I upgraded 3k proxies in my sleep

And you can too!

Mitch Connors/@therealmitchconnors

IstioCon

Istio Makes Your Service Mesh Secure

Security is one of the leading motivations for adoption of Istio, with best-in-class features like:

- Zero Trust
- mTLS
- Certificate Rotation
- AuthN/AuthZ

But...

88%

Of Istio Installations are running known CVEs

Agenda

Why Aren’t Users Upgrading?

How I upgraded 3k proxies in my sleep

How you can too!

Takeaways

Terminology Warning

For this presentation, Proxy, Sidecar, and Data Plane all refer to the Envoy Sidecar Proxy provided by Istio.

Istio Architecture Primer

Istio

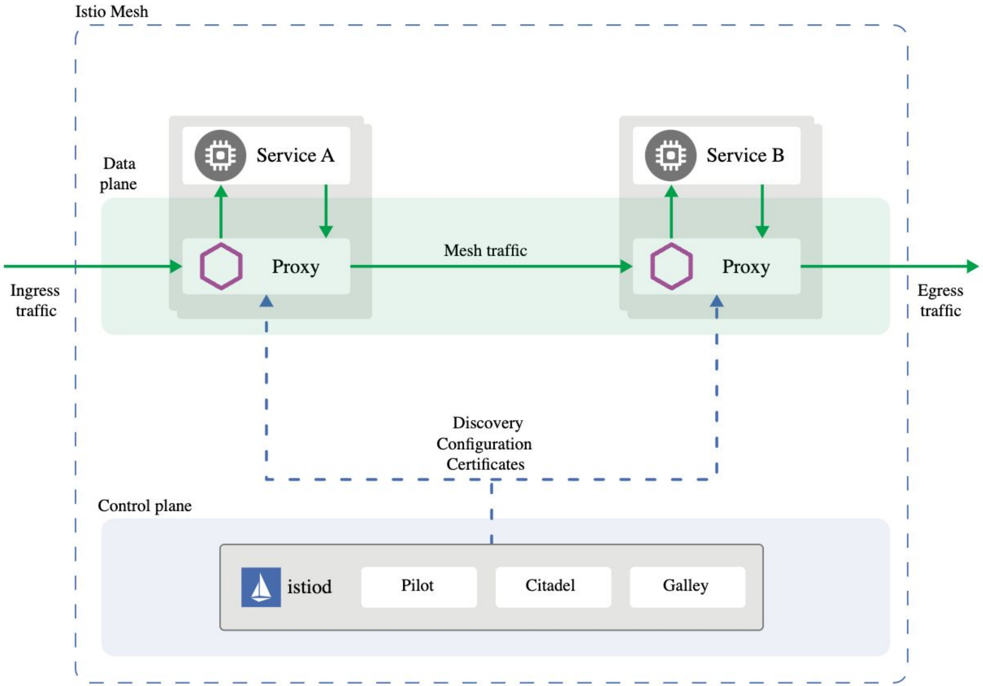
=

1x Control Plane

+

Data Plane

(N Proxies)



Istio Architecture

Why Won’t they Upgrade

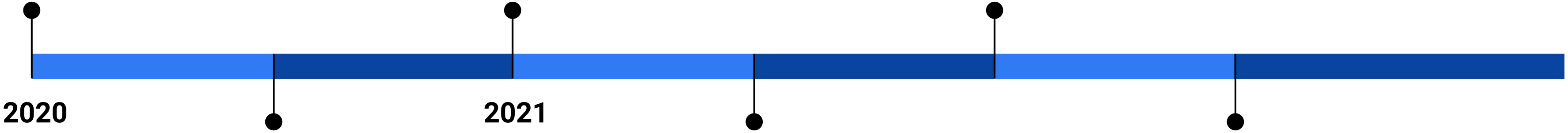
Q2 2020 - Hypothesis: They don’t know they’re vulnerable

Q3 2020 - Hypothesis: Upgrading is too hard

Q2 2021 - Hypothesis: Minor version upgrades are needed too often

Q1 2022 - Hypothesis: Humans are bad at repetitive tasks

Why Won’t they Upgrade



2020

Q2 Hypothesis: Users lack information

Many users may be running vulnerable versions without knowing it.

Q3 Upgrade Survey

We asked users which versions they ran, whether they were vulnerable, and why they weren’t running something newer.

2021

Q4 Hypothesis: Upgrades are too difficult and dangerous

Many users know they are out of date, but don’t have time to fix it.

Q1 Upgrade Roadmap

Istio’s 2021 and 2022 roadmaps are all about upgrade ease, and a dedicated WG is created to execute on this roadmap.

Q2 Hypothesis: Minor Version Upgrades are dangerous and frequent

Asking users to perform 4x minor version upgrades a year is too much.

Q3 Reduced Upgrade Cadence

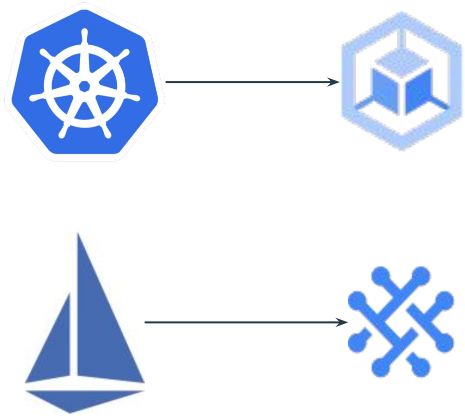
By allowing users to skip minor versions, and extending our support window, we support users upgrading 2x/year instead of 4x/year.

Humans are Bad at Monotonous, Repetitive Labor

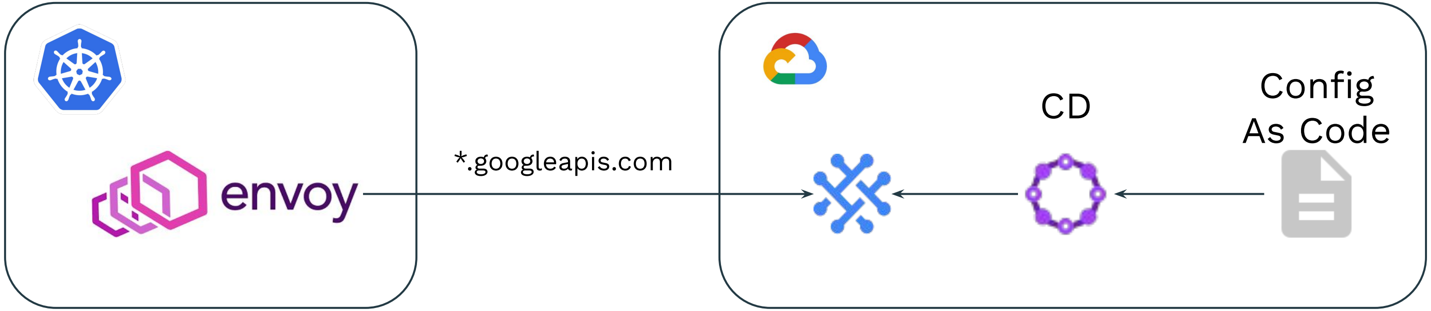
New Hypothesis, Q1 2022

Solving Problems the Google Way

Let’s build a managed service!



IstioD as a Managed Service

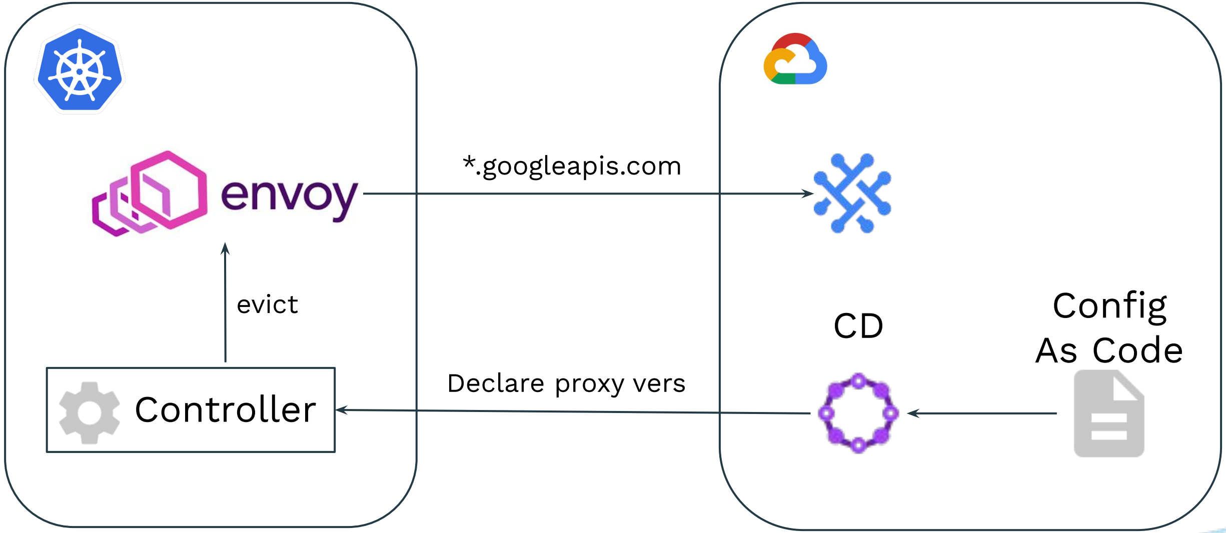


Are We Managed Yet?

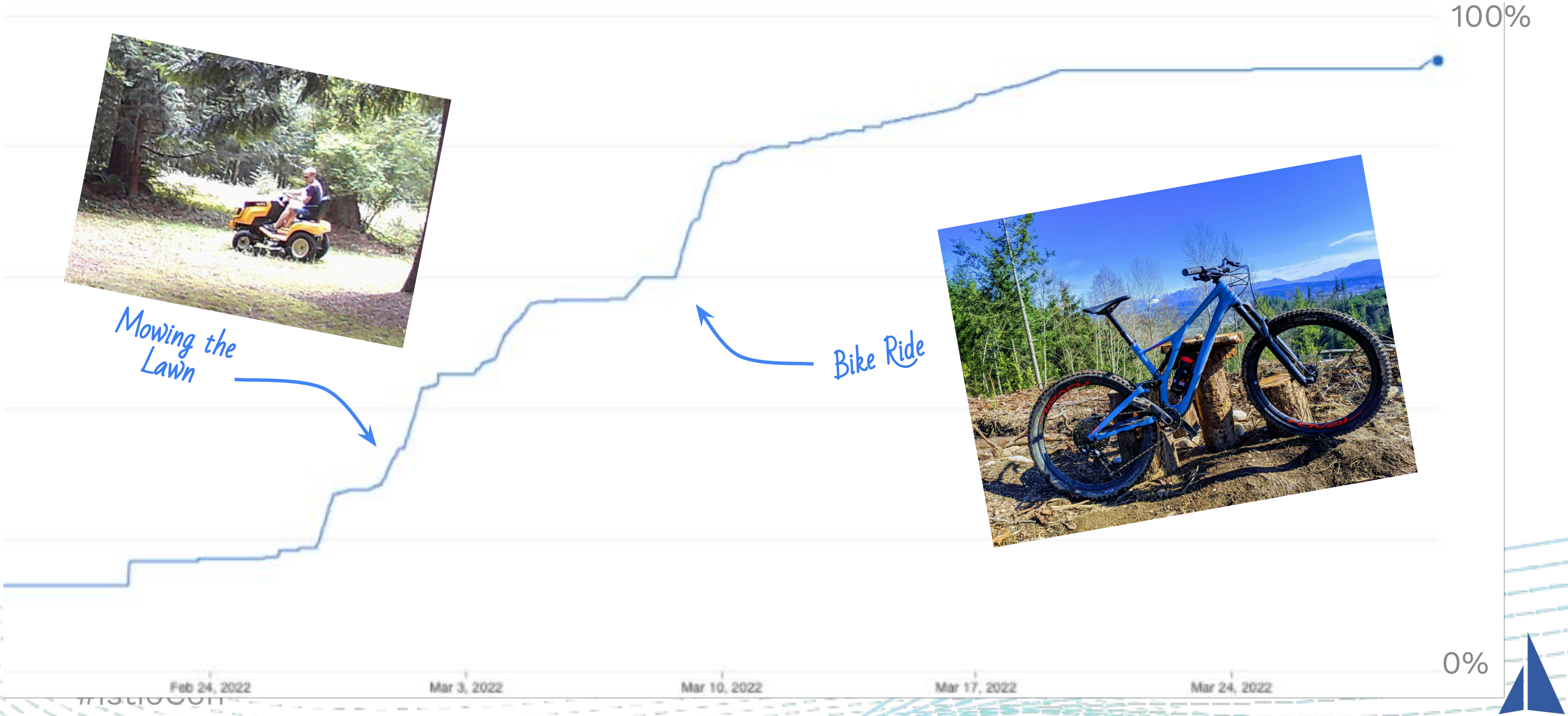
What does Managed mean for a sidecar?

A managed component is one that users don’t need to think about. It just works.

Managing the ASM Data Plane



My Rollout Timeline



How to Upgrade 3000 Proxies in Your Sleep

Or on a bike

...

Or a lawnmower

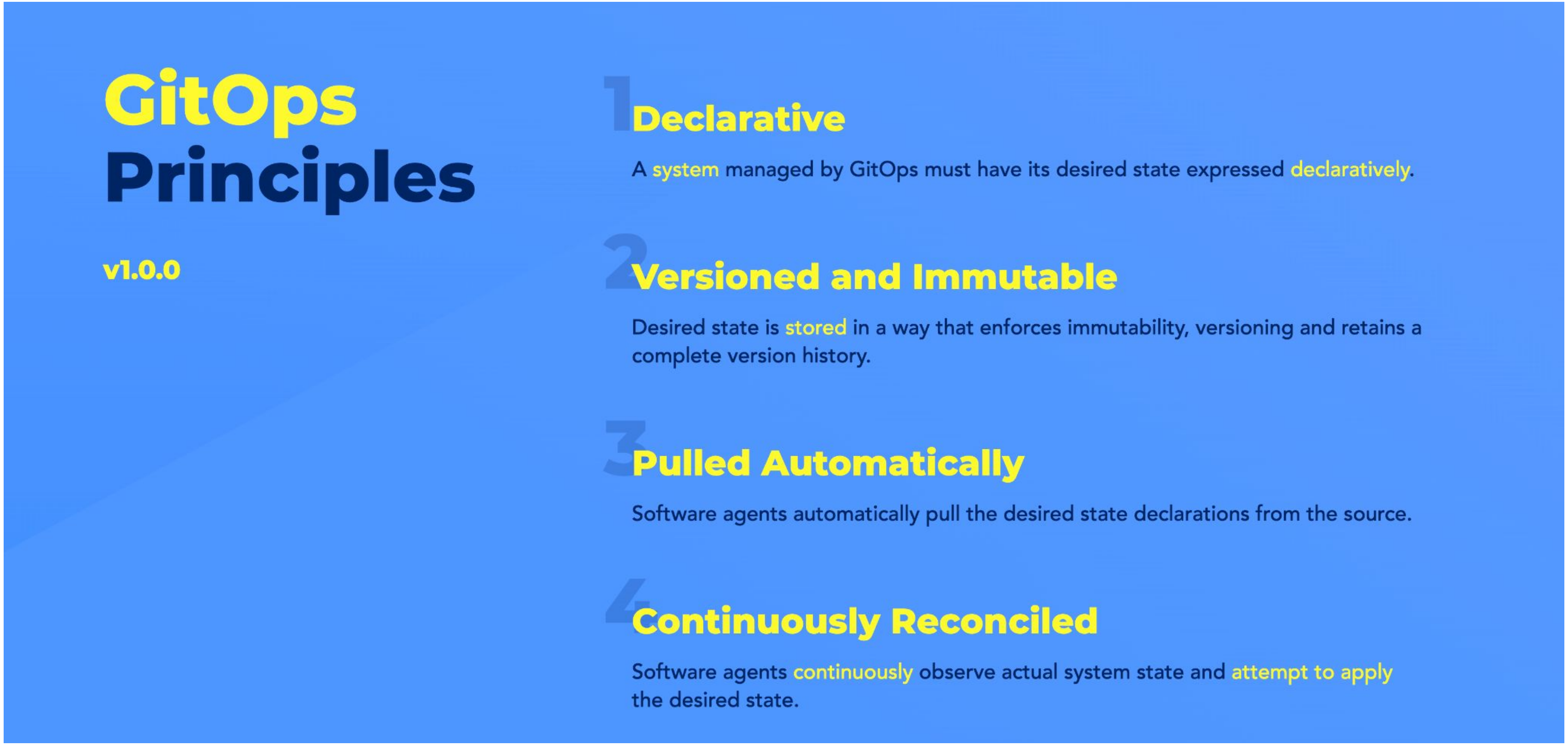
Building an OSS Config-as-Code system

Can we accomplish similar goals in OSS?

Can we define 100% of our Service Mesh's state in source control?

Rollbacks should be as simple as reverting a Pull Request.

Let's build a system that pulls from source, and pushes to k8s...



GitOps Principles

v1.0.0

1 Declarative

A system managed by GitOps must have its desired state expressed declaratively.

2 Versioned and Immutable

Desired state is stored in a way that enforces immutability, versioning and retains a complete version history.

3 Pulled Automatically

Software agents automatically pull the desired state declarations from the source.

4 Continuously Reconciled

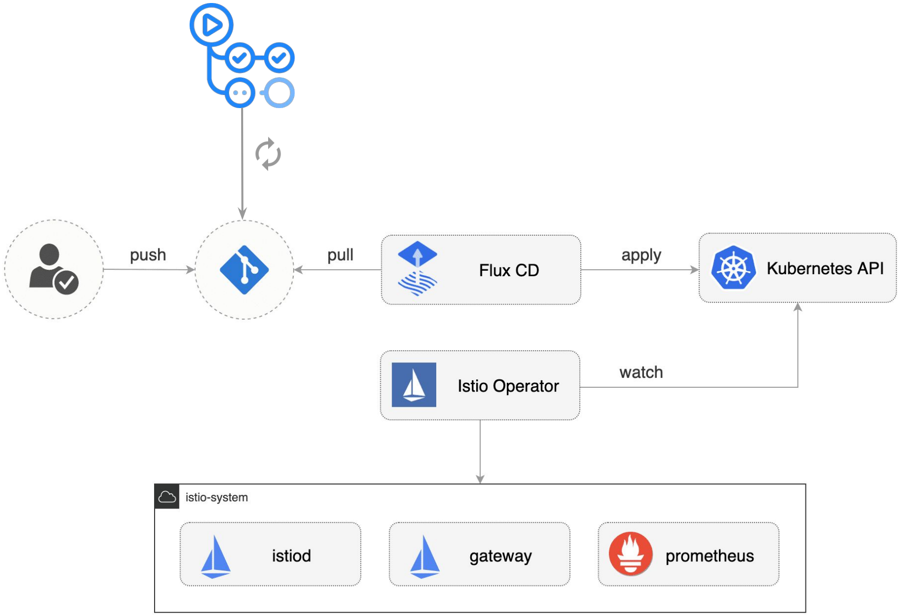
Software agents continuously observe actual system state and attempt to apply the desired state.

https://opengitops.dev/

GitOps for IstioD

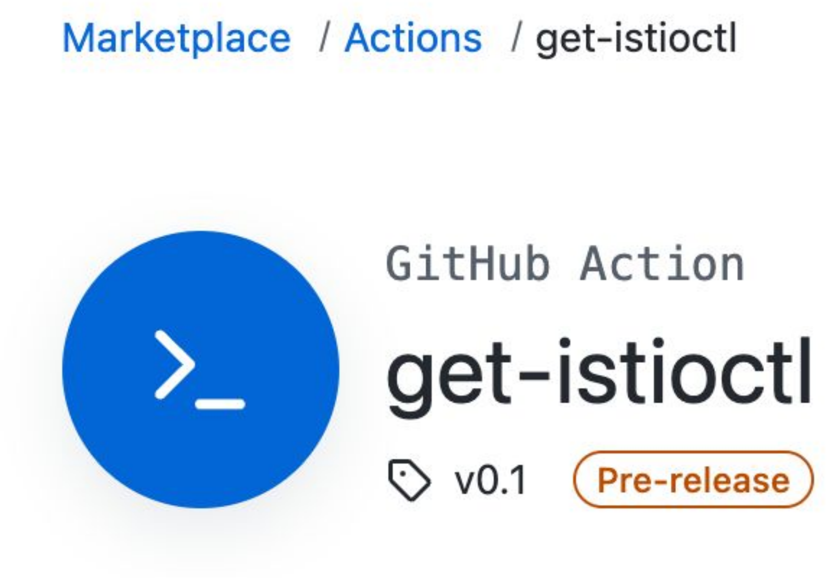
Created by the Stefan @ WeaveWorks

GitHub Actions check for new Istio Versions, open Pull Requests



https://github.com/stefanprodan/gitops-istio

Launched: GitHub Actions for Istio



Marketplace / Actions / get-istioctl

GitHub Action

get-istioctl

v0.1 Pre-release

Demo - Control Plane Upgrade

Advantages

- Control Plane maintained up-to-date within semver range
- Out-of-range updates trigger pull requests from GitHub Actions
- Full stack tested in kind e2e tests before production rollout
- Istioctl Analyze runs on every change

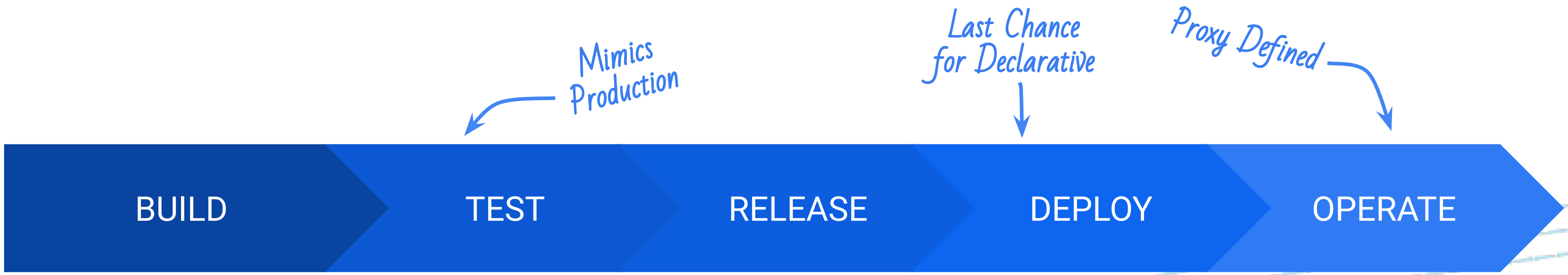
Disadvantages

- Proxy Upgrades still uncontrolled
- Needs updates for Helm, Revision-based upgrades

The Problem with Proxies

Imperative Definition

- Reduces Test Determinacy
- Updates Rollout Uncontrolled
- Frequent Global Restarts Required



Mimics Production

Last Chance for Declarative

Proxy Defined

BUILD

TEST

RELEASE

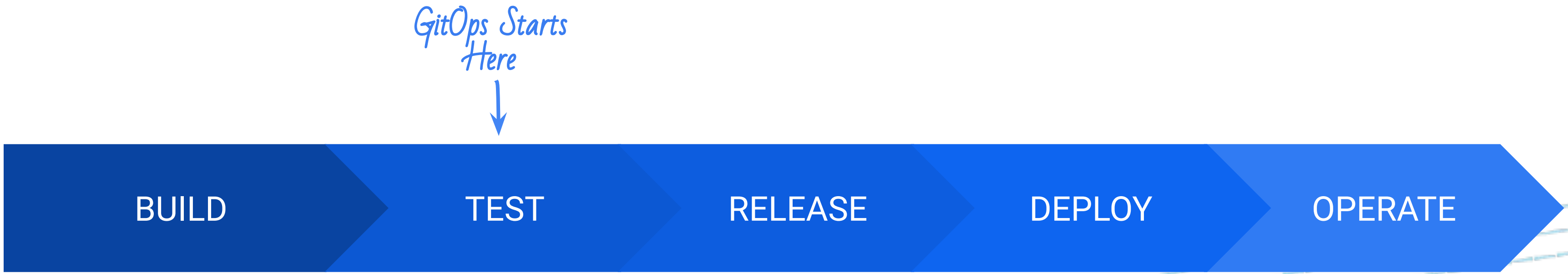
DEPLOY

OPERATE

Shifting Left

Define the Proxy in Git

- All environments use same proxy
- Proxy Upgrades according to Application Rules
- Rollbacks are as simple as git revert



GitOps Starts Here

BUILD

TEST

RELEASE

DEPLOY

OPERATE

How to Define your Sidecar in Git

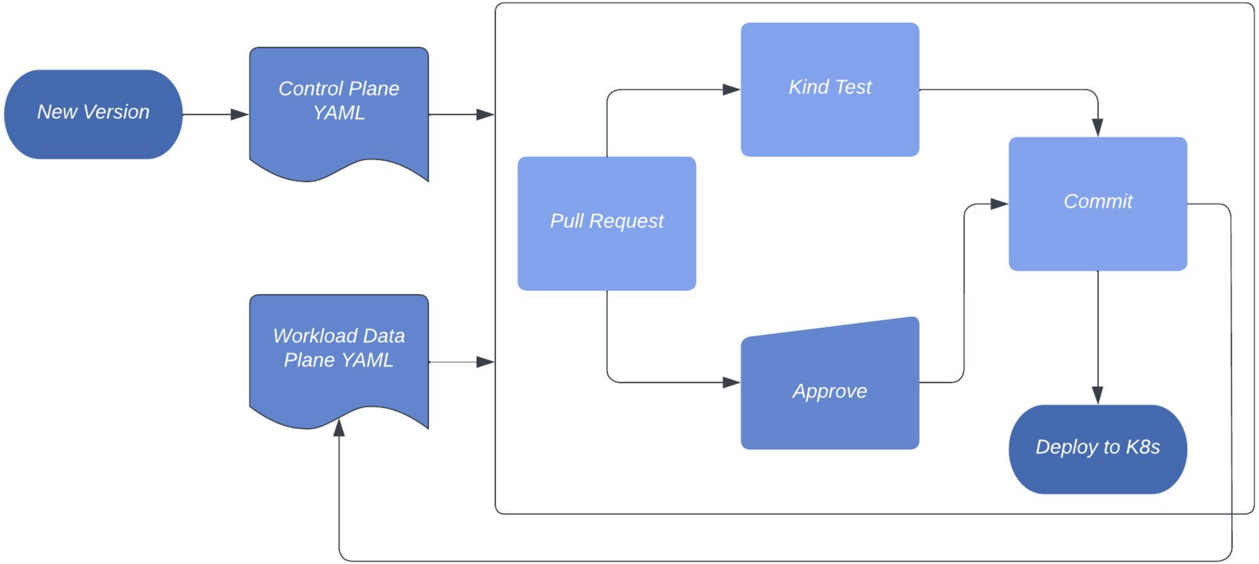
Disable Sidecar Injection

Run kube-inject from GitHub Actions

Deployment.yaml
+
istioctl kube-inject
=
Deployment.yaml w/sidecar

```
$ istioctl kube-inject --help

kube-inject manually injects the Istio sidecar into
Kubernetes
workloads. When in doubt re-run istioctl kube-inject
on deployments to get the most up-to-date changes.

It's best to do kube-inject when the resource is
initially created.

Usage:
  istioctl kube-inject [flags]

Examples:
  # Update resources on the fly before applying.
  kubectl apply -f <(istioctl kube-inject -f
<resource.yaml>)

  # Update an existing deployment.
  kubectl get deployment -o yaml | istioctl
kube-inject -f - | kubectl apply -f -
```

Current Workflow



Demo - Data Plane Upgrade

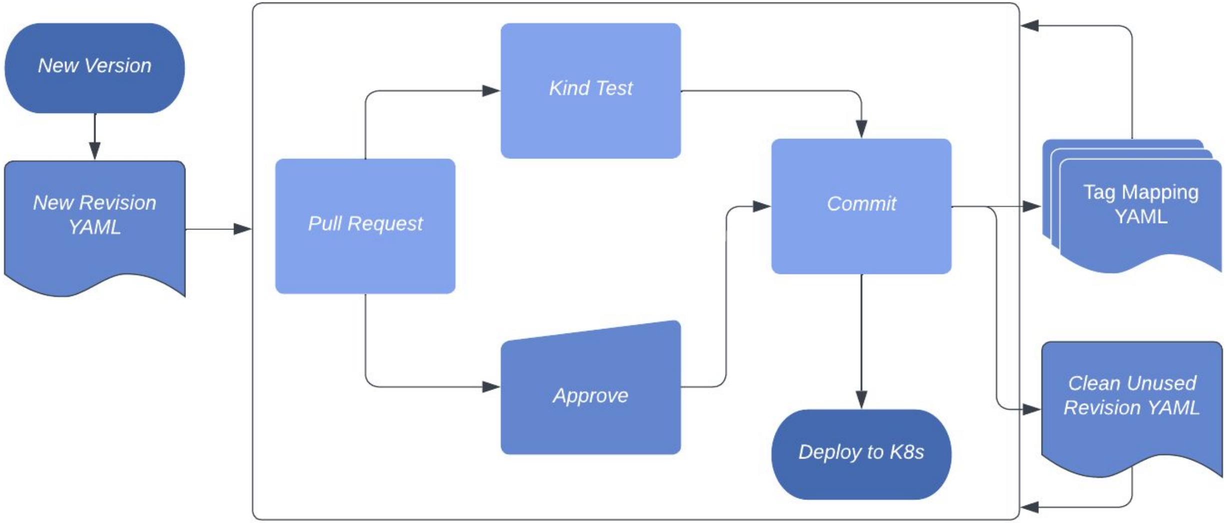
Advantages

- Proxy maintained up-to-date within semver range
- Side effect: Canaries rollout out updates to proxy, as well as application
- Automated Canary rollback

Disadvantages

- Defining Proxy in Git Adds Lots of noise to workload.yamls
- Doesn’t respect Revisions
- Rollouts to multiple clusters must be manually coordinated

Future Workflow



Key Takeaway

Have a plan for staying up to date

Automate Your Istio Upgrades

OR

Pay a vendor to upgrade Istio

OR

Budget ~1 engineer to upgrade Istio

Related Resources

Upgrade Listening Session - Tuesday 11:00

ASM Workshop - Wed 11:00

ASM @ WP Engine - Wed 11:30

Chat w/ ASM Team - https://bit.ly/asmchat22

Thank you!

@therealmitchconnors
github.com/therealmitchconnors/gitops-istio