Kubernetes Architecture

Nigel Poulton

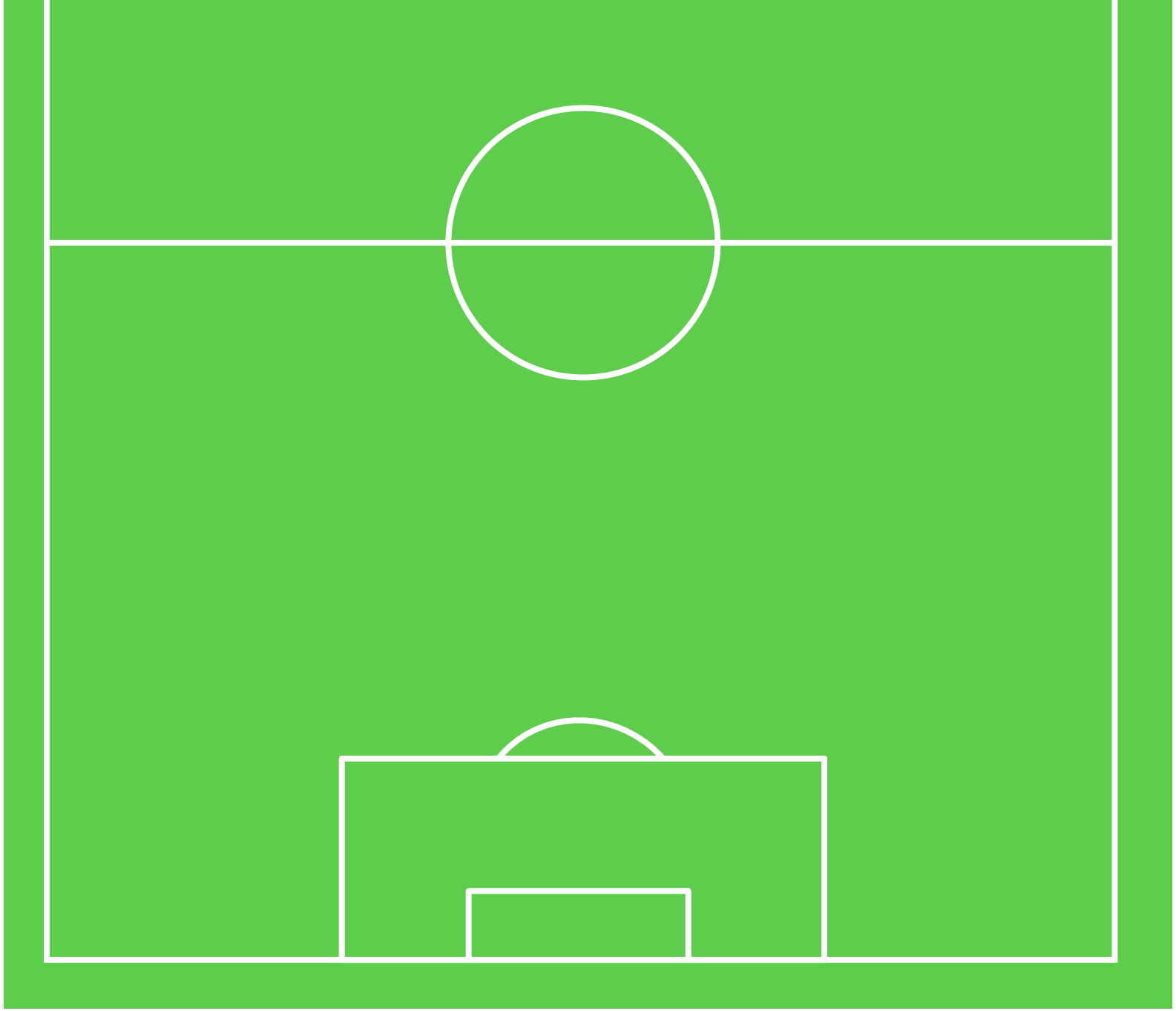
@nigelpoulton www.nigelpoulton.com

Overview

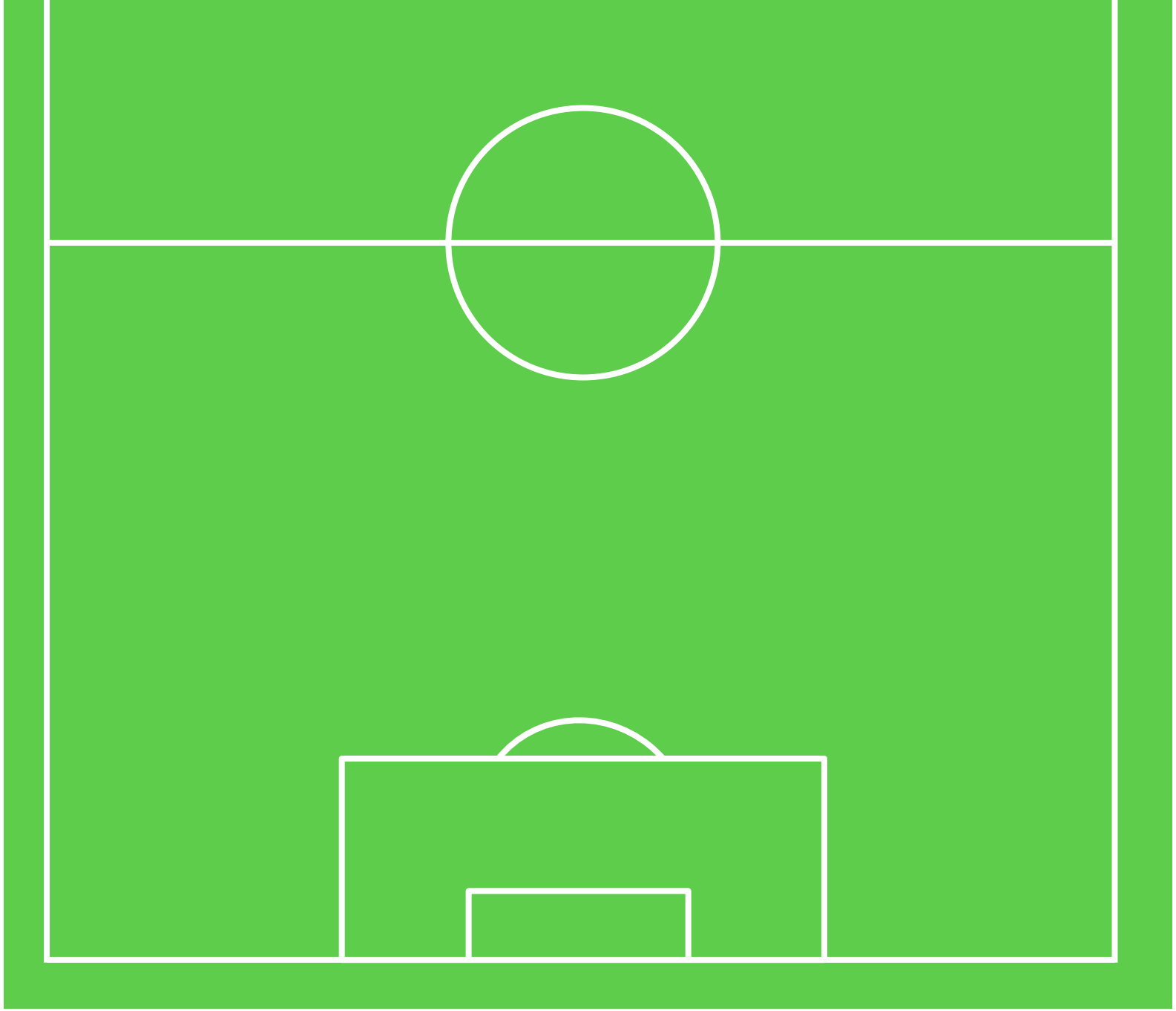
- Big Picture View
- Masters
- Nodes
- Pods
- Services
- Deployments
- The API and API Server
- Recap

Up Next:
Kubernetes: Big Picture Overview

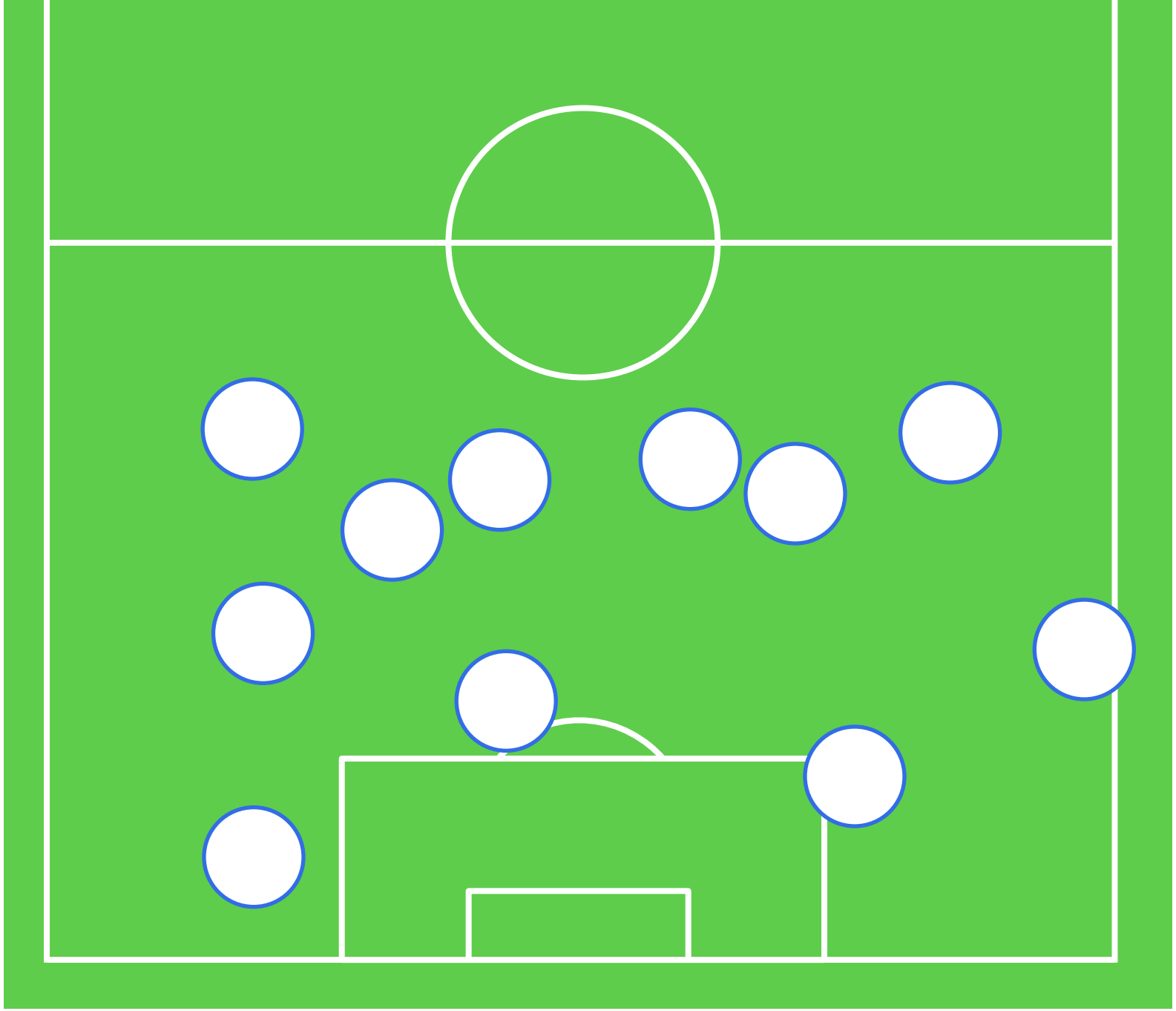
Kubernetes: Big Picture Overview



Team

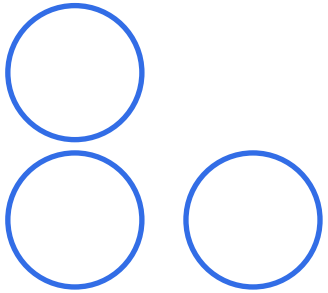


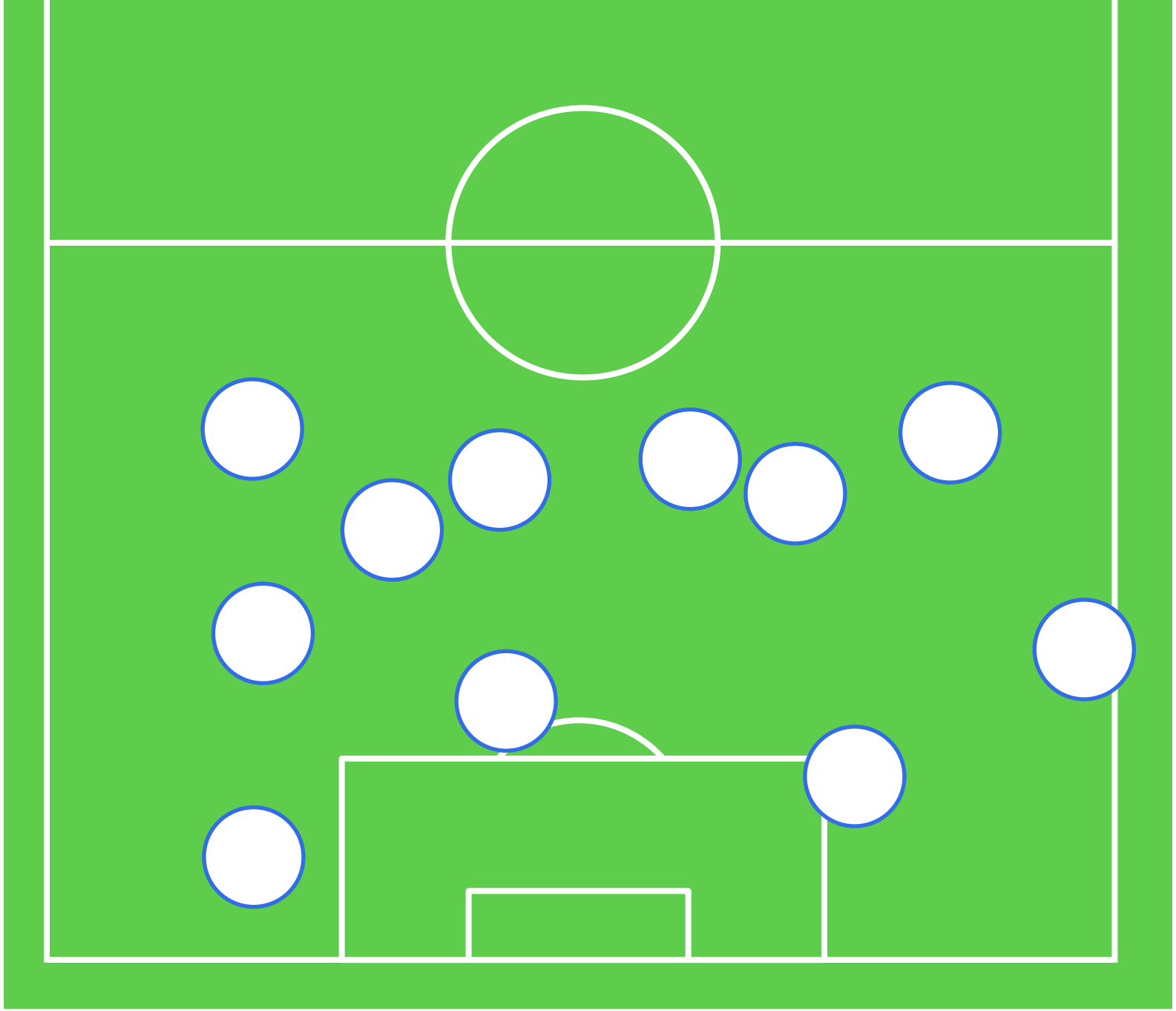
Team



Team

Manager
(coach)





Team

Manager
(coach)

S1

S2

S3

HTTPS

HTTPS

Search

Auth

K/V store

MySQL

Log

Load balancer

HTTPS

HTTPS

Search

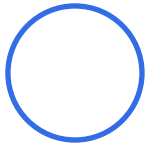
Auth

K/V store

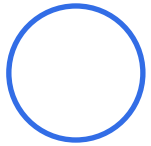
MySQL

Log

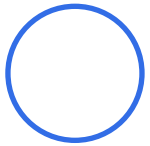
Load balancer



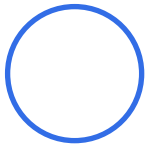
Load
balancer



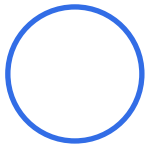
HTTPS



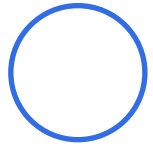
HTTPS



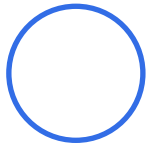
Auth



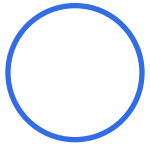
Search



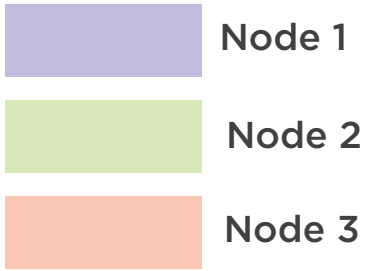
Log

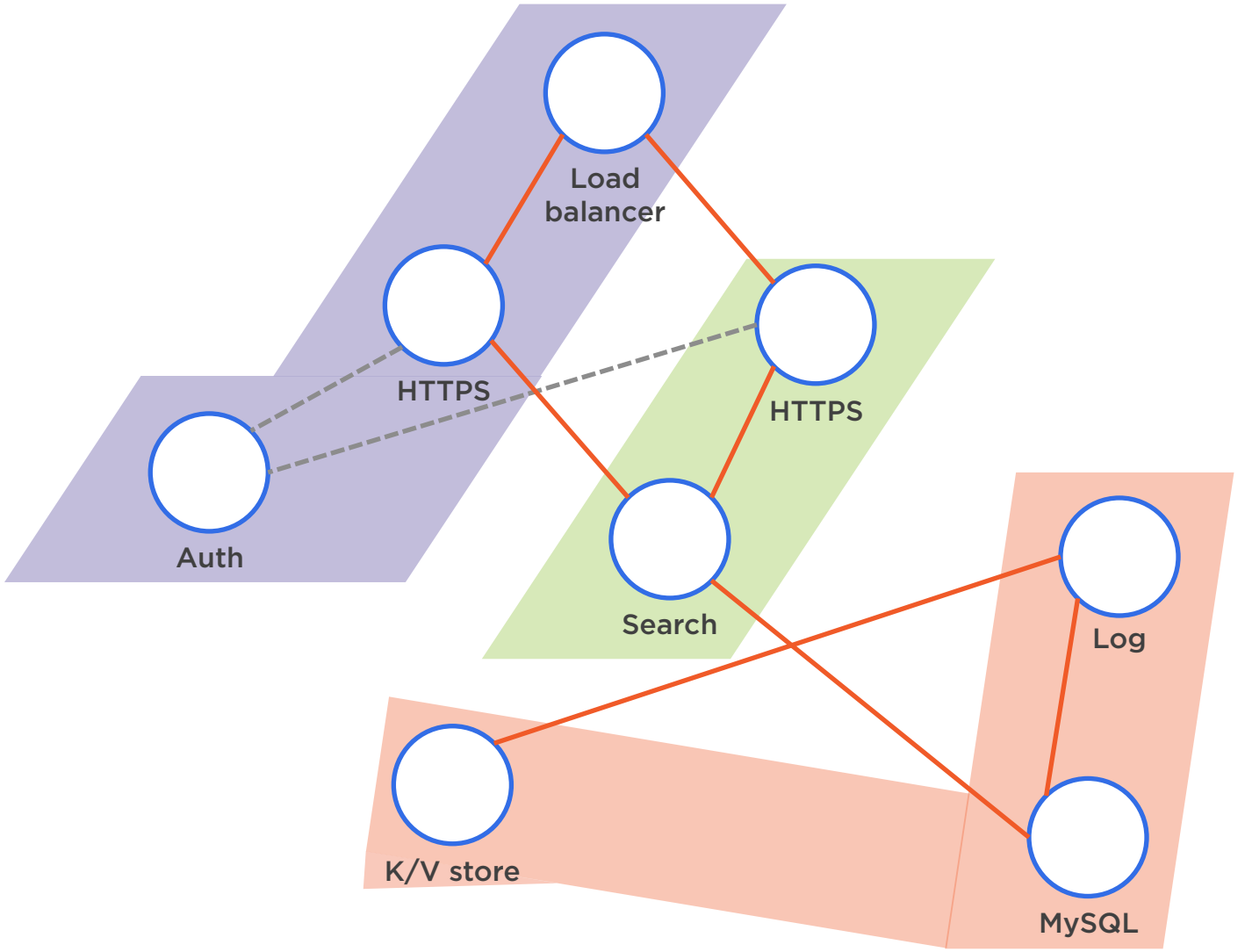


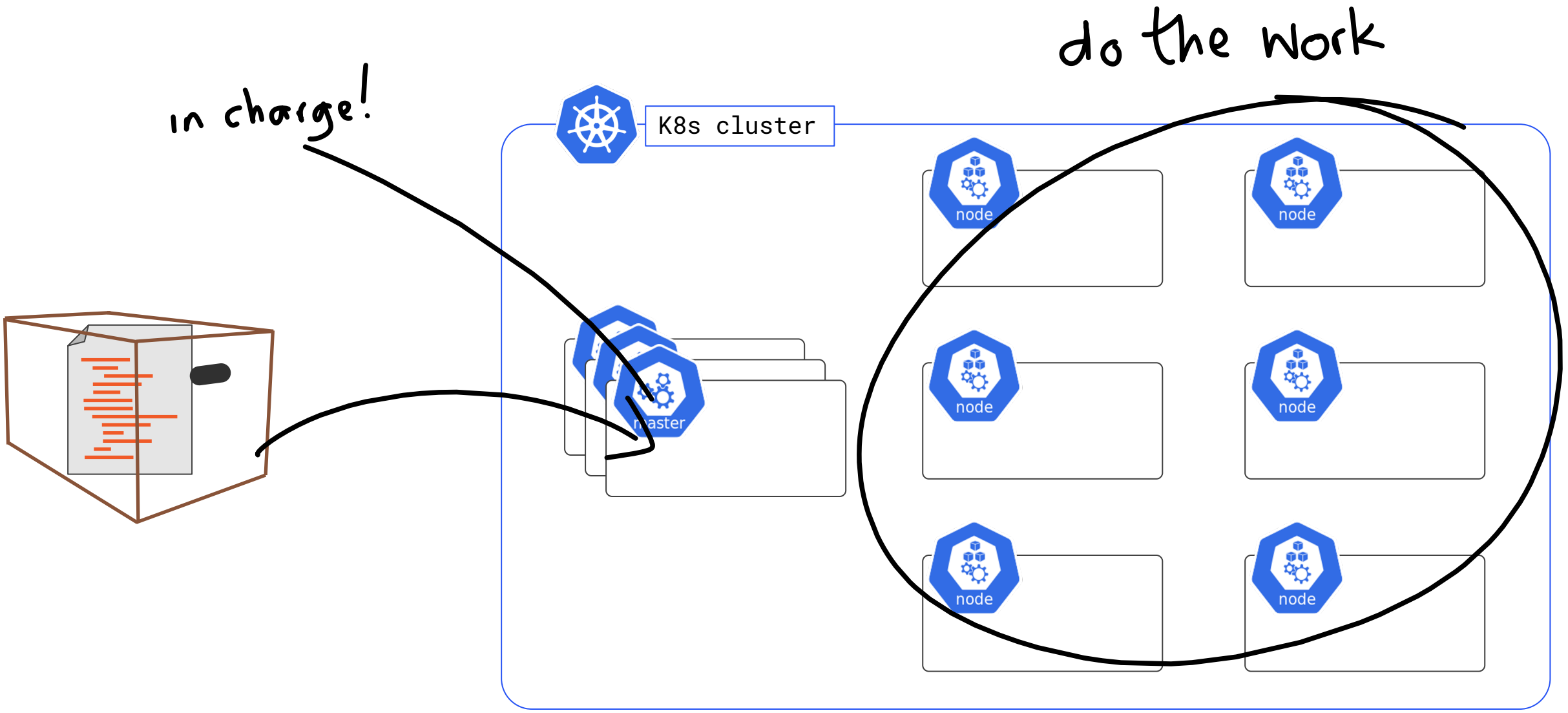
K/V store

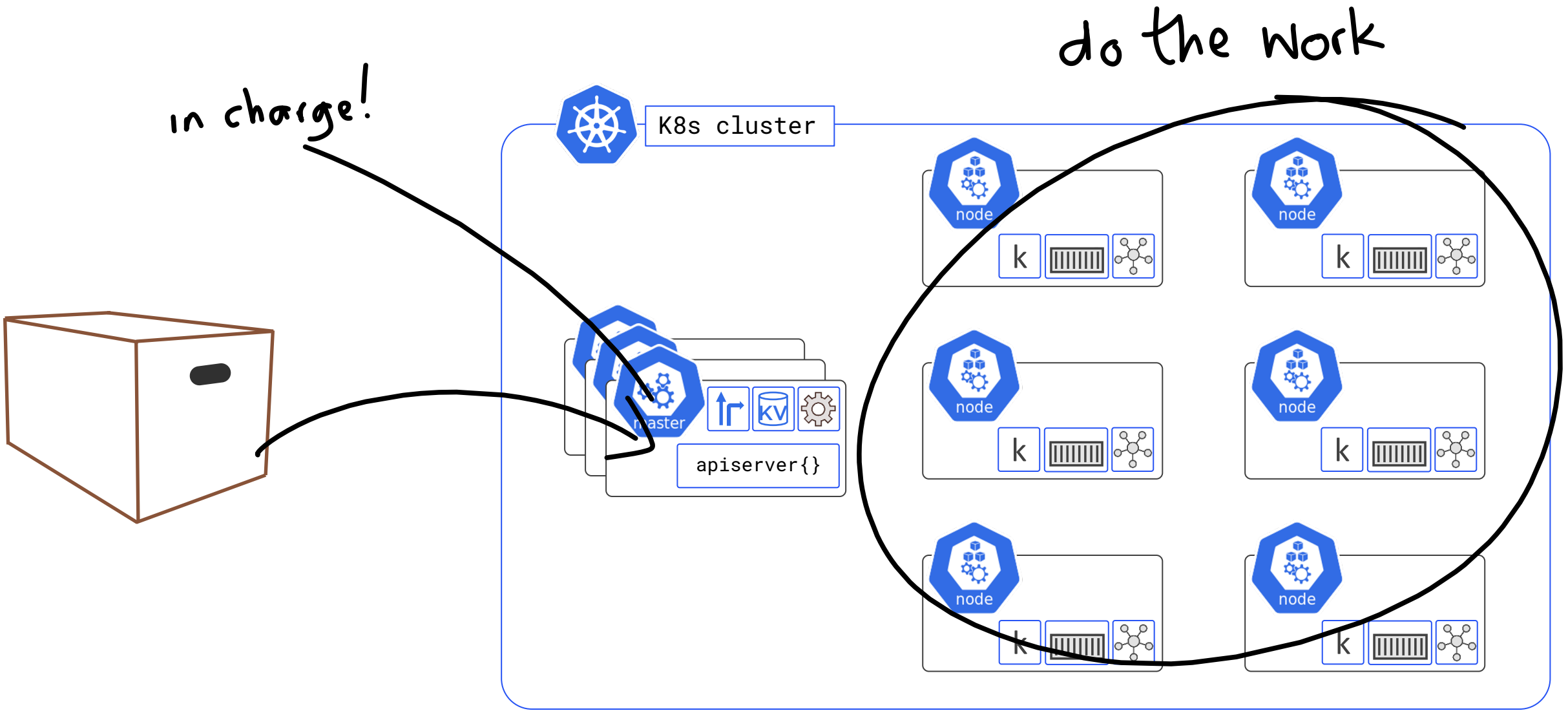


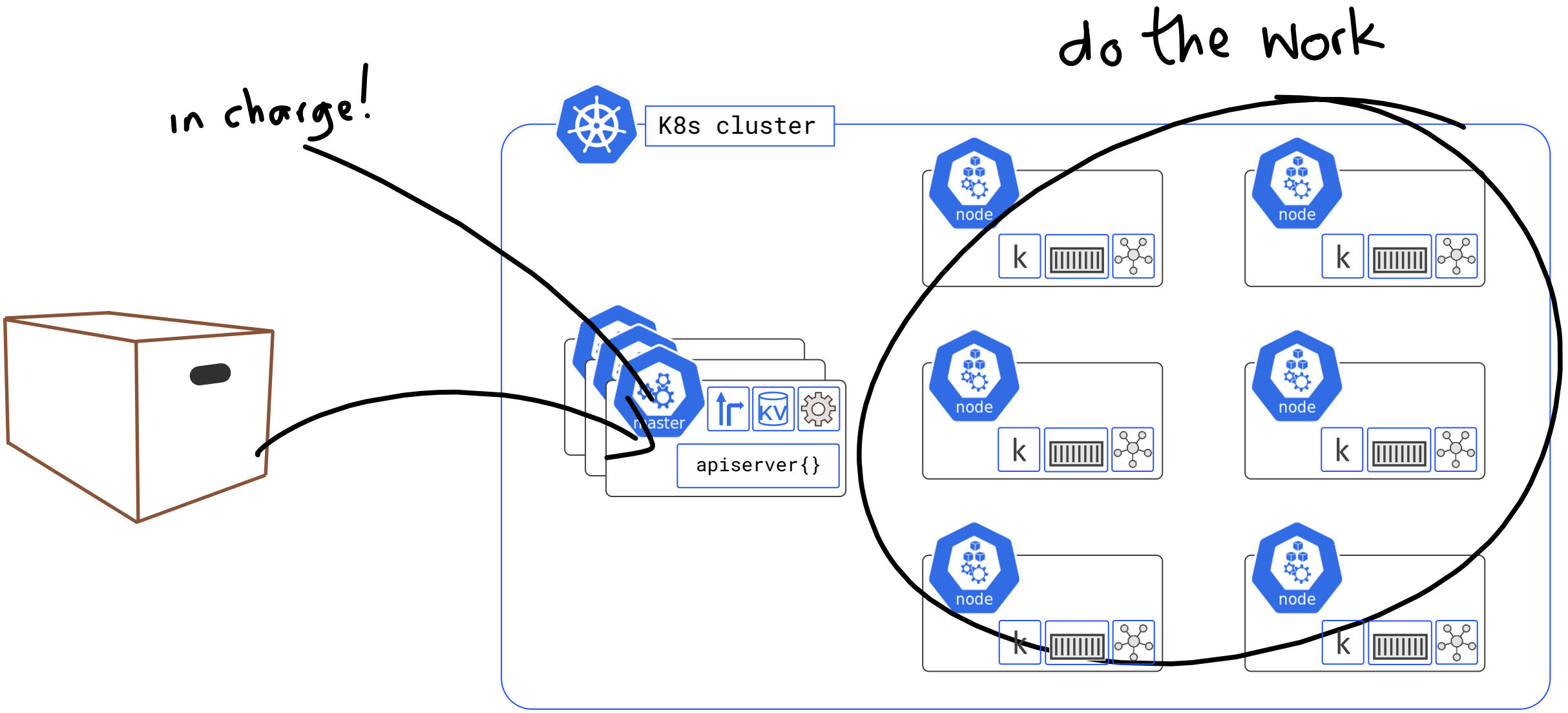
MySQL

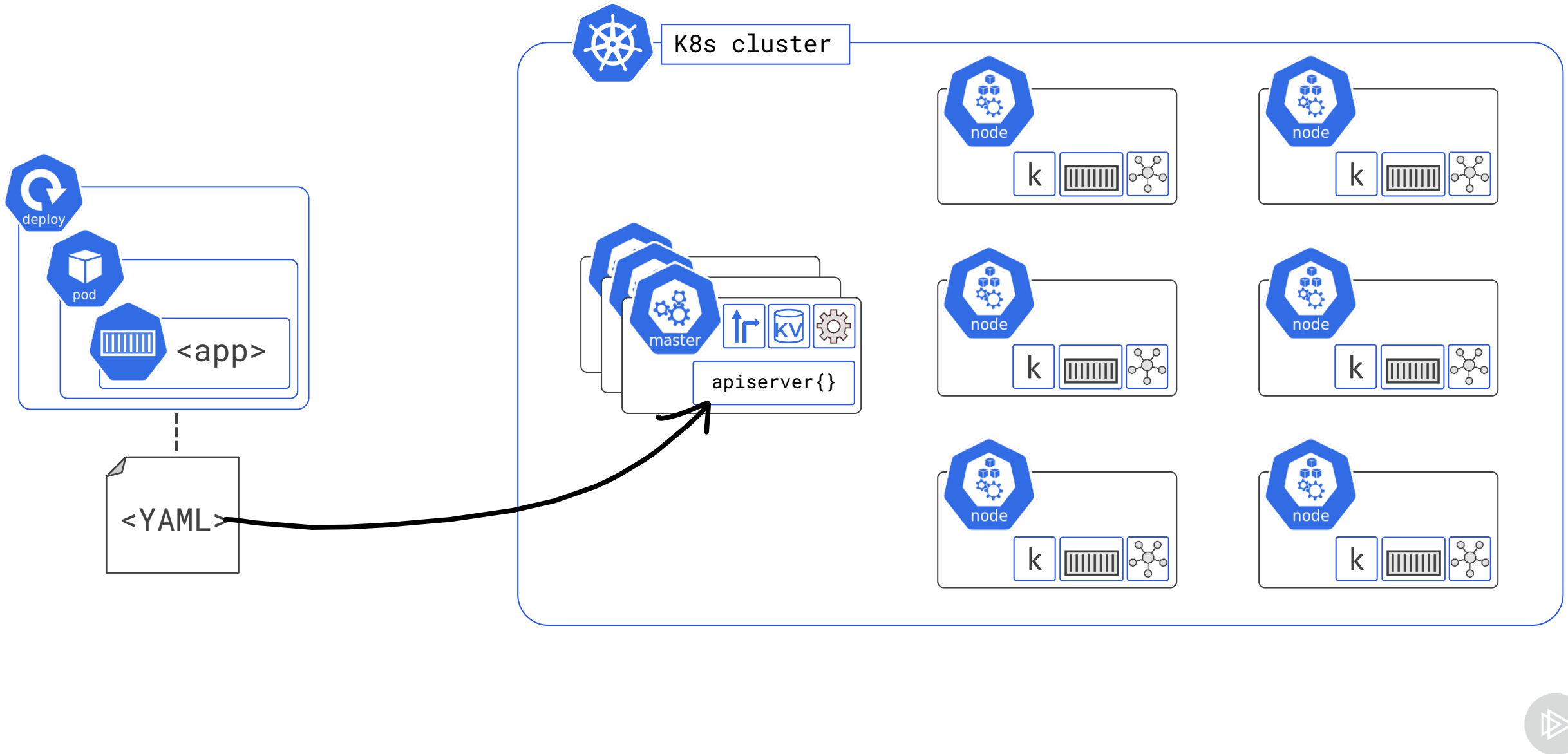

Node 1
Node 2
Node 3


Load
balancer
HTTPS
HTTPS
Auth
Search
Log
K/V store
MySQL


in charge!
do the work
K8s cluster
master
node
node
node
node
node
node


in charge!
do the work
K8s cluster
master
kv
apiserver{}
node
k


in charge!
do the work
K8s cluster
master
kv
apiserver{}
node
k

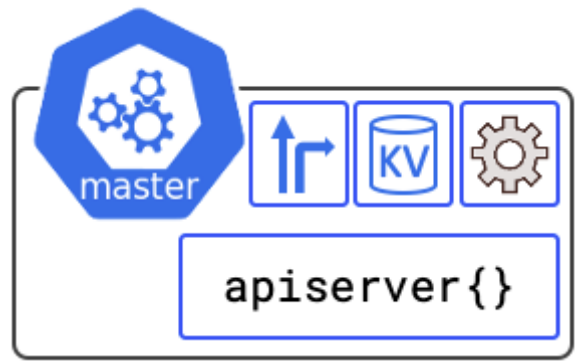

K8s cluster
deploy
pod
<app>
<YAML>
master
kv
apiserver{}
node
k
node
k
node
k
node
k
node
k
node
k

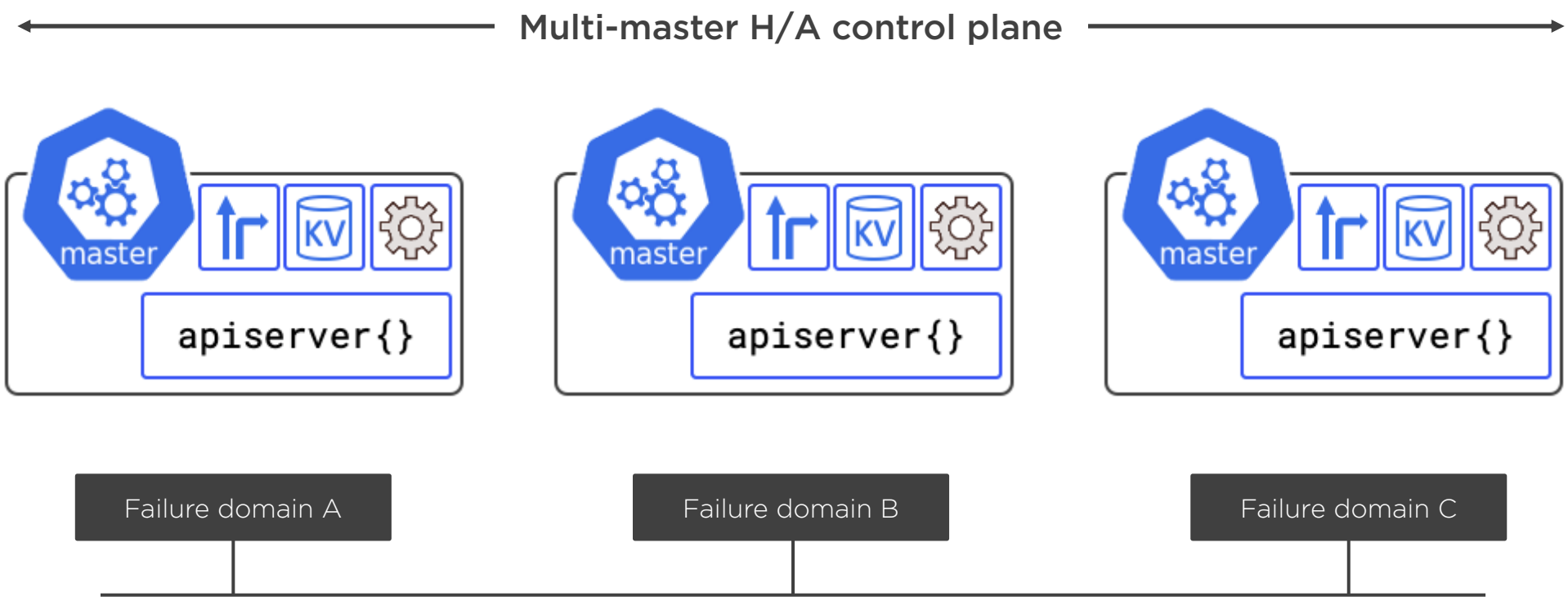
Up Next:
Masters

Masters

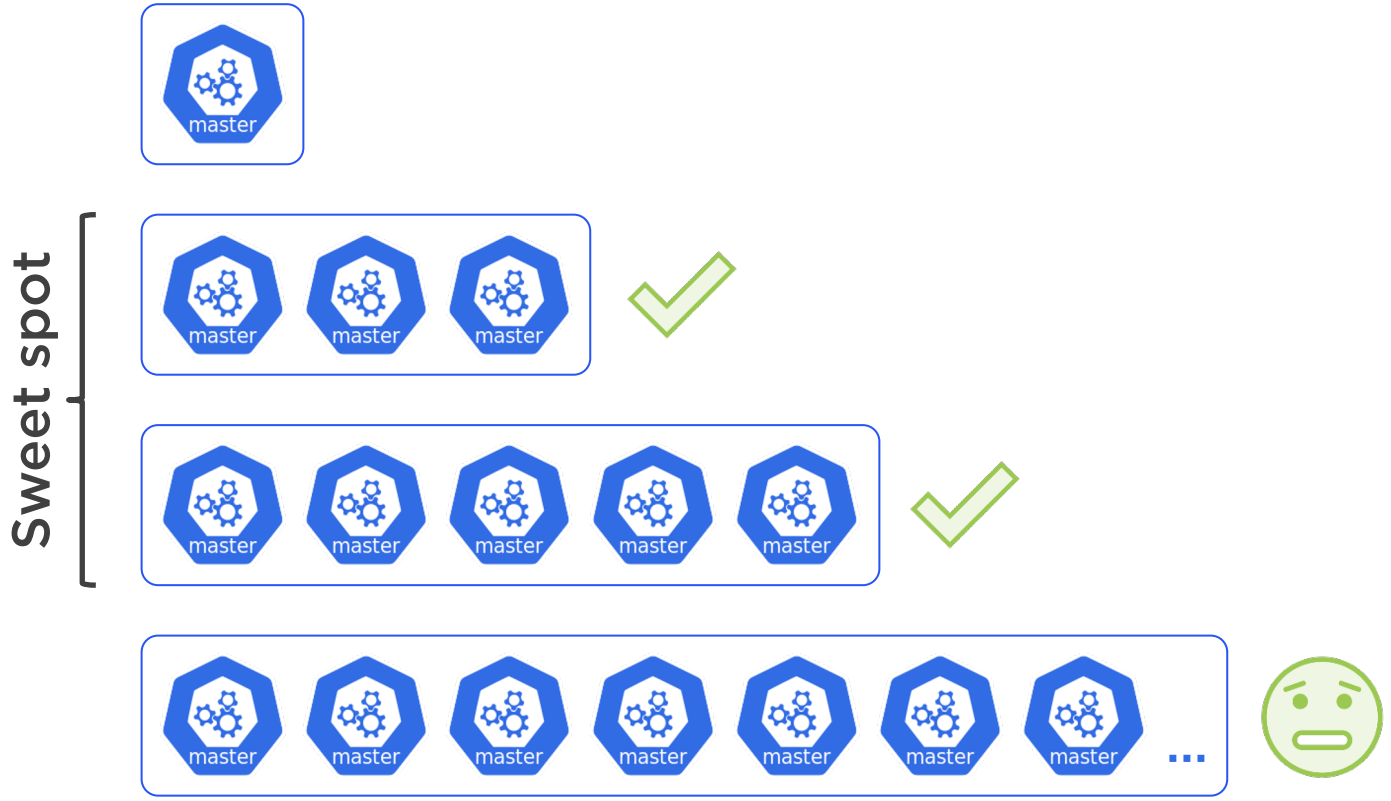
Masters

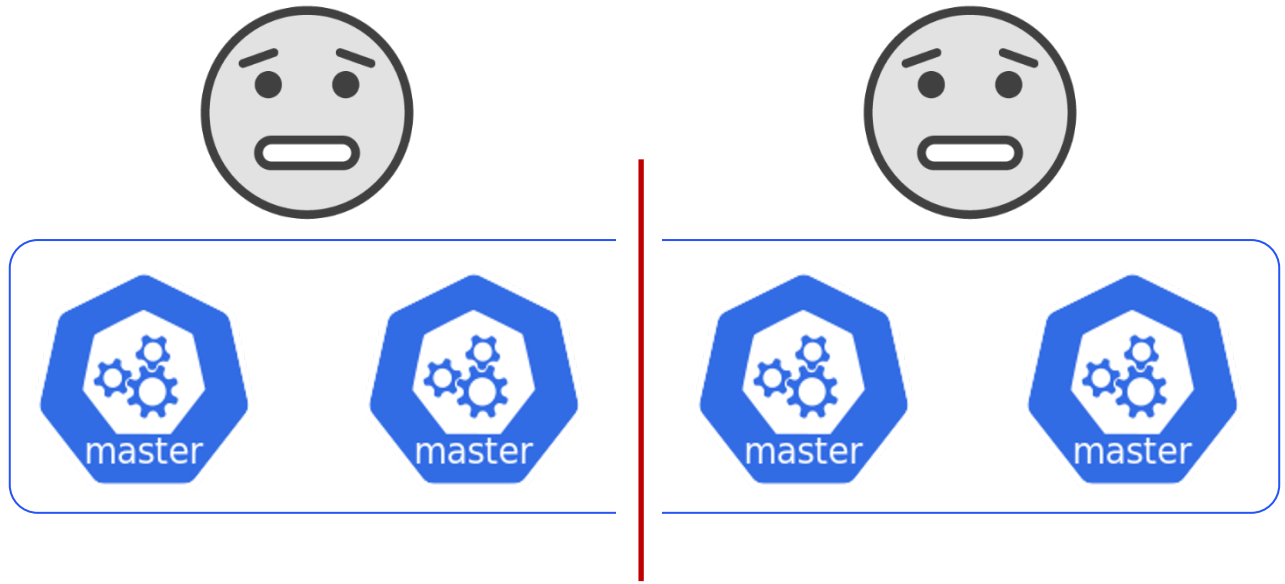
Also known as *head nodes* or the *control plane*

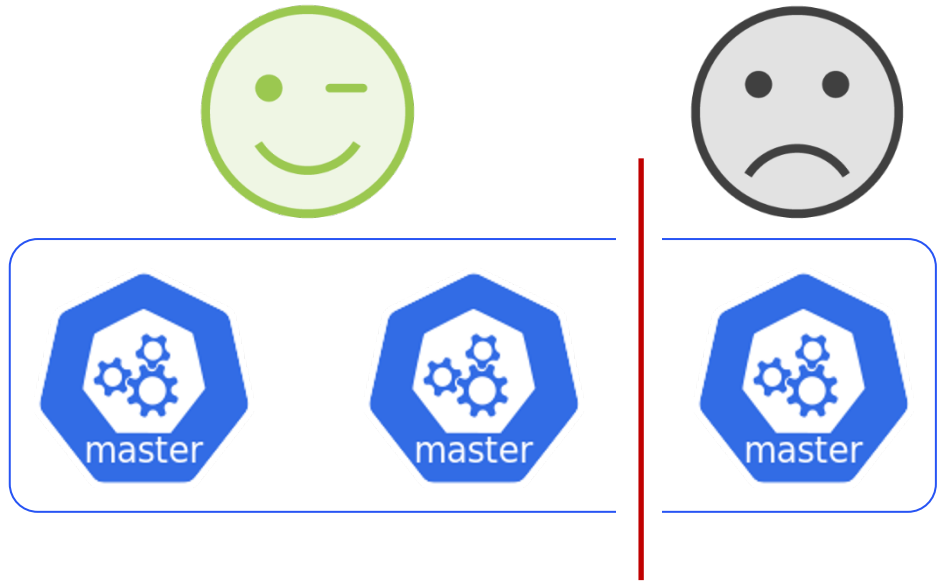

master
KV
apiserver{}

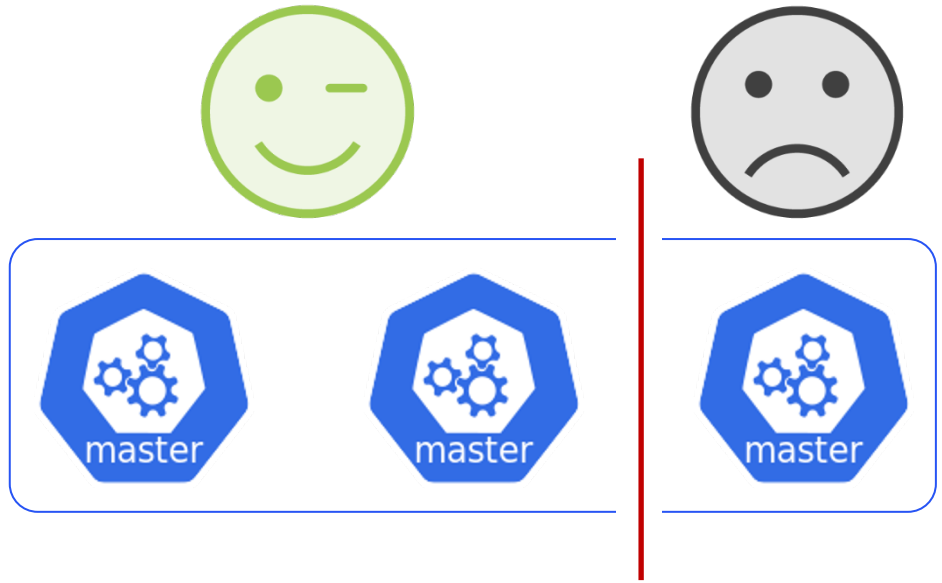

Multi-master H/A control plane
master
KV
apiserver{}
master
KV
apiserver{}
master
KV
apiserver{}
Failure domain A
Failure domain B
Failure domain C

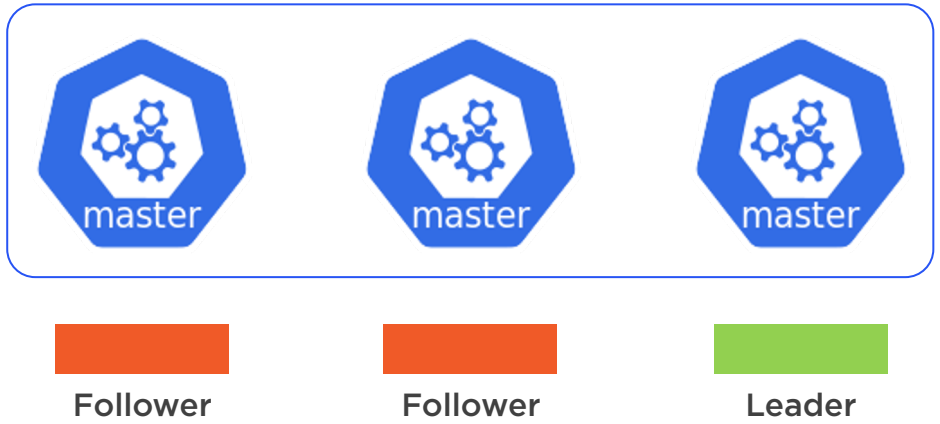
H/A Design

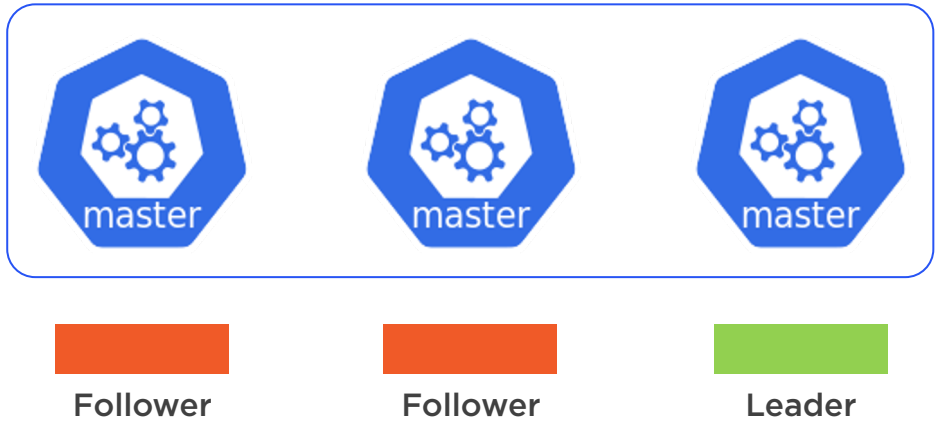

master
Sweet spot
master master master
master master master master master
master master master master master master master ...


master
master
master
master

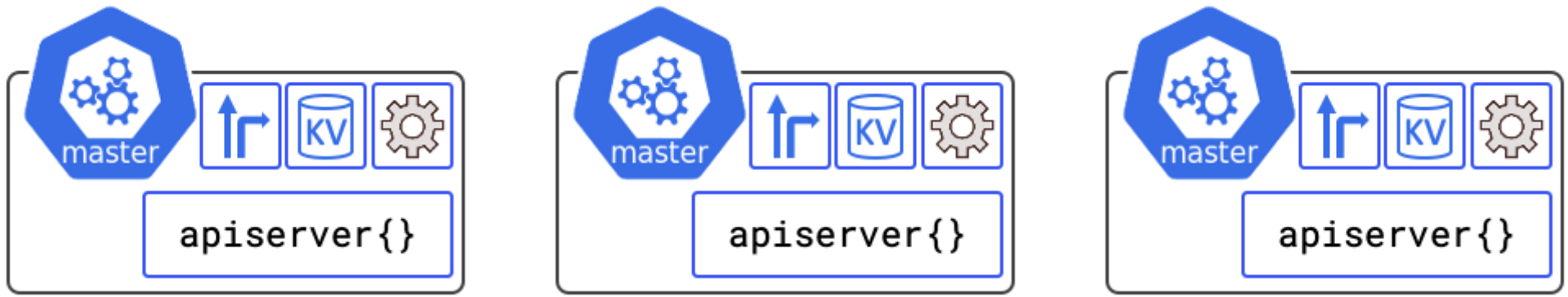

master
master
master

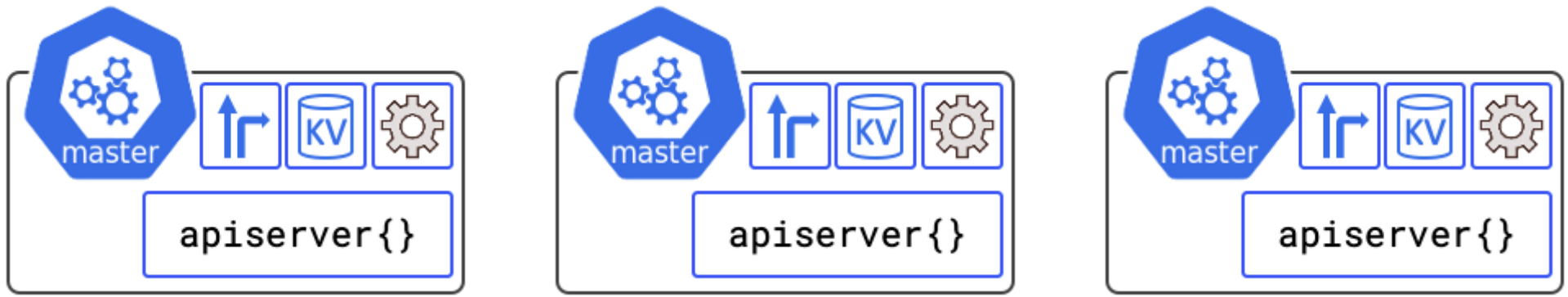

master
master
master

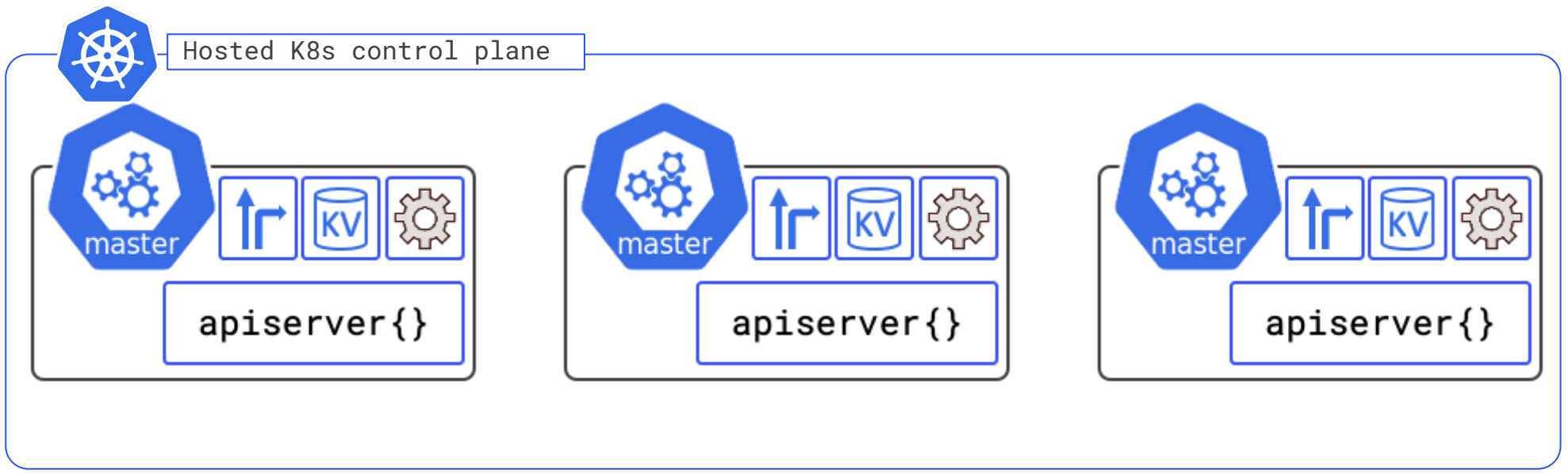

master
master
master
Follower
Follower
Leader


master
master
master
Follower
Follower
Leader


master
master
master


master
KV
apiserver{}
master
KV
apiserver{}
master
KV
apiserver{}


master
KV
apiserver{}
master
KV
apiserver{}
master
KV
apiserver{}


Hosted K8s control plane
master
KV
apiserver{}
master
KV
apiserver{}
master
KV
apiserver{}

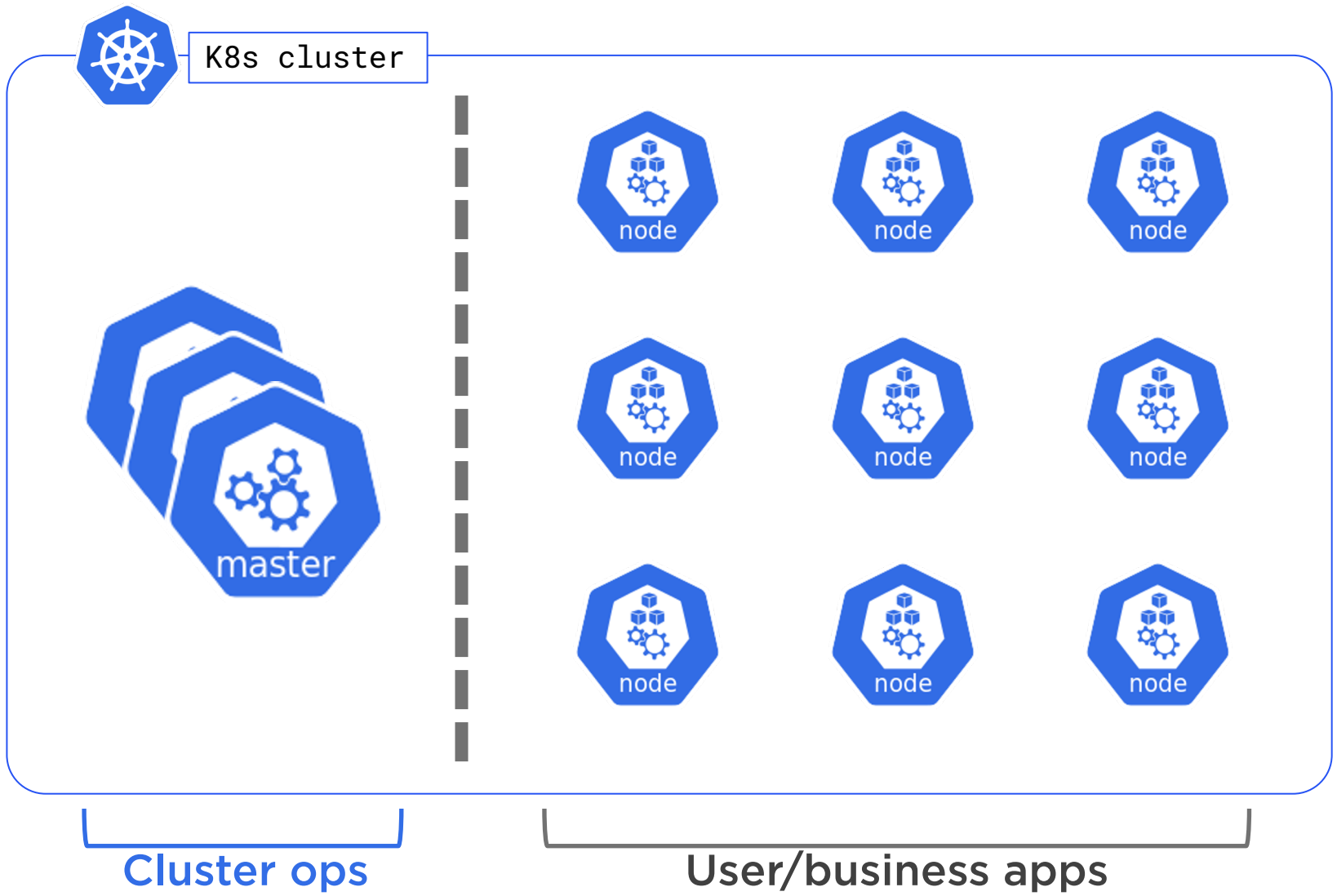
Hosted K8s control plane

- Performance
- Availability
- Updates/Upgrades
- API Server Flags
- ...

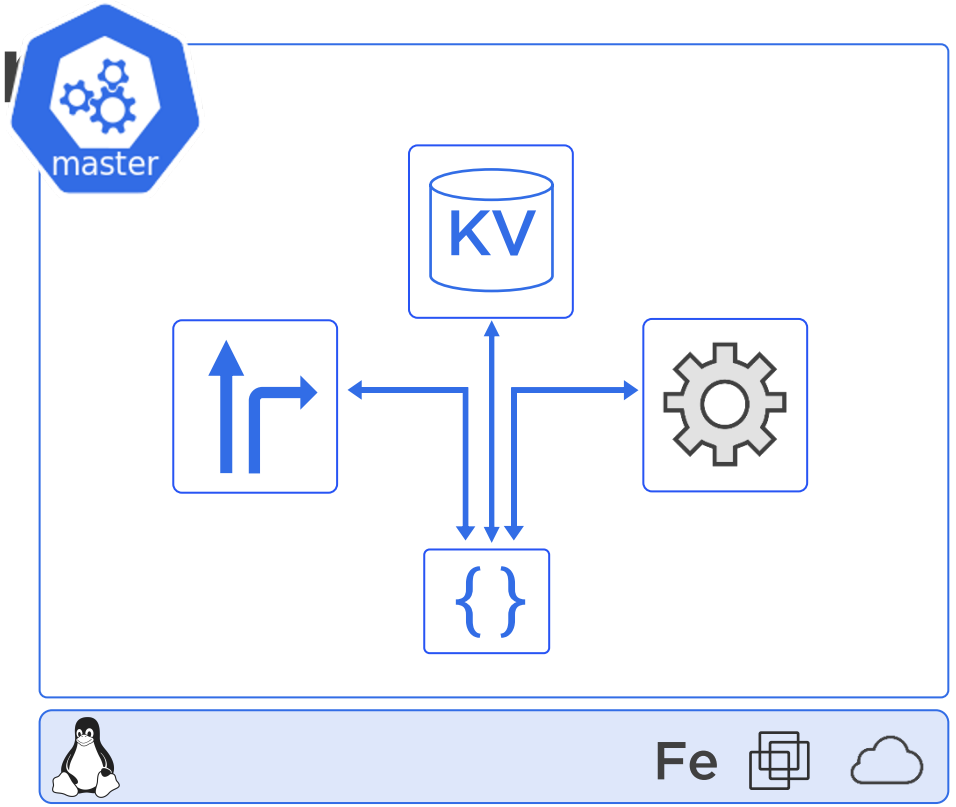
Hidden

Exposed

Highly available API Endpoint

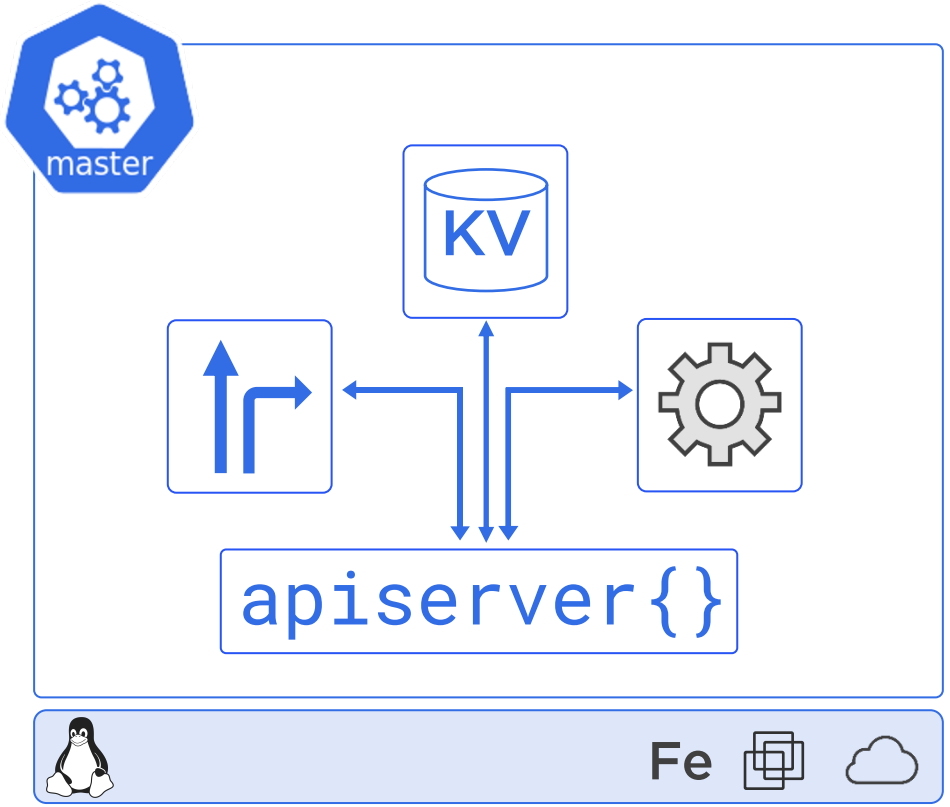

K8s cluster
master
node
node
node
node
node
node
node
node
node
Cluster ops
User/business apps

kube-apiserver



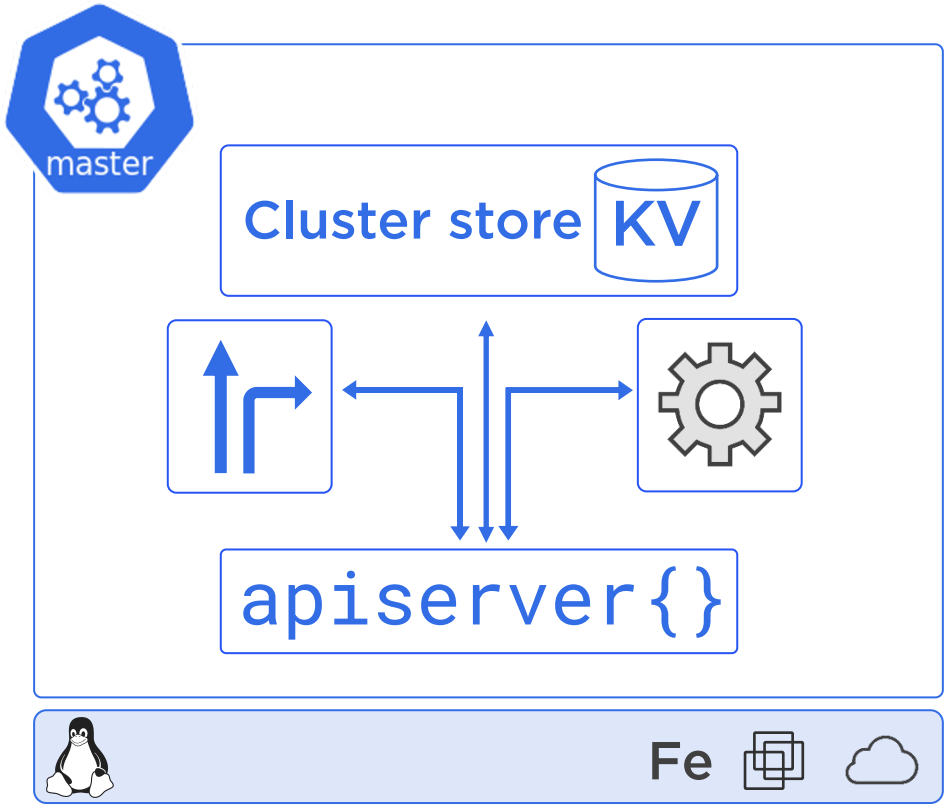
kube-apiserver

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



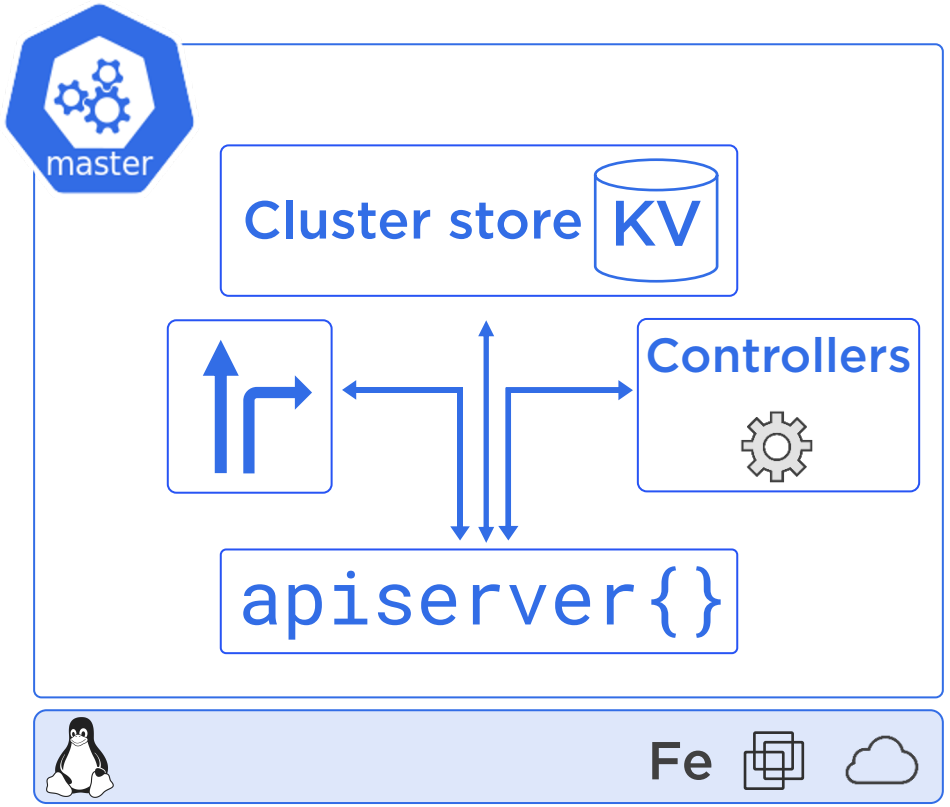
Cluster Store

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



Kube-controller-manager

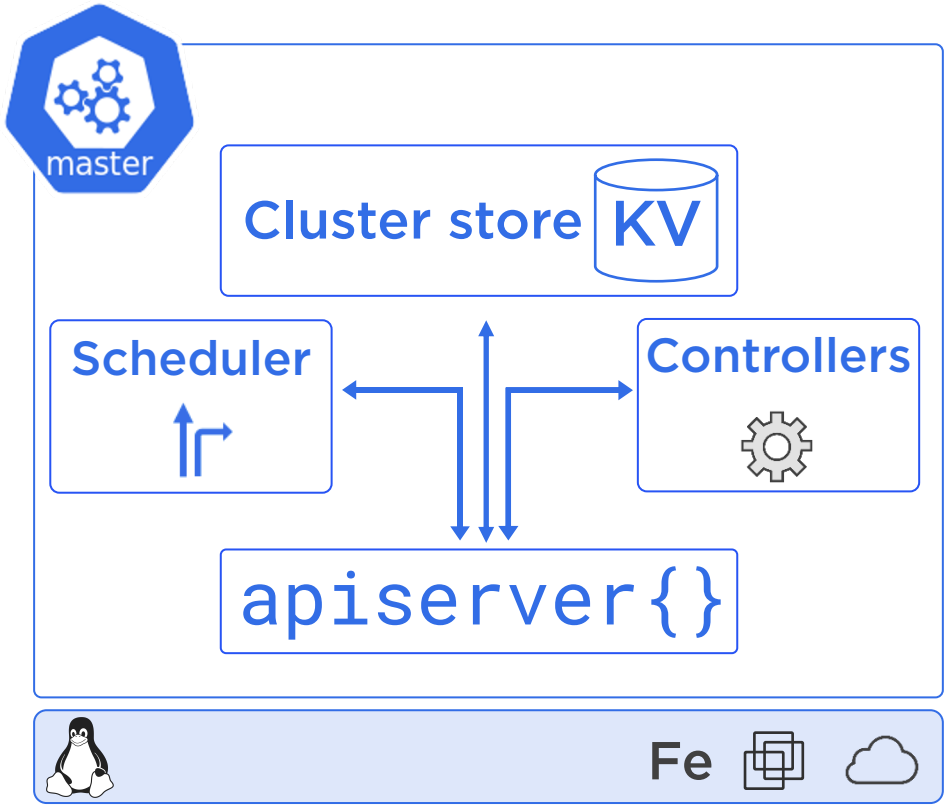
- Controller of controllers
  - Node controller
  - Deployment controller
  - Endpoints/EndpointSlice controller...
- Watch loops
- Reconciles observed state with desired state

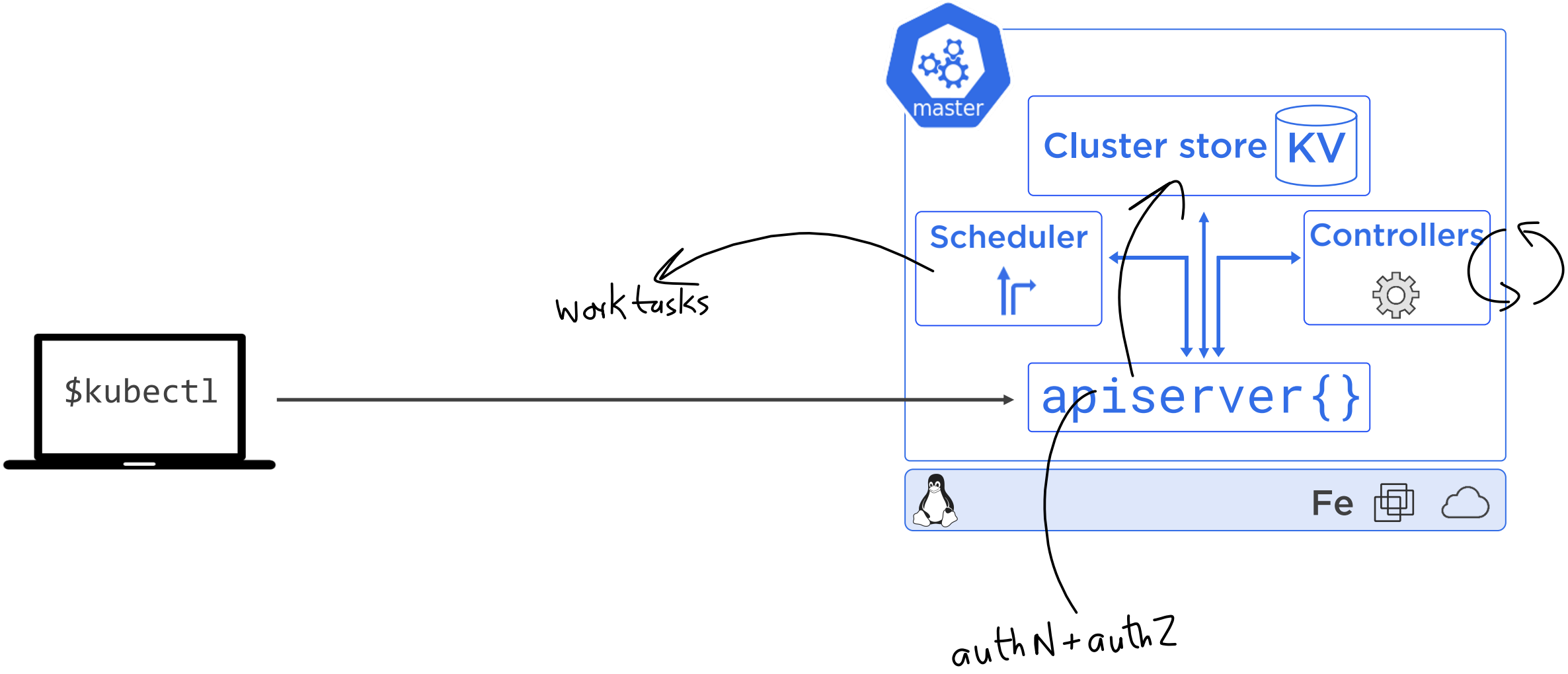


Kube-scheduler

- Watches API Server for new work tasks

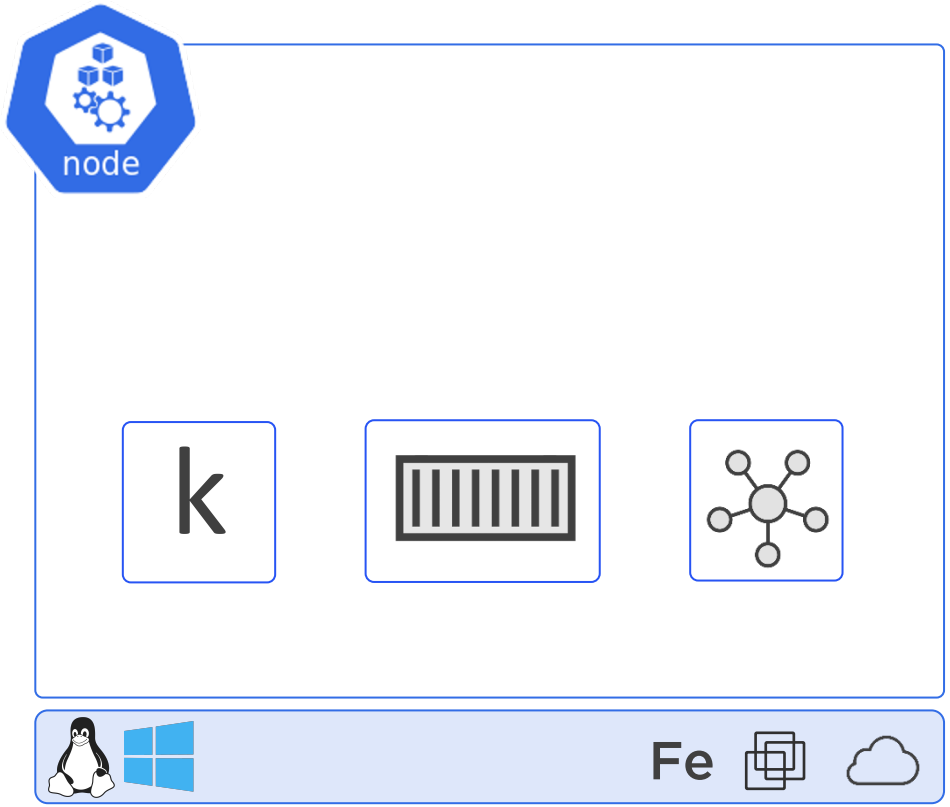
- Assigns work to cluster nodes
  - Affinity/Anti-affinity
  - Constraints
  - Taints
  - Resources...




master
Cluster store
KV
Scheduler
Controllers
work tasks
$kubectl
apiserver{}
Fe
authN+authZ

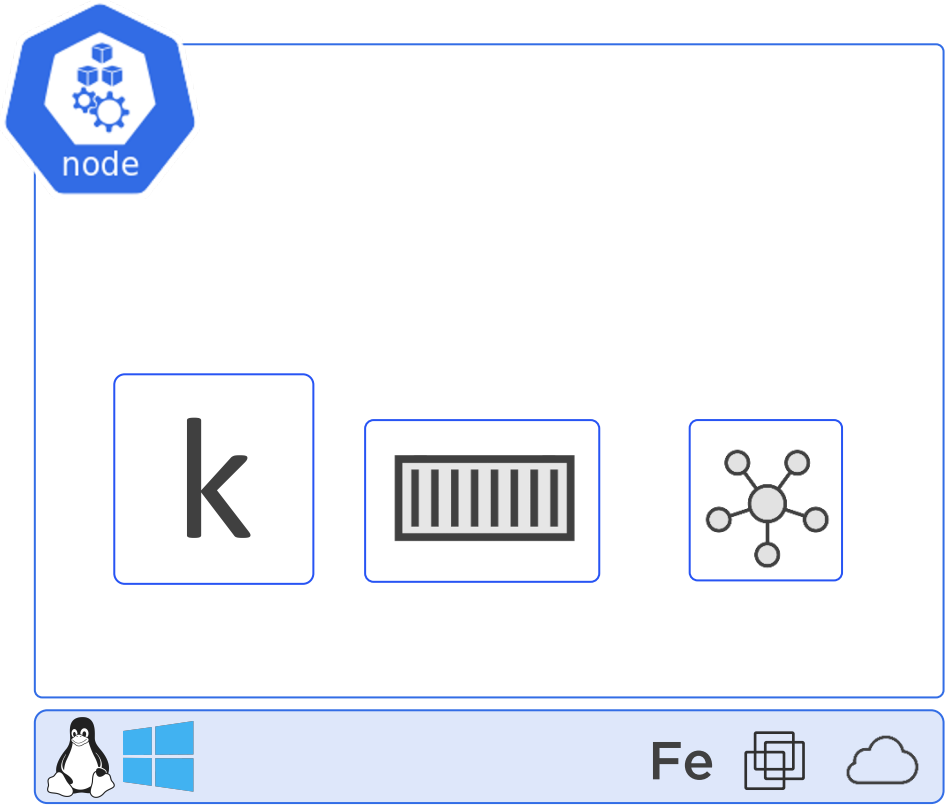
Up Next:
Nodes

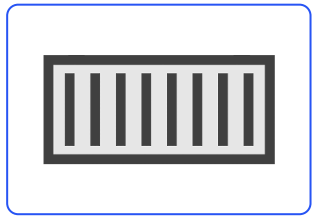
Nodes


node
k
Fe

Kubelet

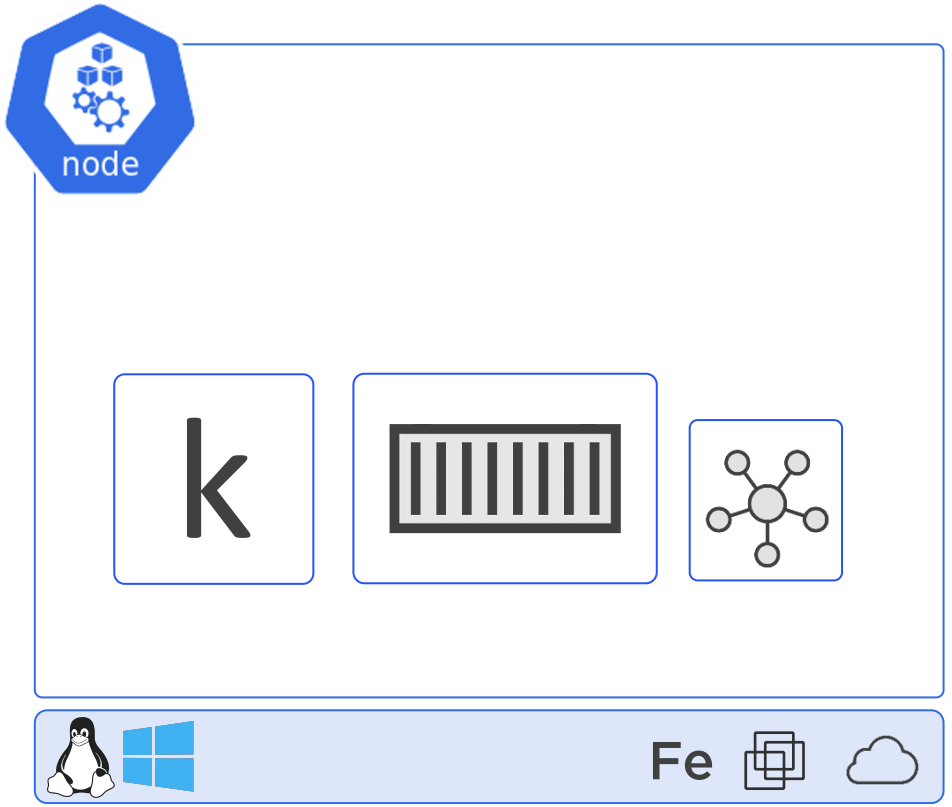
- Main Kubernetes agent
- Registers node with cluster
- Watches API Server for work tasks (Pods)
- Executes Pods
- Reports back to Masters





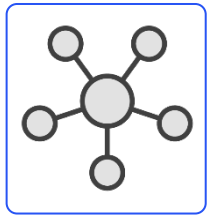
Container runtime

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



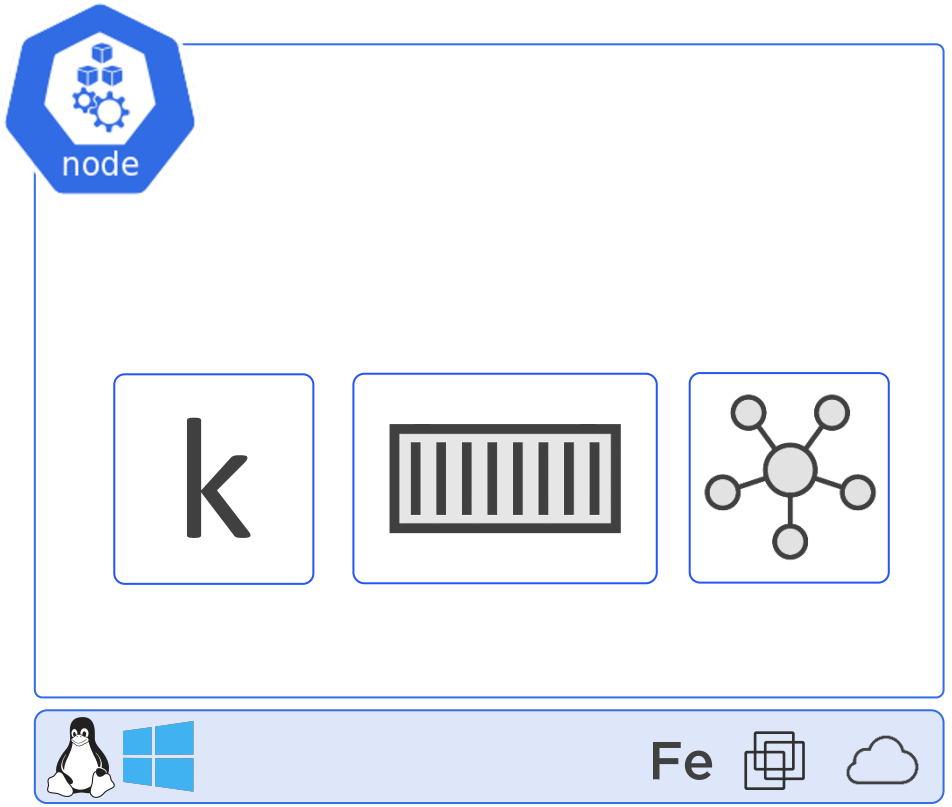
gVisor

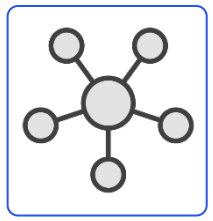
katacontainers



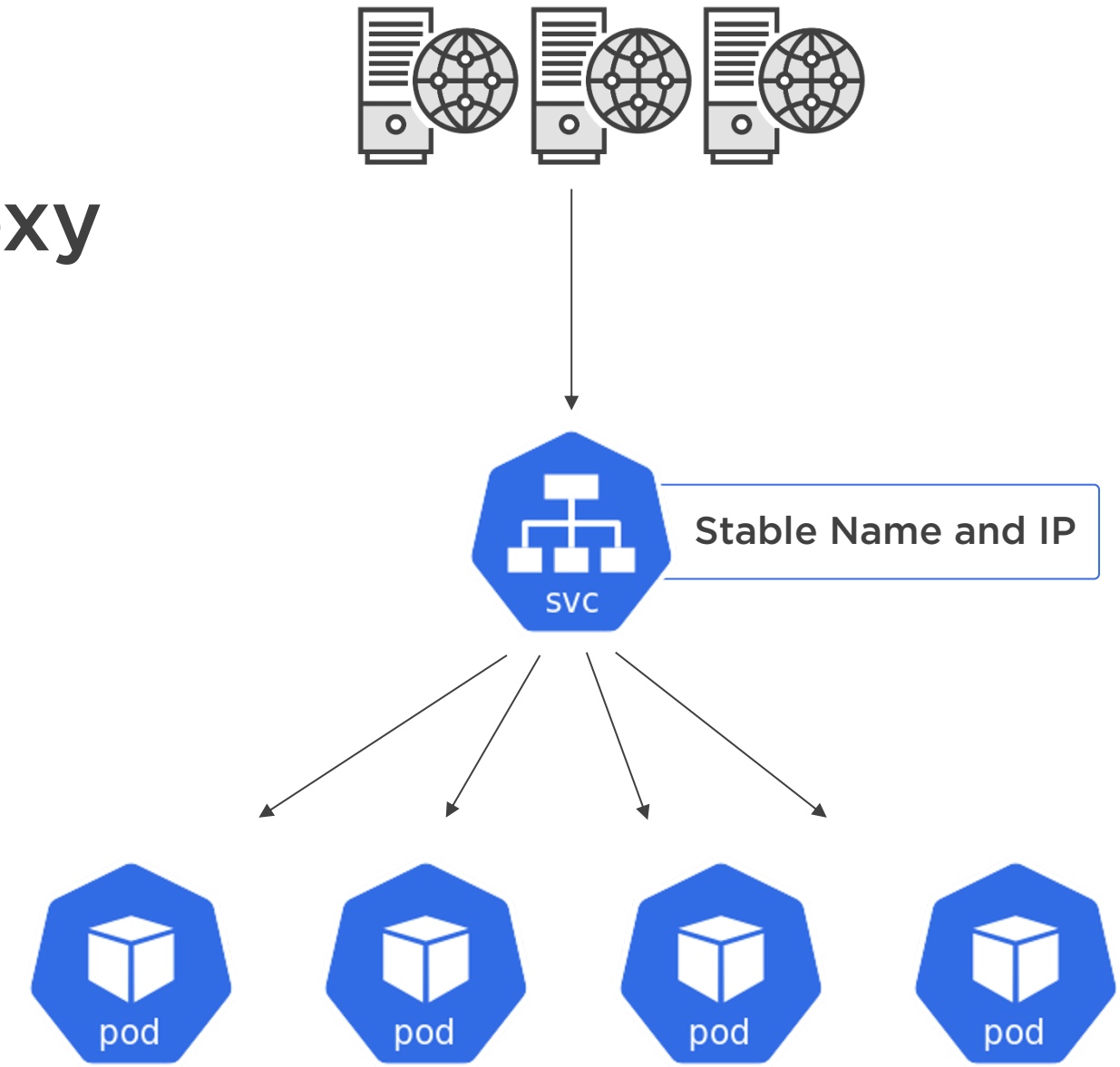
Kube-proxy

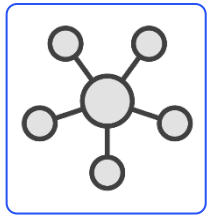
- Networking component
- Pod IP addresses





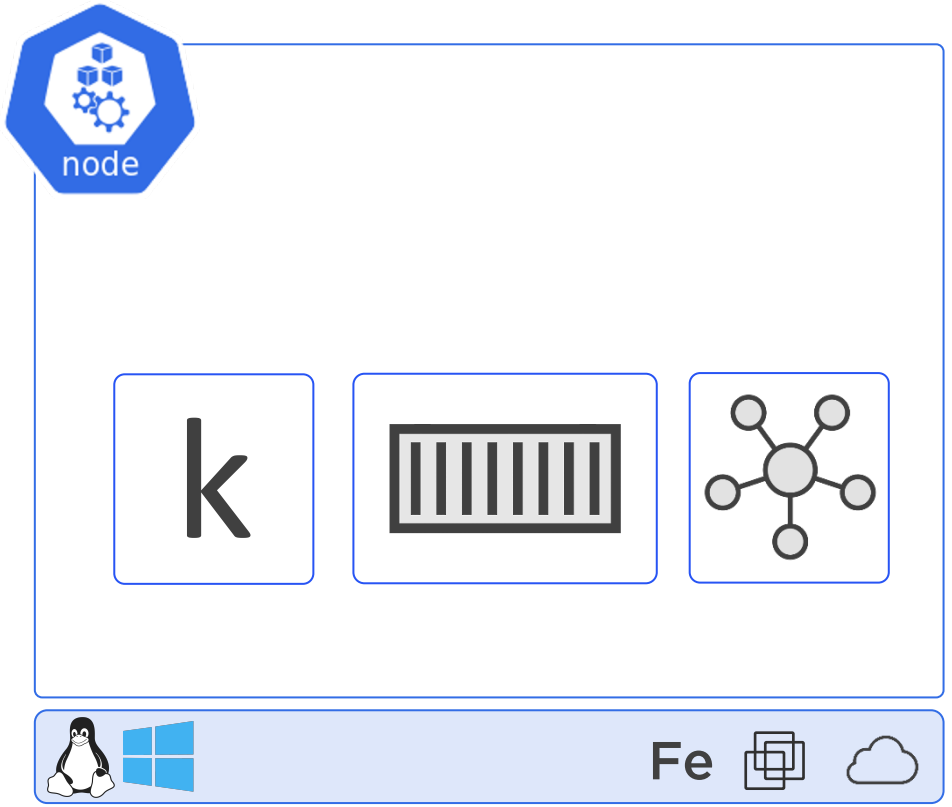
Kube-proxy



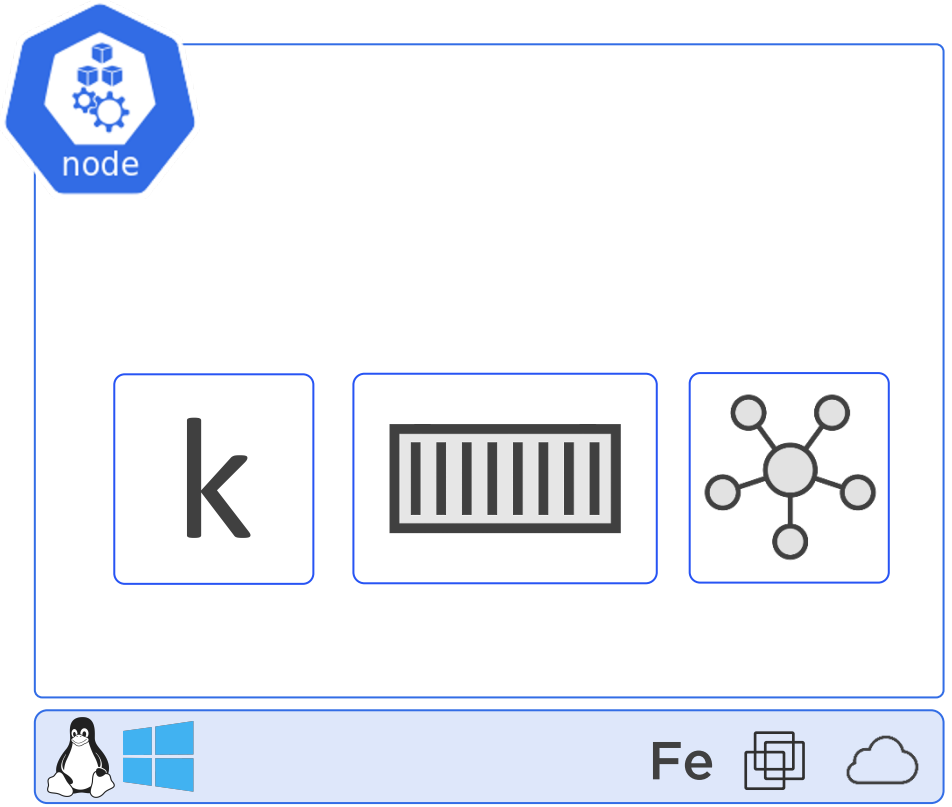


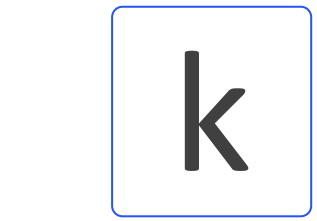
Kube-proxy

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



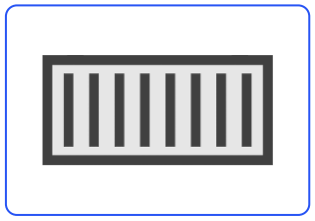

k


node
k
Fe



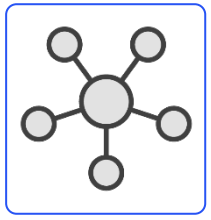
Kubelet

Main K8s agent



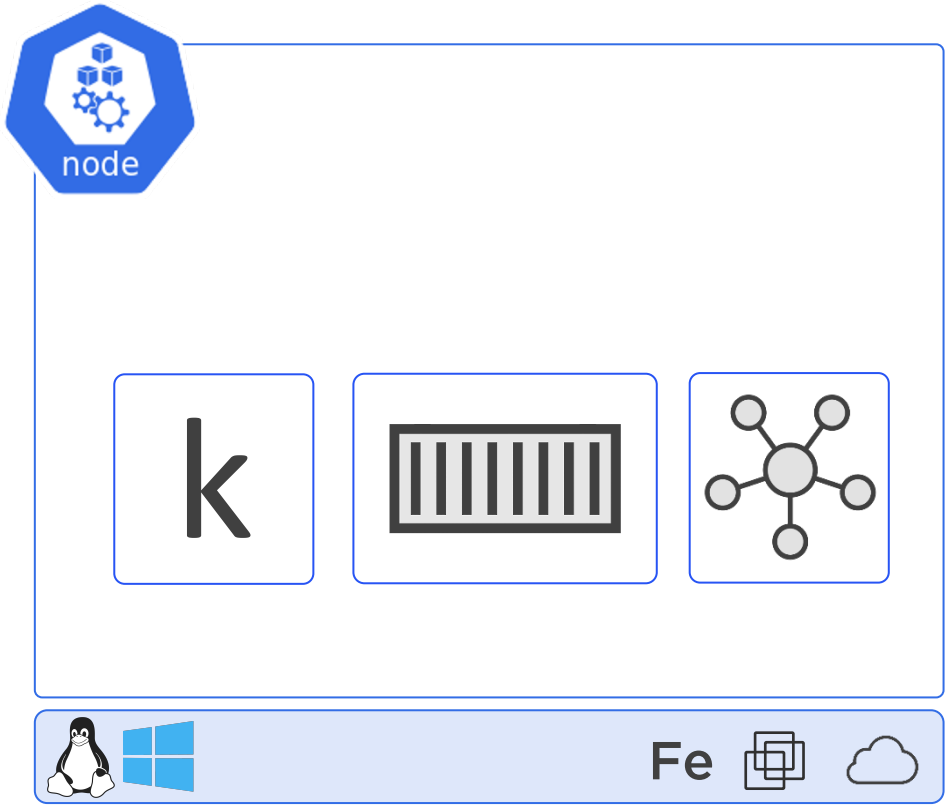
Container runtime

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



Kube-proxy

Vital role in networking



Virtual Kubelet

Nodeless Kubernetes

Virtual Kubelet

Nodeless Kubernetes

Virtual Kubelet

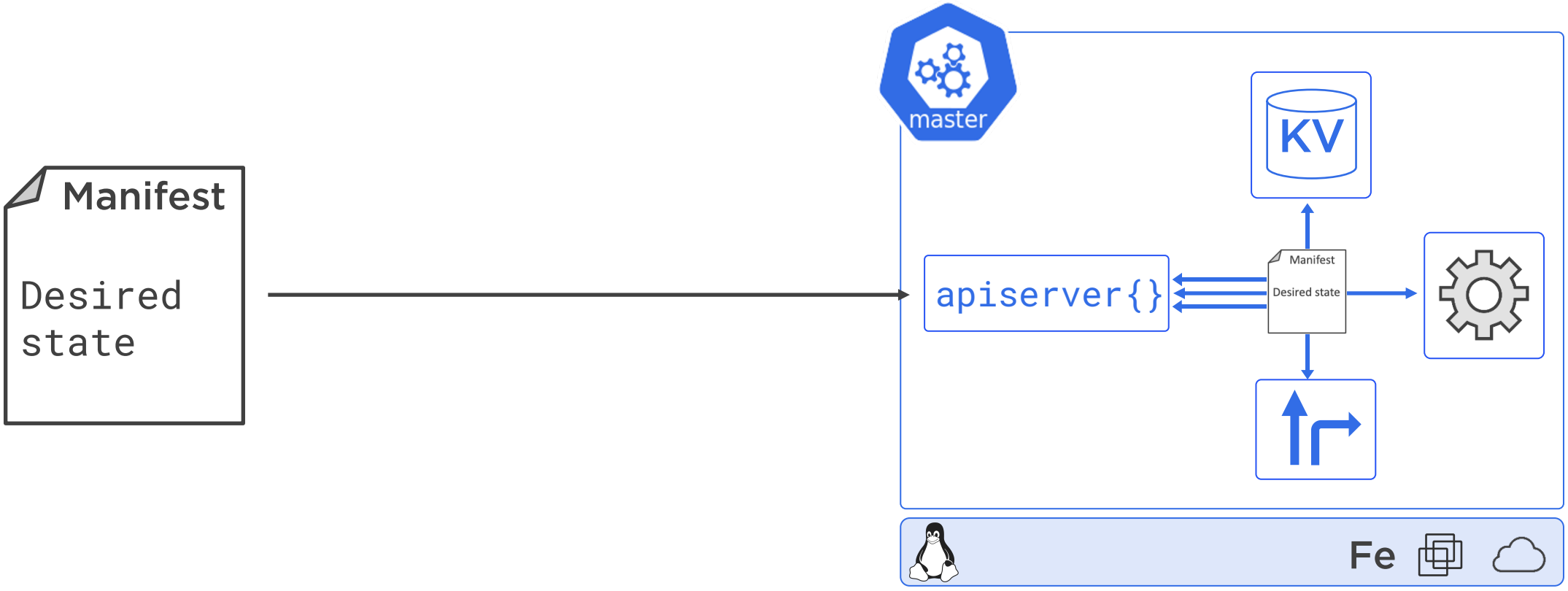
Pods run on cloud's hosted container back-end

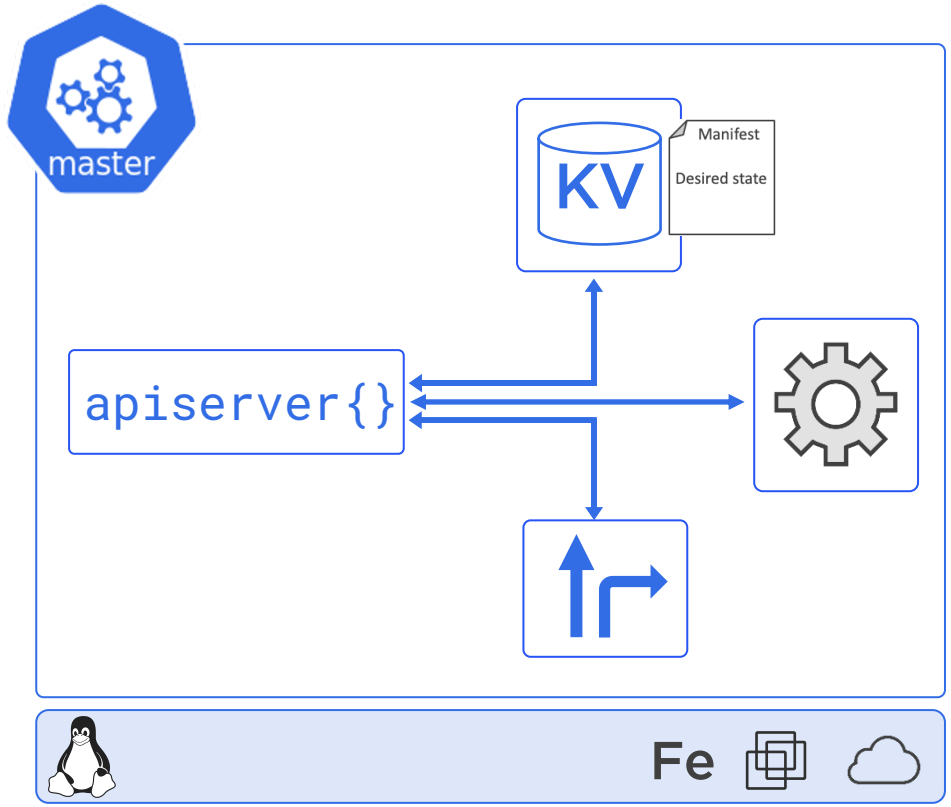
Up Next:
The Declarative Model & Desired State

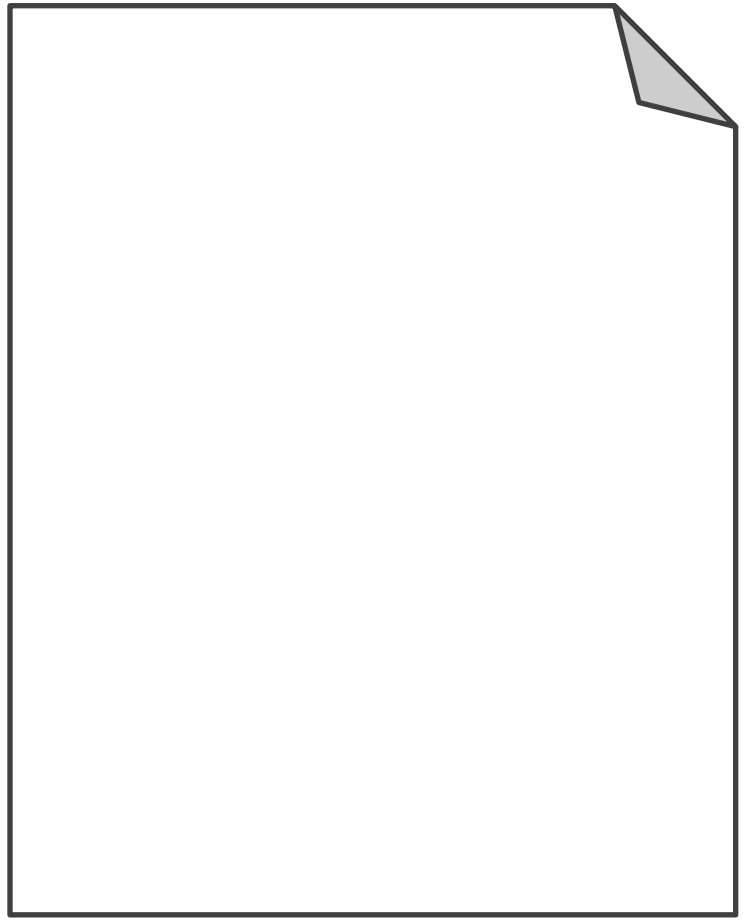
The Declarative Model and Desired State

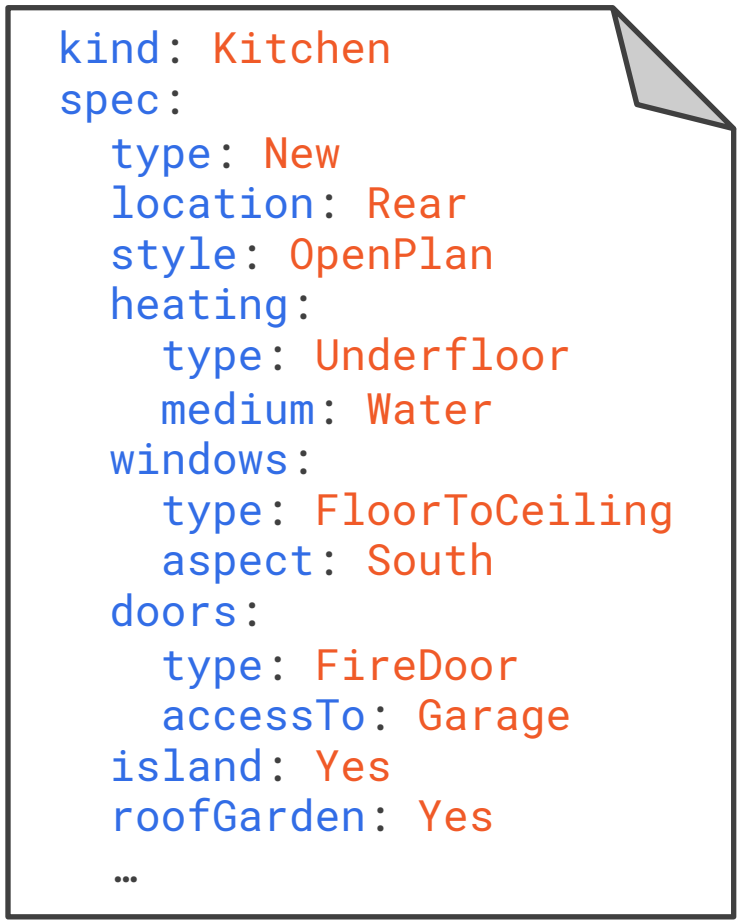
Declarative model

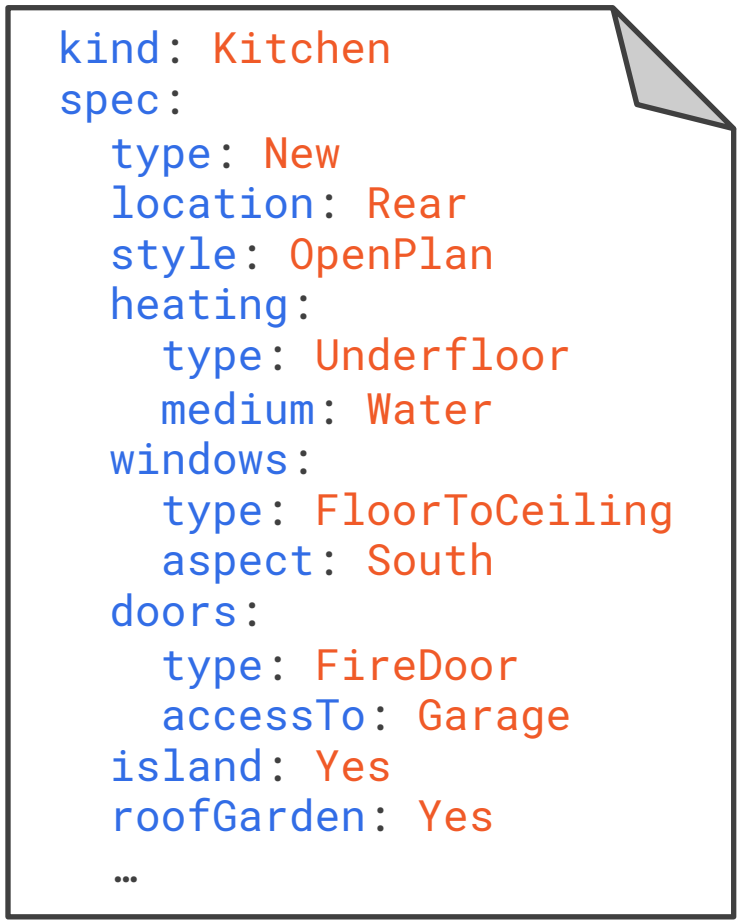
Describe what you *want (desired state)* in a manifest file

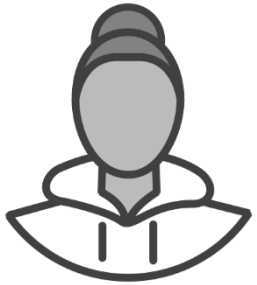

Manifest
Desired state
master
KV
apiserver{}
Manifest
Desired state
Fe


master
KV
Manifest
Desired state
apiserver{}
Fe




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
…

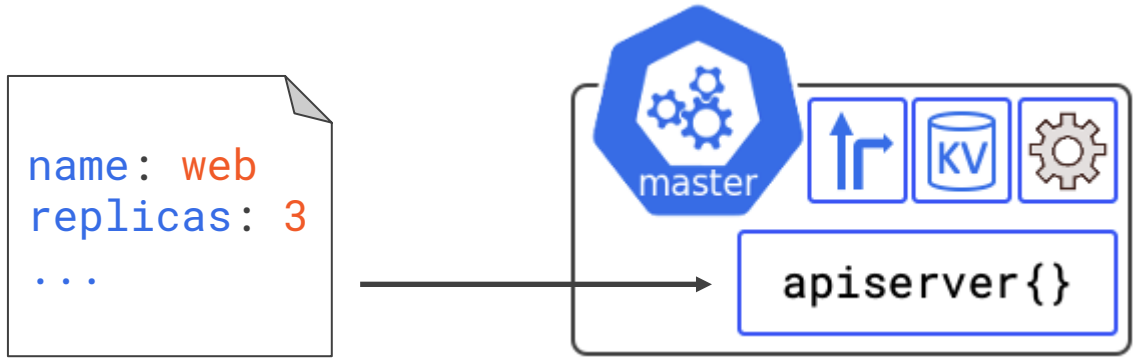

kind: Kitchen
spec:
type: New
location: Rear
style: OpenPlan
heating:
type: Underfloor
medium: Water
windows:
type: FloorToCeiling
aspect: South
doors:
type: FireDoor
accessTo: Garage
island: Yes
roofGarden: Yes
…



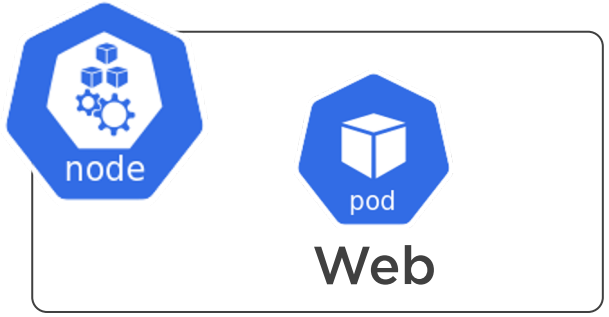
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: test
spec:
  replicas: 6
  selector:
    matchLabels:
      app: ps-test
  template:
    spec:
      containers:
        - name: c1
          image: web1:1.3
          ports: 8080
          - containerPort
            ...
```

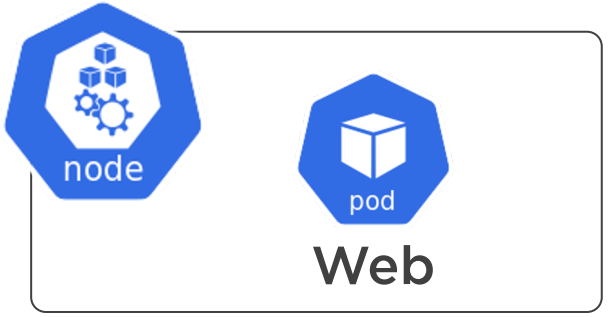
Declarative

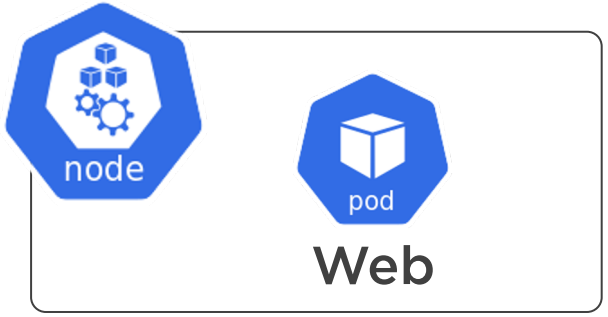
(Declaring what you want)

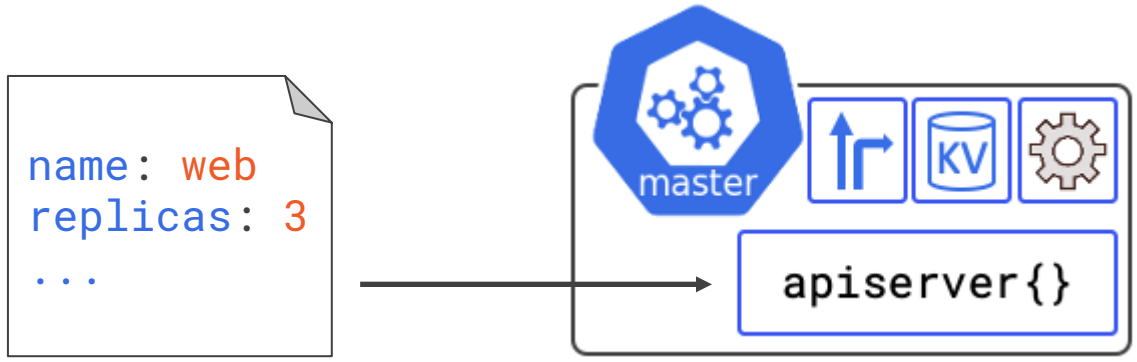


Desired state: 3
Observed state: 3

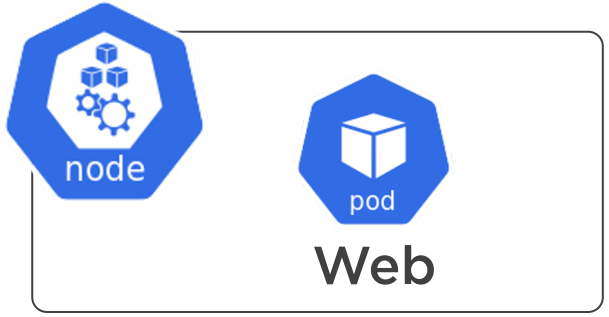


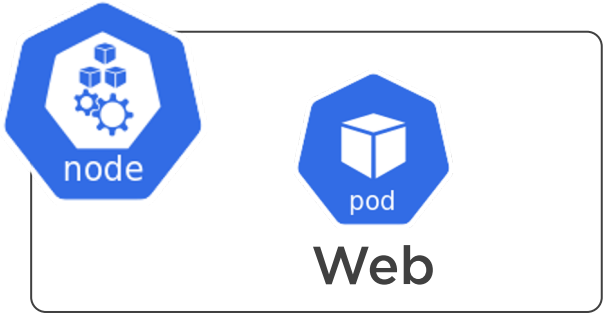


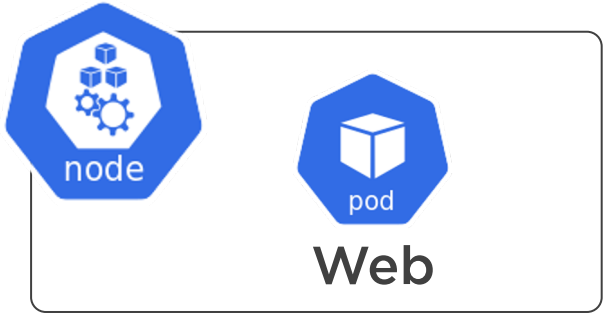


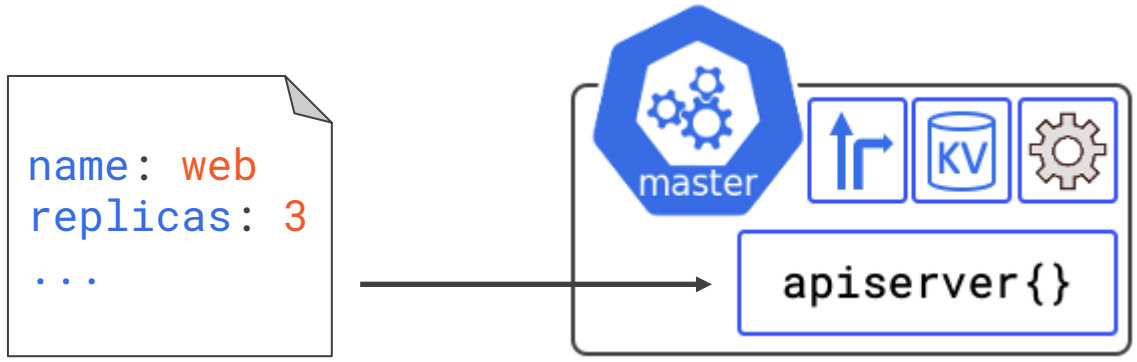


Desired state: 3
Observed state: 3

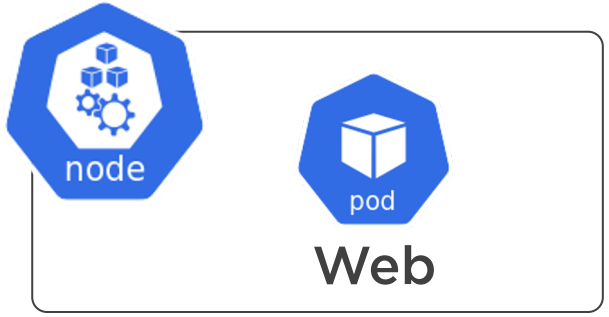


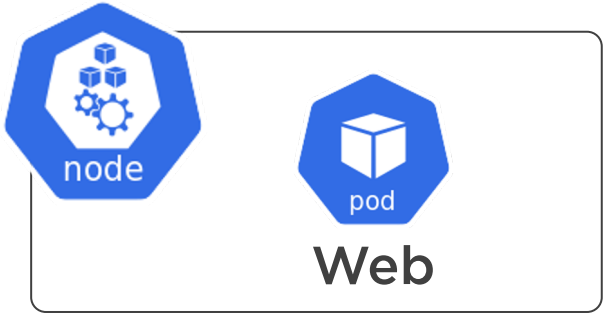


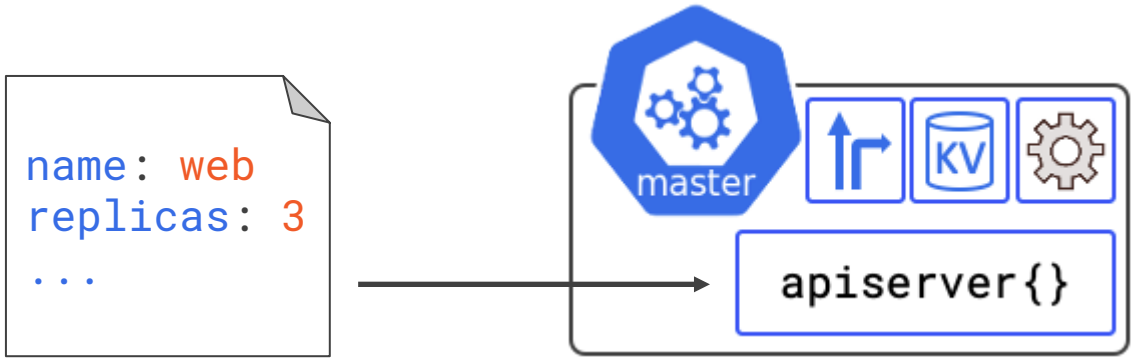




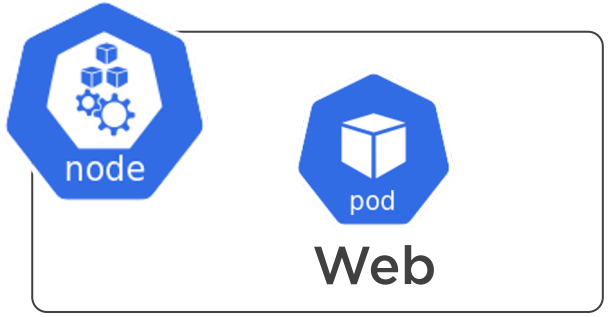
Desired state: 3
Observed state: 3

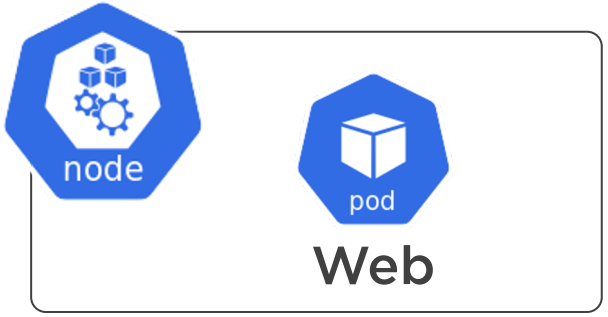


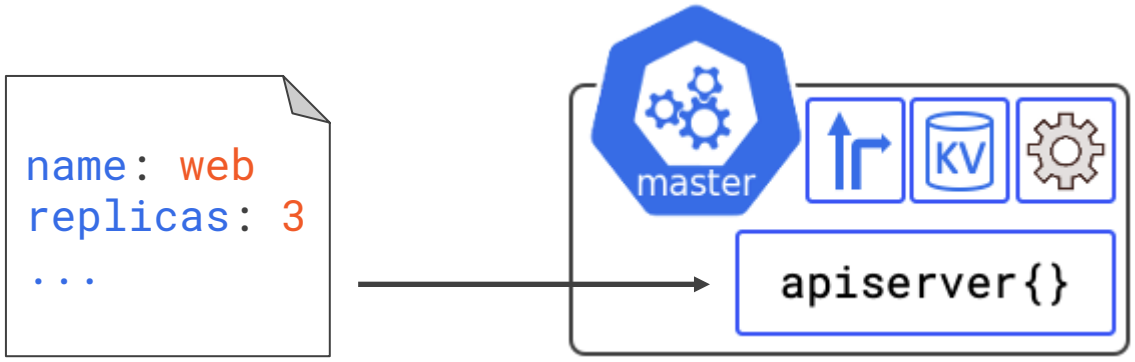




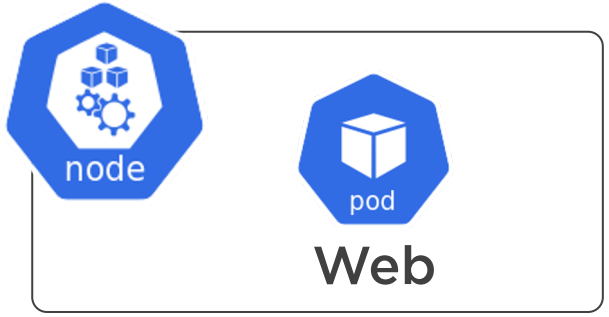
Desired state: 3
Observed state: 2

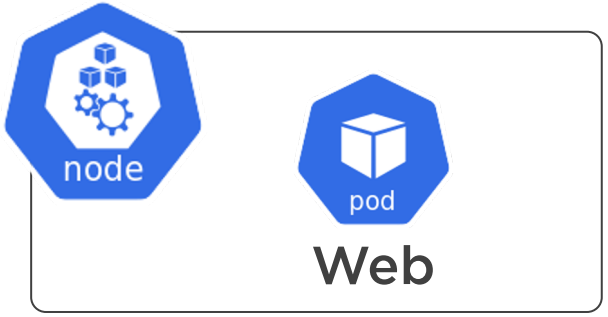


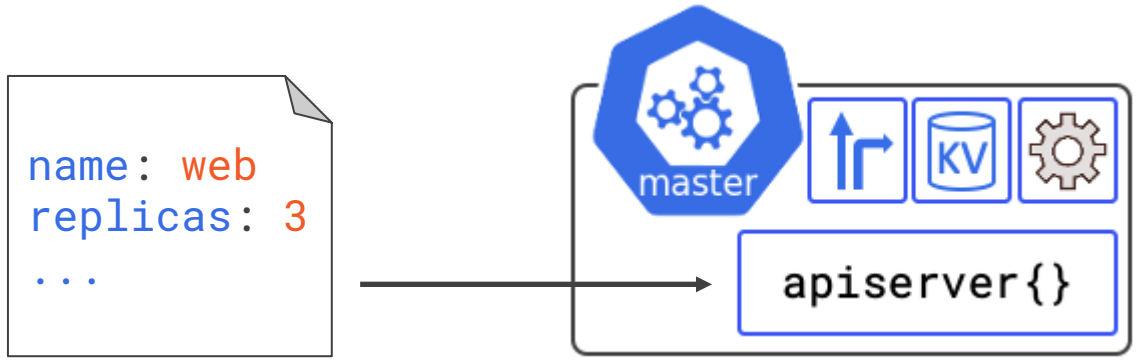




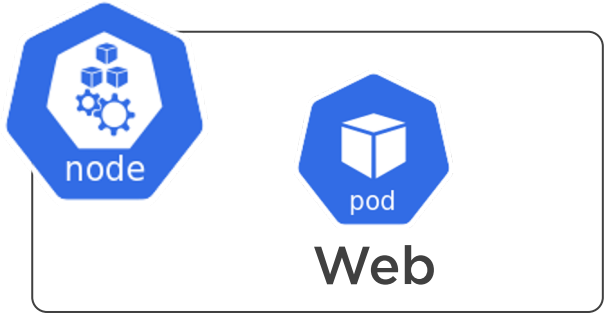
Desired state: 3
Observed state: 2

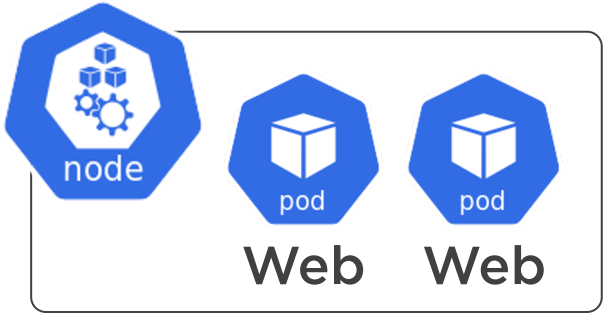


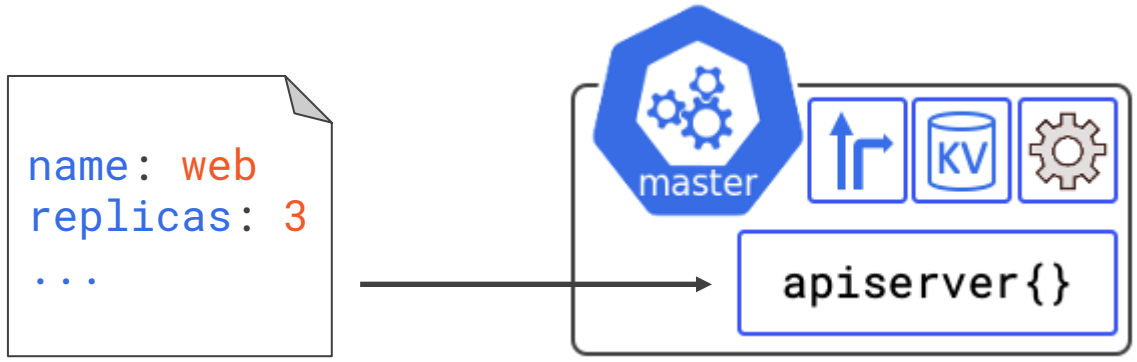




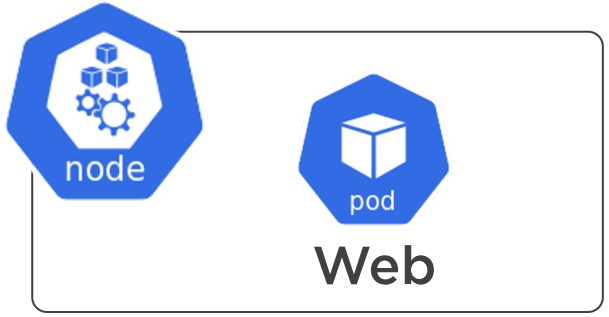
Desired state: 3
Observed state: 2

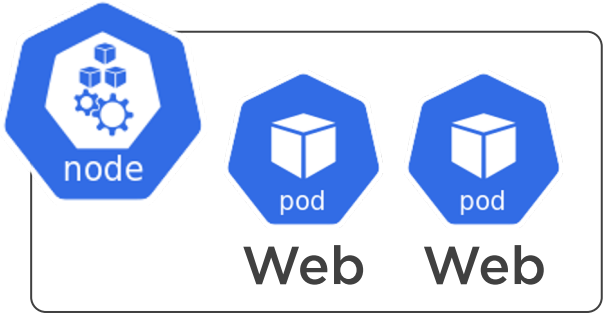


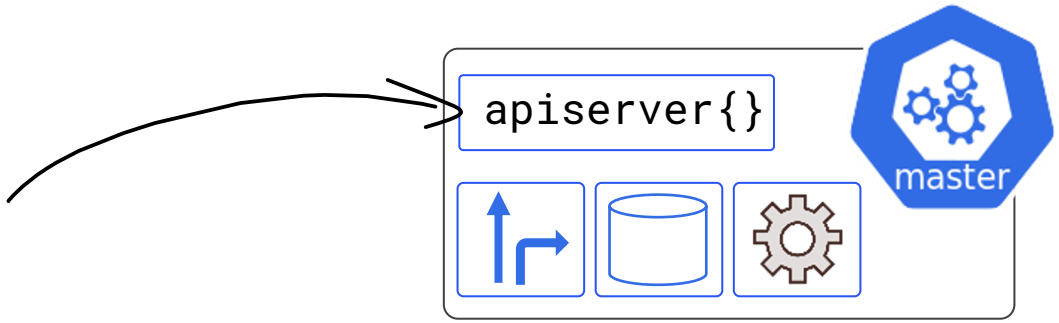




Desired state: 3
Observed state: 3



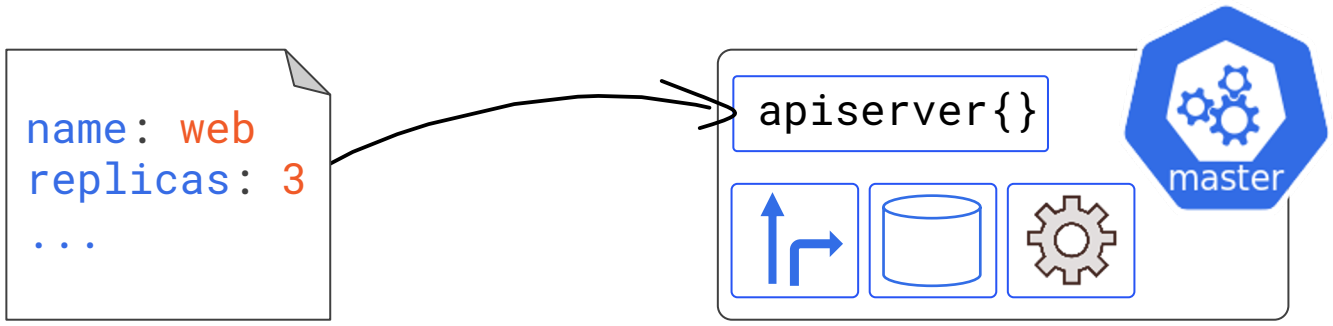



apiserver{}
master

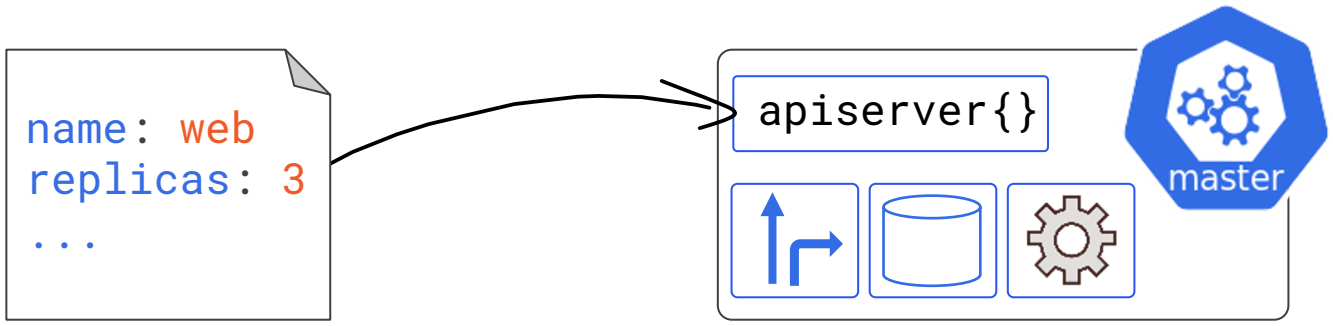

node


node

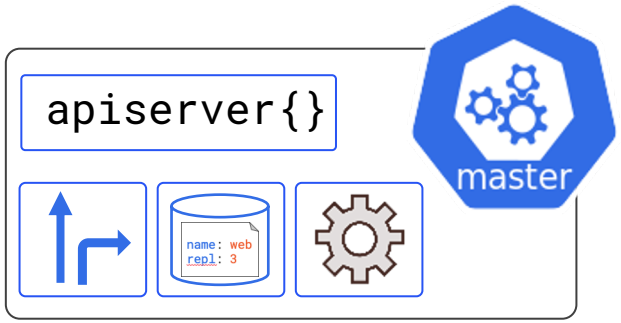

node


name: web
replicas: 3
...
apiserver{}
master


node
node
node


name: web
replicas: 3
...
apiserver{}
master

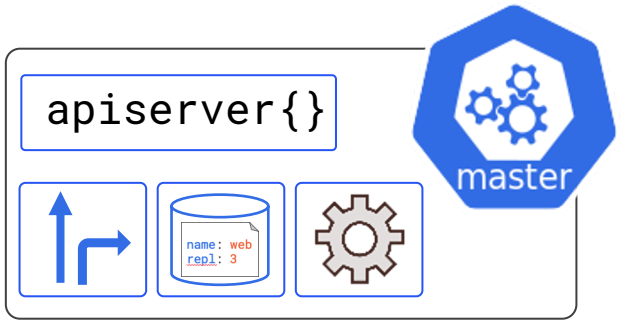

node
node
node

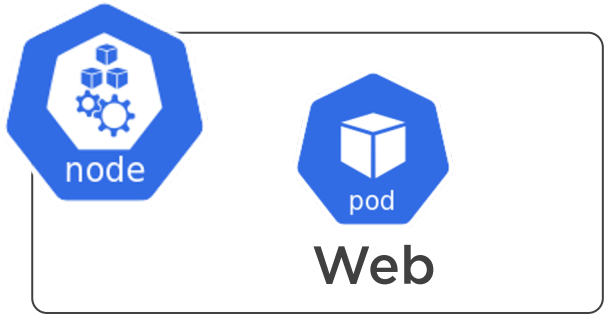

apiserver{}
master
name: web
repl: 3


node


node

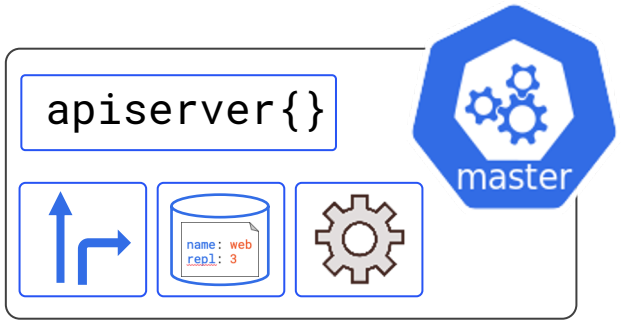

node

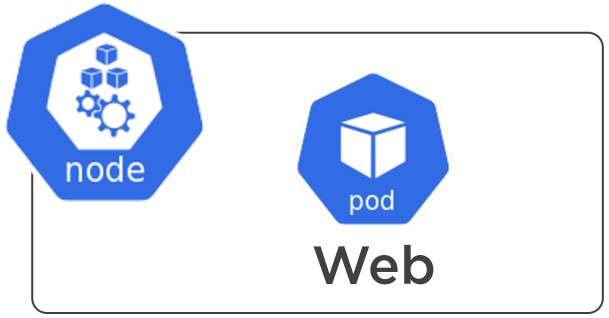

apiserver{}
master
name: web
repl: 3

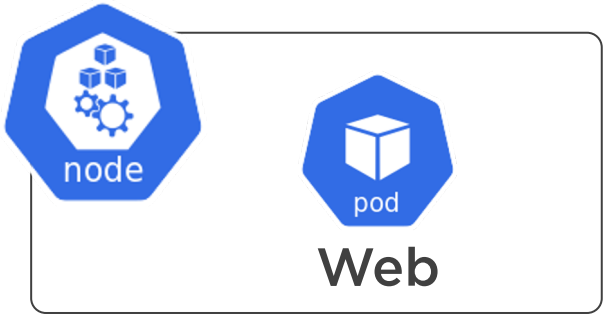

node
pod
Web


node

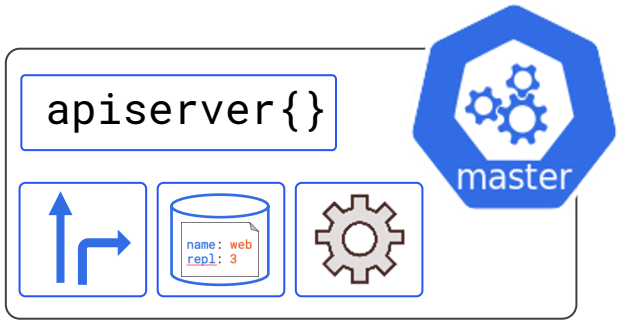

node

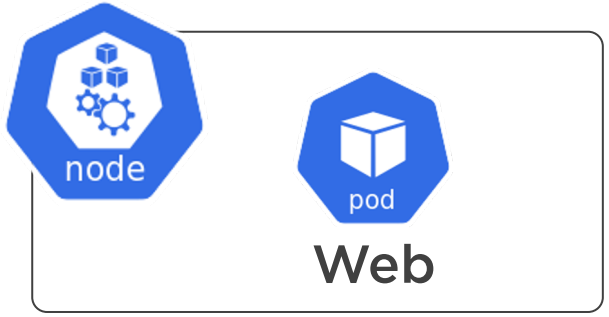

apiserver{}
master
name: web
repl: 3

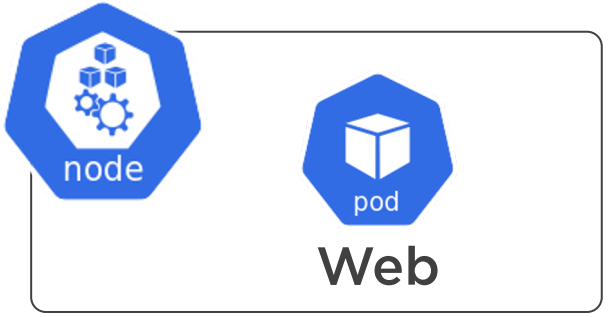

node
pod
Web

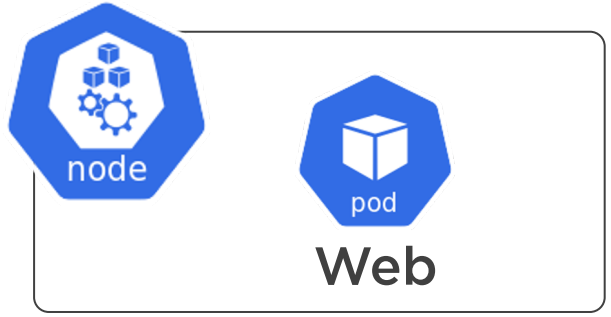

node
pod
Web

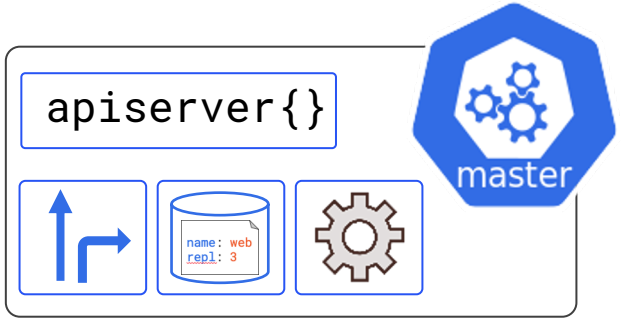

node


apiserver{}
master
name: web
repl: 3

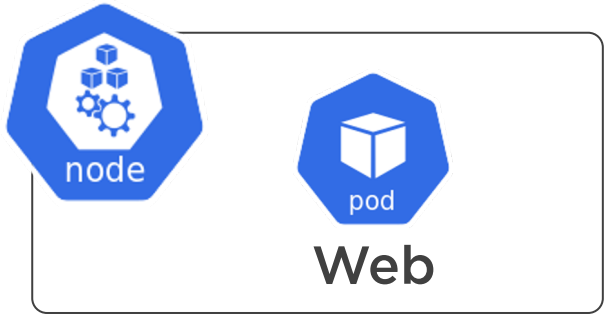

node
pod
Web

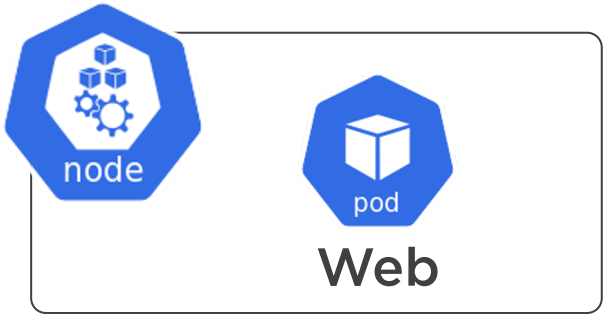

node
pod
Web

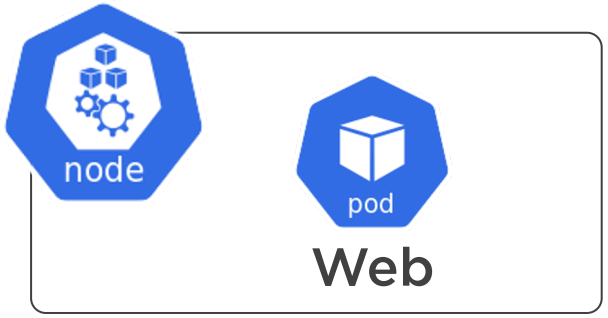

node
pod
Web

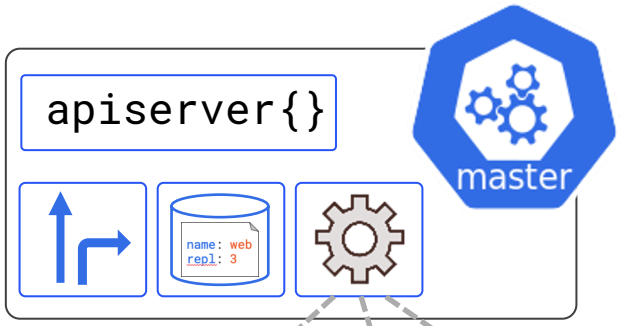


Desired state: 3
Observed state: 3

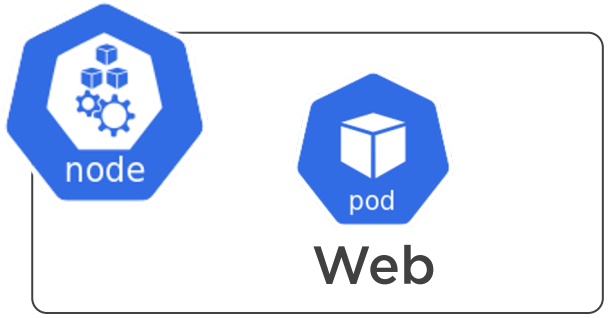


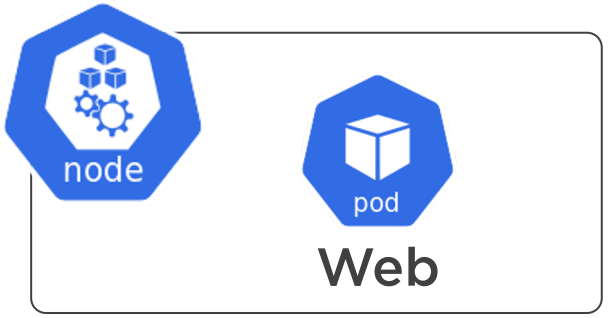


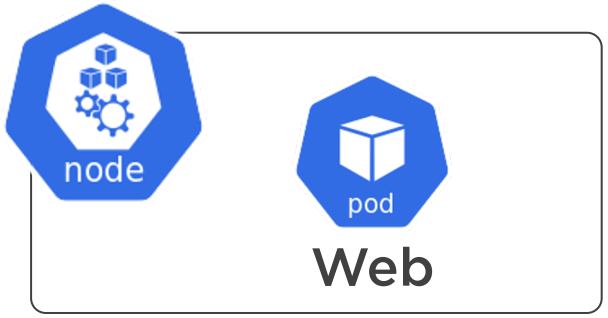




Desired state: 3
Observed state: 3

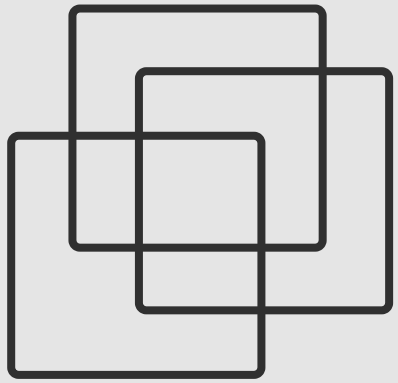






Up Next:
The Mighty Pod

The Mighty Pod

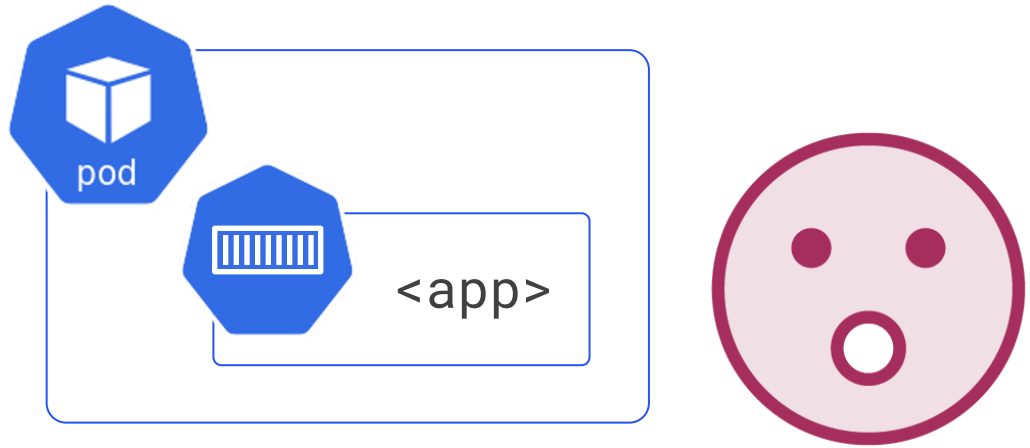


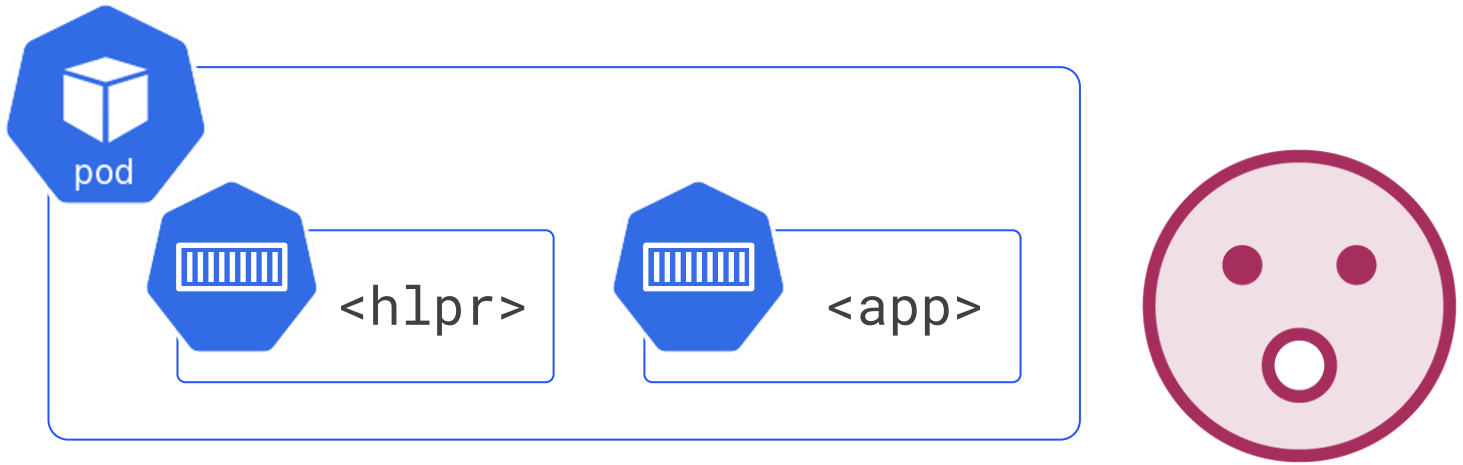
Virtual Machine

Container

Pod

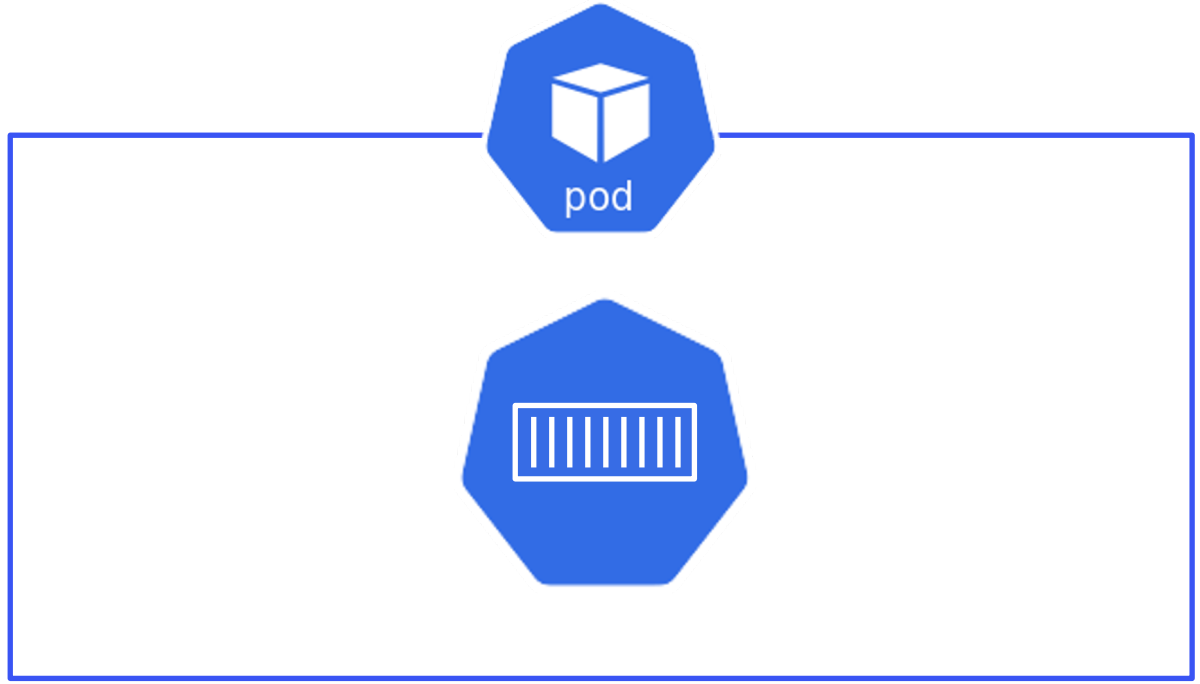
Atomic units of scheduling

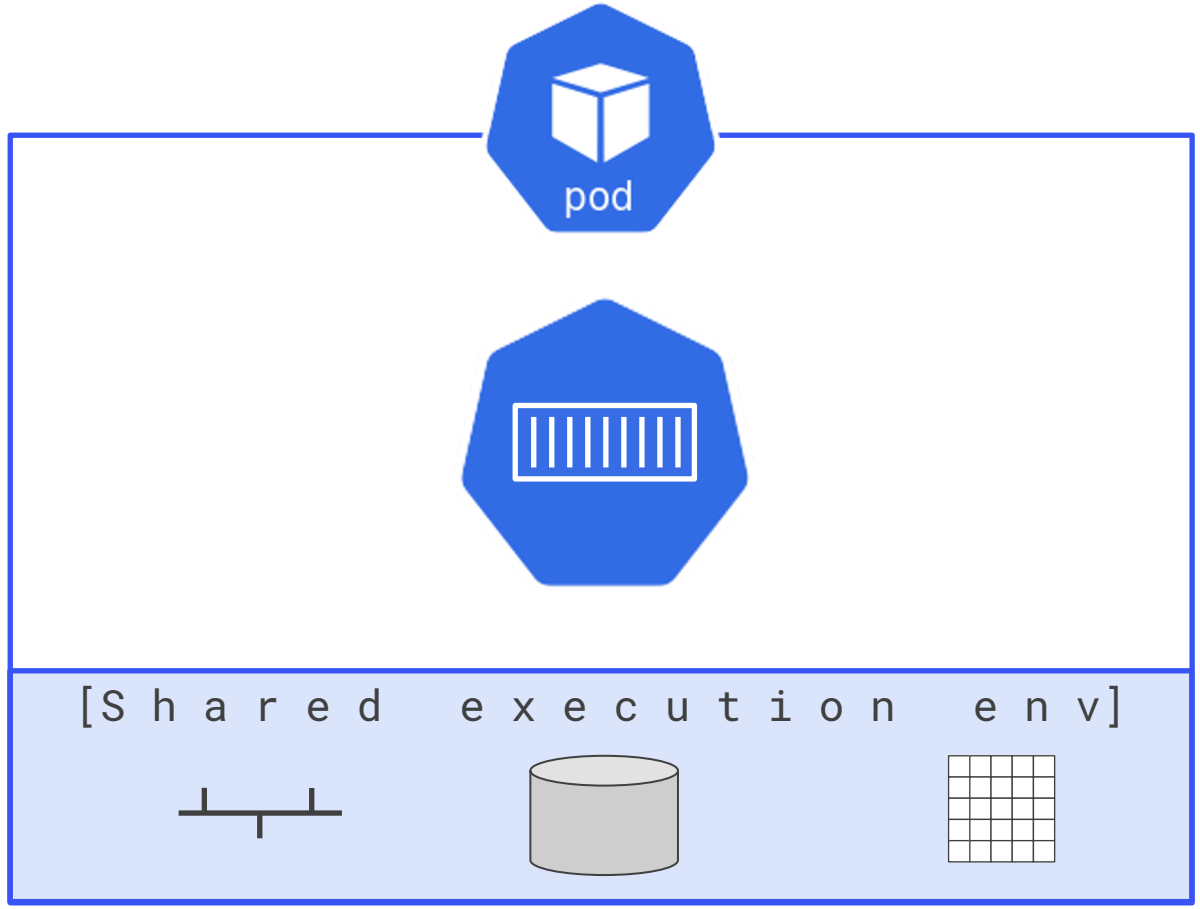

pod
<app>

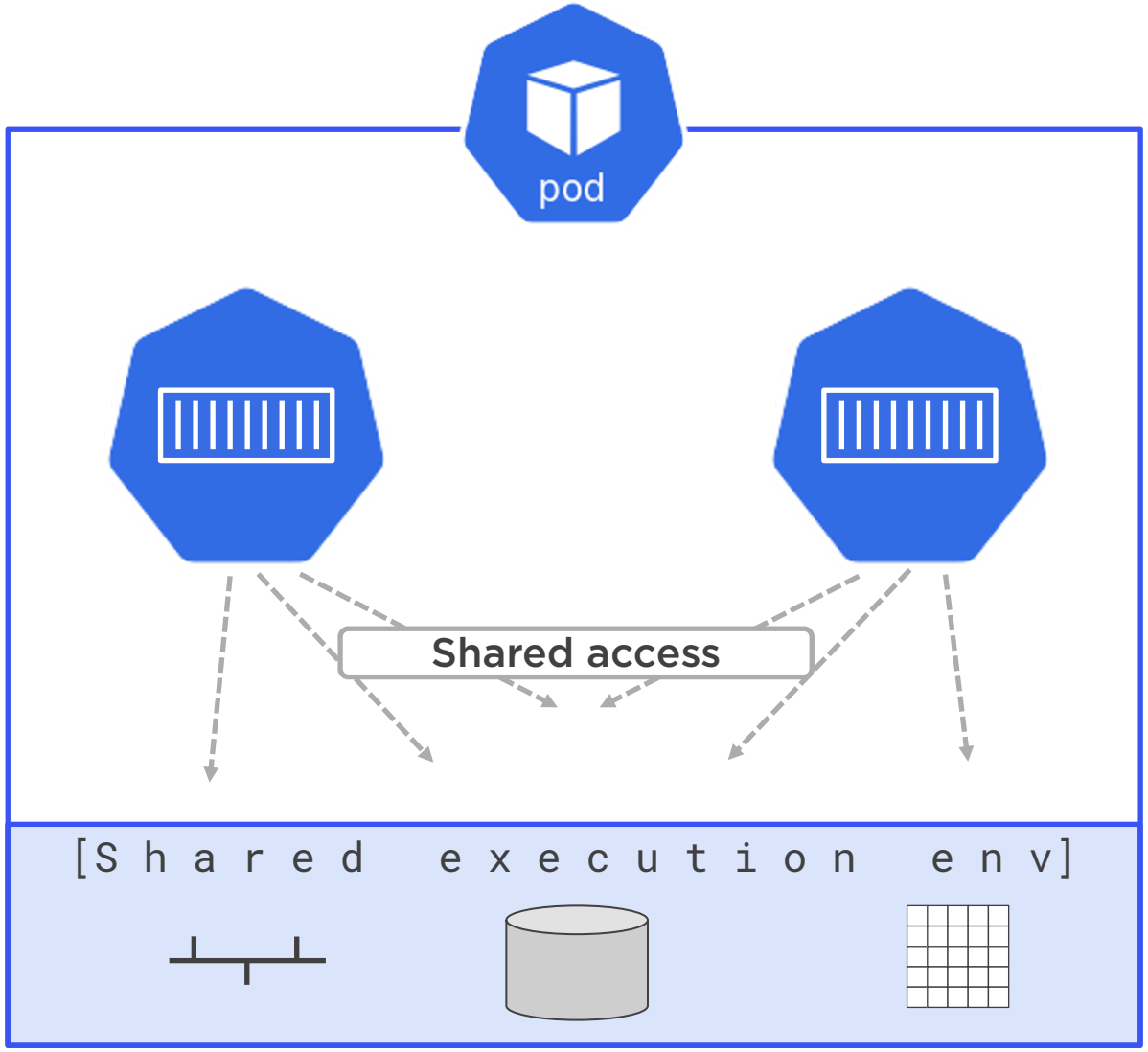

pod
<hlpr>
<app>

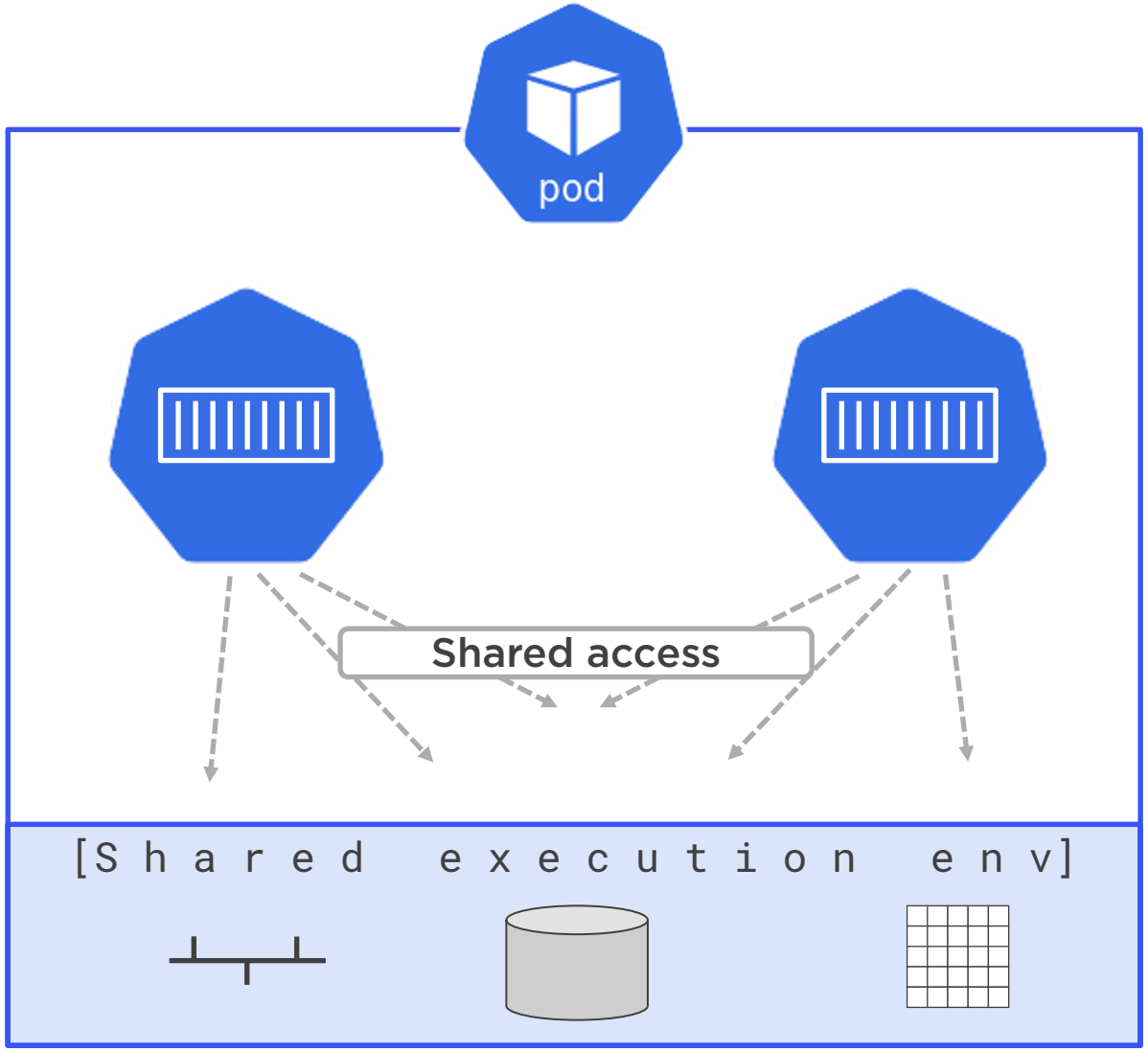

pod
<hlpr>
<app>

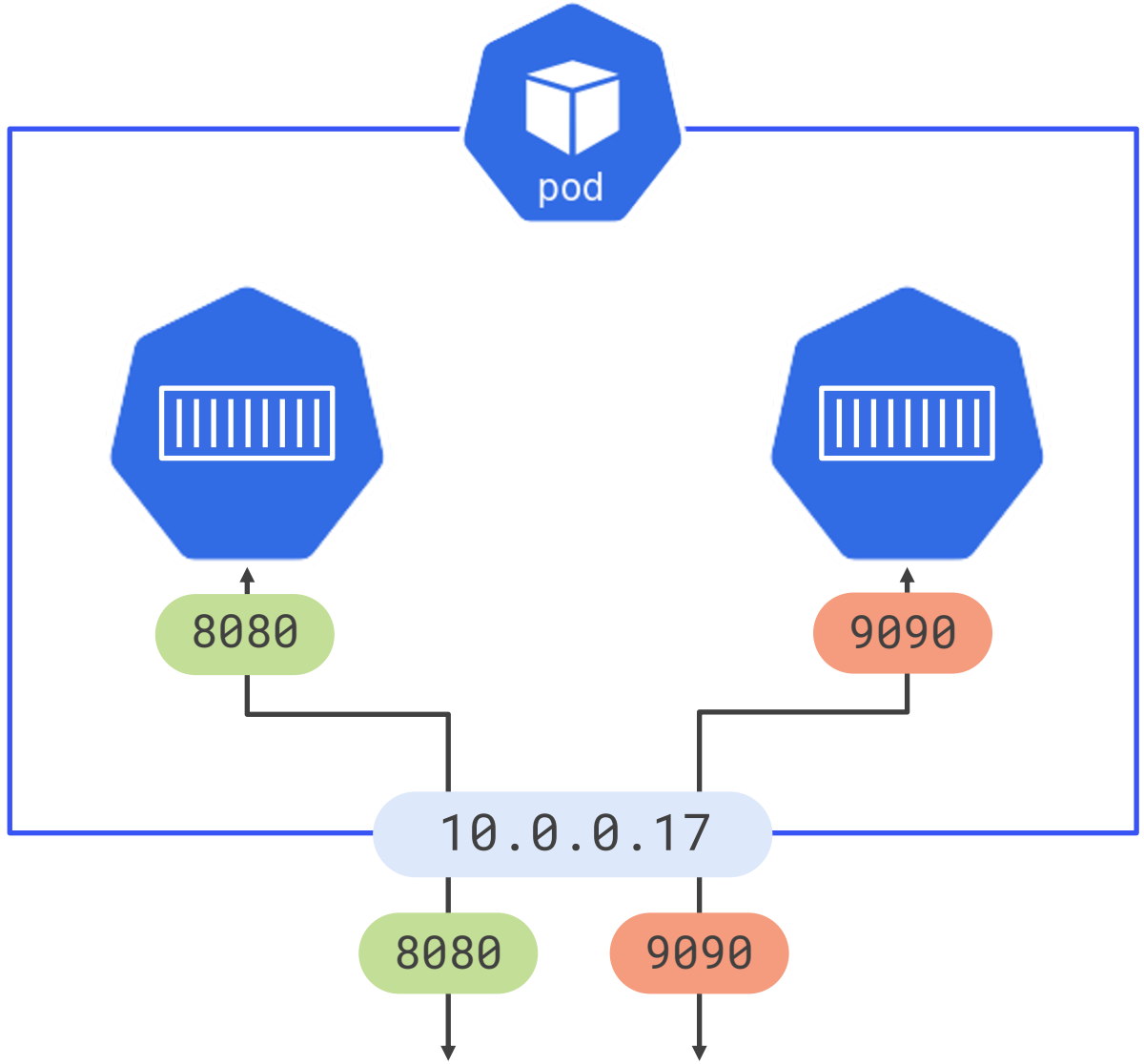
Kubernetes for Developers: Integrating Volumes and Using Multi-container Pods

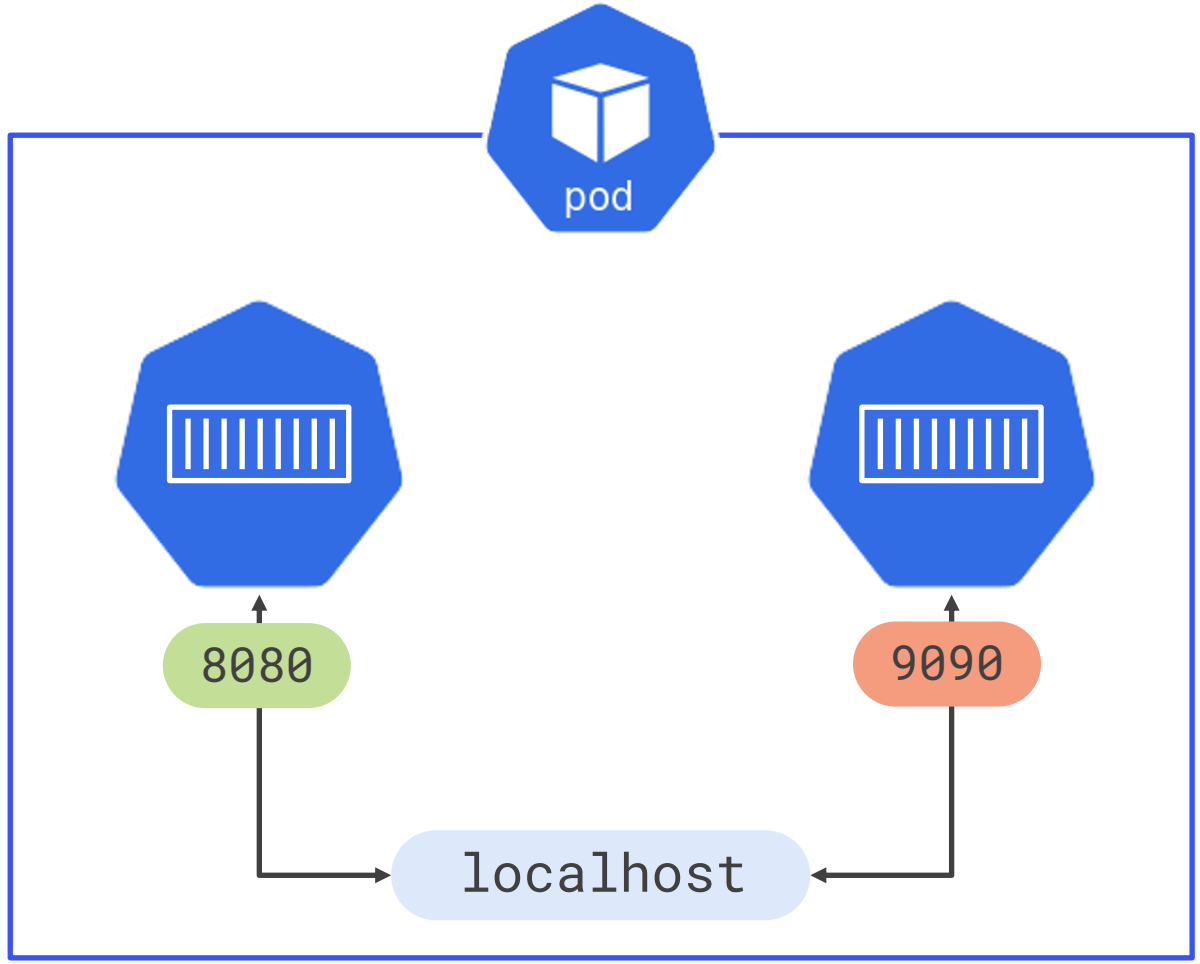

pod

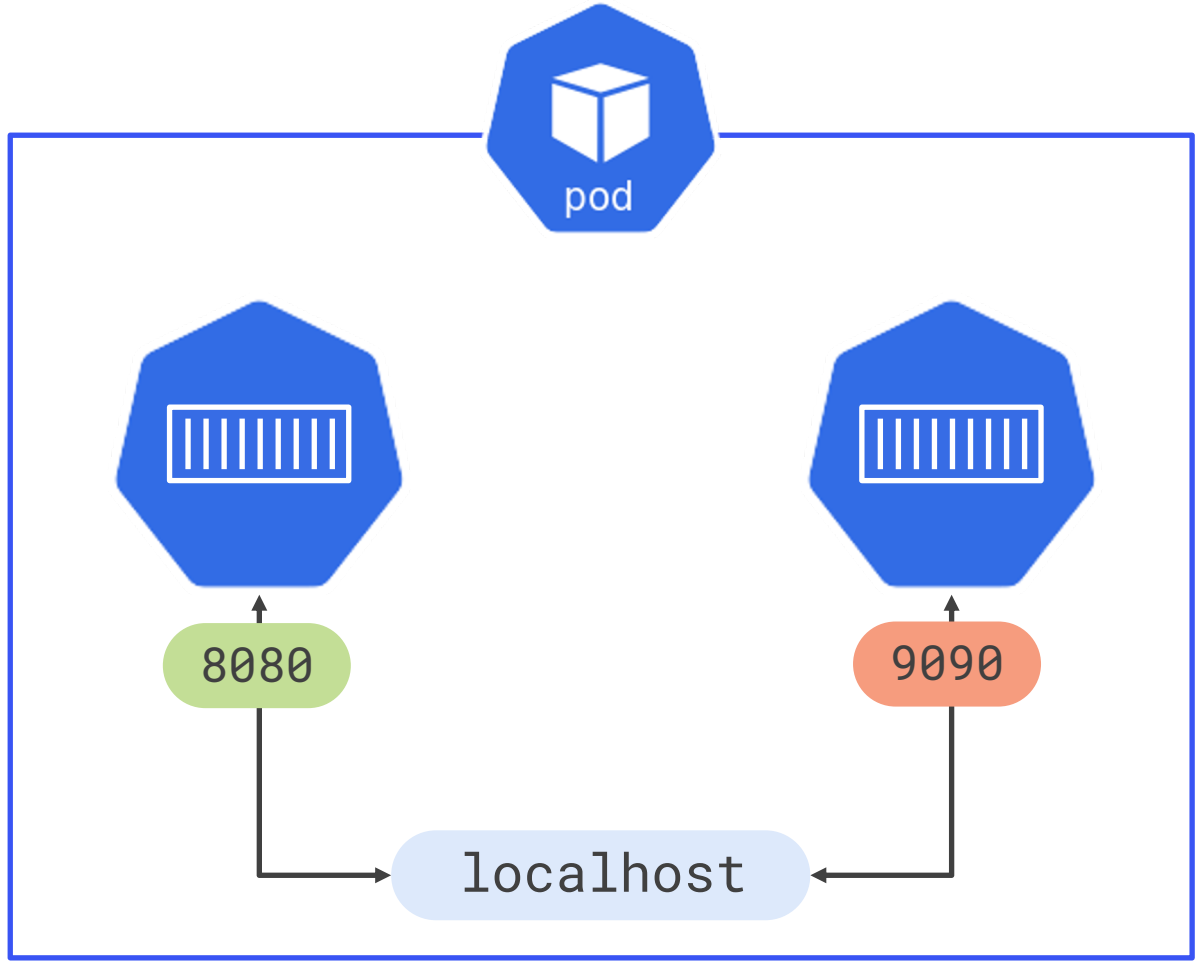

pod
[Shared execution env]

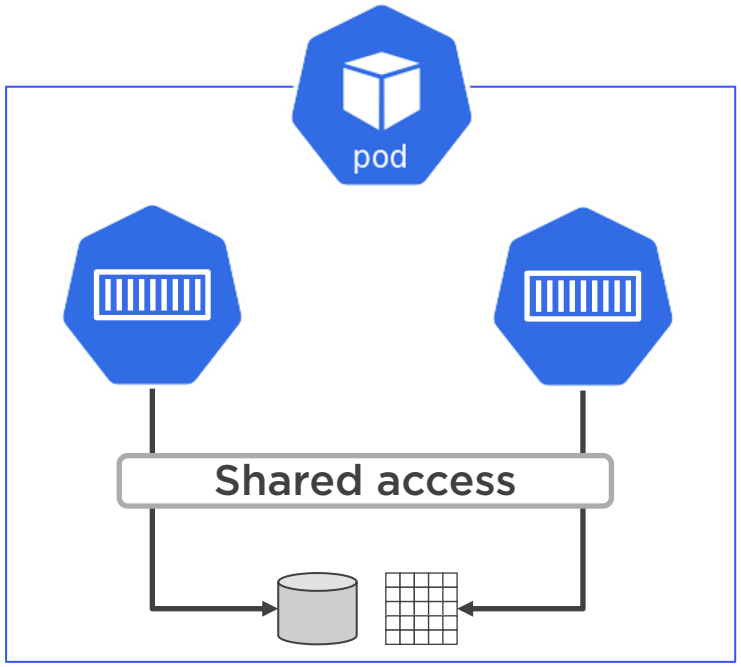

pod
Shared access
[Shared execution env]


pod
Shared access
[Shared execution env]


pod
8080
9090
10.0.0.17
8080
9090

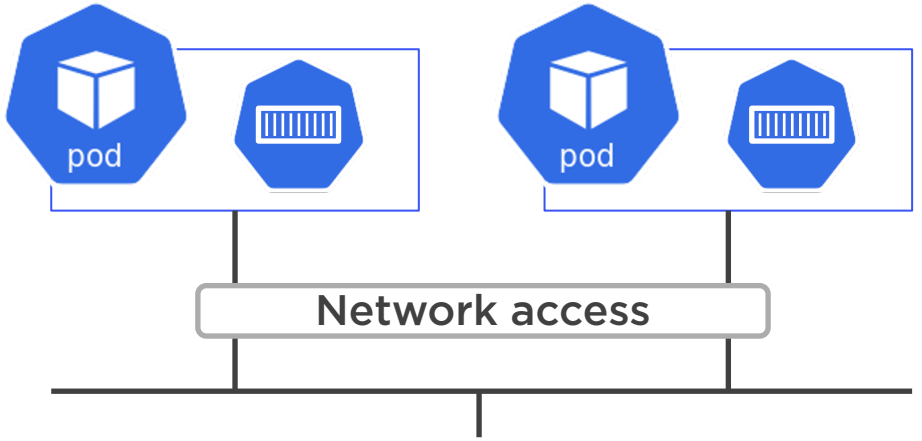

pod
8080
9090
localhost


pod
8080
9090
localhost



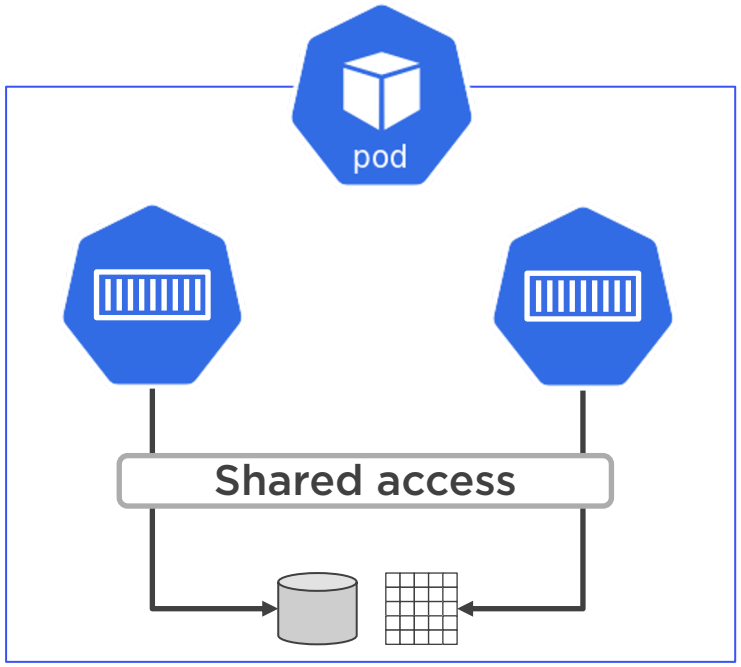
Tightly coupled

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



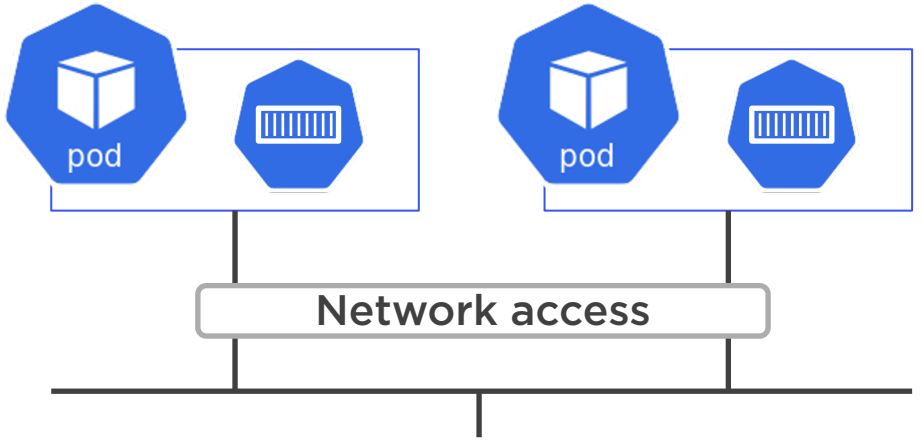
Loosely coupled

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



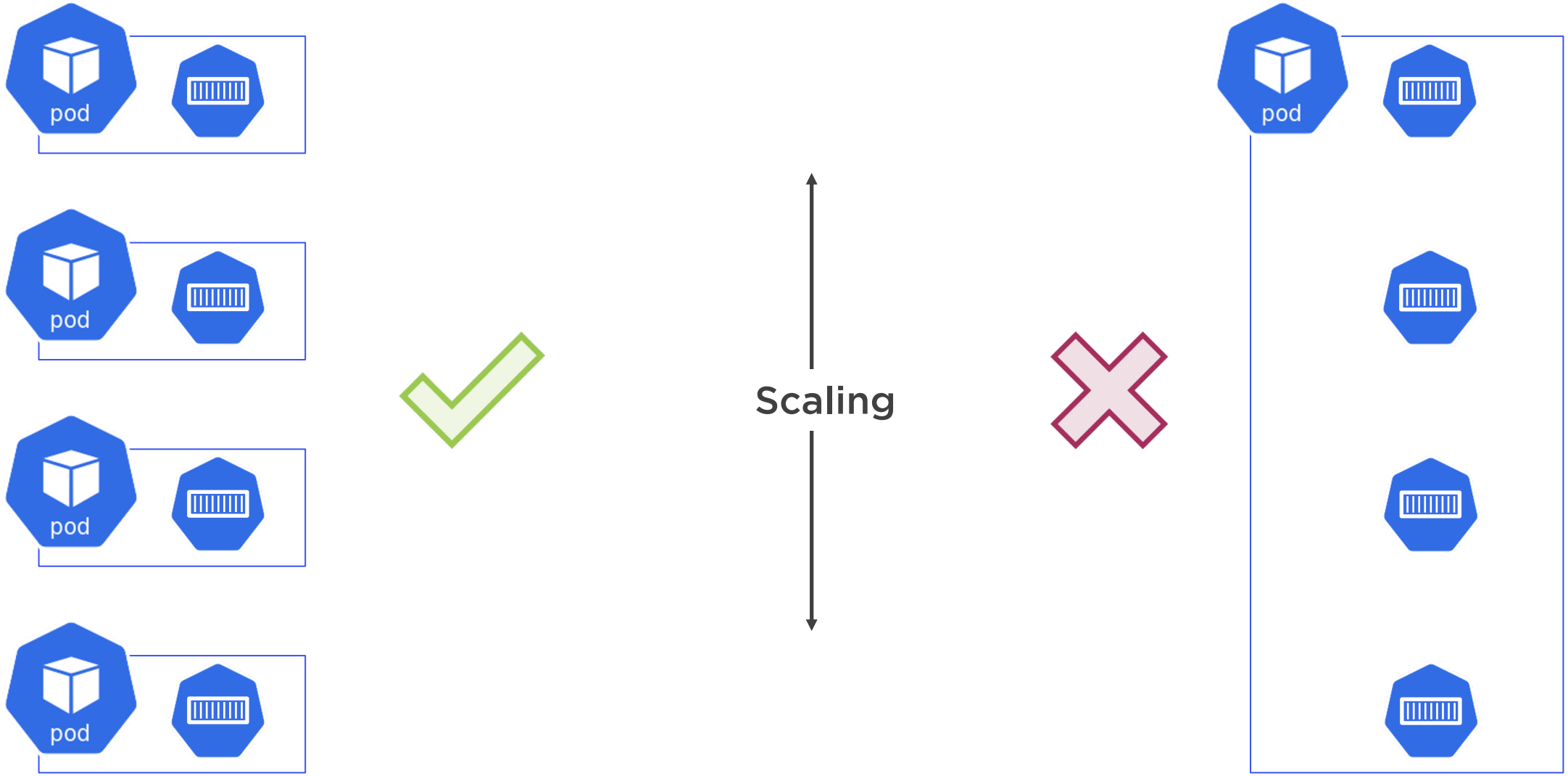
Tightly coupled

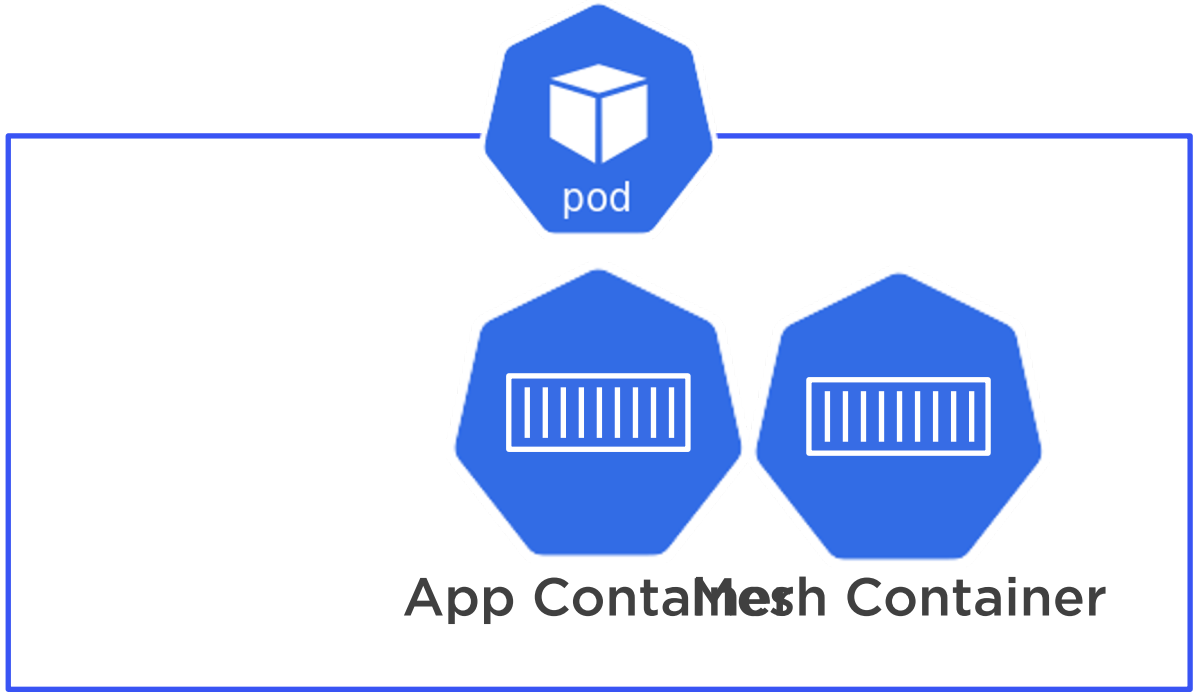
When two containers (app services) **absolutely need** to share vols, memory etc.

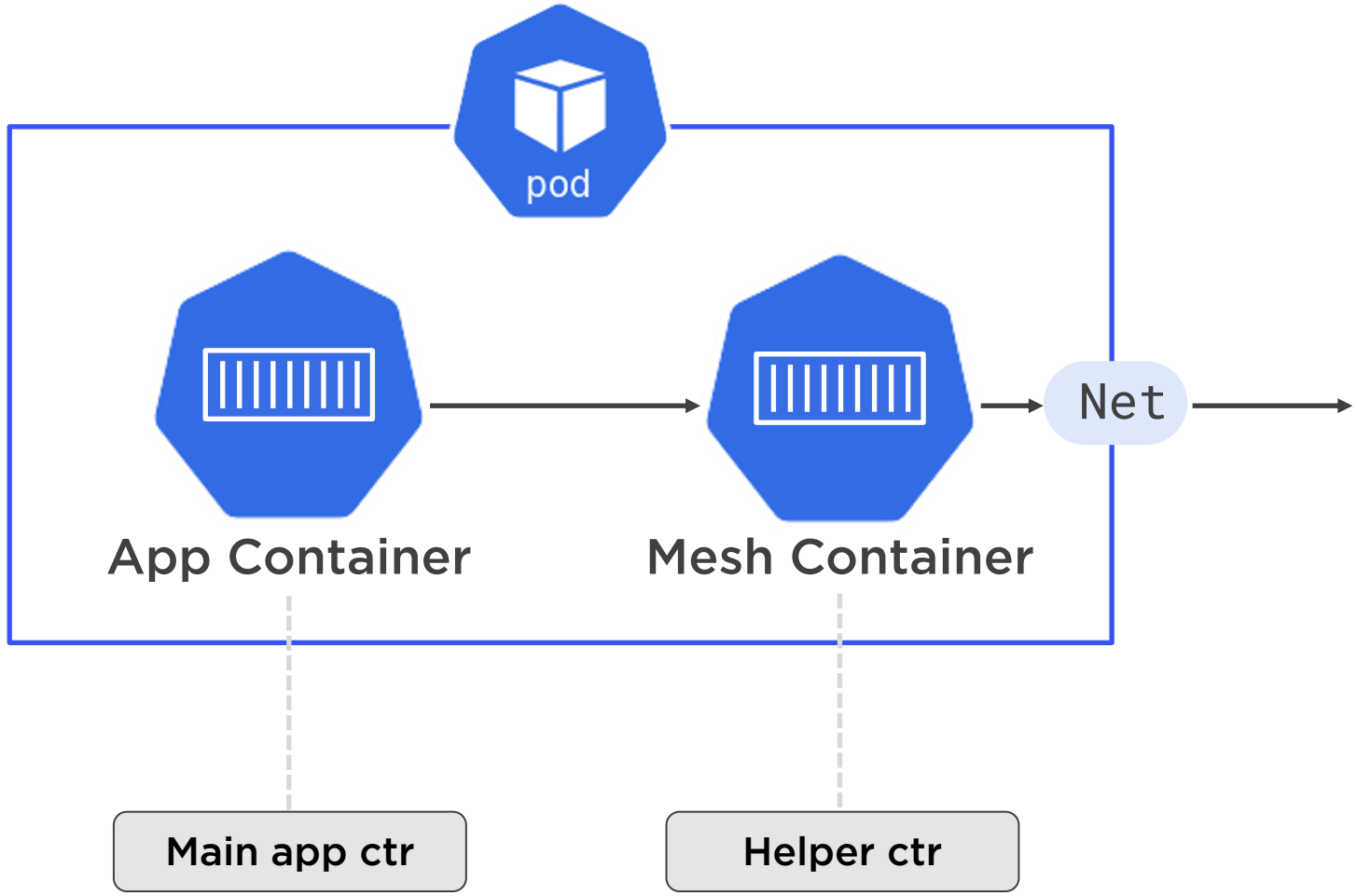


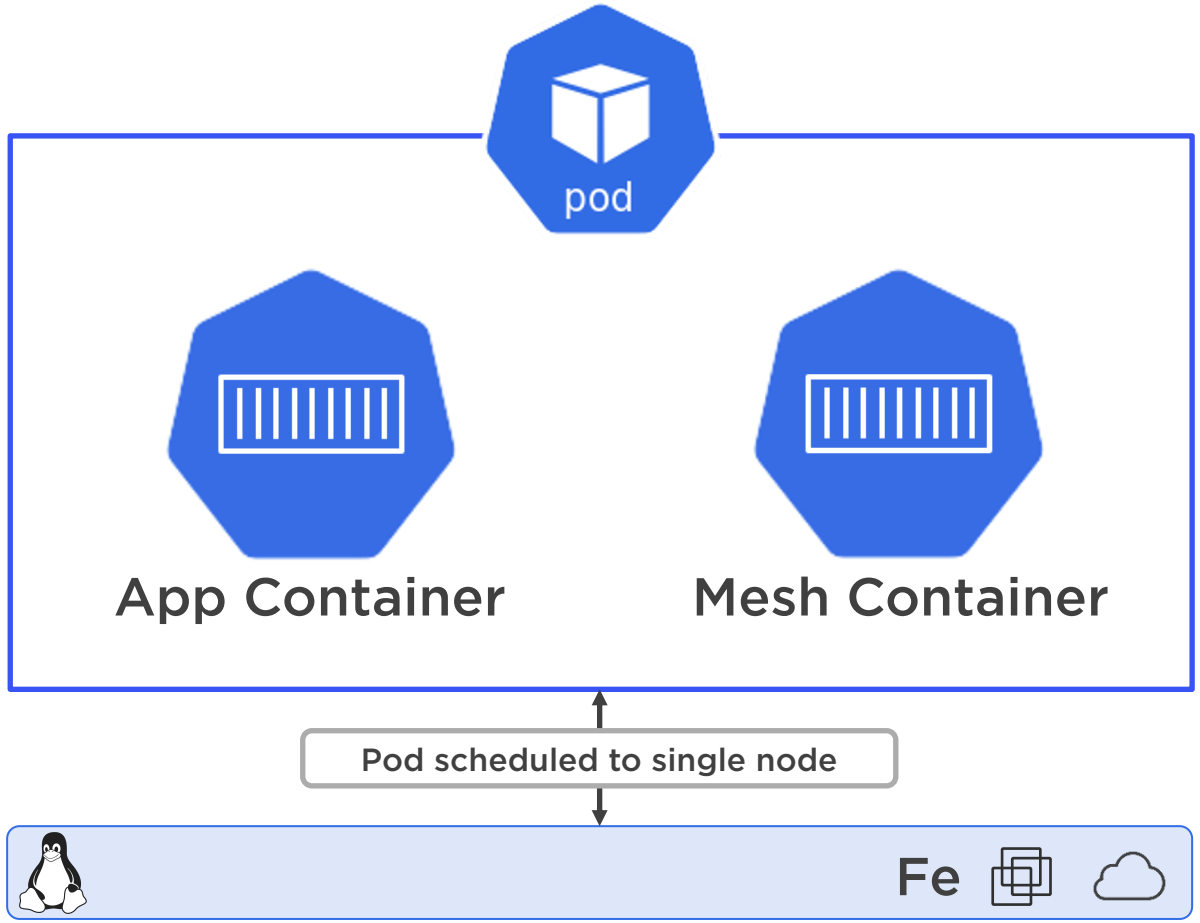
Loosely coupled

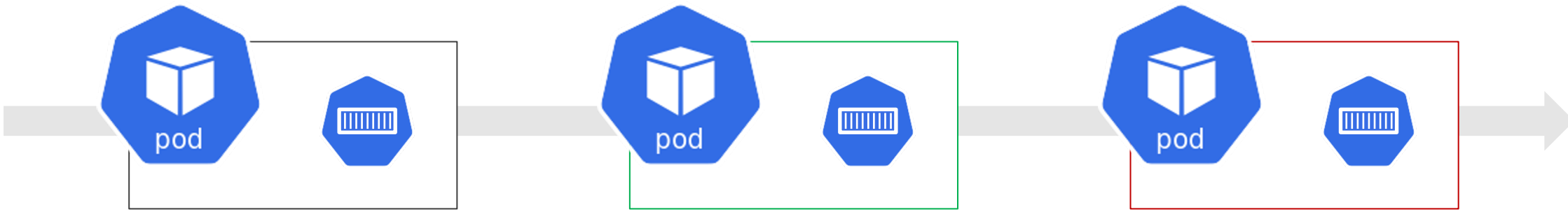
When two containers (app services) **don't** absolutely need to share resources


pod
pod
pod
pod
Scaling
pod


pod
App Container
Mesh Container


pod
App Container
Mesh Container
Net
Main app ctr
Helper ctr

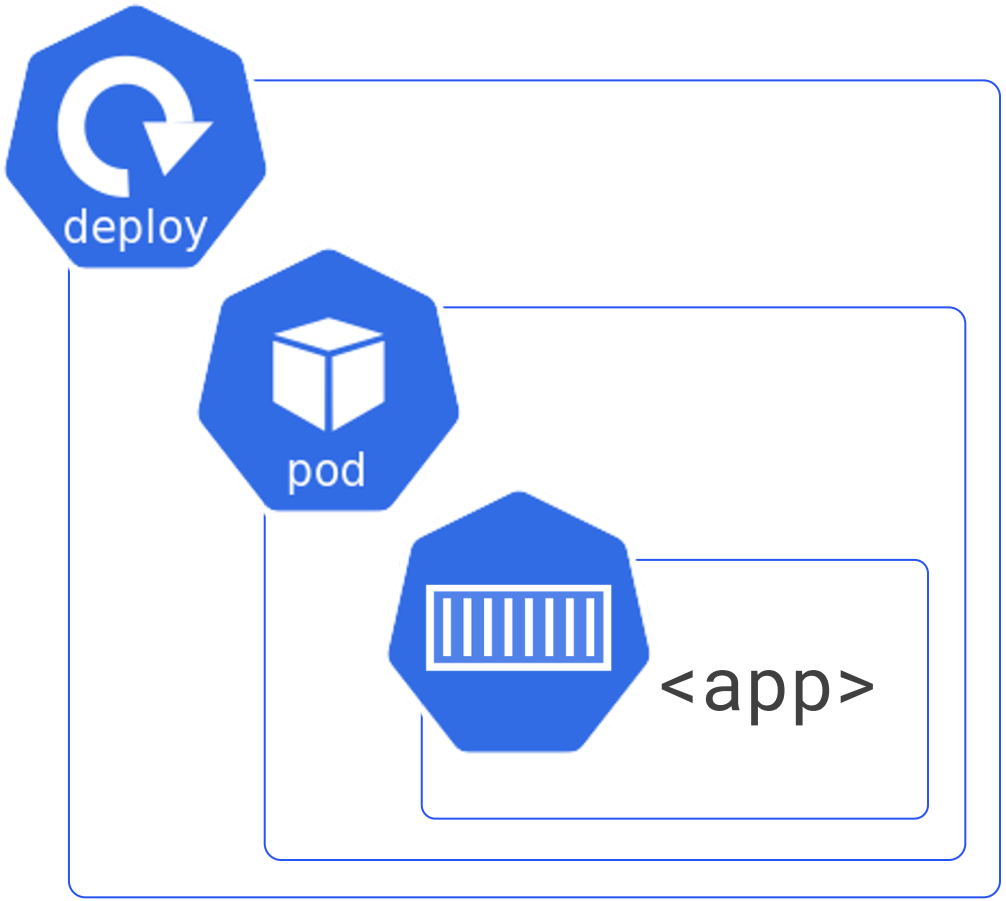

pod
App Container
Mesh Container
Pod scheduled to single node
Fe

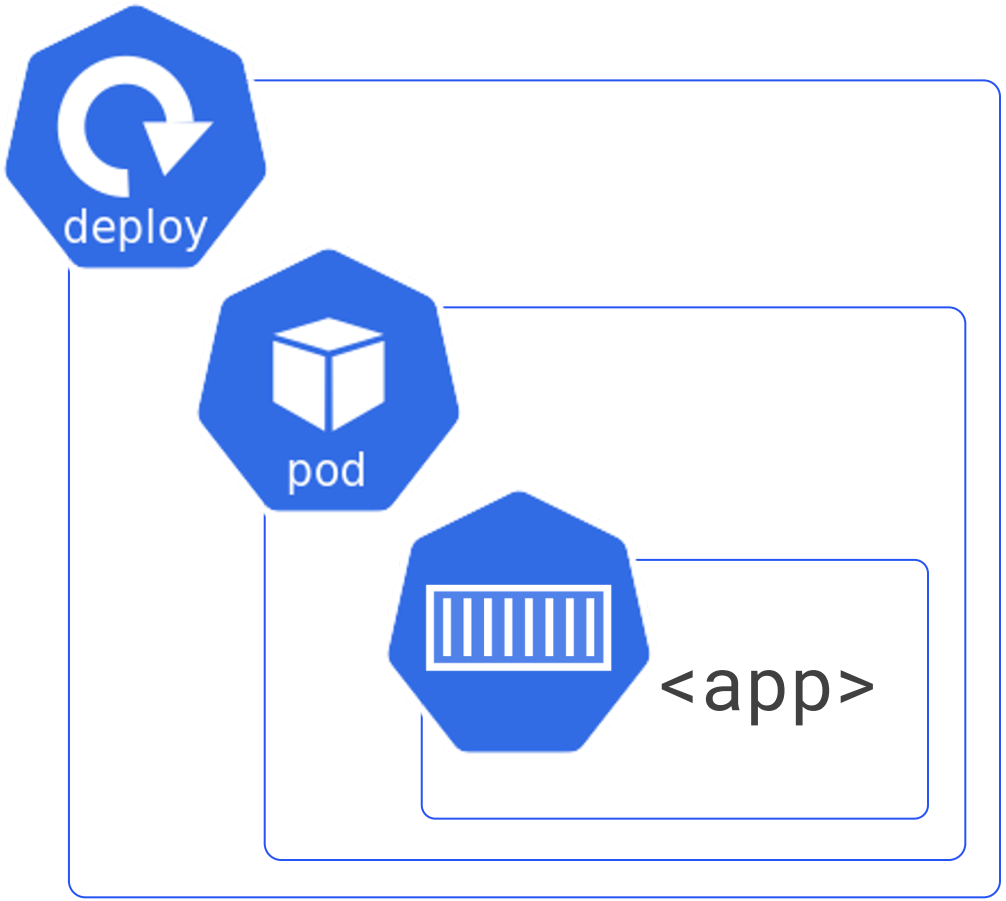


Phase: pending

Phase: running

Phase: succeeded/failed


deploy
pod
<app>


deploy
pod
<app>

Annotations

Labels

Policies

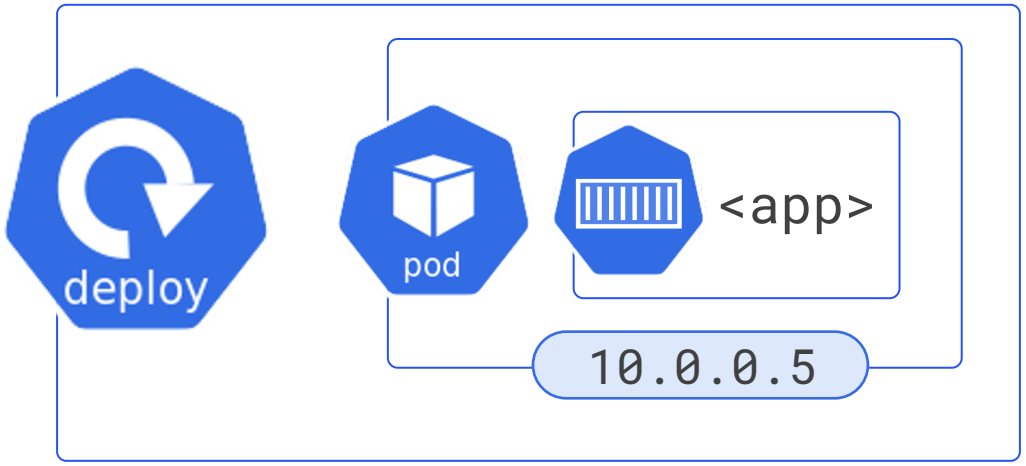
Resources

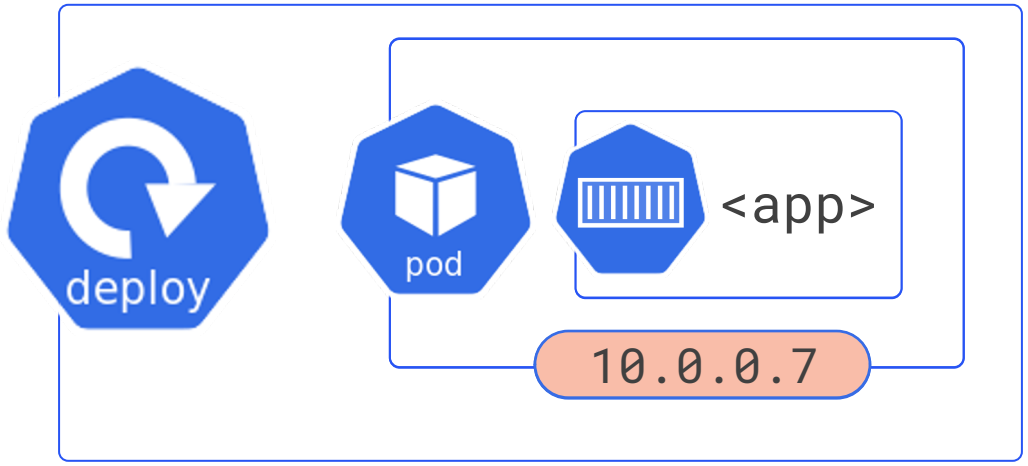
Co-scheduling containers

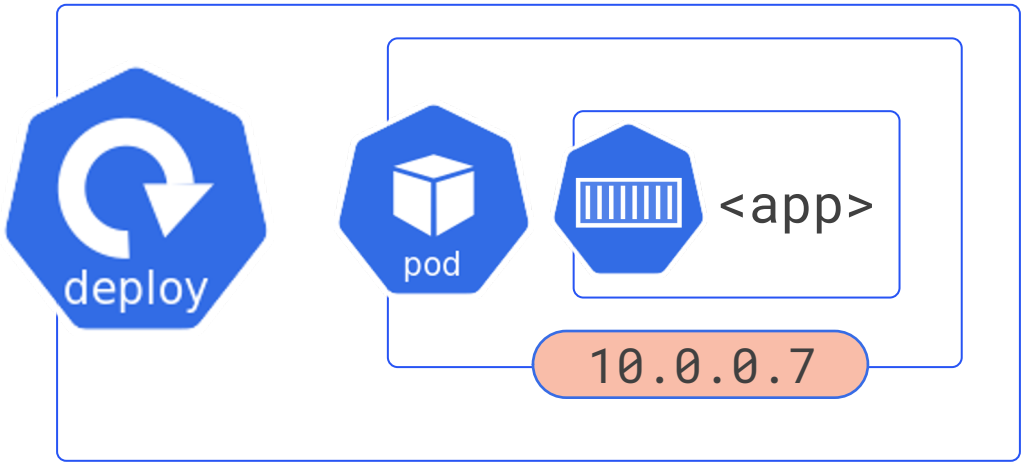
...

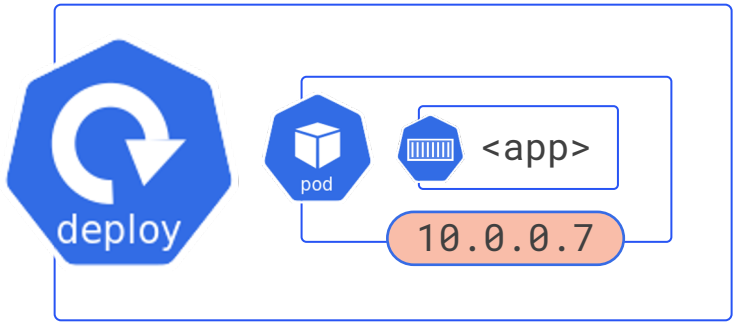
Up Next:
Stable Networking with Services

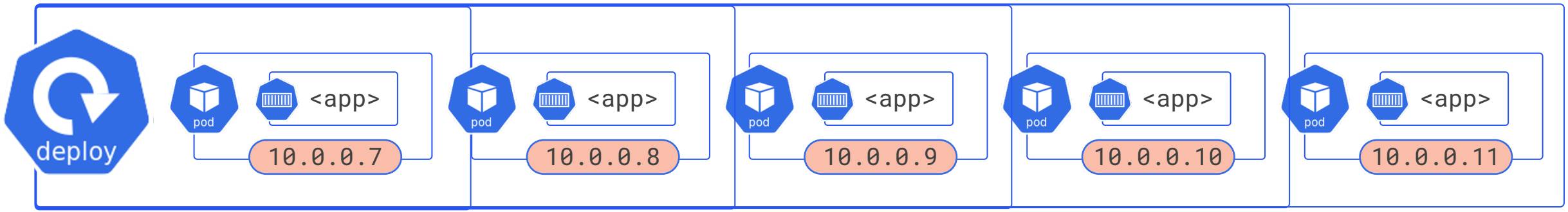
Stable Networking with Services

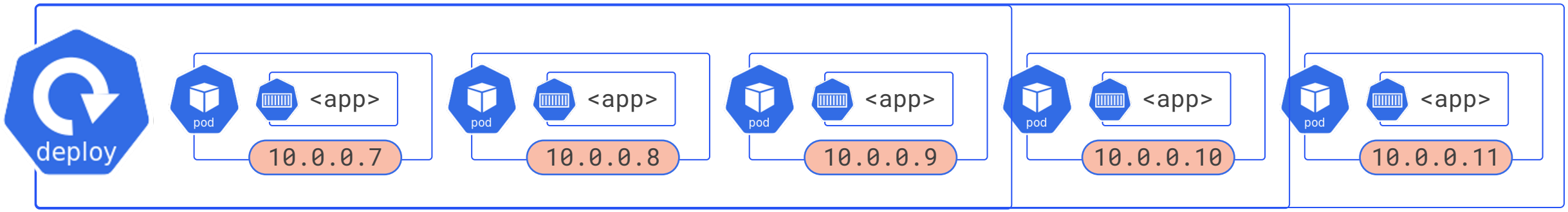

deploy
pod
<app>
10.0.0.5

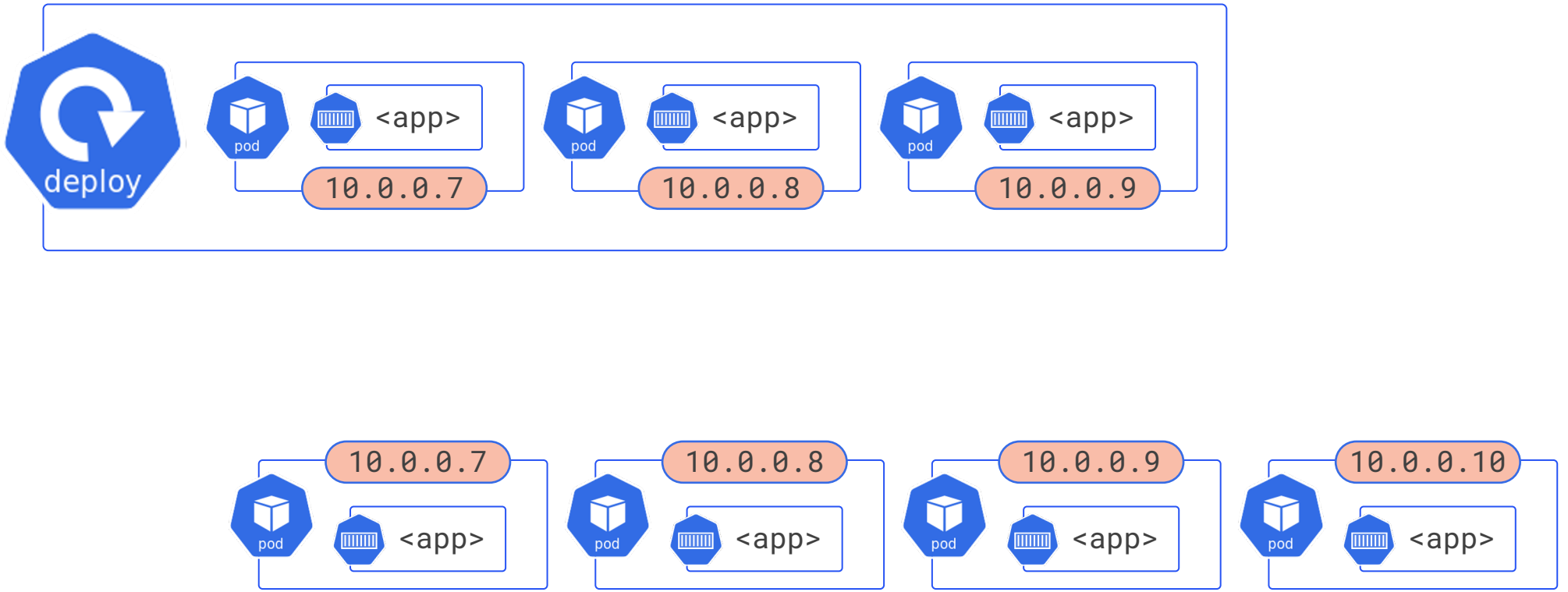

deploy
pod
<app>
10.0.0.7


deploy
pod
<app>
10.0.0.7

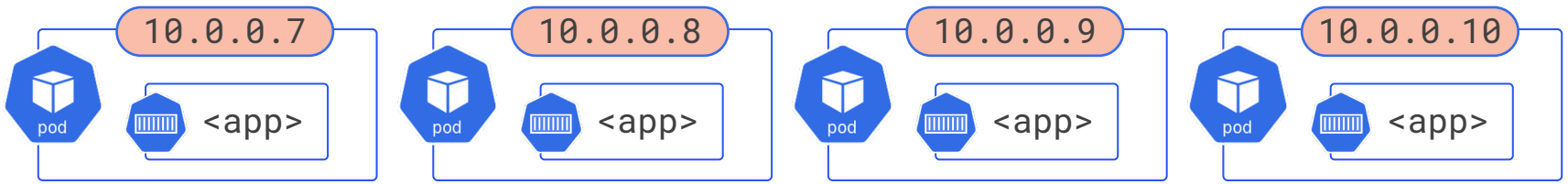

deploy
pod
<app>
10.0.0.7


deploy
pod
<app>
10.0.0.7
pod
<app>
10.0.0.8
pod
<app>
10.0.0.9
pod
<app>
10.0.0.10
pod
<app>
10.0.0.11

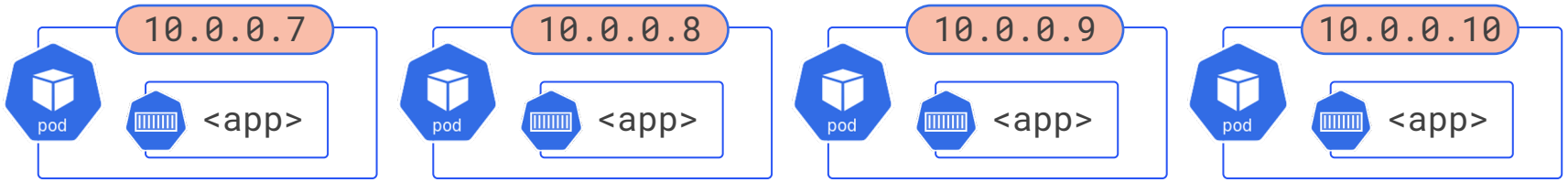

deploy
pod
<app>
10.0.0.7
pod
<app>
10.0.0.8
pod
<app>
10.0.0.9
pod
<app>
10.0.0.10
pod
<app>
10.0.0.11


deploy
pod
<app>
10.0.0.7
pod
<app>
10.0.0.8
pod
<app>
10.0.0.9
10.0.0.7
pod
<app>
10.0.0.8
pod
<app>
10.0.0.9
pod
<app>
10.0.0.10
pod
<app>

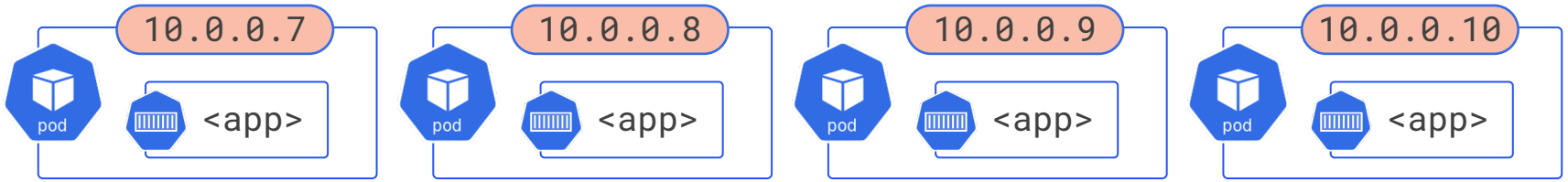

App/
client

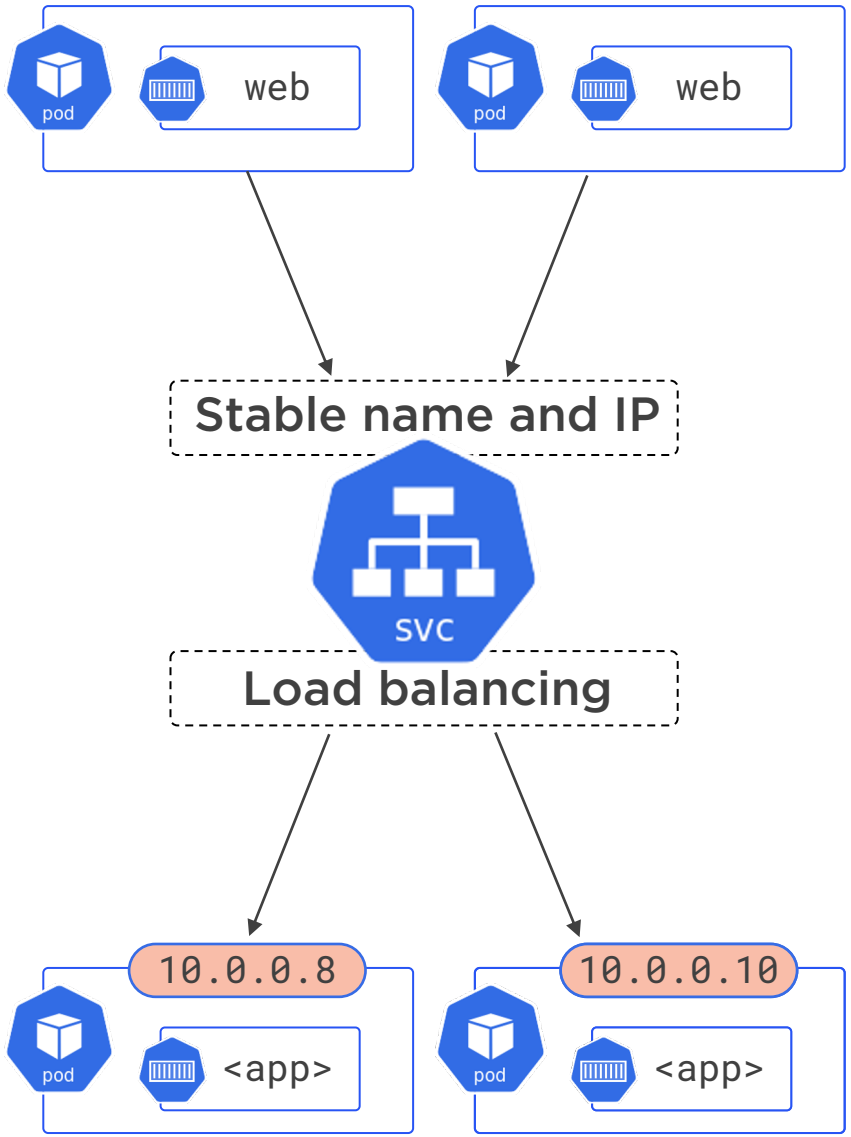

10.0.0.7
pod
<app>
10.0.0.8
pod
<app>
10.0.0.9
pod
<app>
10.0.0.10
pod
<app>

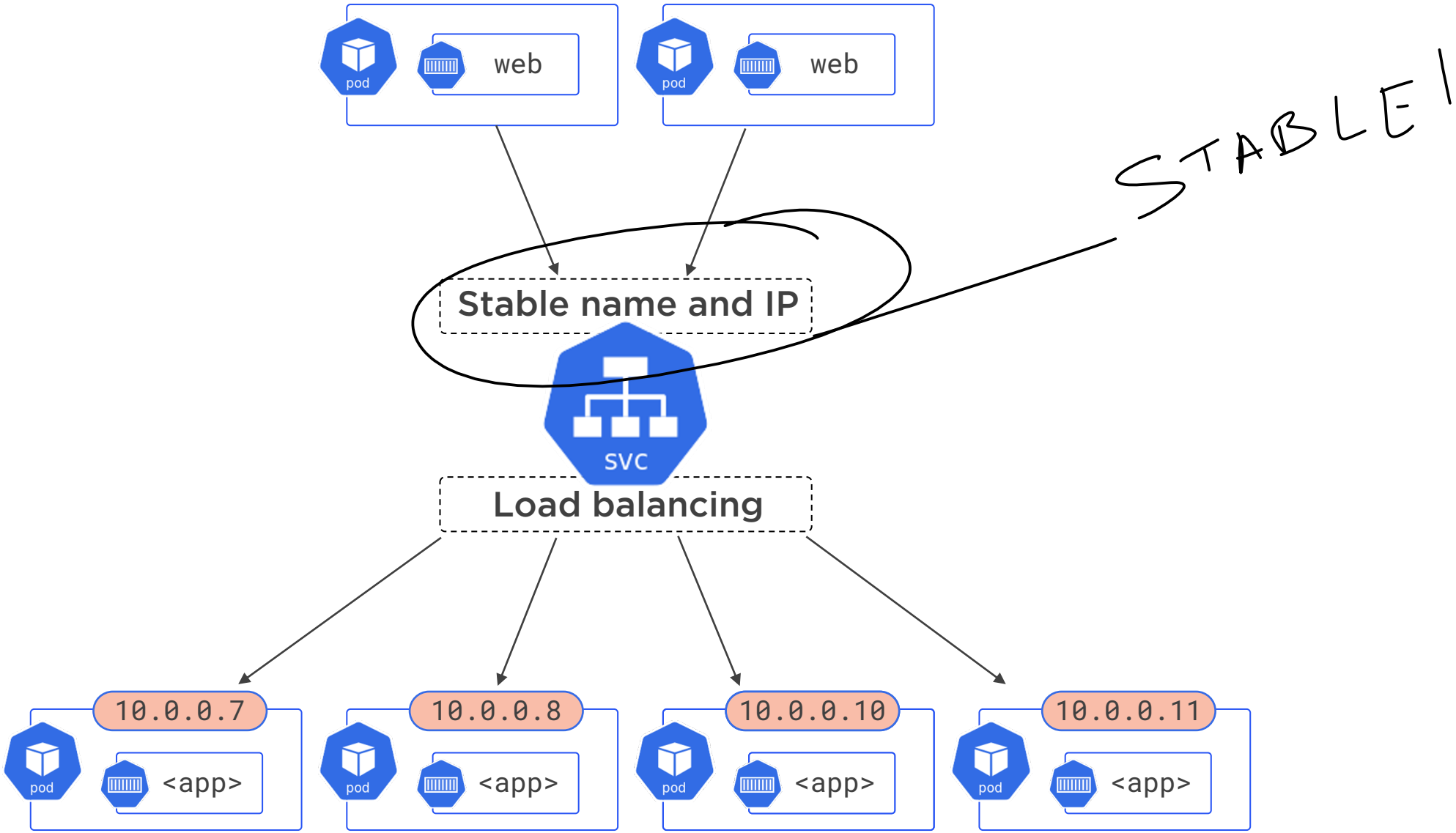

Unreliable Pod IPs

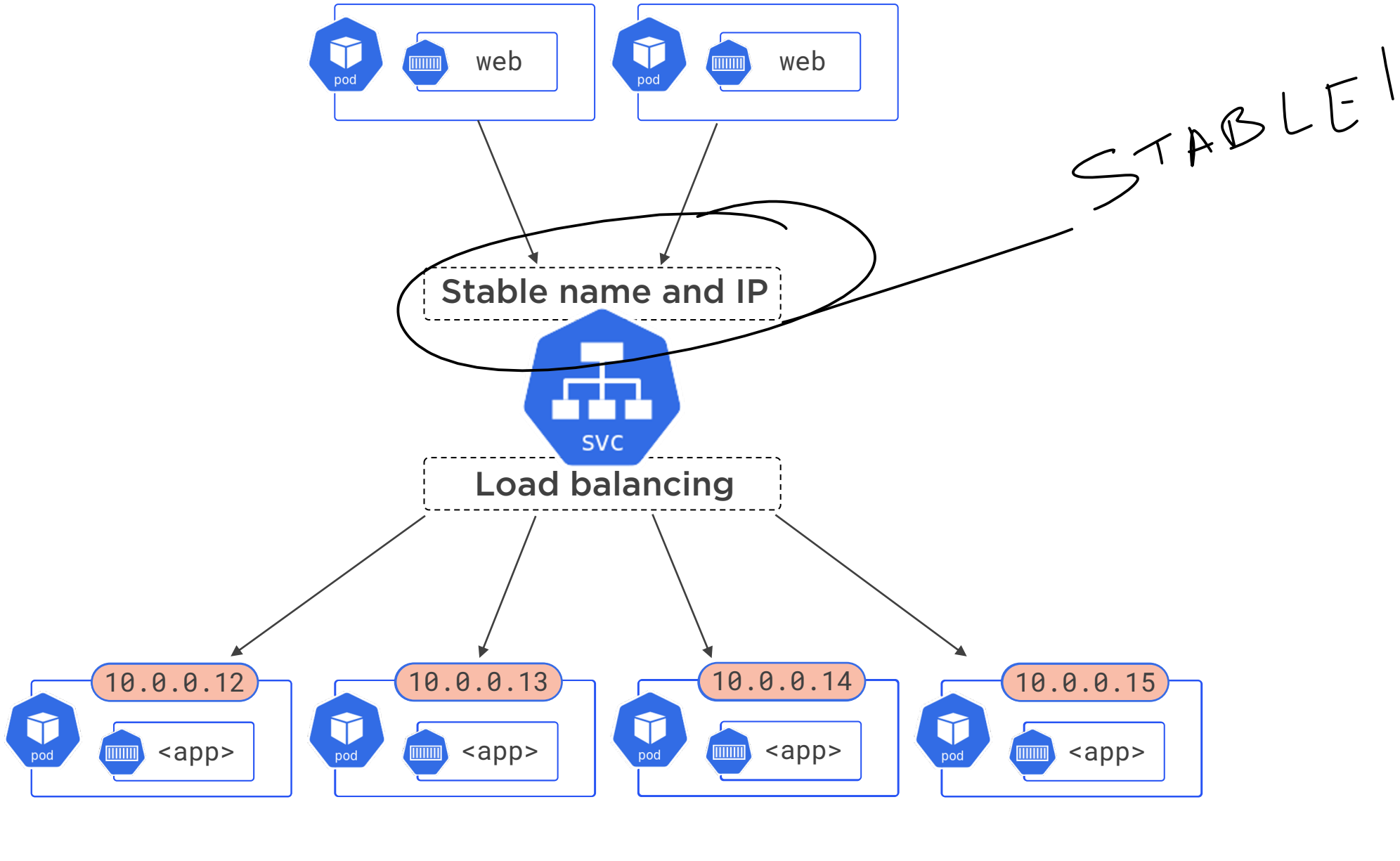

10.0.0.7
pod
<app>
10.0.0.8
pod
<app>
10.0.0.9
pod
<app>
10.0.0.10
pod
<app>


Unreliable Pod IPs

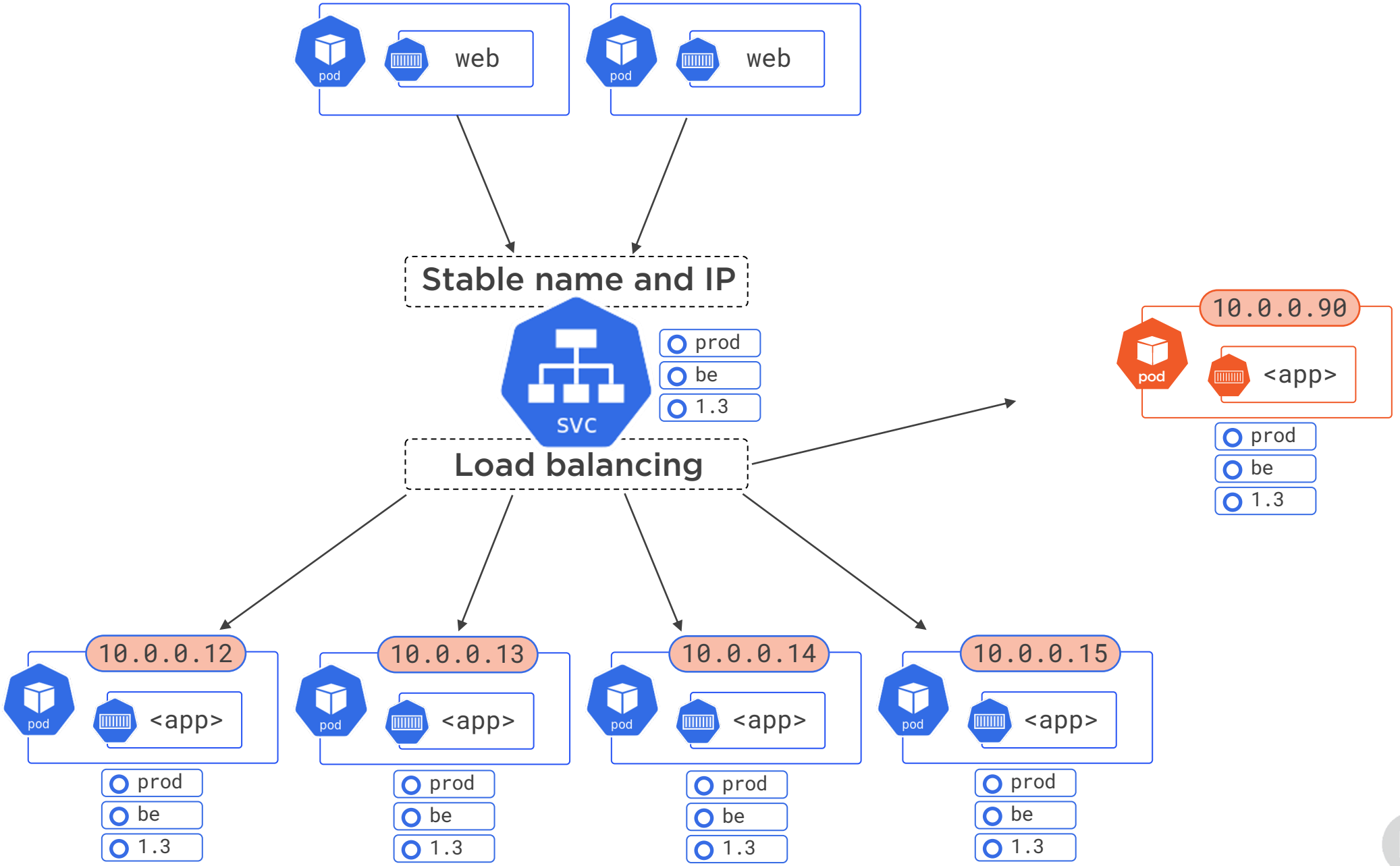

10.0.0.7
pod
<app>
10.0.0.8
pod
<app>
10.0.0.9
pod
<app>
10.0.0.10
pod
<app>

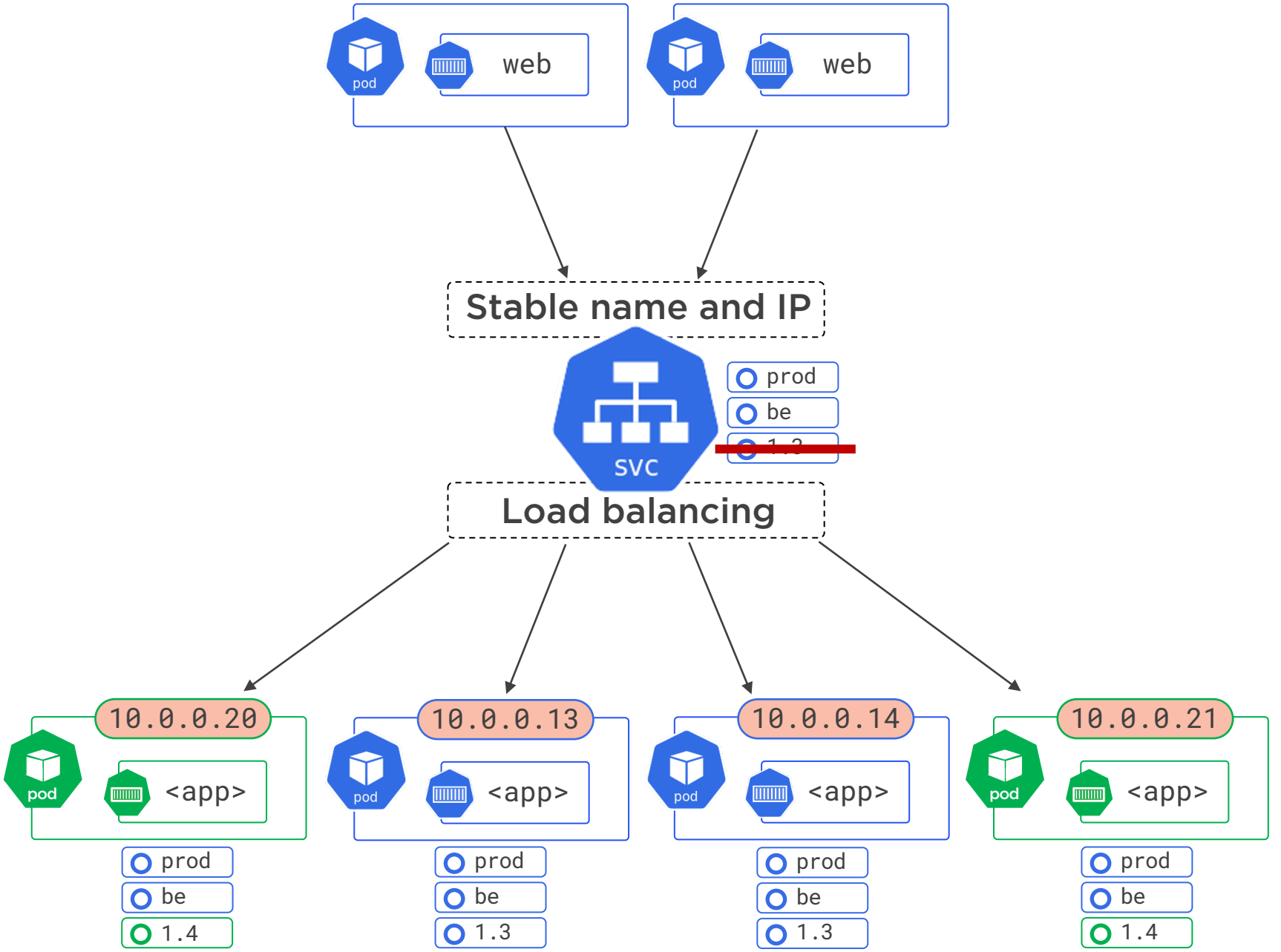

pod
web
pod
web
Stable name and IP
SVC
Load balancing
10.0.0.8
pod
<app>
10.0.0.10
pod
<app>

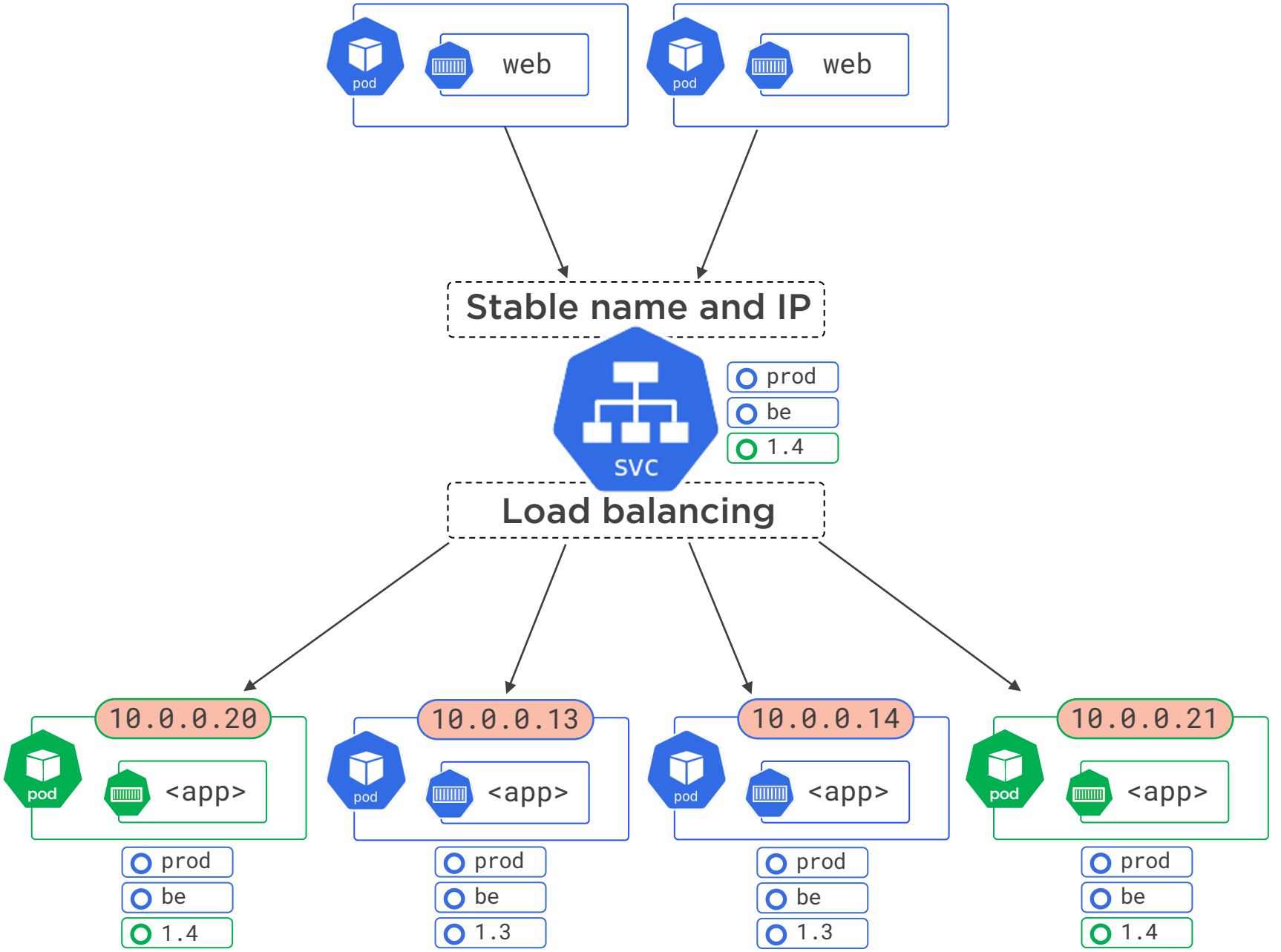

pod
web
pod
web
STABLE!
Stable name and IP
SVC
Load balancing
10.0.0.7
pod
<app>
10.0.0.8
pod
<app>
10.0.0.10
pod
<app>
10.0.0.11
pod
<app>

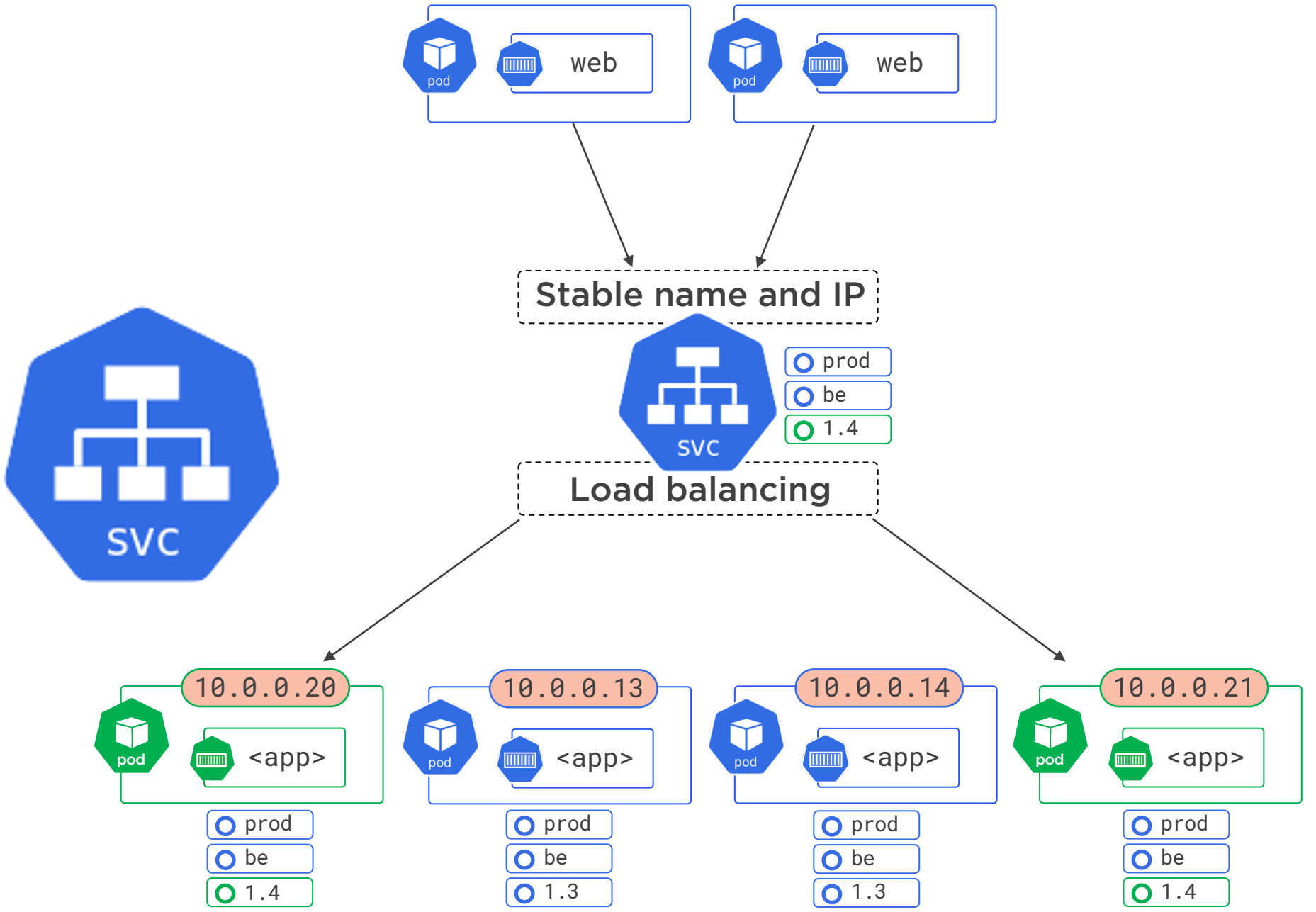

pod
web
pod
web
STABLE!
Stable name and IP
SVC
Load balancing
10.0.0.12
pod
<app>
10.0.0.13
pod
<app>
10.0.0.14
pod
<app>
10.0.0.15
pod
<app>

Labels are very simple and very powerful!


pod
web
pod
web
Stable name and IP
SVC
prod
be
1.3
Load balancing
10.0.0.90
pod
<app>
prod
be
1.3
10.0.0.12
pod
<app>
prod
be
1.3
10.0.0.13
pod
<app>
prod
be
1.3
10.0.0.14
pod
<app>
prod
be
1.3
10.0.0.15
pod
<app>
prod
be
1.3


pod
web
pod
web
Stable name and IP
SVC
prod
be
Load balancing
10.0.0.20
pod
<app>
prod
be
1.4
10.0.0.13
pod
<app>
prod
be
1.3
10.0.0.14
pod
<app>
prod
be
1.3
10.0.0.21
pod
<app>
prod
be
1.4


pod
web
pod
web
Stable name and IP
SVC
prod
be
1.4
Load balancing
10.0.0.20
pod
<app>
prod
be
1.4
10.0.0.13
pod
<app>
prod
be
1.3
10.0.0.14
pod
<app>
prod
be
1.3
10.0.0.21
pod
<app>
prod
be
1.4


pod
web
pod
web
Stable name and IP
SVC
prod
be
1.4
Load balancing
SVC
10.0.0.20
pod
<app>
prod
be
1.4
10.0.0.13
pod
<app>
prod
be
1.3
10.0.0.14
pod
<app>
prod
be
1.3
10.0.0.21
pod
<app>
prod
be
1.4

Only sends traffic to healthy Pods

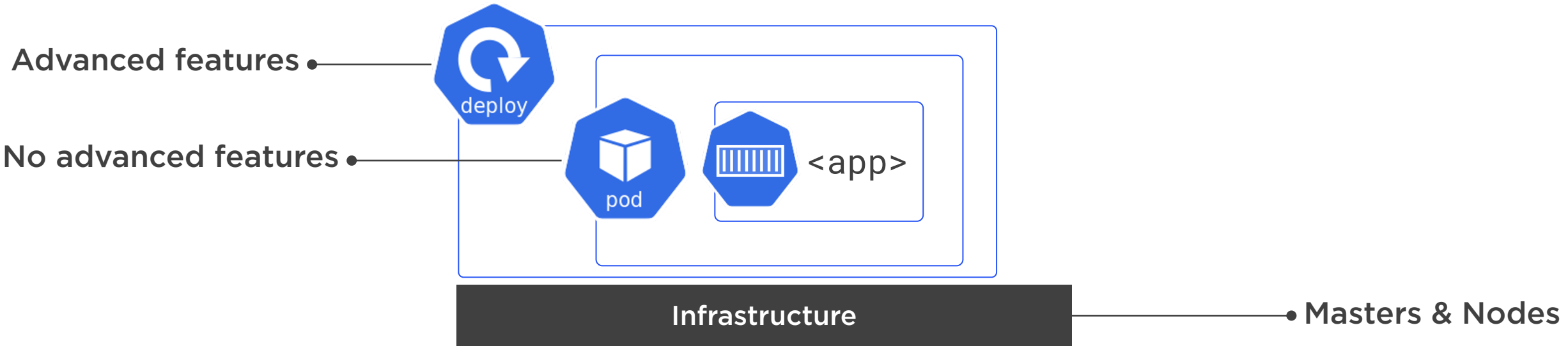
Can do session affinity

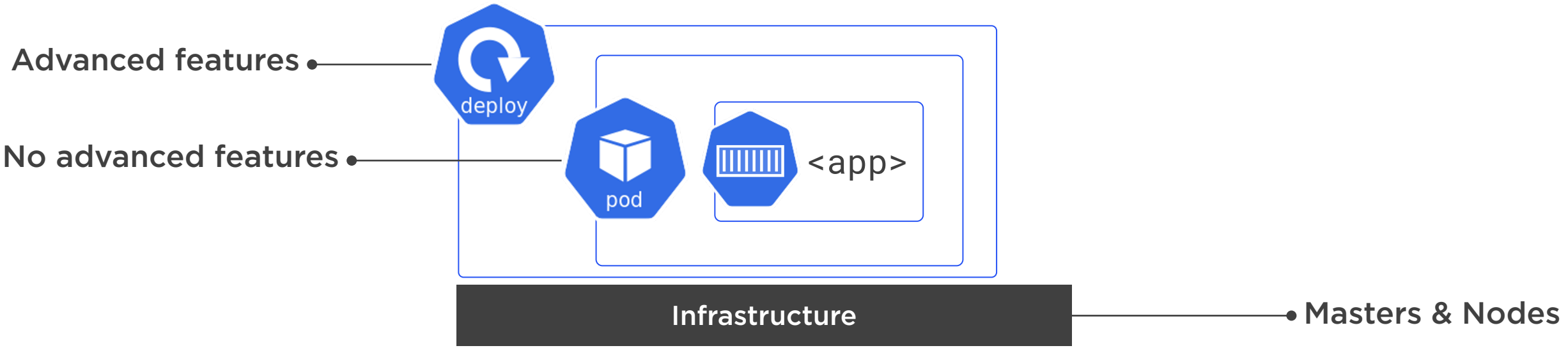
Can send traffic to endpoints outside the cluster

Can do TCP and UDP

Up Next:
Game-changing Deployments

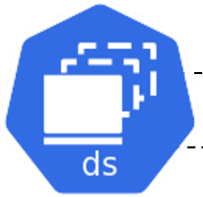
Game-changing Deployments


Advanced features
deploy
No advanced features
pod
<app>
Infrastructure
Masters & Nodes


Advanced features
deploy
No advanced features
pod
<app>
Infrastructure
Masters & Nodes

Stateless apps

Stateful apps...

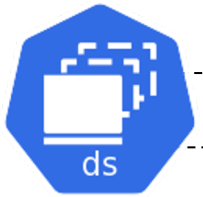


One instance on every node

Time-based short-lived jobs

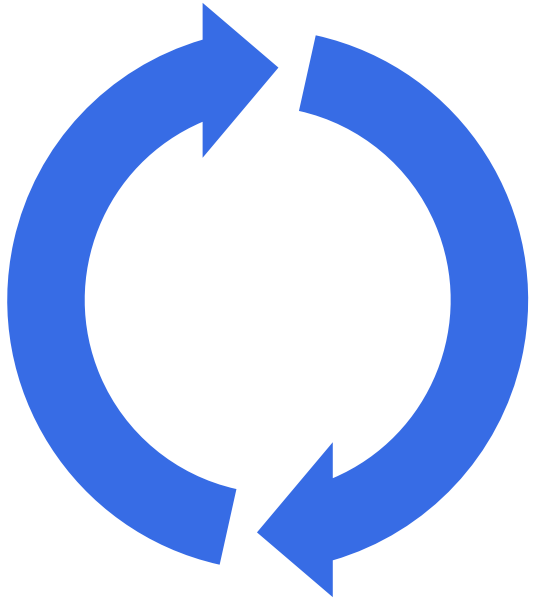
Stateless apps

Stateful apps...



One instance on every node

Time-based short-lived jobs

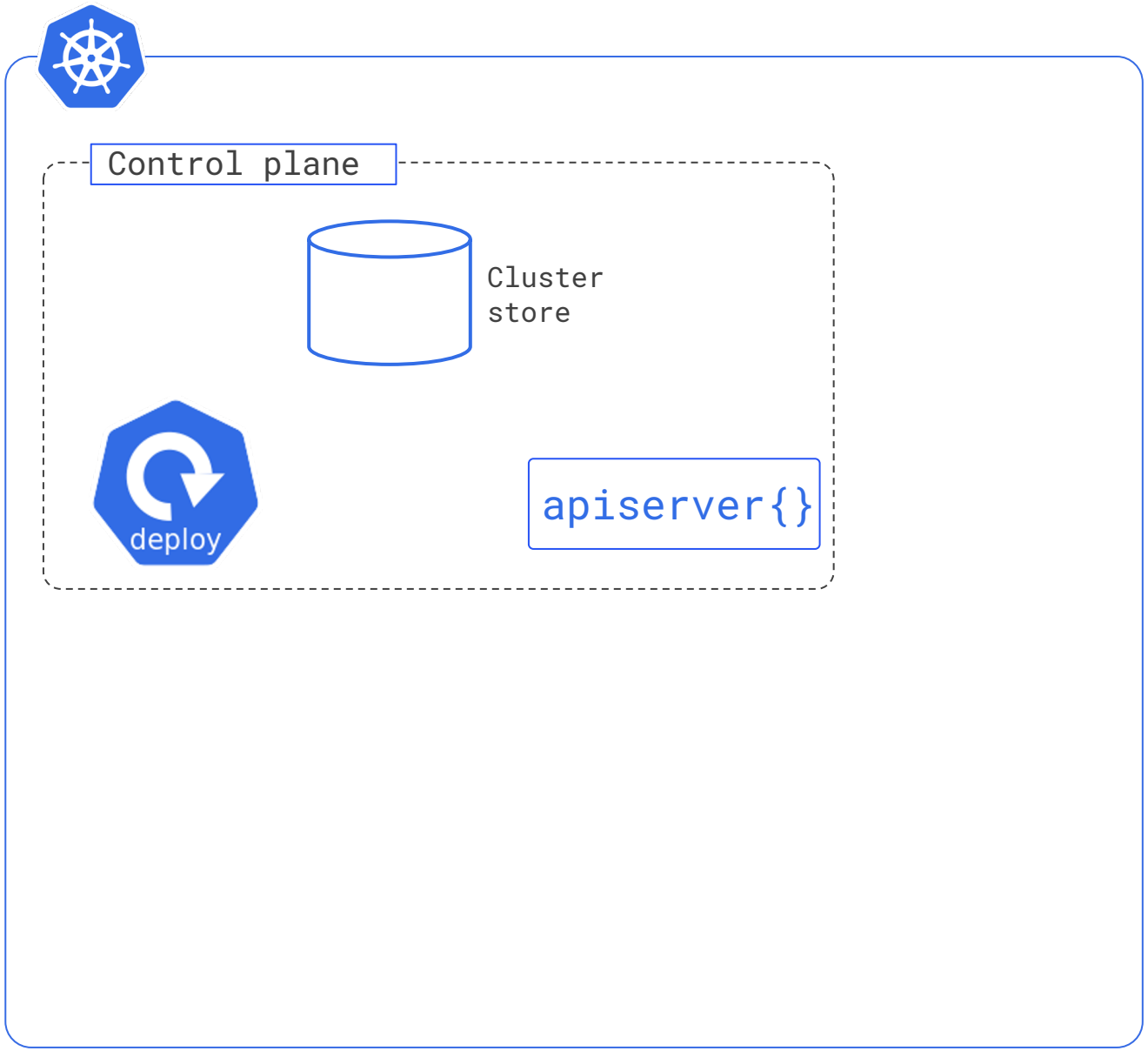


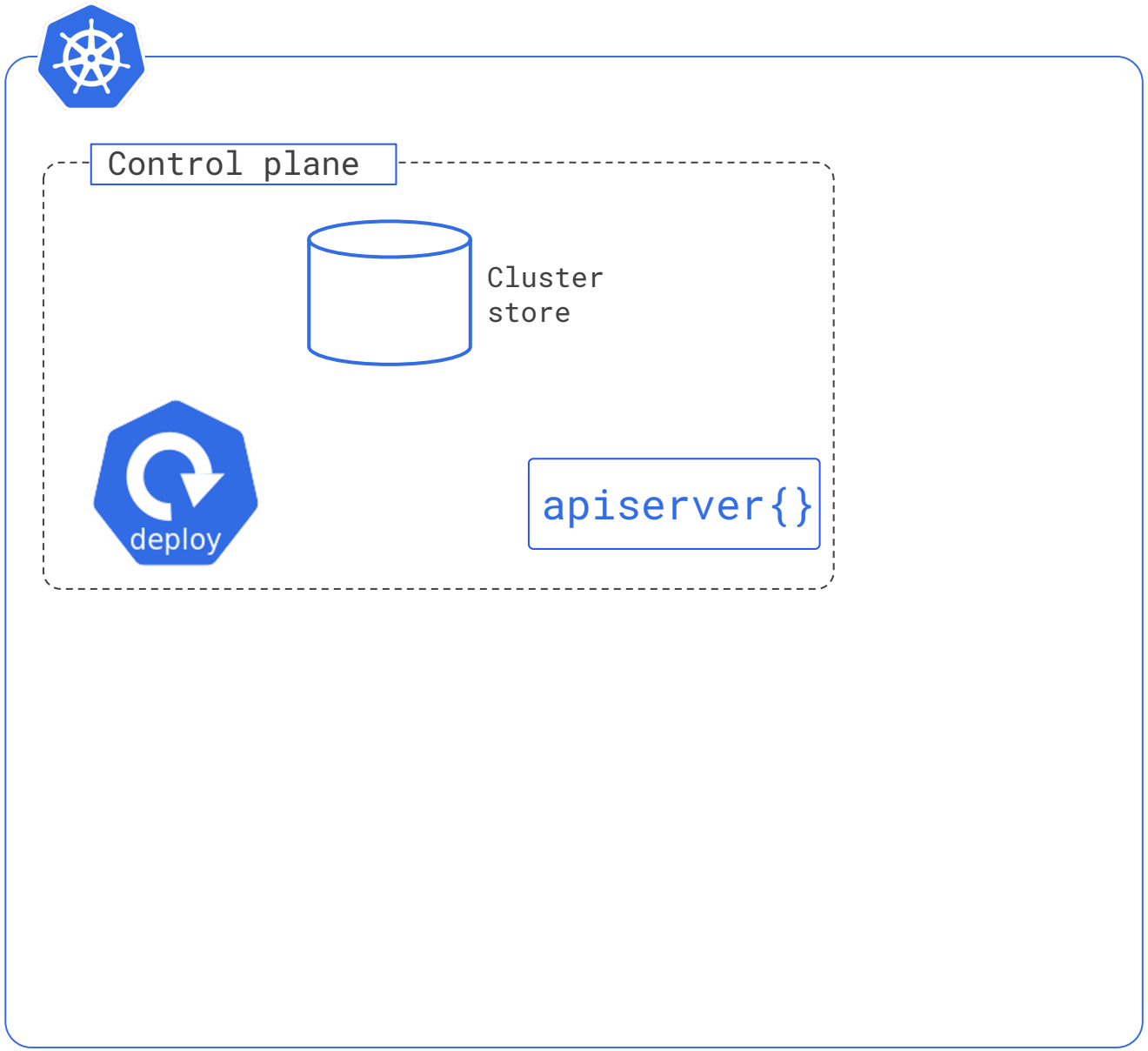
Deployment Controller/Reconciliation loop

Watches API Server for new Deployments

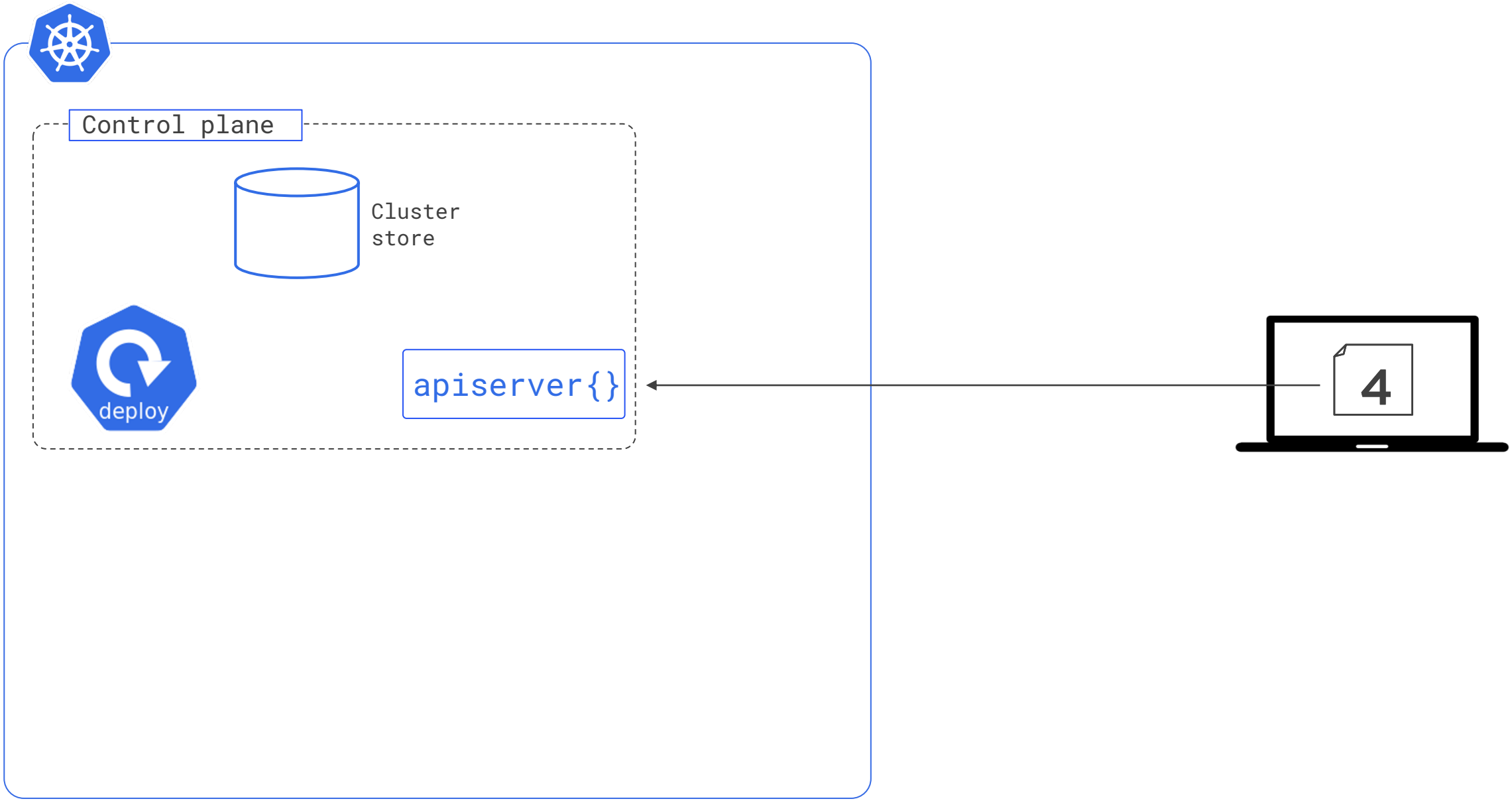
Implements them

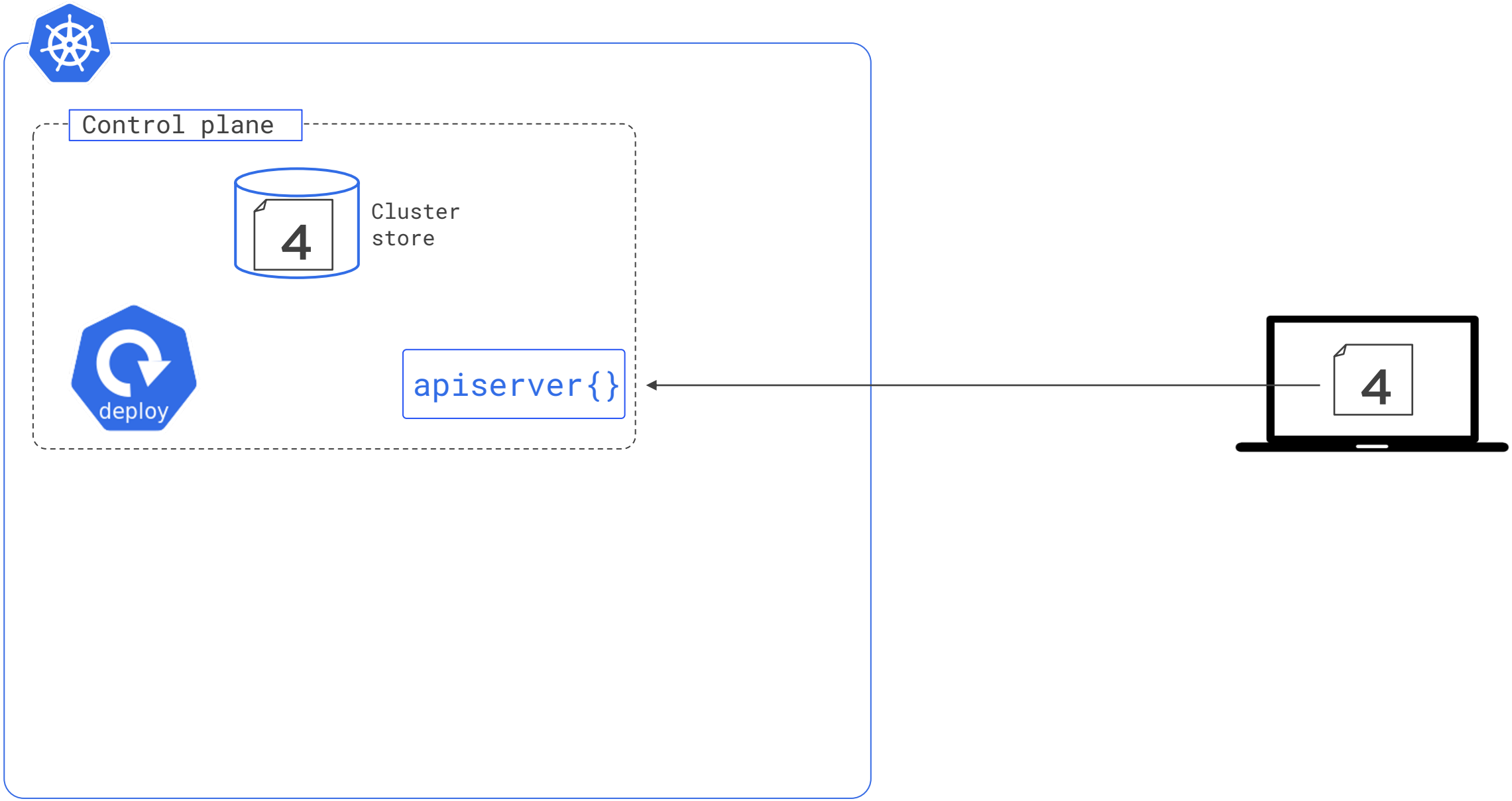
Constantly compares ***observed state*** with ***desired state***

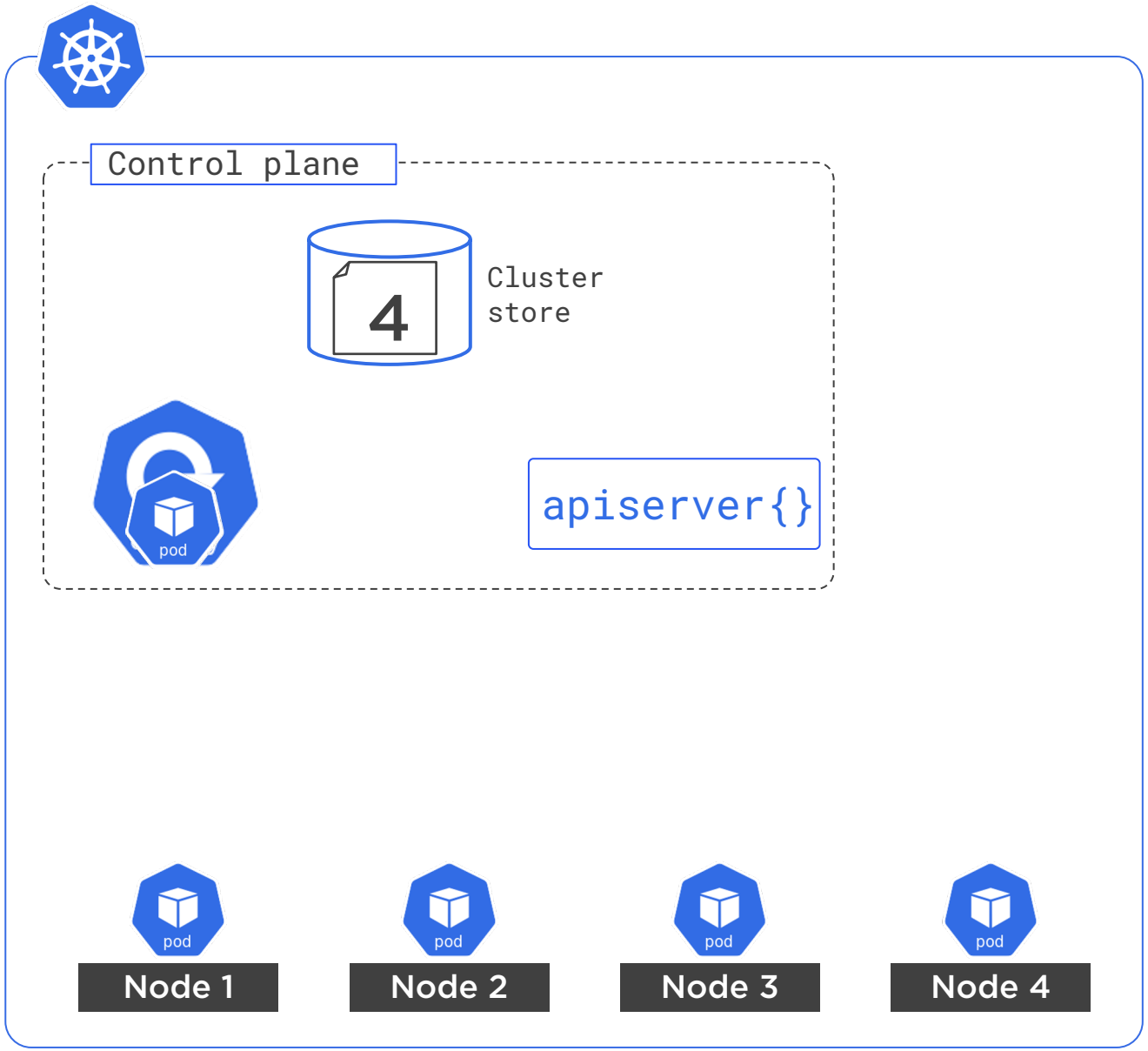

Control plane
Cluster
store
deploy
apiserver{}


Control plane
Cluster
store
deploy
apiserver{}

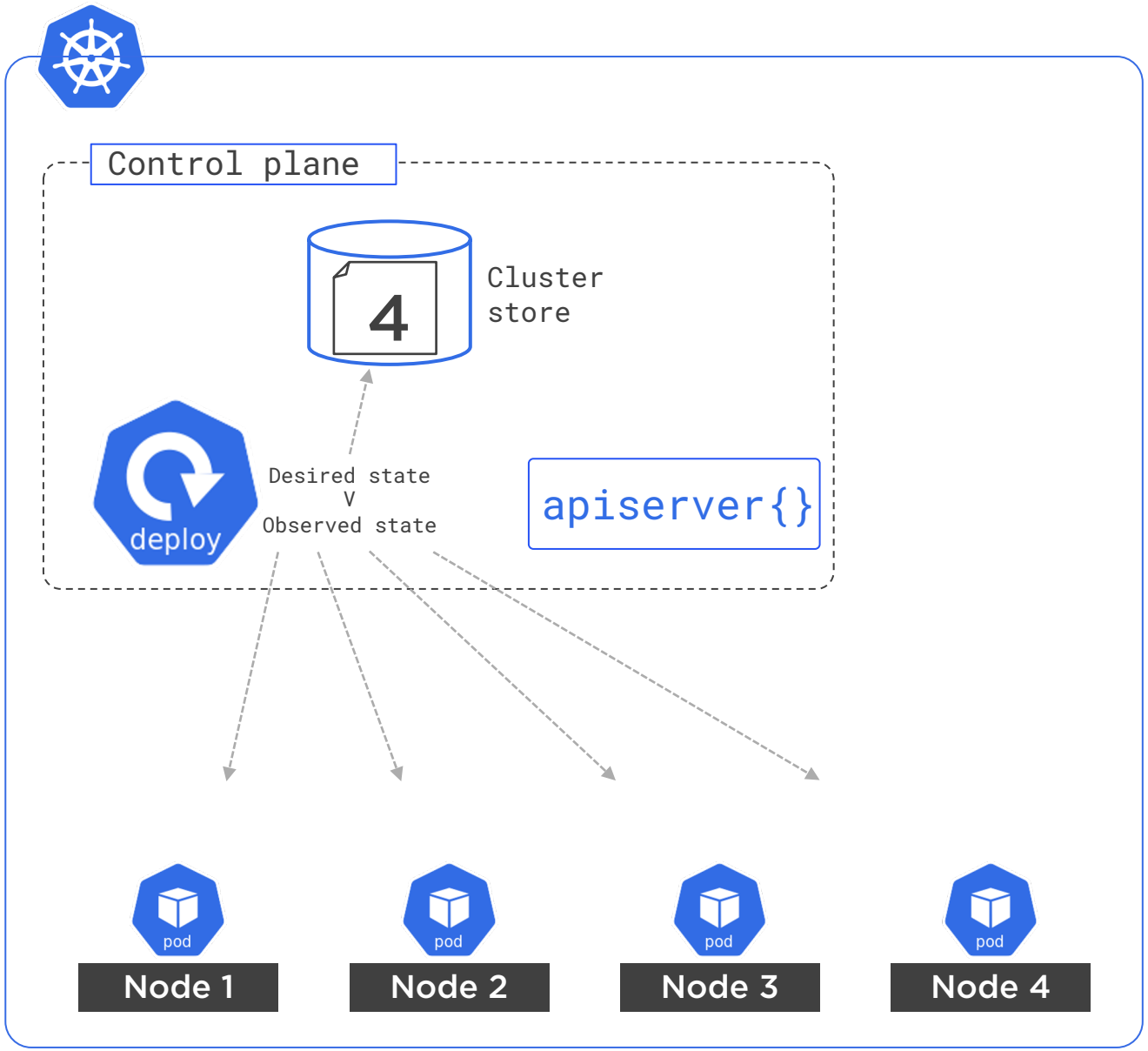

4


Control plane
Cluster
store
deploy
apiserver{}
4

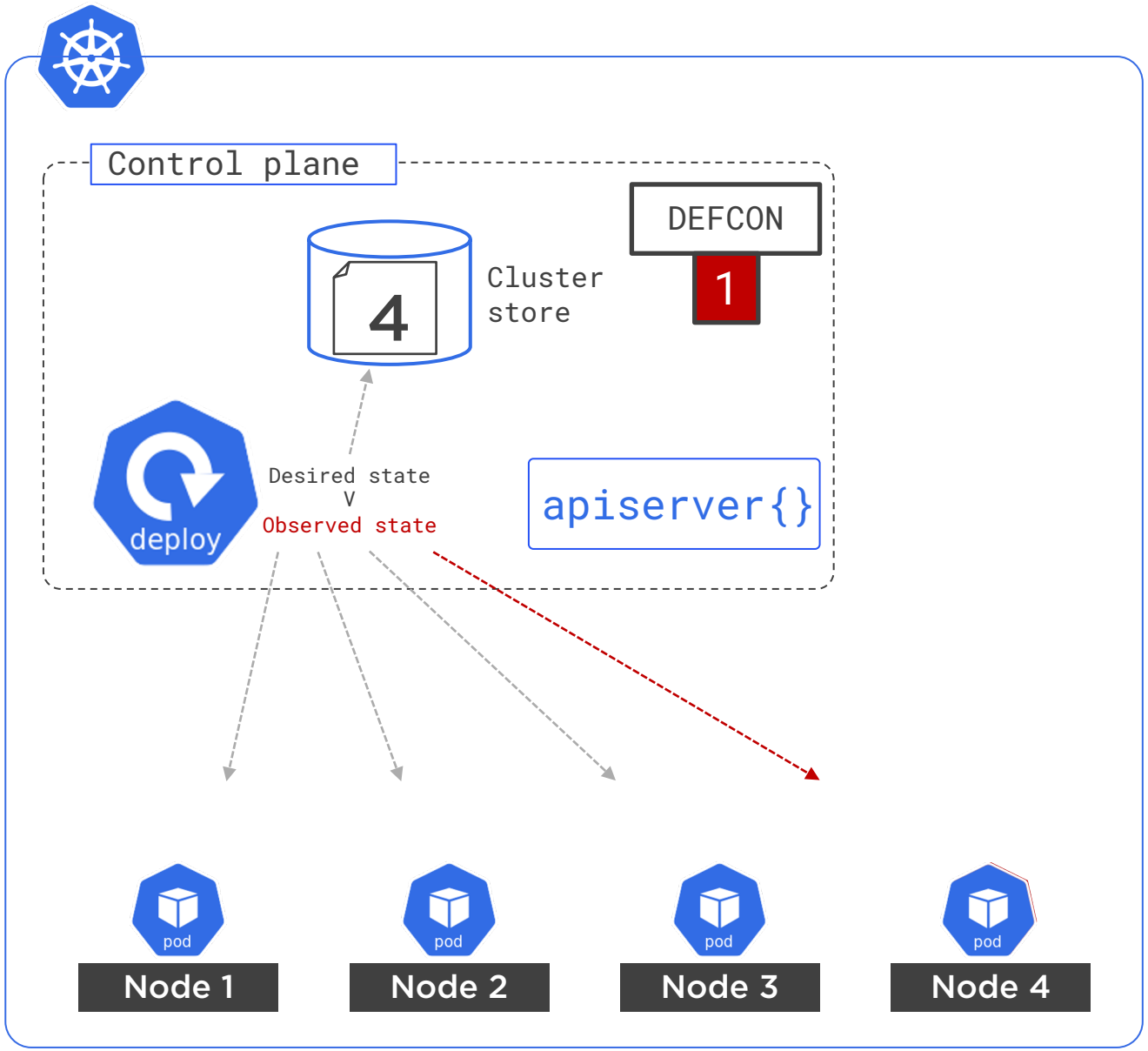

Control plane
4
Cluster
store
deploy
apiserver{}
4


Control plane
4
Cluster
store
pod
apiserver{}
pod
Node 1
pod
Node 2
pod
Node 3
pod
Node 4

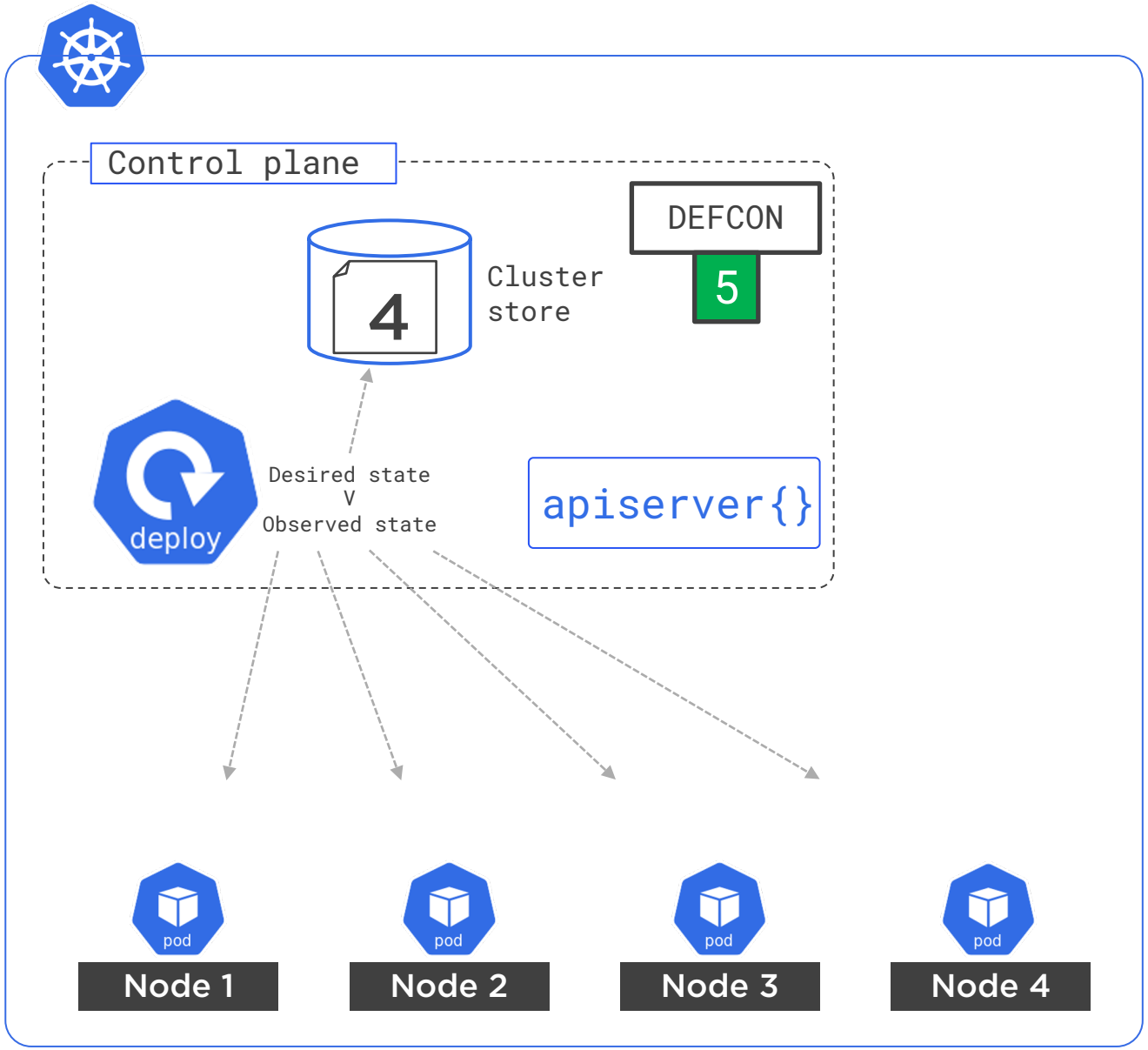

4

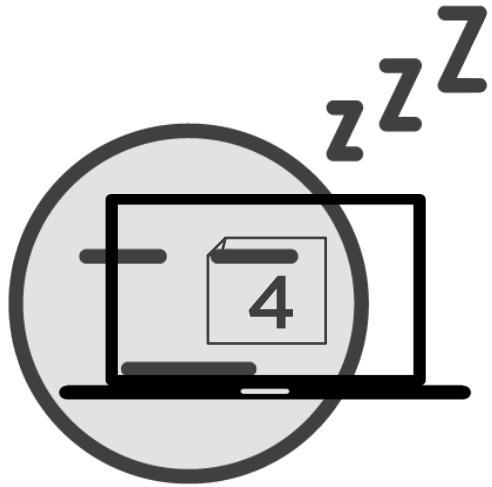

Control plane
4
Cluster
store
deploy
Desired state
V
Observed state
apiserver{}
pod
pod
pod
pod
Node 1
Node 2
Node 3
Node 4


4

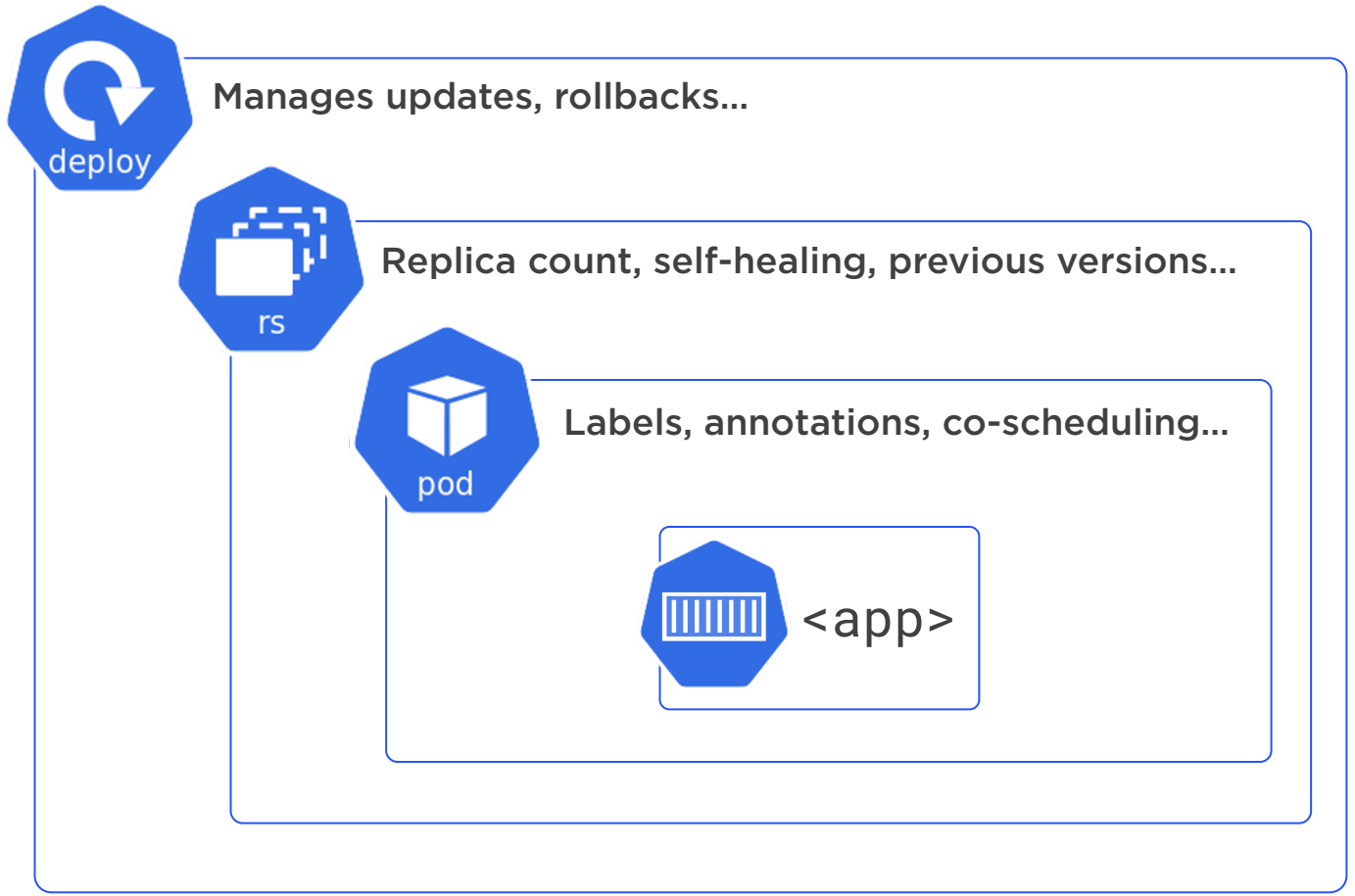

Control plane
DEFCON
1
4
Cluster
store
deploy
Desired state
V
Observed state
apiserver{}
pod
pod
pod
pod
Node 1
Node 2
Node 3
Node 4

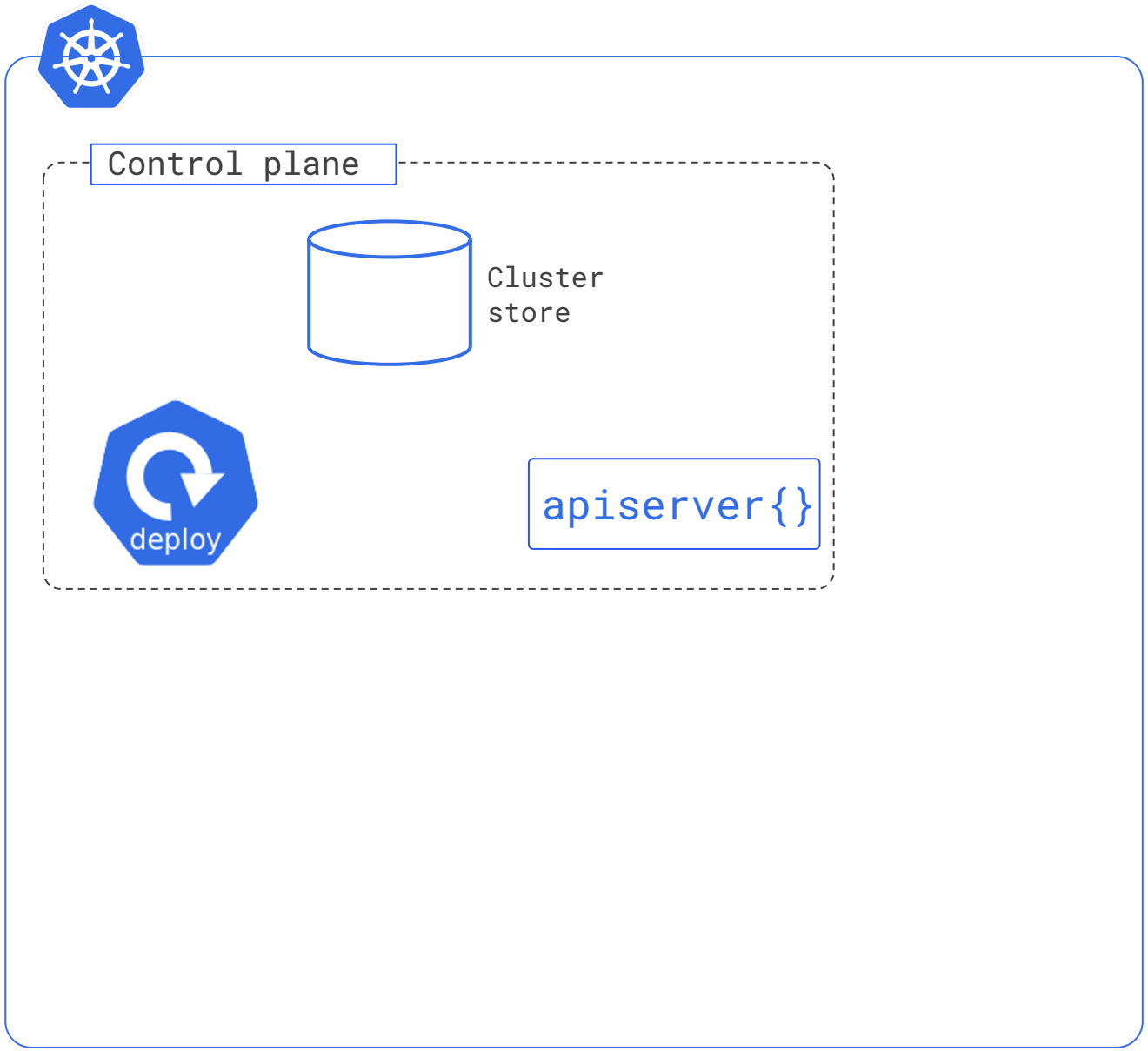

4

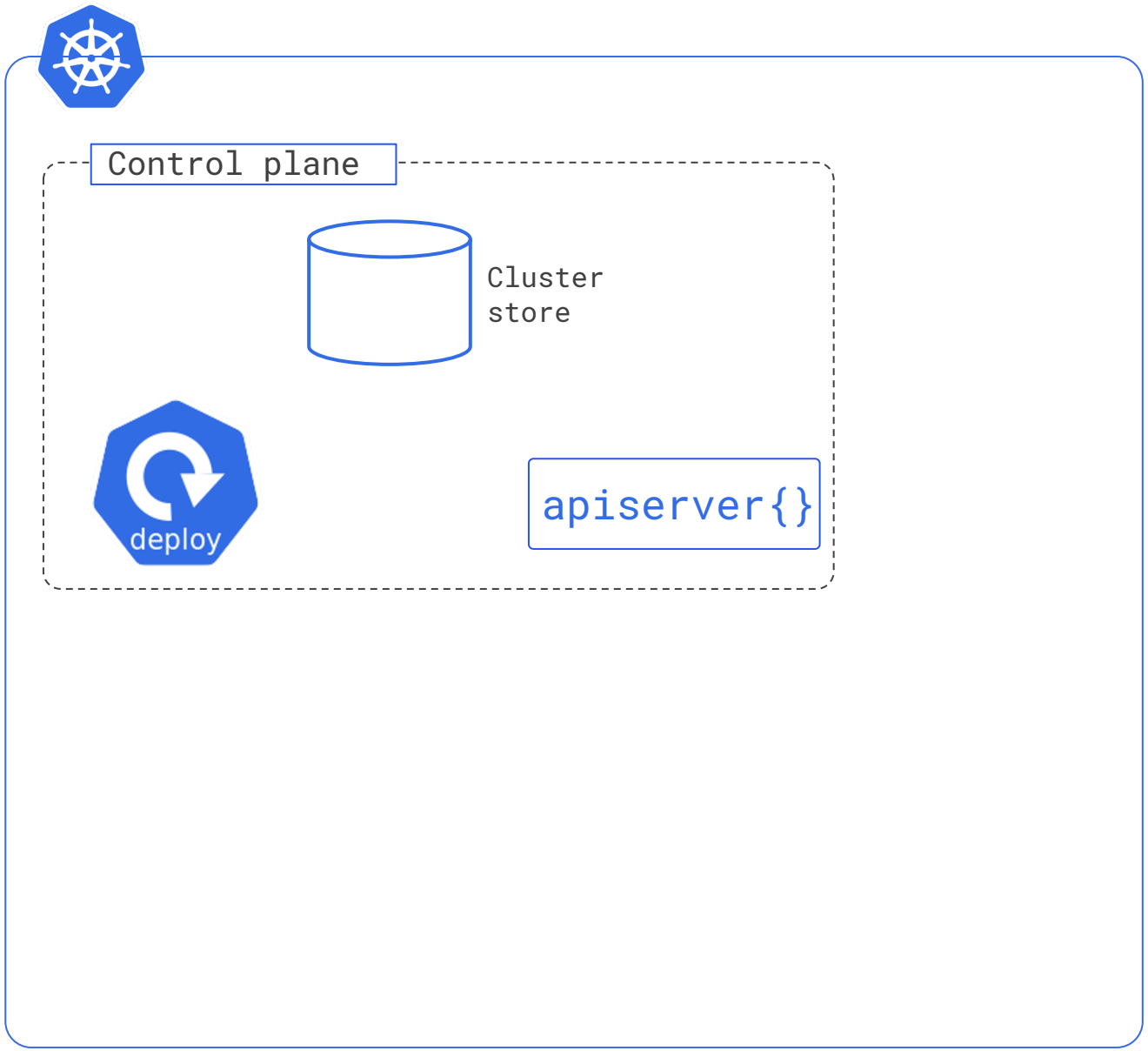

Control plane
4
Cluster
store
DEFCON
5
deploy
Desired state
V
Observed state
apiserver{}
pod
Node 1
pod
Node 2
pod
Node 3
pod
Node 4


4

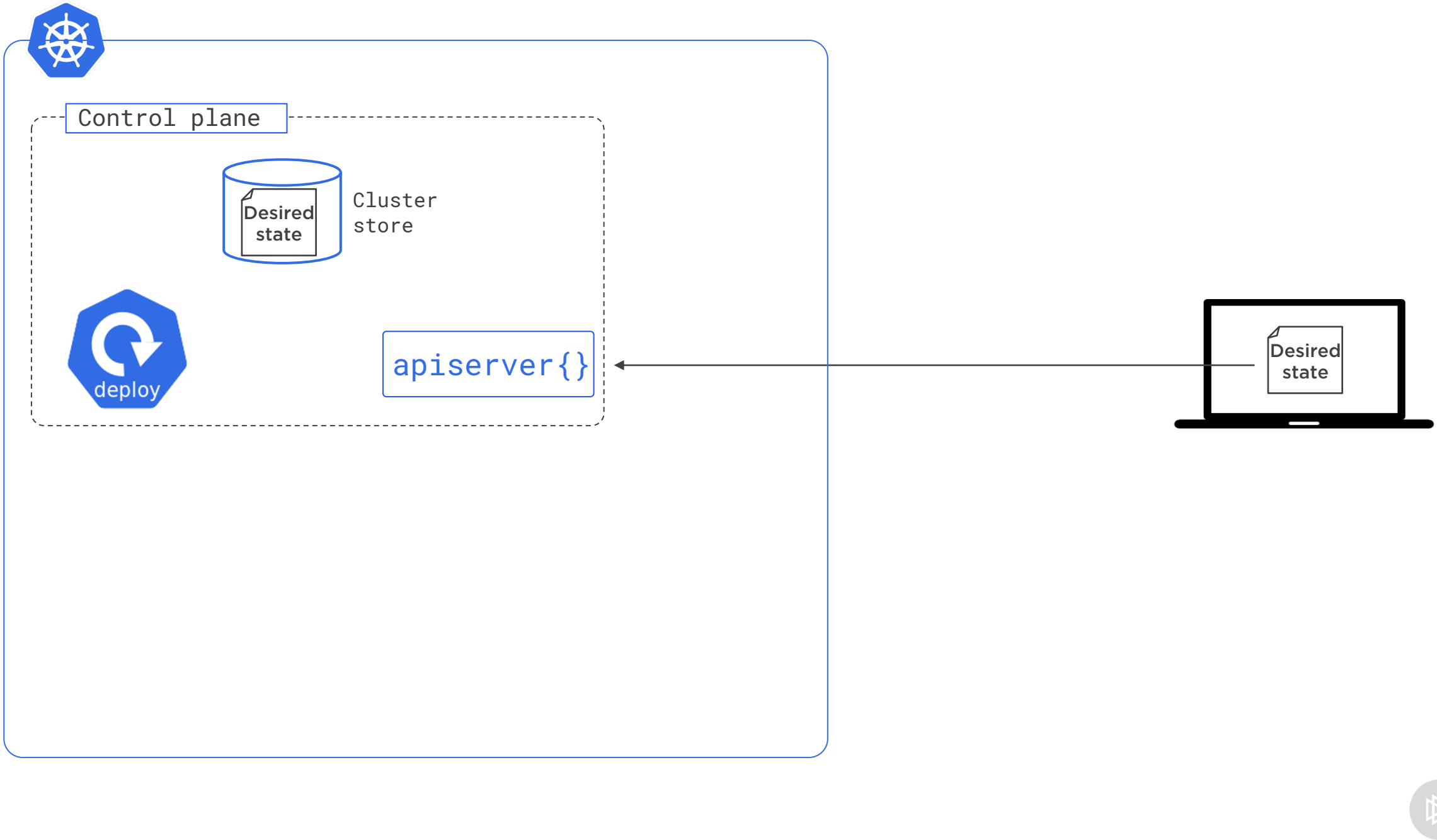
You/me

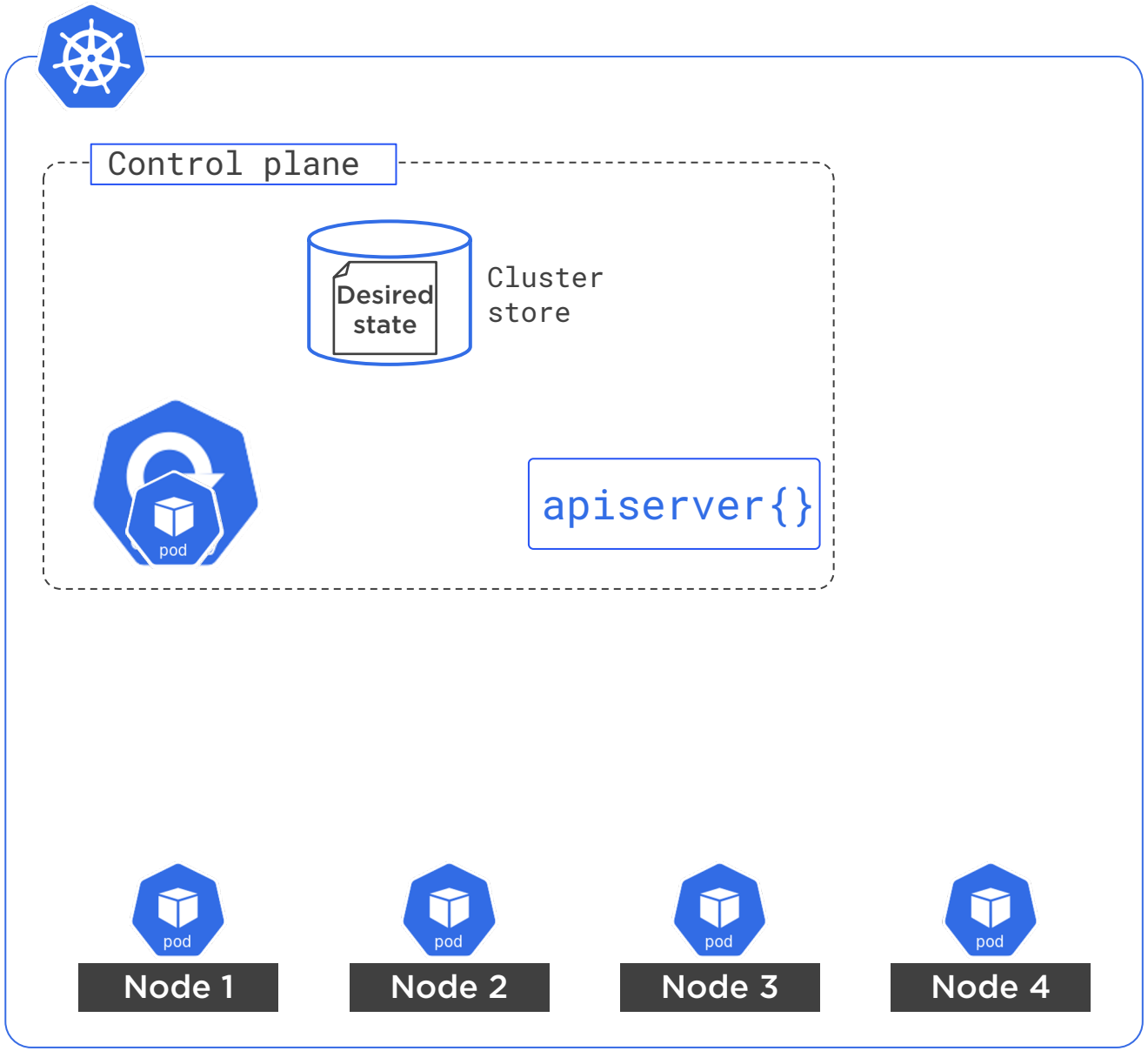

deploy
Manages updates, rollbacks...
rs
Replica count, self-healing, previous versions...
pod
Labels, annotations, co-scheduling...
<app>


Control plane
Cluster
store
deploy
apiserver{}

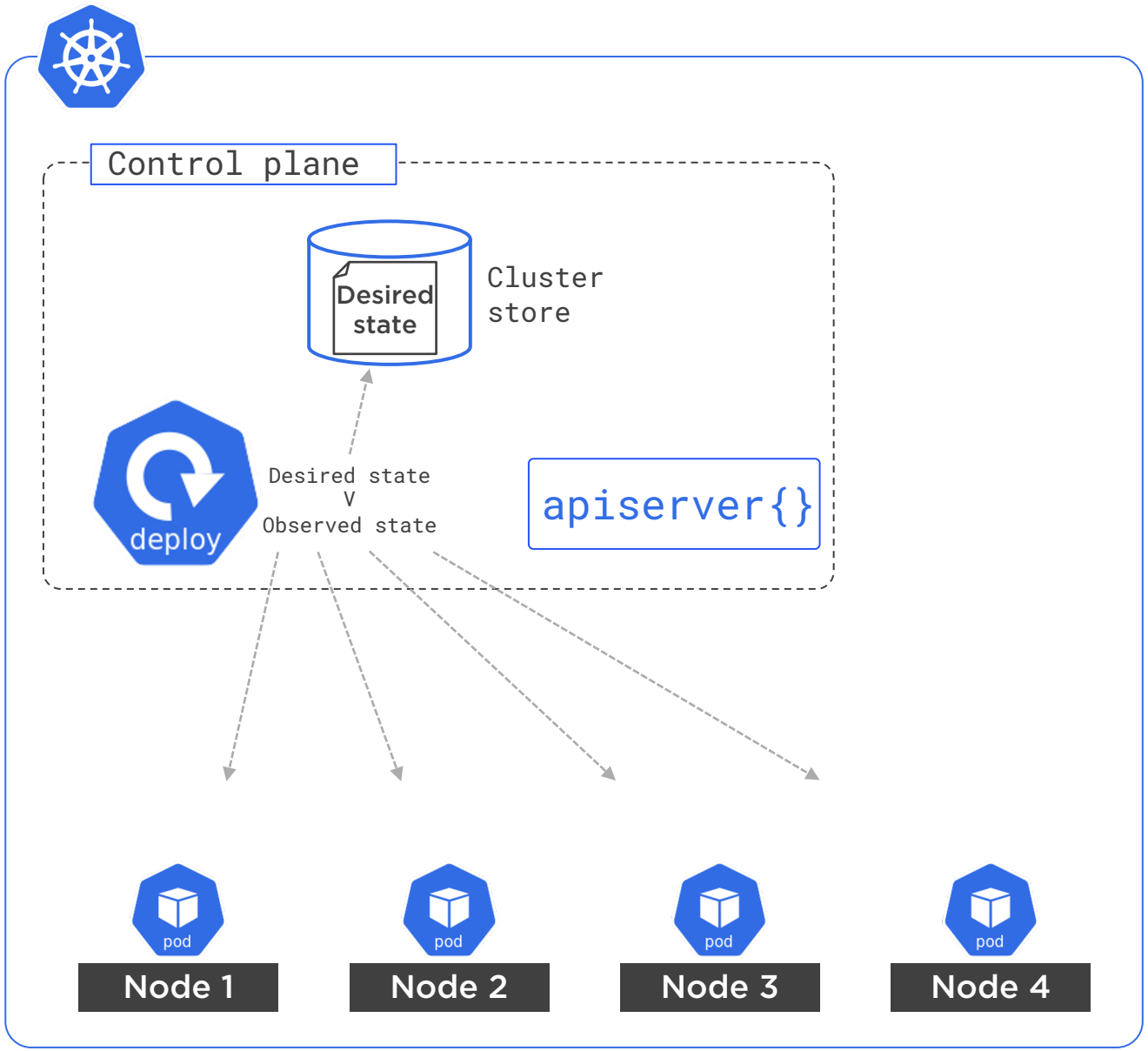

Control plane
Cluster
store
deploy
apiserver{}


Desired
state


Control plane
Desired state
Cluster store
deploy
apiserver{}
Desired state


Control plane
Desired state
Cluster store
pod
apiserver{}
pod
pod
pod
pod
Node 1
Node 2
Node 3
Node 4


Desired state


Control plane
Desired state
Cluster store
deploy
Desired state
V
Observed state
apiserver{}
pod
Node 1
pod
Node 2
pod
Node 3
pod
Node 4


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

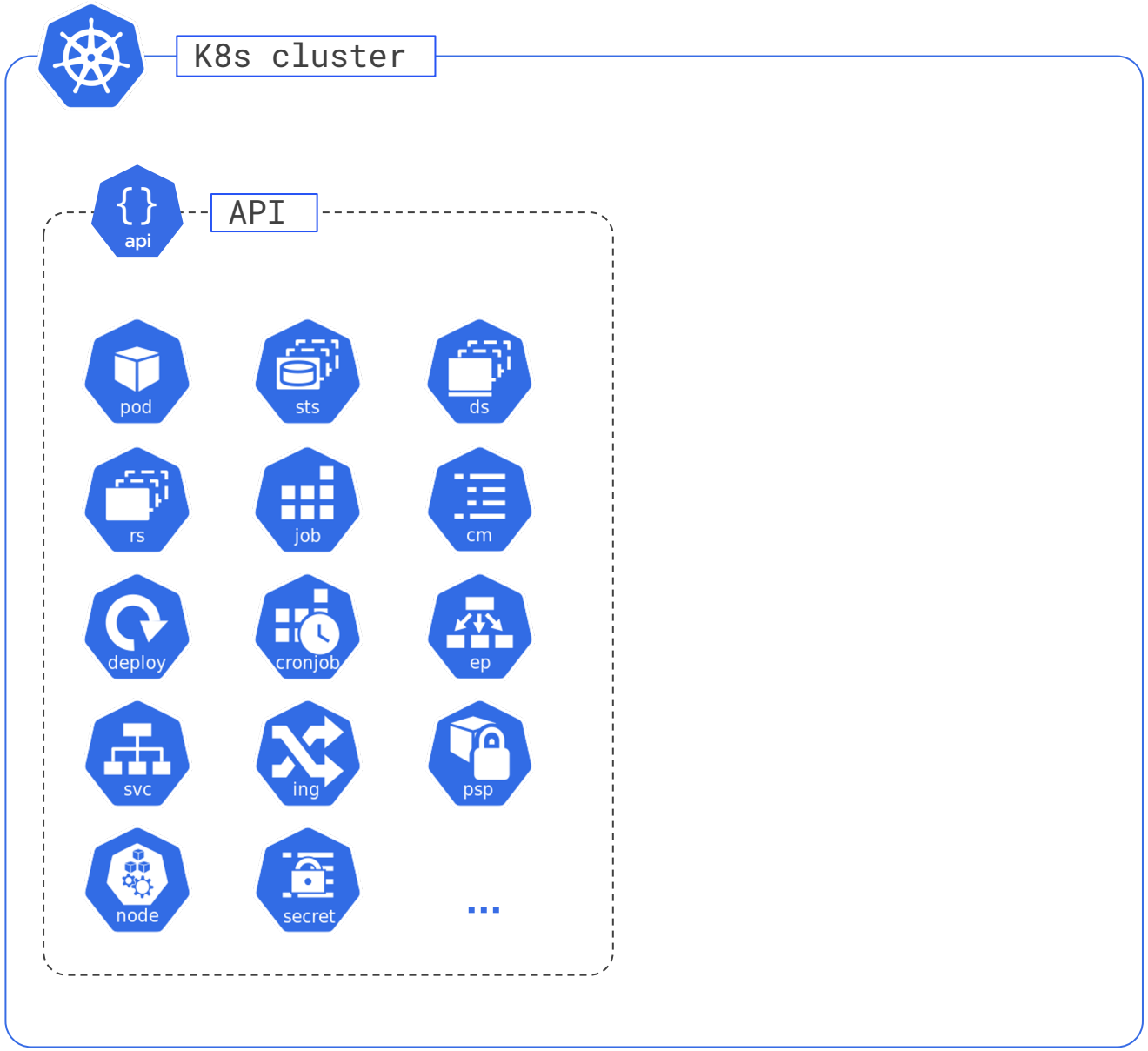
Stable network abstraction

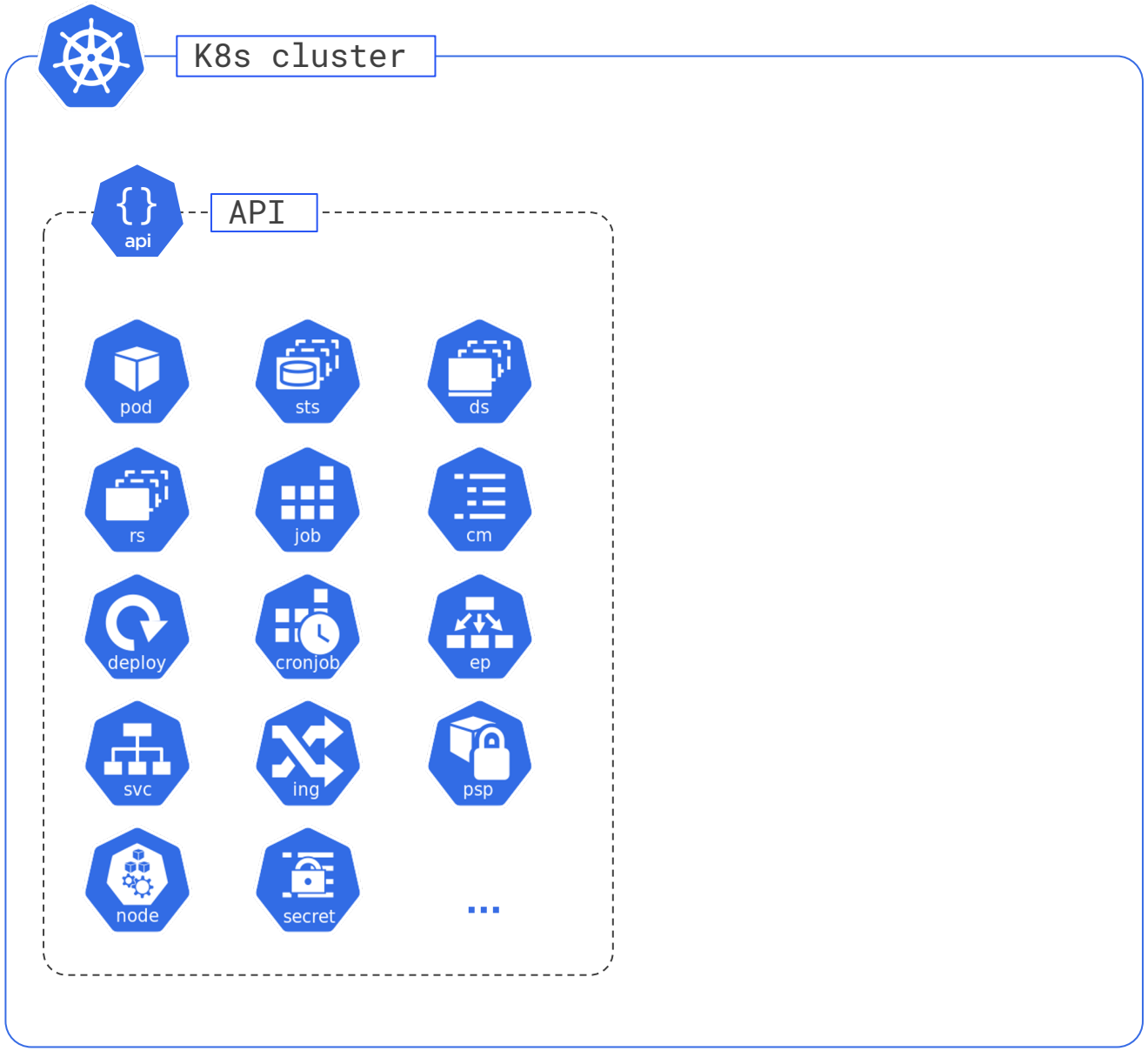
Atomic unit of scheduling

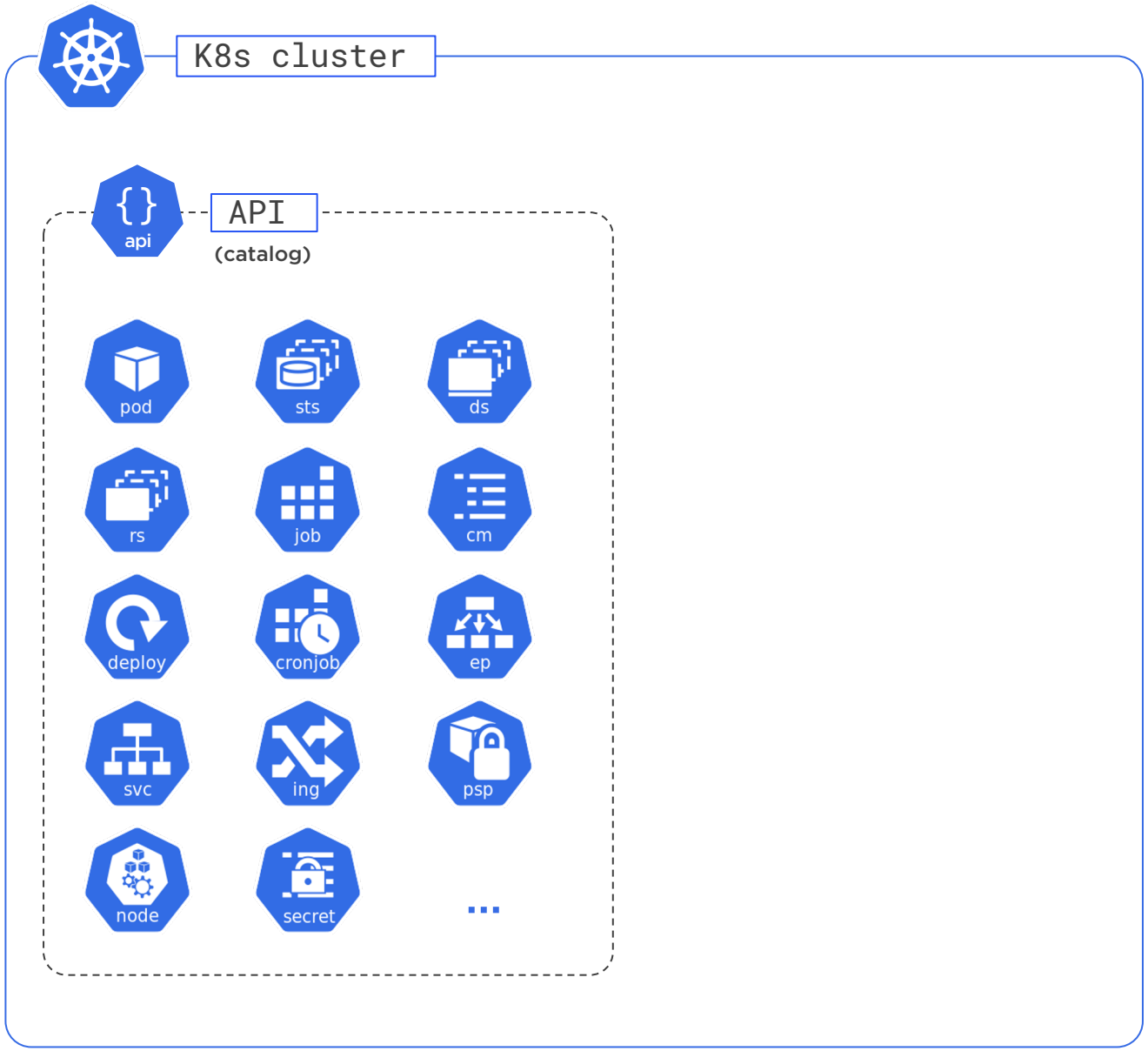
Replica count

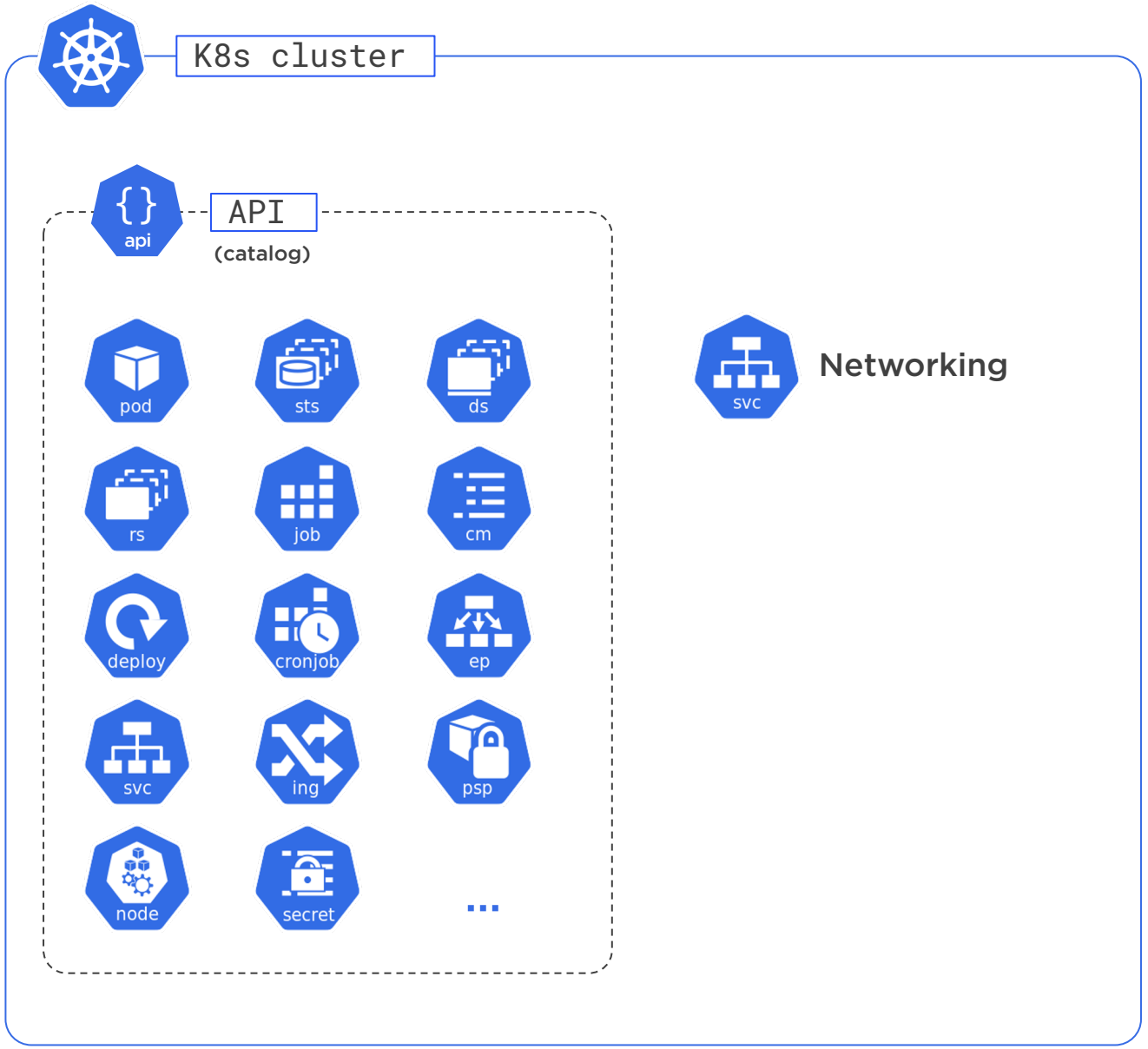
Updates and rollbacks

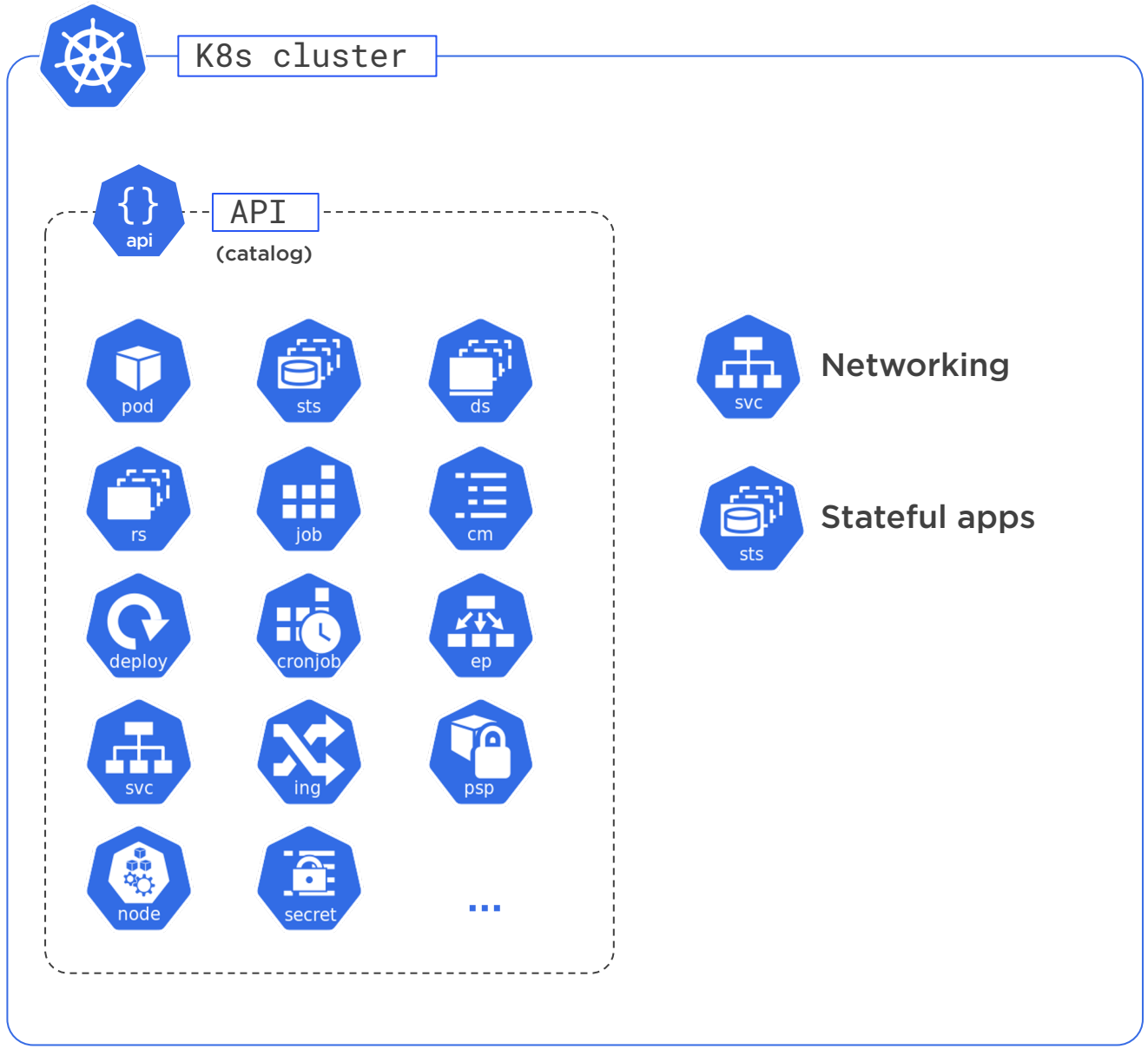
Stable network abstraction

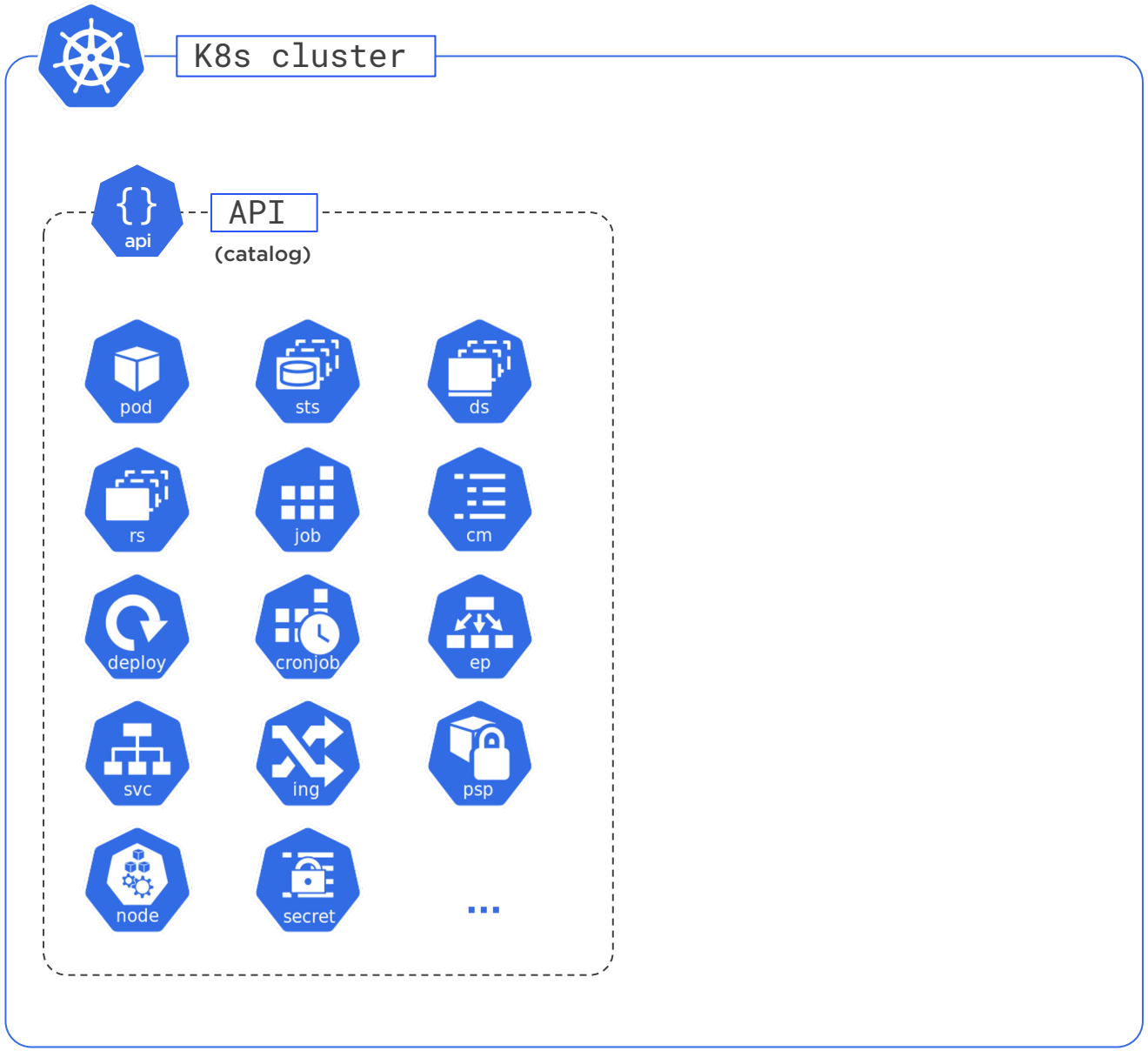

K8s cluster
api
API
pod
sts
ds
rs
job
cm
deploy
cronjob
ep
svc
ing
psp
node
secret
...

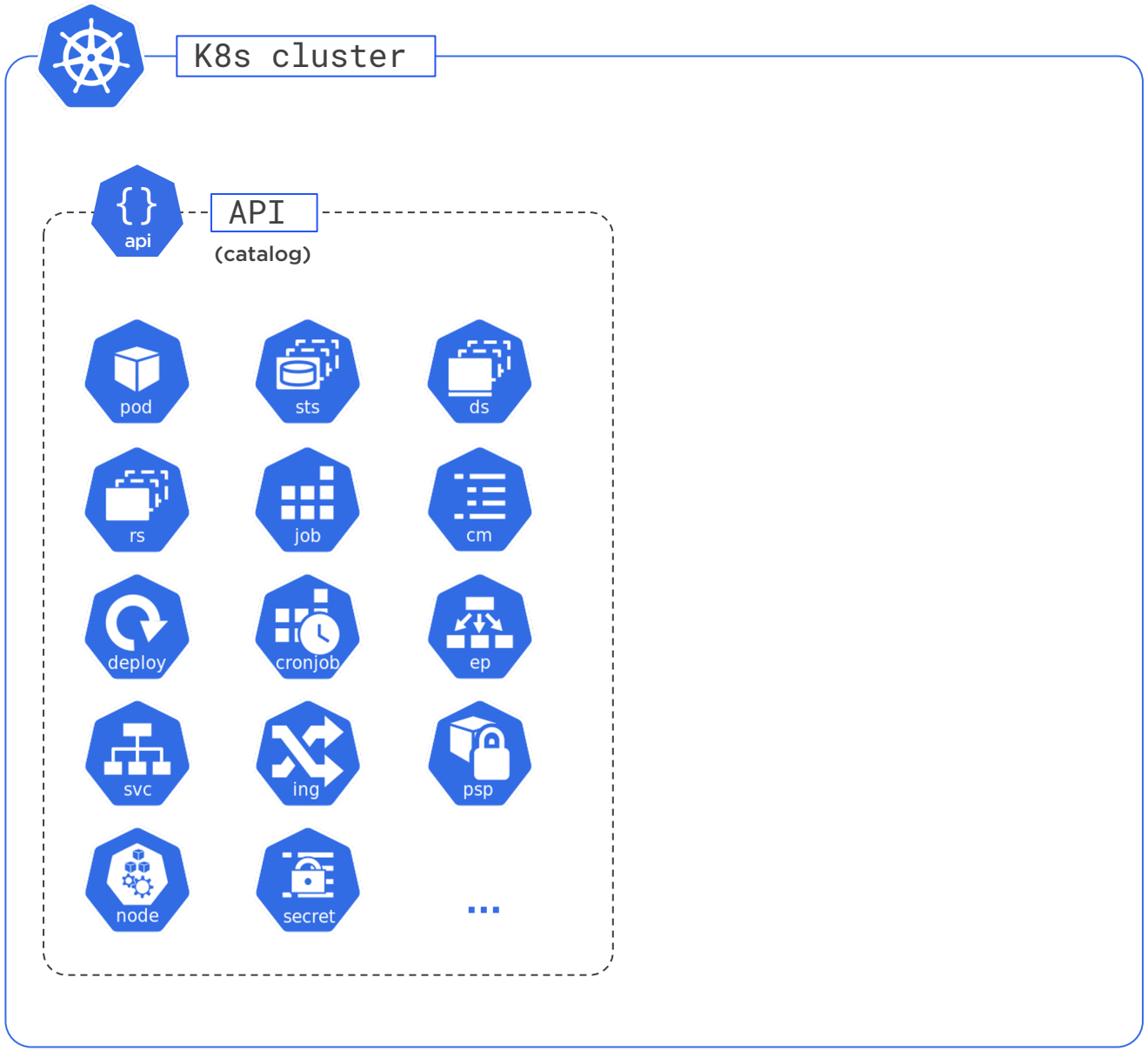

K8s cluster
api
API
pod
sts
ds
rs
job
cm
deploy
cronjob
ep
svc
ing
psp
node
secret
...

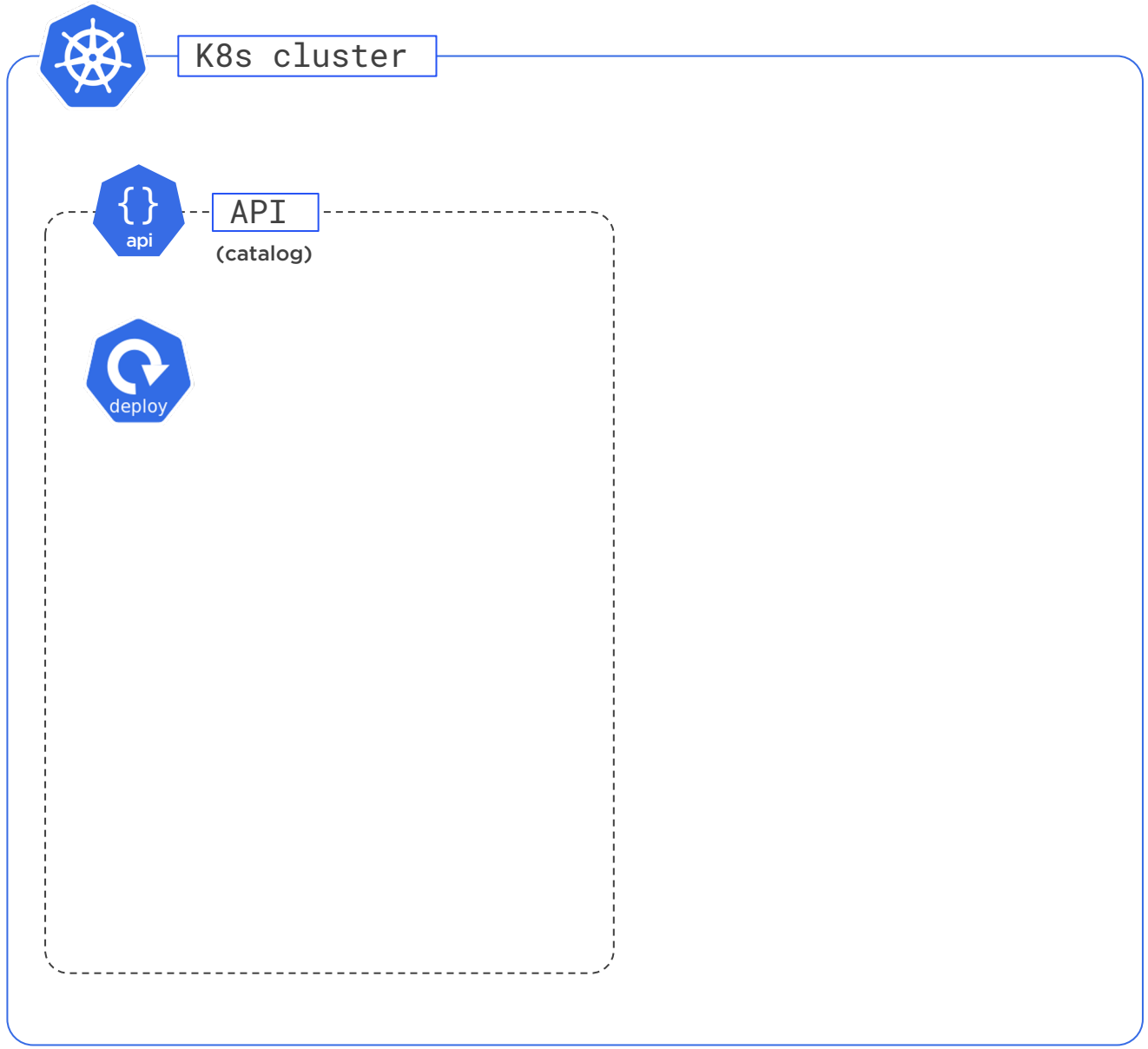

K8s cluster
api
API
(catalog)
pod
sts
ds
rs
job
cm
deploy
cronjob
ep
svc
ing
psp
node
secret
...


K8s cluster
api
API
(catalog)
pod
sts
ds
rs
job
cm
deploy
cronjob
ep
svc
ing
psp
node
secret
...
svc
Networking

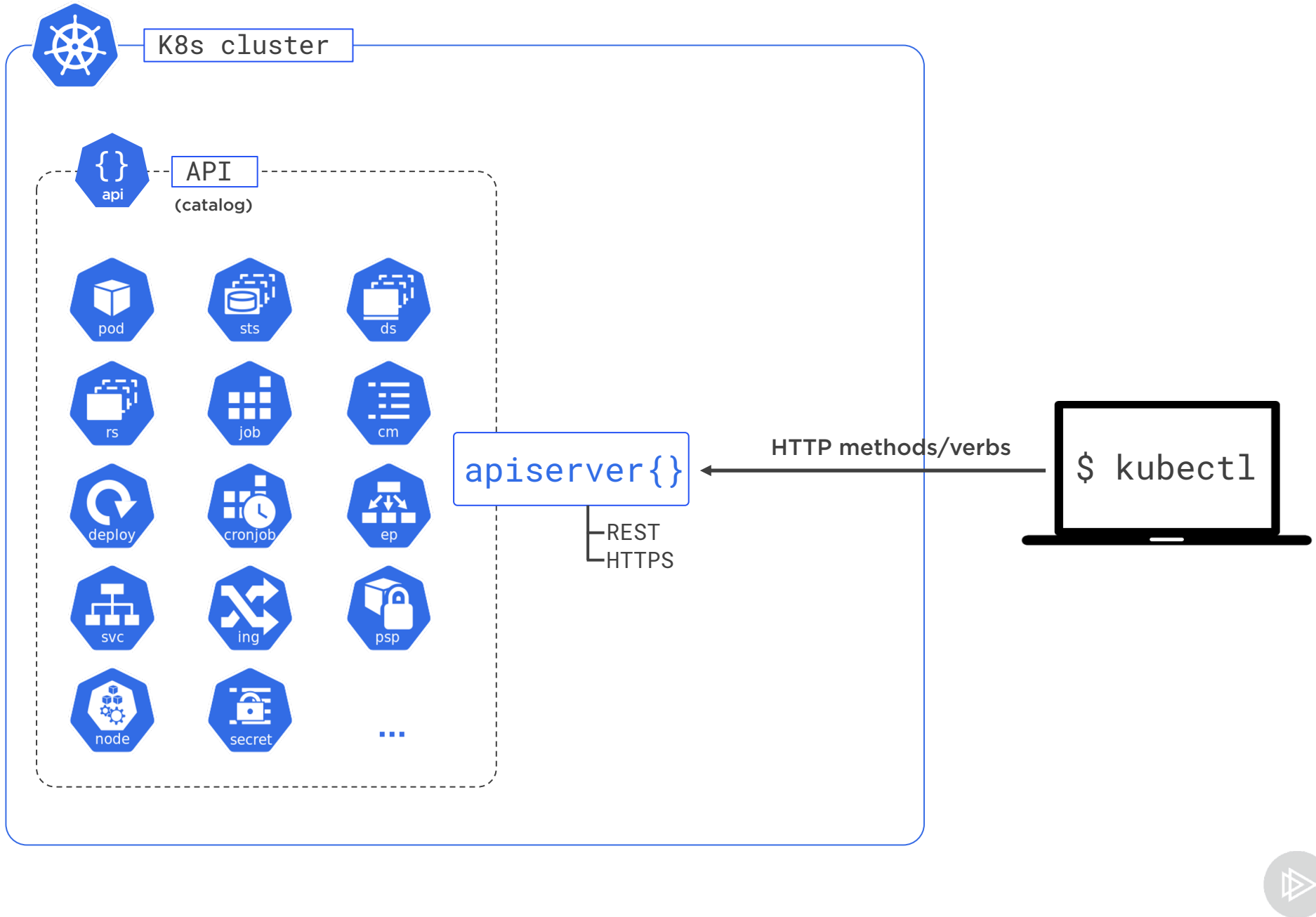

K8s cluster
api
API
(catalog)
pod
sts
ds
rs
job
cm
deploy
cronjob
ep
svc
ing
psp
node
secret
...
svc
Networking
sts
Stateful apps

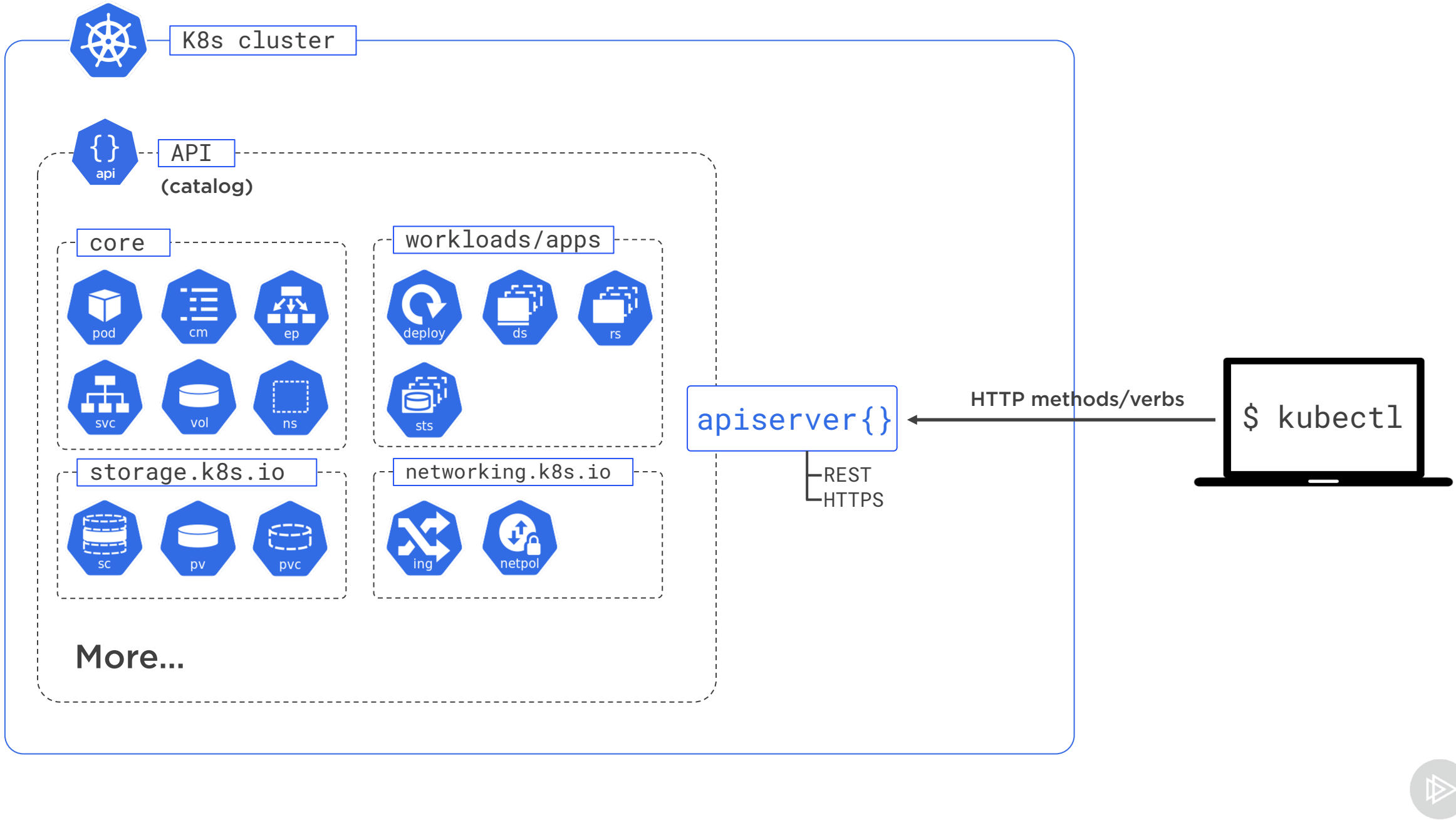

K8s cluster
api
API
(catalog)
pod
sts
ds
rs
job
cm
deploy
cronjob
ep
svc
ing
psp
node
secret
...


K8s cluster
api
API
(catalog)
pod
sts
ds
rs
job
cm
deploy
cronjob
ep
svc
ing
psp
node
secret
...


K8s cluster
{}
api
API
(catalog)
deploy

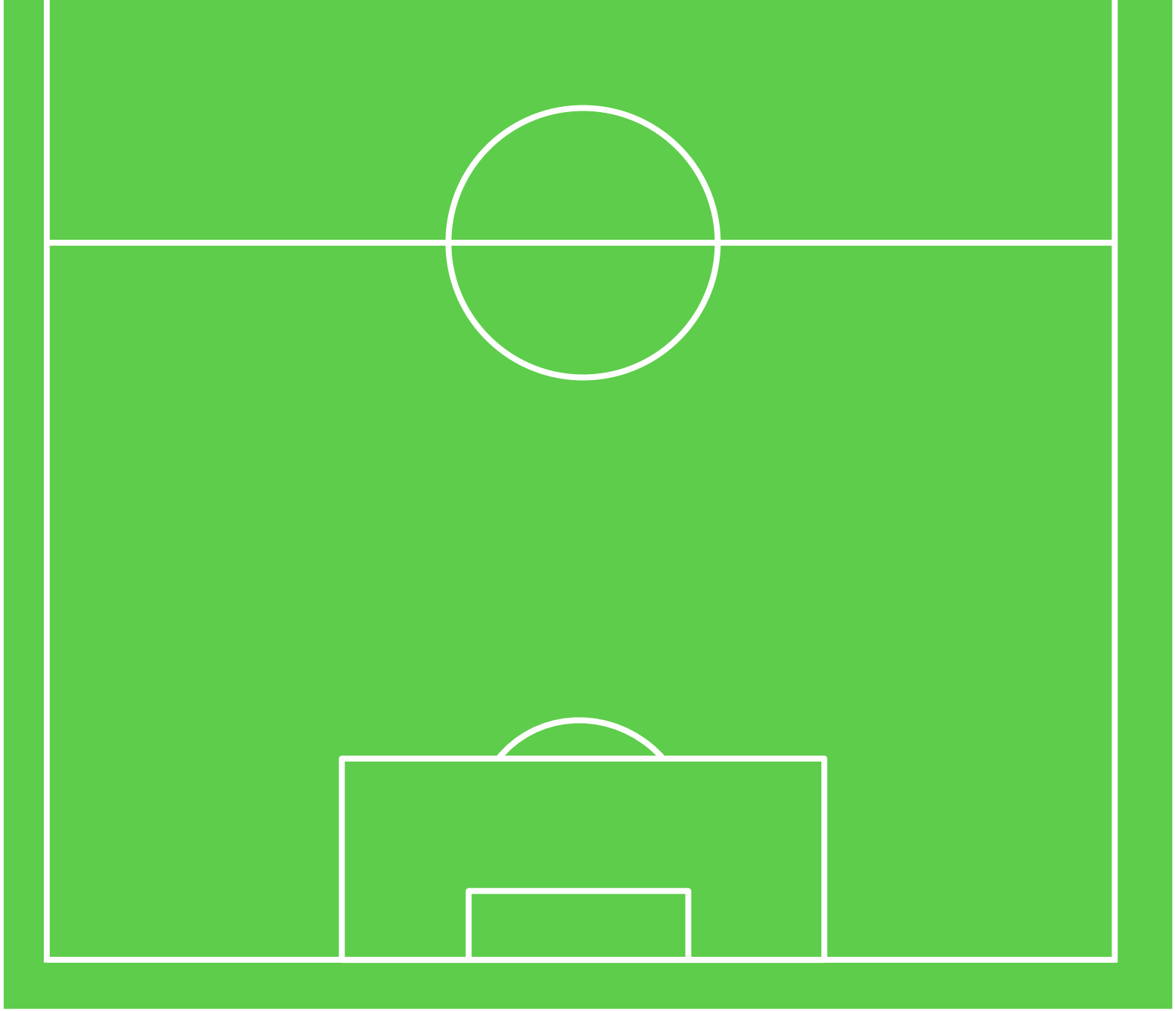

K8s cluster
{ }
api
API
(catalog)
deploy
Deployment object definition
• API sub-group: [apps/v1]
• replicas (integer)
• minReadySeconds (integer)
• progressDeadlineSeconds (integer)
• paused (boolean)
• revisionHistoryLimit (integer)
• selector (LabelSelector)
• strategy (DeploymentStrategy)
• template (PodTemplateSpec)


K8s cluster
api
API
(catalog)
pod
sts
ds
rs
job
cm
deploy
cronjob
ep
svc
ing
psp
node
secret
...
apiserver{}
REST
HTTPS
HTTP methods/verbs
$ kubectl

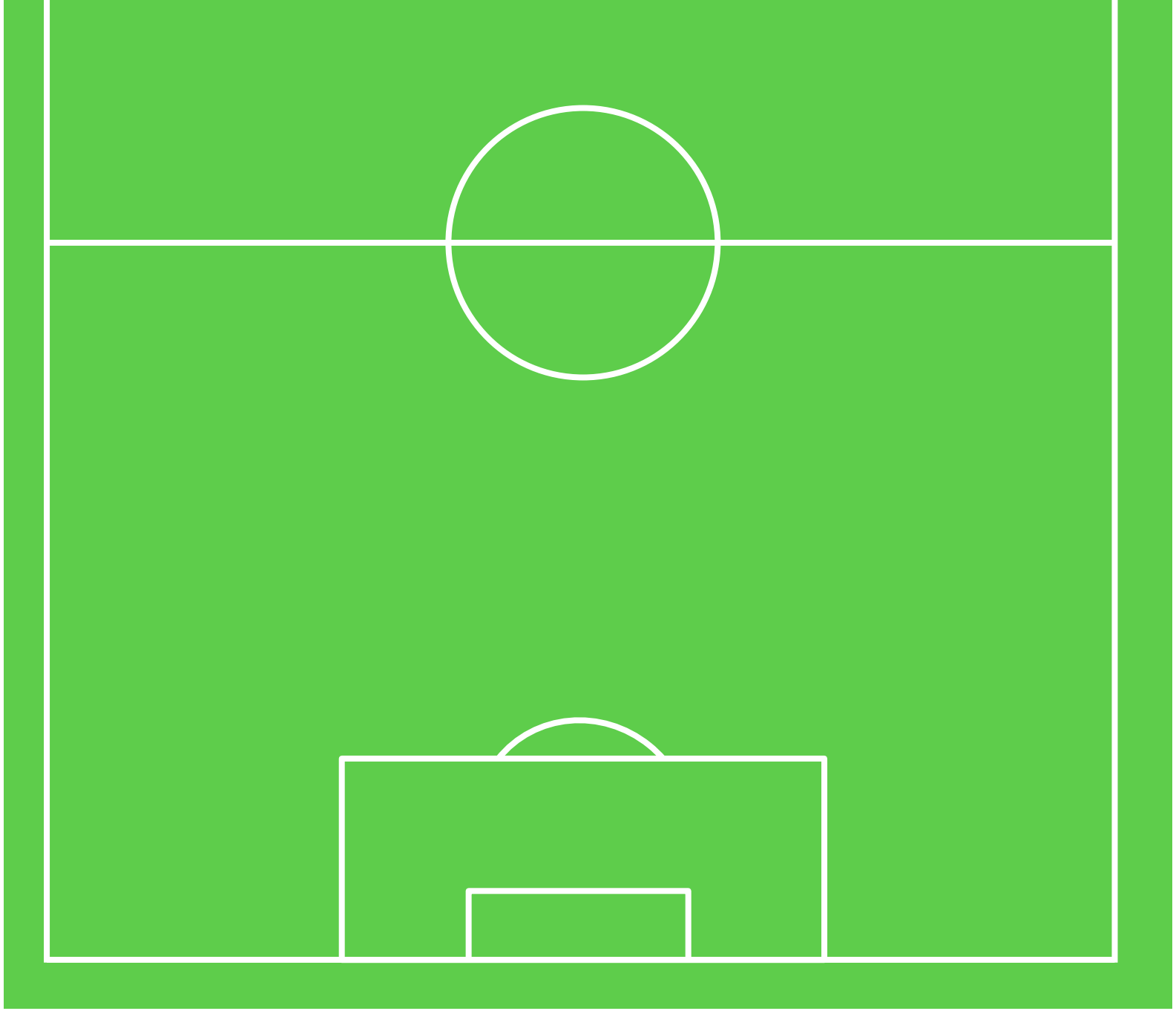

K8s cluster
API
(catalog)
core
pod
cm
ep
svc
vol
ns
workloads/apps
deploy
ds
rs
sts
storage.k8s.io
sc
pv
pvc
networking.k8s.io
ing
netpol
More...
apiserver{}
REST
HTTPS
HTTP methods/verbs
$ kubectl

Up Next:
Epic Recap

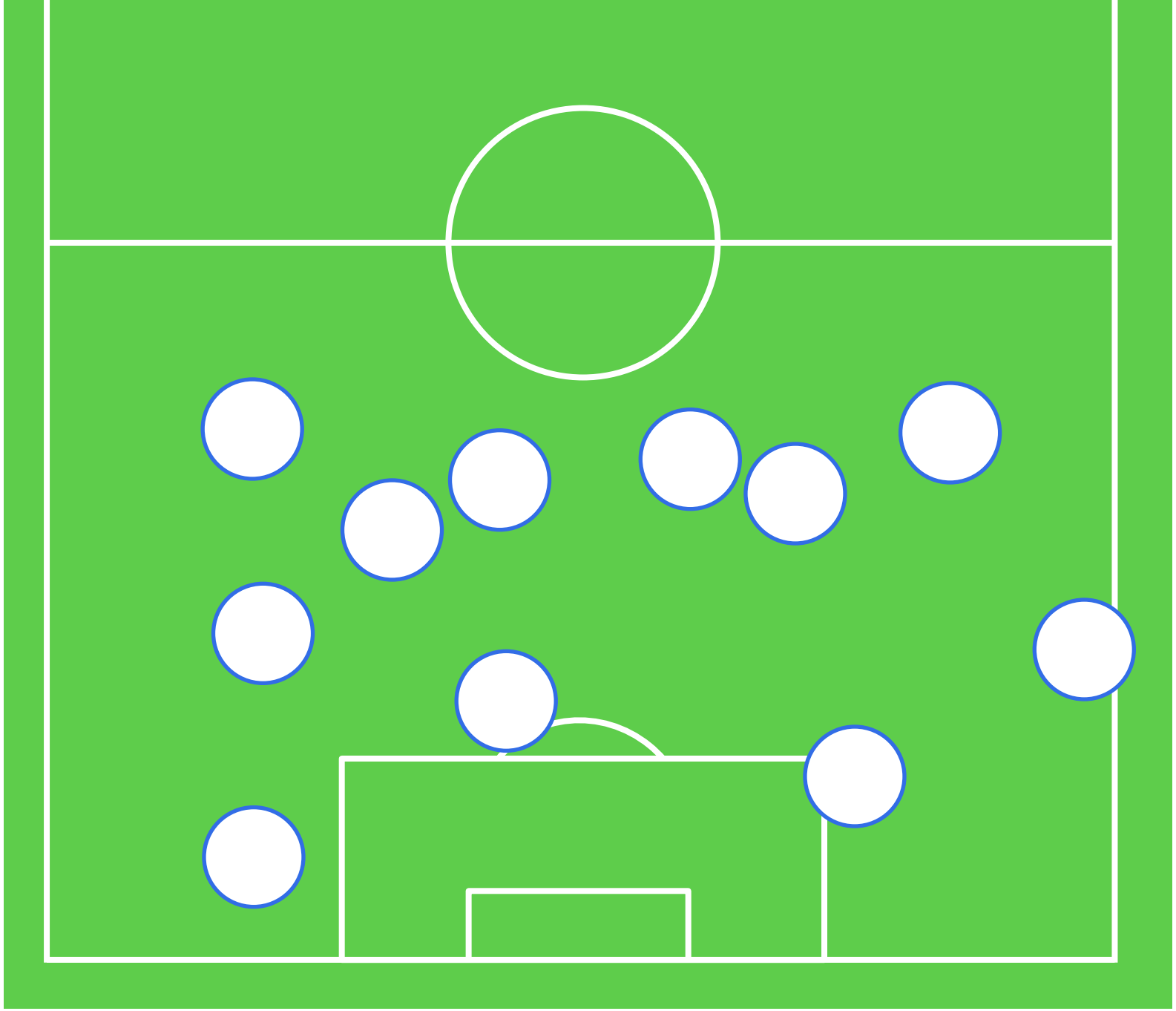
Epic Recap



Team

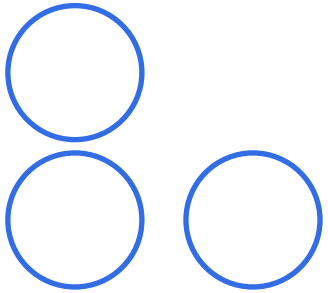


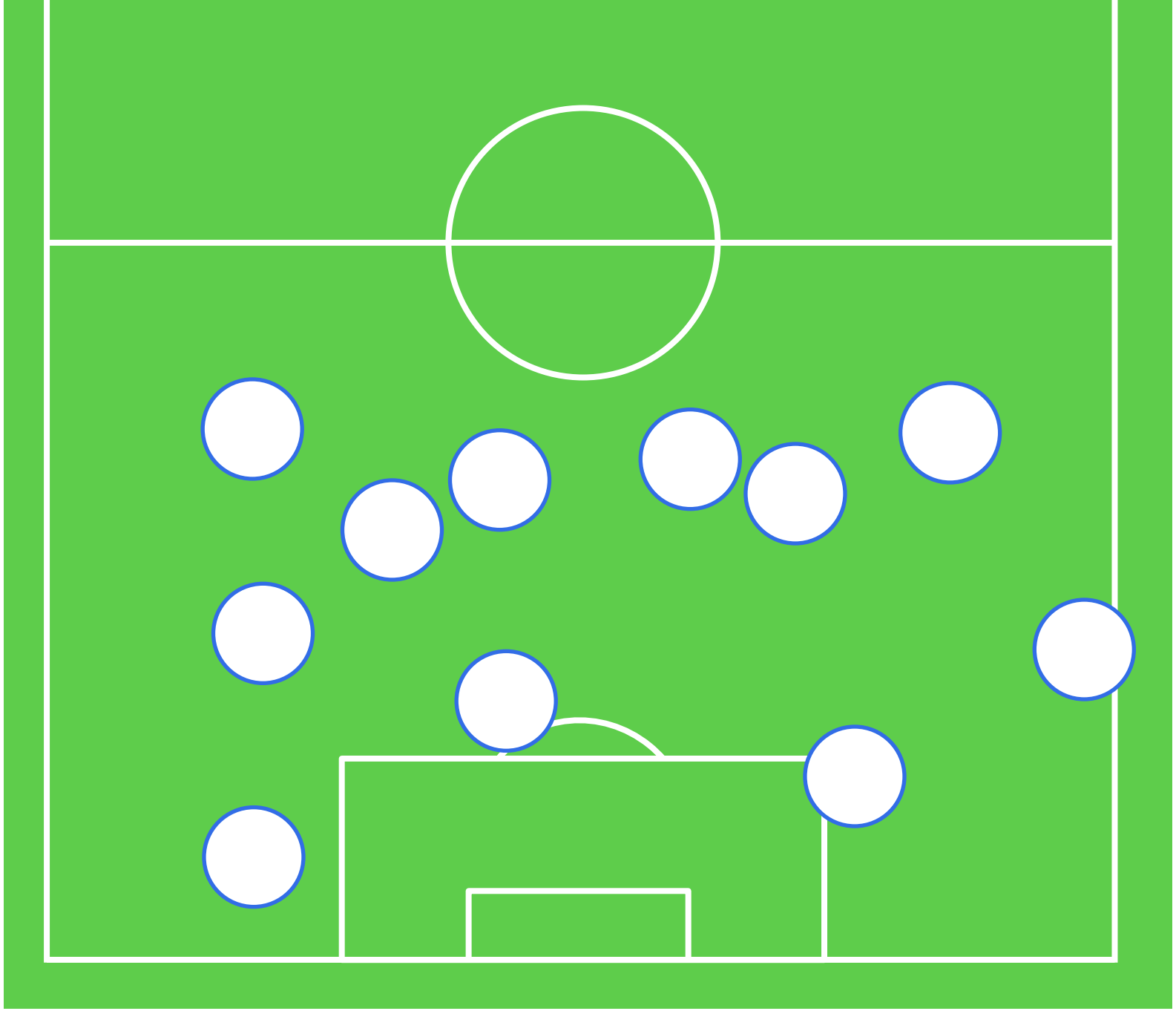
Team



Team

Manager
(coach)





Team

Manager
(coach)

S1

S2

S3

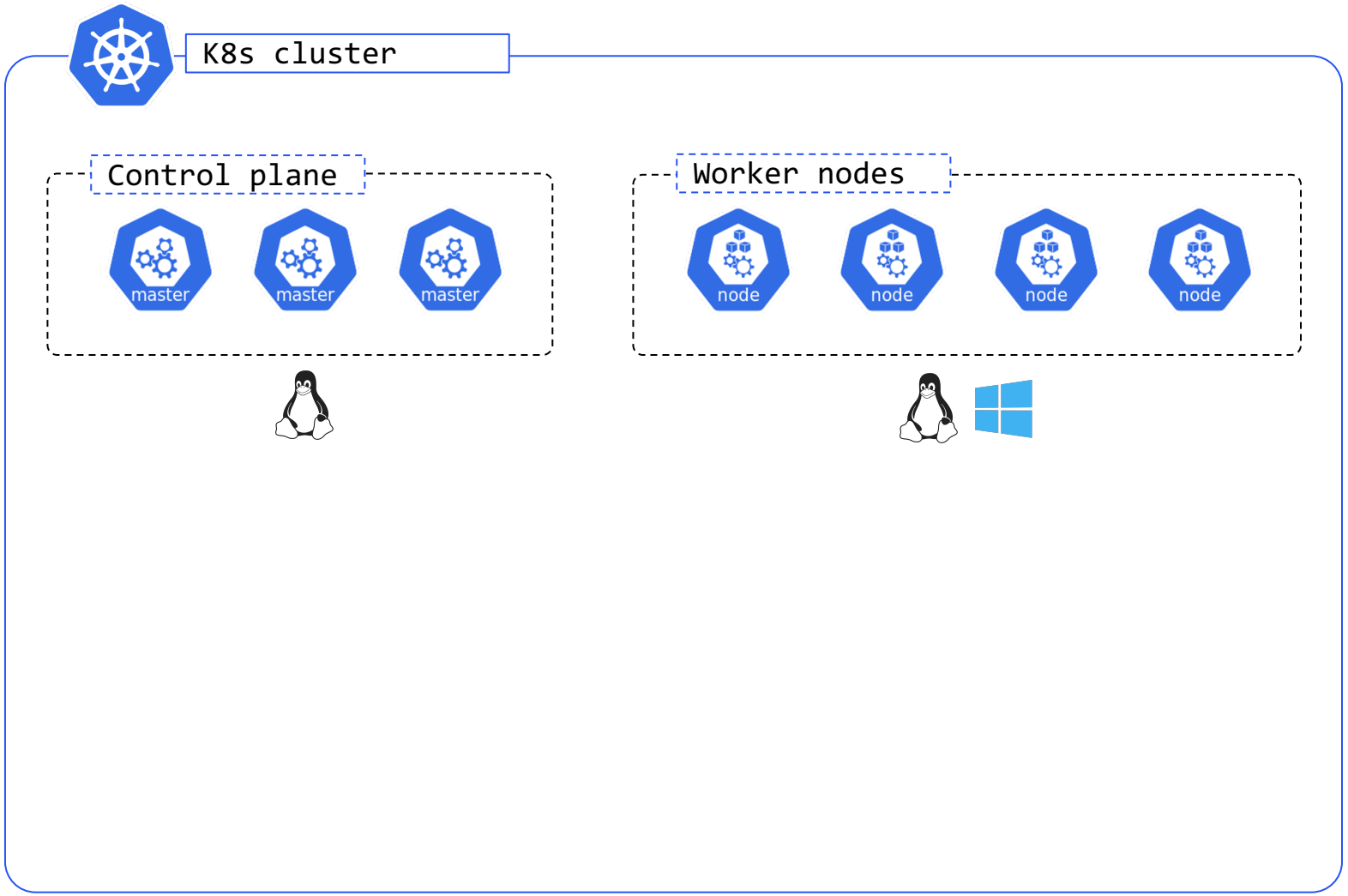
Team

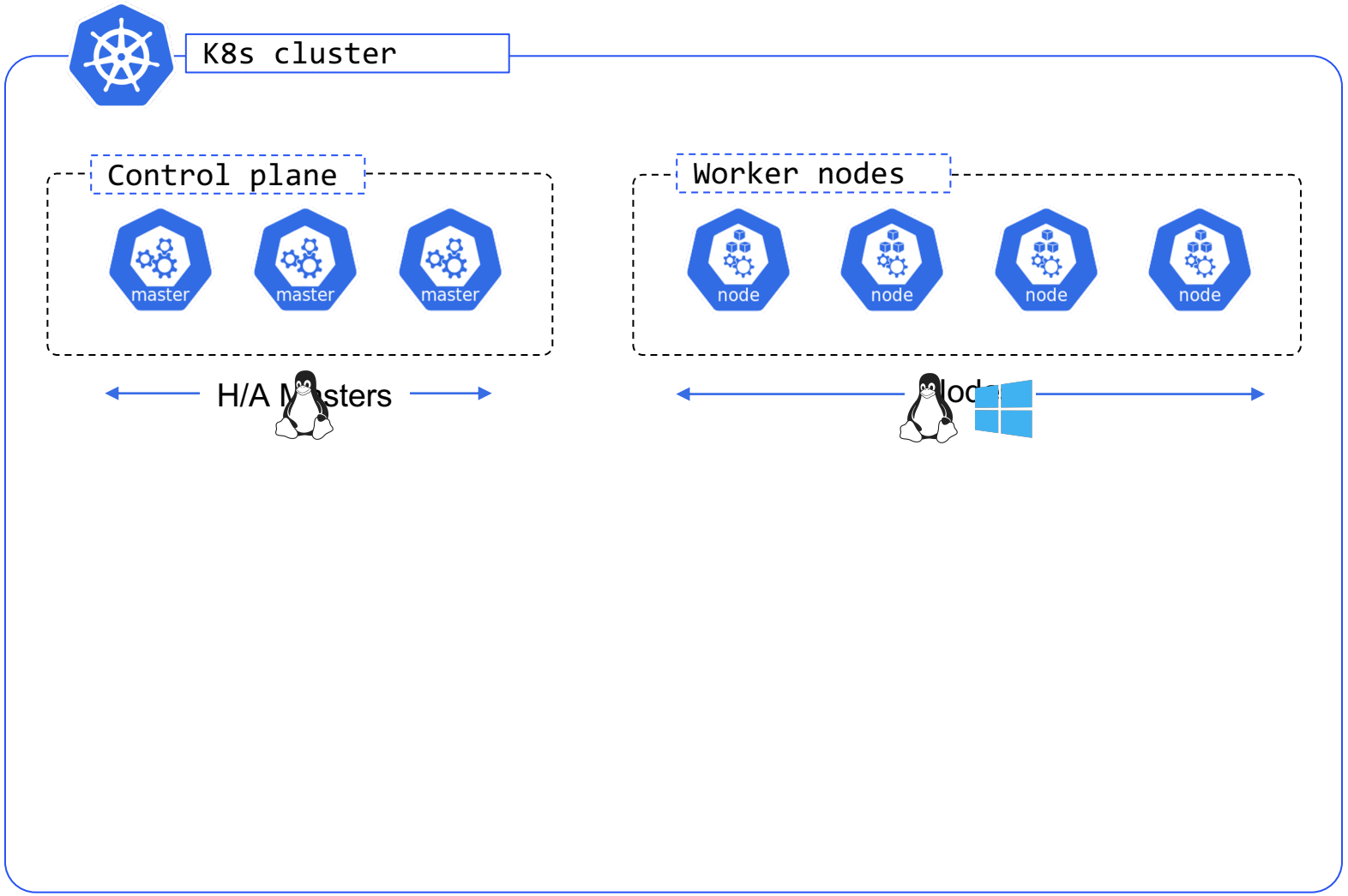
Manager
(coach)

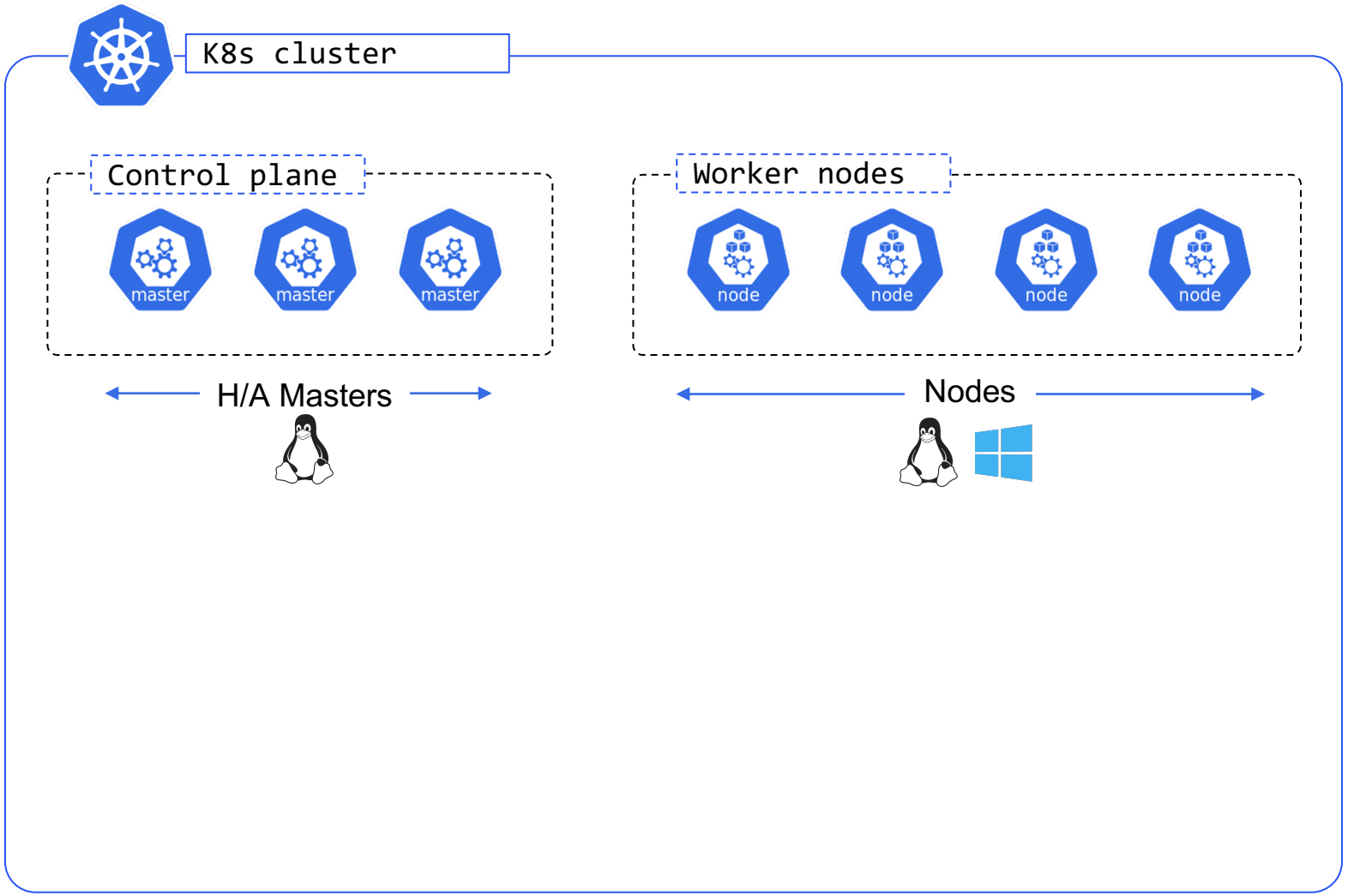
S1

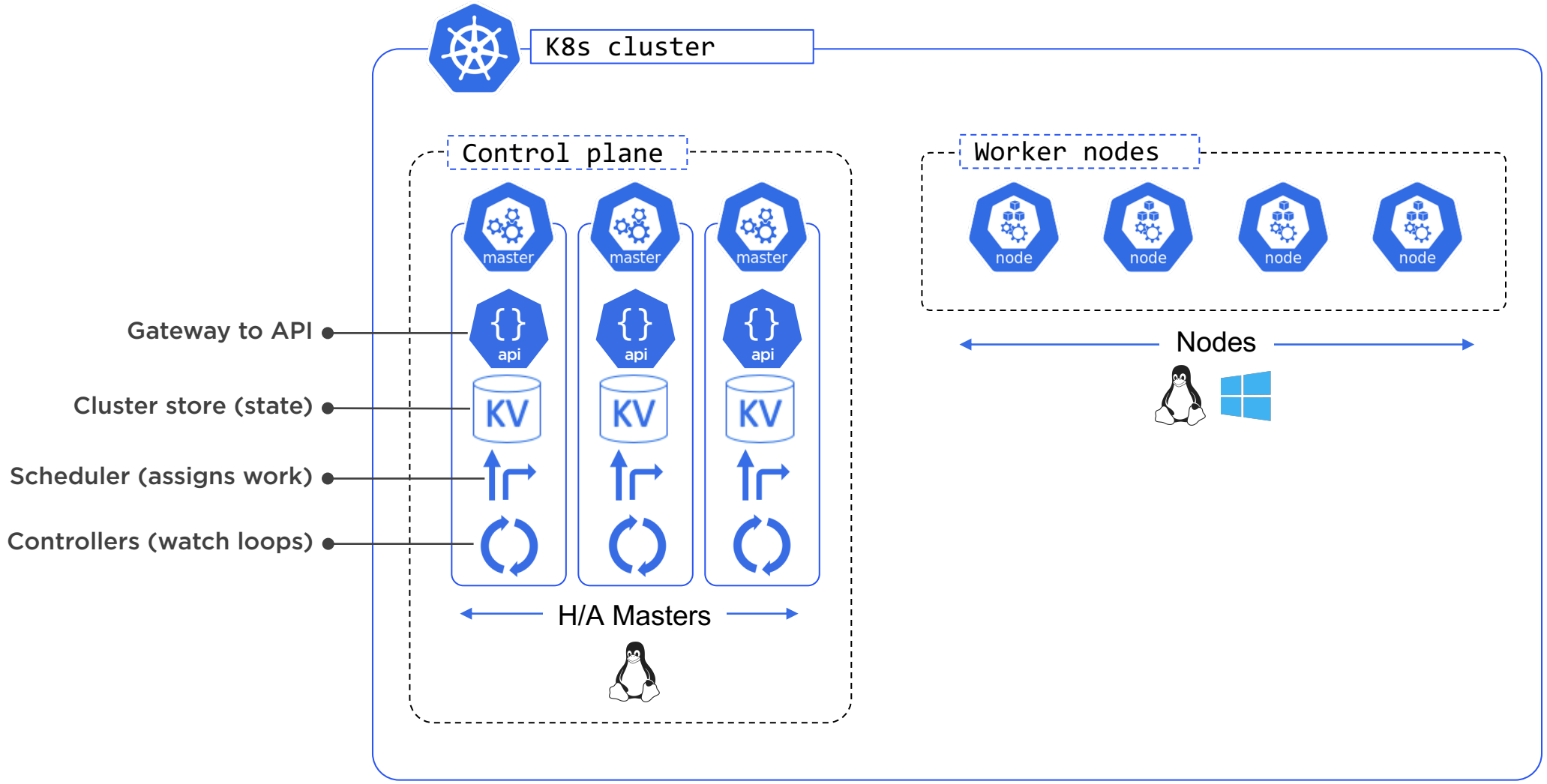
S2

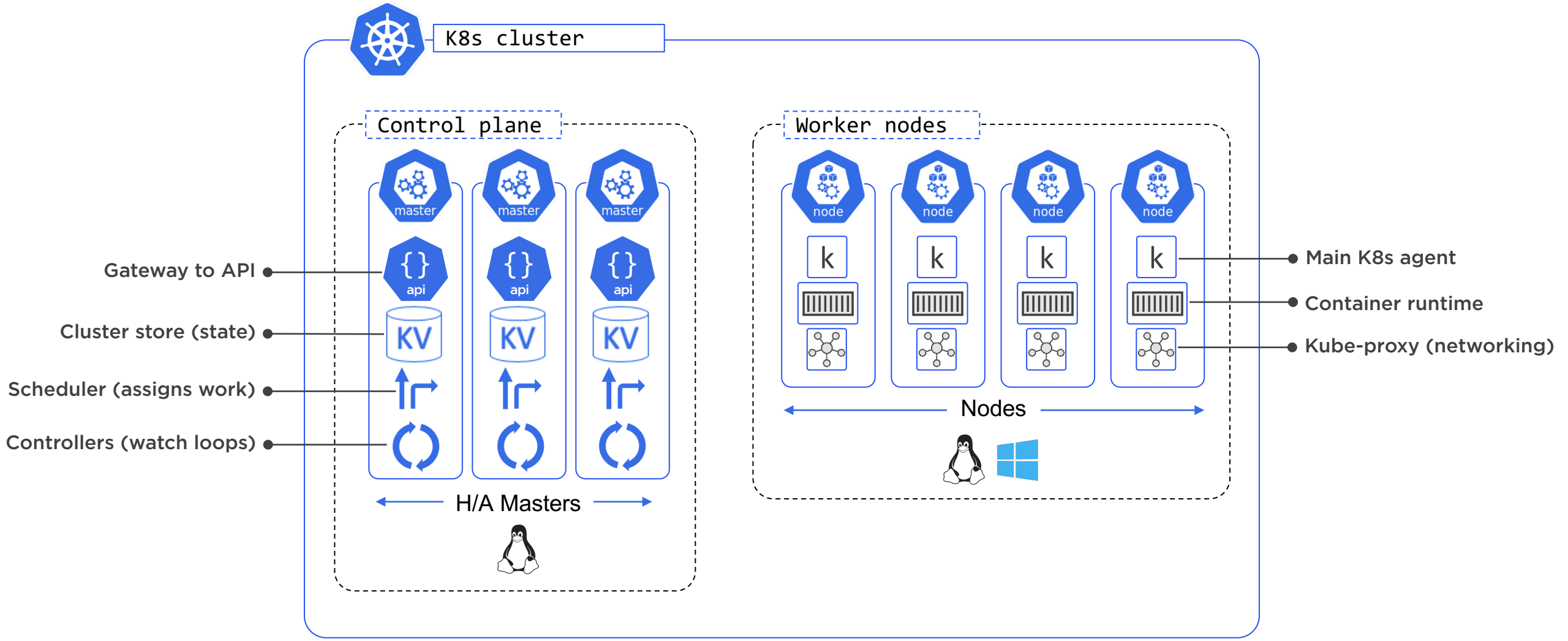
S3

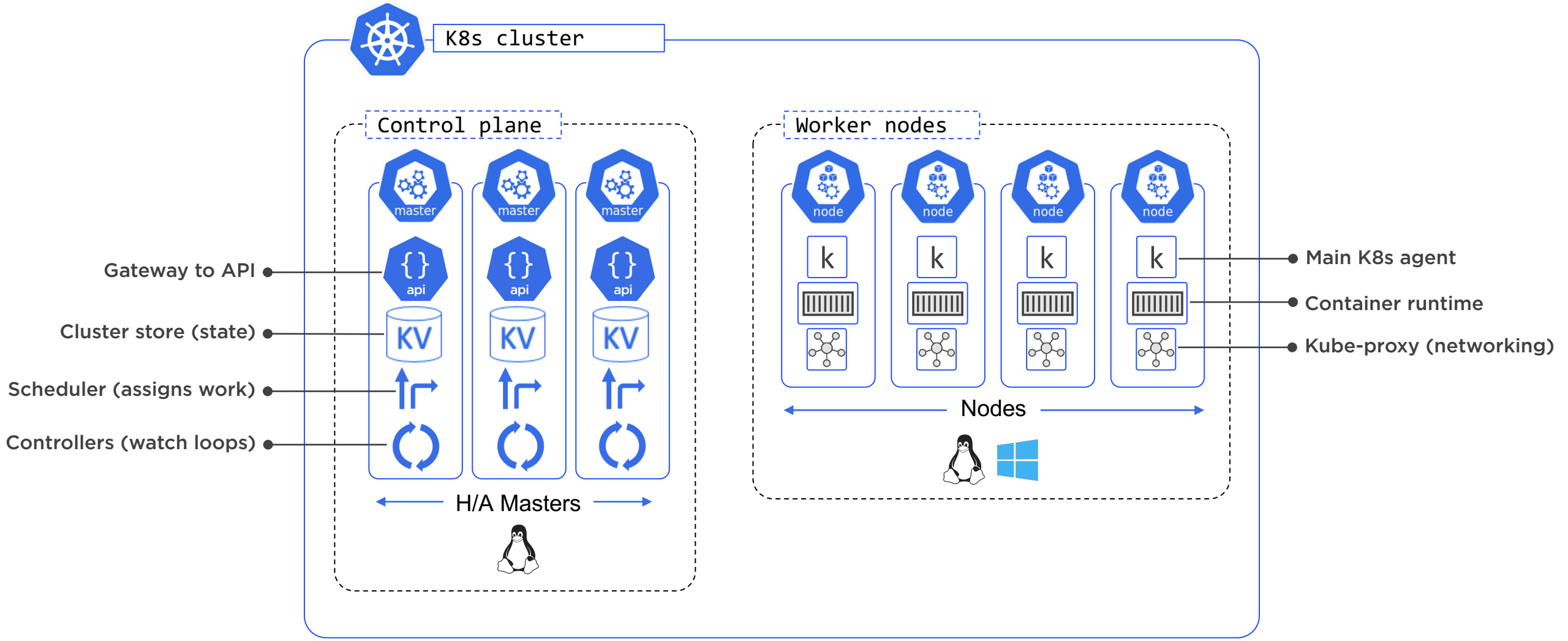

K8s cluster
Control plane
master
master
master
Worker nodes
node
node
node
node

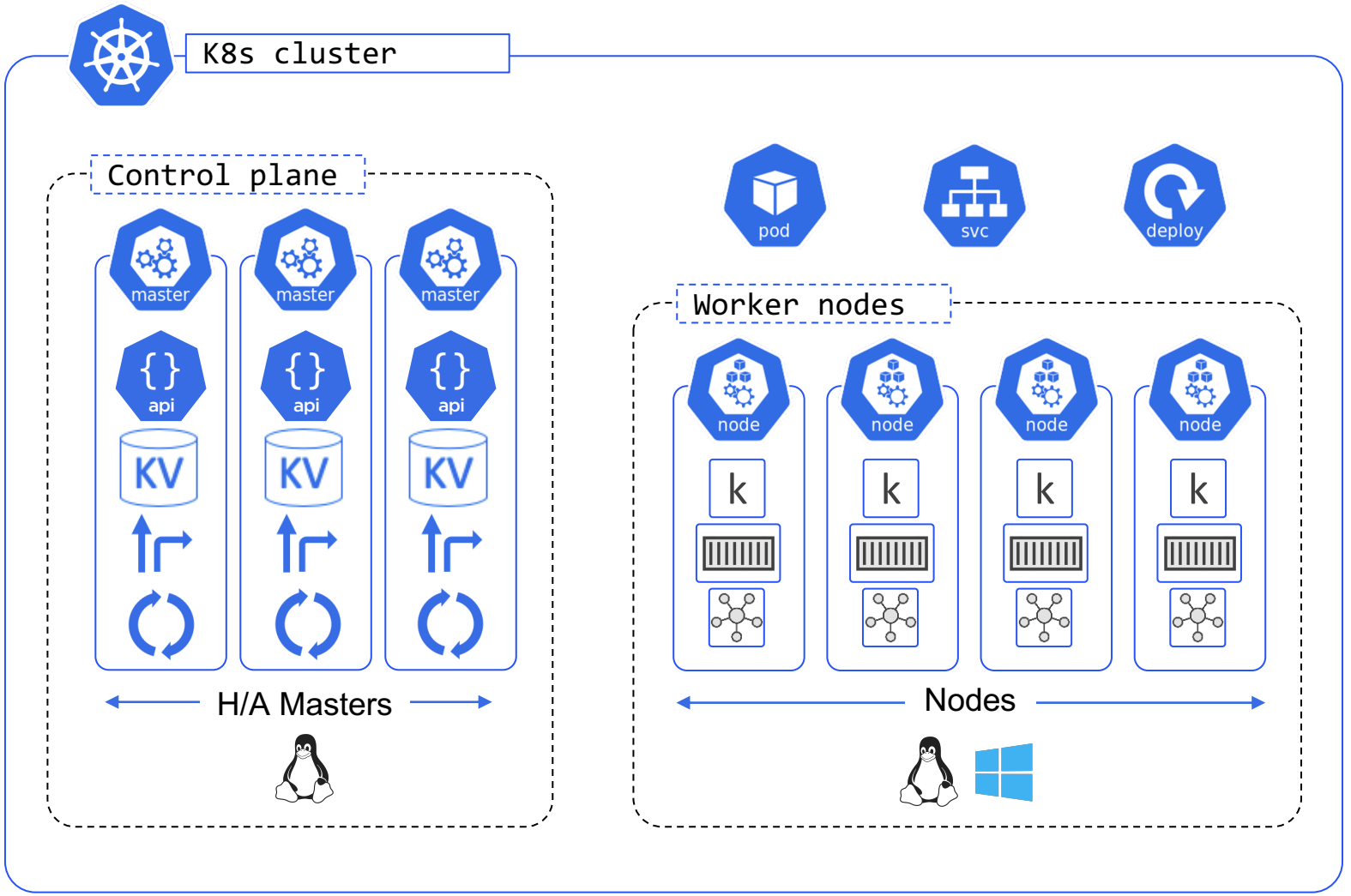

K8s cluster
Control plane
master
master
master
Worker nodes
node
node
node
node
H/A Masters


K8s cluster
Control plane
master
master
master
Worker nodes
node
node
node
node
H/A Masters
Nodes


K8s cluster
Control plane
Worker nodes
master
master
master
node
node
node
node
api
api
api
KV
KV
KV
Gateway to API
Cluster store (state)
Scheduler (assigns work)
Controllers (watch loops)
H/A Masters
Nodes


K8s cluster
Control plane
master
master
master
api
api
api
KV
KV
KV
H/A Masters
Worker nodes
node
node
node
node
k
k
k
k
Nodes
Gateway to API
Cluster store (state)
Scheduler (assigns work)
Controllers (watch loops)
Main K8s agent
Container runtime
Kube-proxy (networking)


K8s cluster
Control plane
Worker nodes
master
master
master
api
api
api
KV
KV
KV
H/A Masters
node
node
node
node
k
k
k
k
Nodes
Gateway to API
Cluster store (state)
Scheduler (assigns work)
Controllers (watch loops)
Main K8s agent
Container runtime
Kube-proxy (networking)


K8s cluster
Control plane
master
master
master
api
api
api
KV
KV
KV
H/A Masters
pod
svc
deploy
Worker nodes
node
node
node
node
k
k
k
k
Nodes

Up Next:
Getting Kubernetes