Service Mesh in K8s

Kavan Dalwadi, Om Dhimar

# What is Service Mesh:

A service mesh is a dedicated infrastructure layer that you can add to your applications. It allows you to transparently add capabilities like observability, traffic management, and security, without adding them to your own code. The term “service mesh” describes both the type of software you use to implement this pattern, and the security or network domain that is created when you use that software.

# 

# Installation Commands:

[Download Istio](#_3hy8rkwzatey)

istioctl version

istioctl profile list

istioctl profile dump demo > istio\_demo.yaml

Istioctl apply -f istio\_demo.yaml

Or

istioctl install –set profile=demo

istioctl verify-install

Or

istioctl manifest generate –set profile=demo > k8s\_istio\_demo.yaml

kubectl apply -f k8s\_istio\_demo.yaml

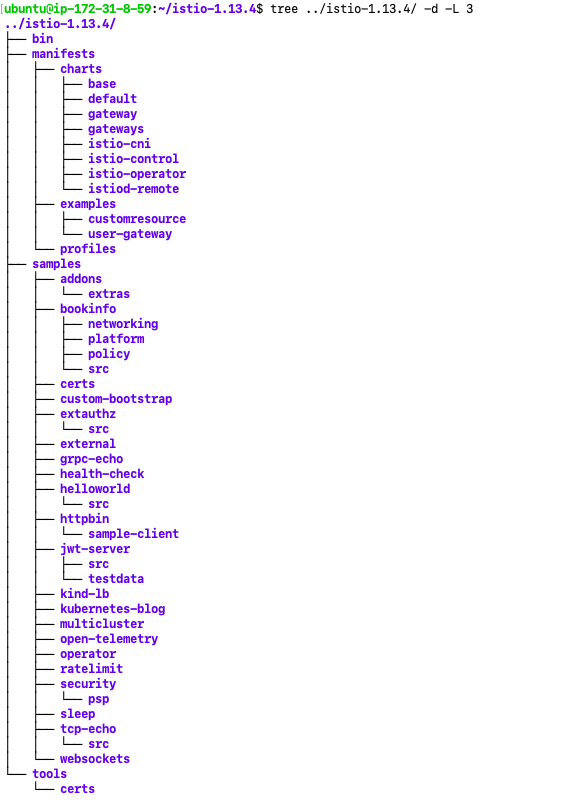
istioctl verify-install -f k8s\_istio\_demo.yaml

kubectl label namespace default istio-injection=enabled

kubectl get all -n istio-system

# Uninstall:

istio x uninstall –purge



# Traffic Management:

<https://istio.io/latest/docs/reference/config/networking/>

<https://istio.io/latest/docs/reference/config/networking/destination-rule/#Subset>

# Security:

<https://istio.io/latest/docs/reference/config/security/>

# Canary using istio:

<https://istio.io/latest/blog/2017/0.1-canary/>

# Download Istio service mesh:

<https://istio.io/latest/docs/setup/getting-started/>

# Circuit breaker:

<https://dzone.com/articles/istio-circuit-breaker-with-outlier-detection>

**Change istio profile config:**

cd /home/ubuntu/istio-1.13.4/manifests/profiles

App yaml config:

apiVersion: v1

kind: Service

metadata:

name: web

labels:

app: web

spec:

type: NodePort

selector:

app: web

ports:

- port: 80

nodePort: 30100

targetPort: 80

---

apiVersion: apps/v1

kind: Deployment

metadata:

name: apache

spec:

selector:

matchLabels:

app: web

version: v1

istio: ingressgateway

replicas: 1

template:

metadata:

labels:

app: web

version: v1

istio: ingressgateway

spec:

containers:

- name: apache

imagePullPolicy: Always

image: httpd:alpine

ports:

- containerPort: 80

---

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx

spec:

selector:

matchLabels:

app: web

version: v2

istio: ingressgateway

replicas: 1

template:

metadata:

labels:

app: web

version: v2

istio: ingressgateway

spec:

containers:

- name: nginx

imagePullPolicy: Always

image: nginx:latest

ports:

- containerPort: 80

—

Istio yaml config:

apiVersion: networking.istio.io/v1alpha3

kind: Gateway

metadata:

name: app-gw

spec:

selector:

istio: ingressgateway

servers:

- port:

number: 80

name: http

protocol: HTTP

hosts:

- "\*"

---

apiVersion: networking.istio.io/v1alpha3

kind: DestinationRule

metadata:

name: web

spec:

host: "\*"

subsets:

- name: v1

labels:

version: v1

- name: v2

labels:

version: v2

---

apiVersion: networking.istio.io/v1alpha3

kind: VirtualService

metadata:

name: web

spec:

hosts:

- "\*"

gateways:

- app-gw

http:

- route:

- destination:

host: web

subset: v1

weight: 50

- destination:

host: web

subset: v2

weight: 50

---