

赠精品教学
资料包

云计算基础与应用

YUNJISUAN YUNWEI KAIFA

策划编辑：李 勇
责任编辑：陈坤朋
封面设计：刘文东



定价：55.00元

大数据与云计算人才培养系列

云计算运维开发

主编 张丹丹 马 力

北京邮电大学出版社



大数据与云计算人才培养系列
校企“双元”合作新形态教材

云计算运维开发

YUNJISUAN YUNWEI KAIFA

主编 张丹丹 马 力



北京邮电大学出版社
www.buptpress.com

大数据与云计算人才培养系列
校企“双元”合作新形态教材

云计算运维开发

YUNJISUAN YUNWEI KAIFA

主 编	张丹丹	马 力	
副主编	郑 赢	张 娜	赵 薇
	李相泽	白淳予	应天龙



北京邮电大学出版社
www.buptpress.com

内 容 简 介

本书主要讲解了云计算运维开发的相关知识,内容由浅入深并且紧贴实战,初学者能够快速学以致用,有一定基础的读者也能从书中学习到更新的理念和技能。本书共有6个项目,全面涵盖了云计算的基础知识、云计算平台操作与管理、私有云运维、Docker容器技术、系统运维与开发及自动化运维工具部署等知识。

本书以真实项目进行引入,操作步骤详尽,可以帮助读者更好地理解相关知识,真正做到学以致用。每个项目都设置了课后习题,可以加强读者对重点内容的巩固和吸收。

本书既可以作为高职高专院校云计算技术应用及相关专业的教材,也可以作为云运维工程技术人员的自学指导书。

图书在版编目(CIP)数据

云计算运维开发 / 张丹丹, 马力主编. -- 北京:
北京邮电大学出版社, 2024. -- ISBN 978-7-5635-7451

-3

I. TP393.027

中国国家版本馆 CIP 数据核字第 2024SF6867 号

策划编辑: 李 勇 责任编辑: 陈坤朋 封面设计: 刘文东

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路10号

邮政编码: 100876

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 三河市骏杰印刷有限公司

开 本: 787 mm×1 092 mm 1/16

印 张: 17.75

字 数: 367 千字

版 次: 2024 年 12 月第 1 版

印 次: 2024 年 12 月第 1 次印刷

ISBN 978-7-5635-7451-3

定 价: 55.00 元

• 如有印装质量问题,请与北京邮电大学出版社发行部联系 •

服务电话: 400-615-1233



PREFACE

前言

随着数字化时代的来临,云计算作为一种新兴的技术模式,已经广泛渗透到各个领域,并对传统 IT 架构产生了深远的影响。了解云计算的起源、发展历程、现状以及未来趋势,掌握云计算的关键技术,对于我们准确把握这个领域的脉搏非常重要。

IT 行业需要的不仅仅是只懂理论知识的人才,而是那些技术过硬,综合能力强的技能型人才。因此,一本教材如果只罗列理论知识是远远不够的,它必须以真实项目及实操过程为主要内容,将理论知识与真实的岗位能力相结合,全面培养学生的应用能力。

学习本书的原因在于:云计算的发展极为活跃,它不断推动着信息技术领域的进步,并深度融入各行各业的数字化转型中。随着数字化转型加速,中国云计算行业正面临着前所未有的发展机遇。

本书从岗位能力出发,提取典型的工作任务,对云计算运维的关键技术和实际操作步骤进行了详尽的介绍,旨在帮助读者建立云计算技术的知识体系,并掌握云计算运维开发的核心能力。

本书系统且全面地介绍了云计算运维开发的相关知识,全书共分为 6 个项目,每个项目的具体任务内容概述如下。

项目 1 旨在引导读者深入了解云计算的基础知识,包括云计算的发展历程、优势及关键技术等核心内容。通过本项目的学习,读者能够对云计算技术形成初步的认识,为后续的学习奠定坚实的基础。

项目 2 主要介绍云计算技术的实际应用,包括私有云平台、容器云平台、公有云平台,以及 Linux 操作系统的具体配置与管理方法。通过本项目的学习,读者能够掌握云运维开发的基础操作,为后续的具体运维工作营造良好的环境。

项目 3 主要介绍私有云平台运维的关键技术,包括计算资源的管理、存储与网络管理等多个方面。其中,特别强调了 Glance 服务、Nova 服务、Barbican 服务、Cinder 块存储、Swift 对象存储及 VPNaas 网络等关键技术的实际操作内容,这些内容在云计算运维技术中占据着重要地位。

项目 4 主要介绍 Docker 容器技术的具体应用,包括 Docker 的部署与管理、编排、负载均衡、Swarm 高可用及 Rancher 持续集成等主要操作。这些内容是云计算技术运维中的重要组成部分。

项目 5 主要介绍系统的运维与开发,包括 Shell 脚本开发和 Python 脚本开发两部分内容。通过本项目的学习,读者能够掌握自动化运维脚本编写的基本思路与方法,为日后的自动化运维工作打下坚实的基础。

项目 6 主要介绍自动化运维工具的部署,包括部署 Nginx、优化 Ansible 及扩展 Ansible 等内容。通过本项目的学习,读者能够掌握自动化运维中常用的方法和手段,进一步提升把握实际工作任务的能力。

本书由沈阳职业技术学院张丹丹、马力任主编并统稿,由沈阳职业技术学院郑赢、张娜、赵薇、李相泽、白淳予和无锡瑞诺尔安道科技有限公司应天龙任副主编。沈阳职业技术学院陆宜梅也参与全书的案例验证与审阅,在此表示诚挚的感谢!

由于编者水平有限,书中可能存在不足之处,敬请广大读者批评指正。

编 者



CONTENTS

目录

项目 1 云计算的基础知识	1
任务 1 认识云计算	1
子任务 1 了解云计算的基础概念	1
子任务 2 了解云计算的优势	4
子任务 3 认识云计算服务模型	6
任务 2 探索云计算技术的应用	8
子任务 1 了解虚拟化技术	8
子任务 2 了解分布式存储技术	12
子任务 3 了解分布式监控技术	15
项目 2 探索云计算技术的应用	19
任务 1 认识主流云平台	19
子任务 1 了解私有云平台	19
子任务 2 了解容器云平台	22
子任务 3 了解公有云平台	27
任务 2 掌握 Linux 操作系统的配置与管理	30
子任务 1 认识 Linux	30
子任务 2 安装虚拟机	33
子任务 3 配置 Linux 系统	38
项目 3 私有云运维实战	55
任务 1 管理计算资源	55
子任务 1 配置和管理 Glance 服务	55
子任务 2 配置和管理 Nova 服务	63
子任务 3 配置和管理 Barbican 服务	72
任务 2 存储与网络管理	82

子任务 1 Cinder 块存储	82
子任务 2 Swift 对象存储	92
子任务 3 VPNaaS 网络	98

项目 4 应用 Docker 容器技术 104

任务 1 掌握 Docker 基础应用	104
子任务 1 认识 Docker	104
子任务 2 部署与管理 Docker	107
子任务 3 编排 Docker	116
任务 2 掌握 Docker 高级应用	124
子任务 1 掌握负载均衡	124
子任务 2 Swarm 高可用	126
子任务 3 Rancher 持续集成	128

项目 5 熟练掌握系统运维与开发 151

任务 1 应用 Shell 脚本编程	151
子任务 1 了解 Shell 的基础语法	151
子任务 2 了解正则表达式	168
子任务 3 Shell 实战	187
任务 2 应用 Python 脚本编程	192
子任务 1 了解 Python 的基础语法	192
子任务 2 了解 Python 的基本元素	197
子任务 3 Python 实战	205

项目 6 掌握自动化运维工具部署 214

任务 1 实现 Ansible 自动化运维	214
子任务 1 了解 Ansible 的基础知识	214
子任务 2 应用 Ansible-Playbook	227
子任务 3 应用 Ansible 变量	233
任务 2 自动化运维实战	247
子任务 1 部署 Nginx	247
子任务 2 优化 Ansible	260
子任务 3 扩展 Ansible	268

参考文献 277

云计算的基础知识

项目引入

某高校计划部署私有云平台,在进入需求分析阶段之前,需要先就云计算的基本概念进行沟通交流,并整理出相关的云计算基础知识。

任务 1 认识云计算

子任务 1 了解云计算的基础概念

职业能力目标和要求

- 了解云计算的基础概念。
- 了解云计算的发展。
- 掌握云计算的应用方向。

1. 云计算介绍

云计算可以理解为网络上的计算机为用户提供按需付费的服务,并且云计算还具备以下优势。

(1)成本效益高:用户无须在前期大量投资于硬件设备和维护,而是按照实际使用量付费,从而降低了单个用户的成本。

(2)灵活性与多样性:云计算提供了丰富的实例规格和多种服务选项(如 IaaS、PaaS、SaaS),能够满足不同业务场景的需求,支持快速创新和多样化服务的开发。

(3)高可用性与容错性:云计算平台通常提供 99.99% 以上的服务可用性,通过自动故障检测和恢复、数据多副本容错等技术手段,确保服务的连续性,有效减少业务中断的风险。

2. 云计算发展

1) 发展背景

在 20 世纪 50 年代末至 60 年代,McCarthy 提出了“效用计算”的概念,主张计算能力应像公共事业一样提供给用户;而 Strachey 则在论文中初步阐述了虚拟化的概念,这为现代云计算理念的形成奠定了基础。20 世纪 80 年代的网格计算、90 年代的公用计算,以及 21 世纪初的虚拟化技术、服务导向架构(SOA)和软件即服务(SaaS)等技术的发展,为云计算的诞生奠定了坚实的技术基础。进入 21 世纪初期,随着 Web 2.0 的兴起,网站和业务系统需要处理的数据量急剧增加,这促使企业寻求能够快速扩展且成本效益高的 IT 解决方案,从而推动了云计算的快速发展。

云计算的发展是长期技术积累与市场需求的共同作用,它融合了计算机科学、网络技术、经济学等多个领域的成果,最终形成了人们现在所熟知的云计算模式。

2) 云计算技术发展

随着数字化时代的到来,云计算作为一种新兴的技术模式,已经广泛渗透到各个领域,并对传统 IT 架构产生了深远的影响。深入了解云计算的起源、发展历程、当前现状以及未来趋势,对于我们准确把握这个领域的动态非常重要。

云计算的起源可追溯到 20 世纪 60 年代,当时计算机科学家 John McCarthy 首次提出了“资源共享”的概念,他认为计算资源能够通过网络实现共享。然而,受限于当时的技术条件,这一理念并未得到迅速发展。直至 20 世纪 90 年代,随着互联网的普及和信息量的迅猛增长,这一概念重新引起了人们的广泛关注。

云计算的发展可以分为以下几个阶段。

萌芽期:这个阶段主要聚焦于云计算理念与技术的初步探索,并伴随着一些基础设施的初步建立。

发展期:在这个阶段,云计算开始获得实际应用,众多云计算服务提供商应运而生,如 Amazon、Google 等。

成熟期:云计算技术在此阶段逐渐趋于完善,其应用领域也日益拓宽。在这个阶段,安全性、稳定性与灵活性成为推动云计算进一步发展的关键因素。

云计算发展的时间轴如图 1-1 所示。



图 1-1 云计算发展的时间轴

云计算的关键技术包括虚拟化、分布式计算、存储即服务和弹性伸缩等,这些技术能够显著提升计算和存储资源的利用效率,确保数据的高可用性和良好的可扩展性。基础设施包括物理硬件、网络设施、操作系统和中间件等,它们共同构成了云计算的基础架构。

时至今日,混合云和多云策略的应用已成为企业追求的新趋势。面对不同业务场景下对安全性、可靠性和灵活性的多样化需求,企业开始寻求将私有云、公有云以及本地数据中

心资源进行有效整合的解决方案,以实现资源的最优配置和管理。同时,随着物联网(IoT)和5G技术的飞速发展,边缘计算作为一种新兴技术与云计算相结合,满足了实时数据处理和分析需求日益增长的场景,从而开启了云计算技术发展的新篇章。

3. 云计算的应用方向

1) 商业应用

云计算经历了从网格计算、效用计算、自主计算到最终形态的云计算的演变过程,逐渐发展成为一种重要的商业计算模型。随着企业的不断发展,云计算企业级应用市场规模持续扩大,同时,人工智能、大数据等技术的融合应用也将为企业级应用市场带来新的增长点。云计算在商业应用领域的应用范围十分广泛。例如,支付宝、微信支付等利用云计算技术实现了高效、安全的在线支付服务;一些制造企业借助云计算技术实现了生产过程的数字化和智能化,提升了生产效率和产品质量;一些医疗平台则利用云计算技术推出了在线问诊、健康数据监测等服务,极大地方便了患者就医和医生诊断。

2) 社会经济应用

经过近年来各方的共同努力,我国云计算产业取得了快速发展,已成为推动经济增长、加速产业转型的重要力量。各地相继出台了一系列的政策和措施,旨在培育信息产业新业态,确立了云计算在我国信息化建设中的重要地位,将其作为制造强国、网络强国重要支撑的战略角色。云计算的发展带来了以下贡献。

(1)经济增长点:云计算的迅猛发展带动了上下游产业的蓬勃兴起,成为新的经济增长点,催生了产业发展的新业态、新模式。

(2)关键技术取得突破:目前,我国云计算领军企业的单机服务器规模已达数万台,数据处理能力能够达到拍字节(PB)级别,支撑金融支付服务等业务的峰值处理能力达到世界领先水平。

(3)行业深化,带动产业升级:云计算在各个领域得到广泛应用,通过与互联网的深度融合,有力推动了“互联网+”行动的实施,成为互联网基础设施的重要组成部分。

(4)降低创业门槛:目前,云计算已成为我国创新创业的重要技术平台,吸引了大量新应用、新业态、新模式的开发团队,显著带动了就业,产生了显著的社会效益和经济效益。

3) 个人生活应用

云计算对个人生活带来的影响主要体现在以下几个方面。

(1)购物。借助云计算服务,用户能够随时随地进行网购,无须顾虑存储空间和计算能力的问题。同时,云计算也为商家提供了更为高效的物流和库存管理方案。

(2)出行。云计算能够为用户提供更加便捷的出行服务,如共享单车、网约车等。这些服务依托云计算平台,能够实时监测车辆位置、可用性和交通情况等信息。

(3)娱乐。在娱乐领域,云计算为音乐、视频、游戏等应用提供了更为强大的计算和存储支持。用户可以通过云服务提供商的应用程序接口(API),享受到更加流畅的视听体验。

(4)学习。云计算为用户提供了更加丰富的在线学习资源,如在线课程、电子书、视频教程等。用户只需通过云服务访问在线学习平台,即可随时随地学习自己感兴趣的知识。

总而言之,云计算是继20世纪80年代大型计算机到客户端-服务器模式的大转变之后的又一次巨大变革。它是分布式计算、并行计算、效用计算、网络存储、虚拟化、负载均衡等传统计算机和网络技术融合发展的产物。

子任务2 了解云计算的优势

职业能力目标和要求

- 了解弹性和可扩展性。
- 了解虚拟化的基础概念。
- 掌握高可用性的定义。

1. 高可用性

1) 高可用性的定义

高可用性(high availability, HA)是指一个系统或组件在设计时采取了专门的措施,以确保在预定的运行时间内尽可能减少中断,维持服务的持续可用状态。这意味着,即便系统遇到硬件故障、软件错误、网络问题或其他潜在的中断因素,仍能保持服务的提供,从而最大限度地降低对用户的影响。高可用性不仅着眼于减少停机时间,还包括故障发生时的快速恢复服务能力,以及在不影响用户体验的前提下进行维护和升级的处理能力。

2) 高可用性的关键技术策略

高可用性的关键技术策略如下。

(1)资源冗余:当某个核心组件发生故障时,其他位置的资源能够接管服务,这包括计算实例、存储和网络资源的冗余配置。

(2)负载均衡:通过负载均衡器将流量分配到多个实例上,以避免单一节点过载,确保服务请求能够得到均匀处理,从而提升系统的响应速度和可用性。

(3)自动故障检测与恢复:利用监控工具实时监测系统状态,一旦检测到故障,自动触发恢复流程,如自动重启实例、切换到备用服务等。

(4)可扩展性:通过水平扩展(增加实例数量)或垂直扩展(增强单个实例的能力)来应对突发流量,确保系统在高负载下仍能维持高性能。

(5)数据备份与恢复:定期备份数据,并确保备份数据分布在不同的地理位置,以便在数据丢失或损坏时能够迅速恢复。

(6)异地多活:在多个地理区域部署服务,每个区域都能独立处理用户请求,即使某个区域出现故障,用户服务也不会受到影响。

(7)灾难恢复计划:制订详细的灾难恢复(DR)计划,包括定期的灾难恢复演练,以确保在极端情况下能快速恢复业务。

(8)安全措施:实施严格的安全措施以保护云资源免受攻击或数据泄露,如防火墙、入侵检测系统、数据加密等。

(9)运用高可用性的技术策略可以提供高度可靠的服务,显著降低因技术故障导致的服务中断风险。

3) 高可用性和灾难恢复

一个好的高可用性架构会涵盖业务连续性规划中的各个层面,包括监控和自动化等关键要素。这样的架构能够有效应对各种类型的故障,无论是特定的局部故障还是整体停机事件。高可用性设计甚至能确保系统在计划维护窗口和其他服务中断期间仍能持续运行。灾难恢复或连续计划会针对每种潜在故障制定相应的应对策略。

(1)预测并应对特定故障:对于每个潜在的故障领域,IT 架构首先会确保系统具备冗余性,以便在出现故障时能够启用备份系统。接下来,系统会实现自动化的故障检测和转移流程,以便快速识别故障系统,并将服务无缝切换至备份系统。

(2)主动管理性能:虽然容错机制能够有效解决中断问题,但不一定能处理性能下降的问题。此时,可以借助负载均衡和可扩展性工具来应对。在这种情况下,IT 架构会密切监测系统性能,并部署多个系统来管理用户请求和操作。负载均衡器和流量管理能够实时根据带宽、系统性能、用户类型或请求类型智能地路由流量。

(3)应对灾难性事件:尽管云供应商瘫痪或数据中心遭遇自然灾害等大规模基础架构故障较为罕见,但我们必须为此类故障做好充分准备。应对这类故障需要比单纯处理硬件/软件故障更为全面的策略。不仅要确保基础架构能够迅速恢复运行,还必须提供最新的数据。为此,我们可以采取同步复制数据(但需注意可能带来的性能风险)或异步备份数据(但有丢失数据的风险)的方法。

云计算中的高可用性是指在云计算环境中,借助一系列技术和架构设计,确保服务和应用程序在遭遇各种故障时仍能持续稳定运行的能力。高可用性是评估系统稳定性和服务质量的重要指标之一。设计高可用性系统是一个涉及多方面的系统工程,涵盖了从硬件到软件、从架构设计到运维管理的各个环节。通过综合运用上述策略和技术,可以显著提升系统的稳定性和服务质量,从而确保业务的连续性。随着技术的不断发展,新的工具和方法不断涌现,因此,持续关注并应用这些最佳实践是实现高可用性的关键。

2. 按需服务

计算按需服务是云计算服务的一种交付模型,它允许用户根据实时的业务需求,动态地获取和释放计算资源,包括但不限于计算能力、存储空间、网络带宽以及软件应用等资源,而无须预先投资硬件设施或签订长期合同。这种模式极大地提高了资源利用效率,降低了运营成本,并为企业和服务提供商带来了前所未有的灵活性与可扩展性。在这种模式下,资源的获取变得如同获取水电一样便捷,用户只需根据实际使用的资源量支付费用即可。云计算按需服务的一些关键特点和优势如下。

(1)即时获取资源:用户能够通过自我服务门户或 API 即时访问计算能力、存储、数据库等各种云服务,无需等待硬件的采购和部署。

(2)弹性资源扩展:根据业务负载变化,用户可以迅速调整资源规模,既能在需求高峰时快速扩大资源,又能在非高峰时段缩小资源以降低成本。

(3)成本效益显著:按需付费模式有效避免了前期的大额投资,用户仅需根据实际消耗的资源付费,这有助于精确控制成本并优化预算分配。

(4)多样化服务选项:云服务商提供了从基础设施即服务(IaaS)、平台即服务(PaaS)到软件即服务(SaaS)的全方位解决方案,以满足不同企业的多样化业务需求。

(5)运维负担减轻:云服务商负责底层基础设施的日常维护和更新工作,这极大地减轻了用户的运维压力,使用户更专注于自身的核心业务。

(6)全球访问能力:借助云服务商遍布全球的数据中心网络,用户可以从任何地点轻松访问所需的云资源和服务,这极大地促进了跨国业务的开展。

(7)安全与合规保障:云平台通常配备了高级的安全防护措施,并严格遵守多种行业合规标准,为用户提供全方位的安全保障。

(8)环境一致性:云计算平台支持标准化的开发和部署环境,这确保了应用在不同阶段(开发、测试、生产)之间能够保持高度的一致性。

3. 虚拟化

云计算中的虚拟化技术是“基础设施即服务”的重要组成部分。IaaS 允许用户根据需要快速部署、销毁和弹性伸缩的计算、存储和网络资源。IaaS 云平台能够根据业务需求随时扩展、收缩和迁移云资源,从而降低成本并提升整体性能。为了实现 IaaS 云平台的功能,云厂商通常需要提供底层的虚拟化技术支持。云计算虚拟化技术通过利用虚拟化技术,实现了云平台的资源池化、调度和分配,以及业务之间的隔离和保护。主要的云计算虚拟化技术如下。

(1)服务器虚拟化:这是最为基础且广为人知的虚拟化形式。它允许多个虚拟机(VMS)在同一物理服务器上并行运行,每个虚拟机都能独立运行不同的操作系统和应用程序,且彼此之间实现完全隔离。

(2)容器化:以 Docker 为代表的容器技术,提供了一种更为轻量级的虚拟化方案。通过将应用程序及其所有依赖项封装进一个可移植的容器中,容器化技术实现了应用的快速启动和更高的资源利用效率。与虚拟机相比,容器更为轻便,因为它们共享主机操作系统。

(3)网络虚拟化:将物理网络资源(如交换机、路由器)抽象为虚拟网络设备和网络服务,便于灵活配置、管理和隔离网络资源,支持动态网络拓扑和策略管理。

(4)存储虚拟化:将物理存储介质(如硬盘、磁盘阵列)聚合为一个或多个逻辑存储池,提供统一的存储管理,支持动态分配、迁移和回收存储资源,提高存储利用率。

(5)桌面虚拟化:将用户的桌面环境(如操作系统、应用和数据)封装在服务器的虚拟机里面,用户可通过客户端或任何支持远程桌面协议的设备访问自己的个性化桌面。

(6)应用虚拟化:将应用程序与底层操作系统分离,使应用能够在不受底层操作系统限制的情况下运行,简化了应用的部署和升级过程。

云计算虚拟化技术支撑了云计算的灵活性、可扩展性和高可用性,使得云服务提供商能够快速响应用户需求,实现动态分配资源,并提供稳定可靠的服务。通过虚拟化,企业能够更高效地利用 IT 资源,降低成本,同时加速业务创新。

子任务 3 认识云计算服务模型

职业能力目标和要求

- 掌握云计算提供的不同服务层次,包括基础设施、平台和软件即服务形式。
- 了解云计算服务模型三大类型各自的特点和应用场景。
- 了解基础设施、平台和软件即服务之间的区别。

1. IaaS(基础设施即服务)

IaaS 是云计算的一个基本服务模型,它提供虚拟化的计算资源,包括虚拟服务器、存储和网络。在 IaaS 模型中,用户可以租用虚拟机、存储空间和网络资源,而无须关注底层的硬件和基础架构细节。用户可以根据实际需求进行资源的弹性扩展和收缩,并享有更高的控

制权来配置操作系统、网络设置等。IaaS 的特点如下。

(1)提供虚拟化的计算、存储和网络资源,为用户带来了前所未有的灵活性和便利性。用户可以迅速根据实际需求获取所需资源,并根据业务变化进行动态调整。这种高度的灵活性使得 IaaS 成为众多企业和组织选择云计算服务的重要原因之一。

(2)用户可以自定义配置操作系统、应用程序和中间件,这一特性使得 IaaS 成为那些希望保持技术灵活性和控制权的企业或个人的理想选择。通过自定义配置,用户能够确保自己的 IT 环境精确匹配业务需求和技术栈要求,从而实现更高效、更可靠的业务运营。

(3)弹性扩展和收缩资源,按需付费的特点,使得 IaaS 成为一个高效、灵活且成本效益显著的解决方案。它赋予用户极大的灵活性和控制权,帮助他们更有效地管理 IT 资源,满足业务需求,并降低总体成本。无论是初创企业还是大型企业,都可以通过采用 IaaS 服务来优化其 IT 基础设施,提高业务效率和竞争力。

2. PaaS(平台即服务)

PaaS 是一种更高级别的云服务模型,它在 IaaS 的基础上,为开发者提供了更为丰富的服务层次。PaaS 为开发者提供了一个集开发、部署应用程序于一体的平台,该平台包括操作系统、中间件、开发工具、数据库以及运行时环境等关键组件。开发者可以专注于应用程序的开发工作,而无须关注底层的基础架构。PaaS 的特点如下。

(1)通过提供应用程序开发和部署的平台,包括开发工具和运行时环境等特性,使得 PaaS 成了一个高效、便捷且易于管理的应用程序开发和部署解决方案。它极大地简化了开发者的工作流程,提高了开发效率,优化了应用程序的质量,使得开发者能够更快速地响应市场需求和业务变化。

(2)开发者可以专注于应用程序的开发,而不需要关注底层的操作系统和中间件的特点,使得 PaaS 平台成了一个高效、便捷且易于管理的应用程序开发解决方案。它让开发者能够更加专注于业务逻辑和用户体验,提高开发效率和质量,从而更快速地响应市场需求和业务变化。

(3)提供了自动化的资源管理和扩展的特点使得 PaaS 服务成了一个高效、灵活且易于管理的 IT 资源解决方案。它让开发者能够更专注于应用程序的开发和优化,而无需担心底层资源的配置和管理。同时,它也降低了运维人员的工作负担,提高了系统的可扩展性和可用性。

3. SaaS(软件即服务)

SaaS 是一种最高级别的云服务模型,它提供了完整的应用程序作为服务。用户只需通过网络即可直接访问应用程序,无须自行安装、维护或管理应用程序所需的基础设施。SaaS 模型通常以订阅方式提供,用户按照使用量或订阅时长来付费。SaaS 的特点如下。

(1)提供完整的应用程序作为服务:用户通过浏览器即可访问应用程序,这种方式不仅简化了操作流程,还实现了跨平台、跨设备的无缝使用体验。用户只需打开浏览器,输入指定网址,即可随时随地访问并管理应用程序。这种服务模式不仅提高了用户的使用效率,还为用户提供了更加灵活和便捷的工作方式,满足了现代工作环境中对高效、便捷应用程序的需求。

(2)彻底免除了用户安装、维护或管理应用程序基础设施的烦恼。用户无须担心硬件设

备的购置、软件系统的安装、更新和升级,以及服务器的配置和管理等复杂任务。服务商已准备好了所有必要的基础设施,并且负责其日常的维护和管理,确保应用程序的稳定运行和高效性能。用户只需专注于使用应用程序,即可享受其带来的便利和价值,无须分心于烦琐的技术细节。这样的服务模式极大地降低了用户的使用门槛,使更多人能够轻松上手并充分利用应用程序的功能。

(3)采用订阅模式付费:用户可根据自身需求和预算选择合适的订阅计划。这种付费方式意味着用户无须一次性支付高额费用,而是根据使用期限和服务内容来支付相应费用。订阅模式不仅为用户提供了更加经济实惠的选择,还让用户能够按需付费,避免了资源浪费。用户可根据业务规模、使用频率等因素灵活选择合适的订阅计划,调整费用支出,实现成本效益最大化。这种灵活的付费方式为用户提供了更加便捷和灵活的服务体验,让用户能够更加轻松地享受服务商所提供的服务。

任务2 探索云计算技术的应用

子任务1 了解虚拟化技术

职业能力目标和要求

- 了解计算机实体资源及虚拟化技术。
- 了解虚拟化技术分类、虚拟化各个技术的特征及优势。
- 掌握云计算在技术层面上的关键技术。

1. 虚拟化介绍

1) 虚拟化技术的基础知识

(1)虚拟化的定义。在计算机领域,虚拟化指的是创建某事物的虚拟(而非物理)版本,这包括虚拟的计算机硬件平台、存储设备,以及计算机网络资源等。可见,虚拟化是一种资源管理技术,它将计算机的各种实体资源(如 CPU、内存、存储、网络等)进行抽象和转化,实现资源的分割与重新组合,以达到最大化利用物理资源的目的。

(2)虚拟化的概念。虚拟化是指计算元件在虚拟而非真实的基础上运行,通过“虚”的软件来代替或模拟“实”的服务器、CPU、网络等硬件产品,从而提供一种简化管理、优化资源的解决方案。虚拟化技术将物理资源转变为可管理性的逻辑资源,打破了物理结构之间的隔离,将物理资源整合为一个统一的整体。虚拟机则是通过软件模拟的、具有完整硬件系统的计算机,从理论层面看,它与实体物理计算机在功能上完全等同。

(3)虚拟化的三层含义。

- ①虚拟化的对象包括各种各样的资源。
- ②经过虚拟化处理后,逻辑资源向用户隐藏了不必要的细节。
- ③在虚拟环境中,用户可以实现其在真实环境中的部分或全部功能。
- (4)虚拟化的关键特征。

- ①分区:在单一物理服务器上能够同时运行多个虚拟机。

- ②隔离:每一个虚拟机都与其他位于同一服务器上的虚拟机相隔离。
- ③封装:虚拟机将整个系统(包括硬件配置、操作系统及应用等)封装在文件里。
- ④独立性:可以在其他兼容的机器上不加修改地使用虚拟机。

2) 虚拟化技术的作用

虚拟化技术通过将物理服务器分割成多个虚拟服务器,使每个虚拟服务器都能运行相互隔离的操作系统和应用程序,从而大大提高了资源的利用率,降低了硬件成本,并使资源的分配和管理变得更加灵活。在虚拟化环境中,虚拟机可以快速迁移到其他物理服务器,这对于灾难恢复和确保业务连续性至关重要,有助于提高系统的安全性并减少潜在的系统间冲突。

简单来说,虚拟化技术能将物理资源虚拟化。在传统架构当中,一个物理主机通常只能支持一个操作系统及其上层应用;而采用虚拟化技术后,一个物理主机可以被分割成多个虚拟主机,每个虚拟主机都能支持一个或多个操作系统及其应用程序。

3) 虚拟化管理工具

虚拟化管理工具是一种与虚拟化环境及其背后的实体硬件进行对接的软件,它的主要作用是简化资源管理、数据分析及运维流程。尽管每个虚拟化管理系统都各有特色,但大多数系统都会提供直观的用户界面,用于简化虚拟机(VM)的创建流程、监控虚拟环境、分配资源、生成报告,以及自动执行管理规定。此外,一些解决方案还集成了多个硬件和软件品牌的功能,使用户能够选择并安装最适合他们需求的管理系统。

在当今这个高度数字化的世界中,计算机资源的管理变得尤为重要。虚拟化作为一种先进的资源管理技术,为应对这一挑战提供了有效的解决方案。它通过将计算机的各种实体资源(如 CPU、内存、存储和网络)进行抽象化并重新组合,实现了物理资源的高效利用。在这个过程中,虚拟机监控器(virtual machine monitor, VMM)发挥了至关重要的作用。

虚拟机监控器也被称为 Hypervisor,它是为了实现虚拟化而引入的软件层。VMM 可以被视为一个专为虚拟化环境而设计的操作系统核心。它全面掌控着所有资源,包括 CPU、内存和 I/O 设备,并负责管理这些资源。VMM 向下能够感知并控制实际的各种物理资源,向上则可以为虚拟机划分逻辑资源,从而营造出虚拟机独享物理资源的假象。

2. 虚拟化的分类

1) 虚拟化执行层面分类

从实现方式上可以分为以下两类。

(1)硬件虚拟化:硬件虚拟化技术是指计算机硬件本身具备的能力,它允许客户机指令独立执行,而无须 VMM 截获并重定向这些指令,如 Intel VT 技术。

(2)软件虚拟化:通过软件模拟来构建 VMM 层,利用纯软件环境来模拟执行客户机中的指令,如 QEMU。

对软件虚拟化进行细分,可以进一步分为以下几类。

(1)完全虚拟化。在这种虚拟化方式中,虚拟机完整地模拟底层硬件环境和特权指令的执行过程,从而使得客户机操作系统能够独立于宿主操作系统运行。支持完全虚拟化的软件有:Parallels Workstation、VirtualBox、Virtual Iron、Oracle VM、Virtual PC、Virtual Server、Hyper-V、VMware Workstation 及 QEMU 等。

(2)部分虚拟化。与完全虚拟化不同,部分虚拟化仅针对部分硬件资源进行虚拟化,虚拟机仅模拟部分底层硬件环境,特别是地址空间。这样的虚拟化环境支持资源共享和线程独立,但通常不允许客户机操作系统以完全独立的方式运行,即不支持建立完全独立的客户机操作系统实例。

(3)平行虚拟化。在这种虚拟化方式中,虚拟机并不需要完全模拟硬件,而是将部分硬件接口以软件的形式提供给客户机操作系统。如早期的 Xen 就采用了这种技术。

(4)操作系统层虚拟化。这种技术通过对操作系统内核进行虚拟化,使得用户空间软件实例可以被分割成多个独立的单元,在内核中并行运行,而非仅有一个单一实例运行。这些软件实例通常被称为容器(containers)、虚拟化引擎(virtualization engine)或虚拟专用服务器(virtual private servers)。每个容器内的进程是相互独立的,对于用户而言,仿佛是在使用各自独立的专用服务器。Docker 容器技术便是操作系统层虚拟化技术的一个典型代表。

2)虚拟化技术层面的分类

虚拟化是云计算的入门技术,从云计算资源的角度来看,虚拟化可以分为计算虚拟化、存储虚拟化和网络虚拟化。

(1)计算虚拟化。

①计算虚拟化主要包括 CPU 虚拟化、内存虚拟化和 I/O 虚拟化。

CPU 虚拟化:确保虚拟机上的指令能正常执行,并且执行效率接近物理计算机的水平。

内存虚拟化:通过为虚拟机提供内存空间隔离,使其认为自己拥有独立的内存地址空间。

I/O 虚拟化:使得虚拟机能够访问所需的 I/O 资源,并且这些资源在虚拟机之间是隔离的。

②计算虚拟化的优势。在虚拟化之前,每台主机通常只能运行一个操作系统,软硬件紧密结合。这导致在同一主机上运行多个应用程序时容易发生冲突,系统资源利用率低下,硬件成本高昂且缺乏灵活性。

虚拟化之后,操作系统和硬件之间的依赖关系被打破。通过封装技术,操作系统和应用程序被管理为独立的个体。这提供了强大的安全和故障隔离能力,并且虚拟机是独立于硬件的,能够在任何兼容的硬件上运行。

(2)存储虚拟化。存储虚拟化主要涉及 RAID、分布式存储系统及分布式文件系统。

RAID:RAID 技术主要用于解决单块硬盘在设备可靠性方面的问题。

分布式存储系统:主要是为应对大数据时代的数据存储需求而诞生的。

分布式文件系统:可以屏蔽底层存储设备的差异,实现数据的统一管理和访问。

(3)网络虚拟化。网络虚拟化是一个广泛的概念,主要包含了不同层面的虚拟化技术,如四层、七层网络虚拟化以及软件定义网络(SDN)等。其中,四层和七层网络虚拟化主要解耦了网络设备和租户之间的逻辑关系,使得租户可以更加灵活地配置和管理自己的网络资源;而二层和三层网络虚拟化则主要解耦了服务器和交换机之间的物理连接,实现了网络流量的灵活调度和优化。

网络虚拟化的核心目标是将传统基于硬件的网络转变为基于软件的网络,从而在物理硬件与使用该硬件的应用和服务之间引入一个抽象层。与所有形式的 IT 虚拟化一样,这一转变旨在提高网络的灵活性、可扩展性和资源利用率,同时降低运维成本。

3. KVM 虚拟化技术

KVM(kernel-based virtual machine)虚拟化技术是一种基于 Linux 内核的开源虚拟化技术,它能够将物理计算机资源划分为多个独立的虚拟机,每个虚拟机都能够运行自己的操作系统和应用程序。

KVM 作为集成在 Linux 内核中的系统虚拟化模块,利用 Linux 自身的调度器进行管理。它主要工作在 X86 架构上,并且需要硬件支持辅助虚拟化技术,如 Intel 的 VT 和 AMD 的 AMD-V。KVM 采用全虚拟化技术,并结合了混合虚拟化架构的优势,以提供高效且灵活的虚拟化环境。

1) KVM 概述

(1)起源与发展。KVM 最初由德国 Qumranet 公司开发,后来被 Red Hat 公司收购,并被整合为其企业级虚拟化解方案的重要部分。如今,KVM 已经被发展成为 Linux 操作系统的一个标准组件,得到了大多数主流 Linux 发行版的广泛支持。

(2)核心思想。KVM 的核心思想是将宿主机的硬件资源(如 CPU、内存、磁盘等)进行抽象和虚拟化,从而创建出多个独立的虚拟机。这些虚拟机可以同时运行在同一个物理服务器上,并且每个虚拟机都拥有自己独立的操作系统、应用程序和用户空间,彼此之间实现了完全的隔离。

2) KVM 架构

KVM 架构包含一个为处理器提供底层虚拟化、可加载的核心模块 `kvm. ko`(`kvm-intel. ko` 或 `kvm-amd. ko`)。同时,KVM 利用 QEMU(或称为 QEMU-KVM)作为虚拟机的上层管理和控制工具。值得注意的是,KVM 无须对 Linux 或 Windows 操作系统进行任何修改即可运行。

(1)KVM 模块:KVM 是 Linux 内核的一个组成部分,作为可加载模块存在。用户可以通过 `modprobe` 命令来加载 KVM 模块。然而,仅仅加载 KVM 模块是不足以创建和运行虚拟机的。因为用户无法直接通过内核模块进行操作,还需要一个运行在用户空间的工具来进行管理。为此,KVM 的开发者选择了已经成熟的开源虚拟化软件 QEMU 作为用户空间的工具。QEMU 具有虚拟不同的 CPU 架构的能力,KVM 在此基础上进行了部分改造和优化,使其成为能够控制和管理 KVM 虚拟机的用户空间工具。

(2)KVM 基本结构:KVM 是一种基于硬件虚拟化技术的完全虚拟化技术,其基本架构包括 KVM 驱动和 QEMU。KVM 驱动主要负责虚拟机的创建、虚拟内存分配、vCPU 寄存器读写和运行管理,而 QEMU 则用于模拟虚拟机的用户空间组件,提供 I/O 设备模型和访问外设的途径。

KVM 模块能让 Linux 主机转变为一个虚拟机监视器,在 Linux 原有的两种执行模式基础上,新增加了客户模式这一层级。客户模式内部进一步细分为内核模式和用户模式,各自拥有独特的运行环境。在虚拟机运行时,这三种模式的工作职责如下。

①客户模式(guest mode):在此模式下,虚拟机执行非 I/O 操作的客户代码。简而言之,它是客户机操作系统在其内部运行的模式。值得注意的是,客户模式自身还细分为内核模式与用户模式,分别处理不同的任务。

②用户模式(user mode):该模式为用户提供虚拟机管理的用户空间工具,并负责执行用户发起的 I/O 指令。QEMU 虚拟化软件即在此模式下运行,通过它用户可以管理虚拟机

并与其进行交互。

③内核模式(kernel mode):此模式负责模拟 CPU 及内存等硬件资源,实现客户模式之间的切换,并处理因 I/O 操作或其他指令触发的从客户模式退出的事件(即 VM_EXIT 事件)。KVM 模块主要在此模式下工作,通过它实现了对虚拟机硬件资源的虚拟化和管理。

子任务 2 了解分布式存储技术

职业能力目标和要求

- 了解分布式存储的概念。
- 掌握 Ceph 分布式存储技术。

1. 分布式存储介绍

分布式存储是一种计算机数据存储架构,它将数据分散存储在多台计算机或服务服务器上,旨在提升数据的高可靠性、可扩展性和整体性能。在此架构中,每一台计算机或服务器都充当一个存储节点,它们通过网络相互连接,进行通信与协作,共同实现数据的分布式存储和管理。分布式存储通常使用数据分片和副本复制技术,以确保数据的可靠性和可用性。

数据分片技术是指将数据切割成多个部分,每个部分独立地存储在不同的节点上,以此实现数据的广泛分布和高效管理。而副本复制技术则是为每个数据创建多个副本,并将这些副本分别存储在多个不同的节点上。这样的设计确保了即使部分节点发生故障,系统仍能从其他健康的节点中恢复数据,从而维持数据的高可用性。其工作原理如下。

1) 数据分片

在分布式存储系统中,数据被切割成多个部分,每个部分被分配到不同的节点上进行存储,以此实现数据的分布式存储和高效管理。数据分片的过程通常依赖于哈希函数或一致性哈希算法来完成。

2) 副本复制

为了增强数据的可靠性和可用性,分布式存储系统广泛采用副本复制技术。该技术确保每个数据副本都被保存在不同的节点上,这样即使某些节点发生故障,系统仍能从其他节点中恢复数据。副本复制的实现通常依赖于特定的复制策略,如简单复制、多副本复制和跨区域复制等。

3) 数据一致性

在分布式存储系统中,数据一致性占据着至关重要的地位。为了保障数据在不同节点间的一致性,分布式存储系统普遍采用数据同步与管理机制。例如,使用 Paxos 算法、Raft 算法或 ZooKeeper 等分布式协调服务来实现数据的有效同步和管理。

4) 数据访问

在分布式存储系统中,数据能够并行地从多个节点上进行读取和写入操作,从而显著提升读写性能和吞吐量。为实现高效的数据访问,系统通常会使用负载均衡机制,这些机制包括分布式哈希表、分布式缓存或分布式文件系统等先进技术。

5) 分布式存储的性能

易于扩展:得益于其合理的分布式架构,分布式存储系统能够实现可预估且灵活的扩

展,包括计算能力、存储容量和性能的弹性增长。

高性能:高性能的分布式存储系统通常能够高效地管理读缓存和写缓存,并且支持自动的存储层级划分与优化。

支持分级存储:分布式存储通过网络的松耦合链接,允许高速存储与低速存储分开部署,或者按照任意比例混合部署,提供了灵活的存储层级配置。

2. 分布式存储技术的分类

1) 分布式文件系统

数据被分散存储在多台机器上,这些专门用于存储数据的机器被称为存储节点。多个这样的节点共同构成了一个分布式集群,而每个节点上的小型分布式文件系统则组合成了整体的分布式文件系统。这一系统由主服务器进行统一管理。用户只需访问集群中的任意一台主机,就能轻松获取自己所需的目标文件。常见的分布式文件系统包括 GFS、HDFS、Lustre、Ceph、GridFS、MogileFS、TFS 和 FastDFS 等。

2) 对象存储

对象存储是一种用于存储和管理非结构化数据(这些数据以对象的形式存在)的技术。随着现代组织需要创建和分析的海量非结构化数据(如照片、视频、电子邮件、网页、传感器数据和音频文件)不断增加,云对象存储系统应运而生。这些系统将数据分布在多个物理设备上,但为用户提供了一个统一的虚拟存储库,使他们能够高效地访问所需内容。对象存储解决方案非常适合用于构建需要高度扩展性和灵活性的云原生应用程序;此外,这些解决方案还可以用于导入现有数据存储,以便进行分析、备份或存档。

3) 数据块存储

数据块存储是一种控制数据存储和存储设备管理的技术。它能够将任何类型的数据(如文件或数据库条目)分割成大小相等的数据块。随后,数据块存储系统会以优化的方式,将这些数据块存储在底层的物理存储介质上,以便于快速访问和检索。对于那些需要高效、迅速且可靠数据访问的应用程序而言,开发人员更倾向于采用数据块存储。数据块存储可以被视为一种更为直接的数据管道。与之相比,文件存储则包含了一个额外的文件系统层(如 NFS、SMB),在访问数据之前需要进行额外的处理。

4) 分布式数据库

分布式数据库系统通常使用较小的计算机系统,每台计算机都可单独放在一个地方,每台计算机中都可能存有 DBMS 的一份完整拷贝副本,或者部分拷贝副本,并具有自己局部的数据库,位于不同地点的许多计算机通过网络互相连接,共同组成一个完整的、全局的逻辑上集中、物理上分布的大型数据库。可分为三大类:同构同质型 DDBS、同构异质型 DDBS、异构型 DDBS。

5) 列式存储

列式存储是一种存储结构化数据的技术,在底层的存储介质上,数据是以列为单位进行组织的。具体来说,它会先存储完若干条记录的首个字段后,接着存储这些记录的第二个字段,然后是第三个字段,以此类推,直至所有字段都存储完毕。之后,按照同样的方式,继续组织存储下一批若干条记录的所有字段。目前,最受欢迎的开源列式存储引擎当属 Apache Parquet。

6) NoSQL 存储

NoSQL 存储是一类专为大规模数据存储而设计的非关系型数据库系统。它们通常不使用传统的表格关系模型,而是采用更加灵活的数据模型,如键值对、文档、列存储或图形数据库,以支持快速开发和水平扩展。NoSQL 数据库因其能够处理大量分布式数据、支持多种数据结构和快速查询性能,而被广泛应用于现代应用程序开发,特别适合于大数据和实时 Web 应用。典型的 NoSQL 数据库包括键值数据库、文档数据库、列式存储数据库、多模数据库等。

3. Ceph 分布式存储技术

1) Ceph 简介

Ceph 是当前最热门的分布式存储系统之一,是软件定义存储 (software defined storage, SDS) 解决方案中的典范,应用范围包括块存储 (rados block device, RBD)、文件存储 (CephFS, ceph file system) 和对象存储 (reliable、autonomic、distributed、object storage gateway, RADOSGW),被广泛应用于云计算底层 IAAS 平台,更是 OpenStack 首选的后端存储,不管是互联网企业还是金融等传统行业,都被广泛应用。

2) 核心组件

(1) Monitor: 一个 Ceph 集群需要多个 Monitor 组成的小集群,它们通过 Paxos 同步数据,用来保存 OSD 的元数据。

(2) OSD: 全称 object storage device,也就是负责响应客户端请求返回具体数据的进程。一个 Ceph 集群一般有多个 OSD。

(3) MDS: 全称 ceph metadata server,是 CephFS 服务依赖的元数据服务。

(4) Object: Ceph 最底层的存储单元是 Object 对象,每个 Object 包含元数据和原始数据。

(5) RADOS: 全称 reliable autonomic distributed object store,是 Ceph 集群的精华,用户实现数据分配、Failover 等集群操作。

(6) CRUSH: Ceph 使用的数据分布算法,类似一致性哈希,将数据分配到预期的位置。

(7) RBD: 是 Ceph 对外提供的块设备服务。

(8) CephFS: 全称 ceph file system,是 Ceph 对外提供的文件系统服务。

3) Ceph 的特性

(1) 高扩展性: 随着节点数量的增加,Ceph 的存储容量和性能呈线性增长,支持上千个存储节点规模,满足从 TB 到 PB 级的数据存储需求。

(2) 高性能: Ceph 采用独特的分布式架构,利用缓存一致性协议和数据复制技术,可提供高性能的 I/O 操作。

(3) 高可靠性: 通过数据冗余和故障恢复机制,保障数据的强一致性和高可用性。在发生故障时,Ceph 能够自动检测并修复故障,保证数据的可靠性和系统的稳定性。

(4) 去中心化: Ceph 采用去中心化的架构设计,避免了传统中心化存储系统的单点故障问题。

(5) 灵活的副本数: 根据实际需求,可以灵活配置副本数,以在保证数据可靠性的同时降低存储成本。

(6) 丰富的存储接口: Ceph 支持块存储、文件存储和对象存储三种存储接口,并支持自

定义接口和多种语言驱动,满足不同应用场景的需求。

子任务 3 了解分布式监控技术

职业能力目标和要求

- 了解分布式监控系统的核心概念。
- 了解什么是分布式监控技术及其优势特点。
- 了解掌握常用监控工具的基础架构及其区别。

1. 分布式监控介绍

分布式监控是一种用于大型分布式系统或网络的监控技术,它能够跨越多个节点、服务器、应用程序及设备,全面收集、分析并展示系统的运行状态和性能数据。分布式监控系统的设计目标是提供全面、实时且准确的监控信息,以帮助系统管理员、开发人员和运维团队及时发现和解决问题,从而确保系统的稳定运行。分布式监控技术的算法原理如下。

(1)时间序列数据库:用于存储和高效查询海量的时间序列数据,如 OpenTSDB 和 InfluxDB。这些数据库能够高效地处理大量的监控数据,并提供灵活的查询功能。

(2)异常检测算法:包括统计方法和机器学习方法。统计方法基于历史数据的统计规律来设定阈值,如平均值和方差;机器学习方法则通过训练数据来识别异常模式。

(3)关联分析算法:用于识别不同指标之间的关联关系,帮助定位潜在的问题根源。例如,将 CPU 利用率和内存使用情况关联起来分析,有助于发现系统潜在的性能瓶颈。

2. 分布式监控系统的分类

1) 分布式追踪工具(Jaeger)

(1)简介。Jaeger 是一种分布式追踪工具,用于跟踪和监控微服务之间的请求链路,旨在帮助用户发现性能瓶颈和故障点。分布式追踪(distributed tracing)技术专门用于监测和分析分布式系统中请求的传播路径和性能。在分布式系统中,一个请求通常会经过多个不同的服务和组件,每个组件都可能会对请求进行处理并做出响应。分布式追踪的目的是跟踪和记录请求在系统中的传播路径和各个组件的性能指标,以便于分析和优化系统的性能和可靠性。

(2)特点。

①分布式追踪:Jaeger 能够全面跟踪和可视化微服务之间的请求和事务,使得用户能够清晰地了解请求在系统中的流动情况。

②高可扩展性:Jaeger 支持大规模分布式系统,能够应对复杂且庞大的微服务架构。

③强大的集成性:Jaeger 能够与多种编程语言和框架集成实现无缝集成,为用户提供便捷的使用体验。

2) 指标监控和时序数据库(Prometheus)

(1)简介。Prometheus 用于收集、存储和可视化系统的关键指标和性能数据,帮助用户实时了解系统的状态和性能。

指标监控是一种对系统或应用性能、健康状况、安全状况等关键指标进行持续追踪、度量和管理的过程。这些指标可能包括但不限于 CPU 使用率、内存占用、网络流量、响应时间

及错误率等。

时序数据库(time series database, TSDB)是一种专门用于处理时间序列数据的数据库。时间序列数据是按照时间顺序排列的数据点集合,通常用于记录系统或设备的实时运行状态、性能指标等。

(2)特点。

①时间序列数据:Prometheus 专注于时间序列数据的收集、存储和查询。

②丰富的数据源:支持从各种服务和应用收集数据,包括数据库、HTTP 服务和缓存等。

③强大的查询语言:PromQL(prometheus query language)允许用户以复杂的方式查询时间序列数据。

④告警与通知:提供告警机制,当监控指标达到预设阈值时,可以发送通知。

3)日志管理和分析工具(Elasticsearch)

Elasticsearch 用于收集、存储并分析微服务的日志数据,以便高效地诊断和解决问题。该工具集提供了一个流行的日志管理和分析解决方案,由 Elasticsearch、Logstash 和 Kibana 组成,它强大的搜索和可视化工具特别适用于大型和分布式环境。

Elasticsearch 是一个分布式搜索和分析引擎,用于存储和查询日志数据。

Logstash 是一个数据收集管道,用于从各种来源接收、转换和发送数据到 Elasticsearch。

Kibana 是一个数据可视化工具,允许用户通过仪表板、图表和地图来探索和分析存储在 Elasticsearch 中的数据。

4)综合监控工具(Zabbix)

(1)简介。Zabbix 能够监控各种 IT 组件,包括网络设备、服务器和虚拟机等。Zabbix 提供了丰富的监控模板和灵活的触发器配置,可以实时监控系统的性能和可用性。

(2)特点。

①跨平台支持:Zabbix 支持多种操作系统,如 Linux、Unix、Windows 等,这使得它们可以在不同的 IT 环境中进行部署和使用。

②广泛的监控范围:能够监控各种 IT 资源,包括网络设备、服务器、虚拟机、数据库及应用程序等;监控项涵盖了 CPU 利用率、内存使用情况、网络流量及磁盘空间等关键性能指标。

③实时监控与报警:Zabbix 提供实时监控功能,能够实时地收集和分析被监控资源的性能指标。当性能指标超过预设的阈值时,能够触发报警,并通过多种方式通知管理员。

3. 常用监控工具(Zabbix\Prometheus)

1)Zabbix 监控系统介绍

Zabbix 是一个基于 Web 界面的提供分布式系统监视以及网络监视功能的企业级的开源解决方案。Zabbix 能监视各种网络参数,保证服务器系统的安全运营;并提供灵活的通知机制以让系统管理员快速定位/解决存在的各种问题。

Zabbix 的核心由两部分构成:Zabbix server 和 Zabbix agent。Zabbix server 负责接收 agent 发送的数据,并存储在数据库中。Zabbix agent 安装在被监控的目标服务器上,负责收集目标服务器的各种监控数据,并将数据发送给 Zabbix server。此外,Zabbix 还支持 SNMP、JMX 等多种监控方式,可以实现对网络设备、Java 应用等的监控。

Zabbix 提供了丰富的可视化界面和报表功能,可以展示实时的监控数据、历史数据以及各种统计图表,能够帮助系统管理员更好地了解系统的运行状况。

2)Prometheus 监控系统介绍

Prometheus 是一个开源的服务监控系统和时序数据库,特别适合用于容器化环境的监控。它使用 Go 语言开发,并通过 HTTP 进行数据抓取。Prometheus 与云原生技术(如 Kubernetes)紧密结合,适用于现代微服务架构的监控需求。

Prometheus 的基本原理是通过 HTTP 周期性抓取被监控组件的状态,任意组件只要提供对应的 HTTP 接口并且符合 Prometheus 定义的数据格式,就可以接入 Prometheus 监控。

Prometheus Server 负责定时从目标系统上抓取 Metrics(指标)数据,并将这些数据保存到本地存储中。Prometheus 采用了一种 Pull(拉取)模式来获取数据,这种模式不仅降低了客户端的复杂度——客户端只需要采集数据,无须了解服务端的具体情况,同时也使得服务端可以更加容易地实现水平扩展。

当监控数据达到预设的告警阈值时,Prometheus Server 会通过 HTTP 将告警信息发送给告警管理模块 Alertmanager。Alertmanager 在接收到后,会进行告警的抑制处理,并根据配置触发相应的邮件通知或 Webhook。

Prometheus 支持 PromQL 查询语言,提供了丰富的多维度数据模型和灵活的查询能力。通过监控指标与多个标签(tag)的关联,用户可以将监控数据进行任意维度的组合与聚合,以满足不同的监控需求。

3)综合对比

Zabbix 与 Prometheus 的对比如表 1-1 所示。

表 1-1 Zabbix 与 Prometheus 的对比

类 别	Zabbix	Prometheus
后端	后端用 C 语言开发,界面用 PHP 语言开发,定制化难度较高	后端用 golang 语言开发,前端可视化通常与 Grafana 集成,用户只需通过 JSON 进行配置编辑即可完成定制。定制化难度相对较低
集群规模	集群规模上限为 10 000 个节点	支持更大的集群规模,速度也更快
监控数据存储	监控数据存储在关系型数据库内,如 MySQL,很难从现有数据中扩展维度	监控数据存储在基于时间序列的数据库内,便于对已有数据进行新的聚合
安装	安装简单,zabbix-server 一个软件包中包括了所有的服务端功能	安装相对复杂,监控、告警和界面都分属于不同的组件
界面	图形化界面比较成熟,界面上基本上能完成全部的配置操作	界面相对较弱,很多配置需要修改配置文件

课后习题

一、选择题

- 云计算最显著的优势之一是()。
 - 需要大量本地硬件
 - 提供无限的存储空间
 - 降低了对本地硬件的依赖
 - 无法从任何地方访问数据
- 分布式系统中,监控系统通过()确保数据的一致性。
 - 使用分布式文件系统
 - 依赖于单一节点进行数据汇总
 - 运行一致性算法如 Raft 或 Paxos
 - 定期重启监控服务
- ()常用于分布式系统中远程监控和管理。
 - HTTP
 - SNMP
 - FTP
 - SMTP
- ()不属于分布式文件系统。
 - HDFS
 - GFS
 - Ceph
 - RAID
- ()不是分布式存储系统中的主要数据冗余技术。
 - 副本技术
 - 分布式哈希表
 - 分布式存储网络
 - 纠删码

二、填空题

- 云计算具备_____、_____、_____优势。
- 云计算的关键技术包括_____、_____、_____、_____。
- 在云计算中,_____服务模型为用户提供对网络应用的访问,这些应用运行在云基础设施上并处理用户数据。
- 计算虚拟化主要包括_____。
- 在公有云模型中,云基础设施由_____的云服务提供商所有和运营,为广泛的、公开的用户群体提供服务。

三、简答题

- 分布式追踪工具的特点是什么?
- 虚拟化的三层含义是什么?
- 分布式存储的工作原理是什么?



在线测试 1