本资料版权归 KEUC 所有，未经允许不得私自以任何形式传播，违反必究。

2017 China Kubernetes End User Conference
kubernetes 中国用户大会
KEUC 2017
2017.10.15 / 中国 · 杭州

质量测试部的容器化实践

基于Kubernetes的实践

中国移动杭州研发中心 质量测试部 魏少文

主办：CLOUD NATIVE COMPUTING FOUNDATION caicloud

本资料版权归 KEUC 所有，未经允许不得私自以任何形式传播，违反必究。

简介

遇到的问题

Kubernetes实践

展望

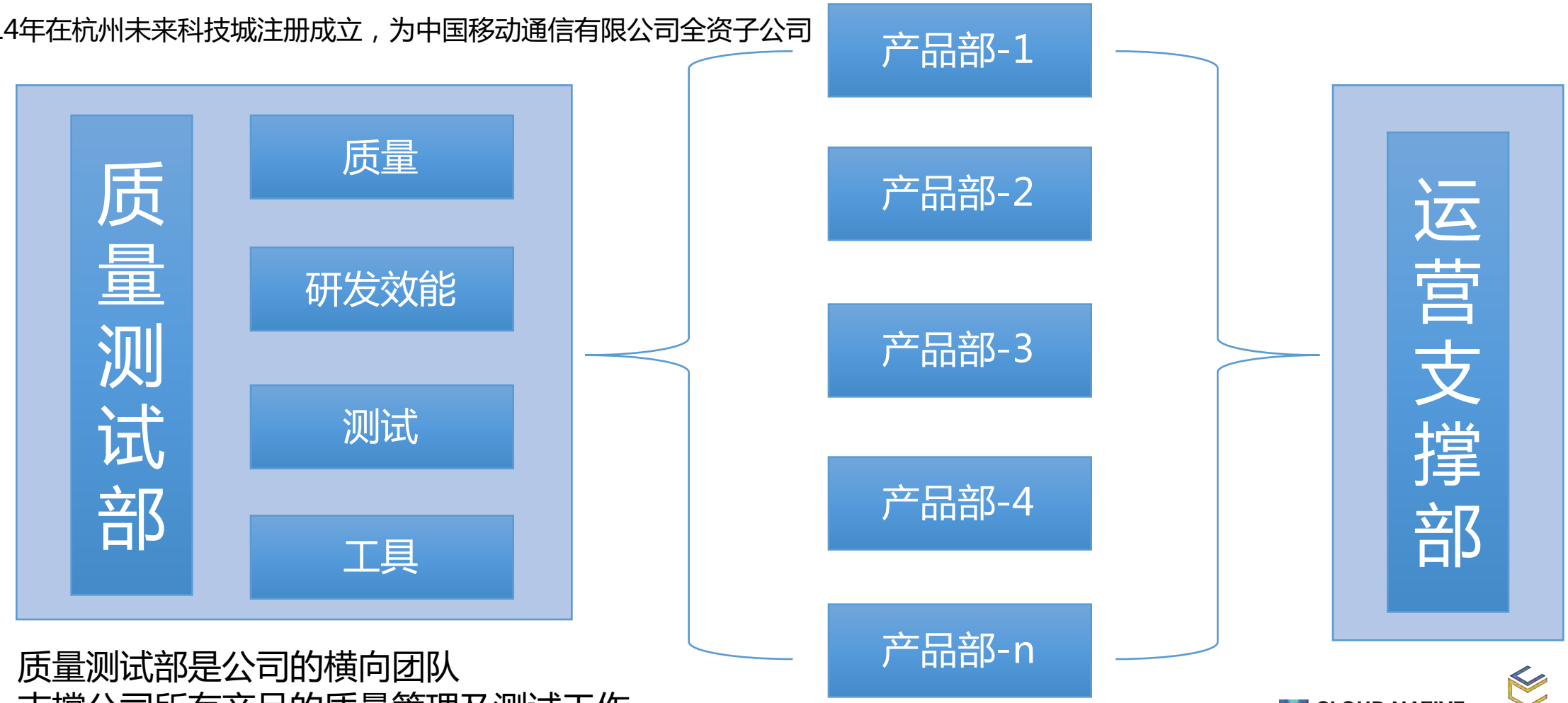
Q&A

本资料版权归 KEUC 所有，未经允许不得私自以任何形式传播，违反必究。

简介

中国移动杭州研发中心（中移（杭州）信息技术有限公司）

于2014年在杭州未来科技城注册成立，为中国移动通信有限公司全资子公司



质量测试部是公司的横向团队
支撑公司所有产品的质量管理及测试工作。

主办：

本资料版权归 KEUC 所有，未经允许不得私自以任何形式传播，违反必究。

简介

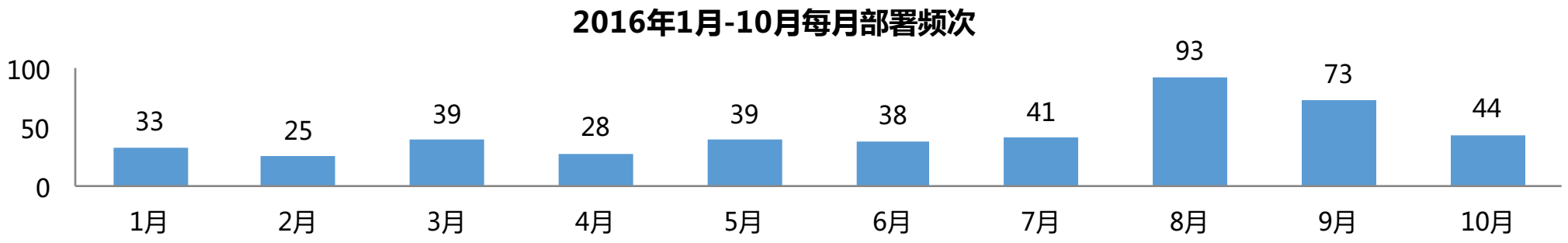
公司成立3年多以来，业务越来越多，越来越复杂
同时对交付的时间要求也越来越高，随之而来对研发模式敏捷化，持续交付越来越重要

业界互联网公司交付

公司	周期时间	部署频率	可靠性	客户响应
Amazon	分钟	23000次/天	高	高
Google	分钟	5500次/天	高	高
Netflix	分钟	500次/天	高	高

除此之外，Facebook、Twitter等国际互联网巨头，都有非常高的部署频率
数据来源：《The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win》

杭研交付



主办：

本资料版权归 KEUC 所有，未经允许不得私自以任何形式传播，违反必究。

简介

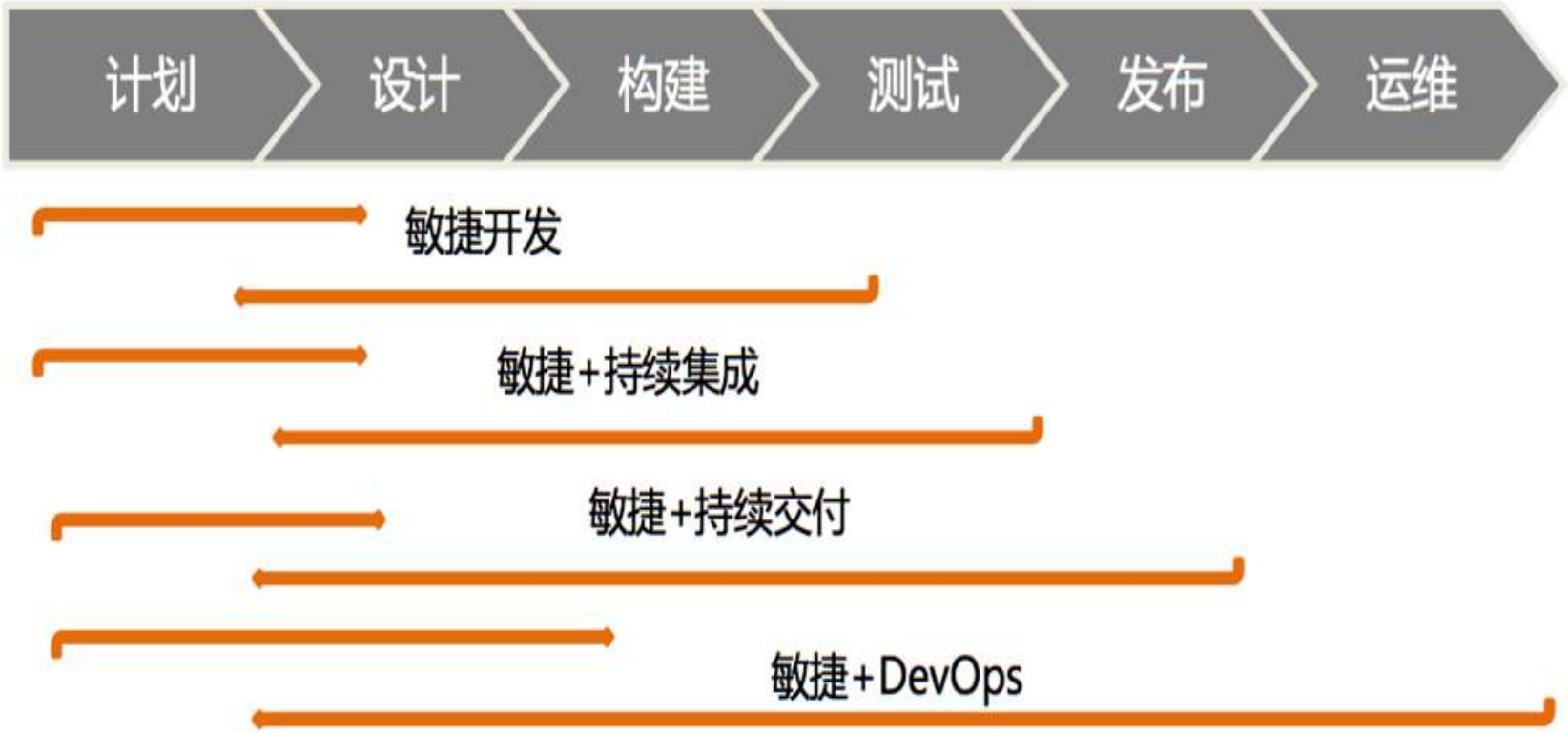
Devops:

Development和Operations的组合，是一组过程、方法与系统的统称，用于促进开发、技术运营和质量保障部门之间的沟通、协作与整合。

持续集成：
指软件个人研发的部分向软件整理部分交付，频繁进行集成以便更快地发现其中的错误

持续交付：
将集成后的代码部署到更贴近真实运行环境的类生产环境中。

持续部署：
持续部署是指当交付的代码通过评审之后，自动部署到生产环境中。



本资料版权归 KEUC 所有，未经允许不得私自以任何形式传播，违反必究。

简介

遇到的问题

基于Kubernetes的容器化实践

展望

Q&A

主办： CLOUD NATIVE COMPUTING FOUNDATION caicloud 才云 K8S Meetup 中国社区

本资料版权归 KEUC 所有，未经允许不得私自以任何形式传播，违反必究。

遇到的问题

环境部署的效率问题

2016年在测试部门的一次调查中显示，环境部署问题占用了整个测试过程的5%~7%的时间。上线时间长，其中配置问题也占了部分因素

产品-A

- 开发环境
- 联调环境
- 功能测试环境
- 验收测试环境
- 性能测试环境
- Stage环境

服务器资源申请 | 虚拟机创建 | 依赖包工具安装 | 公共组件安装 | 配置项修改确认 | 应用打包部署

首次部署时间长：

- 功能测试环境部署2-5天不等，
- 性能测试环境部署慢的可能会大于5天

非首次部署磕磕绊绊：

- 配置更新问题
- 依赖包更新问题
- 数据库变更问题等影响测试效率

服务器资源紧张，利用率低

- 由于部署效率低，服务器回收较难
- 资源利用率较低，造成资源紧张
- 性能测试时服务器申请尤为困难

部署自动化程度低，对人的依赖较大，不同的环境服务器的变更，重新部署时效率提升有限，影响开发测试效率

开发测试环境

沟通问题

配置问题

SQL问题

性能问题

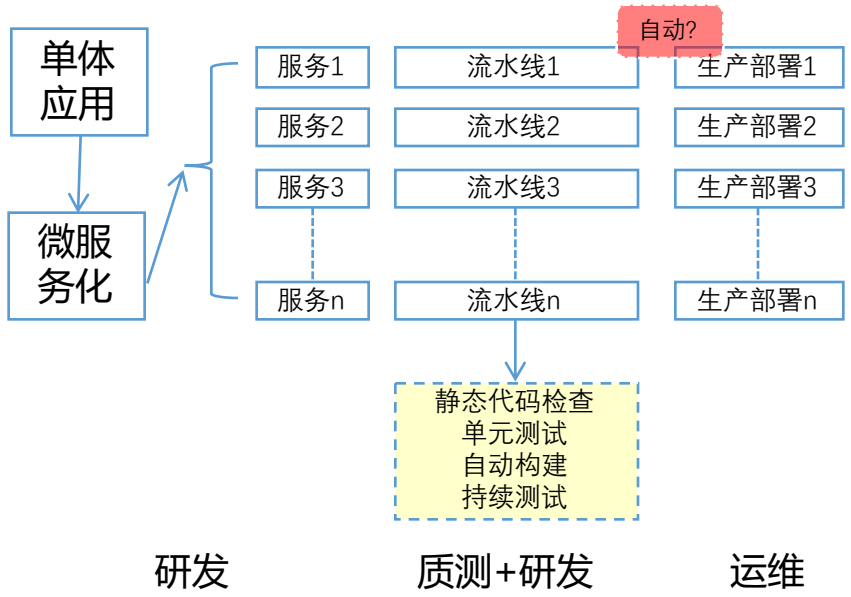
每当夜深人静
一帮人在默默的坚持着

生产环境

本资料版权归 KEUC 所有，未经允许不得私自以任何形式传播，违反必究。

遇到的问题

敏捷化，快速的有质量进行持续交付，在架构上也需要进行微服务化的改造



更大的挑战

- 微服务化改造使得需要部署的模块更多
- 部署环境时配置、依赖将变得更加复杂
- 环境多：Dev、QA、Stage、Production
- 持续审查、持续构建，持续测试的压力更大

Kubernetes+Docker

- Kubernetes + Docker提供诸多能力，如高可用性、自我修复、弹性伸缩、服务发现、负载均衡、配置管理等

对于新接触的技术，有疑虑！从测试开始！质量测试部从2016年Q4开始Kubernetes+Docker的实践

本资料版权归 KEUC 所有，未经允许不得私自以任何形式传播，违反必究。

简介

遇到的问题

基于Kubernetes的容器化实践

展望

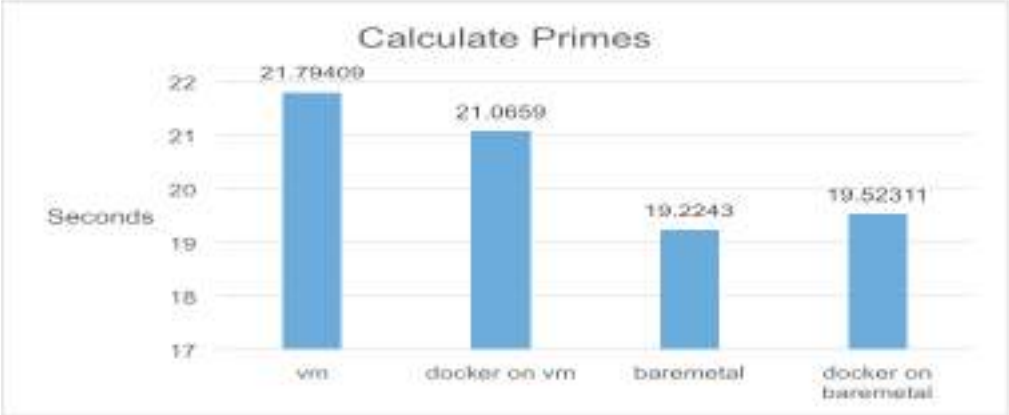
Q&A

主办：CLOUD NATIVE COMPUTING FOUNDATION caicloud 才云 K8S Meetup 中国社区

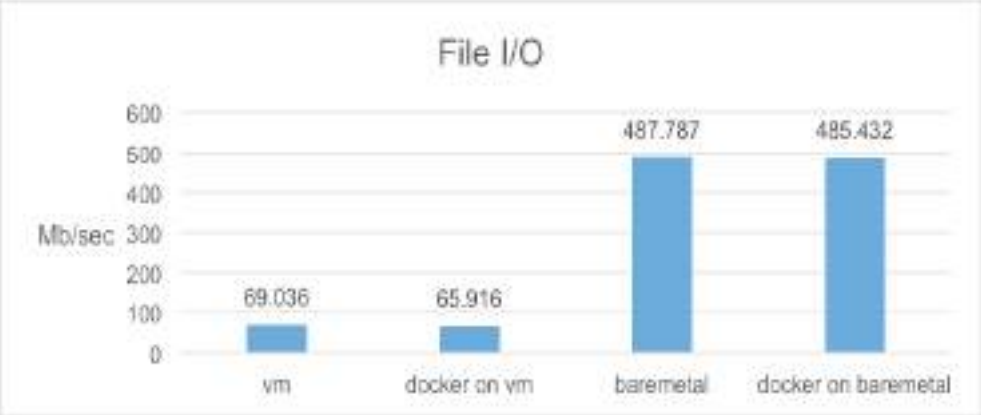
本资料版权归 KEUC 所有，未经允许不得私自以任何形式传播，违反必究。

基于Kubernetes的容器化实践

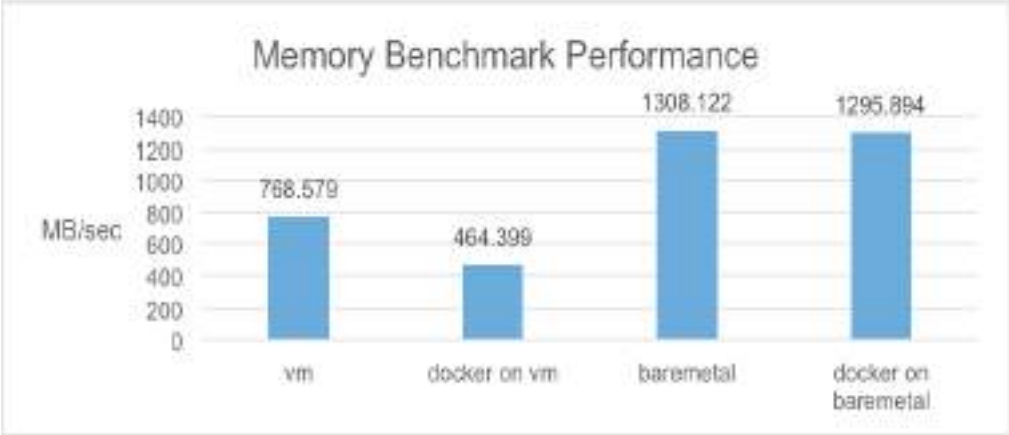
性能测试实践 – 基础性能对比



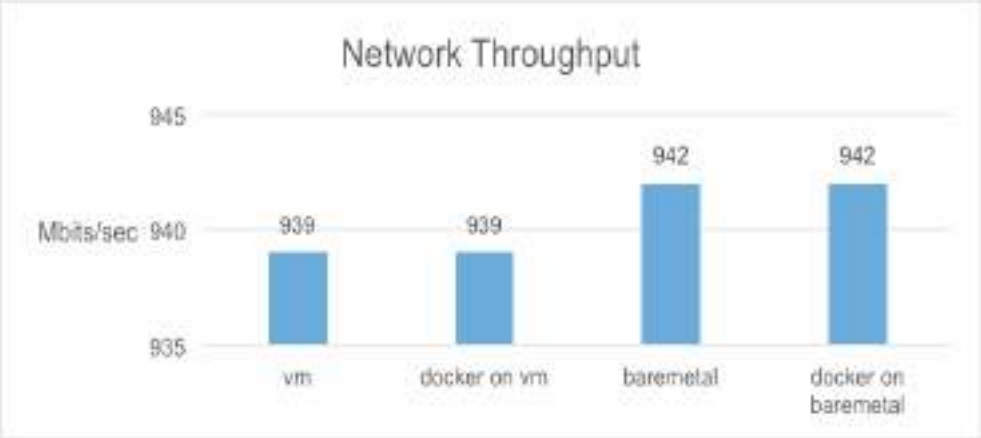
CPU：寻找最大素数



文件 IO：本地文件读写速度



内存：在内存中传输16G数据，统计内存带宽



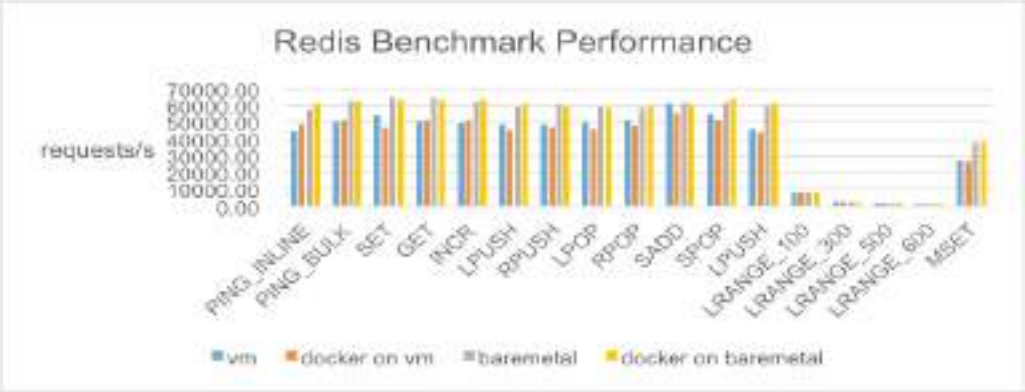
网络：网络带宽

主办：

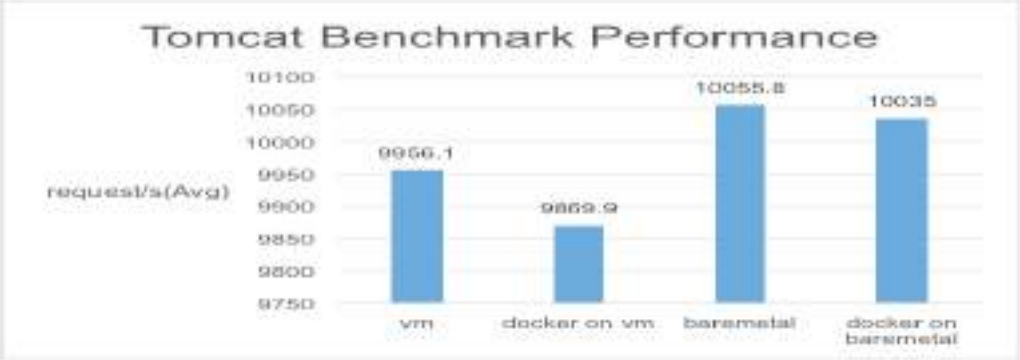
本资料版权归 KEUC 所有，未经允许不得私自以任何形式传播，违反必究。

基于Kubernetes的容器化实践

性能测试实践 – 中间件对比

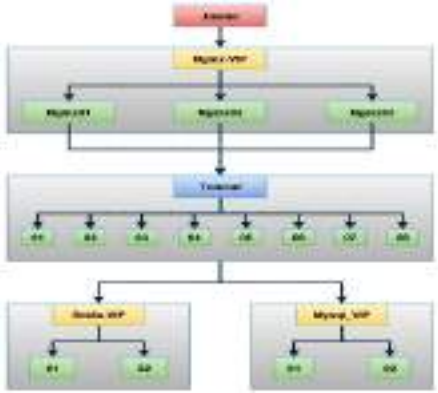


Redis：使用Redisbench进行测试

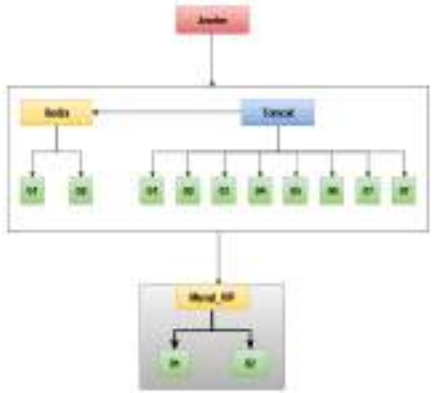


Tomcat：Jmeter简单压测

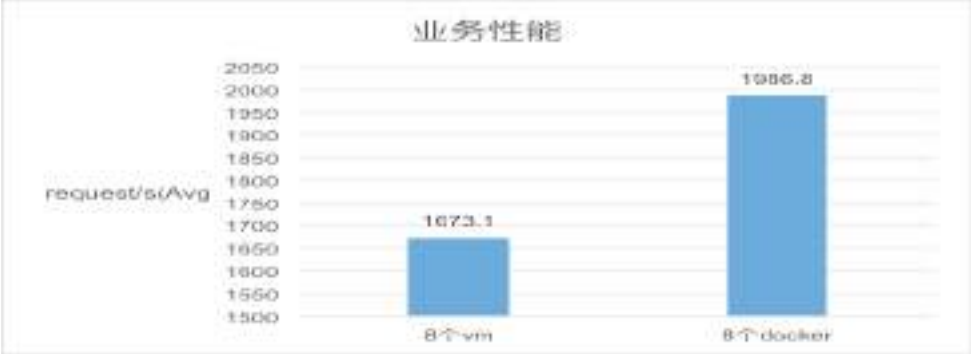
性能测试实践 – 业务性能对比



虚拟机



Kubernetes+docker



简单业务接口压测性能对比

主办：CLOUD NATIVE COMPUTING FOUNDATION caicloud 才云 中国社区

本资料版权归 KEUC 所有，未经允许不得私自以任何形式传播，违反必究。

基于Kubernetes的容器化实践

测试环境容器化

CI/CD

Registry

物理机
资源池

资源监控

日志监控

存储

- CI/CD: Jenkins
- Registry: Harbar
- 物理机服务器资源（CentOS7.2）
- Kubernetes
- Docker
- 资源监控 – Prometheus
- 日志监控 – ELK
- 存储 – NFS/GlusterFS

- 50%产品测试环境容器化
- 积累了适用于公司产品的基础镜像，包括Centos、JDK、Jetty、Tomcat、Redis、RabbitMQ、ES、ZK、Nginx等
- 测试人员即掌握了Kubernetes和Docker的使用又提升了对产品部署架构、配置、依赖的了解，促进了测试深度

本资料版权归 KEUC 所有，未经允许不得私自以任何形式传播，违反必究。

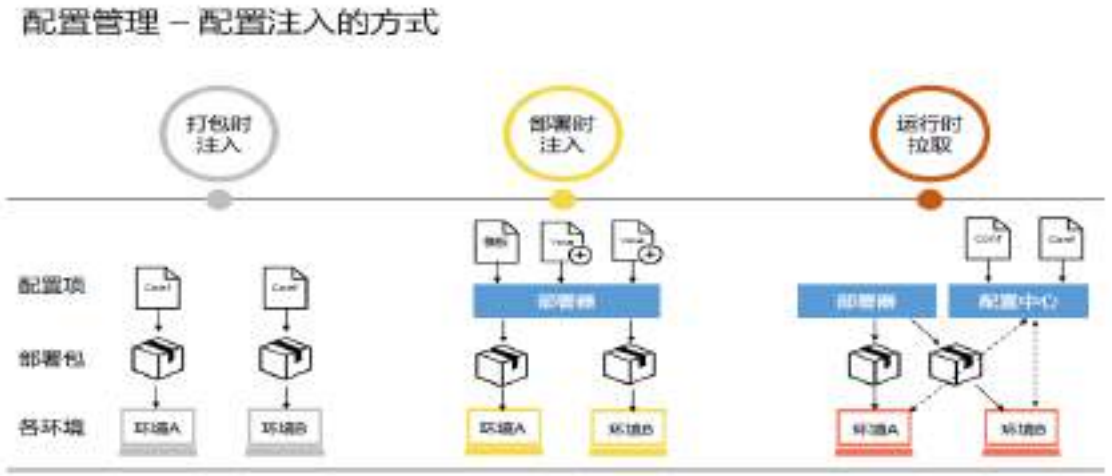
基于Kubernetes的容器化实践

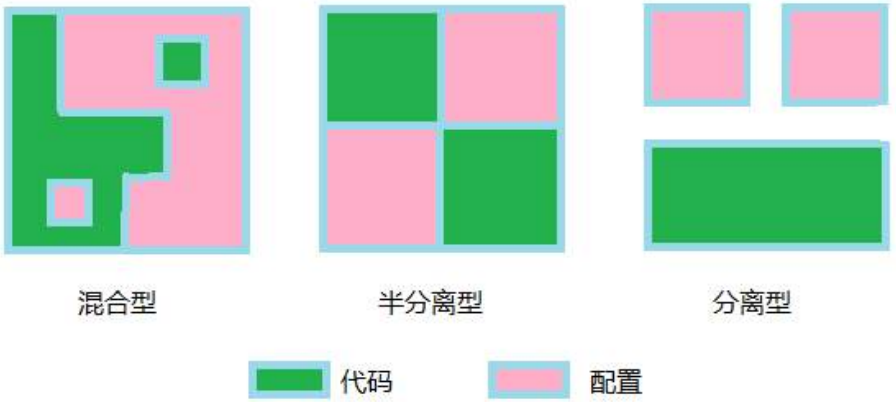
促进配置分离

容器化过程中，我们发现了不少配置问题。比如

- IP类配置不灵活
- JVM配置不灵活
- 配置分离不彻底

以上的问题都会导致容器环境部署出现问题，这在之前就会导致开发、测试、生产环境的配置失败的问题





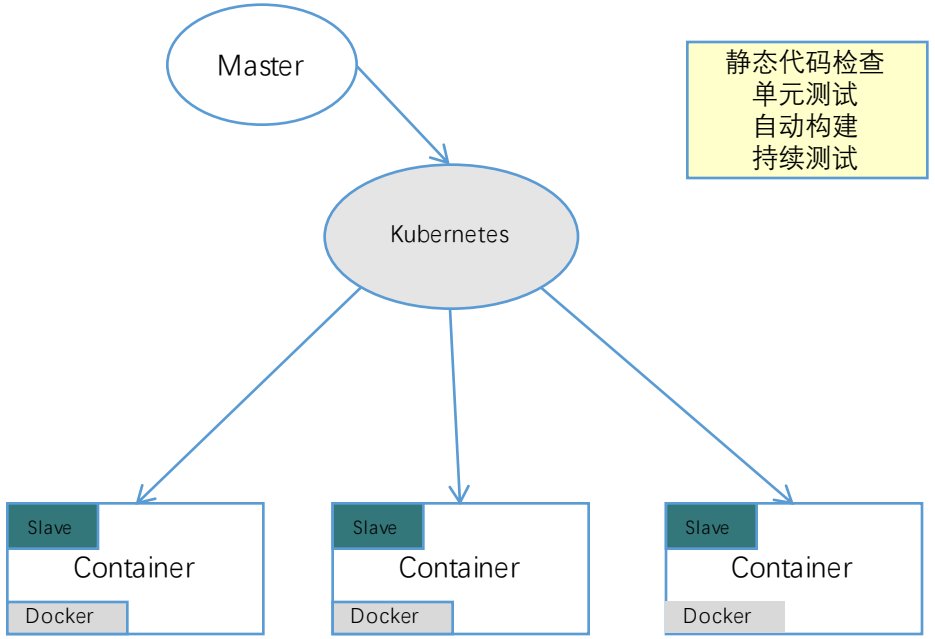
- 代码组织方式的优化：
  - 混合型：配置与代码放在同一个代码库中或同一工程下，甚至是代码目录与配置目录、代码文件与配置文件混杂在一起。
  - 半分离型：配置与代码虽放在同一个代码库中或同一工程下，但是往往都是能够以明确的目录进行区分的。
  - 分离型：代码库与配置库彻底分离，代码与配置的变更互不影响。开发可以更加专注于业务逻辑；测试、运维可以保证配置、环境独立性不受其他角色干扰。

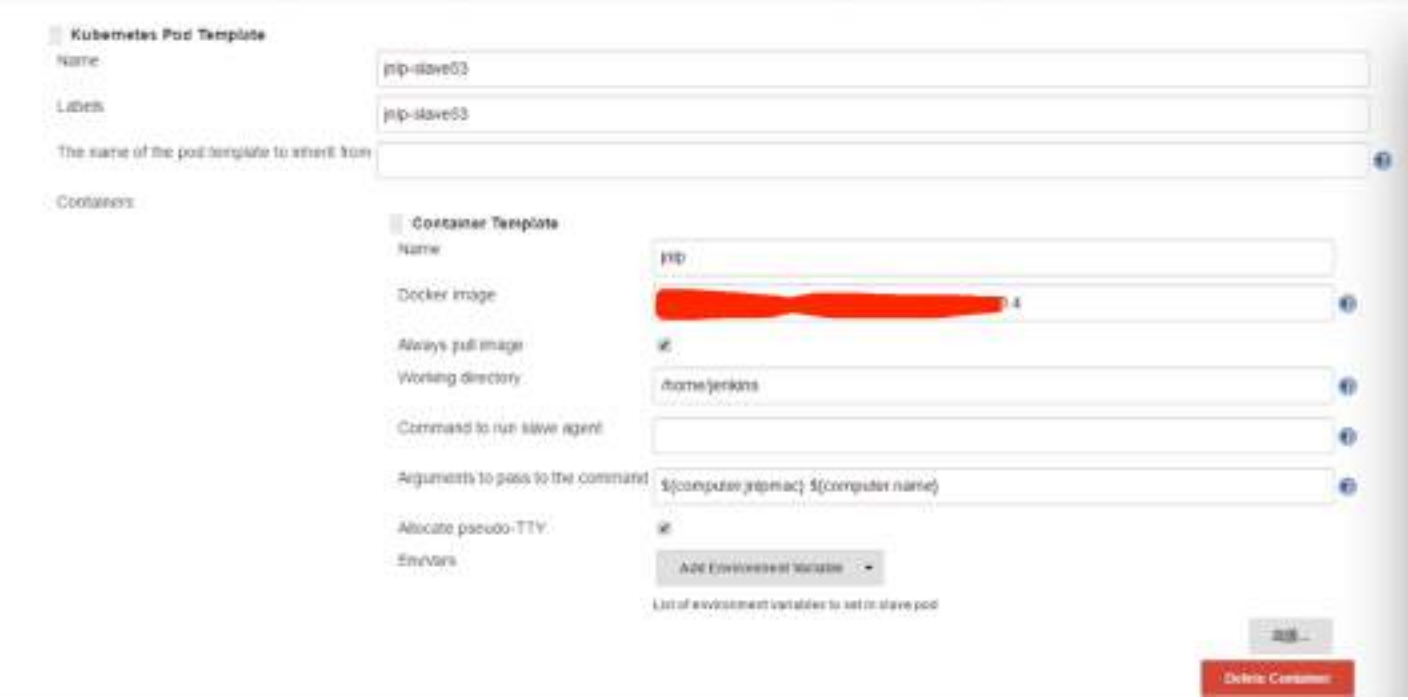
本资料版权归 KEUC 所有，未经允许不得私自以任何形式传播，违反必究。

2017 China Kubernetes End User Conference KEUC 2017
kubernetes 中国用户大会
2017.10.15 / 中国·杭州

可伸缩的持续集成

随着业务和服务模块的增长Jenkins的压力也随之增长，下图是利用Jenkins Kubernetes plugin实现的动态的jenkins slave

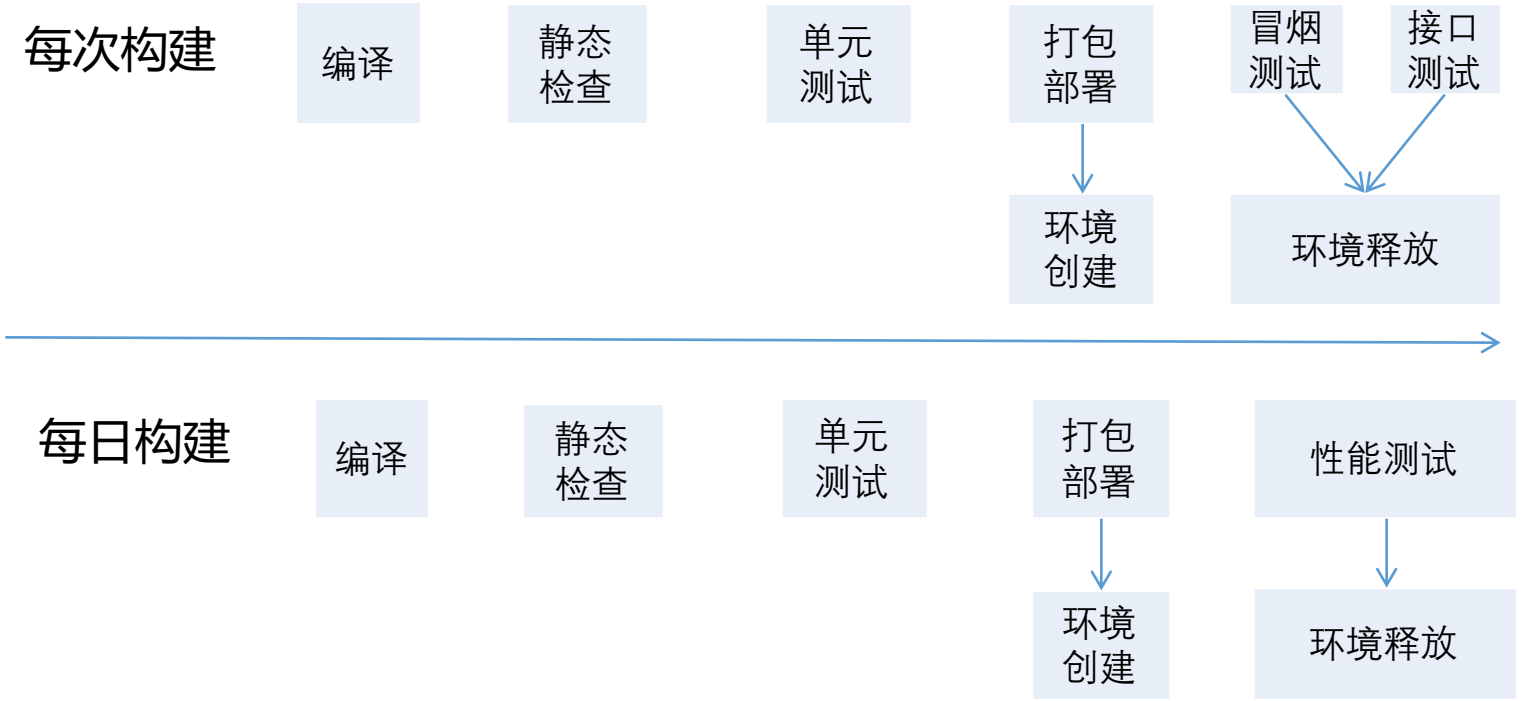




本资料版权归 KEUC 所有，未经允许不得私自以任何形式传播，违反必究。

基于Kubernetes的容器化实践

持续测试



为了能够更快有质量的做到持续交付，我们希望能做更多，更频繁的测试，包括冒烟、接口自动化测试、甚至是性能自动化测试
Kubernetes可动态的创建和删除测试环境，让自动化测试环境得到了保障

本资料版权归 KEUC 所有，未经允许不得私自以任何形式传播，违反必究。

基于Kubernetes的容器化实践

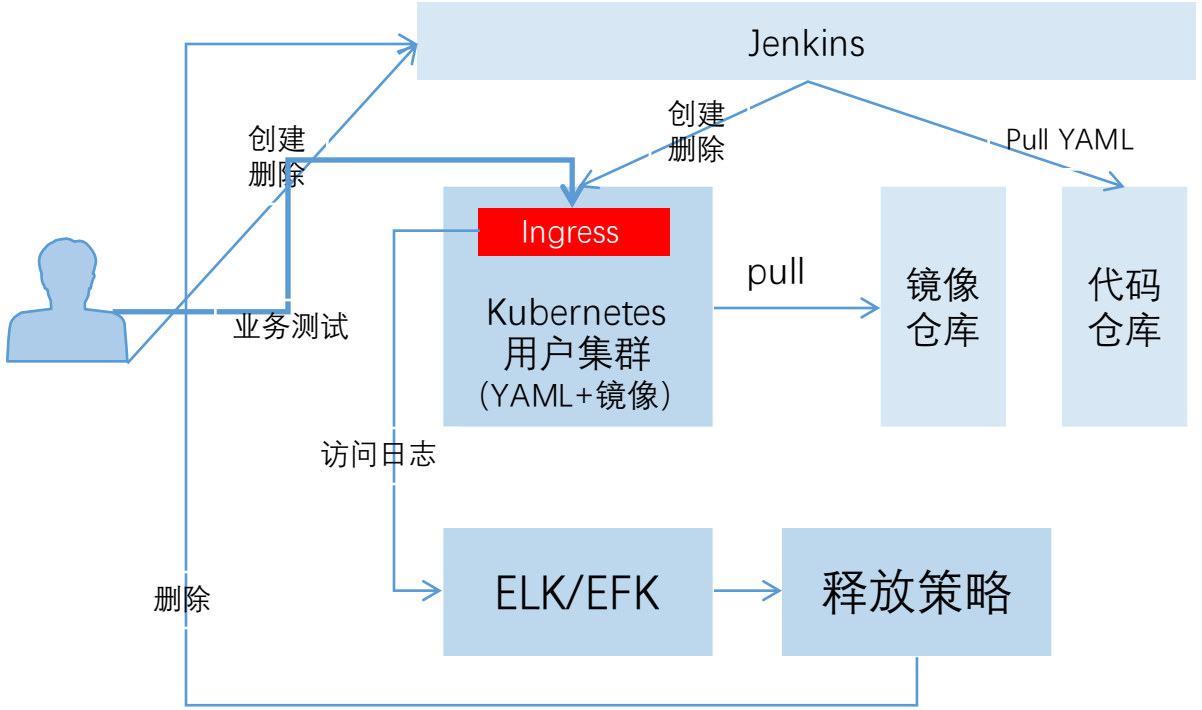
智能的测试环境

问题：

环境搭建不易，释放难
产品业务越来越多，对资源的需求越来越大，造成资源紧张，资源利用的效率比较低。
期望通过Kubernetes的使用，解决该问题

实践：

- Kubernetes使得环境搭建变得容易，分钟级、甚至秒级创建
- 业务访问都通过Ingress
- 通过ELK/EFK监控Ingress访问日志，识别业务
- 根据释放策略，发起应用删除指令，达到环境的自动释放功能
- 需要时通过Jenkins自动创建



示意图

主办：CLOUD NATIVE COMPUTING FOUNDATION caicloud 才云 中国社区

本资料版权归 KEUC 所有，未经允许不得私自以任何形式传播，违反必究。

简介

遇到的问题

基于Kubernetes的容器化实践

展望

Q&A

本资料版权归 KEUC 所有，未经允许不得私自以任何形式传播，违反必究。

展望

杭研持续交付框架

价值交付

版本控制、自动化、内建质量、持续改进

能力管理

- 配置管理
- 代码管理
- 集成管理
- 测试管理
- 环境管理
- 部署管理

平台管理

- 持续交付平台
- 可视化平台
- 立体化监控平台

管理过程

- 持续交付文化
- 数据度量文化
- 灰度实施
- 持续改进

构建实践
自动化构建；构建脚本从IDE分离；集中放置源码；针对所有环境构建；每日构建；快速构建；分阶段构建

持续审查
代码复杂度
持续代码升级；减少重复代码；判断代码覆盖率；代码复用率

持续测试
代码复杂度
自动化单元/冒烟/功能/专项测试；让测试过程可重复；将测试快慢分离

持续部署
一次构建，四处运行；构建结果打上标签；执行所有的测试；回滚部署能力；灰度部署能力

持续反馈
反馈软件的服务状态；持续的反馈机制

可靠可重复交付的流水线

主办： CLOUD NATIVE COMPUTING FOUNDATION caicloud 才云 中国社区

本资料版权归 KEUC 所有，未经允许不得私自以任何形式传播，违反必究。

2017 China Kubernetes End User Conference
kubernetes 中国用户大会
KEUC 2017
2017.10.15 / 中国 · 杭州

Q&A

Thank you for your time

主办：CLOUD NATIVE COMPUTING FOUNDATION caicloud 才云 K8S Meetup 中国社区