

# PROGRESSIVE DELIVERY WITH



Carlos Sanchez / [csanchez.org](http://csanchez.org) / @csanchez



# **PROGRESSIVE DELIVERY**



# LaunchDarklyBlog

[Home](#)[Feature Management](#)[DevOps](#)[Continuous Delivery](#)[To Be Continuous](#)

[Home](#) > [Continuous Delivery](#)

## Progressive Delivery, a History.... Condensed

By Adam Zimman - August 6, 2018

• 4326



the developer-focused industry analyst firm

---

[Videos](#)   [Research](#)   [Events](#)   [About](#)   [Team](#)   [Services](#)   [Clients](#)   [Contact](#)

JAMES GOVERNOR'S MONKCHIPS

# Towards Progressive Delivery

By [James Governor](#) | [@monkchips](#) | August 6, 2018

*Progressive Delivery* is a term that includes deployment strategies that try to avoid the pitfalls of all-or-nothing deployment strategies

*New versions being deployed do not replace existing versions but run in parallel for an amount of time receiving live production traffic, and are evaluated in terms of correctness and performance before the rollout is considered successful.*

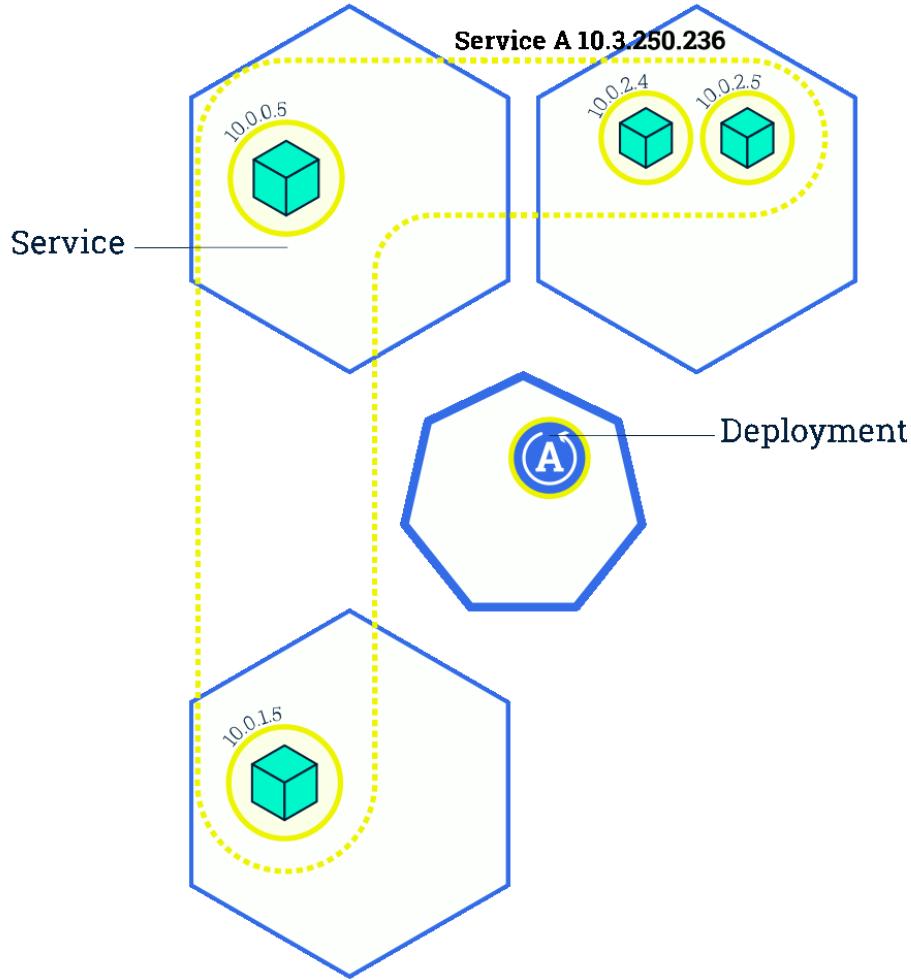
Continuous Delivery is hard

Progressive Delivery makes Continuous Delivery easier  
to adopt

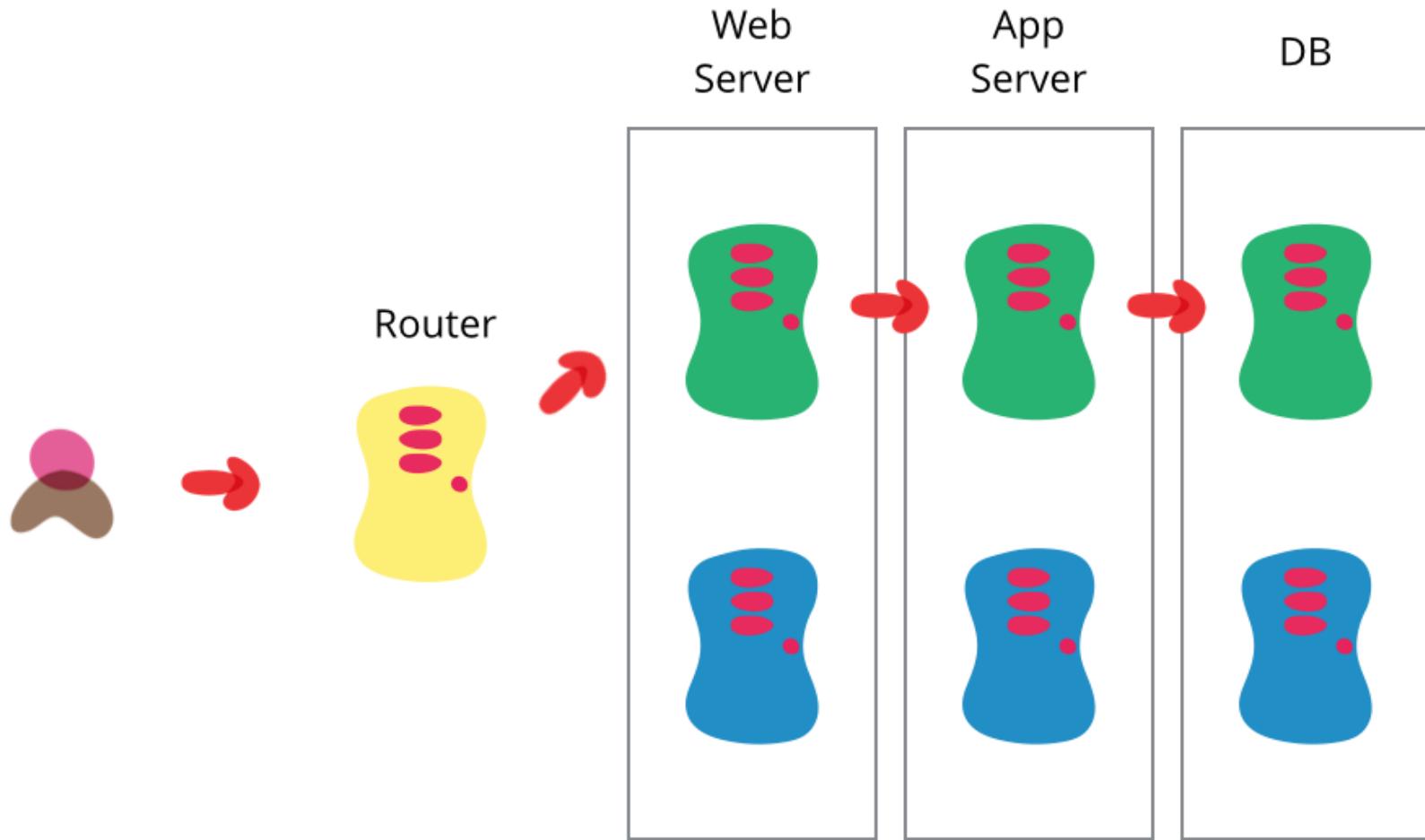
reduces the risk associated with Continuous Delivery

# **PROGRESSIVE DELIVERY TECHNIQUES**

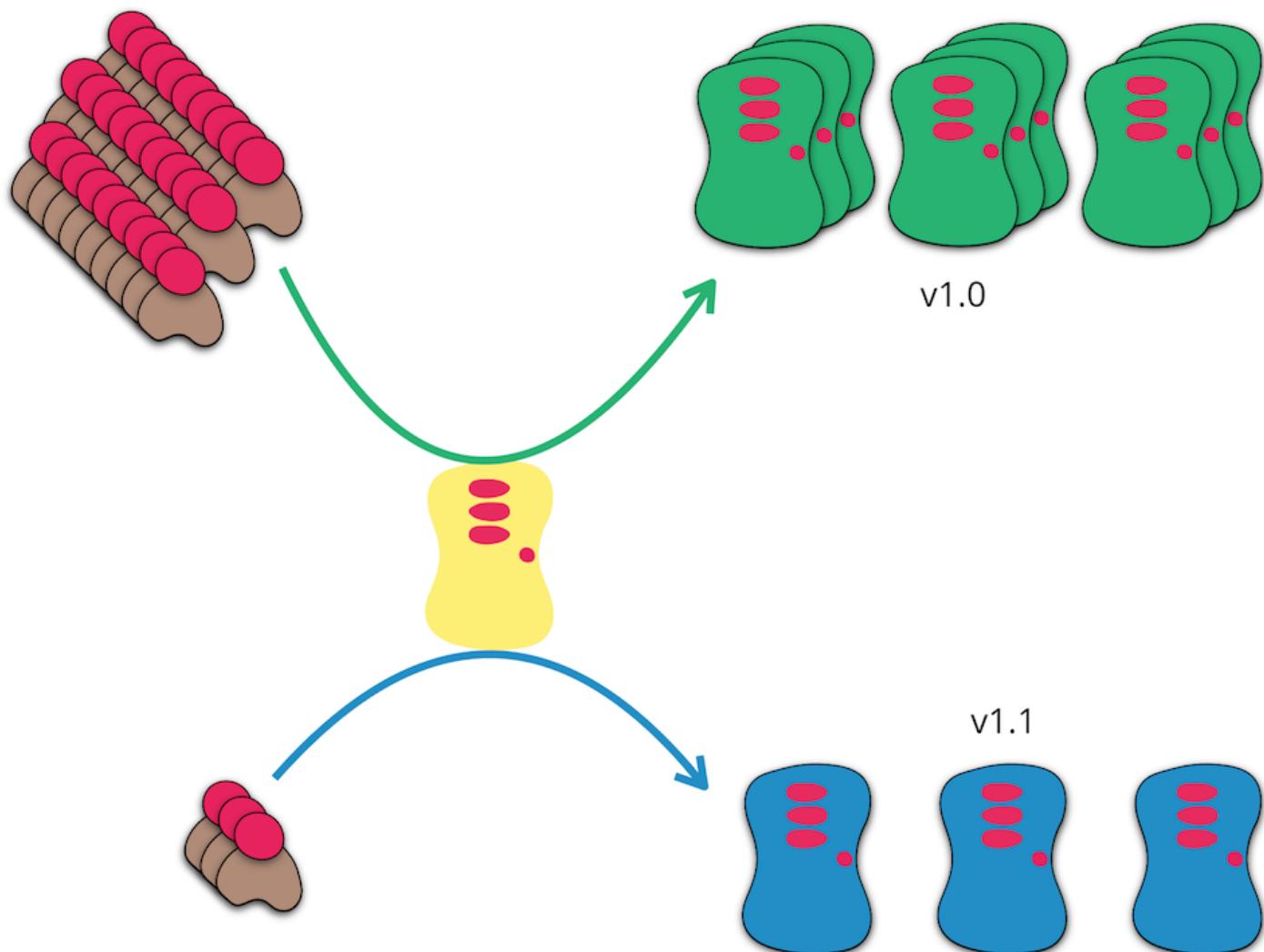
# **ROLLING UPDATES**



# BLUE-GREEN DEPLOYMENT



# CANARY DEPLOYMENT





# **MONITORING IS THE NEW TESTING**

Know when users are experiencing issues in  
**production**

**React to the issues automatically**



**@DEVOPS\_BORAT**

DevOps Borat

To make error is human. To propagate  
error to all server in automatic way is  
**#devops.**

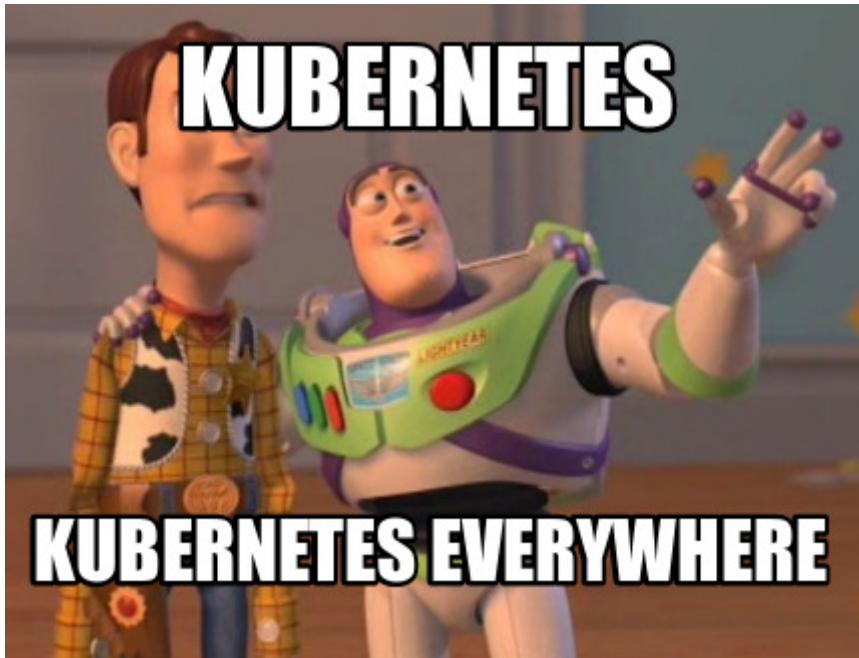
*If you haven't automatically destroyed  
something by mistake, you are not  
automating enough*

JENKINS X



**kubernetes**

**KUBERNETES**





I Am Devloper  
@iamdevloper



## Heisenberg's Uncertainty Principle for software:

The more you try and understand Kubernetes, the more impossible it becomes to do so.

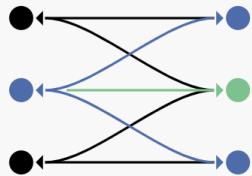
# **PROGRESSIVE DELIVERY WITH JENKINS X**

[jenkins-x.io/developing/progressive-delivery](https://jenkins-x.io/developing/progressive-delivery)



# Istio

Connect, secure, control, and observe services.



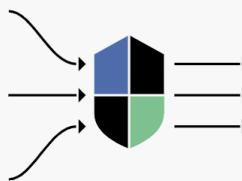
## Connect

Intelligently control the flow of traffic and API calls between services, conduct a range of tests, and upgrade gradually with red/black deployments.



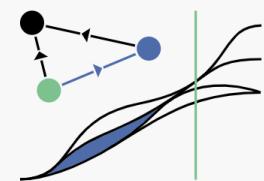
## Secure

Automatically secure your services through managed authentication, authorization, and encryption of communication between services.



## Control

Apply policies and ensure that they're enforced, and that resources are fairly distributed among consumers.



## Observe

See what's happening with rich automatic tracing, monitoring, and logging of all your services.

# PROMETHEUS

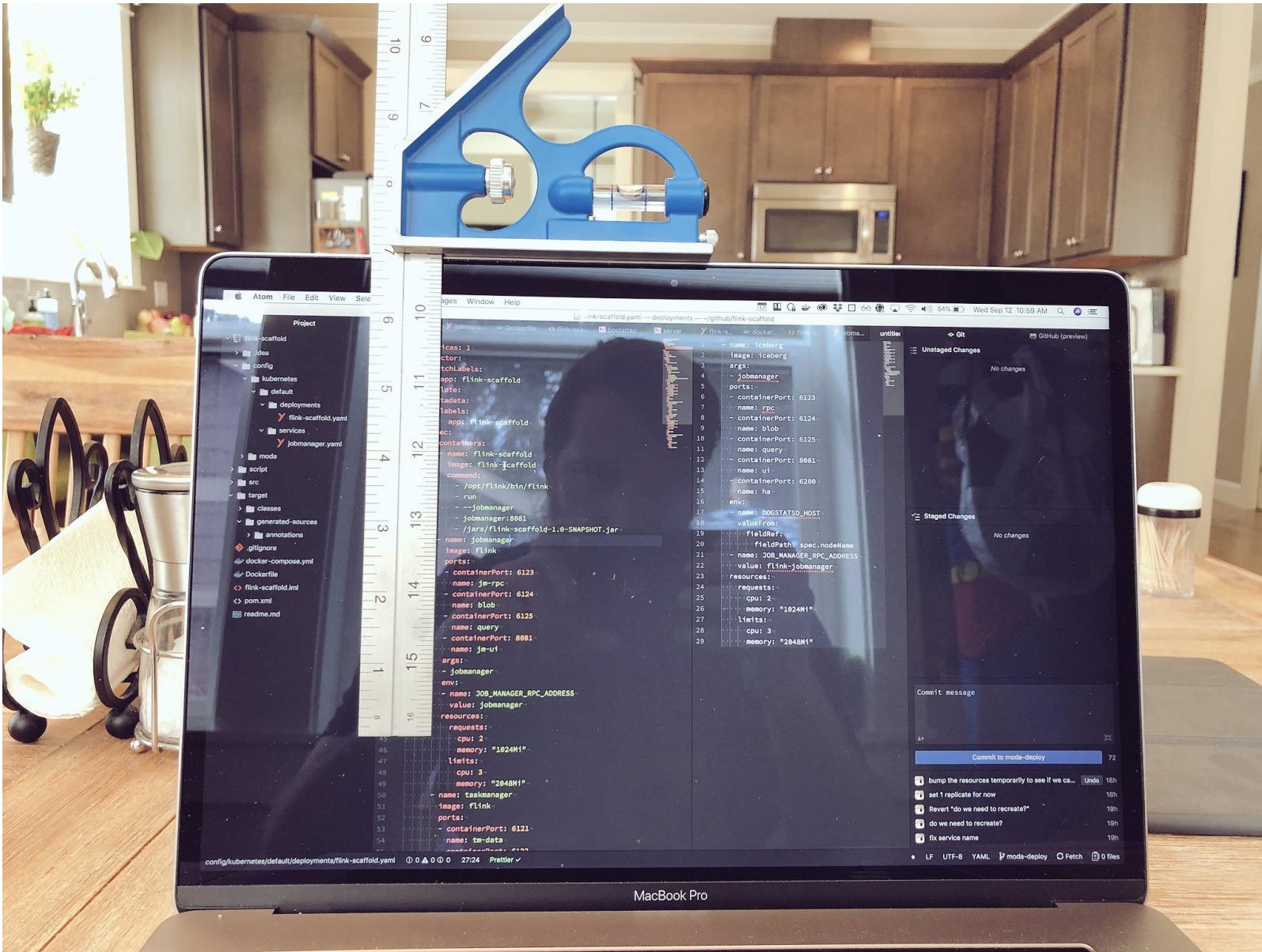


A systems monitoring and alerting toolkit

# FLAGGER

[flagger.app](#)

*automates the promotion of canary deployments by using Istio's traffic shifting and Prometheus metrics to analyse the application's behaviour during a controlled rollout*



# Get the ip of the Istio ingress and point your wildcard domain to it

```
kubectl -n istio-system get service istio-ingressgateway \
-o jsonpath='{.status.loadBalancer.ingress[0].ip}'
```

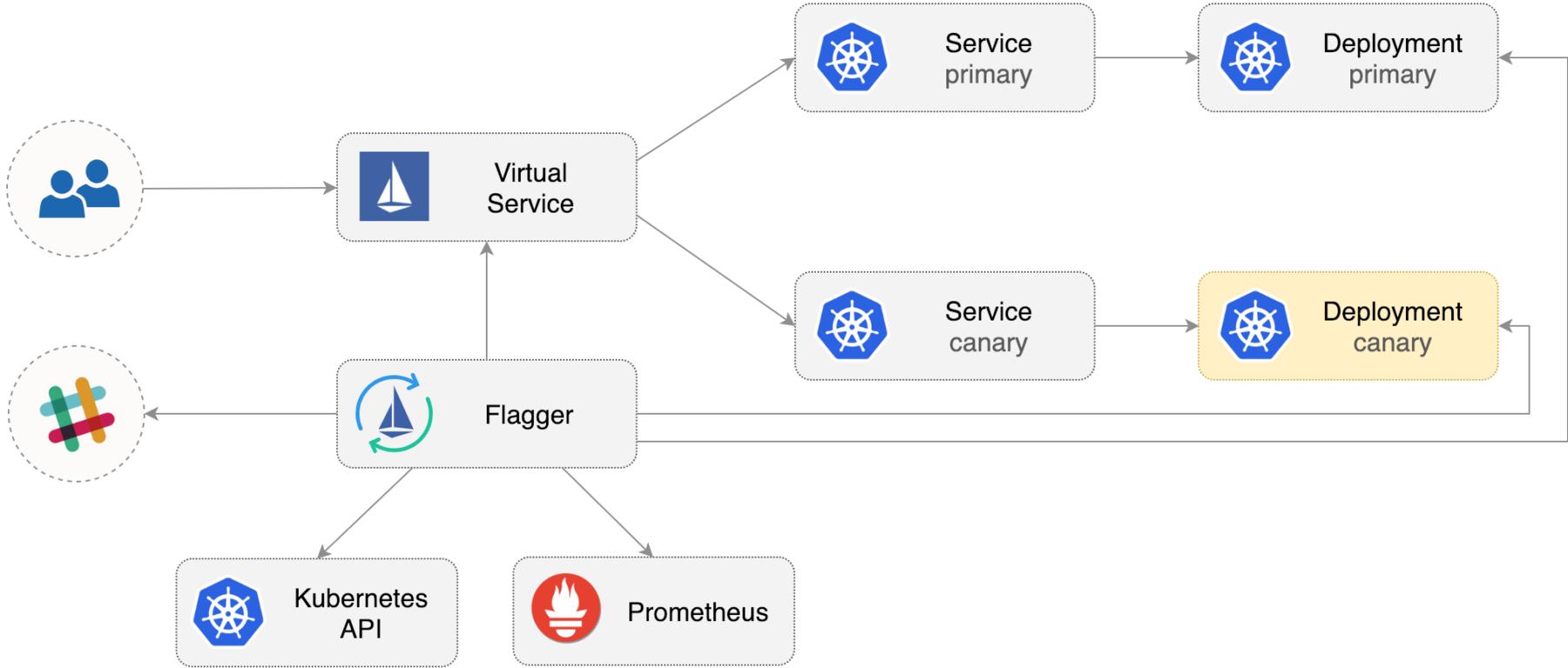
# Add the canary section to our application Helm chart values.yaml

```
...
canary:
  enable: true
  service:
    hosts:
      - croc-hunter.istio.us.g.csanchez.org
    gateways:
      - jx-gateway.istio-system.svc.cluster.local
  canaryAnalysis:
    interval: 60s
    threshold: 5
    maxWeight: 50
    stepWeight: 10
```

```
metrics:
- name: request-success-rate
  # minimum req success rate (non 5xx responses)
  # percentage (0-100)
  threshold: 99
  interval: 60s
- name: request-duration
  # maximum req duration P99
  # milliseconds
  threshold: 500
  interval: 60s
```

# PROFIT!

```
jx promote croc-hunter-java \  
  --version 0.0.130 \  
  --env production
```

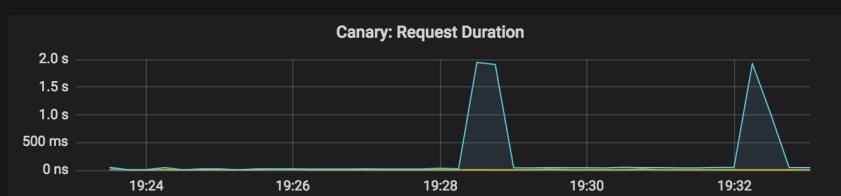
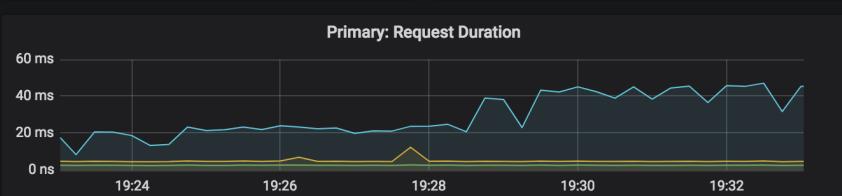


Namespace test ▾

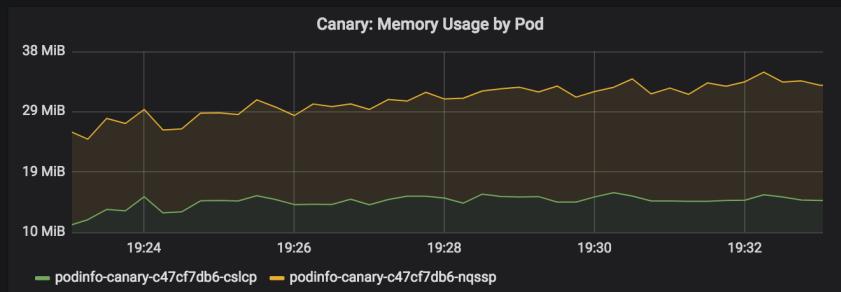
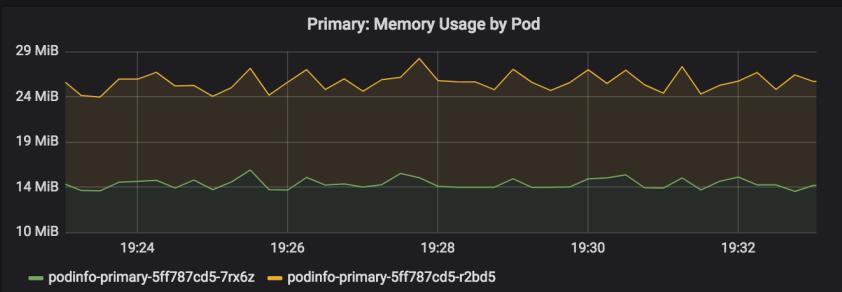
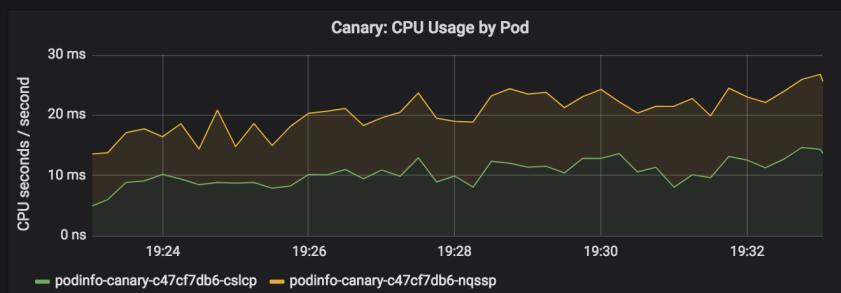
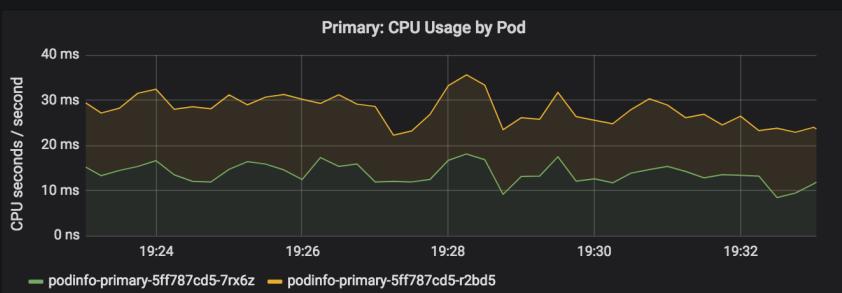
Primary podinfo-primary ▾

Canary podinfo-canary ▾

## RED: podinfo-primary.test



## USE: podinfo-primary.test







**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

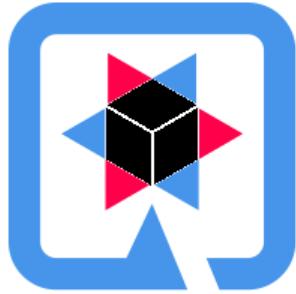
Progress deadline exceeded deployment does not have minimum availability for more than 60s



**flagger** APP 12:18 PM

podinfo.test

Failed checks threshold reached 10



# QUARKUS

[quarkus.io](https://quarkus.io)

A Kubernetes Native Java stack tailored for GraalVM & OpenJDK HotSpot, crafted from the best of breed Java libraries and standards

---

# THE DEVOPS 2.6 TOOLKIT



## Jenkins X

---

Viktor Farcic

CLOUD-NATIVE  
KUBERNETES-FIRST  
CONTINUOUS DELIVERY

[csanchez.org](http://csanchez.org)

