



Kubernetes



实战系列 — 入门篇

龚正 2016-01-27

大纲

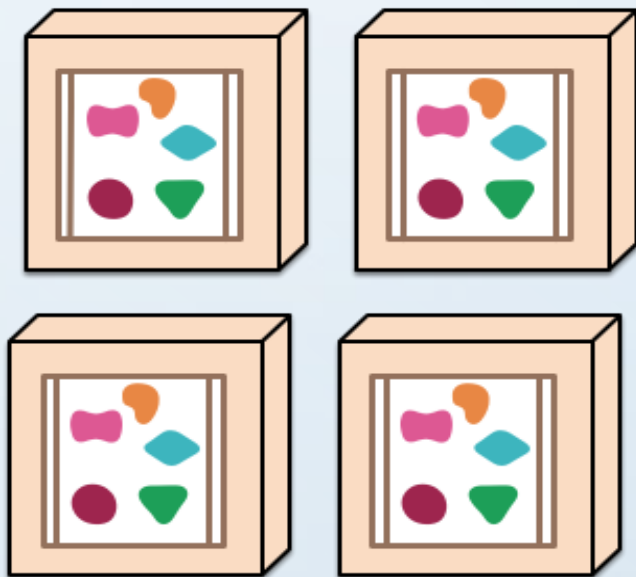
- Kubernetes概述
- Kubernetes核心概念
- Demo

微服务

A monolithic application puts all its functionality into a single process...



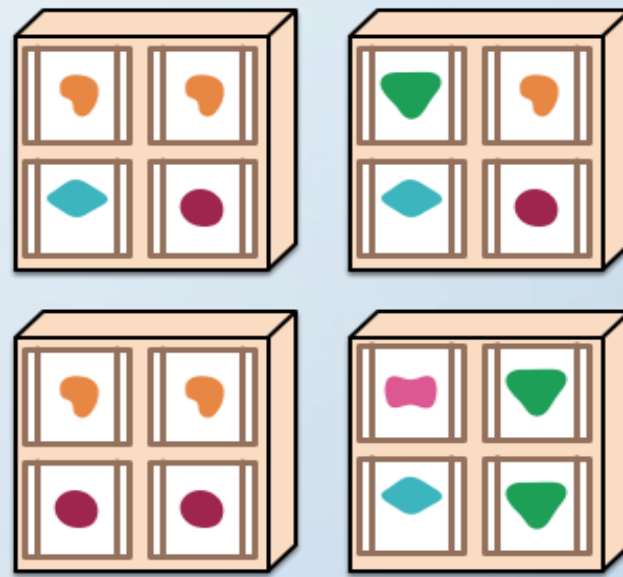
... and scales by replicating the monolith on multiple servers



A microservices architecture puts each element of functionality into a separate service...



... and scales by distributing these services across servers, replicating as needed.

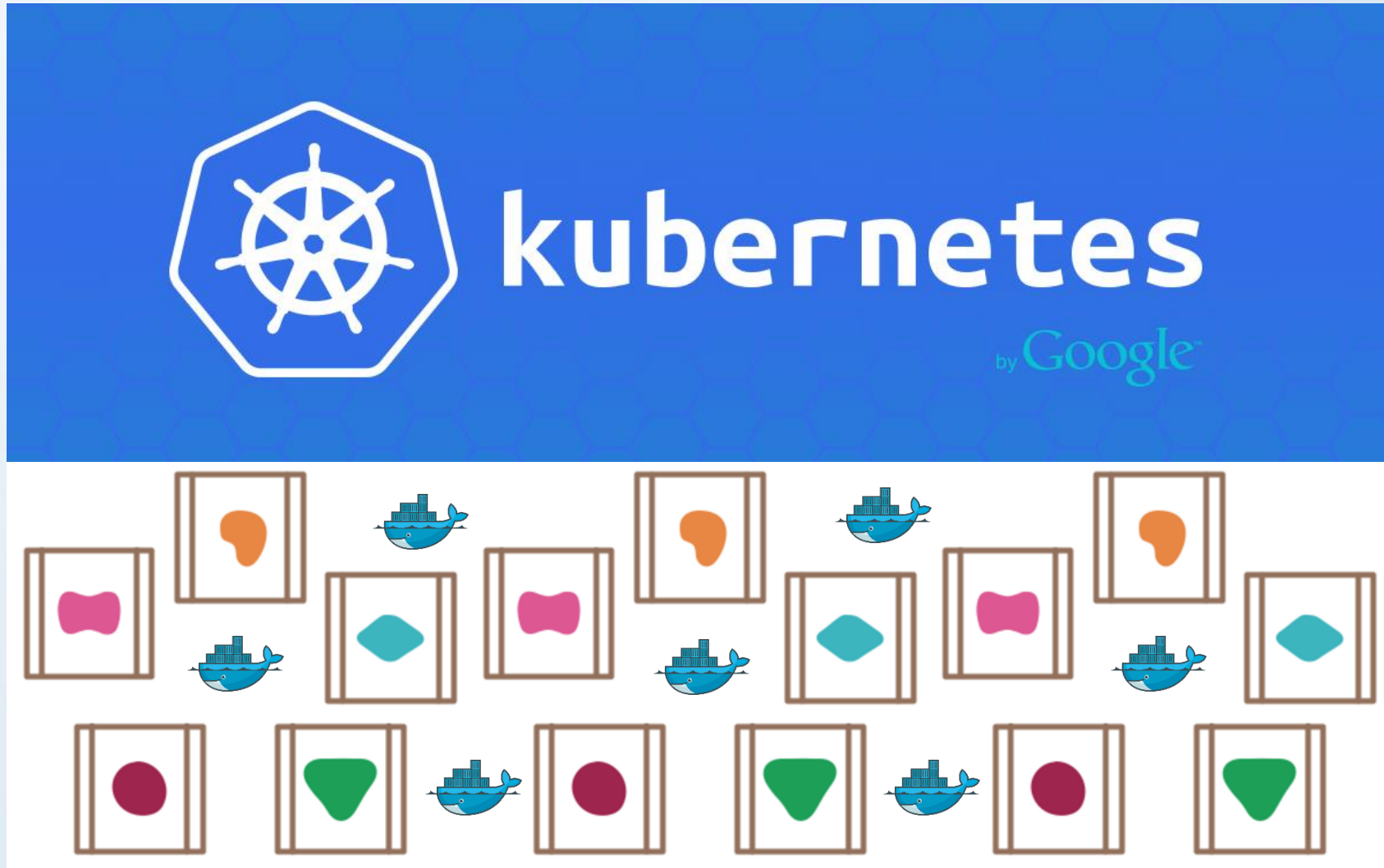


容器

- 轻量级虚拟化
- 应用程序+依赖包=镜像
- 运行环境 (namespaces、 cgroups)



Kubernetes



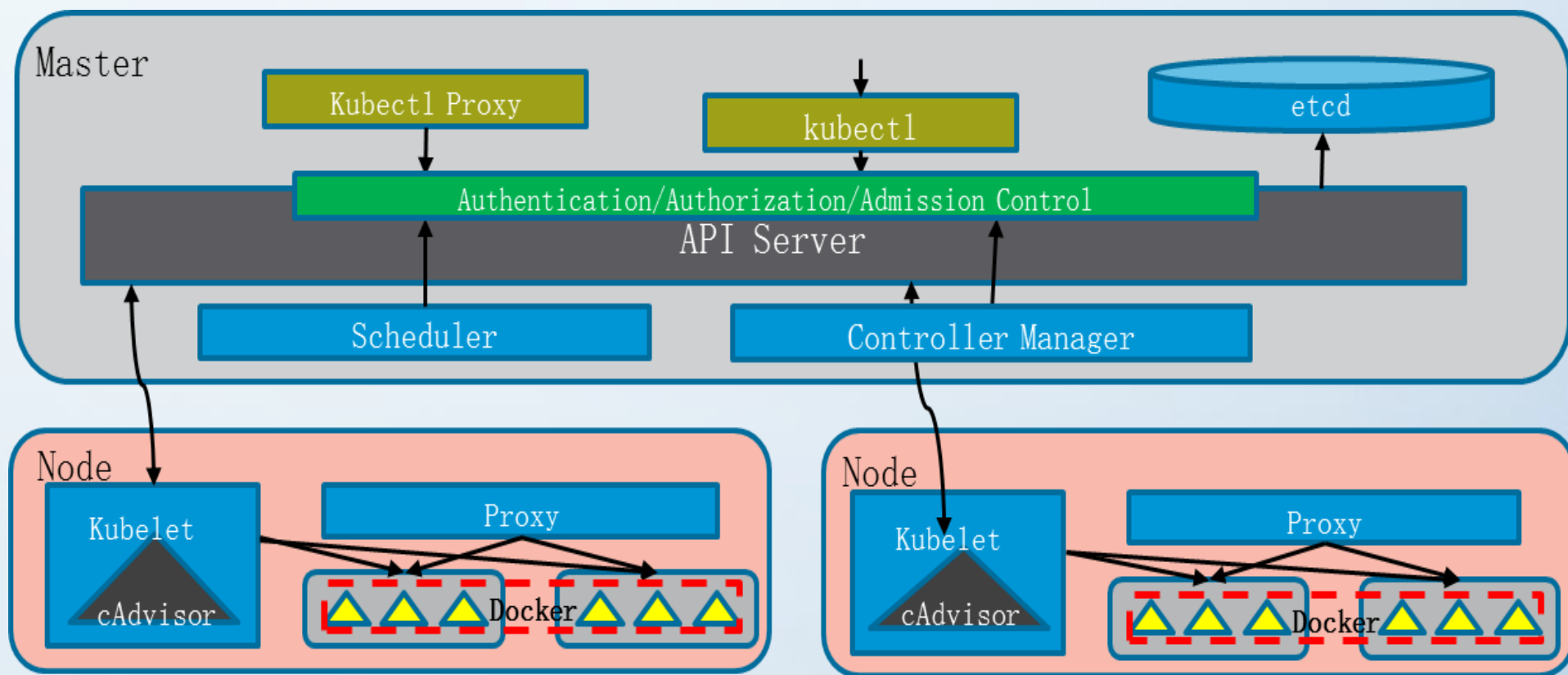
Kubernetes 简介

κυβερνήτης 古希腊语，意为舵手或导航员

- 容器集群管理
- 大量借鉴Borg的运营经验
- 2015年7月 v1.0版本发布
- 开源，Go语言
- 面向应用，而不是面向机器



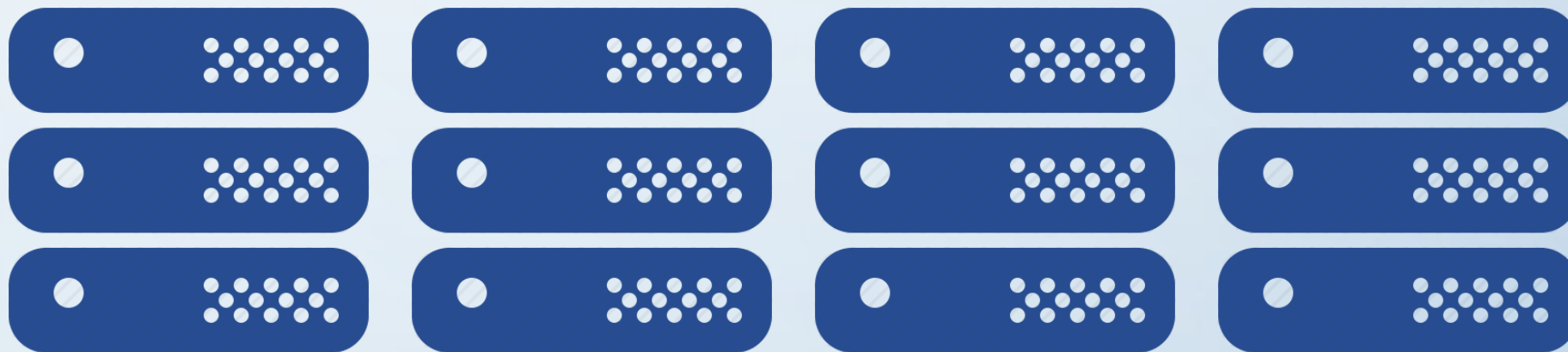
Kubernetes 总体系统架构



Kubernetes 核心概念 – Node

提供计算资源和存储资源

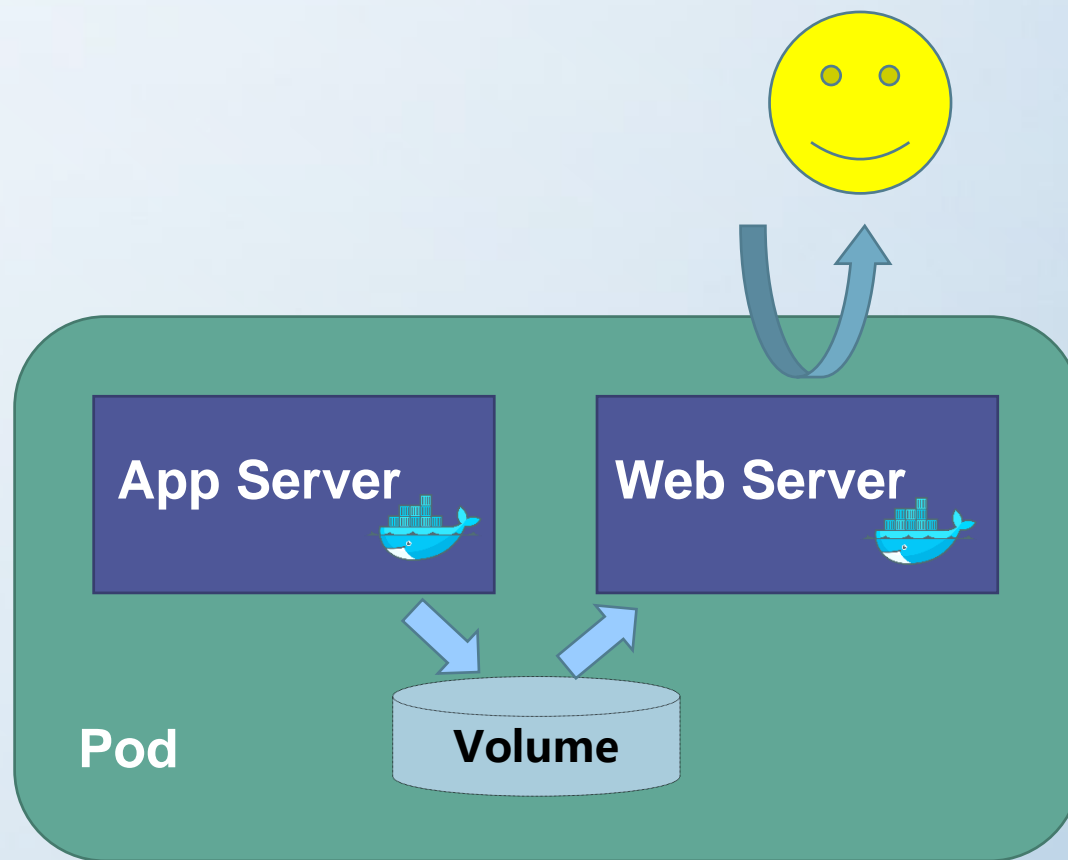
- docker
- kubelet、 proxy



Kubernetes 核心概念 – Pod

最小调度单元

- 包含一组容器 (1~n)
- 能够提供服务
- 容器间紧耦合
 - 共享namespace
 - 共享网络空间



Kubernetes 核心概念 – Scheduler

将Pod调度到Node上

- 基于Pod需求选择最适合的Node
- 插件式调度算法



Kubernetes 核心概念 – Replication Controller

管理一组Pod

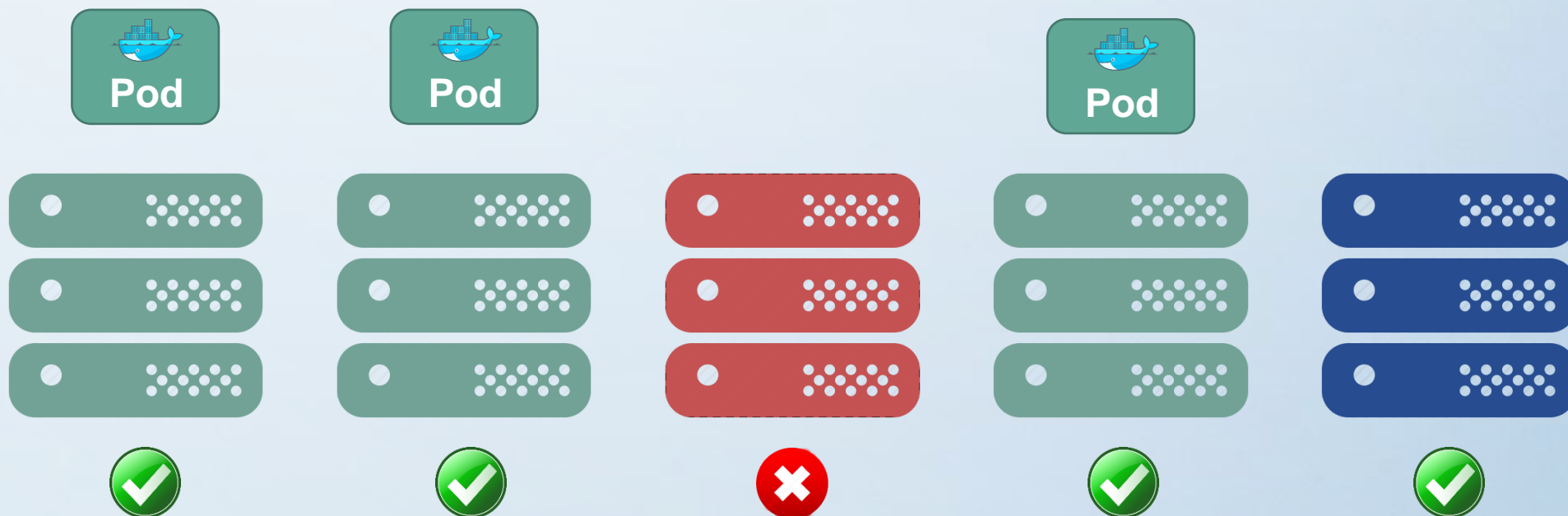
- 确保用户期望的Pod数量（创建或销毁Pod）
- 在线Pod数量调整



Kubernetes 核心概念 – Replication Controller

管理一组Pod

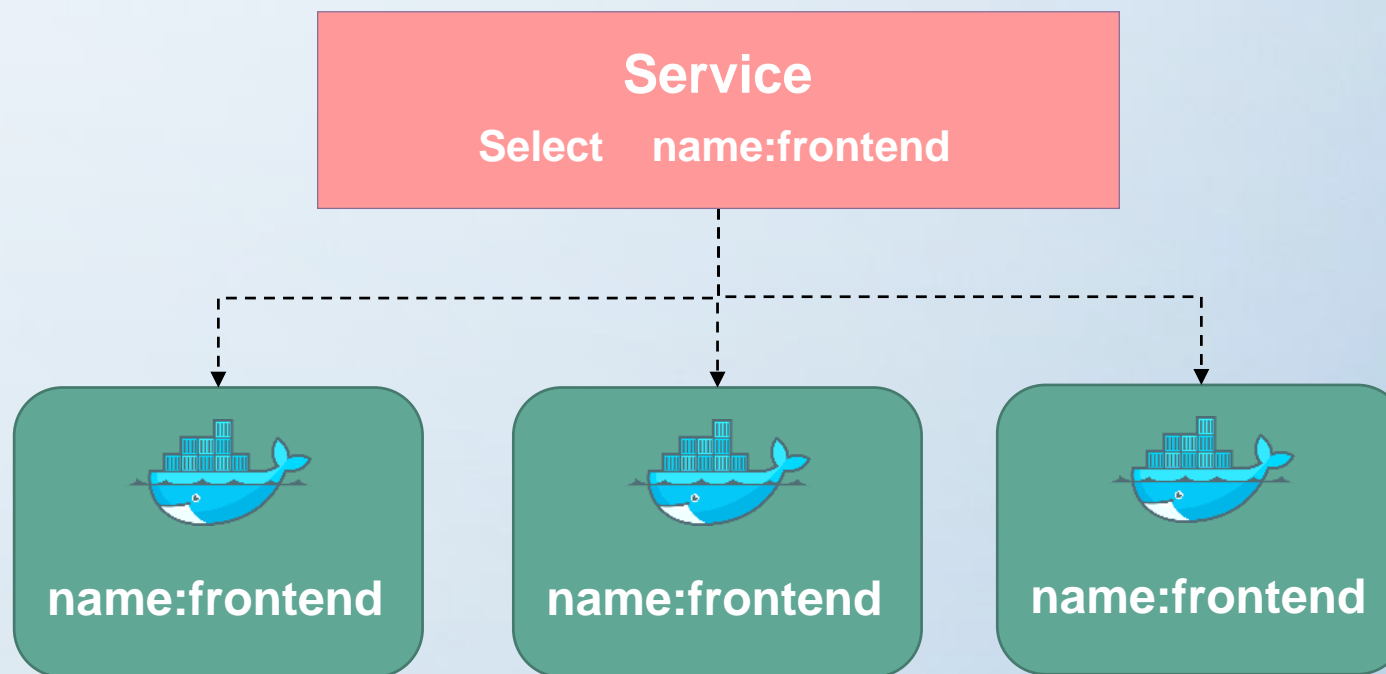
- 确保用户期望的Pod数量（创建或销毁Pod）
- 故障恢复



Kubernetes 核心概念 – Service

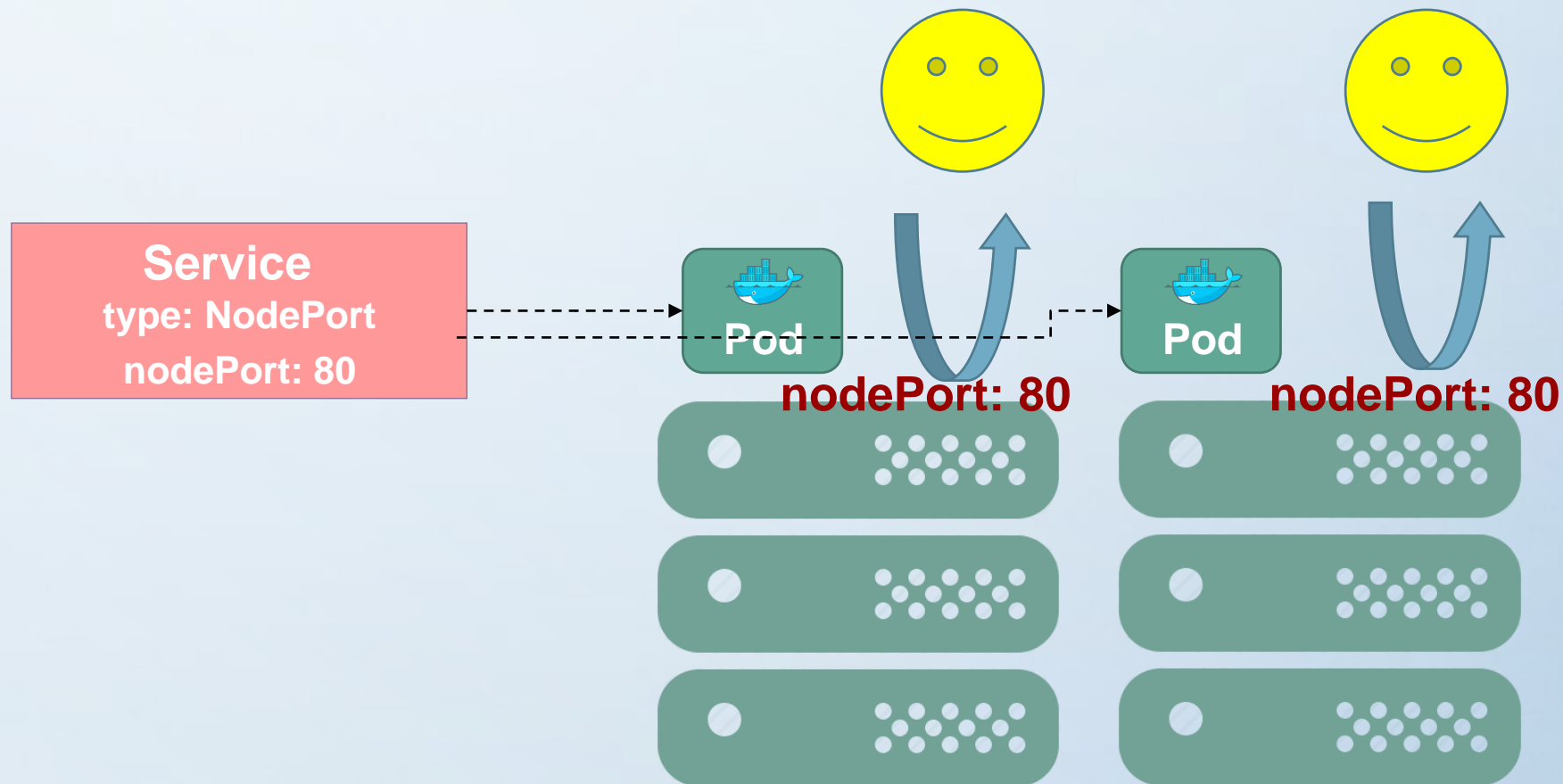
对Pod的服务发现机制

- 虚拟ClusterIP地址，虚拟端口号
- 动态负载分发到后端Pod



Kubernetes 核心概念 – Service

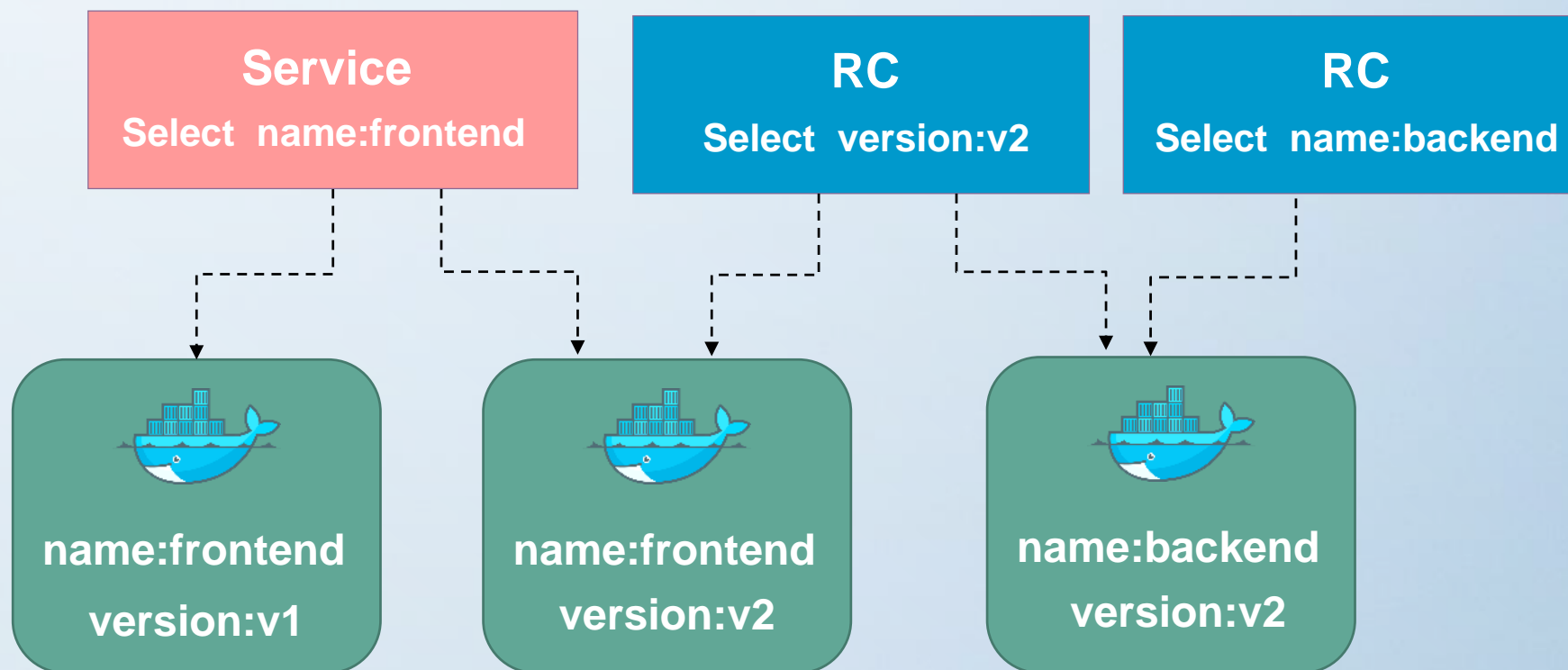
- 外部访问：NodePort
- 多端口



Kubernetes 核心概念 – Label

给任意对象打标签

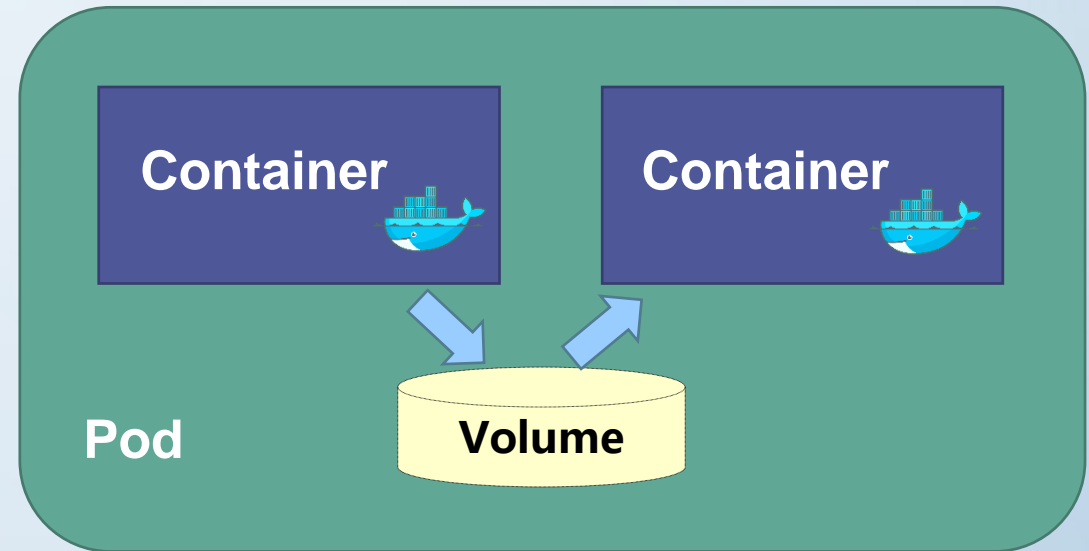
- 对象元数据
- 查询关键字



Kubernetes 核心概念 – Volume

容器共享存储卷

- EmptyDir
- hostPath
- nfs
- gcePersistentDisk
- awsElasticBlockStore
-



Kubernetes 核心概念 – Namespace

分组管理

- 多租户
- 集群资源共享
- 资源限制
 - ResourceQuota (整个Namespace的配额)
 - LimitRange (Pod和容器的限制)

声明型的配置

- 给我服务A创建5个Pod
- 5个不够，给我再增加10个
- 如果有某个Pod挂了，请自动帮我恢复
- 我需要将应用部署到5台服务器上，每台服务器安装一个应用，每个应用的配置为xxx...
- 我需要在新增的10台服务器上新部署10个应用，每个应用的配置为xxx...

Pod的声明

frontend-pod.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: frontend
  labels:
    name: frontend
spec:
  containers:
  - name: php-redis
    image: 192.168.1.128:1180/
      kubeguide/guestbook-php-frontend
    ports:
    - containerPort: 80
```

frontend-pod.json

```
{
  "apiVersion": "v1",
  "kind": "Pod",
  "metadata": {
    "name": "frontend",
    "labels": { "name": "frontend" }
  },
  "spec": {
    "containers": [
      {
        "name": "php-redis",
        "image": "192.168.1.128:1180/kubeguide
          /guestbook-php-frontend",
        "ports": [
          { "containerPort": 80 }
        ]
      }
    ]
  }
}
```

Replication Controller的声明

frontend-rc.yaml

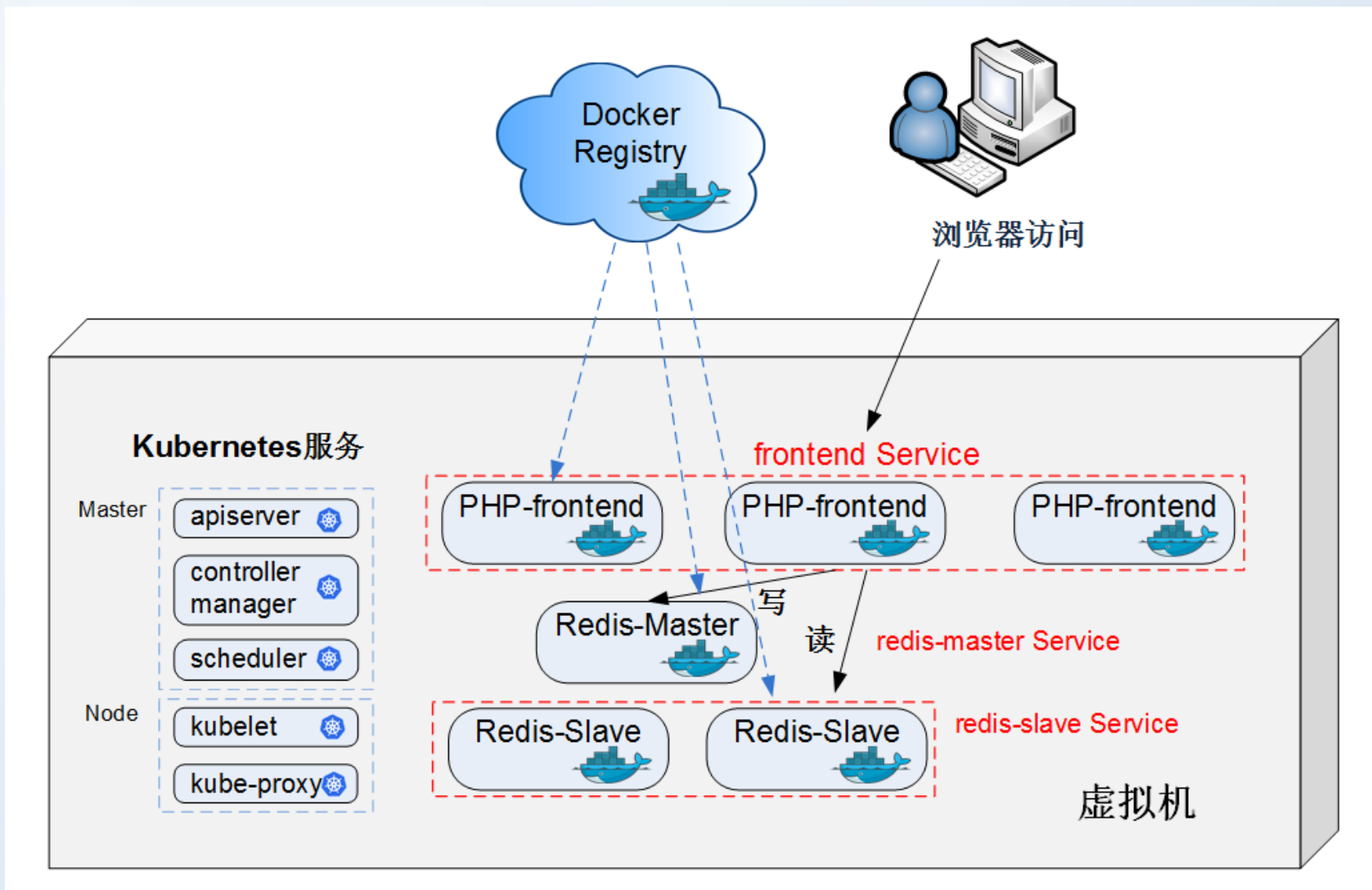
```
apiVersion: v1
kind: ReplicationController
metadata:
  name: frontend
  labels:
    name: frontend
spec:
  replicas: 3
  selector:
    name: frontend
  template:
    metadata:
      labels:
        name: frontend
    spec:
      containers:
        - name: php-redis
          image: 192.168.1.128:1180/kubeguide/guestbook-php-frontend
          ports:
            - containerPort: 80
```

Service的声明

frontend-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: frontend
  labels:
    name: frontend
spec:
  selector:
    name: frontend
  type:
    NodePort
  ports:
    - port: 80
      targetPort: 80
      nodePort: 30001
```

Demo



《Kubernetes权威指南》

