

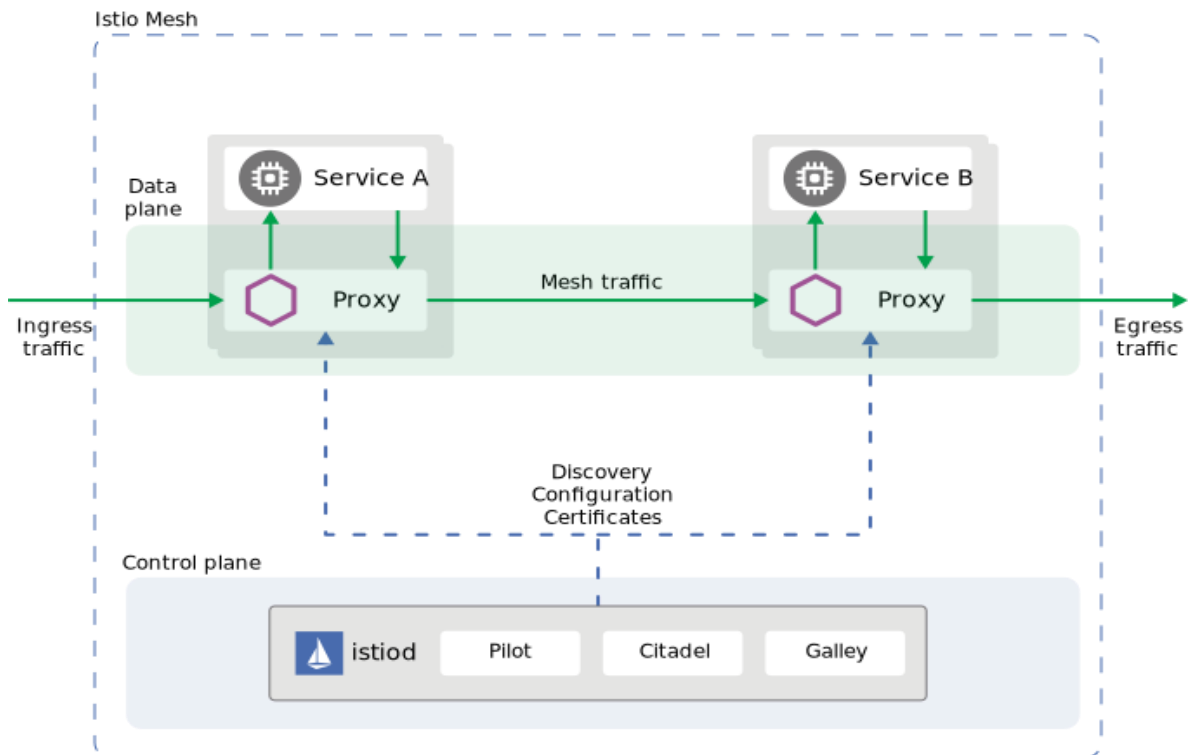
Docente: EDWIN MARAVI PÉREZ

POR: CARRION HUACANI GEAN CARLO

ACTIVIDAD ASINCRÓNICA N° 14

ISTIO

Istio es una malla de servicios (es decir, una capa de redes de servicios modernizada) que ofrece una manera transparente e independiente de cualquier lenguaje de automatizar las funciones de red de una aplicación de forma flexible y sencilla. Es una solución muy popular para gestionar los diferentes microservicios que conforman una aplicación nativa de la nube. La malla de servicios de Istio también es compatible con las formas de comunicarse y compartir datos entre ellos que utilizan estos microservicios.



¿Cómo implementarlo?

La clave para comprender Istio y su arquitectura es conocer Envoy y Kubernetes. No se trata de comparar Istio con Envoy o con Kubernetes. A menudo, las tres soluciones se usan a la vez para conseguir que el entorno en contenedores basado en microservicios tenga un mejor rendimiento. Por ejemplo, las mallas de servicio como Istio cuentan con un plano de control y otro de datos. Istio utiliza una versión ampliada de Envoy como plano de datos. Después, Envoy gestiona todo el tráfico entrante y saliente en la malla de servicios de Istio.

Por otra parte, Kubernetes es una plataforma de código abierto que automatiza y orquesta las tareas de despliegue y escalado de aplicaciones en contenedores para que puedas librarte de gran parte de los procesos manuales. Aunque Istio no depende de ninguna plataforma, muchos desarrolladores lo utilizan junto con Kubernetes.

Istio puede usarse con cualquier plataforma y está diseñado para ejecutarse en diversos entornos: Cloud, On-premise, Kubernetes, Mesos

Utilización de Istio

Inyectar pods en el mallado de servicios

Para integrar una aplicación en el servicio mesh, existen dos soluciones:

- El uso de anotaciones (en el espacio de nombres o en el pod).
- El uso del comando `istioctl`.

a. Instalar istioctl

El comando `istioctl` es un binario que permite realizar determinadas operaciones en el sistema Istio. La instalación se puede hacer usando el comando `arkade` siguiente:

```
| $ arkade get istioctl
```

En caso de ausencia de Arkade, su instalación se hace con el siguiente comando:

```
| $ curl -L https://git.io/getLatestIstio | sh -
```

El comando debería devolver instrucciones como la siguiente:

```
Add /home/yannig/istio-1.13.2/bin to your path; e.g copy paste in
your shell and/or ~/.profile:
export PATH="$PATH:/home/yannig/istio-1.13.2/bin"
```

Añada la línea `export PATH` al archivo de perfil del usuario (`~/.zshrc` para `zsh` o `~/.bashrc` para `bash`). Seguidamente, reinicie su sesión para tener en cuenta el cambio.

b. Inyectar el sidecar con istioctl

Una vez completada la modificación, la inyección de declaraciones se hace con el comando `istioctl` seguido de estas opciones:

- La opción `kube-inject`.
- La opción `-f` seguida de los recursos en los que inyectar la configuración de Istio.

A continuación, se muestra un ejemplo de ejecución de un archivo de despliegue:

```
| $ istioctl kube-inject -f deployment.yaml
```

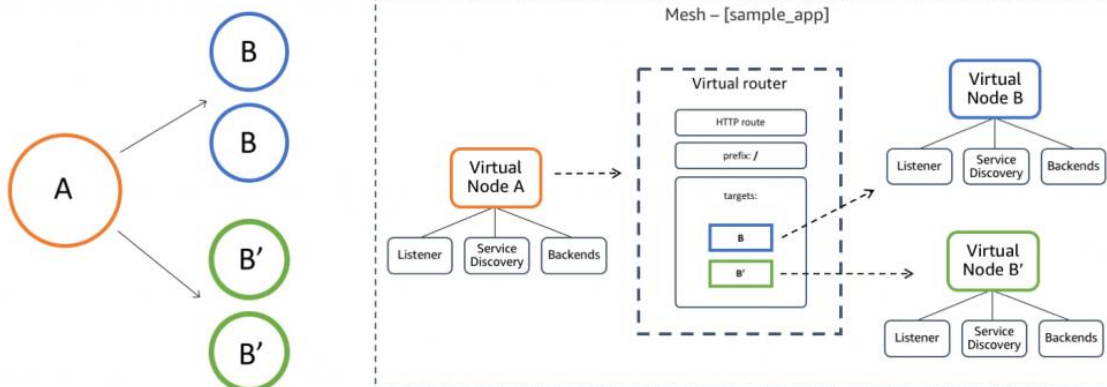
Tal como está, el comando solo muestra el contenido del archivo.

¿Qué beneficios nos trae?

- **Consigue una red de servicios uniforme:** Los operadores de red pueden gestionar las redes de todos sus servicios de forma coherente sin necesidad de sobrecargar a los desarrollers.
- **Protege los servicios con las ventajas de Istio:** Tus operadores de seguridad pueden implementar fácilmente medidas de seguridad entre servicios, como funciones de autenticación, autorización y encriptado.
- **Mejora el rendimiento de las aplicaciones:** Implementa prácticas recomendadas (como los lanzamientos Canary) y mejora la visibilidad de tus aplicaciones para identificar dónde tienes que concentrar tus esfuerzos para mejorar el rendimiento.

Comparar ISTIO con otro sistema que realice las mismas funcionalidades.

AWS App Mesh es un servicio administrado que brinda soporte de comunicación de servicios y redes a nivel de aplicación para la arquitectura de microservicios. También utiliza el proxy Envoy, lo que lo hace compatible con múltiples enfoques de soluciones. Es compatible con AWS Fargate , Amazon EC2 , Amazon ECS , Amazon EKS y Kubernetes que se ejecutan en AWS.



COMPARACIÓN DE CARACTERÍSTICAS

Si bien no existe una bala de plata para la comparación, depende del contexto del sistema y la aplicabilidad para cumplir con los requisitos funcionales y no funcionales. La siguiente lista incluye todas las características clave requeridas para el sistema bajo consideración como parte de nuestra investigación con Alto y Bajo o Sí y No como medidas relativas:

Rasgo	AWS APP MESH	istio	Notas y observaciones clave
Costo de operaciones	Bajo	Alto	Istio proporciona un plano de control autogestionado y tiene una sobrecarga operativa si no utiliza un servicio gestionado.
Portabilidad	Bajo	Alto	Istio está diseñado para Kubernetes y también es compatible con máquinas virtuales y es compatible con SMI (a través del adaptador). Se puede integrar con K8 autogestionados, EKS, GKE, etc. Soporte de primera clase para Google Cloud (Istio en GKE)
Flexibilidad	Bajo	Alto	Istio es App Mesh extensible y neutral para el proveedor: soporte de primera clase para EKS, ECS, Fargate. Sin embargo, incluso con Kubernetes en EC2, necesita usar AWS
Facilidad de configuración (con EKS)	Medio	Alto	Istio : configurar EKS requiere una curva de aprendizaje y una comprensión más profunda de Istio. Haga clic aquí para ver los detalles. Malla de aplicaciones : con EKS, la configuración es fácil, especialmente debido al soporte y la integración de AWS.

Observabilidad	Medio	Alto	<p>Istio: el seguimiento mediante la adición de un encabezado de solicitud está disponible de forma inmediata. Haga clic aquí para ver los detalles.</p> <p>App Mesh : el código de la aplicación debe actualizarse para permitir el seguimiento.</p>
Soporte de Kubernetes	Bajo	Alto	Istio se integra de forma nativa con Kubernetes
Compatibilidad con varios clústeres	Alto	Sí	<p>AWS App Mesh : sobre el uso de App Mesh en la configuración de varias cuentas.</p> <p>Istio : para obtener información sobre la configuración de varios clústeres con Istio.</p>
Enrutamiento ponderado	Sí	Sí	Útil para implementaciones A/B, Canary, etc.
Limitación de velocidad	No	Sí	Característica de App Mesh planificada en la hoja de ruta de App Mesh
Gestión de fallos	Parcial	Sí	<p>Ambos admiten el tiempo de espera, vuelva a intentar los disyuntores. Sin embargo, Istio también admite fluctuaciones variables entre reintentos, comprobaciones de estado e inyección de fallos.</p> <p>App Mesh no permite el uso de códigos de error personalizados para la configuración de reintentos y solo aplica la política predeterminada.</p>
Duplicación de tráfico	No	Sí	Istio permite la duplicación de tráfico actualmente, mientras que App Mesh no tiene esta función en este momento.
Seguridad: mTLS y autenticación	Sí	Sí	<p>Ambos admiten mTLS. Sin embargo, Istio ofrece un control más granular sobre la configuración de mTLS, incluida la transferencia automática de certificados. Además, Istio permite solicitar autenticación y tiene integración disponible con Auth0, Keycloak, etc.</p> <p>Haga clic aquí para obtener más detalles sobre Istio.</p> <p>App Mesh tiene integración con IAM, lo que es una ventaja para las cargas de trabajo y la configuración de AWS.</p>