



# 云原生实战

Docker、Kubernetes、KubeSphere上云实战  
架构师的第一课

- ① Docker基础
- ② Kubernetes实战
- ③ KubeSphere容器平台
- ④ 应用上云实战

# 课程简介

- 云上基础架构
- Docker容器化
- 云原生Kubernetes实战
- 云上中间件部署
- 云上应用部署
- 云上ServiceMesh实战

- 开发工程师
- 传统运维工程师
- 运维开发工程师
- 测试工程师
- 传统架构师
- .....

## 不适合：

- 深度掌握Kubernetes、二次开发、云原生周边开发人员
- 资深运维需要自建私有云平台
- 云原生周边方案整合原理

- 实战方式快速系统了解云原生核心
- 完全云上实战、消除环境差异性
- 资源与笔记齐全

<https://www.yuque.com/leifengyang/oncloud>

- 提供答疑渠道

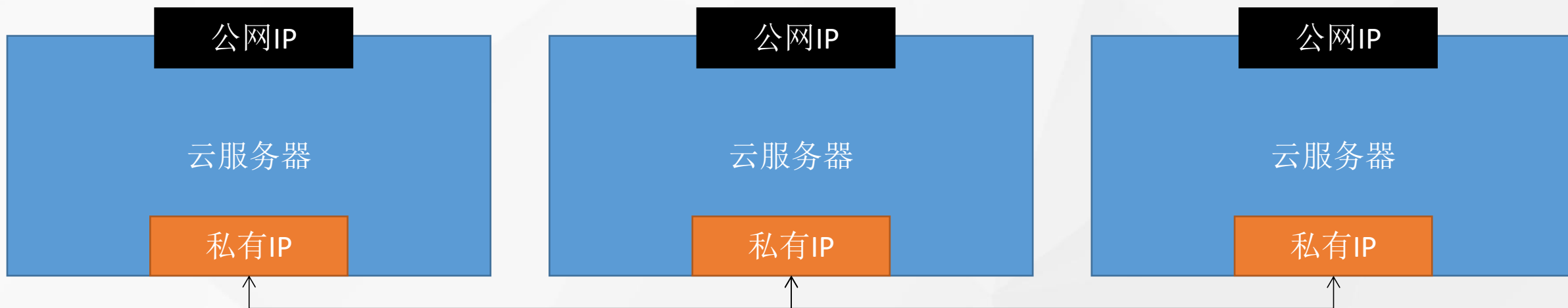
## 前置要求

- 会基本的linux操作
- 了解常用中间件，比如Nginx，MySQL，Redis等
- 分布式经验更佳，如SpringCloud

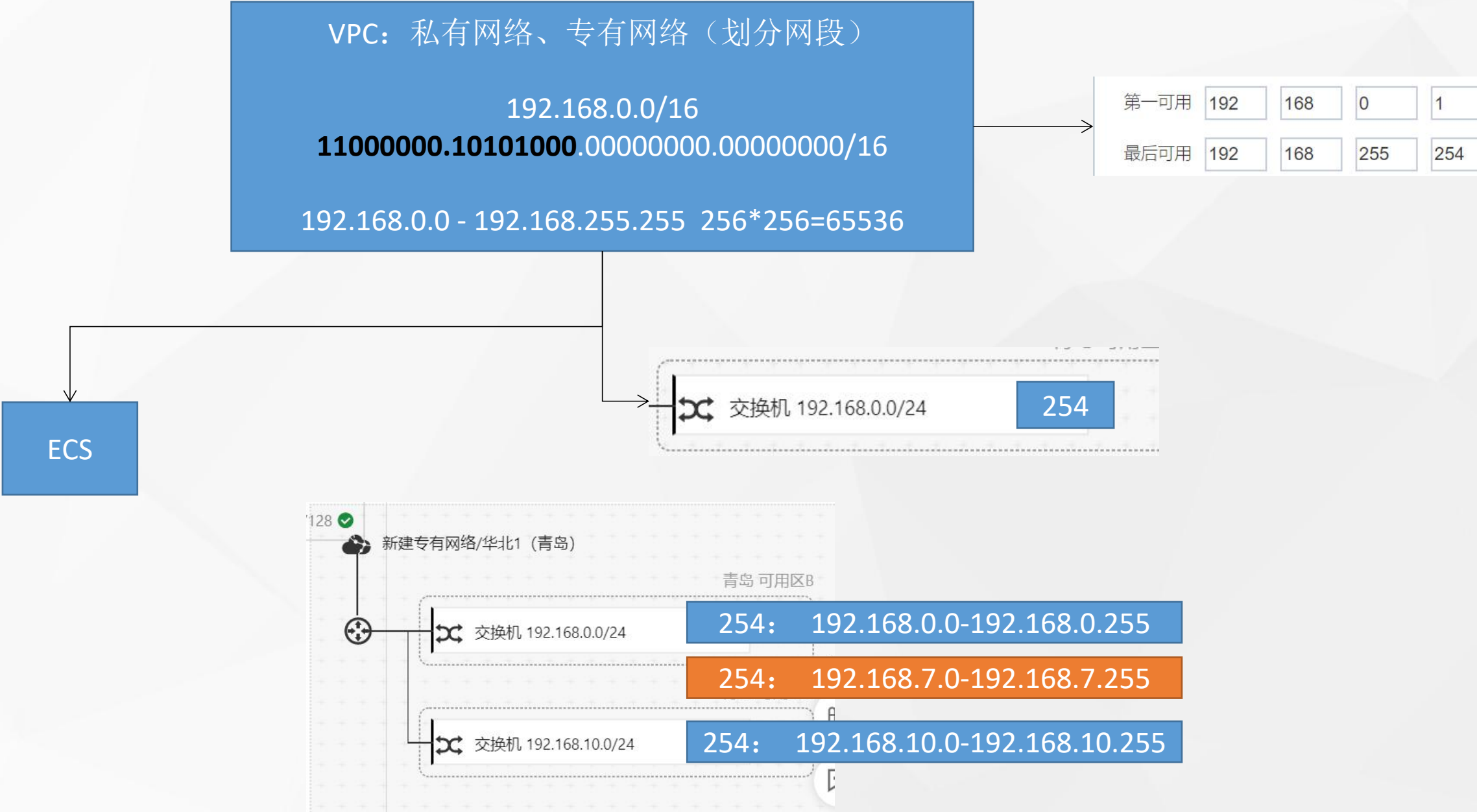
# 云平台操作

服务器、VPC、安全组...

安全组：防火墙相关的端口设置







VPC: 私有网络、专有网络（划分网段）

192.168.0.0/16

**11000000.10101000.00000000.00000000/16**

192.168.0.0 - 192.168.255.255 256\*256=65536

VPC: 私有网络、专有网络（划分网段）

192.168.0.0/16

**11000000.10101000.00000000.00000000/16**

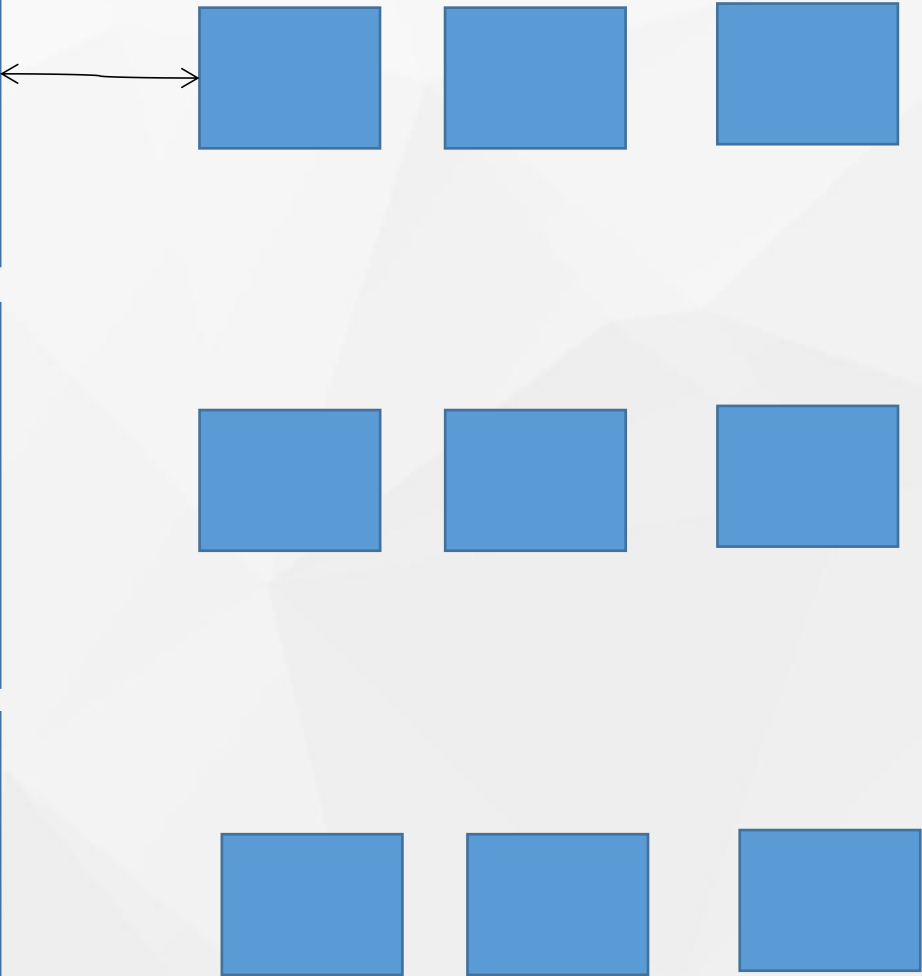
192.168.0.0 - 192.168.255.255 256\*256=65536

VPC: 私有网络、专有网络（划分网段）

192.168.0.0/16

**11000000.10101000.00000000.00000000/16**

192.168.0.0 - 192.168.255.255 256\*256=65536

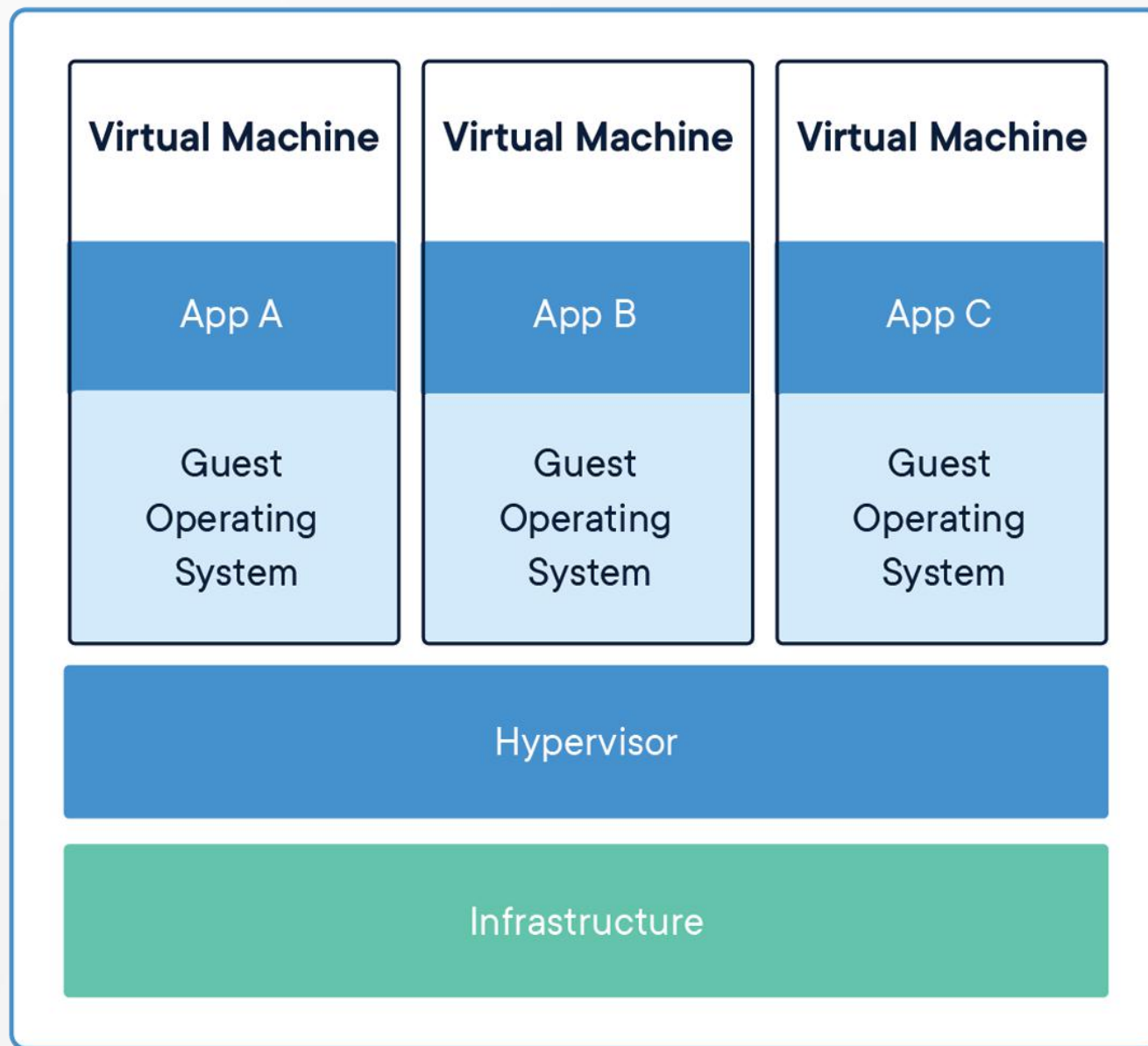


# Docker容器化实战

## 基础概念

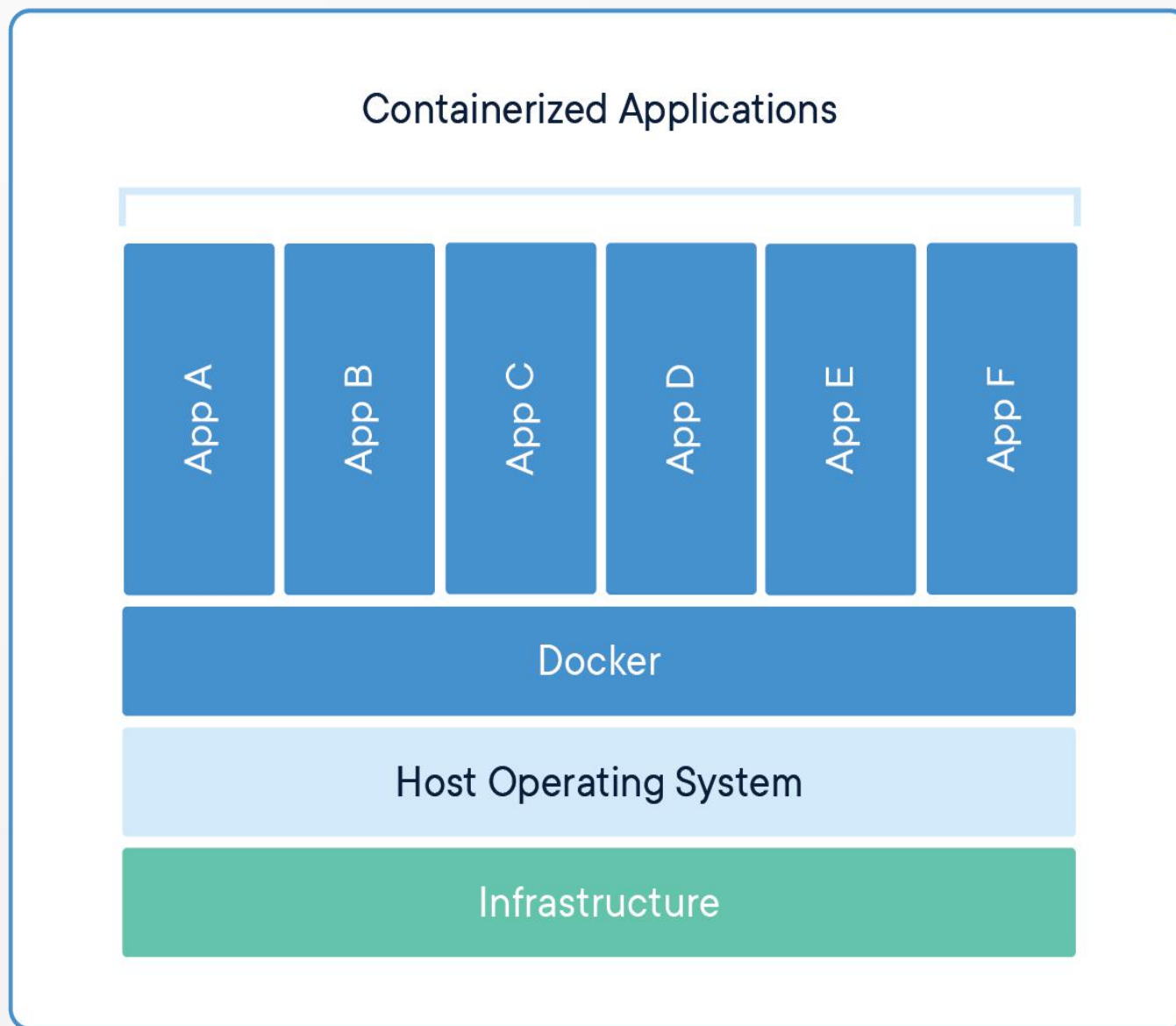
## 虚拟化技术：

- 1、基础镜像GB级别
- 2、创建使用稍微复杂
- 3、隔离性强
- 4、启动速度慢
- 5、移植与分享不方便
- ...



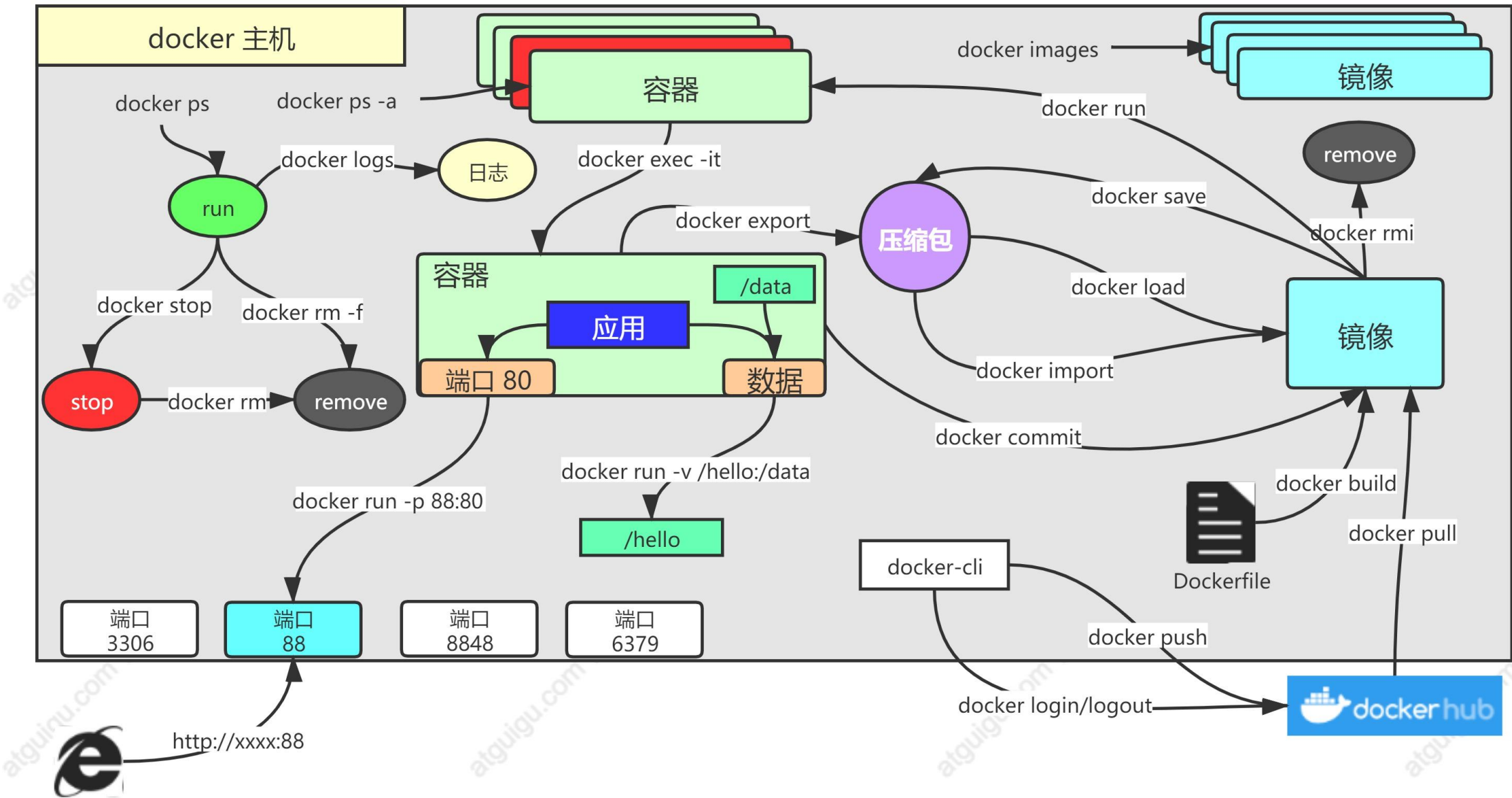
## 容器化技术：

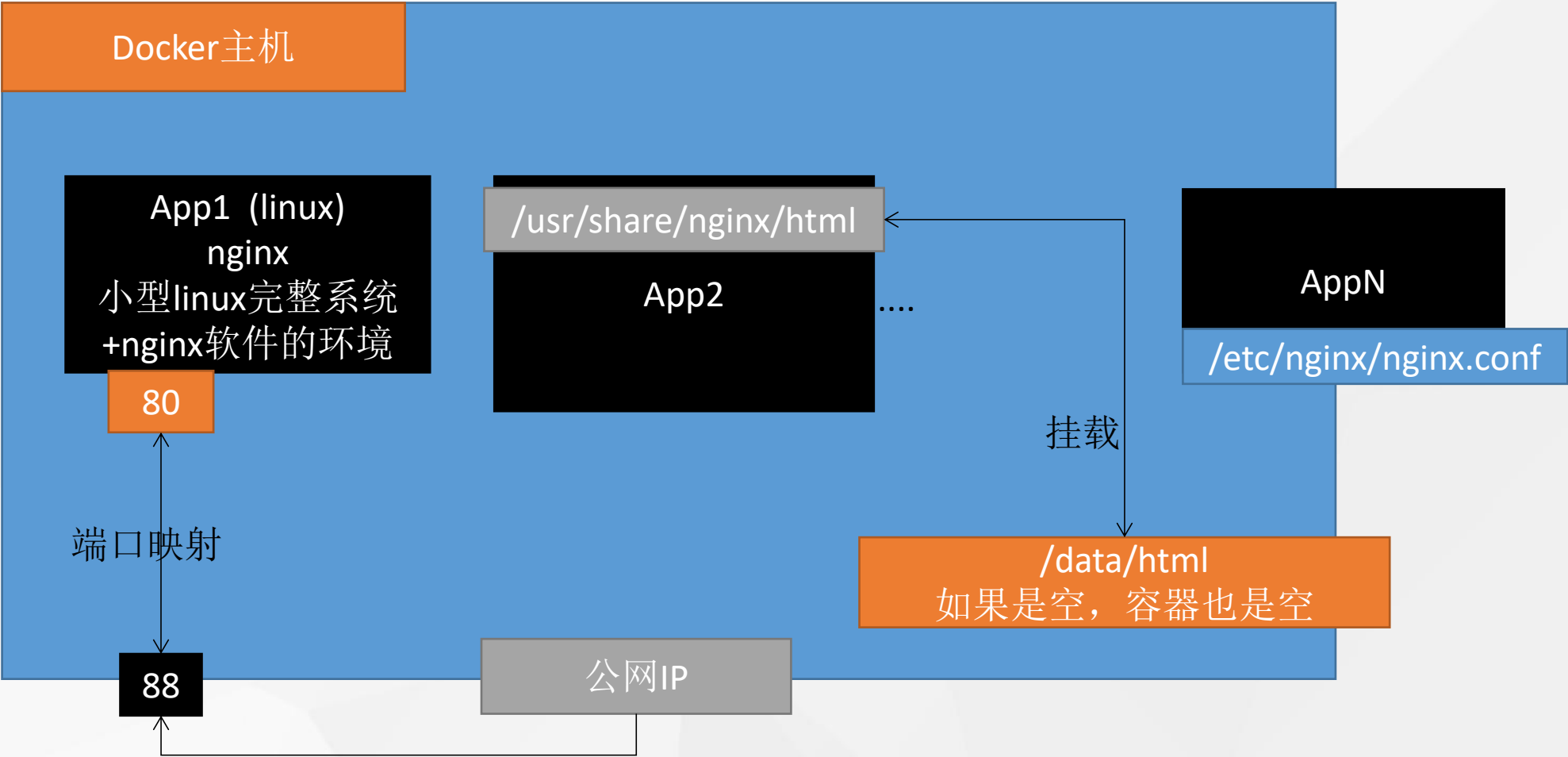
- 1、基础镜像MB级别
- 2、创建简单
- 3、隔离性强
- 4、启动速度秒级
- 5、移植与分享方便
- ...



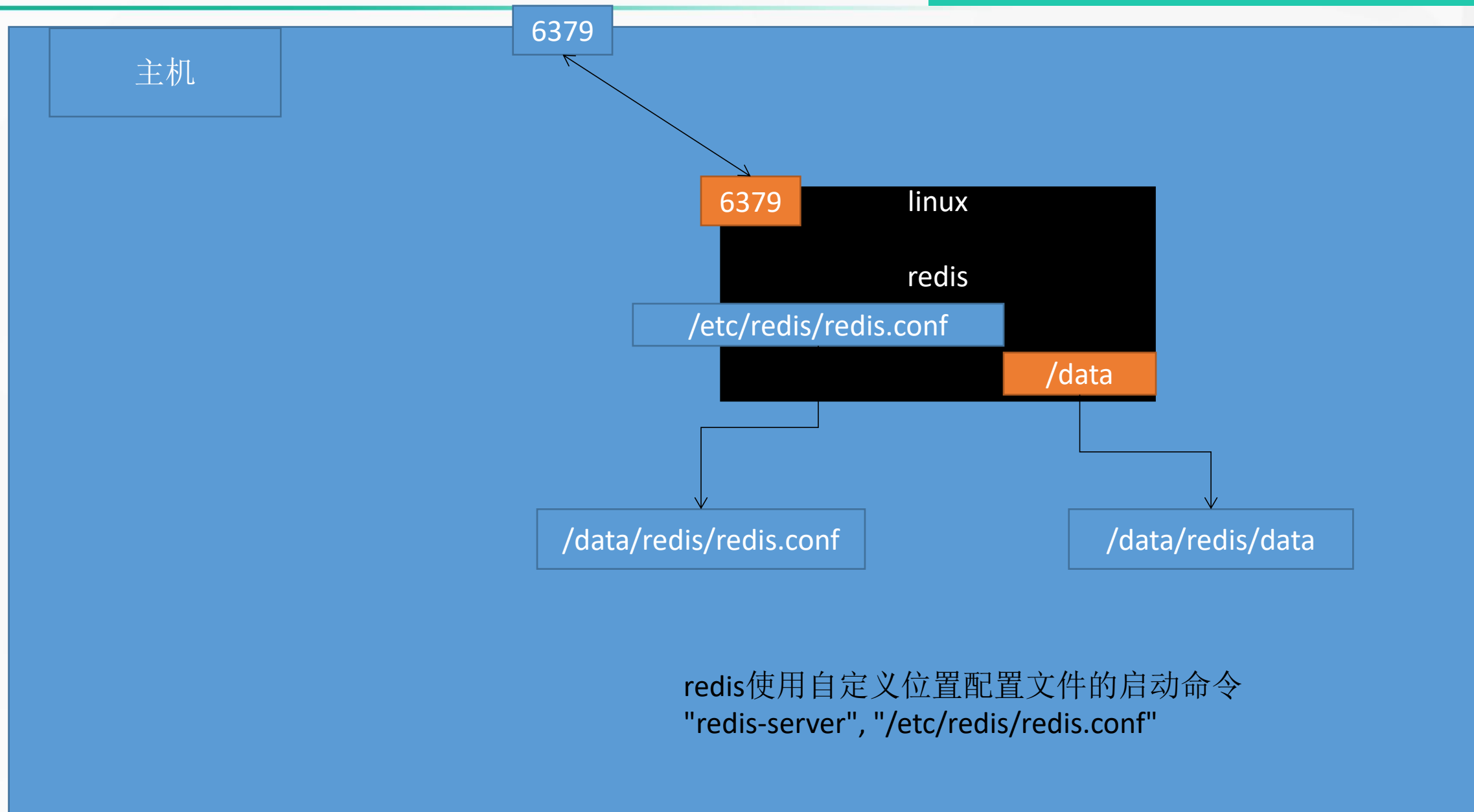
# Docker容器化实战

## 常用命令









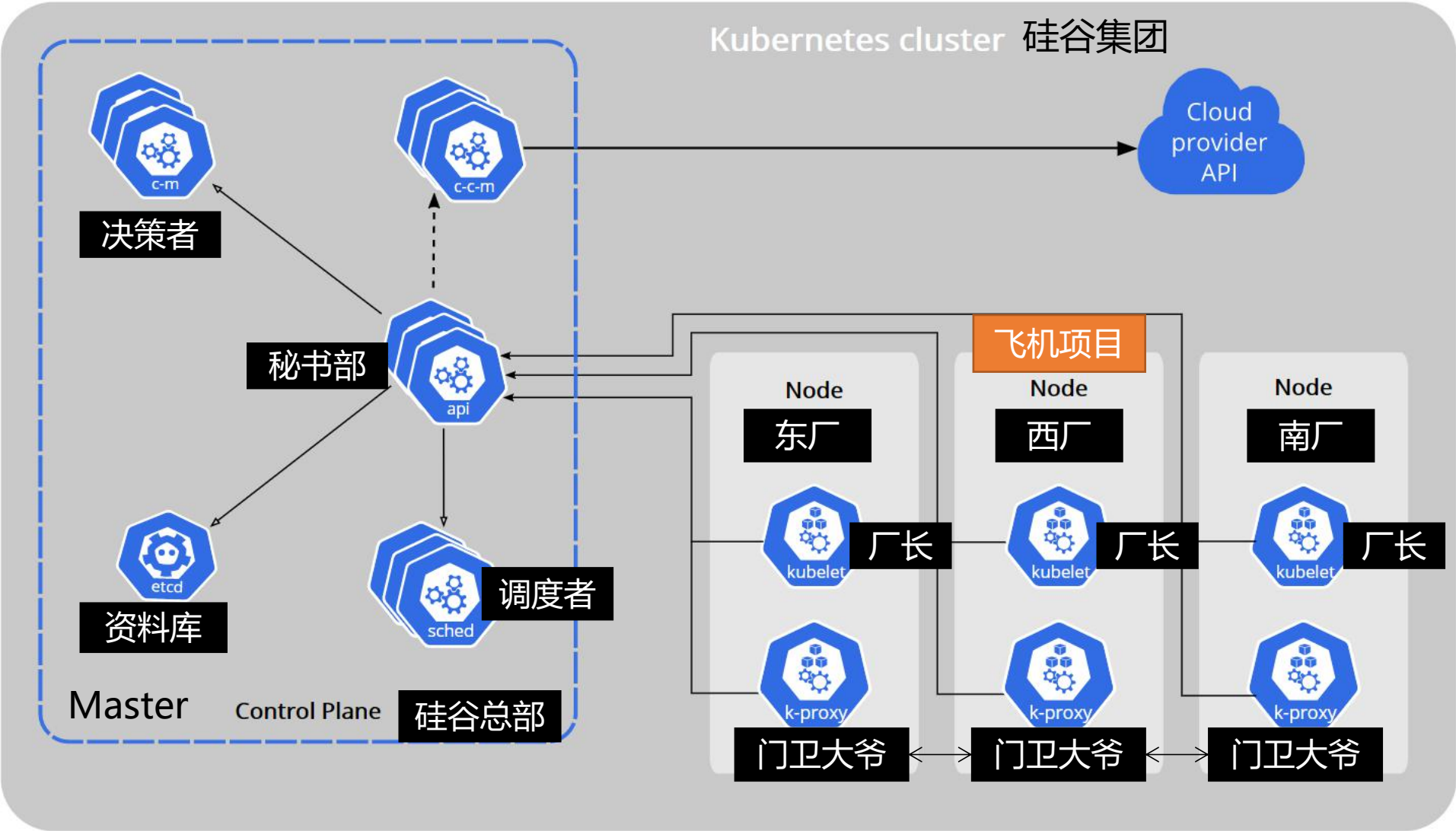
完整的linux

Java

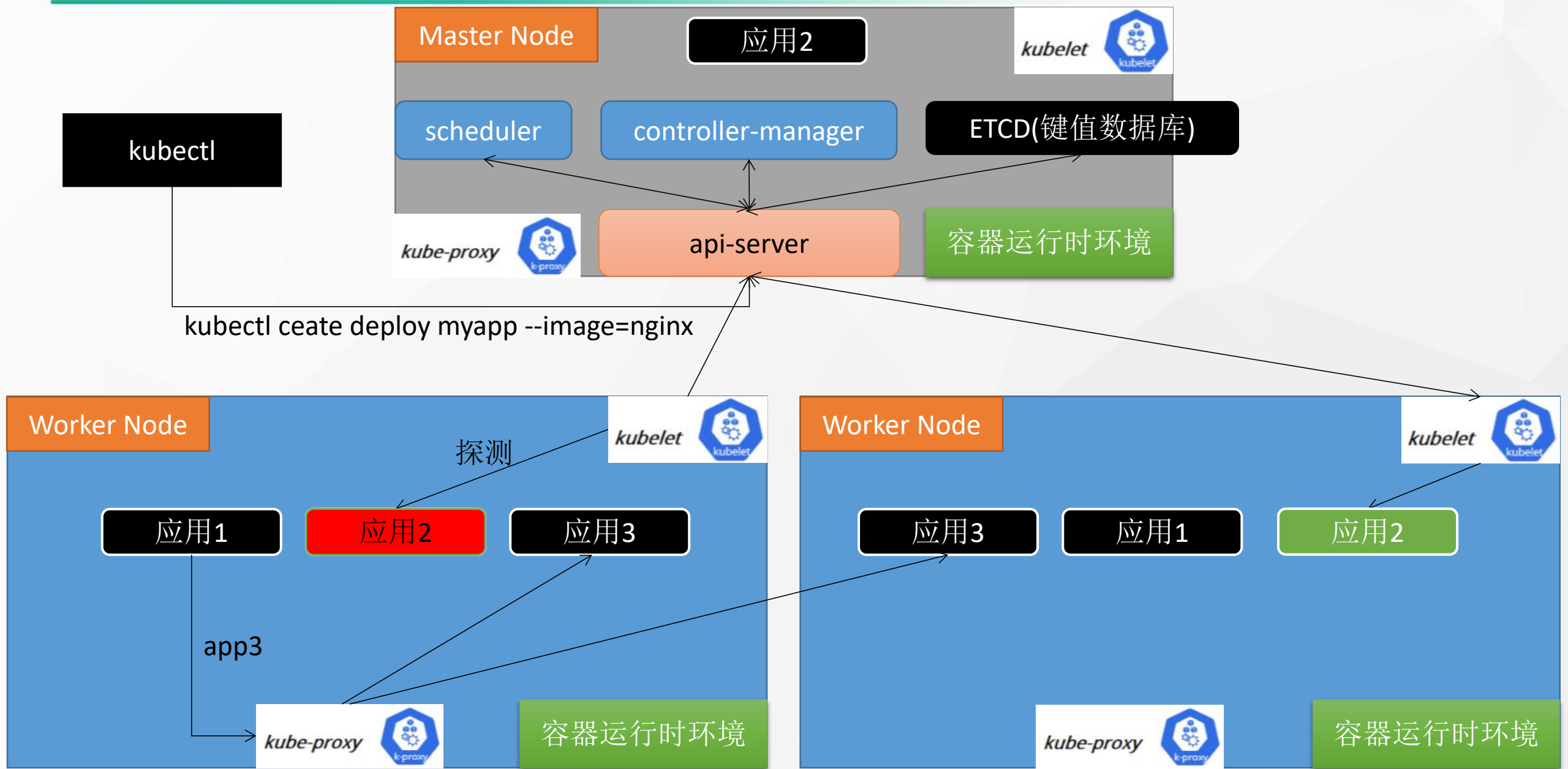
/app.jar

# Kubernetes基础

## 架构

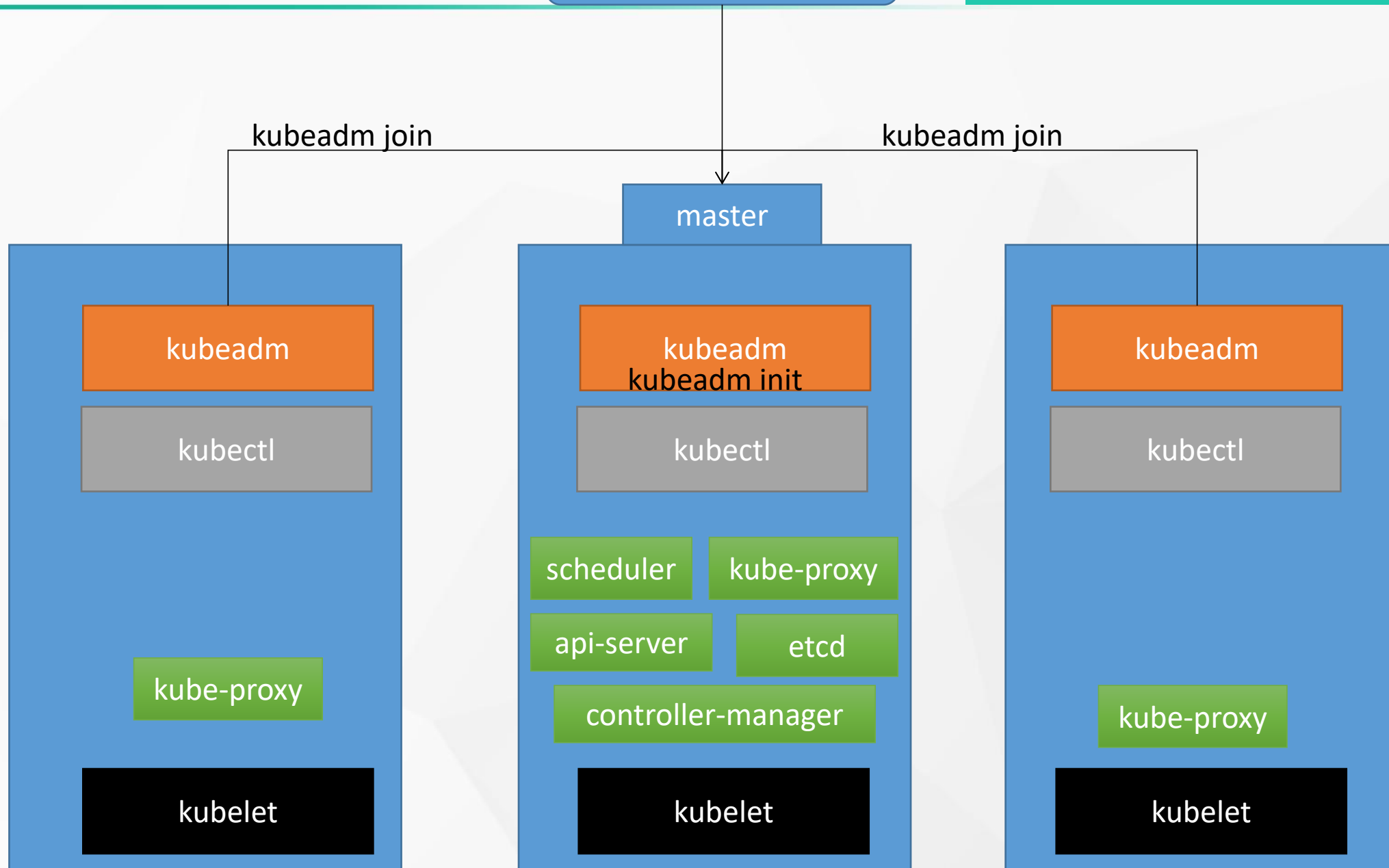


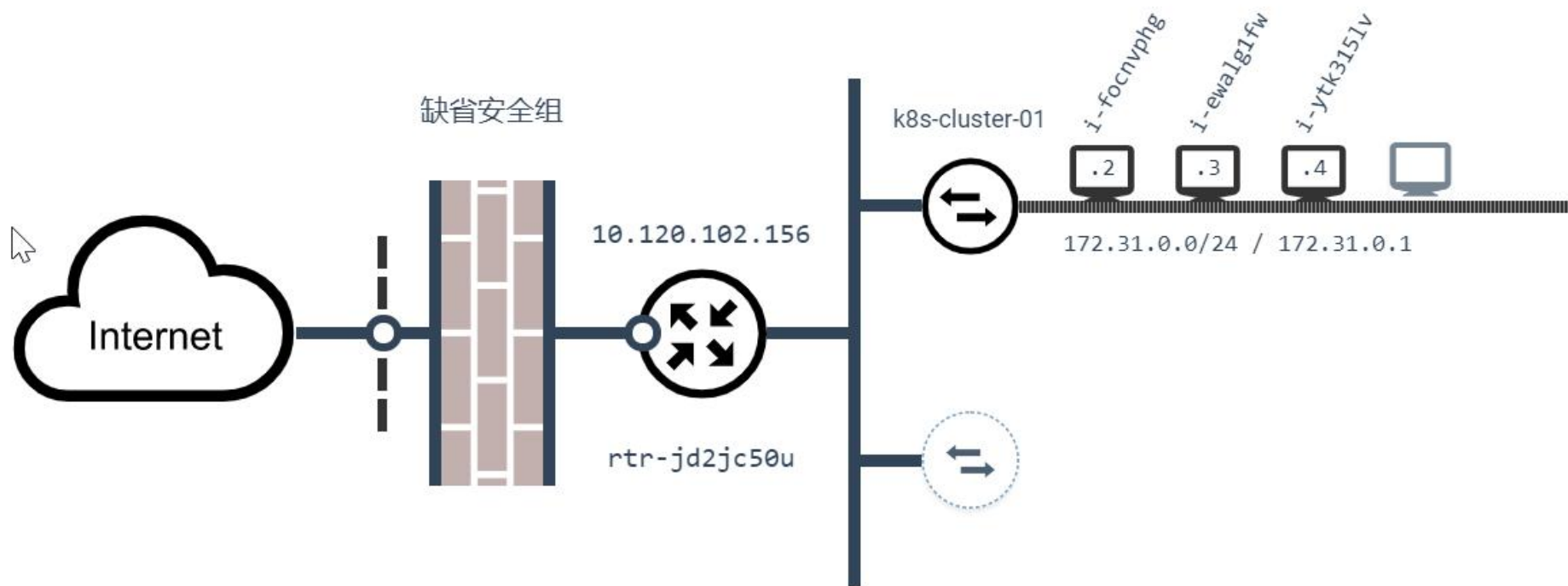
# Kubernetes架构-动画演示



# Kubernetes基础

## 集群安装





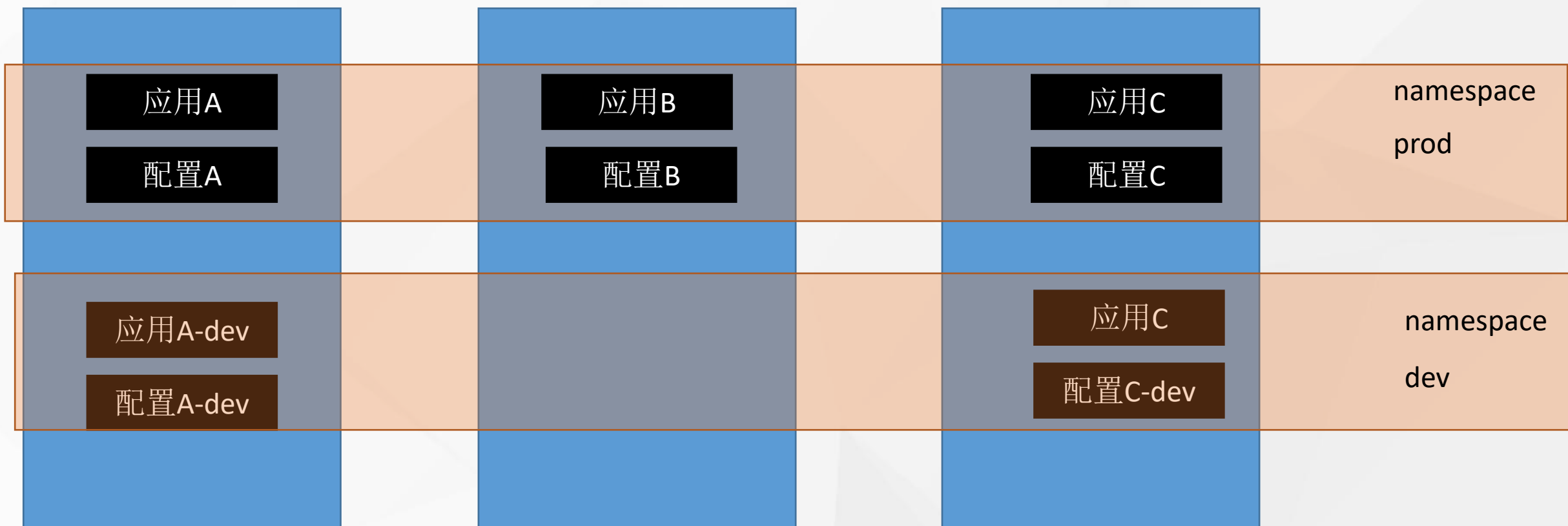


# Kubernetes核心实战

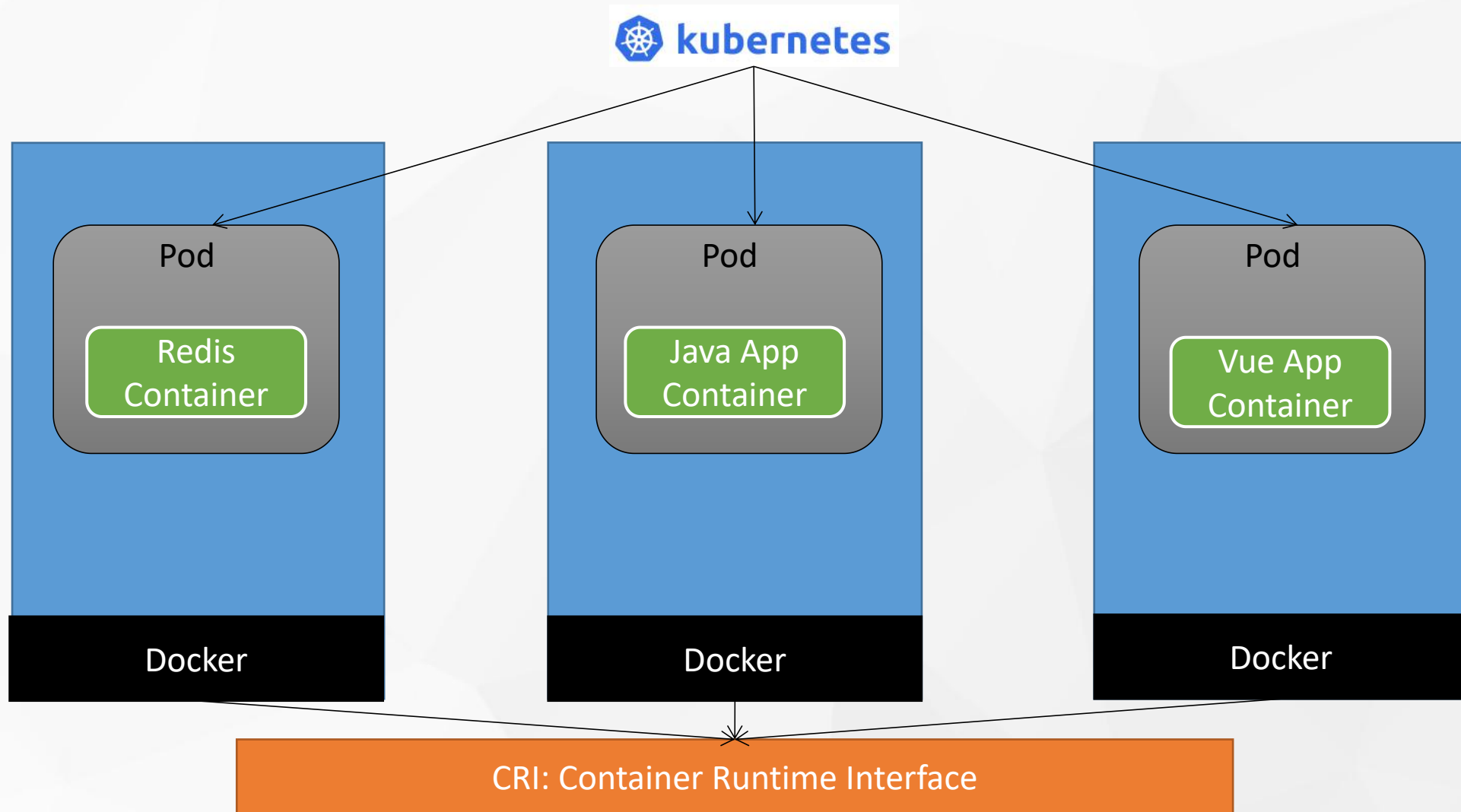
## 核心概念理解

## Namespace:

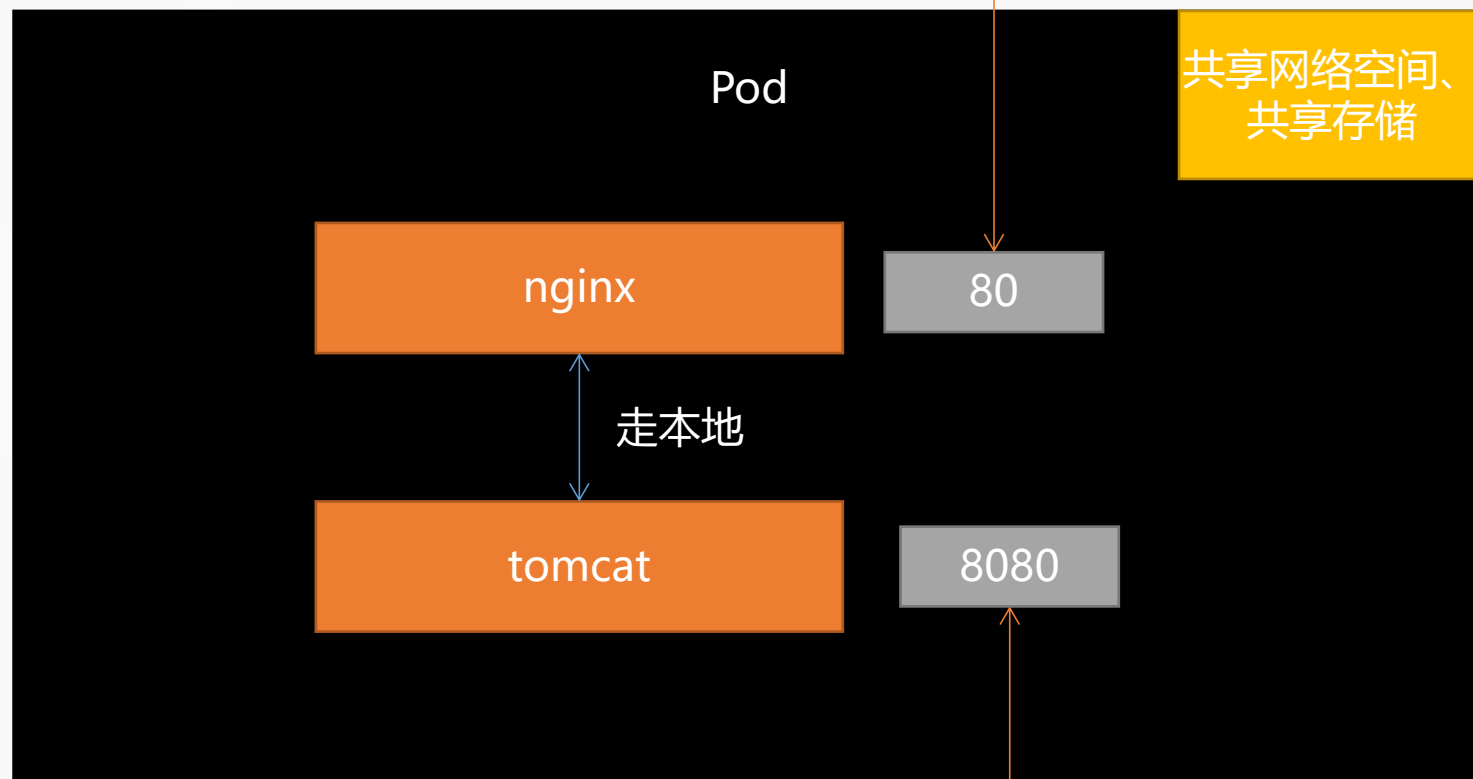
名称空间，用来对集群资源进行隔离划分。默认只隔离资源，不隔离网络



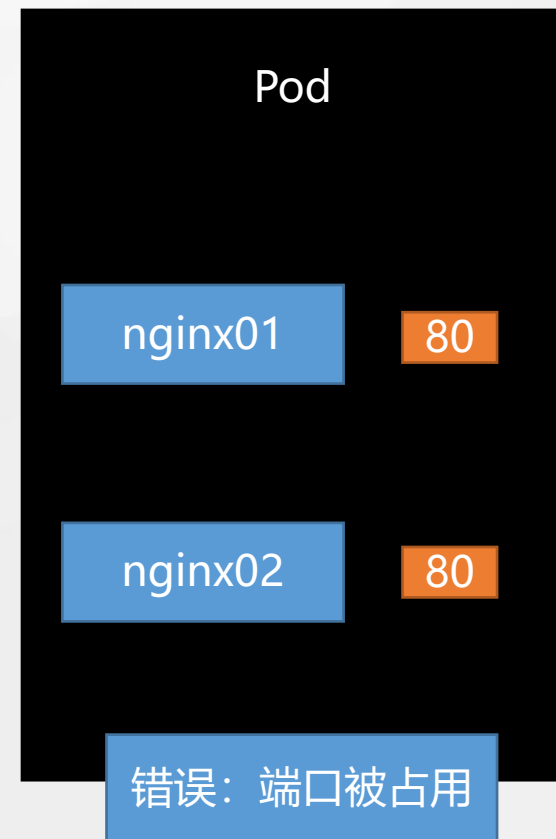
Pod: 运行中的一组容器, Pod是kubernetes中应用的最小单位.



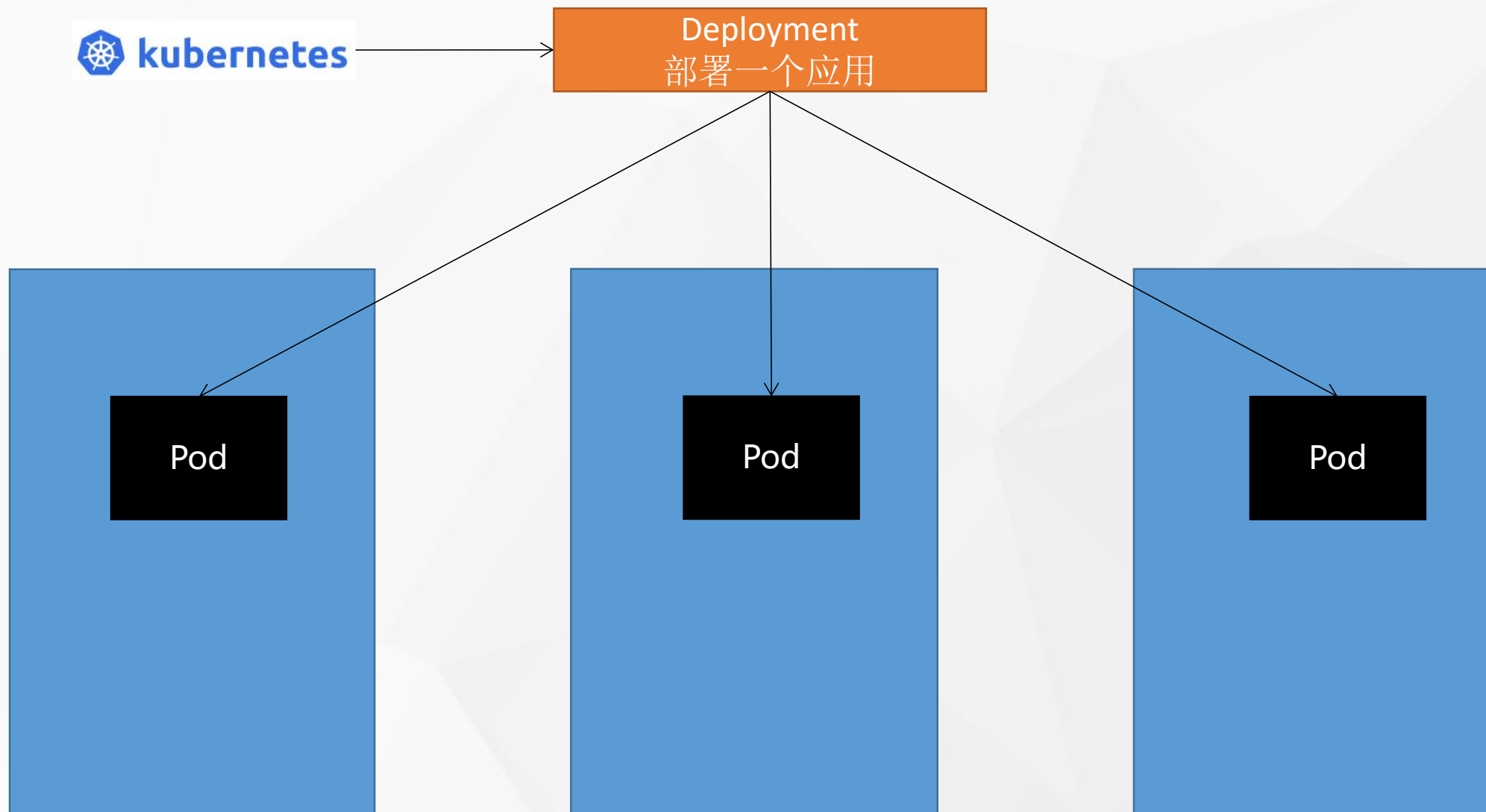
192.168.36.68:80



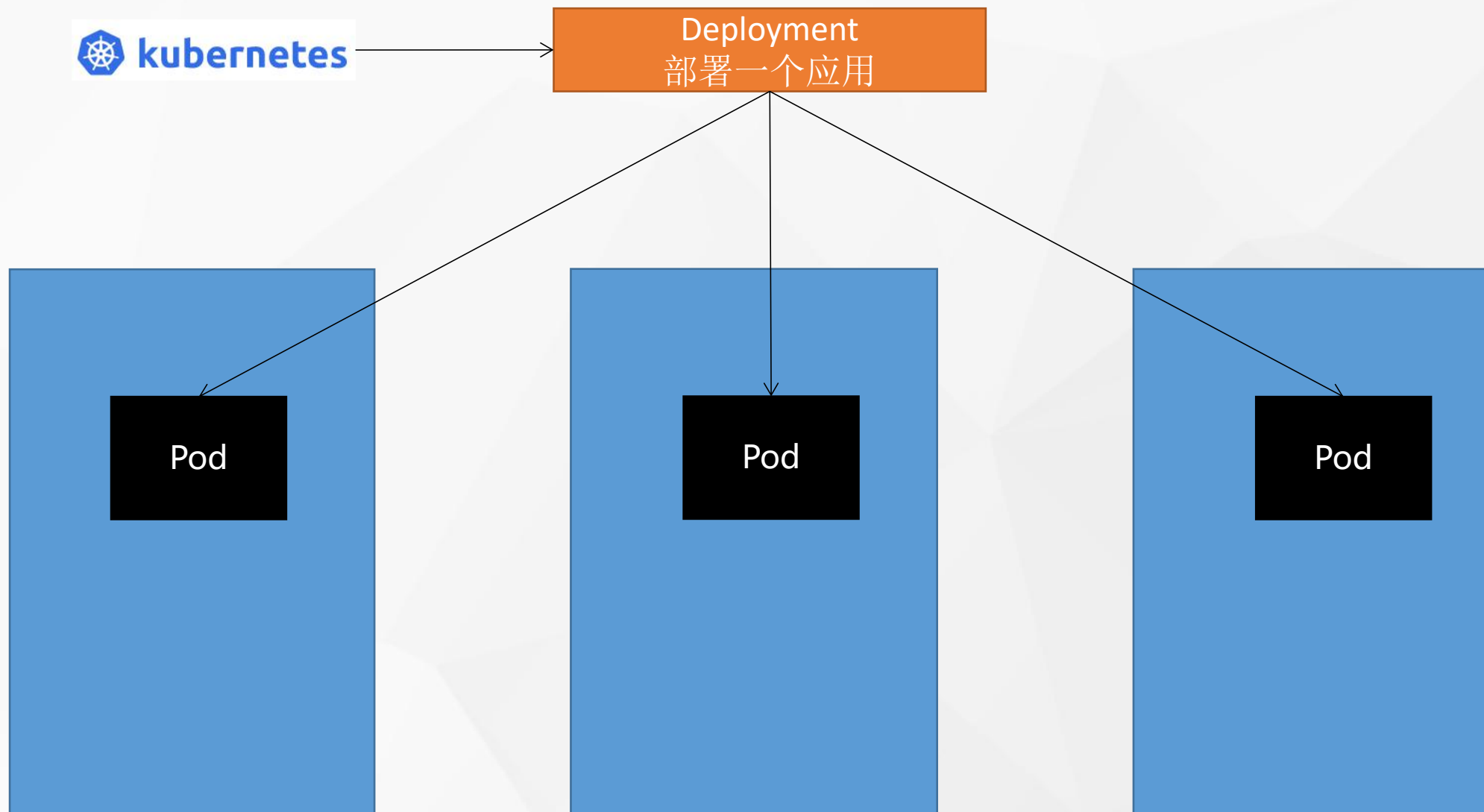
192.168.36.68:8080



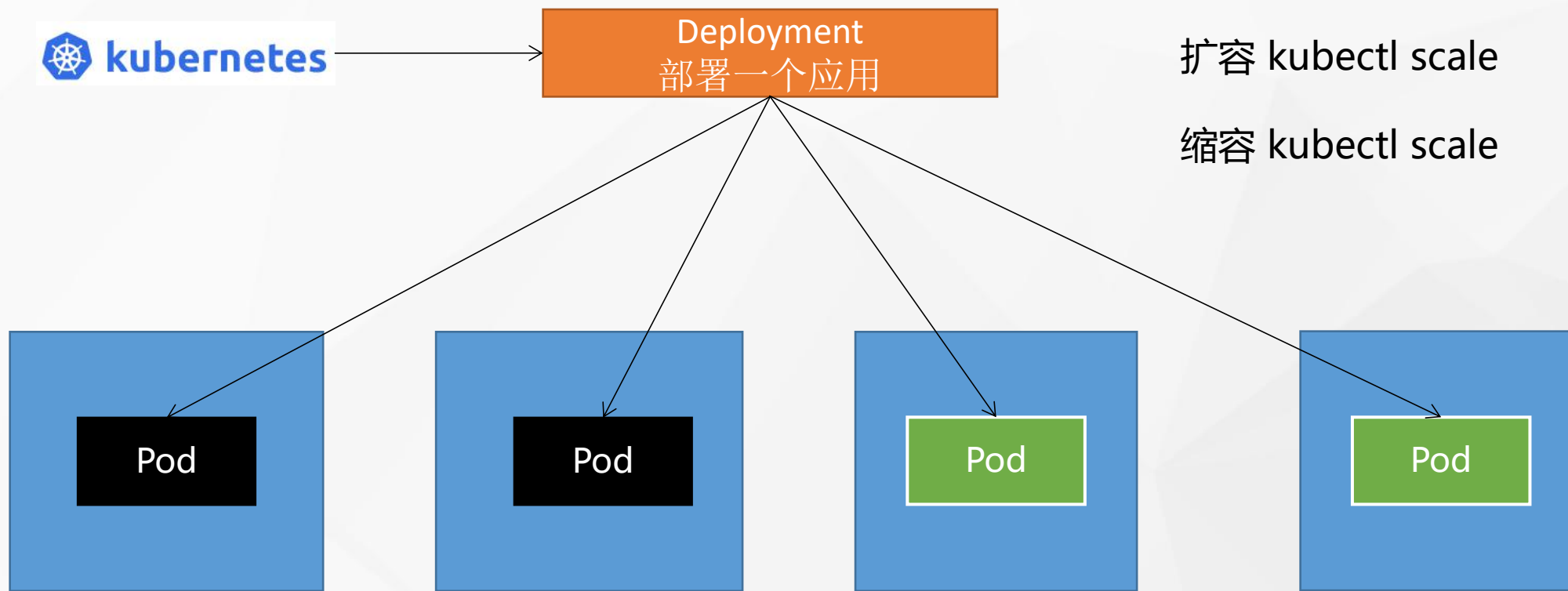
Deployment: 代表一次部署，产生1个或多个Pod



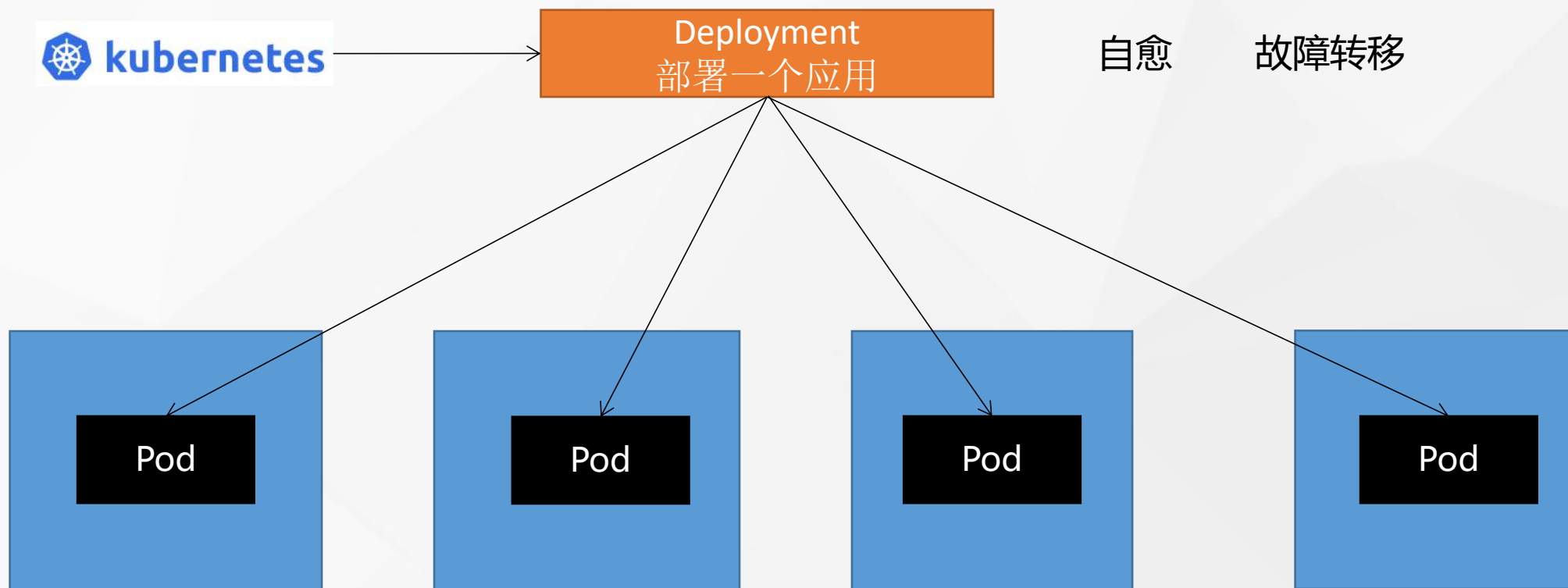
Deployment: 多副本



Deployment: 扩缩容

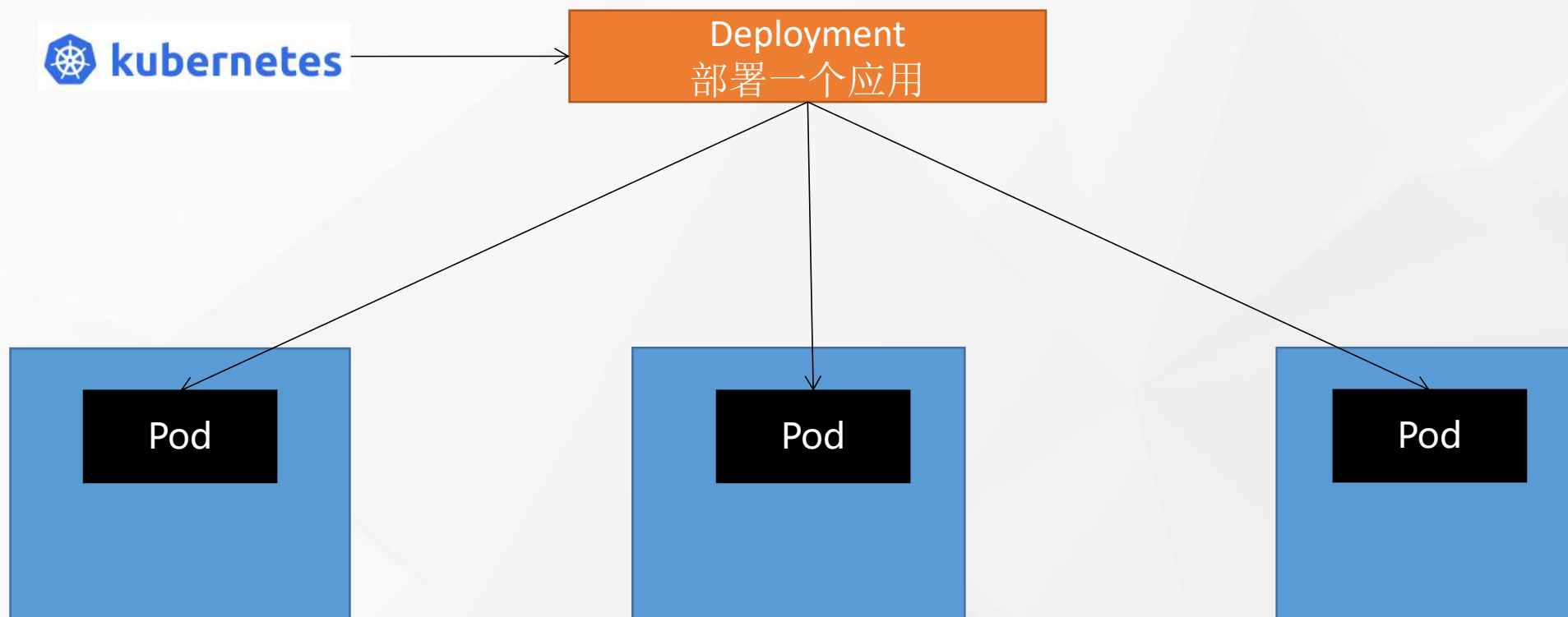


Deployment: 自愈&故障转移

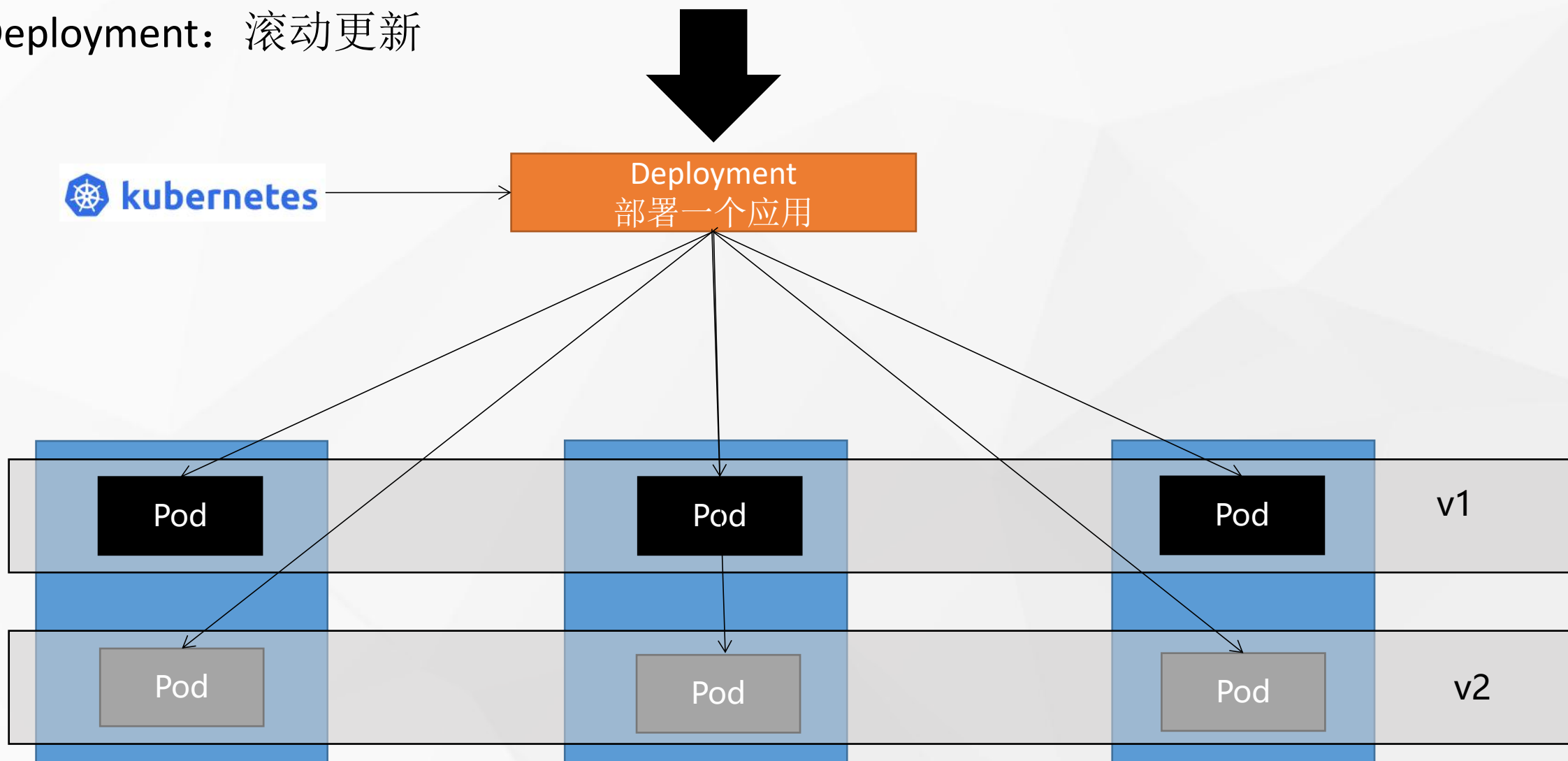


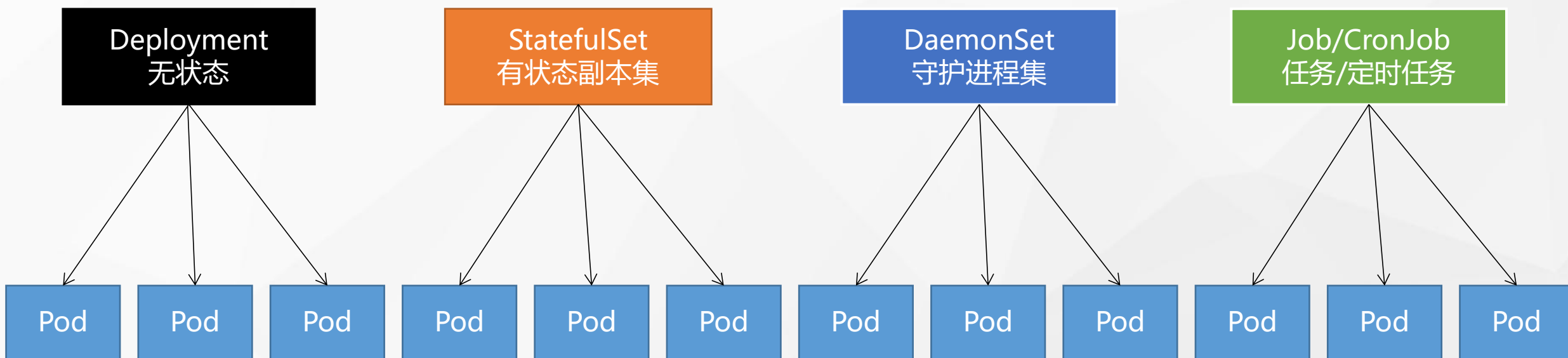


Deployment: 滚动更新



Deployment: 滚动更新





**Deployment:** 无状态应用部署, 比如微服务, 提供多副本等功能

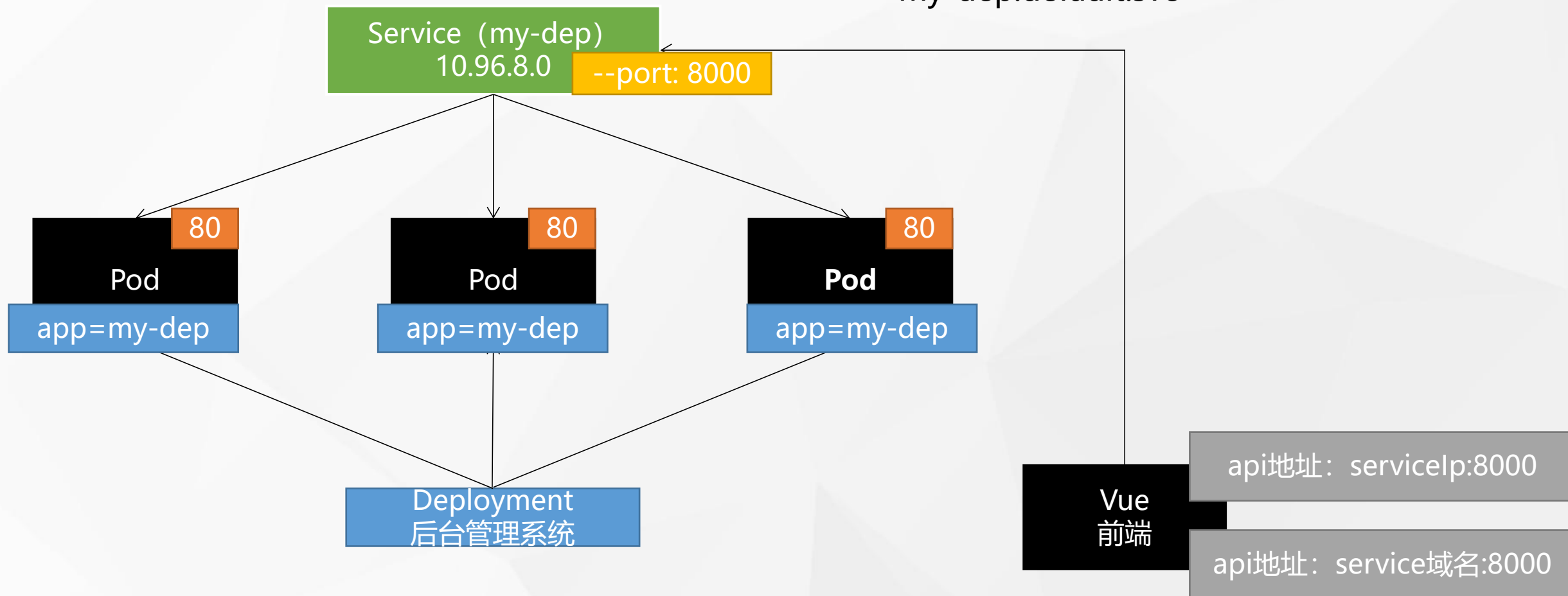
**StatefulSet:** 有状态应用部署, 比如redis, 提供稳定的存储、网络等功能

**DaemonSet:** 守护型应用部署, 比如日志收集组件, 在每个机器都运行一份

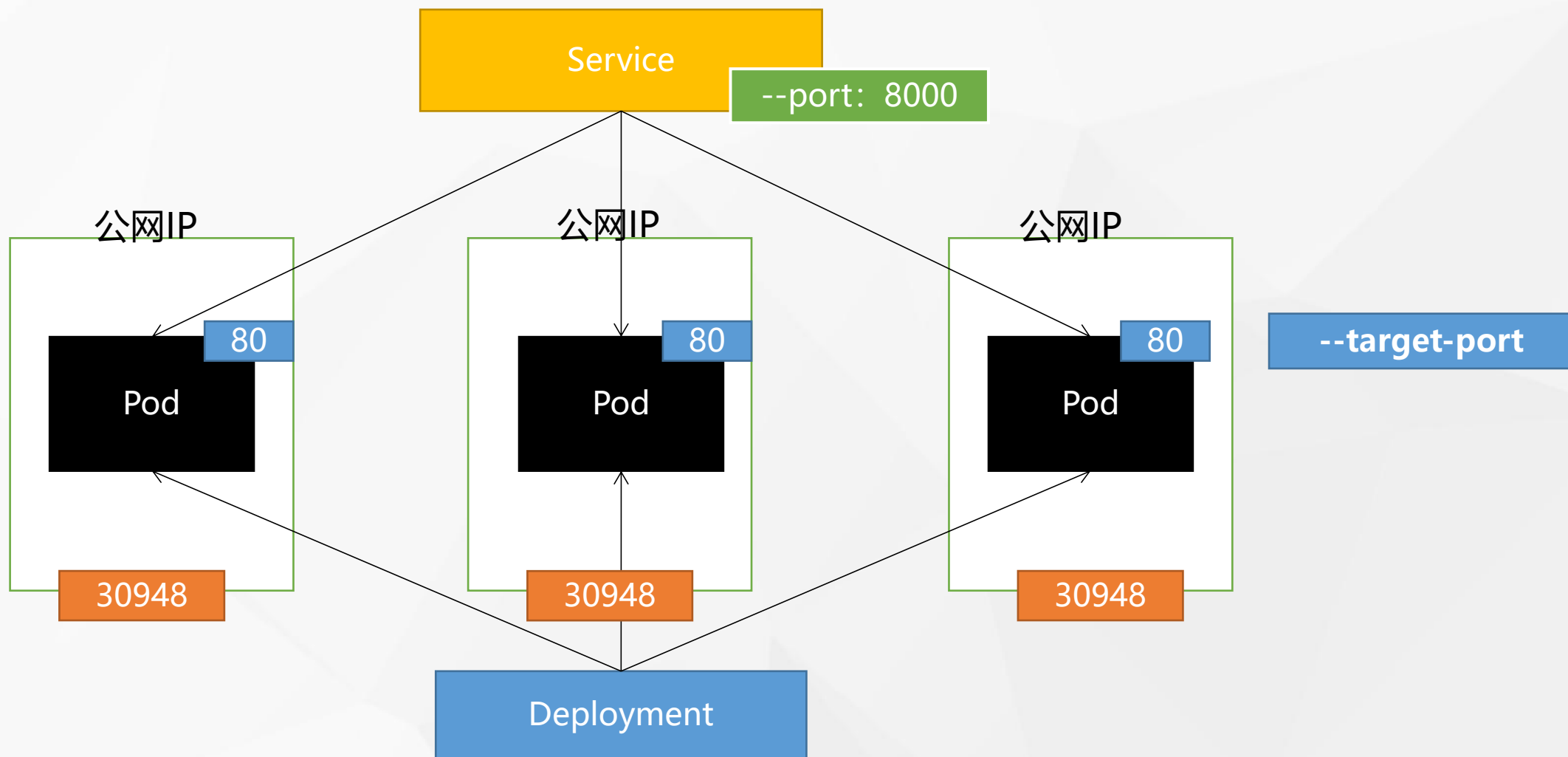
**Job/CronJob:** 定时任务部署, 比如垃圾清理组件, 可以在指定时间运行

Service: Pod的服务发现与负载均衡

服务名.所在名称空间.svc  
my-dep.default.svc

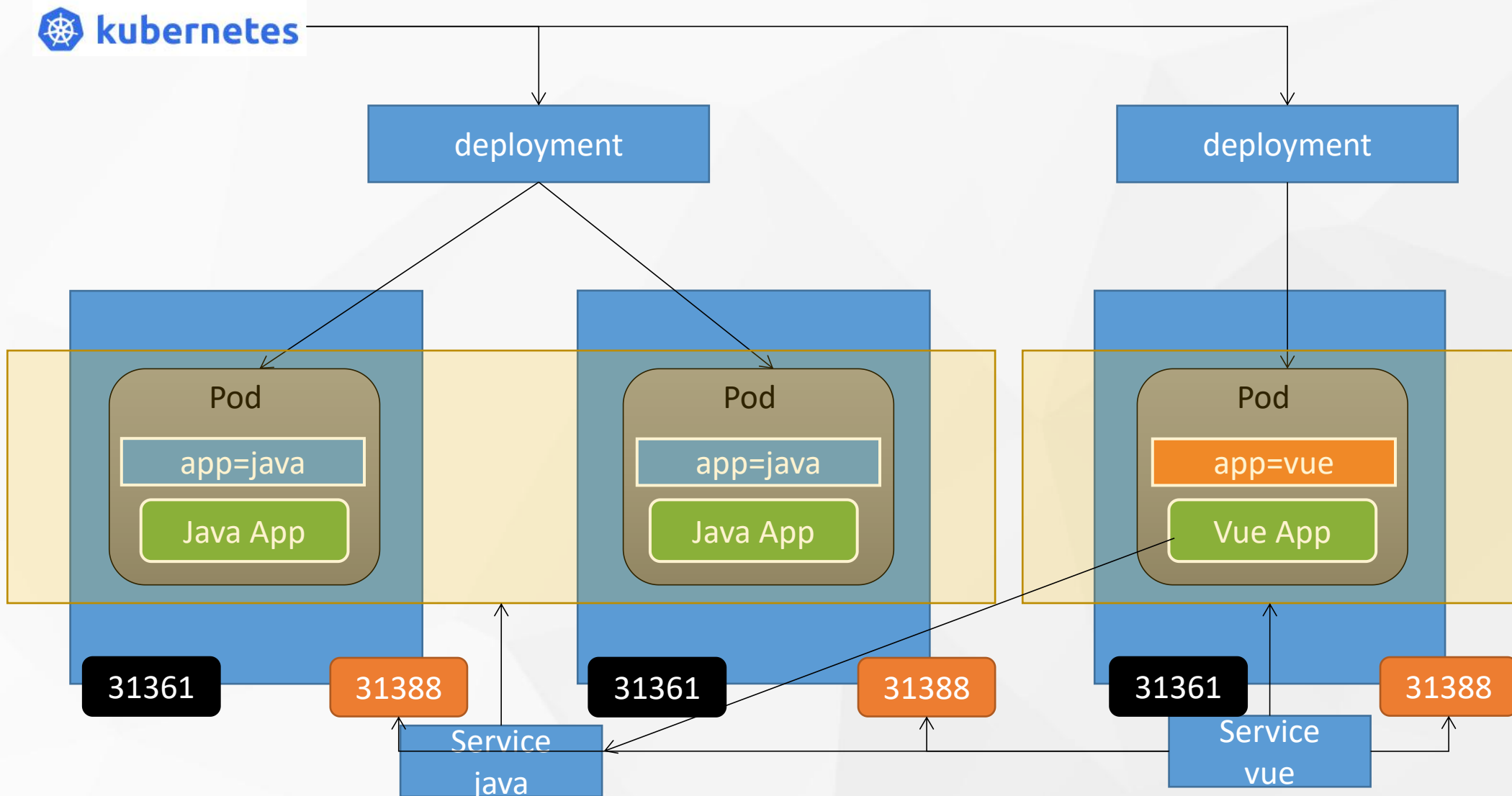


kubectl expose deploy my-dep --port=8000 --target-port=80 --type=ClusterIP  
集群内使用service的ip:port就可以负载均衡的访问



kubectl expose deploy my-dep --port=8000 --target-port=80 --type=ClusterIP: 集群内部的访问  
kubectl expose deploy my-dep --port=8000 --target-port=80 --type=NodePort: 集群外也可以访问

Service: Pod的服务发现与负载均衡



# Kubernetes-核心概念-Ingress

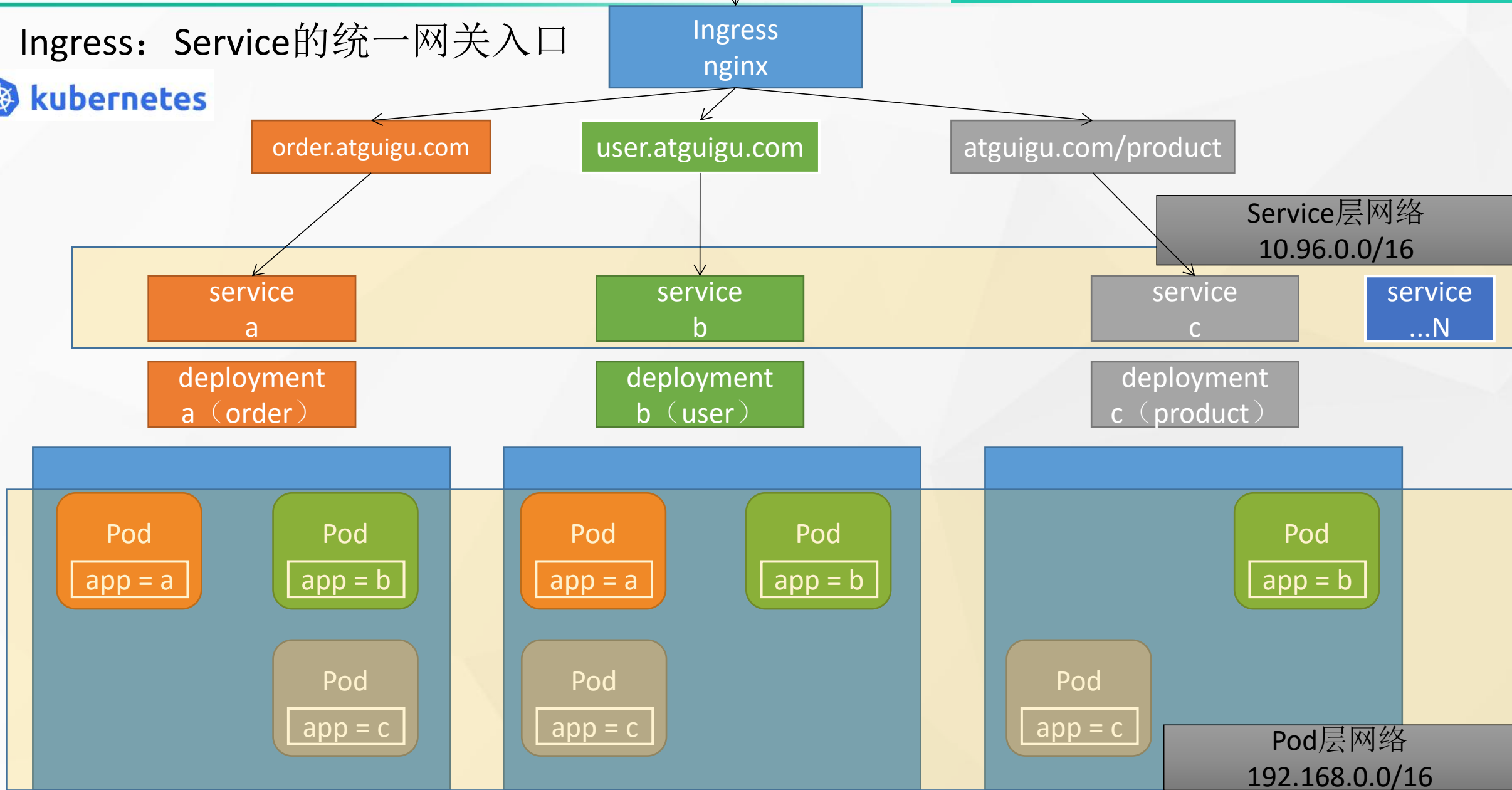
<http://xx.atguigu.com>



KUBESPHERE

尚硅谷

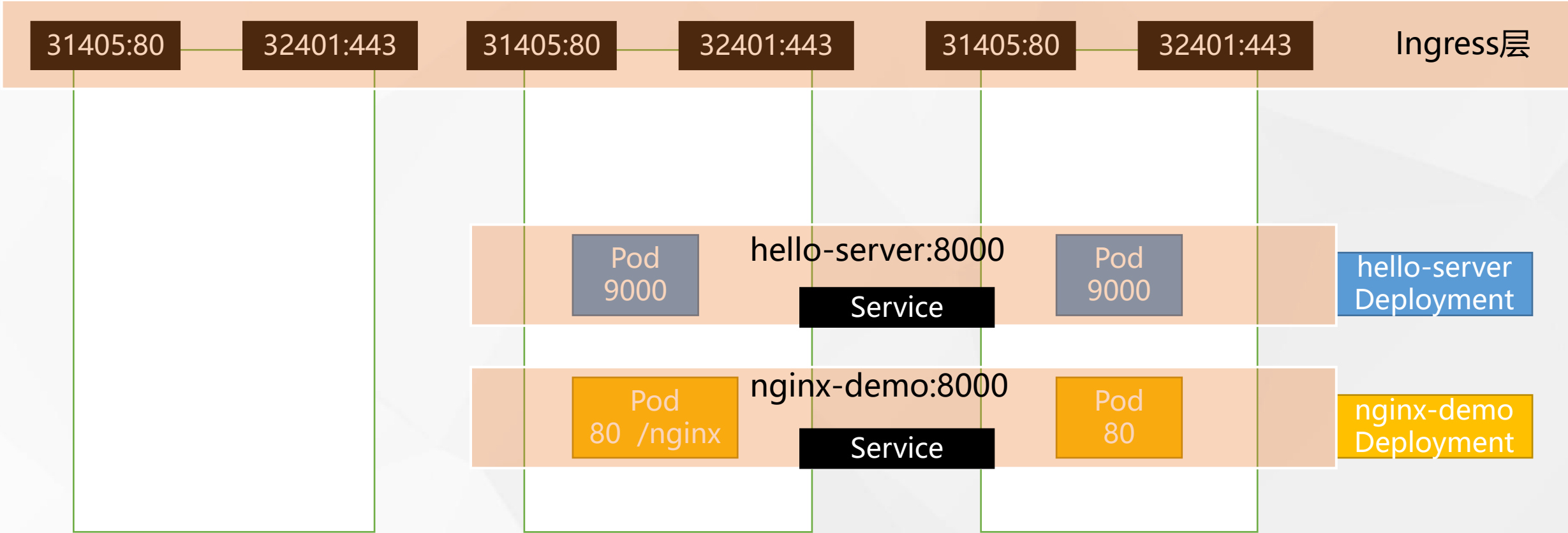
Ingress: Service的统一网关入口



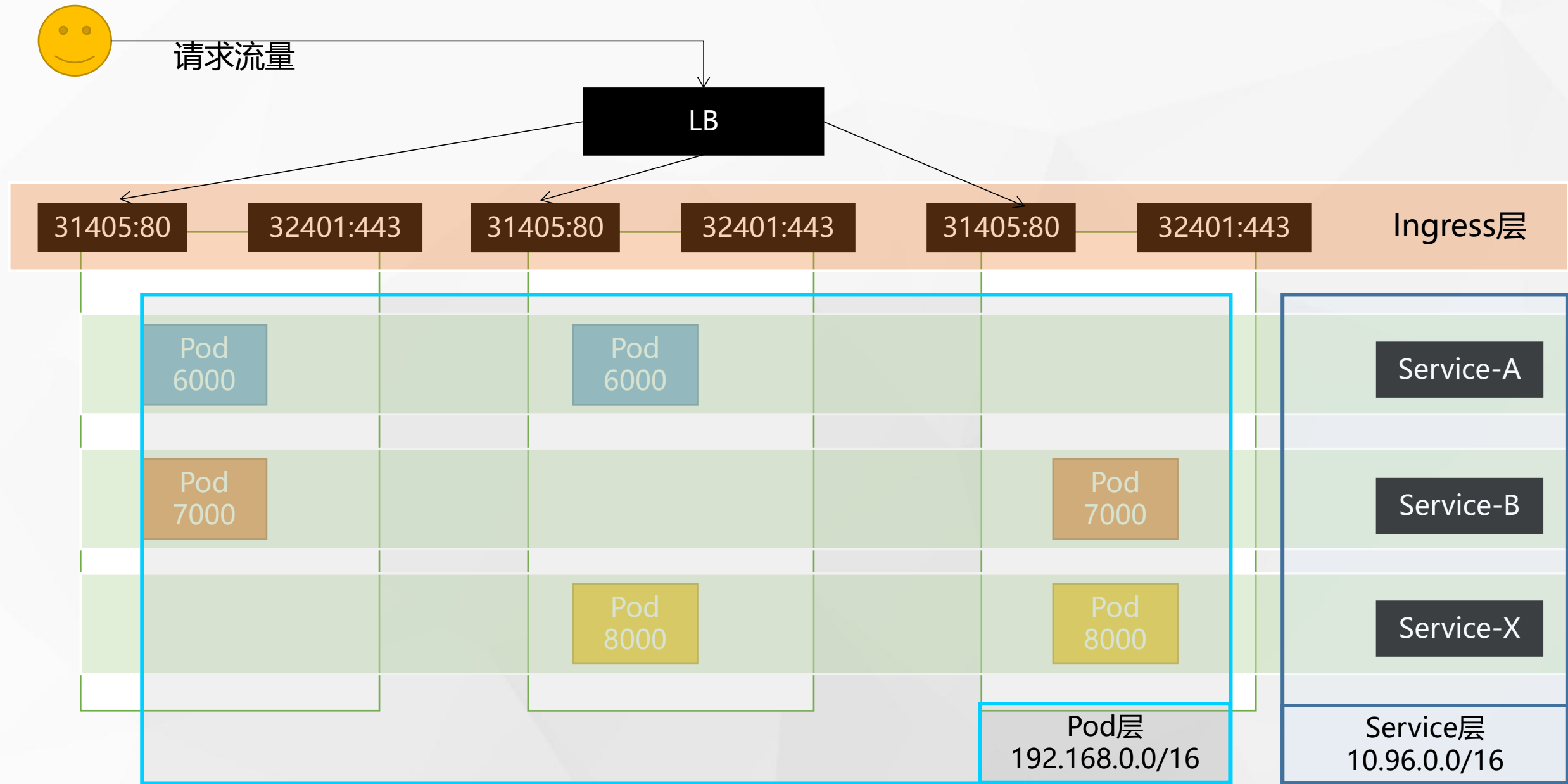


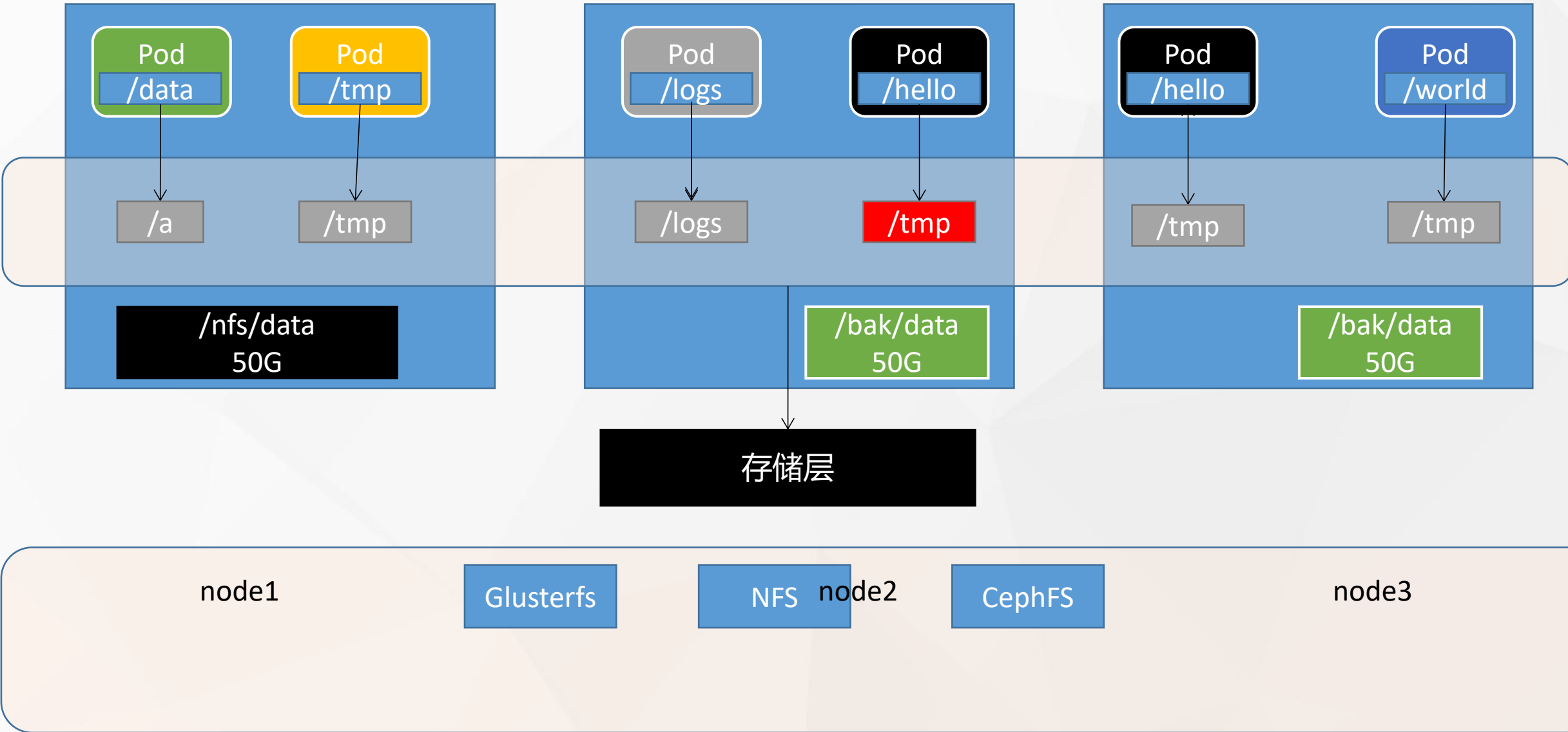
请求流量

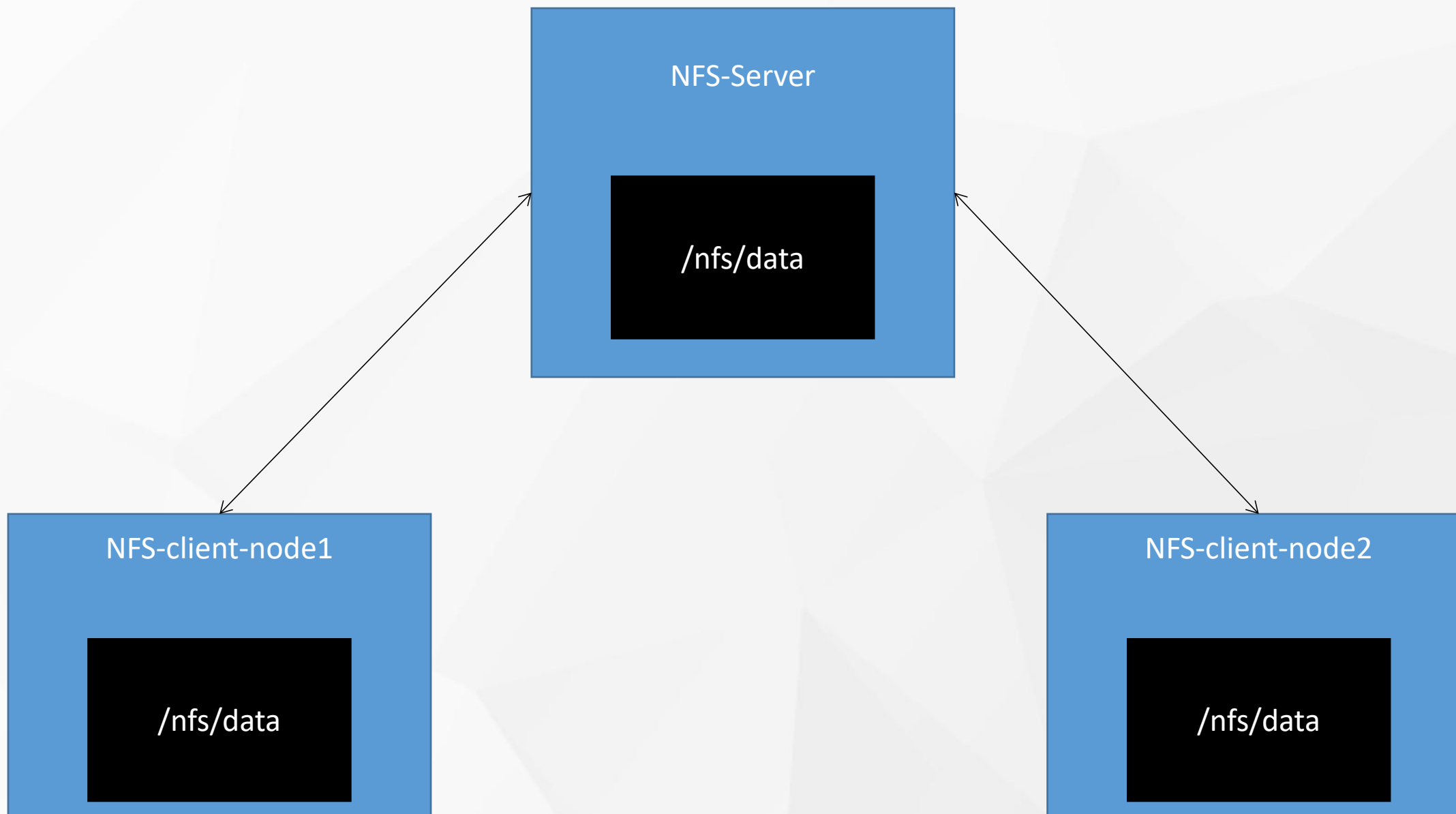
hello.atguigu.com:31405 把请求转给 hello-server进行处理  
demo.atguigu.com:31405 把请求转给nginx-demo进行处理  
demo.atguigu.com:31405/nginx 把请求转给nginx-demo进行处理

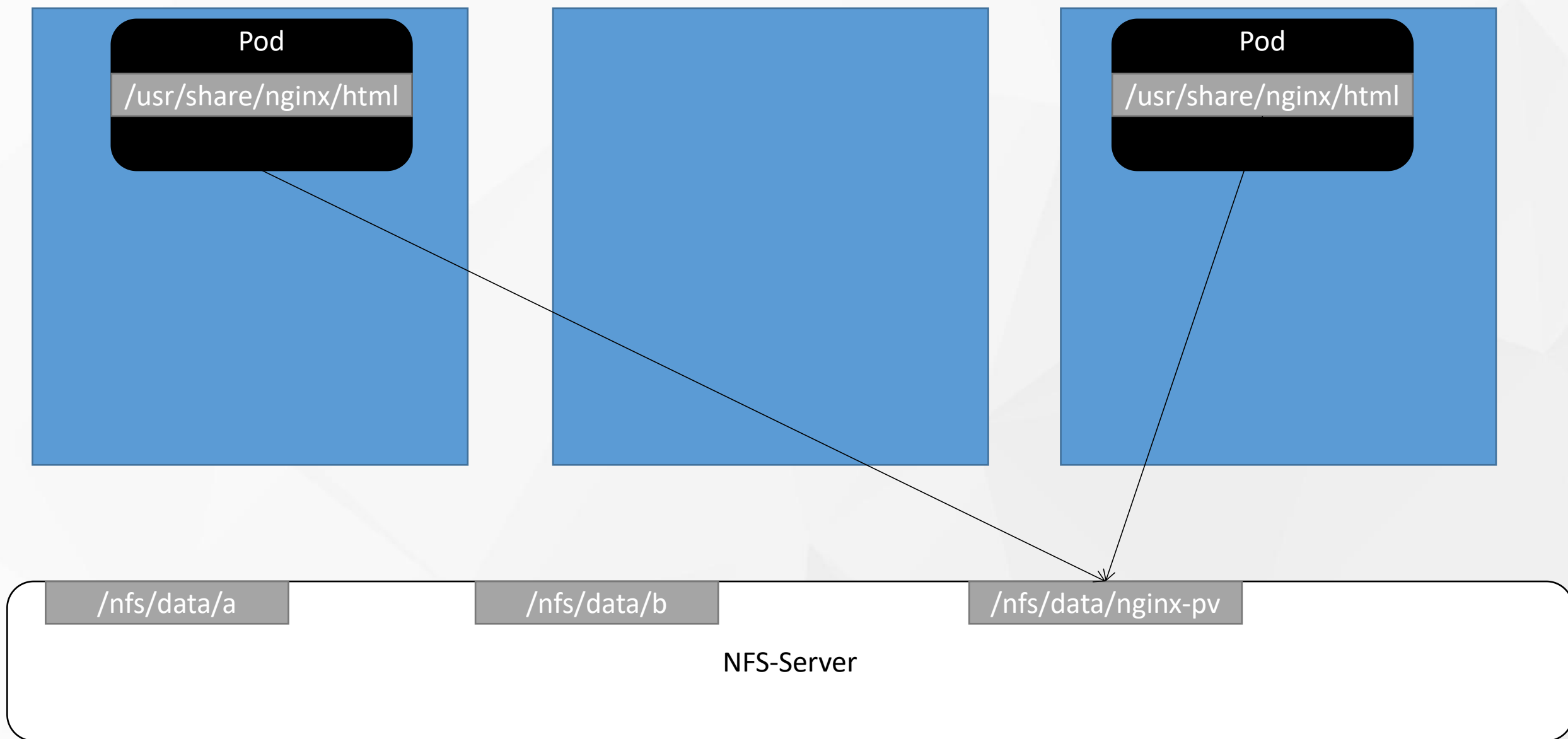






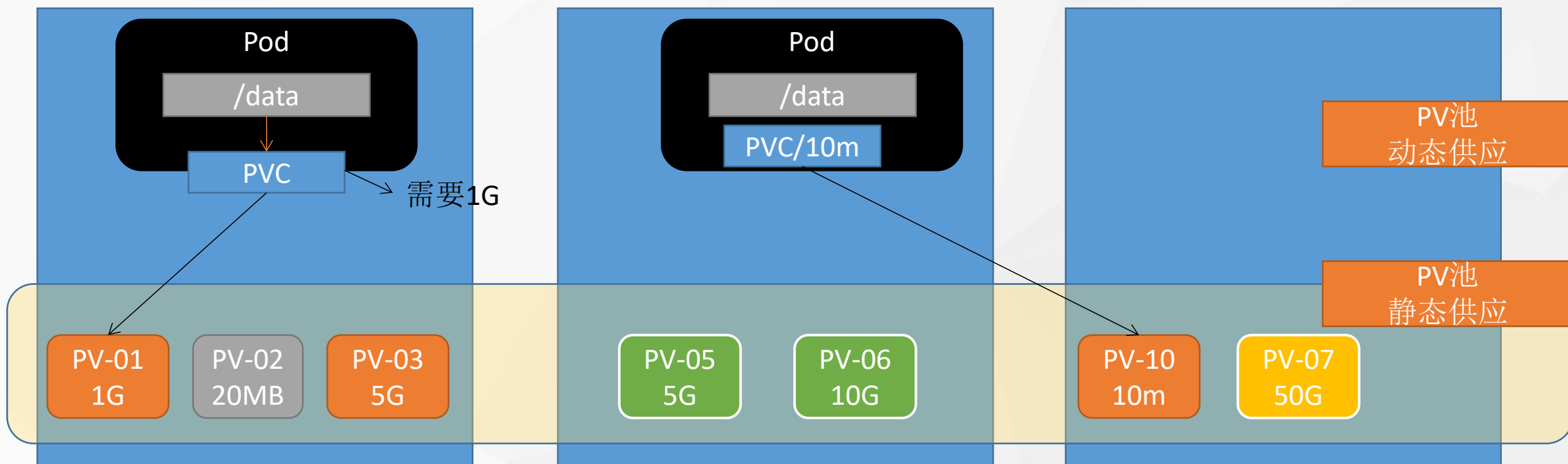




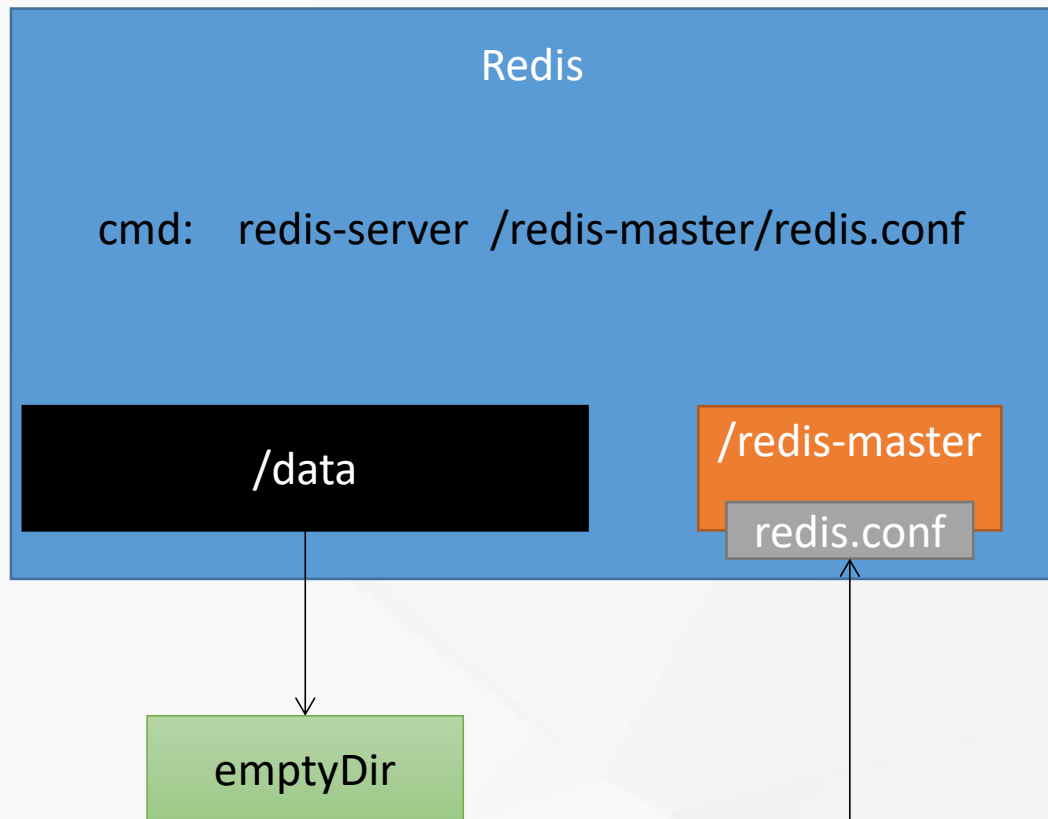


PV: PersistentVolume 持久卷

PVC: PersistentVolumeClaim 持久卷申明



ConfigMap: 配置集。抽取应用的配置



```
apiVersion: v1
data:      #data是所有真正的数据
redis.conf: <
  appendonly yes
kind: ConfigMap
metadata:
  name: redis-conf <
  namespace: default
```

```
14   volumeMounts:
15     - mountPath: /data
16       name: data
17     - mountPath: /redis-master
18       name: config
19   volumes:
20     - name: data
21       emptyDir: {}
22     - name: config
23       configMap:
24         name: redis-conf <
25         items:
26           - key: redis.conf <
27             path: redis.conf <
```

Secret: 密钥，用来保存敏感信息，例如密码、OAuth 令牌和 SSH 密钥。

```
apiVersion: v1
kind: Pod
metadata:
  name: private-reg
spec:
  containers:
  - name: private-reg-container
    image: <your-private-image>
  imagePullSecrets:
  - name: regcred
```

# KubeSphere平台实战

安装&使用



KubeSphere 容器平台

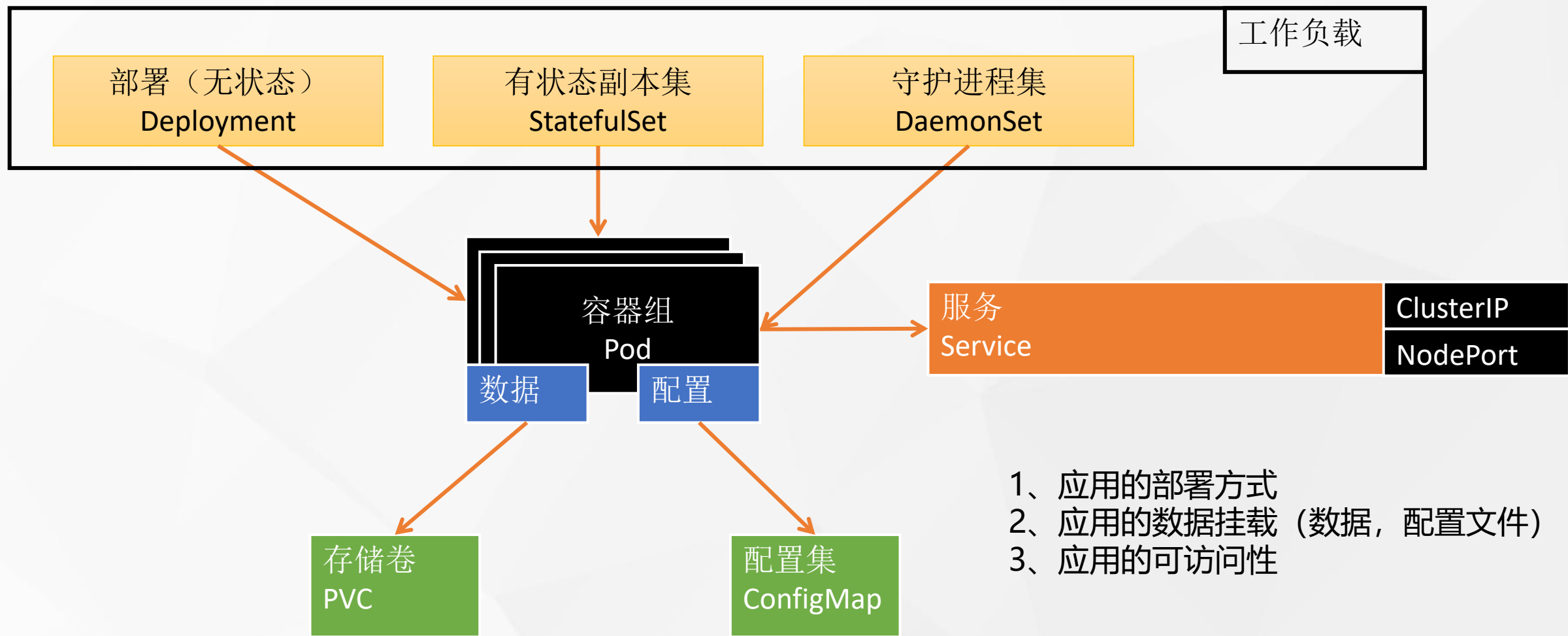
# 面向云原生应用的 容器混合云

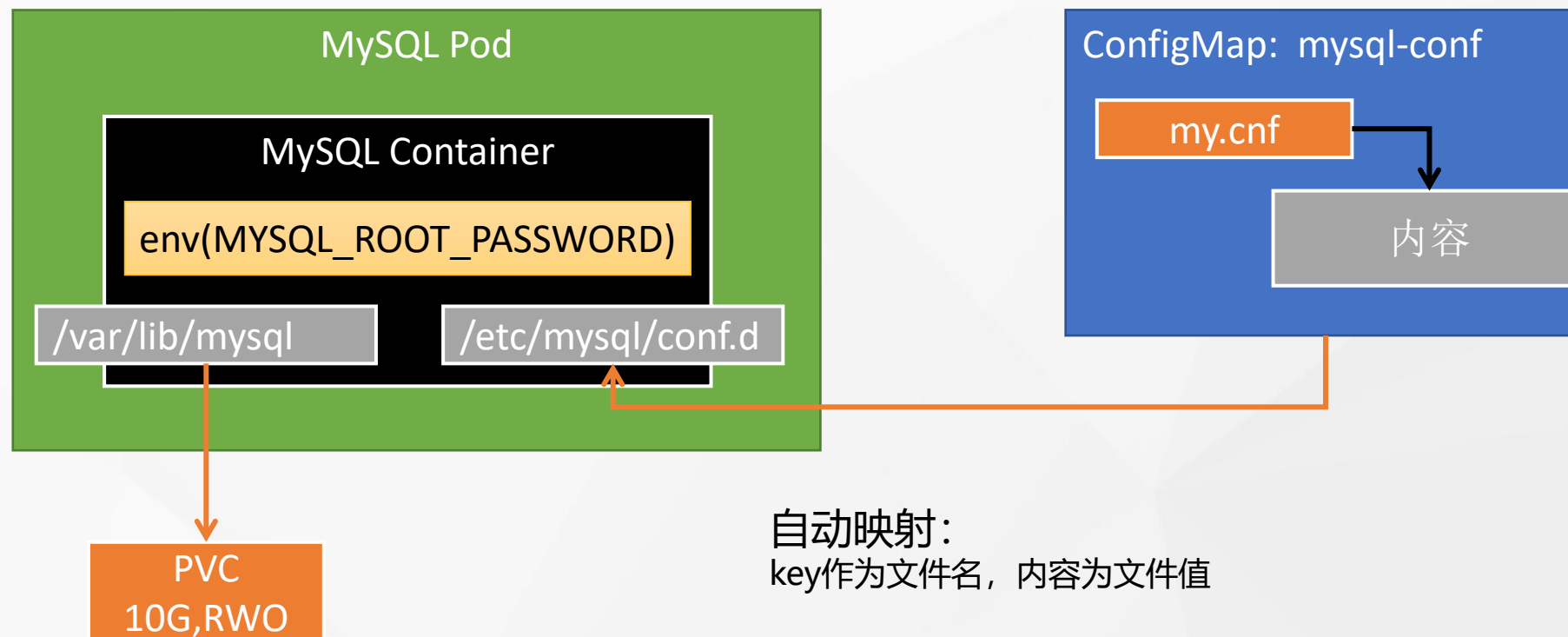
KubeSphere 愿景是打造一个以 Kubernetes 为内核的云原生分布式操作系统，它的架构可以非常方便地使第三方应用与云原生生态组件进行即插即用（plug-and-play）的集成，支持云原生应用在多云与多集群的统一分发和运维管理。

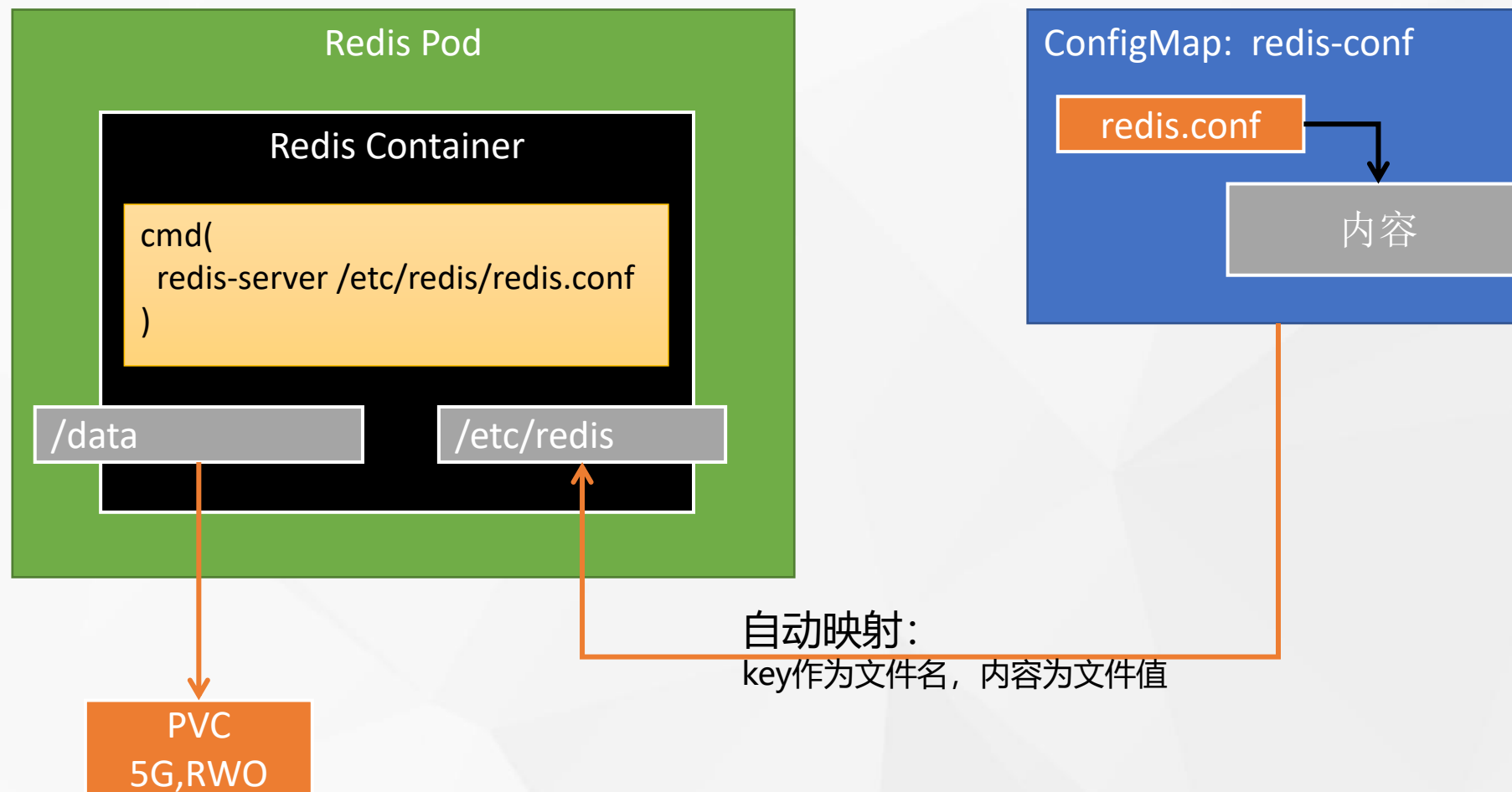
<https://kubesphere.io/zh/>

# KubeSphere-中间件部署实战

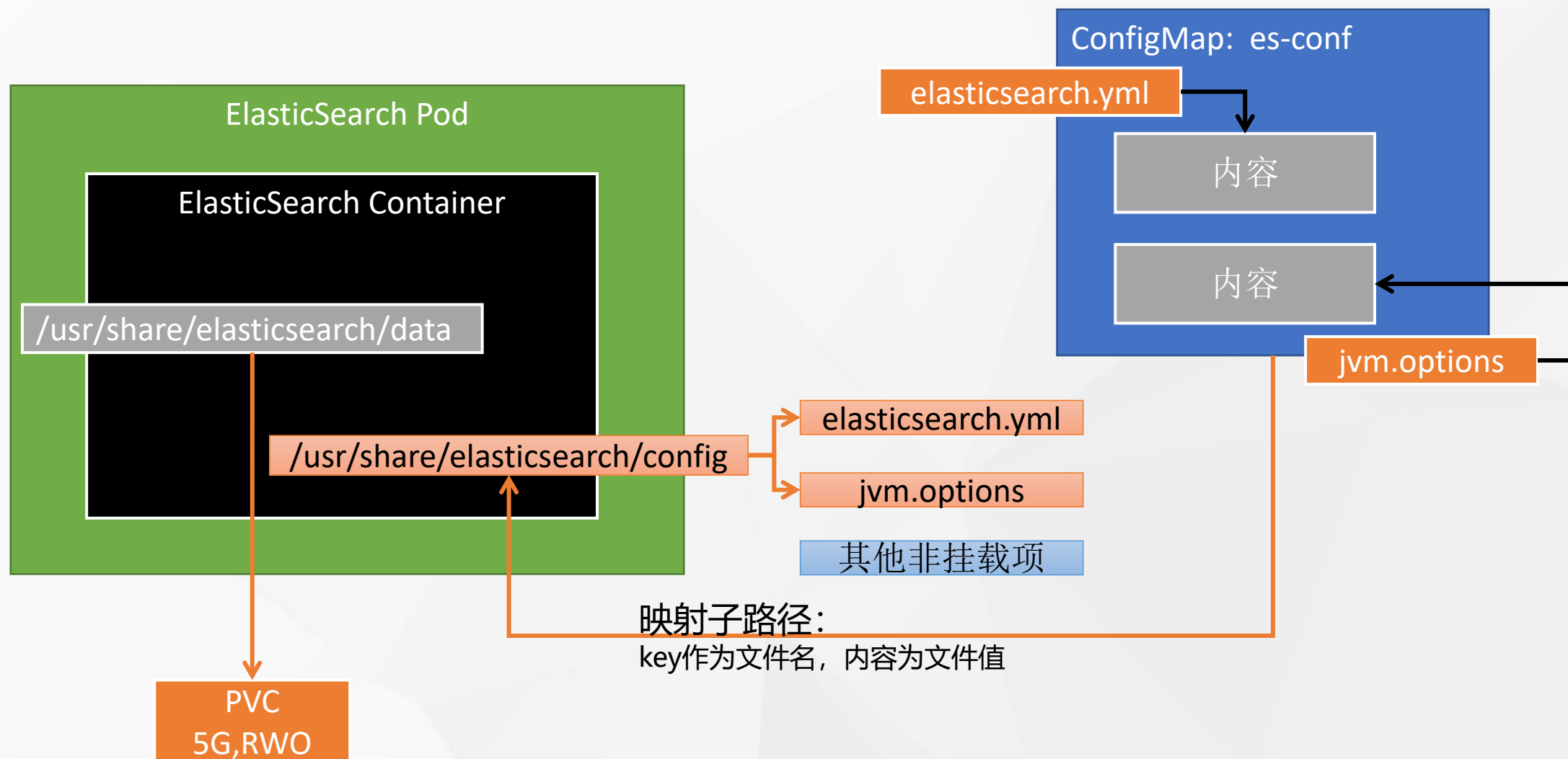
MySQL、Redis、ElasticSearch

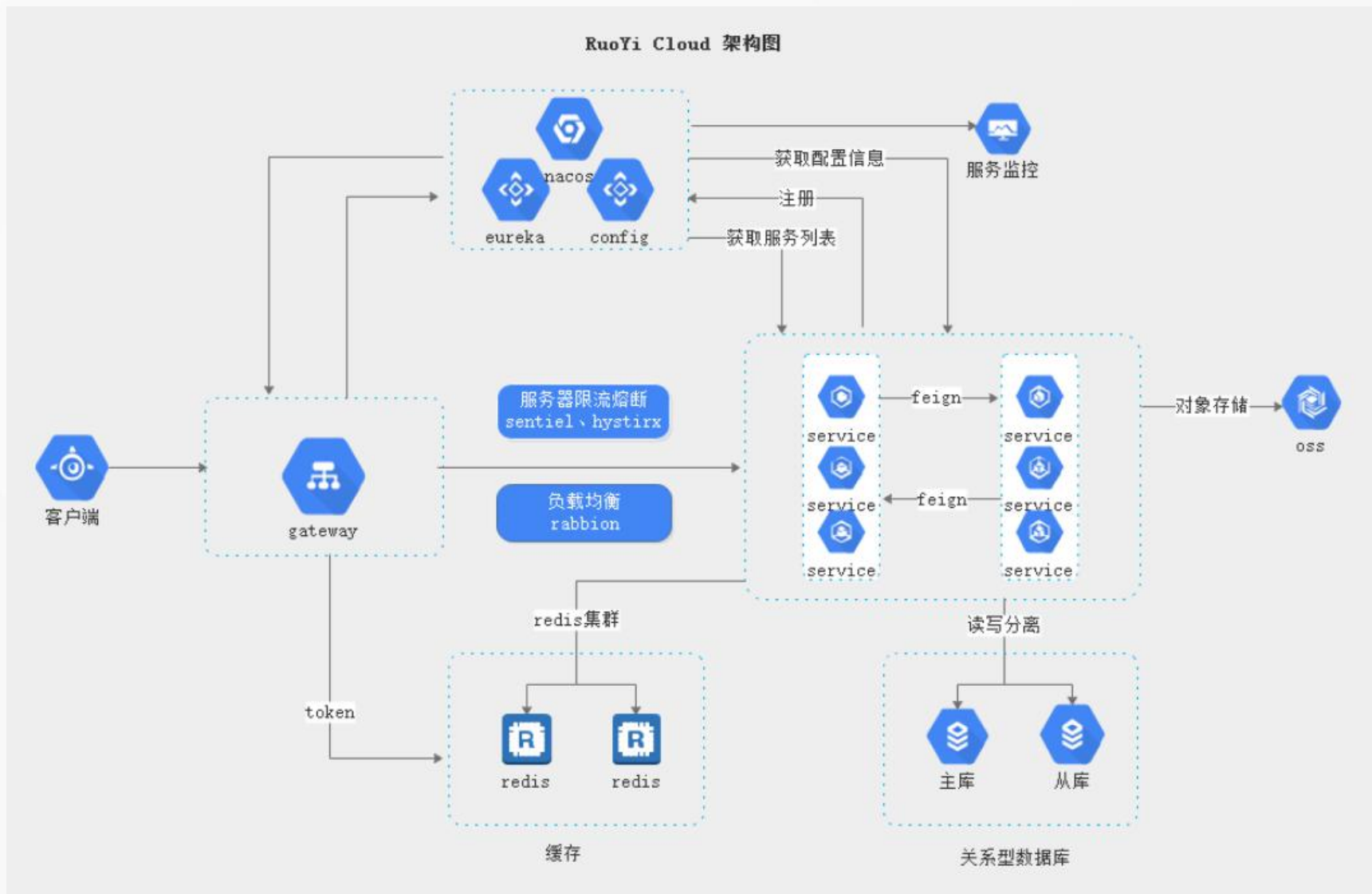






# KubeSphere-部署ElasticSearch





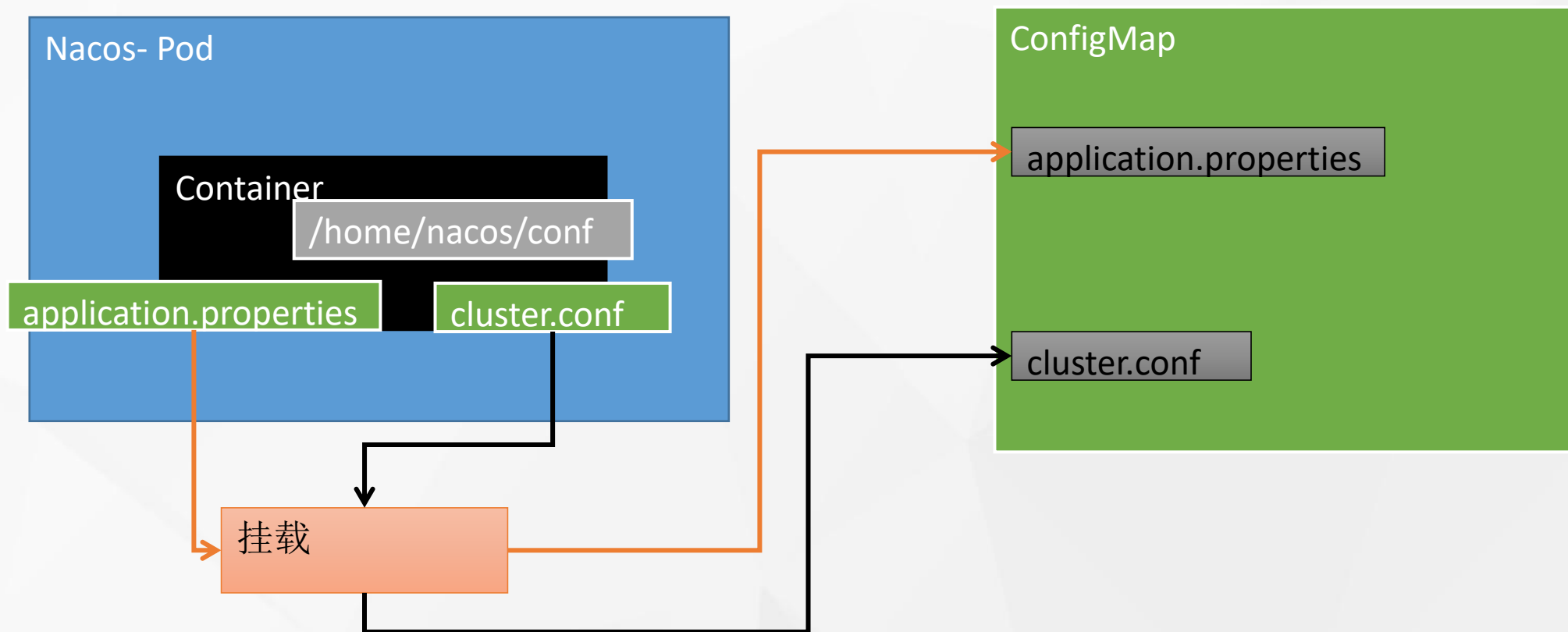
中间件  
有状态、数据导入

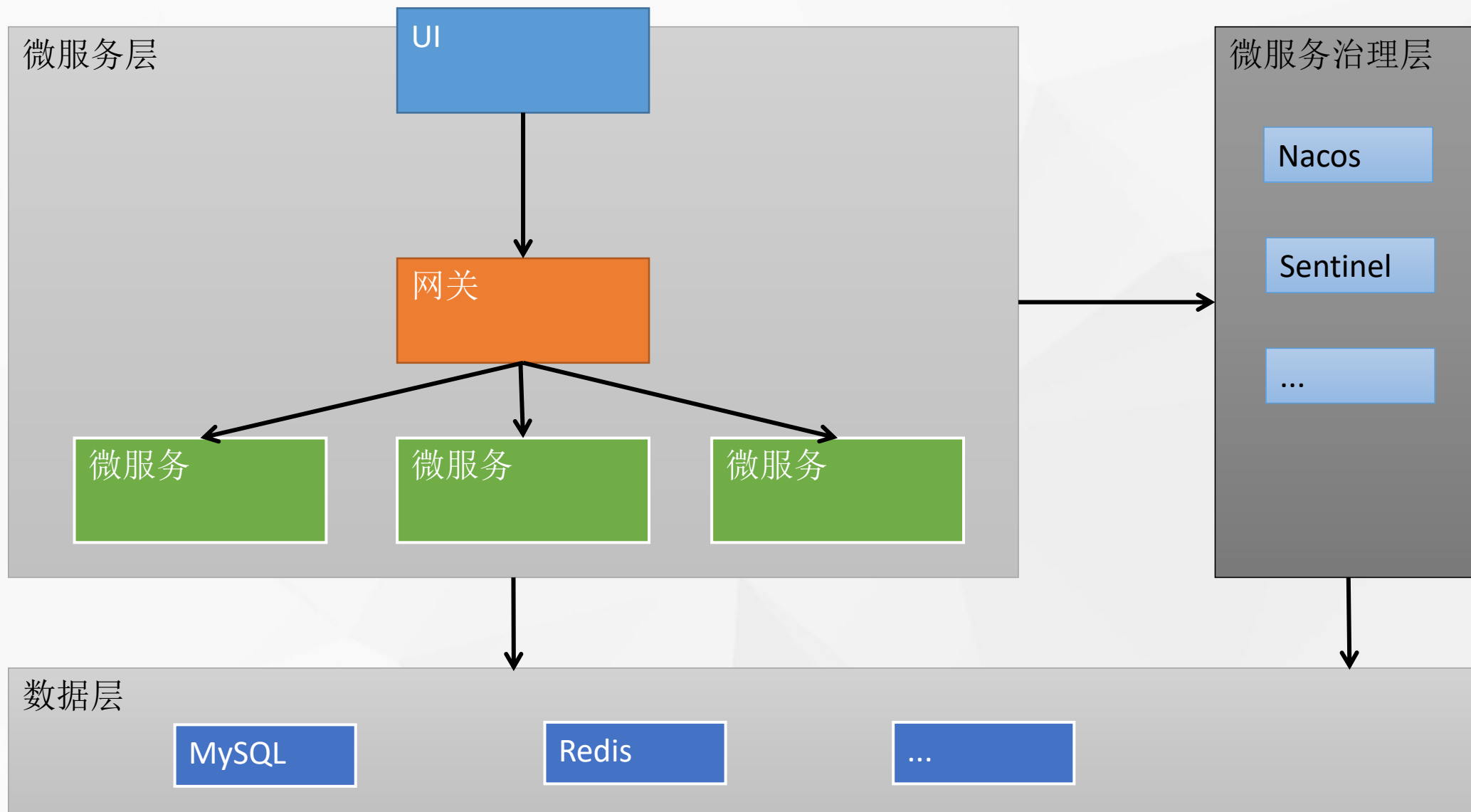
微服务  
无状态、制作镜像

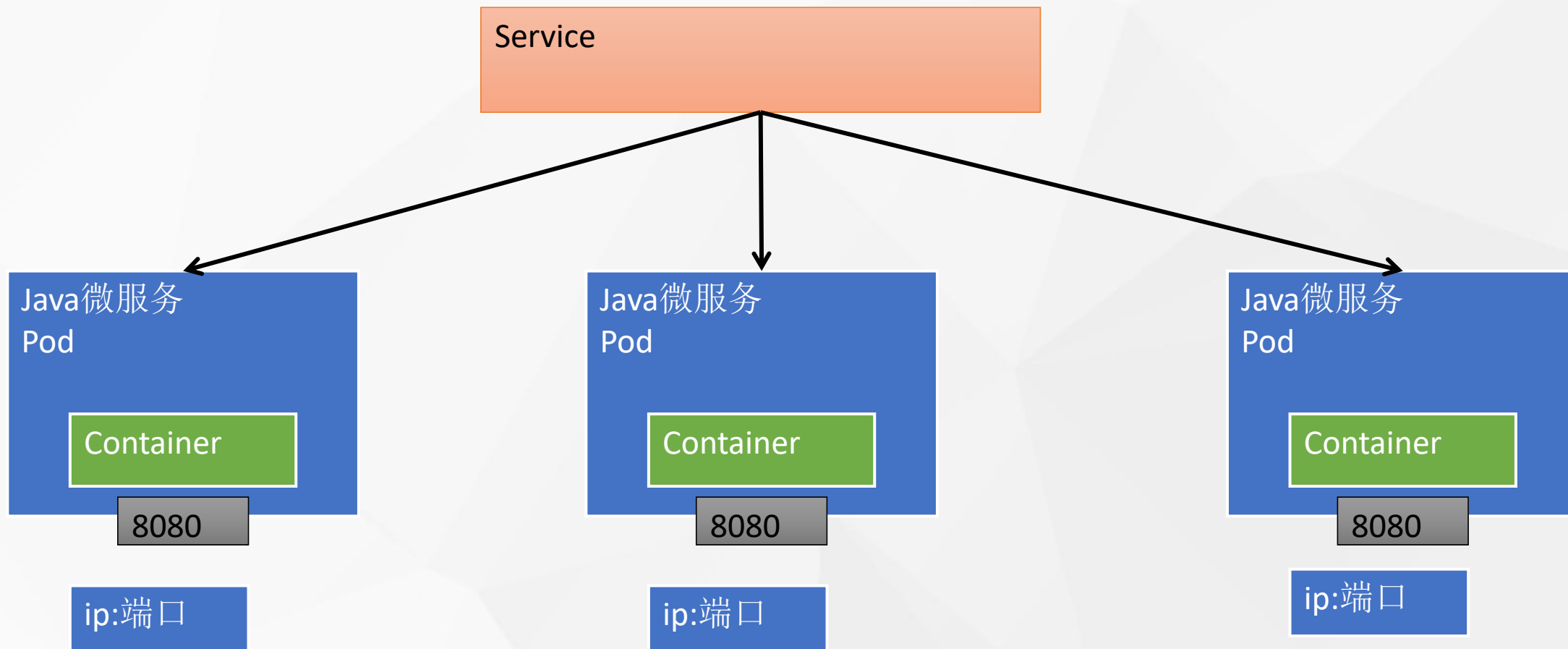
网络  
各种访问地址

配置  
生产配置分离、URL









打包：  
maven打成可  
执行jar，上传  
给服务器  
(master)

制作镜像：  
docker根据  
Dockerfile把包打  
成指定的镜像

推送镜像：  
将镜像推送给  
docker hub（阿  
里云镜像仓库）。

应用部署：  
给k8s部署应用，  
node1节点部署应  
用

自动化制作镜像  
流程

Harbor

```
+ ls hospital-manage/  
deploy  
Dockerfile  
pom.xml  
src  
target  
资源文件
```



谢谢观看