

Doing More with Service Mesh

As you add more services to your Kubernetes cluster, operational complexity increases. In modern polyglot distributed application environments, this complexity increases even further. There is an overhead involved in maintaining versions of the same operational capabilities (security, observability, robustness, policy) in different languages for each of your services.

What if there was an approach that could decouple these operational and policy capabilities from our code base? Enter the service mesh!

Service Mesh

Linkerd and Istio are two of the most popular service meshes.

"Linkerd moves visibility, reliability, and security constraints down to the infrastructure layer, out of the application layer."

- William Morgan, CEO, Buoyant

"Istio decouples operational aspects of the services from the implementation of the services."

- Eric Brewer, VP Infrastructure & Google Fellow, Google Cloud

A service mesh typically provides the following set of capabilities:

- **Traffic Management**
 - Protocol Support – http/s, grpc, tcp
 - Dynamic Routing – conditional, weighting, mirroring
 - Resiliency – timeouts, retries, circuit breakers
 - Policy – access control, rate limits, quotas
 - Testing - fault injection
- **Security**
 - Encryption – mTLS, certificate management
 - Strong Identity – SPIFFE or similar
 - Auth – authentication, authorisation
- **Observability**
 - Metrics – golden metrics
 - Tracing - traces across workloads

Challenge

Your team's goal in this challenge is to deploy a service mesh into your AKS cluster. The service mesh will augment and expand the capabilities you have already explored and added to your Trip Insights services as part of the previous challenges. None of the aspects of this challenge will require changes to your currently deployed code.

First, your team must build on the security foundations laid in previous challenges. The team must enable mutual TLS for all intra-cluster communication within the Trip Insights platform. Ensure that you also understand how you can verify that your traffic is now secure.

Next, your team must expand on the observability we introduced in a previous challenge. Service meshes provide you with the capability to generate additional application level metrics. The team must show latency, traffic, and error metrics in a dashboard to provide additional insight into the operational health of the Trip Insights services.

Finally, your team must leverage the service mesh to make your services more robust. Intermittent network or service errors can be mitigated by using retry and timeout policies. Ensure that your team has configured these appropriately for each service.

Your team has been offered the following guidance. Istio is a feature rich, customisable and extensible service mesh, and can be more difficult to configure and get running successfully. Linkerd is an easy to use and lightweight service mesh, and makes it very easy to get up and running quickly.

Success Criteria

- **Your team** must have successfully deployed a service mesh into your Kubernetes cluster. Share your understanding of each of the relevant components of the service mesh and what capability they support.
- **Your team** has secured all intra-cluster communication with mutual TLS. Demonstrate to your coach how you can verify that secure communication has been established.
- **Your team** must show a dashboard to your coach that provides the following 3 metrics for the TriplInsights services. These 3 metrics are a subset of the four golden signals:
 - Latency - The time it takes to service a request. Show latency for P50, P95 and P99.
 - Traffic - A measure of how much demand is being placed on the service. Show http requests per second.
 - Errors - The rate of requests that fail. This can also be expressed inversely as the success rate. Show errors per second.
- **Your team** has added retry and timeout policies to the TriplInsights services. Demonstrate these in action to your coach.
- **Your team** must demonstrate your cluster is overall "Healthy"

References

Service Meshes

- [About Service Meshes \(https://docs.microsoft.com/en-us/azure/aks/servicemesh-about\)](https://docs.microsoft.com/en-us/azure/aks/servicemesh-about).

Linkerd

- [Linkerd - Getting Started \(https://linkerd.io/2/getting-started/\)](https://linkerd.io/2/getting-started/)
- [Linkerd - Overview \(https://docs.microsoft.com/en-us/azure/aks/servicemesh-linkerd-about\)](https://docs.microsoft.com/en-us/azure/aks/servicemesh-linkerd-about).
- [Install Linkerd in Azure Kubernetes Service \(AKS\) \(https://docs.microsoft.com/en-us/azure/aks/servicemesh-linkerd-install\)](https://docs.microsoft.com/en-us/azure/aks/servicemesh-linkerd-install).

Istio

- [What is Istio? \(https://istio.io/docs/concepts/what-is-istio/\)](https://istio.io/docs/concepts/what-is-istio/)
- [Istio - Overview \(https://docs.microsoft.com/en-us/azure/aks/servicemesh-istio-about\)](https://docs.microsoft.com/en-us/azure/aks/servicemesh-istio-about).
- [Install and use Istio in Azure Kubernetes Service \(AKS\) \(https://docs.microsoft.com/en-us/azure/aks/servicemesh-istio-install\)](https://docs.microsoft.com/en-us/azure/aks/servicemesh-istio-install).

Distributed Systems

- [The Four Golden Signals \(https://landing.google.com/sre/sre-book/chapters/monitoring-distributed-systems/\)](https://landing.google.com/sre/sre-book/chapters/monitoring-distributed-systems/).