

微服务化的不同阶段

Kubernetes 的不同玩法



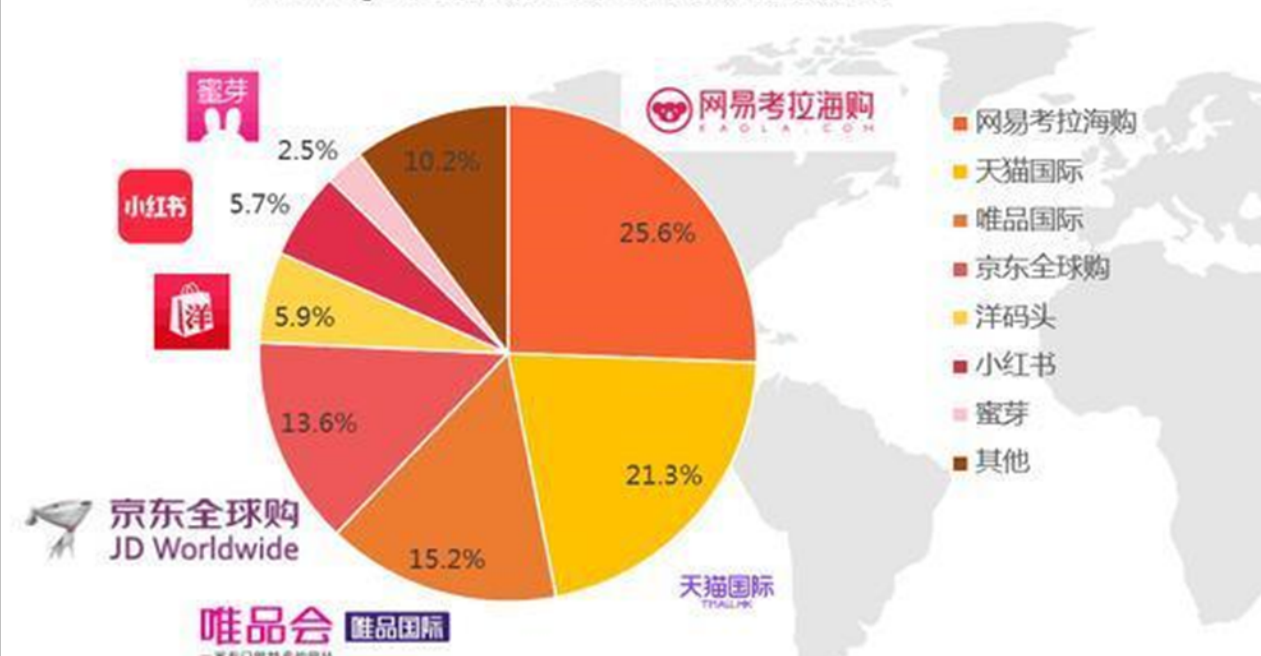
网易考拉海购

中国最大跨境电商平台之一



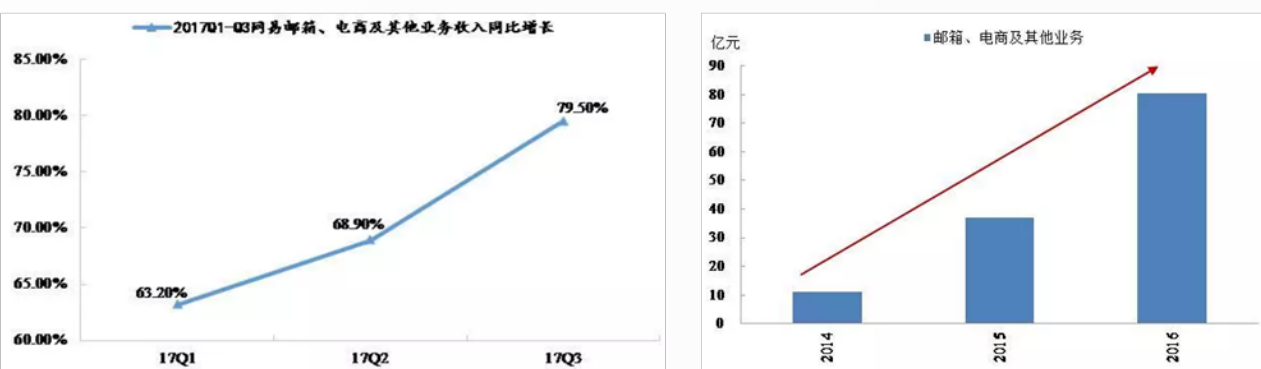
接连参战双十一、黑色星期五、双十二大促活动
连任全国多个主要跨境保税区销售额第一

2017Q3中国跨境电商平台市场份额分布



数据来源: iiMedia Research

数据来源: 网易公司公告

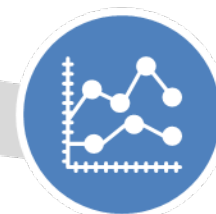


互联网产品的发展阶段



1、产品冷启动阶段

立项、快速上线、验证



2、产品高速增长阶段

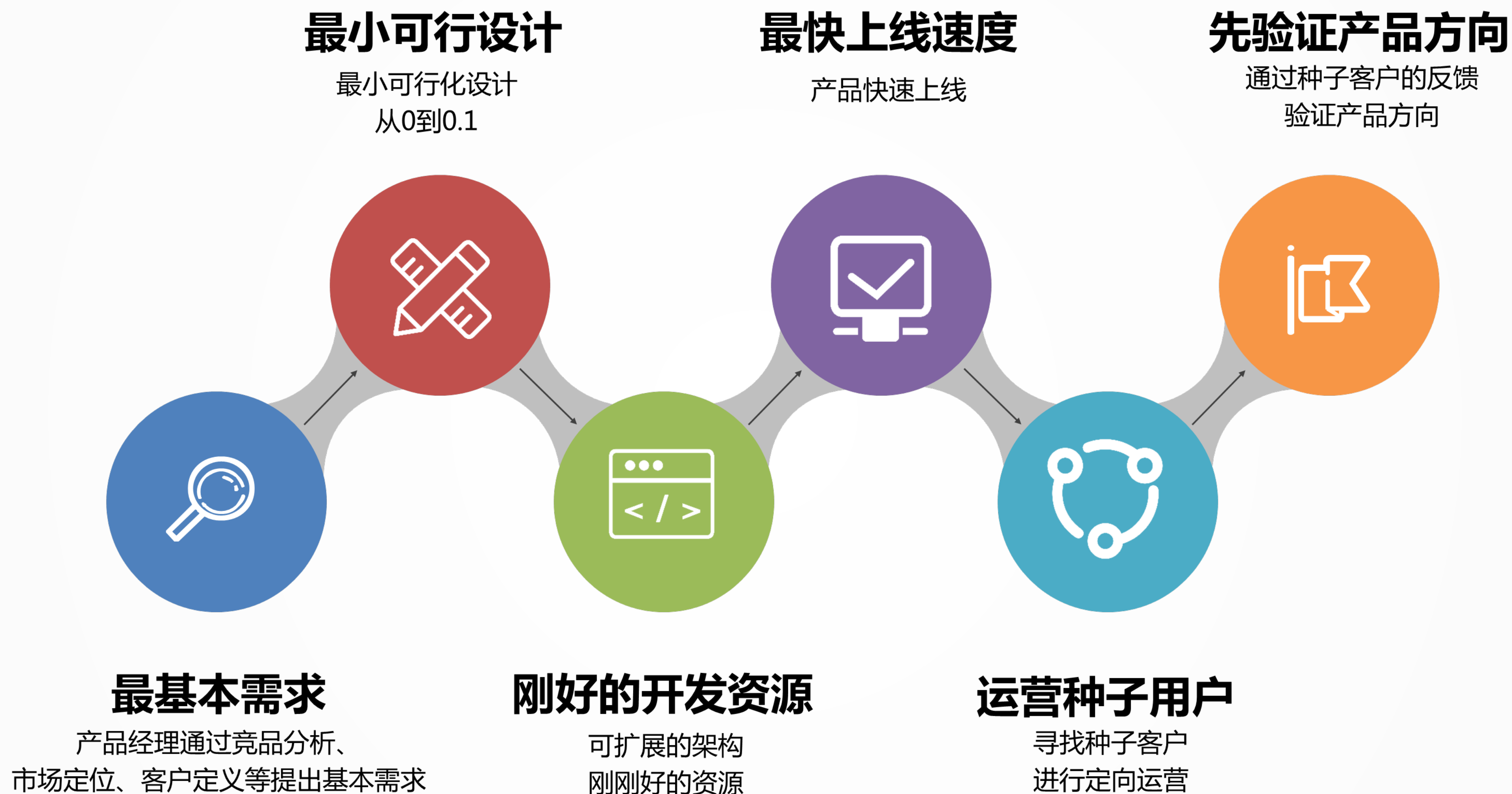
功能完善、快速迭代、营销获客



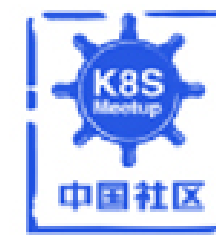
3、产品成熟阶段

精准营销、用户体验、业务扩展

冷启动阶段 - 以最快速度从需求到验证



考拉架构前身



Online:

前台服务集合，包含了所有的用户交互流程，和订单转账相关的定时任务，主要是WAP与WEB服务

CacheIndex:

缓存刷新，搜索索引创建，推荐资源创建

Offline:

推广拉新定时任务，物流爬取，CPS计算，第三方接口

Ms:

运营后台、进销存后台、配置管理



考拉第一个版本：快速上线



容器化的本质？

容器

我秒级启动

我一台机器上启动1000个nginx

我有镜像，原子升级回滚

我秒级自修复

我服务发现

我弹性伸缩

虚拟机

你没算应用吧，我也秒级

然并卵

说的谁好像没有似得

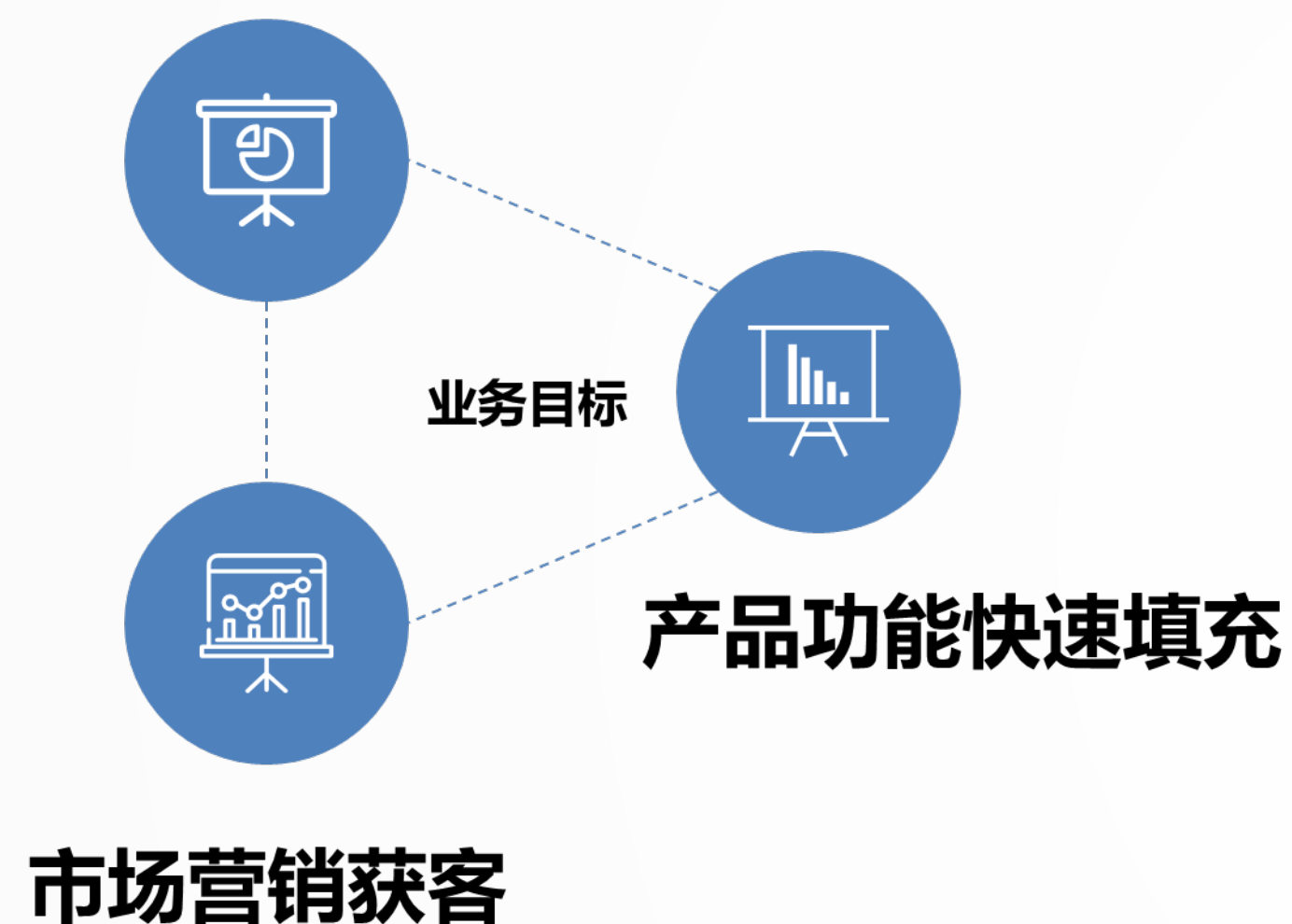
你确认这一秒没丢啥

Dubbo和springcloud笑而不语，熔断，降级

你听说过autoscaling group么？

快速增长阶段 – 业务目标与现实情况

保障核心服务的高可靠和高可用

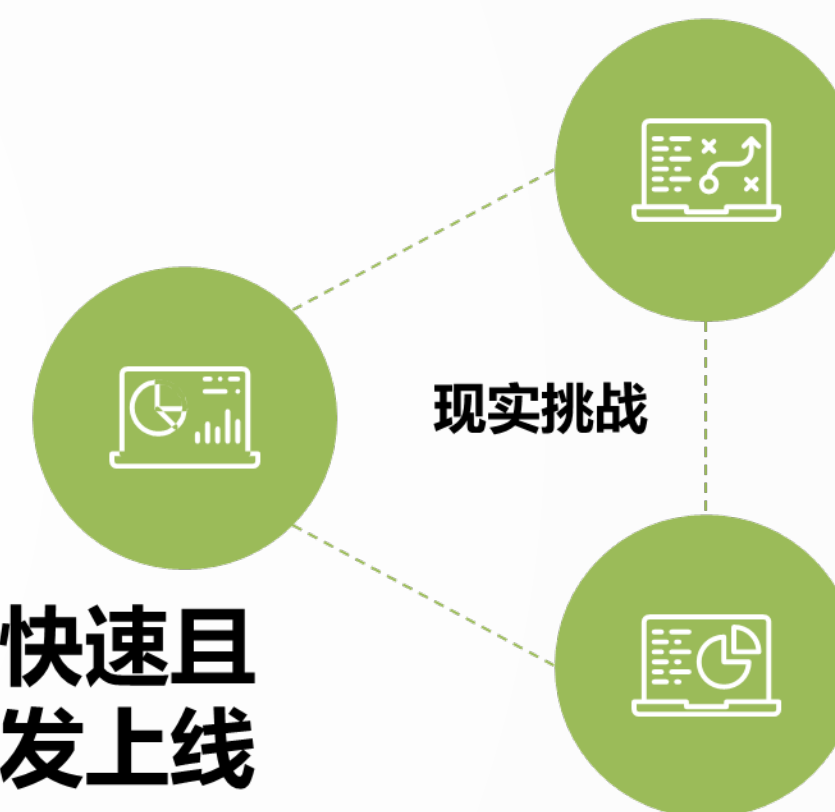


VS

不可控的高并发对可靠性和可用性产生较大压力

新功能无法快速且高质量地开发上线

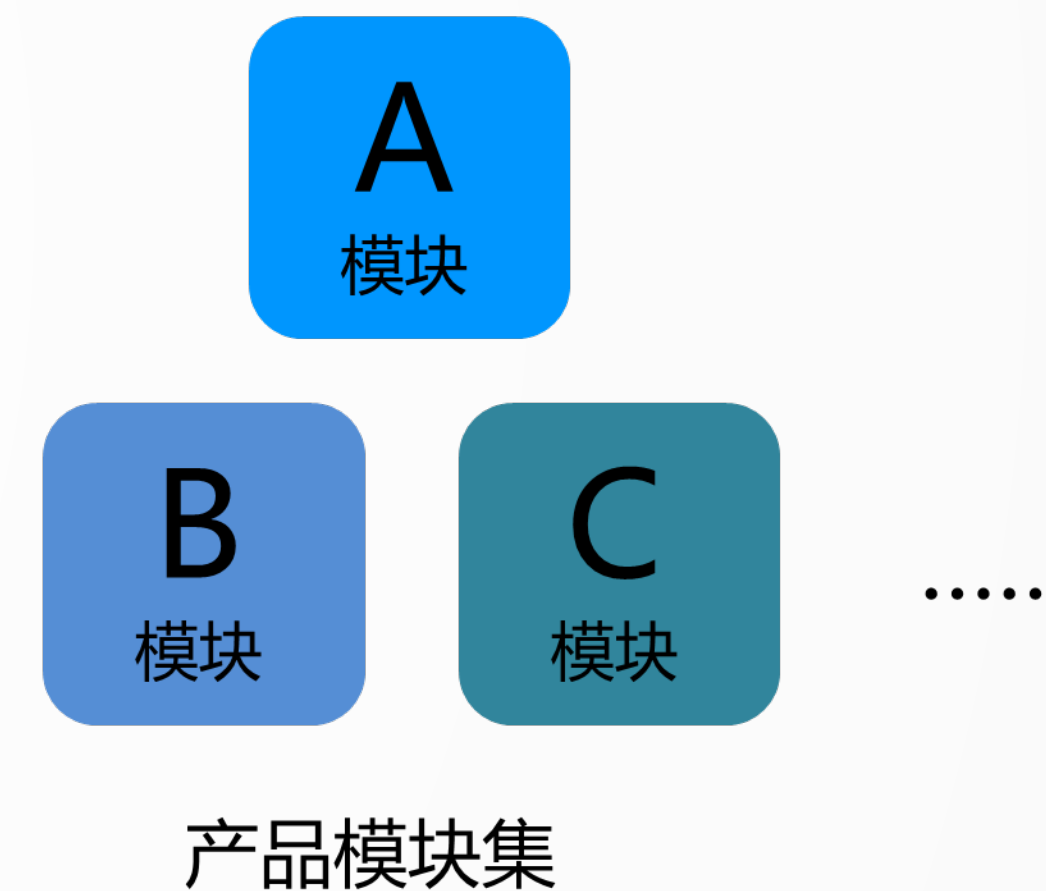
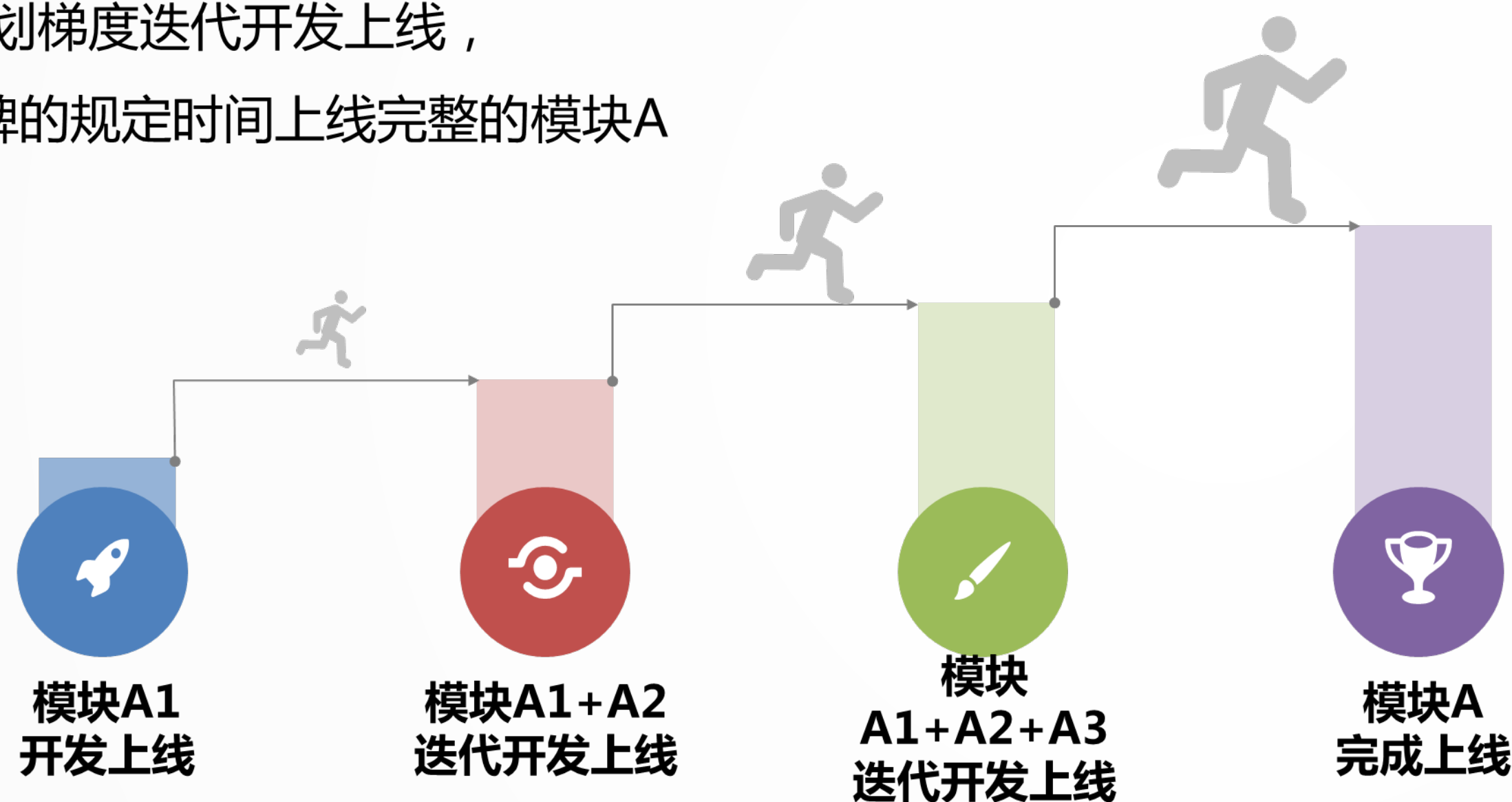
缺少专业技能团队
临时招聘来不及



快速成长阶段 – 拆分项目模块，实现快速迭代

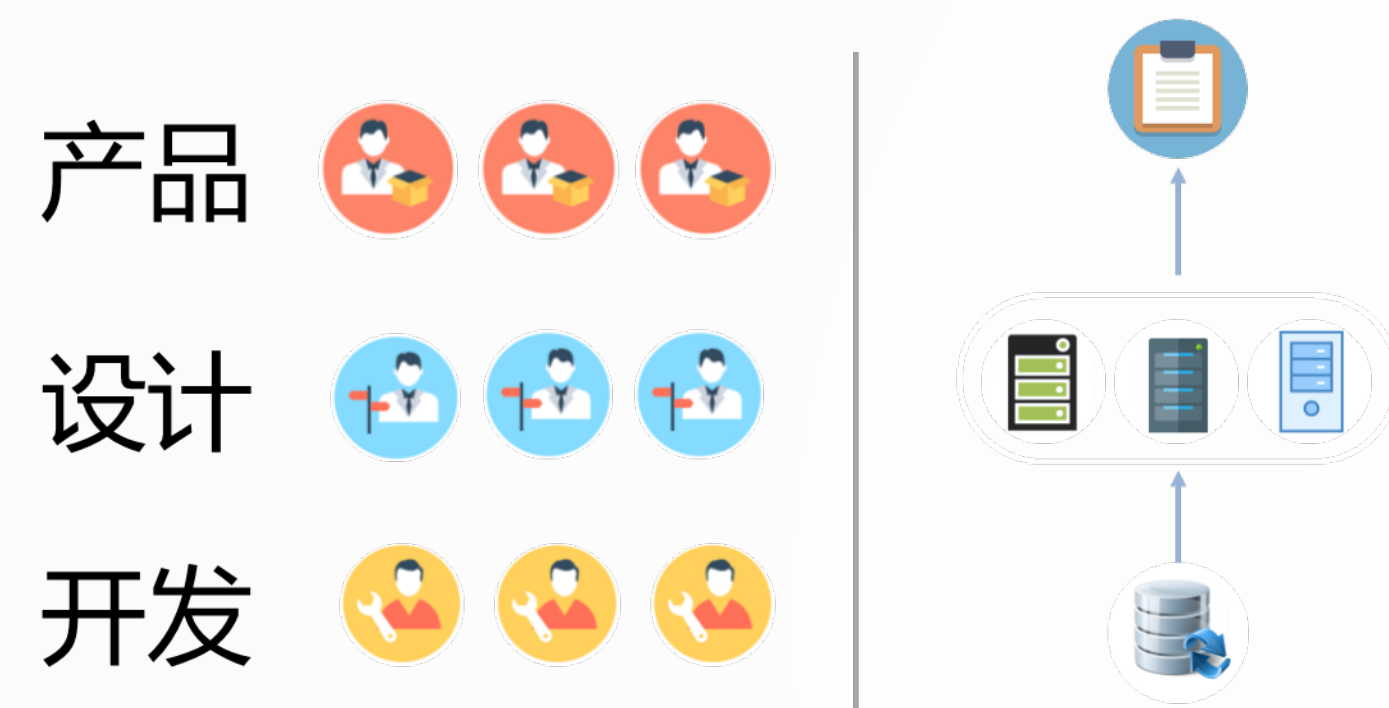
里程碑规划 + 迭代

1. 将模块A拆分为A1、A2、A3细粒度功能
2. 按计划梯度迭代开发上线，
3. 最后在里程碑的规定时间上线完整的模块A



如何做到快速迭代（1/2）-向微服务架构演进

根据系统层次划分不同的职能团队



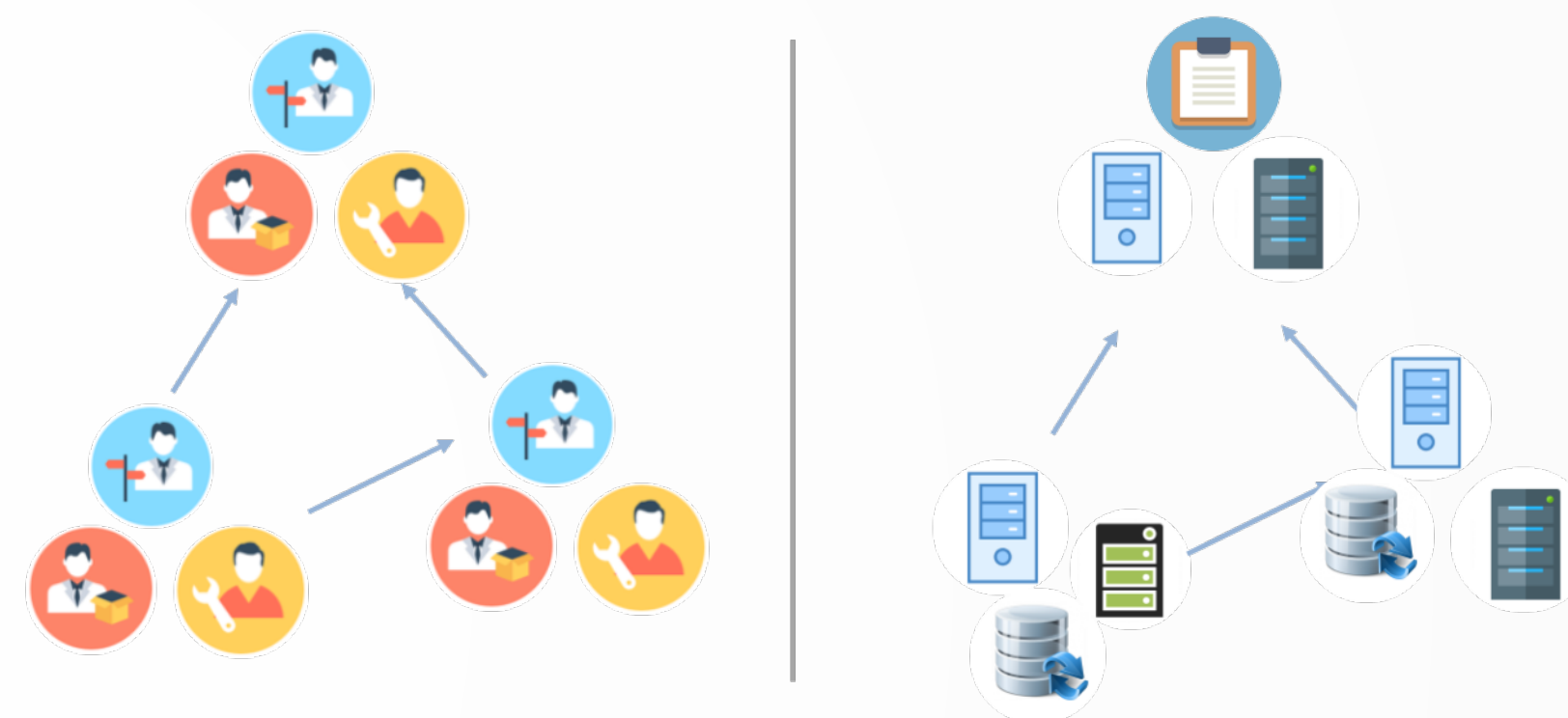
组织层面

一个层次的变更会涉及其他层的团队，需要跨团队沟通

技术层面

不同层次迭代速度不同，进度、变更相互影响，维护成本很高

按照业务的功能划分
单一业务功能对应一个独立的职能团队



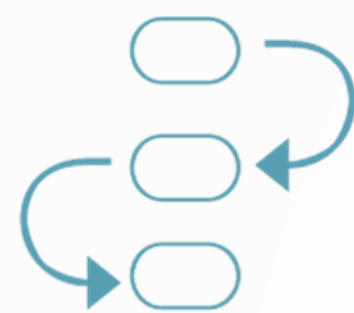
组织层面

服务内部变更仅发生在团队内

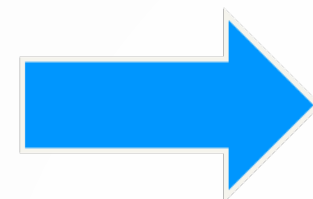
技术层面

各个服务模块对外提供接口，内部迭代速度、变化不影响其他服务模块，维护成本低

考拉：快速增长阶段



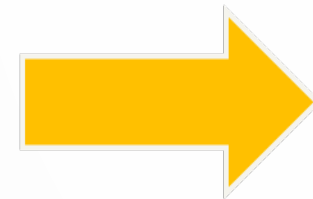
自动化部署工具实现一键式生成环境、触发测试和发布



- ✓ 变更的频率由每天2次增长到每天1000次+
- ✓ 软件产品质量提升52%



全链路的性能监控体系

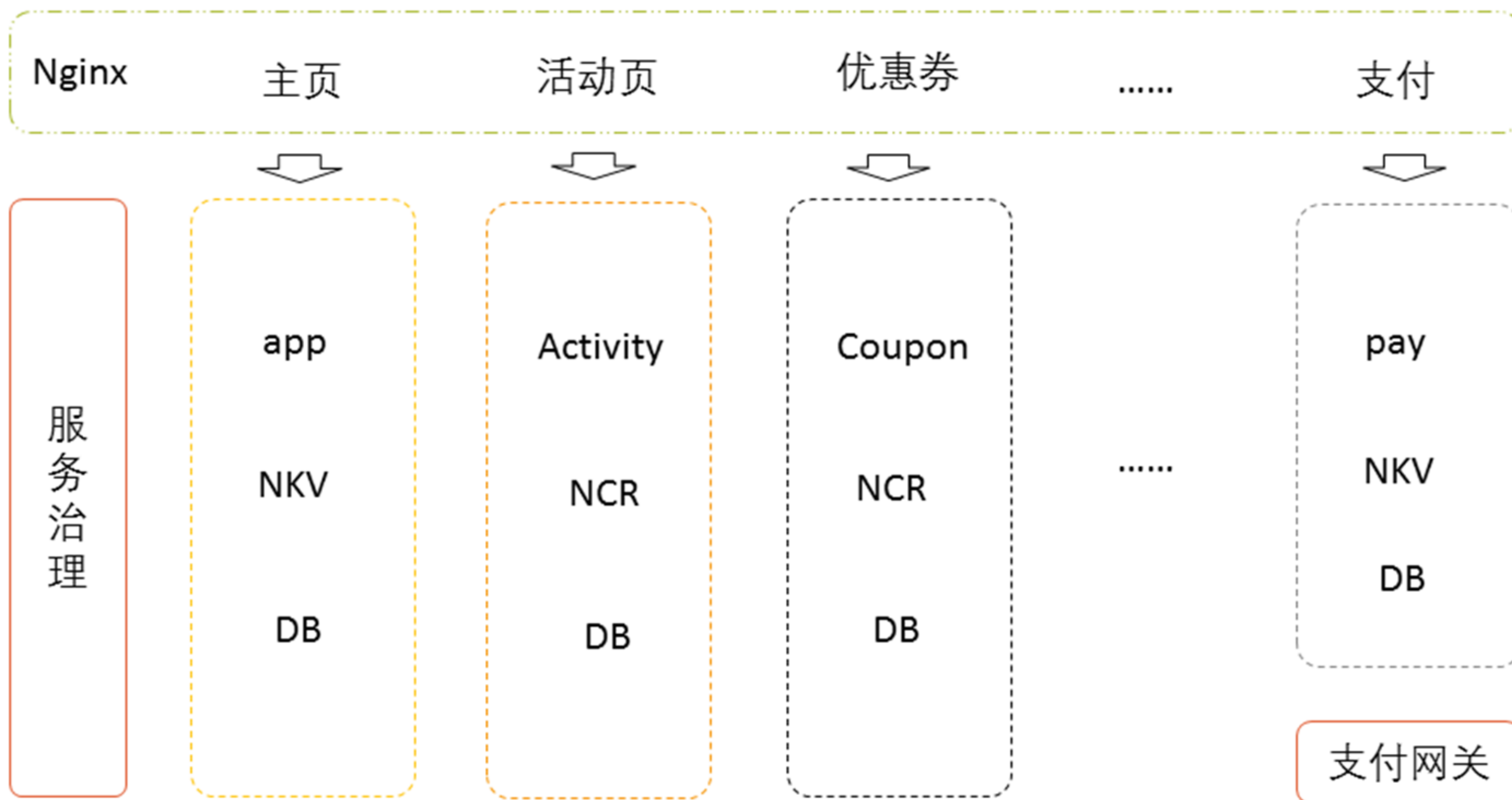


- ✓ 大促前压测，加入购物车超过1秒
- ✓ 通过回溯整个链路的调用过程，发现数据库操作环节的时延占整个时延的95%以上
- ✓ 快速优化数据库操作逻辑，平稳应对大促

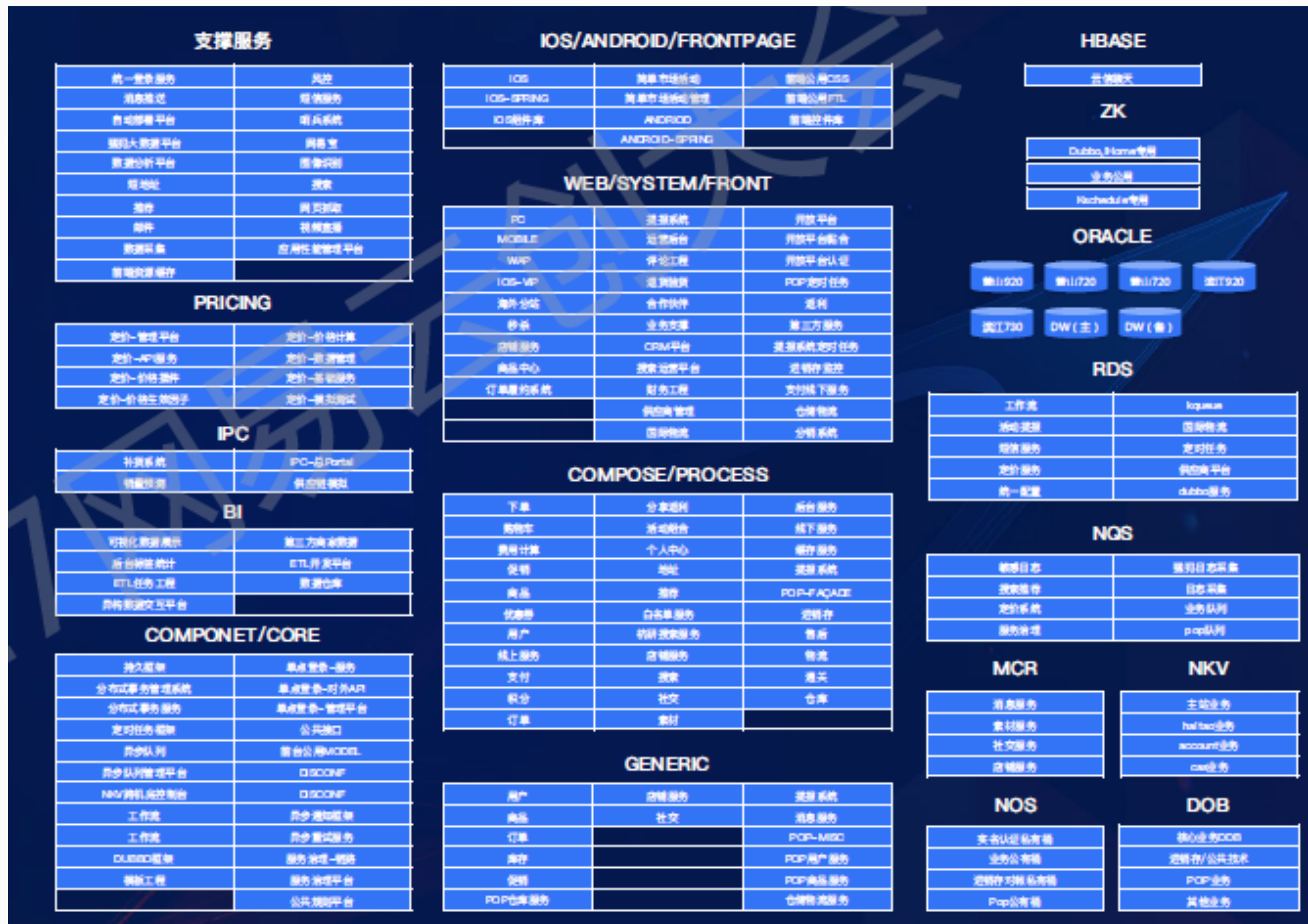
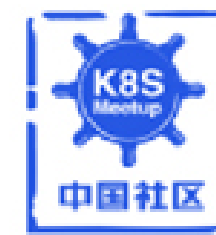
考拉服务化架构



考拉服务化架构



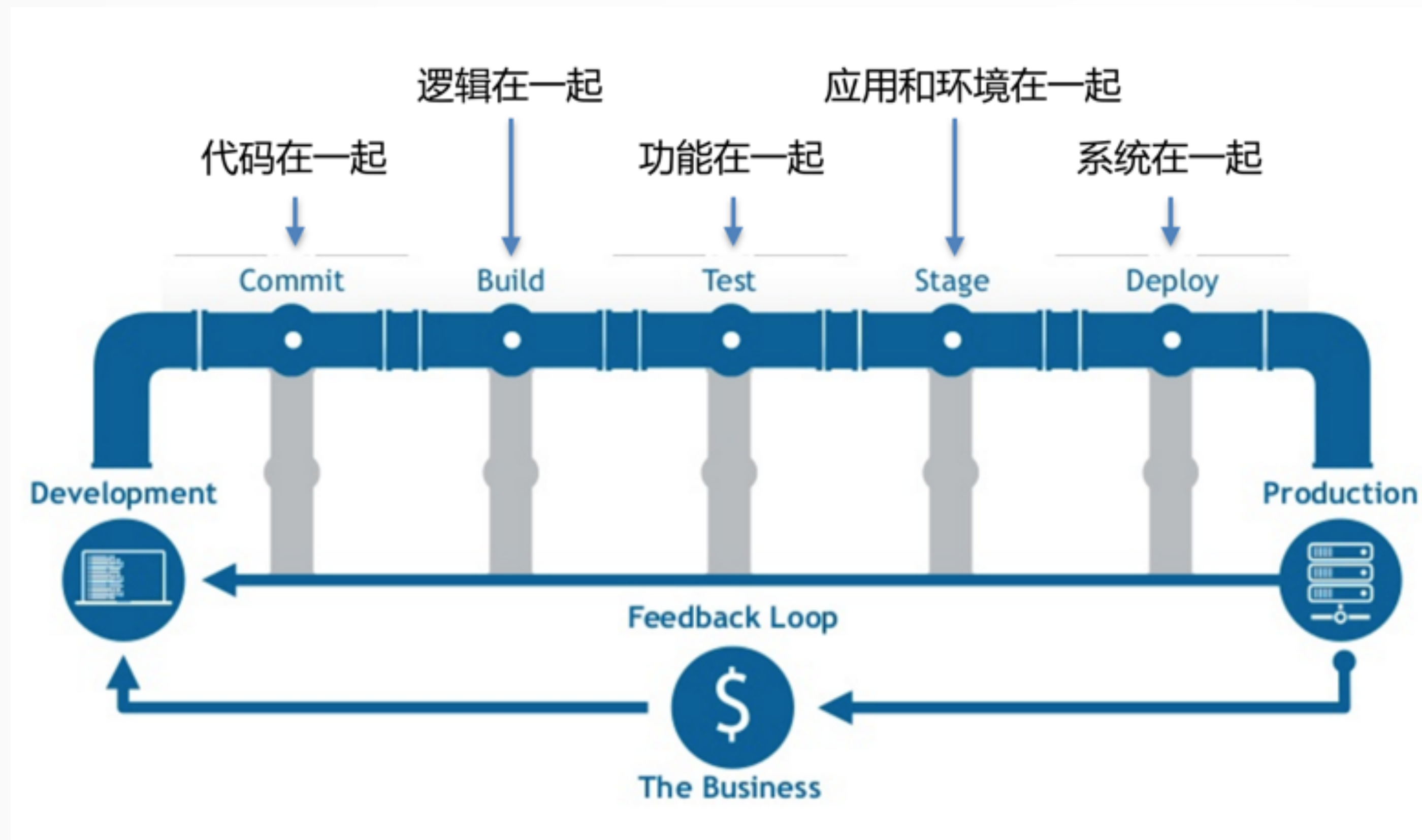
考拉服务化架构——不断演进中



拆之前应该解决合的问题

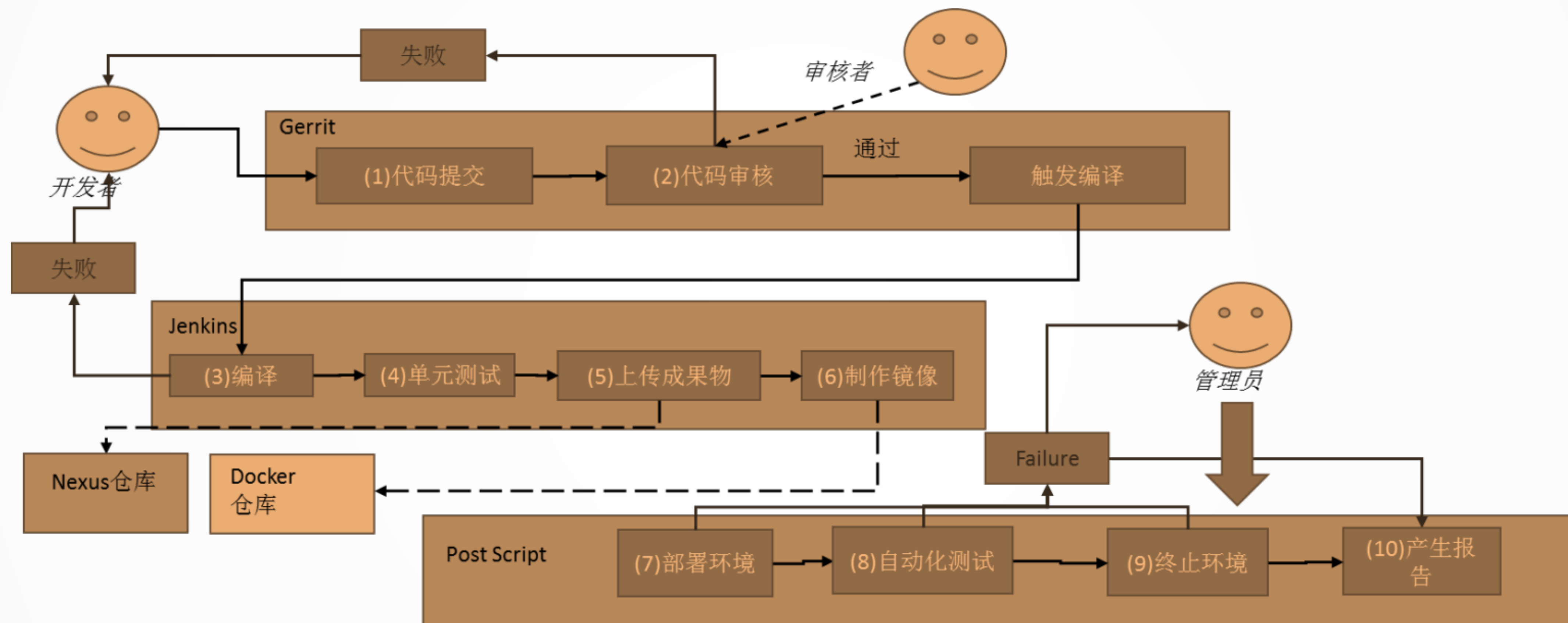
- 如何控制拆的风险
- 如何保证代码质量
- 如何保证功能不变，不引入新的Bug

集成，并且持续集成

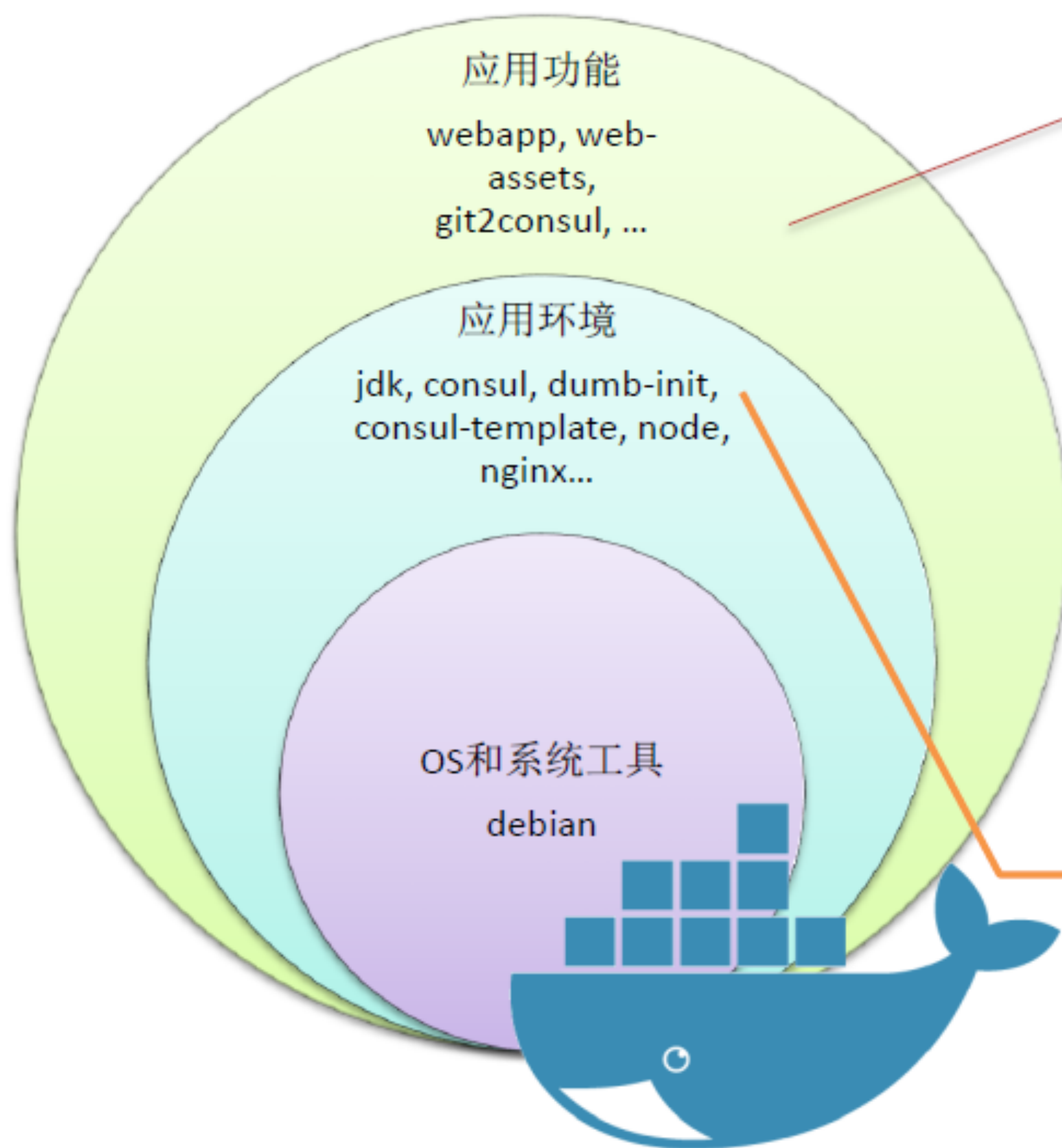


- **在一起**就是集成，每次集成都应该有**反馈**。
- 只有不停的集成才是**持续集成**。越少持续，每次反馈**代价越大**。
- 交付：就是将最终的产品发布到线上/生产环境，给用户使用。
- 多次集成产生一次**交付**

开发流程



容器镜像的组织



Dockerfile 104 Bytes

```
1 FROM cloud-app
2 MAINTAINER wyzc <wyzc@service.metease.com>
3
4 add target/zc-payment.jar app.jar
```

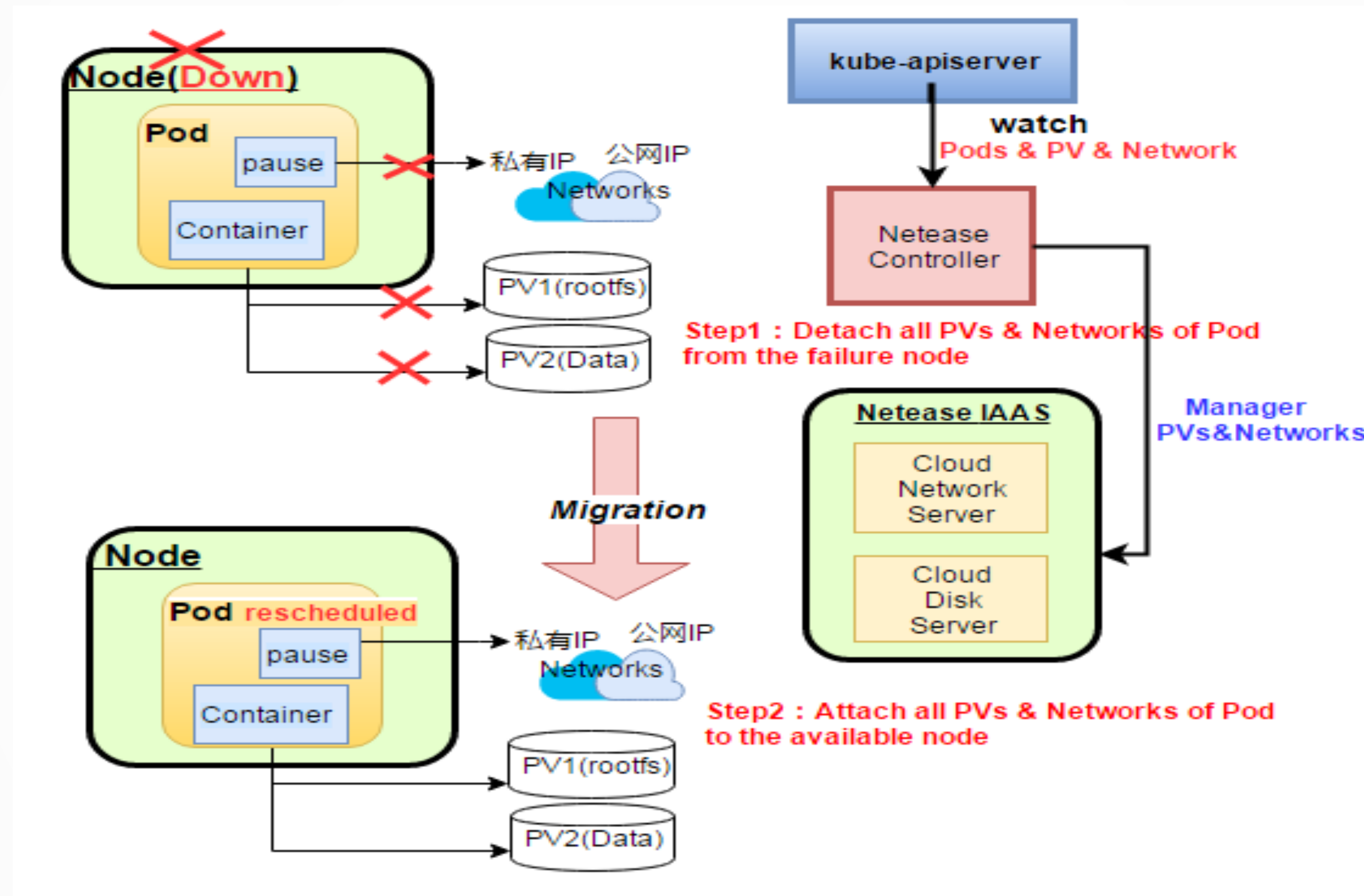
Dockerfile 121 Bytes

```
1 FROM cloud-nginx
2 LABEL BUILD_NAME=nginx-cdn
3
4 ADD docker/nginx-cdn-sites.ctmpl nginx.d/sites/
5 ADD dist nginx.d/cdn-assets
```

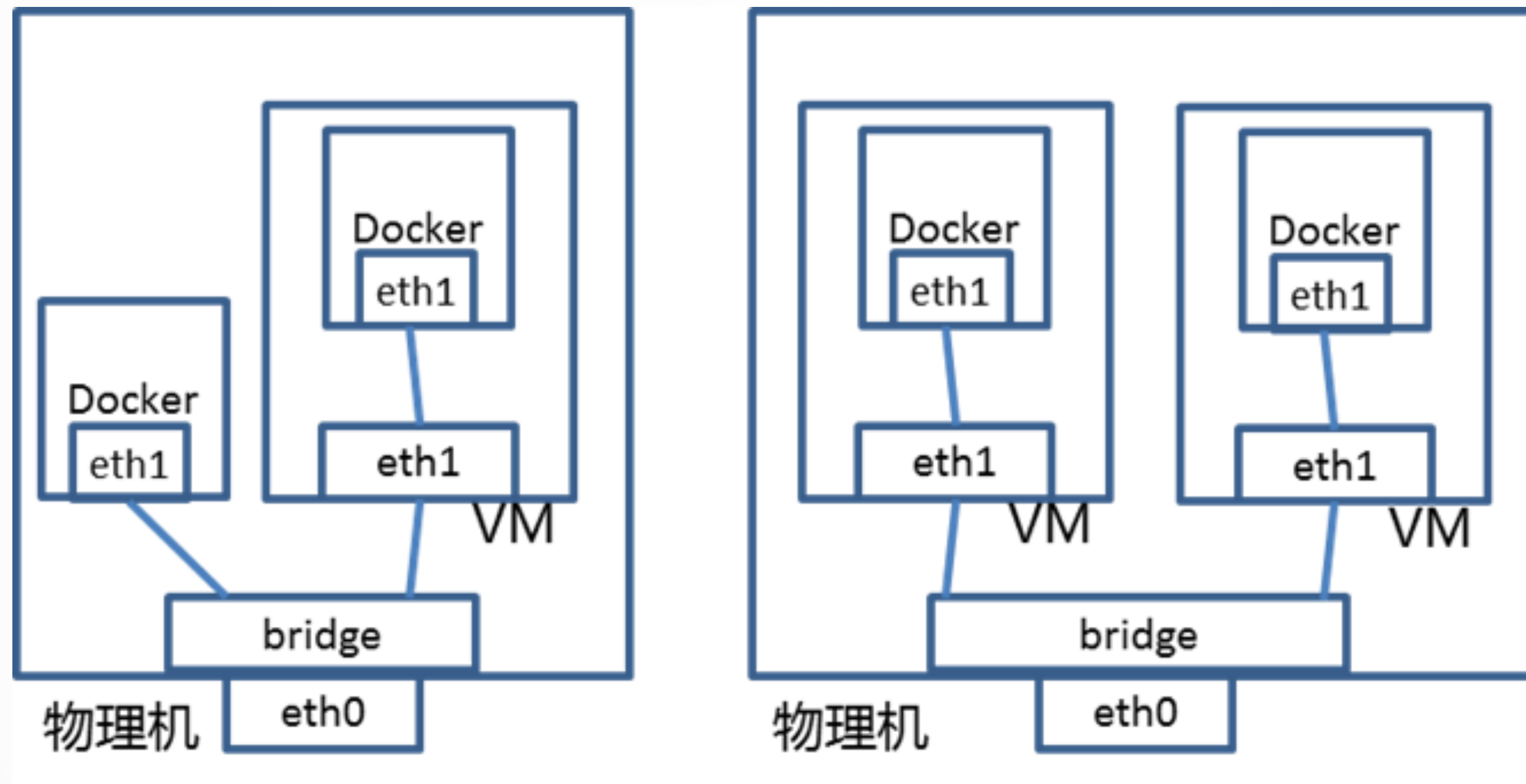
Yelp/dumb-init

1. **多进程支持:** 代理信号及接管子进程
2. **Signal rewriting:** `dumb-init --rewrite 15:2 xxx-app`

场景二：Kubernetes + Docker + VM + Host Netwo



场景三：Kubernetes + Docker + PM + Bridge Netw



Kubernetes配置网络

• 手动配置Docker网络

`docker inspect '--format={{ .State.Pid }}' test` 获取容器的PID，是namespace的名字

DOCKERPID=12065

NSPID=12065

默认Docker创建的网络namespace不在默认路径下，ip netns看不到，所以需要ln一下

`rm -f /var/run/netns/12065`

`ln -s /proc/12065/ns/net /var/run/netns/12065`

LOCAL_IFNAME=veth3pl12065 在宿主机上的网卡

GUEST_IFNAME=veth3pg12065 在容器内的网卡

创建网卡对

`ip link add name veth3pl12065 mtu 1500 type veth peer name veth3pg12065 mtu 1500`

将主机上的网卡连接到网桥上

`ip link set veth3pl12065 master testbr`

`ip link set veth3pl12065 up`

将容器的网卡塞到容器的namespace里面

`ip link set veth3pg12065 netns 12065`

将容器内网卡重命名

`ip netns exec 12065 ip link set veth3pg12065 name eth3`

给容器内网卡设置ip地址

`ip netns exec 12065 ip addr add 192.168.0.2/24 dev eth3`

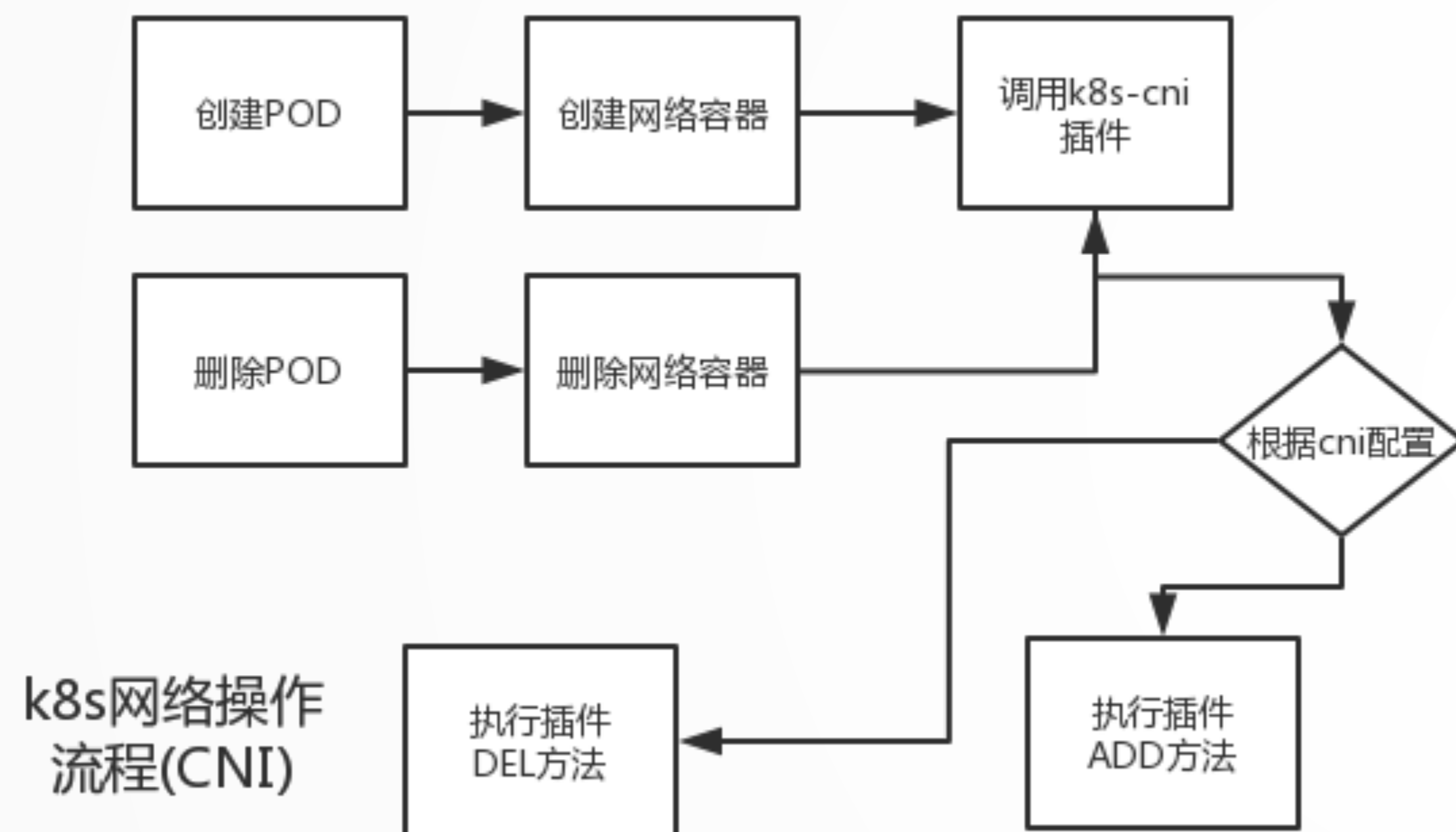
`ip netns exec 12065 ip link set eth3 up`

发送arp

`ip netns exec 12065 arping -c 1 -A -I eth3 192.168.0.2`

Kubernetes配置网络

• CNI



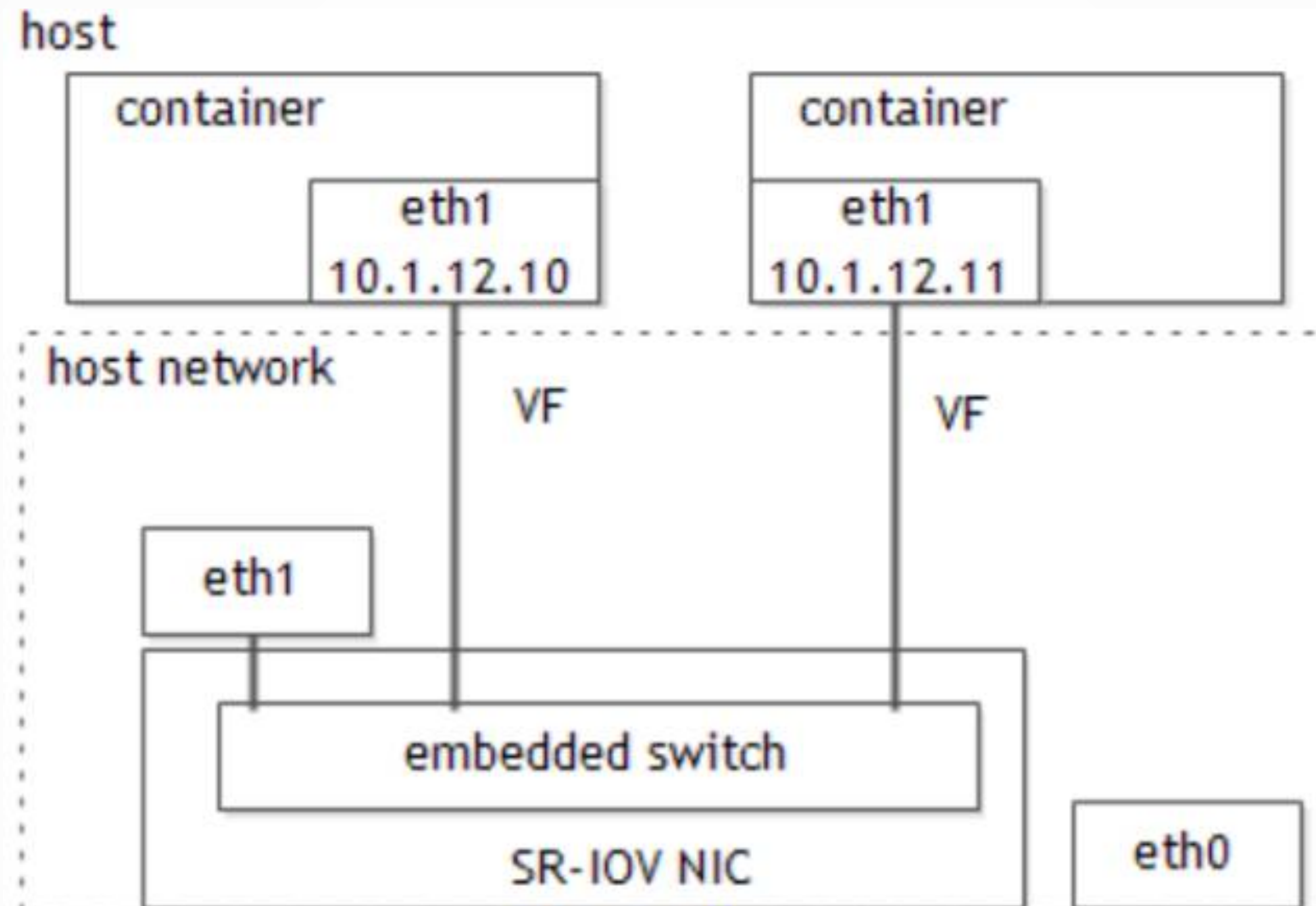
```

{
  "name": "mynet",
  "type": "bridge",
  "bridge": "mynet0",
  "isDefaultGateway": true,
  "forceAddress": false,
  "ipMasq": true,
  "hairpinMode": true,
  "ipam": {
    "type": "host-local",
    "subnet": "10.10.0.0/16"
  }
}
  
```

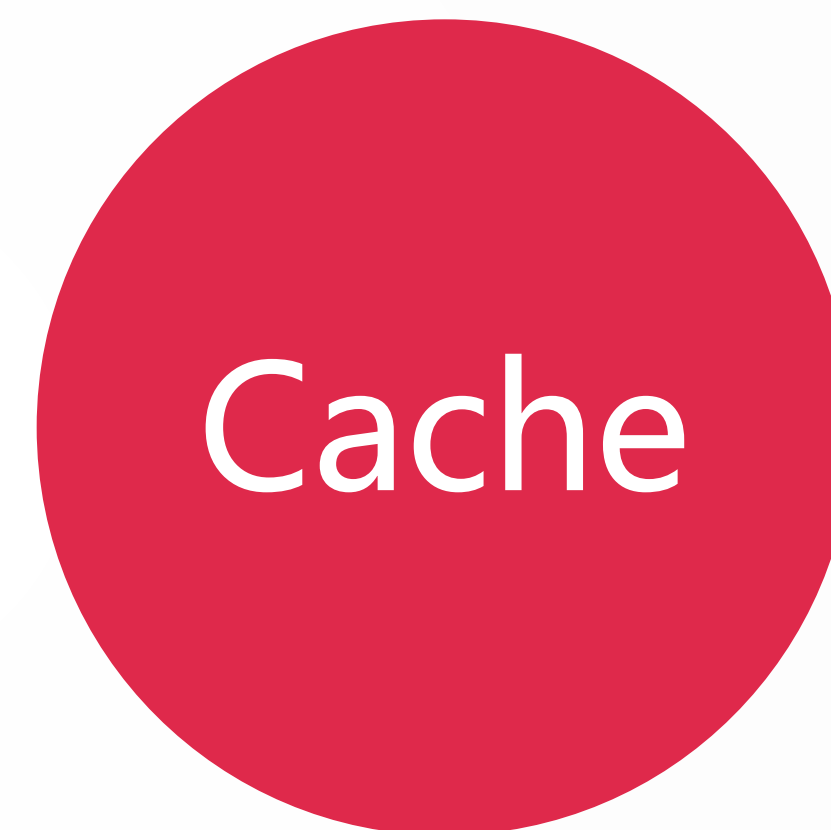
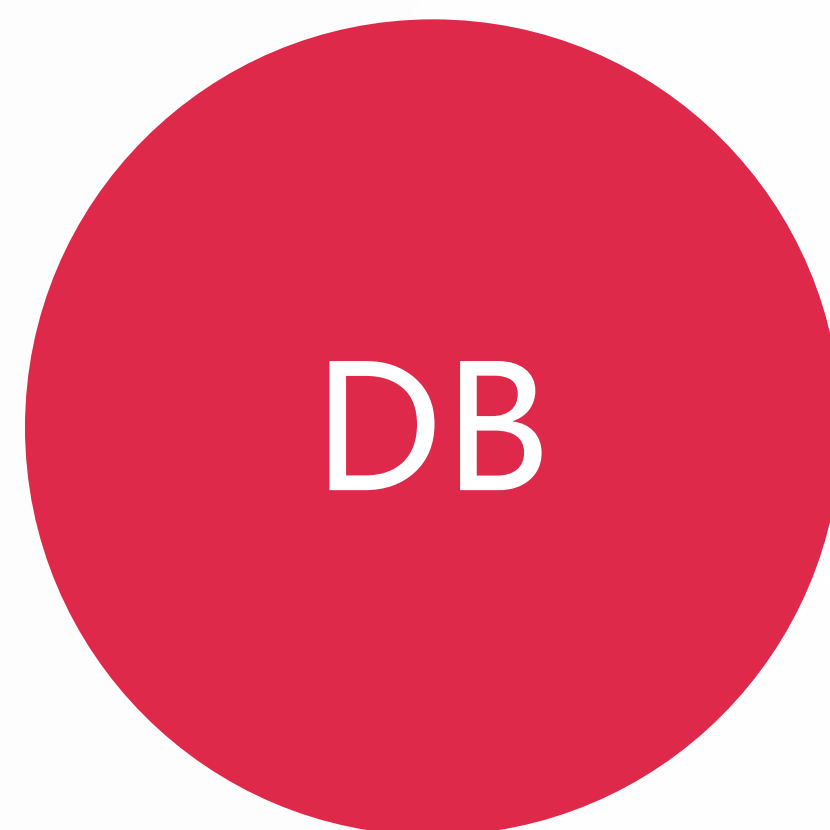
如果没有网桥则创建网桥ensureBridge
 获取容器所在的namespace
 配置veth pair，一个在namespace里面，一个在namespace外面，外面的网卡添加到网桥上
 调用IPAM来配置容器的IP地址，由于配置文件中类型为host-local，调用host-local命令
 配置网关和路由

<https://github.com/containernetworking/plugins/tree/master/plugins/main/bridge>

场景四：Kubernetes + Docker + PM + SR-IOV Netv



高可用设计要点

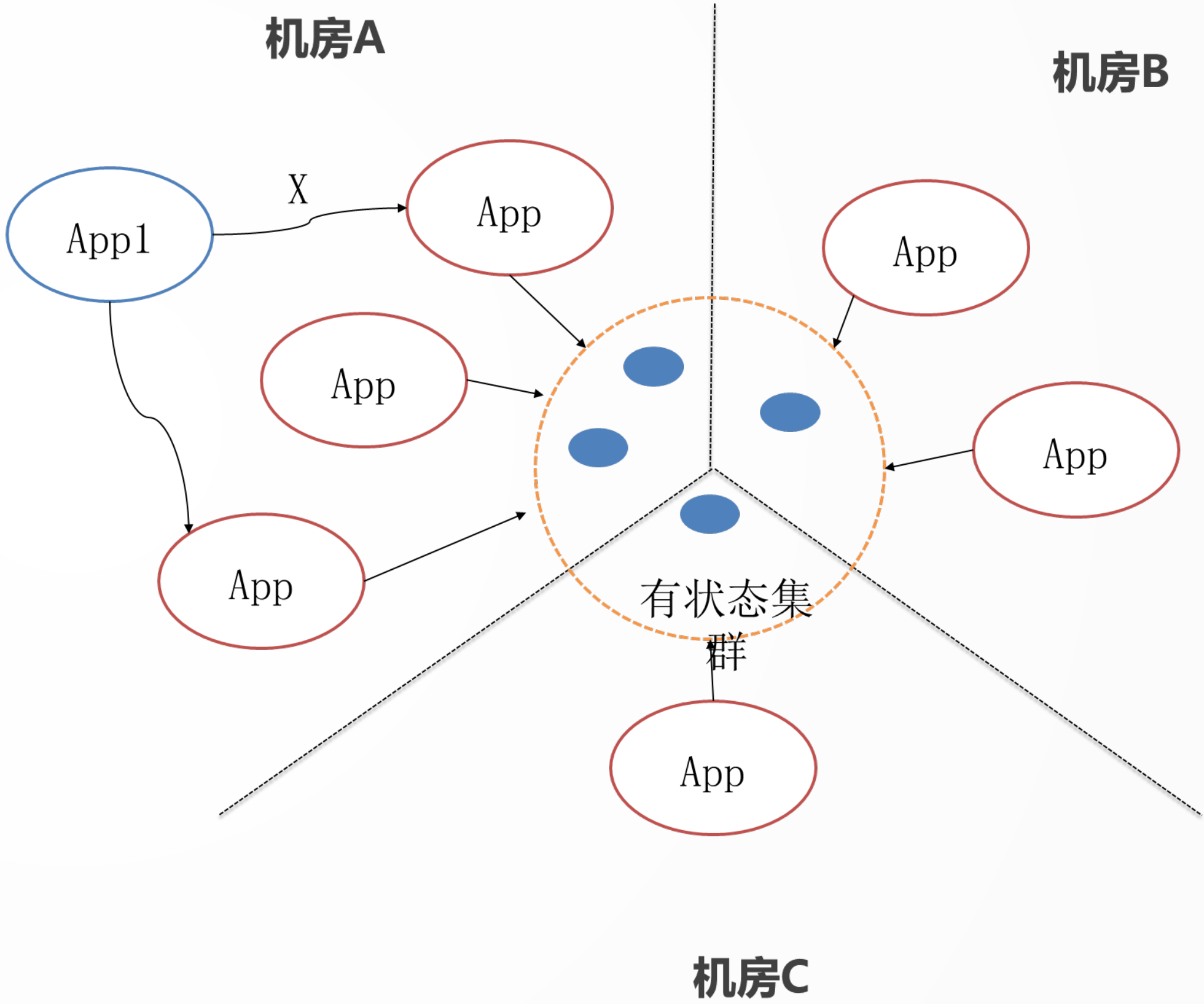


无状态

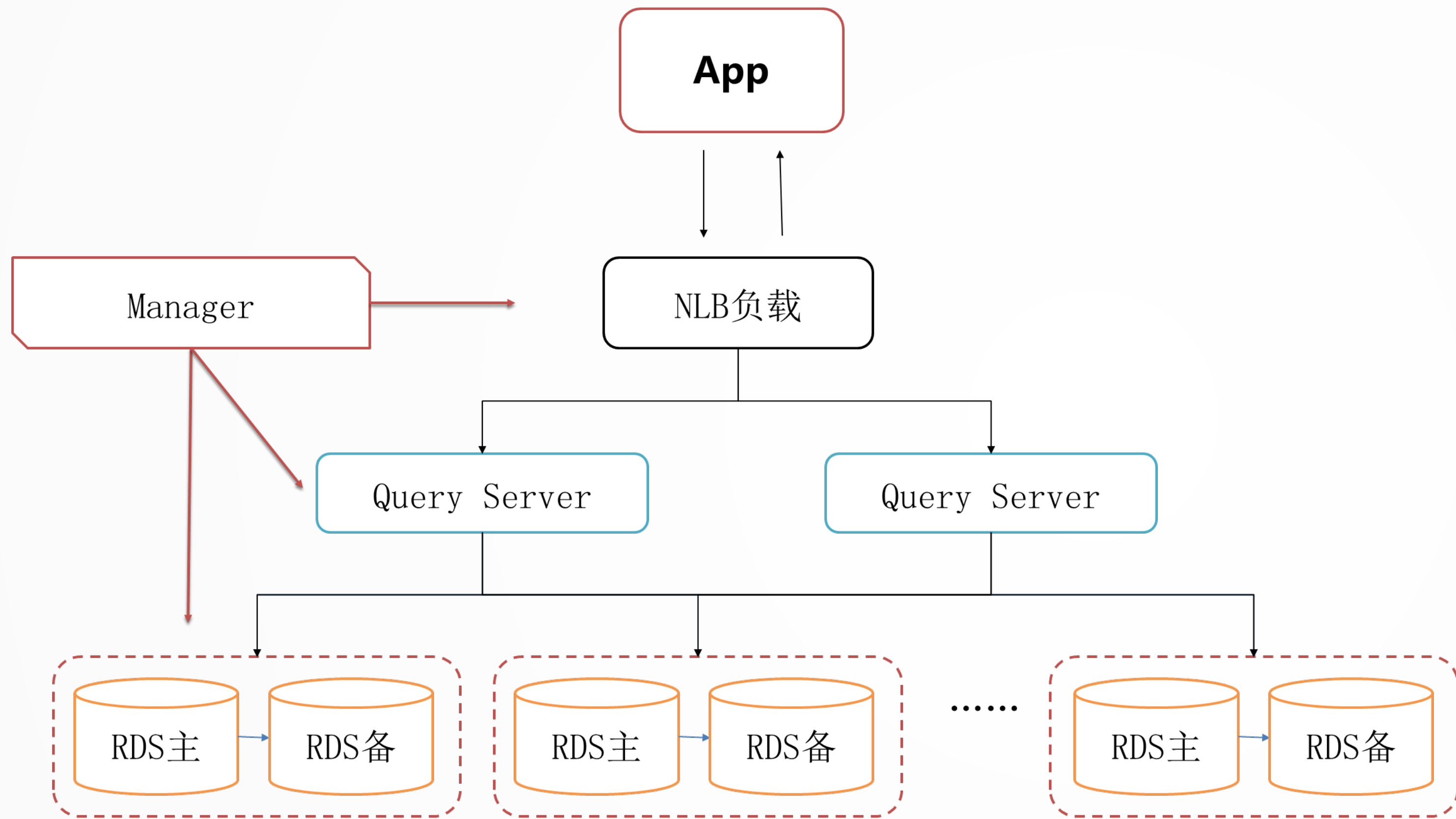


- 独立状态集群
- 服务自动发现
- 失败自动熔断
- 多机房部署

多机房部署



分布式数据库



分布式部署

底层节点主备

负载均衡

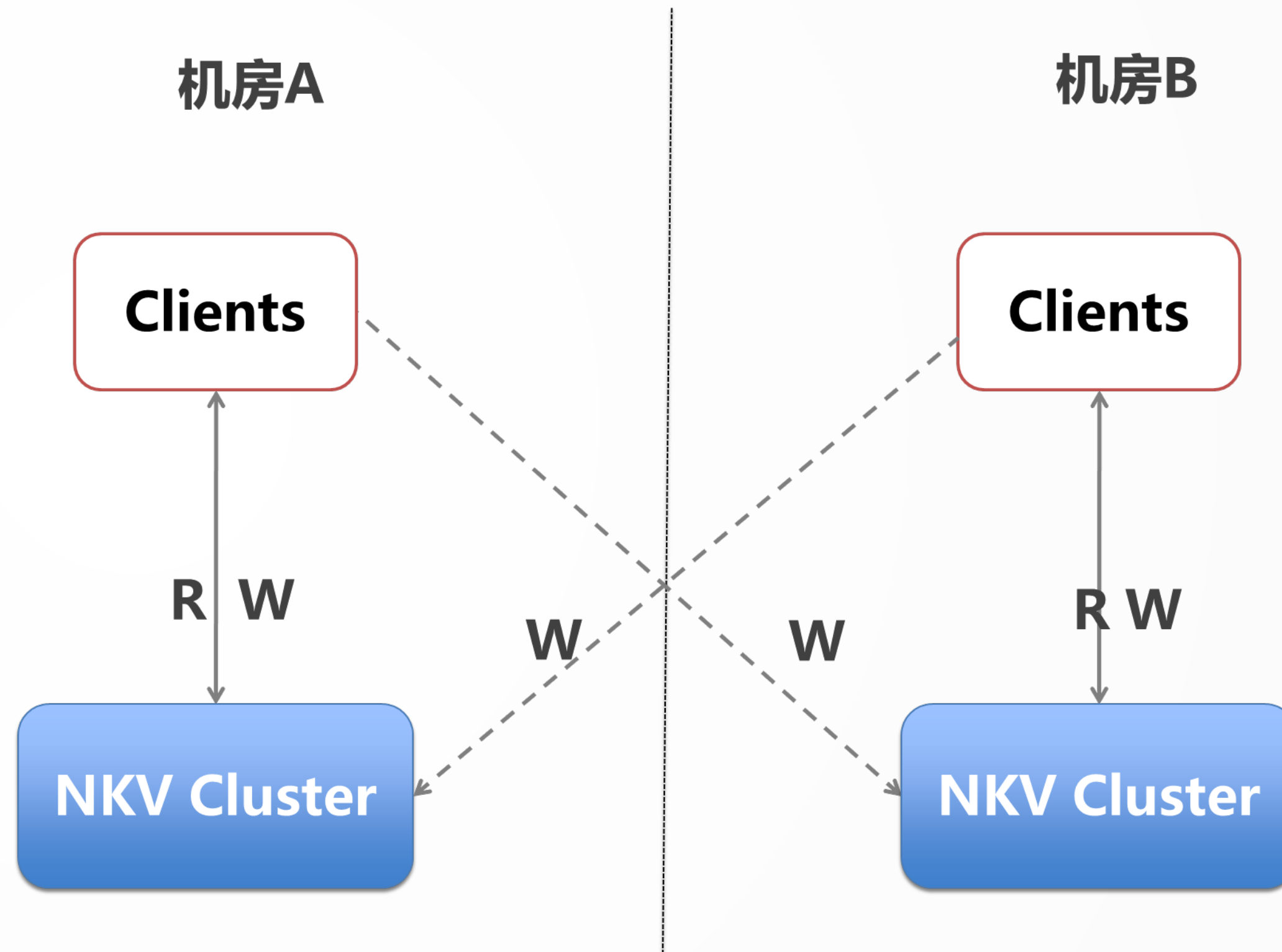
QS自动切换

双机房部署

读写分离

APP缓存
CDN
接入层
静态资源
动态资源静态化
应用本地缓存
分布式缓存
数据库为中心
缓存为中心

双机房部署

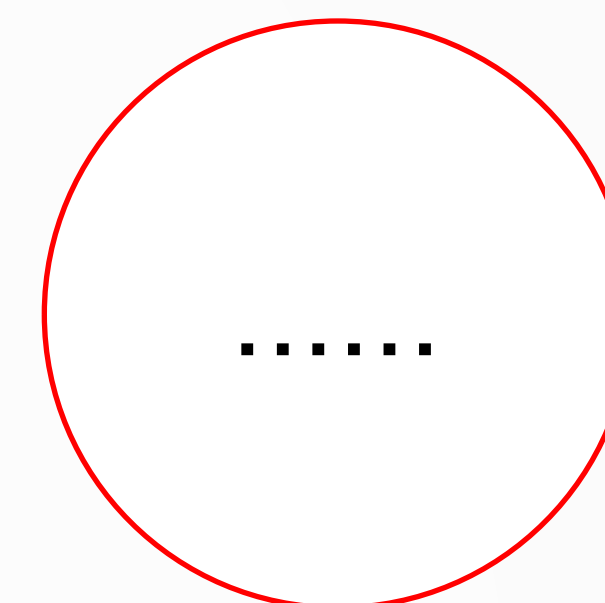




业务隔离
三机房部署



业务隔离
多副本 + HA+ack

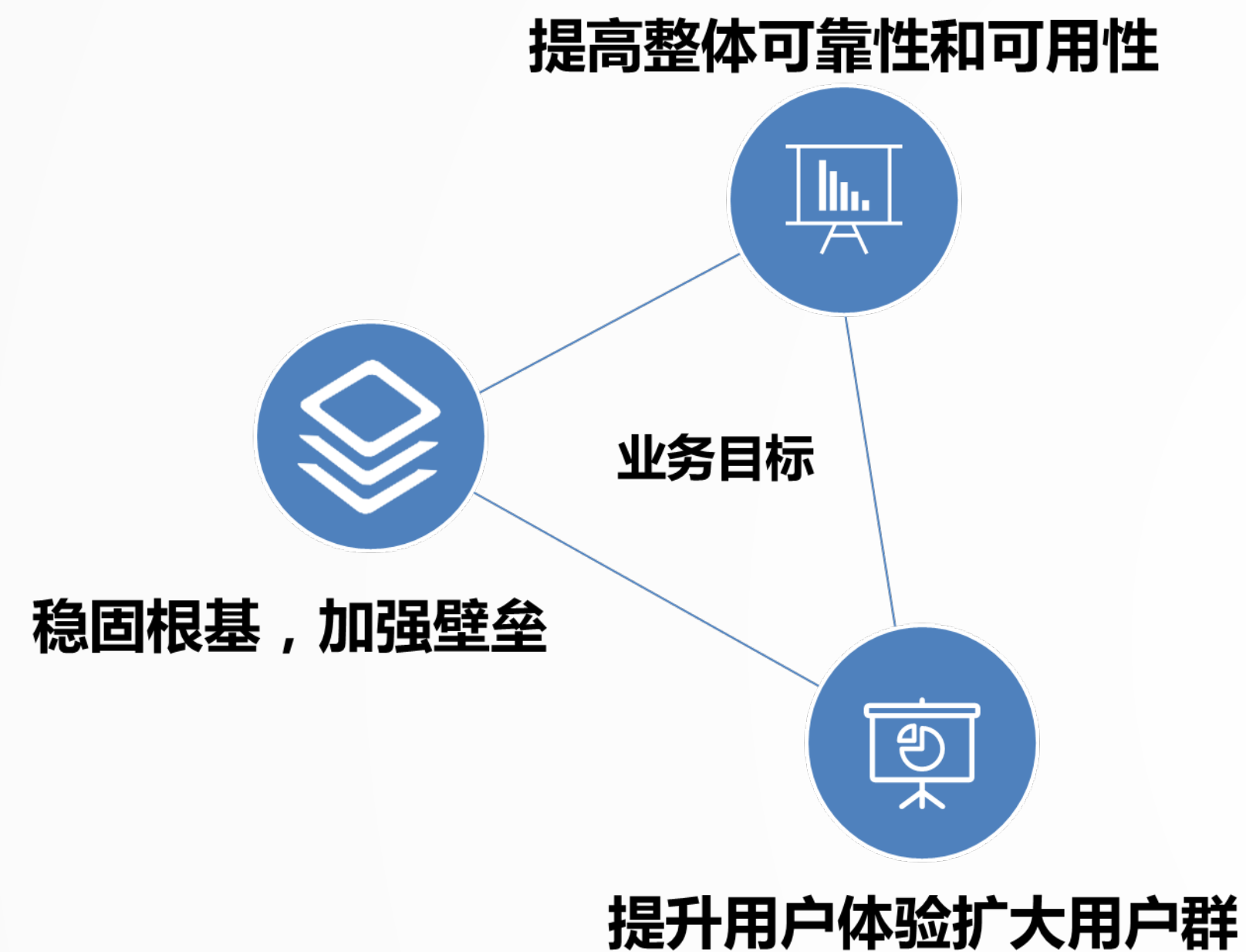


VIP+多实例主从



业务隔离+
Replication

成熟阶段 – 业务目标与现实挑战



VS



考拉：成熟阶段实践



7
个工程
2017双十一前



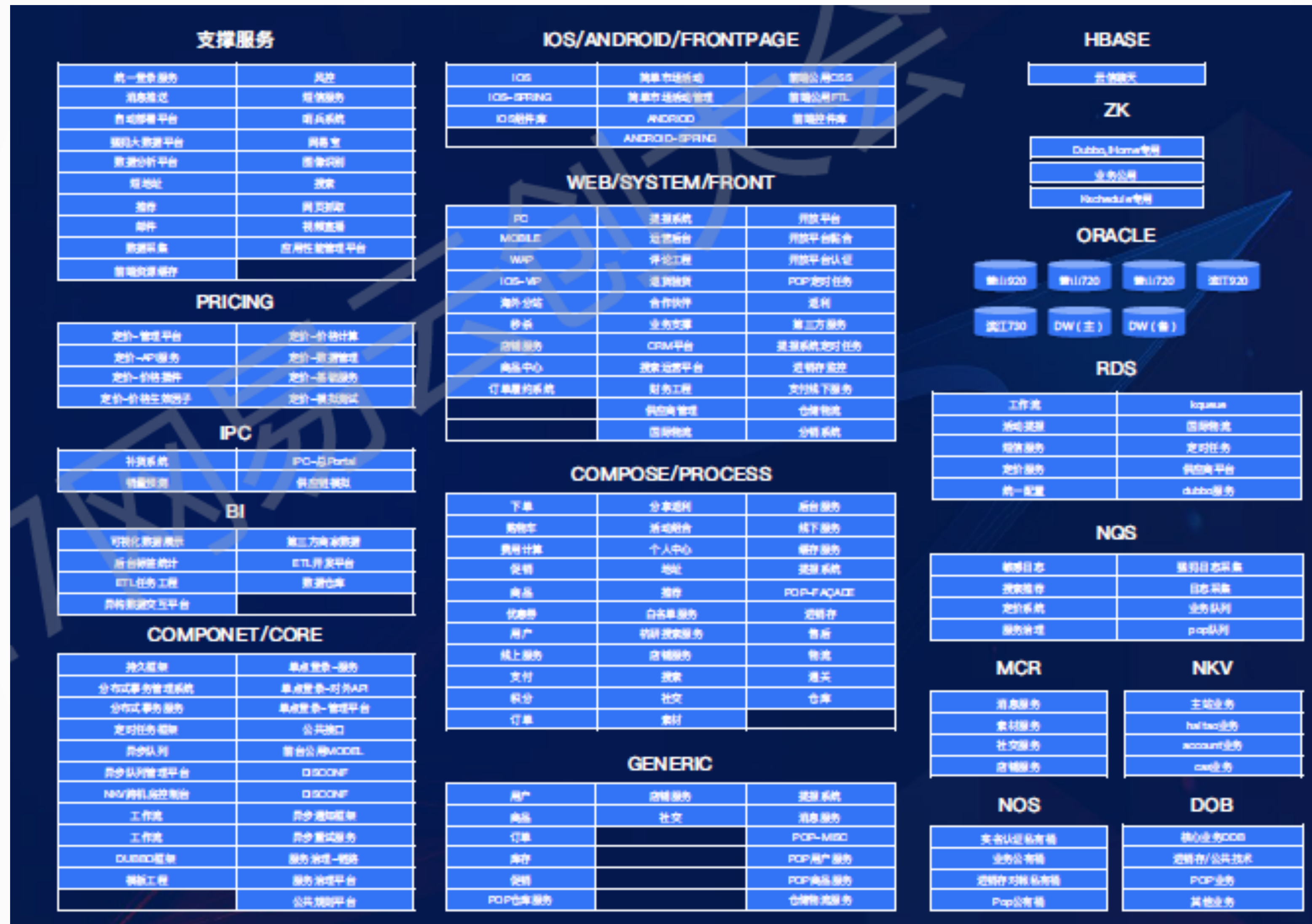
全面服务化进行中

5
倍增长
应用、存储集群

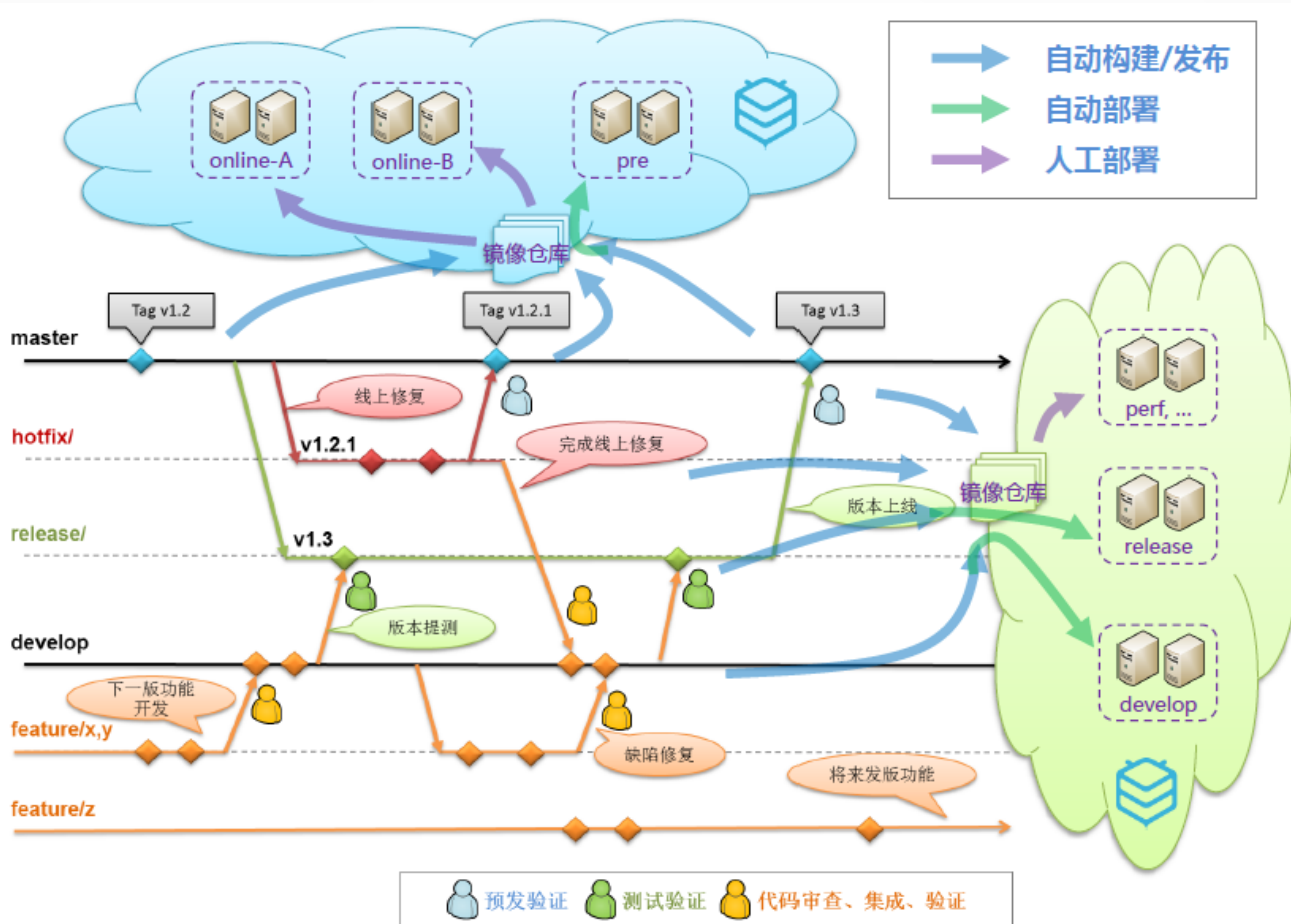


150+
个工程
2017双十一时

考拉服务化架构



不可变基础设施



IaC 部署与扩容 1200->6000节点

```
template "cloud" {
  spec = "C1M2S20"
  init_join = "cw-prod"
  ports = [22]
  env = {
    CONSUL_OPTS = "-client=0.0.0.0"
    TZ = "Asia/Shanghai"
    SERVICE_PROFILE = "online"
    AGENT_SSHD = "Y"
  }
}

template "cloud-nginx" {
  from = "cloud"
  ports = [80,443]
  check http {
    port = 8500
    path = "/v1/agent/checks"
    jq = "all(.Status == \"passing\")"
  }
}

template "cloud-app" {
  from = "cloud"
  ports = [8000,9000]
  env = {
    JAVA_OPTS = "-Xmx1500m -Xms1500m -Xss256k"
    AGENT_HAPROXY = "Y"
  }
  check http {
    port = 9000
    path = "/health"
  }
}
```

```
service "zc-platform-{a,b}" {
  from = "cloud-app"
  image = "//zc-platform:v1.13"
}

service "zc-platform-order-{a,b}" {
  from = "zc-platform-a"
  env = {
    APP_PROFILES = "order"
  }
}

service "zc-admin-{a,b}" {
  from = "cloud-app"
  image = "//zc-admin:v1.12.1"
}

service "zc-payment-{a,b}" {
  from = "cloud-app"
  image = "//zc-payment:v1.6"
}

service "zc-members-{a,b}" {
  from = "cloud-app"
  image = "//zc-members:v1.6"
}
```

```
service "nginx-web-{a,b}" {
  from = "cloud-nginx"
  image = "//nginx-web:v2.13.1"
  depends = ["zc-platform-b","zc-platform-order-b"]
}

service "nginx-cdn-{a,b}" {
  from = "cloud-nginx"
  image = "//nginx-cdn:v2.12.1"
  depends = ["zc-platform-b","zc-platform-order-b"]
}
```

Merged by 向贻祥 4 days ago
The changes were merged into master. You can remove the source branch now.

[Remove Source Branch](#) [Revert](#) [Cherry-pick](#)

Discussion 0 Commits 8 Changes 4

Showing 4 changed files with 15 additions and 7 deletions

[Hide whitespace changes](#) [Inline](#) [Side-by-side](#)

config/comb/wyzc-app-online.conf

```
... @@ -37,7 +37,7 @@
37 37 // web + cdn
38 38 service "nginx-web-online" {
39 39   from = "nginx-online"
40 - image = "hub.c.163.com/wyzcop/nginx-web:v2.13.1"
40 + image = "hub.c.163.com/wyzcop/nginx-web:v2.14.1"
41 41
42 42   env = {
43 43     PLATFORM_SERVICE = "zc-platform-online"
... @@ -50,14 +50,14 @@
50 50
51 51
72 72   service "zc-admin-online-{a,b}" {
73 73     from = "app-online"
74 - image = "hub.c.163.com/wyzcop/zc-admin:v1.12.1"
74 + image = "hub.c.163.com/wyzcop/zc-admin:v1.13"
75 75     spec = 4
76 76     env = {
77 77       JAVA_OPTS = "-Xmx3000m -Xms3000m"
... @@ -98,3 +98,8 @@
98 98   from = "app-online"
99 99   image = "hub.c.163.com/wyzcop/zc-promotion:v1.0"
100 100 }
101 +
102 + service "zc-comments-online-{a,b}" {
103 +   from = "app-online"
104 +   image = "hub.c.163.com/wyzcop/zc-comments:v1.0"
105 + }
```

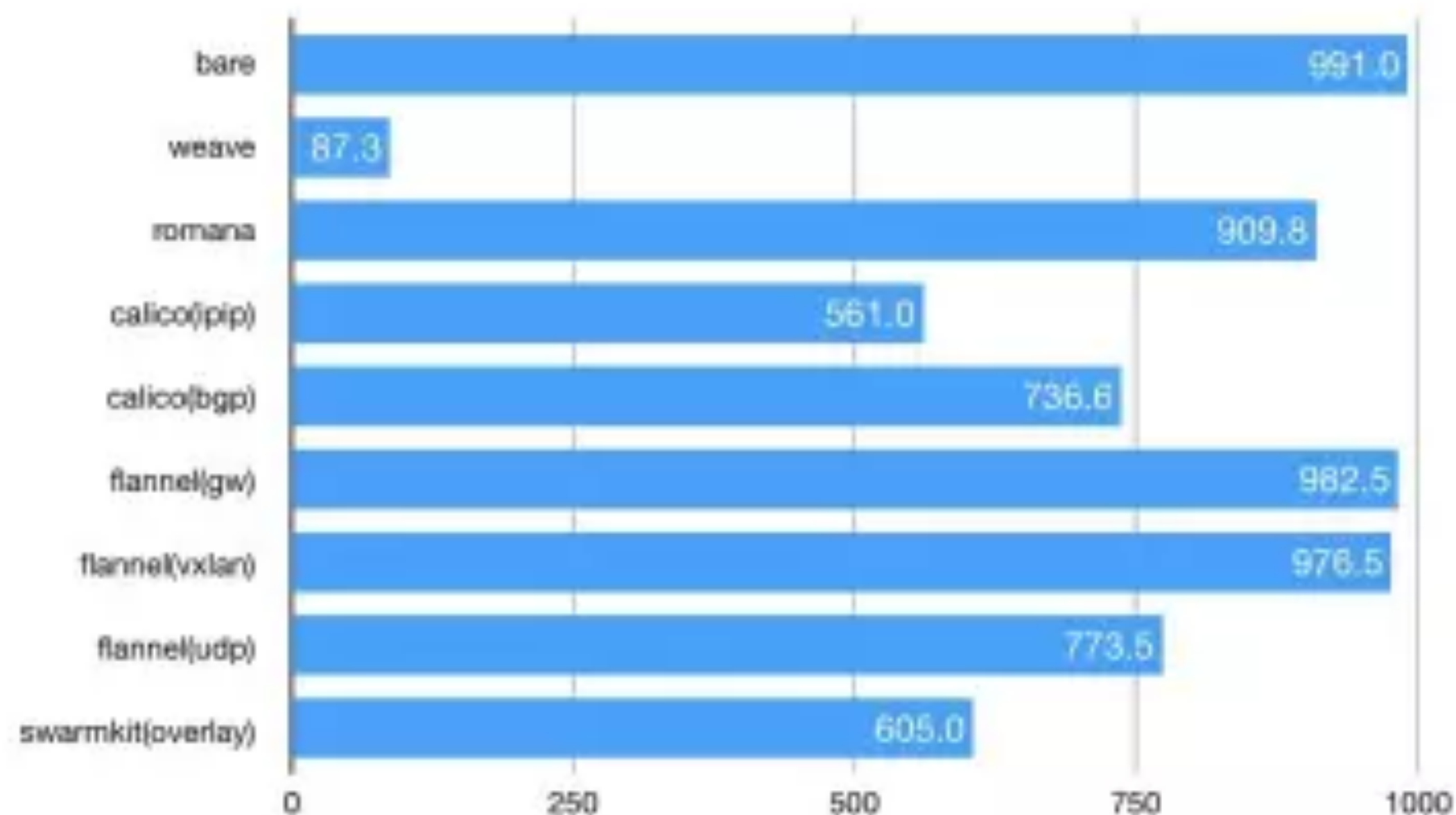
config/wyzc/global-online.properties

```
... @@ -10,4 +10,6 @@ zc-investment.service=zc-investment-online
10 10 # 推广系统服务
11 11 zc-promotion.service=zc-promotion-online
12 12 # 下单服务
13 -zc-platform-order.service=zc-platform-order-online
13 +zc-platform-order.service=zc-platform-order-online
14 +# 评论系统服务
15 +zc-comments.service=zc-comments-online
... \ No newline at end of file
```

场景五：Kubernetes + Docker + PM + Overlay Net



非官方CNI网络性能测试（带宽）



测试环境：

亚马逊云首尔区
(实例类型 t2-small)

软件版本：

Kubernetes	v1.6.2
Weave	v1.9.7
Romana	v1.1.0
Calico	v2.3.0
Flannel	v0.7.1

* 这个是一个非常不严谨的
测试环境，结果仅供参考

娱乐

理论结果：

Bare > Romana
Flannel(gw) ≈ Calico(bgp) > Calico(ipip) ≈ Swarm(overlay)
Flannel(vxlan) > Flannel(udp)
Weave(fastpath) > Weave(sleeve)



场景六：Kubernetes和IaaS层深度融合

1-10

全球首批通过K8S一致性认证

业界领先的多租户动态运行模式，
比AWS的FARGATE到18年才发布

国内首个K8S公有云容器平台
稳定运行1000+天

深度融合先进的基础设施服务
容器与云主机间扁平网络互通

裸机容器消除虚拟化开销
缩短IO路径，性能提升15%+

先进性

安全保障，坚如磐石

数据持久性

$\geq 99.9999999\%$

每 1 亿个云服务器磁盘，每月最多只有 1 个云服务器磁盘可能发生数据丢失

服务可用性

$\geq 99.95\%$

故障恢复

5分钟内自动迁移

服务等级协议（SLA）

支持超大规模集群，业界领先

30000+ 台
集群调度规模（节点数）

450000+ 个
应用规模（容器数）

集群规模支持

动态资源创建优化

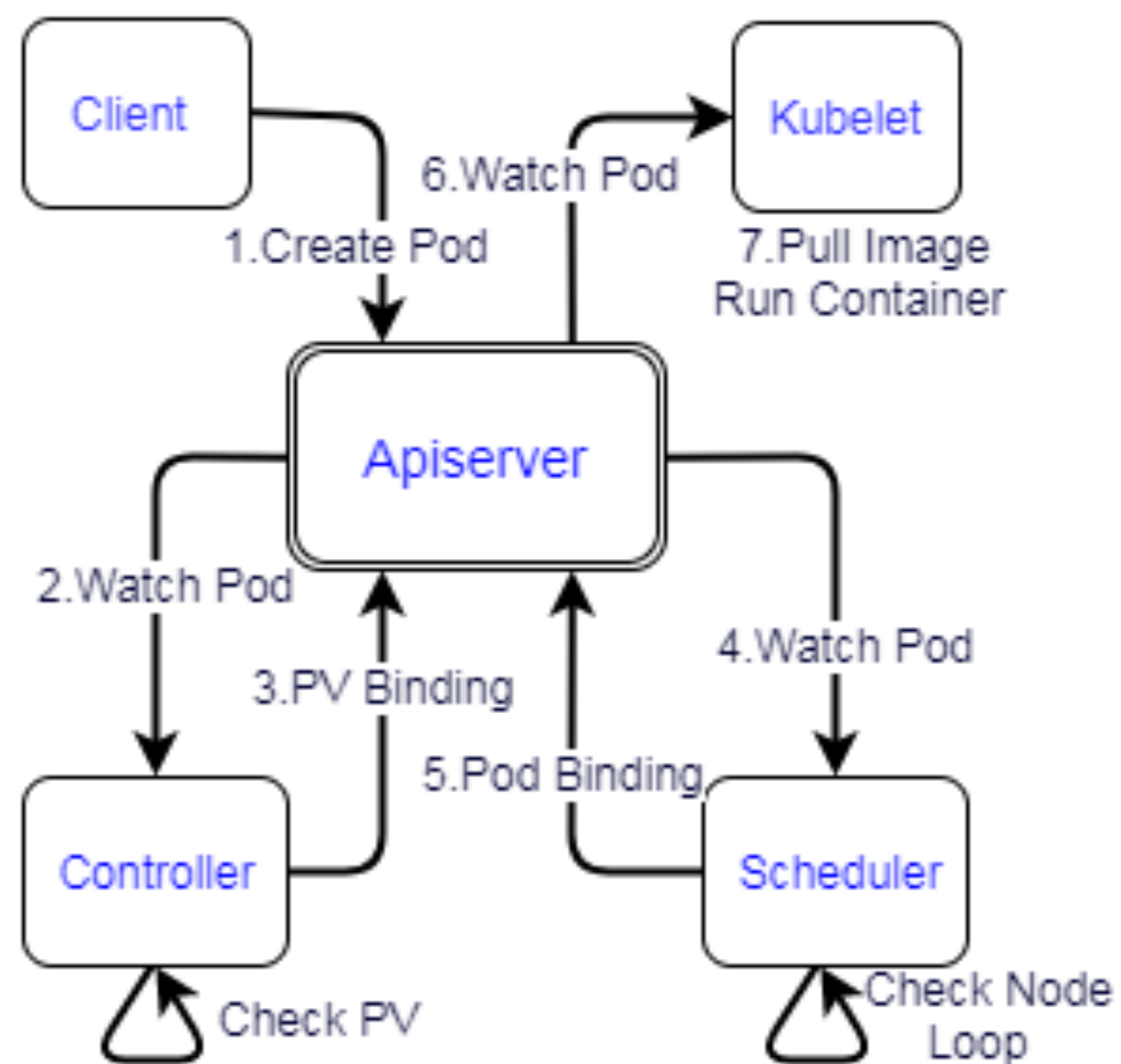


图3 原生kubernetes 创建容器流程

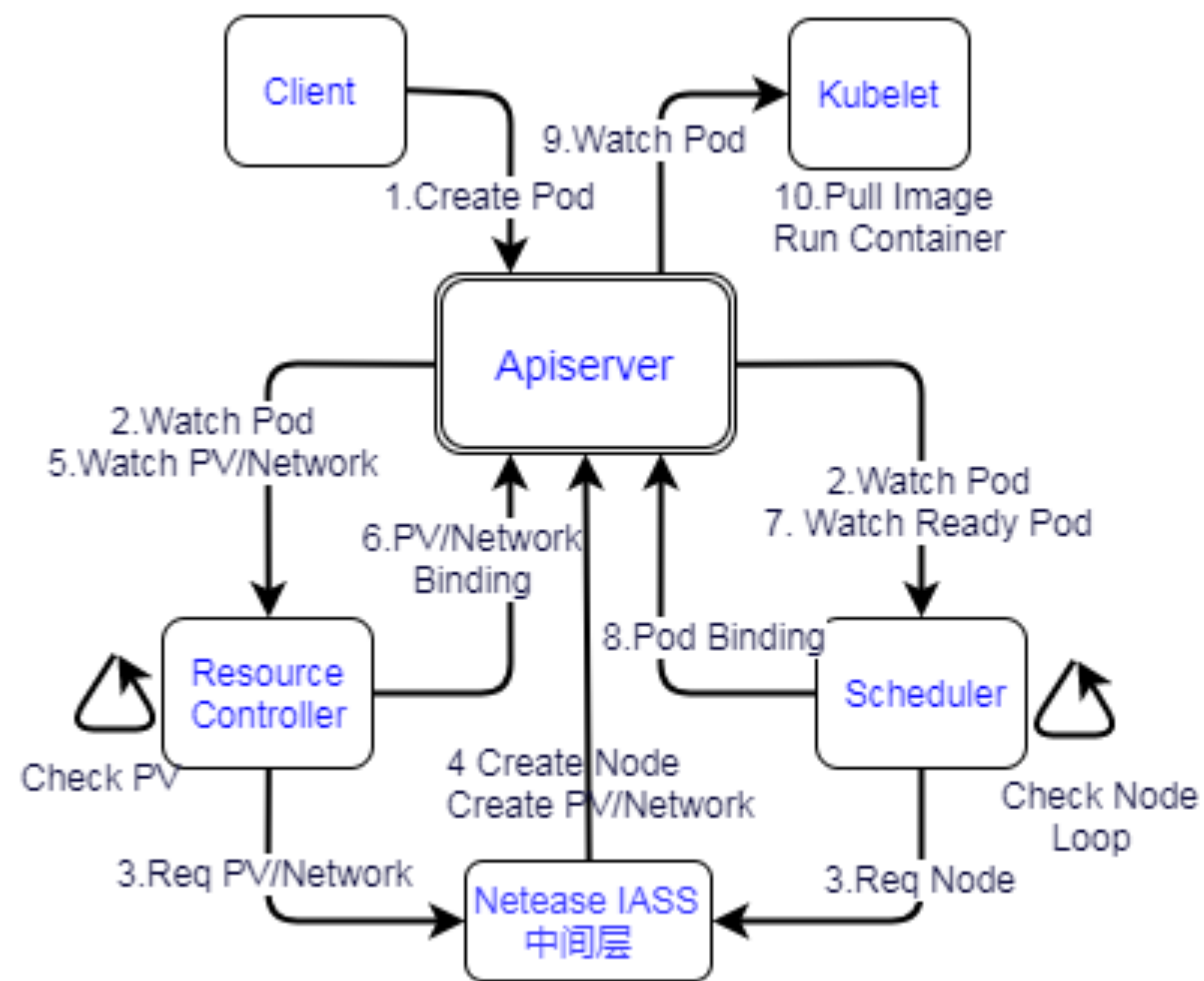
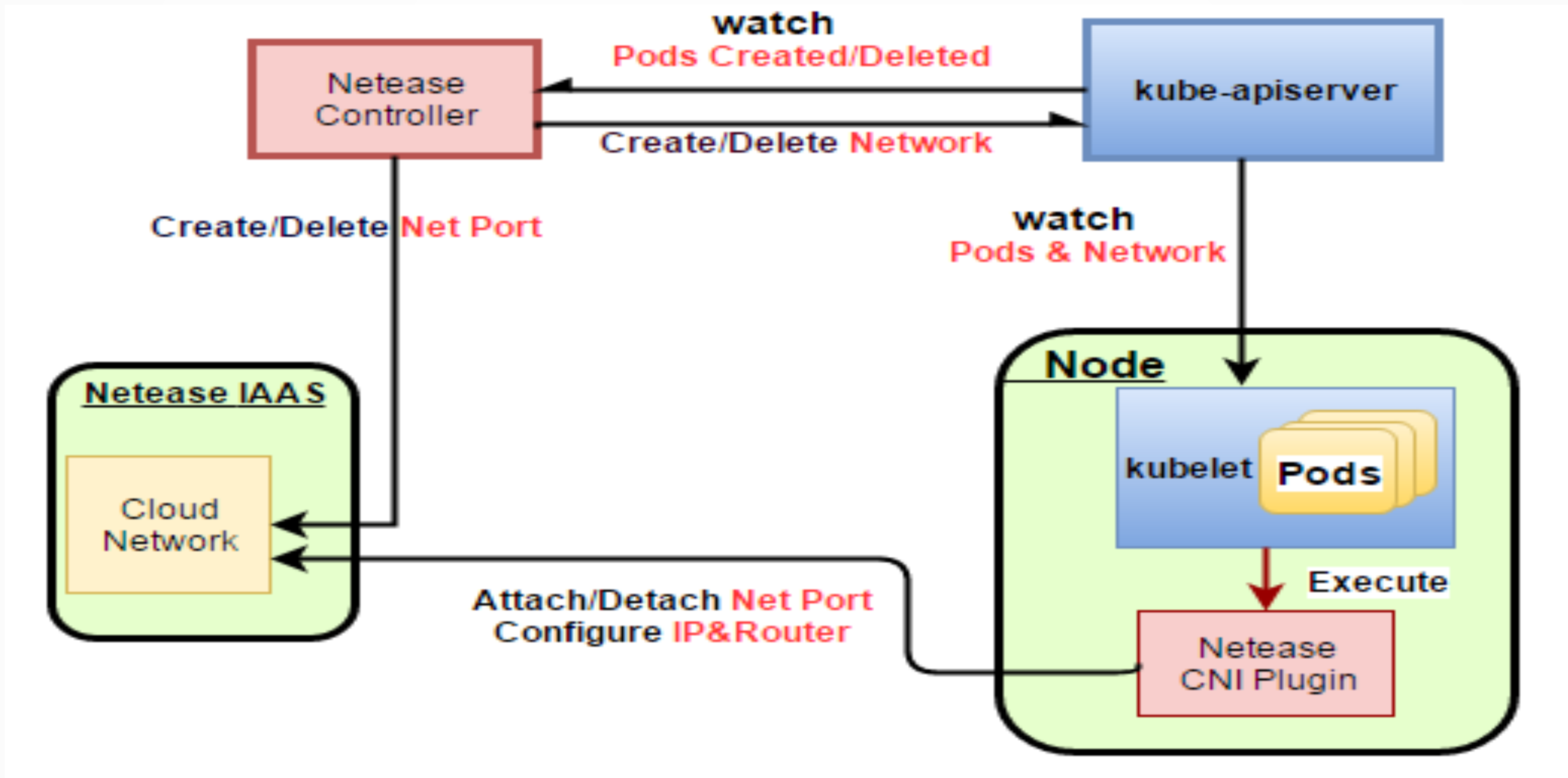
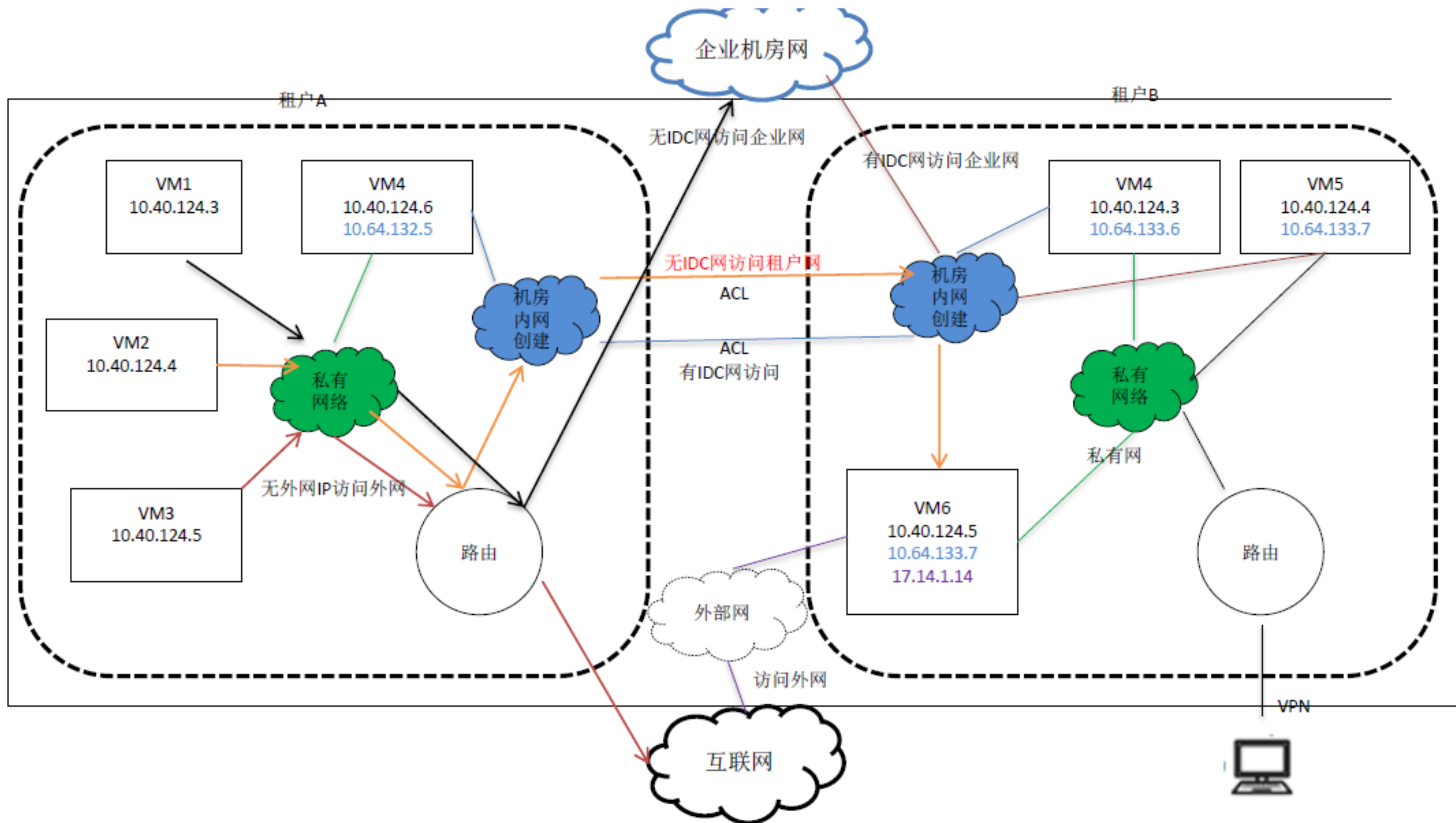
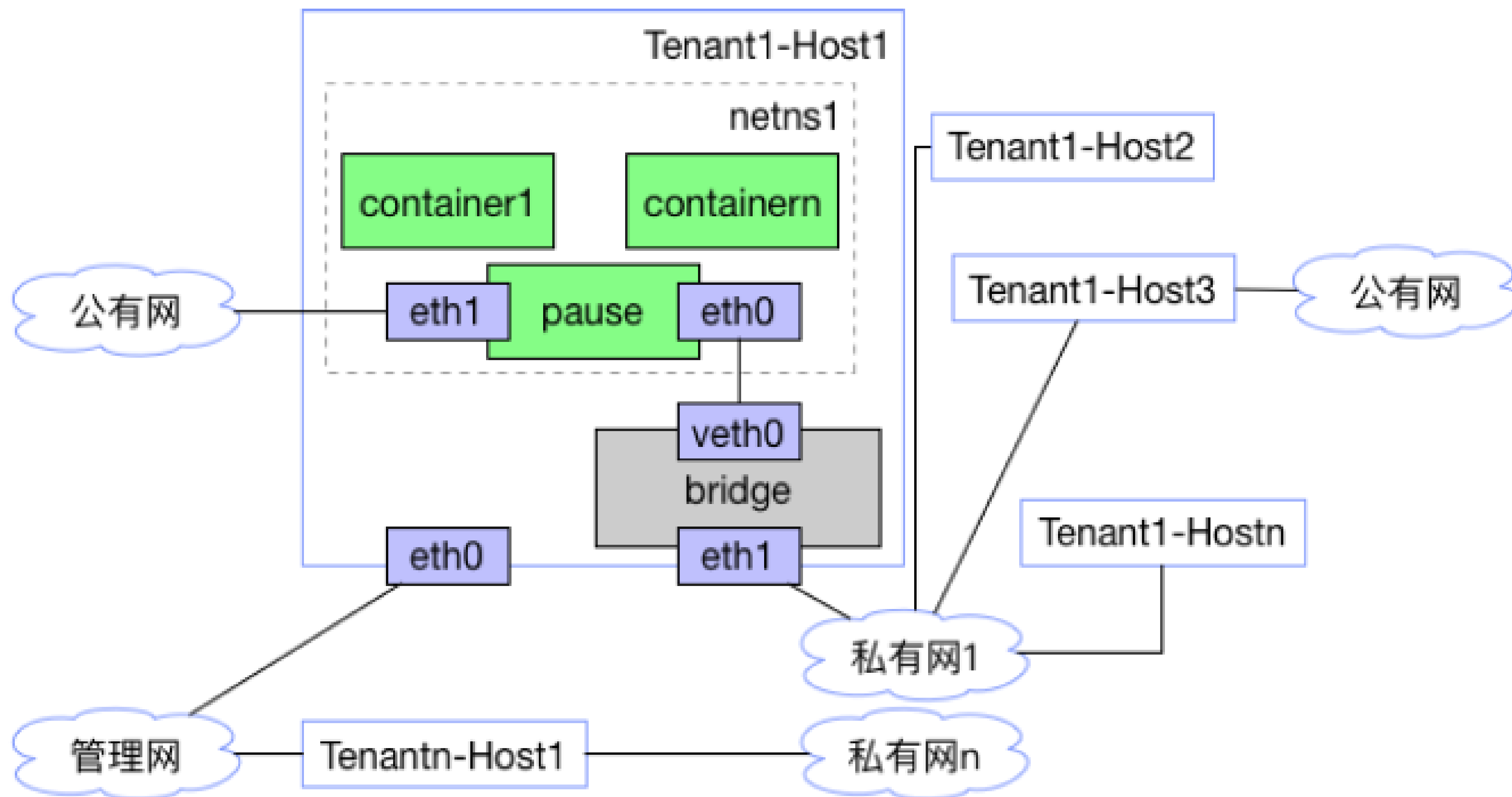
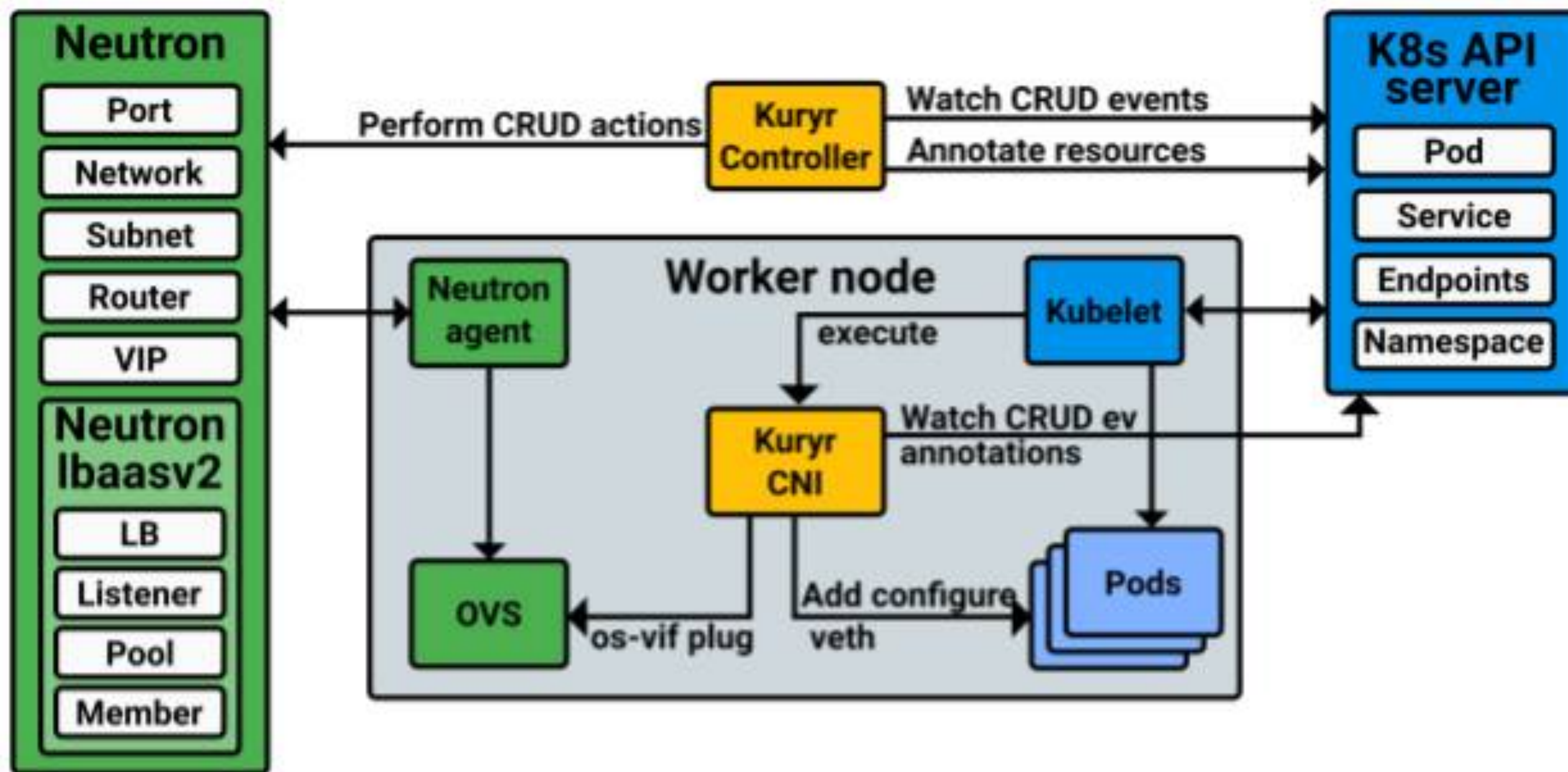


图4 网易云改造后的 kubernetes 创建流程









The two Kuryr-Kubernetes components depicted with all the main components they interact with.



网易云

共创云上精彩世界