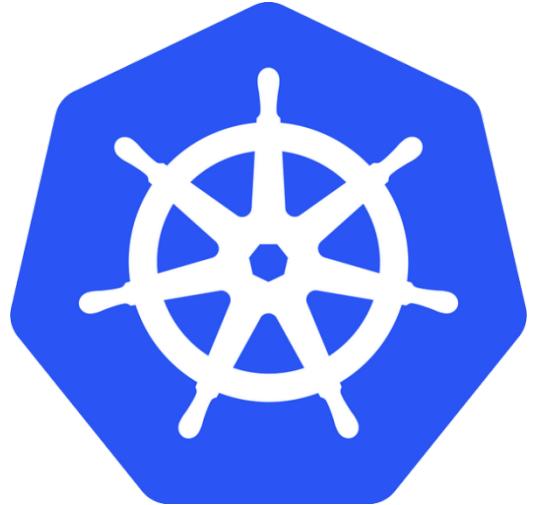


Kubernetes
K8s

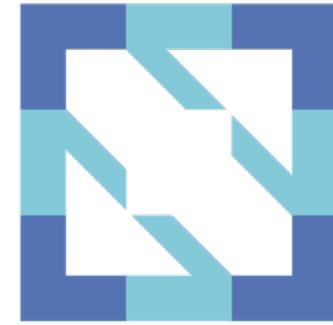
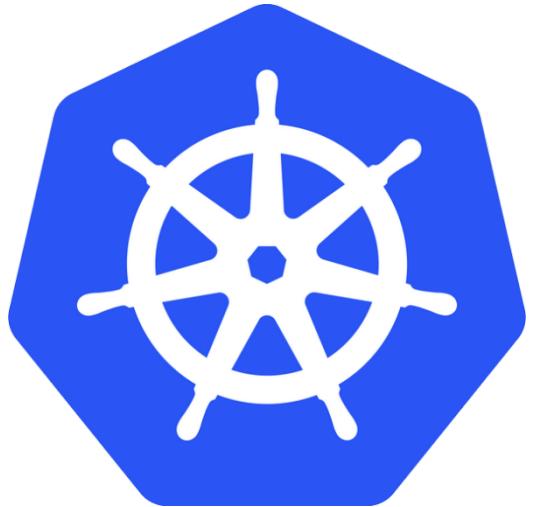




Kubernetes

Greek for “Helmsman” < the person who steers a ship





**CLOUD NATIVE
COMPUTING FOUNDATION**





github.com/kubernetes/kubernetes

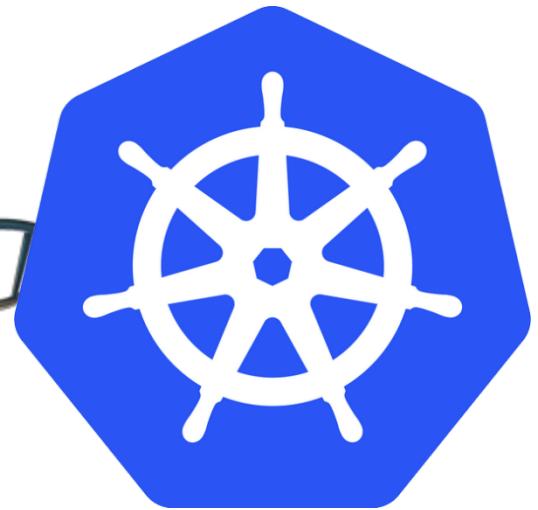




Borg
(Proprietary)

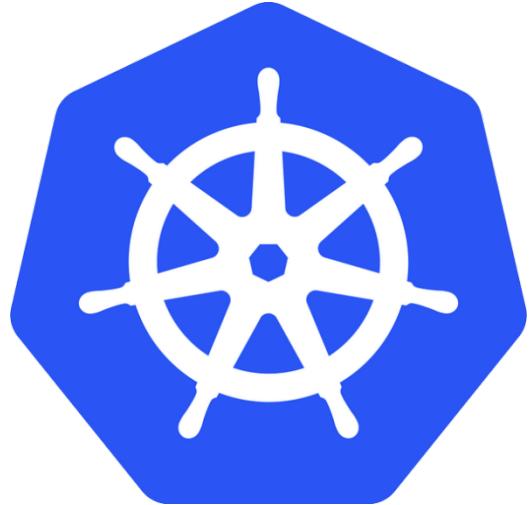


Omega
(Proprietary)



Kubernetes
(open-source)

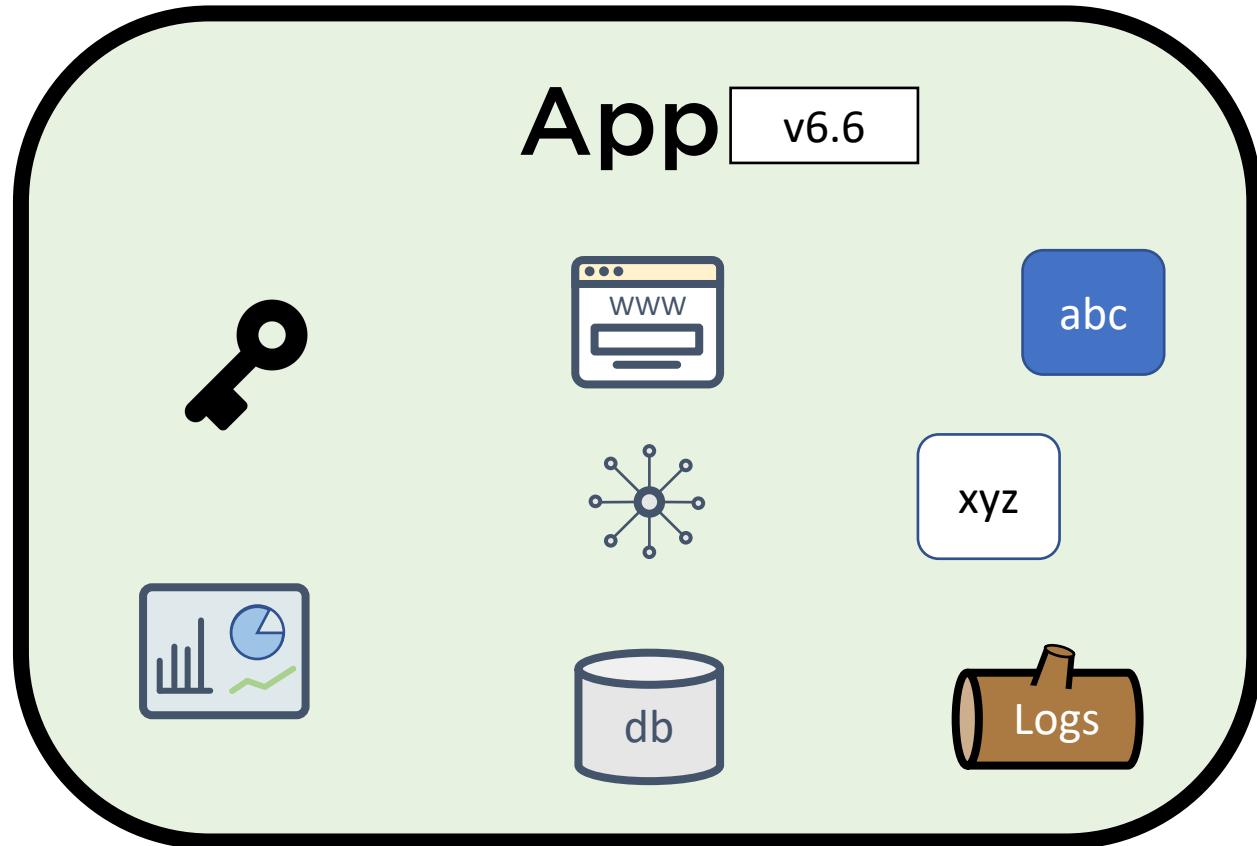




Version 1 in July 2015

Increasingly stable and mature

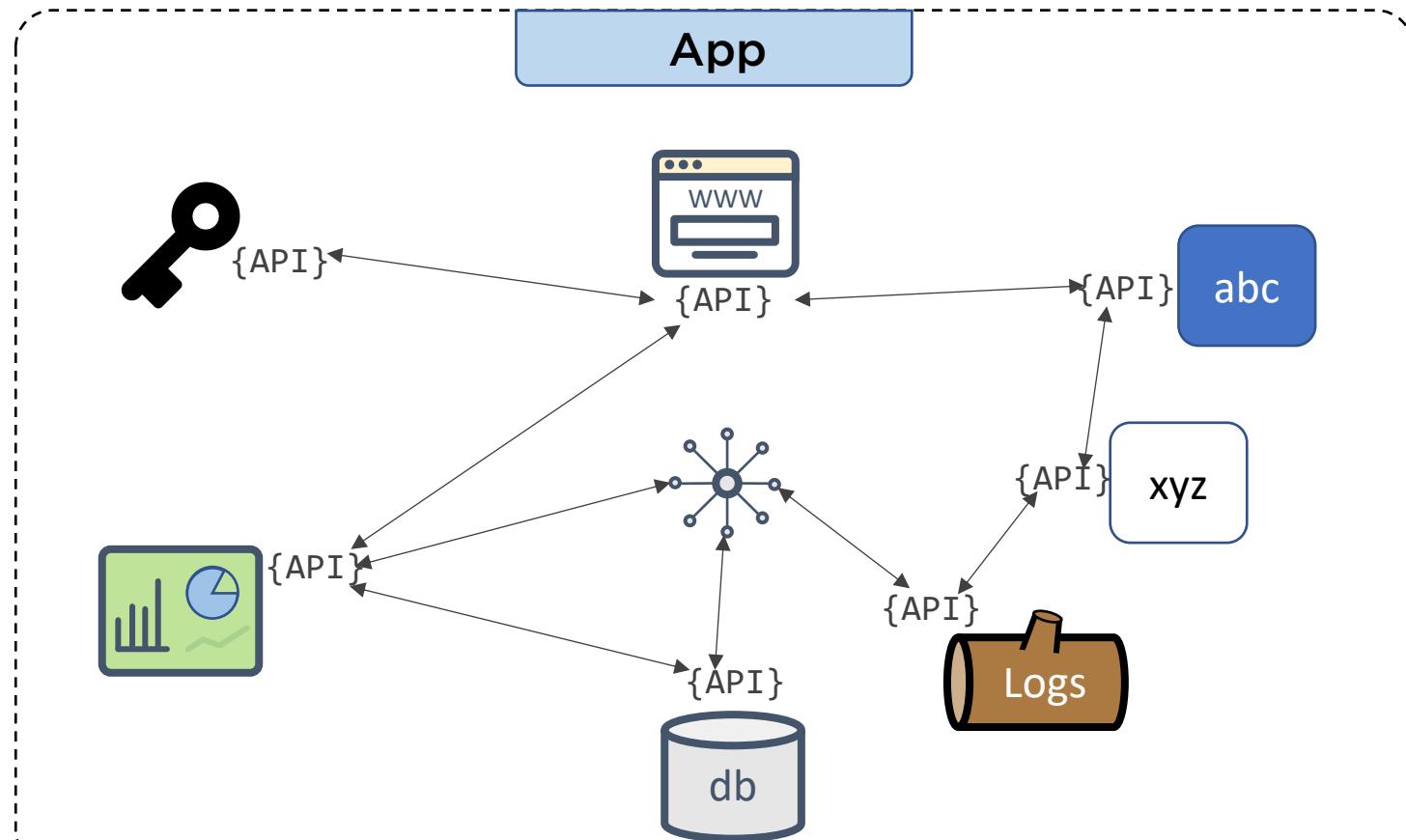


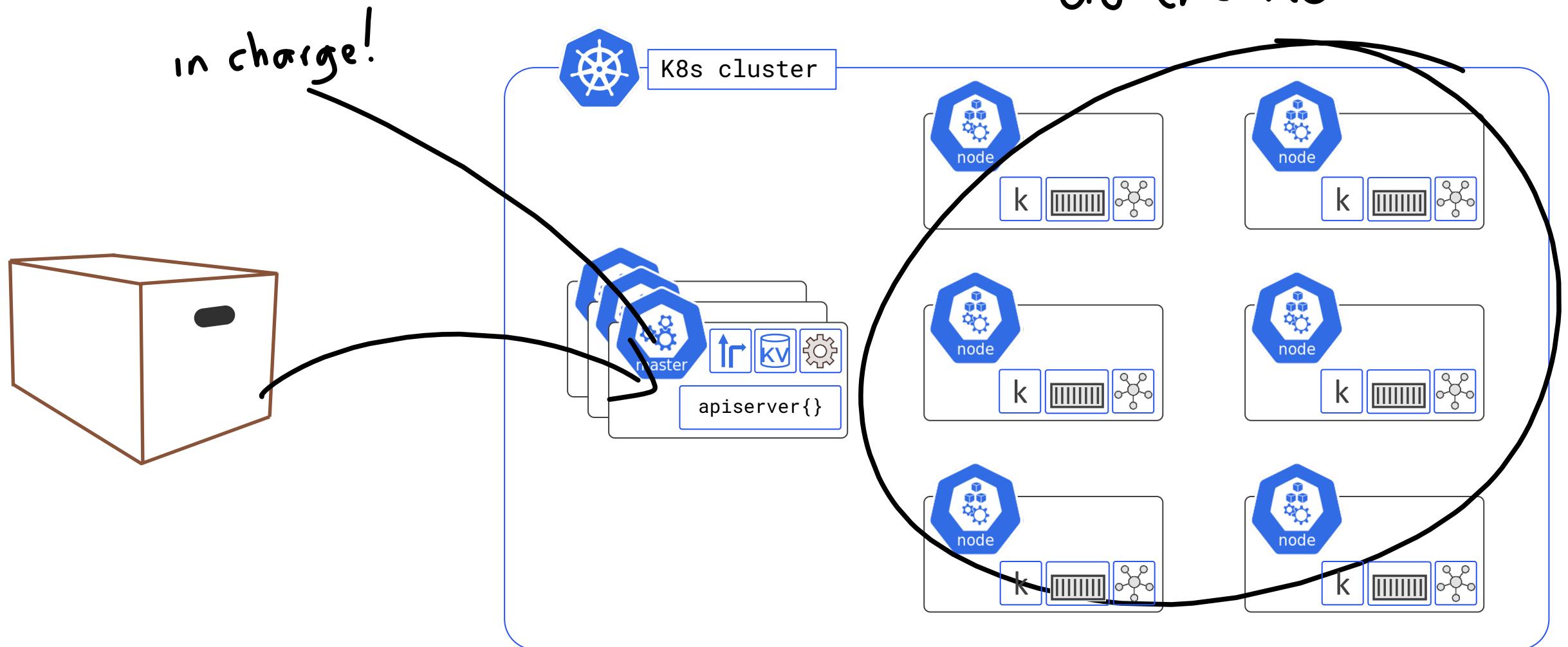


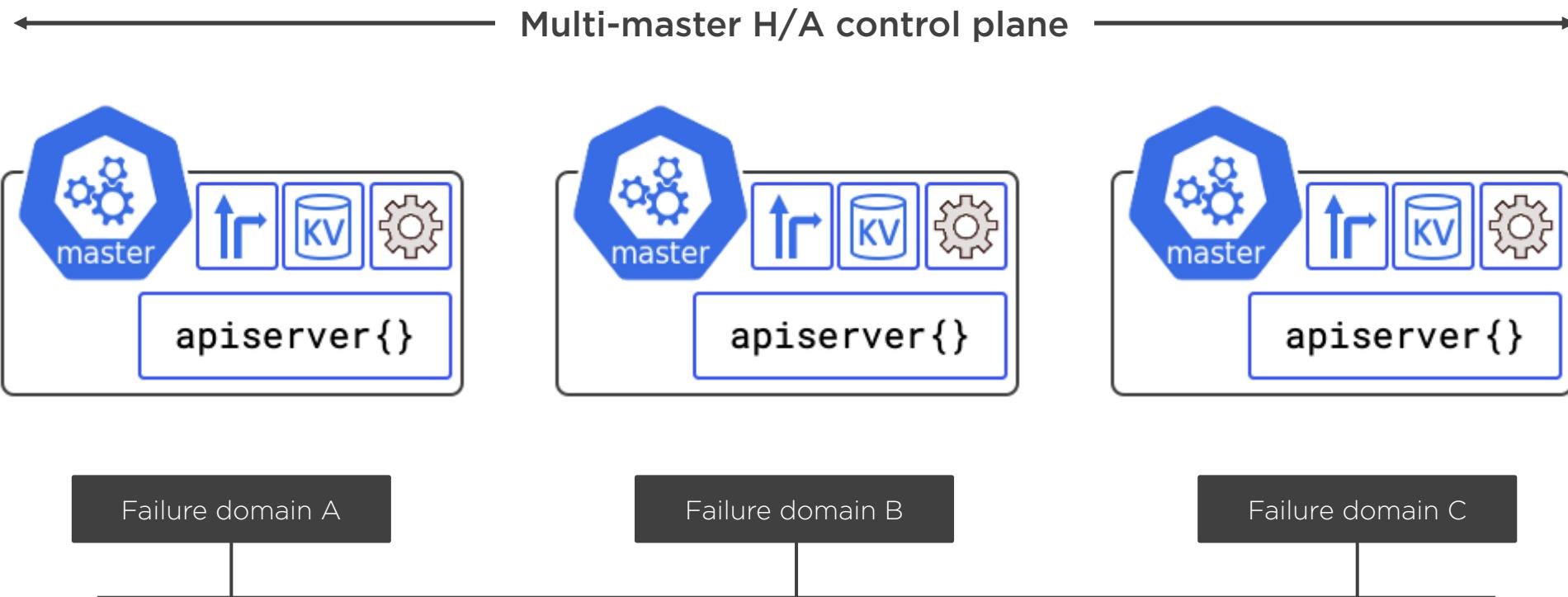
monolith /'mɒn(ə)lɪθ/

(noun) Insulting word for old/legacy application

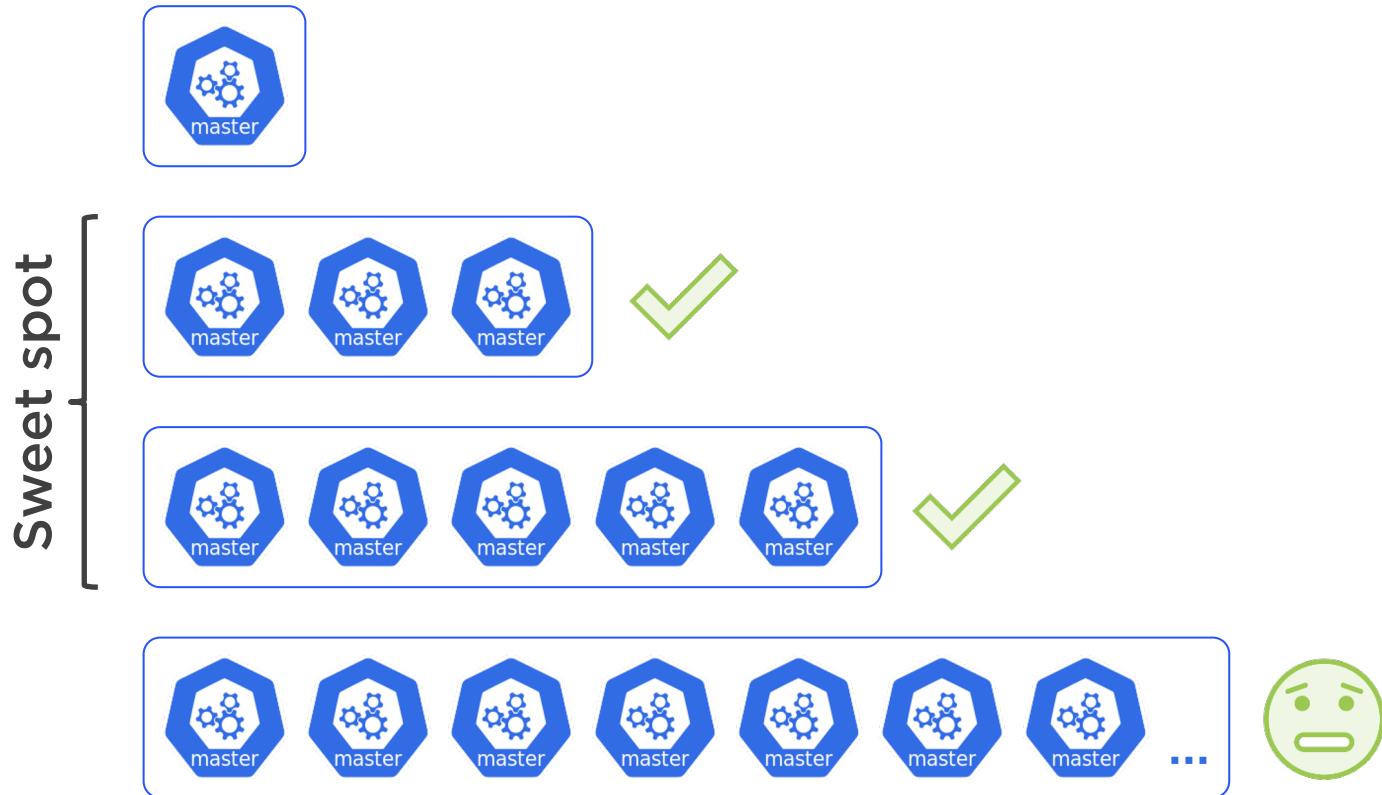






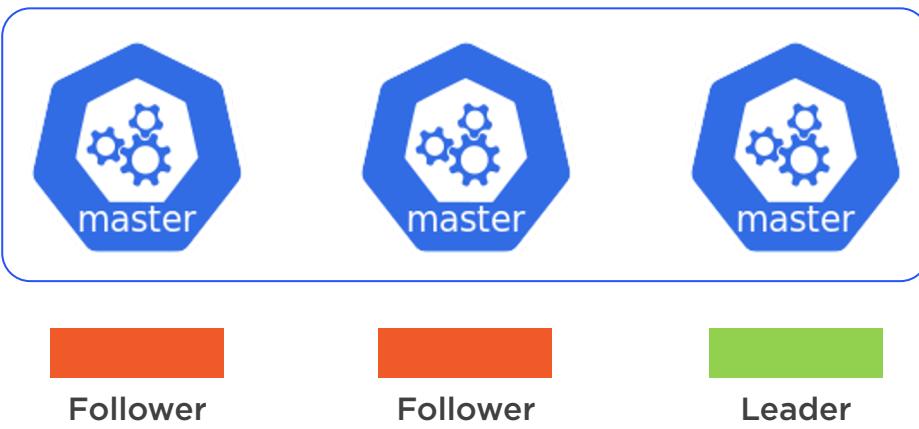


H/A Design

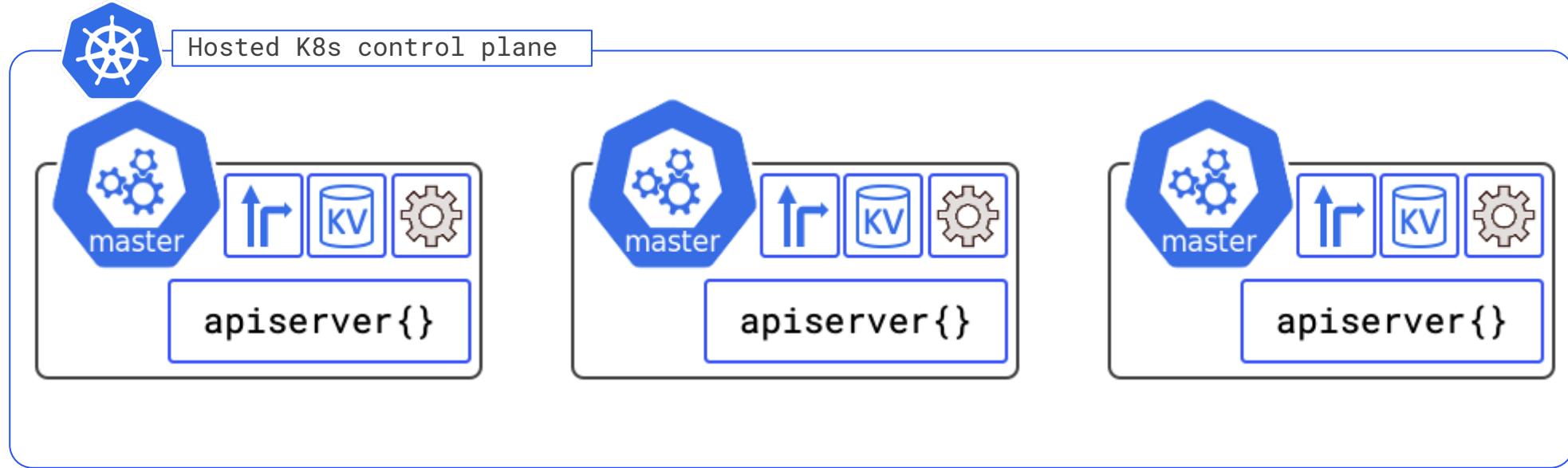


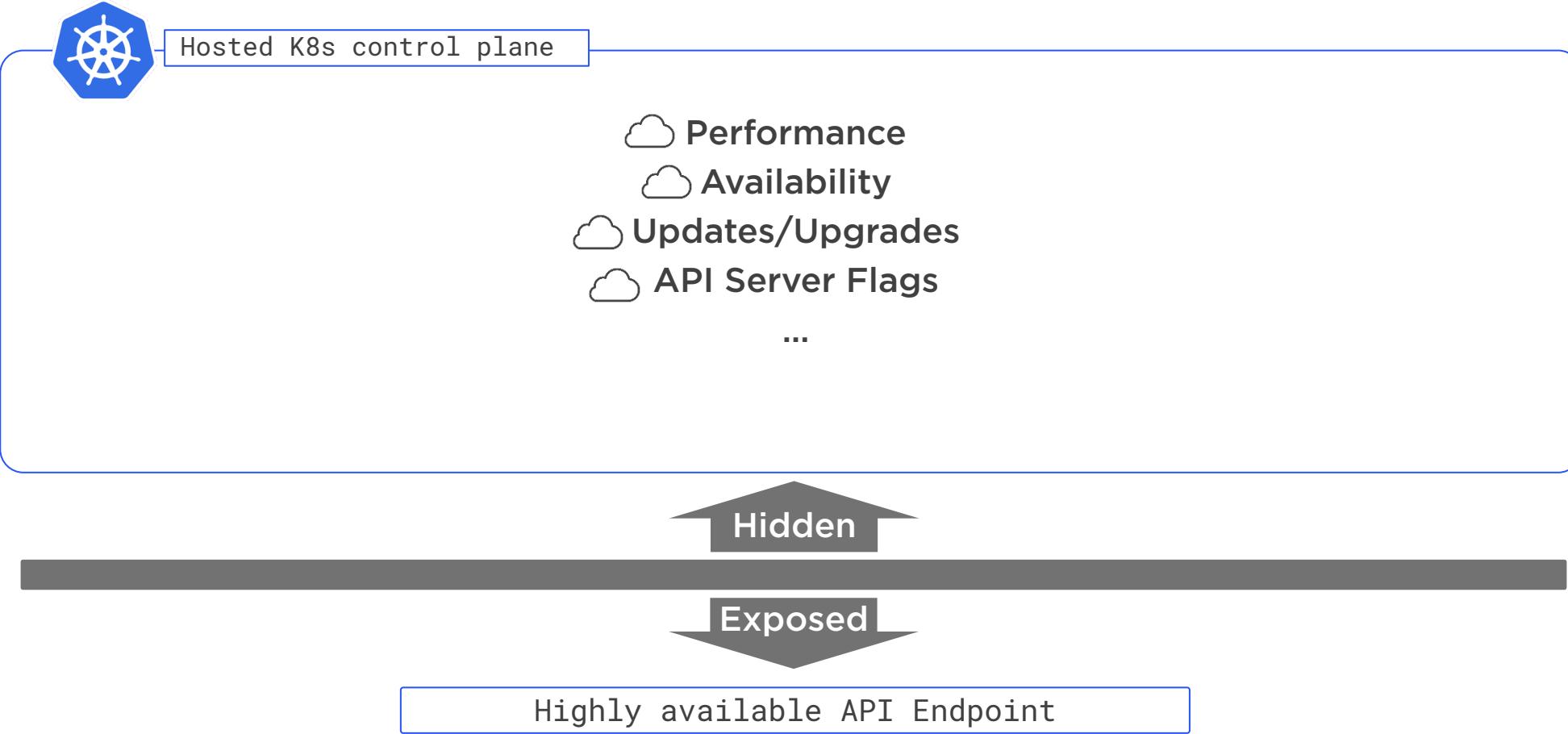






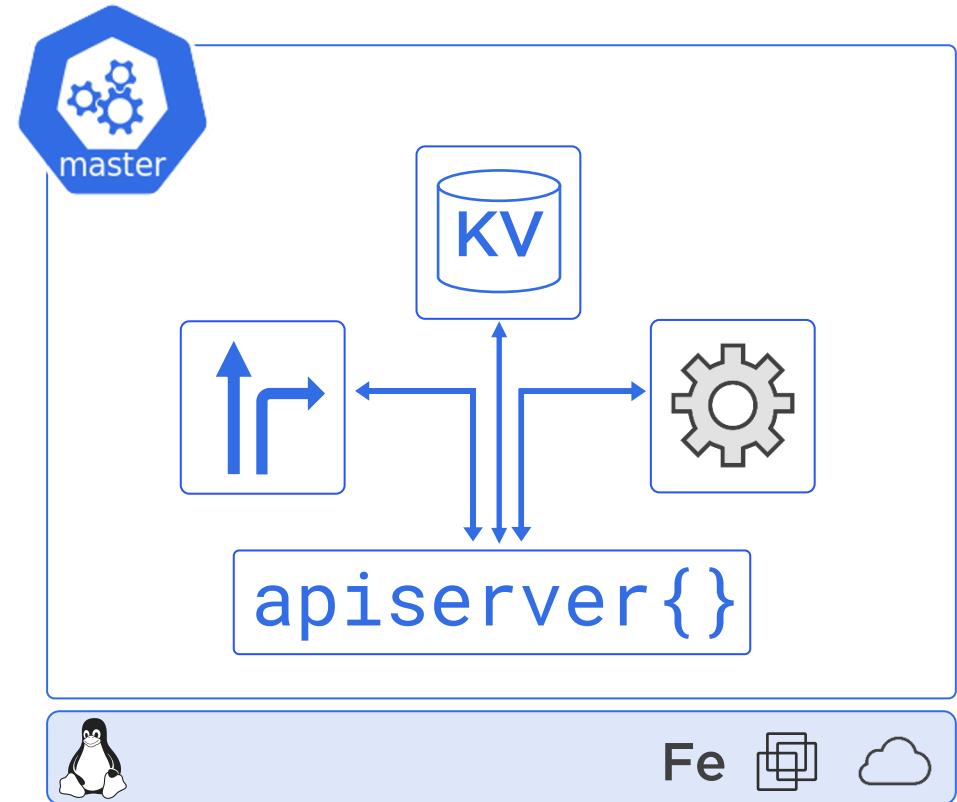






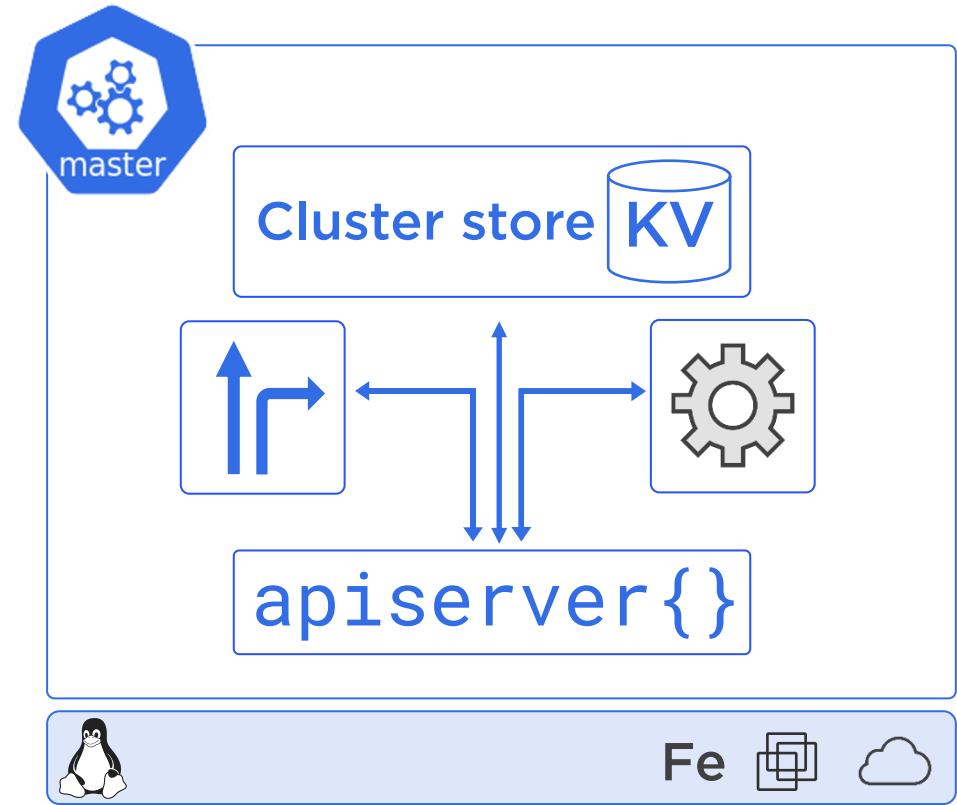
kube-apiserver

- Front-end to the control plane
- Exposes the API (REST)
- Consumes JSON/YAML



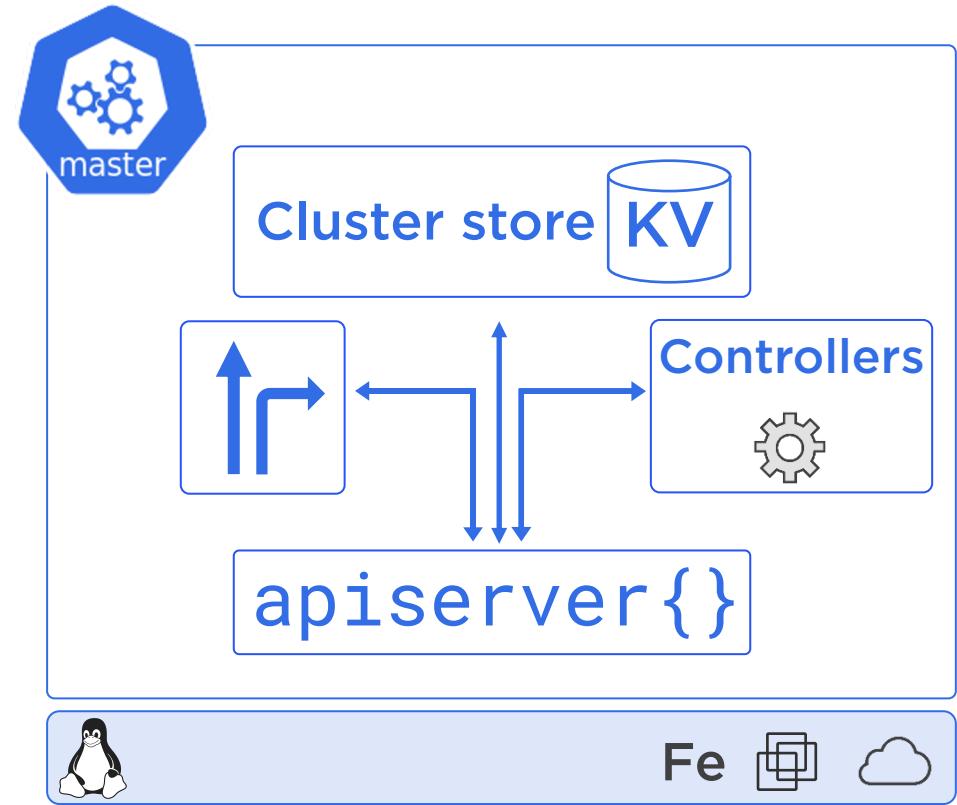
Cluster Store

- Persists cluster state and config
- Based on **etcd**
- Performance is critical
- Have recovery plans in place



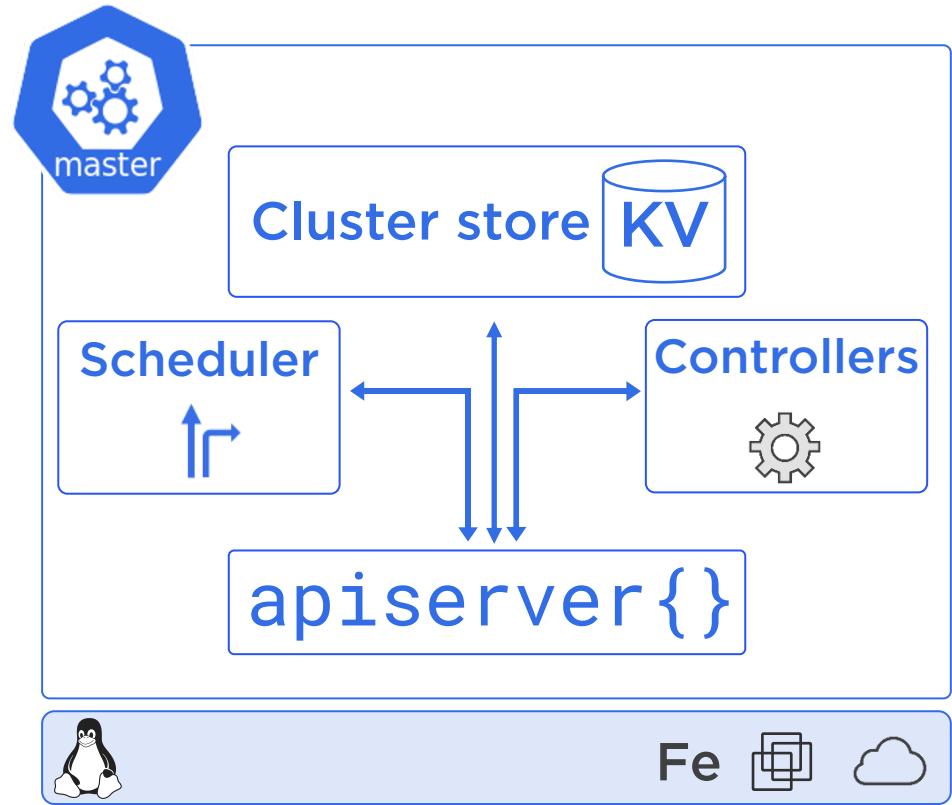
Kube-controller-manager

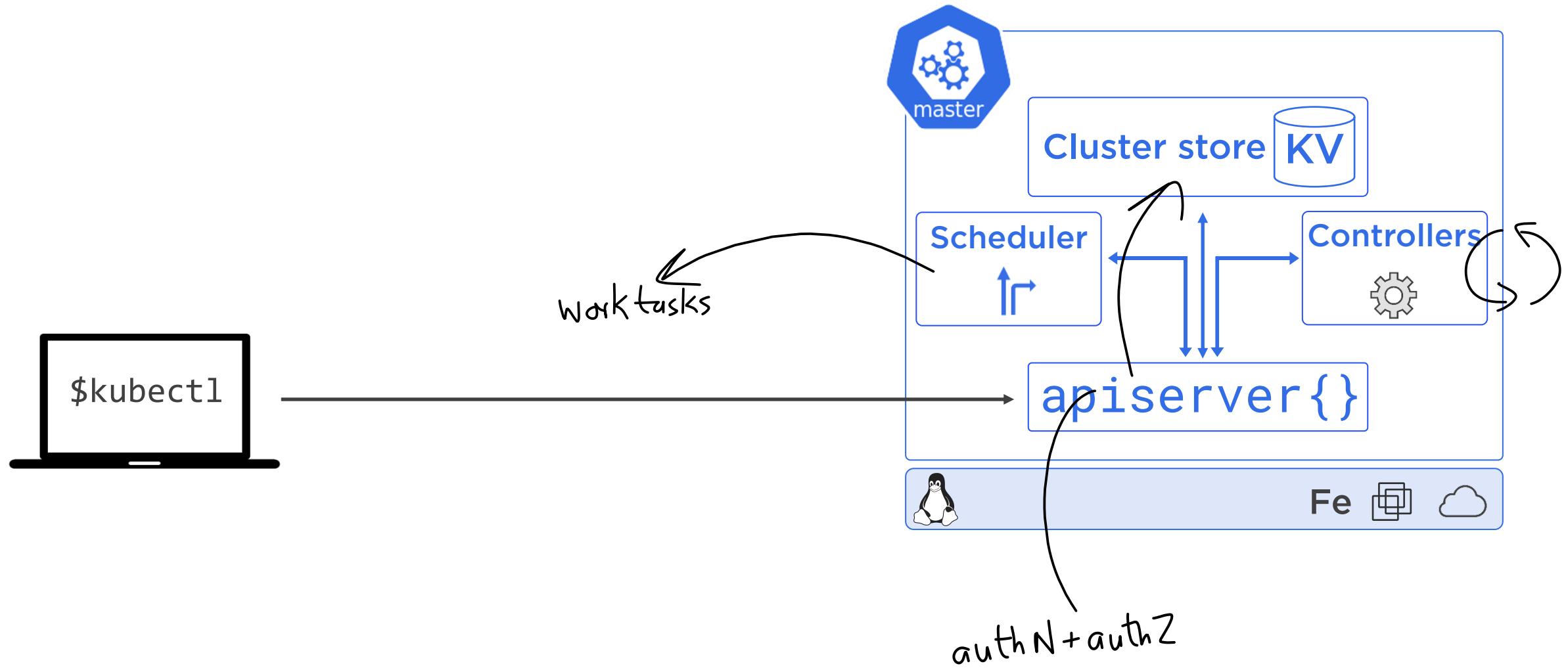
- Controller of controllers
 - Node controller
 - Deployment controller
 - Endpoints/EndpointSlice controller...
- Watch loops
- Reconciles observed state with desired state

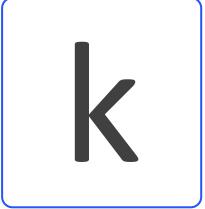


Kube-scheduler

- Watches API Server for new work tasks
- Assigns work to cluster nodes
 - Affinity/Anti-affinity
 - Constraints
 - Taints
 - Resources...







k

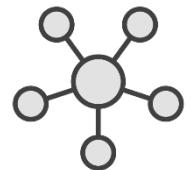
Kubelet

Main K8s agent



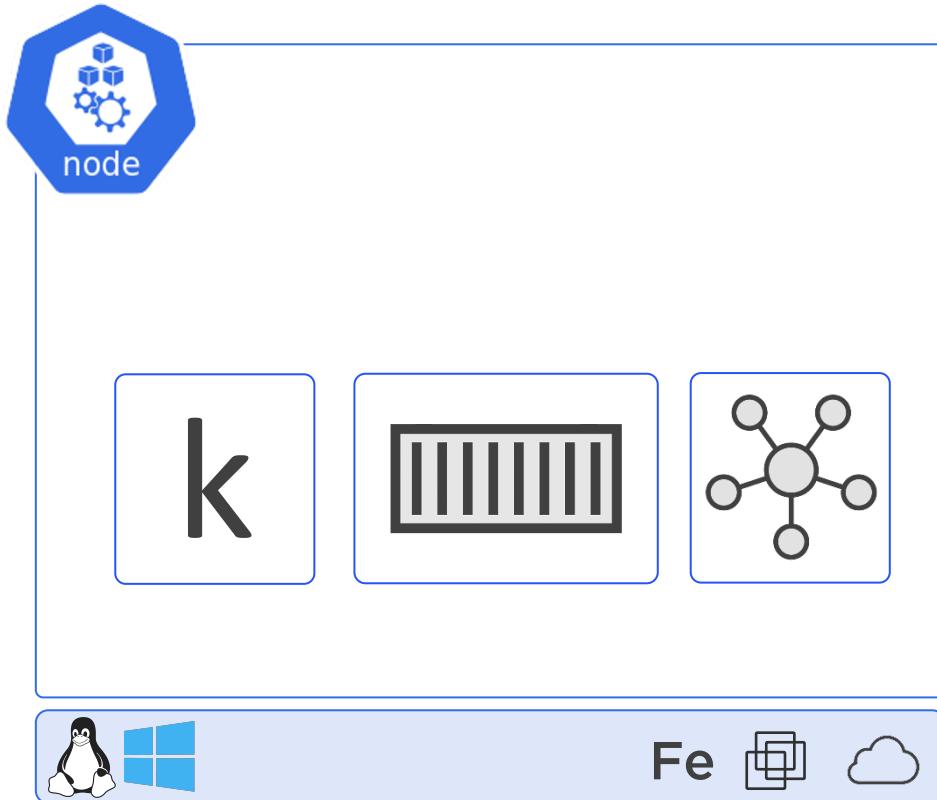
Container runtime

Docker, containerd, CRI-O, more...



Kube-proxy

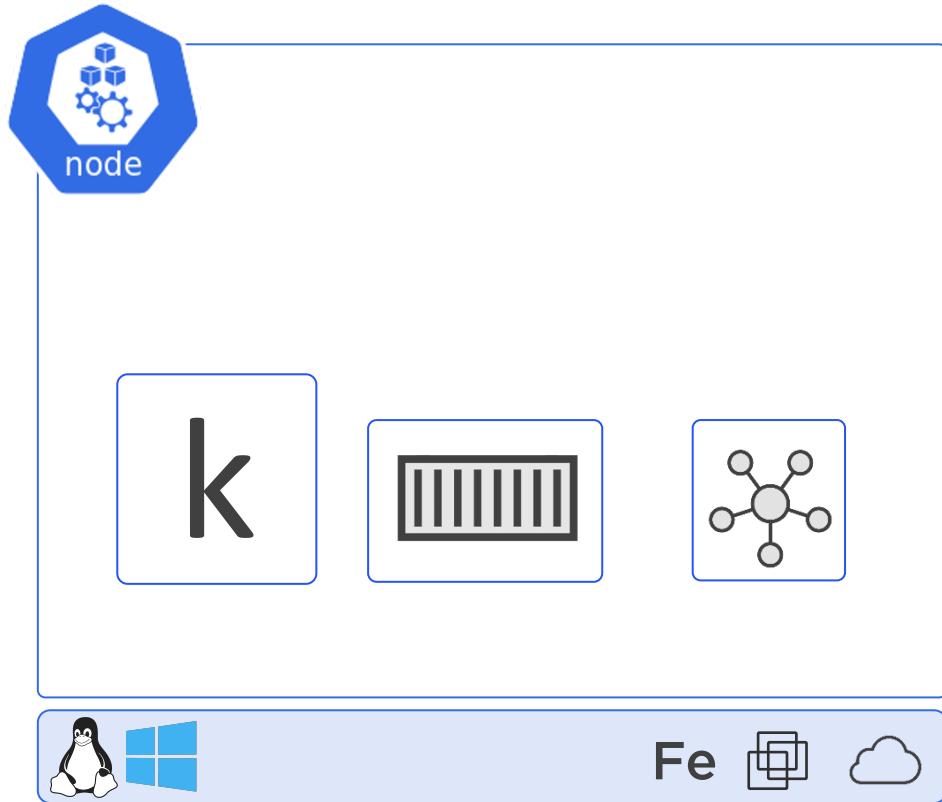
Vital role in networking





Kubelet

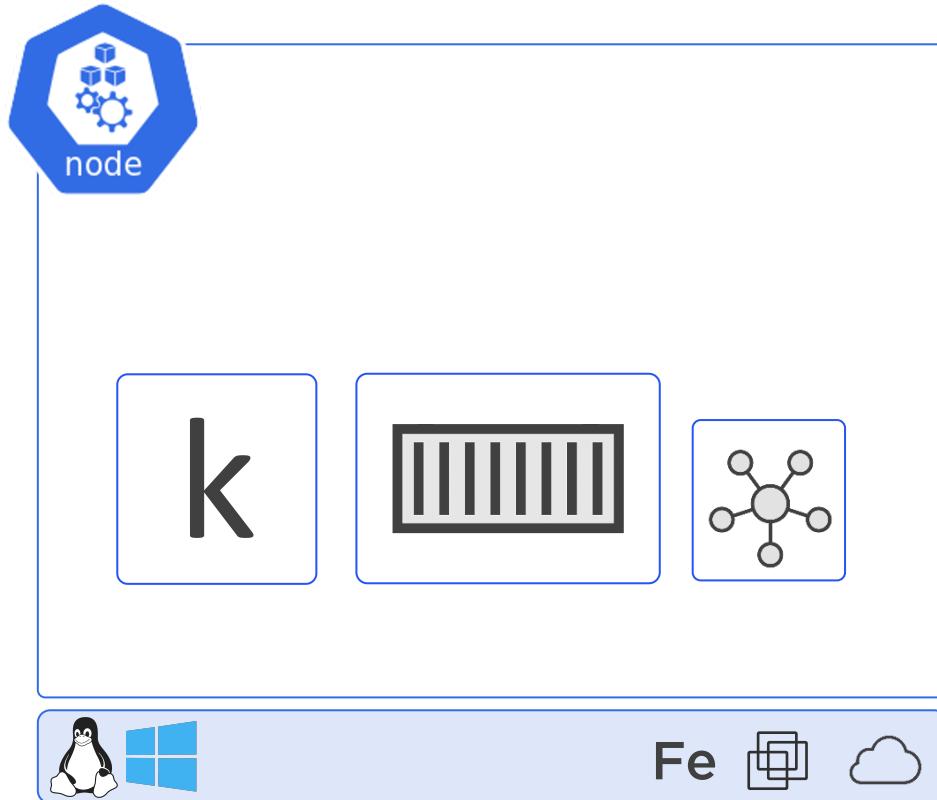
- Main Kubernetes agent
- Registers node with cluster
- Watches API Server for work tasks (Pods)
- Executes Pods
- Reports back to Masters

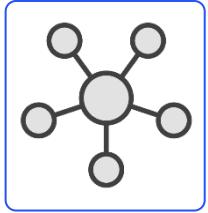




Container runtime

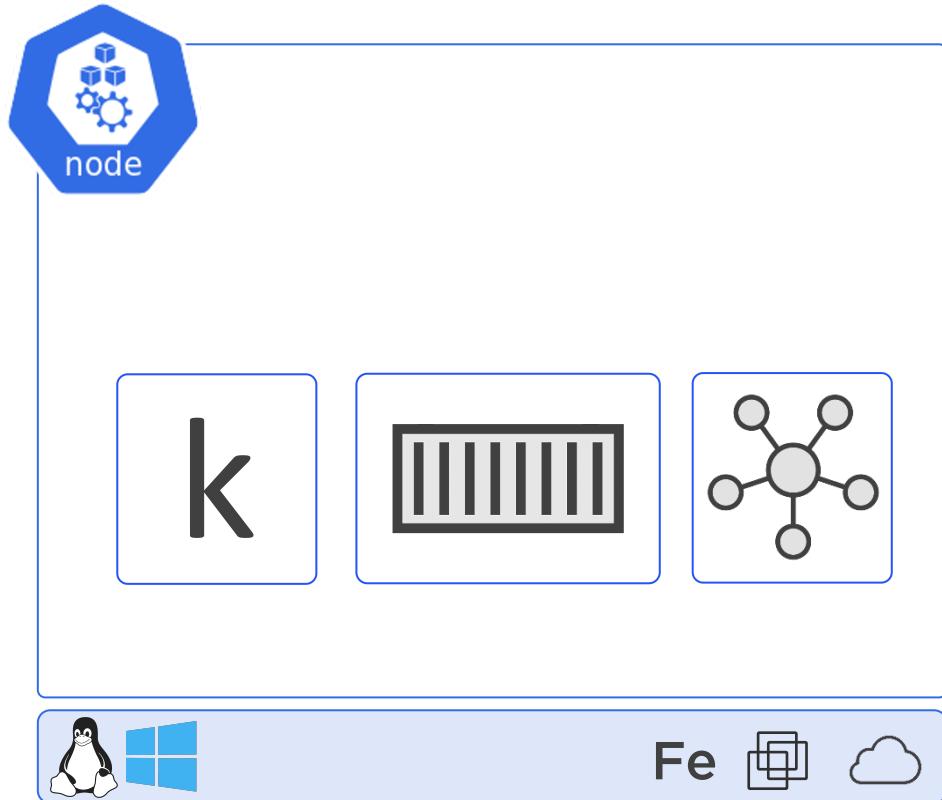
- Can be Docker
- Pluggable: Container Runtime Interface (CRI)
 - Docker, containerd, CRI-O, Kata...
- Low-level container intelligence

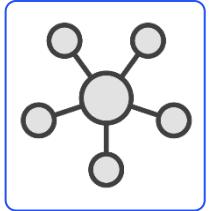




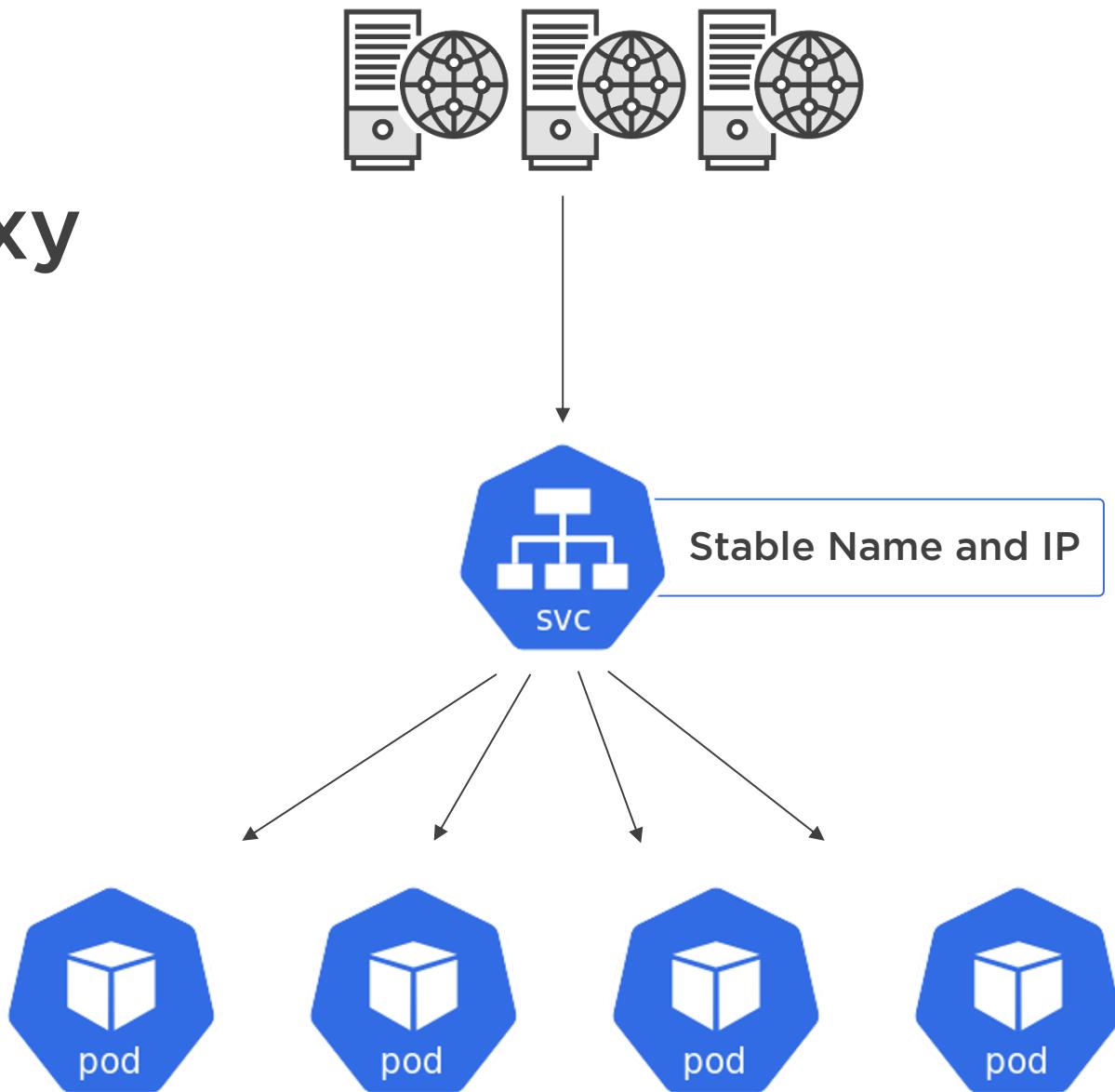
Kube-proxy

- Networking component
- Pod IP addresses
- Basic load-balancing





Kube-proxy





Virtual Kubelet



Nodeless Kubernetes





Virtual Kubelet

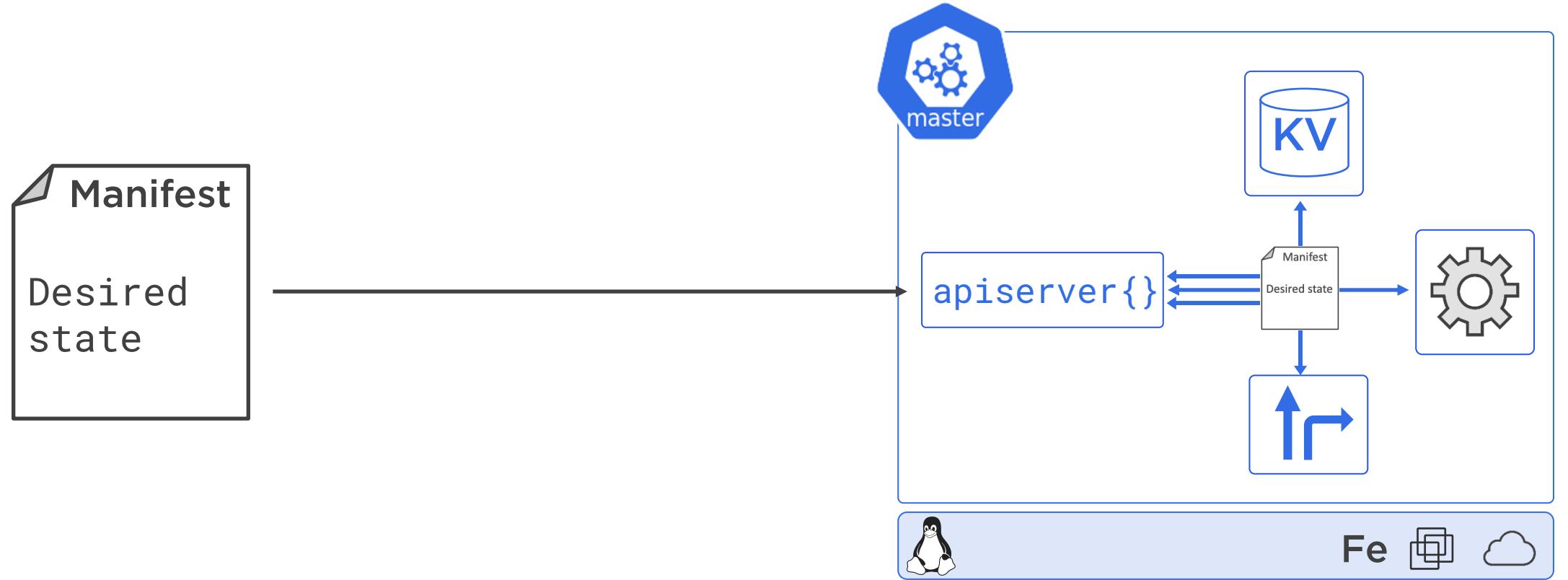
Pods run on cloud's hosted container back-end



Declarative model

Describe what you *want (desired state)* in a manifest file







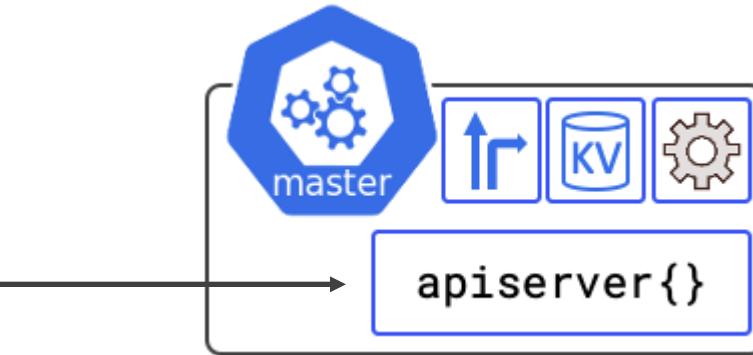
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: test
spec:
  replicas: 6
  selector:
    matchLabels:
      app: ps-test
  template:
    spec:
      containers:
        - name: c1
          image: web1:1.3
          ports: 8080
            - containerPort
              ...

```

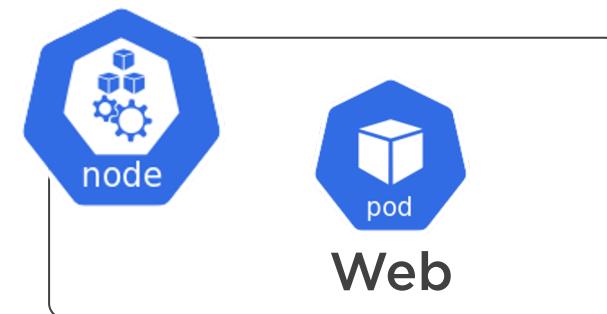
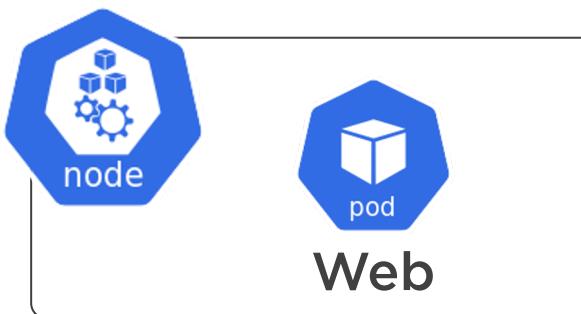
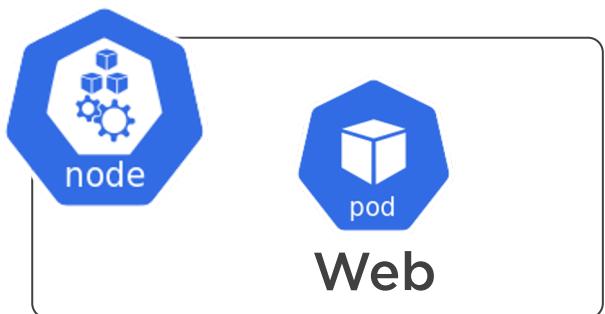
Declarative
(Declaring what you want)



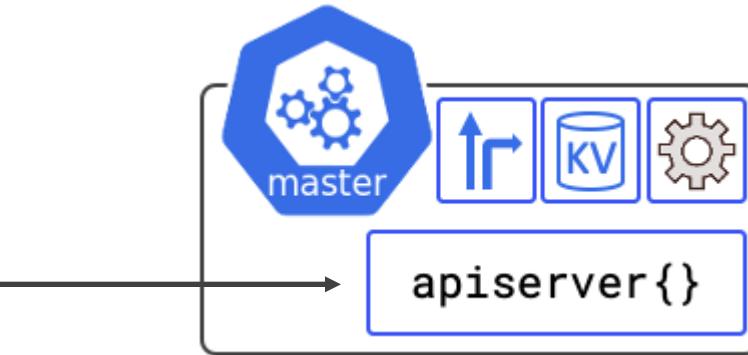
```
name: web  
replicas: 3  
...
```



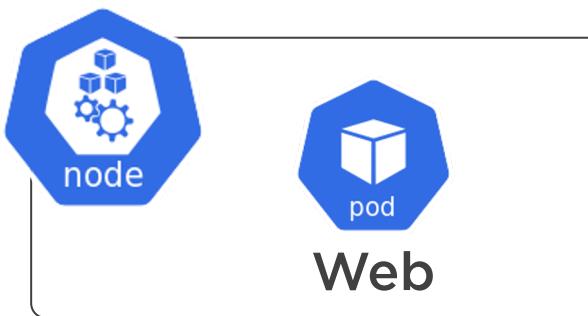
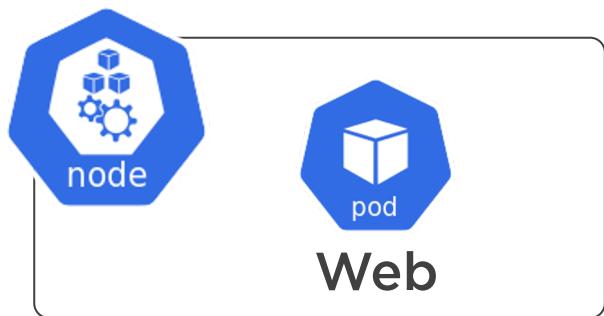
Desired state: 3
Observed state: 3



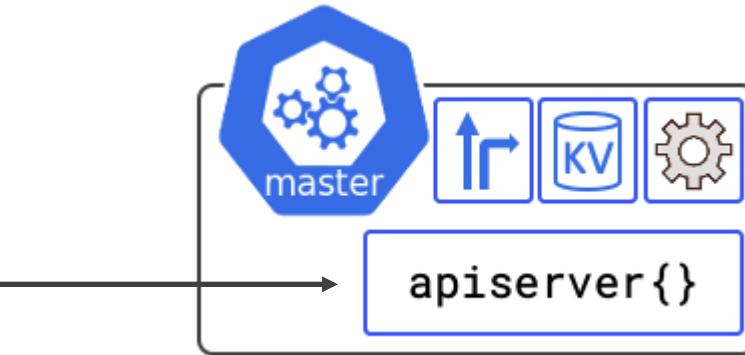
```
name: web  
replicas: 3  
...
```



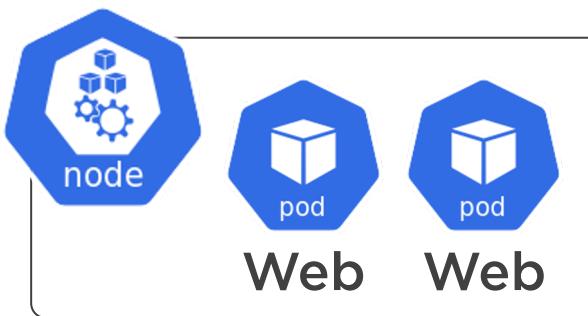
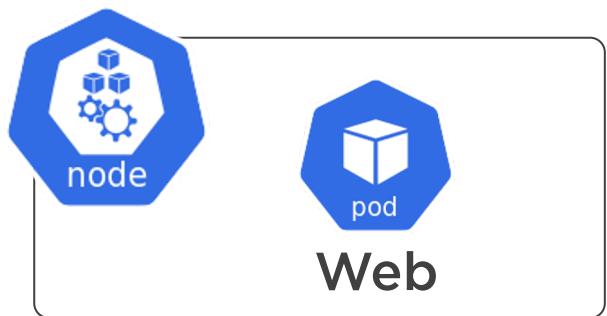
Desired state: 3
Observed state: 2

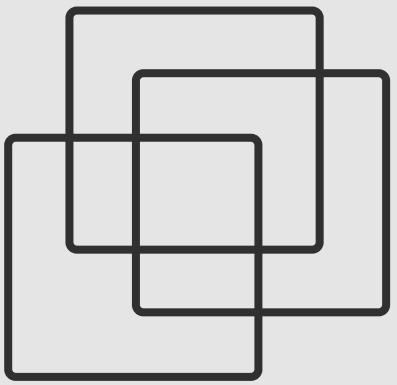


```
name: web  
replicas: 3  
...
```

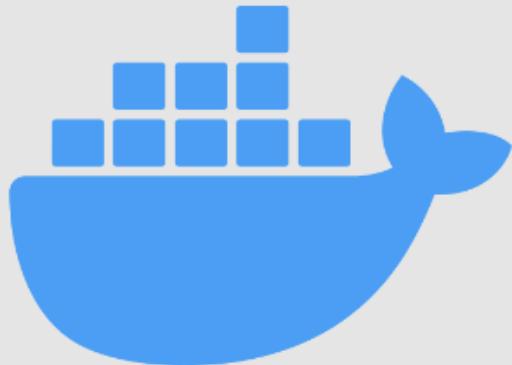


Desired state: 3
Observed state: 3





Virtual Machine

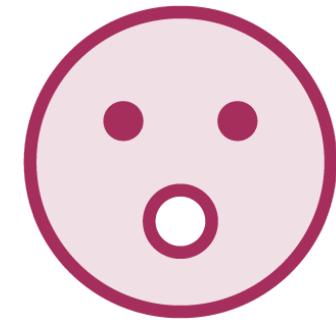


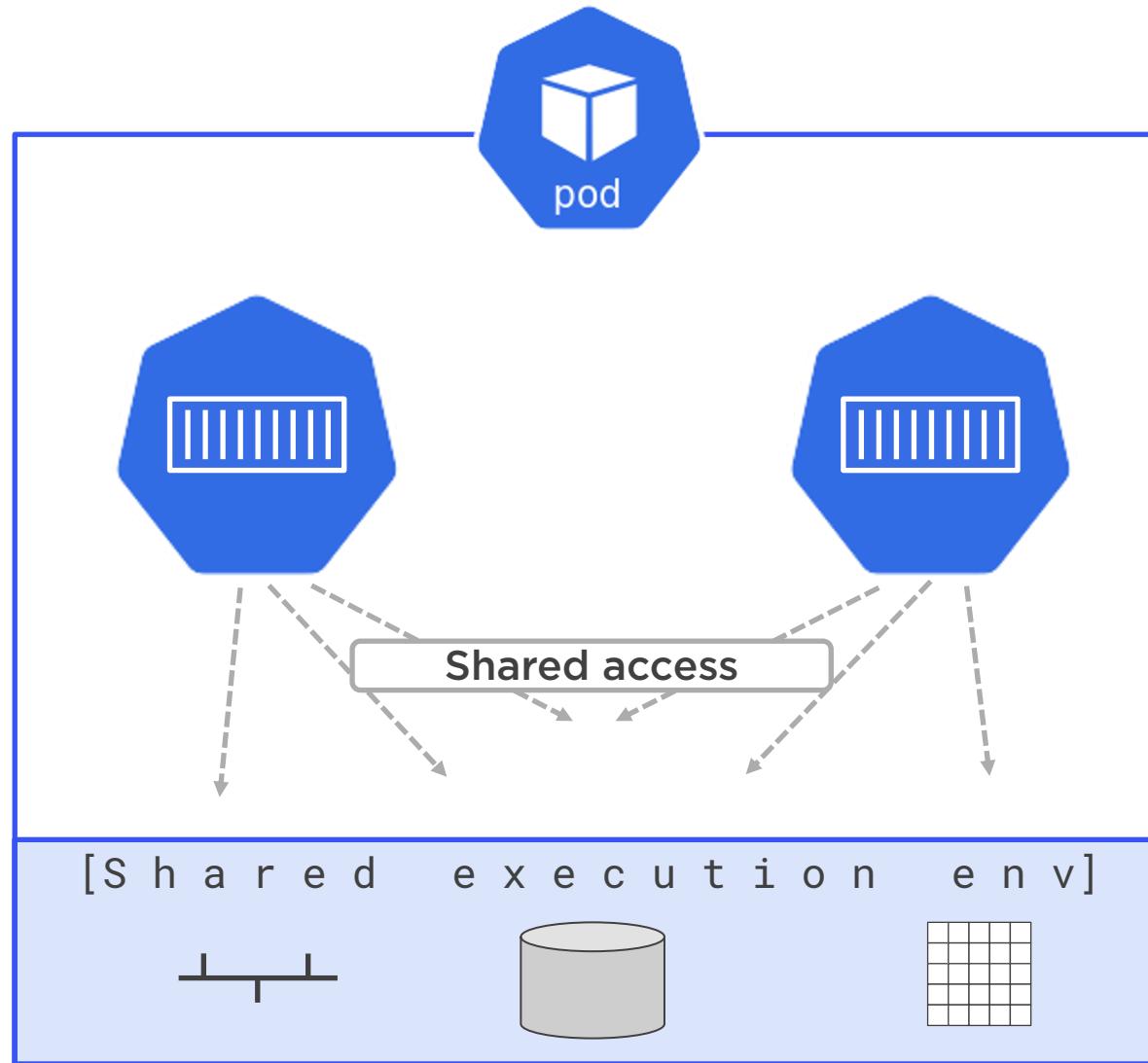
Container

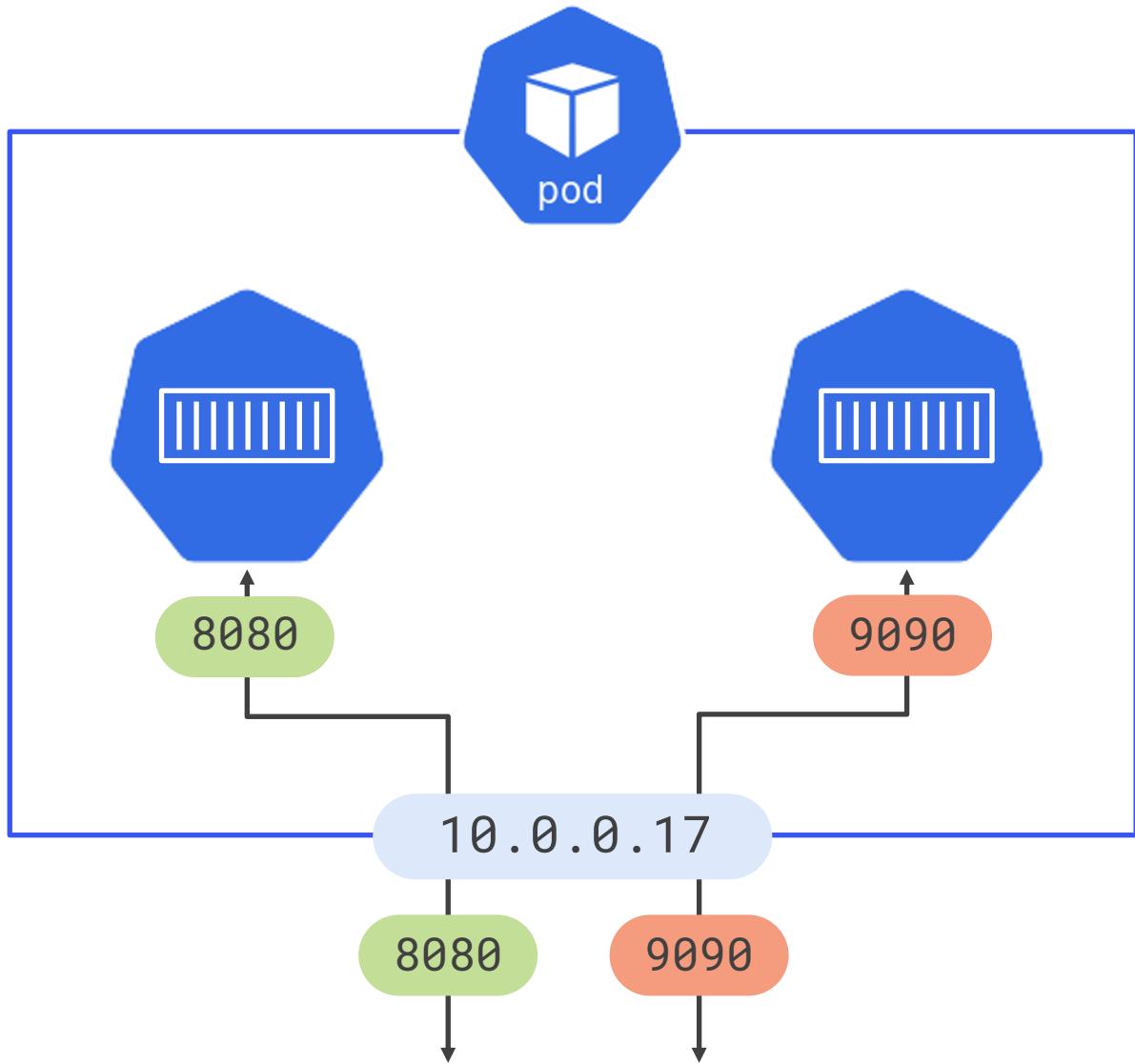


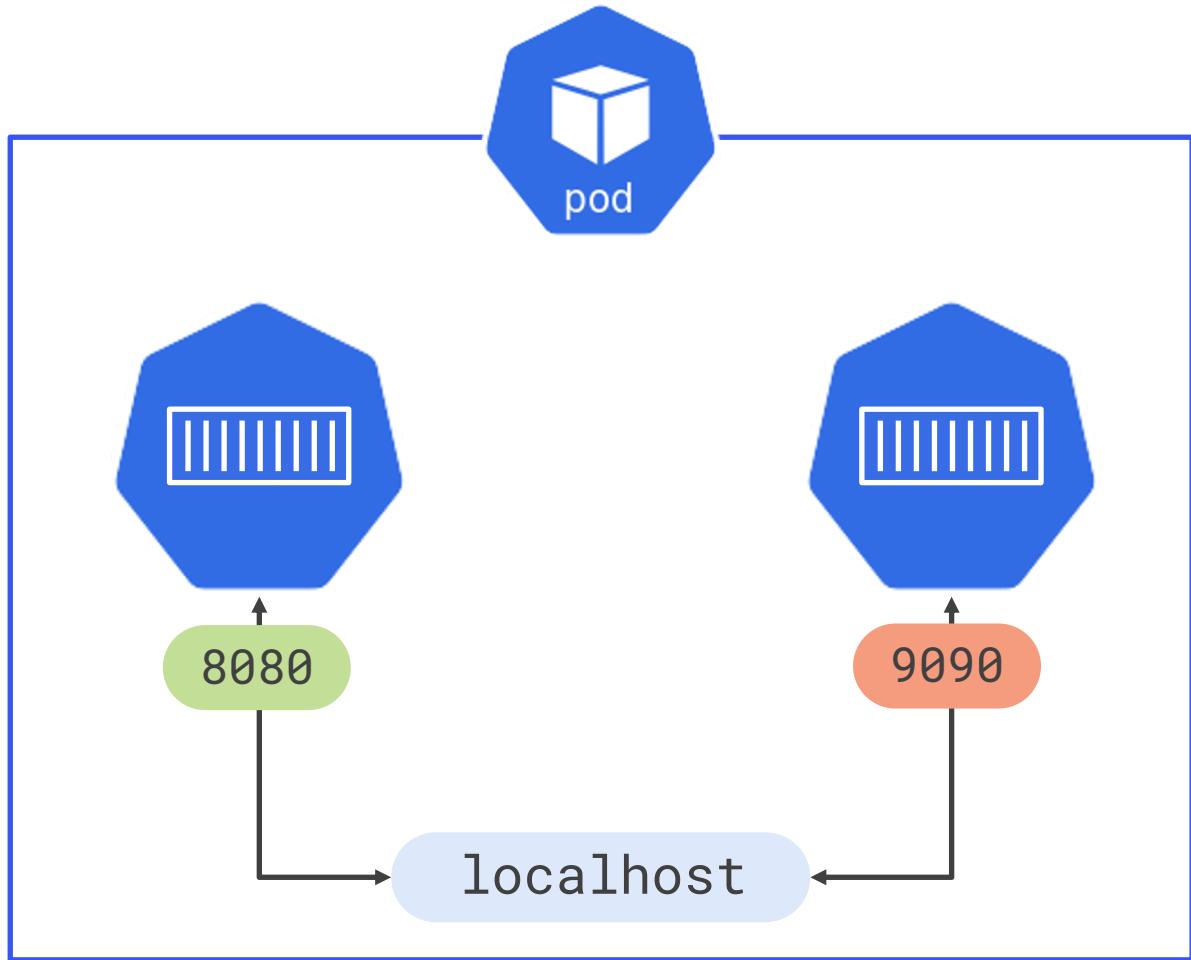
Pod

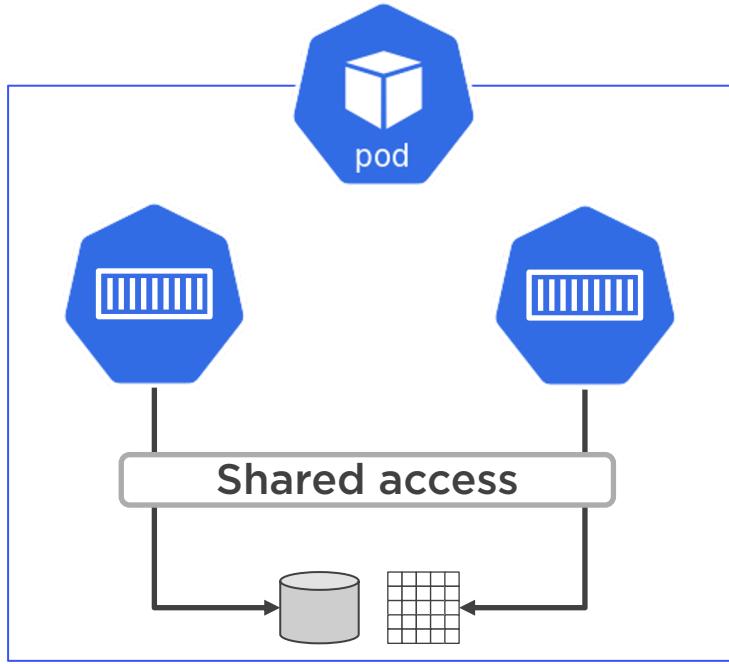






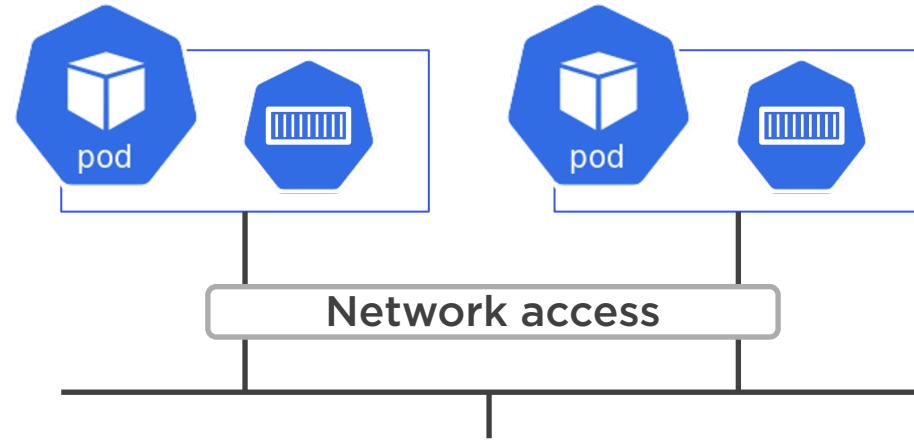






Tightly coupled

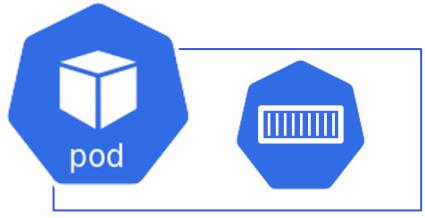
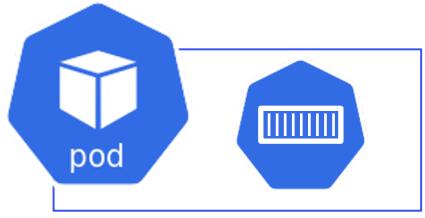
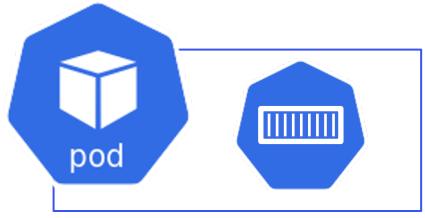
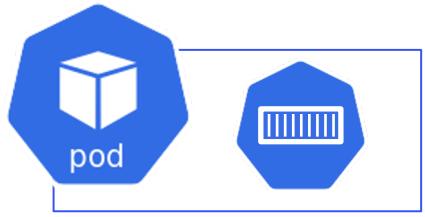
When two containers (app services) absolutely need to share vols, memory etc.



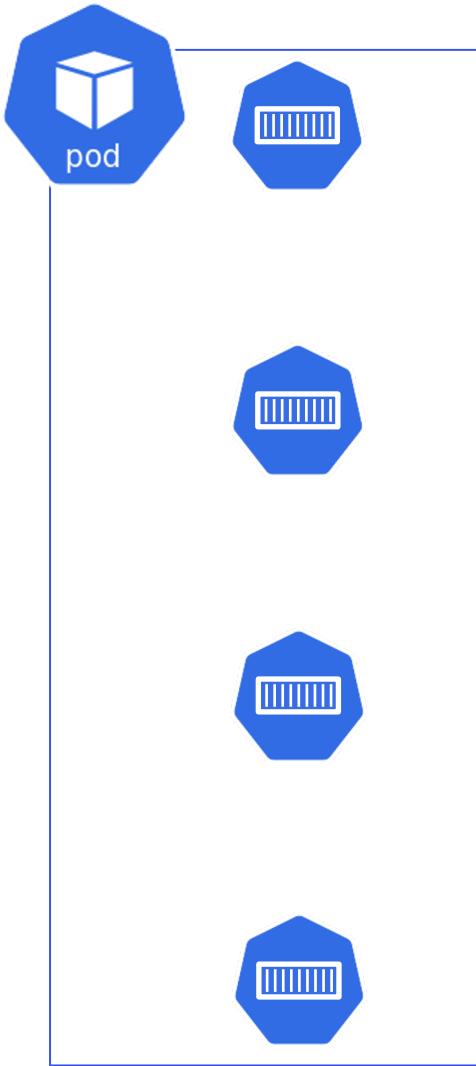
Loosely coupled

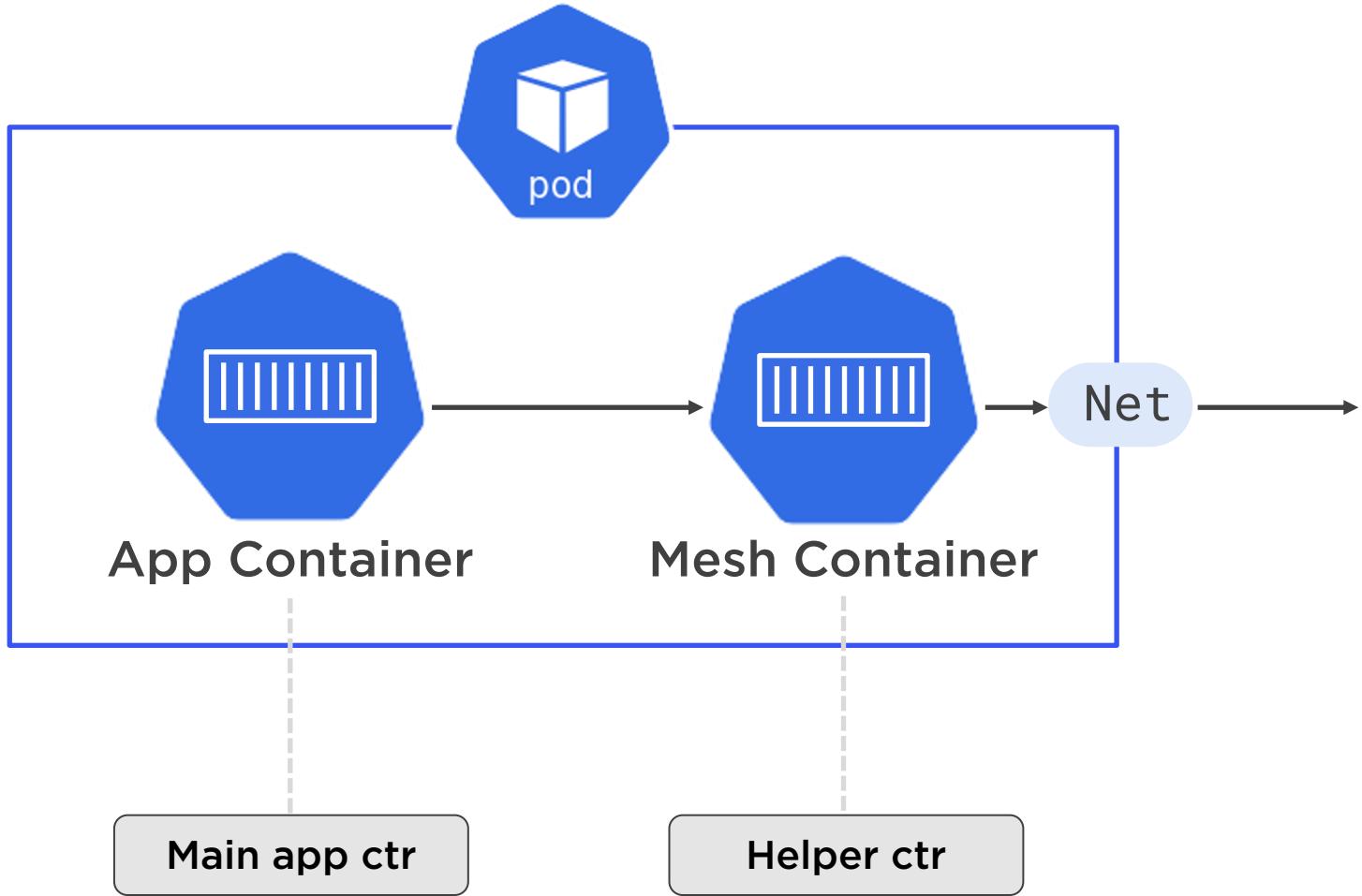
When two containers (app services) don't absolutely need to share resources

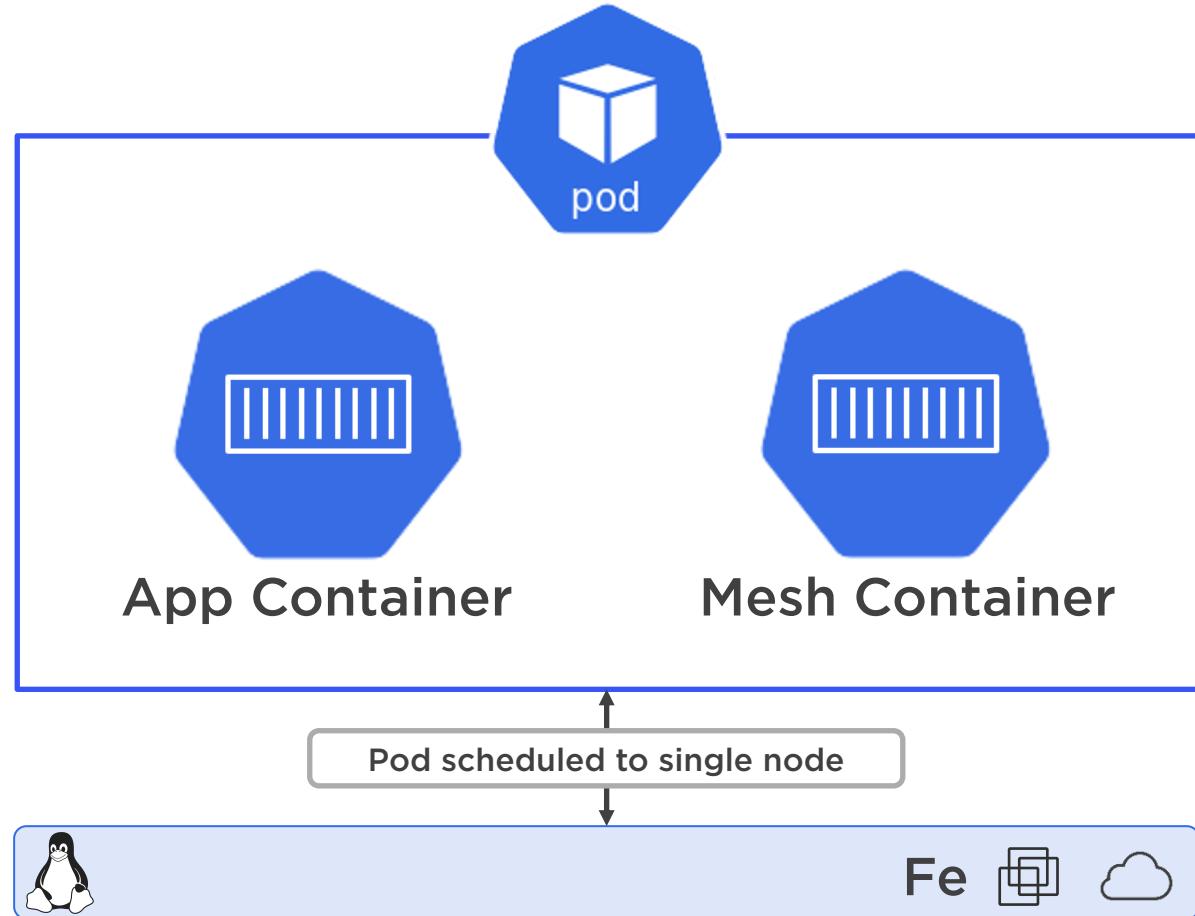




Scaling











Annotations

Labels

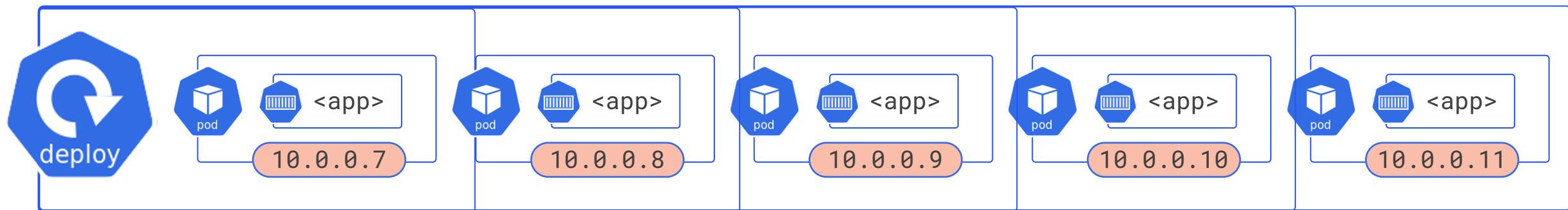
Policies

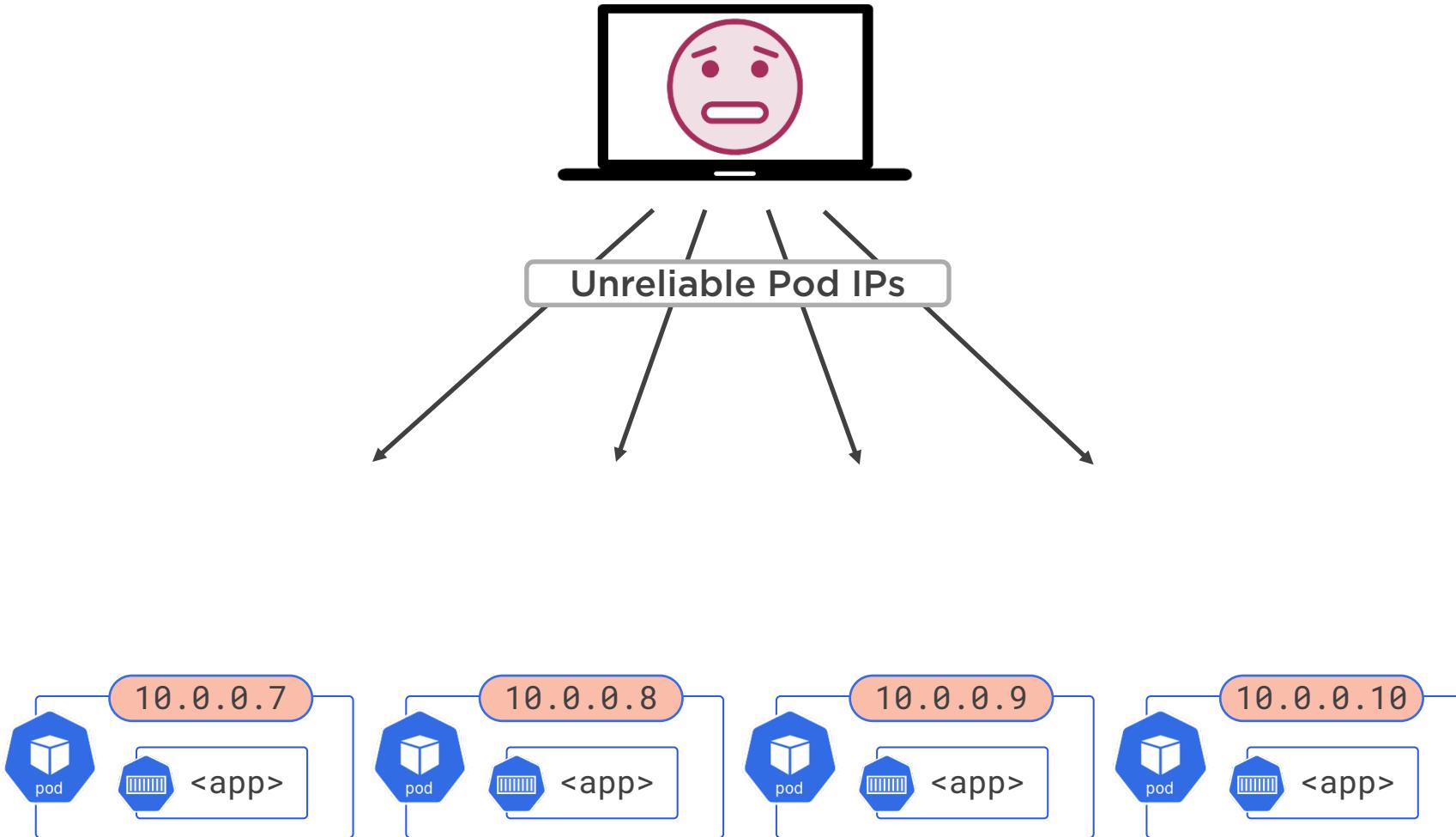
Resources

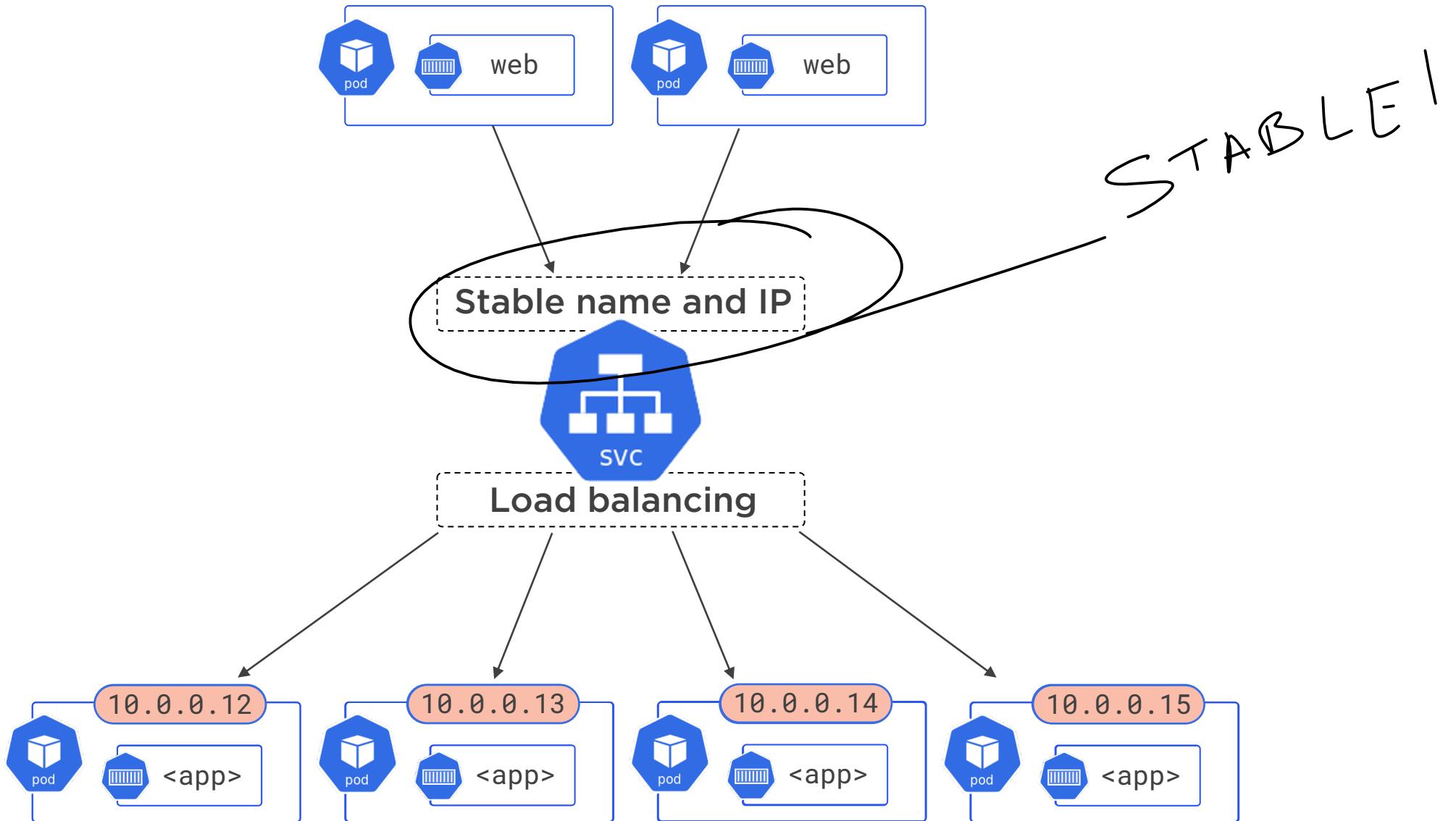
Co-scheduling containers

...







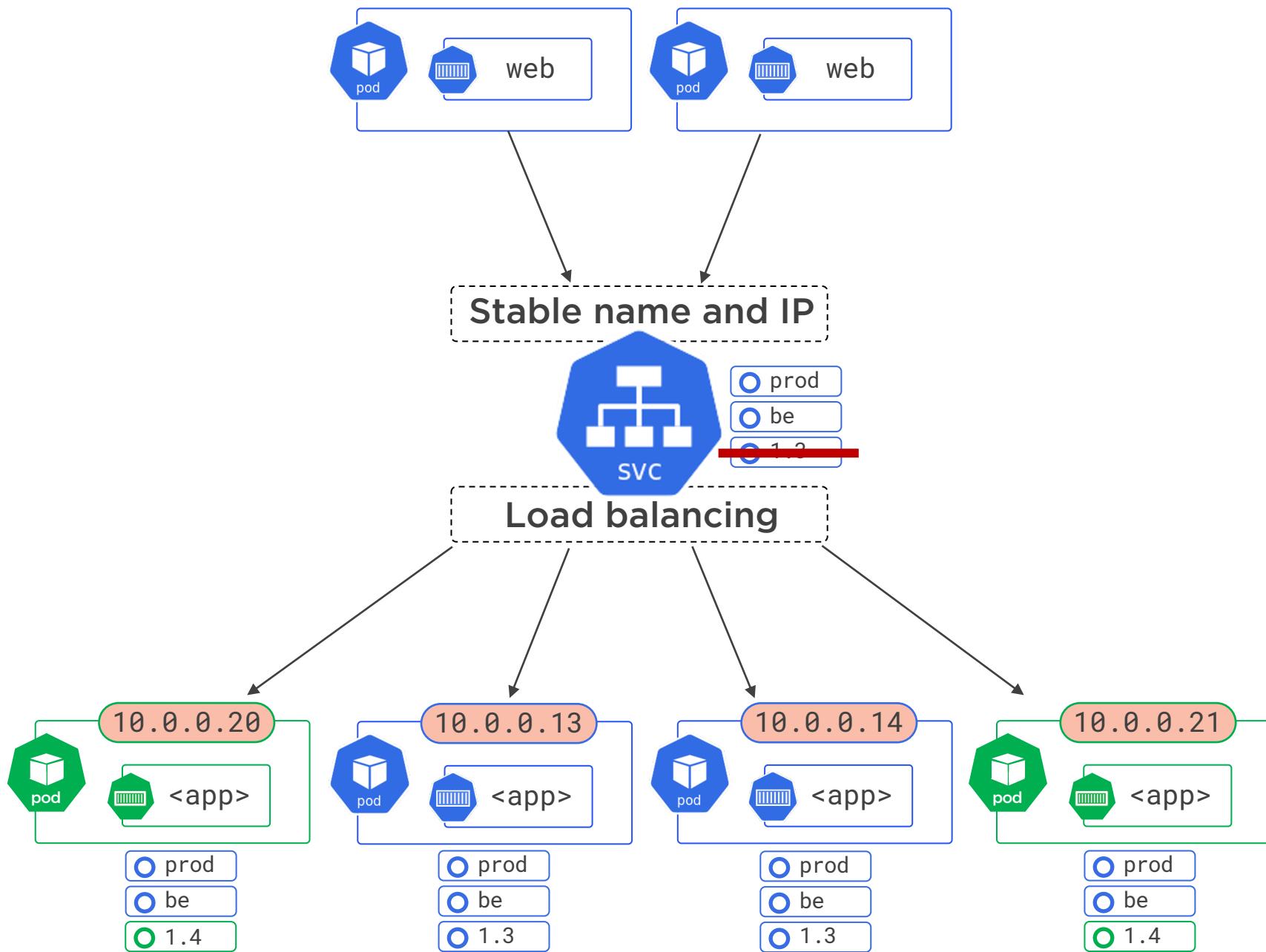


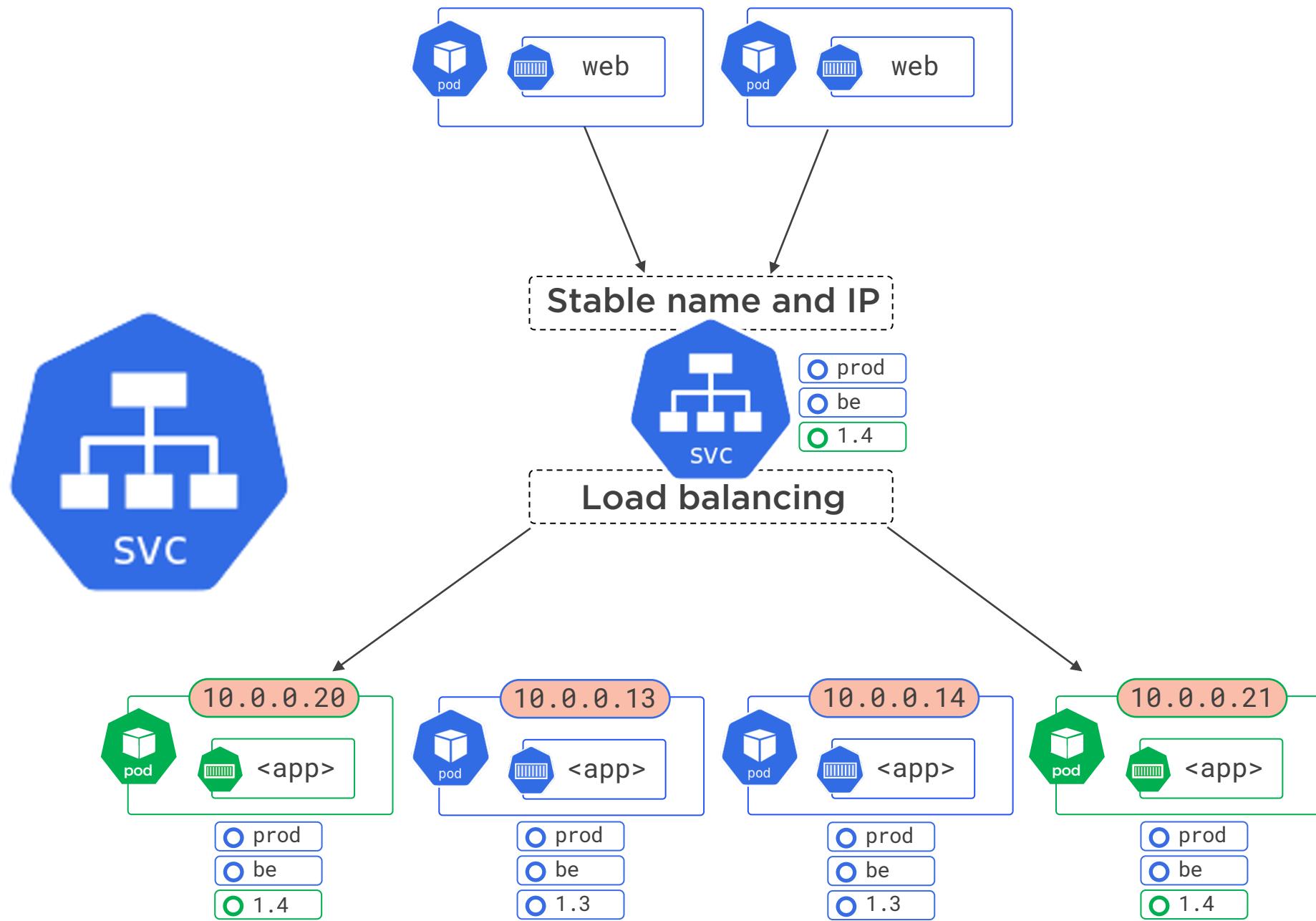


Labels

Labels are very simple and very powerful!



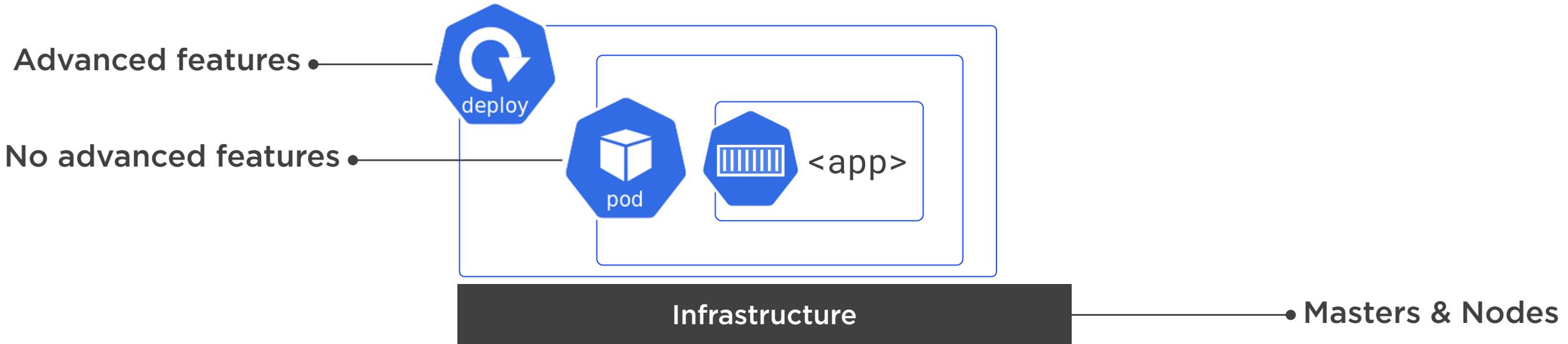






- Only sends traffic to healthy Pods**
- Can do session affinity**
- Can send traffic to endpoints outside the cluster**
- Can do TCP and UDP**







Stateless apps



One instance on every node

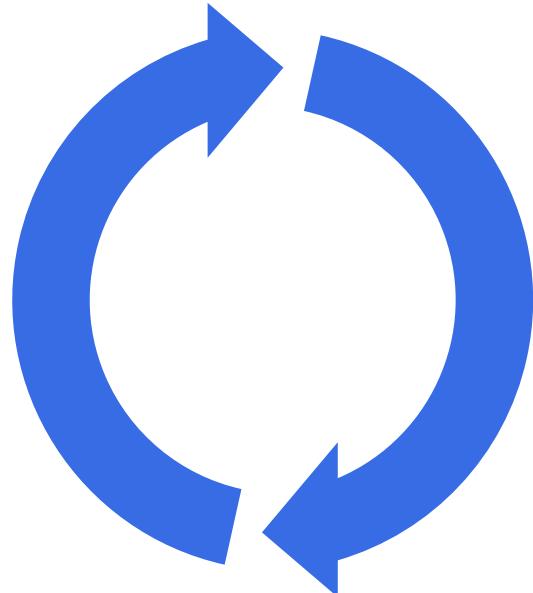


Stateful apps...



Time-based short-lived jobs





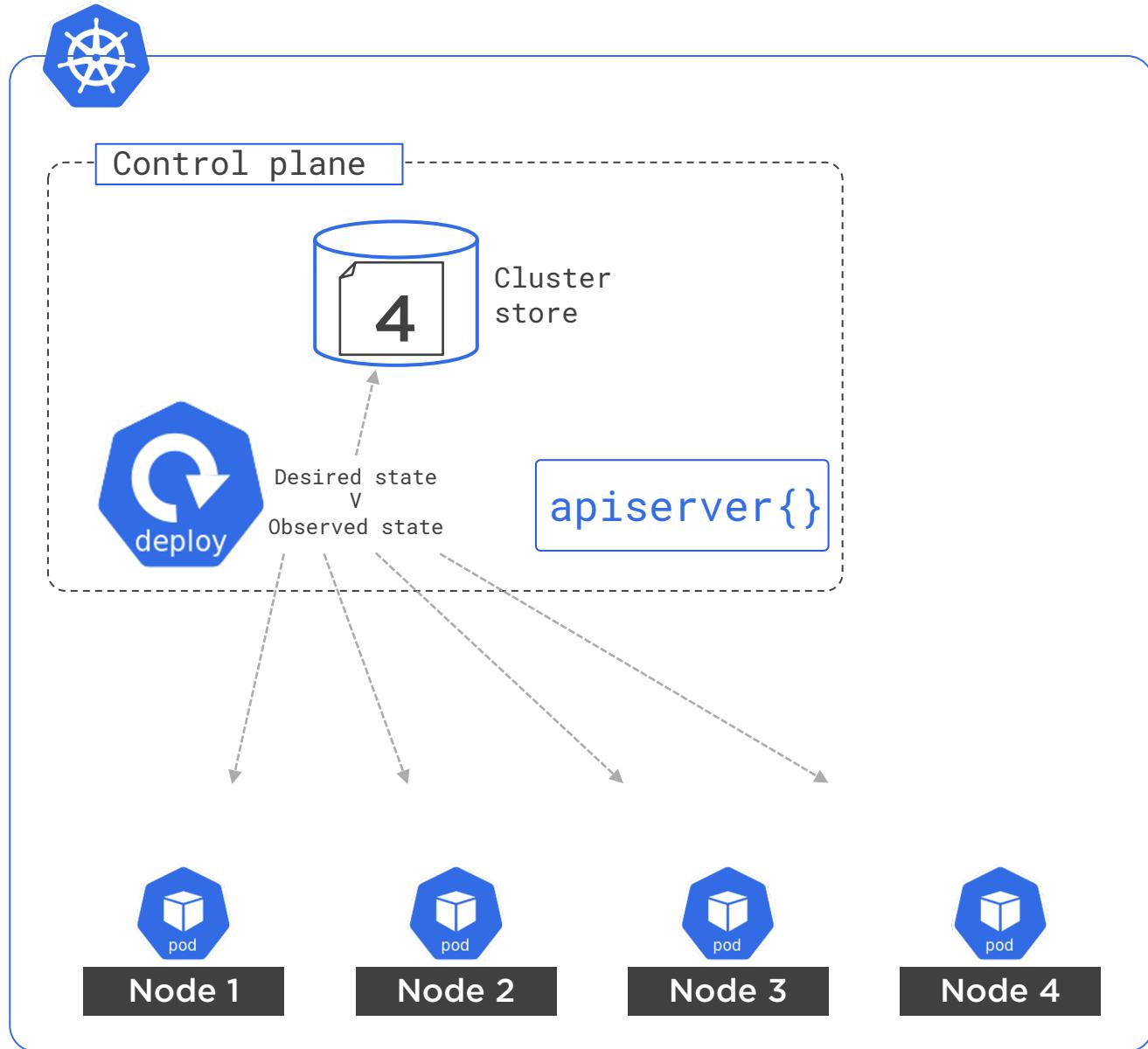
Deployment Controller/Reconciliation loop

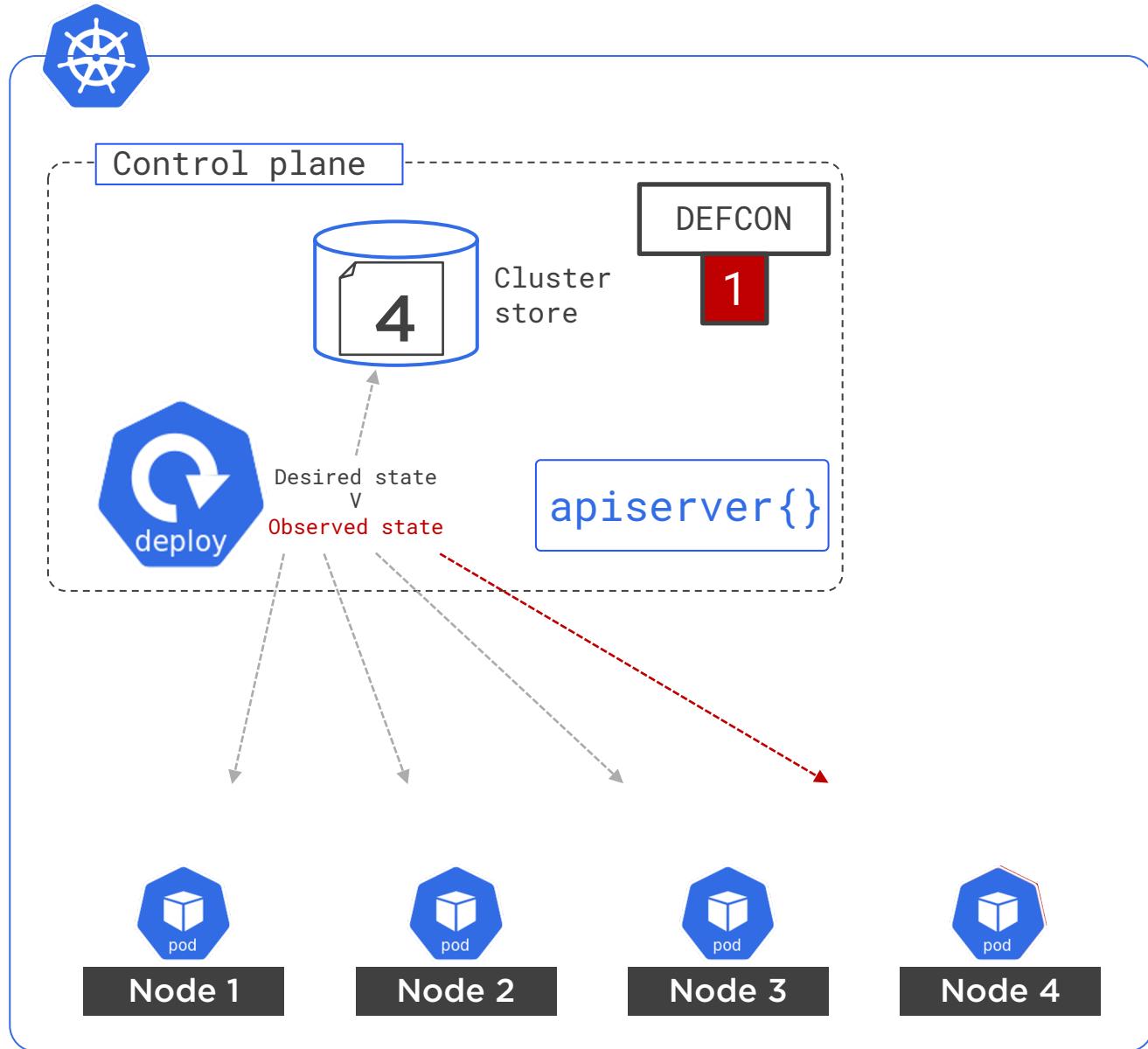
Watches API Server for new Deployments

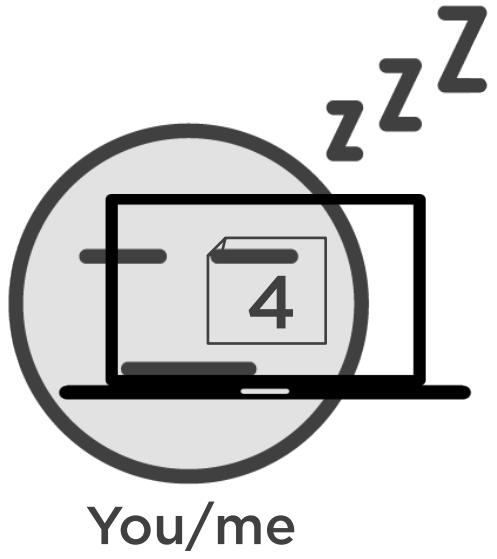
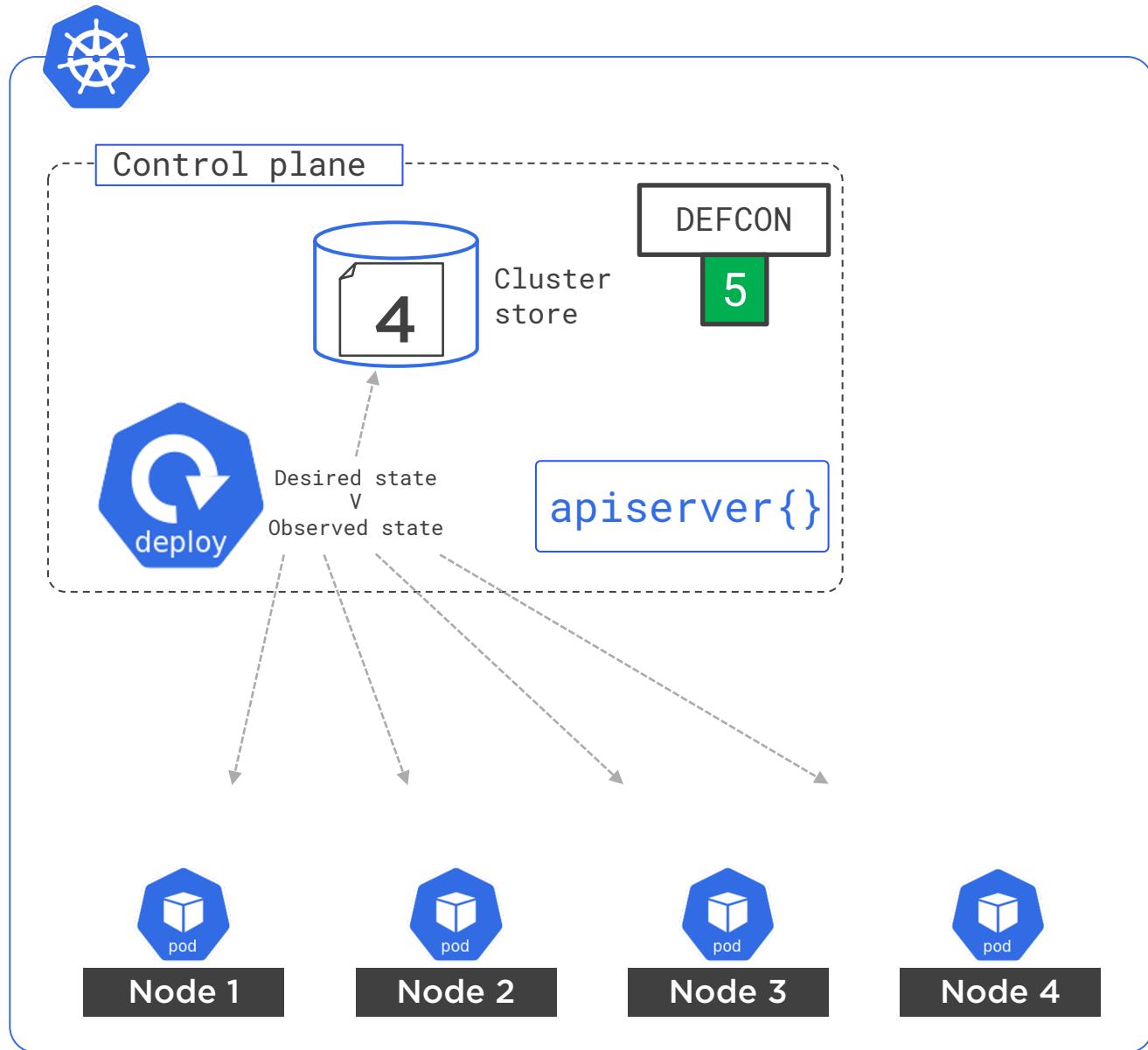
Implements them

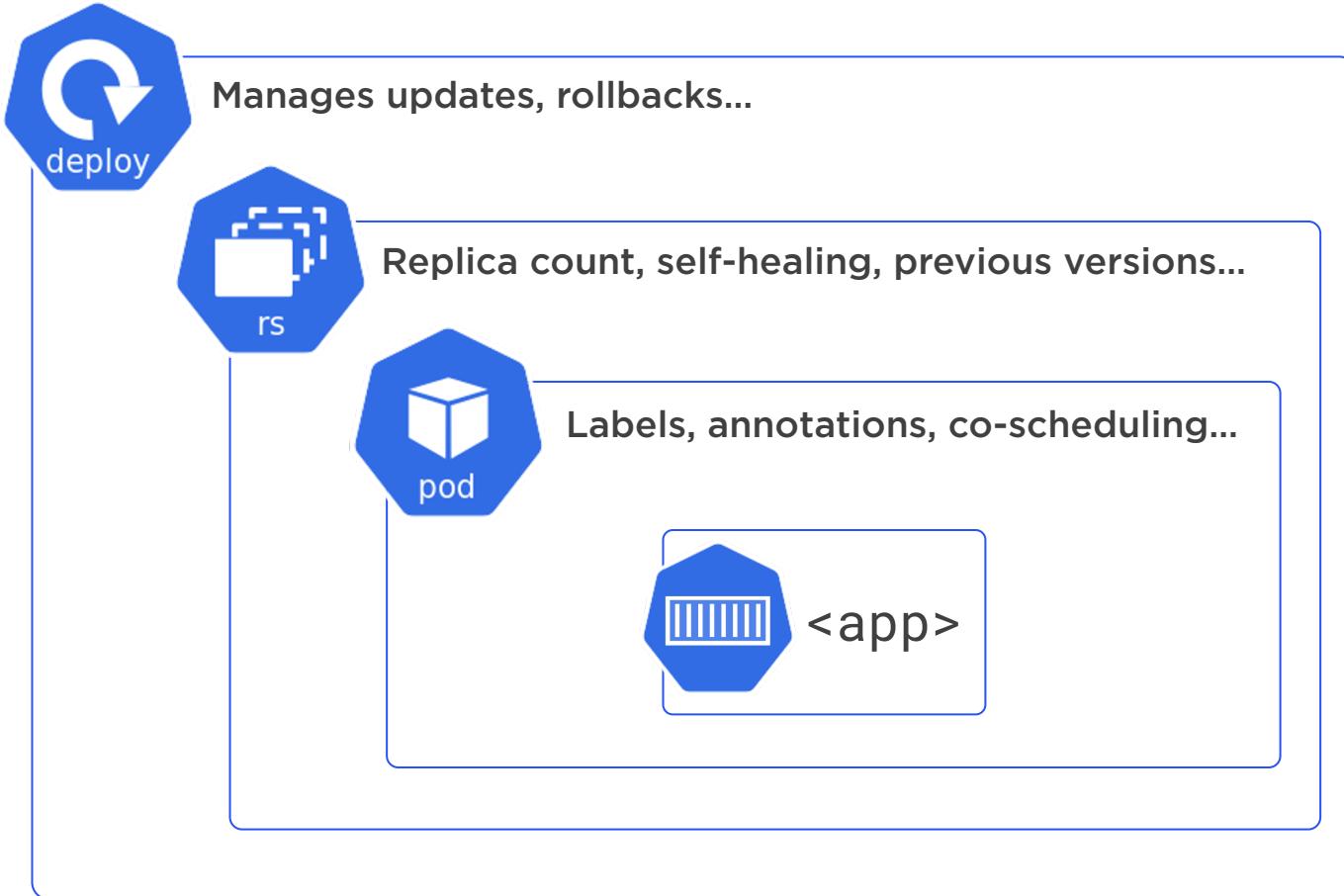
Constantly compares *observed state* with
desired state

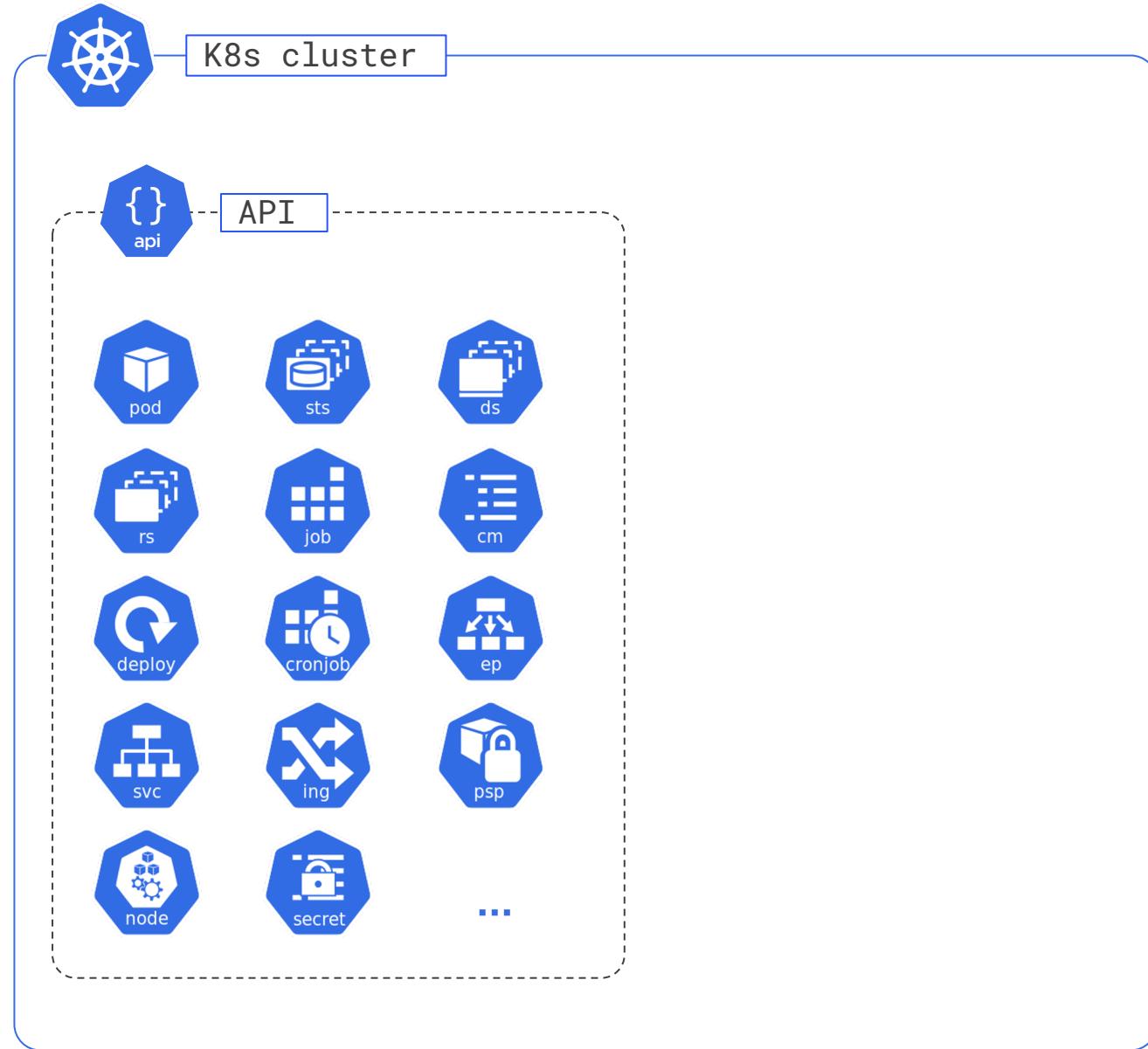


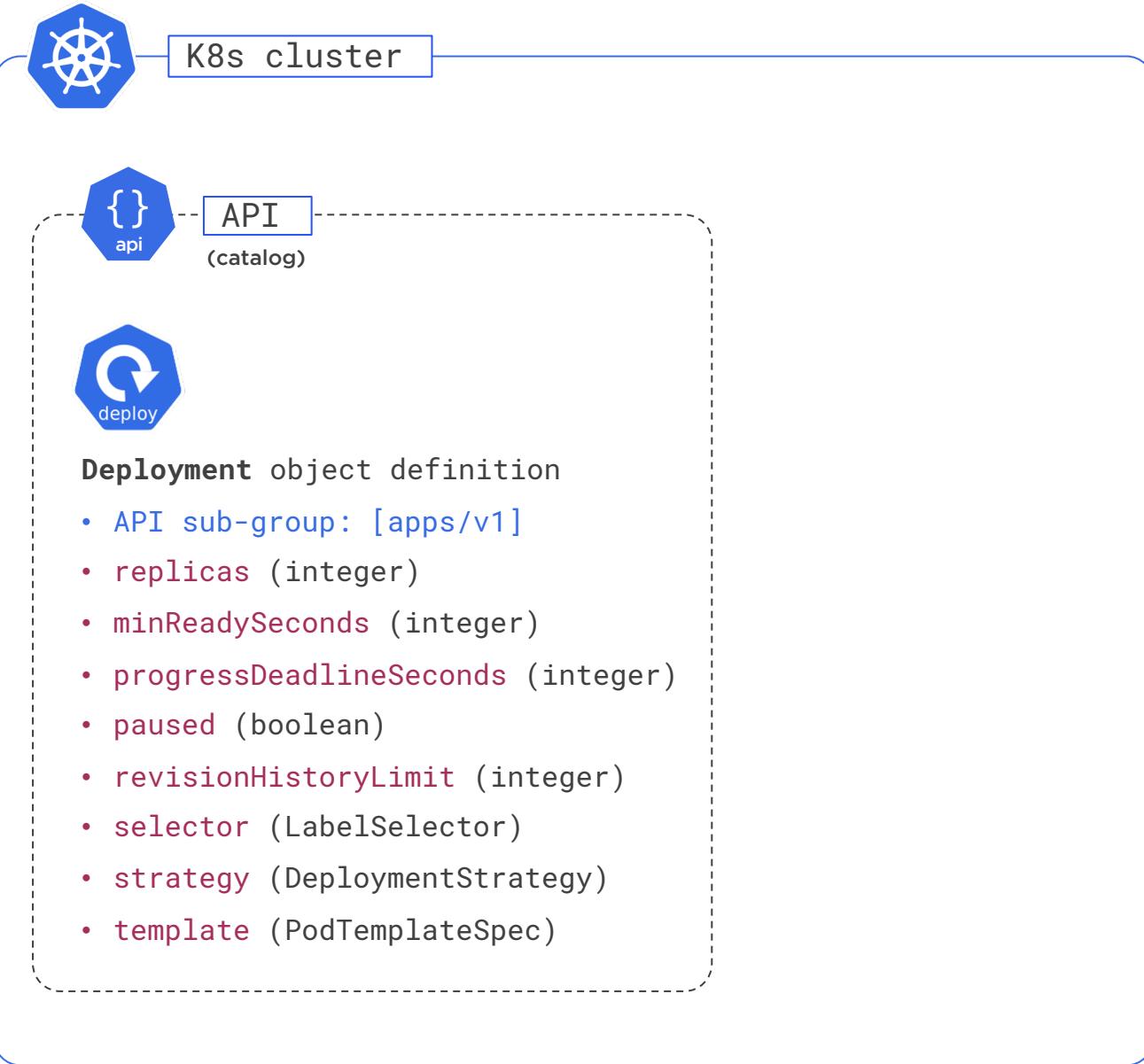


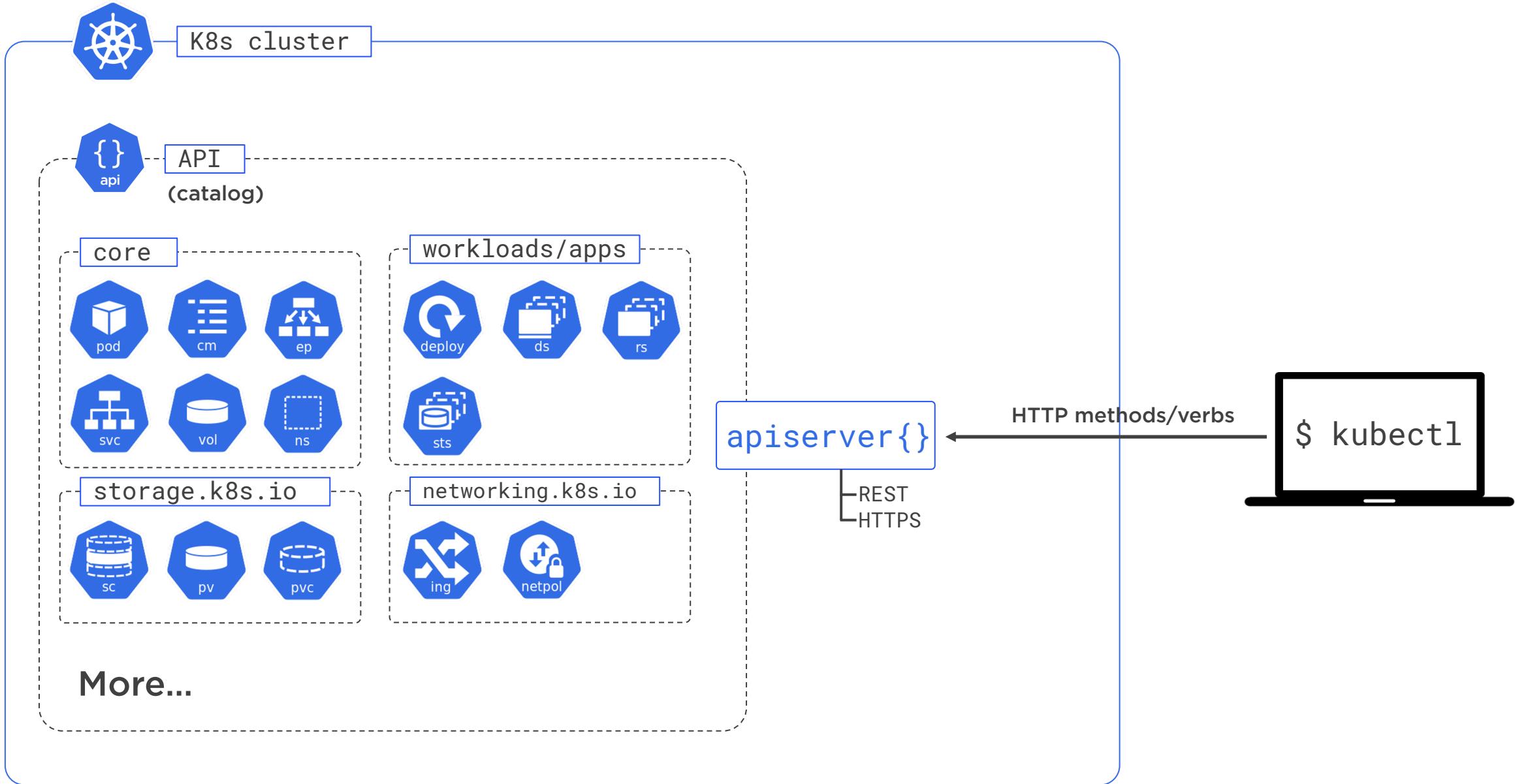


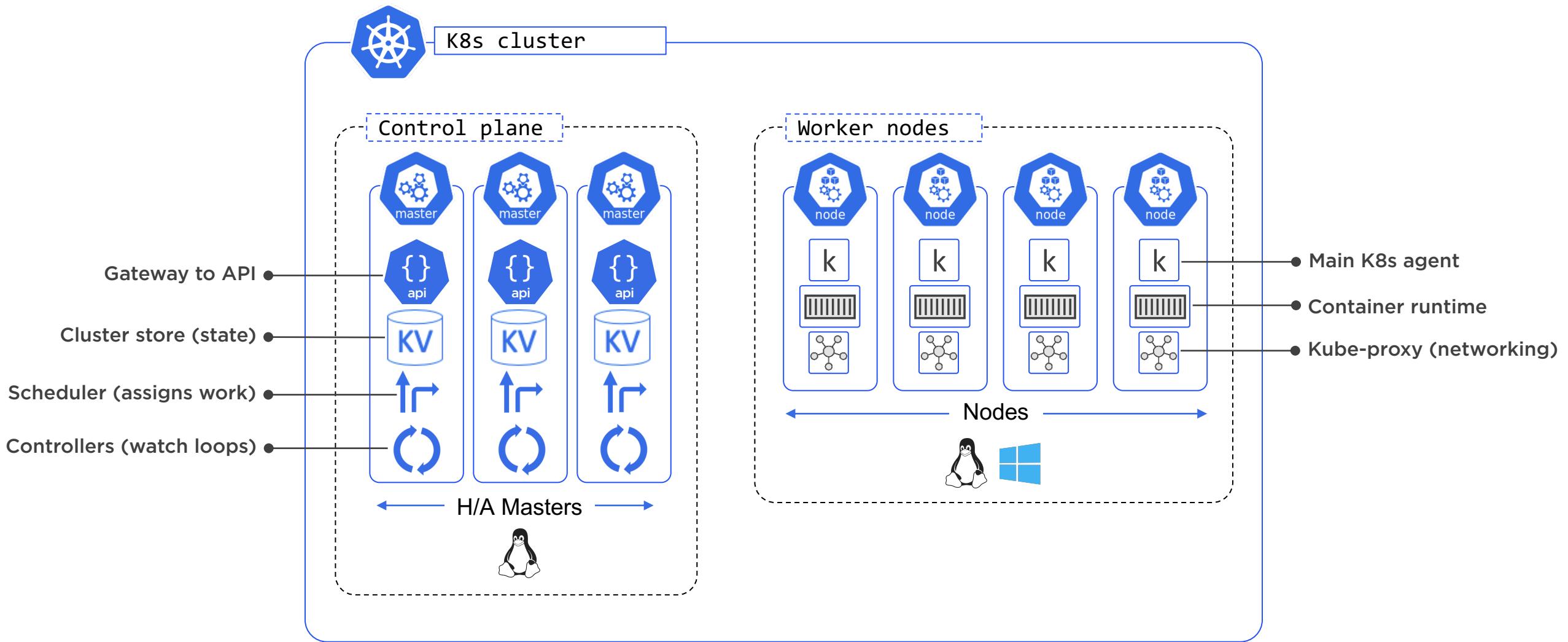


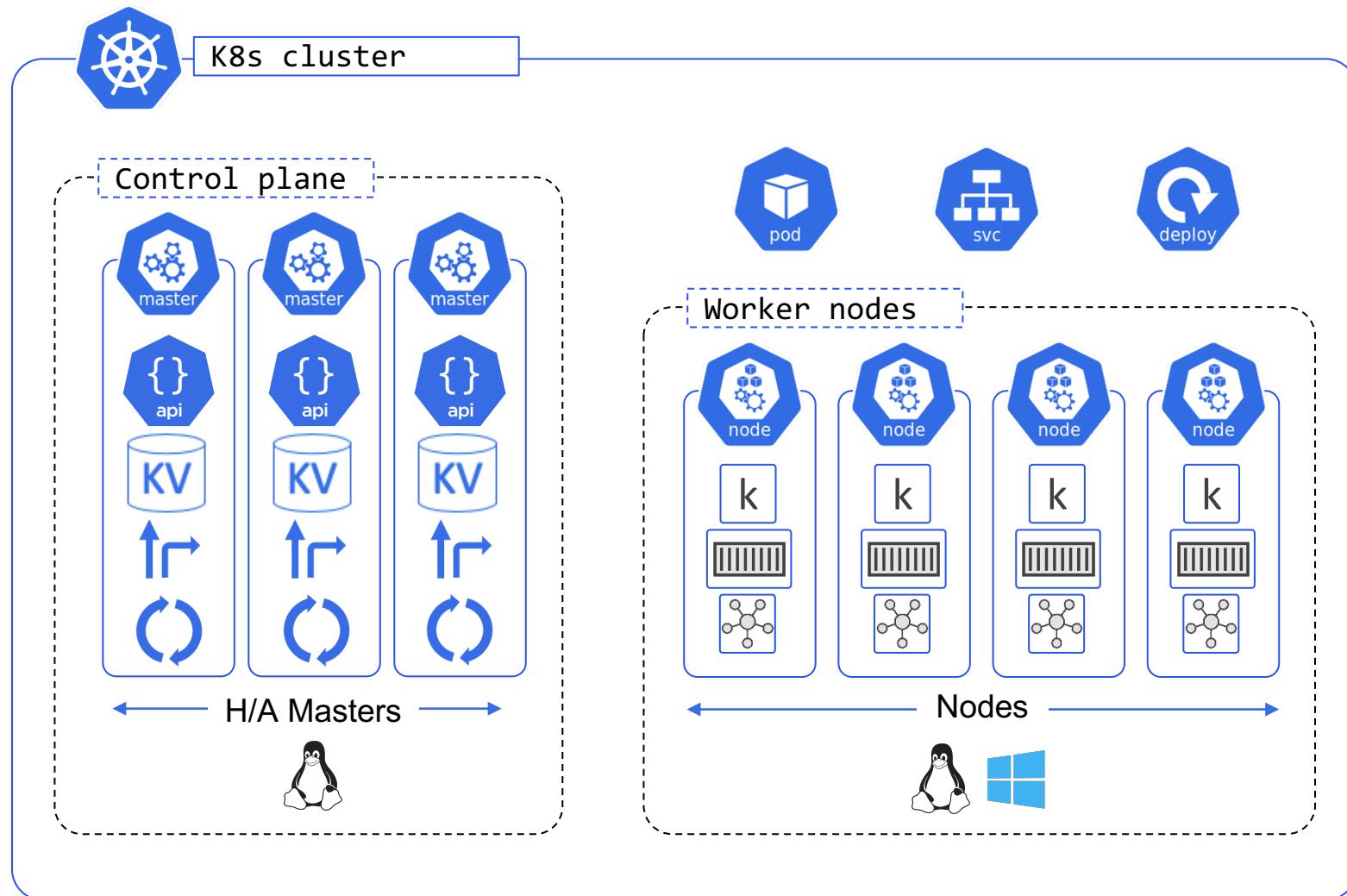


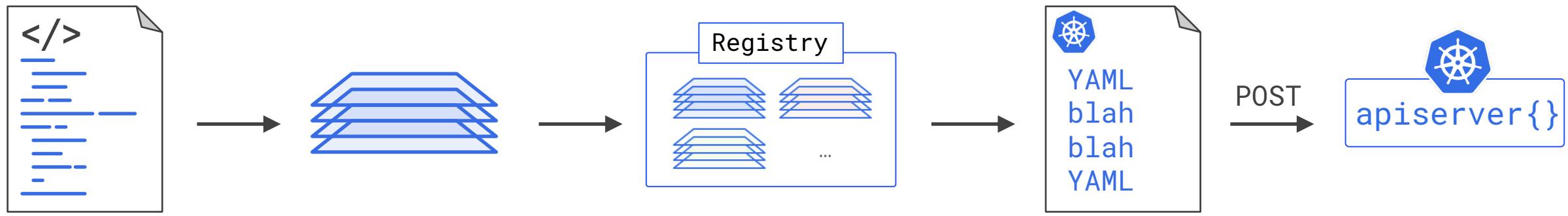












Quick run through these bits...

Our main focus

Course recommendations

Docker Deep Dive

By Nigel Poulton

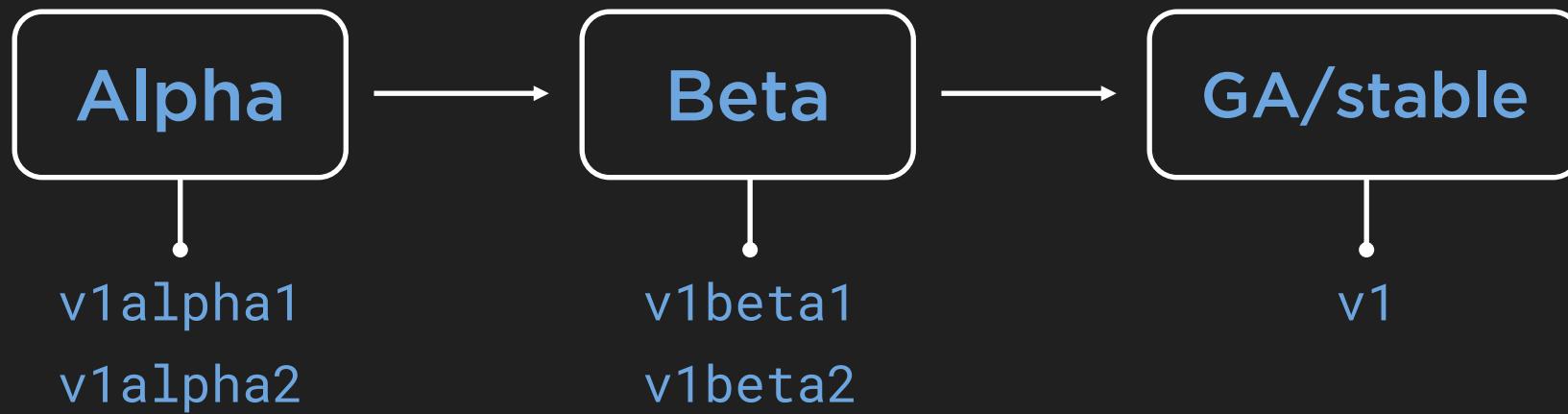
Intermediate · ★★★★★ (529)

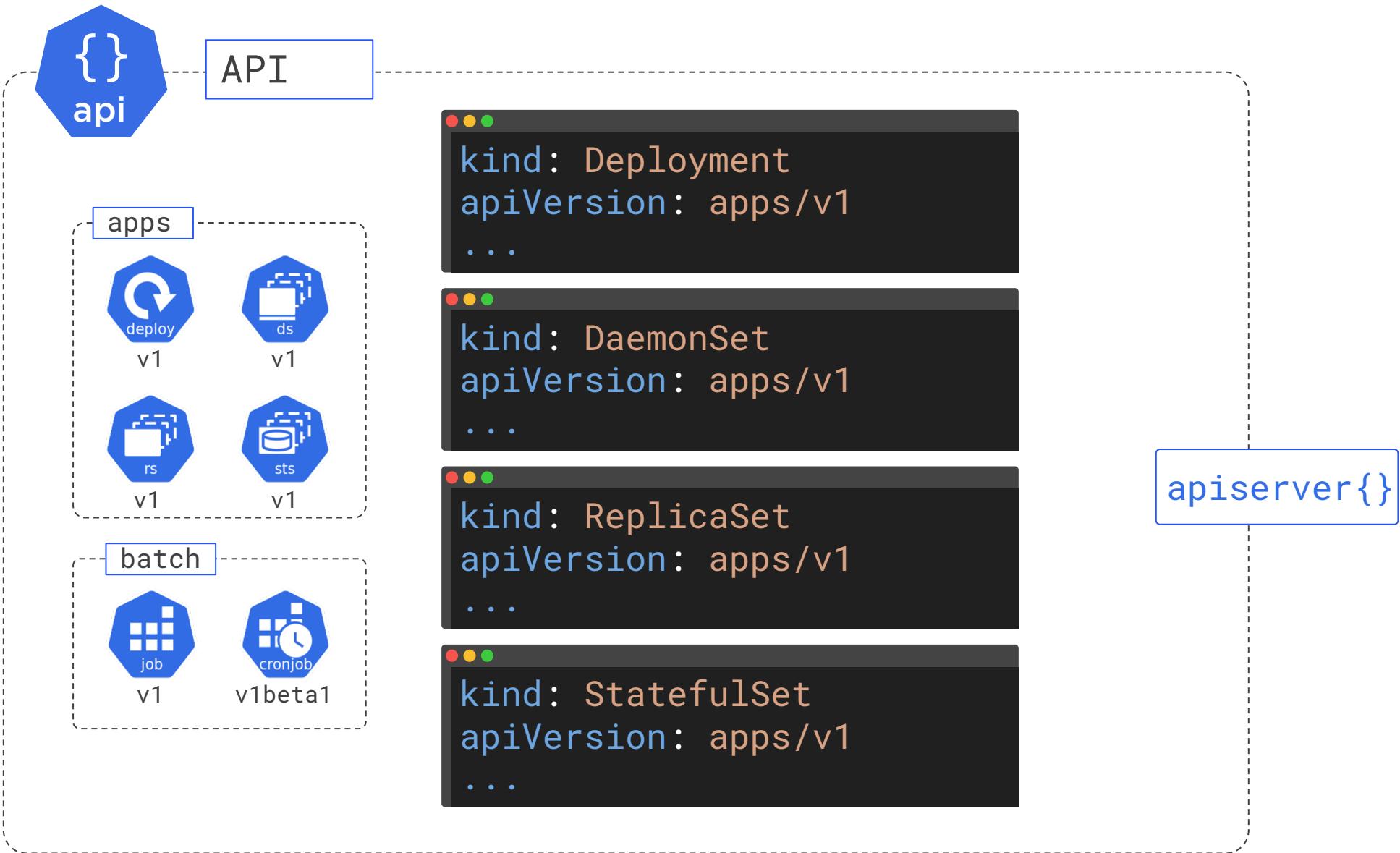
Getting Started with Docker

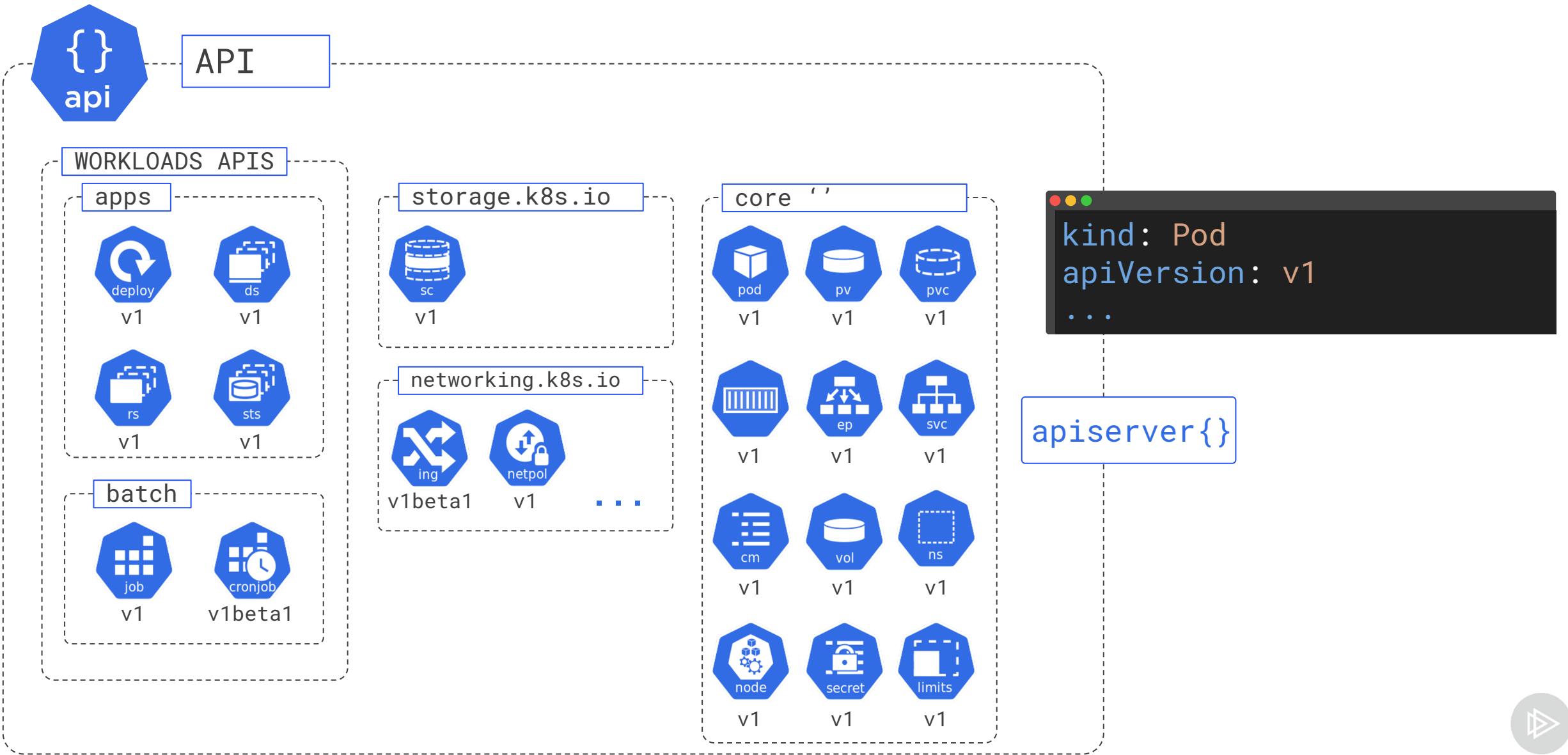
By Nigel Poulton

Beginner · ★★★★★ (1103)











API



Pod object definition

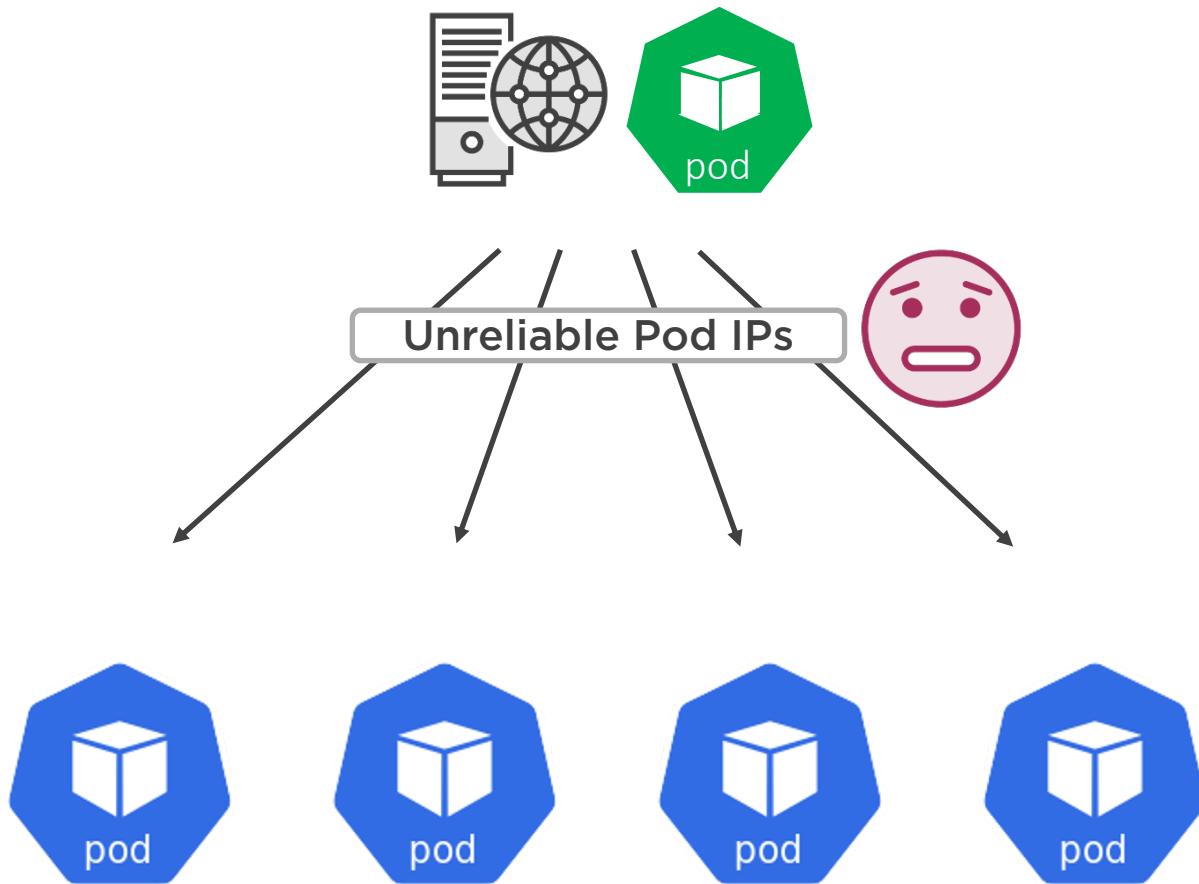
- API sub-group: [core ""]
- metadata (ObjectMeta)
 - name (string)
 - annotations (object)
 - labels (object)
- spec (PodSpec)
 - nodeName (integer)
 - containers (container array)
 - name (string)
 - image (string)
 - imagePullPolicy (string)
 - ports (ContainerPort array)
 - ...

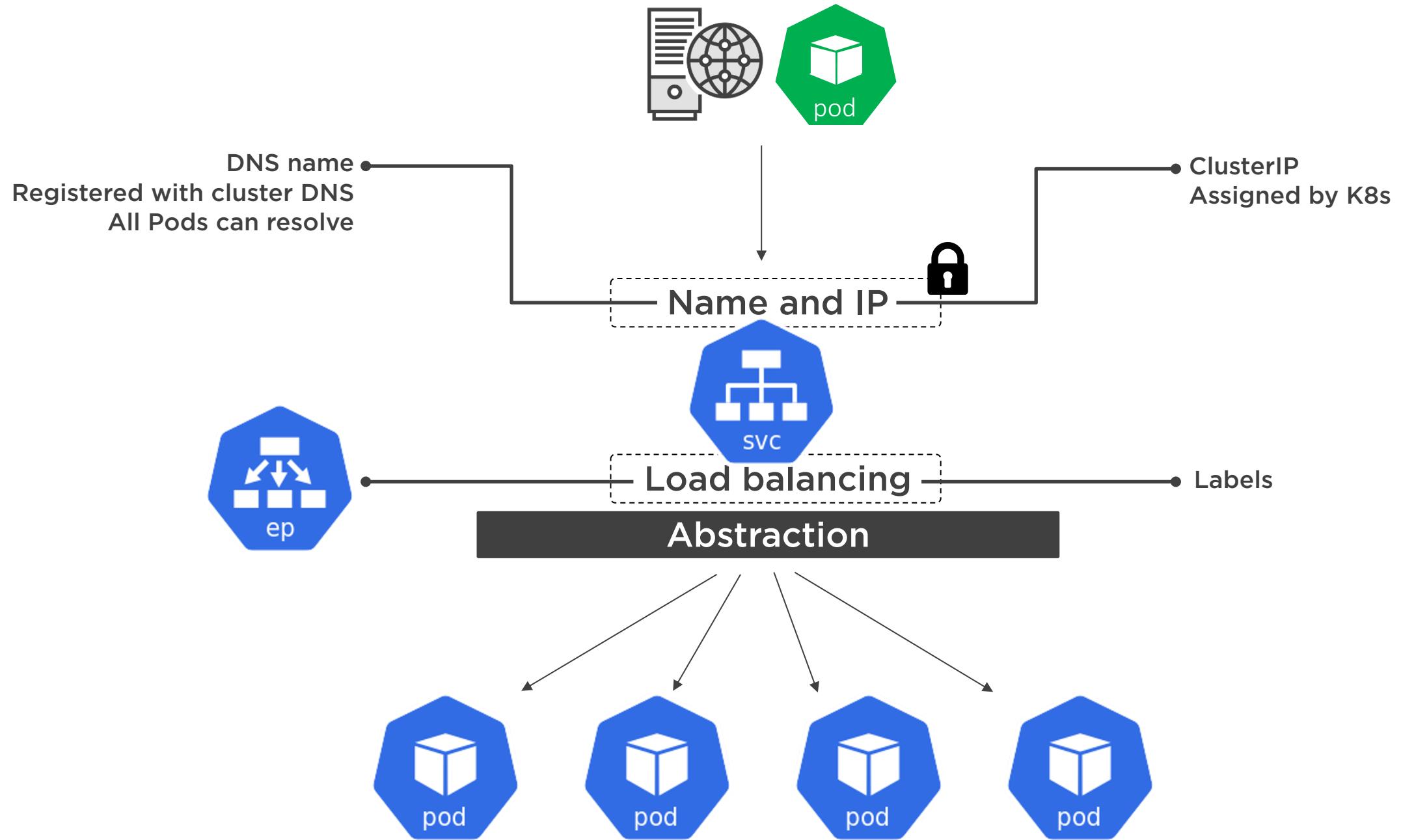
Pod manifest

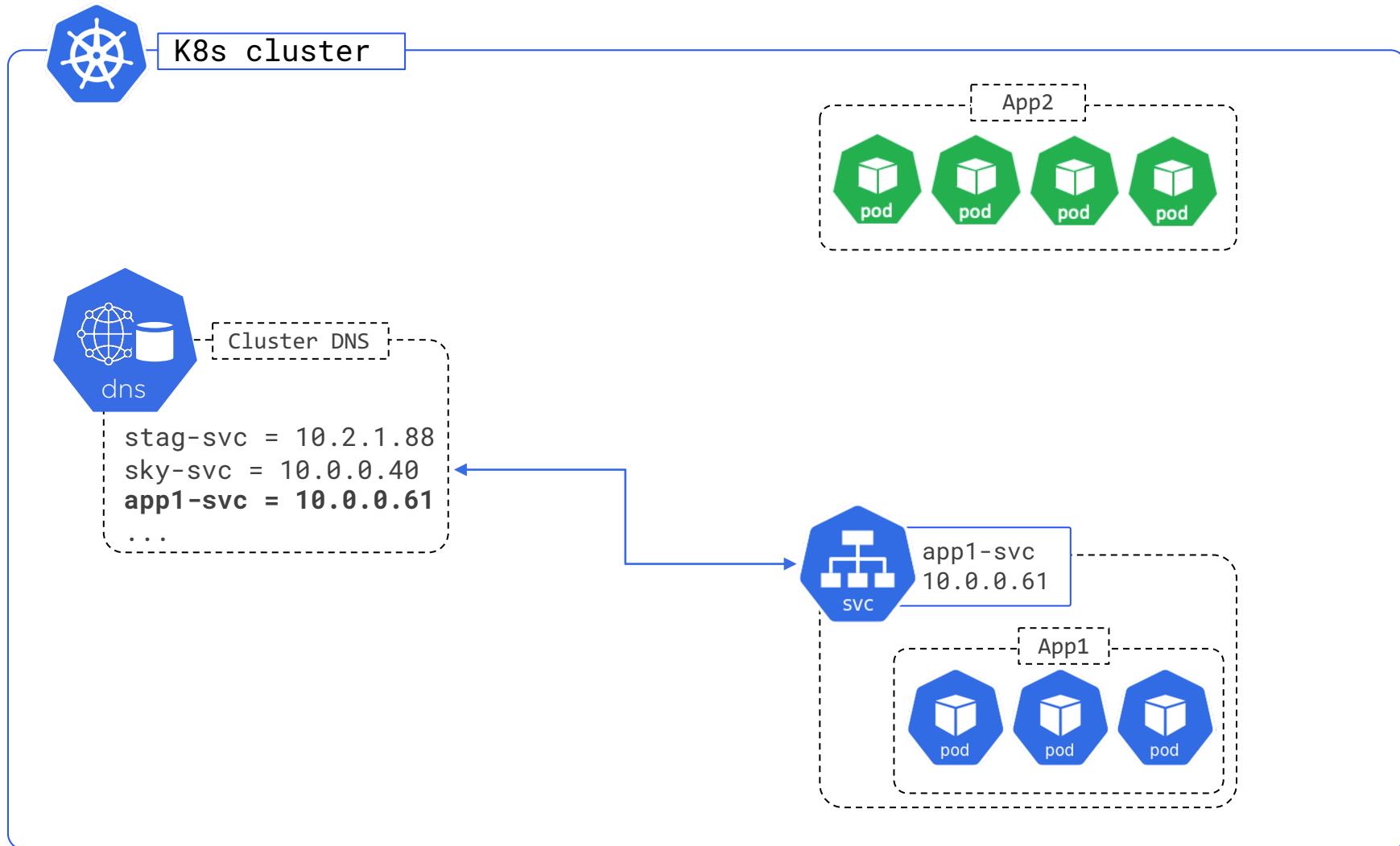
```
kind: Pod
apiVersion: v1
metadata:
  name: test-pod
  labels:
    ver: 1.0
spec:
  containers:
  - name: main
    image: web-server:1.0
    imagePullPolicy: Always
  ports:
    containerPort: 8080
```

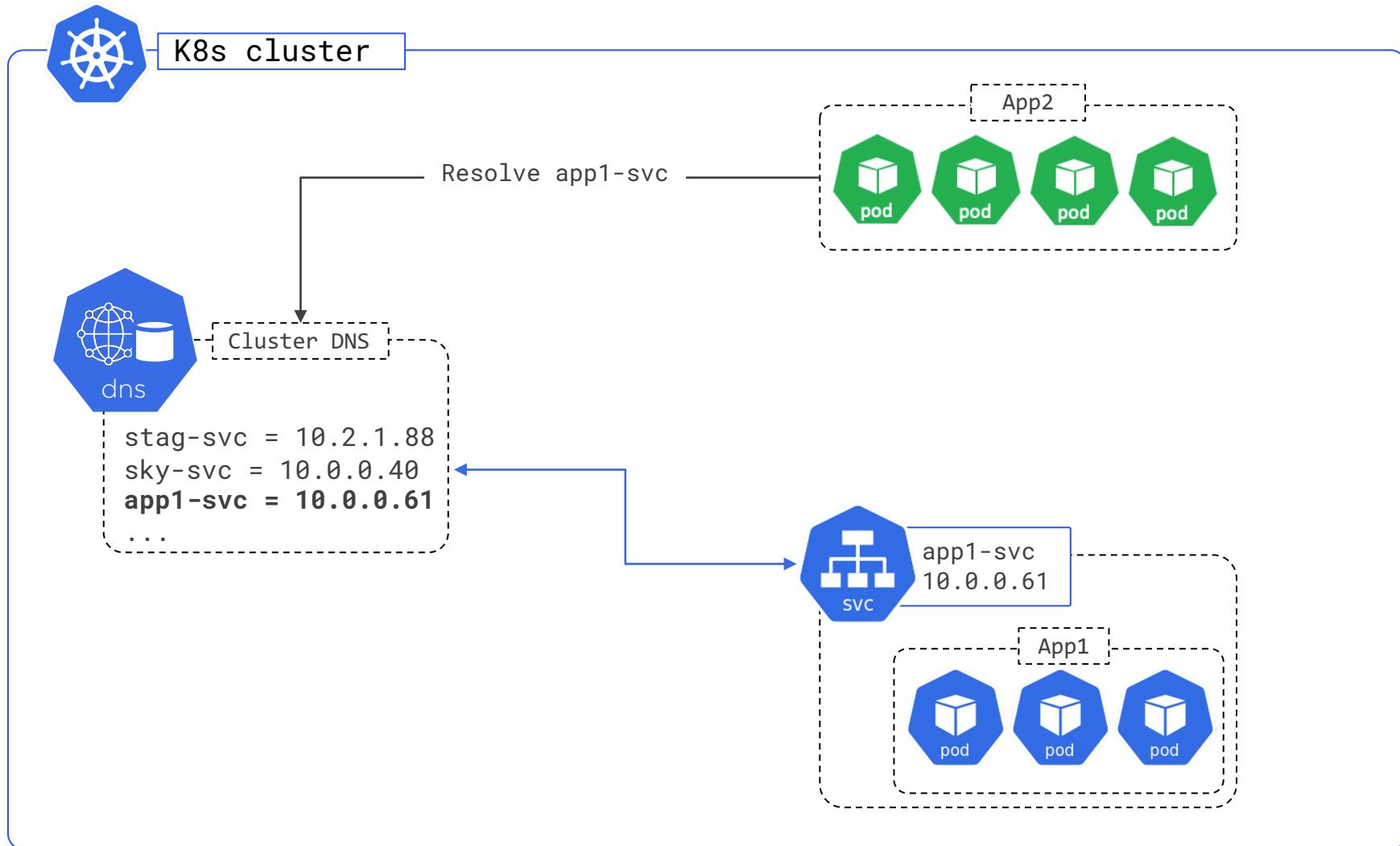
apiserver{}

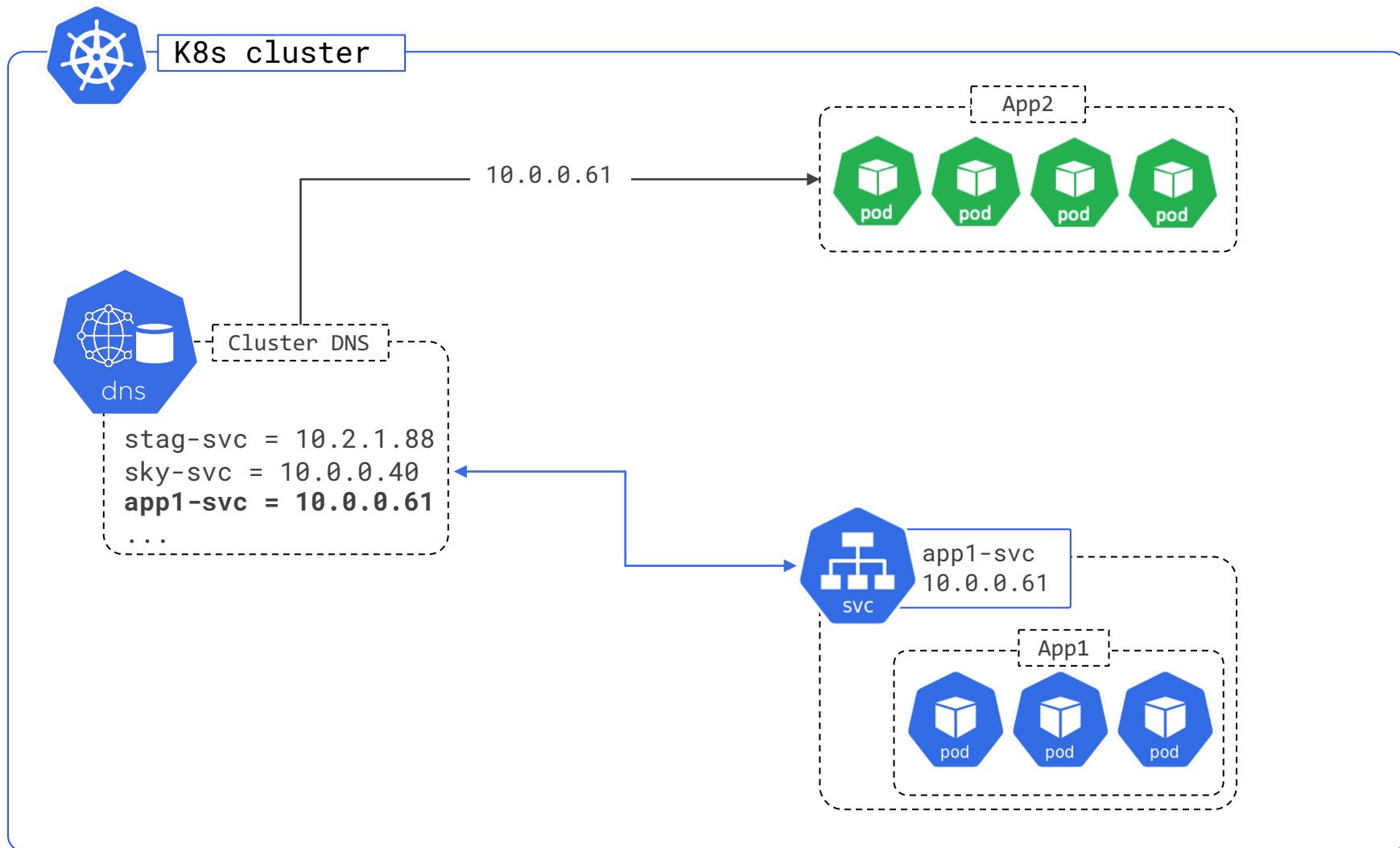


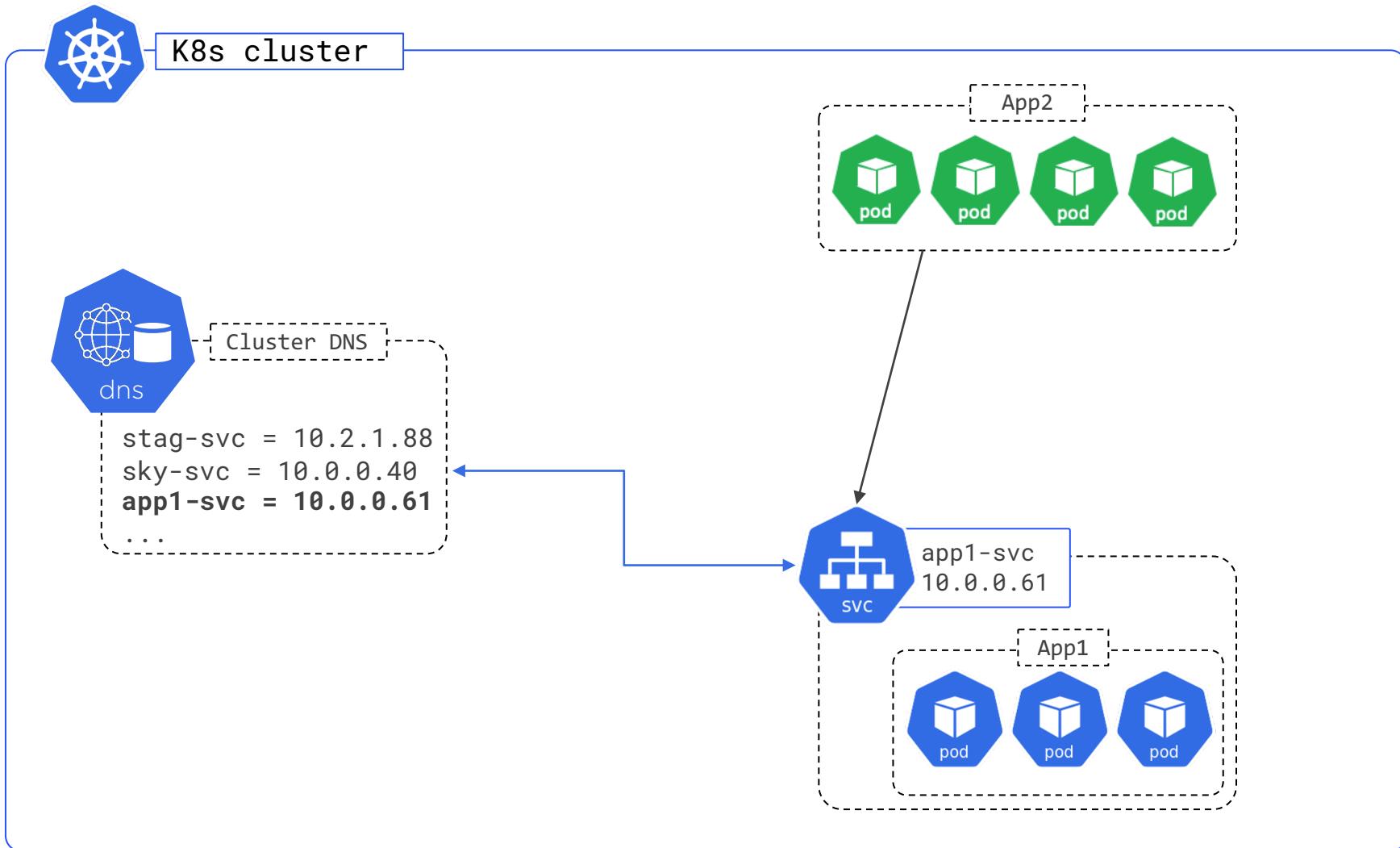




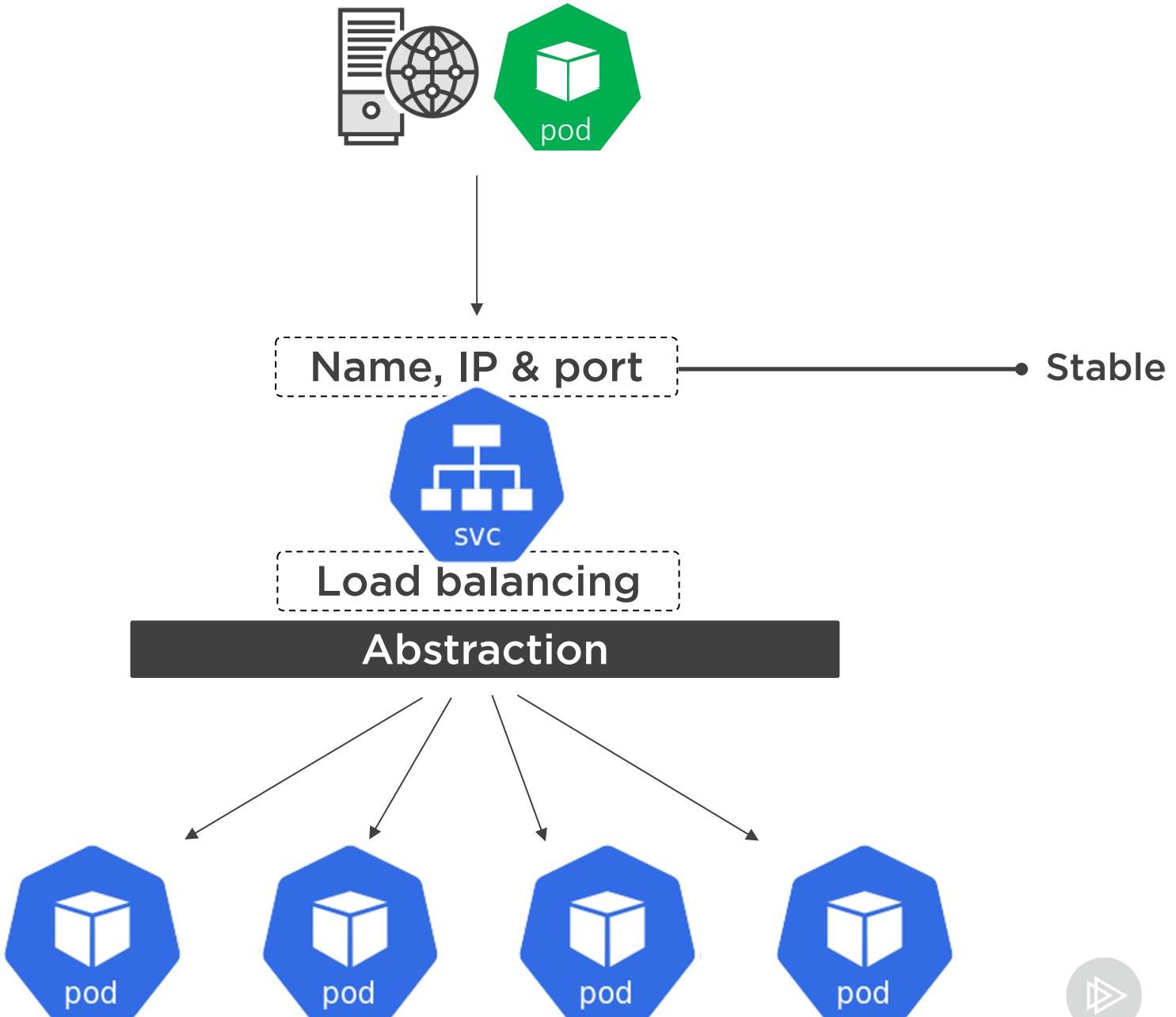




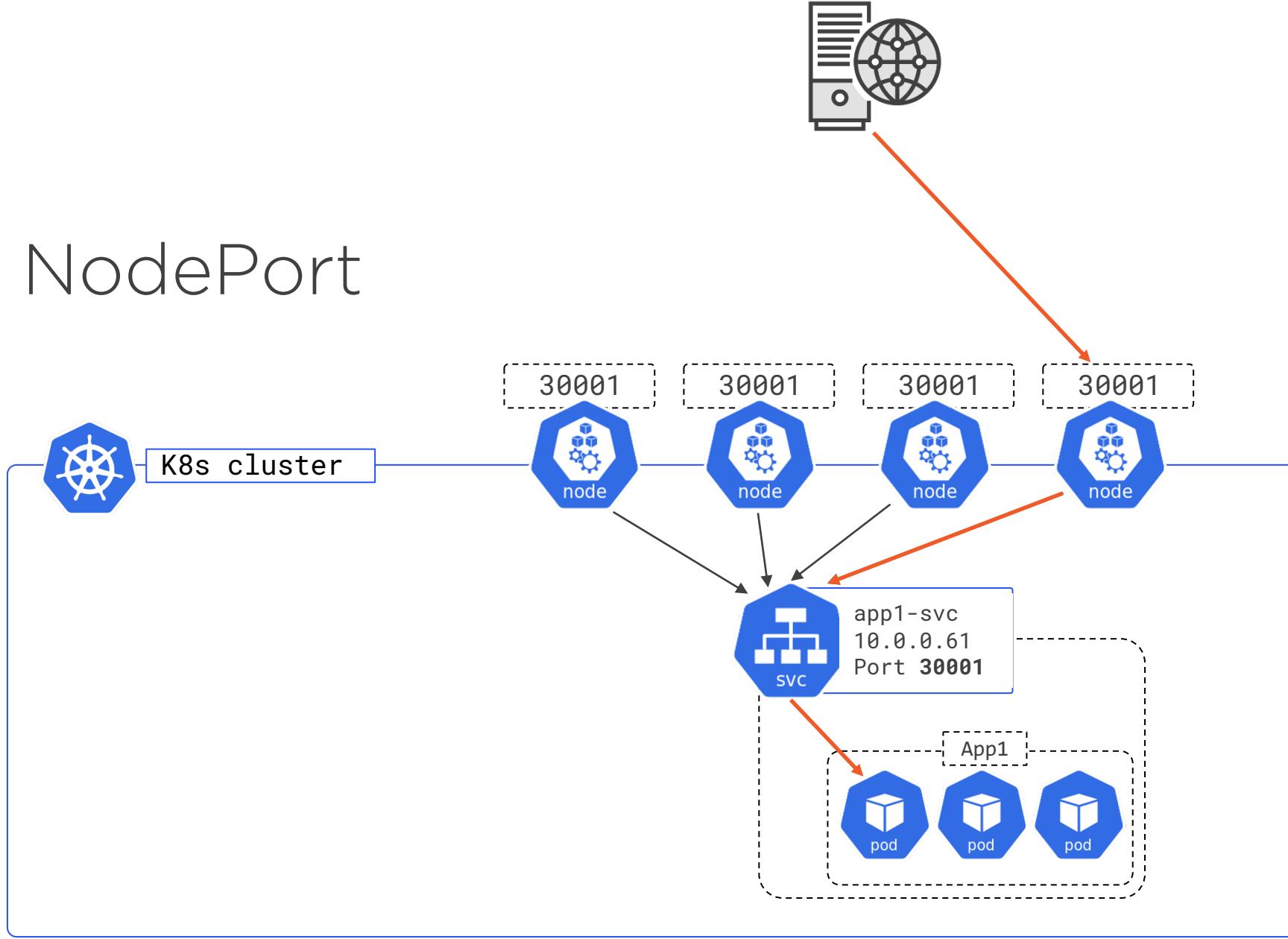




- LoadBalancer**
External access via cloud load-balancer
- NodePort**
External access via nodes
- ClusterIP (default)**
Internal cluster connectivity

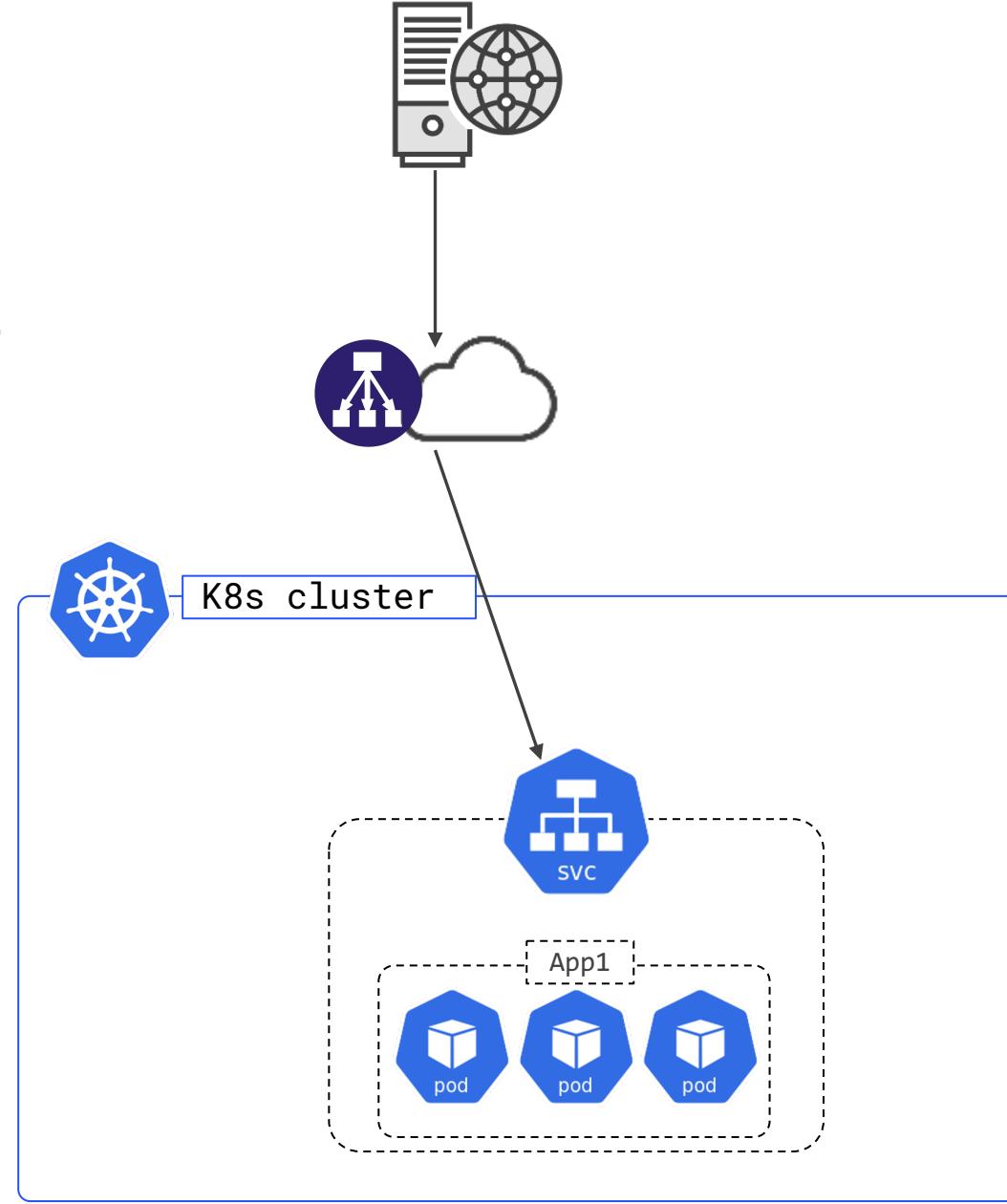


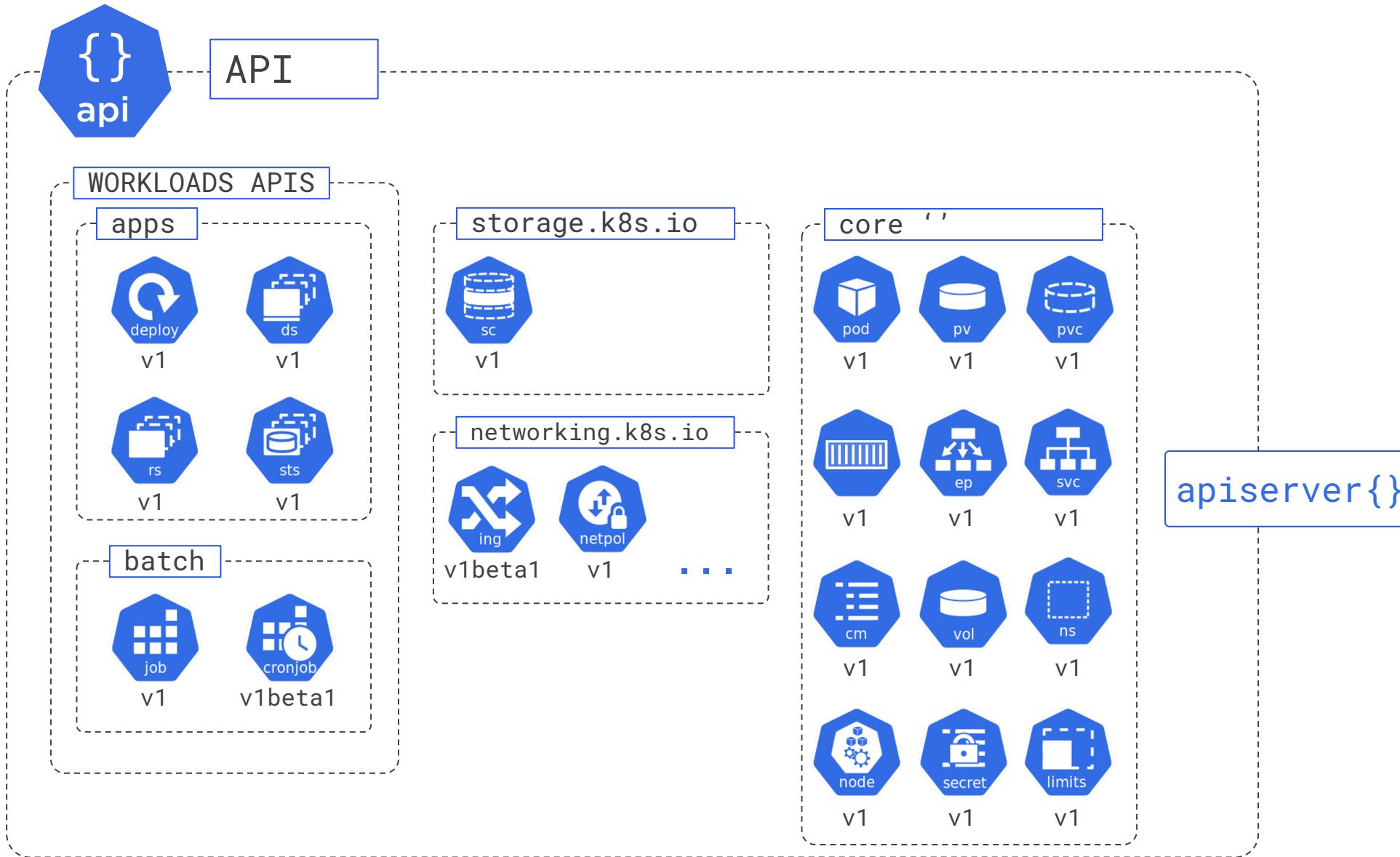
NodePort

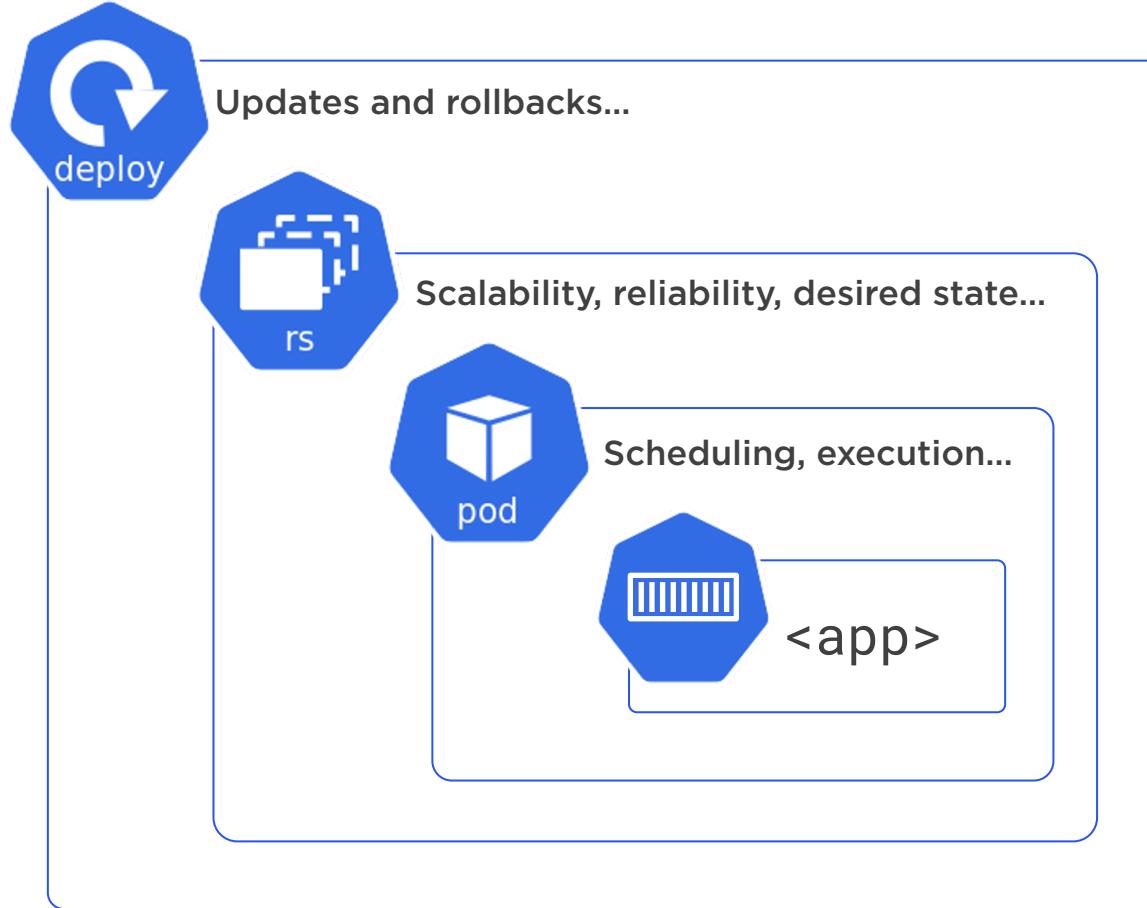


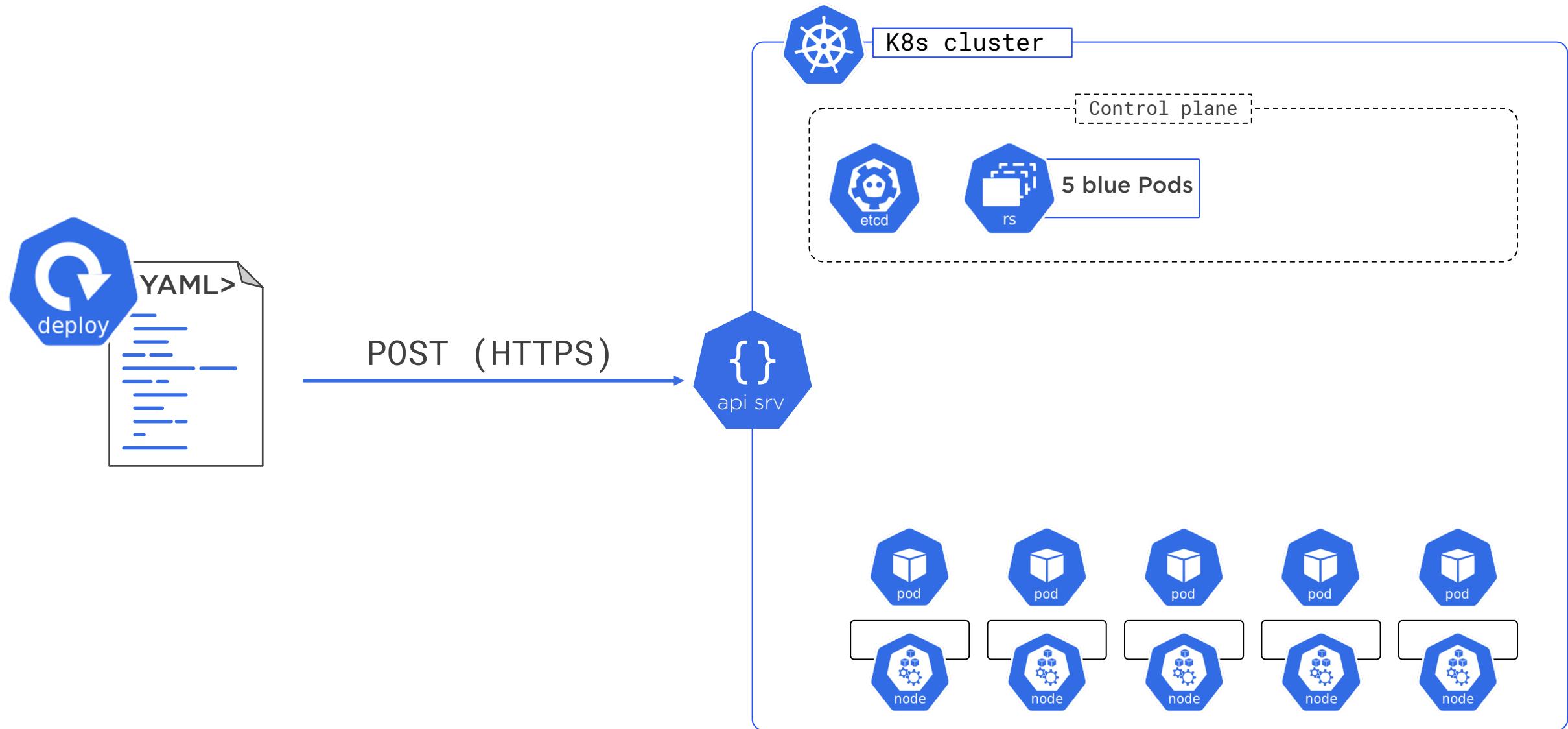
LoadBalancer

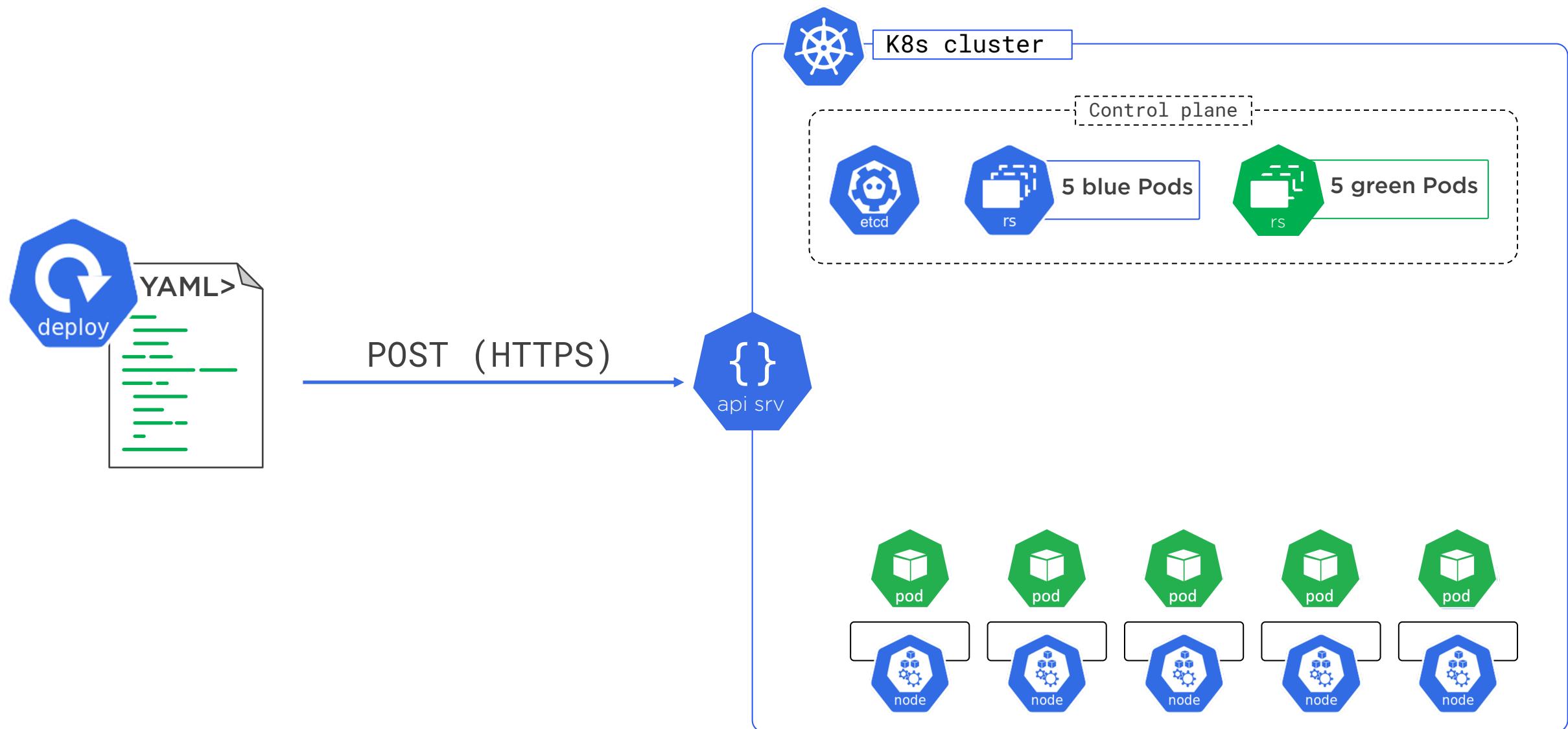
```
apiVersion: v1
kind: Service
metadata:
  name: ps-lb
spec:
  type: LoadBalancer
ports:
- port: 80
  targetPort: 8080
selector:
  app: web
```

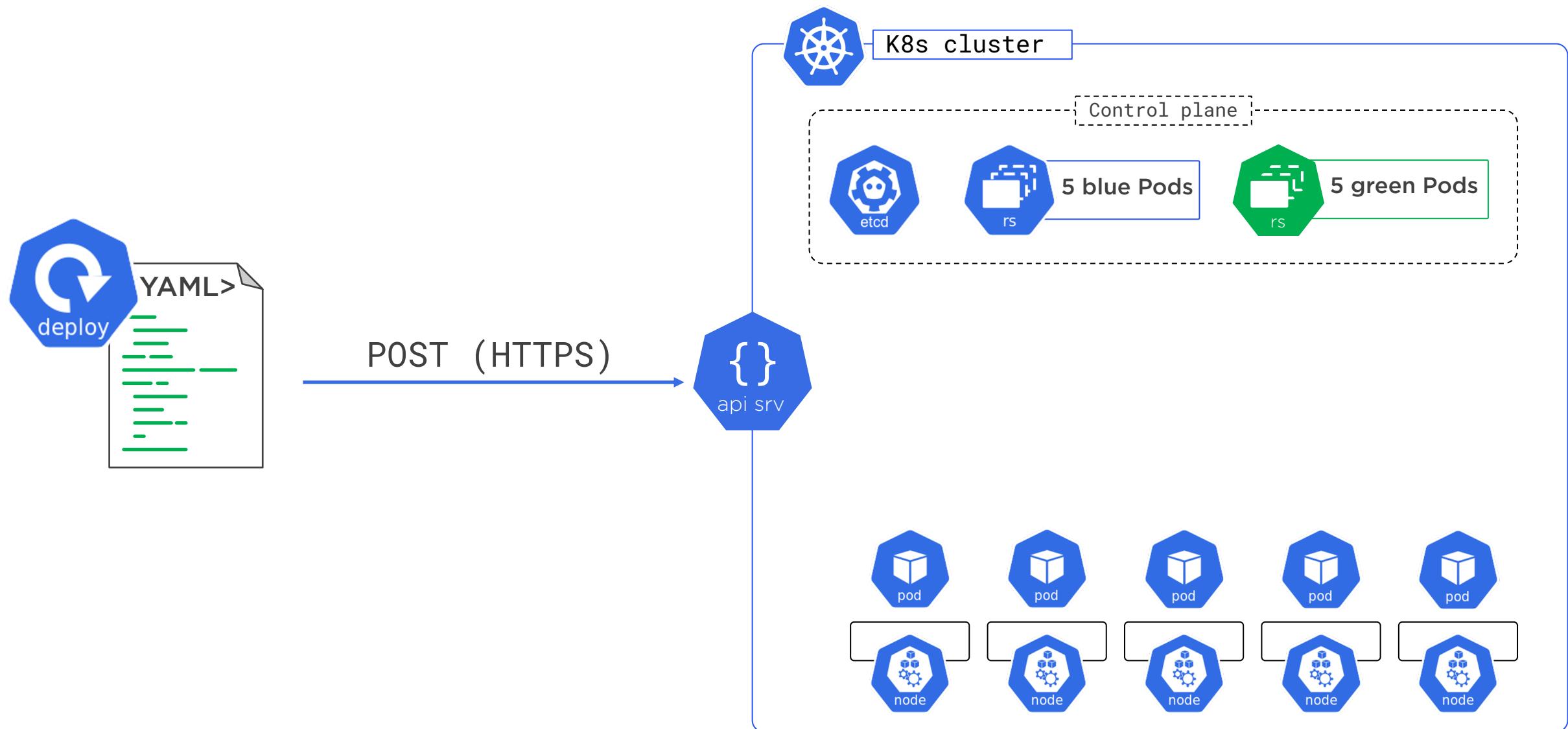


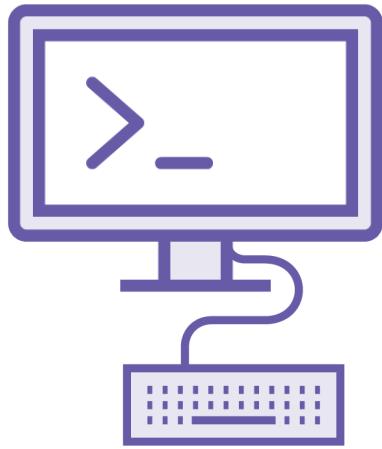








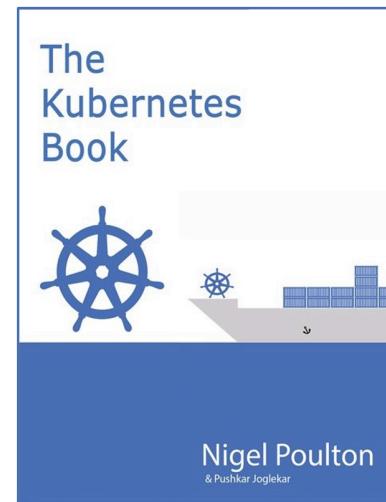




Feel free to
make mistakes!



K8s courses and
learning paths



Also available
on Audible

meetup



KubeCon

Come and say
Hi!



@nigelpoulton

