

CNCF Annual Survey 2021

ARE YOU USING KUBERNETES?

■ Yes, in production ■ Yes, in test poc ■ Not yet, but we are evaluating ■ No ■ Not sure

AFRICA



AUSTRALIA & OCENIA



N. AMERICA



ASIA



EUROPE



S.& C. AMERICA



kubernetes



A Quick Introduction to



Your hosts

Cornelius Schaub



- MSc in Computer Science
- Consultant at ipt since Nov 2023
- Projects at UCC and BIT
- Currently building automations to provide services on private Cloud

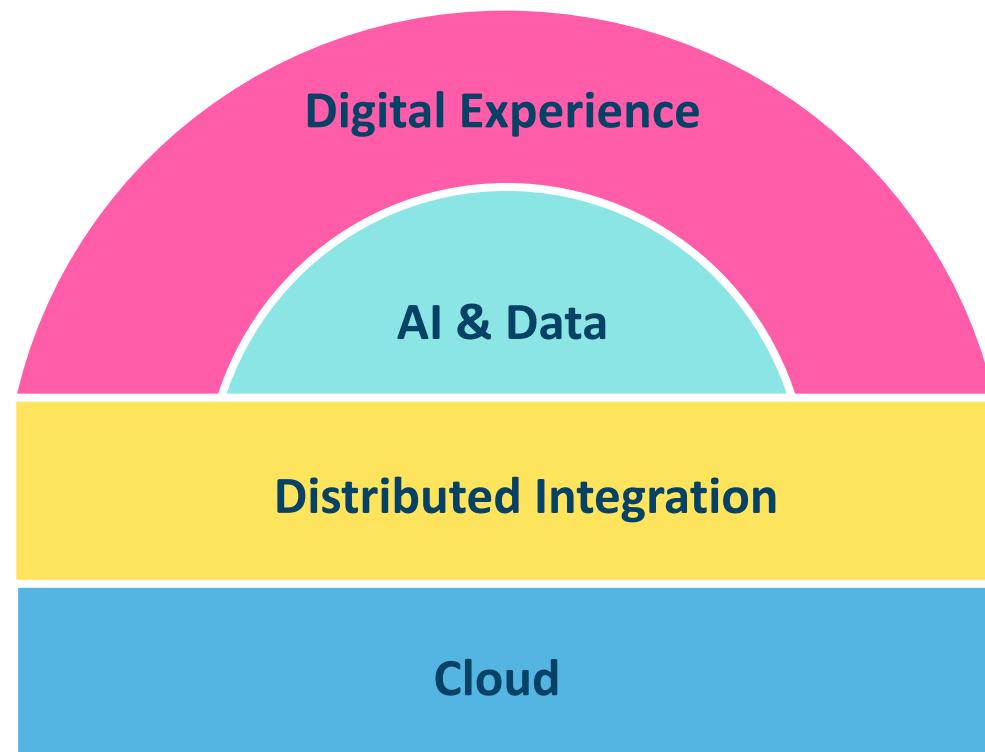
Benjamin Bürgisser



- MSc in Computer Science
- Consultant at ipt since Oct 2018
- Projects at LGT, Helsana, BIT, SBB
- Working with Kubernetes for ~5 years
- Certified Kubernetes Application Developer



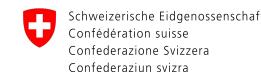
Partners:



Customers:



Vontobel



Timetable

13:00 - 13:50

- Introduction to Kubernetes and Containers
- Lab 1 

14:00 - 15:00

- Advanced topics
- Lab 2 

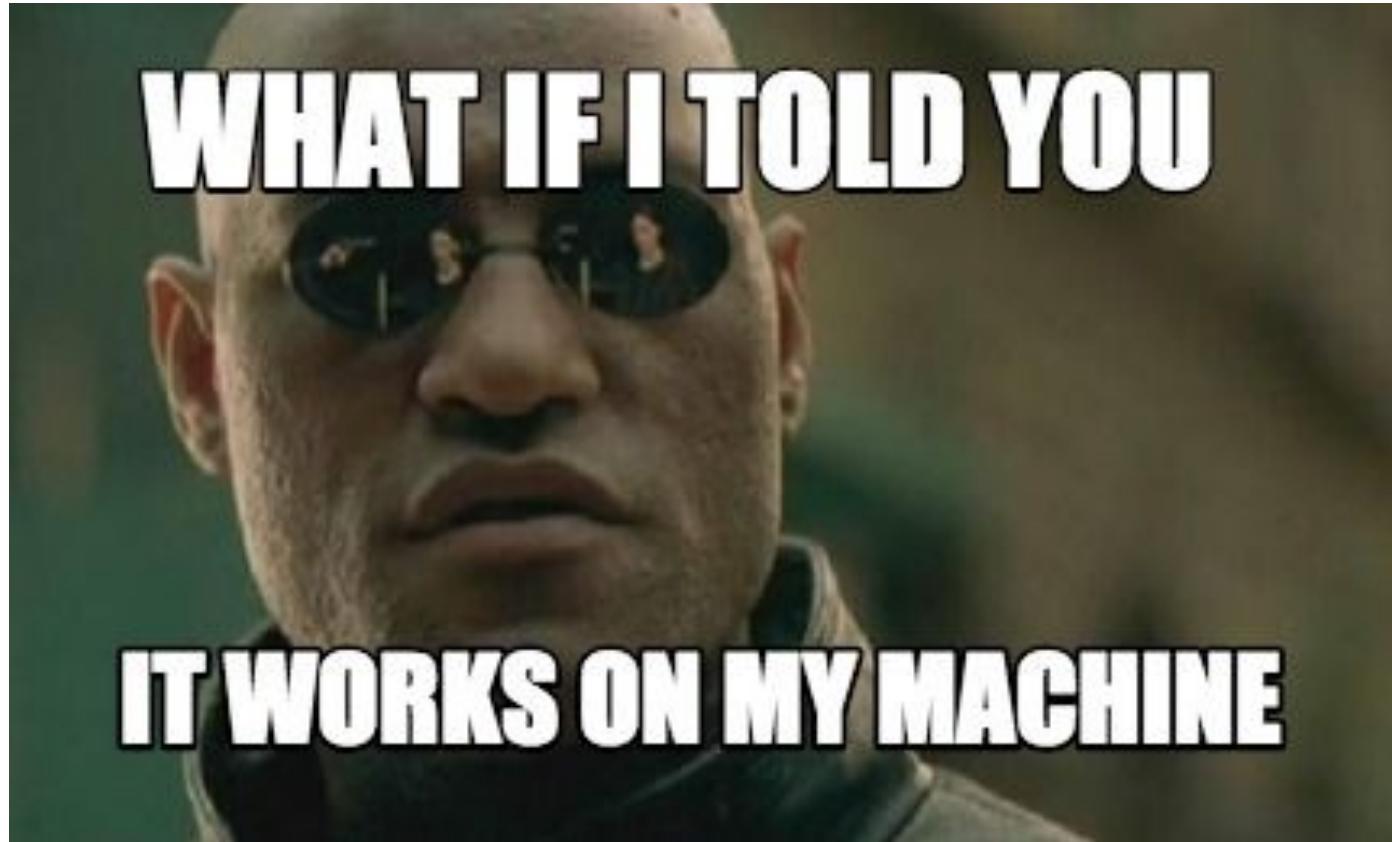
15:10 - 16:00

- High Availability and Conclusions
- Lab 3 

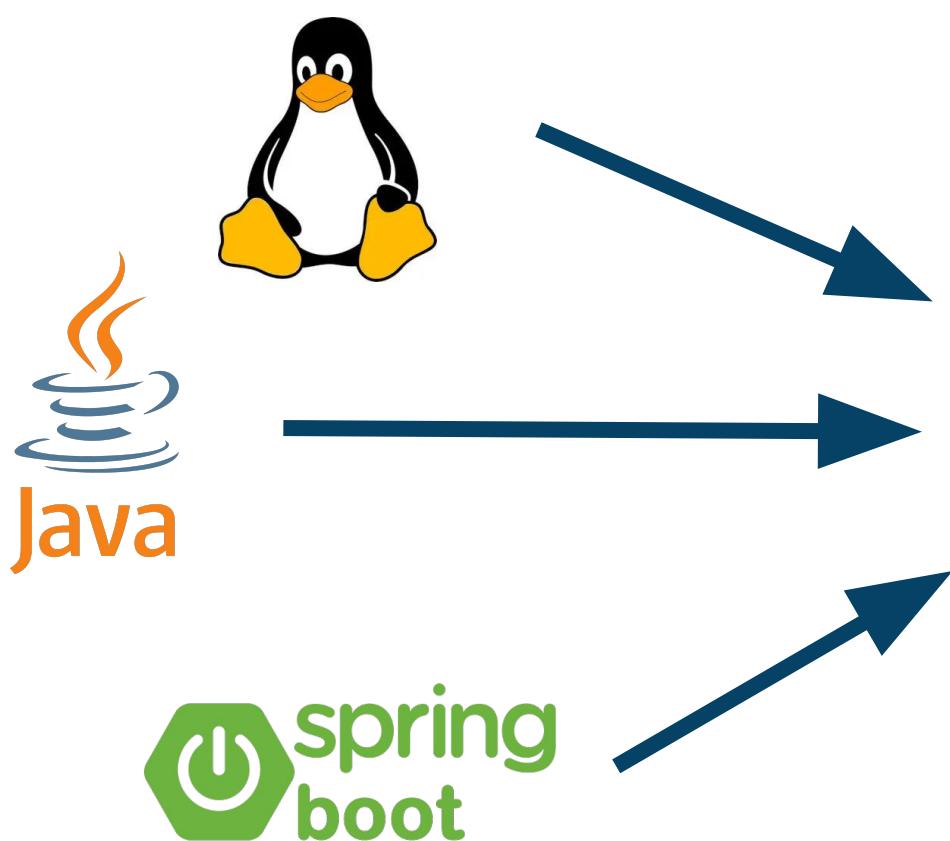
Let's Start with Docker



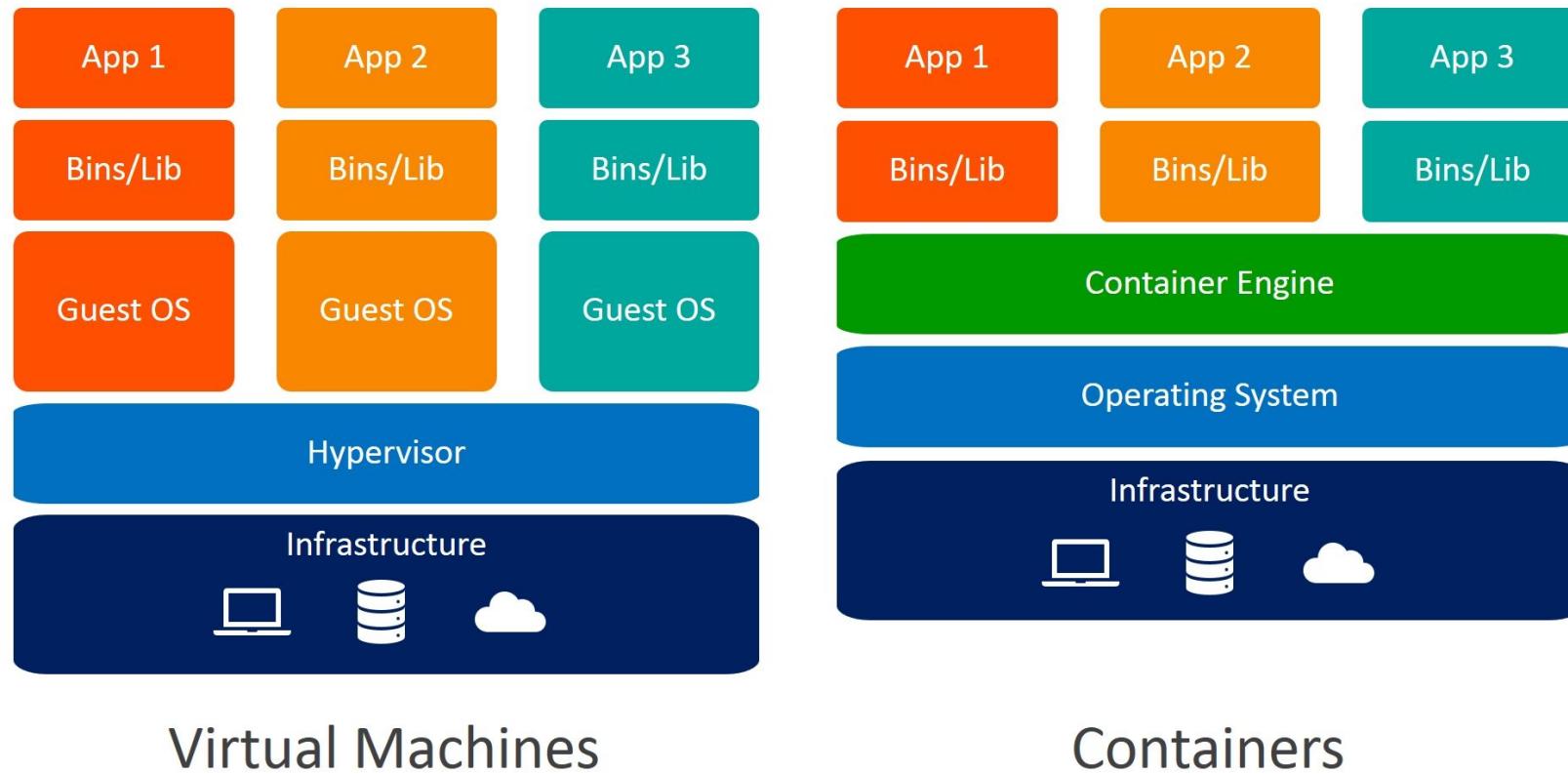
Has This Ever Happened to You?



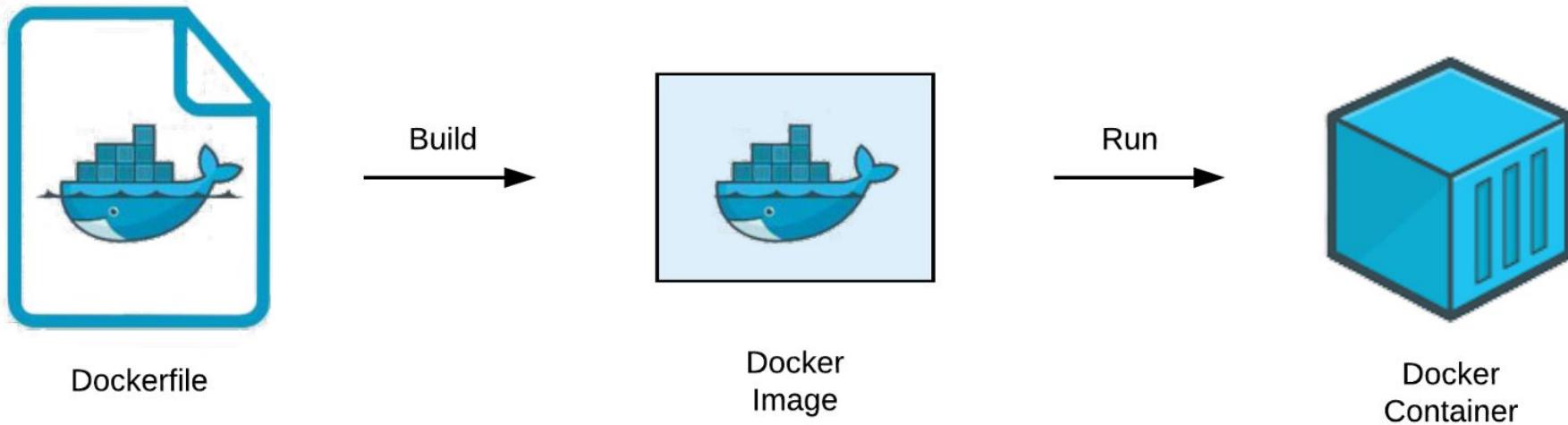
Docker: Everything Our App Needs in One Container



Docker: Shared kernel → Lightweight



Docker: Dockerfile



Docker: Dockerfile

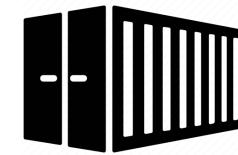
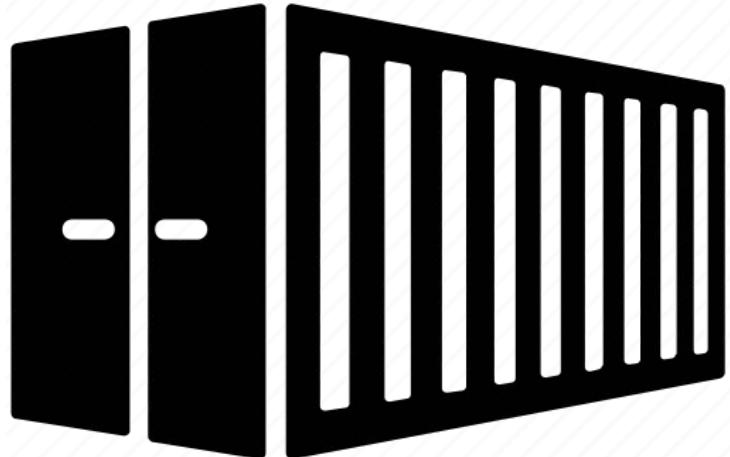
```
# Select base image
FROM ubuntu:16.04

# Install some requirements
RUN apt-get update &&
    apt-get install python3.8

# Copy our code into the image
COPY /src/build /myapp/
COPY config.conf /myapp/

# Set the default command to run when starting the container
CMD ["/myapp/run", "-c", "config_file=/myapp/config.conf"]
```

Cloud Native Development - Small and Stateless



Kubernetes: Let's Load Those Containers onto a Ship



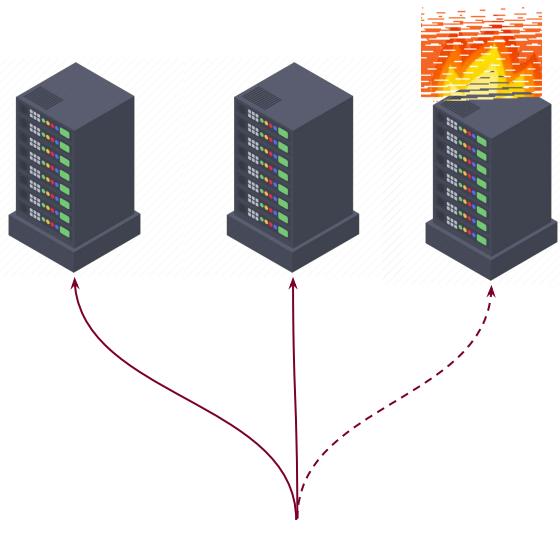
Kubernetes (helmsman, pilot): Steers the Ship



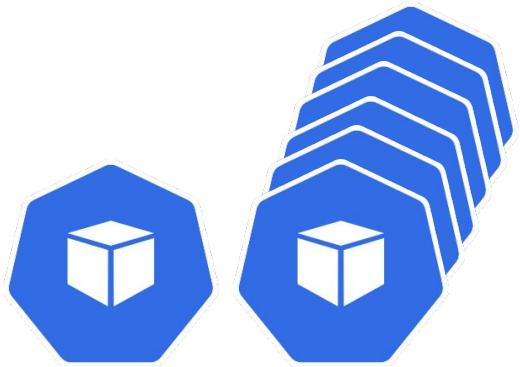
The Birth of Kubernetes (aka K8s)



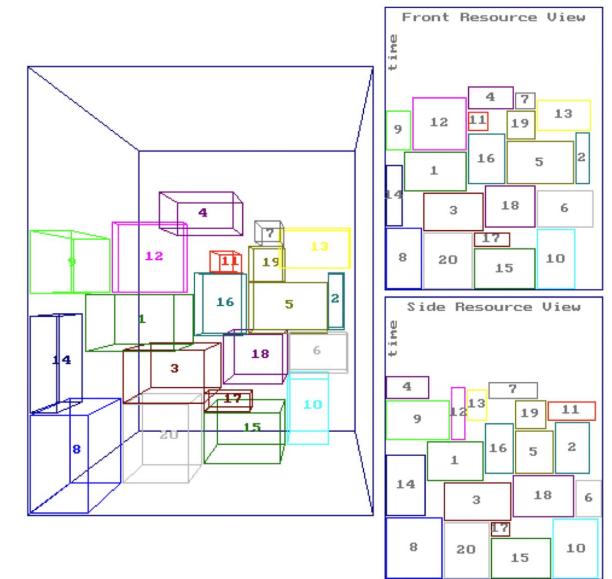
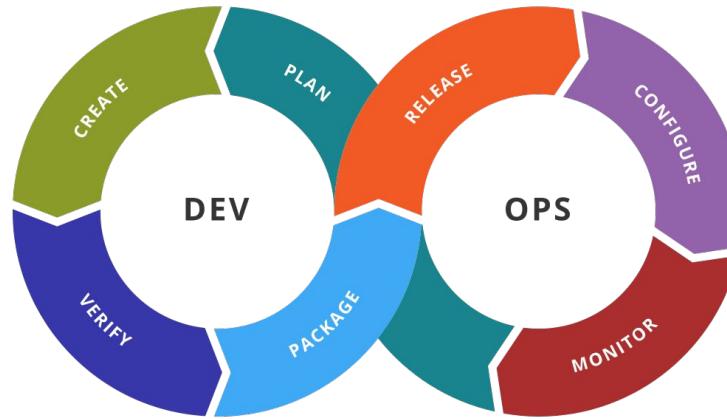
The Challenges Driving Kubernetes



Fault Tolerance



Scalability



Scheduling of Resources

Kubernetes: The Master of Deployment

Declarative Deployments

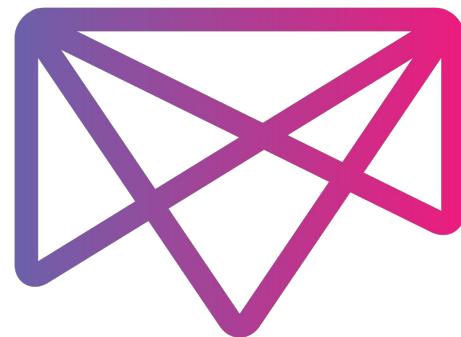
```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: my-deployment
  name: my-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: my-deployment
  template:
    metadata:
      labels:
        app: my-deployment
    spec:
      containers:
        - name: busybox
          image: busybox
```

Kubernetes goes Open-Source



open source

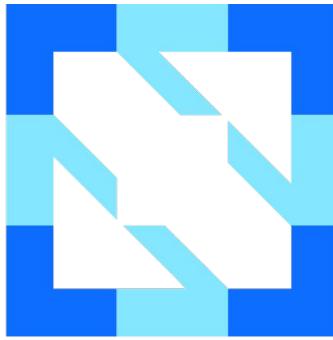
K8s open sourced @DockerCon June 2014



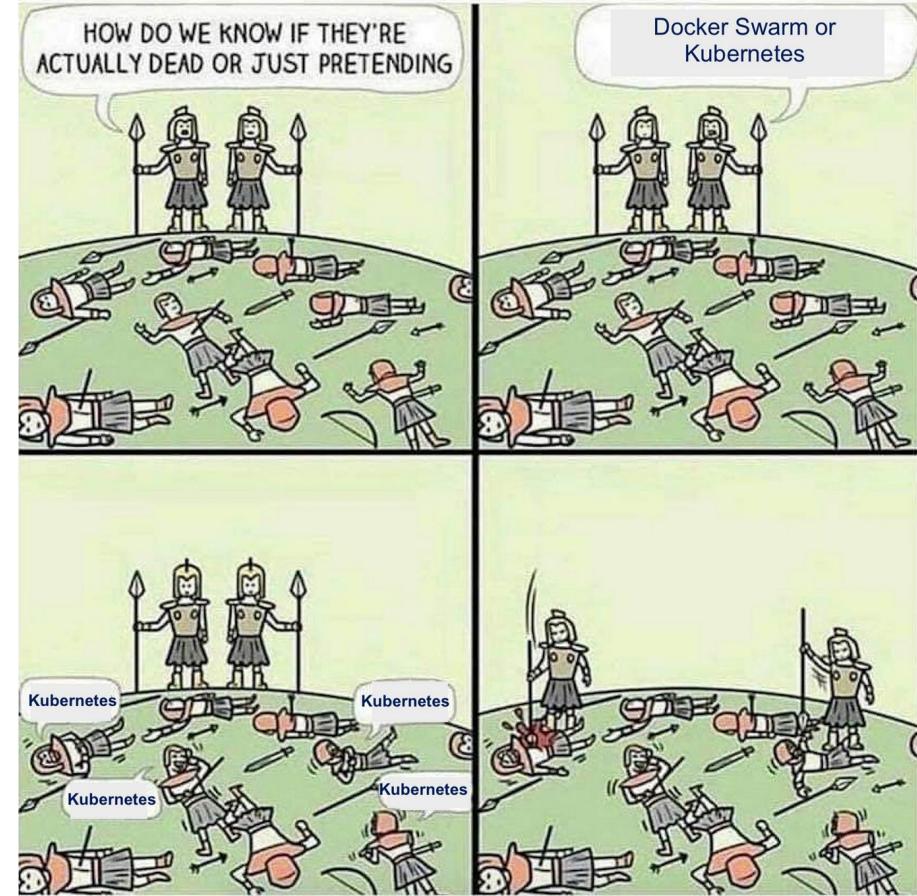
MESOSPHERE

Closed Source but very successful

Uniting Forces under CNCF



**CLOUD NATIVE
COMPUTING FOUNDATION**



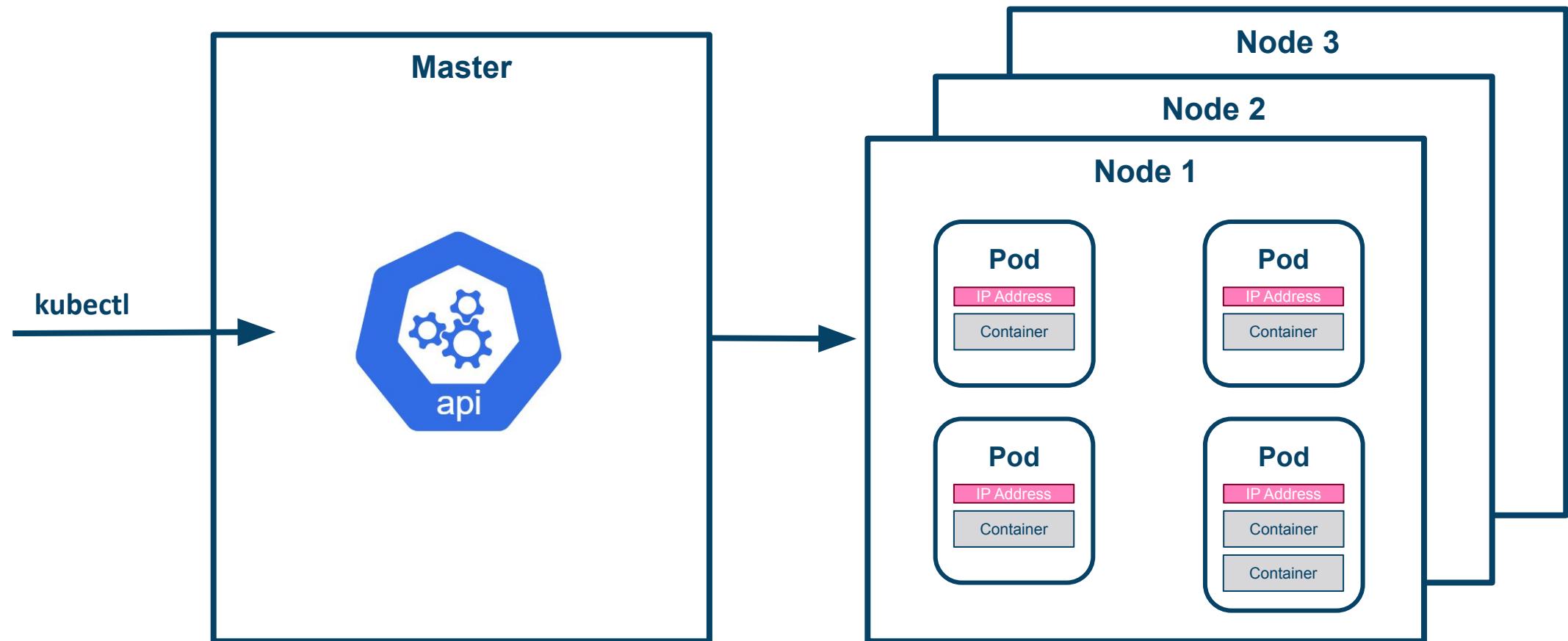
First Big Success



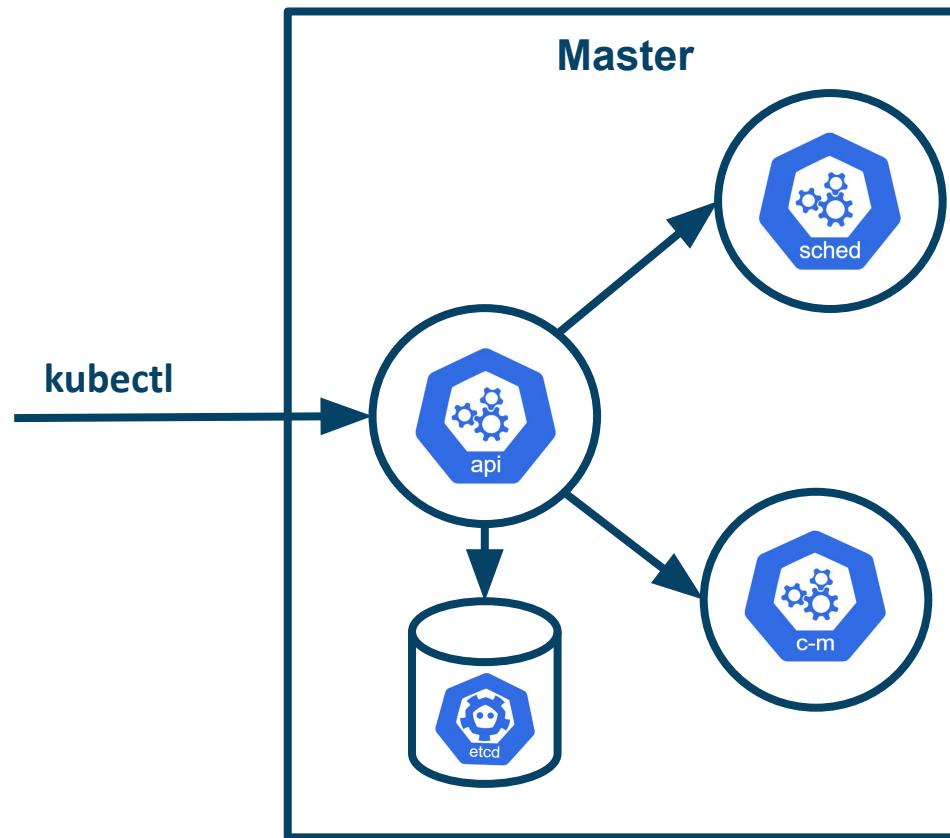
How does Kubernetes work?



Kubernetes Architecture



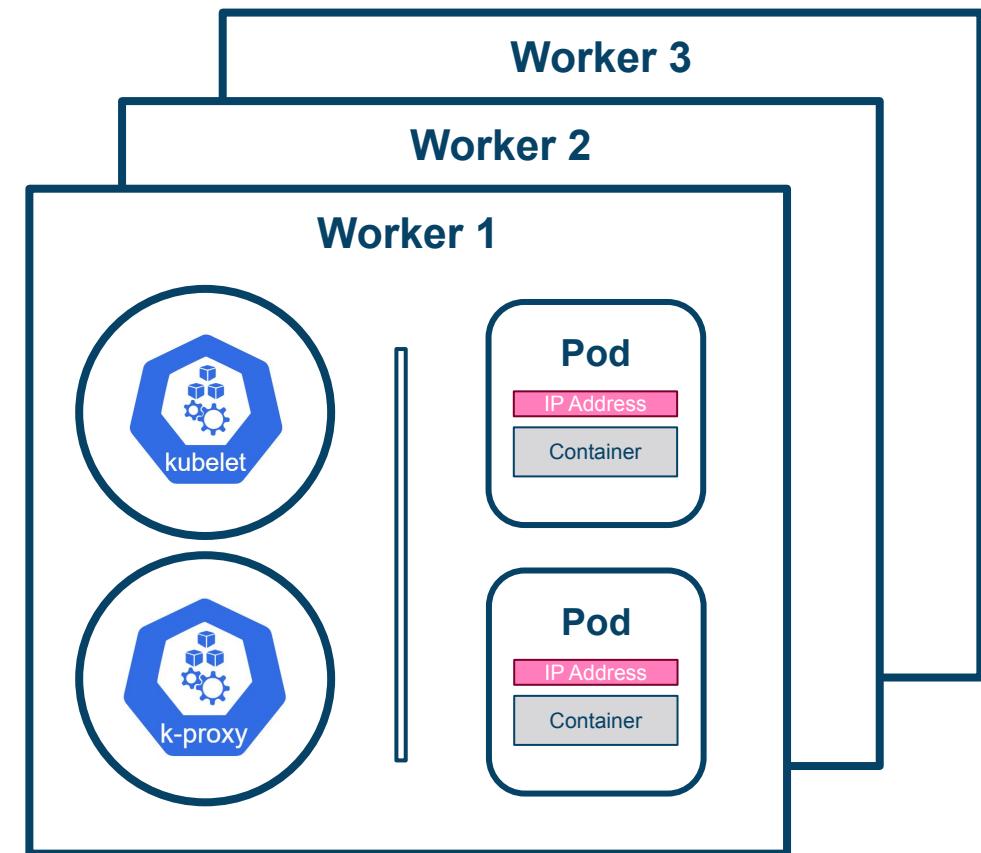
Kubernetes Architecture: Master



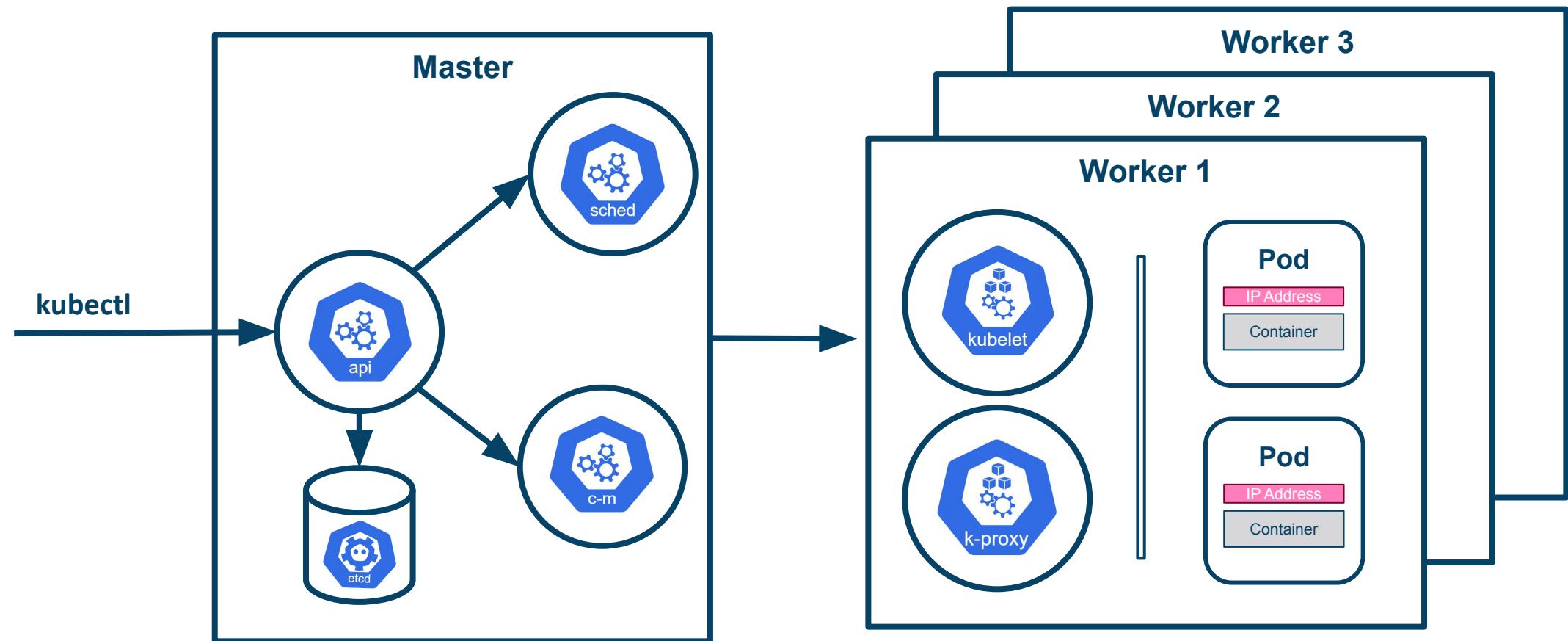
- **kube-apiserver**
 - Exposes API
 - Validates and configures
- **etcd**
 - Configuration database
- **kube-controller-manager**
 - Controls the processes
 - Detects state changes (e.g. crashing pods)
- **kube-scheduler**
 - Pushes workload to a worker node
 - Decides where the next pod is scheduled

Kubernetes Architecture: Workers

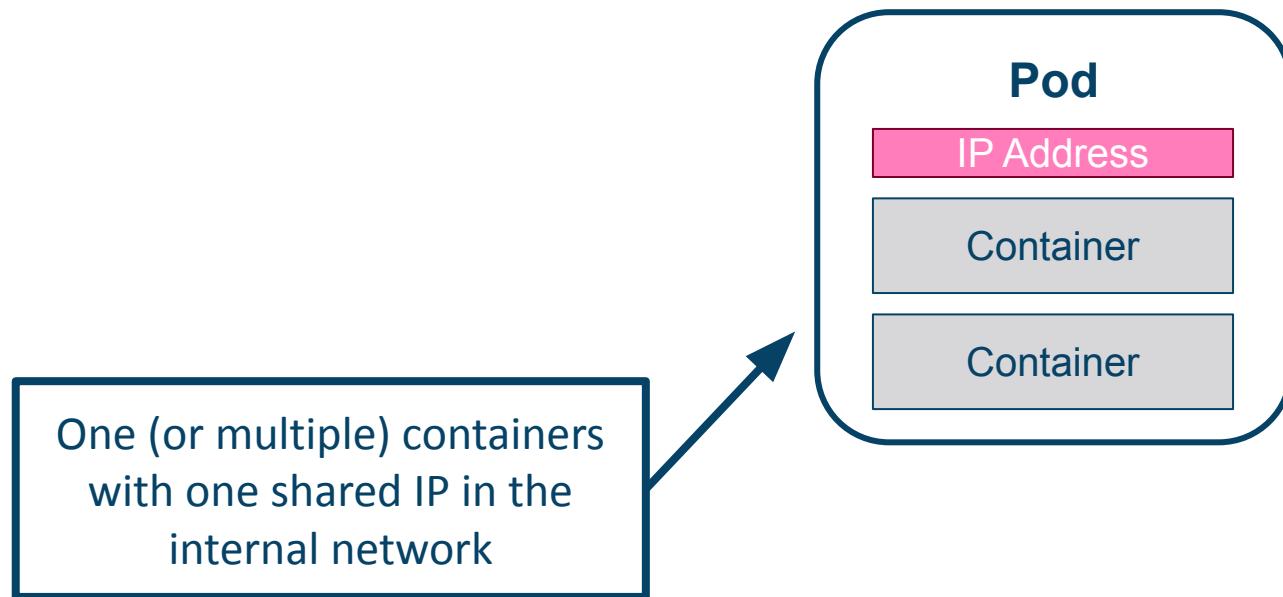
- **kubelet:**
 - Kubernetes agent
 - Runs the pods with its containers
- **kube-proxy**
 - Maintains network rules on nodes
 - Reflects services as defined in the Kubernetes AP
 - Forwards traffic to the pods



Kubernetes Architecture



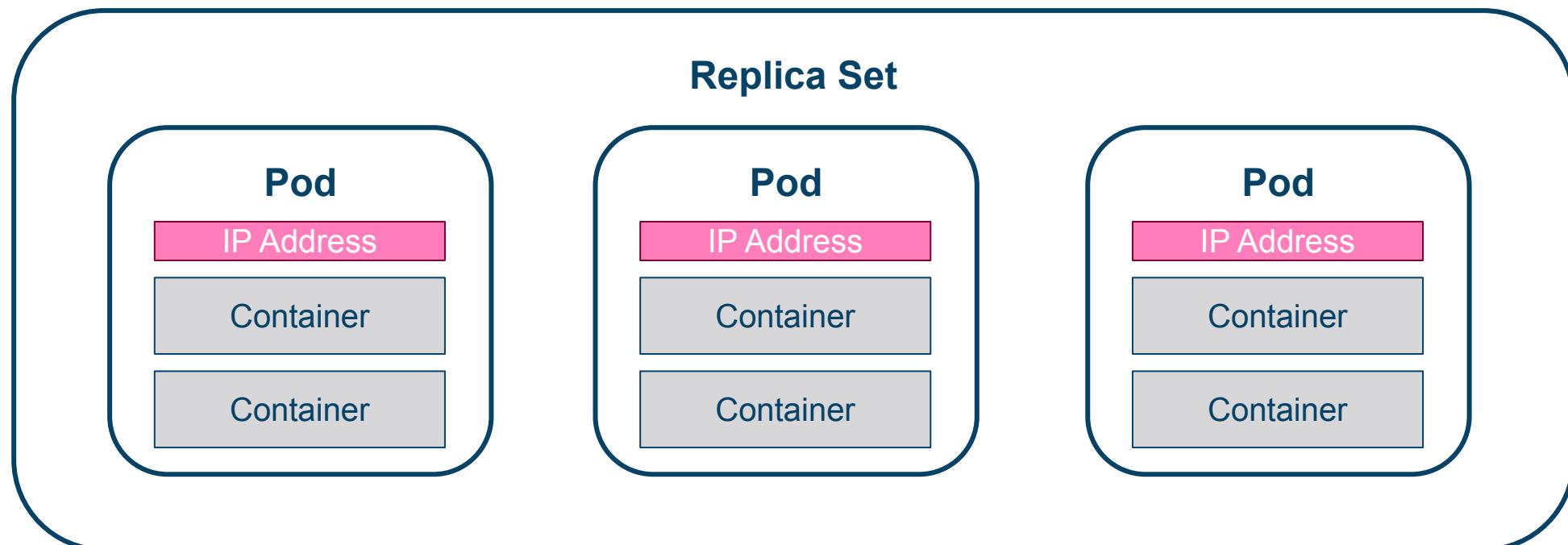
Kubernetes Resources: Pods



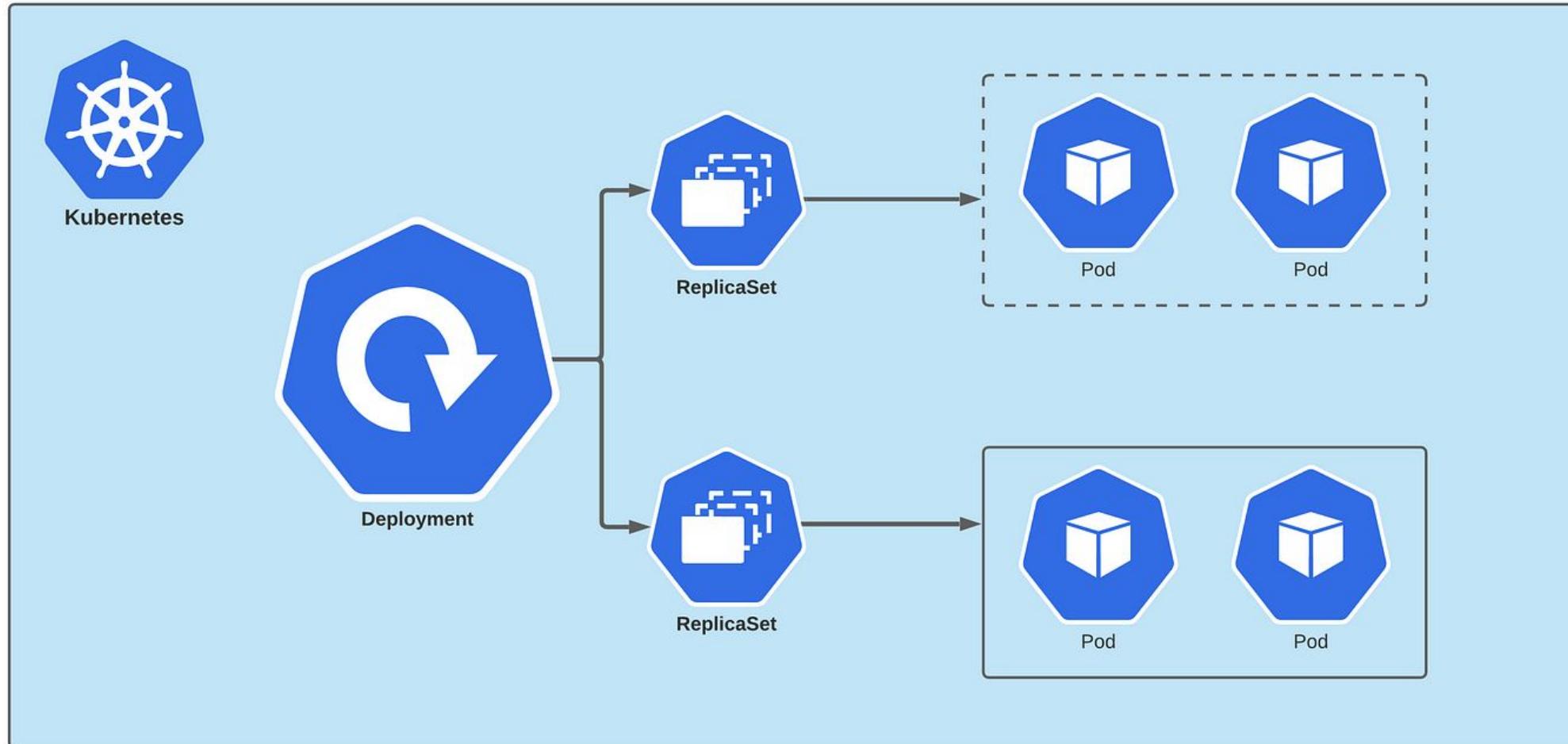
Kubernetes Resources: Pods

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: my-pod
    name: my-pod
spec:
  containers:
    - name: my-pod
      image: busybox
```

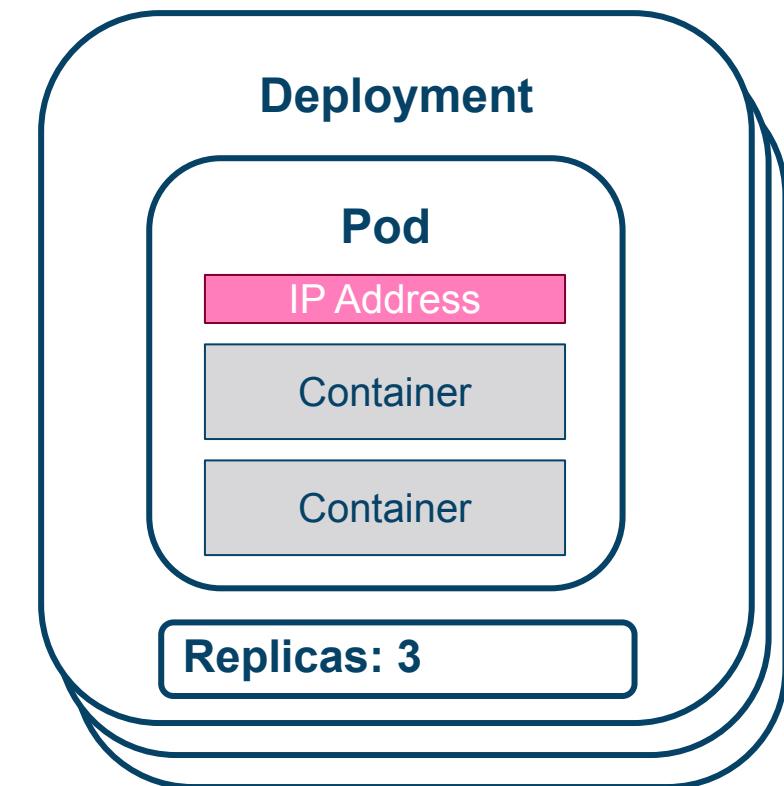
Kubernetes Resources: Replica Sets



Kubernetes Resources: Deployments



Kubernetes Resources: Deployments



Kubernetes Resources: Deployments

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: my-deployment
  name: my-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: my-deployment
  template:
    metadata:
      labels:
        app: my-deployment
    spec:
      containers:
        - name: busybox
          image: busybox
```

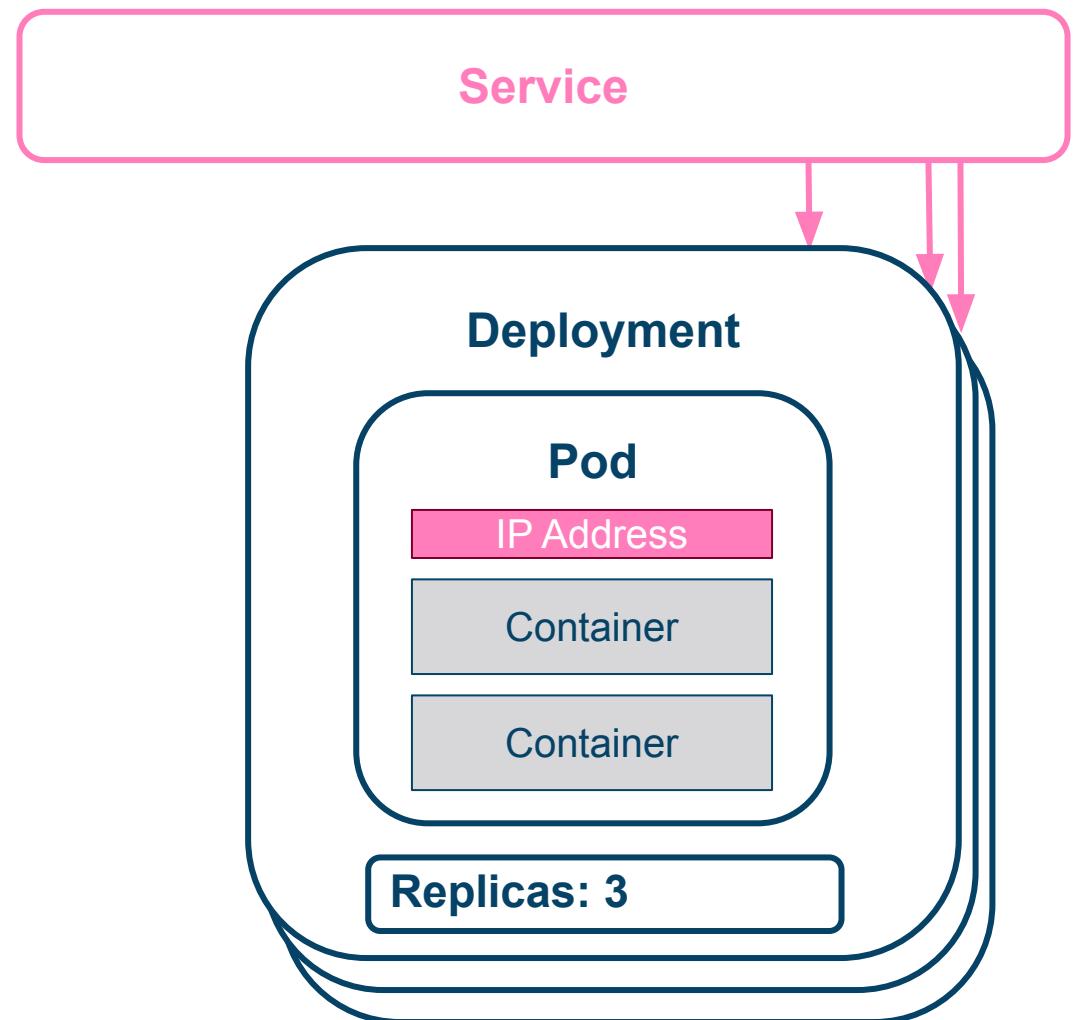
Kubernetes Resources: Deployments

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: my-deployment
  name: my-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: my-deployment
  template:
    metadata:
      labels:
        app: my-deployment
    spec:
      containers:
        - name: busybox
          image: busybox
```

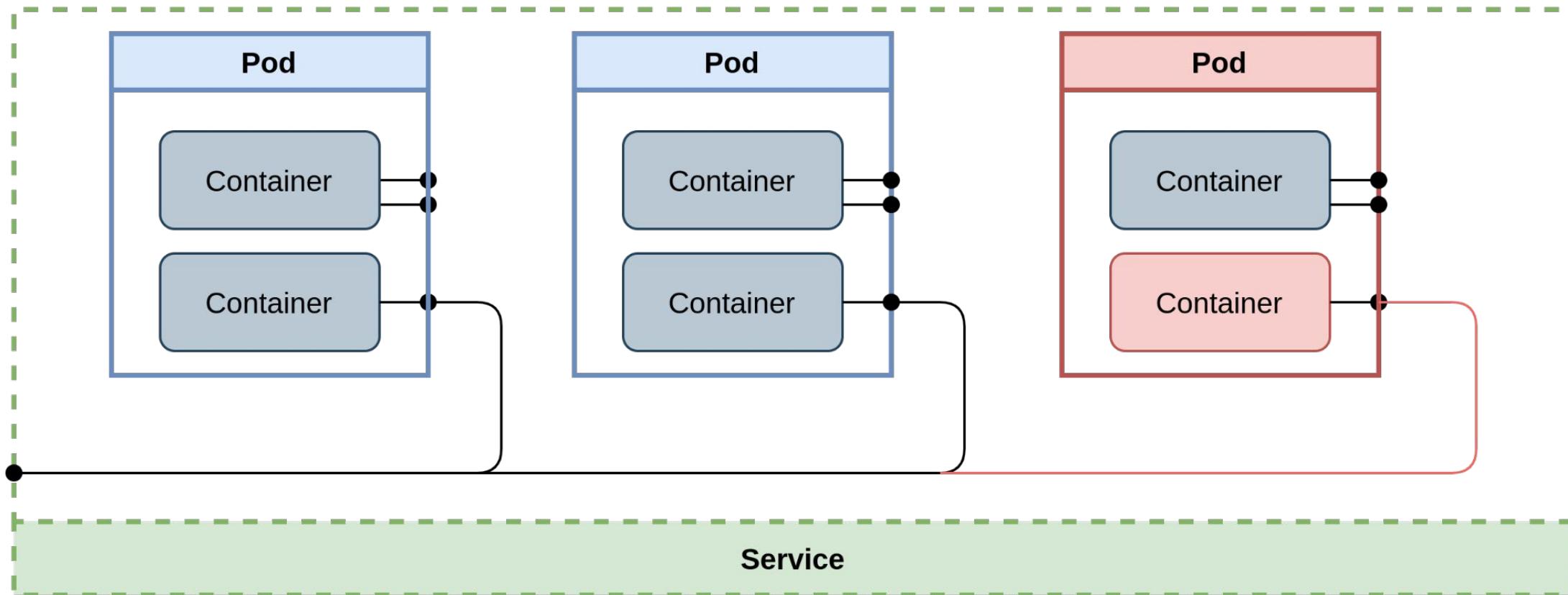
Kubernetes Resources: Deployments

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: my-deployment
  name: my-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: my-deployment
  template:
    metadata:
      labels:
        app: my-deployment
    spec:
      containers:
        - name: busybox
          image: busybox
```

Kubernetes Resources: Services



Traffic Routing - Kubernetes Services



Traffic Routing - Kubernetes Services

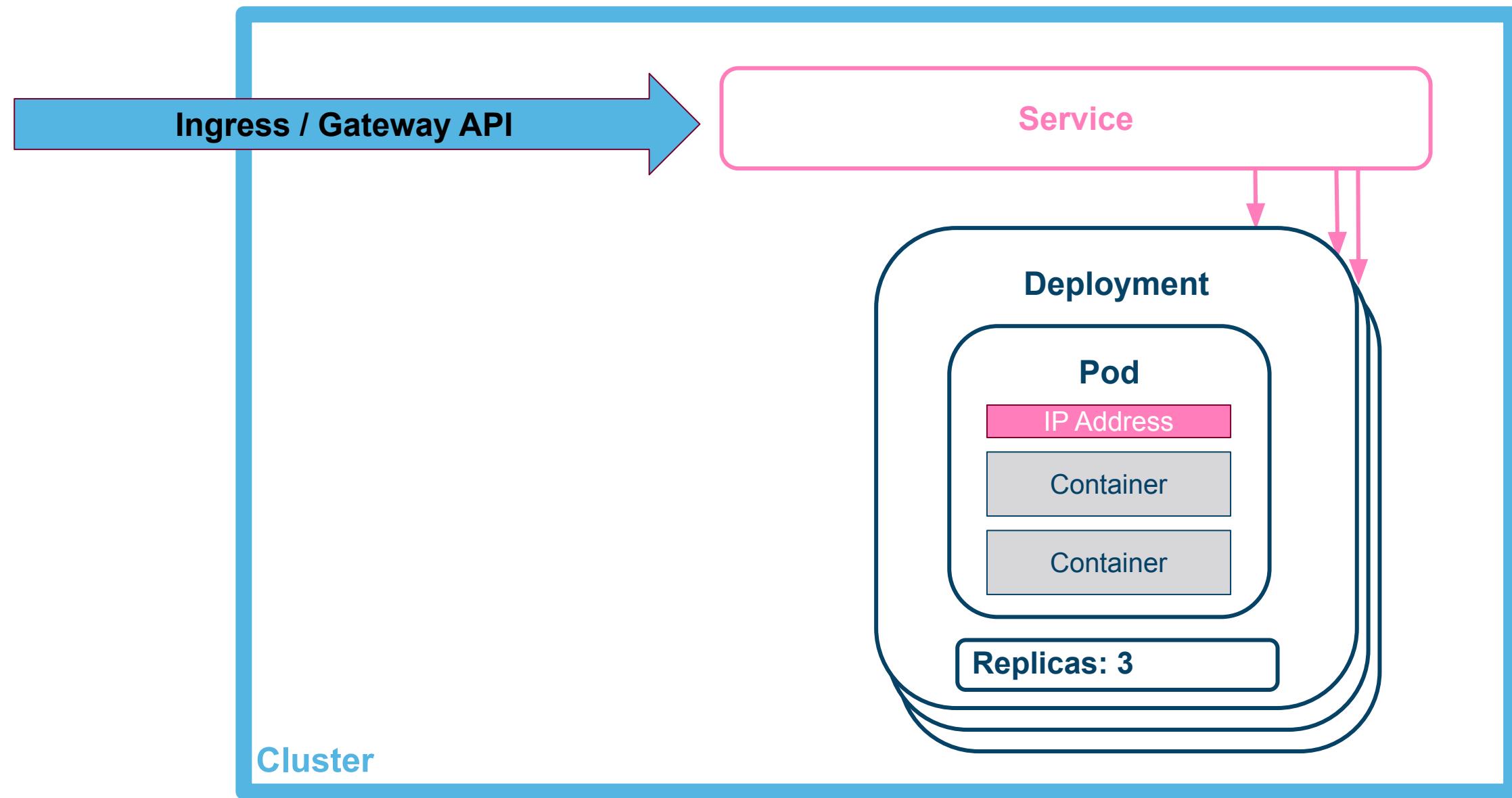
```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: my-deployment
    name: my-deployment
spec:
  ports:
  - port: 8080
    protocol: TCP
    targetPort: 80
  selector:
    app: my-deployment
```

Traffic Routing - Kubernetes Services

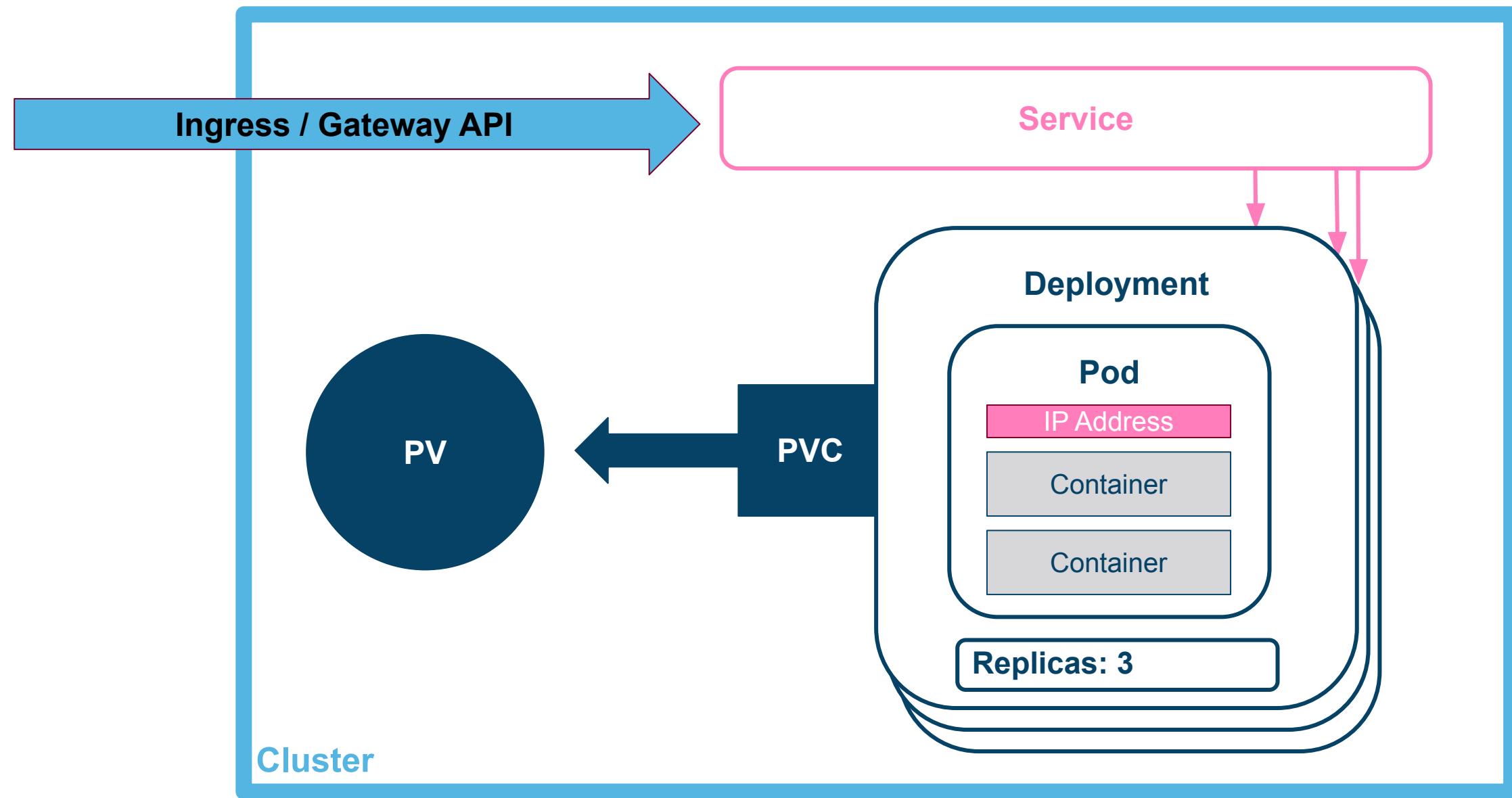
```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: my-deployment
    name: my-deployment
spec:
  ports:
    - port: 8080
      protocol: TCP
      targetPort: 80
  selector:
    app: my-deployment
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: my-deployment
    name: my-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: my-deployment
  template:
    metadata:
      labels:
        app: my-deployment
    spec:
      containers:
        - name: busybox
          image: busybox
```

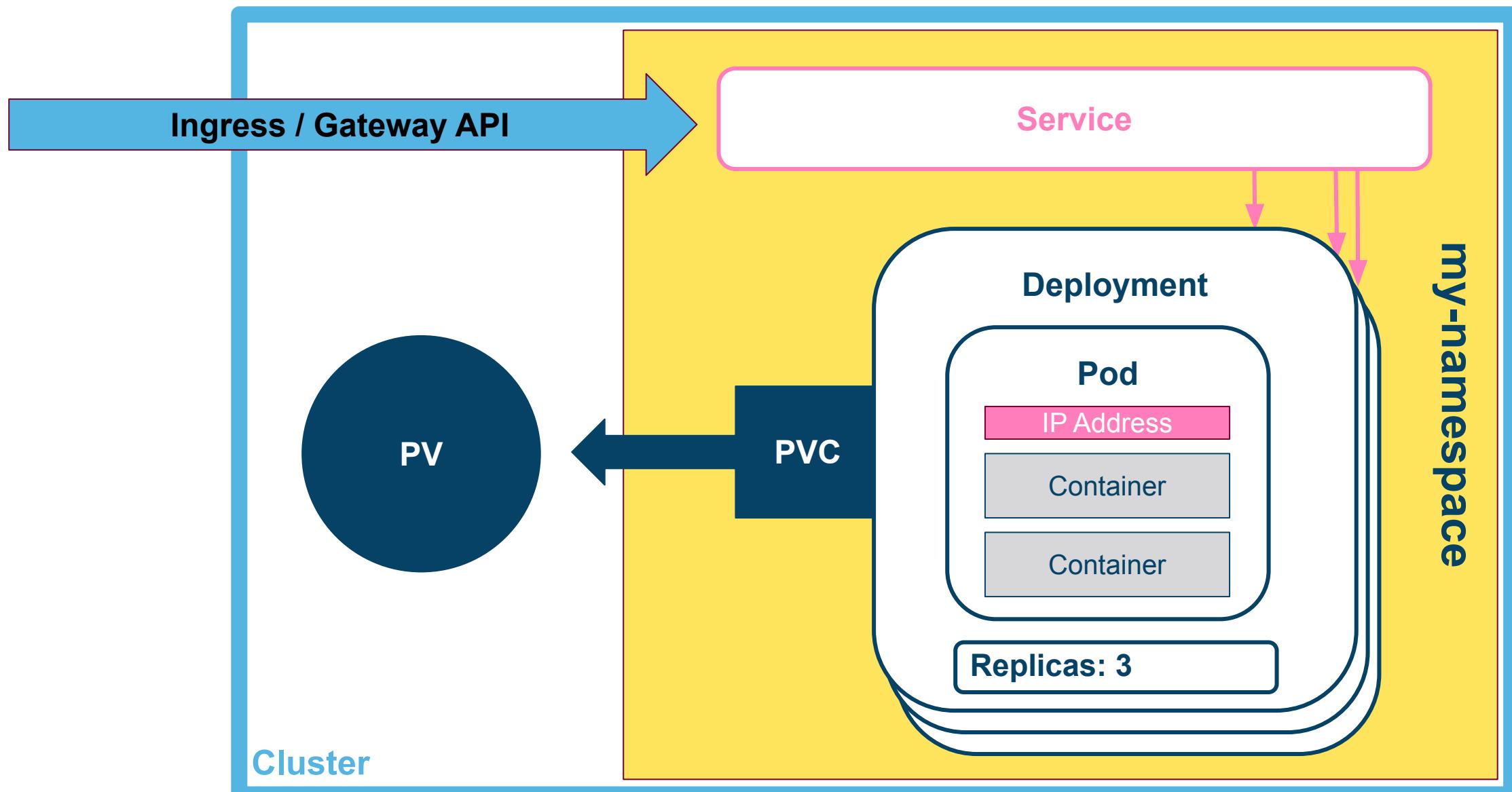
Kubernetes Resources: Ingress / Gateway API



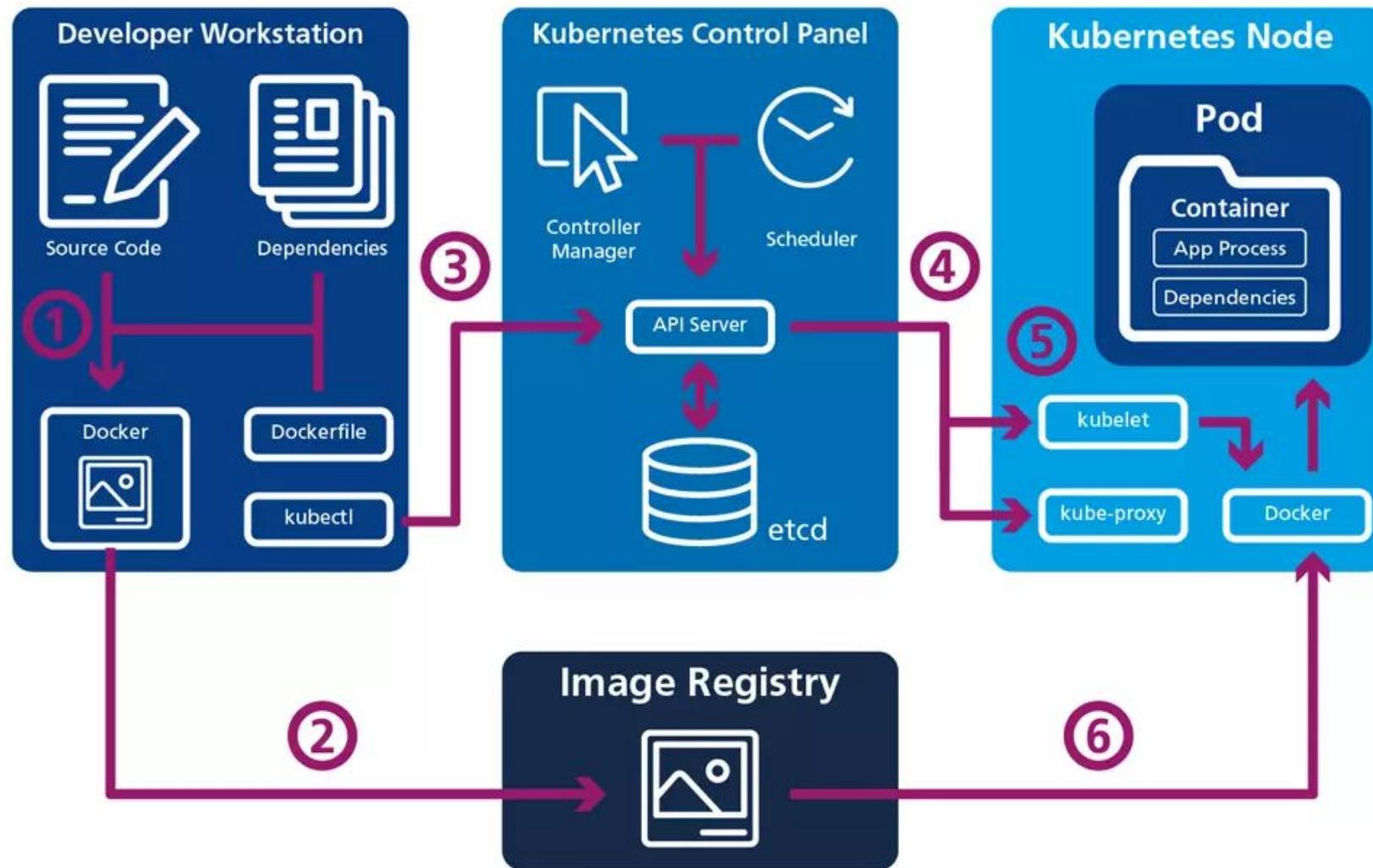
Kubernetes Resources: Persistent Volumes (PVs)



Kubernetes Resources: Namespaces



Kubernetes Workflow Summarized



kubectl: Kubernetes Command Line Tool

```
# Create namespace
$ kubectl create namespace my-namespace

# Create deployment with 3 replicas
$ kubectl create deployment my-dep --image=nginx
--replicas=3 -n my-namespace

# Create service
$ kubectl expose deployment my-dep --port=80
--target-port=8000

# Show pods, deployment and service
$ kubectl get pods -n my-namespace
$ kubectl get deployments -n my-namespace
$ kubectl get services -n my-namespace

...
```



<https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands>

Questions?



Let's try this ourselves

Kubectl:

1. Download kubectl:

```
winget install kubectl  
brew install kubectl
```

2. Download config file into the .kube folder in your home directory

- a. Password: **viscon2024**

3. Run a command to test if it works:

```
kubectl get ns
```

Lab exercises: <https://github.com/iptch/kubernetes-hands-on>

Lab 1

Lab exercises:

<https://github.com/iptch/kubernetes-hands-on>



Some Advanced Topics



ConfigMaps

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: game-demo
data:
  # property-like keys; each key maps to a simple value
  player_initial_lives: "3"
  ui_properties_file_name: "user-interface.properties"

  # file-like keys
  game.properties: |
    enemy.types=aliens,monsters
    pLayer.maximum-lives=5
  user-interface.properties: |
    color.good=purple
    color.bad=yellow
    allow.textmode=true
```

Env variables

Mount
as
Volume

Container

Secrets

```
apiVersion: v1
kind: Secret
metadata:
  name: secret-dockercfg
type: kubernetes.io/dockercfg
data:
  .dockercfg: |  
eyJhdXRocyI6eyJodHRwczovL2V4YW1wbGUvdjEvIjp7ImF1dGgiOiJvcGVuc2VzYW1lIn19fQo=
```

```
apiVersion: v1
kind: Secret
metadata:
  name: my-password
data:
  PASSWORD: dmFsdWUtMg0KDQo=
```

Env variables

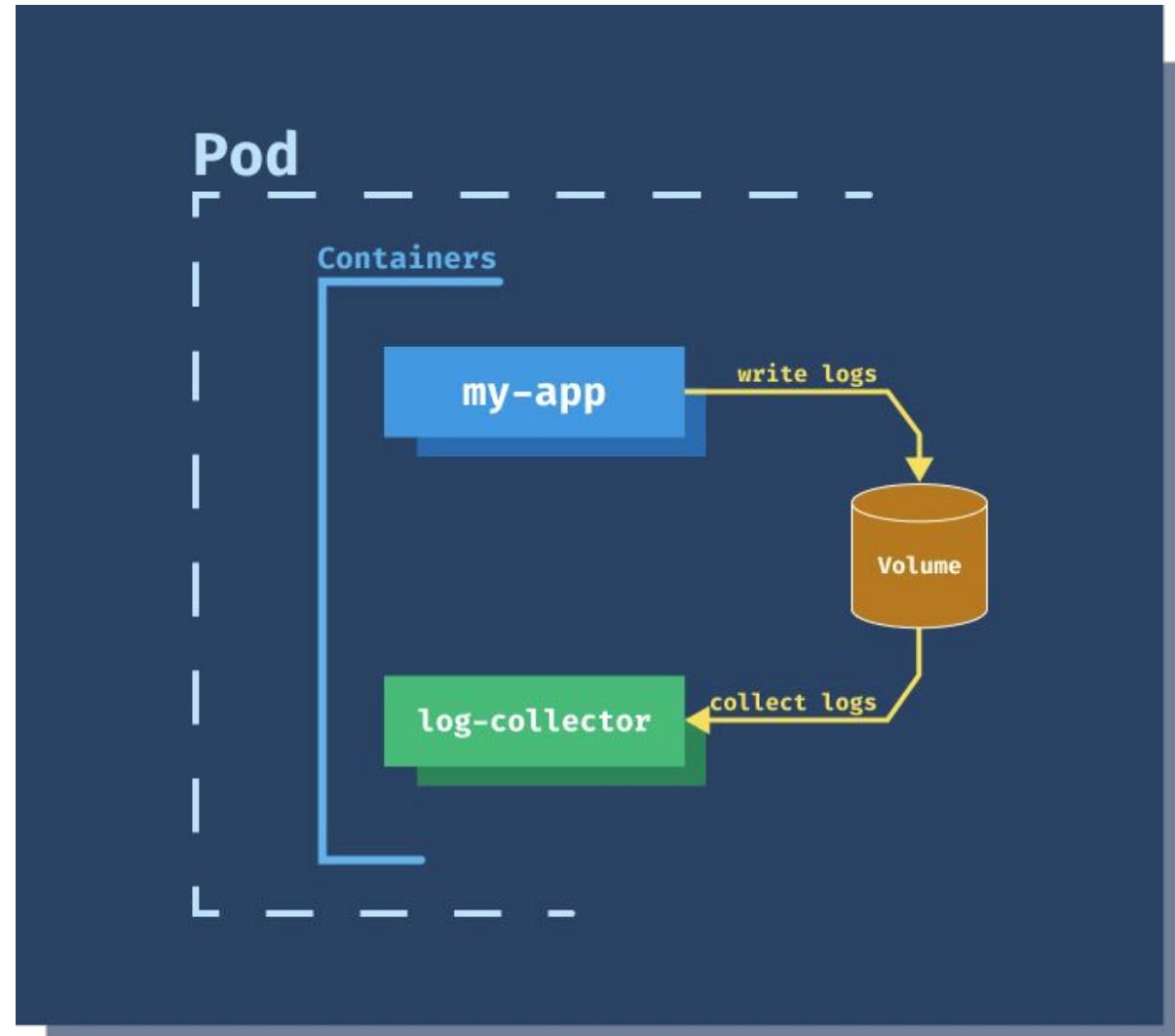
Mount
as
Volume

Container

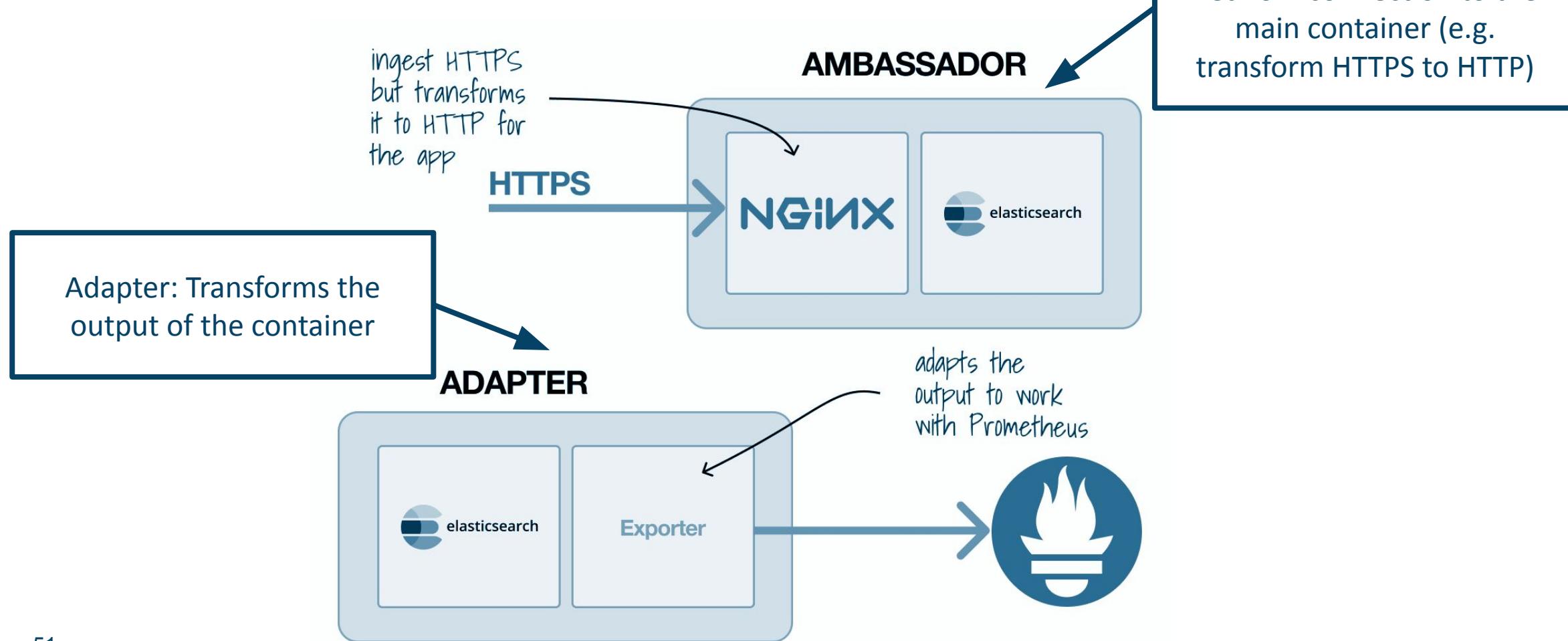
Multi Container Patterns

What if you can't change the code of your app?

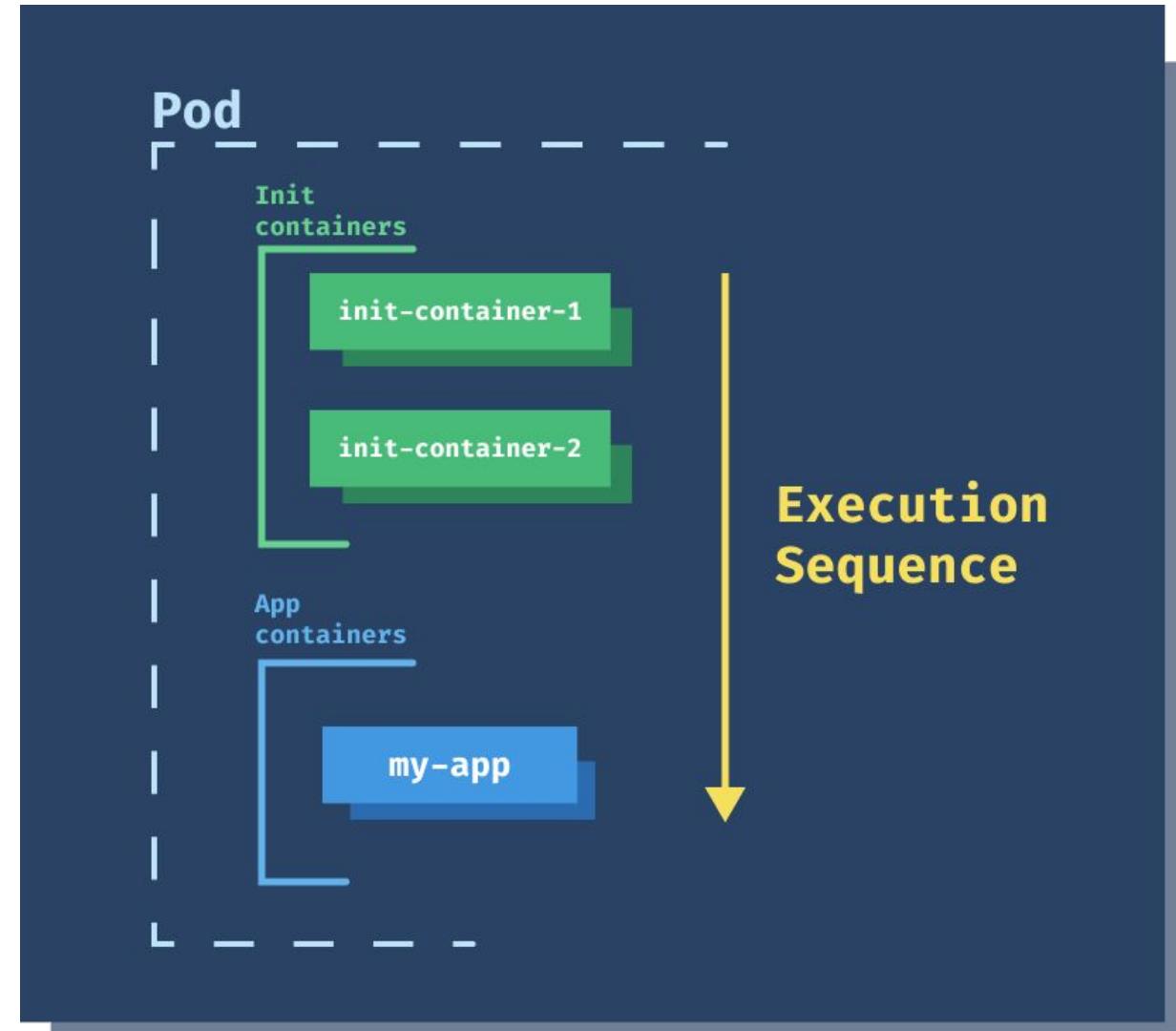
Sidecars



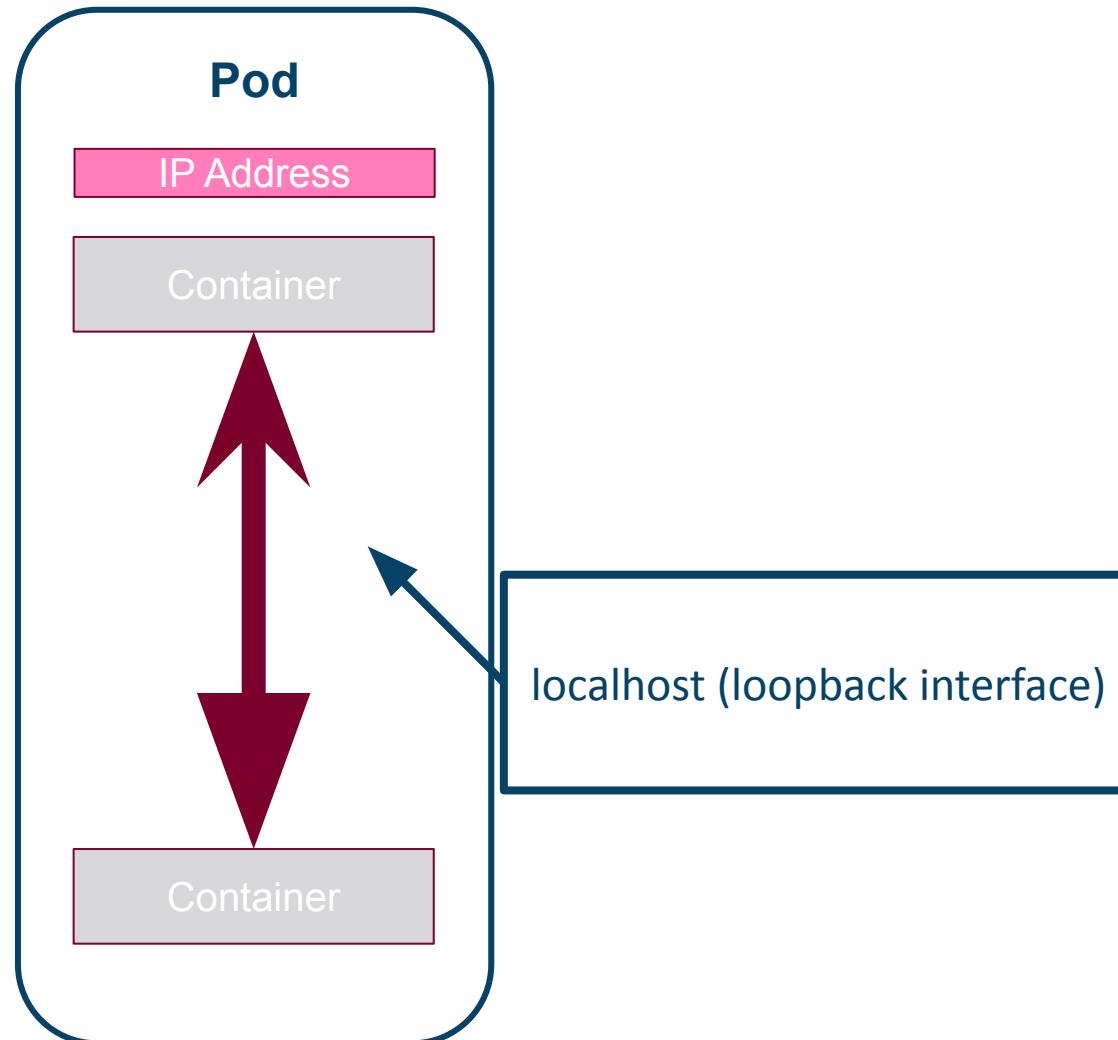
Adapter vs Ambassador



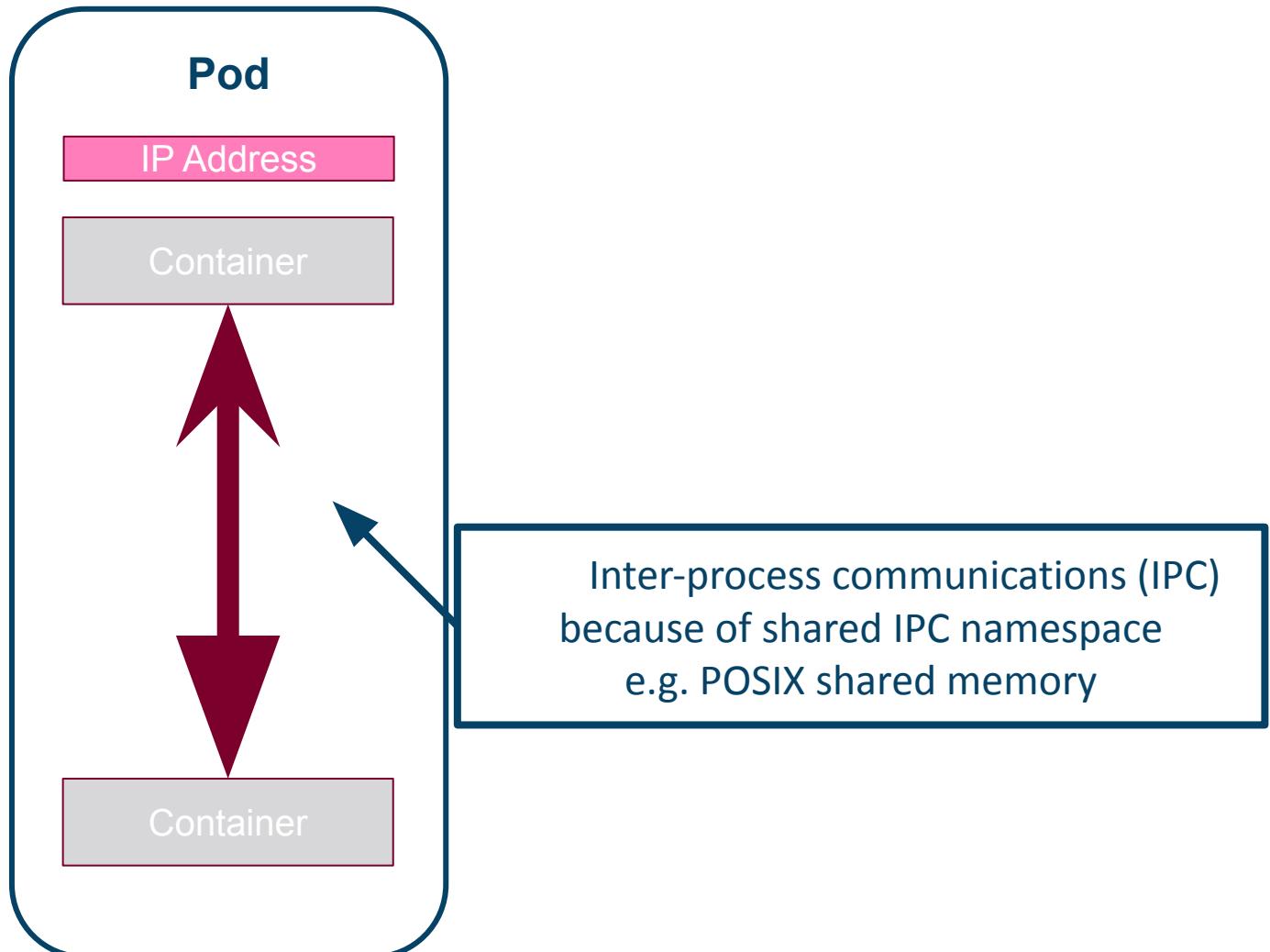
Init Containers



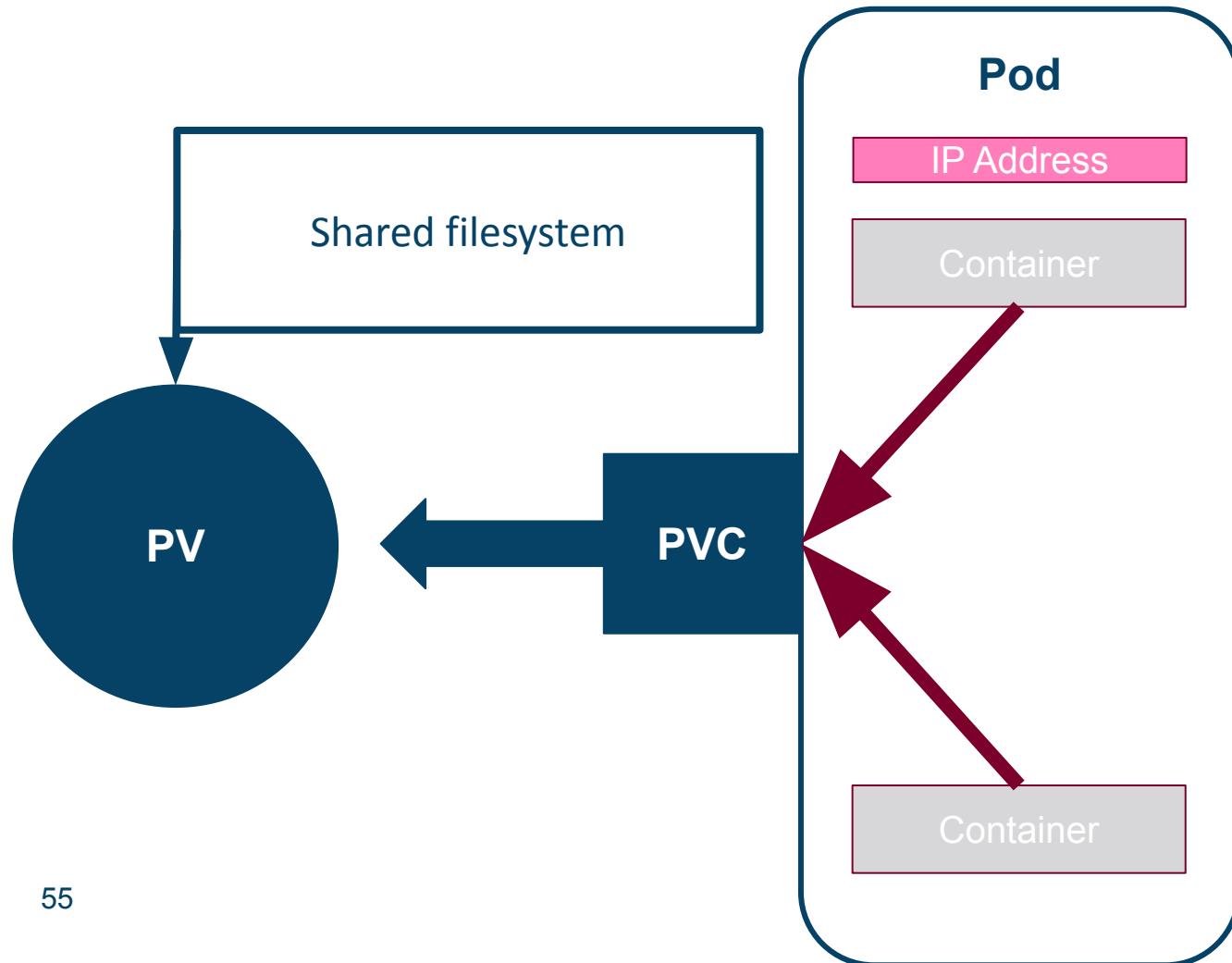
Exchange data between containers



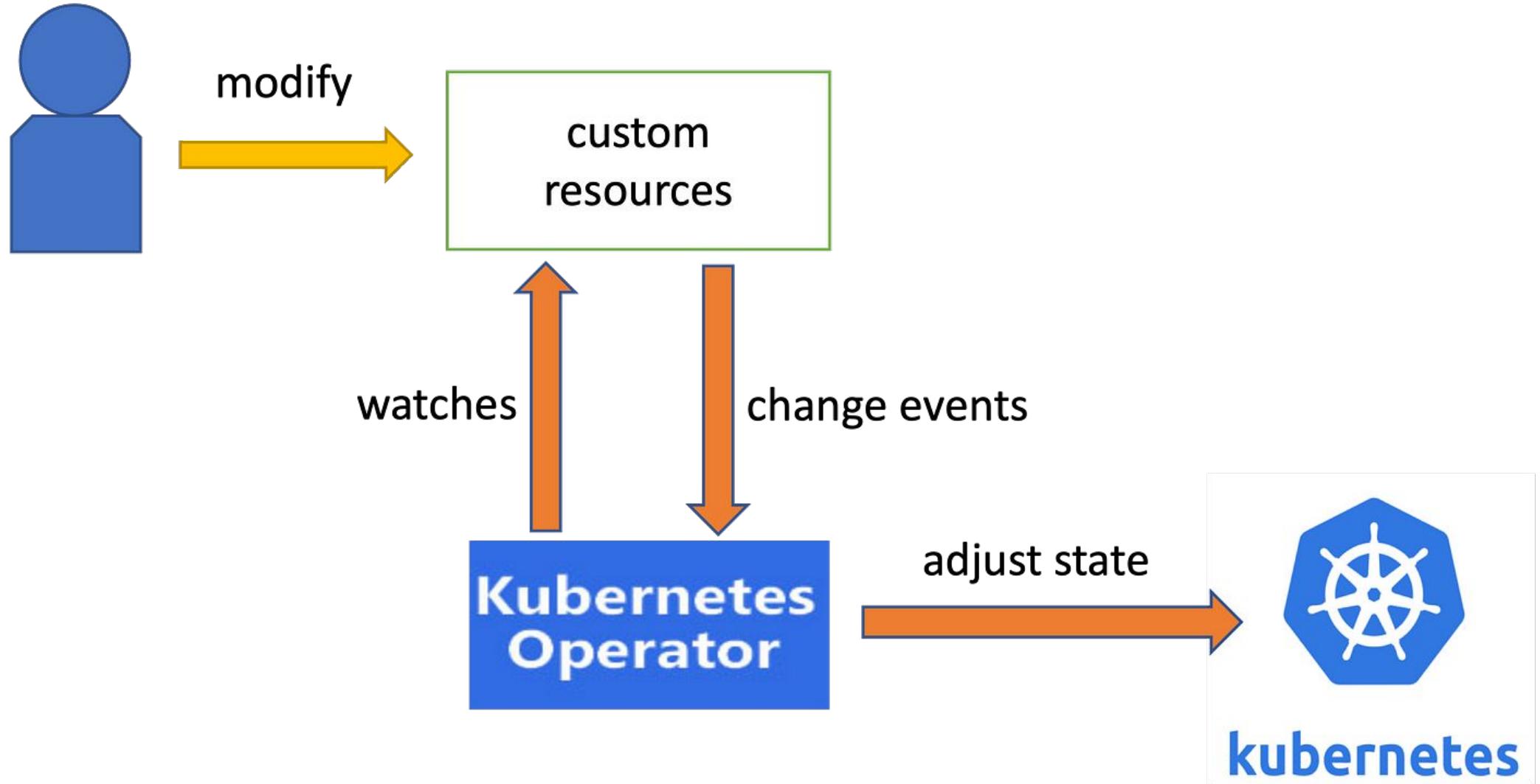
Exchange data between containers



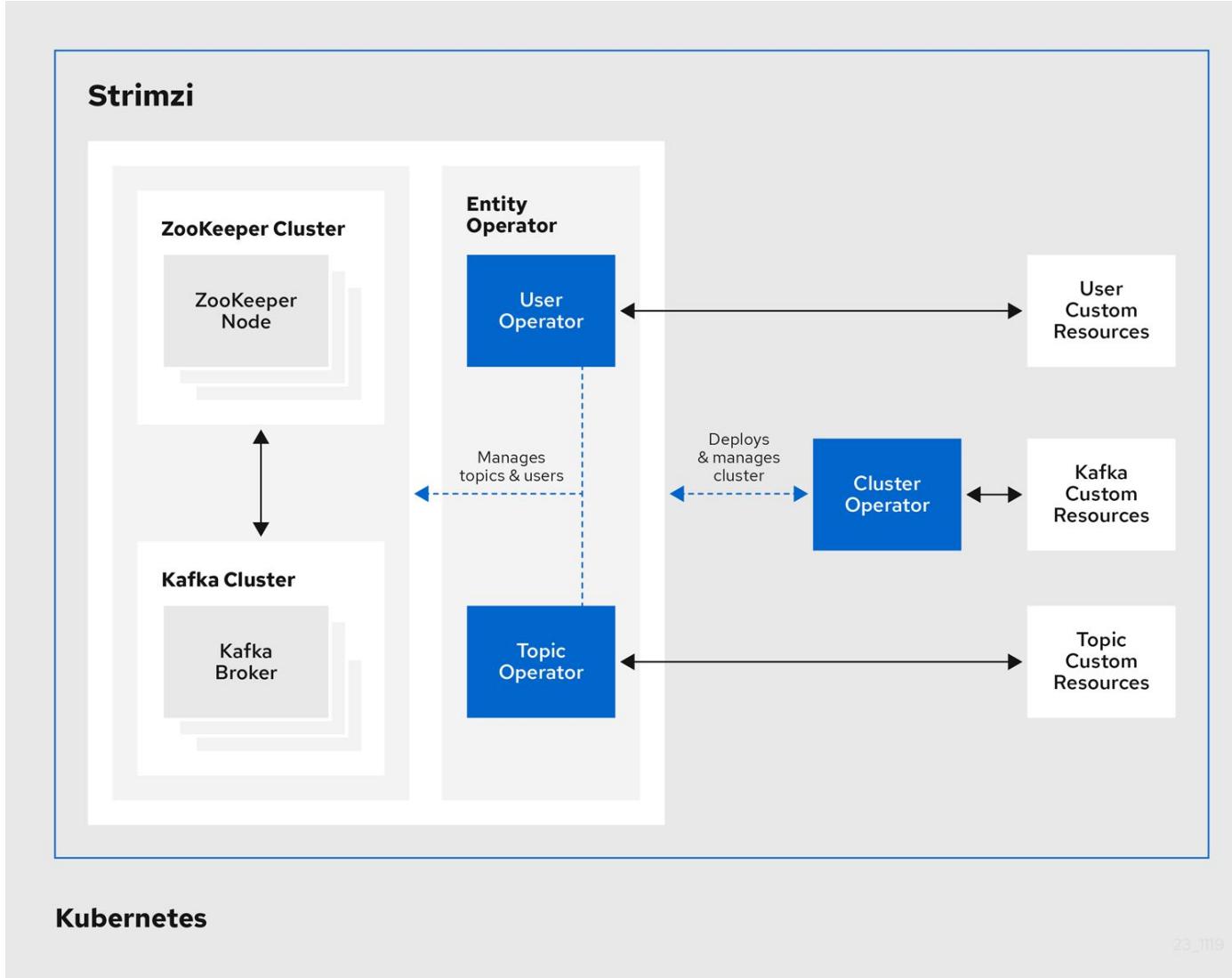
Exchange data between containers



Operators and Custom Resources



Example: Kafka (Strimzi)



Questions?



Lab 2

Lab exercises:

<https://github.com/iptch/kubernetes-hands-on>



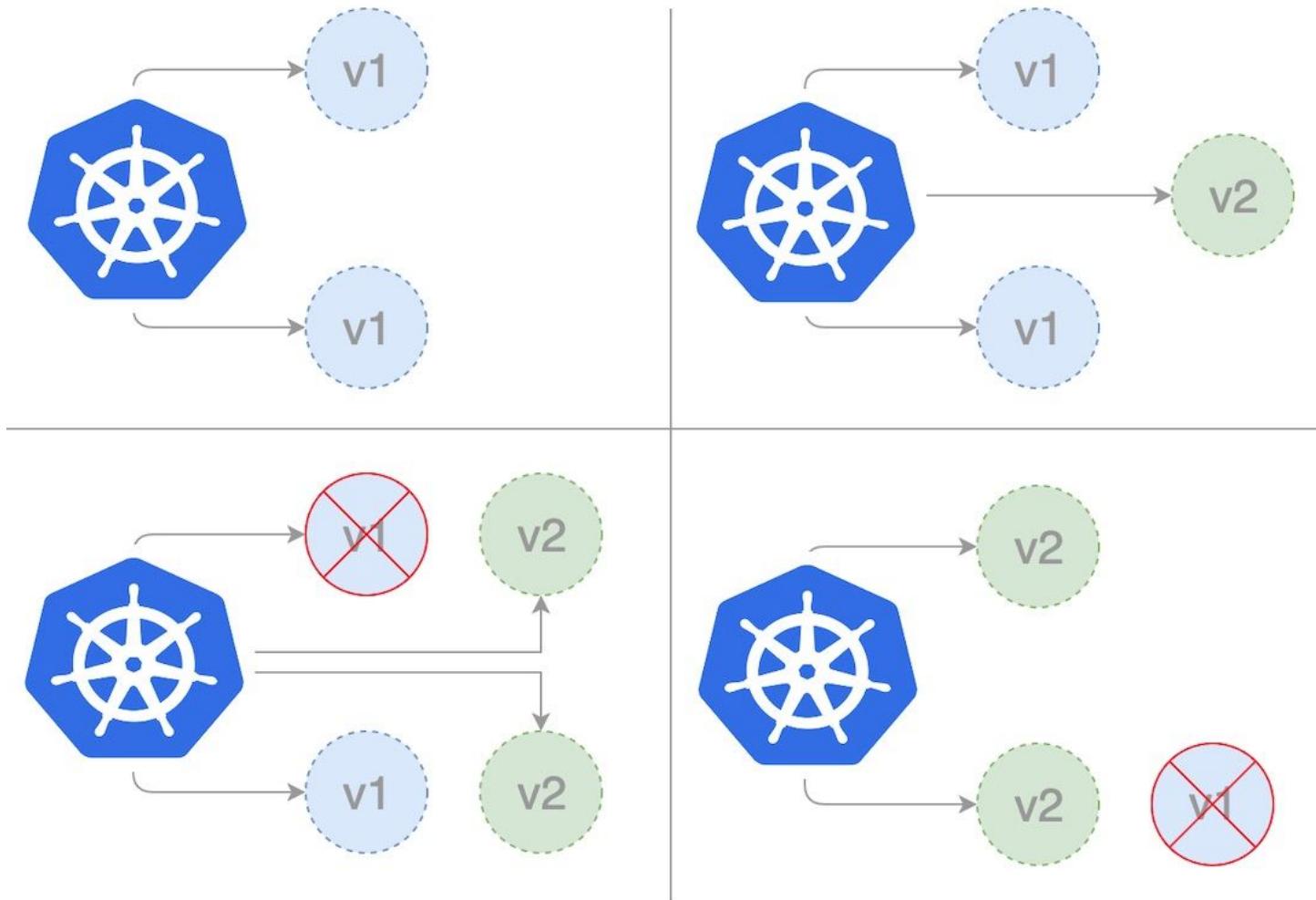
High Availability



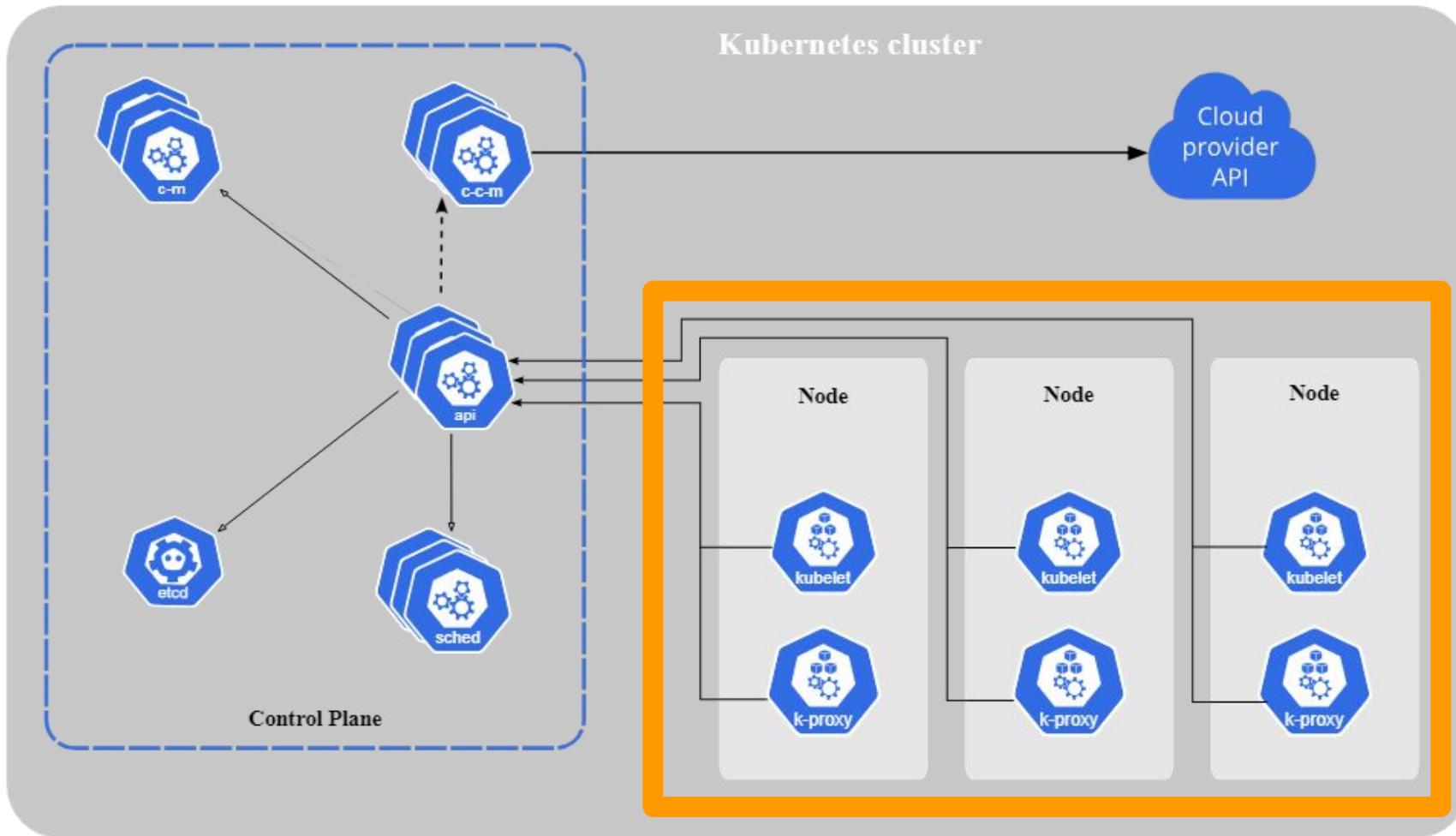
Traffic Routing - Probes

```
1 vim_tmp_1631507957+ ━━>
  1 livenessProbe:
  2   httpGet:
  3     path: /liveness
  4     port: 8080
  5     initialDelaySeconds: 1
  6     periodSeconds: 5
  7 readinessProbe:
  8   httpGet:
  9     path: /readiness
 10    port: 8080
 11    initialDelaySeconds: 1
 12    periodSeconds: 3
 13
```

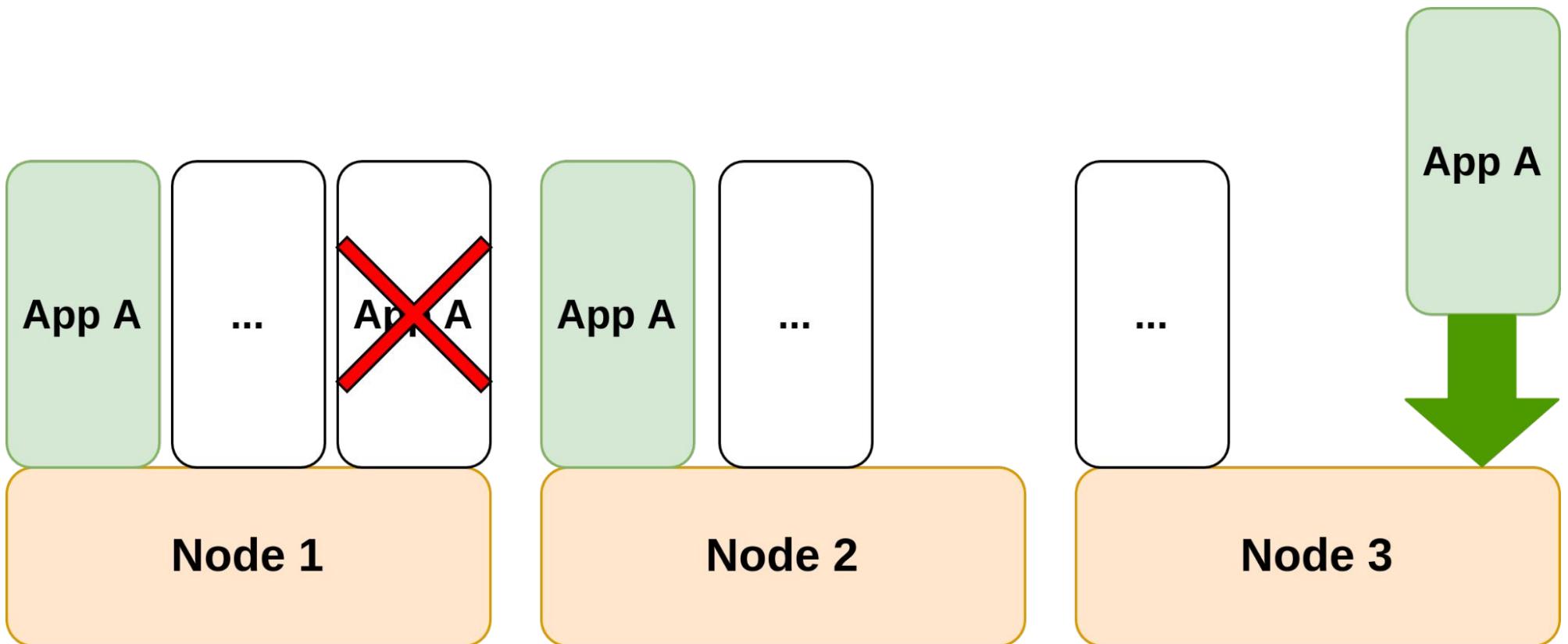
Lifecycles & Rolling Updates



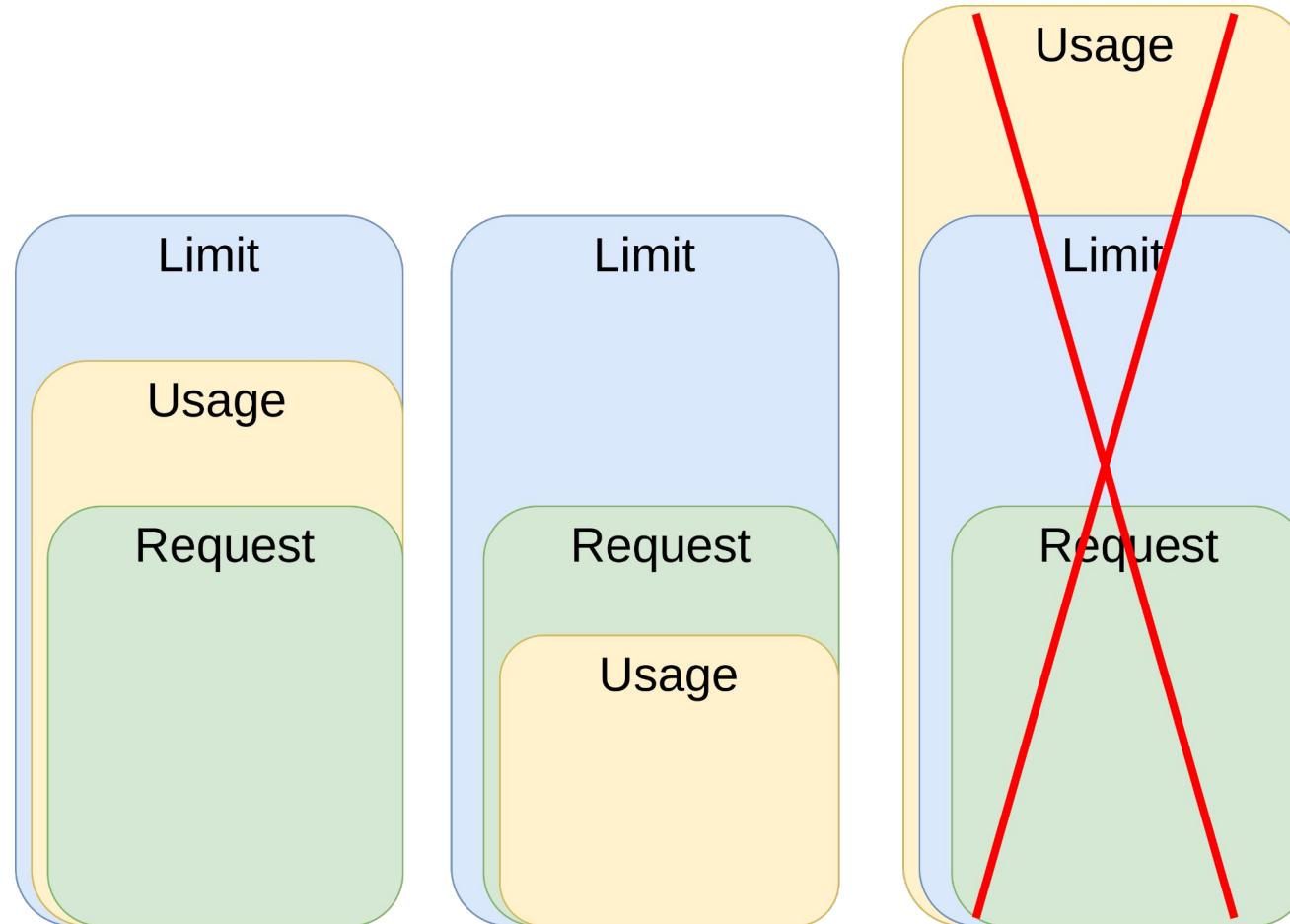
Scheduling - Anti-Affinities



Scheduling - Anti-Affinities



Pod/Container Requests and Limits



Pod/Container Requests and Limits

```
1  containers:
2    - name: app
3      image: images.my-company.example/app:v4
4      resources:
5        requests:
6          memory: "64Mi"
7          cpu: "250m"
8        limits:
9          memory: "128Mi"
10         cpu: "500m"
11
```

Questions?



Lab 3

Lab exercises:

<https://github.com/iptch/kubernetes-hands-on>



Deployments with HELM



The package manager for Kubernetes



CLOUD NATIVE
COMPUTING FOUNDATION



Artifact **HUB**

Manage
Complexity

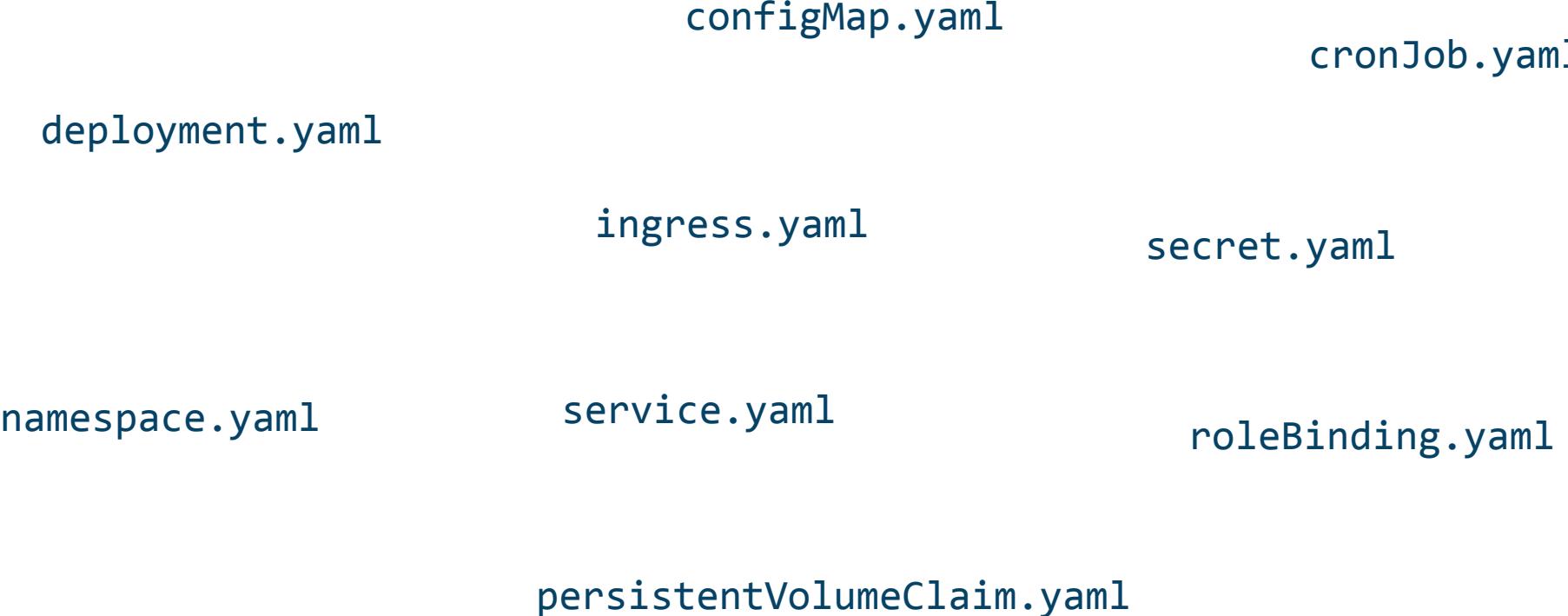


Easy Updates

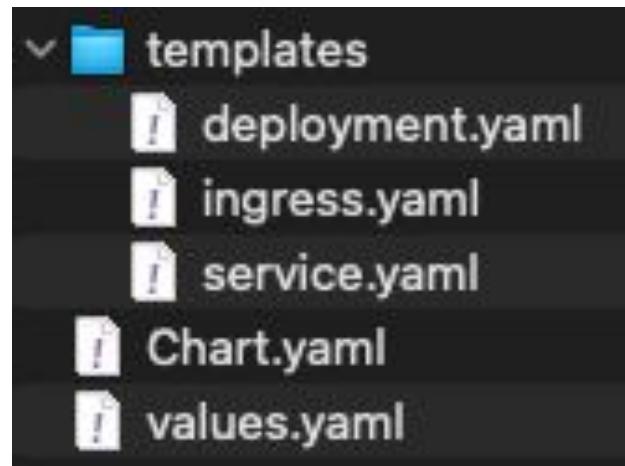
Simple Sharing

Rollbacks

How to handle complex deployments?



Organises Deployments in Charts



```
apiVersion: v1
kind: Service
metadata:
  name: {{ .Values.service.name }}
spec:
  type: {{ .Values.service.type }}
  ports:
    - port: {{ .Values.service.port }}
```

```
apiVersion: v2
name: testchart
description: A Helm chart for Kubernetes
```

```
service:
  name: mein-service
  type: ClusterIP
  port: 80
```

Go Templates

```
apiVersion: v1
kind: Service
metadata:
  name: {{ include "hello-world.fullname" . }}
  labels:
    {{- include "hello-world.labels" . | nindent 4 }}
spec:
  type: {{ .Values.service.type }}
  ports:
    - port: {{ .Values.service.port }}
      targetPort: http
      protocol: TCP
      name: http
  selector:
    {{- include "hello-world.selectorLabels" . | nindent 4 }}
```

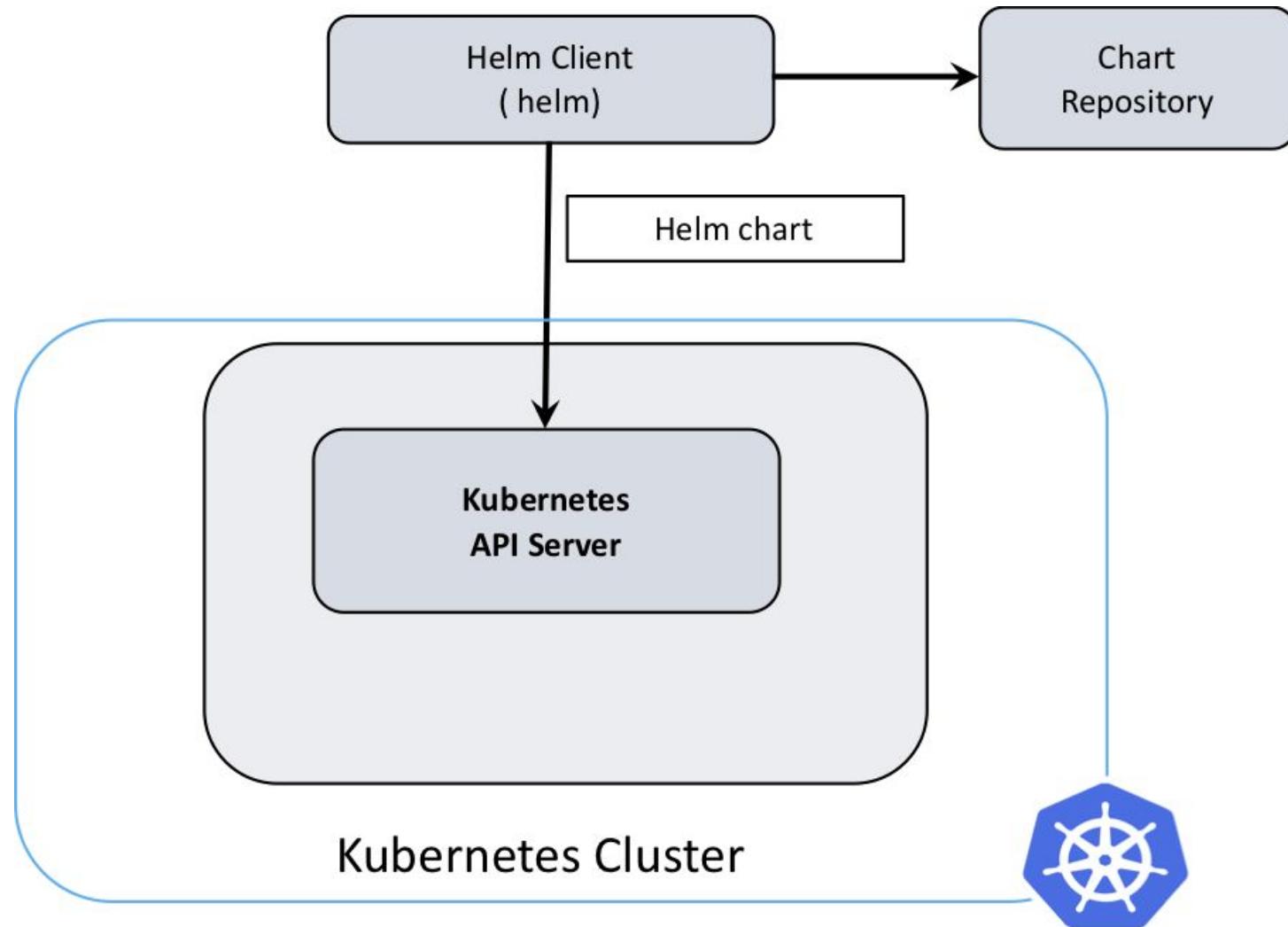
- parameterize Kubernetes resource definitions
- conditionals
- looping over lists
- manipulating data
- highly customizable deployments

Rendering Go Templates

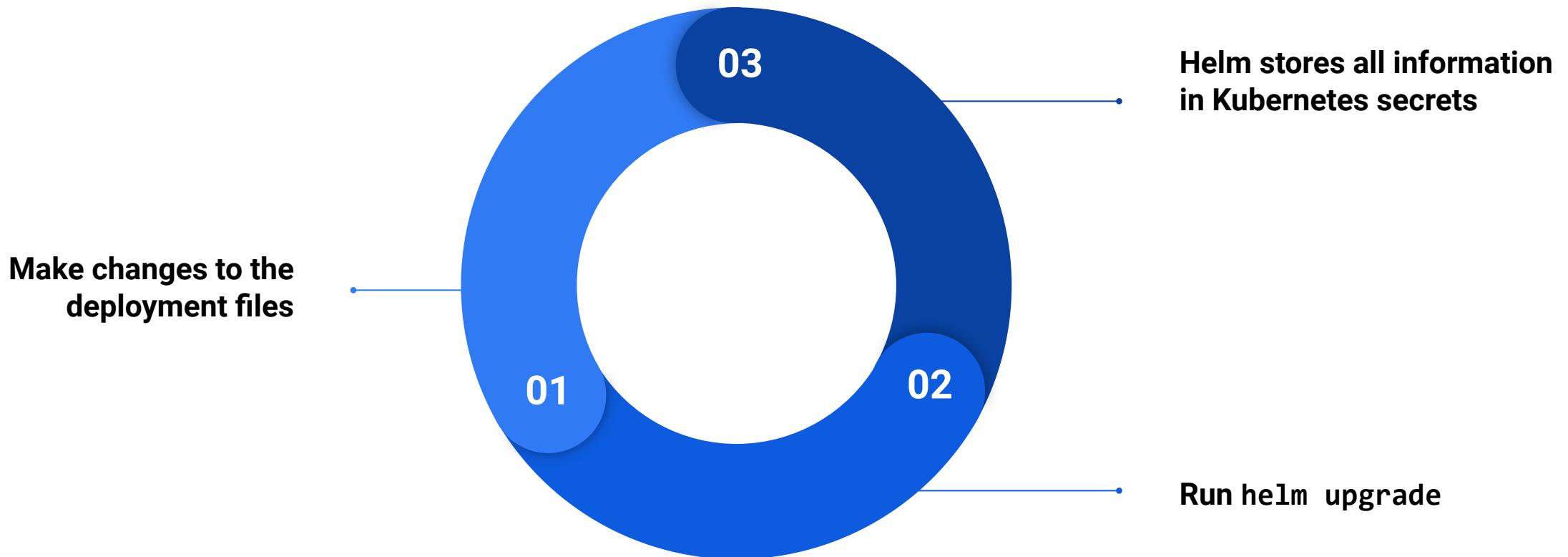
```
helm install --dry-run mydeployment .
```

```
apiVersion: v1
kind: Service
metadata:
  name: mydep-handson
  labels:
    helm.sh/chart: handson-0.1.0
    app.kubernetes.io/name: handson
    app.kubernetes.io/instance: mydep
    app.kubernetes.io/version: "1.16.0"
    app.kubernetes.io/managed-by: Helm
spec:
  type: ClusterIP
  ports:
    - port: 80
      targetPort: http
      protocol: TCP
      name: http
  selector:
    app.kubernetes.io/name: handson
    app.kubernetes.io/instance: mydep
```

Helm Architecture



Update/Rollback



Update/Rollback

```
> kubectl get secrets  
NAME                                     TYPE  
sh.helm.release.v1.nginx-csb.v1          helm.sh/release.v1  
sh.helm.release.v1.nginx-csb.v2          helm.sh/release.v1  
sh.helm.release.v1.nginx-csb.v3          helm.sh/release.v1  
sh.helm.release.v1.nginx-csb.v4          helm.sh/release.v1
```

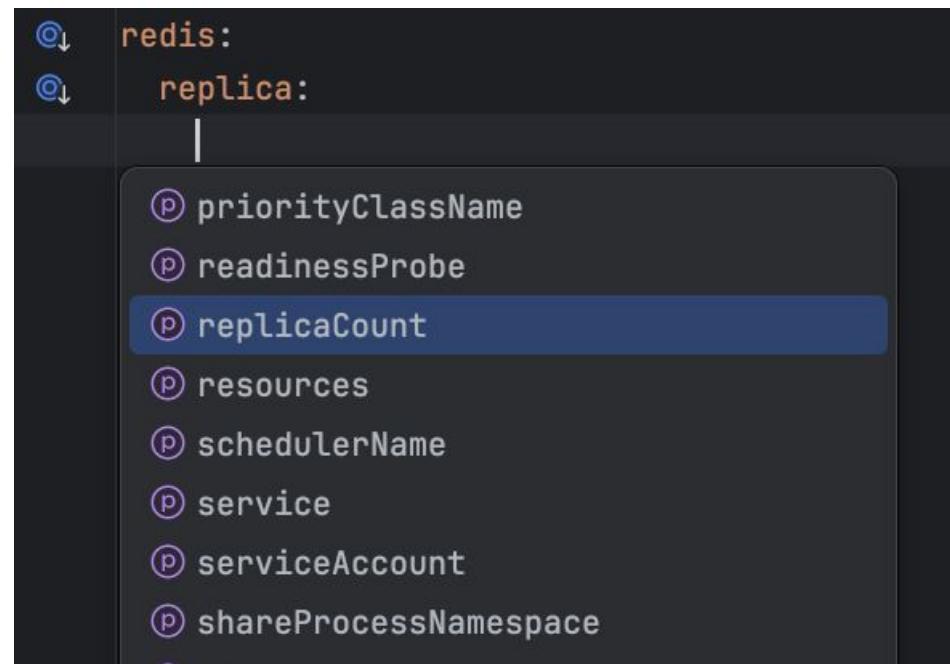
Using dependencies in HELM

Chart.yaml

```
dependencies:
  - name: redis
    version: 17.3.11
    repository: https://marketplace.azurecr.io/helm/v1/repo
```

Using dependencies in HELM

values.yaml



The image shows a code editor interface with a dark theme. A dropdown menu is open over a portion of a YAML file. The file contains the following code:

```
redis:
  replica:
```

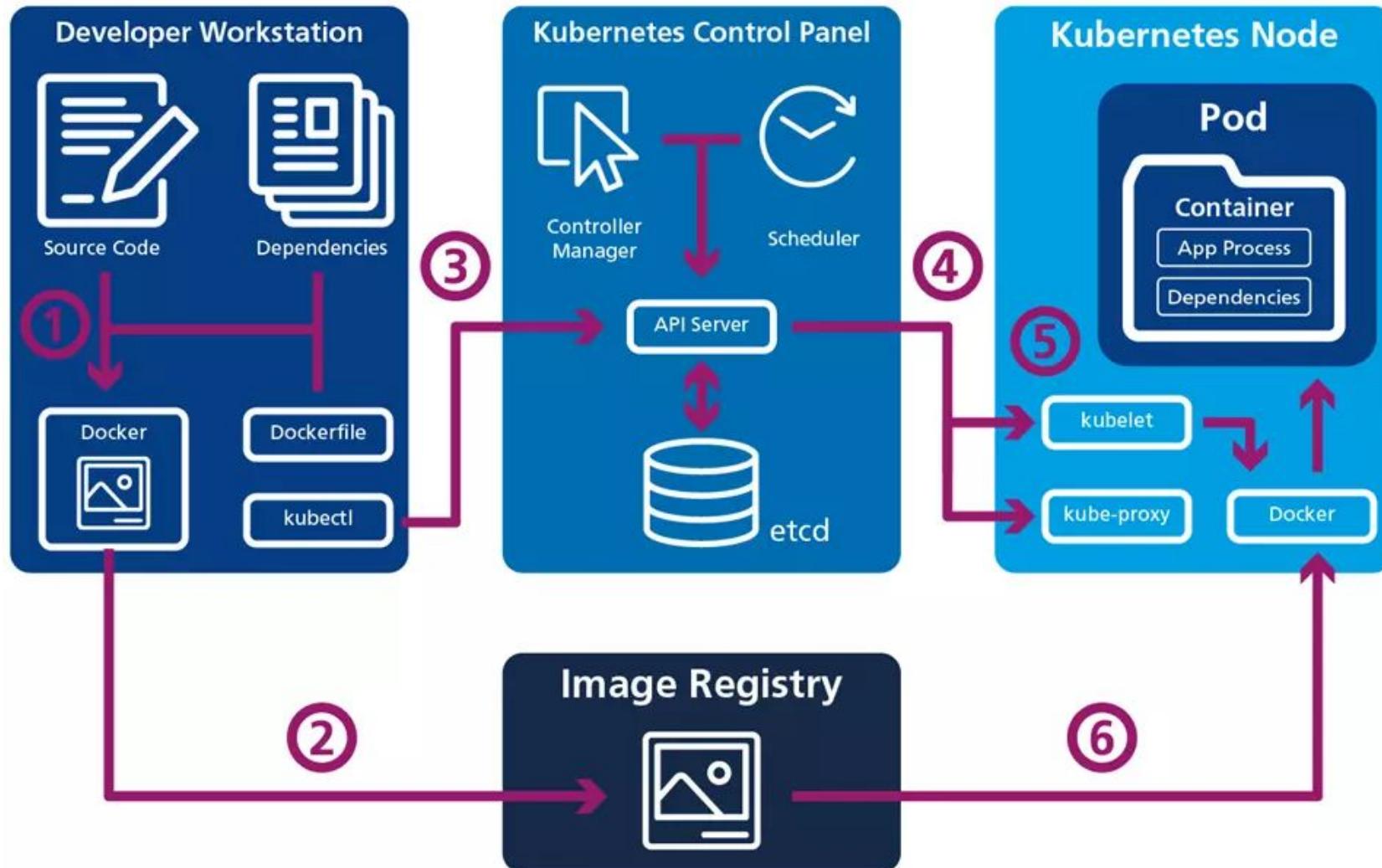
The dropdown menu lists several properties, each preceded by a purple circular icon containing a lowercase 'p':

- priorityClassName
- readinessProbe
- replicaCount** (This item is highlighted with a blue background)
- resources
- schedulerName
- service
- serviceAccount
- shareProcessNamespace

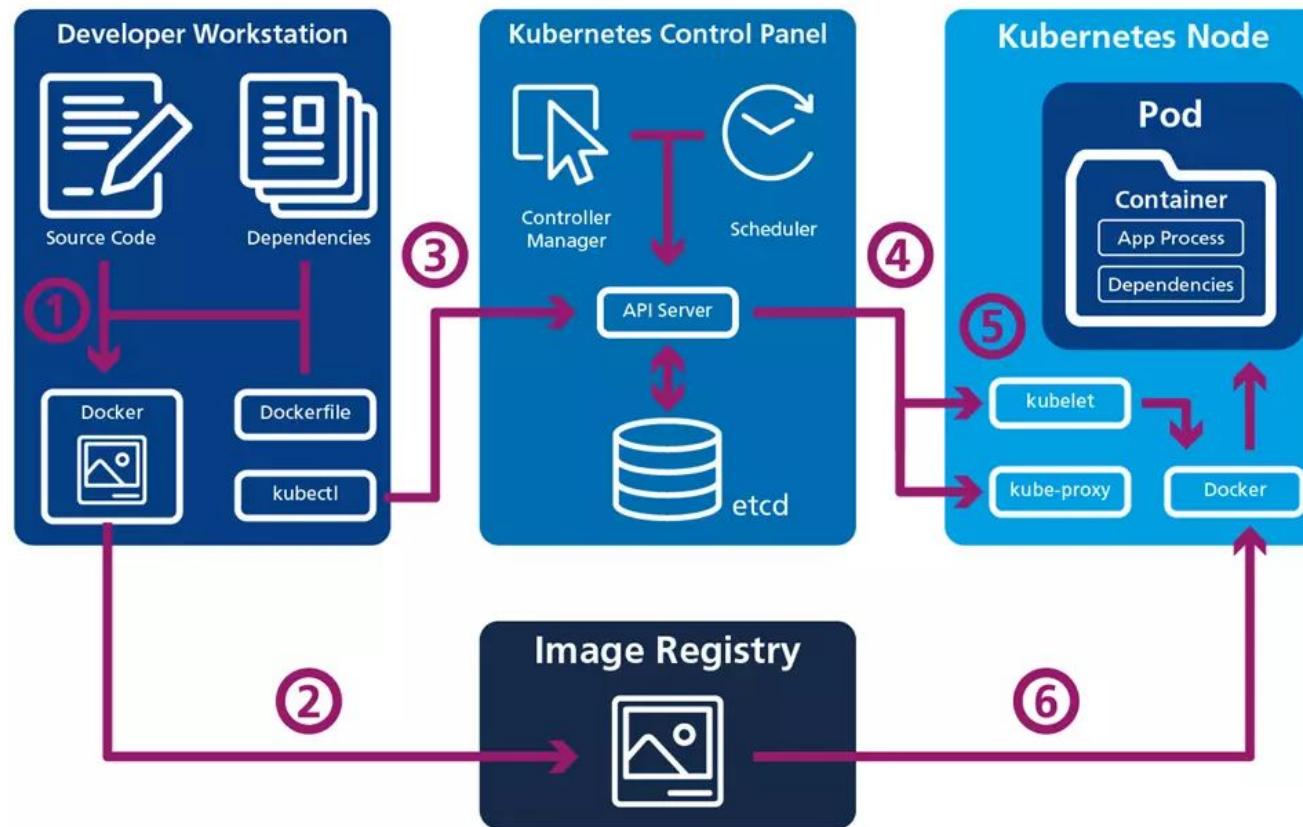
Summary and Open Questions



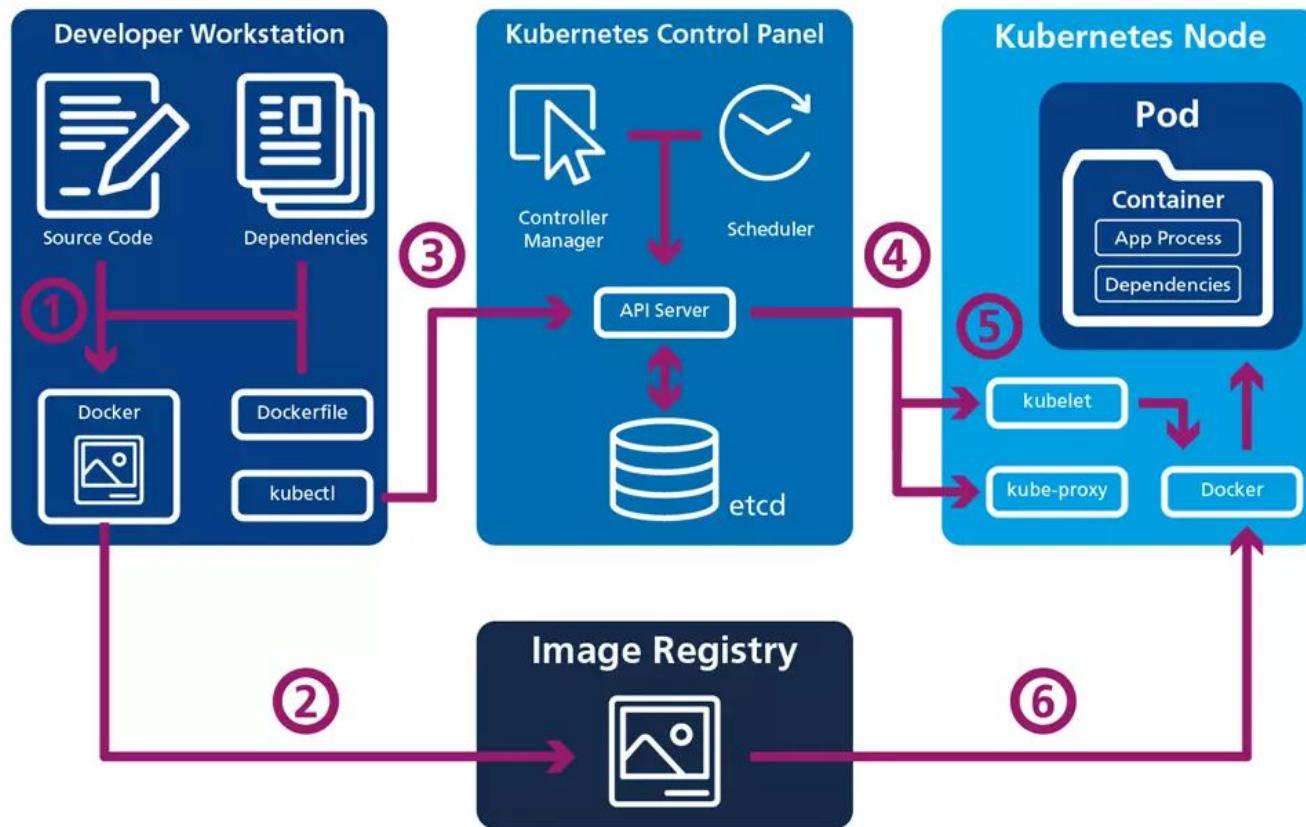
Summary



Summary



Summary

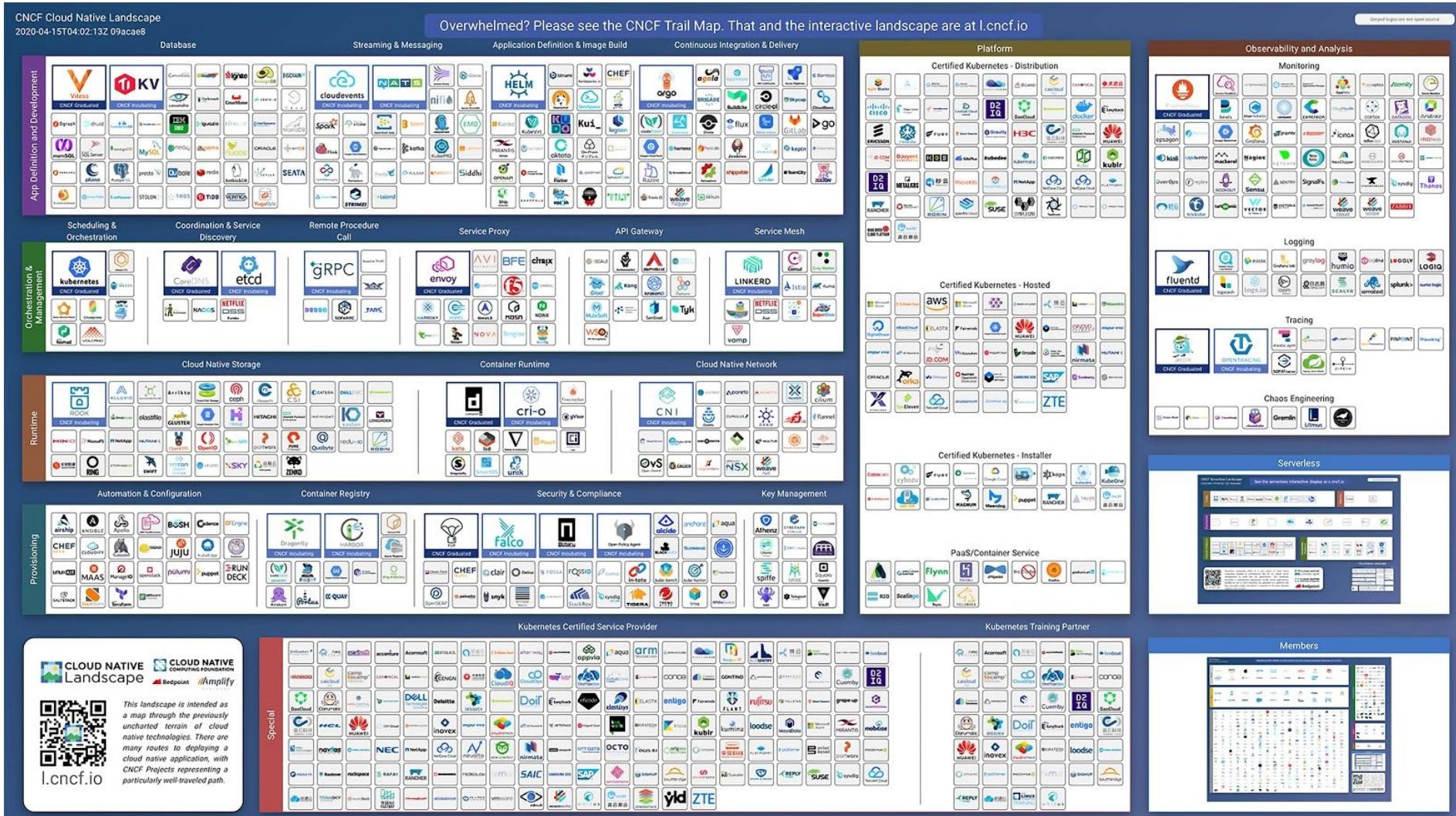


Prometheus

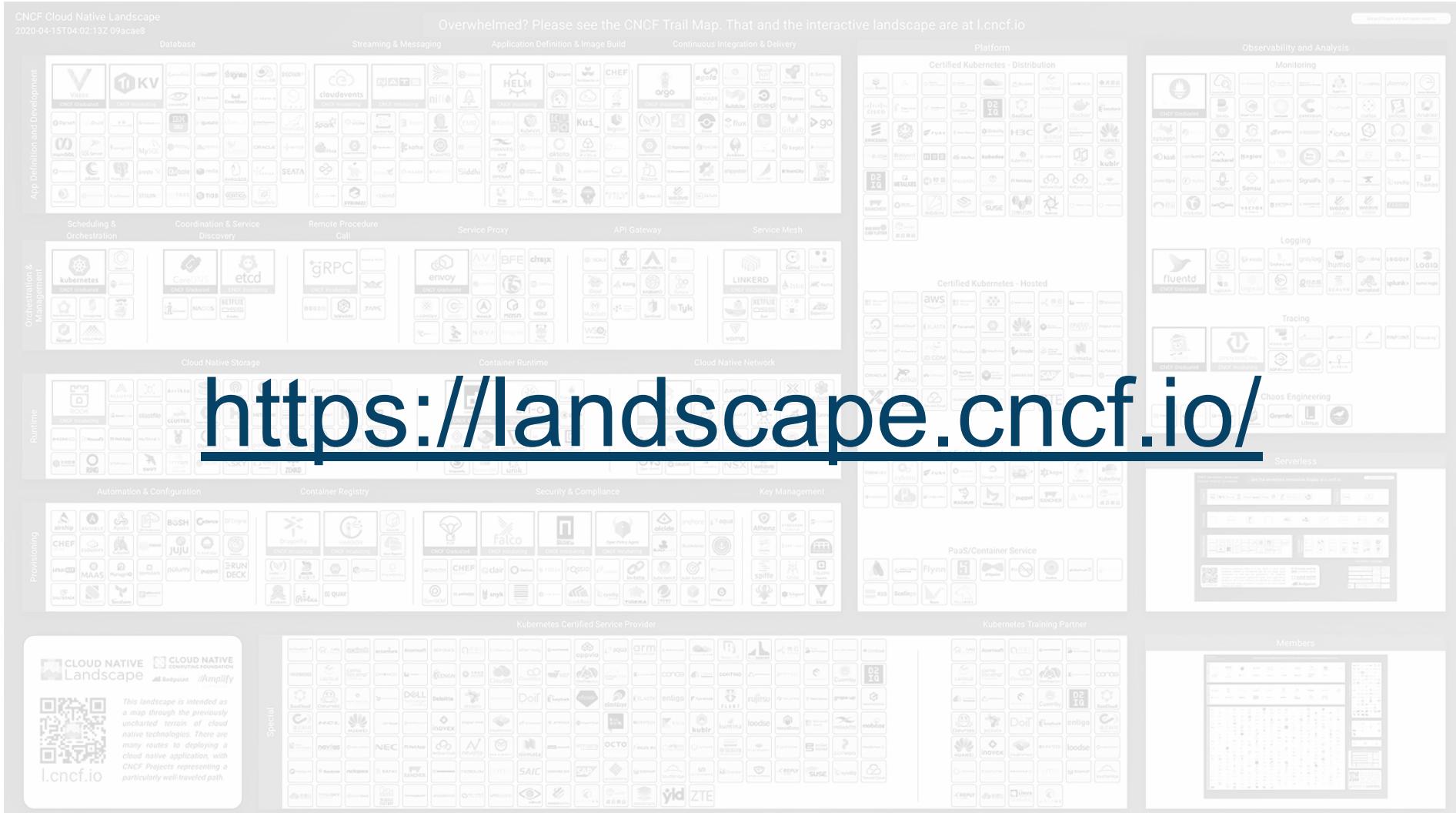


Grafana

Kubernetes Ecosystem

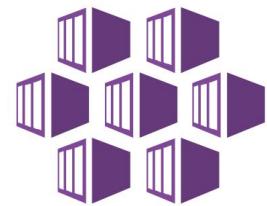


Kubernetes Ecosystem



Kubernetes Ecosystem

Cloud:



Azure Kubernetes Service (AKS)

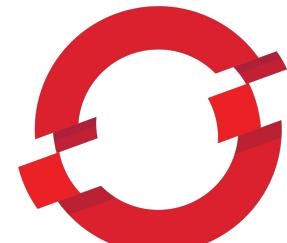


Google
Kubernetes Engine



Amazon
EKS

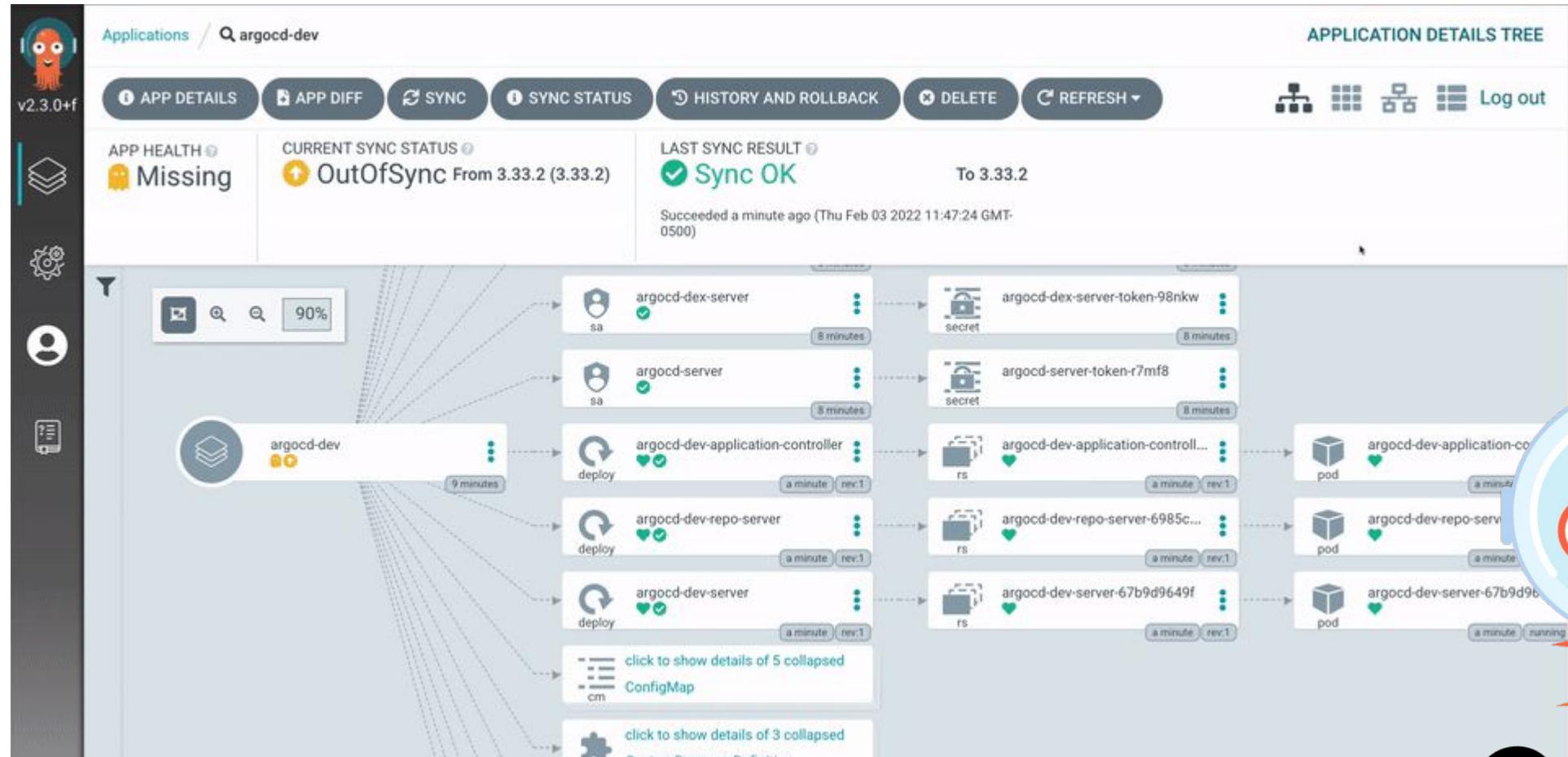
On-Prem:



OPENSHIFT



Kubernetes Ecosystem



Questions?



Join us at ipt.ch!



Feedback!



<https://scrumblr.io/board/2aa31c38-c2df-44da-86c4-c104cc4684f1>



I NEED TO KNOW WHY MOVING
OUR APP TO THE CLOUD DIDN'T
AUTOMATICALLY SOLVE ALL OUR
PROBLEMS.



YOU WOULDN'T
LET ME RE-
ARCHITECT THE
APP TO BE
CLOUD-NATIVE. JUST PUT IT
IN
CONTAINERS.

Dilbert.com
© 2017 Scott Adams, Inc. Dist. by Andrews McMeel

YOU CAN'T
SOLVE A
PROBLEM JUST
BY SAYING
TECHY THINGS. KUBERNETES.



11-06-17 © 2017 Scott Adams, Inc. Dist. by Andrews McMeel