

Kubernetes Overview

Leon Jalfon

DevOps & Cloud Solution Architect

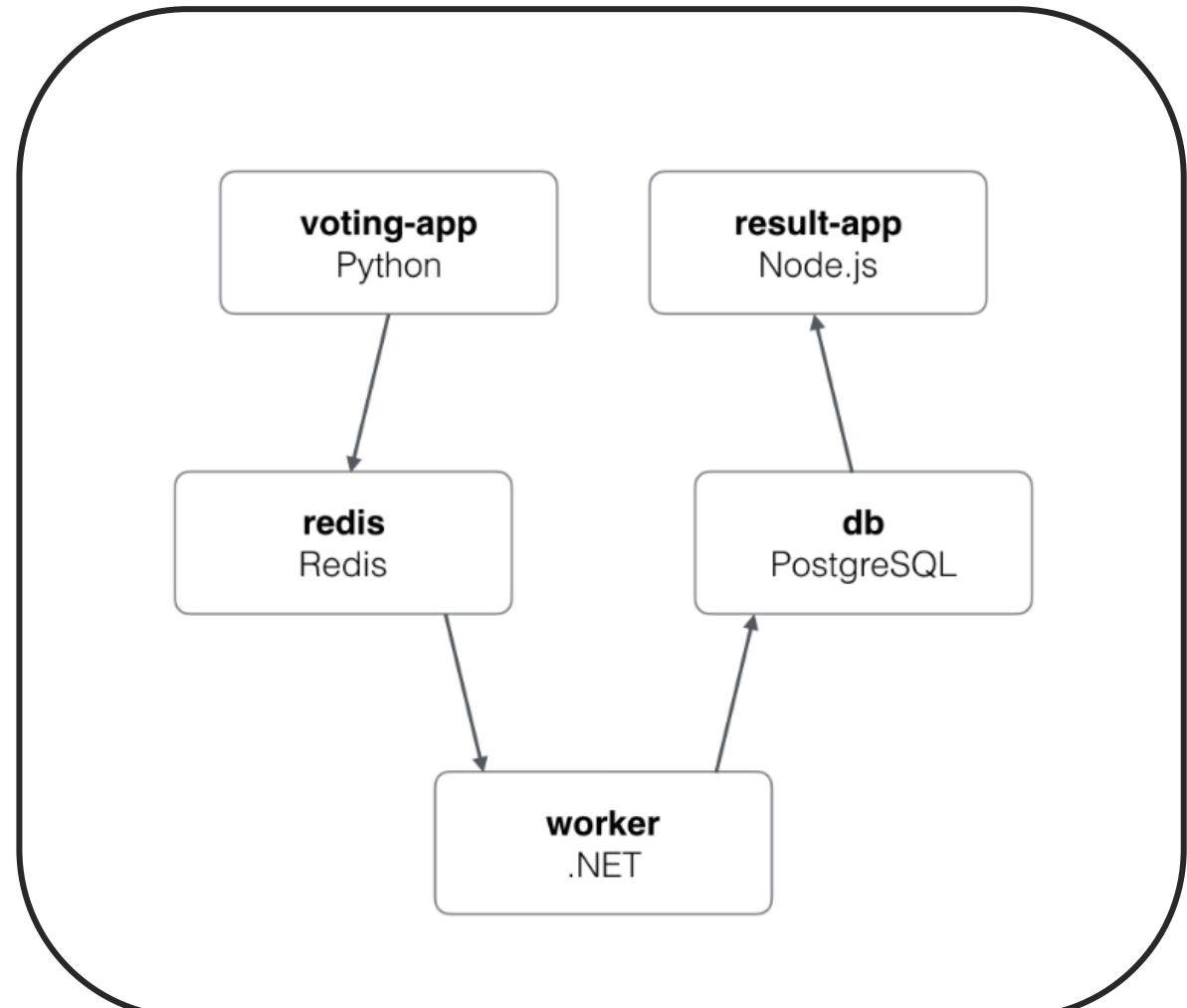
leonj@sela.co.il

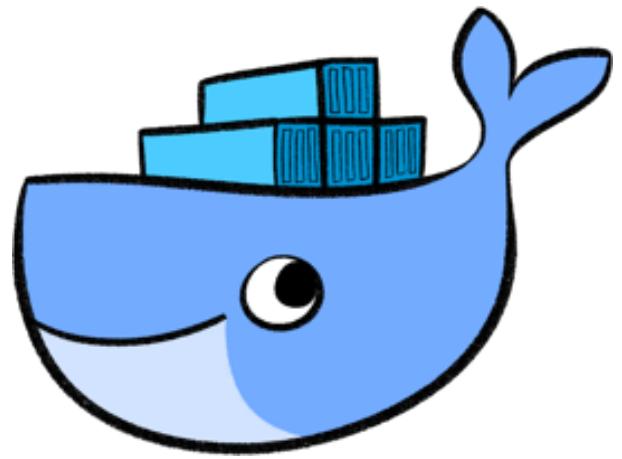


Agenda

- Docker in a Nutshell
- Kubernetes Architecture
- Pods
- Replicaset
- Deployments
- Services
- Helm
- What next?

Demo - Voting App

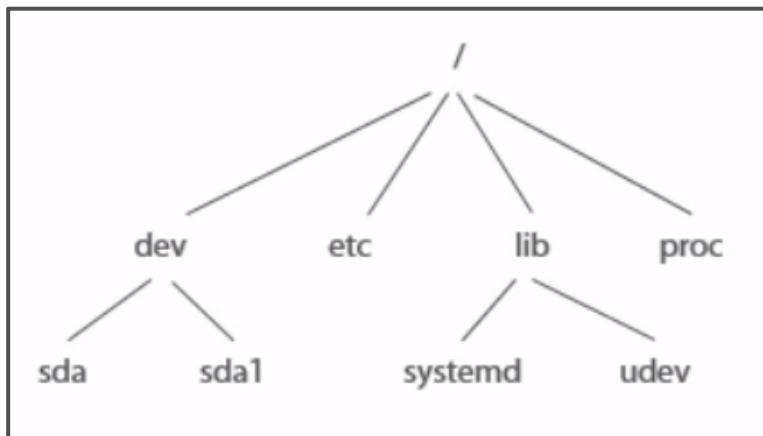




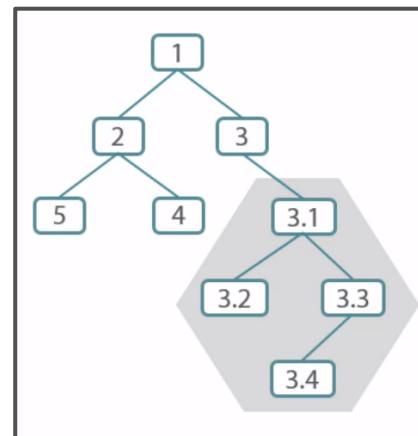
Docker

Linux Containers (LXC)

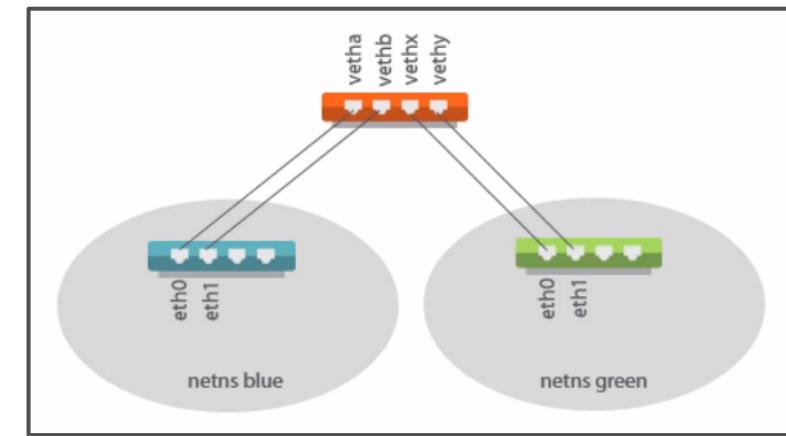
- Is an operating-system-level virtualization method.
- Run multiple isolated Linux systems (containers) on a control host using a single Linux kernel.



File System



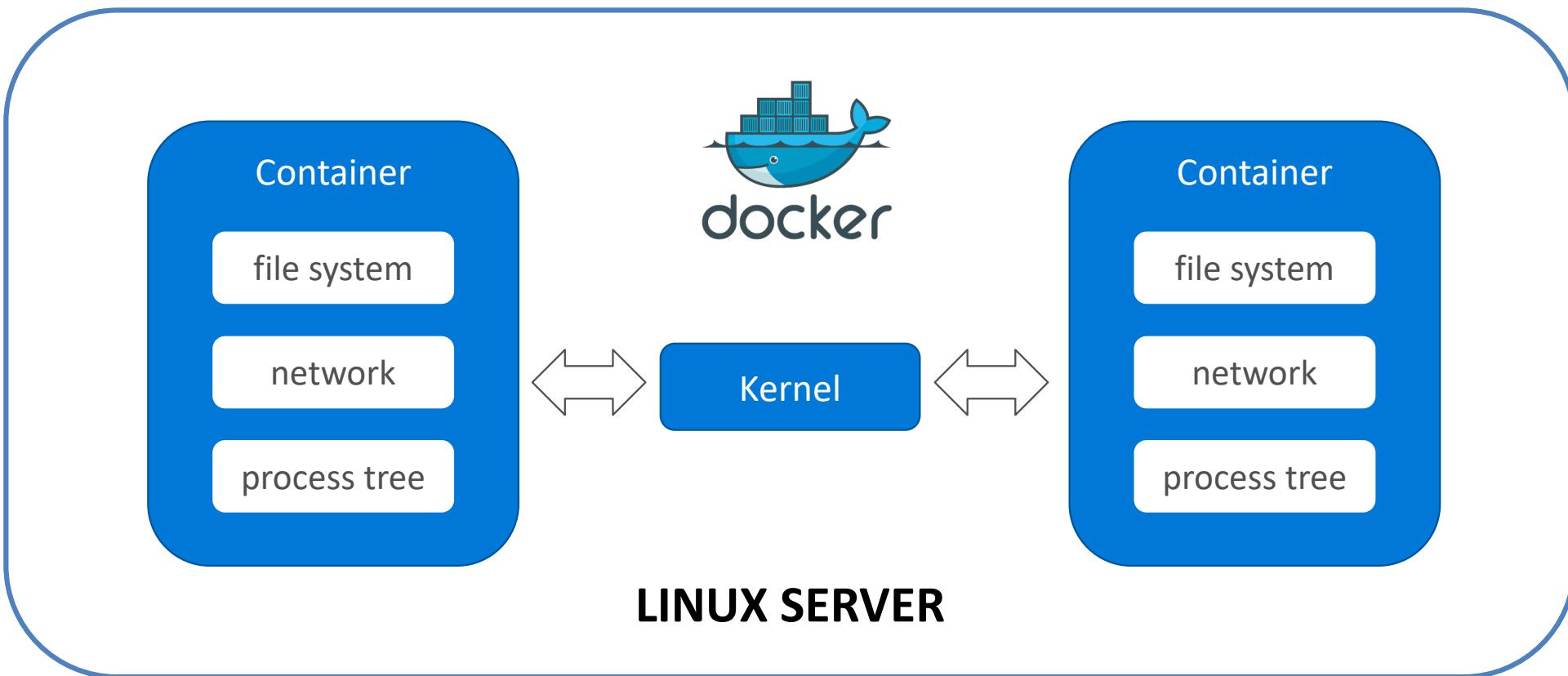
Process Tree



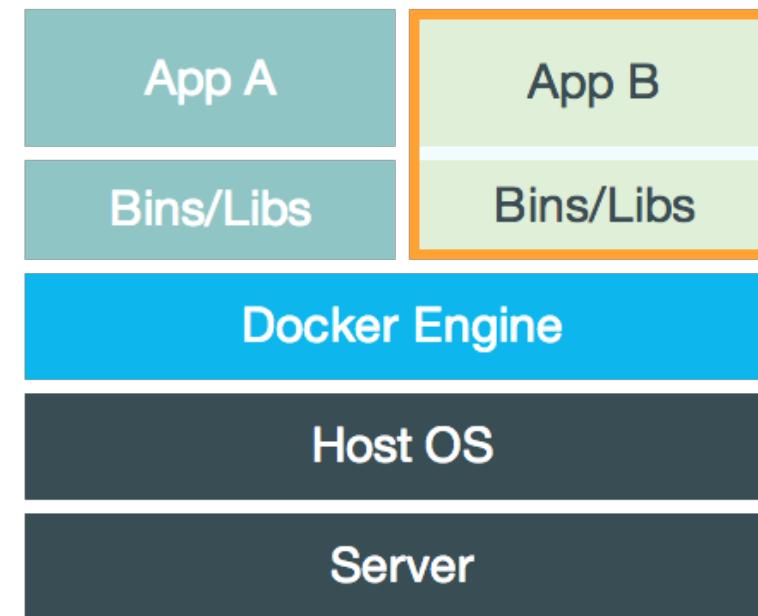
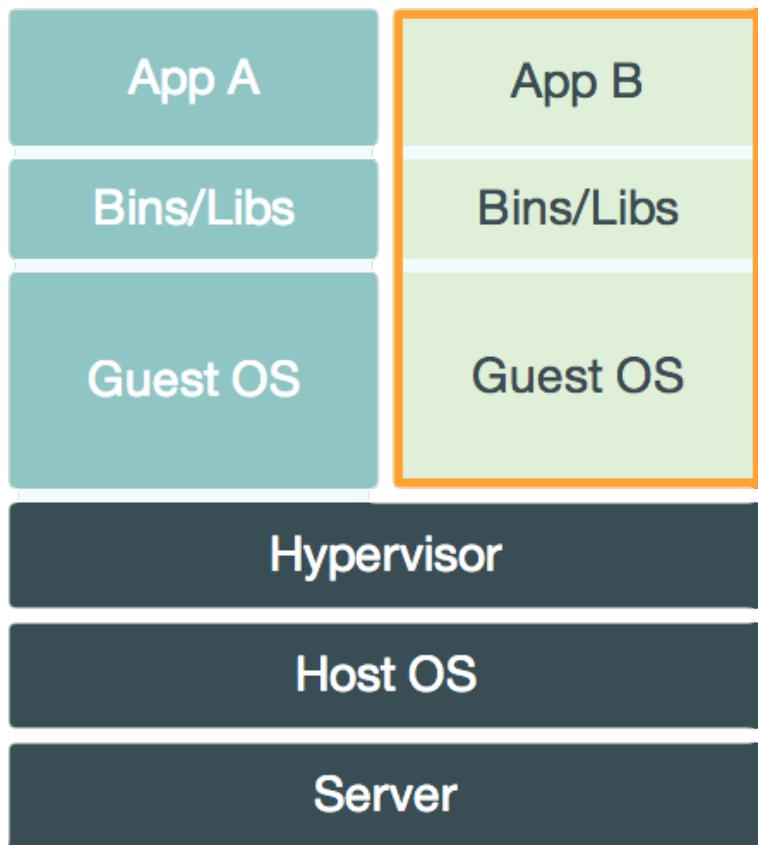
Networking Stacks

Docker

- Is an actual implementation of a container technology



Docker Containers VS Virtual Machines



Docker in Production



X



Containers Need Management

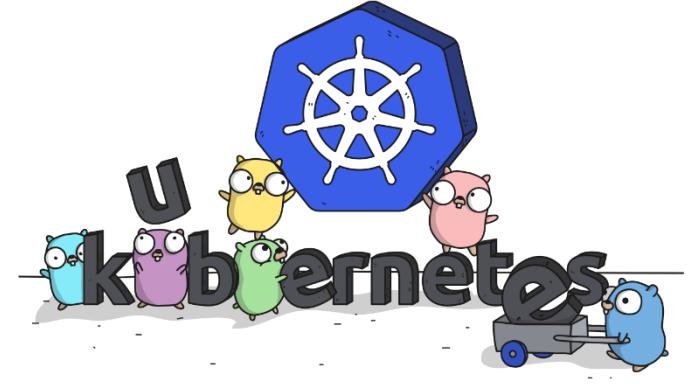
- Scheduling
- Failover
- Scaling
- Load Balancing
- Networking
- Upgrades
- Service Discovery
- Monitoring





Kubernetes

What is Kubernetes?

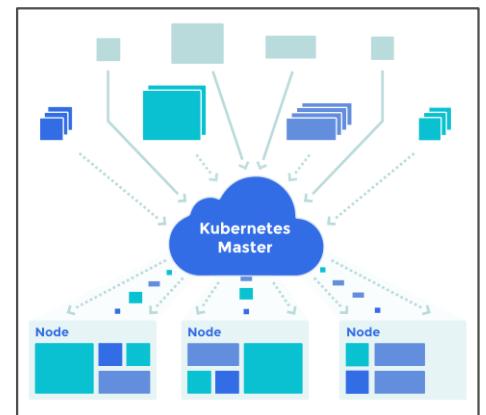


- Kubernetes = Greek for “Helmsman”
- Is a production-grade container orchestration system
- Google-grown, based on Borg and Omega
- Support multiple container runtimes
- Support multiple cloud and bare-machines environments
- Founded and operated by CNCF
- One of the most popular open source projects
- Written in Go

What Does Kubernetes Do?

- Abstract away the underlying hardware
- Manage your applications like cattle instead of like pets
- Is responsible for maintaining the desired state
- Improves reliability (scale, failover, self-healing)
- Better use of infrastructure resources
- Manage networking and volumes

Manages your infrastructure so you can concentrate on developing better applications!



Kubernetes ❤️ YAML

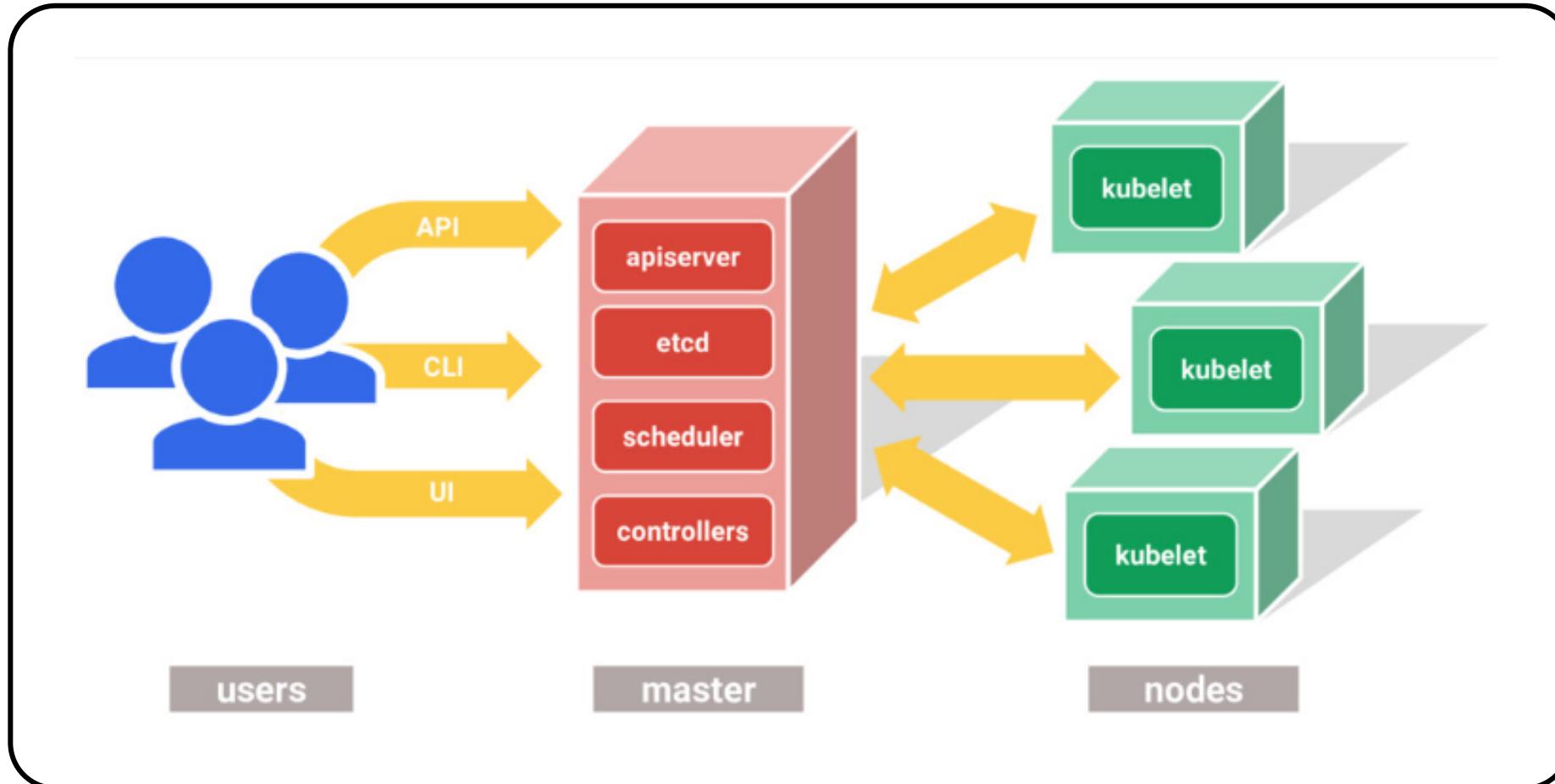
```
apiVersion: v1
kind: Pod
metadata:
  namespace: default
  name: sample-pod
spec:
  containers:
    - image: nginx
      name: container-name
```

```
apiVersion: apps/v1beta2
kind: ReplicaSet
metadata:
  name: nginx
  annotations:
    description: "nginx frontend"
  labels:
    app: nginx
    tier: frontend
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
      tier: frontend
  template:
    metadata:
      labels:
        app: nginx
        tier: frontend
    spec:
      containers:
        - name: nginx
          image: nginx:latest
          ports:
            - containerPort: 80
```

```
apiVersion: apps/v1beta2
kind: Deployment
metadata:
  name: nginx
  annotations:
    description: "nginx frontend"
  labels:
    app: nginx
    tier: frontend
spec:
  replicas: 3
  minReadySeconds: 10
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 5
      maxUnavailable: 2
  selector:
    matchLabels:
      app: nginx
      tier: frontend
  template:
    metadata:
      labels:
        app: nginx
        tier: frontend
    spec:
      containers:
        - name: nginx
          image: nginx:latest
          ports:
            - containerPort: 80
```

```
kind: Service
apiVersion: v1
metadata:
  name: nginx
spec:
  type: ClusterIP
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

Kubernetes Basic Architecture

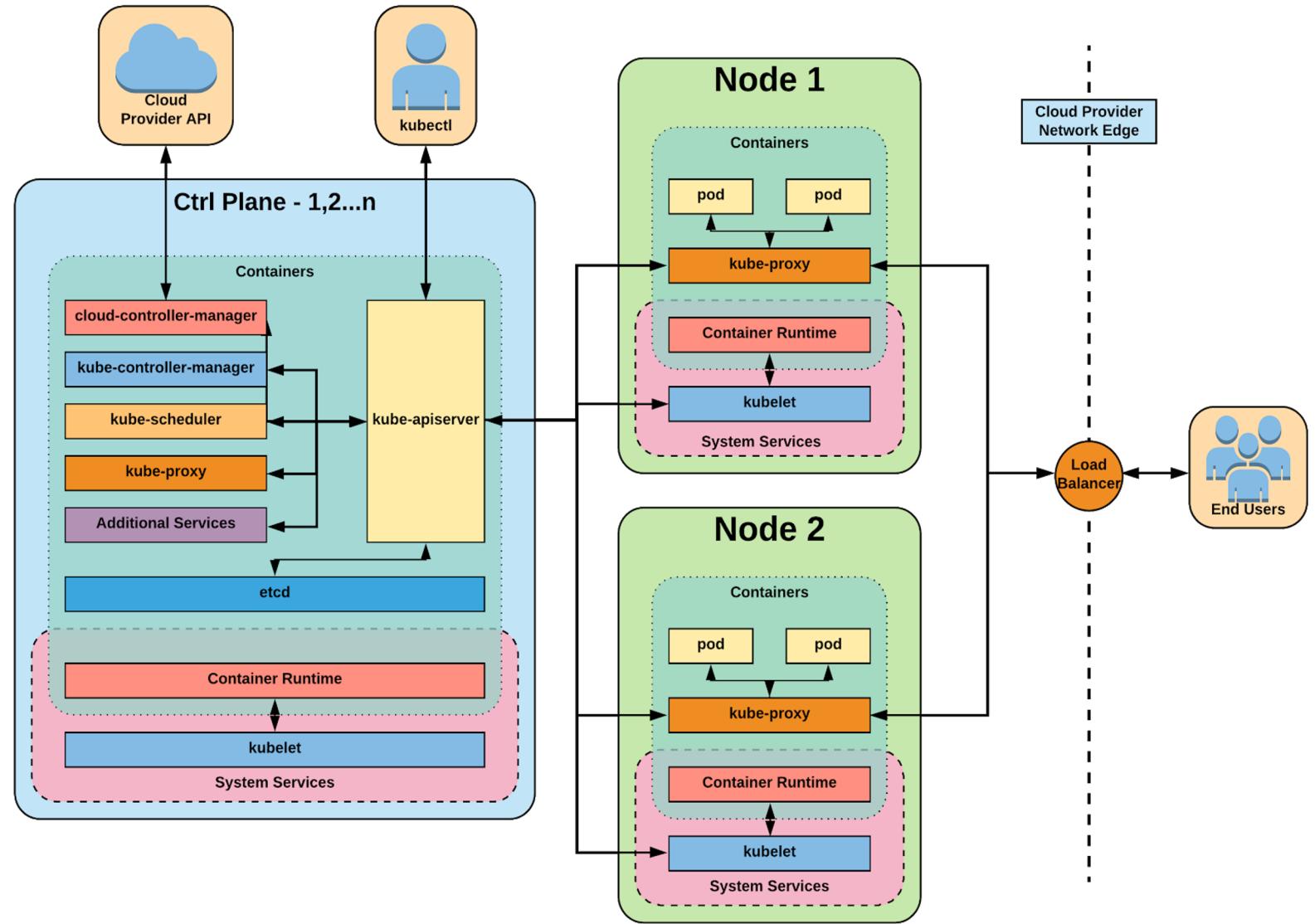


Master Components

- API Server
- etcd
- Controller Manager
- Kube-Scheduler

Worker Components

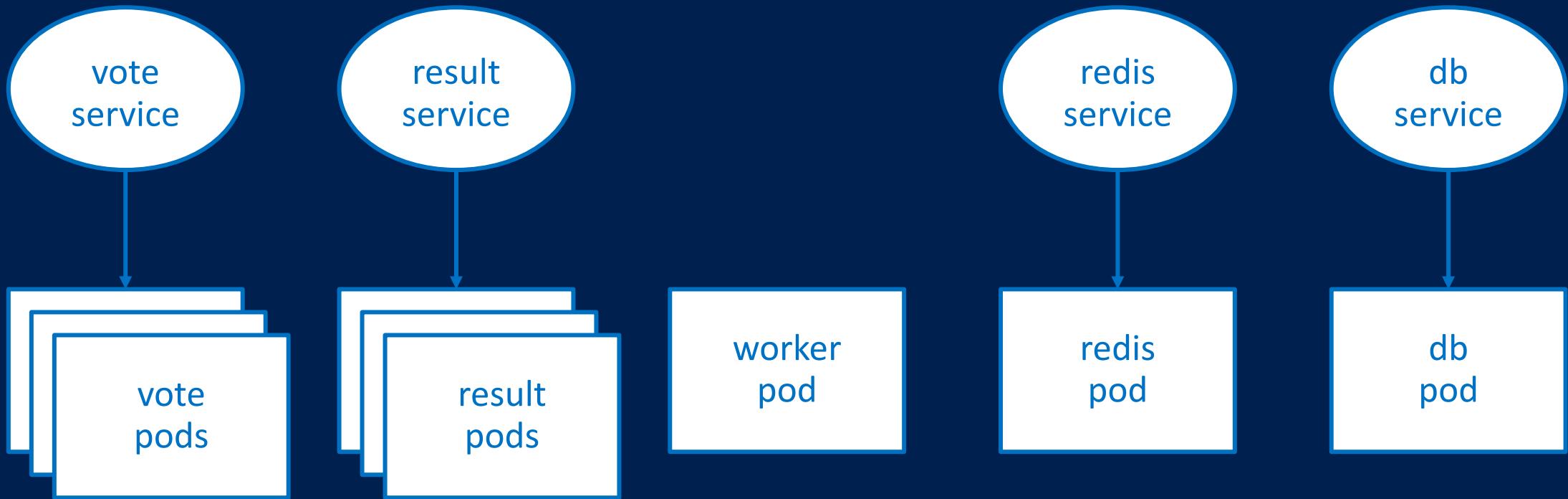
- Kubelet
- Kube-proxy
- Container Runtime



Kubernetes Building Blocks

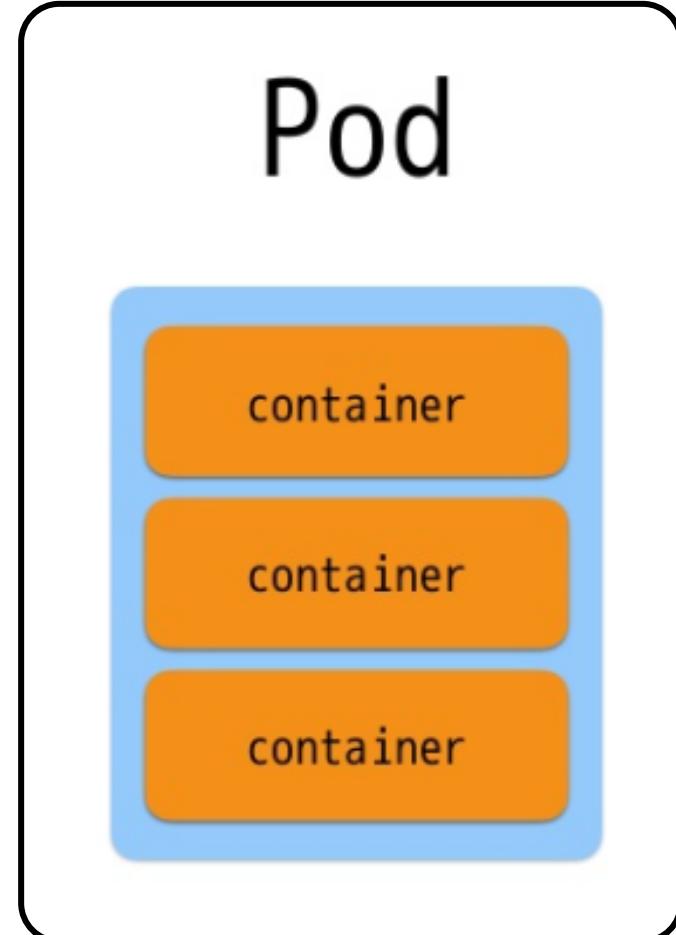


Demo - Voting App

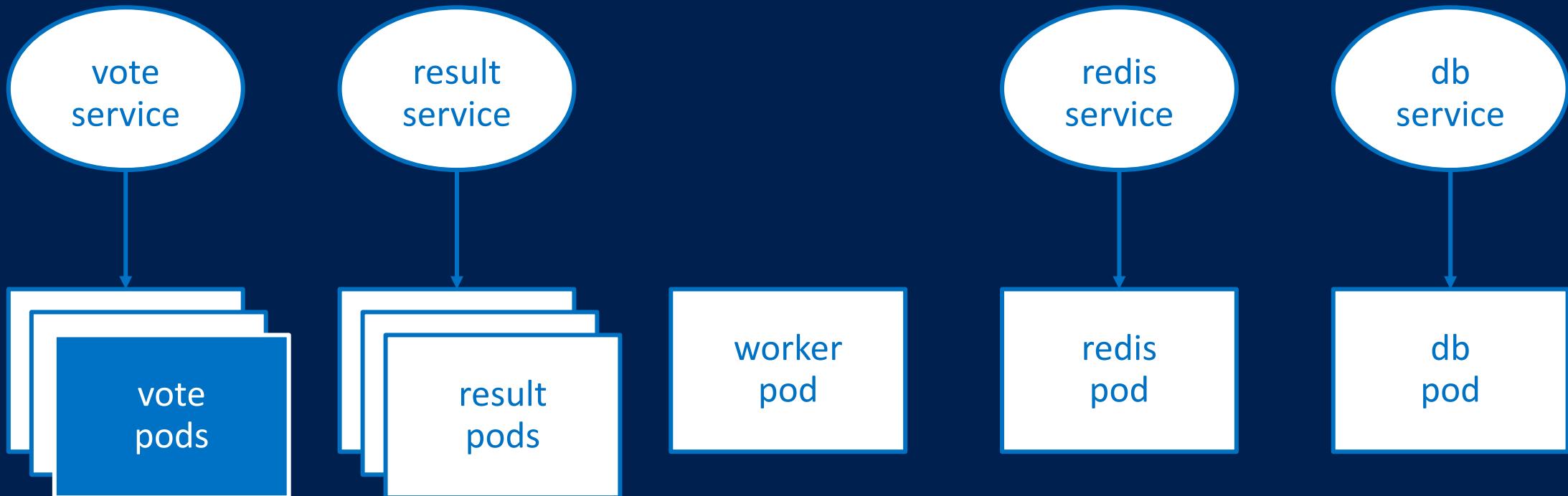


Core Concepts - Pod

- Basic and atomically schedulable building block of Kubernetes.
- It's comprised of one or more containers that share the same context (including the same IP)



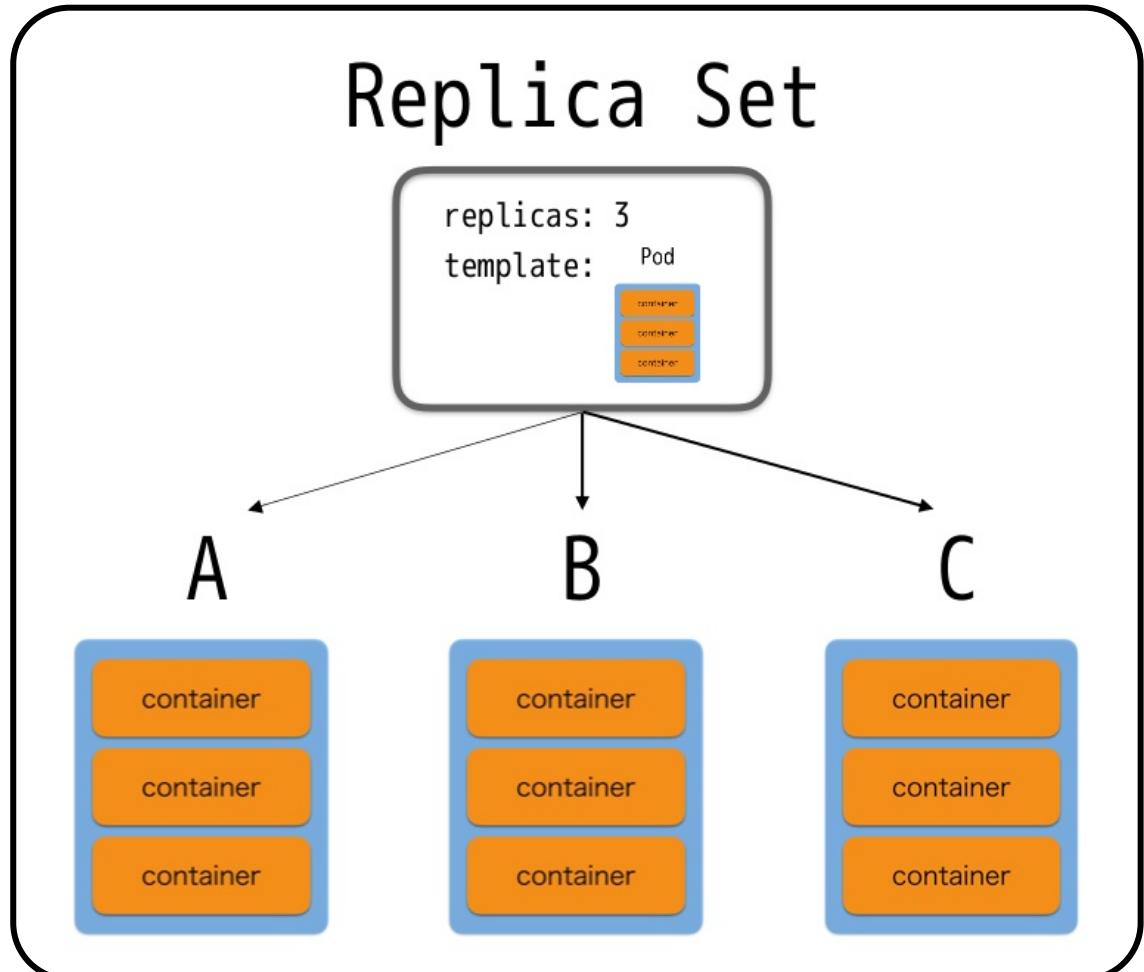
Demo - Voting App



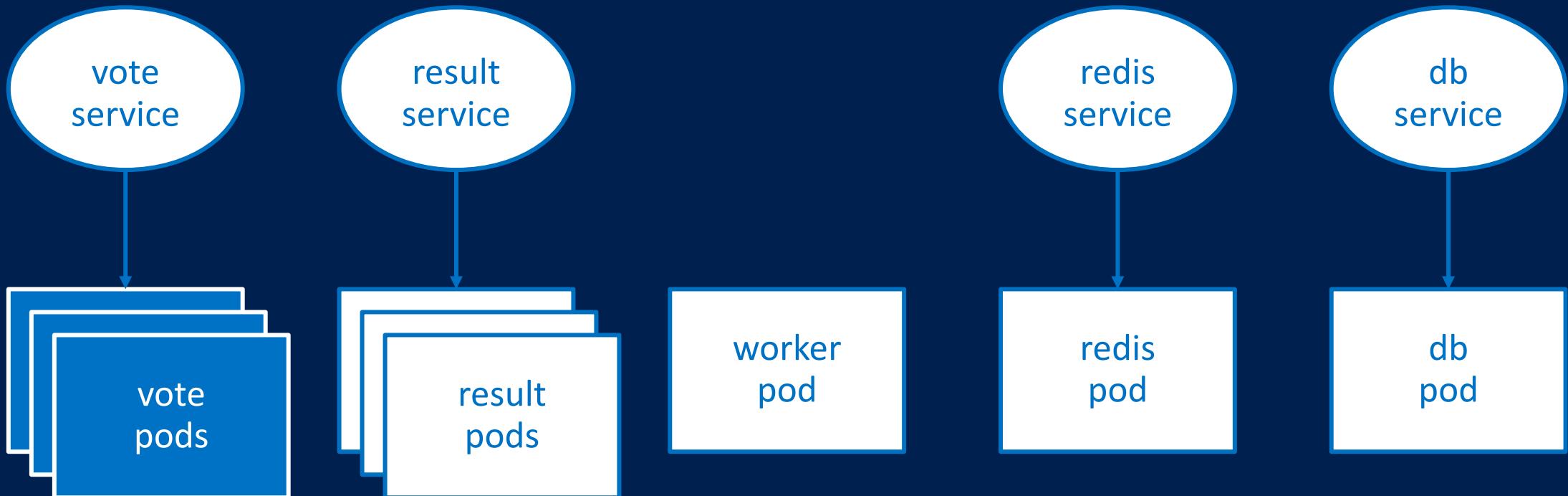
Core Concepts - Replica Set

- Method of managing pod replicas and their lifecycle:

- Scheduling
- Scaling
- Deletion

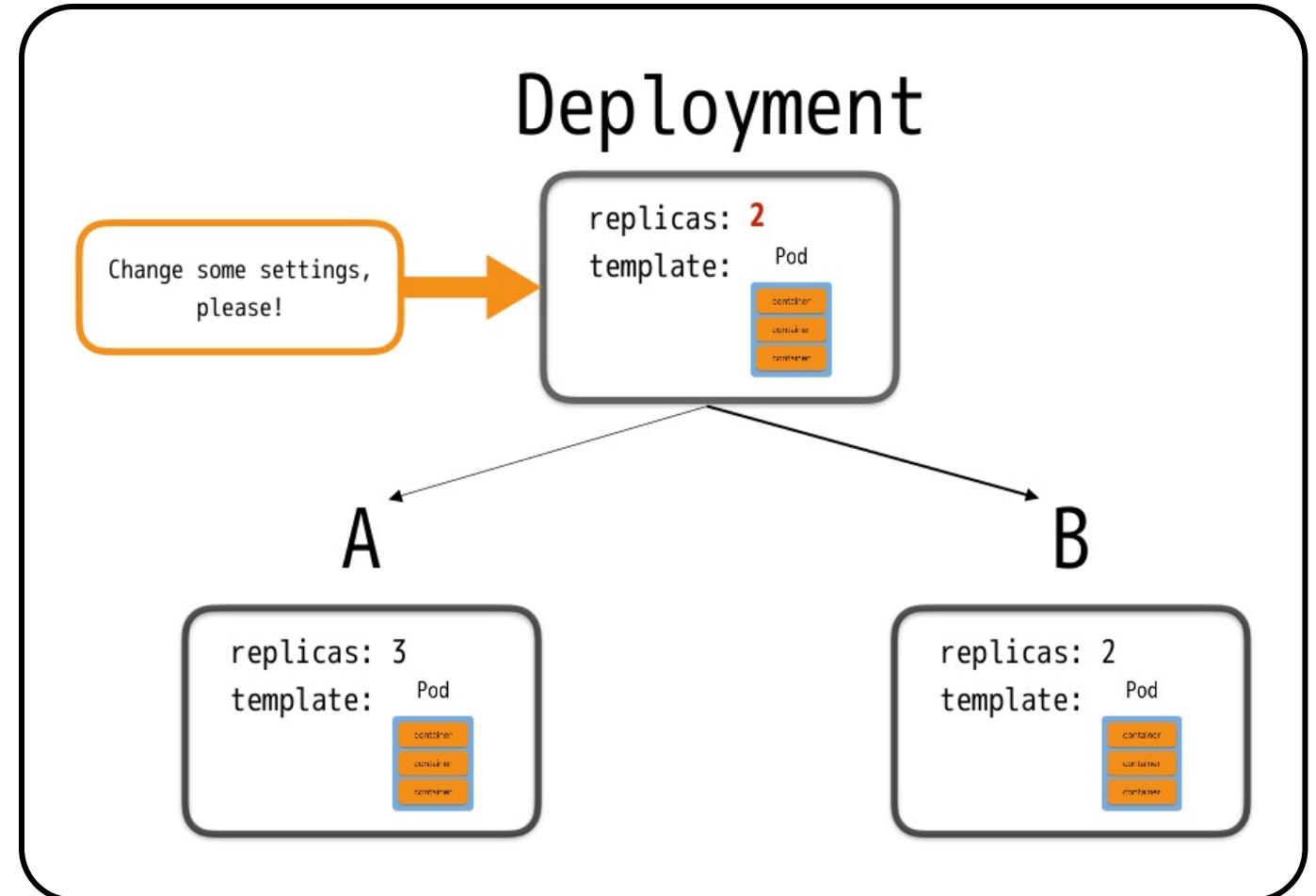


Demo - Voting App

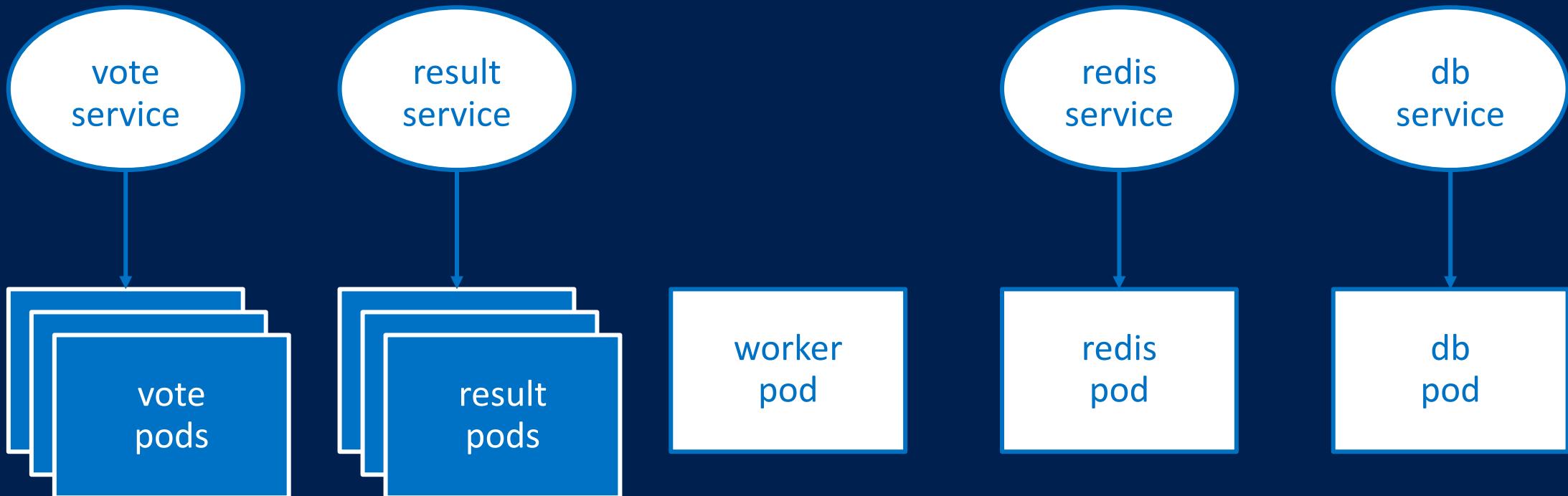


Core Concepts - Deployment

- Includes a pod template and a replica field
- Kubernetes will make sure the actual state always matches the desired state.

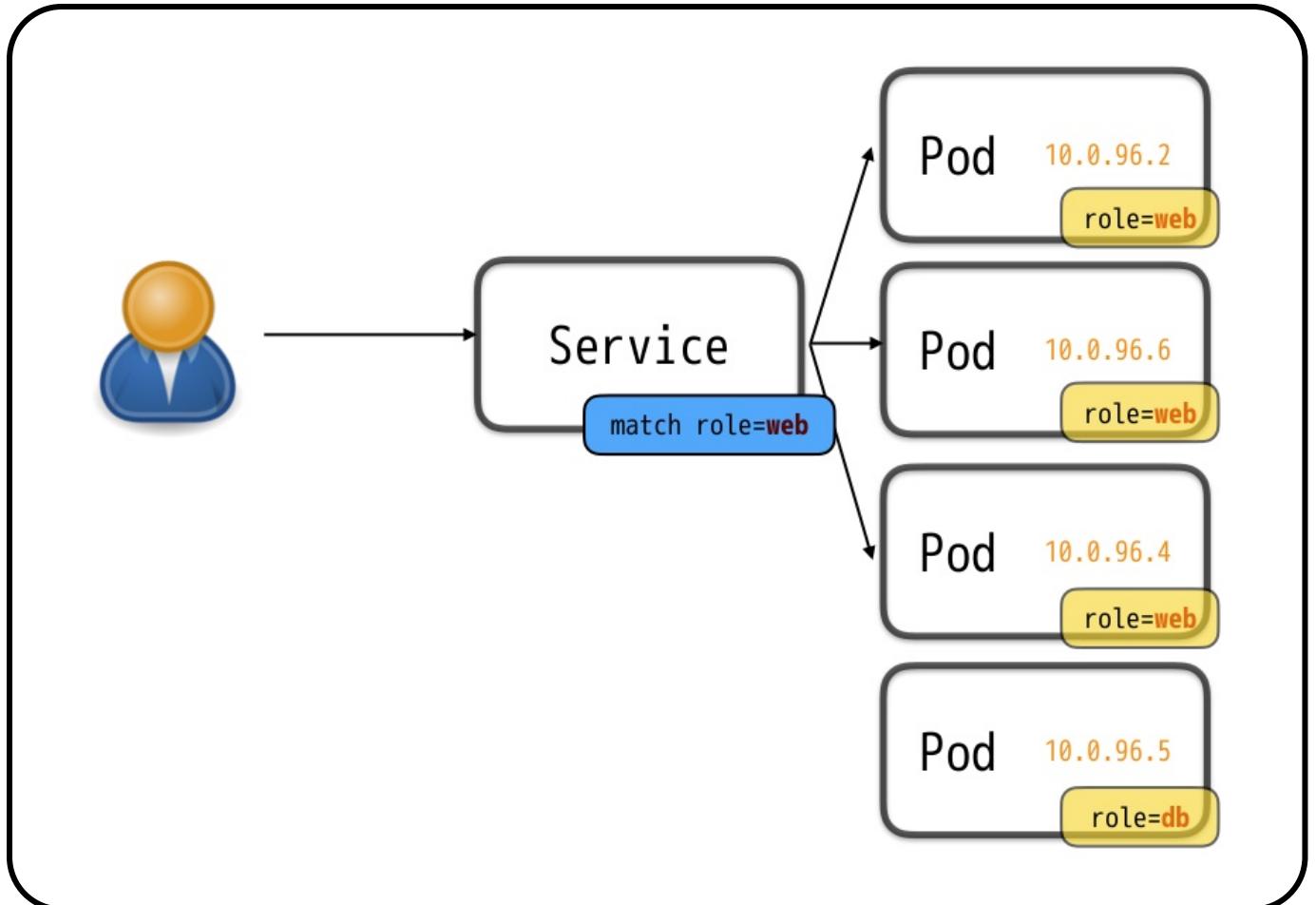


Demo - Voting App

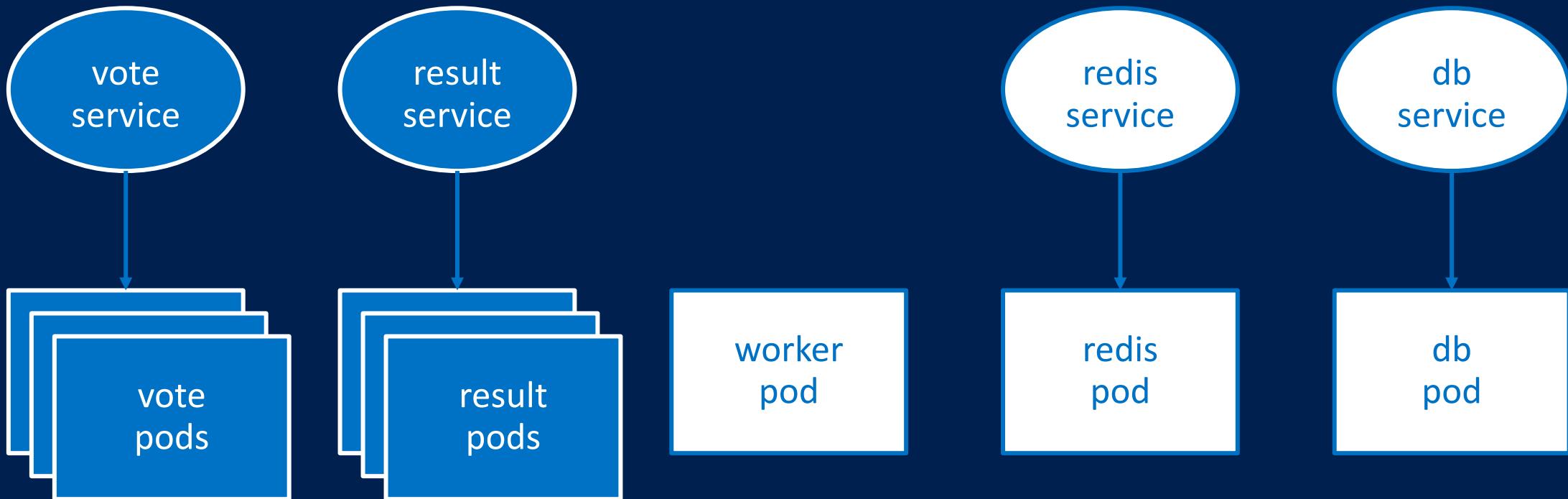


Core Concepts - Service

- Select pods by a matching label selector
- Provides a stable, immortal way to talk to your application.

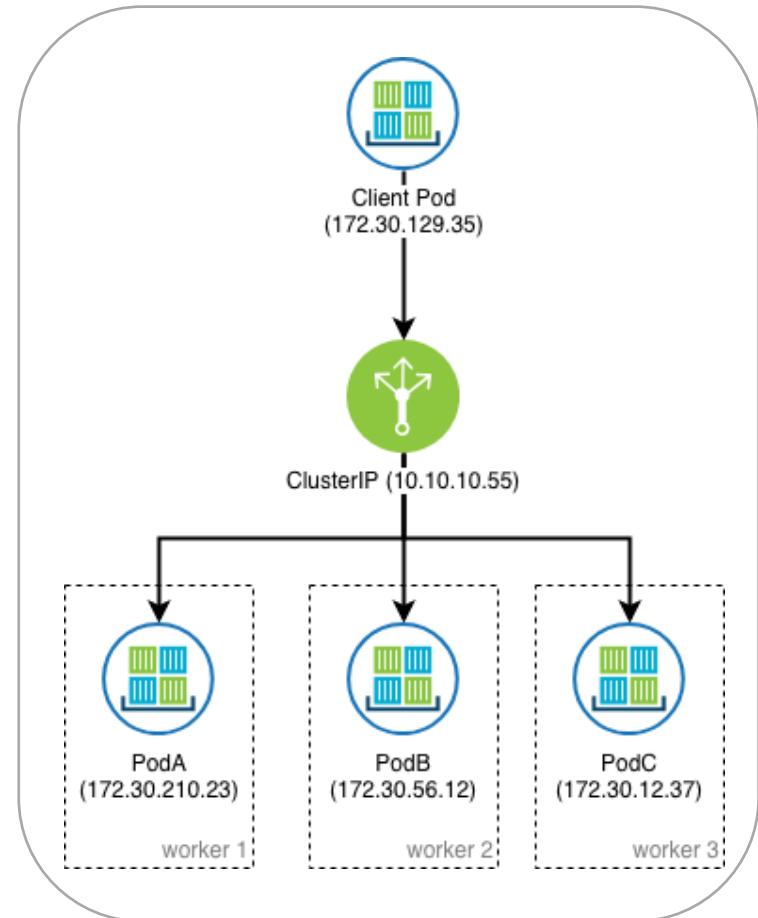


Demo - Voting App



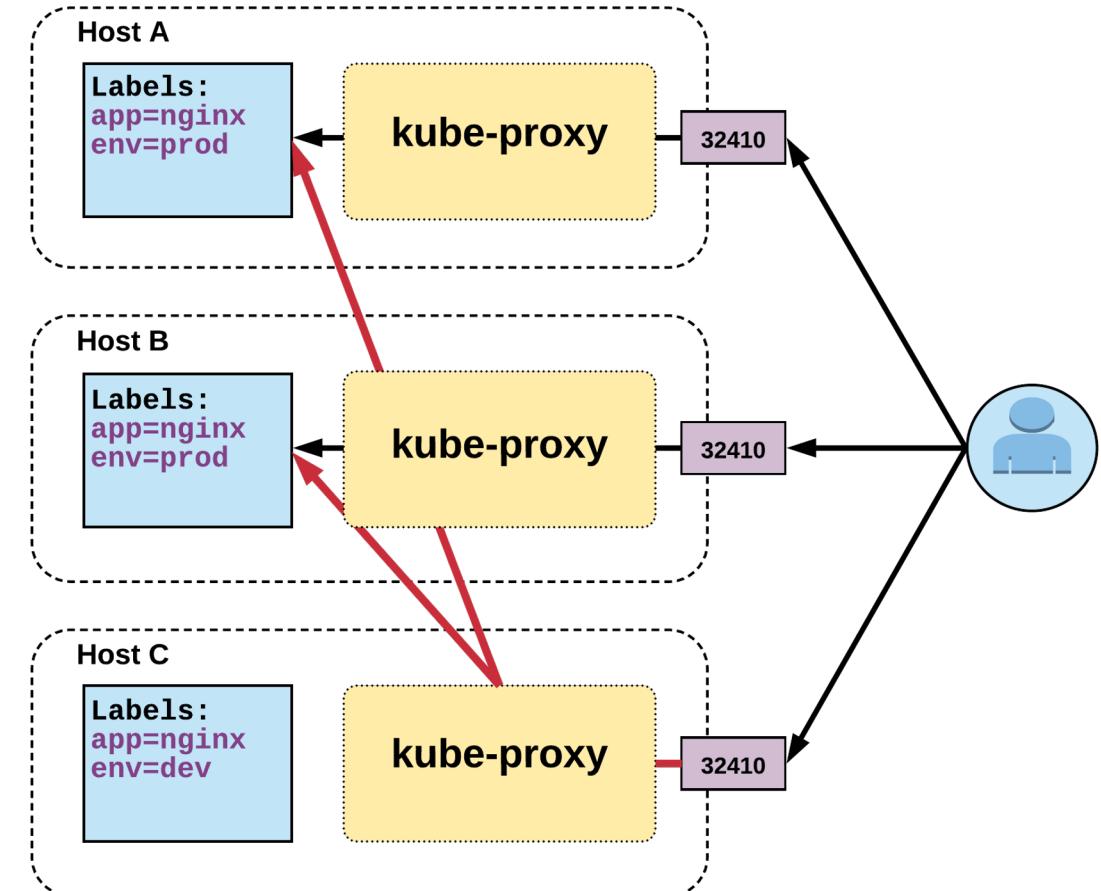
Service Types - ClusterIP

- Exposes the service on a cluster-internal IP
- Choosing this value makes the service only reachable from within the cluster
- This is the default ServiceType



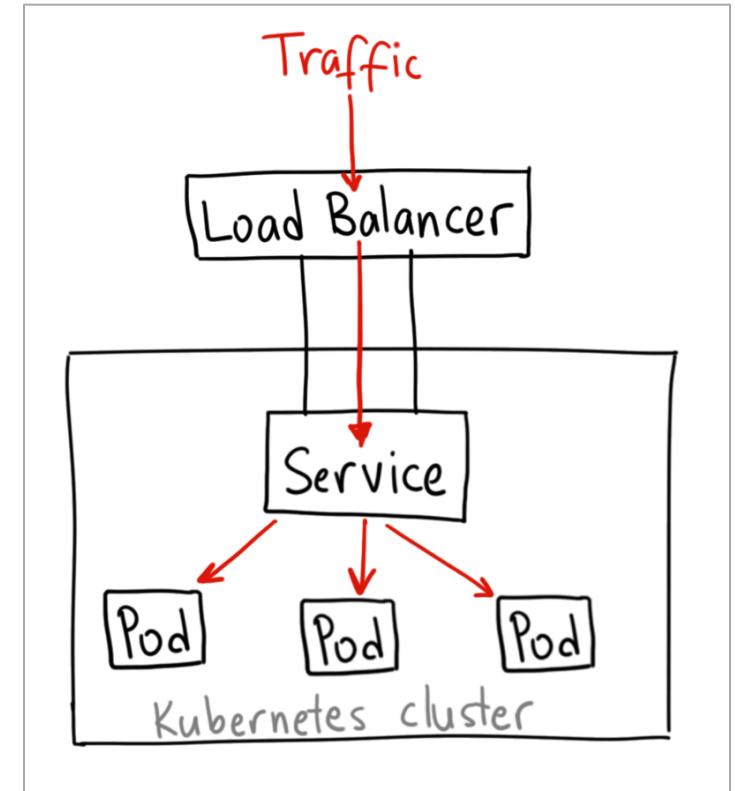
Service Types - NodePort

- Exposes the service on each Node's IP at a static port (the NodePort)
- A ClusterIP service, to which the NodePort service will route, is automatically created
- Ports being used are between 30000-32767

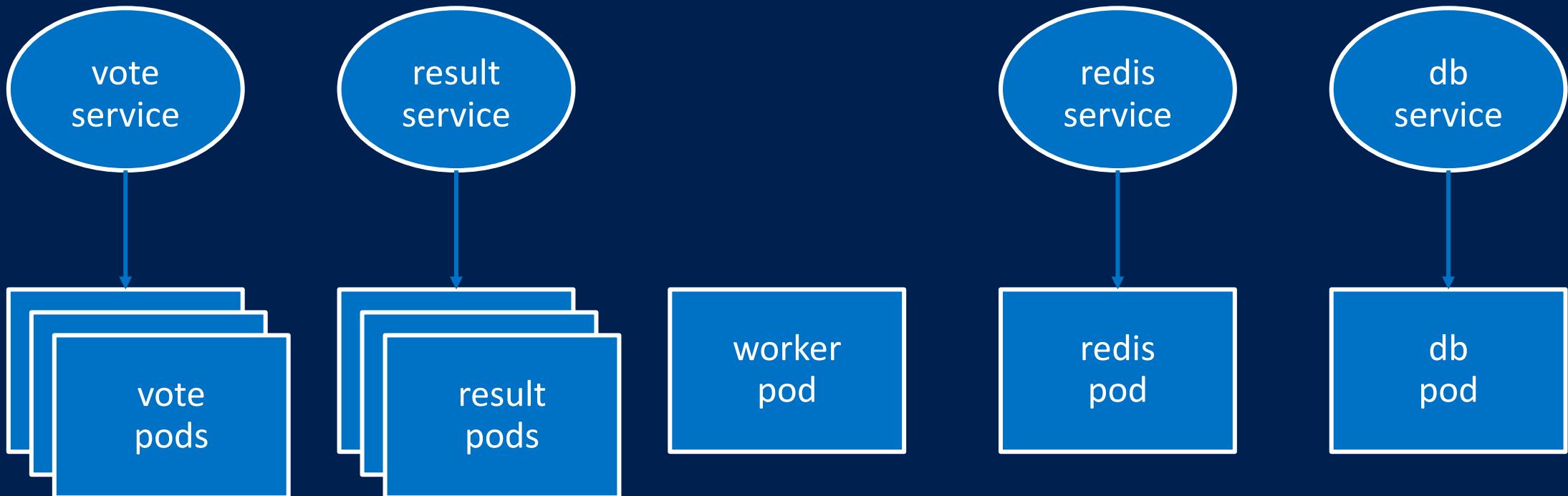


Service Types - LoadBalancer

- Exposes the service externally using a cloud provider's load balancer
- NodePort and ClusterIP services, to which the external load balancer will route, are automatically created.



Demo - Voting App



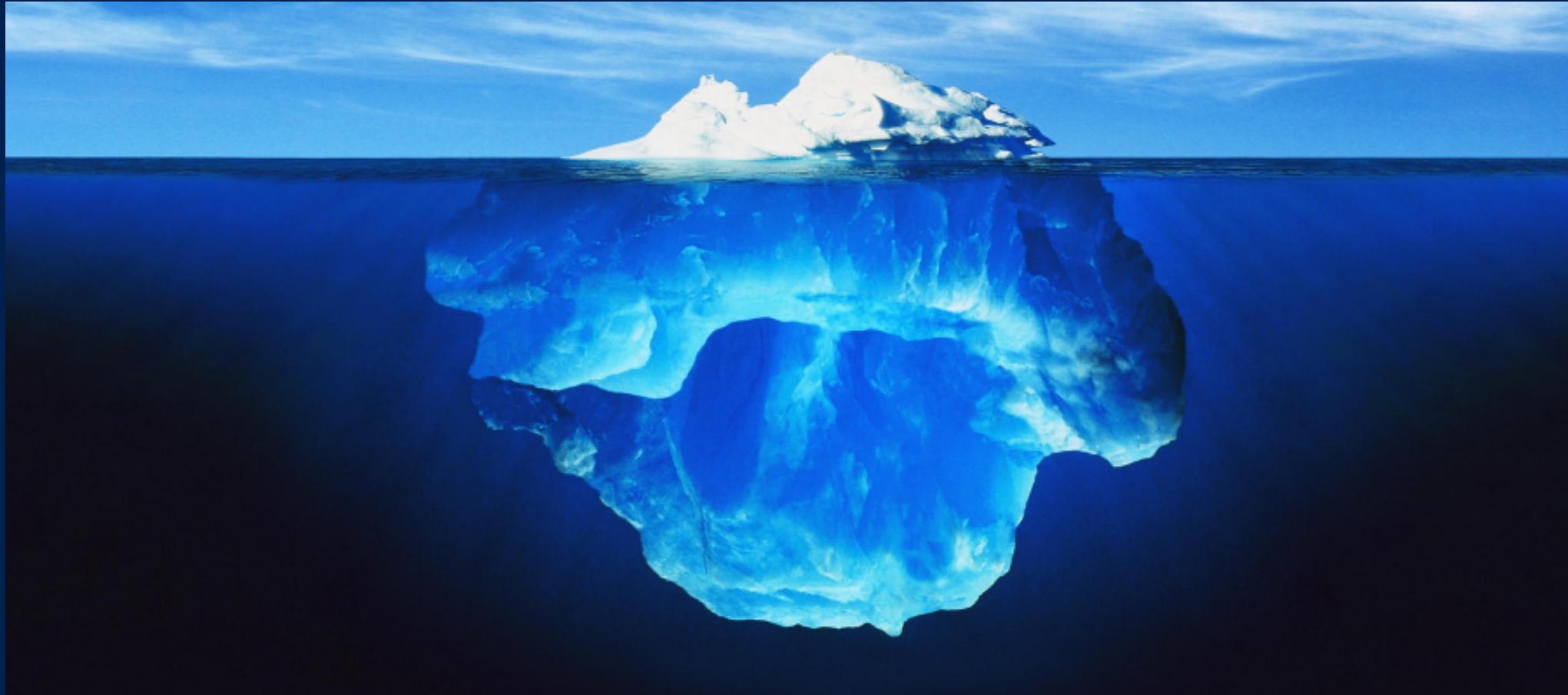
What is Helm?

- Helm is a **Package Manager** for Kubernetes
 - Package multiple K8s resources into a single logical deployment unit
 - But it's not just a Package Manager
- Helm is also a **Deployment Management** for kubernetes
 - Do a repeatable deployment
 - Manage dependencies and multiple configurations
 - Update, rollback and test application deployments

Helm Concepts

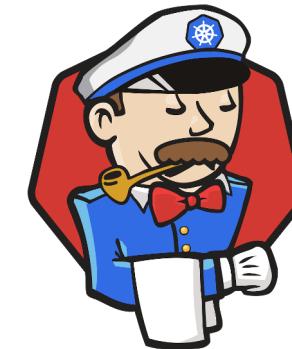
- **Chart:** a package; bundle of Kubernetes resources
- **Release:** a chart instance is loaded into Kubernetes
 - Same chart can be installed several times into the same cluster; each will have its own Release
- **Repository:** a repository of published charts
- **Template:** a K8s configuration file mixed with go/Sprig template

But this is just the tip of the iceberg...



What's Next?

- Ingress Controllers
- Daemonset
- Jobs & Cronjobs
- Statefulset & Storage
- Autoscaling
- Advanced Scheduling
- Service Mesh
- Kubernetes Operators





Thank you!

Leon Jalfon

DevOps & Cloud Solution Architect

leonj@sela.co.il

