PROGRESSIVE DELIVERY IN KUBERNETES WITH JENKINS X

Carlos Sanchez

csanchez.org / @csanchez

cloudbees


docker


SKAFFOLD

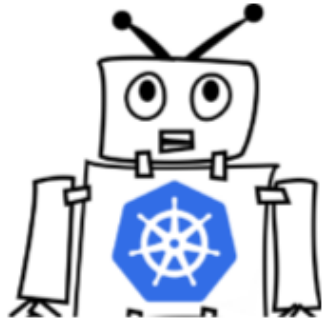

DRAFT


HELM


Nexus


CHARTMUSEUM


k
n



PROGRESSIVE DELIVERY

Progressive Delivery is the next step after Continuous Delivery where new versions are deployed to a subset of users and are evaluated in terms of correctness and performance before rolling them to the totality of the users and rolled back if not matching some key metrics.

Continuous Delivery is hard

Progressive Delivery makes Continuous Delivery easier to adopt

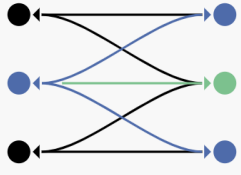
PROGRESSIVE DELIVERY WITH JENKINS X

Install Istio, Prometheus and Flagger

```
jx create addon istio
jx create addon prometheus
jx create addon flagger
```

Istio

Connect, secure, control, and observe services.

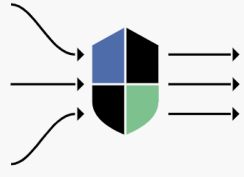


Connect

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

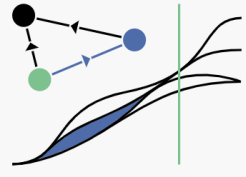
Secure

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.

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 object that will add our deployment to Flagger

```
{{- if eq .Release.Namespace "jx-production" }}
{{- if .Values.canary.enable }}
apiVersion: flagger.app/v1alpha2
kind: Canary
metadata:
  name: {{ template "fullname" . }}
spec:
  targetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: {{ template "fullname" . }}
  progressDeadlineSeconds: 60
  service:
    port: {{.Values.service.internalPort}}
{{- if .Values.canary.service.gateways }}
    gateways:
{{ toYaml .Values.canary.service.gateways | indent 4 }}
{{- end }}
```

```
{{- if .Values.canary.service.hosts }}
    hosts:
{{ toYaml .Values.canary.service.hosts | indent 4 }}
{{- end }}
  canaryAnalysis:
    interval: {{ .Values.canary.canaryAnalysis.interval }}
    threshold: {{ .Values.canary.canaryAnalysis.threshold }}
    maxWeight: {{ .Values.canary.canaryAnalysis.maxWeight }}
    stepWeight: {{ .Values.canary.canaryAnalysis.stepWeight }}
{{- if .Values.canary.canaryAnalysis.metrics }}
    metrics:
{{ toYaml .Values.canary.canaryAnalysis.metrics | indent 4 }}
{{- end }}
{{- end }}
{{- end }}
```

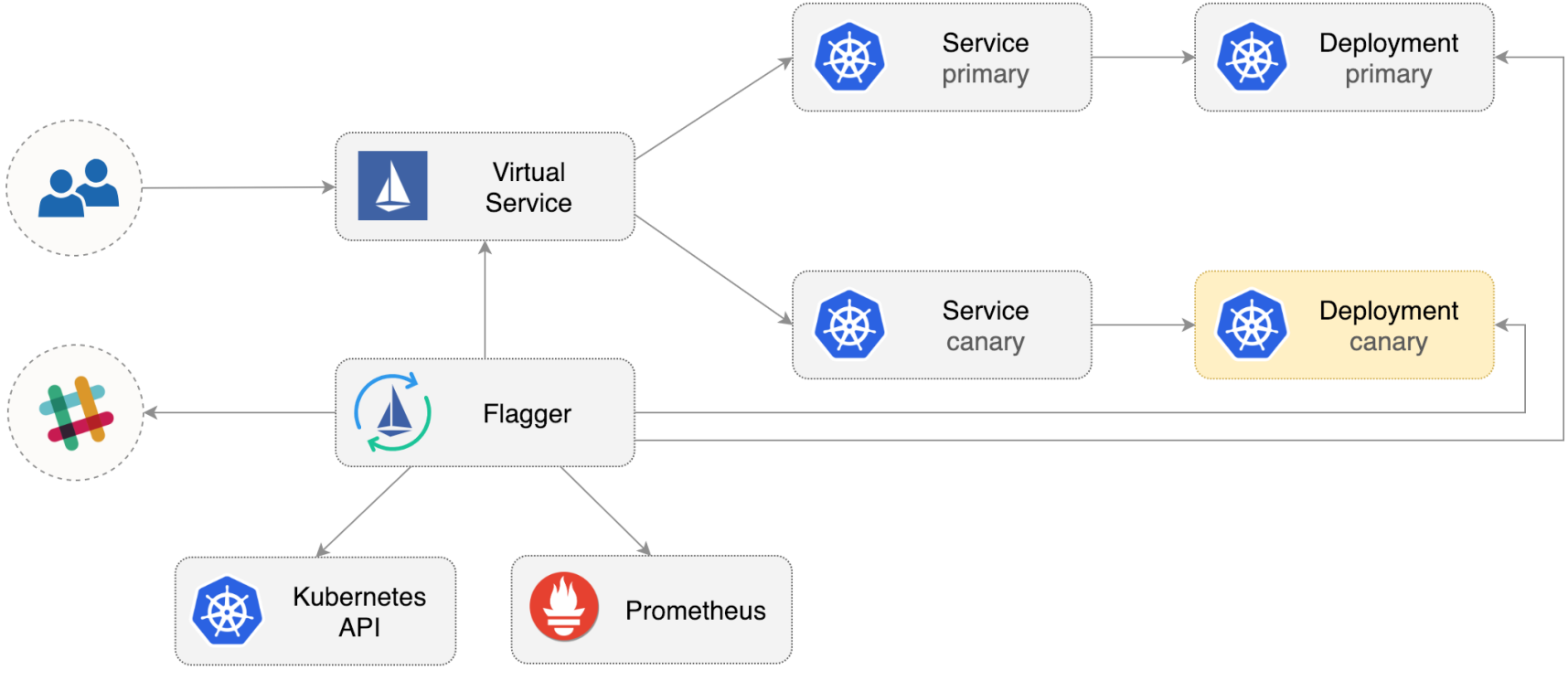
Add the `canary` section to `values.yaml`

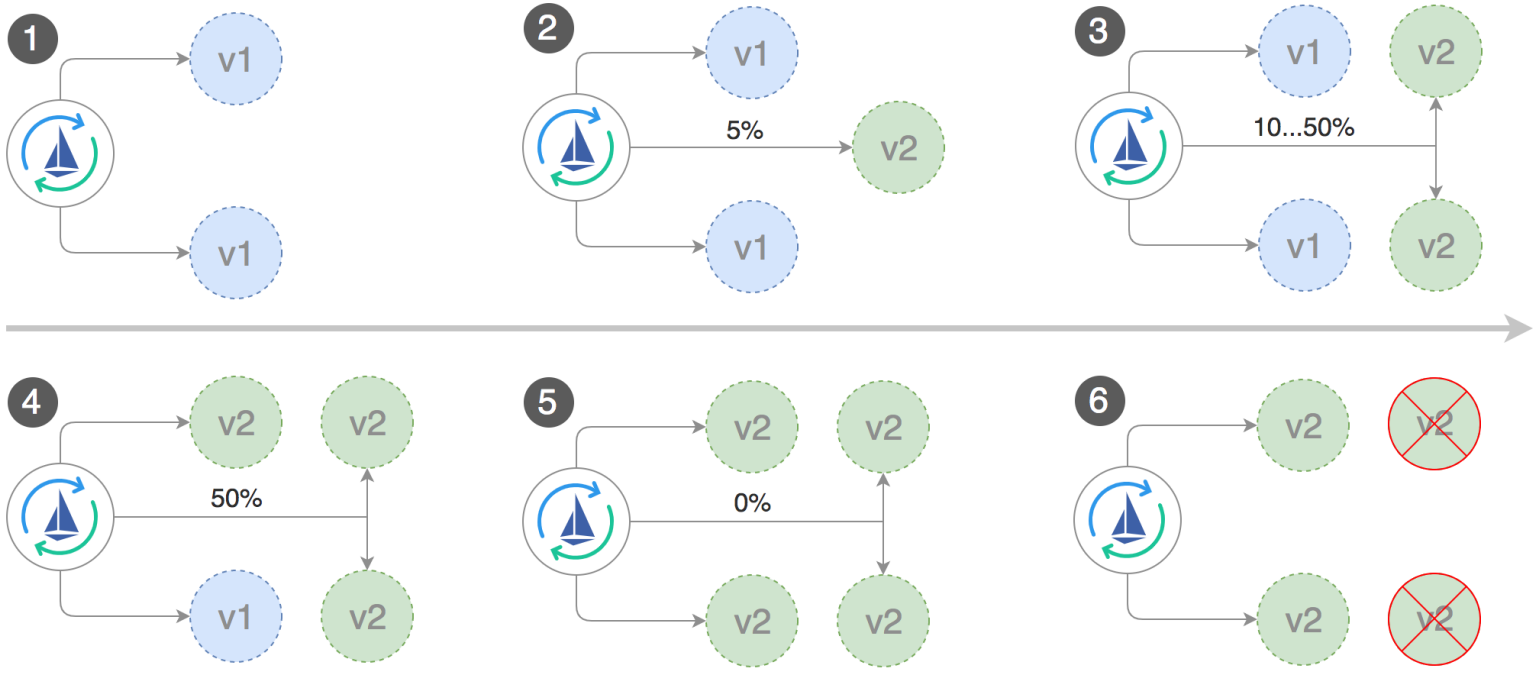
```
...
canary:
  enable: true
  service:
    hosts:
    - croc-hunter.istio.us.g.csanchez.org
    gateways:
    - jx-gateway.istio-system.svc.cluster.local
  canaryAnalysis:
    interval: 60s
    threshold: 5
    maxWeight: 50
    stepWeight: 10
```

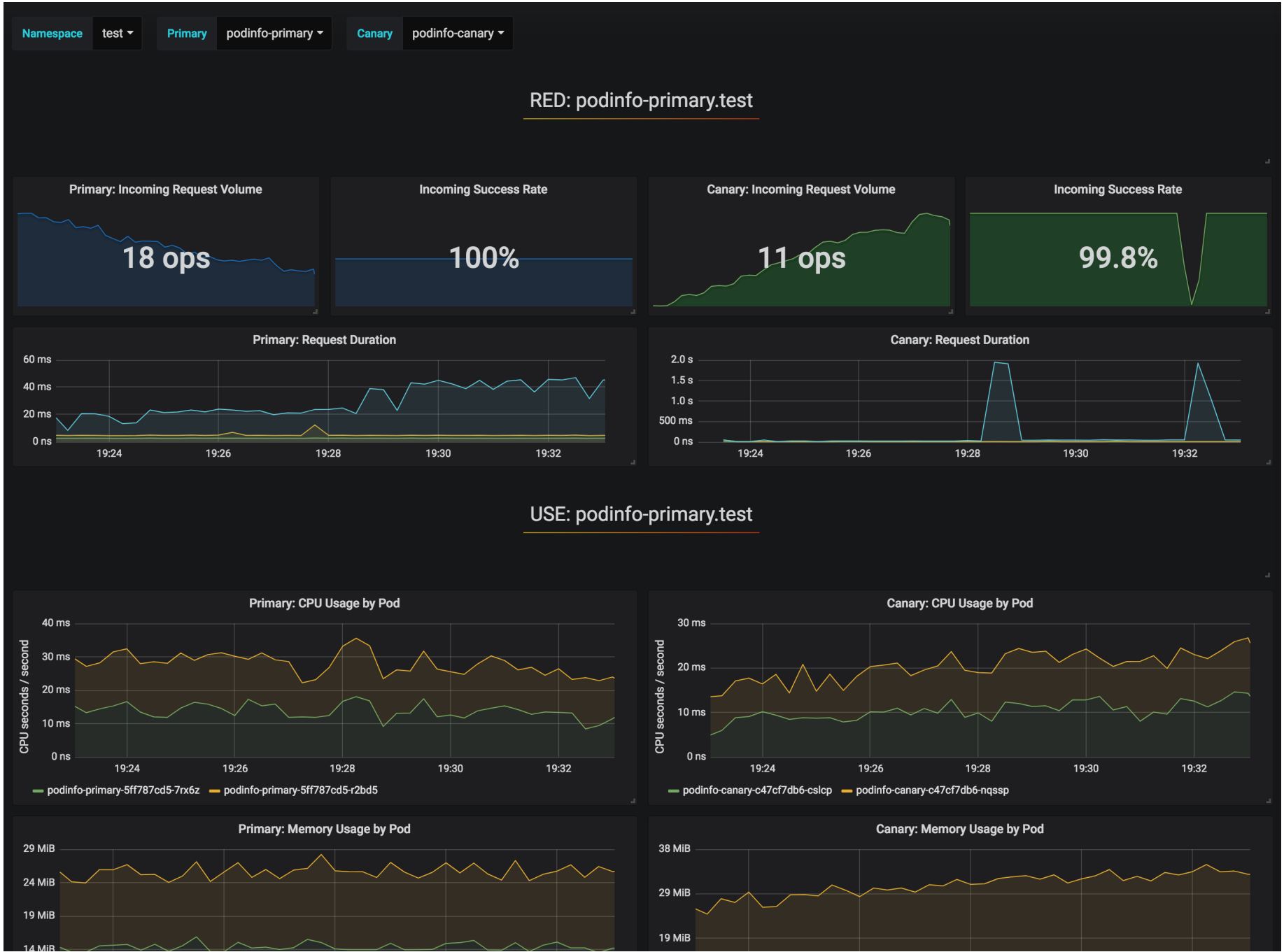
```
    metrics:
    - name: istio_requests_total
      # minimum req success rate (non 5xx responses)
      # percentage (0-100)
      threshold: 99
      interval: 60s
    - name: istio_request_duration_seconds_bucket
      # maximum req duration P99
      # milliseconds
      threshold: 500
     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

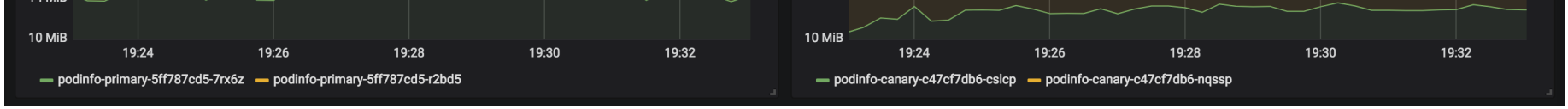
PROFIT!

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


Service
primary
Deployment
primary
Virtual
Service
Service
canary
Deployment
canary
Flagger
Kubernetes
API
Prometheus


1
v1
v1
2
v1
5%
v2
v1
3
v1
v2
10...50%
v1
v2
4
v2
v2
50%
v1
v2
5
v2
v2
0%
v2
v2
6
v2
v2
v2
v2


Namespace
test
Primary
podinfo-primary
Canary
podinfo-canary
RED: podinfo-primary.test
Primary: Incoming Request Volume
18 ops
Incoming Success Rate
100%
Canary: Incoming Request Volume
11 ops
Incoming Success Rate
99.8%
Primary: Request Duration
60 ms
40 ms
20 ms
0 ns
19:24
19:26
19:28
19:30
19:32
Canary: Request Duration
2.0 s
1.5 s
1.0 s
500 ms
0 ns
19:24
19:26
19:28
19:30
19:32
USE: podinfo-primary.test
Primary: CPU Usage by Pod
CPU seconds / second
40 ms
30 ms
20 ms
10 ms
0 ns
19:24
19:26
19:28
19:30
19:32
podinfo-primary-5ff787cd5-7rx6z
podinfo-primary-5ff787cd5-r2bd5
Canary: CPU Usage by Pod
CPU seconds / second
30 ms
20 ms
10 ms
0 ns
19:24
19:26
19:28
19:30
19:32
podinfo-canary-c47cf7db6-cslcp
podinfo-canary-c47cf7db6-nqssp
Primary: Memory Usage by Pod
29 MiB
24 MiB
19 MiB
14 MiB
Canary: Memory Usage by Pod
38 MiB
29 MiB
19 MiB


10 MiB
19:24
19:26
19:28
19:30
19:32
podinfo-primary-5ff787cd5-7rx6z
podinfo-primary-5ff787cd5-r2bd5
10 MiB
19:24
19:26
19:28
19:30
19:32
podinfo-canary-c47cf7db6-cslcp
podinfo-canary-c47cf7db6-nqssp

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

carlossg/croc-hunter-jenkinsx-serverless

csanchez.org

csanchez

carlossg

cloudbees