**基本Docker的分布式应用控制系统**

**关键设计**

**1 总体设计**

系统Nginx做受控端的反向代理，Docker作为受控端监听本地地址的某个端口，Iptables用来做主机防火墙，ThinkPHP作为后台开发框架，通过HTTPS协议通信，主控端使用MySql作为关系型存储数据库，前端框架使用Angular且前端工程构建使用Gulp，单元测试和黑盒测试作为整个项目主要测试方式。

大概流程如下：用户在前端输入容器操作，发送请求至后台，后台收到请求进行一系列处理，根据操作发送POST或GET请求至Docker服务端监听端口，Docker服务端根据监听到的内容解析Docker Remote API，执行对应的curl命令。

本项目主要实现Dcoker常用命令集的可视化操作，在此基础上，为保证处理安全性，还将添加权限设置相关功能。从Dcoker命令控制，到数据库设计，再到PHP开发，最后前端展示。

**2 系统框架结构图**



**3 具体功能设计**

基于Dcoker的分布式应用控制系统主要包括主机管理、Container管理、Image管理以及系统日志等模块。具体内容如下：

镜像管理：用户可以从阿里云（或DCloud）拉取镜像，查看所有镜像列表，也可以对想要查看的镜像进行筛选，如果镜像不再被需要，也可删除该镜像。

容器管理：容器是Docker技术的核心，所以容器操作比较多。用户除了可以搜索查看容器列表信息外，还可以对容器进行启动、重启、暂停、恢复暂停、重命名、终止、移除等操作。容器操作一般会生成记录，所以可以通过查看容器日志排查操作异常。容器运行过程中，查看容器进程也是必不可少的。

主机管理：系统会显示主机列表以及主机分组信息，用户可以查看到某一主机具体在哪个分组里，主机分组是可以进行动态创建和删除的。

系统日志：系统会记录登录用户的一切操作，生成系统日志，运行环境出现问题时，方便排查以及责任追究。对于一定日期后的日志，管理员也可进行删除。

用户管理：该模块只包含用户登录功能，所需账号和密码由系统自行分配，不能通过注册实现。