

## Lab 03 - Deploy Istio

cluster-1

Region: us-east-1

Agent: 2.3.7

K8s: 1.25.8+k3s1

Istio: 1.16.4-solo

Configuration

Namespaces ..... 8

Services ..... 8

Gateways ..... 3

Hardware

Nodes ..... 1

Istio Overview

Version: 1.16.4-solo

Revision: 1-16

Root Namespace ..... istio-system

Trust Domain ..... cluster-1

Managed By Gloo ..... No

Health State ..... HEALTHY

cluster-2

Region: us-west-2

Agent: 2.3.7

K8s: 1.25.8+k3s1

Istio: 1.16.4-solo

Configuration

Namespaces ..... 8

Services ..... 6

Gateways ..... 1

Hardware

Nodes ..... 1

Istio Overview

Version: 1.16.4-solo

Revision: 1-16

Root Namespace ..... istio-system

Trust Domain ..... cluster-2

Managed By Gloo ..... No

Health State ..... HEALTHY

Gloo Platform works with Open Source Istio distributions but Solo.io offers a number of different distributions of Istio for different types of environments and use cases such as FIPS, Arm, and distroless. To learn more about the different distributions view [Gloo Istio Distributions](#).

Links:

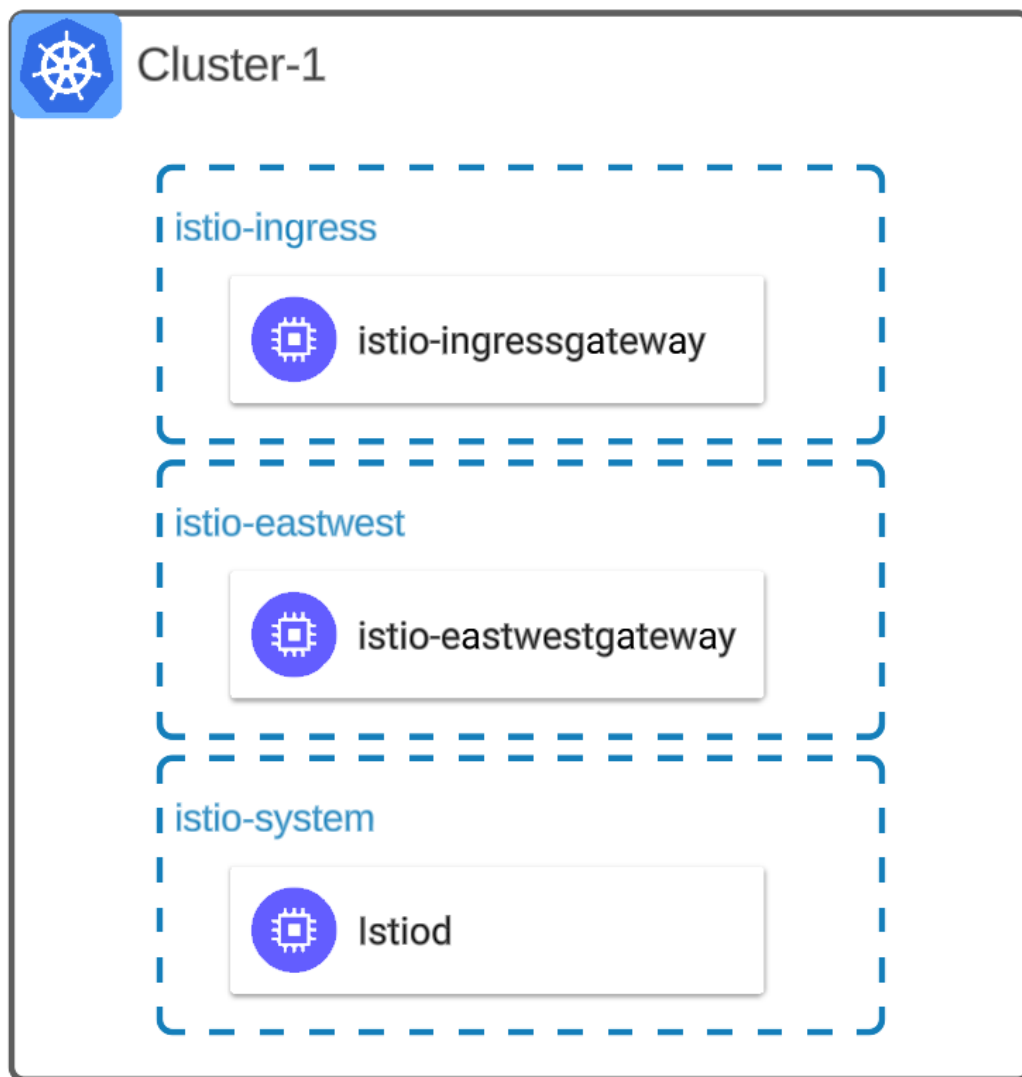
- [Gloo Istio](#)
- [Supported Istio Versions](#)
- [Gloo Platform Managed Istio](#)
- [Manual Istio Installs](#)

Installing Istio via `helm` is the preferred method for installing Istio manually. If you prefer to not manage you Istio installations you can trust that responsibility to Gloo Platform via its Istio [Istio Lifecycle Manager](#) Istio will be installed using `revisions` which is the optimal way to deploy Istio for production environments. It allows for the ability to **canary** Istio components that is safe and will prevent outages.

- First add the Istio helm repositories

```
helm repo add istio https://istio-release.storage.googleapis.com/charts
helm repo update
```

## Install Istio on Cluster: cluster-1



- Create `istio-system`, `istio-eastwest`, `istio-ingress` namespaces

```
kubectl apply --context cluster-1 -f data/namespaces.yaml
```

- Before installing Istio or upgrading the istio/base must be run to install or reconcile the CRDs within the kubernetes cluster.

```
helm upgrade -i istio-base istio/base \
  -n istio-system \
  --version 1.16.4 \
  --set defaultRevision=1-16 \
  --kube-context=cluster-1
```

- Install the `istiod` control plane

```
helm upgrade -i istiod-1-16 istio/istiod \
  --set revision=1-16 \
  --version 1.16.4 \
  --namespace istio-system \
  --kube-context=cluster-1 \
  --set "global.multiCluster.clusterName=cluster-1" \
  --set "meshConfig.trustDomain=cluster-1" \
  -f data/istiod-values.yaml
```

- Install the Istio eastwest gateway which is used for multi-cluster communication between clusters using mTLS.

```
helm upgrade -i istio-eastwestgateway istio/gateway \
  --set revision=1-16 \
  --version 1.16.4 \
  --namespace istio-eastwest \
  --kube-context=cluster-1 \
  -f data/eastwest-values.yaml
```

- Install the Istio ingress gateway with a ClusterIP service type. For best proudction practices and to support multiple revisions a standalone Service will be created to allow easy migration of traffic.

```
helm upgrade -i istio-ingressgateway-1-16 istio/gateway \
  --set revision=1-16 \
  --version 1.16.4 \
  --namespace istio-ingress \
  --kube-context=cluster-1 \
  -f data/ingress-values.yaml
```

- Create the standalone Kubernetes service to sit in front of the Istio ingressgateway.

```
kubectl apply --context cluster-1 -f - <<EOF
apiVersion: v1
kind: Service
```

```

metadata:
  name: istio-ingressgateway
  namespace: istio-ingress
  annotations:
    service.beta.kubernetes.io/aws-load-balancer-type: "external"
    service.beta.kubernetes.io/aws-load-balancer-nlb-target-type: "instance"
    service.beta.kubernetes.io/aws-load-balancer-scheme: "internet-facing"
  labels:
    istio: ingressgateway
    app: gloo-gateway
spec:
  type: LoadBalancer
  selector:
    istio: ingressgateway-1-16
  ports:
    # Port for health checks on path /healthz/ready.
    # For AWS ELBs, this port must be listed first.
    - name: status-port
      port: 15021
      targetPort: 15021
    # main http ingress port
    - port: 80
      targetPort: 8080
      name: http2
    # main https ingress port
    - port: 443
      targetPort: 8443
      name: https
EOF

```

- Verify pods are running

```

kubectl get pods --context cluster-1 -n istio-system
kubectl get pods --no-headers --context cluster-1 -n istio-ingress
kubectl get pods --no-headers --context cluster-1 -n istio-eastwest

```

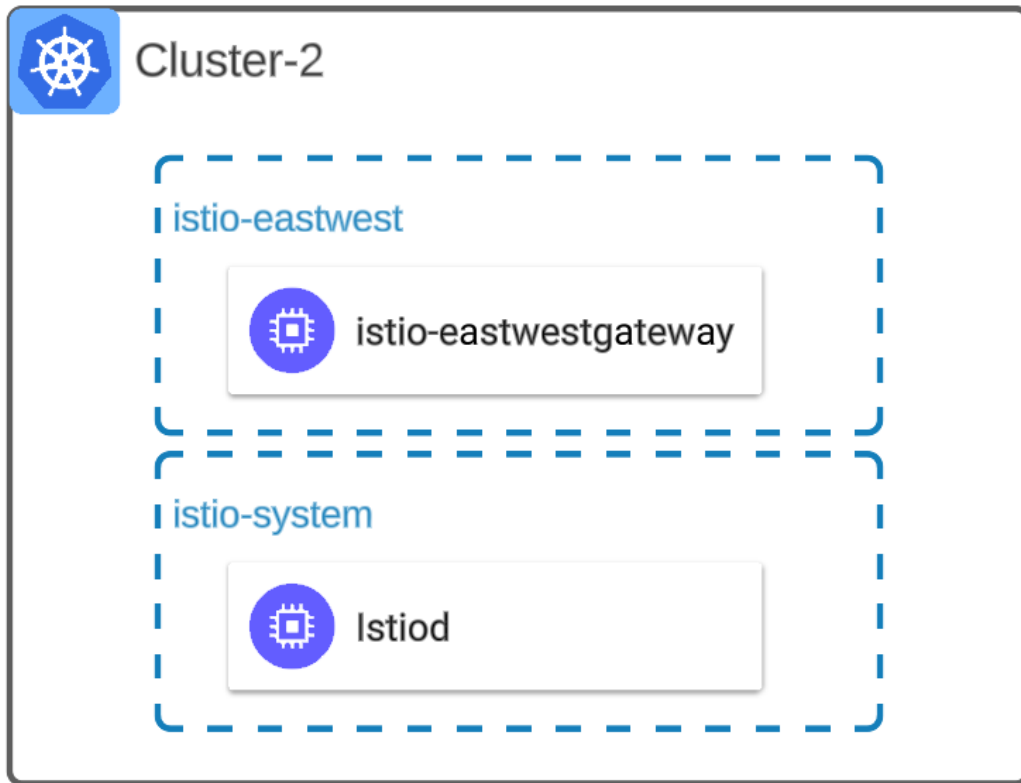
- Verify the load balancer is created`

```

kubectl get service --context cluster-1 -n istio-ingress
kubectl get service --context cluster-1 -n istio-eastwest

```

## Install Istio on Cluster: cluster-2



- Create `istio-system`, `istio-eastwest`, `istio-ingress` namespaces

```
kubectl apply --context cluster-2 -f data/namespaces.yaml
```

- Install the Istio specific CRDs

```
helm upgrade -i istio-base istio/base \
  -n istio-system \
  --version 1.16.4 \
  --set defaultRevision=1-16 \
  --kube-context=cluster-2
```

- Install the `istiod` control plane

```
helm upgrade -i istiod-1-16 istio/istiod \
  --set revision=1-16 \
  --version 1.16.4 \
  --namespace istio-system \
  --kube-context=cluster-2 \
  --set "global.multiCluster.clusterName=cluster-2" \
```

```
--set "meshConfig.trustDomain=cluster-2" \  
-f data/istiod-values.yaml
```

- Install istio eastwest gateway

```
helm upgrade -i istio-eastwestgateway istio/gateway \  
  --set revision=1-16 \  
  --version 1.16.4 \  
  --namespace istio-eastwest \  
  --kube-context=cluster-2 \  
  -f data/eastwest-values.yaml
```

- Verify pods are running

```
kubectl get pods --context cluster-2 -n istio-system  
kubectl get pods --no-headers --context cluster-2 -n istio-ingress  
kubectl get pods --no-headers --context cluster-2 -n istio-eastwest
```

- Verify the load balancer is created`

```
kubectl get service --context cluster-2 -n istio-ingress  
kubectl get service --context cluster-2 -n istio-eastwest
```