SDCC 2016

中国软件开发者大会

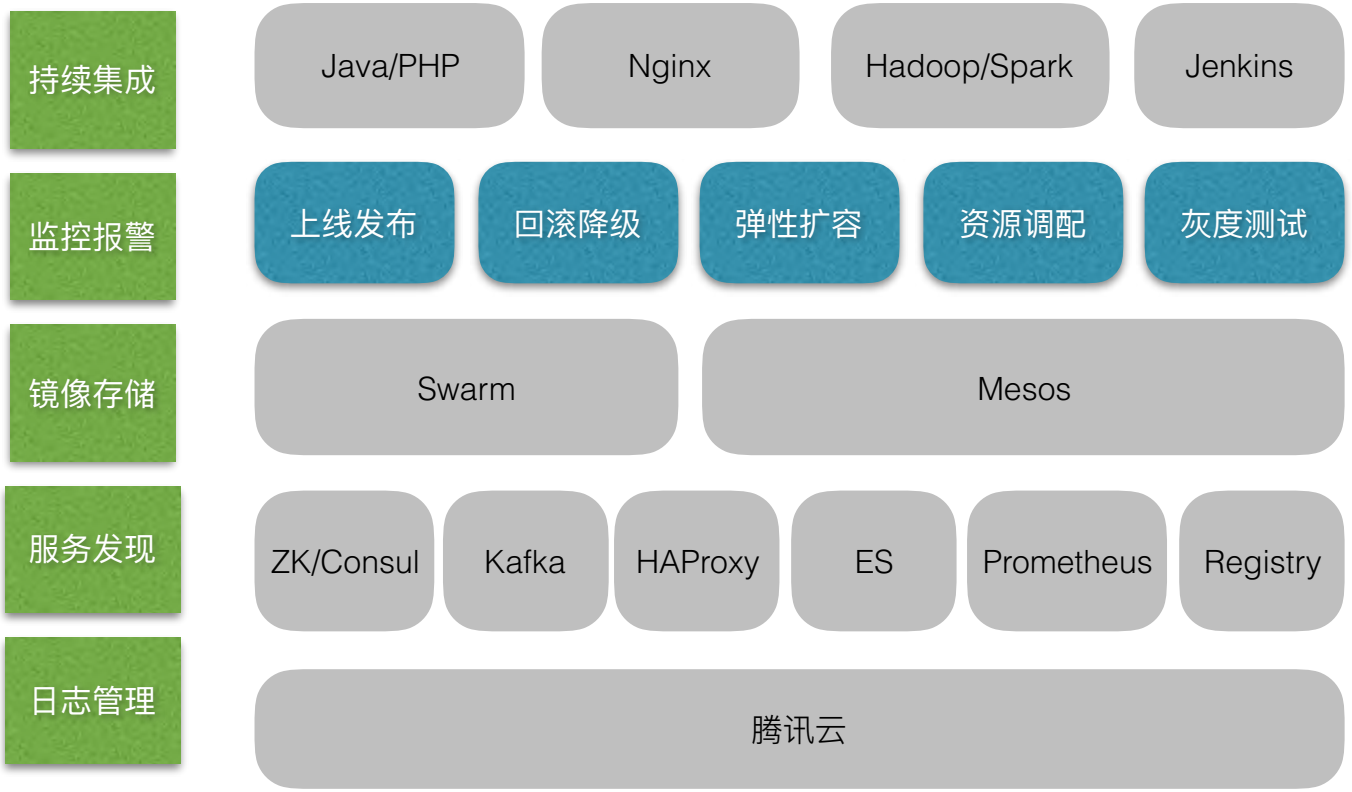
SOFTWARE DEVELOPER CONFERENCE CHINA

Docker应用

Agenda

1. 基于容器的架构
2. Mesos相关技术使用
3. 遇到的一些问题

1. 基于容器的架构

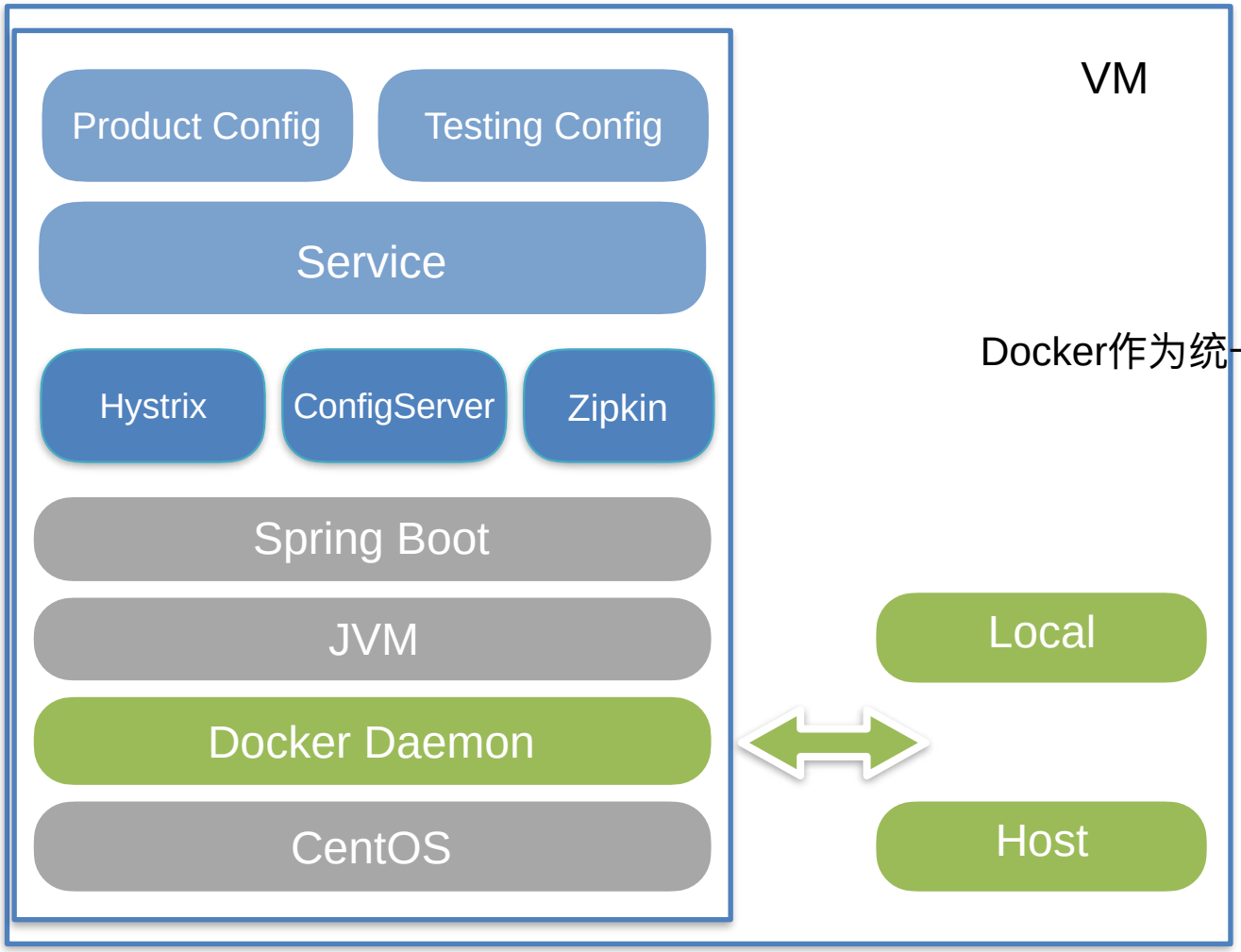

持续集成
监控报警
镜像存储
服务发现
日志管理
Java/PHP
Nginx
Hadoop/Spark
Jenkins
上线发布
回滚降级
弹性扩容
资源调配
灰度测试
Swarm
Mesos
ZK/Consul
Kafka
HAProxy
ES
Prometheus
Registry
腾讯云

必备的组件

使用容器之后，基于VM上一些流程需要重新改造，才能提供基本的服务。我们的系统现有的基本组件如下：

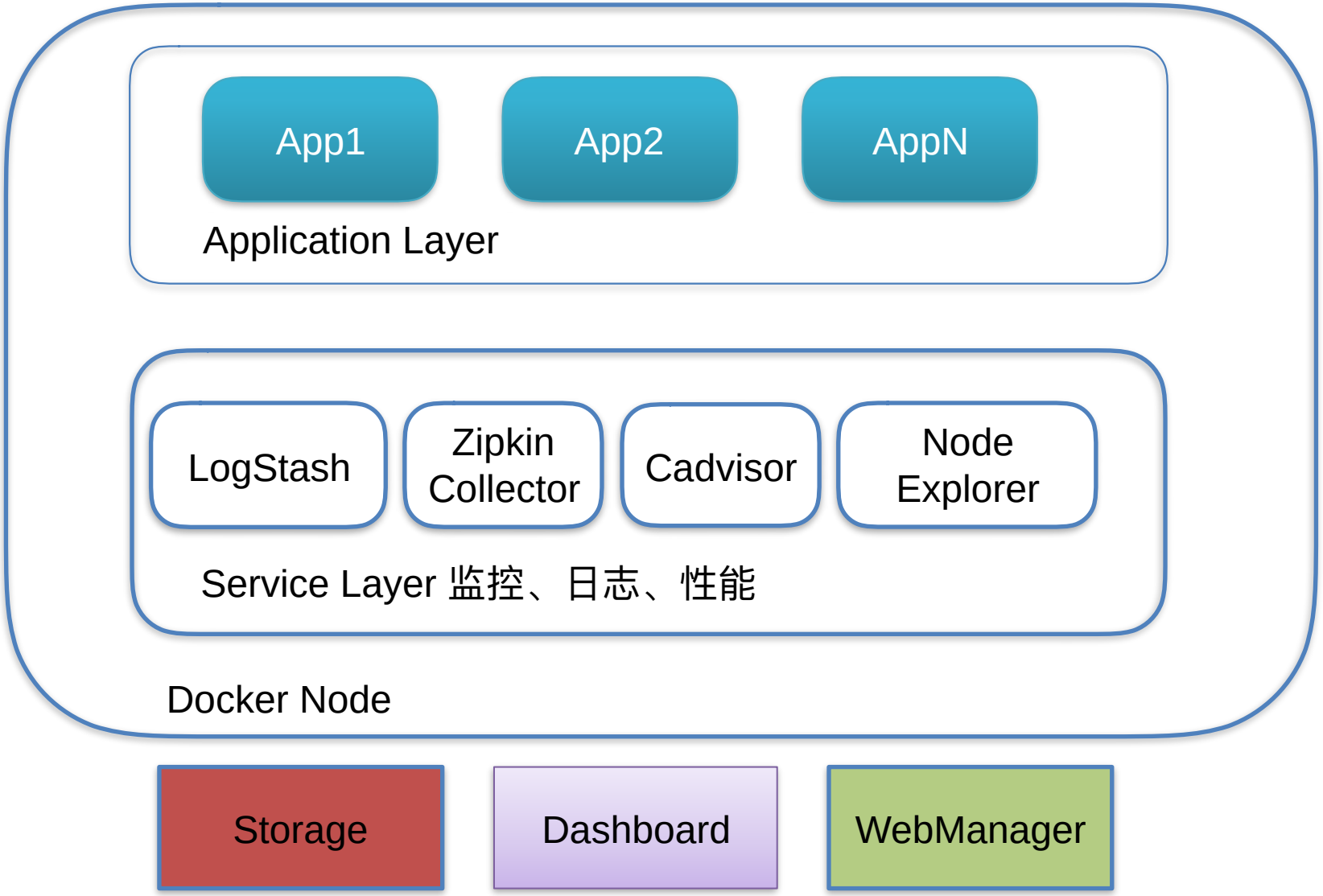
- 容器化->包管理->CI/CD
- 配置管理
- 服务发现
- 日志收集
- 监控，报警
- 升级回滚策略
- 服务域名解析
- 负载均衡

单机模式



Docker作为统一的打包部署方案

Docker节点

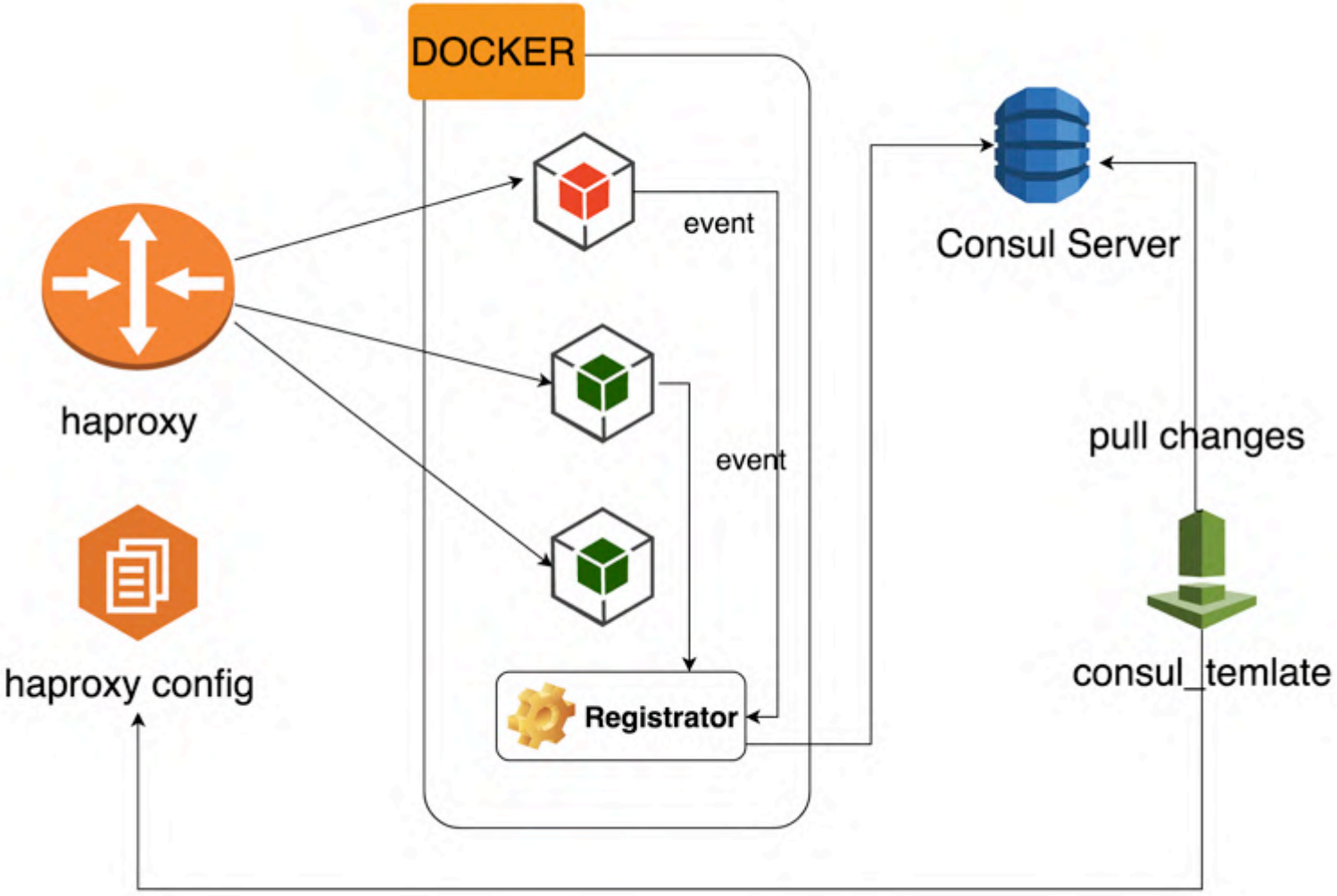

App1
App2
AppN
Application Layer
LogStash
Zipkin Collector
Cadvisor
Node Explorer
Service Layer 监控、日志、性能
Docker Node
Storage
Dashboard
WebManager

资源利用率

	VM	Mesos已分配	Mesos实际使用
CPU	< 5%	70%	30%
Mem	< 30%	80%	70%

- 之前的很多非核心服务，为避免单点，至少部署两个节点，但导致利用率很低
- 实际使用和分配的比例之间还有空间。Mesos提供的over-subscription可以进一步提高利用率

服务发现


DOCKER
event
event
haproxy
haproxy config
Registrator
Consul Server
pull changes
consul_temlate

日志收集策略

1. 异常日志
   - 高优先级，第一时间处理
2. 结构化的Info/Error日志
   - 低延时，统计报警，实时分析
3. 非结构化的Debug日志
   - 调试，备查，审计
   - 轮转和集中存放

Email logger

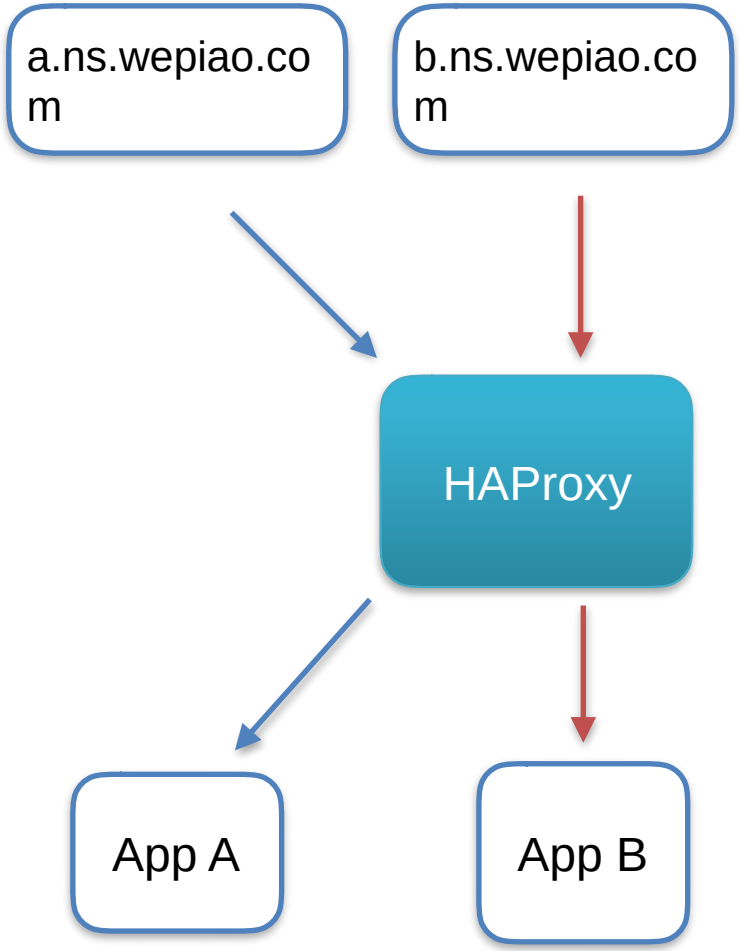
Logstash -> ES

logstash -> archive

容器中的动态服务，不再提供ssh登陆的方式来查看日志

泛域名

- 泛域名解析到 *.ns.wepiao.com
- 为测试环境，持续集成提供无限制的动态域名服务。不同工程和分支都有特定的域名对应
- 容器启动/停止事件会自动触发修改解析规则



微服务

- 拆大为小；各服务之间耦合小。每个服务的规模小到可控制的粒度，例如花几周可以重写某个服务
- 一定要有超时，熔断，隔离：hystrix
- 追踪请求依赖和耗时：requestId/zipkin

容器 vs 微服务 vs DevOps

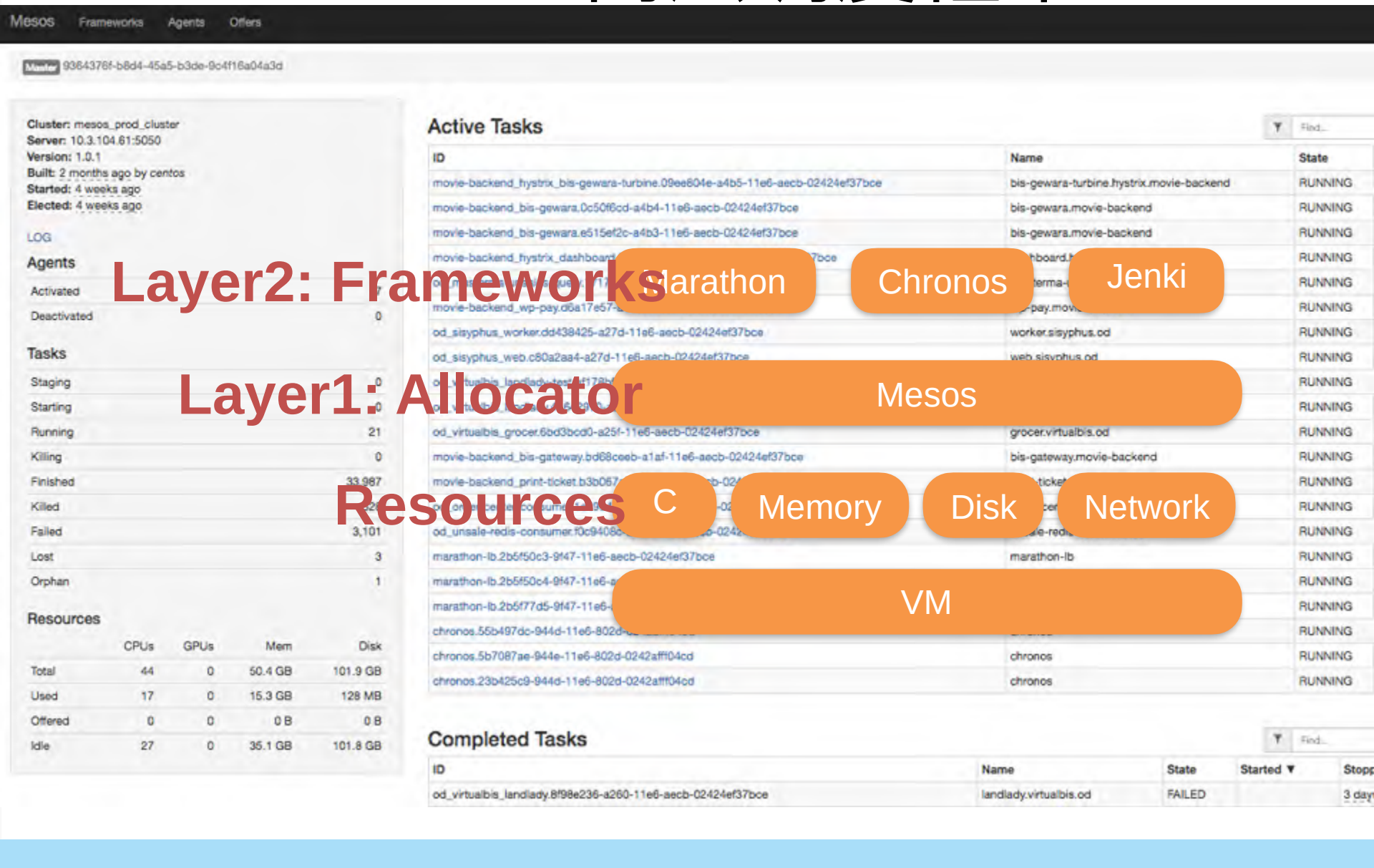
- 容器本身只是一种技术，为上层建筑提供便利
- 微服务是软件架构上的新潮流
- 而DevOps更多关注的软件工程中的人和流程的变化

一致性

- 核心业务：强一致mysql
- 服务之间有关联的数据：事后检查，业务监控，再恢复修补
- 幂等操作，调用者重试不会产生副作用

2.Mesos相关技术使用

Mesos: 两层调度框架



Layer2: Frameworks

Marathon Chronos Jenki

Layer1: Allocator

Mesos

Resources

C Memory Disk Network

VM

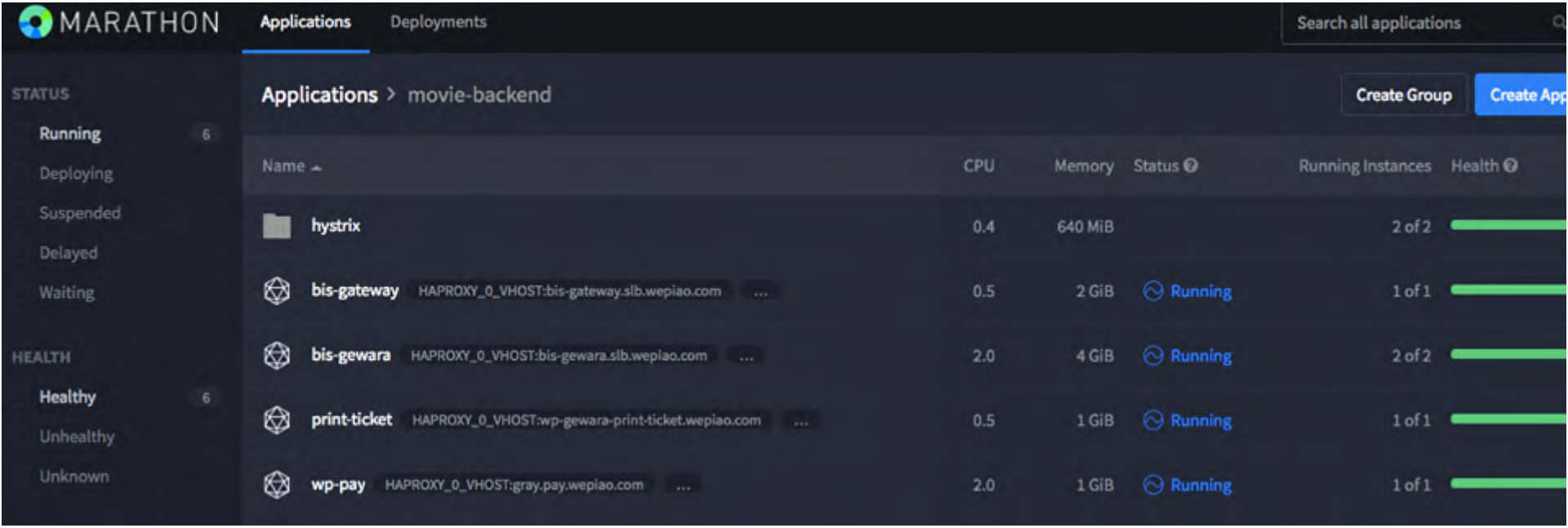
Marathon

优点

- 运行long-running services
- 功能强大：蓝绿部署，健康检查，动态扩缩容，分组管理，服务发现，滚动升级等

不足

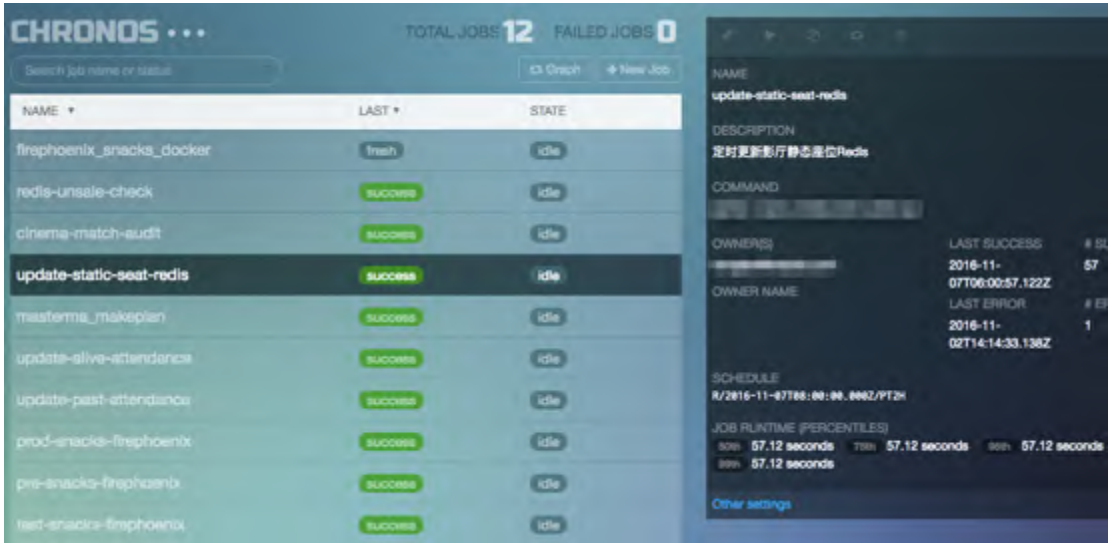
- 业务隔离，分组的功能欠缺



Chronos vs. Cron

优点

- 分布式环境工作，定时任务不再有单点问题
- 简单易用，稳定
- API功能完整



不足

- 界面太简单，甚至有bug
- 对容器环境的任务只能通过API操作
- 控制时间的方式单一
- 缺少分组管理和权限管理
- 自己包装前台页面，来调用API完整定制化的需求

Jenkins on Mesos

利用mesos集群的资源，而不再需要单独维护jenkins集群

1. 使用Marathon来维护Jenkins Master的稳定
2. Jenkins作为Mesos上的一个Framework来获得集群的资源
3. 最后通过Docker container把任务运行在Mesos Slave上

遇到的一些的问题

- 容器社区“百花齐放”有好处也有坏处
  - 三分天下，mesos/k8s/swarm，都需要投入精力
  - 各种框架，方案大同小异，选型，评估，试错的成本都不低
- 核心服务应用较少，被验证的程度尚浅
- 同时维护两套流程：VM vs. 容器
- 疑难杂症（重启，重装，升级，有不少问题我们找不到问题）
- 需要开发可视化的管理界面，而不是命令行
- 多租户：不同生产服务之间互不影响。mesos 的role/weight/reservation等概念太底层。需要更高层的管理工具
- 分离的集群：kafka, es, hadoop, spark等。每个集群都需要单独维护。有状态的服务合并到集群还不是很成熟
- HAProxy默认配置转发部分中文url出400错误。需要开启兼容一些上太不合法的url
- 蓝绿部署的过程中，CDN回源，导致js/css和html指向了两个不兼容的版本
- 容器分配的内存不足，会被突然杀死

SDCC 2016

中国软件开发者大会

SOFTWARE DEVELOPER CONFERENCE CHINA

谢 谢！