

Getting Started with Kubernetes

COURSE OVERVIEW



Nigel Poulton

@nigelpoulton www.nigelpoulton.com



Course:
“Docker and Kubernetes: The Big Picture”

Fundamentals and hands-on with containers



Overview



What is Kubernetes

Kubernetes Architecture

Getting Kubernetes

Working with Pods

Kubernetes Networking with Services

Kubernetes Deployments

What Next



Code and examples...

<https://github.com/nigelpoulton/getting-started-k8s>





Playback Speed!

Not too fast in case you stop
learning ;-)



@nigelpoulton



Up Next:
What is Kubernetes



What is Kubernetes



Nigel Poulton

@nigelpoulton www.nigelpoulton.com



Agenda

Kubernetes History

Kubernetes Primer

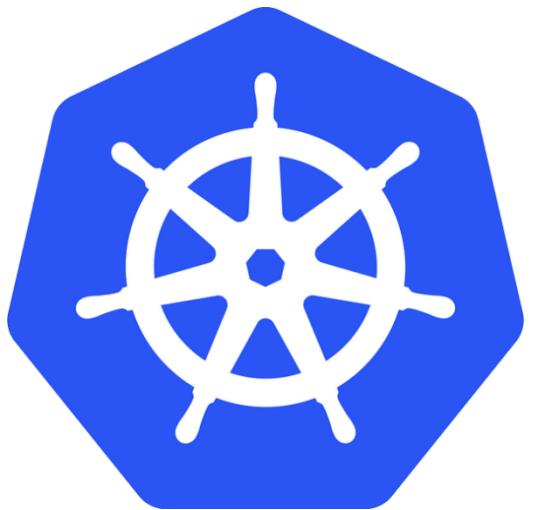


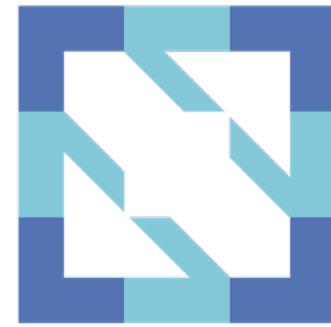
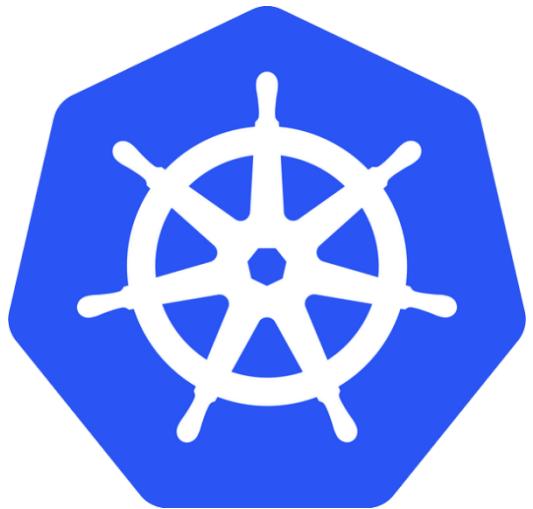
G





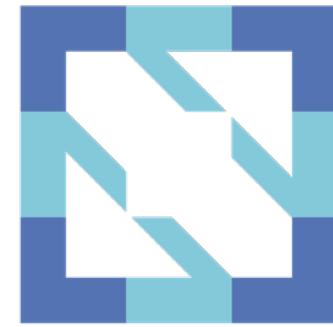
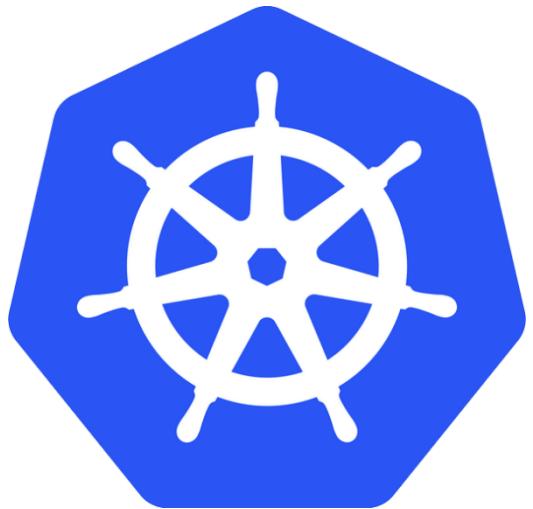






**CLOUD NATIVE
COMPUTING FOUNDATION**





**CLOUD NATIVE
COMPUTING FOUNDATION**







github.com/kubernetes/kubernetes





github.com/kubernetes/kubernetes





 @kubernetesio

 slack.k8s.io

 meetup

 @nigelpoulton





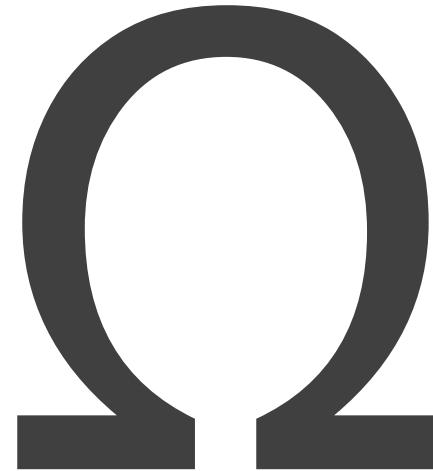
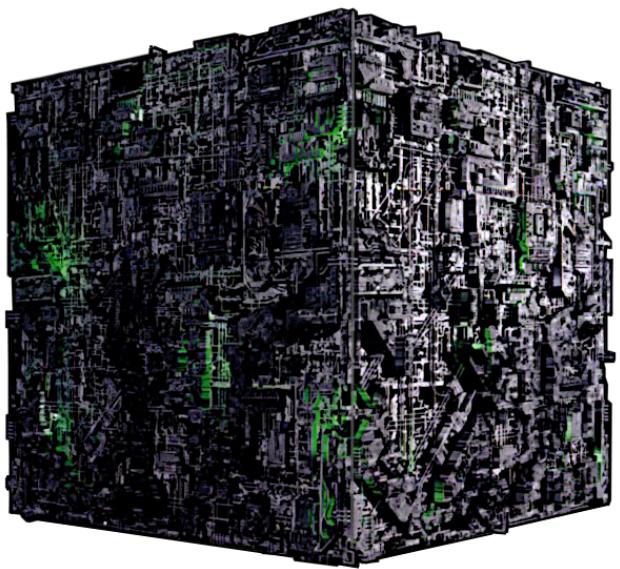
 @kubernetesio

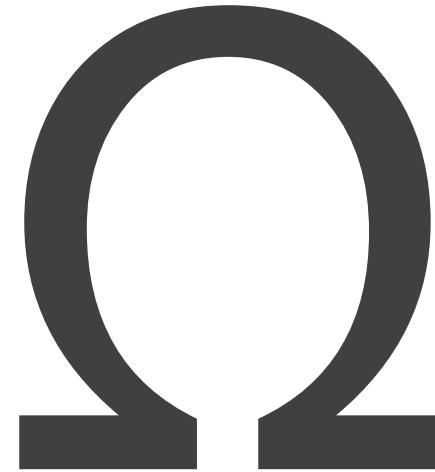
 slack.k8s.io

 meetup

 @nigelpoulton





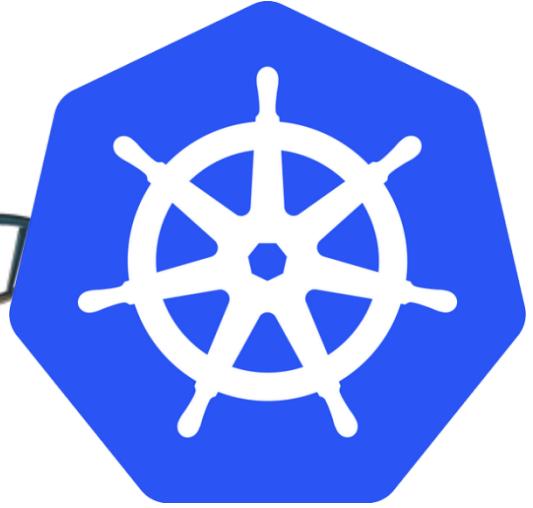




Borg
(Proprietary)



Omega
(Proprietary)



Kubernetes
(open-source)

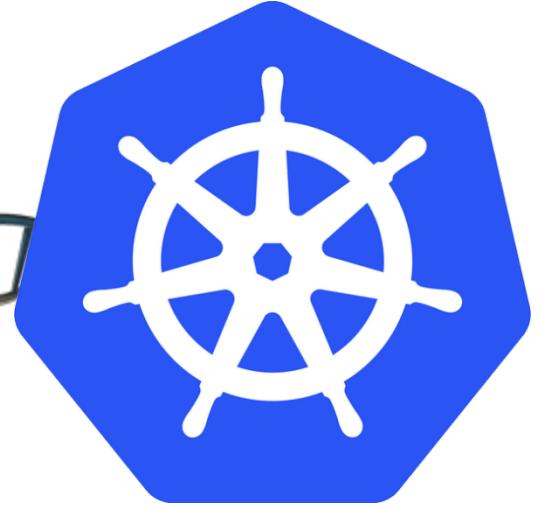




Borg
(Proprietary)



Omega
(Proprietary)



Kubernetes
(open-source)

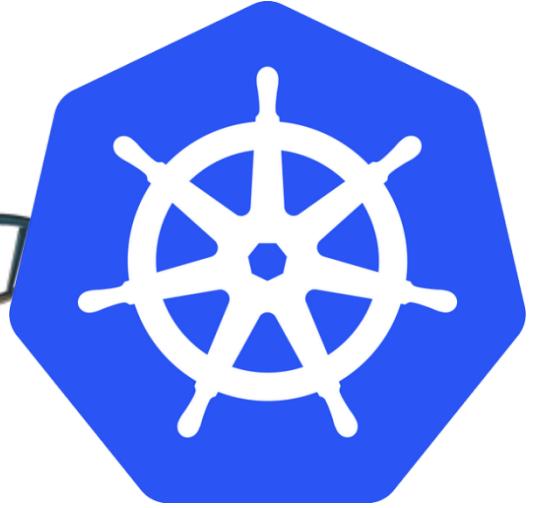




Borg
(Proprietary)



Omega
(Proprietary)



Kubernetes
(open-source)



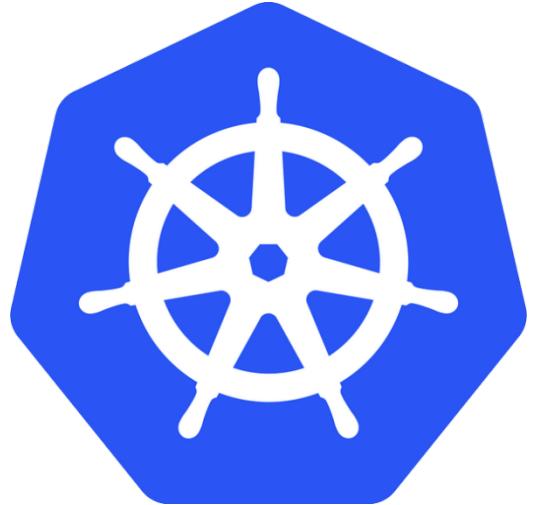




Version 1 in July 2015

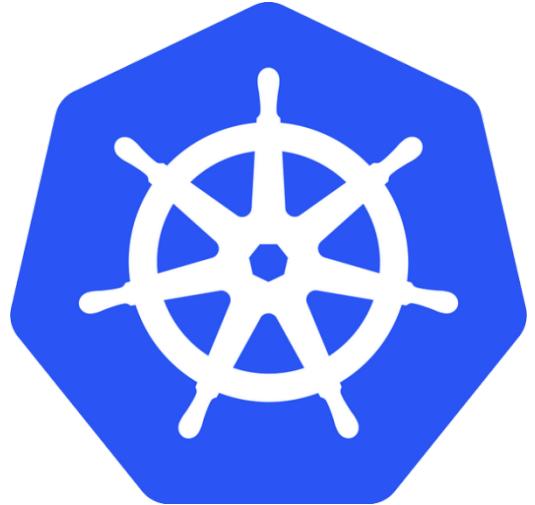
Increasingly stable and mature





Kubernetes





Kubernetes

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

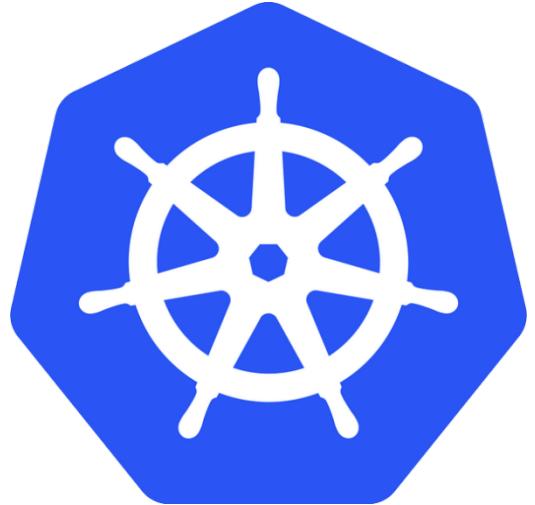




Kubernetes

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





Seven of Nine





Seven of Nine

Borg drone rescued by USS Voyager under the command of Captain Kathryn Janeway.

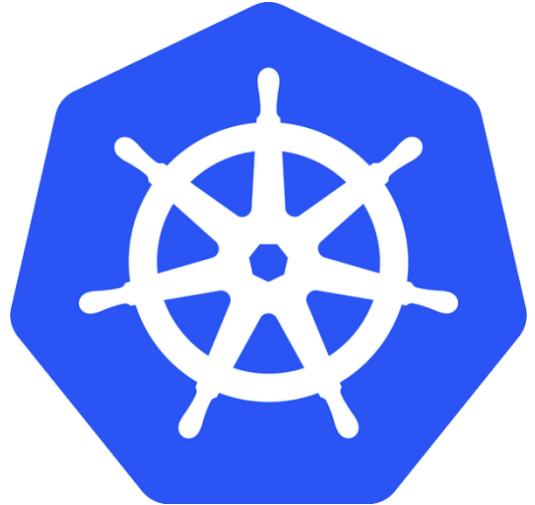




Seven of Nine

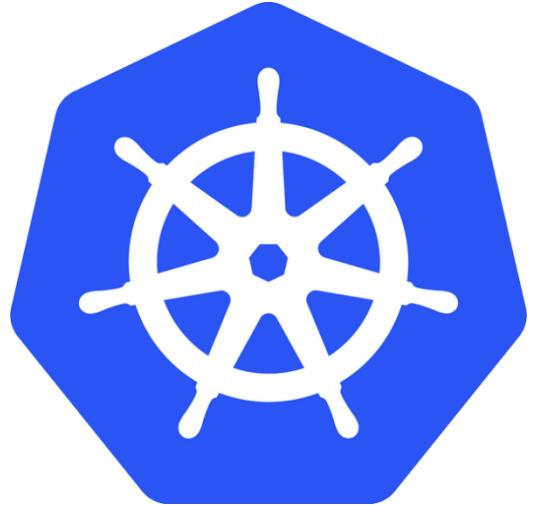
Borg drone rescued by USS Voyager under the command of Captain Kathryn Janeway.





Kubernetes





Kubernetes





K8s





Kubernetes
K8s





Kubernetes
K8s



Up Next:
Kubernetes: What and Why



Kubernetes: What and Why



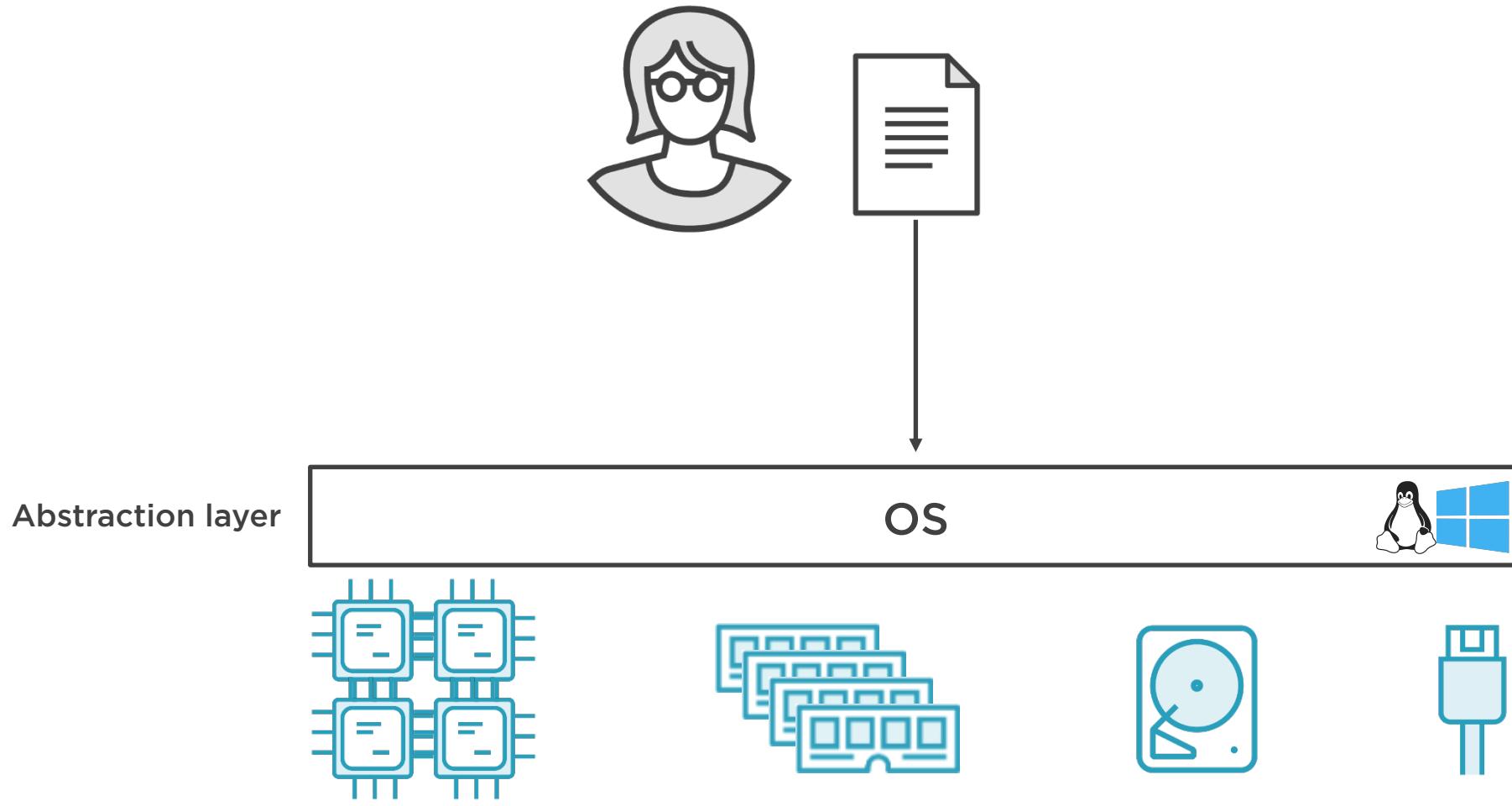
Docker and Kubernetes: The Big Picture

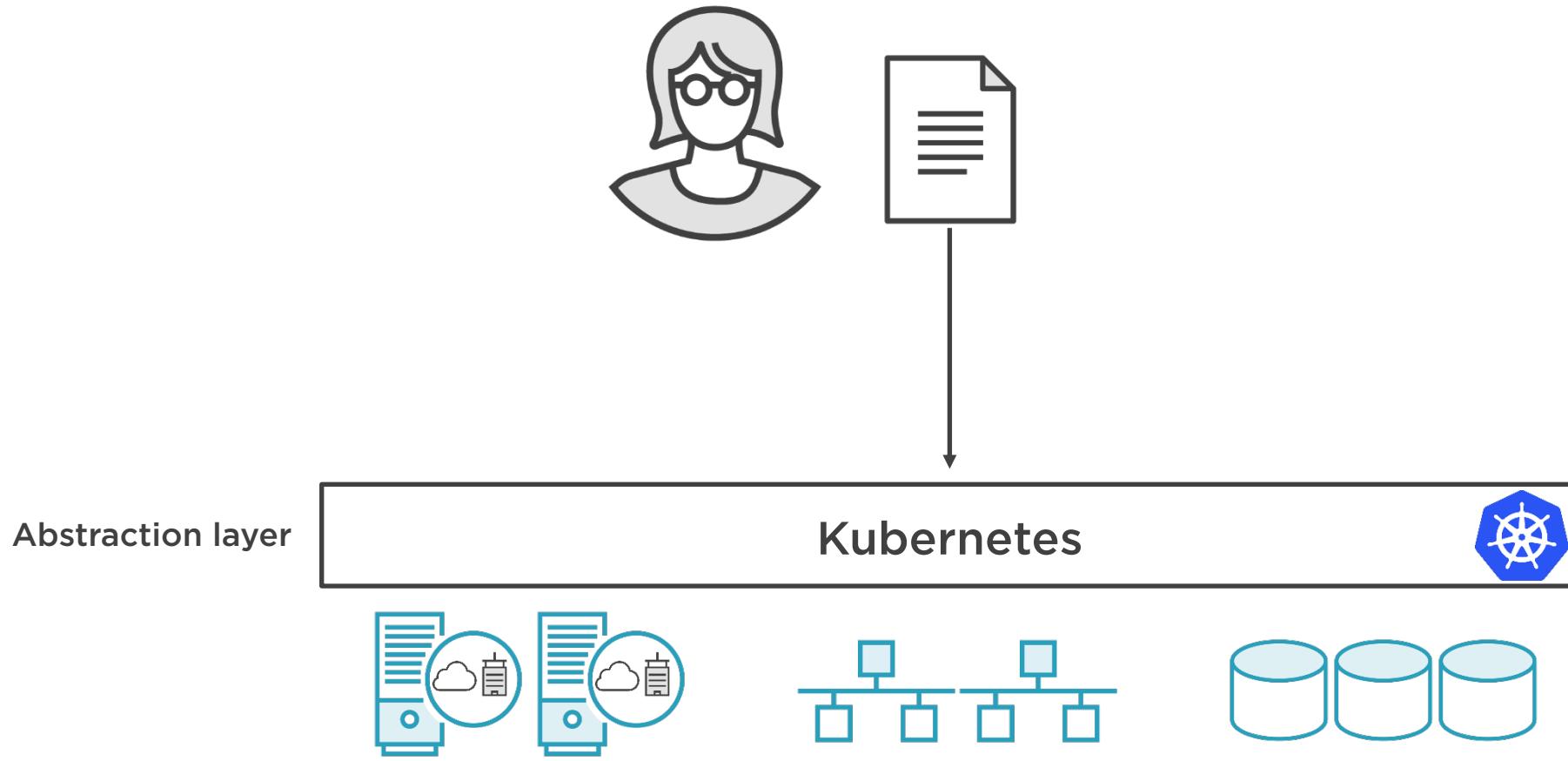


Containers and microservices bring a whole new set of management challenges...

Kubernetes to the rescue!









- Standard package format
- Manifest
- Job done!





- **Standard package format**
- **Manifest**
- **Job done!**



Up Next:
Quick Jargon Busting



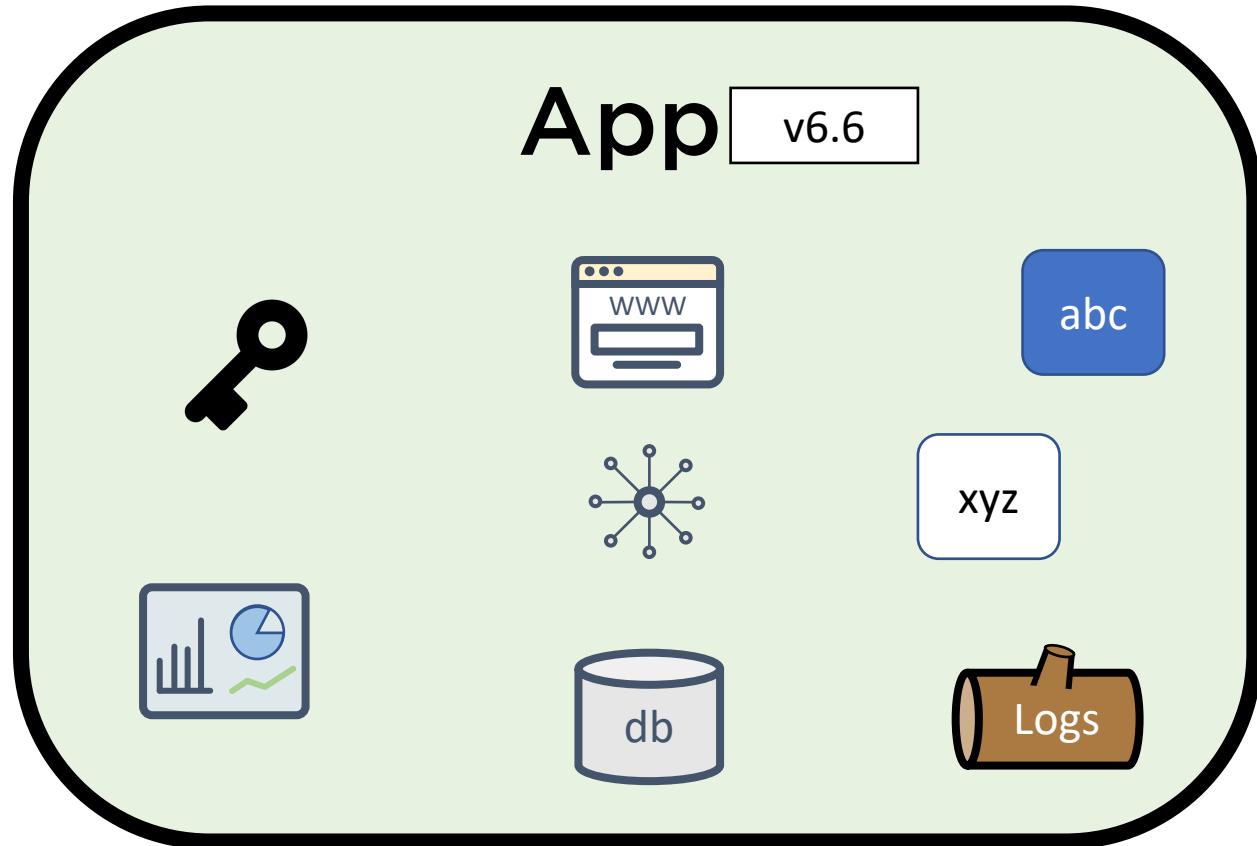
Quick Jargon Busting



monolith /'mɒn(ə)lɪθ/

(noun) Insulting word for old/legacy application





monolith /'mɒn(ə)lɪθ/

(noun) Insulting word for old/legacy application



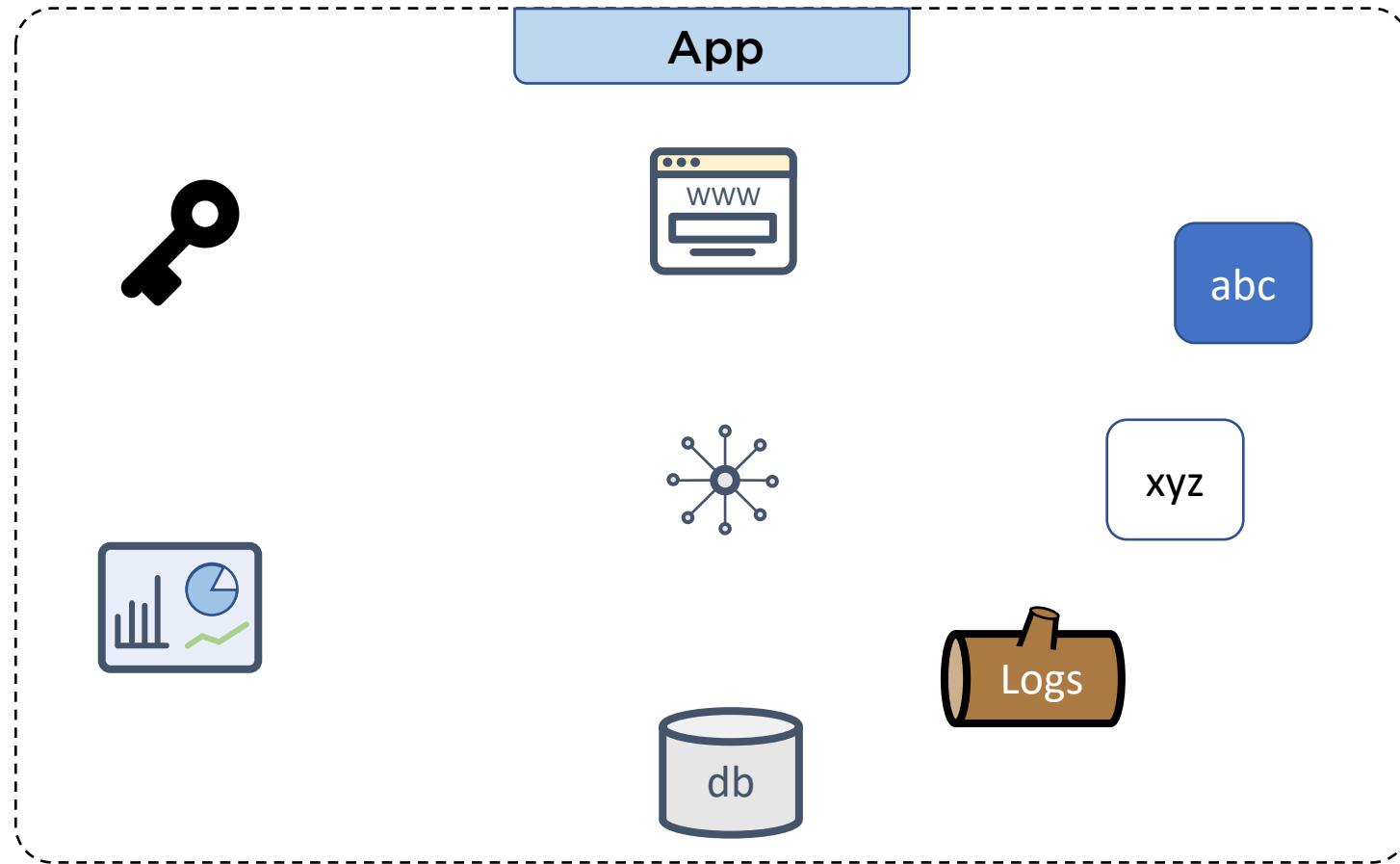


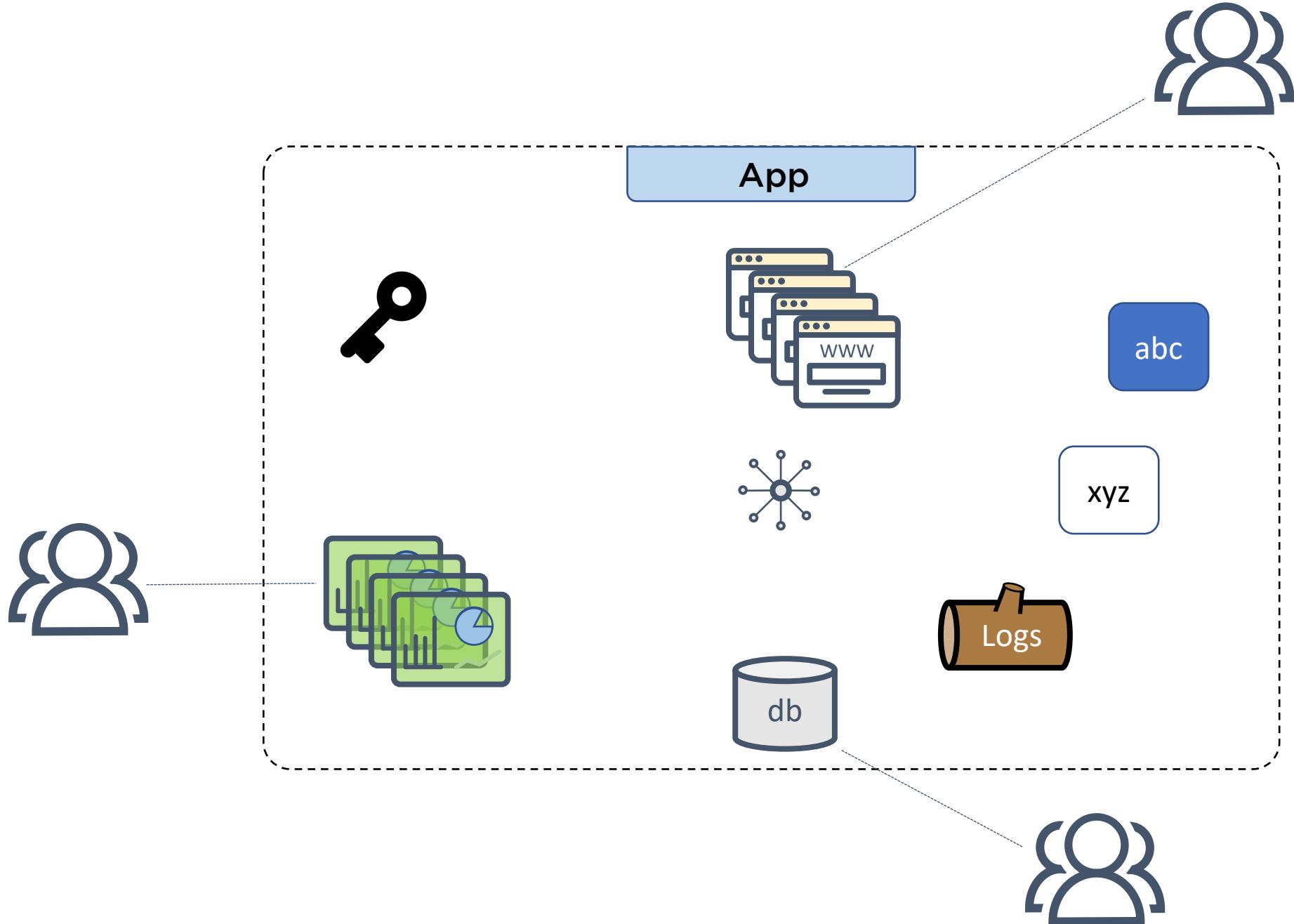
abc

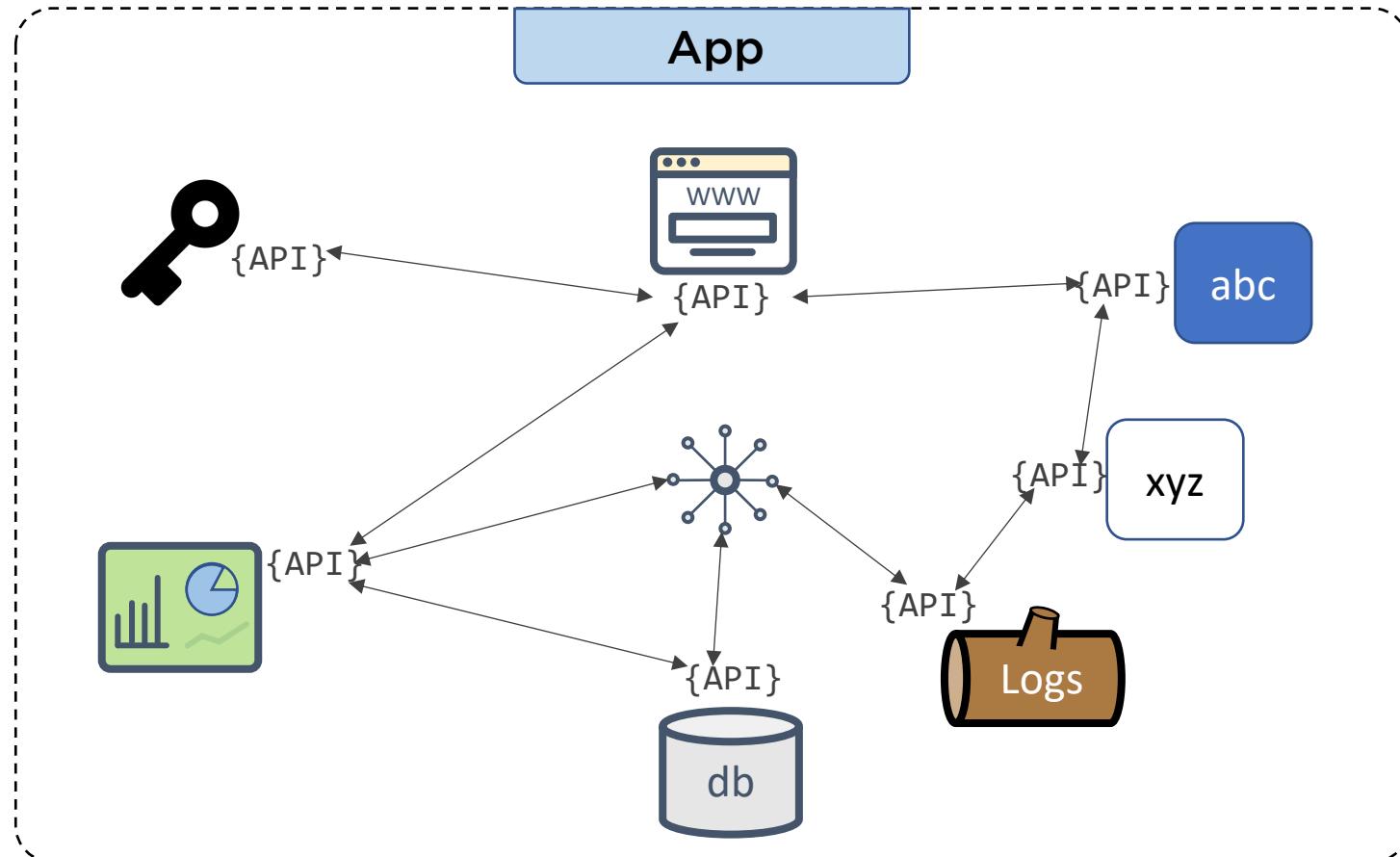


xyz







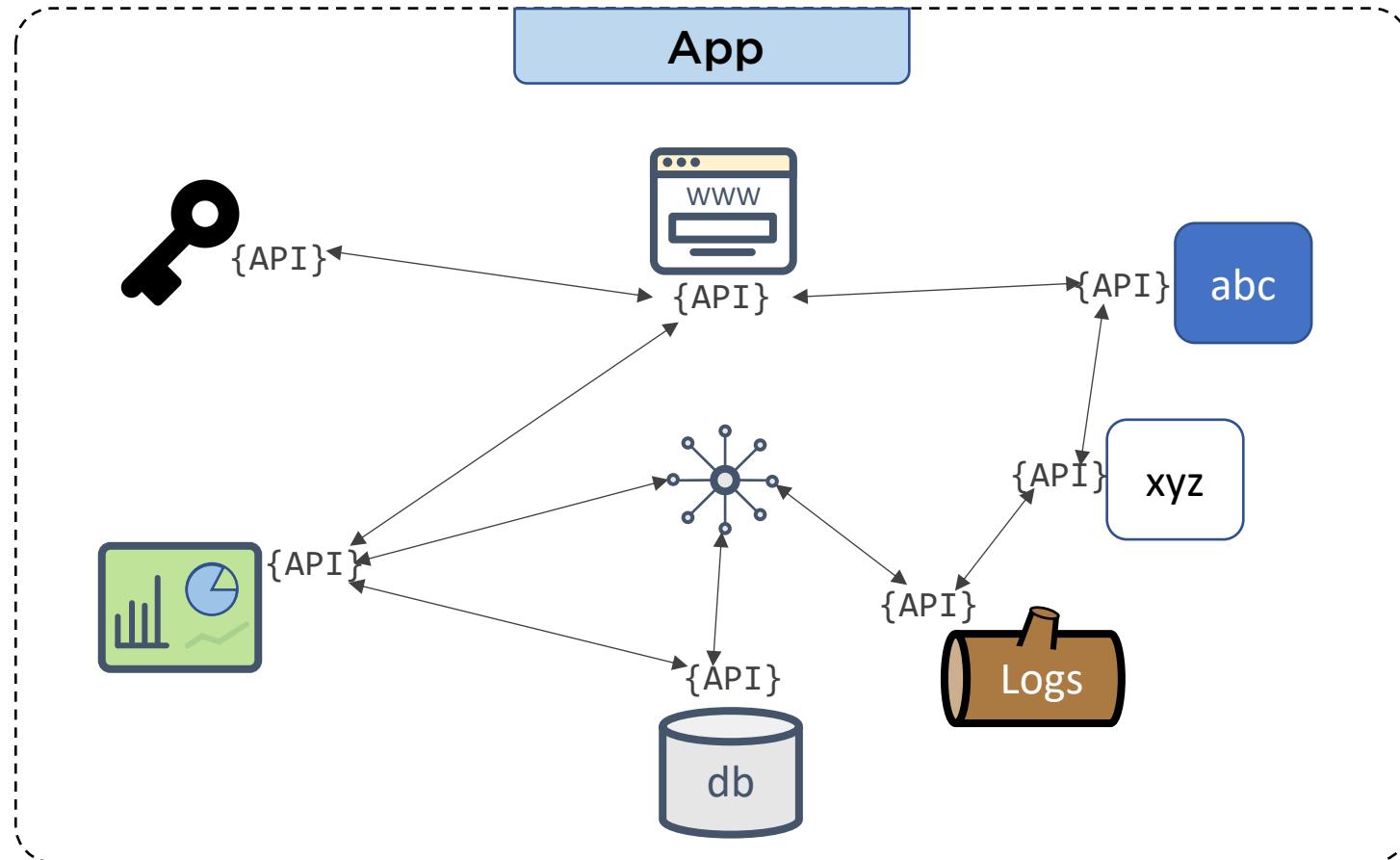


Public
cloud

Private/hybrid
cloud

On prem
Data center



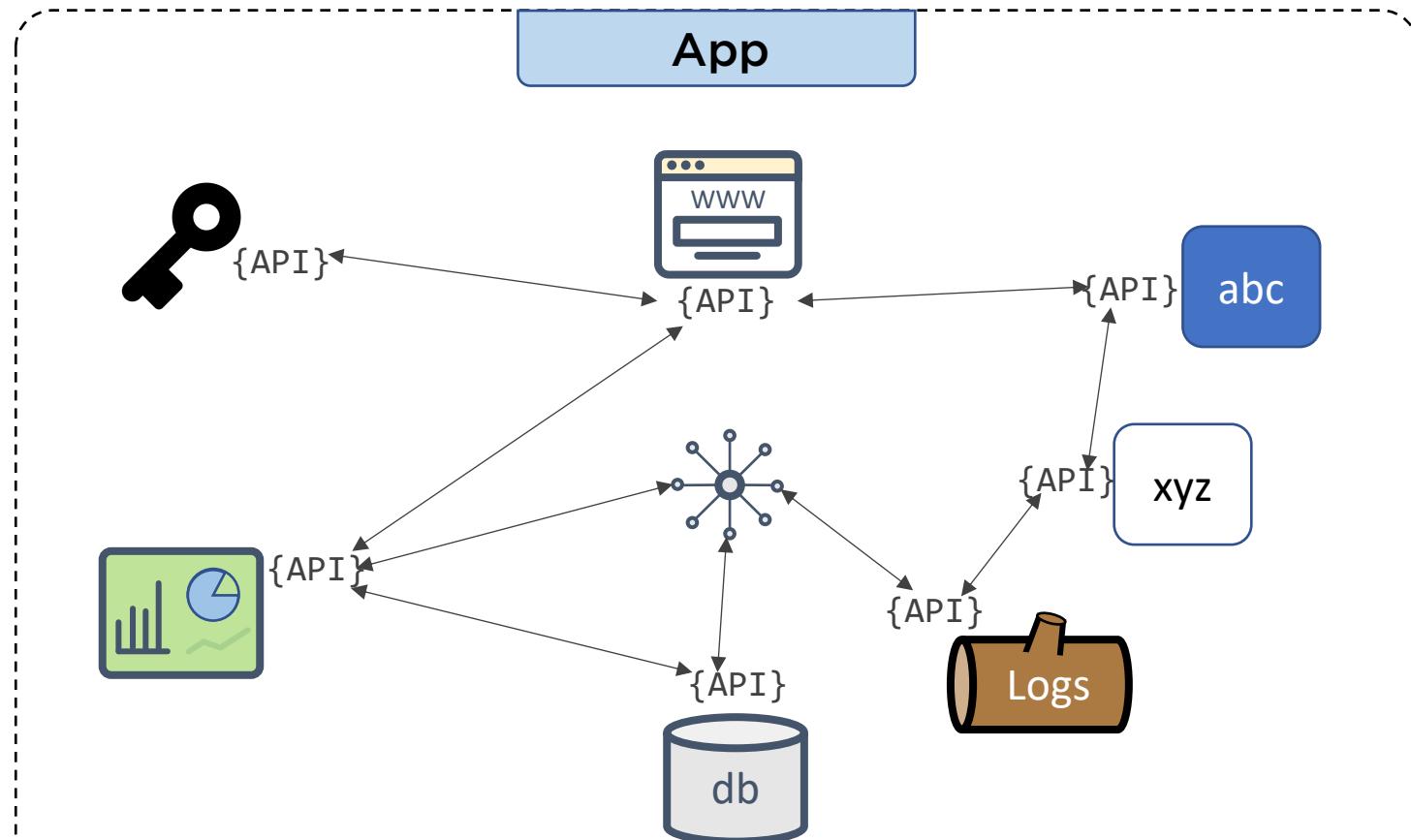


Public
cloud

Private/hybrid
cloud

On prem
Data center





Docker and Kubernetes: The Big Picture





**Kubernetes is
developing fast**



**Kubernetes is a
valuable skill**



Up Next:
Kubernetes Architecture



Kubernetes Architecture



Nigel Poulton

@nigelpoulton www.nigelpoulton.com



Overview



Big Picture View

Masters

Nodes

Pods

Services

Deployments

The API and API Server

Recap

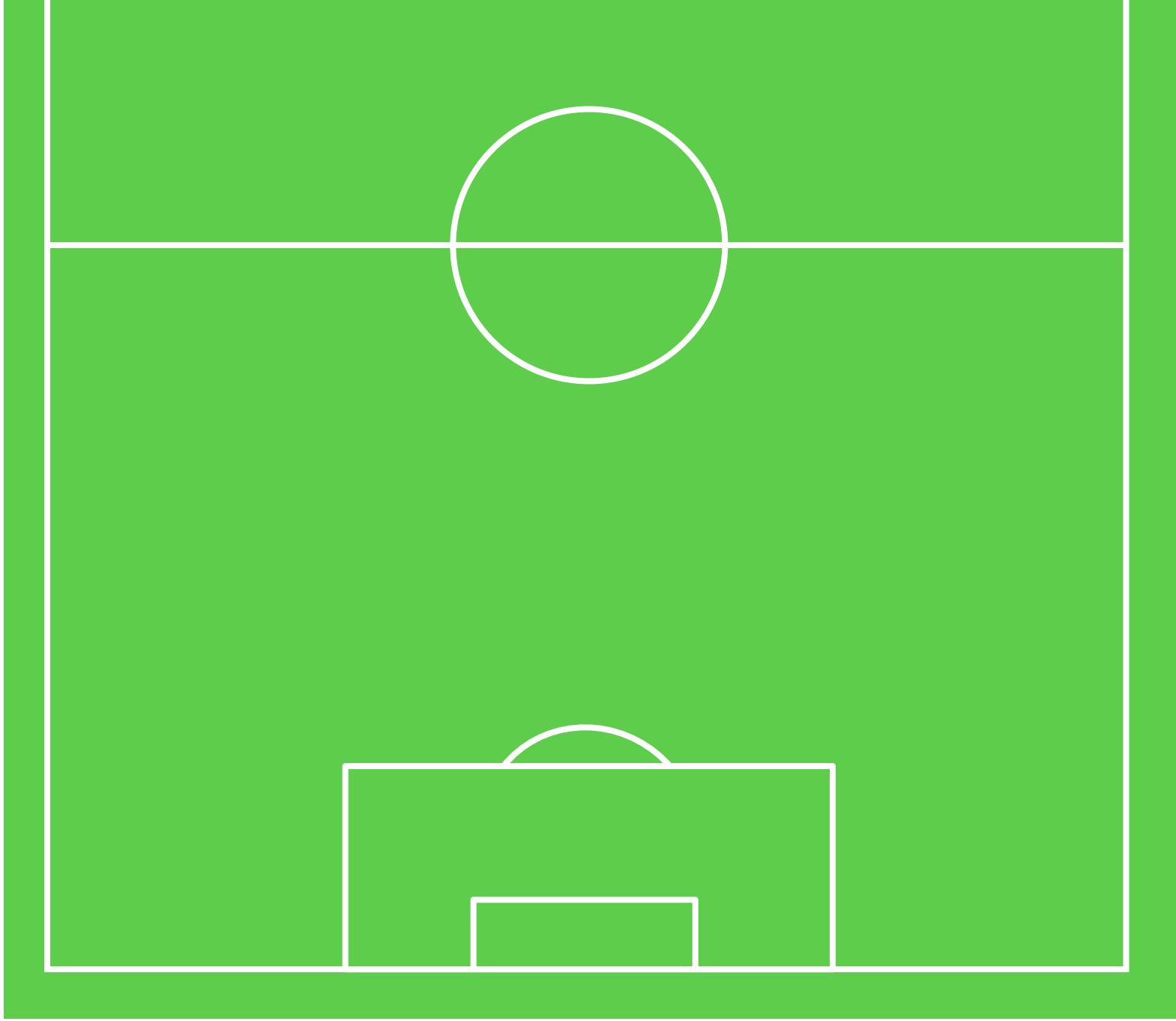


Up Next:
Kubernetes: Big Picture Overview

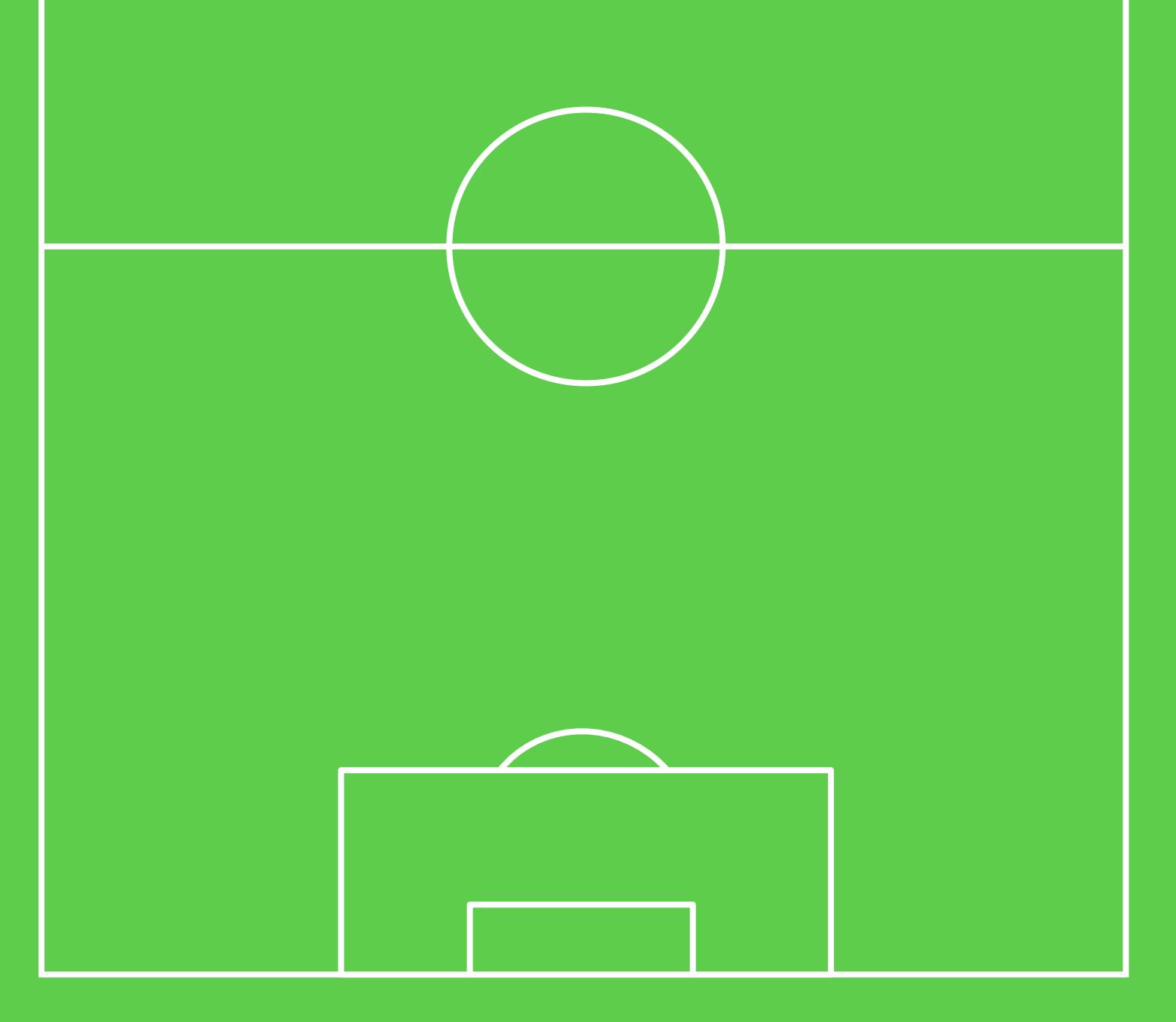
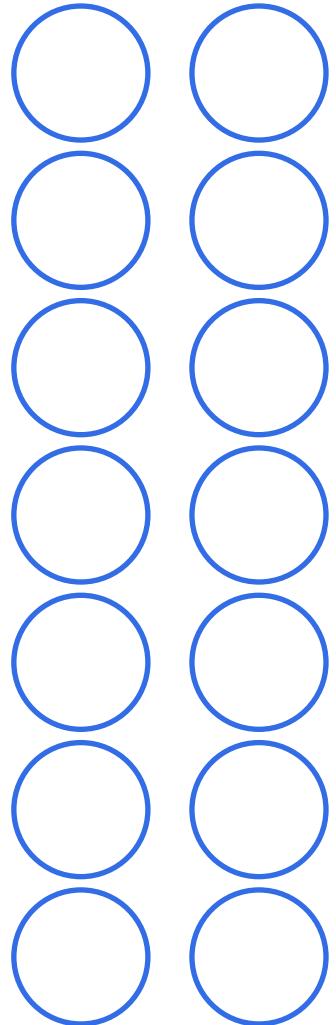


Kubernetes: Big Picture Overview

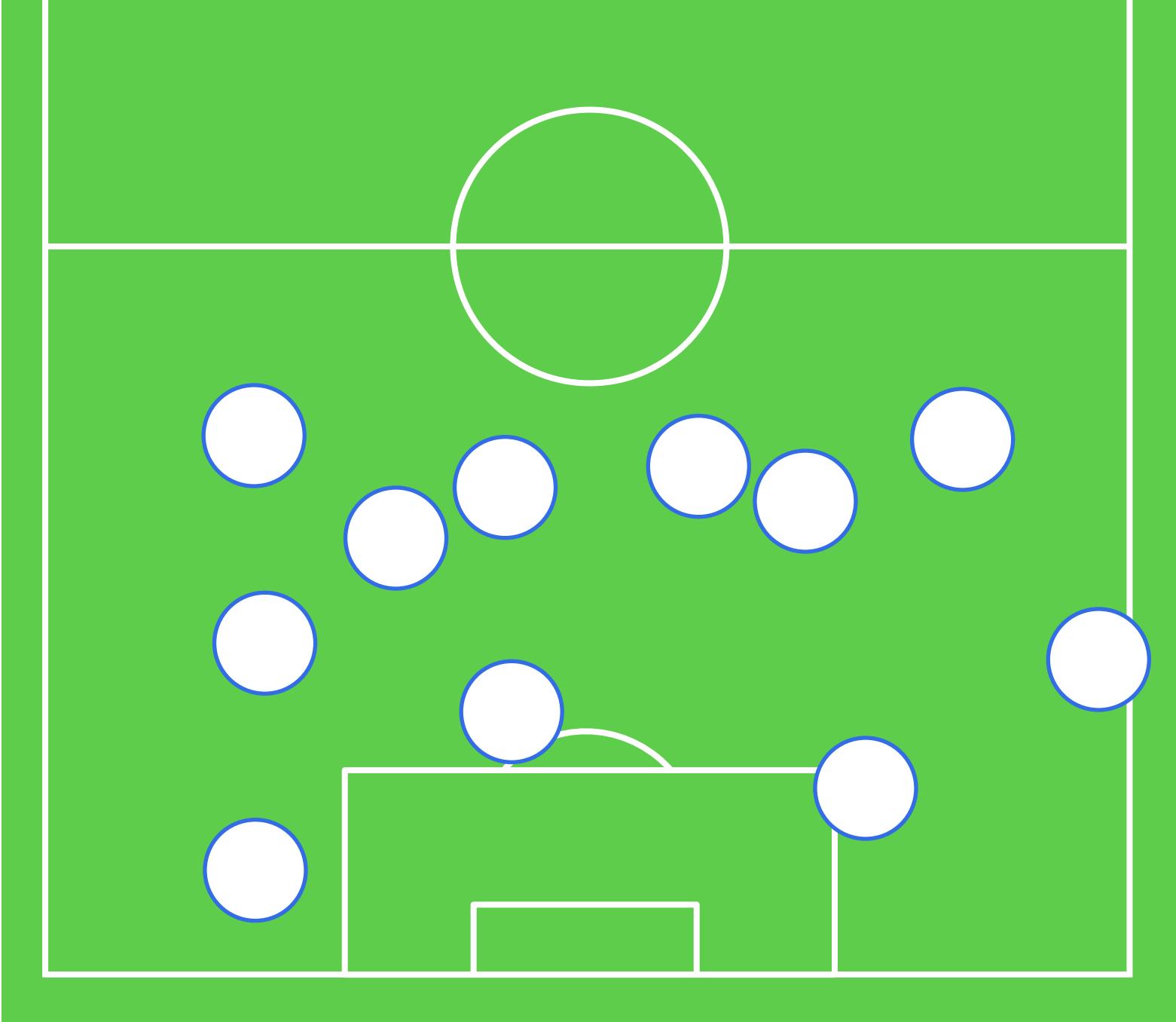
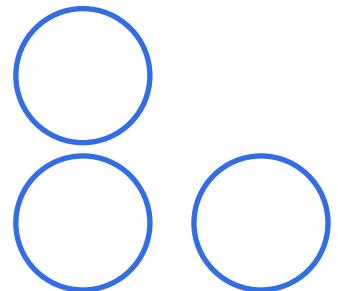




Team



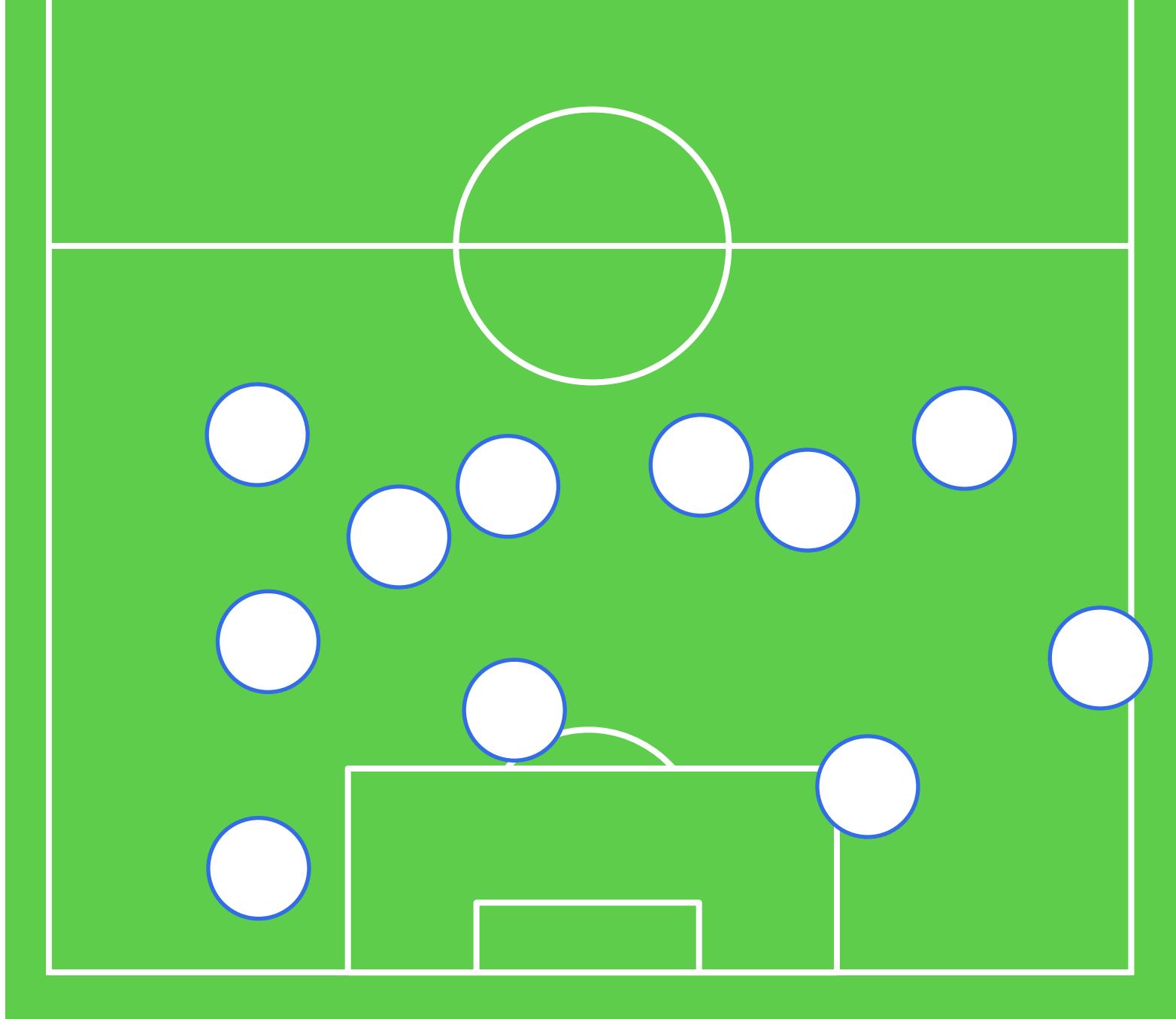
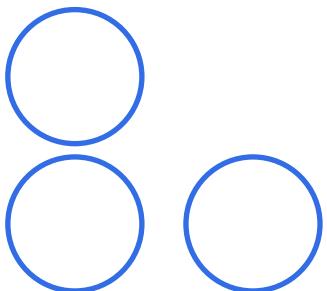
Team



Team



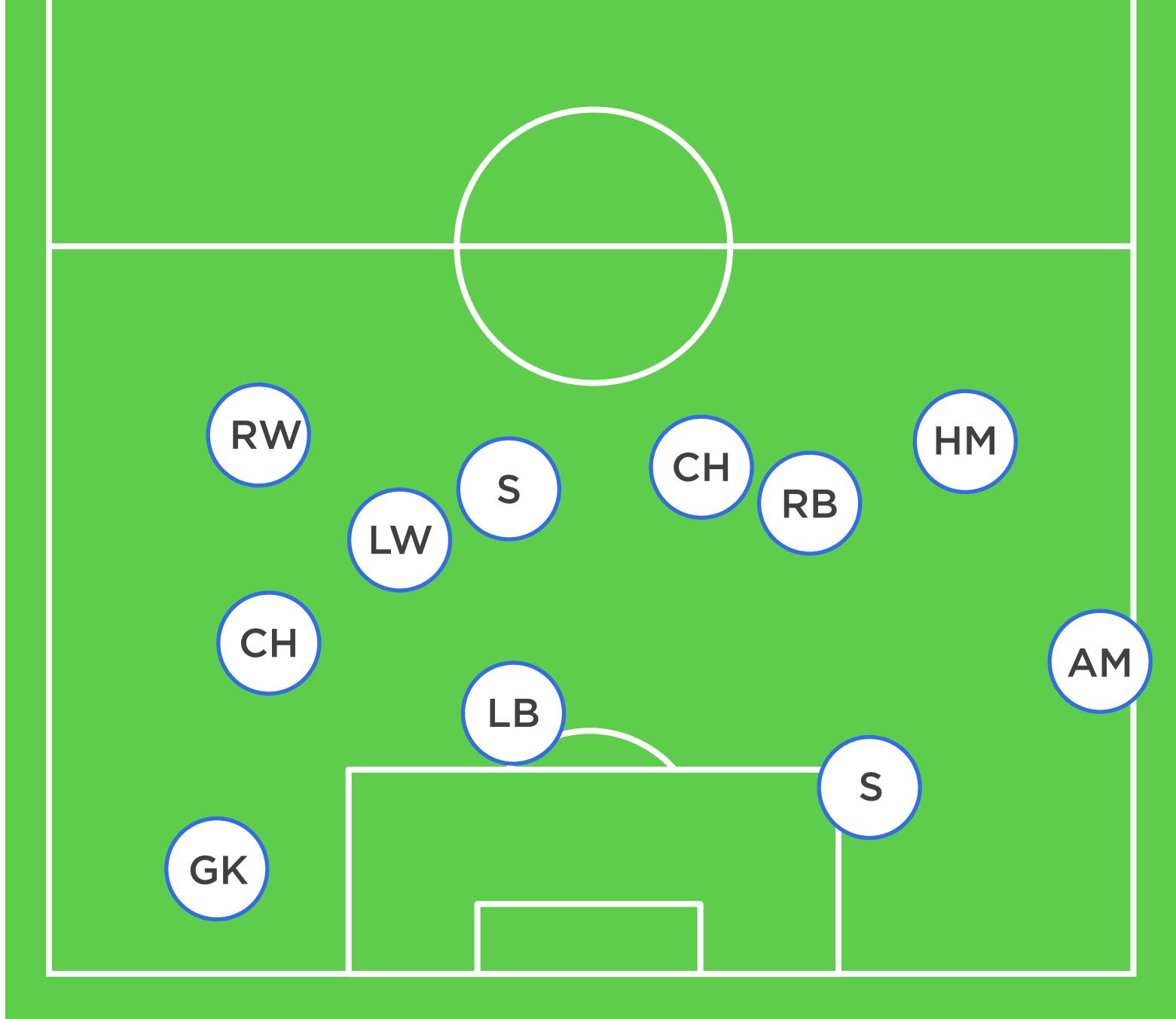
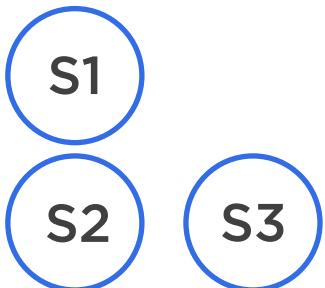
Manager
(coach)

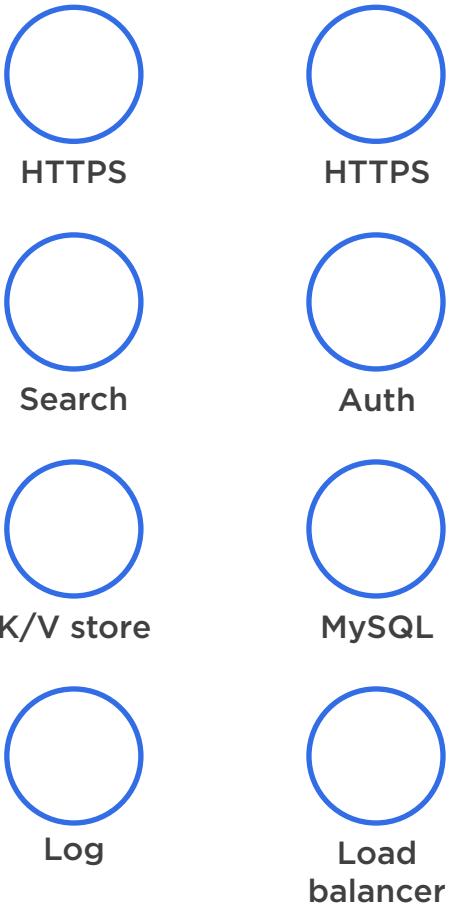


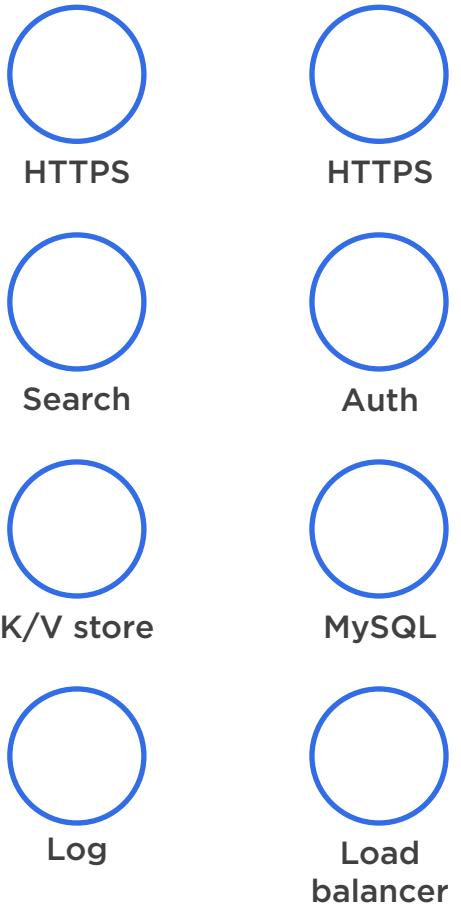
Team

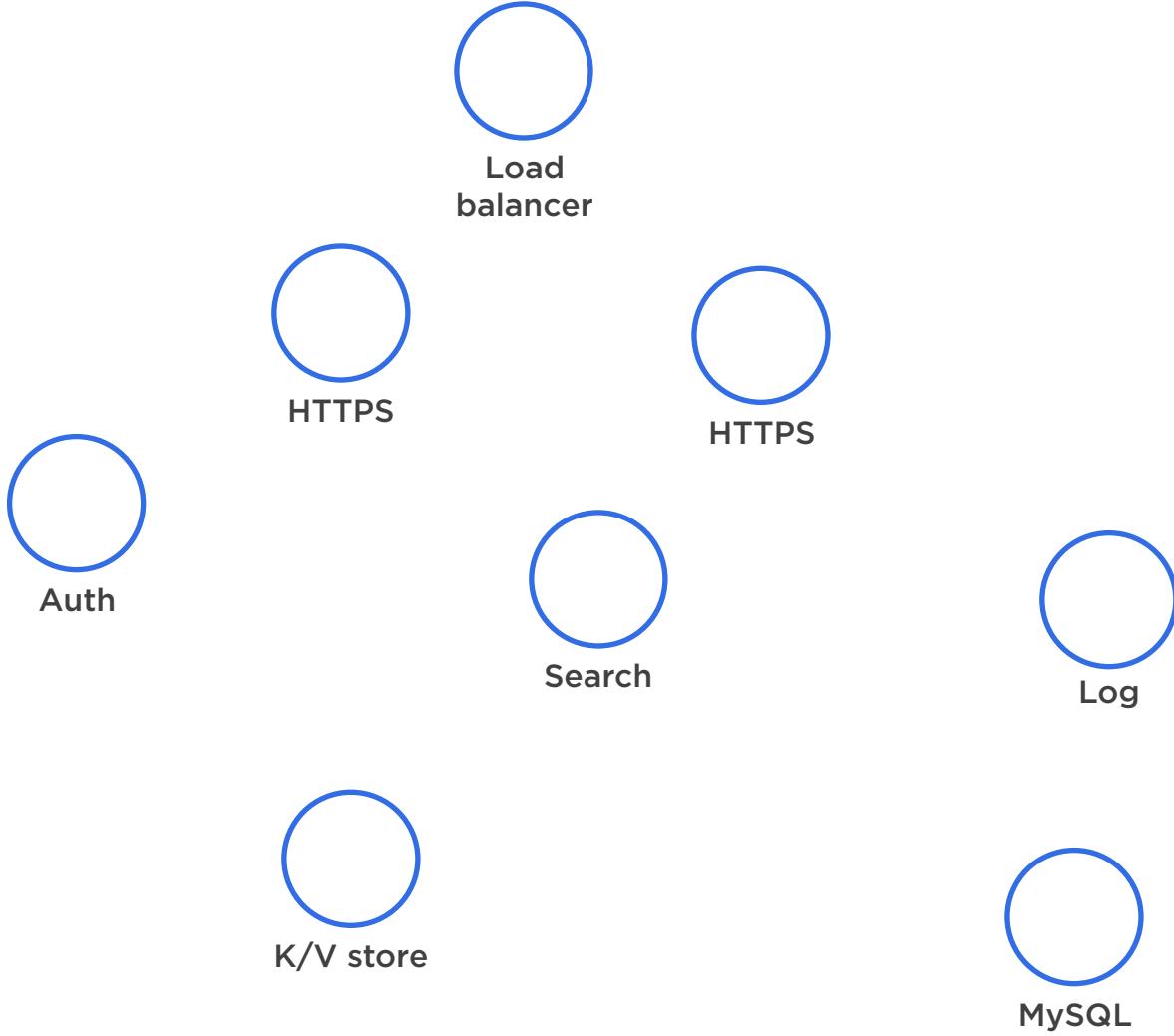


Manager
(coach)



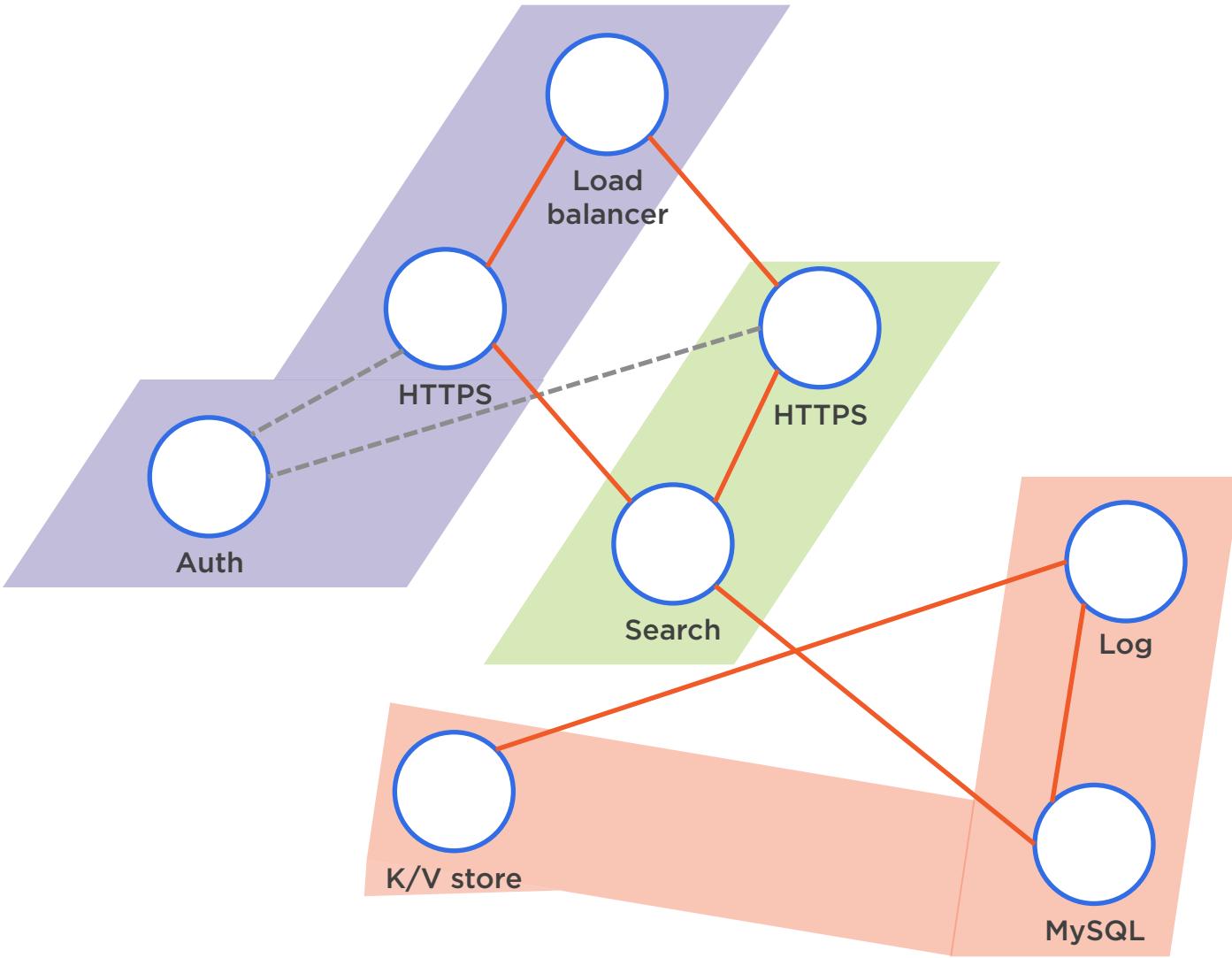


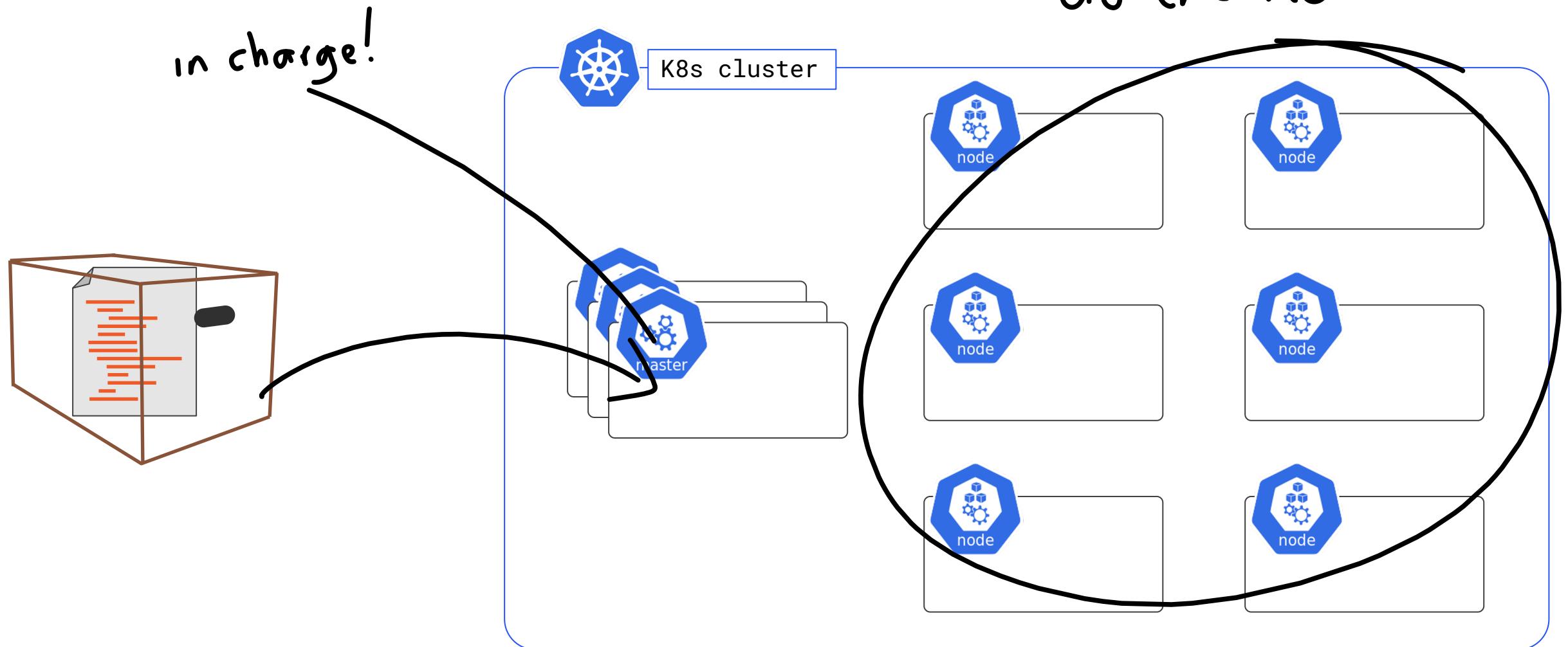


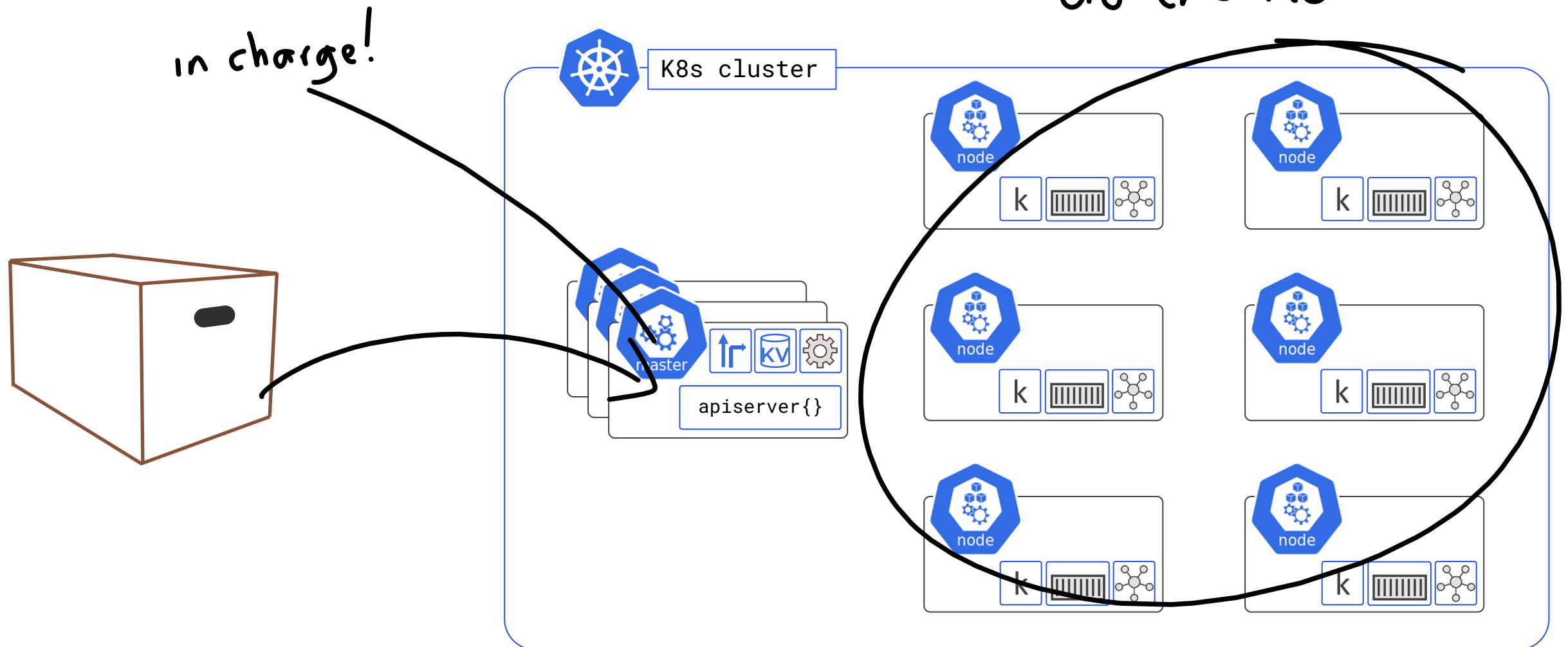


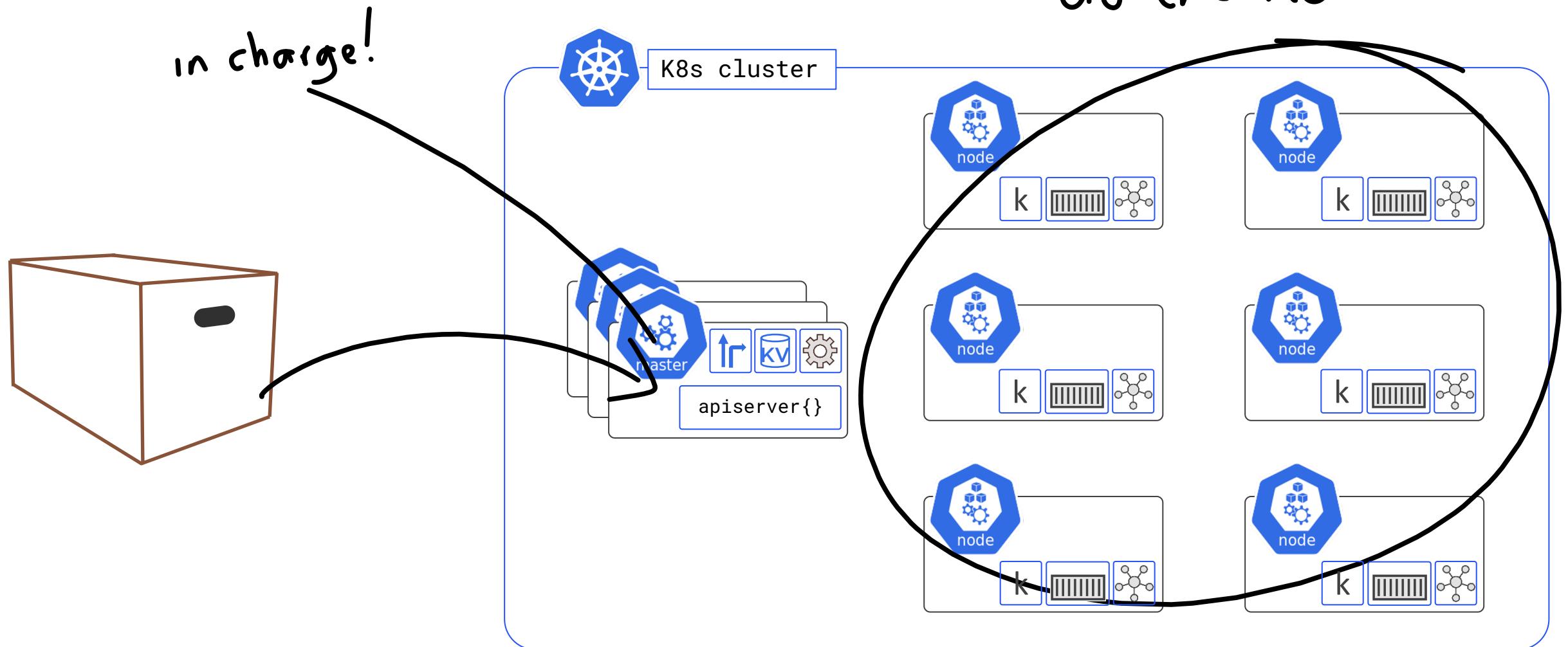


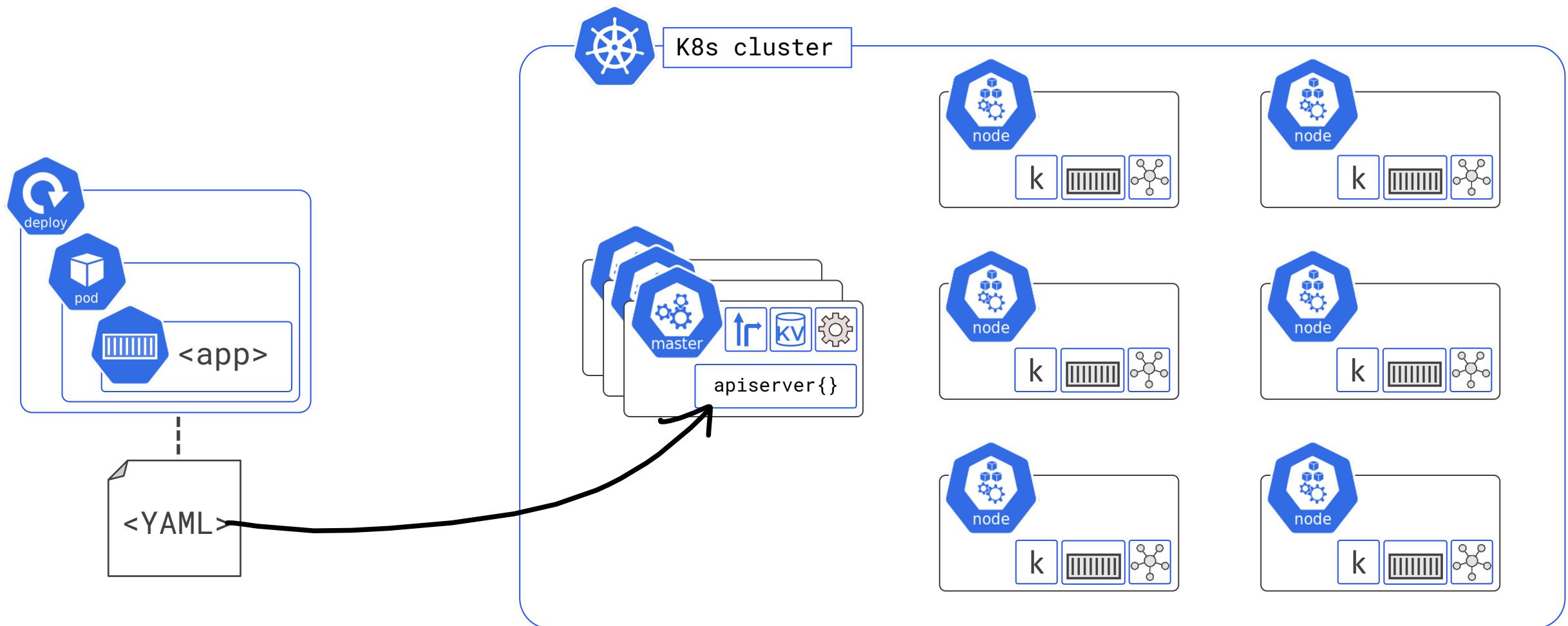
- Node 1
- Node 2
- Node 3











Up Next:
Masters



Masters

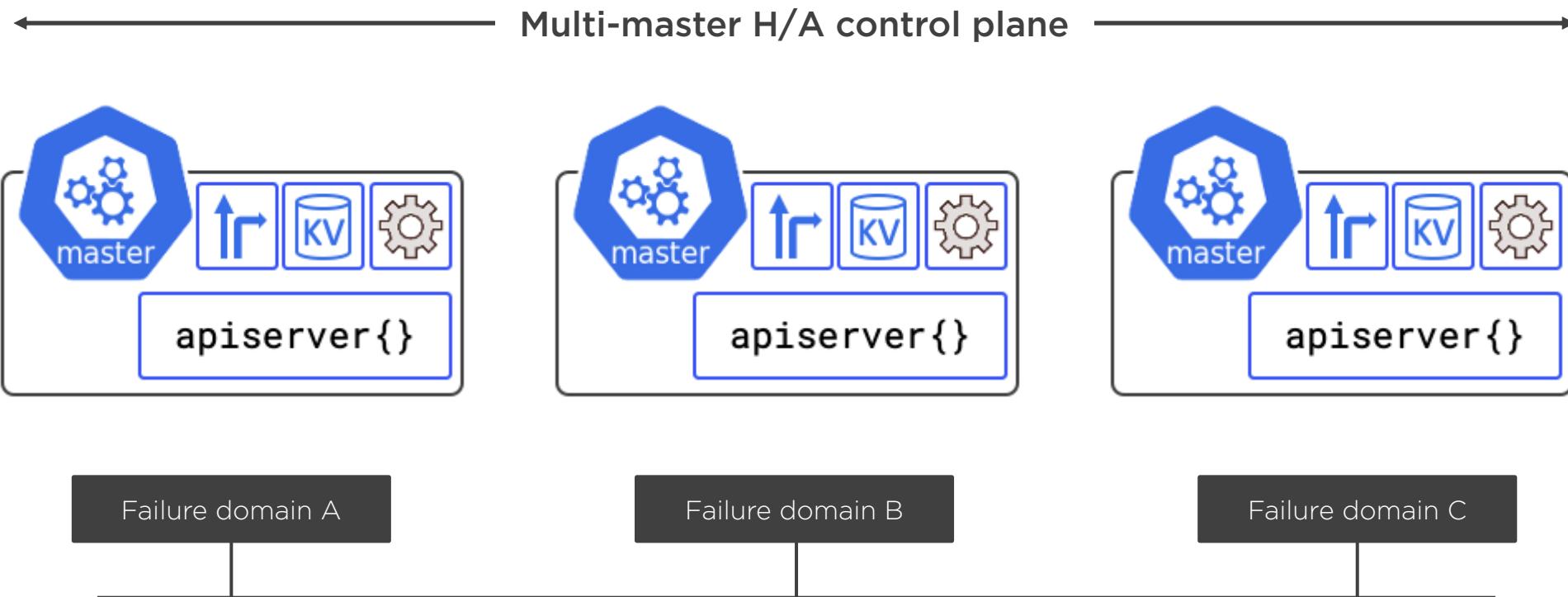


Masters

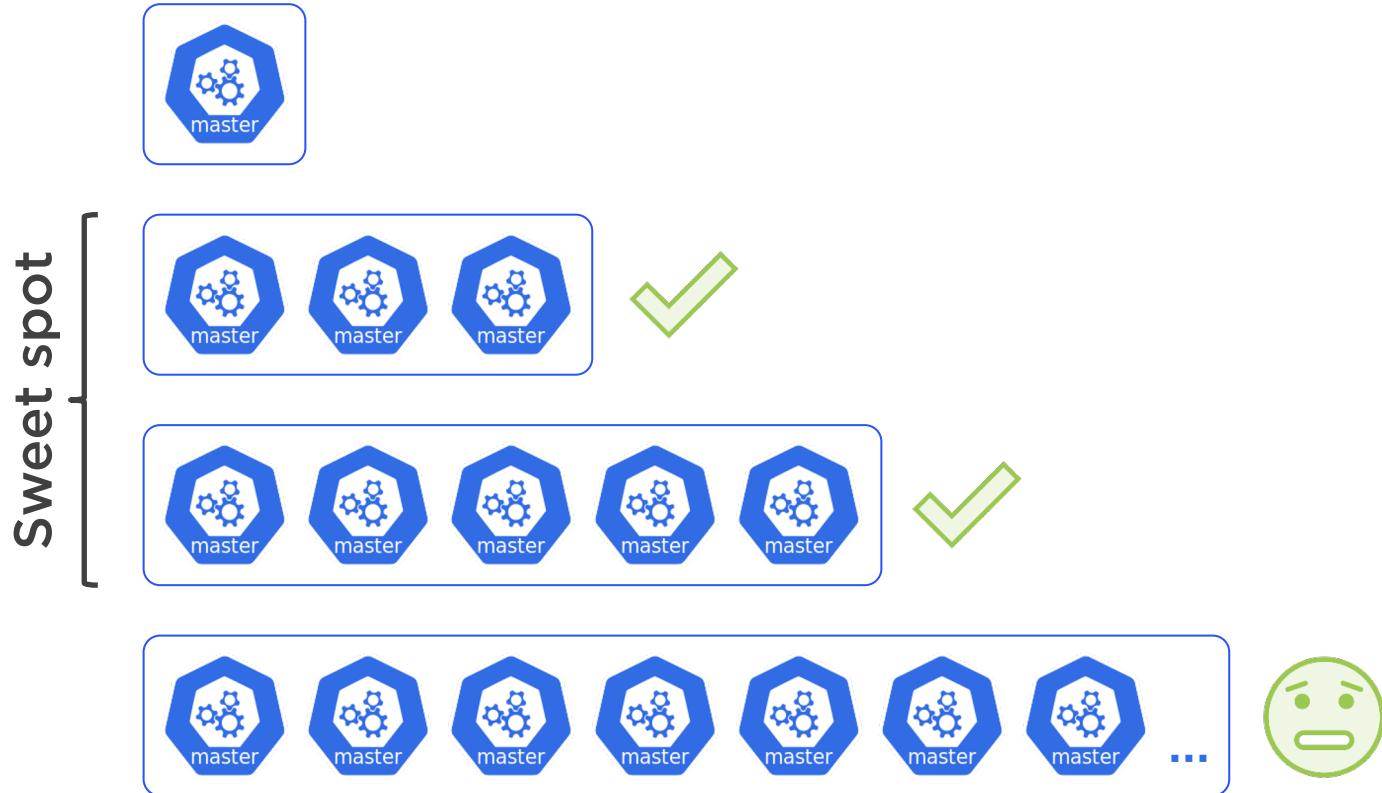
Also known as *head nodes* or the *control plane*







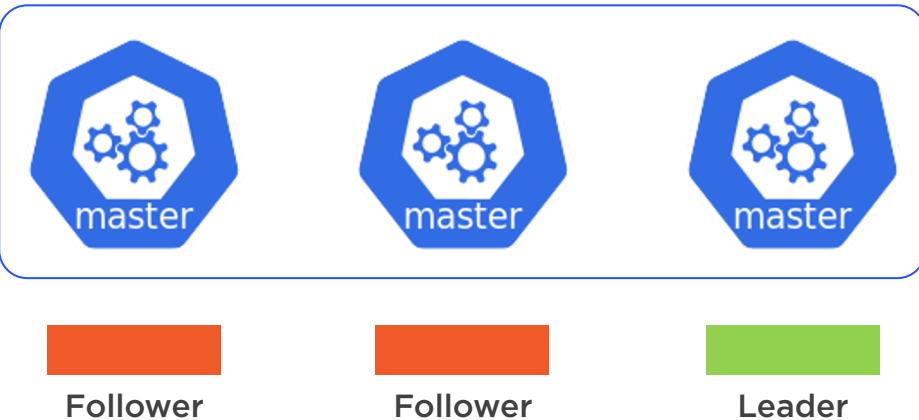
H/A Design

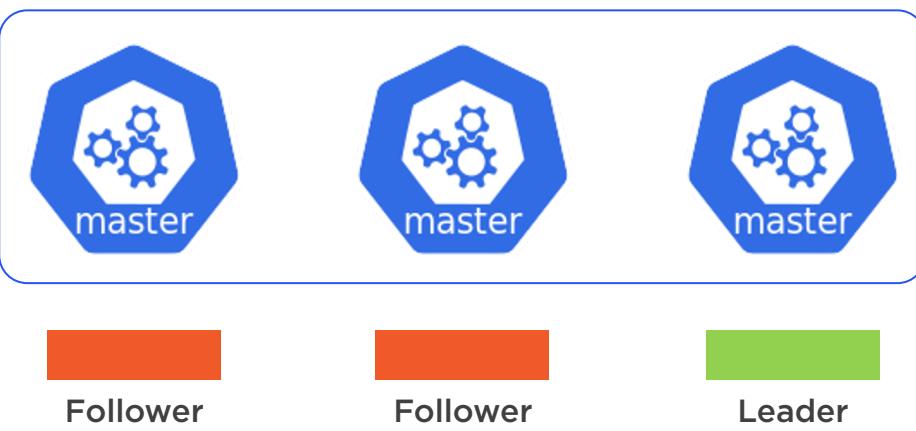




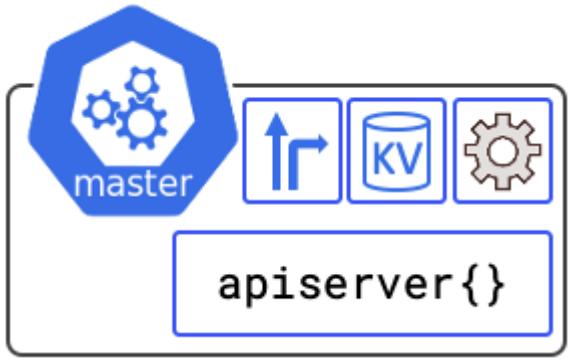


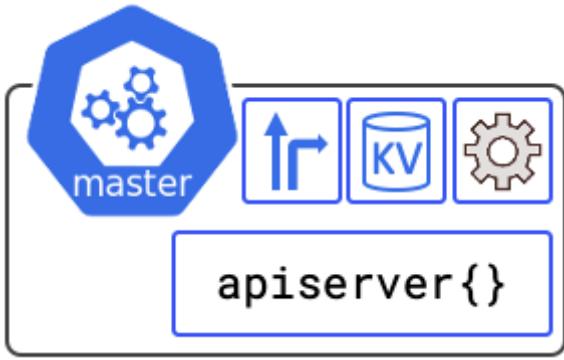
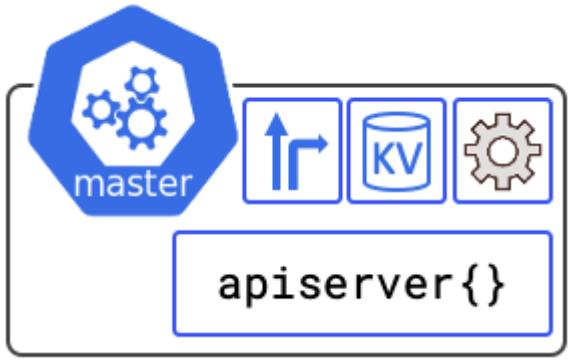


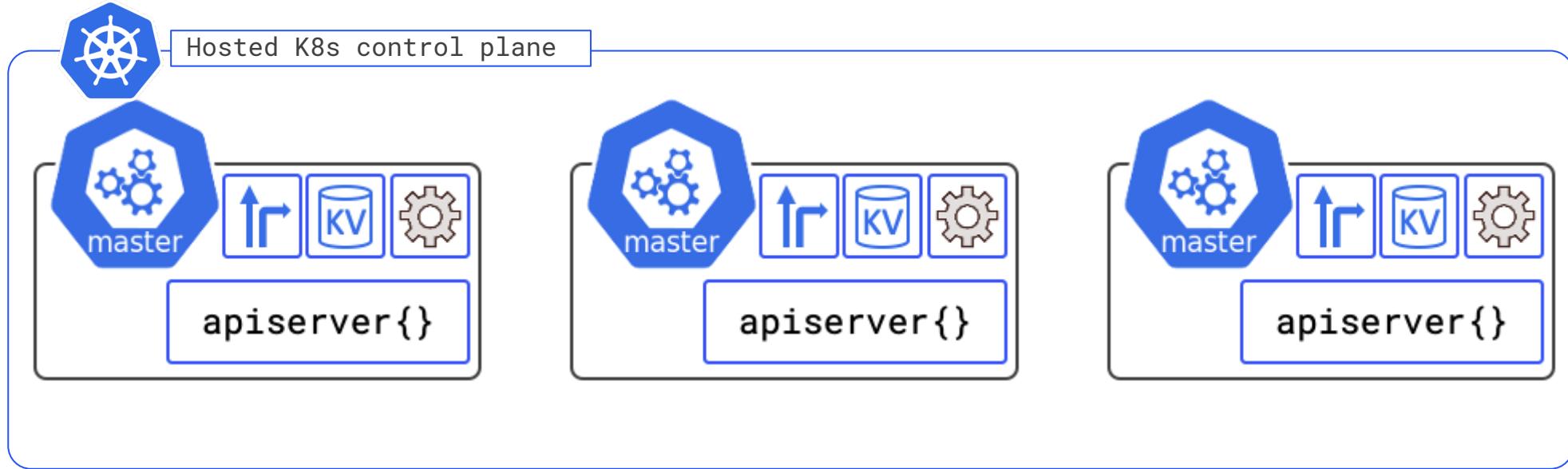


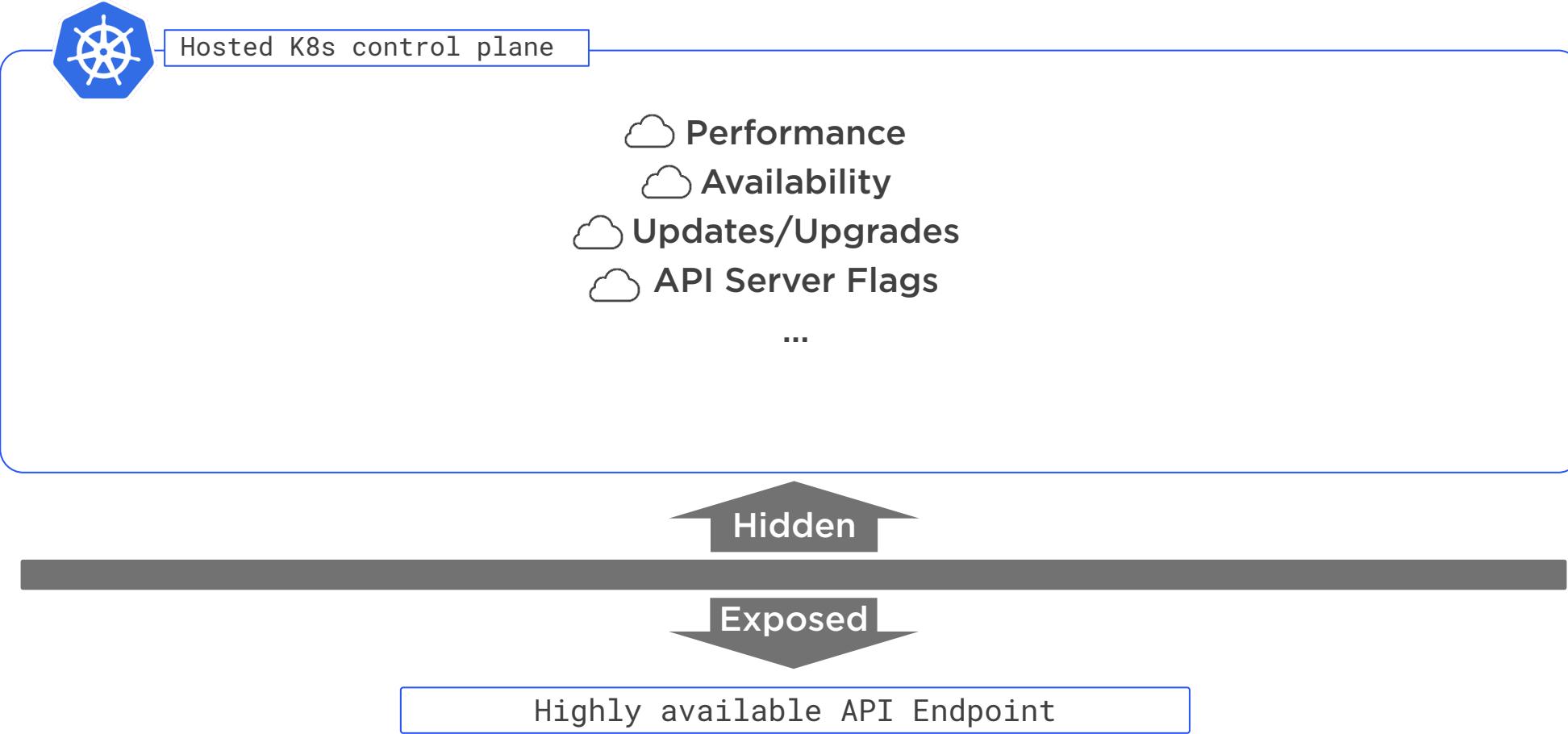


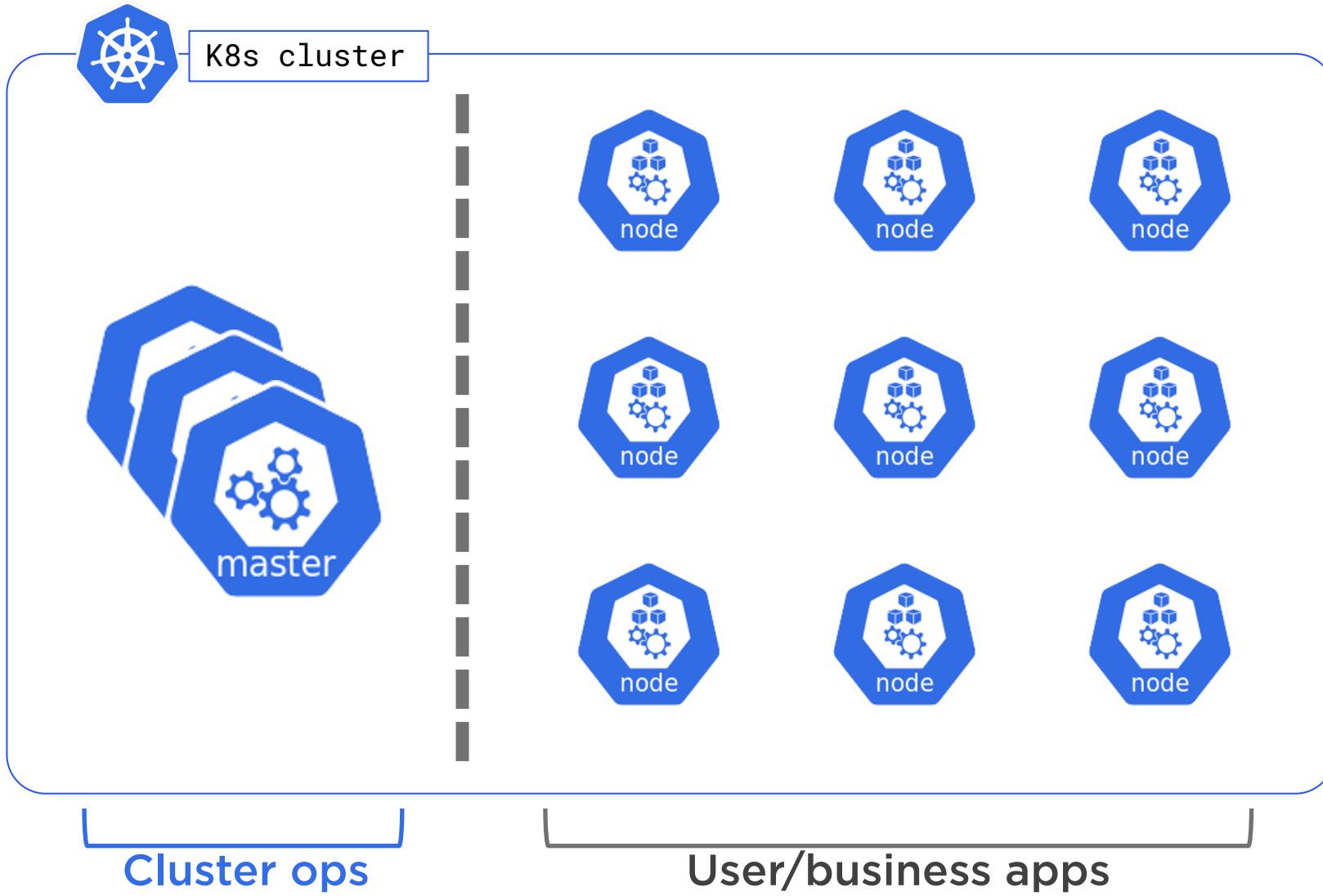




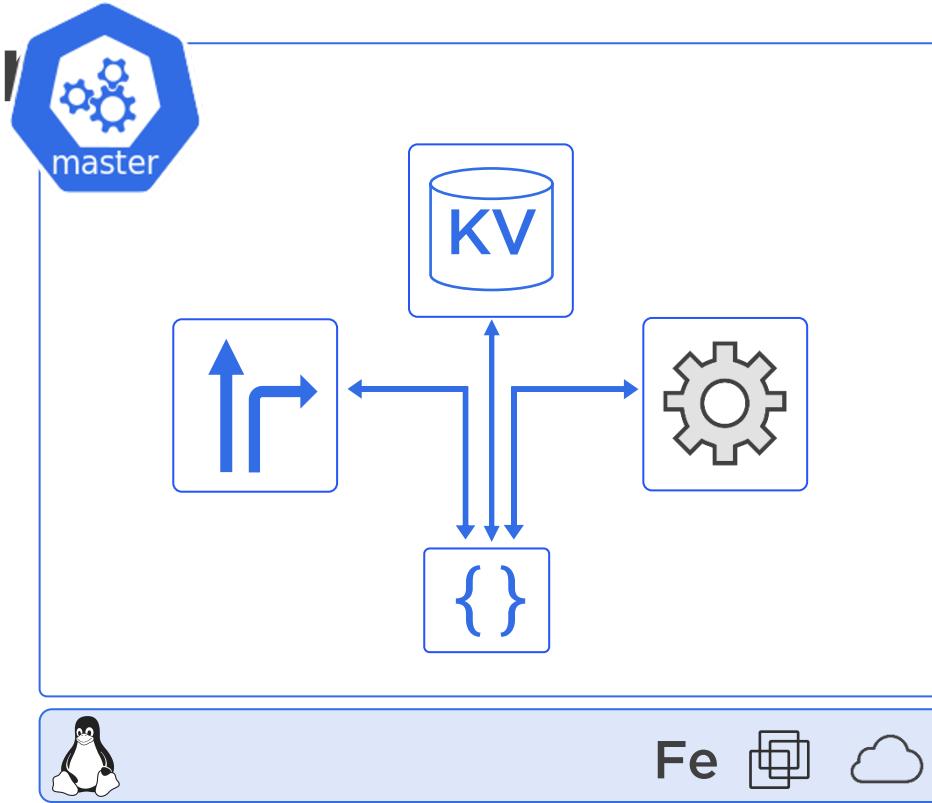






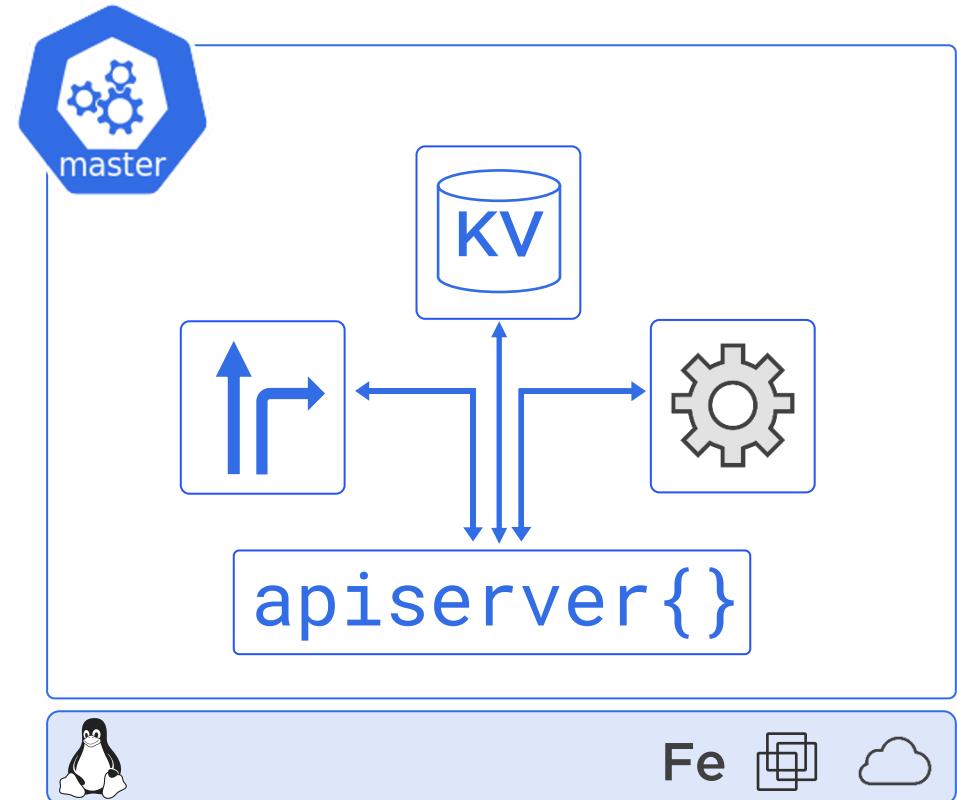


kube-apiserver



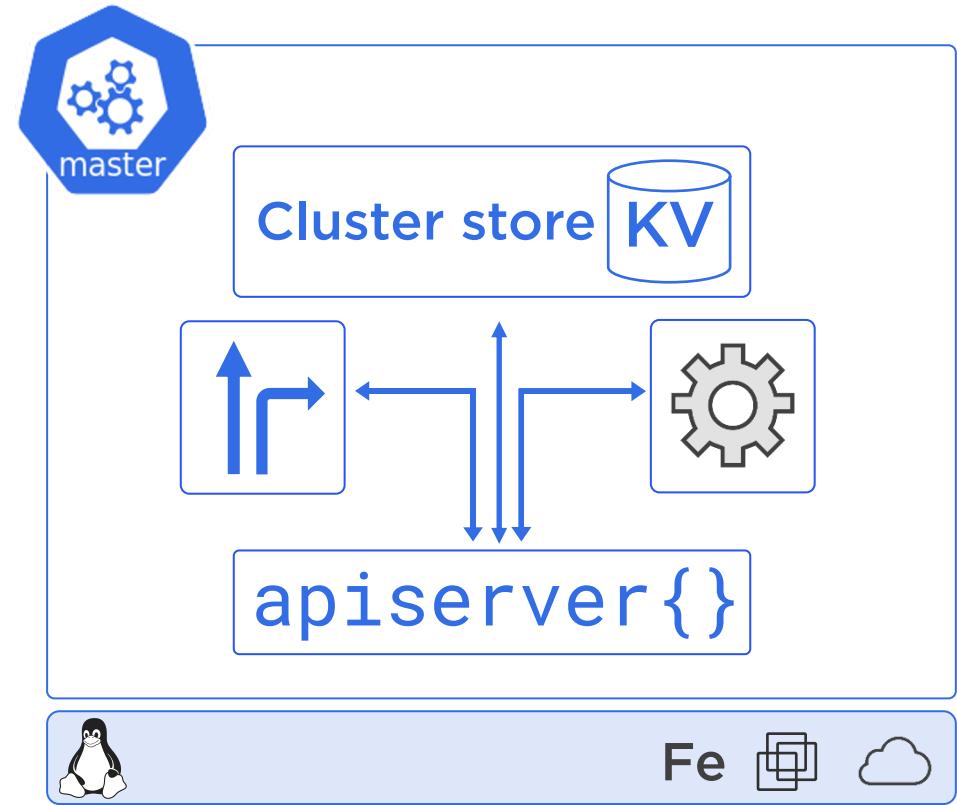
kube-apiserver

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



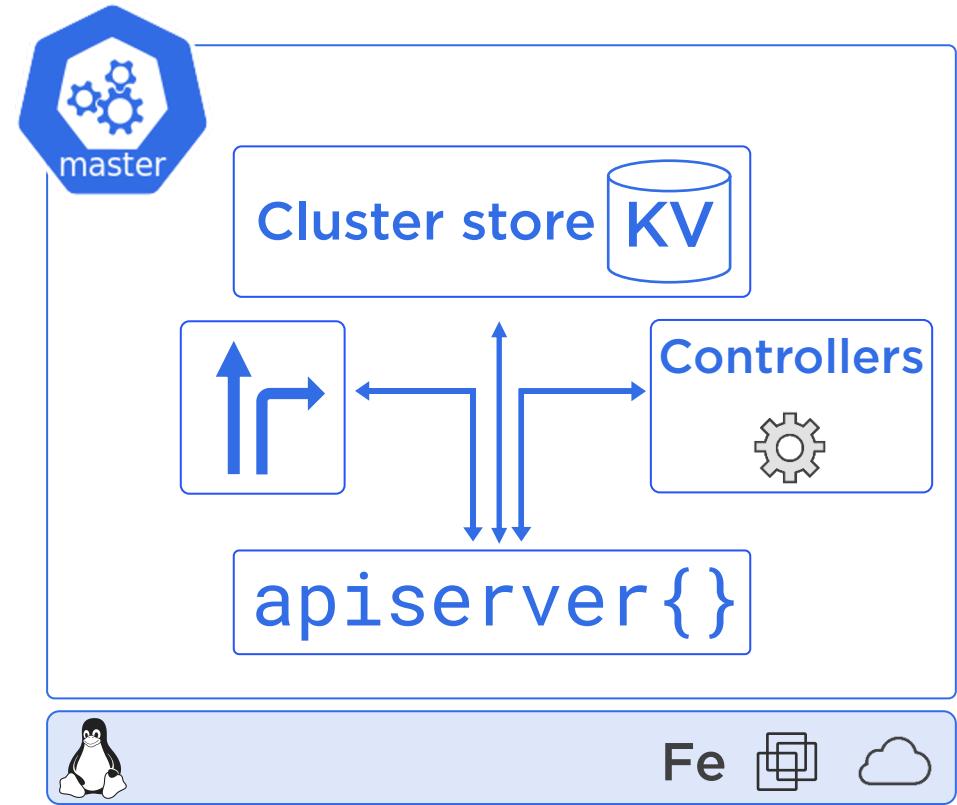
Cluster Store

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



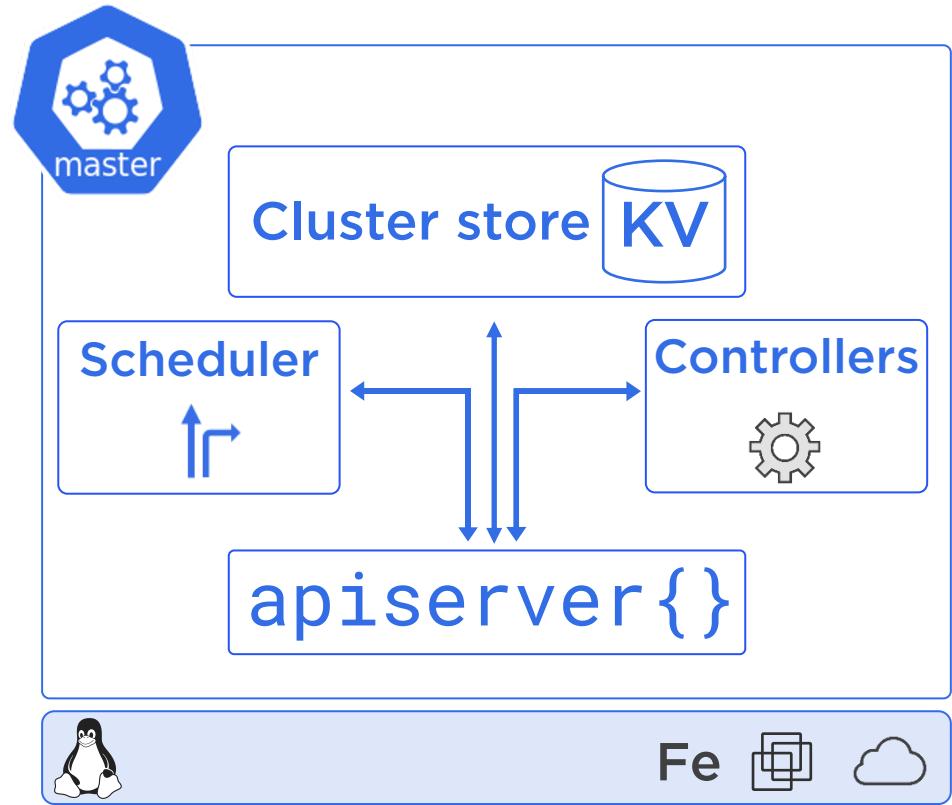
Kube-controller-manager

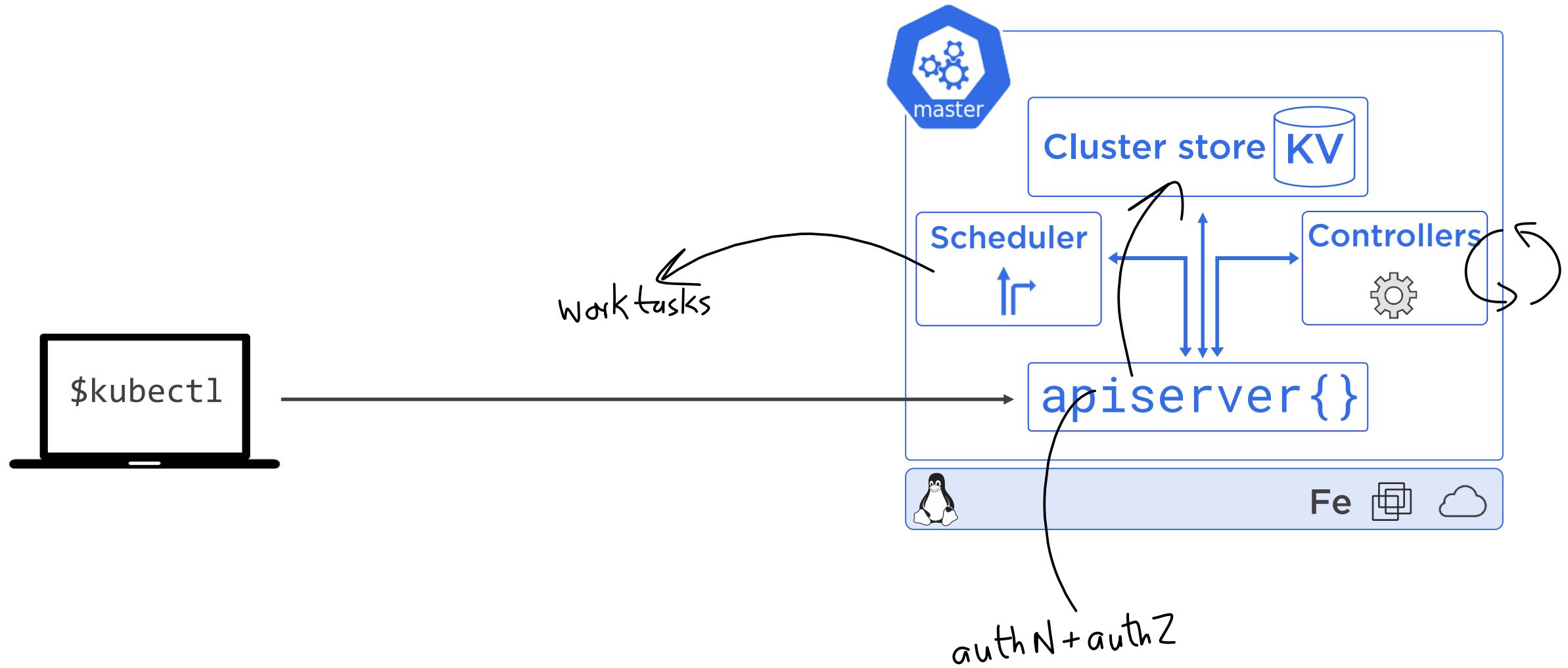
- Controller of controllers
 - Node controller
 - Deployment controller
 - Endpoints/EndpointSlice controller...
- Watch loops
- Reconciles observed state with desired state



Kube-scheduler

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



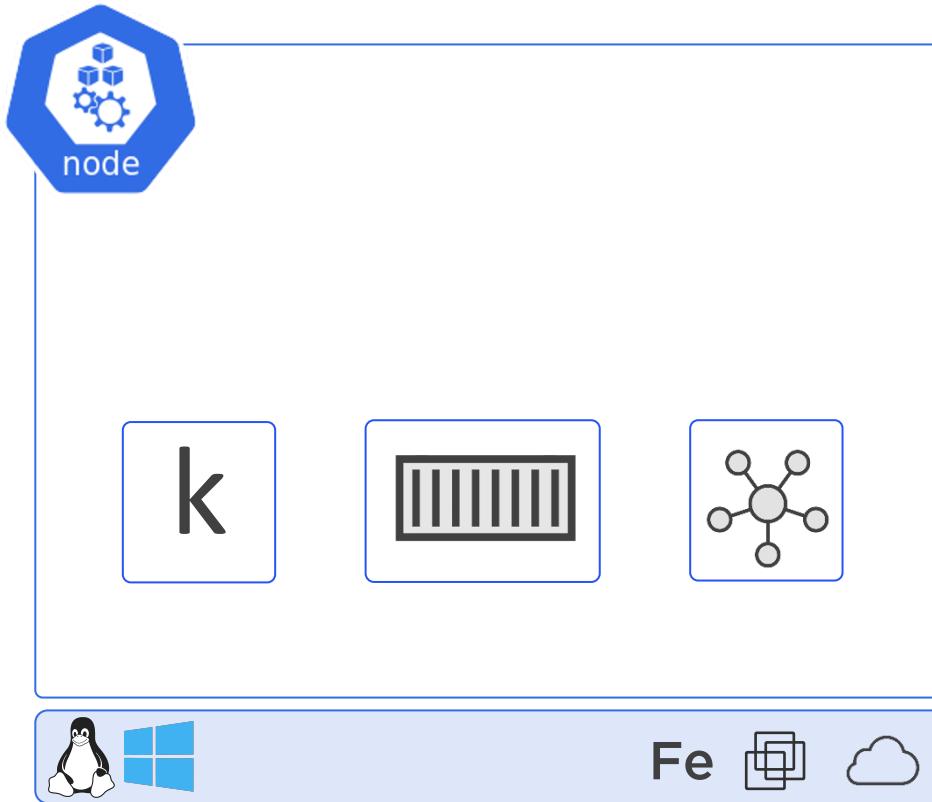


Up Next:
Nodes



Nodes

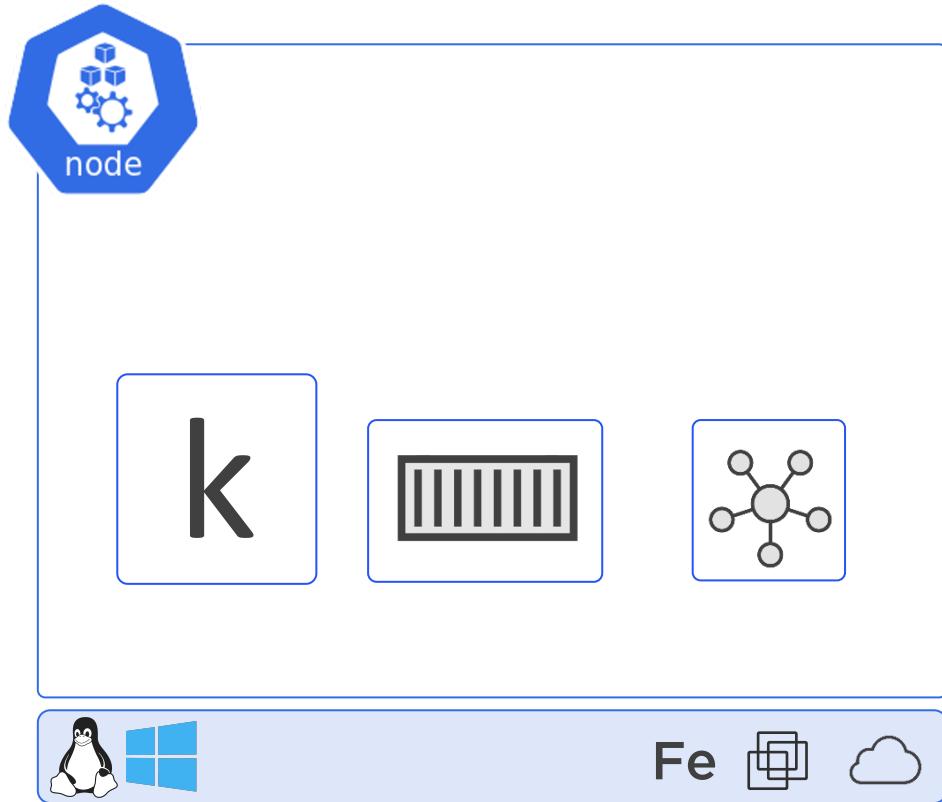






Kubelet

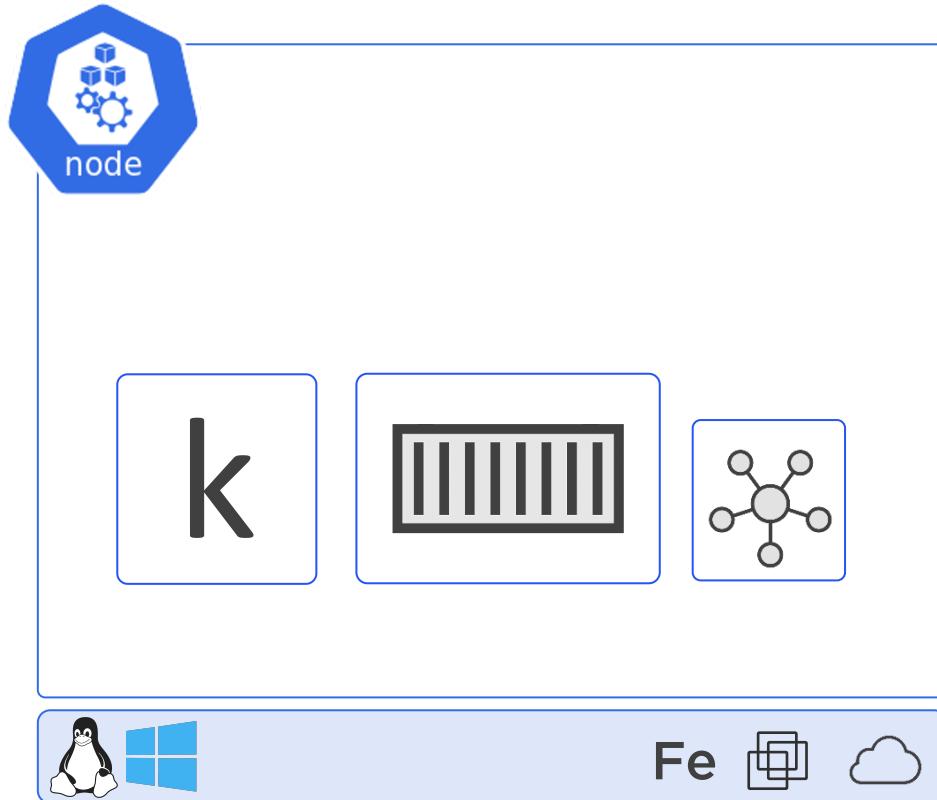
- Main Kubernetes agent
- Registers node with cluster
- Watches API Server for work tasks (Pods)
- Executes Pods
- Reports back to Masters

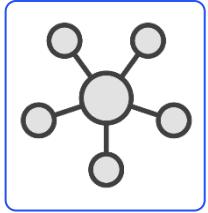




Container runtime

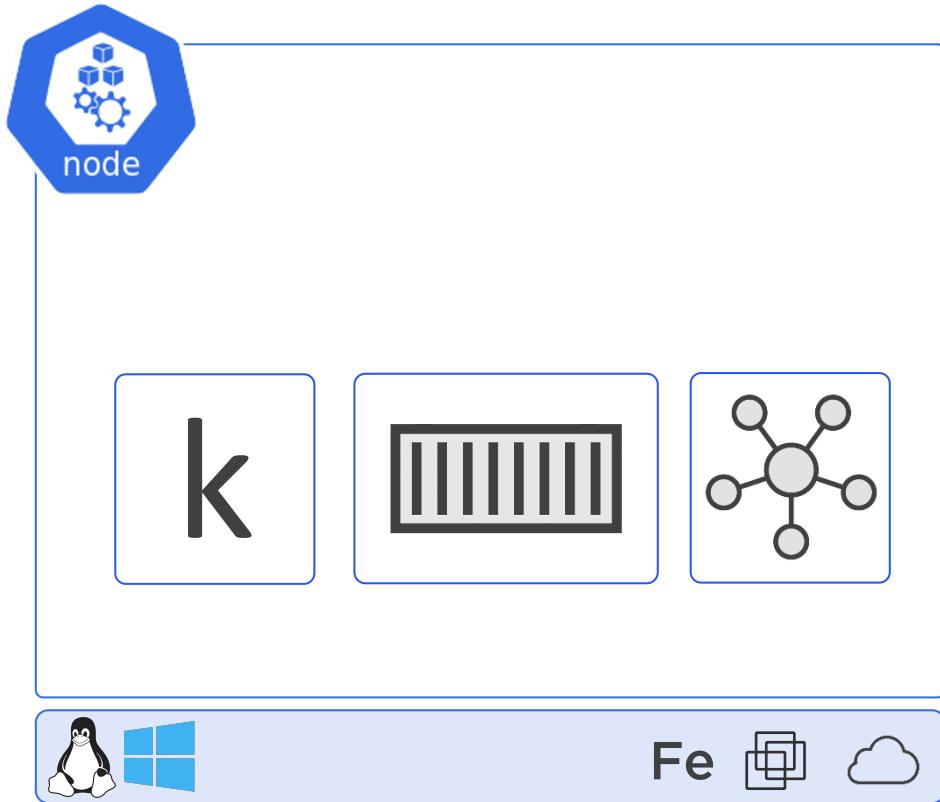
- Can be Docker
- Pluggable: Container Runtime Interface (CRI)
 - Docker, containerd, CRI-O, Kata...
- Low-level container intelligence

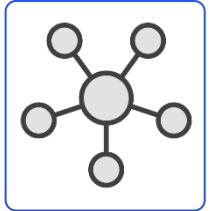




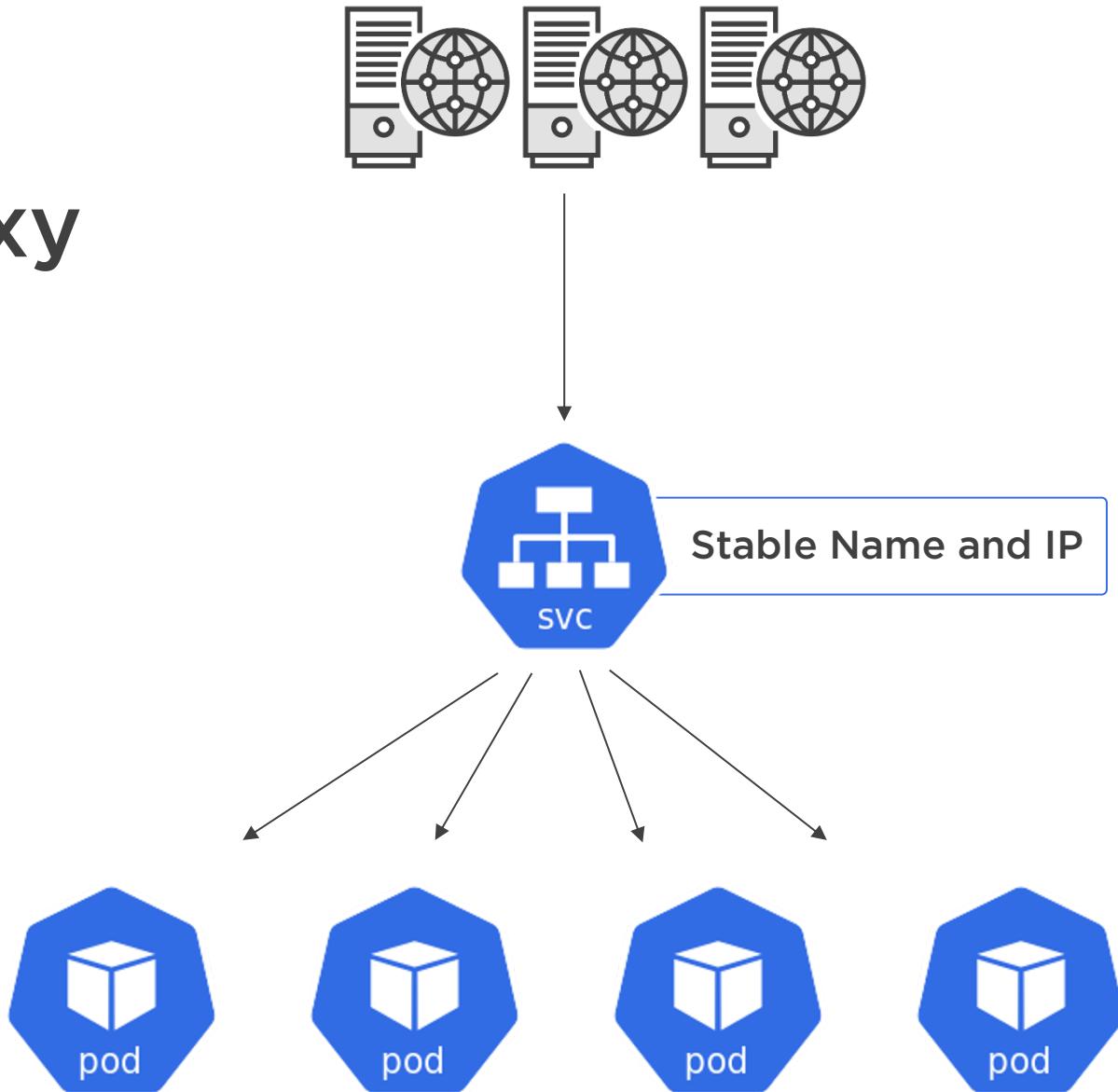
Kube-proxy

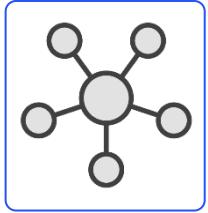
- Networking component
- Pod IP addresses





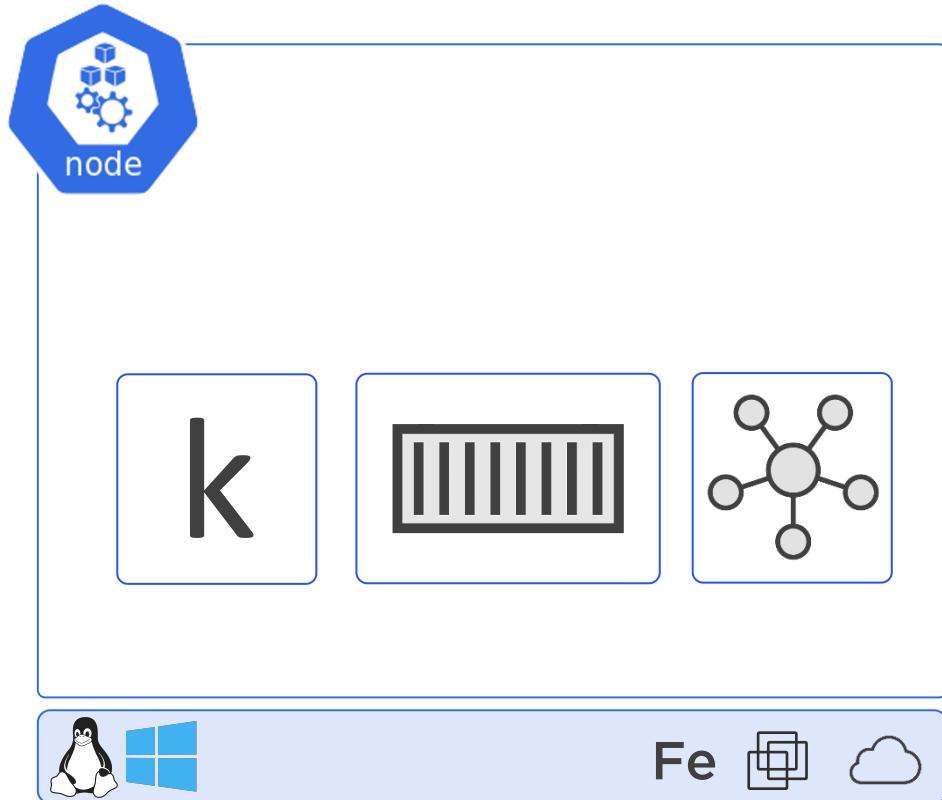
Kube-proxy

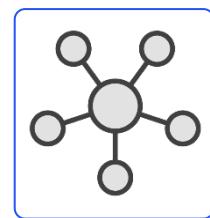
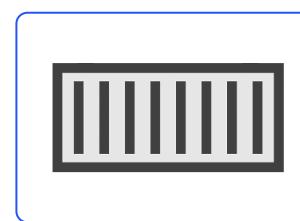
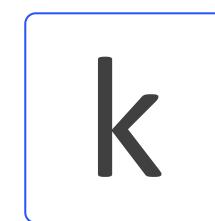
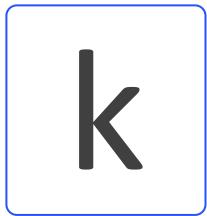




Kube-proxy

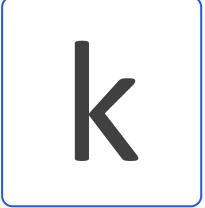
- Networking component
- Pod IP addresses
- Basic load-balancing





Fe





k

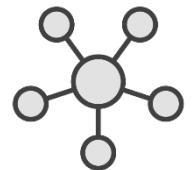
Kubelet

Main K8s agent



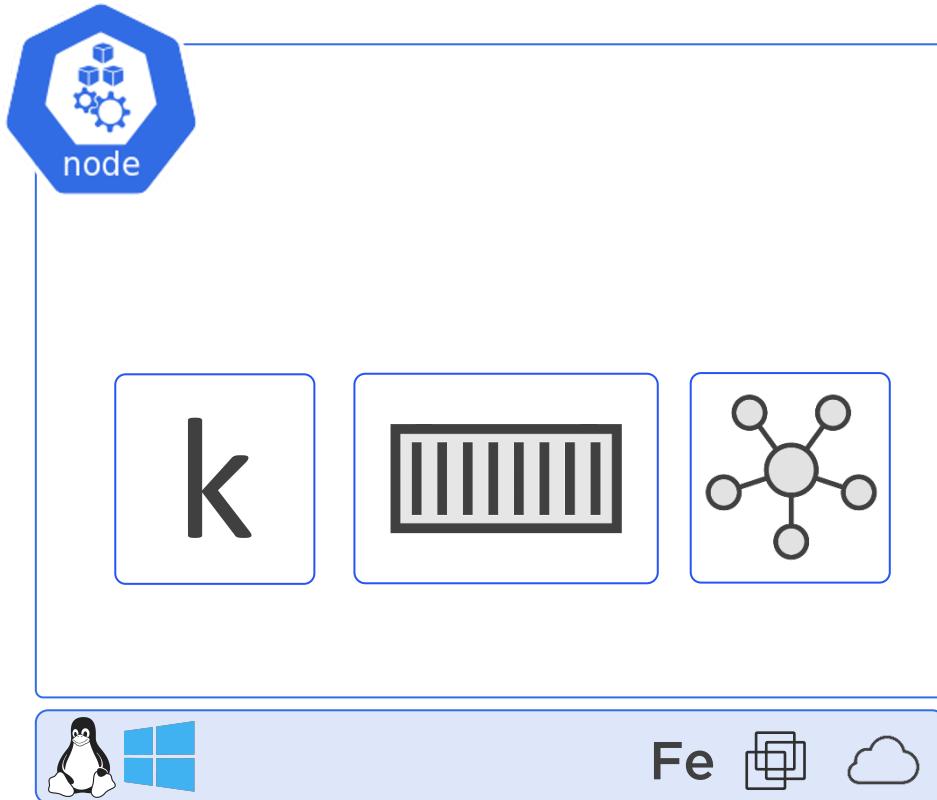
Container runtime

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



Kube-proxy

Vital role in networking





Virtual Kubelet



Nodeless Kubernetes





Virtual Kubelet



Nodeless Kubernetes





Virtual Kubelet

Pods run on cloud's hosted container back-end



Up Next:
The Declarative Model & Desired State



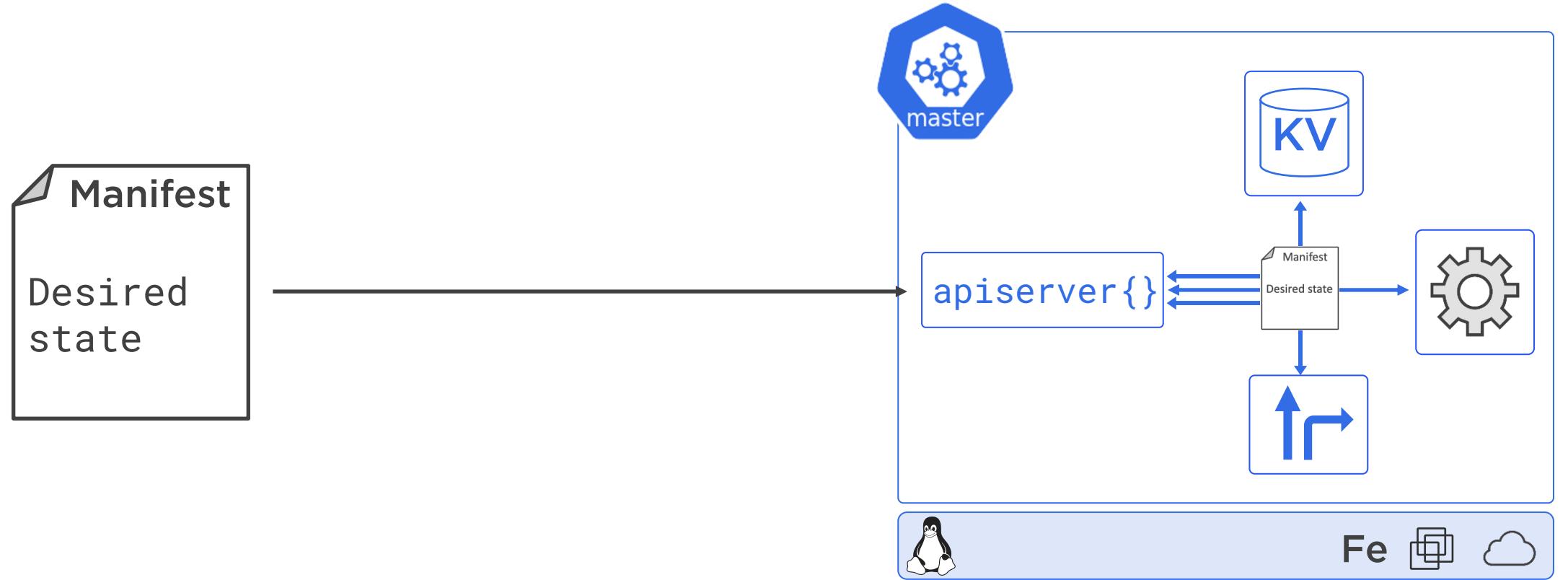
The Declarative Model and Desired State

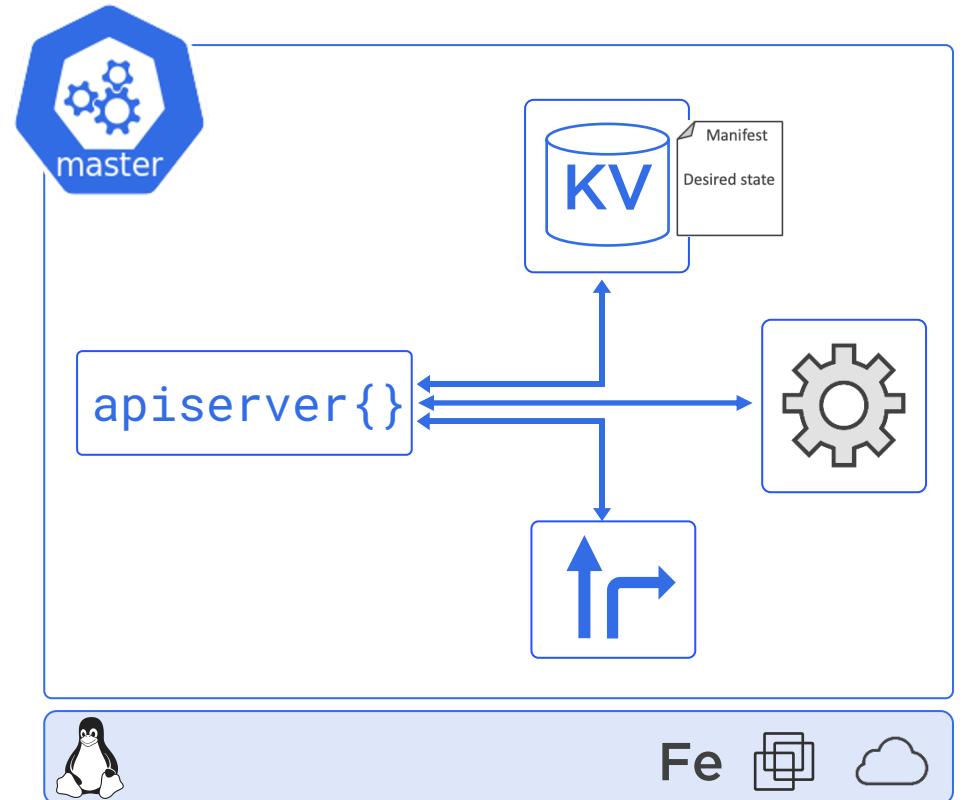


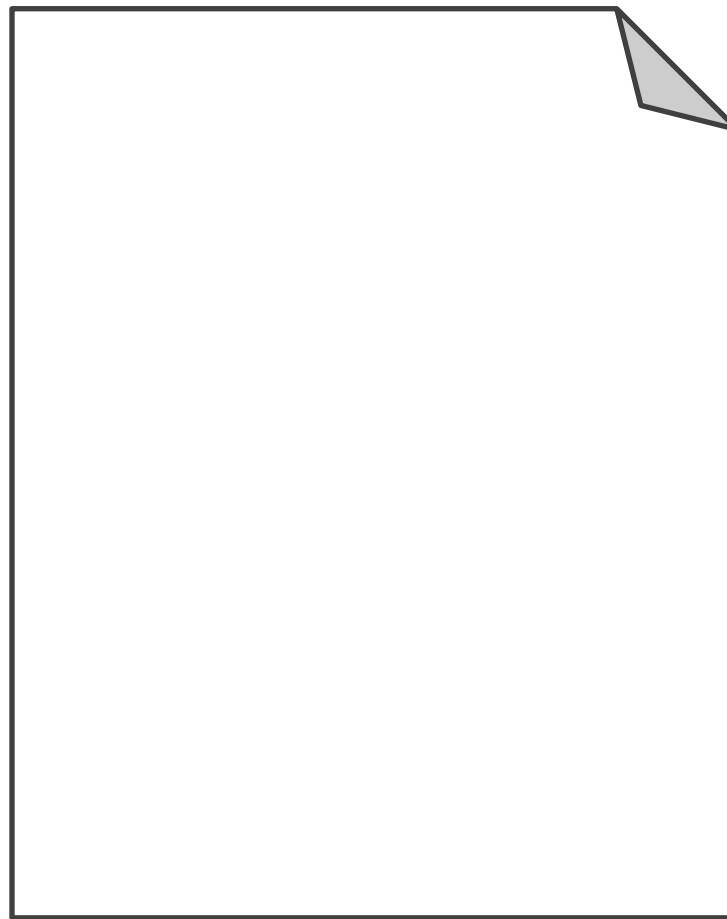
Declarative model

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









```
kind: Kitchen
spec:
  type: New
  location: Rear
  style: OpenPlan
  heating:
    type: Underfloor
    medium: Water
  windows:
    type: FloorToCeiling
    aspect: South
  doors:
    type: FireDoor
    accessTo: Garage
  island: Yes
  roofGarden: Yes
  ...
```



```
kind: Kitchen
spec:
  type: New
  location: Rear
  style: OpenPlan
  heating:
    type: Underfloor
    medium: Water
  windows:
    type: FloorToCeiling
    aspect: South
  doors:
    type: FireDoor
    accessTo: Garage
  island: Yes
  roofGarden: Yes
  ...
```





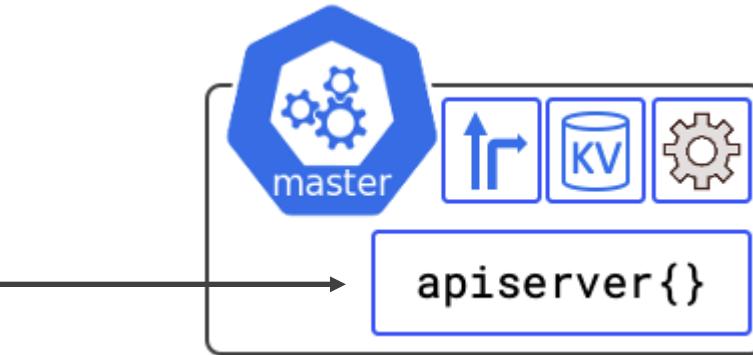
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: test
spec:
  replicas: 6
  selector:
    matchLabels:
      app: ps-test
  template:
    spec:
      containers:
        - name: c1
          image: web1:1.3
          ports: 8080
            - containerPort
              ...

```

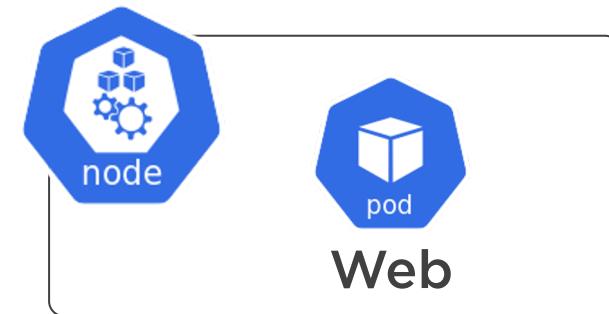
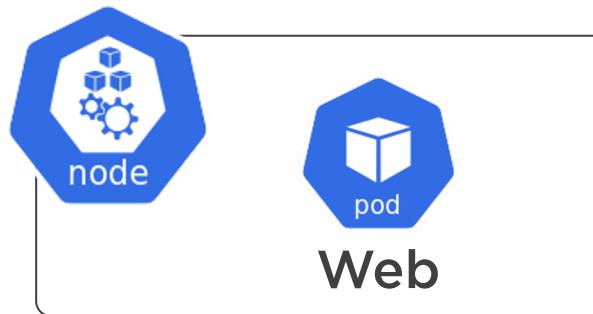
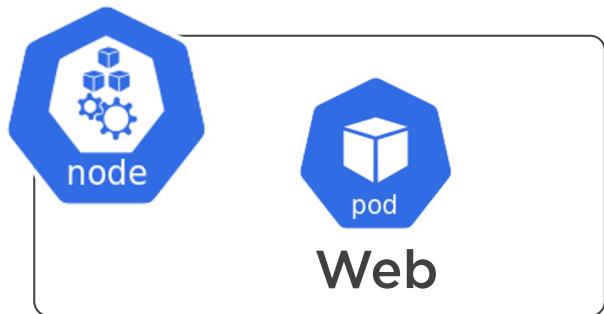
Declarative
(Declaring what you want)



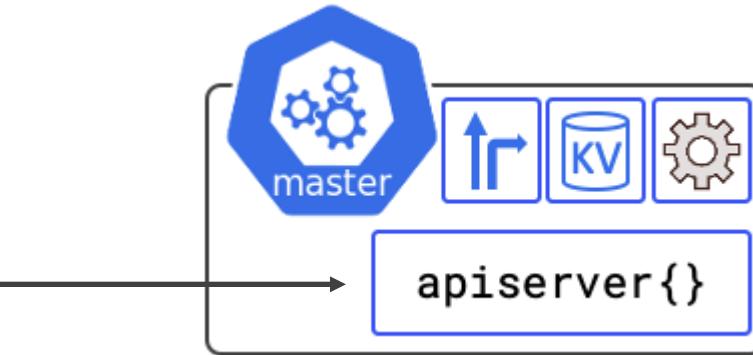
```
name: web  
replicas: 3  
...
```



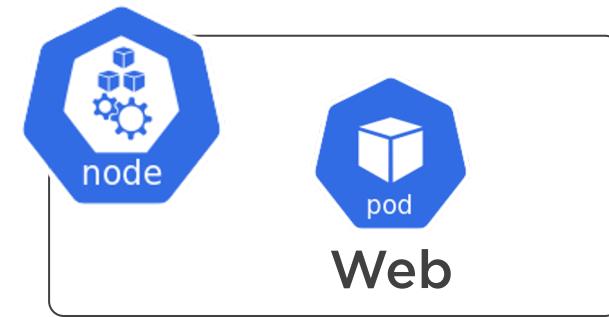
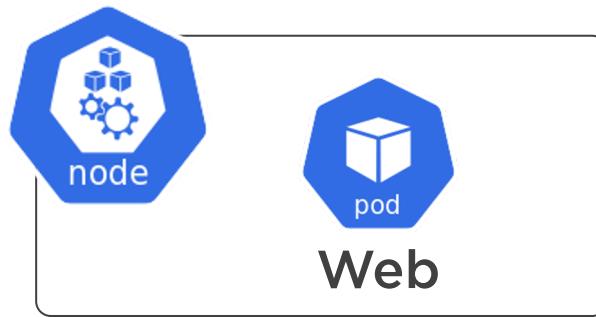
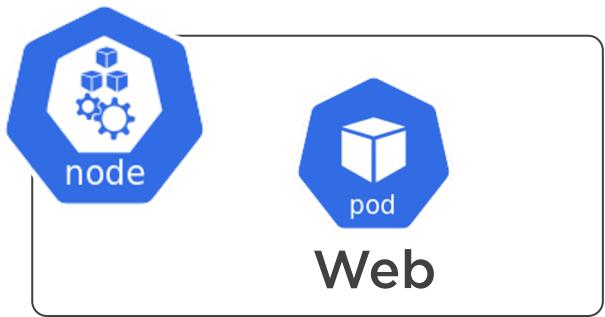
Desired state: 3
Observed state: 3



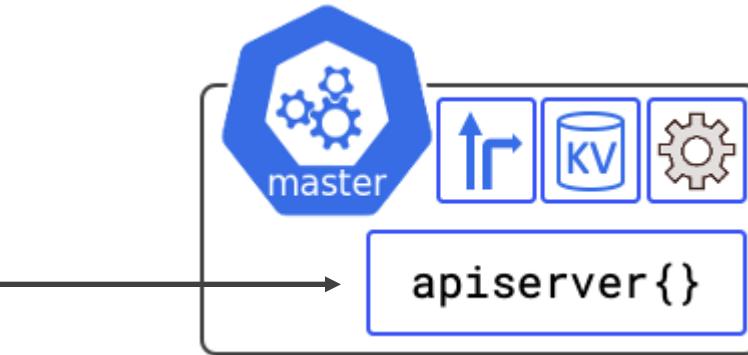
```
name: web  
replicas: 3  
...
```



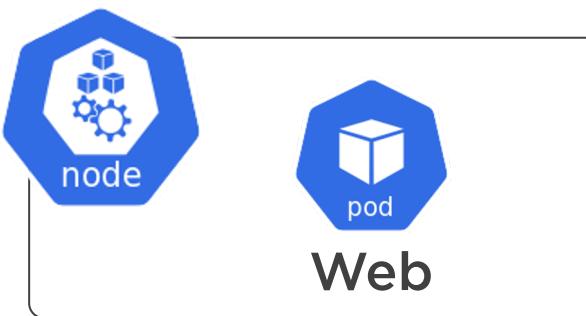
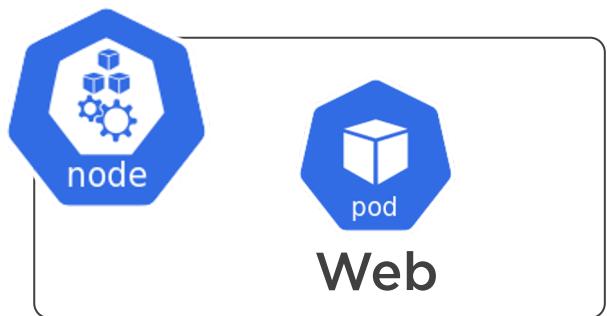
Desired state: 3
Observed state: 3



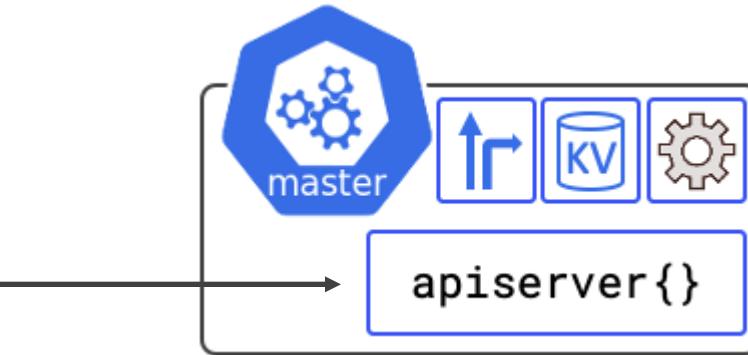
```
name: web  
replicas: 3  
...
```



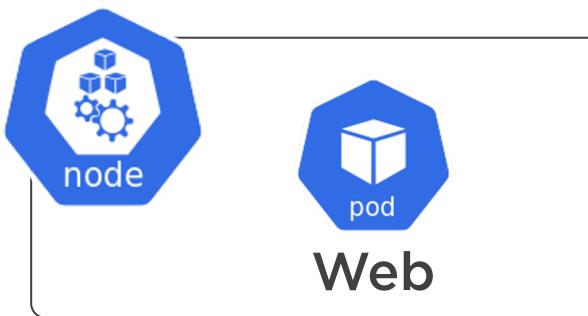
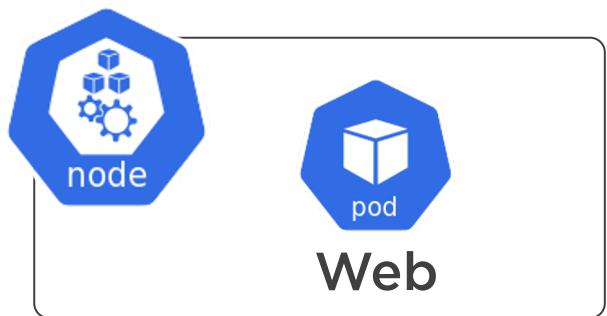
Desired state: 3
Observed state: 3



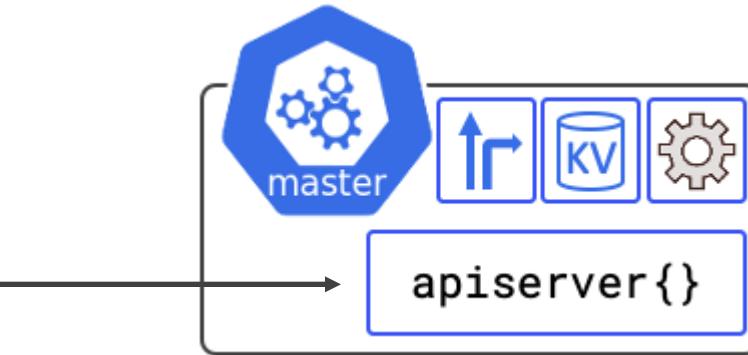
```
name: web  
replicas: 3  
...
```



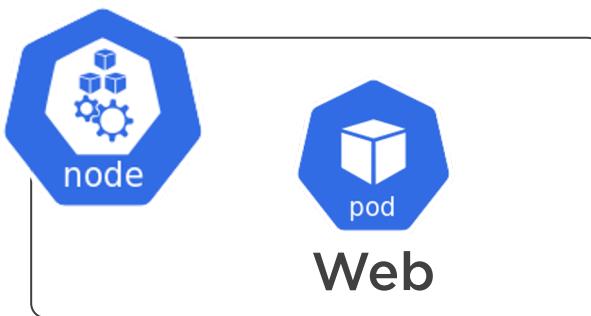
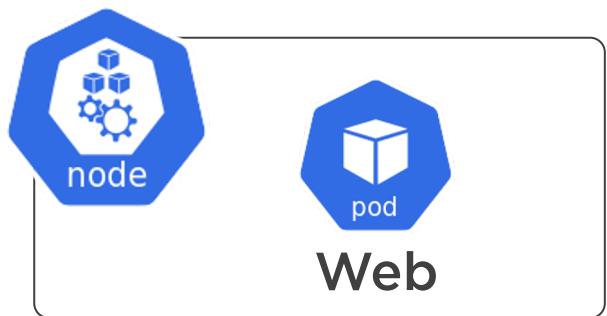
Desired state: 3
Observed state: 2



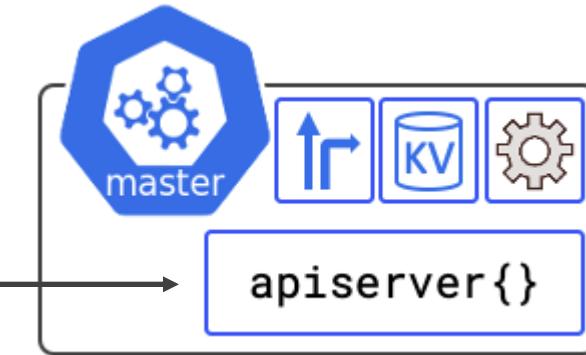
```
name: web  
replicas: 3  
...
```



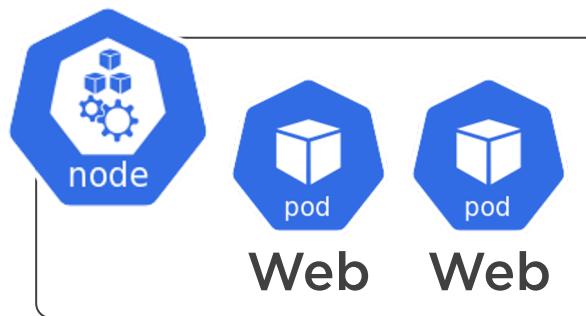
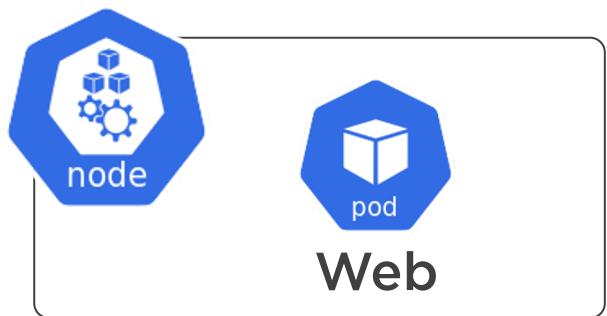
Desired state: 3
Observed state: 2



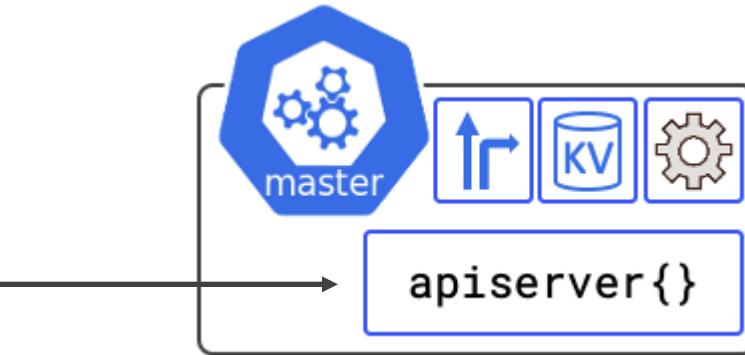
```
name: web  
replicas: 3  
...
```



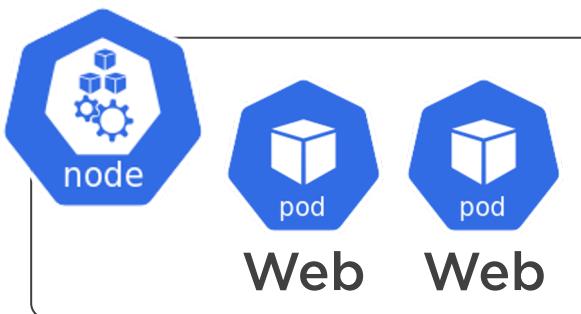
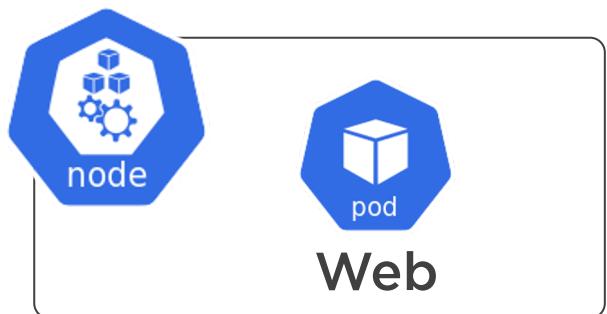
Desired state: 3
Observed state: 2

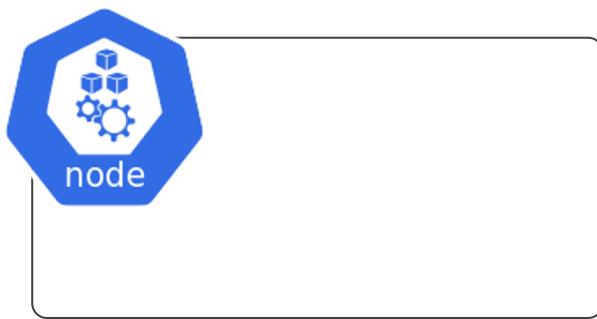
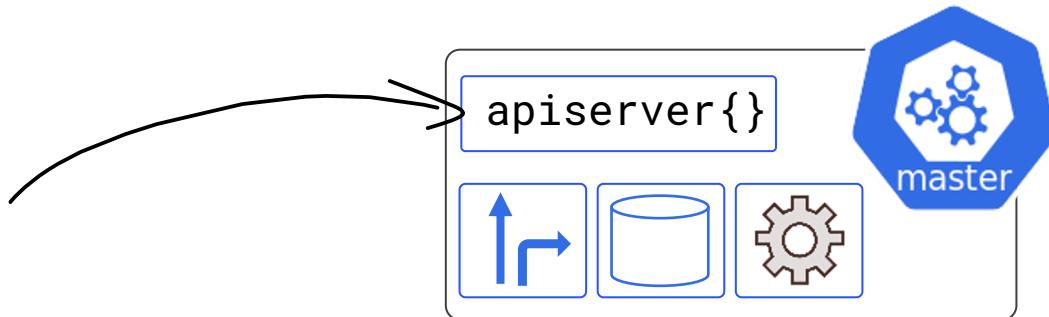


```
name: web  
replicas: 3  
...
```

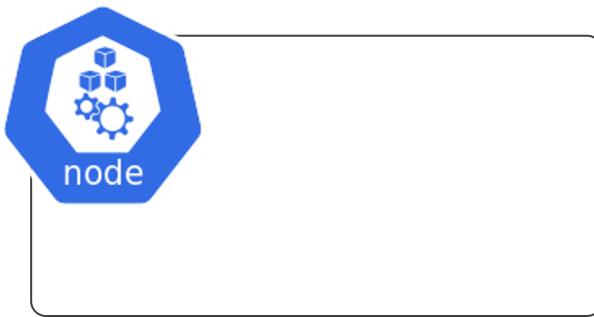
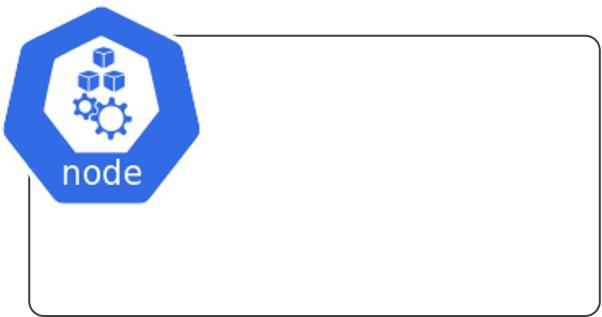
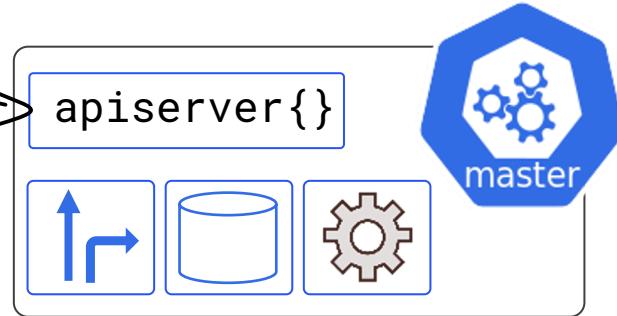


Desired state: 3
Observed state: 3

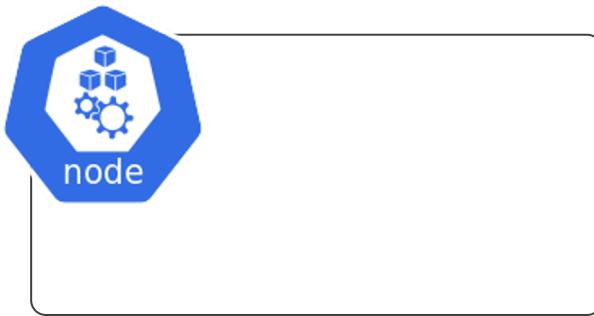
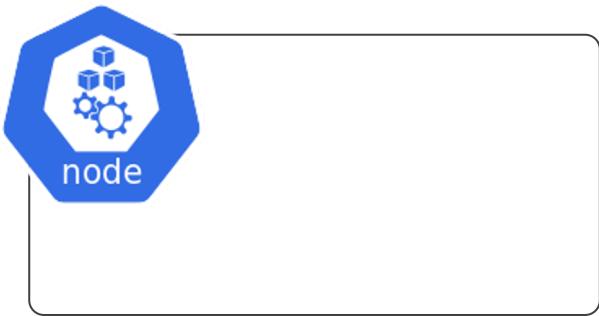
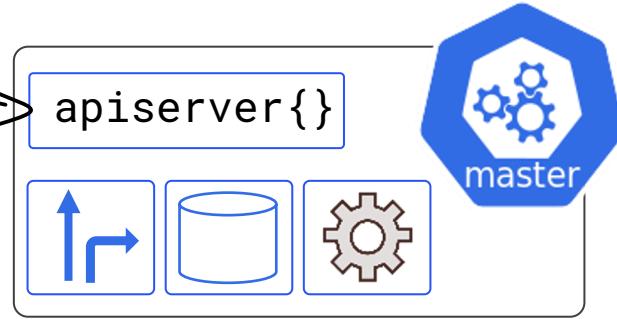


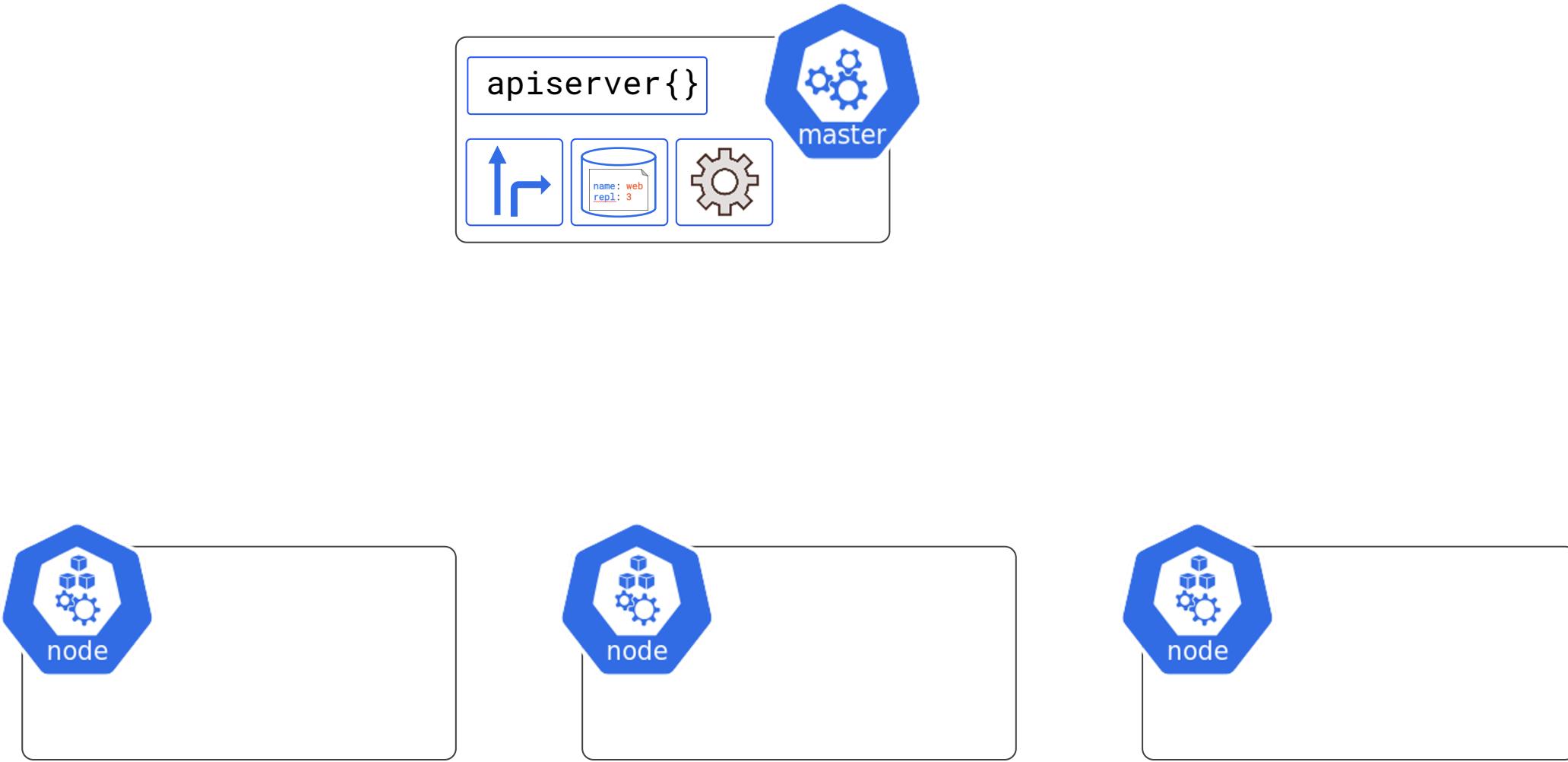


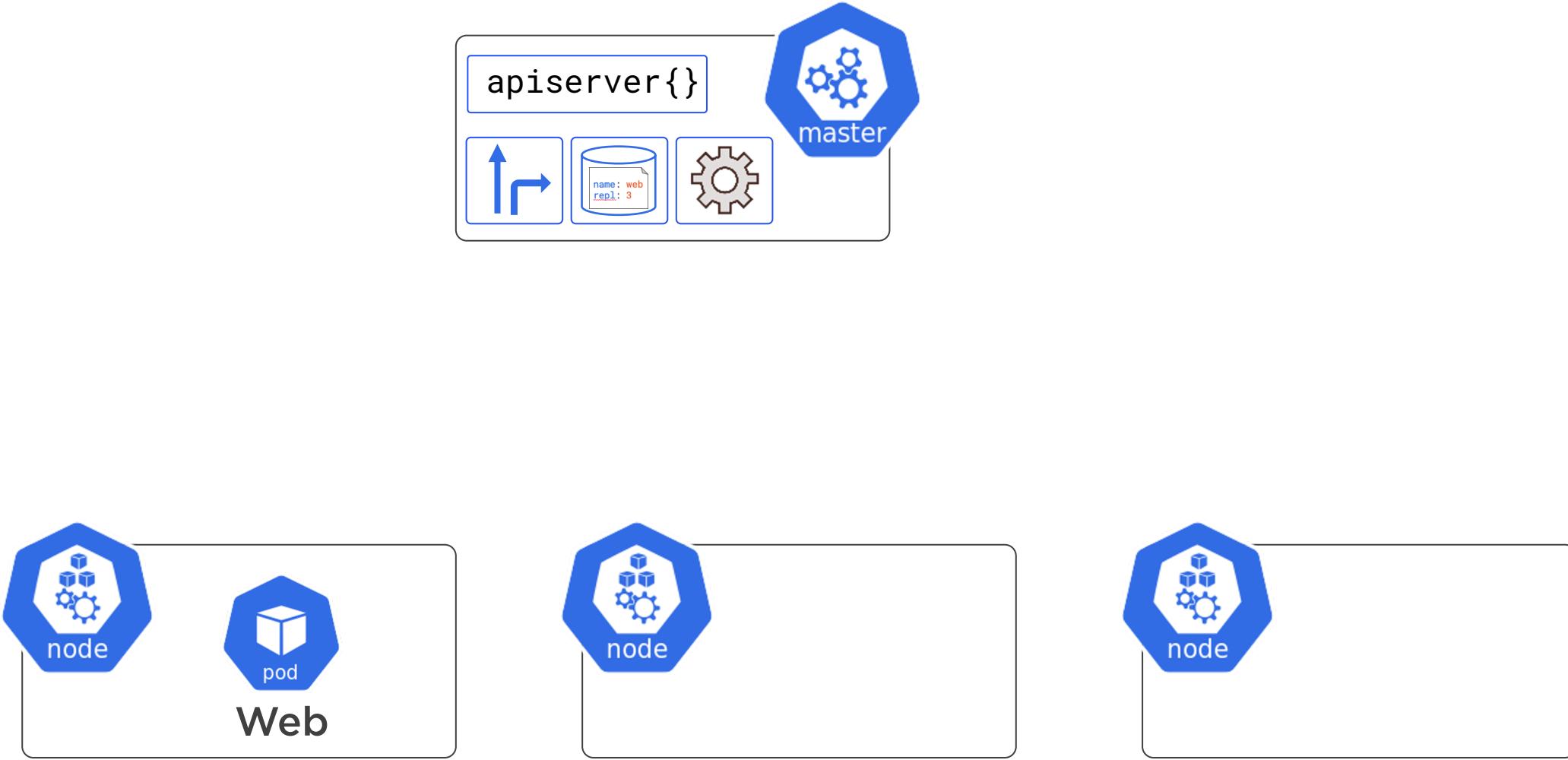
```
name: web  
replicas: 3  
...
```

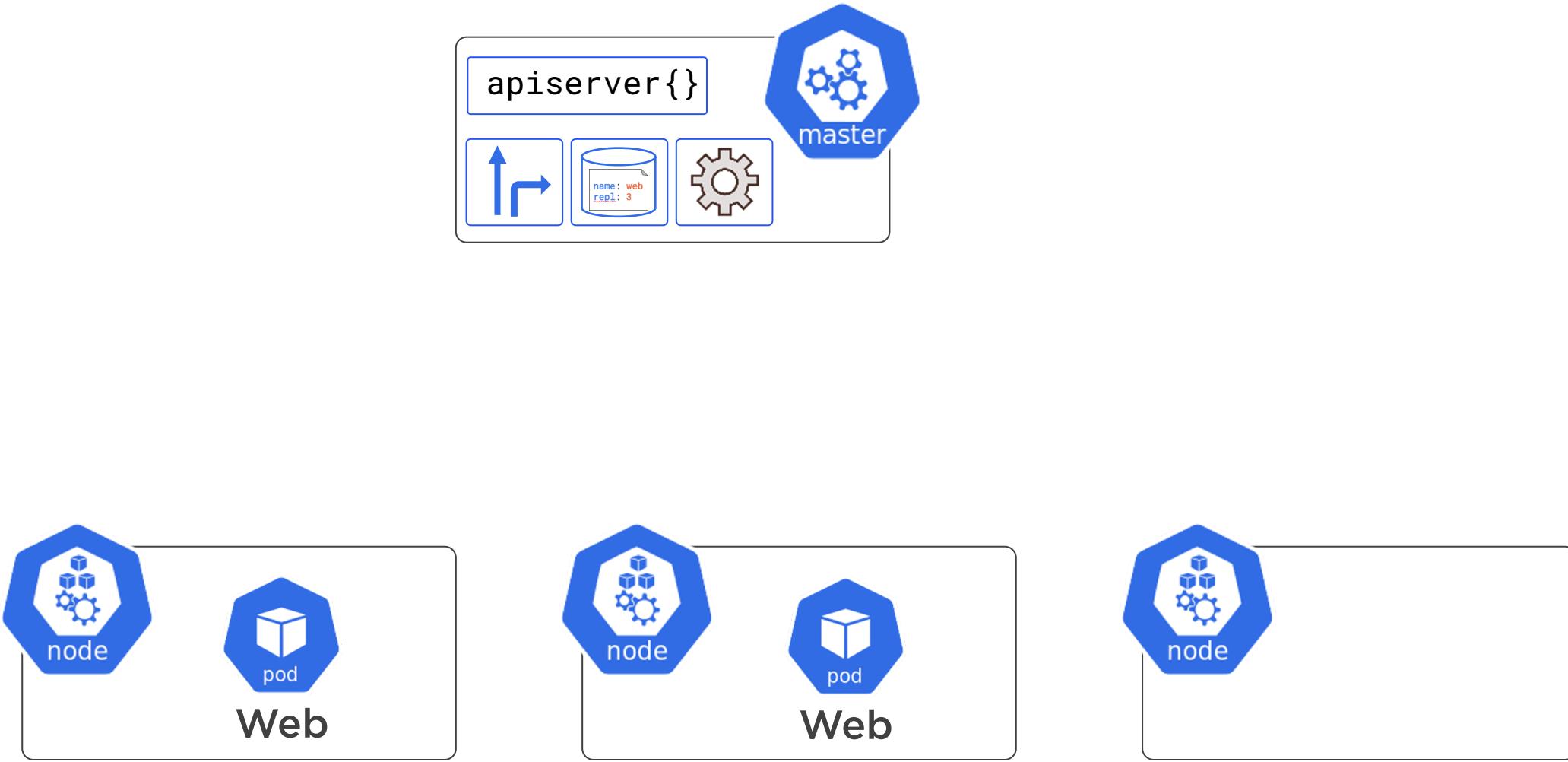


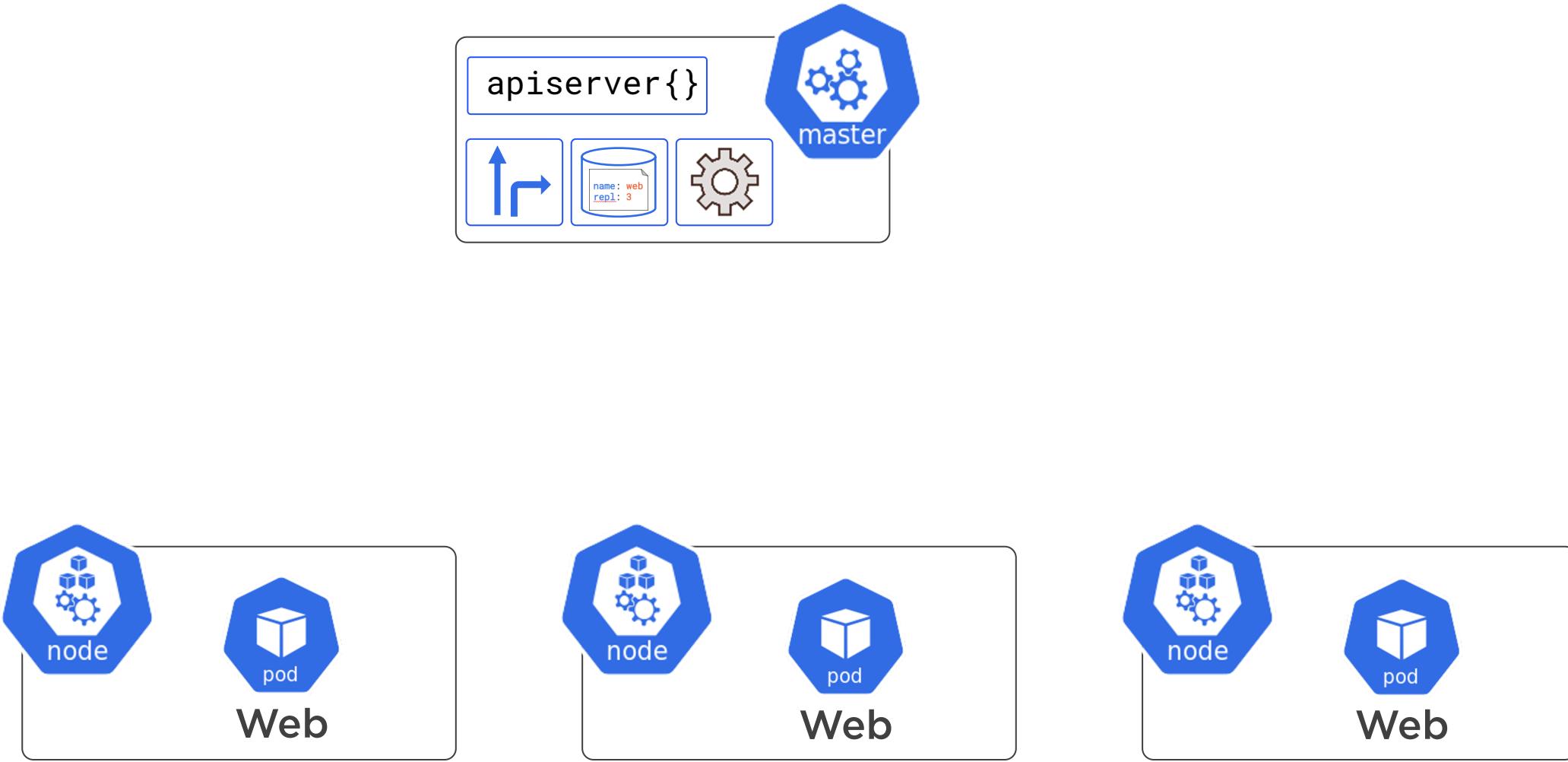
```
name: web  
replicas: 3  
...
```

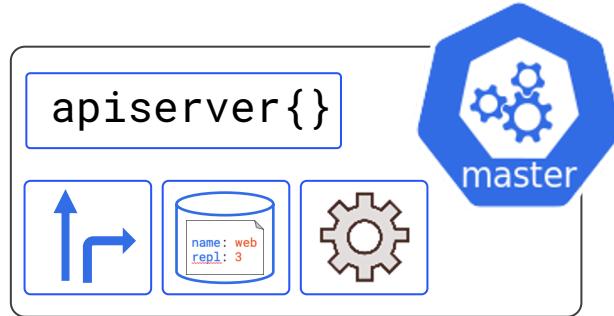




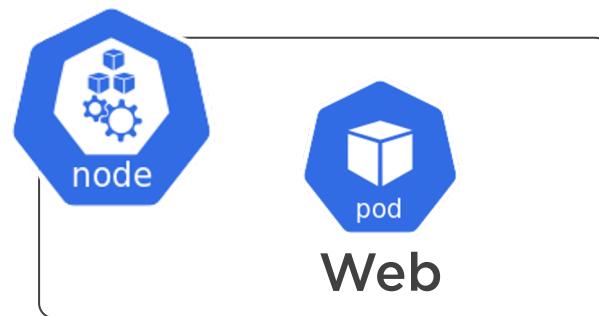
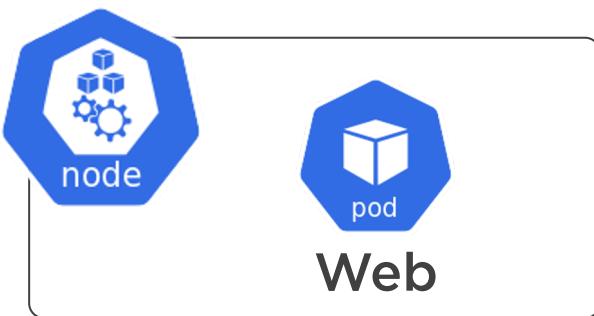
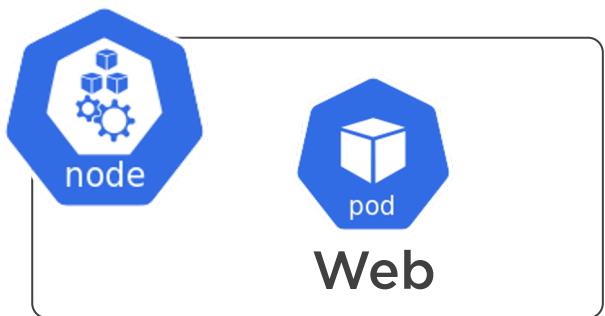


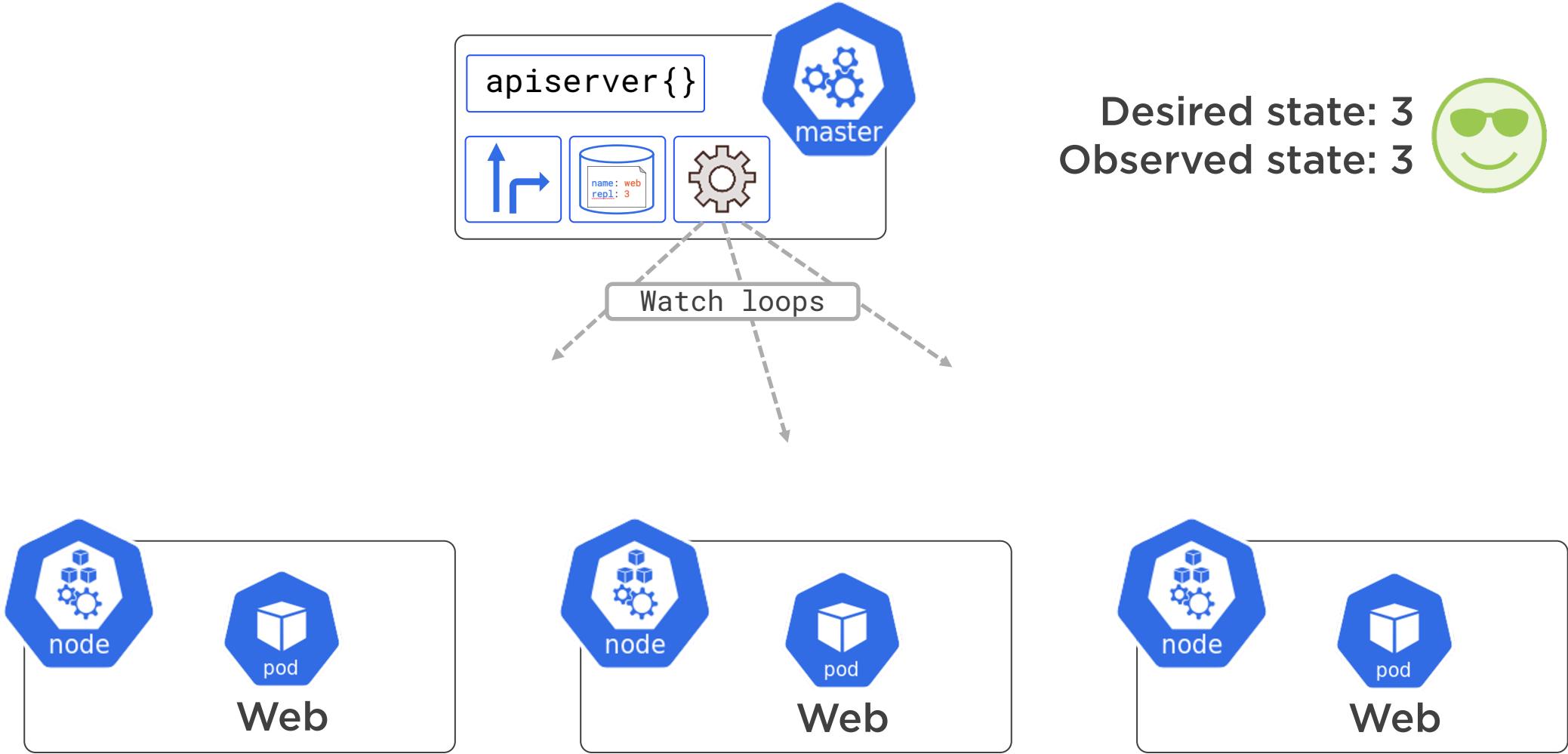






Desired state: 3
Observed state: 3



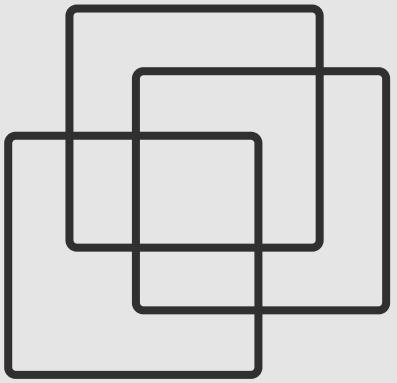


Up Next:
The Mighty Pod

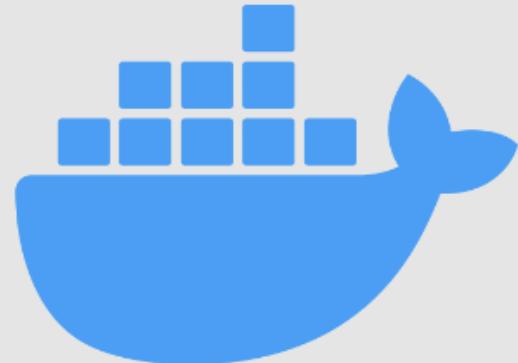


The Mighty Pod





Virtual Machine

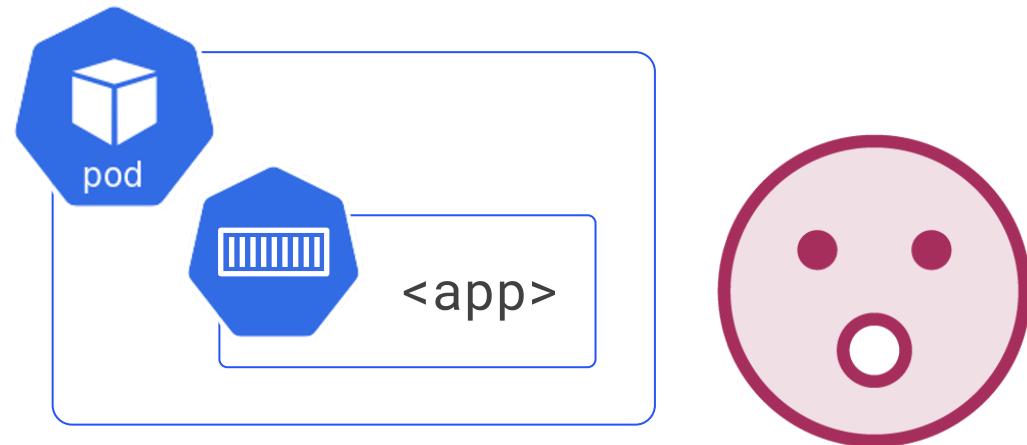


Container



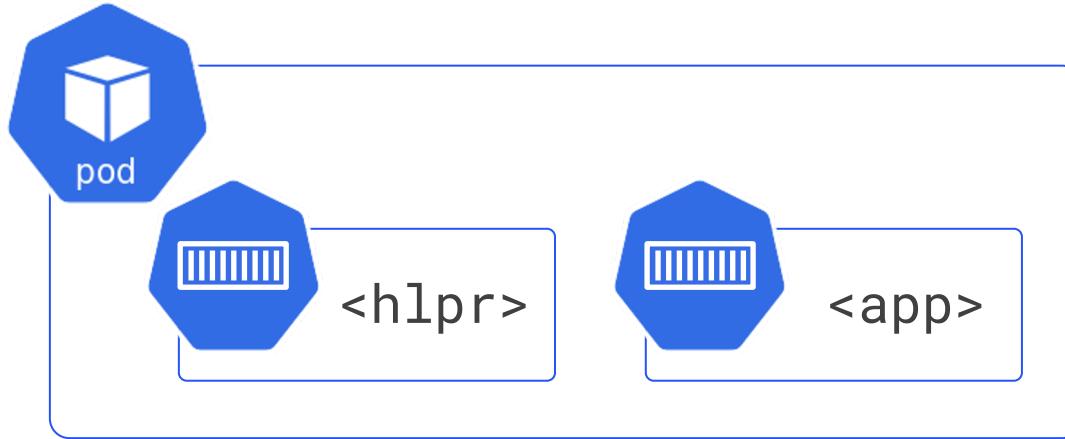
Pod





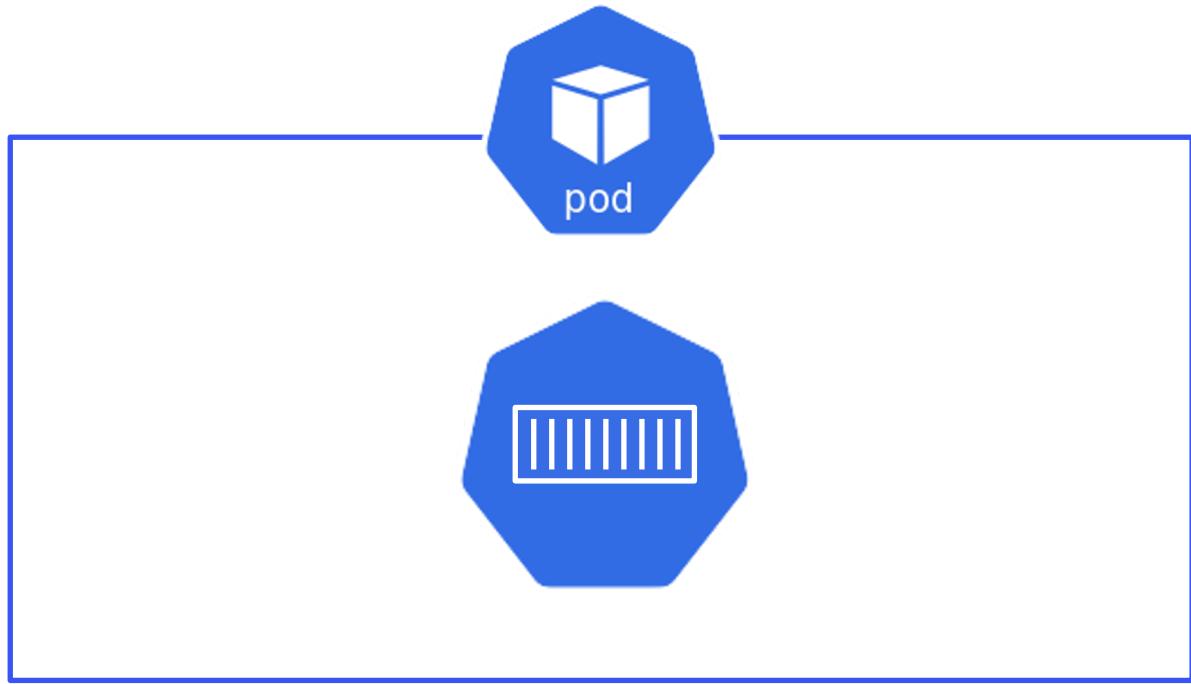


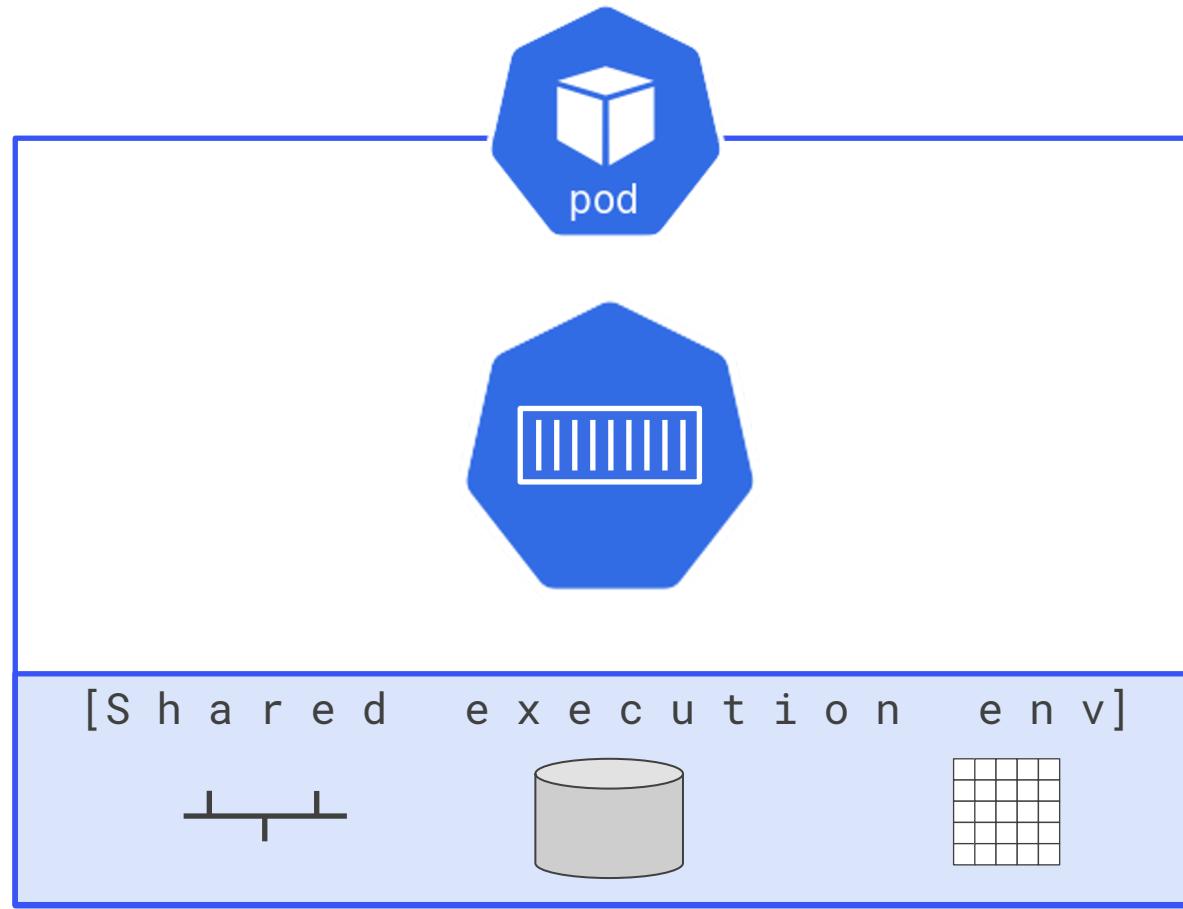


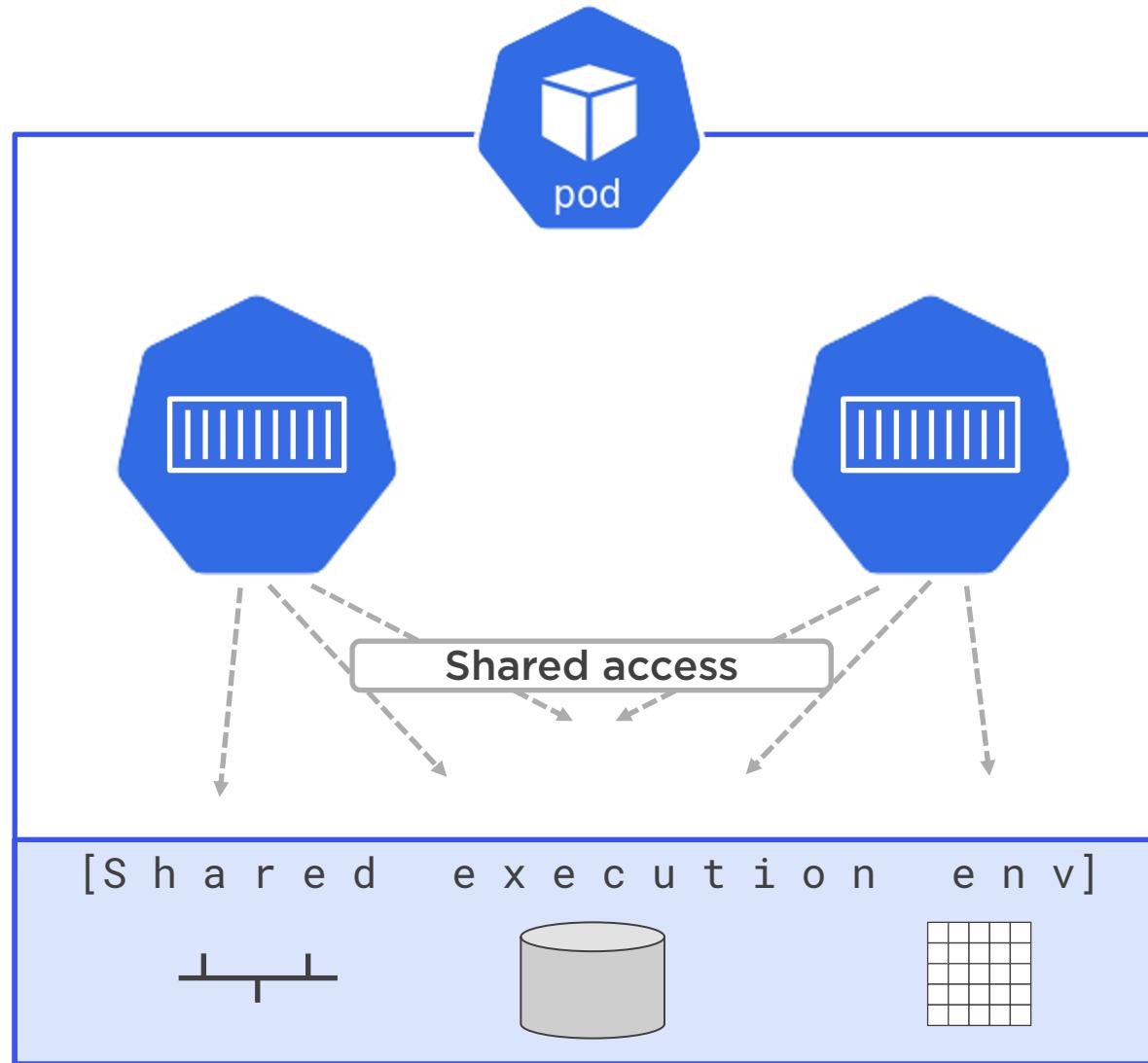


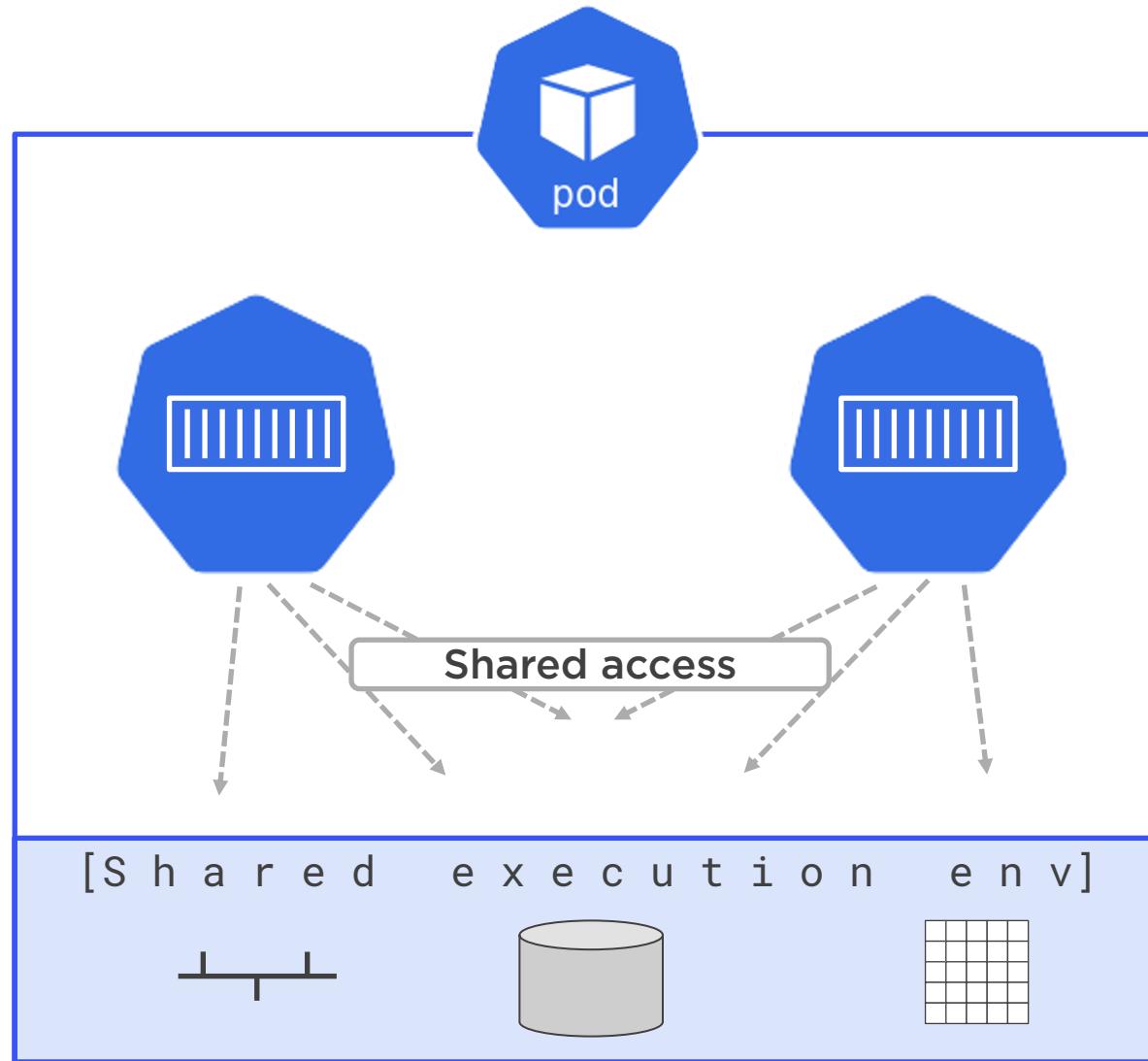
Kubernetes for Developers: Integrating Volumes and Using Multi-container Pods

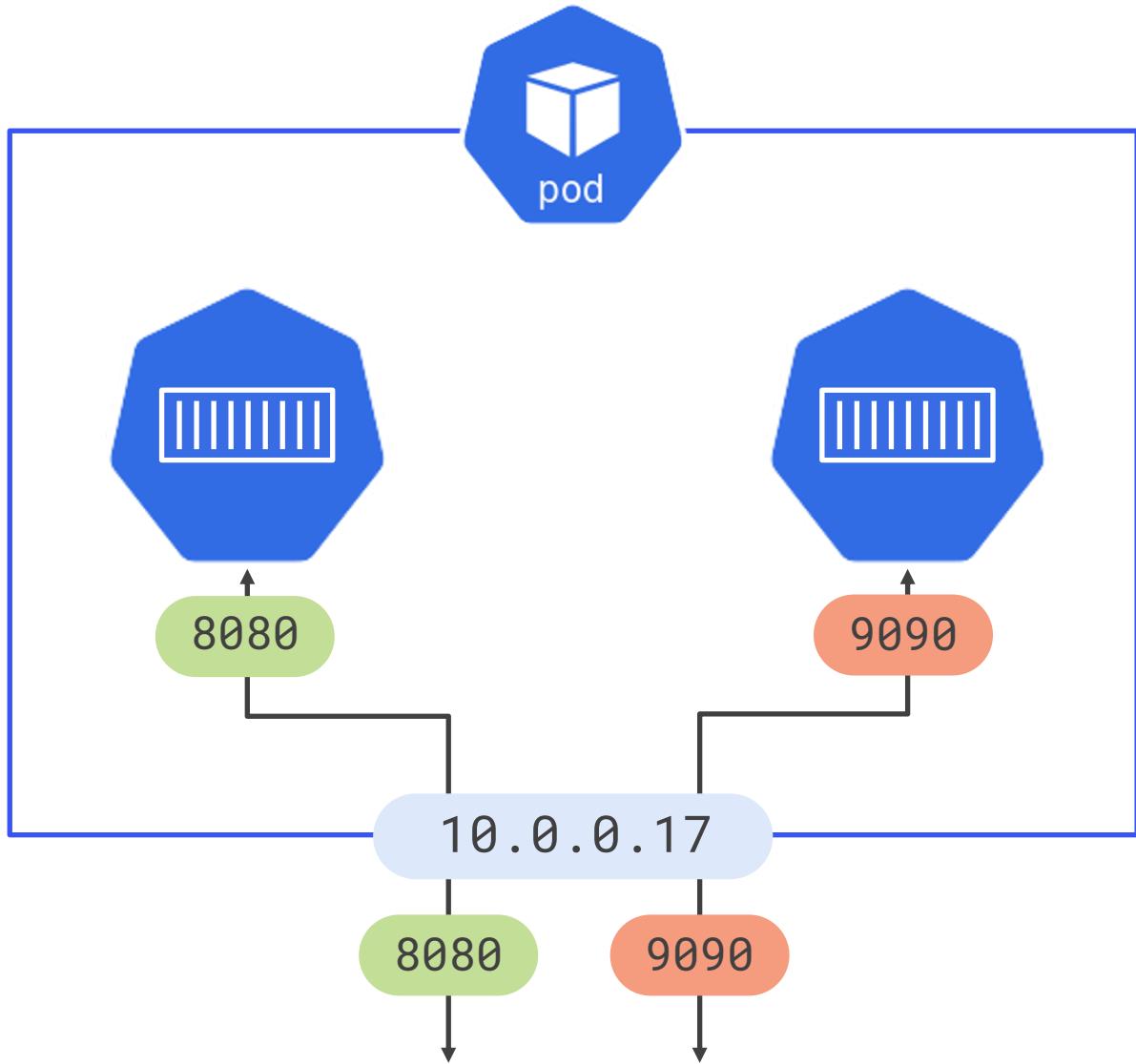


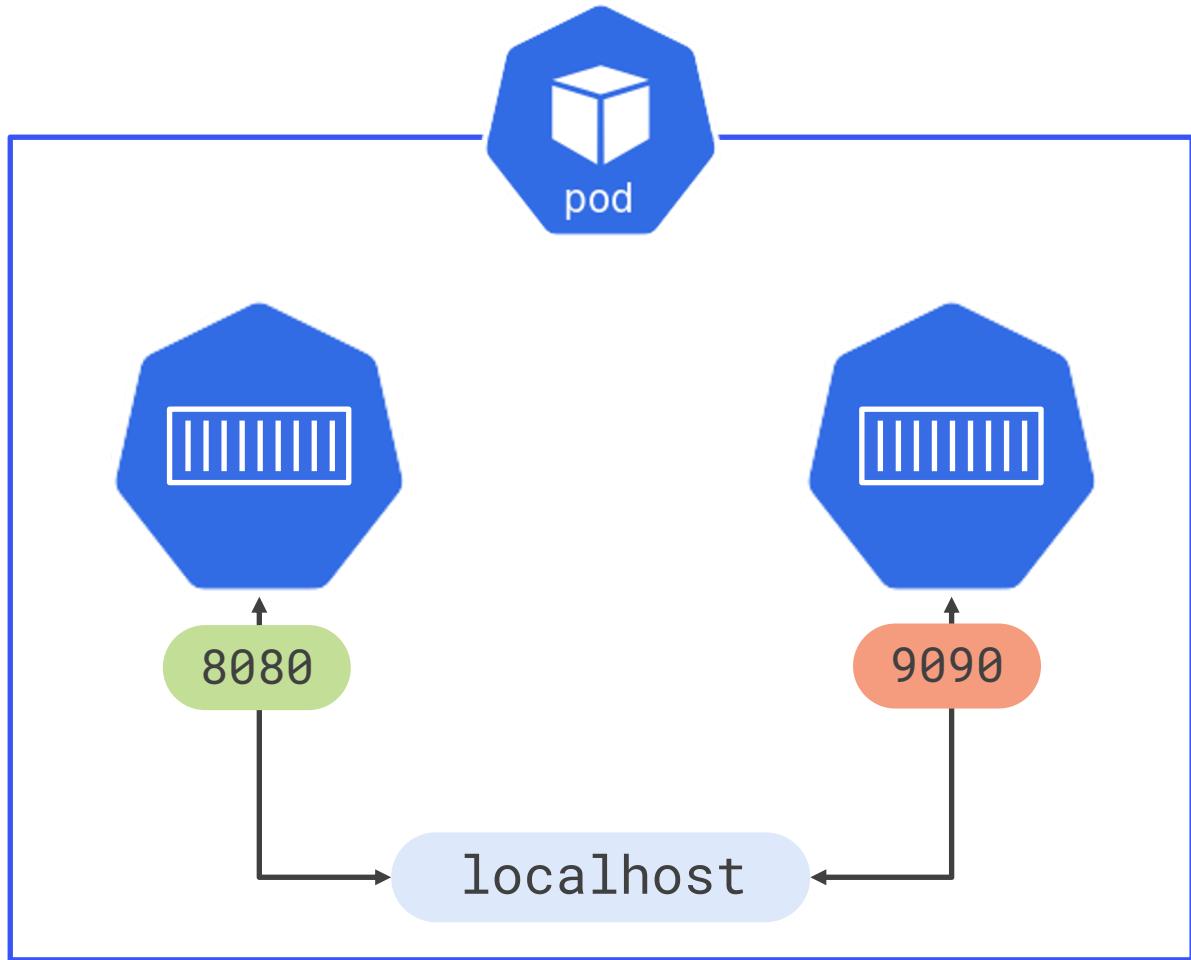


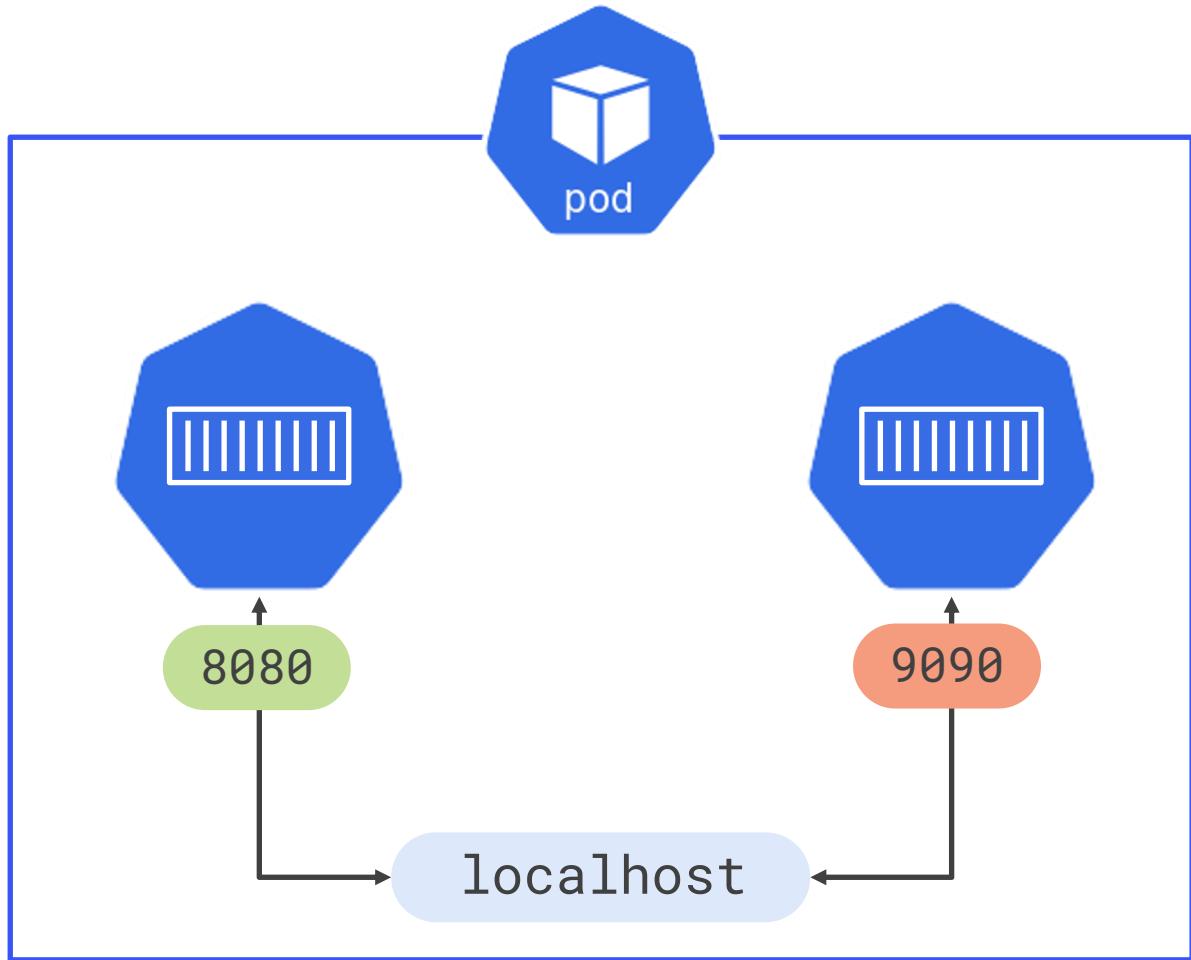


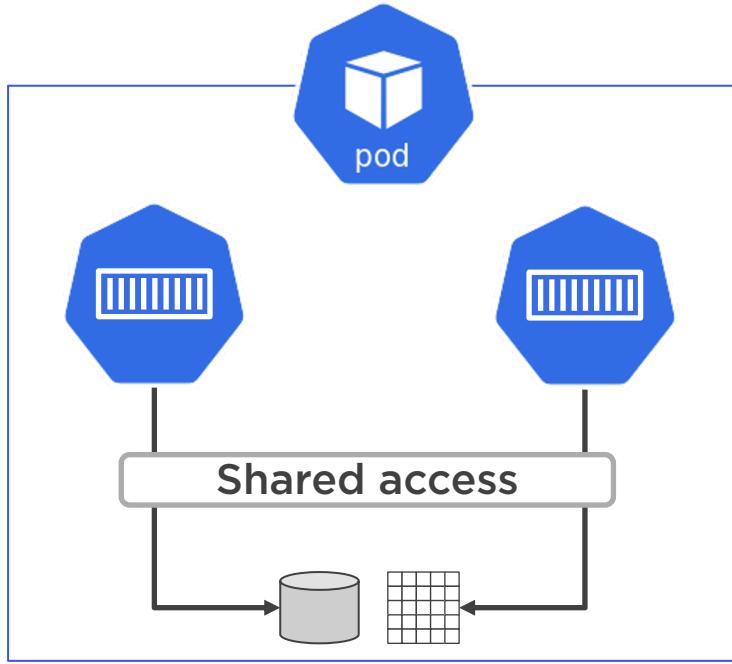






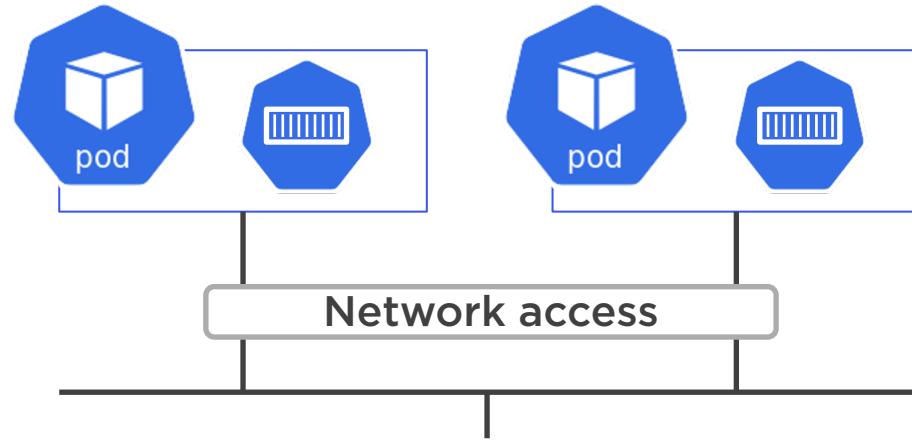






Tightly coupled

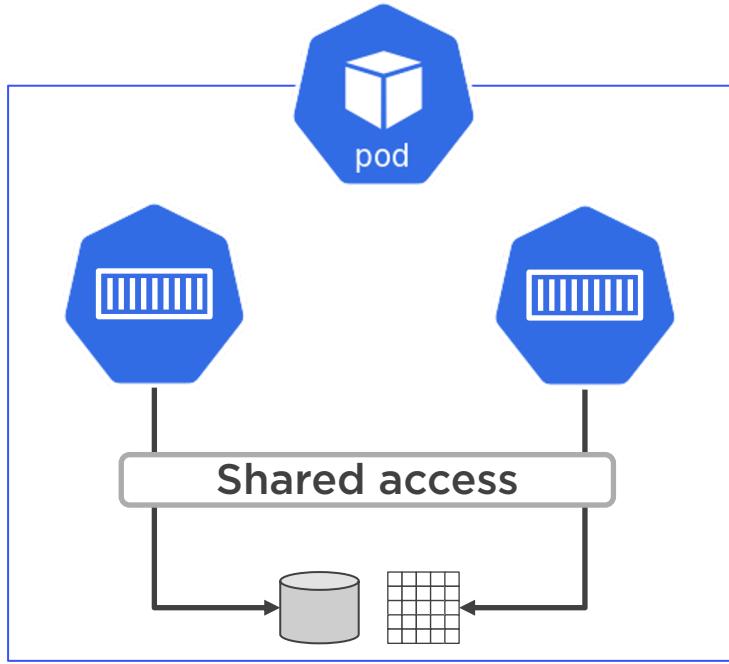
When two containers (app services) **absolutely need** to share vols, memory etc.



Loosely coupled

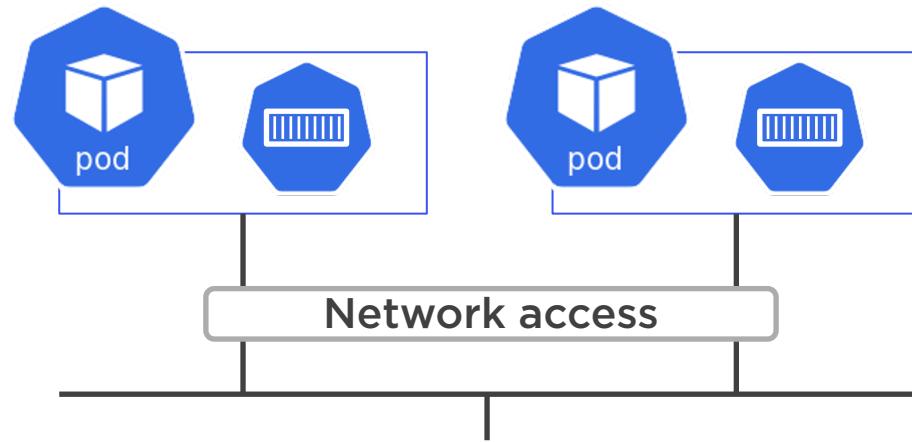
When two containers (app services) **don't absolutely need** to share resources





Tightly coupled

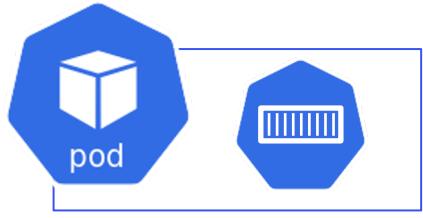
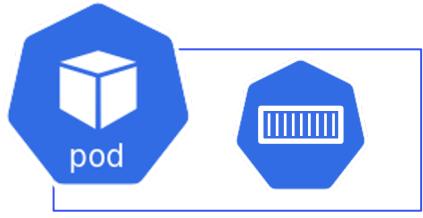
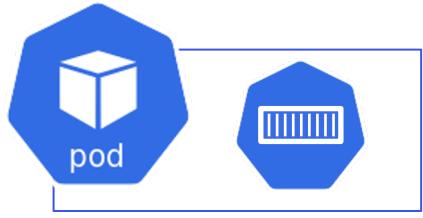
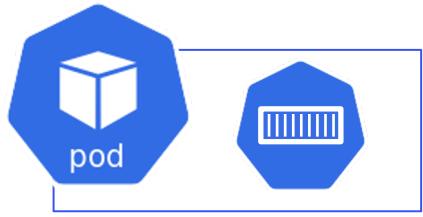
When two containers (app services) absolutely need to share vols, memory etc.



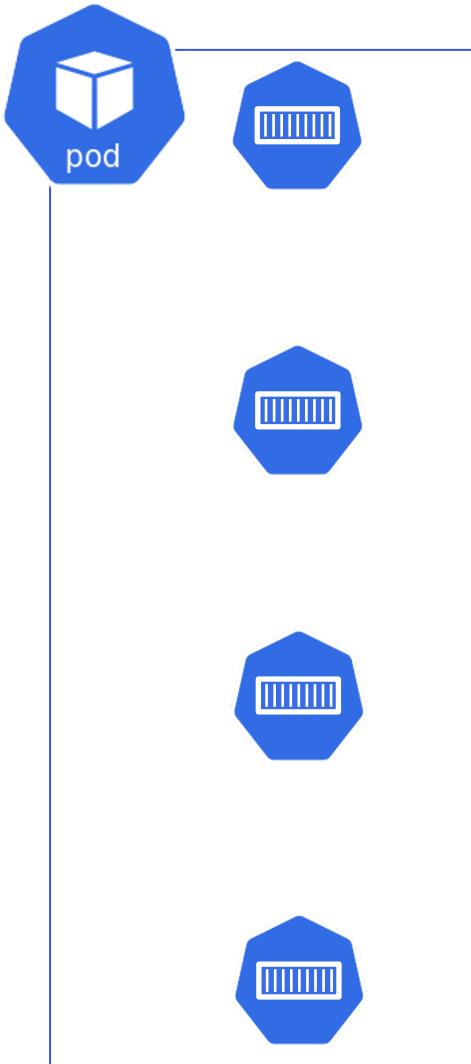
Loosely coupled

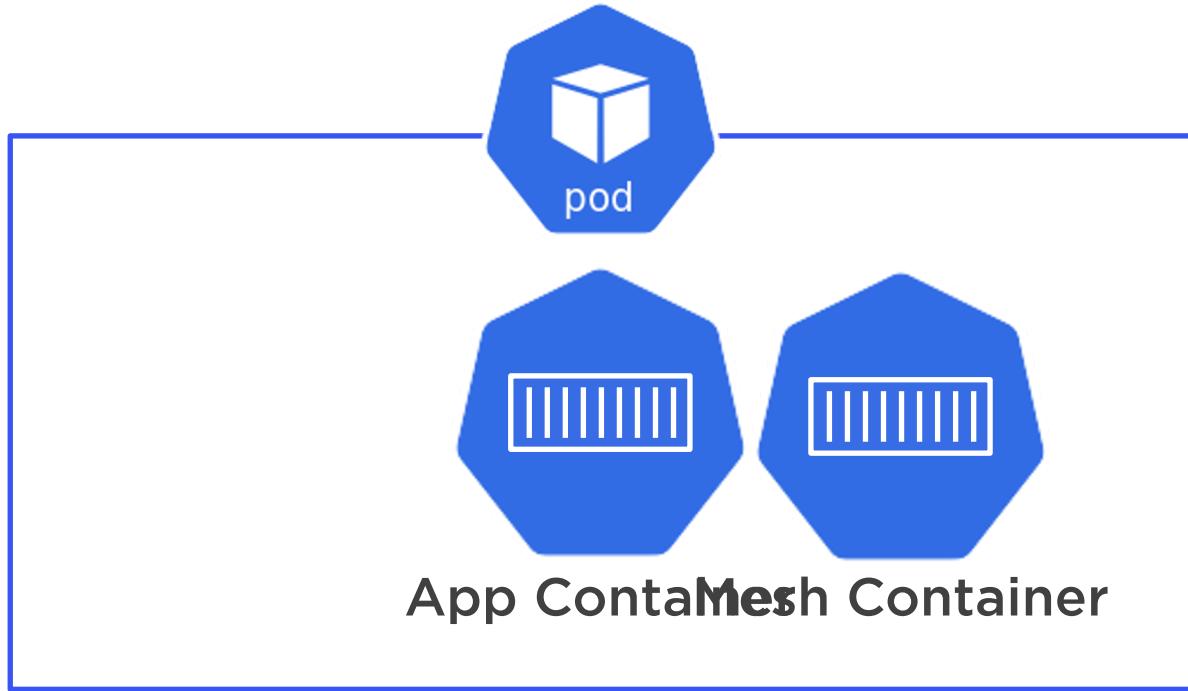
When two containers (app services) don't absolutely need to share resources

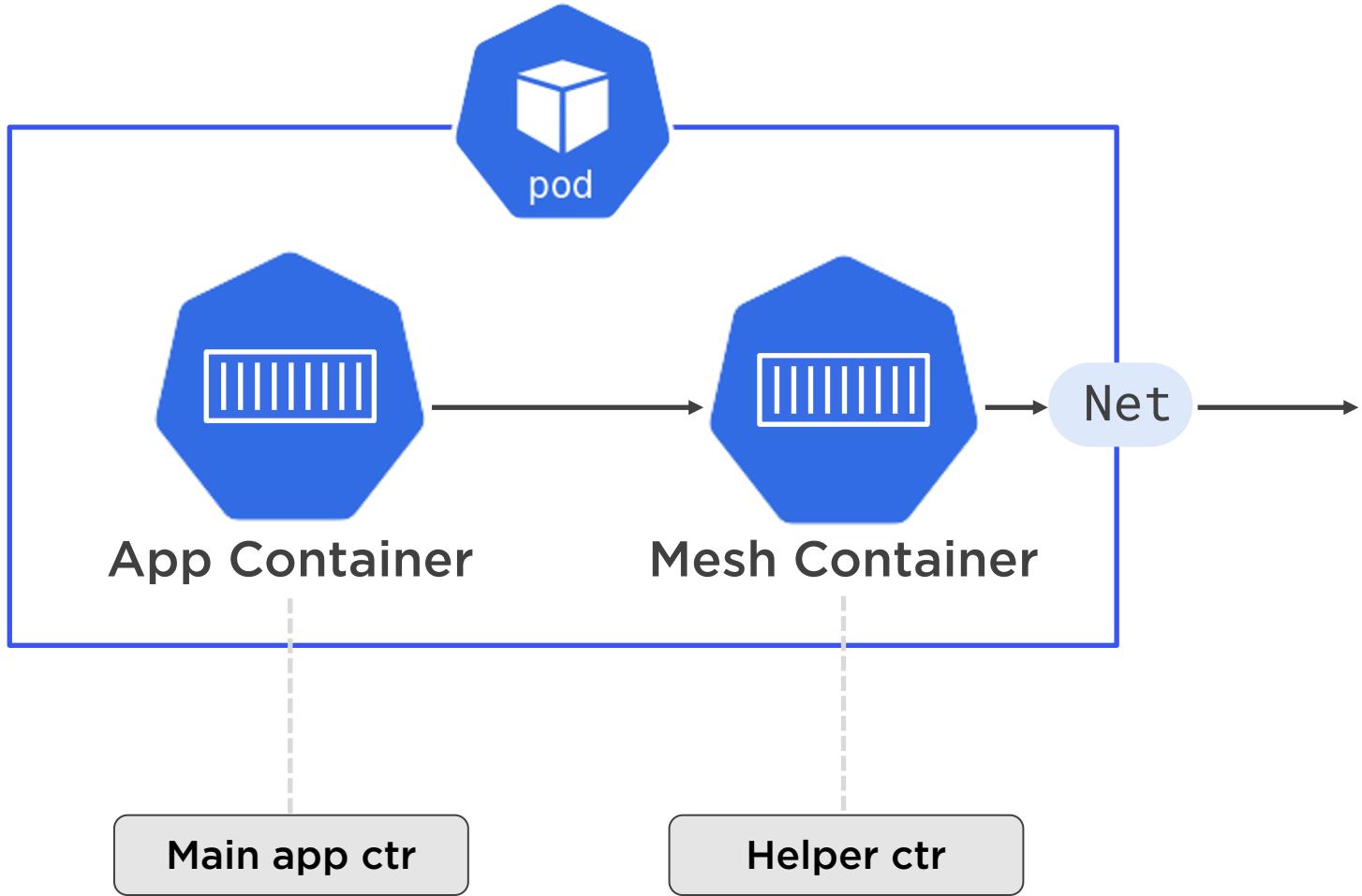


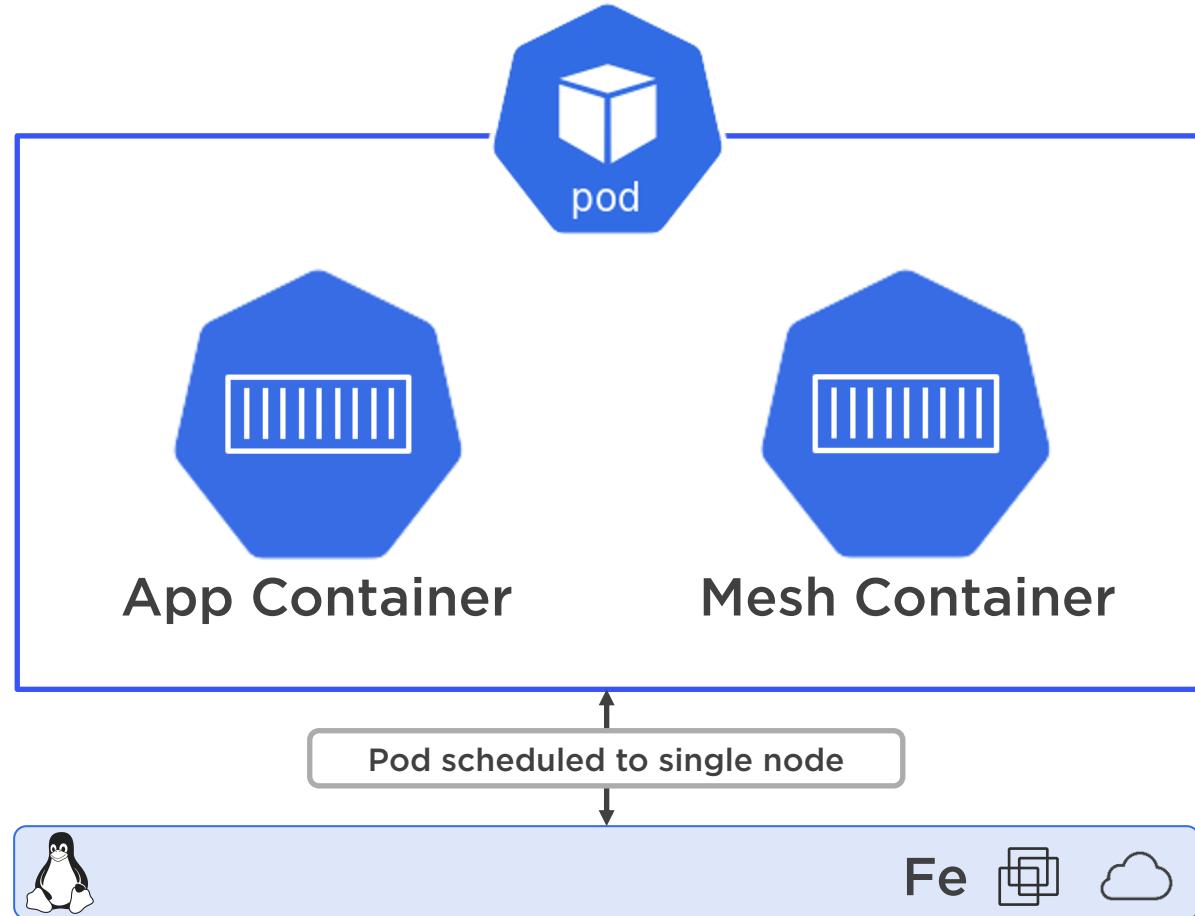


Scaling

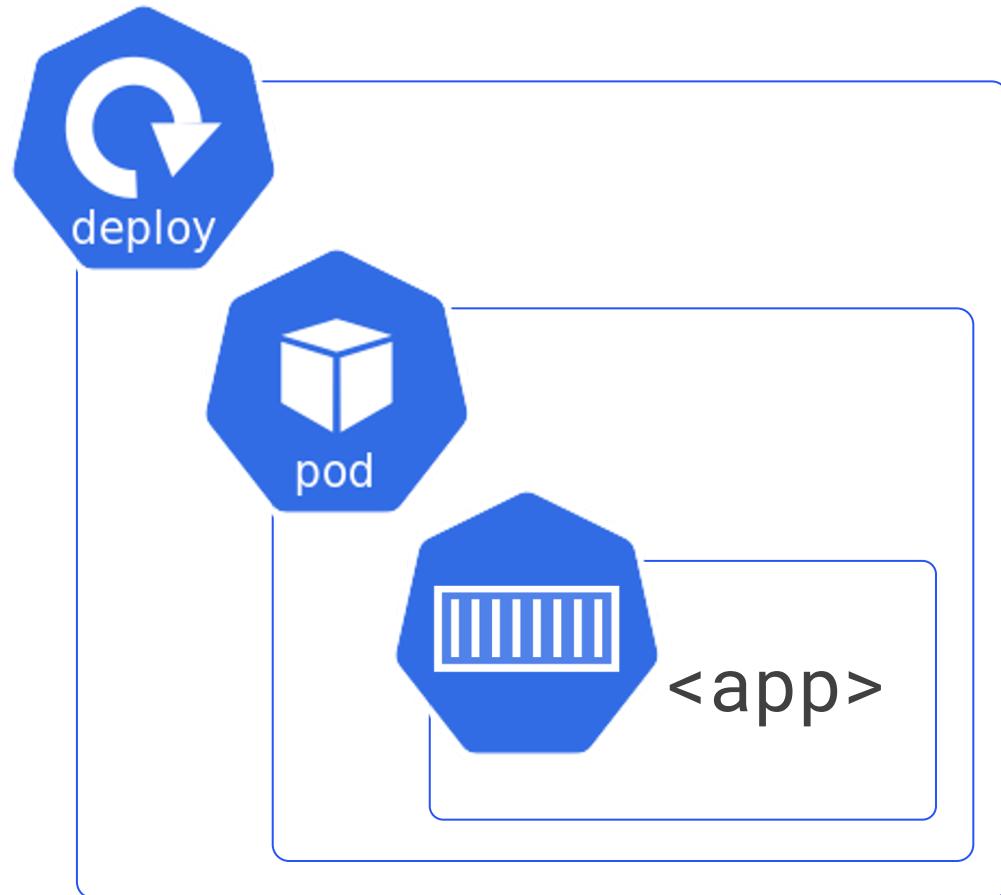


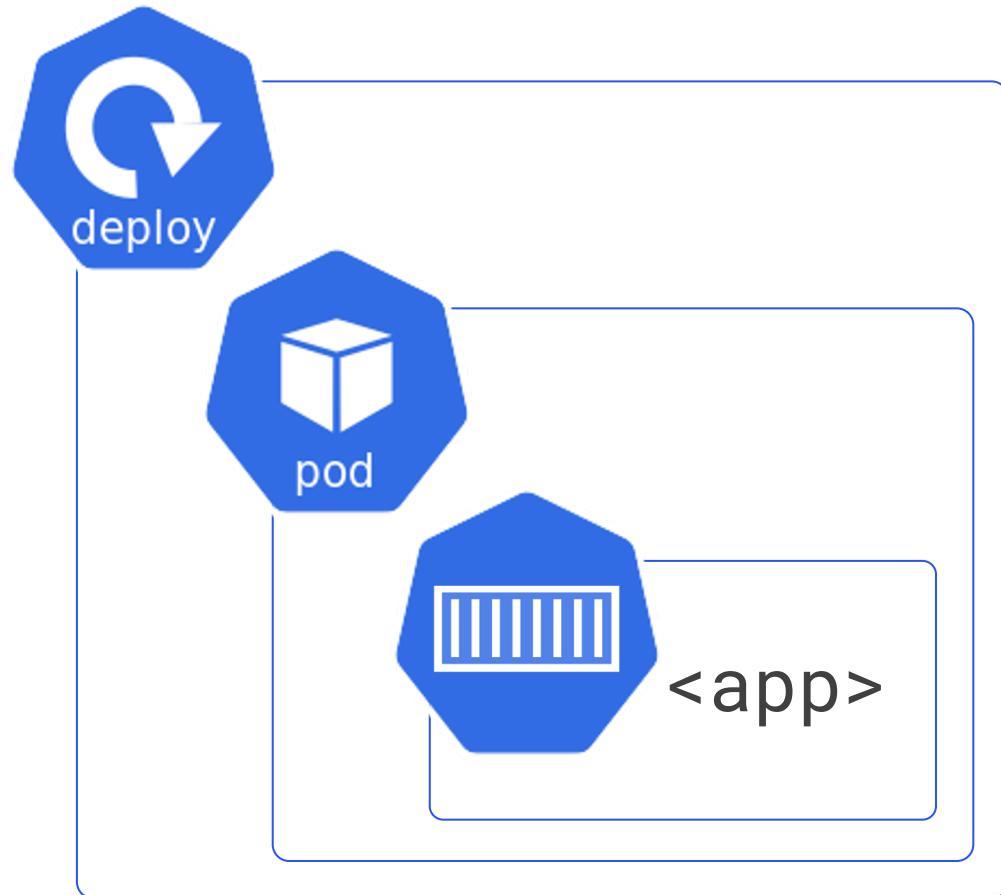














Annotations

Labels

Policies

Resources

Co-scheduling containers

...

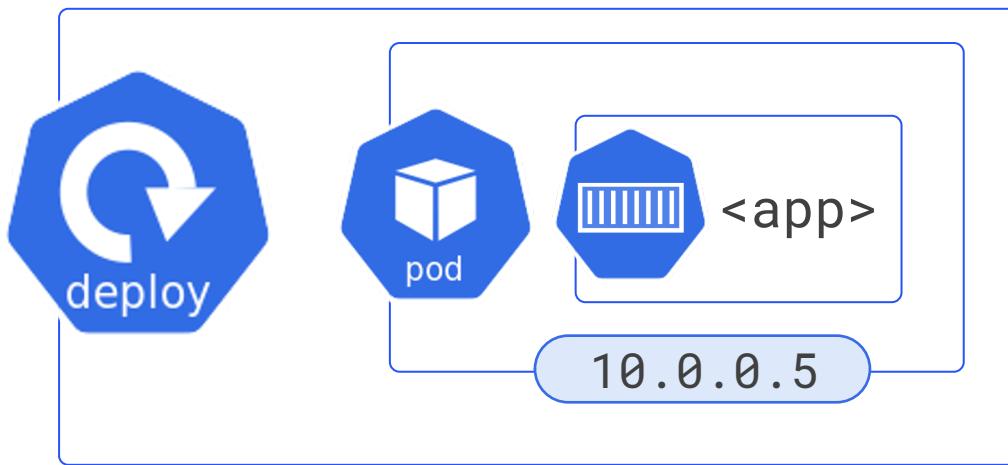


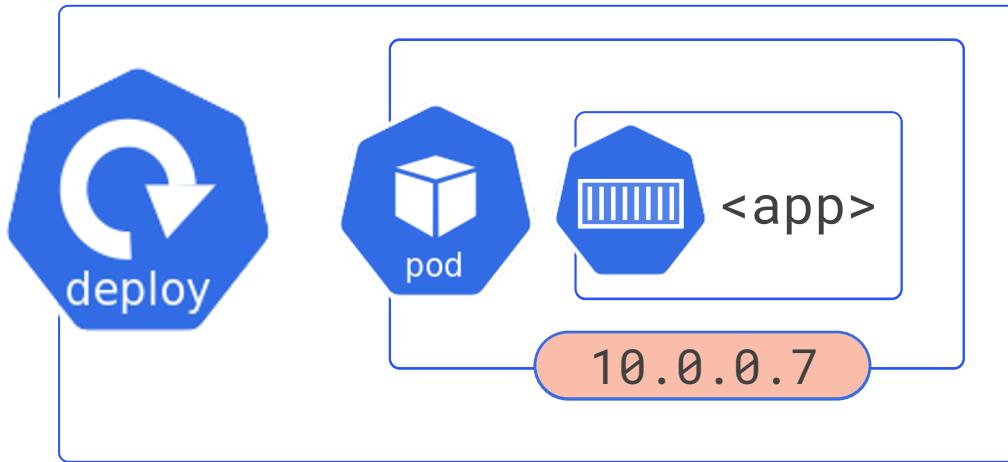
Up Next:
Stable Networking with Services

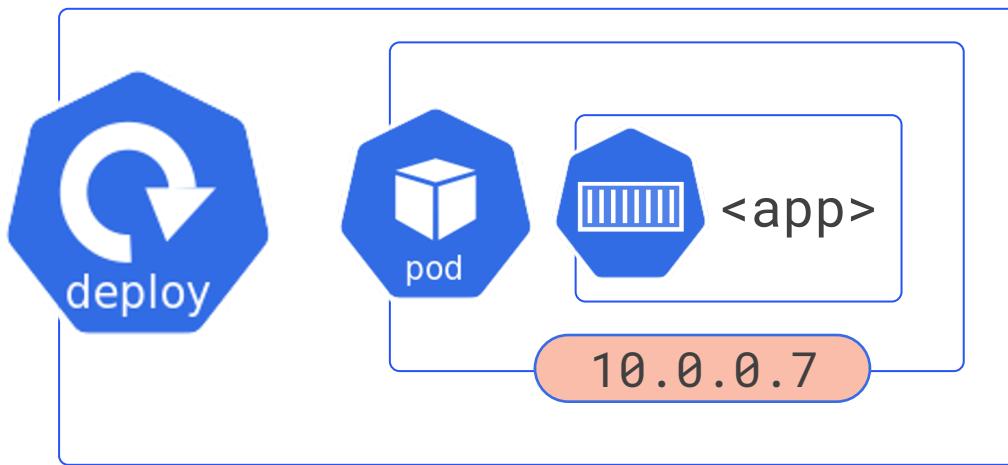


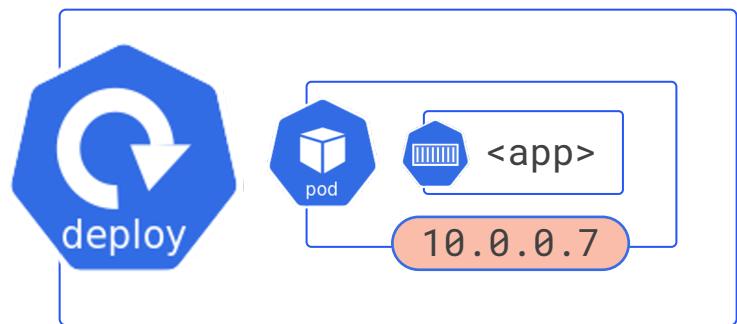
Stable Networking with Services

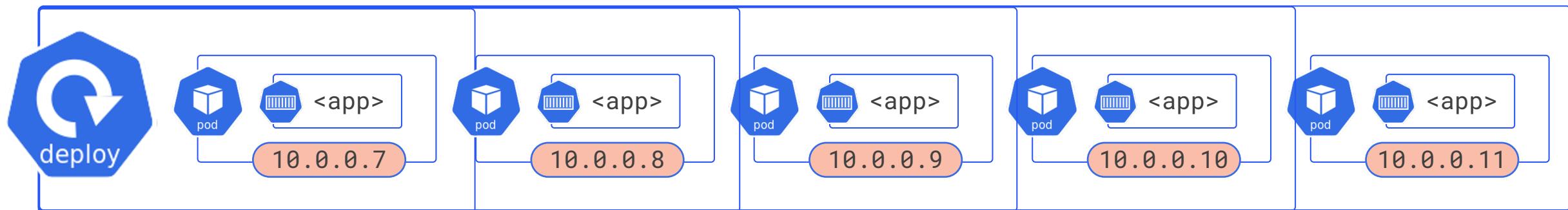


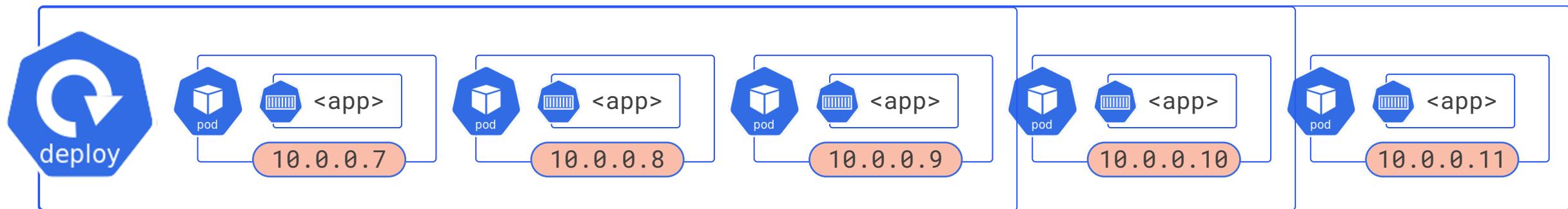


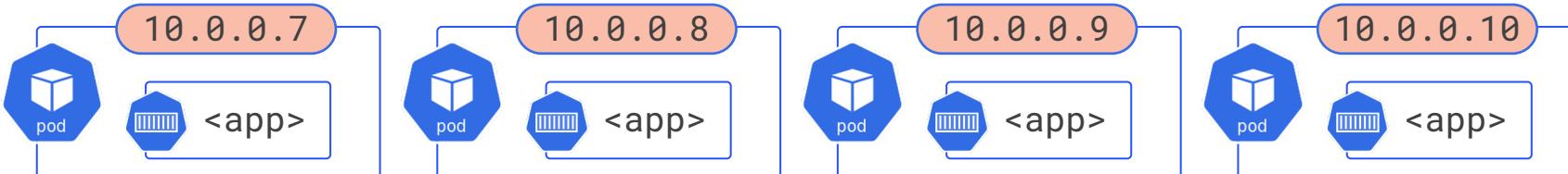
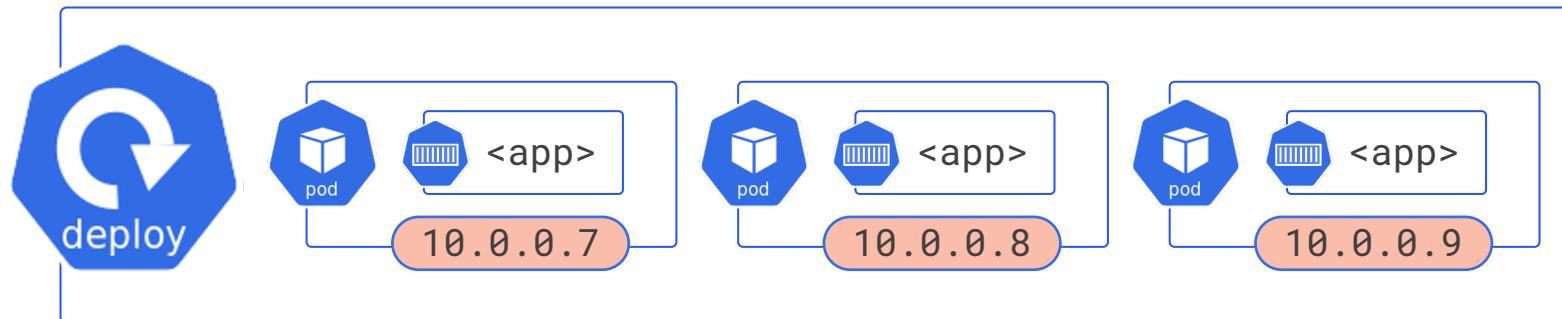


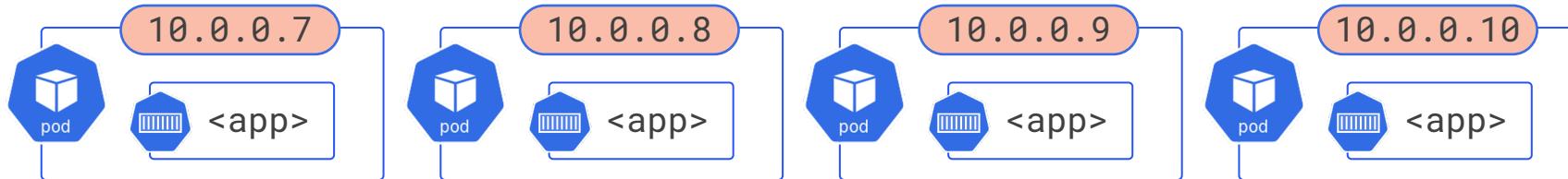


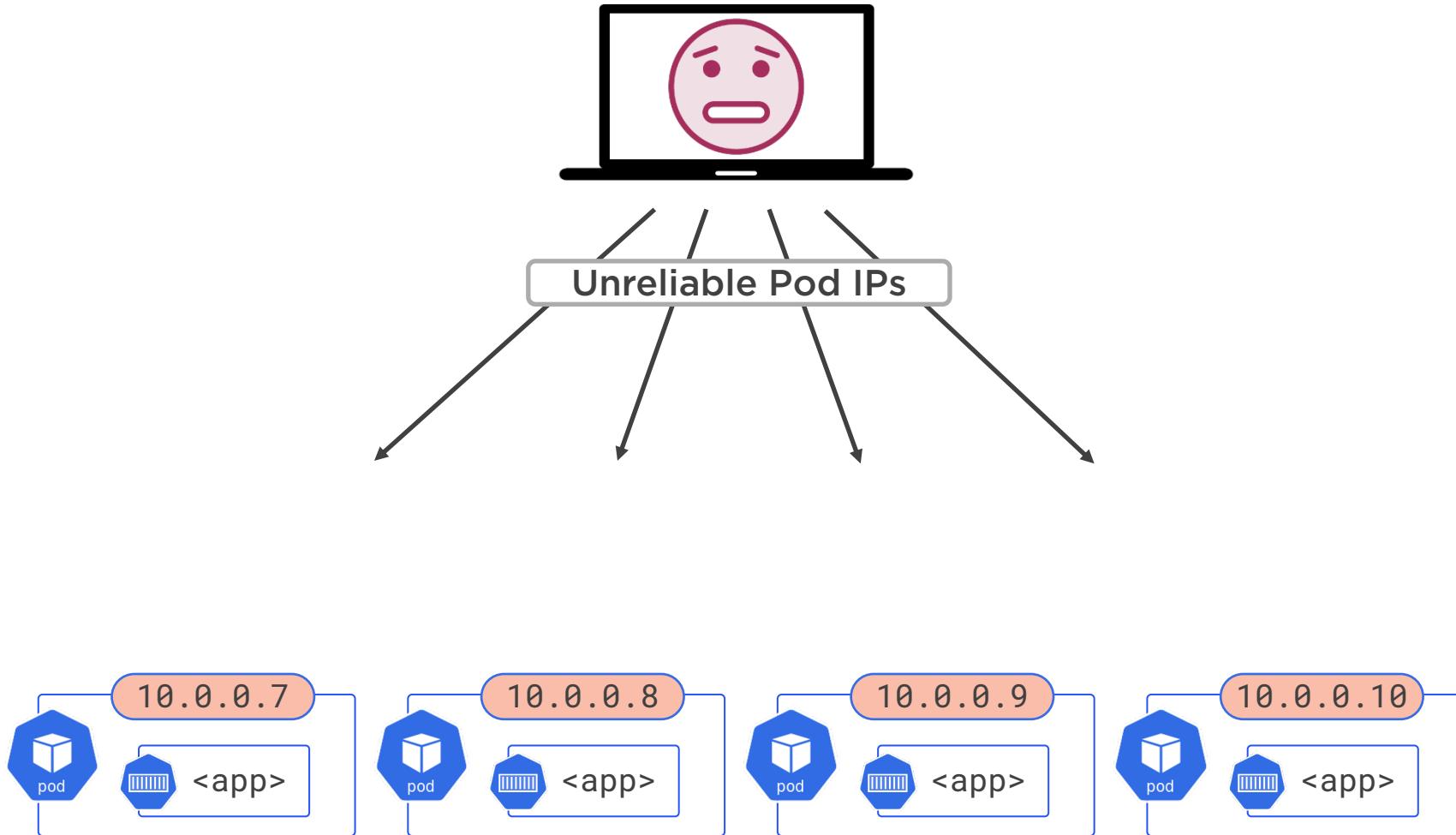


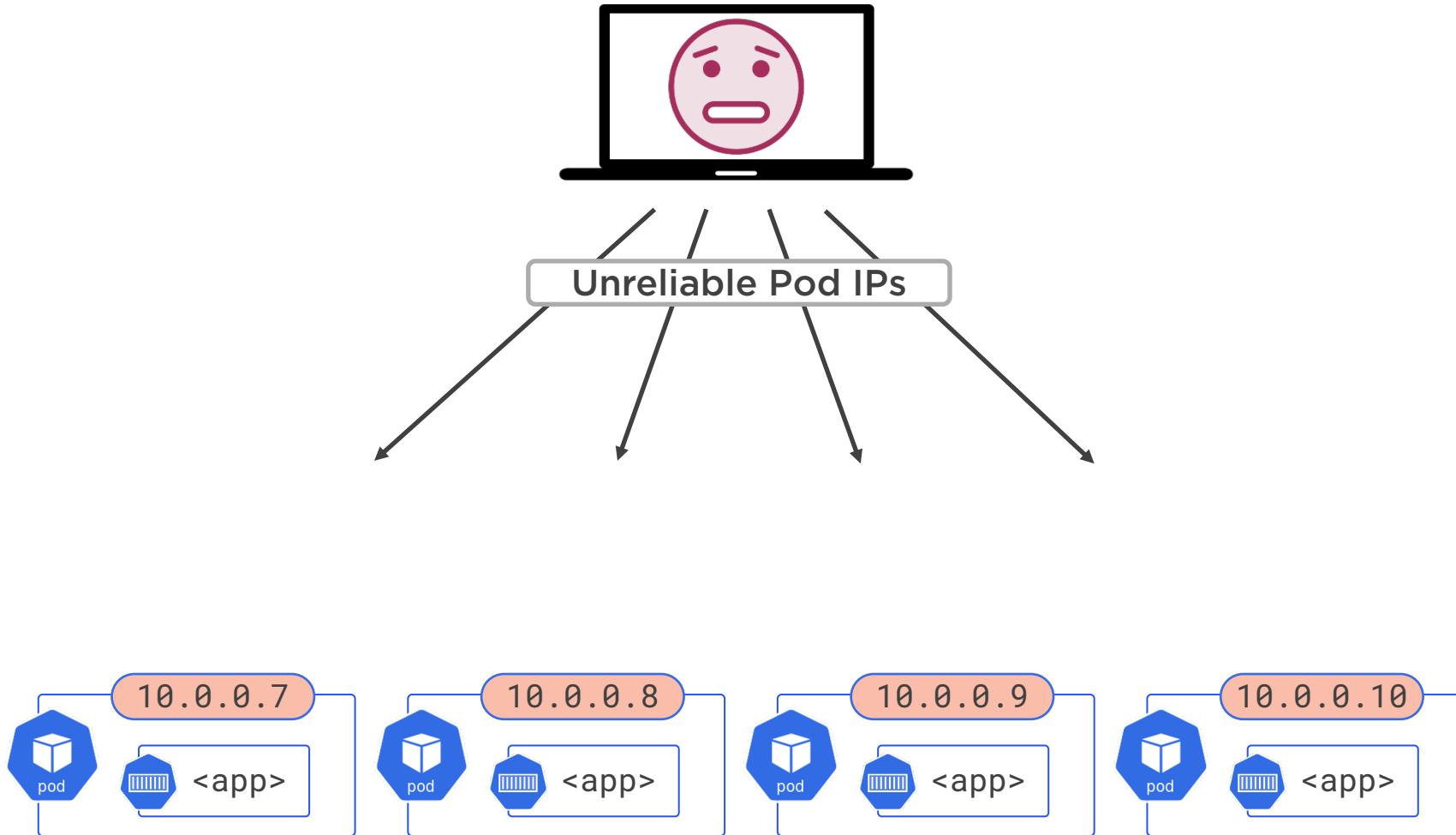


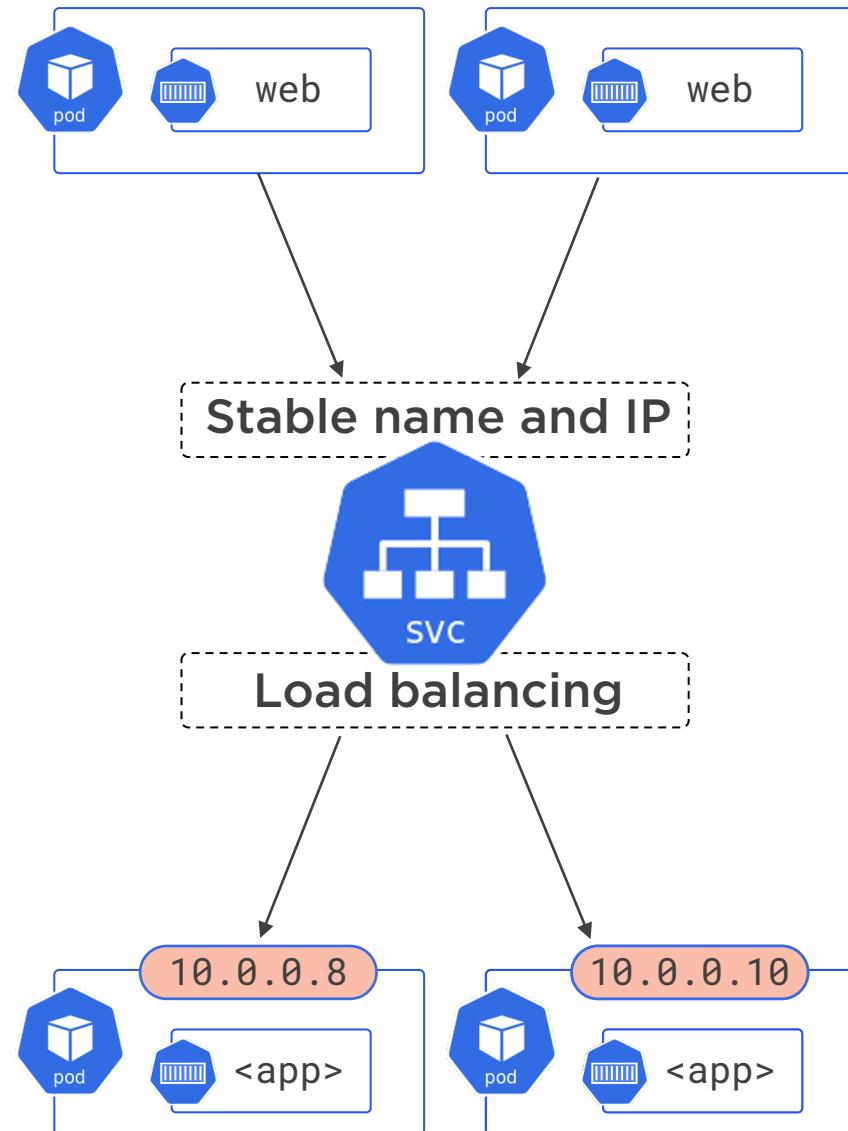


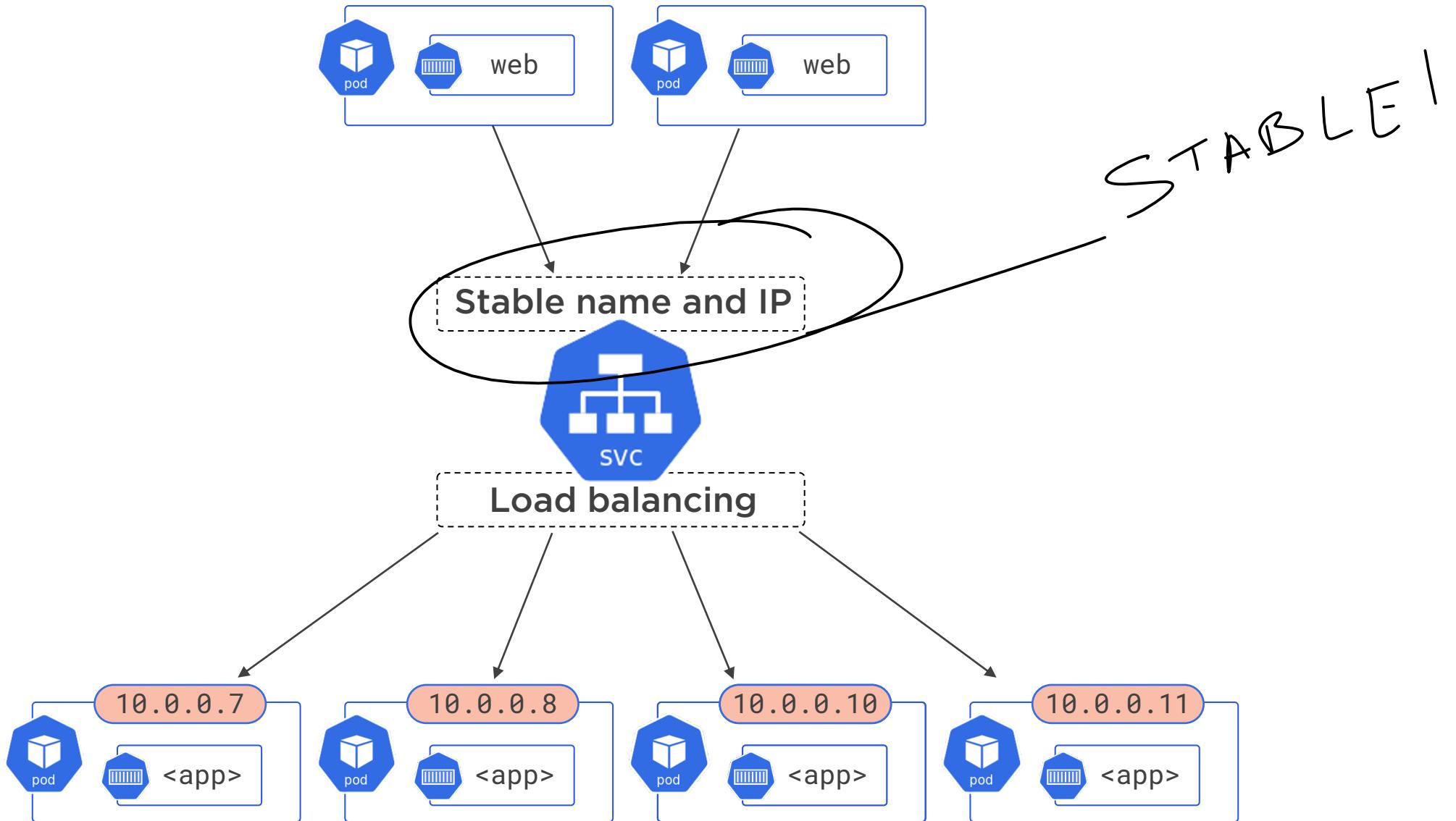


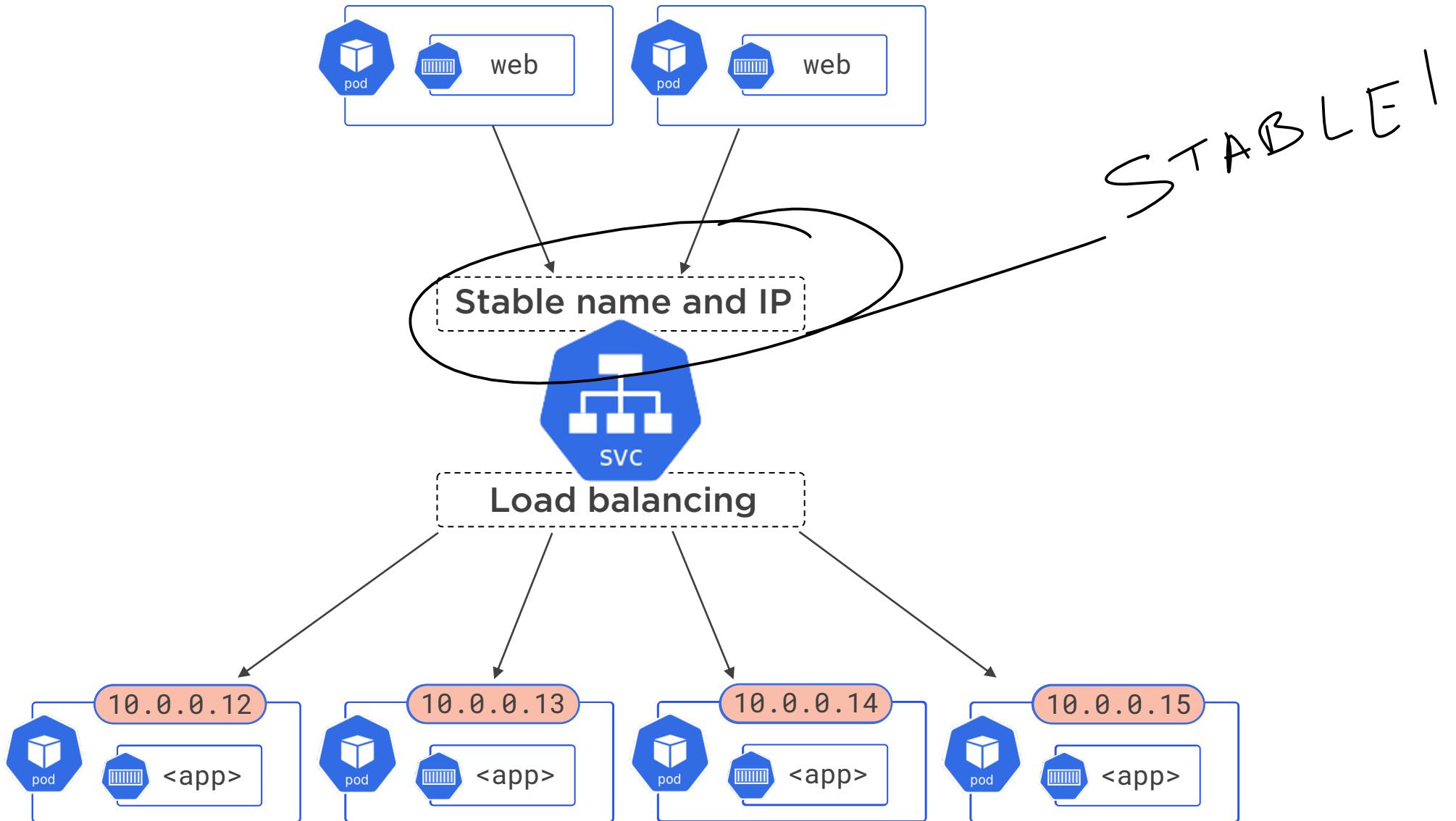








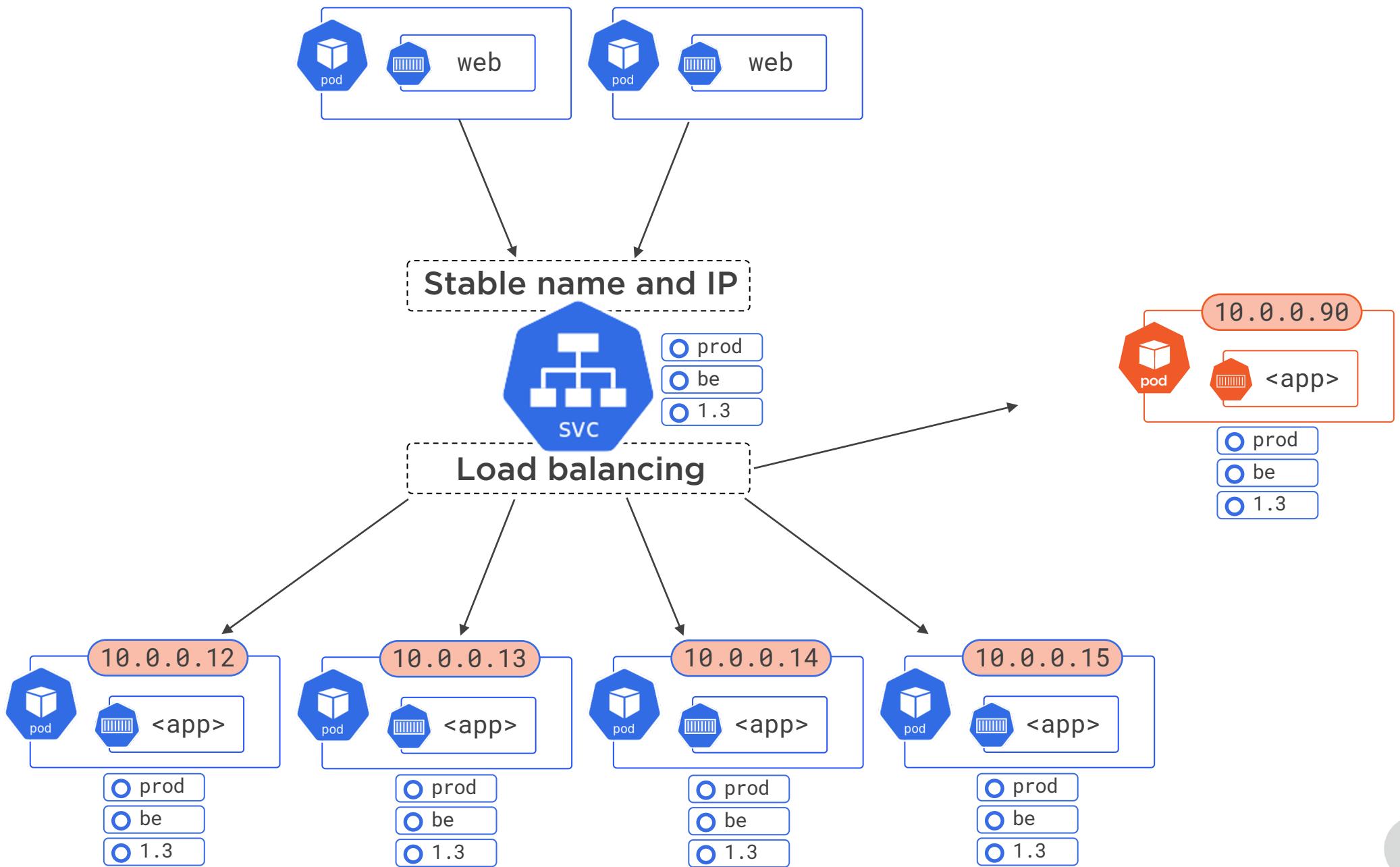


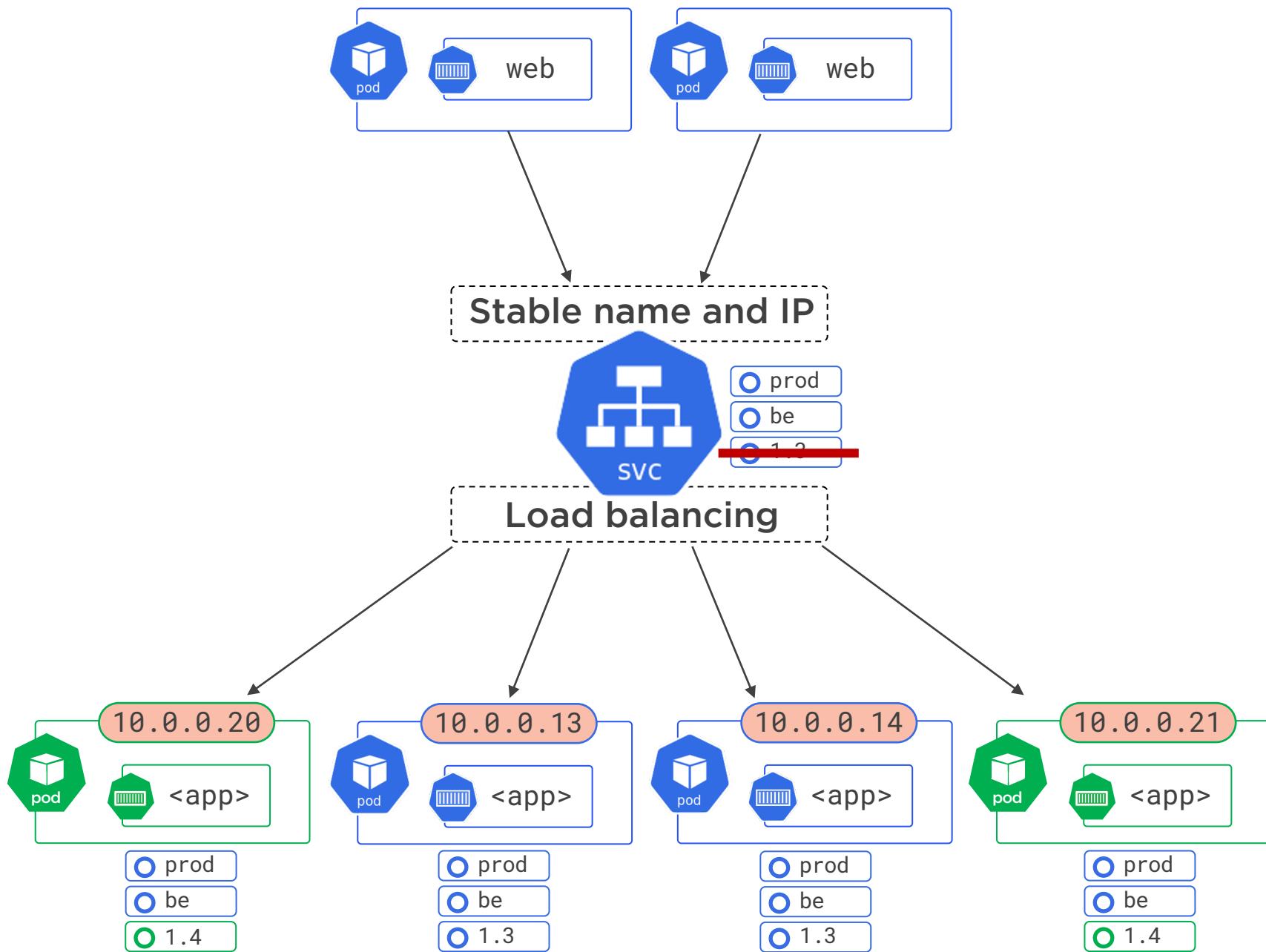


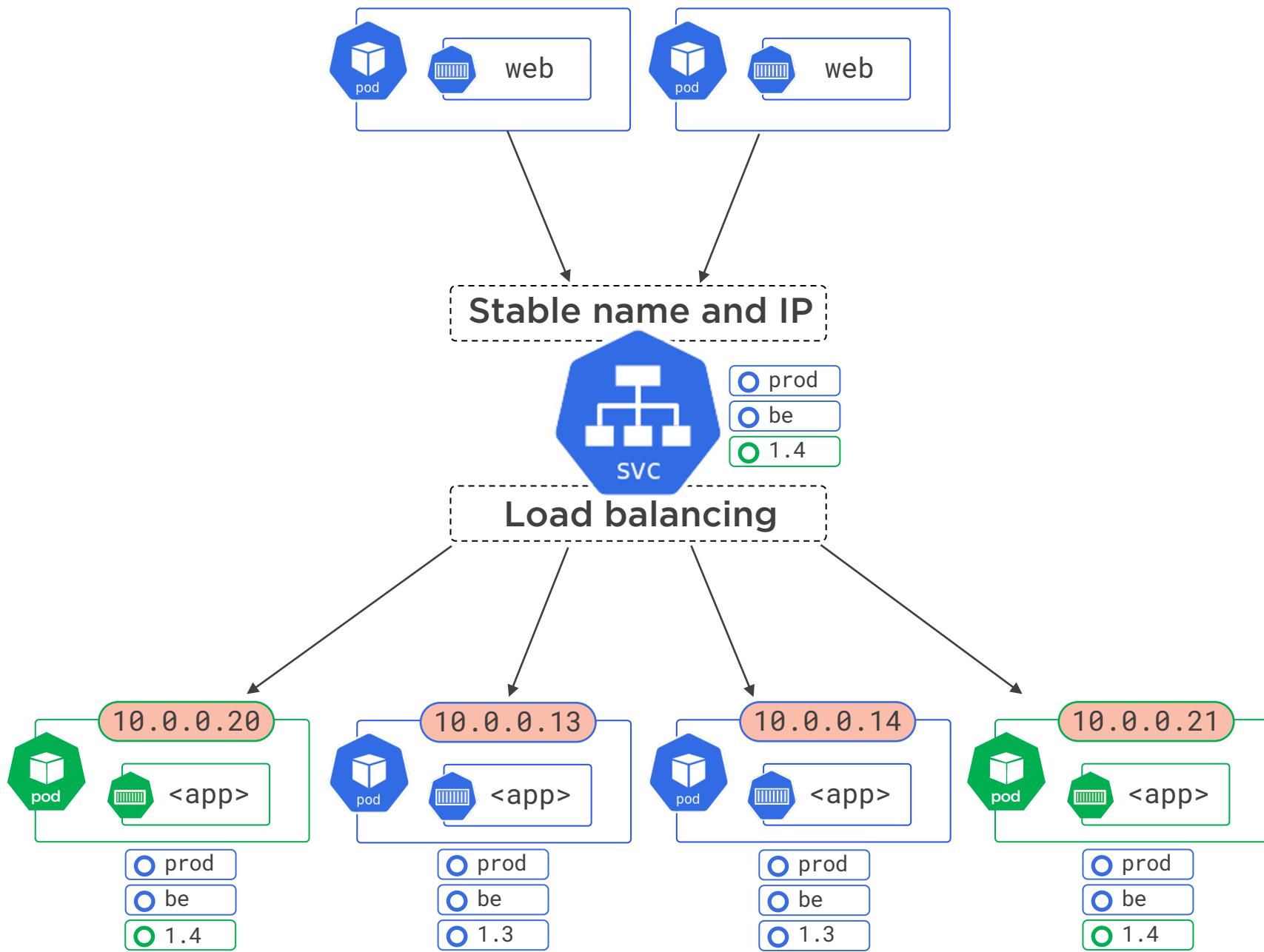
O Labels

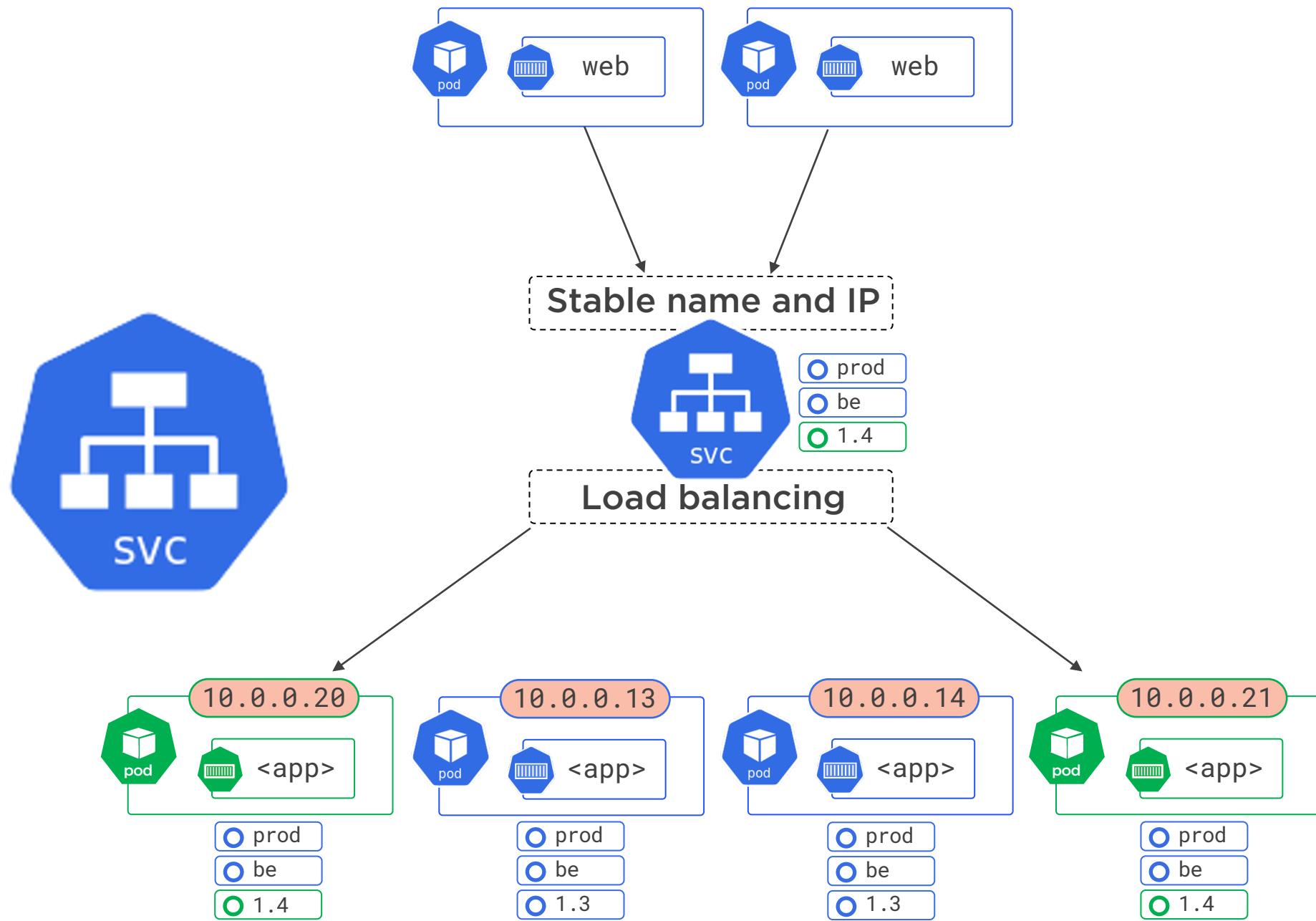
Labels are very simple and very powerful!













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



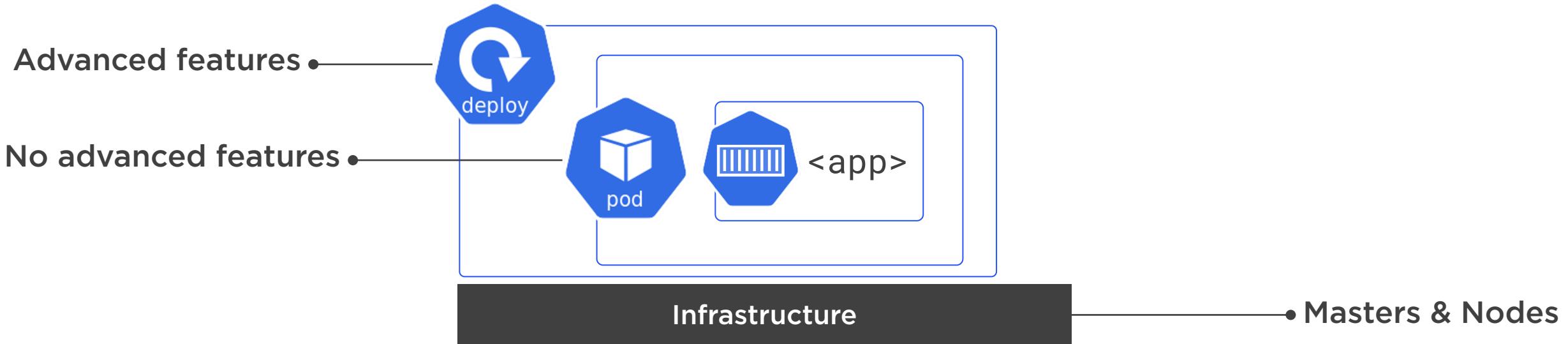
Up Next:

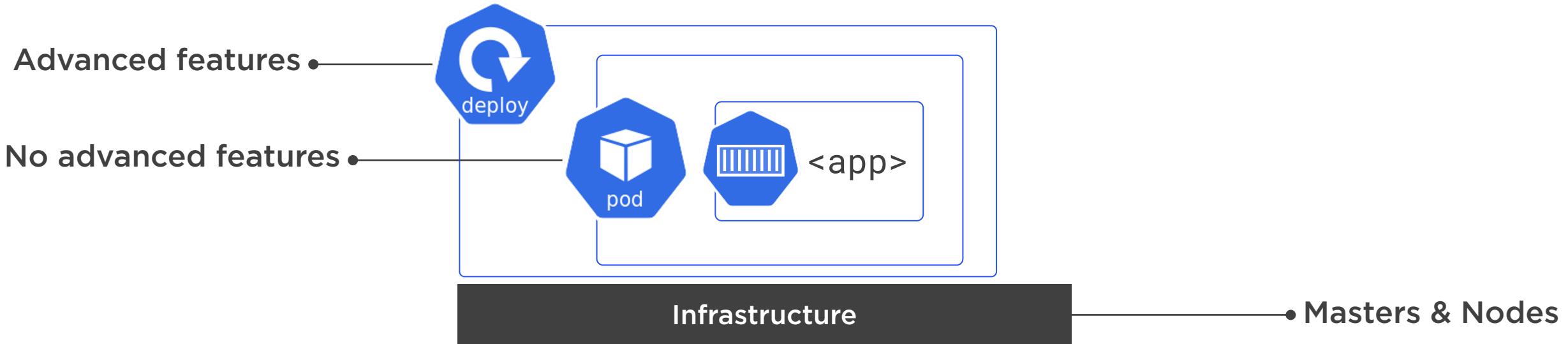
Game-changing Deployments



Game-changing Deployments









Stateless apps



One instance on every node



Stateful apps...



Time-based short-lived jobs





Stateless apps



One instance on every node

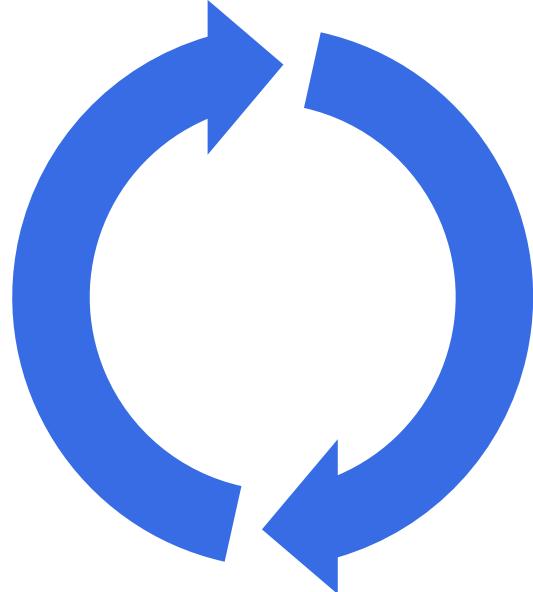


Stateful apps...



Time-based short-lived jobs





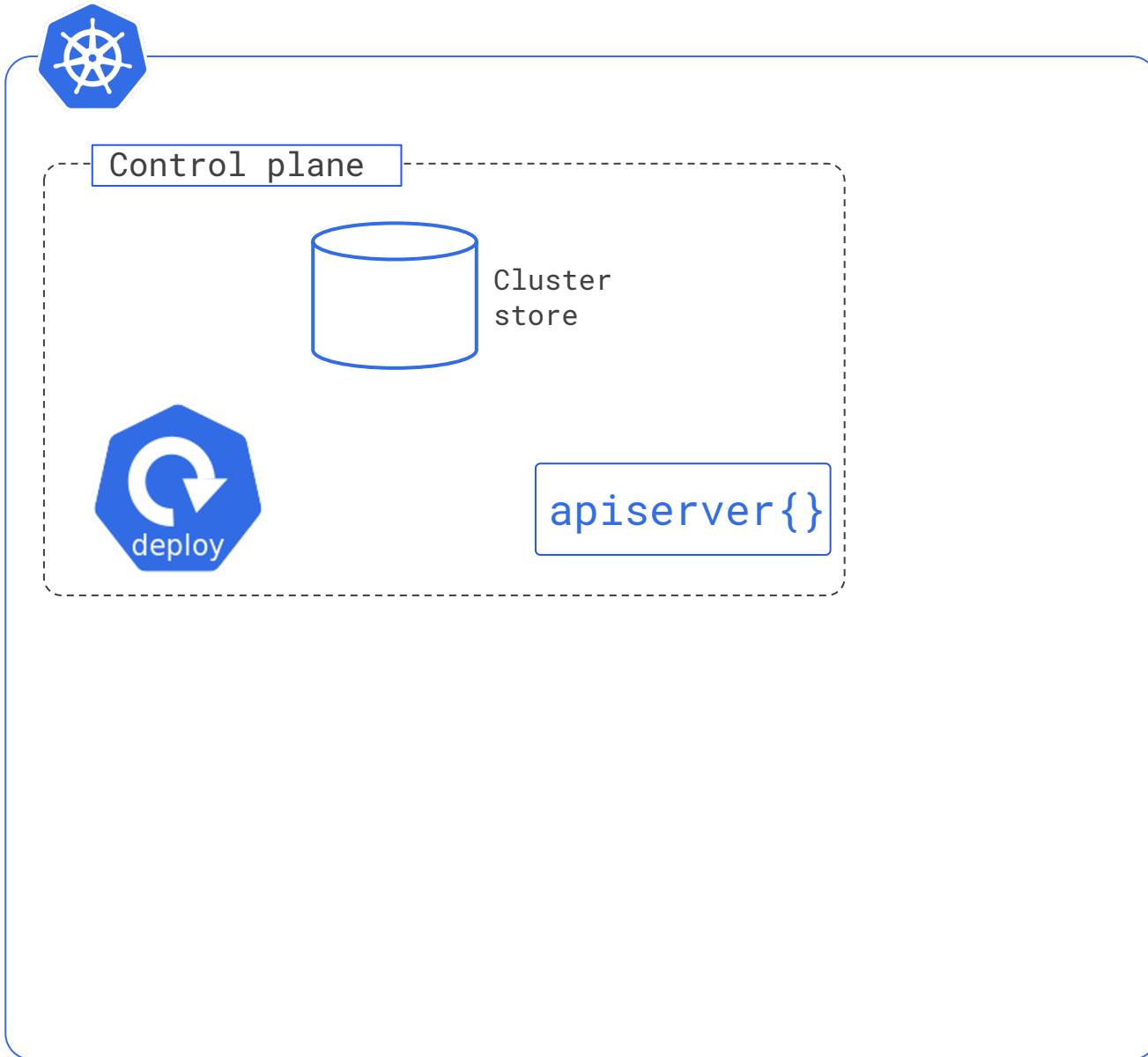
Deployment Controller/Reconciliation loop

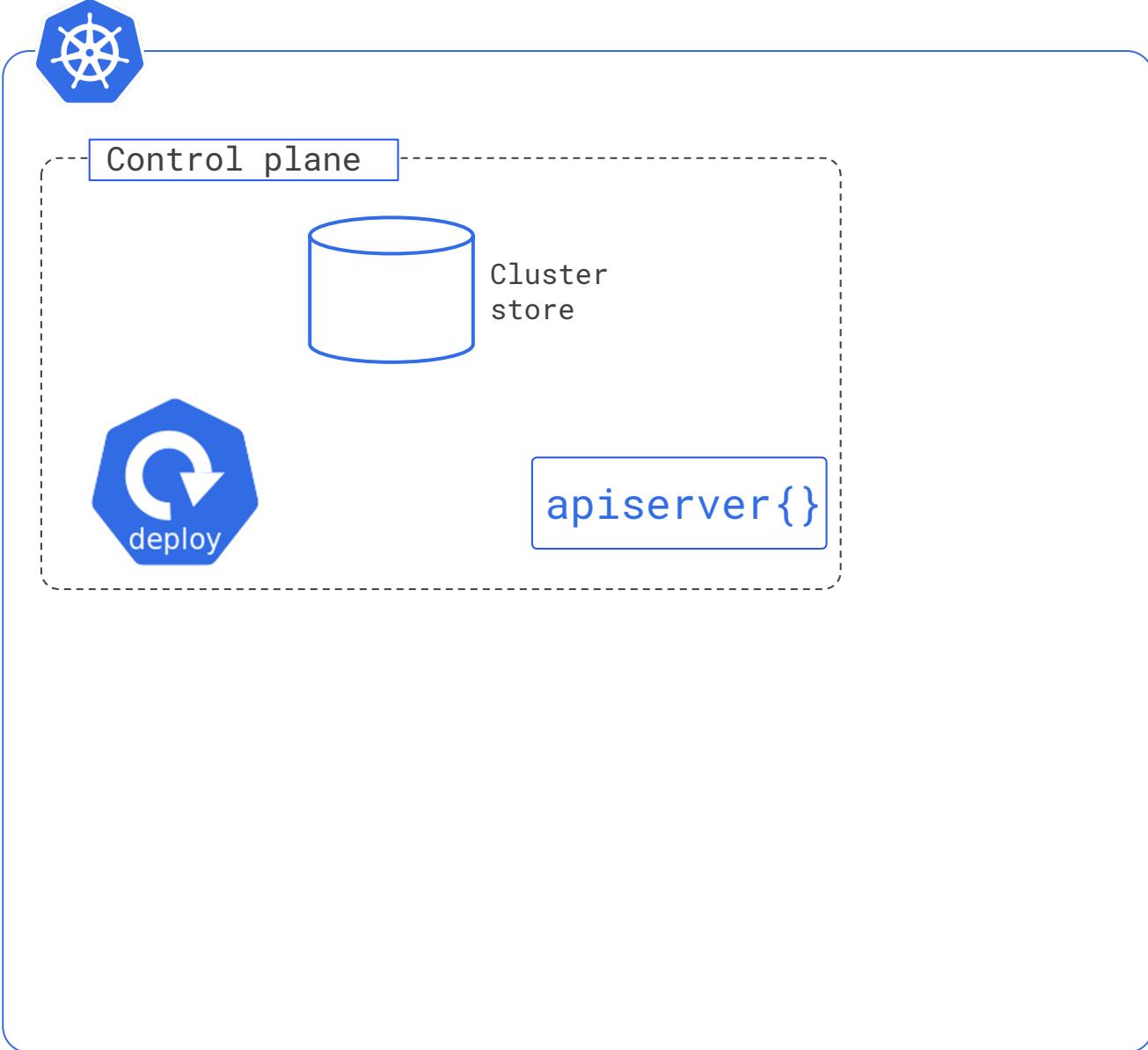
Watches API Server for new Deployments

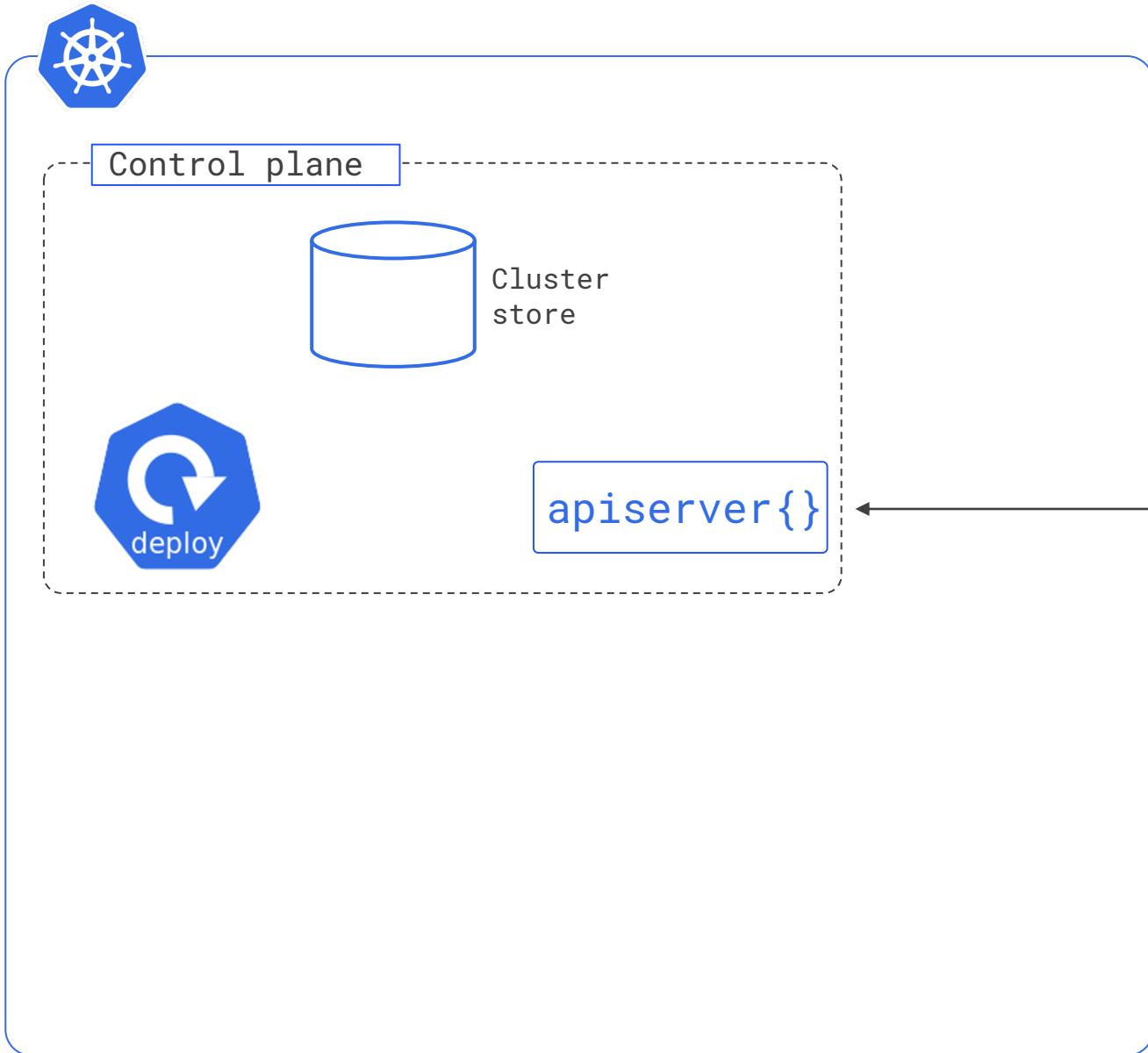
Implements them

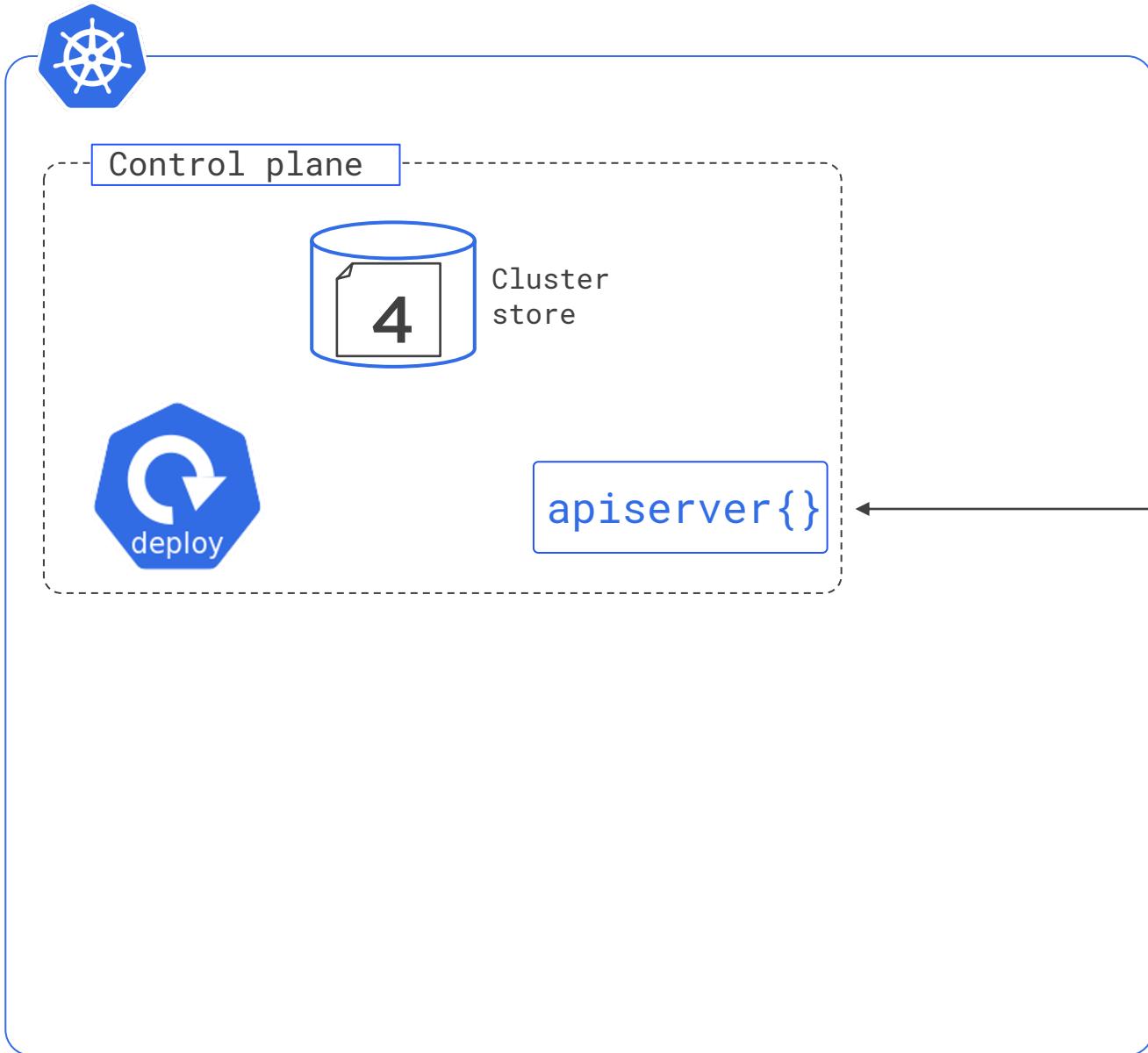
Constantly compares *observed state* with
desired state

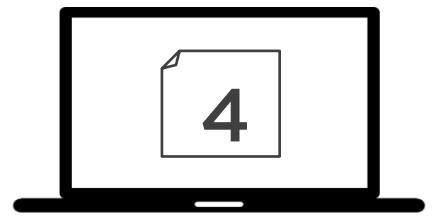
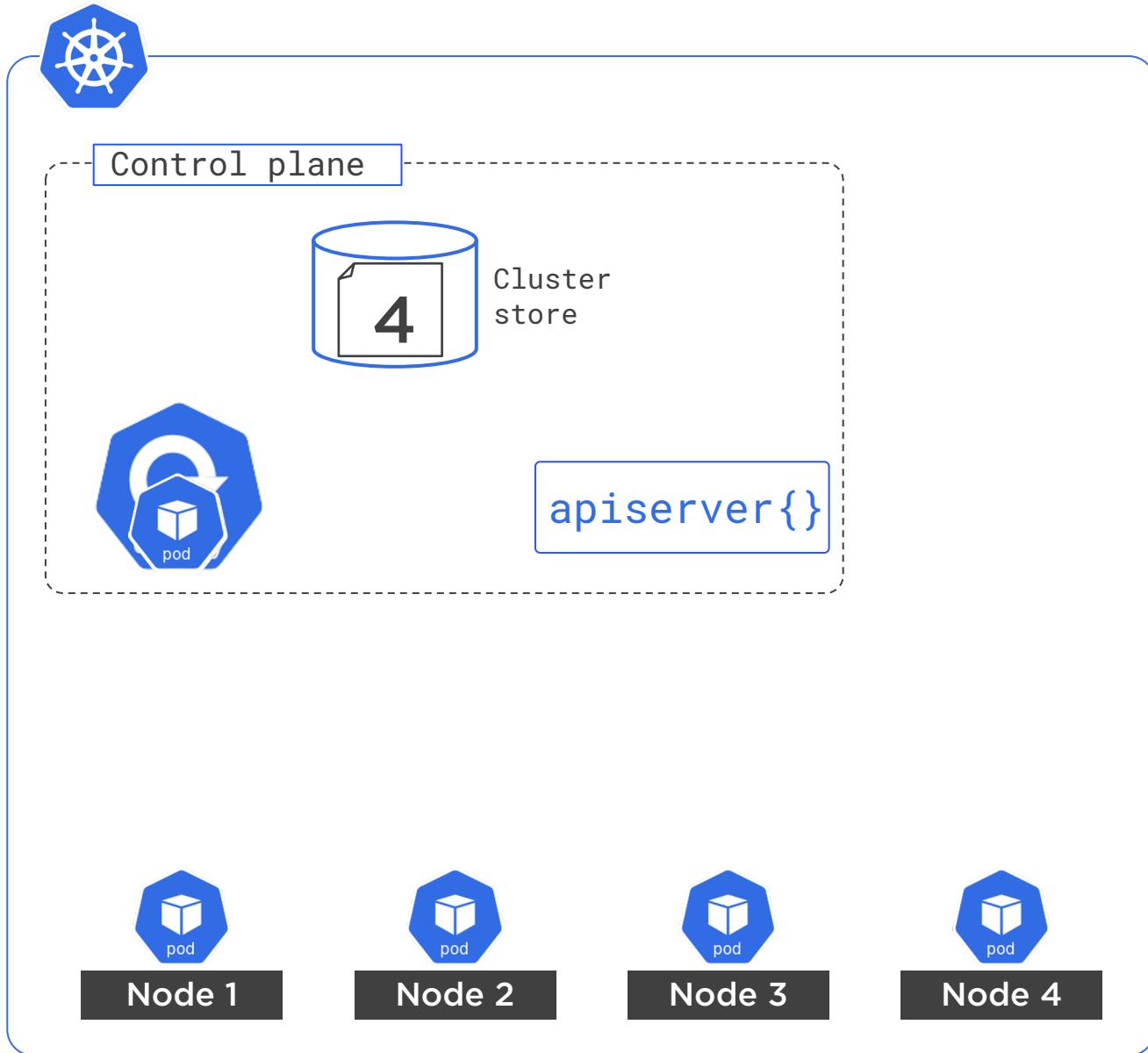


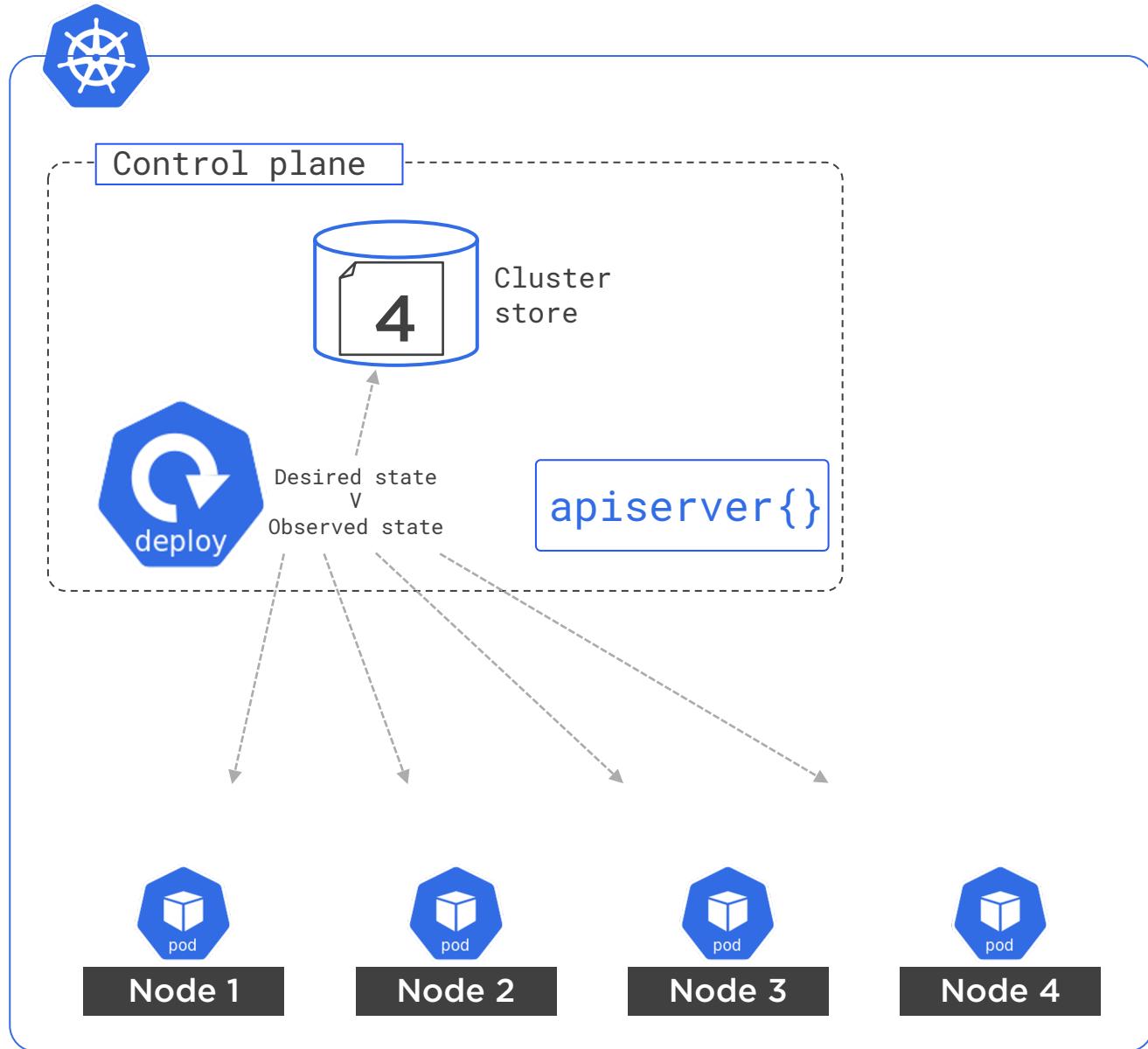


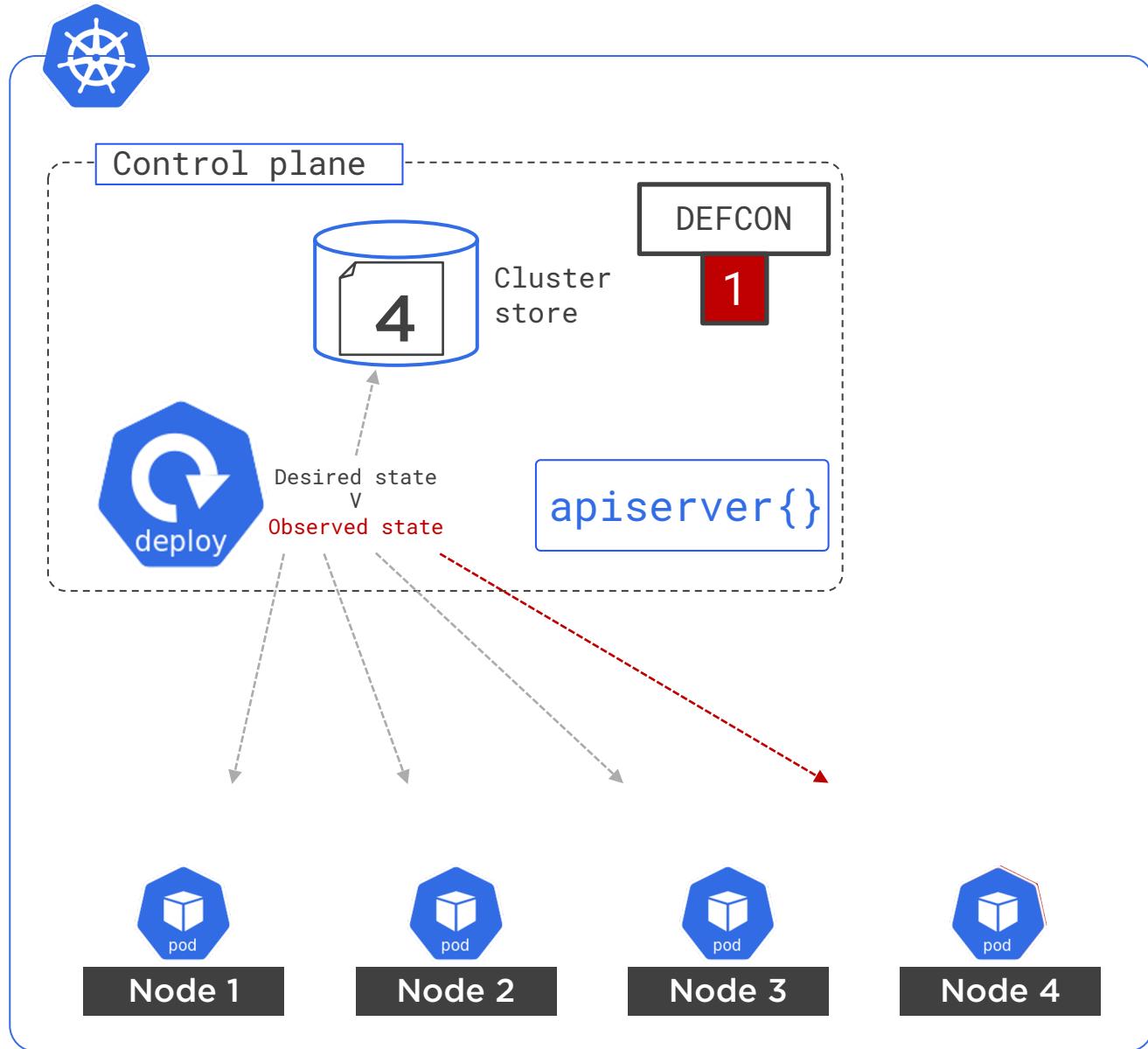


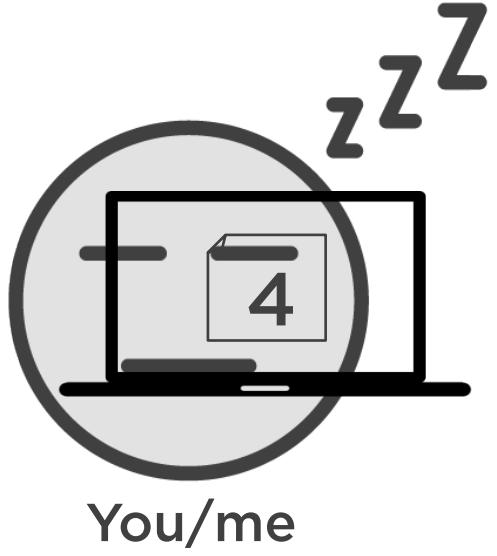
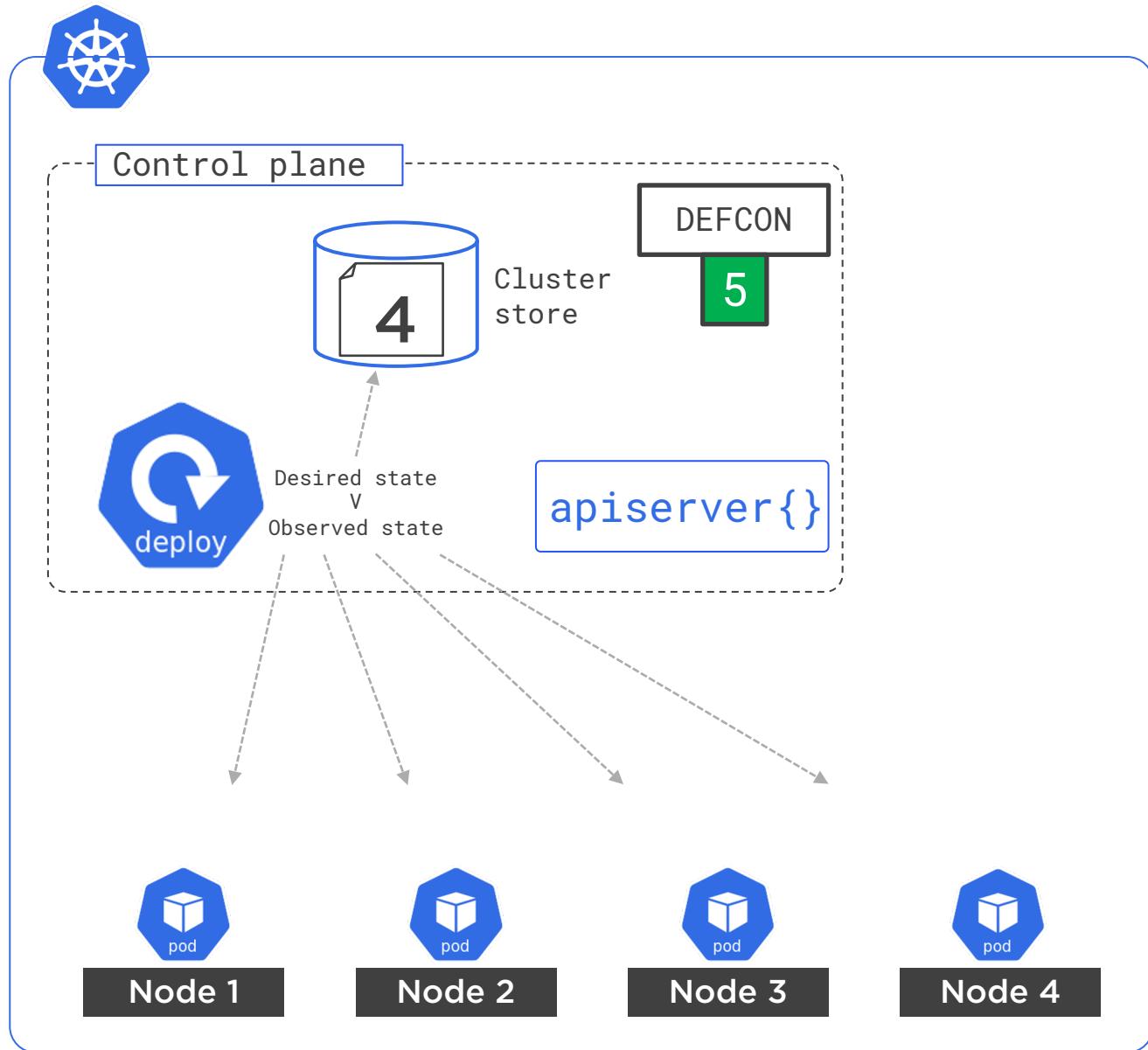


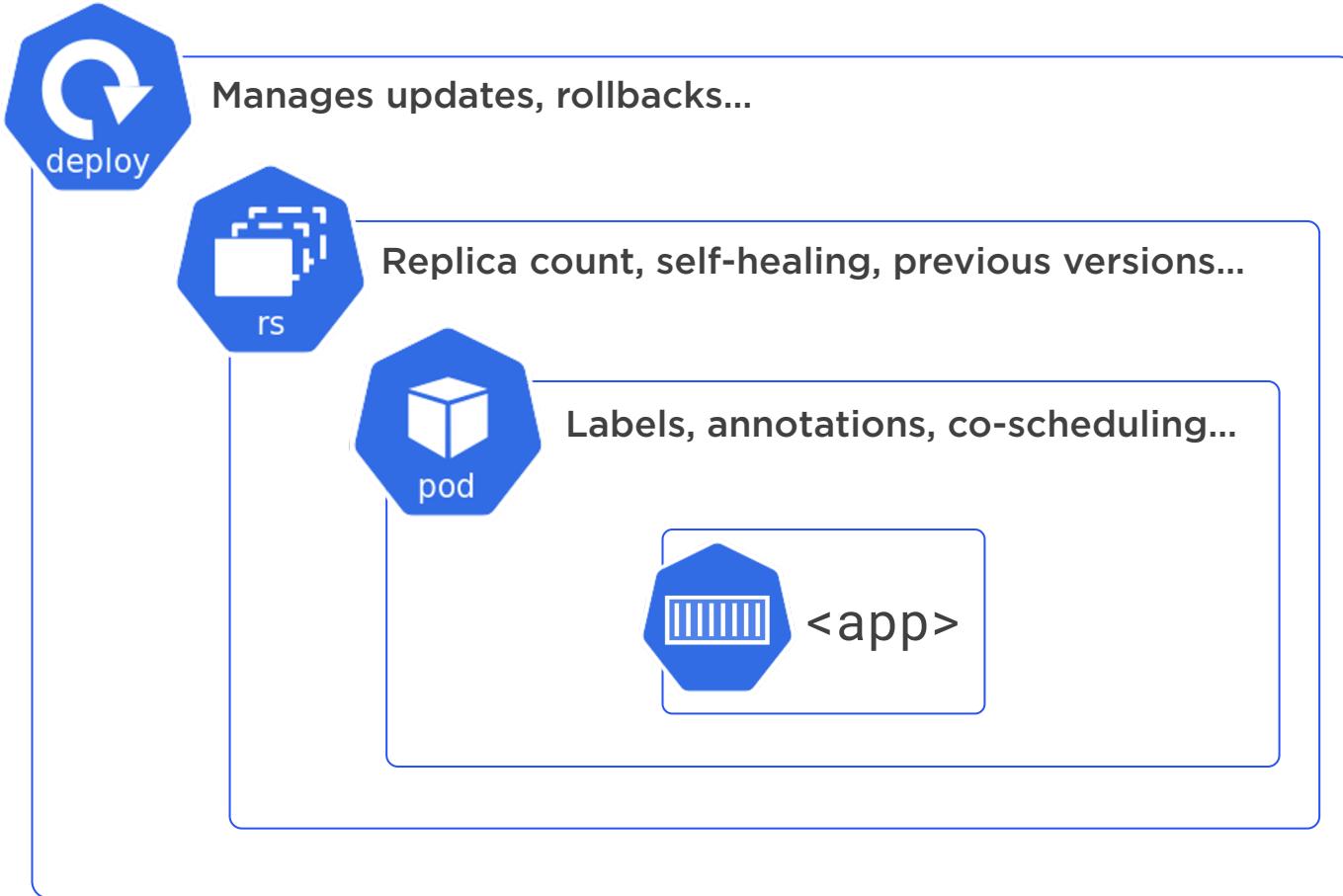


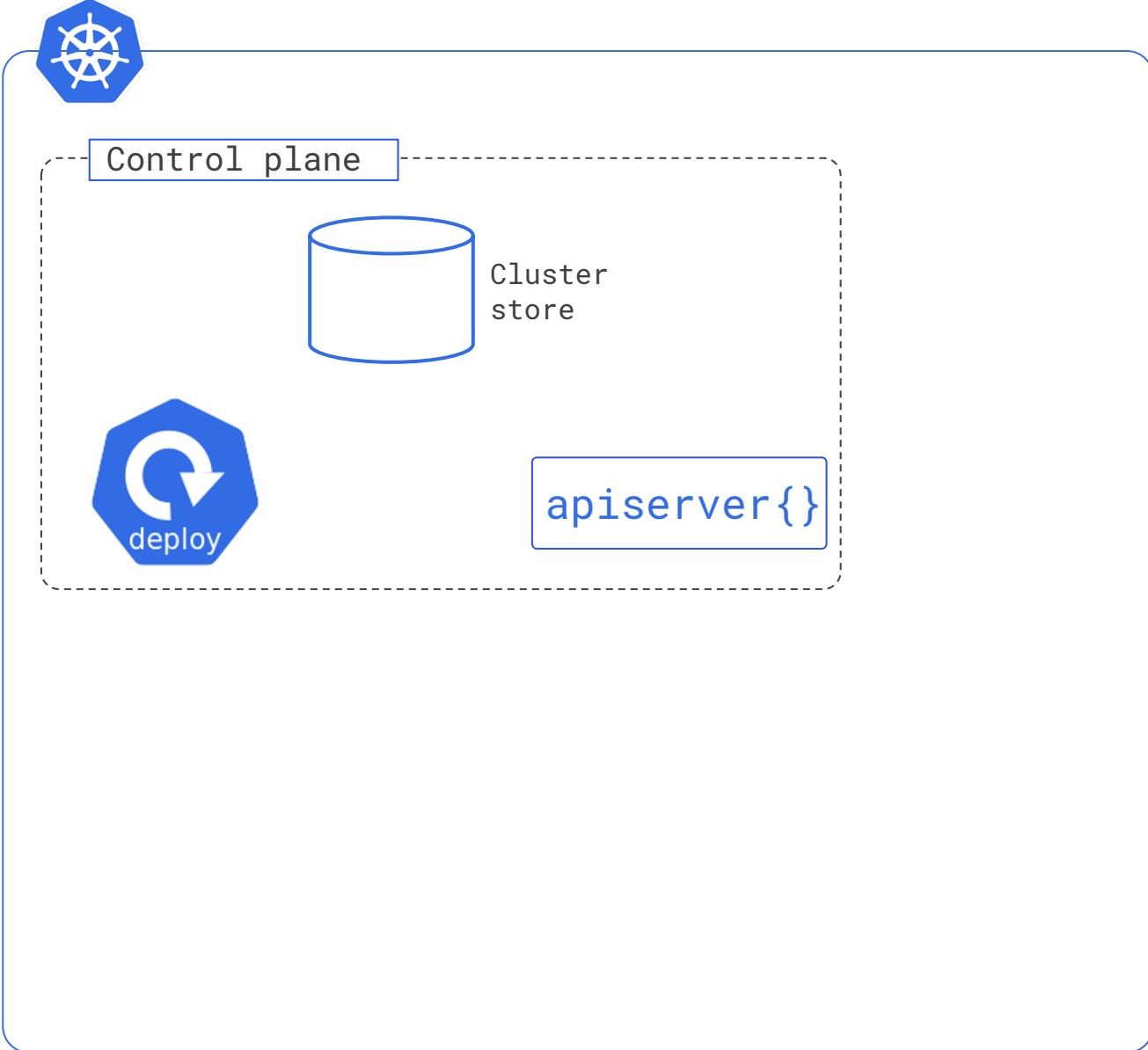


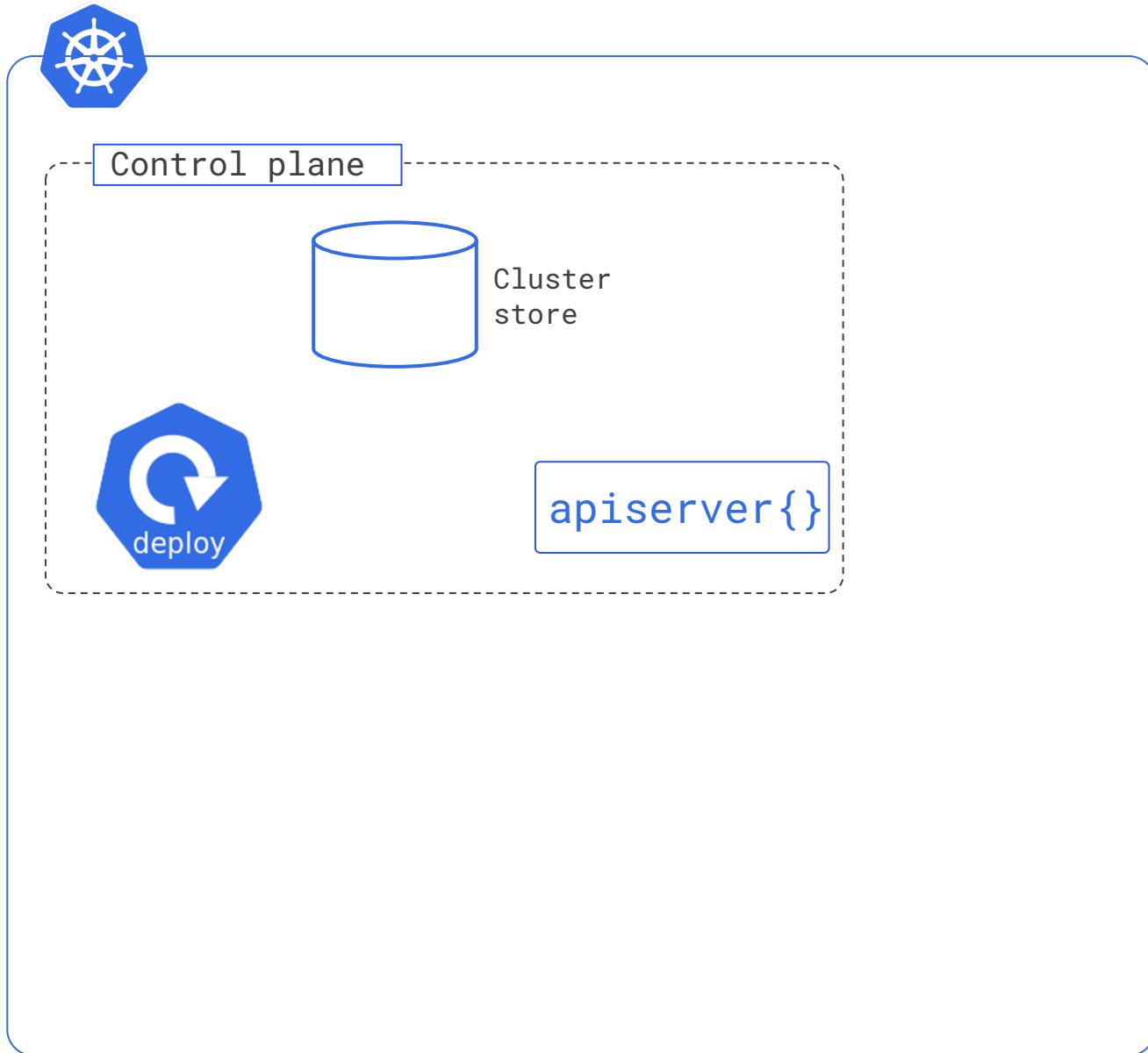


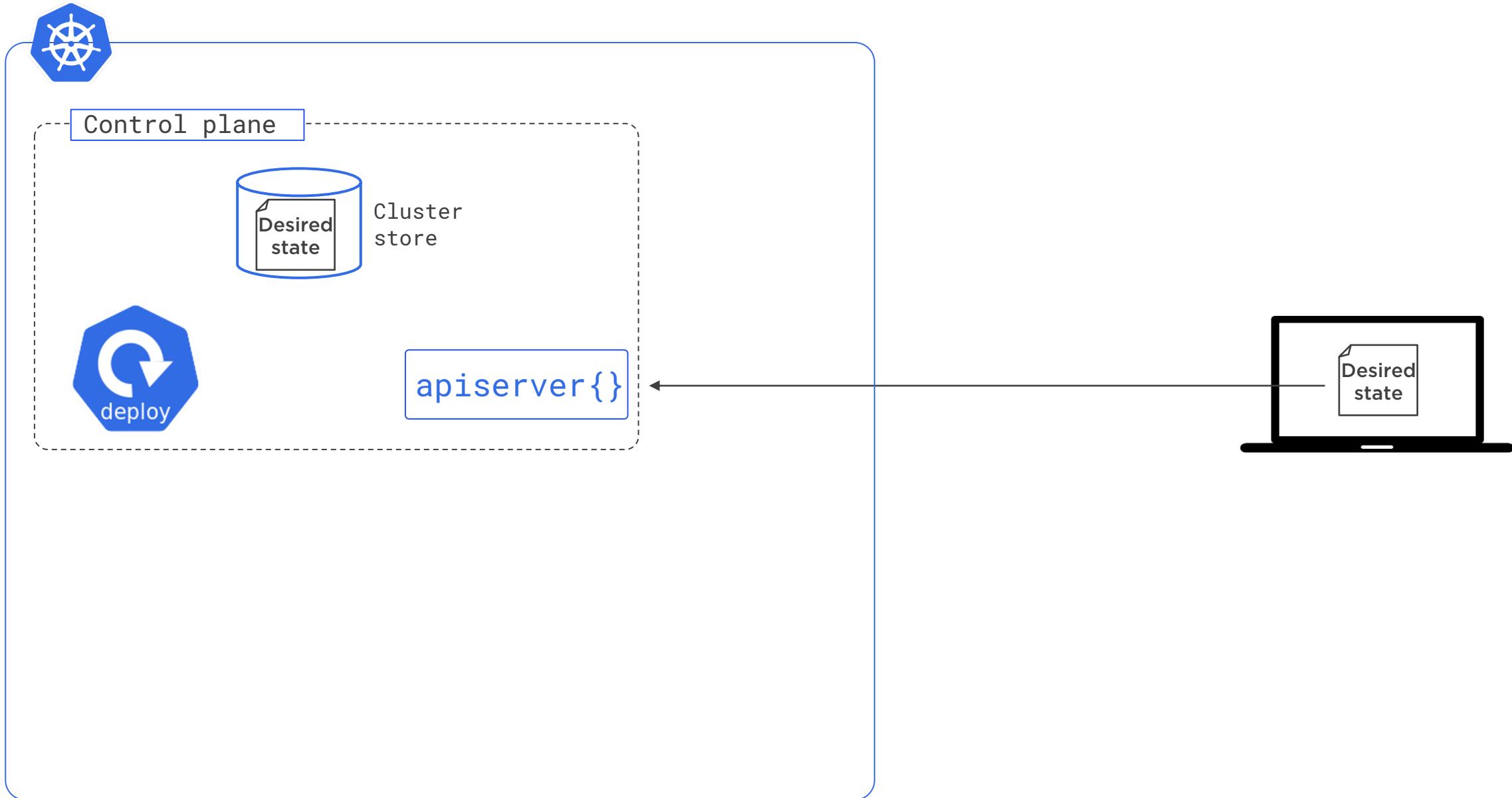


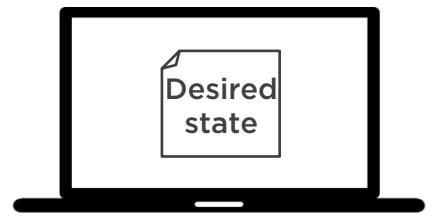
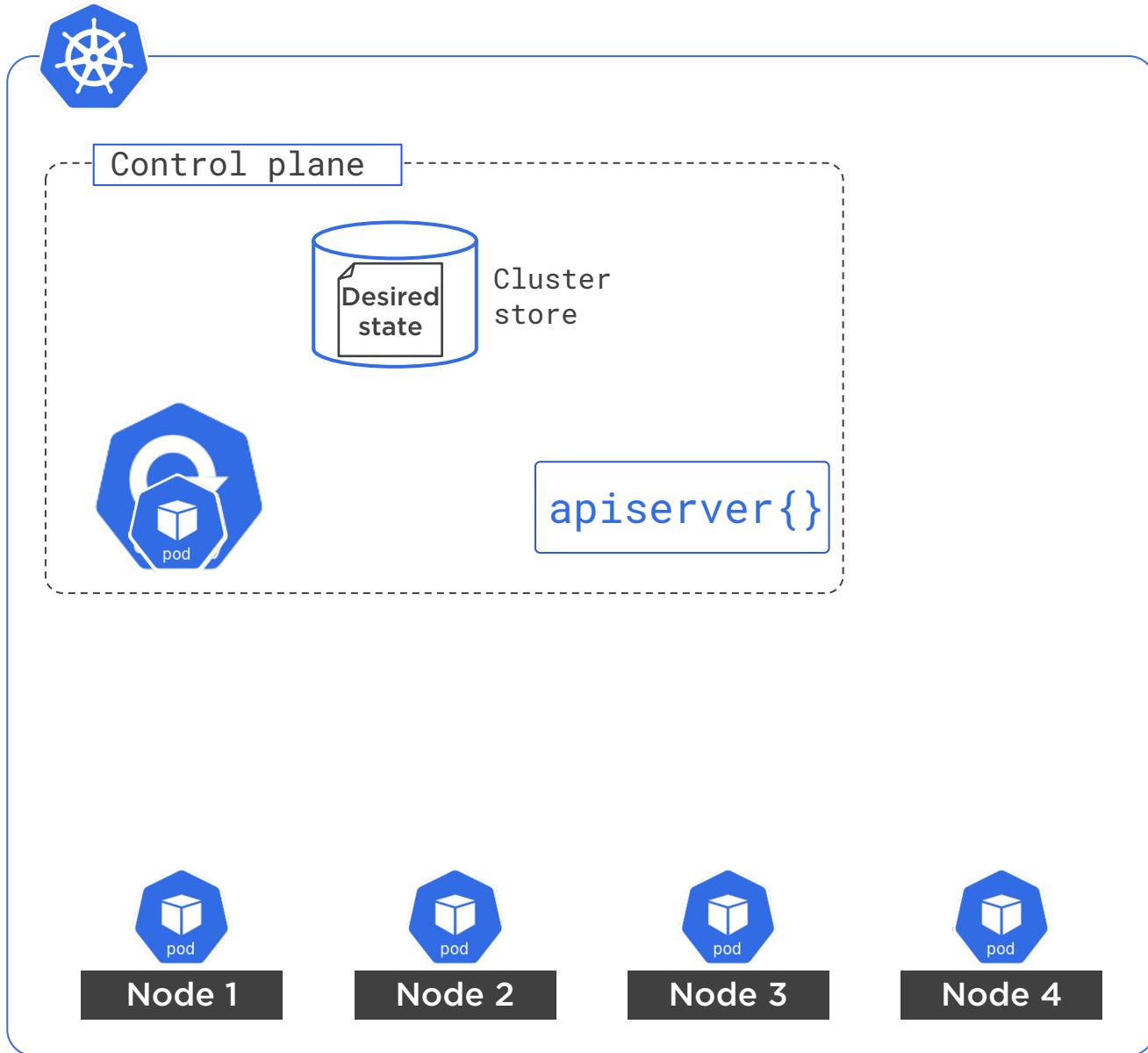


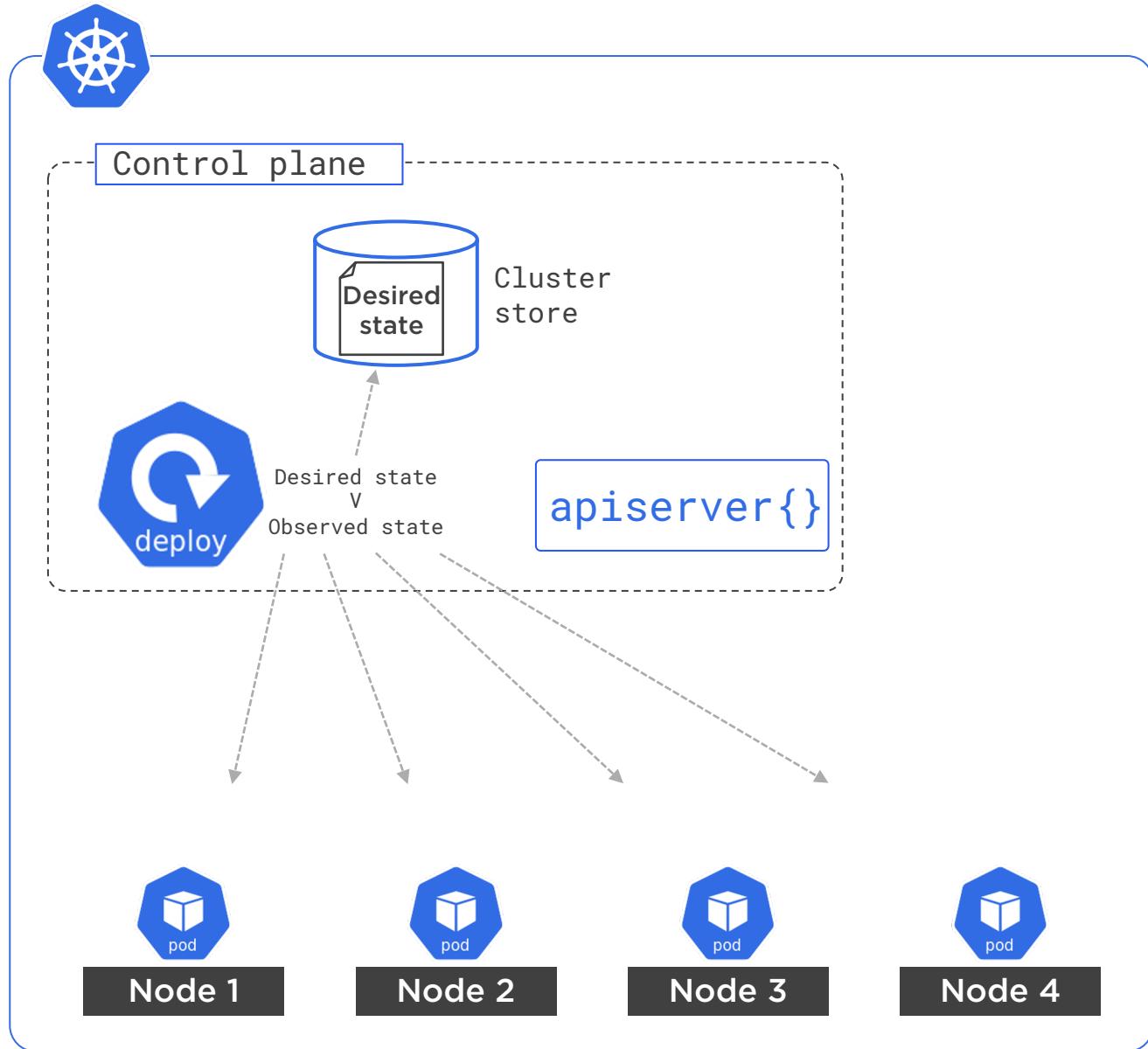












Up Next:
The K8s API and API Server



The K8s API and API Server





Atomic unit of scheduling



Replica count



Updates and rollbacks



Stable network abstraction





Atomic unit of scheduling



Replica count

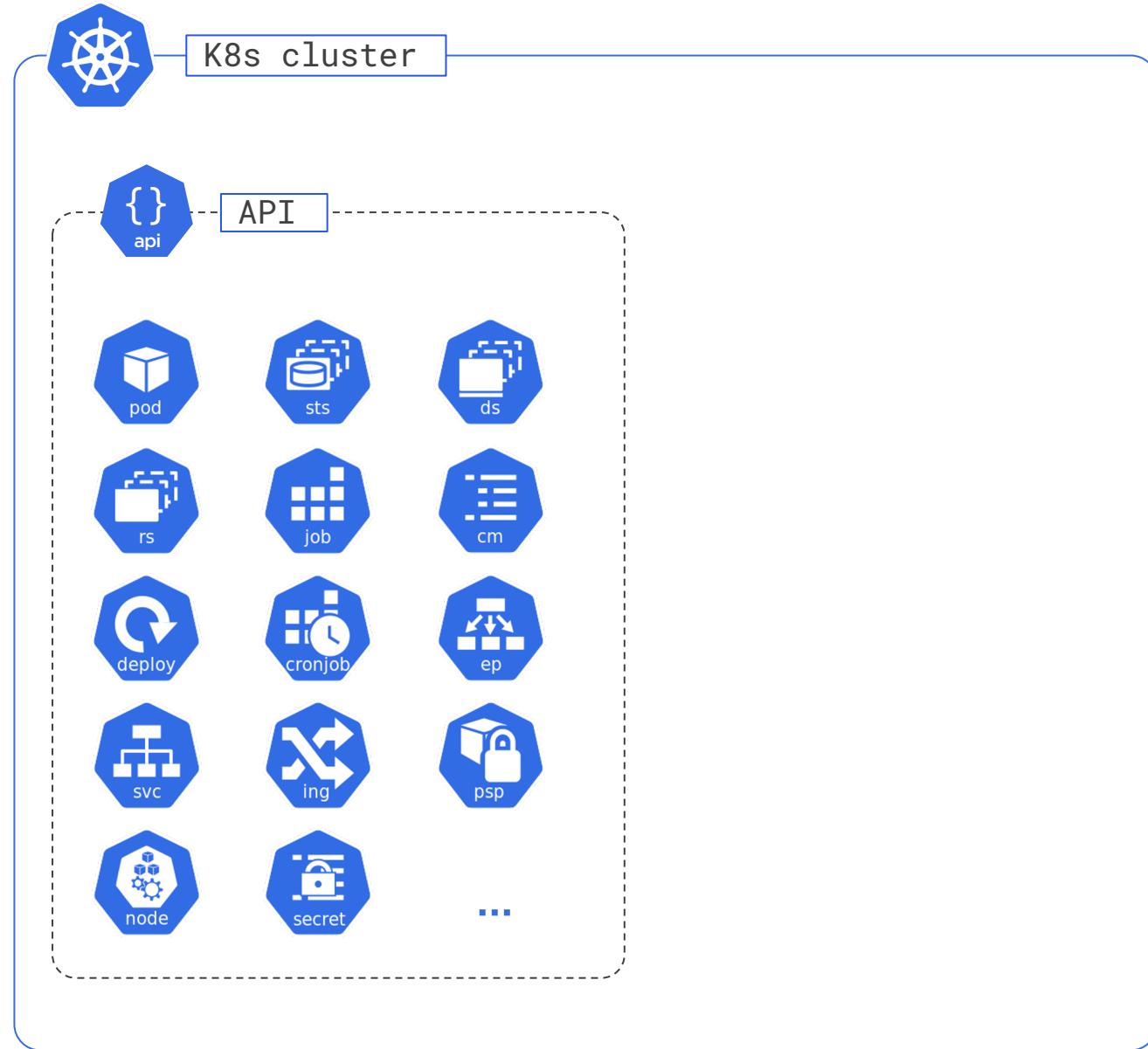


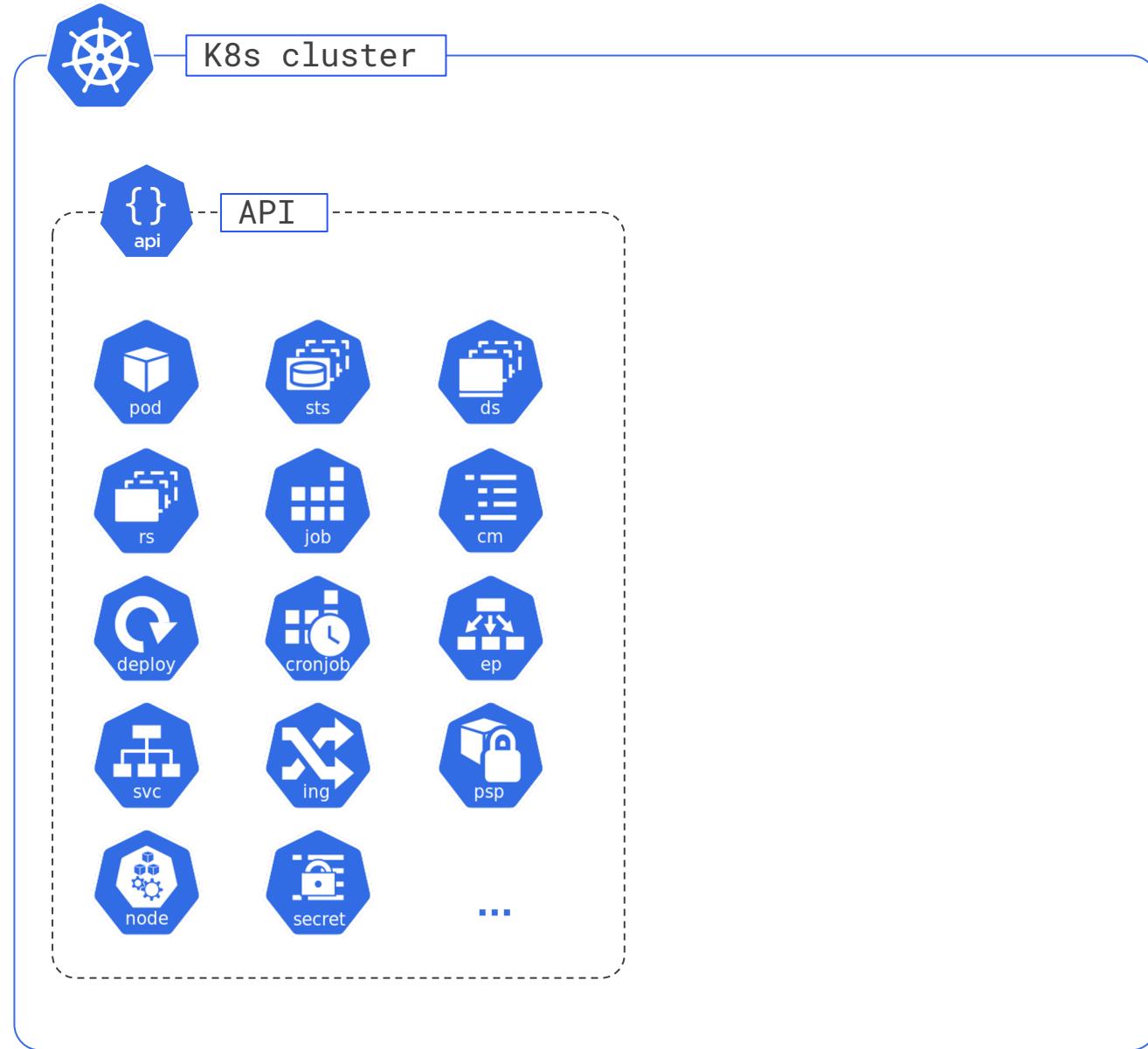
Updates and rollbacks

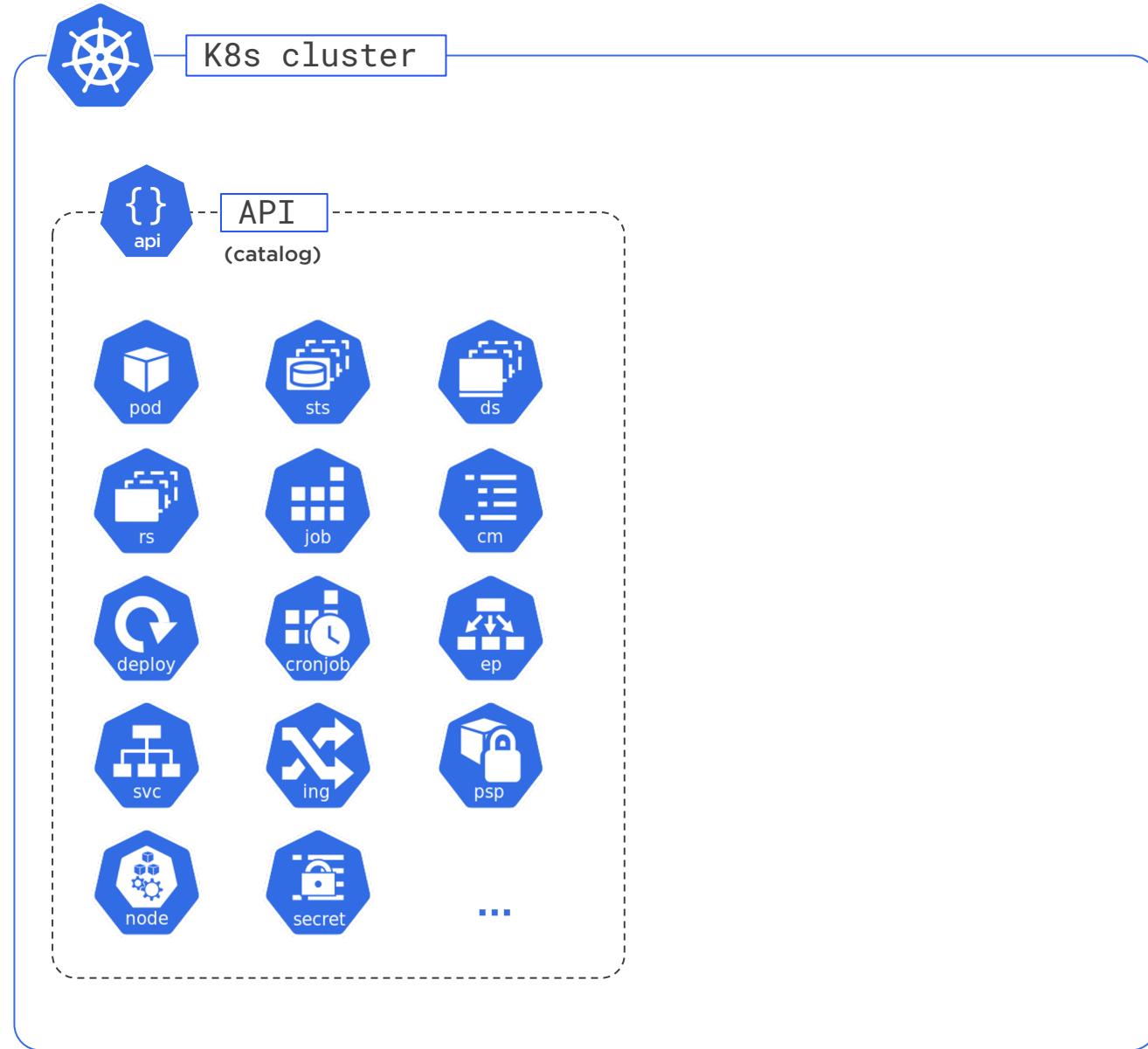


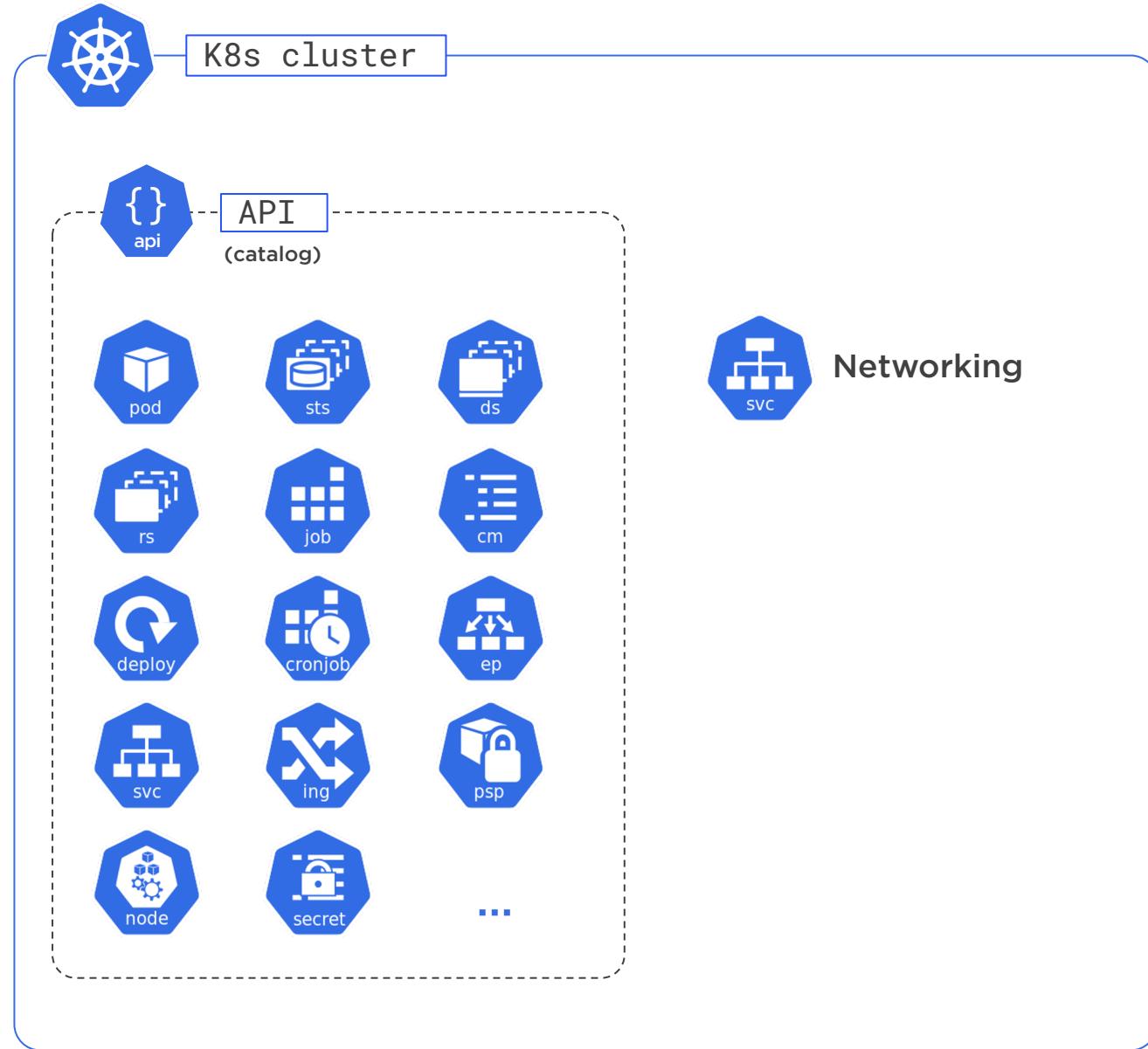
Stable network abstraction

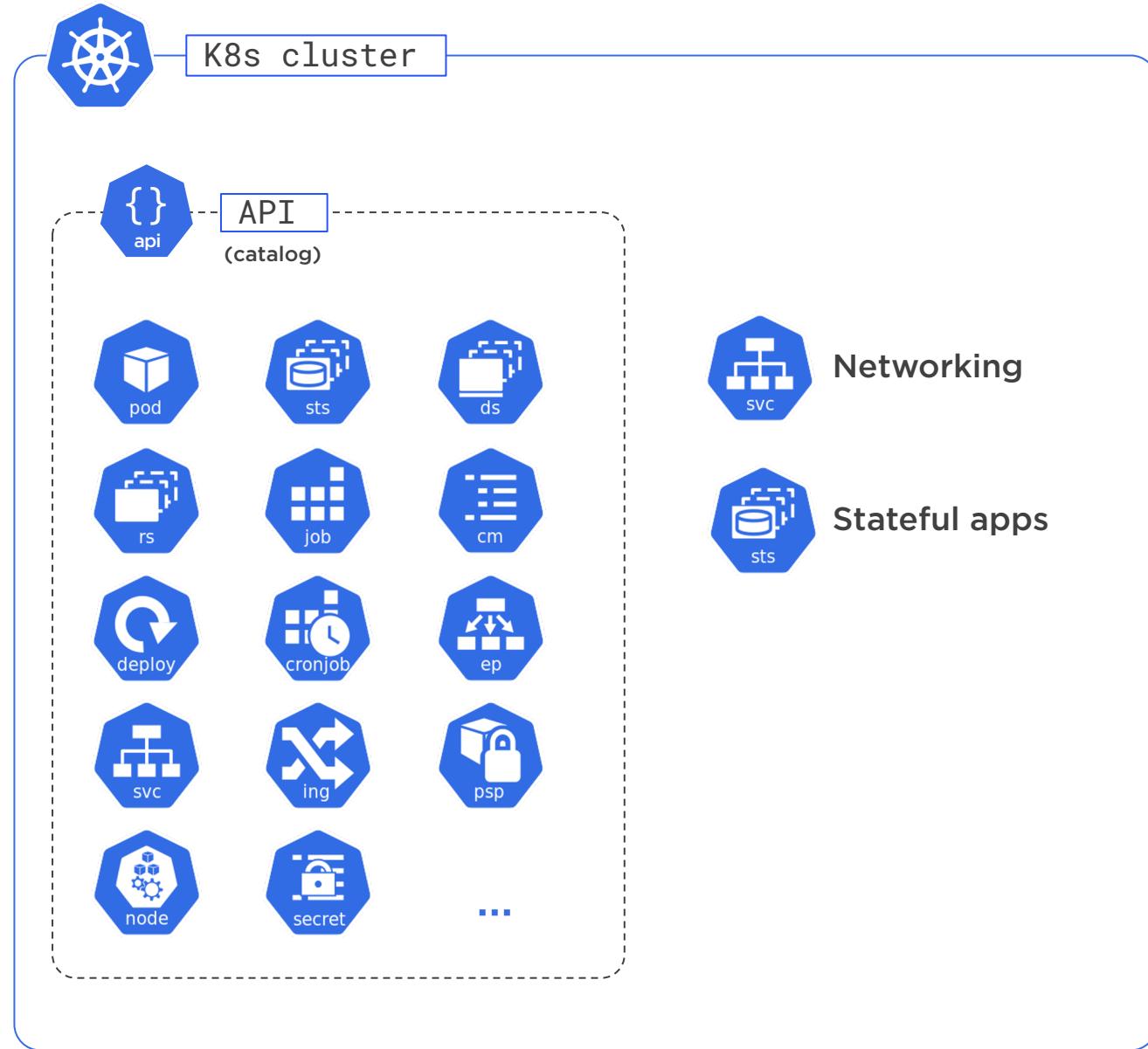


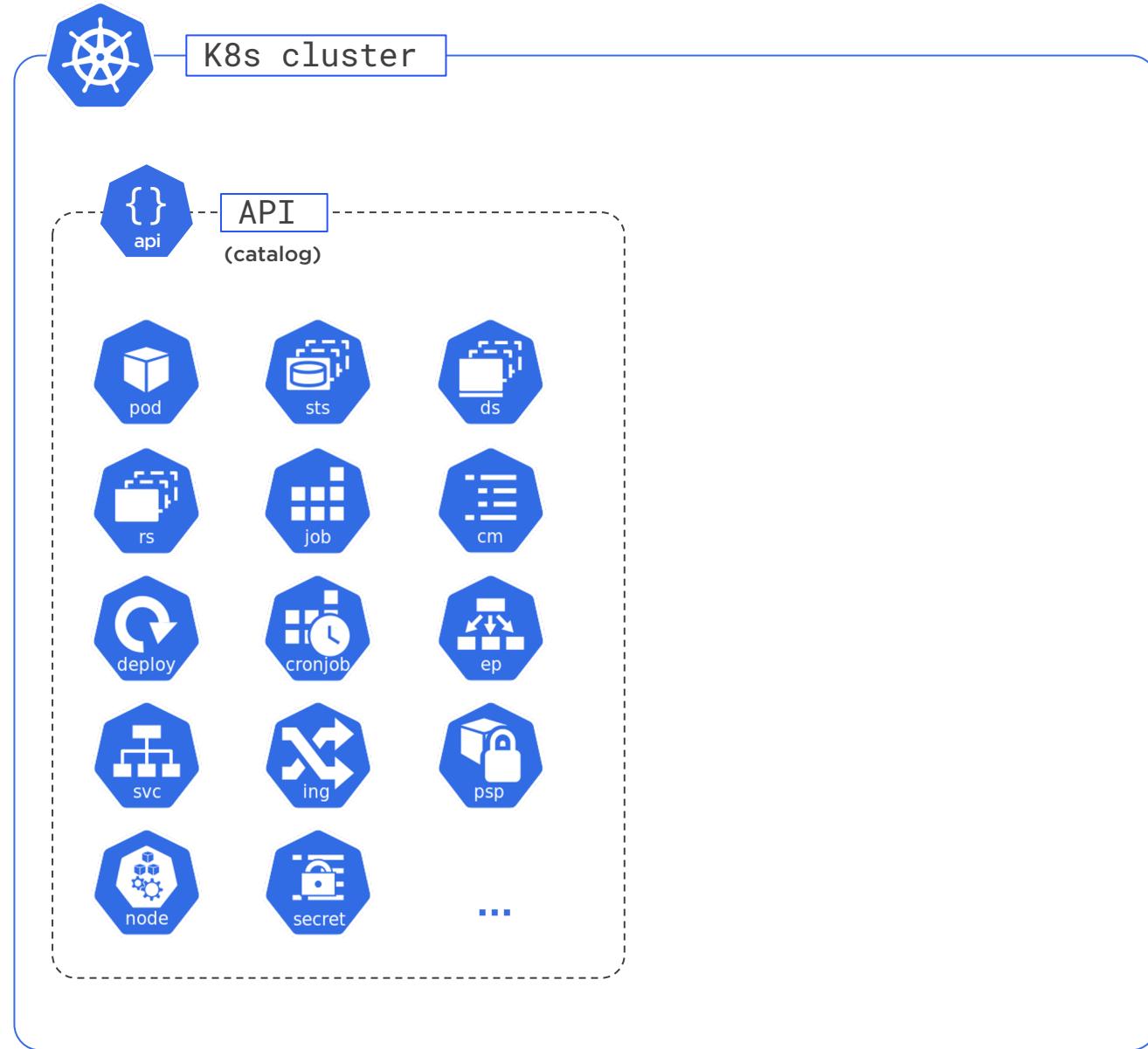


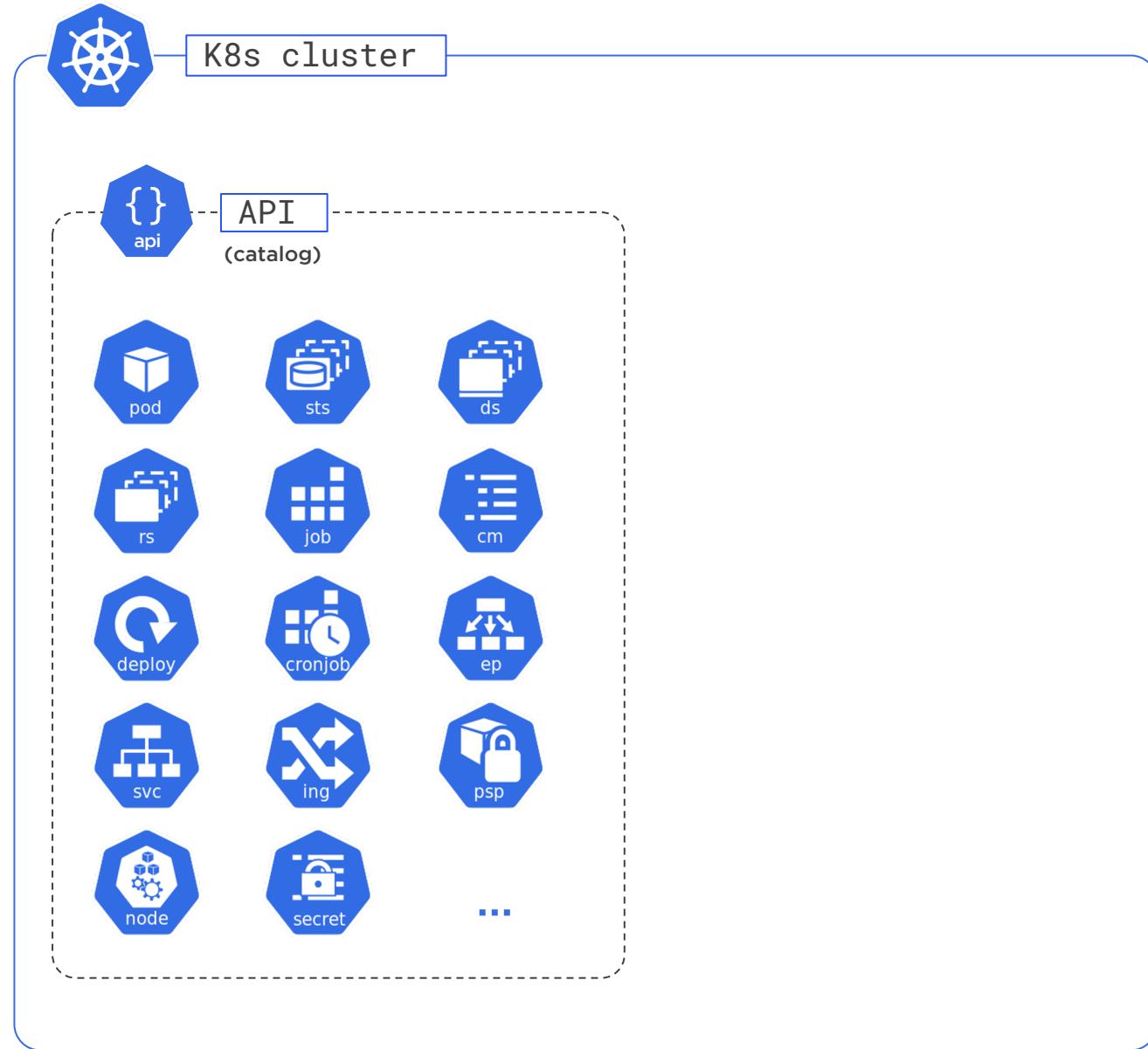


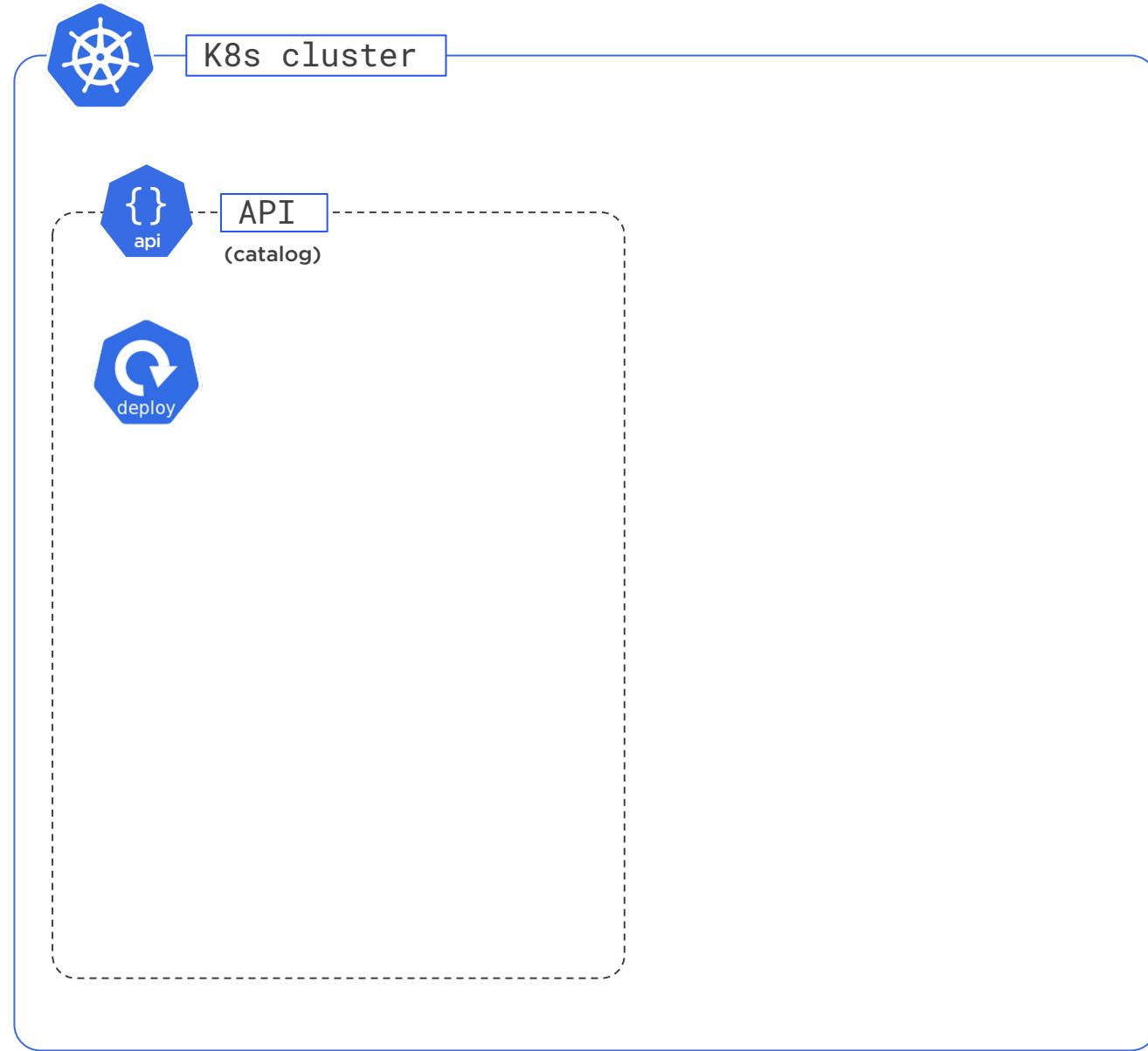


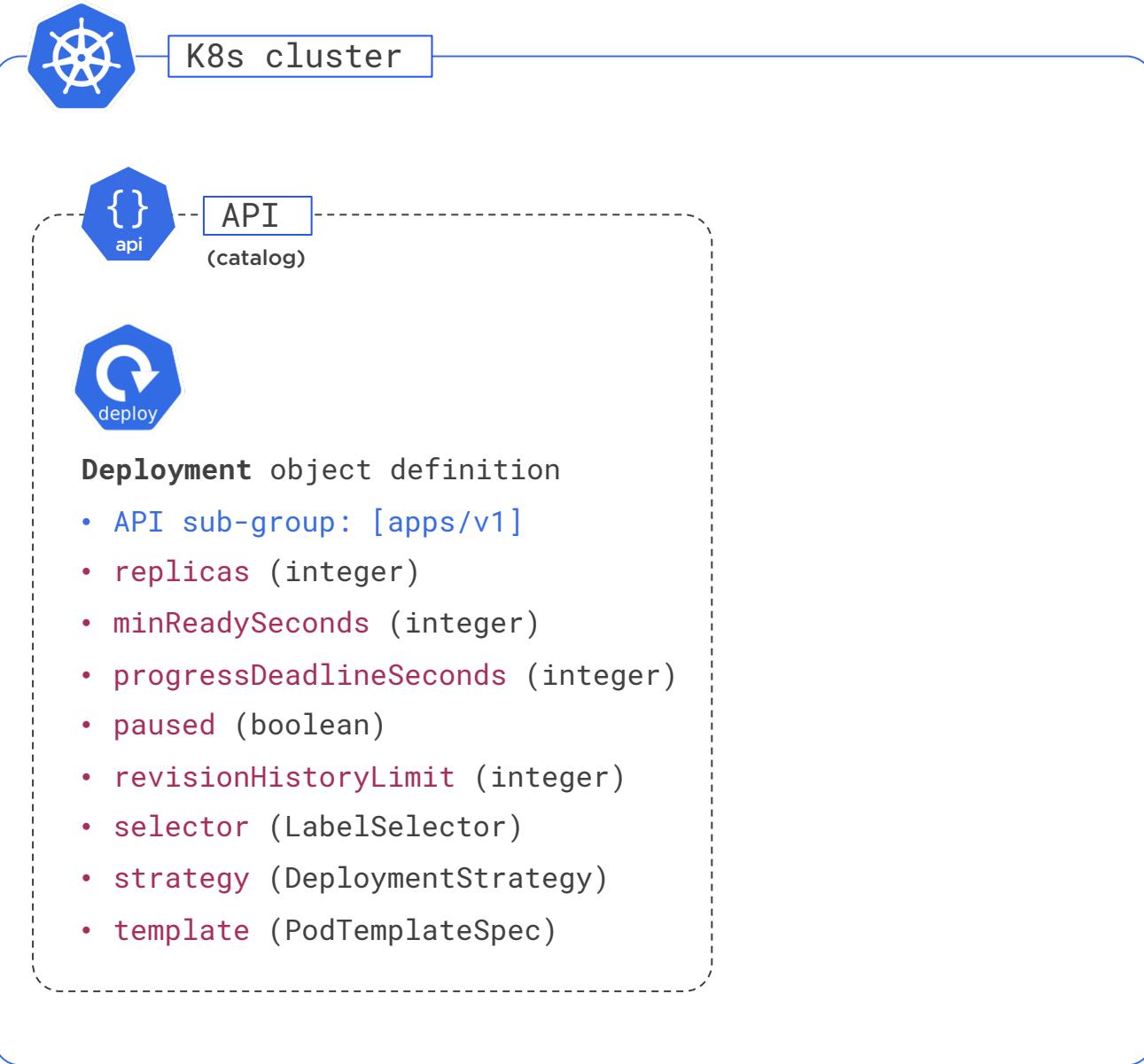


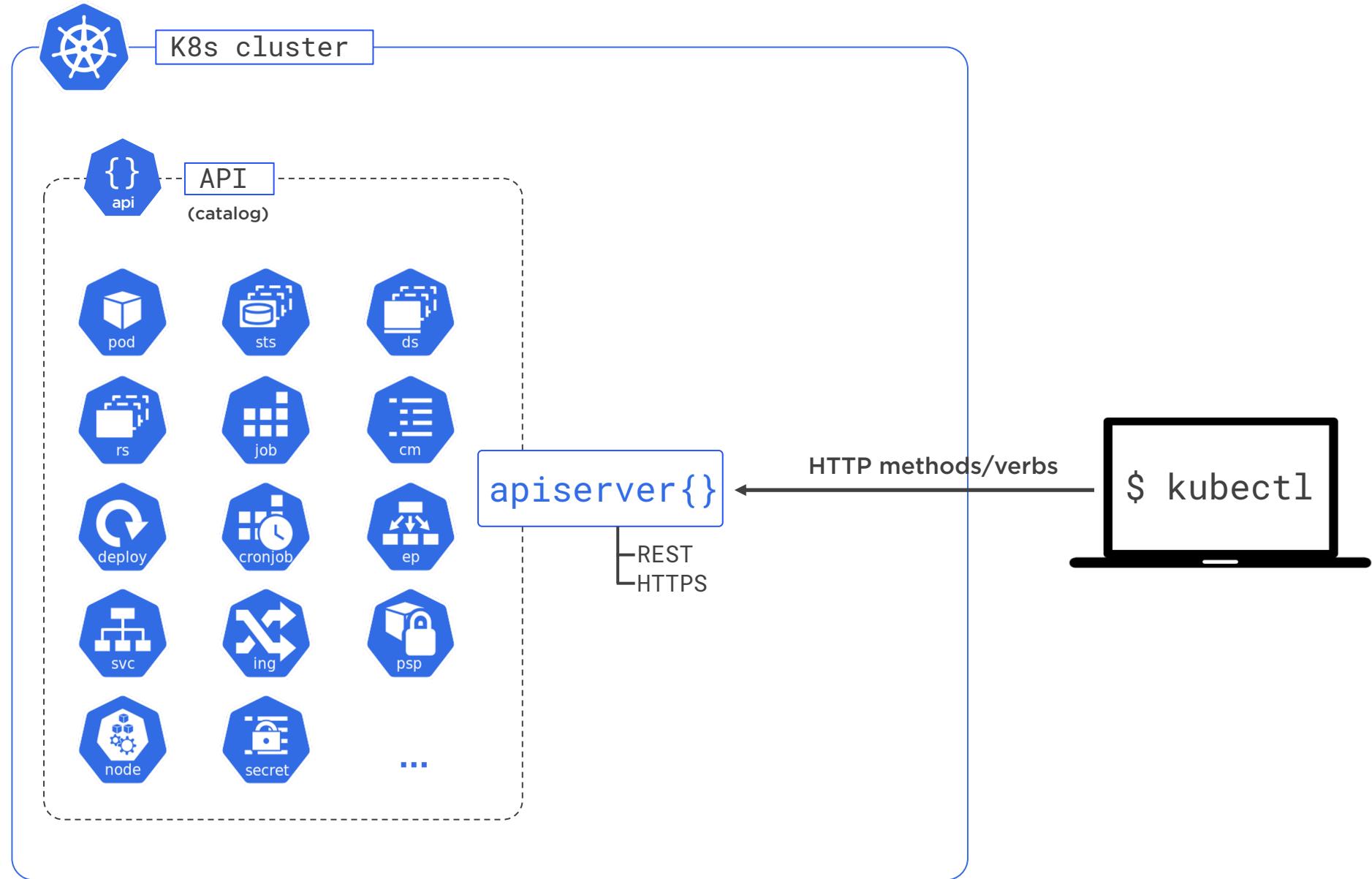


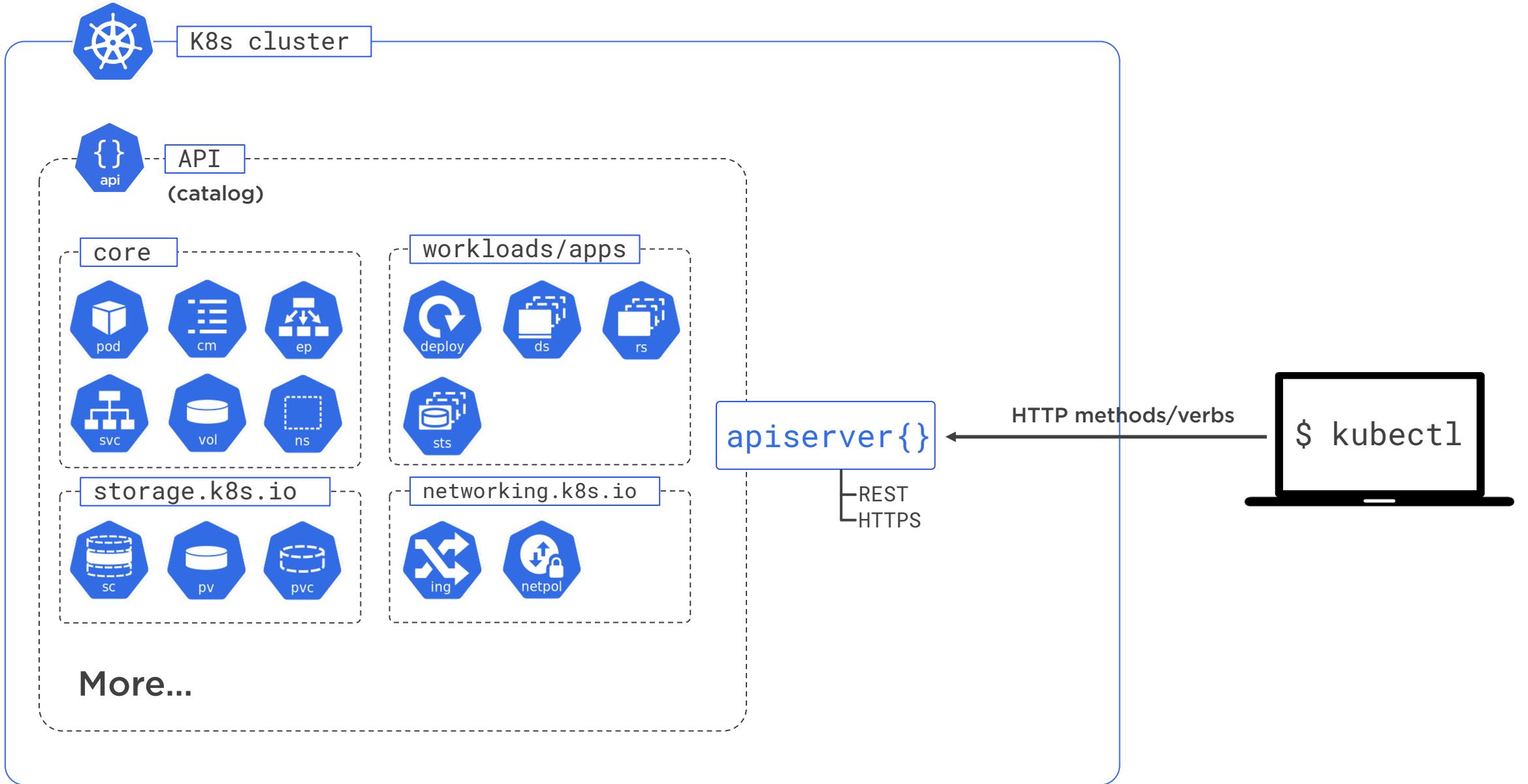










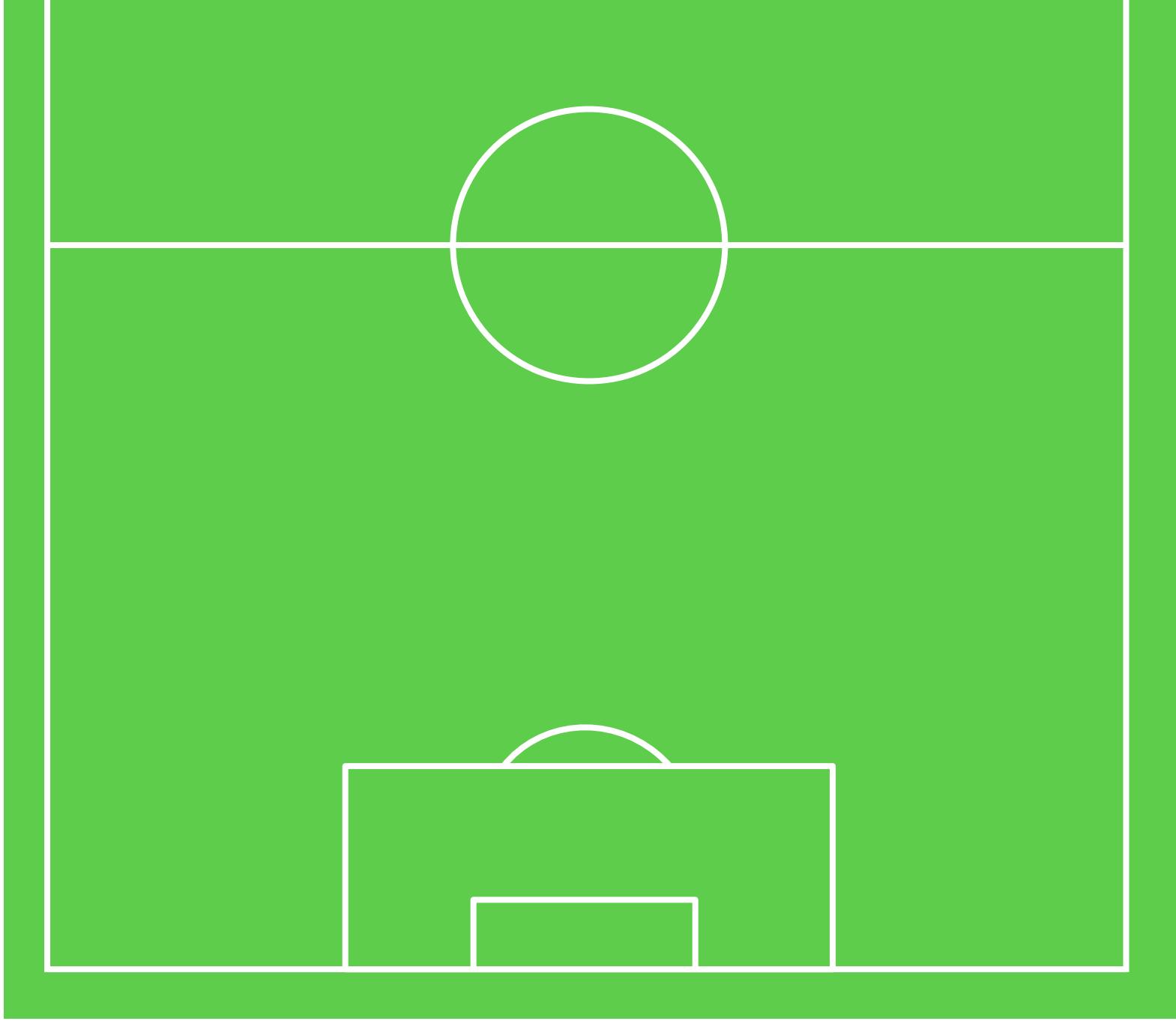


Up Next:
Epic Recap

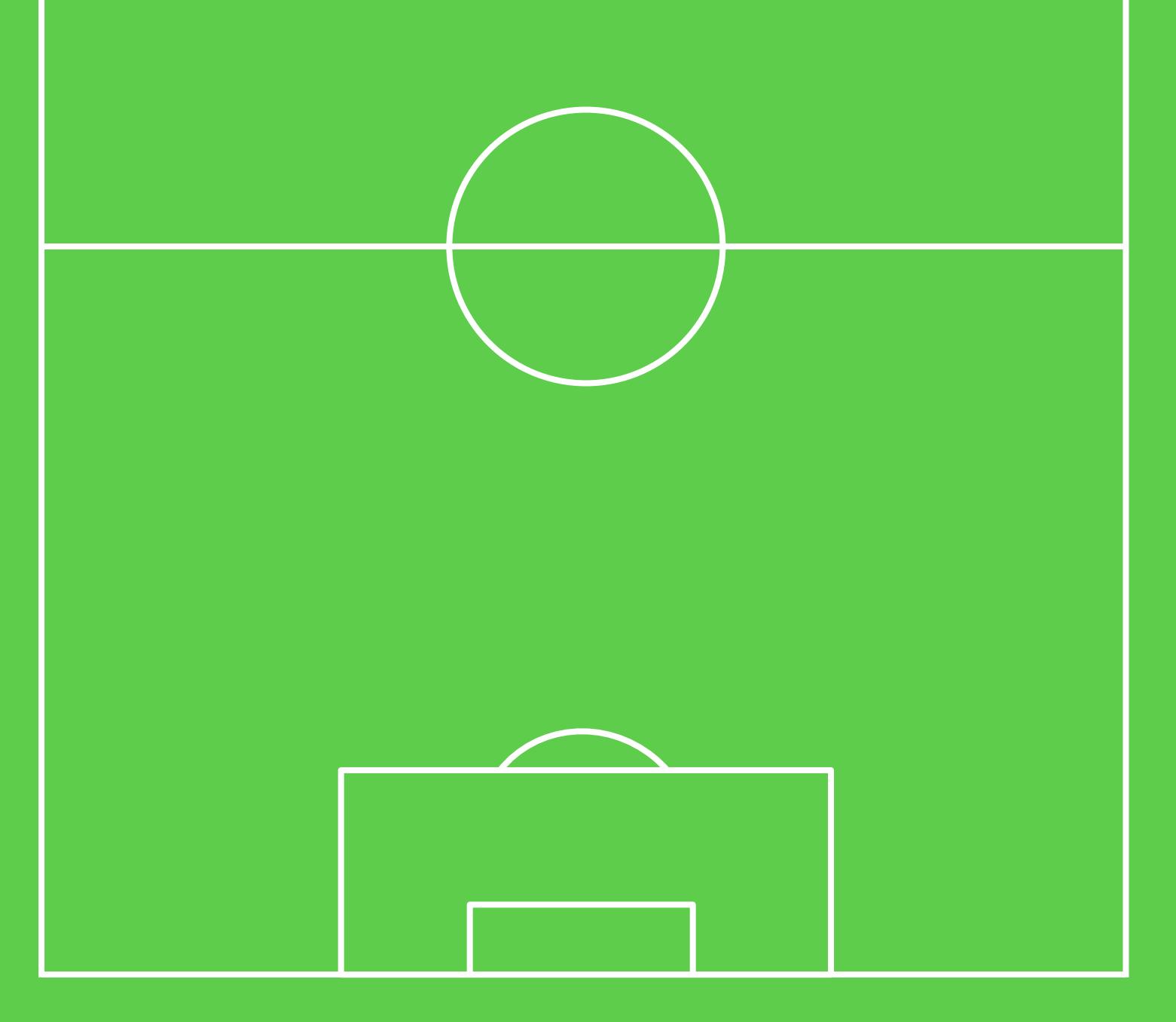
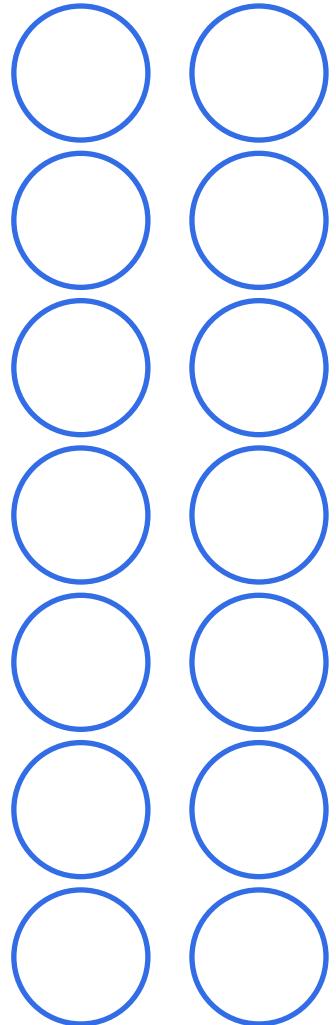


Epic Recap

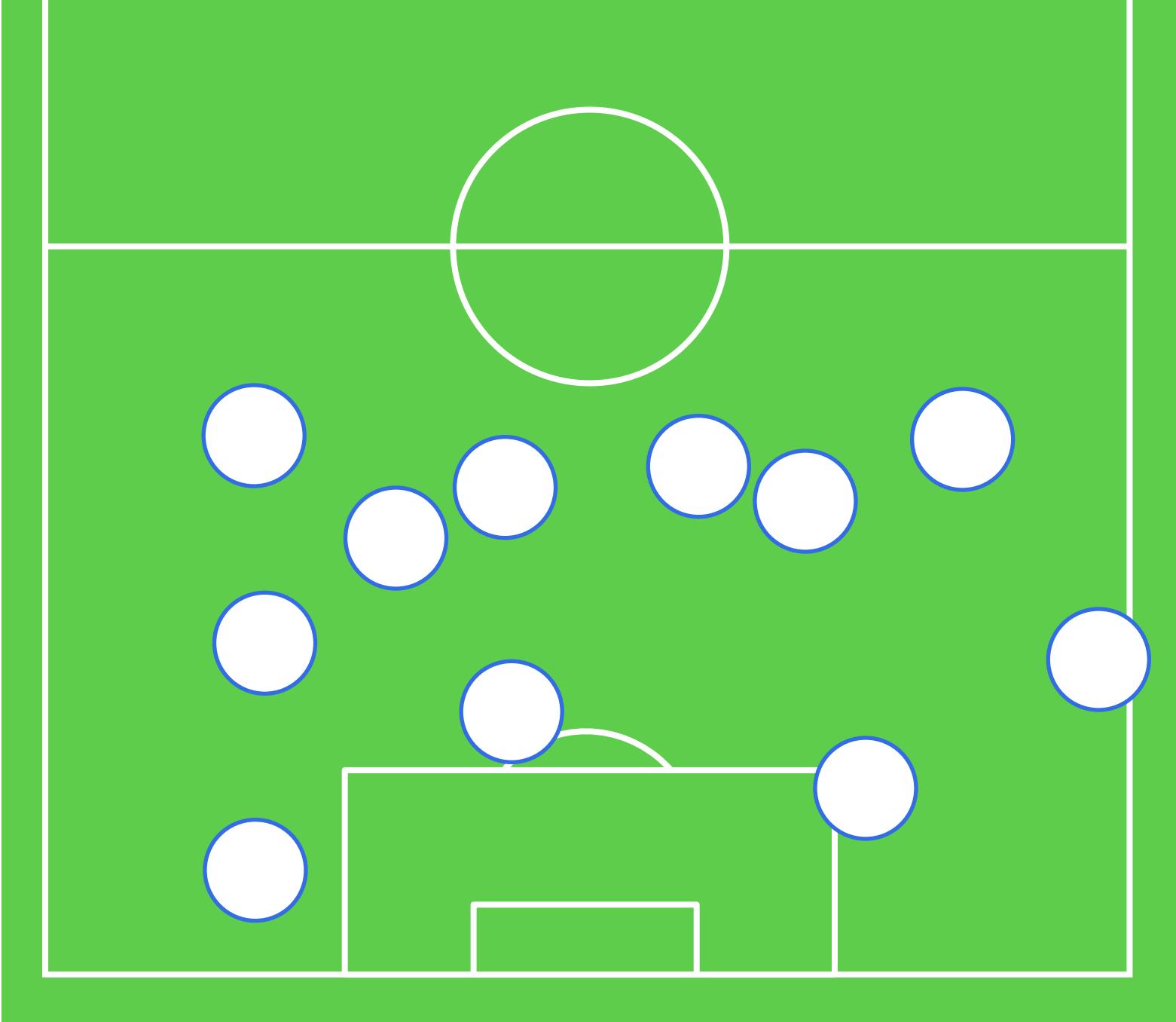
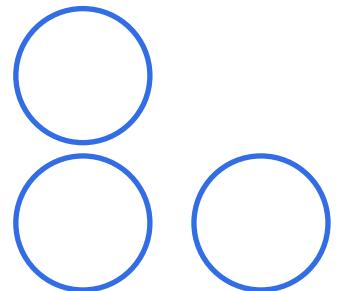




Team



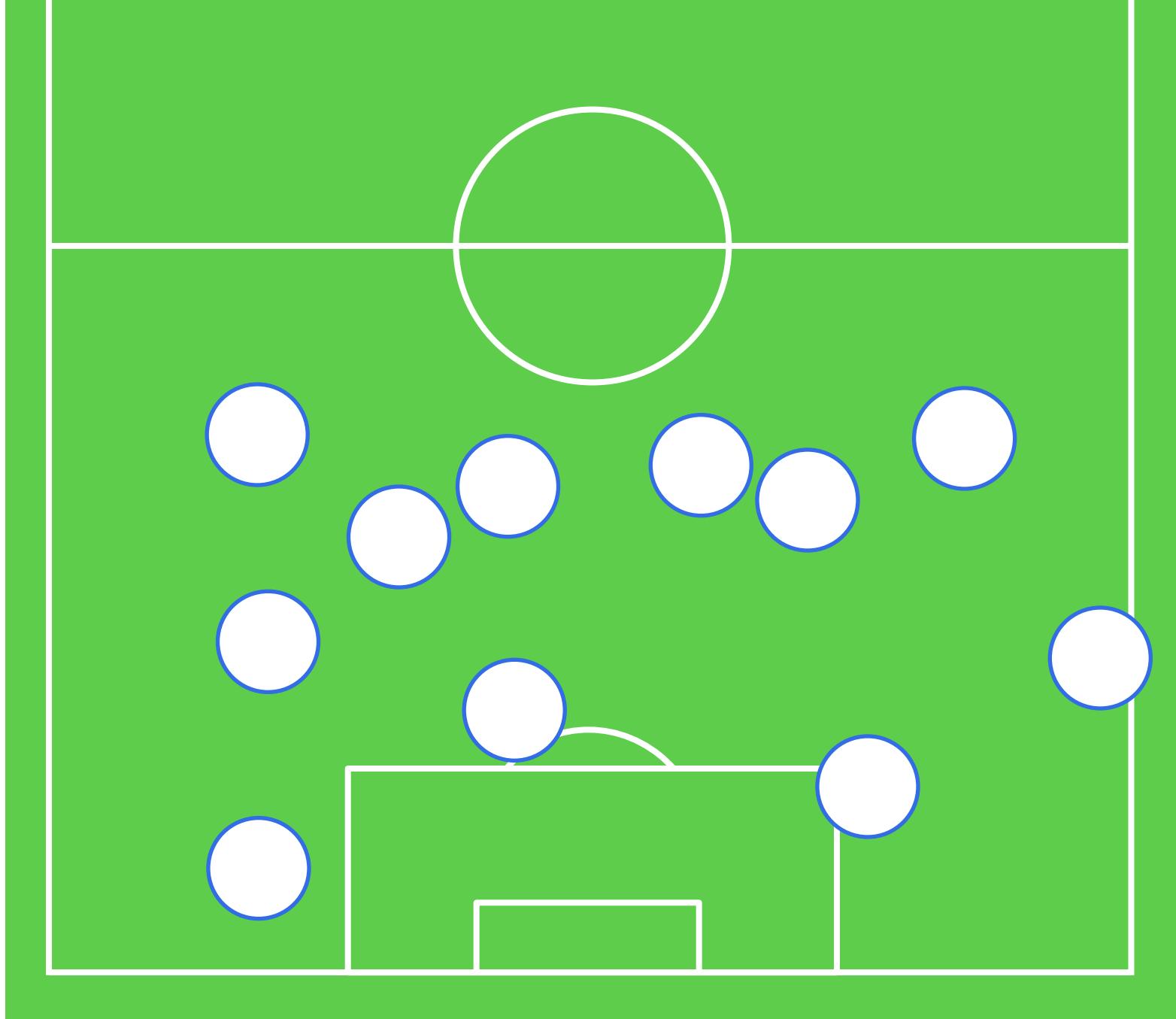
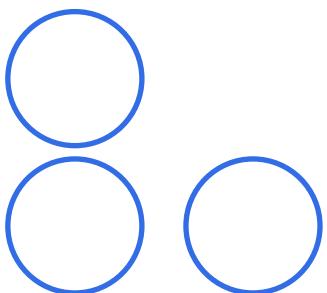
Team



Team



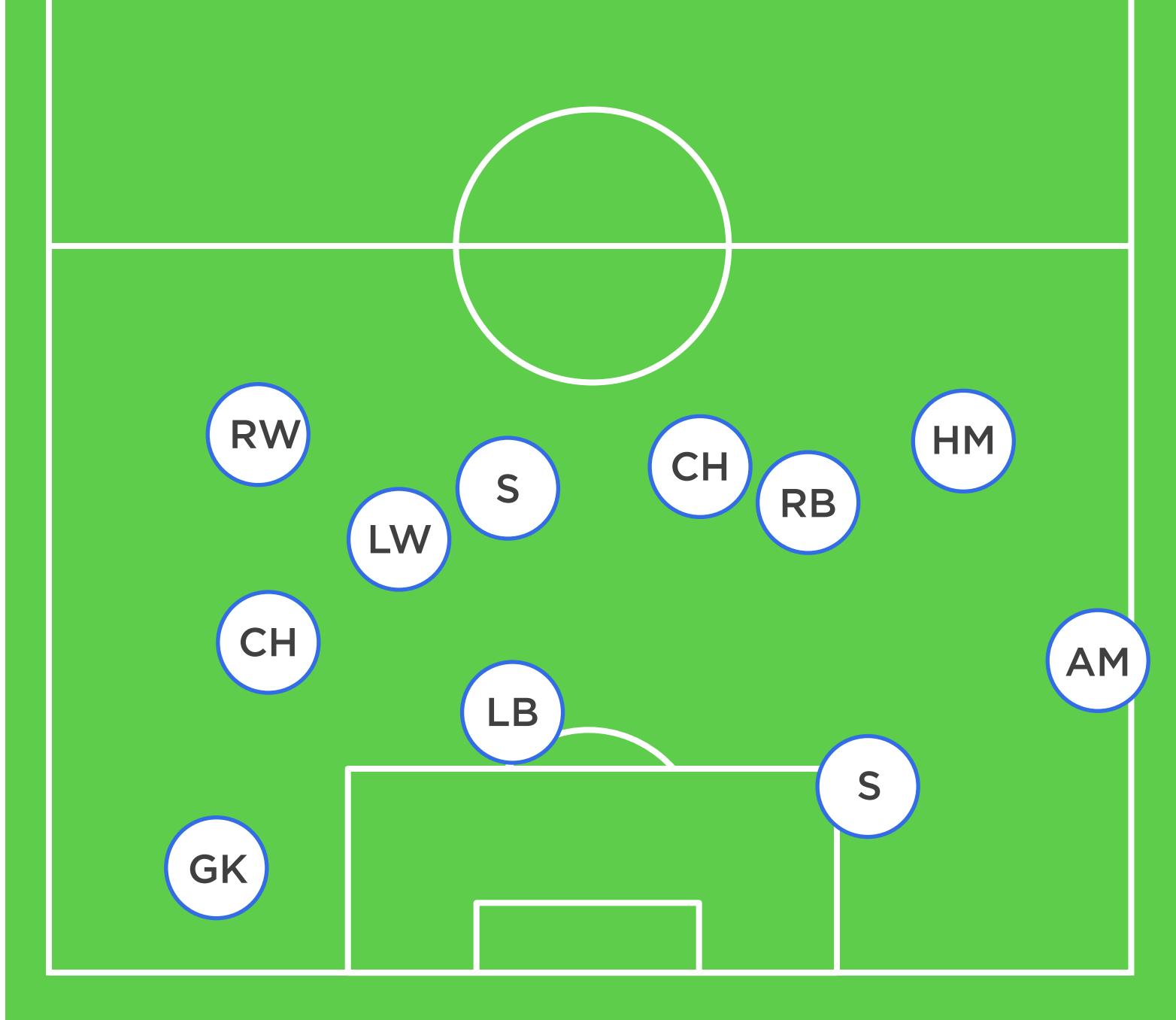
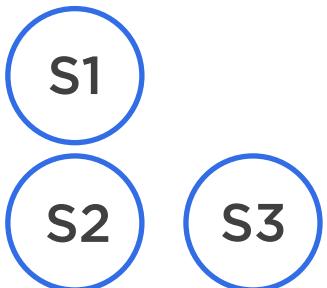
Manager
(coach)



Team



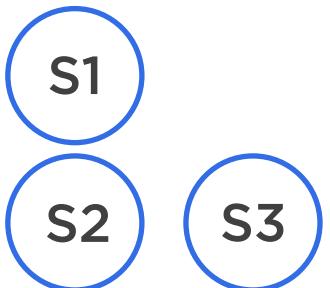
Manager
(coach)

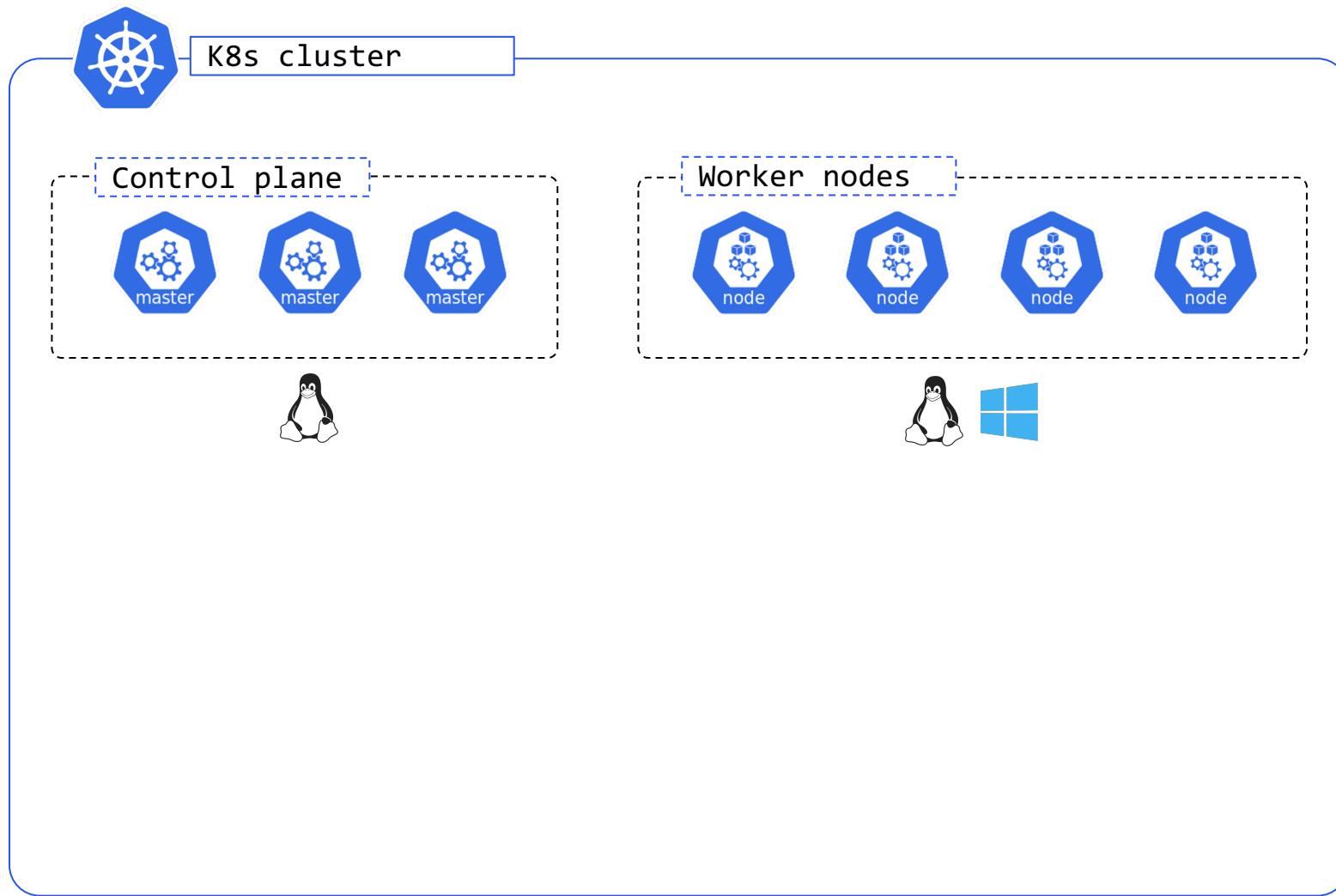


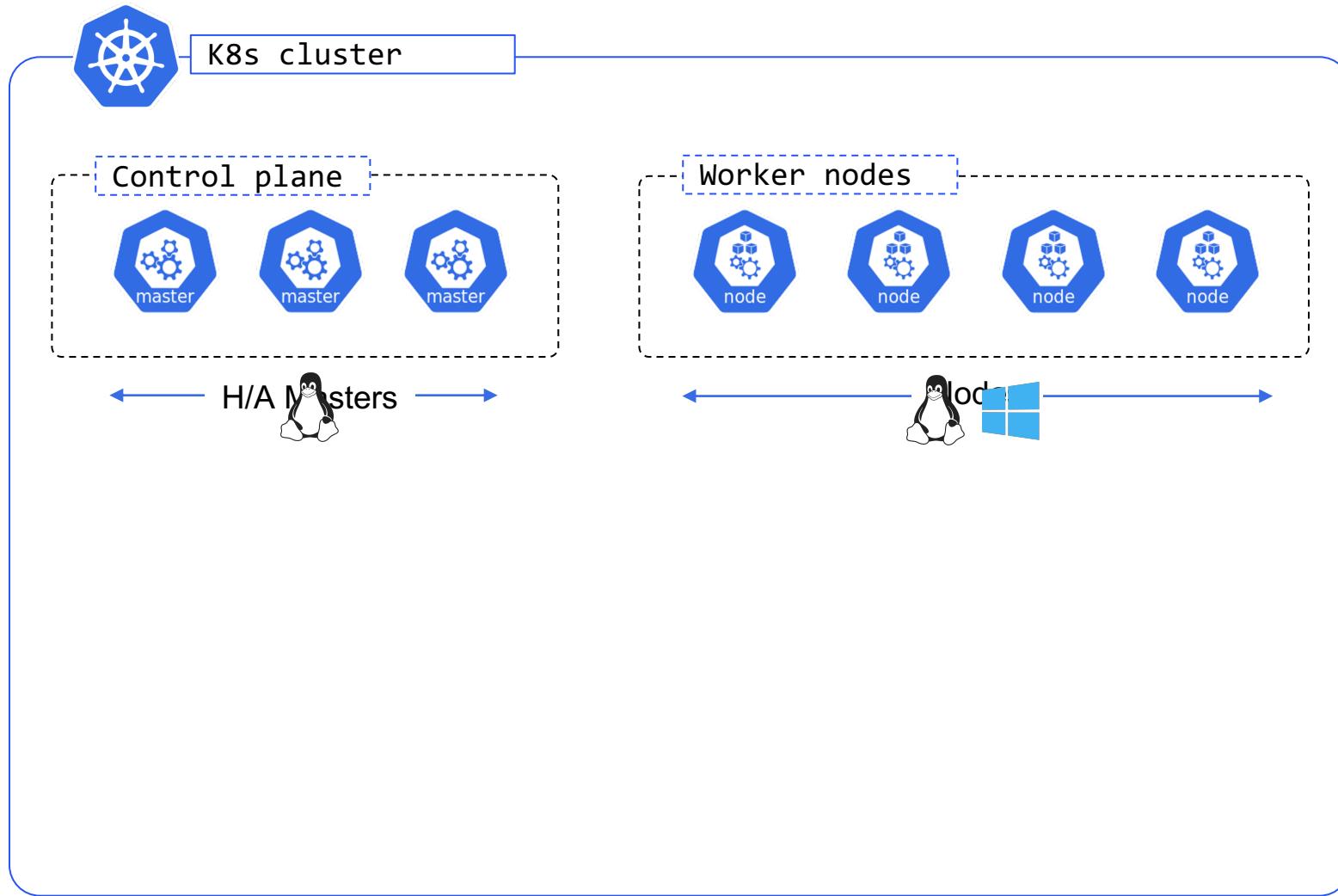
Team

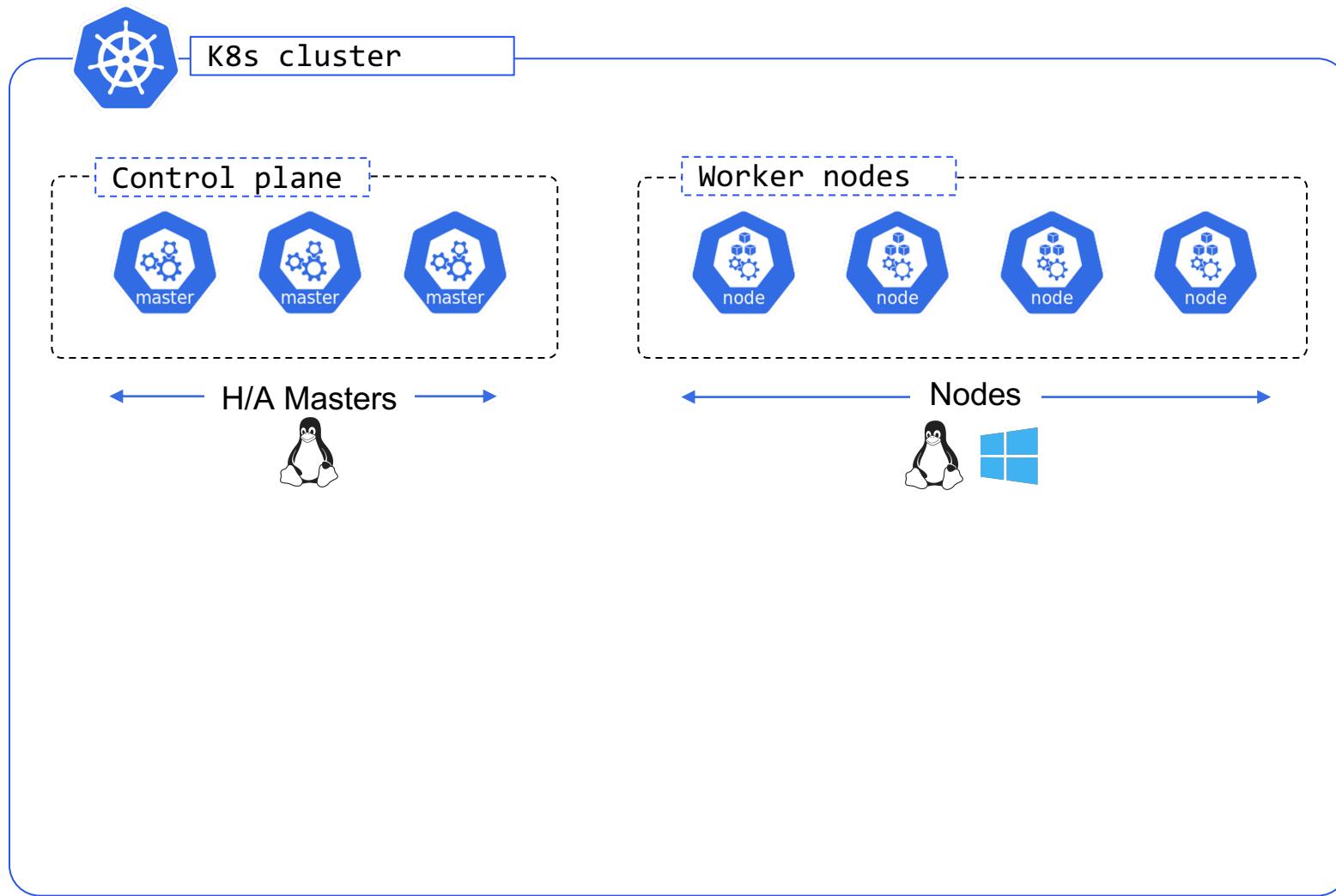


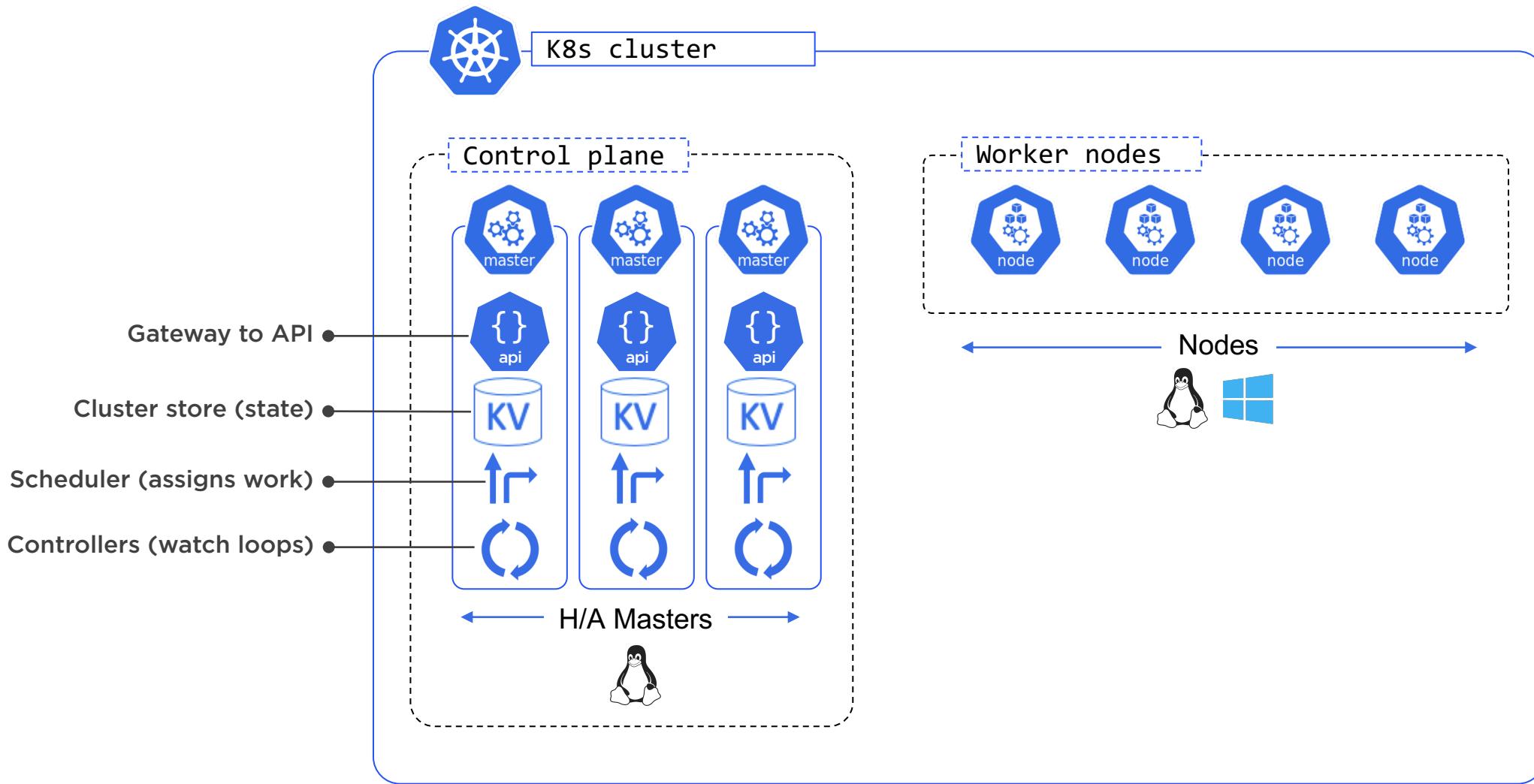
Manager
(coach)

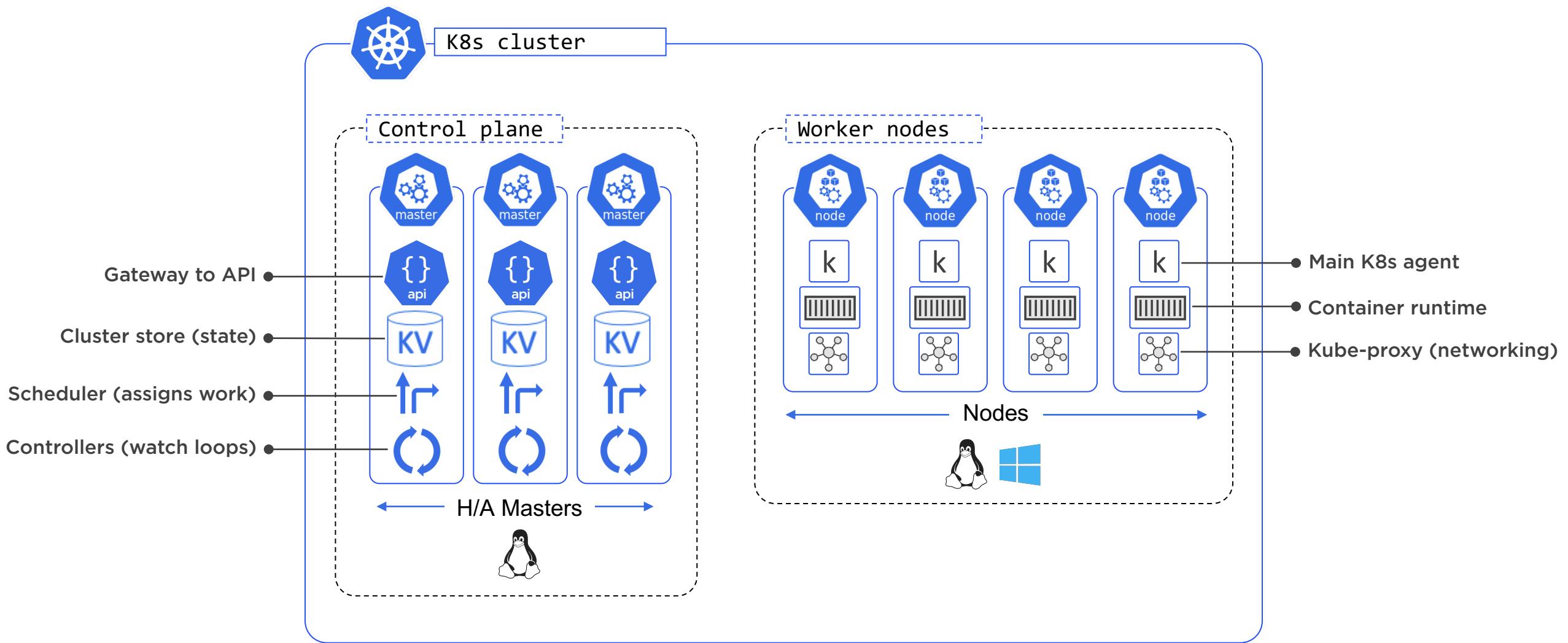


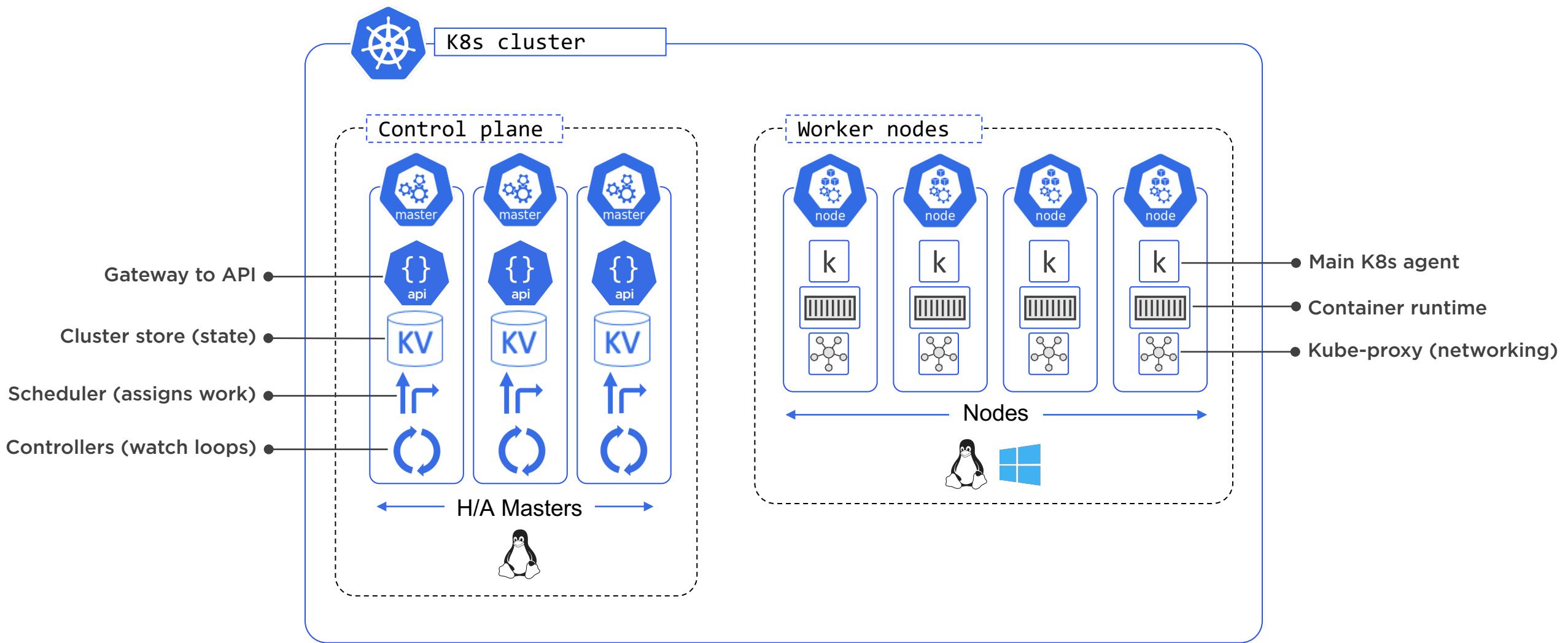


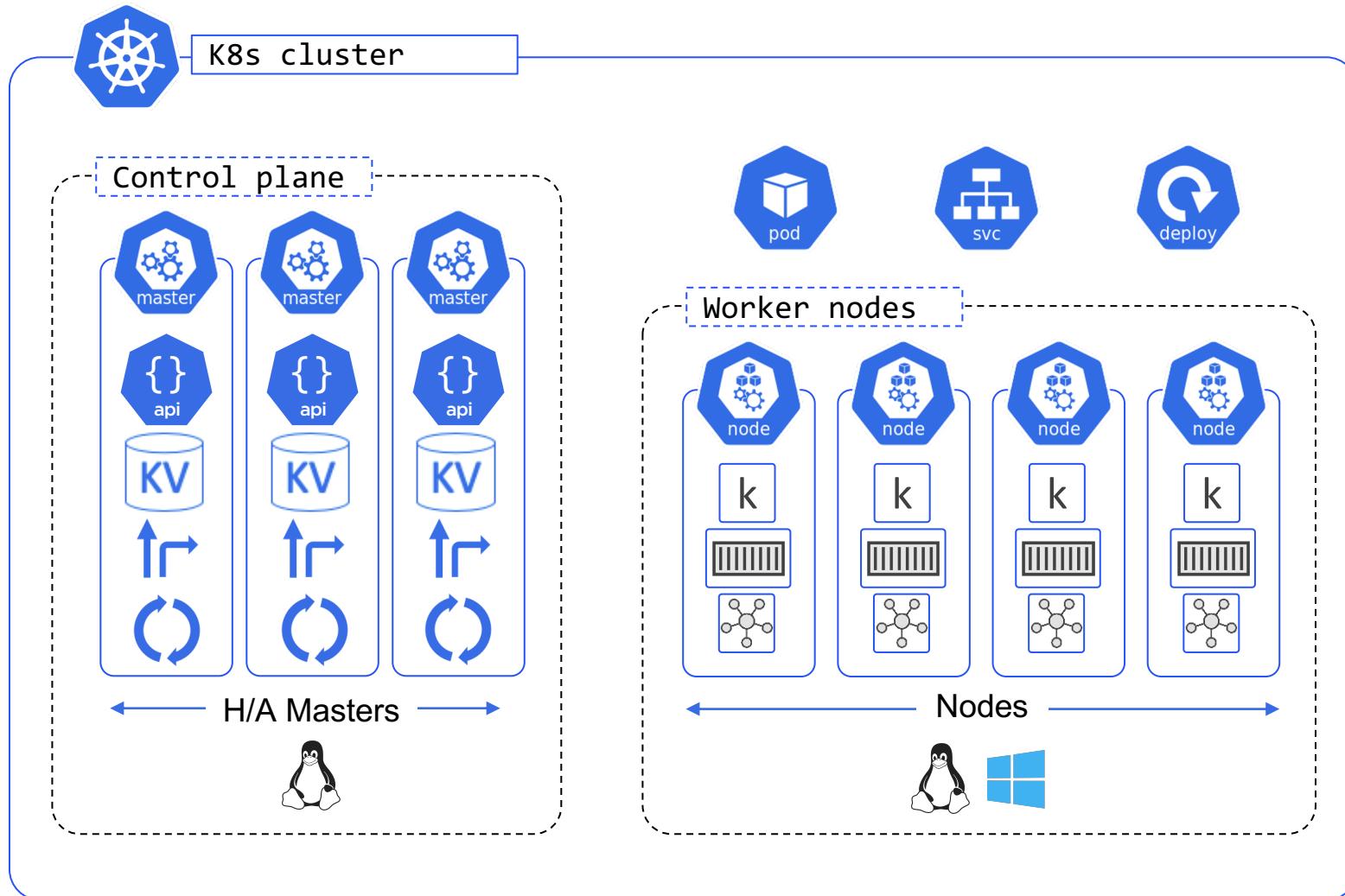












Up Next:
Getting Kubernetes



Getting Kubernetes



Nigel Poulton

@nigelpoulton www.nigelpoulton.com



Overview



Getting kubectl

Getting K8s on Your Laptop

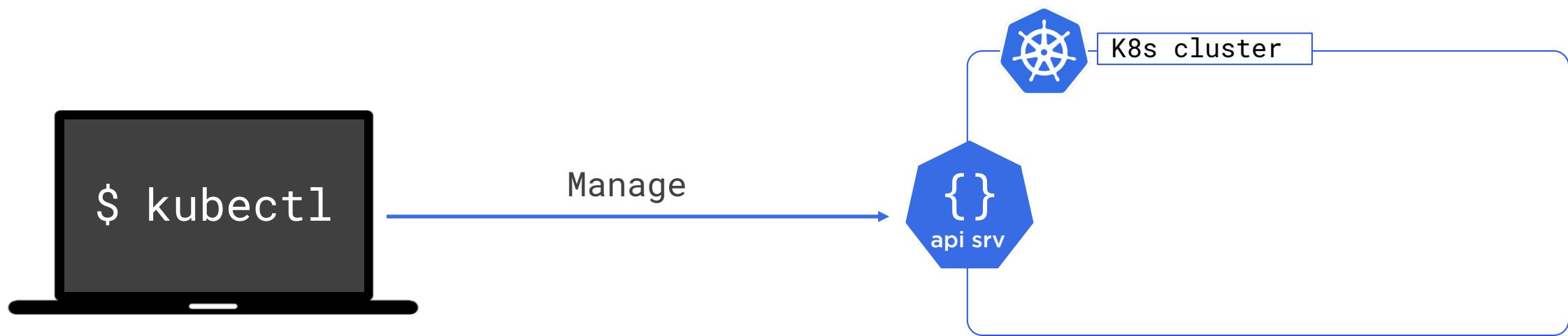
Getting K8s in the Cloud

Recap



Getting kubectl





Powershell

```
> Install-Script -Name 'install-kubectl' -Scope CurrentUser -Force  
> install-kubectl.ps1 c:\kubectl
```

Choco

```
> choco install Kubernetes-cli
```

Scoop

```
> scoop install kubectl
```



Up Next:
Getting K8s on Your Laptop



Getting K8s on Your Laptop



Kubernetes on Your Laptop

Not for production!

Requires
virtualization in
BIOS and OS

The way to install
stuff changes over
time



Beware of cloud costs!

I've got the scars...



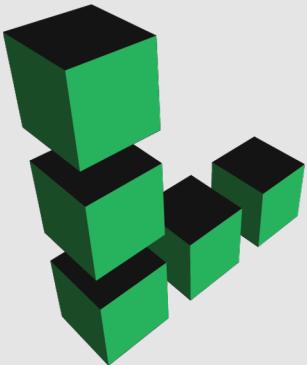
Up Next:
Getting K8s in the Cloud



Getting K8s in the Cloud



K8s in the Cloud



**Linode Kubernetes Engine
(LKE)**



**Google Kubernetes Engine
(GKE)**

*Lots of other great options exist!



Recap



Up Next:
Rocking It with Kubernetes Pods



Working with Pods



Nigel Poulton

@nigelpoulton www.nigelpoulton.com



Overview



App Deployment Workflow

Creating a Pod Manifest (YAML)

Deploying a Pod

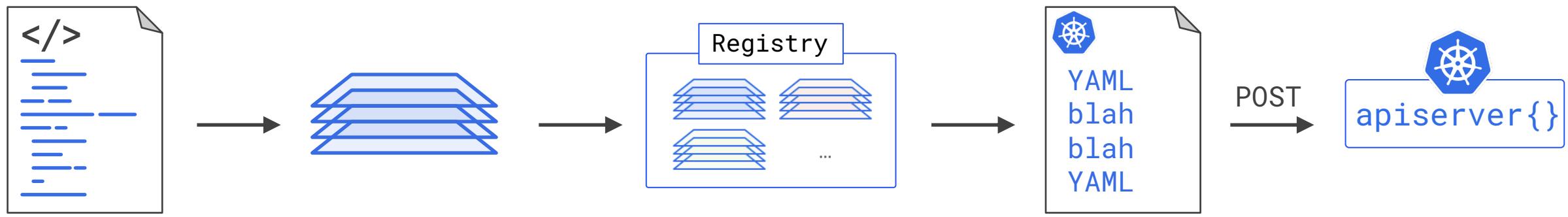
Multi-container Pod example

Recap



App Deployment Workflow





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)



Getting Started with Docker

By Nigel Poulton

Beginner · ★★★★☆ (1103)

Docker Deep Dive

By Nigel Poulton

Intermediate · ★★★★★ (529)



Up Next:
Creating a Pod Manifest (YAML)



Creating a Pod Manifest (YAML)



Alpha

v1alpha1
v1alpha2



Alpha

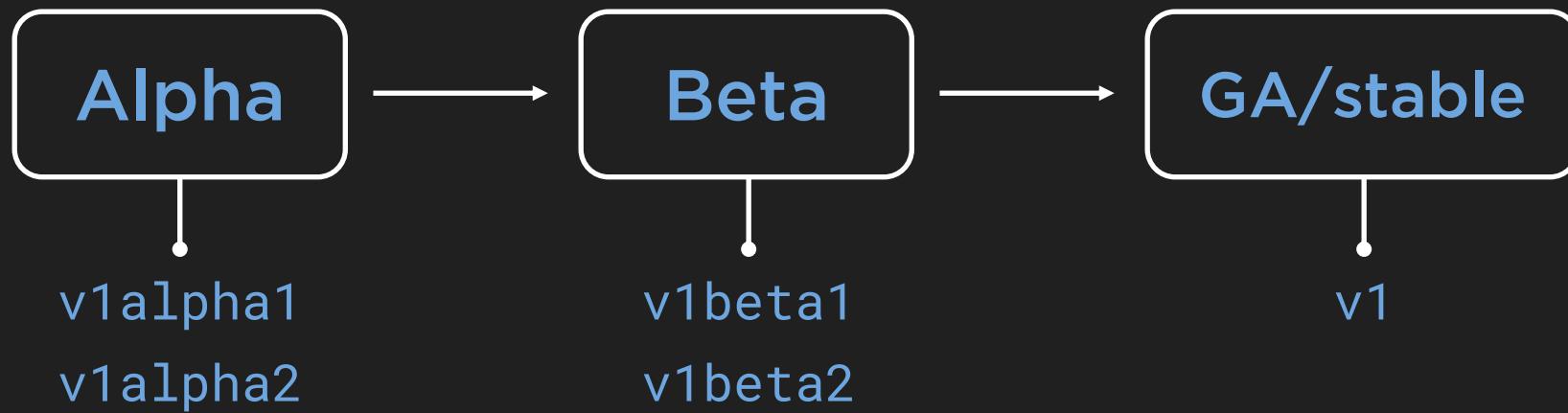
v1alpha1
v1alpha2

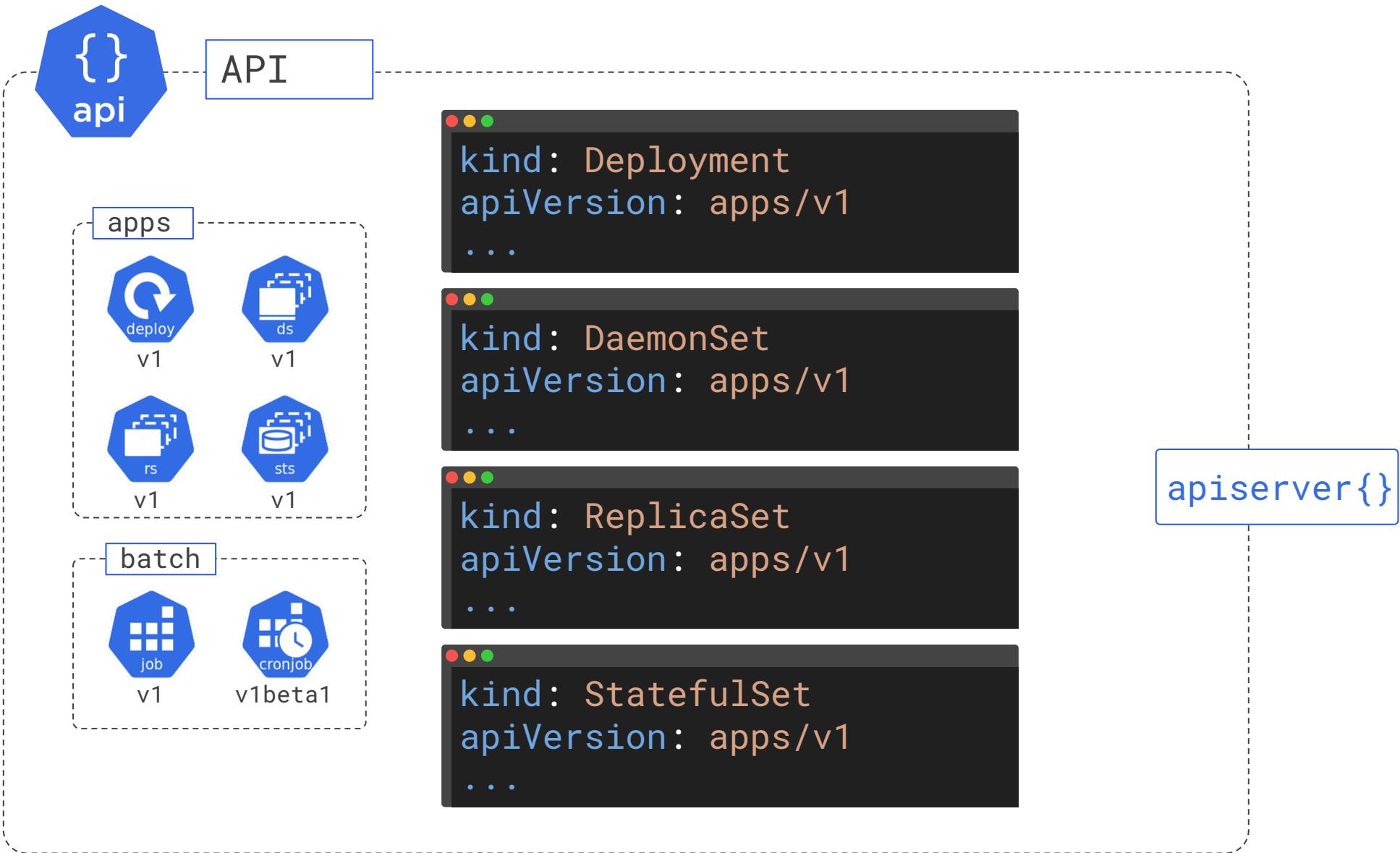


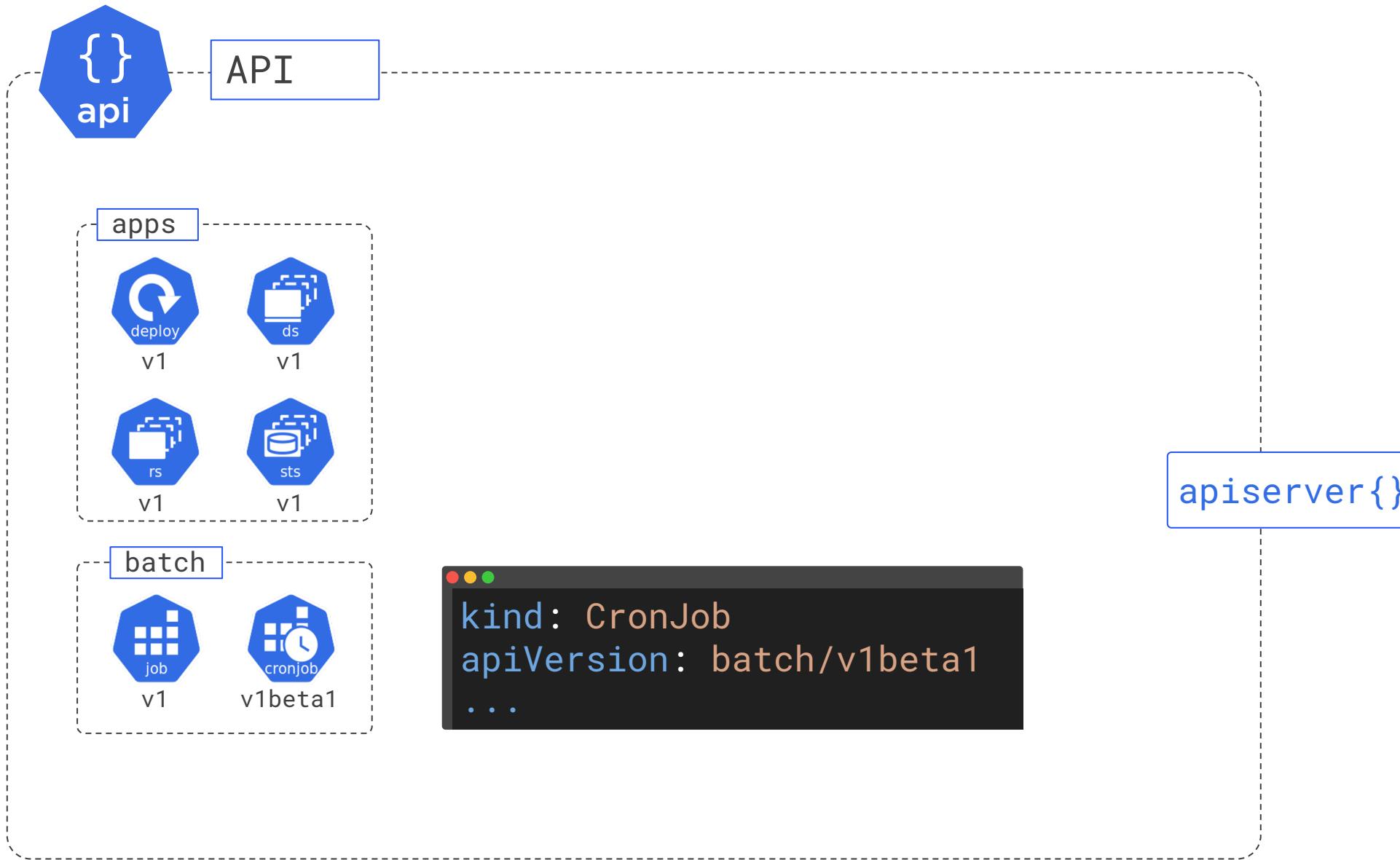
Alpha

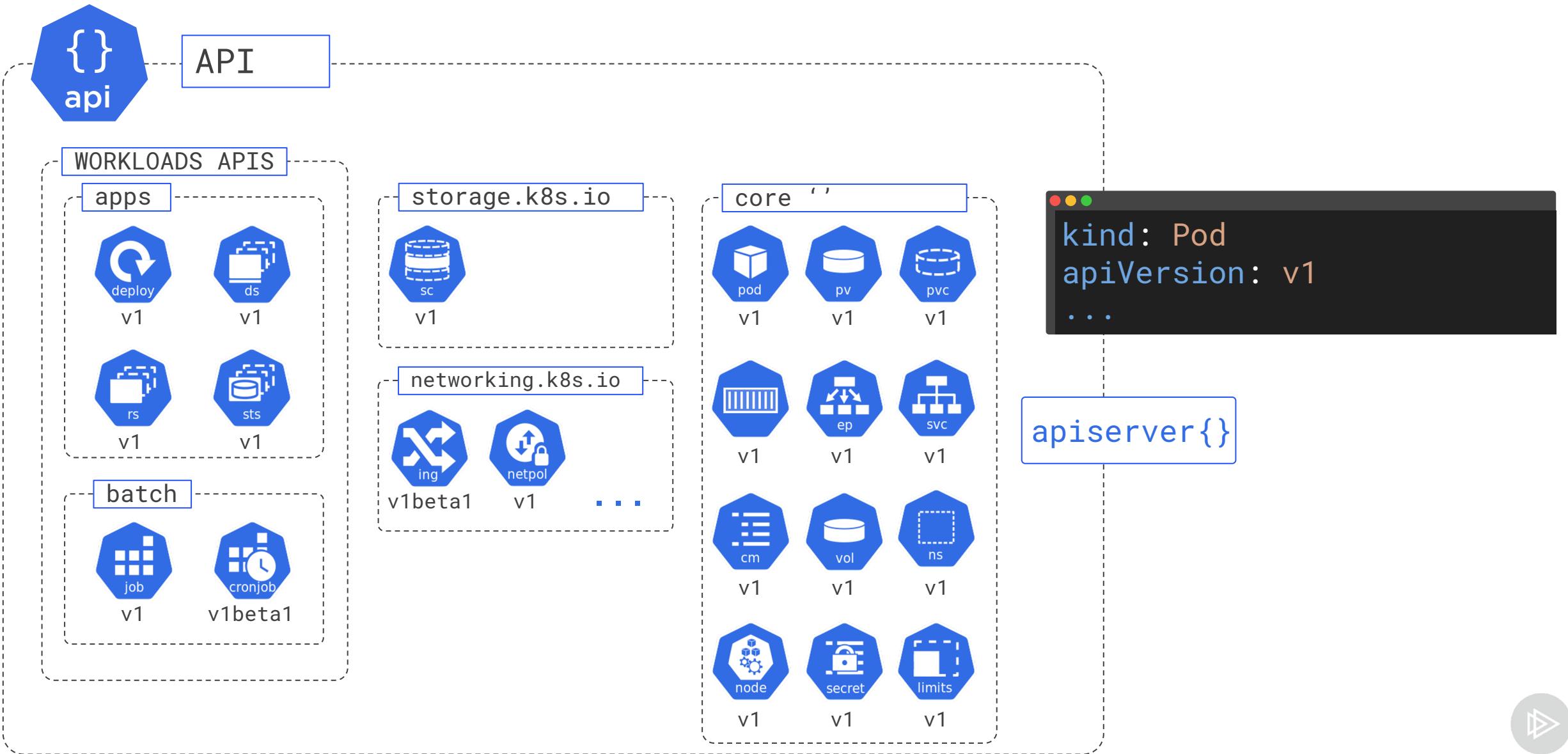
v1alpha1
v1alpha2

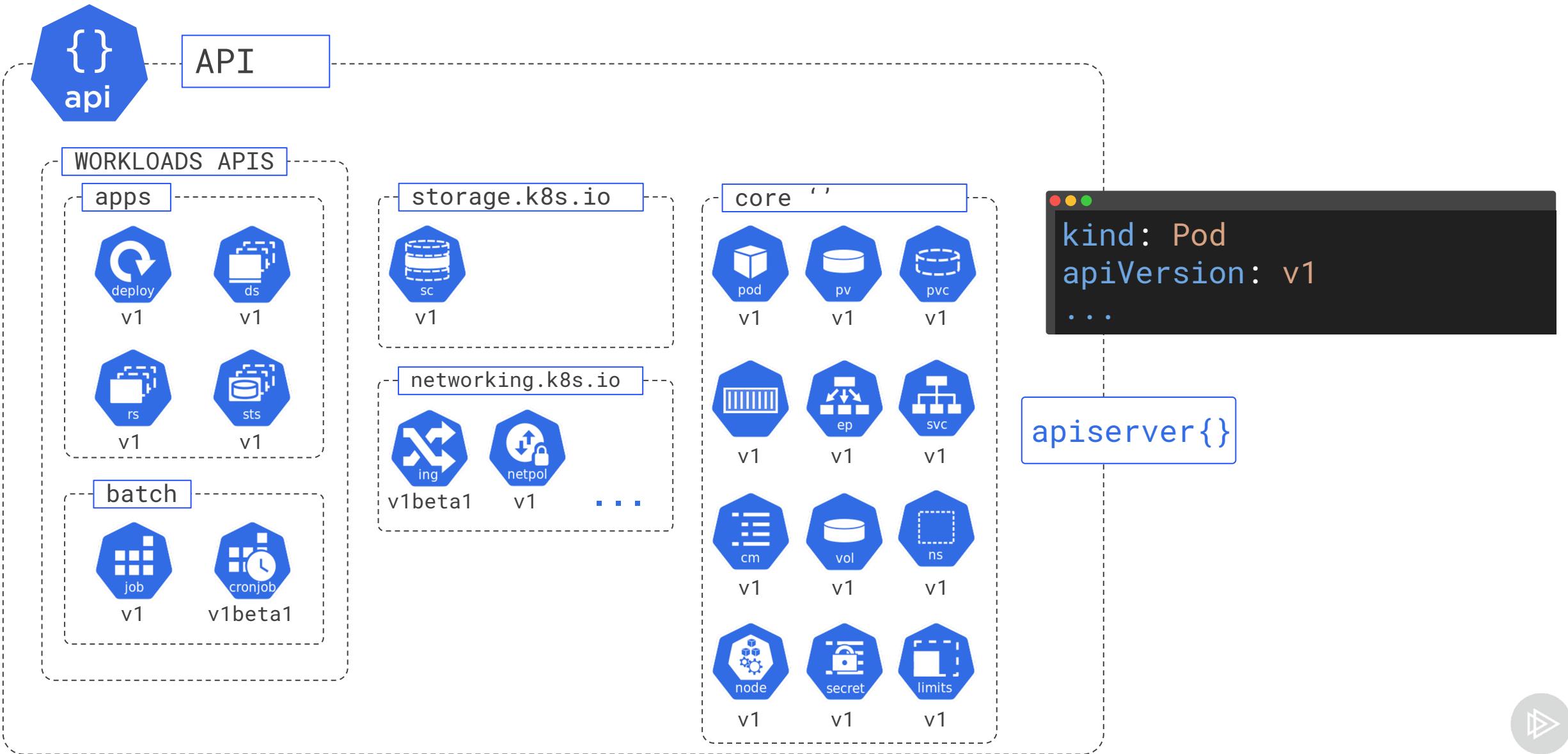


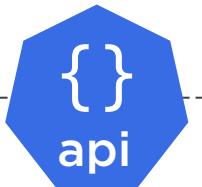












API



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)

apiserver{}





API



Pod object definition

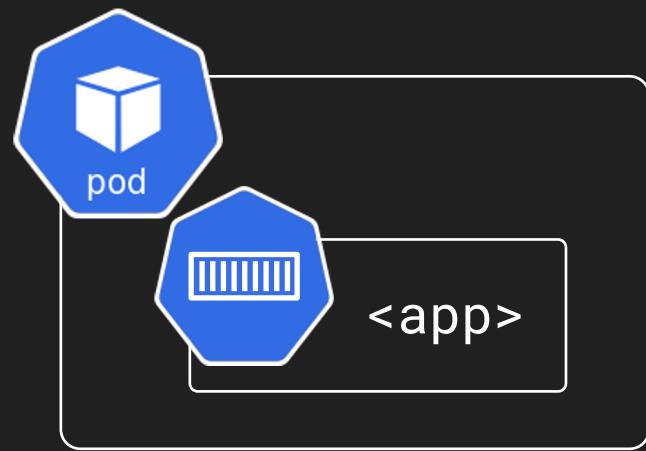
- API sub-group: [core ""]
- metadata (ObjectMeta)
 - name (string)
 - annotations (object)
 - labels (object)
- spec (PodSpec)
 - nodeName (integer)
 - containers (container array)
 - name (string)
 - image (string)
 - imagePullPolicy (string)
 - ports (ContainerPort array)
 - ...

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{}





Up Next:
Deploying a Pod



Deploying a Pod

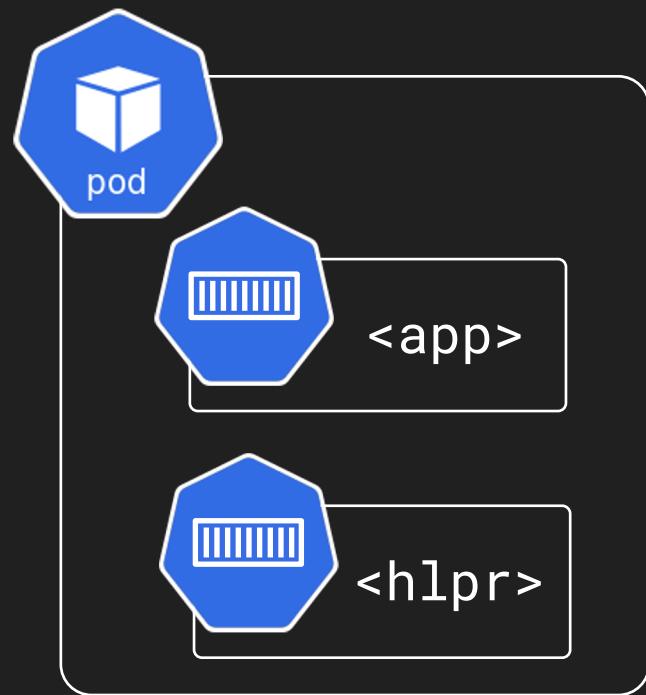


Up Next:
Multi-container Pod Example



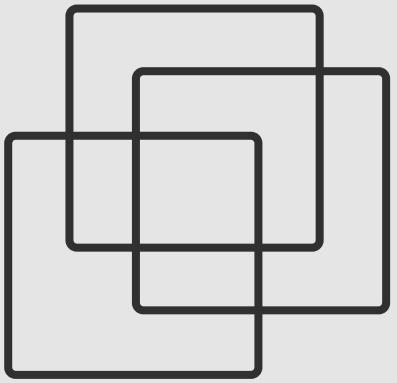
Multi-container Pod Example



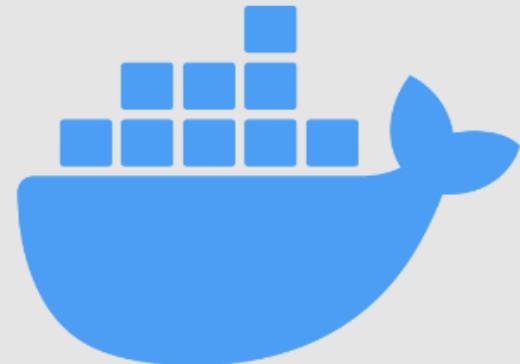


Quick Recap





Virtual Machine

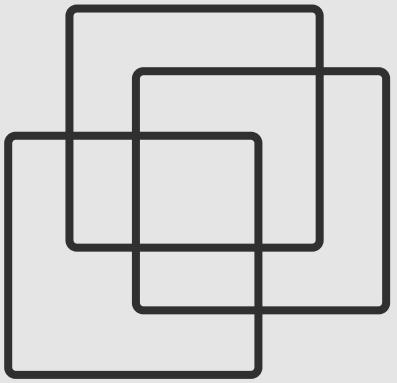


Container

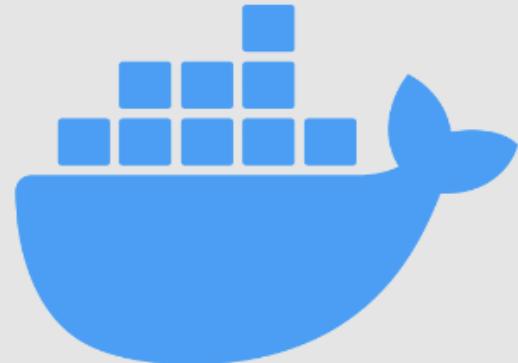


Pod





Virtual Machine

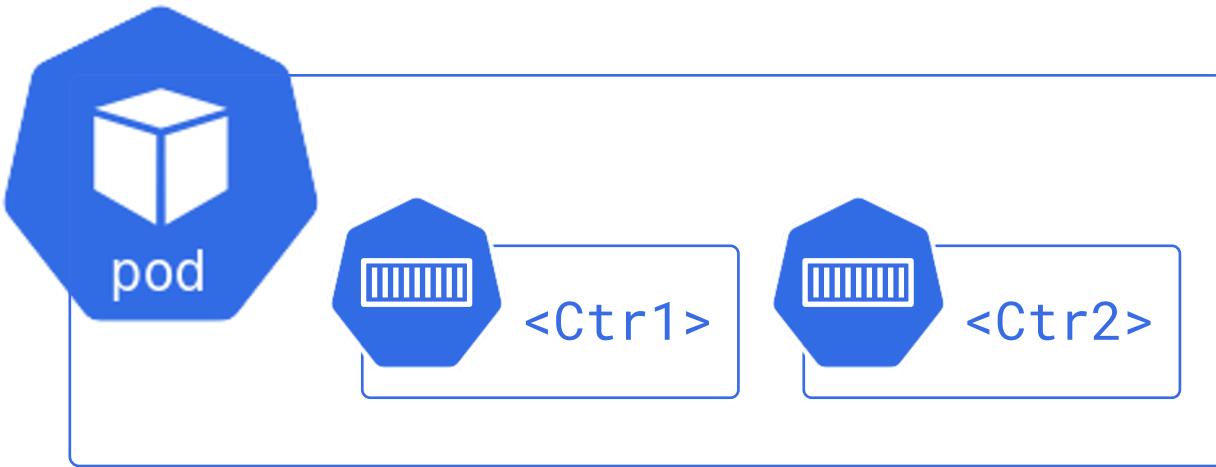


Container



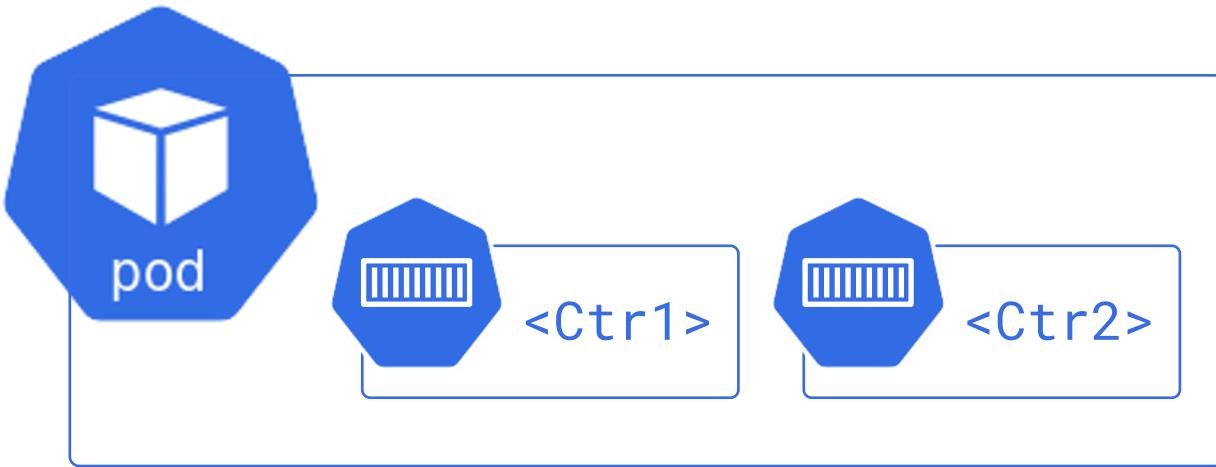
Pod





Node





Node



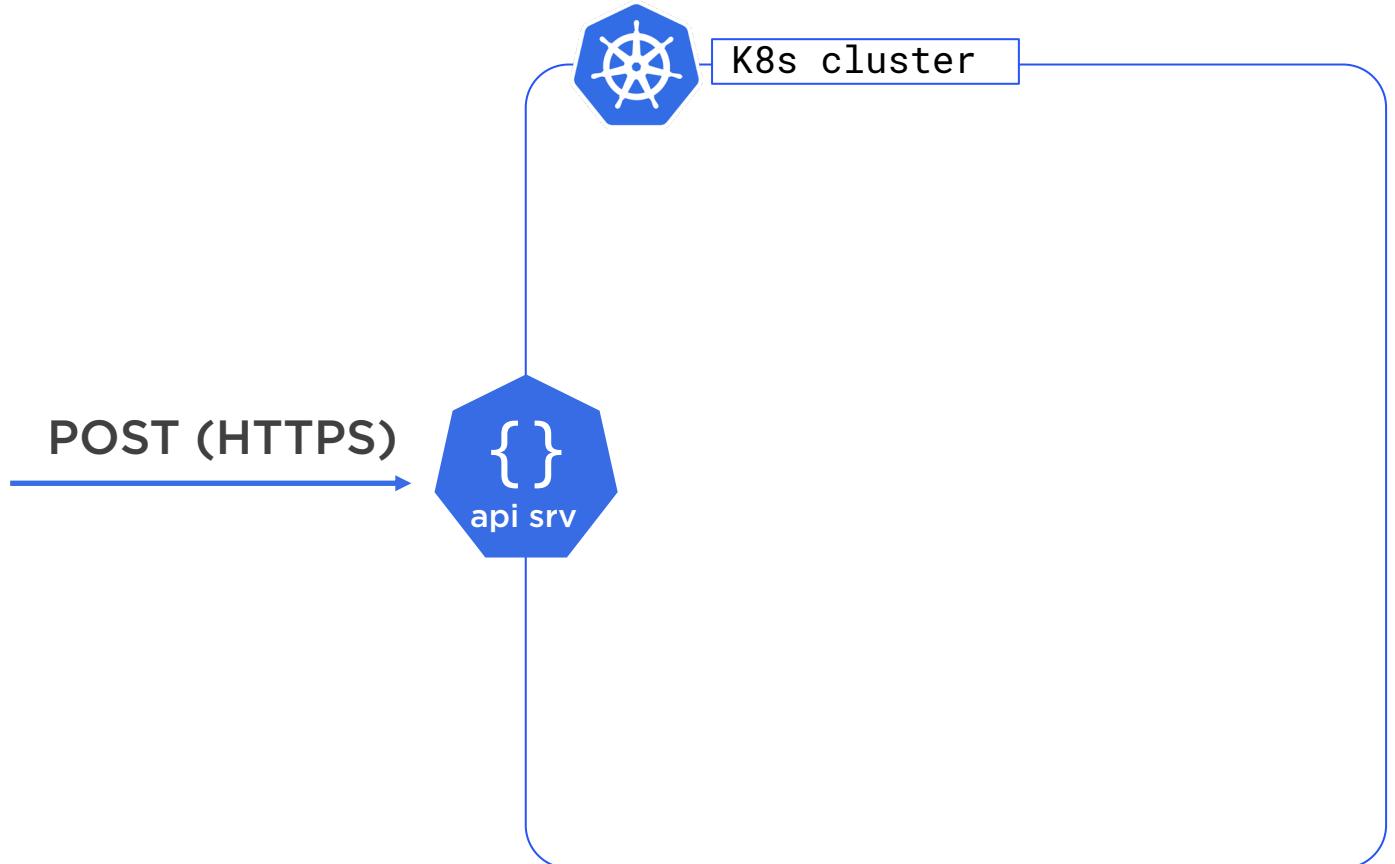
```
kind: Pod
apiVersion: v1
metadata:
  name: test-pod
  labels:
    ver: 1.0
spec:
  containers:
  - name: main
    image: web-server:1.0
    imagePullPolicy: Always
    ports:
      containerPort: 8080
```



```
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
```



```
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
```



Up Next:
Kubernetes Networking with Services



Kubernetes Networking with Services



Nigel Poulton

@nigelpoulton www.nigelpoulton.com



Overview



Theory

Create a Service Imperatively

Create a Service Declaratively

Create an Internet LoadBalancer Service

Recap



Kubernetes Service Theory

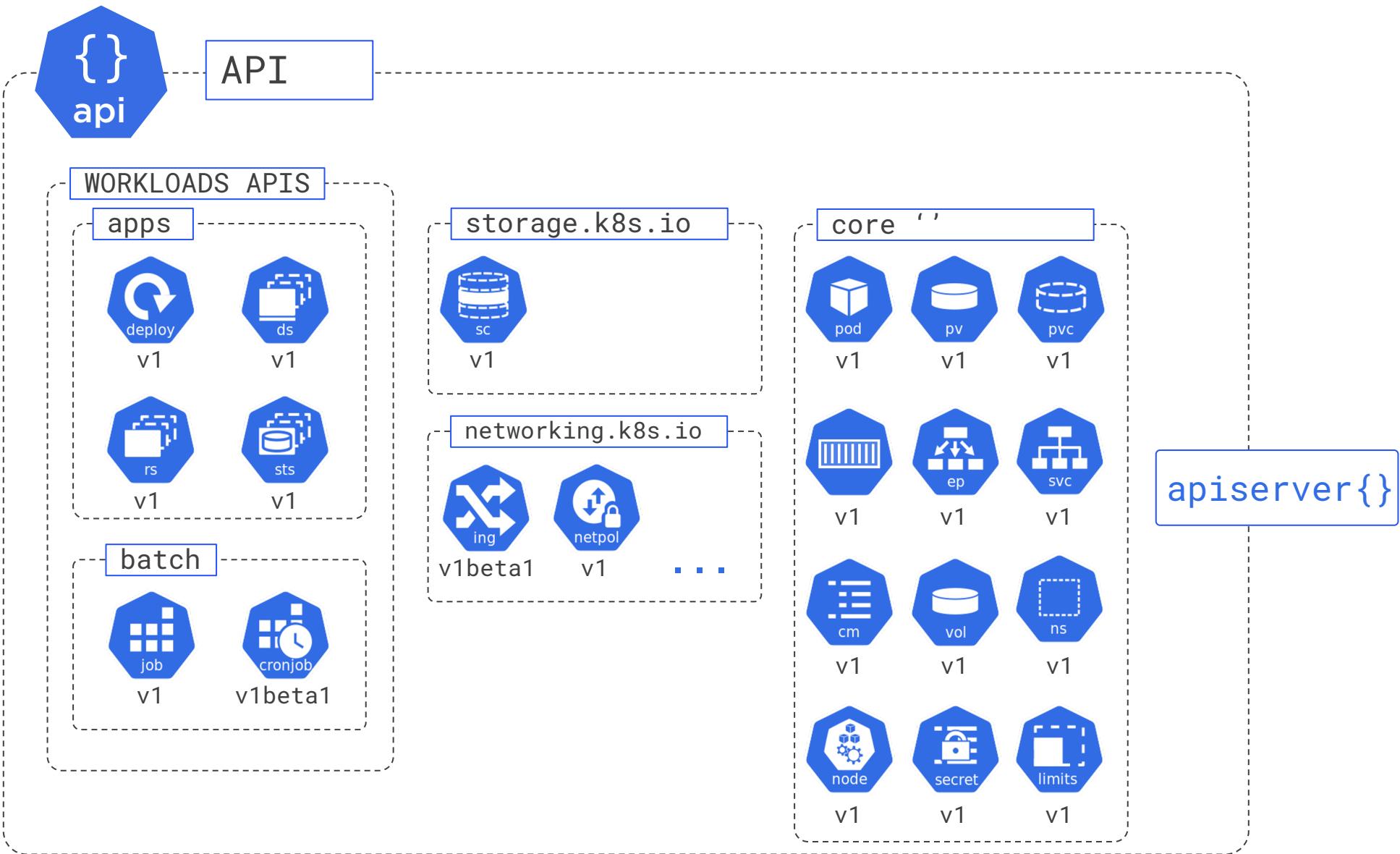


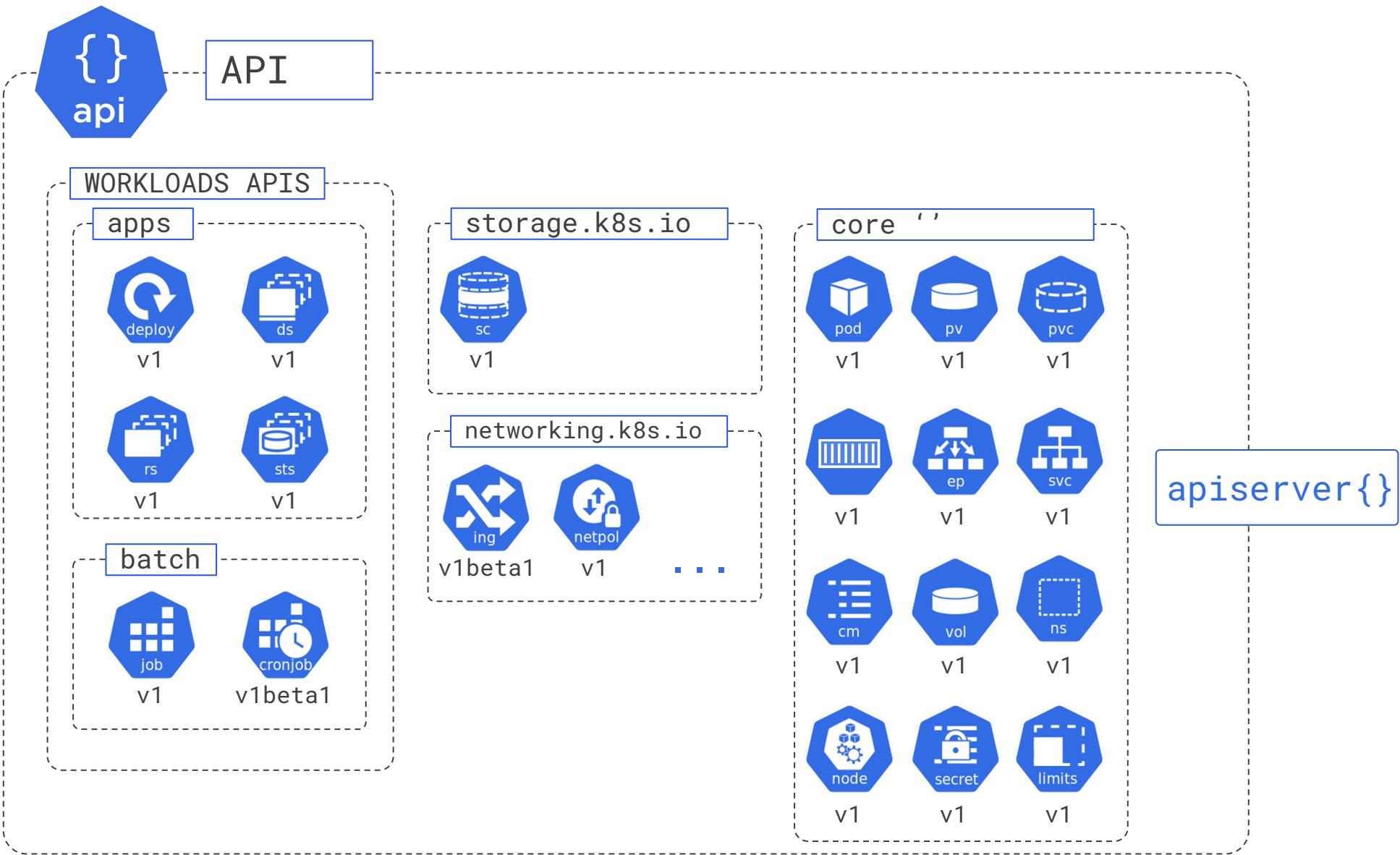
```
$ git clone https://github.com/nigelpoulton/getting-started-k8s.git  
$ cd getting-started-k8s/Pods  
$ kubectl apply -f pod.yml
```

Getting the Code

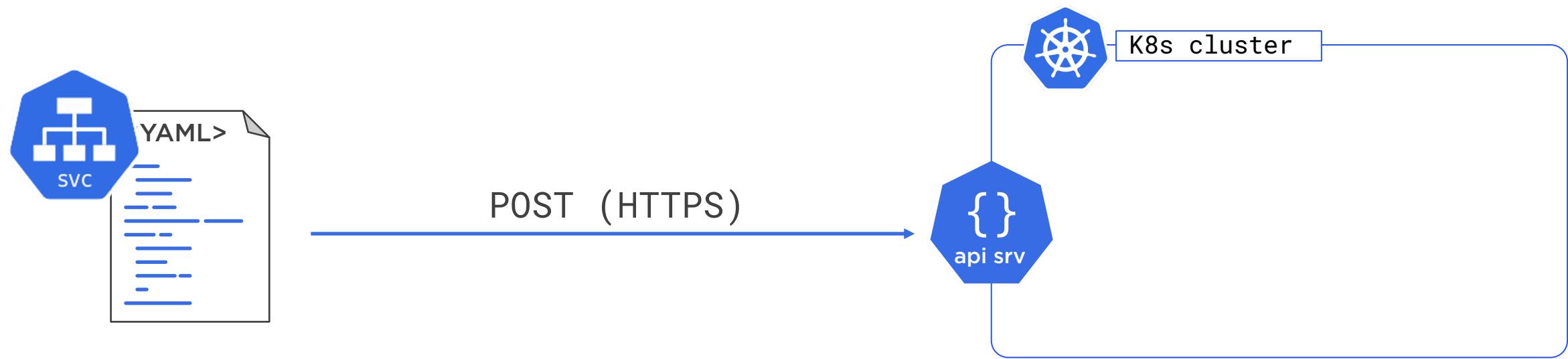
You'll need **git** installed, and **kubectl** configured to talk to your K8s cluster.

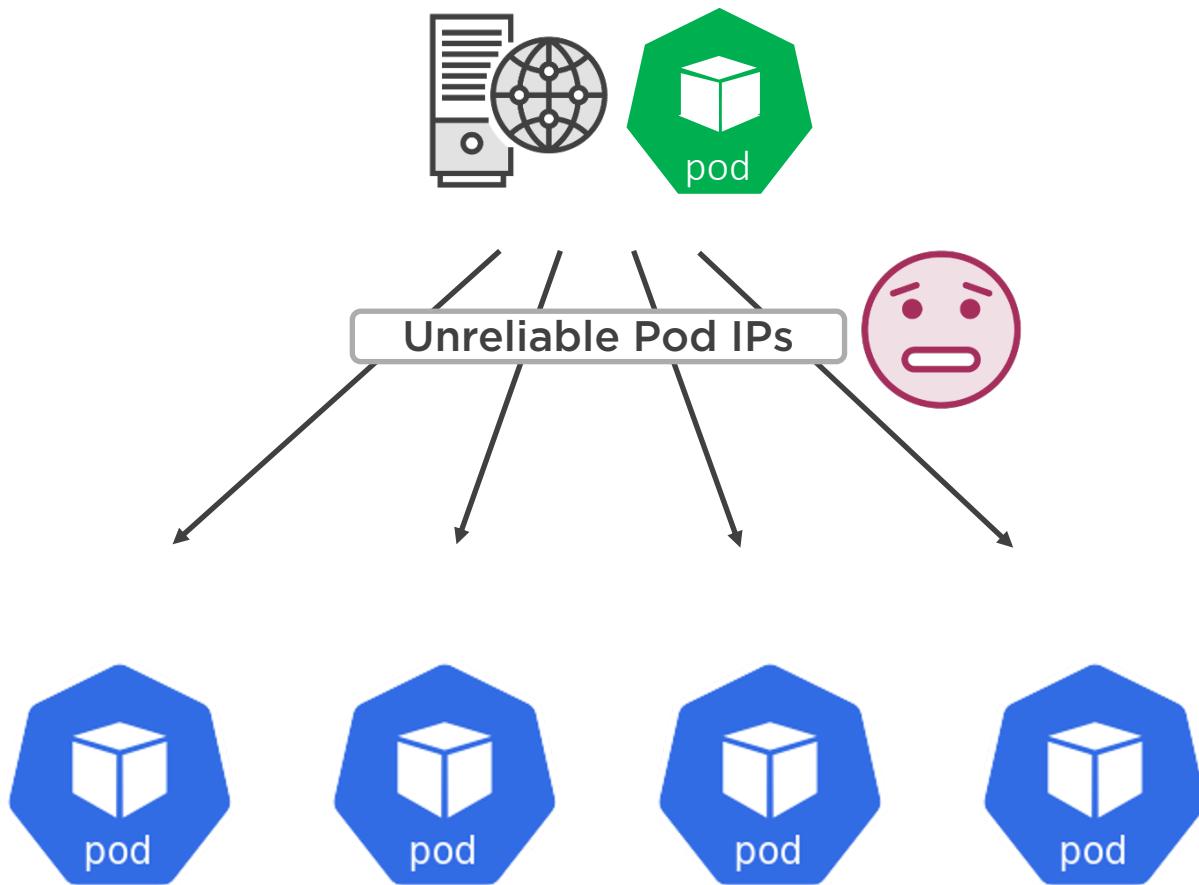


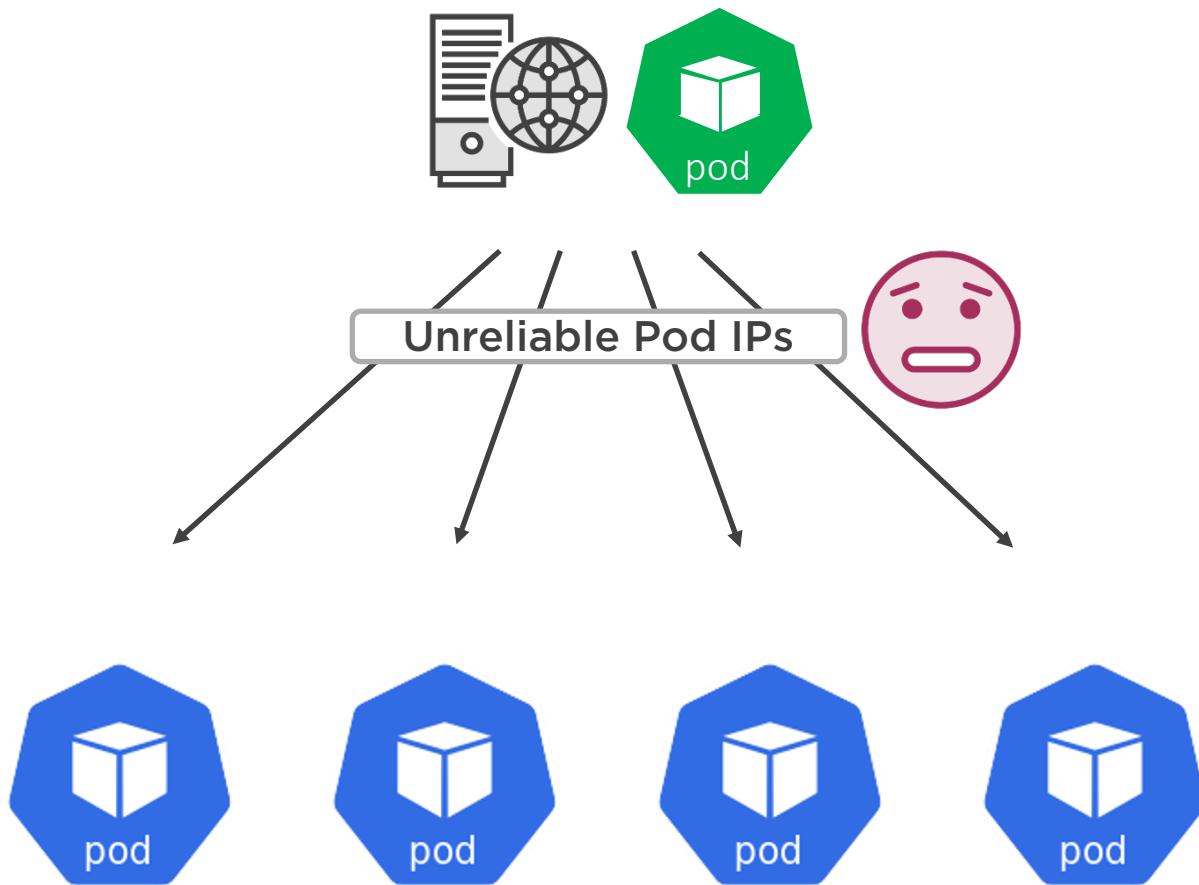


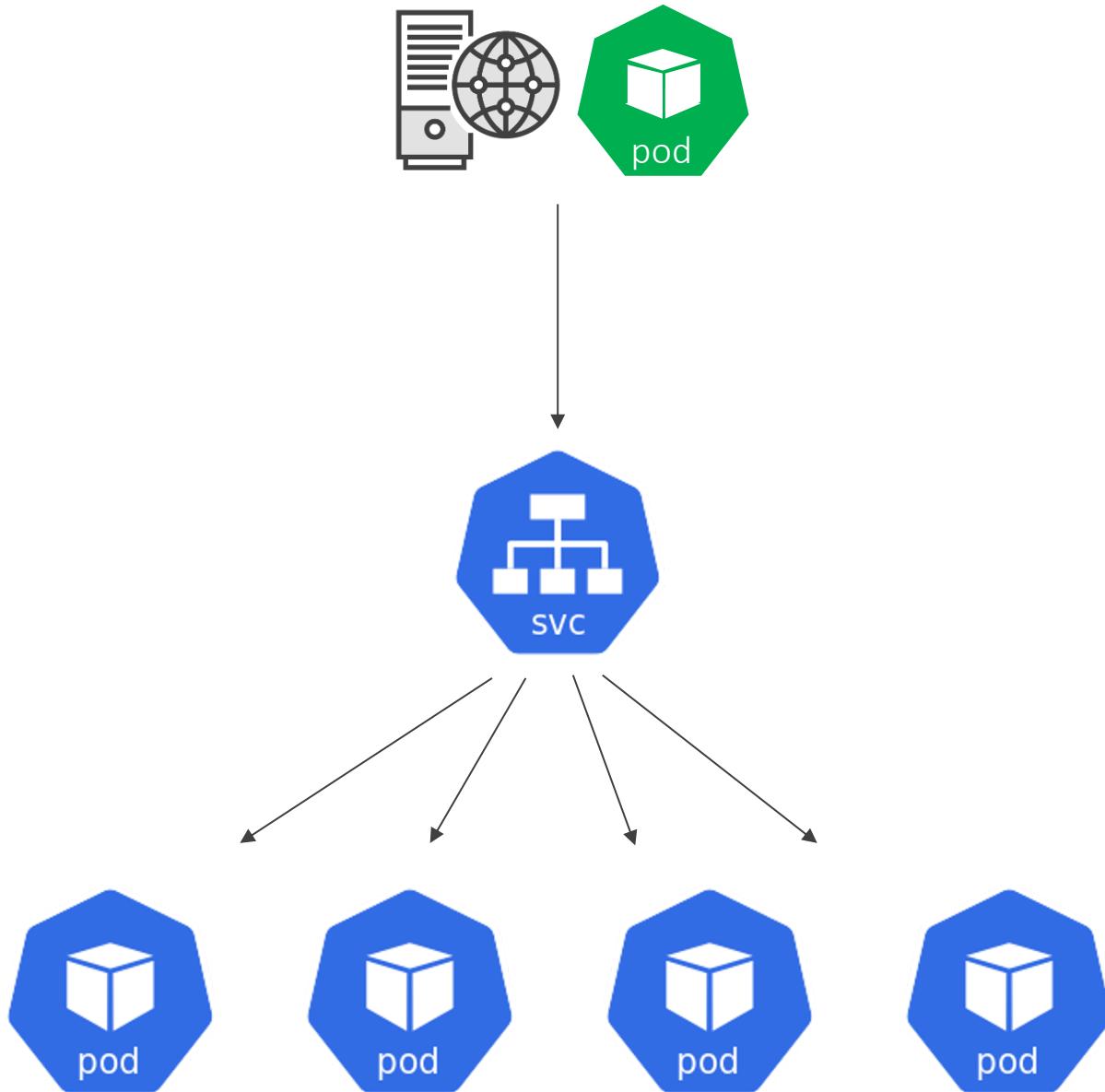


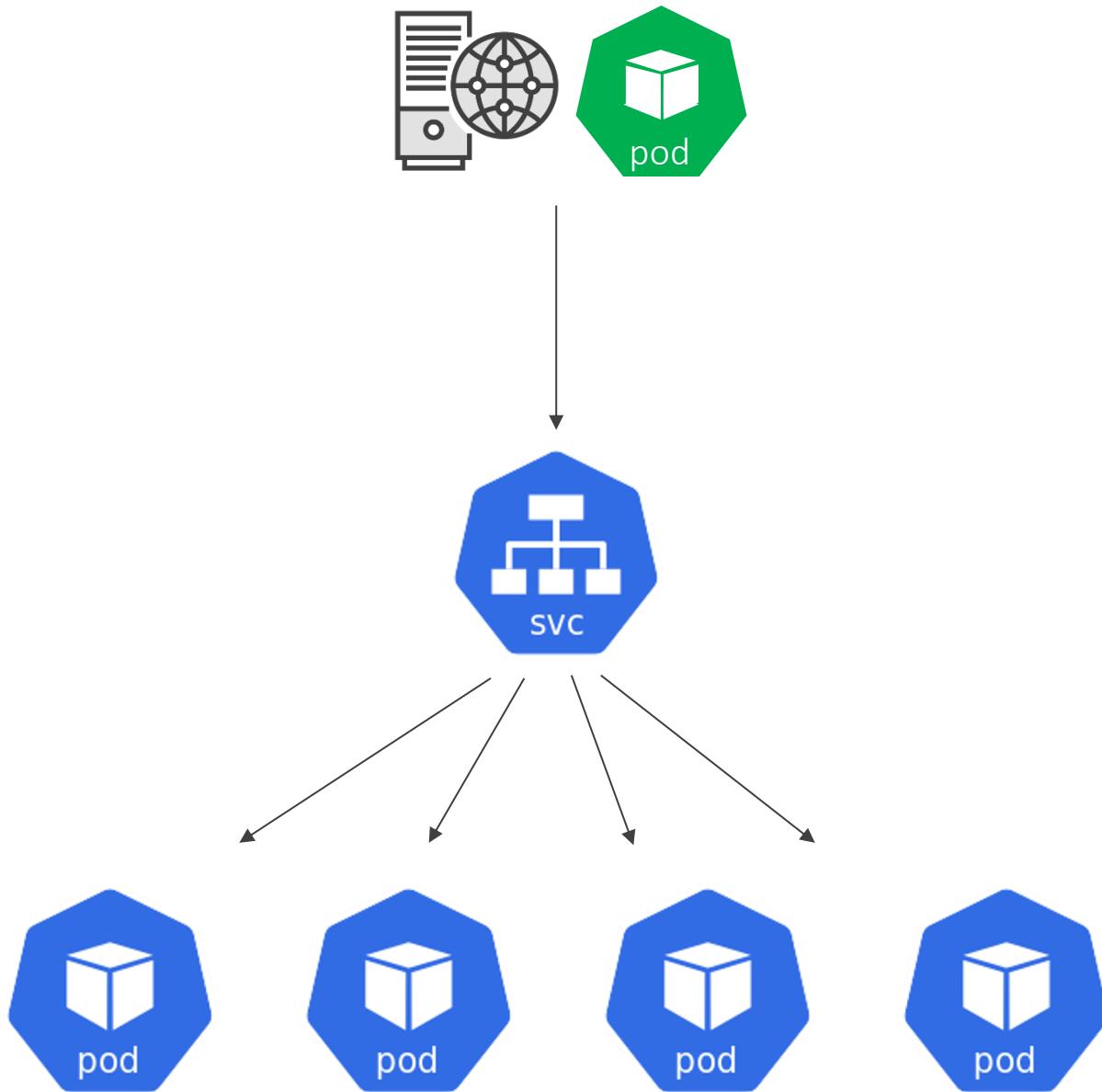


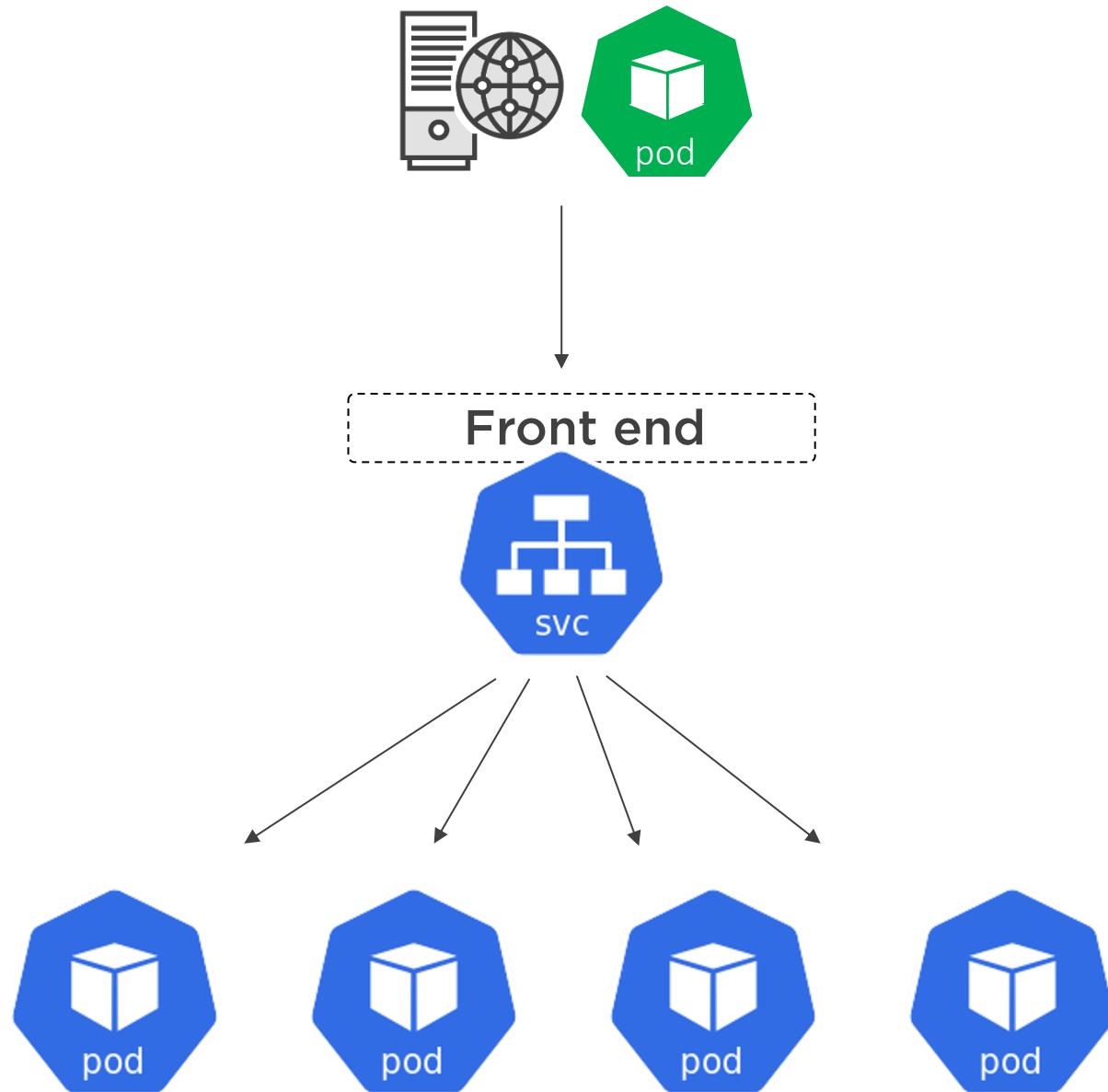


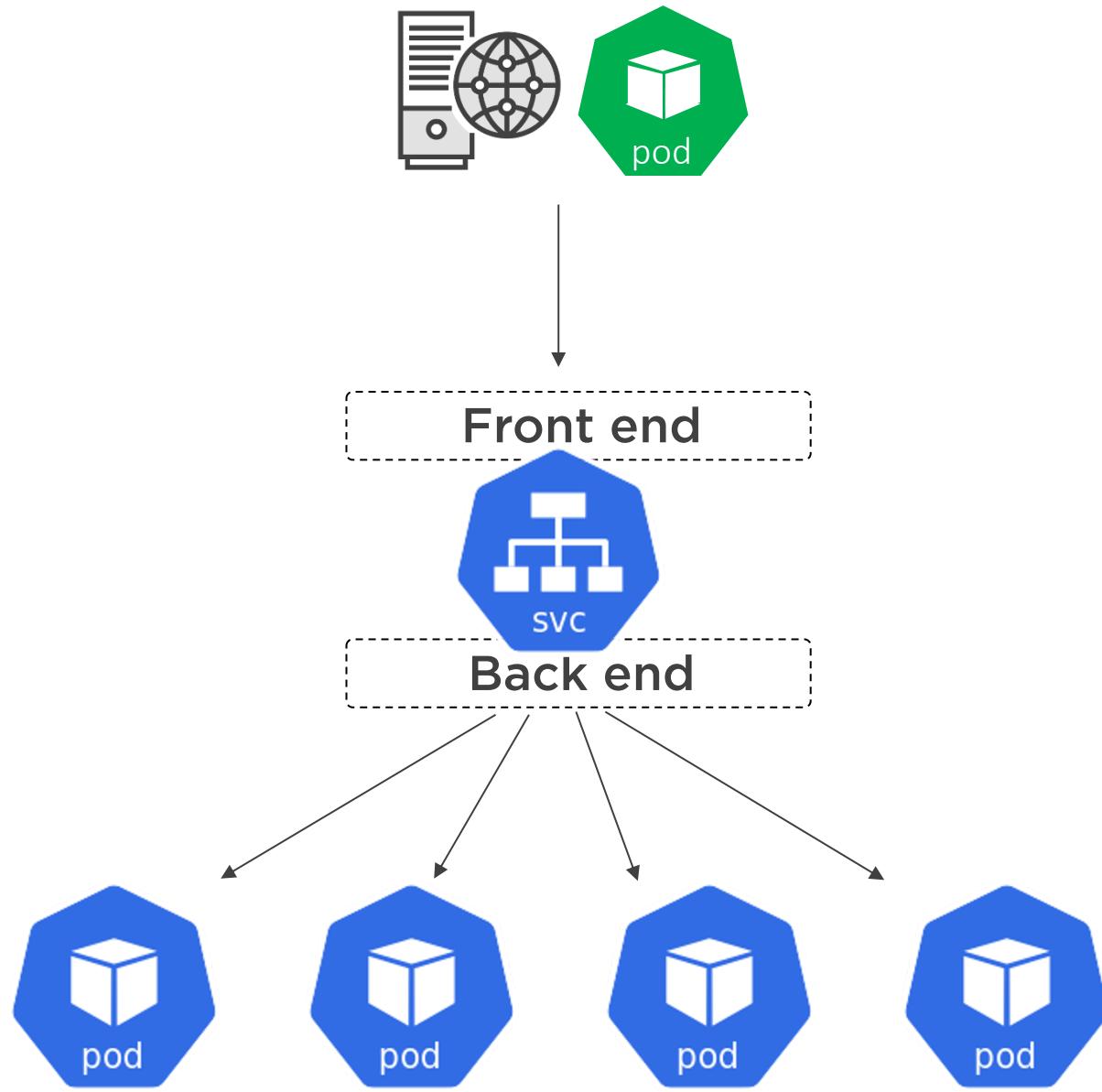


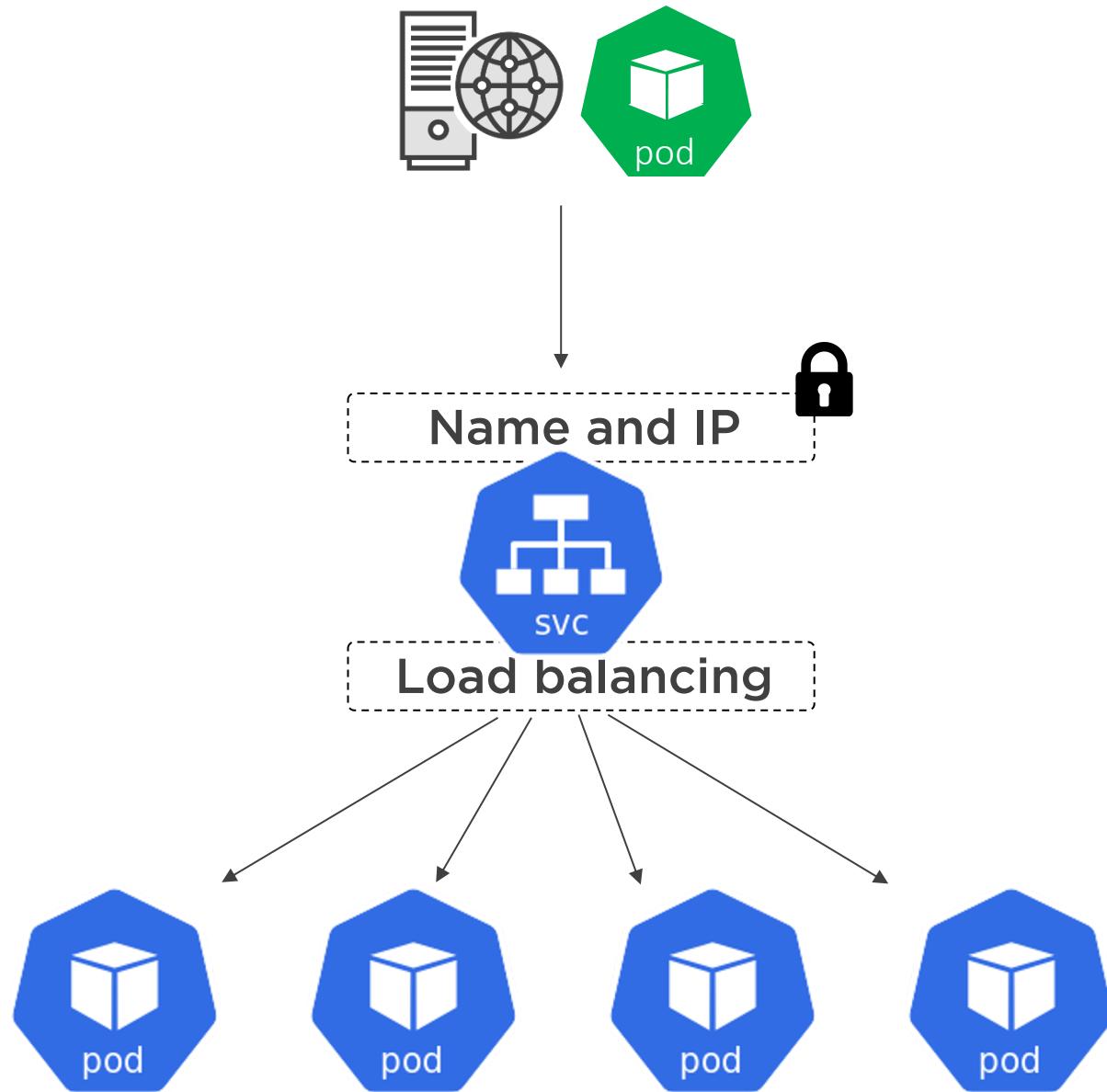


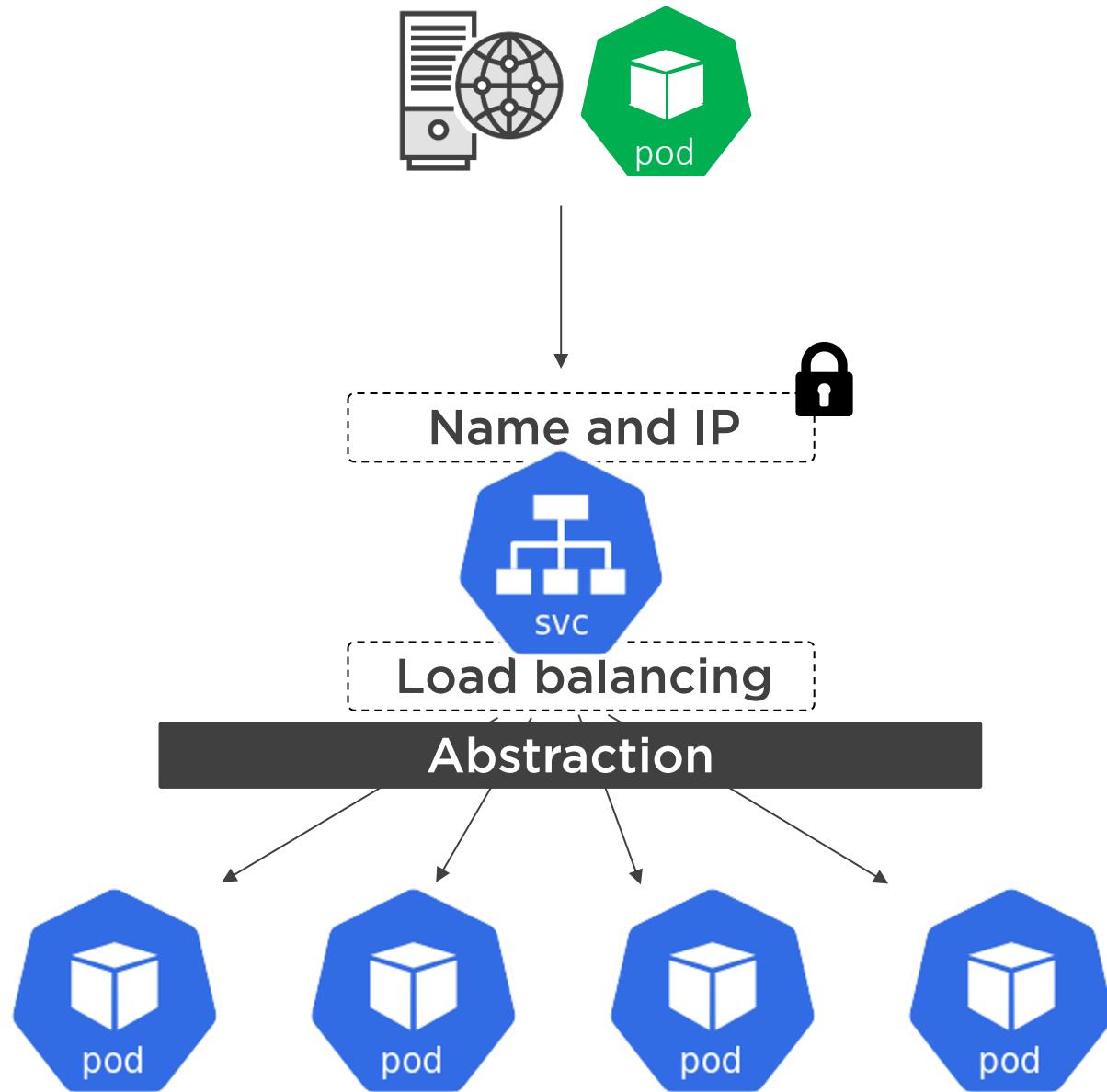


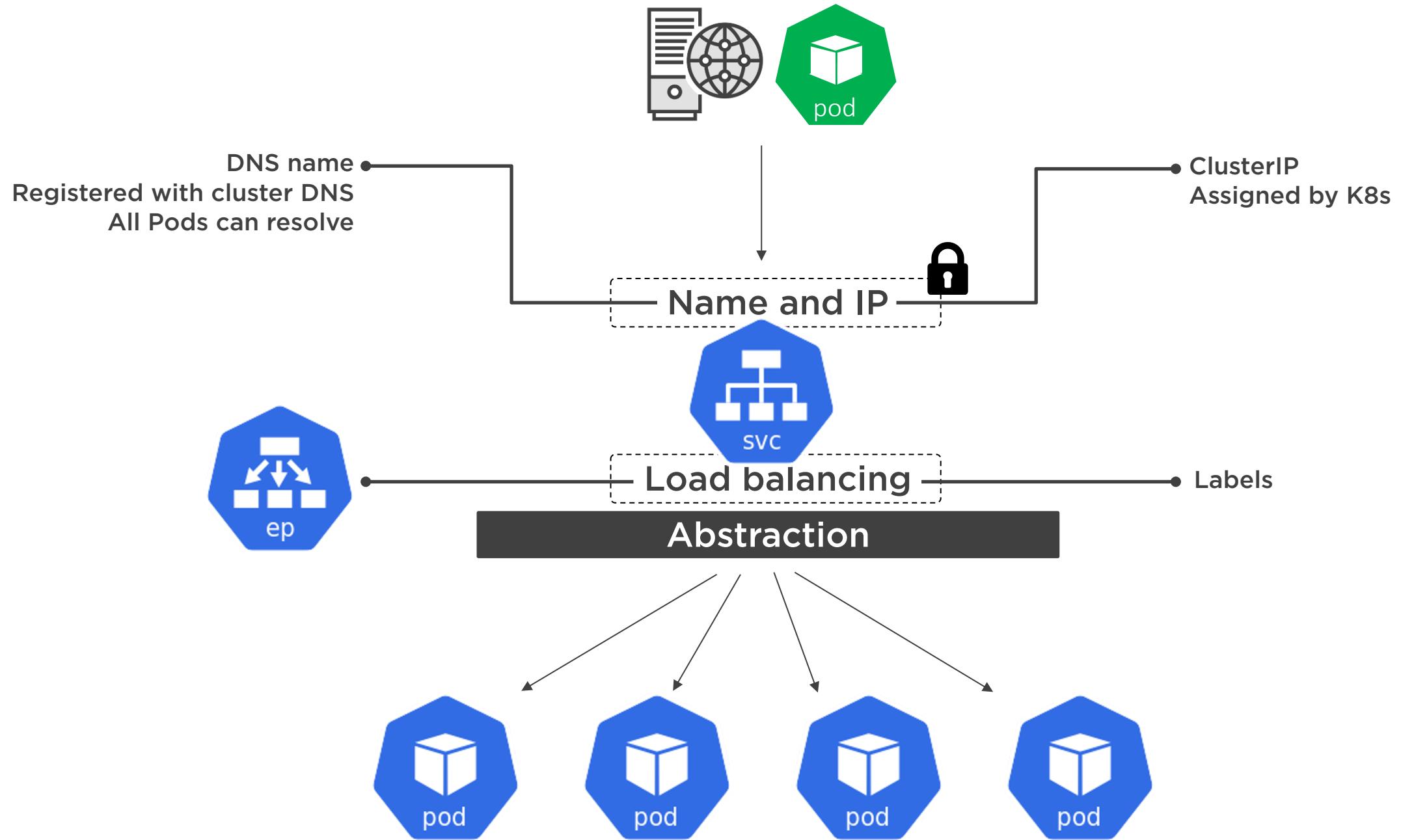


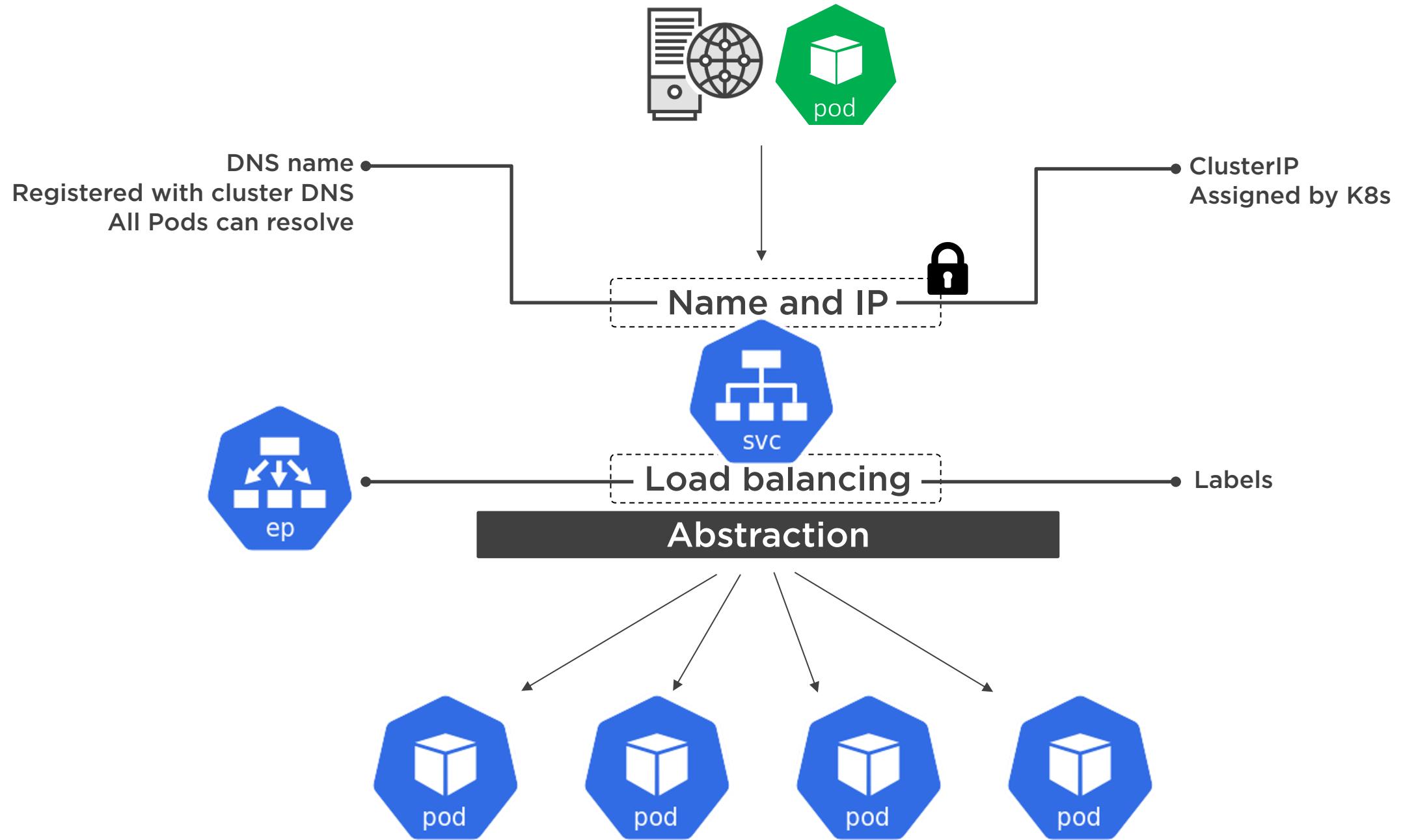


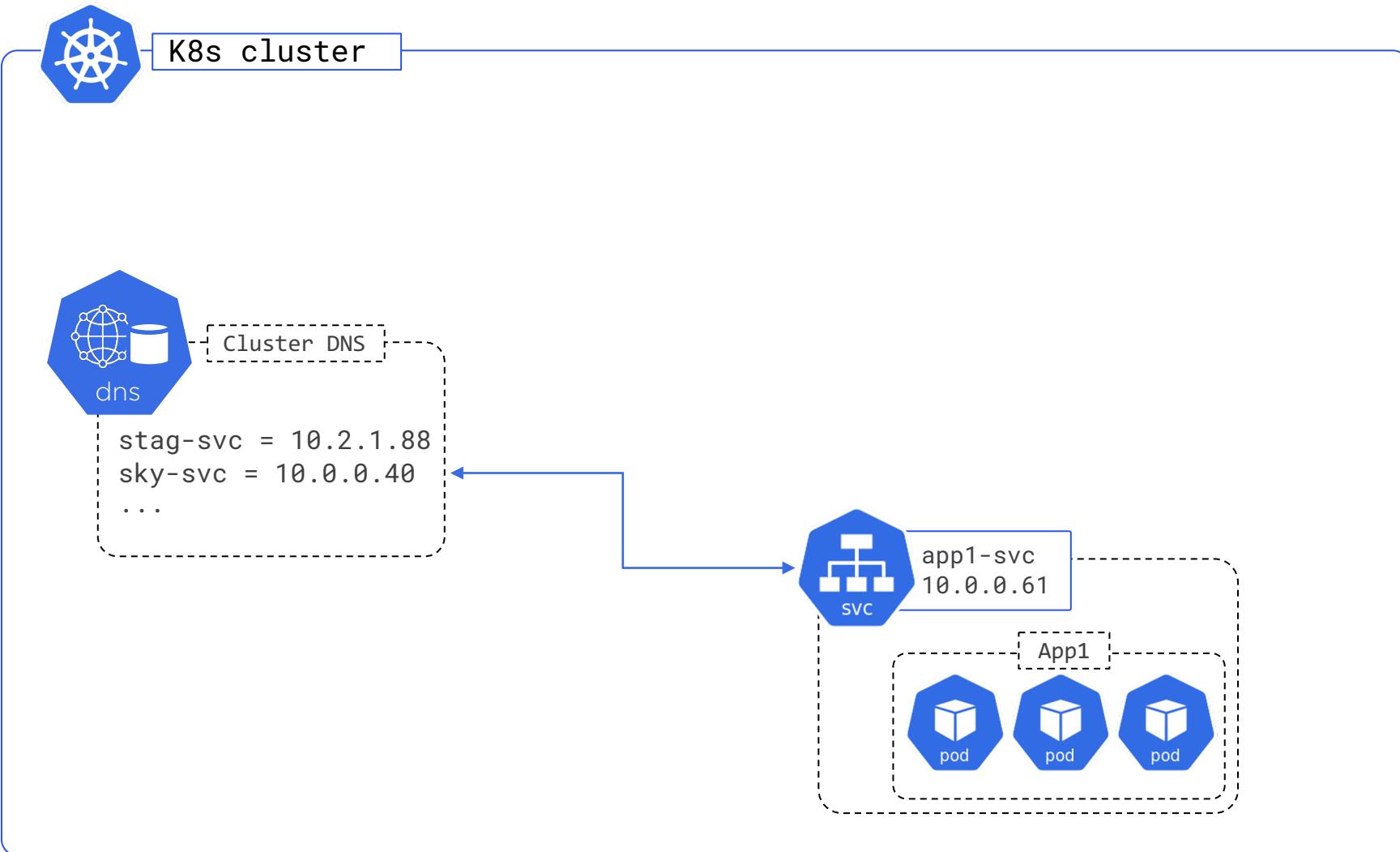


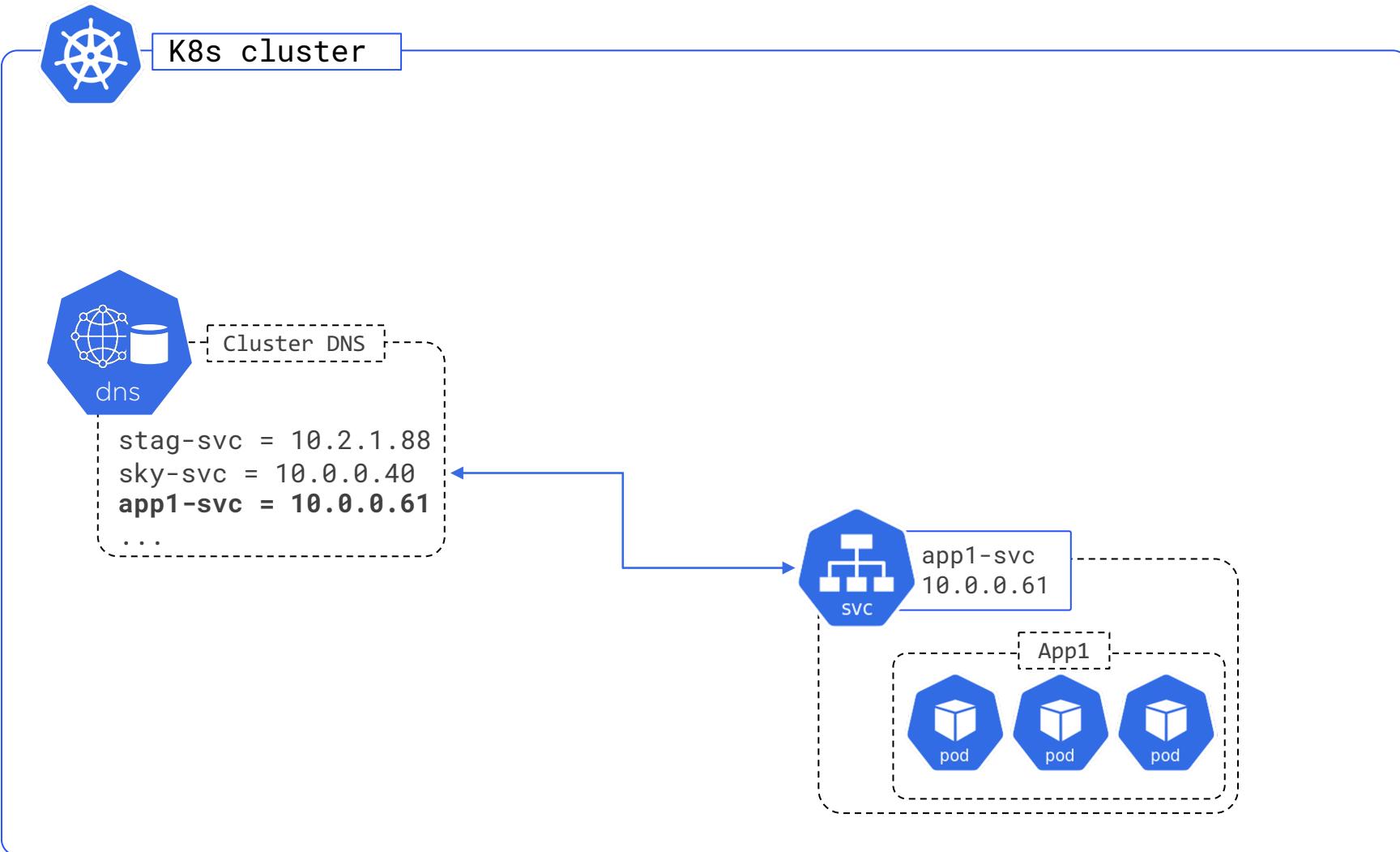


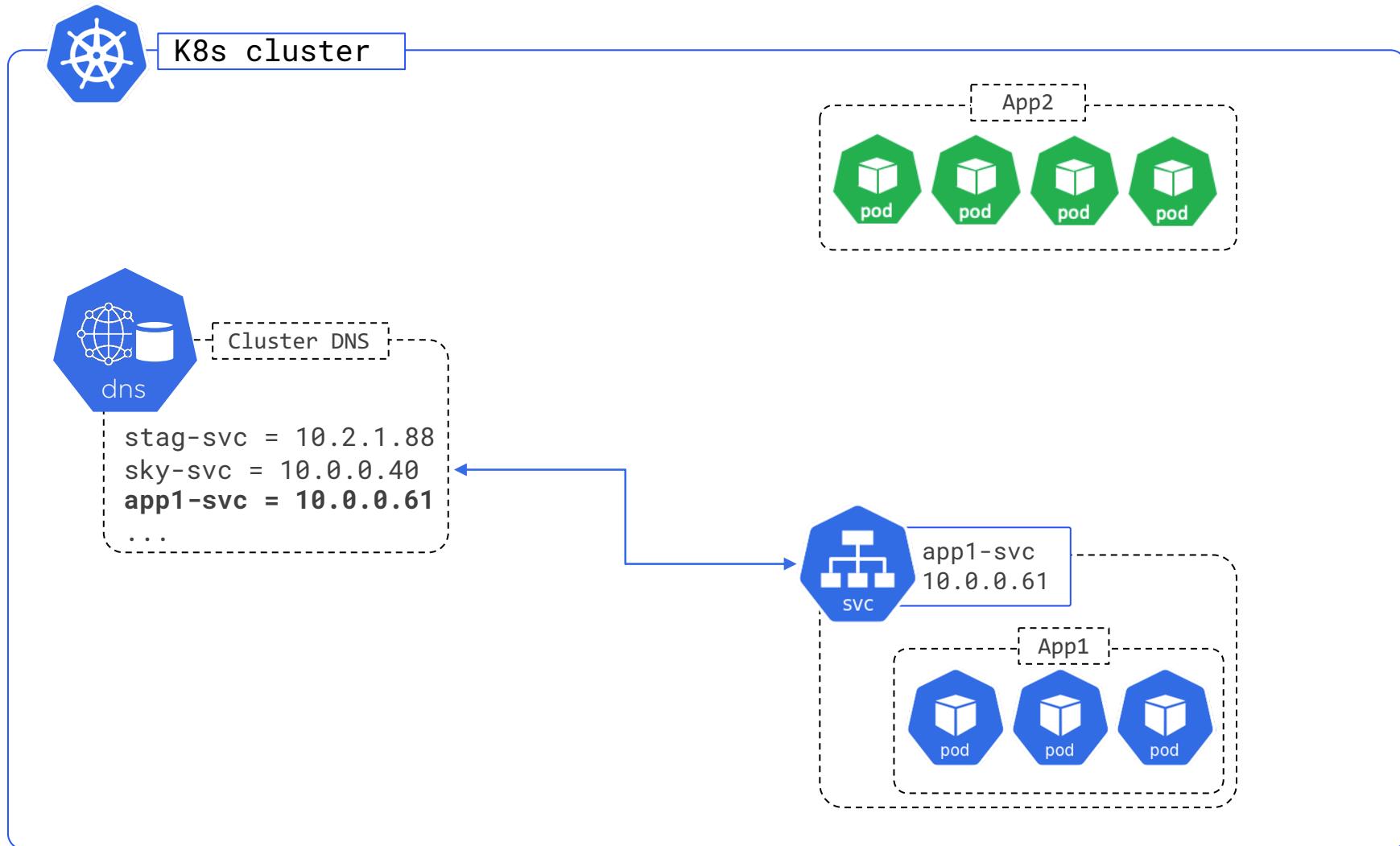


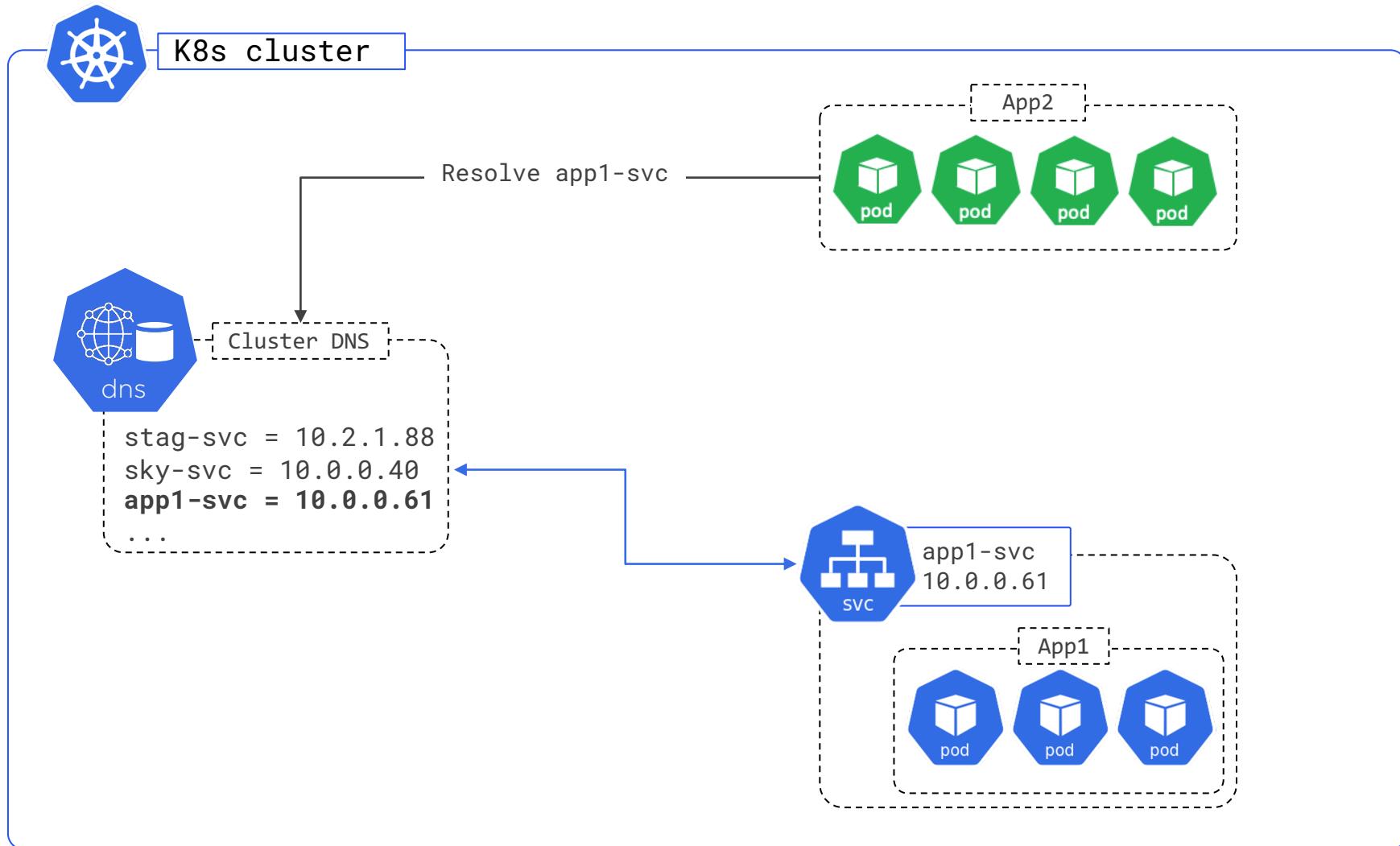


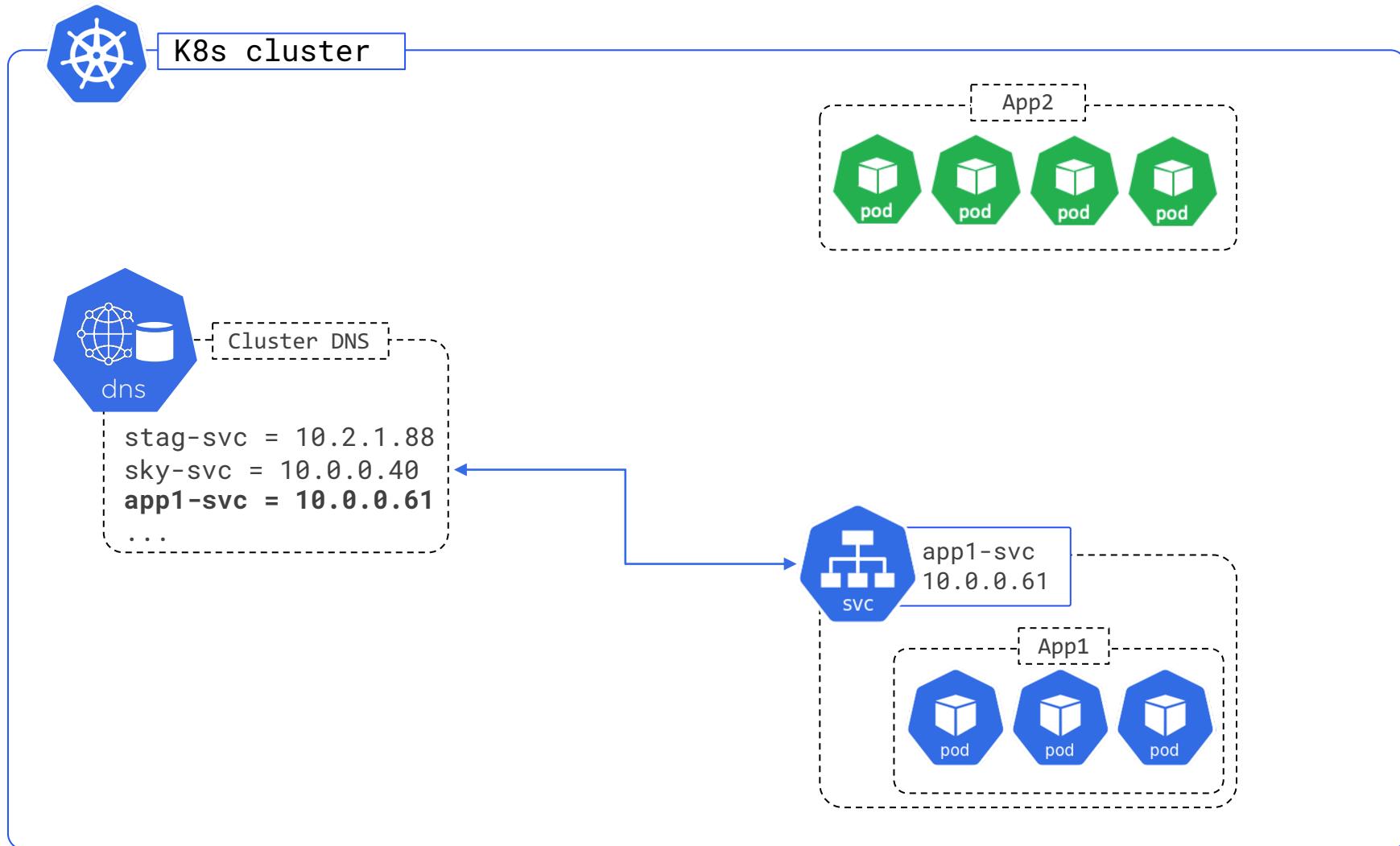


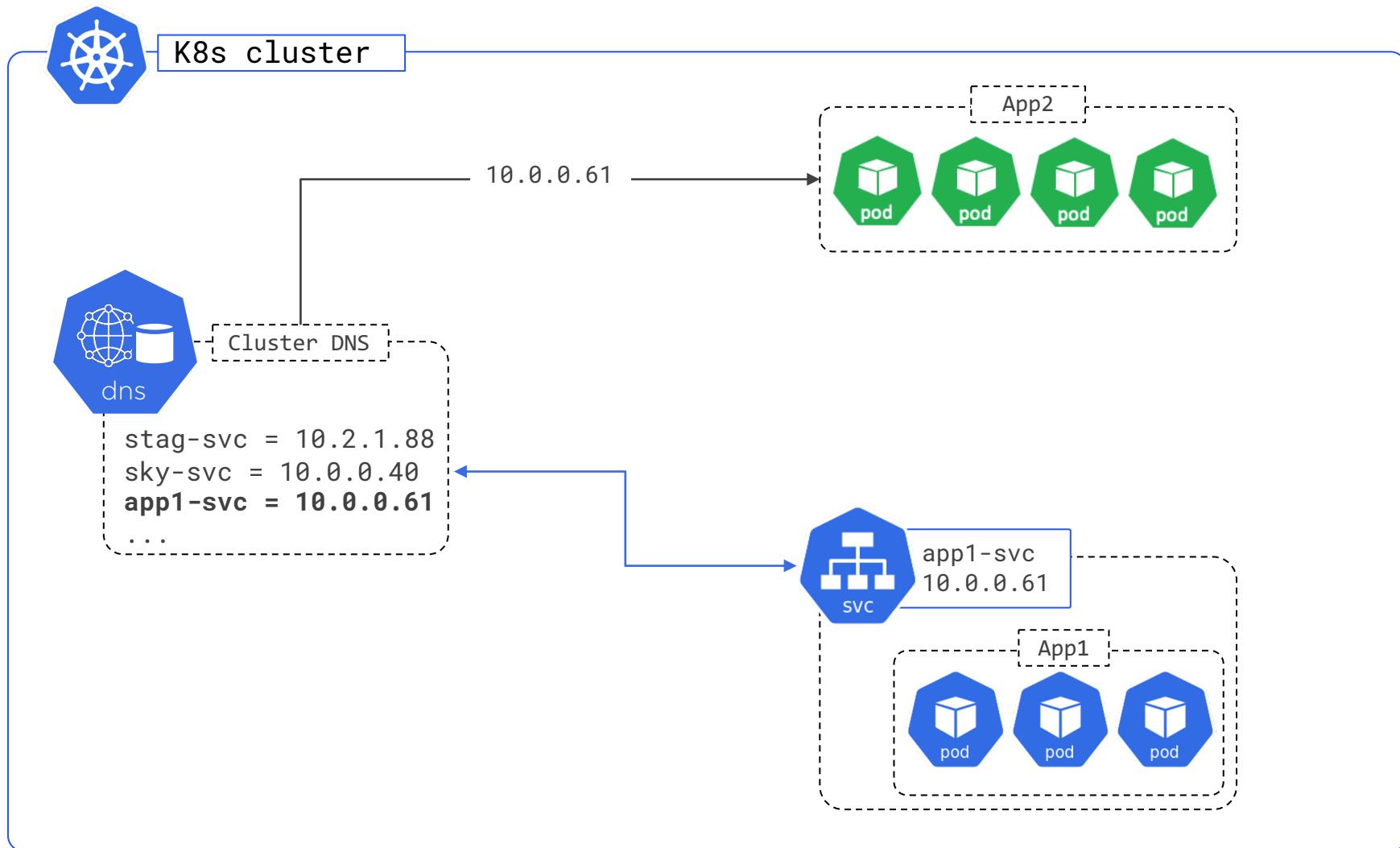


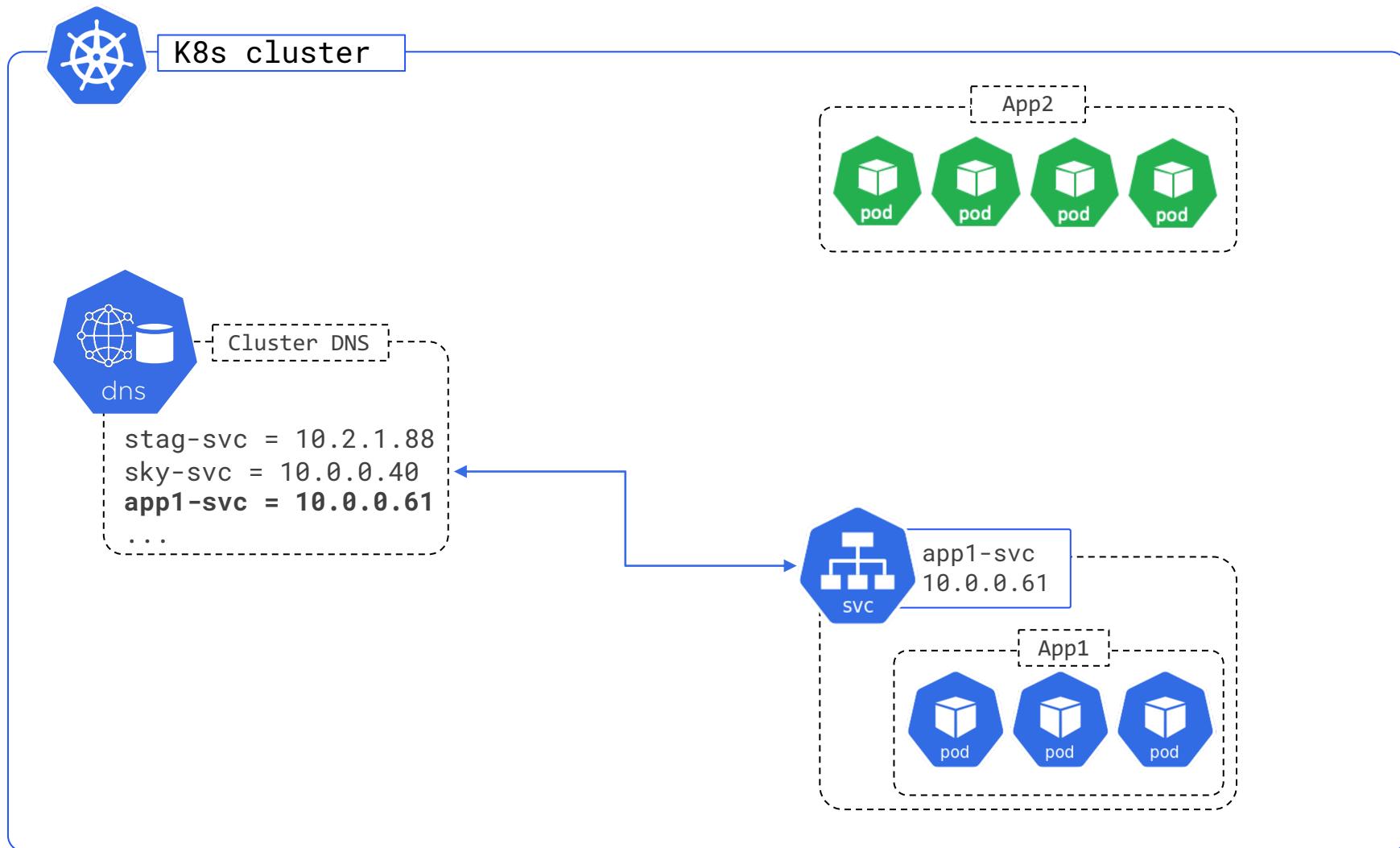


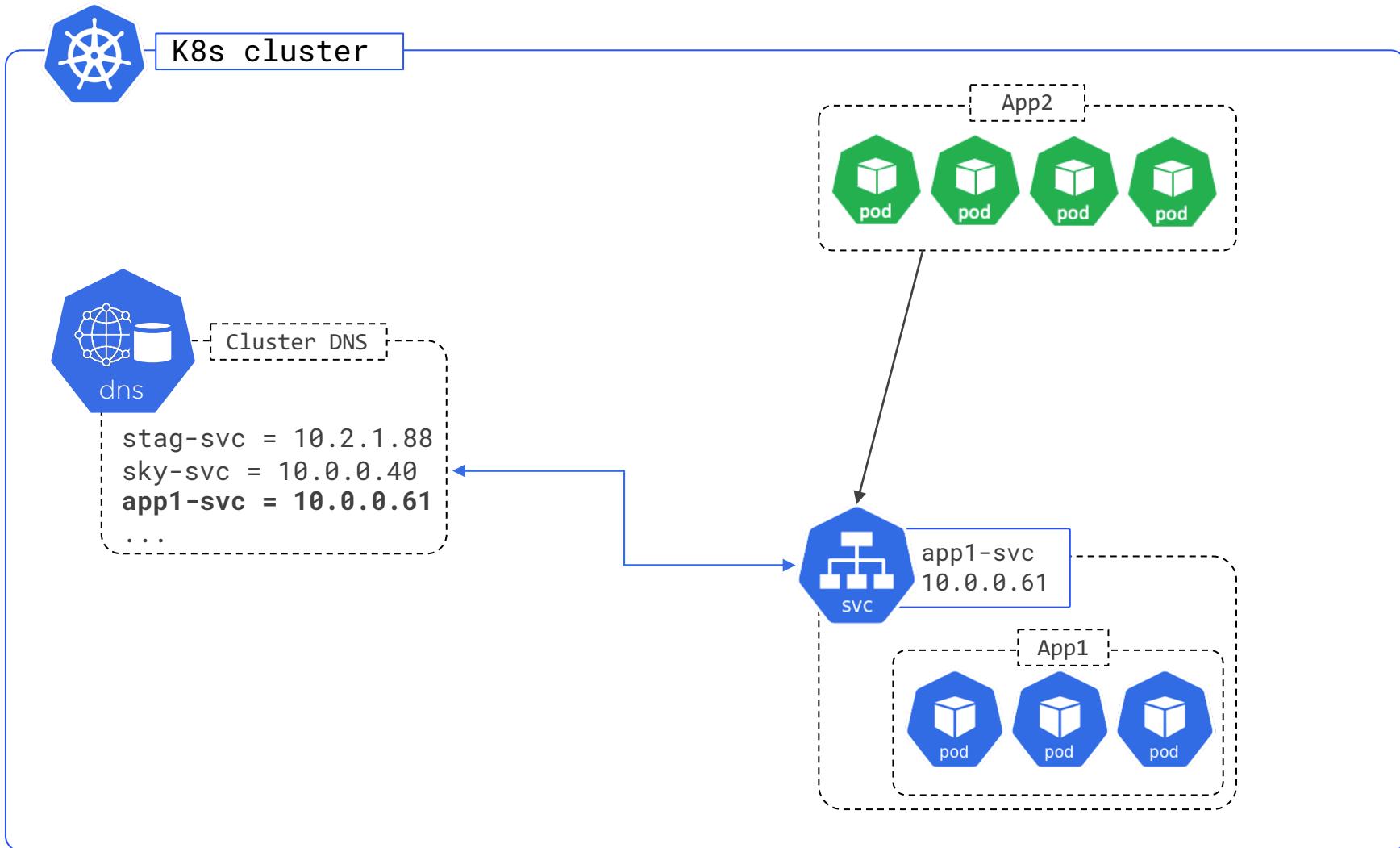


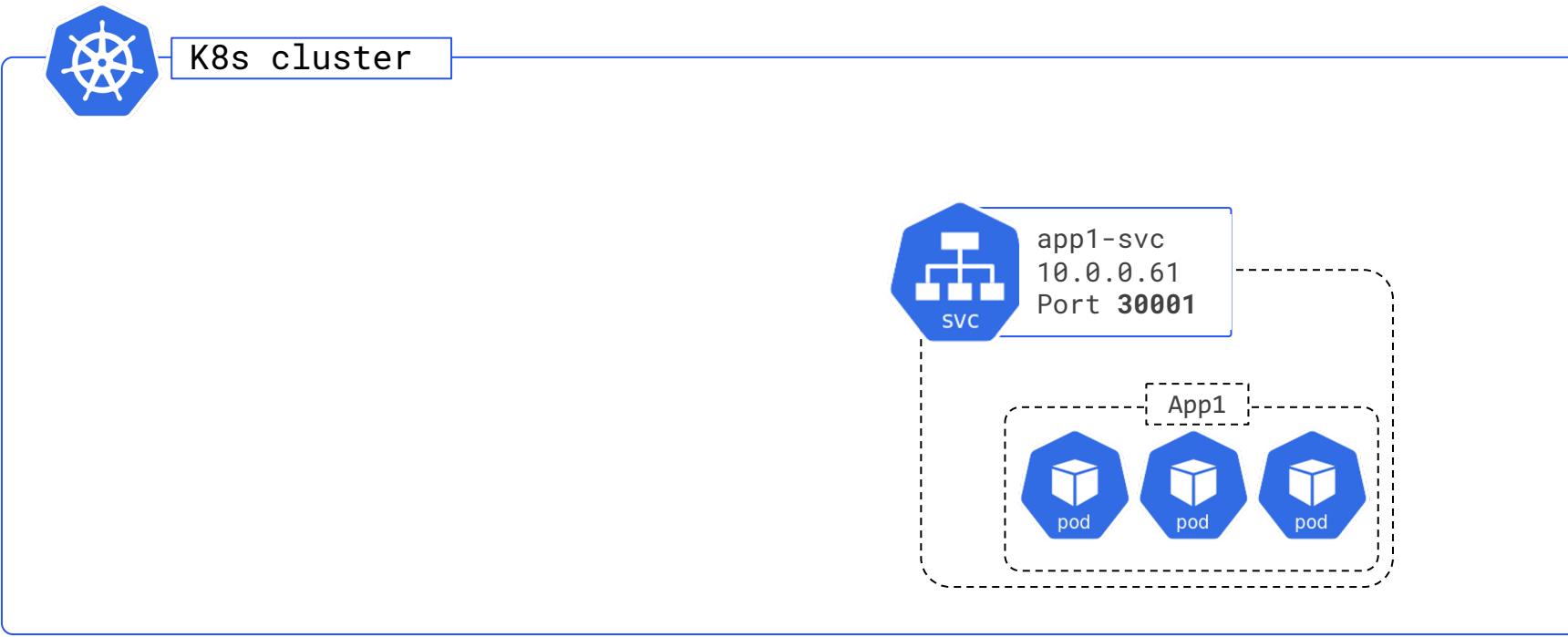




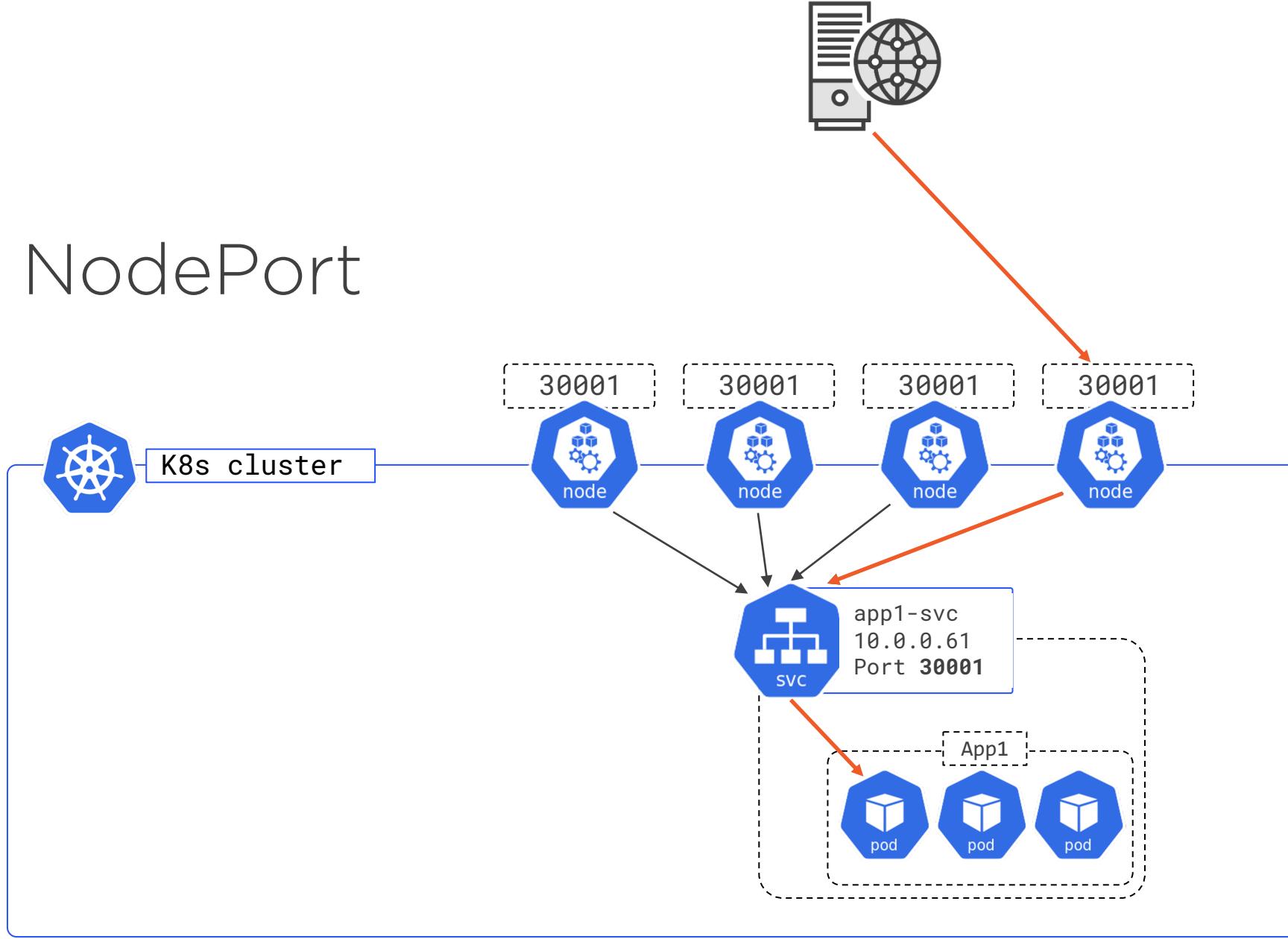






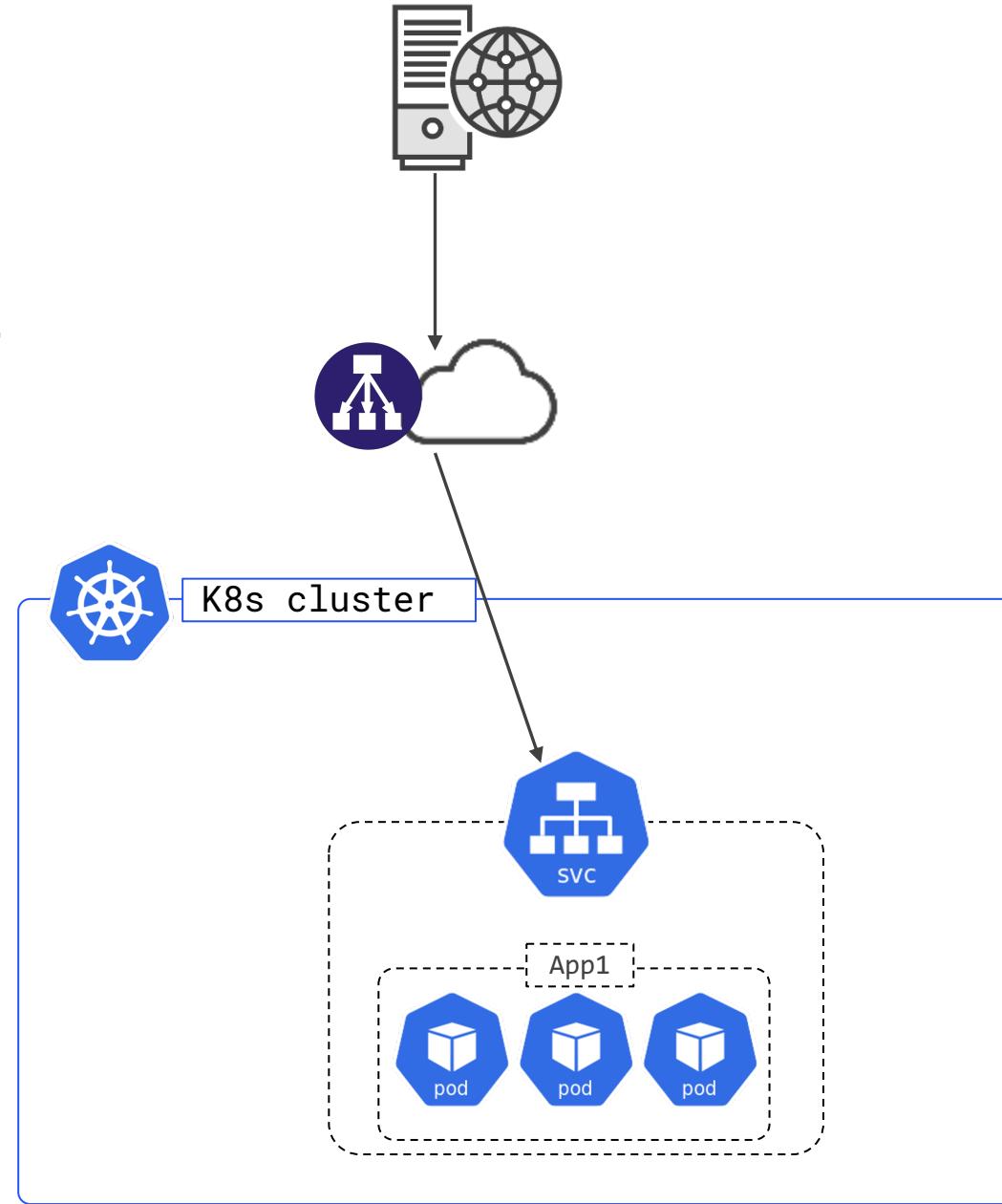


NodePort



LoadBalancer

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

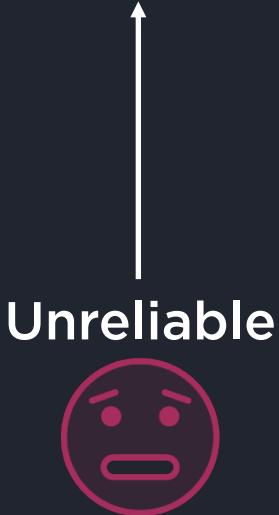


Up Next:
Creating a Service (the imperative way)



Creating a Service (the imperative way)





External access



Internal access



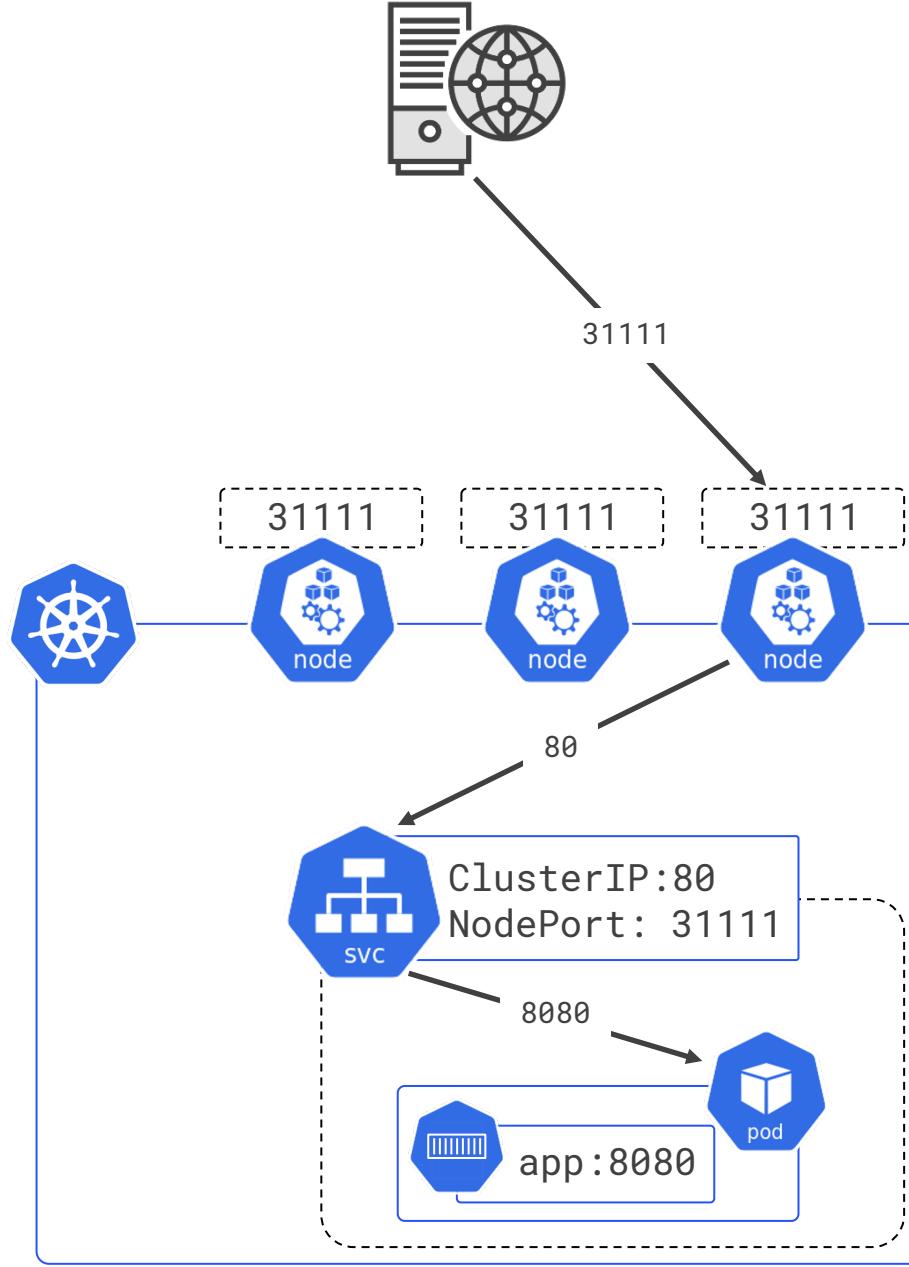
Up Next:
Creating a Service (the declarative way)



Creating a Service (the declarative way)



NodePort



LoadBalancer

External access via cloud load-balancer

NodePort

External access via nodes

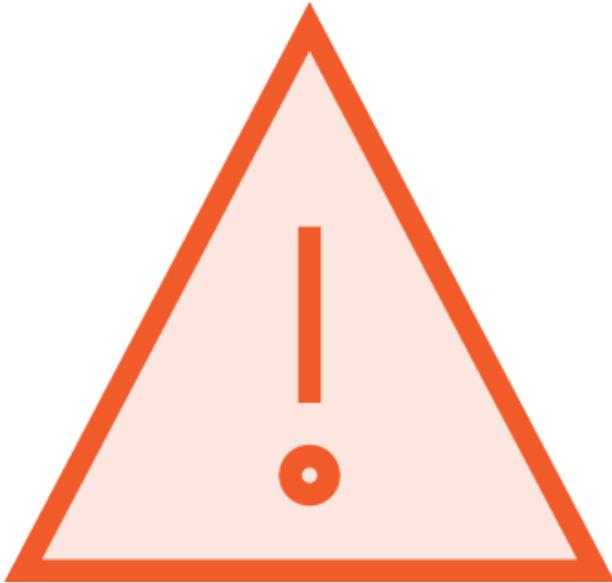
ClusterIP (default)

Internal cluster connectivity



Creating Internet LoadBalancer Services





**Only works on supported platforms
and supported load balancers**



Up Next: Services Recap



Recap



Stable Networking

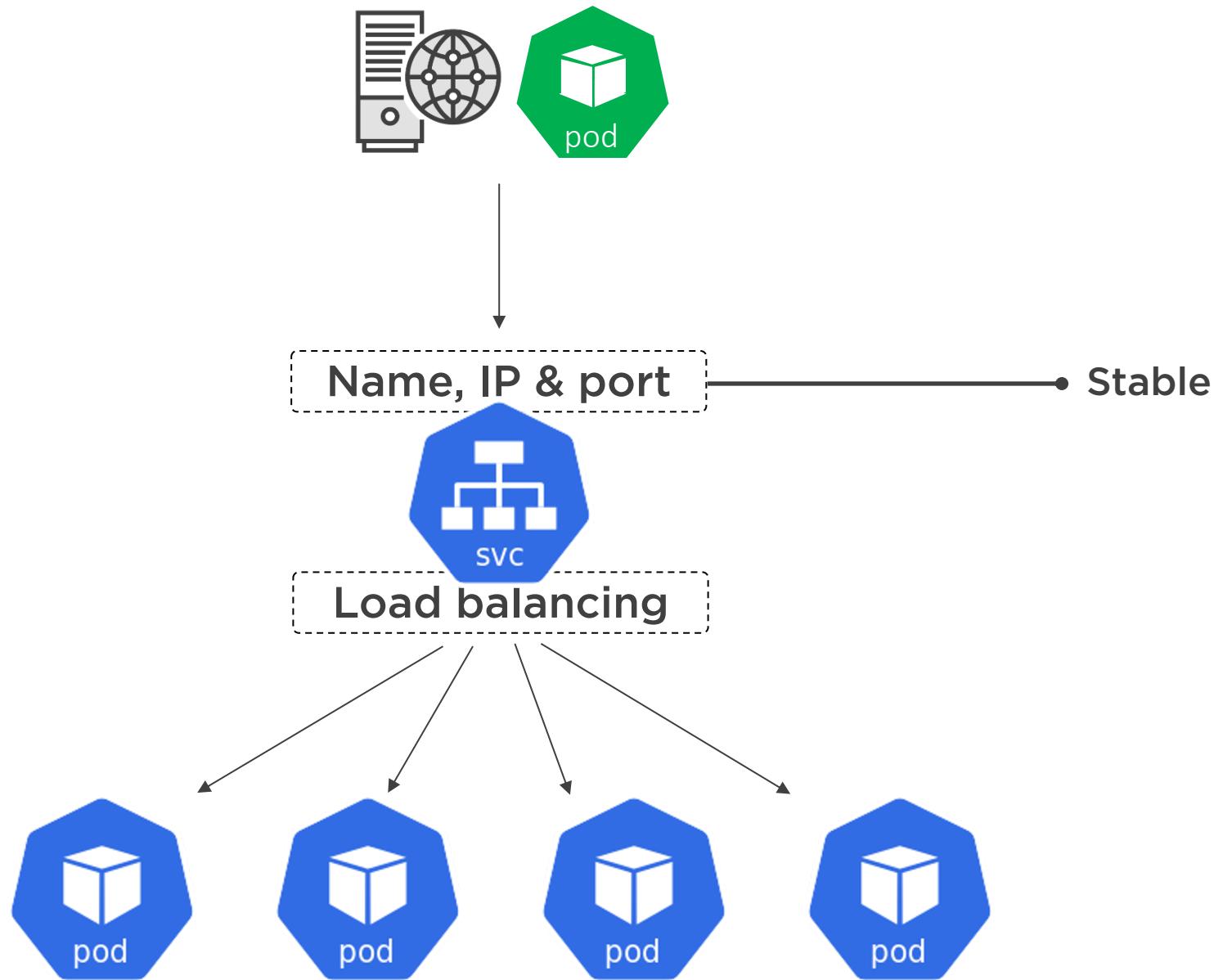
Reliable networking for a set
of Pods

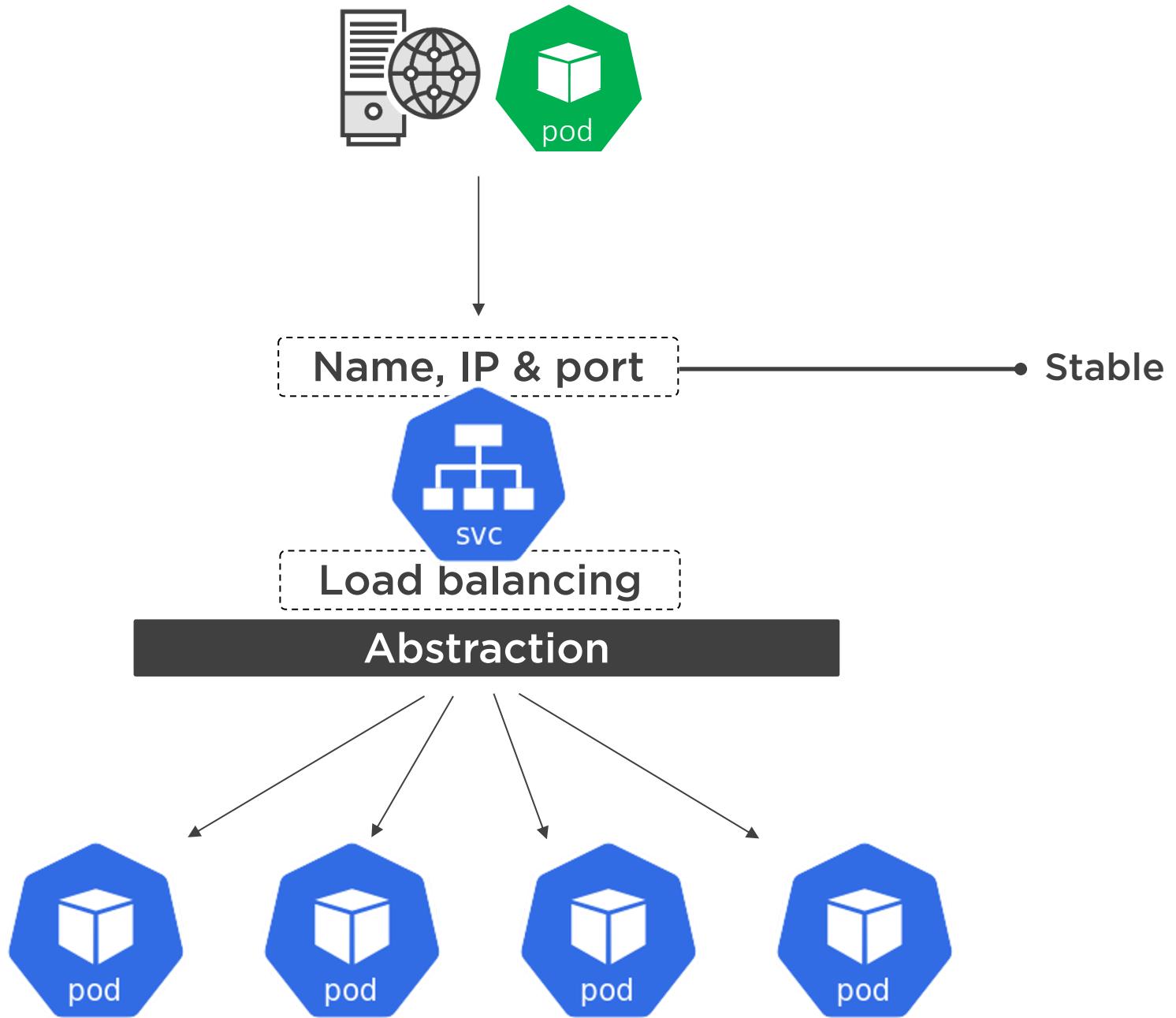


Abstraction

Hides the unreliable nature of
Pods

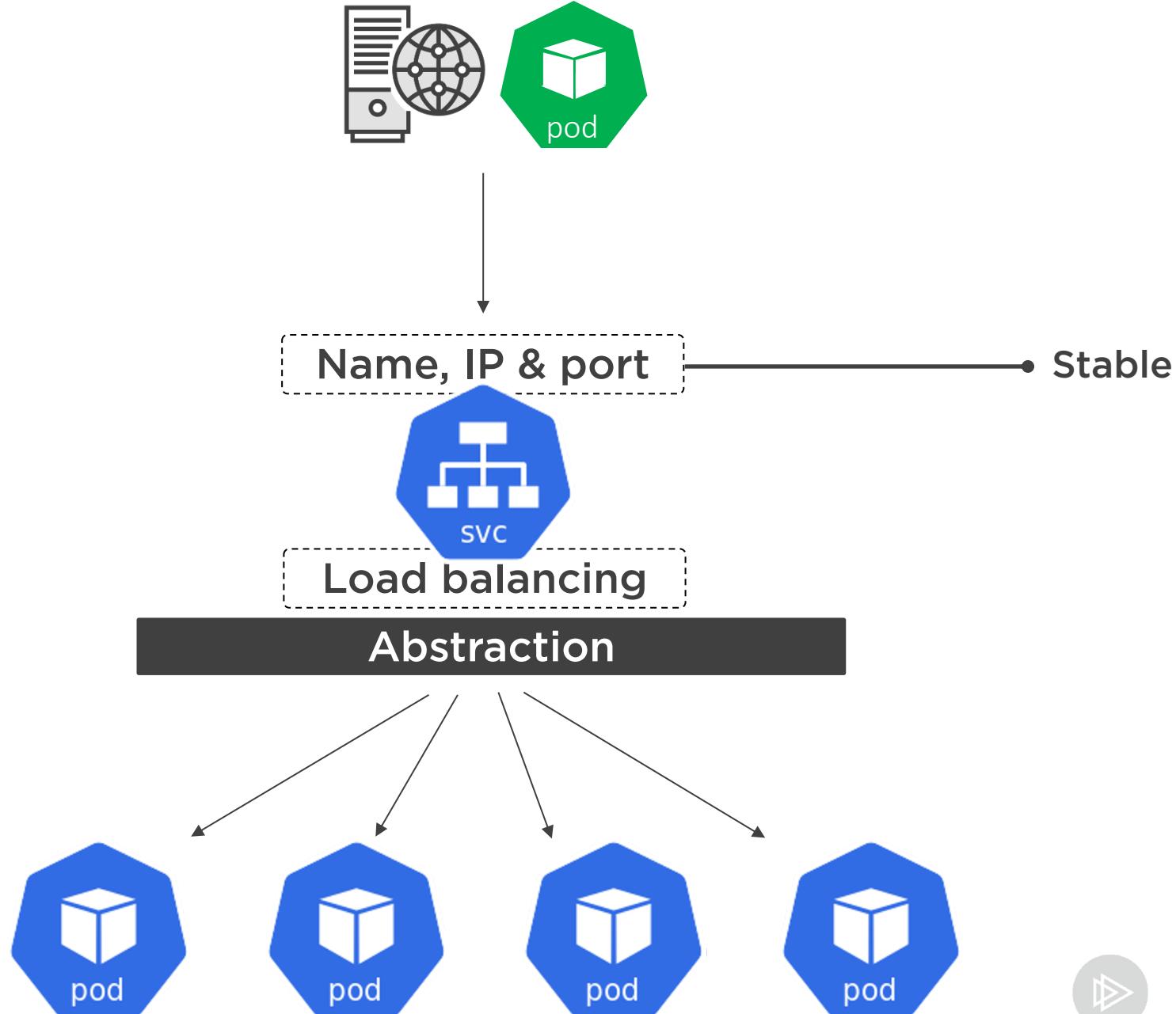






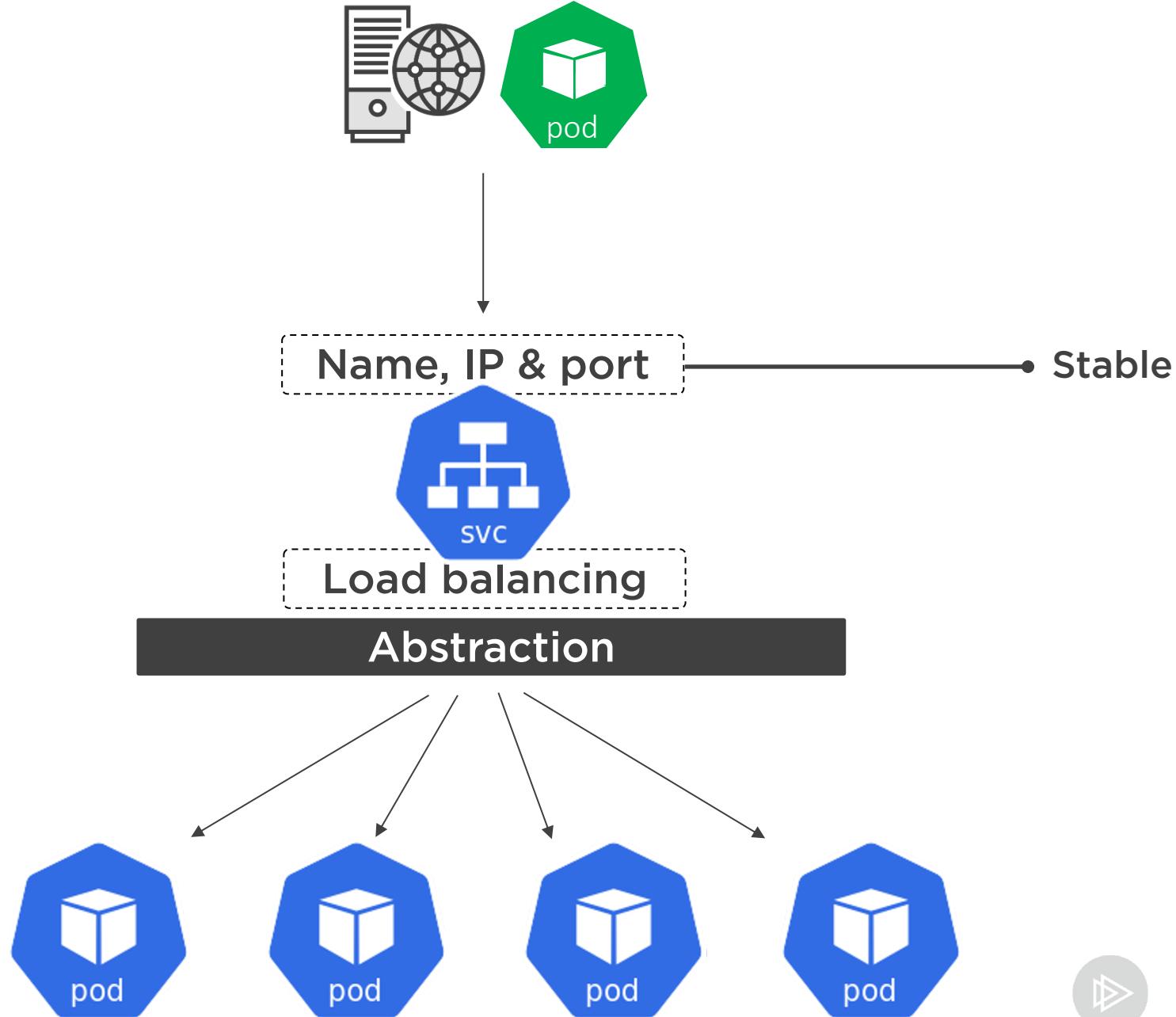
ClusterIP (default)

Internal cluster connectivity

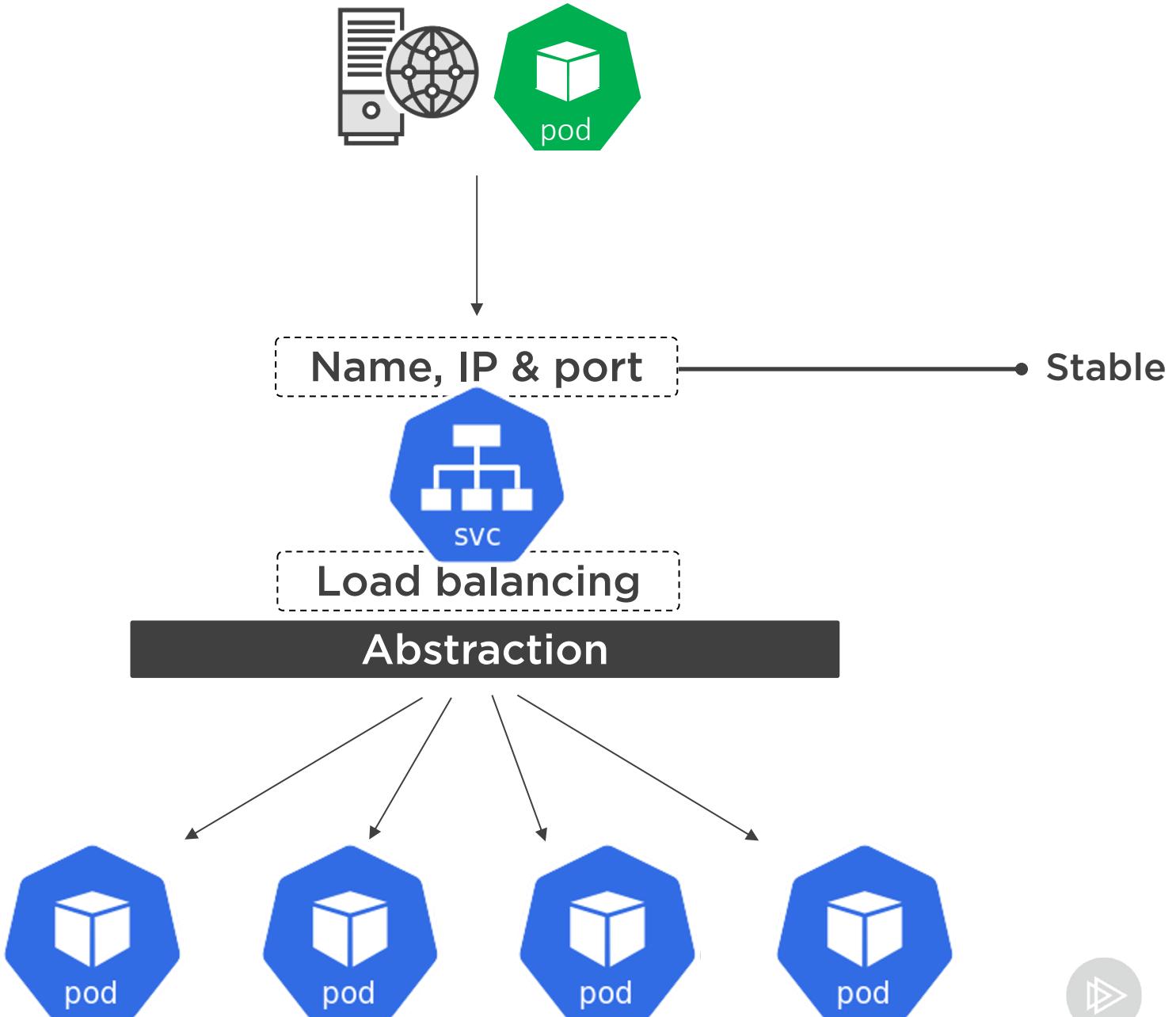


NodePort
External access via nodes

ClusterIP (default)
Internal cluster connectivity



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



Up Next:
Kubernetes Deployments



Kubernetes Deployments



Nigel Poulton

@nigelpoulton www.nigelpoulton.com



Overview



Theory

Creating a Deployment YAML

Deploying a Deployment

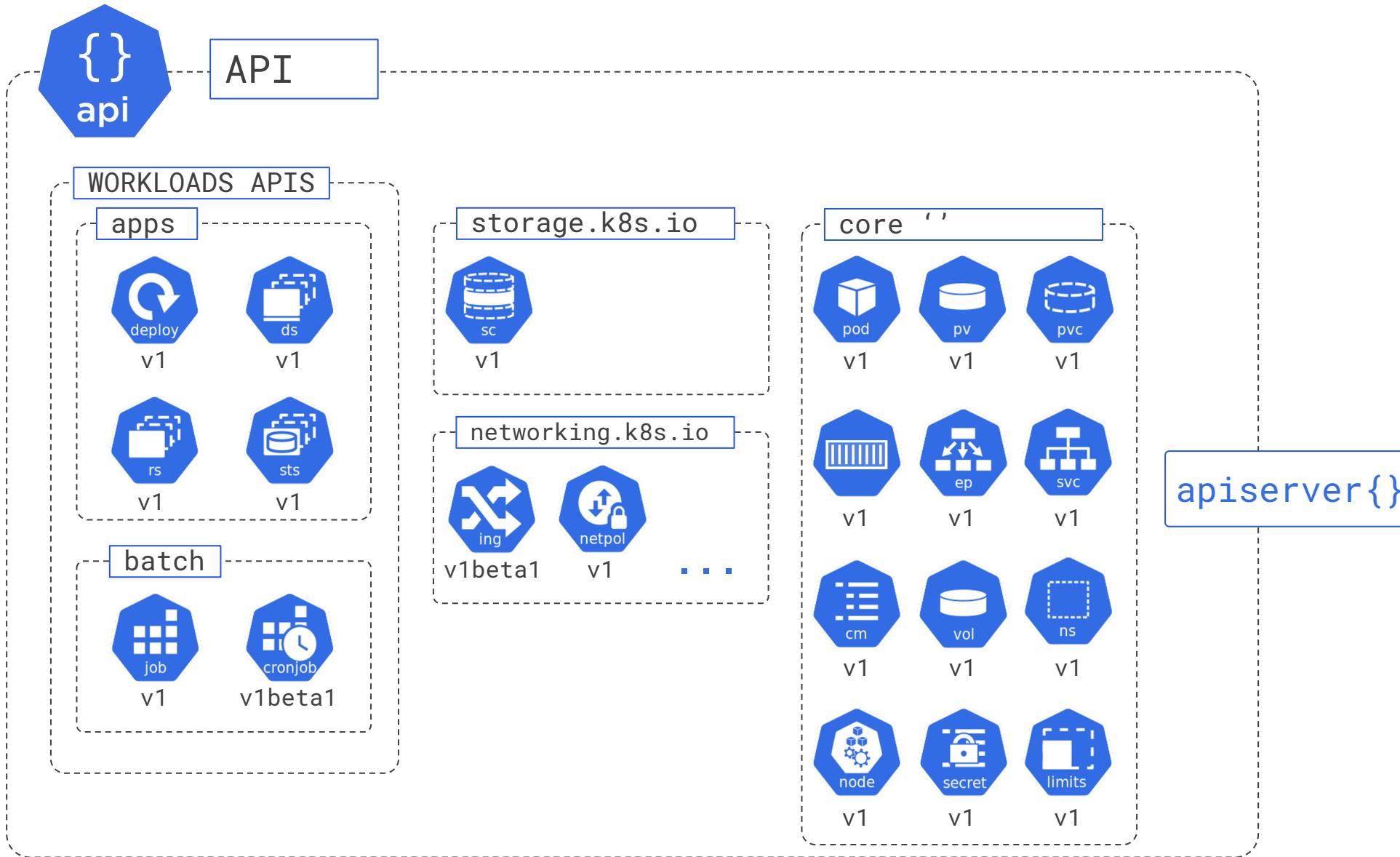
Self-healing and Scaling

Rolling Updates and Rollbacks



Kubernetes Deployment Theory



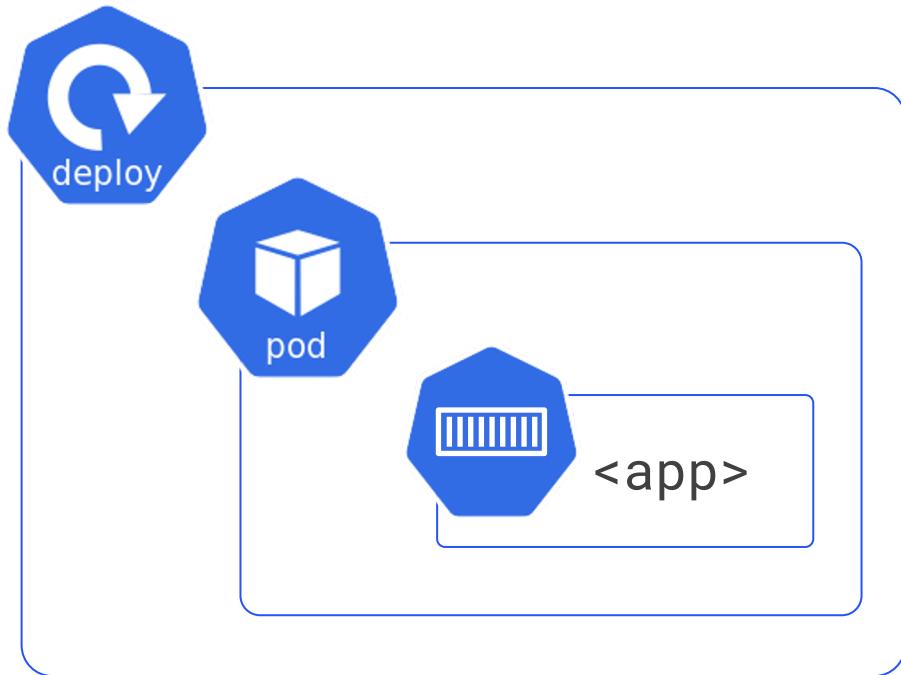


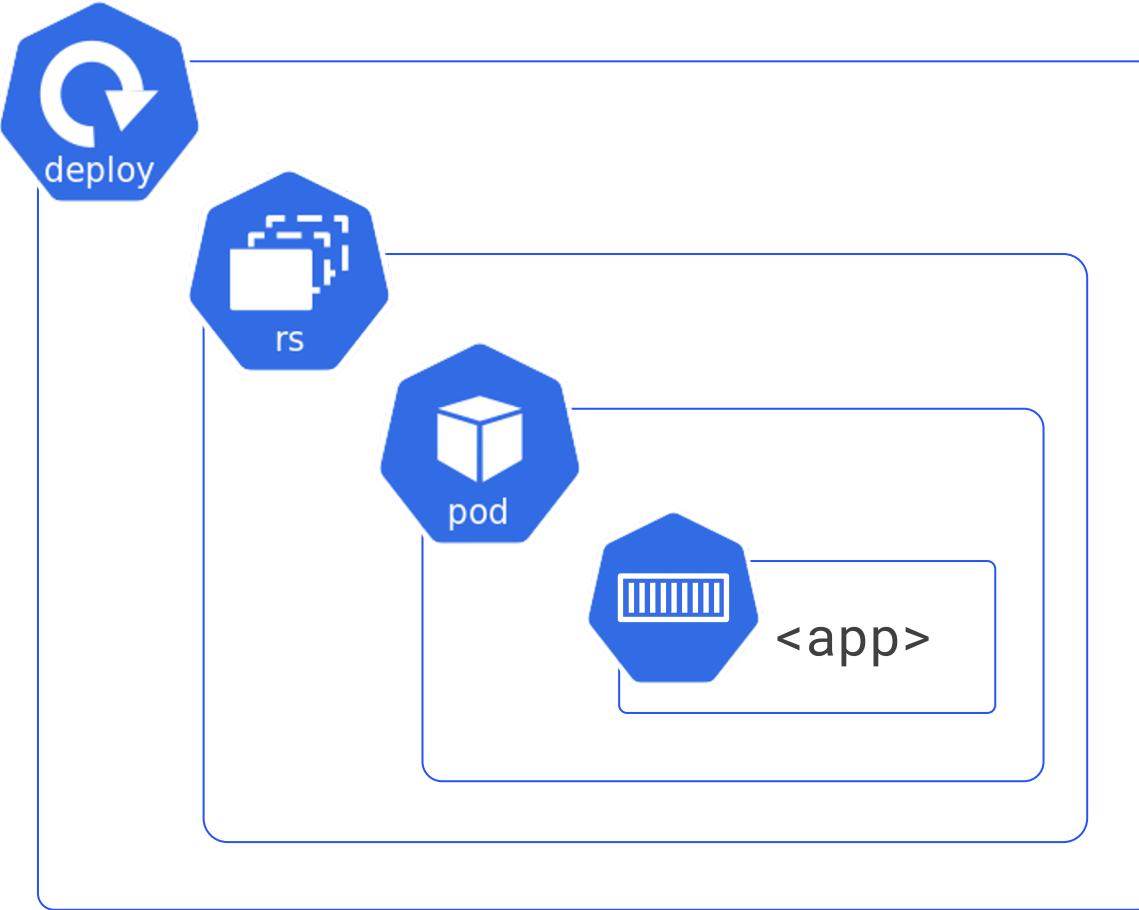


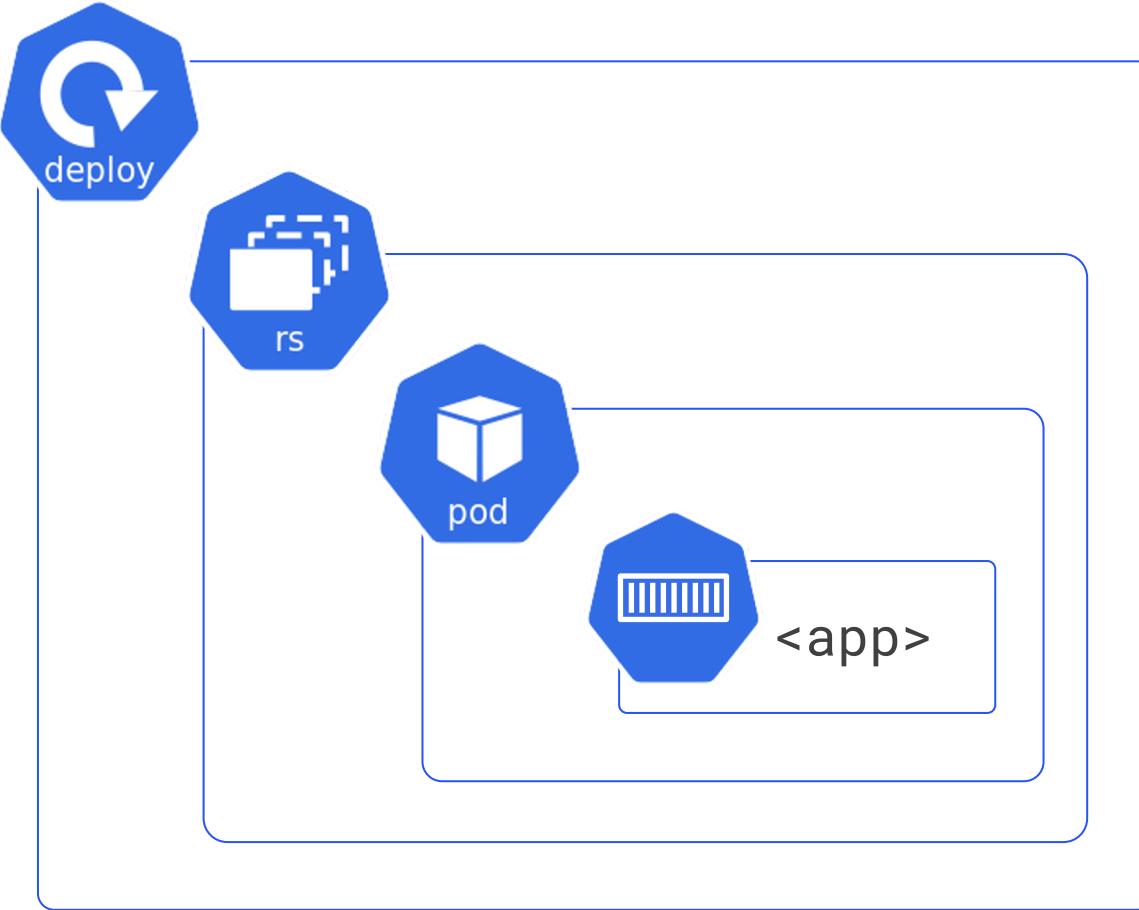


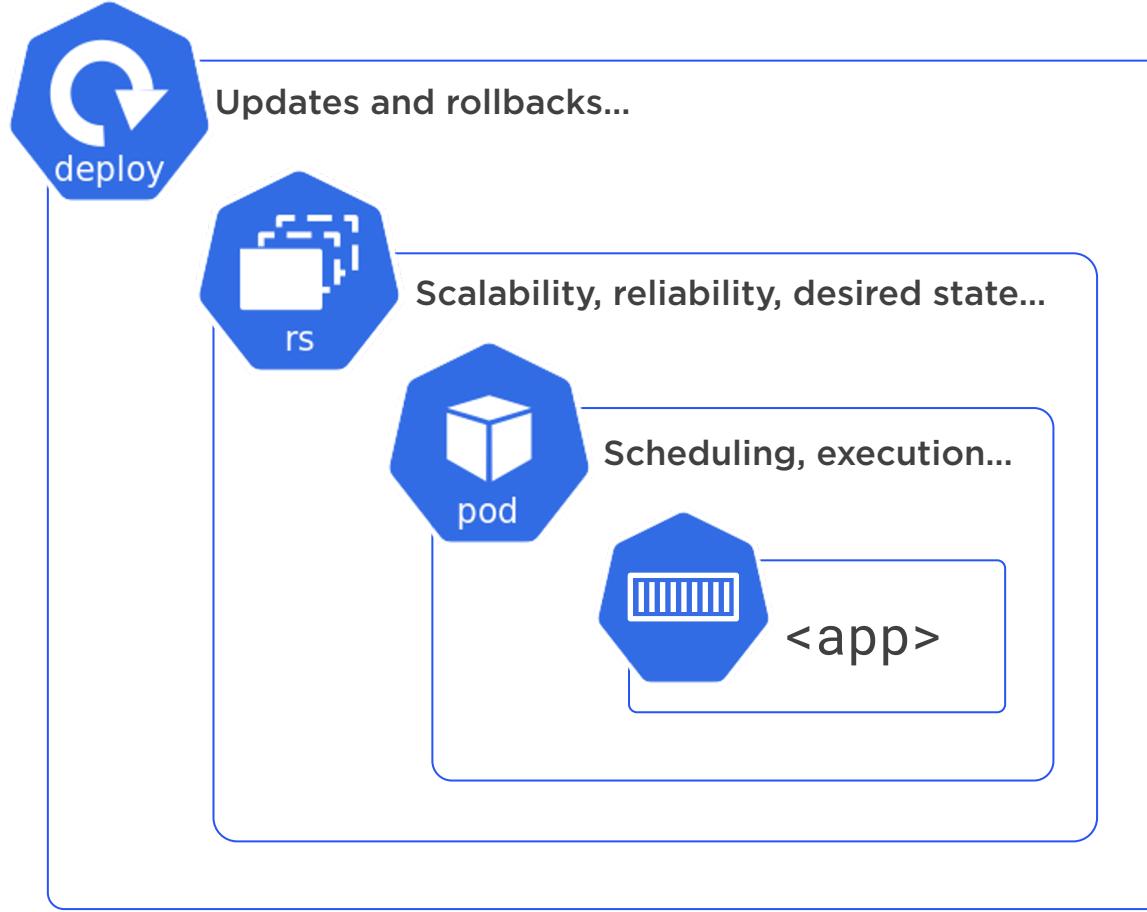
Self-healing, scaling, rolling updates, rollbacks

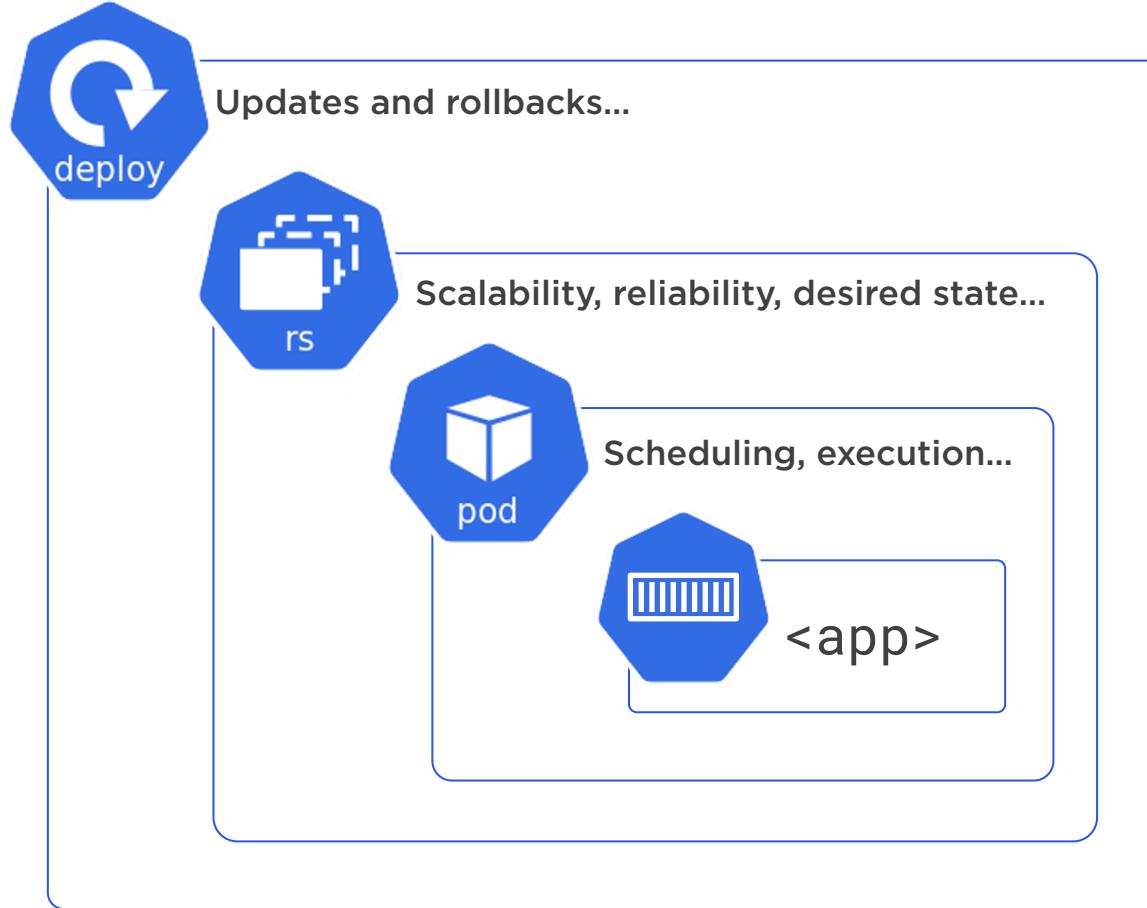


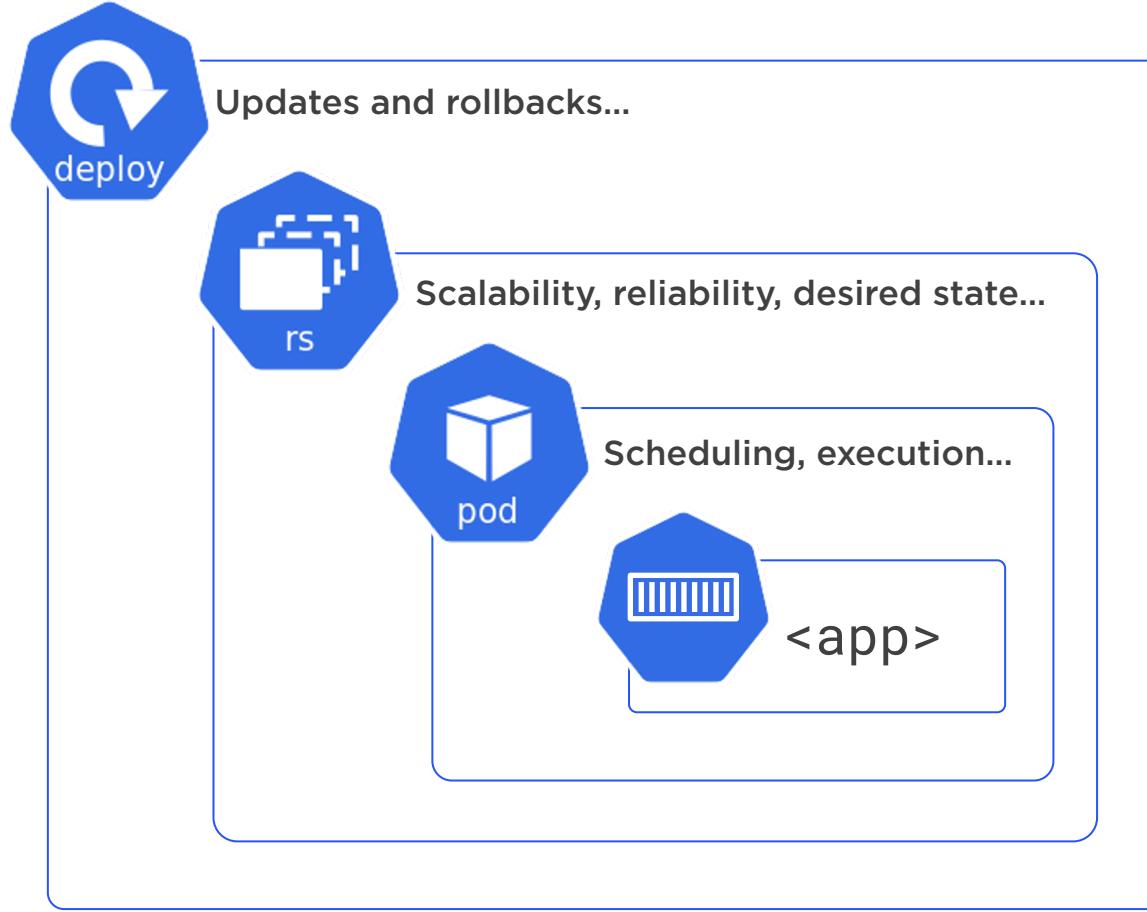




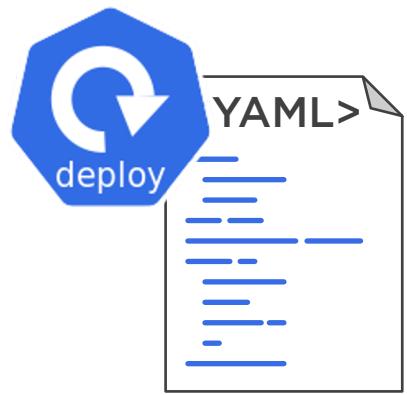


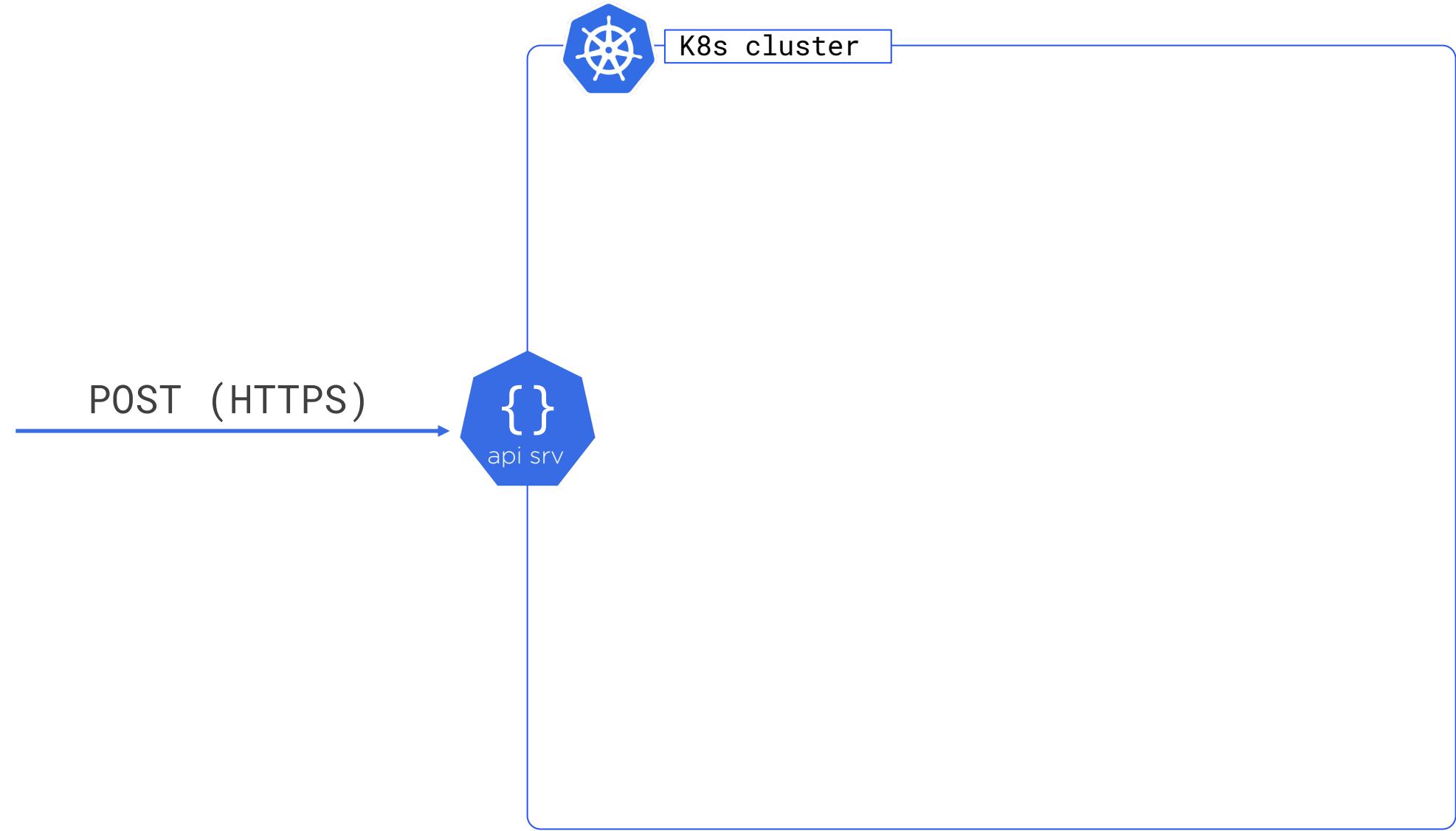
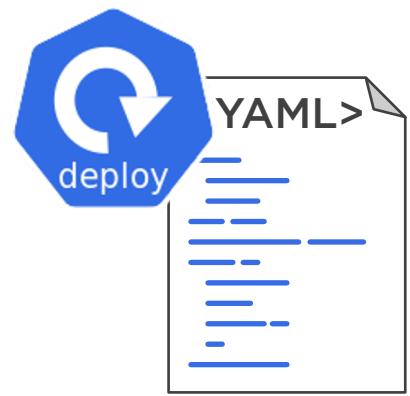


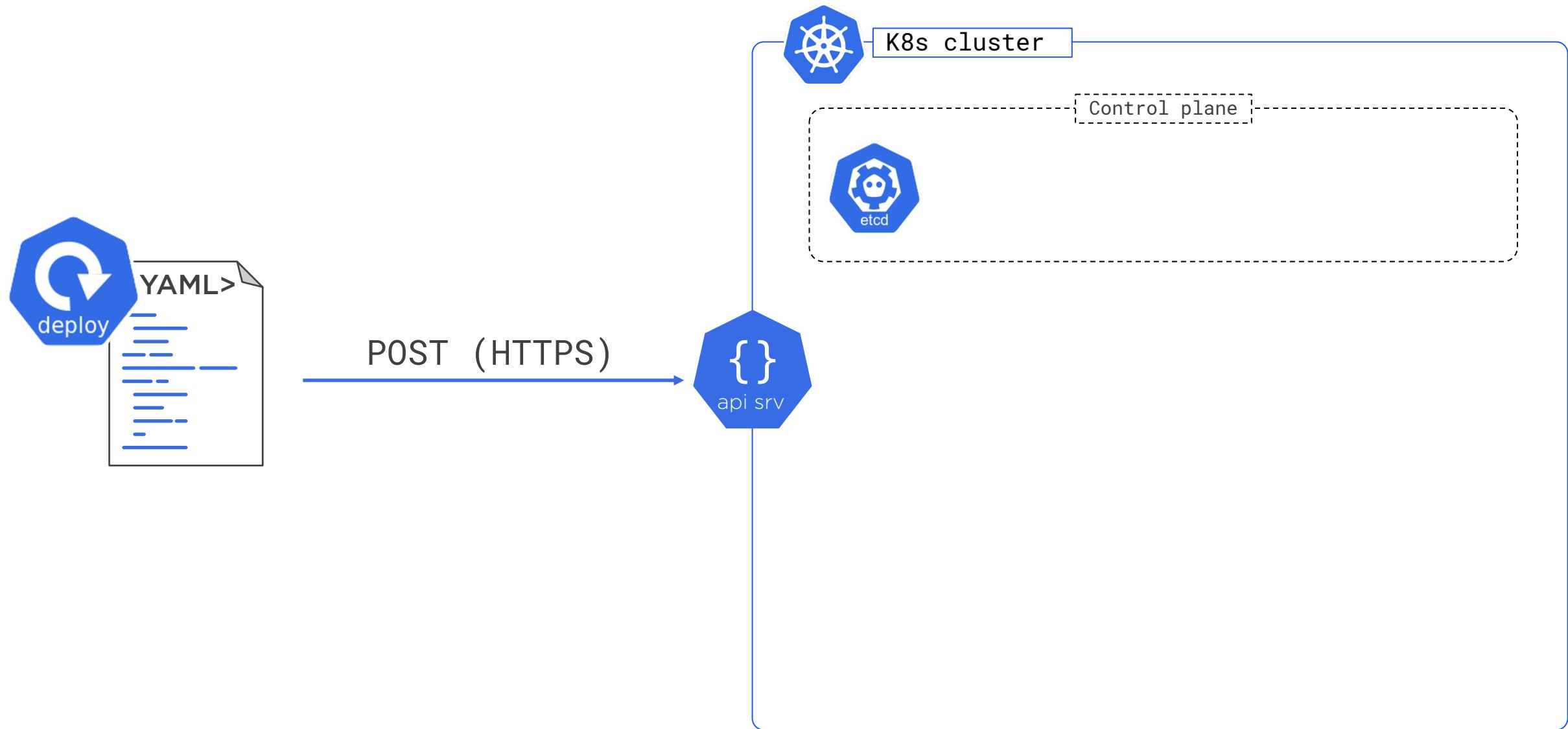


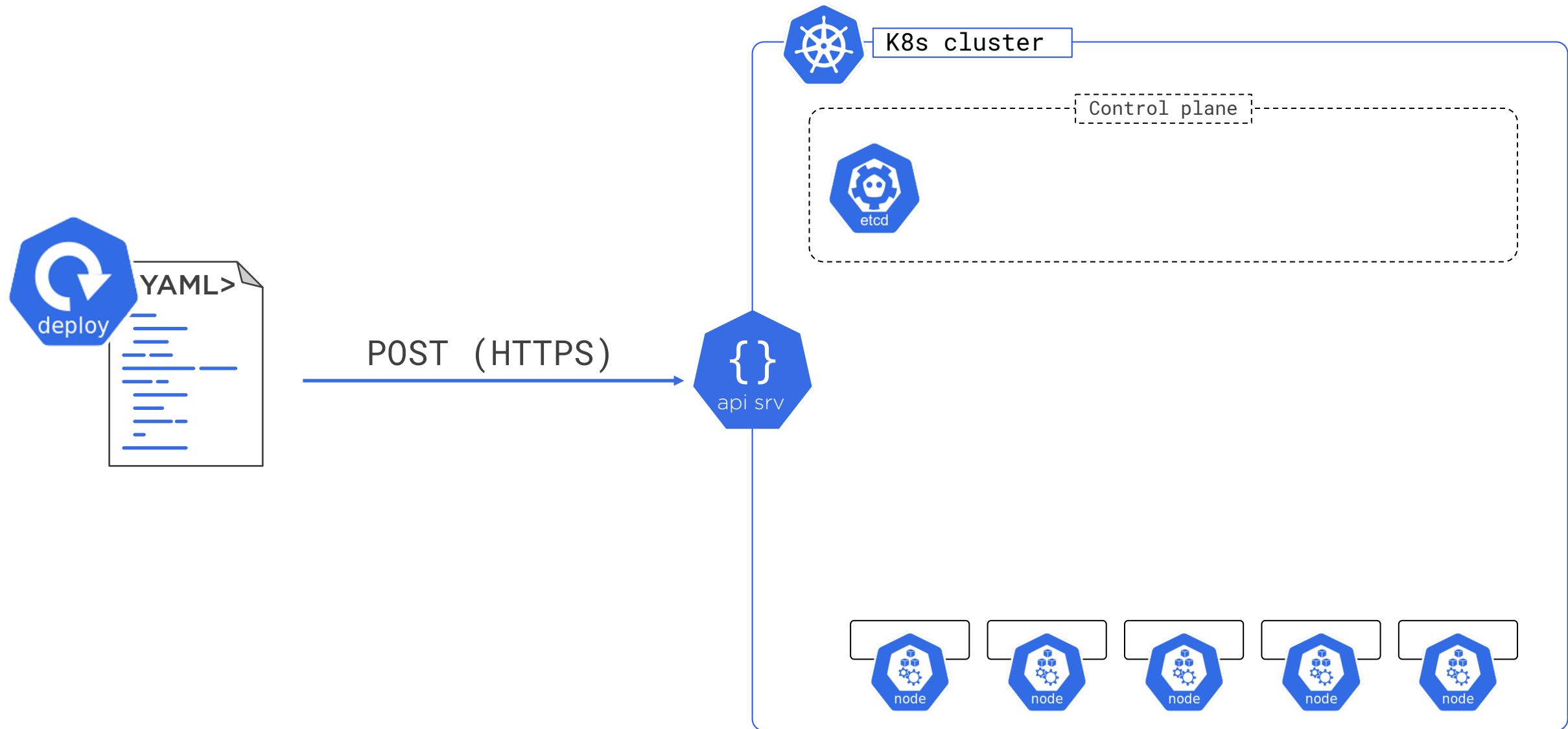


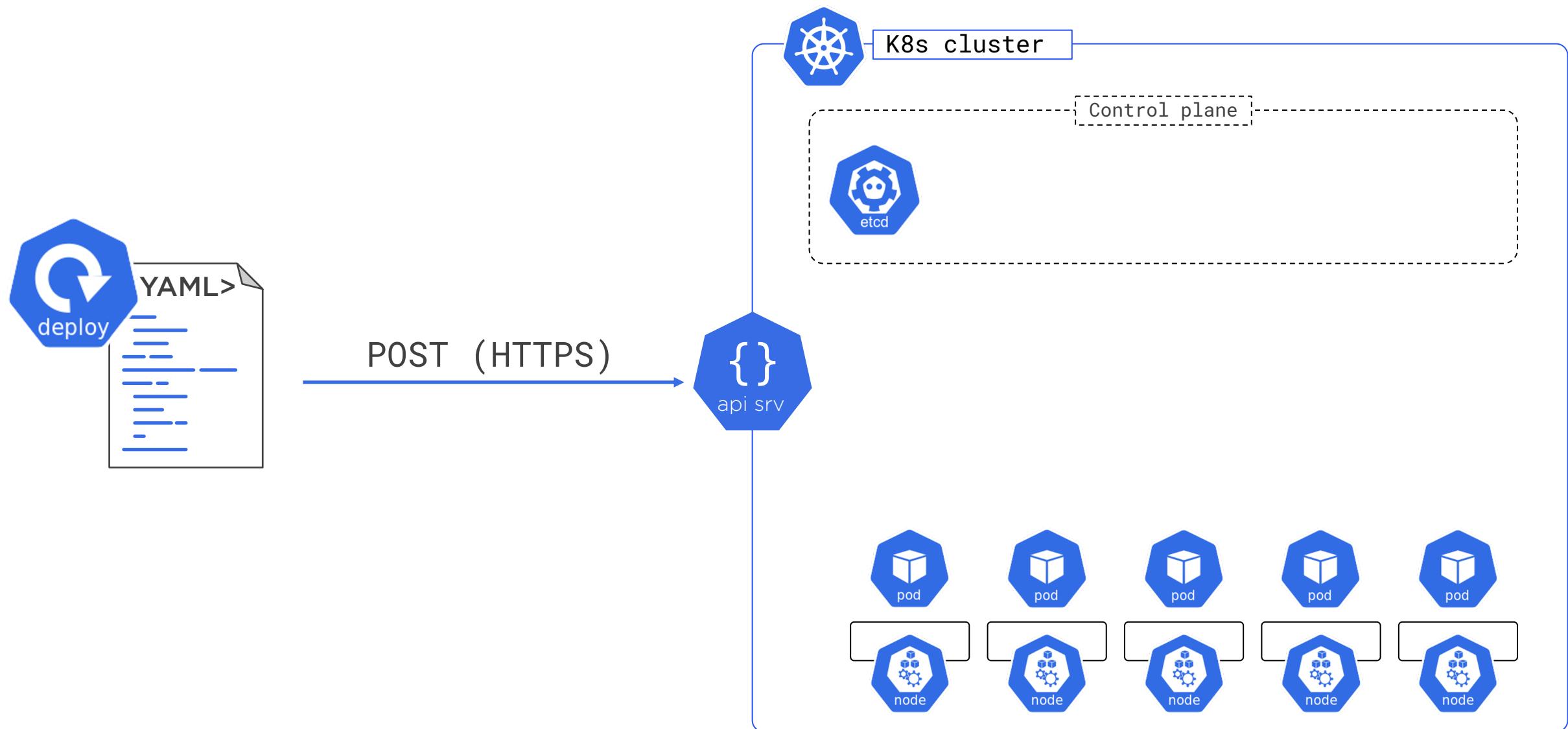


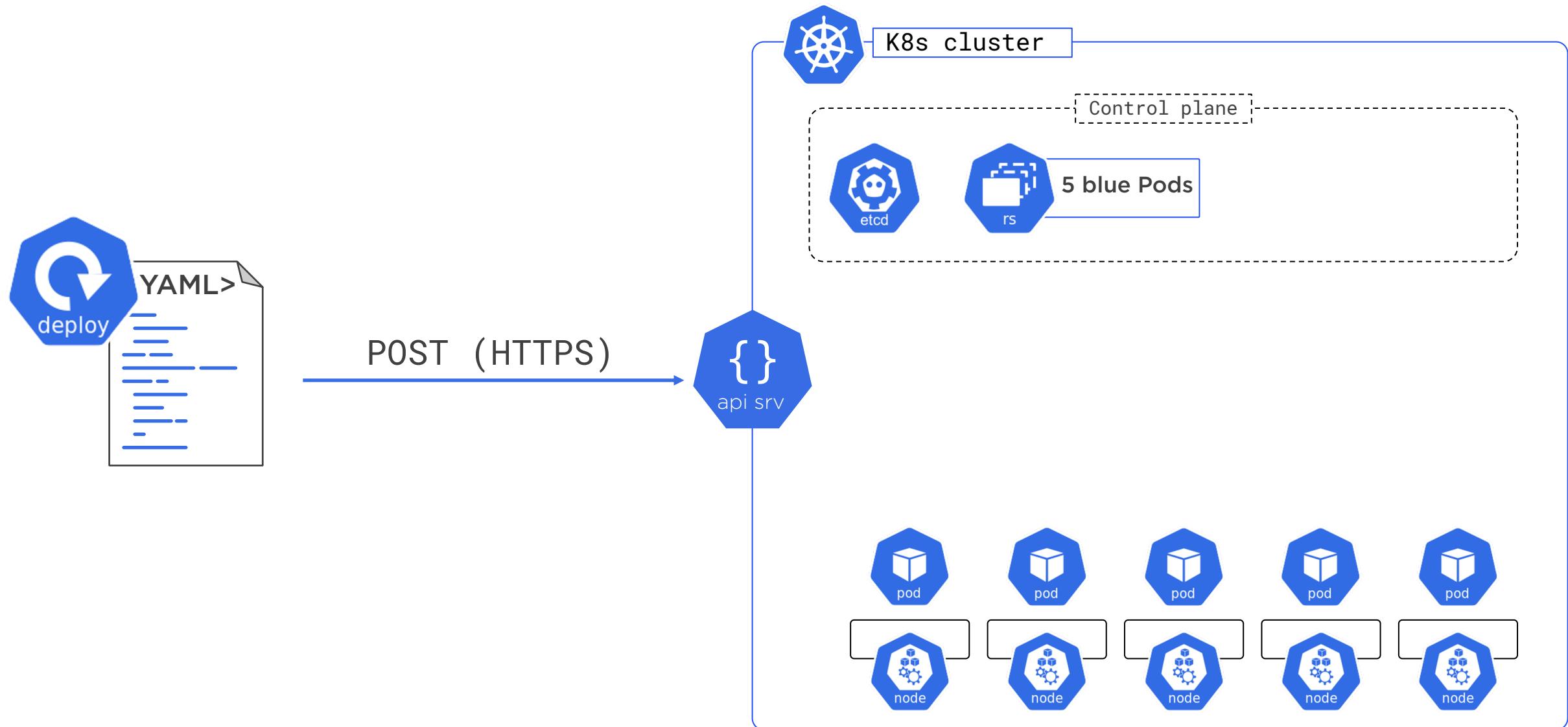


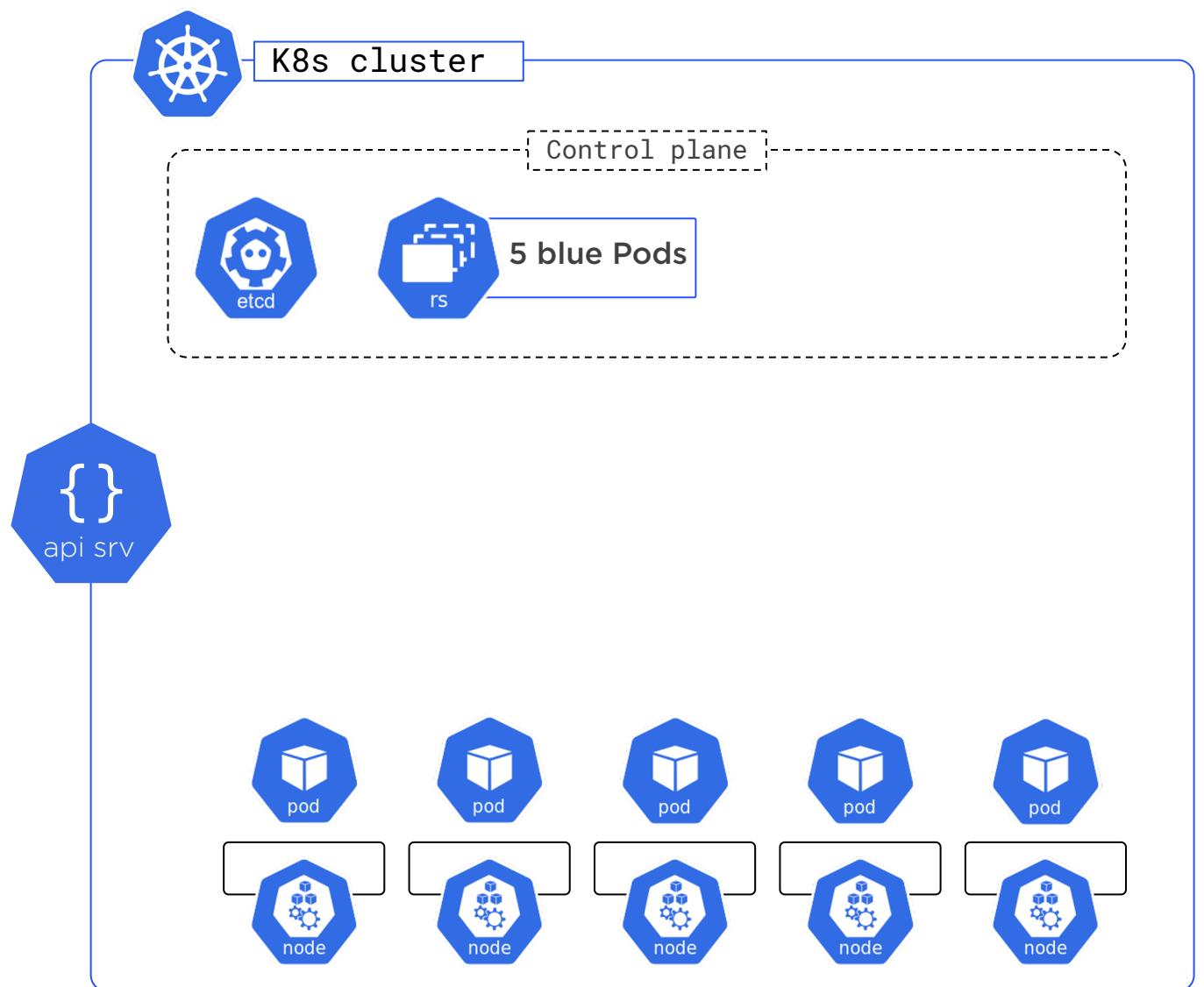
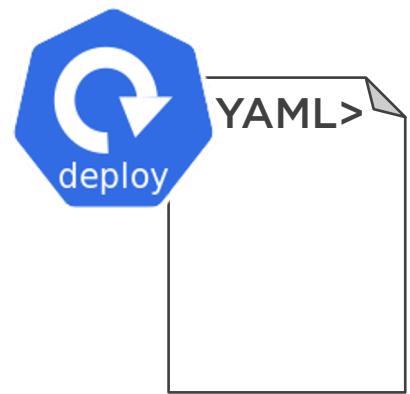


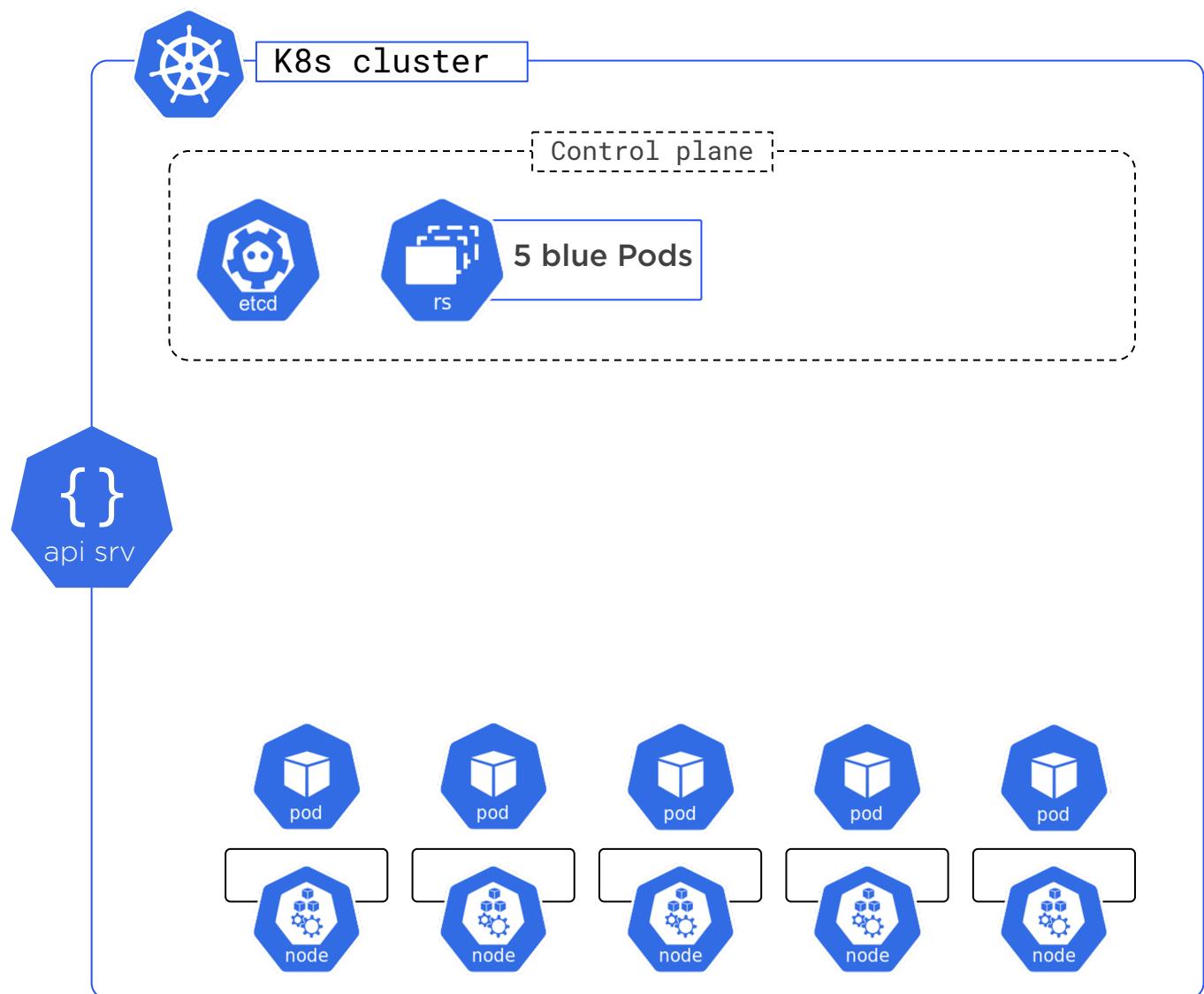
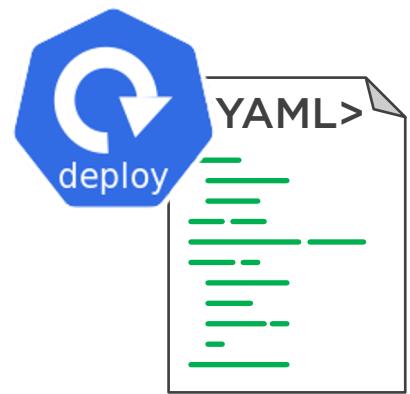


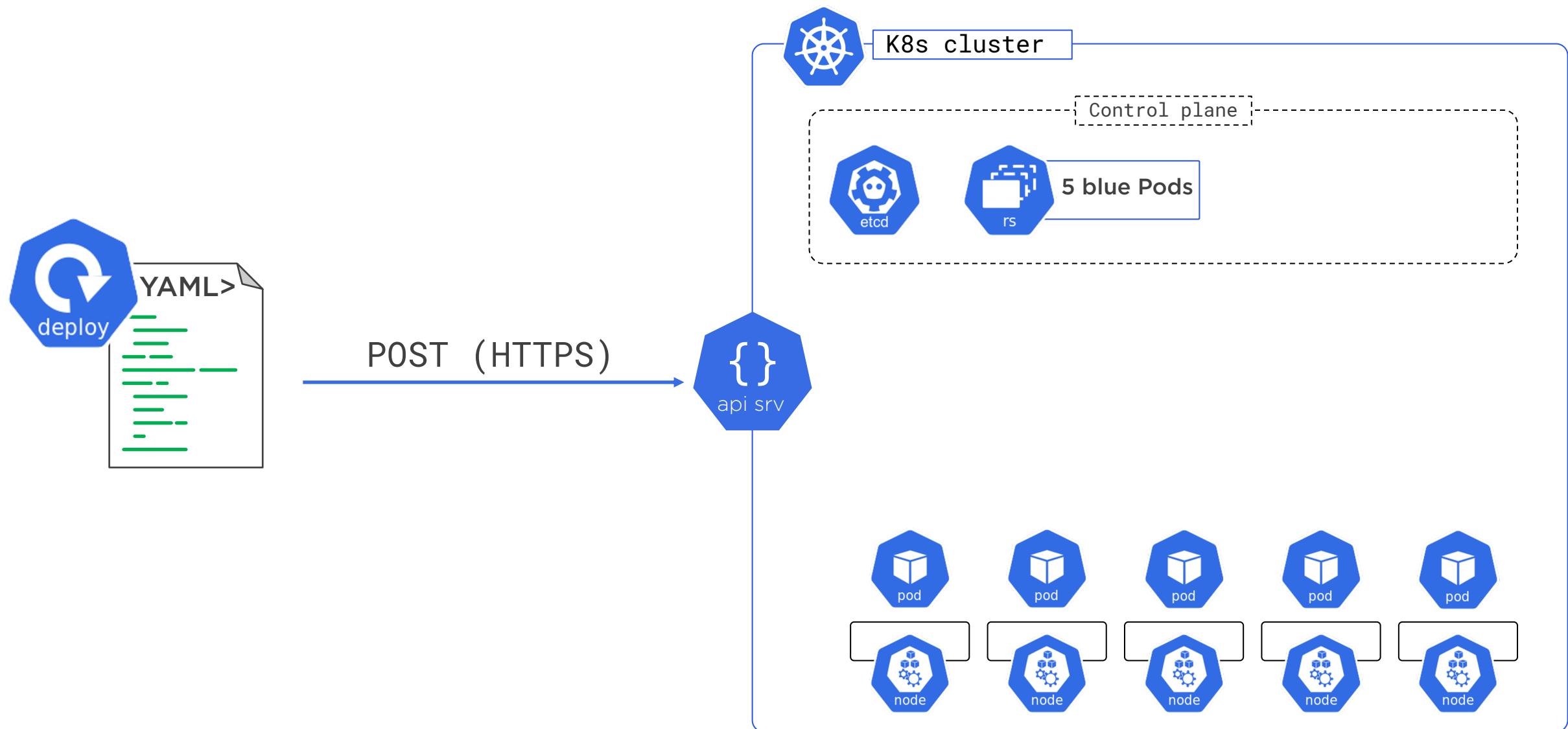


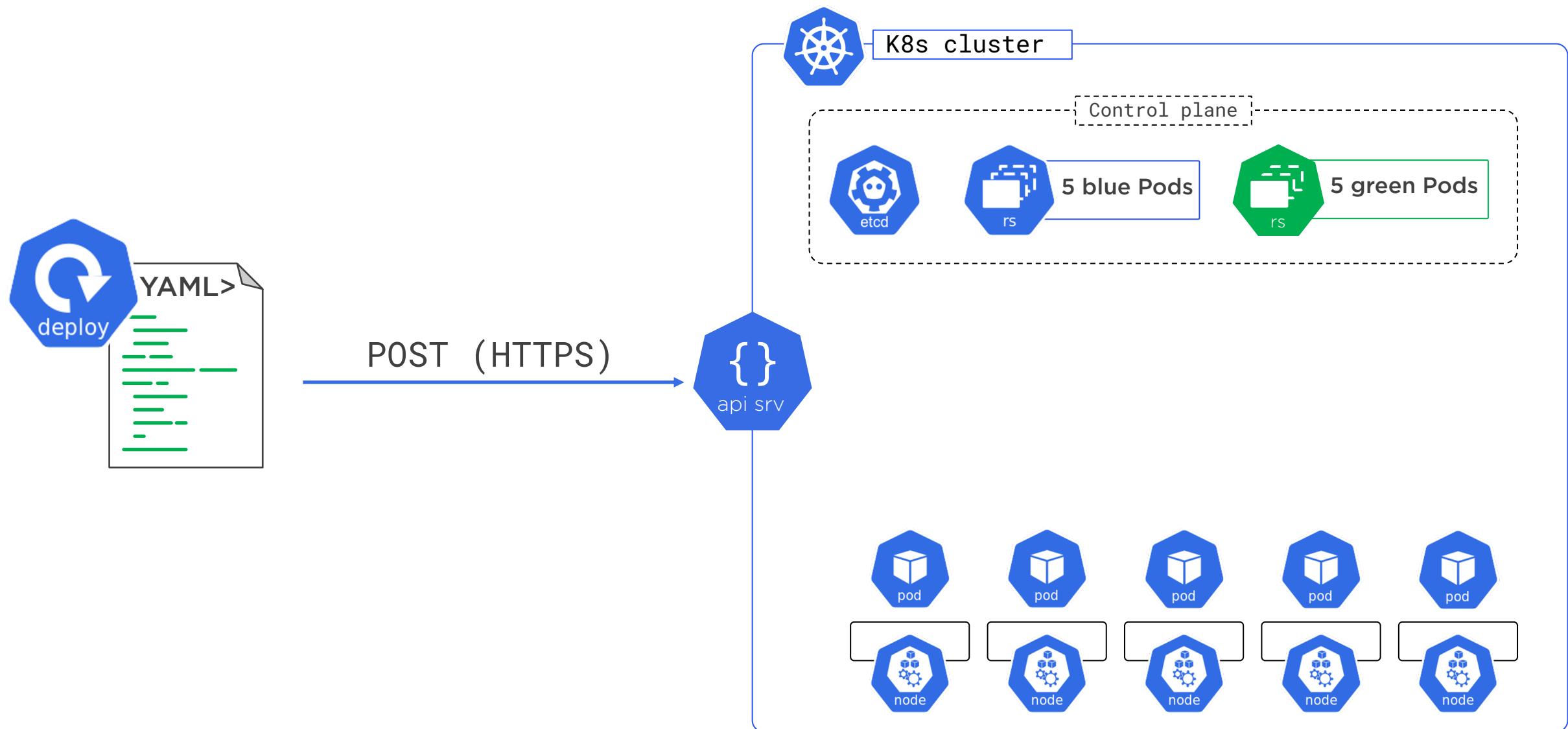


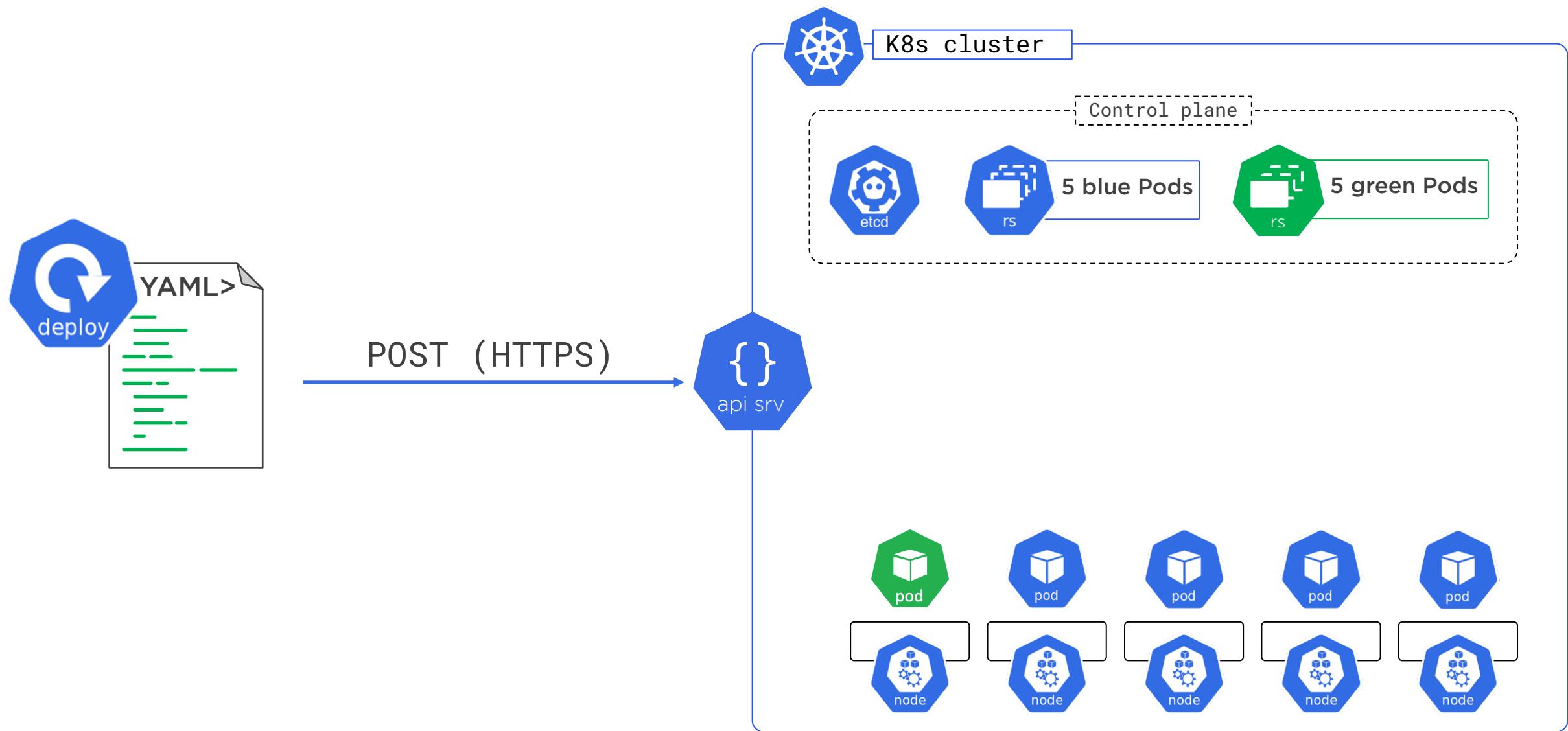


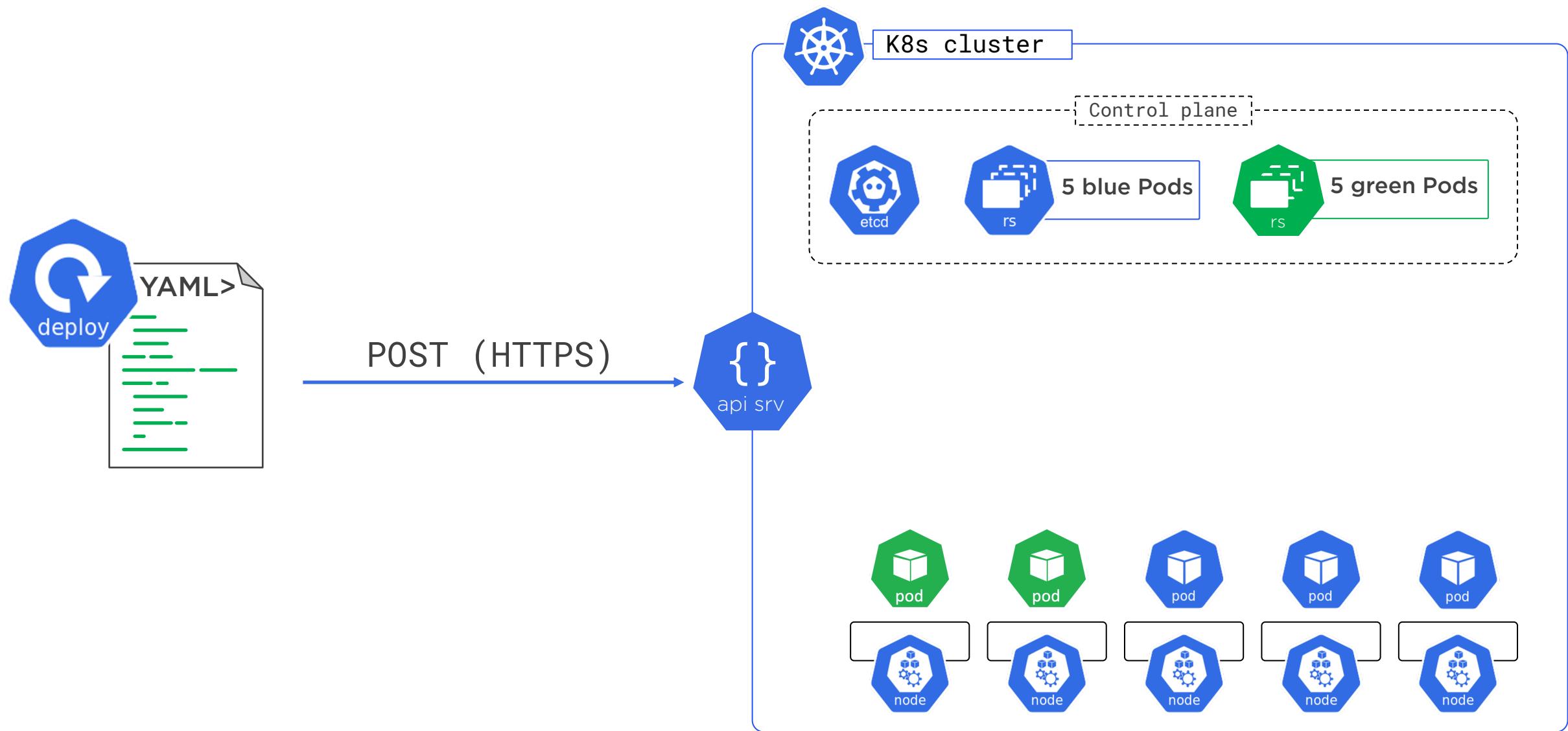


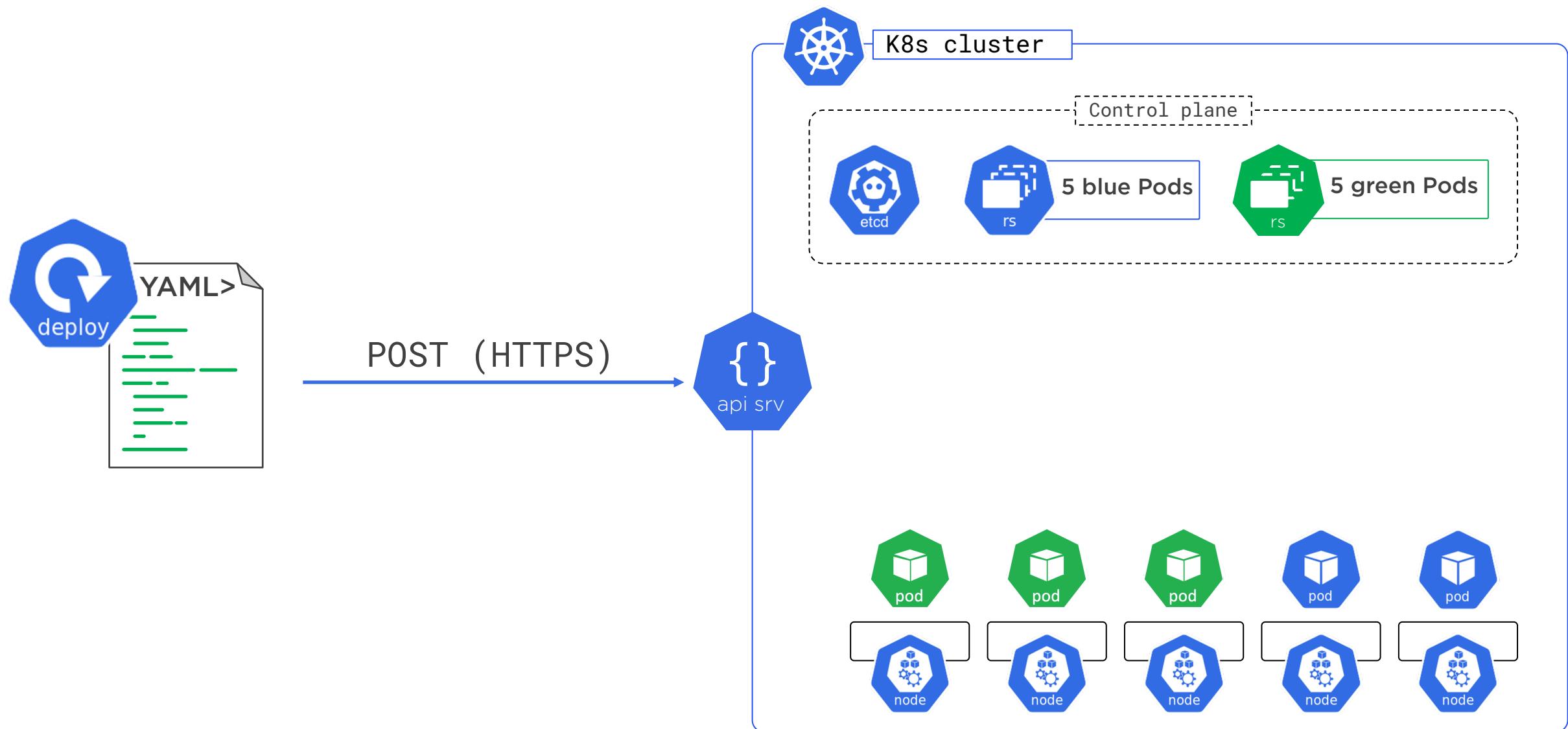


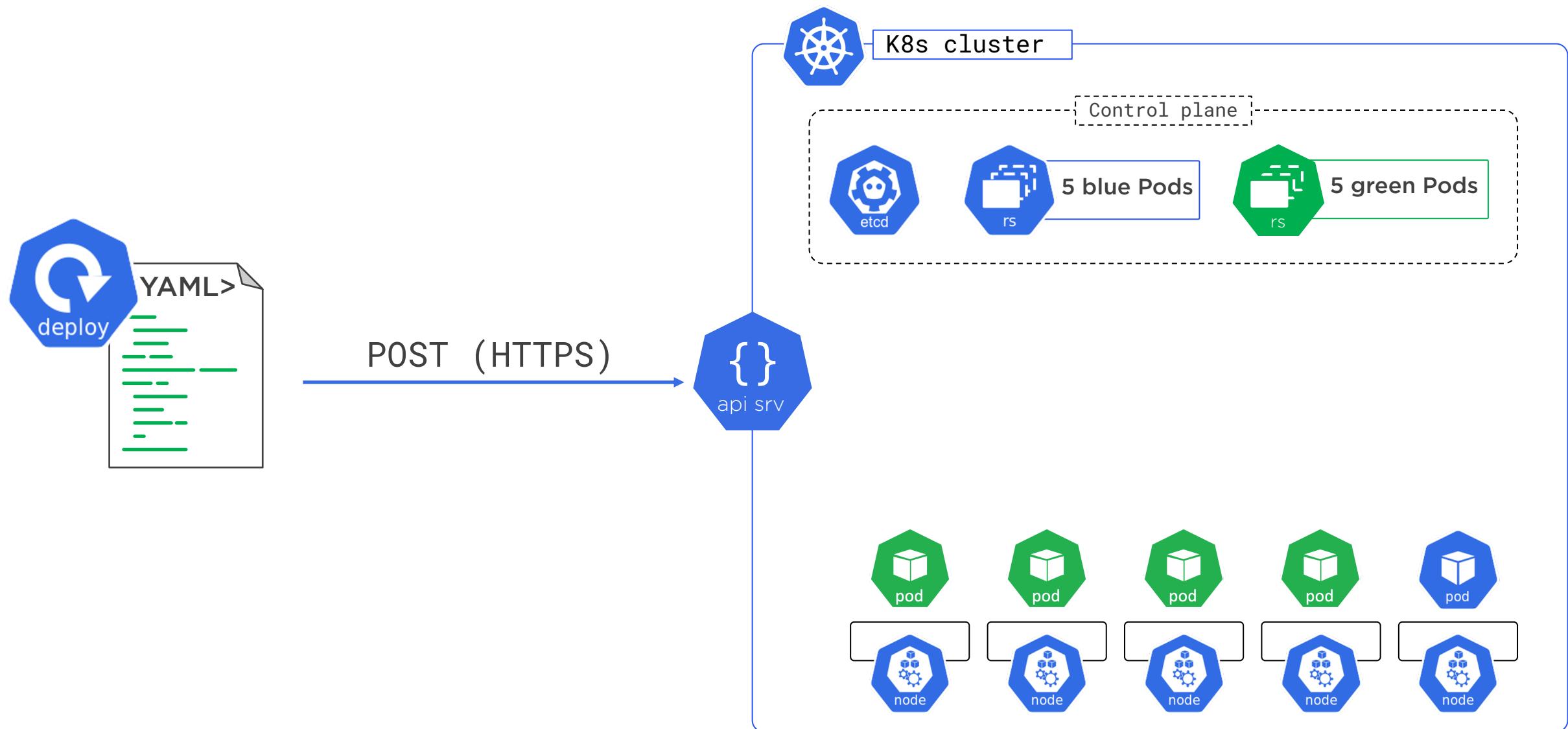


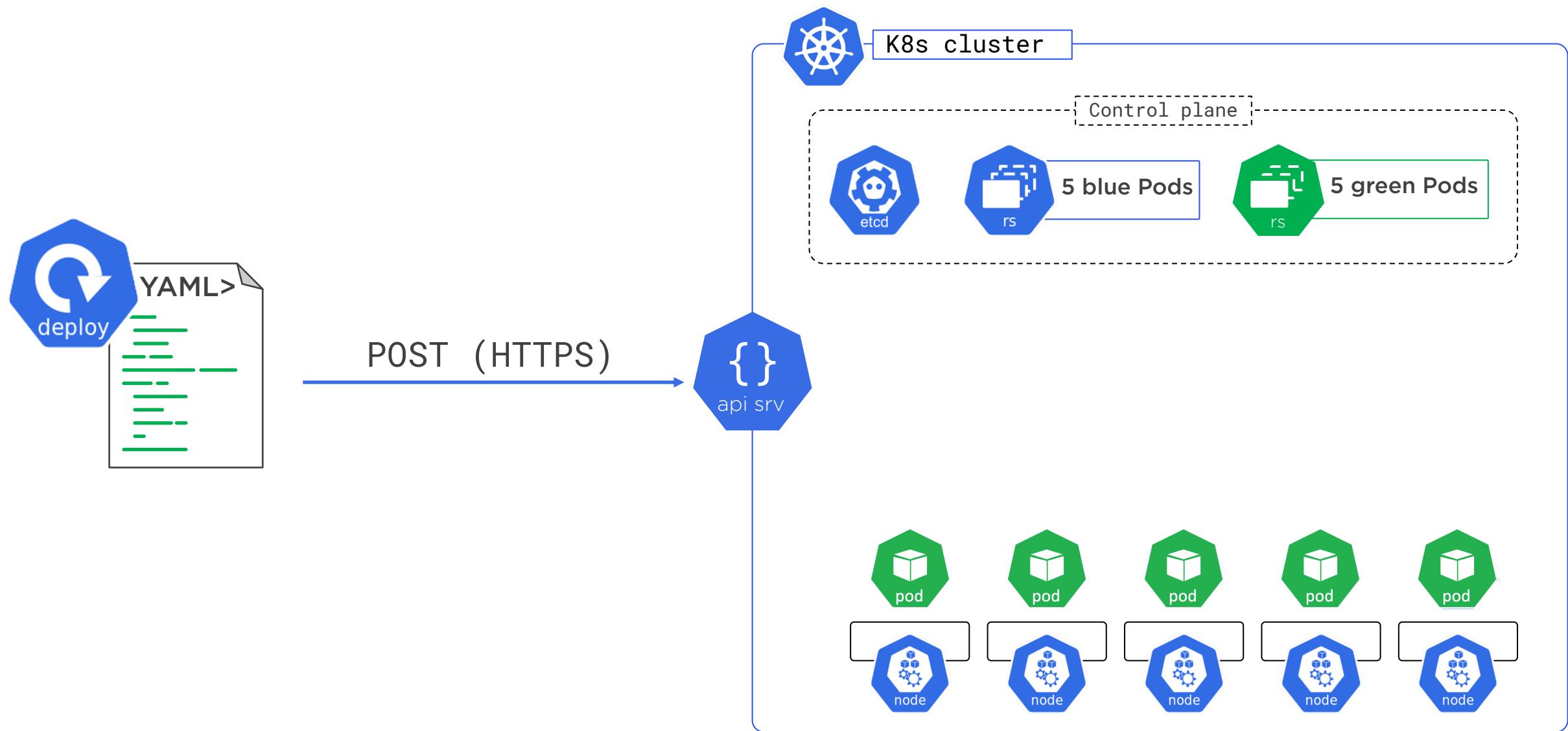


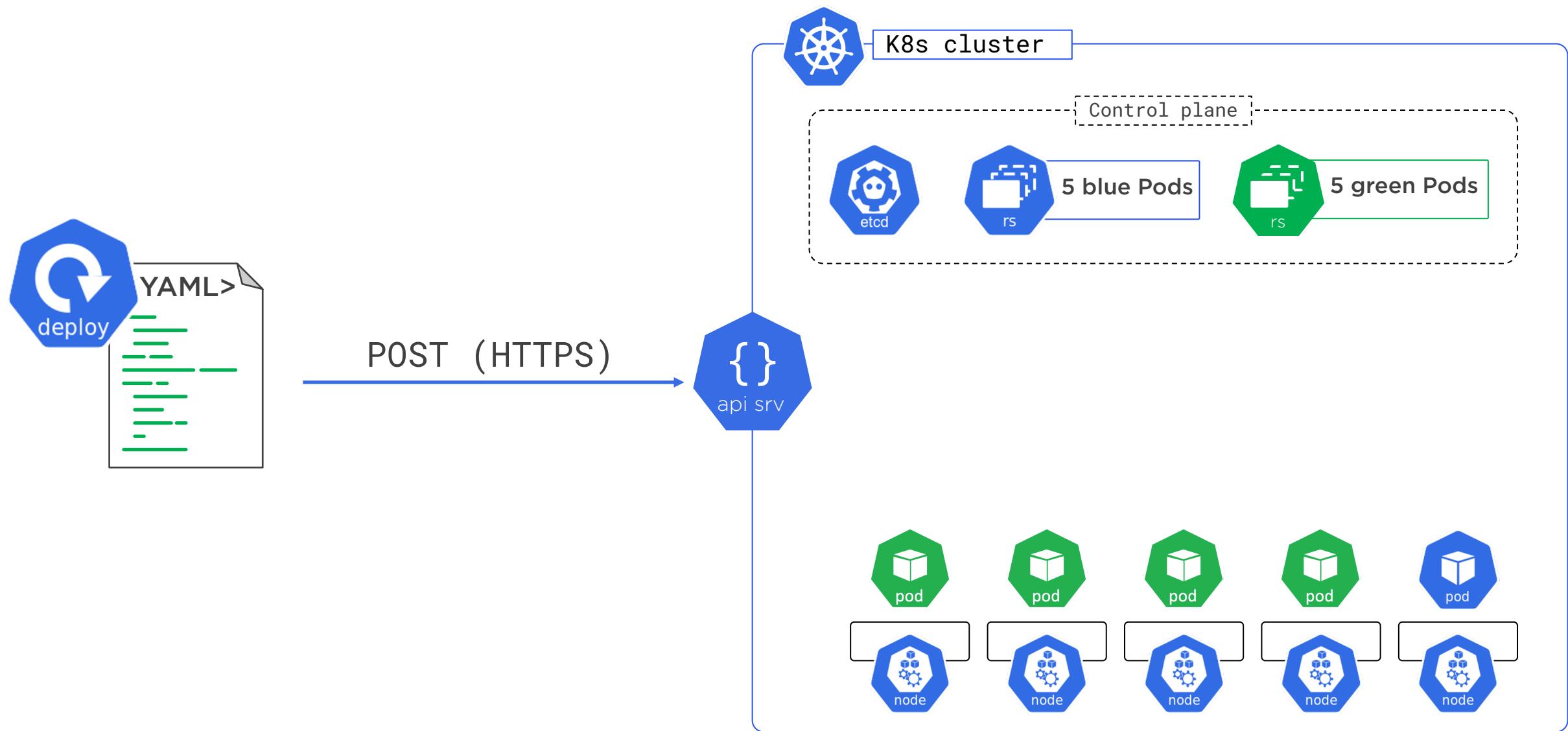


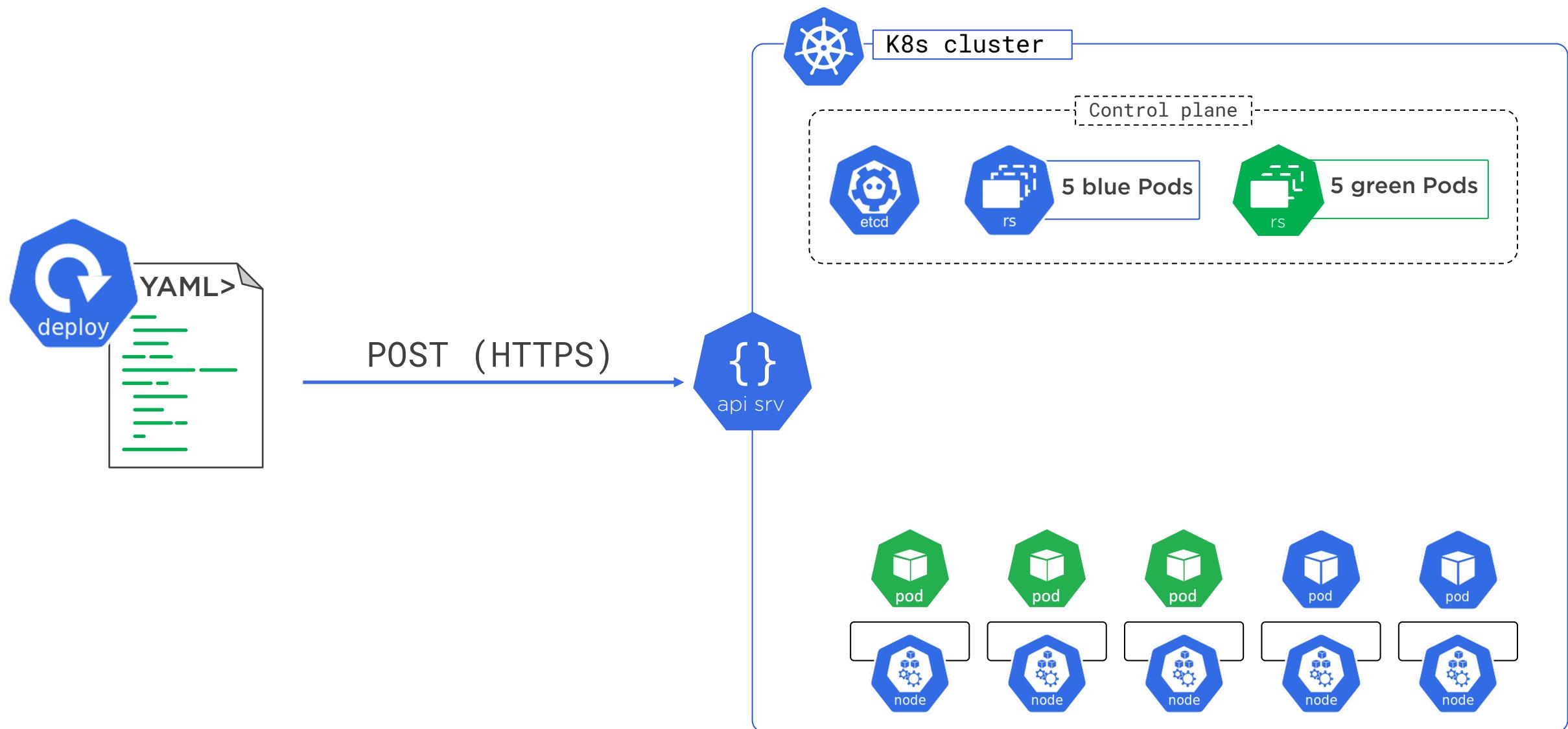


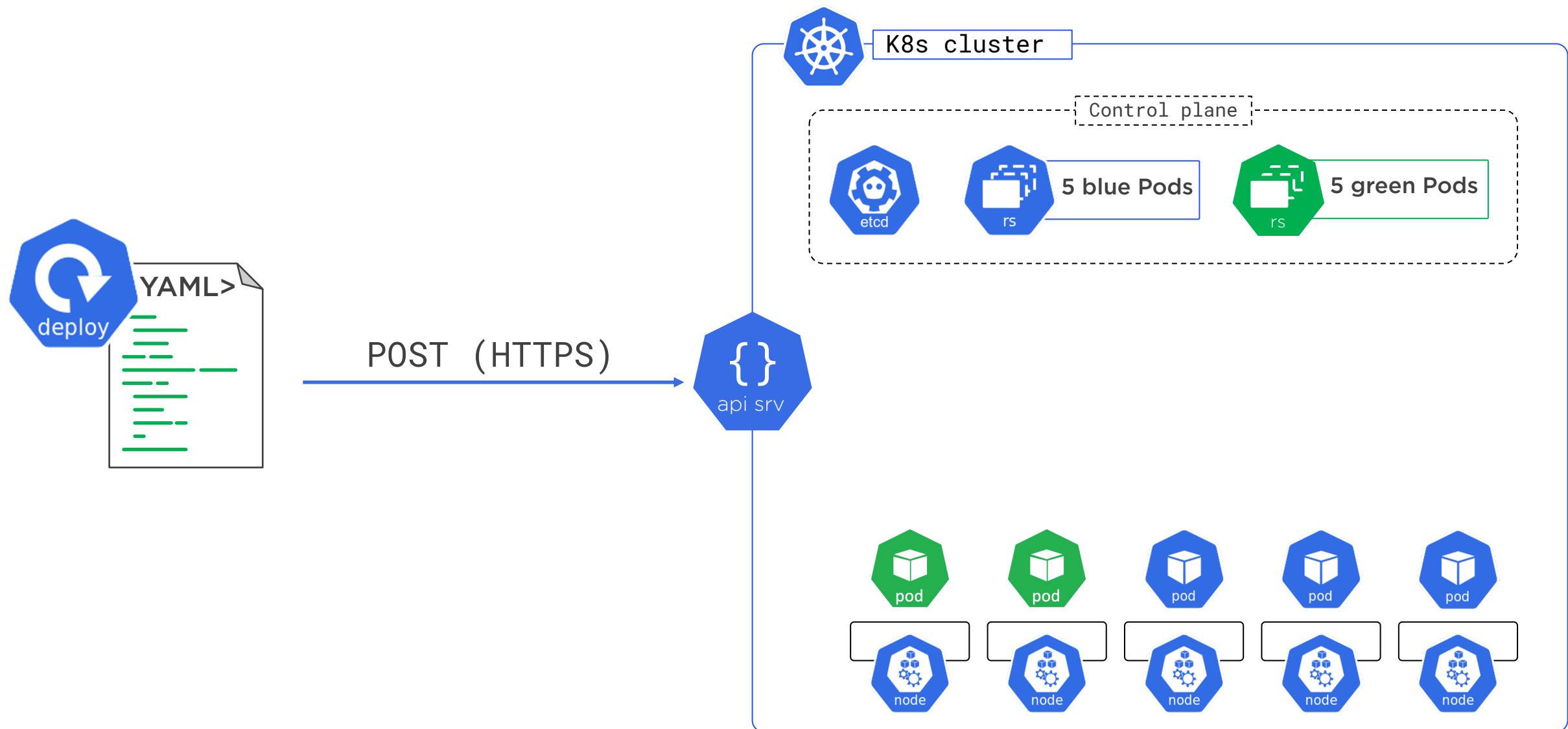


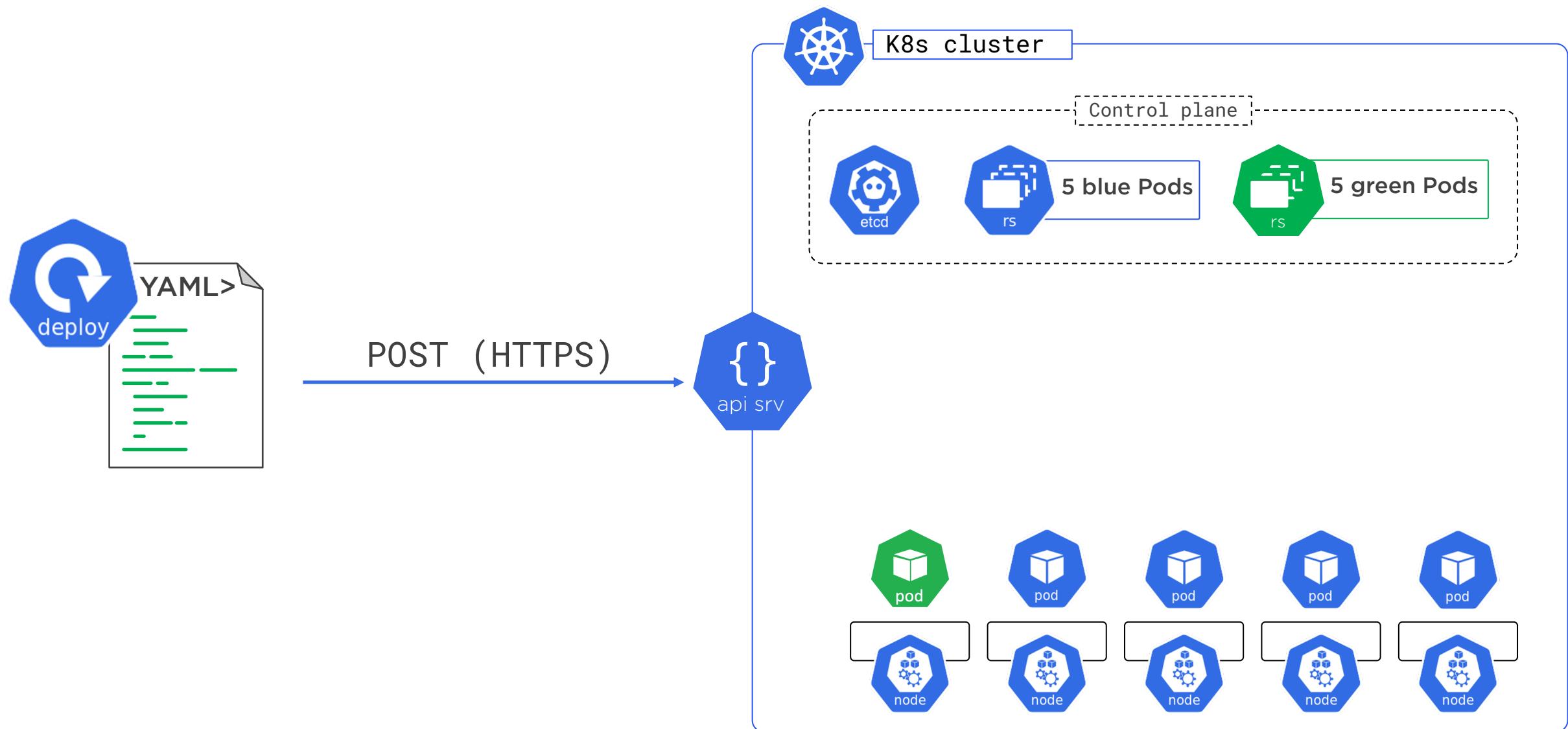


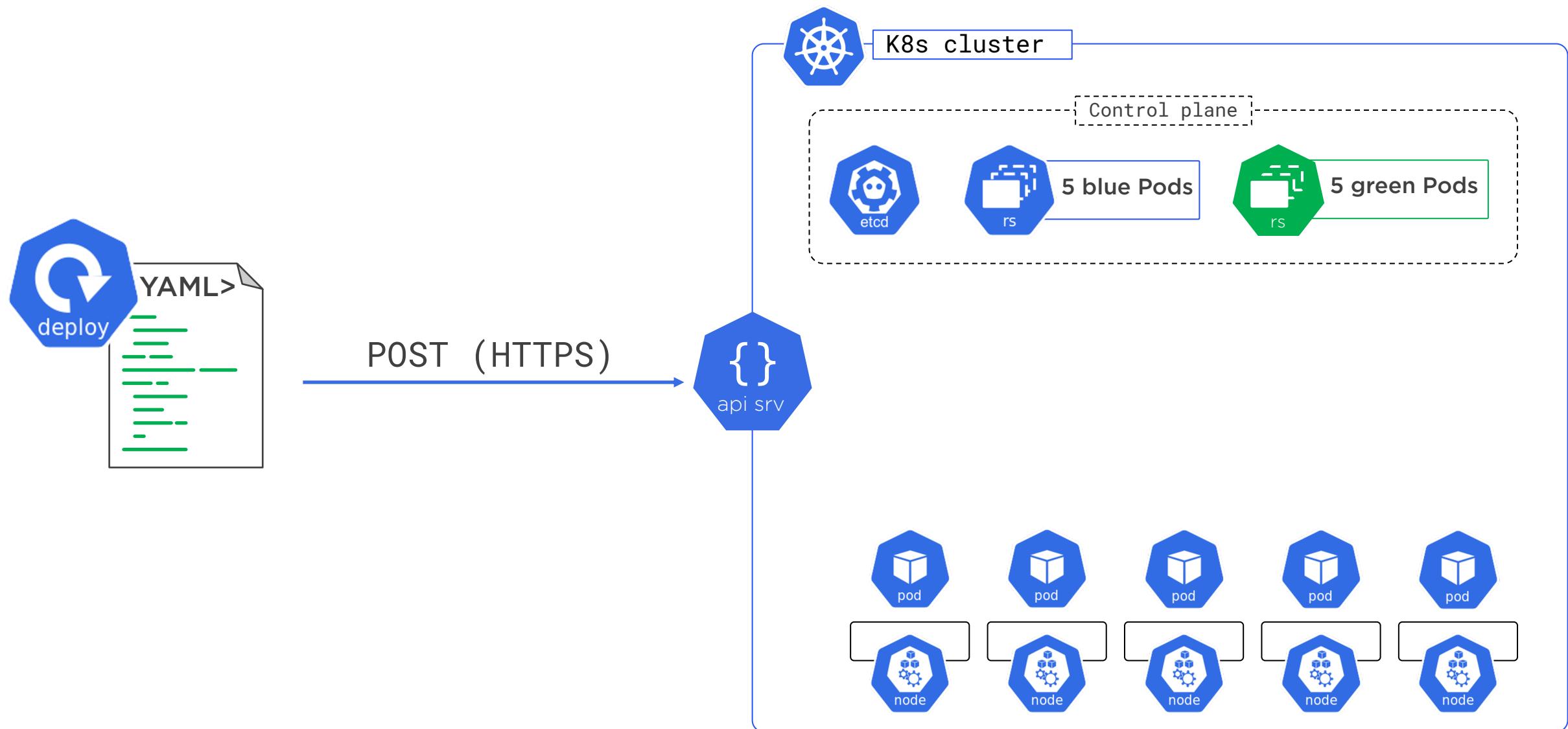






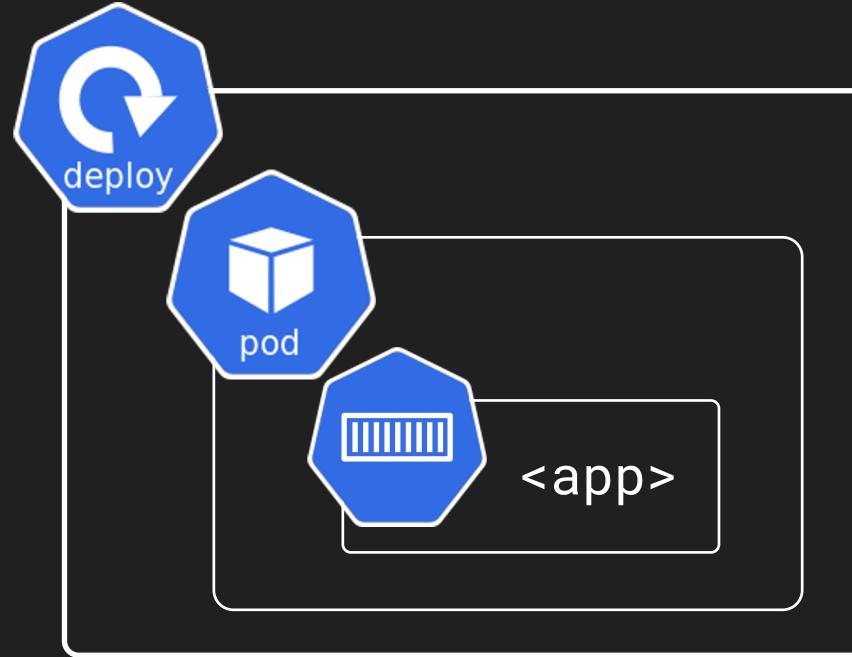






Up Next:
Creating a Deployment YAML





Creating a Deployment YAML



Up Next:
Deploying a Deployment



Deploying a Deployment



Up Next:
Self-healing and Scaling



Self-healing and Scaling



Up Next:
Rolling Updates and Rollbacks



Rolling Updates and Rollbacks



Up Next: Next Steps



What Next



Nigel Poulton

@nigelpoulton www.nigelpoulton.com



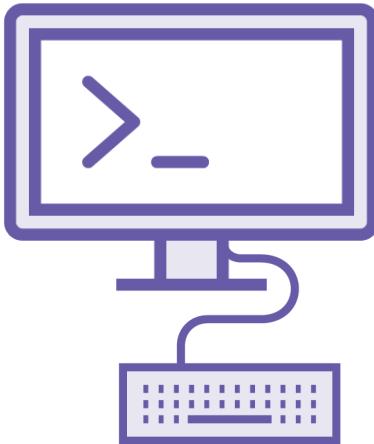
You've learned loads!

Loads of theory and a bunch of hands-on...



@nigelpoulton



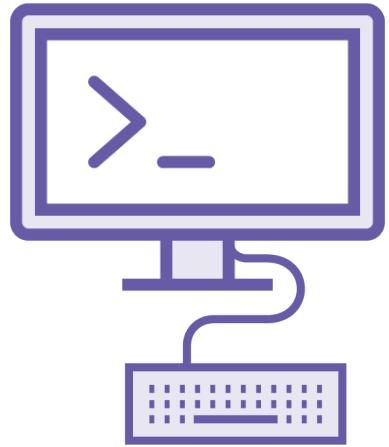


Feel free to
make mistakes!



@nigelpoulton





Feel free to
make mistakes!

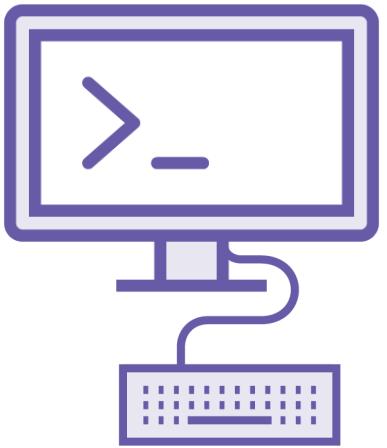


K8s courses and
learning paths



@nigelpoulton

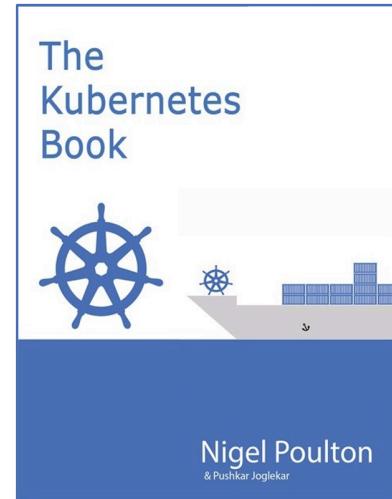




Feel free to
make mistakes!



K8s courses and
learning paths

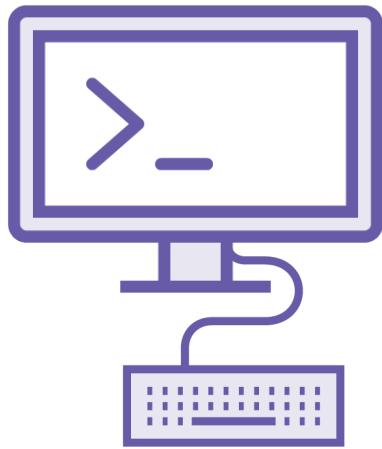


Also available
on Audible



@nigelpoulton

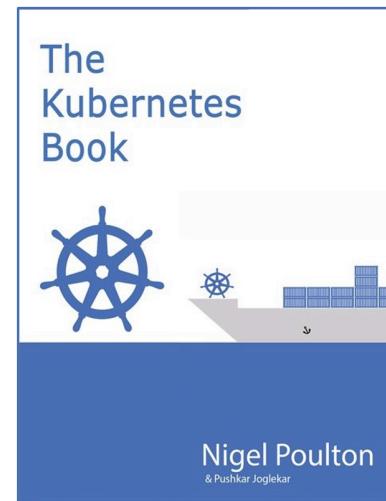




Feel free to
make mistakes!



K8s courses and
learning paths



Also available
on Audible

meetup



KubeCon

Come and say
Hi!



@nigelpoulton



Up Next:
A future full of containers...

