

## 讨论课03 — 复用云技术

---

### 容器技术的理解

- 资源独立、隔离

每个容器内都包含一个独享的完整用户环境空间，并且一个容器内的变动不会影响到其他容器的运行环境。为了能达到这种效果，容器技术使用了一系列的系统级别的机制诸如利用Linux namespaces来进行空间隔离，通过文件系统的挂载点来决定容器可以访问哪些文件，通过cgroups来确定每个容器可以利用多少资源。此外容器之间共享同一个系统内核，这样当同一个库被多个容器使用时，内存的使用效率会得到提升。

- 环境一致性

开发工程师完成应用开发后build一个docker image，基于这个image创建的container像是一个集装箱，里面打包了各种“散件货物”（运行应用所需的程序，组件，运行环境，依赖）。无论这个集装箱在哪里：开发环境、测试环境、生产环境，都可以确保集装箱里面的“货物”种类与个数完全相同，软件包不会在测试环境缺失，环境变量不会在生产环境忘记配置，开发环境与生产环境不会因为安装了不同版本的依赖导致应用运行异常。

- 轻量化

相比传统的虚拟化技术（VM），使用docker在cpu, memory, disk IO, network IO上的性能损耗都有同样水平甚至更优的表现。Container的快速创建、启动、销毁受到很多赞誉。

- Build Once, run everywhere

未来可能仅需少数几个运维人员运维超大规模装载线上应用的容器集群

### Tahiti项目为什么需要使用容器技术

在使用Tahiti之前，每个用户都得进行一系列的配置工作。首先需要下载Java运行环境，下载配置Protobuf，为Maven配置环境变量，编译Protocol Serializer，运行jar包。有些用户可能只是想做简单的功能验证，而他们可能就

要花费几十分钟去下载和配置应用本身，并且下次换一个服务器使用仍然要重复前面的经历，更别提让一个新手来安装应用。我们可以使用Docker技术做一个一键创建服务，用户一键就可以完成配置环境和依赖的所有步骤，直接使用Tahiti。

## 容器技术方案选择

希望将该项目部署在OpenShift平台上，OpenShift是Red Hat公司推出的PaaS（Platform as a Service，平台即服务）。开发人员可以直接在平台上部署应用进行运行、测试。使用OpenShift，可以：（1）快速上传和部署应用代码；（2）基于开发堆栈，不用担心应用被锁定；（3）支持Java, Ruby, Node.js, Python, PHP等多种语言；（4）十分容易的实现对应用的扩展。

### 参考资料

1. <http://blog.csdn.net/gaoyingju/article/details/49616295>
2. <http://dockone.io/article/750>
3. <http://dockone.io/article/326>
4. <http://www.programgo.com/article/62612736437/>
5. <http://www.live-in.org/archives/1818.html>