Yourun Cloud

微服务容器化的挑战和解决之道

马洪喜 有容云联合创始人兼首席架构师

深圳市云舒网络技术有限公司

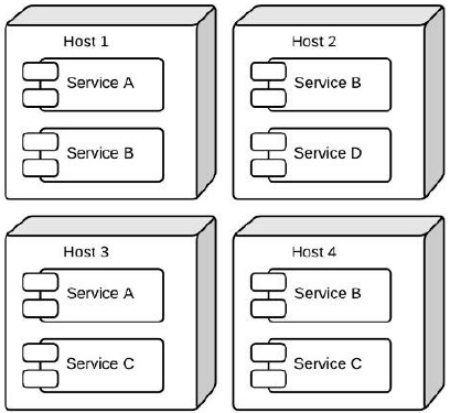
马洪喜 —— 有容云首席架构师

- 2016：有容云联合创始人、首席架构师
- 2015：Rancher Labs 中国区技术负责人
- 2010：思杰公司架构师，构建桌面云、IaaS云
- 2006：Oracle公司Linux、虚拟化团队Team Lead
- 2000：多平台、多语言“全栈”Developer

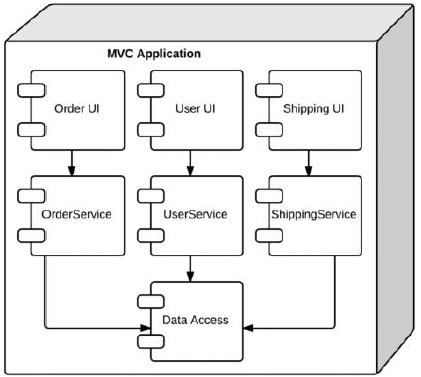
传统单体架构VS微服务架构

微服务架构是由一系列职责单一的细粒度服务构成的分布式网状结构，服务之间通过轻量机制进行通信，一般以集群方式提供服务。

微服务架构



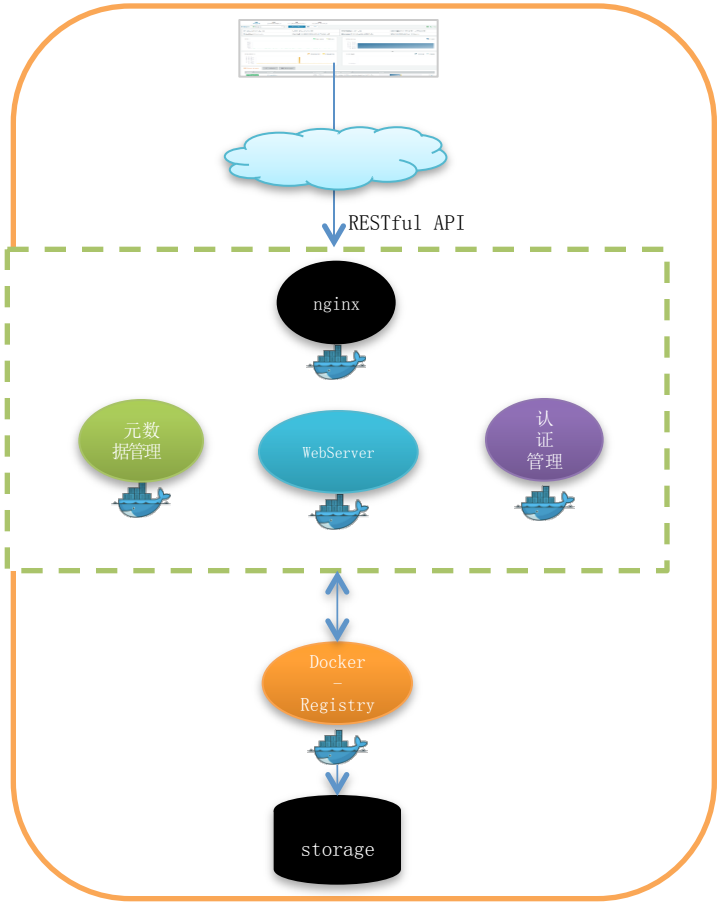
传统单体架构



- 模块可独立提供服务，边界清晰、易于维护
- 不同语言编写，易于引入新技术
- 微服务商店模式，快速的组合和重构
- 模块间松耦合，不同SLA保障计划
- 更好的可扩展性和鲁棒性

- 加载、编译耗时长
- 代码管理复杂
- 横向扩展难
- 各模块之间的耦合程度高

微服务架构示例 —— 有容云 AppHouse



设计理念

•微服务架构，容器化
•灵活配置，随需而变
•模块复用，便捷高效

Index模块

•WebServer接收前端访问
•用户访问认证
•Token管理
•元数据管理（访问权限、统计等）

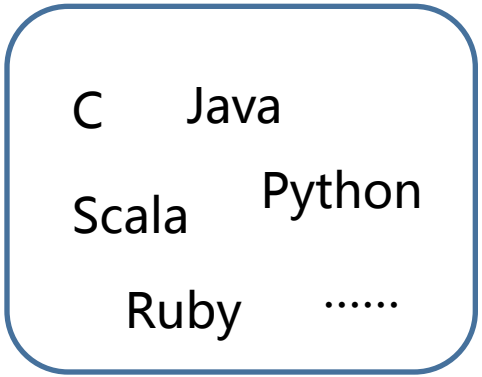
Docker-Registry模块

•镜像层次存储
•接入外部存储

后端存储

•支持公有对象存储
•支持本地文件存储/对象存储

微服务给DevOps带来的挑战 1



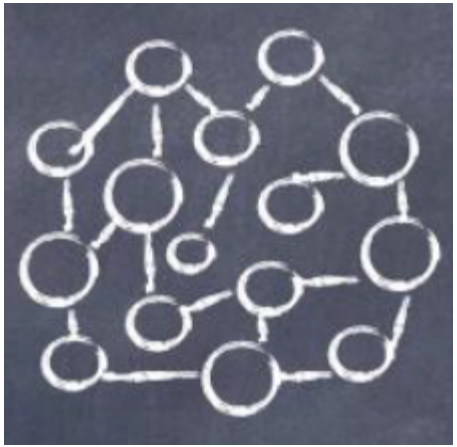
服务使用不同的编程语言将开发

为每种语言准备编译环境

为每个服务的部署准备不同的库和框架

让服务的开发和部署变得非常复杂

微服务给DevOps带来的挑战 2

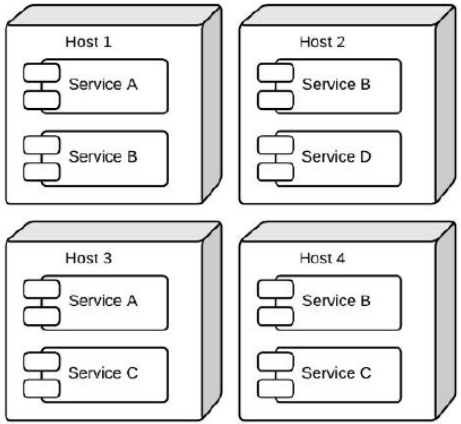


大量的微服务模块

复杂的版本管理和Bug跟踪

间接导致项目管理成本增加

微服务给DevOps带来的挑战 3

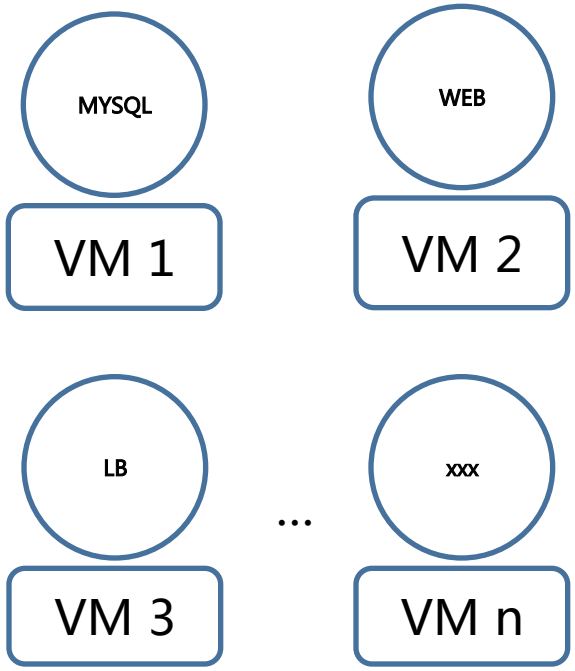


与生俱来的分布式架构特性，
服务分布在多个主机集群上

很难持续跟踪指定服务究竟运行在哪台主机上

后续的系统维护不方便

微服务给DevOps带来的挑战 4



一个服务一个虚拟机

随着微服务架构不停的横向扩展，主机数量将以一个非常恐怖的速度增长

最小规模的主机配置可能也会超过了很多微服务对资源的要求，从而造成了超量配置并浪费开销

容器可以轻松实现微服务化后的DevOps

Netflix云架构总监Adrian Cockcroft
“微服务和docker的结合是一种颠覆”

Docker可以为微服务提供一个完美的运行环境

独立性：
一个容器就是一个完整的执行环境，不依赖外部任何的东西

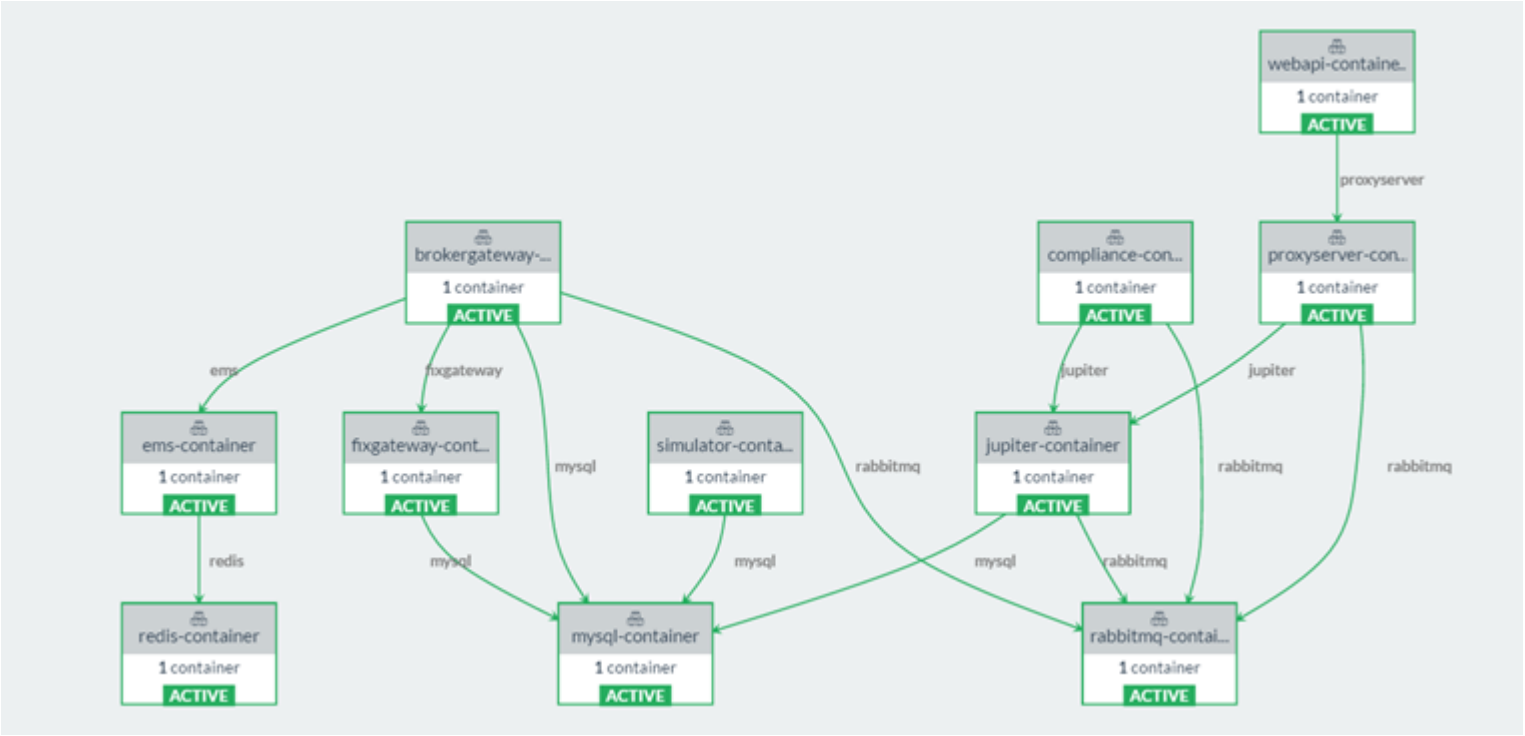
细粒度：
一台物理机器可以同时运行成百上千个容器，其计算粒度足够的小

快速创建和销毁：
容器可以在秒级进行创建和销毁，非常适合服务的快速构建和重组

完善的管理工具：
数量众多的容器编排管理工具，能够快速的实现 服务的组合和调度

微服务化与容器化，谁先谁后?

微服务进程中的一种过渡状态



支持微服务的容器管理平台需要解决的问题

基础架构服务管理

- 容器主机管理
- 容器SDN网络的实现
- 服务注册和发现
- 容器负载均衡
- 容器和服务的健康检查
- 服务升级和灰度发布
- 容器持续存储和数据管理

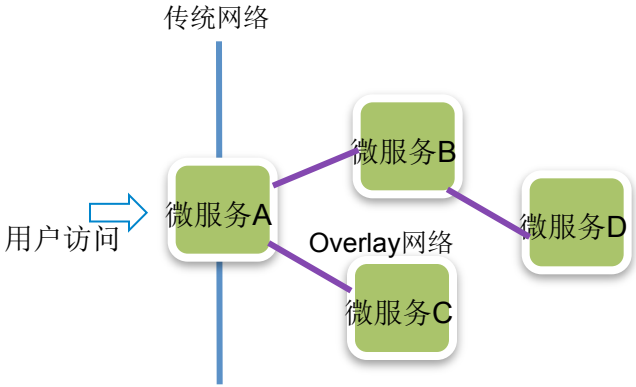
生命周期管理和团队协作

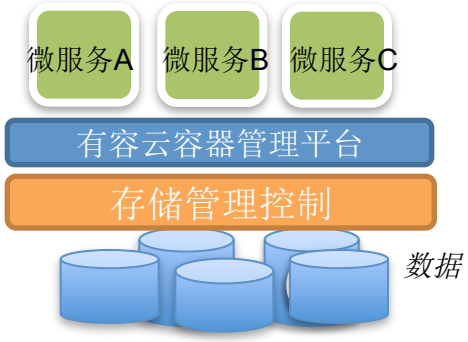
- 容器生命周期管理
- 容器运行状态监控
- 访问容器的日志
- 访问容器的Shell
- 鉴权系统和账户管理
- 容器环境和多租户

配置管理和部署支持

- 应用的配置管理
- 丰富的Compose文件格式
- 企业镜像库管理
- 功能完备的API接口
- 兼容Docker CLI命令行

面向微服务的容器网络和存储实现架构讨论





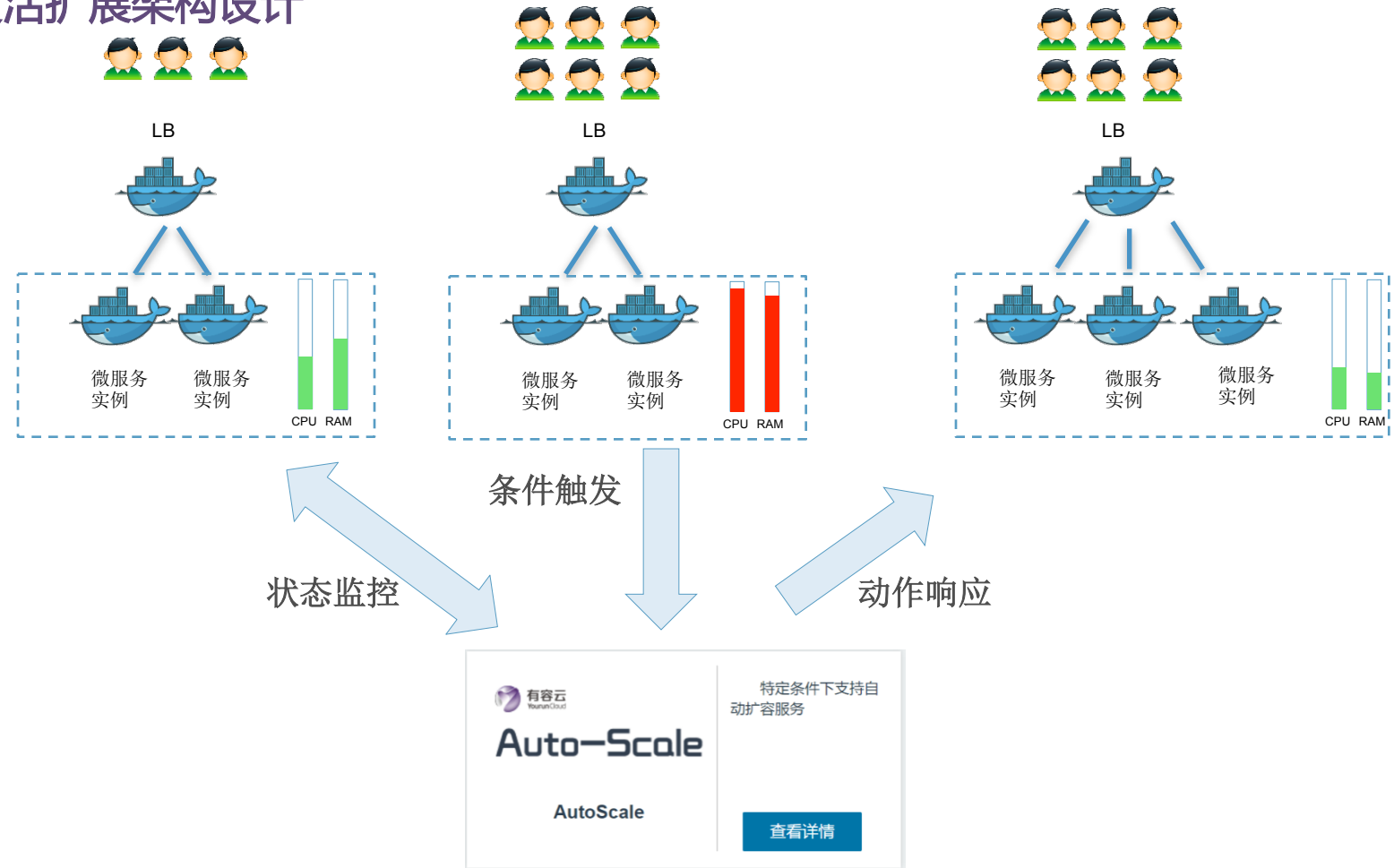
网络服务

- ✓ 微服务内部通讯采用Overlay网络以节省大量IP资源，实现网络上租户隔离
- ✓ Frontier 服务层采用传统的扁平网络模式与传统网络层实现无缝对接

存储服务

- ✓ 满足不同微服务对后端存储的不同需求
- ✓ 按微服务本身的业务等级设置不同的存储SLA
- ✓ 实现针对微服务的快照、备份、回滚等数据策略

微服务的灵活扩展架构设计

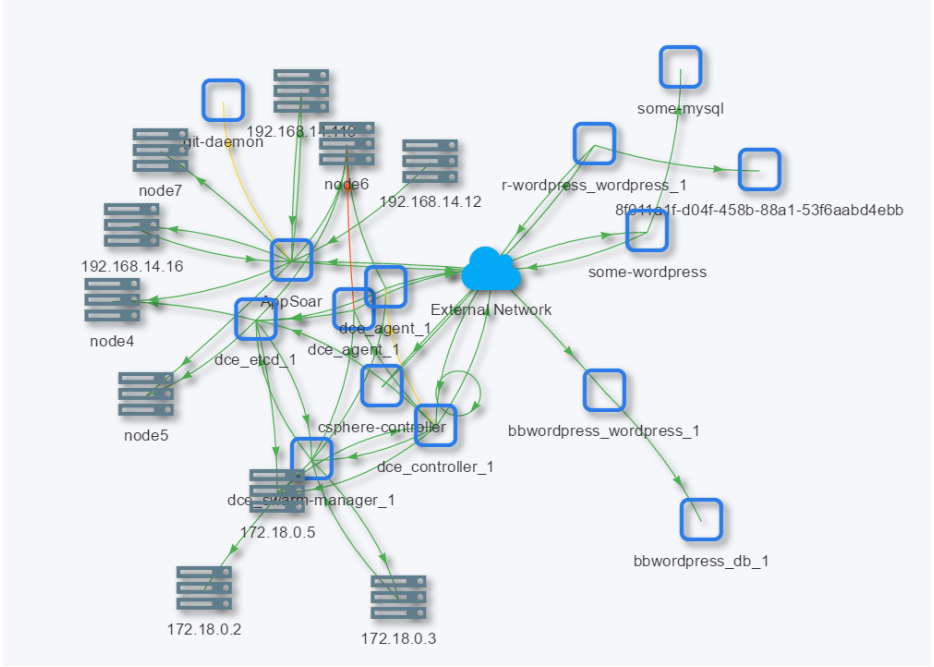


微服务容器化下的安全架构考虑

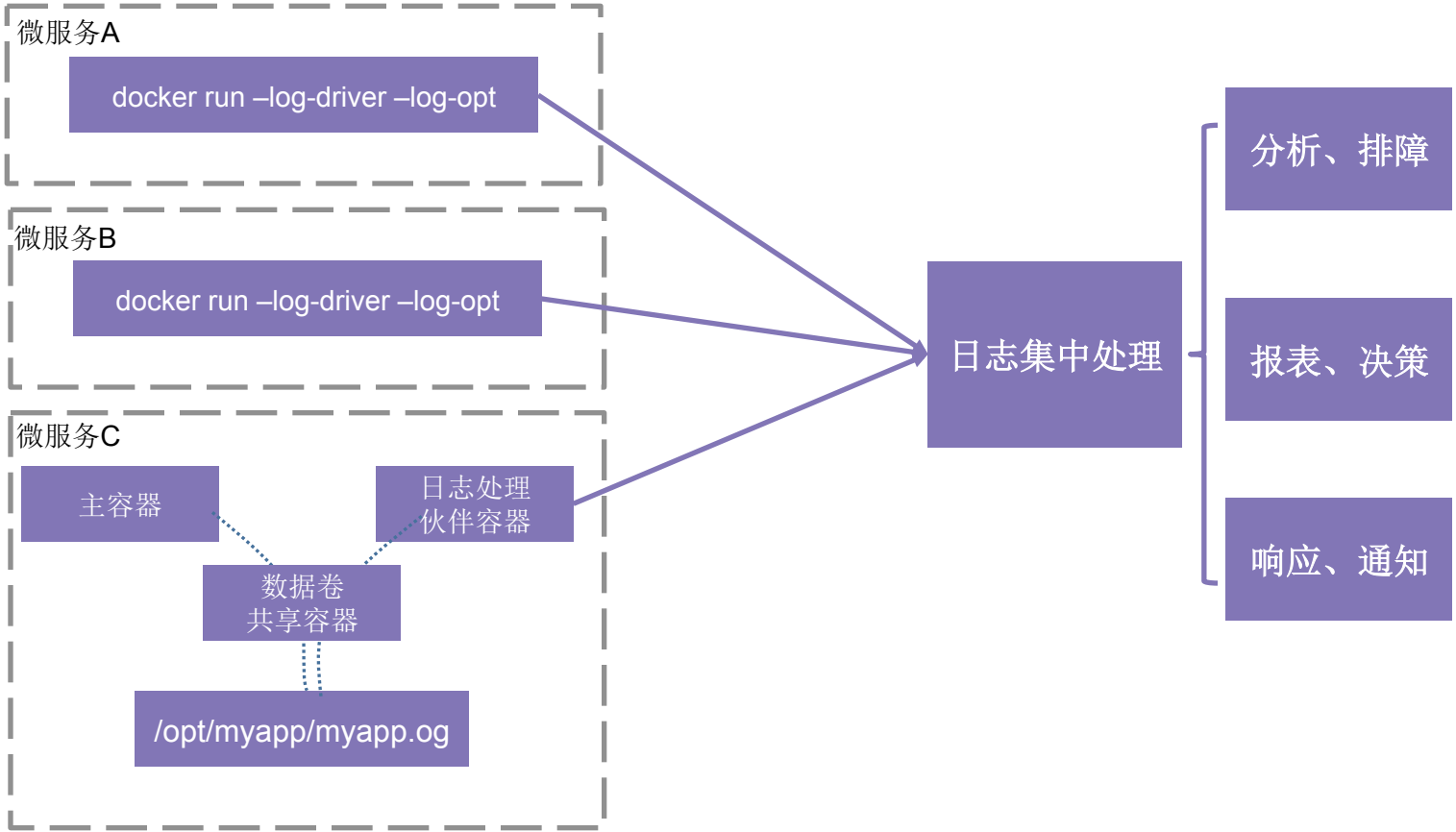
- 需要以应用（微服务）为中心
- 需要有能见度(Visibility)
- 更智能的自学习安全策略
- 运行时漏洞热补丁技术
- 静态/动态镜像扫描潜在风险

主要威胁

类型	严重程度	应用程式
TCP.Small.Window	high	HTTP
TCP.No.Client.Data	medium	HTTP
TCP.No.Client.Data	medium	
TCP.Split.Handshake	medium	



微服务的日志聚合设计


微服务A
docker run –log-driver –log-opt
微服务B
docker run –log-driver –log-opt
微服务C
主容器
日志处理
伙伴容器
数据卷
共享容器
/opt/myapp/myapp.og
日志集中处理
分析、排障
报表、决策
响应、通知

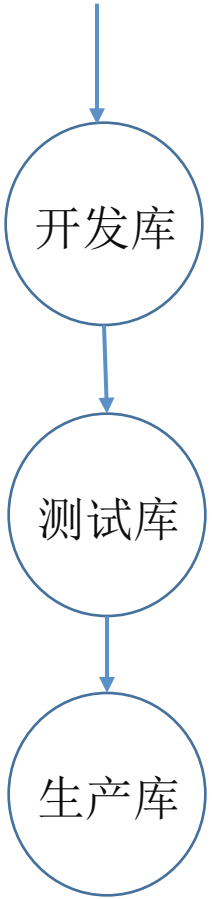
私有镜像服务在DevOps环境的选择

以有容云AppHouse为例

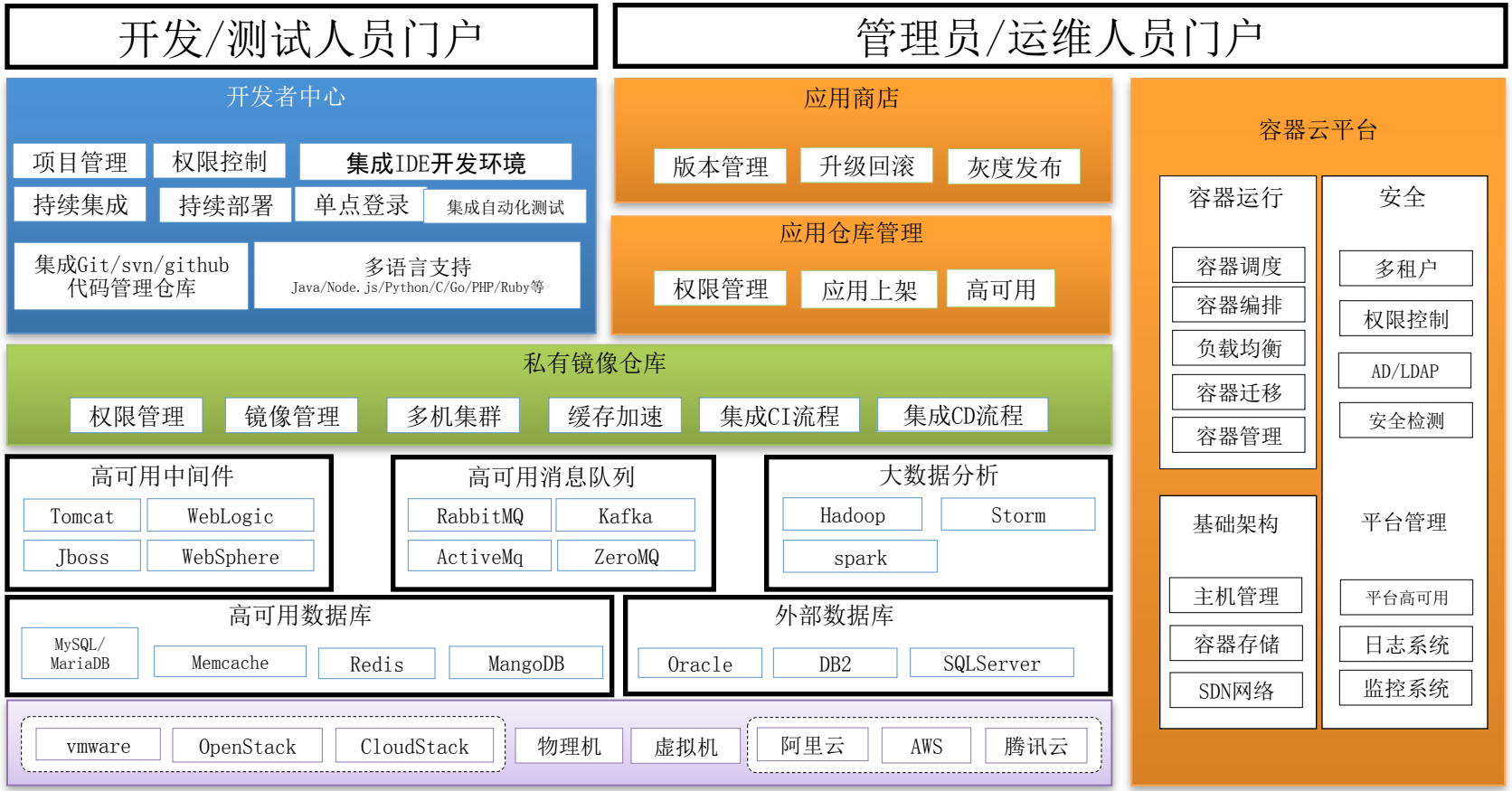
私有内网可扩展存储
优化的开发团队协作支持
简洁的部署管理
支持企业级目录服务
支持OAuth用户认证
细粒度的权限控制体系

与持续集成工具完美对接
零宕机时间部署
节点和数据中心间负载均衡支持
支持安全弹性的链接
高可用冗余架构

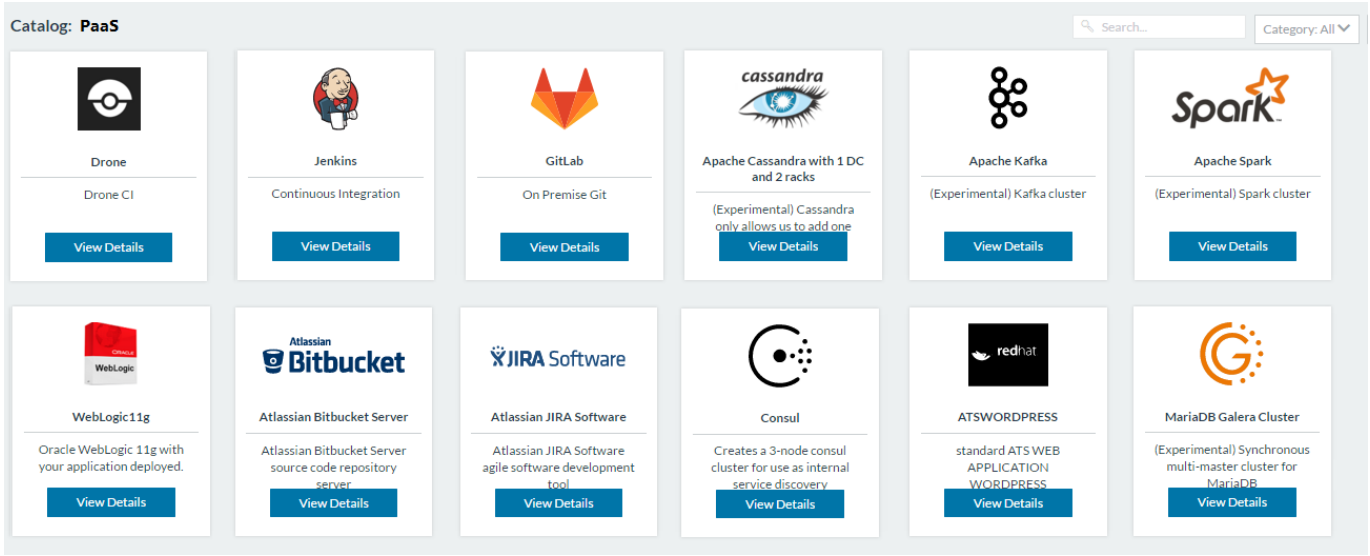
镜像使用率监控，热点镜像提示和告警
存储空间和网络带宽监控
传输断点续传，中断提醒



容器驱动的轻量级PaaS解决方案架构示例



通过灵活扩展、可配置的方式实现微服务应用商店模型



微服务模块以插件或是他人分享方式提供

微服务模块以API或是手工方式消费