ArchSummit
全球架构师峰会 2017

主办方 Geekbang 极客邦科技 InfoQ

美丽联合容器云平台建设的实战分享

张振华 （郭嘉）

美丽联合集团-基础平台 虚拟化

主办方 Geekbang. 极客邦科技 InfoQ

AiCon

全球人工智能与机器学习技术大会

助力人工智能落地

2018.1.13 - 1.14　北京国际会议中心

扫描关注大会官网

极客时间

重拾极客精神·提升技术认知

下载极客时间App

获取有声IT新闻、技术产品专栏，每日更新

扫一扫下载极客时间App

SPEAKER
INTRODUCE

张振华 （郭嘉）

美丽联合集团 高级技术专家

- 2014年加入美丽联合，虚拟化团队负责人，带领团队从无到有建设集团的私有 IaaS 平台和 PaaS 平台，见证了美丽联合集团从物理机、虚拟机到容器的技术演进。
- 目前聚焦在美丽联合集团容器云平台的研发和基于容器的 DevOps 项目落地。
- 十余年软件研发管理经验，在加入美丽联合集团之前，曾在英特尔、思科等公司工作。

TABLE OF
CONTENTS 大纲

1

美丽联合容器平台的演进

稳定 & 效率

2

技术方案

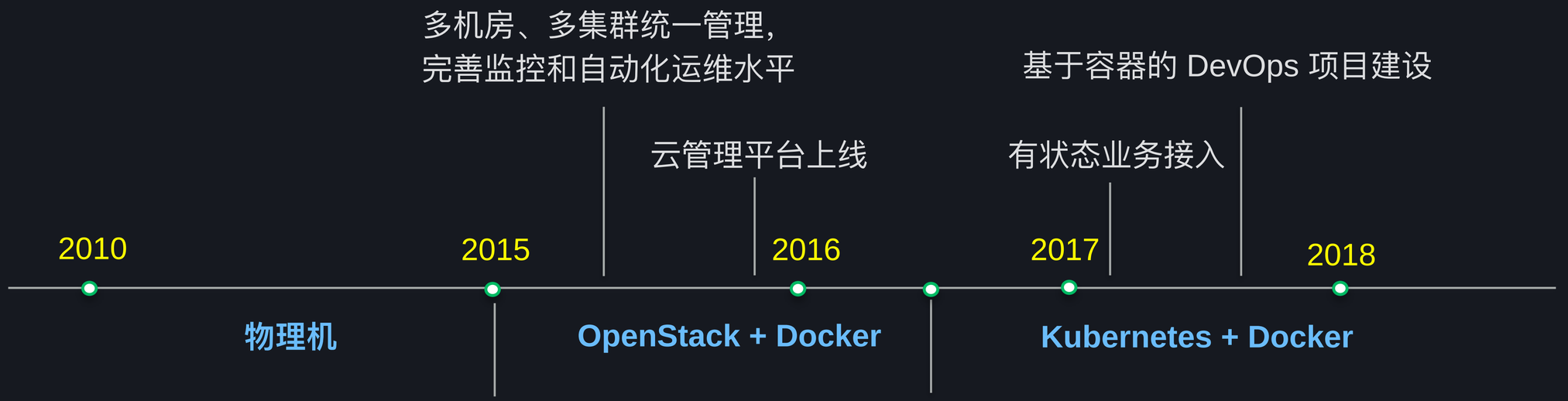
有状态业务 & 如何应对

3

DevOps 平台建设

经验 & 体会

容器云平台演进



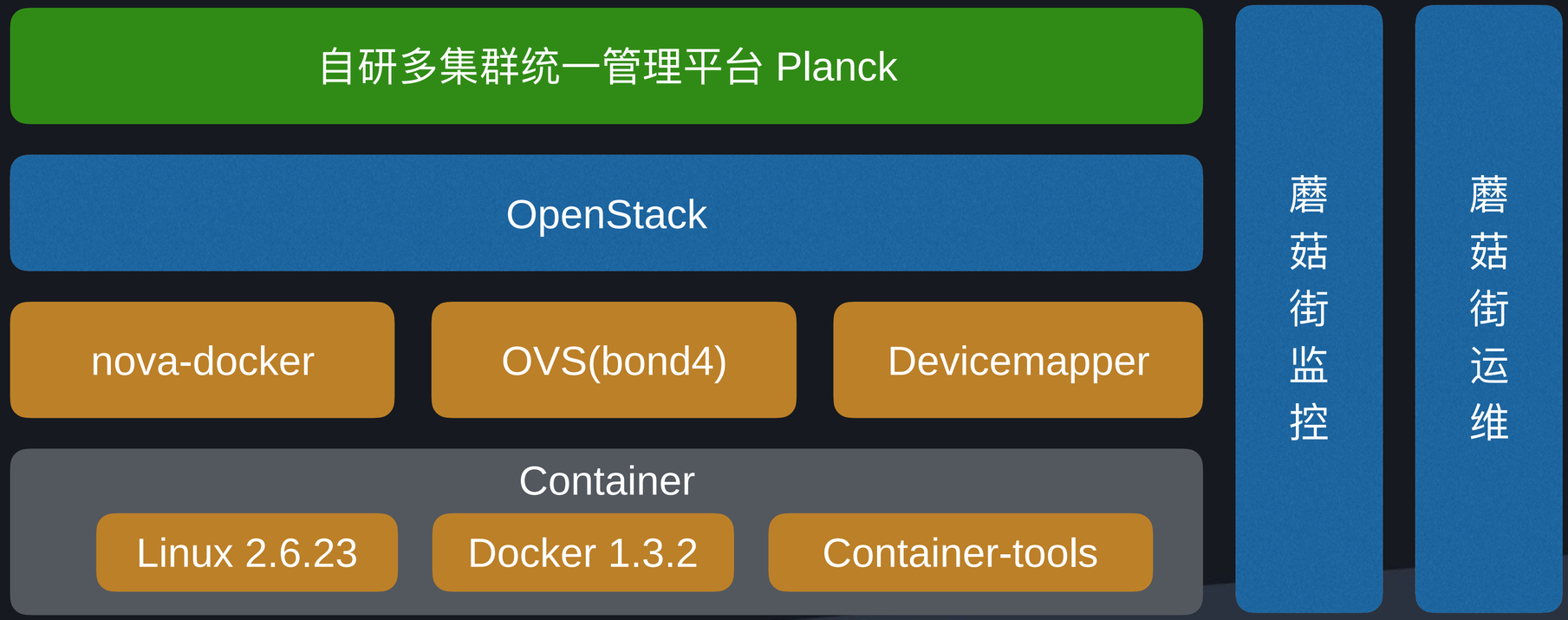
- 传统的数据中心
- 机器分散管理
- 运维、监控不完善
- 发布、扩容比较人肉

- 从零构建私有云平台
- 多机房多集群统一管理平台
- VM 化，静态资源分配
- 和运维、监控对接，逐步完善
- 面向机器的发布

- 演进至 Kubernetes 方案
- 多机房多集群统一管理平台
- 动态资源分配
- 与运维、监控对接，镜像发布
- 接入有状态业务

OpenStack 时代

基于 OpenStack + Docker 方案



Web Portal — Planck

Kubernetes 时代

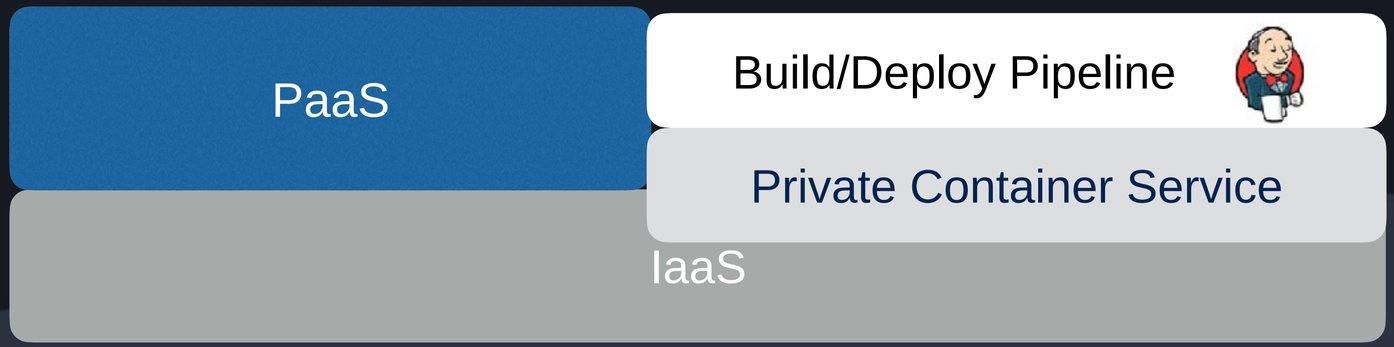
? 适合的才是最好的!

优点：先进的理念、可扩展性好、社区活跃

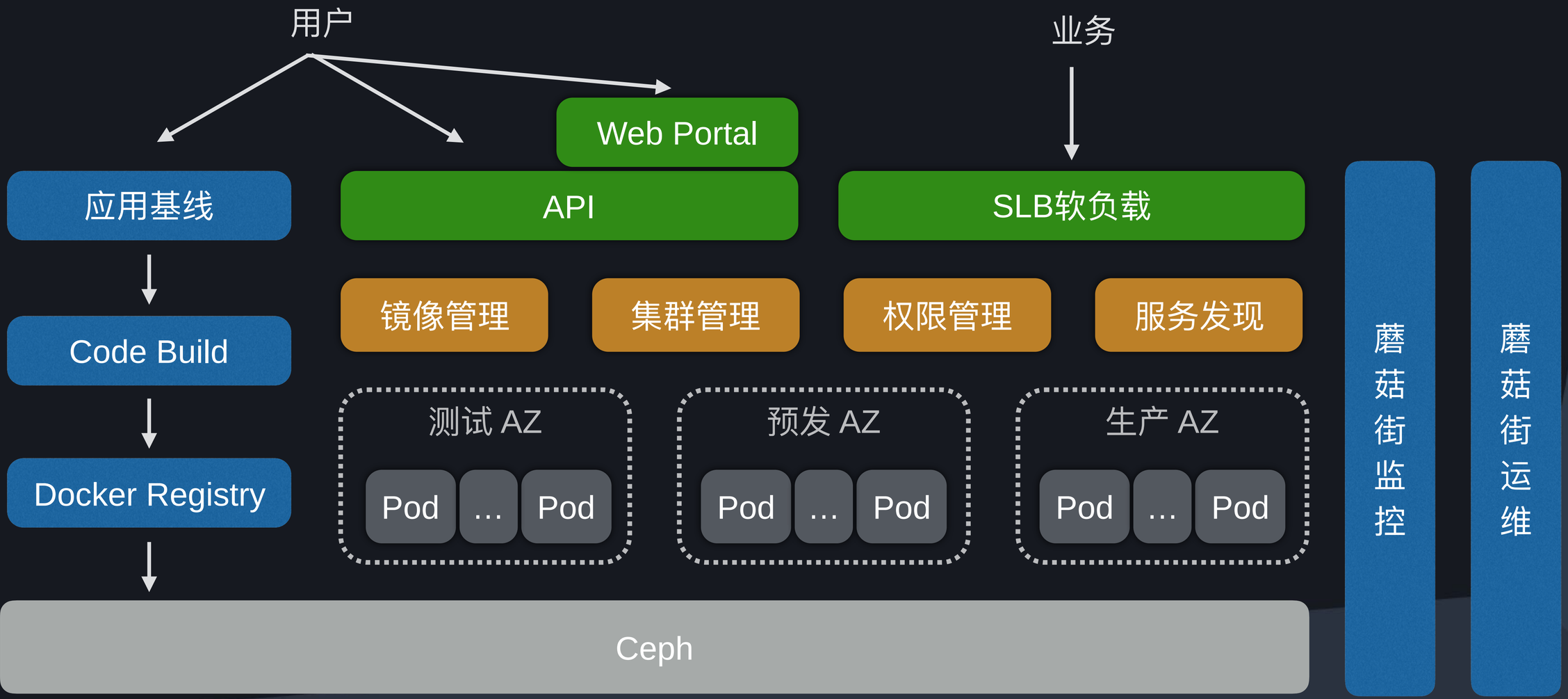
要引擎还是要汽车?

缺点：自建成本高、仍在高速演进中

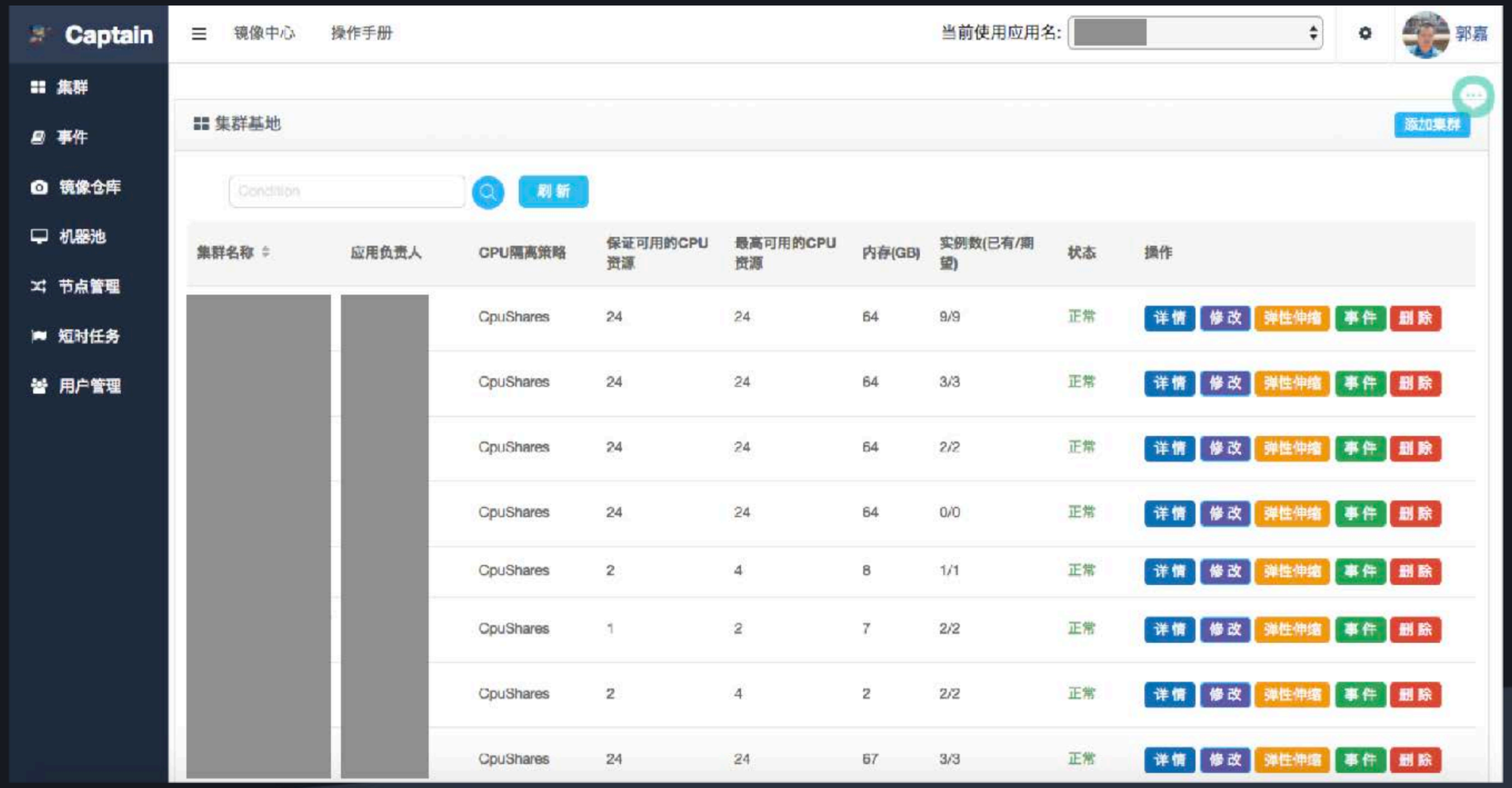
不仅仅是 PaaS !



总体架构


用户
业务
Web Portal
应用基线
API
SLB软负载
Code Build
镜像管理
集群管理
权限管理
服务发现
Docker Registry
测试 AZ
预发 AZ
生产 AZ
Pod
…
Pod
Pod
…
Pod
Pod
…
Pod
蘑菇街监控
蘑菇街运维
Ceph

Web Portal — Captain



提升稳定性

来自稳定性的挑战

硬件故障

- 完善日志，进程等监控
- 建立日常值班制度
- 故障演练，模拟硬件故障，网络中断
- 冗灾预案演练

软件问题

- 统一集群配置，并不断优化
- 自动化定期健康巡检
- 完善性能压测
- 核心数据定期备份
- 完善自动化测试

安全漏洞

- 和安全团队紧密合作评估安全漏洞的影响

人为事故

- 制定操作红线和规范
- 规范流程
- 账号体系

提升效率

多维度提升效率

机器的效率

- CPU / 内存动态扩缩容
- 动态资源申请
- 统一资源池

监控效率

- Docker Pool 监控
- 容器 oom 监控
- TCP连接状态监控
- TCP重传率监控
- pod迁移时的自动通知

人的效率

- IM自动应答（宿主机信息查询）

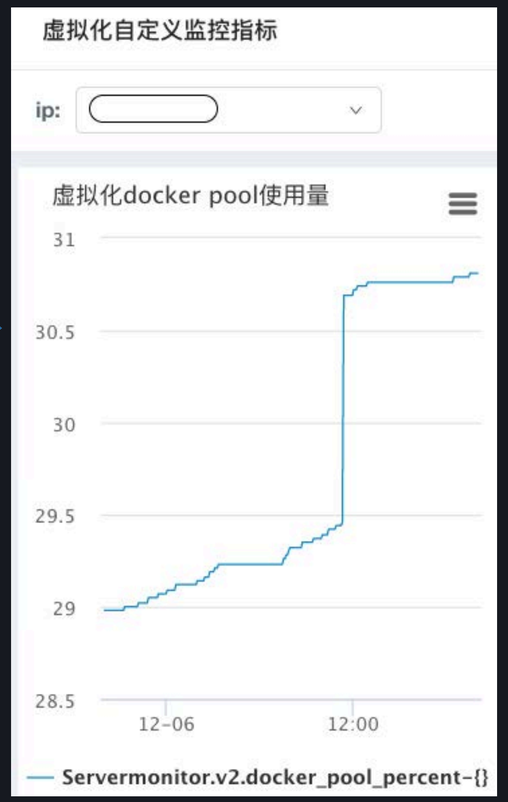
解决问题效率

- 系统问题定位排查指南
- 系统问题分析定位平台
- 工具化，体系化
- 知识库沉淀

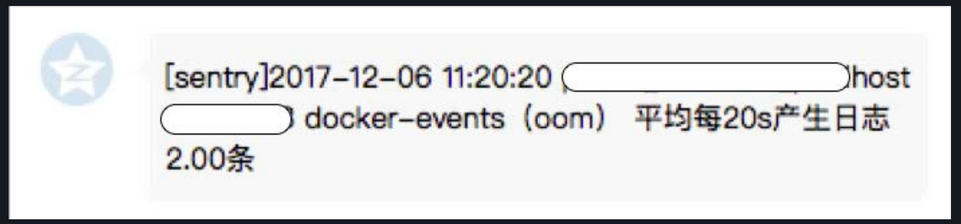
监控与告警

```
Server Version: 1.10.3
Storage Driver: devicemapper
 Pool Name: dockerpool-docker--pool
 Pool Blocksize: 524.3 kB
 Base Device Size: 107.4 GB
 Backing Filesystem: xfs
 Data file:
 Metadata file:
 Data Space Used: 132.7 GB
 Data Space Total: 430.7 GB
 Data Space Available: 297.9 GB
 Metadata Space Used: 14.41 MB
 Metadata Space Total: 482.3 MB
 Metadata Space Available: 467.9 MB
 Udev Sync Supported: true
 Deferred Removal Enabled: false
 Deferred Deletion Enabled: false
 Deferred Deleted Device Count: 0
 Library Version: 1.02.107-RHEL7 (2015-10-14)
```

devicemapper docker pool使用率监控



$ cat /sys/fs/cgroup/memory/docker/<pid>/memory.oom_control
oom_kill_disable 0
under_oom 0



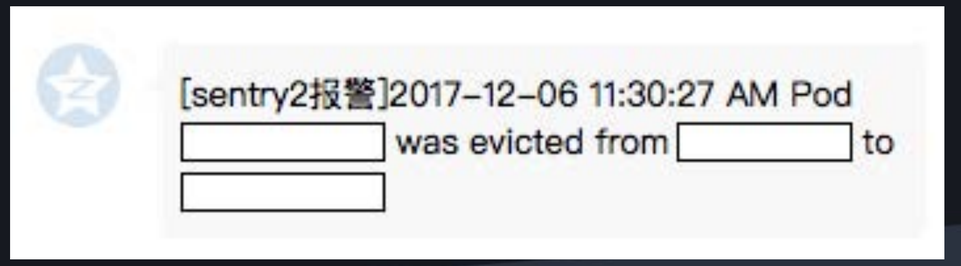


TABLE OF

CONTENTS 大纲

1

美丽联合容器平台的演进

稳定 & 效率

2

技术方案

有状态业务 & 如何应对

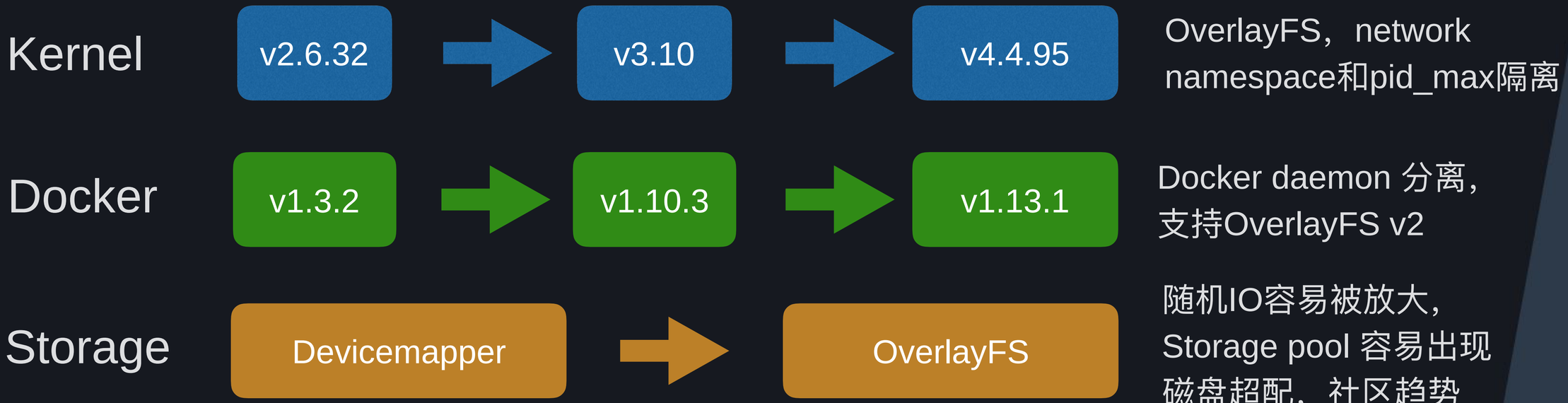
3

DevOps 平台建设

经验 & 体会

版本演进的动机

更好的隔离性 & 更高的稳定性 & 更强的性能



关键技术与创新

隔离性

- CPU Set 支持
- 网络QoS
- 异步IO隔离

稳定性&可用性

- 集群自身监控
- 定期检查
- Cadvisor
- top hook

高可用

- 管理节点 HA
- 有状态业务 Stateful Set 增强

Kubernetes Cluster

网络

- 基于 OVS自研的mogunet
- 多网段支持

存储

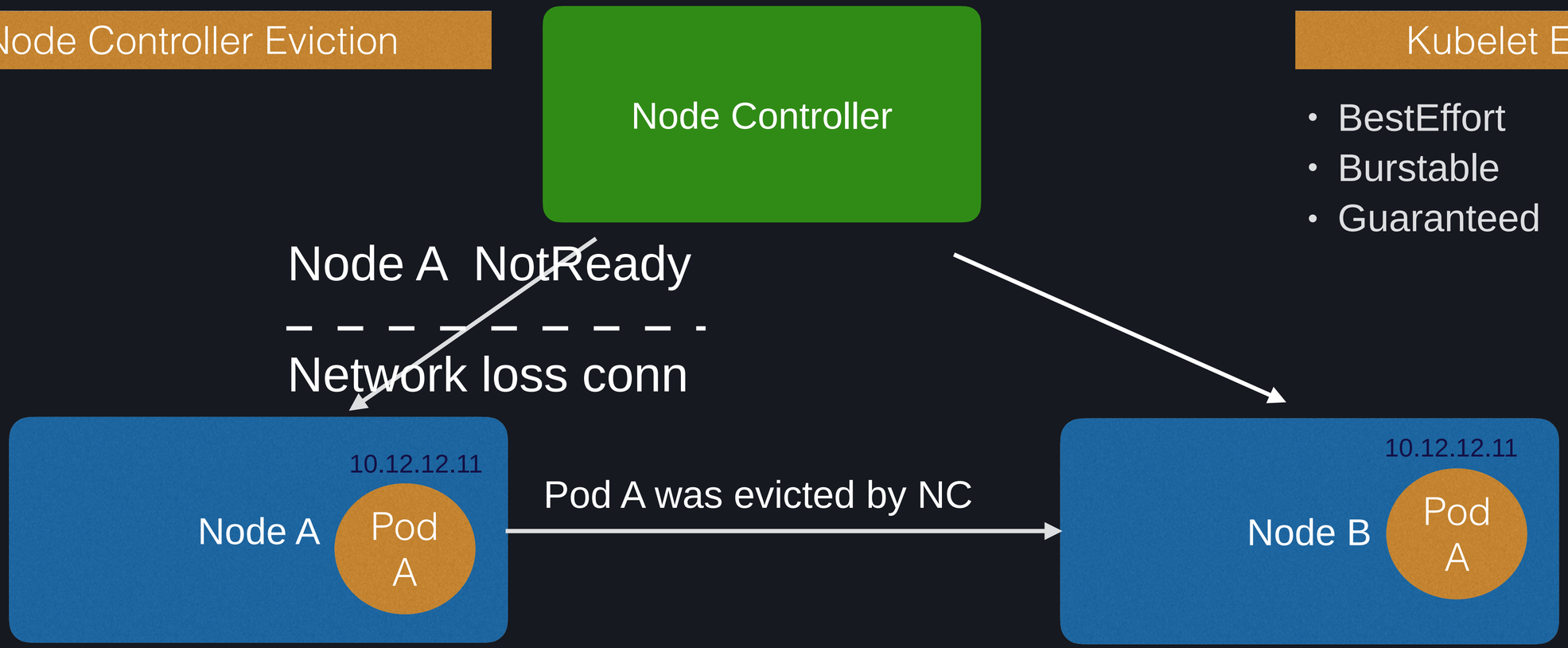
- DeviceMapper -> OverlayFS
- mount宿主机目录的配额

调度

- 基于磁盘类型调度
- 基于业务 / 资源亲和性调度

Pod驱逐策略

Node Controller Eviction

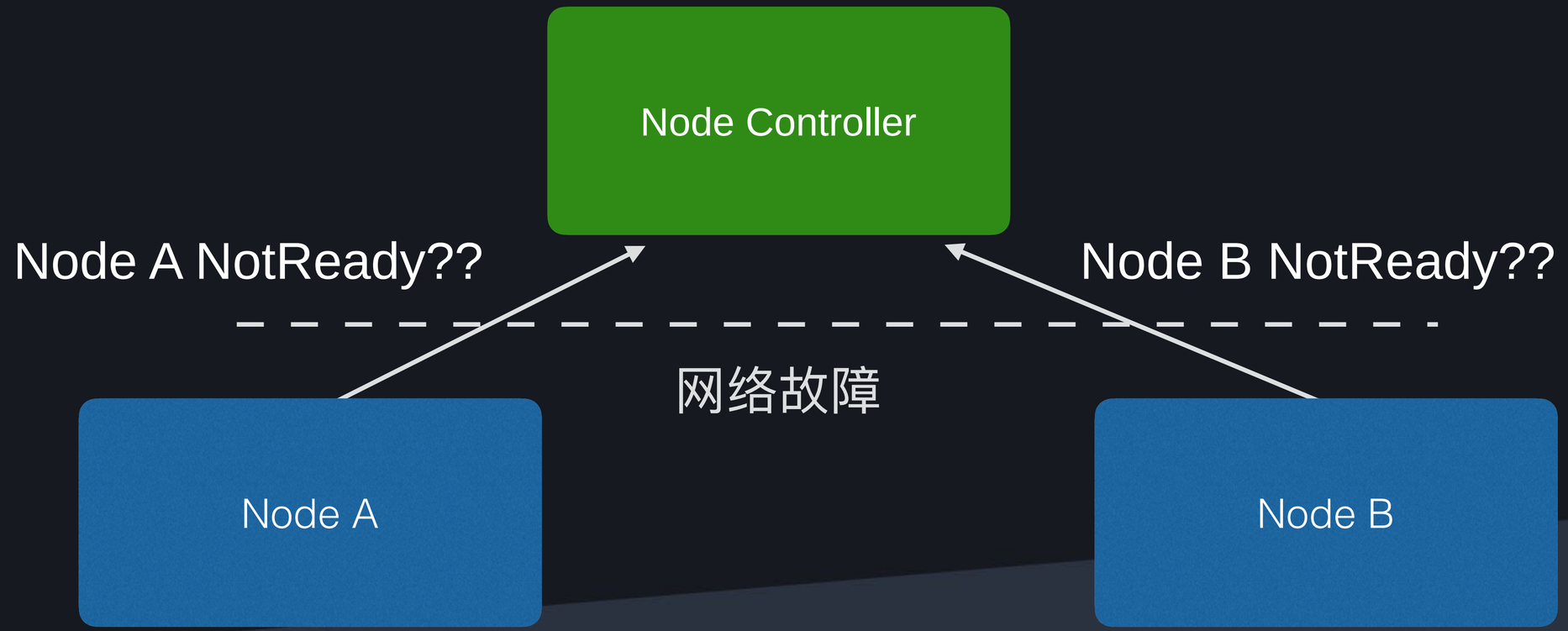


Kubelet Eviction

- BestEffort
- Burstable
- Guaranteed

Pod驱逐策略

参考《记一次 k8s 集群单点故障引发的血案》



有状态业务的挑战

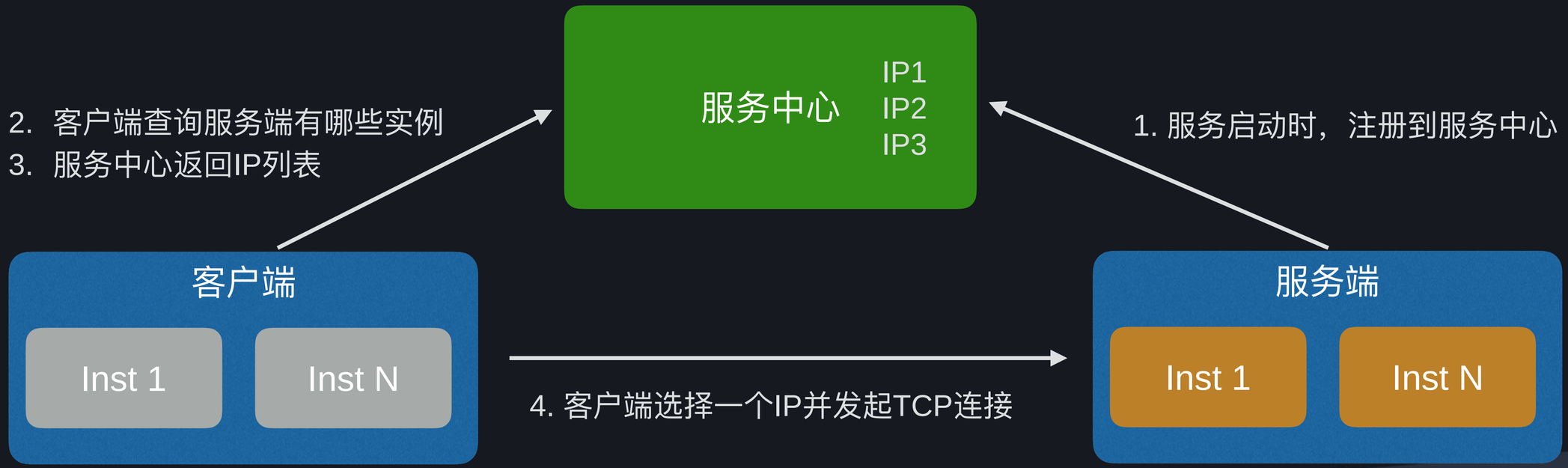
应用调用的依赖

DNS等配置的依赖

数据持久层的依赖

应用依赖于服务端 IP 地址不变

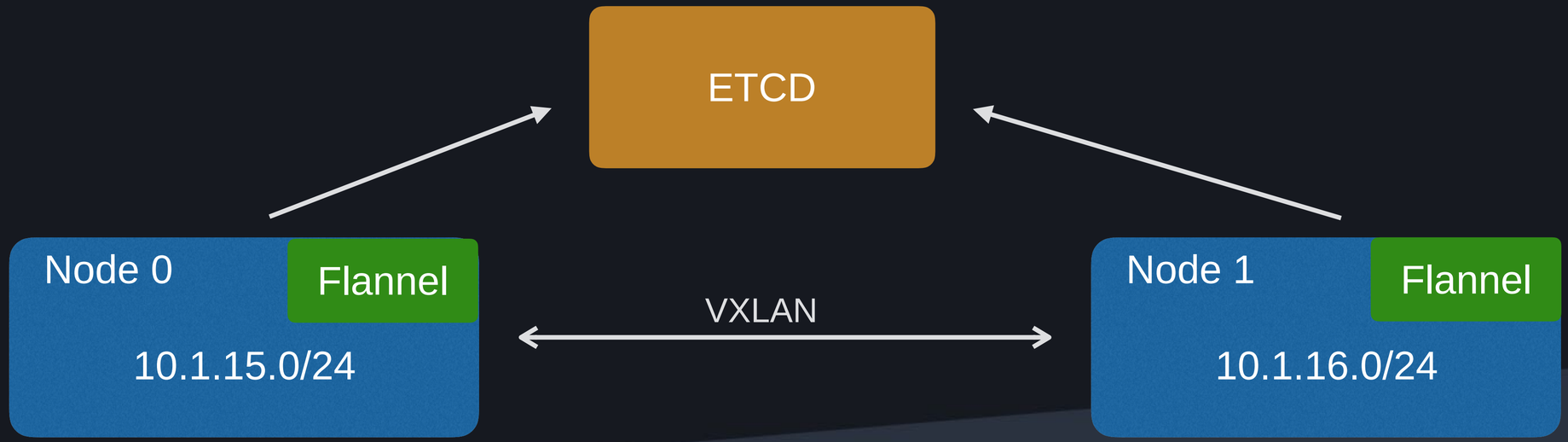
服务发现依赖于服务实例的IP 地址不变



Flannel 网络方案

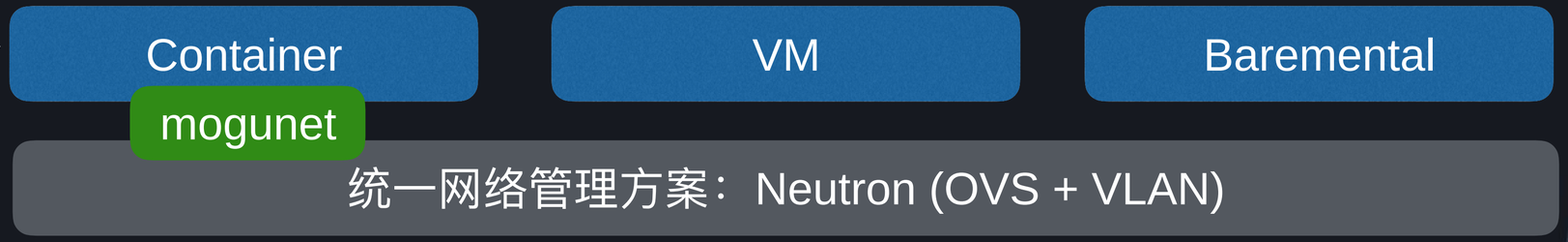
Flannel 方案静态规划网络资源，每个物理节点独占一个C类网段。虽然简单易用，但缺点也很明显：

- 容器跨物理机迁移时，**IP一定会变化**
- Overlay 带来的性能开销



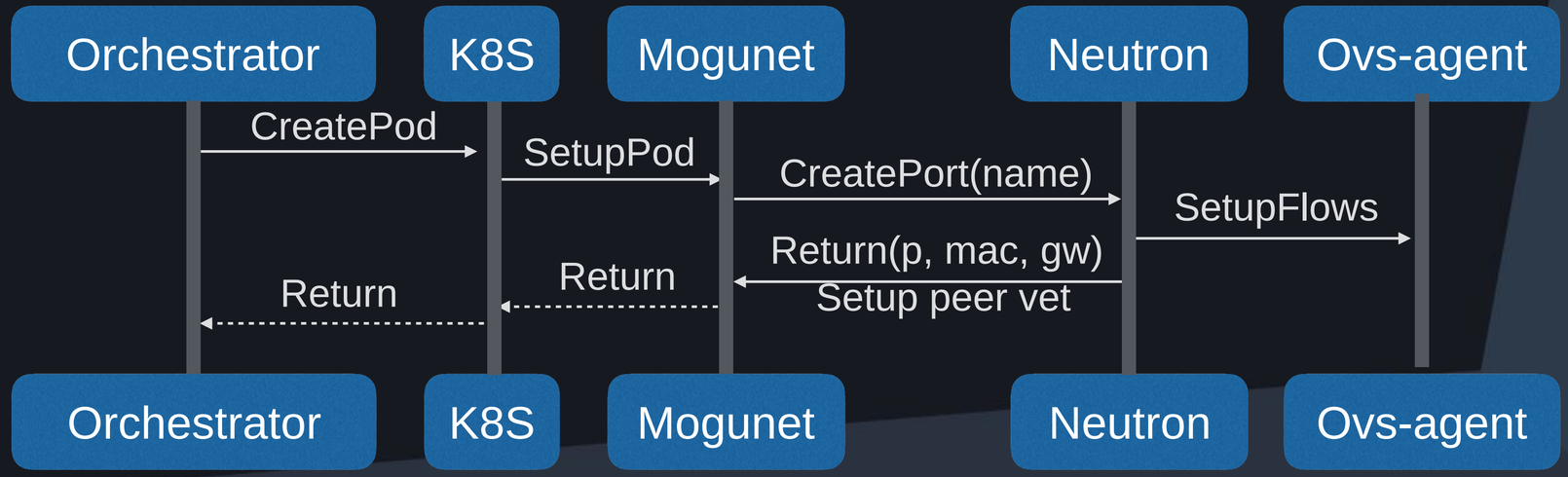
自研网络插件 mogunet

基于 Neutron/OVS 自研 K8S 网络插件，统一管理 container，VM，baremental 的网络资源分配。

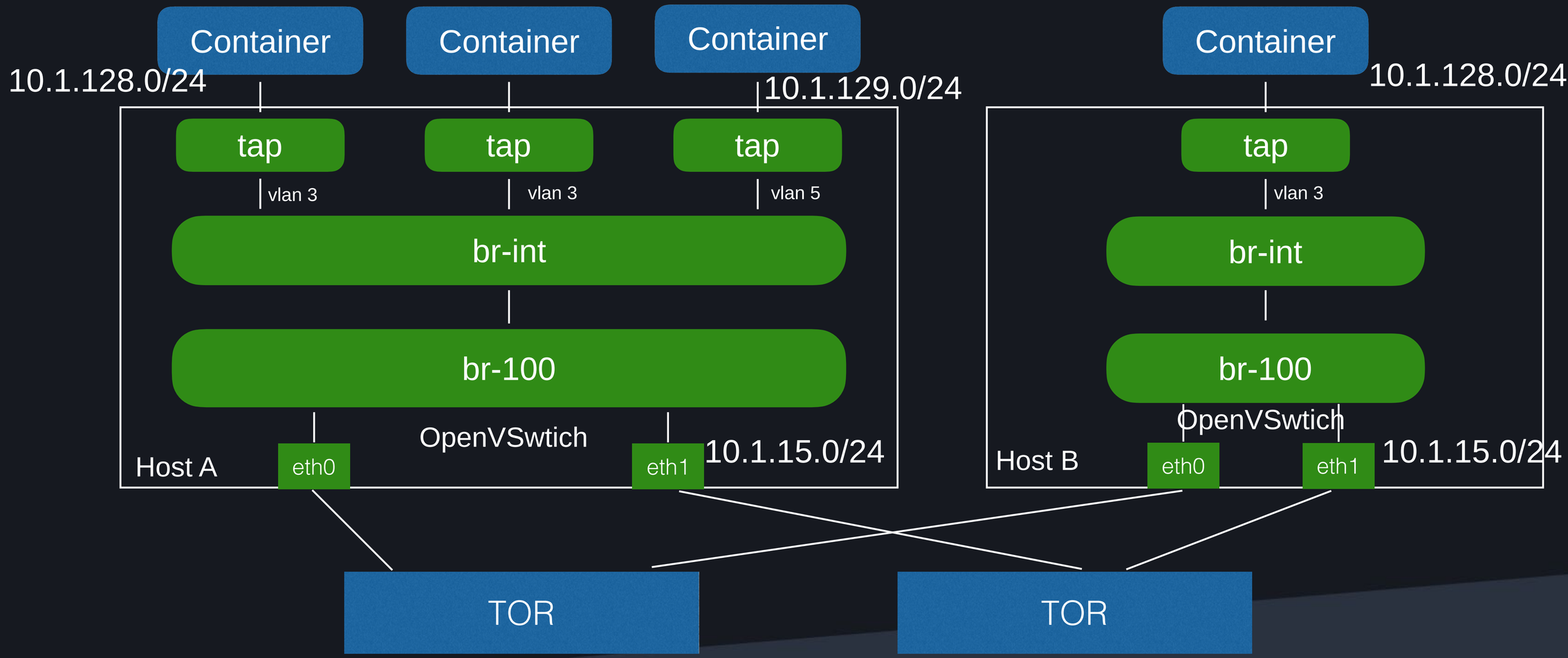


创建 Pod 的网络流程

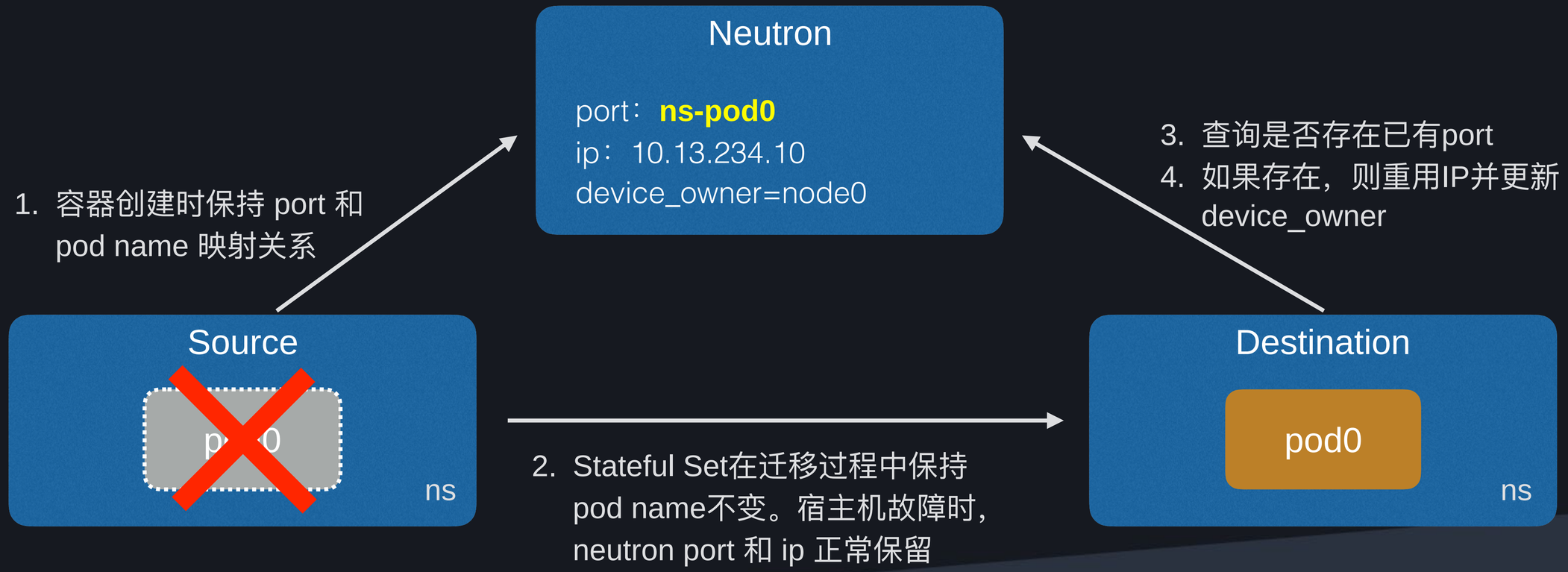
容器通过 veth 连到 OVS br-int 上，采用 VLAN 模型。兼具隔离性和性能。



OVS+Vlan


Container
Container
Container
10.1.128.0/24
10.1.129.0/24
tap
tap
tap
vlan 3
vlan 3
vlan 5
br-int
br-100
OpenVSwtich
Host A
eth0
eth1
10.1.15.0/24
Container
10.1.128.0/24
tap
vlan 3
br-int
br-100
OpenVSwtich
Host B
eth0
eth1
10.1.15.0/24
TOR
TOR

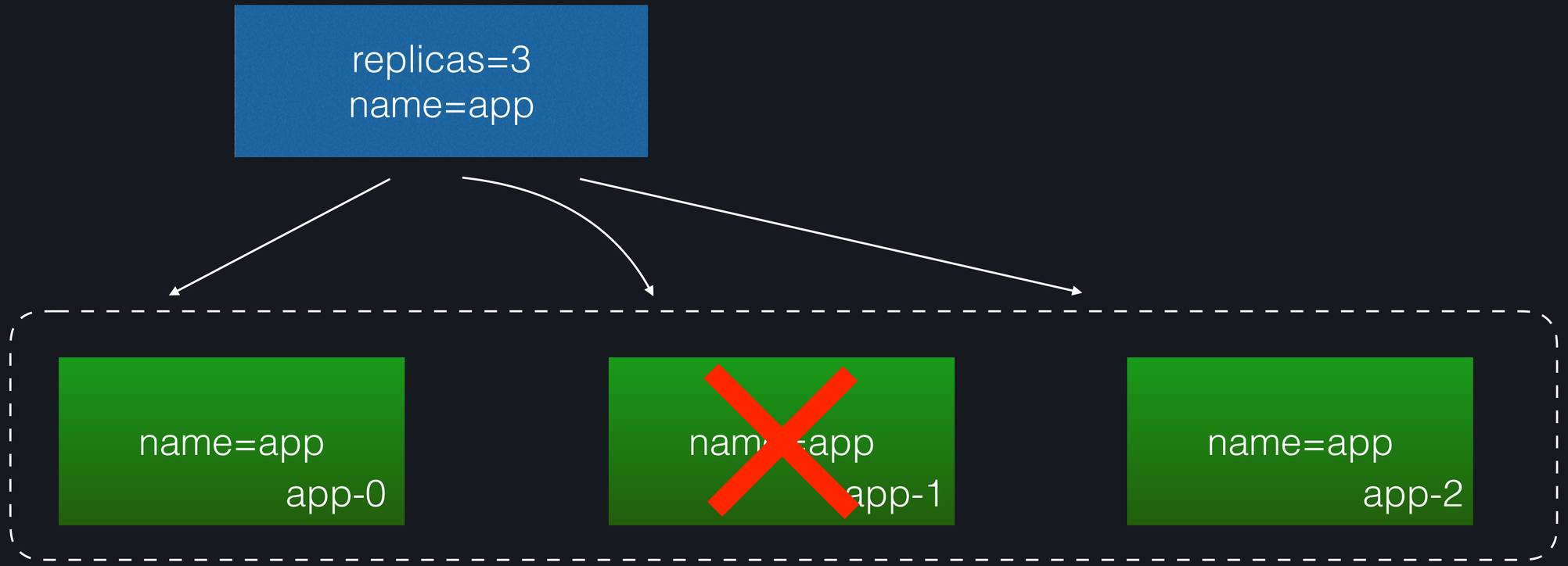
跨物理机迁移时容器 IP 保持不变



指定IP删除pod

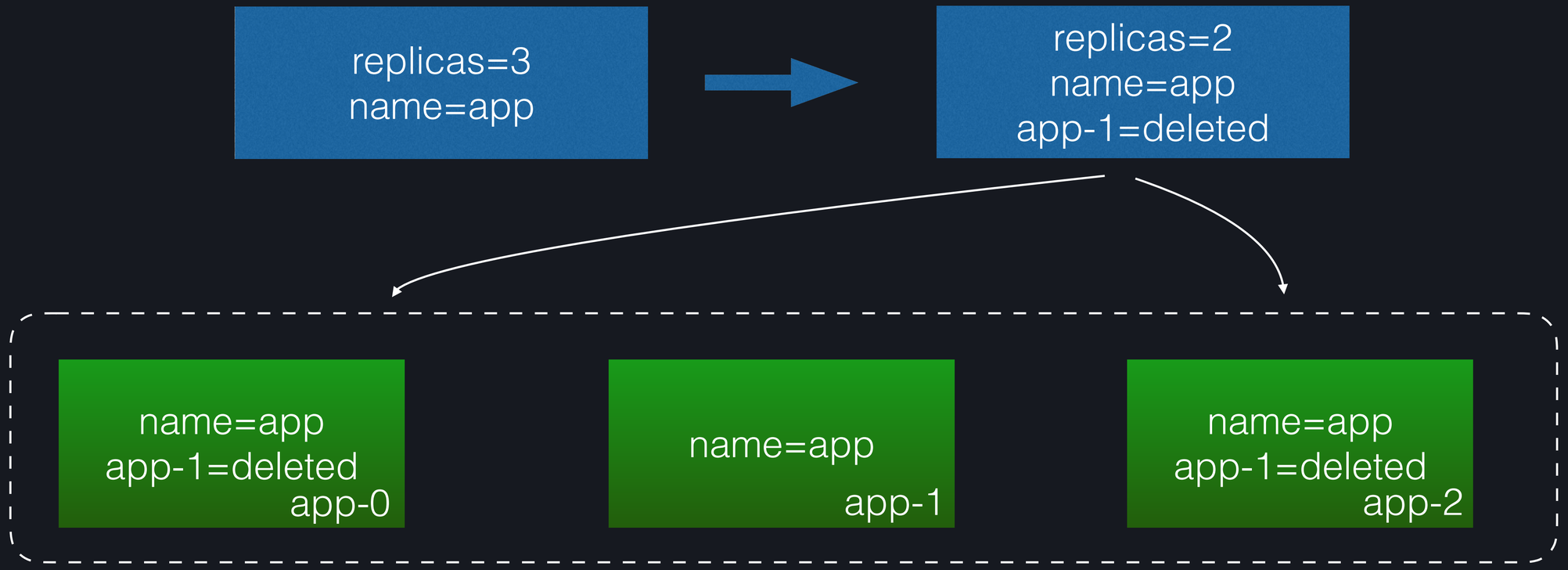
- statefulset只支持按序删除实例，不能满足所有业务场景。
- 业务有状态依赖关系，或当某些实例出现业务逻辑问题时，需要销毁。

指定IP删除pod



Statefulset不支持删除某个pod

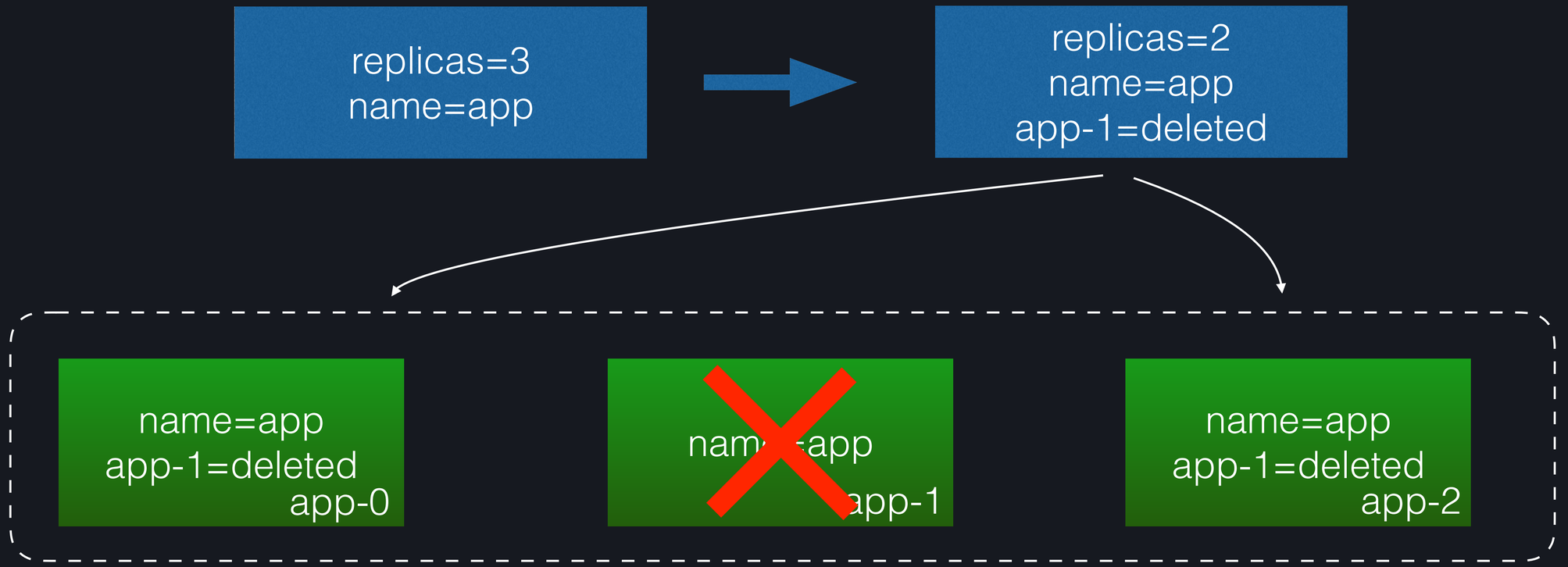
指定IP删除pod



Step 1. 添加一个app-1=deleted的label

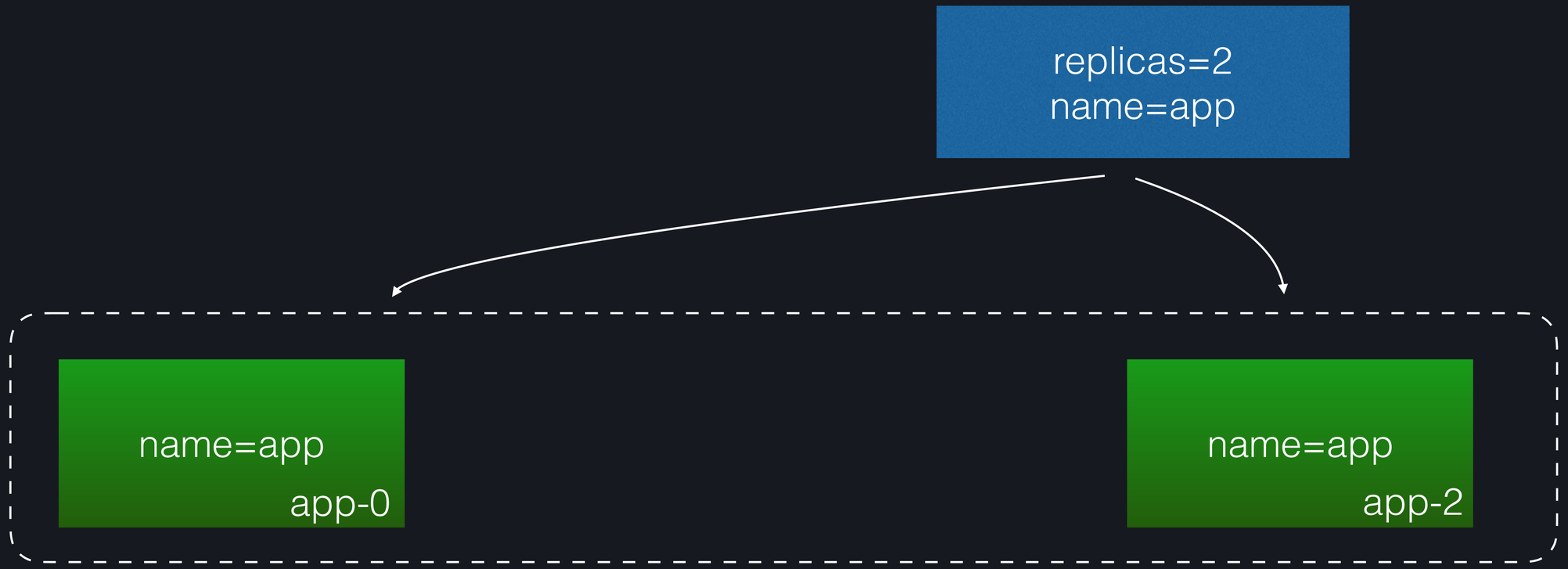
Step 2. 更新statefulset controller，也添加app1=deleted这个label，同时将replicas数量减1

指定IP删除pod



Step 3. 通过删除pod接口，将app-1这个pod删除

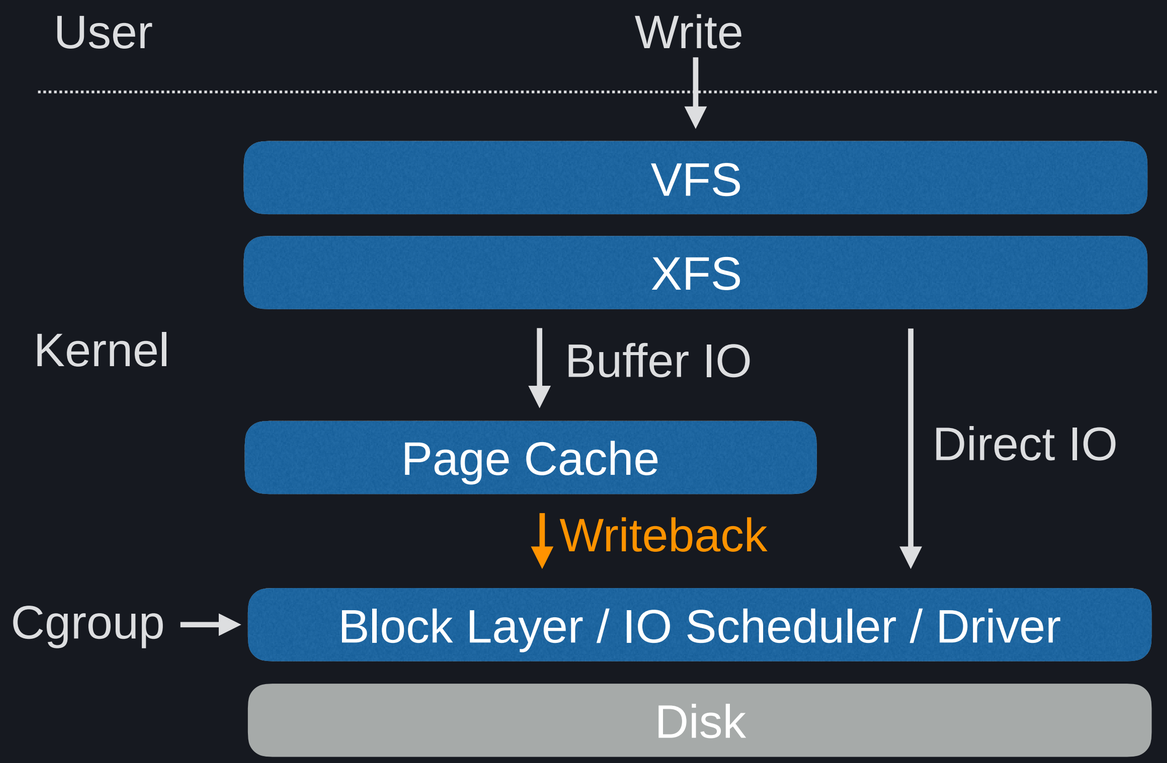
指定IP删除pod



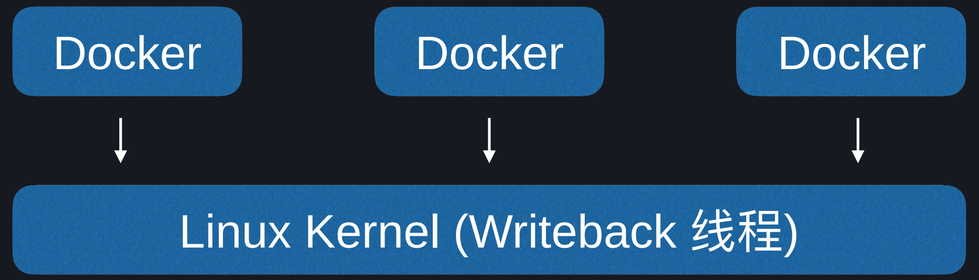
Step 4. 将app-1=deleted这个label清除掉

容器间异步 IO 隔离增强

存在什么问题?

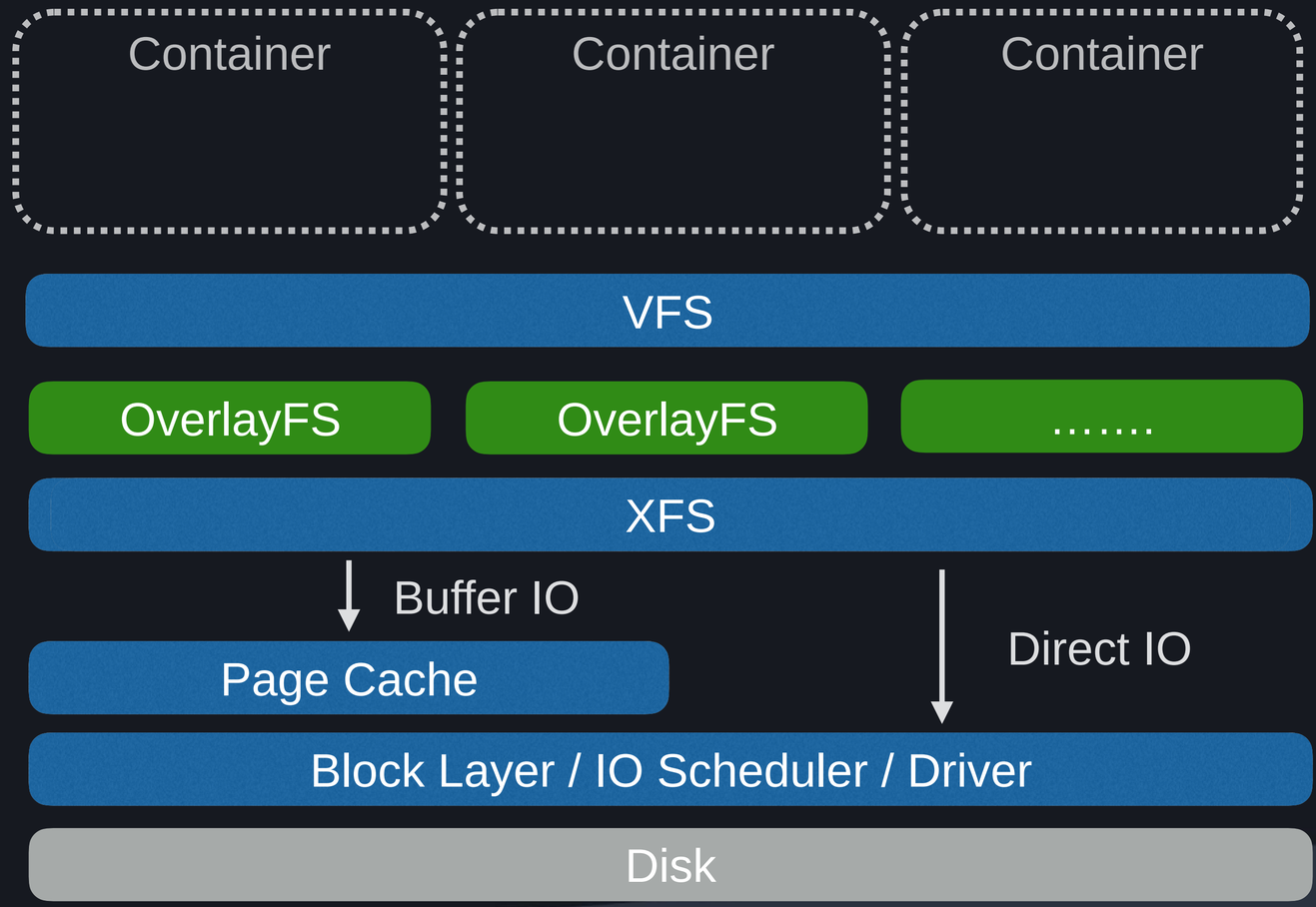


Cgroup V1 or V2？ 皆不尽完美!



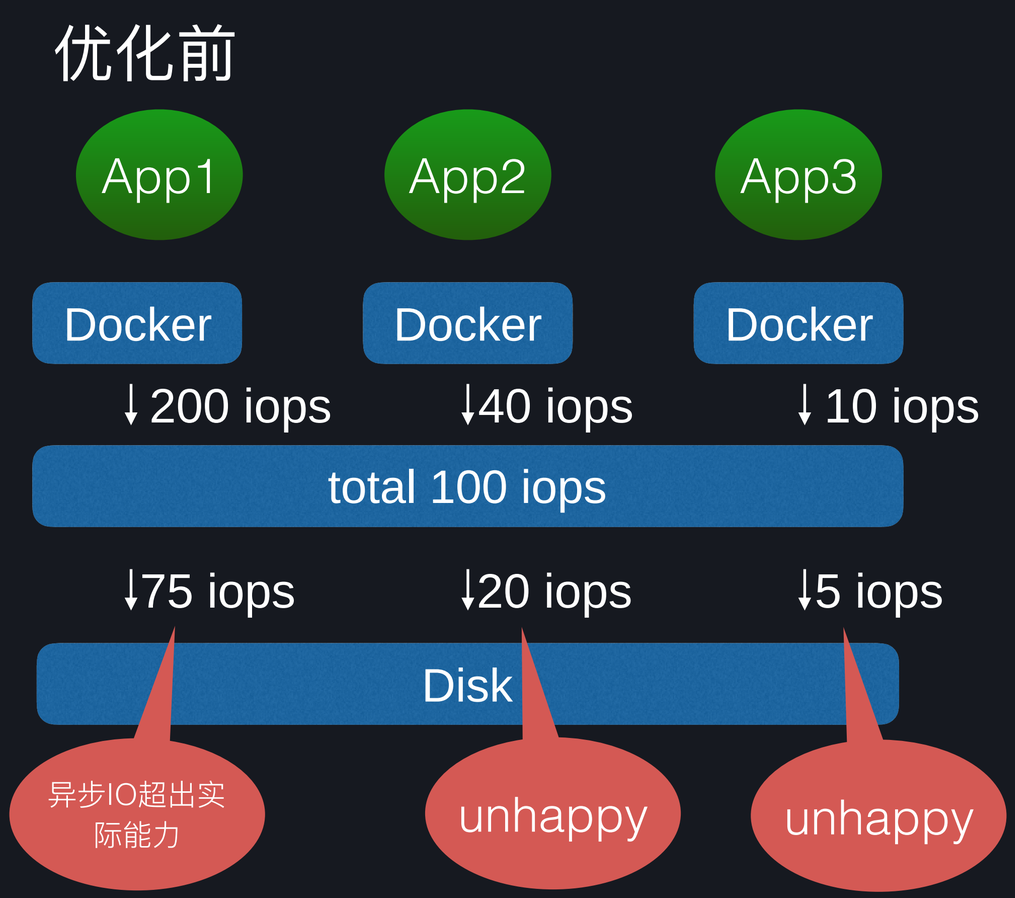
writeback线程无法区分脏数据是由哪个容器产生，无法做到隔离

基于 OverlayFS 的异步 IO 限速



每个容器有独立的 OverlayFS

容器间异步 IO 隔离


优化前
App1
App2
App3
Docker
Docker
Docker
200 iops
40 iops
10 iops
total 100 iops
75 iops
20 iops
5 iops
Disk
异步IO超出实际能力
unhappy
unhappy

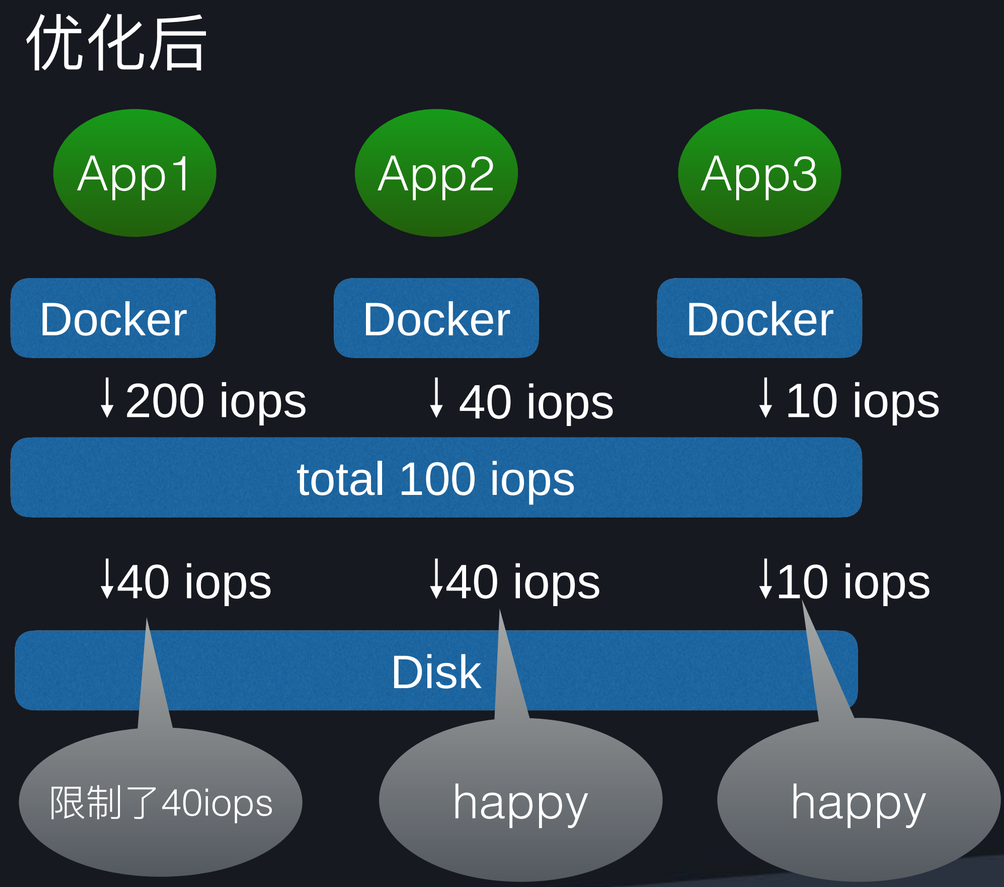

优化后
App1
App2
App3
Docker
Docker
Docker
200 iops
40 iops
10 iops
total 100 iops
40 iops
40 iops
10 iops
Disk
限制了40iops
happy
happy

TABLE OF

CONTENTS 大纲

1

美丽联合容器平台的演进

稳定 & 效率

2

技术方案

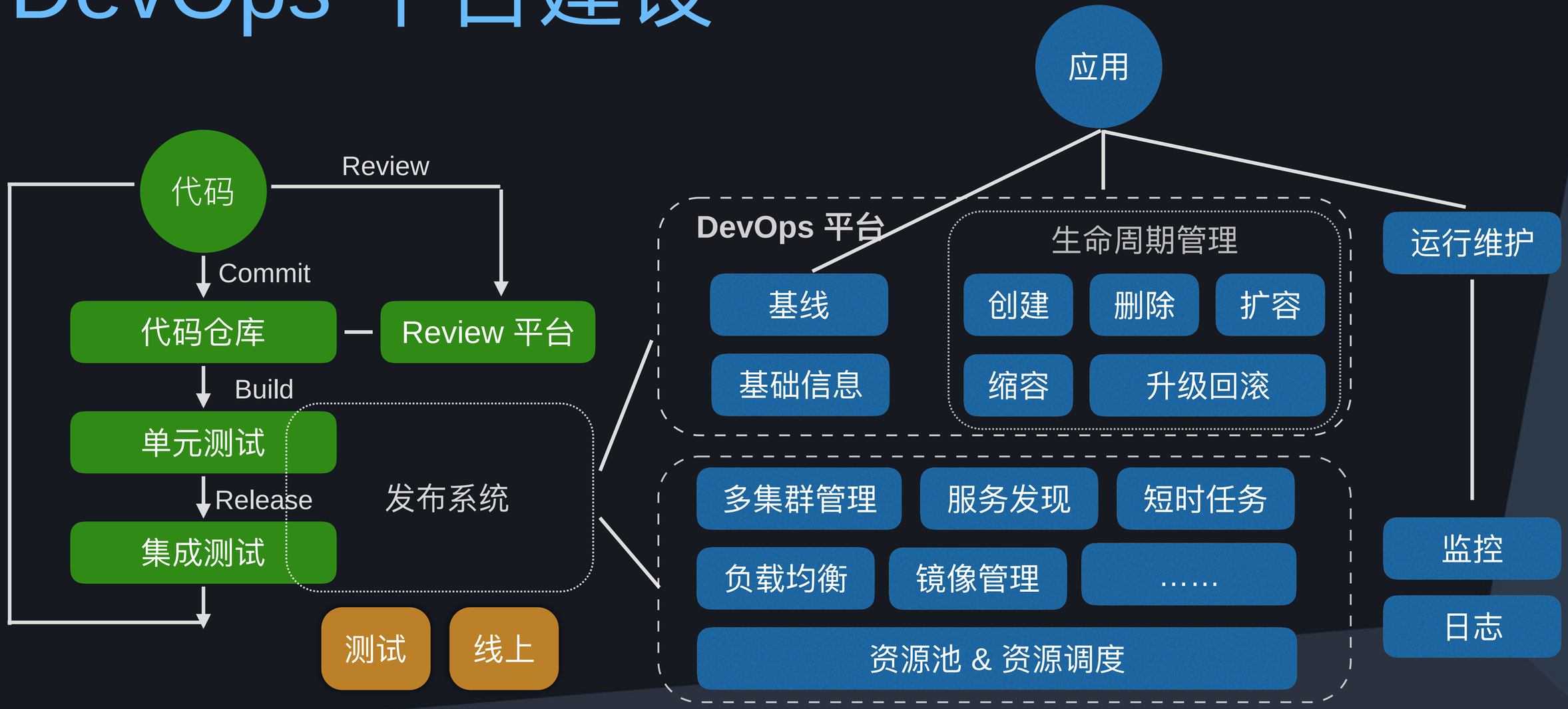
有状态业务 & 如何应对

3

DevOps 平台建设

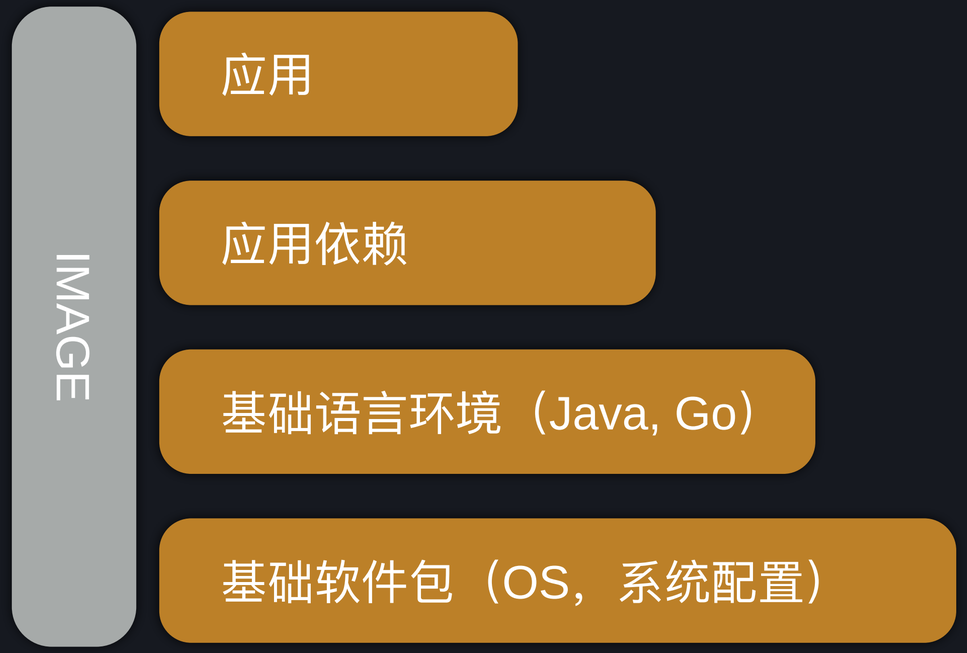
经验 & 体会

DevOps 平台建设

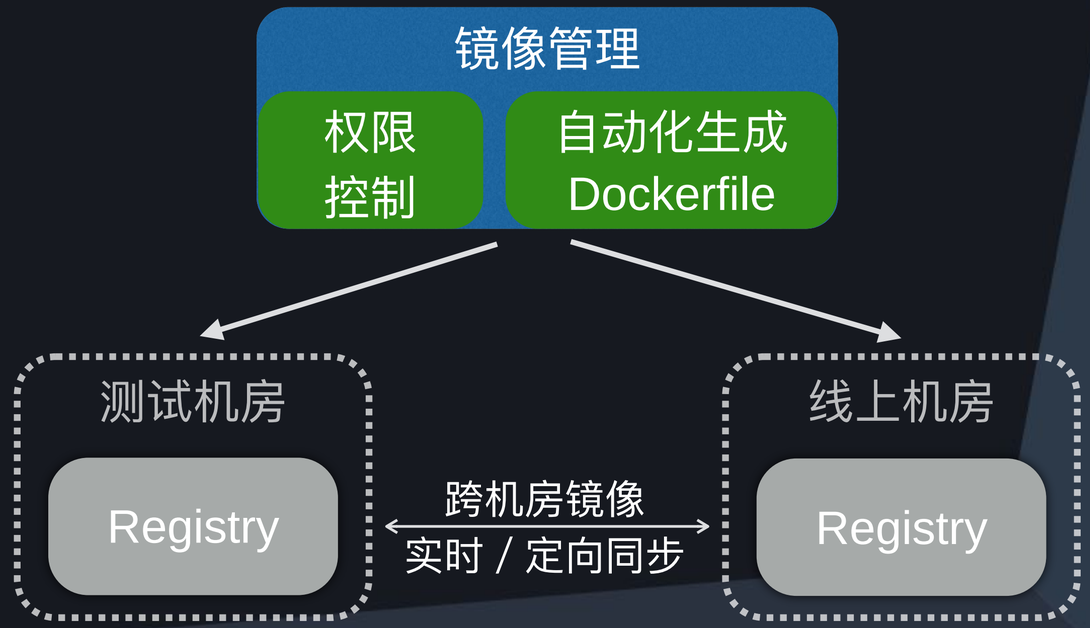

代码
Review
Commit
代码仓库
Review 平台
Build
单元测试
Release
集成测试
发布系统
测试
线上
应用
DevOps 平台
基线
基础信息
生命周期管理
创建
删除
扩容
缩容
升级回滚
运行维护
监控
日志
多集群管理
服务发现
短时任务
负载均衡
镜像管理
……
资源池 & 资源调度

镜像管理

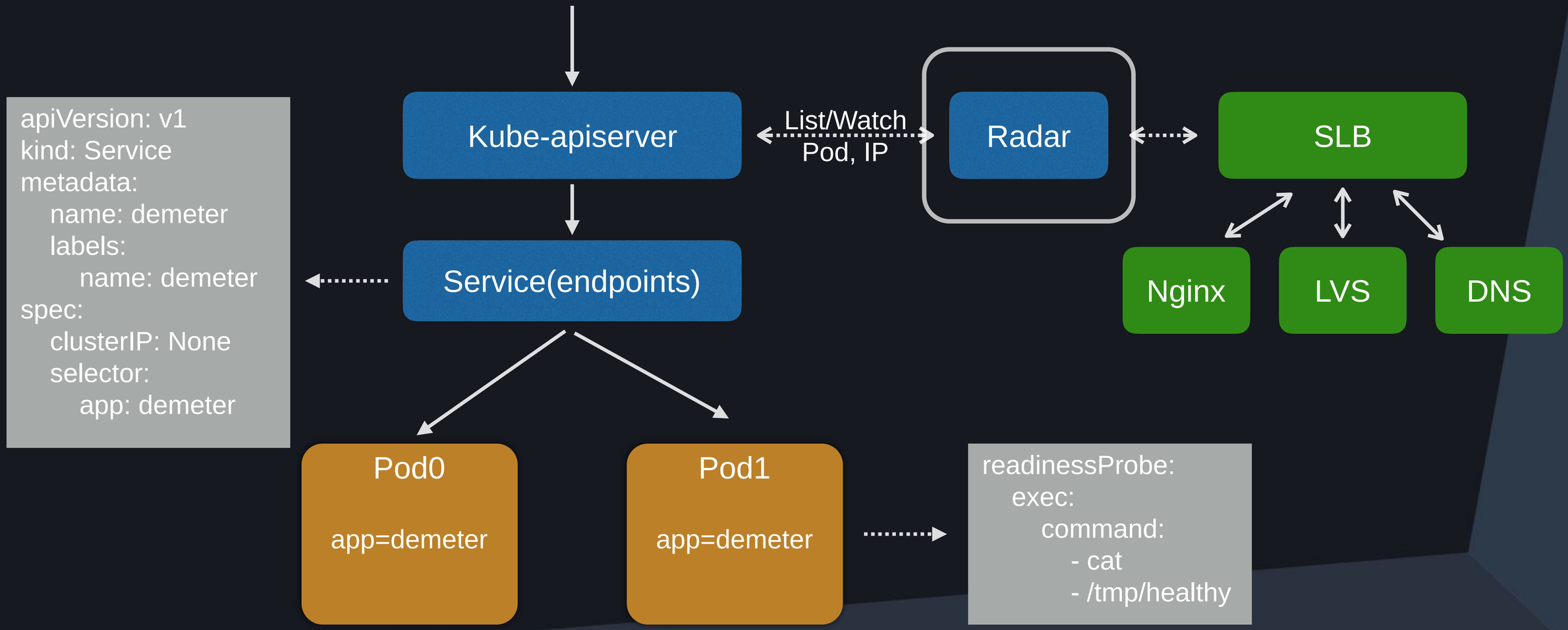
标准化的镜像分层方案



统一的镜像管理方案

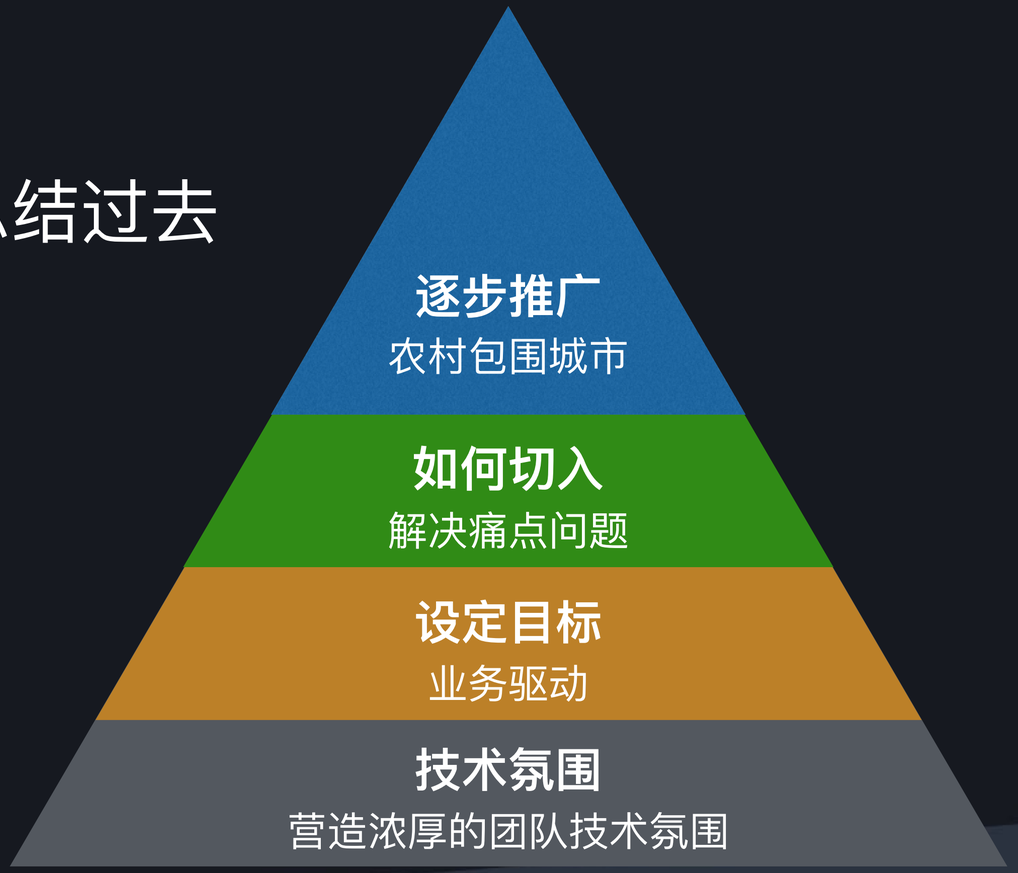


服务发现与健康检查


apiVersion: v1
kind: Service
metadata:
name: demeter
labels:
name: demeter
spec:
clusterIP: None
selector:
app: demeter
Kube-apiserver
List/Watch
Pod, IP
Radar
SLB
Nginx
LVS
DNS
Service(endpoints)
Pod0
app=demeter
Pod1
app=demeter
readinessProbe:
exec:
command:
- cat
- /tmp/healthy

总结 & 体会 & 未来

总结过去



展望未来

DevOps
微服务落地

在线离线
实例混部

For AI
GPU 资源调度

THANK YOU

如有需求，欢迎至 [讲师交流会议室] 与我们的讲师进一步交流

ArchSummit
全球架构师峰会 2017