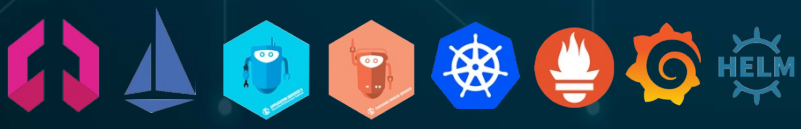
Aspen Mesh Service Mesh

Multi-Cluster Deployment Guide (ISTIO Replicated Control Plane)



Foo-Bang
30 Dec 2019
v1.0

Change History

Revision Number	Date	Comments
1.0	30 December 2019	Initial document.

Table of Content

1 Document Purposes

The purpose of this document is to achieve the following:

- Document steps / getting started to install and onboard Aspen Mesh multi-cluster deployment with Istio Replicated Control Plane deployment model (Kubernetes on private and public cloud).
- Document integration of Aspen Mesh with F5's Container Ingress Services (CIS) as ingress services to Aspen Mesh dashboard.
- Document integration of F5's CIS and AS3 with Istio ingressgateway to deliver external traffic to Istio service mesh.
- Document integration of BIG-IP and CIS to provide inter-cluster application resiliency and global service failover (application failover from GKE public cloud to on-prem Kubernetes private cloud).
- Demonstrate the following example use case with integration to Aspen Mesh to provide observability and visibility.
    - Use Case#1: Single cluster service mesh with Bookinfo sample application.
    - Use Case#2: Multi-cluster service mesh with Bookinfo sample application (6 individual microservices distributed across two independent on-prem Kubernetes with different Kubernetes version).
    - Use Case#3: Multi-cluster service mesh with Google hipster sample application (10 individual microservices distributed across 3 different clouds – private, public and edge)
    - Use Case #4: Application resiliency and global service failover of Google hipster application (from GKE edge cloud failover to on-prem Kubernetes when edge cloud unavailable)

- Installation/howto instruction to on-board other dependent tools for Istio multi-cluster deployment
    - Creating Root CA and generating Intermediate CA.
    - Helm and tiller installation
    - Prometheus installation.

2 Prerequisite

- Functional Kubernetes cluster with Container Network Interface (CNI) installed.
- BIG-IP VE/appliance (TMOS version v13.1.x or higher) installed and operational.
- Root CA and intermediate CA created with the following characteristic (for multi-cluster services mesh with replicated control plane).
    - Root CA does not have name constrains signing
    - CA private key are not password protected (Non Encrypted Private Key)
- Prometheus installed and working with auto-discover and scrape new Kubernetes pods.
- Helm and Tiller installed (Helm version 2.x and below)
- Internet connectivity.

3 Introduction to Aspen Mesh and Multi-Cluster Service Mesh

The objective of this section is to set a common high level baseline understanding on what Aspen Mesh and multi-cluster service mesh and benefit service mesh provides. For details in-dept information, please refer to respective collateral from official website.

Aspen Mesh in a brief

Aspen Mesh, an innovation from F5 Networks, is the enterprise-ready service mesh for Kubernetes built on Istio (validated and supported distribution of Istio) and includes a self-hosted control plane and dashboard. It harnesses enterprise Istio adoption without the headaches with support from Aspen Mesh team of expertise. On top of accessing to Aspen Mesh teams of expertise, Aspen Mesh provides the following additional benefit on top of Open Source Istio

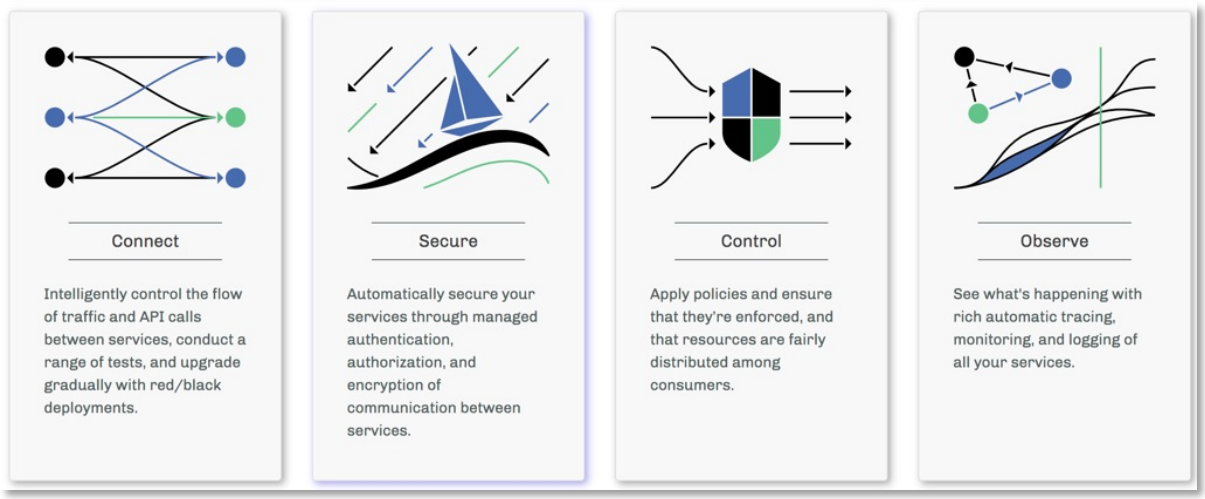
- A policy framework that allows you to specify, measure and enforce business goals
  - RBAC for service traffic
  - Simplified mTLS management
- A simpler user experience delivered through the Aspen Mesh dashboard that makes Istio easier to deploy, monitor and configure
- Analytics and alerting that help you make smarter decisions quickly.
  - Rich visualization of services
  - Real-time performance and security monitoring

Istio Service Mesh in brief

Service meshes are designed to scale, secure, and provide visibility into container environments. It is highly configurable, dedicated and low-latency infrastructure layer design to handle and provide reliable service-to-service communication, implemented as lightweight proxies deployed alongside applications. Example benefit provided by service mesh are

- Service Discovery – automatically detect services and endpoint in a cluster and integrated into service register to provides advance traffic management (e.g. A/B testing or canary deployment)
- Load balancing – allow you to control the flow of traffic and API calls between services.
- Encryption - enforce secure pod-to-pod or service-to-service communication via mutual transport layer security at scale.
- Observability - Insight with tracing, monitoring and logging of microservices to provides better application visibility and performance of all services.
- Security – Enforces Zero trust service domain model, manages authentication and authorization and enforces consistent policies across multiple protocol and runtimes with minimal application changes.

Istio service mesh promises to provide single solution for modern microservices architecture to Connect, Secure, Control and Observe transparently without modifying any application code using envoyproxy as a sidecar. It is an open source project.



It solved a range of challenges that you may face when adopting microservices architecture. In addition, it also provides zero-trust service domains that are automatically encrypted with TLS not only from client-to-service but from service-to-service, and provide visibility and insight that make it practical to measure and debug distributed applications.

3.1 What is Multi-Cluster Service Mesh

Multi-cluster service mesh is an architecture where your workload instances deployed in one or more cluster to form a service mesh. There are few deployment model for multi-cluster service mesh

- **Shared control plane (single-network)** – single Istio control plane with shared network to form a service mesh where pod talk directly to remote pod.
- **Shared control plan (multi-network)** – single Istio control plane with different network via Istio gateway to form a service mesh where pod talk to remote pod via Istio ingressgateway.
- **Replicated control plane** – Replicated control plan where each cluster has its own Istio control plane and pod talk to remote pod directly or via Istio ingressgateway.

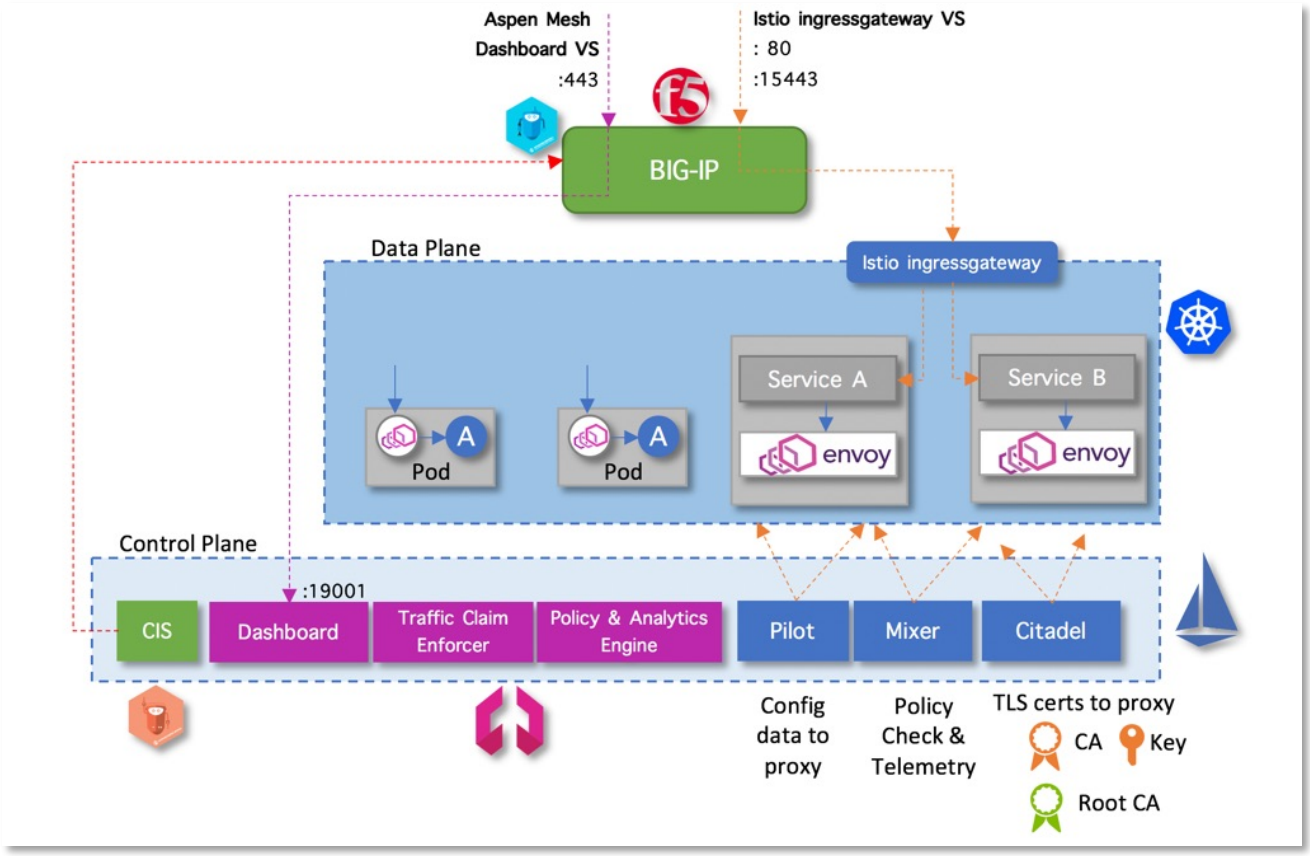
Refer to https://istio.io/docs/ops/deployment/deployment-models/ for details information of those deployment models

Multi-cluster service meshes offers a number of benefits as describe in https://www.infoq.com/articles/kubernetes-multicluster-comms/. There are:-

- Single pane of glass
- Unified trust domain
- Independent fault domains
- Intercluster traffic
- Heterogenous/non-flat network

3.2 Aspen Mesh: Single Cluster Service Mesh High Level Overview

Below is an example topology of a typical single cluster Aspen Mesh / Istio integrated with BIG-IP

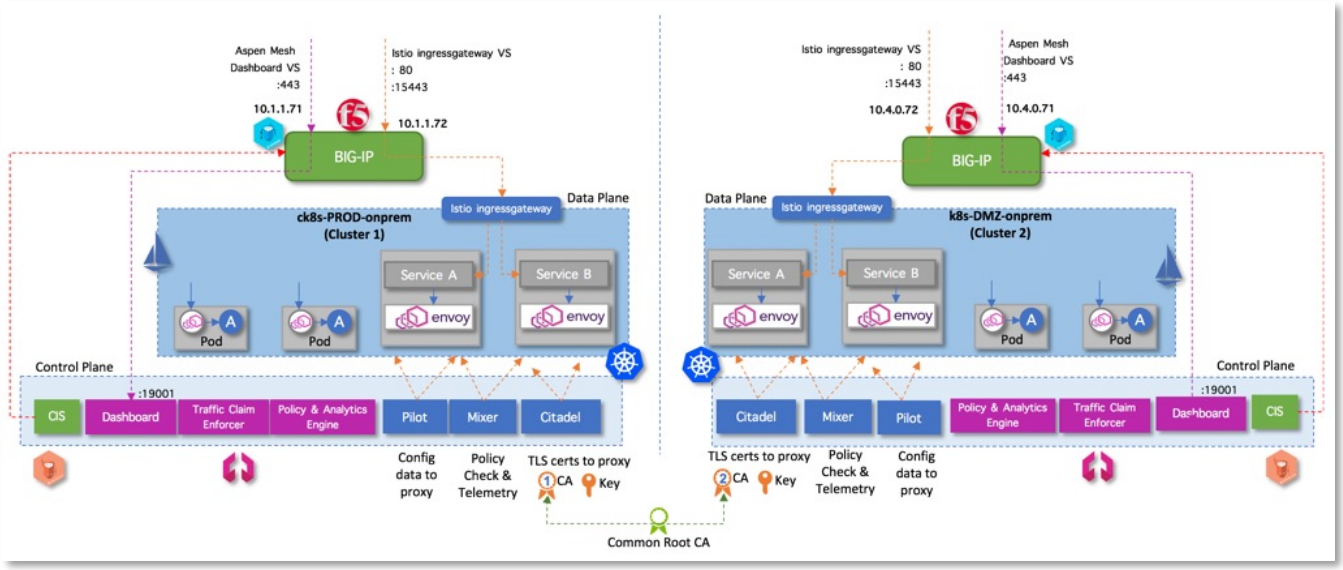


Components	Description
Istio Ingressgateway	Handle incoming requests from outside cluster
Aspen Mesh: Dashboard	Aspen Mesh modern web interface to visualize, monitor and configure Aspen Mesh.
Aspen Mesh: Traffic Claim Enforcer	Traffic Claim Enforcer is a Kubernetes admission controller for Istio that prevents errant application configuration from taking over traffic within the service mesh. Details in https://aspenmesh.io/traffic-claim-enforcer/ or https://www.youtube.com/watch?v=47HzynDsD8w
Aspen Mesh: Policy & Analytics Engine	Aspen Mesh policy enforcement and analytic engine for metrics reporting.
BIG-IP	Act as data plane function to handle external/incoming traffic to Kubernetes microservices / service mesh via Istio ingressgateway (e.g. Istio ingressgateway and Aspen Mesh dashboard).
CIS	BIG-IP Control plane function to configure, update and remove BIG-IP configuration based on changes to monitored containerized applications (e.g. Aspen Mesh dashboard, Istio ingressgateway and etc)

AS3	BIG-IP Declarative endpoint for managing application specific configuration on BIG-IP.
Common Root CA	Root CA use to generate Istio CA that use by Citadel. This Root CA is common across all Istio clusters for multi-cluster deployment.
CA	Certificate Authority use by Citadel (in PEM format)
Key	CA private key (in PEM format – non encrypted private key)
Citadel	Citadel is Istio key and certificate management. It manages the lifecycle of keys and certificates issued for services. When Istio establishes mutual TLS authentication, it uses these keys and certificates to exchange the identities of services
Mixer	Mixer provides a rich intermediation layer between the Istio components as well as Istio-based services, and the infrastructure backends used to perform access control checks and telemetry capture. This layer enables operators to have rich insights and control over service behavior without requiring changes to service binaries.
Pilot	Istio Pilot is responsible for consuming and propagating Istio configuration to Istio components. It also provides an abstraction layer over the underlying cluster management platform, such as Kubernetes, and proxy controllers for dynamic reconfiguration of Istio proxies.

3.3 Aspen Mesh: Multi-Cluster Service Mesh High Level Overview

Below is an example topology of a typical dual cluster (multi-cluster) Aspen Mesh / Istio integrated with BIG-IP. This topology can be expanded to more than 2 cluster where each cluster certificate authority (CA) is signed by a common Root CA.



In a multi-cluster service mesh with replicated control plane architecture, each Istio cluster has its own control plane where each managing its own endpoints. All of the clusters are under a shared administrative control for the purposes of policy enforcement and security.

A single Istio service mesh across the clusters is achieved by replicating shared services and namespaces and using a common root CA in all of the clusters. Cross-cluster communication occurs over Istio gateways (Ingress / Egress) of the respective clusters.

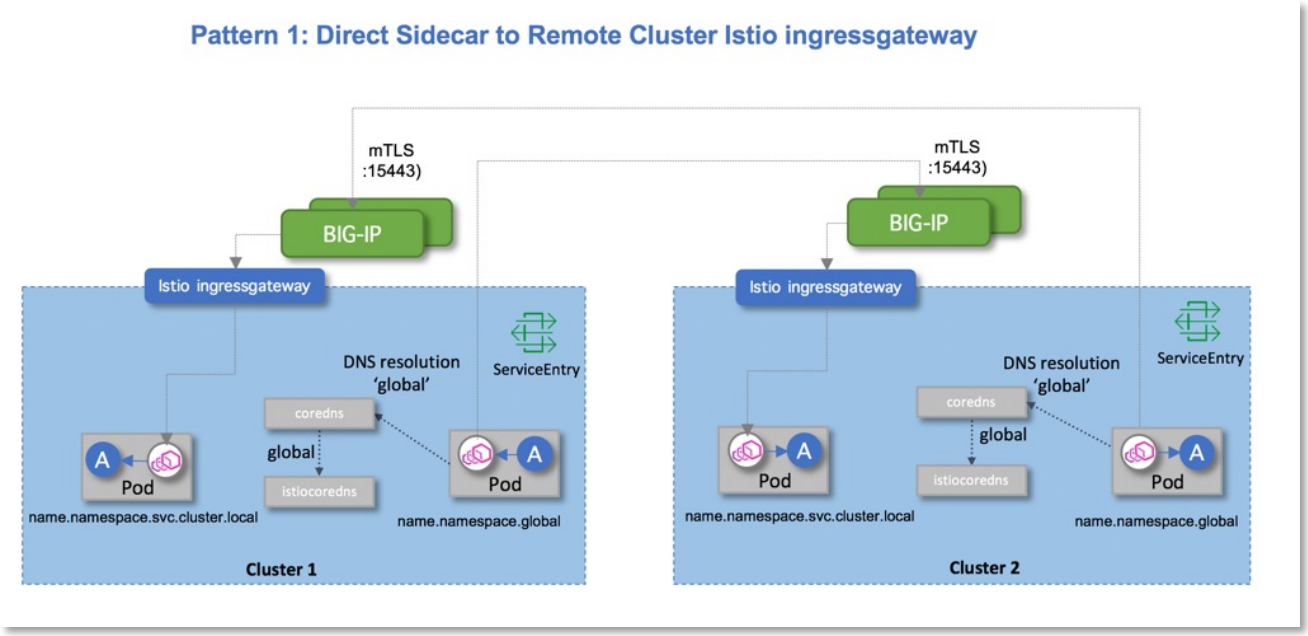
3.4 Aspen Mesh: Replicated Control Plane - Inter-Cluster traffic delivery options

When applications need to communicate with another service, application itself will expect to resolve services by their DNS names and access the resulting IP. Typically, services that reside locally shared a common DNS suffix (e.g. svc.cluster.local) which resolve by local DNS (kubedns/coredns reside in kube-system namespace). To access services that reside on a remote cluster, Kubernetes DNS (in kube-system namespace) need to be configured to stub a domain from .global. '.global' domain will be forwarded to istiocoredns (in istio-system namespace) for DNS resolution for remote services. Remote services will be identified via DNS suffix of '<name>.<namespace>.global'

Every service in a given cluster that needs to be accessed from a different remote cluster requires a ServiceEntry configuration in the remote cluster. The host used in the service entry should be of the form <name>.<namespace>.global where name and namespace correspond to the service's name and namespace respectively.

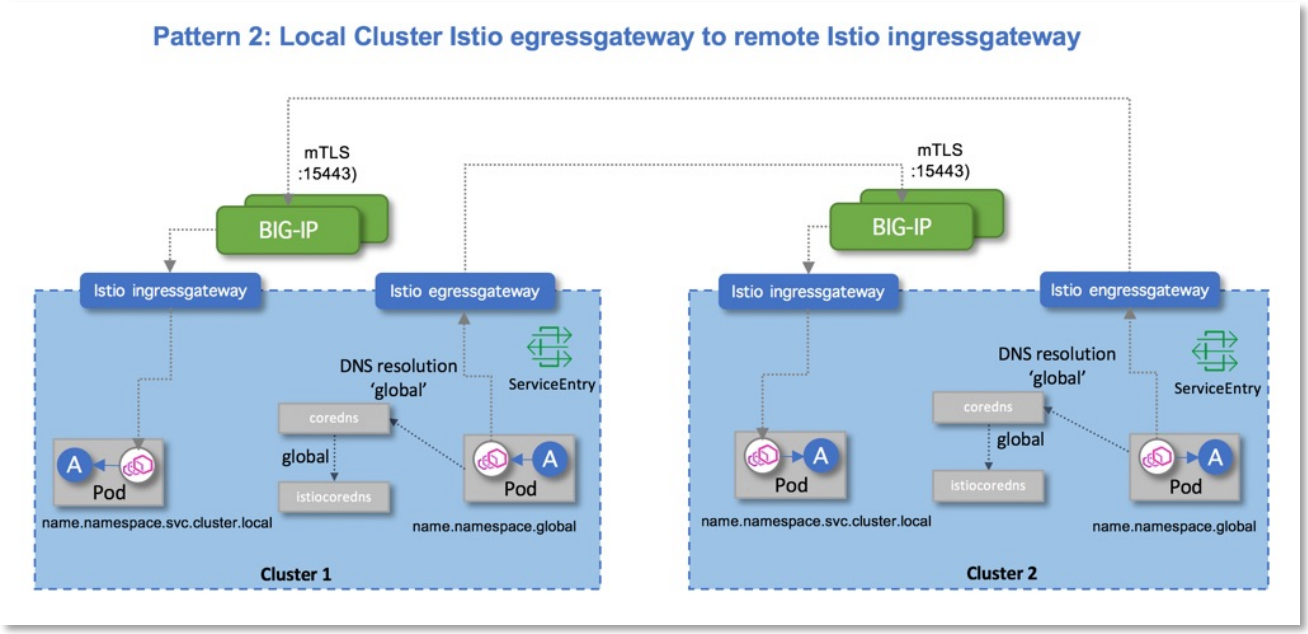
There are 2 distinct traffic pattern for accessing remote service (ServiceEntry definition options) as shown below:-

3.4.1 Pattern 1: Direct Sidecar to Remote Cluster Istio Ingressgateway



In this pattern, pod (envoy proxy) send traffic directly (via mTLS) to remote cluster Istio ingressgateway and Istio ingressgateway send to respective service pod. Local pods will source its IP from the local cluster node IP. Depend on where local pods reside, remote Istio ingressgateway may see different originating node IP.

3.4.2 Pattern 2: Local Cluster Istio Egressgateway to remote Istio Ingressgateway



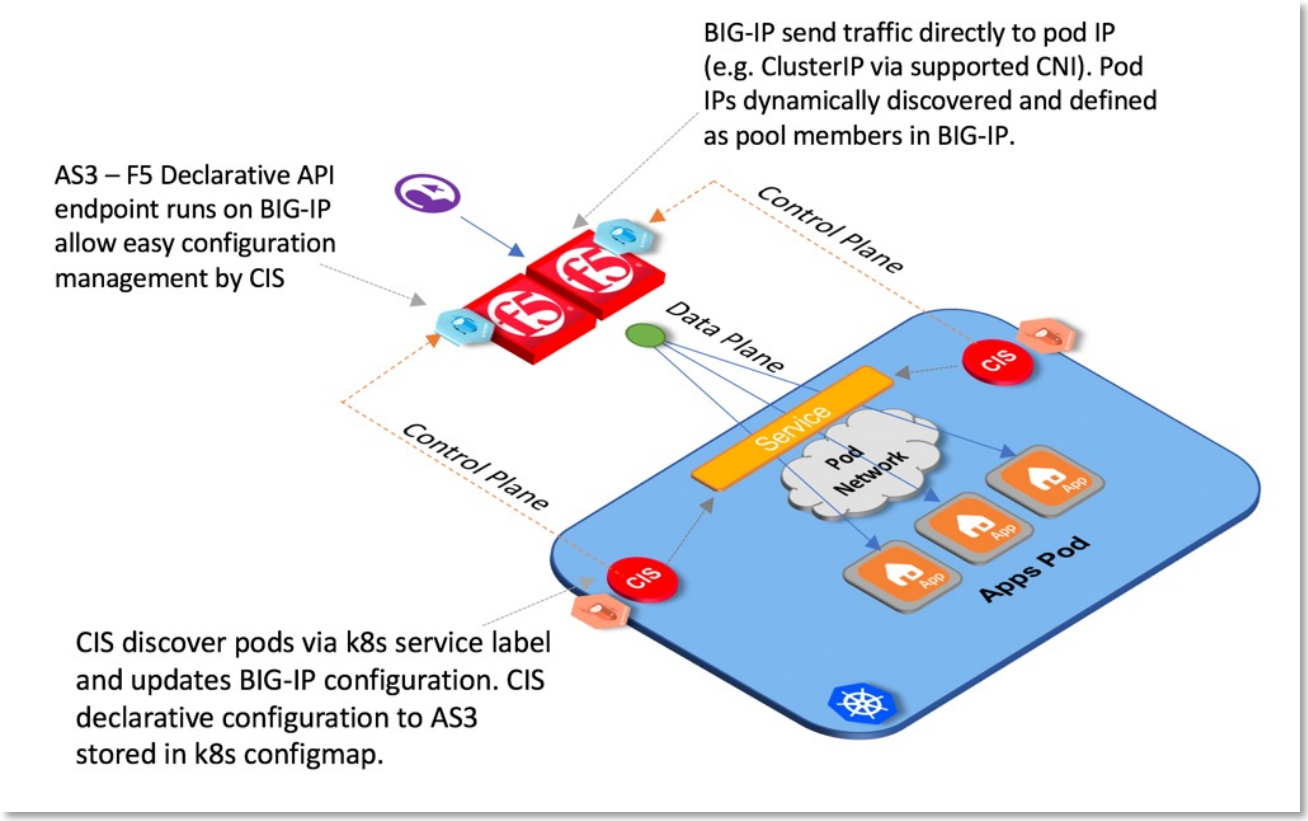
In this pattern, pod (envoy proxy) send to local Istio egressgateway and local Istio egressgateway will send to remote Istio ingressgateway and vice versa. All pods will send via a common egress gateway. Istio egressgateway allow you to apply Istio features, for example, monitoring and route rules to traffic existing the mesh.

For detail use case of egress gateway, please refer to https://istio.io/docs/tasks/traffic-management/egress/egress-gateway/ and https://istio.io/docs/tasks/traffic-management/egress/egress-gateway-tls-origination/

4 Introduction to F5 CIS and AS3.

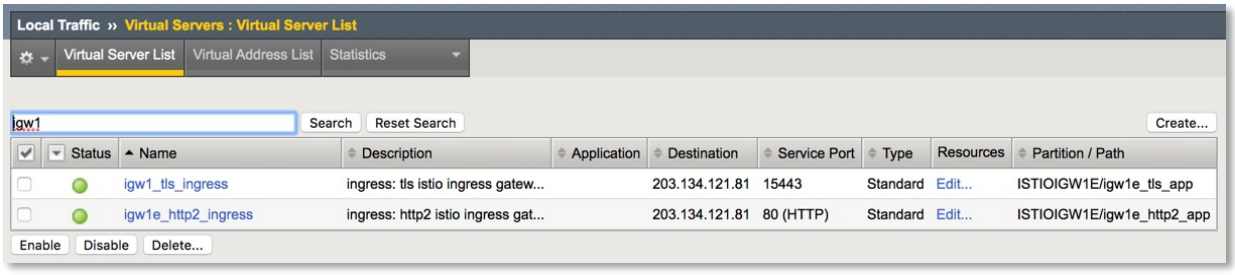
F5 Container Ingress Services (CIS) integrates with container orchestration environments (e.g. Kubernetes or OpenShift) to dynamically create L4/L7 services on F5 BIG-IP systems, and load balance network traffic across those services. Monitoring the orchestration API server, CIS is able to modify the BIG-IP system configuration based on changes made to containerized applications.

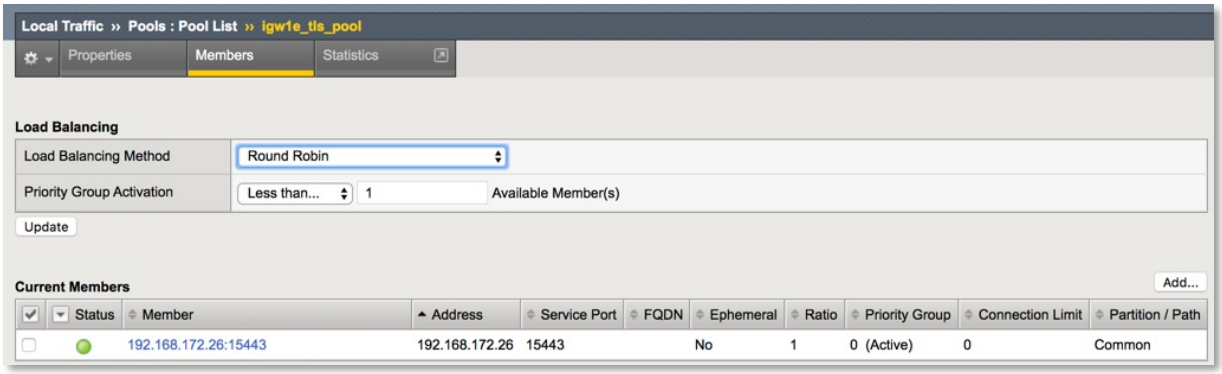
F5 AS3 (Application Services 3 Extension) is a flexible, low-overhead mechanism for managing application-specific configuration on a BIG-IP system. AS3 uses a declarative model, meaning you provide a JSON declaration rather than a set of imperative command. The declaration represents the configuration which AS3 is responsible for creating on a BIG-IP system. AS3 is well-defined according to the rules of JSON Schema, and declarations validate according to JSON Schema. AS3 accepts declaration updates via REST (push), reference (pull) or CLI.



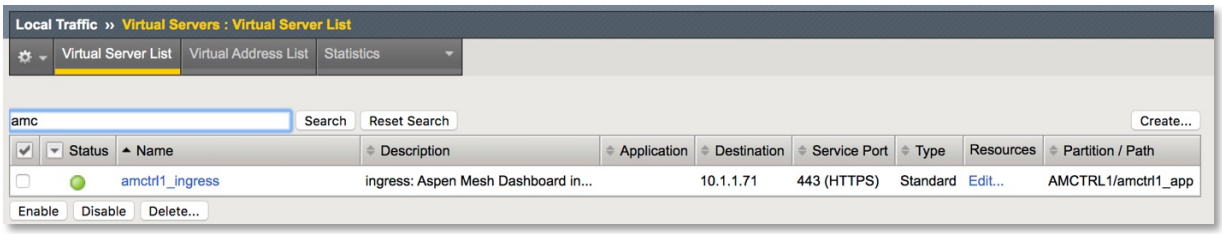
BIG-IP + CIS is an Ingress controller for Kubernetes, where BIG-IP handling the data plane component and CIS the control plane component.

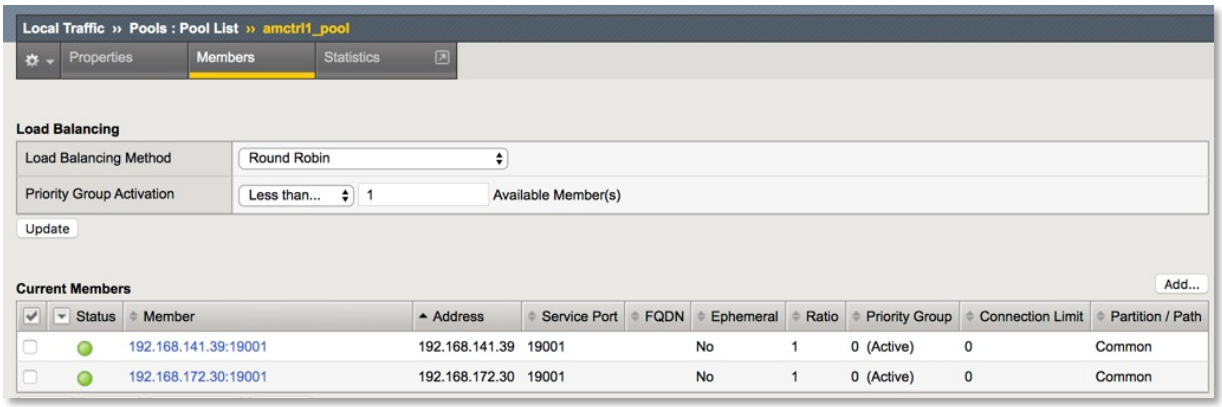
Example BIG-IP virtual servers dynamically discovered and configured by CIS on BIP-IP to handle ingress traffic to Istio Service Mesh





Example BIG-IP virtual server dynamically discovered and configured by CIS on BIP-IP to handle ingress traffic to Aspen Mesh Dashboard





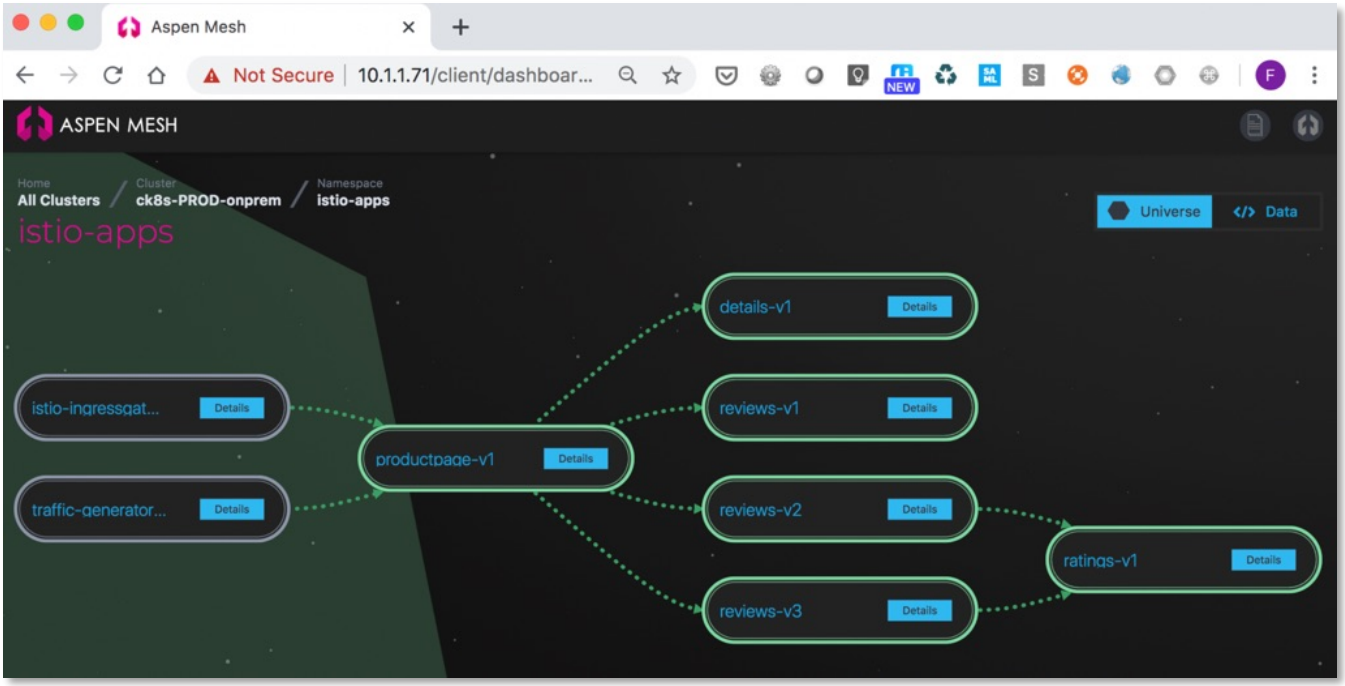
5 Application of multi-cloud, multi-cluster service mesh

Details deployment instructions (installation of Aspen Mesh multi-cluster) will be described in next section. This section is to demonstrate various use cases and application of multi-cloud and multi-cluster service mesh. Example configuration manifest can be found in https://github.com/fbchan/aspen-mesh-multi-cluster. This section assume that you already have Aspen Mesh multi-cluster installed and working properly across 3 different cloud

- 2 x Kubernetes on-prem (ck8s-PROD-onprem and k8s-DMZ-onprem)
- 1 x GKE in Google Cloud (gke-DEV-Edge1)

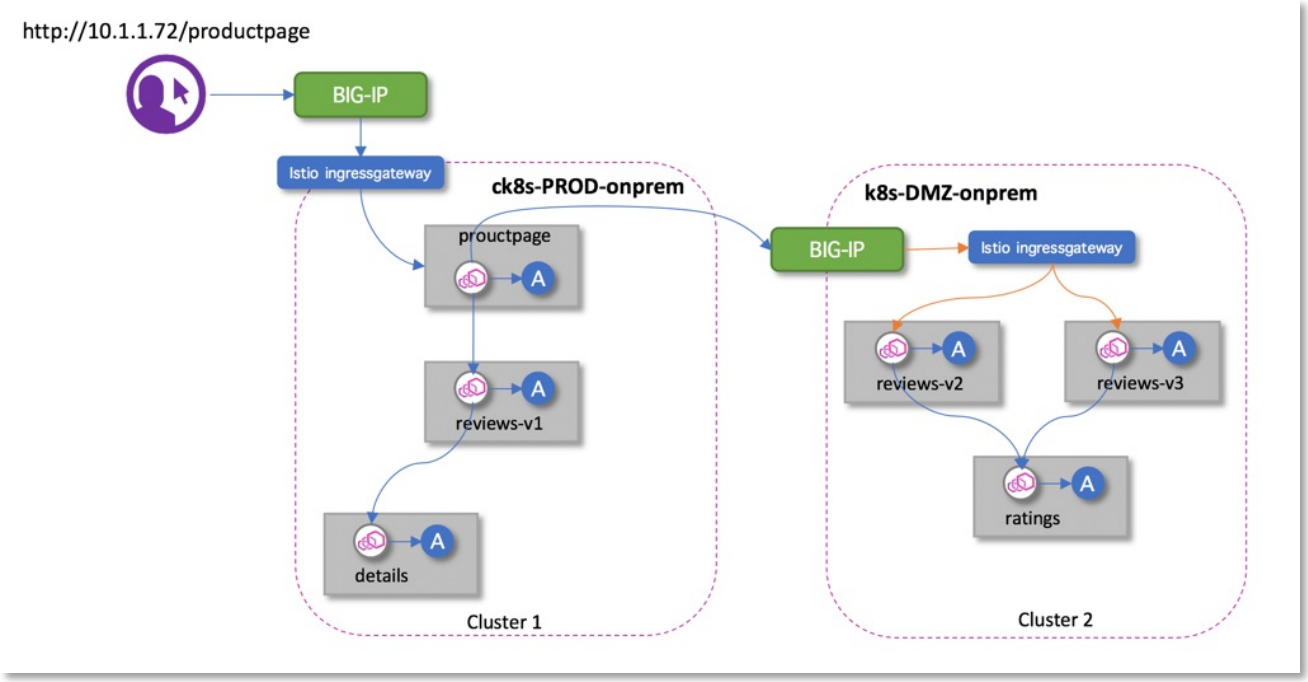
5.1 Use Case#1: Single cluster service mesh with Bookinfo sample application

Please refer to “Aspen Mesh Multi-Cluster Deployment Guide” for step by step instructions on installation of Aspen mesh multi-cluster. Bookinfo sample application will be part of the installation. Upon successful of the installation, you should see the following Aspen Mesh dashboard.

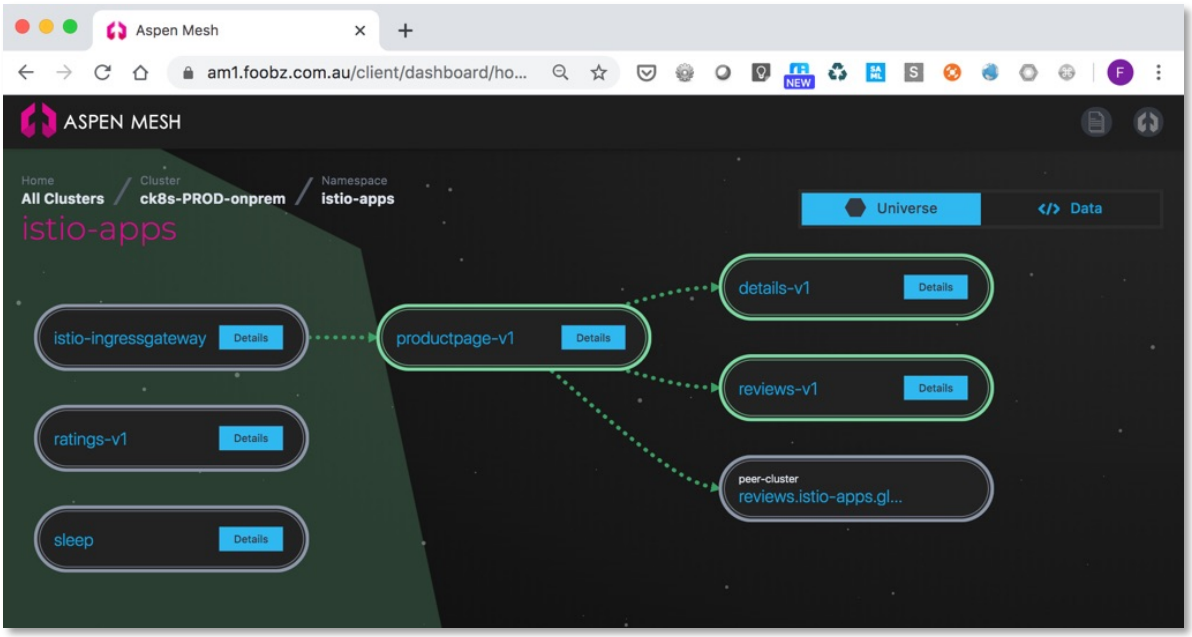


5.2 Use Case#2: Multi-cluster service mesh with Bookinfo sample application

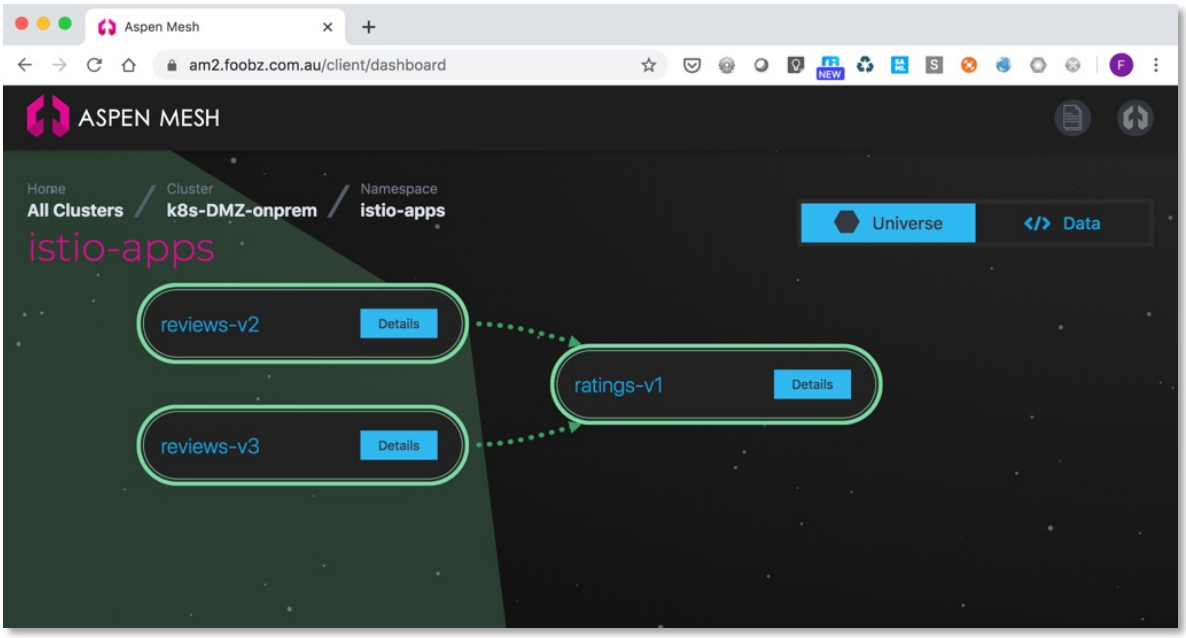
In this use case, sample Bookinfo application 6 individual microservices will be distributed across 2 different Kubernetes cluster as shown below. Both Kubernetes clusters runs on-prem with different version of Kubernetes.



Aspen Mesh Dashboard on Cluster 1 will show traffic visibility from productpage to review-v1, details-v1 and peer-cluster (remote cluster)



When jump onto Aspen Mesh Dashboard in Cluster 2, review-v2, review-v3 and ratings-v1 will be shown.

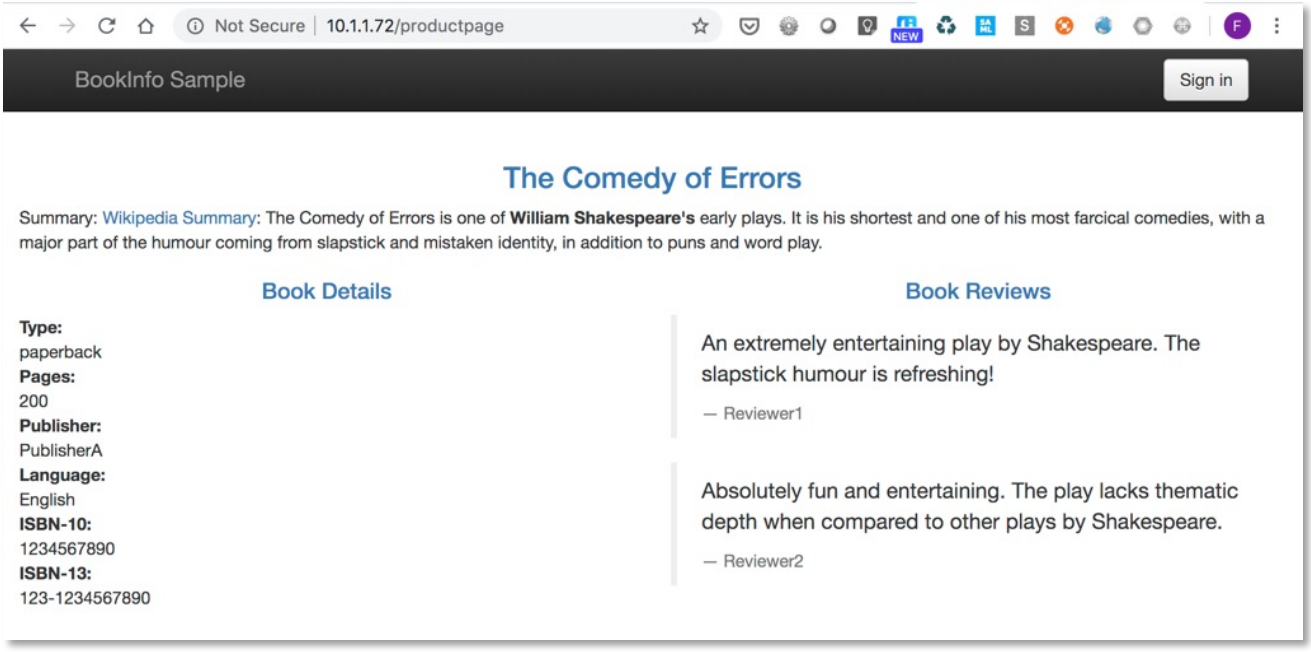


Note: You can either deploy with a new copies of BookInfo application or modify existing BookInfo application. Instruction shown below are based on modification of exiting BookInfo application installed in istio-apps namespace.

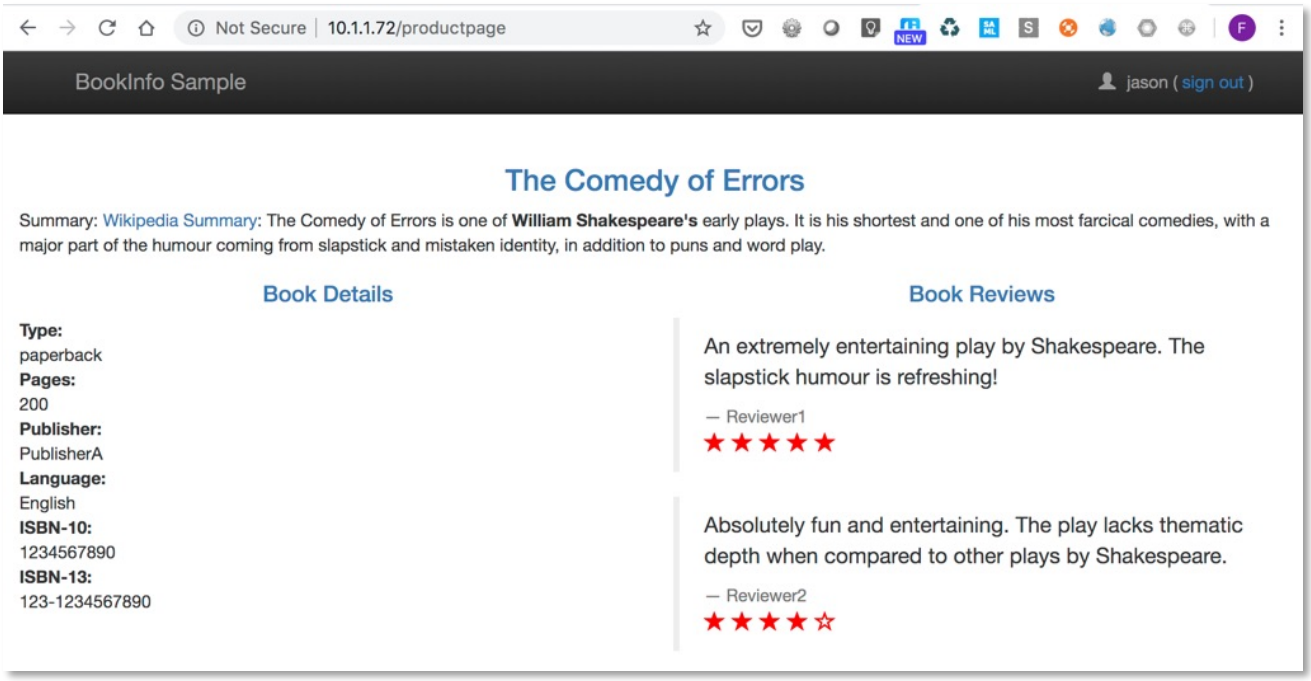
```
On ck8s-prod-onprem (Cluster1)
fbchan@ck8s-1:~$ kubectl -n istio-apps get deployment
NAME                          READY   UP-TO-DATE   AVAILABLE   AGE
details-v1                    1/1     1            1           41h
productpage-v1                1/1     1            1           41h
ratings-v1                    1/1     1            1           41h
reviews-v1                    1/1     1            1           41h
reviews-v2                    1/1     1            1           41h
reviews-v3                    1/1     1            1           41h

fbchan@ck8s-1:~$ kubectl -n istio-apps delete deploy reviews-v2 reviews-v3 ratings-v1
deployment.apps "reviews-v2" deleted
deployment.apps "reviews-v3" deleted
deployment.apps "ratings-v1" deleted

fbchan@ck8s-1:~$ kubectl -n istio-apps get pod
NAME                           READY   STATUS    RESTARTS   AGE
details-v1-78d78fbddf-2txn8    2/2     Running   0          7d22h
productpage-v1-596598f447-qncpm   2/2     Running   0          7d22h
reviews-v1-7bb8ffd9b6-l6gnr    2/2     Running   0          7d22h

fbchan@ck8s-1:~/aspenmesh-1.3.6-am1/multi-cluster-apps$ kubectl -n istio-apps apply -f review-
istio-apps-dst_rule_v1.yml
destinationrule.networking.istio.io/reviews created

fbchan@ck8s-1:~/aspenmesh-1.3.6-am1/multi-cluster-apps$ kubectl -n istio-apps apply -f review-
virtualservice.yml
virtualservice.networking.istio.io/reviews created

fbchan@ck8s-1:~/aspenmesh-1.3.6-am1/multi-cluster-apps$ kubectl apply -f review-istio-apps-
se.yml
serviceentry.networking.istio.io/reviews-istio-apps created
destinationrule.networking.istio.io/reviews-global created

fbchan@ck8s-1:~/aspenmesh-1.3.6-am1/multi-cluster-apps$ kubectl -n istio-apps get se
NAME                HOSTS                          LOCATION        RESOLUTION   AGE
reviews-istio-apps   [reviews.istio-apps.global]   MESH_INTERNAL   DNS          7d22h
```

```
On k8s-prod-onprem (Cluster2)
fbchan@k8s-1:~/aspenmesh-1.3.6-am1$ kubectl -n istio-apps delete deploy reviews-v1 productpage-v1 details-v1
deployment.extensions "reviews-v1" deleted
deployment.extensions "productpage-v1" deleted
deployment.extensions "details-v1" deleted

fbchan@k8s-1:~/aspenmesh-1.3.6-am1$ kubectl -n istio-apps get deploy
NAME              READY   UP-TO-DATE   AVAILABLE   AGE
ratings-v1        1/1     1            1           6d22h
reviews-v2        1/1     1            1           16d
reviews-v3        1/1     1            1           16d
fbchan@k8s-1:~/aspenmesh-1.3.6-am1$

fbchan@k8s-1:~/aspenmesh-1.3.6-am1$ kubectl -n istio-apps apply -f review-istio-apps-dst_rule_v2_v3.yml
destinationrule.networking.istio.io/reviews created

fbchan@k8s-1:~/aspenmesh-1.3.6-am1$ kubectl -n istio-apps get dr
NAME      HOST                                  AGE
reviews   reviews.istio-apps.svc.cluster.local   13d
```

When accessing Bookinfo application, by default, only review-v1 will be shown. Additional information on review provided by review-v2 and review-v3 (without colour stars ratings – black and red) will not be shown.

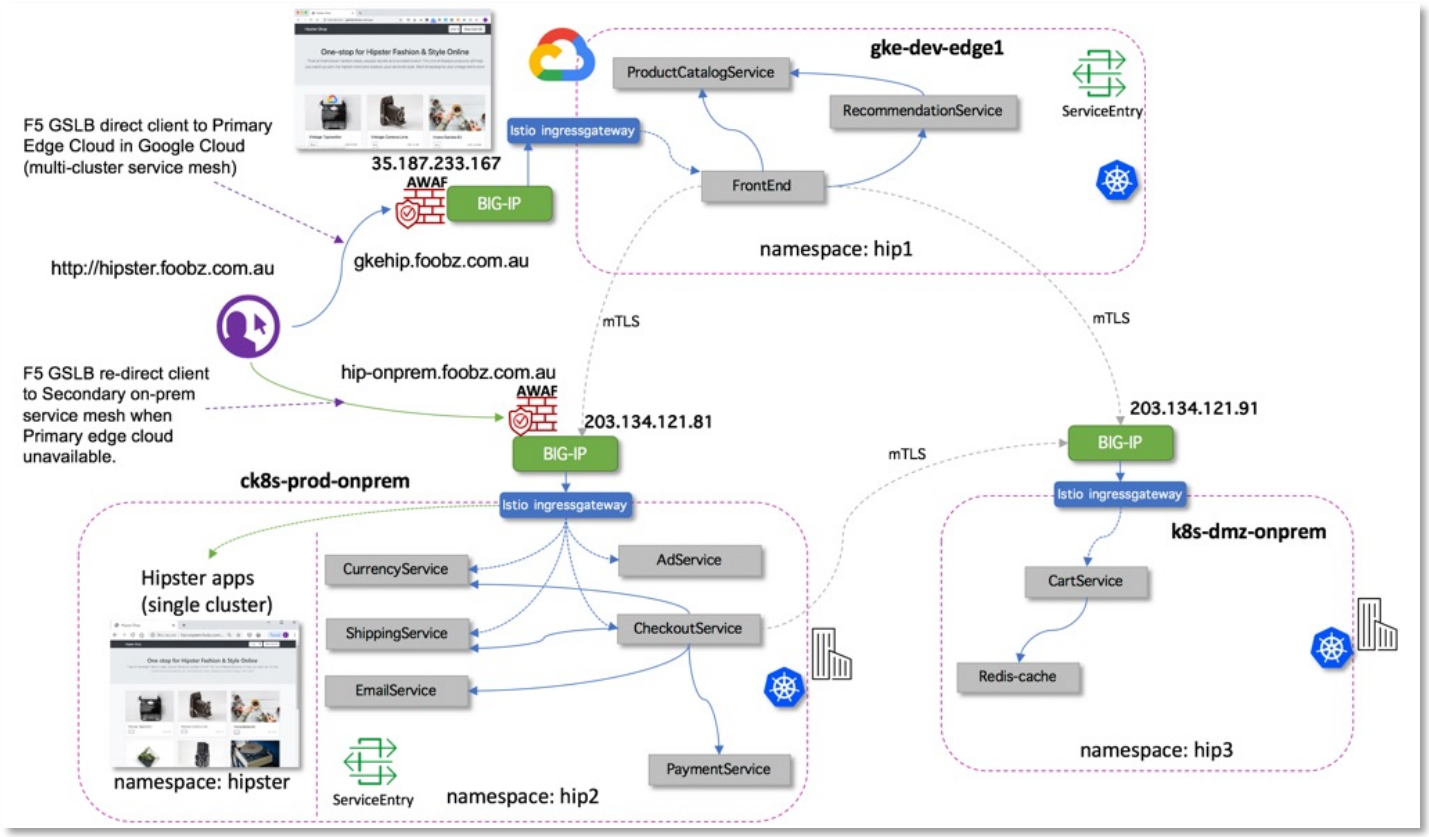


When login as ‘jason’, additional review information provided by review-v2 and review-v3 will be fetched remotely from Cluster 2


Not Secure | 10.1.1.72/productpage
BookInfo Sample
jason (sign out)
The Comedy of Errors
Summary: Wikipedia Summary: The Comedy of Errors is one of William Shakespeare's early plays. It is his shortest and one of his most farcical comedies, with a major part of the humour coming from slapstick and mistaken identity, in addition to puns and word play.
Book Details
Type:
paperback
Pages:
200
Publisher:
PublisherA
Language:
English
ISBN-10:
1234567890
ISBN-13:
123-1234567890
Book Reviews
An extremely entertaining play by Shakespeare. The slapstick humour is refreshing!
— Reviewer1
Absolutely fun and entertaining. The play lacks thematic depth when compared to other plays by Shakespeare.
— Reviewer2

5.3 Use Case#3: Multi-cluster service mesh with Google hipster application

Hipster application (10 microservices) distributed across 3 different Kubernetes cluster (2 x on-prem Kubernetes and 1 x Google GKE) as shown below. Note: This example was derived from Google Cloud Platform Istio multi-cluster dual control plan demo located at https://github.com/GoogleCloudPlatform/istio-samples/tree/master/multicluster-gke/dual-control-plane. Details configuration manifest use in this example be obtained from https://github.com/fbchan/aspen-mesh-multi-cluster. Hipster application publicly accessible endpoint (port 80) is protected by F5 Advance Web Application Firewall dynamically provisioned by CIS and AS3.



Create all three namespaces (hip1, hip2 and hip3) and labelled for istio-injection on respective Kubernetes cluster

```
On GKE Cluster (gke-dev-edge1)
fbchan@logos:~/k8s-clusterX/gke-1/github/hipster apps/hip1-gke$ kubectl create namespace hip1
namespace/hip1 created

fbchan@logos:~/k8s-clusterX/gke-1/github/hipster apps/hip1-gke$ kubectl label --overwrite
namespace hip1 istio-injection=enabled
namespace/hip1 labeled

fbchan@logos:~/k8s-clusterX/gke-1/github/hipster apps/hip1-gke$ kubectl get namespace -L istio-
injection
NAME              STATUS   AGE     ISTIO-INJECTION
default           Active   7d10h
hip1              Active   61s     enabled
istio-apps        Active   7d8h    enabled
istio-system      Active   7d8h
kube-node-lease   Active   7d10h
kube-public       Active   7d10h
kube-system       Active   7d10h
monitoring        Active   7d8h
sock-shop         Active   6d18h
```

```
On on-prem ck8s Cluster (ck8s-prod-onprem)
fbchan@ck8s-1:~/aspenmesh-1.3.6-am3/github/hipster apps/hip2-ck8s$ kubectl create namespace hip2
namespace/hip2 created

fbchan@ck8s-1:~/aspenmesh-1.3.6-am3/github/hipster apps/hip2-ck8s$ kubectl label --overwrite
namespace hip2 istio-injection=enabled
namespace/hip2 labeled

fbchan@ck8s-1:~/aspenmesh-1.3.6-am3/github/hipster apps/hip2-ck8s$ kubectl get namespace -L
istio-injection
NAME              STATUS   AGE     ISTIO-INJECTION
default           Active   9d
hip2              Active   48s     enabled
hipster           Active   14h     enabled
istio-apps        Active   7d15h   enabled
istio-system      Active   9d
kube-node-lease   Active   9d
kube-public       Active   9d
kube-system       Active   9d
monitoring        Active   9d



On on-prem k8s Cluster (k8s-dmz-onprem)
fbchan@k8s-1:~/aspenmesh-1.3.6-am3/github/hipster apps/hip3-k8s$ kubectl create namespace hip3
namespace/hip3 created

fbchan@k8s-1:~/aspenmesh-1.3.6-am3/github/hipster apps/hip3-k8s$ kubectl label --overwrite
namespace hip3 istio-injection=enabled
namespace/hip3 labeled

fbchan@k8s-1:~/aspenmesh-1.3.6-am3/github/hipster apps/hip3-k8s$ kubectl get namespace -L istio-
injection
NAME              STATUS   AGE    ISTIO-INJECTION
default           Active   212d
hip3              Active   34s    enabled
istio-apps        Active   10d    enabled
istio-system      Active   10d
kube-node-lease   Active   212d
kube-public       Active   212d
kube-system       Active   212d
kubeapps          Active   211d
metallb-system    Active   211d
monitoring        Active   193d
nginxp-ingress    Active   72d
sock-shop         Active   71d    enabled
weave             Active   206d
```

Note: For hip2, a duplicate hipster application deployed in ‘hipster’ namespace. This namespace stored backup copies of hipster application. F5 GSLB will redirect to this namespace in the event if GKE hip1 not available.

Deploy application on namespace for hip3, hip2 and hip1 (Start with hip3)

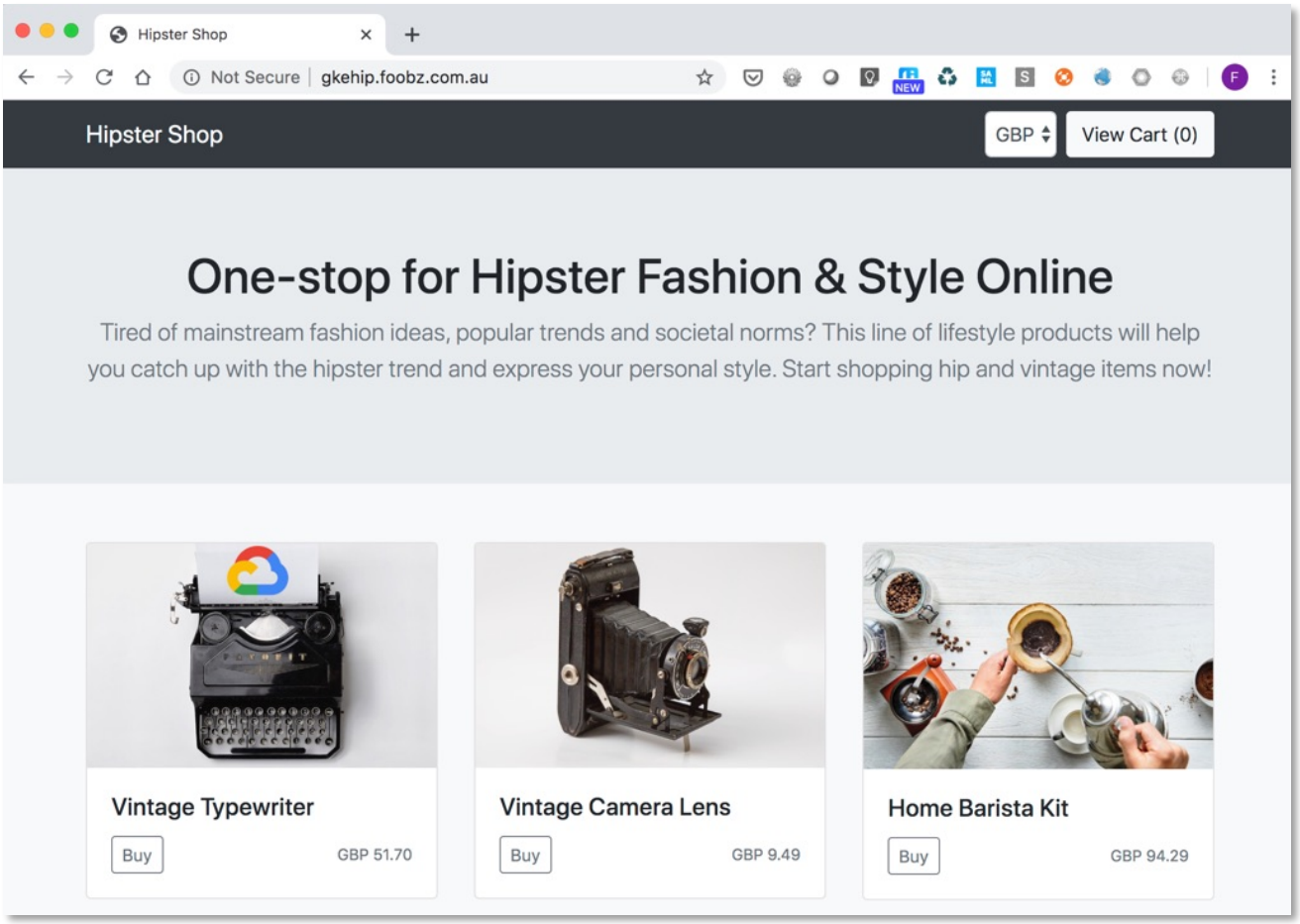
```
On on-prem k8s Cluster (k8s-dmz-onprem)
fbchan@k8s-1:~/aspenmesh-1.3.6-am3/github/hipster apps/hip3-k8s$ kubectl -n hip3 apply -f 01-
hip3-services-local.yml
service/redis-cart created
service/cartservice created

fbchan@k8s-1:~/aspenmesh-1.3.6-am3/github/hipster apps/hip3-k8s$ kubectl -n hip3 apply -f 02-
hip3-deployment.yml
deployment.apps/redis-cart created
deployment.apps/cartservice created
```

```
fbchan@k8s-1:~/aspenmesh-1.3.6-am3/github/hipster apps/hip3-k8s$ kubectl -n hip3 get pod
NAME                            READY   STATUS    RESTARTS   AGE
cartservice-6cbc9b899c-qtp2d    2/2     Running   0          35m
redis-cart-6448dcbdcc-x6lp5     2/2     Running   0          35m

fbchan@k8s-1:~/aspenmesh-1.3.6-am3/github/hipster apps/hip3-k8s$ kubectl -n hip3 get svc
NAME          TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
cartservice   ClusterIP   10.98.171.131   <none>        7070/TCP   40m
redis-cart    ClusterIP   10.102.50.117   <none>        6379/TCP   40m


On on-prem ck8s Cluster (ck8s-prod-onprem)
fbchan@ck8s-1:~/aspenmesh-1.3.6-am3/github/hipster apps/hip2-ck8s$ kubectl -n hip2 apply -f 01-
hip2-services-local.yaml
service/emailservice created
service/checkoutservice created
service/paymentservice created
service/currencyservice created
service/shippingservice created
service/adservice created

fbchan@ck8s-1:~/aspenmesh-1.3.6-am3/github/hipster apps/hip2-ck8s$ kubectl -n hip2 apply -f 02-
hip2-service-entries.yaml
serviceentry.networking.istio.io/frontendservice-entry created
serviceentry.networking.istio.io/productcatalogservice-entry created
serviceentry.networking.istio.io/cartservice-entry created

fbchan@ck8s-1:~/aspenmesh-1.3.6-am3/github/hipster apps/hip2-ck8s$ kubectl -n hip2 apply -f 03-
hip2-istio-defaults.yaml
serviceentry.networking.istio.io/currency-provider-external created
serviceentry.networking.istio.io/whitelist-egress-googleapis created
serviceentry.networking.istio.io/whitelist-egress-google-metadata created

fbchan@ck8s-1:~/aspenmesh-1.3.6-am3/github/hipster apps/hip2-ck8s$ kubectl -n hip2 apply -f 04-
hip2-deployments.yaml
deployment.apps/emailservice created
deployment.apps/checkoutservice created
deployment.apps/paymentservice created
deployment.apps/currencyservice created
deployment.apps/shippingservice created
deployment.apps/adservice created

fbchan@ck8s-1:~/aspenmesh-1.3.6-am3/github/hipster apps/hip2-ck8s$ kubectl -n hip2 get pod
NAME                                 READY   STATUS    RESTARTS   AGE
adservice-55f9757757-ltn9j           2/2     Running   0          37m
checkoutservice-74fbbd5f65-d4kpm     2/2     Running   0          37m
currencyservice-6c7c479d45-d695m     2/2     Running   0          37m
emailservice-8dd9b76cc-fvzrh         2/2     Running   0          37m
paymentservice-84ffc75c55-m7zns      2/2     Running   0          37m
shippingservice-b6db65f7f-5kkwn      2/2     Running   0          37m

fbchan@ck8s-1:~/aspenmesh-1.3.6-am3/github/hipster apps/hip2-ck8s$ kubectl -n hip2 get svc
NAME              TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)     AGE
adservice         ClusterIP   10.96.81.80      <none>        9555/TCP    40m
checkoutservice   ClusterIP   10.108.240.86    <none>        5050/TCP    40m
currencyservice   ClusterIP   10.104.222.177   <none>        7000/TCP    40m
emailservice      ClusterIP   10.96.17.100     <none>        5000/TCP    40m
paymentservice    ClusterIP   10.105.171.129   <none>        50051/TCP   40m
shippingservice   ClusterIP   10.101.45.149    <none>        50051/TCP   40m

fbchan@ck8s-1:~/aspenmesh-1.3.6-am3/github/hipster apps/hip2-ck8s$ kubectl -n hip2 get se
NAME                          HOSTS                                  LOCATION        RESOLUTION   AGE
cartservice-entry             [cartservice.hip3.global]              MESH_INTERNAL   DNS          39m
currency-provider-external    [www.ecb.europa.eu]                                    DNS          39m
frontendservice-entry         [frontend.hip1.global]                 MESH_INTERNAL   DNS          39m
productcatalogservice-entry   [productcatalogservice.hip1.global]    MESH_INTERNAL   DNS          39m
```

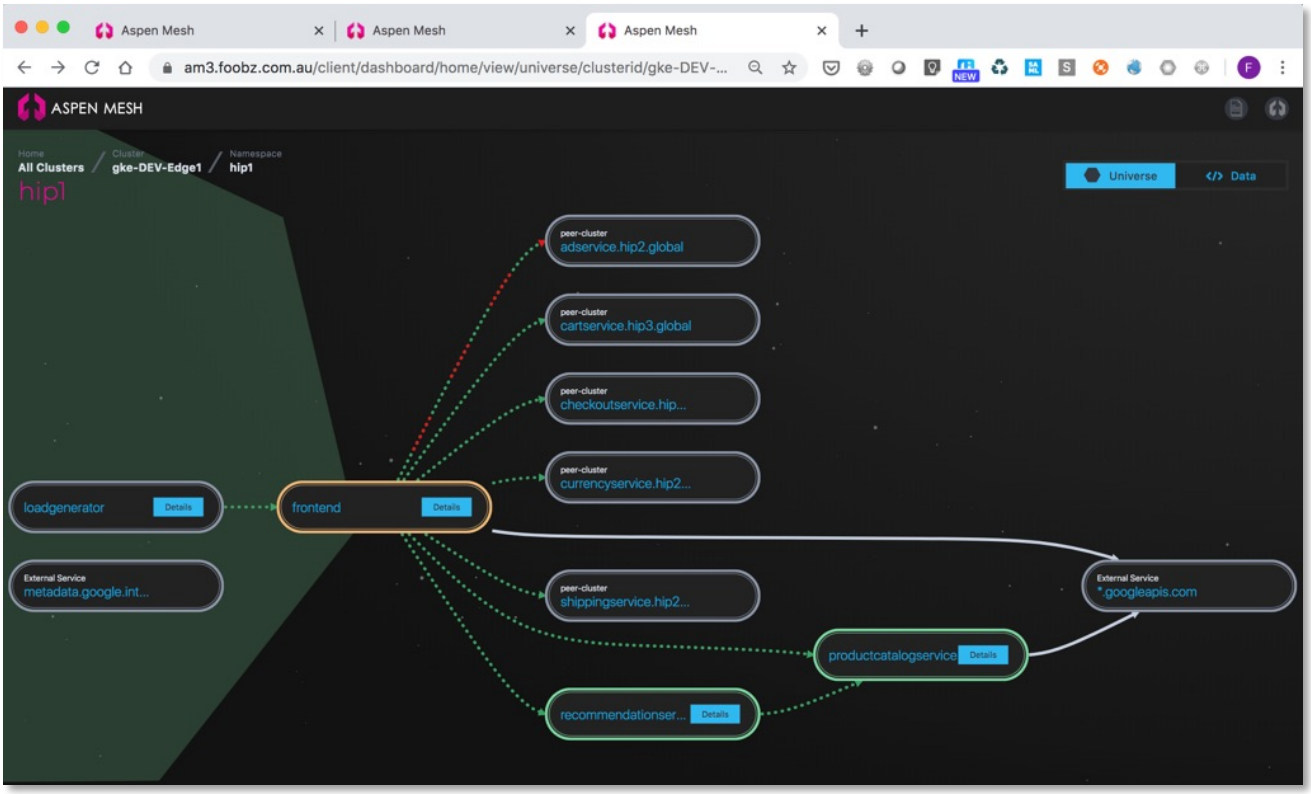
```
whitelist-egress-google-metadata   [metadata.google.internal]                              39m
whitelist-egress-googleapis        [accounts.google.com *.googleapis.com]                 39m



On GKE Cluster (gke-dev-edge1)
fbchan@logos:~/k8s-clusterX/gke-1/github/hipster apps/hip1-gke$ kubectl -n hip1 apply -f 01-
hip1-services-local.yaml
service/recommendationservice created
service/frontend created
service/productcatalogservice created

fbchan@logos:~/k8s-clusterX/gke-1/github/hipster apps/hip1-gke$ kubectl -n hip1 apply -f 02-
hip1-service-entries.yaml
serviceentry.networking.istio.io/adservice-entry created
serviceentry.networking.istio.io/checkoutservice-entry created
serviceentry.networking.istio.io/currencyservice-entry created
serviceentry.networking.istio.io/shippingservice-entry created
serviceentry.networking.istio.io/cartservice-entry created

fbchan@logos:~/k8s-clusterX/gke-1/github/hipster apps/hip1-gke$ kubectl -n hip1 apply -f 03-
hip1-istio-defaults.yaml
gateway.networking.istio.io/frontend-gateway created
virtualservice.networking.istio.io/frontend-ingress created
virtualservice.networking.istio.io/frontend created
serviceentry.networking.istio.io/whitelist-egress-googleapis created
serviceentry.networking.istio.io/whitelist-egress-google-metadata created

fbchan@logos:~/k8s-clusterX/gke-1/github/hipster apps/hip1-gke$ kubectl -n hip1 apply -f 04-
hip1-deployments.yaml
deployment.apps/frontend created
deployment.apps/productcatalogservice created
deployment.apps/cartservice created
deployment.apps/recommendationservice created

fbchan@logos:~/k8s-clusterX/gke-1/github/hipster apps/hip1-gke$ kubectl -n hip1 apply -f 05-
hip1-load-generator.yml
deployment.apps/loadgenerator created

fbchan@logos:~/k8s-clusterX/gke-1/github/hipster apps/hip1-gke$ kubectl -n hip1 get pod
NAME                                     READY   STATUS    RESTARTS   AGE
frontend-7fcd8dfcb6-pk7r5                2/2     Running   0          30m
loadgenerator-7fbdd84d67-2wgk7           2/2     Running   1          26m
productcatalogservice-79cbdffdcf-nllfp   2/2     Running   0          30m
recommendationservice-666fffd4-ksdpp     2/2     Running   0          30m

fbchan@logos:~/k8s-clusterX/gke-1/github/hipster apps/hip1-gke$ kubectl -n hip1 get svc
NAME                    TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)    AGE
frontend                ClusterIP   10.12.20.83    <none>        80/TCP     44m
productcatalogservice   ClusterIP   10.12.31.105   <none>        3550/TCP   44m
recommendationservice   ClusterIP   10.12.17.189   <none>        8080/TCP   44m

fbchan@logos:~/k8s-clusterX/gke-1/github/hipster apps/hip1-gke$ kubectl -n hip1 get vs
NAME               GATEWAYS             HOSTS                                AGE
frontend                                [frontend.hip1.svc.cluster.local]    41m
frontend-ingress   [frontend-gateway]   [gkehip.foobz.com.au hipster.foobz.com.au] 41m

fbchan@logos:~/k8s-clusterX/gke-1/github/hipster apps/hip1-gke$ kubectl -n hip1 get gw
NAME               AGE
frontend-gateway   41m

fbchan@logos:~/k8s-clusterX/gke-1/github/hipster apps/hip1-gke$ kubectl -n hip1 get se
NAME                          HOSTS                                LOCATION        RESOLUTION   AGE
adservice-entry               [adservice.hip2.global]              MESH_INTERNAL   DNS          41m
cartservice-entry             [cartservice.hip3.global]            MESH_INTERNAL   DNS          41m
checkoutservice-entry         [checkoutservice.hip2.global]        MESH_INTERNAL   DNS          41m
```

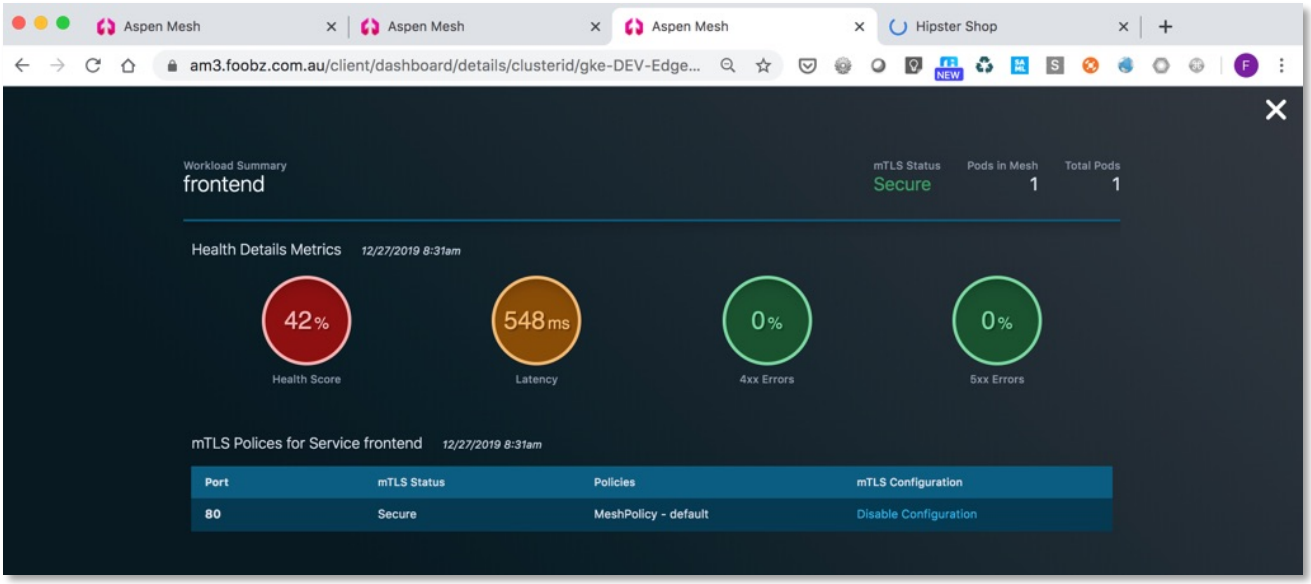
```
currencyservice-entry              [currencyservice.hip2.global]          MESH_INTERNAL   DNS          41m
shippingservice-entry              [shippingservice.hip2.global]          MESH_INTERNAL   DNS          41m
whitelist-egress-google-metadata   [metadata.google.internal]                                          41m
whitelist-egress-googleapis        [accounts.google.com *.googleapis.com]                           41m


Note:
This is an optional. To replace typewriter.jpg with a modified typewriter.jpg so that we know
which copies of apps we are accessing during a global failover – for Use Case #4.

fbchan@logos:~/k8s-clusterX/gke-1/github/hipster apps/hip1-gke$ kubectl -c server cp
./typewriter.jpg hip1/$(kubectl -n hip1 get pod -l app=frontend -o
jsonpath={.items..metadata.name}):/frontend/static/img/products
```

Access hipster shop (http://gkehip.foobz.com.au) via browser via FQDN.
Note: A modified image with GCP logo on typewriter image.

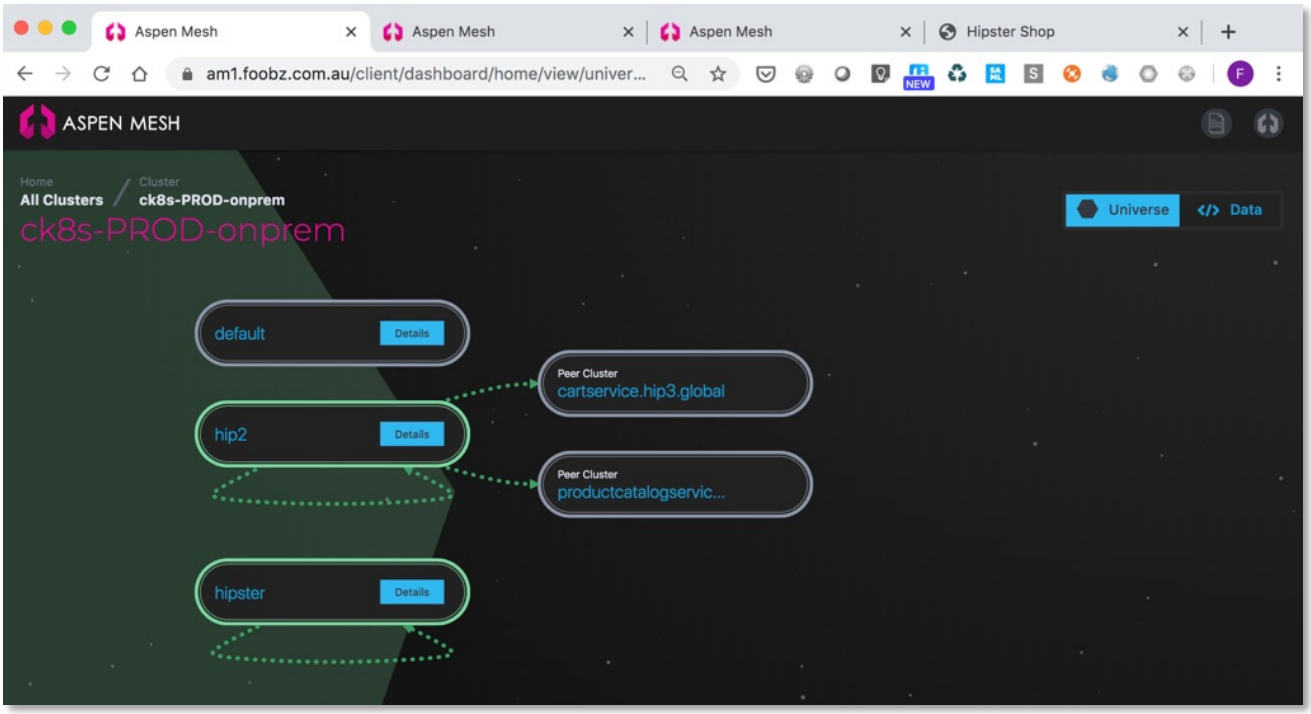


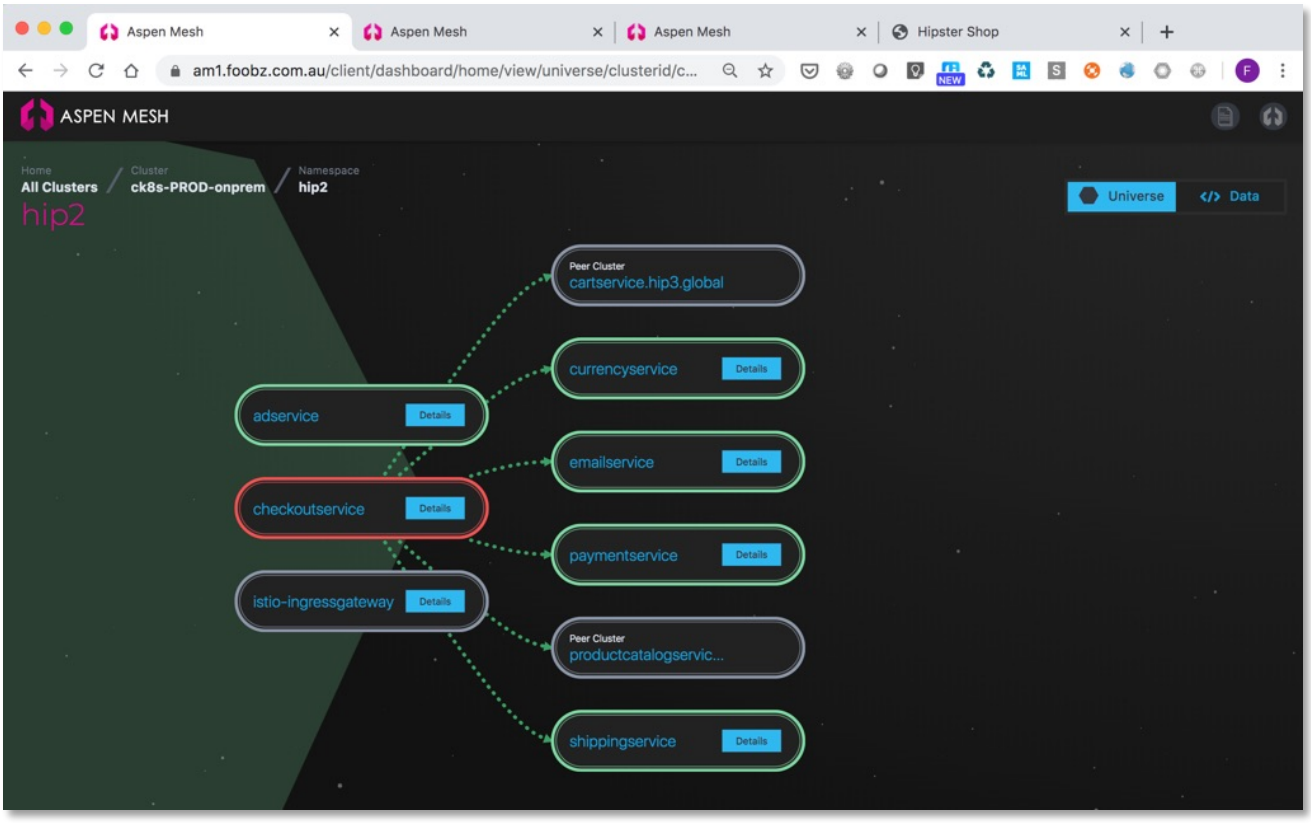
Aspen Mesh Dashboard 3 (GKE Cluster) shown all related connectivity from frontend to other services.

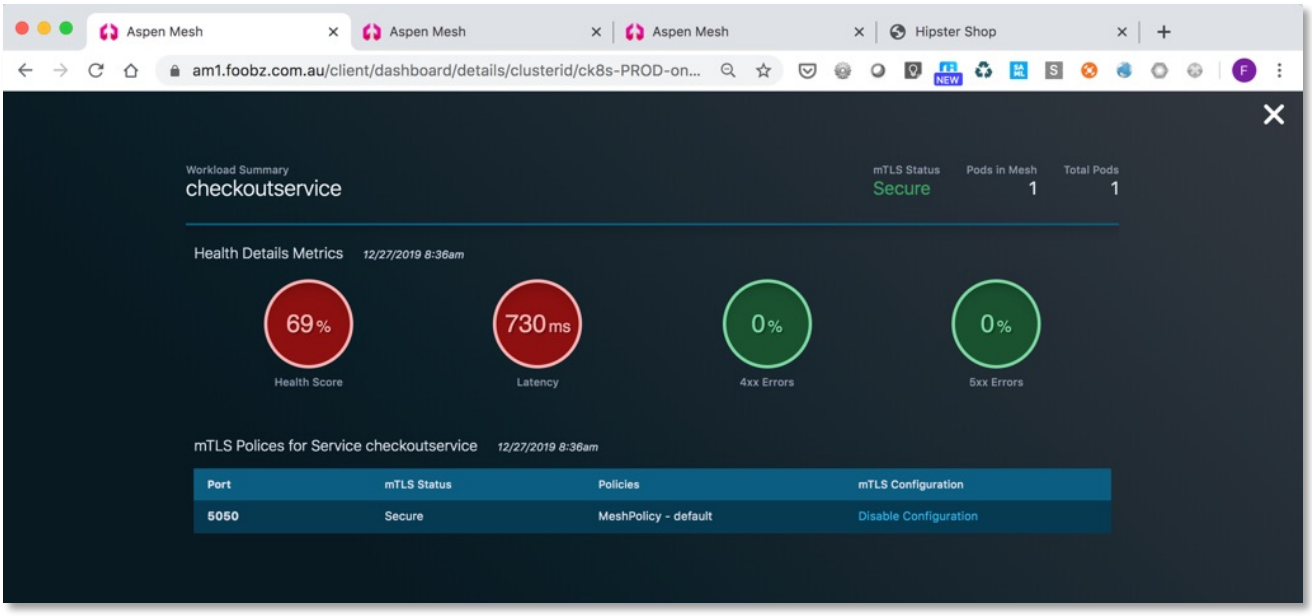




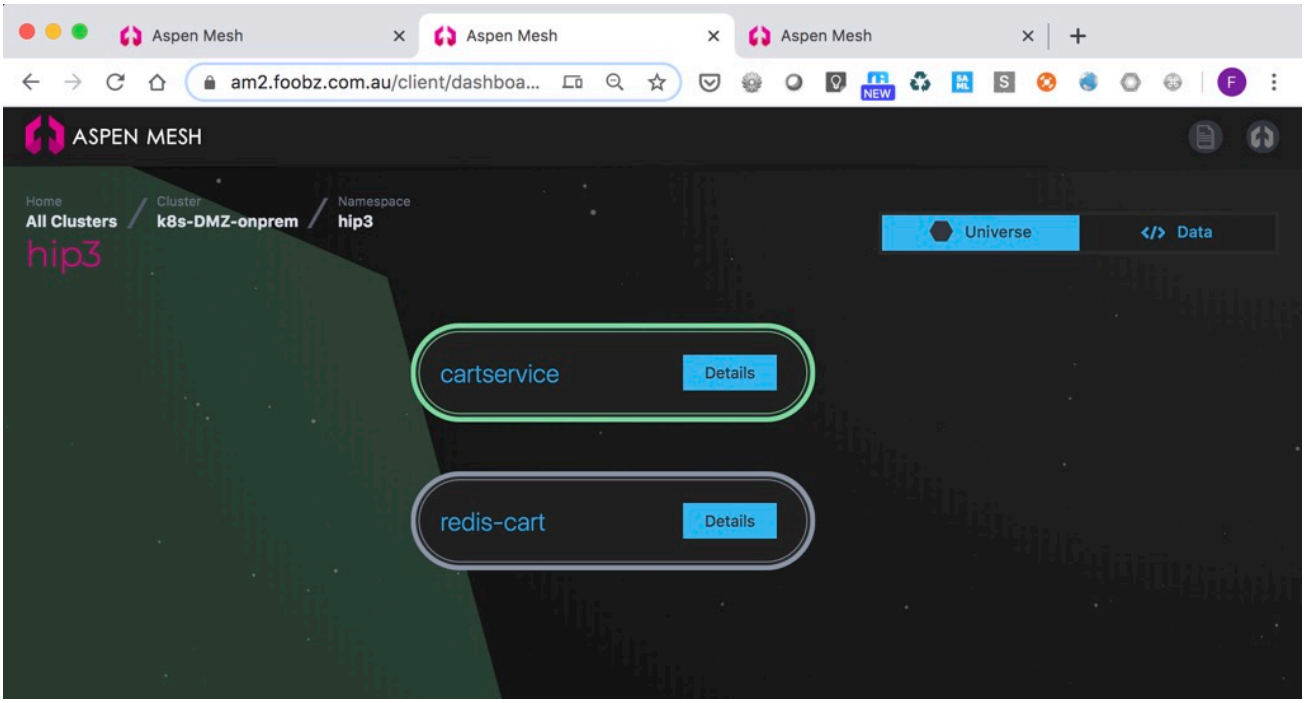
Note: Not very healthy frontend. Probably connectivity due to high latency from frontend (in GKE) to peer-cluster (on-prem) (adservice on hip2, cluster2 and hip3 on cluster3). Latency between GKE on Google cloud (deployed in SouthEast Asia Region) via Internet link to on-prem cluster in Melbourne, Australia. In addition, cK8s-PROD-onprem ('cluster2') and k8s-DMZ-onprem ('cluster3') Kubernetes are highly utilized cluster.


Aspen Mesh
Aspen Mesh
Aspen Mesh
Hipster Shop
am1.foobz.com.au/client/dashboard/home/view/univer...
ASPEN MESH
Home
All Clusters
Cluster
ck8s-PROD-onprem
ck8s-PROD-onprem
Universe
Data
default
Details
hip2
Details
hipster
Details
Peer Cluster
cartservice.hip3.global
Peer Cluster
productcatalogservic...

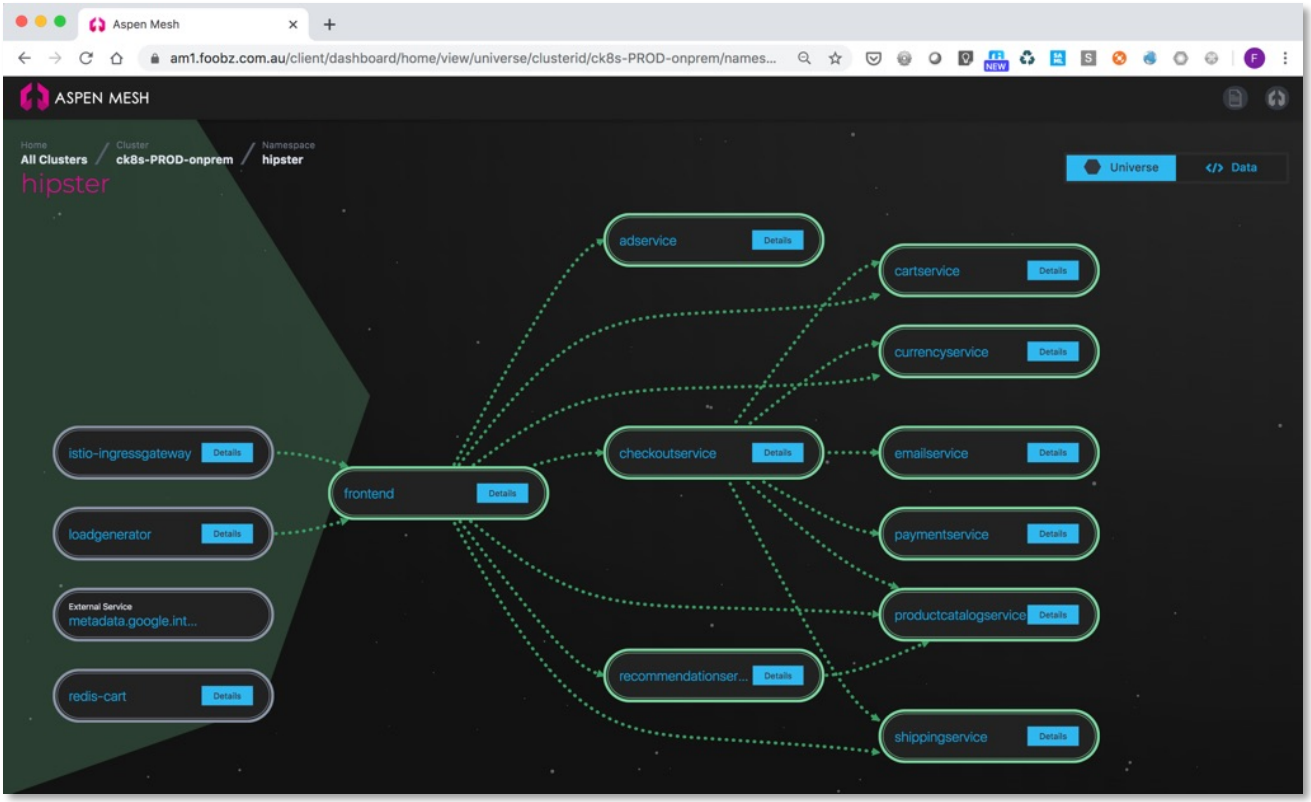

Aspen Mesh
Aspen Mesh
Aspen Mesh
Hipster Shop
am1.foobz.com.au/client/dashboard/home/view/universe/clusterid/c...
ASPEN MESH
Home
All Clusters
Cluster
ck8s-PROD-onprem
Namespace
hip2
hip2
Universe
Data
Peer Cluster
cartservice.hip3.global
currencyservice
Details
adservice
Details
emailservice
Details
checkoutservice
Details
paymentservice
Details
istio-ingressgateway
Details
Peer Cluster
productcatalogservic...
shippingservice
Details



Note: Example shown above conclude that there is no error for the inter cluster communication. Health score low due to high latency.



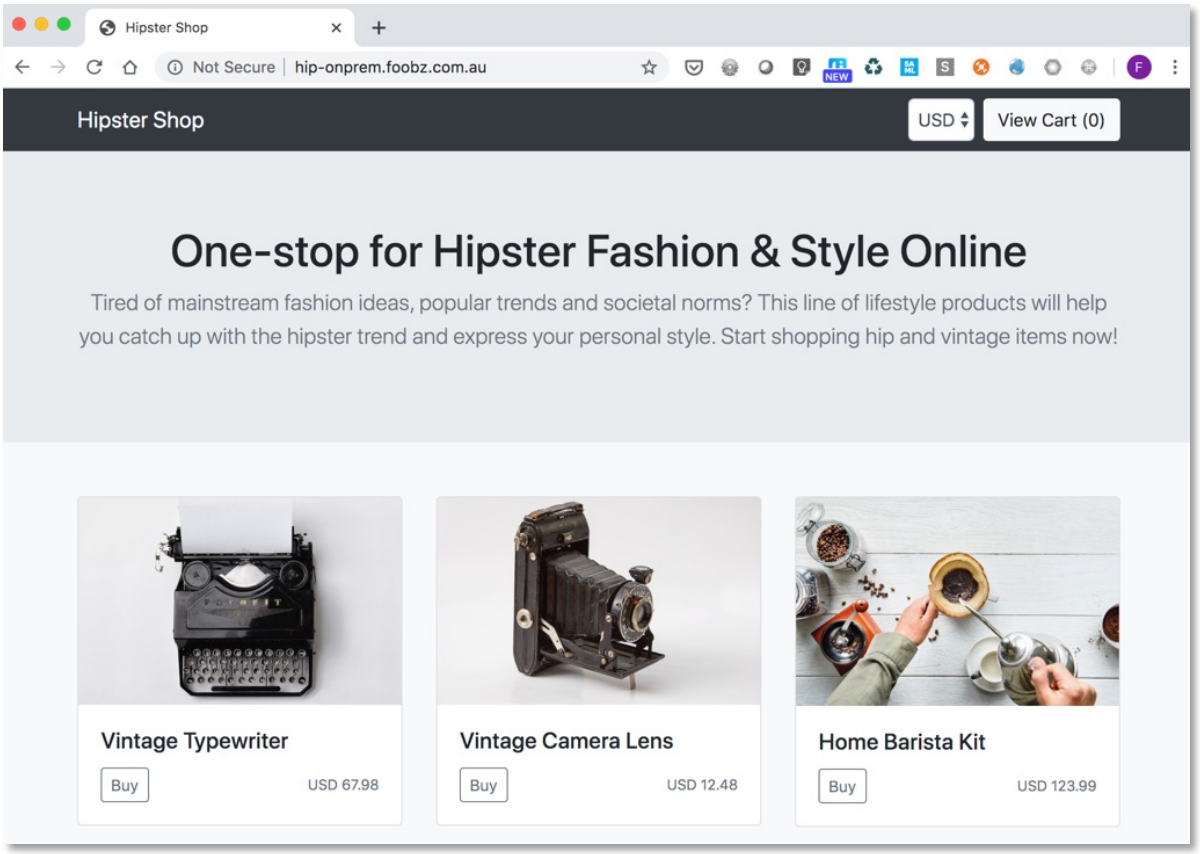
This is a backup namespace (hipster) and will only be use in the even if primary hipster apps not available. Note: This apps can also be configured to leverage cross namespace where namespace in hipster accessing services on namespace in hip2 on the same cluster. This will eliminate duplicate pods or services in hipster namespace.




```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am3$ kubectl -n hipster get pod
NAME                                     READY   STATUS    RESTARTS   AGE
adservice-55f9757757-lhbhp               2/2     Running   0          18h
cartservice-684bb46b44-7q55j             2/2     Running   1          18h
checkoutservice-6fcc84467f-zmscc         2/2     Running   0          18h
currencyservice-6c7c479d45-tnz4d         2/2     Running   0          18h
emailservice-8dd9b76cc-zs64s             2/2     Running   0          18h
frontend-7d8cfc75b5-dqj7c                2/2     Running   0          18h
loadgenerator-5db67d555-r6ptx            2/2     Running   3          18h
paymentservice-84ffc75c55-6gw42          2/2     Running   0          18h
productcatalogservice-d564bdf4c-gh59c    2/2     Running   0          18h
recommendationservice-76598d5889-d4wpg   2/2     Running   0          18h
redis-cart-5f59546cdd-4f225              2/2     Running   0          18h
shippingservice-b6db65f7f-dv7lg          2/2     Running   0          18h


fbchan@ck8s-1:~/aspenmesh-1.3.6-am3$ kubectl -n hipster get svc
NAME                    TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
adservice               ClusterIP      10.100.172.224   <none>        9555/TCP       18h
cartservice             ClusterIP      10.107.64.194    <none>        7070/TCP       18h
checkoutservice         ClusterIP      10.107.216.3     <none>        5050/TCP       18h
currencyservice         ClusterIP      10.105.56.102    <none>        7000/TCP       18h
emailservice            ClusterIP      10.111.66.106    <none>        5000/TCP       18h
frontend                ClusterIP      10.108.54.127    <none>        80/TCP         18h
frontend-external       LoadBalancer   10.97.176.165    <pending>     80:31991/TCP   18h
paymentservice          ClusterIP      10.109.100.151   <none>        50051/TCP      18h
productcatalogservice   ClusterIP      10.107.167.33    <none>        3550/TCP       18h
recommendationservice   ClusterIP      10.107.57.230    <none>        8080/TCP       18h
redis-cart              ClusterIP      10.108.203.177   <none>        6379/TCP       18h
shippingservice         ClusterIP      10.110.83.153    <none>        50051/TCP      18h
```

```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am3$ kubectl -n hipster get gw
NAME               AGE
frontend-gateway   18h

fbchan@ck8s-1:~/aspenmesh-1.3.6-am3$ kubectl -n hipster get vs
NAME               GATEWAYS             HOSTS                                                AGE
frontend                                [frontend.hipster.svc.cluster.local]                 18h
frontend-ingress   [frontend-gateway]   [hip-onprem.foobz.com.au hipster.foobz.com.au]        18h
```

http://hip-onprem.foobz.com.au (on prem version of hipster – without GCP logo on typewriter image)



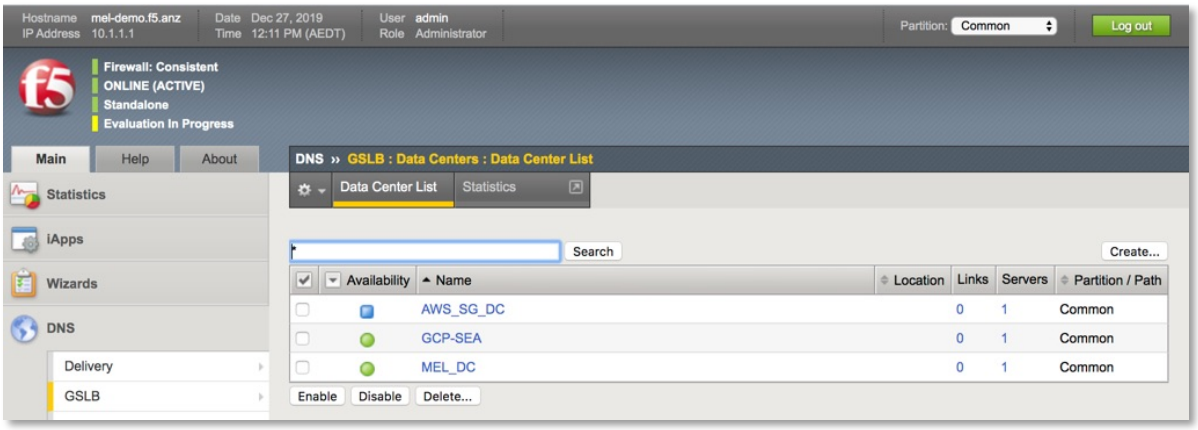
5.4 Use Case#4: Application resiliency and global service failover of Google hipster application

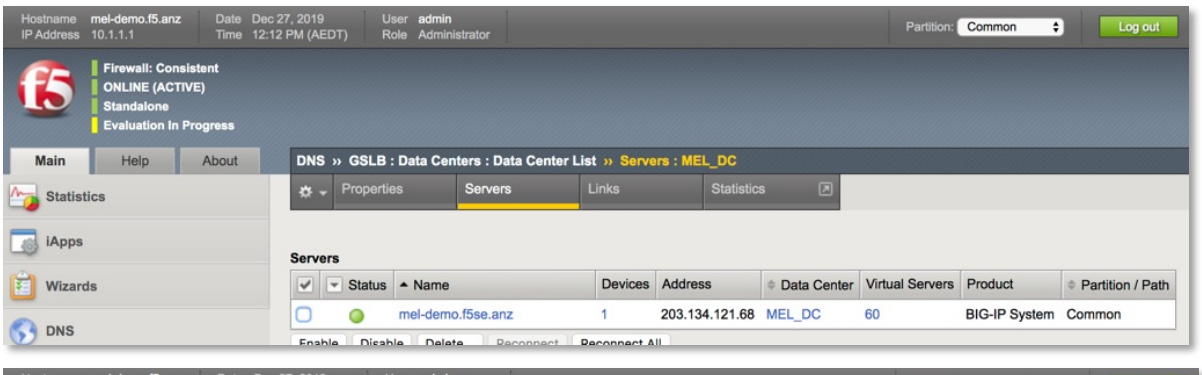
To configure application resiliency, F5 DNS/GTM (Global Traffic Manager) will be configured to enable intelligent traffic redirection (via DNS) from GKE on Google Cloud Platform (GCP) to on-prem Kubernetes Platform on private cloud. Please refer to F5 official documentation to configure and enable DNS/GTM function.

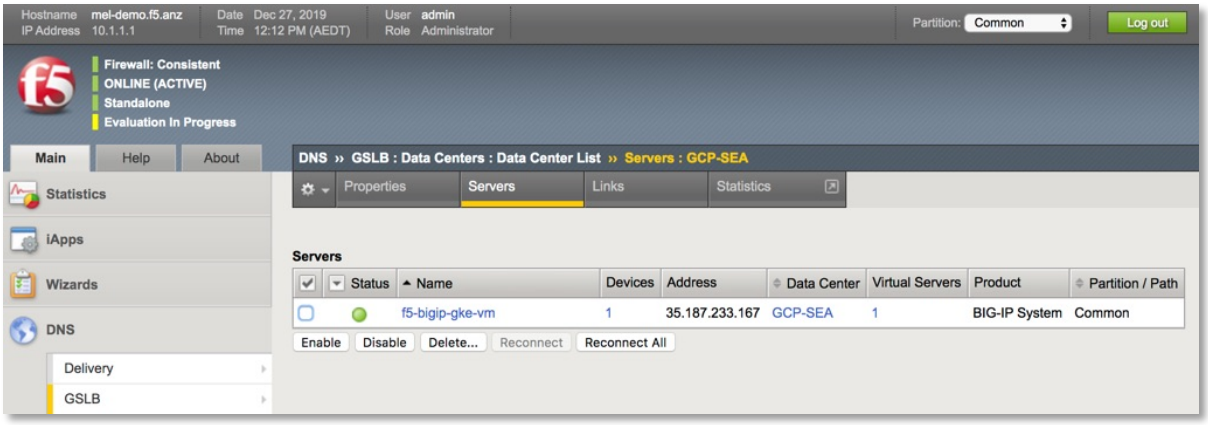
On-Prem BIG-IP

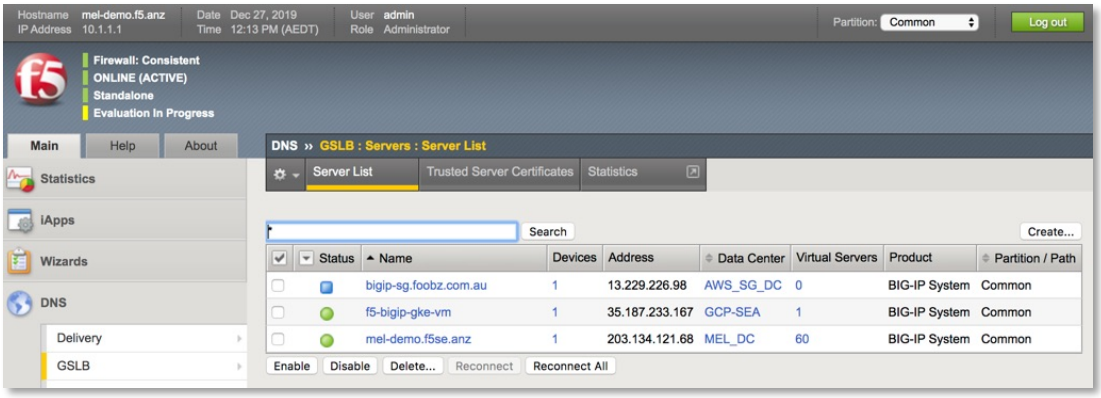
Add remote BIG-IP

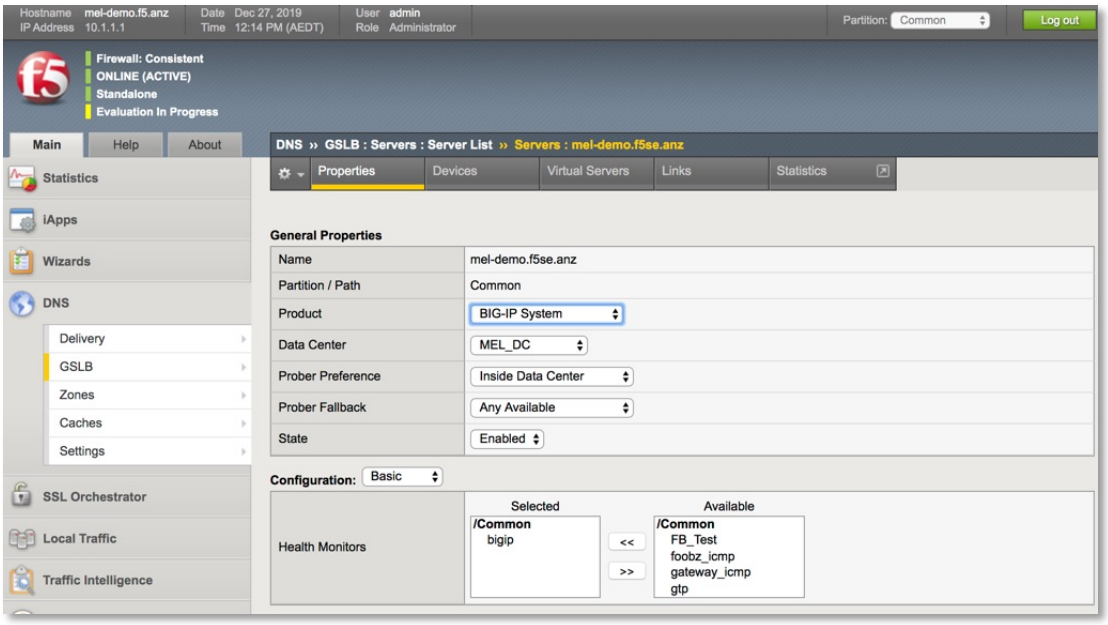
```
[root@mel-demo:TimeLimitedModules::Active:Standalone] config # bigip_add 35.187.233.167
Retrieving remote and installing local BIG-IP's SSL certs ...
Enter root password for 35.187.233.167 if prompted
The authenticity of host '35.187.233.167 (35.187.233.167)' can't be established.
RSA key fingerprint is SHA256:TgO2EQcdkaxz+TCD0M4umiwC1+vdDS+zdkCzMzb+Lig.
RSA key fingerprint is MD5:c7:68:d6:8f:3b:f6:95:49:c6:30:35:78:23:b1:2b:e1.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '35.187.233.167' (RSA) to the list of known hosts.
Password:

==> Done <==
```

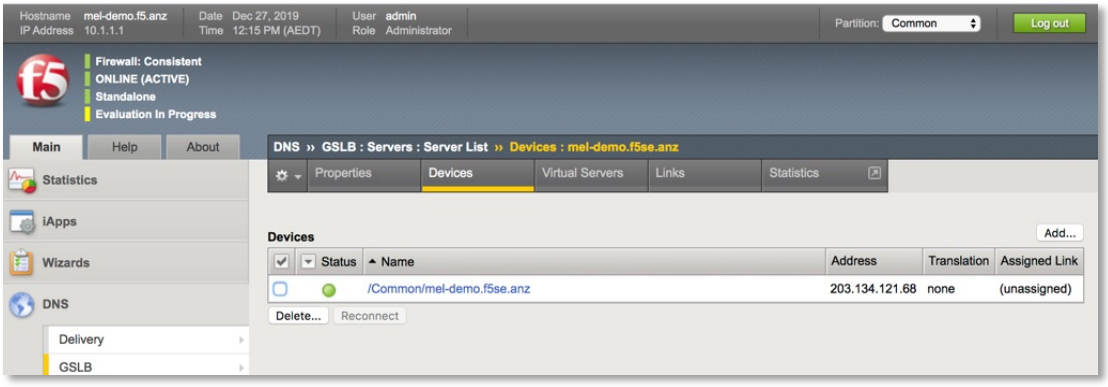


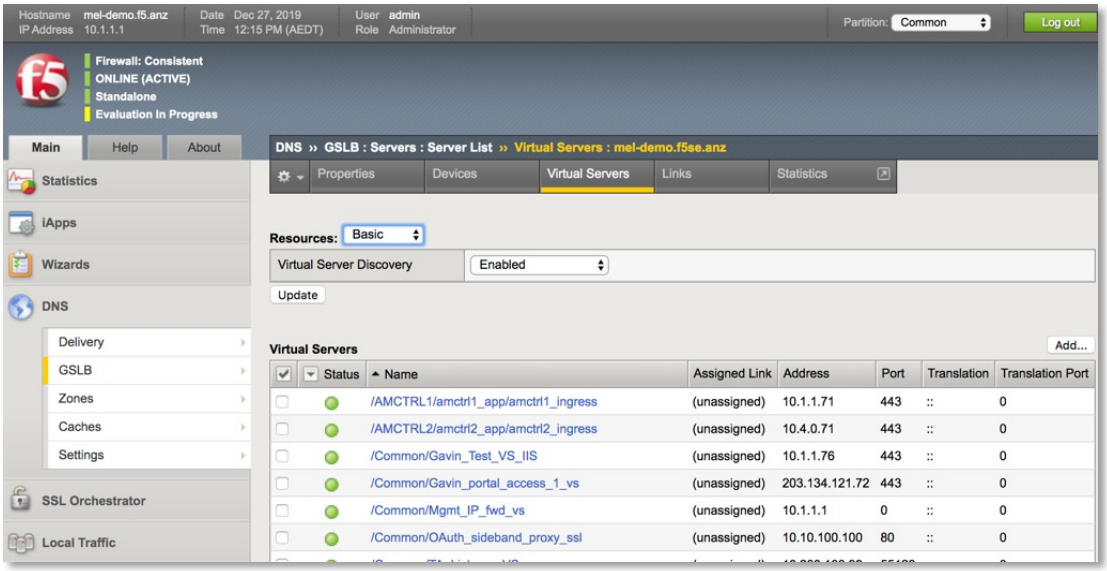


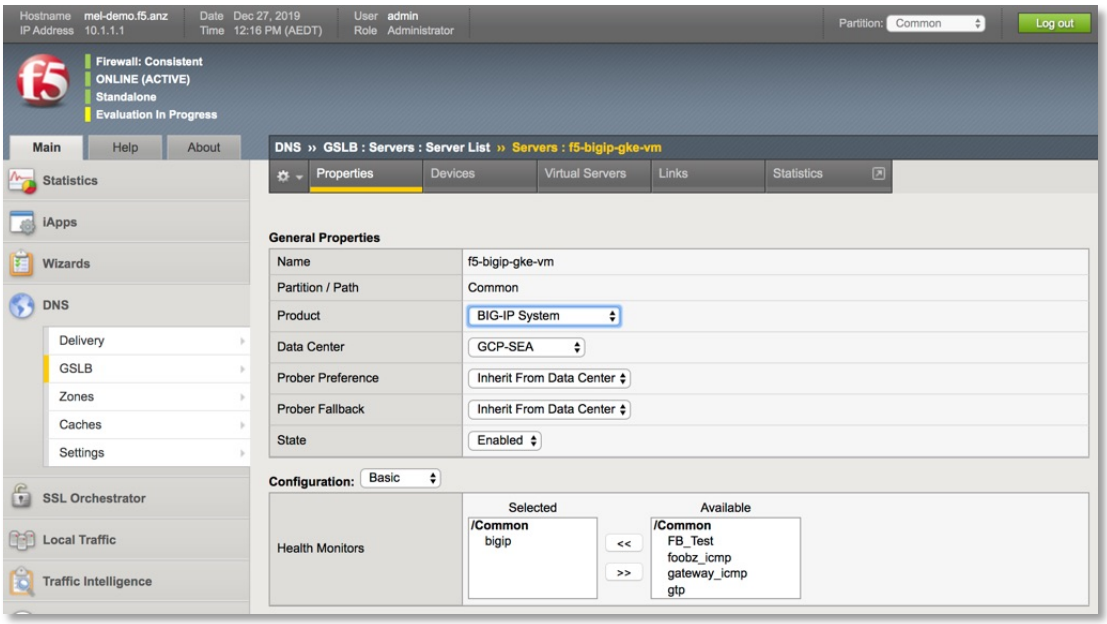


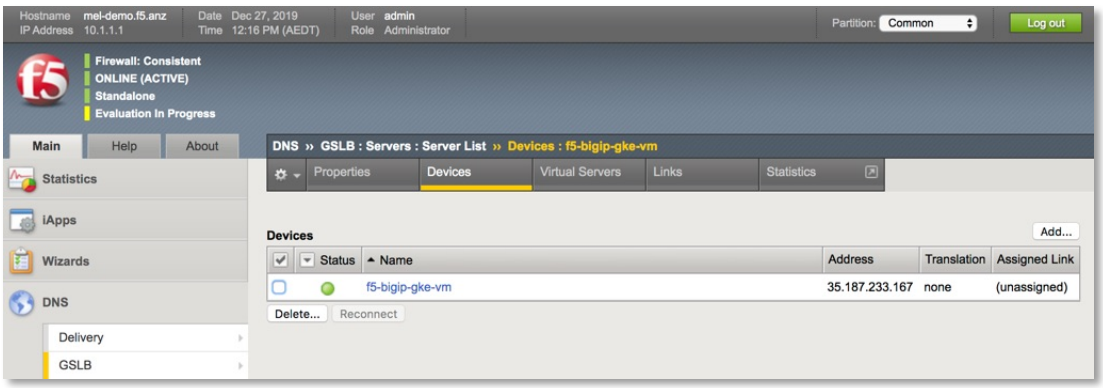

Hostname mel-demo.f5.anz
IP Address 10.1.1.1
Date Dec 27, 2019
Time 12:13 PM (AEDT)
User admin
Role Administrator
Partition: Common
Log out
Firewall: Consistent
ONLINE (ACTIVE)
Standalone
Evaluation In Progress
Main
Help
About
DNS » GSLB : Servers : Server List
Server List
Trusted Server Certificates
Statistics
Statistics
iApps
Wizards
DNS
Delivery
GSLB
Search
Create...
Status
Name
Devices
Address
Data Center
Virtual Servers
Product
Partition / Path
bigip-sg.foobz.com.au 1 13.229.226.98 AWS_SG_DC 0 BIG-IP System Common
f5-bigip-gke-vm 1 35.187.233.167 GCP-SEA 1 BIG-IP System Common
mel-demo.f5se.anz 1 203.134.121.68 MEL_DC 60 BIG-IP System Common
Enable
Disable
Delete...
Reconnect
Reconnect All

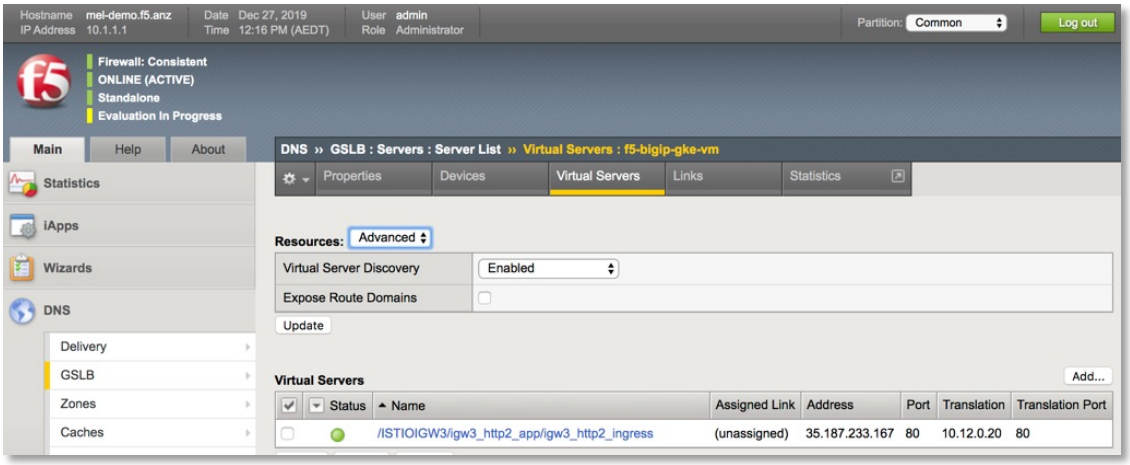

Hostname mel-demo.f5.anz
IP Address 10.1.1.1
Date Dec 27, 2019
Time 12:14 PM (AEDT)
User admin
Role Administrator
Partition: Common
Log out
Firewall: Consistent
ONLINE (ACTIVE)
Standalone
Evaluation In Progress
Main
Help
About
DNS » GSLB : Servers : Server List » Servers : mel-demo.f5se.anz
Properties
Devices
Virtual Servers
Links
Statistics
Statistics
iApps
Wizards
DNS
Delivery
GSLB
Zones
Caches
Settings
SSL Orchestrator
Local Traffic
Traffic Intelligence
General Properties
Name mel-demo.f5se.anz
Partition / Path Common
Product BIG-IP System
Data Center MEL_DC
Prober Preference Inside Data Center
Prober Fallback Any Available
State Enabled
Configuration: Basic
Health Monitors
Selected
/Common
bigip
<<
>>
Available
/Common
FB_Test
foobz_icmp
gateway_icmp
gtp

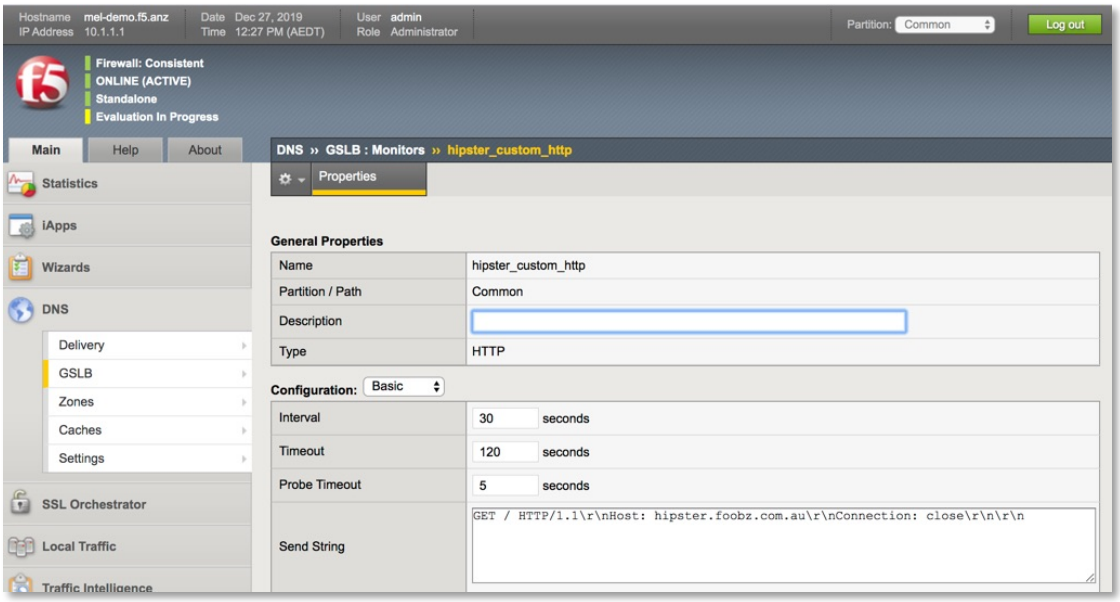

Hostname mel-demo.f5.anz
IP Address 10.1.1.1
Date Dec 27, 2019
Time 12:15 PM (AEDT)
User admin
Role Administrator
Partition: Common
Log out
Firewall: Consistent
ONLINE (ACTIVE)
Standalone
Evaluation In Progress
Main
Help
About
DNS » GSLB : Servers : Server List » Devices : mel-demo.f5se.anz
Properties
Devices
Virtual Servers
Links
Statistics
Statistics
iApps
Wizards
DNS
Delivery
GSLB
Devices
Add...
Status
Name
Address
Translation
Assigned Link
/Common/mel-demo.f5se.anz 203.134.121.68 none (unassigned)
Delete...
Reconnect

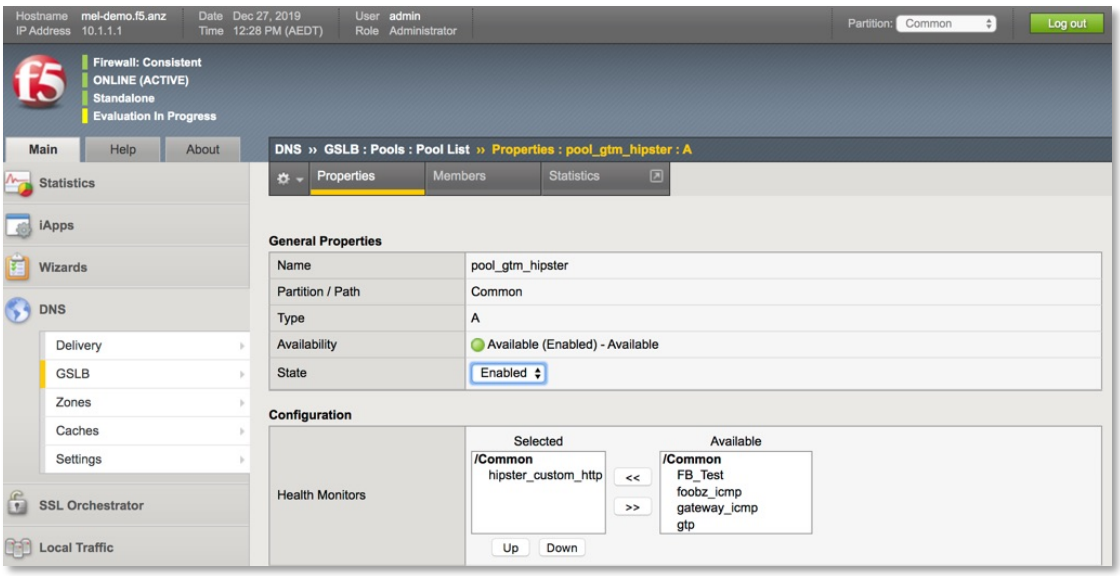

Hostname mel-demo.f5.anz
IP Address 10.1.1.1
Date Dec 27, 2019
Time 12:15 PM (AEDT)
User admin
Role Administrator
Partition: Common
Log out
Firewall: Consistent
ONLINE (ACTIVE)
Standalone
Evaluation In Progress
Main
Help
About
DNS » GSLB : Servers : Server List » Virtual Servers : mel-demo.f5se.anz
Properties
Devices
Virtual Servers
Links
Statistics
Statistics
iApps
Wizards
DNS
Delivery
GSLB
Zones
Caches
Settings
SSL Orchestrator
Local Traffic
Resources: Basic
Virtual Server Discovery
Enabled
Update
Virtual Servers
Add...
Status
Name
Assigned Link
Address
Port
Translation
Translation Port
/AMCTRL1/amctrl1_app/amctrl1_ingress (unassigned) 10.1.1.71 443 :: 0
/AMCTRL2/amctrl2_app/amctrl2_ingress (unassigned) 10.4.0.71 443 :: 0
/Common/Gavin_Test_VS_IIS (unassigned) 10.1.1.76 443 :: 0
/Common/Gavin_portal_access_1_vs (unassigned) 203.134.121.72 443 :: 0
/Common/Mgmt_IP_fwd_vs (unassigned) 10.1.1.1 0 :: 0
/Common/OAuth_sideband_proxy_ssl (unassigned) 10.10.100.100 80 :: 0


Hostname mel-demo.f5.anz
IP Address 10.1.1.1
Date Dec 27, 2019
Time 12:16 PM (AEDT)
User admin
Role Administrator
Partition: Common
Log out
Firewall: Consistent
ONLINE (ACTIVE)
Standalone
Evaluation In Progress
Main
Help
About
DNS » GSLB : Servers : Server List » Servers : f5-bigip-gke-vm
Properties
Devices
Virtual Servers
Links
Statistics
Statistics
iApps
Wizards
DNS
Delivery
GSLB
Zones
Caches
Settings
SSL Orchestrator
Local Traffic
Traffic Intelligence
General Properties
Name f5-bigip-gke-vm
Partition / Path Common
Product BIG-IP System
Data Center GCP-SEA
Prober Preference Inherit From Data Center
Prober Fallback Inherit From Data Center
State Enabled
Configuration: Basic
Health Monitors
Selected
/Common
bigip
<<
>>
Available
/Common
FB_Test
foobz_icmp
gateway_icmp
gtp

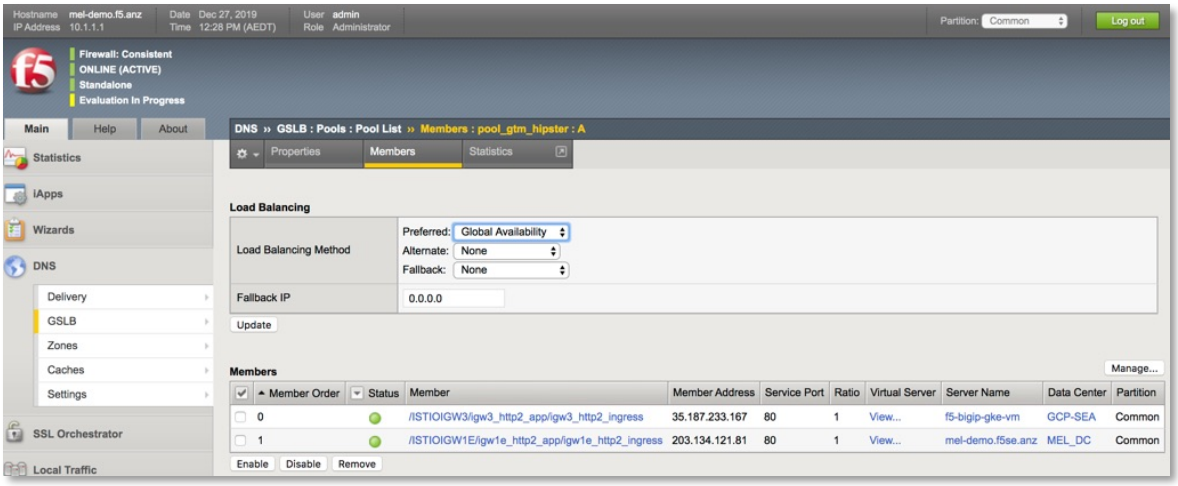

Hostname mel-demo.f5.anz
IP Address 10.1.1.1
Date Dec 27, 2019
Time 12:16 PM (AEDT)
User admin
Role Administrator
Partition: Common
Log out
Firewall: Consistent
ONLINE (ACTIVE)
Standalone
Evaluation In Progress
Main
Help
About
DNS » GSLB : Servers : Server List » Devices : f5-bigip-gke-vm
Properties
Devices
Virtual Servers
Links
Statistics
Statistics
iApps
Wizards
DNS
Delivery
GSLB
Devices
Add...
Status
Name
Address
Translation
Assigned Link
f5-bigip-gke-vm 35.187.233.167 none (unassigned)
Delete...
Reconnect


Hostname mel-demo.f5.anz
IP Address 10.1.1.1
Date Dec 27, 2019
Time 12:16 PM (AEDT)
User admin
Role Administrator
Partition: Common
Log out
Firewall: Consistent
ONLINE (ACTIVE)
Standalone
Evaluation In Progress
Main
Help
About
DNS » GSLB : Servers : Server List » Virtual Servers : f5-bigip-gke-vm
Properties
Devices
Virtual Servers
Links
Statistics
Statistics
iApps
Wizards
DNS
Delivery
GSLB
Zones
Caches
Resources: Advanced
Virtual Server Discovery
Enabled
Expose Route Domains
Update
Virtual Servers
Add...
Status
Name
Assigned Link
Address
Port
Translation
Translation Port
/ISTIOIGW3/igw3_http2_app/igw3_http2_ingress
(unassigned)
35.187.233.167
80
10.12.0.20
80

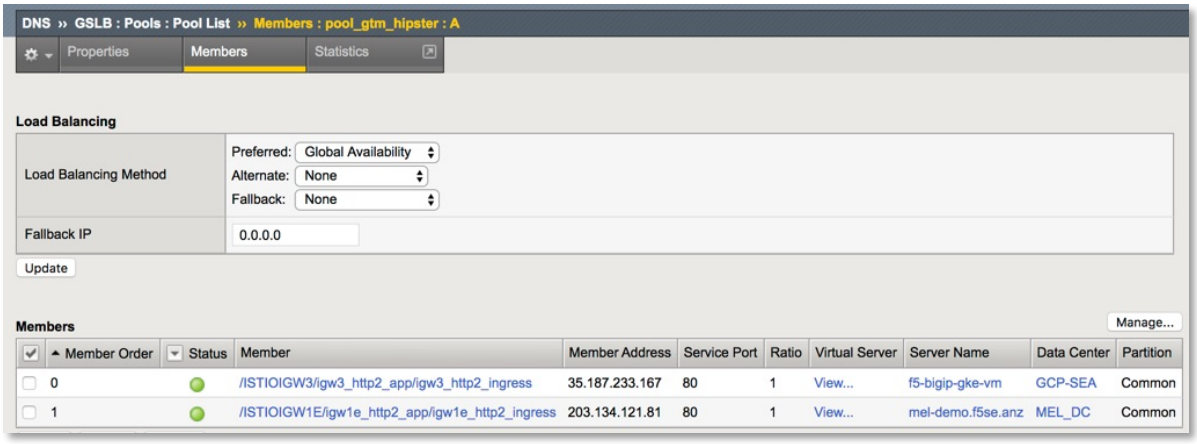

Hostname mel-demo.f5.anz
IP Address 10.1.1.1
Date Dec 27, 2019
Time 12:27 PM (AEDT)
User admin
Role Administrator
Partition: Common
Log out
Firewall: Consistent
ONLINE (ACTIVE)
Standalone
Evaluation In Progress
Main
Help
About
DNS » GSLB : Monitors » hipster_custom_http
Properties
Statistics
iApps
Wizards
DNS
Delivery
GSLB
Zones
Caches
Settings
SSL Orchestrator
Local Traffic
Traffic Intelligence
General Properties
Name
hipster_custom_http
Partition / Path
Common
Description
Type
HTTP
Configuration: Basic
Interval
30
seconds
Timeout
120
seconds
Probe Timeout
5
seconds
Send String
GET / HTTP/1.1\r\nHost: hipster.foobz.com.au\r\nConnection: close\r\n\r\n


Hostname mel-demo.f5.anz
IP Address 10.1.1.1
Date Dec 27, 2019
Time 12:28 PM (AEDT)
User admin
Role Administrator
Partition: Common
Log out
Firewall: Consistent
ONLINE (ACTIVE)
Standalone
Evaluation In Progress
Main
Help
About
DNS » GSLB : Pools : Pool List » Properties : pool_gtm_hipster : A
Properties
Members
Statistics
Statistics
iApps
Wizards
DNS
Delivery
GSLB
Zones
Caches
Settings
SSL Orchestrator
Local Traffic
General Properties
Name
pool_gtm_hipster
Partition / Path
Common
Type
A
Availability
Available (Enabled) - Available
State
Enabled
Configuration
Health Monitors
Selected
/Common
hipster_custom_http
<<
>>
Available
/Common
FB_Test
foobz_icmp
gateway_icmp
gtp
Up
Down

Global Availability configuration of F5 DNS/GTM where GKE on GCP (Istio Ingressgateway 3) the primary and Kubernetes on-prem on ck8s-PROD-onprem (Istio Ingressgateway 1E) backup/secondary.



Validate to ensure DNS working



Linux dig to hipster.foobz.com.au with responses from BIG-IP DNS with Google Cloud public IP.

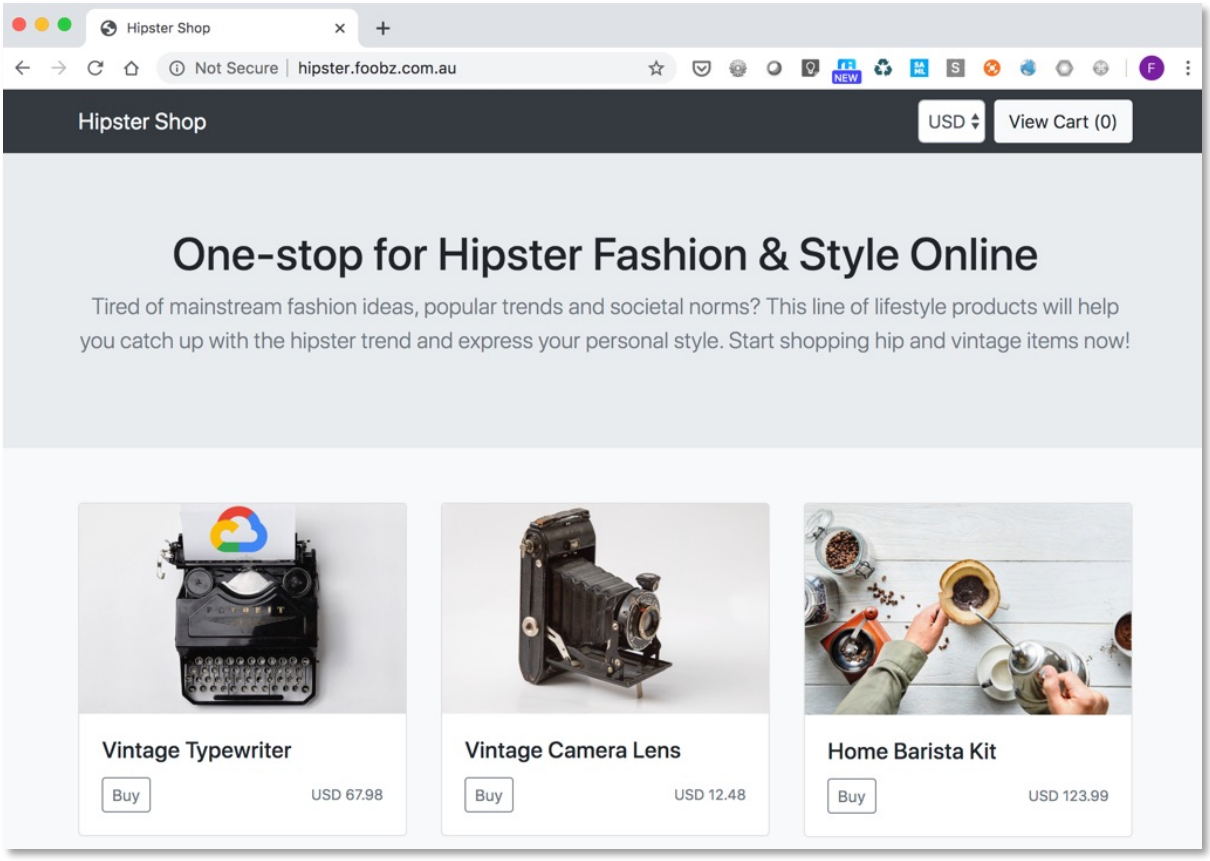
```
From remote client

fbchan@foobz-linux:~$ dig hipster.foobz.com.au

; <<>> DiG 9.10.3-P4-Ubuntu <<>> hipster.foobz.com.au
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3080
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1280
;; QUESTION SECTION:
;hipster.foobz.com.au.              IN      A

;; ANSWER SECTION:
hipster.foobz.com.au. 30      IN      A       35.187.233.167

;; Query time: 216 msec
;; SERVER: 168.63.129.16#53(168.63.129.16)
;; WHEN: Fri Dec 27 01:34:23 UTC 2019
;; MSG SIZE  rcvd: 65
```

Hipster website (http://hipster.foobz.com.au) accessible from client

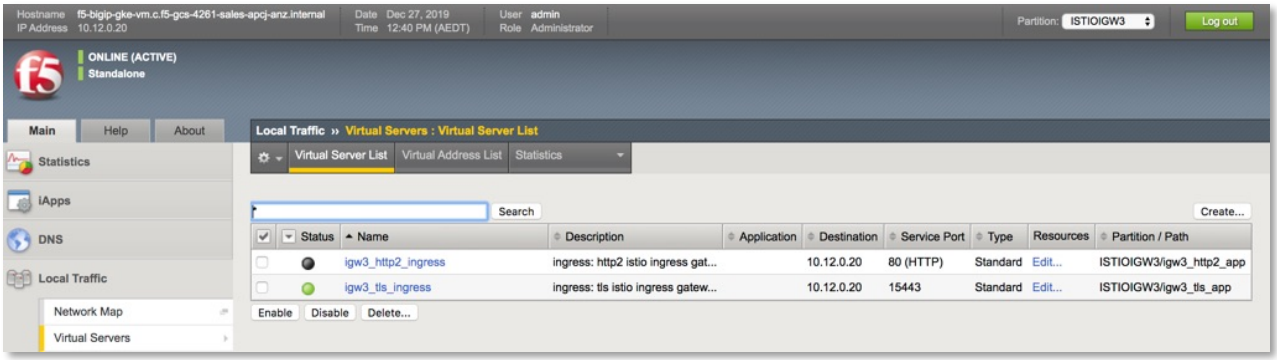
```
fbchan@foobz-linux:~$ ping hipster.foobz.com.au
PING hipster.foobz.com.au (35.187.233.167) 56(84) bytes of data.
```

```
64 bytes from 167.233.187.35.bc.googleusercontent.com (35.187.233.167): icmp_seq=1 ttl=224
time=216 ms
```

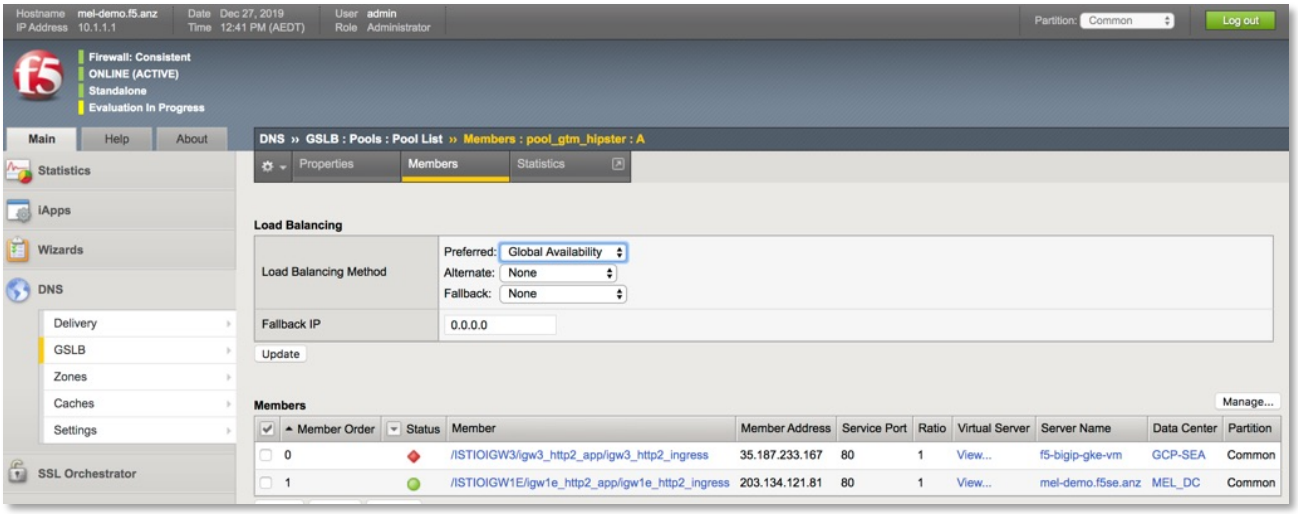
Note: This shown that the GCP copy of hipster application being fetched.



To simulate Google Cloud GKE unavailable, disable Istio ingressgateway on port 80 virtual server in Google Cloud BIG-IP. Alternatively, this can also be simulated by deleting frontend deployment in Google Cloud GKE.



F5 DNS will mark remote Istio ingressgateway as unavailable.



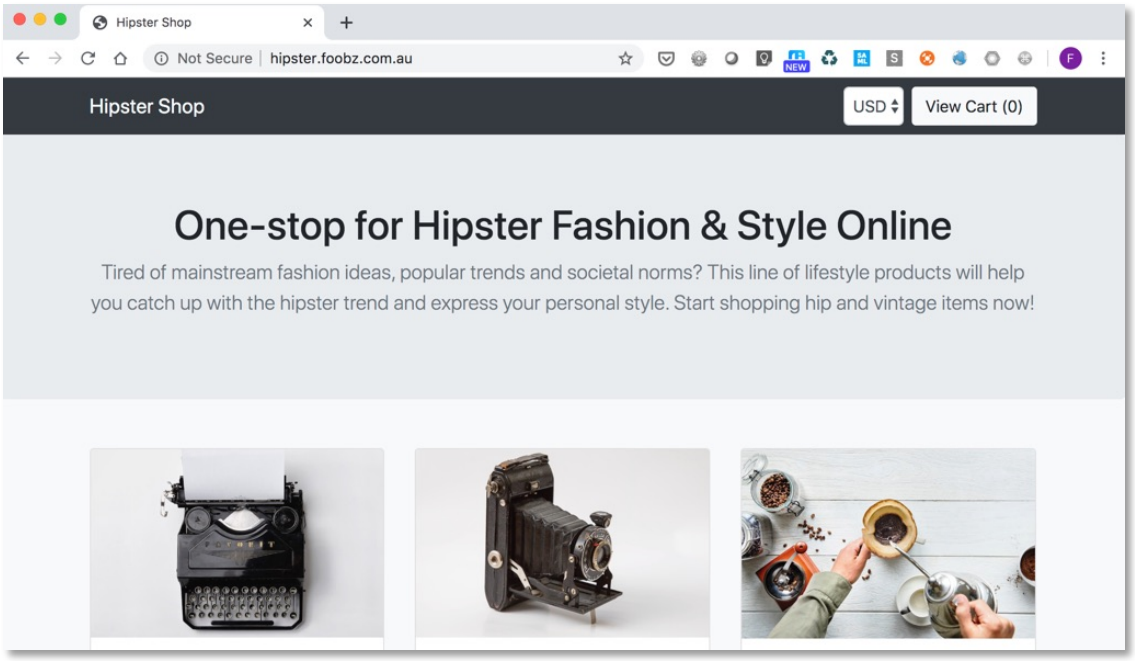
Linux dig to hipster.foobz.com.au with responses from BIG-IP DNS with on-prem public IP address.

```
From remote client

fbchan@foobz-linux:~$ dig hipster.foobz.com.au

; <<>> DiG 9.10.3-P4-Ubuntu <<>> hipster.foobz.com.au
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22995
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1280
;; QUESTION SECTION:
;hipster.foobz.com.au.              IN      A

;; ANSWER SECTION:
hipster.foobz.com.au. 30      IN      A       203.134.121.81

;; Query time: 215 msec
;; SERVER: 168.63.129.16#53(168.63.129.16)
;; WHEN: Fri Dec 27 01:42:58 UTC 2019
;; MSG SIZE  rcvd: 65
```

Hipster website (http://hipster.foobz.com.au) still accessible from client

```
fbchan@foobz-linux:~$ ping hipster.foobz.com.au
PING hipster.foobz.com.au (203.134.121.81) 56(84) bytes of data.
```

Note: This shown that the on-prem copy of hipster application being fetched (without GCP logo).



Re-Enable Google Cloud BIG-IP Istio ingressgateway virtual server will redirect traffic back to Google Cloud GKE.

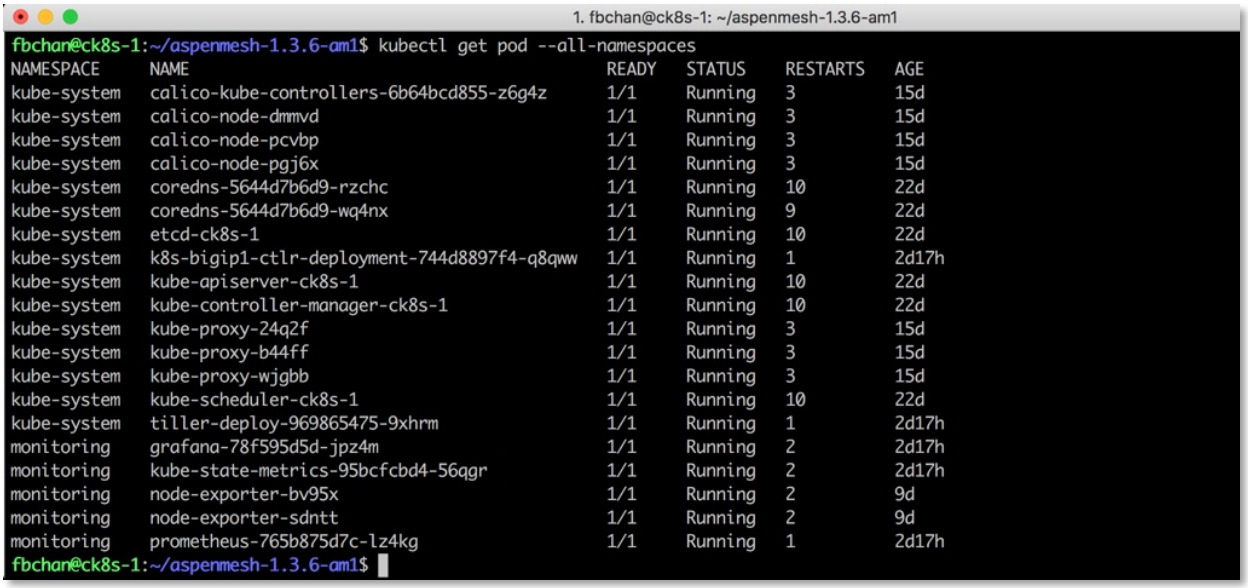
6 Aspen Mesh Multi-Cluster Deployment Guide

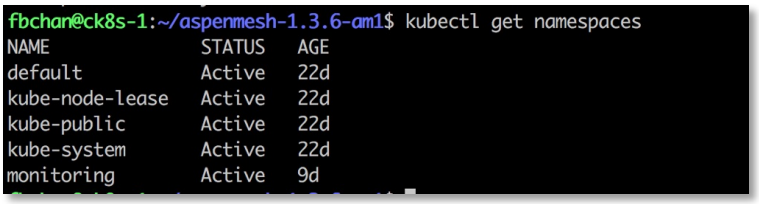
Detail installation instruction can be obtained from Aspen Mesh official site https://my.aspenmesh.io/client/docs/latest/getting-started/ and/or official Istio and Kubernetes websites.

6.1 Aspen Mesh Installation

This installation guide is based on Aspen Mesh v1.3.6 am1 where only Single Aspen Mesh Dashboard is supported. Please refer to Aspen Mesh Upgrade instructions to upgrade to Aspen Mesh v1.3.6 am3 for a preview/development release to enable Multi-Cluster Aspen Mesh Dashboard.

Pods prior installation of Aspen Mesh





Create istio-system namespace

```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl create namespace istio-system
namespace/istio-system created
```

Create Kubernetes secret for generated CA

Please ensure names are exactly as shown. Common Root CA are being use across all Aspen Mesh cluster. Each Aspen Mesh cluster will have its own dedicated CA and key.

```
kubectl create secret generic cacerts -n istio-system \
    --from-file=./ca-cert.pem \
    --from-file=./ca-key.pem \
    --from-file=./root-cert.pem \
    --from-file=./cert-chain.pem

Note:
```

```
Repeat this on respective ca directory for other cluster.

fbchan@ck8s-1:~/foobz-istio-cert/ca1$ kubectl create secret generic cacerts -n istio-system \
>      --from-file=./ca-cert.pem \
>      --from-file=./ca-key.pem \
>      --from-file=./root-cert.pem \
>      --from-file=./cert-chain.pem
secret/cacerts created


fbchan@ck8s-1:~/foobz-istio-cert/ca1$ kubectl -n istio-system describe secret cacerts
Name:         cacerts
Namespace:    istio-system
Labels:       <none>
Annotations:  <none>

Type:  Opaque

Data
====
ca-cert.pem:     2122 bytes
ca-key.pem:      2484 bytes
cert-chain.pem:  2122 bytes
root-cert.pem:   2228 bytes
```

Ensure admissionregistration enable

Ensure MutatingAdmissionWebhook and ValidatingAdmissionWebhook admission controllers enabled in the API server (default)

```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl api-versions | grep admissionregistration
admissionregistration.k8s.io/v1
admissionregistration.k8s.io/v1beta1
```

Install istio-init

Use helm (must be helm version prior 3.x) to install istio-init.

Note: This installation instruction based on production release of Aspen Mesh v1.3.6 am1. To have Aspen Mesh multi-cluster observability and visibility (preview), you can either upgrade after installation or directly install Aspen Mesh v1.3.6 am3.

```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ helm install install/kubernetes/helm/istio-init --name
istio-init --namespace istio-system
NAME:   istio-init
LAST DEPLOYED: Mon Dec 16 20:12:08 2019
NAMESPACE: istio-system
STATUS: DEPLOYED

RESOURCES:
==> v1/ClusterRole
NAME                     AGE
istio-init-istio-system  0s

==> v1/ClusterRoleBinding
NAME                                        AGE
istio-init-admin-role-binding-istio-system  0s

==> v1/ConfigMap
NAME          AGE
istio-crd-10  0s
istio-crd-11  0s
istio-crd-12  0s
```

```
==> v1/Job
NAME                      AGE
istio-init-crd-10-1.3.6   0s
istio-init-crd-11-1.3.6   0s
istio-init-crd-12-1.3.6   0s

==> v1/Pod(related)
NAME                            AGE
istio-init-crd-10-1.3.6-pr8n7   0s
istio-init-crd-11-1.3.6-zzvmb   0s
istio-init-crd-12-1.3.6-59jq6   0s

==> v1/ServiceAccount
NAME                          AGE
istio-init-service-account    0s
```

```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ helm list
NAME            REVISION    UPDATED                     STATUS    CHART                     APP VERSION     NAMESPACE
istio-init      1           Mon Dec 16 20:12:08 2019    DEPLOYED  istio-init-1.3.6-am1      1.3.6-am1       istio-
system
```

Ensure all 23 Istio CRDs were committed to the Kubernetes apiserver

```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl get crds | grep 'istio.io\|certmanager.k8s.io\|aspenmesh.io' |
wc -l
23
```

Ensure values are setup as below to enable multi-cluster Aspen Mesh

```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ cat install/kubernetes/helm/istio/values-aspenmesh.yaml
global:
  controlPlaneSecurityEnabled: true
  mtls:
    enabled: true

  imagePullSecrets:
  - aspen-mesh-pull-secret

  proxy:
    accessLogFile: "/dev/stdout"

  outboundTrafficPolicy:
    mode: REGISTRY_ONLY

  tracer:
    zipkin:
      # address is the Host:Port for reporting trace data in zipkin format. If
      # not specified, will default to zipkin service (port 9411) in the same
      # namespace as the other istio components.
      address: ""

  ##### MULTICLUSTER #######
  podDNSSearchNamespaces:
  - global
  - "{{ valueOrDefault .DeploymentMeta.Namespace \"default\" }}.global"

  multiCluster:
    enabled: true
  ##########################

pilot:
  # Aspen Mesh has decided to disable this experimental feature until
  # both outbound and inbound protocol sniffing is merged and is
  # stable
  enableProtocolSniffingForOutbound: false

aspen-mesh-controlplane:
  enabled: true
```

```
  image: quay.io/aspenmesh/releases:controlplane-1.3.6-am1

  imagePullSecrets:
  - aspenmesh-kubernetes-pull-secret

  replicaCount: 2

  clusterId: ck8s-PROD-onprem

  userAuth:
    type: none
    #jwt:
      # jwks must provide a valid JWKs endpoint.
      # jwks:
      # redirectUrl must provide an unauthenticated redirect URL.
      #redirectUrl:
      # claims can be added here to require claims be present in the JWT
      # (e.g.: `claims: aud=aspenmesh:io,role=k8s:admin`).

  # prometheusUrl must provide a valid Prometheus URL.
  prometheusUrl: http://prometheus.monitoring.svc.cluster.local:9090

  resources:
    requests:
      memory: "128Mi"
      cpu: "100m"
    limits:
      memory: "128Mi"
      cpu: "100m"

aspen-mesh-dashboard:
  enabled: true
  image: quay.io/aspenmesh/releases:dashboard-1.3.6-am1
  replicaCount: 2

traffic-claim-enforcer:
  enabled: true
  image: quay.io/aspenmesh/releases:traffic-claim-enforcer-1.3.6-am1
  resources: {}

prometheus:
  enabled: true


# MULTICLUSTER #######

# Multicluster with gateways requires a root CA
# Cluster local CAs are bootstrapped with the root CA.
security:
  selfSigned: false

# Provides dns resolution for service entries of form
# name.namespace.global
istiocoredns:
  enabled: true

gateways:
  istio-egressgateway:
    enabled: true
    env:
      # Needed to route traffic via egress gateway if desired.
      ISTIO_META_REQUESTED_NETWORK_VIEW: "external"
```

Install Istio with helm.

```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ helm install install/kubernetes/helm/istio --name istio --
namespace istio-system --values install/kubernetes/helm/istio/values-aspenmesh.yaml
```

```
NAME:   istio
LAST DEPLOYED: Mon Dec 16 20:14:43 2019
NAMESPACE: istio-system
STATUS: DEPLOYED

RESOURCES:
==> v1/ClusterRole
NAME                                AGE
istio-citadel-istio-system          53s
istio-galley-istio-system           53s
istio-mixer-istio-system            53s
istio-pilot-istio-system            53s
istio-reader                        53s
istio-sidecar-injector-istio-system  53s
istiocoredns                        53s
prometheus-istio-system             53s

==> v1/ClusterRoleBinding
NAME                                                AGE
istio-citadel-istio-system                          53s
istio-galley-admin-role-binding-istio-system        53s
istio-istiocoredns-role-binding-istio-system        53s
istio-mixer-admin-role-binding-istio-system         53s
istio-multi                                         53s
istio-pilot-istio-system                            53s
istio-sidecar-injector-admin-role-binding-istio-system  53s
prometheus-istio-system                             53s

==> v1/ConfigMap
NAME                                AGE
coredns                             53s
istio                               53s
istio-galley-configuration          53s
istio-security-custom-resources     53s
istio-sidecar-injector              53s
prometheus                          53s
traffic-claim-enforcer-configurator  53s

==> v1/Deployment
NAME                     AGE
aspen-mesh-controlplane  52s
aspen-mesh-dashboard     52s
istio-citadel            52s
istio-egressgateway      52s
istio-galley             52s
istio-ingressgateway     52s
istio-pilot              52s
istio-policy             52s
istio-sidecar-injector   52s
istio-telemetry          52s
istiocoredns             52s
prometheus               52s
traffic-claim-enforcer   52s

==> v1/Pod(related)
NAME                                    AGE
aspen-mesh-controlplane-77c7b69c56-dhqb7  51s
aspen-mesh-controlplane-77c7b69c56-qgpfx  51s
aspen-mesh-dashboard-64d78d7999-mgz75     52s
aspen-mesh-dashboard-64d78d7999-vnq4f     52s
istio-citadel-8445bfb9c8-6wknv            52s
istio-egressgateway-596867b5f4-qnm5r      51s
istio-galley-78f78c94db-68qss             52s
istio-ingressgateway-688d4c5c7-sjftz      51s
istio-pilot-67545c7d67-ljv9x              51s
istio-policy-6dfb4bc9f9-ddrrq             51s
```

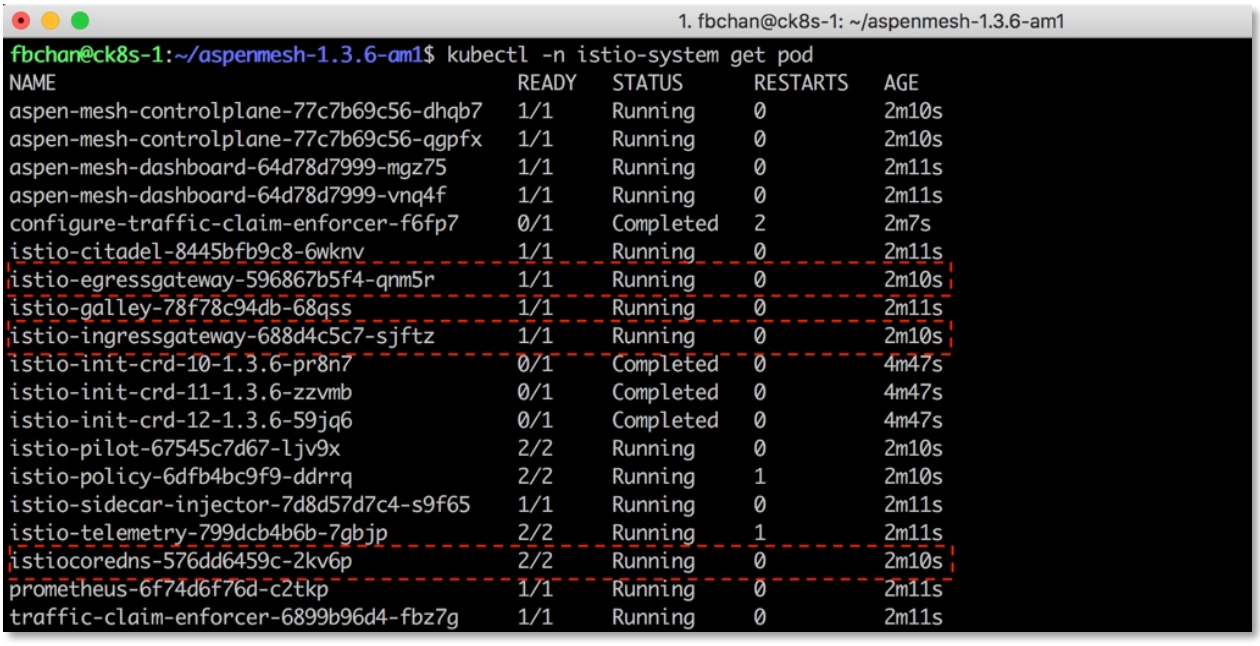
```
istio-sidecar-injector-7d8d57d7c4-s9f65   52s
istio-telemetry-799dcb4b6b-7gbjp          52s
istiocoredns-576dd6459c-2kv6p             51s
prometheus-6f74d6f76d-c2tkp               52s
traffic-claim-enforcer-6899b96d4-fbz7g    52s

==> v1/Role
NAME                                AGE
istio-ingressgateway-sds            53s
traffic-claim-enforcer-configurator  53s

==> v1/RoleBinding
NAME                       AGE
istio-ingressgateway-sds   53s

==> v1/Secret
NAME                     AGE
aspen-mesh-pull-secret   53s

==> v1/Service
NAME                            AGE
aspen-mesh-controlplane         53s
aspen-mesh-dashboard            52s
istio-citadel                   52s
istio-egressgateway             52s
istio-galley                    52s
istio-ingressgateway            52s
istio-pilot                     52s
istio-policy                    53s
istio-sidecar-injector          52s
istio-telemetry                 53s
istiocoredns                    52s
prometheus                      53s
traffic-claim-enforcer-webhook  53s

==> v1/ServiceAccount
NAME                                                AGE
aspen-mesh-controlplane                             53s
aspen-mesh-dashboard                                53s
istio-citadel-service-account                       53s
istio-egressgateway-service-account                 53s
istio-galley-service-account                        53s
istio-ingressgateway-service-account                53s
istio-mixer-service-account                         53s
istio-multi                                         53s
istio-pilot-service-account                         53s
istio-security-post-install-account                 53s
istio-sidecar-injector-service-account              53s
istiocoredns-service-account                        53s
prometheus                                          53s
traffic-claim-enforcer-configurator-service-account  53s
traffic-claim-enforcer-service-account              53s

==> v1alpha2/attributemanifest
NAME        AGE
istioproxy  51s
kubernetes  51s

==> v1alpha2/handler
NAME           AGE
kubernetesenv  51s
prometheus     51s

==> v1alpha2/instance
NAME                  AGE
attributes            51s
```

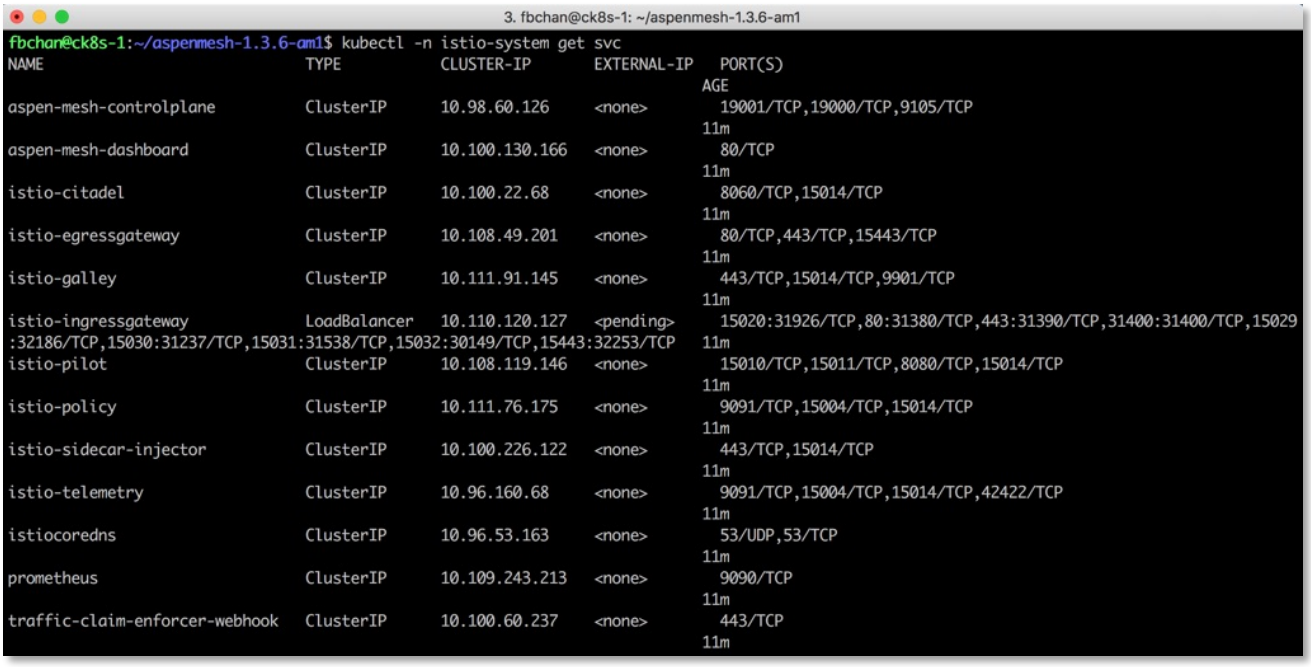
```
requestcount          51s
requestduration       51s
requestsize           51s
responsesize          51s
tcpbytereceived       51s
tcpbytesent           51s
tcpconnectionsclosed  51s
tcpconnectionsopened  51s

==> v1alpha2/rule
NAME                      AGE
kubeattrgenrulerule       51s
promhttp                  51s
promtcp                   51s
promtcpconnectionclosed   51s
promtcpconnectionopen     51s
tcpkubeattrgenrulerule    51s

==> v1alpha3/DestinationRule
NAME                                AGE
istio-multicluster-destinationrule  52s
istio-policy                        52s
istio-telemetry                     52s

==> v1alpha3/EnvoyFilter
NAME                               AGE
istio-multicluster-ingressgateway  52s

==> v1alpha3/Gateway
NAME                               AGE
istio-multicluster-egressgateway   51s
istio-multicluster-ingressgateway  51s

==> v1beta1/ClusterRole
NAME                                     AGE
aspen-mesh-controlplane-view             53s
istio-security-post-install-istio-system  53s
traffic-claim-enforcer                   53s
traffic-claim-enforcer-configurator      53s

==> v1beta1/ClusterRoleBinding
NAME                                                AGE
aspen-mesh-controlplane                             53s
istio-security-post-install-role-binding-istio-system  53s
traffic-claim-enforcer                              53s
traffic-claim-enforcer-configurator                 53s

==> v1beta1/CustomResourceDefinition
NAME                                 AGE
trafficclaims.networking.aspenmesh.io  53s

==> v1beta1/MutatingWebhookConfiguration
NAME                     AGE
istio-sidecar-injector   51s

==> v1beta1/PodDisruptionBudget
NAME                     AGE
istio-egressgateway      53s
istio-galley             53s
istio-ingressgateway     53s
istio-pilot              53s
istio-policy             53s
istio-sidecar-injector   53s
istio-telemetry          53s

==> v1beta1/RoleBinding
```

```
NAME                                  AGE
traffic-claim-enforcer-configurator   53s

==> v2beta1/HorizontalPodAutoscaler
NAME                   AGE
istio-egressgateway    52s
istio-ingressgateway   52s
istio-pilot            52s
istio-policy           52s
istio-telemetry        52s


NOTES:
Thank you for installing Istio.

Your release is named Istio.

To get started running application with Istio, execute the following steps:
1. Label namespace that application object will be deployed to by the following command (take
default namespace as an example)

$ kubectl label namespace default istio-injection=enabled
$ kubectl get namespace -L istio-injection

2. Deploy your applications

$ kubectl apply -f <your-application>.yaml

For more information on running Istio, visit:
https://istio.io/
```

```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ helm list
NAME            REVISION    UPDATED                     STATUS    CHART                   APP VERSION     NAMESPACE
istio           1           Mon Dec 16 20:14:43 2019    DEPLOYED  istio-1.3.6-am1         1.3.6-am1       istio-
system
istio-init      1           Mon Dec 16 20:12:08 2019    DEPLOYED  istio-init-1.3.6-am1    1.3.6-am1       istio-
system
```

Upon successful installation, 3 additional pod will be seen as below for replicated control plane Istio service mesh





Ensure all 24 Istio CRDs and Aspen Mesh CRDs were committed to the Kubernetes apiserver and ensure Aspen Mesh Istio services and deployments are running

```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl get crds | grep
'istio.io\|certmanager.k8s.io\|aspenmesh.io' | wc -l
24

fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl -n istio-system get svc -o custom-
columns=NAME:.metadata.name,CREATED\ AT:.metadata.creationTimestamp
NAME                              CREATED AT
aspen-mesh-controlplane           2019-12-16T09:14:44Z
aspen-mesh-controlplane-cis       2019-12-16T09:27:37Z
aspen-mesh-dashboard              2019-12-16T09:14:45Z
cis-istio-ingressgateway-http2    2019-12-16T09:37:51Z
istio-citadel                     2019-12-16T09:14:45Z
istio-egressgateway               2019-12-16T09:14:45Z
istio-galley                      2019-12-16T09:14:45Z
istio-ingressgateway              2019-12-16T09:14:45Z
istio-pilot                       2019-12-16T09:14:45Z
istio-policy                      2019-12-16T09:14:44Z
istio-sidecar-injector            2019-12-16T09:14:45Z
istio-telemetry                   2019-12-16T09:14:44Z
istiocoredns                      2019-12-16T09:14:45Z
prometheus                        2019-12-16T09:14:44Z
traffic-claim-enforcer-webhook    2019-12-16T09:14:44Z

fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl -n istio-system get deploy -o custom-
columns=NAME:.metadata.name,CREATED\ AT:.metadata.creationTimestamp
NAME                        CREATED AT
aspen-mesh-controlplane     2019-12-16T09:14:45Z
aspen-mesh-dashboard        2019-12-16T09:14:45Z
istio-citadel               2019-12-16T09:14:45Z
istio-egressgateway         2019-12-16T09:14:45Z
istio-galley                2019-12-16T09:14:45Z
istio-ingressgateway        2019-12-16T09:14:45Z
istio-pilot                 2019-12-16T09:14:45Z
istio-policy                2019-12-16T09:14:45Z
istio-sidecar-injector      2019-12-16T09:14:45Z
istio-telemetry             2019-12-16T09:14:45Z
istiocoredns                2019-12-16T09:14:45Z
```

```
prometheus                2019-12-16T09:14:45Z
traffic-claim-enforcer    2019-12-16T09:14:45Z
```

Verify Aspen Mesh multi-cluster gateways are ready

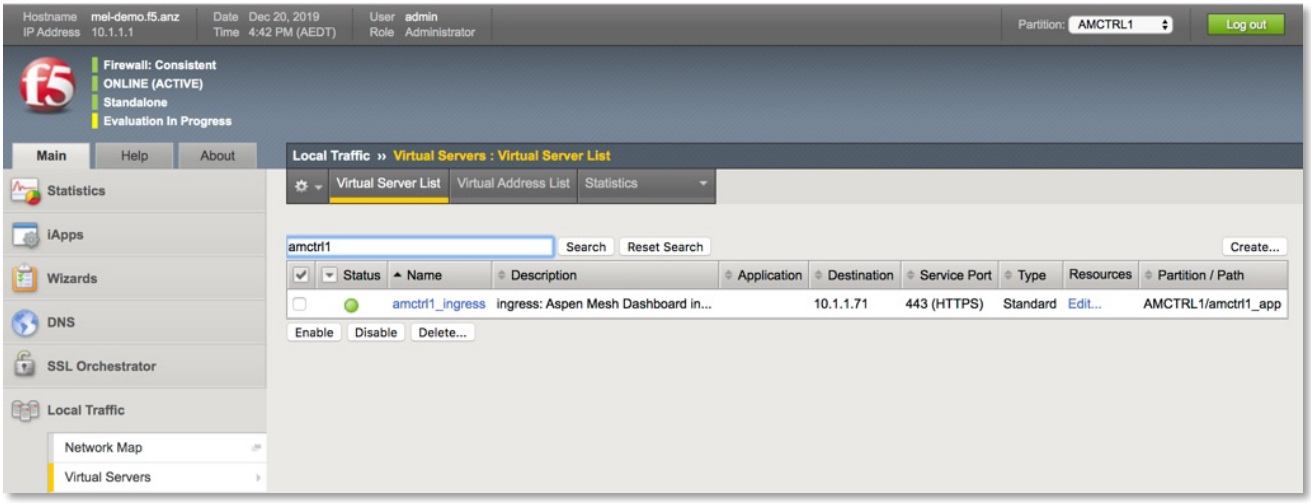
```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl -n istio-system get gateways
NAME                                AGE
istio-multicluster-egressgateway    1h
istio-multicluster-ingressgateway   1h
```

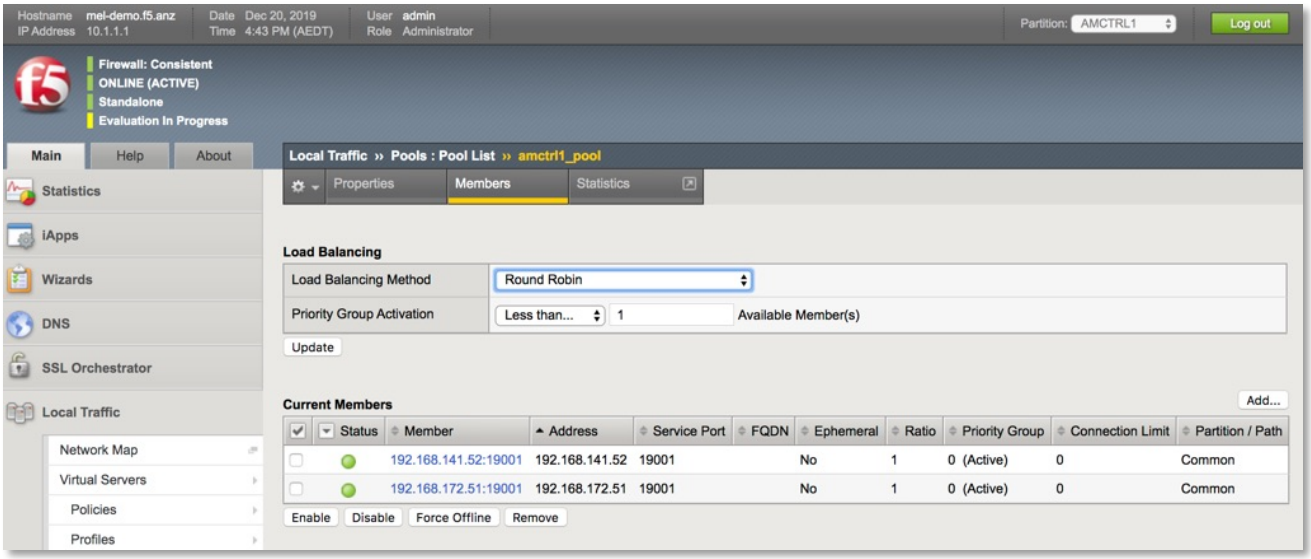
Deploy CIS Aspen Mesh dashboard configuration manifest

```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl apply -f cis-aspen-mesh-controlplan.yml
service/aspen-mesh-controlplane-cis created

fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl apply -f cis-aspen-mesh-controlplan-AMCTL1-
configmap.yml
configmap/f5-as3-amctrl1 created
```

This will allow Aspen Mesh dashboard to be access directly from browser via BIG-IP. Alternatively, accessing Aspen Mesh dashboard via NodePort or LoadBalancer type.

CIS via AS3 declarative API orchestrate creations of Aspen Mesh Dashboard virtual server on BIG-IP

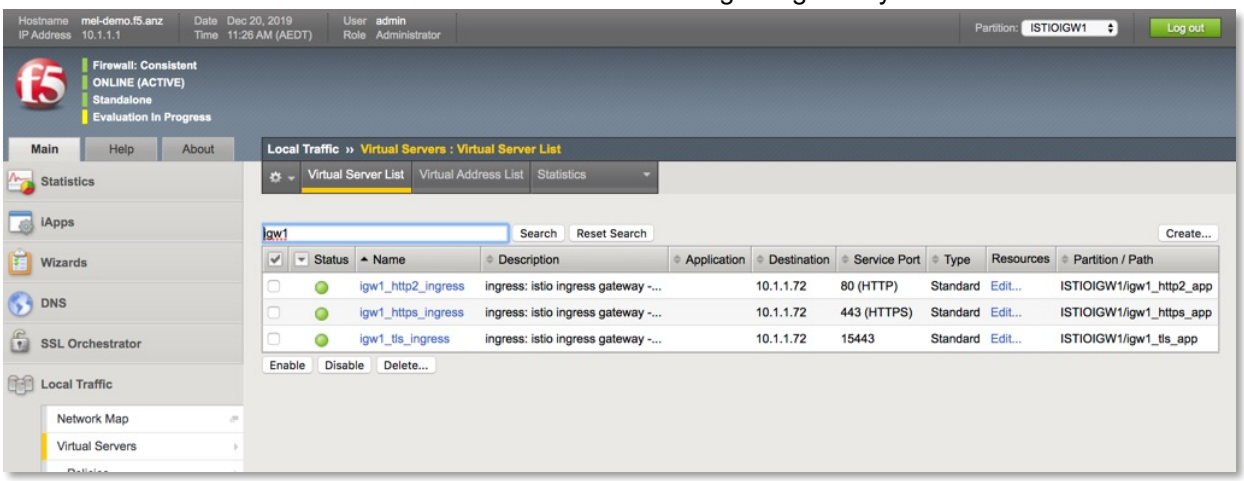


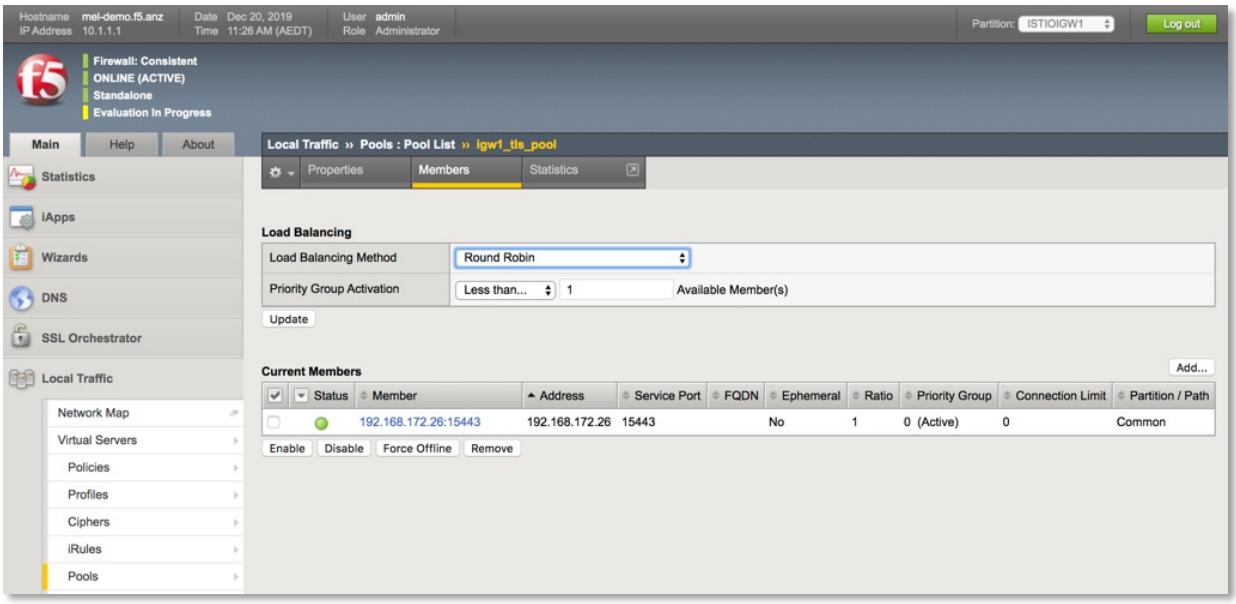


Deploy CIS Istio ingressgateway configuration manifest

```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl apply -f cis-istio-ingress-gw1-svc.yml
service/cis-istio-ingressgateway-http2 created
service/cis-istio-ingressgateway-tls created

fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl apply -f cis-istio-ingress-gw1-configmap.yml
configmap/f5-as3-istioigw1 created
```

CIS via AS3 declarative API orchestrate creations of ISTIO ingress gateway virtual server on BIG-IP





Accessing Aspen Mesh dashboard via BIG-IP VS

Note: No applications setup yet. Hence, no observability and analytic data.

Deployment of Sample Apps

Create a namespace to host service mesh application and enable sidecar auto injection via namespace labelling.

```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl create namespace istio-apps
namespace/istio-apps created

fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl label --overwrite namespace istio-apps istio-
injection=enabled
namespace/istio-apps labeled

fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl get namespace -L istio-injection
NAME              STATUS   AGE     ISTIO-INJECTION
default           Active   22d
istio-apps        Active   17s     enabled
istio-system      Active   9m25s
kube-node-lease   Active   22d
kube-public       Active   22d
kube-system       Active   22d
monitoring        Active   9d
```

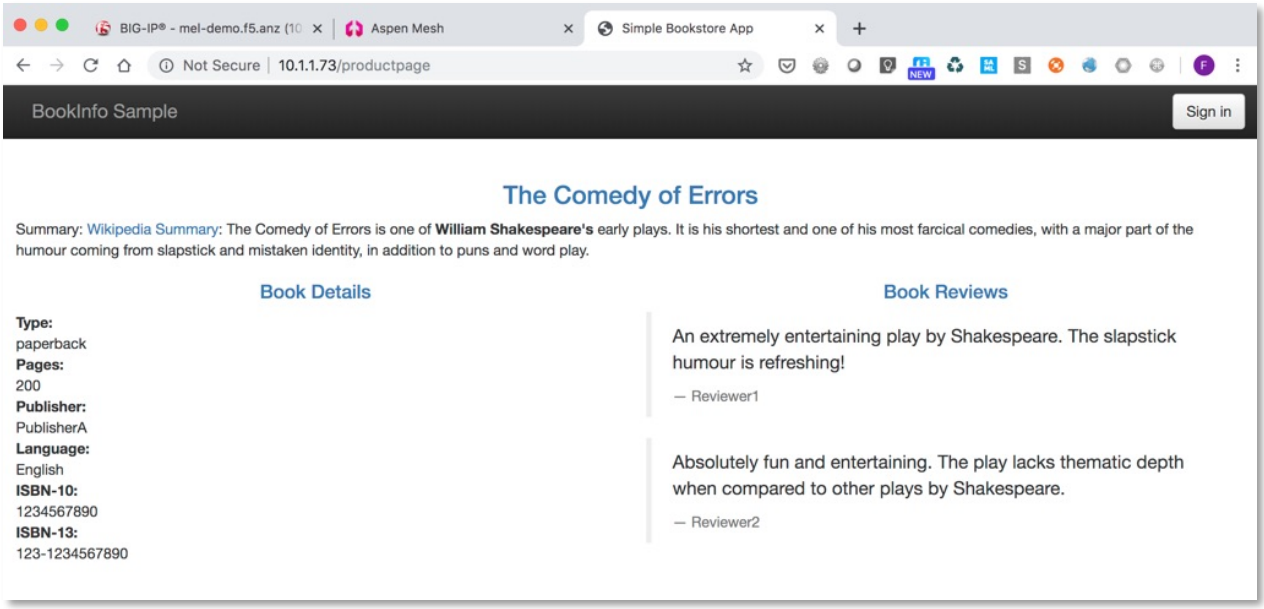
```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl -n istio-apps apply -f
./samples/bookinfo/platform/kube/bookinfo.yaml
service/details created
serviceaccount/bookinfo-details created
deployment.apps/details-v1 created
service/ratings created
serviceaccount/bookinfo-ratings created
deployment.apps/ratings-v1 created
service/reviews created
serviceaccount/bookinfo-reviews created
deployment.apps/reviews-v1 created
deployment.apps/reviews-v2 created
deployment.apps/reviews-v3 created
service/productpage created
serviceaccount/bookinfo-productpage created
deployment.apps/productpage-v1 created

fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl -n istio-apps get pod
NAME                              READY   STATUS    RESTARTS   AGE
details-v1-78d78fbddf-s8mdn       2/2     Running   0          44s
productpage-v1-596598f447-qp4fj   2/2     Running   0          43s
ratings-v1-6c9dbf6b45-4q6q8       2/2     Running   0          43s
reviews-v1-7bb8ffd9b6-97z9q       2/2     Running   0          44s
reviews-v2-d7d75fff8-572s5        2/2     Running   0          44s
```

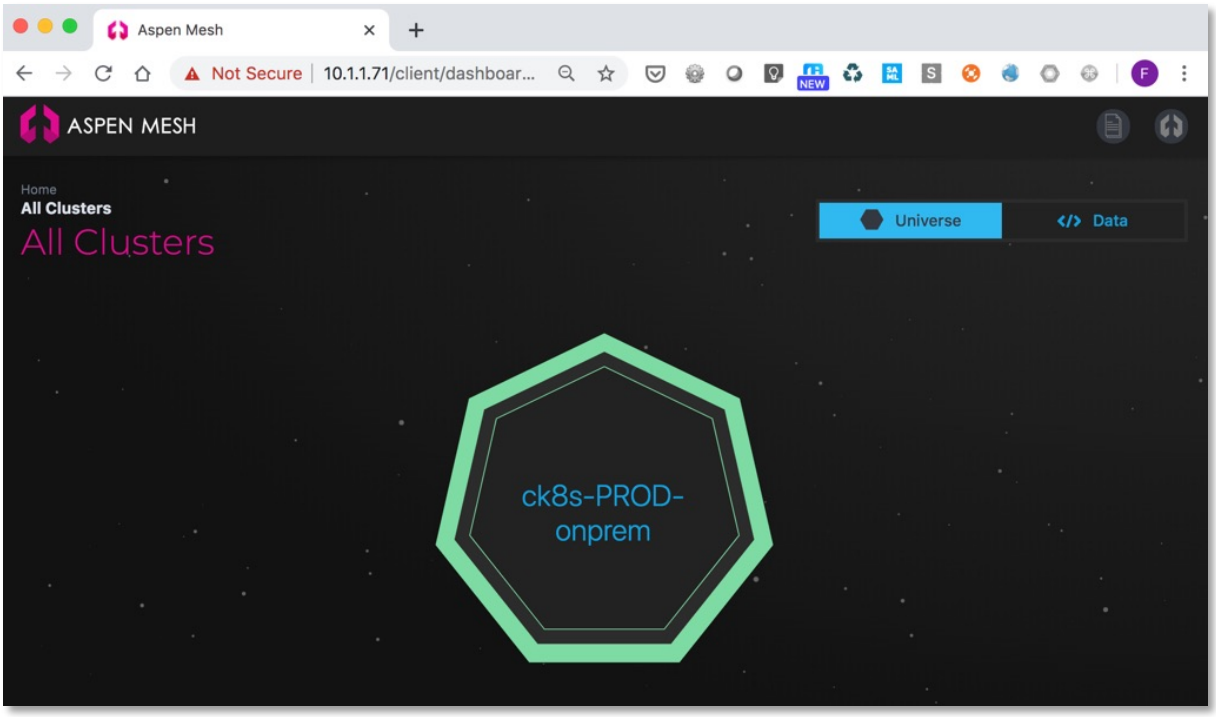
```
reviews-v3-68964bc4c8-sr5lw       2/2     Running   0          44s

fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl -n istio-apps apply -f
./samples/bookinfo/networking/bookinfo-gateway.yaml
gateway.networking.istio.io/bookinfo-gateway created
virtualservice.networking.istio.io/bookinfo created

fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl -n istio-apps get gateway
NAME               AGE
bookinfo-gateway   7m56s

fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl -n istio-apps describe gateway bookinfo-gateway
Name:         bookinfo-gateway
Namespace:    istio-apps
Labels:       <none>
Annotations:  kubectl.kubernetes.io/last-applied-configuration:

{"apiVersion":"networking.istio.io/v1alpha3","kind":"Gateway","metadata":{"annotations":{},"name
":"bookinfo-gateway","namespace":"istio-ap...
API Version:  networking.istio.io/v1alpha3
Kind:         Gateway
Metadata:
  Creation Timestamp:  2019-12-16T09:36:02Z
  Generation:          1
  Resource Version:    2729443
  Self Link:           /apis/networking.istio.io/v1alpha3/namespaces/istio-
apps/gateways/bookinfo-gateway
  UID:                 df8ae33b-59c7-4d90-a940-4ffa85fa50ac
Spec:
  Selector:
    Istio:  ingressgateway
  Servers:
    Hosts:
      *
    Port:
      Name:      http
      Number:    80
      Protocol:  HTTP
Events:          <none>

fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl -n istio-apps get virtualservice
NAME       GATEWAYS             HOSTS   AGE
bookinfo   [bookinfo-gateway]   [*]     8m19s

fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl -n istio-apps describe virtualservice bookinfo
Name:         bookinfo
Namespace:    istio-apps
Labels:       <none>
Annotations:  kubectl.kubernetes.io/last-applied-configuration:

{"apiVersion":"networking.istio.io/v1alpha3","kind":"VirtualService","metadata":{"annotations":{
},"name":"bookinfo","namespace":"istio-app...
API Version:  networking.istio.io/v1alpha3
Kind:         VirtualService
Metadata:
  Creation Timestamp:  2019-12-16T09:36:02Z
  Generation:          1
  Resource Version:    2729444
  Self Link:           /apis/networking.istio.io/v1alpha3/namespaces/istio-
apps/virtualservices/bookinfo
  UID:                 4ef36112-4f2e-429d-9832-acc567cc5e0d
Spec:
  Gateways:
    bookinfo-gateway
  Hosts:
    *
```

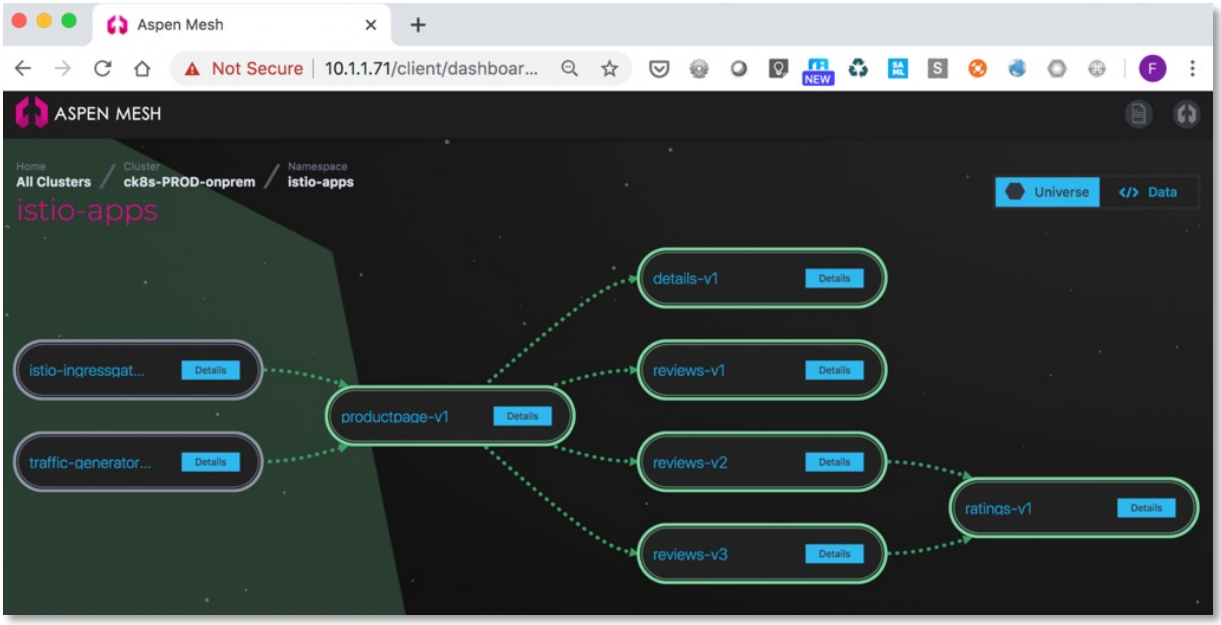
```
  Http:
    Match:
      Uri:
        Exact:  /productpage
      Uri:
        Prefix:  /static
      Uri:
        Exact:  /login
      Uri:
        Exact:  /logout
      Uri:
        Prefix:  /api/v1/products
    Route:
      Destination:
        Host:  productpage
        Port:
          Number:  9080
Events:            <none>
```

Deploy bookinfo traffic generator

```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl apply -f ./samples/aspenmesh/bookinfo-traffic-generator.yaml
configmap/traffic-generator-productpage created
service/traffic-generator-productpage created
deployment.apps/traffic-generator-productpage created
```







Repeat above for other 2 cluster (k8s-DMZ-onprem and gke-DEV-Edge1)

Installation instruction to enable multi-cluster Istio communication

Setup DNS (kube-system Kube-DNS/CoreDNS) for .global – on-prem Kubernetes Istio cluster

```
fbchan@ck8s-1:~$ kubectl -n istio-system get pod | grep coredns
istiocoredns-576dd6459c-2kv6p              2/2     Running     0          12h
fbchan@ck8s-1:~$ kubectl -n istio-system logs istiocoredns-576dd6459c-2kv6p -c coredns
2019/12/16 09:14:53 [INFO] CoreDNS-1.1.2
2019/12/16 09:14:53 [INFO] linux/amd64, go1.10.1, 582f91f3
2019/12/16 09:14:53 [INFO] plugin/reload: Running configuration MD5 =
1208c932ca1e3a314511d689c594a25a
.:53
CoreDNS-1.1.2
linux/amd64, go1.10.1, 582f91f3
```

```
BEFORE CHANGES
fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl -n kube-system describe configmap coredns
Name:         coredns
Namespace:    kube-system
Labels:       <none>
Annotations:  <none>

Data
====
Corefile:
----
.:53 {
    errors
    health
    ready
    kubernetes cluster.local in-addr.arpa ip6.arpa {
       pods insecure
       fallthrough in-addr.arpa ip6.arpa
       ttl 30
    }
    prometheus :9153
    forward . /etc/resolv.conf
    cache 30
    loop
    reload
    loadbalance
}

Events:  <none>

Cluster1
fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ vi ck8s-prod-onprem-coredns.yml
fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl apply -f ck8s-prod-onprem-coredns.yml
Warning: kubectl apply should be used on resource created by either kubectl create --save-config
or kubectl apply
configmap/coredns configured

Note:
Repeat similar with other Cluster
```

Configure kube-system ‘coredns’ to forward DNS resolution for ‘global’ to istiocoredns.

```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am1/multi-cluster-apps$ kubectl get svc -n istio-system
istiocoredns -o jsonpath={.spec.clusterIP}
10.107.207.164

fbchan@ck8s-1:~/aspenmesh-1.3.6-am1/multi-cluster-apps$ cat ck8s-prod-onprem-coredns.yml
apiVersion: v1
kind: ConfigMap
metadata:
  name: coredns
  namespace: kube-system
data:
  Corefile: |
    .:53 {
        errors
        health
        ready
        kubernetes cluster.local in-addr.arpa ip6.arpa {
           pods insecure
           fallthrough in-addr.arpa ip6.arpa
           ttl 30
        }
        prometheus :9153
```

```
        forward . /etc/resolv.conf
        cache 30
        loop
        reload
        loadbalance
    }
    global:53 {
        log
        errors
        cache 30
        forward . 10.107.207.164
    }
```

Note:

10.107.207.164 is istiocoredns clusterIP. Can be obtains via '`kubectl get svc -n istio-system istiocoredns -o jsonpath={.spec.clusterIP}`'

Note:

For GKE (e.g. gke-dev-edge1) as it is using KubeDNS instead of CoreDNS. Follow below configuration

```
fbchan@logos:~/k8s-clusterX/gke-1/aspenmesh-1.3.6-am1$ kubectl get svc -n istio-system
istiocoredns -o jsonpath={.spec.clusterIP}
10.12.16.94


fbchan@logos:~/k8s-clusterX/gke-1/aspenmesh-1.3.6-am1$ cat gke-kube-dns-multi-cluster.yml
apiVersion: v1
kind: ConfigMap
metadata:
  labels:
    addonmanager.kubernetes.io/mode: EnsureExists
  name: kube-dns
  namespace: kube-system
metadata:
  name: kube-dns
  namespace: kube-system
data:
  stubDomains: |
    {"global": ["10.12.16.94"]}


fbchan@logos:~/k8s-clusterX/gke-1/aspenmesh-1.3.6-am1$ kubectl apply -f gke-kube-dns-multi-
cluster.yml
configmap/kube-dns changed
```

Validate configuration with simple application (sleep on local cluster and httpbin and nettools on remote cluster).

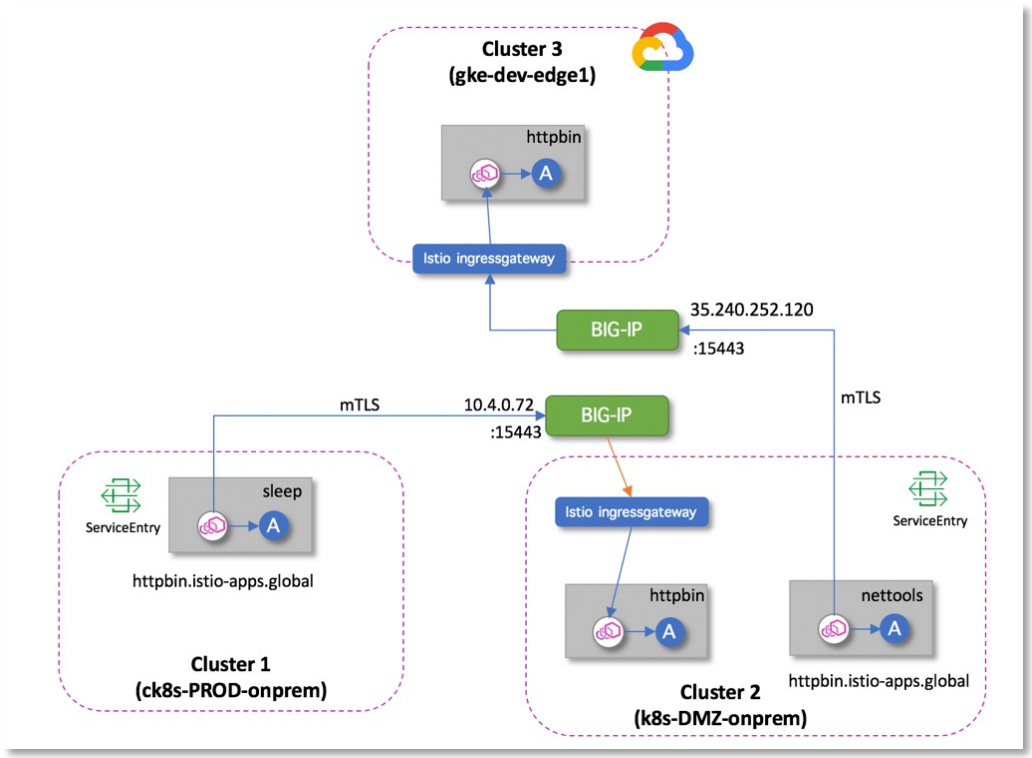
Note: You need to have both/all Aspen Mesh multi-cluster installed to validate configuration.

```
Cluster 1 (cK8s-PROD-onprem)

fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl apply -n istio-apps -f samples/sleep/sleep.yaml
serviceaccount/sleep created
service/sleep created
deployment.apps/sleep created

fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl -n istio-apps get pod | grep sleep
sleep-cd4674f5b-wprz4             2/2     Running   0          2m39s


Cluster 2 (k8s-DMZ-onprem)
fbchan@k8s-1:~/aspenmesh-1.3.6-am1$ kubectl apply -n istio-apps -f samples/httpbin/httpbin.yaml
service/httpbin created
deployment.apps/httpbin created
```

```
fbchan@k8s-1:~/aspenmesh-1.3.6-am1$ kubectl -n istio-apps get pod | grep httpbin
httpbin-5446f4d9b4-67frl                          2/2     Running   0          51s

Cluster 2 (k8s-DMZ-onprem)
fbchan@k8s-1:~/aspenmesh-1.3.6-am1/multi-cluster-apps$ kubectl -n istio-apps apply -f nettools-
deploy.yml
deployment.extensions/nettools-deploy created


Cluster 3 (gke-dev-edge1) – Google Cloud
fbchan@logos:~/k8s-clusterX/gke-1/aspenmesh-1.3.6-am1$ kubectl apply -n istio-apps -f
samples/httpbin/httpbin.yaml
service/httpbin created
deployment.apps/httpbin created

fbchan@logos:~/k8s-clusterX/gke-1/aspenmesh-1.3.6-am1$ kubectl -n istio-apps get pod | grep
httpbin
NAME                                          READY     STATUS    RESTARTS   AGE
details-v1-74f858558f-7tchk                   2/2       Running   0          12h
```

Create ServiceEntry on Cluster1 for httpbin.isito-apps.global to point to Cluster 2 BIG-IP virtual server that pointing to Cluster 2 istio ingressgateway

Create ServiceEntry on Cluster2 for httpbin.istio-apps.global to point to Cluster 3 BIG-IP virtual server (Google Cloud) that pointing to Cluster 3 istio ingressgateway




```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am1/multi-cluster-apps$ cat httpbin-istio-apps-ServiceEntry.yml
apiVersion: networking.istio.io/v1alpha3
kind: ServiceEntry
metadata:
  name: httpbin-istio-apps
spec:
  hosts:
  # must be of form name.namespace.global
```

```
  - httpbin.istio-apps.global
  # Treat remote cluster services as part of the service mesh
  # as all clusters in the service mesh share the same root of trust.
  location: MESH_INTERNAL
  ports:
  - name: http1
    number: 8000
    protocol: http
  resolution: DNS
  addresses:
  # the IP address to which httpbin.istio-apps.global will resolve to
  # must be unique for each remote service, within a given cluster.
  # This address need not be routable. Traffic for this IP will be captured
  # by the sidecar and routed appropriately.
  - 240.0.0.2
  endpoints:
  # This is the routable address of the ingress gateway in cluster2 that
  # sits in front of sleep.foo service. Traffic from the sidecar will be
  # routed to this address.
  - address: 10.4.0.72
    ports:
     http1: 15443 # Do not change this port value
```

All traffic in Cluster 1 for httpbin.istio-apps.global on any port will be routed to endpoint 10.4.0.72:15443 over a mutual TLS connection.

The gateway for port 15443 is a special SNI-aware Envoy preconfigured and installed when you deployed the Istio control plane in the cluster. Traffic entering port 15443 will be load balanced among pods of the appropriate internal service of the target cluster (in this case, httpbin.istio-apps in cluster2).

Validate Cluster 1 to Cluster 2

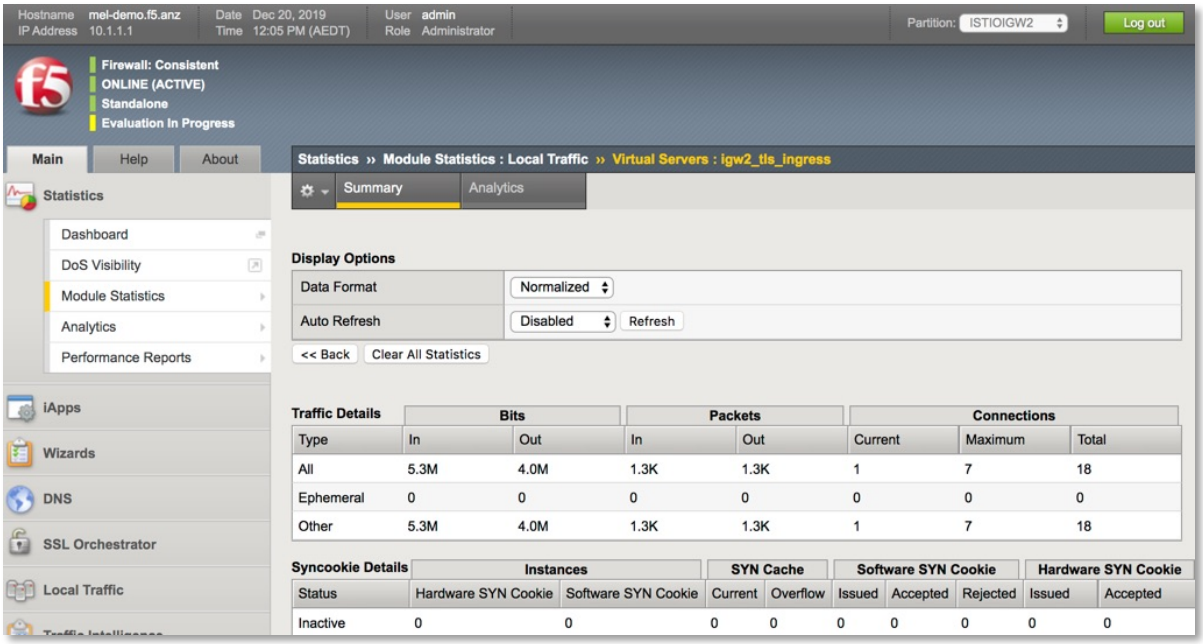
```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am1/multi-cluster-apps$ kubectl -n istio-apps exec -it sleep-
cd4674f5b-wprz4 -c sleep sh
/ #

Ensure httpbin.istio-apps.global resolved to an IP defined in ServiceEntry

/ # ping httpbin.istio-apps.global
PING httpbin.istio-apps.global (240.0.0.2): 56 data bytes
^C
--- httpbin.istio-apps.global ping statistics ---
2 packets transmitted, 0 packets received, 100% packet loss


Run curl (sleep pod on Cluster 1) to access httpbin pod on remote Cluster2
/ # curl http://httpbin.istio-apps.global:8000/headers
{
  "headers": {
    "Accept": "*/*",
    "Content-Length": "0",
    "Host": "httpbin.istio-apps.global:8000",
    "User-Agent": "curl/7.64.0",
    "X-B3-Parentspanid": "8bb73e6b57ee86d1",
    "X-B3-Sampled": "0",
    "X-B3-Spanid": "bea772076c2b6a1f",
    "X-B3-Traceid": "7d7520a9495800ef8bb73e6b57ee86d1",
    "X-Forwarded-Client-Cert": "By=spiffe://cluster.local/ns/istio-
apps/sa/default;Hash=293e9ffd157d5ed4bed8cfe97d05a5fa504928740a1119607e1e3cc62773b40f;Subject=\"
\";URI=spiffe://cluster.local/ns/istio-apps/sa/sleep"
  }
}
/ #
```

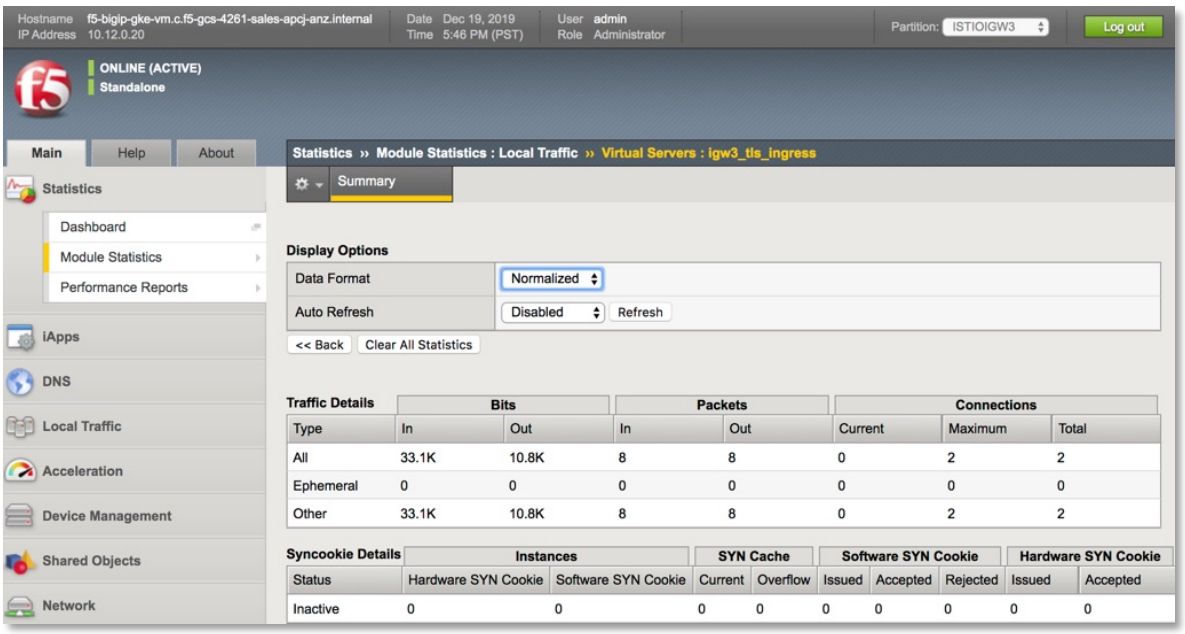
Validate Cluster 2 to Cluster 3

```
fbchan@k8s-1:~/aspenmesh-1.3.6-am1/multi-cluster-apps$ cat httpbin-istio-apps-SE-GKE.yml
apiVersion: networking.istio.io/v1alpha3
kind: ServiceEntry
metadata:
  name: httpbin-istio-apps
spec:
  hosts:
  # must be of form name.namespace.global
  - httpbin.istio-apps.global
  # Treat remote cluster services as part of the service mesh
  # as all clusters in the service mesh share the same root of trust.
  location: MESH_INTERNAL
  ports:
  - name: http1
    number: 8000
    protocol: http
  resolution: DNS
  addresses:
  # the IP address to which httpbin.istio-apps.global will resolve to
  # must be unique for each remote service, within a given cluster.
  # This address need not be routable. Traffic for this IP will be captured
  # by the sidecar and routed appropriately.
  - 240.0.0.6
  endpoints:
  # This is the routable address of the ingress gateway in cluster2 that
  # sits in front of sleep.foo service. Traffic from the sidecar will be
  # routed to this address.
  - address: 35.240.252.120
    ports:
     http1: 15443 # Do not change this port value
```

```
Run curl (nettools pod on Cluster 2) to access httpbin pod on remote Cluster 3 in Google GKE
fbchan@k8s-1:~/aspenmesh-1.3.6-am1/multi-cluster-apps$ kubectl -n istio-apps exec -it nettools-
deploy-79577bdb5b-l5g7p -c nettools bash

bash-5.0# ping httpbin.istio-apps.global
PING httpbin.istio-apps.global (240.0.0.6) 56(84) bytes of data.

--- httpbin.istio-apps.global ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 9ms


bash-5.0# curl http://httpbin.istio-apps.global:8000/headers
{
  "headers": {
    "Accept": "*/*",
    "Content-Length": "0",
    "Host": "httpbin.istio-apps.global:8000",
    "User-Agent": "curl/7.65.1",
    "X-B3-Parentspanid": "12d2315ddcdaada1",
    "X-B3-Sampled": "0",
    "X-B3-Spanid": "5f218ecf91f16ec2",
    "X-B3-Traceid": "9786114daefb791812d2315ddcdaada1",
    "X-Forwarded-Client-Cert": "By=spiffe://cluster.local/ns/istio-
apps/sa/default;Hash=ac2f6a02903ab8403df29737656a684ad89ba8c518509eeb75727a6a8eff7082;Subject=\"
\";URI=spiffe://cluster.local/ns/istio-apps/sa/default"
  }
}
```

You can also validate this to ensure traffic hitting TLS VS on BIG-IP via TLS statistic.


Hostname mel-demo.f5.anz
IP Address 10.1.1.1
Date Dec 20, 2019
Time 12:05 PM (AEDT)
User admin
Role Administrator
Partition: ISTIOIGW2
Log out
Firewall: Consistent
ONLINE (ACTIVE)
Standalone
Evaluation In Progress
Main
Help
About
Statistics » Module Statistics : Local Traffic » Virtual Servers : igw2_tls_ingress
Summary
Analytics
Statistics
Dashboard
DoS Visibility
Module Statistics
Analytics
Performance Reports
iApps
Wizards
DNS
SSL Orchestrator
Local Traffic
Display Options
Data Format
Normalized
Auto Refresh
Disabled
Refresh
<< Back
Clear All Statistics
Traffic Details
Bits
Packets
Connections
Type
In
Out
In
Out
Current
Maximum
Total
All 5.3M 4.0M 1.3K 1.3K 1 7 18
Ephemeral 0 0 0 0 0 0 0
Other 5.3M 4.0M 1.3K 1.3K 1 7 18
Syncookie Details
Instances
SYN Cache
Software SYN Cookie
Hardware SYN Cookie
Status
Hardware SYN Cookie
Software SYN Cookie
Current
Overflow
Issued
Accepted
Rejected
Issued
Accepted
Inactive 0 0 0 0 0 0 0 0 0


Hostname f5-bigip-gke-vm.c.f5-gcs-4261-sales-apcj-anz.internal
IP Address 10.12.0.20
Date Dec 19, 2019
Time 5:46 PM (PST)
User admin
Role Administrator
Partition: ISTIOIGW3
Log out
ONLINE (ACTIVE)
Standalone
Main
Help
About
Statistics » Module Statistics : Local Traffic » Virtual Servers : igw3_tls_ingress
Summary
Statistics
Dashboard
Module Statistics
Performance Reports
iApps
DNS
Local Traffic
Acceleration
Device Management
Shared Objects
Network
Display Options
Data Format
Normalized
Auto Refresh
Disabled
Refresh
<< Back
Clear All Statistics
Traffic Details
Bits
Packets
Connections
Type
In
Out
In
Out
Current
Maximum
Total
All 33.1K 10.8K 8 8 0 2 2
Ephemeral 0 0 0 0 0 0 0
Other 33.1K 10.8K 8 8 0 2 2
Syncookie Details
Instances
SYN Cache
Software SYN Cookie
Hardware SYN Cookie
Status
Hardware SYN Cookie
Software SYN Cookie
Current
Overflow
Issued
Accepted
Rejected
Issued
Accepted
Inactive 0 0 0 0 0 0 0 0 0

6.2 Aspen Mesh Upgrade Guide

Download “aspenmesh-1.3.6-am3-linux.tar.gz” and extract to local folder.

To enable Multi-Cluster Aspen Mesh Dashboard, ensure each Aspen Mesh Dashboard is accessible from network or client browser. Create and apply the following multi-cluster dashboard configmap.

```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am3/multi-cluster-EA$ cat aspen-mesh-controlplane.yml
apiVersion: v1
kind: ConfigMap
metadata:
 # the name of the ConfigMap must be named aspen-mesh-controlplane
 name: aspen-mesh-controlplane
 namespace: istio-system
data:
 # the data key must be named multicluster.yaml
 multicluster.yaml: |-
   peerClusters:
   - id: k8s-DMZ-onprem
     dashboardUrlPrefix: https://am2.foobz.com.au/
   - id: gke-dev-edge1
     dashboardUrlPrefix: https://am3.foobz.com.au/


fbchan@ck8s-1:~/aspenmesh-1.3.6-am3/multi-cluster-EA$ kubectl -n istio-system apply -f aspen-
mesh-controlplane.yml
configmap/aspen-mesh-controlplane created

fbchan@ck8s-1:~/aspenmesh-1.3.6-am3/multi-cluster-EA$ ping am2.foobz.com.au
PING am2.foobz.com.au (10.4.0.71) 56(84) bytes of data.
64 bytes from 10.4.0.71 (10.4.0.71): icmp_seq=1 ttl=255 time=0.330 ms

fbchan@ck8s-1:~/aspenmesh-1.3.6-am3/multi-cluster-EA$ ping am3.foobz.com.au
PING am3.foobz.com.au (35.240.252.120) 56(84) bytes of data.
64 bytes from 120.252.240.35.bc.googleusercontent.com (35.240.252.120): icmp_seq=1 ttl=238
time=108 ms
```

Where https://am1.foobz.com.au and https://am2.foobz.com.au accessible from browser. Alternatively, if you don’t have FQDN or BIG-IP VS configured, you can specify http://<IP address>:19001 for the peerCluster.

The peerClusters list must contain both the id and dashboardUrlPrefix for each peer cluster (defined in multicluster.yaml configmap). The list should exclude the current cluster. The example above is for a 3-cluster configuration (there are 2 peer clusters, and the cluster we are configuring).

Add multiclusterEnabled to value-aspenmesh.yaml file. This can be found in the release under install/kubernetes/helm/istio/values-aspenmesh.yaml. In the aspen-mesh-controlplane section, add multiclusterEnabled: true

```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am3$ cat install/kubernetes/helm/istio/values-aspenmesh.yaml
global:
  controlPlaneSecurityEnabled: true
  mtls:
    enabled: true

  imagePullSecrets:
  - aspen-mesh-pull-secret

  proxy:
    accessLogFile: "/dev/stdout"
```

```
  outboundTrafficPolicy:
    mode: REGISTRY_ONLY

  tracer:
    zipkin:
      # address is the Host:Port for reporting trace data in zipkin format. If
      # not specified, will default to zipkin service (port 9411) in the same
      # namespace as the other istio components.
      address: ""

  ### Aspen Mesh Multi-cluster
  aspen-mesh-controlplane:
   multiclusterEnabled: true

  ##### MULTICLUSTER #######
  podDNSSearchNamespaces:
  - global
  - "{{ valueOrDefault .DeploymentMeta.Namespace \"default\" }}.global"

  multiCluster:
    enabled: true
  ########################

pilot:
  # Aspen Mesh has decided to disable this experimental feature until
  # both outbound and inbound protocol sniffing is merged and is
  # stable
  enableProtocolSniffingForOutbound: false

aspen-mesh-controlplane:
  enabled: true
  image: quay.io/aspenmesh/releases:controlplane-1.3.6-am3

  imagePullSecrets:
  - aspenmesh-kubernetes-pull-secret

  replicaCount: 2

  clusterId: ck8s-PROD-onprem

  userAuth:
    type: none
    # jwt:
      # jwks must provide a valid JWKs endpoint.
      # jwks:
      # redirectUrl must provide an unauthenticated redirect URL.
      # redirectUrl:
      # claims can be added here to require claims be present in the JWT
      # (e.g.: `claims: aud=aspenmesh:io,role=k8s:admin`).

  # prometheusUrl must provide a valid Prometheus URL.
  prometheusUrl: http://prometheus.monitoring.svc.cluster.local:9090

  resources:
    requests:
      memory: "128Mi"
      cpu: "100m"
    limits:
      memory: "128Mi"
      cpu: "100m"

aspen-mesh-dashboard:
  enabled: true
  image: quay.io/aspenmesh/releases:dashboard-1.3.6-am3
  replicaCount: 2
```

```
traffic-claim-enforcer:
  enabled: true
  image: quay.io/aspenmesh/releases:traffic-claim-enforcer-1.3.6-am3
  resources: {}

istio_cni:
  # To enable CNI-based installations, set this to true and follow additional
  # CNI installation instructions to install the istio-cni Helm chart
  enabled: false

prometheus:
  enabled: true

# MULTICLUSTER #######

# Multicluster with gateways requires a root CA
# Cluster local CAs are bootstrapped with the root CA.
security:
  selfSigned: false

# Provides dns resolution for service entries of form
# name.namespace.global
istiocoredns:
  enabled: true

gateways:
  istio-egressgateway:
    enabled: true
    env:
      # Needed to route traffic via egress gateway if desired.
      ISTIO_META_REQUESTED_NETWORK_VIEW: "external"
```

```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am3$ helm upgrade --install --force istio-init
install/kubernetes/helm/istio-init --namespace=istio-system
Release "istio-init" has been upgraded.
LAST DEPLOYED: Fri Dec 20 15:59:32 2019
NAMESPACE: istio-system
STATUS: DEPLOYED

RESOURCES:
==> v1/ClusterRole
NAME                     AGE
istio-init-istio-system  2d21h

==> v1/ClusterRoleBinding
NAME                                        AGE
istio-init-admin-role-binding-istio-system  2d21h

==> v1/ConfigMap
NAME          AGE
istio-crd-10  2d21h
istio-crd-11  2d21h
istio-crd-12  2d21h

==> v1/Job
NAME                      AGE
istio-init-crd-10-1.3.6   2d21h
istio-init-crd-11-1.3.6   2d21h
istio-init-crd-12-1.3.6   2d21h

==> v1/Pod(related)
NAME                            AGE
istio-init-crd-10-1.3.6-sjcnd   2d21h
```

```
istio-init-crd-11-1.3.6-dktwx   2d21h
istio-init-crd-12-1.3.6-zslhf   2d21h

==> v1/ServiceAccount
NAME                         AGE
istio-init-service-account   2d21h
```

```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am3$ helm upgrade istio install/kubernetes/helm/istio --
namespace istio --values install/kubernetes/helm/istio/values-aspenmesh.yaml
Release "istio" has been upgraded.
LAST DEPLOYED: Fri Dec 20 15:57:09 2019
NAMESPACE: istio-system
STATUS: DEPLOYED

RESOURCES:
==> v1/ClusterRole
NAME                                AGE
istio-citadel-istio-system          2d21h
istio-galley-istio-system           2d21h
istio-mixer-istio-system            2d21h
istio-pilot-istio-system            2d21h
istio-reader                        2d21h
istio-sidecar-injector-istio-system  2d21h
istiocoredns                        2d21h
prometheus-istio-system             2d21h

==> v1/ClusterRoleBinding
NAME                                                AGE
istio-citadel-istio-system                          2d21h
istio-galley-admin-role-binding-istio-system        2d21h
istio-istiocoredns-role-binding-istio-system        2d21h
istio-mixer-admin-role-binding-istio-system         2d21h
istio-multi                                         2d21h
istio-pilot-istio-system                            2d21h
istio-sidecar-injector-admin-role-binding-istio-system  2d21h
prometheus-istio-system                             2d21h

==> v1/ConfigMap
NAME                                AGE
coredns                             2d21h
istio                               2d21h
istio-galley-configuration          2d21h
istio-security-custom-resources     2d21h
istio-sidecar-injector              2d21h
prometheus                          2d21h
traffic-claim-enforcer-configurator  2d21h

==> v1/Deployment
NAME                     AGE
aspen-mesh-controlplane  2d21h
aspen-mesh-dashboard     2d21h
istio-citadel            2d21h
istio-egressgateway      2d21h
istio-galley             2d21h
istio-ingressgateway     2d21h
istio-pilot              2d21h
istio-policy             2d21h
istio-sidecar-injector   2d21h
istio-telemetry          2d21h
istiocoredns             2d21h
prometheus               2d21h
```

```
traffic-claim-enforcer   2d21h

==> v1/Pod(related)
NAME                                    AGE
aspen-mesh-controlplane-774cc47966-c7r5g  26s
aspen-mesh-controlplane-77c7b69c56-tsd74  58m
aspen-mesh-controlplane-77c7b69c56-xjpn9  58m
aspen-mesh-dashboard-569c47c65c-lq79q     26s
aspen-mesh-dashboard-569c47c65c-rdbvk     12s
aspen-mesh-dashboard-64d78d7999-phvmr     83m
istio-citadel-8445bfb9c8-j8wkl            2d21h
istio-egressgateway-596867b5f4-txwbj      2d21h
istio-galley-78f78c94db-xtnjw             2d21h
istio-ingressgateway-688d4c5c7-p2nd5      2d21h
istio-pilot-67545c7d67-k9plh              2d21h
istio-policy-6dfb4bc9f9-wrvjv             2d21h
istio-sidecar-injector-7d8d57d7c4-8pmlw   2d21h
istio-telemetry-799dcb4b6b-x6k7c          2d21h
istiocoredns-576dd6459c-lv9kk             2d21h
prometheus-6f74d6f76d-gbdgz               2d21h
traffic-claim-enforcer-7699cb5858-mldmw   26s

==> v1/Role
NAME                                AGE
istio-ingressgateway-sds            2d21h
traffic-claim-enforcer-configurator  2d21h

==> v1/RoleBinding
NAME                      AGE
istio-ingressgateway-sds  2d21h

==> v1/Secret
NAME                    AGE
aspen-mesh-pull-secret  2d21h

==> v1/Service
NAME                           AGE
aspen-mesh-controlplane        7m51s
aspen-mesh-dashboard           2d21h
istio-citadel                  2d21h
istio-egressgateway            2d21h
istio-galley                   2d21h
istio-ingressgateway           2d21h
istio-pilot                    2d21h
istio-policy                   2d21h
istio-sidecar-injector         2d21h
istio-telemetry                2d21h
istiocoredns                   2d21h
prometheus                     2d21h
traffic-claim-enforcer-webhook  2d21h

==> v1/ServiceAccount
NAME                                    AGE
aspen-mesh-controlplane                 2d21h
aspen-mesh-dashboard                    2d21h
istio-citadel-service-account           2d21h
istio-egressgateway-service-account     2d21h
istio-galley-service-account            2d21h
istio-ingressgateway-service-account    2d21h
istio-mixer-service-account             2d21h
istio-multi                             2d21h
istio-pilot-service-account             2d21h
istio-security-post-install-account     2d21h
istio-sidecar-injector-service-account  2d21h
istiocoredns-service-account            2d21h
prometheus                              2d21h
```

```
traffic-claim-enforcer-configurator-service-account  2d21h
traffic-claim-enforcer-service-account               2d21h

==> v1alpha2/attributemanifest
NAME         AGE
istioproxy   2d21h
kubernetes   2d21h

==> v1alpha2/handler
NAME           AGE
kubernetesenv  2d21h
prometheus     2d21h

==> v1alpha2/instance
NAME                   AGE
attributes             2d21h
requestcount           2d21h
requestduration        2d21h
requestsize            2d21h
responsesize           2d21h
tcpbytereceived        2d21h
tcpbytesent            2d21h
tcpconnectionsclosed   2d21h
tcpconnectionsopened   2d21h

==> v1alpha2/rule
NAME                      AGE
kubeattrgenrulerule       2d21h
promhttp                  2d21h
promtcp                   2d21h
promtcpconnectionclosed   2d21h
promtcpconnectionopen     2d21h
tcpkubeattrgenrulerule    2d21h

==> v1alpha3/DestinationRule
NAME                                AGE
istio-multicluster-destinationrule  2d21h
istio-policy                        2d21h
istio-telemetry                     2d21h

==> v1alpha3/EnvoyFilter
NAME                                AGE
istio-multicluster-ingressgateway   2d21h

==> v1alpha3/Gateway
NAME                                AGE
istio-multicluster-egressgateway    2d21h
istio-multicluster-ingressgateway   2d21h

==> v1beta1/ClusterRole
NAME                                      AGE
aspen-mesh-controlplane-view              2d21h
istio-security-post-install-istio-system  2d21h
traffic-claim-enforcer                    2d21h
traffic-claim-enforcer-configurator       2d21h

==> v1beta1/ClusterRoleBinding
NAME                                                AGE
aspen-mesh-controlplane                             2d21h
istio-security-post-install-role-binding-istio-system  2d21h
traffic-claim-enforcer                              2d21h
traffic-claim-enforcer-configurator                 2d21h

==> v1beta1/CustomResourceDefinition
NAME                                  AGE
trafficclaims.networking.aspenmesh.io  2d21h
```

```
==> v1beta1/MutatingWebhookConfiguration
NAME                     AGE
istio-sidecar-injector   2d21h

==> v1beta1/PodDisruptionBudget
NAME                     AGE
istio-egressgateway      2d21h
istio-galley             2d21h
istio-ingressgateway     2d21h
istio-pilot              2d21h
istio-policy             2d21h
istio-sidecar-injector   2d21h
istio-telemetry          2d21h

==> v1beta1/RoleBinding
NAME                                AGE
traffic-claim-enforcer-configurator  2d21h

==> v2beta1/HorizontalPodAutoscaler
NAME                   AGE
istio-egressgateway    2d21h
istio-ingressgateway   2d21h
istio-pilot            2d21h
istio-policy           2d21h
istio-telemetry        2d21h


NOTES:
Thank you for installing Istio.

Your release is named Istio.

To get started running application with Istio, execute the following steps:
1. Label namespace that application object will be deployed to by the following command (take
default namespace as an example)

$ kubectl label namespace default istio-injection=enabled
$ kubectl get namespace -L istio-injection

2. Deploy your applications

$ kubectl apply -f <your-application>.yaml

For more information on running Istio, visit:
https://istio.io/
```

```
fbchan@ck8s-1:~/aspenmesh-1.3.6-am3$ helm list
NAME            REVISION UPDATED                  STATUS   CHART                   APP VERSION     NAMESPACE
istio           7        Fri Dec 20 15:57:09 2019 DEPLOYED istio-1.3.6-am3        1.3.6-am3       istio-
system
istio-init      3        Fri Dec 20 15:59:32 2019 DEPLOYED istio-init-1.3.6-am3   1.3.6-am3       istio-
system
fbchan@ck8s-1:~/aspenmesh-1.3.6-am3$
```

NOTE: After making those changes, deployments of aspen-mesh-controlplane depend on the ConfigMap you created in the previous step. If you remove this ConfigMap later, you need to turn off the multiclusterEnabled: true in the values file as well, or new aspen-mesh-controlplane Pods will never start.

7 F5 CIS and AS3 Deployment Guide

7.1 Container Ingress Services Installation

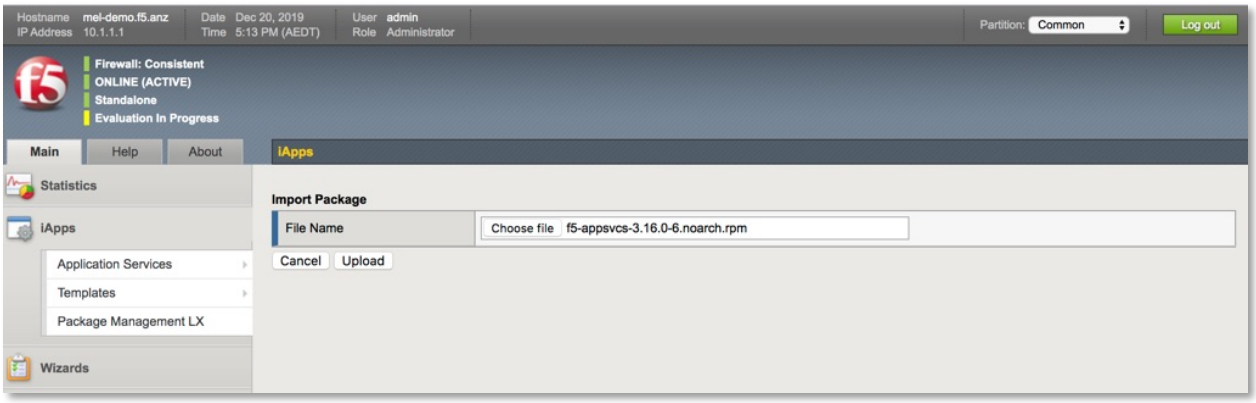
F5 Container Ingress Services (CIS) integrates with container orchestration environments to dynamically create L4/L7 services on F5 BIG-IP systems, and load balance network traffic across the services. Monitoring the orchestration API server, CIS is able to modify the BIG-IP system configuration based on changes made to containerized applications. Details can be obtained from https://clouddocs.f5.com/containers/v2/

```
Note:
Replace XXXXX with BIG-IP admin password.

fbchan@ck8s-1:~/cis$ kubectl create secret generic bigip-login -n kube-system --from-
literal=username=admin --from-literal=password=XXXXXXX
secret/bigip-login created

fbchan@ck8s-1:~/cis$ kubectl create serviceaccount k8s-bigip-ctlr -n kube-system
serviceaccount/k8s-bigip-ctlr created

fbchan@ck8s-1:~/cis$ kubectl create clusterrolebinding k8s-bigip-ctlr-clusteradmin --
clusterrole=cluster-admin --serviceaccount=kube-system:k8s-bigip-ctlr
clusterrolebinding.rbac.authorization.k8s.io/k8s-bigip-ctlr-clusteradmin created

fbchan@ck8s-1:~/cis$ kubectl apply -f cis-as3-deploy.yml
deployment.apps/k8s-bigip1-ctlr-deployment created

Note:
Replace '--bigip-url' with your BIG-IP management URL.

fbchan@ck8s-1:~/cis$ cat cis-as3-deploy.yml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: k8s-bigip1-ctlr-deployment
  namespace: kube-system
spec:
  selector:
    matchLabels:
      app: k8s-bigip-ctlr
  replicas: 1
  template:
    metadata:
      name: k8s-bigip-ctlr
      labels:
        app: k8s-bigip-ctlr
    spec:
      serviceAccountName: k8s-bigip-ctlr
      containers:
        - name: k8s-bigip-ctlr
          image: "f5networks/k8s-bigip-ctlr:latest"
          imagePullPolicy: IfNotPresent
          env:
            - name: BIGIP_USERNAME
              valueFrom:
                secretKeyRef:
                  name: bigip-login
                  key: username
            - name: BIGIP_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: bigip-login
                  key: password
```

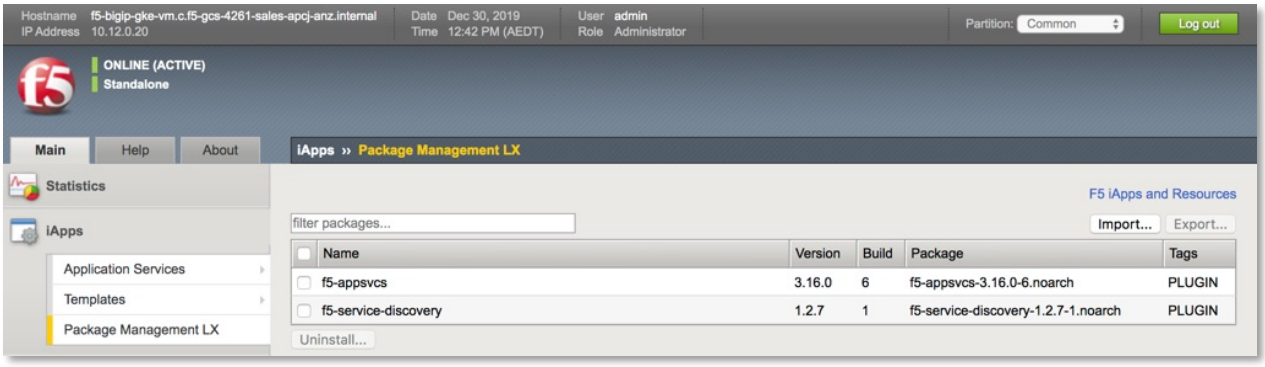
```
          command: ["/app/bin/k8s-bigip-ctlr"]
          args: [
            "--bigip-username=$(BIGIP_USERNAME)",
            "--bigip-password=$(BIGIP_PASSWORD)",
            "--bigip-url=10.1.1.1",
            "--bigip-partition=AS3",
            "--insecure=true",
            "--pool-member-type=cluster",
            "--manage-ingress=false"
            #"--log-level=debug"
            ]

fbchan@ck8s-1:~/cis$ kubectl -n kube-system get pod | grep bigip
k8s-bigip1-ctlr-deployment-595cf8d74f-lqzrq   1/1     Running   0          148m
```

7.2 AS3 Installation

Details installation can be obtained from https://clouddocs.f5.com/products/extensions/f5-appsvcs-extension/latest/userguide/installation.html. Alternatively, download .rpm and install via F5 Package Management LX.



Click upload and AS3 will be installed on BIG-IP.



8 Uninstall Aspen Mesh and clean up CRD.

In the event if you need to completely and cleanly remove Aspen Mesh / Istio for rebuilt, follow the following instructions

```
fbchan@ck8s-1:~$ helm list
NAME            REVISION UPDATED                  STATUS   CHART                  APP VERSION     NAMESPACE
istio           1        Fri Dec 13 16:57:54 2019 DEPLOYED istio-1.3.6-am1       1.3.6-am1       istio-
system
istio-init      1        Fri Dec 13 16:57:22 2019 DEPLOYED istio-init-1.3.6-am1  1.3.6-am1       istio-
system

fbchan@ck8s-1:~$ helm delete --purge istio && helm delete --purge istio-init
release "istio" deleted
release "istio-init" deleted

fbchan@ck8s-1:~$ kubectl get customresourcedefinition  -n istio-system | grep 'istio'|awk
'{print $1}'|xargs kubectl delete customresourcedefinition  -n istio-system
warning: deleting cluster-scoped resources, not scoped to the provided namespace
customresourcedefinition.apiextensions.k8s.io "adapters.config.istio.io" deleted
customresourcedefinition.apiextensions.k8s.io "attributemanifests.config.istio.io" deleted
customresourcedefinition.apiextensions.k8s.io "authorizationpolicies.rbac.istio.io" deleted
customresourcedefinition.apiextensions.k8s.io "clusterrbacconfigs.rbac.istio.io" deleted
customresourcedefinition.apiextensions.k8s.io "destinationrules.networking.istio.io" deleted
customresourcedefinition.apiextensions.k8s.io "envoyfilters.networking.istio.io" deleted
customresourcedefinition.apiextensions.k8s.io "gateways.networking.istio.io" deleted
customresourcedefinition.apiextensions.k8s.io "handlers.config.istio.io" deleted
customresourcedefinition.apiextensions.k8s.io "httpapispecbindings.config.istio.io" deleted
customresourcedefinition.apiextensions.k8s.io "httpapispecs.config.istio.io" deleted
customresourcedefinition.apiextensions.k8s.io "instances.config.istio.io" deleted
customresourcedefinition.apiextensions.k8s.io "meshpolicies.authentication.istio.io" deleted
customresourcedefinition.apiextensions.k8s.io "policies.authentication.istio.io" deleted
customresourcedefinition.apiextensions.k8s.io "quotaspecbindings.config.istio.io" deleted
customresourcedefinition.apiextensions.k8s.io "quotaspecs.config.istio.io" deleted
customresourcedefinition.apiextensions.k8s.io "rbacconfigs.rbac.istio.io" deleted
customresourcedefinition.apiextensions.k8s.io "rules.config.istio.io" deleted
customresourcedefinition.apiextensions.k8s.io "serviceentries.networking.istio.io" deleted
customresourcedefinition.apiextensions.k8s.io "servicerolebindings.rbac.istio.io" deleted
customresourcedefinition.apiextensions.k8s.io "serviceroles.rbac.istio.io" deleted
customresourcedefinition.apiextensions.k8s.io "sidecars.networking.istio.io" deleted
customresourcedefinition.apiextensions.k8s.io "templates.config.istio.io" deleted
customresourcedefinition.apiextensions.k8s.io "virtualservices.networking.istio.io" deleted

fbchan@ck8s-1:~/aspenmesh-1.3.6-am1$ kubectl -n istio-apps delete -f
./samples/bookinfo/platform/kube/bookinfo.yaml
service "details" deleted
serviceaccount "bookinfo-details" deleted
deployment.apps "details-v1" deleted
service "ratings" deleted
serviceaccount "bookinfo-ratings" deleted
deployment.apps "ratings-v1" deleted
service "reviews" deleted
serviceaccount "bookinfo-reviews" deleted
deployment.apps "reviews-v1" deleted
deployment.apps "reviews-v2" deleted
deployment.apps "reviews-v3" deleted
service "productpage" deleted
serviceaccount "bookinfo-productpage" deleted
deployment.apps "productpage-v1" deleted

fbchan@k8s-1:~/aspenmesh-1.3.6-am1$ kubectl delete -f ./samples/aspenmesh/bookinfo-traffic-
generator.yaml
configmap "traffic-generator-productpage" deleted
deployment.apps "traffic-generator-productpage" deleted

fbchan@k8s-1:~/aspenmesh-1.3.6-am1$ kubectl delete namespace istio-apps
namespace "istio-apps" deleted
```

```
fbchan@k8s-1:~/aspenmesh-1.3.6-am1$ kubectl delete namespace istio-system
namespace "istio-system" deleted
```

9 Deployment Guide for dependent tools

Cross cluster communication requires mutual TLS connection between services. To enable mutual TLS communication across clusters, each cluster's Citadel will be configured with unique intermediate CA credentials generated by a shared/common root CA.

9.1 Generating Root and Intermediate CA

The following steps are reference from https://roll.urown.net/ca/ca_root_setup.html. Please refer to the link for details.

Creating Root CA

```
fbchan@logos:~$ mkdir -p foobz.lab.ca/root-ca/{certreqs,certs,crl,newcerts,private}
fbchan@logos:~$ cd foobz.lab.ca/
fbchan@logos:~/foobz.lab.ca$ cd root-ca/
fbchan@logos:~/foobz.lab.ca/root-ca$ ls
certreqs  certs  crl  newcerts  private
fbchan@logos:~/foobz.lab.ca/root-ca$ chmod 700 private
fbchan@logos:~/foobz.lab.ca/root-ca$ touch root-ca.index
fbchan@logos:~/foobz.lab.ca/root-ca$ echo 00 > root-ca.crlnum
fbchan@logos:~/foobz.lab.ca/root-ca$ openssl rand -hex 16 > root-ca.serial
fbchan@logos:~/foobz.lab.ca/root-ca$ vi root-ca.cnf
fbchan@logos:~/foobz.lab.ca/root-ca$ export OPENSSL_CONF=./root-ca.cnf
fbchan@logos:~/foobz.lab.ca/root-ca$ openssl req -new -out root-ca.req.pem
Generating a 4096 bit RSA private key
.....................................................++
.....................................................................................++
writing new private key to 'private/root-ca.key.pem'
Enter PEM pass phrase: *******
Verifying - Enter PEM pass phrase: *******
-----
fbchan@logos:~/foobz.lab.ca/root-ca$ chmod 400 private/root-ca.key.pem
```

Generate CSR

```
openssl req -new \
    -key private/root-ca.key.pem \
    -out root-ca.req.pem

fbchan@logos:~/foobz.lab.ca/root-ca$ openssl req -new \
>     -key private/root-ca.key.pem \
>     -out root-ca.req.pem
Enter pass phrase for private/root-ca.key.pem:
```

Validate CSR

```
openssl req -verify -in root-ca.req.pem \
    -noout -text \
    -reqopt no_version,no_pubkey,no_sigdump \
    -nameopt multiline

fbchan@logos:~/foobz.lab.ca/root-ca$ openssl req -verify -in root-ca.req.pem \
>     -noout -text \
>     -reqopt no_version,no_pubkey,no_sigdump \
>     -nameopt multiline
verify OK
Certificate Request:
    Data:
        Subject:
```

```
            organizationName          = FOOBZ
            commonName                = FOOBZ Root CA
        Attributes:
        Requested Extensions:
            X509v3 Subject Key Identifier:
                D5:CD:6F:FC:6B:06:FB:E0:3B:A0:CC:7F:5D:98:E2:45:A4:E7:E8:13
            X509v3 Subject Alternative Name:
                URI:http://ca.foobz.lab/, email:certmaster@foobz.lab
```

Signing Root certificate

```
fbchan@logos:~/foobz.lab.ca/root-ca$ openssl rand -hex 16 > root-ca.serial

openssl ca -selfsign \
    -in root-ca.req.pem \
    -out root-ca.cert.pem \
    -extensions root-ca_ext \
    -startdate `date +%y%m%d000000Z -u -d -1day` \
    -enddate `date +%y%m%d000000Z -u -d +10years+1day`

fbchan@logos:~/foobz.lab.ca/root-ca$ openssl ca -selfsign \
>     -in root-ca.req.pem \
>     -out root-ca.cert.pem \
>     -extensions root-ca_ext \
>     -startdate `date +%y%m%d000000Z -u -d -1day` \
>     -enddate `date +%y%m%d000000Z -u -d +10years+1day`
Using configuration from ./root-ca.cnf
Enter pass phrase for ./private/root-ca.key.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            05:22:2f:e8:cb:b5:24:3b:1e:a7:03:1d:92:bc:fe:cf
        Issuer:
            organizationName          = FOOBZ
            commonName                = FOOBZ Root CA
        Validity
            Not Before: Dec 12 00:00:00 2019 GMT
            Not After : Dec 14 00:00:00 2029 GMT
        Subject:
            organizationName          = FOOBZ
            commonName                = FOOBZ Root CA
        X509v3 extensions:
            X509v3 Basic Constraints: critical
                CA:TRUE
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Subject Key Identifier:
                D5:CD:6F:FC:6B:06:FB:E0:3B:A0:CC:7F:5D:98:E2:45:A4:E7:E8:13
            X509v3 Authority Key Identifier:
                keyid:D5:CD:6F:FC:6B:06:FB:E0:3B:A0:CC:7F:5D:98:E2:45:A4:E7:E8:13

            X509v3 Issuer Alternative Name:
                <EMPTY>

            Authority Information Access:
                CA Issuers -
URI:http://ca.foobz.lab/certs/foobz.lab_Root_Certification_Authority.cert.pem

            X509v3 CRL Distribution Points:

                Full Name:
                  URI:http://ca.foobz.lab/crl/foobz.lab_Root_Certification_Authority.crl
```

```
            X509v3 Subject Alternative Name:
                URI:http://ca.foobz.lab/, email:certmaster@foobz.lab
Certificate is to be certified until Dec 14 00:00:00 2029 GMT (3653 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Ensure certificate valid

```
openssl verify -verbose -CAfile root-ca.cert.pem \
    root-ca.cert.pem

fbchan@logos:~/foobz.lab.ca/root-ca$ openssl verify -verbose -CAfile root-ca.cert.pem \
>     root-ca.cert.pem
root-ca.cert.pem: OK
```

Example root-ca.cnf

```
#
# OpenSSL configuration for the Root Certification Authority.
#

#
# This definition doesn't work if HOME isn't defined.
CA_HOME                 = .
RANDFILE                = $ENV::CA_HOME/private/.rnd

#
# Default Certification Authority
[ ca ]
default_ca              = root_ca

#
# Root Certification Authority
[ root_ca ]
dir                     = $ENV::CA_HOME
certs                   = $dir/certs
serial                  = $dir/root-ca.serial
database                = $dir/root-ca.index
new_certs_dir           = $dir/newcerts
certificate             = $dir/root-ca.cert.pem
private_key             = $dir/private/root-ca.key.pem
default_days            = 1826 # Five years
crl                     = $dir/root-ca.crl
crl_dir                 = $dir/crl
crlnumber               = $dir/root-ca.crlnum
name_opt                = multiline, align
cert_opt                = no_pubkey
copy_extensions         = copy
crl_extensions          = crl_ext
default_crl_days        = 180
default_md              = sha256
preserve                = no
email_in_dn             = no
policy                  = policy
unique_subject          = no

#
# Distinguished Name Policy for CAs
[ policy ]
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = supplied
organizationalUnitName  = optional
commonName              = supplied
```

```
#
# Root CA Request Options
[ req ]
default_bits            = 4096
default_keyfile         = private/root-ca.key.pem
encrypt_key             = yes
default_md              = sha256
string_mask             = utf8only
utf8                    = yes
prompt                  = no
req_extensions          = root-ca_req_ext
distinguished_name      = distinguished_name
subjectAltName          = @subject_alt_name

#
# Root CA Request Extensions
[ root-ca_req_ext ]
subjectKeyIdentifier    = hash
subjectAltName          = @subject_alt_name

#
# Distinguished Name (DN)
[ distinguished_name ]
organizationName        = FOOBZ
commonName              = FOOBZ Root CA

#
# Root CA Certificate Extensions
[ root-ca_ext ]
basicConstraints        = critical, CA:true
keyUsage                = critical, keyCertSign, cRLSign
#nameConstraints         = critical, @name_constraints
subjectKeyIdentifier    = hash
#subjectAltName          = @subject_alt_name
authorityKeyIdentifier  = keyid:always
issuerAltName           = issuer:copy
authorityInfoAccess     = @auth_info_access
crlDistributionPoints   = crl_dist

#
# Intermediate CA Certificate Extensions
[ intermed-ca_ext ]
basicConstraints        = critical, CA:true, pathlen:0
keyUsage                = critical, keyCertSign, cRLSign
subjectKeyIdentifier    = hash
subjectAltName          = @subject_alt_name
authorityKeyIdentifier  = keyid:always
issuerAltName           = issuer:copy
authorityInfoAccess     = @auth_info_access
crlDistributionPoints   = crl_dist

#
# CRL Certificate Extensions
[ crl_ext ]
authorityKeyIdentifier  = keyid:always
issuerAltName           = issuer:copy

#
# Certificate Authorities Alternative Names
[ subject_alt_name ]
URI                     = http://ca.foobz.lab/
email                   = certmaster@foobz.lab

#
# Name Constraints
[ name_constraints ]
permitted;DNS.1         = foobz.lab
permitted;DNS.2         = foobz.local
permitted;DNS.3         = *.foobz.local
permitted;DNS.4         = *.foobz.lab
#
# Certificate download addresses for the root CA
[ auth_info_access ]
caIssuers;URI           = http://ca.foobz.lab/certs/foobz.lab_Root_Certification_Authority.cert.pem
```

```
#
# CRL Download address for the root CA
[ crl_dist ]
fullname                = URI:http://ca.foobz.lab/crl/foobz.lab_Root_Certification_Authority.crl

# EOF
```

9.2 Create Intermediate CAs for ISTIO Citadel

For simplicity, this generate 4 individual intermediate CA to be used by 4 different ISTIO cluster

```
fbchan@logos:~$ mkdir -p foobz.lab.ca/intermed-ca1/{certreqs,certs,crl,newcerts,private}
fbchan@logos:~$ mkdir -p foobz.lab.ca/intermed-ca2/{certreqs,certs,crl,newcerts,private}
fbchan@logos:~$ mkdir -p foobz.lab.ca/intermed-ca3/{certreqs,certs,crl,newcerts,private}
fbchan@logos:~$ mkdir -p foobz.lab.ca/intermed-ca4/{certreqs,certs,crl,newcerts,private}
fbchan@logos:~$ cd foobz.lab.ca/intermed-ca1/
fbchan@logos:~/foobz.lab.ca/intermed-ca1$ ls
certreqs  certs  crl  newcerts  private
fbchan@logos:~/foobz.lab.ca/intermed-ca1$ chmod 700 private/
fbchan@logos:~/foobz.lab.ca/intermed-ca1$ chmod 700 ../intermed-ca2/private/
fbchan@logos:~/foobz.lab.ca/intermed-ca1$ chmod 700 ../intermed-ca3/private/
fbchan@logos:~/foobz.lab.ca/intermed-ca1$ chmod 700 ../intermed-ca4/private/
fbchan@logos:~/foobz.lab.ca/intermed-ca1$ touch intermed-ca.index
fbchan@logos:~/foobz.lab.ca/intermed-ca1$ touch ../intermed-ca2/intermed-ca.index
fbchan@logos:~/foobz.lab.ca/intermed-ca1$ touch ../intermed-ca3/intermed-ca.index
fbchan@logos:~/foobz.lab.ca/intermed-ca1$ touch ../intermed-ca4/intermed-ca.index
fbchan@logos:~/foobz.lab.ca/intermed-ca1$ echo 00 > intermed-ca.crlnum
fbchan@logos:~/foobz.lab.ca/intermed-ca1$ echo 00 > ../intermed-ca2/intermed-ca.crlnum
fbchan@logos:~/foobz.lab.ca/intermed-ca1$ echo 00 > ../intermed-ca3/intermed-ca.crlnum
fbchan@logos:~/foobz.lab.ca/intermed-ca1$ echo 00 > ../intermed-ca4/intermed-ca.crlnum
fbchan@logos:~/foobz.lab.ca/intermed-ca1$ openssl rand -hex 16 > intermed-ca.serial
fbchan@logos:~/foobz.lab.ca/intermed-ca1$ openssl rand -hex 16 > ../intermed-ca2/intermed-
ca.serial
fbchan@logos:~/foobz.lab.ca/intermed-ca1$ openssl rand -hex 16 > ../intermed-ca3/intermed-
ca.serial
fbchan@logos:~/foobz.lab.ca/intermed-ca1$ openssl rand -hex 16 > ../intermed-ca4/intermed-
ca.serial
```

Example intermed-ca.cnf

```
#
# OpenSSL configuration for the Intermediate Certification Authority.
#

#
# This definition doesn't work if HOME isn't defined.
CA_HOME                 = .
RANDFILE                = $ENV::CA_HOME/private/.rnd
oid_section             = new_oids

#
# XMPP address Support
[ new_oids ]
xmppAddr          = 1.3.6.1.5.5.7.8.5
dnsSRV            = 1.3.6.1.5.5.7.8.7

#
# Default Certification Authority
[ ca ]
default_ca              = intermed_ca

#
# Intermediate Certification Authority
[ intermed_ca ]
dir                     = $ENV::CA_HOME
certs                   = $dir/certs
serial                  = $dir/intermed-ca.serial
database                = $dir/intermed-ca.index
```

```
new_certs_dir           = $dir/newcerts
certificate             = $dir/intermed-ca.cert.pem
private_key             = $dir/private/intermed-ca.key.pem
default_days            = 730 # Two years
crl                     = $dir/crl/intermed-ca.crl
crl_dir                 = $dir/crl
crlnumber               = $dir/intermed-ca.crlnum
name_opt                = multiline, align
cert_opt                = no_pubkey
copy_extensions         = copy
crl_extensions          = crl_ext
default_crl_days        = 30
default_md              = sha256
preserve                = no
email_in_dn             = no
policy                  = policy
unique_subject          = no

#
# Distinguished Name Policy
[ policy ]
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied

#
# Distinguished Name Policy for Personal Certificates
[ user_policy ]
countryName             = supplied
stateOrProvinceName     = optional
localityName            = supplied
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = supplied
#xmppAddr               = optional # Added to SubjAltName by req

#
# Intermediate CA request options
[ req ]
default_bits            = 3072
default_keyfile         = private/intermed-ca.key.pem
encrypt_key             = no
default_md              = sha256
string_mask             = utf8only
utf8                    = yes
prompt                  = no
req_extensions          = req_ext
distinguished_name      = distinguished_name
subjectAltName          = subject_alt_name

#
# Intermediate CA Request Extensions
[ req_ext ]
subjectKeyIdentifier    = hash
subjectAltName          = @subject_alt_name

#
# Distinguished Name (DN)
[ distinguished_name ]
organizationName        = ISTIO
commonName              = FOOBZ Istio CA1

#
# Server Certificate Extensions
[ server_ext ]
basicConstraints        = CA:FALSE
keyUsage                = critical, digitalSignature, keyEncipherment
extendedKeyUsage        = critical, serverAuth, clientAuth
subjectKeyIdentifier    = hash
authorityKeyIdentifier  = keyid:always
```

```
issuerAltName           = issuer:copy
authorityInfoAccess     = @auth_info_access
crlDistributionPoints   = crl_dist

#
# Client Certificate Extensions
[ client_ext ]
basicConstraints        = CA:FALSE
keyUsage                = critical, digitalSignature
extendedKeyUsage        = critical, clientAuth
subjectKeyIdentifier    = hash
authorityKeyIdentifier  = keyid:always
issuerAltName           = issuer:copy
authorityInfoAccess     = @auth_info_access
crlDistributionPoints   = crl_dist

#
# User Certificate Extensions
[ user_ext ]
basicConstraints        = CA:FALSE
keyUsage                = critical, digitalSignature
extendedKeyUsage        = critical, clientAuth, emailProtection
subjectKeyIdentifier    = hash
authorityKeyIdentifier  = keyid:always
issuerAltName           = issuer:copy
authorityInfoAccess     = @auth_info_access
crlDistributionPoints   = crl_dist

#
# CRL Certificate Extensions
[ crl_ext ]
authorityKeyIdentifier  = keyid:always
issuerAltName           = issuer:copy

#
# Certificate Authorities Alternative Names
[ subject_alt_name ]
URI                     = http://istio.foobz.lab/
email                   = certmaster@foobz.lab

#
# Certificate download addresses for the intermediate CA
[ auth_info_access ]
caIssuers;URI           =
http://istio.foobz.lab/certs/foobz.lab_Intermediate_Certification_Authority.cert.pem

#
# CRL Download address for the intermediate CA
[ crl_dist ]
fullname                = URI:http://istio.foobz.lab/crl/foobz.lab_Intermediate_Certification_Authority.crl

# EOF
```

Each intermed-ca.cnf with different common name.

Repeat for all 4 intermediate CA

```
fbchan@logos:~/foobz.lab.ca/intermed-ca1$ export OPENSSL_CONF=./intermed-ca.cnf


fbchan@logos:~/foobz.lab.ca/intermed-ca1$ openssl req -new -out intermed-ca.req.pem
Generating a 3072 bit RSA private key
.....................................................................................++
.....................................................++
writing new private key to 'private/intermed-ca.key.pem'
-----
fbchan@logos:~/foobz.lab.ca/intermed-ca1$
fbchan@logos:~/foobz.lab.ca/intermed-ca1$
fbchan@logos:~/foobz.lab.ca/intermed-ca1$ chmod 400 private/intermed-ca.key.pem

openssl req -new \
    -key private/intermed-ca.key.pem \
    -out intermed-ca.req.pem
```

```
fbchan@logos:~/foobz.lab.ca/intermed-ca1$ openssl req -new \
>     -key private/intermed-ca.key.pem \
>     -out intermed-ca.req.pem

openssl req  -verify -in intermed-ca.req.pem \
    -noout -text \
    -reqopt no_version,no_pubkey,no_sigdump \
    -nameopt multiline


fbchan@logos:~/foobz.lab.ca/intermed-ca1$ openssl req  -verify -in intermed-ca.req.pem \
>     -noout -text \
>     -reqopt no_version,no_pubkey,no_sigdump \
>     -nameopt multiline
verify OK
Certificate Request:
    Data:
        Subject:
            organizationName          = ISTIO
            commonName                = FOOBZ Istio CA1
        Attributes:
        Requested Extensions:
            X509v3 Subject Key Identifier:
                90:4D:D6:DC:4F:A9:FB:D2:BC:CD:B1:88:65:9A:08:B1:A6:85:82:31
            X509v3 Subject Alternative Name:
                URI:http://istio.foobz.lab/, email:certmaster@foobz.lab

fbchan@logos:~/foobz.lab.ca/intermed-ca1$ cp intermed-ca.req.pem ../root-ca/certreqs/intermed-
ca1.req.pem
```

Sign intermediate certificate with root-ca

```
fbchan@logos:~/foobz.lab.ca/intermed-ca1$ cd ../root-ca/
fbchan@logos:~/foobz.lab.ca/root-ca$ export OPENSSL_CONF=./root-ca.cnf
fbchan@logos:~/foobz.lab.ca/root-ca$ openssl rand -hex 16 > root-ca.serial

openssl ca \
    -in certreqs/intermed-ca1.req.pem \
    -out certs/intermed-ca1.cert.pem \
    -extensions intermed-ca_ext \
    -startdate `date +%y%m%d000000Z -u -d -1day` \
    -enddate `date +%y%m%d000000Z -u -d +5years+1day`

fbchan@logos:~/foobz.lab.ca/root-ca$ openssl ca \
>     -in certreqs/intermed-ca1.req.pem \
>     -out certs/intermed-ca1.cert.pem \
>     -extensions intermed-ca_ext \
>     -startdate `date +%y%m%d000000Z -u -d -1day` \
>     -enddate `date +%y%m%d000000Z -u -d +5years+1day`
Using configuration from ./root-ca.cnf
Enter pass phrase for ./private/root-ca.key.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            e6:bd:f5:b4:bf:e5:13:c8:b1:dd:c2:74:2f:62:e5:c1
        Issuer:
            organizationName          = FOOBZ
            commonName                = FOOBZ Root CA
        Validity
```

```
            Not Before: Dec 15 00:00:00 2019 GMT
            Not After : Dec 17 00:00:00 2024 GMT
        Subject:
            organizationName          = ISTIO
            commonName                = FOOBZ Istio CA1
        X509v3 extensions:
            X509v3 Basic Constraints: critical
                CA:TRUE, pathlen:0
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Subject Key Identifier:
                90:4D:D6:DC:4F:A9:FB:D2:BC:CD:B1:88:65:9A:08:B1:A6:85:82:31
            X509v3 Subject Alternative Name:
                URI:http://ca.foobz.lab/, email:certmaster@foobz.lab
            X509v3 Authority Key Identifier:
                keyid:D5:CD:6F:FC:6B:06:FB:E0:3B:A0:CC:7F:5D:98:E2:45:A4:E7:E8:13

            X509v3 Issuer Alternative Name:
                URI:http://ca.foobz.lab/, email:certmaster@foobz.lab
            Authority Information Access:
                CA Issuers -
URI:http://ca.foobz.lab/certs/foobz.lab_Root_Certification_Authority.cert.pem

            X509v3 CRL Distribution Points:

                Full Name:
                  URI:http://ca.foobz.lab/crl/foobz.lab_Root_Certification_Authority.crl

Certificate is to be certified until Dec 17 00:00:00 2024 GMT (1827 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated


openssl x509 -in certs/intermed-ca1.cert.pem \
    -noout -text \
    -certopt no_version,no_pubkey,no_sigdump \
    -nameopt multiline

fbchan@logos:~/foobz.lab.ca/root-ca$ openssl x509 -in certs/intermed-ca1.cert.pem \
>     -noout -text \
>     -certopt no_version,no_pubkey,no_sigdump \
>     -nameopt multiline
Certificate:
    Data:
        Serial Number:
            e6:bd:f5:b4:bf:e5:13:c8:b1:dd:c2:74:2f:62:e5:c1
    Signature Algorithm: sha256WithRSAEncryption
        Issuer:
            organizationName          = FOOBZ
            commonName                = FOOBZ Root CA
        Validity
            Not Before: Dec 15 00:00:00 2019 GMT
            Not After : Dec 17 00:00:00 2024 GMT
        Subject:
            organizationName          = ISTIO
            commonName                = FOOBZ Istio CA1
        X509v3 extensions:
            X509v3 Basic Constraints: critical
                CA:TRUE, pathlen:0
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Subject Key Identifier:
```

```
                90:4D:D6:DC:4F:A9:FB:D2:BC:CD:B1:88:65:9A:08:B1:A6:85:82:31
            X509v3 Subject Alternative Name:
                URI:http://ca.foobz.lab/, email:certmaster@foobz.lab
            X509v3 Authority Key Identifier:
                keyid:D5:CD:6F:FC:6B:06:FB:E0:3B:A0:CC:7F:5D:98:E2:45:A4:E7:E8:13

            X509v3 Issuer Alternative Name:
                URI:http://ca.foobz.lab/, email:certmaster@foobz.lab
            Authority Information Access:
                CA Issuers -
URI:http://ca.foobz.lab/certs/foobz.lab_Root_Certification_Authority.cert.pem

            X509v3 CRL Distribution Points:

                Full Name:
                  URI:http://ca.foobz.lab/crl/foobz.lab_Root_Certification_Authority.crl
```

Verify to ensure intermediate CA working

```
openssl verify -verbose -CAfile root-ca.cert.pem \
    certs/intermed-ca1.cert.pem

fbchan@logos:~/foobz.lab.ca/root-ca$ openssl verify -verbose -CAfile root-ca.cert.pem \
>     certs/intermed-ca1.cert.pem
certs/intermed-ca1.cert.pem: OK

fbchan@logos:~/foobz.lab.ca/root-ca$ cp certs/intermed-ca1.cert.pem ../intermed-ca1/
```

Prepare root-ca and intermediate-ca for Istio cluster

```
fbchan@logos:~/foobz.lab.ca$ cd foobz-istio-cert/
fbchan@logos:~/foobz.lab.ca/foobz-istio-cert$ cp ../root-ca/root-ca.cert.pem .
fbchan@logos:~/foobz.lab.ca/foobz-istio-cert$ mkdir ca1
fbchan@logos:~/foobz.lab.ca/foobz-istio-cert$ mkdir ca2
fbchan@logos:~/foobz.lab.ca/foobz-istio-cert$ mkdir ca3
fbchan@logos:~/foobz.lab.ca/foobz-istio-cert$ mkdir ca4
fbchan@logos:~/foobz.lab.ca/foobz-istio-cert$ cp ../intermed-ca1/intermed-ca1.cert.pem ca1/
fbchan@logos:~/foobz.lab.ca/foobz-istio-cert$ cp ../intermed-ca1/private/intermed-ca.key.pem
ca1/intermed-ca1.key.pem

fbchan@logos:~/foobz.lab.ca/foobz-istio-cert$ cp ../intermed-ca2/intermed-ca2.cert.pem ca2/
fbchan@logos:~/foobz.lab.ca/foobz-istio-cert$ cp ../intermed-ca2/private/intermed-ca.key.pem
ca2/intermed-ca2.key.pem

fbchan@logos:~/foobz.lab.ca/foobz-istio-cert$ cp ../intermed-ca3/intermed-ca3.cert.pem ca3/
fbchan@logos:~/foobz.lab.ca/foobz-istio-cert$ cp ../intermed-ca3/private/intermed-ca.key.pem
ca3/intermed-ca4.key.pem

fbchan@logos:~/foobz.lab.ca/foobz-istio-cert$ cp ../intermed-ca4/intermed-ca4.cert.pem ca4/
fbchan@logos:~/foobz.lab.ca/foobz-istio-cert$ cp ../intermed-ca4/private/intermed-ca.key.pem
ca4/intermed-ca4.key.pem

fbchan@logos:~/foobz.lab.ca/foobz-istio-cert$ ls -al
total 32
drwxrwxr-x  6 fbchan fbchan 4096 Dec 16 11:28 .
drwxrwxr-x 10 fbchan fbchan 4096 Dec 16 11:28 ..
drwxrwxr-x  2 fbchan fbchan 4096 Dec 16 11:29 ca1
drwxrwxr-x  2 fbchan fbchan 4096 Dec 16 11:30 ca2
drwxrwxr-x  2 fbchan fbchan 4096 Dec 16 11:41 ca3
drwxrwxr-x  2 fbchan fbchan 4096 Dec 16 11:40 ca4
-rw-rw-r--  1 fbchan fbchan 7870 Dec 16 11:28 root-ca.cert.pem
fbchan@logos:~/foobz.lab.ca/foobz-istio-cert$ du -a
8       ./root-ca.cert.pem
8       ./ca4/intermed-ca4.cert.pem
8       ./ca4/intermed-ca4.key.pem
```

```
20      ./ca4
4       ./ca2/intermed-ca2.key.pem
8       ./ca2/intermed-ca2.cert.pem
16      ./ca2
8       ./ca3/intermed-ca3.key.pem
8       ./ca3/intermed-ca3.cert.pem
20      ./ca3
4       ./ca1/intermed-ca1.key.pem
8       ./ca1/intermed-ca1.cert.pem
16      ./ca1
84      .
```

Prepare 4 PEM file in accordance to ISTIO multi-cluster requirement. PEM file only require '---BEGIN CERTIFICATE --- till to --- END CERTIFICATE --' section.

```
fbchan@logos:~/foobz.lab.ca/foobz-istio-cert/ca1$ ls
intermed-ca1.cert.pem  intermed-ca1.key.pem
fbchan@logos:~/foobz.lab.ca/foobz-istio-cert/ca1$ vi ca-cert.pem
fbchan@logos:~/foobz.lab.ca/foobz-istio-cert/ca1$ vi ca-key.pem
fbchan@logos:~/foobz.lab.ca/foobz-istio-cert/ca1$ cat ca-cert.pem > cert-chain.pem
fbchan@logos:~/foobz.lab.ca/foobz-istio-cert/ca1$ vi root-cert.pem

fbchan@logos:~/foobz.lab.ca/foobz-istio-cert/ca1$ ls -l
total 28
-rw-rw-r-- 1 fbchan fbchan 2122 Dec 16 11:43 ca-cert.pem
-rw-rw-r-- 1 fbchan fbchan 2484 Dec 16 11:43 ca-key.pem
-rw-rw-r-- 1 fbchan fbchan 2122 Dec 16 11:44 cert-chain.pem
-rw-rw-r-- 1 fbchan fbchan 7248 Dec 16 11:29 intermed-ca1.cert.pem
-r-------- 1 fbchan fbchan 2484 Dec 16 11:29 intermed-ca1.key.pem
-rw-rw-r-- 1 fbchan fbchan 2228 Dec 16 11:44 root-cert.pem
fbchan@logos:~/foobz.lab.ca/foobz-istio-cert/ca1$
```

9.3 Install Prometheus Monitoring

Aspen Mesh depend on Prometheus for metric collection. Hence, ensure Prometheus is installed prior installation of Aspen Mesh. Prometheus URL and port are required as part of the input value for Aspen Mesh installation. Aspen Mesh can also reference to an existing Prometheus. Please ensure Prometheus is configured to auto-discover and scrape new pod in Kubernetes as shown in the following section.

Note: Various installation guide can be obtained from Internet or directly from official Prometheus website. The following installation procedure are being used. Configuration manifest files used can be obtained from https://github.com/fbchan/aspen-mesh-multi-cluster.

Prometheus namespace and RBAC

```
fbchan@ck8s-1:~/monitoring$ kubectl apply -f 1_namespace.yml
namespace/monitoring created

fbchan@ck8s-1:~/monitoring$ kubectl apply -f 2_prometheus-rbac.yml
serviceaccount/prometheus created
clusterrole.rbac.authorization.k8s.io/prometheus created
clusterrolebinding.rbac.authorization.k8s.io/prometheus created
```

Prometheus configuration

Prometheus must be configured to auto-discover and scrape new pod in Kubernetes cluster in order for Aspen Mesh to report visualization.

```
fbchan@ck8s-1:~/monitoring$ kubectl apply -f 3_prometheus-config.yml
configmap/prometheus-config created

fbchan@ck8s-1:~/monitoring$ kubectl -n monitoring describe configmap prometheus-config
Name:         prometheus-config
Namespace:    monitoring
Labels:       <none>
Annotations:  kubectl.kubernetes.io/last-applied-configuration:
                {"apiVersion":"v1","data":{"prometheus.yml":"global:\nscrape_configs:\n -
job_name: 'kubernetes-kubelet'\n   scheme: https\n   tls_config:...

Data
====
prometheus.yml:
----
global:
scrape_configs:
 - job_name: 'kubernetes-kubelet'
   scheme: https
   tls_config:
     ca_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
     insecure_skip_verify: true
   bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
   kubernetes_sd_configs:
   - role: node
   relabel_configs:
   - action: labelmap
     regex: __meta_kubernetes_node_label_(.+)
   - target_label: __address__
     replacement: kubernetes.default.svc.cluster.local:443
   - source_labels: [__meta_kubernetes_node_name]
     regex: (.+)
     target_label: __metrics_path__
```

```
      replacement: /api/v1/nodes/${1}/proxy/metrics
  - job_name: 'kubernetes-cadvisor'
    scheme: https
    tls_config:
      ca_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
      insecure_skip_verify: true
    bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
    kubernetes_sd_configs:
    - role: node
    relabel_configs:
    - action: labelmap
      regex: __meta_kubernetes_node_label_(.+)
    - target_label: __address__
      replacement: kubernetes.default.svc.cluster.local:443
    - source_labels: [__meta_kubernetes_node_name]
      regex: (.+)
      target_label: __metrics_path__
      replacement: /api/v1/nodes/${1}/proxy/metrics/cadvisor
  - job_name: 'kubernetes-kube-state'
    kubernetes_sd_configs:
    - role: pod
    relabel_configs:
    - action: labelmap
      regex: __meta_kubernetes_pod_label_(.+)
    - source_labels: [__meta_kubernetes_namespace]
      action: replace
      target_label: kubernetes_namespace
    - source_labels: [__meta_kubernetes_pod_name]
      action: replace
      target_label: kubernetes_pod_name
    - source_labels: [__meta_kubernetes_pod_label_grafanak8sapp]
      regex: .*true.*
      action: keep
    - source_labels: ['__meta_kubernetes_pod_label_daemon', '__meta_kubernetes_pod_node_name']
      regex: 'node-exporter;(.*)'
      action: replace
      target_label: nodename
  - job_name: kubernetes
    kubernetes_sd_configs:
    - api_server: 'https://kubernetes.default.svc'
      role: pod
      tls_config:
        ca_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
      bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
    relabel_configs:
    - source_labels:
      - __meta_kubernetes_pod_name
      target_label: container_label_io_kubernetes_pod_name
    - source_labels:
      - __meta_kubernetes_namespace
      target_label: container_label_io_kubernetes_pod_namespace
    - source_labels:
      - __meta_kubernetes_pod_container_port_number
      target_label: container_label_port

Events:  <none>
```

Deploy Prometheus and other services

```
fbchan@ck8s-1:~/monitoring$ kubectl apply -f 4_prometheus-deploy.yml
deployment.apps/prometheus created


fbchan@ck8s-1:~/monitoring$ kubectl apply -f 5_prometheus-service.yml
service/prometheus created
```

```
fbchan@ck8s-1:~/monitoring$ kubectl apply -f 6_grafana.yml
deployment.apps/grafana created

fbchan@ck8s-1:~/monitoring$ kubectl apply -f 7_node-exporter.yml
daemonset.apps/node-exporter created

fbchan@ck8s-1:~/monitoring$ kubectl apply -f 8_state-metrics-deploy.yml
deployment.apps/kube-state-metrics created

fbchan@ck8s-1:~/monitoring$ kubectl apply -f 9_state-metrics-rbac.yml
serviceaccount/kube-state-metrics created
role.rbac.authorization.k8s.io/kube-state-metrics created
rolebinding.rbac.authorization.k8s.io/kube-state-metrics created
clusterrole.rbac.authorization.k8s.io/kube-state-metrics created
clusterrolebinding.rbac.authorization.k8s.io/kube-state-metrics created


fbchan@ck8s-1:~/monitoring$ kubectl -n monitoring get pod
NAME                                  READY   STATUS    RESTARTS   AGE
grafana-78f595d5d-72zqx               1/1     Running   0          13s
kube-state-metrics-95bcfcbd4-5wwpk    1/1     Running   0          3m15s
node-exporter-fhm8g                   1/1     Running   0          3m31s
node-exporter-jm9gs                   1/1     Running   0          3m31s
prometheus-765b875d7c-5zpmd           1/1     Running   0          3m58s
```

9.1 Install Helm and Tiller

Helm is an application package manager for Kubernetes. Helm is required to install Aspen Mesh. Another companion server component, tiller, that runs on Kubernetes cluster, listens for command from helm and handles the configuration and deployment of software releases on the cluster.

Note: Helm 3 is not supported for ISTIO releases as documented by Istio. Hence, ensure you install Helm 2.x

Download and install helm and tiller

```
fbchan@ck8s-1:~$ curl https://raw.githubusercontent.com/kubernetes/helm/master/scripts/get |
bash
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  7164  100  7164    0     0  25677      0 --:--:-- --:--:-- --:--:-- 25769
Downloading https://get.helm.sh/helm-v2.16.1-linux-amd64.tar.gz
Preparing to install helm and tiller into /usr/local/bin
helm installed into /usr/local/bin/helm
tiller installed into /usr/local/bin/tiller
Run 'helm init' to configure helm.
fbchan@ck8s-1:~$ helm init
Creating /home/fbchan/.helm
Creating /home/fbchan/.helm/repository
Creating /home/fbchan/.helm/repository/cache
Creating /home/fbchan/.helm/repository/local
Creating /home/fbchan/.helm/plugins
Creating /home/fbchan/.helm/starters
Creating /home/fbchan/.helm/cache/archive
Creating /home/fbchan/.helm/repository/repositories.yaml
Adding stable repo with URL: https://kubernetes-charts.storage.googleapis.com
Adding local repo with URL: http://127.0.0.1:8879/charts
$HELM_HOME has been configured at /home/fbchan/.helm.

Tiller (the Helm server-side component) has been installed into your Kubernetes Cluster.

Please note: by default, Tiller is deployed with an insecure 'allow unauthenticated users'
policy.
To prevent this, run `helm init` with the --tiller-tls-verify flag.
For more information on securing your installation see:
https://docs.helm.sh/using_helm/#securing-your-helm-installation
fbchan@ck8s-1:~$
```

Create service account and assign roles

Note: Ensure appropriate roles are assigned. For simplicity and proof-of-concept, cluster-admin is being assigned.

```
fbchan@ck8s-1:~$ kubectl create serviceaccount -n kube-system tiller
serviceaccount/tiller created

fbchan@ck8s-1:~$ kubectl create clusterrolebinding tiller-cluster-admin --clusterrole=cluster-
admin --serviceaccount=kube-system:tiller
clusterrolebinding.rbac.authorization.k8s.io/tiller-cluster-admin created
```

Update deployment for tiller

```
fbchan@logos:~/k8s-clusterX/gke-1$ kubectl --namespace kube-system patch deploy tiller-deploy -p
'{"spec":{"template":{"spec":{"serviceAccount":"tiller"}}}}'
deployment.extensions/tiller-deploy patched
```

```
fbchan@ck8s-1:~$ kubectl --namespace kube-system get deploy tiller-deploy -o yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "2"
  creationTimestamp: "2019-11-25T01:44:26Z"
  generation: 2
  labels:
    app: helm
    name: tiller
  name: tiller-deploy
  namespace: kube-system
  resourceVersion: "105426"
  selfLink: /apis/apps/v1/namespaces/kube-system/deployments/tiller-deploy
  uid: ab6822dc-dffa-46ed-adbb-5c9b161bbb39

...
...
...
      restartPolicy: Always
      schedulerName: default-scheduler
      securityContext: {}
      serviceAccount: tiller
      serviceAccountName: tiller
      terminationGracePeriodSeconds: 30
status:
  availableReplicas: 1
...
...
...
```

10 Lab Cluster environment information

Cluster 1


```
fbchan@ck8s-1:~$ kubectl get node -o wide
NAME     STATUS   ROLES    AGE     VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION      CONTAINER-RUNTIME
ck8s-1   Ready    master   2d18h   v1.16.3   10.1.1.61     <none>        Ubuntu 18.04.3 LTS   4.15.0-72-generic   docker://18.6.2
ck8s-2   Ready    <none>   2d18h   v1.16.3   10.1.1.62     <none>        Ubuntu 18.04.3 LTS   4.15.0-72-generic   docker://18.6.2
ck8s-3   Ready    <none>   2d18h   v1.16.3   10.1.1.63     <none>        Ubuntu 18.04.3 LTS   4.15.0-72-generic   docker://18.6.2
fbchan@ck8s-1:~$
```

Cluster 2


```
fbchan@k8s-1:~$ kubectl get node -o wide
NAME    STATUS   ROLES    AGE    VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION      CONTAINER-RUNTIME
k8s-1   Ready    master   205d   v1.14.2   10.4.0.81     <none>        Ubuntu 16.04.6 LTS   4.4.0-148-generic   docker://18.9.2
k8s-2   Ready    <none>   205d   v1.14.2   10.4.0.82     <none>        Ubuntu 16.04.6 LTS   4.4.0-148-generic   docker://18.9.2
k8s-3   Ready    <none>   205d   v1.14.2   10.4.0.83     <none>        Ubuntu 16.04.6 LTS   4.4.0-148-generic   docker://18.9.2
fbchan@k8s-1:~$
```

Cluster 3


```
fbchan@logos:~/k8s-clusterX/gke-1$ kubectl get node -o wide
NAME                                                STATUS   ROLES    AGE   VERSION          INTERNAL-IP   EXTERNAL-IP      OS-IMAGE                             KERNEL-VER
SION   CONTAINER-RUNTIME
gke-gke-dev-edge1-default-pool-244b5d43-4gd9   Ready    <none>   15h   v1.15.4-gke.22   10.12.0.17    35.198.204.46    Container-Optimized OS from Google   4.19.76+
       docker://19.3.1
gke-gke-dev-edge1-default-pool-244b5d43-z3qk   Ready    <none>   15h   v1.15.4-gke.22   10.12.0.18    35.198.215.205   Container-Optimized OS from Google   4.19.76+
       docker://19.3.1
gke-gke-dev-edge1-default-pool-244b5d43-zp1s   Ready    <none>   15h   v1.15.4-gke.22   10.12.0.16    34.87.84.1       Container-Optimized OS from Google   4.19.76+
       docker://19.3.1
fbchan@logos:~/k8s-clusterX/gke-1$
```