文章编号：2096-1618(2020)05-0505-04

面向 Docker Compose 多容器构建管理工具的设计与实现

胥 柯， 张新有， 李泽慧， 李 彬
(西南交通大学信息科学与技术学院，四川 成都 611756)

摘要：在多容器管理中，Docker Compose 有着举足轻重的地位。但在使用 Compose 进行多容器管理时，由于其语法复杂性，YAML(YAML ain't markup language)文件对格式的严格要求及没有对可能存在于模版文件中的错误进行有效的检测，从而易导致多容器服务构建失败。为了提高工作效率，减少主观错误的发生，设计一个面向 Docker Compose 的多容器构建管理工具是非常必要的，以方便用户持续地集成和快速部署。该工具利用 SSM 框架、可视化编辑、错误检测等关键技术进行实现。测试结果表明该工具使用方便，容器编排效率有明显提高。

关 键 词：SSM 框架；可视化编辑；YAML；Docker Compose
中图分类号：TP311.1 **文献标志码**：A
doi：10.16836/j.cnki.jcuit.2020.05.004

0 引言

容器是一种基于操作系统的虚拟技术，它运行在操作系统之上的用户空间，所有的容器都共用一个系统内核和公共库[1]。2013 年 3 月，dotCloud 公司推出的 Docker 是最成功的容器技术之一，它拥有早期容器技术不具备的一些能力，提供简单、安全地创建和控制容器的接口。开发人员可以将应用程序打包成轻便的 Docker 容器，可以在几乎任何地方使用，无需修改[2]。

DockerCompose 项目是 Docker 官方的开源项目，负责实现对 Docker 容器集群的快速编排。在传统的使用过程中，通过文本编辑 docker-compose.yaml 模版文件的方式缺乏语法错误的检测，且由于 Compose 语法复杂，使配置过程困难且学习成本高。同时，由于 Docker 系统本身没有对 Docker Compose 模版文件提供有效的检测方案，进而验证它的正确性，因此出现任何细微错误都有可能导致多容器构建失败。面对上述问题，本设计实现一个面向 Docker Compose 多容器的构建管理工具，降低学习成本，提高工作效率。

1 相关技术背景

1.1 Compose 工作原理

Compose 允许用户在一个模版文件(YAML 格式)中定义一组相关联的应用容器，只要编写好模版文件，就可以使用 docker-compose up 命令让多个相关容器启动[3]。

Compose 有两个核心概念即 project 和 service。Project：即是通过 Docker Compose 管理的一个项目，被抽象称为一个 project，它是由一组关联的应用容器组成的一个完整的业务单元。Service 则为运行一个或多个相同镜像的服务。一个 docker-compose.yml 定义一个 Docker Compose 的 project。一个项目可以由多个服务(容器)关联而成，Compose 面向项目进行管理。Compose 的一次使用过程如图 1 所示。

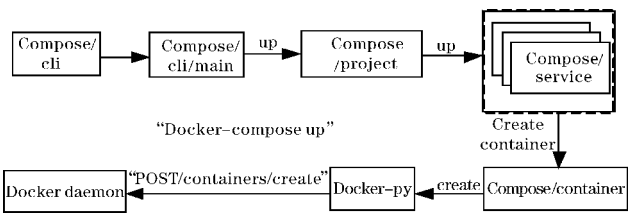

图 1 Docker Compose 的一次调用

图 1 右上角的 Docker Compose 定义了一组“服务”组成一个 Docker Compose 的 project，再通过 service 建立 docker-compose.yml 参数，从而与 container 建立关系，最后使用 container 完成对 docker-py(Docker Client 的 Python 版)的调用，向 Docker Daemon 发起 HTTP 请求[4]。

1.2 Compose 模版文件

Compose 的核心是构建出 docker-compose.yaml(.yml)模版文件，该文件遵循 YAML 的语法。YAML 是一种序列号标准，便于阅读基于 unicode 编码的各种语言，它的用途广泛，多用于模版文件、日志文件、跨语言数据共享、对象持久化以及复杂的数据结构。

收稿日期：2019-12-04
基金项目：国家自然科学基金资助项目(61802319)

YAML 严格区分大小写,使用缩进表示 key 之间的层级关系,如图 2 所示。

```
version: "3.3"
services:
  mysql:
    container_name: mymysql
    image: mysql:5.7
    ports:
      - 12345:3306
    environment:
      MYSQL_ROOT_PASSWORD: *******
    volumes:
      - "./conf/my.cnf:/etc/my.cnf"
      - "./init:/docker-entrypoint-initdb.d/"
    networks:
      - mynet
networks:
    mynet:
```

图 2 docker-compose. yaml 模版文件

docker-compose. yaml 模版文件的主要命令如表 1 所示。

表 1 docker-compose. yaml 模版文件的主要命令

命 令	功 能
version	定义 Compose 文件格式的版本
service	定义不同的应用服务
networks	所加入的网络
image	指定为镜像名称或镜像 ID
links	链接到其他服务中的容器
environment	设置环境变量
volumes	数据卷所挂载的路径设置
command	覆盖容器启动后的默认执行命令
ports	暴露端口信息

2 总体设计

设计和实现了一种面向 Docker Compose 的多容器构建管理工具,便于持续地集成和快速部署。

2.1 总体方案设计

采用 B/S 的三层体系结构,包括数据层、业务逻辑层和表现层,这样有利于系统的开发、维护、部署和扩展[5]。系统采用 SpringMVC+Spring+Mybatis(SSM)框架设计实现,凭借良好的性能、较快的开发效率和轻量级的配置,逐渐成为主流的 Web 应用开发框架组合[6]。并且该框架将注解开发发挥到极致,ORM 实现更加灵活,层次的独立性、可扩展性能满足工具的性能需求。SpringMVC 是使用 MVC 设计思想的轻量级 Web 框架,对 Web 层进行解耦,使开发更简洁,且与 Spring 无缝衔接;Spring 将对象之间的依赖关系交给 Spring 控制,方便解耦,简化了开发;Mybatis 将数据库的操作(sql)采用 xml 文件配置,解除了 sql 和代码的耦合。SpringMVC、Spring 和 Mybatis 分别应用于工具的表示层、业务逻辑层和数据持久层[7-8]。图 3 描述了工具的整体架构。

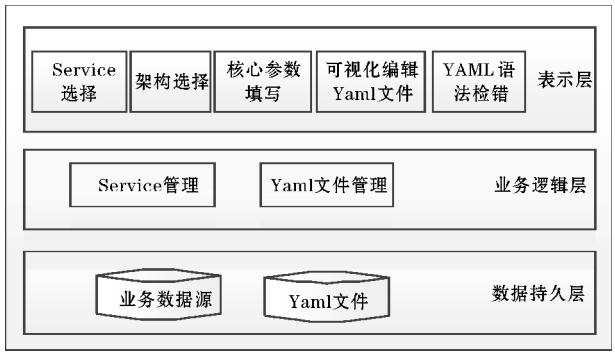

图 3 工具整体架构

表示层包含可视化编辑编辑 YAML 文件、YAML 语法纠错、服务选择、项目架构选择以及关键参数填写等 5 个功能模块。

业务逻辑层主要为服务管理和 YAML 文件管理。

数据持久层为业务逻辑层提供数据,其中业务数据源中存储关系型数据,YAML 文件存储涉及的服务配置信息。

2.2 功能设计

根据实际的应用需求以及操作流程,且考虑到项目间存在架构的差异[9],对传统架构以及分布式架构进行区分,针对性地填写核心参数,从而生成模版文件。详细的功能模块如图 4 所示。

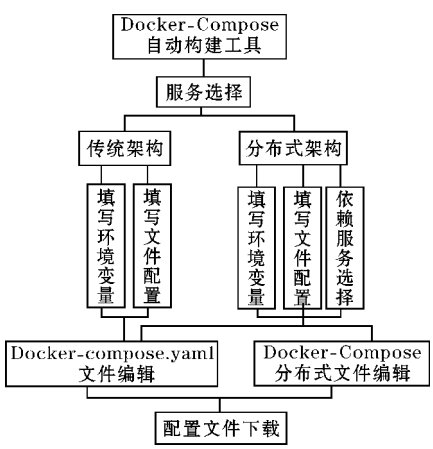

图 4 总体功能模块图

服务选择模块:以列表的形式对服务进行展示,对服务可进行版本修改、添加、删除功能。

架构选择模块:用单选控件实现即将部署项目架构的选择,有传统架构、分布式架构两个选项。

参数填写模块:根据前两个模块的选择结果,跳转到不同的参数填写页面,对参数进行填写选择。

在线编辑模块:实现 YAML 文件的在线编辑以及纠错功能。

文件下载模块:可对编写好的模版文件进行下载保存,并使用该文件直接进行部署。

3　系统功能的实现与测试

3.1　主要功能模块的实现

3.1.1　服务选择模块

界面通过控件展示数据库中的默认服务选项,可对服务添加、删除以及对版本修改,提供的默认服务是当前项目比较常用的服务,如:redis、mongo、nginx 等。对于模块的安全性,针对首次打开系统的用户,为了防止用户进行不规范操作,通过对 HTTP 请求进行解析,并采用 session 判断用户是否对服务进行选择。若 session 不存在服务选择结果,则进行强制跳转到服务选择界面。

为防止数据出现乱码,对数据均进行过滤,统一采用 UTF-8 的编码方式。对于模块的可扩展性,使用设计模式中的策略模式与工厂模式,对服务操作进行单独封装,从而实现了开闭原则,对修改关闭,对扩展开放。

3.1.2　架构选择与详细参数填写模块

系统采用单选按钮控件,实现传统架构与分布式架构的区分。根据架构与服务的选择,有不同的详细信息填写页面。通过输入框实现输入项目名、镜像、nginx 模版文件、sql 密码等必要配置内容。在分布式架构中需要选择每个模块依赖的服务,当点击“依赖服务选择”按钮时,自动弹出服务弹窗,针对模块涉及的服务进行选择,选择完成后数据通过 Ajax 以数组的形式实时传递到对应的 controller 进行解析,将解析处理后的数据,重新写入数据库中的 YAML 文件[10]。

对 YAML 文件的处理采用导入 snakeyaml 包的形式,调用 YAML 对象中的 load 方法,将 YAML 文件中的数据,解析成 map 的格式进行处理。

针对不同的框架和不同的服务,信息填写页面不同。传统架构信息填写界面如图 5 所示,分布式架构根据模块数,生成具体的信息填写页面,如图 6 所示。

图 5　传统架构参数填写页

图 6　分布式架构信息填写页

3.1.3　在线编辑模块

数据在线编辑的实现,将极大提高本工具的时效性和可用性[11]。数据以 map 的形式进行存储修改,可保证数据在格式上的准确性。以字节流的形式,按行读取处理后的 YAML 文件内容并存入 seesion。在前端,通过 sessionScope 读取 session 的内容编辑模块对模版文件进行查看。在线编辑模块可对 docker-compose. yaml 文件与 docker-compose-app. yaml 文件进行在线编辑。编写时如有错误产生,能对 YAML 文本语法错误进行定位,反馈出存在语法错误的行号,方便使用者进行针对性的修改。其显示如图 7 所示,在“Result”准确显示存在语法错误的行号与列号。

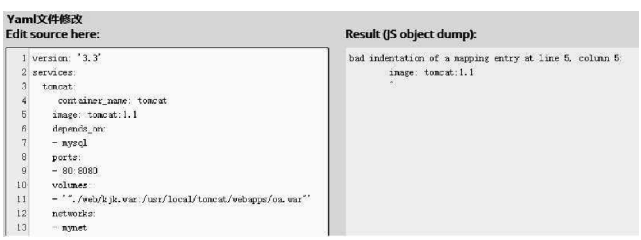

图 7　Docker-compose 模版文件在线编辑

3.1.4　文件下载模块

利用 springMVC 提供的 ResponseEntity 类型实现文件下载。该类将响应头、文件数据(以字节存储)、状态封装在一起交给浏览器处理以实现浏览器的文件下载。可以使用任意类型作为响应体,指明响应状态,并根据不同场景返回不同状态,设置 http 响应头。

3.2　软件测试

执行效率测试方面,对多个涉及不同服务的项目进行部署,记录工具进行编写配置文件时间以及错误率,与手动编写配置文件进行对比,测试结果如表 2 所示,耗费时间对比折线如图 8 所示。

表 2　测试结果

部署项目数	耗费时间(手动)/min	错误率(手动)/%	耗费时间(工具)/min	错误率(工具)/%
3	14.23	1.2	8.34	0
5	18.14	0.4	10.25	0.2
8	27.47	1.1	12.56	0

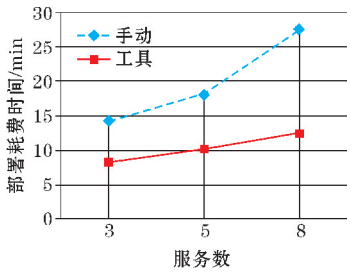

图 8 手动与本工具部署耗费时间对比图

由图 8 可知,工具大幅节省部署时间,在部署的项目涉及的服务越多时效果越明显,执行效率与准确率都有显著的提升。

4 结束语

面向 Docker Compose 的多容器构建管理工具考虑到使用 Compose 进行管理容器时所遇到的困难,提出自动化生成模版文件以及错误检测,对模版文件提供确定的语法格式以及正确的拼写,极大程度上降低了语法层面出错的概率[5]。测试证明构建工具使用方便,容器编排效率提高。

随着云计算的快速发展,Docker 的应用愈发广泛,挑战也随之即来[12]。Docker Compose 作为 Docker 的核心之一,针对不同的业务场景也面临着更复杂的功能需求。今后将进一步完善工具功能,优化性能,为更多种类框架的项目部署提供支持。

参考文献:

[1] 伍阳. 基于 Docker 的虚拟化技术研究[J]. 信息技术,2016(1):121-123.

[2] 高礼,高昕. Docker 技术在软件开发过程中的应用研究[J]. 软件,2016,37(3):110-113.

[3] List M. Using Docker Compose for the Simple Deployment of an Integrated Drug Target Screening Platform[J]. Nephron Clinical Practice,2017,14(2):643-7.

[4] 浙江大学 SEL 实验室. Docker:容器与容器云[M]. 北京:人民邮电出版社,2015:230-234.

[5] 耿朋,陈伟,魏峻. 面向 Dockerfile 的容器镜像构建工具[J]. 计算机系统应用,2016,25(11):14-21.

[6] 李洋. SSM 框架在 Web 应用开发中的设计与实现[J]. 计算机技术与发展,2016,26(12):190-194.

[7] 陈颖慧. 基于 SSM 的英语学习网站的设计与实现[D]. 武汉:华中科技大学,2019.

[8] 曹珍,杨帆. 基于 SSM 框架的商户管理平台设计与实现[J]. 计算技术与自动化,2017,36(4):119-121.

[9] Merkel D. Docker:Lightweight linux containers for consistent development and deployment[J]. Linux Journal,2014,239:2.

[10] 王艳清,陈红. 基于 SSM 框架的智能 web 系统研发设计[J]. 计算机工程与设计,2012,33(12):4751-4757.

[11] 徐震,徐士进,董少春,等. 油田地理信息系统在线编辑功能的设计与实现[J]. 科学技术与工程,2009,9(14):3964-3968.

[12] 王刘飞. Docker 虚拟化安全隔离系统设计与实现[D]. 西安:西安电子科技大学,2018.

Design and Implementation of Multi-container Construction Management Tool by Docker Compose

XU Ke, ZHANG Xinyou, Li Zehui, Li Bin

(The School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China)

Abstract: Docker Compose plays an important role in multi-containers management. However, when using Compose to manage multiple containers, due to its syntax complexity, the strict requirements of yaml (yaml ain't markup language) file on the format and lack of inspection of errors that may exist in the template file, these will lead to the failure of multi-containers service construction. In order to improve work efficiency and reduce subjective errors, it is necessary to design a multi-container build and management tool by Docker Compose. The tool will facilitate the continuous integration and rapid deployment for users. This tool is achieved by using SSM framework, visual editing, error detection and other key technologies. The test results show that the tool is easy to use and the efficiency of container arrangement is improved obviously.

Keywords: SSM framework; visual editing; YAML; Docker Compose