

基于 Kubernetes 的容器云平台研究与设计

张有帅, 余 霞, 尹雪龙

(中汇信息技术(上海)有限公司, 上海 201210)

摘要: 针对有效进行平台运用维护的目的, 采用 Kubernetes 和容器技术, 提出了云平台的设计构想, 旨在提升平台运行灵活度和自动化运维水平, 降低运维成本, 并将容器云平台融入研发流程中。通过在虚拟测试机上部署 Kubernetes 集群、cloud-controller-manager 管理器和存储模块的实验, 实验结果表明, 该平台运行平稳, 基于 Kubernetes 的容器云平台应用部署效率高、应用监控能力得以提升、人员成本下降, 在实际运用中具有可推广性。

关键词: Kubernetes; 容器技术; 云平台; 设计与实现

中图分类号: TN80

文献标识码: A

文章编号: 1674-6236(2021)22-0180-04

DOI: 10.14022/j.issn1674-6236.2021.22.038

Research and design of container cloud platform based on Kubernetes

ZHANG Youshuai, SHE Jia, YIN Xuelong

(CFETS Information Technology(Shanghai) Co., Ltd., Shanghai 201210, China)

Abstract: For the purpose of effective platform operation and maintenance, Kubernetes and container technology are adopted, and the design concept of cloud platform is proposed. It aims to improve platform operation flexibility and automatic operation and maintenance level, reduce operation and maintenance costs, and integrate container cloud platform into it In the research and development process. Through the deployment of Kubernetes clusters, cloud-controller-manager manager and storage modules on the virtual test machine, the test proves that the platform is running smoothly, and it is concluded that the Kubernetes-based container cloud platform has high application deployment efficiency, improved application monitoring capabilities, and reduced personnel costs. It can be promoted in actual application.

Keywords: Kubernetes; container technology; cloud platform; design and implementation

从平台运行维护方面来看,传统的操作手段以人工结合辅助脚本的方法为主,人工巡检的方式容易出错、稳定性较差。在平台运行出现故障时难以及时发现和修复,需要投入大量的人力、物力才能确保平台再次正常运行,上述问题影响了平台服务的深入开展,因此,急需一种提升平台运营灵活性、减少运维投入的先进平台管理方法。文中借助 Kubernetes 等开源工具构建容器云平台,对原有业务系统进行适当改造,以此达到设计目的。

收稿日期: 2020-10-13 稿件编号: 202010032

作者简介: 张有帅(1988—),男,河南南阳人,工程师。研究方向: 系统运维、配置管理、自动化。
-180-

1 容器编排工具概述

1.1 Kubernetes

Kubernetes 开源工具最早应用于 Google 的开源容器集群管理工作,即向企业平台提供了超强的集群管理能力,还能及时发现并修复系统故障,在平台需要升级时还会提供适宜的升级方案,在向平台提供服务时拥有强大的调度能力和弹性伸缩,正是因为 Kubernetes 开源工具于 Google 企业平台运营中的优异表现,使其成为大型企业在构建容器云平台时的主要选择,近年来应用得较为广泛^[1-2]。

1.2 Docker

Docker也是应用较为广泛的应用容器引擎,但是在性能上略逊于Kubernetes,该开源引擎的工作原理是将应用和依赖项共同打包成镜像文件,以起到检索作用,且适配于多款主机,可以将其部署在任何可以确保其正常运行的主机中,可以最大程度地缩减平台开发环境和部署环境之间的差异。工作人员通过访问Docker镜像仓库,可以对镜像和交付模块进行便捷管理。此外,因自身的软件特性,Docker和微服务架构之间存在天然契合度,因此,在构建容器云平台时可将其当作基础软件使用。

2 容器云平台设计

2.1 总体设计

创建好的容器云平台需要安装在物理服务器上,由资源层、平台层和应用层构成,其总体结构如图1所示。

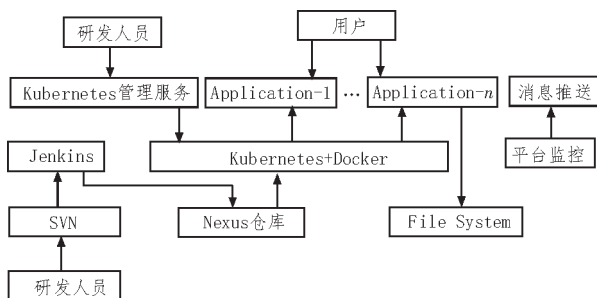


图1 容器云平台总体架构图

资源层位于容器云平台的最底部,其主要工作内容是对容器云平台中的数据资源进行存储和管理,存储资源中涵盖了进行代码版本管理的SVN,以搭建Maven和Docker镜像仓库的Nexus和文件存储服务。平台层中的功能模块都是容器云平台的核心部分,对集群监控程序进行举例,该程序可为平台提供具备执行能力的Docker,用于容器镜像中,提升了Docker和Kubernetes的契合力、统一编排能力和调度能力,并将创建好的容器镜像传送到资源层的Nexus仓库中。应用层是最贴近用户的层,工作内容是开发业务应用和集群控制软件,以供用户选择和使用^[3-4]。

2.2 容器云平台功能设计

基于Kubernetes的容器云平台从两个方面为工作人员提供服务,一个是项目全生命周期支持,另一个则是自动化运营维护支持。项目全生命周期支

持涉及代码管理、外部环境和集中部署等多个方面,这些功能支持从项目启动开始,到部署上线为止,为开发人员提供全生命周期支持。自动化运营维护支持则为工作人员提供自动化工具,例如故障自动修复、监控工具等高效便捷的工具,减轻运营维护人员的工作量,分散工作压力。

2.2.1 项目全生命周期支持

容器云平台中增设项目全生命周期支持功能,可为工作人员提供以下服务:1)代码管理。在容器云平台中将SVN当做代码管理的基础,即可以管理现有代码还可以进行新版本开发,容器云平台研发人员将代码成批量的传送至SVN,系统性的保存起来,为后续研发提供代码支持,代码管理功能也是项目构建的基础。2)持续构建。文中提到的容器云平台选择Jenkins作持续构建工具,借助定时和手动触发技术,定期从SVN中抓取项目源码用来进行持续构建,构建完成后打包发送到镜像仓库中存储起来,方便后续使用。3)环境隔离。容器云平台运用Kubernetes开源工具自身具备的命名空间资源隔离特性为研发人员提供辅助力,包含研发、预发布和外部环境3个环节。3个环节独立运行,彼此隔离部署,研发人员在进行新业务测试时不会影响到已经投入运行的既有服务,更具安全性。4)集中部署。容器云平台为研发人员提供集中部署工具,可确保平台内项目结构趋于统一。还可以减轻操作流程的繁琐度,对研发人员的技术要求有所下降。集中部署工具可以详细的记录用户的操作流程,在构建新平台时可以从用户实际需求出发,能做到有迹可循^[5-6],为潜在的业务升级规避风险,并且提供了修复参考。

2.2.2 自动化运营维护支持

1)平台监控工具。可对平台运营状态、应用状态和服务调配负载状态进行多方位监控,为研发人员提供全面的监控视图。在发现容器云平台出现运行异常时,会进行钉钉自动报警,可以及时地发现故障点,更具时效性,还能大幅度减少人工巡检成本的投入。

2)故障自动修复。容器云平台借助镜像仓库的存储能力,将以往应用过的版本都进行了保存,在升级出现故障或不兼容等情况时,会直接调转至原有的应用以确保运行正常。

3)水平扩容支持。利用容器封装应用所提供的

服务项目,可以通过增设副本的方式对平台进行水平扩容,研发人员可以将负载需求当做参考依据来增加副本数量。

2.3 平台可靠性设计

在容器云平台的运行过程中难免会出现节点失效的问题,如因为内核出现安全漏洞,相应的基础软件也会出现缺陷或BUG,还会出现不易被工作人员发现的潜在故障和阶段性的停机维护,这些故障都需要通过重启宿主机来解决^[7-8]。因此基于Kubernetes的容器云平台在构建过程中应通过相关技术方法降低节点重启所产生的影响,确保其具备高可靠性,对应用到的部分关键技术方法进行阐述和说明如下。

2.3.1 安全性高的文件、数据存储系统

容器云平台会对重要文件进行存储,应用的是CEPH分布式系统,通过使用安全性高的文件、数据存储系统来提升文件存储的可靠性和扩展性。在创建数据库集群时选择主备方式对用户数据进行管理,同时融入了ETCD集群作为辅助功能软件,进一步提升平台运行的可靠性和高效性,提供一致性的管理服务。

2.3.2 Kubernetes集群的高可用部署

基于Kubernetes的容器云平台中包含的用户应用在运行时有限定条件,必须在Kubernetes环境下运行,所以Kubernetes开源工具也是容器云平台的核心组成部分,因此,对其是否具备可靠性也有着严格的要求。出于对Kubernetes中Master和ETCD存储组建重要度的考虑,在进行云平台框架部署时,借助Active-Active的Master多节点同时运行模式和ETCD的服务集群功能来提升Kubernetes集群的高可用部署和运行稳定性,通过运用多种技术,Kubernetes集群的稳定性得到了大幅度提升,只有出现全部Master节点同步故障的状况下,容器云平台的运行才会受到影响,单个或多个节点宕机的情况下并不会影响集群正常运行^[9-10]。其具体结构如图2所示。

2.3.3 应用的高可用部署

平台用户所使用的应用必须以容器的形式在Kubernetes开源工具中运行,Kubernetes借助Replication Controller/Replica Set技术软件来提升应用副本的冗余性。用户可以将部署应用通过打包的形式变成容器镜像,通过Deployment的方式运行在

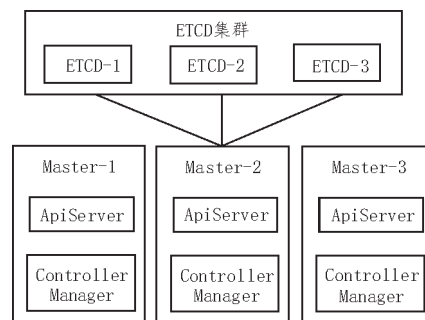


图2 Kubernetes高可用集群结构图

Kubernetes中,并通过设置健康检测探针和副本数量的方式来提升部署应用运行的可靠性^[11-12]。在发生节点故障或程序崩溃时,Kubernetes可自行重启或创建全新的应用副本,确保副本数量满足用户设置的需求。

2.3.4 全面监控和自动报警

平台运行时,容器云平台会从系统、应用和平台3个方面对其进行全面监控,以此来判断其是否处于正常运行状态。1)系统层面。将Zabbix监控软件安装在全部运行机器上,实时采集和报备系统运行的各项指标,发现故障时会立即通过Zabbix触发器发出报警信号,以提醒工作人员。2)平台层面。借助Kubernetes的管理监控服务功能,可调用ApiServer定时查询重要指标,包括节点运行情况、网络传输情况等。在出现集群内部组件异常时,可以及时通知容器云平台的工作人员。3)应用层面。借助Kubernetes开源工具健康状态监测针的能力,对应用运行状态进行定期抽查,发现异常时,平台系统会自动重启出现故障的副本并将故障信息上报给运营维护人员。

3 操作流程设计

在平台开发设计过程中,可利用Kubernetes引擎在项目全生命周期及运营维护方面支持的特性来完成平台功能的环境部署、代码管控、应用部署、功能升级与测试、版本发布等流程,具体步骤如下:

1)借助Kubernetes开源软件的命名空间机制将平台现有的资源划分成3个独立运行互不干扰的子空间,分别为开发、预发布和生产环境,将不同环境下的资源进行分离,以此来确保平台拥有安全稳定的生产环境。

2)工作人员将源代码存储于SVN中,在开发过程中调取使用,以此实现合作开发。

3)利用 Jenkins 组件定期抓取存储于 SVN 中的源代码,并根据项目配置参数创建应用镜像,并将其传送至 Nexus 镜像仓库,并设置相应的版本标签。

4)在开发和测试阶段,工作人员根据运行情况对系统程序进行适当的升级和调整。

5)在新版本发布阶段,运营维护人员应该在预发布环境下进行部署,除了生产程序,还要添加一定比例的生产环境,来进行灰度测试。

6)所有测试完成且指标正常后,将全新的版本程序完全部署在生产环境下,实现程序的研发迭代^[13-14]。研发迭代流程如图3所示。

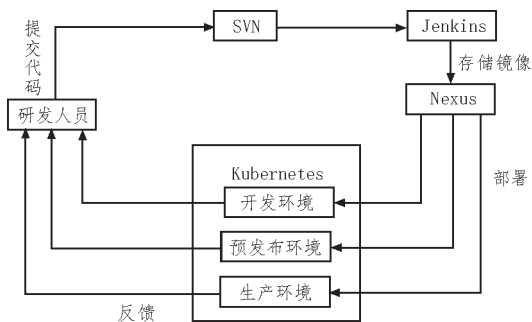


图3 研发迭代流程示意图

4 实验与验证

4.1 实验环境

为了对基于 Kubernetes 容器云平台是否具备可用性和实际功能性进行测试,对实验环境进行了设定,方便观察实验结果,集成测试服务器配置如表1所示。

表1 集成测试服务器配置

配置项目	配置信息
CPU	4 虚拟核心
内存	8 GB
磁盘	高效云盘 100 GB
操作系统	CentOS 7.4 64 位

在虚拟测试机上部署 Kubernetes 集群、cloud-controller-manager 管理器和存储模块,其组件部署分布情况如表2所示。

根据以上测试环境,对容器云平台进行功能性测试,得到以下测试结果。

4.2 实施效果

4.2.1 应用部署效率提高

区别于传统的手动拷贝副本并重启的方式,基于 Kubernetes 的容器云平台更具便利性,只需要在

表2 集群组件部署分布

集群节点身份	节点上部署的组件
Master	kube-apiserver、kube-scheduler、Kube-controller-manager、etcd、cloud-controller-manager、Provision-controller
Node1	Kubelet、kube-proxy、flexvolume-driver
Node2	Kubelet、kube-proxy、flexvolume-driver
Node3	Kubelet、kube-proxy、flexvolume-driver

管理页面输入所需应用镜像和副本数量即可,既降低了部署难度,又提升了成功率和部署效率,将原本需要用3~5天才能完成的升级工作缩短到1天就可以完成,且不影响平台运行的稳定性^[15]。部署效率的提高可以确保后台能够更加快速地响应产品需求,提供最完善的服务效果。

4.2.2 系统监控能力提升

基于 Kubernetes 的容器云平台采用的是多级监控机制,运营维护人员可以通过反馈数据对平台进行全面监控,监控效果可以涵盖每一个节点。监控数据更加丰富,可以使工作人员更快地获取故障信息,并加以修复。

4.2.3 人员成本下降

基于 Kubernetes 容器云平台减少了运营维护人员的日常工作量,通过智能操作代替手动操作,工作人员不需要手动记录部署信息,智能系统的介入既提升了数据记录的准确性,也降低了人员成本^[16]。

5 结束语

伴随着微服务技术的不断完善,基于 Kubernetes 的容器云平台应用得也越发广泛。文中对总体设计和操作流程设计作了具体阐述,并通过仿真实验可知,基于 Kubernetes 的容器云平台的使用可以有效减少运营维护人员的工作量,与此同时还提升了平台运行的可靠性,失败冗杂和容灾恢复等性能也得到了大幅度的提升,可以为企业提供更好的服务,有助于企业的健康稳定发展。

参考文献:

- [1] 赫彭海,徐成龙,刘一田.基于 Kafka 和 Kubernetes 的云平台监控告警系统[J].计算机系统应用,2020(6):121-126.
- [2] 李小孟.基于 Docker 容器技术的云实训平台建设[J].电脑编程技术与维护,2020(3):105-106,115.
- [3] 孟向导.基于 Docker 云平台容器迁移及运维管理(下转第188页)

- development on a humanoid robot[C].IEEE Transactions on Autonomous Mental Development,2014: 93-109.
- [3] Jeannette B,Daniel K,Stefan S.Exemplar-based prediction of global object shape from local shape similarity[C].IEEE International Conference on Robotics and Automation(ICRA),2016:16-21.
- [4] 林立民,刘成菊,马璐,等.基于STFT的仿人机器人手臂轨迹生成方法[J].机器人,2019,41(5):591-600.
- [5] Simon H,Emre U,Sandor S.Reactive task-specific object manipulation by metric reinforcement learning[C].International Conference on Advanced Robotics (ICAR),2015:27-31.
- [6] 李江伟.家庭移动服务机器人三维建模与运动规划研究[D].武汉:华中科技大学,2017.
- [7] Luo D S,Hu F,Deng Y.An infant-inspired model for robot developing its reaching ability[C].Joint IEEE International Conference on Development and Learning and Epigenetic Robotics,2016:19-22.
- [8] 董戈.基于深度学习和图像处理的水果收获机器人抓取系统法[J].农机化研究,2021,43(3):260-264.
- [9] 李秀智,李家豪,张祥银,等.基于深度学习的机器人最优抓取姿态检测方法[J].仪器仪表学报,2020,41(5):108-117.
- [10] 谢宇坤,吴青聪,陈柏,等.基于单目视觉的移动机械臂抓取作业方法研究[J].机电工程,2019,36(1): 71-76.
- [11] 张俊杰.基于RoboCup标准足球机器人的目标识别方法研究[D].南京:南京理工大学,2016.
- [12] 张驰,廖华丽,周军.基于单目视觉的工业机器人智能抓取系统设计[J].机电工程,2018,35(3): 283-287.
- [13] 李航,朱明.基于深度卷积神经网络的小目标检测算法[J].计算机工程与科学,2020,42(4):649-657.
- [14] 张索非,冯焯,吴晓富.基于深度卷积神经网络的目标检测算法进展[J].南京邮电大学学报(自然科学版),2019,39(5):72-80.
- [15] 卢栋才.服务机器人任务理解的关键技术研究[D].合肥:中国科学技术大学,2017.
- [16] Wu R Q,Zhou C L,Chao F.A developmental learning approach of mobile manipulator via playing [J].Frontiers in Neurorobotics,2017,11(11):53-60.
-
- (上接第183页)
- 系统的设计与实现[D].济南:山东大学,2020.
- [4] 韩沈钢.易于Kubernetes的食品处理软件系统设计和开发[D].杭州:浙江工业大学,2020.
- [5] 王伟军.基于Kubernetes的容器云平台建设[J].电脑知识与技术,2019(2):47-48.
- [6] 董昕,郭勇,王杰.基于DevOps能力模型的持续集成方法[J].计算机工程与设计,2018(7):1930-1937.
- [7] 冯文超.容器云平台网络资源配置管理系统的设计[J].工业仪表与自动化装置,2018(5):70-73,104.
- [8] 王骏翔,郭磊.基于Kubernetes和Docker技术的企业级容器云平台解决方案[J].伤害船舶运输科学研究所学报,2018(7):51-57.
- [9] 马席琳.基于Kubernetes容器军魂资源调度策略研究[D].西安:西安科技大学,2019.
- [10] 周佳威.Kubernetes跨集群管理的设计与实现[D].杭州:浙江大学,2017.
- [11] 黄丹池,何震苇,严丽云,等.Kubernetes容器云平台多租户方案研究与设计[J].电信科学,2020(9): 102-111.
- [12] 翁涅元,单杏花,阎志远,等.基于Kubernetes的容器云平台设计与实践[J].铁路计算机应用,2019 (12):49-53.
- [13] 盛乐标,周庆林,游伟倩,等.Kubernetes高可用集群的部署实践[J].电脑知识与技术,2018(26): 40-43.
- [14] 宗序梅,任彦辉.基于Kubernetes的PaaS平台研究与实践[J].江苏通信,2018(2):76-79.
- [15] 刘明,李俊峰.容器化部署Kubernetes及容器内IPv6通信[J].电子技术与软件工程,2019(16):32-33.
- [16] 郑冰.基于Kubernetes的企业级容器云平台设计[J].数字技术与应用,2019(6):138-141.