ZERO DOWNTIME DEPLOYMENT STRATEGIES WITH K8S AND CLOUDNATIVE

Wojciech Barczyński [Head of Engineering]
SMACC.io | Hypatos.ai
4Developers, Warsaw, 8 April 2019

:?

- Wojciech Barczyński
- Software Developer & System Engineer
- Machine-Learning FinTech API and SaaS product
- Since 2016, k8s startup practionier

Github: wojciech12 | Linked: IN | HP: wbarczynski.pl | T: @wbarczynski

WHAT WE KNOW

- Release often
- Release in small batches
- Let the business evaluate early
- Take and pay back the tech debts

WHAT WE KNOW

- Some bugs, you see in production
- Some features, the business can verify in production [1]

[1] all is an ongoing discovery

Once upon a time 1

- an upgrade
- but the customers
- you, a dark office, 20:00 on Friday,
  5 min to deploy

Once upon a time 2

- was an error
- 4:00 AM Saturday,
  the first light and birds singing

ZERO-DOWNTIME DEPLOYMENTS STRATEGIES

Black (Blue) Box

Infrastructure (almost) invisible

Easy* Continuous Deployment

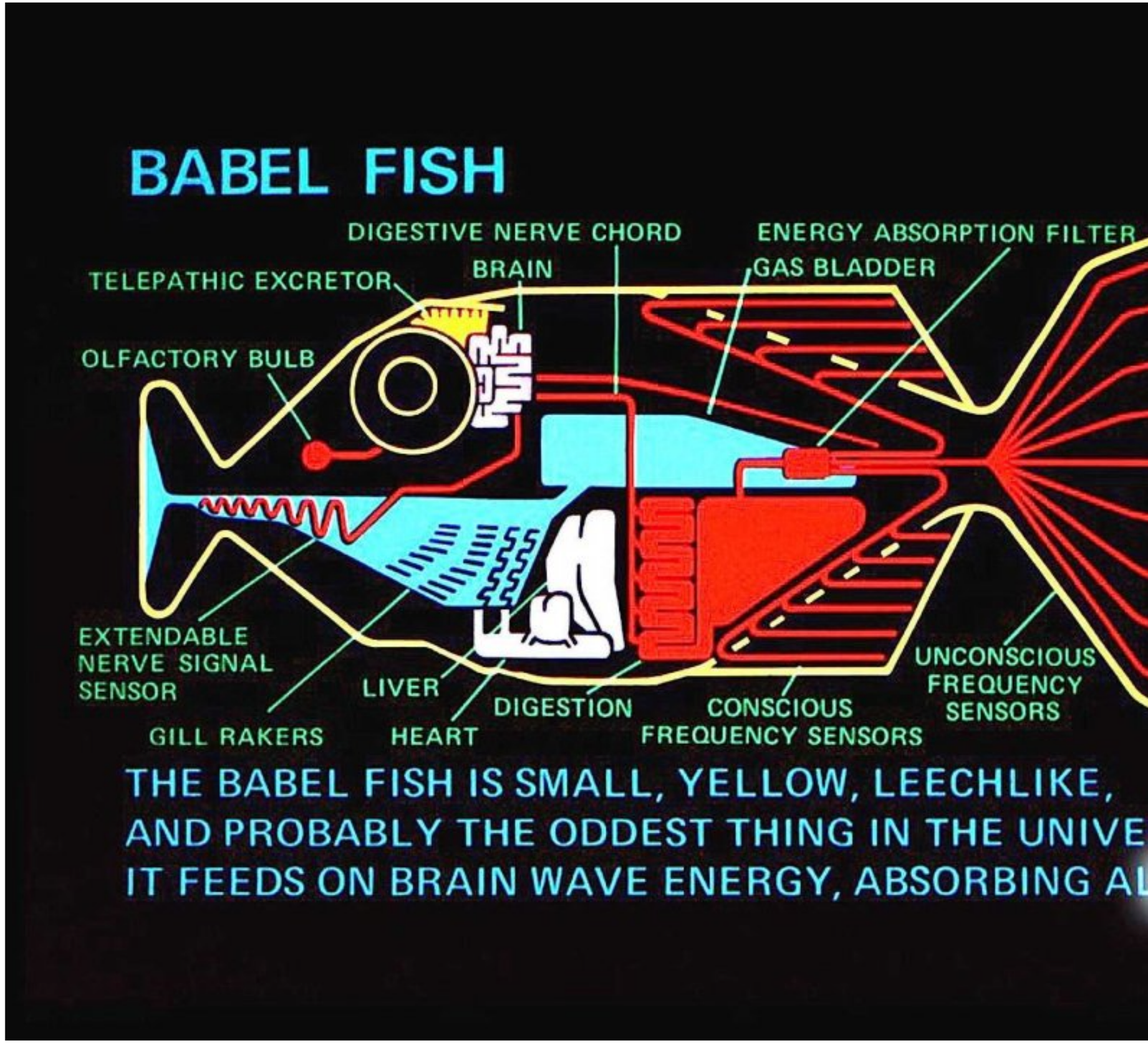
https://en.wikipedia.org/wiki/File:Dr_Who_(316350537).jpg

Common

Language

Artifacts

Platform

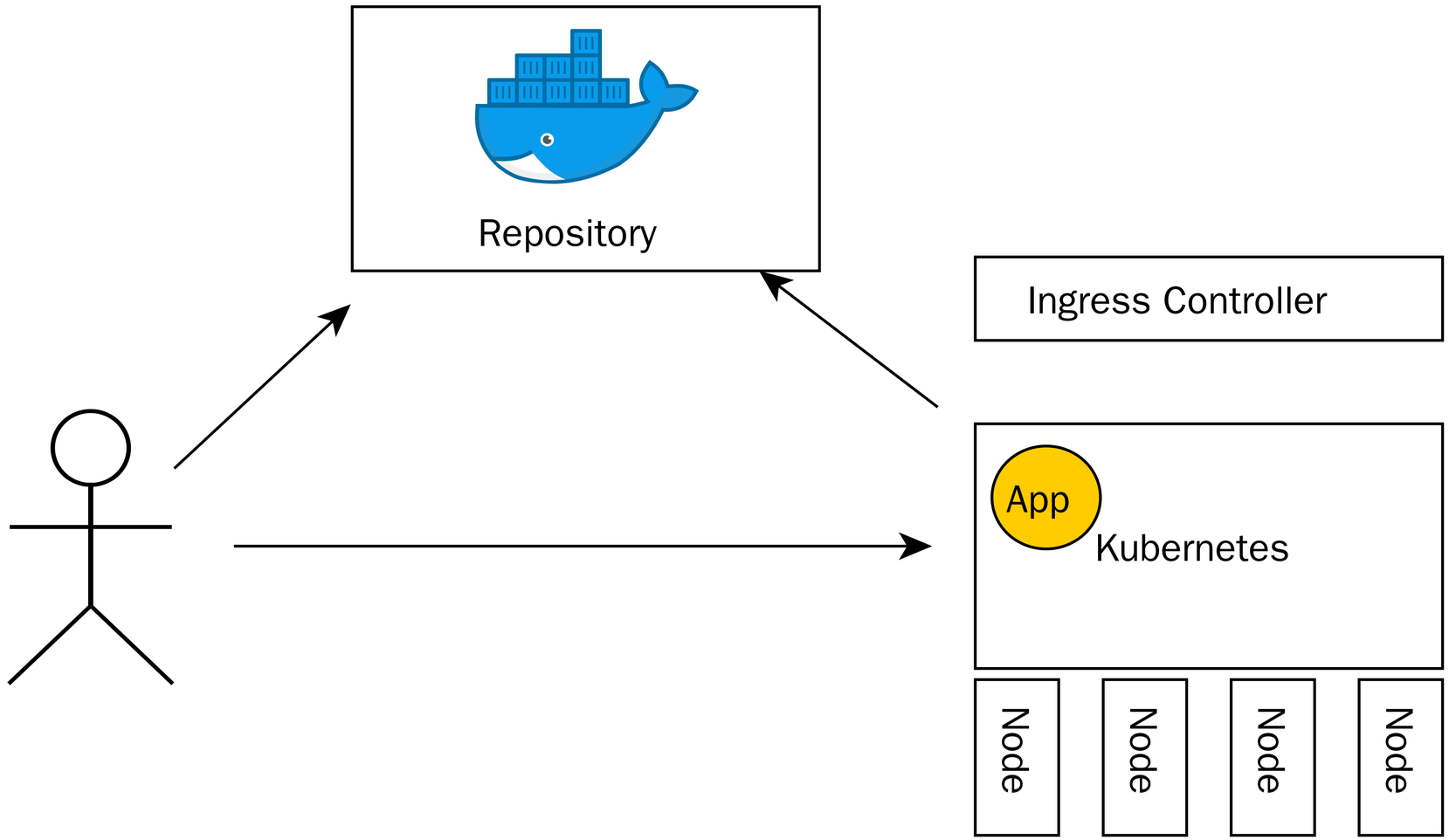


KUBERNETES

- Orchiestrator of your applications packed in Dockers
- Scale, move, migrate, load-balances...
- Service-discovery? Check!

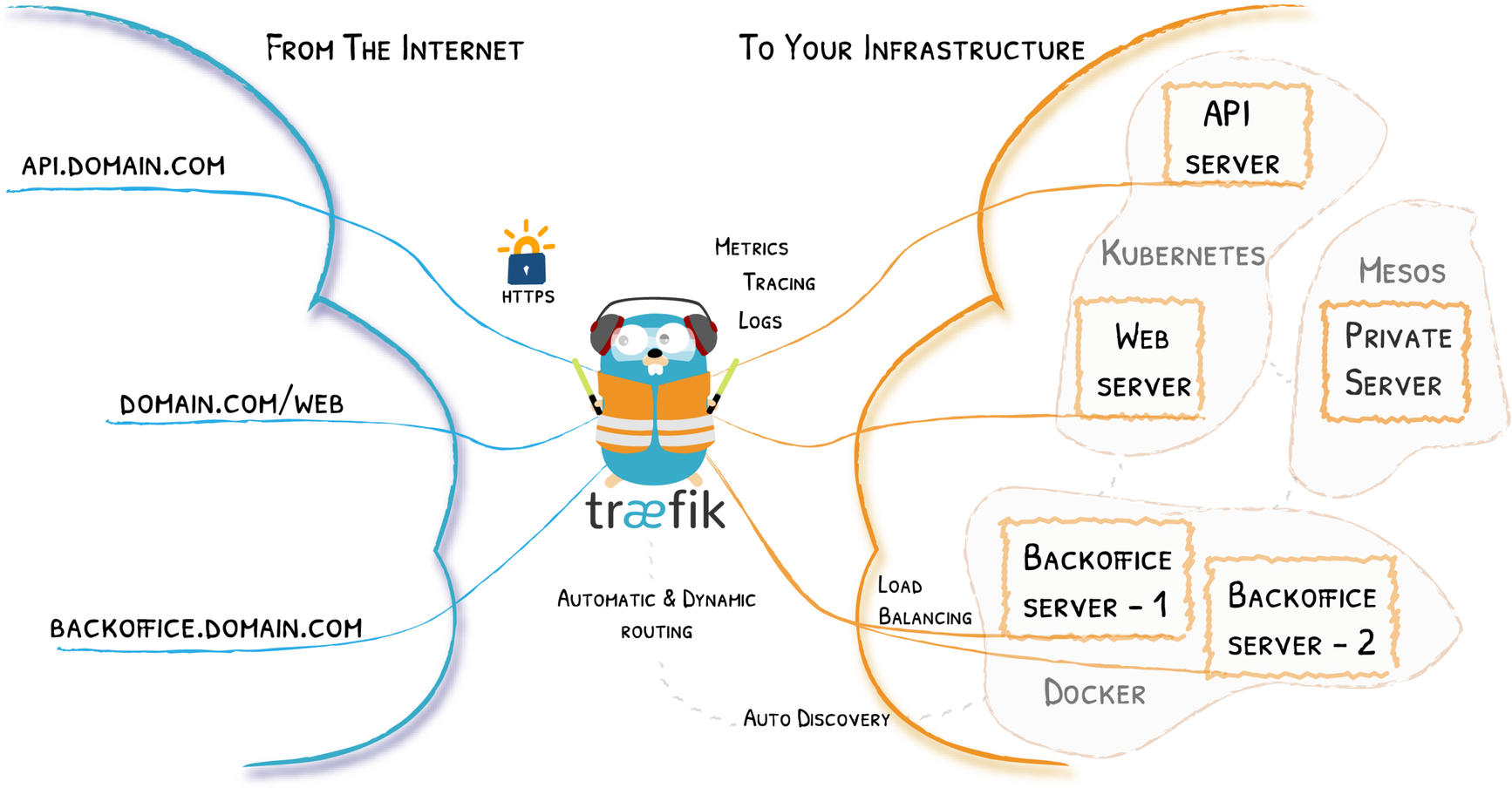
Operation-system, Application server, New linux.

KUBERNETES




```
make docker_push; kubectl create -f app-srv-dpl.yaml
```

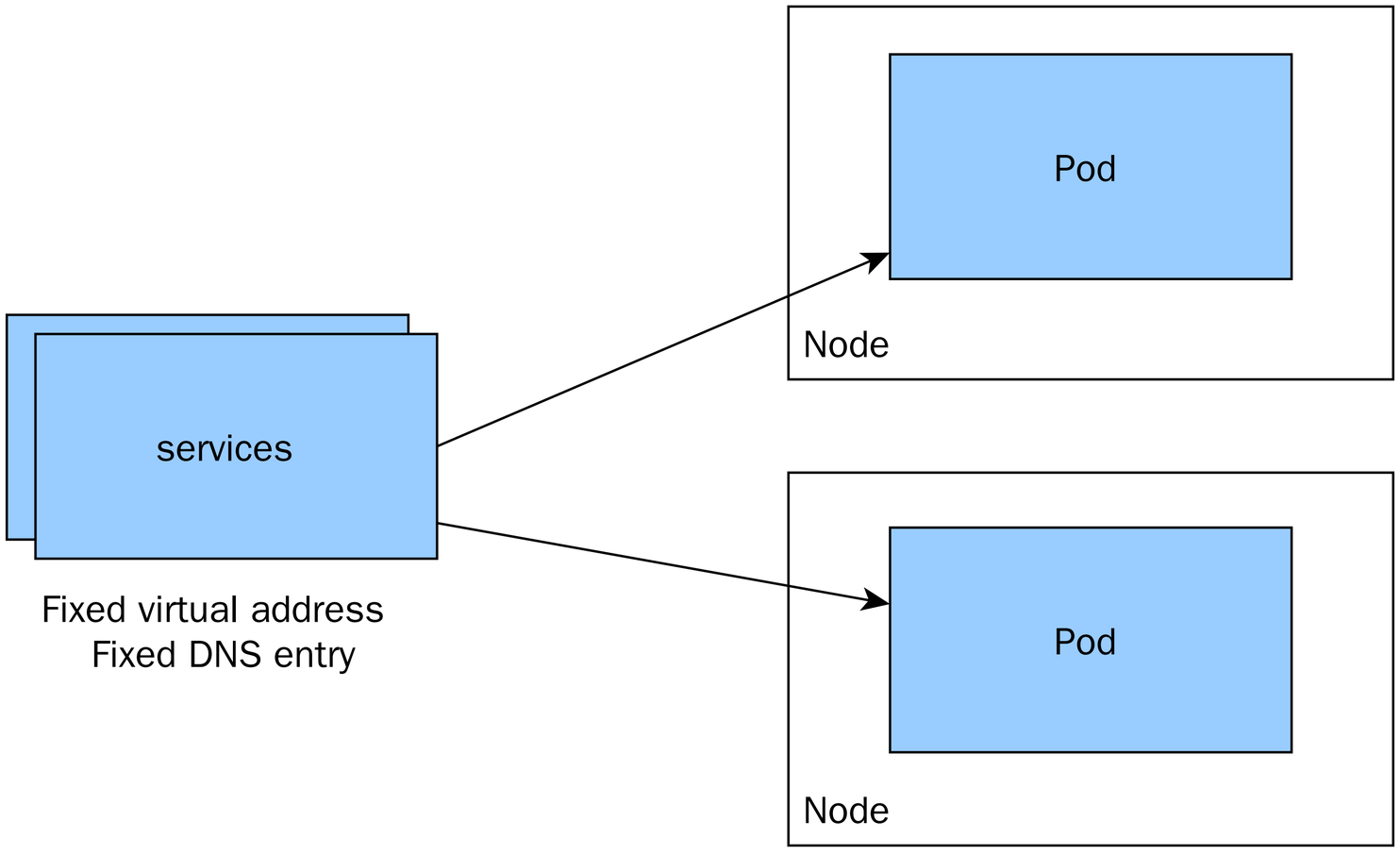
INGRESS


From The Internet
To Your Infrastructure
API.DOMAIN.COM
HTTPS
Metrics
Tracing
Logs
DOMAIN.COM/WEB
træfik
BACKOFFICE.DOMAIN.COM
Automatic & Dynamic routing
Load Balancing
Auto Discovery
API server
Kubernetes
Web server
Mesos
Private Server
Backoffice server - 1
Backoffice server - 2
Docker

INGRESS

Pattern	Service
api.hypatos.ai/v1/users	users-v1
api.hypatos.ai/v2/users	users-v2
smacc.io	website

SERVICE TO PODS



Service matches **labeled** pods

Service

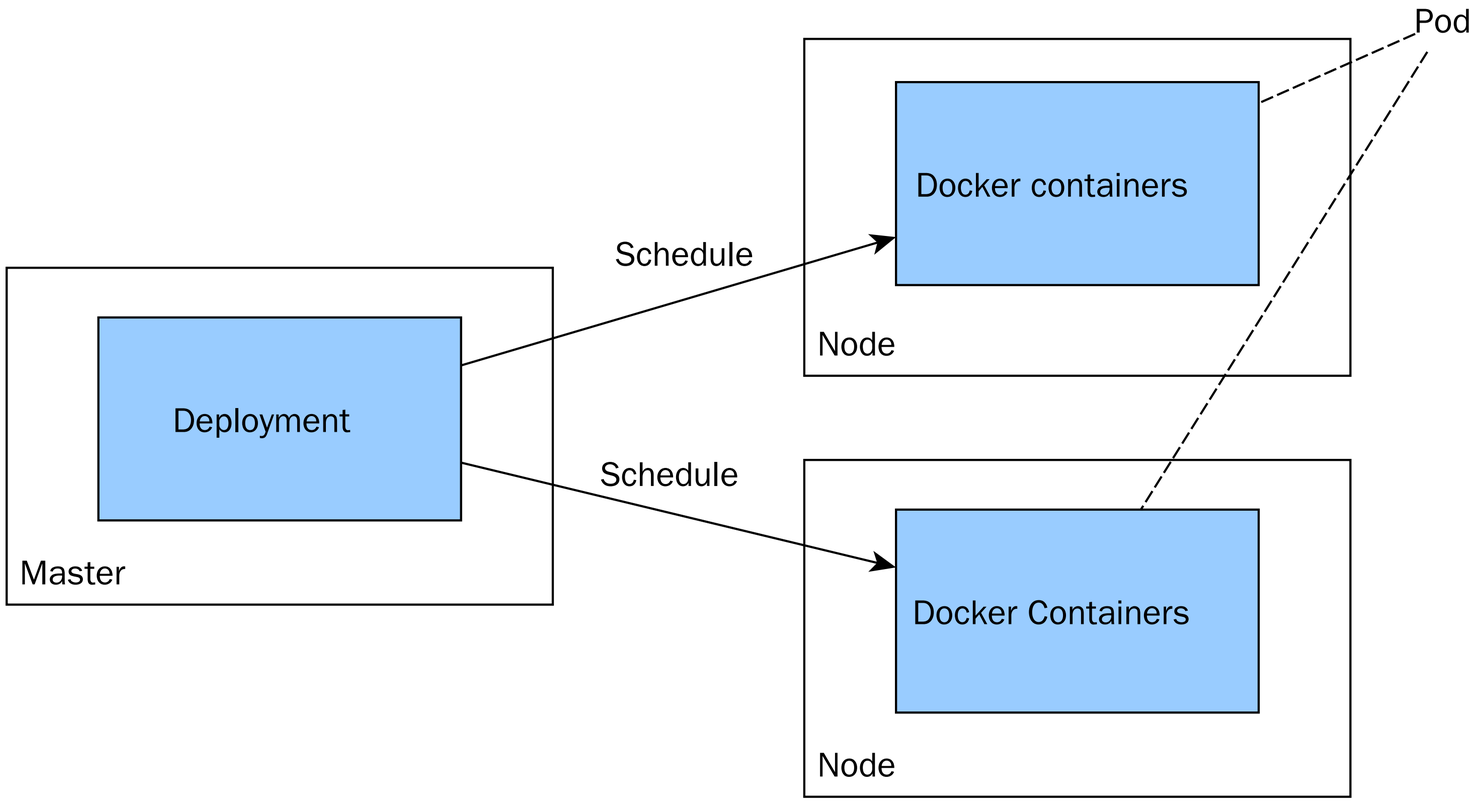
`curl http://users-v1/list`

Load balanced inside cluster!

service.yml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: website
spec:
  ports:
  - port: 80
    protocol: TCP
  selector:
    app: website-smacc
  type: LoadBalancer
```

DEPLOYMENT AND PODS



deployment.yml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: demo-api
  labels:
    app: demo-api
spec:
  replicas: 3
  strategy:
    type: Recreate
  selector:
    matchLabels:
      app: demo-api
  template:
```

BASIC CONCEPTS

Name	Purpose	
Service	Interface	Entry point (Service Name)
Deployment	Factory	How many pods, which pods
Pod	Implementation	1+ docker running

DEPLOYMENT STRATEGIES

STRATEGIES

We will see:

- Replace (downtime visible)
- Rolling updates
- Blue Green
- Canary

STRATEGIES

We will not cover in demo:

- Shadow deployment
- Synthetic canary [1]
- Feature toggles (also as an A/B)

[1] blog post on ambassador.io, see also diferencia

FIRST THE HOMEWORK

Need to support:

- liveness - am I dead?
- readiness - can I serve requests?

and monitoring for 4 golden signals/RED.

K8S LIVENESS PROBE

```yaml
livenessProbe:
  httpGet:
    path: /model
    port: 8000
    httpHeaders:
      - name: X-Custom-Header
        value: Awesome
  initialDelaySeconds: 600
  periodSeconds: 5
  timeoutSeconds: 18
  successThreshold: 1
  failureThreshold: 3
```

LIVENESS PROBE

- our pod gets restarted
- too many restarts -> CrashLoop

K8S READINESS PROBE

```
readinessProbe:
  exec:
    command:
    - cat
     - /tmp/healthy
  initialDelaySeconds: 5
  periodSeconds: 5
```

YOUR APP SHOULD ON STOP

1. we get SIGTERM signal

YOUR APP SHOULD ON STOP

1. we get SIGTERM signal
2. app SHOULD give 500 (readinessProbe)

YOUR APP SHOULD ON STOP

1. we get SIGTERM signal
2. app SHOULD give 500 (readinessProbe)
3. k8s does not send more requests

YOUR APP SHOULD ON STOP

1. we get SIGTERM signal
2. app SHOULD give 500 on readinessProbe
3. k8s does not send more requests
4. app shuts down gracefully

YOUR APP SHOULD ON STOP

1. we get SIGTERM signal
2. app gives 500 on readinessProbe
3. k8s does not send more requests
4. app shuts down gracefully
5. kubernetes forces kill if 30s limit exceeded

ALWAYS

Implement readiness for:

- ML model-based components
- slow starting time

DEMO SERVICE IMPLEMENTATION

- graceful shutdown
- demo service

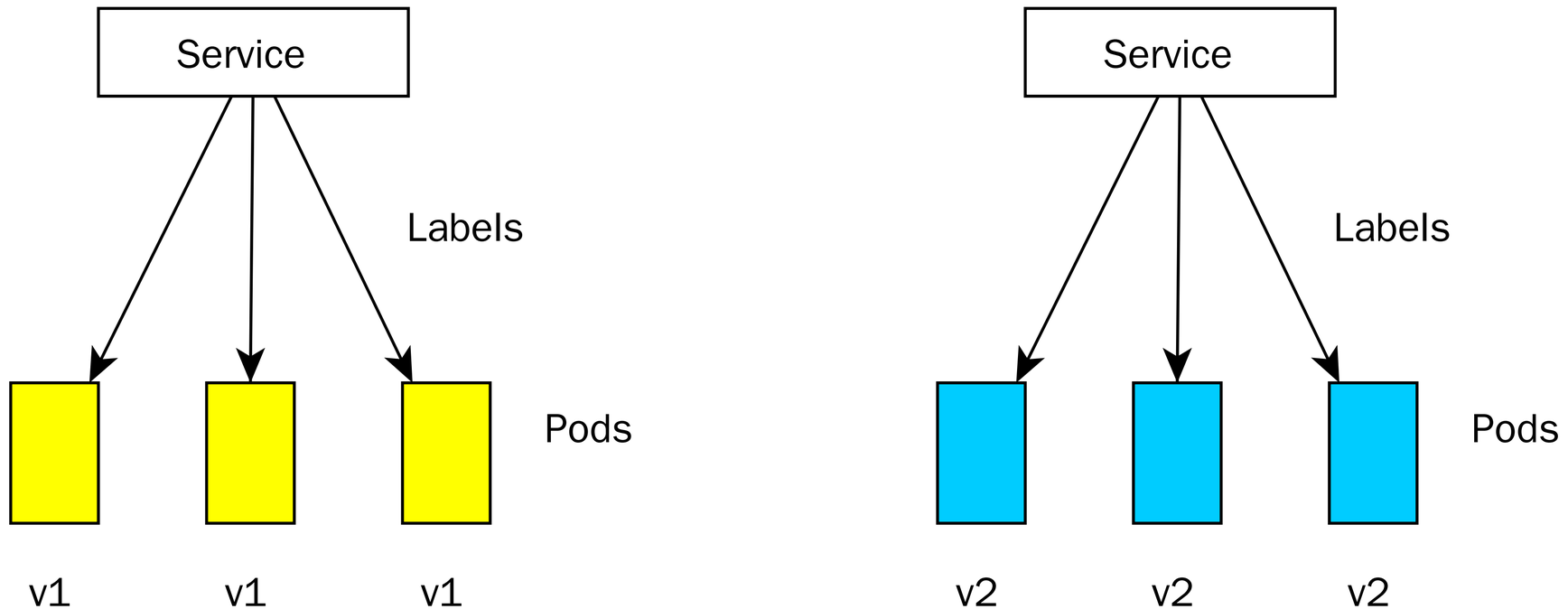
DotNet? See demo_net/ZeroDowntimeDeployment

GRACEFUL SHUTDOWN

From missy:

```
func (s *Service) prepareShutdown(h Server) {
    signal.Notify(s.Stop, os.Interrupt, syscall.SIGTERM)
    <-s.Stop
    s.StatusNotReady()
    shutdown(h)
}
```

DEMO - RECREATE



DEMO - RECREATE

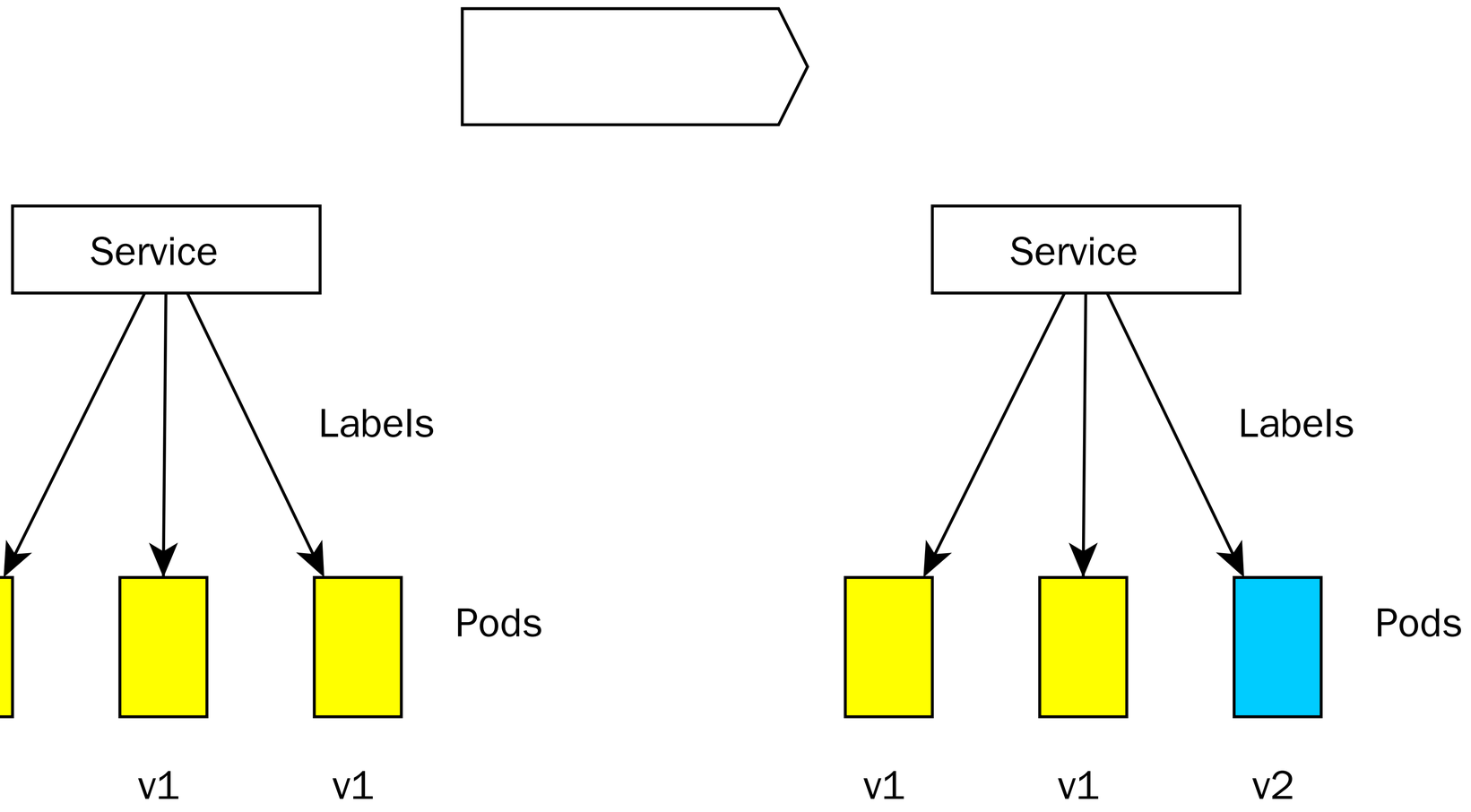
```
spec:
  replicas: 3
  strategy:
    type: Recreate
```

```
kubectl set image  deployment/demo-api \
    app=wojciech11/api-status:2.0.0
```

DEMO - RECREATE

- quick
- downtime visible

DEMO - ROLLING UPDATES



DEMO - ROLLING UPDATES

```
strategy:
  type: RollingUpdate
  rollingUpdate:
    maxSurge: 2
    maxUnavailable: 0
```

docs

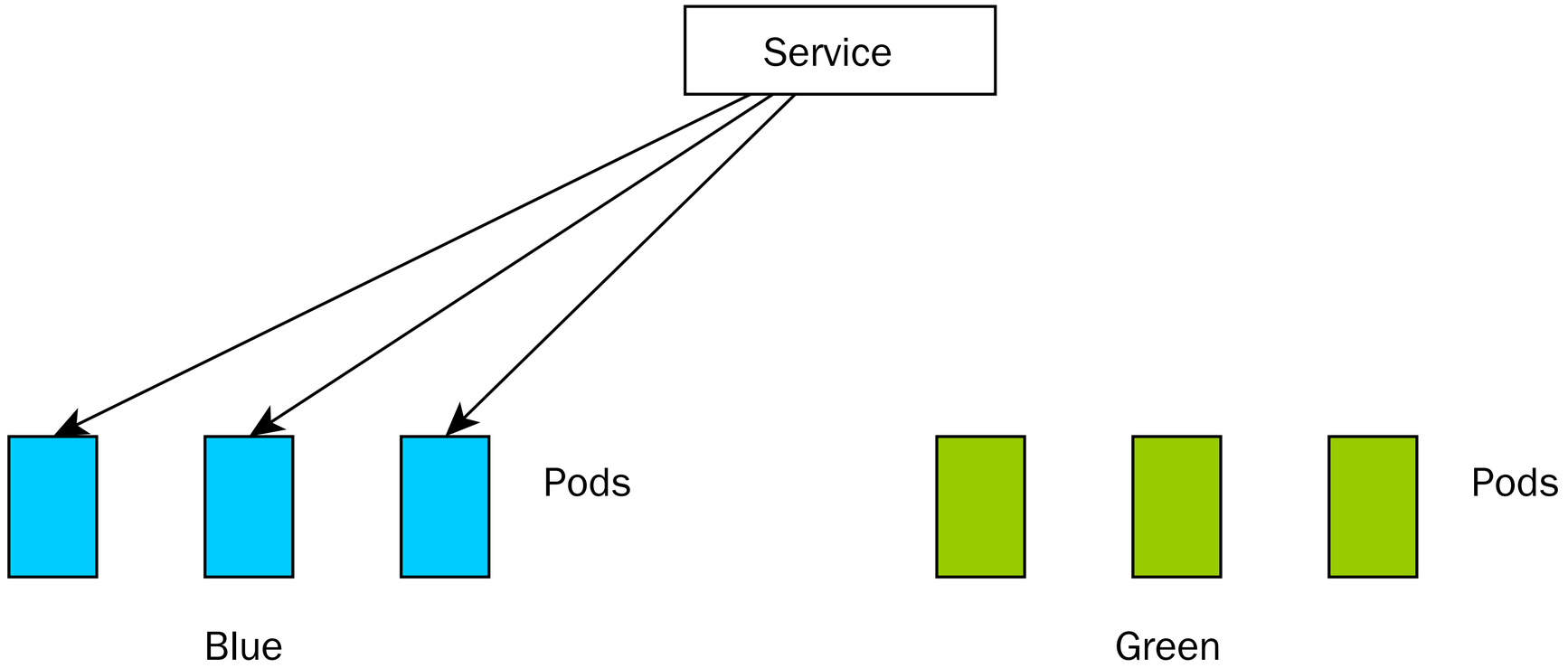
DEMO - ROLLING UPDATES

```
kubectl set image  deployment/demo-api
    app=wojciech11/api-status:2.0.0
```

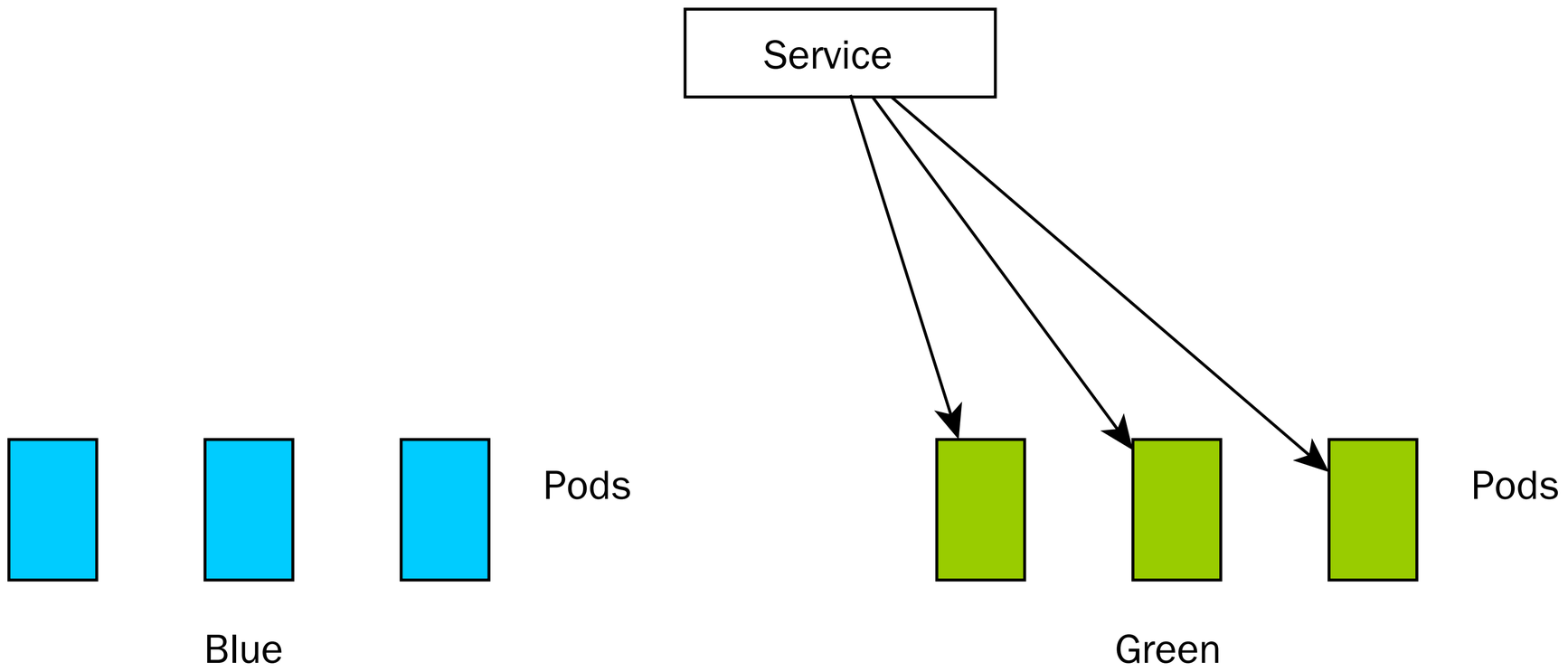
DEMO - ROLLING UPDATES

- the most popular

DEMO - GREEN/BLUE



DEMO - GREEN/BLUE



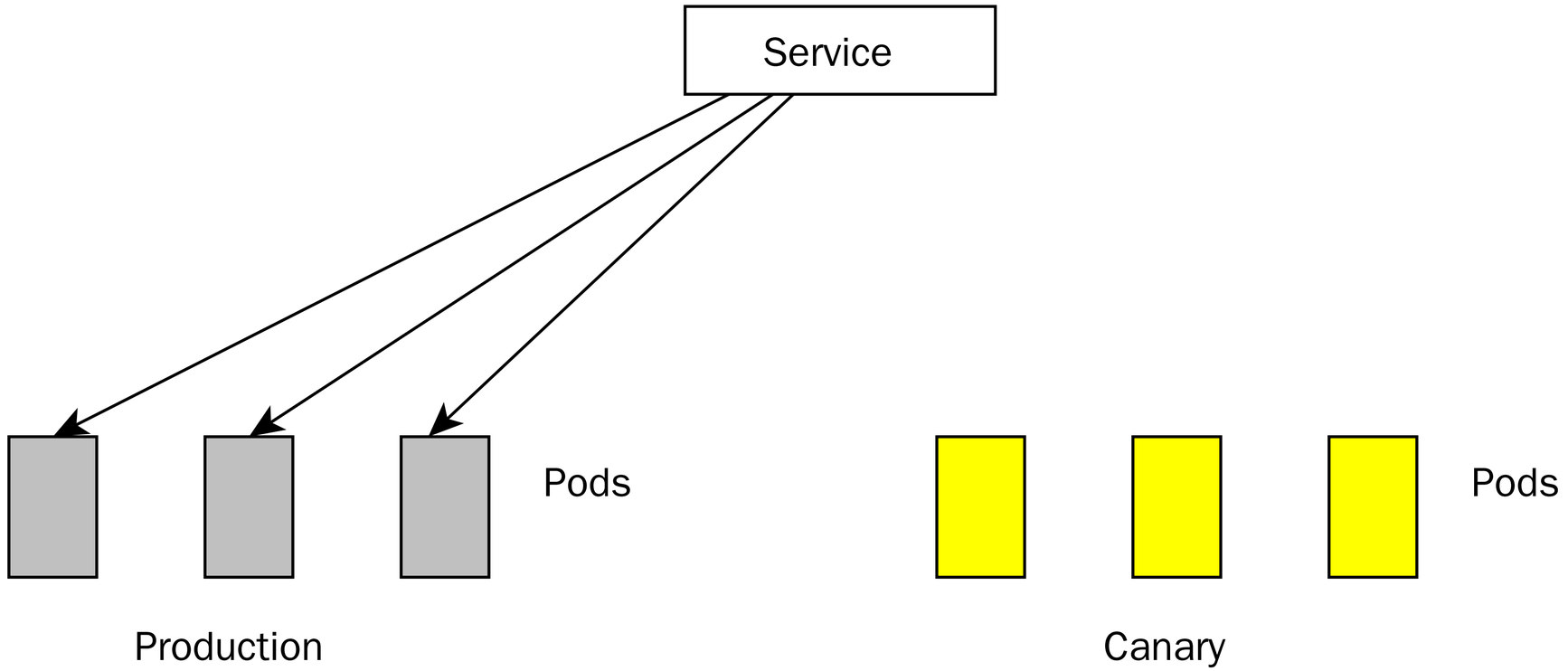
DEMO - GREEN/BLUE

```
kubectl patch service api-status \
    -p '{"spec":{"selector": {"label": "green"} }}'
```

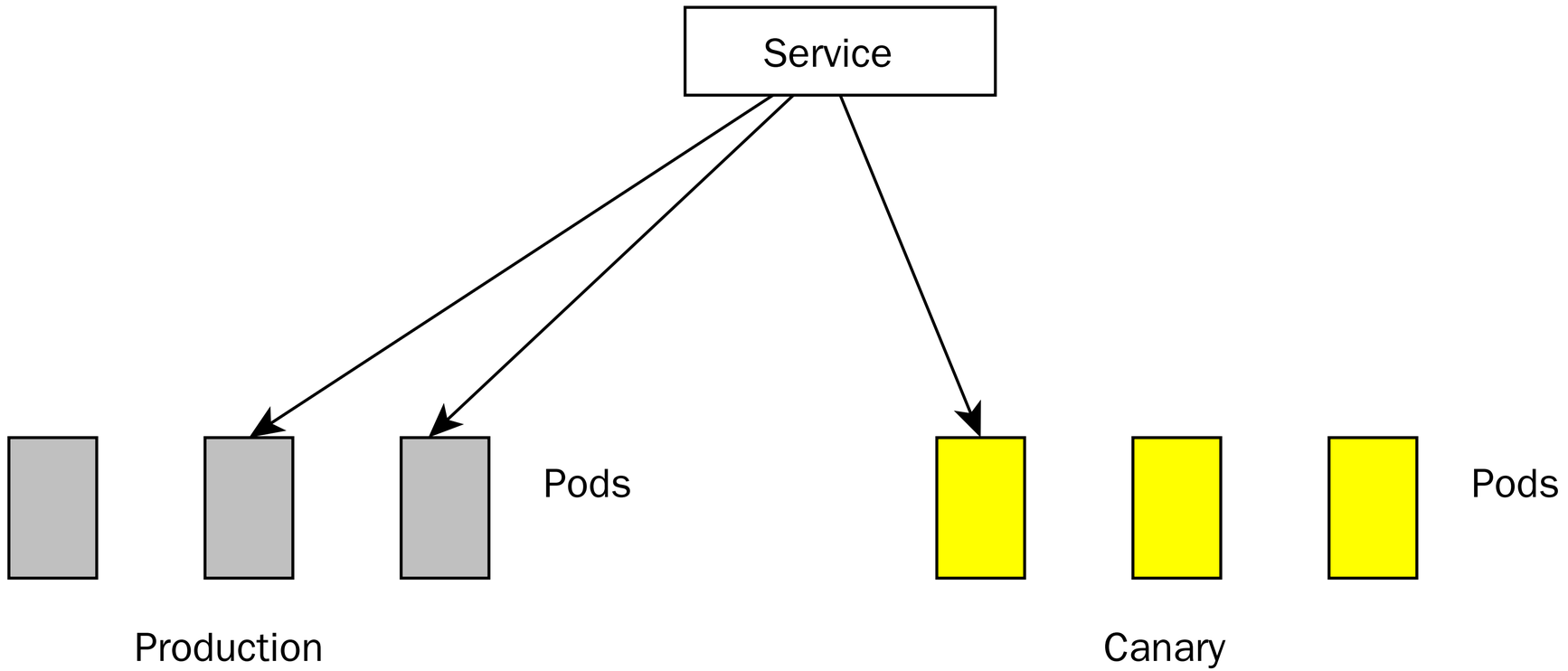
DEMO - GREEN/BLUE

- For big changes
- Less common
- Might be implemented with *Ingress*

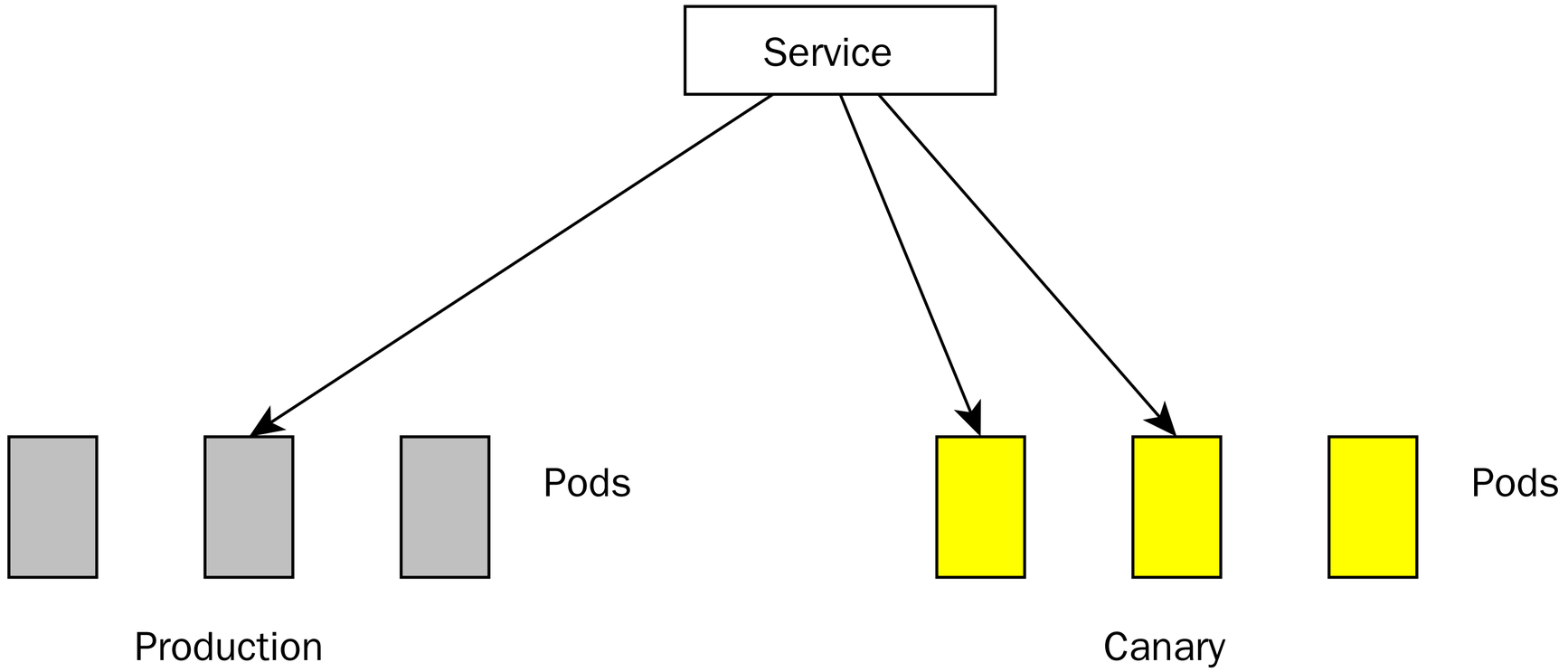
DEMO - CANARY



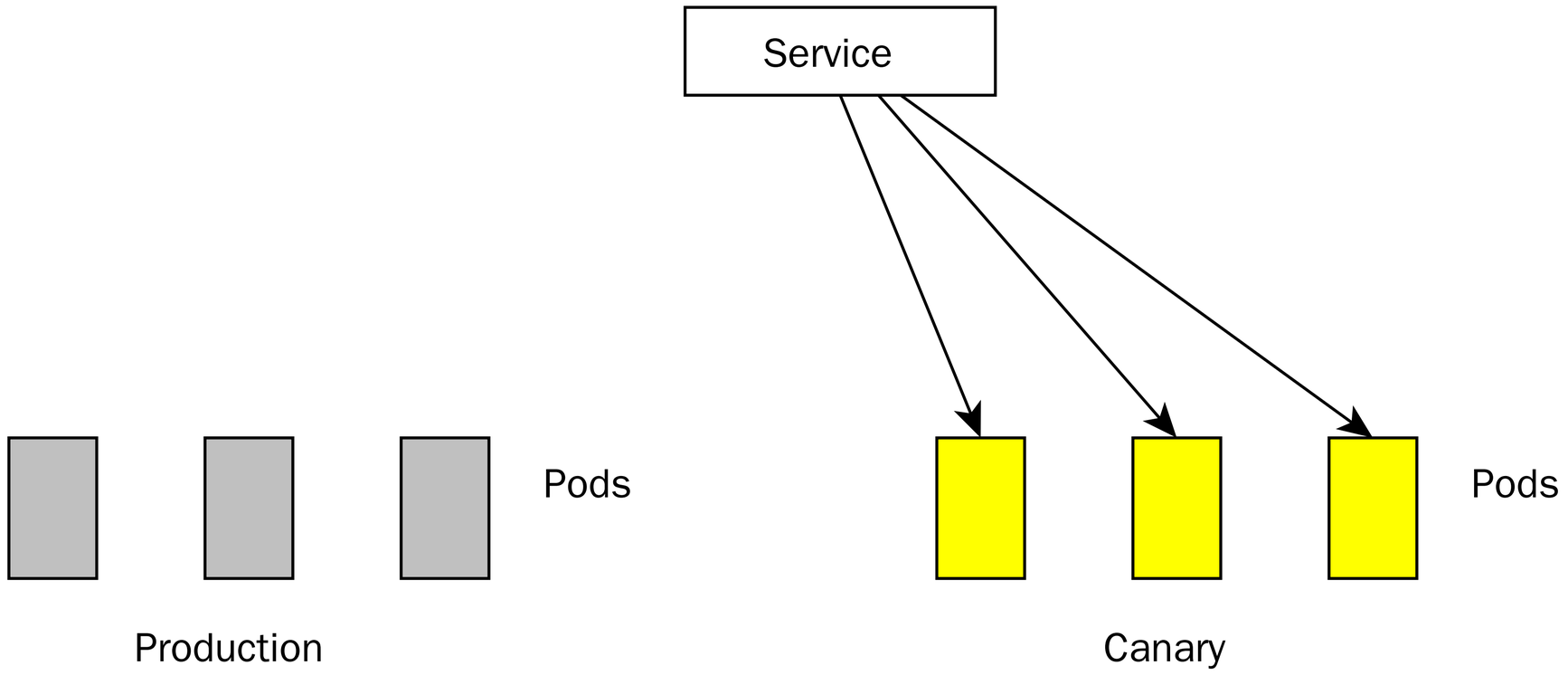
DEMO - CANARY



DEMO - CANARY



DEMO - CANARY



DEMO - CANARY

- manually

DEMO - CANARY

```
kubectl scale --replicas=3 deploy/api-status-nginx-blue
kubectl scale --replicas=1 deploy/api-status-nginx-green

# no errors, let's continoue
kubectl scale --replicas=2 deploy/api-status-nginx-blue
kubectl scale --replicas=2 deploy/api-status-nginx-green
```

CANARY WITH TRAEFIK

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  annotations:
    traefik.ingress.kubernetes.io/service-weights: |
      my-app: 99%
      my-app-canary: 1%
  name: my-app
spec:
  rules:
  - http:
      paths:
      - backend:
          serviceName: my-app
```

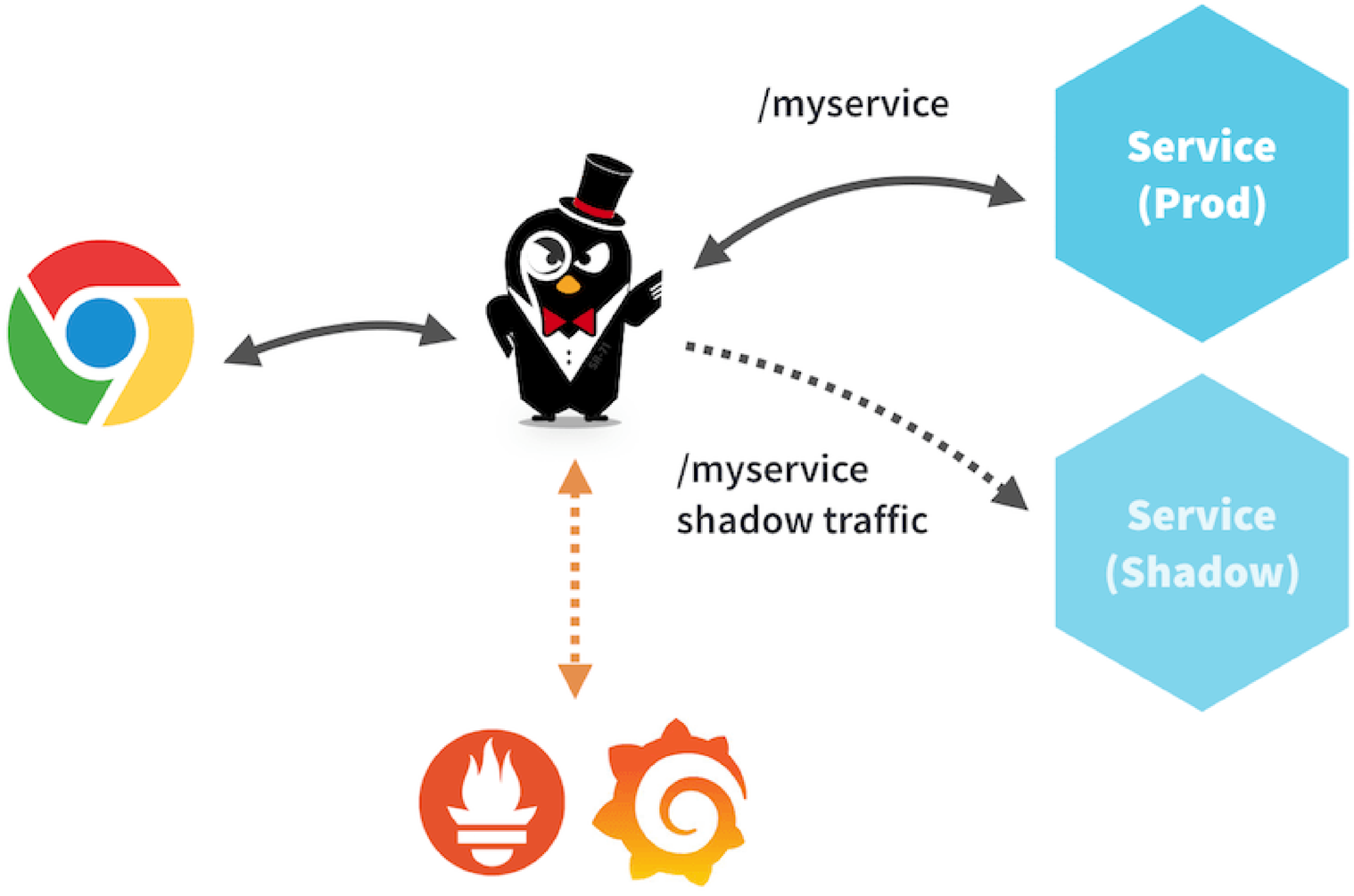
traffic-splitting

CANARY WITH ISTIO

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: helloworld
spec:
  hosts:
    - helloworld
  http:
  - route:
    - destination:
        host: helloworld
        subset: v1
      weight: 90
    - destination:
```

traffic shifting with Istio

SHADOW DEPLOYMENT

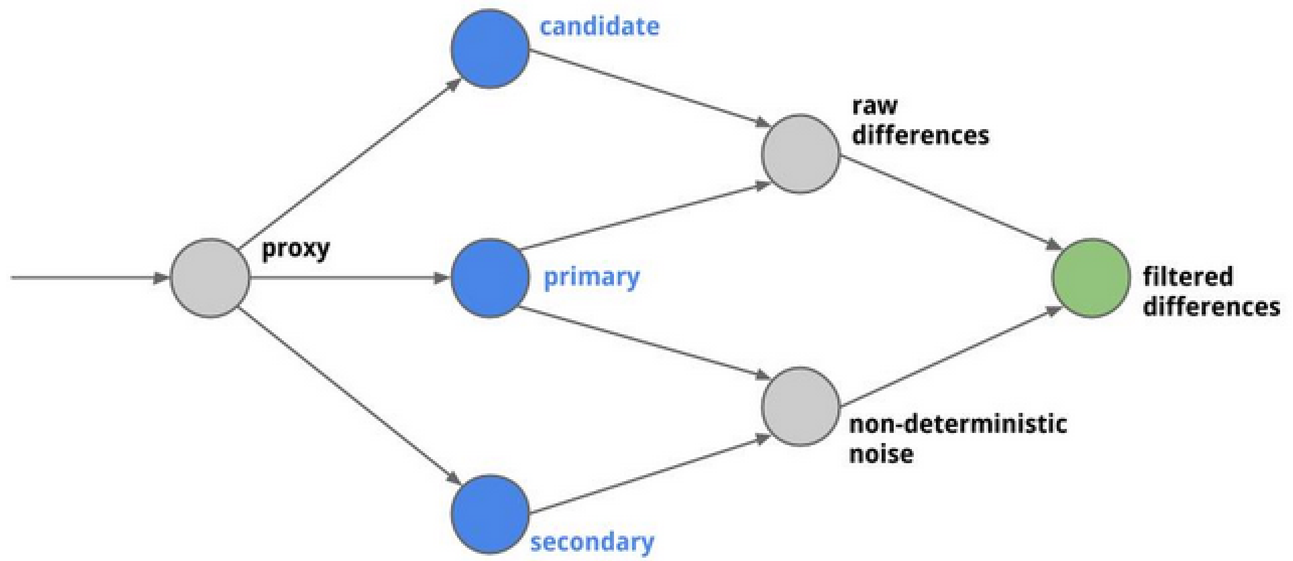


blog post on ambassador.io

SHADOW DEPLOYMENT

- With message or event bus, you can shadow individual components
- Powerfull techniques for running early development in prod!

SHADOW DEPLOYMENT - ADVANCE



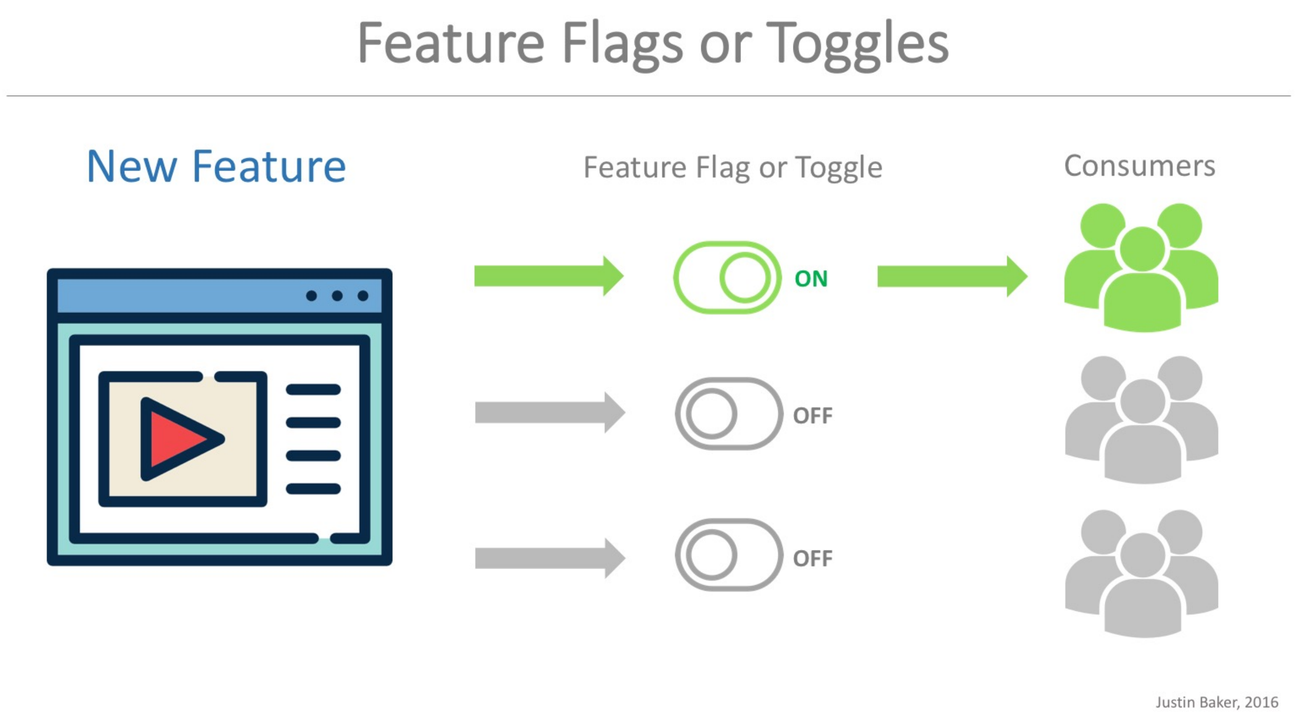
twitter engineering blog

SYNTHETIC CANARYING

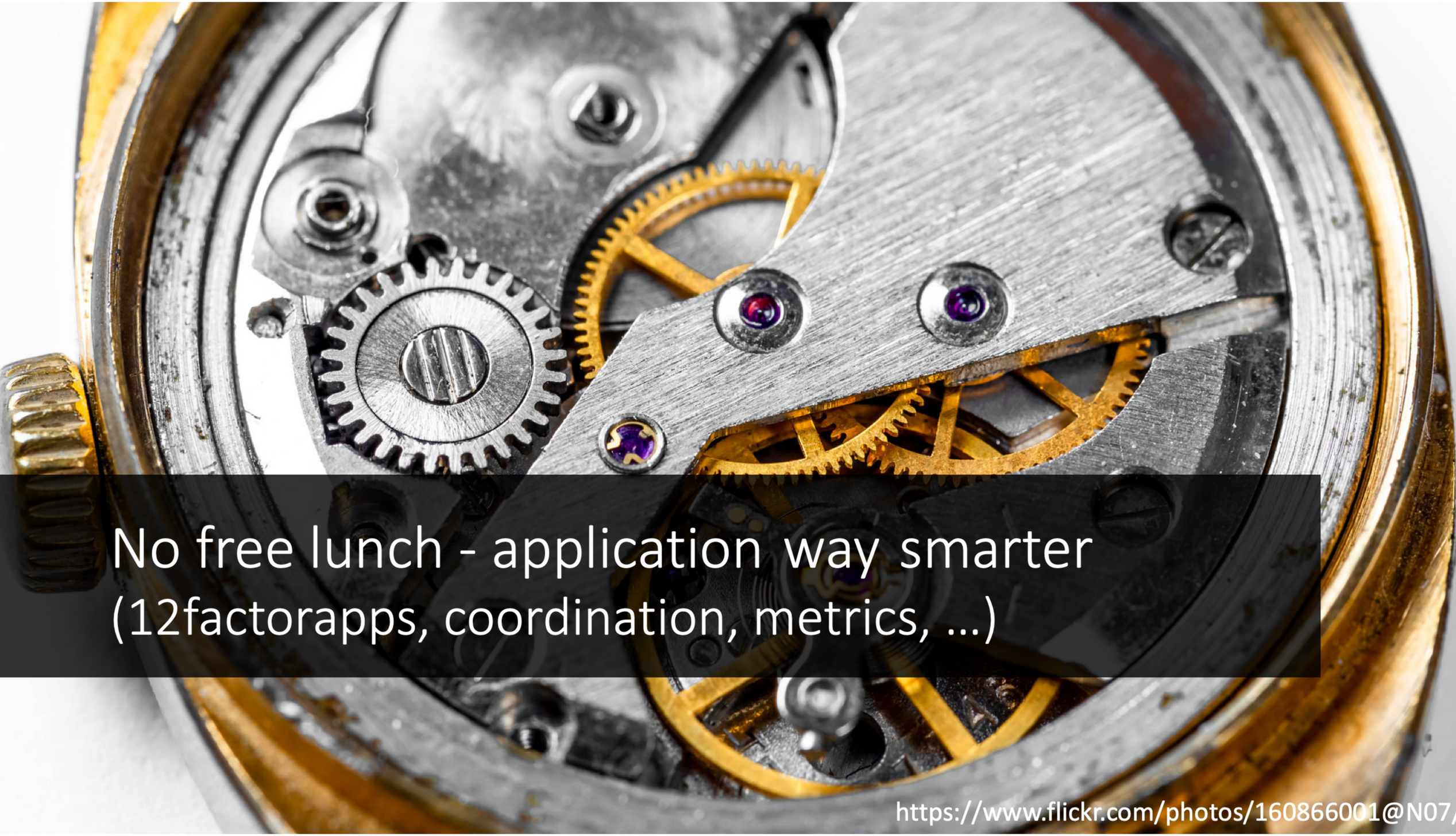
- Prepared requests similar to the user requests
- It may be an extended end-to-end automated tests

Synthetic Monitoring

FEATURE TOGGLES



see: martinfowler blog, featureflags.io



No free lunch - application way smarter
(12factorapps, coordination, metrics, …)

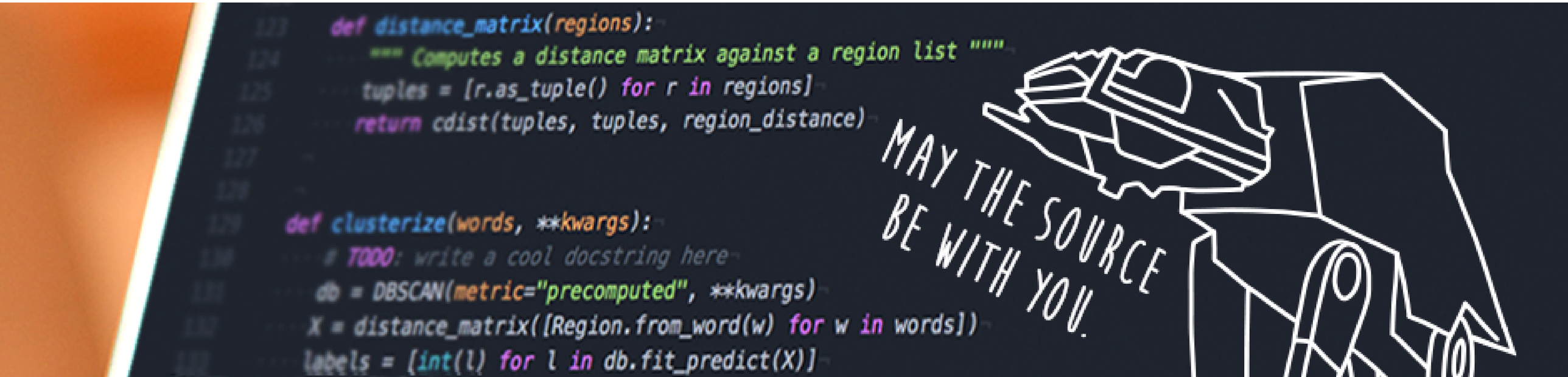
https://www.flickr.com/photos/160866001@N07/

SUMMARY

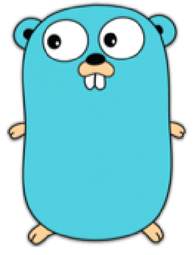
- K8S makes it easier...
- your application and storage must be smarter
- Keep the changes small

THANK YOU. QUESTIONS?

https://github.com/wojciech12/talk_zero_downtime_deployment_with_kubernetes



Hypatos Go



PYTORCH

TensorFlow™

amazon web services™

Azure

BACKUP SLIDES