DEPLOYMENT STRATEGIES WITH K8S FOR YOUR MICRO-SERVICE

+ HOW TO PREPARE YOUR SERVICE

Wojciech Barczynski - SMACC.io | Hypatos.ai

Listopad 2018

WOJCIECH BARCZYŃSKI

- Lead Software Engineer & System Engineer
- Interests: working software
- Hobby: teaching software engineering

SMACC

STORY

Go + Kubernetes

- **SMACC** - Fintech / ML - [10.2017- ...]
- **Lyke** - Mobile Fashion app - [12.2016, 07.2017]

AGENDA

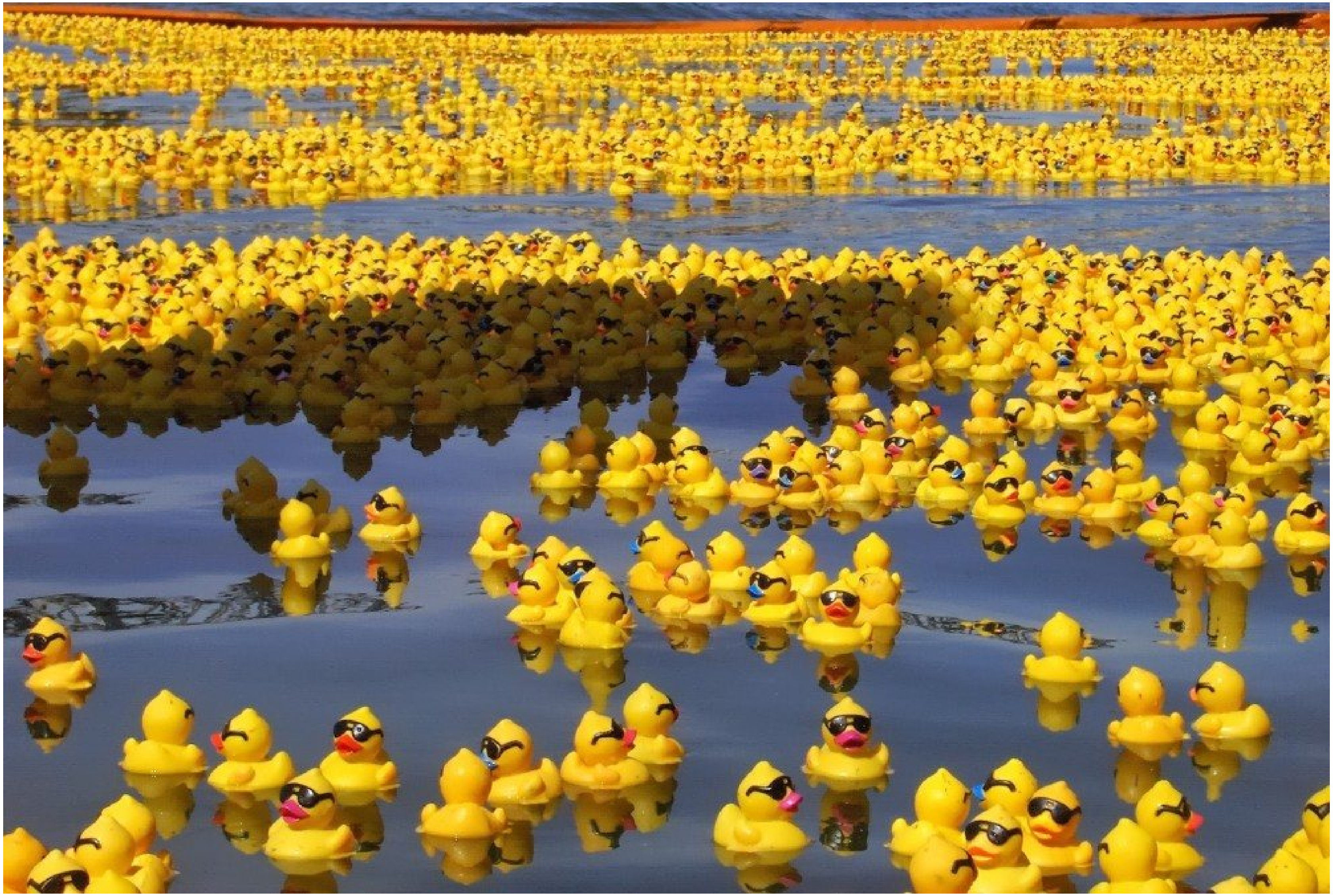
- Key Kubernetes Concepts
- How to prepare your service
- Deployment strategies

KUBERNETES

WHY?

- Operations and Admin is hard
- A lot of moving parts

MIKROSERWISY AAA!



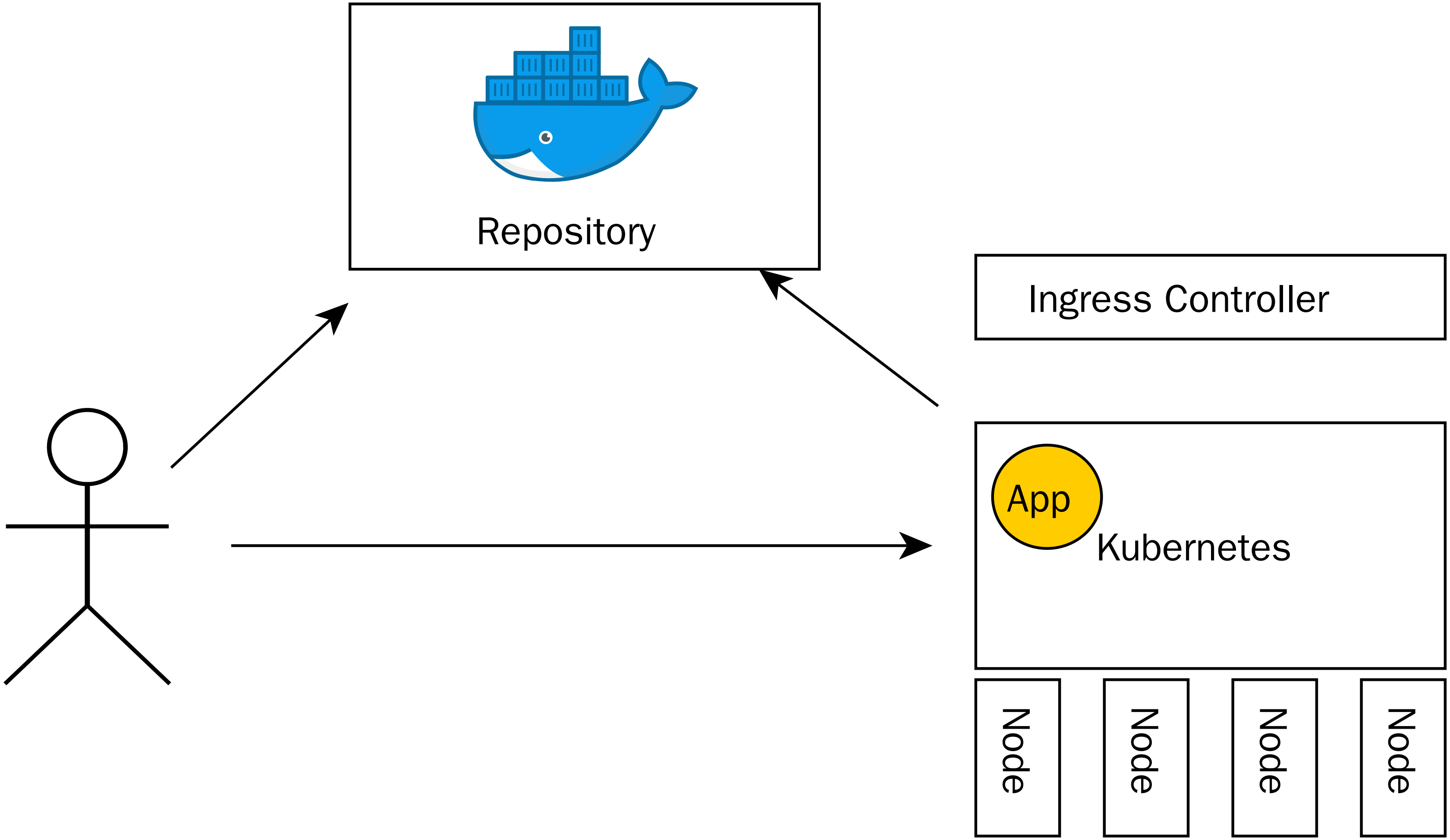
KUBERNETES

- Container management
- Service and application mindset
- Simple Semantic*
- Independent from IaaS provider

KUBERNETES

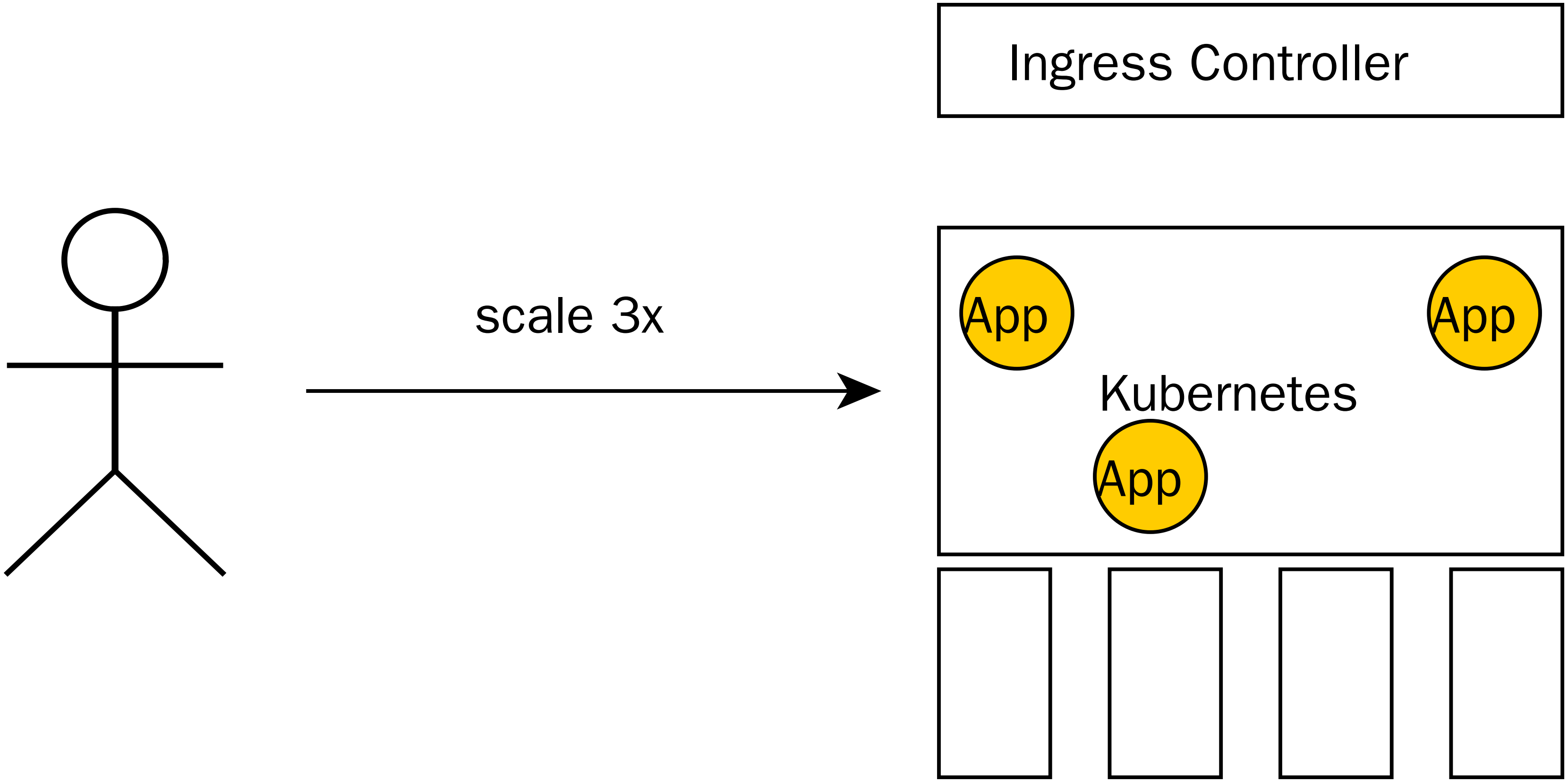
- Data Center as black box
- Batteries for your 12factory apps
- Service discovery, meta-data support

KUBERNETES



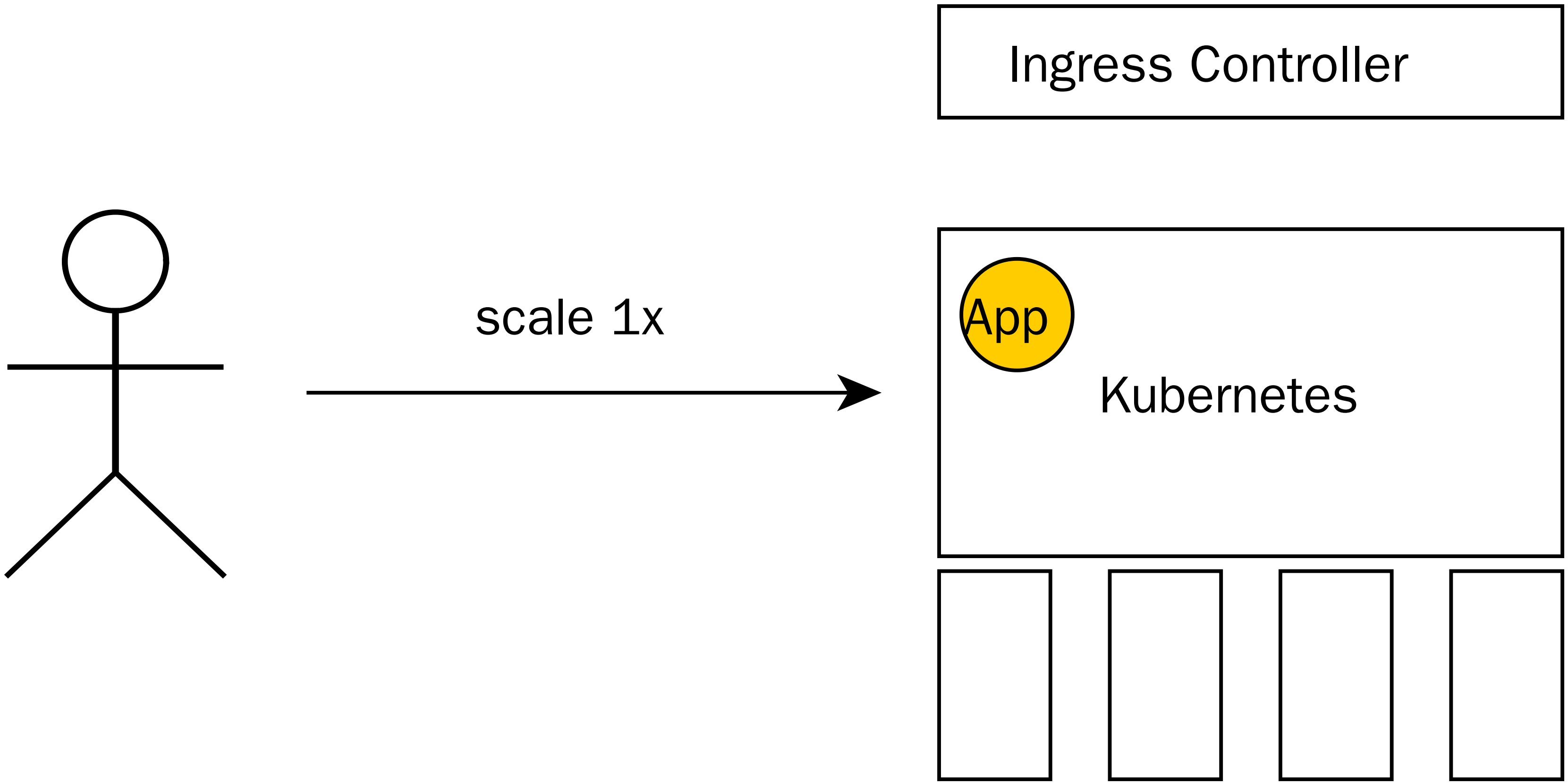
```
make docker_push; kubectl create -f app-srv-dpl.yaml
```

SCALE UP! SCALE DOWN!



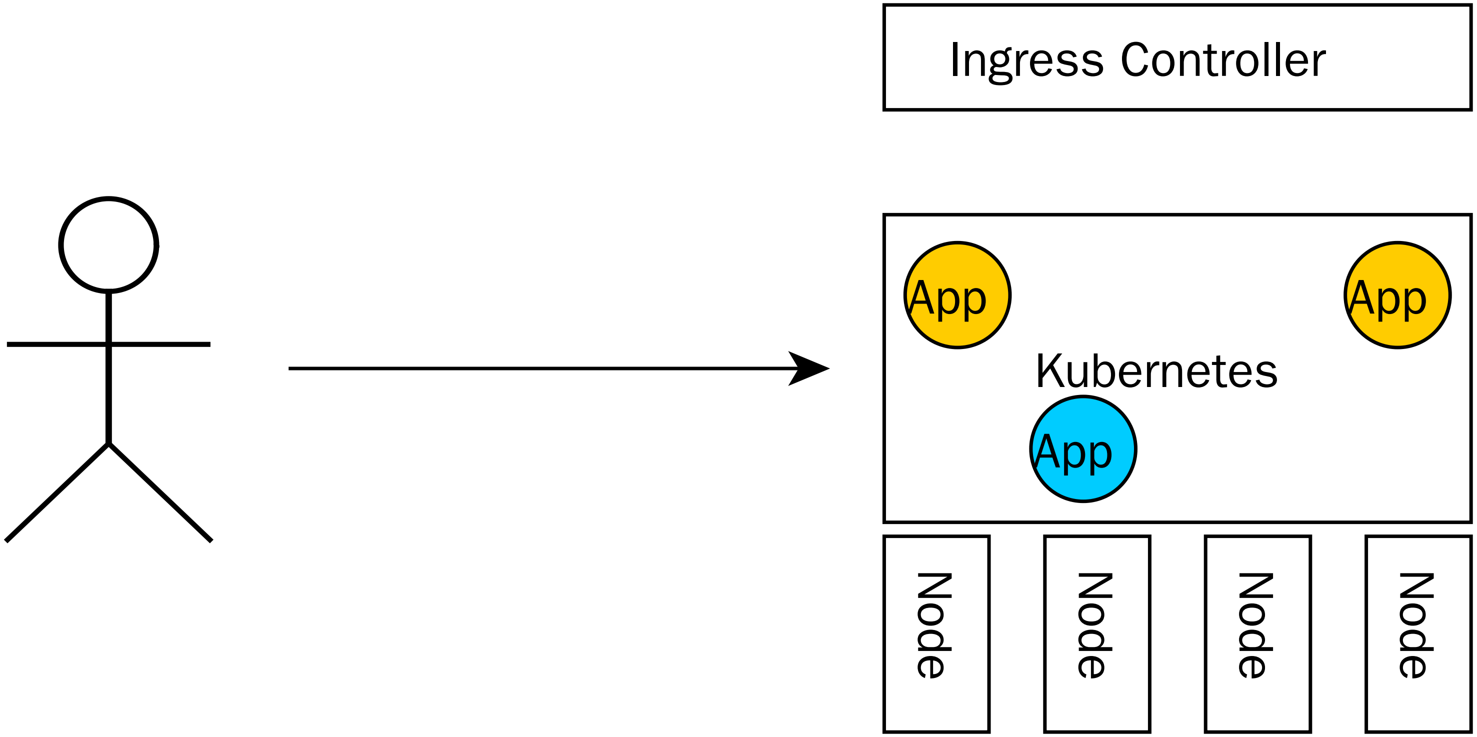
```
kubectl --replicas=3 -f app-srv-dpl.yaml
```

SCALE UP! SCALE DOWN!



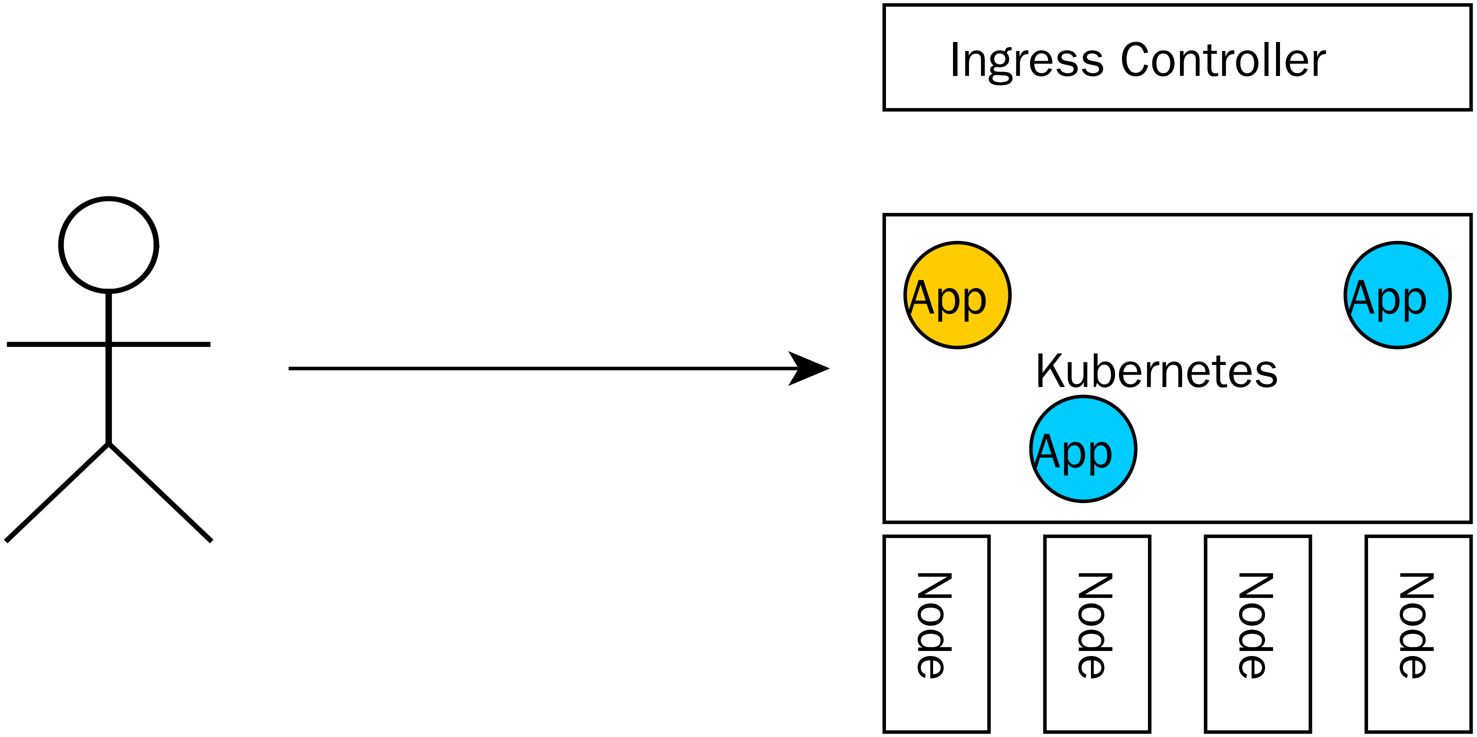
```
kubectl --replicas=1 -f app-srv-dpl.yaml
```

ROLLING UPDATES!



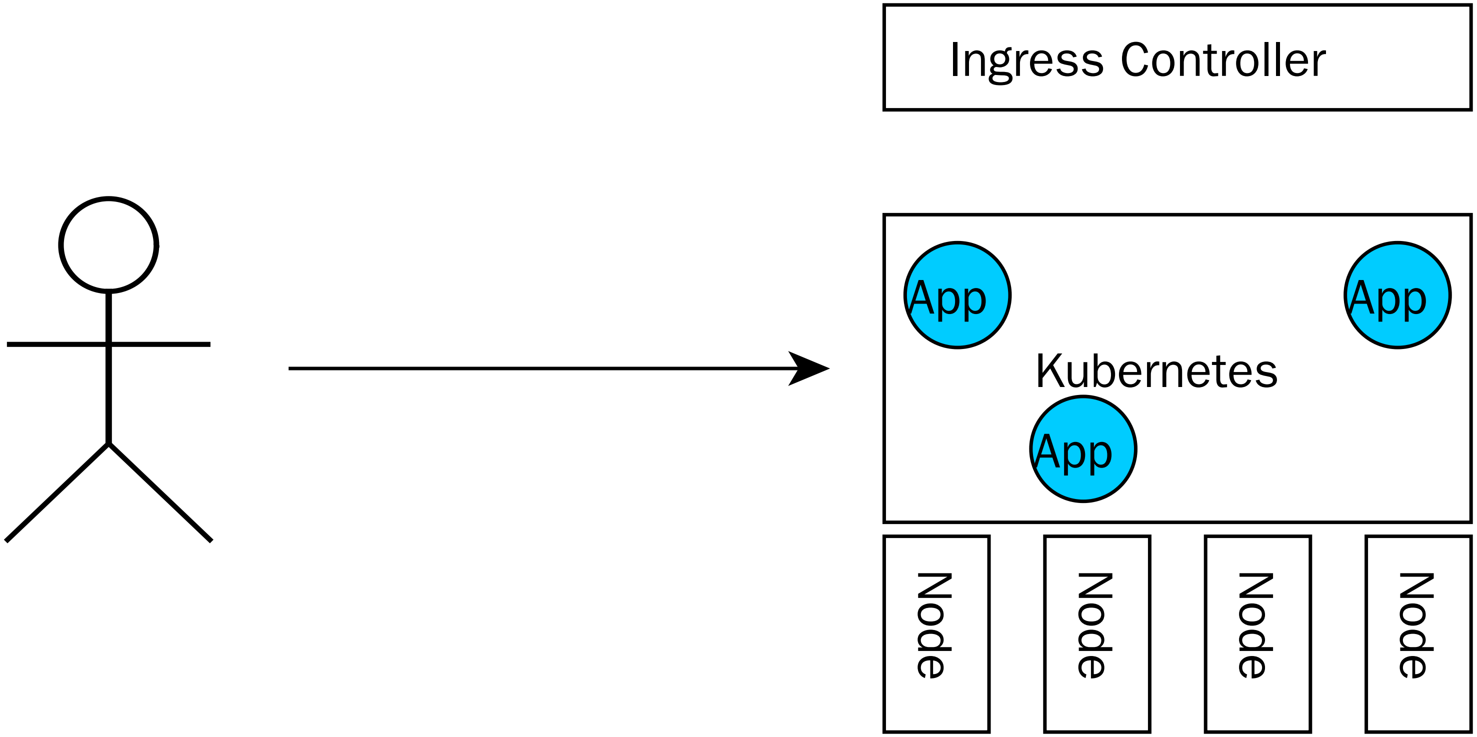
```
kubectl set image deployment/app app=app:v2.0.0
```

ROLLING UPDATES!



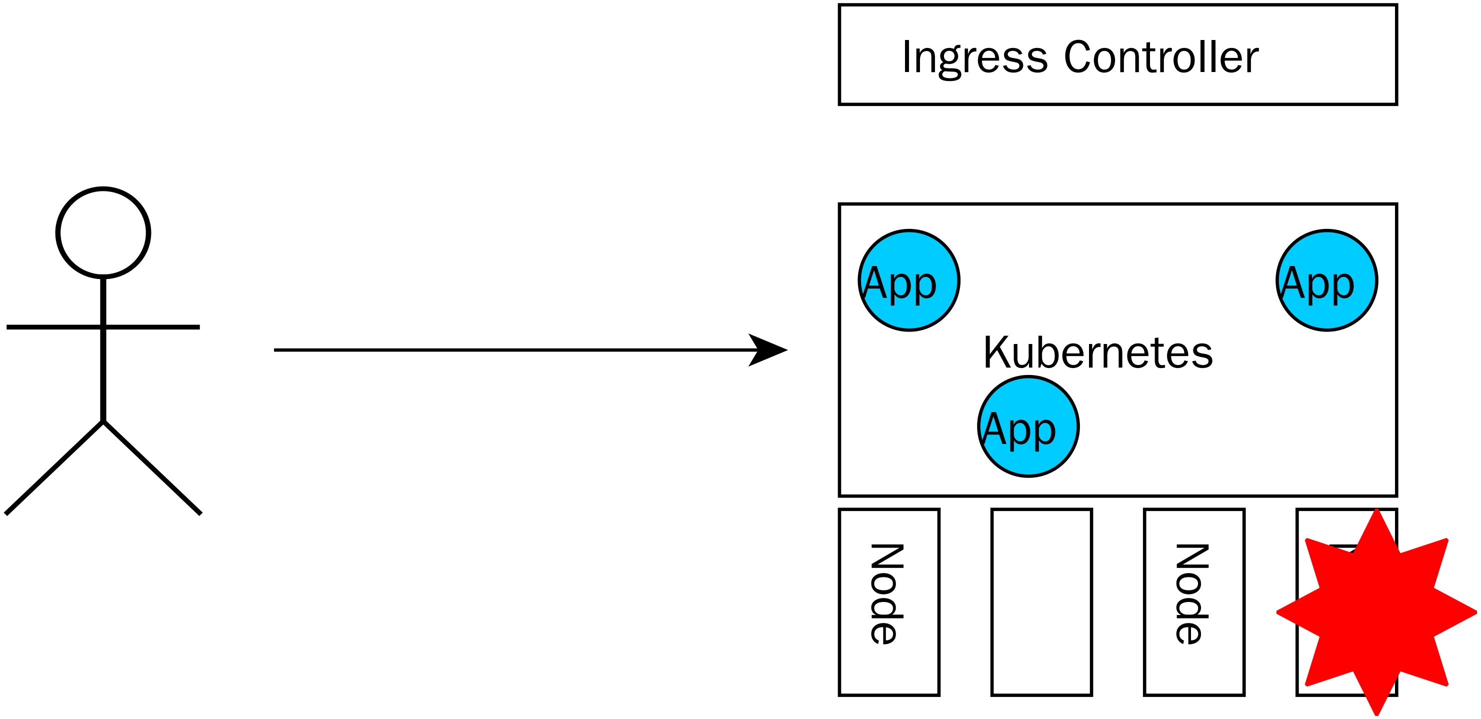
```
kubectl set image deployment/app app=app:v2.0.0
```

ROLLING UPDATES!

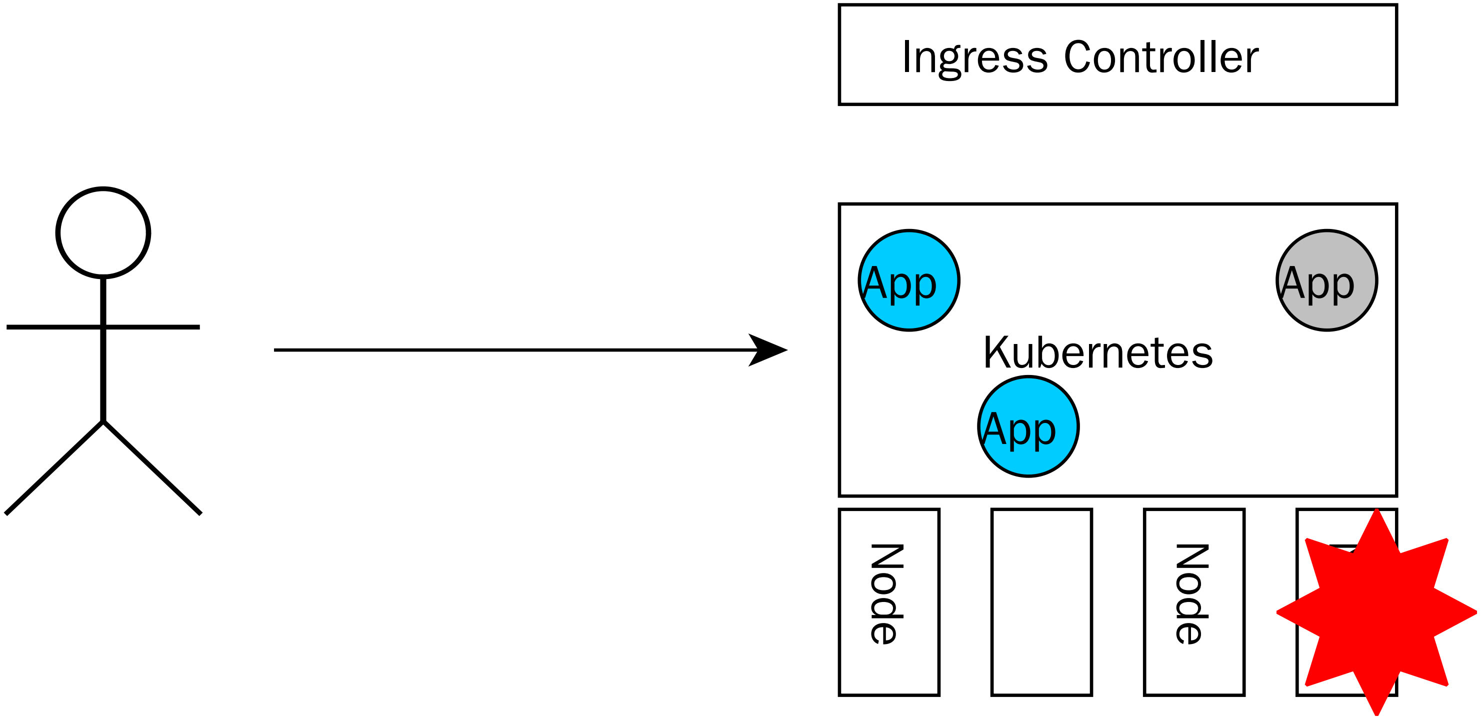


```
kubectl set image deployment/app app=app:v2.0.0
```

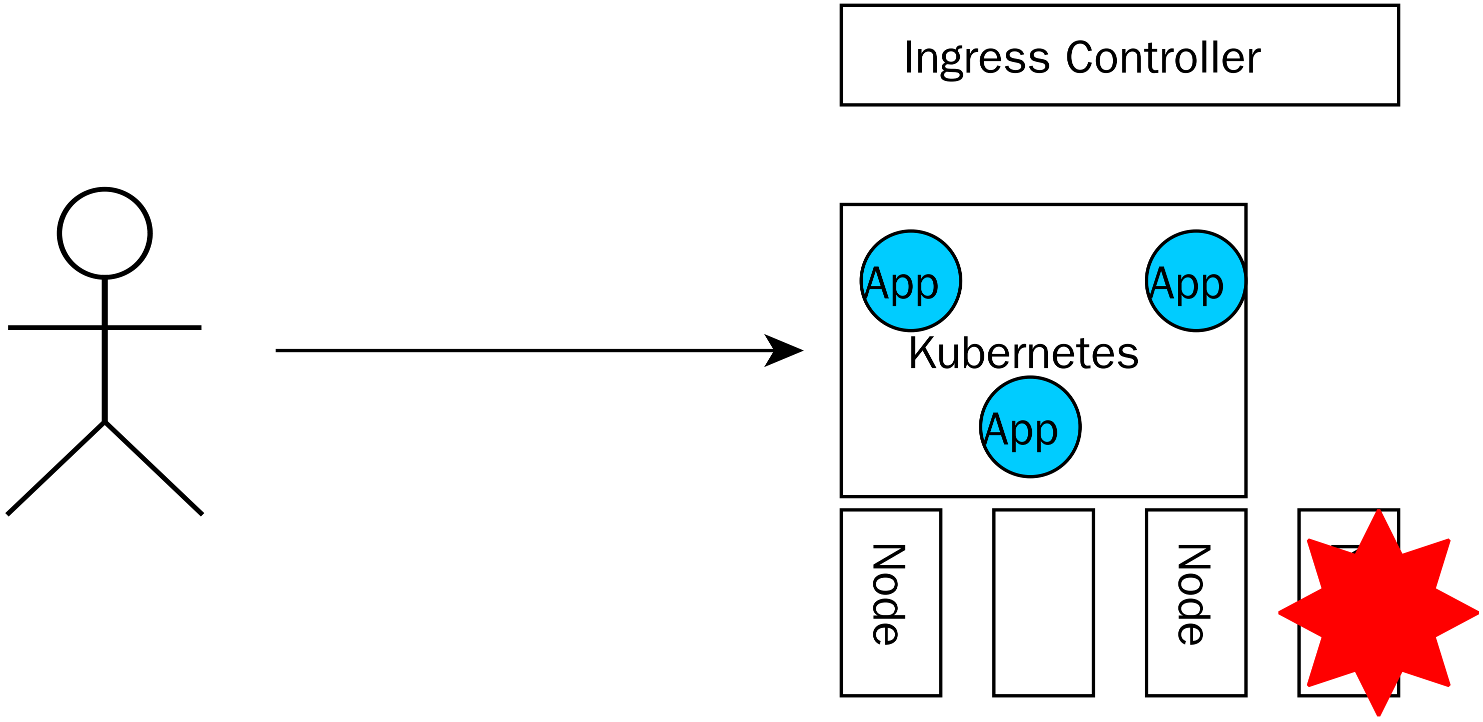
RESISTANCE / MIGRATION



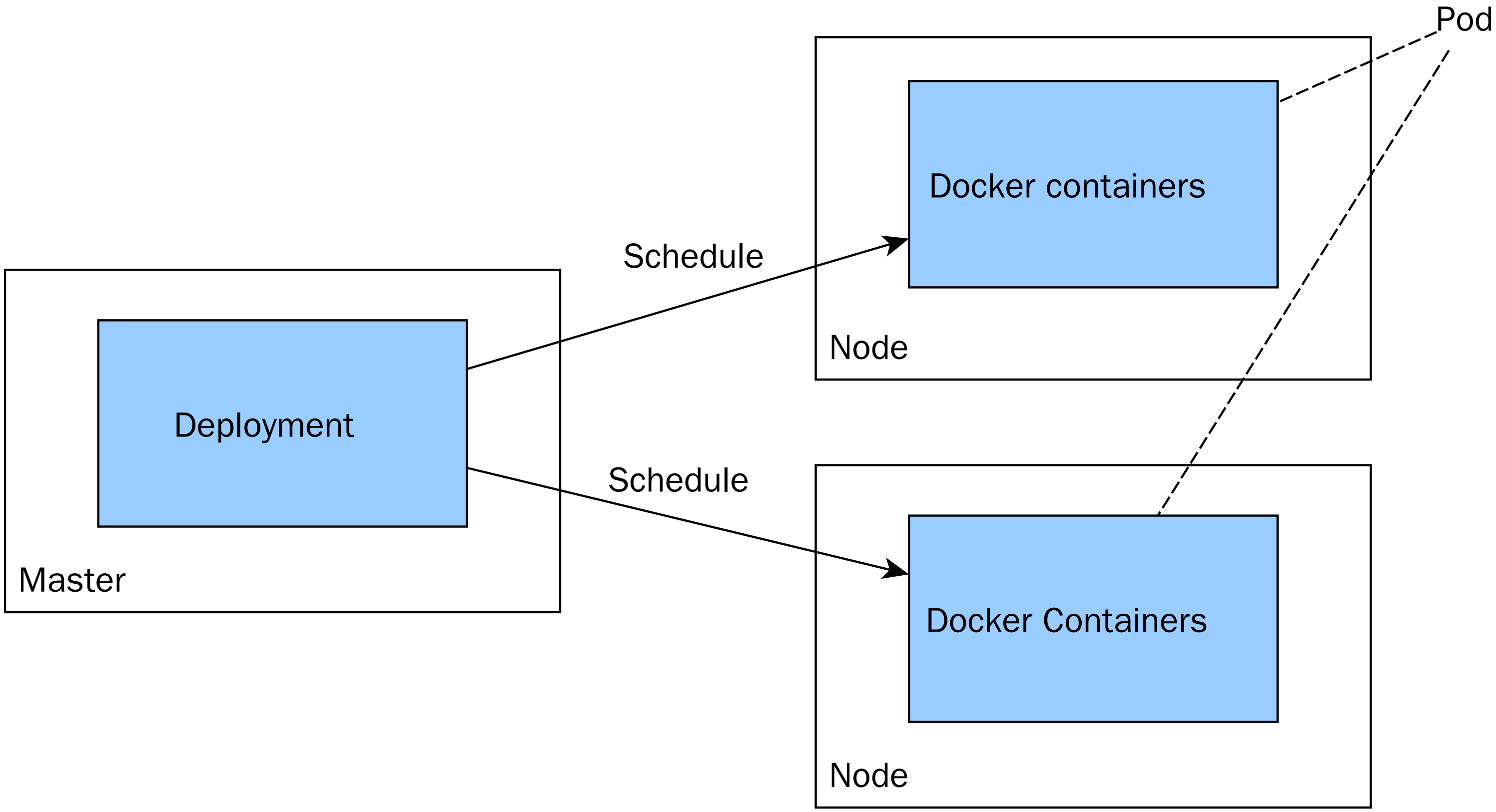
RESISTANCE / MIGRATION



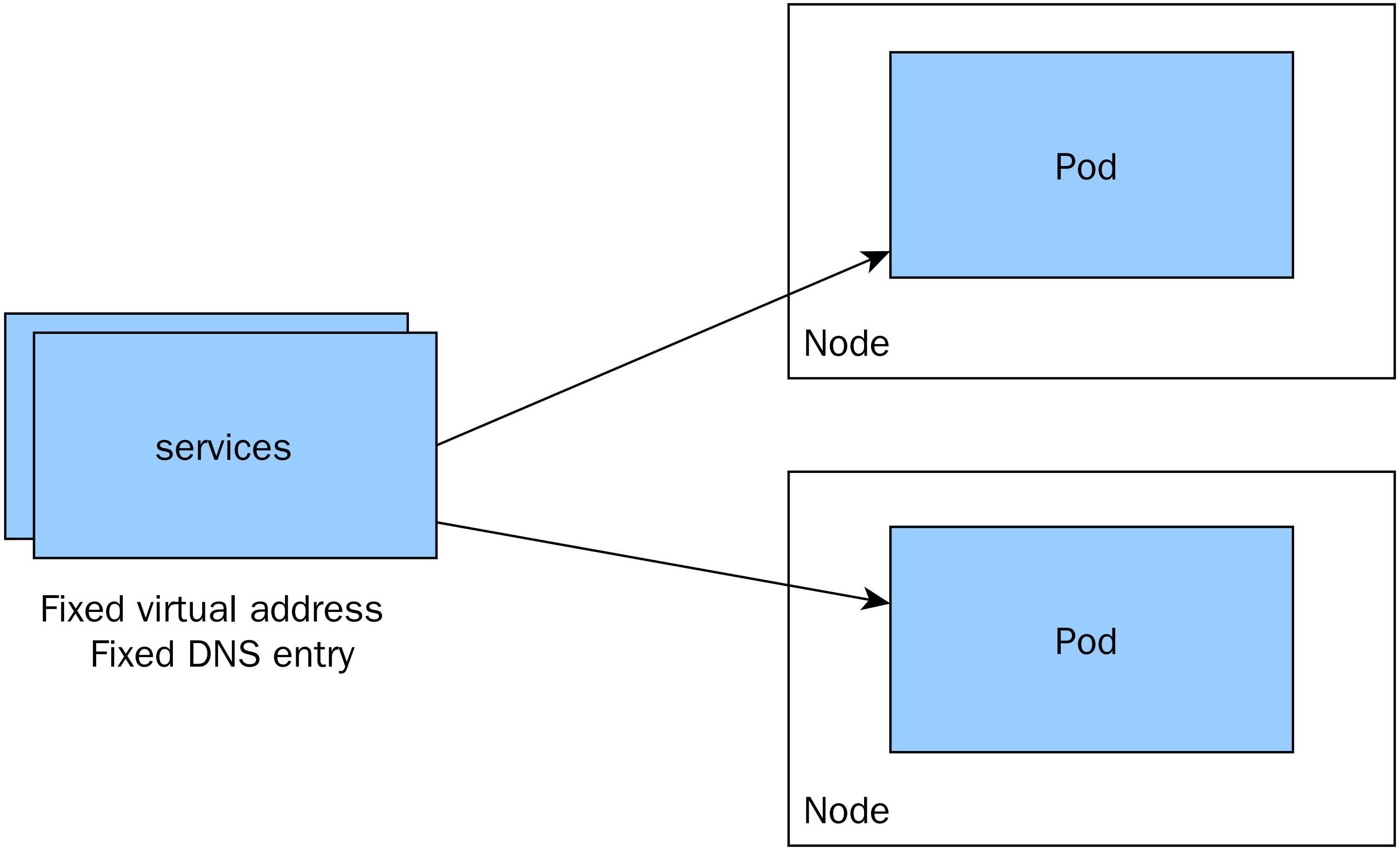
RESISTANCE / MIGRATION



DEPLOYMENT AND PODS



SERVICE AND PODS



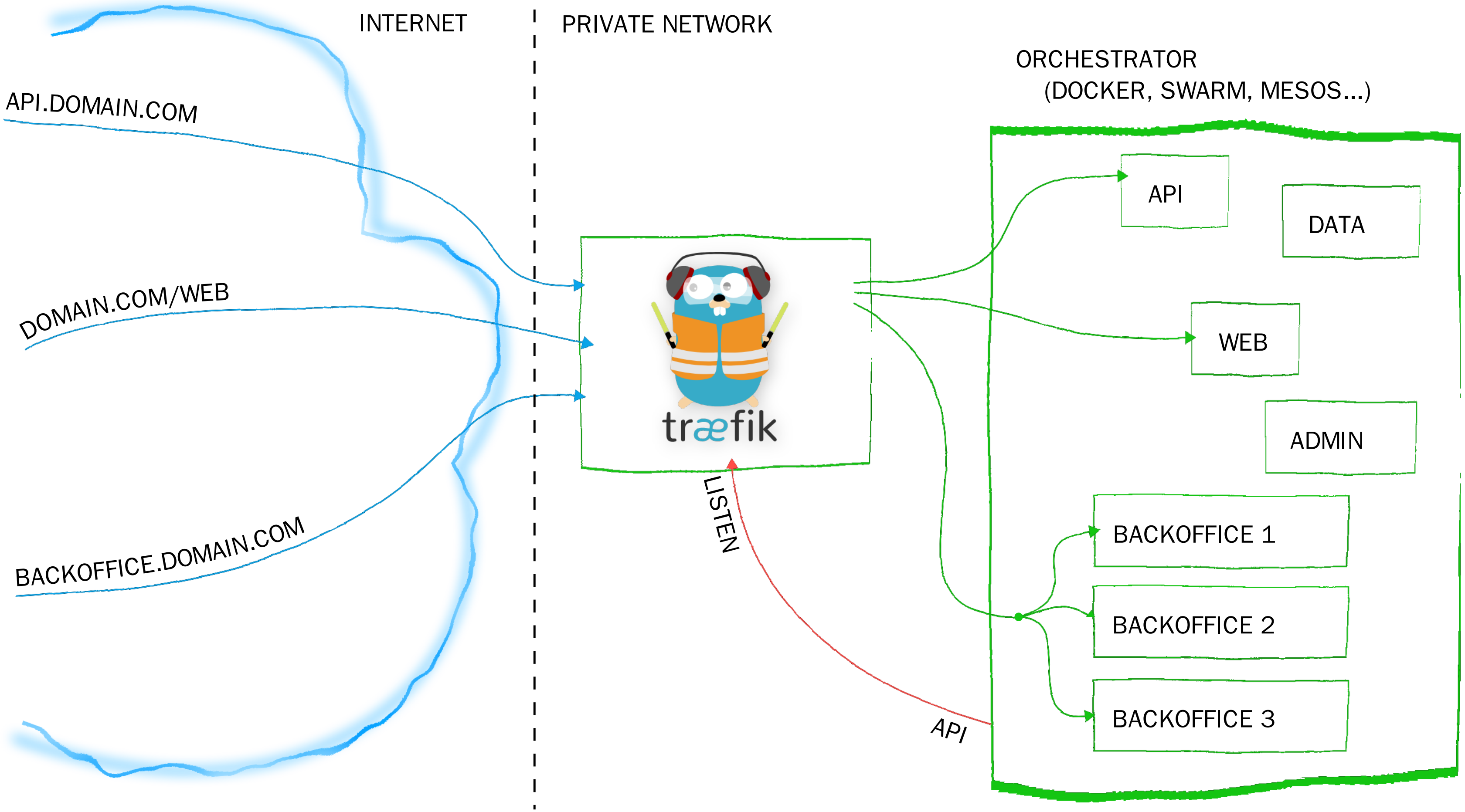
Fixed virtual address
Fixed DNS entry

Service matches pods based on labels

BASIC CONCEPTS

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

HOW GET USER REQUESTS?



Ingress Controller

INGRESS

Pattern	Target App Service
api.smacc.io/v1/users	users-v1
api.smacc.io/v2/users	users-v2
smacc.io	web

SERVICE DISCOVERY

- names in DNS:

  ```
  curl http://users/list
  ```

- labels:

  ```
  name=value
  ```

- annotations:

  ```
  prometheus.io/scrape: "true"
  ```

DROP-IN

- traefik / Ingress / Envoy
- prometheus
- audit checks
- ...

DEPLOYMENT STRATEGIES

STRATEGIES

We will see:

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

OTHER

We will not cover:

- Feature toggles
- A/B like
- Shadow deployment

FIRST THE HOMEWORK

Need to support:

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

KUBE LIVENESS PROBE

```
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

KUBE 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 gives 500 on readinessProbe

YOUR APP SHOULD ON STOP

1. we get SIGTERM signal
2. app gives 500 on readinessProbe
3. k8s does not send more requests

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

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. kuberenetes forces kill if 30s limit exceeded

ALWAYS

Implement readiness for:

- ML Model-based components
- slow starting time

DEMO SERVICE IMPLEMENTATION

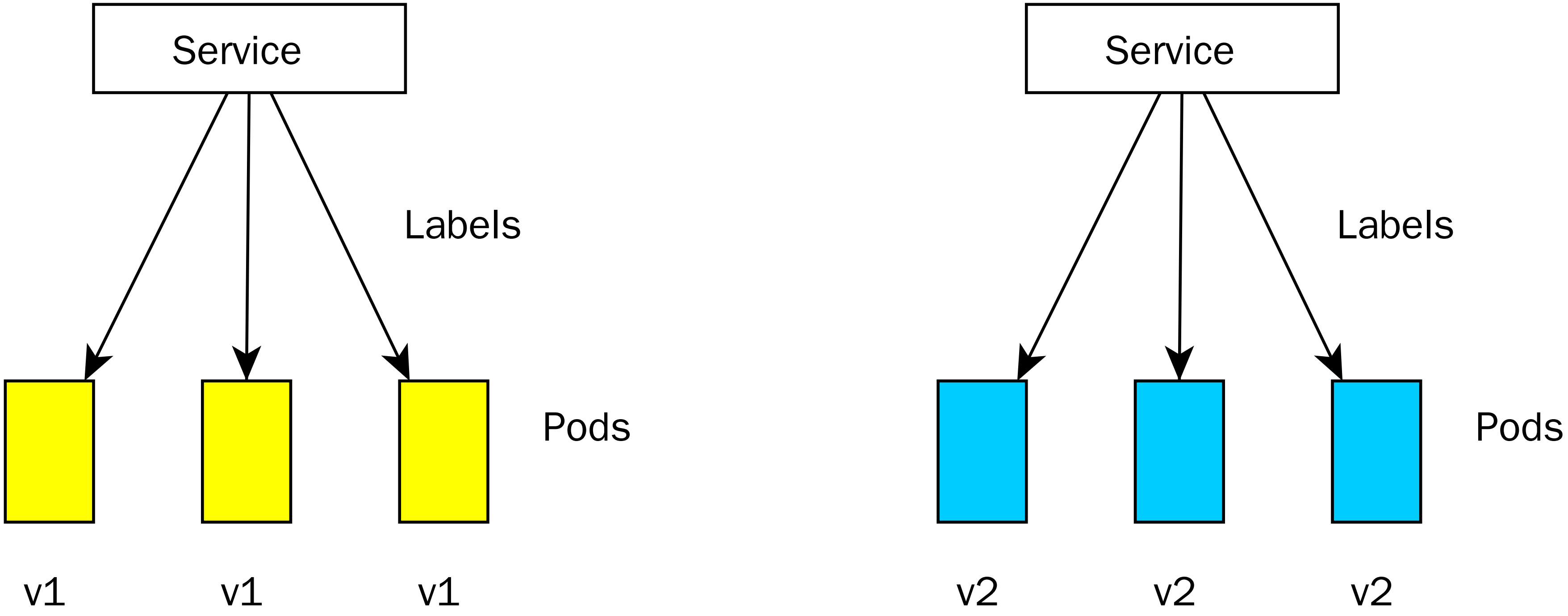
- graceful shutdown
- demo service

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

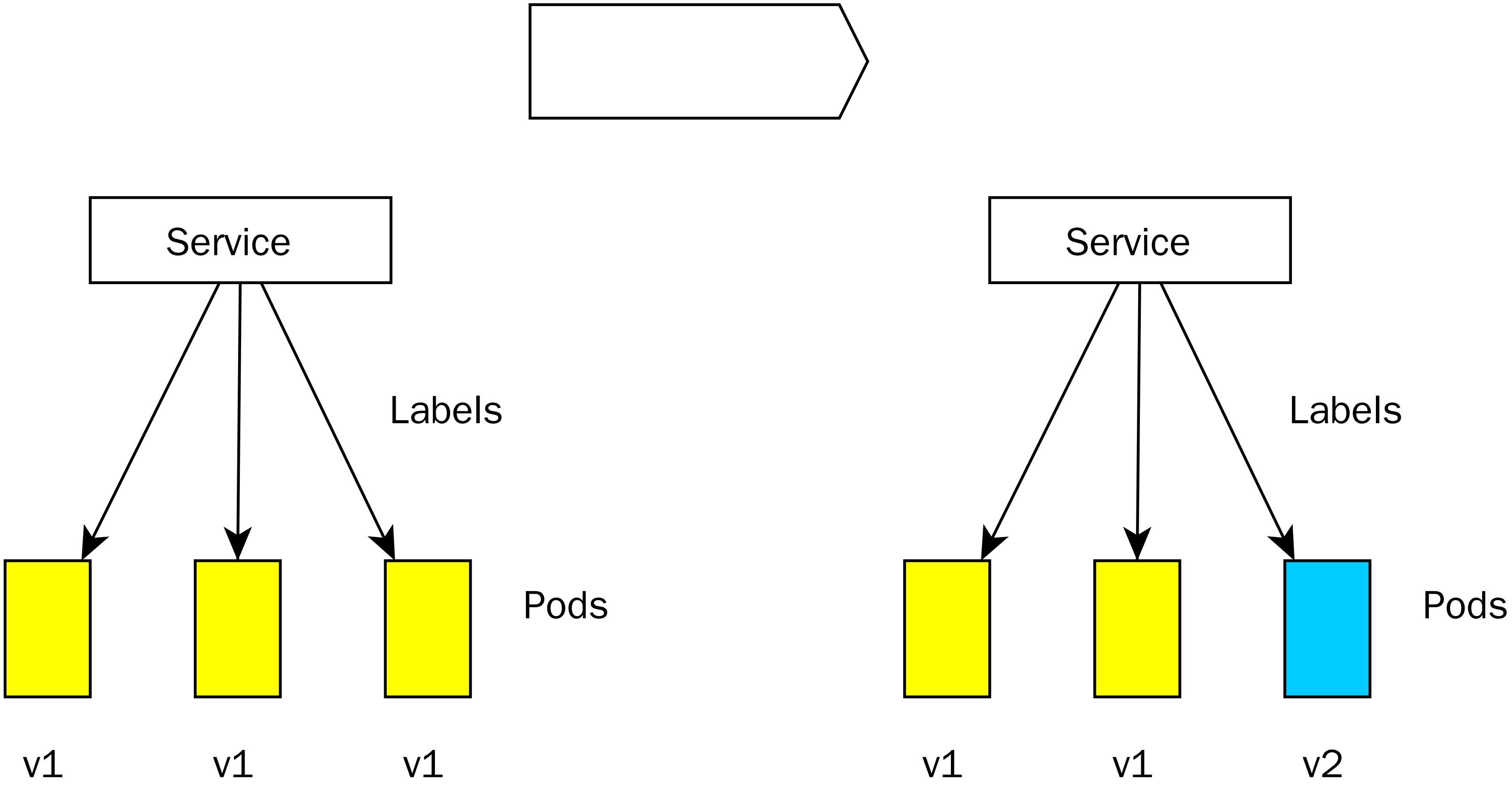
```yaml
spec:
  replicas: 3
  strategy:
    type: Recreate
```

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

DEMO - RECREATE

- quick
- downtime visible

DEMO - ROLLING UPDATES



DEMO - ROLLING UPDATES

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

docs

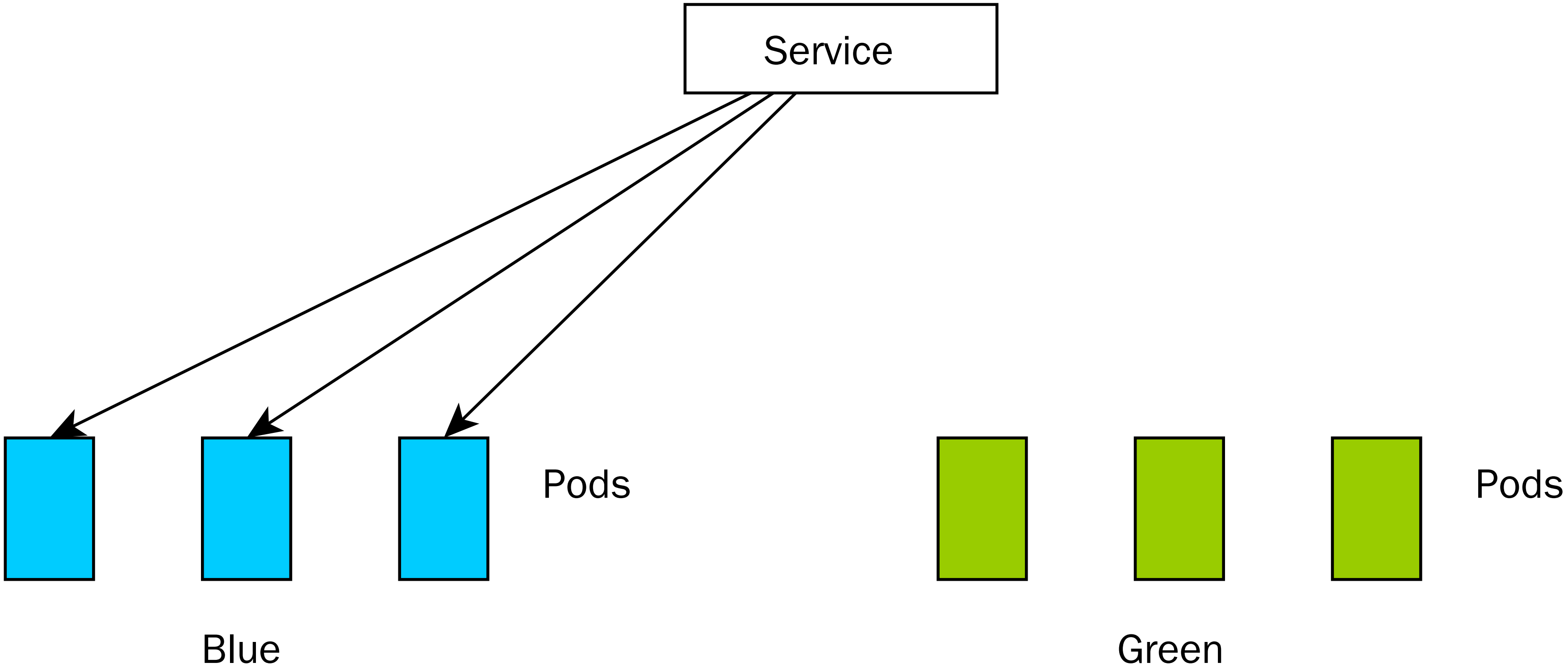
DEMO - ROLLING UPDATES

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

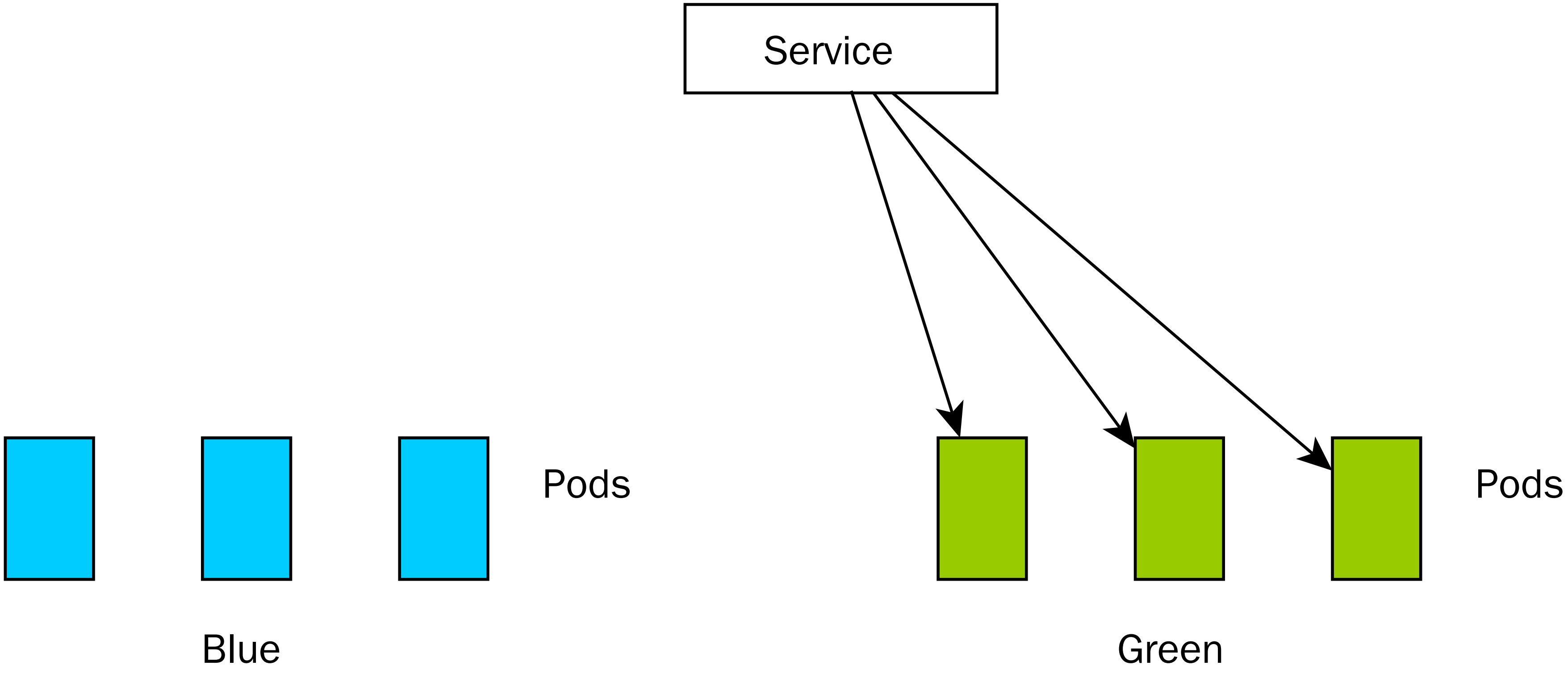
DEMO - ROLLING UPDATES

- the most popular

DEMO - GREEN/BLUE



DEMO - GREEN/BLUE



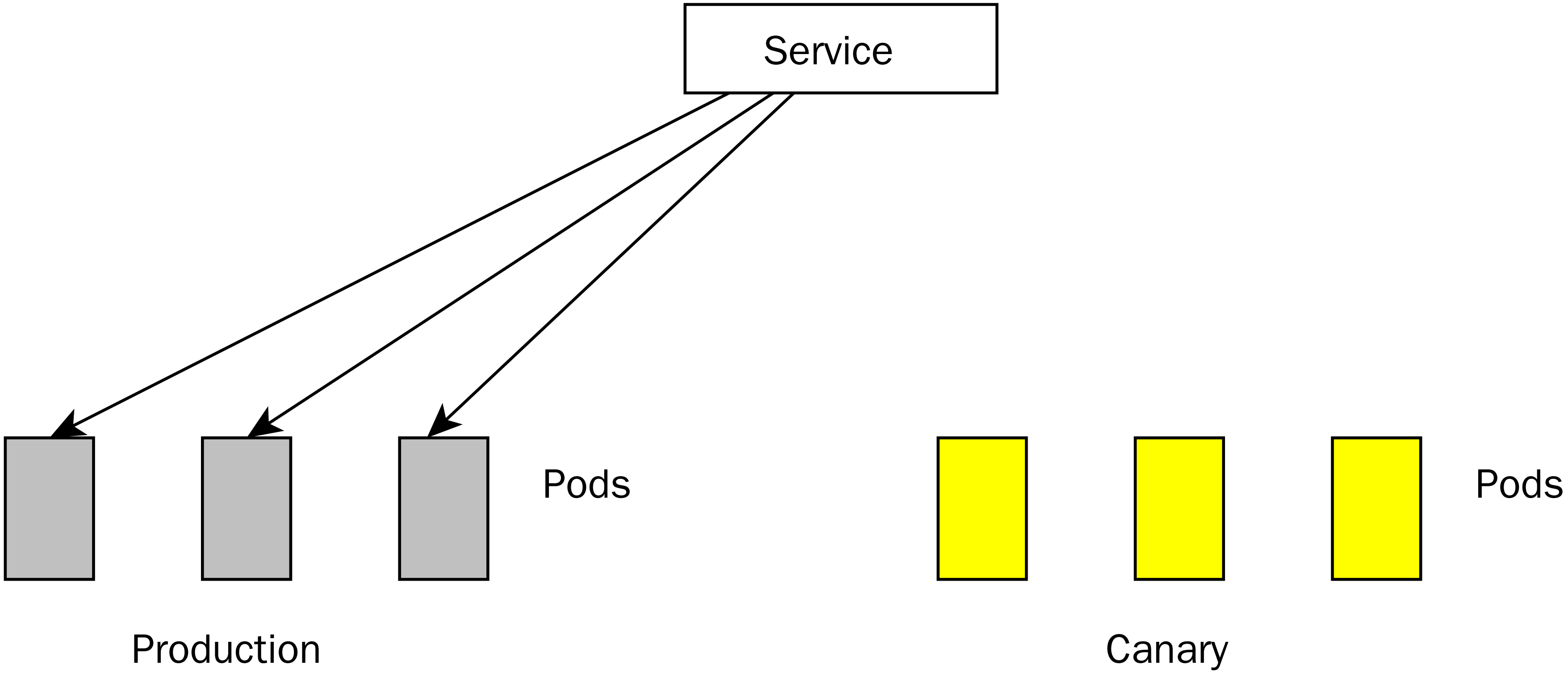
DEMO - GREEN/BLUE

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

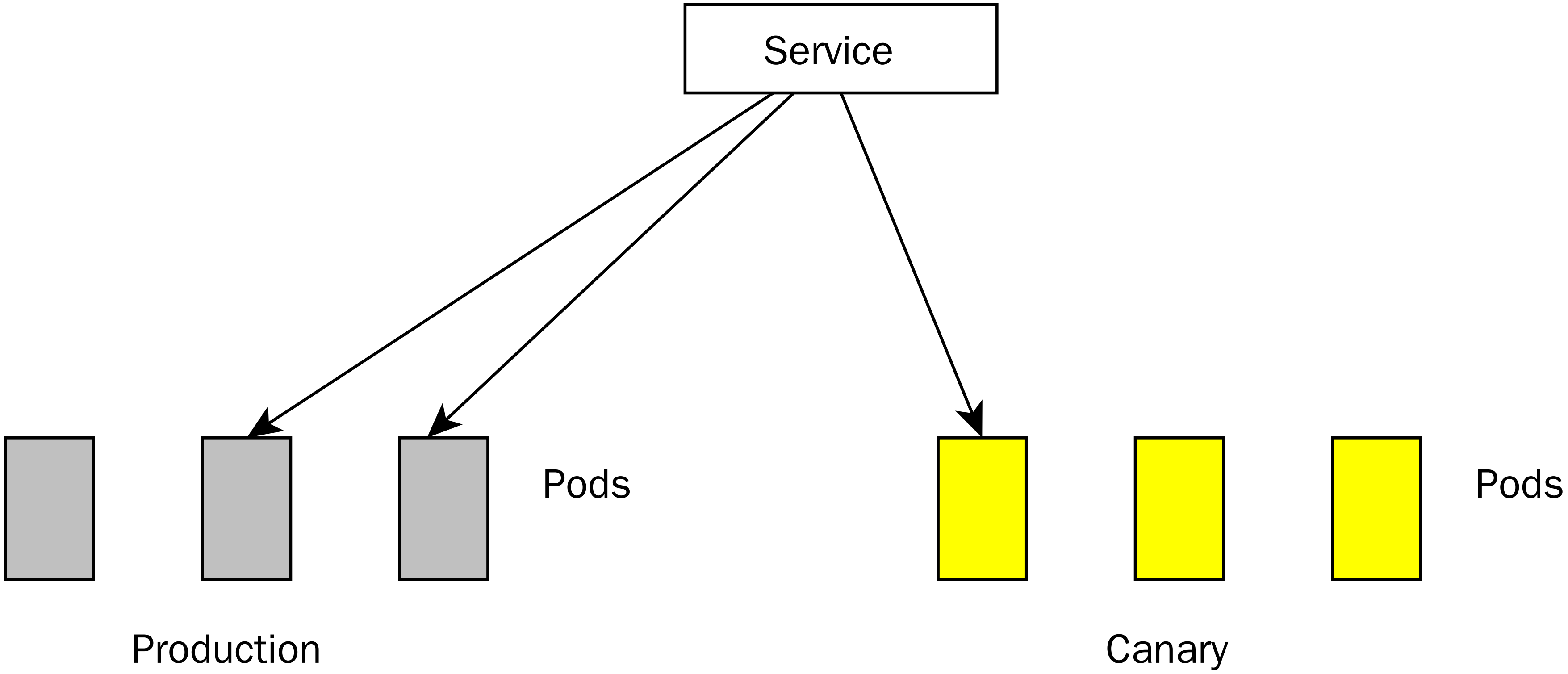
DEMO - GREEN/BLUE

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

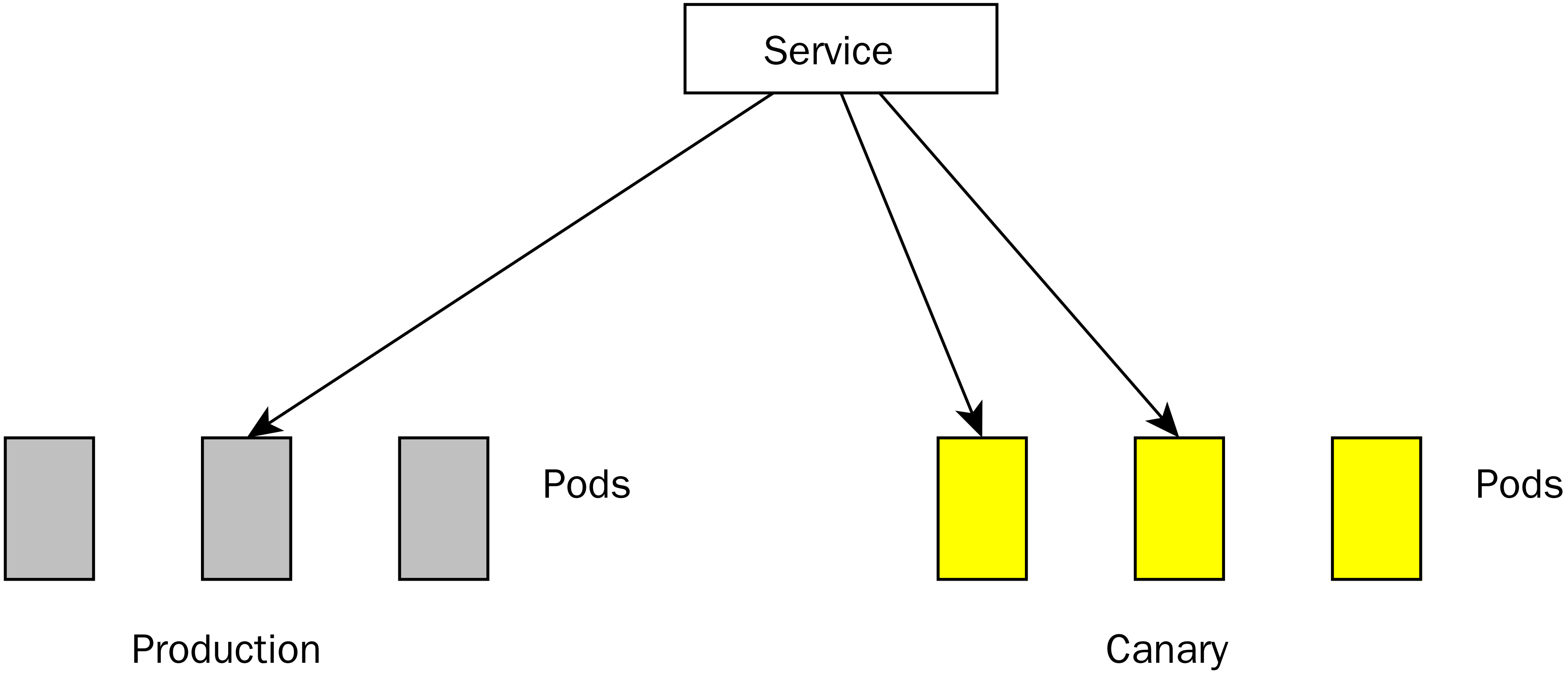
DEMO - CANARY



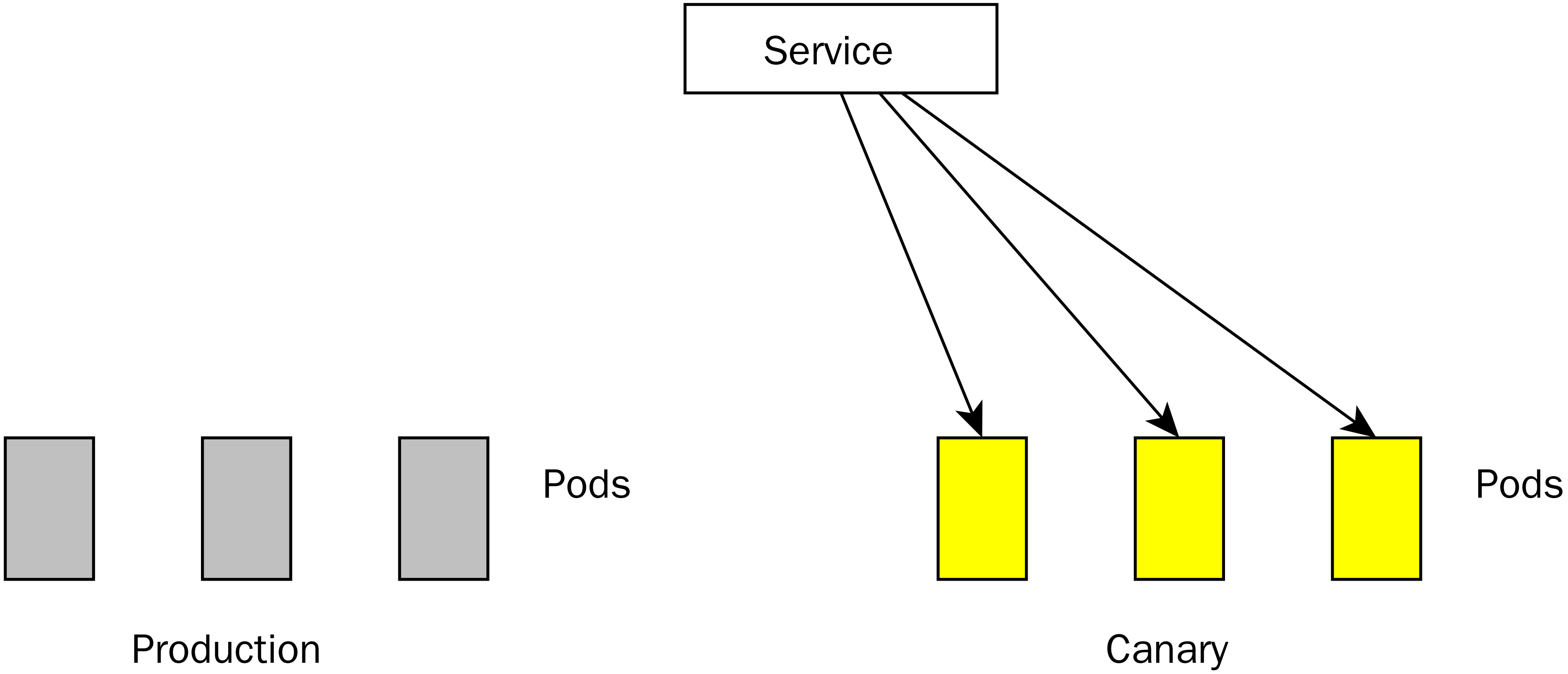
DEMO - CANARY



DEMO - CANARY



DEMO - CANARY



```
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
```

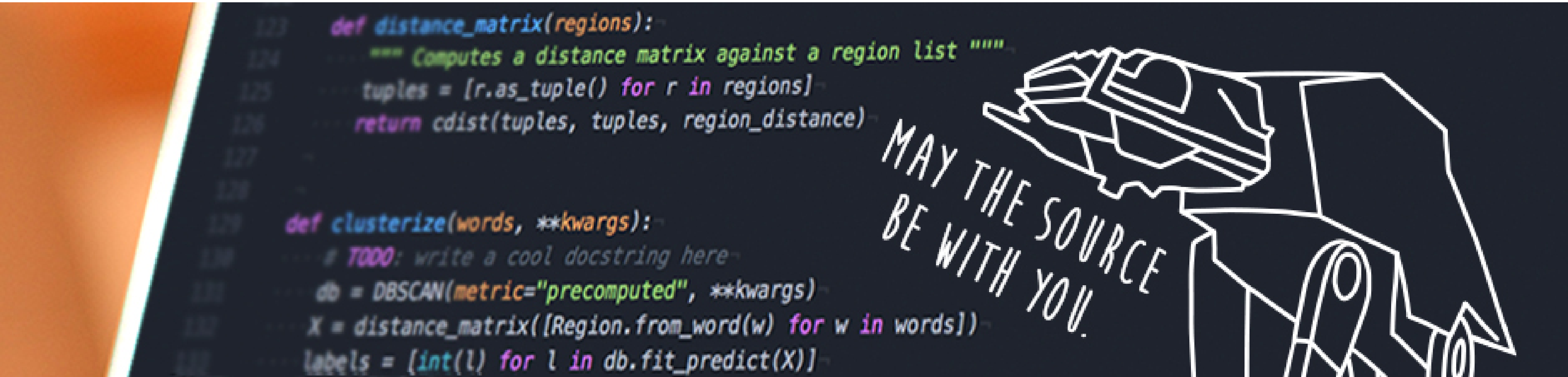
DEMO - CANARY

- manually
- with help of Traefik / Istio / ...

SUMMARY

- kubernetes simple semantic
- easy deployment of your applications
- will work for any application type

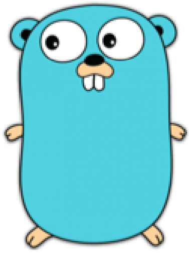
DZIĘKUJĘ. PYTANIA?



```
def distance_matrix(regions):
    """ Computes a distance matrix against a region list """
    tuples = [r.as_tuple() for r in regions]
    return cdist(tuples, tuples, region_distance)


def clusterize(words, **kwargs):
    # TODO: write a cool docstring here
    db = DBSCAN(metric="precomputed", **kwargs)
    X = distance_matrix([Region.from_word(w) for w in words])
    labels = [int(l) for l in db.fit_predict(X)]
```

SMACC



Go

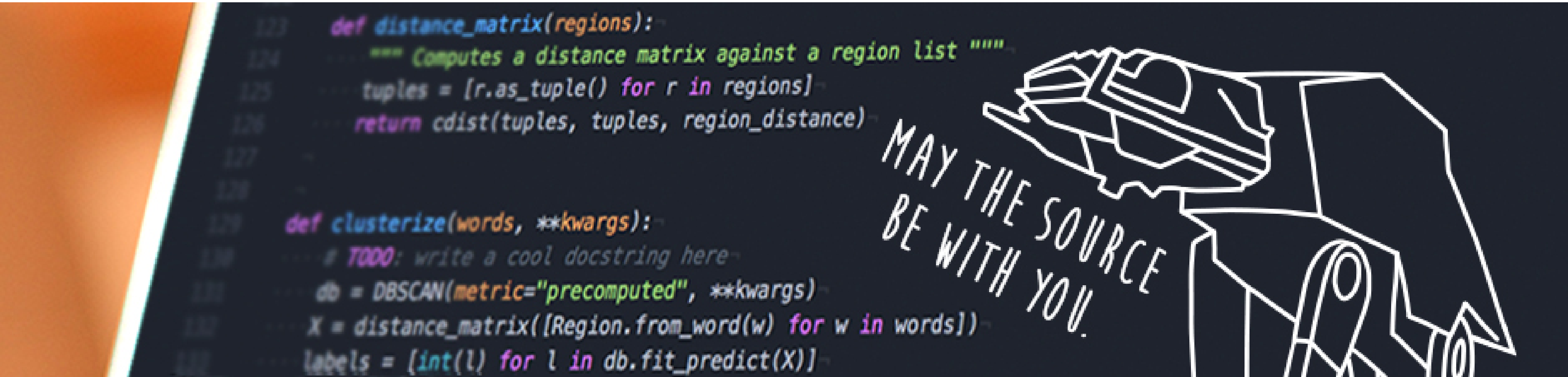
PYTORCH

TensorFlow™

amazon web services™

Azure

BACKUP SLIDES



```
def distance_matrix(regions):
    """ Computes a distance matrix against a region list """
    tuples = [r.as_tuple() for r in regions]
    return cdist(tuples, tuples, region_distance)


def clusterize(words, **kwargs):
    # TODO: write a cool docstring here
    db = DBSCAN(metric="precomputed", **kwargs)
    X = distance_matrix([Region.from_word(w) for w in words])
    labels = [int(l) for l in db.fit_predict(X)]
```

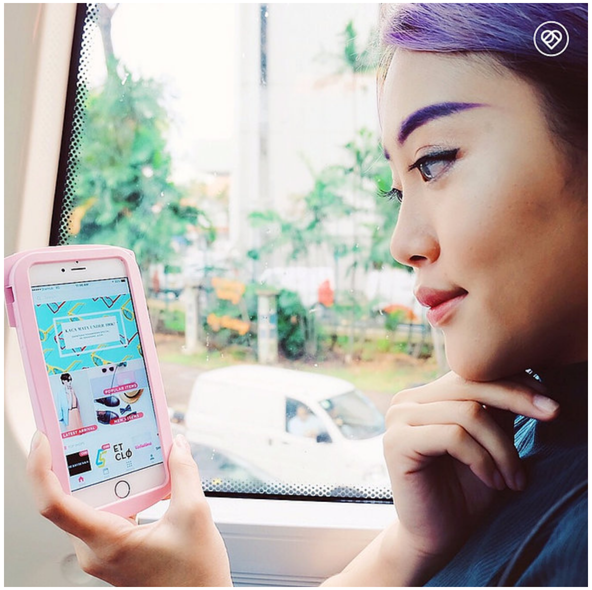
STORY

- Lyke - [12.2016 - 07.2017]
- SMACC - [10.2017 - present]

LYKE

LYKE

- E-commerce
- Mobile-only
- 50k+ users
- 2M downloads
- Top 10 Fashion Apps w Google Play Store



http://www.news.getlyke.com/single-post/2016/12/02/Introducing-the-New-Beautiful-LYKE

Now JollyChic Indonesia

GOOD PARTS

- Fast Growth
- A/B Testing
- Data-driven
- Product Manager, UI Designer, Mobile Dev, and tester - one body

CHALLENGES

- 50+ VMs in Amazon, 1 VM - 1 App, idle machine
- Puppet, hilarious (manual) deployment process
- Fear
- Forgotten components
- sometimes performance issues

SMACC

Hypatos

SMACC

- Machine Learning FinTech
- SaaS and API platform
- From Enterprise (Deutsche Bank, AoK) to SME
- Well-known FinTech Startup in Germany

STORY

- Legacy on AWS, experiments with AWS ECS :/
- Self-hosted K8S on ProfitBricks
- Get to Microsoft ScaleUp, welcome Azure
- Luckily - Azure-Kubernetes-Service

DIFFERENCE ☠

- Two teams in Berlin and Warsaw
- Me in Warsaw

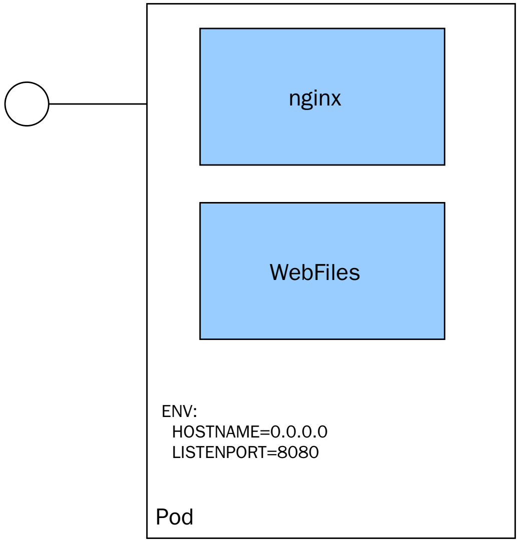
APPROACH

- Simplify, Simplify
- Hide K8S magic
- `git tag` driven Continoues Deployment

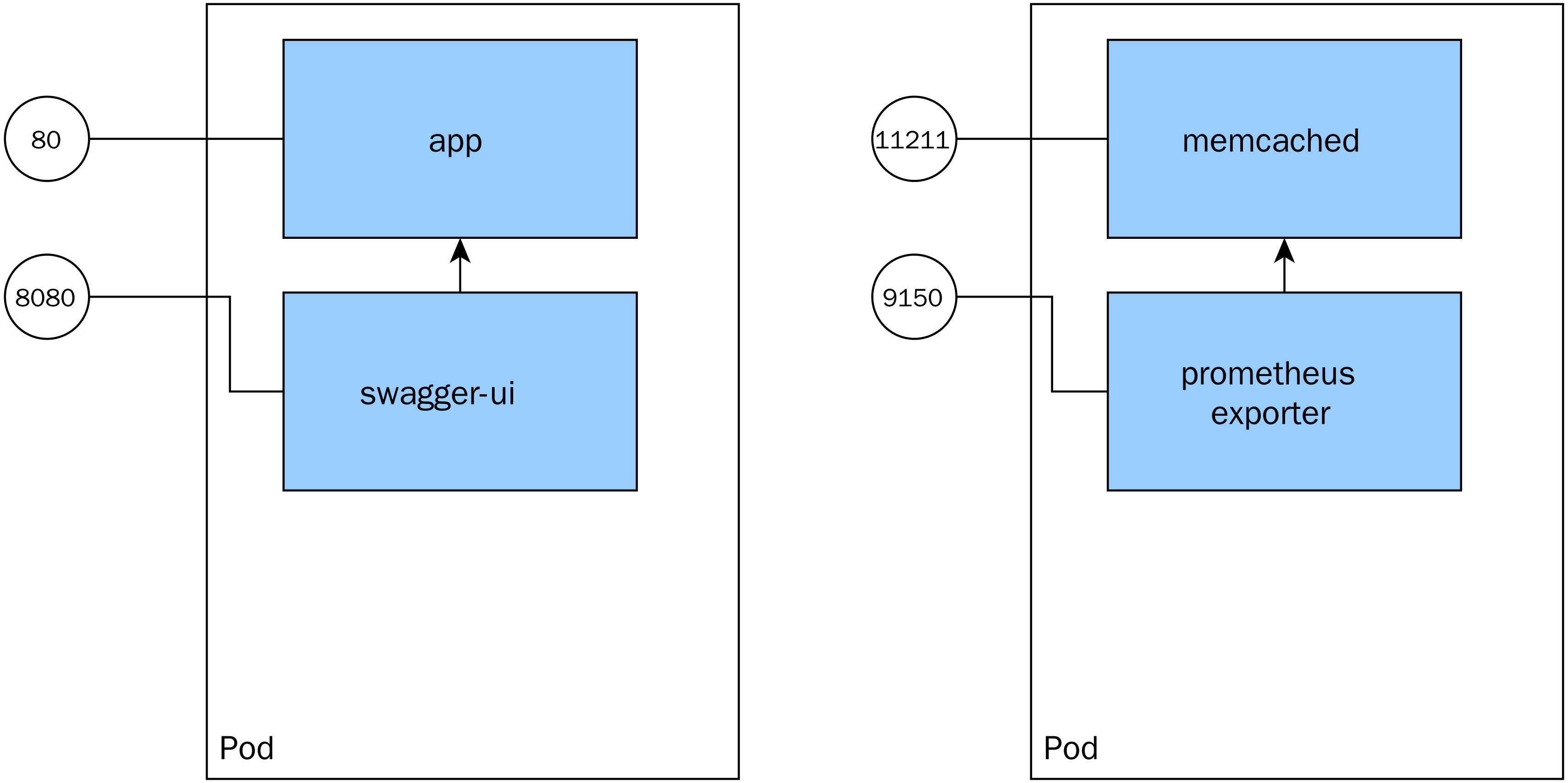
KUBERNETES CONCEPTS

PODS

- See each other on localhost
- Live and die together
- Can expose multiple ports

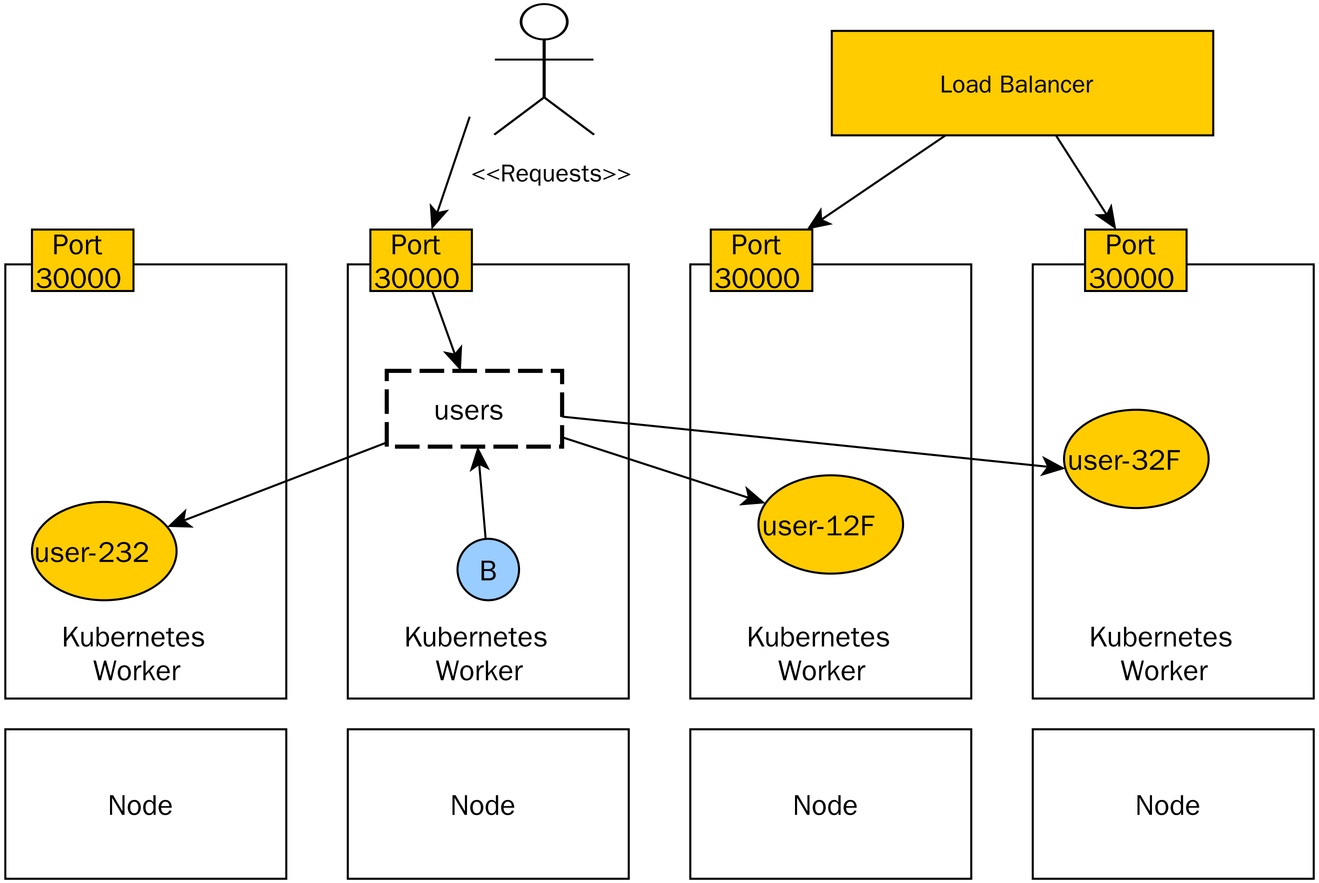


SIDE-CARS



80
8080
app
swagger-ui
Pod
11211
9150
memcached
prometheus exporter
Pod

LOAD BALANCING


Load Balancer
<<Requests>>
Port 30000
Port 30000
Port 30000
Port 30000
users
user-232
B
user-12F
user-32F
Kubernetes Worker
Kubernetes Worker
Kubernetes Worker
Kubernetes Worker
Node
Node
Node
Node