



# Flagger

*Service Mesh Progressive Delivery Operator*

Stefan Prodan @stefanprodan

KubeCon Barcelona May 2019

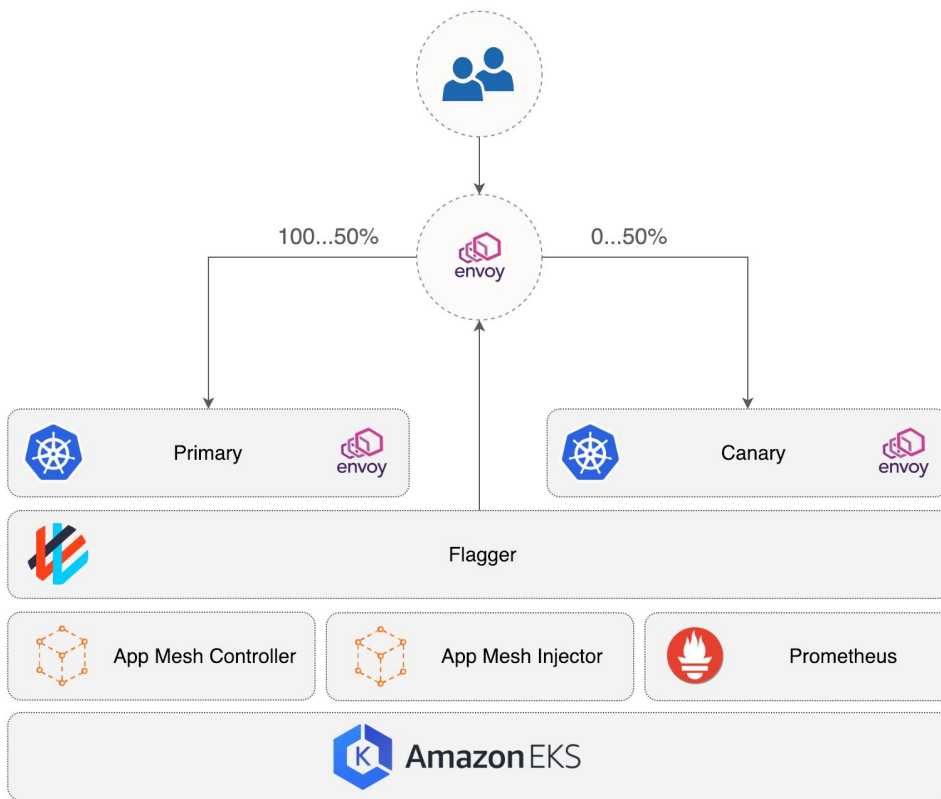
# Introducing Flagger

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

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

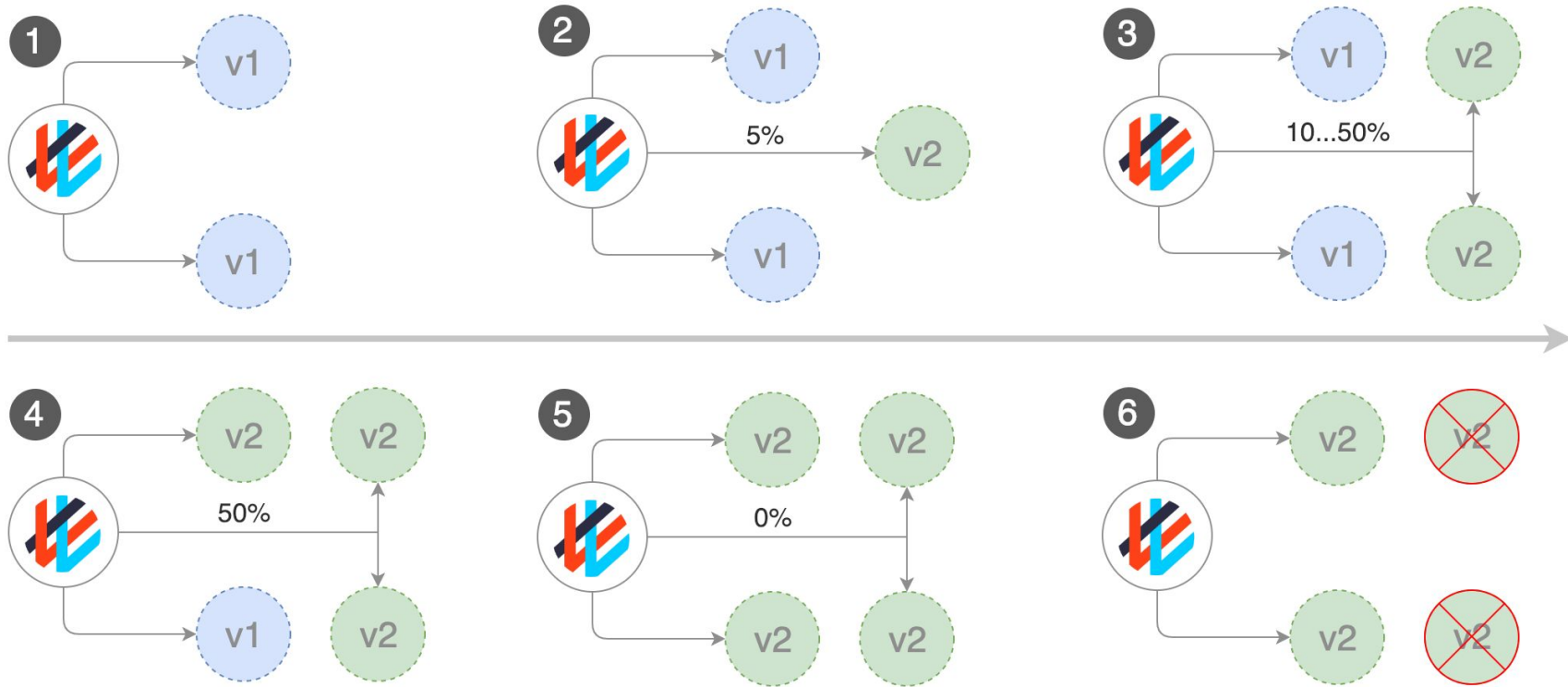


# Flagger Goals

- Give developers confidence in automating the production releases
  - Have control over the blast radius (traffic management)
  - Have control over the validation process (perf metrics)
- Make the deployment process observable
  - Real time feedback (Prometheus/Grafana)
  - Alerting (Slack/Alertmanager/PagerDuty/etc)
  - Emit Kubernetes events every time something changes
- Write as little YAML as possible
- Manage the whole process from Git



# Have control over the blast radius



# Have control over the validation process

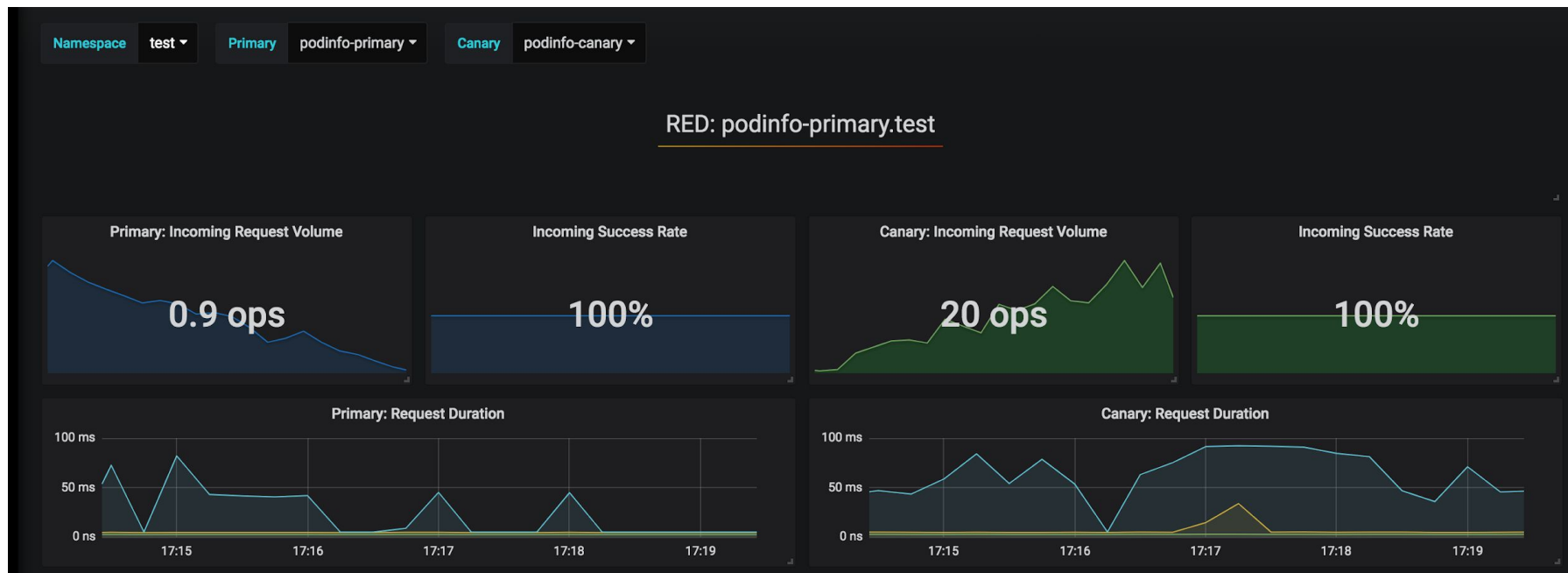
Flagger lets you define **key performance indicators** and **thresholds**. The decision to pause the traffic shift, abort or promote a canary is based on:

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



# Make the deployment process observable

Flagger comes with a Grafana dashboard for canary analysis.



# Make the deployment process observable

Flagger emits Kubernetes events related to the advancement and final status of a canary analysis.

```
Status:
  Canary Revision: 19871136
  Failed Checks: 0
  State: finished
Events:
  Type      Reason  Age   From      Message
  ----      -
Normal      Synced  3m    flagger   New revision detected podinfo.test
Normal      Synced  3m    flagger   Scaling up podinfo.test
Warning     Synced  3m    flagger   Waiting for podinfo.test rollout to finish: 0 of 1 updated replicas are available
Normal      Synced  3m    flagger   Advance podinfo.test canary weight 10
Normal      Synced  2m    flagger   Advance podinfo.test canary weight 20
Normal      Synced  1m    flagger   Advance podinfo.test canary weight 30
Normal      Synced  55s   flagger   Advance podinfo.test canary weight 40
Normal      Synced  35s   flagger   Advance podinfo.test canary weight 50
Normal      Synced  25s   flagger   Copying podinfo.test template spec to podinfo-primary.test
Warning     Synced  15s   flagger   Waiting for podinfo-primary.test rollout to finish: 1 of 2 updated replicas are available
Normal      Synced  5s    flagger   Promotion completed! Scaling down podinfo.test
```





# Make the deployment process observable

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



**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



# Write less YAML

## 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

### App Mesh objects

1. Canary virtual node
2. Primary virtual node
3. HTTP routes
4. Virtual router
5. Virtual service

## Automated canary setup

### Kubernetes objects

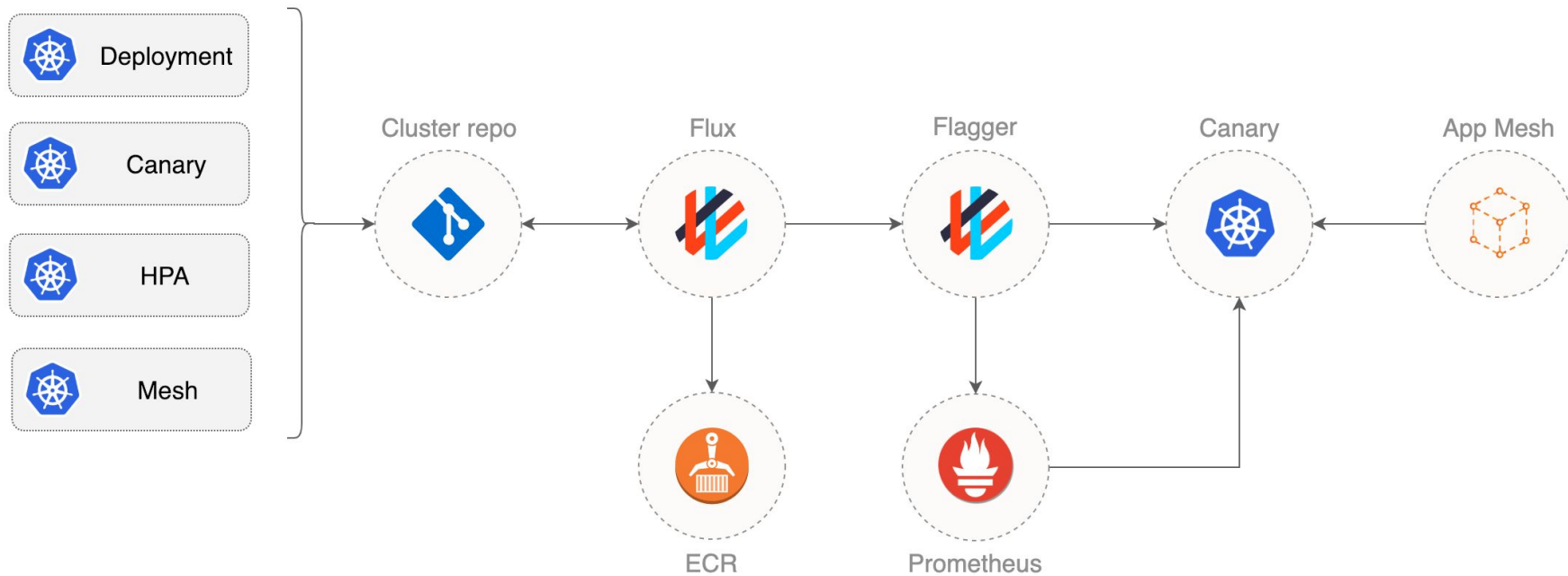
1. Deployment
2. Horizontal Pod Autoscaler

### Flagger objects

1. Canary



# GitOps pipeline





## ***GitOps Progressive Delivery Demo***

# Links

Flagger Repo

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

Flagger Docs

<https://docs.flagger.app>