



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

BEIJING 2017

Swarm的演进与Docker的雄心

陈萌辉

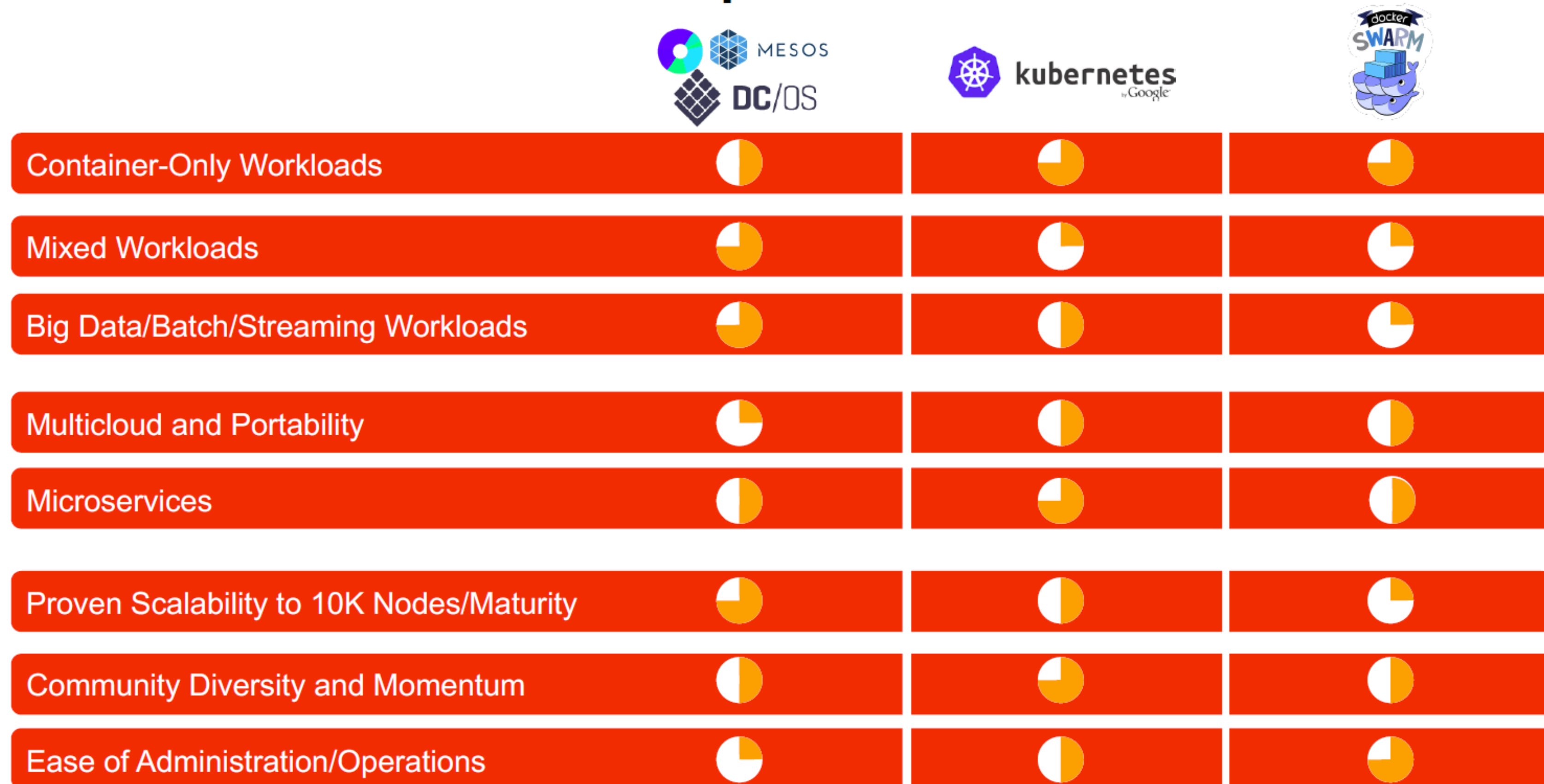
内容提要

- Docker与容器编排
- Swarm简介
- SwarmMode简介
- Swarm在阿里的应用

Docker与容器编排

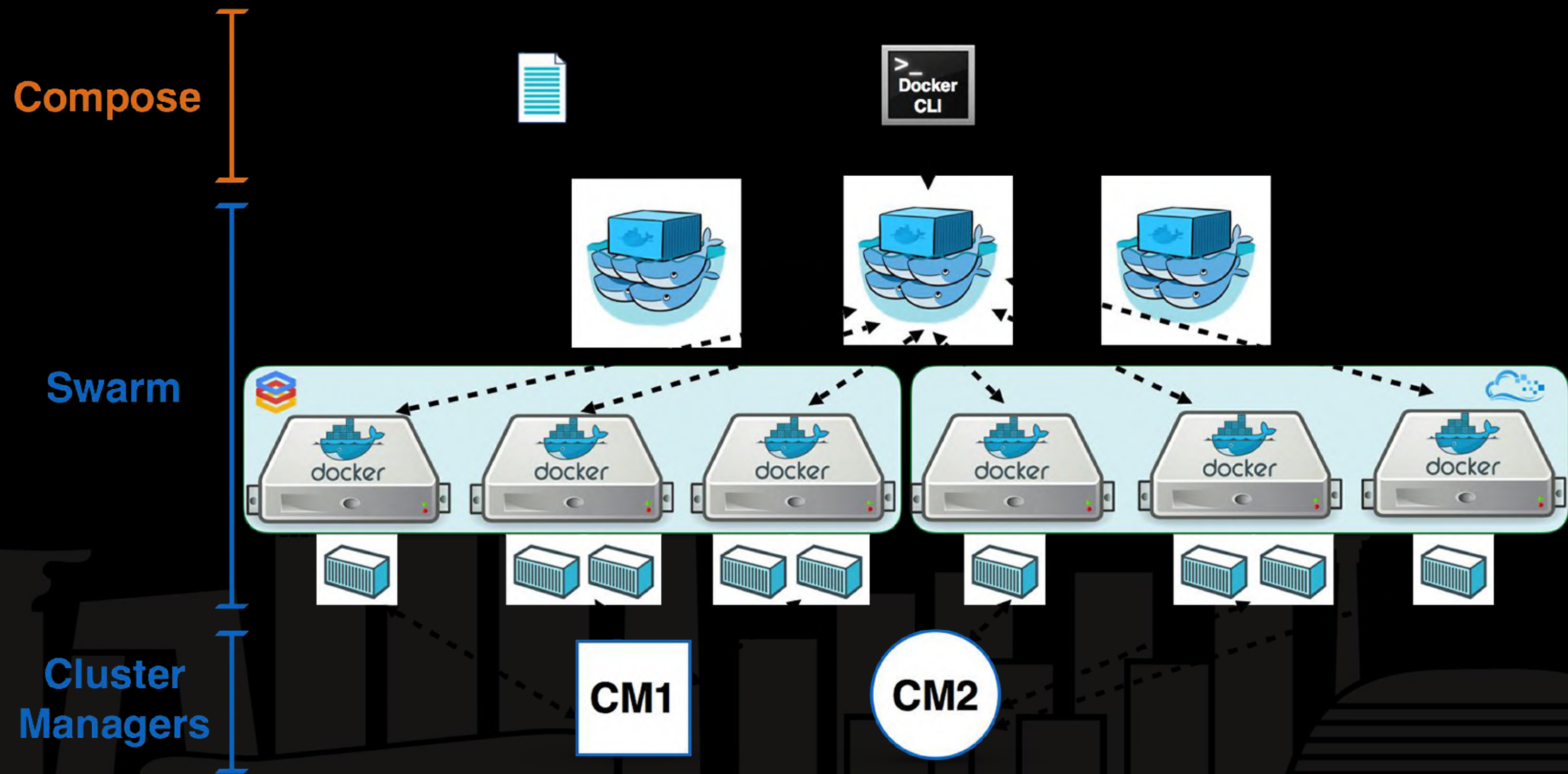
- 容器编排三分天下

How Do Orchestrators Compare?



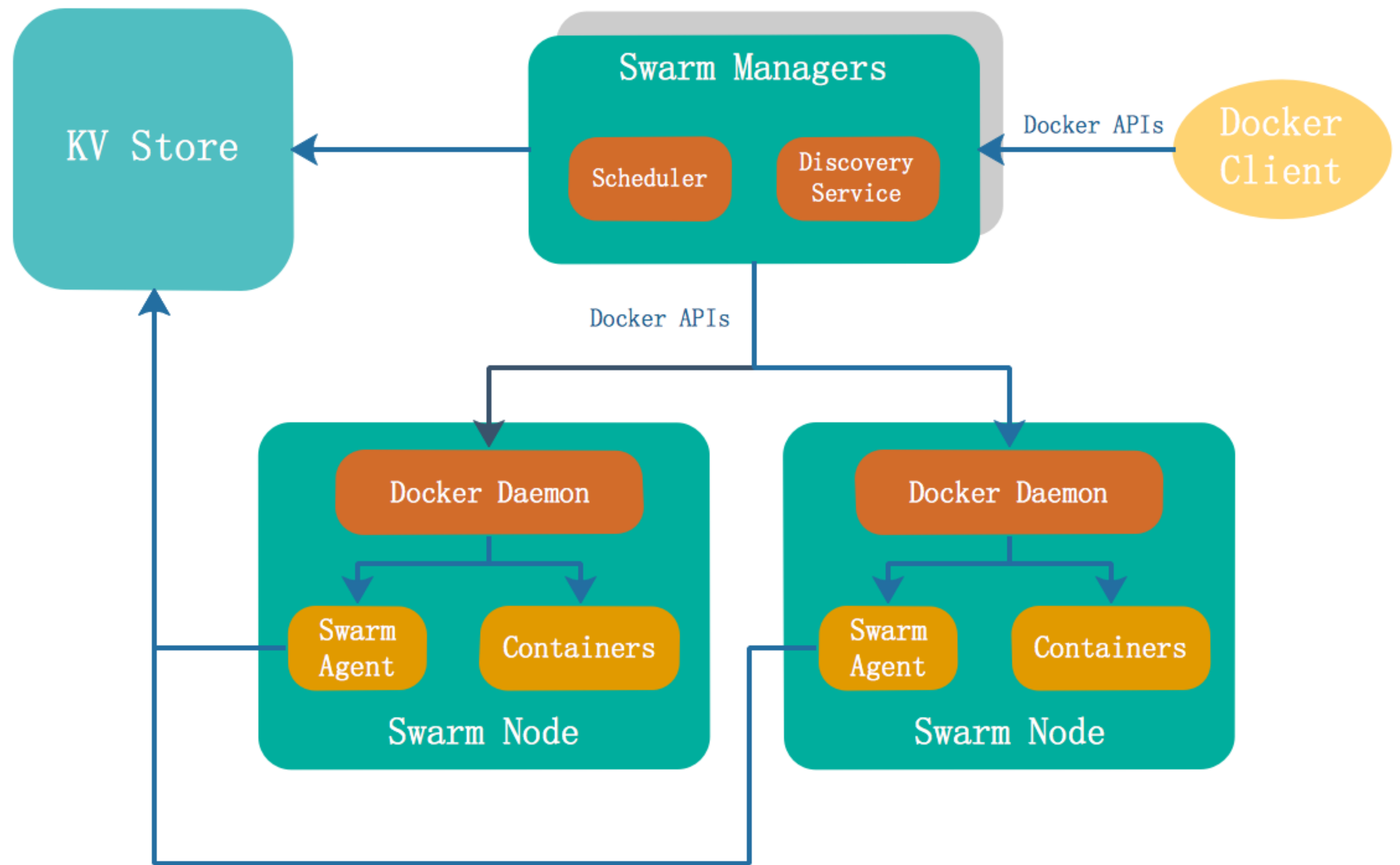
Swarm是什么

- Docker公司继Docker Engine之后的重要产品
- 集群管理系统
- 容器编排与调度系统



Swarm : 架构

- 依赖外部存储来完成节点发现并保证一致性
- Manager只跟Daemon通信，不跟Agent通信
- Manager可以有多副本



Swarm : 架构

- Swarm Manager的高可用设计
 - 一主多热备
 - 所有manager都同时连接所有Daemon
 - 备转发请求至主
 - 依赖外部KV选主
 - 抢锁 -> 保活

Swarm : API

- 集群类

- Info

- events

- 容器类

- get/list

- create

- start

- stop

- stats

- exec

- ...

- 镜像类

- get/list

- create

- delete

- push

- tag

- ...

- 数据卷类

- get/list

- create

- delete

- 网络类

- get/list

- create

- delete

- connect

- disconnect

Swarm : API

- 高度兼容Docker Engine API
 - 集群级汇总
 - 转发到相应节点的Docker Daemon
 - 在集群中广播
- 单个容器级别的API

Swarm : 调度

- 资源调度
 - 资源维度 : CPU / Memory / 端口
 - CPU / Memory支持超卖
 - 调度策略 : spread / binpack
 - 不支持优先级、抢占

Swarm : 调度

- 节点约束
 - 节点名 : `constraint:node===XXX`
 - 标签 : `constraint:key===value`
- 亲和性
 - 镜像 : `affinity:image===foo`
 - 服务 : `affinity:service===foo`

Swarm : 总结

- 部署简洁
 - 只依赖KV Store和Docker Daemon
 - 所有组件都容器化
- 高效友好的用户交互
 - 高度兼容Docker Engine API, 可直接使用Docker Client
- 灵活的约束与亲和性描述

Swarm : 总结

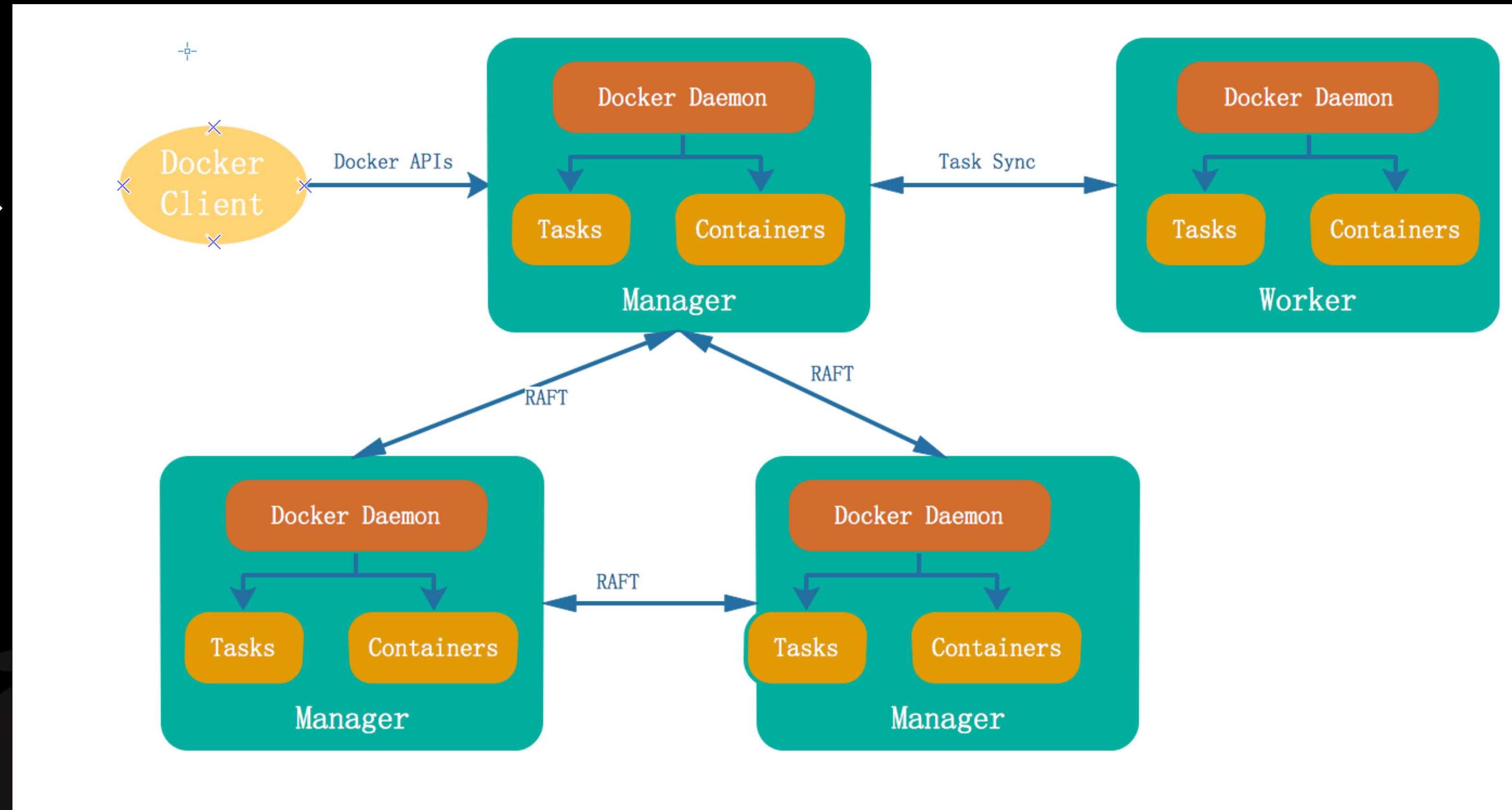
- 不足
 - 容器级别的API，抽象层次不够
 - 响应式设计，无后台程序
 - overlay网络，对KV Store压力大

SwarmMode : 进化

- Docker 1.12版开始提供
- 将Swarm的集群管理、容器调度功能集成进Docker Enging
- 提供Service级别抽象
- 自带负载均衡

SwarmMode : 架构

- 无任何外部依赖
- Daemon身兼Engine、Manager、Agent三职
- Managers之间通过RAFT协议组成分布式强一致性KV Store
- Manager与Worker的Daemon不通信



SwarmMode : 架构

- 高可用设计
 - Manager数量需要 ≥ 3
 - 一主多热备
 - 动态添加 / 删除Manager



SwarmMode : API

- Swarm
 - init
 - token
 - join
 - leave
- Service
 - get/list
 - create
 - ps

SwarmMode : API

- 两类API
 - Swarm、Service、network类，只有Manager能处理
 - 容器、镜像、数据卷类，所有节点都能处理
- 高度兼容旧API



SwarmMode : Service

- Service、Task、Container三级概念
 - Service : 相同功能的一组容器
 - Task : 任务调度单元, 由Manager生成, 同步至Worker
 - Container : Task落地
- Rolling Update

SwarmMode : Service

- Replicated Service
 - 用户指定副本数
 - Reconciled : 自动确保副本数
 - constraint
 - node.id node.hostname
 - node.role
 - node.labels engine.labels

SwarmMode : Service

- Global Service
 - 每个节点有且仅有一个容器
 - 添加加点时自动扩展
 - 可附加constraint

SwarmMode : Service

- 网络模型
 - 仅支持overlay网络，同一网络内，服务名、容器名可解析
 - 一个服务一个网络
 - 服务发现：不同服务可加入同一个网络

SwarmMode : RoutingMesh

- Service自带的负载均衡
- 两种模式
 - VIP : 每个服务一个VIP , 通过LVS实现 ; 服务名解析至VIP
 - DNS : 服务名解析至容器IP , RoundRobin方式
- 服务发生变化时 , 自动调整后端

SwarmMode : 总结

- 部署特别简洁
 - 无任何依赖，只需安装Engine + 一个命令
- 无中心架构
- 部署高可用服务
 - Service API + RoutingMesh
- Secure by default :
 - 自带证书颁发、更新功能，Manager与Worker之间通过SSL连接

SwarmMode : 总结

- 不足
 - 只有Service级抽象，Stack级抽象仍无API
 - 不支持有状态服务
 - Service API有很多容器特性不支持，如host network、host pid、privileged等
 - 无法自举，需要手工init

Swarm在阿里的应用

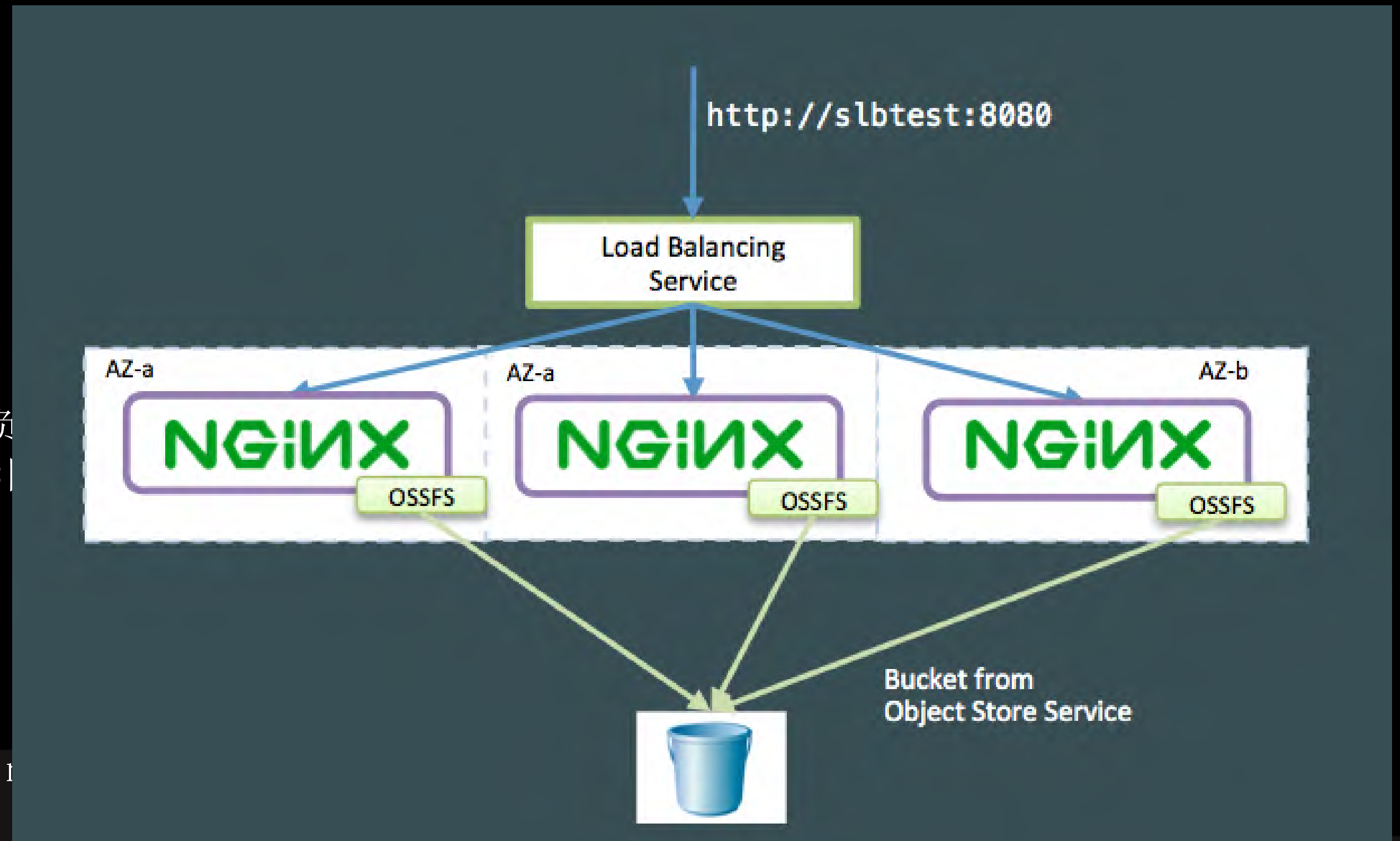
- 支付宝， 淘宝的应用运维Docker化
- 阿里云容器服务
- 阿里云高性能计算 HPC

Swarm在阿里的应用

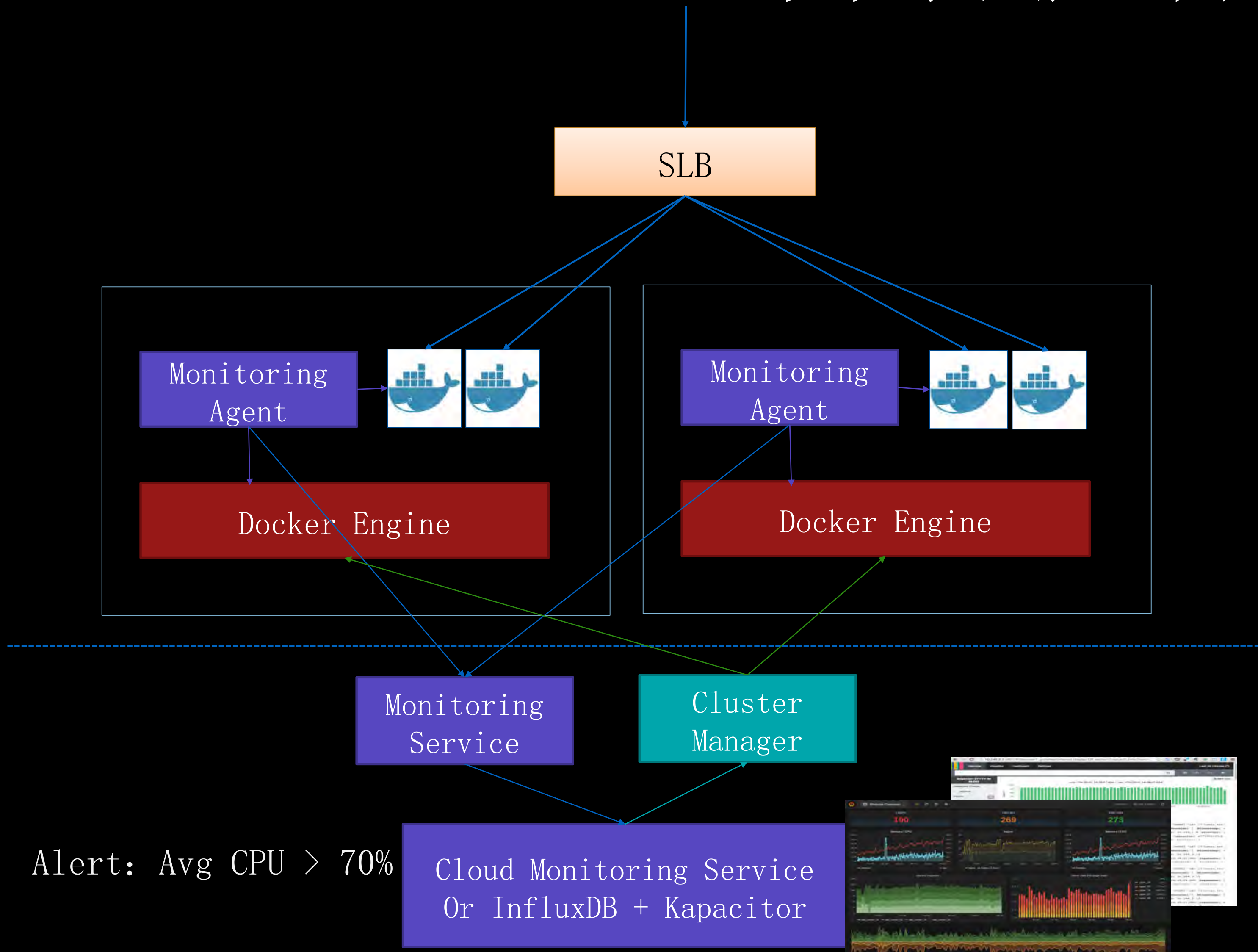
- 阿里云容器服务的扩展
 - 规模与性能
 - 声明式扩展，支持负载均衡、日志、监控
 - 集成共享存储
 - 离线与定时任务

编排与插件

```
version: "3.0" #支持v1 v2 v3 compose模板
services:
  nginx:
    image: nginx:latest
    deploy:
      mode: replicated
      replicas: 3
    ports:
      - 8080:80
    labels:
      aliyun.lb.port_8080` : tcp://slbtest:8080 #负
      aliyun.log_store_dbstdout: stdout #
      aliyun.log_store_varlog: /var/log/*.log
    volumes:
      - 'website:/usr/share/nginx/html'
    volumes:
      website:
        driver: ossfs #共享存储数据卷, 支持oss、r
        driver_opts:
          bucket: acs-sample
```



自动扩容



• 声明式自动扩容

aliyun.auto_scaling.max_cpu: 70

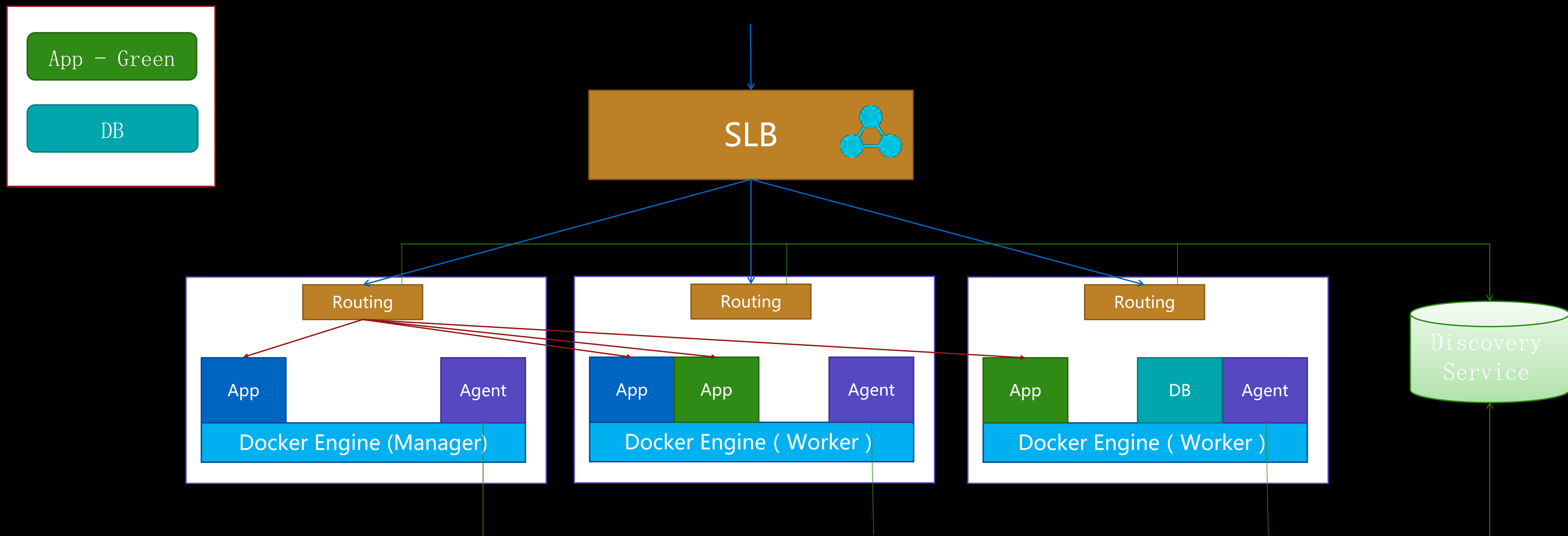
aliyun.auto_scaling.step: 2

• 监控插件

输入: nagios, apache, docker, UDP, ...

输出: Influxdb, prometheus, kafka

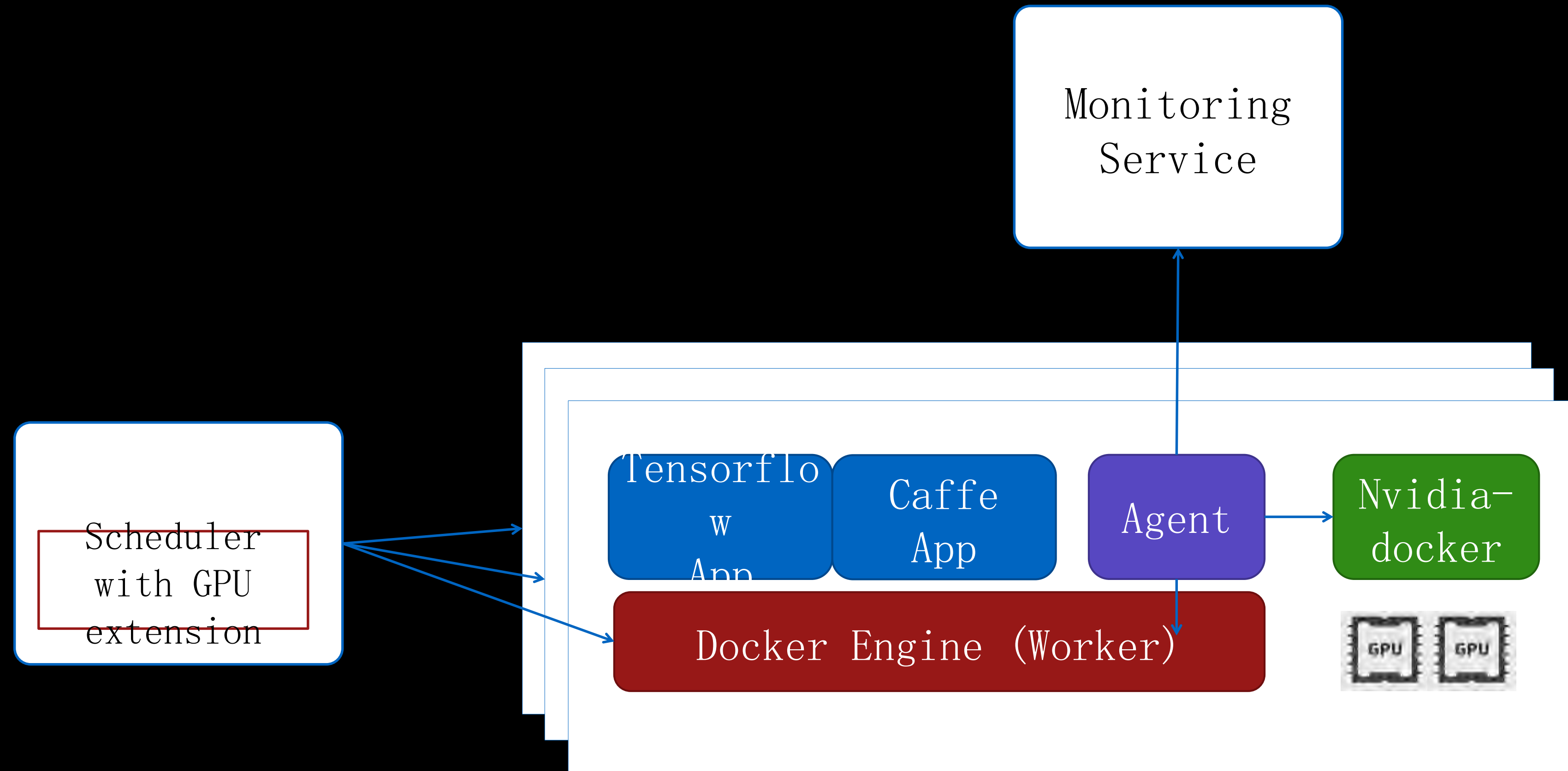
蓝绿发布



机器学习

基于Docker的GPU/CPU混合调度

```
version: '2'  
services:  
  inception:  
    image: acs_sample/inception:demo  
    volumes:  
      - inception_model/inception_model  
    labels:  
      - aliyun.gpu=2  
    ports:  
      - "9000:9000"  
volumes:  
  inception_model:  
    driver: nas
```



更多信息

云栖社区



阿里-Docker客户技术支持

862人



扫一扫群二维码，立刻加入该群。



关注QCon微信公众号，
获得更多干货！

Thanks!



主办方 **Geekbang** > **InfoQ**
极客邦科技