大数据在物流行业应用突破

——大规模云上调度实践

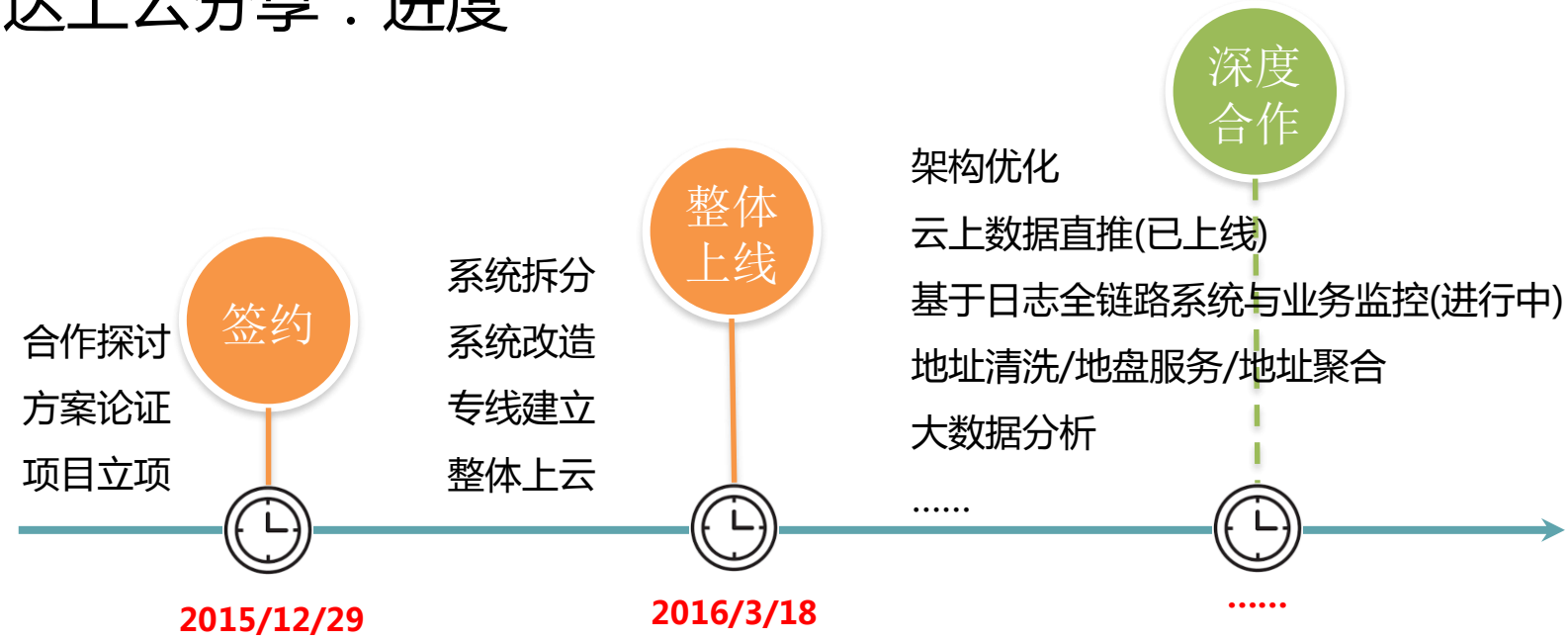
张磊
韵达快运集团 高级总监

主办单位：杭州 Alibaba Group 阿里巴巴集团 战略合作伙伴：intel

目 录
content

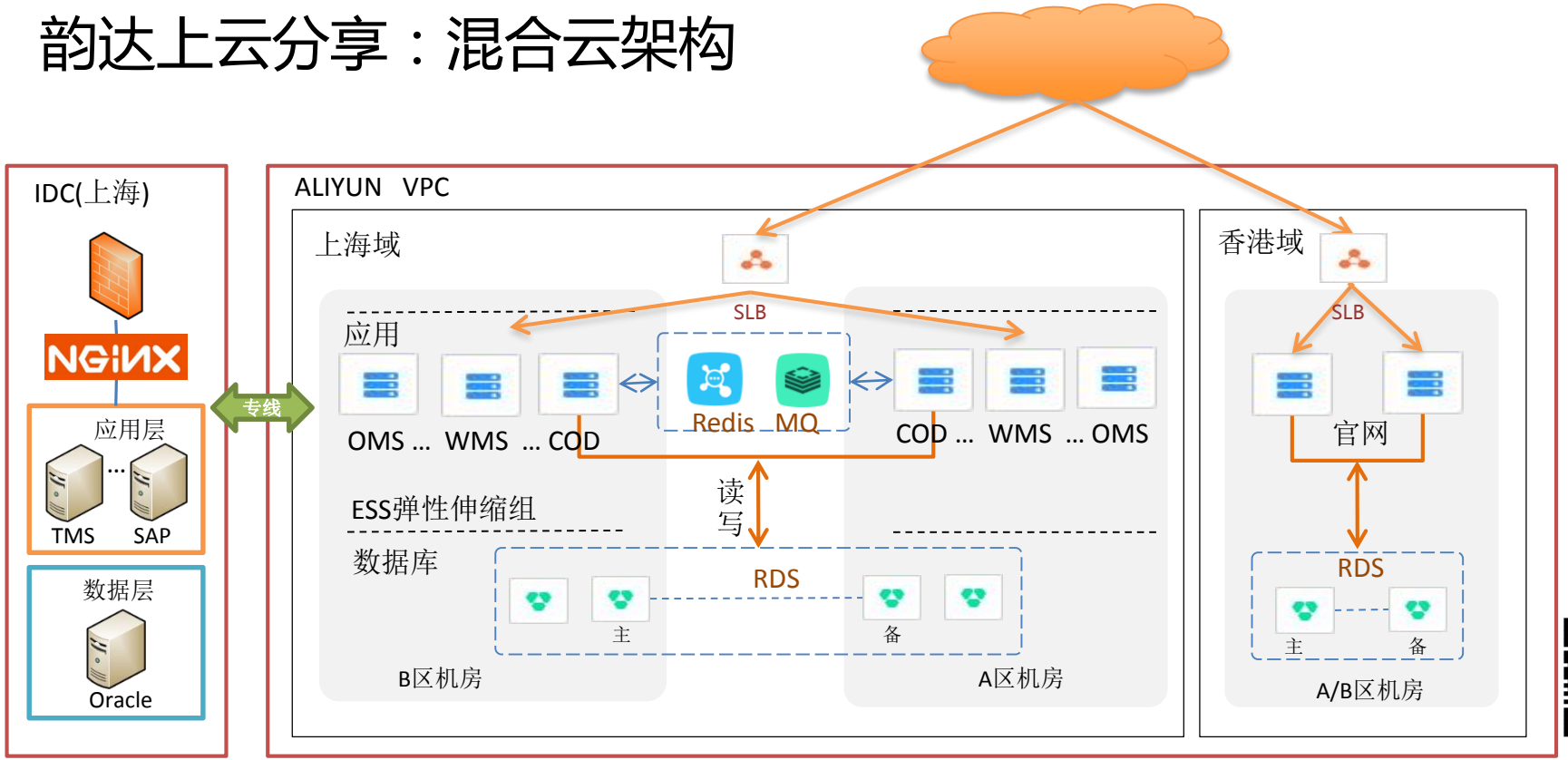
扫码观看大会视频

韵达上云分享：进度

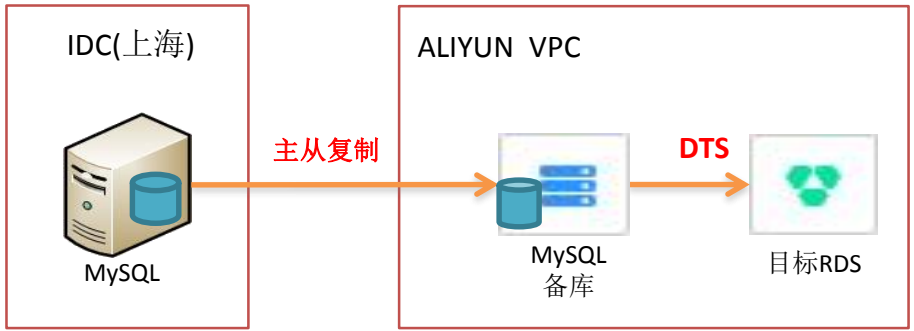


扫码观看大会视频

韵达上云分享：混合云架构



韵达上云分享：碰到哪些问题？如何解决？



数据迁移：

部分数据库数据量TB级别，直接利用DTS迁移线上数据库，会对线上造成影响。

1）中间库：云上ECS上安装MYSQL，作为云下MySQL的备库；

2）目标RDS：使用DTS，从中间库向目标RDS进行数据同步(含数据迁移)；

本方案可同时实现存量数据迁移、增量数据同步到云上，数据同步的网络延迟毫秒级（DTS同步速度理论上可达到70Mbps）

韵达上云分享：成本与效益

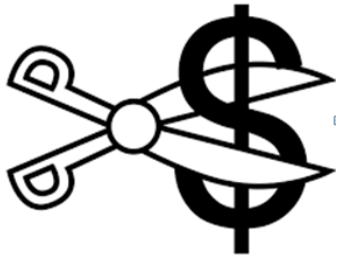
人员投入

6人(韵达)

2人(物流云)

上云项目周期

2个月



云资源使用成本

相比之前自建硬件

投入成本下降

云上运维

So easy！

扫码观看大会视频

二、云上资源调度实践

变革软件交付方式的技术:Docker

--“没有集装箱，不可能有全球化。”

什么是Docker？

- Docker基于容器技术的轻量级虚拟化解决方案
- Docker是容器引擎，把Linux的cgroup、namespace等容器底层技术进行封装抽象，为用户提供了创建和管理容器的便捷界面（包括命令行和API）
- Docker 是一个开源项目，诞生于 2013 年初，基于 Google 公司推出的
- Go 语言实现
- 微软，红帽Linux，Oracle等主流IT厂商在各自产品增加对Docker的支持。

Docker原理

Docker会在隔离的容器中运行进程。当运行docker run命令时，Docker会启动一个进程，并为这个进程分配其独占的文件系统、网络资源和以此进程为根进程的进程组。在容器启动时，镜像可能已经定义了要运行的二进制文件、暴露的网络端口等，但是用户可以通过docker run命令重新定义（译者注：docker run可以控制一个容器运行时的行为，它可以覆盖docker build在构建镜像时的一些默认配置。

为什么要使用Docker？

- 作为一种新兴的虚拟化方式，Docker 跟传统的虚拟化方式相比具有众多的优势。
- 首先，Docker 容器的启动可以在秒级实现，这相比传统的虚拟机方式要快得多。
- 其次，Docker 对系统资源的利用率很高，一台主机上可以同时运行数千个 Docker 容器。
- 容器除了运行其中应用外，基本不消耗额外的系统资源，使得应用的性能很高，同时系统的开销尽量小。传统虚拟机方式运行 10 个不同的应用就要起 10 个虚拟机，而Docker 只需要启动 10 个隔离的应用即可。

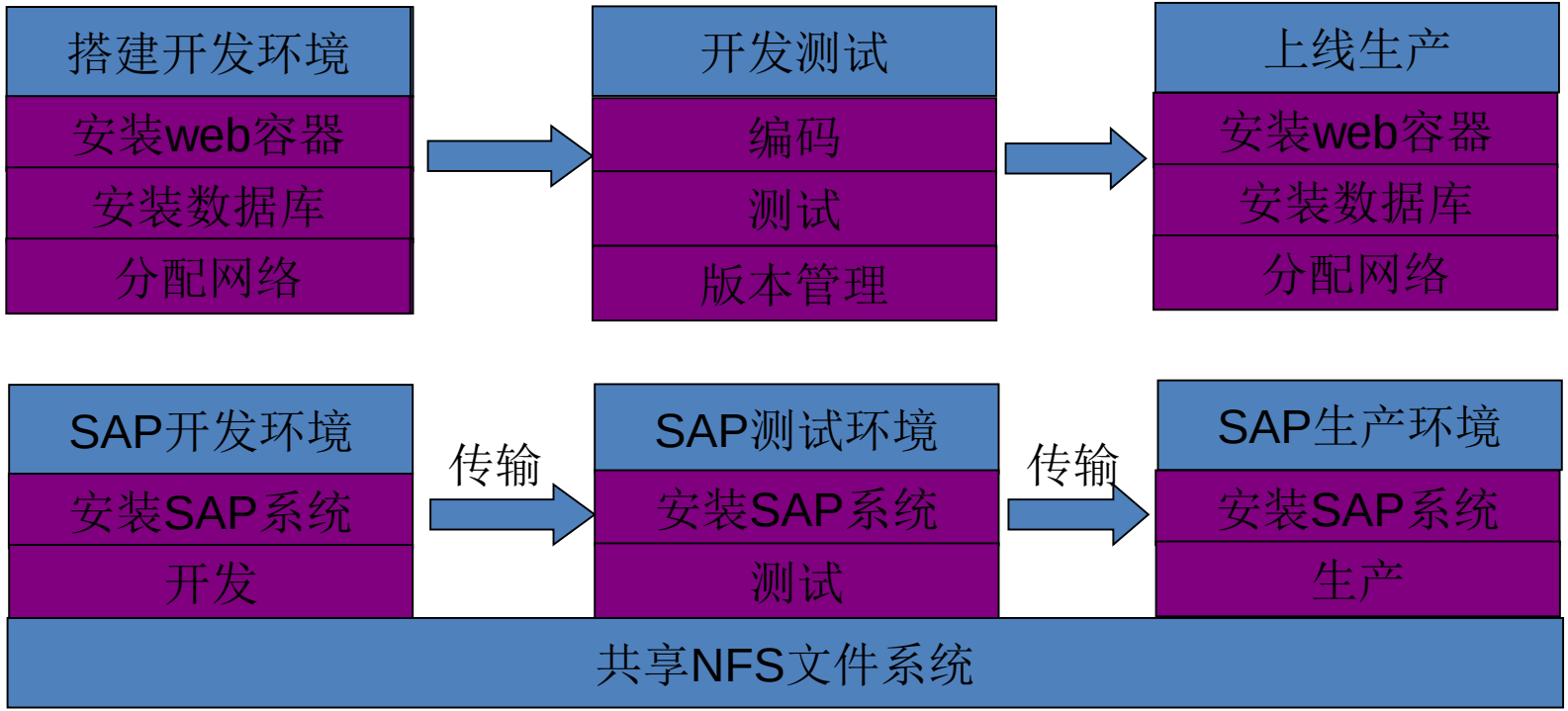
更快速的交付和部署

- 对开发和运维（devop）人员来说，最希望的就是一次创建或配置，可以在任意地方正常运行。
- 开发者可以使用一个标准的镜像来构建一套开发容器，开发完成之后，运维人员可以直接使用这个容器来部署代码。 Docker 可以快速创建容器，快速迭代应用程序，并让整个过程全程可见，使团队中的其他成员更容易理解应用程序是如何创建和工作的。 Docker 容器很轻很快！容器的启动时间是秒级的，大量地节约开发、测试、部署的时间。

对比传统虚拟机总结

	Docker	虚拟机
启动速度	秒级	分钟级
复杂度	基于内核的namespace技术，对现有基础设施的侵入较少	部署复杂度较高，并且很多基础设施不兼容
执行性能	在内核中实现，所以性能几乎与原生一致	对比内核级实现，性能较差
可控性	依赖简单，与进程无本质区别	依赖复杂，并且存在跨部门问题
体积	与业务代码发布版本大小相当MB级别	GB级别
并发性	可以启动几百几千个容器	最多几十个虚拟机
资源利用率	高	低

传统型软件开发、测试、上线过程



传统型软件开发、测试、上线过程不足之处

1、资源利用效率低

2、单物理机多应用无法有效隔离（进程空间，cpu资源，磁盘）

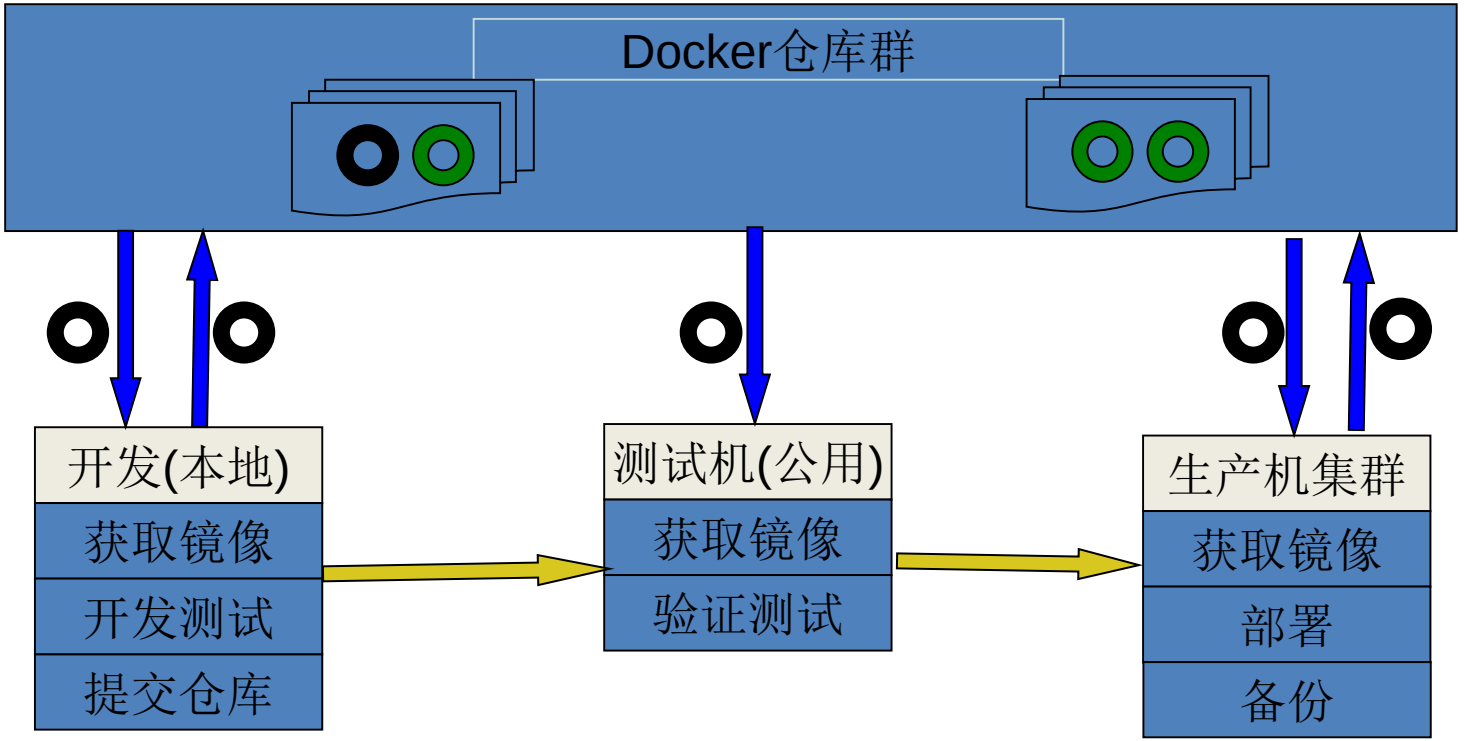
3、运维部署不便

4、测试、版本管理复杂

5、迁移成本高

6、传统虚拟机，空间占用大，启动慢，管理复杂

以Docker为单位的开发部署流程设计



应用Docker化交付的过程

代码：
…/src/main
…web.xml
…一个描述文件
… …

环境定义：
… OS版本
… 中间件类型
… 环境变量配置
… 编译之后的应用包
… …

运行镜像：
… container
… container
… …

Docker-compose描述依赖环境

1. 应用级别的依赖怎么办？

比如，一个应用依赖一个mysq数据库。docker官方提供docker-compose解决这个问题，管理docker image

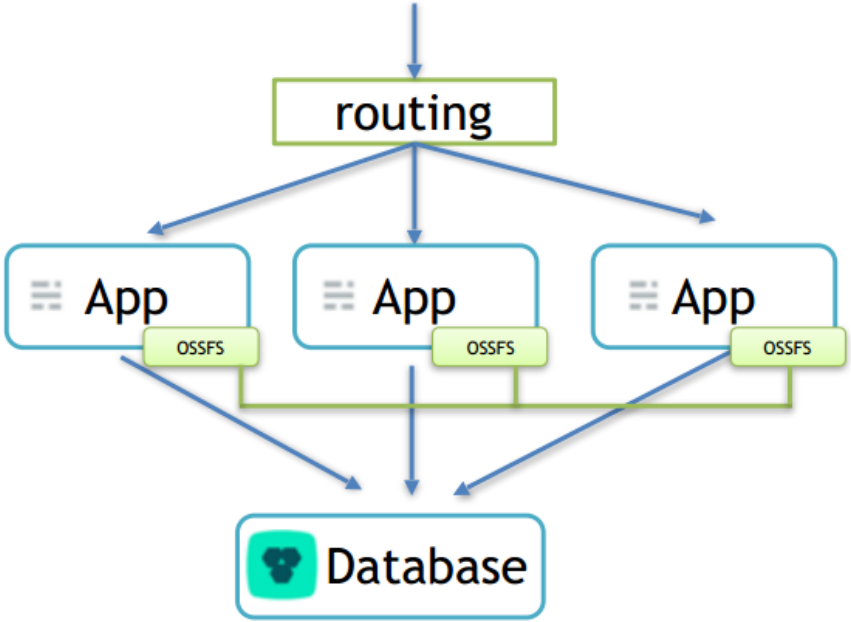
2. 用来docker image的编排

(1)write your dockerfile

(2)write your compose.yml file

compose.yml管理多个image，管理image启动顺序等，然后要把多个image一起运行只需要用一个命令就可以全部运行起来

用Docker描述集成/运行环境



声明观念：

- I need 负载均衡（haproxy，Nginx）
- I need 数据库（mysql)
- I need 文件存储(通过-v，ossfs)
- I need 缓存服务(redis,kv-store)
- ...

扫码观看大会视频

Docker踩过的“坑”

1. dockerfile不要放到代码根目录下

   docker编译dockerfile时，会把dockerfile同级目录所有文件传递给docker deamon，避免大量文件传递给docker deamon，导致内存爆掉

2. Dockerfile行数尽可能少，否则最终生成的镜像会很大

3. Dockerfile注意字符编码环境变量以及时区设置

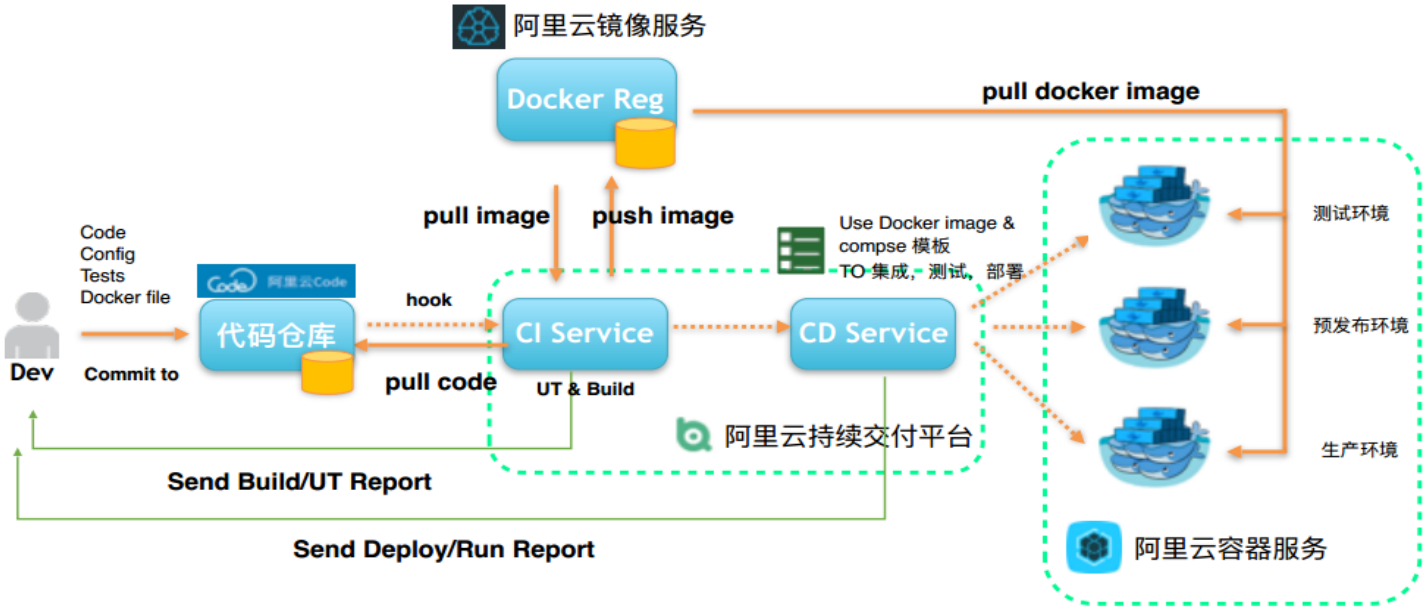
Docker相对虚拟机不足之处

1.安全性问题。docker目前并不能分辨具体执行指令的用户，只要一个用户拥有执行docker的权限，那么他就可以对docker的容器进行所有操作，不管该容器是否是由该用户创建。比如A和B都拥有执行docker的权限，由于docker的server端并不会具体判断docker client是由哪个用户发起的，A可以删除B创建的容器，存在一定的安全风险

2.docker目前还在版本的快速更新中，细节功能调整比较大。一些核心模块依赖于高版本内核，存在版本兼容问题

三、未来与挑战

完整的容器持续交付流程



扫码观看大会视频

未来与挑战：TODO

- **优化**：Docker本身的性能优化

比如：dockercontainer 在stdout/stderr有大量数据传输会导致内存泄露

- 监控：服务级别的监控
- 负载均衡：容器根据机器负载情况自动迁移
- 统一管理页面：统一的根据服务来管理的WEB页面
- 微服务：应用程序功能模块拆分，适应Docker

2016 The Computing Conference

THANKS