kubernetes Workshop

Introduction

2023.08.22.

Agenda

Introduction workshop

- What is Kubernetes
- K8s architecture
- Main components
- K8s configuration
- Hands-on workshop

https://github.com/peternagy652/k8s_workshop

- (Later down the line: Helm, Operators, Monitoring, RBAC, etc…)

What is Kubernetes?

What is Kubernetes?

Open source container orchestration tool, framework

Developed by Google

The goal is to manage containerised applications in different deployment environments

Automates many of the manual processes involved in deploying, managing and scaling these applications

Aims to solve the scripting hell and customised

Tasks of Kubernetes

High availability or no downtime

Make the application always accessible by the users.

Scalability or high performance

Flexibly scale the application fast according to the

Disaster recovery - backup and restore

If any data is lost, the servers go down, the infra backs up and restores it’s state.

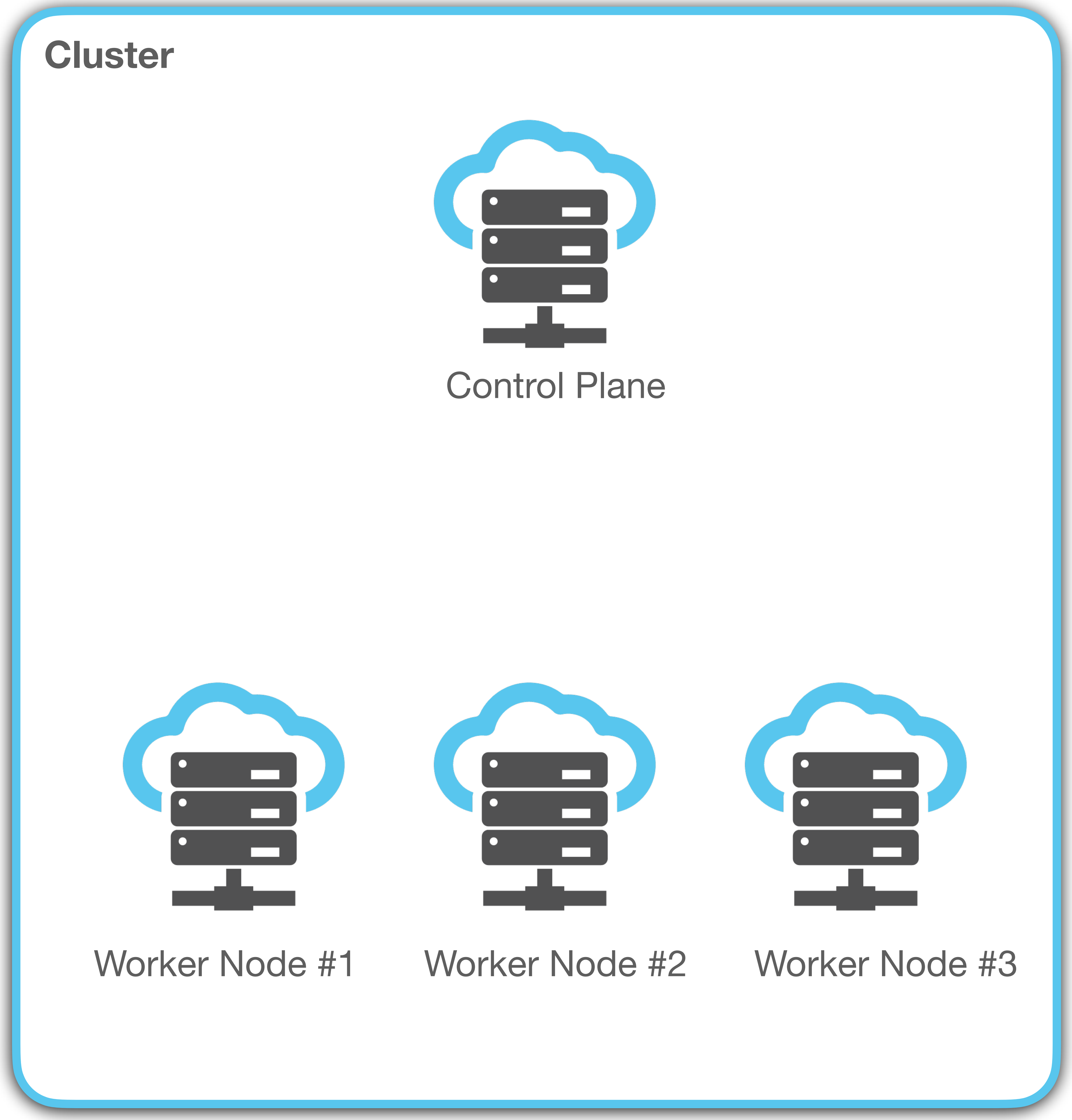
Kubernetes Architecture

Kubernetes Architecture

Node: virtual or physical machine

Worker nodes (usually referred to as simply “Nodes”), do the actual work, the applications are running here. They have higher workload, usually have way more resources then the Master Nodes.

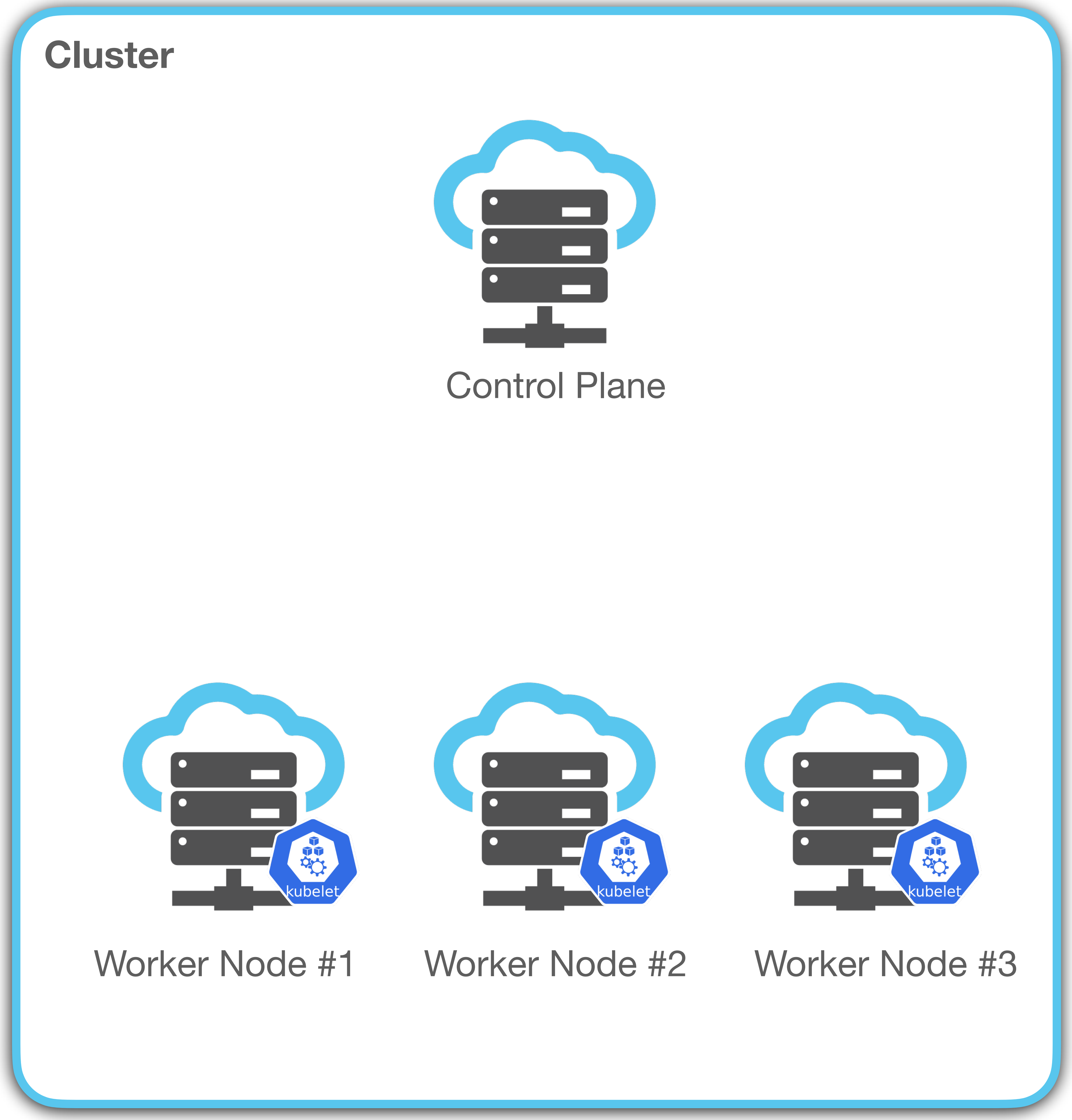
The Control Plane manages the worker nodes. In production environments, the Control Plane usually runs across multiple computers and a cluster usually runs multiple of these, providing fault-tolerance and high availability



Kubernetes Architecture

Primary agent running on nodes - makes it possible for the nodes to communicate to each other, and execute tasks on nodes

Each worker node has containers deployed on it. Based on the actual workload, different number of containers are running on each node.



Kubernetes Architecture

K8s processes running on the Master Node

api

API Server, running as a container. The entry point to K8s cluster, for UIs, APIs, CLIs.

c-m

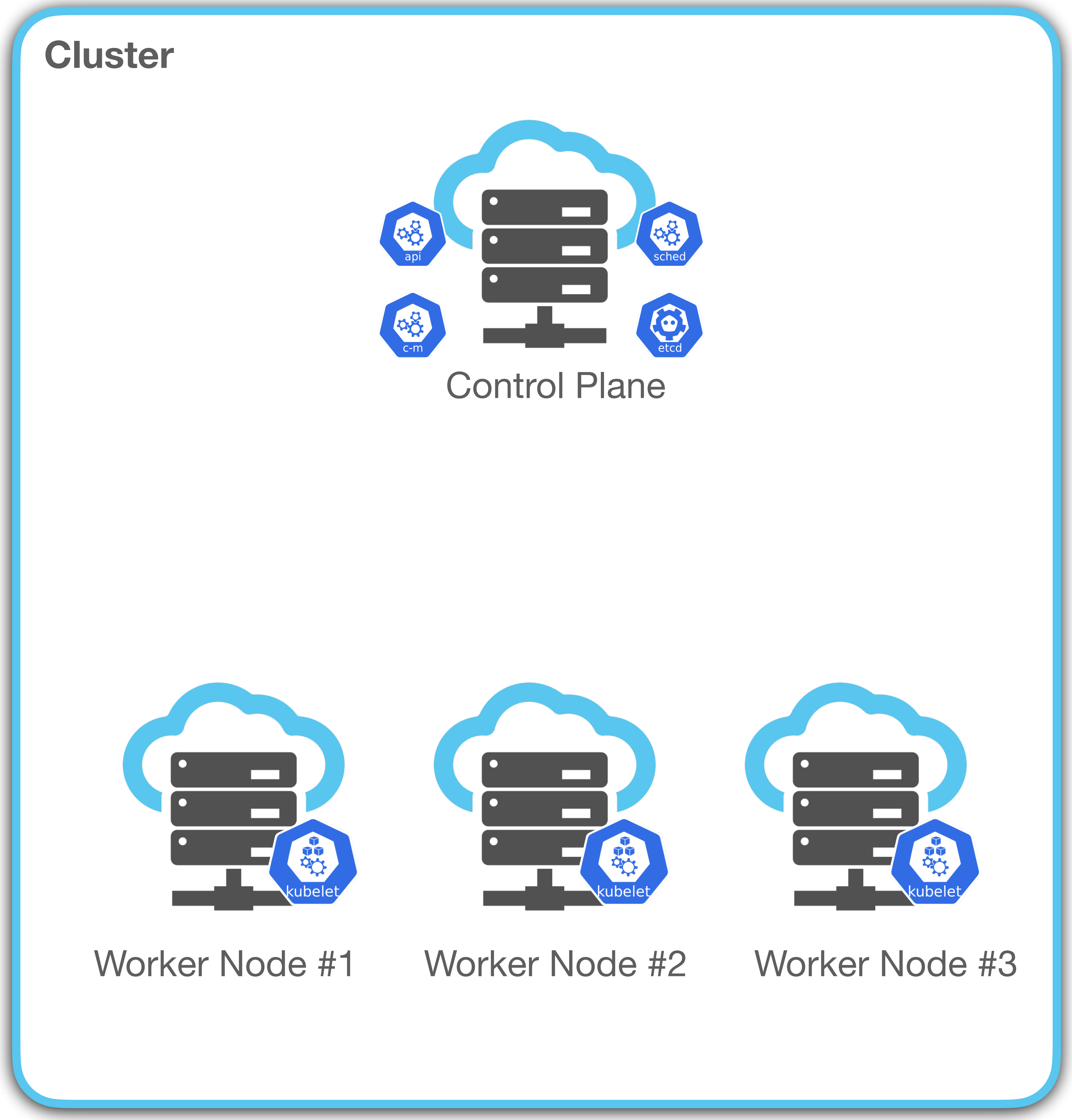
Controller Manager, keeps track of what's happening in the cluster

sched

Scheduler, ensures Pods placement, decides where a Pod should be scheduled

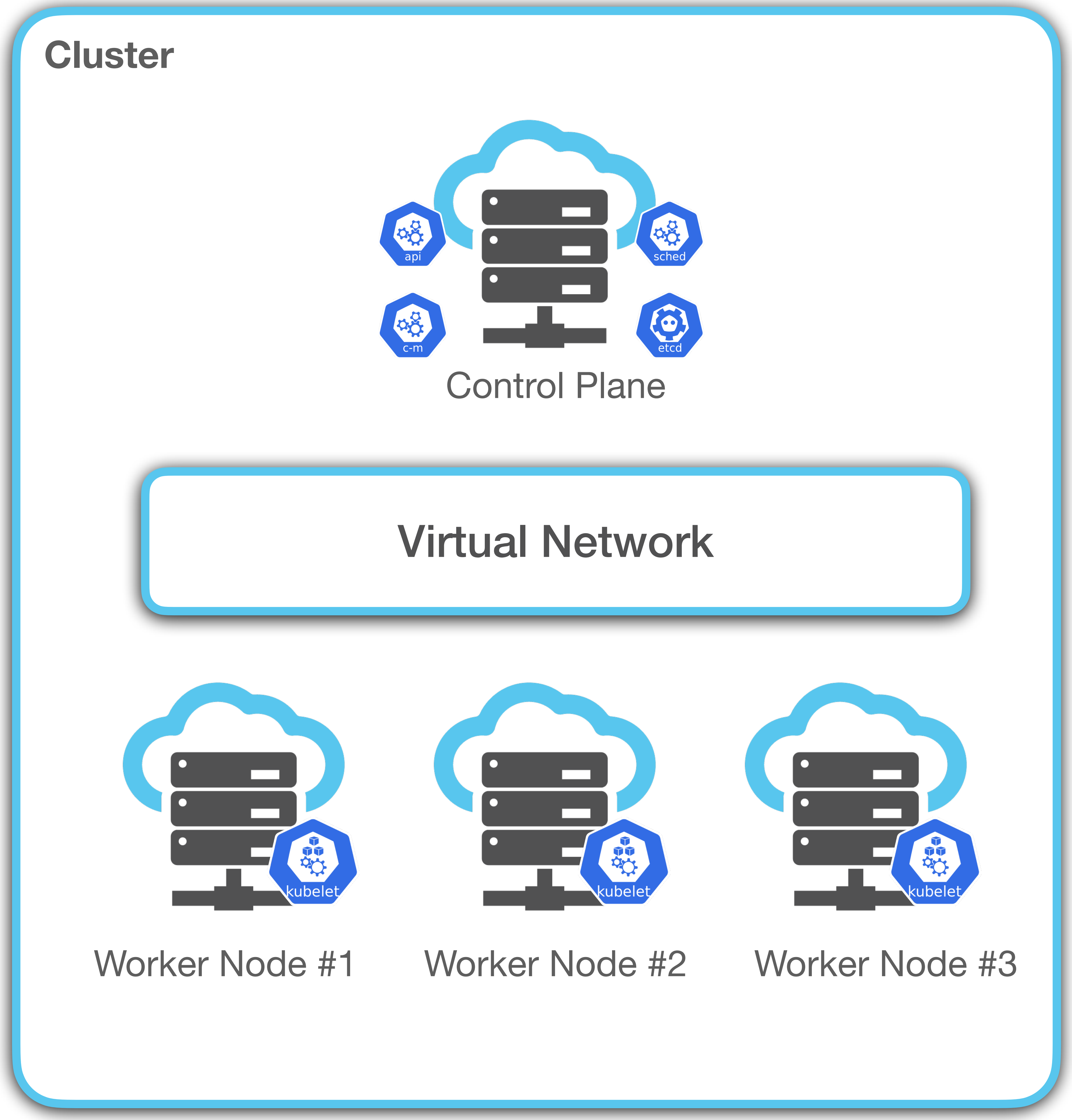
etcd

etcd, key-value store, holds the current state of the cluster, configuration and statuses



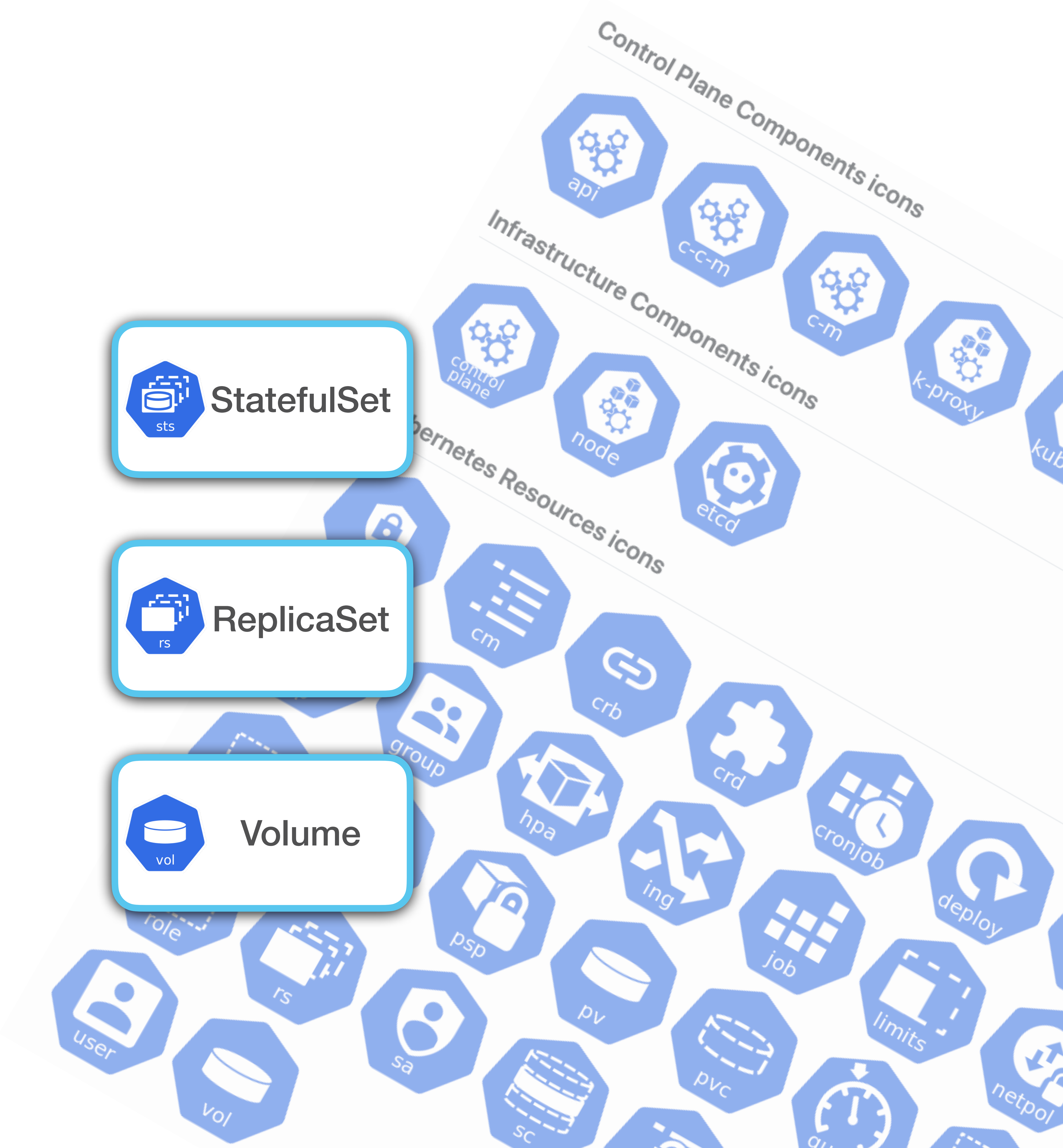
Kubernetes Architecture

Virtual Network enables the communication, spans all Nodes in the cluster. Turns all Nodes into one big machine, basically forms the cluster.



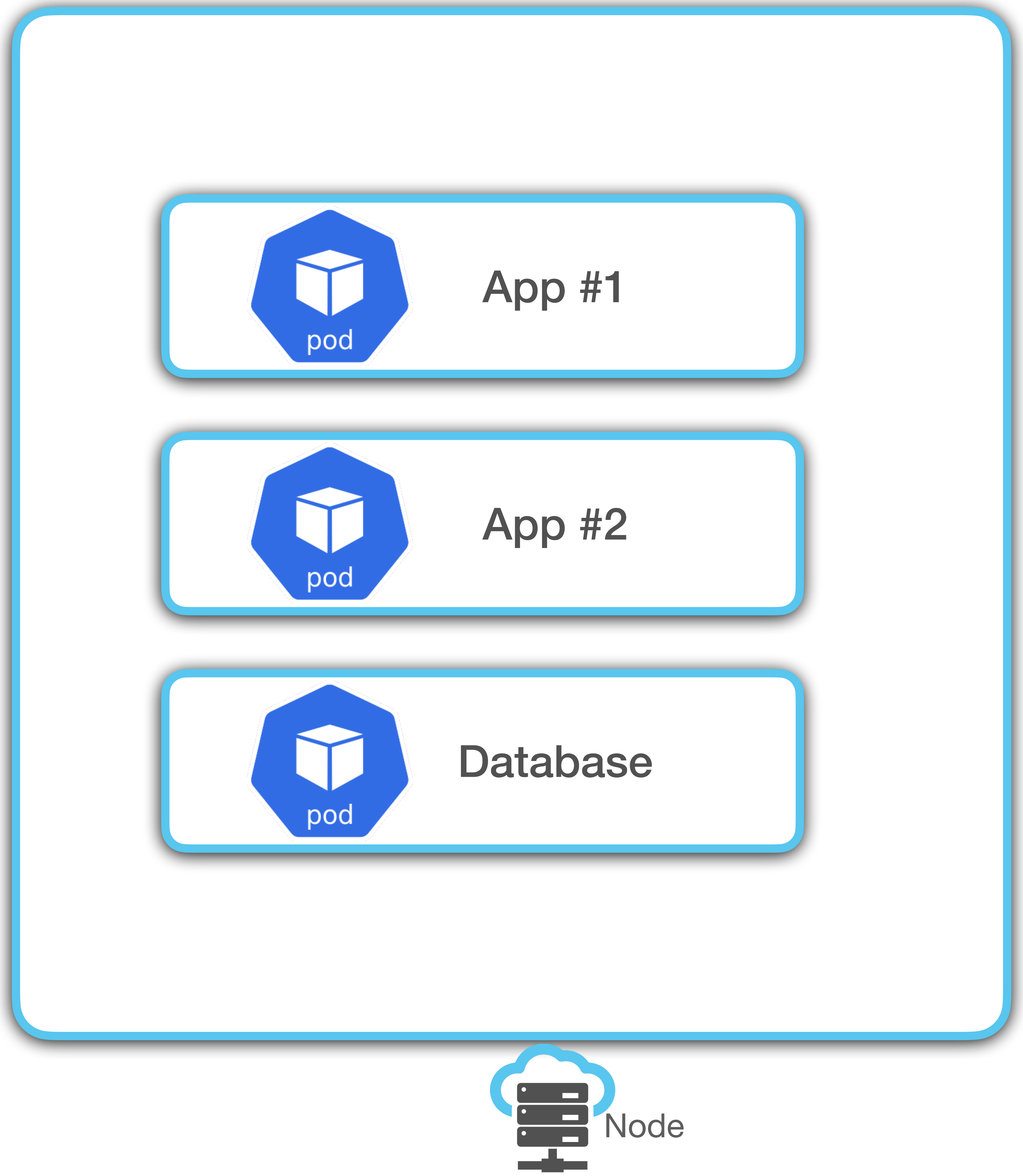
K8s Components

- pod Pod
- cm ConfigMap
- sts StatefulSet
- svc Service
- secret Secret
- rs ReplicaSet
- ing Ingress
- deploy Deployment
- vol Volume



Kubernetes Components

Pod



Smallest unit in K8s

Abstraction over a container

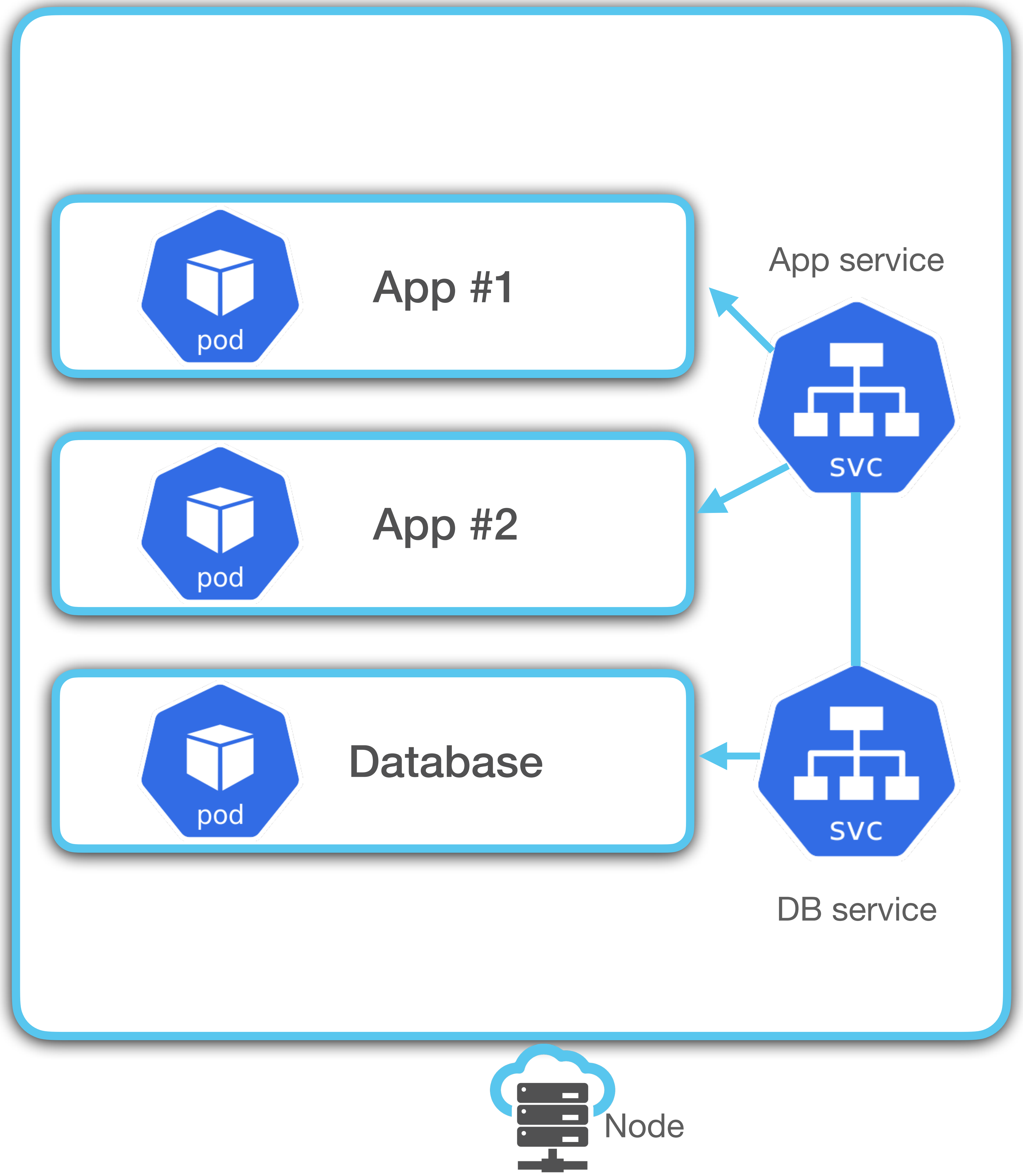
K8s abstracts away the container runtime

Usually 1 application per Pod (Not always tho…)

Each Pod has it’s own IP, internal

They are ephemeral (new IP on recreation)

Service



Permanent IP address

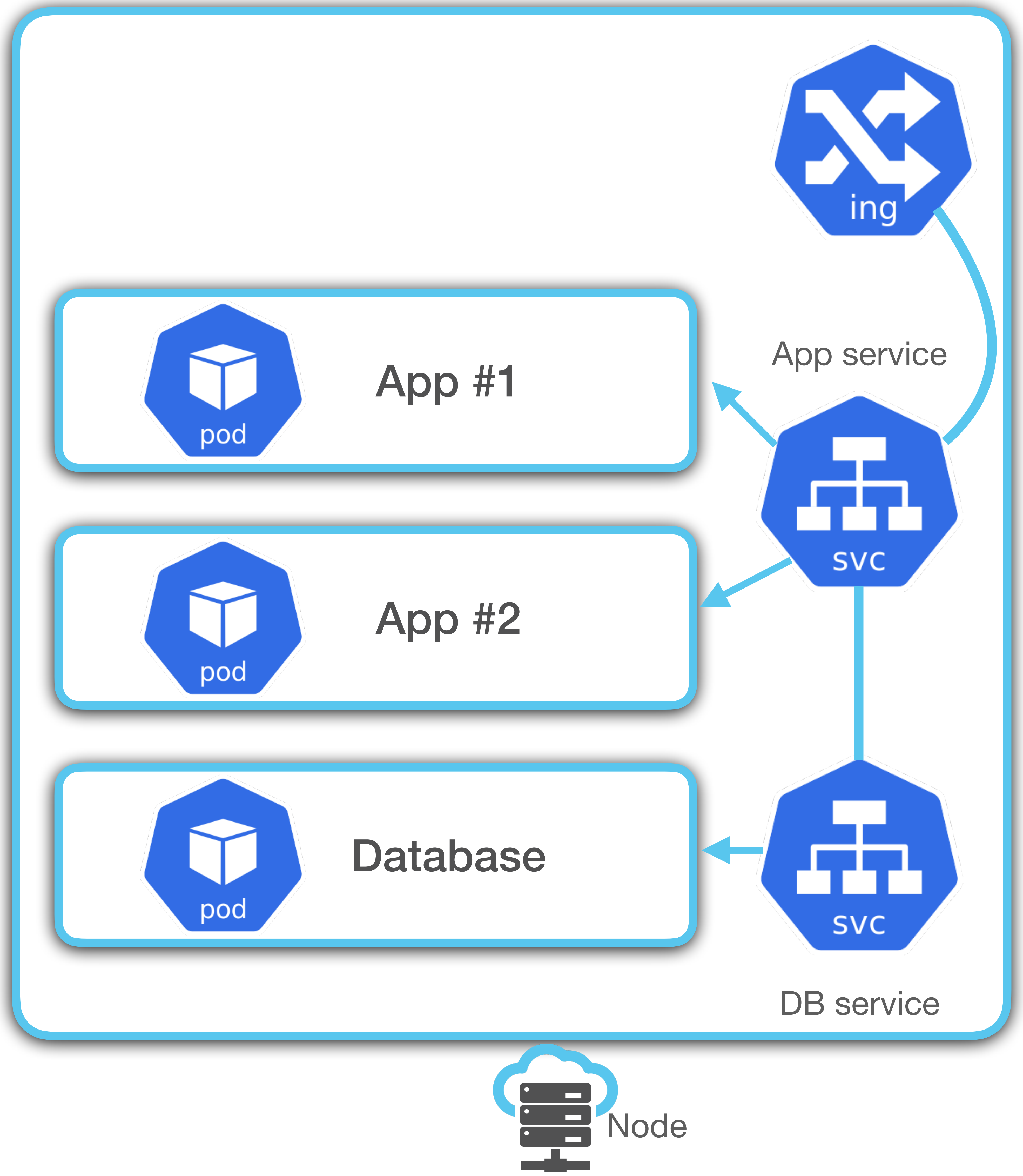
Lifecycle of Pods and Services are not connected

A Service selects the Pods

ClusterIP, NodePort, LoadBalancer

Internal services vs External services

Ingress



External service: http://node-ip:port

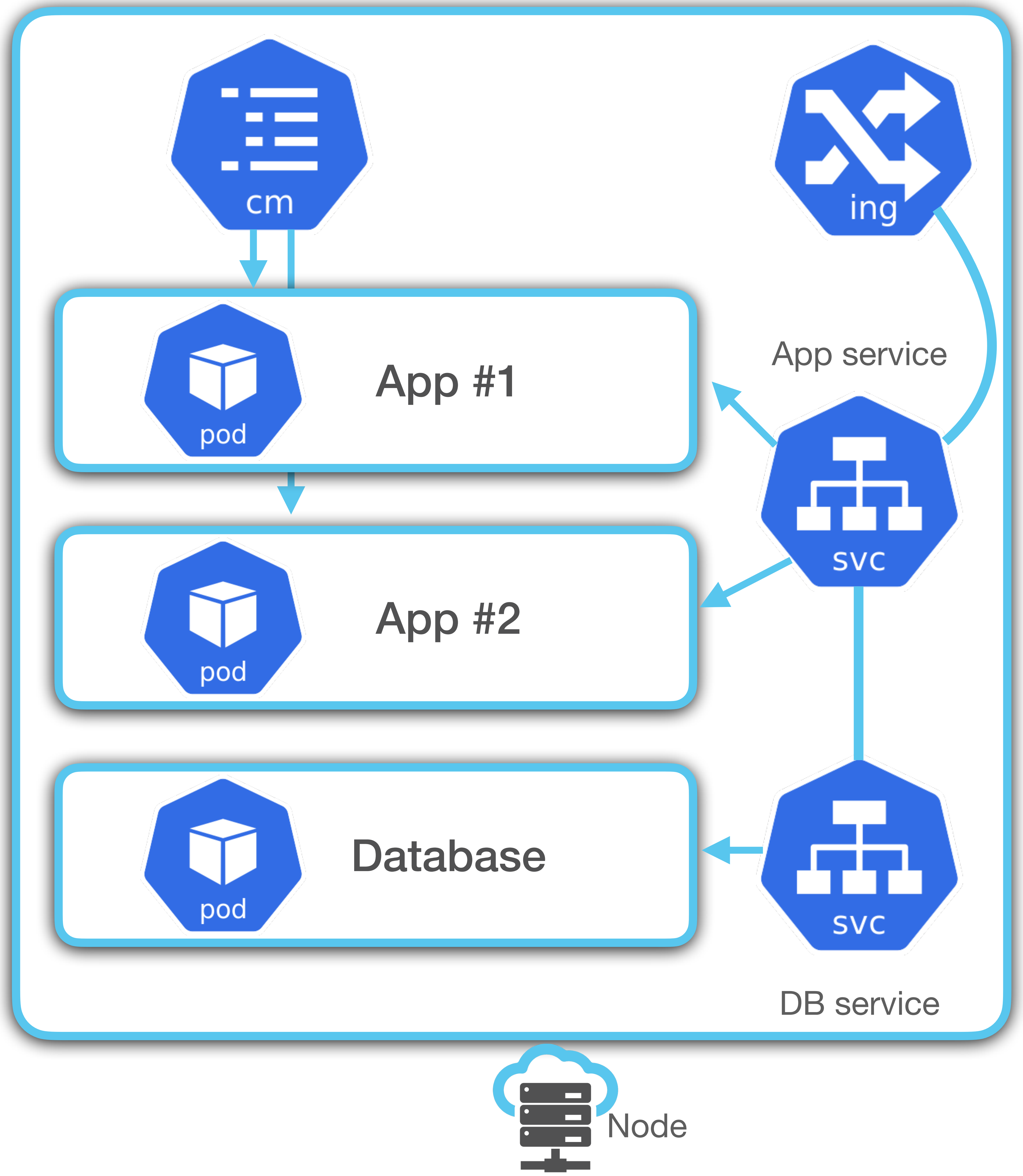
We want: https://my-app.com

Manages external access to the services in a cluster, typically HTTP

Exposes HTTP and HTTPS routes from outside to services within

Traffic routing is controlled by rules defined on the Ingress resource

ConfigMap



App tries to reach DB, needs configuration

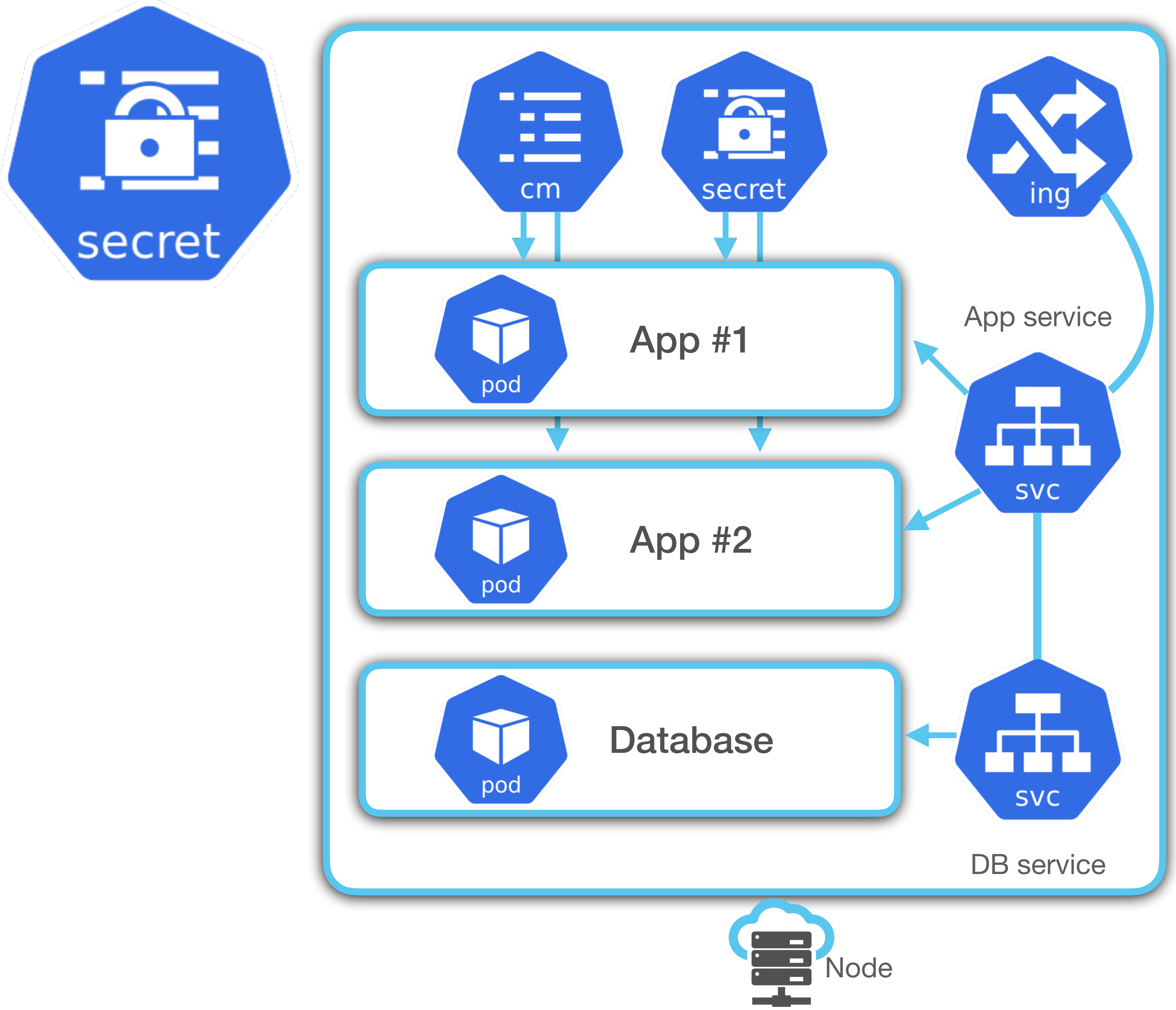
External configuration for applications

Key-value pairs, files, needs to be referenced in the pod

Plain text

Non-confidential data only!

Secret



Used to store secret data

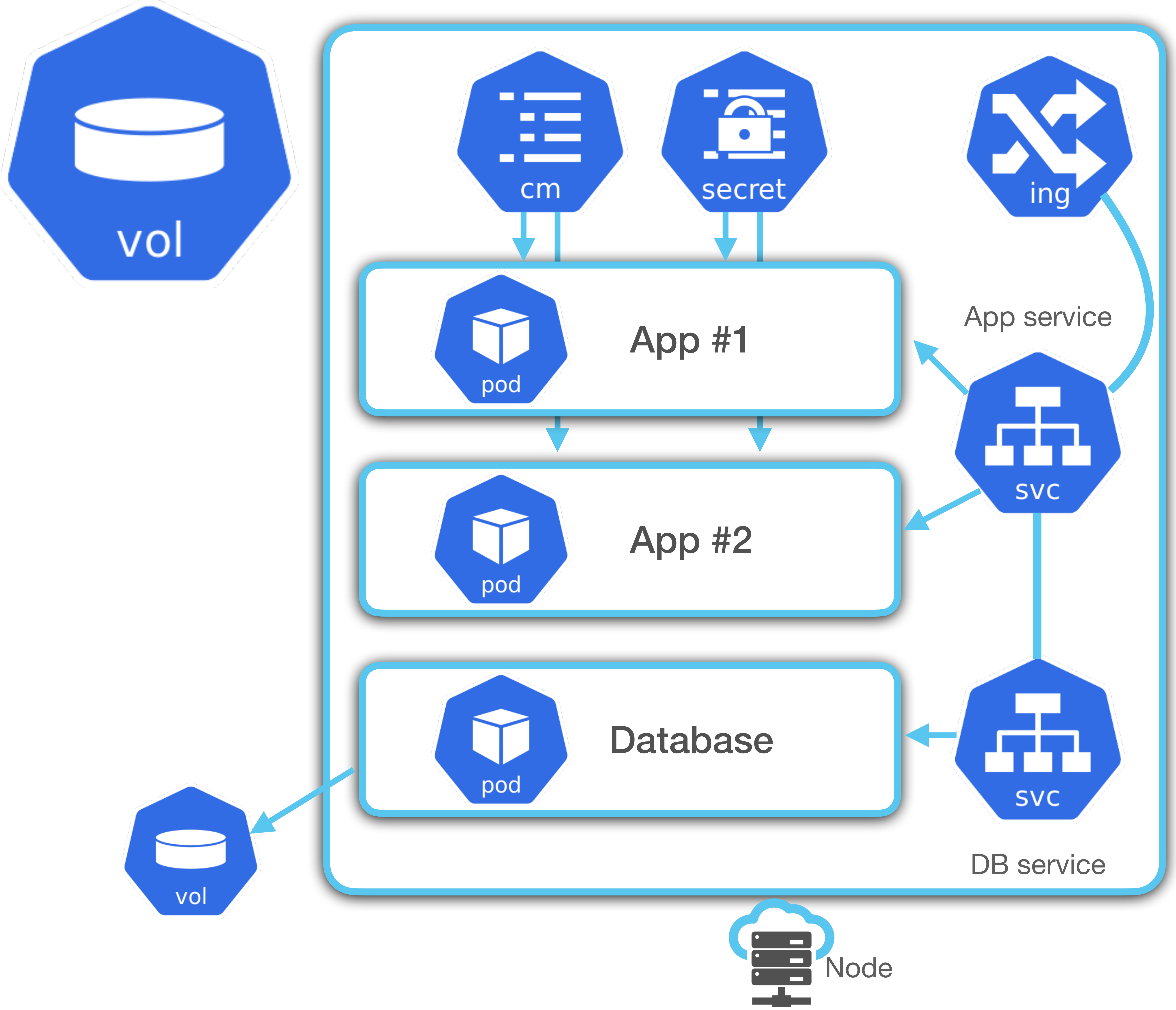
Not plain text, but only base64

Built-in security mechanisms are not enabled by default

Still not secure out of the box, needs to be referenced to tmpfs

Key-value pairs, files, certificates, etc…

Volume



DB gets restarted, data is gone

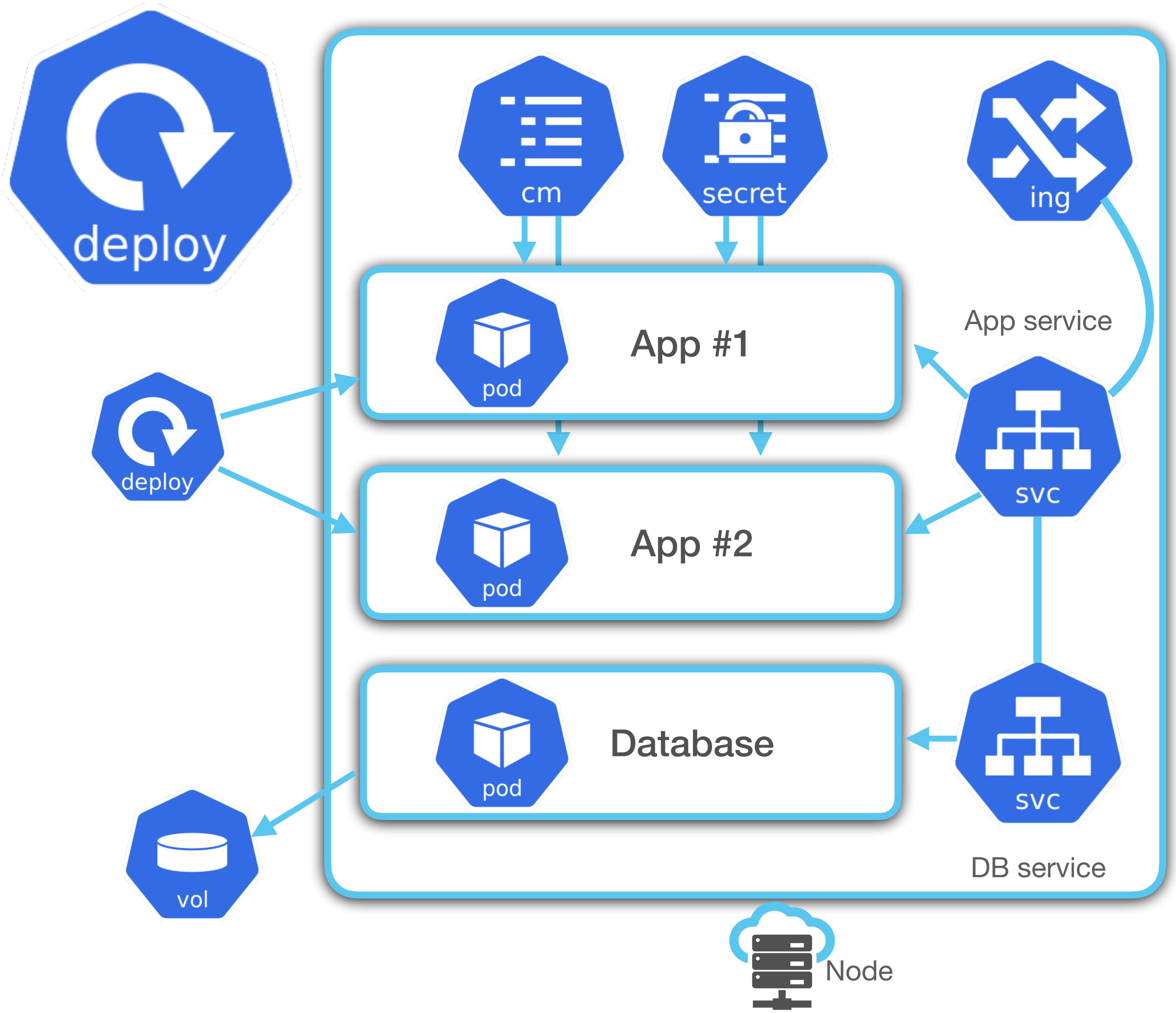
Attaches physical storage on a hard drive to the Pod

Storage can be even outside the cluster

Has to be attached to the Pod

K8s doesn’t manage data persistence, you are responsible for it

Deployment



App dies, needs to be restarted, or I just want multiple replicas

Blueprint for Pods

Replicas should be connected to same service, reproducibly

rs

Creates a ReplicaSet for stateless apps

sts

StatefulSets for stateful apps or databases

Kubernetes configuration

Kubernetes configuration

All configuration goes through the API Server on a Master Node

This is the only entry point into the cluster for configuration requests

Requests should be in yaml or in json

Kubernetes configuration

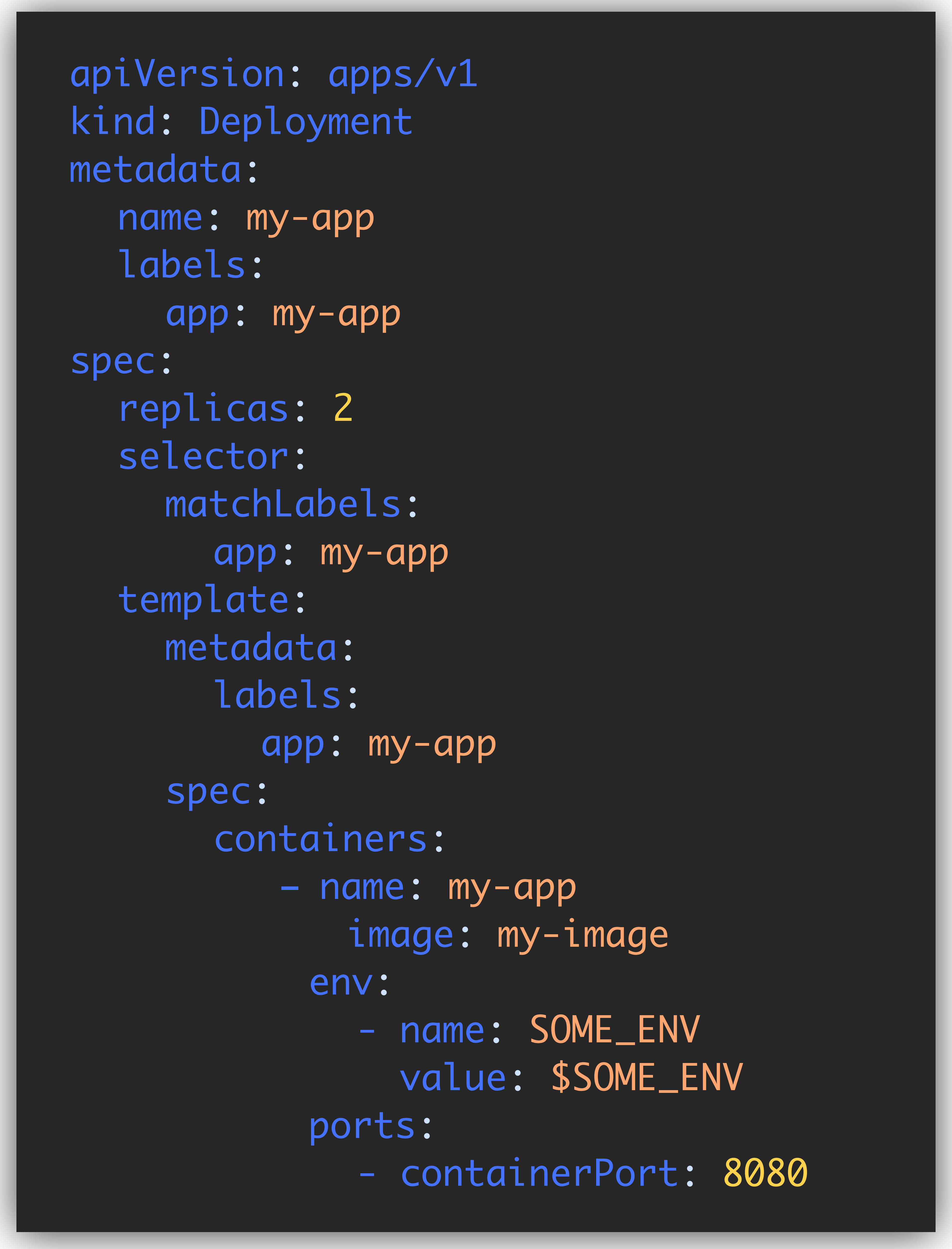
Declarative, K8s tries to fulfil it, Controller Manager checks state

Every config file has: metadata

Every config file has: specification

Every config file has: declarations

Implicitly contains status as well from etcd



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
  labels:
    app: my-app
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
    spec:
      containers:
         - name: my-app
            image: my-image
          env:
            - name: SOME_ENV
              value: $SOME_ENV
          ports:
            - containerPort: 8080
```

Workshop