



QCon 全球软件开发大会
INTERNATIONAL SOFTWARE
DEVELOPMENT CONFERENCE

BEIJING 2017

搜狗商业云平台实践与思考

王宇



王宇

About me

搜狗商业平台研发部 资深开发工程师

主要负责商业平台基础研发工作，包括报文系统、分布式缓存、商业云平台等系统研发工作

目前重点关注分布式、大数据、云计算等相关领域

曾就职于腾讯



CONTENT

1 | 商业平台与云

2 | 云平台设计选型

3 | 云平台实践

4 | 总结&展望

搜狗商业平台特点

见多识广

- 多—服务个数多，迭代版本多
- 广—技术体系涵盖广

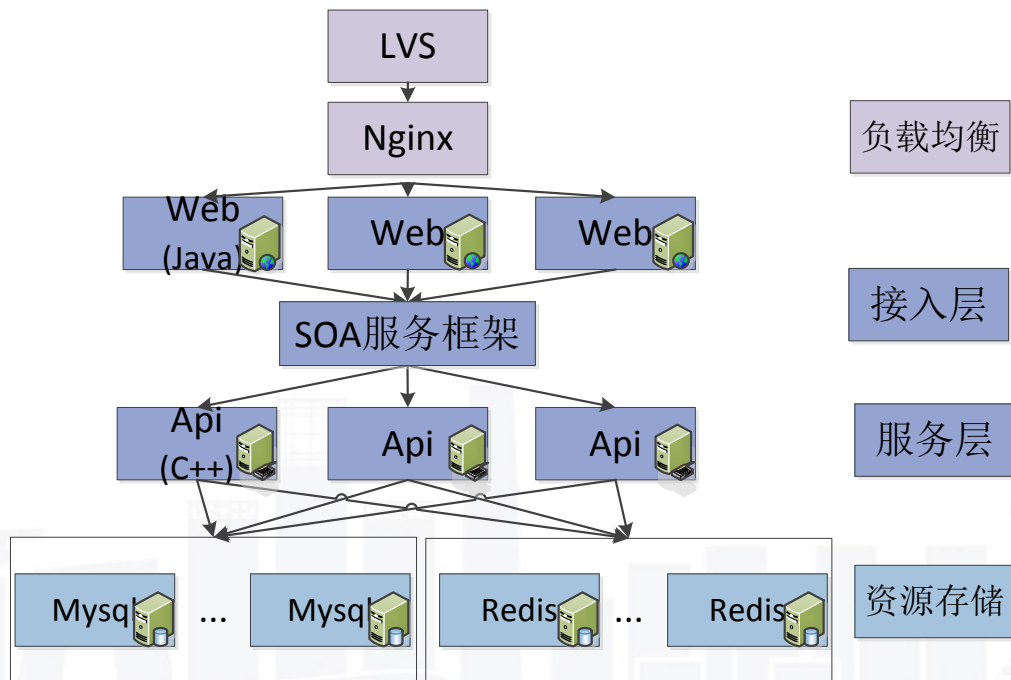
又快又稳

- 快—性能决定收益
- 稳—稳定压倒一切



典型应用的系统架构

- 分层结构、服务化
- 跨语言、多环境
- 物理隔离



面临的挑战



环境复杂性



业务低峰期资源利用率低



弹性伸缩能力不足



商业云平台





CONTENT

1 | 商业平台与云

2 | 云平台设计选型

3 | 云平台实践

4 | 总结&展望

行业现状

Container RunTime

Docker



Rkt



OCI



Scheduling &Orchestration

Kubernetes



MESOS



SWARM



Docker

标准化的构建、交付、运维手段

65%

use Docker to deliver development agility.

48%

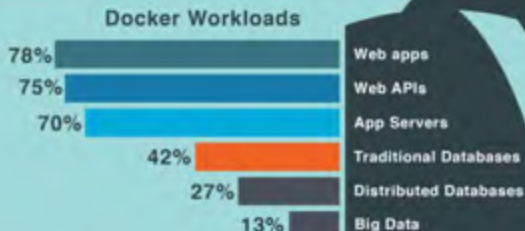
use Docker to control app environments.

41%

use Docker to achieve app portability.

90%

use Docker for apps in development.



58%

use Docker for apps in production.



90%

plan dev environments around Docker.

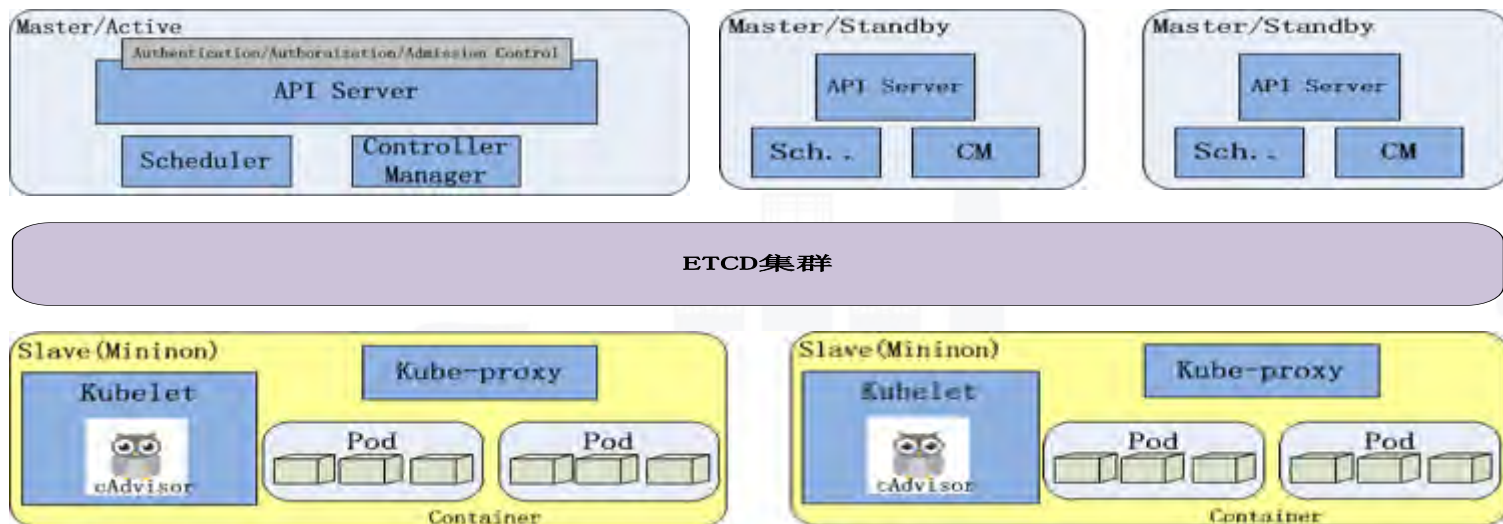


80%

plan DevOps around Docker.

Kubernetes(K8s)

- 蓬勃发展的社区：GitHub、Stackflow
- 优秀的设计理念：Pod、ReplicationController、Deployment、Label 等
- 项目匹配：规模在数千台、需要快速落地

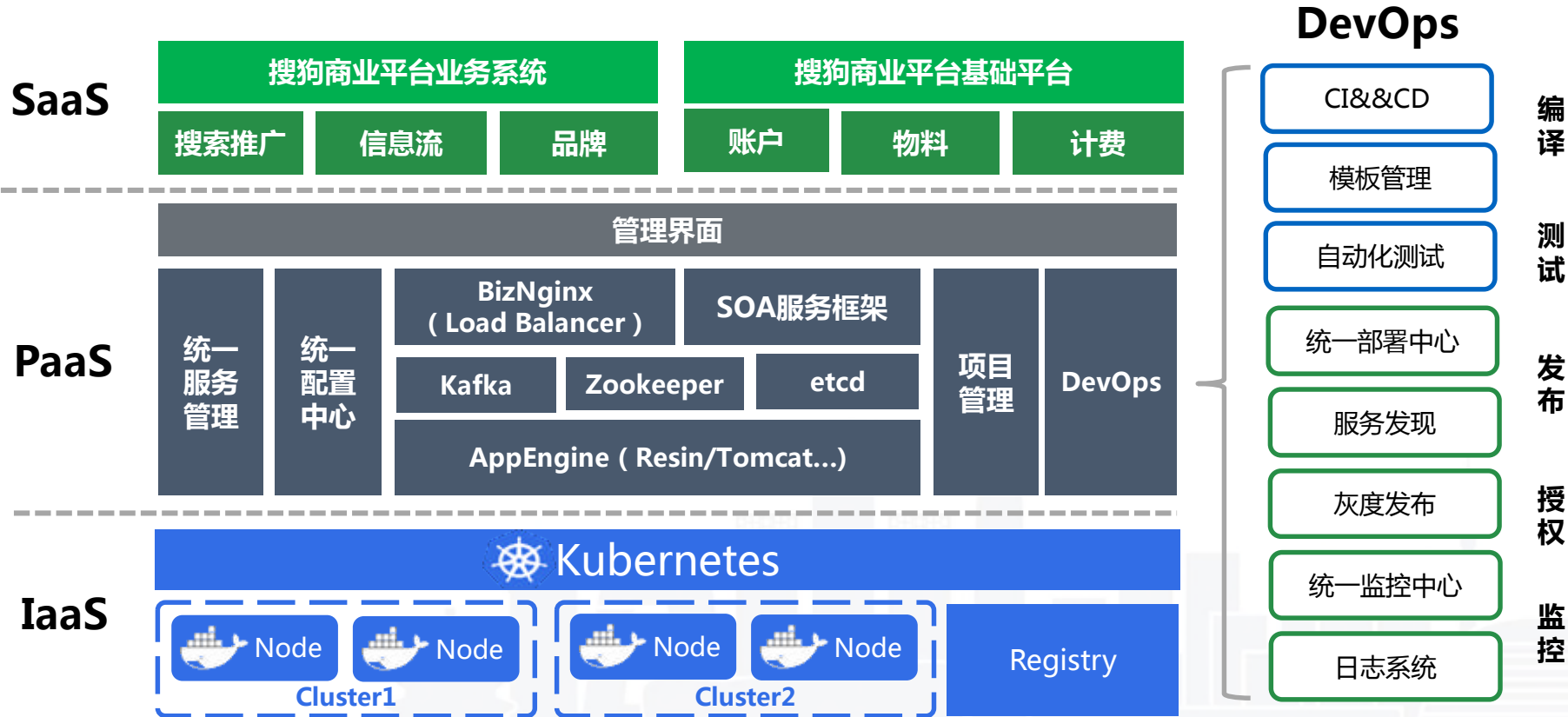


商业云平台

- 云平台=K8s+Docker ?
 - 持续集成、服务发现、服务授权 ?
 - 已有开发、测试、运维流程如何对接 ?
 - 自动化平台如何实现 ?
- 自研PaaS



商业云平台





CONTENT

1 | 商业平台与云

2 | 云平台设计选型

3 | 云平台实践

4 | 总结&展望

云平台实践



实践

基础集群

服务管理

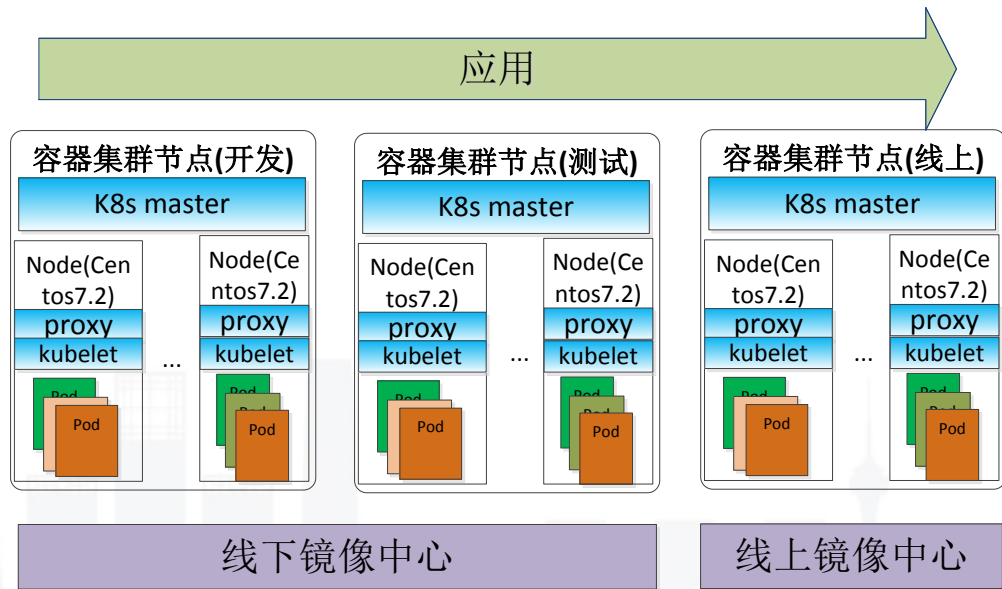
服务授权

灰度发布



基础集群

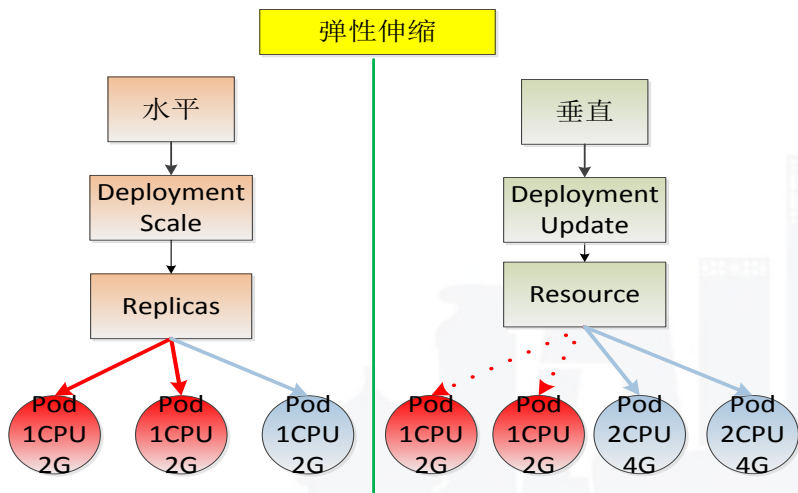
- 基础集群：
 - 多环境隔离
 - K8s集群：线上、测试、开发
 - 镜像中心：线上、线下
- 应用
 - 多环境隔离



基础集群

- 自动Failover
- 提升弹性：
 - 水平扩缩容
 - 垂直扩缩容

- 提升资源利用率：
 - 任务混布
 - Qos资源定义：limit,request



实践

基础集群

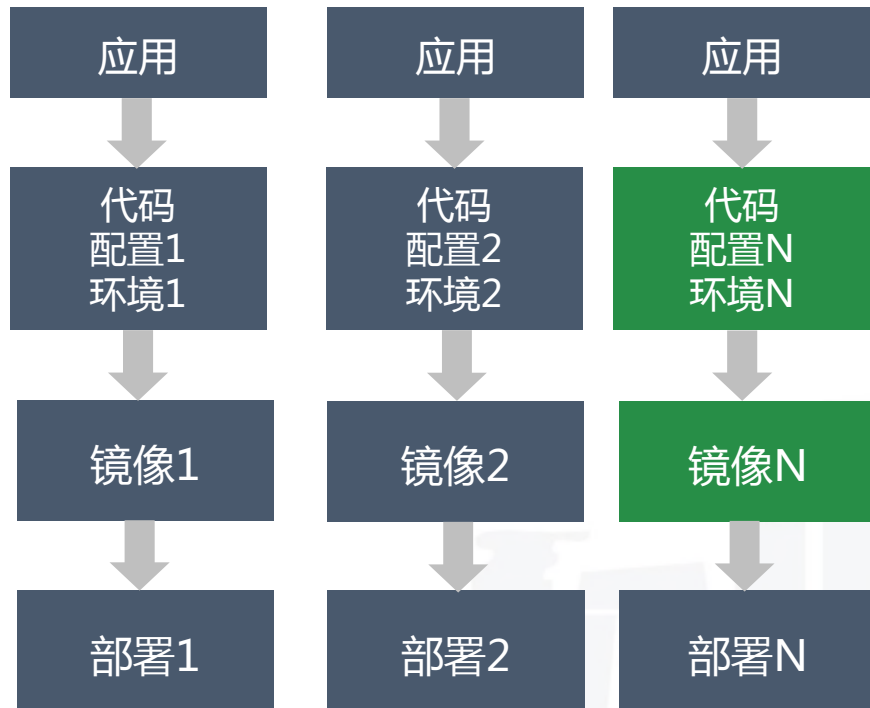
服务管理

服务授权

灰度发布



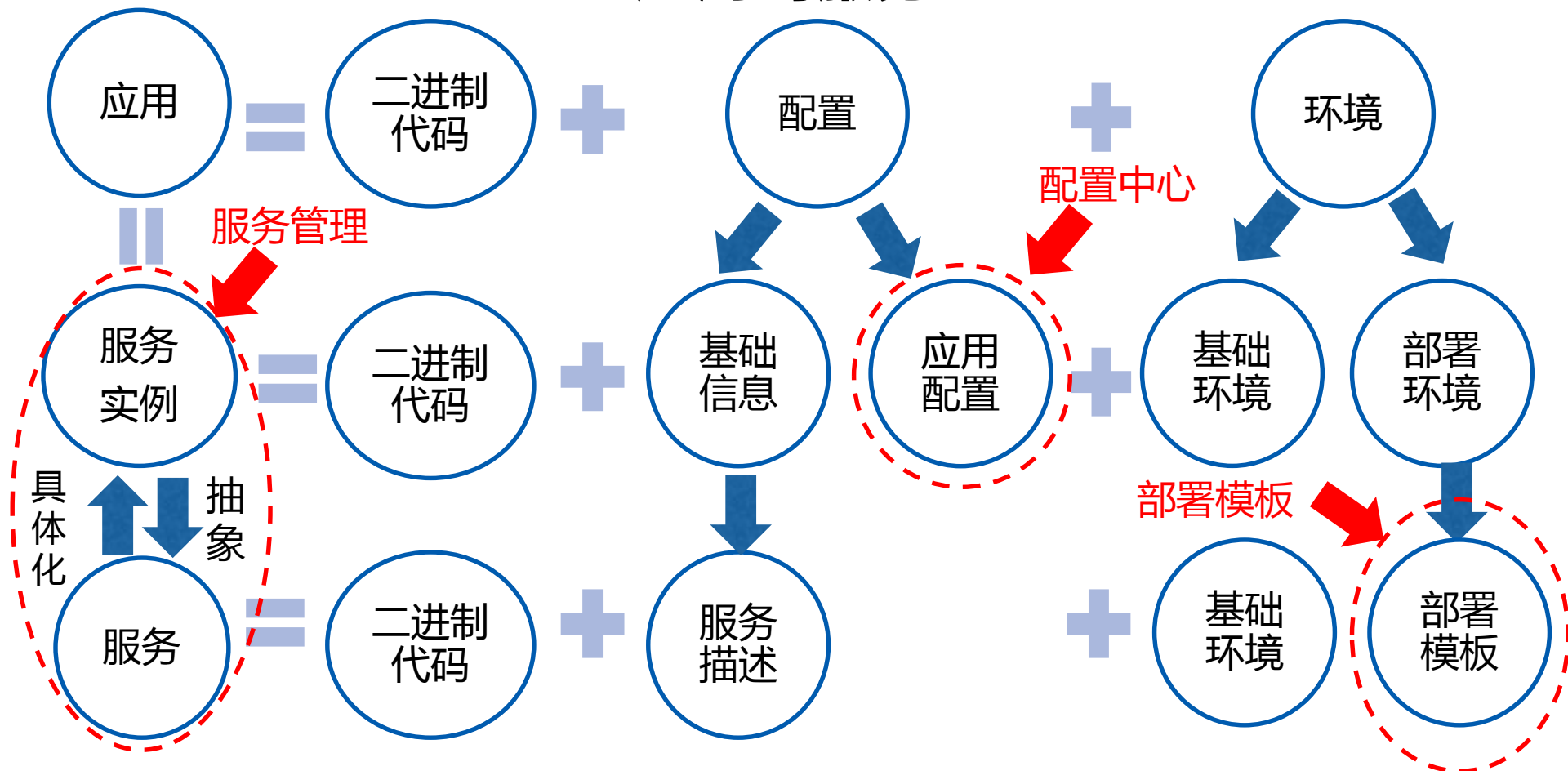
挑战



提升可重用性



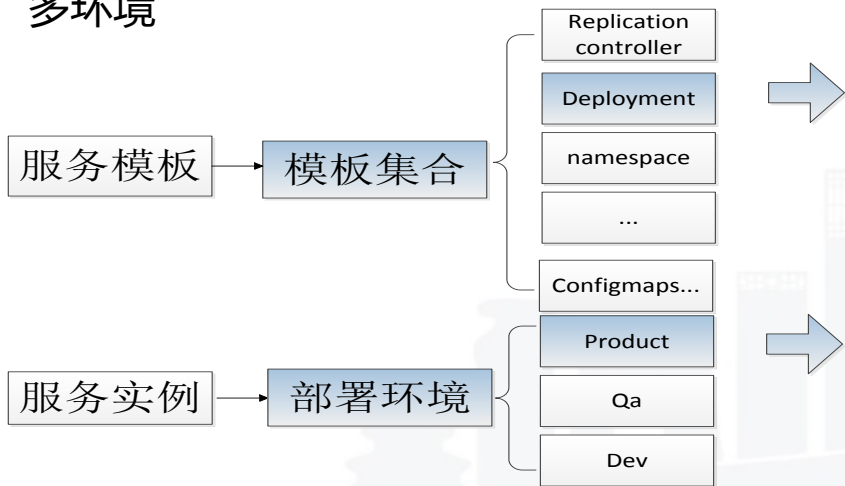
应用到服务



部署模板

部署模板

- Deployment、namespace、Configmaps等
- 多环境

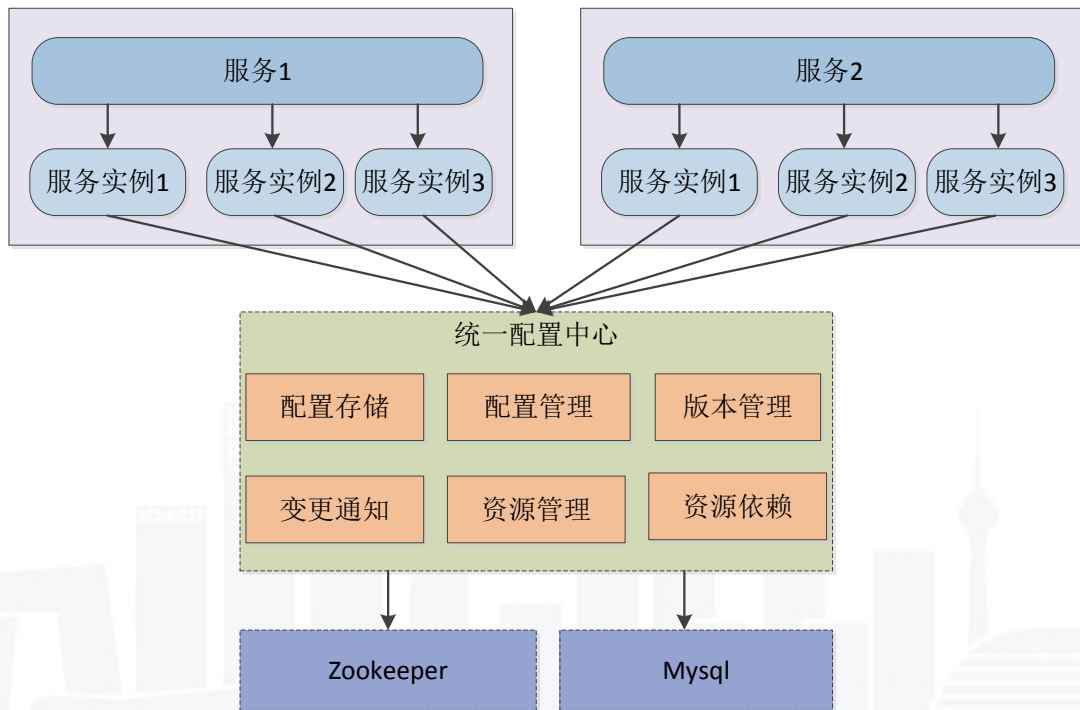


Deployment模板

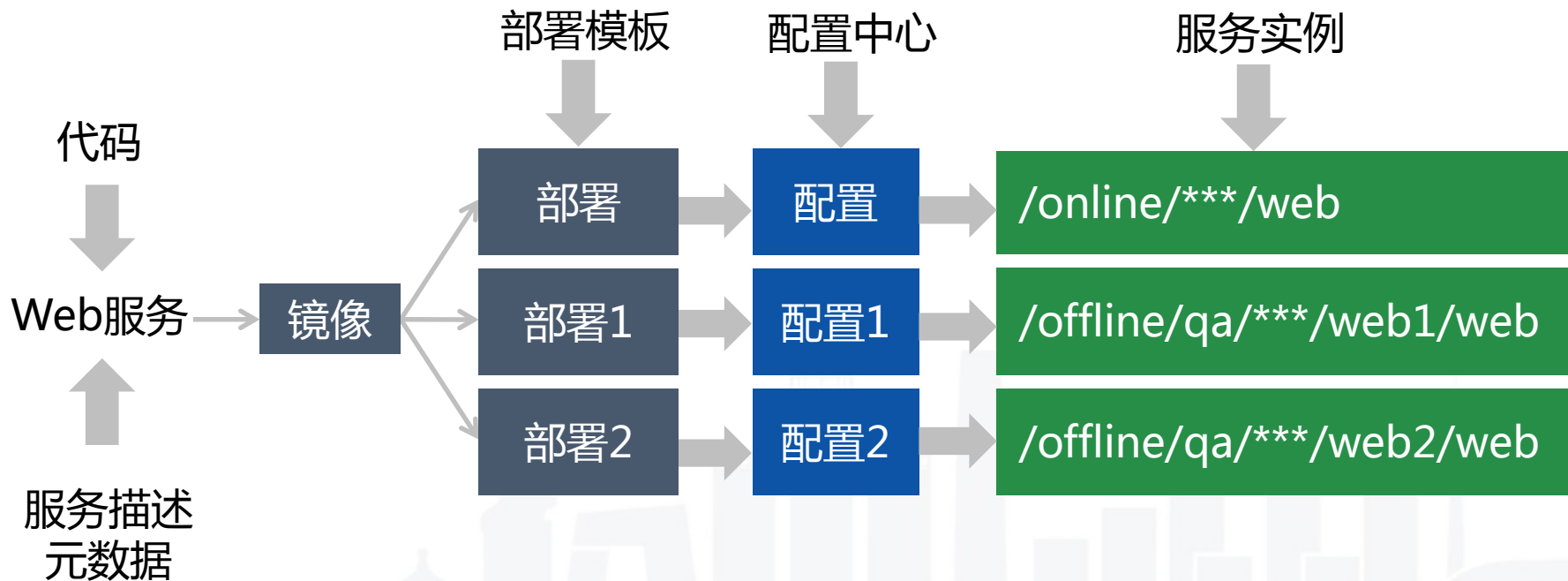
```
kind: Deployment
apiVersion: extensions/v1beta1
metadata:
  name: {{.ServiceInstanceName}} ← 服务实例名
  namespace: {{.ServiceName}}-{{.ServiceInstanceId}}
  labels:
    name: {{.ServiceInstanceName}} ← 服务名
spec:
  replicas: {{.Replicas}} ← 副本数
  selector:
    matchLabels:
      app: {{.ServiceInstanceName}}
  template:
    metadata:
      labels:
        app: {{.ServiceInstanceName}}
        version: {{.SrcVersion}}.{{.PubVersion}} ← 版本号
    spec:
      hostNetwork: {{.IsHost}}
      containers:
        - name: {{.ServiceName}}
          image: {{.ImageName}} ← 镜像名
          env:
            - name: KUBE_SERVICE_NAME
```

统一配置中心

- 多环境支持：
- 服务实例上传应用配置
- 应用配置管理：
 - 支持配置备份存储
 - 支持配置变更热加载



服务管理示例



实践

基础集群

服务管理

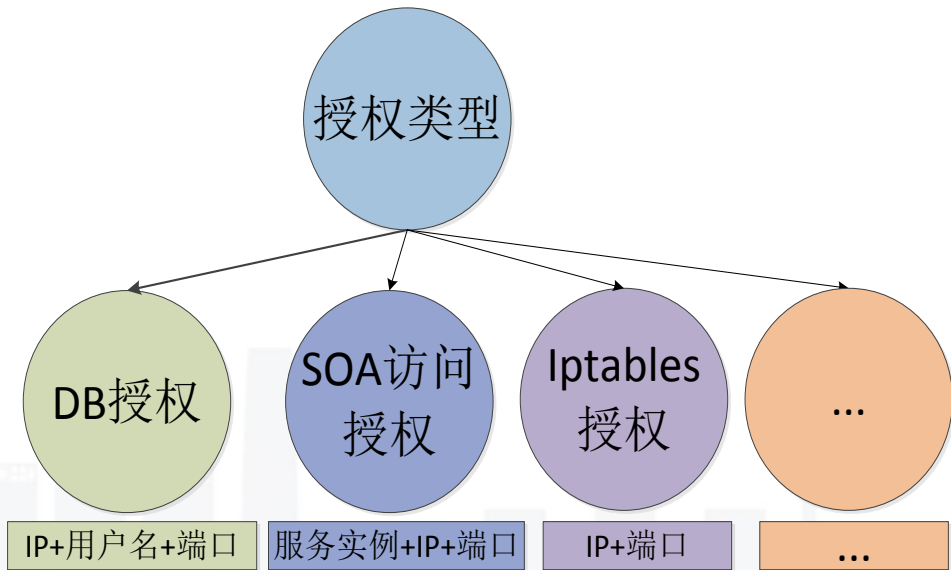
服务授权

灰度发布



服务授权

- 服务授权：
 - 授权的重要性
 - 防止测试流量打到线上
 - 防止恶意访问等
 - 授权是云平台的核心功能
- 服务授权的挑战
 - 服务依赖关系
 - 在云平台下，容器IP会随时变动
 - 多种授权类型：粒度不同



服务授权

Who

- 如何自动计算给哪些服务授权

When

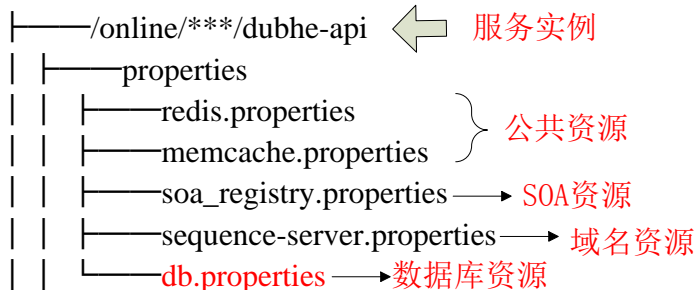
- 什么时候进行授权

How

- 怎么样进行授权

Who: 配置中心

配置文件



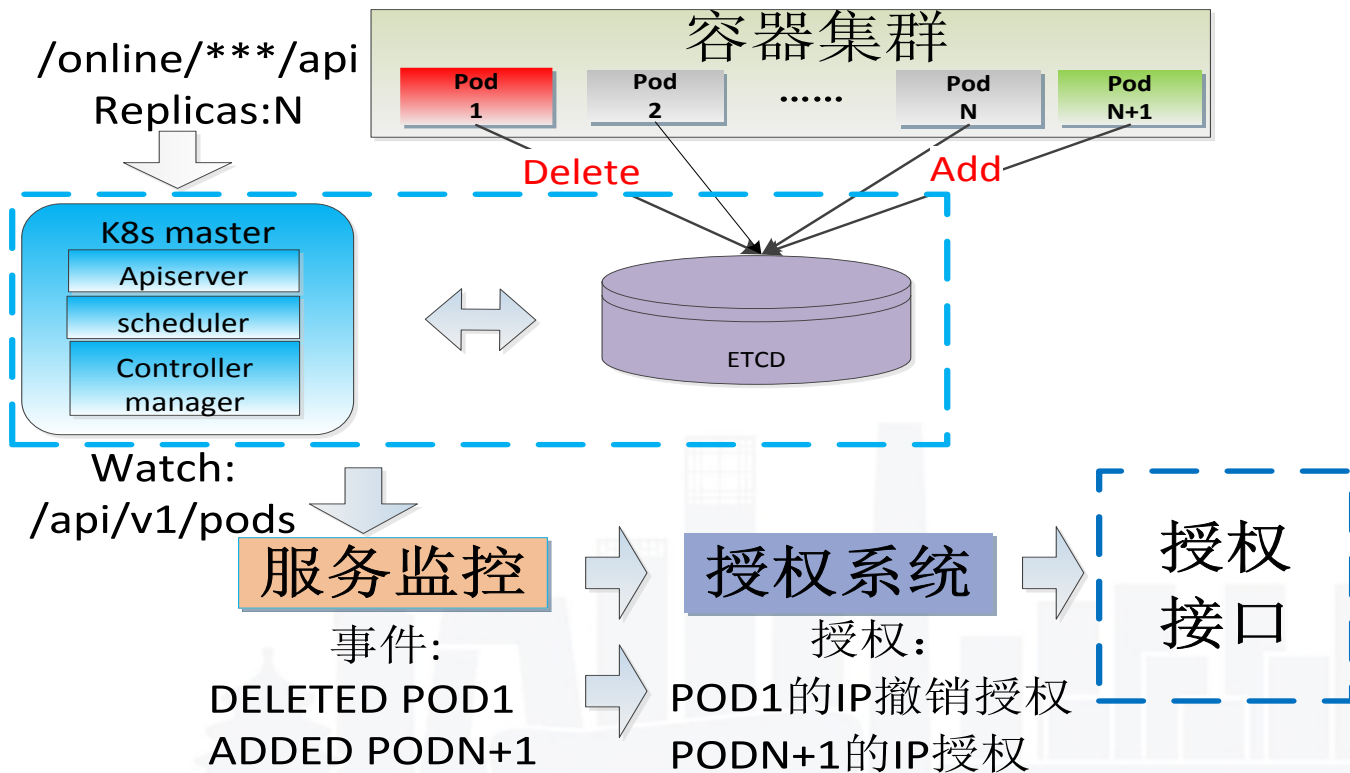
```
# Shard DB common Configuration #
datasource.shard.driverClassName=com.mysql.jdbc.Driver
datasource.shard.username=***
datasource.shard.password=***
# Shard DB Configuration #
datasource.shard01=jdbc:mysql://***/dbname01.mysql:3306/
dbname01?
datasource.shard01.s1=jdbc:mysql://***/s1dbname01.mysql:3306/
dbname01?
datasource.shard01.s2=jdbc:mysql://***/s2dbname01.mysql:3306/
dbname01?
...
```

唯一标识

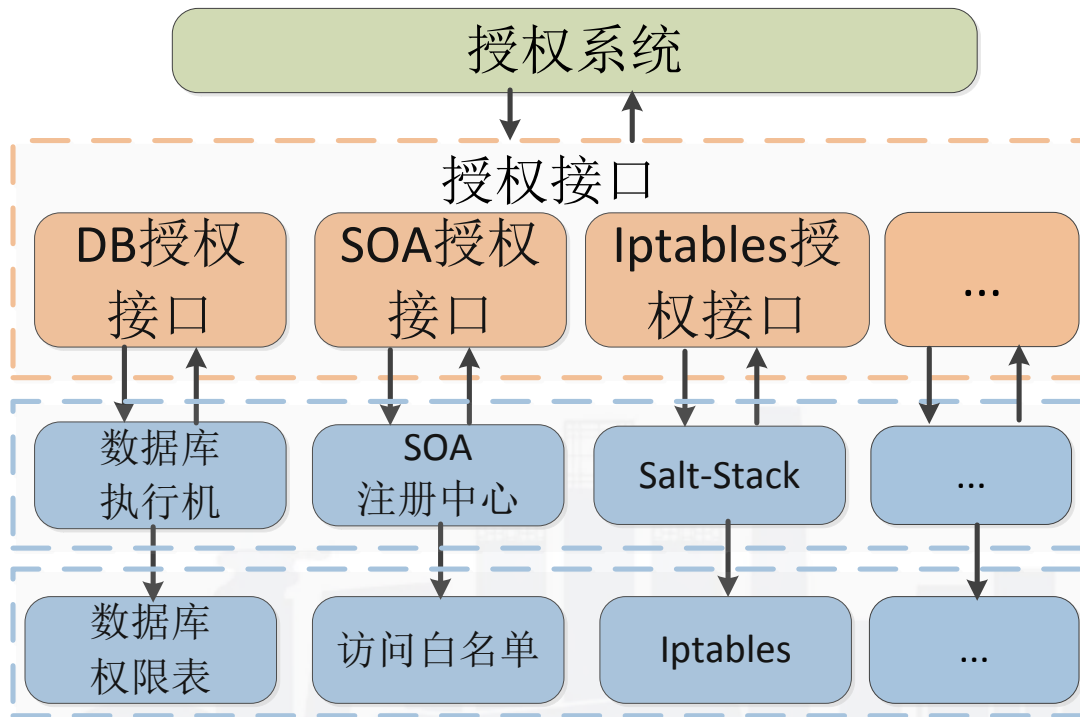
配置中心依赖视图



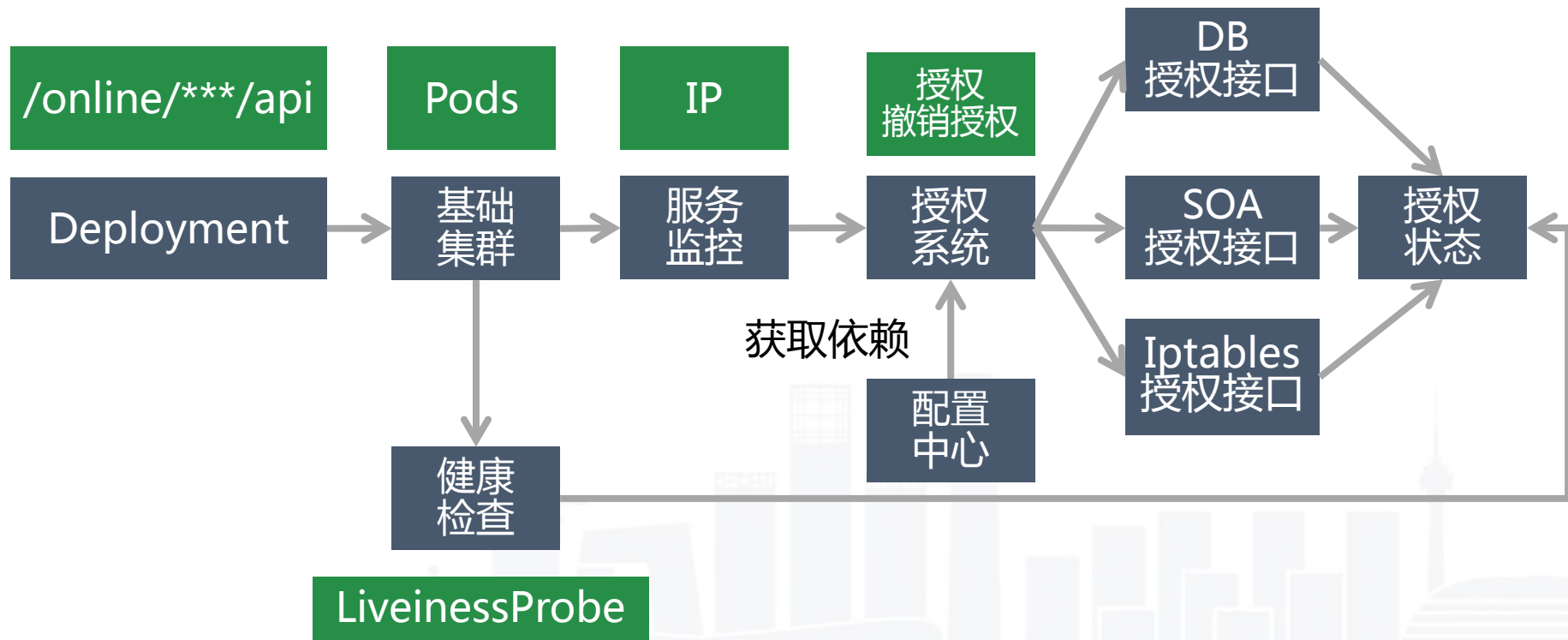
When: 容器变更



How: 授权接口



服务授权示例



实践

基础集群

服务管理

服务授权

灰度发布

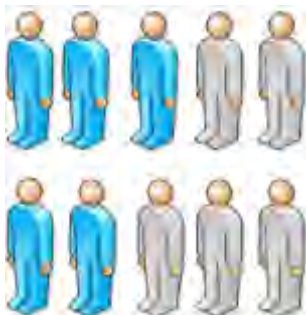


灰度发布



阶段一

推送20%



阶段二

推送50%



阶段三

推送100%

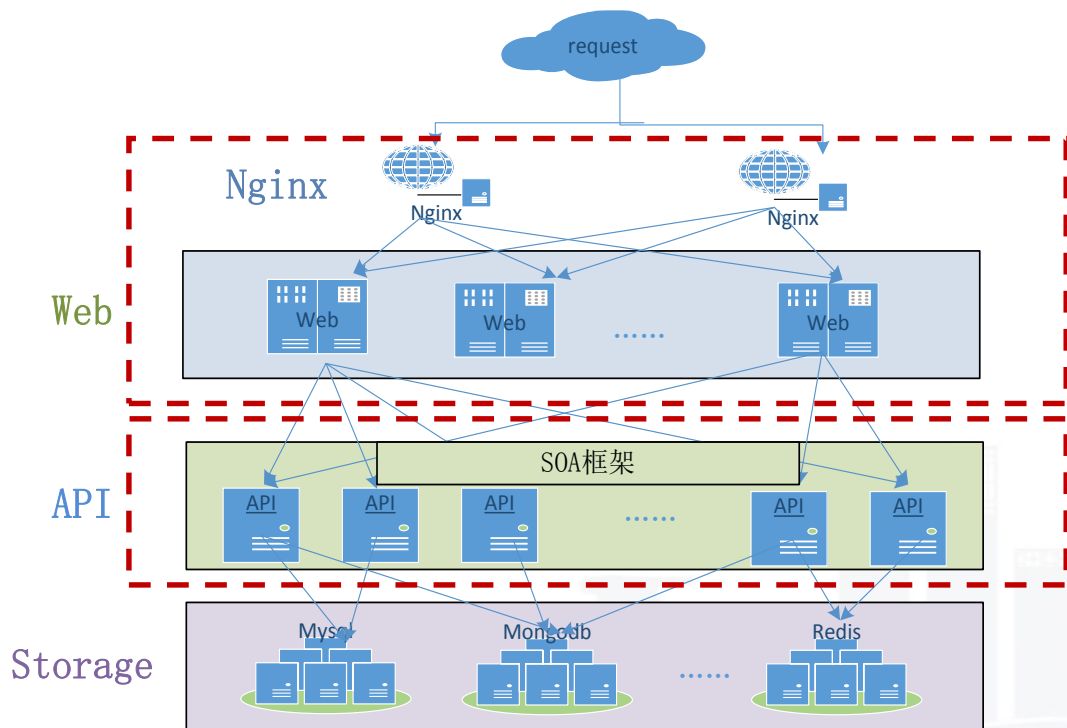
作用

降低风险

提升上线效率

收集使用反馈

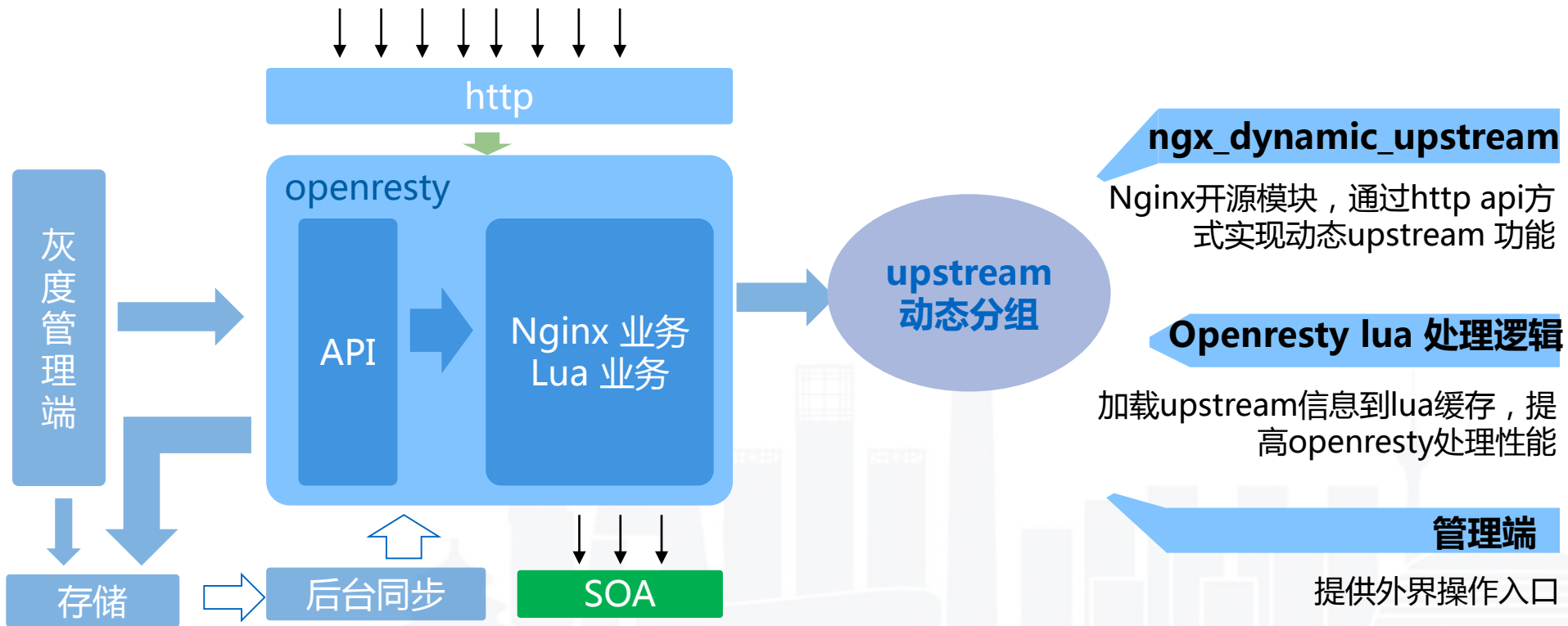
灰度发布



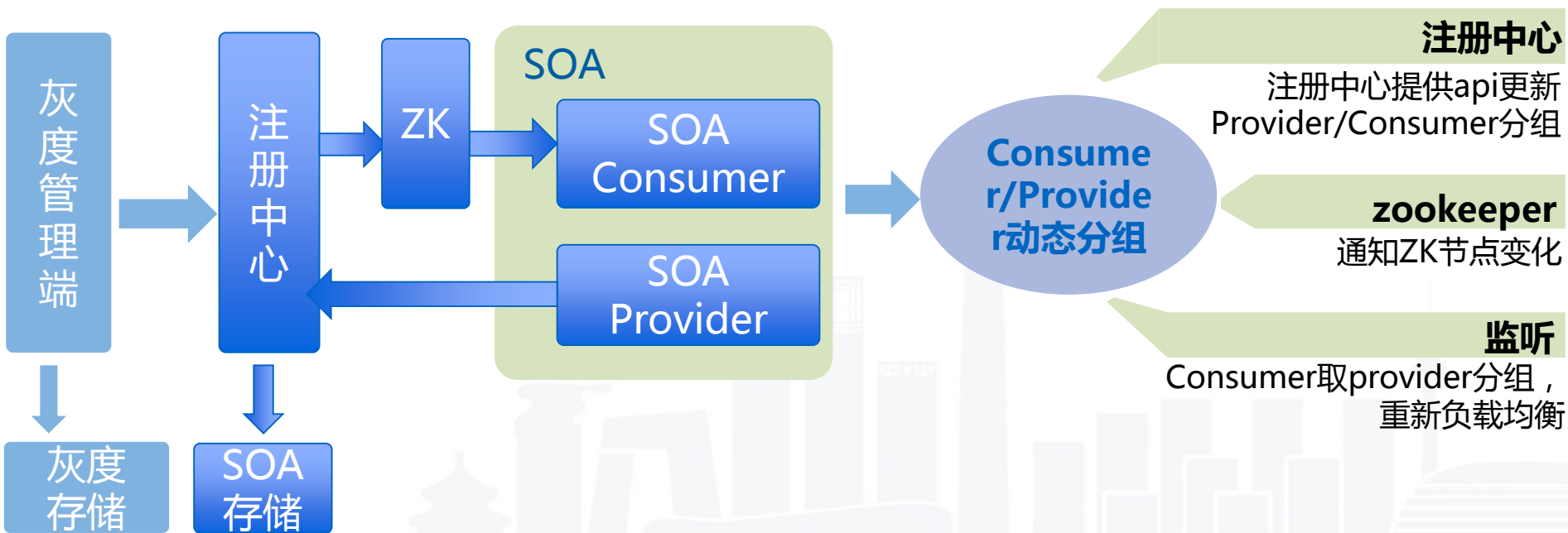
挑战：

- 需要在用户接入层和SOA层统筹考虑灰度机制
- 实现容器类灰度发布功能
 - 滚动升级！=灰度发布
 - 服务发现
 - 非容器兼容

接入层灰度

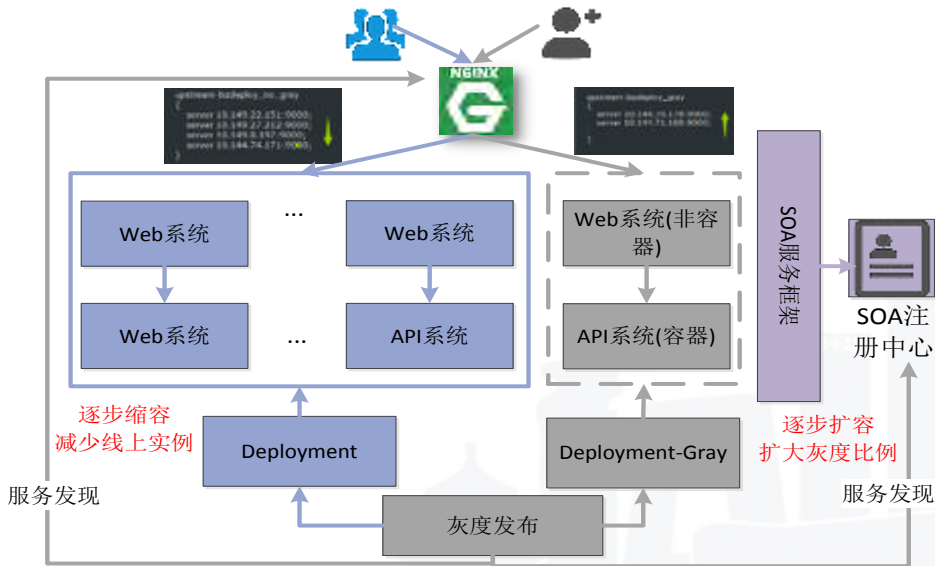


SOA层灰度



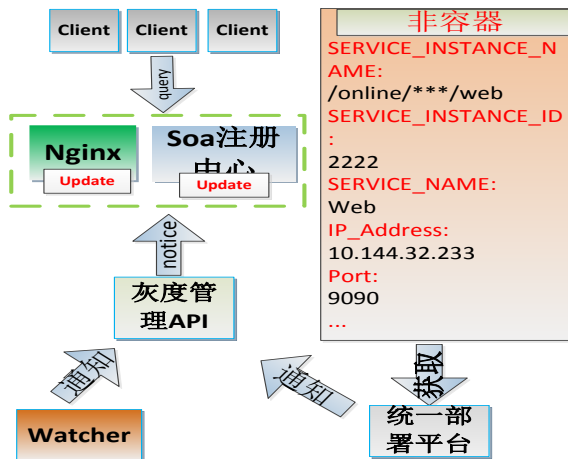
灰度发布

- 灰度deployment
- 服务分组
- 动态调节两个部署实例数
- 服务发现



```

容器
SERVICE_INSTANCE_NAME:
online.***.api
SERVICE_INSTANCE_ID:
1111
SERVICE_NAME:
API
ContainerID:
123***645
IP_Address:
10.144.28.128
Port:
8080
...
    
```

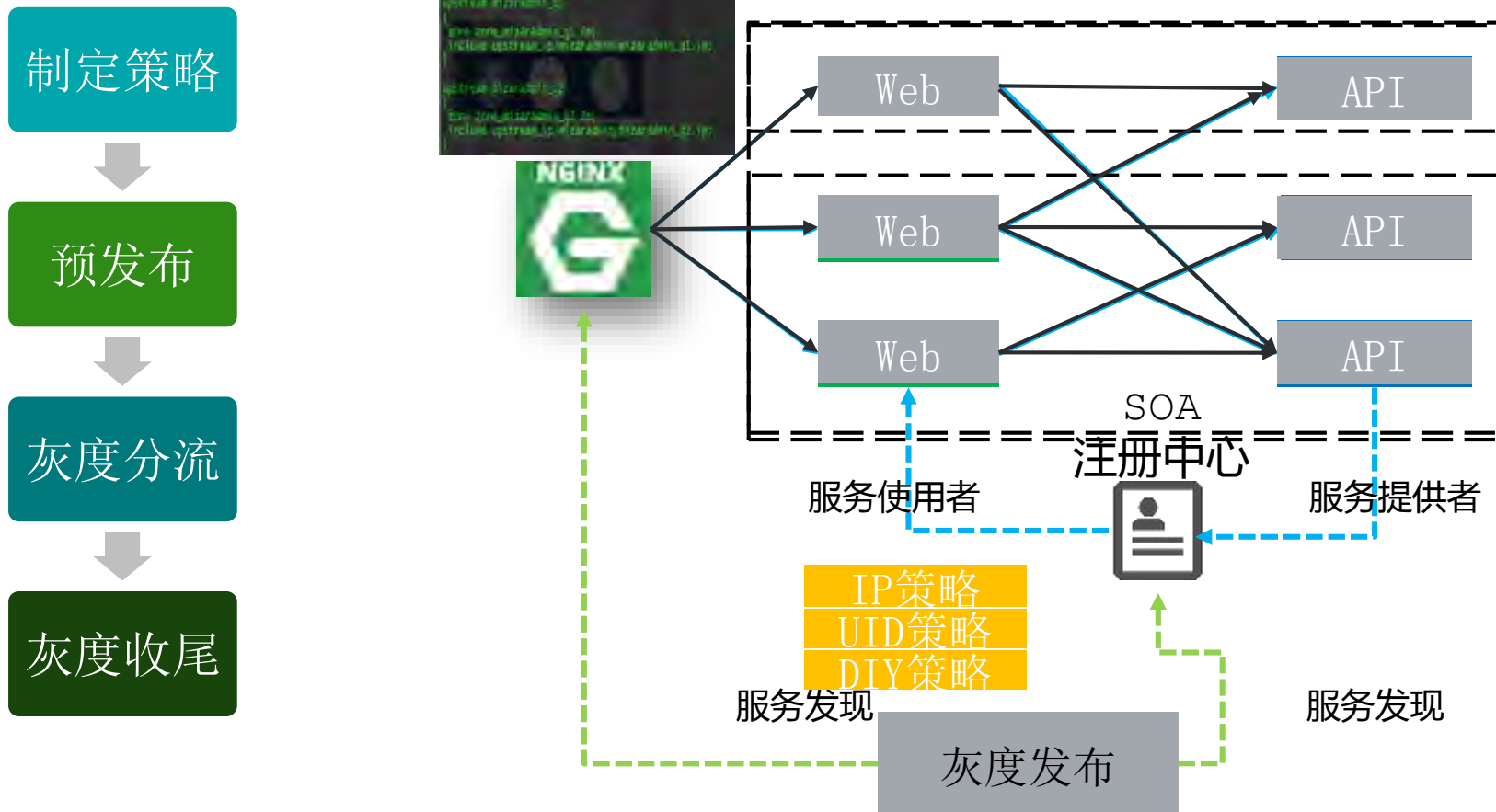


- 服务发现：
- 实例注册ETCD
 - Watch变更
 - Nginx动态update
 - Soa注册中心update

灰度发布流程



灰度发布示例(一组服务)





CONTENT

1 | 商业平台与云

2 | 云平台设计选型

3 | 云平台实践

4 | 总结&展望

总结

- 分享了：
 - 研发商业云平台的背景和目标
 - 我们的技术选型
 - 我们的实践之路：基础集群、服务管理、服务授权、灰度发布



总结

- 效果：
 - 降低了整体成本，缩短项目周期
 - 资源率利用率有效提升
 - 弹性计算能力有效提升
- 经验
 - 需求驱动：前期需求调研的重要性
 - 快速落地：不追求最完美的设计，追求尽快落地，逐步完善
 - 平滑迁移：兼容已有流程、用户习惯
 - 持续优化：优化无止境



展望

- 基础资源容器化
- 联邦集群
- 更智能的调度



Thanks



简历砸来：
bizjob@sogou-inc.com