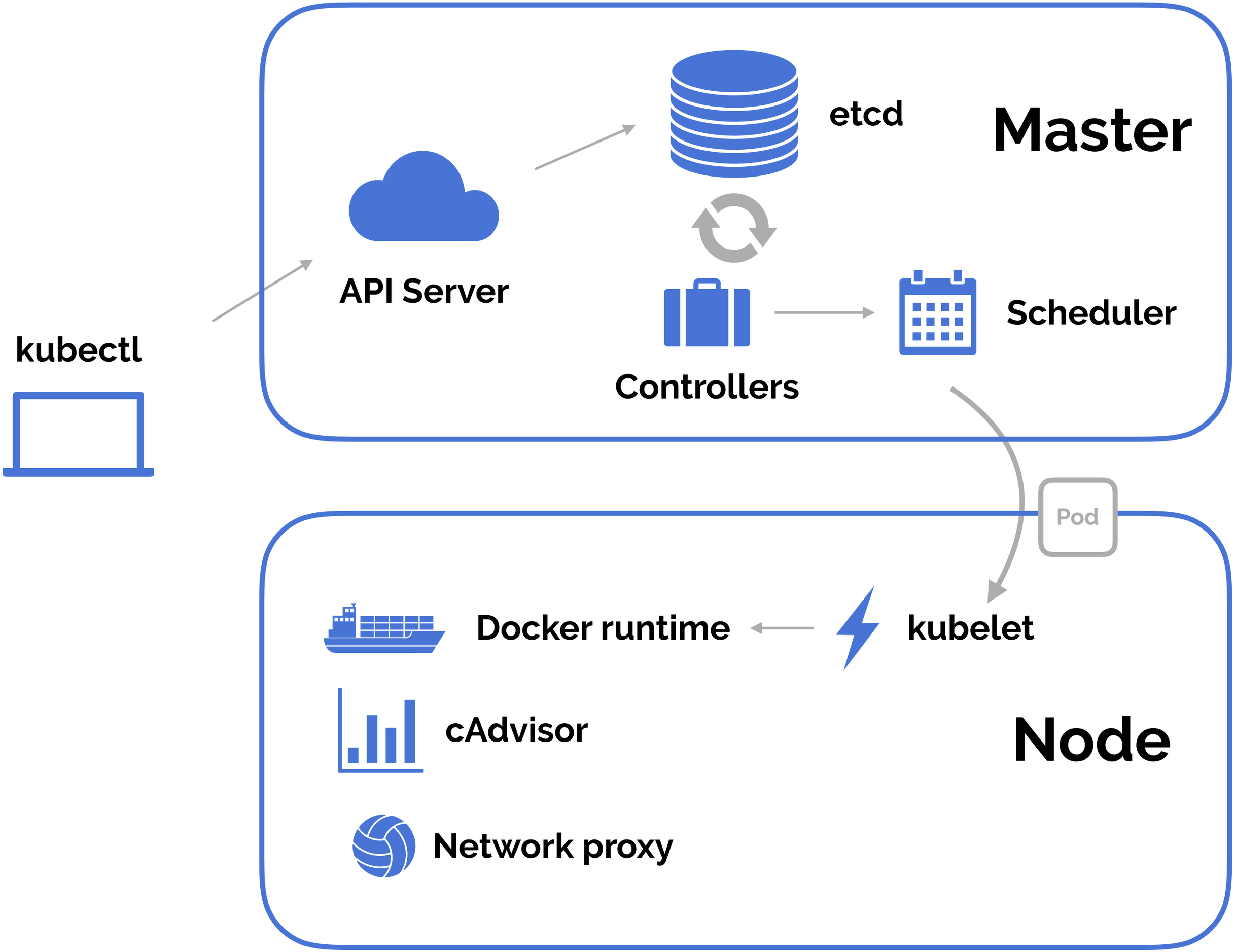
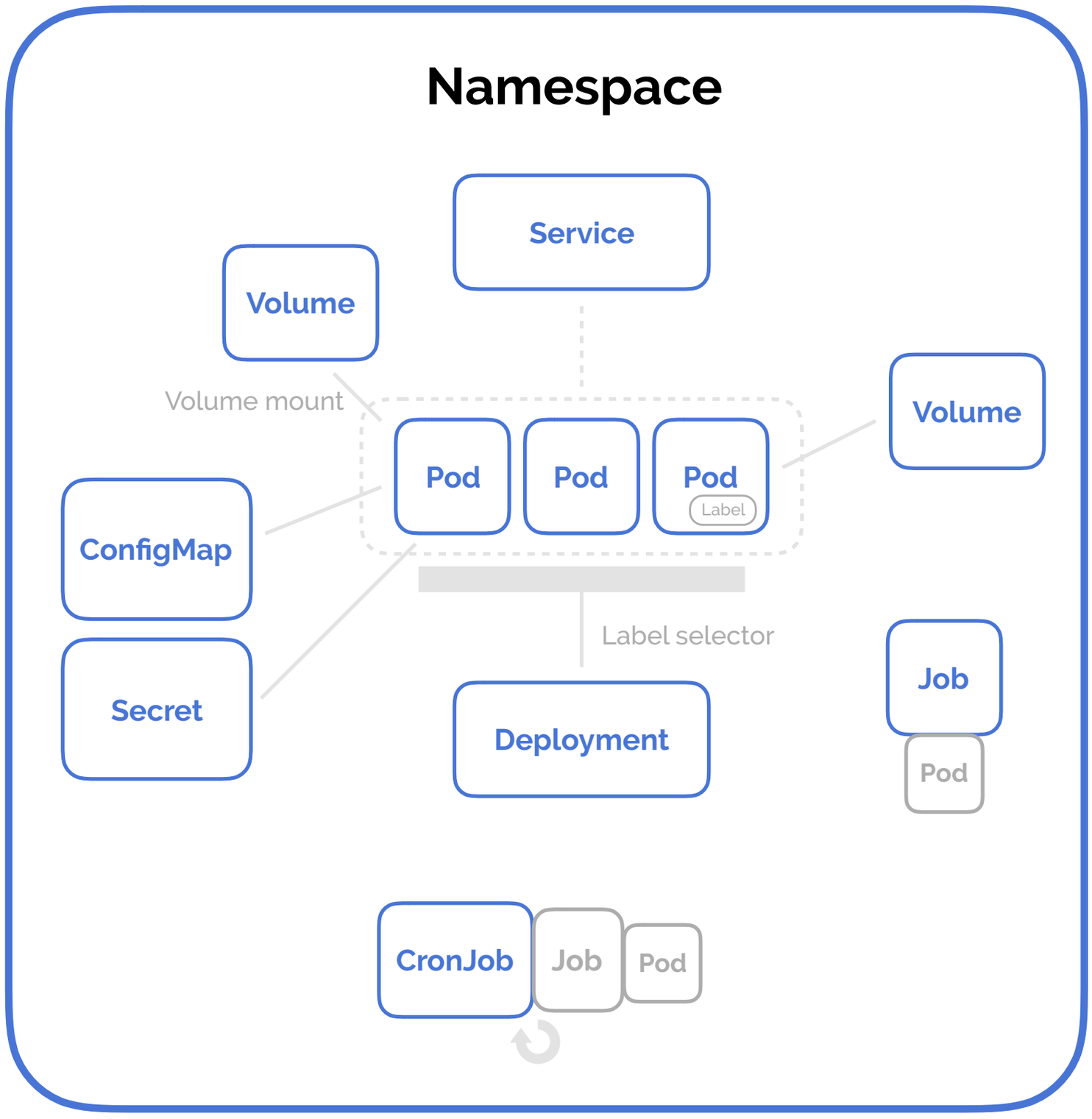
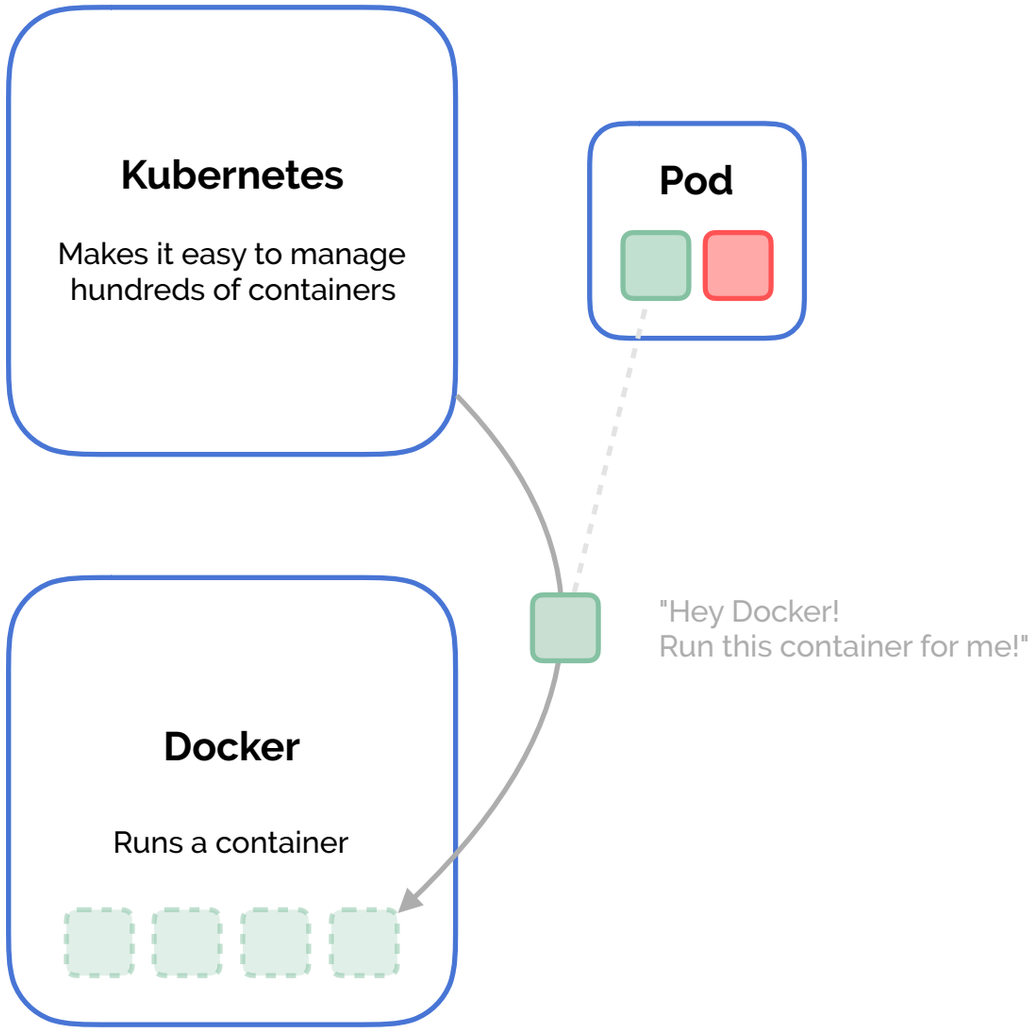
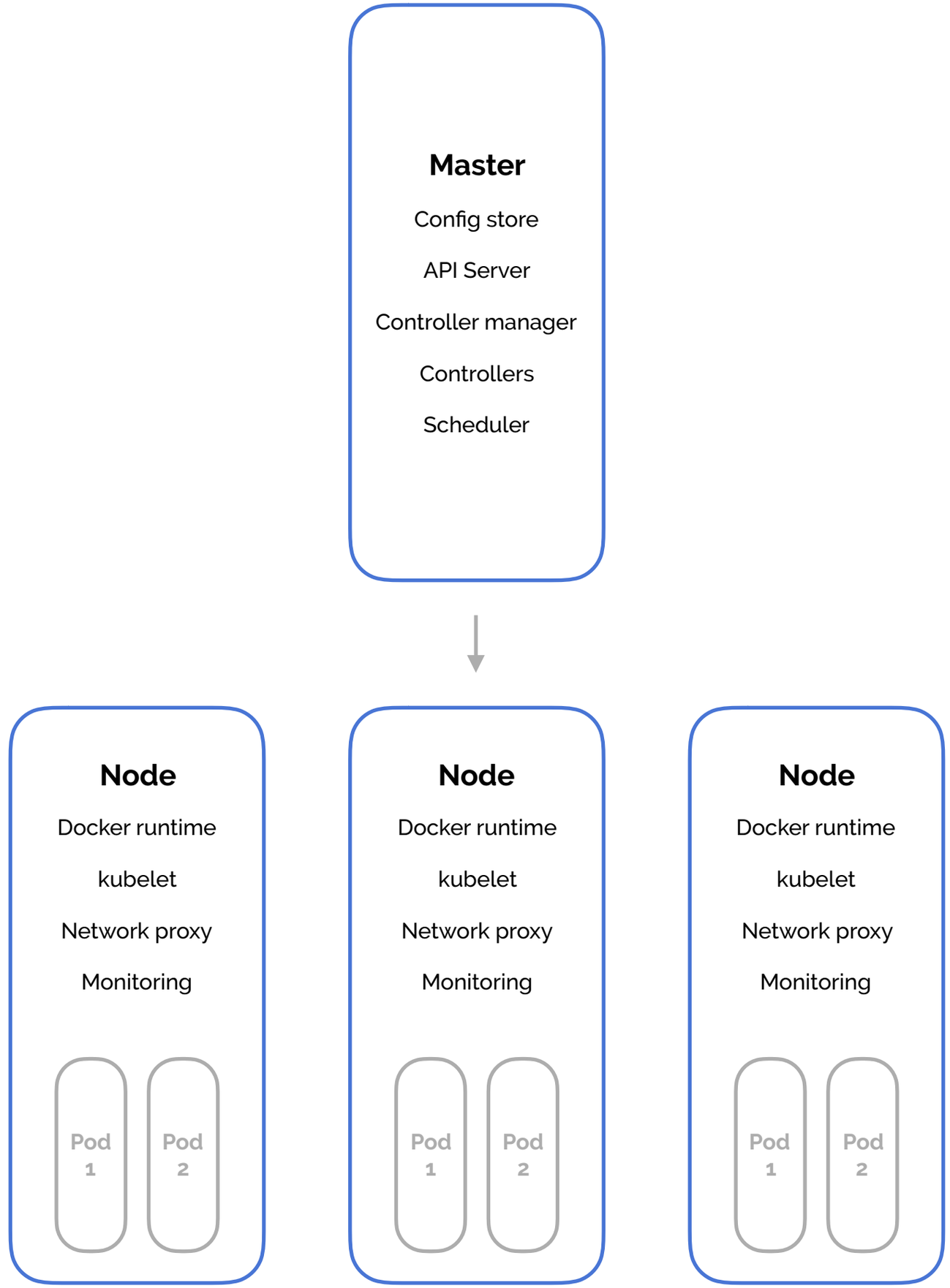
Dockerfile
"Source code"
for your image
docker build
Image
Template to
create containers
docker run
Containers
Runtime instances
of your image

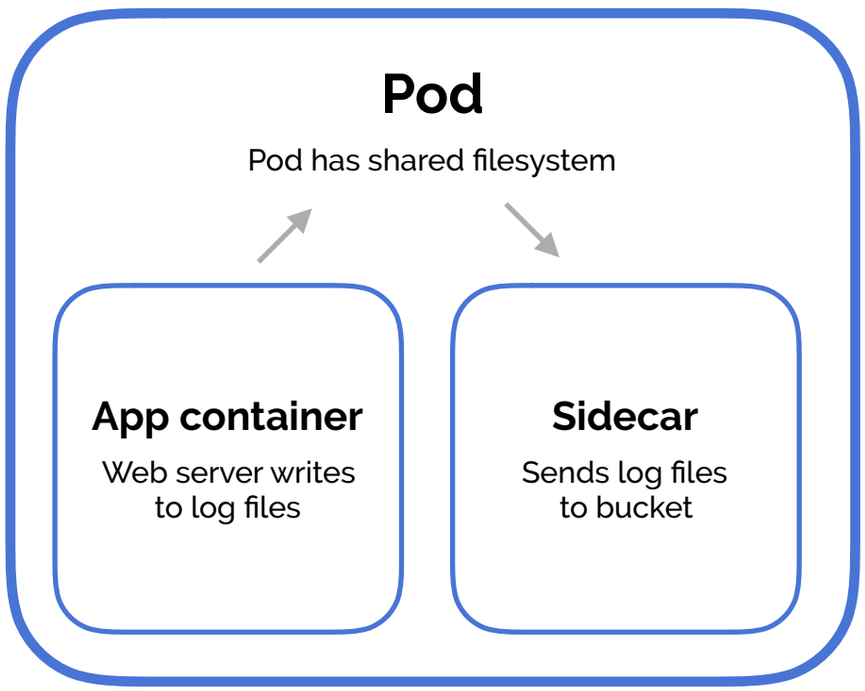

Master
etcd
API Server
Controllers
Scheduler
kubectl
Pod
Node
Docker runtime
kubelet
cAdvisor
Network proxy


Namespace
Service
Volume
Volume mount
Pod
Pod
Pod
Label
Volume
ConfigMap
Secret
Label selector
Deployment
Job
Pod
CronJob
Job
Pod

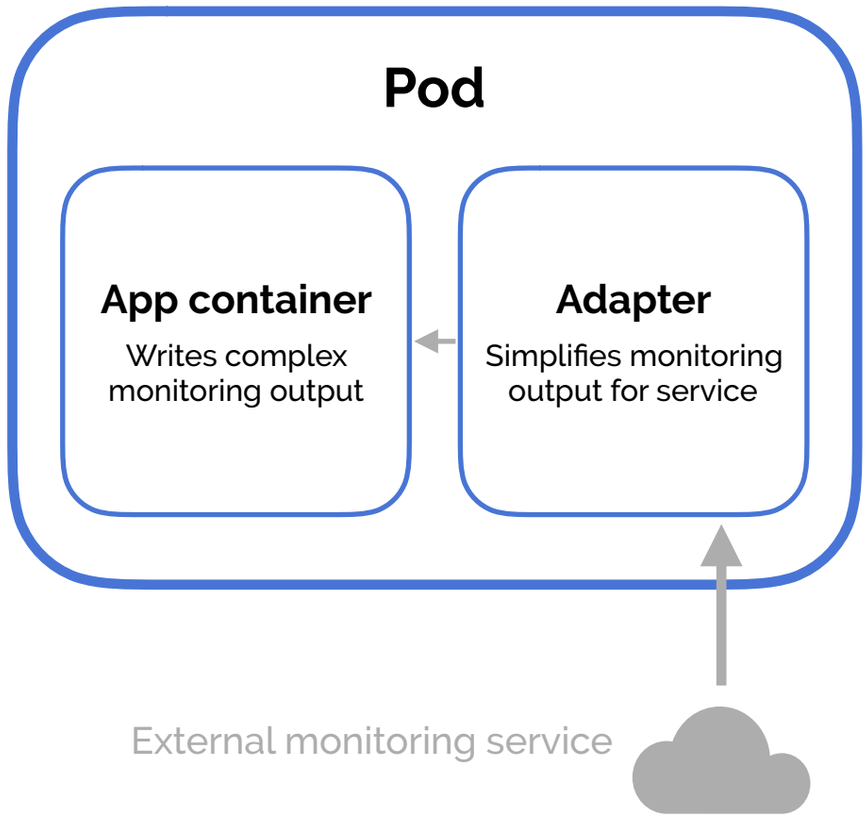

Kubernetes
Makes it easy to manage hundreds of containers
Pod
"Hey Docker!
Run this container for me!"
Docker
Runs a container


Master
Config store
API Server
Controller manager
Controllers
Scheduler
Node
Docker runtime
kubelet
Network proxy
Monitoring
Pod 1
Pod 2
Node
Docker runtime
kubelet
Network proxy
Monitoring
Pod 1
Pod 2
Node
Docker runtime
kubelet
Network proxy
Monitoring
Pod 1
Pod 2

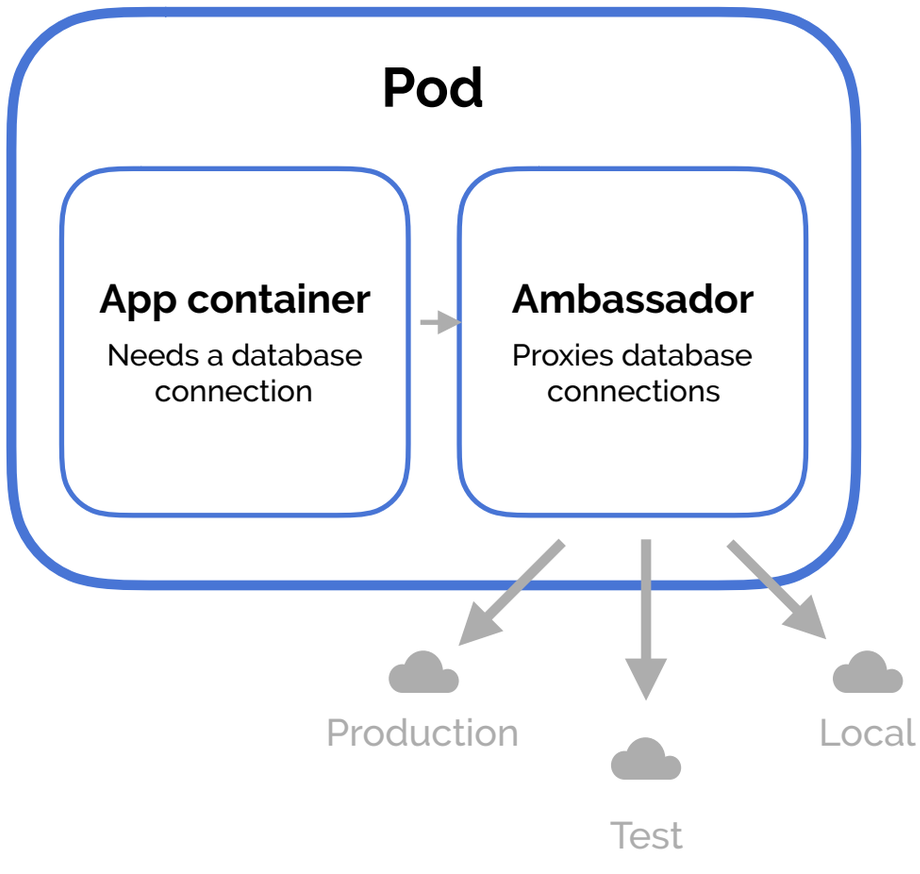
Sidecar



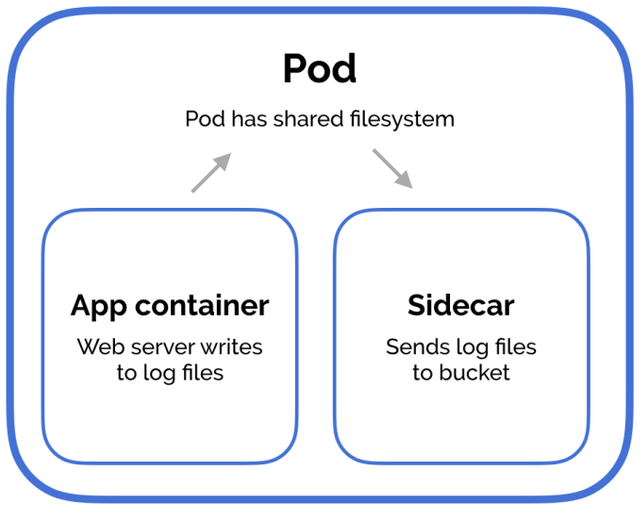
Adapter



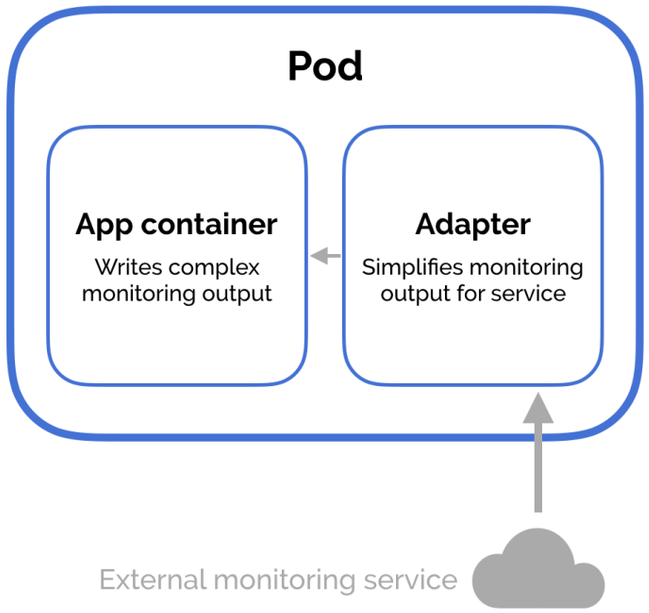
Ambassador



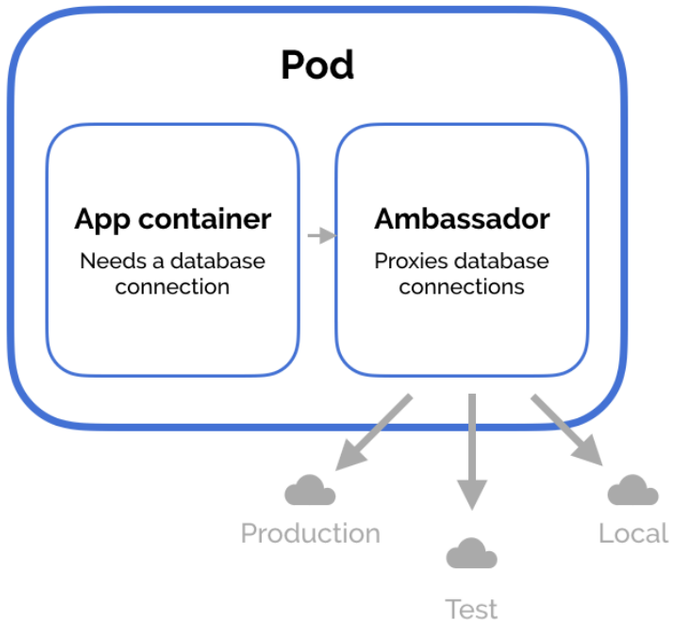
Sidecar

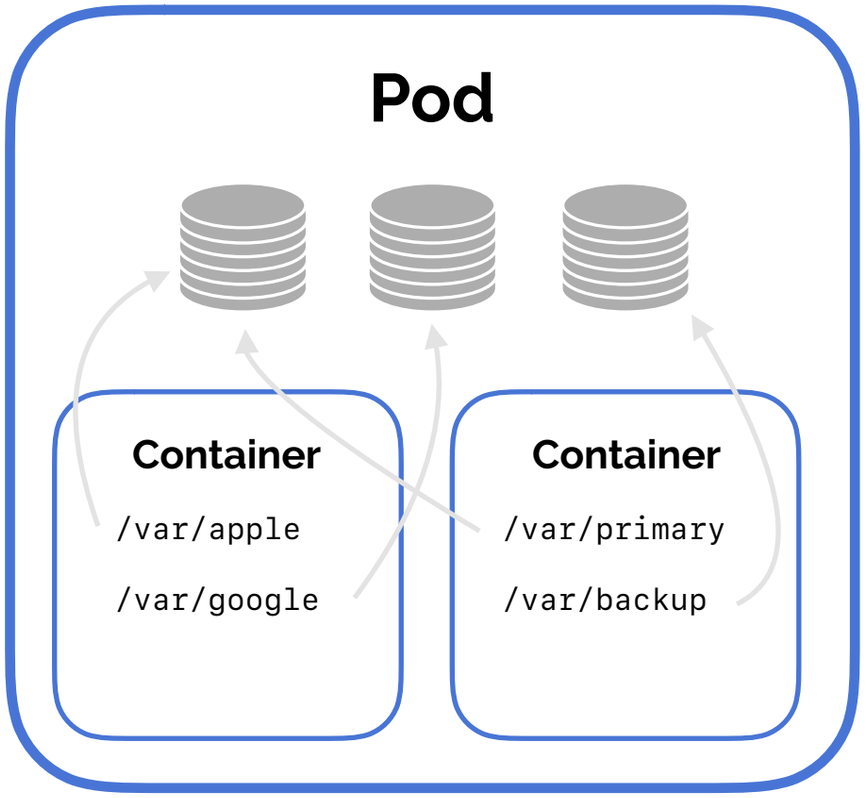


Adapter



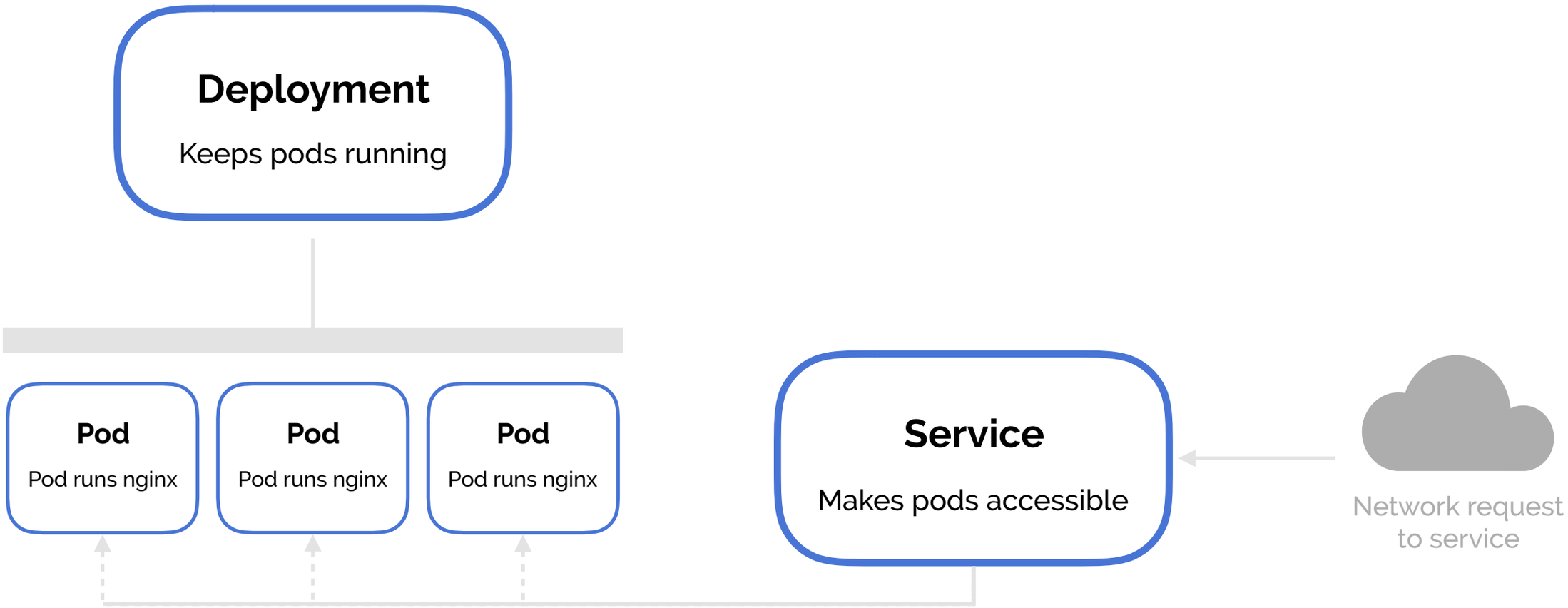
Ambassador

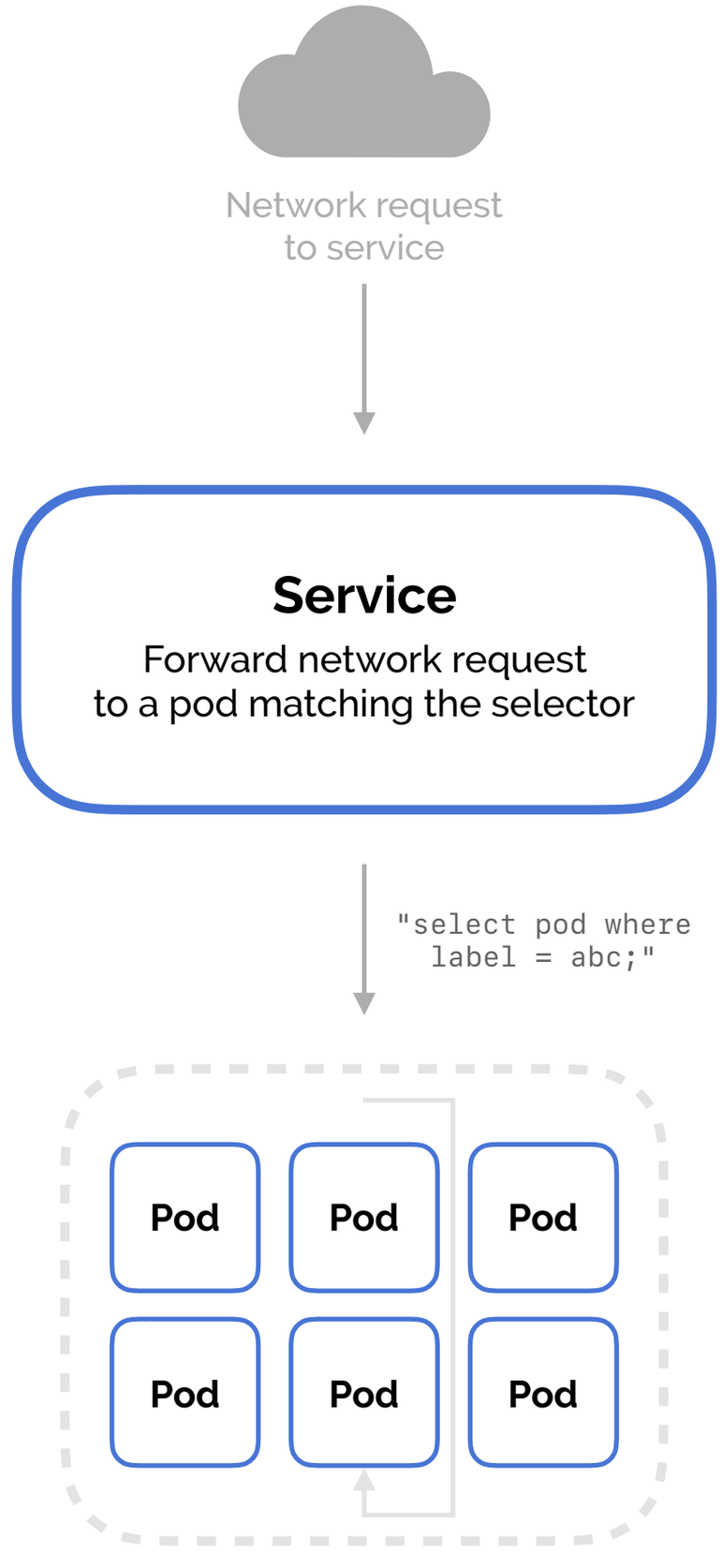


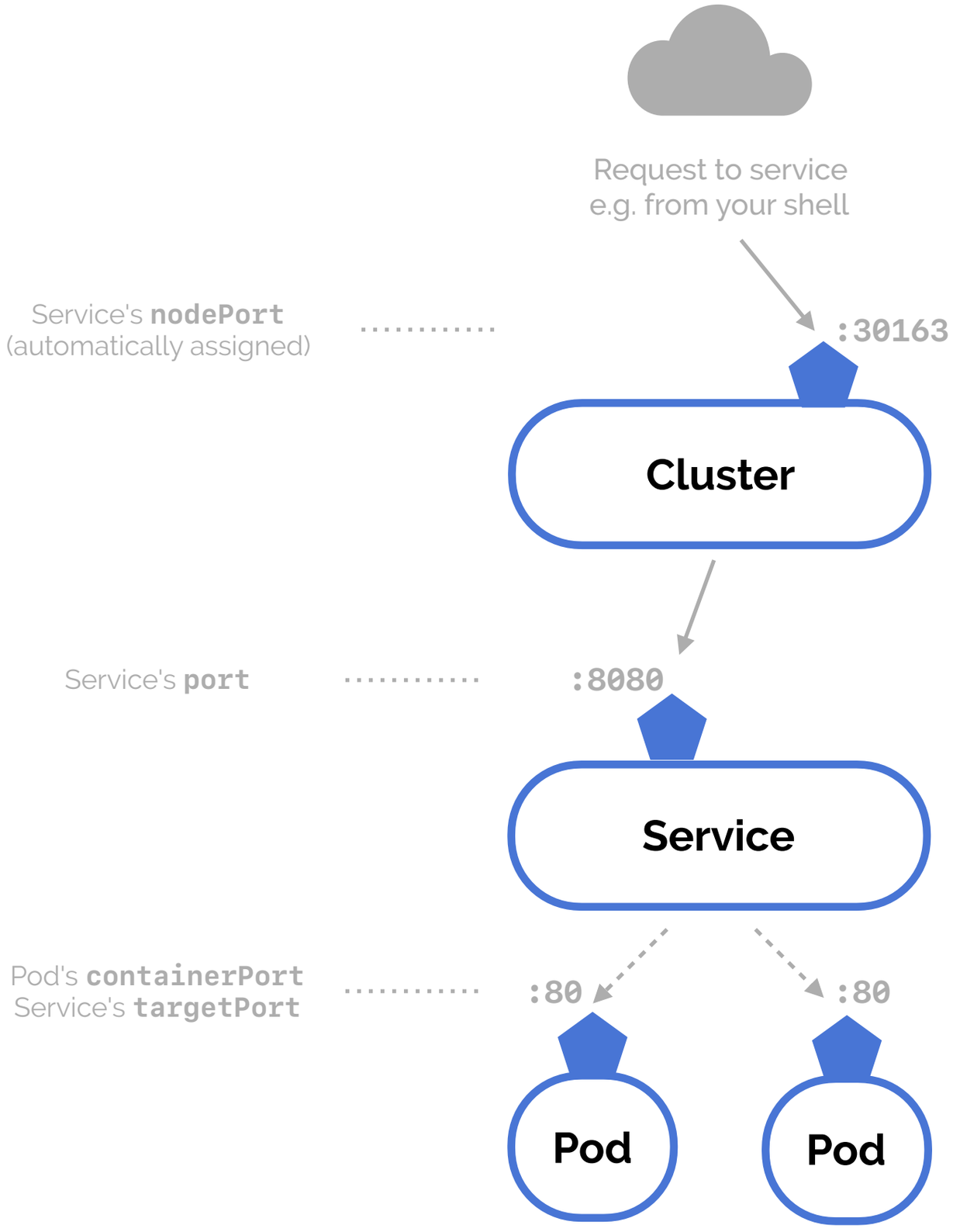


Volumes in **pod**
Defined in `pod.spec.volumes` field

Volume mounts in **container**
Defined in
`pod.spec.containers.volumeMounts`
Uses one of the volumes


Deployment
Keeps pods running
Pod
Pod runs nginx
Pod
Pod runs nginx
Pod
Pod runs nginx
Service
Makes pods accessible
Network request
to service


Network request
to service
Service
Forward network request
to a pod matching the selector
"select pod where
label = abc;"
Pod
Pod
Pod
Pod
Pod
Pod



Cluster

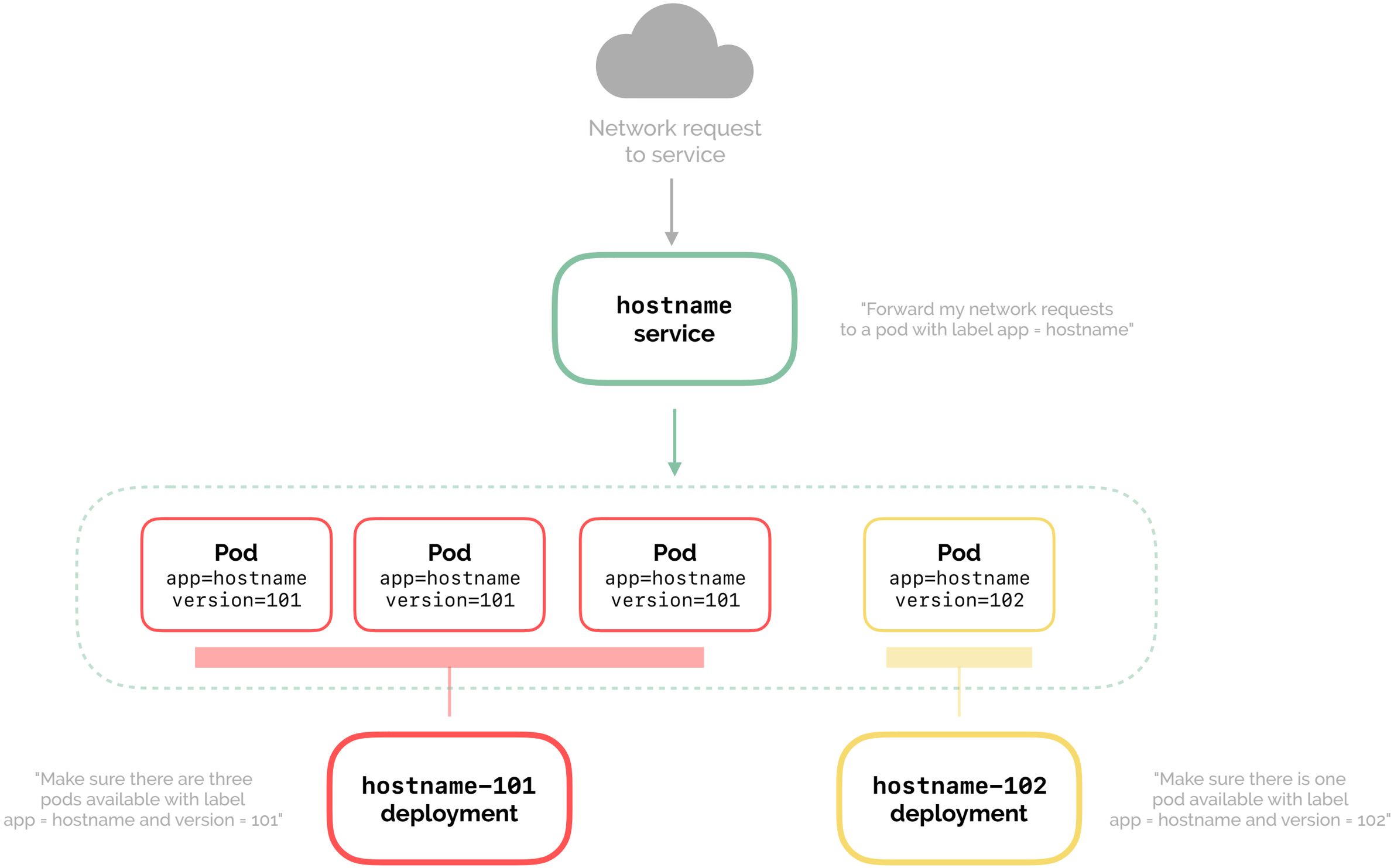
- Access via cluster's IP address or domain name, e.g. 192.168.99.100
- Cluster port available because service is of type NodePort

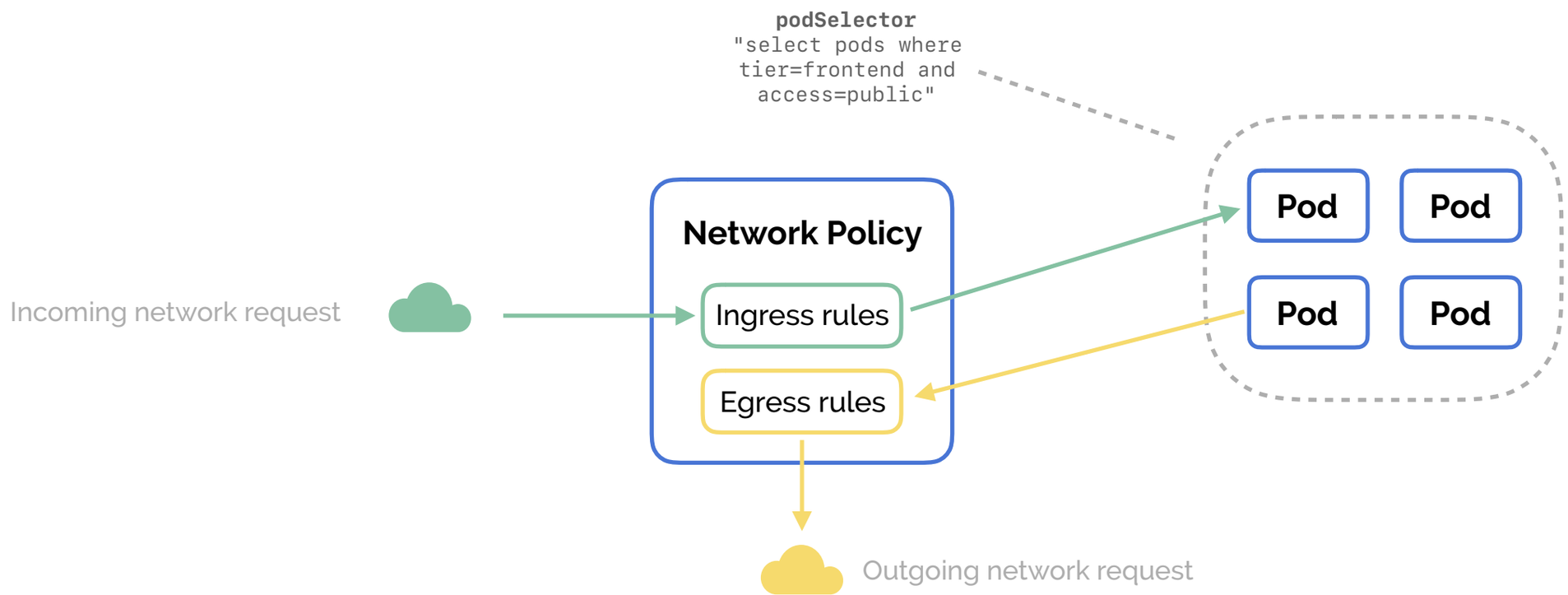
Service

- Access only within the cluster via the service's clusterIP
- Available at 10.100.213:8080 or hostname-service.default:8080
- Service exposes port 8080, targets pod port 80

Pod

- Access via the pod's IP address inside the cluster
- Pod exposes port 80 (which nginx uses by default)


Network request
to service
hostname
service
"Forward my network requests
to a pod with label app = hostname"
Pod
app=hostname
version=101
Pod
app=hostname
version=101
Pod
app=hostname
version=101
Pod
app=hostname
version=102
"Make sure there are three
pods available with label
app = hostname and version = 101"
hostname-101
deployment
hostname-102
deployment
"Make sure there is one
pod available with label
app = hostname and version = 102"


podSelector
"select pods where
tier=frontend and
access=public"
Network Policy
Ingress rules
Egress rules
Pod
Pod
Pod
Pod
Incoming network request
Outgoing network request

Kubernetes cluster
cross-team

Nodes

User Accounts

Namespace
red-team

Service Accounts

Resource Quota

Pods

Services

Deployments

Configuration

ConfigMaps

Secrets

Volumes

Namespace
green-team

Service Accounts

Resource Quota

Pods

Services

Deployments

Configuration

ConfigMaps

Secrets

Volumes

YAML config file

Declares your pod

`kubectl create`, `kubectl apply`

Execute command to create pod

Pod

Container

master

1

API Server

Handles change request

2

etcd

Stores configuration

3

Controllers

Watch etcd for changes

4

Scheduler

Chooses node to put pod on

Pod

Container

1

Pull image

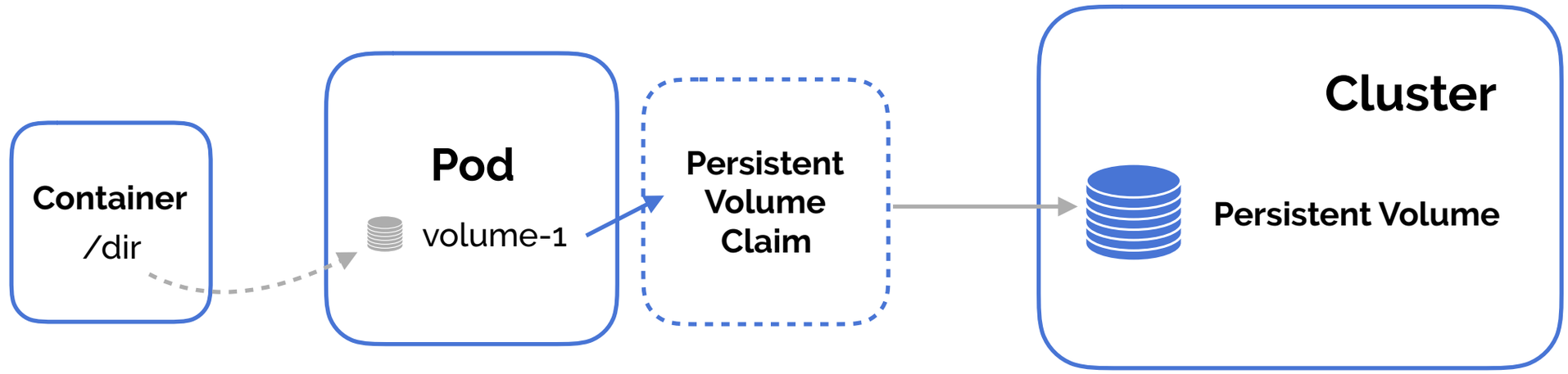
2

Instantiate container

3

Run container

node



`volumeMount`

`volume`
`type: pvc`

Gives pod control of persistent volume

Persistent storage resource owned by cluster