

# 蘑菇街基于Docker的 私有云实践

@郭嘉

[guojia@mogujie.com](mailto:guojia@mogujie.com)



# 关于我

- 花名：郭嘉 – 张振华
- 05年浙大毕业
- 14年加入蘑菇街
- 虚拟化团队负责人
- 热爱新技术, 开源。。。。



## 关于蘑菇街 ABOUT US

中国最大的女性时尚社交电商平台。

成立于2011年，总部位于浙江杭州，目前拥有1.3亿注册用户，日活跃用户超过800万，2014年全年实际交易额超过36亿元，团队总人数超过800人。无论在用户规模上，还是交易额上，都已经成长为中国最大的女性时尚社交电商平台。

自公司成立以来，蘑菇街一直坚持社交与电商相结合的发展方向，致力于开创全新的社交电商商业模式，面向新一代年轻时尚人群提供优质的社交和购物体验。蘑菇街的核心用户群体是18-26岁之间年轻时尚的都市女性，他们崇尚自由独立，个性解放，拥有独到的审美品位与时尚主张，以及巨大的消费潜力。

天生爱占有  
一生是杀手  
喜欢就要独占，  
我好货独揽



# 我们为什么想做私有云

- 越来越多的机器，集群管理，基础平台的建设
- 提高资源的利用率
- 服务化，平台化，可视化
- 提升发布和部署的效率
- 实现业务的弹性，水平扩展



我们想到了OpenStack



我们还想用Docker



# Docker@蘑菇街

- 2014年圣诞节期间上线，OpenStack IceHouse + Docker 1.3.2。
- Machine Container 或“胖容器”。
- 三个集群，经历过4次大促，包括双11，线上运行稳定。
- 搭建有内部多个的镜像仓库Docker Registry。
- Docker支持OpenvSwitch VLAN和Linux Bridge两种网络模式。
- 每个集群可以同时管理KVM，Docker。
- 自研了基于OpenStack的轻量级PaaS平台。
- 自研了虚拟化交付系统。
- 自研了虚拟化管理控制台。

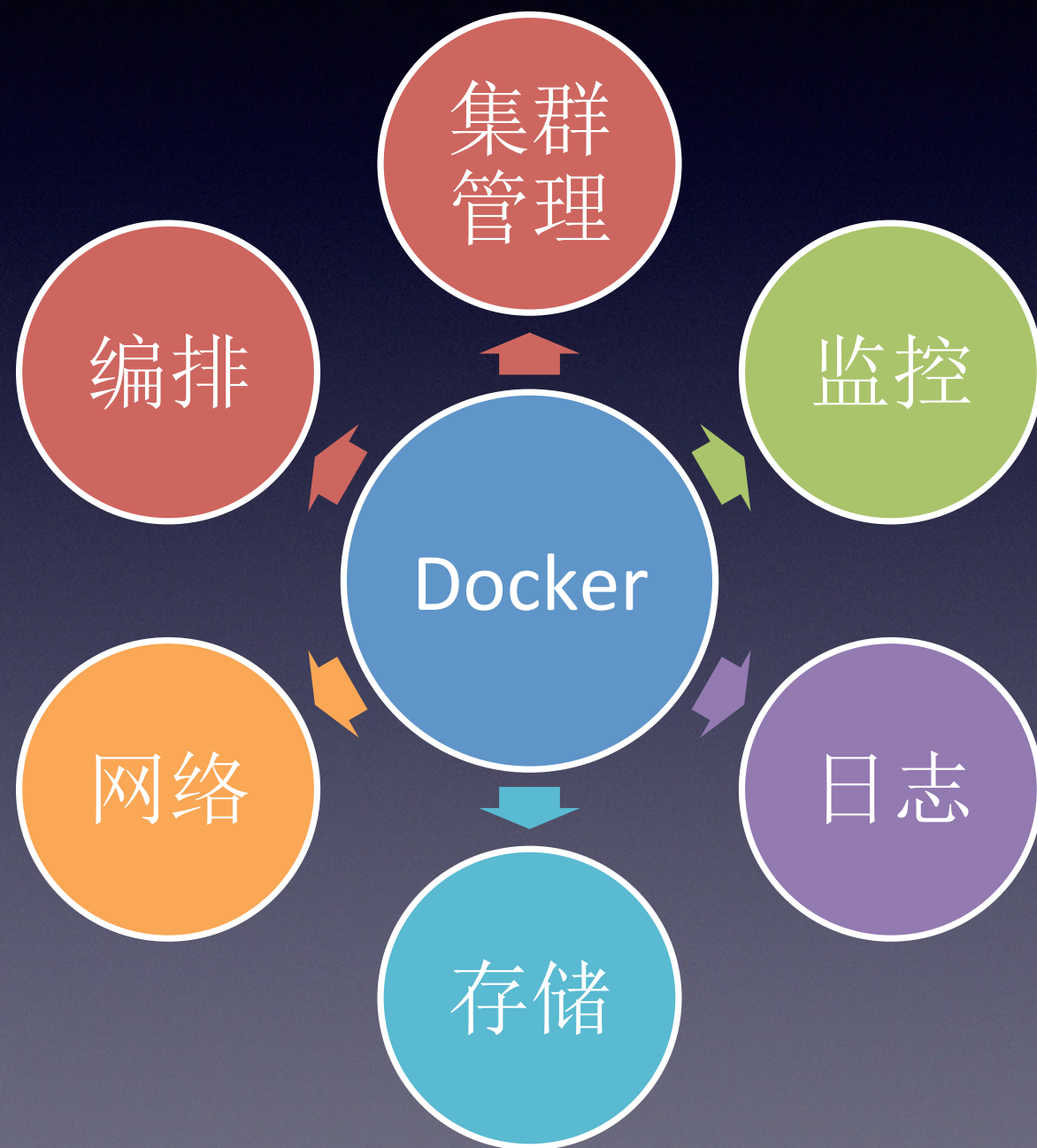


# Docker的优势

- 轻量，秒级的快速启动速度
- 简单，易用，活跃的社区
- 标准统一的打包/部署/运行方案
- 镜像支持增量分发，易于部署
- 易于构建，良好的REST API，也很适合自动化测试和持续集成
- 性能，尤其是内存和IO的开销



# 只有Docker是不够的



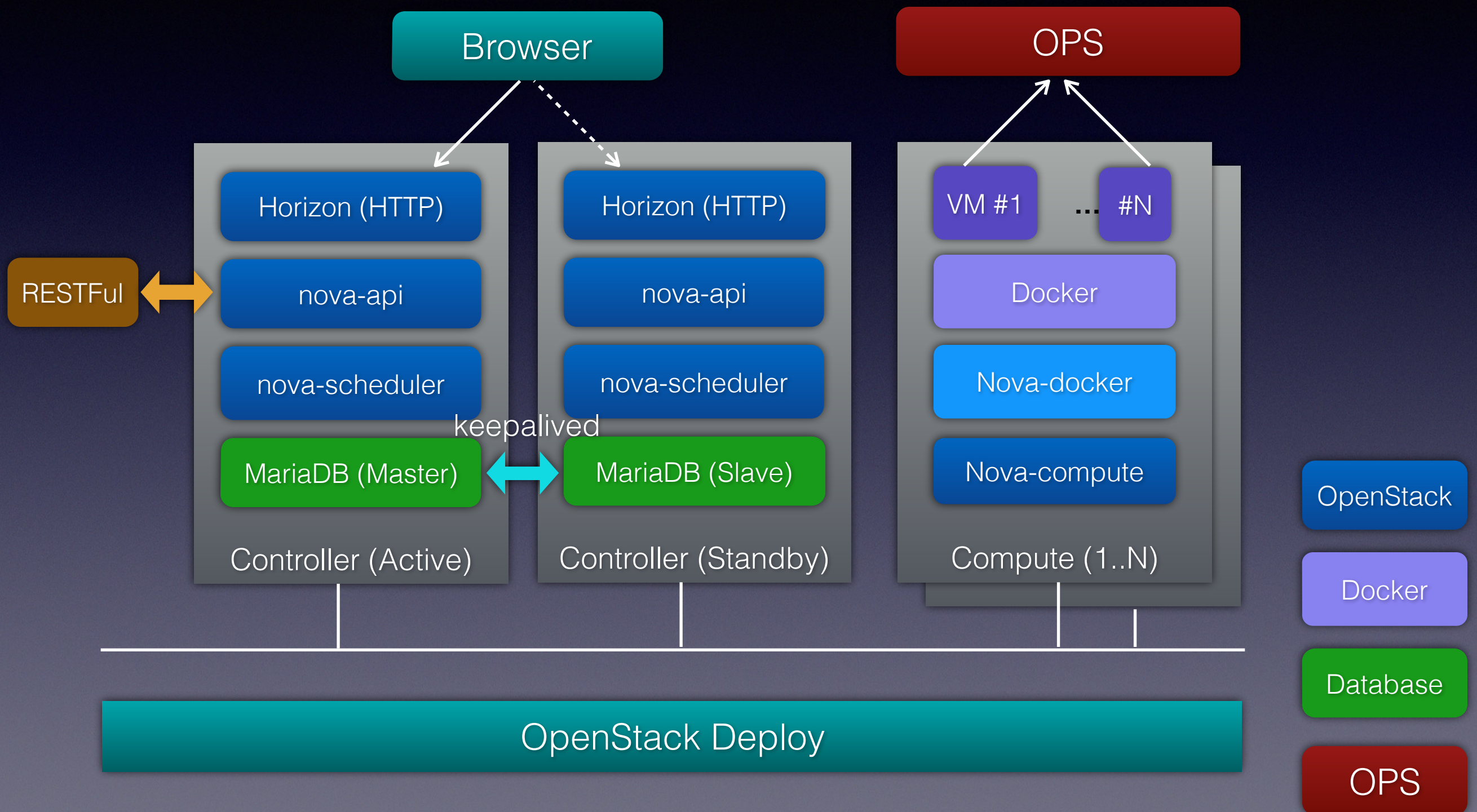


# Docker@蘑菇街





# 逻辑架构图

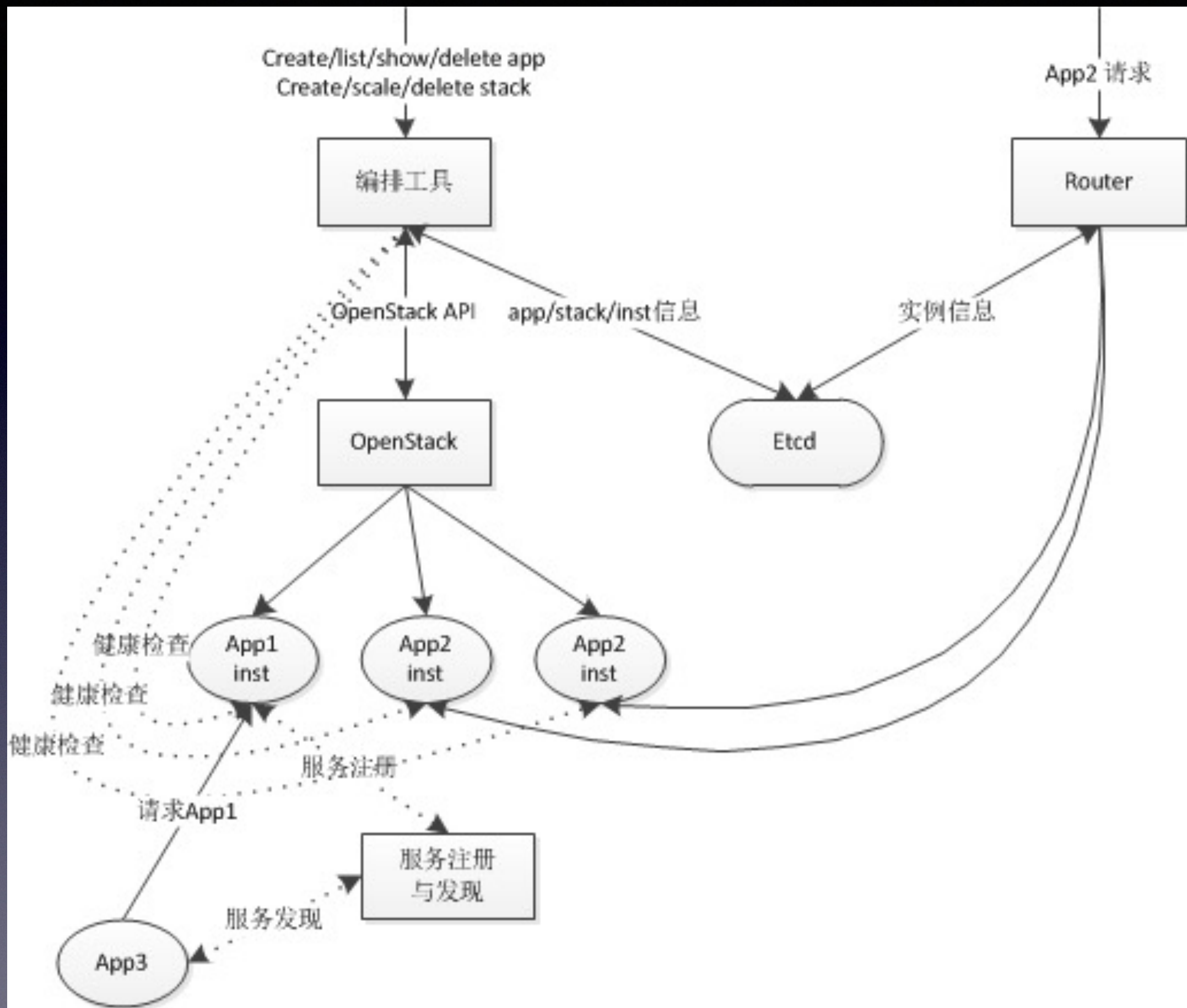




# PaaS

- 目标：
  - 快速的系统构建
  - 业务的平滑部署升级
  - 自动的运维管理
- 概念：
  - App：一个应用包含一个或多个Stack
  - Stack：相同的Docker实例，一个Stack中的不同实例尽量部署在不同的物理机上
- 支持基于容器的持续集成： Jenkins + Docker，从编译到构建全自动化







# 监控

- 和已有监控系统的深度集成。
- 实时监控和阈值报警：节点存活性/语义监控，关键进程，内核日志，实时pid数量，网络连接跟踪数，容器oom报警。
- 阈值报警：短信报警，IM报警等多种形式。
- 健康检查：部署环境／配置的一致性检查。
- container-tools：
  - 容器内实现load值计算，支持load和qps限流。
  - 替换了uptime, top, free, df，类似docker stats。



# 容灾

- 离线恢复docker容器中的数据的能力。
- Docker实例跨物理机的冷迁移: docker commit, docker push。
- 动态的CPU内存扩容: cgroup。
- 网络IO / 磁盘IO的限速: cgroup/tc。



# 网络

- NAT – 20% performance lost
- Host mode? No network isolation
- Linux bridge **without** iptables
- OVS VLAN **without** iptables
- other\_args="—**bridge=none**"



# 体会和思考

- 相比KVM，容器技术还有不完善的地方。
- 容器下的运维手段和运维经验的冲击。



# Docker目前的局限

- 系统／内核层面的隔离性
- 缺乏成熟的集群管理（K8S/Swarm/Mesos）
- 业务无感知的升级，Docker daemon live upgrade



# 未来的畅想

- App Container + PaaS
- Kubernetes/Swarm/Mesos + Docker
- 更高效更便捷的运维
- 弹性的资源交付
- 统一的部署方式
- 热迁移
- 公有云