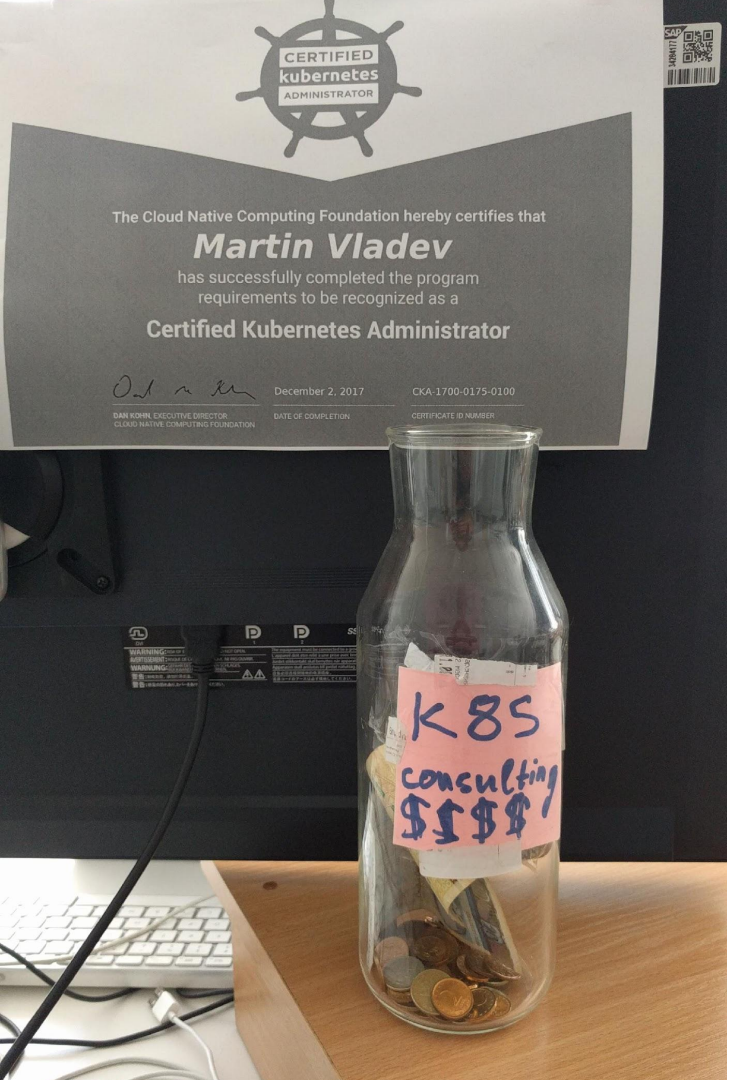
kubernetes introduction

K8S & CNCF Bulgaria

CLOUD NATIVE COMPUTING FOUNDATION

Martin Vladev SAP

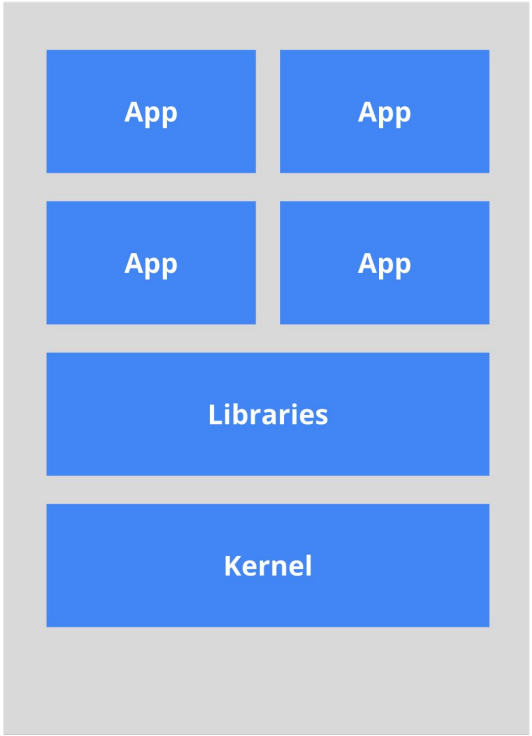


Martin Vladev

- Senior Developer @ SAP Labs Bulgaria
- 3+ years experience with K8S
- Certified Kubernetes Administrator (aka K8S consulting JAR!)
- Working on Gardener project for multi-cloud Kubernetes Cluster as a Service https://gardener.cloud
- Participating in several K8S SIGs – apimachinery, cluster-lifecycle and auth + Cluster API
- Buy me a beer and we can talk about Prometheus, Istio and other related CNCF technologies.

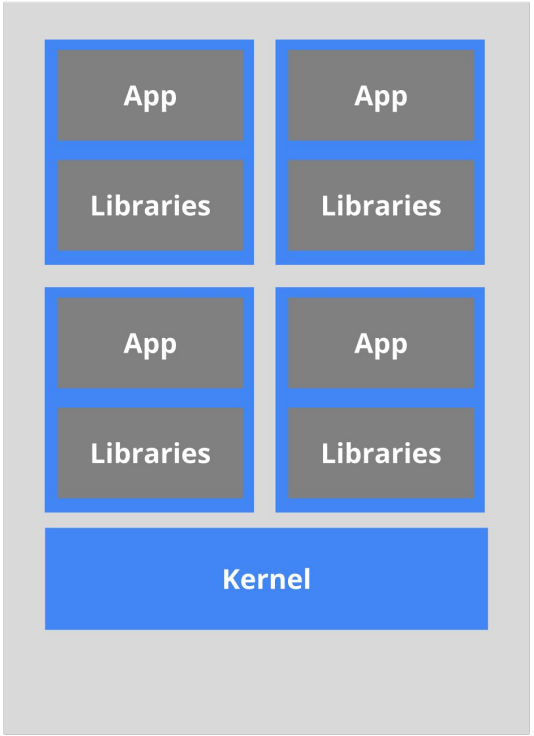
Why containers?

The old way: Applications on host



Heavyweight, non-portable
Relies on OS package manager

The new way: Deploy containers



Small and fast, portable
Uses OS-level virtualization

What do we need?

- **Scheduling**: Where should my containers run?
- **Lifecycle and health**: How to keep my containers running despite failures?
- **Discovery**: Where are my containers now?
- **Auth{n,z}**: Who can do things to my containers?
- **Aggregates**: How to compose sets of containers into workloads?
- **Scaling**: When to make workloads bigger or smaller?

kubernetes is an open-source system for automating deployment, scaling and management of containerized applications.

First Graduated CNCF Project.

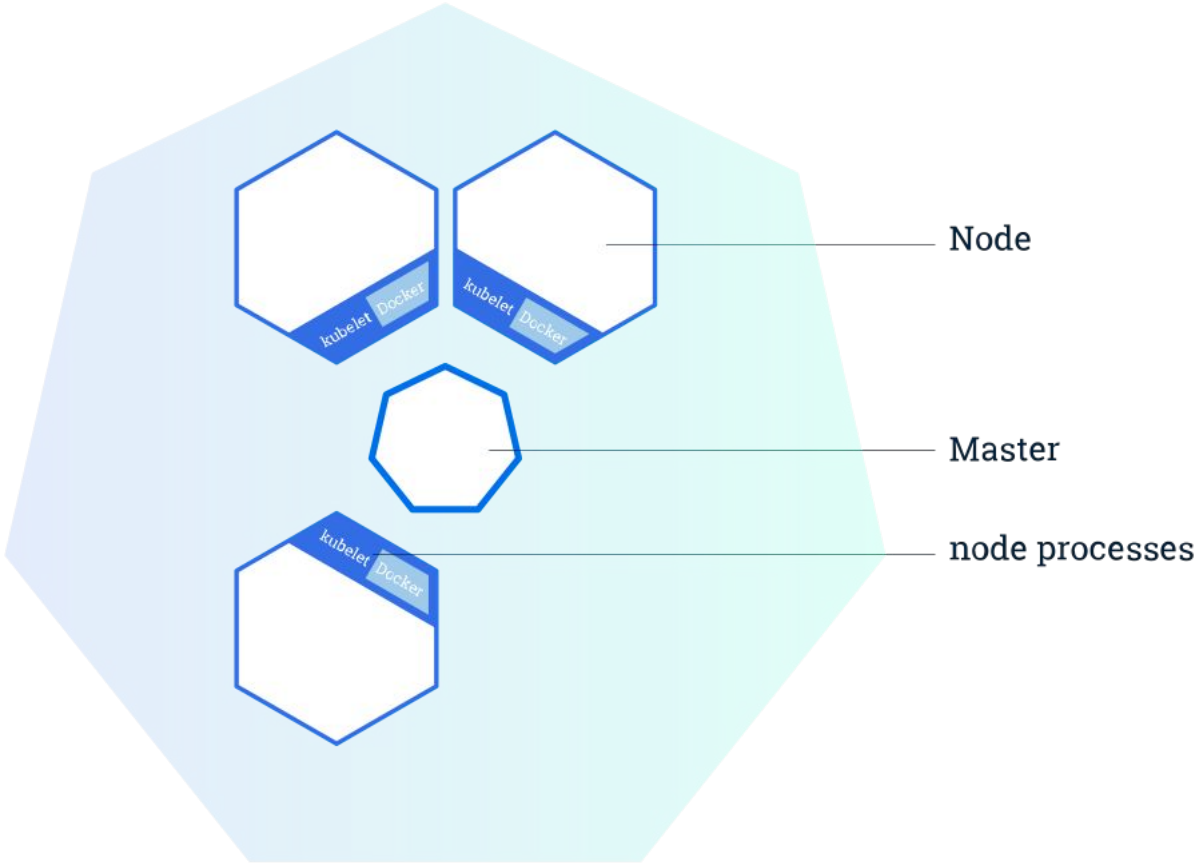
Core features

- Horizontal scaling
- Automated rollouts and rollbacks
- Storage orchestration
- Self-healing
- Service discovery and load balancing
- Secret and configuration management
- Job execution
- Log access and ingestion
- Extendable

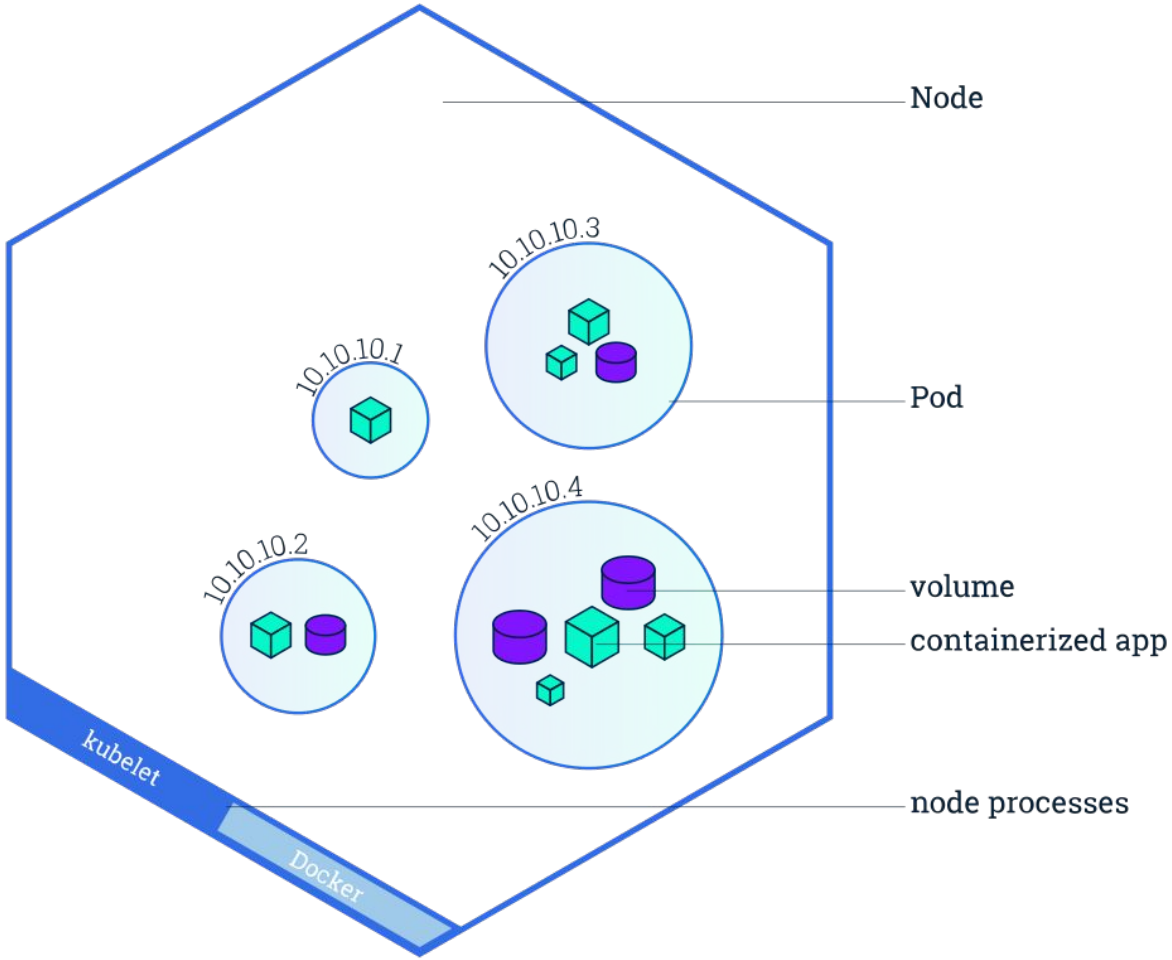
and more...

Concepts

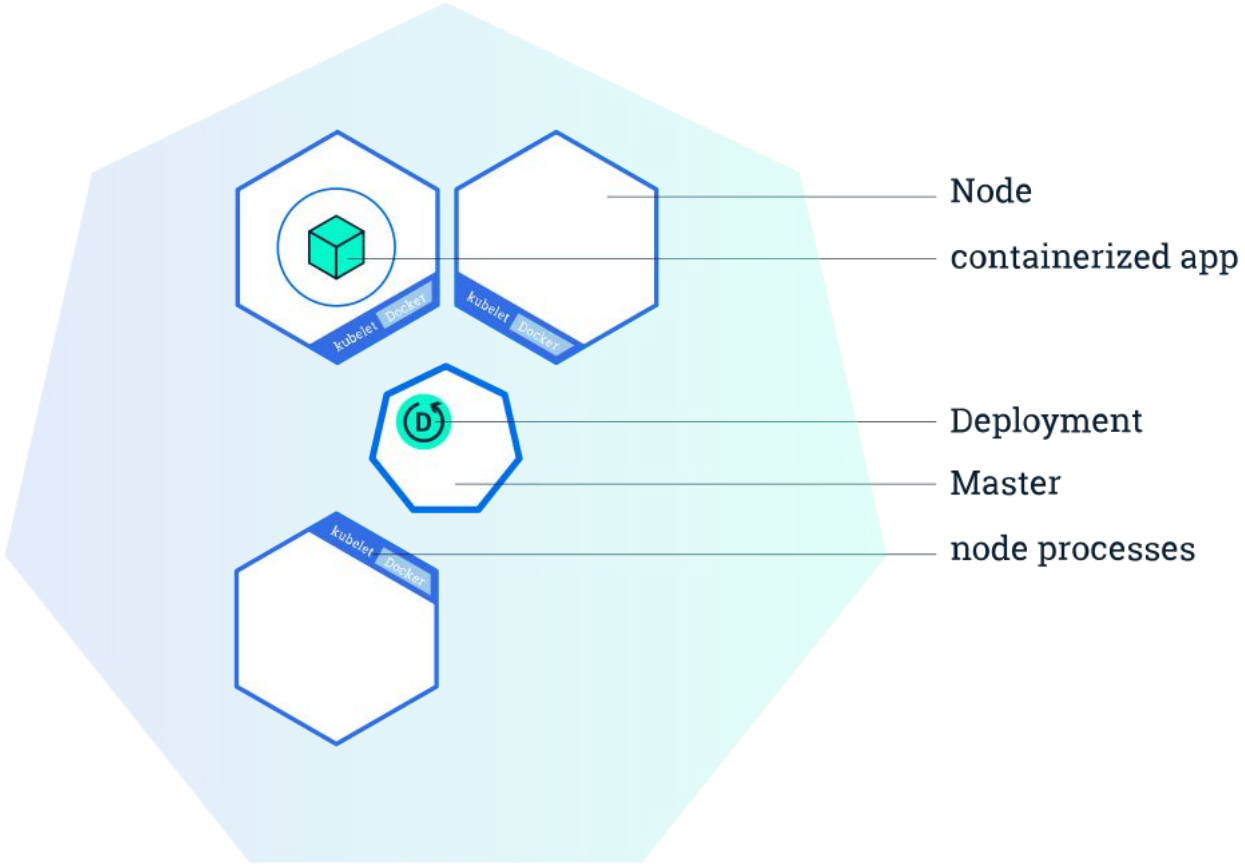
- **Container**: A sealed application package
- **Pod**: A small group of tightly coupled Containers
- **Labels**: Identifying metadata attached to objects
- **Namespace**: Separate objects virtually
- **Selector**: A query against labels, producing a set result
- **Controller**: A reconciliation loop that drives current state towards desired state
- **Service**: A set of pods that work together



Kuberneters cluster

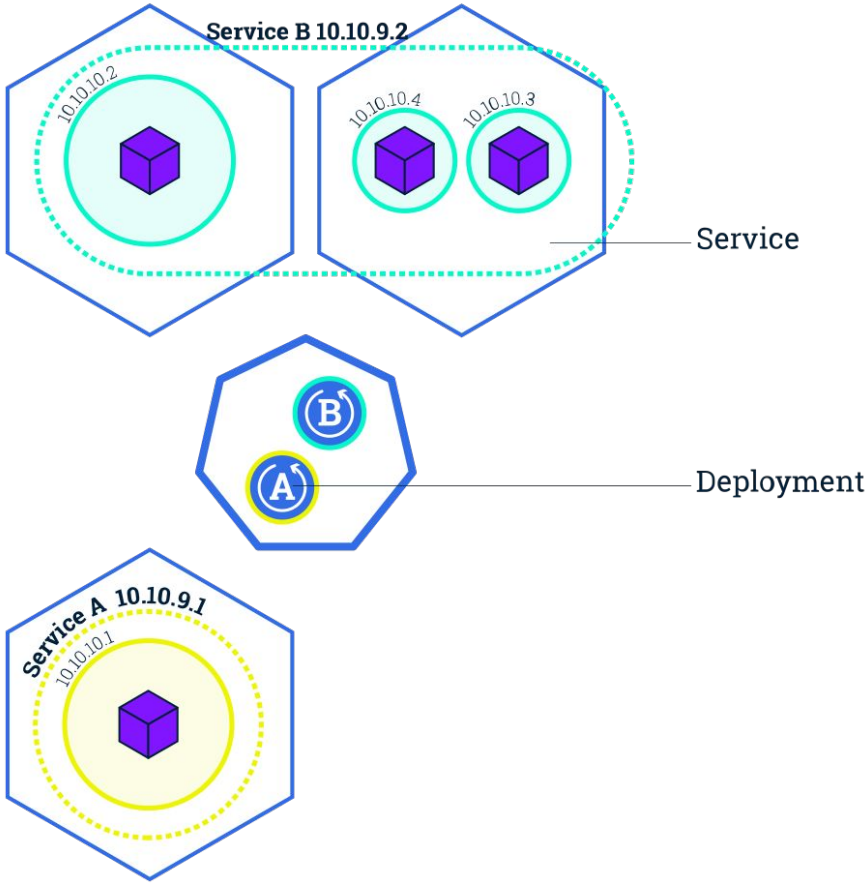

Node
10.10.10.3
10.10.10.1
Pod
10.10.10.4
10.10.10.2
volume
containerized app
kubelet
Docker
node processes

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.7.9
    ports:
    - containerPort: 80
    volumeMounts:
    - name: html-storage
      mountPath: /data/html
  volumes:
  - name: html-storage
    emptyDir: {}
```

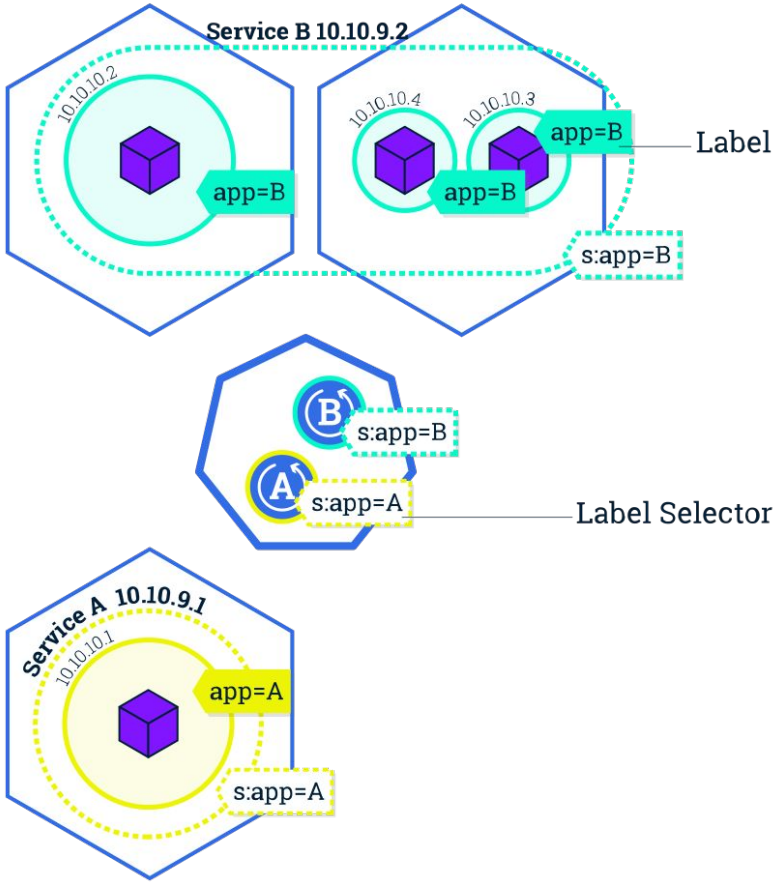


Kuberneters Cluster

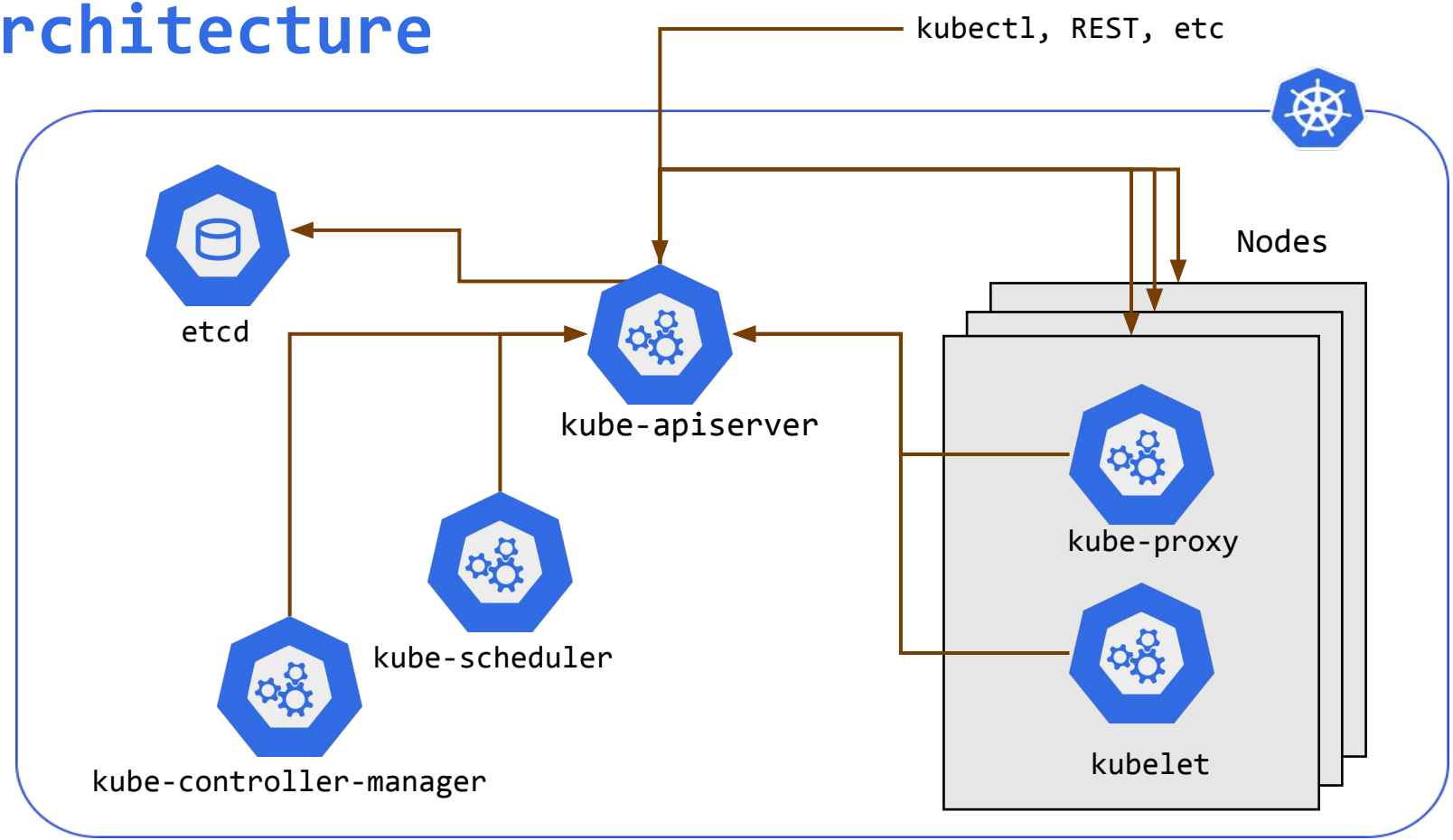
Services


Service B 10.10.9.2
10.10.10.2
10.10.10.4
10.10.10.3
Service
B
A
Deployment
Service A 10.10.9.1
10.10.10.1

Labels


Service B 10.10.9.2
10.10.10.2
app=B
10.10.10.4
10.10.10.3
app=B
app=B
Label
s:app=B
B
s:app=B
A
s:app=A
Label Selector
Service A 10.10.9.1
10.10.10.1
app=A
s:app=A

Architecture



Control loops

- Drive **current state => desired state**
- Act **independently**
- APIs - **no shortcuts** or back doors
- Observed **state** is truth
- Recurring pattern in the system

Example: **ReplicaSet**

DEMO

Q&A