

# KUBERNETES

**BJÖRN BESKOW**

2016-01-27 | [CALLISTAENTERPRISE.SE](http://CALLISTAENTERPRISE.SE)

## BACKGROUND

Many small, moving parts

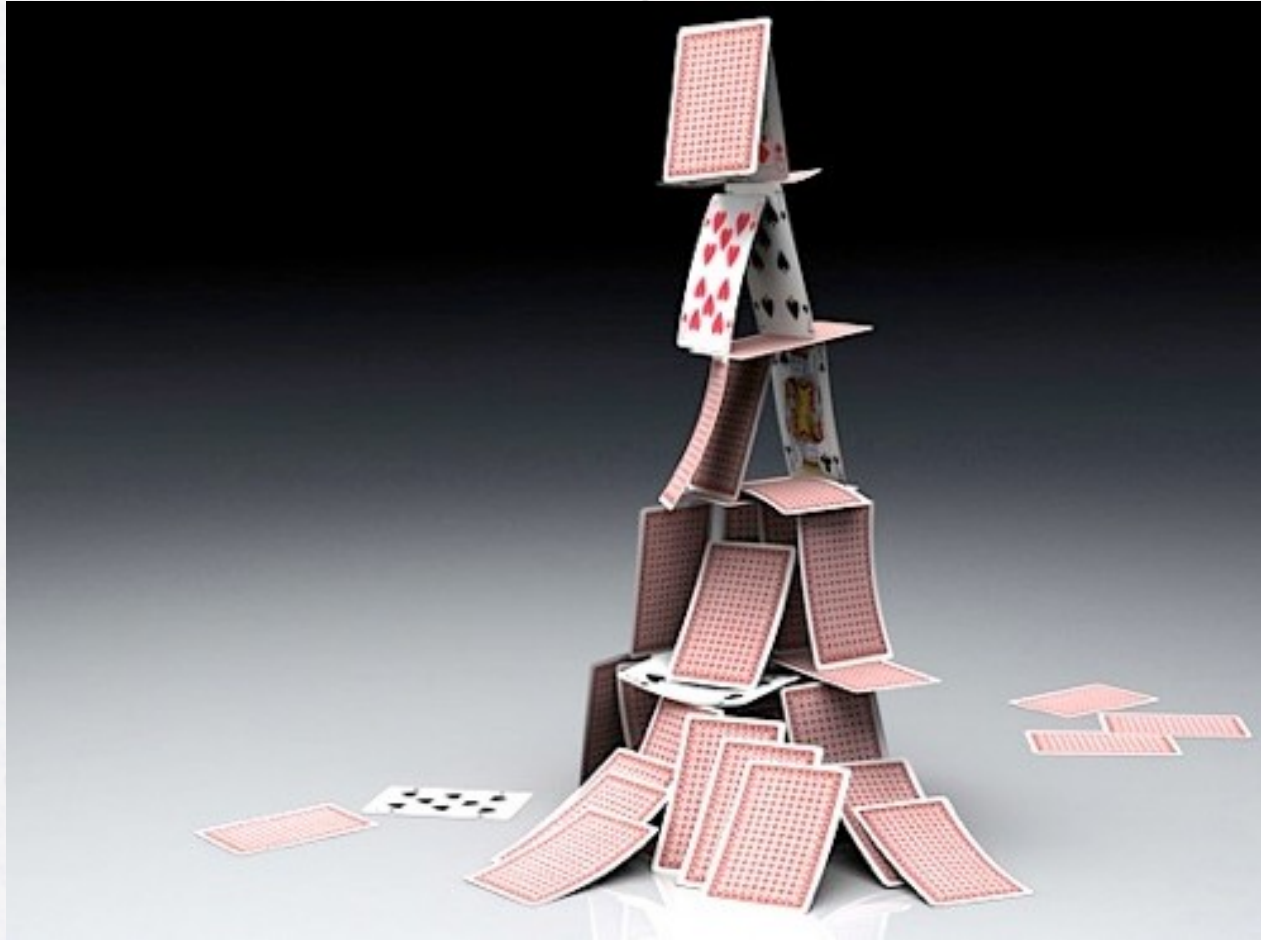


**BACKGROUND**

# Software, Hardware & Networks

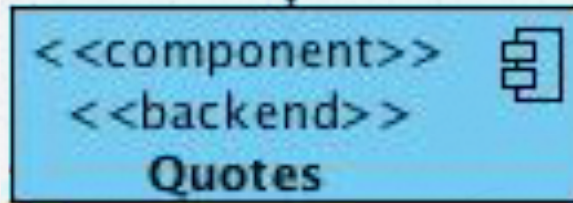
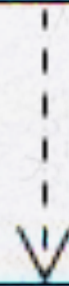
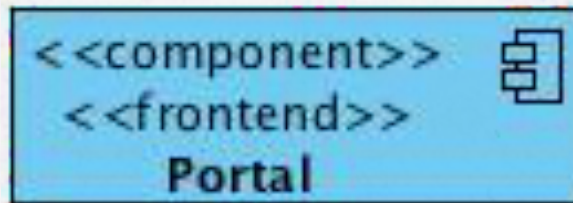
## BACKGROUND

# Failures are inevitable

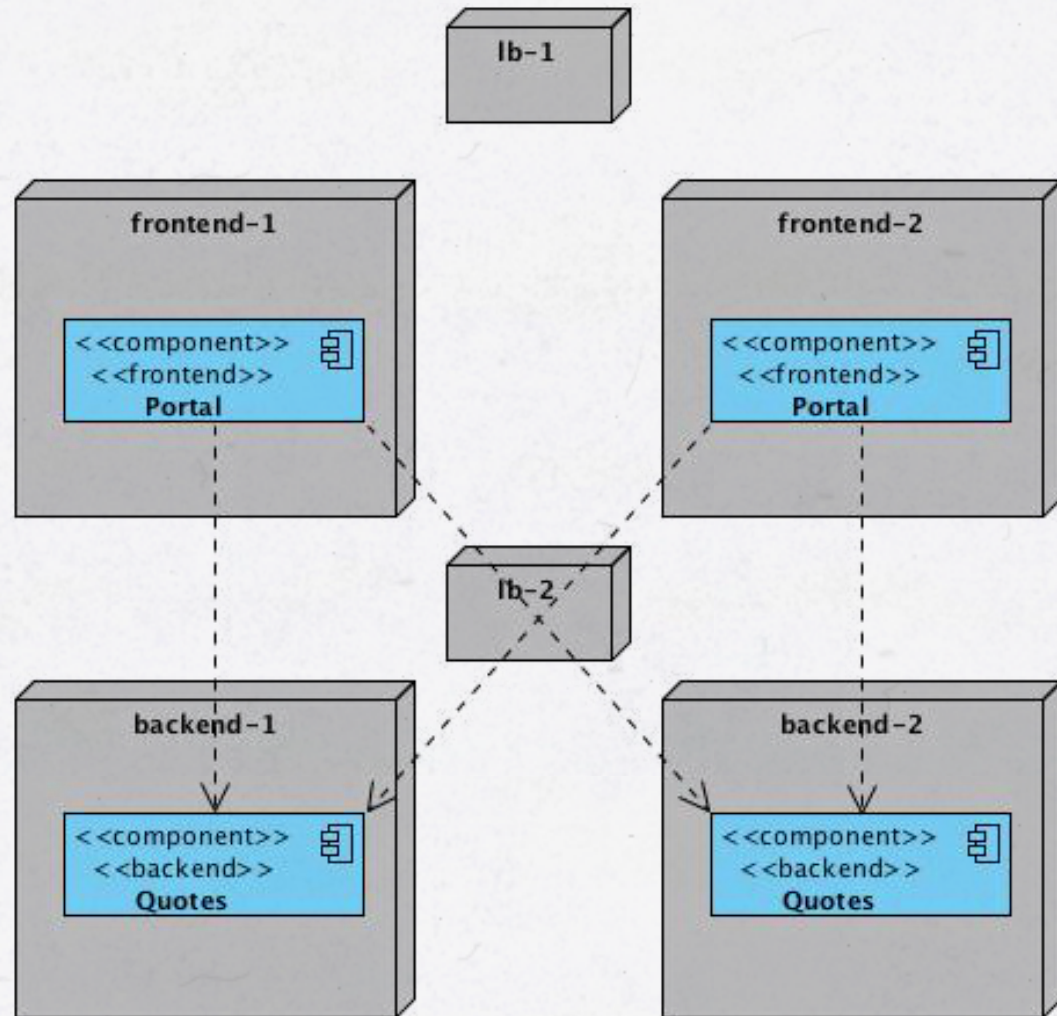


**BACKGROUND**

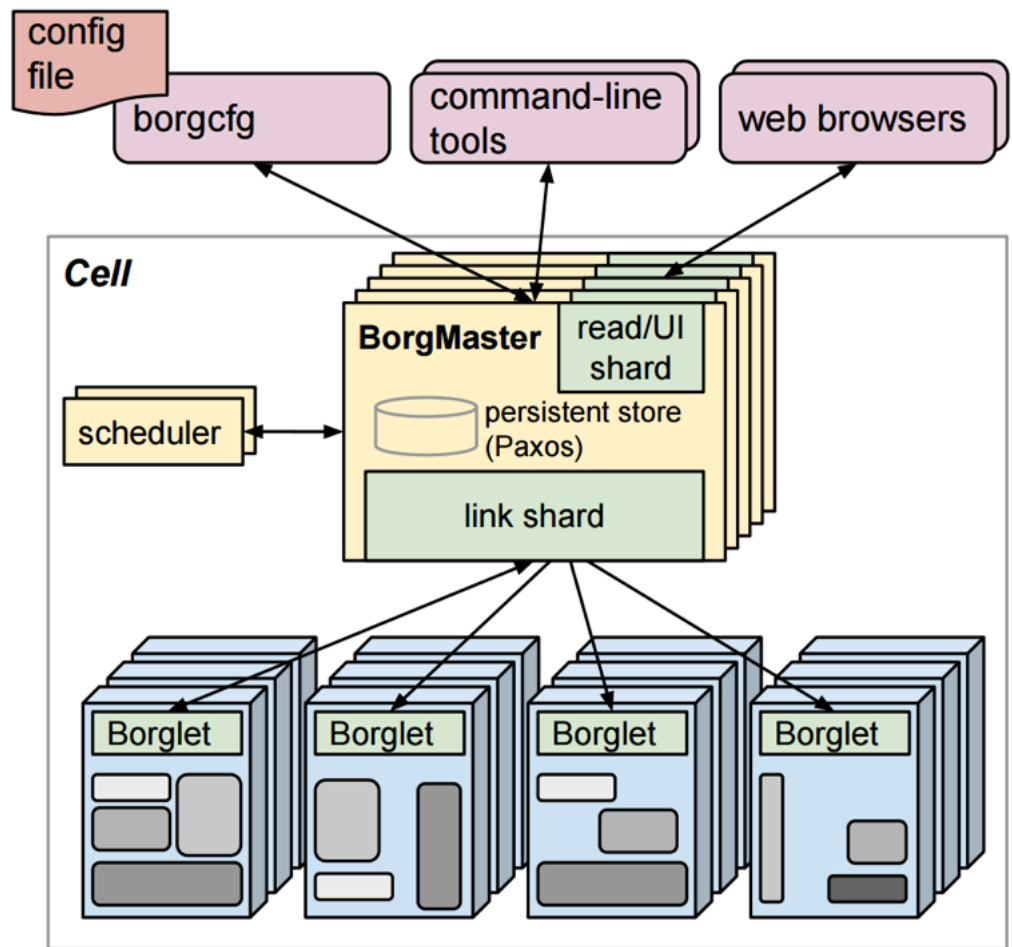
# Ephemeral











## GOALS

Treat all hardware nodes  
as one giant logical  
machine

## GOALS

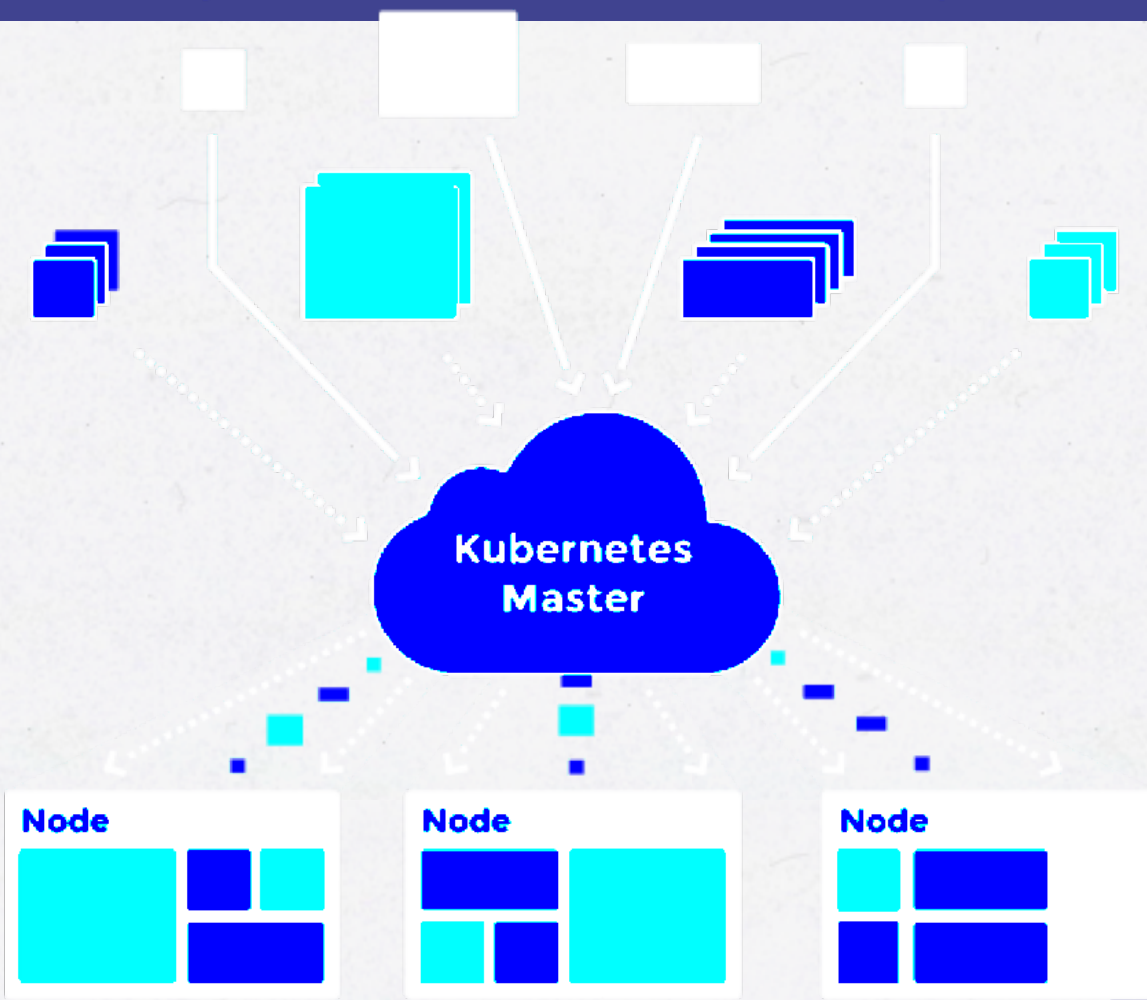
Focus on your  
"applications"

## GOALS

Manage your applications  
through "Wishful Thinking":  
Declare how it *ought to be*



kubernetes by Google



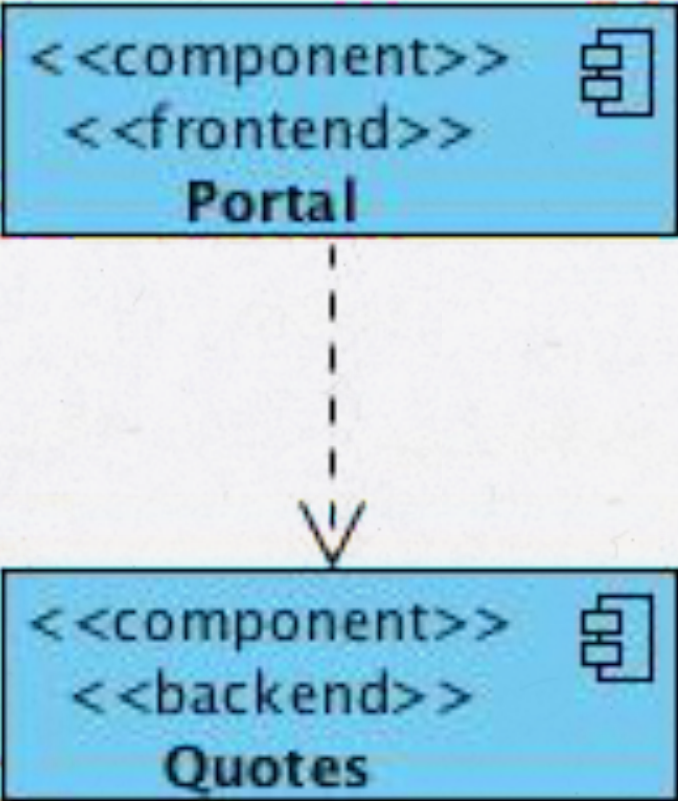
Source: <http://kubernetes.io/>



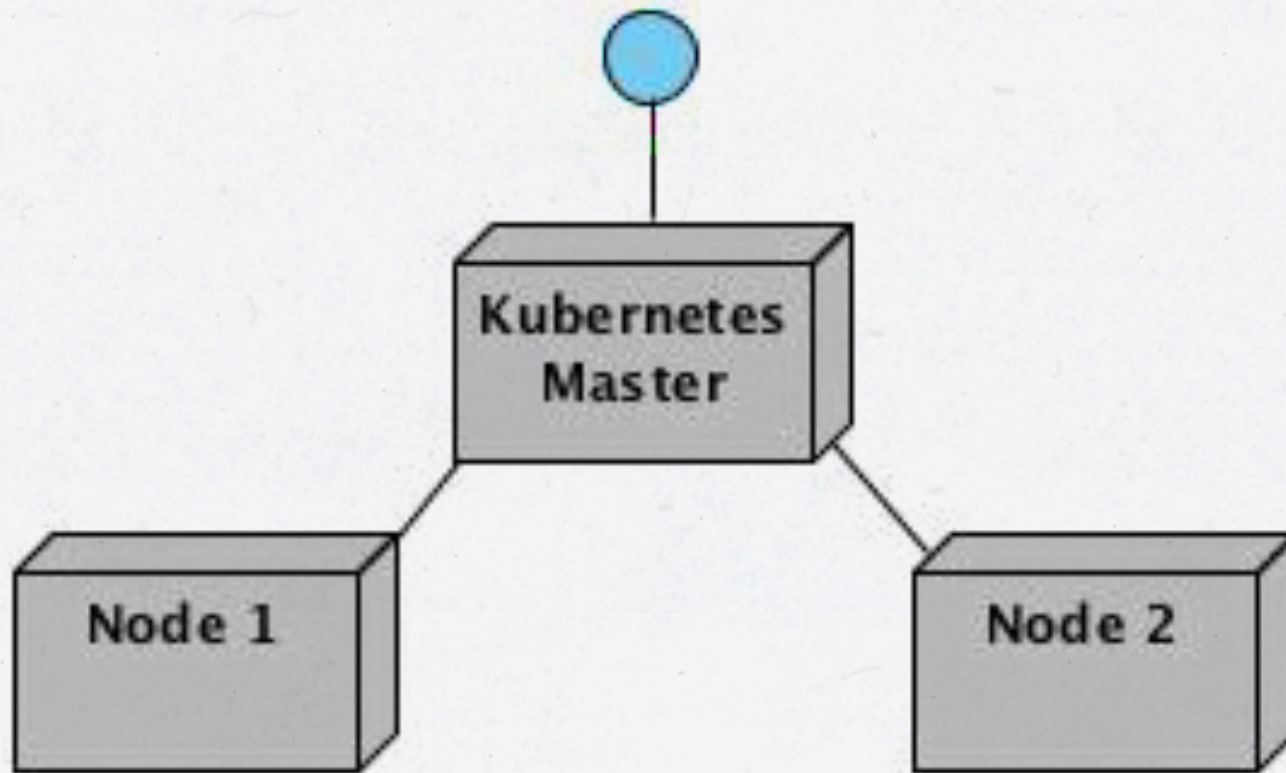
## DEMO SETUP

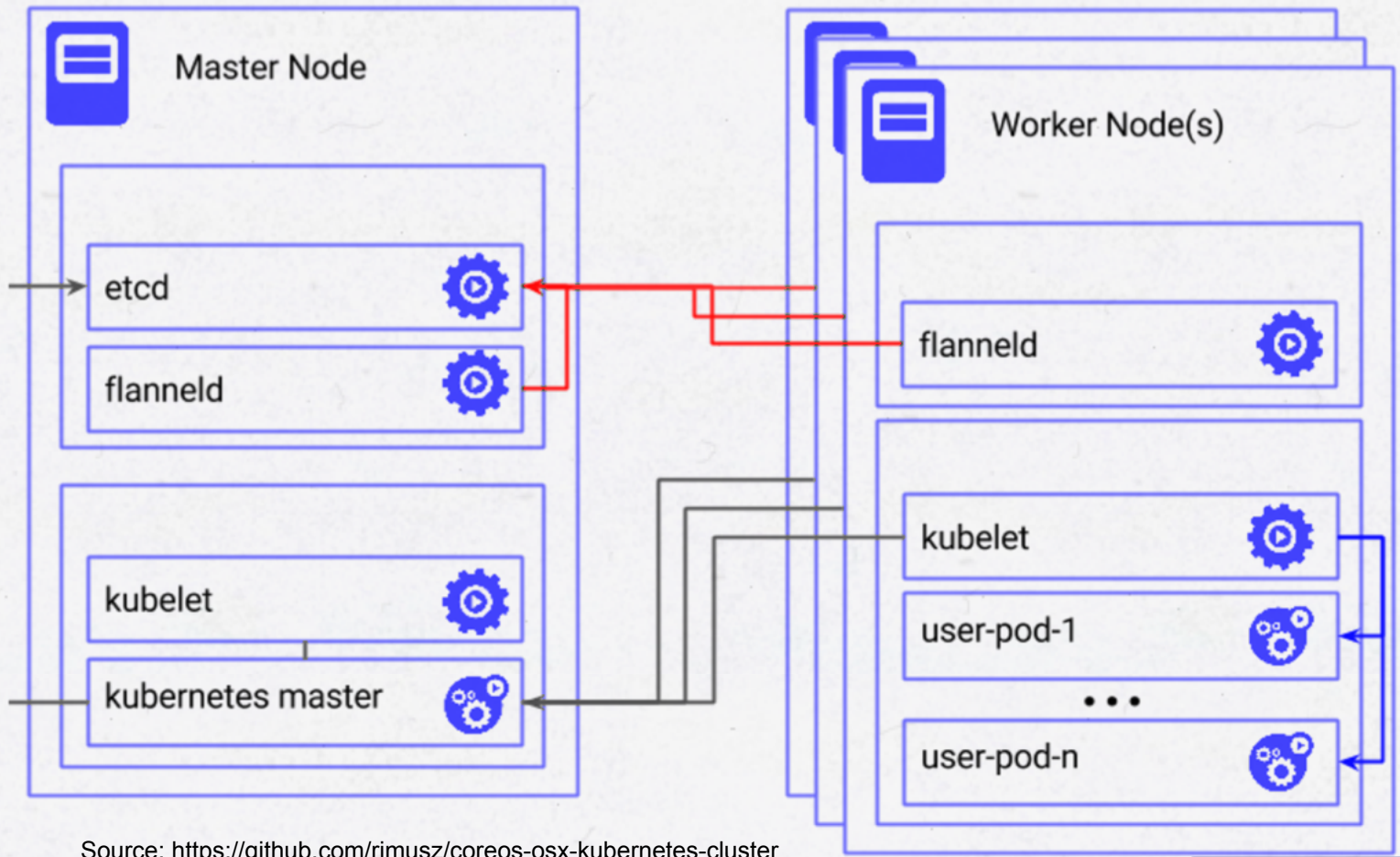


# DEMO SETUP



## DEMO SETUP





Source: <https://github.com/rimusz/coreos-osx-kubernetes-cluster>

**DEMO**

## CORE CONCEPTS

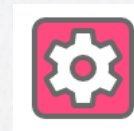
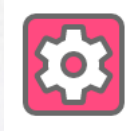
# Kubectl CLI



# CORE CONCEPTS



- Node



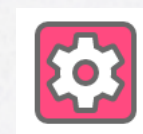
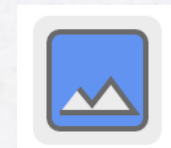
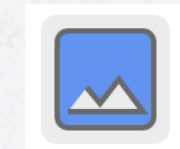
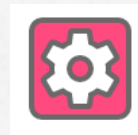
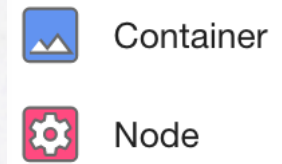


**DEMO**



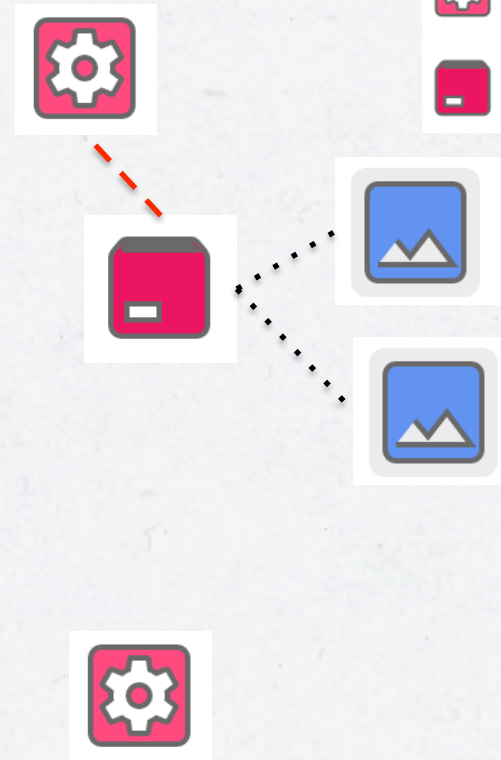
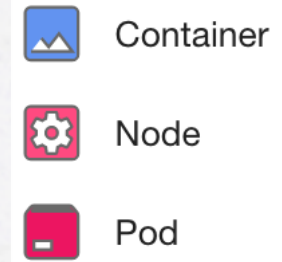
# CORE CONCEPTS

- Node
- Container



# CORE CONCEPTS

- Node
- Container
- Pod



## QUOTES-POD.YAML

```
apiVersion: v1
kind: Pod
metadata:
  name: quotes
spec:
  containers:
  - name: quotes
    image: docker:5000/quotes:1
    ports:
    - containerPort: 9090
      hostPort: 9090
```



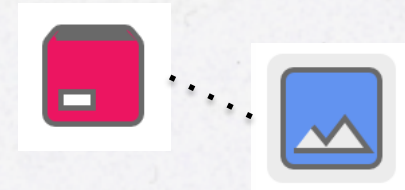
Container



Node



Pod

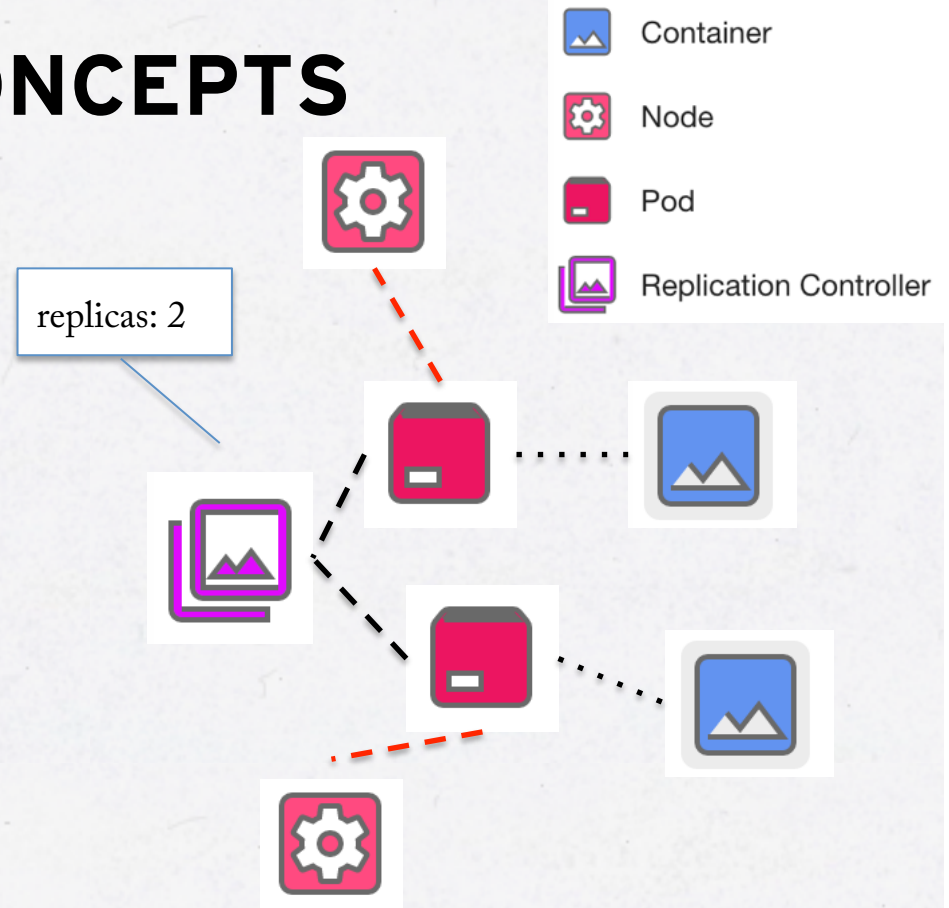




**DEMO**

# CORE CONCEPTS

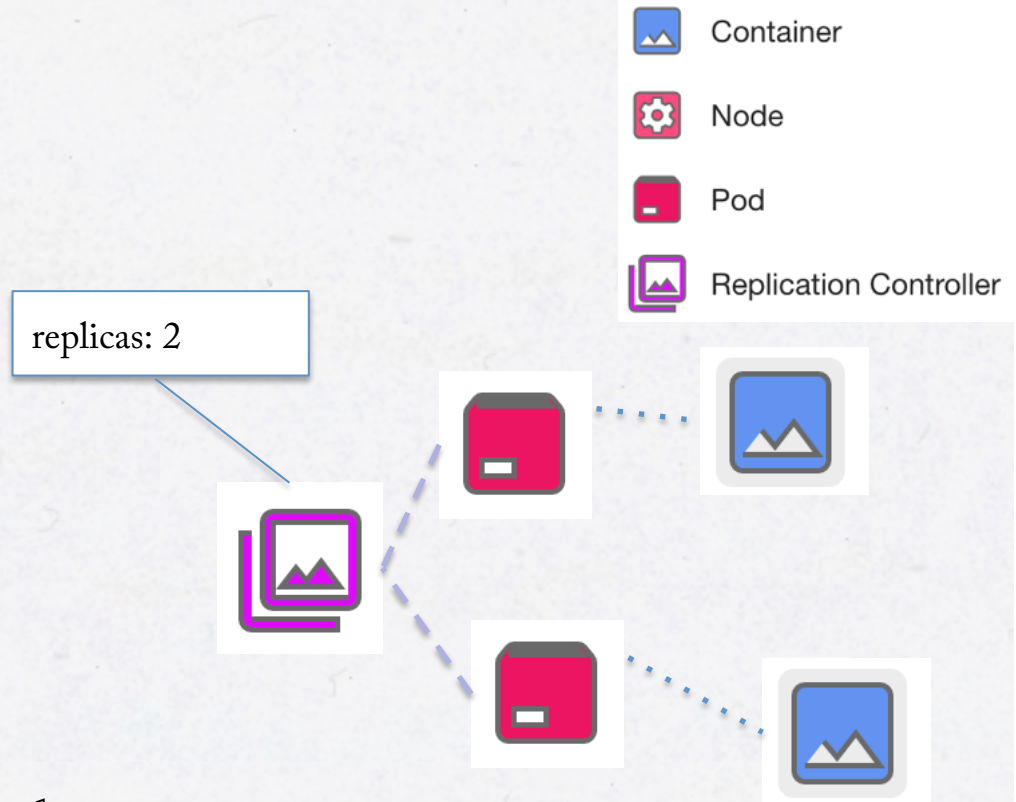
- Node
- Container
- Pod
- Replication Controller



Upcoming feature: auto-scaling based on load: see <https://github.com/kubernetes/kubernetes/blob/master/docs/design/horizontal-pod-autoscaler.md>

## QUOTES-CONTROLLER.YAML

```
apiVersion: v1
kind: ReplicationController
metadata:
  name: quotes
spec:
  replicas: 2
  ...
  template:
    spec:
      containers:
      - name: quotes
        image: docker:5000/quotes:1
```

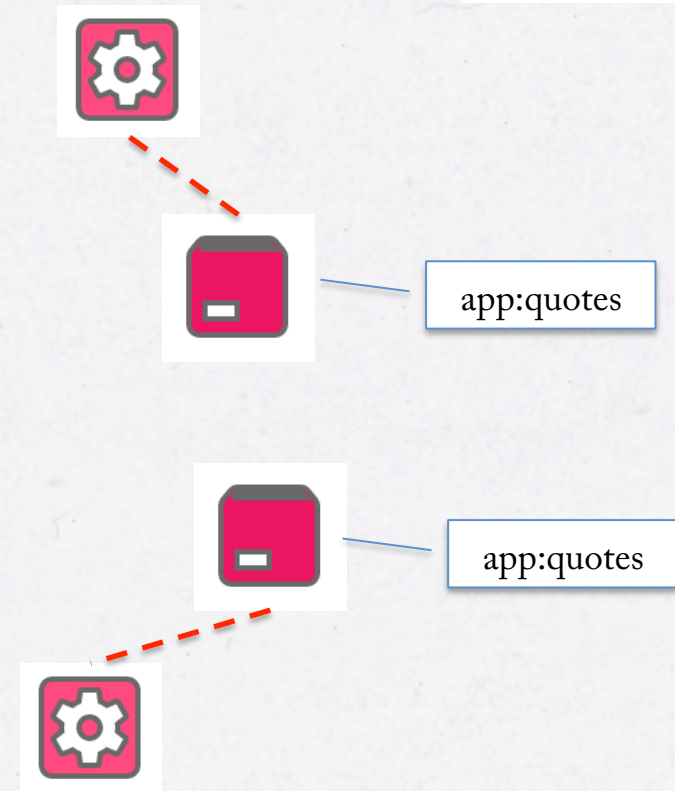
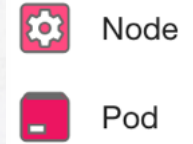




**DEMO**

# CORE CONCEPTS

- Node
- Container
- Pod
- Replication Controller
- Labels





# CORE CONCEPTS

- Node
- Container
- Pod
- Replication Controller
- Labels



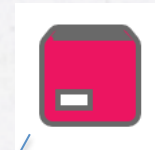
app:portal  
version: 1  
tier: frontend



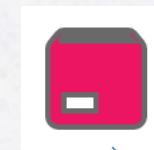
app:quotes  
version: 1  
tier: backend



app:portal  
version: 2  
tier: frontend

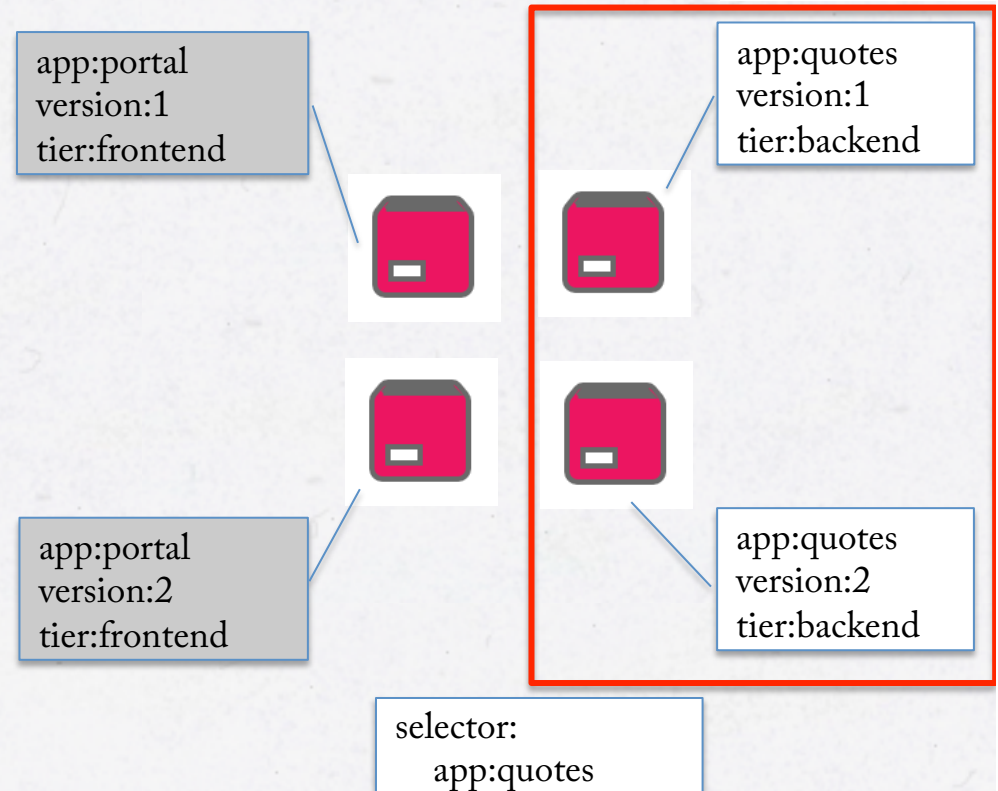
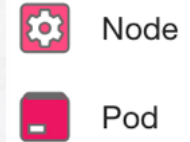


app:quotes  
version: 2  
tier: backend



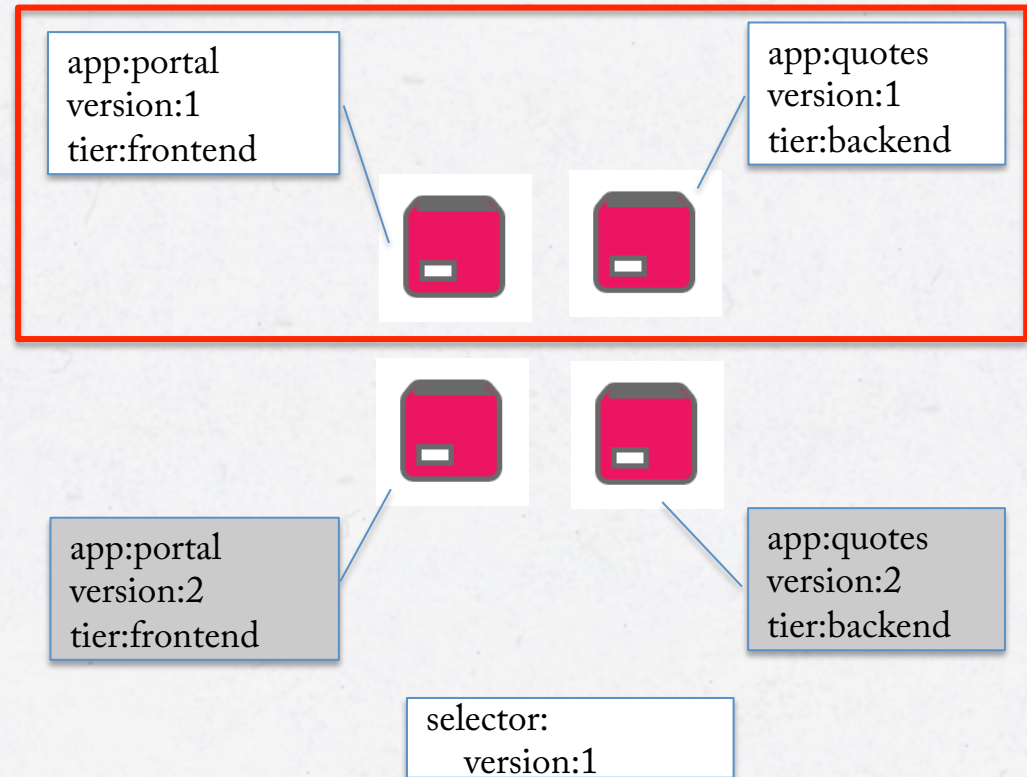
# CORE CONCEPTS

- Node
- Container
- Pod
- Replication Controller
- Labels



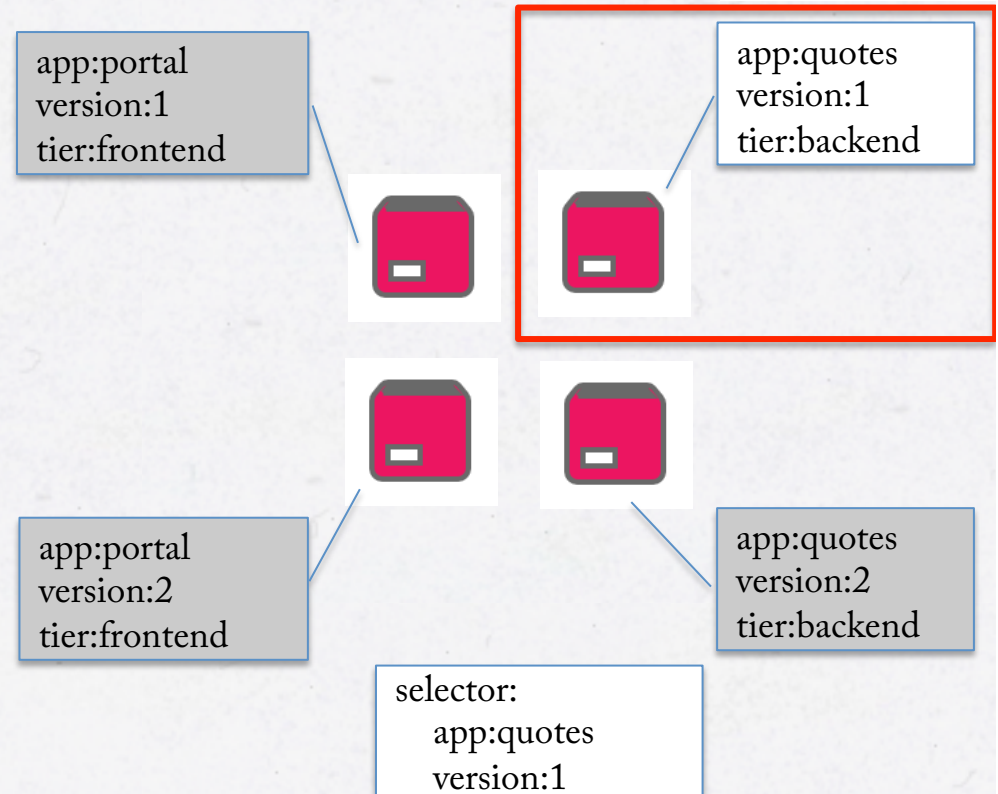
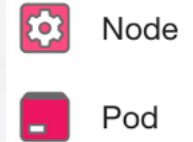
# CORE CONCEPTS

- Node
- Container
- Pod
- Replication Controller
- Labels



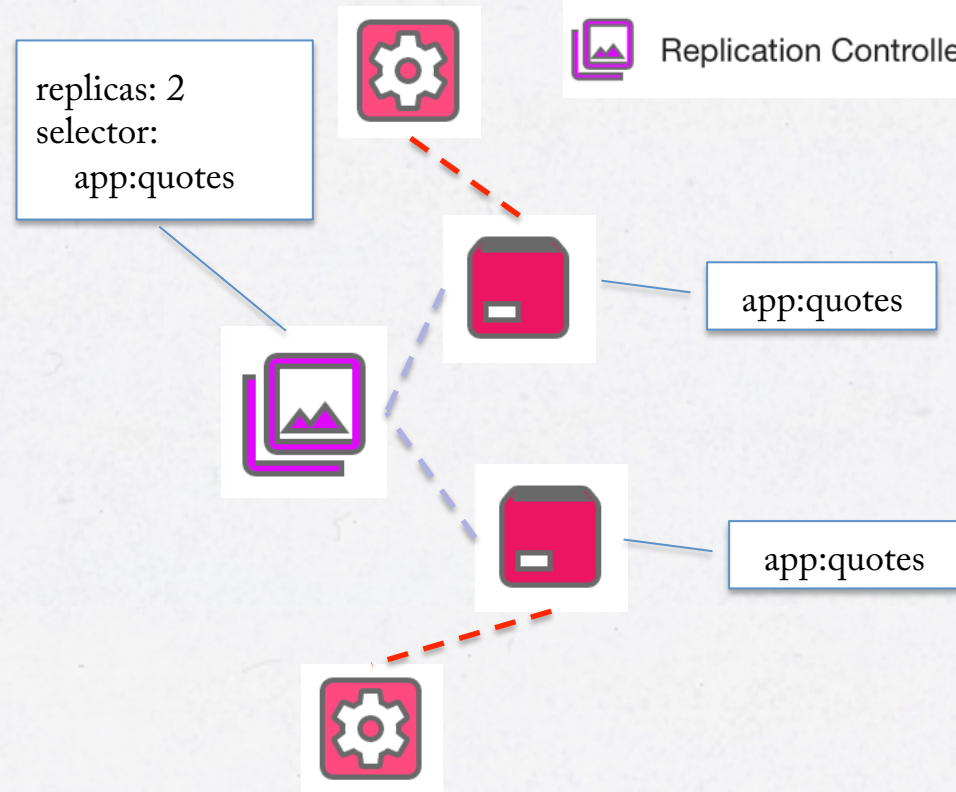
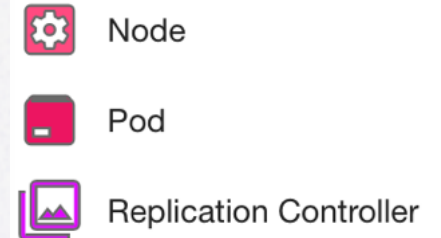
# CORE CONCEPTS

- Node
- Container
- Pod
- Replication Controller
- Labels



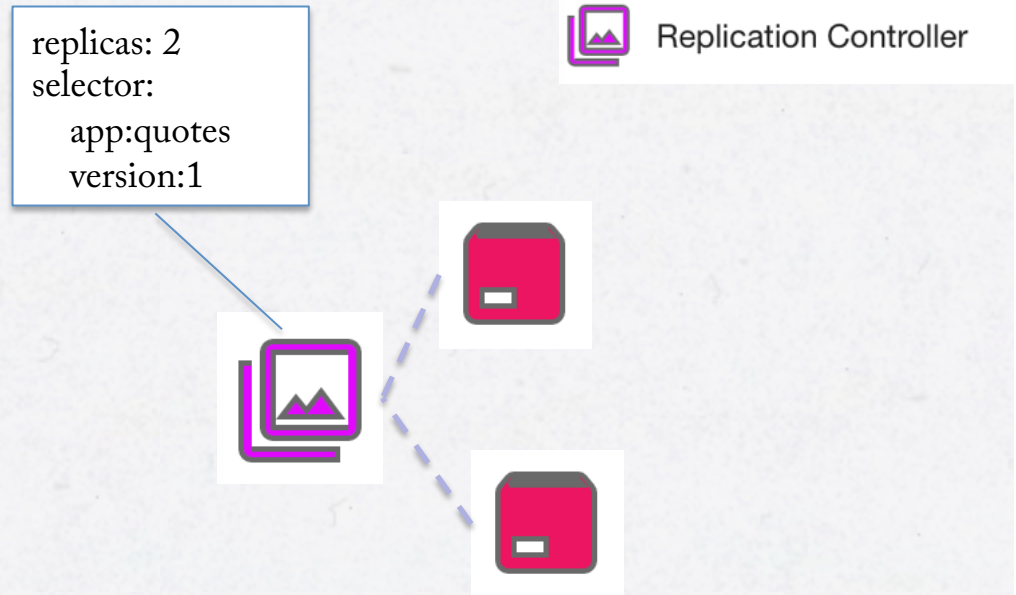
# CORE CONCEPTS

- Node
- Container
- Pod
- Replication Controller
- Labels



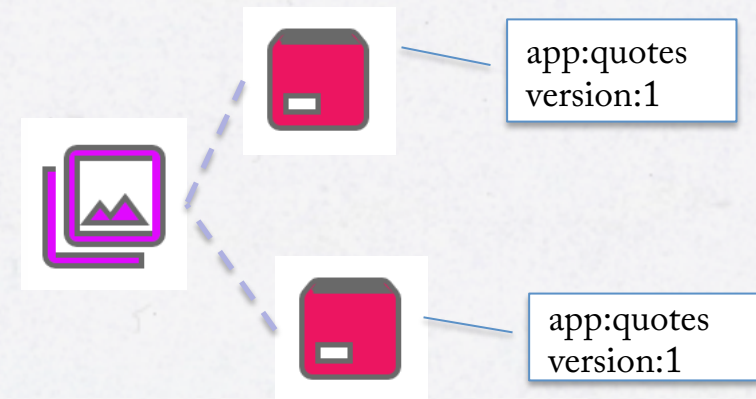
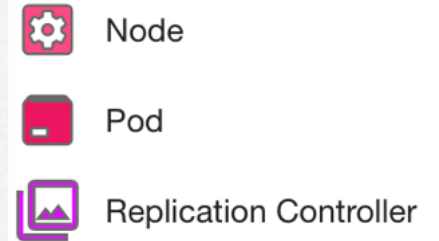
## QUOTES-CONTROLLER.YAML

```
apiVersion: v1
kind: ReplicationController
metadata:
  name: quotes
  labels:
    app: quotes
spec:
  replicas: 2
  selector:
    app: quotes
    version: 1
  template:
    ...
```



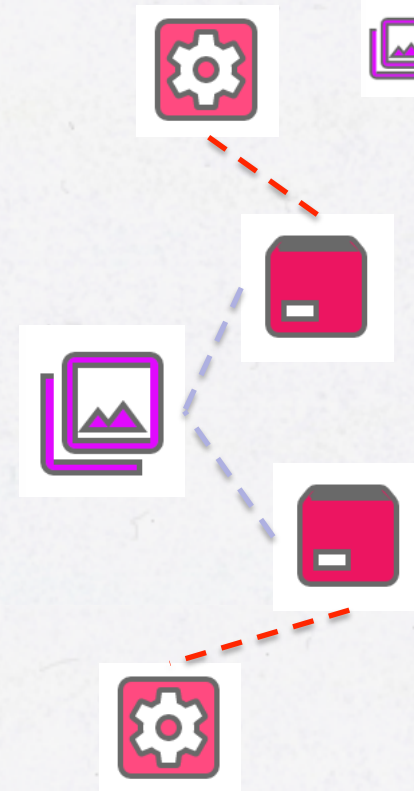
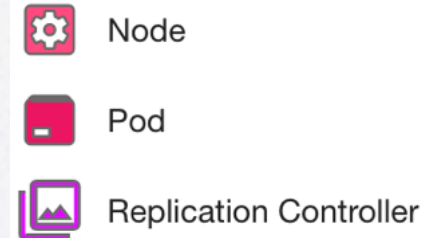
## QUOTES-CONTROLLER.YAML

```
...  
  template:  
    metadata:  
      labels:  
        app: quotes  
        version: 1  
    spec:  
      containers:  
      - name: quotes  
        image: docker:5000/quotes:1  
        ports:  
        - containerPort: 9090
```



# CORE CONCEPTS

- Node
- Container
- Pod
- Replication Controller
- Label
- Liveness





## QUOTES-CONTROLLER.YAML

...

livenessProbe:

httpGet:

path: **/health**

port: **9090**

initialDelaySeconds: 10

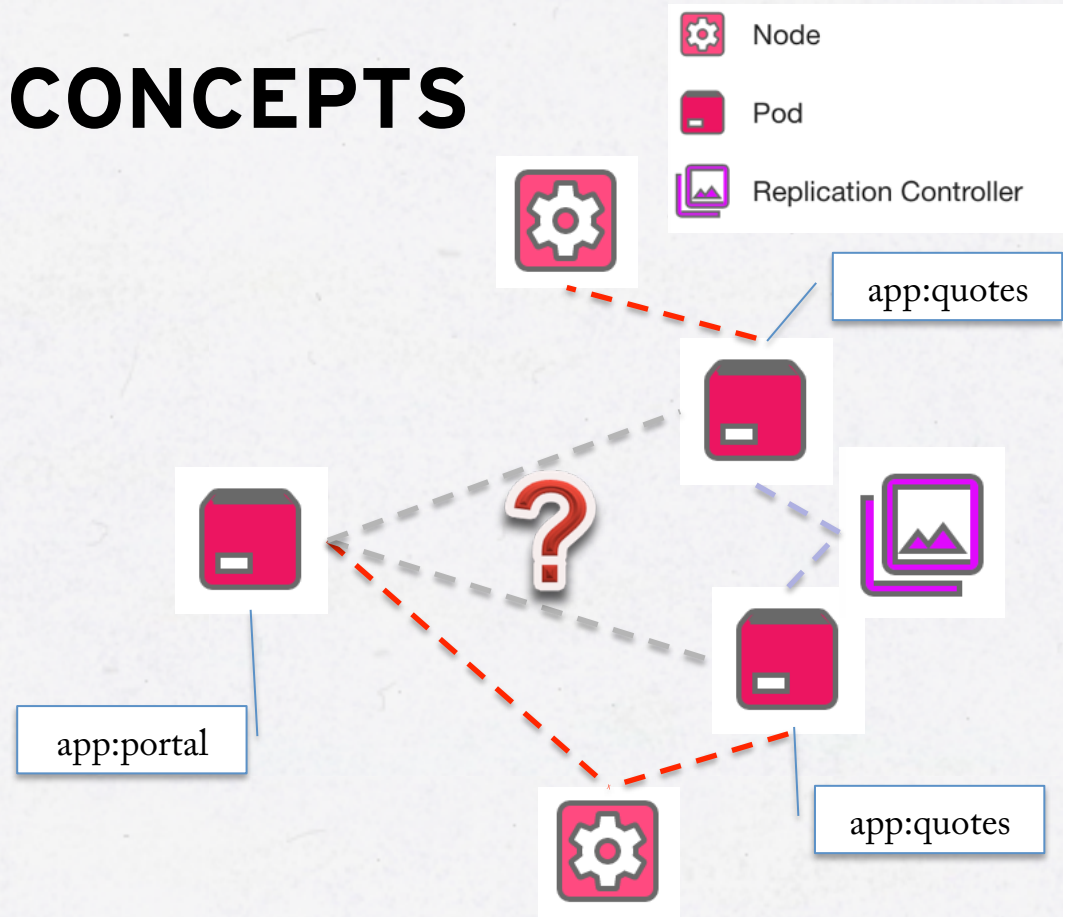
timeoutSeconds: 1



**DEMO**

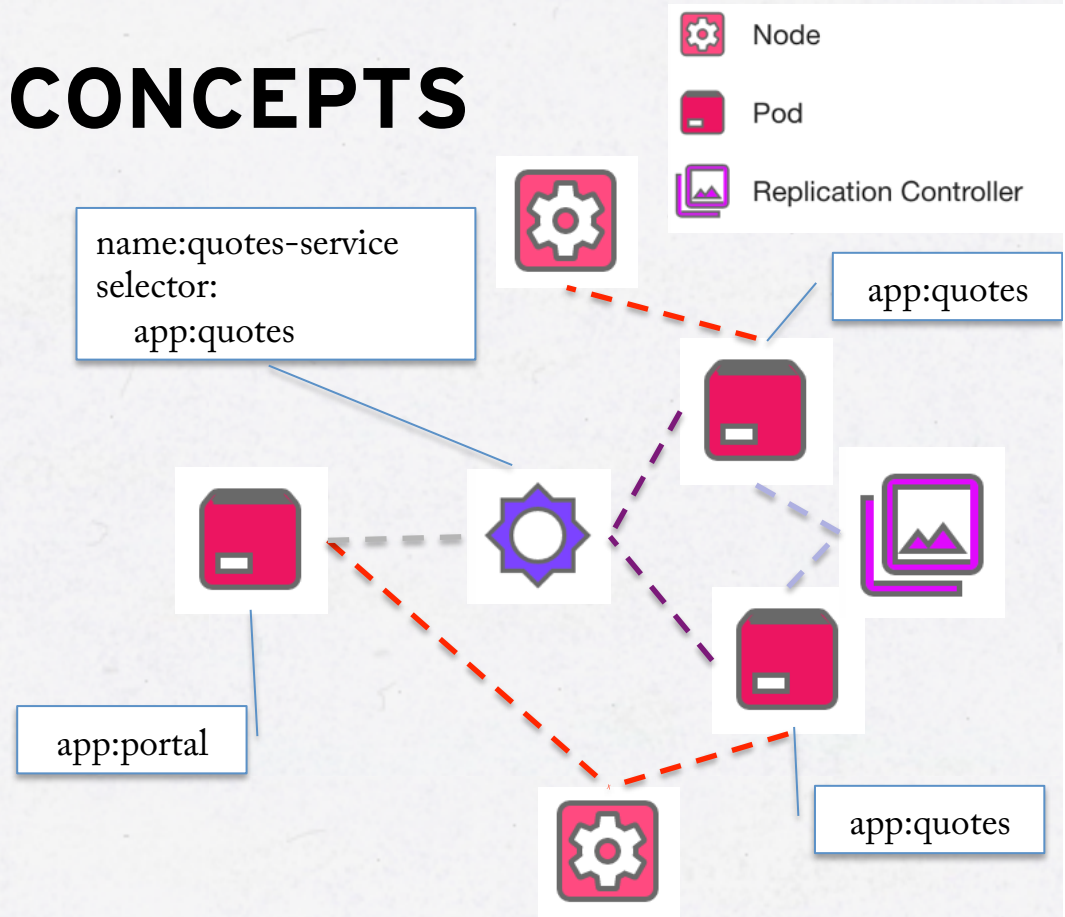
# CORE CONCEPTS

- Node
- Container
- Pod
- Replication Controller
- Label
- Liveness
- Service



# CORE CONCEPTS

- Node
- Container
- Pod
- Replication Controller
- Label
- Liveness
- Service



## QUOTES-SERVICE.YAML

```
apiVersion: v1
kind: Service
metadata:
  name: quotes-service
  labels:
    app: quotes
    tier: frontend
spec:
  ports:
    - port: 8080
      targetPort: 8080
  selector:
    app: quotes
```

name:quotes-service  
selector:  
app:quotes



Node



Pod



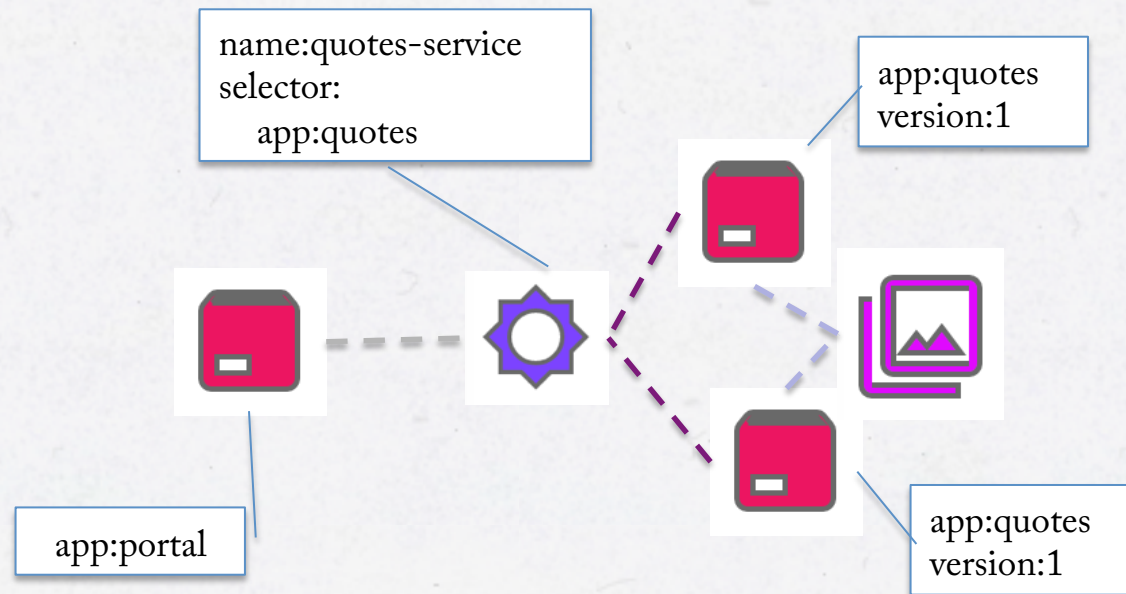
Replication Controller



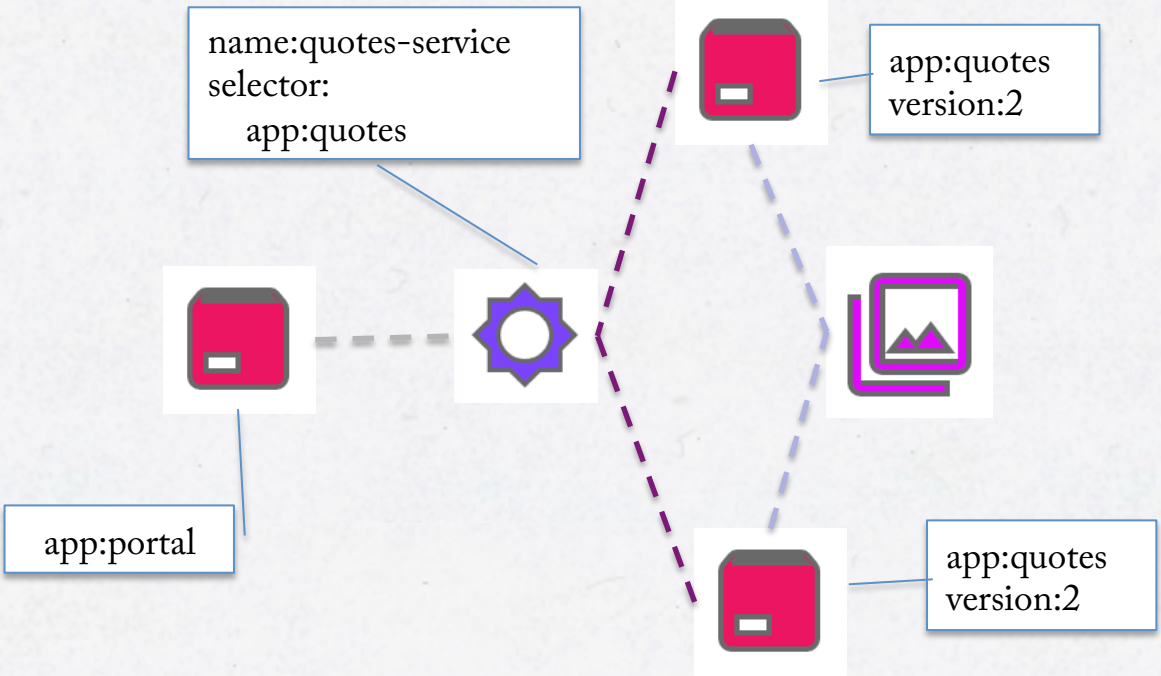


**DEMO**

# ADVANCED CONCEPTS: ROLLING UPGRADE

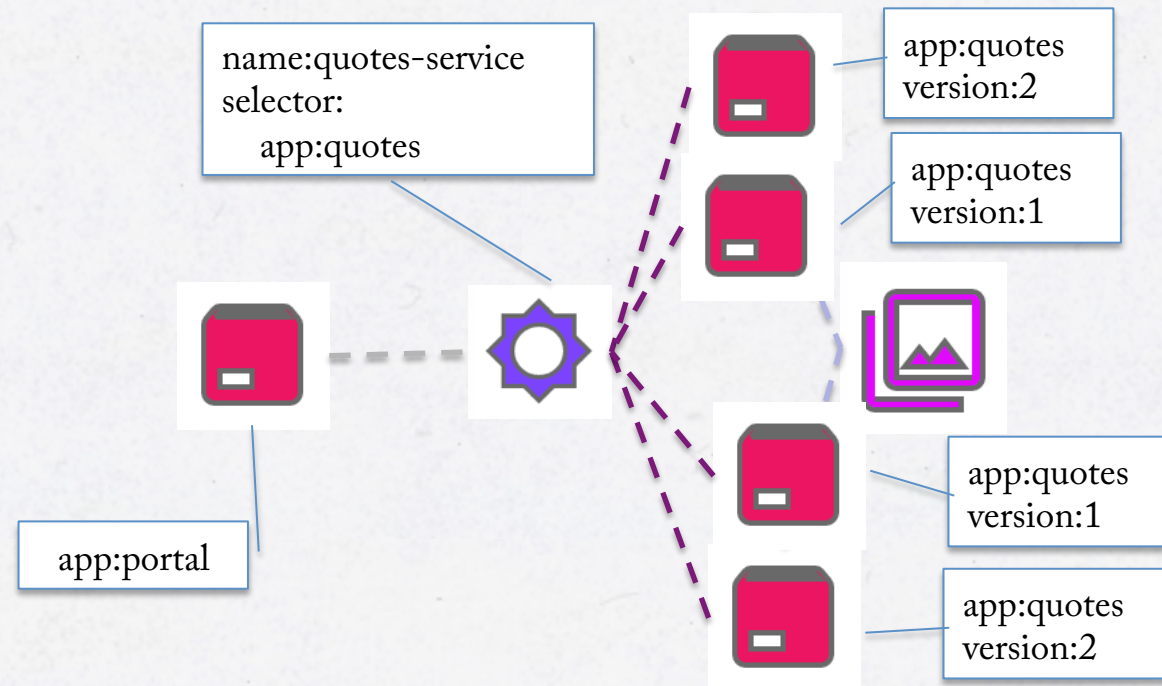


# ADVANCED CONCEPTS: ROLLING UPGRADE





# ADVANCED CONCEPTS: ROLLING UPGRADE





**DEMO**

## CONCLUSIONS



Kubernetes

- Container Orchestration software like Kubernetes provides highly valuable capabilities for Microservice architectures:
  - Decoupling logical components from each other and from the infrastructure
- Kubernetes is just one of many alternatives (Mesos, Helios, Docker Swarm, ...), but its Google background makes it one of the most interesting. Stay tuned!

## TIME FOR QUESTIONS?

