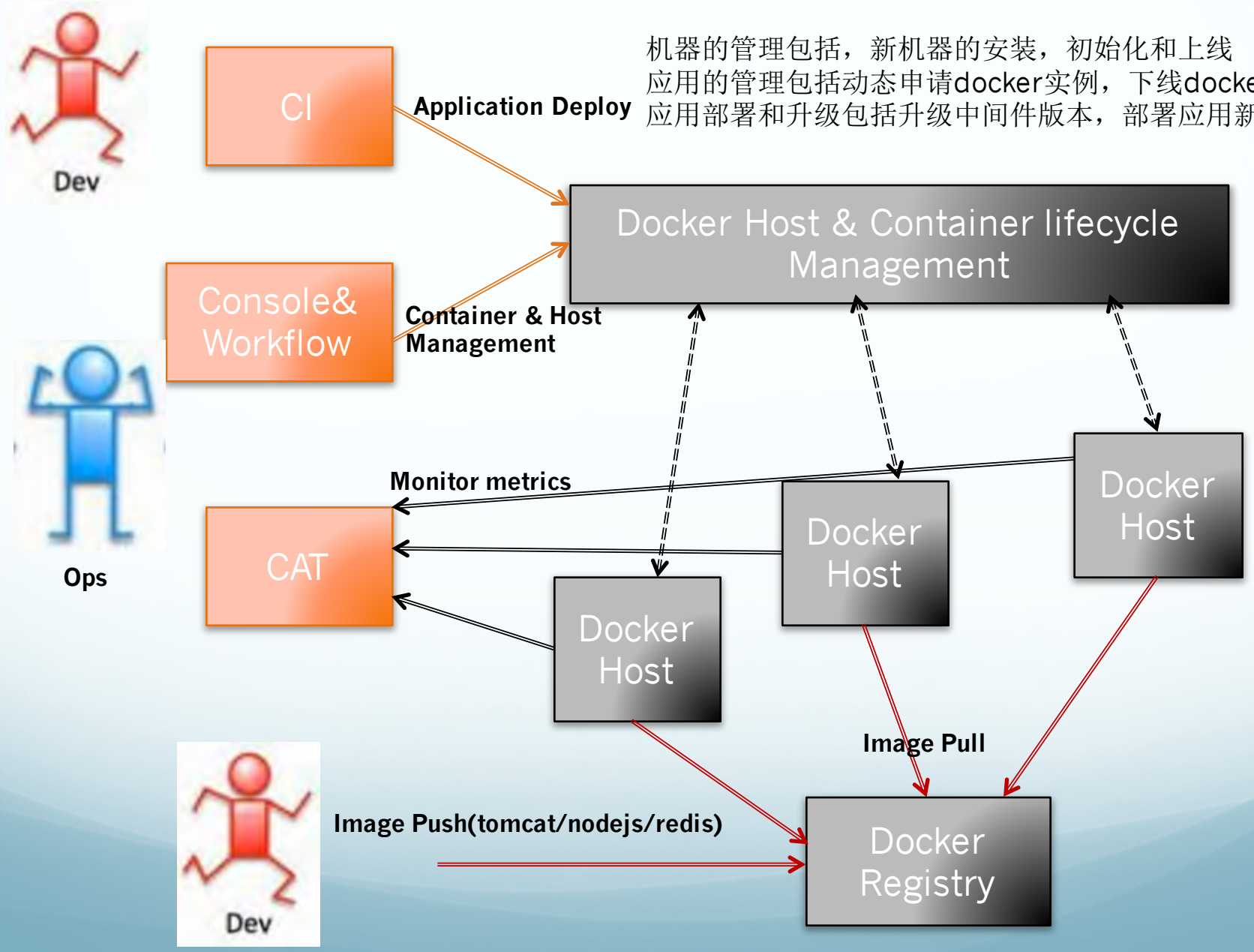
Docker在点评私有云平台的应用

大众点评网　盛延敏

主要内容

- 用户场景和设计理念
- 点评的PAAS平台
- 碰到的问题和解决方案
- 总结与展望

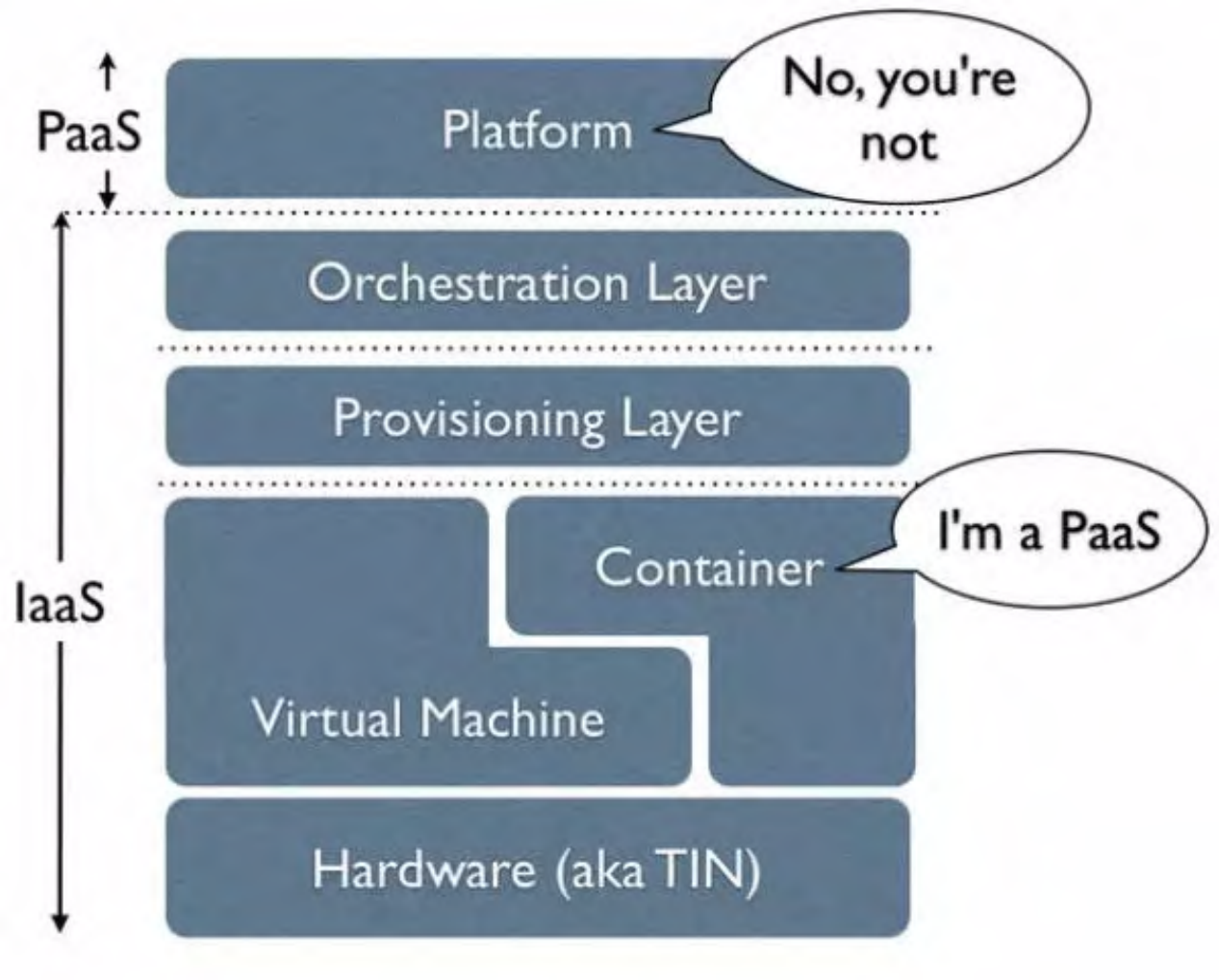
用户场景


Dev
CI
Application Deploy
机器的管理包括，新机器的安装，初始化和上线
应用的管理包括动态申请docker实例，下线docker实例等
应用部署和升级包括升级中间件版本，部署应用新版本等
Docker Host & Container lifecycle Management
Console& Workflow
Container & Host Management
Ops
Monitor metrics
CAT
Docker Host
Docker Host
Docker Host
Image Pull
Image Push(tomcat/nodejs/redis)
Dev
Docker Registry

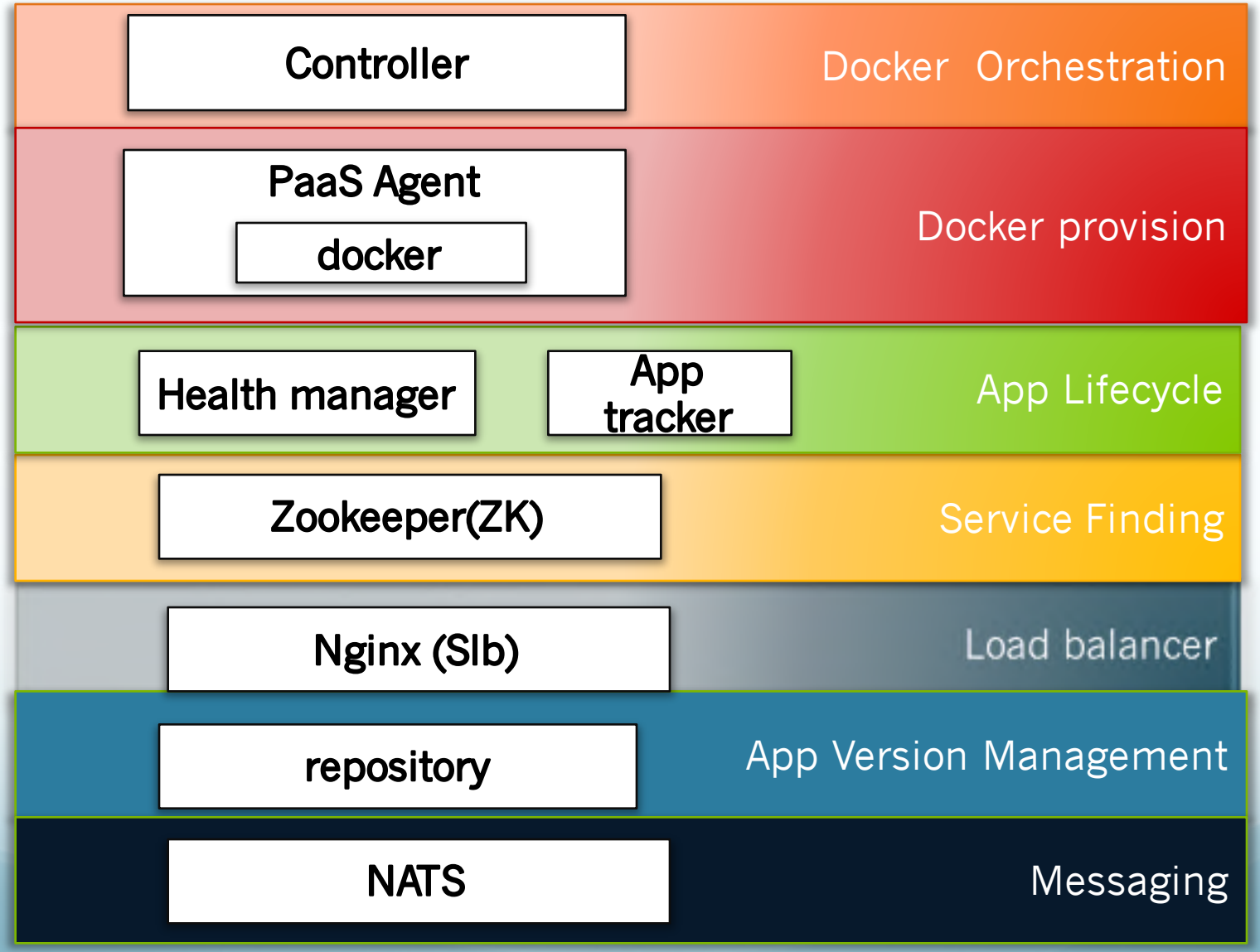
几个设计理念

- Docker container实例一旦创建，需要运行很长时间，表现出一个虚拟机的特性。(can we not put all into docker image?)
- Docker container和KVM一样，拥有一个唯一的内部IP.
- 根据内部VLAN规划，IPAM需要自己实现
- Docker container可以被ssh登陆访问，账号分为管理和只读等。

PAAS vs DOCKER



主要组件


Controller
Docker Orchestration
PaaS Agent
docker
Docker provision
Health manager
App tracker
App Lifecycle
Zookeeper(ZK)
Service Finding
Nginx (Slb)
Load balancer
repository
App Version Management
NATS
Messaging

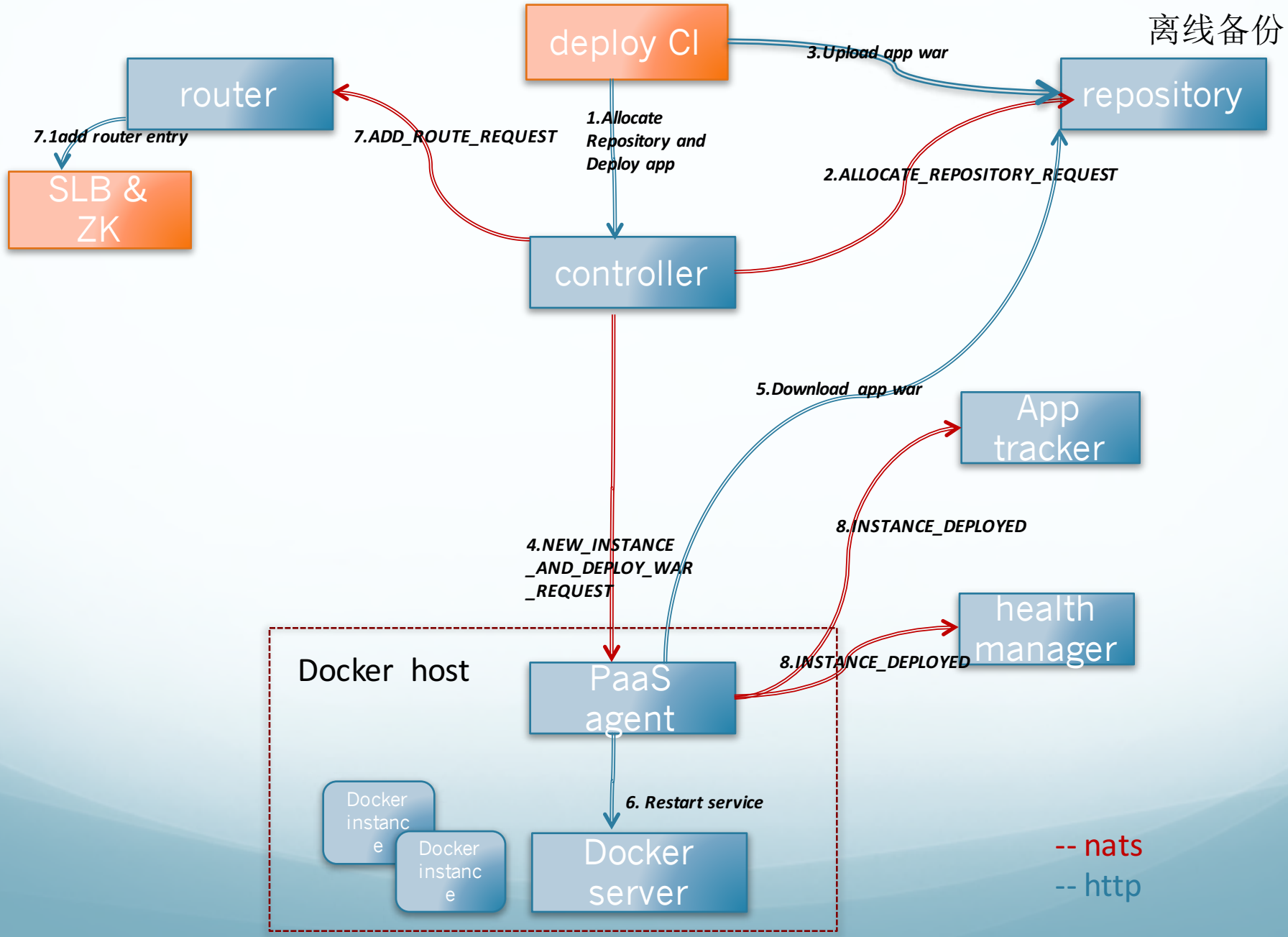
Controller

- 调度请求落到集群中
- 根据物理机器的资源动态状况， 请求的CPU/内存等大小进行动态调度
- 考虑物理机房的AZ概念，将应用在物理上尽量离散
- 可以进行按tag的混布或者不混布

Agent

- 接收controller的指令，创建和回收docker instance。
- 为docker rootfs准备mount目录，应用对应的程序包和中间件包通过mount挂载，方便升级和维护 (Can we put them into dockerfile?)
- 对管理的docker container做健康检查和资源上报。
- 接收中间件管控中心调用，完成中间件包的升级和应用重启。

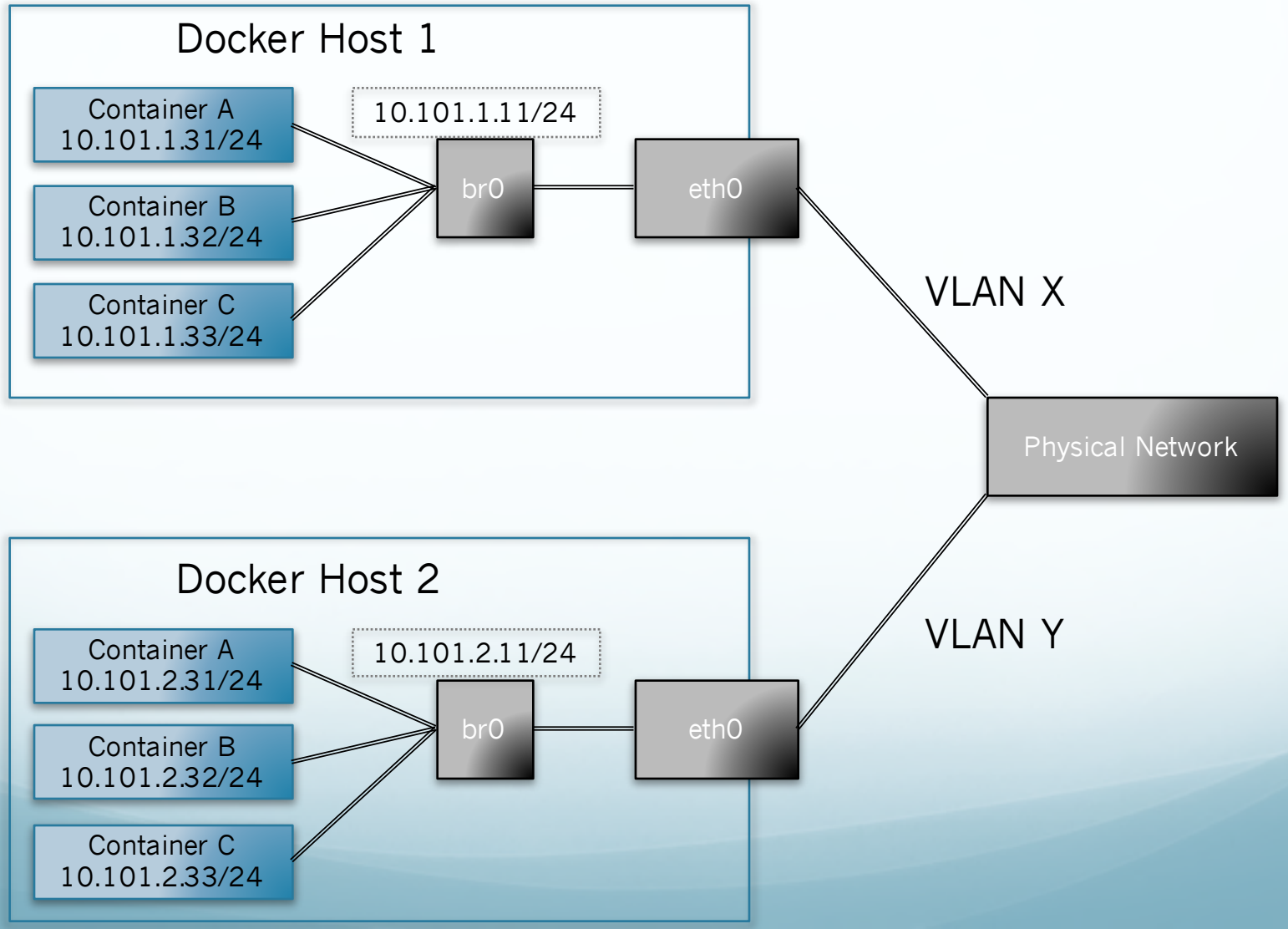
应用版本升级


离线备份
deploy CI
3.Upload app war
repository
router
7.1add router entry
SLB & ZK
7.ADD_ROUTE_REQUEST
1.Allocate Repository and Deploy app
2.ALLOCATE_REPOSITORY_REQUEST
controller
5.Download app war
App tracker
8.INSTANCE_DEPLOYED
4.NEW_INSTANCE _AND_DEPLOY_WAR _REQUEST
health manager
8.INSTANCE_DEPLOYED
Docker host
PaaS agent
6. Restart service
Docker instanc e
Docker instanc e
Docker server
-- nats
-- http

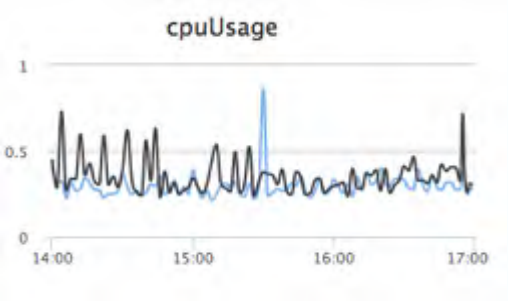
针对docker的定制

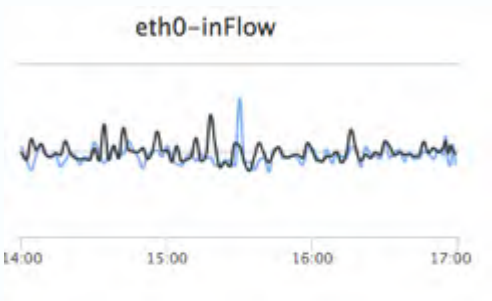
- 网络使用bridge的方式暴露内网VLAN的IP
- IP使用自己实现的管理和分配模块
- 增强监控功能，包括垃圾文件的自动清理

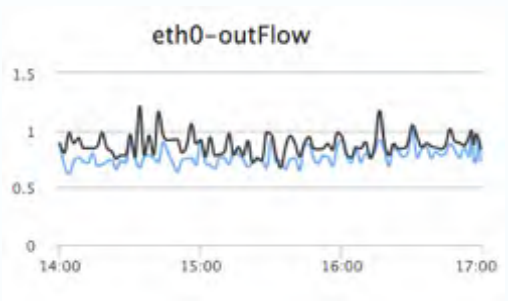
基于VLAN的网络定制

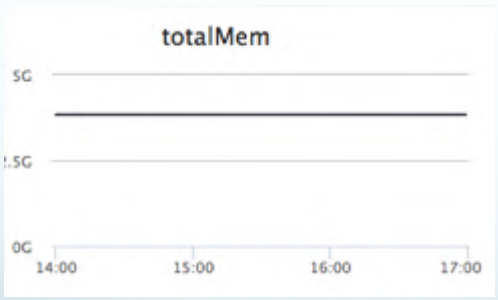


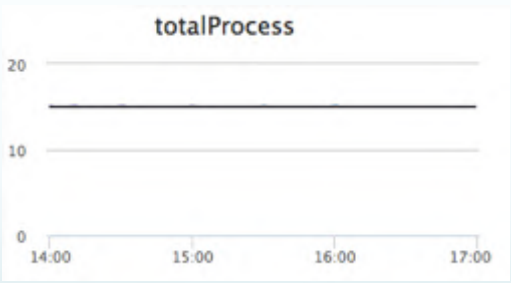
监控数据收集到CAT(Central Application Tracking)

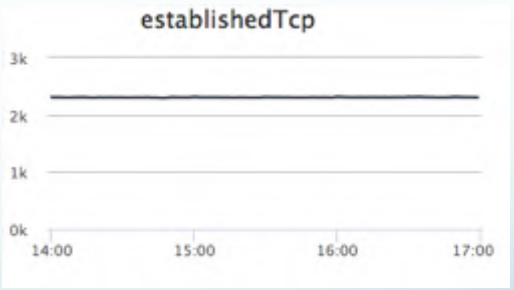












Dockerfile的维护

- 问题：多种Runtime谁负责来写Dockerfile?
- 我们的经验是
  - 建立基础镜像
  - 各种Runtime在基础镜像上增加自己独有的东西
  - 由开发和PaaS运维一起来做
  - 定期更新镜像，修改bug

物理机批量上线

- 问题：初始化和配置是一件非常耗时的事
- 我们的经验是
  - 使用Ansible完成批量安装和配置工作
  - Controller暴露API完成物理机的入库操作
  - 集成到界面，可以随时查看升级和安装的log等

Problem

- Docker VM 的磁盘I/O较慢
- centOS 使用了devicemapper作为存储

Solution

- Device mapper 使用裸设备，而不是loop device上。

Problem

- 在Docker VM内使用top/vmstat/free 等cpu/memory的资源监测有问题
- Docker VM和 Docker Host共享 cpu/memory等资源，不能有效的分辨出Docker VM的资源

Solution

- 给内核打补丁, 使得在Docker VM内部执行的操作只统计当前Docker VM所在的 cpuset/memory等。

- 小插曲: nodejs npm crash问题

Problem

- Docker Host物理机随机崩溃
- 打开Vmcore-dmesg可以看到是dm_thin针对device mappder的操作引起了crash

Solution

- 规避该问题的方法，设置blk discard为false。

docker –d --storage-driver=devicemapper **--storage-opt dm.mountopt=nodiscard --storage-opt dm.blkdiscard=false**

Problem

- Puppet 不能更新Docker VM里的配置。

- Docker VM的rootfs主要来自
  - Layed image produced by Dockerfile
  - Volumes produced in host machine by PaaS agent (dynamically generated at create time)

- Puppet更新的原理是先删除旧的再创建一个新的。而Docker VM里的挂载卷是不能删除文件的(文件在mounted dir下面是可以删除的)

Solution

- 将不可以删除的挂载卷文件复制到可写得rootfs层。

总结与展望

- 和点评devops结合，提供端到端的用户体验
- 抽象和标准化应用的运行时环境
- 提供类似KVM的单虚拟机用户体验
- 高密度部署
- 快速弹性伸缩

开源

- https://github.com/dianping/Dolphin

THANKS