

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

内容



- 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

适合人群



- 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

课程特色



- 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



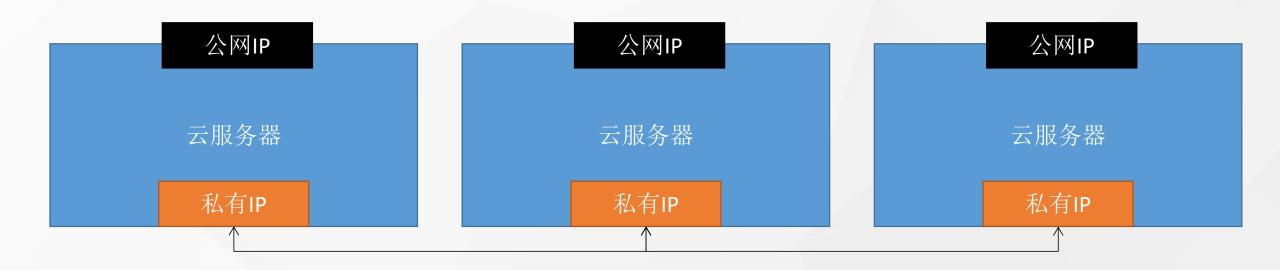
云平台操作

服务器、VPC、安全组...

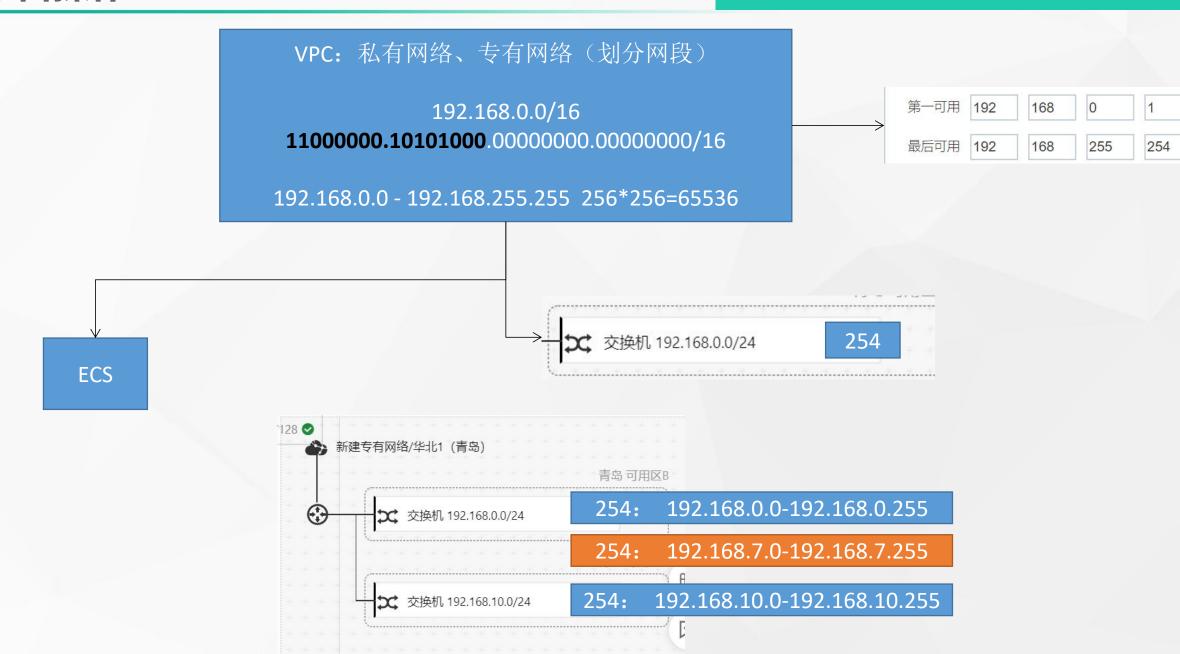
云平台操作



安全组: 防火墙相关的端口设置







云平台操作



VPC: 私有网络、专有网络(划分网段)

192.168.0.0/16 **11000000.10101000**.000000000.00000000/16

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**.000000000.00000000/16

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**.000000000.00000000/16

192.168.0.0 - 192.168.255.255 256*256=65536





Docker容器化实战

基础概念

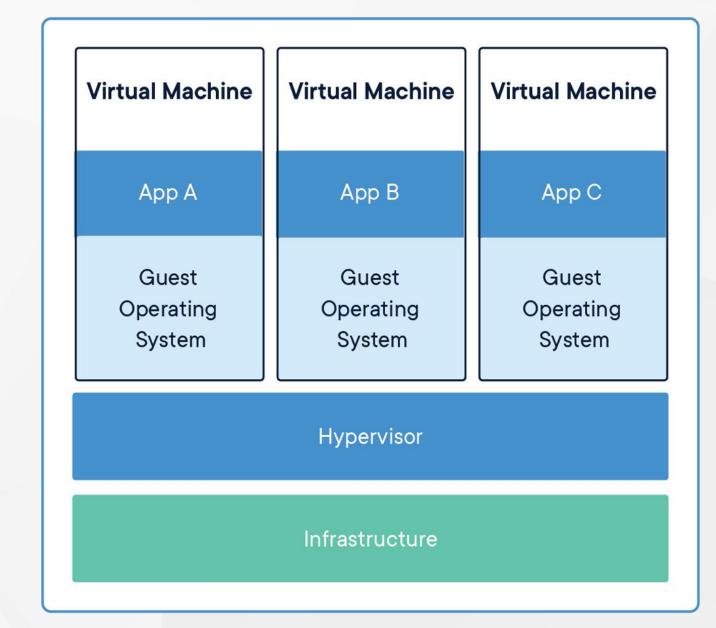
Docker基础



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

• • •



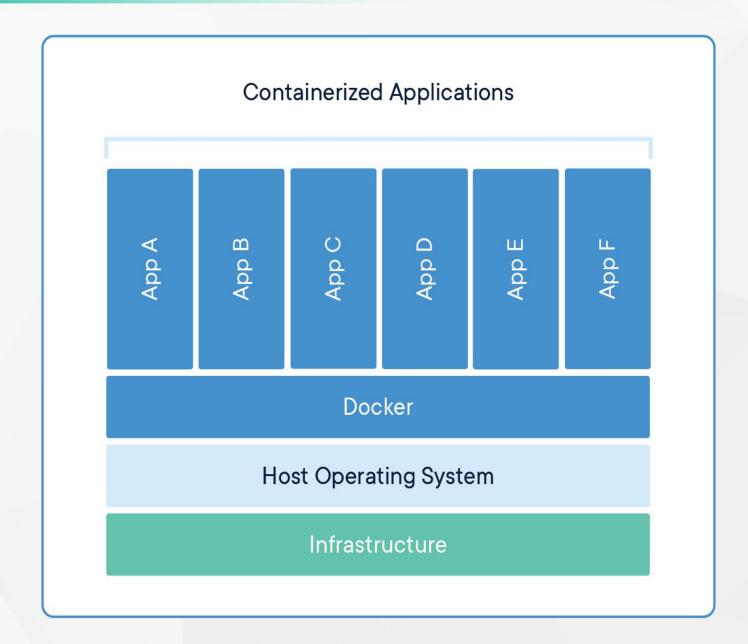
Docker基础



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

• • •



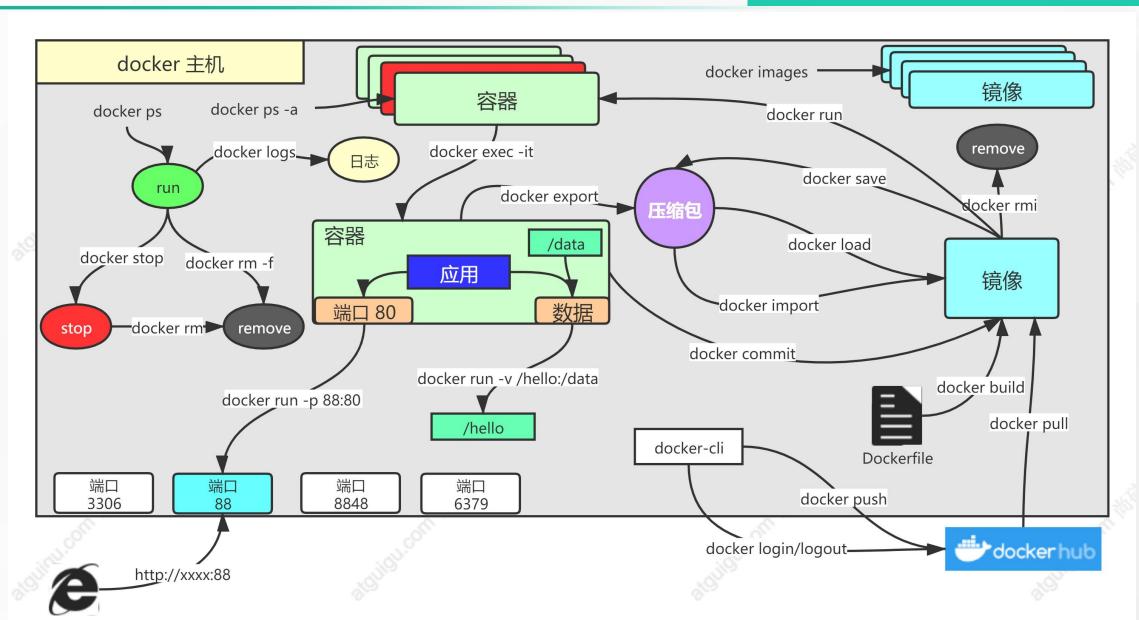


Docker容器化实战

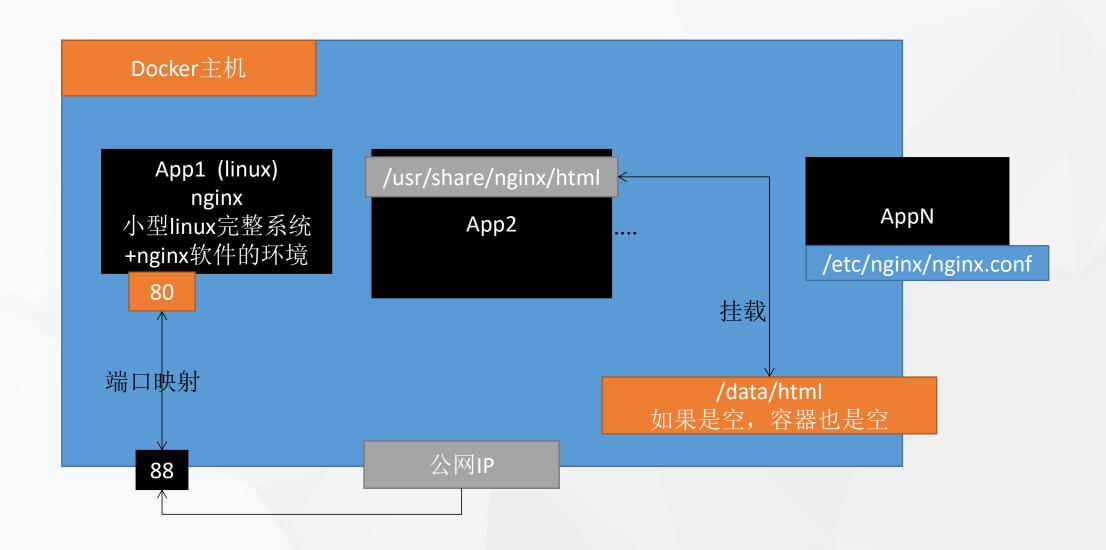
常用命令

Docker常用命令



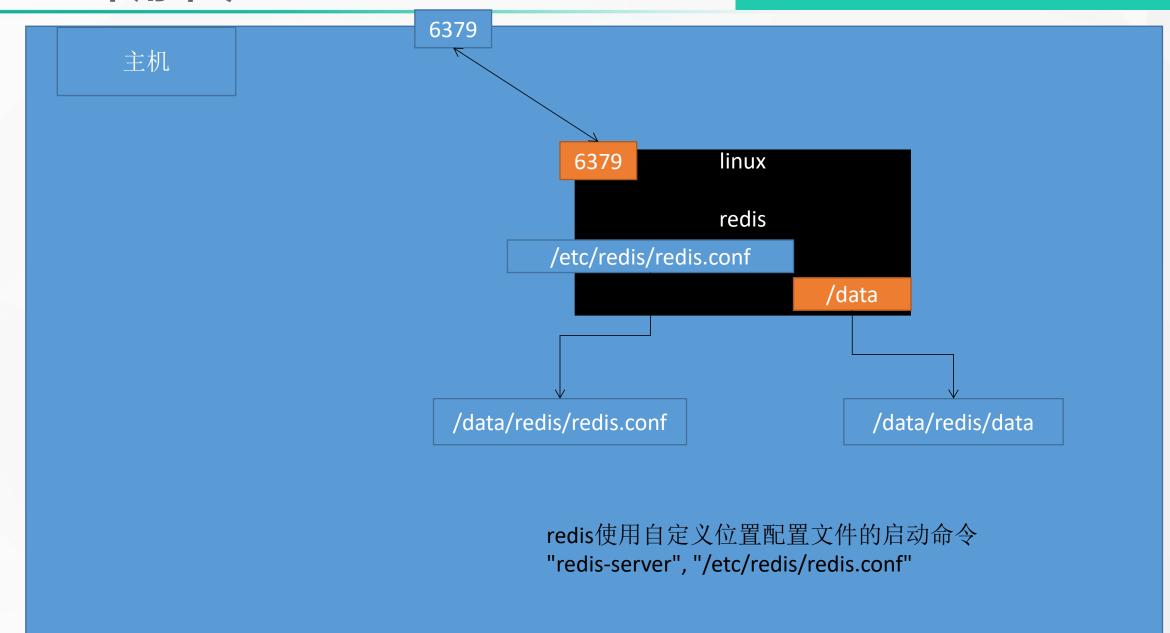






Docker常用命令







完整的linux

Java

/app.jar

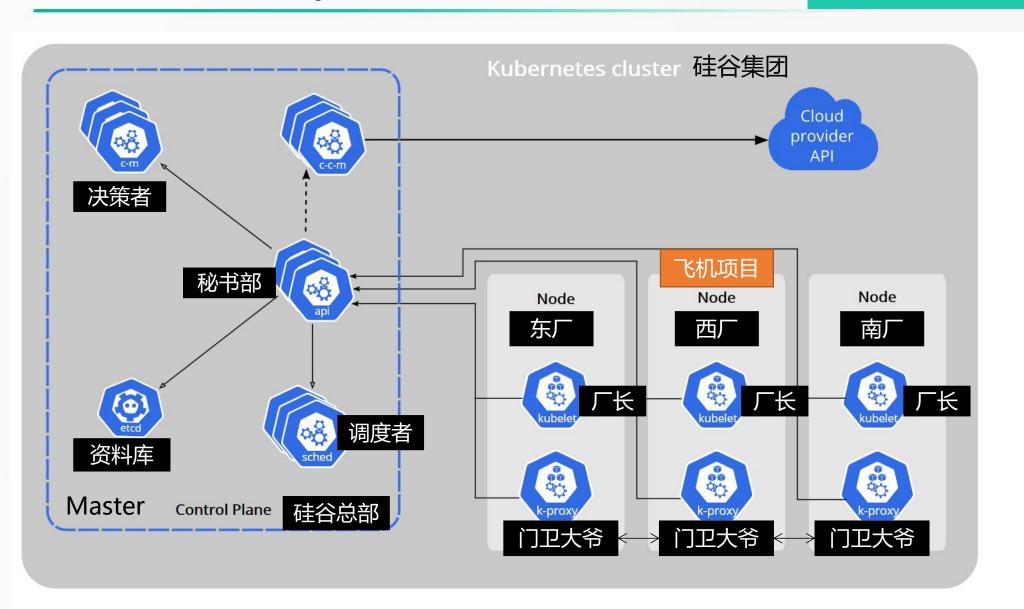


Kubenetes基础

架构

Kubernetes基础

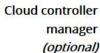




API server

















(persistence store)









kube-proxy











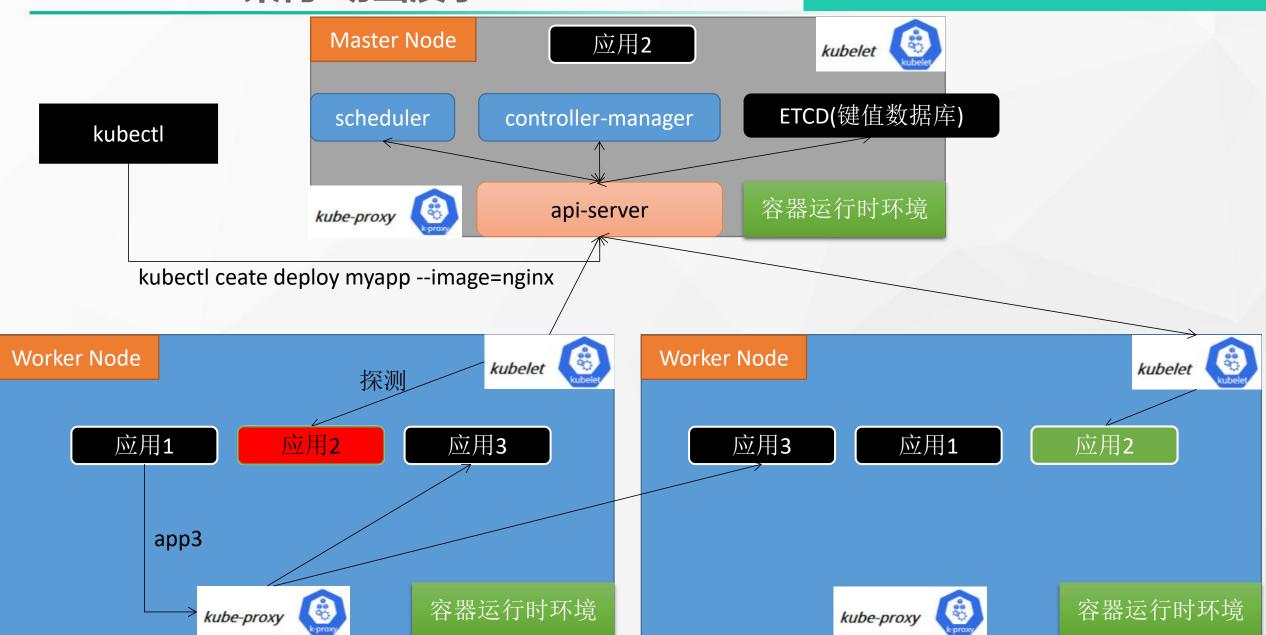






Kubernetes架构-动画演示



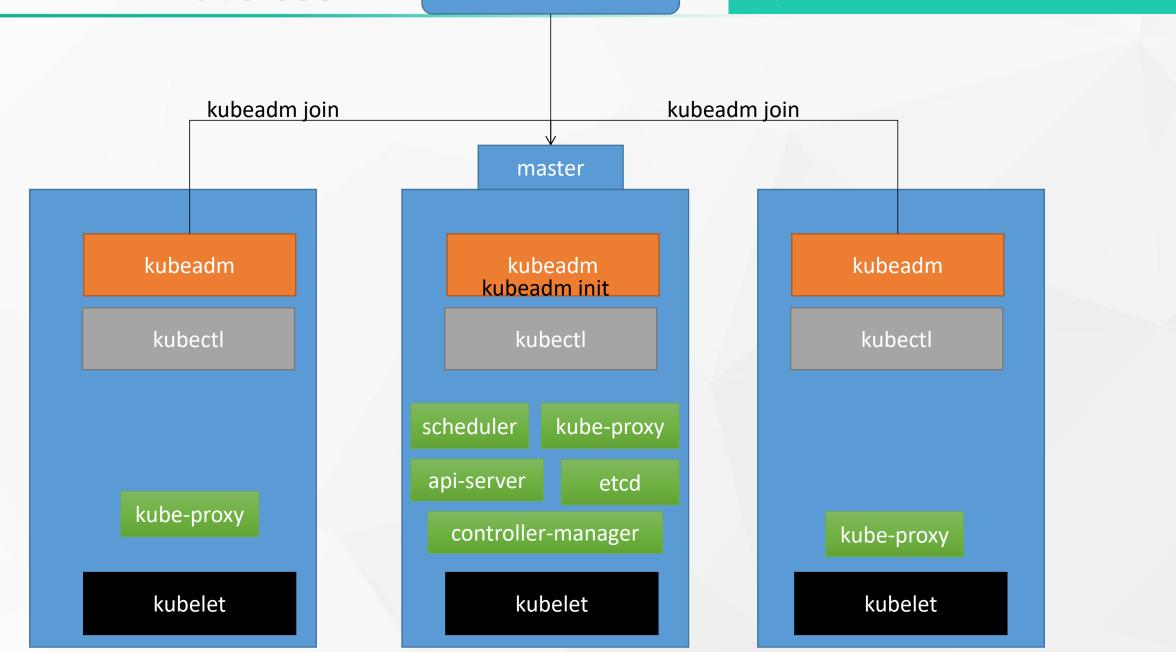




Kubenetes基础

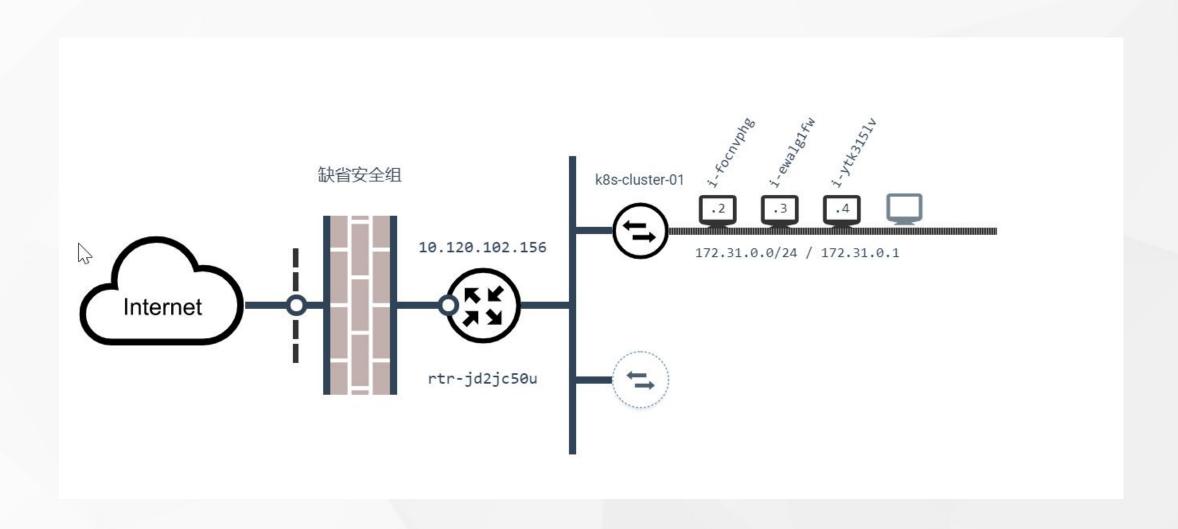
集群安装





Kubernetes-集群部署







Kubenetes核心实战

核心概念理解

Kubernetes-核心概念-Namespace



NameSpace:

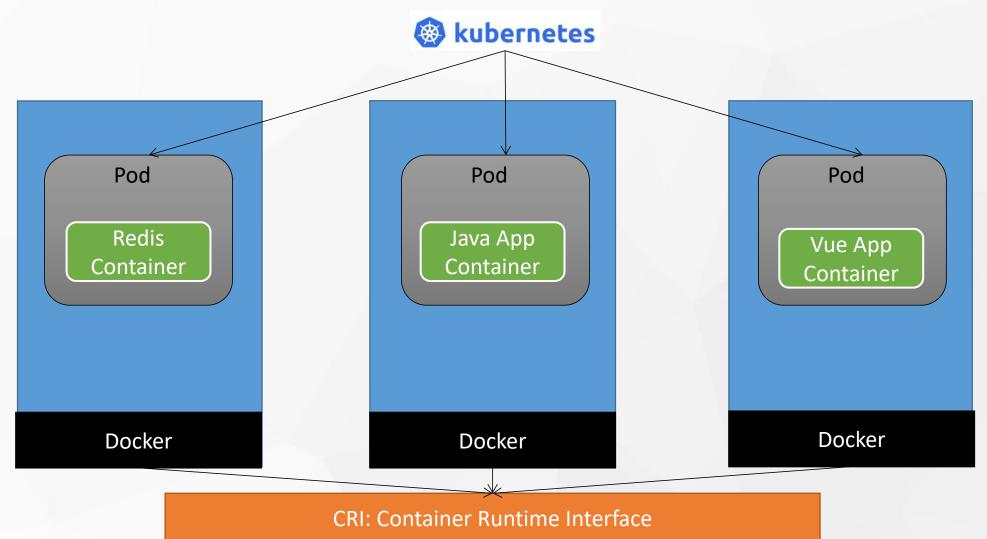
名称空间,用来对集群资源进行隔离划分。默认只隔离资源,不隔离网络

应用A 配置A	应用B 配置B	应用C 配置C	namespace prod
应用A-dev 配置A-dev		应用C 配置C-dev	namespace dev

Kubernetes-核心概念-Pod

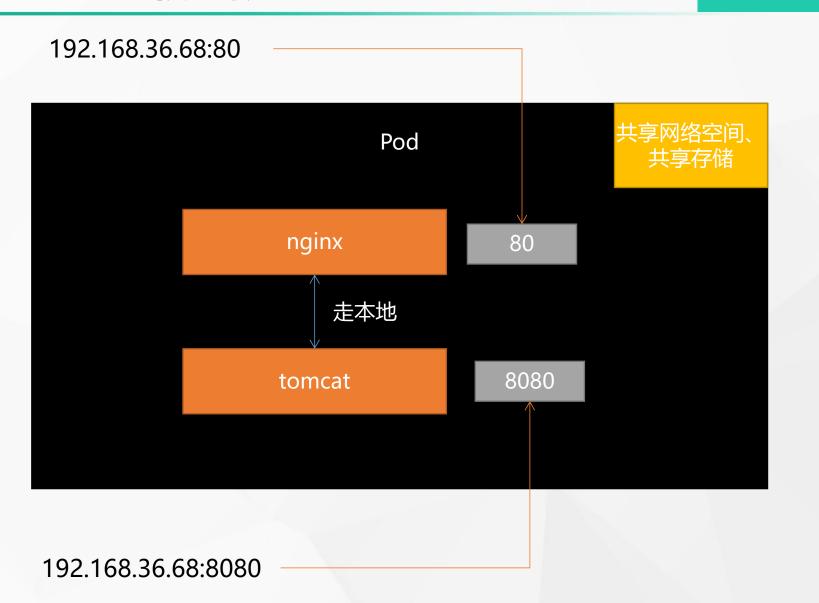


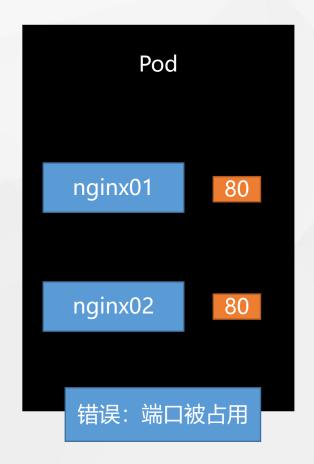
Pod:运行中的一组容器,Pod是kubernetes中应用的最小单位.



Kubernetes-核心概念-Pod

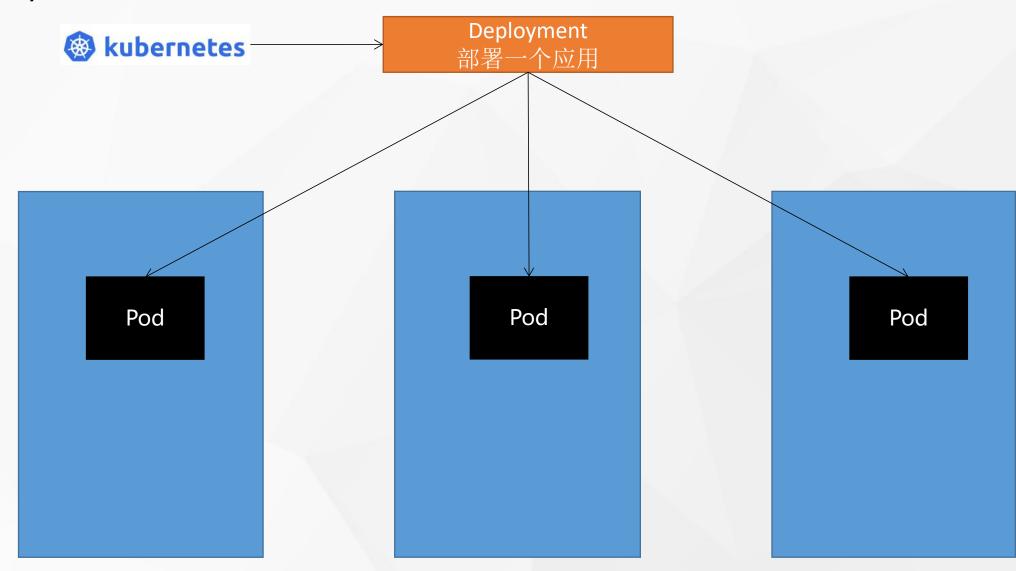






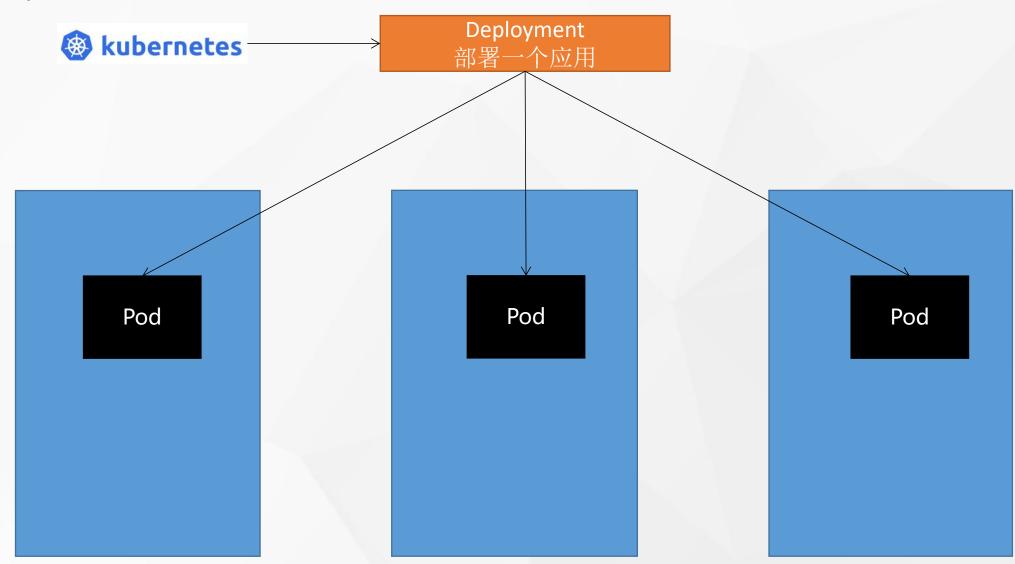


Deployment: 代表一次部署,产生1个或多个Pod



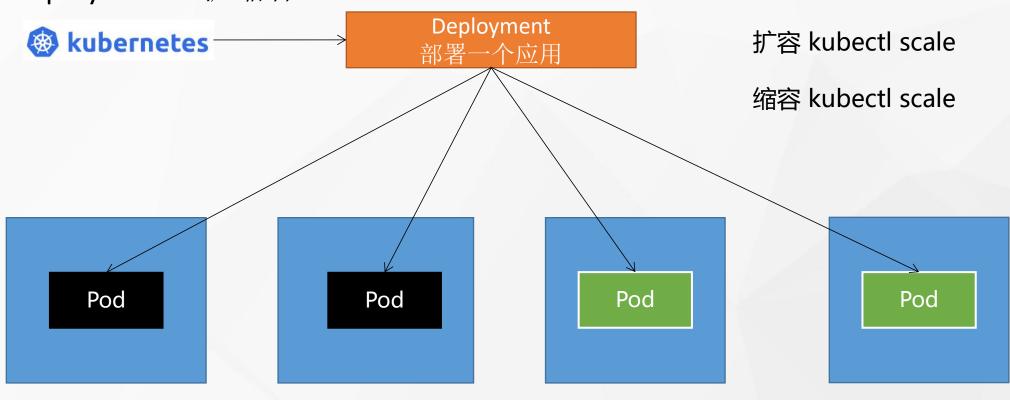


Deployment: 多副本



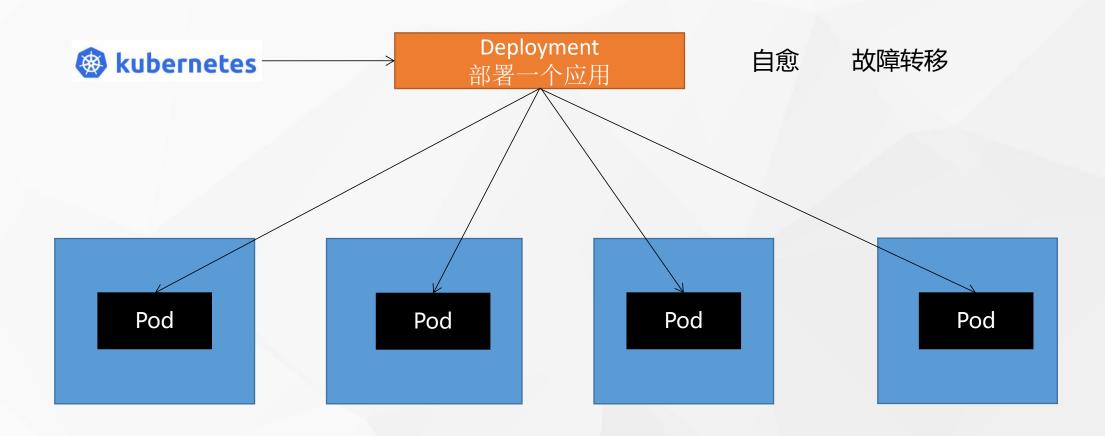


Deployment: 扩缩容



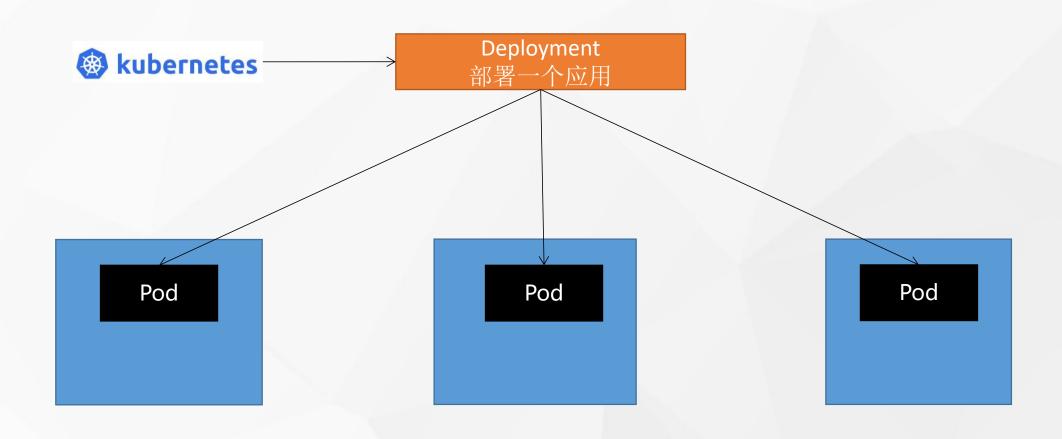


Deployment: 自愈&故障转移

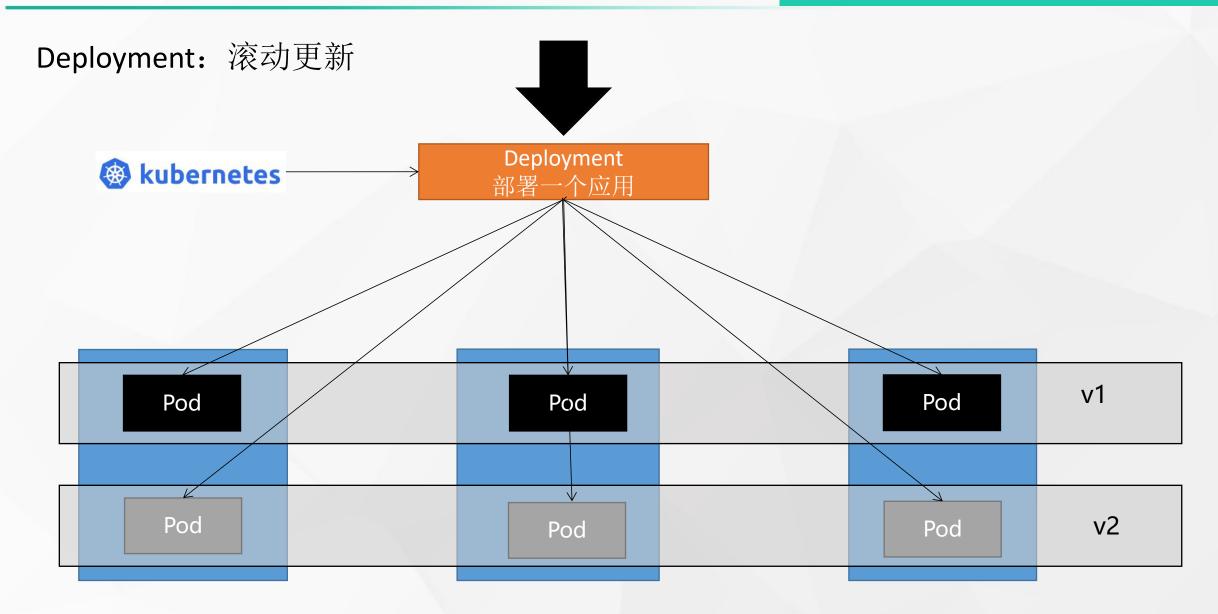




Deployment: 滚动更新

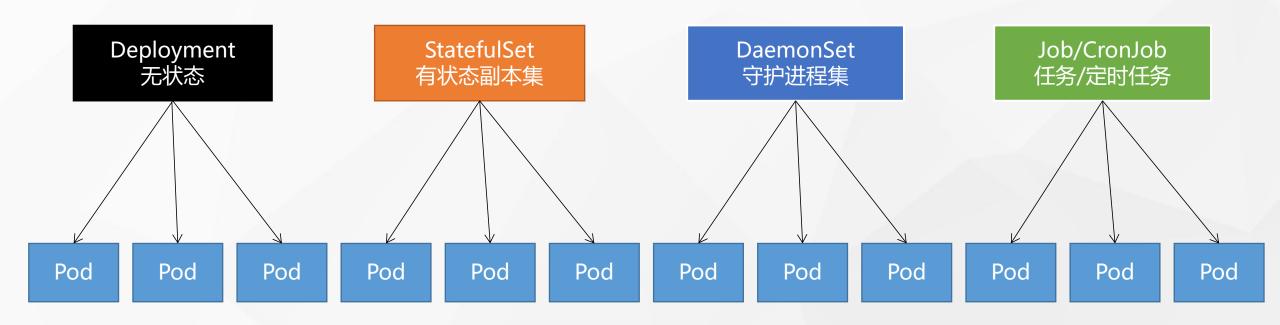






Kubernetes-核心概念-其他工作负载



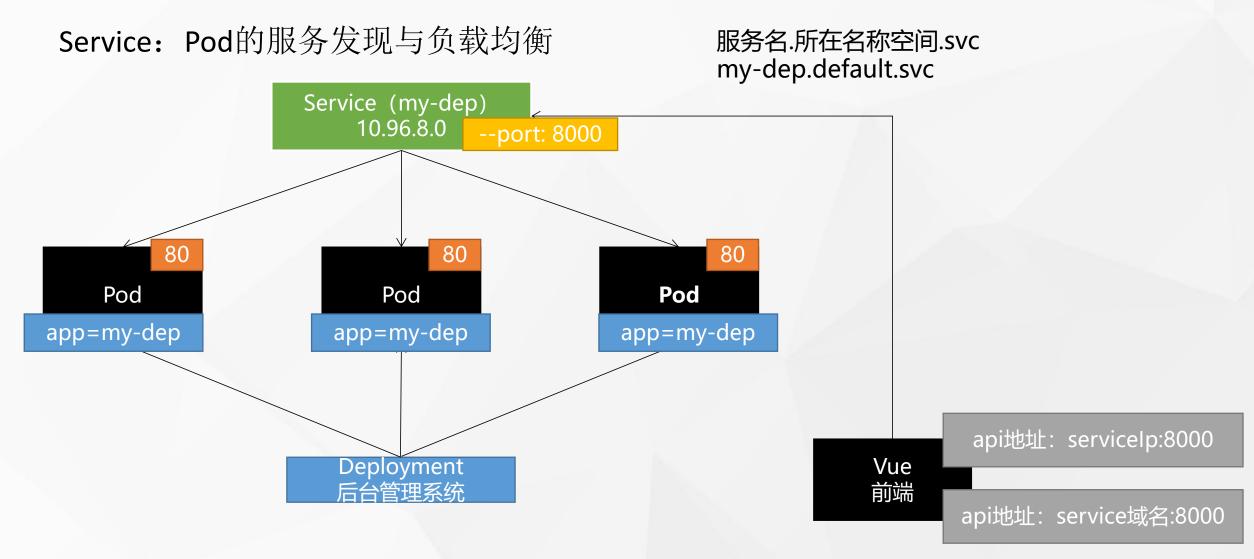


Deployment: 无状态应用部署, 比如微服务, 提供多副本等功能

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

Kubernetes-核心概念-Service-ClusterIP模式

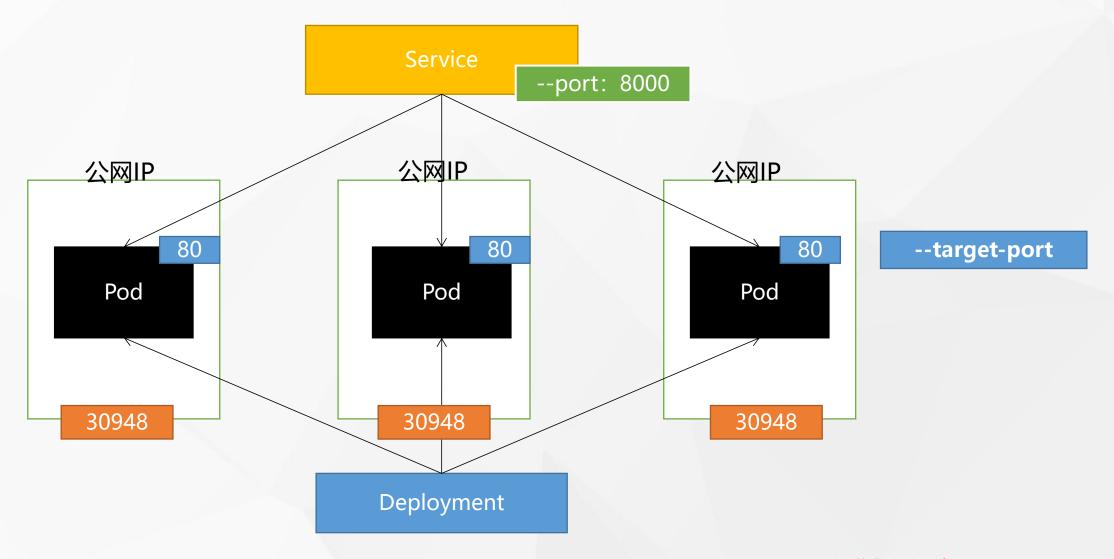




kubectl expose deploy my-dep --port=8000 --target-port=80 --type=ClusterIP 集群内使用service的ip:port就可以负载均衡的访问

Kubernetes-核心概念-Service-NodePort模式



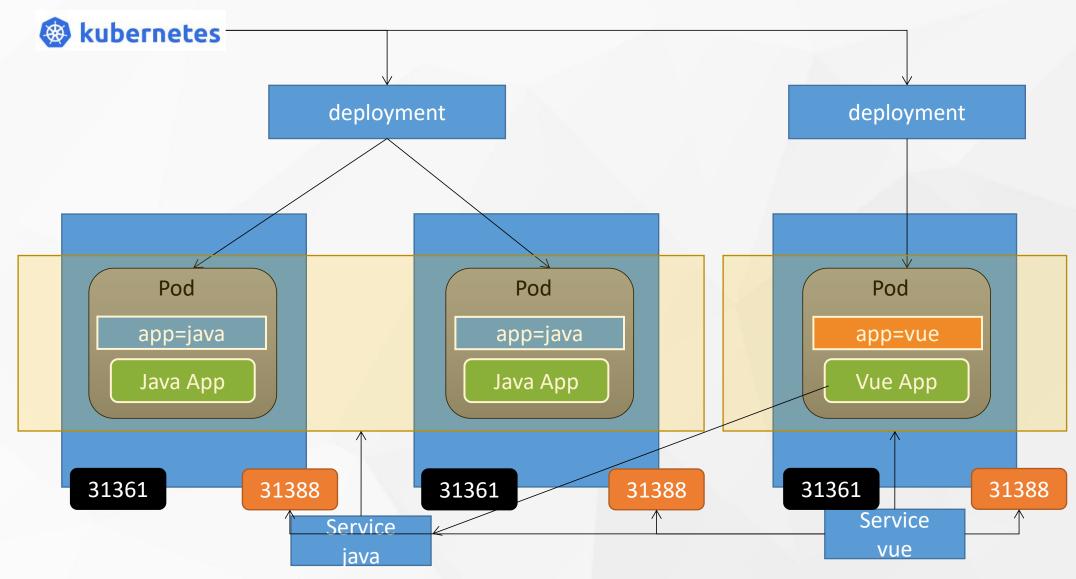


kubectl expose deploy my-dep --port=8000 --target-port=80 --type=ClusterIP: 集群内部的访问 kubectl expose deploy my-dep --port=8000 --target-port=80 --type=NodePort: 集群外也可以访问

Kubernetes-核心概念-Service



Service: Pod的服务发现与负载均衡



Kubernetes-核心概念-Ingress Ingress Ingress: Service的统一网关入口 nginx **kubernetes** atguigu.com/product user.atguigu.com order.atguigu.com Service层网络 10.96.0.0/16 service service servi<u>ce</u> service ...N deployment deployment deployment a (order) b (user) c (product) Pod Pod Pod Pod Pod app = b app = b app = a app = b app = a Pod Pod Pod app = capp = capp = cPod层网络 192.168.0.0/16

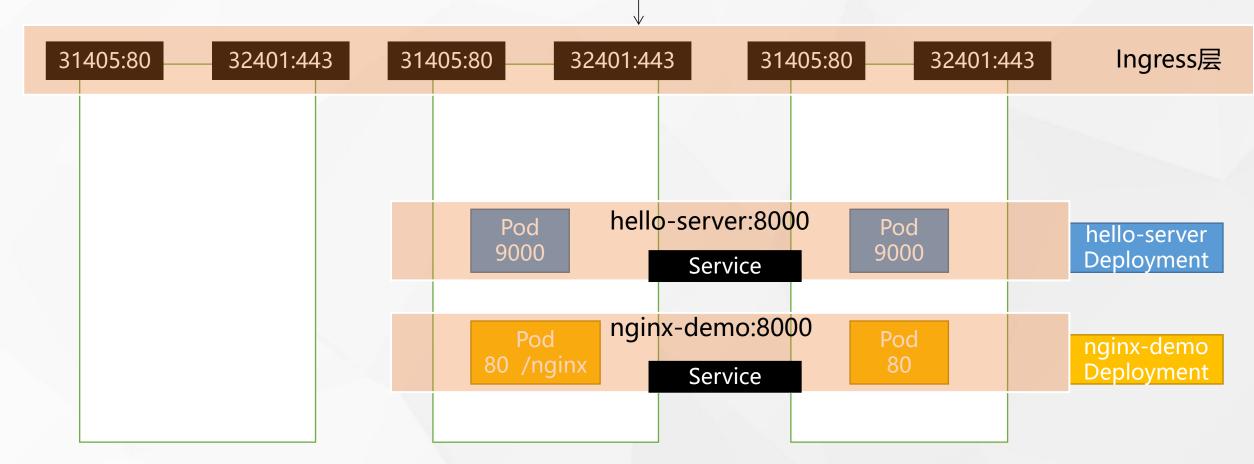
Kubernetes-核心概念-Ingress-实战





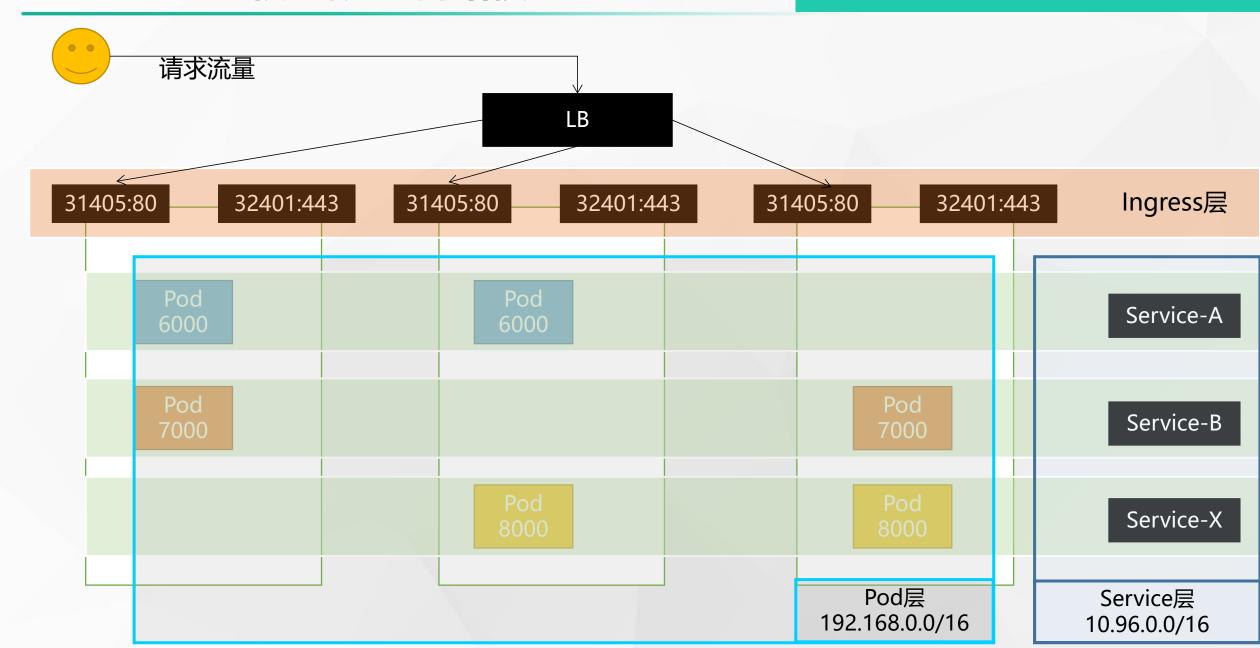
请求流量

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



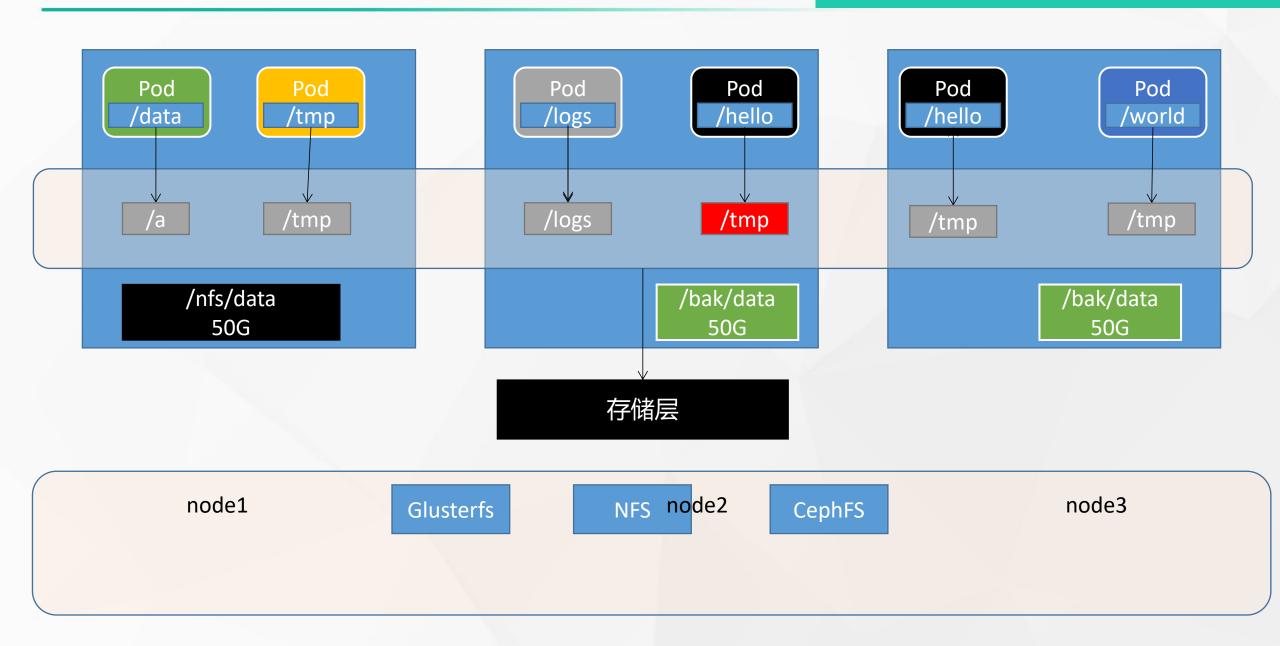
Kubernetes-核心概念-网络模型





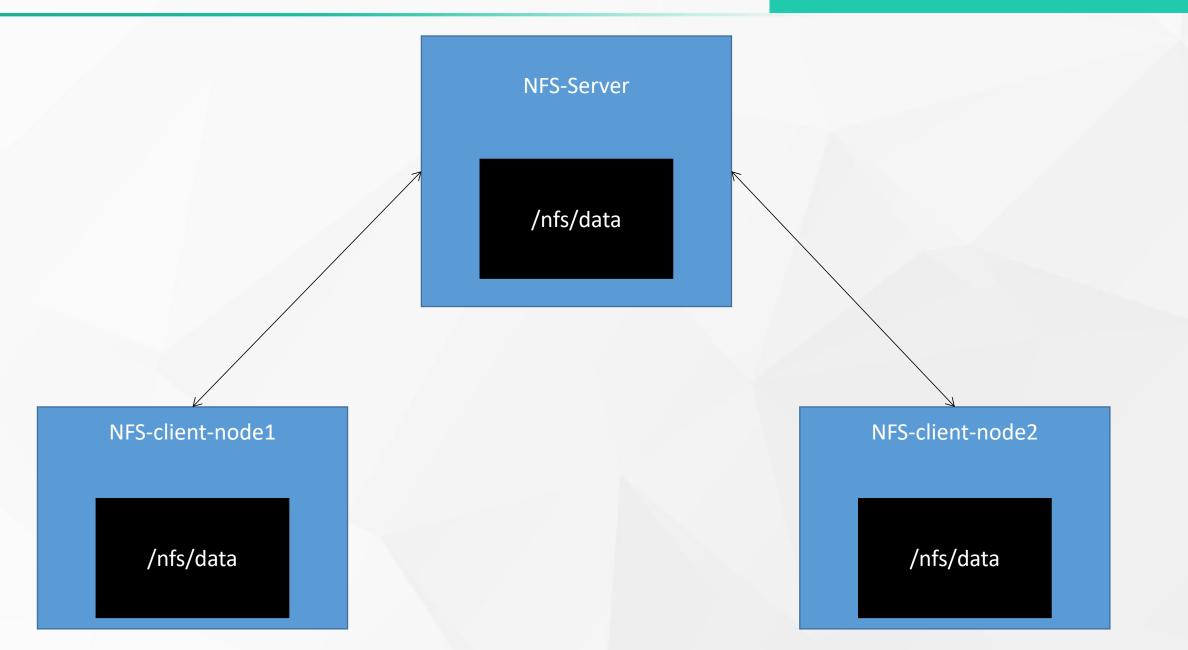
Kubernetes-存储抽象





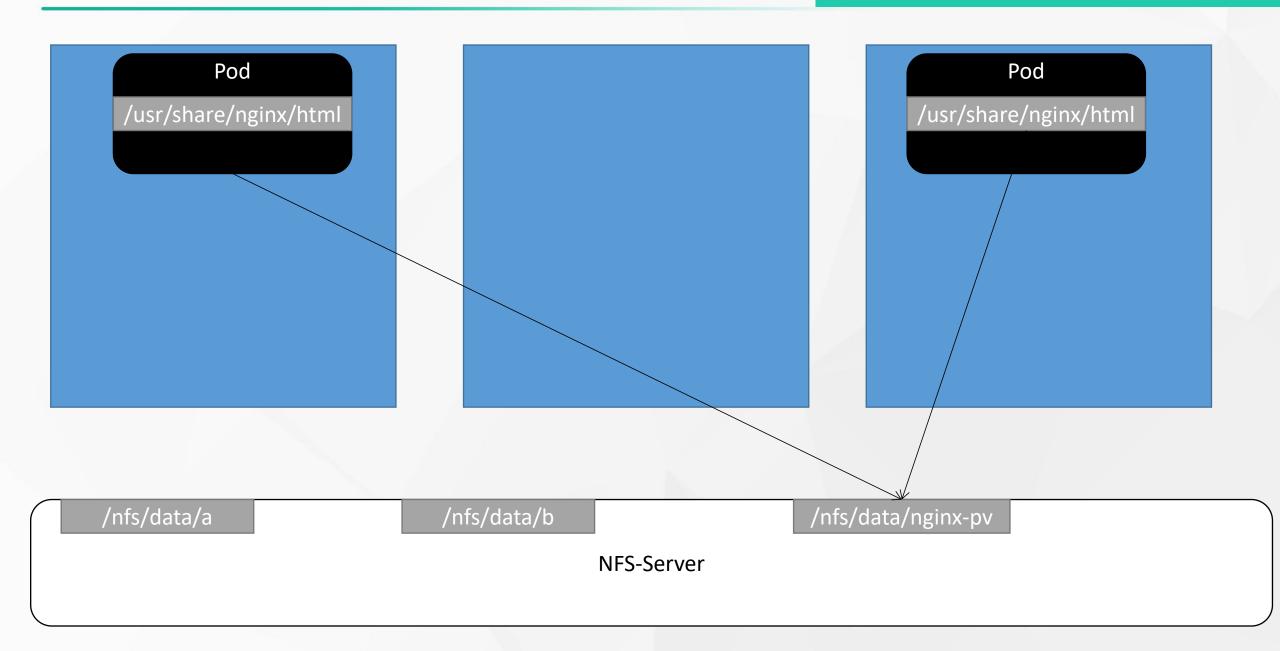
Kubernetes-NFS





Kubernetes-卷挂载



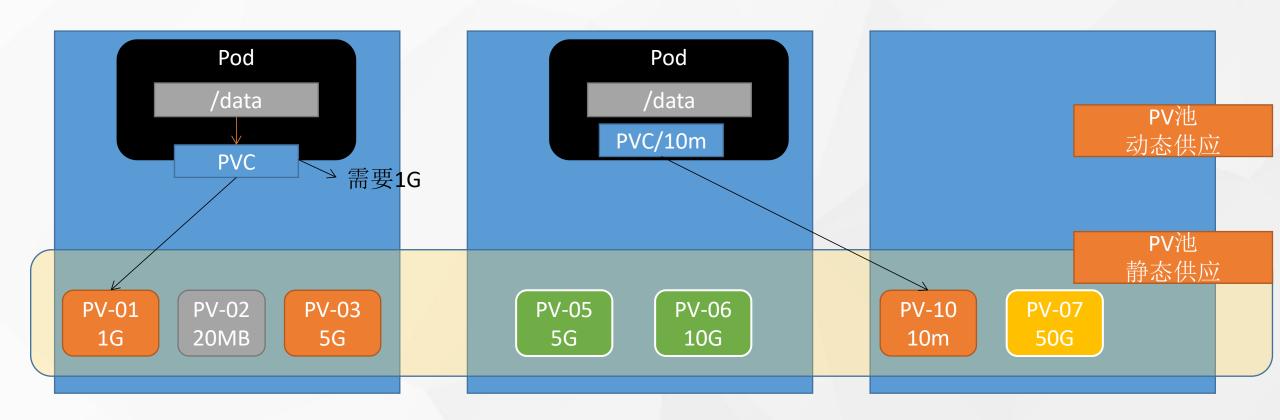


Kubernetes-PV与PVC



PV: PersistentVolume 持久卷

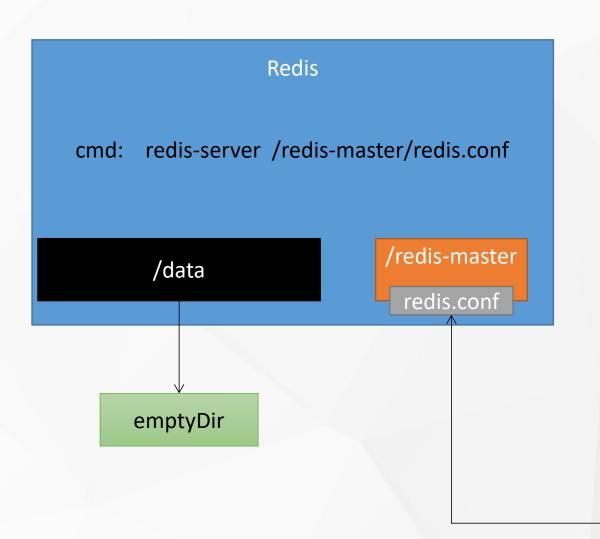
PVC: PersistentVolumeClaim 持久卷申明



Kubernetes-ConfigMap



ConfigMap: 配置集。抽取应用的配置



```
apiVersion: v1
         #data是所有真正的数据
 data:
   redis.conf:
     appendonly yes
 kind: ConfigMap
 metadata:
   name: redis-conf
   namespace: default
14
         volumeMounts:
         - mountPath: /data
15
           name: data
16
17
          - mountPath: /redis-master
           name: config
18
       volumes:
19
20
         - name: data
21
           emptyDir: {}
22
          name: config
23
           configMap:
             name: redis-conf
24
25
             items:
26
             - key: redis.conf
                path: redis.conf
```

Kubernetes-Secret



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: regared
```



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/

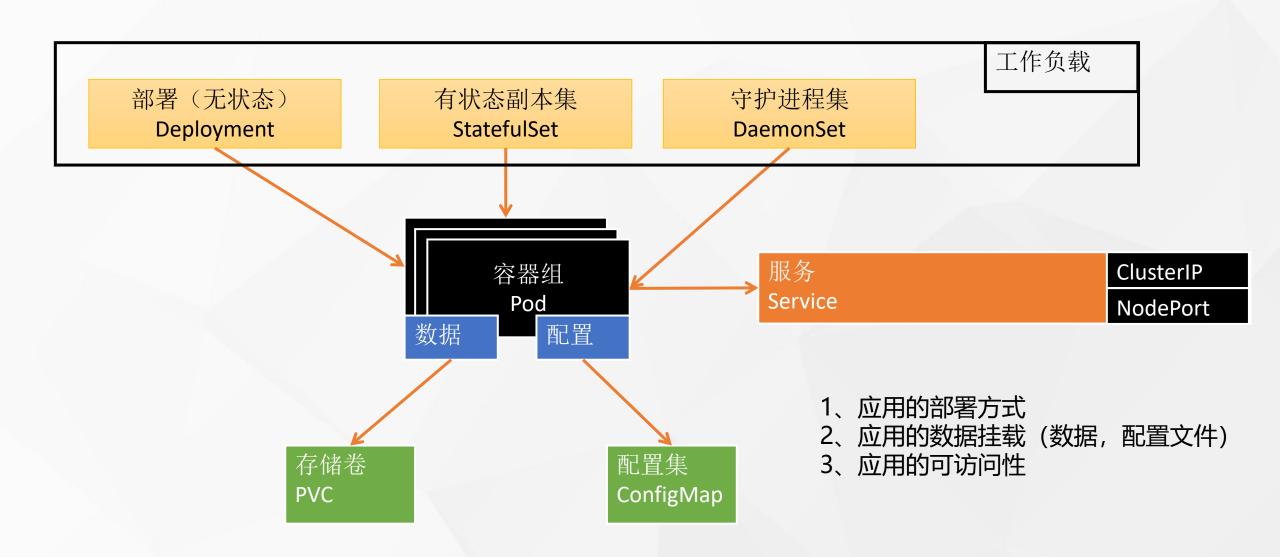


KubeSphere-中间件部署实战

MySQL, Redis, ElasticSearch

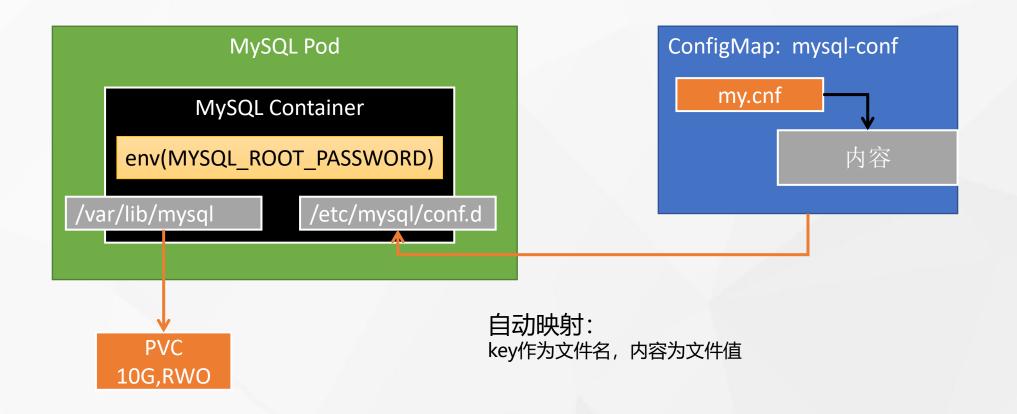
KubeSphere-部署应用





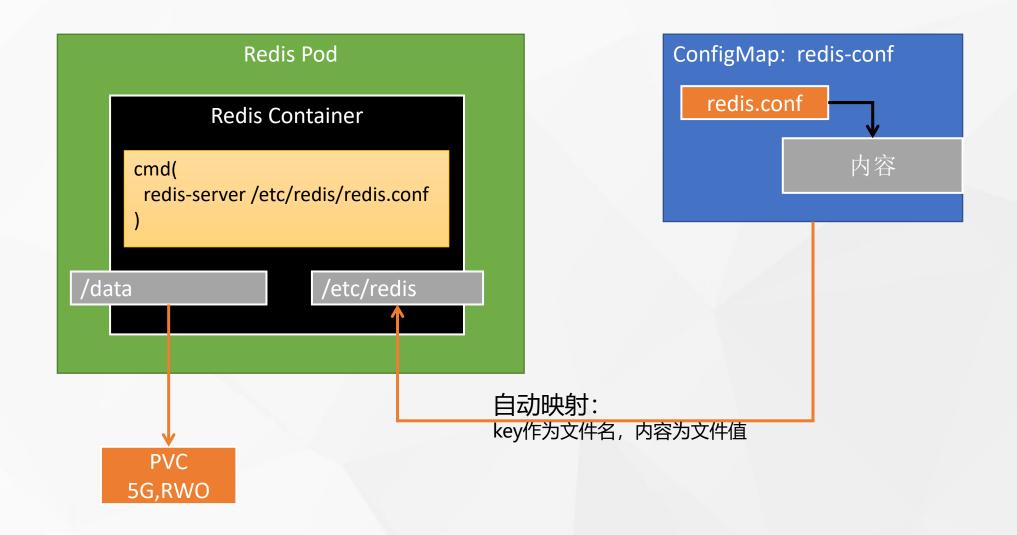
KubeSphere-部署MySQL





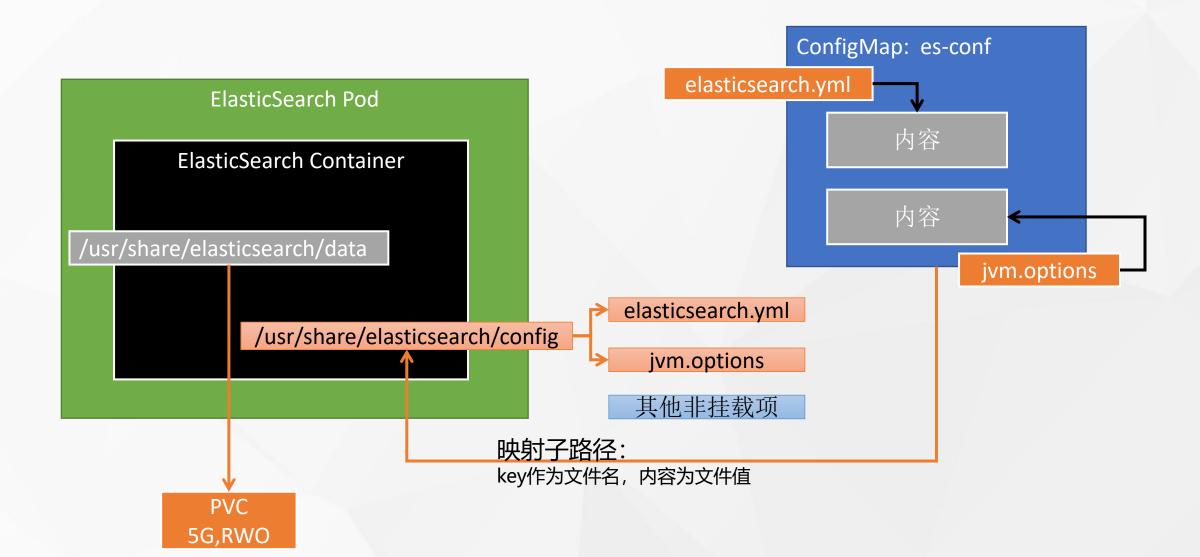
KubeSphere-部署Redis





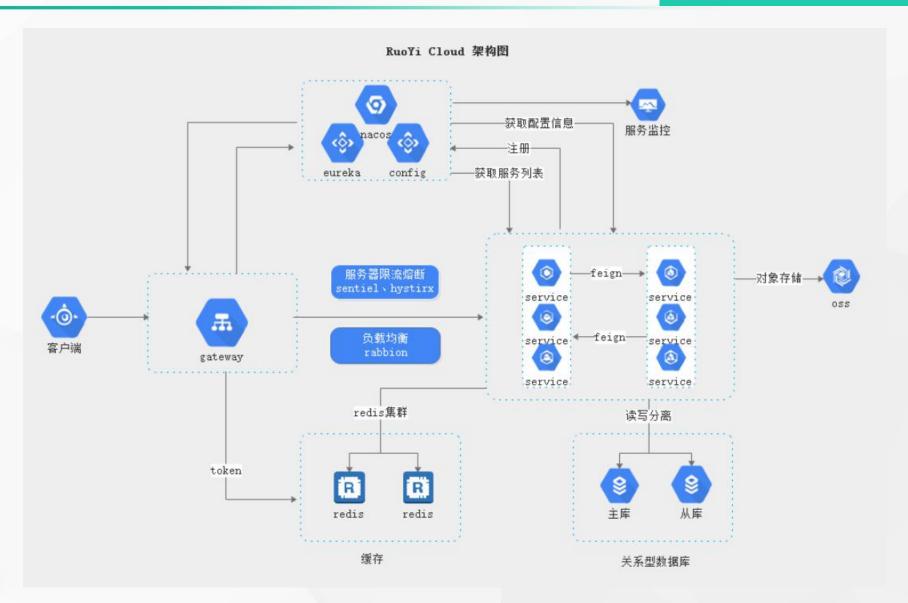
KubeSphere-部署ElasticSearch





KubeSphere-部署Ruoyi-Cloud





https://gitee.com/zhangmrit/ruoyi-cloud/blob/nacos/doc/ruoyi-cloud.png



中间件 有状态、数据导入

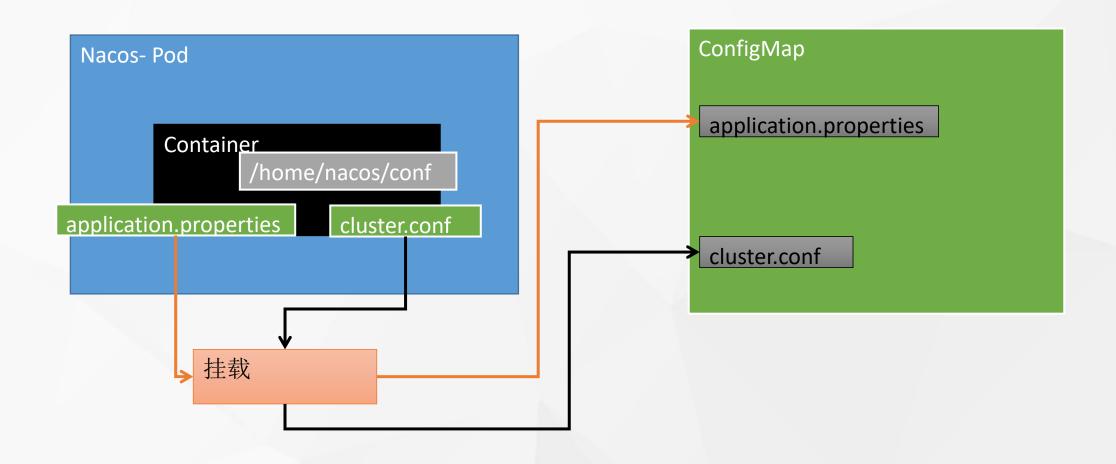
微服务 无状态、制作镜像

网络 各种访问地址

配置 生产配置分离、URL

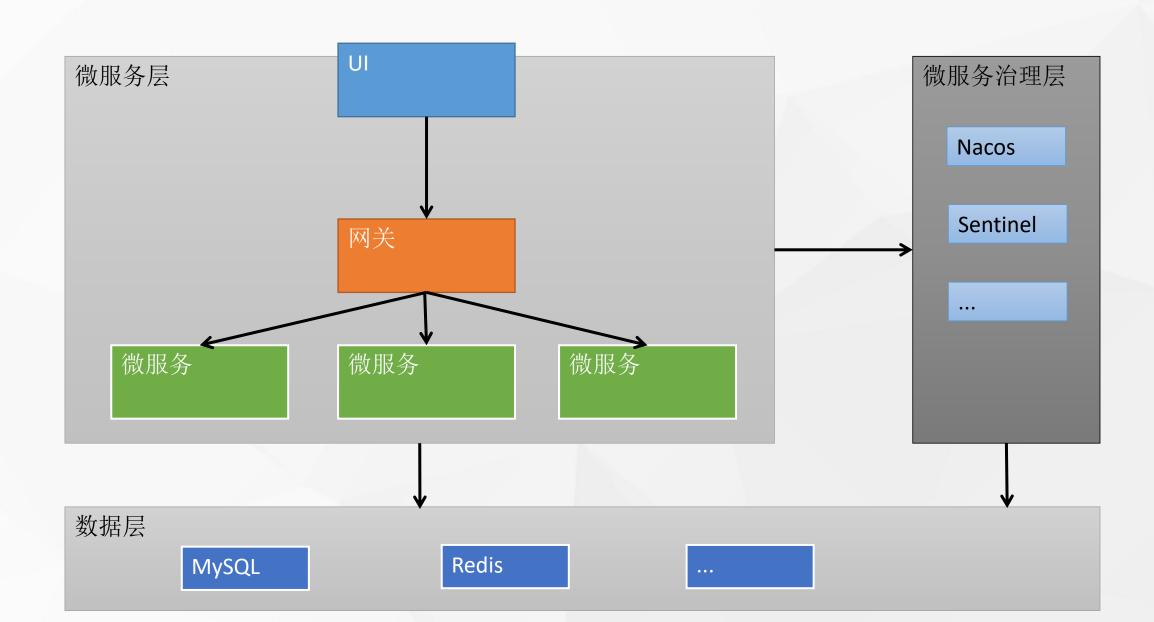
KubeSphere-Ruoyi-Cloud-上云-Nacos





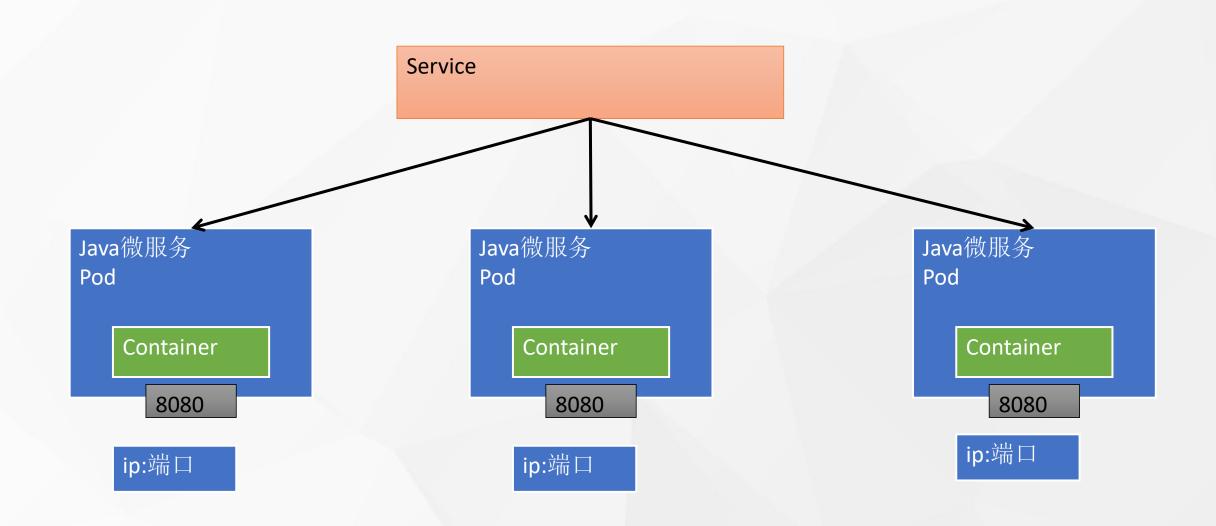
KubeSphere-Ruoyi-Cloud-上云-架构





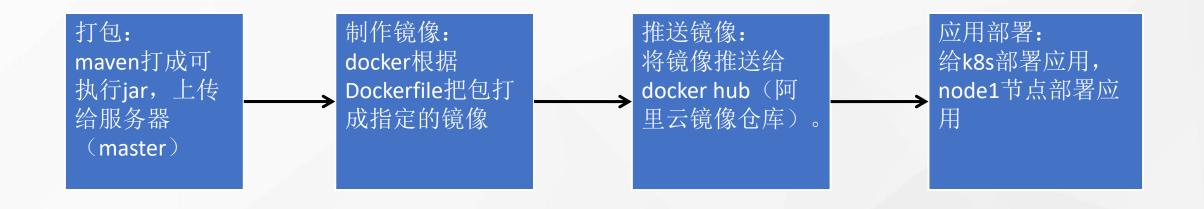
KubeSphere-Ruoyi-Cloud-上云-微服务





KubeSphere-Java微服务k8s部署





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