

Getting Started and Beyond: Istio Multicluster with GitOps

Ryota Sawada / @rytswd / CTO at UPSIDER, Inc.



#IstioCon

Welcome to the IstioCon 2021

IstioCon 2021 is the inaugural community conference for the industry's most popular service mesh. IstioCon is a community-led event, showcasing the lessons learned from running Istio in production, hands-on experiences from the Istio community, and featuring maintainers from across the Istio ecosystem. The conference offers a mix of keynotes, technical talks, lightning talks, workshops and roadmap sessions. Fun and games are also included with two social hours to take the load off and mesh with the Istio community, vendors, and maintainers!

#IstioCon



Agenda

- Introduction
- Target Audience
- Topics covered / NOT covered
- About Istio Multicloud
- Brief brush up on Istio resources
- First Demo
- About GitOps
- Second Demo
- What's Next?



Who is Ryota?

- Find me at usual places with @rytswd (Ryota Sawada)
- CTO at UPSIDER, Inc., leading Platform team
- Worked in finance space for about a decade

- Based in London
- Have 2 cats



Briefly about UPSIDER



- Startup providing B2B payment service
- Uses Istio extensively in production
- Headquarter in Tokyo, Japan
- Remote team around the world

We are hiring.



#IstioCon

@rytswd



Materials

This slide will be also made available later.

You can find all the examples at:

<https://github.com/rytswd/get-istio-multicloud>

Ryota's GitHub profile page. The profile picture shows a blue plush toy (Totoro) and a brown cat. The bio says "Hi there, I'm Ryota 🐱🐱" and "It's pronounced as REE-YO-TA 🎩". The highlights section includes "Arctic Code Vault Contributor" and "Makefile". The organizations section lists "UPSIDER". The contributions chart shows 6,516 contributions in the last year, with a color scale from light green (Less) to dark green (More). Other pinned repositories include "get-istio-multicloud" and "comment".



Target audience



If you want to ...

- play with Istio offerings
- see the actual configuration files
- understand how Istio can be used / installed with other services
- know what multicloud challenges and solutions there are



This is probably not for you if you...

- already have Istio deployed in multicloud environment



What is Getting Started and Beyond?

Getting Started

- Install into some cluster
- Tweak some simple configuration
- Add / remove offerings you are keen to test and see in action
- Add custom configuration on top of simple setup

Proof of Concept

- Install in cluster with other business applications, where there is some specific limitation, security requirements, etc.
- Configure business applications to confirm it provides what business requires
- Break, debug, pinpoint, and fix

Production onboarding

- Security
 - Observability
 - Supportability
- etc.



Goal of today

Getting Started

- Install into some cluster
- Tweak some simple configuration
- Add / remove offerings you are keen to test and see in action
- Add custom configuration on top of simple setup

Proof of Concept

- Install in cluster with other business applications, where there is some specific limitation, security requirements, etc.
- Configure business applications to confirm it provides what business requires
- Break, debug, pinpoint, and fix

With requirements such as:

- Multicluster
- GitOps integration

These are not just specific requirements. They help you understand Istio offerings more in detail with ready-to-be-used configuration files.



What is covered in the talk

- Istio multicluster installation and setup
- All configurations in files
- Traffic management offerings in action
- GitOps and declarative setup
- Multicluster challenges



What is NOT covered in the talk

- Multiclutser observability challenges
- Security considerations
- Secret management
- GitOps implementation details



Istio installation patterns

As of Feb 2021, with Istio v1.9,
there are 5 mentioned in doc:

- istioctl
- IstioOperator
- Helm
- Multicluster
- Virtual Machine

The screenshot shows the Istio 1.9 documentation homepage. The top navigation bar includes links for Docs, Blog, News, FAQ, About, and a search icon. The main content area has a dark background with white text. On the left is a sidebar with a sailboat logo and sections for Concepts, Setup, Getting Started (with sub-links for Platform Setup, Install, Upgrade, and More Guides), Tasks, Examples, Operations, and Reference. The main title "Installation Guides" is centered above three sections: "Install with Istioctl", "Install with Helm", and "Virtual Machine Installation". Below these is a note: "Choose the guide that best suits your needs and platform." At the bottom, a callout box states: "Istio 1.9 has been tested with these Kubernetes releases: 1.17, 1.18, 1.19, 1.20." The footer contains a copyright notice and a small sailboat icon.

Docs Blog News FAQ About

Installation Guides

Choose the guide that best suits your needs and platform.

Install with Istioctl
Install and customize any Istio configuration profile for in-depth evaluation or production use.

Install with Helm
Install and configure Istio for in-depth evaluation.

Virtual Machine Installation
Deploy Istio and connect a workload running within a virtual machine to it.

If you need help choosing, refer to our [which Istio installation method should I use?](#) FAQ page.

Istio 1.9 has been tested with these Kubernetes releases: 1.17, 1.18, 1.19, 1.20.

Istio installation patterns (cont'd)

In this talk, we will be looking at:

- IstioOperator
- Manifest generation

The screenshot shows the Istio 1.9 documentation website. The top navigation bar includes links for Docs, Blog, News, FAQ, About, and a search icon. The main content area has a title 'Generate a manifest before installation'. It explains that you can generate a manifest before installing Istio using the `manifest generate` sub-command. An example command is shown in a code block: `$ istioctl manifest generate > $HOME/generated-manifest.yaml`. Below this, it states that the generated manifest can be used to inspect installed resources and track changes over time. It also notes that the output from `manifest generate` can be used with `kubectl apply`, but this may not always reflect the actual installed resources due to sequencing differences.

In this section, we will learn how to generate a manifest before installing Istio using the `manifest generate` sub-command. This allows us to inspect what exactly is installed as well as to track changes to the manifest over time. While the `IstioOperator` CR represents the full user configuration and is sufficient for tracking it, the output from `manifest generate` also captures possible changes in the underlying charts and therefore can be used to track the actual installed resources.

The output from `manifest generate` can also be used to install Istio using `kubectl apply` or equivalent. However, these alternative installation methods may not apply the resources with the same sequencing of dependencies as `istioctl install` and are not tested in an Istio release.



Understanding Istio setup

- Easy to get started with istioctl, but so much happening behind the scenes
 - Many moving parts with the installation
 - So much offering out of the box
-
- Even more complicated when there are multiple clusters



What is Multicloud? Why?

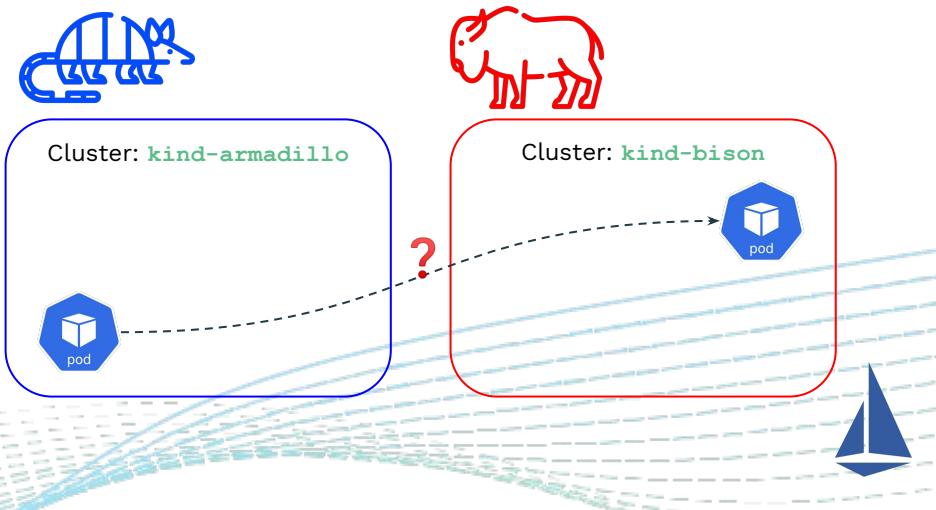
Simply put, it is about combining multiple Kubernetes clusters.

For example, UPSIDER currently has 4 clusters with Istio multicloud setup.

Benefits are:

- Separate cluster management for better control and audit
- Redundancy and High Availability

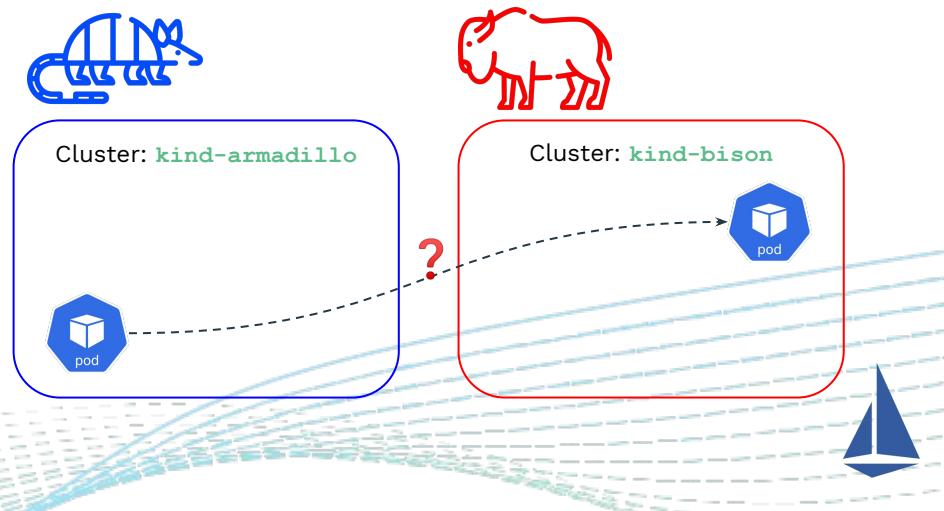
But it does (did) pose challenges...



Challenge of Multicluster

- Multicluster requires configuring multiple clusters (duh 😐)
- Configurations are similar, but slightly different
- Each cluster needs to securely talk to other clusters
- Managing those differences in repeatable manner can be surprisingly challenging

But solving this help you understand how Istio works and its offerings in detail.



Challenge of Multicloud (cont'd)

And, multicloud is not just one approach; in fact, there are 4 categories:

Within the same network,

- Primary cluster + Primary cluster
- Primary cluster + Remote cluster

On different networks,

- Primary cluster + Primary cluster
- Primary cluster + Remote cluster

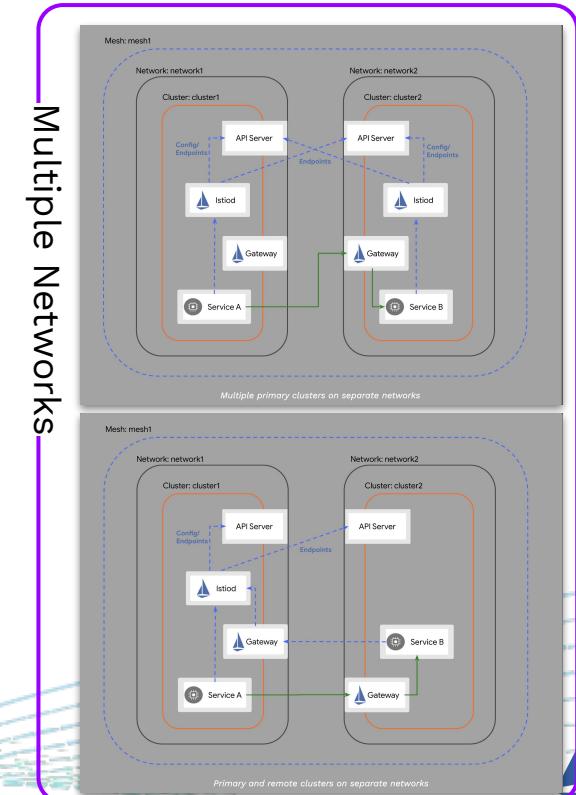
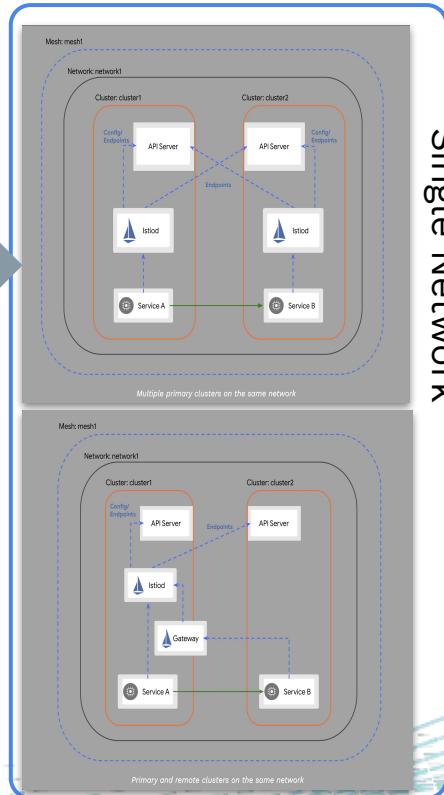
The screenshot shows the Istio 1.9 documentation website. The top navigation bar includes links for Docs, Blog, News, FAQ, About, and a search icon. The main page title is "Install Multicloud". The left sidebar has a "Concepts" section, followed by a "Setup" section with "Getting Started" and "Install" sub-sections. The "Install" section contains links for "Install with Istioctl", "Istio Operator", "Install", "Install with Helm", and "Install Multicloud" (which is highlighted in orange). Below these are "Upgrade" and "More Guides" sections. The main content area starts with a heading "Install Multicloud" and a sub-heading "Install an Istio mesh across multiple Kubernetes clusters.". It lists several steps: "Before you begin", "Install Multi-Primary", "Install Primary-Remote", "Install Multi-Primary on different networks", "Install Primary-Remote on different networks", and "Verify the installation". A note below says "Follow this guide to install an Istio service mesh that spans multiple clusters.". Another note states "This guide covers some of the most common concerns when creating a multicloud mesh:". At the bottom, there are two bullet points: "Network topologies: one or two networks" and "Control plane topologies: multiple primary clusters, a primary and remote cluster". A callout box at the bottom right says "For meshes that span more than two clusters, you can extend the steps in this guide to configure more complex topologies." and "See deployment models for more information."

Challenge of Multicluster (cont'd)

For the following demos,
we will be using:

- Single Network
- Primary + Primary

NOTE: we will be taking
slightly different approach
from Istio official
documentation today.



Single Network

Multiple Networks

Brush up on Istio resources



Control Plane

Manages data plane components
Handles Custom Resources



Istio Operator

Manages Istio installation with
IstioOperator Custom Resource



Data Plane

Handles actual traffic
Can be standalone or sidecar



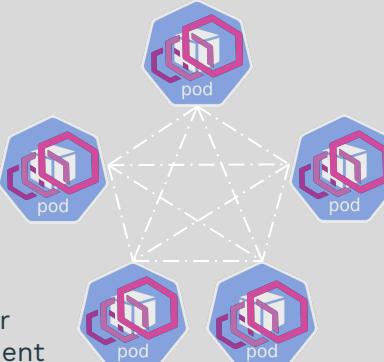
Istio Ingress Gateway
Istio Egress Gateway



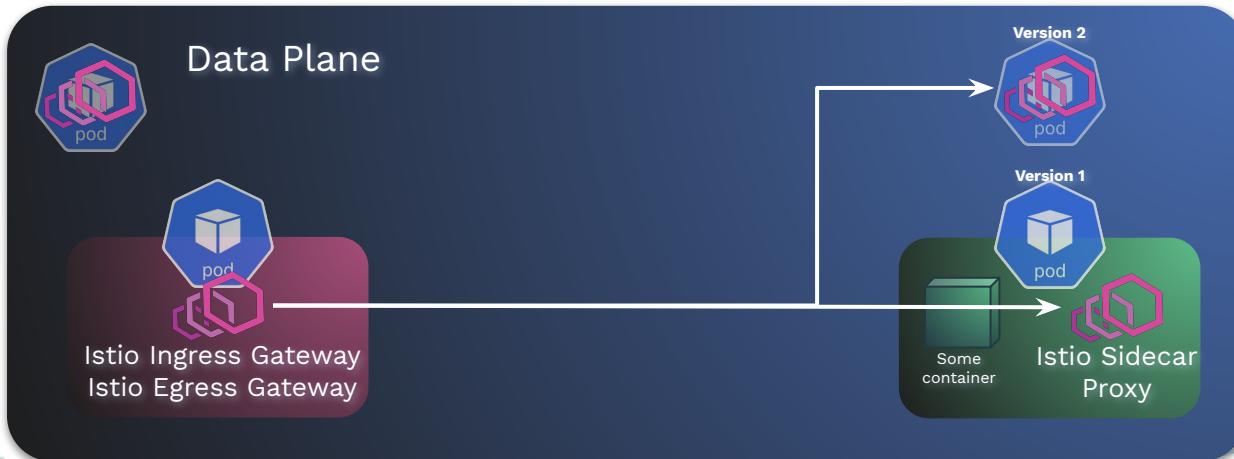
Istio Sidecar
Proxy

Example

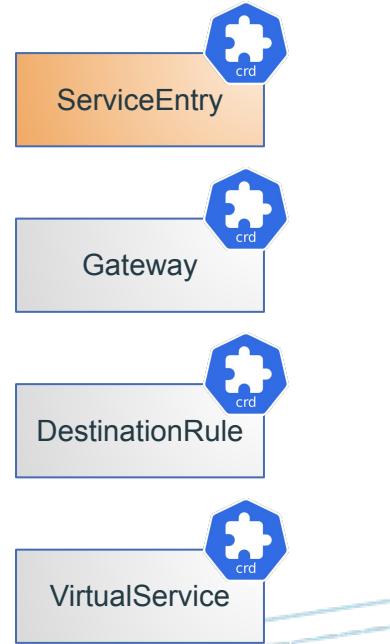
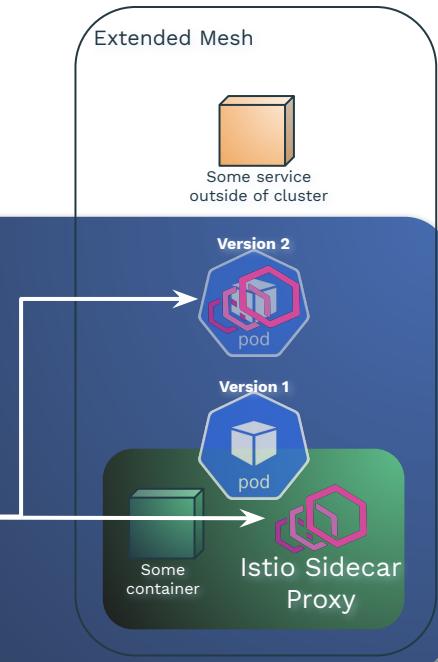
- Data Plane with 5 proxies
- Each pod knows endpoint details of other pods
- Can be Sidecar or Gateway component



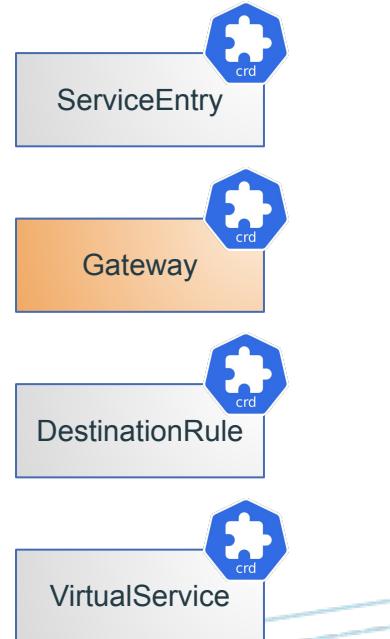
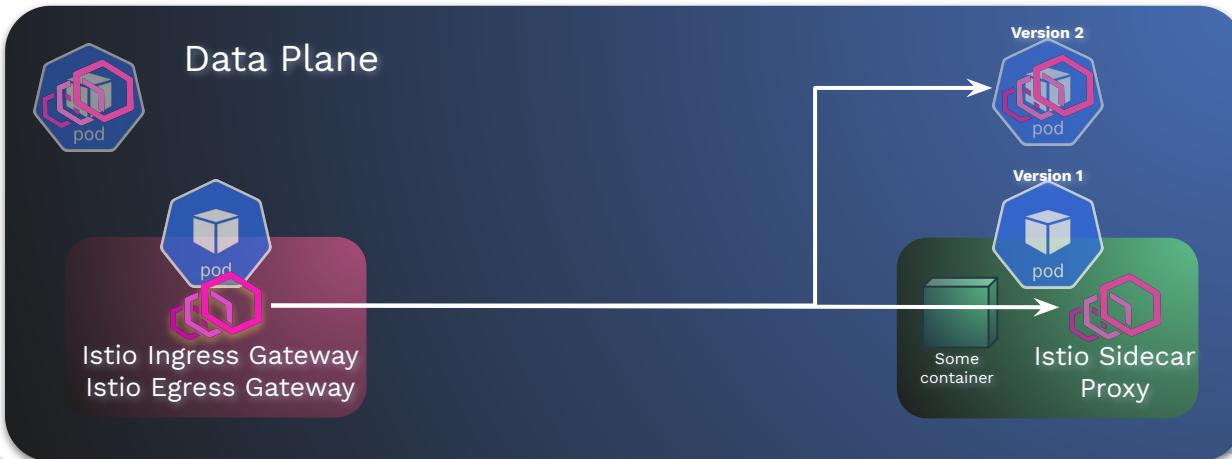
Brush up on Istio resources (cont'd)



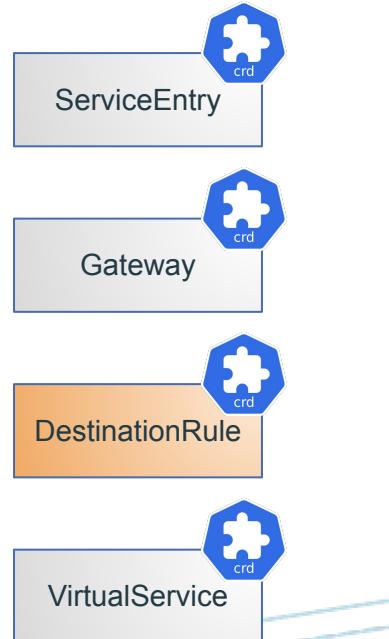
Brush up on Istio resources (cont'd)



Brush up on Istio resources (cont'd)



Brush up on Istio resources (cont'd)



Brush up on Istio resources (cont'd)



Before diving into demo...

This demo uses following tools / versions



KinD

<https://github.com/kubernetes-sigs/kind>

Version v0.10.0

Kubernetes v1.17.17



K9s

<https://github.com/derailed/k9s>



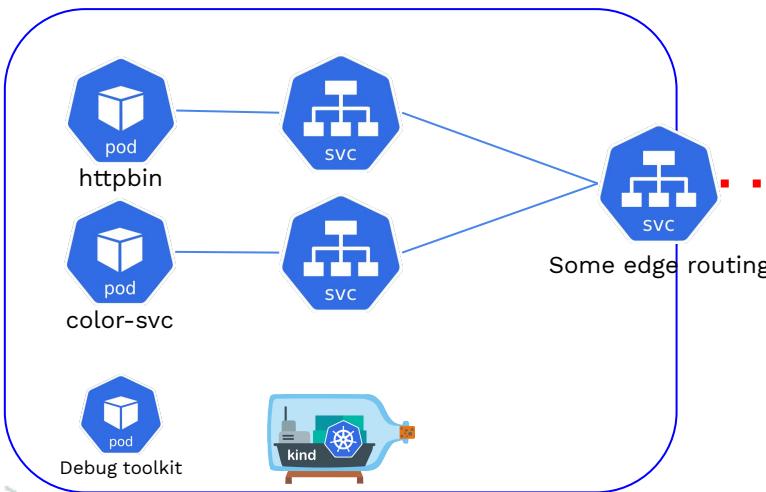
NOTE: This demo is based on Istio Version 1.7.5.



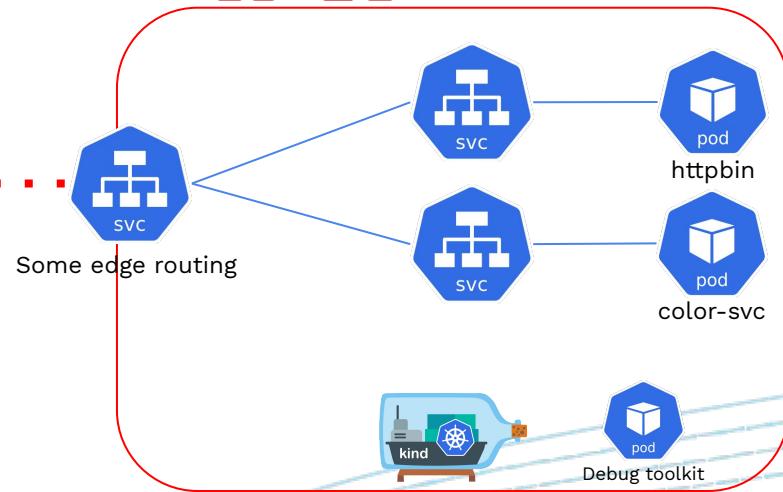
Goal of demo



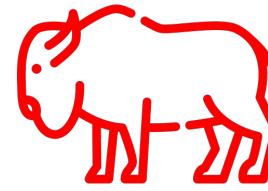
Cluster: kind-armadillo



Cluster: kind-bison



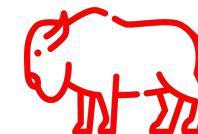
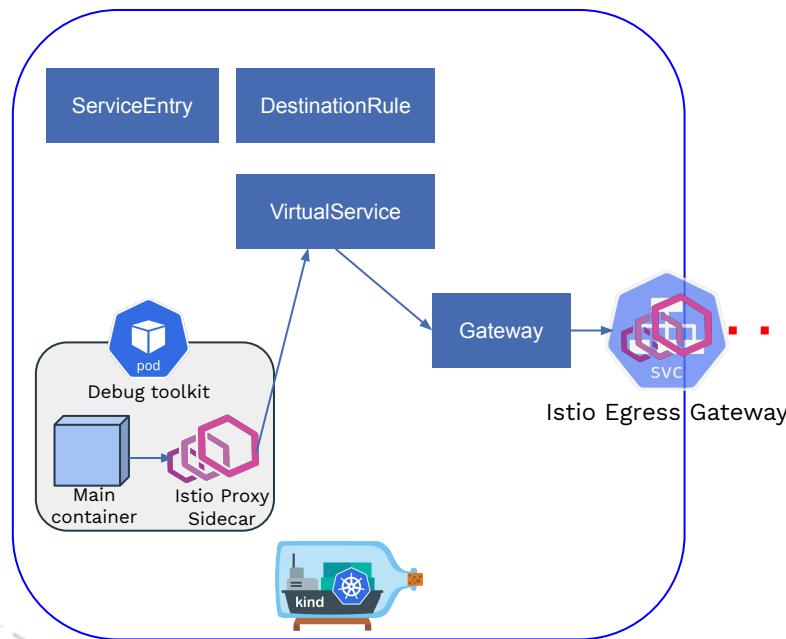
Demo Time!



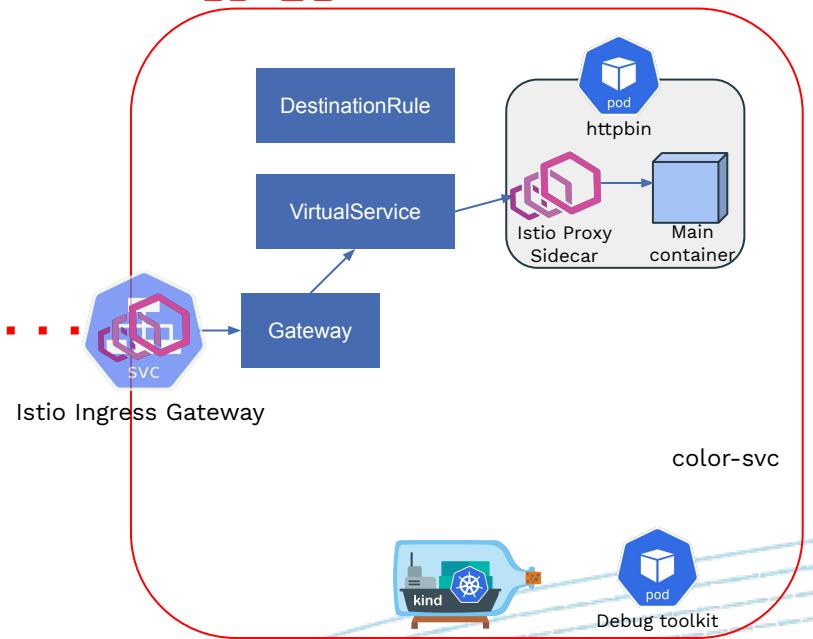
Summary of demo



Cluster: kind-armadillo



Cluster: kind-bison



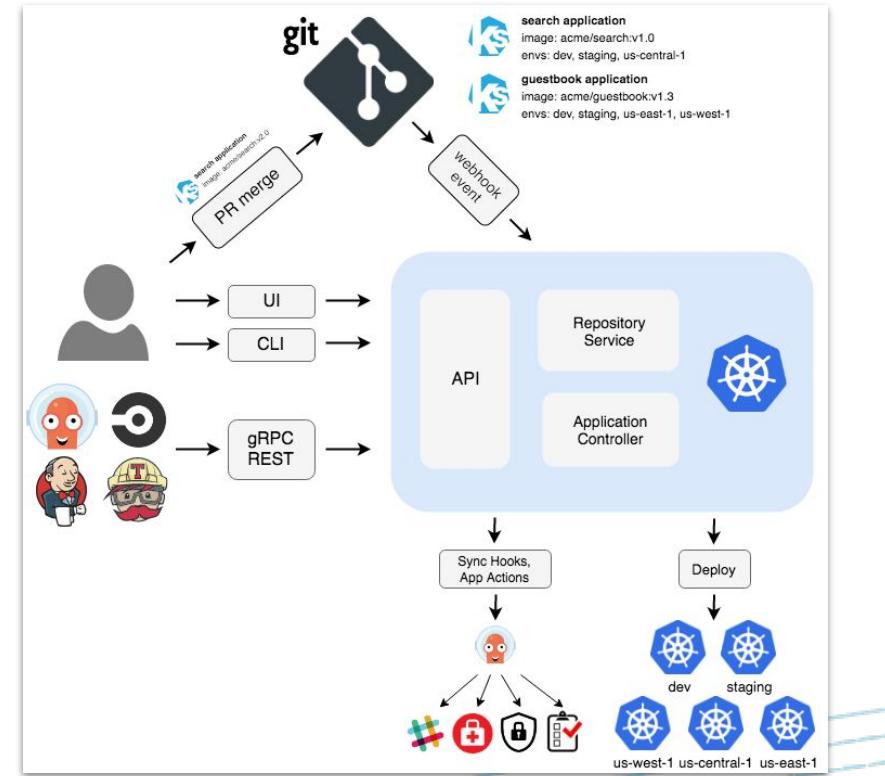
What is GitOps?

- Declarative cluster management
- All changes are driven by Git repo

This demo uses Argo CD.

Benefits are:

- Git repo becomes source of truth
- Clear change history
- Pull Request approval flow
- Non-Git based changes get reverted
- Easy to take down, recreate, and/or replicate cluster(s)



Before diving into demo...

This demo uses following tools / versions



KinD

<https://github.com/kubernetes-sigs/kind>
Version v0.10.0
Kubernetes v1.17.17



Prometheus

<https://github.com/prometheus/prometheus>
Prometheus Operator Version v0.45.0



K9s

<https://github.com/derailed/k9s>



Kiali

<https://github.com/kiali/kiali>
Kiali Operator Version v1.29.0



Argo CD

<https://github.com/argoproj/argo-cd>
Version v1.8.2



Grafana

<https://github.com/grafana/grafana>
Version v7.4.0, Helm Chart Version v6.3.0



Jaeger

<https://github.com/uber/jaegertracing/jaeger>
Jaeger Operator Version v1.21.3

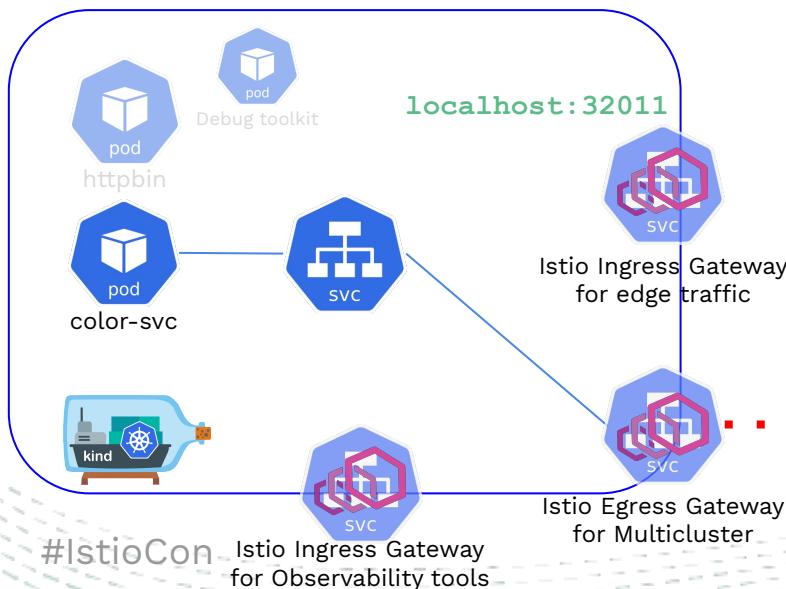
NOTE: This demo is based on Istio Version 1.7.5.



Goal of demo



Cluster: kind-armadillo



Goal of demo

Color Grid for Demo

Base Address Number of Cells Submit

Endpoint Submit Endpoint Submit

Periodic Query Periodic Query

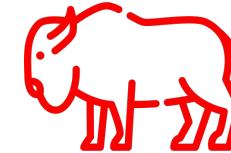
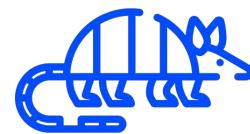
Unset	Unset	Unset	Unset
Unset	Unset	Unset	Unset
Unset	Unset	Unset	Unset



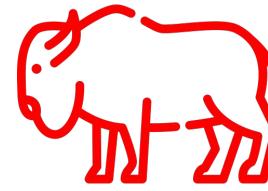
color-svc

<https://github.com/rytswd/color-svc>

```
$ curl 'http://localhost:8800/random'  
  
# Output  
Generated Color  
"Green" - with HEX "#008000"
```



Demo Time!



Summary of Demo

- Istio's powerful traffic management can pose some configuration challenges
- Even for Getting Started, multicloud setup allows you to understand Istio configuration options
- Having declarative setup such as GitOps helps cluster management, and also monitor Istio resources

Color Grid for Demo

Base Address `http://localhost:32011` Number of Cells 6 Submit

Endpoint `/chaos?fmt=json` Submit

Periodic Query

BlueViolet	Navy
Aqua	Teal
Blue	Navy
Fuchsia	Fuchsia
Fuchsia	Maroon
Fuchsia	Fuchsia

Endpoint `/bison?fmt=json` Submit

Periodic Query



What's the next step from here?

Some interesting talks at IstioCon 2021

- What Envoy Hears When Istio Speaks
by Rob Salmond (Monday)
- Istio Multicluster Workshop
by Denis Jannot & Christian Posta (Tuesday)
- Taming Istio Configuration with Helm
by Ryan Michela (Wednesday)
- Know your peers
by Alex Van Boxel (Wednesday)



What's the next step from here? (Cont'd)

The screenshot shows a blog post titled "Understanding Envoy Proxy HTTP Access Logs" on the Ambassador website. The post discusses the default access log format used by Envoy. A sample log entry is shown with various fields labeled:

- START_TIME**: [2019-08-06T16:30:11.746Z]
- METHOD**: GET
- X-ENVOY-ORIGINAL-PATH**: /backend/debug/
- PROTOCOL**: HTTP/1.1
- RESPONSE_CODE**: 200
- BYTES_RECEIVED**: 0
- BYTES_SENT**: 1124
- DURATION(ms)**: 0
- REQUEST-ID**: a70851b76b86511e9b8c60ebd9abcaa2-807989241.us-east-1.elb.amazonaws.com
- HOST**: 10.100.216.61:8080
- USER-AGENT**: curl/7.63.0
- UPSTREAM_HOST**: 192.168.28.113
- X-FORWARDED-FOR**: 10.52.0.22:8080

[Link](#)

#IstioCon

@rytswd

Megan O'Keefe (@askmeegs) shared a tweet about Istio proxy logs. The tweet includes a hand-drawn diagram of a log entry with various fields labeled:

- user-agent**: curl/7.47.0
- start time**: 2019-08-01T17:45:16.683Z
- Request ID**: 4344078e-59cf-4303-bd60-1a7b77982e3d
- Protocol**: HTTP/1.1
- Method**: GET
- Response Flags**: 200 - 0 13 3 0
- Response Code**: 200
- Duration**: 16.683 ms
- Bytes sent (ms)**: 13 ms
- Bytes received**: 3 bytes
- Upstream svc. time**: 0 ms
- x-forwarded-for**: 10.52.0.22:8080
- downstream local address**: 10.52.1.18:54982
- upstream cluster**: inbound|80||auth.default.svc.cluster.local|
- upstream host**: 127.0.0.1:8080
- upstream local address**: 10.52.0.22:8080
- Request Authority**: auth.default.svc.cluster.local

9:43 PM · Aug 1, 2019

[Link](#)



Appendix: References

Repository used:

- <https://github.com/rytswd/get-istio-multicloud>
- <https://github.com/rytswd/docker-toolkit-images>
- <https://github.com/rytswd/color-svc>
- <https://github.com/rytswd/color-grid>

Other repositories referenced:

- <https://github.com/rytswd/get-gitops-k8s>

Link to this slide

- <https://tinyurl.com/istiocon-2021-gitops>



Thank you!

Ryota Sawada / @rytswd
CTO at UPSIDER, Inc.

#IstioCon

