

SERVICE MESH EN K8S CON ISTIO

Sergio García Ojeda

Administración de Sistemas Informáticos en Red



¿Qué es Kubernetes?

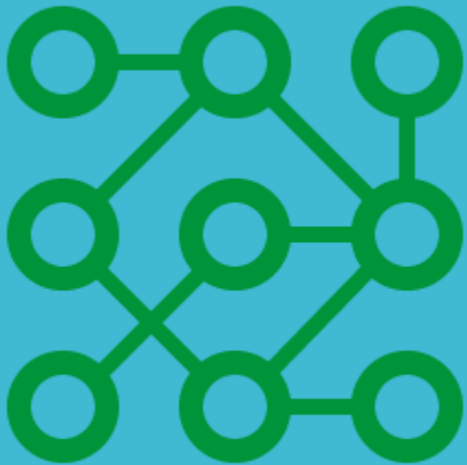
- Sistema Open Source creado por Google.
- Orquestación de Contenedores.
- Ideal para Microservicios.

Nuestro Escenario.

- 3 maquinas:
 - Nieve. Maquina Debian. 4GB RAM. Master
 - Arya. Maquina Debian. 2GB RAM.
 - Sansa. Maquina Debian. 2GB RAM.
- Cluster de Kubernetes con Kubeadm

Service Mesh

¿Qué es?



- Capa de infraestructura entre el servicio y la red.
- Controla la comunicación entre los servicios y la red.
- Seguridad, rapidez y confiabilidad.
- Implantación mediante "sidecar proxy".

Service Mesh. Ventajas y Desventajas

- Ventajas:



- Simplifica la comunicación entre los microservicios.
- Facilita la localización de fallos en la comunicación.
- Permite cifrar, autenticar y autorizar.
- Administración de las redes a un nivel más alto.

Service Mesh. Ventajas y Desventajas

- Desventajas:

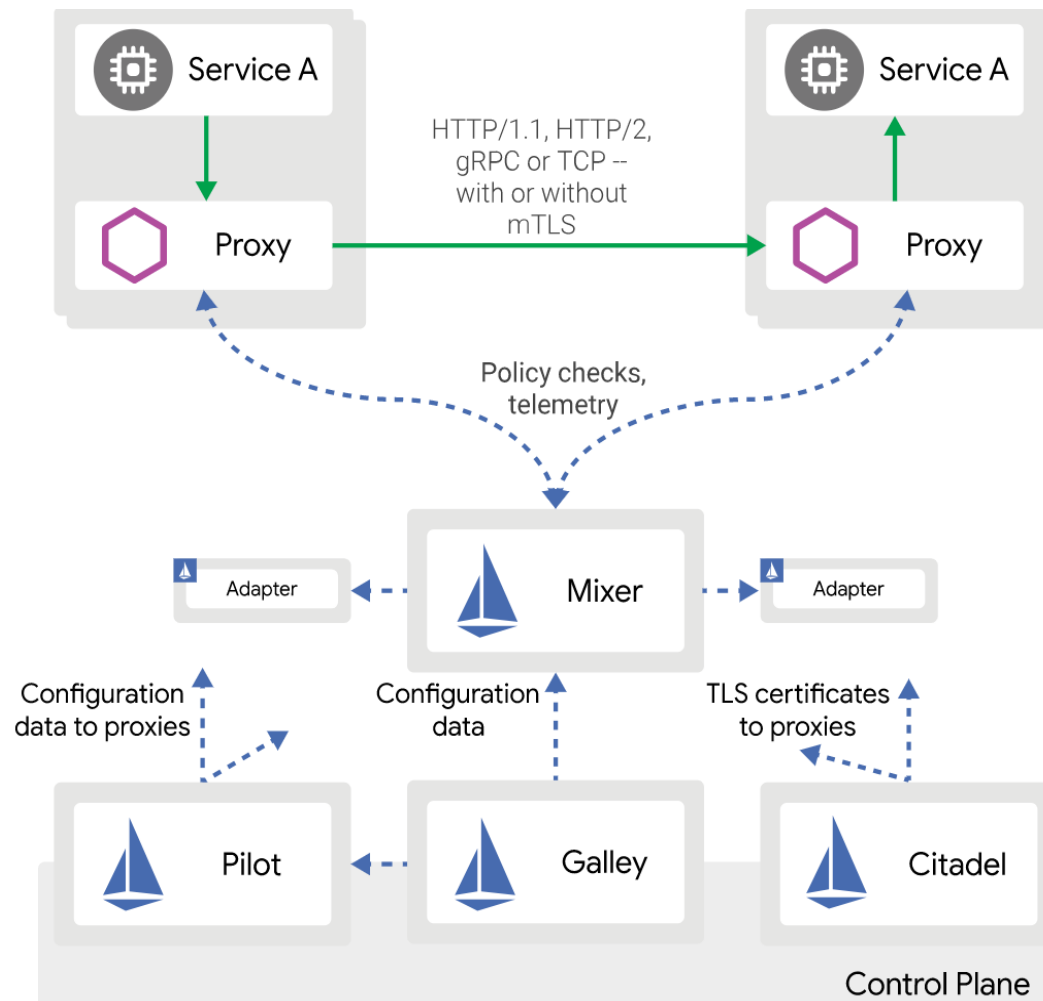


- El tiempo de ejecución de las instancias aumenta exponencialmente.
- Añade un paso más en la comunicación. Paso por el proxy.

Istio

- Implementación de Service Mesh en Kubernetes.
- Facilidad de crear una red de servicios desplegados con balanceo de carga, autenticación, monitorización, etc.
- No se modifica el Código.

Estructura de Istio



<https://istio.io/docs/concepts/what-is-istio/arch.svg>

Istio. Objetivos.



- Transparencia Máxima.
- Incrementabilidad.
- Portabilidad.
- Uniformidad de Políticas.

Funcionalidad. Gestión del tráfico



- Istio toma el control del tráfico.
- Pilot se encarga de las configuraciones en los proxys.
- Muy útil a la hora de filtrar tráfico y balancear carga.

Funcionalidad.
Seguridad.



- Red de Confianza Nula.
- Defensa en profundidad.
- Cifrado y Certificación.
- Citadel gestiona claves y certificados, Pilot distribuye las políticas y Mixer gestiona autenticación y autorización.

Funcionalidad. Políticas.



- Mixer es el encargado de establecer políticas.
- Abstracción del Backend.
- Control en todos los elementos de la mesh.
- Deshabilitada por defecto

Funcionalidad. Telemetría.



- Mixer es el encargado de establecer la telemetría.
- Mixer realiza una llamada antes y después de cada solicitud.
- Almacenamiento en caché. Pocas llamadas a mixer de forma saliente.
- Deshabilitado por defecto.

Istio. Instalación.

- Descargamos la última versión en el Master.

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

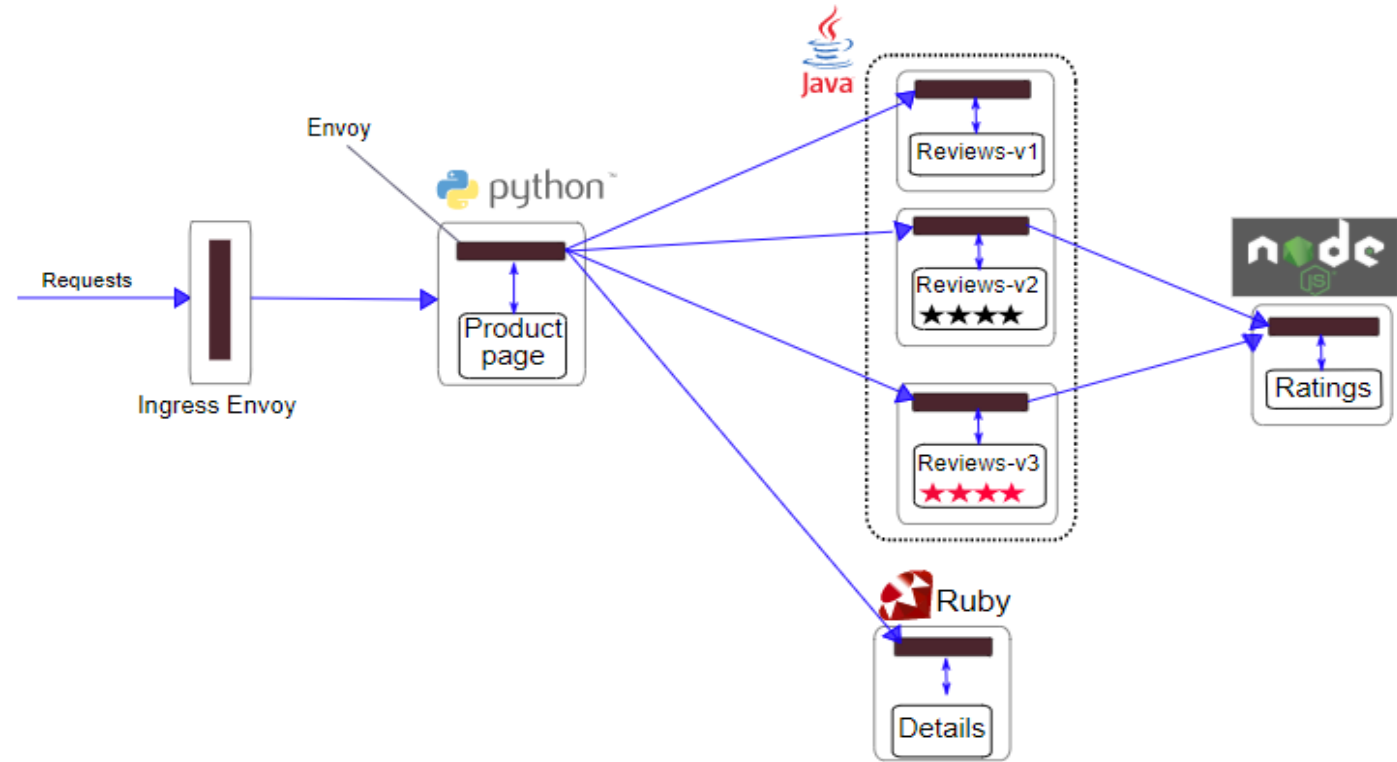
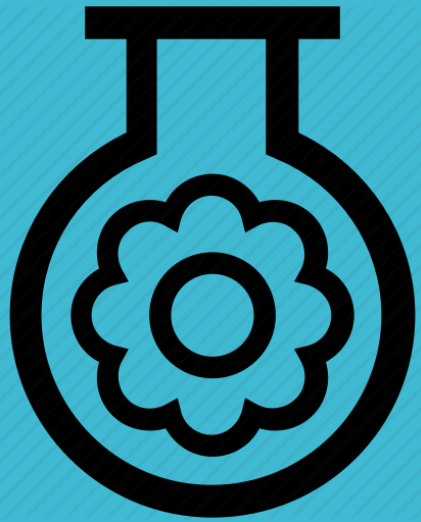
- Entramos en el directorio que contiene los instaladores.

```
cd istio-1.1.7/install/kubernetes/Helm/istio-init/files
```

- Ejecutamos cada instalador con kubectl.

```
kubectl apply -f crd-10.yaml  
kubectl apply -f crd-11.yaml  
kubectl apply -f crd-certmanager-10.yaml  
kubectl apply -f crd-certmanager-11.yaml
```

Escenario de Pruebas.



<https://istio.io/docs/examples/bookinfo/withistio.svg>

Pruebas de Funcionamiento.



- Blue Green deployment.
- Canary Release.
- Request Timeouts.

Conclusiones y Trabajos Futuros



- Futuro de Kubernetes.
- Pequeña Relentización en la comunicación (prácticamente nula).
- Herramienta muy potente en situaciones concretas.
- Posibles Trabajos futuros. Servicio de Monitorización en una Mesh.

Gracias.
¿Alguna
Pregunta?

