

# 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 Multicluster
- 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

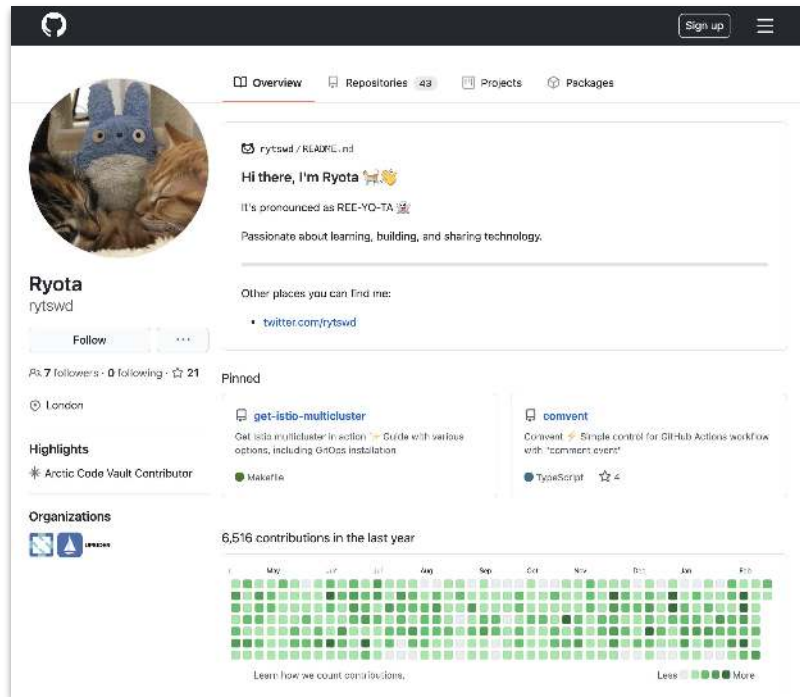


# Materials

This slide will be also made available later.

You can find all the examples at:

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



# 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 multicluster challenges and solutions there are

✗ This is probably not for you if you...

- already have Istio deployed in multicluster 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

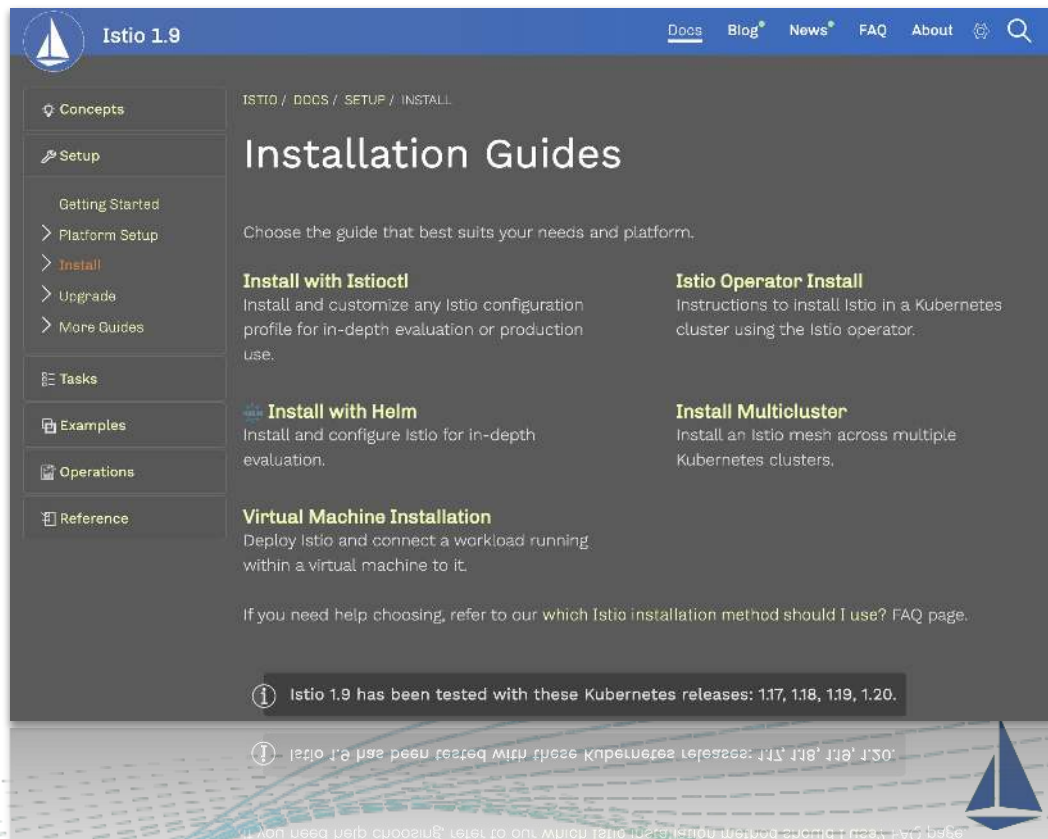
- Multicluster observability challenges
- Security considerations
- Secret management
- GitOps implementation details



# Istio installation patterns

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

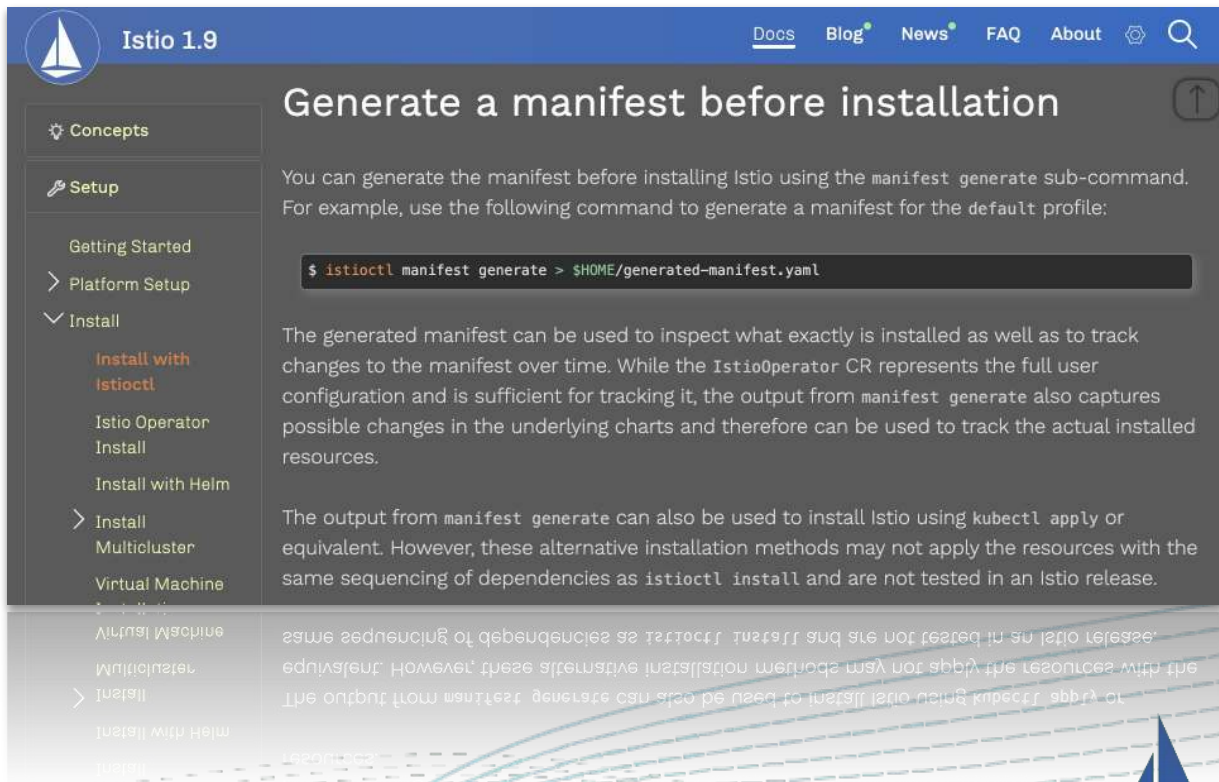
- istioctl
- IstioOperator
- Helm
- Multicluster
- Virtual Machine



# Istio installation patterns (cont'd)

In this talk, we will be looking at:

- IstioOperator
- Manifest generation



The screenshot shows the Istio 1.9 documentation page. The left sidebar contains a navigation menu with sections: Concepts, Setup, Getting Started, Platform Setup, Install, and Virtual Machine. The 'Install' section is expanded, showing 'Install with Istioctl', 'Istio Operator Install', 'Install with Helm', 'Install Multicloud', 'Virtual Machine', and 'Virtual Machine Multicloud'. The main content area is titled 'Generate a manifest before installation' and contains the following text:

You can generate the manifest before installing Istio using the `manifest generate` sub-command. For example, use the following command to generate a manifest for the default profile:

```
$ istioctl manifest generate > $HOME/generated-manifest.yaml
```

The generated manifest can be used 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.

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



# 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 Multicluster? Why?

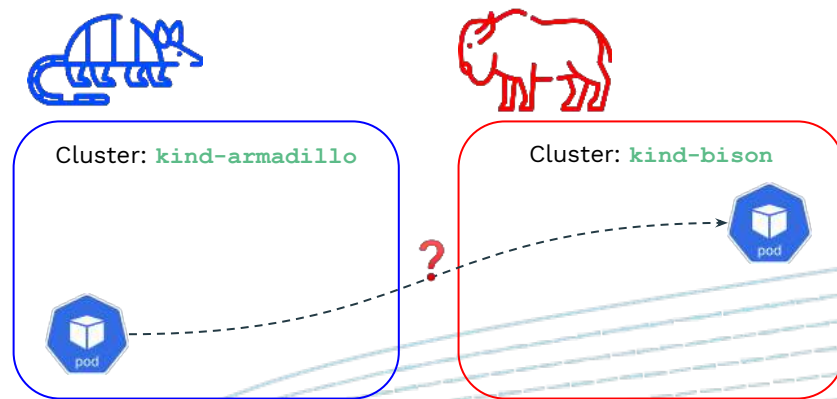
Simply put, it is about combining multiple Kubernetes clusters.

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

Benefits are:

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

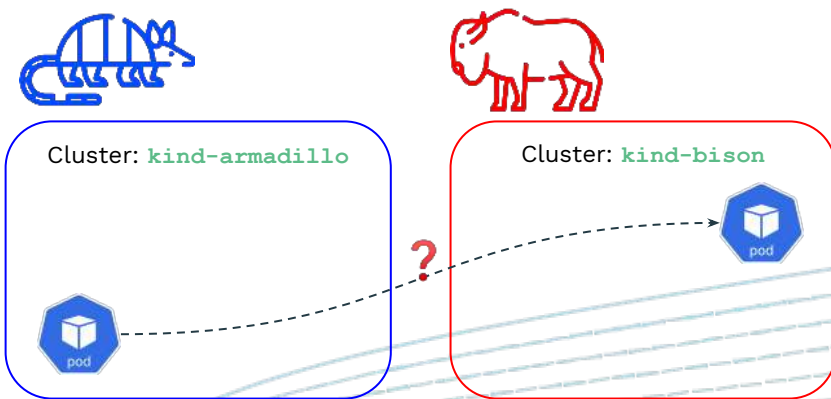
But it does (did) pose challenges...



# 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 Multicluster (cont'd)

And, multicluster 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

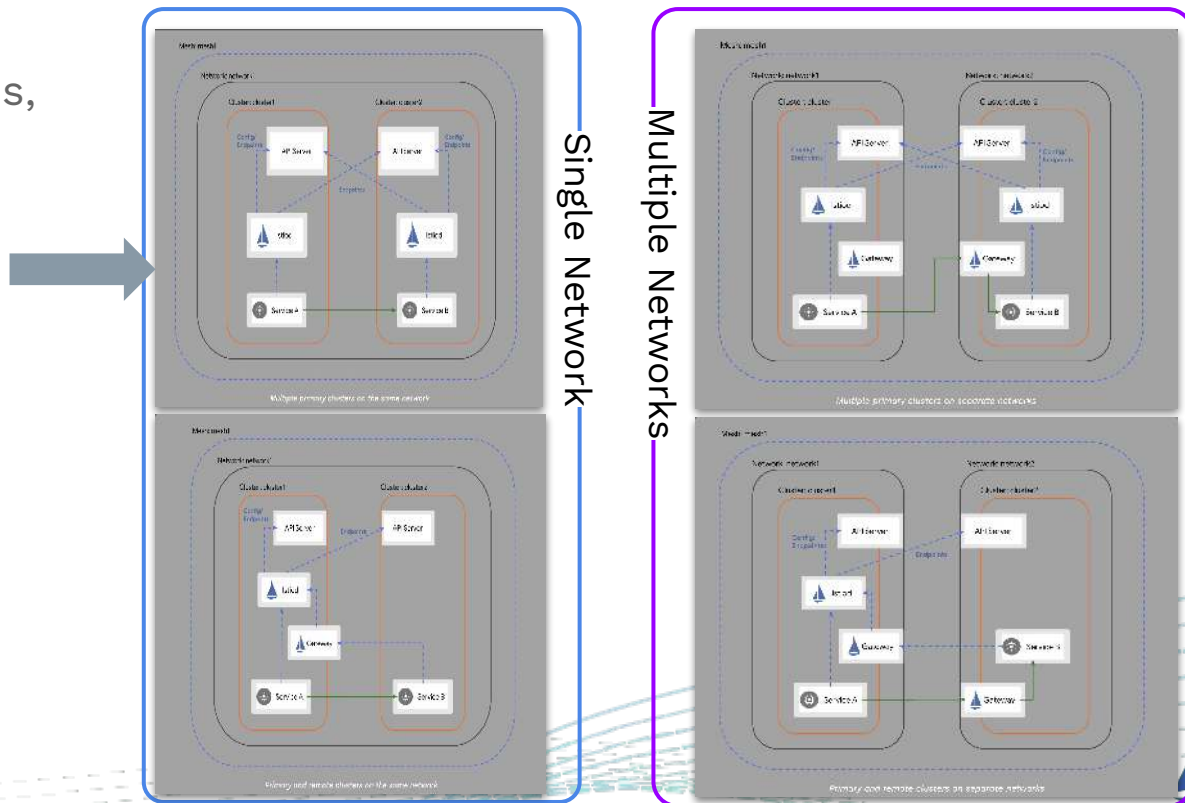


# Challenge of Multicluster (cont'd)

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.



# Brush up on Istio resources



## Control Plane

Manages data plane components  
Handles Custom Resources



## Istio Operator

Manages Istio installation with  
IstioOperator Custom Resource

istio-operator



## Data Plane

Handles actual traffic  
Can be standalone or sidecar



Istio Ingress Gateway  
Istio Egress Gateway



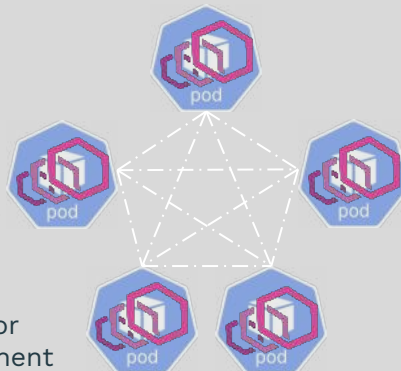
Istio Sidecar  
Proxy



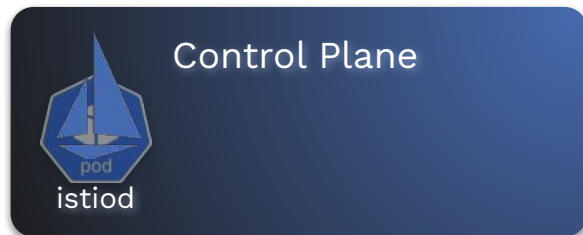
Other  
Container

## Example

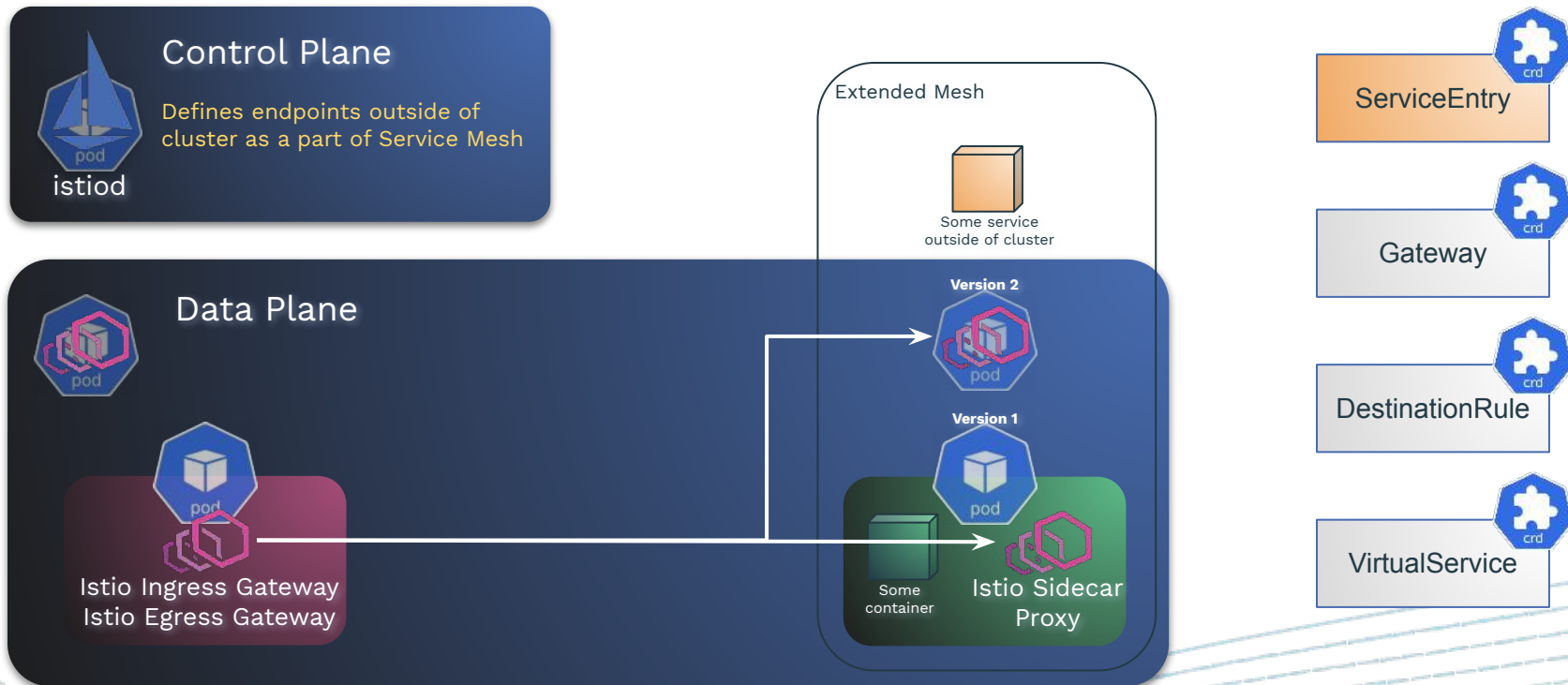
- Data Plane with 5 proxies
- Each pod knows endpoint details of other pods
- Can be Sidecar or Gateway component



# Brush up on Istio resources (cont'd)



# Brush up on Istio resources (cont'd)



# Brush up on Istio resources (cont'd)

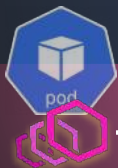


## Control Plane

Manage inbound and outbound traffic, allowing non-sidecar components to work with VirtualService setup



## Data Plane



Istio Ingress Gateway  
Istio Egress Gateway



Version 2



Version 1



Some container

Istio Sidecar Proxy



ServiceEntry



Gateway



DestinationRule



VirtualService





# Brush up on Istio resources (cont'd)



## Control Plane

Update traffic after routing has occurred, allowing more fine-grained load balancing, connection pool handling, etc.



## Data Plane



Istio Ingress Gateway  
Istio Egress Gateway



Some service  
outside of cluster



Version 2



Version 1



Some container

Istio Sidecar  
Proxy



ServiceEntry



Gateway



DestinationRule



VirtualService



# Brush up on Istio resources (cont'd)

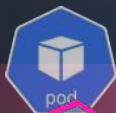


## Control Plane

Defines routing, and the details are propagated to all Proxies



## Data Plane



Istio Ingress Gateway  
Istio Egress Gateway



Version 2



Version 1



Some container

Istio Sidecar Proxy



ServiceEntry



Gateway



DestinationRule



VirtualService





# 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.

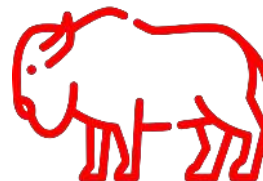
#IstioCon @rytswd



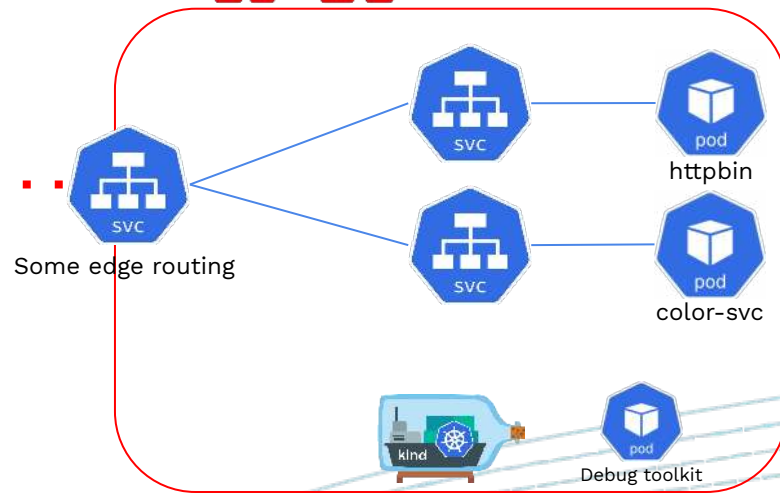
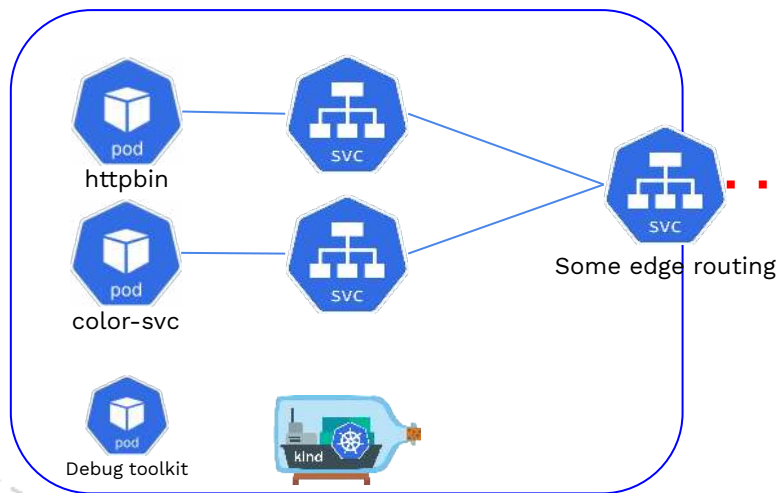
# Goal of demo



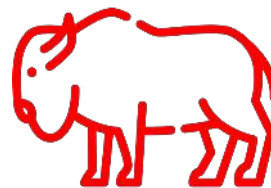
Cluster: `kind-armadillo`



Cluster: `kind-bison`



# Demo Time!



#IstioCon

@rytswd



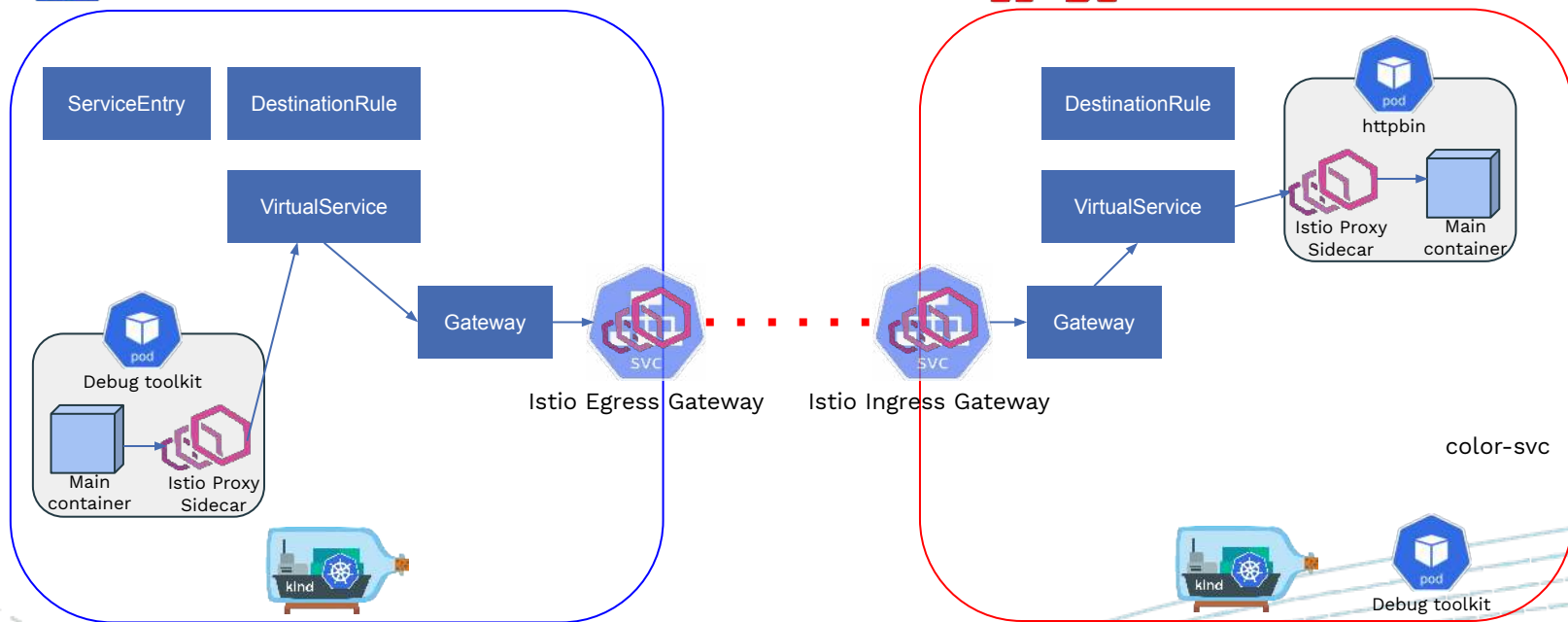
# Summary of demo



Cluster: **kind-armadillo**



Cluster: **kind-bison**



# What is GitOps?

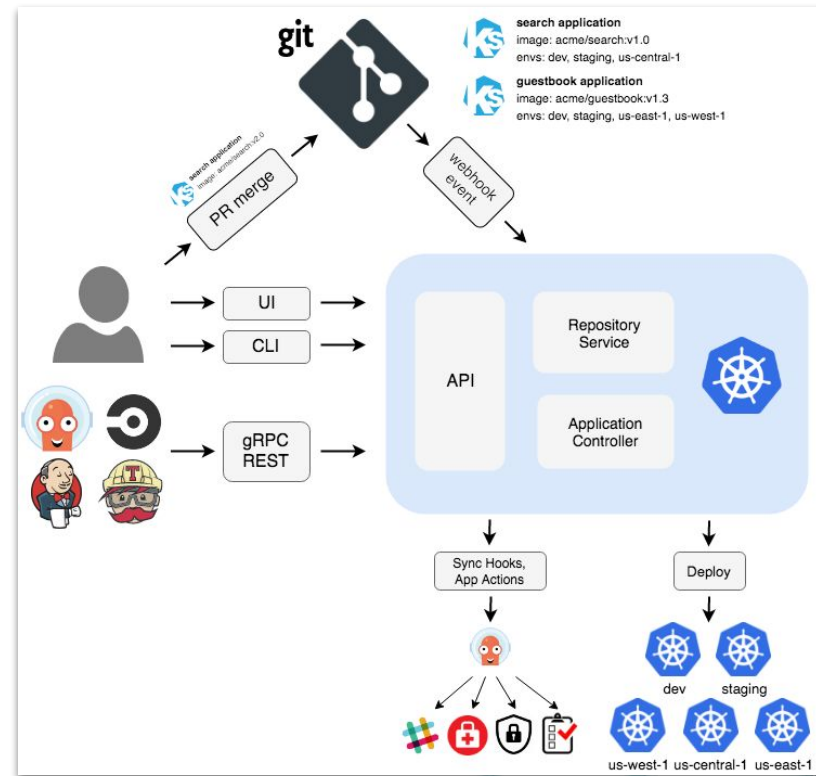
- Declarative cluster management
- All changes are driven by Git repo

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)

#IstioCon @rytswd



# 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>



**Argo CD**

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

Version v1.8.2



**Prometheus**

<https://github.com/prometheus/prometheus>

Prometheus Operator Version v0.45.0



**Kiali**

<https://github.com/kiali/kiali>

Kiali Operator Version v1.29.0



**Grafana**

<https://github.com/grafana/grafana>

Version v7.4.0, Helm Chart Version v6.3.0



**Jaeger**

<https://github.com/jaegertracing/jaeger>

Jaeger Operator Version v1.21.3

NOTE: This demo is based on Istio Version 1.7.5.

#IstioCon

@rytswd





# Goal of demo

Color Grid for Demo

Base Address:  Number of Cells:

Endpoint:

Periodic Query: ☐

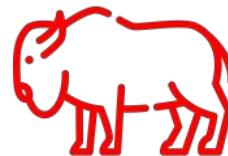
Unset	Unset
Unset	Unset
Unset	Unset



color-svc

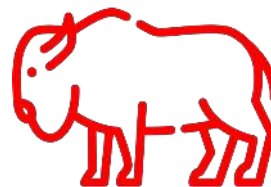
<https://github.com/rytswd/color-svc>

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





# Demo Time!



#IstioCon

@rytswd



# Summary of Demo

- Istio's powerful traffic management can pose some configuration challenges
- Even for Getting Started, multicluster 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  Number of Cells

Endpoint

Periodic Query ☐

Endpoint

Periodic Query ☐

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



# What's the next step from here?

Some interesting talks at IstioCon 2021

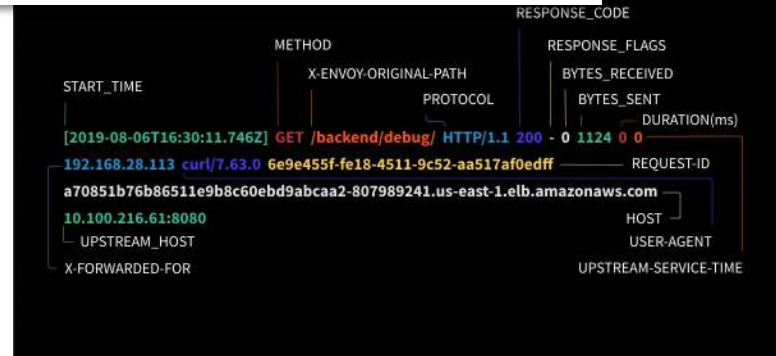
- What Envoy Hears When Istio Speaks  
by Rob Salmond (Monday)
- Istio Multicluster Workshop  
by Denis Jannot & Christian Posta (Tuesday)
- Taming Istio Configuration with Helm  
by Ryan Michela (Wednesday)
- Know your peers  
by Alex Van Boxel (Wednesday)

#IstioCon

@rytswd



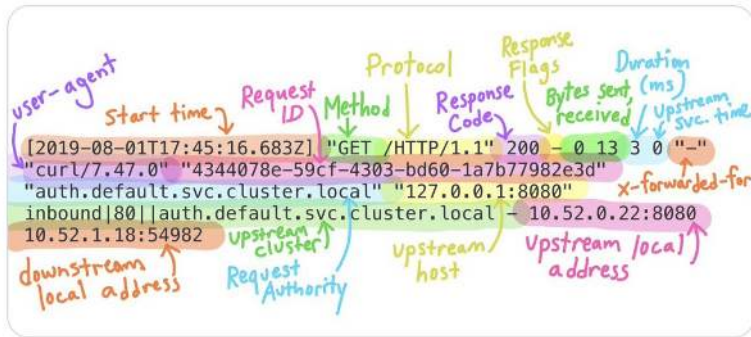
Link



@rytswd



These are @EnvoyProxy access logs, and contain lots of helpful info! [envoyproxy.io/docs/envoy/latest/logs/access\\_log](https://envoyproxy.io/docs/envoy/latest/logs/access_log)



9:43 PM · Aug 1, 2019

 133
  42
  [Copy link to Tweet](#)

Link



# Appendix: References

Repository used:

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

