THE LINUX FOUNDATION | Training & Certification

# Lab 5.2. Install an Istio Service Mesh

## Overview

In this lab, you'll be installing a service mesh using Istio.

In the previous labs, you created an Ingress Controller with annotations. One of those annotations was for Istio. Because the configuration is already set, you don't have to do anything other than installing Istio itself because the annotation already exists inside of the Ingress Controller. Below is the Nginx Ingress Controller manifest with the Istio annotation highlighted in orange to remind you of what it looks like. **Please note** that you do **not** have to re-run the configuration as you already did in the previous labs.

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  namespace: emojivoto
  name: ingress-emojivoto
  annotations:
    ingress.kubernetes.io/rewrite-target: /
    nginx.ingress.kubernetes.io/service-upstream: "true"
    linkerd.io/inject: enabled
    kubernetes.io/ingress.class: istio
    consul.hashicorp.com/connect-inject: “true”
spec:
  ingressClassName: nginx
  rules:
    - http:
        paths:
        - pathType: Prefix
          path: "/"
          backend:
```

```
          service:
            name: web-svc
            port:
              number: 80
```

1. Connect to the VM that hosts your Kubernetes clusters.

2. To start the cluster that will contain the Istio service mesh, issue this command:

   ```
   yourname@ubuntu-vm:~$ docker start $(docker ps -a -f
   name=istio-control-plane -q)

   2d1d09fadf21
   ```

3. To switch the **kind** context to the Istio cluster, use this command:

   ```
   yourname@ubuntu-vm:~$ kubectl config use-context kind-istio

   Switched to context "kind-istio".
   ```

4. Download the Istio version 1.6.8 release:

   ```
   yourname@ubuntu-vm:~$ curl -L https://istio.io/downloadIstio | sh
   ```

```
      curl -L https://istio.io/downloadIstio | sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time
Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   101  100   101    0     0   1373      0 --:--:-- --:--:-- --:--:--  1578
100  4926  100  4926    0     0  23399      0 --:--:-- --:--:-- --:--:-- 23399

Downloading istio-1.14.2 from
https://github.com/istio/istio/releases/download/1.14.2/istio-1.14.2-osx-arm64
.tar.gz ...

Istio 1.14.2 Download Complete!

Istio has been successfully downloaded into the istio-1.14.2 folder on your
system.

Next Steps:
See https://istio.io/latest/docs/setup/install/ to add Istio to your
Kubernetes cluster.
```

```
To configure the istioctl client tool for your workstation,
add the /Users/michael/istio-1.14.2/bin directory to your environment path
variable with:
       export PATH="$PATH:/Users/michael/istio-1.14.2/bin"

Begin the Istio pre-installation check by running:
       istioctl x precheck
```

5. Add **istioctl** to your current **PATH** by using the **export PATH** command specified below:

```
yourname@ubuntu-vm:~$ export PATH=$PWD/bin:$PATH

[no output if the command succeeds]
```

6. You should also add the linkerd CLI to the **$PATH** variable for your shell with the command below (assuming you are using **bash** for your shell). This will permanently add it to your **PATH**. Without doing this once, if you log out and log back in, your **PATH** will be replaced and your **linkerd** commands will not work correctly until you manually update your **PATH** as you did in the previous step.

```
yourname@ubuntu-vm:~$ echo "export
PATH=$PATH:$HOME/istio-1.14.2/bin" >> ~/.bashrc

[no output if the command succeeds]
```

7. Install Istio in the cluster with the following command. Note that you may be prompted about proceeding with the install; make sure to press ‘y’ and hit ‘enter’ to continue. Simply hitting ‘enter’ without first pressing ‘y’ will cause the default (capitalized) option, ‘N’, to be chosen and the install will be terminated.

```
yourname@ubuntu-vm:~$ istioctl install \
  --set values.gateways.istio-ingressgateway.enabled=false

This will install the default Istio profile into the cluster.
Proceed? (y/N) y
Detected that your cluster does not support third party JWT
authentication. Falling back to less secure first party
JWT. See
https://istio.io/docs/ops/best-practices/security/#configure-thir
d-party-service-account-tokens for details.
✔ Istio core installed
✔ Istiod installed
✔ Addons installed
✔ Installation complete
```

8. Let’s explore what you just installed! There is a lot of configuration that goes into making sure Istio runs the way it is supposed to. Taking a look at the containers that are installed, Istio is a relatively simple deployment with a single pod running the control plane processes, **`prometheus`** for collecting metrics from the control plane and data plane, and a **`grafana`** instance for viewing those metrics. The following command will generate the equivalent to the **`yaml`** installed above. Running through this file will give you a better sense of Istio’s architecture.

```
yourname@ubuntu-vm:~$ istioctl manifest generate \
  --set values.gateways.istio-ingressgateway.enabled=false
  --set values.global.jwtPolicy=first-party-jwt >
istio-manifest.yaml

[no output if the command succeeds]
```

9. Next, you can install Grafana to work with Istio by capturing output from Istio so you can see it in a dashboard.

```
yourname@ubuntu-vm:~$ kubectl apply -f
https://raw.githubusercontent.com/istio/istio/release-1.14/samples/addons/graf
ana.yaml
```

10. Nginx Ingress is now configured to route to services in your Istio mesh. You need to add the demo application you installed to the mesh. **`istioctl kube-inject`** takes the Kubernetes configuration manifest and outputs it with the service proxy configurations added to any object that deploys applications in Kubernetes (Deployment, Pod, Job, etc.). Pairing this with **`kubectl`** allows you to manually inject service proxies to the application

```
yourname@ubuntu-vm:~$ curl -sL
https://run.linkerd.io/emojivoto.yml \
  | istioctl kube-inject -f - \
  | kubectl apply -f -

namespace/emojivoto unchanged
serviceaccount/emoji unchanged
serviceaccount/voting unchanged
```

```
serviceaccount/web unchanged
service/emoji-svc unchanged
service/voting-svc unchanged
service/web-svc unchanged
deployment.apps/emoji configured
deployment.apps/vote-bot configured
deployment.apps/voting configured
deployment.apps/web configured
```

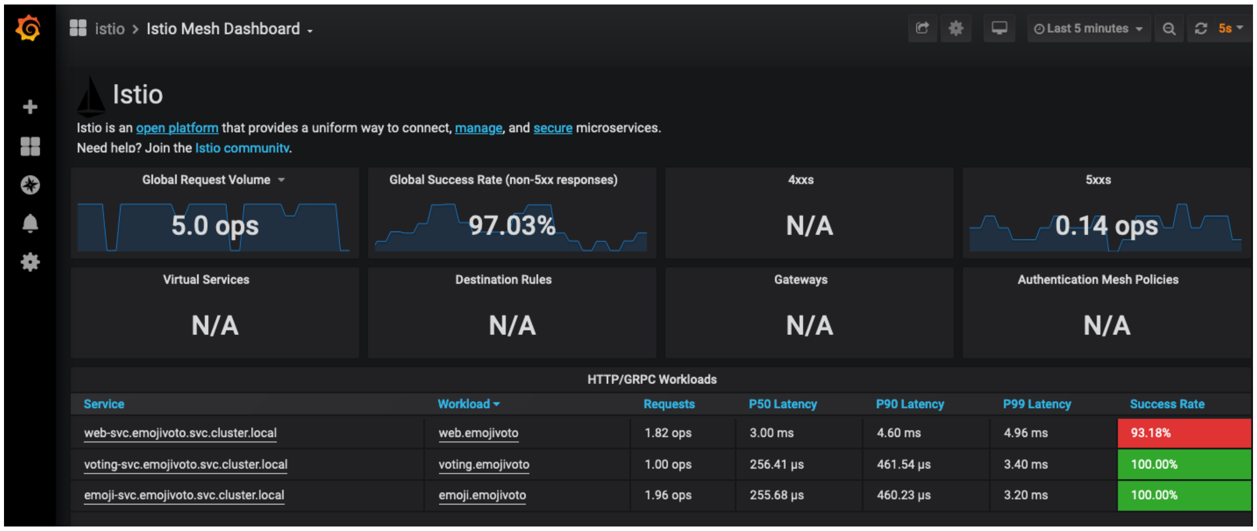
11. The demo application is now meshed in the Istio mesh and ready to receive traffic from Nginx Ingress. You can access this dashboard in two steps.

    a. Expose the Istio dashboard by port-forwarding to the VM:

    ```
    yourname@ubuntu-vm:~$ kubectl port-forward -n istio-system
    svc/grafana 8080:3000

    Forwarding from 127.0.0.1:8080 -> 3000
    Forwarding from [::1]:8080 -> 3000
    ```

    b. Access the default Istio dashboard by opening **`http://localhost:8080`** in a web browser, where you replace **`{{VM_IP_ADDRESS}}`** with your VM's IP address. Through this dashboard, you can get a live look at how services in the mesh are performing. For example, you can see that there appears to be an issue with the **`web-svc.emojivoto.svc.cluster.local`** service. You will learn how to address this in a future lab.



Istio

Istio is an open platform that provides a uniform way to connect, manage, and secure microservices.
Need help? Join the Istio community.

| Global Request Volume | Global Success Rate (non-5xx responses) | 4xxs | 5xxs |
|---|---|---|---|
| 5.0 ops | 97.03% | N/A | 0.14 ops |

| Virtual Services | Destination Rules | Gateways | Authentication Mesh Policies |
|---|---|---|---|
| N/A | N/A | N/A | N/A |

HTTP/GRPC Workloads

| Service | Workload | Requests | P50 Latency | P90 Latency | P99 Latency | Success Rate |
|---|---|---|---|---|---|---|
| web-svc.emojivoto.svc.cluster.local | web.emojivoto | 1.82 ops | 3.00 ms | 4.60 ms | 4.96 ms | 93.18% |
| voting-svc.emojivoto.svc.cluster.local | voting.emojivoto | 1.00 ops | 256.41 μs | 461.54 μs | 3.40 ms | 100.00% |
| emoji-svc.emojivoto.svc.cluster.local | emoji.emojivoto | 1.96 ops | 255.68 μs | 460.23 μs | 3.20 ms | 100.00% |

12. You’ve reached the end of this lab. If you’re not starting the next lab right away, you can stop the Istio cluster by running this command:

```
yourname@ubuntu-vm:~$ docker stop $(docker ps -a -f
name=istio-control-plane -q)

2d1d09fadf21
```