开源社
kaiyuanshe

2016中国开源年会

China Open Source Conference 2016

时间：2016年10月15日-16日

地点：北京航空航天大学

Docker开源社区初探

魏世江@希云cSphere

希云cSphere——企业级私有容器云平台 https://csphere.cn

个人简介

- 魏世江（https://github.com/mountkin）
- 2009~2013年在新浪SAE负责公有PaaS服务管理系统的设计及开发
- 2013年底联合创立云栈科技，推出企业级私有容器云平台产品cSphere
- Docker社区活跃开发者

大纲

- Docker开源社区简介
- Docker的代码合并流程
- Docker版本发布流程
- 如何参与Docker开源项目
- Docker未来展望

Docker开源社区简介

Docker开源社区简介

Docker项目的历史

社区运营情况

社区管理用到的工具

Docker项目历史

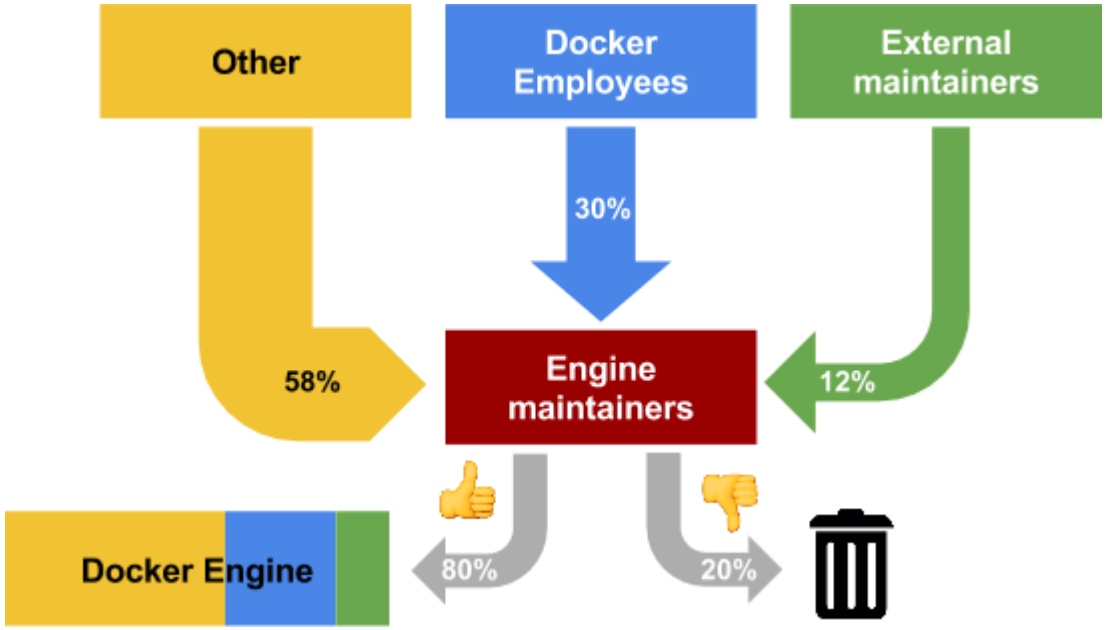
1982	unix引入chroot功能
2008	cgroups和名空间合并进Linux 2.6.24内核
2008	IBM开始开发LXC
2013年1月	Docker项目首次提交代码，基于LXC
2013年3月	Docker 0.1.0发布
2014年6月	Docker 1.0发布
2015年4月	cSphere首次向Docker项目提交代码
2015年6月	Linux基金会牵头成立OCI，发布容器运行时和镜像的业界标准

Docker项目历史（续）

2016年2月	Docker 1.10.0发布，抛弃了LXC
2016年4月	Docker 1.11.0发布，引入containerd, runc来管理容器生命周期，全面对接OCI标准
2016年7月	Docker 1.12.0发布，内置了编排引擎swamkit

社区运营情况

2000名左右代码贡献者（Docker所有的开源项目贡献者之和）



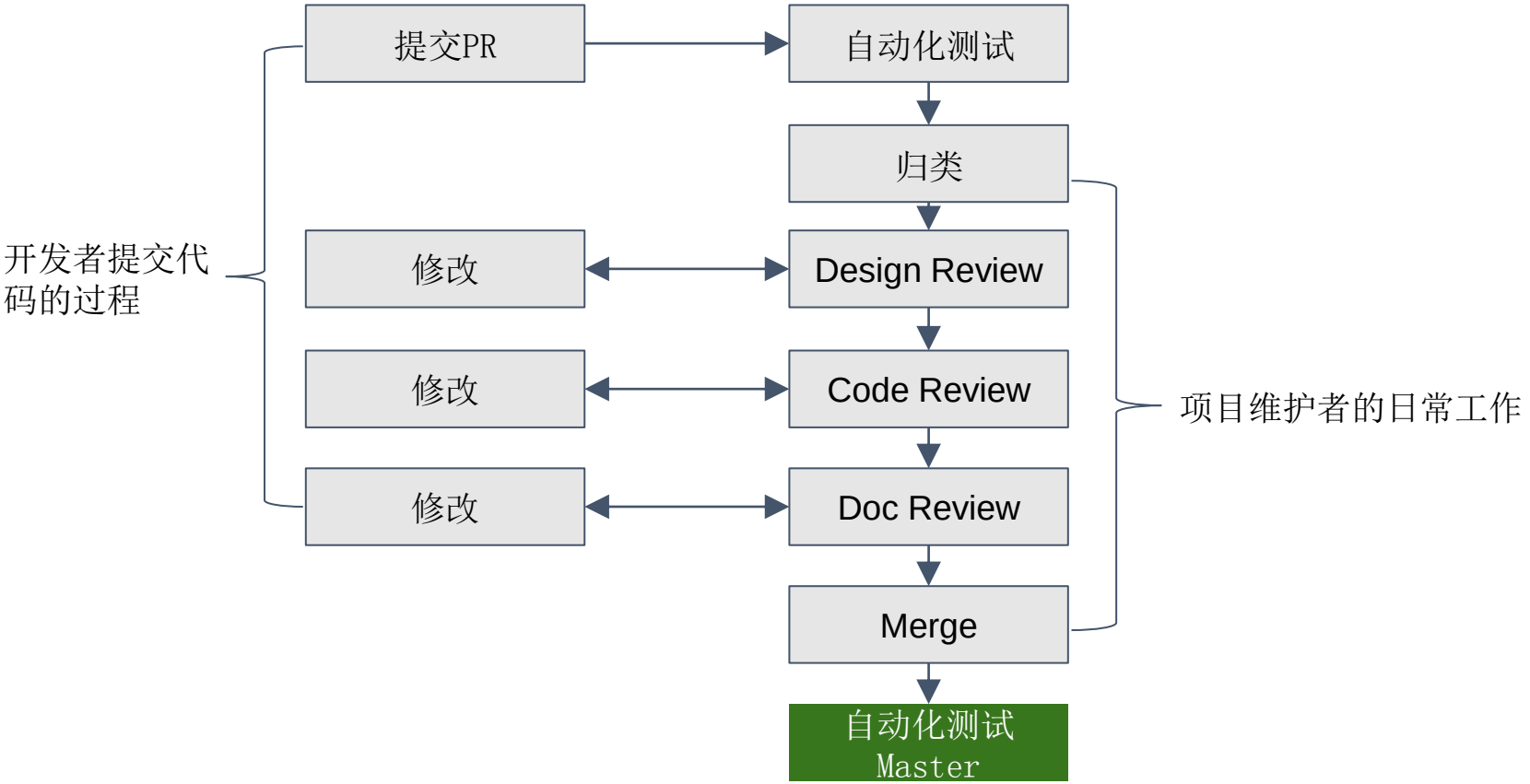
图片引自 https://blog.docker.com/2016/05/open-source-docker-part-1-people/

社区管理用到的工具

- github
- leeroy github机器人
- Jenkins
- Elasticsearch + Kibana + github API，用于分析项目参与者活跃情况

Docker的代码合并流程

Docker的代码合并流程



代码提交

- 原子提交，每次commit应该只变更尽可能单一的功能，方便日后审计或回滚
- 提交日志应该尽可能详尽
- 每个PR尽量保证只包含一个commit，便于代码审核
- 先本地执行相关测试，测试通过后再提交PR
- 及时解决冲突，git rebase & force push

PR归分类及过程跟进

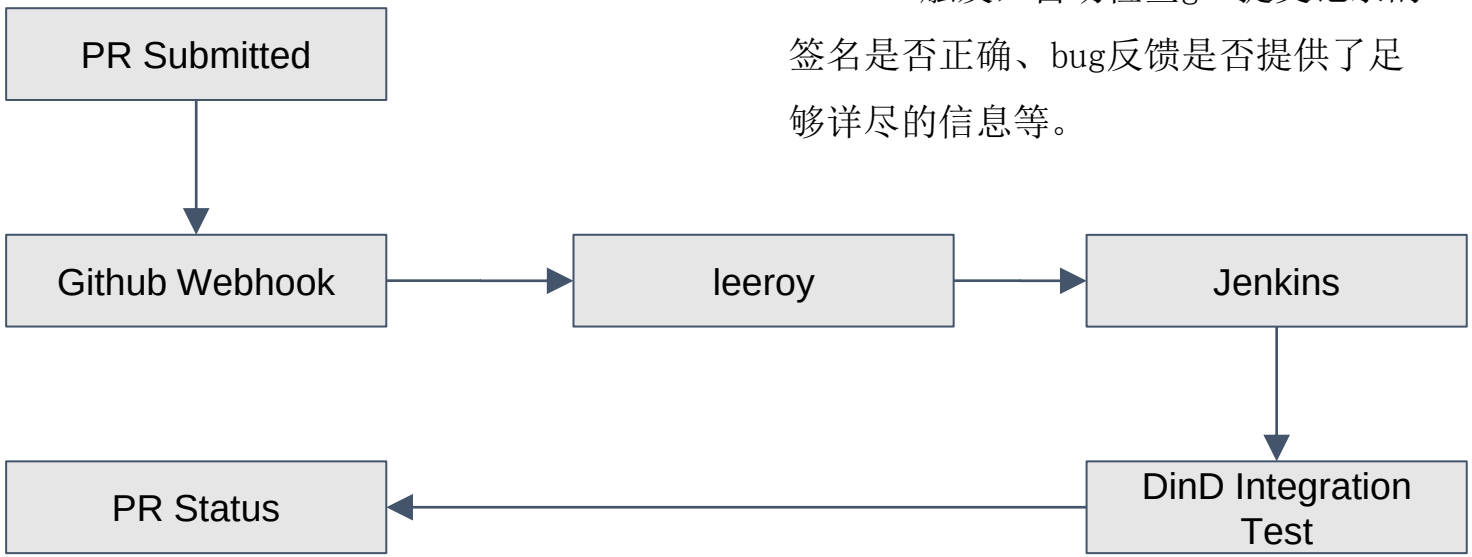
- leeroy自动分类
- 项目维护者人工分类
- 代码/文档审核过程的每个阶段完成后及时更新Label

Review

- 任何新特性或者breaking changes需要Design Review
- 任何代码变更需要Code Review
- 任何涉及到用户使用习惯的改变、新特性等需要Doc Review

CI

leeroy是Docker开发团队用来与github集成的一个机器人，通过github webhook触发，自动检查git提交记录的签名是否正确、bug反馈是否提供了足够详尽的信息等。



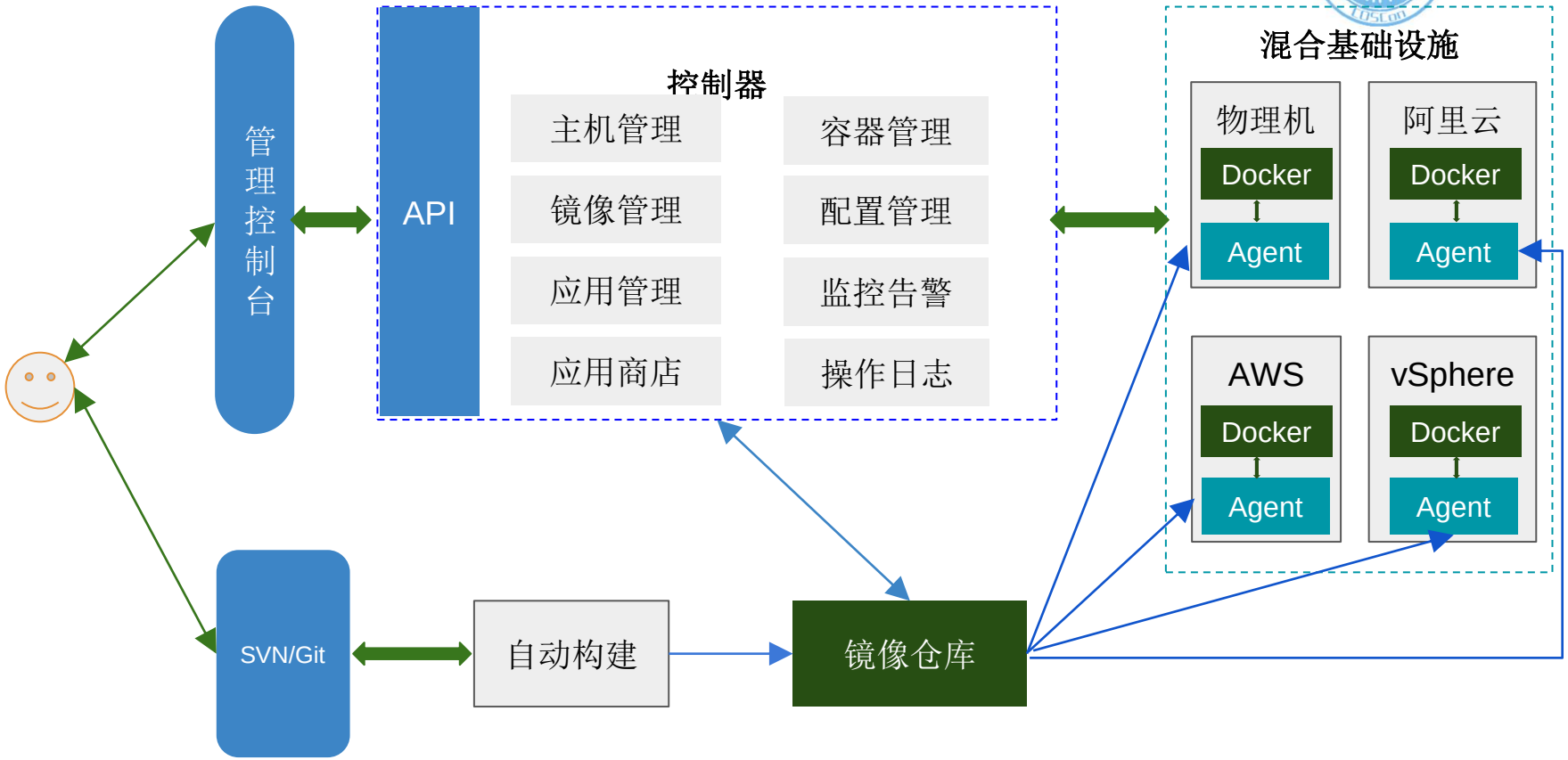
CI

所有测试用例运行于容器中，通过DinD在容器里启动若干Docker Daemon进行集成测试。

优点：

- 多次测试完全独立，不会相互影响
- 不污染宿主机环境
- 测试环境搭建简单，只要能运行Docker就能随时随地运行测试

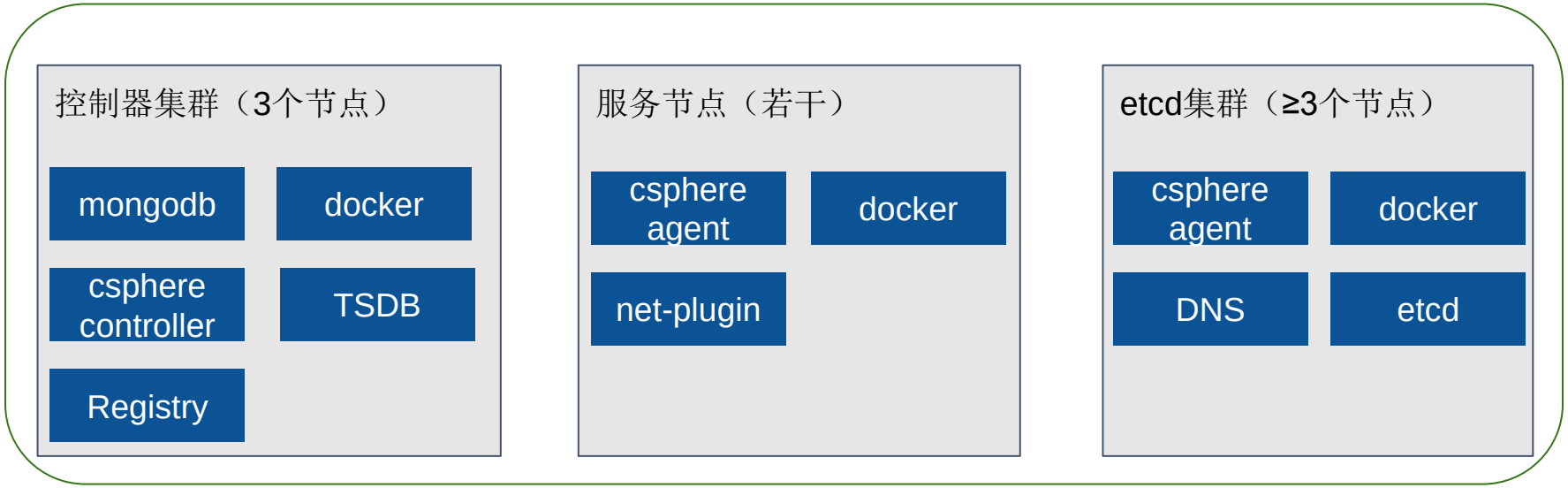
借鉴Docker的CI流程



cSphere功能模块示意图

借鉴Docker的CI流程

cSphere内部组件



cSphere HA模式部署一共涉及至少7台主机27个单元，通过DinD在笔记本上即可一键部署所有组件并执行集成测试

版本发布流程

版本发布流程

版本发布周期

2015年以前两个月左右发布一个大版本，2016年以来放慢了发布速度，大约3个月发布一个大版本。

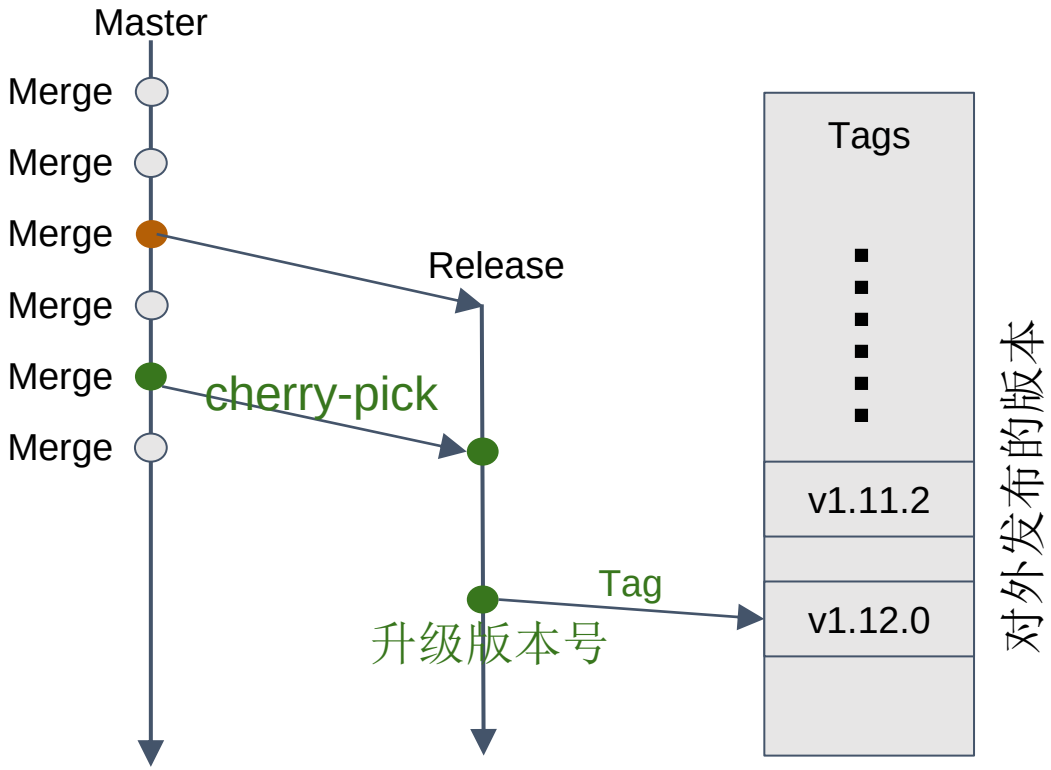
在两个大版本之间，往往会有一两个bugfix版发布。

版本命名规范

严格依据semver标准命名软件版本，参考http://semver.org/

基于git的版本发布流程

基于git的版本发布流程



基于git的版本发布流程

- 所有改动自动测试+Review完成后直接合并进Master
- Master分支达到可发布状态时（里程碑中的issue/PR完成到一定比例），创建Release分支
- 版本发布窗口，Master分支不用完全冻结，期间涉及到与待发布版本相关的改动仍然提交到Master，然后git cherry-pick到release分支
- 里程碑里的PR/Issue完成时，在Release分支升级版本号，创建Tag，更新Changelog，发布新版本
- 小版本发布仍然基于上一个版本的Release分支，从master分支cherry-pick相关变更

如何参与Docker开源项目

为什么要参与开源项目？

- 个人成就感
- 资源互补
- 认识牛人
- 学习成功开源项目的管理经验，应用于日常工作
- ……

参与Docker开源项目的一些渠道

- Github https://github.com/docker
- 邮件列表
  https://groups.google.com/forum/#!forum/docker-dev
- IRC irc.freenode.net #docker #docker-dev

如何开始

- 反馈BUG
- 协助其他用户解决问题
- 提出改进意见并参与讨论
- 改进文档
- 修复BUG
- 开发新特性
- ……

Docker的未来

Docker的未来

容器引擎的多样化

容器在企业落地遇到的问题

容器引擎的多样化

- Docker https://github.com/docker/docker
- rkt https://github.com/coreos/rkt
- cri-o https://github.com/kubernetes-incubator/cri-o
- *hyper https://github.com/hyperhq/runv

容器在企业落地遇到的问题

- 虚拟机-->容器管理/使用方式的变化
- 使用容器后应用代码发布方式的变化
- 现有应用向容器迁移的成本
- 容器的inmutable特性在某些场景下很不灵活，如：配置变更

cSphere推动容器落地的经验

- 对普通用户屏蔽容器技术，用户真正需要的是服务，而不是容器或者虚拟机本身
- 尊重用户的使用习惯
- 与用户熟悉的工具进行整合，如：jenkins,eclipse

Thanks & QA