline operativo donde puede activar o desactivar a demanda istio-injection, una vez hecho esto debe reiniciar los pods, Enlace pipeline documentación

¿Qué es mTLS?

mTLS (mutual Transport Layer Security) en Istio se refiere a la autenticación y el cifrado basado en certificados para la comunicación segura entre servicios dentro de un entorno de microservicios gestionado por Istio.

mTLS implica el uso de certificados TLS tanto en el lado del cliente como en el lado del servidor para establecer una comunicación segura y autenticada entre los servicios. Cada servicio tiene su propio certificado y clave privada, y se establece una confianza mutua entre los servicios mediante el intercambio y la validación de certificados (el sidecar de istio se encarga de esto).

CA interna de Istio: Istio también proporciona su propio sistema de CA interna, conocido como Istio Citadel, que puede ser utilizado para la administración de certificados.

IMPORTANTE: El envoy o sidecar es quien facilita la comunicación entre microservicios de manera segura.

Recurso PeerAuthentication

PeerAuthentication define cómo se tunelizará (o no) el tráfico al sidecar.

¿Cuáles son los modos existentes?

Nombre	Descripción
UNSET	Heredar del padre, si tiene uno. De lo contrario, se tratará como PERMISIVO.
DISABLE	La conexión no tiene un túnel
PERMISSIVE	La conexión puede ser texto plano o túnel mTLS.
STRICT	La conexión es un túnel mTLS (se debe presentar TLS con certificado).

Para nuestro caso únicamente usaremos el modo STRICT debido a que si se configura en PERMISSIVE tendríamos el mismo problema "comunicaciones sin cifrar"

Habilitar mTLS

Para habilitar mTLS solo se debe aplicar este manifiesto en el namespace de istio-system y quedará de manera global en el cluster.

```yaml
apiVersion: security.istio.io/v1beta1
kind: PeerAuthentication
metadata:
  name: mtls-global
  namespace: istio-system
spec:
  mtls:
    mode: STRICT
```

¿Sobre la latencia?

Por defecto cuando la comunicación entra por el balanceador la toma istio y es cifrada, por ende no agregaría una latencia que sea evidente una vez aplicado el mTLS

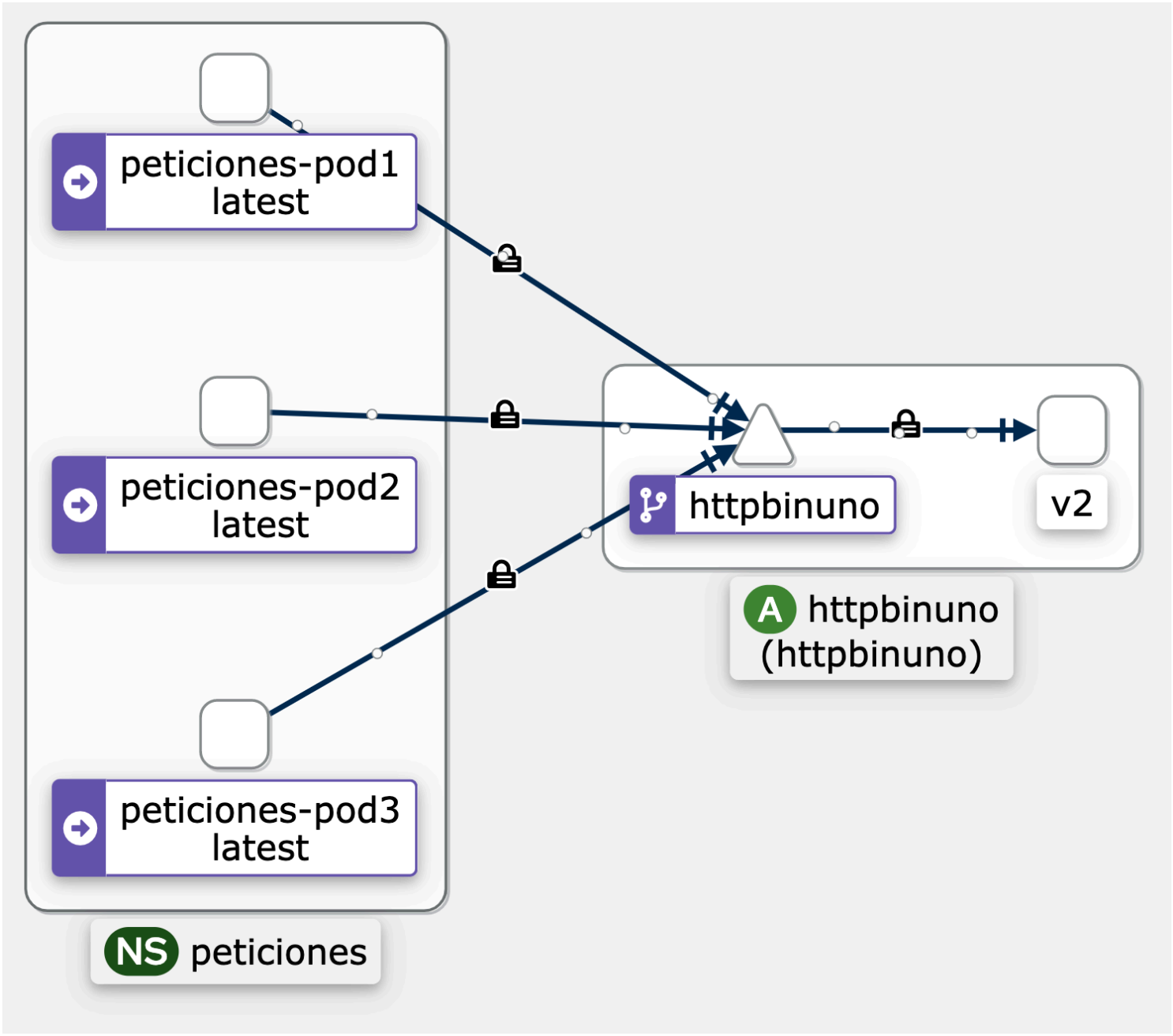
¿Comunicación entre namespaces?

Cuando se quiere comunicar el namespace A con el namespace B y se tiene mTLS habilitado es una comunicación segura que implica el uso de certificados de ambos lados (para el ejemplo namespace A y B), el hecho de cifrar y descifrar los mensajes puede agregar latencia a las aplicaciones con este modo de comunicación.

IMPORTANTE: Es de aclarar que desde arquitectura las comunicaciones se deben dar a través del balanceador de carga, por ende no existiría dicha latencia, se tiene configurado el recurso NETWORK POLICY en todos los namespaces que impiden la comunicación directa entre namespace A y B para el ejemplo.

Ejemplo de laboratorios con mTLS y tráfico cifrado

Gráfica



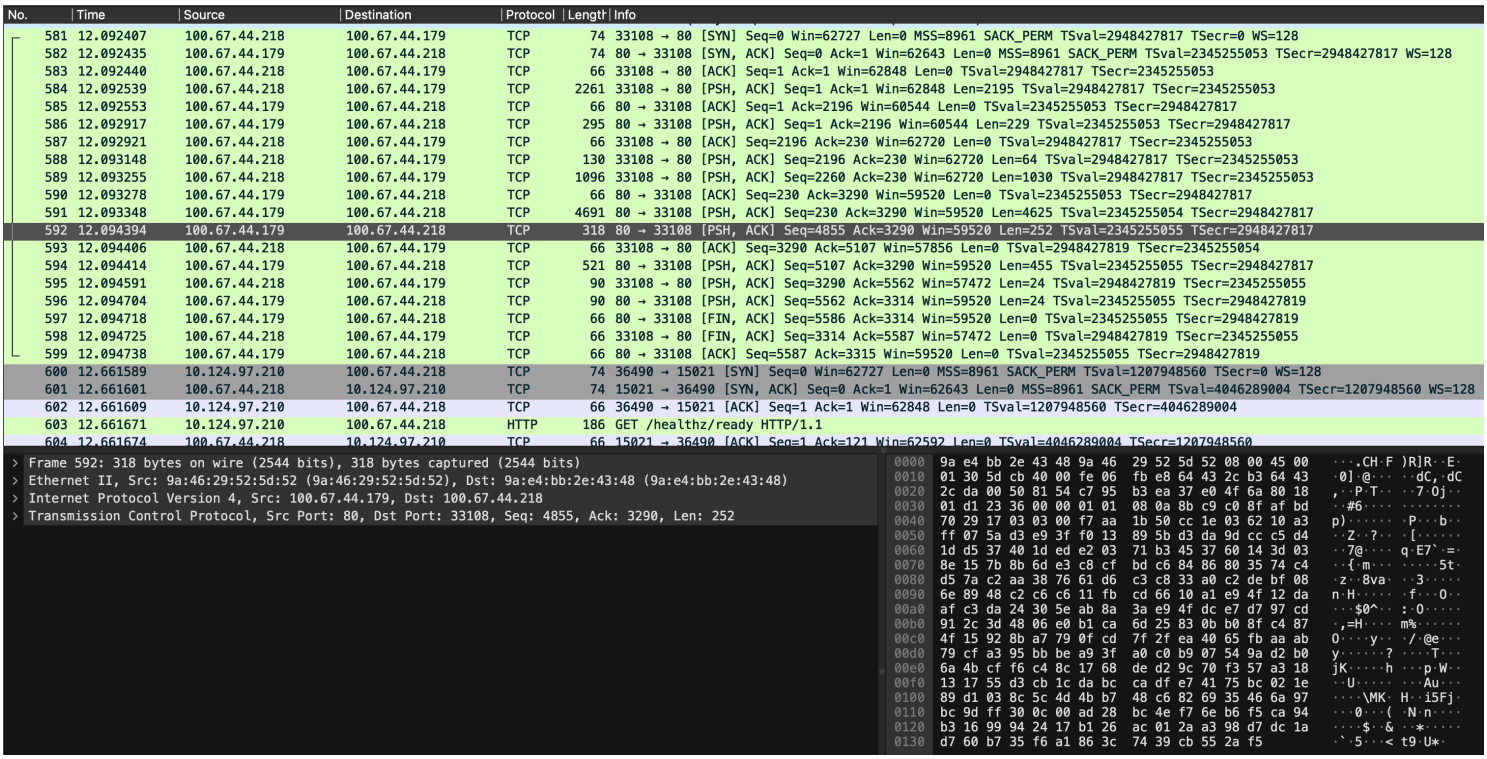
Overview | Traffic | Inbound Metrics | Traces

Inbound Traffic

Status	Name	Rate	Percent Success	Protocol	Actions
	W peticiones-pod1	635.44rps	100.0%	TCP	View metrics
	W peticiones-pod3	677.73rps	100.0%	TCP	View metrics
	W peticiones-pod2	665.00rps	100.0%	TCP	View metrics

Outbound Traffic

Status	Name	Rate	Percent Success	100 % of mTLS traffic	Actions
	W httpbinuno	1978.18rps	100.0%	TCP	View metrics



CONTROLES DE SEGURIDAD

Para mayor información sobre los controles de seguridad aplicables a este servicio, consulte CODI

Si tiene problemas para acceder a la herramienta CODI, por favor consulte el siguiente enlace

ENLACES DE INTERES

- Istio
- PeerAuthentication
- mTLS
- Acuerdos de nivel de serivcios (SLA)

CONTROL DE VERSIONES

Fecha	Descripción	Realizó	Aprobó
2022/10/20	Versión inicial	**@Yonier Manuel Asprilla Gomez** **@Mauricio Bohorquez Orozco**	
2023/10/10	Istio mTLS	**@Yonier Manuel Asprilla Gomez**	