

# 基于 Docker 下 DevOps 系统的设计与实现研究

陈 敏 杨 阳

(中国电建集团华东勘测设计研究院有限公司, 浙江 杭州 311122)

**摘 要:** 笔者简要介绍了 Docker 平台与 DevOps 系统, 在 Docker 平台的支持下设计 DevOps 系统, 以设计总目标为指导, 构建数据库集群、网络管理、镜像以及 DNS 管理等多个核心模块, 并阐述各个模块的功能。此外, 通过实例分析的方式, 验证系统功能, 展现系统的应用价值。应用 DevOps 系统, 能够有效克服以往系统设计的弊端, 同时, 将其应用到 IT 企业, 能够显著提升工作效率。

**关键词:** Docker; DevOps 系统; 技术开发

**中图分类号:** TP393.09 **文献标识码:** A **文章编号:** 1003-9767 (2019) 11-097-03

## Research on the Design and Implementation of DevOps System Based on Docker

Chen Min, Yang Yang

(PowerChina Huadong Engineering Corporation Limited, Hangzhou Zhejiang 311122, China)

**Abstract:** The author briefly introduces the Docker platform and DevOps system, and designs the DevOps system with the support of Docker platform. Guided by the overall design goal, the author constructs several core modules such as database cluster, network management, mirror and DNS management, and elaborates the functions of each module. In addition, through the way of case analysis, the system functions are verified and the application value of the system is displayed. The application of DevOps system can effectively overcome the drawbacks of the previous system design. At the same time, it can be applied to IT enterprises to significantly improve work efficiency.

**Key words:** Docker; DevOps system; technology development

### 0 引言

现阶段, IT 部门的开发与运维团队缺乏沟通, 造成诸多矛盾, 如开发团队对目标市场变化不敏感、运维团队不清楚开发内容等。为了缓解二者的关系, 跨职能部门应积极采用 DevOps 系统, 消除两个团队之间的矛盾, 开发、测试、运维能够得到一键化部署, 为 Docker 平台提供强有力的支持。

### 1 Docker 与 DevOps 简介

#### 1.1 Docker 平台

Docker 是一个开源应用容器引擎, 开发者可以打包应用及依赖包到一个可移植容器, 然后发布到任何流行的 Linux 机器, 也可以实现虚拟化。Docker 守护进程能够直接与主操作系统通信, 为各个 Docker 容器分配资源, 且使各个容器处于独立状态。虚拟机启动需要数分钟, 而 Docker 容器可在数

毫秒内启动。与虚拟机相比, 其具有操作便捷性、高效部署等特征, 在系统发布与运维等方面得到广泛应用。从技术层面来看, 开发团队将依赖包整合成一个整体, 并放置到一个可移植容器, 该容器内容可借助网络渠道在任意 Linux 容器中应用。

#### 1.2 DevOps 系统

DevOps 作为一种新型工作模式, 以项目开发、测试、部署等服务为主, 有效精简和优化各个环节, 消除重复内容, 显著提升 IT 部门之间的协同度, 真正实现一站式管理。现阶段, DevOps 系统已在多个领域运维管理中得到应用, 主要具备以下特点: (1) 提高运维部门对微型化系统调整的敏感度; (2) 开发人员与软件实际运行环境有机结合, 提高软件项目的可控性; (3) 运维重心发生转移, 从各节点设施逐渐转移到应用程序, 管理效率显著提升; (4) 精简软

**作者简介:** 陈敏 (1981—), 男, 浙江杭州人, 本科, 高级工程师。研究方向: 企业信息化管理、企业流程引擎集成及开发等。

件研发流程,帮助团队缩短交付周期,有效控制运维成本<sup>[1]</sup>。

## 2 DevOps 系统设计与实现

### 2.1 设计目标

本文系统提出图形化 Web 页面,利用 MySQL 数据库完成容器镜像、持续集成等操作,有效精简操作流程。为了克服以往 Docker 网络负载问题,本系统利用 Flannel 插件实现集群管理,将虚拟网络与容器编排相结合,利用 POD 封装 Docker 容器,在标签服务的基础上与 Ngnix 有机结合,有效解决 IP 地址随意变更的问题。该系统通过设置统一管理页面,构建虚拟容器镜像,并结合项目实际情况完成集群拓展,在同一编译镜像的基础上彻底解决开发、测试、运维不相符等问题。通过营造各类依赖环境,完善系统关键动作。比如,“构建”,主要作用是维护程序包版本,生成和应用程序包;“编排”,根据系统拓扑、软件包等依赖关系,完成架构编排与动作设置等。

### 2.2 系统设计

DevOps 系统中,主要包括 Etcd 数据集群、Kubernetes 集群、GitLab 管理模块、网络与域名系统等部门。其中,数据集群具有较高的配置能力,主要负责 Docker 的调度与管理;GitLab 管理模块能够管理和配置源码;集成编译镜像可统一配置环境,进而实现统一编译。Docker 下 DevOps 系统设计主要包括以下几个方面。第一,数据库集群。该设计应确保数据库分布方式完整,通常采用 key-value 存储方式,为整个系统营造安全有利的存储环境,包括多种类型的插件配置等。第二,网络管理。该管理内容主要集中在容器网路、服务容器等方面。网络管理方面,主要利用 Flannel 插件在运行主机中设立虚拟网卡,通过插件服务传输数据包,有效解决跨

主机容器相关网络通信问题,优化网络管理配置,促进容器之间的交流。DNS 解析方面,利用 Kube2sky 插件服务控制子节点的变化情况,条件允许时,将 IP 地址、DNS 关系输入到数据库,利用相应模块深入编辑和处理数据。第三,镜像。Jenkins 镜像可编译源代码,为 GitLab 提供服务,并管理源代码。统一集成镜像作为主要的容器,能够为镜像构建提供诸多帮助,确保构建过程与环境相符合,加快构建速度,克服额外依赖。Docker 下构建的 GitLab 镜像为:

```
# 基础镜像
FROM ubuntu: 16.04
# 作者
MAINTAINER xx" xx@163.com"
# 安装所需要的包
Git-annex &&.rm-rf/var/lib/apt/list/*
# 指明容器对外端口
EXPOSE 443 80 22
# 在镜像中创建路径
CMD[" /assets/wrapper" ]
```

第四,网络与 DNS 管理。立足于网络管理的相关需求,本系统利用 Flannel 插件精简管理流程,并实现多种容器相互连接的目标。实际运行中,全部安装镜像的主机均会构建虚拟网卡,开启插件服务,遵循网络 TCP 协议,完成报文转发、封装等工作,并以各个容器为节点,构建畅通的通信网络。针对 DNS 解析需求,监控处于工作状态的 Service,并将信息写入数据库。SkyDNS 的主要作用是读取和解析数据库中的记录信息,最终获得多个 IP 地址对应的域名<sup>[2]</sup>。

### 2.3 系统运作

DevOps 系统的应用使软件运维工作效率显著提升,本文利用 Docker 技术展示系统运作的主要流程,如图 1 所示。

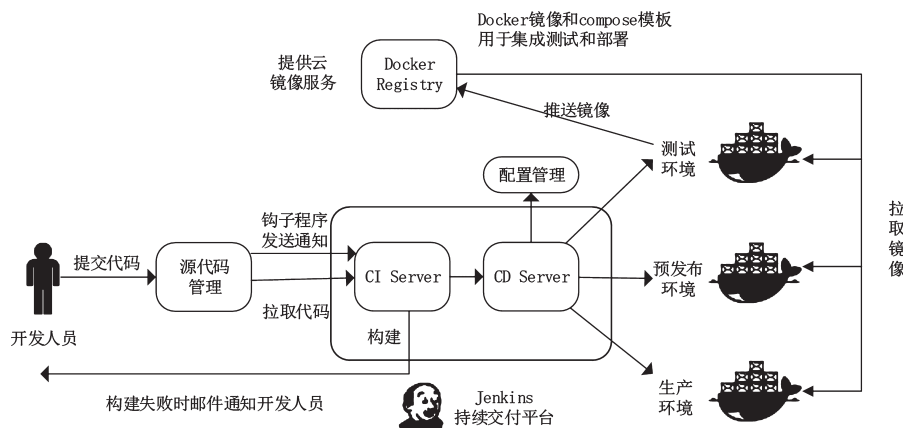


图 1 Docker 下的 DevOps 系统运作流程

本系统主要运作流程如下:第一,开发团队将源代码转变为 Git 格式,并发送到 Github 服务器;第二, Github 服务器钩子程序将代码更新信息发送给 Jenkins;第三, Jenkins 接收到信息后,根据实际需求调取最新代码,获取相应资源后构建和打包项目,并将打包好的项目存储到云镜像容器,

按照指令集构建子节点镜像;第四,子节点对镜像进行实例化,并开启系统中的测试功能调试代码,如若程序运行正常,则将该镜像存储到 Docker Registry;第五,利用 Kubernetes 集群中的持续发布功能,将项目发布至生产、测试环境。

(下转第 101 页)

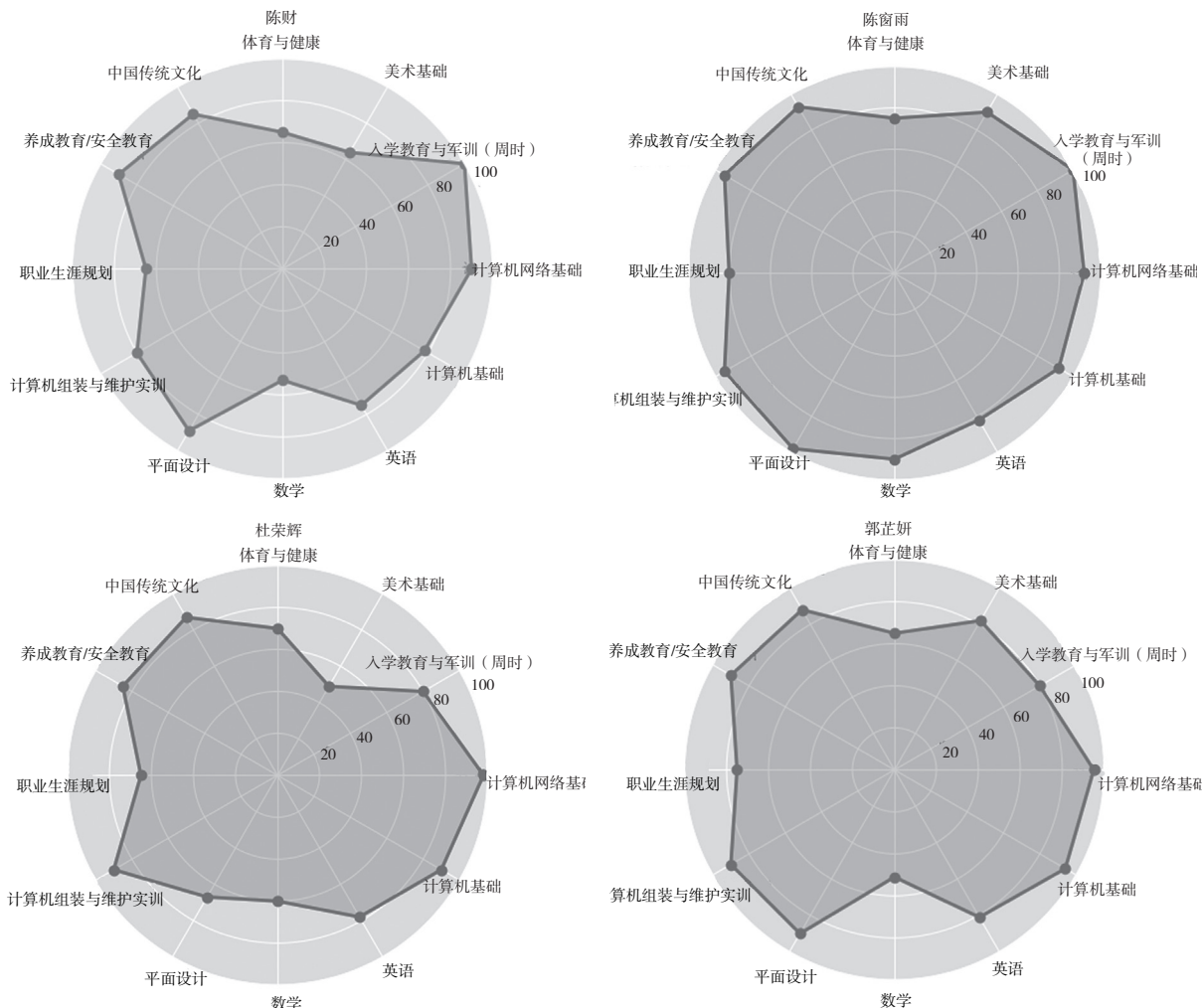


图 1 学生综合成绩分析雷达图谱

- [3] 嵩天,黄天羽,礼欣 .Python 语言 : 程序设计课程教学改革的理想选择 [J]. 中国大学教学 ,2016(2) :42-47.

[4] 刘春茂 ,裴雨龙 .Python 程序设计案例课堂 [M]. 北京 :清华大学出版社 ,2017:135.
- [5] 谢克武 . 大数据环境下基于 Python 的网络爬虫技术 [J]. 软件开发 ,2017(9):44-45.

(上接第 98 页)

### 2.4 系统实现

以某软件开发企业为例,该企业拥有员工百余人,利用 Java 进行软件开发,以 Tomcat 作为中间服务器,并利用多种功能模块处理数据。传统的 DevOps 系统中,集成流程为本地环境下利用 Java 开发新功能,测试完毕后下线,开始测试其他环节,该测试部署服务器中提交的代码无法顺利应用,一点就出错。由于受到多种因素限制,导致整个处理工作难度较高、效率较低。与此相比, Docker 平台的 DevOps 系统能够有效克服上述问题。在整个项目中设立一个构架师,研发、测试、运维人员的工作进度直接上报构架师,一旦发现

异常问题,可及时调整,工作效率显著提升。

## 3 结 语

本系统能够在 Docker 平台的基础上克服以往系统问题,优化工作流程,极大提升工作效率。

## 参考文献

- [1] 凌云 . 基于 Docker 平台的 DevOps 运维系统的研究与改进 [J]. 电脑知识与技术 ,2018,14(26):215-217.
- [2] 丁海斌,崔隽,陆凯 . 基于 Docker 的 DevOps 系统设计与实现 [J]. 指挥信息系统与技术 ,2017,8(3):87-92.