



Progressive Delivery Techniques

with Flagger and Service Mesh

Paul Curtis, Weaveworks
ServiceMeshCon
18 November, 2019

Flagger - The Progressive Delivery Operator

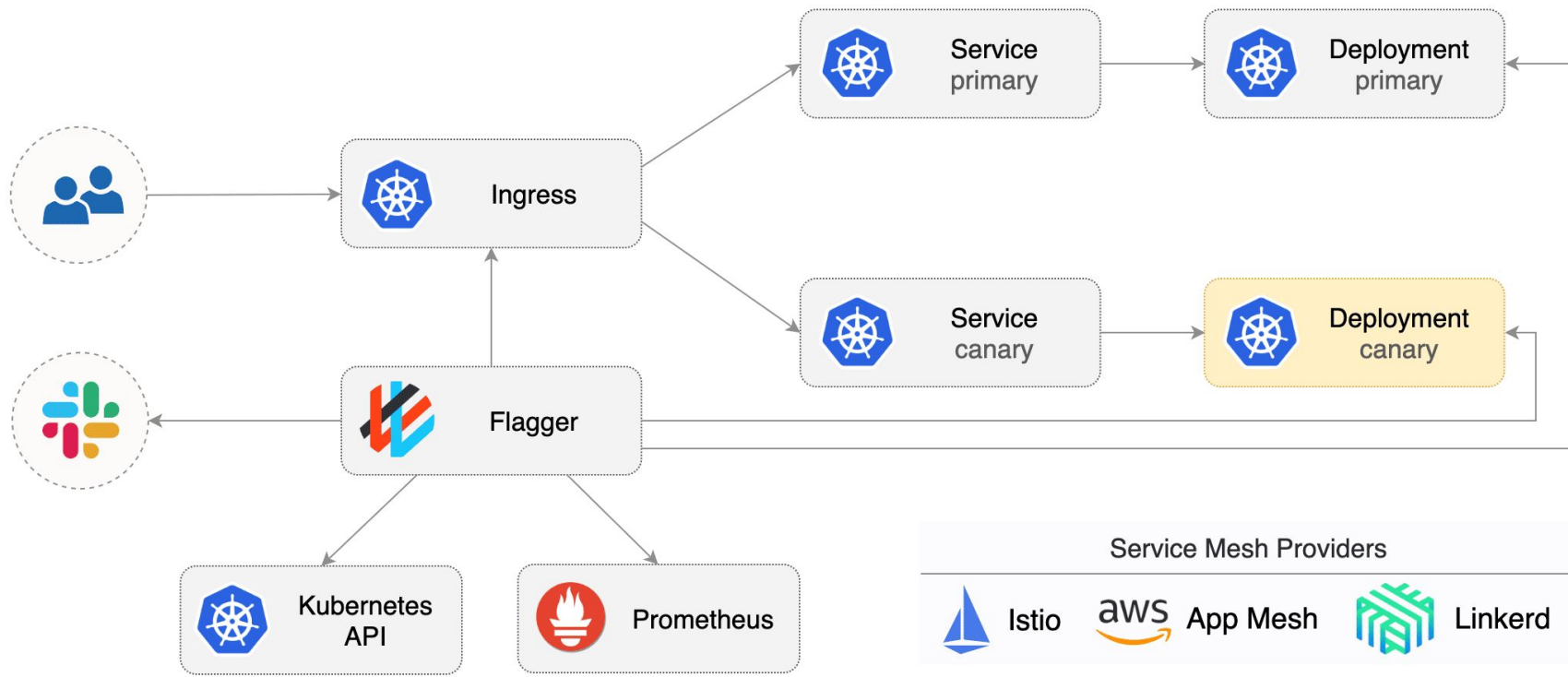
Flagger is a Kubernetes operator that automates the promotion of canary deployments using **App Mesh**, **Istio**, **Linkerd** routing for traffic shifting and **Prometheus** metrics for canary analysis.

When not using a service mesh, Flagger can orchestrate canary releases with ingress controllers like **Gloo** and **NGINX**.

Flagger implements a **control loop** that gradually shifts traffic to the canary while measuring key performance indicators. Based on the KPIs **analysis** a canary is promoted or aborted.



Flagger overview



Deploying Services Seamlessly with No Down Time

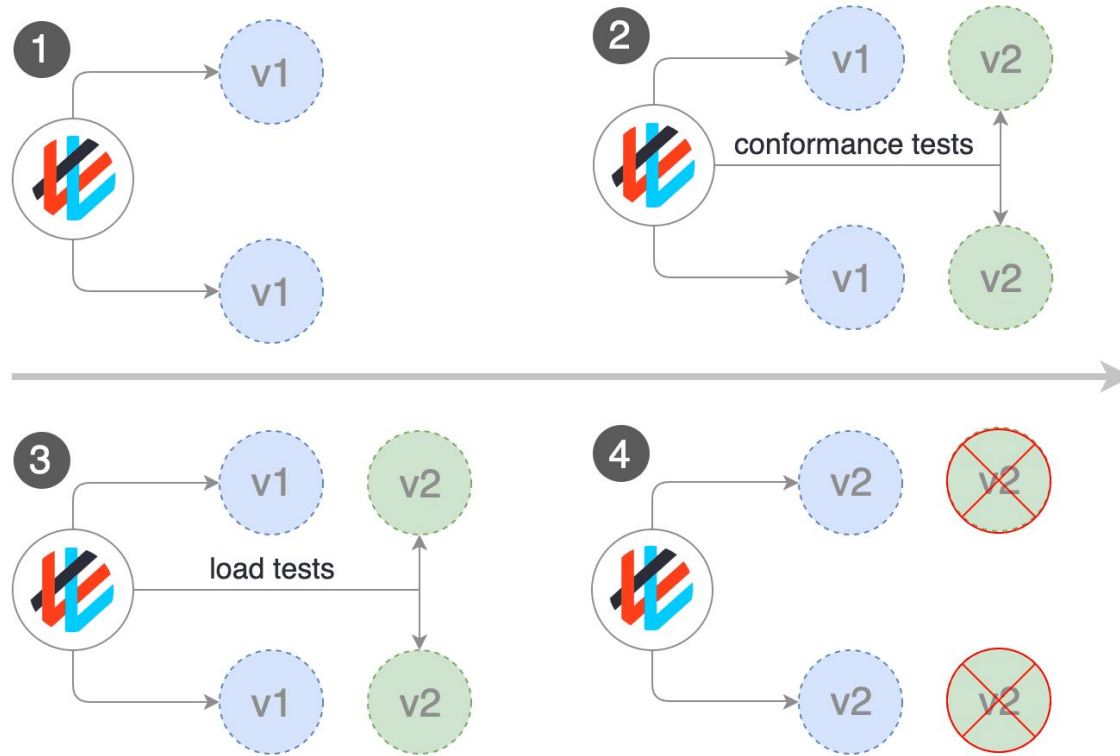


Deployment Strategies

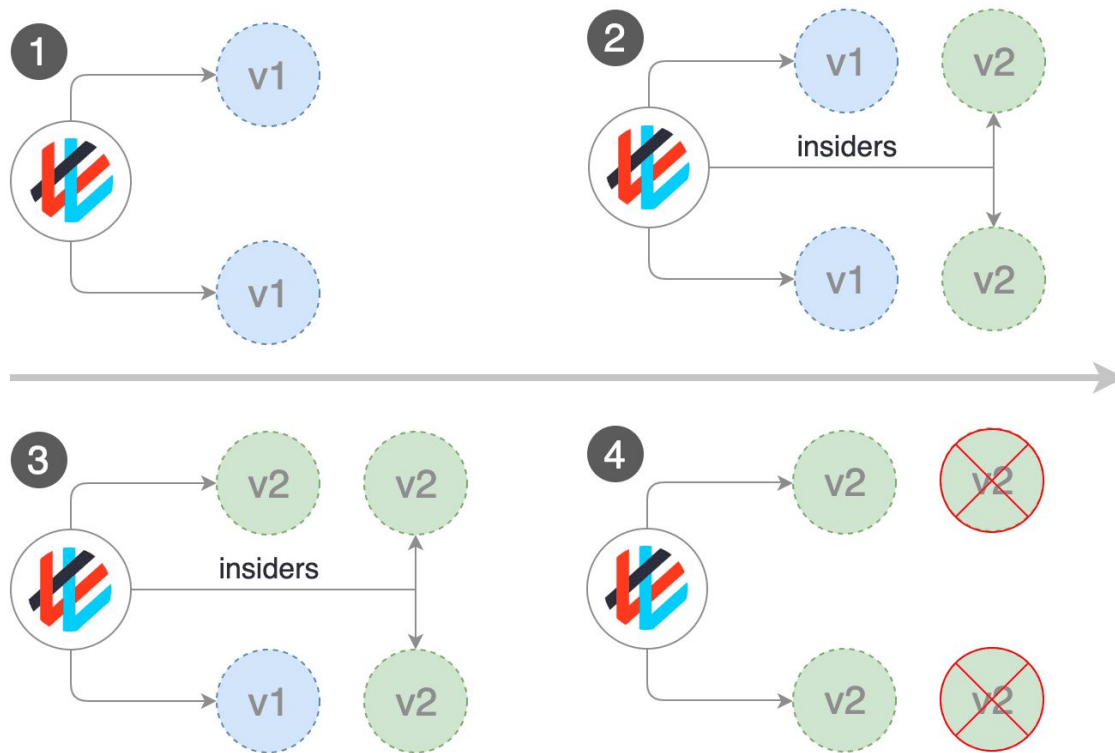
- **Canary** (progressive traffic shifting)
 - *Istio, Linkerd, App Mesh, NGINX, Gloo*
- **A/B Testing** (HTTP headers and cookies traffic routing)
 - *Istio, App Mesh, NGINX*
- **Blue/Green** (traffic switching)
 - *Kubernetes CNI, Istio, Linkerd, App Mesh, NGINX, Gloo*
- **Mirroring** (traffic shadowing)
 - *Istio*



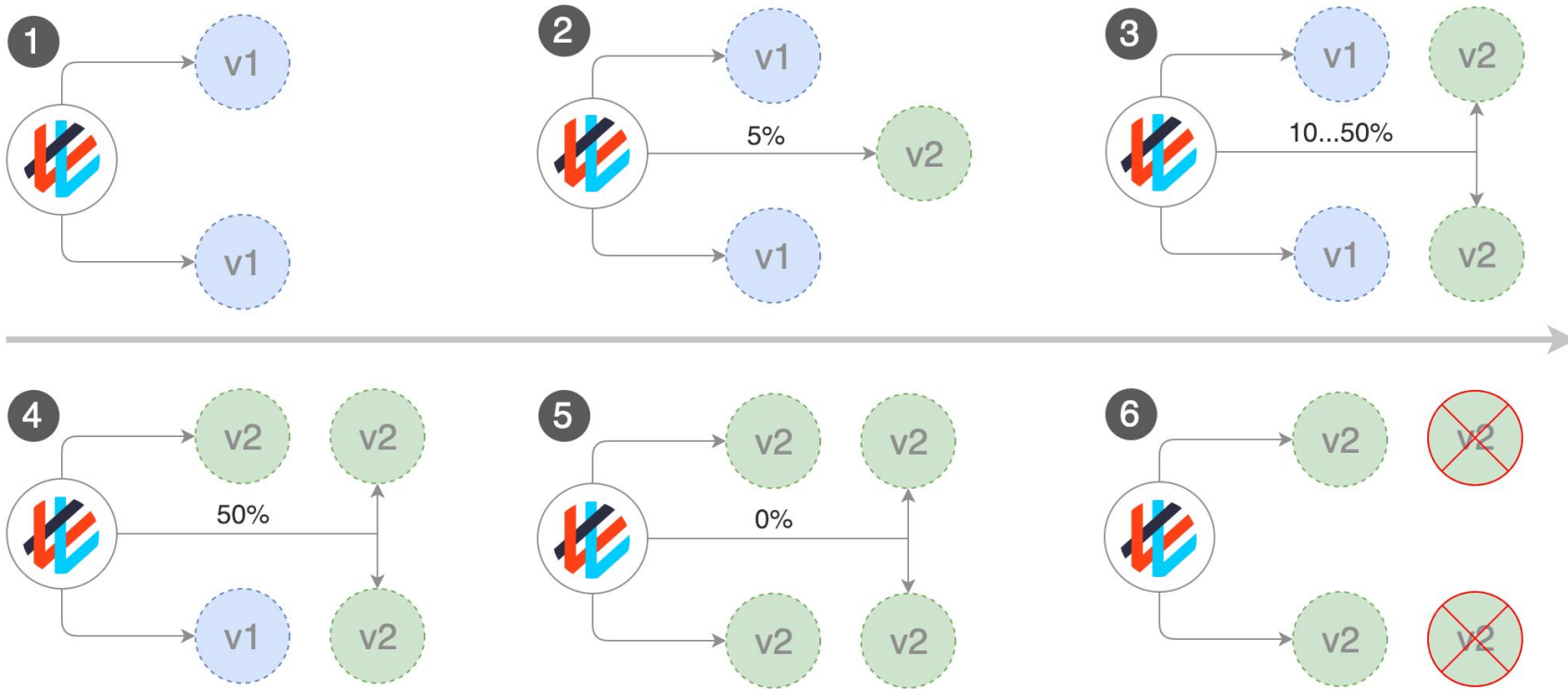
Blue/Green - Deployment Strategy



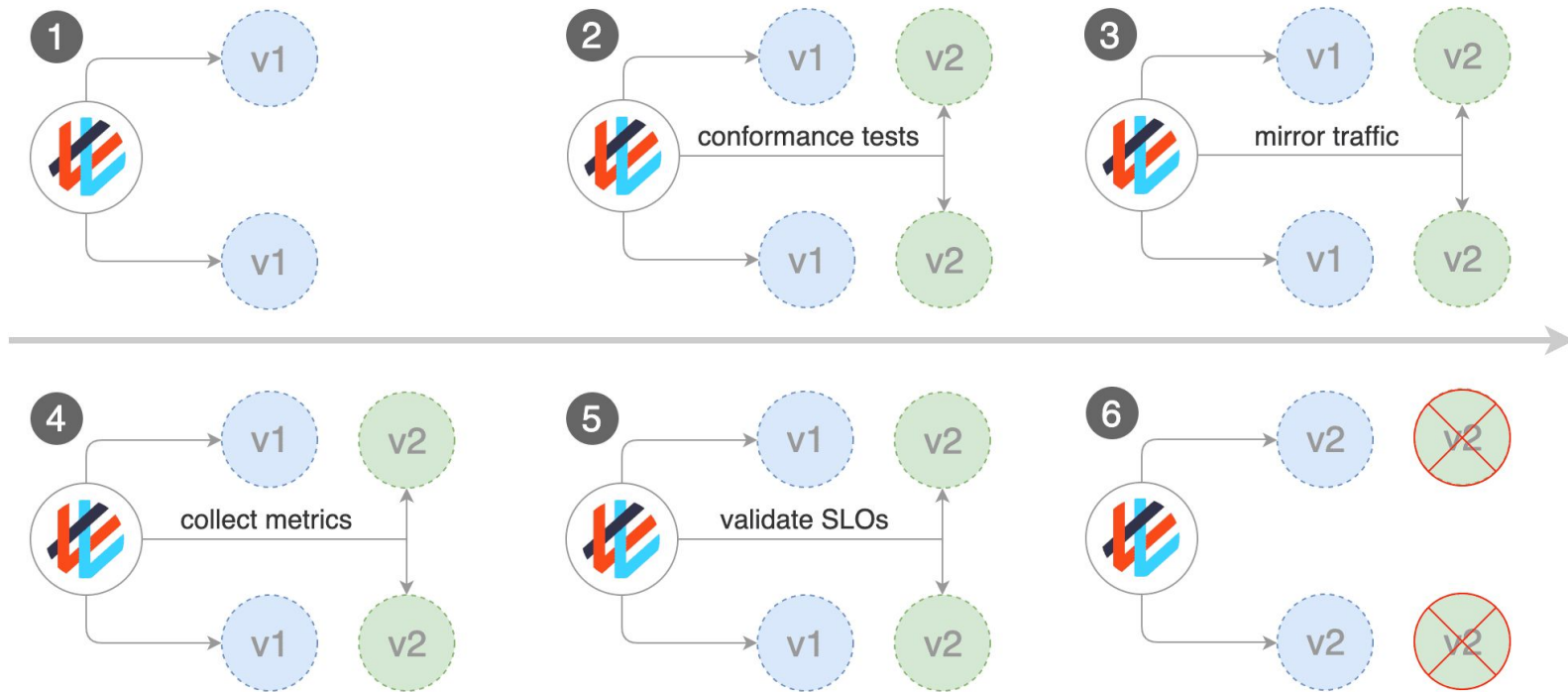
A/B Testing - Deployment Strategy



Canary - Deployment Strategy



Traffic Mirroring - Deployment Strategy



Using Flagger to Bootstrap Canary Enabled Deployments



Flagger - Service mesh automation

Manual canary setup

Kubernetes objects

1. Canary Deployment
2. Canary ClusterIP Service
3. Canary Horizontal Pod Autoscaler
4. Primary Deployment
5. Primary ClusterIP Service
6. Primary Horizontal Pod Autoscaler

Service mesh objects

1. Canary routes/destination rules/nodes
2. Primary routes/destination rules/nodes
3. Virtual services
4. Traffic split

Automated canary setup

Kubernetes objects

1. Deployment
2. Horizontal Pod Autoscaler

Flagger objects

1. Canary



Flagger - Canary CRD

`apiVersion: flagger.app/v1alpha1`

`kind: Canary`

`metadata:`

`name: podinfo`

`namespace: test`

`spec:`

`# deployment reference`

`targetRef:`

`apiVersion: apps/v1`

`kind: Deployment`

`name: podinfo`

`# hpa reference (optional)`

`autoscalerRef:`

`apiVersion: autoscaling/v2beta1`

`kind: HorizontalPodAutoscaler`

`name: podinfo`

`# the maximum time in seconds for the canary deployment`

`# to make progress before it is rollback (default 600s)`

`progressDeadlineSeconds: 60`

`service:`

`# container port`

`port: 9898`

`# port name (can be http or grpc)`

`portName: http`

`# timeout (optional)`

`timeout: 15s`

`# host names (optional)`

`hosts:`

`- app.example.com`

Canary CRD - Analysis spec

canaryAnalysis:

schedule interval (default 60s)

interval: 1m

maximum number of failed metric checks

before rolling back the canary

threshold: 10

max traffic percentage routed to canary

maxWeight: 50

canary increment step

stepWeight: 5

metrics:

- name: request-success-rate

minimum req success rate percentage (non 5xx)

threshold: 99

interval: 1m

- name: request-duration

maximum req duration P99 (milliseconds)

threshold: 500

interval: 30s

webhooks:

- name: load-test

url: http://flagger-loadtester.test/

timeout: 5s

metadata:

cmd: "hey -z 1m -q 10 -c 2 http://podinfo.test:9898/"

Flagger - Validation process

Flagger lets you define **key performance indicators** and **thresholds**. The decision to pause the traffic shift, abort or promote a canary is based on:

- Deployment health status
- Request success rate percentage (Service Mesh metric)
- Request latency average value (Service Mesh metric)
- Custom metric checks (Prometheus queries)
- Webhooks (integration testing, load testing, etc)



Alerting

Flagger can be configured to publish the canary analysis result to Slack or Microsoft Teams.



flagger APP 3:30 PM

podinfo.test

New revision detected, starting canary analysis.

Target

Deployment/podinfo.test

Traffic routing

Weight step: 5 max: 50

Failed checks threshold

10

Progress deadline

60s

podinfo.test

Canary analysis completed successfully, promotion finished.



flagger APP 12:12 PM

podinfo.test

Progress deadline exceeded deployment does not have minimum availability for more than 60s



flagger APP 12:18 PM

podinfo.test

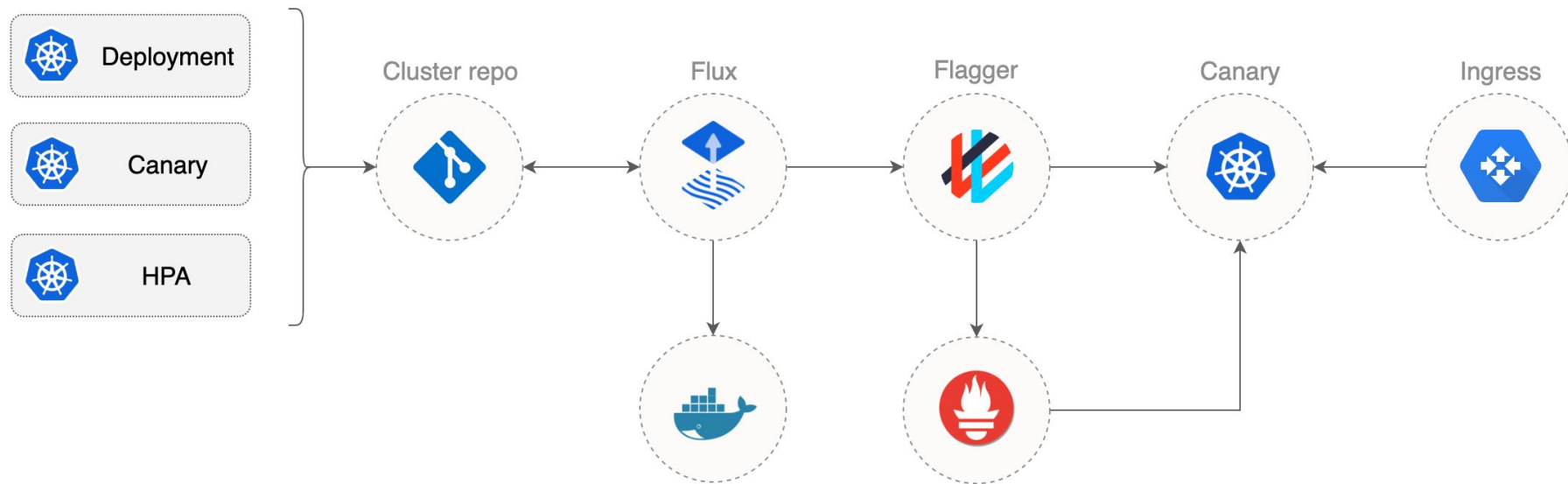
Failed checks threshold reached 10



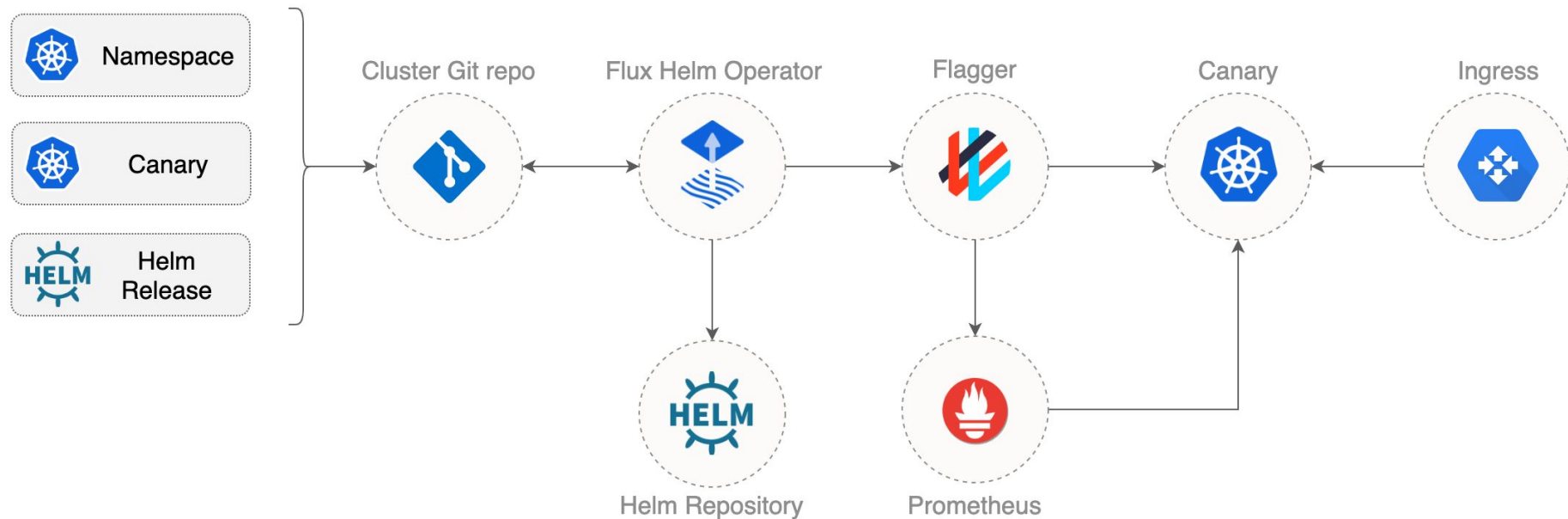
Progressive Delivery GitOps Pipelines



GitOps Pipeline - Flux + Flagger



GitOps Pipeline - Flux + Helm Operator + Flagger



More Information

Flagger Repo

<https://github.com/weaveworks/flagger>

Progressive Delivery for Istio with Flagger and FluxCD

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

Progressive Delivery for Linkerd with Flagger, FluxCD and Helm v3

<https://helm.workshop.flagger.dev/>

Progressive Delivery for AWS App Mesh with Flagger and FluxCD

<https://eks.hands-on.flagger.dev/>





Visit the **Weaveworks** booth!
(#S51)

Try our **GitOps Hands-On** for a prize!

**Stefan Prodan, Deep Dive: Flux the GitOps
Operator for Kubernetes**

Wednesday, 11:50am - 12:25pm
(Room 6C - Upper Level)

