项目-安装 openEuler 操作系统

任务六 实现 openEuler 的电源管理

【任务介绍】

通过控制台命令实现 openEuler 的电源管理,掌握重启系统、待机、休眠、关闭操作系统的 方法。

【任务目标】

(1) 掌握使用 systemctl 管理系统电源的方法。

(2) 掌握对操作系统进行电源管理的基本命令。

【操作步骤】

步骤 1: 了解 systemctl 命令与 Linux 管理命令的关系。

openEuler 中通过 systemctl 命令进行系统电源管理,实现系统重启、休眠、关机操作。当前仍 兼容部分 Linux 中的管理命令, systemctl 命令与 Linux 管理命令对应关系见表 1-6-1。建议用户使 用 systemctl 命令来进行管理。

序号	Linux 管理命令	systemctl 命令	命令描述
1	reboot	systemctl reboot	重启系统
2	halt	systemctl halt	关闭系统但不下电
3	poweroff	systemctl poweroff	关闭系统并下电

表 1-6-1 命令对应关系

步骤 2: 重启系统。

重启系统,执行如下命令。

操作命令:

- 1. # 设置重启系统
- 2. [root@Project-01-Task-01 ~]# systemctl reboot

操作命令+配置文件+脚本程序+结束

步骤 3:设置系统待机。

设置系统待机,执行如下命令。

操作命令:

项目-

- 1. # 设置系统待机
- 2. [root@Project-01-Task-01 ~]# systemctl suspend

操作命令+配置文件+脚本程序+结束

步骤4:设置系统休眠。

(1) 设置系统休眠,执行如下命令。

操作命令:

- 1. # 设置系统休眠
- 2. [root@Project-01-Task-01 ~]# systemctl hibernate

操作命令+配置文件+脚本程序+结束

(2) 设置系统待机且处于休眠状态,执行如下命令。

操作命令:

1. # 设置系统待机且处于休眠状态

2. [root@Project-01-Task-01 ~]# systemctl hybrid-sleep

步骤 5:关闭系统。

(1) 关闭系统但不下电。

操作命令:

- 1. # 关闭系统但不下电
- 2. [root@Project-01-Task-01 ~]# systemctl halt

操作命令+配置文件+脚本程序+结束

操作命令+配置文件+脚本程序+结束

(2) 关闭系统并下电。

操作命令:

- 1. # 关闭系统并下电
- 2. [root@Project-01-Task-01 ~]# systemctl poweroff

操作命令+配置文件+脚本程序+结束

【任务扩展】

电源管理策略包括重启、休眠和关机,详细描述如下。

(1) 重启:系统重新启动,但不会关闭或切断电源。

(2)休眠:系统休眠有三种模式,分别是 suspend(suspend to RAM)、hibernate(suspend to disk)、 hybrid-sleep(suspend to both)。suspend 是除了内存外的机器部件都进入断电状态,特点是系统状态 保存在内存中,恢复速度快; hibernate(suspend to disk)是内存中的系统状态写入交换空间内,当系 统启动时从交换空间内读回系统状态,特点是系统状态保存到交换空间,恢复速度慢; hybrid-sleep (suspend to both)是结合了 suspend 和 hibernate 两种类型,像 hibernate 一样将系统状态存入交换空 间内,同时也像 suspend 一样不关闭电源,特点是电源电量未耗尽之前,快速从休眠状态恢复,休 眠期间电源耗尽,则从交换空间恢复系统状态。

(3)关机:管理有两种模式,分别是 halt(关闭系统但不下电)和 poweroff(关闭系统并下 电)。halt 是关闭并停止系统,并未关闭电源; poweroff 是关闭计算机电源。 项目—

项目一

and the second s



任务七 服务管理

【任务介绍】

openEuler 中的服务是一种特殊的进程,可以在后台运行并提供一系列的功能和服务,本任务 讲解如何使用 systemd 进行服务管理。

本任务在任务一的基础上进行。

【任务目标】

- (1) 掌握服务的启动、停止和重启。
- (2) 掌握配置信息载入。
- (3) 掌握配置服务开机自启动。
- (4) 理解自定义服务注册。

【操作步骤】

步骤 1: 了解 systemd。

systemd 是系统和服务管理器,使用 systemetl 命令来运行、关闭、重启、显示、启用/禁用系 统服务。

步骤 2: 查看服务状态。

(1) 查看服务列表,此处不再列出全部查看的结果。

操作命令:

- 1. # 显示当前正在运行的服务
- 2. [root@Project-03-Task-01 ~]# systemctl list-units --type service

3.

4. # 显示所有的服务

5. [root@Project-03-Task-01 ~]# systemctl list-units --type service --all

6.

操作命令+配置文件+脚本程序+结束

项目三

(2) 查看 crontab 服务状态。

操作命令:

.....

1.	# 查看 crontab 服务状态	
2.	[root@Project-03-Task-01 ~]# systemctl status crond.service	
3.	• crond.service - Command Scheduler	
4.	Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; vendor preset: enabled)	
5.	Active: active (running) since Sun 2023-11-12 18:31:14 CST; 1h 8min ago	
6.	Main PID: 924 (crond)	
7.	Tasks: 1 (limit: 9128)	
8.	Memory: 1.3M	
9.	CGroup: /system.slice/crond.service	
10.	└── 924 /usr/sbin/crond -n	I
11.		
12.	11 月 12 18:31:14 Project-03-Task-01 systemd[1]: Started Command Scheduler.	
13.	11 月 12 18:31:14 Project-03-Task-01 crond[924]: (CRON) STARTUP (1.6.1)	
14.	11 月 12 18:31:14 Project-03-Task-01 crond[924]: (CRON) INFO (Syslog will be used instead of send	
	mail.)	
15.	11月 12 18:31:14 Project-03-Task-01 crond[924]: (CRON) INFO (RANDOM_DELAY will be scaled with	
	factor 16% if used.)	
16.	11 月 12 18:31:14 Project-03-Task-01 crond[924]: (CRON) INFO (running with inotify support)	
17.	11 月 12 18:39:01 Project-03-Task-01 crond[924]: (root) RELOAD (/var/spool/cron/root)	
18	11 \exists 12 19:01:01 Project-03-Task-01 CROND[1698]: (root) CMD (run-parts /etc/cron hourly)	
10.	-11 /j 12 17.01.01 110jee-05-1ask-01 exono[1070]. (1001) ewin (101-parts / exonolinoutry)	
	探作命令+配直义件+脚本程序+结束	

服务状态显示参数说明如下。



- ▶ Loaded:显示服务是否被加载,服务对应的绝对路径以及是否开机启动
- Active: 显示服务是否正在运行,以及开始运行的时间节点
- Main PID: 系统服务的 PID 值
- CGroup: 相关控制组的信息

步骤 3: 服务的启动、停止、重启。

(1) 使用 systemctl is-active 命令检查服务是否运行,以查看防火墙为例。

操作命令:

- 1. # 查看防火墙状态
- 2. [root@Project-03-Task-01 ~]# systemctl is-active firewalld.service
- 3. active

操作命令+配置文件+脚本程序+结束



- 使用 is-active 返回结果及意义如下。
- active: 有一个或多个程序正在系统中执行
- inactive: 服务没有运行

(2) 使用 systemctl stop name.service 命令停止服务。

操作命令:

- 1. # 停止防火墙服务
- 2. [root@Project-03-Task-01 ~]# systemctl stop firewalld.service
- 3. # 查看防火墙状态
- 4. [root@Project-03-Task-01 ~]# systemctl is-active firewalld.service
- 5. inactive

操作命令+配置文件+脚本程序+结束

(3) 使用 systemctl start name.service 命令启动服务。

操作命令:

项目三

- 1. # 启动防火墙服务
- 2. [root@Project-03-Task-01 ~]# systemctl start firewalld.service
- 3. # 查看防火墙状态
- 4. [root@Project-03-Task-01 ~]# systemctl is-active firewalld.service
- 5. active

操作命令+配置文件+脚本程序+结束

(4) 使用 systemctl restart name.service 命令重启服务。

操作命令:

- 1. # 重启防火墙服务
- 2. [root@Project-03-Task-01 ~]# systemctl restart firewalld.service
- 3. # 查看防火墙状态
- [root@Project-03-Task-01 ~]# systemctl is-active firewalld.service
 active

操作命令+配置文件+脚本程序+结束

步骤4:服务的配置信息载入。

使用 systemctl reload firewalld.service 命令重新加载 firewalld 服务。

操作命令:

- 1. # 重新加载 firewalld 服务
- 2. [root@Project-03-Task-01 ~]# systemctl reload firewalld.service

操作命令+配置文件+脚本程序+结束

步骤 5: 设置服务开机自启动。

(1) 使用 systemetl disable name.service 命令禁止服务开机自启动。

操作命令:

1. # 禁止服务开机自启动



2. [root@Project-03-Task-01 system]# systemctl daemon-reload

操作命令+配置文件+脚本程序+结束

(4) 启动自定义服务。

操作命令:

8

1.	# 启动自定义服务
2.	[root@Project-03-Task-01 system]# systemctl start project0307.service
	操作命令+配置文件+脚本程序+结束
	(5) 查看服务状态。
操作	命令:
1.	# 查看服务状态
2.	[root@Project-03-Task-01 opt]# systemctl status project0307
3.	• project0307.service - Project 0307
4.	Loaded: loaded (/usr/lib/systemd/system/project0307.service; disabled; vendor preset: disabled)
5.	Active: active (running) since Mon 2023-11-13 01:03:47 CST; 4s ago
6.	Main PID: 2565 (bash)
7.	Tasks: 2 (limit: 9128)
8.	Memory: 344.0K
9.	CGroup: /system.slice/project0307.service
10.	└── 2565 /usr/bin/bash /opt/project0307.sh
11.	└── 2575 sleep 1
12.	
13.	11 月 13 01:03:47 Project-03-Task-01 systemd[1]: Started Project 03 07.

操作命令+配置文件+脚本程序+结束

项目三

项目五 使用 Nginx 实现代理服务

任务五 使用 Apache Proxy 实现负载均衡



【任务介绍】

本任务使用 Apache 的 mod_proxy 模块实现反向代理和负载均衡服务,并介绍 Apache 负载均 衡服务的运行监控。

本任务和任务三实现的功能基本相同,本任务使用 Apache 实现,任务三使用 Nginx 实现。 本任务在任务二、任务三的基础上进行。

【任务目标】

- (1) 掌握使用 Apache 反向代理与负载均衡发布网站服务。
- (2) 掌握 Apache 负载均衡的测试。
- (3) 掌握 Apache 负载均衡的运行监控。

【任务设计】

本任务拓扑结构如图 5-5-1 所示。

服务器规划见表 5-5-1, 服务器网络规划见表 5-5-2, 网站业务规划见表 5-5-3。

表 5-5-1 服务器规划

虚拟机名称	主机名	服务器	作用
VM-Project-05-Task-05-10.10.2.55	Project-05-Task-05	代理服务器-Apache	使用代理实现网站负载均 衡发布
VM-Project-05-Task-02-172.16.0.1	Project-05-Task-02	Web 服务器-内部-1	发布内部网站业务-1
VM-Project-05-Task-03-172.16.0.2	Project-05-Task-03	Web 服务器-内部-2	发布内部网站业务-2



图 5-5-1 拍扑结构

表 5-5-2 服务器网络规划

主机名	网卡	IP 地址	子网掩码	网关	DNS	说明
Project-05-Task-05	网卡1	10.10.2.55	255.255.255.0	10.10.2.1	8.8.8.8	用于提供 Web 服务
Project-05-Task-05	网卡2	172.16.0.253	255.255.255.0	不配置	不配置	用于与内部网站所 在主机进行通信
D : 405 T 102	网卡1	10.10.2.52	255.255.255.0	10.10.2.1	8.8.8.8	Web服务器软件安装 阶段配置网络信息
Project-05-Task-02	网卡1	172.16.0.1	255.255.255.0	不配置	不配置	提供内部网站服务 阶段网络配置信息
D : 405 T 1 02	网卡1	10.10.2.53	255.255.255.0	10.10.2.1	8.8.8.8	Web服务器软件安装 阶段配置网络信息
Project-05-Task-03	网卡1	172.16.0.2	255.255.255.0	不配置	不配置	提供内部网站服务 阶段网络配置信息

表 5-5-3 Web 服务器-内部-网站规划

网站名称	服务器	网站目录	访问地址	网站首页内容
Site-Clone-1	Web 服务器-内部-1	/var/www/html	http://172.16.0.1	Site-Clone-1: http://172.16.0.1
Site-Clone-2 Web 服务器-内部-2		/var/www/html	http://172.16.0.2	Site-Clone-2: http://172.16.0.2

【操作步骤】

10

步骤 1:发布网站 Site-Clone-1。 该步骤在本项目的任务二中已经完成。

项目五

项目五

步骤 2: 发布网站 Site-Clone-2。

该步骤在本项目的任务三中已经完成。

步骤 3: 创建虚拟机并完成 openEuler 的安装。

在 VirtualBox 中创建虚拟机,完成 openEuler 的安装。虚拟机与操作系统的配置信息见表 5-5-4,注意虚拟机网卡的工作模式为桥接。

表 5-5-4 虚拟机与操作系统配置

虚拟机配置	操作系统配置
虚拟机名称: VM-Project-05-Task-05-10.10.2.55	主机名: Project-05-Task-05
内存: 1GB	IP地址: 10.10.2.55
CPU: 1 颗 1 核心	子网掩码: 255.255.255.0
虚拟硬盘: 20GB	网关: 10.10.2.1
网卡:1块,桥接	DNS: 8.8.8.8

步骤 4: 完成虚拟机的主机配置、网络配置及通信测试。

启动并登录虚拟机,依据表 5-5-4 完成主机名和网络的配置,能够访问互联网和本地主机。

项目五

11



(1)虚拟机的创建、操作系统的安装、主机名与网络的配置,具体方法参见项目一。

(2)建议通过虚拟机复制快速创建所需环境。通过复制创建的虚拟机需依据本任务虚拟机与操作系统规划配置信息设置主机名与网络,实现对互联网和本地主机的访问。

步骤 5: 虚拟机增加网卡并配置网络。

将主机名为 Project-05-Task-05 的虚拟机关机,在 VirtualBox 虚拟机清单中选中虚拟机 VM-Project-05-Task-05-10.10.2.55,右击"设置"命令,单击"网络"标签,选择"网卡2"选项卡, 配置网卡2网络,选中"启用网络连接"前的复选框,连接方式设置为内部网络,如图 5-5-2 所示。

VM-Project-05-Ta	usk-05-10.10.2.55 - 设置	2		-	0	×
常规	网络					
系统	网卡1 网卡2	网卡 3 网卡 4				
显示	启用网络连接(E) 连接方式(A):	内解7络~				
存储	名称(2): ▼ 高級(A)	intnet				~
网络	控制芯片(工):	Intel PRO/1000 MT 桌面 (82540EM)				~
串口	混杂模式(g):	拒绝				×
USB设备	nac fight (n):	08002773E36D				1
】 共享文件夹 】 用户界面		■ 18AP193 €/				
			確定	取消	裁則	h(н)

图 5-5-2 虚拟机增加网卡

配置完成后,单击"确定"按钮,并将此虚拟机开机。查看虚拟机网络配置信息,具体操作步骤如下。

操作命令:

- 1. #查看网络配置
- 2. [root@Project-05-Task-05 ~]# ip addr
- 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
- 4. link/loopback 00:00:00:00:00 brd 00:00:00:00:00
- 5. inet 127.0.0.1/8 scope host lo
- 6. valid_lft forever preferred_lft forever
- 7. inet6 ::1/128 scope host
- 8. valid_lft forever preferred_lft forever
- 9. #网卡1 网络配置

项目五

- 2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group def ault qlen 1000
- 11. link/ether 08:00:27:64:a9:00 brd ff:ff:ff:ff:ff:ff
- 12. inet 10.10.2.55/24 brd 10.10.2.255 scope global noprefixroute enp0s3
- 13. valid_lft forever preferred_lft forever
- 14. inet6 fe80::a00:27ff:fe64:a900/64 scope link noprefixroute
- 15. valid_lft forever preferred_lft forever
- 16. #网卡2 网络配置

12

- 17. 3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group def ault qlen 1000
- 18. link/ether 08:00:27:73:e3:6d brd ff:ff:ff:ff:ff:ff

操作命令+配置文件+脚本程序+结束

使用 nmtui 配置网卡 2,如图 5-5-3 所示,单击"编辑连接"命令,单击"确定"按钮,单击 "添加"命令,连接类型选择"以太网",依据表 5-5-2 网络规划,填写网卡 2 信息,完成网络配置,如图 5-5-4 所示。



()	ē用 Nginx 实	现代理服务	项目五
電量集名称 enpose 设备 enpose (08:00:27:73:E3:60) = 以太网 ■ IPv4 配置 <f3> - (72.16:0.223/24 - (72.16:0.223/24</f3>	<显示> <隐蔵>		
- IPV6 配置 <自动> [X] 自动连接 [X] 对所有用户有效	<显示>		
	<取消> 200000		

图 5-5-4 网卡 2 网络配置

操作命令:

- 1. #可视化编辑网络配置
- 2. [root@Project-05-Task-05 ~]# nmtui

操作命令+配置文件+脚本程序+结束

项目五

步骤 6: 安装并配置 Apache HTTP Server。

操作命令:

- 1. #完成 Apache 安装并配置开机启动
- 2. [root@Project-05-Task-05 ~]# yum install -y httpd
- 3. [root@Project-05-Task-05 ~]# systemctl start httpd
- 4. [root@Project-05-Task-05 ~]# systemctl enable httpd

操作命令+配置文件+脚本程序+结束

步骤 7: 配置 Apache 实现负载均衡。

本步骤通过 Apache 的 mod_proxy、mod_proxy_balancer 和 mod_lbmethod_byrequests 模块实现 负载均衡发布网站。通过 Apache mod_proxy 模块的 ProxyPass 指令可以实现网站的反向代理,本 任务不作介绍。

使用 vi 工具编辑 Apache 的 httpd.conf 配置文件,编辑后的文件信息如下所示。

配置文件: /etc/httpd/conf/httpd.conf

操作命令:

- 1. #httpd.conf 配置文件内容较多,本部分仅显示与负载均衡配置有关的内容
- 2. #配置 Apache mod_proxy 禁止使用反向代理,以负载均衡方式发布网站服务
- 3. ProxyRequests Off
- 4. #定义负载均衡网站服务器组名称、网站服务器地址信息、权重信息
- 5. <Proxy balancer://load2>
- 6. BalancerMember http://172.16.0.1:80 loadfactor=1
- 7. BalancerMember http://172.16.0.2:80 loadfactor=3
- 8. </Proxy>
- 9. #定义代理转发请求到负载均衡网站服务器组 load2



10. ProxyPass / balancer://load2

小贴士

操作命令+配置文件+脚本程序+结束

Apache 实现负载均衡,主要有以下4种算法模型。

- byrequests,按照请求次数进行负载均衡,未设置负载均衡模式时,默认使用该项
- bytraffic,按照请求字节数进行负载均衡
- ▶ bybusyness,按照繁忙程度进行负载均衡,总是分配给活跃请求最少的节点
- heartbeat,按照心跳流量进行负载均衡,Apache 2.3 及之后版本才支持的 负载均衡模式,属于实验性质的模块

步骤 8: 配置 Apache 负载均衡服务的实时监控。

Apache 通过 mod_status 和 mod_proxy_balancer 两个模块实现负载均衡运行状态和性能的实时 监控。

1. 配置 Apache 实现负载均衡服务的实时监控

实现 Apache 的负载均衡服务实时监控功能需要修改配置文件 httpd.conf,通过 vi 工具编辑 httpd.conf 配置文件,编辑后的文件信息如下所示。

配置文件: /etc/httpd/conf/httpd.conf

操作命令:

1. #httpd.conf 配置文	件内容较多,	本部分仅显示与开启负载均衡运行监控有关的内容
--------------------	--------	------------------------

- 2. #配置网站负载均衡实时监控访问路径
- 3. <Location "/lb-status">
- 4. #设置不进行转发
- 5. proxypass !
- 6. #设置监控负载均衡性能
- 7. SetHandler balancer-manager
- 8. </Location>

操作命令+配置文件+脚本程序+结束

2. 配置完成后重新载入配置文件使其生效

操作命令:

- 1. #重新载入 Apache 配置文件
- 2. [root@Project-05-Task-05 ~]# systemctl reload httpd

操作命令+配置文件+脚本程序+结束

3. 开启防火墙 http 80 端口

操作命令:

- 1. #开启防火墙 80 端口
- 2. [root@Project-05-Task-05 ~]# firewall-cmd --add-port=80/tcp --permanent
- 3. #重新加载防火墙配置

14

项目五

项目五

4. [root@Project-05-Task-05 ~]# firewall-cmd --reload

操作命令+配置文件+脚本程序+结束

4. 查看 Apache 负载均衡服务的实时监控信息

在本地主机上打开浏览器访问 Apache 负载均衡实时监控页(http://10.10.2.55/lb-status),将显示当前 Apache 代理服务器的负载均衡业务状态和性能等信息。



(1)负载均衡运行监控反映了负载均衡的运行状态等敏感信息,建议设置为 指定范围访问。

(2)该页面不仅可以实时监控负载均衡性能情况,还可以进行负载均衡网站 服务器组、负载均衡节点的参数配置。

步骤 9: 验证负载均衡服务。

本步骤通过两个方法验证负载均衡服务,一种方法是通过本地主机浏览器访问以人工验证;另 一种方法是通过负载均衡实时监控信息以确认验证。

(1) 在本地主机打开浏览器,输入 Apache 代理服务器的默认网站访问地址,即可看到内部 网站页面,手动进行多次刷新页面,则网站 Site-Clone-1、网站 Site-Clone-2 将相继出现,两者出现 的频率比例接近于 1:3。

(2)使用 Fiddler 软件模拟客户端测试。在本地主机使用 Fiddler 软件发送 1000 个请求,对 Apache 代理服务器进行压力测试,具体操作方法参见本项目的任务四。在本地主机上打开浏览器 访问 Apache 负载均衡实时监控页,如图 5-5-5 所示。通过列表可以看出请求转发到网站 Site-Clone-1 与 Site-Clone-2 之间比例约为 1:3,通过验证证实负载均衡业务服务正常,权重比例起效。

← C ▲ RR# 10:10:255/0-utatas ●●	ф п () н	ancer Manager		× -	H.															-	0	×
Server Version: Apache/2.4.51 (Unix) Server Version: Apache/2.4.51 (Unix) Server Version: Apache/2.4.51 (Unix) Server Buil: Jun 28 2023 14:1120 Balancer changes will NOT be persisted on restart. Balancer sare inherited from main server. ProorPass settings are inherited from main server. 2 (2 Used) (None) Off 0 Vorker URL Route RouteRedir Factor Set Status Elected Busy Load Term HC Method HC Interval Passes Fails HC url HC Expr http://122.16.01 1.00 1 int Ck (251 0 - 200 (49k (44k N)) NONE 30000mm (10) (10)	← C ▲	不安全 10.1	0.2.55/lb-	status													аљ	Aø.	☆			b
Server Version: Apache/2.451 (Unit) Server Built: Unit 20221 (41:12) Balancer changes will NOI be persisted on restart. Balancers are inherited from main server. ProxyPass settings are inherited from main server. MaxMembers StickySession DisableFailover Timeout FailoverAttempts MaxMembers StickySession DisableFailover Timeout FailoverAttempts Morker URL Route RouteRedir Factor Set Status Elected Busy Load. To From HC Method HC Interval Passes Fails HC uri HC Expr http://12.16.01 1.00 0.1mt Ck, 251 0.200 49K 14K NONE 30000ms 1(0) 1(0)	Load Ba	ancer	Ma	nag	jer	fo	r 10	.10) <mark>.</mark> 2.	55												
MaxMembers StickySession DisableFailover Timeout FailoverAttempts Method Path Active 2 [2 Used] (None) Off 0 1 byrequests / Yes Worker URL Route RouteRedir Factor Set Status Elected Sussy Load To From HC Method HC Interval Passes Fails HC uri HC Expr http://172.16.0.1 1.00 0 int Ok 2.51 0 -200 49K 14K NONE 30000ms 1 (0) 1 (0) http://172.16.0.2 3.00 0 Int Ok 7.40 200 44K AVN NONE 30000ms 1 (0) 1 (0)	Server Version: Aj Server Built: Jun 2 Balancer changes Balancers are inhe ProxyPass setting LoadBalancer	pache/2.4.51 88 2023 14:1 will NOT be erited from r s are inherite Status for	I (Unix) 1:20 persist main ser ed from balan	ed on r rver. i main s cer://	estart. erver. load2	[p74	1af886	ic_loi	ad2]													
2 [2 Used] (None) Off 0 1 byrequests / Yes Worker URL Route RouteRedir Factor Set Status Elected Busy Load To From HC Method HC Interval Passes Fails HC uri HC Expr http://12.16.0.1 1.00 0 int Ok, 251 0.200 49K 14K NONE 30000ms 1(0) 1(0) http://12.16.0.1 0.00 0 int Ok, 749 0.200 49K 14K NONE 30000ms 1(0) 1(0)	MaxMembers St	ickySession	Disabl	eFailov	er Tin	neout	Failove	Atte	mpts	Me	thod	Path Active										
Worker URL Route RouteRedir Factor Set Status Elected Busy Load To From HC Method HC Interval Passes Fails HC uri HC Expr http://12.16.01 1.00 0 int Ck 251 0 -200 49k 14k NONE 30000ms 1 (0) 1 (0) http://12.16.02 3.00 1 int Ck 274 9 2 Col 14k4 42K NONE 3000ms 1 (0) 1 (0)	2 [2 Used]	(None)		Off		0		1		byred	quests	/ Yes]									
http://172.15.0.1 1.00 0 init Ok 251 0 -200 49K 14K NONE 30000ms 1 (0) 1 (0) http://172.15.0.2 3.00 0 init Ok 749 0 200 14KK 42K NONE 30000ms 1 (0) 1 (0)	Worker URL	Route Rout	eRedir	Factor	Set St	atus I	Elected	Busy	Load	То	From	HC Method	HC Interval	Passes	Fails	HC uri	HC E	xpr				
http://172.16.0.2 3.00 0 Init Ok 749 0 200 144K 42K NONE 30000ms 1 (0) 1 (0)	http://172.16.0.1			1.00	0 In	it Ok	251	0	-200	49K	14K	NONE	30000ms	1 (0)	1 (0)							
	http://172.16.0.2			3.00	0 In	it Ok	749	0	200	144K	42K	NONE	30000ms	1 (0)	1 (0)							

图 5-5-5 负载均衡性能监控页面

监控信息说明见表 5-5-5。

表 5-5-5 监控信息说明

字段	字段名	功能
1	Server Version	网站服务器当前版本信息
2	Server Built	服务器创建时间



续表

字段	字段名	功能
3	MaxMembers	负载均衡节点最大数量及使用数量
4	StickySession	粘滞会话设置信息
5	DisableFailover	禁用故障转移
6	Timeout	超时时间
7	FailoverAttempts	故障转移尝试次数
8	Method	负载均衡模式
9	Path	负载均衡配置路径
10	Active	是否活跃
11	Worker URL	负载均衡节点地址
12	Route	设置路由信息
13	Route Redirect	路由重定向地址
14	Factor	节点权重信息
15	Set	设置负载均衡节点编号
16	Status	该负载均衡节点状态
17	Elected	向该节点转发的请求数
18	Busy	该节点处于繁忙状态的请求数
19	Load	该节点负载值
20	То	该节点响应的总字节数
21	From	该节点接收的总字节数
22	HC Method	健康状态检查模式
23	HC Interval	健康检查时间间隔
24	Passes	成功的运行状况健康检查次数,默认值为1
25	Fails	失败的运行状况健康检查次数,默认值为1
26	HC uri	设置的信息将附加到 Worker URL 以进行健康检查
27	HC Expr	表达式名称,用于检查响应头健康状态

步骤 10:验证负载均衡对内部业务的容灾性。

关闭 Project-05-Task-03 虚拟机,内部网站业务仅保留网站 Site-Clone-1 正常提供服务。

(1) 在本地主机访问代理服务器的默认网站地址,多次刷新页面将只能看到网站 Site-Clone-1 的内容。

(2) 查看 Apache 负载均衡监控页面,可以看到网站 Site-Clone-2 关闭后,网站 Site-Clone-2 的节点状态为 Init Err,如图 5-5-6 所示,所有的请求都转发到了网站 Site-Clone-1。

				-				使月	月 Ng	inx 实现 [,]	代理月	员务	项目五	
о п Пы	lancer Manager	×	+								-	o x	1	
← C (▲)	不安全 10.10.2.55/	lb-status								as A	୍ର ଜୁ	··· 🜔)	
oad Bal	lancer M	2020	or fo	r 10	10.2	55								
Joau Dai	iancer w	anay	jei iu	10.	. 10.2									
erver Version: Ap	pache/2.4.51 (Uni	x)												
erver Built: Jun Z	28 2023 14:11:20	icted on a	octort											
erver Built: Jun 2 alancer changes alancers are inhe	28 2023 14:11:20 will NOT be pers erited from main	isted on 1 server.	estart.											
ierver Built: Jun 2 Galancer changes Galancers are inhe ProxyPass setting:	28 2023 14:11:20 s will NOT be pers erited from main is are inherited fro	isted on r server. om main s	estart. server.											
Server Built: Jun 2 Balancer changes Balancers are inhu ProxyPass setting: .oadBalancer 1	28 2023 14:11:20 s will NOT be pers erited from main is are inherited fro Status for bala tickySession Disa	isted on r server. om main s ancer://	restart. server. load2 [p7- rer Timeout	4af886 Failove	c_load2]	Method	Path Active							
Server Built: Jun 2 Balancer change 2 Balancers are inhi ProxyPass setting: .oadBalancer 1 MaxMembers St 2 [2 Used]	28 2023 14:11:20 s will NOT be pers erited from main s are inherited fro Status for bala tickySession Disa (None)	isted on r server. om main s ancer:// bleFailov Off	restart. server. load2 [p7· ver Timeout 0	4af886 t Failove	c_load2] rAttempts	Method	Path Active s / Yes							
Server Built: Jun 2 Balancer changes Balancers are inhi ProxyPass setting: LoadBalancer 1 MaxMembers St 2 [2 Used] Worker URL	28 2023 14:11:20 s will NOT be pers erited from main is are inherited fro Status for bala tickySession Disa (None) Route RouteRed	isted on r server. om main s incer:// ibleFailov Off	restart. server. load2 [p7- ver Timeout 0 Set Status	4af886 Failove Elected I	c_load2] rAttempts 1 Busy Load	Method byrequest	Path Active s / Yes n HC Method	HC Interva	Passes F	ails HC uri HC Expr				
Barver Built: Jun 2 Balancer changes Balancers are inhe ProxyPass setting LoadBalancer 1 MaxMembers St 2 [2 Used] Worker URL attp://172.16.0.1	28 2023 14:11:20 will NOT be pers- erited from main is are inherited fro Status for bala tickySession Disa (None) Route RouteRed	isted on r server. om main s nncer:// bbleFailov Off ir Factor 1.00	restart. server. load2 [p7- rer Timeout 0 Set Status 0 Init Ok	4af886 Failove Elected I 261	c_load2] rAttempts 1 Busy Load 0 -100	Method byrequest To Fror 55K 14k	Path Active s / Yes n HC Method	HC Interva 30000ms	I Passes F 1 (0) 1	ails HC uri HC Expr (0)	r			

通过验证证实负载均衡业务服务正常,在内部网站出现故障后,所有用户请求都将转发到正常 服务的网站服务器上,负载均衡对内部业务有一定的容灾性,可有效地提升业务服务的可靠性。





图 5-5-6 负载均衡性能监控页面

项目六 使用 MySQL Server 实现 数据库服务

任务四 使用 Navicat Monitor 监控 MySQL



【任务介绍】

Navicat Monitor 是一套安全、简单而且无代理的远程服务器监控工具。支持的数据库有 MySQL、MariaDB、PostgreSQL 和 SQL Server,并与 Amazon RDS、Amazon Aurora、Oracle Cloud、 Microsoft Azure、阿里云、腾讯云和华为云等云数据库兼容。Navicat Monitor 包含一组丰富的实时 和历史图表,包括其可用性、磁盘使用率、网络 I/O、表锁等。

本任务通过部署 Navicat Monitor,实现 MySQL 数据库基础结构运行状况的监控与故障排除, 提高数据库的性能。

【任务目标】

- (1) 掌握 Navicat Monitor 的部署。
- (2) 掌握 MySQL 的监控与管理。

【操作步骤】

步骤 1: 安装 Navicat Monitor。

Navicat Monitor 支持 Windows、macOS、macOS Homebrew、Linux、Linux Repos、Docker、 FreeBSD 等。本项目选用面向 Windows 平台的 3.0.1 版本,其下载地址为 https://www.navicat.com.cn/ download/navicat-monitor。

(1) 双击启动安装程序,进入安装欢迎页后单击"下一步(N)"按钮,如图 6-4-1 所示。

_____ 使用 MySQL Server 实现数据库服务

(2)选中"我接受协议(A)"前的复选框,单击"下一步(N)"按钮;设置"选择安装位置", 单击"下一步(N)"按钮;设置"选择开始菜单文件夹",单击"下一步(N)"按钮;最后选择开始 安装,安装完成如图 6-4-2 所示。

Navicat Monitor	次迎使用PremiumSoft Navicat Monitor安装向导 本向导将在您的电频上安康FremiumSoft Navicat Monitor 3.0 建议您在继续之前关闭其他所有应用程序。 点击"下一步"继续,或"取消"通出安顿同导。	Navicat Monitor	PremiumSoft Navicat Monitor安装 完成 安康同号三空恋的車幅上安康PremiumSoft Navicat Monitor、可以通过已爱娘的快捷方式來打开死应用種序。 点击 "結束"通出安義。 「 懂行NavicatMonitorManager.exe
	下 一步 (X) 取消		鋪束(<u>E</u>)

图 6-4-1 Navicat Monitor 安装向导

图 6-4-2 安装完成

(3)选中"运行 NavicatMonitorManager.exe"前的复选框,单击"结束(F)"按钮以启动 Navicat Monitor,自动跳转至浏览器进行初始化配置,完成后单击"Next>"按钮,如图 6-4-3 所示。

Initial Settings Superuser Profile		
USERNAME *	FULL NAME *	
mysql	mysql	
PASSWORD * 🚱	PREFERRED LANGUAGE	
	简体中文 🔻	
CONFIRM PASSWORD *	EMAIL 🔞	
	MOBILE 🚱	
	+1 *	
	K Next	

(4) 配置外观,单击"Next>"按钮;配置监控数据的存储库数据库,完成后单击"Next>"按钮,如图 6-4-4 所示。

项目六

19

Navicat N	Monitor	
	Initial Settings	
	Repository Database	
	DATABASE TYPE 🕜	DATABASE NAME *
	MySQL -	monitor
	HOST NAME *	USERNAME *
	10.10.2.61	root
	PORT*	PASSWORD
	3306	
		Back Test Connection Next >

图 6-4-4 配置存储库数据库

(5) 配置应用信息,完成后单击"Next>"按钮,如图 6-4-5 所示。

Navicat M	onitor			
	Initial Settings			
	Application			
	HOST NAME		IP ADDRESS	
	DESKTOP-707GFIE		0.0.0.0	
	PORT *		IPV6	
	3000		Enaureu	
	WEB URL *		DNS SERVER 114,114,114,114	
	http://127.0.0.1:3000			
		图 6-4-5	配 置应田信自	
		ы 0-4- 3		
- 20				

🙌 Navicat Monito Initial Settings Confirmation PORT 3000 MySQL mysql HOST NAME 10.10.2.61 WEB UR http://127.0.0.1:3000 PORT 3306 IP ADDRI 0.0.0.0 < Back

(6)确认以上填写信息无误,单击"Finish>"按钮以完成初始化配置,如图 6-4-6 所示。

图 6-4-6 确认信息

- Navicat Monitor

 N
 Preparing your registery database..

 Prese wat for the initial configuration of Navicat Monitor. This may take a few minutes.
 Do not turn of your computer.

 N
 Ext (A A 7 → 20146/4 L K2 H4
- (7)进行初始化应用并准备存储数据库,如图 6-4-7 所示。



(8) 进入 Navicat Monitor 登录界面,则表示安装成功,设置语言为"简体中文"并输入用户

21

名与密码进行登录,如图 6-4-8 所示。

Navicat Monitor		
	注册成功日	语言: 徽体中文 👻
	登录并立即开始使用	
	Navicat Monitor 3.0	
	用户名	
	mysql	
	密码	
	🖌 保持登录状态	
	登录	

图 6-4-8 登录界面

(9)登录成功后,进入首页如图 6-4-9 所示。

		試用將在14天后过期		数活令牌 购买
₩ 概览 警报 查询分析	器 SQL 性能分析工具 复制	和置		🌲 🚺 mysql 🔻
+新建实例 Q. 搜索实例				
查看: 组 数据库关型 展干 卡片设计	f =		н	没有开启的警报
第选: 🗌 🚺 MySQL 🗌 🛃 MariaDe	🛛 🔛 PostgreSQL 🔄 🎦 SQL Sen	ver	排序方式:警报严重性 »	查看全部历史记录 >
	没有实例			

图 6-4-9 首页

步骤 2: 创建 MySQL 监控实例。

(1) 单击左上角的 "+新建实例" 按钮,选择 "MySQL",并新建 MySQL 实例,单击"新建" 按钮,如图 6-4-10 所示。



				0			
			_ (使用 MySQI	_ Server §	实 现 敛 据 库 	5 坝日7	V
-NN 105127 12.5	8 查询公标架 501 性		RUME 14 JUNUUR				
+ REEPT-04	0 1992-94						
· NASOL							
MariaDB	新建 MySQL 实例				×		
SQL Server	实则名*	(東例名(最多 25 个字符)			·史记录 >		
Amazon AW	相	Production	· +	新建组			
Oracle Cloud			0				
副 副 副 副 副 国 里 云		Navicat Monitor	数据库				
 	(1) 网关服务器						
	 通过 SSH 随道连接到的 MySQL 服务器 	《天服労器					
	主机名 *	localhost					
	56D *	3306					
	用户名。	root					
	密码	8569	۲				
	服务器类型	● 类 Unix 系统 (Windows				
	 使用 SSL 验证 CPU 和内存 						
	☑ 登录 SSH 以伪间 CPU	和內存的数据					
	主机名 +	localhost					
	第日*	22					
	用户名 *	用户名					
	物证方法	燃码	Ψ.				
	200		۲				山
			RX.	四 数试生接	W RZ		*
	图 6-4	4-10 创致	建 MySQL 监	控实例			
(2) 创建成功如图 6-4	-11 所示。						
N saritér station	青海公析実 くつい 純粋		新在 14 大后过期				
1 4 100 A BELLEVILLE	0 1949/2/64	MARTING 08(0)	P-0-1-ana				
62. 40 wereast	WE 44404 -				· .		

Eacon)



鐵造: 🗹 💽 MySQL

CPU 100 %

Production (1)
 MySQL-10.10.2.6

步骤 3: 查看 MySQL 监控。

(1)单击"新建的 MySQL 实例",查看该实例监控的概览,如图 6-4-12 所示。





24

_____ 使用 MySQL Server 实现数据库服务

(2)单击横向导航中的"警报"选项卡以查看警报,其高级筛选支持筛选实例类型、警报类型等,如图 6-4-13 所示。

		試用將在14天后过	朔	激活《	# 购买令牌
₩ 概览 警报 查询分析器	SQL 性能分析工具	复制 配置		÷ ທ n	nysql ▼ #₽83
全部实例 ▼ 管服					
Q. 搜索实例 ←	警报历史记录				
▼ 全部实例	第选:开启 ▼ 高级铸造 ▼				
Production	实例类型	警报类型			
MySQL-10.10.2.61	All Instance Types 📼	选择警报英型或输	↓美韻字 ▼		
🔻 🛅 Testing	开始时间	严重性	状态	厩性	
	● 全部	一戸重	☑ 开启	未读	
	○ 最近:30分钟 ▼	86	□ 已关闭 □ 已结束		
	O 从 2023-10-11.1			□ 有备注 □ 没有第记	
	2023+10+11, 1			□ 受理人:任何人 ▼	
				未分配	
	应用筛选 另存为自知	主义新选 隐藏新选			
	显示或隐藏列 👻				10/ 3 =
	总数:4个警报				
	□ ▼ 类型 ⇒		ID 💠 案例 🗧 严重性	○ 秋恋 ○ 开始时间 🜩	结束时间 🔅
	CPU 使用率		4 NySQL-1 严重	开启 下午02:26	尚未結束
	CPU 使用率		3 NySQL-1 習管	开启 下午02:26	尚未結束
	MySQL 的密码	內策略	2 NySQL-1 警告	开启 下午02:22	尚未結束
	- 显示数据库权	限	1 NySQL-1 警告	开启 下午02:22	海米結束
					÷

图 6-4-13 查看警报

(3)单击横向导航中的"查询分析器"选项卡,选择步骤一创建的实例,以查看"查询分析器",如图 6-4-14 所示。

▶ 概览 警报 查询分析器	SQL 性能分析工具	复制 配置			🚺 mysql 🐨	秘密
lySQL-10.10.2.61 ▼ 查询分析器						н
费财查询 显示最后 12小时 =						
模計查询	5					
		15	日本指定时间在期的数据			
	03.4M 04.4M 0	05 AM 06 AM 07 AM	08 AM 09 AM 10	AM 11.444 12.PM 01.PM 02.PM		
	03 AM 04 AM 0	25 AM 06 AM 07 AM	05 AM 09 AM 10	АМ 11 АМ 12 РМ 01 РМ 02 РМ		
副游的污动力功	03 844 0 04 844 0	25 AM 06 AM 07 AM	08 AM 09 AM 10 原有全部	AM 11 AM 12 PM 01 PM 02 PM 記録記録載 (月、12 个語句)	查看全	15
8360/F80:039	03.444 04.444 C	05 AM 06 AM 07 AM	05 AM 09 AM 10 日前全部	AM 11-MA 122PM 01-PM 02-PM 通程的時度(共 12 小进程) 4024 — Steep uuer11g1 n12050079 安田市・	查看全江	0 s
副面的环境的合同	азим оним с	азілан орілан азгілан	09 AM 09 AM 10 京市士市	AM 11.AM 12.PM 01.PM 02.PM 連載的意味(11.0.12.1.550.00179 取扱い) 4324 — Skep undf1101.0.2.1.050.00179 取扱い。 4357 — Execute undf1101.0.2.0.00528 取扱い。monter	查看全	0 s 0 s
最新的所能性态为	10.00 (0.000)	55 AM 00 AM 07 AM	os Xan 09 Xan 10 京田士元	AM 11 AM 12 PM 07 PM 02 PM 超程が発意(見 12 个/研究) 4324 — Sleep user10 (10.10.2.100.50079 数数度): 4357 — Exctel user10 (10.10.2.100.50725 数数度): monitor user10 (10.10.2.100.50727 数数度): 4058 — Query user10 (10.10.50027 数数度):	會單全	85 0 s 0 s 0 s
40.0609/580-0-30	100,000 (04,000) (04,000 (04,000) (04,000 (04,000) (04,000) (04,000) (04,000) (04,000) (04,000 (04,000) (ts kan to kan to kan	897AM 097AM 10 息司순部	AM 11.AA 127M 017M 027M	重霉全的	0 s 0 s 0 s 0 s

项目六

- Office -

	拍 5 个总由间层长的查询					政國		总时间
	SHOW SLAVE STATUS					2422		37.7
	SELECT 'EVENT_NAME' , 'COUNT_STAR' , 'SU	M_TIMER_WAIT' FRO	OM 'performance	_schema' .	'events_waits_sur	n 2411		33.76
J.	SHOW GLOBAL STATUS					6358		22.97
	SELECT 'performance_schema' . 'events_sta	tements_summary_	by_digest" . 'QUEF	RY_SAMPLE	_TEXT' . 'performa	a 4822		17.66
	SELECT COLUMN_NAME FROM 'Information	_schema" . "columns	s' WHERE 'table_s	chema' = ?	AND TABLE_NAM	E 4821		12.55
2 接衆査询			皇示或國	M91 - 10	1 - 10 行, (共 507 i	行) 10/西	< 1 2 3	4 5 6 51)
19月 (1)		1998 0	查询发生率,	÷	时间总计	量大財詞 =	平均时间 MS 💿	最后发生 🐘
ELECT @@'version'		7633		9.29	1.86	0.0697	0.2434	7月17日上午08:20
HOW GLOBAL STATUS		6358		7.74	22.97	2.32	3.61	7月17日上午08:20
ELECT 'performance_schema' . `	events_statements_summary_by_digest'	4822		5.87	17.66	0.1529	3.66	7月17日上午08:20
ELECT COLUMN_NAME FROM 'Ir	formation_schema' . 'columns' WHERE '	4821	1.	5.87	12.55	0.1443	2.6	7月17日上午08:20
ELECT 'conn_status' . 'channel_n	ame' AS 'channel_name' , 'conn_status'	4821		5.87	2.04	0.0335	0.424	7月17日上午08:20
ELECT NAME , 'subsystem' , TYPI	E, COMMENT, 'count' FROM 'informatio	4821		5.87	8.23	0.2503	1.71	7月17日上午08:20
ELECT "CHANNEL_NAME" AS "cha	annel_name", 'MEMBER_ID' AS 'member	4821		5.87	2.26	0.1032	0.4684	7月17日上午08:20
ELECT "CHANNEL_NAME" AS "cha	innel_name' , 'MEMBER_ID' AS 'member	4821		5.87	1.98	0.0084	0.4102	7月17日上午08:20
ELECT COUNT_TRANSACTIONS	IN_QUEUE' AS 'transactions_in_queue' , '	4821		5.87	1.83	0.0199	0.379	7月17日上午08:20
LECT COUNT TRANSACTIONS	IN OUFUE' AS 'transactions in queue'	4821		5.87	1.7	0.0263	0.3519	7月17日上午08:20

图 6-4-14 (二) 查询分析器

项目六

项目七 使用 MongoDB 实现数据库服务

任务四 监控 MongoDB



【任务介绍】

监控是数据库管理的重要内容之一,通过监控可以实时了解数据库的运行状态,及时发现系统 异常,有效地避免数据库故障,保障业务顺利可用。本任务通过云服务、实用工具和内置命令实现 MongoDB 的监控。

本任务在任务三的基础上进行(虚拟机使用的服务器是 Project-07-Task-02)。

【任务目标】

- (1) 实现使用 mongostat 监控 MongoDB 的运行状态。
- (2) 实现使用 mongotop 获取 MongoDB 的运行性能。
- (3) 实现使用 MongoDB Web Console 监控 MongoDB。
- (4) 实现使用 MongoDB Cloud Manager 监控 MongoDB。

【操作步骤】

步骤 1: 通过 MongoDB Database Tools 监控 MongoDB。

1. 通过 RPM 包安装 MongoDB Database Tools

MongoDB Database Tools 用于管理和监控 MongoDB 的命令行实用程序的集合。

通过 wget 工具下载 MongoDB Database Tools 的 RPM 包,并进行安装。本任务的使用版本为 100.8.0。

操作命令:

- 1. # 通过 wget 下载 mongodb-database-tools-100.8.0
- [root@Project-07-Task-02 ~]# wget https://fastdl.mongodb.org/tools/db/mongodb-database-tools-rhel80-x86_ 64-100.8.0.rpm
- 3. --2023-09-05 06:43:13-- https://fastdl.mongodb.org/tools/db/mongodb-database-tools-rhel80-x86_64-100.8.0.
- 4. Resolving fastdl.mongodb.org (fastdl.mongodb.org)... 54.192.18.79, 54.192.18.31, 54.192.18.99, ...
- 5. Connecting to fastdl.mongodb.org (fastdl.mongodb.org)|54.192.18.79|:443... connected.
- 6. HTTP request sent, awaiting response... 200 OK
- 7. Length: 53745748 (51M) [binary/octet-stream]
- 8. Saving to: 'mongodb-database-tools-rhel80-x86_64-100.8.0.rpm'
- 10. 2023-09-05 06:43:30 (10.0 MB/s) 'mongodb-database-tools-rhel80-x86_64-100.8.0.rpm' saved [5374574 8/53745748]
- 11. # 通过 RPM 命令进行安装
- 12. [root@Project-07-Task-02 ~]# rpm -ivh mongodb-database-tools-rhel80-x86_64-100.8.0.rpm
- warning: mongodb-database-tools-rhel80-x86_64-100.8.0.rpm: Header V3 RSA/SHA256 Signature, key I D 90cfb1f5: NOKEY

####################################[100%]

#################################[100%]

16. Updating / installing...

15. Preparing ...

返目七

17. 1:mongodb-database-tools-100.8.0-1

操作命令+配置文件+脚本程序+结束

MongoDB Database Tools 集合包含了多种类型的工具,具体见表 7-4-1。

表 7-4-1 MongoDB Database Tools 工具列表

名称	类型	描述
mongodump	Binary Import / Export	以二进制格式导出数据
mongorestore	Binary Import / Export	将 mongodump 导出的数据进行转储还原
bsondump	Binary Import / Export	将 BSON 格式转换为 JSON
mongoimport	Data Import / Export	导入数据,支持 JSON、CSV 或 TSV 格式
mongoexport	Data Import / Export	导出数据,支持 JSON、CSV 格式
mongostat	Diagnostic Tools	查看单实例或集群的状态
mongotop	Diagnostic Tools	查看实例的读写耗时
mongofiles	GridFS Tools	从其他成员来看,该成员不可访问

2. mongostat

28

mongostat 工具根据数据库的操作类型(例如插入、查询、更新、删除等)捕获并返回统计报告服务器上的负载分布。

使用 MongoDB 实现数据库服务

项目七

命令详解:	mongostat
-------	-----------

【语法】	
mongostat [options]	
[options]	
host= <hostname><:port>:</hostname>	数据库服务器主机地址,可使用半角逗号隔开指定多个
port= <port>:</port>	数据库服务器主端口号
username= <username>, -u=<username>:</username></username>	数据库连接用户名
password= <password>, -p=<password>:</password></password>	数据库连接密码
authenticationDatabase= <dbname>:</dbname>	认证数据库
uri= <connectionstring>:</connectionstring>	使用连接字符串格式连接,但不可与认证字段同时使用
humanReadable= <boolean>:</boolean>	格式化输出日期等字段
-O= <field list="">:</field>	指定输出字段和字段名称,可使用半角逗号隔开指定多个
json:	以 JSON 格式输出信息
<sleeptime>:</sleeptime>	执行 mongostat 命令执行的间隔时间
	操作命令+配置文件+脚本程序+结束

操作命令:

1.	# 使用	mongosta	ut 监控单个	` mongod :	示例			
2.	[root@P	roject-07	-Task-02	~]# mongo	ostatuser	name=admi	npassv	vord=openeuler@mongodb#123authe
	nticatior	nDatabase	e=admin -o	o=insert,qu	ery,update,	delete,net_ir	n,net_out,	conn,repl
3.	insert	query	update	delete	net_in	net_out	conn	repl
4.	*0	*0	*0	*0	1.68k	66.5k	17	SLV
5.	*0	*0	*0	*0	339b	62.8k	17	SLV
6.	*0	*0	*0	*0	2.29k	64.9k	17	SLV
								操作命令+配置文件+脚本程序+结束

mongostat 命令结果的字段说明如下。

- insert: 检测磁盘设备名称
- query: 每秒的执行查询操作数
- update: 每秒的执行更新操作数
- delete: 每秒的执行删除操作数
- getmore: 每秒获取更多(即游标批处理)操作数
- command: 每秒的执行操作数



- ▶ vsize:进程使用的虚拟内存量(以 MB 为单位)
- res: 进程使用的驻留内存量(以 MB 为单位)
- locked: 全局写锁定中的时间百分比
- qr: 等待读数据的客户端数
- qw: 等待写数据的客户端数
- ar: 执行读数据的客户端数
- aw: 执行写数据的客户端数
- net in: MongoDB 实例接收的网络流量(以字节为单位)

る 日 万 日 七

- net_out: MongoDB 实例发送的网络流量(以字节为单位)
- conn: 数据库连接数
- set: 副本集的名称
- repl: 副本集成员状态
- 3. mongotop

mongotop 提供一种跟踪 MongoDB 实例 mongod 在读写数据上的时间消耗,提供每个集合级别的统计信息。默认情况下,mongotop 每秒返回一次值。

命令详解: mongotop

【语法】	
mongotop [options]	
[options]	
host= <hostname><:port>:</hostname>	数据库服务器主机地址, 可使用半角逗号隔开指定多个
port= <port>:</port>	数据库服务器主端口号
username= <username>,-u=<username>:</username></username>	数据库连接用户名
password= <password>,-p=<password>:</password></password>	数据库连接密码
authenticationDatabase= <dbname>:</dbname>	认证数据库
uri= <connectionstring>:</connectionstring>	使用连接字符串格式连接,但不可与认证字段同时使用
humanReadable= <boolean>:</boolean>	格式化输出日期等字段
-O= <field list="">:</field>	指定输出字段和字段名称,可使用半角逗号隔开指定多个
json:	以 JSON 格式输出信息
<sleeptime>:</sleeptime>	执行 mongostat 命令执行的间隔时间
	操作命令+配置文件+脚本程序+结束

操作命令:

30

1.	# 使用 mongotop 监控单个 mongod 示例					
2.	[root@Project-07-Task-02 ~]# mongotopusername=adminpassword=openeuler@mongodb#123author					
	nticationDatabase=admin					
3.	2023-09-05T09:06:56.410+0800	connected	to: mor	ngodb://loc	calhost/	
4.						
5.	ns	total	read	write	2023-09-05T09:06:57+08:00	
6.	admin.system.keys	0ms	0ms	0ms		
7.	admin.system.users	0ms	0ms	0ms		
8.	admin.system.version	0ms	0ms	0ms		
9.	admin.test_collection	0ms	0ms	0ms		
10.	config.collections	0ms	0ms	0ms		
11.	config.external_validation_keys	0ms	0ms	0ms		
12.	config.image_collection	0ms	0ms	0ms		
13.	config.settings	0ms	0ms	0ms		
14.	config.system.sessions	0ms	0ms	0ms		
15.	config.tenantMigrationDonors	0ms	0ms	0ms		
					操作命令+配置文件+脚本程序+结束	

项目七

」 四

使用 MongoDB 实现数据库服务

mongotop 命令结果的字段说明如下。

- ns: 数据库命名空间
- db: 数据库的名称

小贴士

【小贴士】

- total: 在此命名空间上操作所消耗的总时长
- read: 在此名称空间上执行读操作所消耗的时间
- write: 在此名称空间上执行写操作所消耗的时间
- <timestamp>: 检测的有效时间

步骤 2: 使用 MongoDB Database Methods 监控 MongoDB。

Database Methods(数据库方法)是 mongosh Methods(Mongo Shell 方法)内置方法中的一类。 数据库方法用于管理和监控 MongoDB 实例或集群的方法集。

- (1) mongosh Methods 包含的方法类型说明如下。
- Collection: 集合方法集
- Cursor: 游标方法集
- Database: 数据库方法集
- Query Plan Cache: 查询计划缓存方法集
- Bulk Write Operation: 批量写操作方法集
- User Management: 用户管理方法集
- Role Management: 角色管理方法集
- Replication: 复制方法集
- Sharding: 分片方法集
- Constructors: 构造函数方法集
- Connection: 连接方法集
- Client-Side Field Level Encryption: 客户端字段级加密方法集
- Atlas Search Index Methods:图谱搜索索引方法集

(2) Database Methods 包含的方法类型说明如下

- db.cloneDatabase():从指定主机上克隆数据库
- db.currentOp(): 显示当前正在进行的操作
- db.commandHelp(): 返回数据库命令的帮助信息
- db.createCollection(): 创建一个聚集集合(table)
- db.cloneCollection(): 在 MongoDB 库实例之间复制集合数据
- db.copyDatabase():从指定的机器上复制指定的数据库数据到某个数据库
- db.dropDatabase(): 删除当前使用的数据库
- db.fsyncLock():刷新写入磁盘并锁定该数据库,以防止写入操作,并协助 备份操作



31

项目七

- db.fsyncUnlock(): 允许继续进行写入锁住的数据库(解锁)
- db.getLogComponents(): 返回日志消息详细级别
- db.getLastErrorObj(): 查看完整的错误结果
- db.getMongo(): 查看当前数据库的链接机器地址
- db.getCollection(): 得到指定名称的聚集集合(table)
- db.getName(): 查看当前使用的数据库
- db.getPrevError(): 返回包含自上次错误复位所有的错误状态文件
- db.getCollectionNames(): 得到当前数据库的所有聚集集合
- db.getCollectionInfos(): 返回当前数据库中的所有集合信息
- db.getLastError(): 返回上一次错误,如果没有错误则为空
- db.hostInfo(): 返回当前数据库主机系统的相关信息
- db.killOp():终止指定的操作
- db.listCommands():显示公共数据库的命令列表
- db.logout(): 注销登录
- db.printCollectionStats(): 显示当前数据库所有聚集索引的状态
- db.resetError(): 重置数据库
- getPrevError()和 getPrevError: 返回的错误信息
- db.repairDatabase(): 修复当前数据库
- db.runCommand(): 运行一个数据库命令
- db.serverStatus(): 返回当前数据库状态的概要
- db.setProfilingLevel(): 修改当前数据库的分析级别
- db.stats():显示当前数据库状态
- db.shutdownServer():关闭当前数据库运行实例或安全停止有关操作进程
- db.setLogLevel(): 设置一个单独的日志信息级别
- db.version(): 查看当前数据库版本
- 1. db.serverStatus()

db.serverStatus()返回数据库状态的概述信息,包括连接信息、选举信息、流量控制、数据库锁 信息、网络流量、操作延迟情况、操作统计、内存使用情况等信息。

操作命令:

返目七

- 1. # 使用 mongotop 监控单个 mongod 示例
- 2. [root@Project-07-Task-02 ~]# mongosh
- 3. Current Mongosh Log ID: 64f6829fdd88c4c0bd042bf7
- 4. Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2 000&appName=mongosh+1.9.1
- 5. Using MongoDB: 6.0.8
- 6. Using Mongosh: 1.9.1



项目七 7. For mongosh info see: https://docs.mongodb.com/mongodb-shell/ 8. db-cluster-mongodb [direct: primary] test> use admin; 9. switched to db admin 10. db-cluster-mongodb [direct: primary] admin> db.auth("admin","openeuler@mongodb#123"); 11. $\{ ok: 1 \}$ 12. db-cluster-mongodb [direct: primary] admin> db.serverStatus() 13. { 14. host: 'Project-07-Task-02', 15. version: '6.0.8', 16. process: 'mongod', 17. pid: Long("3433"), 18. uptime: 19610, uptimeMillis: Long("19609809"), 19. 20. uptimeEstimate: Long("19609"), 21. localTime: ISODate("2023-09-05T01:39:08.831Z"), 22. activeIndexBuilds: { 23. total: 0, 24. phases: { 25. scanCollection: 0, 26. drainSideWritesTable: 0. 27. drainSideWritesTablePreCommit: 0, 28. waitForCommitQuorum: 0, 29. drainSideWritesTableOnCommit: 0, 项目七 30. processConstraintsViolatonTableOnCommit: 0, 31. commit: 0 32. } 33. }, 34. asserts: { 35. regular: 0, 36. warning: 0, 37. msg: 0, 38. user: 579, 39. tripwire: 0, 40. rollovers: 0 41. }, 42. batchedDeletes: { batches: 0, docs: 0, stagedSizeBytes: 0, timeMillis: Long("0") }, 43. catalogStats: { collections: 2, 44. 45. capped: 0, 46. clustered: 0, 47. timeseries: 0, 48. views: 0, internalCollections: 21, 49. 50. internalViews: 1 51. }, 52. connections: {



```
53.
         current: 16,
54.
         available: 51184,
55.
         totalCreated: 149,
56.
         active: 6,
57.
         threaded: 16,
         exhaustIsMaster: 0,
58.
59.
         exhaustHello: 3,
60.
         awaitingTopologyChanges: 4
61.
       },
62. # 为了排版方便,以下输出信息中省略了部分信息
63. ok: 1,
64.
       '$clusterTime': {
65.
         clusterTime: Timestamp({ t: 1693877940, i: 1 }),
66.
         signature: {
           hash: Binary(Buffer.from("1a4c66a4b10060e0ed1fb67ab4e85fedf66ef474", "hex"), 0),
67.
68.
           keyId: Long("7275017387823333381")
69.
        }
70.
    },
71.
       operationTime: Timestamp({ t: 1693877940, i: 1 })
72. }
                                                                        操作命令+配置文件+脚本程序+结束
```

2. db.stats()

db.stats()可用于查看单个数据库使用状态的统计信息。

操作命令:

1.	# 使用 db.stats()查看信息
2.	[root@Project-07-Task-02 ~]# mongosh
3.	Current Mongosh Log ID: 65103882ec88e4d630ef2985
4.	Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=
	2000&appName=mongosh+1.9.1
5.	Using MongoDB: 6.0.8
6.	Using Mongosh: 1.9.1
7.	
8.	For mongosh info see: https://docs.mongodb.com/mongodb-shell/
9.	
10.	db-cluster-mongodb [direct: secondary] test> use admin;
11.	switched to db admin
12.	db-cluster-mongodb [direct: secondary] admin> db.auth("admin","openeuler@mongodb#123");
13.	{ ok: 1 }
14.	db-cluster-mongodb [direct: secondary] admin> db.stats()
15.	{
16.	db: 'admin',
17.	collections: 4,
18.	views: 0,
19.	objects: 9,
34	

项目七

项目七 20. avgObjSize: 293.77777777777777, 21. dataSize: 2644, 22. storageSize: 135168, 23. indexes: 5, 24. indexSize: 163840, 25. totalSize: 299008, 26. scaleFactor: 1, 27. fsUsedSize: 4188913664, 28. fsTotalSize: 17782554624, 29. ok: 1, 30. '\$clusterTime': { 31. clusterTime: Timestamp({ t: 1695559310, i: 1 }), 32. signature: { hash: Binary(Buffer.from("173339425d89ae415445eb884e928129228e783e", "hex"), 0), 33. 34. keyId: Long("7275017387823333381") 35. } 36. }, operationTime: Timestamp({ t: 1695559310, i: 1 }) 37. 38. } 39. db-cluster-mongodb [direct: secondary] admin> 操作命令+配置文件+脚本程序+结束

db.stats()方法结果的字段说明如下。

- db: 数据库的名称
- collections: 数据库中的集合数
- views: 数据库中的视图数
- objects: 数据库中对象(即 document)数
- avgObjSize: 每个文档的平均大小(以字节为单位)
- ▶ dataSize: 数据库中保存的未压缩数据的总大小
- storageSize: 分配给数据库中用于文档存储的所有集合的磁盘空间总和

▶ freeStorageSize: 分配给数据库中用于文档存储的所有集合的可用空间的总和

- indexes: 数据库中所有集合的索引总数
- ▶ indexSize: 数据库中所有集合的索引总数
- indexFreeStorageSize: 分配给数据库中所有索引的磁盘空间总和,包括可用索引空间
- totalSize: storageSize 与 indexSize 的总和
- totalFreeStorageSize: freeStorageSize 与 indexFreeStorageSize 的总和
- fsUsedSize: MongoDB 所在的文件系统上使用的所有磁盘空间的总大小





项目七

3. db.collection.collStats()

该方法返回指定集合的存储统计信息。

操作命令:

1.	# 使用 db.collection.collStats()查看信息
2.	[root@Project-07-Task-02 ~]# mongosh
3.	Current Mongosh Log ID: 65103d4963d275001c750663
4.	Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2
	000&appName=mongosh+1.9.1
5.	Using MongoDB: 6.0.8
6.	Using Mongosh: 1.9.1
7.	
8.	For mongosh info see: https://docs.mongodb.com/mongodb-shell/
9.	
10.	db-cluster-mongodb [direct: primary] test> use admin;
11.	switched to db admin
12.	db-cluster-mongodb [direct: primary] admin> db.auth("admin","openeuler@mongodb#123");
13.	{ ok: 1 }
14.	db-cluster-mongodb [direct: primary] admin> db.test_collection.stats()
15.	{
16.	ok: 1,
17.	capped: false,
18.	wiredTiger: {
19.	metadata: { formatVersion: 1 },
20.	creationString: 'access_pattern_hint=none,allocation_size=4KB,app_metadata=(formatVersion=1),assert=
	(commit_timestamp=none,durable_timestamp=none,read_timestamp=none,write_timestamp=off),block_allocat
	ion=best,block_compressor=snappy,cache_resident=false,checksum=on,colgroups=,collator=,columns=,diction
	ary=0,encryption=(keyid=,name=),exclusive=false,extractor=,format=btree,huffman_key=,huffman_value=,ign
	ore_in_memory_cache_size=false,immutable=false,import=(compare_timestamp=oldest_timestamp,enabled=fa
	Ise,file_metadata=,metadata_file=,repair=false),internal_item_max=0,internal_key_max=0,internal_key_trunca
	te=true,internal_page_max=4KB,key_format=q,key_gap=10,leaf_item_max=0,leaf_key_max=0,leaf_page_ma
	x=32KB,leaf_value_max=64MB,log=(enabled=false),lsm=(auto_throttle=true,bloom=true,bloom_bit_count=1
	6,bloom_config=,bloom_hash_count=8,bloom_oldest=false,chunk_count_limit=0,chunk_max=5GB,chunk_siz
	e=10MB,merge_custom=(prefix=,start_generation=0,suffix=),merge_max=15,merge_min=0),memory_page_1
	mage_max=0,memory_page_max=10m,os_cache_dirty_max=0,os_cache_max=0,prefix_compression=false,pr
	enx_compression_min=4,readoniy=raise,source=,split_deepen_min_child=0,split_deepen_per_child=0,split_pc
	t=90,tiered_object=false,tiered_storage=(auth_token=,bucket=,bucket_prefix=,cache_directory=,local_retention
21	=300,name=,object_target_size=0),type=file,value_format=u,verbose=[],write_timestamp_usage=none;
21.	type: Ine,
22.	un: statistics:table:collection-U0598009057/00081459, # 为了排版支通 N下绘山信自由公政了如公信自
23. 24	# /Ŋ J HH队/J 区, 以 广
24.	j, shardadi falsa
25. 26	silatucu. laise,
20.	size. 22,
21.	count. 1,

- 项目七

36

-
			使用 Wongood 头现数1		坝白飞
28.	numOrphanDocs: 0,				
29.	storageSize: 32768,				
30.	totalIndexSize: 32768,				
31.	totalSize: 65536,				
32.	indexSizes: { _id_: 32768	},			
33.	avgObjSize: 22,				
34.	ns: 'admin.test_collection',				
35.	nindexes: 1,				
36.	scaleFactor: 1				
37. }					
			操作	作命令+配置文件+旗	如本程序+结束

db.collection.collStats()方法的结果的字段如下。

db.collection.collStats()方法结果的字段说明如下。

- ns: 集合命名空间
- ▶ size: 内存中所有记录未压缩的存储总量
- timeseries: 集合时序信息
- count: 集合中的对象或文档数
- avgObjSize: 集合中对象或文档的平均存储量
- numOrphanDocs: 集合中独立文档数
- storageSize: 分配给此集合的存储总量
- freeStorageSize: 不适用于内存中的存储量
- nindexes: 集合上的索引数
- indexDetails: 索引的相关信息
- ▶ indexBuilds: 构建的索引

小贴士

- totalIndexSize: 索引总量
- totalSize: storageSize 与 totalIndexSize 的总和
- indexSizes: 集合上每个索引的键和大小
- scaleFactor: 命令使用的比例值
- capped: 集合上限,如果集合有上限,则此字段将为 true
- max: 集合上限中可能存在的文档最大数
- maxSize: 集合上限中可能存在文档的最大存储总量
- wiredTiger: 存储引擎
- inMemory: 正在使用的内存量。

步骤 3: 使用 MongoDB Compass 监控 MongoDB 服务器。

单击 "Performance" 选项卡切换到监控指标选项,在监控指标选项中可查看操作、数据读写、网络、内存、慢查询等实时情况,如图 7-4-1 所示。

项目七

Mongable Company - 10.102,71:27017 Concert Edit Vow Hein	-	n ×
10.10.2.71:27017 My Guerles Databases Performance		
0 My Queries 16(29:40		
B Databases O +		
DEFEATIONS HOTESTICAL ECTIONS		
First@utest.collection	0%	
Tracido		
READ & WRITE		
AREADS ANNTES GREADS CONTINUES		
0 0 0		
NITHORX SLOWEST OPERATIONS		
AND ACCOUNT NOW	0 ms	
Provid	0 ms	
BYTESOUT CONNECTIONS		11
2 70 29		
RAF 00		
Vertual ersteher		
2.59 80m		
>_MONBOSH		^

图 7-4-1 查看性能

MongoDB Compass 监控指标说明,见表 7-4-2。

表 7-4-2 MongoDB Compass 监控指标说明

图表/表格	描述
Operations	显示数据库处理的各类操作数量,与 mongostat 相关
Read & Write	显示活动读取数、排队读取数、活动数写入、排队写入
Network	显示数据库操作的网络性能
Memory	显示内存统计信息
Hottest Collections	显示报告的活动最多的集合。集合对应于图表中的给定时刻;即当在图表上移动时, 相应的集合在表格中突出显示/显示
Slowest Operations	显示最慢的操作。集合对应于图表中的给定时刻;即当在图表上移动时,相应的集 合在表格中突出显示/显示

任务五 使用 Percona Monitoring and Management 监控 MongoDB

【任务介绍】

项目七

Percona Monitoring and Management(简称"PMM")是一款开源数据库监控和管理工具软件,支持 MongoDB、MySQL、MariaDB、PostgreSQL等常用数据库



38

使用 MongoDB 实现数据库服务

项目七

管理系统。PMM 内置监控指标和性能分析模型,并具有可视化监控分析功能。

本任务通过部署 PMM,实现 MongoDB 数据库运行状况的监控与故障排除,提高数据库的性能。

【任务目标】

(1) 实现 PMM 的搭建。

(2) 实现 MongoDB 的监控与管理。

【操作步骤】

步骤 1: 准备 openEuler 服务器。

(1) 创建虚拟机并完成 openEuler 的安装。在 VirtualBox 中创建虚拟机,完成 openEuler 的安装。虚拟机与操作系统的配置信息见表 7-5-1,注意虚拟机网卡的工作模式为桥接。

表 7-5-1	虚拟机与操作系统配置

虚拟机配置	操作系统配置
虚拟机名称: VM-Project-07-Task-05-10.10.2.75	主机名: Project-07-Task-05
内存: 1GB	IP 地址: 10.10.2.75
CPU: 1颗1核心	子网掩码: 255.255.255.0
虚拟硬盘: 20GB	网关: 10.10.2.1
网卡:1块,桥接	DNS: 8.8.8.8

(2)完成虚拟机的主机配置、网络配置及通信测试。启动并登录虚拟机,依据表 7-5-1 完成 主机名和网络的配置,能够访问互联网和本地主机。

是醒

(1)虚拟机的创建、操作系统的安装、主机名与网络的配置,具体方法参见项目一。

(2)建议通过虚拟机复制快速创建所需环境。通过复制创建的虚拟机需依据本任务虚拟机与操作系统规划配置信息设置主机名与网络,实现对互联网和本地主机的访问。

步骤 2: 部署 Docker。

通过 yum 工具在线安装 Docker,本任务使用版本为 18.09.0。

操作命令:

# 通过 yum 岁	F装 Docker				
[root@Project	07-Task-05 ~]# yu	m -y install docker			
Last metadata	expiration check: 0	:02:16 ago on Sat 14	Oct 2023 12:00:12 AM	1 CST.	
	*	0			
Dependencies	resolved.				
Dependencies	resolved.				
Dependencies ====================================	resolved. Architecture	Version	Repository	====== Size	





8.	Installing:					
9.	docker-engine	x86_64	2:18.09.0-323.oe2203sp2	update	39 M	
10.	Installing depende	encies:				
11.	Libcgroup	x86_64	2.0.3-1.oe2203sp2	update	100 k	
12.	libtool-ltdl	x86_64	2.4.7-1.oe2203sp2	OS	33 k	
13.	ta	x86_64	2:1.34-4.0e2203sp2	update	785 k	
14.						
15.	Transaction Sumr	nary				
16.						
17.	Install 4 Packag	es				
18.						
19.	Total download si	ze: 40 M				
20.	Installed size: 166	бM				
21.	Downloading Pac	kages:				
22.	(1/4): libcgroup-2	.0.3-1.0e2203s	sp2.x86_64.rpm	153 kB/s 1	100 kB	00:00
23.	(2/4): libtool-ltdl-	2.4.7-1.0e2203	8sp2.x86_64.rpm	17 kB/s 33	3 kB 00:0	1
24.	(3/4): tar-1.34-4.0	e2203sp2.x86	_64.rpm	105 kB/s 7	785 kB	00:07
25.	(4/4): docker-engi	ne-18.09.0-32	3.oe2203sp2.x86_64.rpm	113 kB/s 3	39 MB05:5	5
26.						
27.	Total			116 kB/s 4	40 MB05:5:	5
28.	Running transacti	on check				
29.	Transaction check	succeeded.				
30.	Running transacti	on test				
31.	Transaction test s	ucceeded.				
32.	Running transacti	on				
33.	Preparing				1/1	
34.						
35.	Verifying	: tar-2:1.34	1-4.0e2203sp2.x86_64		4/4	
36.	# 为了排版方便	此处省略了部	分提示信息			
37.	Installed:					
38.	docker-engine-	2:18.09.0-323.	oe2203sp2.x86_64	li	ibcgroup-2.	0.3-1.oe2203sp2.x86_64
	libtool-ltdl-2.4.7-	1.0e2203sp2.x8	86_64 tar-2:1.34-4.0e	2203sp2.x86_0	64	
39.						
40.	Complete!					
41.	# 查看 Docker 岁	芯态				
42.	[root@Project-07	-Task-05 ~]# s	ystemctl status docker			
43.	 docker.service - 	Docker Appli	cation Container Engine			
44.	Loaded: loa	ded (/usr/lib/s	ystemd/system/docker.service	; enabled; ven	dor preset: o	disabled)
45.	Active: acti	ve (running) si	nce Sat 2023-10-14 00:09:03	CST; 3min 28	s ago	
46.	Docs: htt	ps://docs.dock	er.com			
47.	Main PID: 16	16 (dockerd)				
48.	Tasks: 15	(limit: 2703)				
49.	Memory: 9	9.0M				
50.	CGroup: /s	stem.slice/doc	eker.service			
51.	ŀ	— 1616 /usr/bi	n/dockerdlive-restore			
52.	L	— 1620 contai	nerdconfig /var/run/docker/	containerd/cor	ntainerd.ton	nllog-level info
53.	Oct 14 00:09:00	Project-07-Ta	sk-05 dockerd[1616]: time=	"2023-10-14T	00:09:00.37	75111362+08:00" level=
	warning msg="Fa	iled to cleanup	netns file /var/run/docker/run	ntime-runc: rer	nove /v>	
54.	Oct 14 00:09:01	Project-07-Ta	sk-05 dockerd[1616]: time=	"2023-10-14T	00:09:01.57	'9259958+08:00" level=
	info msg="Defaul	t bridge (dock	er0) is assigned with an IP ad	dress 172.17.0.	.0/16. >	

项目七

使用 MongoDB 实现数据库服务

- 55. Oct 14 00:09:01 Project-07-Task-05 dockerd[1616]: time="2023-10-14T00:09:01.581217039+08:00" level= info msg="Setup IP tables begin"
- 56. Oct 14 00:09:01 Project-07-Task-05 dockerd[1616]: time="2023-10-14T00:09:01.945452797+08:00" level= info msg="Setup IP tables end"
- 57. Oct 14 00:09:02 Project-07-Task-05 dockerd[1616]: time="2023-10-14T00:09:02.243780083+08:00" level= info msg="Loading containers: done."
- 58. Oct 14 00:09:03 Project-07-Task-05 dockerd[1616]: time="2023-10-14T00:09:03.388814637+08:00" level= info msg="Docker daemon" commit=172f8da graphdriver(s)=overlay2 version=18.09.0
- 59. Oct 14 00:09:03 Project-07-Task-05 dockerd[1616]: time="2023-10-14T00:09:03.434029428+08:00" level= info msg="Daemon has completed initialization"
- 60. Oct 14 00:09:03 Project-07-Task-05 dockerd[1616]: time="2023-10-14T00:09:03.860488205+08:00" level= warning msg="Could not register builder git source: failed to find git binary: ex>
- 61. Oct 14 00:09:03 Project-07-Task-05 dockerd[1616]: time="2023-10-14T00:09:03.974121995+08:00" level=I nfo msg="API listen on /var/run/docker.sock"
- 62. Oct 14 00:09:03 Project-07-Task-05 systemd[1]: Started Docker Application Container Engine.
- 63. # 查看 Docker 版本
- 64. [root@Project-07-Task-05 ~]# docker -v
- 65. Docker version 18.09.0, build 172f8da
- 66. # 设置 Docker 自启动
- 67. [root@Project-07-Task-05 ~]# systemctl enable docker

操作命令+配置文件+脚本程序+结束

步骤 3: 部署 PMM Server。

Percona 官方提供 percona/pmm-server 镜像,当前版本为 1.12.6,其官方网站为 https://www.

percona.com/software/database-tools/percona-monitoring-and-management.

操作命令:

- 1. # 拉取 percona/pmm-server 的 Docker 镜像
- 2. [root@Project-07-Task-05 ~]# docker pull percona/pmm-server
- 3. Using default tag: latest
- 4. latest: Pulling from percona/pmm-server
- 5. ccf3a7357f12: Pull complete
- 6. 16ae537e810b: Pull complete
- 7. Digest: sha256:4a4ff78ae4fc42194f77d0361ba03687b24c81dfe5c1ea78bd720fc1167a0f70
- 8. Status: Downloaded newer image for percona/pmm-server:latest
- 9. # 为 pmm-server 创建一个永久数据容器
- 10. [root@Project-07-Task-05 ~]# docker create --volume /srv --name pmm-data percona/pmm-server /bin/tru e
- 11. 2a7c348750714511855967b7d38d95a3ec8e6252289df040e9653fb7f4050d76
- 12. # 运行 pmm-server 镜像
- 13. [root@Project-07-Task-05 ~]# docker run --detach --restart always --publish 443:443 --publish 80:80 --v olumes-from pmm-data --name pmm-server percona/pmm-server
- 14. 6f099192d1c88527e039b6aaaa4fafb93b3ba54d802abc0493e605326436dfb2

操作命令+配置文件+脚本程序+结束

41



运行 PMM Server 镜像后,需等待其自动初始化配置完成,否则将出现"500 Internal Server Error"服务器错误提示。

7

返日

K	×
Percona Mo Manag	onitoring and gement
Email or username email or username	
Password	
	og in
Percona Monitoring and Management Grafana, Prometheus, Clicki	proudly powered by open source projects house, PostgreSQL, and more

完成安装后,使用浏览器中访问服务器 http://10.10.2.75,进入其登录界面,如图 7-5-1 所示。

图 7-5-1 PMM Server 登录界面

初始默认用户密码为 admin/admin,首次登录系统要求修改默认密码,根据提示进行修改完成 后,进入系统如图 7-5-2 所示,说明安装成功。



步骤 4: 部署 PMM Client。

Percona 官方提供 percona/pmm-server 镜像,当前版本为 1.12.6,安装并注册成功后,可在 PMM Server 的设置中的"Nodes"标签页中看到注册节点,如图 7-5-3 所示。

((((PMM Inventory / No	odes				
☆	Services Nodes					
88						
4	Filter				0	
<u>a</u> f	Status Node Name	Node Type	Monitoring	Address	Services	Options
	Ø Up b781f38d932e	container			No services	
¢	O Up pmm-server	generic				
Q	Rows per page: 25 v			Showing 1-2 of	2 items « < 1	> »
5						
0						
Ū						
0						
0						

图 7-5-3 注册节点

操作命令:

- 1. # 拉取 percona/pmm-client 的 Docker 镜像
- 2. [root@Project-07-Task-05 ~]# docker pull percona/pmm-client
- 3. Using default tag: latest
- 4. latest: Pulling from percona/pmm-client
- 5. 9e167103aefa: Pull complete
- 6. 89451b1adfe4: Pull complete
- 7. Digest: sha256:5647f750d04af06460ccd27f59d4b08e3d687fd886c1de23eafa630440e4fec6
- 8. Status: Downloaded newer image for percona/pmm-client:latest
- 9. #为pmm-client创建一个永久数据容器
- 10. [root@Project-07-Task-05 ~]# docker create --volume /srv --name pmm-client-data percona/pmm-client /bin/true
- 11. f34289835c60ea69b7b12e5dbee541ec20d3dcbca1d3ed4c030020a71753d679
- 12. # 运行 pmm-client 镜像并向 pmm-server 注册 pmm-client 节点
- 13. [root@Project-07-Task-05 \sim]# docker run \
- 14. --rm \
- 15. --name pmm-client \setminus
- 16. -e PMM_AGENT_SERVER_ADDRESS=10.10.2.75 \
- 17. -e PMM_AGENT_SERVER_USERNAME=admin \
- 18. -e PMM_AGENT_SERVER_PASSWORD=openEuler@pmm \
- 19. -e PMM AGENT SERVER INSECURE TLS=1 \
- 20. -e PMM_AGENT_SETUP=1 \setminus



项目七

- 21. -e PMM_AGENT_CONFIG_FILE=config/pmm-agent.yaml \
- 22. --volumes-from pmm-client-data \setminus
- 23. percona/pmm-client
- 24. # 当出现以下信息说明注册成功
- 25. time="2023-10-14T09:26:36.516+00:00" level=info msg="Run setup: true Sidecar mode: false" component= entrypoint
- 26. time="2023-10-14T09:26:36.517+00:00" level=info msg="Starting 'pmm-admin setup'..." component= entrypoint
- 27. time="2023-10-14T09:26:36.560+00:00" level=info msg="Loading configuration file /usr/local/percona/ pmm2/config/pmm-agent.yaml." component=setup
- 28. time="2023-10-14T09:26:36.561+00:00" level=info msg="Temporary directory is not configured and will be set to /usr/local/percona/pmm2/tmp" component=setup
- 29. # 为了排版方便,以下输出信息中省略了部分信息
- 30.

返目七

- 31. time="2023-10-14T09:26:36.561+00:00" level=info msg="Updating PMM Server address from \"10.10.2.75\" to \"10.10.2.75:443\"." component=setup
- 32. Checking local pmm-agent status...
- 33. pmm-agent is not running.
- 34. Registering pmm-agent on PMM Server...
- 35. Registered.
- 36. Configuration file /usr/local/percona/pmm2/config/pmm-agent.yaml updated.
- 37. Please start pmm-agent: 'pmm-agent --config-file=/usr/local/percona/pmm2/config/pmm-agent.yaml'.
- 38. time="2023-10-14T09:26:36.848+00:00" level=info msg=""pmm-admin setup' exited with 0" component= entrypoint
- 39. time="2023-10-14T09:26:36.848+00:00" level=info msg="Starting 'pmm-admin run'..." component=entrypoint
- 40. time="2023-10-14T09:26:36.881+00:00" level=info msg="Loading configuration file /usr/local/percona/ pmm2/config/pmm-agent.yaml." component=main
- 41. time="2023-10-14T09:26:36.882+00:00" level=info msg="Using /usr/local/percona/pmm2/exporters/ node_exporter" component=main
- 42. # 为了排版方便,以下输出信息中省略了部分信息
- 43.

- 44. time="2023-10-14T09:26:36.883+00:00" level=info msg="Updating PMM Server address from \"10.10.2.75\" to \"10.10.2.75:443\"." component=main
- 45. time="2023-10-14T09:26:36.883+00:00" level=info msg="Runner capacity set to 32." component=runner
- 46. time="2023-10-14T09:26:36.883+00:00" level=info msg="Window check connection time is 1.00 hour(s)"
- 47. time="2023-10-14T09:26:36.883+00:00" level=info msg=Starting... component=client
- 48. time="2023-10-14T09:26:36.883+00:00" level=info msg="Connecting to https://admin:***@10.10.2.75: 443/ ..." component=client
- 49. time="2023-10-14T09:26:36.885+00:00" level=info msg="Starting local API server on http://127.0.0.1: 7777/ ..." component=local-server/JSON
- 50. time="2023-10-14T09:26:36.889+00:00" level=info msg=Started. component=local-server/JSON
- 51. time="2023-10-14T09:26:36.895+00:00" level=info msg="Connected to 10.10.2.75:443." component=client

使用 MongoDB 实现数据库服务

- 项目七
- 52. time="2023-10-14T09:26:36.895+00:00" level=info msg="Establishing two-way communication channel ..." component=client
- 53. time="2023-10-14T09:26:37.275+00:00" level=info msg="Two-way communication channel established in 379.951194ms. Estimated clock drift: 188.145953ms." component=client
- 54. time="2023-10-14T09:26:38.370+00:00" level=info msg="Starting 2, restarting 0, and stopping 0 agent processes." component=supervisor
- 55. time="2023-10-14T09:26:38.385+00:00" level=info msg="Sending status: STARTING (port 42000)." agentID=/agent_id/2c019ecd-4c29-4f35-bb90-8423d14176cd component=agent-process type=vm_agent

56.

操作命令+配置文件+脚本程序+结束

步骤 5: 配置 MongoDB。

本任务选择监控任务三中的 MongoDB 副本集,在监控前需要对 MongoDB 进行配置,创建具 有创建/还原备份和使用查询分析(QAN)所需权限的自定义角色;使用上述角色以及内置角色创 建/更新数据库用户 clusterMonitor。

操作命令:

1.	[root@Project-07-Task-02~]# mongosh	
2.	Current Mongosh Log ID: 6	51065e3cb1e8b8bbc92f796	
3.	Connecting to:	mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=	
	2000&appName=mongosh+	1.9.1	
4.	Using MongoDB:	6.0.8	
5.	Using Mongosh:	1.9.1	
6.			C
7.	For mongosh info see: https://	//docs.mongodb.com/mongodb-shell/	
8.			
9.	db-cluster-mongodb [direct:	primary] test> use admin;	
10.	switched to db admin		
11.	db-cluster-mongodb [direct:	primary] admin> db.auth("admin","openeuler@mongodb#123");	
12.	{ ok: 1 }		
13.	# 创建具有备份权限的角色		
14.	db-cluster-mongodb [direct:	primary] admin> db.getSiblingDB("admin").createRole({	
15.	role: "explainRole",		
16.	privileges: [{		
17.	resource: {		
18.	db: "",		
19.	collection		
20.	},		
21.	actions: [
22.	"listIndexe	es",	
23.	"listCollec	tions",	
24.	"dbStats",		
25.	"dbHash",		
26.	"collStats"	',	
27.	"find"		



```
28. ...
                  1
29. ...
             }],
30. ...
             roles: []
31. ... });
32. {
33.
       ok: 1,
34.
        '$clusterTime': {
35.
          clusterTime: Timestamp({ t: 1695573530, i: 2 }),
36.
          signature: {
            hash: Binary(Buffer.from("946aac880d7fde127161ad67754207e3c15a5656", "hex"), 0),
37.
38.
            keyId: Long("7275017387823333381")
39.
          }
40.
        },
41.
        operationTime: Timestamp({ t: 1695573530, i: 2 })
42. }
43. # 创建具有 QAN 权限的角色
44. db-cluster-mongodb [direct: primary] admin> db.getSiblingDB("admin").createRole({
45. ... "role": "pbmAnyAction",
46. ... "privileges": [{
47. ... "resource": {
48. ... "anyResource": true
49. ... },
50. ... "actions": [
51. ... "anyAction"
52. ...]
53. ... }],
54. ... "roles": []
55.
     ... });
56.
    {
57.
       ok: 1,
58.
        '$clusterTime': {
59.
          clusterTime: Timestamp({ t: 1695573563, i: 1 }),
60.
          signature: {
61.
            hash: Binary(Buffer.from("5a1324ef6701ae9a807bdf0e98b1a63ecc054143", "hex"), 0),
62.
            keyId: Long("7275017387823333381")
63.
          }
64.
       },
65.
       operationTime: Timestamp({ t: 1695573563, i: 1 })
66. }
67. # 创建 pmm 用户并分配上面创建的角色
68. db-cluster-mongodb [direct: primary] admin> db.getSiblingDB("admin").createUser({
69. ...
             user: "pmm",
70. ...
             pwd: "pmm",
71. ...
             roles: [
                  { role: "explainRole", db: "admin" },
72. ...
73. ...
                  { role: "clusterMonitor", db: "admin" },
74. ...
                  { role: "read", db: "local" },
75. ...
                  { "db" : "admin", "role" : "readWrite", "collection": "" },
76. ...
                 { "db" : "admin", "role" : "backup" },
                 { "db" : "admin", "role" : "clusterMonitor" },
77. ...
  46
```

返目七

78. ... { "db" : "admin", "role" : "restore" }, 79. ... { "db" : "admin", "role" : "pbmAnyAction" } 80. ...] 81. ... }); 82. { 83. ok: 1, '\$clusterTime': { 84. 85. clusterTime: Timestamp({ t: 1695573581, i: 1 }), signature: { 86. hash: Binary(Buffer.from("a02abdf4d6ab164ea3bcce6aaf40226679b9d27a", "hex"), 0), 87. 88. keyId: Long("7275017387823333381") 89. } 90. }, 91. operationTime: Timestamp({ t: 1695573581, i: 1 }) 92. } 操作命令+配置文件+脚本程序+结束

项目七

项目七

47

步骤 6: 添加 Service。

配置数据库服务器后,可以使用用户界面或在命令行上添加 MongoDB 服务。本任务选择前者。 在 PMM Server 的设置中选择 "+Add Service",选择 "MongoDB" 添加 MongoDB 服务。第一步选 择数据库服务类型,如图 7-5-4 所示;第二步配置 MongoDB 服务信息,如图 7-5-5 所示。添加成 功后可在左侧导航中看到 MongoDB 菜单。

C Select service type Select the service type you want to configure and then add it to your Inventory.
Select the service type you want to configure and then add if to your Inventory.
Amazon RDS MySQL
Add a remote instance
MongoDB
Image: Weight of the stance Image: Weight of the stance
Praysol. Material participante Material participante
Conternal Service
0
 @
σ
ⓓ PMM Logs Ŋ Documentation ⓓ Support ಡੇ Community

图 7-5-4 选择服务类型

步骤 7: 查看 PMM 监控。

1. Overview

在 PMM Server 左侧导航选择 MongoDB 项,单击 "Overview"选项卡查看 MongoDB 实例的 基本信息,如图 7-5-6 所示。

	_inux 服务器构建与运维管理	从基础到实战(基于(openEuler)
--	------------------	------------	------------

項目七

48

<u>8</u>	 Inventory / Add service / Step 2 of 2 		Add service
Q	Configuring MongoDB service		
☆	Main detaile		
88	Service name		
Ð	Service name (derauit: Hostname)		
	Hostname	Port 💿	
		27017	
No.	Username ()	Password ()	
10		Password	
	Max query length 💿		
≩	Max query length		
- 			
\$	Labels		
	Please choose some labels to help identify your services. La Editing existing labels may affect your inventory and its data	ibels can be useful, for example, to help define groupings. I.	
	Environment	Cluster	
	Replication set	Region 🕐	
	Availability Zone 🕥		
	Custom labels Follow the format as exemplified below one label per line		
	kev1:value1		
	Additional options		
33	Skip connection check		
	Use TLS for database connections		
C	Skip TLS certificate and hostname validation		
	Use QAN MongoDB Profiler		
0			

图 7-5-5 配置 MongoDB 服务

Ξ,	S8 MongoDB / MongoDB Instances	Overview ☆ ペ					
Q	Interval auto - Environment Al -	Oluster, All + Node Name	All - Service Name All - PMM	Annotations 💽 🕼	Home 25 Query Analytics		obe =
☆	= PMM						
88	~ Overview						
æ	Services	Min MongoD8 Uptime	Total Used Resident Memory	Total Used Virtual Memory	r Total Used Ma	pped Memory	Total Ci
		10.27 hour	49.00 MB	2.94 GB			0.80
	Top Connections		Top Opened Oursons	Min Of		Ma	Latency
alla							
	41		4	0.80 o	ps/s	10	9.33 µs
	Connections Detail (2 prevels)						
0.	> Cursors Detail (t.psnets)						
5	> Latency Detail (i pareta)						
~							
	Min Index Scanned Ratio		Max Index Scanned Ratio	Min Document S	canned Ratio	Max Docum	ent Scanned
	> Owers Efficiency Datail (6 access)						
	-						
	Top Opcounters		Top Document Operations	Top Queued O	perations	Total A	ssert Events
	1.64		0.62	0.00			A
	1.04 ops/s		0.03 ops/s	0.00 8	ps/s	0.2	U ops/s
۲	Opcounters Detail (12 parent)						
	> Document Operations Detail (Specer						
۰	> Queued Operations Detail (4 parent)						
	> Assert Events Detail (10 pinels)						

图 7-5-6 Overview

Overview 中指标说明。

- Command Operations: 命令操作,显示所选时间间隔内平均每秒执行命令 的次数,查找峰值和下降,并将它们与其他图形相关联
- Connections: 连接,显示 mongod 进程中的 TCP 连接(传入)
- Cursors: 游标,显示活动游标与由于应用程序未关闭连接而在 10min 后自 动终止的游标的比较,帮助确定连接增加的原因
- Document Operations: 文档操作,显示给定时间段内平均受影响的文档的 实际数量
- Queued Operations: 队列操作,显示了由于锁定而等待的读取和写入操 作数
- Asserts: 断言,显示给定时间段内平均每秒的断言事件数
- Memory Faults:内存故障,显示内存故障表示由于缺少索引或数据集内存 不足

2. Summary

小贴士

、小贴士)

在 PMM Server 左侧导航选择 MongoDB 项,单击 "Summary" 选项卡查看 MongoDB 实例的 概要信息,如图 7-5-7 所示。

Summary 中指标说明。

- Command Operations: 命令操作,按传统线路协议类型(查询、插入、更新、删除、获取更多)分类的操作数或复制操作数每秒。并且(从内部TTL 线程)文档按TTL 索引每秒删除
- Latency Detail: 延迟详细信息,操作的平均延迟[按读取、写入或(其他) 命令分类]
- Document Operations: 文档操作,每秒插入、更新、删除或返回的文档数
- Query Efficiency: 查询效率,返回的文档或扫描的索引条目/扫描的完整文档的比率的信息
- Scanned and Moved Objects: 扫描和移动的对象,显示对象数[数据 (scanned_objects)和索引(扫描)]以及由于文档大小增大而移动到新位置 的文档数
- Write Time: 写时间,每秒执行命令以确认写入问题所花费的时间数和总和
- Write Operations: 写操作,尝试确认写入问题时超时的命令数码
- Page Faults: 页面错误, UNIX 或 Windows 内存页错误。不一定来自 MongoDB

项目七





50

-

3. High availability

在 PMM Server 左侧导航选择 MongoDB 项,选择"High availability",单击"ReplSet Summary" 选项卡,查看 MongoDB ReplSet 的概要信息,如图 7-5-8 所示。



项目七

ReplSet Summary 中指标说明。

- Replication Lag: 复制滞后,当辅助节点无法以足够快的速度复制数据以 跟上数据写入主节点的速率时,就会发生 MongoDB 复制滞后。它可能是 由网络延迟、网络内的数据包丢失或路由问题等简单原因引起的
- Operations-byservicename:基于服务名称的操作,操作按传统线路协议类型(仅插入、更新和删除)进行分类
- Max Member Ping Time-byservicename: 基于服务名称的最大成员 Ping 时间,以 ping 为衡量指标显示复制滞后值的相关服务
- Max Heartbeat Time: 最大心跳时间,从现在到副本集成员的上次检测信号 之间的时间跨度
- Elections:选举,显示选举计数。通常为零,每次选举中将出现每个健康
 节点的1个计数。当主要角色由于正常维护或故障事件而更改时发生
- Oplog Recovery Window-by service name: 基于服务名称的 Oplog 恢复窗口, Oplog 集合中最新和最旧运算之间的时间跨度"窗口"

4. InMemory

项目七

52

在 PMM Server 左侧导航选择 MongoDB 项, 单击"InMemory"选项卡, 查看 MongoDB ReplSet 的概要信息, 如图 7-5-9 所示。

InMemory 中指标说明。

- InMemory Transactions: 内存事务,显示 WiredTiger 内部交换
- InMemory Capacity: 内存容量, 配置了 WiredTiger 缓存的最大值和当前值
- InMemory Sessions: 内存会话,内部 WiredTiger 存储引擎游标和会话当前 处于打开状态
- InMemory Pages: 内存页面, WiredTiger 缓存中的页面

InMemory Concurrency Tickets: 内存并发票证,在 WiredTiger 存储引擎中
 小贴士
 同时运行的每个操作都会分配一个 WiredTiger "票证","可用门票"=硬

- 同时运行的每个操作都会分配一个 WiredTiger"票证","可用门票"=硬 编码的高价值—"门票售罄"
- Document Changes: 文档更改,混合指标: 在任何类型的节点(主节点或 辅助节点)上每秒插入、更新、删除或返回的文档数,其包括复制写入操 作数和 TTL 每秒删除次数
- InMemory Cache Eviction:内存中缓存逐出,显示给定时间段内已从 WiredTiger缓存中逐出的页数。InMemory存储引擎仅逐出已修改的页面, 这表示压缩数据并删除脏页面



图 7-5-9 InMemory

5. MMAPv1

在 PMM Server 左侧导航选择 MongoDB 项, 单击"MMAPv1"选项卡, 查看 MongoDB MMAPv1 的概要信息, 如图 7-5-10 所示。



项目七



返目七



- MMAPv1 Journal Write Activity: 日志写入活动,显示 MongoDB 通过内存 中的日志进行处理
- MMAPv1 Journal Commit Activity: 日志提交活动,已提交到日志磁盘的 MongoDB 数据
- MMAPv1 Background Flushing Time: 后台刷新时间,在 mongod 进程的完

使用 MongoDB 实现数据库服务

项目七

整正常运行时间内, MMAP 后台刷新已花费的平均时间(单位: ms)

- Queued Operations: 队列操作,显示等待提交到存储引擎层的运算的队列 大小
- Client Operations: 客户端操作,操作数和复制操作数/秒,按传统线路协议类型(查询、插入、更新、删除、获取更多)分类

6. WiredTiger

在 PMM Server 左侧导航选择 MongoDB 项, 单击"InMemory"选项卡, 查看 MongoDB ReplSet 的概要信息, 如图 7-5-11 所示。

	88 MongoDB / MongoDB WiredTiger Details 🌣 🤞	S O Last 12 hours - D, D, 1m - 💭
a'	Internal auto - Charler All - Service Name MorgoOS Replication 10.10.2.72 - INME Associations	👔 lisene 🎊 Query Analytics 🖗 Company 😑 Mangadit 😑 IAA 😑 Services 😑 PMAA
\$	WiredTiger Cache Usage WiredTiger Max Cache Size	Memory Cached Memory Available
88	1.25 MB 256.00 MB	N/A 54.3%
G		Manthias Sada Labolit
1.6	199	25.00 M/v
		200161
		100040
alit		500 kB/w 6 65 KV s
۲		06:00 08:00 12:00 12:00 16:00 16:00
÷	- nded_tack 446 470 - convertined 0.72 0.24 0.23	within hum 0.00 km 21.47 kKm 0.00 km 21.47 kKm 0.00 km 21.47 kKm 0.00 km 0.00
Θ,	WeedToper Block Activity	17 WestTopr Sessons
9		2110
		1500
		1500
θ,		100
5	- wither	0.00 04.00 04.00 10.00 14.00 14.00 14.00
		- Sentore 17.40 19.00 17.47
	WredTiger Concurrency Tickets Available	Coursed Operations
		100
5		40
0		13
		8.00 NOO 1000 1000 1000 1000 1000 1000
		- witer 0.00 0.00
	- Mongali Bileginadet 10,112.72 128.00 128.00	- ma
	U WredTiger Checkpoint Time	V WeetTiger Cache Existion
	1.0/mm 10.001	120
		8
θ,	10.00 mm	\$19 }
5		- 123
	- nai 11.64 11.164	06.00 08.00 10.00 12.00 14.00 56.00
	- mit 1700mi 17.00mi 17.00mi	- Managati Magalasi da 10.2.72 0.00 0.00
	MiredTiger Cache Capacity	WiredTiger Cache Pages
		n
	192.73 Mil	
	0330.448	8
÷.		0600 0600 1000 1200 1400 1600
5	- Mai 25601 MB 25600 MB 25600 MB	- 102 00 11 10
100		- BY 2 1 1
	15.00 spxs	200 00 2/4
		150.003/9
		106.06.00
		0.00 8/2
5	- 500 - 107-00-001-001-001-001-001-001-001-001-0	
<u> </u>	- with 222 april 0.44 april 0.22 april	- stretten 47.54.8/s 153.63.8/s 93.103/s - prefead 22.51.8/s 116.38.8/s 47.64.8/s
	WindTiger Log Records	Document Changes
		181901
	CTI spev	8.65 (30/2) · · · · · · · · · · · · · · · · · · ·
	0.5 spex	\$20 years
		8.00 years 06:00 00:00 10:00 10:00 10:00
	Mile (Hall Beg - an uncompassed d.35 agents 0.12 agents 0.12 agents	mit mit ang - en Marred Ditingen Ditingen Ditingen
		- It.ostel
		1 Page Faults
	0.0 465	
	0.40 op.9	1.89//
		100
0		
0	na na ng- nannad, atjada 8.64 apara 8.62 apara	0600 6800 1100 1200 1408 1600 with man wag-
		faits 8.00 0.00 8.00
	> Mongood Summary (streng)	

图 7-5-11 WiredTiger



WiredTiger(以下简称"WT")中指标说明。

- WiredTiger Transactions: 交换,显示 WT 内部交易
- WiredTiger Cache Activity:缓存活动,即每秒在WT缓存与数据文件之间 传输的数据量。写操作始终意味着数据写入磁盘;读取操作通常从操作系 统文件缓冲区缓存中的RAM获取数据,但如果缓存中没有相应数据,则 从磁盘读取
- WiredTiger Block Activity: 区块活动, WT 块管理器每秒处理的数据量
- WiredTiger Sessions: 会话,当前打开的内部 WT 存储引擎游标和会话
- WiredTiger ConcurrencyTickets Available: 并发票可用性,在WT存储引擎中,每个同时运行的操作都会分配一个WT"票证"。"可用"票证的数量等于硬编码的最大值减去当前已分配的票证数。
- WiredTiger Checkpoint Time: 检查点时间,在WT 检查点阶段花费的时间
- WiredTiger Cache Eviction:缓存逐出,由于WT缓存已满,最近使用最少 的页面被逐出
- WiredTiger Cache Capacity:缓存容量,显示配置WT缓存的最大值和当前值
- WiredTiger Log Operations: 日志操作,显示 WT 内部预写日志操作
- WiredTiger Log Activity: 日志活动,显示 WT 内部预写日志中每秒移动的 数据量
- WiredTiger Log Records: 日志记录,显示 WT 内部日志中每秒追加的记录数



返目七



任务四 实现私有云盘服务

【任务介绍】

Nextcloud 是一款开源免费的私有云存储网盘项目,可以快速便捷地搭建一套属于自己或团队的云同步网盘,从而实现跨平台跨设备文件同步、共享、版本控制、团队协作等功能。其客户端覆盖了 Windows、Mac、Android、iOS、Linux 等各种常见平台,也提供了网页端以及 WebDAV 接口。本任务在 openEuler 上安装 Nextcloud Server 软件,实现私有云盘服务。

【任务目标】

- (1) 实现在线安装 Nextcloud。
- (2) 实现多终端访问 Nextcloud 私有云盘。

【操作步骤】

步骤 1: 创建虚拟机并完成 openEuler 的安装。

在 VirtualBox 中创建虚拟机,完成 openEuler 的安装。虚拟机与操作系统的配置信息见表 8-4-1,注意虚拟机网卡的工作模式为桥接。

虚拟机配置	操作系统配置
虚拟机名称: VM-Project-08-Task-04-10.10.2.84	主机名: Project-08-Task-04
内存: 2GB	IP地址: 10.10.2.84
CPU: 1 颗 1 核心	子网掩码: 255.255.255.0
虚拟硬盘: 20GB	网关: 10.10.2.1
网卡:1块,桥接	DNS: 8.8.8.8

步骤 2: 完成虚拟机的主机配置、网络配置及通信测试。

启动并登录虚拟机,依据表 8-4-1 完成主机名和网络的配置,能够访问互联网和本地主机。

(1)虚拟机的创建、操作系统的安装、主机名与网络的配置,具体方法参见项目一。

(2)建议通过虚拟机复制快速创建所需环境。通过复制创建的虚拟机需依据本任务虚拟机与操作系统规划配置信息设置主机名与网络,实现对互联网和本地主机的访问。

步骤 3: 安装 Nextcloud 及环境依赖。

Nextcloud 的安装支持 AIO VM、Docker image、Web installer 等,本任务选择 Web installer 方 式安装。

1. 安装 Apache

58

提醒

操作命令:

Package	Architecture	Version	Repository	Size	
Installing:					
httpd	x86_64	2.4.51-15.0e2203sp2	update	1.3 M	
Installing dependencies:					
apr	x86_64	1.7.0-6.0e2203sp2	update	110 k	
apr-util	x86_64	1.6.1-14.0e2203sp2	update	109 k	
httpd-filesystem	noarch	2.4.51-15.0e2203sp2	update	11 k	
httpd-tools	x86_64	2.4.51-15.0e2203sp2	update	71 k	
mailcap	noarch	2.1.53-3.0e2203sp2	OS	34 k	
mariadb-connector-c	x86_64	3.1.13-4.0e2203sp2	OS	179 k	
mod_http2	x86_64	1.15.25-2.0e2203sp2	OS	126 k	
openEuler-logos-httpd	noarch	1.0-8.0e2203sp2	OS	11 k	
Transaction Summary					
Install 9 Packages					
T-4-1 d1 -: 1	0 M				
Iotal download size: 1.	9 M				
Downloading Dackages:					
Downloading rackages: (1/0), mailtain 2.1.52.2 ar2002 m2 means man 177.10 ($1.24.10$ 0.00					
(2/9): mancap-2.1.33-3.	$ttpd_1 0_8 \text{ o}e^{2t}$	03sp2 noarch rpm	1/7 KD/S = 2	1 kB	00.00
(3/9): mariadh-connecto	$r_{-c-3} = 1.0 - 0.0022$	$203 \text{ sn}^2 \times 86 64 \text{ rnm}$	$604 \text{ kB/s} \mid 1$	79 kB	00.00
(4/9): mod http://www.	5-2 oe2203sp2 x	2053p2.x00_04.1pm	$25 \text{ kB/s} \mid 1$	26 kB	00.0
(5/9): apr-1.7.0-6.0e220	$3 \sin^2 x 86 64 \text{ rpr}$	n	22 kB/s \perp 1	10 kB	00:0
(6/9): httpd-filesystem-2	.4.51-15.0e2203	sp2.noarch.rpm	34 kB/s 1	1 kB	00:0
			C. 110/0 1		50.0

		实现文件服务	项目八
32.	(8/9): httpd-2.4.51-15.0e2203sp2.x86_64.rpm 2.2 MI	B/s 1.3 MB 00:0	0
33.	(9/9): httpd-tools-2.4.51-15.0e2203sp2.x86_64.rpm 266 kE	B/s 71 kB 00:0	0
34.			~
35.	lotal 345 kE	B/s 1.9 MB 00:0	5
36.	Running transaction check		
37. 20	Transaction check succeeded.		
38. 20	Running transaction test		
39. 40	Prensaction test succeeded.		
40.	Running transaction $P_{\text{transaction}} = 21.12.4 \times 2202 \times 2.09$ (4)	1 /1	
41.	Running scriptiet: mariadb-connector-c-3.1.13-4.0e2203sp2.x86_64	1/1	
42.	Preparing $P_{\rm res}$ is the parameter of the preparing $P_{\rm res}$ is the prepared $P_{$	1/1	
43.	Running scriptlet: apr-1.7.0-6.0e2203sp2.x86_64	1/9	
44.	Installing : apr-1.7.0-6.0e2203sp2.x86_64	1/9	
45.	Running scriptlet: apr-1./.0-6.0e2203sp2.x86_64	1/9	
46.	Kunning scriptlet: httpd-filesystem-2.4.51-15.0e2203sp2.noarch	2/9	
47.	Installing : httpd-filesystem-2.4.51-15.0e2203sp2.noarch	2/9	
48.	Installing : openEuler-logos-httpd-1.0-8.oe2203sp2.noarch	3/9	
49.	Installing : mariadb-connector-c-3.1.13-4.0e2203sp2.x86_64	4/9	
50.	Running scriptlet: apr-util-1.6.1-14.0e2203sp2.x86_64	5/9	
51.	Installing : apr-util-1.6.1-14.oe2203sp2.x86_64	5/9	
52.	Running scriptlet: apr-util-1.6.1-14.0e2203sp2.x86_64	5/9	
53.	Installing : httpd-tools-2.4.51-15.0e2203sp2.x86_64	6/9	
54.	Running scriptlet: mailcap-2.1.53-3.0e2203sp2.noarch	7/9	
55.	Installing : mailcap-2.1.53-3.0e2203sp2.noarch	7/9	
56.	Running scriptlet: mailcap-2.1.53-3.0e2203sp2.noarch	7/9	
57.	Installing : mod_http2-1.15.25-2.0e2203sp2.x86_64	8/9	
58.	Installing : httpd-2.4.51-15.0e2203sp2.x86_64	9/9	
59.	Running scriptlet: httpd-2.4.51-15.oe2203sp2.x86_64	9/9	
60.	Verifying : mailcap-2.1.53-3.0e2203sp2.noarch	1/9	
61.	Verifying : mariadb-connector-c-3.1.13-4.0e2203sp2.x86_64	2/9	
62.	Verifying : mod_http2-1.15.25-2.0e2203sp2.x86_64	3/9	
63.	Verifying : openEuler-logos-httpd-1.0-8.oe2203sp2.noarch	4/9	
64.	Verifying : apr-1.7.0-6.0e2203sp2.x86_64	5/9	
65.	Verifying : apr-util-1.6.1-14.0e2203sp2.x86_64	6/9	
66.	Verifying : httpd-2.4.51-15.0e2203sp2.x86_64	7/9	
67.	Verifying : httpd-filesystem-2.4.51-15.oe2203sp2.noarch	8/9	
68.	Verifying : httpd-tools-2.4.51-15.oe2203sp2.x86_64	9/9	
69.			
70.	Installed:		
71.	apr-1.7.0-6.0e2203sp2.x86_64 apr-util-1.6.1-1	4.oe2203sp2.x86_64	httpd-2.4.51-15.
	oe2203sp2.x86_64 httpd-filesystem-2.4.51-15.oe220	3sp2.noarch	
72.	httpd-tools-2.4.51-15.0e2203sp2.x86_64 mailcap-2.1.53-3	.oe2203sp2.noarch	mariadb-connec
	tor-c-3.1.13-4.oe2203sp2.x86_64 mod_http2-1.15.25-2.oe2203sp2.x86_64	86_64	
73.	openEuler-logos-httpd-1.0-8.0e2203sp2.noarch		
74.			
75.	Complete!		
		操作命令+配置	文件+脚本程序+结束

2. 安装 PHP

60

操作命令:

1.	# 通过 yum 工具安装 php
2.	[root@Project-08-Task-04 ~]# yum -y install php
3.	Last metadata expiration check: 0:00:46 ago on Tue 17 Oct 2023 05:05:26 PM CST.
4.	Dependencies resolved.

•	Package	Architecture	Version	Repository	Size	
•	Installing:					_
	Php	x86_64	8.0.30-1.oe2203sp2	update	1.6 M	
0.	Installing dependen	icies:				
1.	libargon2	x86_64	20190702-3.oe2203sp2	OS	28 k	
2.	nginx-filesystem	noarch	1:1.21.5-5.0e2203sp2	everything	8.9 k	
3.	php-cli	x86 64	8.0.30-1.oe2203sp2	update	3.3 M	
4.	php-common	x86_64	8.0.30-1.oe2203sp2	update	546 k	
5.	Installing weak dep	pendencies:		-		
5. 7	php-fpm	x86_64	8.0.30-1.0e2203sp2	update	1.7 M	
7. 8.	Transaction Summa	ary				
).).	Install 6 Packages	======================================				=
1.						
2.	Total download siz	xe: 7.3 M				
). 1	Downloading Dealer					
+. <	(1/6); nainy filosyst	ages. $1.2155.$	2202 maarah mm	$901 \text{D/}_{\odot} + 90$	1-D	00.10
5. 5	(1/0). ngmx-mesys	M省較了朝公士	2203sp2.noarch.ipin 具元信自	091 D/S 0.9	КD	00.10
). 7	# /y J HF/IX /J 使此	又1mg」即刀1	化小同芯	$10 \text{ MD/}_{a} \pm 22$	MD	00.01
/. >	(0/0): pnp-cn-8.0.50	0-1.0e2203sp2.x	.80_04.rpm	1.9 MD/8 5.5	MD	00:01
).).	Total			624 kB/s 7.3	MB	00:11
).	Running transaction	n check		'		
Ι.	Transaction check	succeeded.				
2.	Running transaction	n test				
3.	Transaction test su	cceeded.				
4.	Running transaction	n				
5.	Preparing				1/1	
5.	Installing	: php-common	-8.0.30-1.0e2203sp2.x86	64	1/6	
7.	# 为了排版方便	此处省略了部分	分提示信息		2.0	
3.	Verifying	: php-fpm-8.0	0.30-1.0e2203sp2.x86_64		6/6	
	Installed					
). 1	libargon2 201007	102 3 0a2202am	v86 61 nainy	filesystem 1.1.21	5 5 00	2203 cm2 noarch
•	nbargon2-201907	203 sp 286.64	nhp ali 8 0 20 1	0e2203cm2 v86	.5-5.0e. 54	2203sp2.110arch
,	php-0.0.30-1.0e2	2053p2.x00_04	$p_{11} - 0.0.30 - 1$	$\frac{1002203 \text{ sp}2.800}{\text{ sp}2.800}$	03 cm 2	86 61
2.	php-common-8.0.	.50-1.0e2203sp2		pm-6.0.30-1.0e22	03sp2.2	100_04
•	Completel					
·	Complete:					
				<u></u>	制作命令	+配置又件+脚本程序-

项目八

实现文件服务

项目八

项目八

3. 为 Nextcloud 配置站点

Carrier

操作命令:

1.	# 创建目录并授权
2.	[root@Project-08-Task-04 ~]# mkdir /var/www/html/nextcloud/
3.	[root@Project-08-Task-04 ~]# chmod 777 /var/www/html/nextcloud/
4.	# 创建并编辑站点配置文件
5.	[root@Project-08-Task-04 ~]# vi /etc/httpd/conf.d/nextcloud.conf
6.	# 配置内容如下
7.	<virtualhost *:80=""></virtualhost>
8.	DocumentRoot /var/www/html/nextcloud/
9.	ServerName your.server.com
10.	
11.	<directory html="" nextcloud="" var="" www=""></directory>
12.	Require all granted
13.	AllowOverride All
14.	Options FollowSymLinks MultiViews
15.	
16.	<ifmodule mod_dav.c=""></ifmodule>
17.	Dav off
18.	
19.	
20.	
21.	
	操作命令+配置文件+脚本程序+结束
	1 准久穴柱亡化
	4. 任田又水入门

操作命令:

下载 Web 安装引导文件至上面站点目录中 froot@Project-08-Task-04 ~]# wget -P /var/www/html/nextcloud/ https://download.nextcloud.com/server/ins
root@Project-08-Task-04 ~]# wget -P /var/www/html/nextcloud/ https://download.nextcloud.com/server/ins
aller/setup-nextcloud.php
-2023-10-17 17:13:06 https://download.nextcloud.com/server/installer/setup-nextcloud.php
Resolving download.nextcloud.com (download.nextcloud.com) 5.9.202.145, 2a01:4f8:210:21c8::145
Connecting to download.nextcloud.com (download.nextcloud.com) 5.9.202.145 :443 connected.
HTTP request sent, awaiting response 200 OK
Length: 158285 (155k)
Saving to: '/var/www/html/nextcloud/setup-nextcloud.php'
setup-nextcloud.php 100%[===================================
===>] 154.58k 233kB/s in 0.7s
2023-10-17 17:13:08 (233 kB/s) - '/var/www/html/nextcloud/setup-nextcloud.php' saved [158285/158285]
操作命令+配置文件+脚本程序+结束

5. 启动 httpd

操作命令:

1. # 启动 httpd

2. [root@Project-08-Task-04 ~]# systemctl start httpd

- 3. # 设置开机自动
- 4. [root@Project-08-Task-04 ~]# systemctl enable httpd
- 5. Created symlink /etc/system/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.s ervice

操作命令+配置文件+脚本程序+结束

6. 配置安全措施

openEuler 默认开启防火墙,为使 Nextcloud 能正常对外提供服务,本任务暂时关闭防火墙等 安全措施。

操作命令:

62

项目ハ

- 1. # 使用 systemctl stop 命令关闭防火墙
- 2. [root@Project-08-Task-04 ~]# systemctl stop firewalld
- 3. # 使用 setenforce 命令将 SELinux 设置为 permissive 模式
- 4. [root@Project-08-Task-04 ~]# setenforce 0

操作命令+配置文件+脚本程序+结束

7. 通过 Web 方式引导安装

(1) 浏览器访问配置的 Nextcloud 站点,通过引导开始安装,访问地址为 https://10.10.2.84/ setup-nextcloud.php,并单击 "Next" 按钮如图 8-4-1 所示。



实现文件服务

项目八



图 8-4-2 环境依赖检测

操作命令:

1.	# 通过 yum 工具	具安装所需的 pł	np 模块依赖					
2.	[root@Project-08-Task-04 ~]# yum -y install php-zip php-dom php-xml php-mbstring php-pdo php-gd							
3.	Last metadata expiration check: 2:16:42 ago on Wed 27 Sep 2023 10:00:08 AM CST.							
4.	Dependencies re	esolved.						
5.							反日	
6.	Package	Architecture	Version	Repository	Size		\geq	
7.								
8.	Installing:							
9.	php-gd	x86_64	8.0.30-1.0e2203sp2	update	34 k			
10.	# 为了排版方便	更此处省略了部分	分提示信息					
11.	php-common	x86_64	8.0.30-1.0e2203sp2	update	546 k			
12.								
13.	Transaction Sun	nmary						
14.								
15.	Install 13 Pack	kages						
16.								
17.	Total download	size: 2.0 M						
18.	Installed size: 1	2 M						
19.	Downloading Pa	ackages:						
20.	(1/13): cmake-fi	lesystem-3.22.0-	-5.0e2203sp2.x86_64.r	pm 2.3 kB/s	s 12 kB	00:05		
21.	# 为了排版方便	更此处省略了部分	分提示信息					
22.	(13/13): php-mb	string-8.0.30-1.c	pe2203sp2.x86_64.rpm	n 950 kB/	/s 426 kB	00:00		
23.								
24.	Total			331 kB/	's 2.0 MB	00:06		
25.	Running transac	tion check						

(2)环境依赖检测。Nextcloud 检测到部分依赖未安装,如图 8-4-2 所示,根据提示进行安装。

Concession of the second

项目八

64

26.	Transaction check succeeded.		
27.	Running transaction test		
28.	Transaction test succeeded.		
29.	Running transaction		
30.	Preparing		1/1
31.	Installing : php-common-8.0.30-	1.oe2203sp2.x86 64	1/13
32.	# 为了排版方便此处省略了部分提示信	言息	
33.	Verifying : php-xml-8.0.30-1.00	e2203sp2.x86 64	13/13
34.			
35.	Installed:		
36.	cmake-filesystem-3.22.0-5.0e2203sp2.x8	6_64 gd-2.3.3-2.0e220	03sp2.x86_64
	libXpm-3.5.13-5.0e2203sp2.x86_64	libwebp-1.2.1-4.0e2203sp	2.x86_64
37.	libxslt-1.1.37-1.0e2203sp2.x86 64	libzip-1.8.0-2.0e	2203sp2.x86 64
	oniguruma-6.9.6-1.0e2203sp2.x86 64	php-common-8.0.30-1.oe2	203sp2.x86 64
38.	php-gd-8.0.30-1.0e2203sp2.x86_64	php-mbstring-	8.0.30-1.0e2203sp2.x86 64
	php-pdo-8.0.30-1.0e2203sp2.x86 64	php-pecl-zip-1.19.3-1.0e22	203sp2.x86_64
39.	php-xml-8.0.30-1.0e2203sp2.x86 64		· _
40.			
41.	Complete!		
			操作命令+配置文件+脚本程序+结束

(3) 指定安装目录, 输入"."即安装至当前目录, 单击"Next"按钮, 如图 8-4-3 所示。



图 8-4-3 指定安装目录

(4) 进入"Success"页面,说明安装成功,单击"Next"按钮,如图 8-4-4 所示。

(5) 创建管理员账户,设置用户名为 admin,密码为 openEuler#123,并单击"安装"按钮为 其安装 SQLite 数据库,如图 8-4-5 所示。

(6)安装完成后,进入"推荐的应用"界面,单击"安装推荐的应用"按钮,如图 8-4-6 所示。



图 8-4-4 安装成功



图 8-4-5 创建管理员账户

	0 0 0
	推荐的应用
ť	Calendar SHEITHEDEW, HESEDERNGMER,
4	Contacts 持恐的周奏和朋友放在一个地方,而不会道着他们的私人很多。
	Mail 前来的电子邮件会项程序地文件,联系人和目标地来更成。
	Nextcload Office 兼于 Collabora Online 的交話。要想与第元交話。
	Notes 无干结的间凹已来应用。
	С Так адеживновалериян, кажа, якия, кіланскоска.
	Rot Standing
	Nextcloud - 指包所有数第一个安全的家



实现文件服务 项目八

项目八



(7)安装推荐的应用完成后,进入 Nextcloud,关闭欢迎窗口,如图 8-4-7 所示。

图 8-4-7 首页

(8)单击左上角横向导航中的"文件"按钮,可看到默认的目录和指导性文件等,如图 8-4-8 所示。

▶ 全部文件	EX 🔹) + 2682			
0 #if:				
★ 收靈	Welcome to Nextcloud! 🗧 d	S 🔳		
< 共享	Here you can add a description or any other info relevant for the fold "Readme.md" and in the web interface also embedded nicely up at th	er. It will show as a ie top.		
● 标签				
共享圈子的文件	Readme.md Talk		emplates credits。 副近相改	nd
	- 名称 -		大小	修改日期
	- 種 模板	<	10.2 MB	an hour ag
	Documents	<	••• 1.1 MB	23 minutes
	Photos	<	5.4 MB	an hour ag
	Talk	<	0 КВ	an hour ag
	Nextcloud.png	<	••• 49 KB	an hour ag
已账除文件	Nextcloud intro.mp4	<	3.8 MB	an hour ag
✿ 37.5 MB 已使用	Nextcloud Manual.pdf	<	14.4 MB	an hour ag
Å ⊽##9 #	Readme.md	<	••• < 1 KB	an hour ag

图 8-4-8 文件

Nextcloud 的 Web 端的文件支持在线查阅,支持拖动上传等。

步骤 4:服务测试。

66

项目八

1. 在 Windows 操作系统中访问 Nextcloud

(1)本任务选择 Windows 桌面客户端进行测试,下载安装后启动,进入"添加 Nextcloud 个人账户",单击"登录"按钮如图 8-4-9 所示。



图 8-4-9 添加 Nextcloud

(2)设置服务器地址,单击"下一步(N)"按钮,如图 8-4-10 所示,此时浏览器自动打开窗口,授权客户端访问,单击"授权访问"按钮,如图 8-4-11 所示。



项目八

图 8-4-10 Windows 端设置服务器地址





(3) 客户端授权成功,并设置本地同步文件夹,完成后可在本地文件夹中看到已同步的云盘 内容,说明访问成功,如图 8-4-12 所示。

A 104 mm 100 mm 1000 mm 1000 mm 100 mm 100 mm 100 mm 100 mm 100 mm 100 m		1 16 #	₩ 項目 • □	P	tTŦ -
オ オ 国 定到快	路快速方式 移动到 复制到 翻除 重命名	● 3.2 新建 文件夹	2松访问 ▼		2 编辑 2 编辑 2 历史记录
剪贴板	组织	新建	2		打开
$\leftrightarrow \rightarrow \cdot \uparrow \odot$ > Nextcl	oud		~ (5	♪ 在!
	名称 ^	状态	类型		
> 🖈 快速访问	Set Set Set Set Set Set Set Set Set				
> 🔷 WPS云盘	Photos	0	文件來		
Neutrilaud	Talk	0	文件夹		
/ Wexicioud	- 模板	0	文件夹		
> 🌰 OneDrive - Personal	.nextcloudsync.log		文本文档		
一世田誌	sync_fe42eda2a098.db		Data Bas	e File	
	.sync_fe42eda2a098.db-wal		DB-WAL	文件	
> 🥩 网络	 Nextcloud intro.mp4 	0	MP4 文件		
	🛃 Nextcloud Manual.pdf	0	WPS PDF	文档	
	log Nextcloud.png	0	PNG 图片	文件	
	Readme.md	0	Markdov	vn 源:	文(
	🛃 Reasons to use Nextcloud.pdf	0	WPS PDF	文档	
	Templates credits.md	0	Markdow	vn 源:	文化

图 8-4-12 文件同步

2. 在 Android 操作系统中访问 Nextcloud

(1) Android 客户端的下载地址为 https://download.nextcloud.com/android/dev/latest.apk, 安装 完成后,输入服务器地址,单击"->"按钮,如图 8-4-13 所示。

(2) 登录到 Nextcloud, 输入账户名和密码, 然后在 Web 端进行授权, 单击"登录"按钮, 如图 8-4-14 所示。

(3) 登录后,可看到云盘中的内容,说明访问成功,如图 8-4-15 所示。



实现文件服务

3. 测试结果

测试结果见表 8-4-2, 通过测试结果可知满足需求。

序号	操作系统	结果
1	Windows	正常
2	Android	正常







任务七 使用 Linux-Dash 实现可视化实时监控



【任务介绍】

Linux-Dash 是基于 Web 的系统状态监控工具。本任务通过对该软件的部署配置,实现对主机系统进程、CPU、内存、网络、磁盘等的可视化实时监控。

【任务目标】

- (1) 实现 Linux-Dash 软件的部署。
- (2) 实现对主机系统运行情况的监控。

【操作步骤】

步骤 1: 创建虚拟机并完成 openEuler 的安装。

在 VirtualBox 中创建虚拟机,完成 openEuler 的安装。虚拟机与操作系统的配置信息见表 11-7-1,注意虚拟机网卡的工作模式为桥接。

表 11-7-1 虚	拟机与操作系统配置
------------	-----------

虚拟机配置	操作系统配置	
虚拟机名称: VM-Project-11-Task-07-10.10.2.117	主机名: Project-11-Task-07	
内存: 1GB	IP 地址: 10.10.2.117	
CPU: 1 颗 1 核心	子网掩码: 255.255.255.0	
虚拟硬盘: 20GB	网关: 10.10.2.1	
网卡:1块,桥接	DNS: 8.8.8.8	

项目十-

步骤 2: 完成虚拟机的主机配置、网络配置及通信测试。

启动并登录虚拟机,依据表 11-7-1 完成主机名和网络的配置,能够访问互联网和本地主机。

(1)虚拟机的创建、操作系统的安装、主机名与网络的配置,具体方法参见项目一。



(2)建议通过虚拟机复制快速创建所需环境。通过复制创建的虚拟机需依据本任务虚拟机与操作系统规划配置信息设置主机名与网络,实现对互联网和本地主机的访问。

步骤 3: 安装部署的基本环境。

本任务基于 PHP 环境进行软件的安装,所需要的基本安装条件见表 11-7-2,具体步骤如下。

网站服务器	Apache 2 或 Nginx
PHP 解析器	5 或更高版本
防火墙开放端口	TCP 80(HTTP 协议)

表 11-7-2 Linux-Dash 软件 PHP 环境部署基本条件表

(1) 使用 yum 工具安装 Apache, 启动 httpd 服务并配置为开机自启动,详细步骤可参见项目 四的任务一。

(2) 使用 yum 工具安装 PHP 解析器,详细步骤可参见项目四的任务三。

(3)为使Linux-dash能够获取本地主机的运行信息,并能够访问监控界面,需将主机的SELinux 设置为 permissive 模式,防火墙开放 httpd 服务的端口。

(4) 在安装操作的过程中,根据不同的部署条件验证是否满足 Linux-Dash 部署要求。 通过查看/proc/net/dev 目录信息,监控当前系统网卡接口通信流量的情况。

操作命令:

1	# 使田 yum 工具安装 Anache
1.	
2.	[root@Project-11-1ask-0/~]# yum install -y httpd
3.	
4.	# 设置 Apache 服务启动
5.	[root@Project-11-Task-07 ~]# systemctl start httpd
6.	
7.	# 设置 Apache 服务开机自启动
8.	[root@Project-11-Task-07 ~]# systemctl enable httpd
9.	$Created \ symlink \ /etc/system/system/multi-user.target.wants/httpd.service \ \rightarrow \ /usr/lib/system/system/httpd.$
	service.
10.	
11.	# 验证服务是否为开机自启动
12.	[root@Project-11-Task-07~]# systemctl is-enabled httpd
13.	enabled
14.	
	71

Linu	X版务器构建与运维官理从基	面到头戗(基丁 openEuler)			
15	# 使田 wum 工具安装 PHP 解料	F哭(木次安装 PHP 版木为 8 0)			
16	[root@Project-11-Task-07~]# vum install -v php php-devel				
17	[100t@110jeet-11-1@sk-07~3]// yuun instan -y prip prip-dever				
18.	# 设置 SELinux 模式(关闭状态)				
19	[root@Project-11-Task-07~]# se	tenforce 0			
20.					
21.	# 配置防火墙规则(允许 http	办议访问)			
22.	[root@Project-11-Task-07~]# fin	ewall-cmdadd-service=httppermanent			
23.	success	1 1			
24.	# 使防火墙规则配置生效				
25.	[root@Project-11-Task-07~]# fin	ewall-cmdreload			
26.	success				
27.					
28.	# 使用 php - v 命令验证已安装	裝的 PHP 版本			
29.	[root@Project-11-Task-07~]# ph	p-v			
30.	PHP 8.0.28 (cli) (built: Feb 14 20				
31.	Copyright (c) The PHP Group				
32.	Zend Engine v4.0.28, Copyright	(c) Zend Technologies			
33.					
34.	# 使用 httpd - v 命令验证 Apa	uche 版本			
35.	[root@Project-11-Task-07 ~]# ht	tpd -v			
36.	Server version: Apache/2.4.51 (Unix)				
37.	Server built: Mar 13 2023 16:30:49				
38. 20		<u>+</u> -+			
39. 40	# 使用 sestatus 验证 SELinux 档				
40.	[root@Project-11-1ask-0/~]# se	status			
41.	SELinux status:				
42.	SELinux root directory:	/sys/is/scillux			
	Loaded policy name:	targeted			
45.	# 查看当前 SELinux 的状态为	nermissive (允许)			
46.	Current mode:	permissive			
47.	Mode from config file:	enforcing			
48.	Policy MLS status:	enabled			
49.	Policy deny unknown status:	allowed			
50.	Memory protection checking:	actual (secure)			
51.	Max kernel policy version:	33			
52.					
53.	# 使用 firewalld-cmd 命令查看	防火墙允许通过服务规则			
54.	[root@Project-11-Task-07~]# fin	ewall-cmdlist-services			
55.	dhcpv6-client http mdns ssh				
			操作命令+配置文件+脚本程序+结束		



72

项目十一

Linux-Dash 目前支持在 PHP、Node.js、python 或 Go 环境下进行安装,其他环 境的安装简要步骤可参照本任务的【任务扩展】内容。
步骤 4: 安装 Linux-Dash 程序。

本任务使用 wget 工具从 github 仓库中下载 Linux-Dash 程序并进行安装。

操作命令:

1.	# 使用 wget 工具下载 Linux-Dash					
2.	[root@Project-11-Task-07 ~]# wget https://github.com/tariqbuilds/linux-dash/archive/master.zip					
3.	2023-07-11 14:59:02 https://github.com/tariqbuilds/linux-dash/archive/master.zip					
4.	正在解析主机 github.com (github.com) 20.205.243.166					
5.	正在连接 github.com (github.com) 20.205.243.166 :443 已连接。					
6.	已发出 HTTP 请求,正在等待回应 302 Found					
7.	位置: https://codeload.github.com/tariqbuilds/linux-dash/zip/refs/heads/master [跟随至新的 URL]					
8.	2023-07-11 14:59:02 https://codeload.github.com/tariqbuilds/linux-dash/zip/refs/heads/master					
9.	正在解析主机 codeload.github.com (codeload.github.com) 20.205.243.165					
10.	正在连接 codeload.github.com (codeload.github.com) 20.205.243.165 :443 已连接					
11.	已发出 HTTP 请求,正在等待回应 200 OK					
12.	长度: 未指定 [application/zip]'					
13.						
14.	master.zip [<=>] 121.52K 163kB/s in 0.7s					
15.	# 下述信息表示文件下载成功					
16.	2023-07-11 14:59:04 (402 kB/s) - 'master.zip' saved [124852]					
17.						
18.	# 使用 unzip 工具将~/master.zip 文件解压到/var/www 目录下					
19.	[root@Project-11-Task-07 ~]# unzip master.zip -d /var/www/					
20.						
21.	# 设置 Linux-Dash 程序目录的属主和属组均为 apache, 权限为 755					
22.	[root@Project-11-Task-07 ~]# chown -R apache.apache /var/www/linux-dash-master					
23.	[root@Project-11-Task-07 ~]# chmod -R 755 /var/www/linux-dash-master/					
	操作命令+配置文件+脚本程序+结束					

步骤 5: 发布 Linux-Dash 程序。

本任务使用 80 端口发布 Linux-Dash 软件,需要完成的操作如下。

```
(1) 配置 Apache 的 httpd.conf 文件,修改默认网站的目录配置信息,发布 Linux-Dash 程序。
```

配置文件: /etc/httpd/conf/httpd.conf

操作命令:

- 1. # 检查修改后的配置文件信息
- 2. [root@Project-11-Task-07 ~]# cat /etc/httpd/conf/httpd.conf | grep -vE "#|^\$"
- 3. # 默认网站配置
- 4. Listen 80
- 5. # 默认网站目录/var/www/html,修改并查看/var/www/linux-dash-master/app
- 6. DocumentRoot "/var/www/linux-dash-master/app"
- 7. <Directory "/var/www/linux-dash-master/app ">

73

项目十-

项目十一

- 8. Options Indexes FollowSymLinks
- 9. AllowOverride None
- 10. Require all granted
- 11. </Directory>

操作命令+配置文件+脚本程序+结束

(2) 配置 Apache 的 welcome.conf 文件,将/etc/httpd/conf.d/welcome.conf 文件的所有内容进行 注释,关闭 Apache 默认网站。

(3) 重新启动 httpd 服务, 使 Apache 服务配置生效。

操作命令:

74

项目十一

1. # 重新启动 httpd 服务, 配置生效

2. [root@Project-11-Task-07 ~]# systemctl restart httpd

操作命令+配置文件+脚本程序+结束

(4)访问 Linux-Dash 程序。在本地主机打开浏览器,输入 Linux-Dash 主机地址即可看到系 统监控界面,如图 11-7-1 所示。



步骤 6: 阅读 Linux-Dash 监控信息。

Linux-Dash 监控系统包含 5 个监控模块,分别为 SYSTEM STATUS(运行状态监控)、BASIC INFO(基本信息监控)、NETWORK(网络监控)、ACCOUNTS(用户访问监控)和 APPS(应用程序监控)。

(1) SYSTEM STATUS。可对主机的内存使用情况、CPU 平均负载、CPU 利用率、磁盘分区、 内存进程及 CPU 进程等信息进行实时监控,如图 11-7-2 所示。Linux-Dash 监控主机运行状态的字 段见表 11-7-3。

Linux Dash system status basic info network accounts apps - ≡ CPU Avg Load ≡ RAM Usage ≡ CPU Utilization 100 1_min_avg 2 % 226 MB (49%) Usage 7 % Used . 5_min_avg 1 % Available 229 MB of 455 MB 15_min_avg 0 % 0 ↔ • **5** ↔ - \equiv CPU Processes \equiv RAM Processes **3** ↔ -NAME STATS USED MOUNT Search Search /dev/map PID USER CPU% RSS VSZ CMD PID USER MEM% RSS VSZ CMD 1.96 per/opene 13% apache 25 4400 7444 ps 176 41360 305980 766 root 8.8 uler-root 19342 apache 25 4344 7444 ps /svs/fs/co 1362 root 5.6 26524 0/4.0M 0% kworker roup 797 root 18172 354480 tuned /0:0-19017 0/4.0M 0% 0 devtmpf /dev mm_per 19060 root 15832 35148 php-fpm tmpfs 0 / 228M 0% /dev/shn cpu_wq Network httpd 4.0M / 19070 0 9836 17452 root 779 root 2.5 11916 270988 Manage tmpfs 5.96 /run 92M kworker - = CPU Temp --≡ Swap Usage

图 11-7-2 系统状态监控界面

表 11-7-3 系统状态监控内容

监控类型	监控说明
RAM Usage(内存使用监控)	已使用的内存大小,以及所占总内存比例。该值包含了缓存和应用系 统实际使用的内存大小,目前主机中还剩余可以被应用程序使用的物 理内存大小
CPU Avg Load (CPU 负载)	最近 1min、5min、15min 平均 CPU 负载
CPU Utilization (CPU 利用率)	CPU 资源占用情况
Disk Partitions (磁盘分区)	磁盘中文件系统分区名称、使用状态、使用率、挂载目录等
CPU Processes (CPU 进程)	进程执行编号、属主、CPU 占用率、内存使用、执行的命令等信息
RAM Processes (内存进程)	进程执行编号、属主、内存占用率、内存使用、执行的命令等信息

项目十-

75

项目十一

76

续表

监控类型	监控说明
Swap Usage(Swap 使用监控)	交换空间的使用情况
Docker Processes(容器进程监控)	容器进程的运行状态
CPU Temp(CPU 温度)	CPU运行的温度情况

(2) BASIC INFO。可对主机的主机名、操作系统、开机时间、内存、CPU、任务计划、历史 任务计划执行以及 IO 状态等信息进行实时监控,如图 11-7-3 所示。Linux-Dash 监控主机基本信息 的字段见表 11-7-4。

监控类型	监控说明
General Info(基本信息)	主要包含主机名、操作系统版本信息、系统时间、运行时间等
Memory Info(内存信息)	主要包括内存使用情况、内存的一些设置信息等
CPU Info (CPU 信息)	主要包括 CPU 的厂商信息、数量、核心数、缓存配置信息等
Scheduled Cron Jobs(任务计划)	主要包括系统中设置的任务计划信息等
Cron Job History(历史任务计划)	主要包括系统中已经运行的任务计划信息等
IO Stats (IO 状态)	主要包括磁盘 IP 的一些读取的数量、状态信息等

表 11-7-4	Linux-Dash 监控主机基本信息的字段	殳



运维管理 项目十一

项目十-

77

(3) NETWORK。可对主机网卡的上下行网络速率、IP 地址、网络连接、网络 ARP 缓存表、 对外访问主机速度等信息进行实时监控,如图 11-7-4 所示。Linux-Dash 监控主机网络情况的字段 见表 11-7-5。

Upload Transf	fer Rate	•	≡ Download Tran	isfer Rate			≡ IP Addresses		0 ↔ 🖃
		100			1	00		Search	
							INTERFACE	IP	
							external	120.216.233.25	50
		0				0			
	enp0s3	0 KB/s		enp0s3	0 KB/s				
	10	0 KB/s		10	0 KB/s				
									ţ,
Network Conn	ections No data	び ↔ -	≡ ARP Cache Tab	le No data	0 ↔ -		≡ Ping Speeds	Search	σ ↔ -
Network Conn	lections No data	♂ ↔ •	≡ ARP Cache Tab	le No data	0 ↔ -		≡ Ping Speeds HOST	Search PING	∦ ن ↔ -
Network Conn	ections No data	₫ ↔ -	∃ ARP Cache Tab	le No data	3 ↔ -		≡ Ping Speeds HOST yahoo.com	Search PING 265.764	0 ↔ -

图 11-7-4 网络信息监控界面

表 11-7-5	Linux-Dash 监控主机网络情况的字段

监控类型	监控说明
Upload Transfer Rate	上行传输率。网卡上行(发送)数据速率
Download Transfer Rate	下行传输率。网卡下行(发送)数据速率
IP Addresses	主机 IP 地址。主机系统的 IP 地址信息
Network Connections	网络连接。主要包括网络连接的数量、来源地址与端口等信息
ARP Cache Table	ARP 缓存表。主机系统中 ARP 缓存表信息
Ping Speeds	测试 Ping 的访问速度。主要包括主机访问外网地址时,通过 Ping 命令测试网络速度等

(4) ACCOUNTS。可对登录用户情况进行监控统计,掌握主机连接客户端信息,如图 11-7-5 所示。Linux-Dash 监控主机用户登录访问情况的字段见表 11-7-6。

mux Da	ISII SISIEMS	STATUS BASIC INFO	NEIWORK	ACCOUNTS	1995		Resources: GI	THUD I GITTET CHAT ROOM
≡ Account	S	⊙ ↔ •	≡ Logged	In Accounts	Ø ↔ 🕤	≡ Recent	Logins	⊍ ↔ •
	Search			Search	i i i i i i i i i i i i i i i i i i i		Search	
TYPE	USER	HOME	USER	FROM	WHEN	USER	IP	DATE
system	root	/root	root	14:57	8:26	reat	10 10 2 101	Jul 11 14:57:55
system	bin	/bin	root	14:20	1:39m	1001	10.10.2.181	+0800 2023
system	daemon	/sbin						
system	adm	/var/adm						
system	lp	/var/spool/lpd						
system	sync	/sbin						
system	shutdown	/sbin						
vetam	halt	/ehin						

图 11-7-5 系统用户访问监控

表 11-7-6 用户访问监控内容

监控类型	监控说明
Accounts(系统用户)	主要包括系统中用户类型、用户名、主目录等
Logged In Accounts (用户登录)	主要包括用户名、登录时间等
Recent Logins (近期登录)	主要包括近期登录用户信息(登录名、登录 IP、登录时间)等

(5) APPS。可对主机上部署的应用程序等信息进行监控,如 PHP、Node、MySQL、MongoDB、 Python、Memcached 缓存、Redis 缓存以及 PM2(系统进程管理工具)等,如图 11-7-6 所示。Linux-Dash 监控主机应用程序情况的字段见表 11-7-7。

Linux Dash system status basic info network accounts apps Resources: <u>GitHub</u> | <u>Gitter Chat Room</u> | <u>Docs</u> ≡ Common Applications $\circlearrowleft \leftrightarrow$ = \blacksquare Memcached **(0** ↔ **(**-) ≡ Redis **v** ↔ • No data No data Search LOCATION INSTALLED BINARY /usr/bin/php /usr/lib64/php php true /usr/include/php node false false mongo false /usr/share/vim true

图 11-7-6 系统应用程序监控

表 11-7-7 应用程序监控内容

监控类型	监控说明
Common Applications(通用应用程序)	主要包括系统应用程序名称、安装位置以及安装状态等
Memcached (内存对象缓存)	主要包括主机缓存中的数据量信息等

78

项目十一

运维管理

续表

项目十-

监控类型	监控说明
Redis (远程字典服务)	主要包括客户端信息、连接数等
PM2(应用进程管理器)	主要包括 Node 应用、负载均衡信息等

【任务扩展】

Linux-Dash 概述如下。

Linux-Dash 是一个监控 Linux 服务器的 Web 化开源工具,可监控主机 CPU、内存、负载、网络、磁盘、用户等系统的运行状况,主要特性如下所示。

(1) 具有基于 Web 的美观、简单易读的 Linux 服务器监控面板。

(2) 工具文件较小, 通过 GitHub 下载不到 1MB。

(3) 支持添加新的监控模块与实时数据可视化更新。

(4) 支持在 PHP、Node.js、Python 和 Go 环境下安装,不同环境下的简要安装过程如下所示,可参照下列操作过程进行扩展应用。

操作命今:

```
1.
    # 第一步: 通过 GitHub 获取项目文件并进行解压缩
2.
    [root@Project-11-Task-07 ~]# wget https://github.com/tariqbuilds/linux-dash/archive/master.zip
3.
    [root@Project-11-Task-07 ~]# unzip master.zip
4.
    # 第二步: 配置防火墙, 允许 TCP 80 端口通过
5.
    [root@Project-11-Task-07 ~]# firewall-cmd --permanent --add-port=80/tcp
6.
    [root@Project-11-Task-07 ~]# firewall-cmd --reload
7.
                                                                                                  项目十-
8.
    # 第三步: 进入 Linux-Dash 的目录文件中
9.
10. [root@Project-11-Task-07 ~]# cd linux-dash-master/app/server
11.
12. # 第四步: 根据不同环境进行执行安装
13. # 在 Node.js 环境下执行安装(需安装 npm 软件包, 创建 Node 语言环境)
14. [root@Project-11-Task-07 server]# yum install -y npm
15. [root@Project-11-Task-07 server]# npm install --production
16. [root@Project-11-Task-07 server]# node index.js
17. Starting http server at: 0.0.0.0:80
18.
19. # 在 Go 环境下执行安装(需安装 go 软件包, 创建 Go 语言环境)
20. [root@Project-11-Task-07 server]# yum install -y go
21. [root@Project-11-Task-07 server]# go run index.go
22. Starting http server at: 0.0.0.0:80
23.
24. # 在 Python 环境下执行安装(需调整系统 Python 版本为 2.x, 创建 Python 语言环境)
25. [root@Project-11-Task-07 server]# python index.py
                                                                                         79
```

26.

27. # 第五步: 访问 Linux-Dash 程序。在本地主机上打开浏览器, 输入 Linux-Dash 主机地址即可看到系统 监控界面, 针对监控数据信息进行查看和分析

操作命令+配置文件+脚本程序+结束



(1)基于 Python 环境中安装 Linux-Dash 软件,其 Python 版本需为 2.x。
 (2) openEuler 操作系统安装完成后,Python 版本默认为 3.x,无法直接进行
 Linux-Dash 的 Python 安装,需要调整系统 Python 版本。

任务八 使用 Monitorix 实现可视化系统监控



【任务介绍】

Monitorix 是一个开源免费、轻量级的系统监视工具。本任务通过对该软件的部署配置,实现 对主机系统进程、CPU、内存、网络、磁盘等多方面的性能可视化监控。

【任务目标】

- (1) 实现 Monitorix 软件的部署。
- (2) 实现对主机系统运行情况的监控。

【操作步骤】

80

步骤 1: 创建虚拟机并完成 openEuler 的安装。

在 VirtualBox 中创建虚拟机,完成 openEuler 的安装。虚拟机与操作系统的配置信息见表 11-7-1,注意虚拟机网卡的工作模式为桥接。

虚拟机配置	操作系统配置
虚拟机名称: VM-Project-11-Task-08-10.10.2.118	主机名: Project-11-Task-08
内存: 1GB	IP地址: 10.10.2.118
CPU: 1 颗 1 核心	子网掩码: 255.255.255.0
虚拟硬盘: 20GB	网关: 10.10.2.1
网卡:1块,桥接	DNS: 8.8.8.8

表 11-8-1 虚拟机与操作系统配置

步骤 2: 完成虚拟机的主机配置、网络配置及通信测试。

启动并登录虚拟机,依据表 11-8-1 完成主机名和网络的配置,能够访问互联网和本地主机。

项目十一

运维管理 项目十一

(1)虚拟机的创建、操作系统的安装、主机名与网络的配置,具体方法参见项目一。



(2)建议通过虚拟机复制快速创建所需环境。通过复制创建的虚拟机需依据本任务虚拟机与操作系统规划配置信息设置主机名与网络,实现对互联网和本地主机的访问。

步骤 3: 安装部署的基本环境。

本任务通过编译安装 Monitorix 监控软件,需先设置 SELinux 模式为关闭状态,并安装软件所 需的依赖包和监控探测工具,其操作过程如下所示。

操作命令:

1.	# 设置 SELinux 模式(关闭状态)
2.	[root@Project-11-Task-08 ~]# setenforce 0
3.	
4.	# 安装监控软件的依赖包和监控探测工具
5.	[root@Project-11-Task-08 ~]# yum install -y perl-CGI perl-Config-General perl-DBI perl-HTTP-Server-Si
	mple perl-MailTools perl-MIME-Lite perl-XML-Simple rrdtool hddtemp lm_sensors perl-DBD-MySQL
	smartmontools tar make
6.	# 为了排版方便此处省略了部分信息
7.	# 下述信息说明将会安装以下软件,且已安装成功
8.	Installed:
9.	gc-8.0.6-3.0e2203sp2.x86_64
10.	# 为了排版方便此处省略了部分安装的软件列表信息
11.	xcb-util-0.4.0-14.oe2203sp2.x86_64
12.	
12	Completel

13. Complete!

操作命令+配置文件+脚本程序+结束

近日・

81

步骤 4: 安装 Monitorix 程序。

使用 wget 工具从 github 仓库中下载 Monitorix 程序并进行编译安装。

操作命令:

```
1.
    # 使用 wget 工具下载 Monitorix 到指定目录
2.
    [root@Project-11-Task-08 ~]# wget https://www.monitorix.org/monitorix-3.15.0.tar.gz
    --2023-10-11 22:29:14-- https://www.monitorix.org/monitorix-3.15.0.tar.gz
3.
4.
   正在解析主机 www.monitorix.org (www.monitorix.org)... 88.99.13.26
5.
    正在连接 www.monitorix.org (www.monitorix.org)|88.99.13.26|:443... 已连接
    已发出 HTTP 请求, 正在等待回应... 200 OK
6.
7.
    长度: 404645 (395K) [application/x-gzip]
8.
    正在保存至: "monitorix-3.15.0.tar.gz"
9.
                              100%[=====>] 395.16K 75.8kB/s 用时 5.7s
10. monitorix-3.15.0.tar.gz
11. # 下述信息表示文件下载成功
12. 2023-07-11 22:29:22 (69.6 kB/s)-已保存 "monitorix-3.15.0.tar.gz" [404645/404645])
13.
14.
```

- 15. # 使用 tar 工具将~/monitorix-3.15.0.tar.gz 文件解压当前目录下
- 16. [root@Project-11-Task-08 ~]# tar -zxvf monitorix-3.15.0.tar.gz
- 17. 18.
- 19. # 进入软件目录进行编译安装
- 20. [root@Project-11-Task-08 ~]# cd monitorix-3.15.0
- 21. [root@Project-11-Task-08 monitorix-3.15.0]# make install-systemd-all
- 22. Installing script and modules...
- 23. install -p -d "/usr/bin"
- 24. install -p -m755 monitorix "/usr/bin/monitorix"
- 25. # 为了排版方便此处省略了部分编译安装过程
- 26. Installing systemd service...
- 27. install -p -d "/etc"
- 28. install -p -d "/usr/lib/systemd/system"
- 29. # 安装此处表明 Monitorix 程序与服务已经安装完成
- 30. install -p -m644 docs/monitorix.service "/usr/lib/systemd/system/monitorix.service"

操作命令+配置文件+脚本程序+结束

步骤 5:发布 Monitorix 程序。

本任务使用 8080 端口发布 Monitorix 软件,具体步骤如下。

- (1) 启动 Monitorix 服务,使 Monitorix 监控程序运行,并设置服务开机自启动。
- (2) 配置防火墙规则,允许通过 TCP 8080 端口进行访问程序。

(3)访问 Monitorix 程序。在本地主机打开浏览器,输入 http://主机地址:8080/monitorix 即可访问监控系统界面,如图 11-8-1 所示。依次选择主机名、图表类型、监控时间节点后单击"OK"按钮,进入系统监控主界面,如图 11-8-2 所示。

操作命令:

82

项目十-

1.	# 启动 Monitorix 服务
2.	[root@Project-11-Task-08 ~]# systemet1 start monitorix
3.	# 验证服务状态
4.	[root@Project-11-Task-08 ~]# systemet1 status monitorix
5.	• monitorix.service - Monitorix
6.	Loaded: loaded (/usr/lib/systemd/system/monitorix.service; disabled; vendo>
7.	#running 代表服务正常运行
8.	Active: active (running) since Wed 2023-10-11 22:39:06 CST; 32s ago
9.	Docs: man:monitorix(8)
10.	Process: 16676 ExecStart=/usr/bin/monitorix -c /etc/monitorix/monitorix.con>
11.	Main PID: 16677 (/usr/bin/monito)
12.	Tasks: 2 (limit: 5968)
13.	Memory: 77.1M
14.	CGroup: /system.slice/monitorix.service
15.	└── 16677 "/usr/bin/monitorix -c /etc/monitorix/monitorix.conf -p />
16.	└── 16742 "monitorix-httpd listening on 8080"
17.	# 为了排版方便此处省略了部分服务状态信息
18.	

	运维管理	项目十一
19.		
20.	# 设置 Monitorix 服务为开机自启动	
21.	[root@Project-11-Task-08 ~]# systemetl enable monitorix	
22.	Created symlink /etc/systemd/system/multi-user.target.wants/monitorix.service → /usr/	ib/systemd/system/m
	onitorix.service.	
23.	# 验证服务是否为开机自启动	
24.	[root@Project-11-Task-08 ~]# systemetl is-enabled monitorix	
25.	# enabled 代表服务已经配置为开机自启动	
26.	enabled	
27.		
28.		
29.	. # 配置防火墙规则(允许通过 TCP 8080 端口访问)	
30.	[root@Project-11-Task-08 ~]# firewall-cmdadd-port=8080/tcppermanent	
31.	success	
32.	# 使防火墙规则配置生效	
33.	[root@Project-11-Task-08 ~]# firewall-cmdreload	
34.	success	
	操作命令+配	置文件+脚本程序+结束



图 11-8-1 Monitorix 监控系统

步骤 6: 阅读 Monitorix 监控信息。

Monitorix 可实现主机全方位监控,监控涉及类型为 System load average and usage (系统平均 负载和使用情况)、Global kernel usage (内核使用情况)、Filesystem usage and I/O activity (文件系 统使用和 I/O 状态)、eth0 Network traffic and usage (网卡流量)、Netstat statistics (网络连接状态统 计)、System services demand (系统服务运行状态)、Network port traffic (网络端口流量)、Users using the system (系统用户)、Devices interrupt activity (系统中断运行统计)。

如图 11-8-2 所示为 Monitorix 监控主界面。



项目十-





图 11-8-2 Monitorix 监控主界面

(1) System load average and usage。可对主机的内存使用情况、CPU 负载、系统开机时间等 信息进行监控,如图 11-8-3 所示。Monitorix 监控主机系统负载的字段见表 11-8-2。



项目十一

监控类型	监控说明
System load	监控系统 CPU 负载。最近 1min、5min、15min 平均 CPU 负载
Active processes	监控系统进程信息,主要呈现系统中进程数量与对应的进程状态
Memory allocation	内存使用监控。主要包含已使用、缓存、激活和未激活等内存大小
Entropy	监控 Linux 的熵值。熵用于衡量当前系统的结构,涵盖了所有因素,包括 系统吞吐量、资源管理、驱动程序、文件系统和性能等
Uptime	监控系统开机时间

表 11-8-2 系统负载与使用情况监控内容

(2) Global kernel usage。可对主机的内核使用情况、上下文交换情况等信息进行监控,如图 11-8-4 所示。Monitorix 监控主机内核使用情况的字段见表 11-8-3。



图 11-8-4 内核使用情况监控界面

表 11-8-3	内核使用情况监控内容
1× 11-0-5	们似仗用用机血红的谷

监控类型	监控说明
Kernel usage	内核使用情况,主要包括内核在操作过程中所占用 CPU 的负载百分比状态
Context switches and forks	监控系统上下文交换和系统调用情况
VFS usage	虚拟文件系统的使用情况

(3) Filesystem usage and I/O activity。可对主机文件系统以及磁盘 I/O 的使用情况进行监控, 如图 11-8-5 所示。Monitorix 监控主机文件系统运行情况的字段见表 11-8-4。

(4) eth0 Network traffic and usage。可对主机网卡流量情况进行实时监控,如图 11-8-6 所示。 Monitorix 监控主机网卡流量的字段见表 11-8-5。 项目十-



图 11-8-5 文件系统使用监控界面

表 11-8-4 文件系统监控内容

监控类型	监控说明
Filesystems usage	监控文件系统使用情况,主要包括根目录、交换分区目录、启动目录等
Disk I/O activity	监控不同目录的 I/O 状态
Inode usage	监控索引节点目录的使用情况
Time spent in I/O activity	监控 I/O 使用时花费时间





项目十一

86

- Anna

运维管理

项目十一

监控类型监控说明eth0 FastEthernet LAN监控网卡接收和输出的数据包流量大小eth0 Network packets监控网卡接收和输出的数据包总数eth0 Network errors监控网卡接收和输出的错误数据包数

表 11-8-5 网卡流量监控内容

(5) Netstat statistics。可对主机不同网络类型连接状态情况进行实时监控,如图 11-8-7 所示。 Monitorix 监控主机网络状态的字段见表 11-8-6。



图 11-8-7 网络连接状态监控界面

监控类型	监控说明
IPv4 states	监控 IPv4 网络连接状态的统计数量,主要包含关闭状态数、监听状态数、发送 请求连接数、接收连接请求数、打开链接数、等待远程 TCP 连接中断请求数、 从远程 TCP 等待连接中断连接数
IPv6 states	监控 IPv6 网络连接状态的统计数量,监控的状态统计与 IPv4 states 相同
Active Close	监控网络连接中主动关闭状态的统计数量
Passive Close	监控网络连接中被动关闭状态的统计数量
UDP statistics	监控网络中 UDP 协议的统计数量

Contraction of the second

87

(6) System services demand。可对主机不同服务(如 SSH、FTP、IMAP 等)调度命中情况进 行实时监控,如图 11-8-8 所示。



图 11-8-8 系统服务调度状态监控界面

(7) Network port traffic。可对主机不同服务(如 SSH、FTP、IMAP 等)端口网络流量进行 实时监控,如图 11-8-9 所示。



图 11-8-9 服务网络流量监控界面

(8) Users using the system。可对主机系统和服务用户情况进行实时监控,如图 11-8-10 所示。 Monitorix 监控主机用户信息的字段见表 11-8-7。

88

项目十一



图 11-8-10 系统用户监控界面

表 11-8-7 系统用户监控内容

监控类型	监控说明
Users logged in	监控当前系统登录用户数量信息
Samba users	监控系统 Samba 服务的用户数量信息
Netatalk users	监控系统 AFP 服务的用户数量信息

(9) Devices interrupt activity。可对主机程序运行中断情况进行实时监控,如图 11-8-11 所示。 Monitorix 监控主机中断情况的字段见表 11-8-8。



项目十--

Charles and

89

图 11-8-11 设备中断活动监控界面

表 11-8-8 设备中断活动监控内容

监控类型	监控说明
Interrupt activity	监控中断程序活动次数
Core activity	监控内核程序活动次数

【任务扩展】

Monitorix 是一个开源免费、轻量级的系统监视工具。该工具主要包括两个程序:一个是自动 收集 Perl 服务程序为 monitorix;另一个为 CGI 脚本程序为 monitorix.cgi。从 Monitorix 3.0 版本以 后,该工具已经内置 HTTP 服务器(占用端口为 TCP 8080),安装后可自动发布程序进行访问。

Monitorix 在系统运行的过程中会有规律地收集系统和网络的信息并以图形化的形式展示出来,主要特性如下所示。

(1)能够监测系统多种类型的运行状态,如 CPU 负载、内存使用、磁盘驱动器的温度和健康 情况、网络流量、Web 服务器、数据库以及系统多种服务等(如 SSH、FTP、SMTP、POP3、IMAP、 POP3 等)。

(2)能够按照每天、每周、每月或每年,通过图形或明文表格查看统计数字。

(3)能够缩放图形,以便更清楚地查看数据信息。

(4) 内置的 HTTP 服务器,程序安装运行后通过 Web 方式访问。

 90	1.000	

五日十一

项目十三 使用图形界面管理 openEuler

任务五 使用 Cockpit 实现多主机集中管理



【任务介绍】

本任务通过在 Cockpit 管理软件中添加其他主机系统,实现多主机集中运维管理。 本任务在任务三的基础上进行。

【任务目标】

- (1) 实现在线安装 Cockpit。
- (2) 实现使用 Cockpit 添加主机进行管理。
- (3) 实现使用 Cockpit 对多主机集中运维管理。

【操作步骤】

步骤 1: 主机安装 Cockpit 软件。

如果使用 Cockpit 进行多主机集中运维管理,必须在所有主机上均安装 Cockpit。

根据任务三操作步骤,完成对任务一主机(主机名: Project-13-Task-01, IP 地址: 10.10.2.131) 和任务二主机(主机名: Project-13-Task-02, IP 地址: 10.10.2.132)的 Cockpit 软件的安装,具体 操作过程如下。

操作命令:

```
1. # 在任务一主机 (Project-13-Task-01) 上操作
```

```
2. # 安装 Cockpit 软件
```

- 3. [root@Project-13-Task-01 ~]# yum install -y cockpit
- 4. # 为了排版方便此处省略了安装过程信息

5.

- 6. # 配置 Cockpit 服务
- 7. [root@Project-13-Task-01 ~]# systemctl start cockpit.socket
- 8. [root@Project-13-Task-01 ~]# systemctl enable cockpit.socket
- Created symlink /etc/system/sockets.target.wants/cockpit.socket → /usr/lib/system/system/cockpit. socket.

10. 11.

- 12. # 在任务二主机 (Project-13-Task-02) 上操作
- 13. # 安装 Cockpit 软件
- 14. [root@Project-13-Task-02 ~]# yum install -y cockpit
- 15. # 为了排版方便此处省略了安装过程信息
- 16.
- 17. # 配置 Cockpit 服务
- 18. [root@Project-13-Task-02 ~]# systemctl start cockpit.socket
- 19. [root@Project-13-Task-02 ~]# systemctl enable cockpit.socket
- 20. Created symlink /etc/system/sockets.target.wants/cockpit.socket → /usr/lib/system/system/cockpit. socket.

操作命令+配置文件+脚本程序+结束



(1)添加主机进行集中管理时,被控主机需安装 Cockpit 服务且保证服务运行, 同时主控主机和被控主机之间通过 SSH 进行通信,需要确保通信畅通。

(2)由于添加的被控主机不需要在本地客户端上通过浏览器访问 Cockpit 的 Web 控制台,所以在本步骤中不再进行防火墙规则配置。

步骤 2: 添加集中管理的主机。

单击左侧导航中"仪表盘"选项,可查看已经存在一台本地主机(Project-13-Task-03),如图 13-5-1 所示。

92

+





单击 "+" 添加按钮,输入添加主机的 IP 地址(图 13-5-2),添加主机时需确认私钥信息及输入主机的密码权限信息(图 13-5-3)。根据操作步骤完成对任务一和任务二的主机进行添加管理。



图 13-5-2 输入添加其他主机

COCKPIT					🛓 root ~
-	CPU 内存 网络 磁盘读/写	登灵到 10.10.2.131		×	
-	100%	Cockpit 无法登录到 10.10.2	2.131、可以变更以下认证凭证,也许想要们	步账号和密码。	
	80%	用户名 root		0	
	60%	认证 输入密码		×	
	40%	卷码		0	
	20%			REGIS \$2532	
	0%	16:28	16:29	16:30	16:31
	服用器				Image: A state of the state
	Project-13-Task-03				openEuler 22.03 (LTS-SP2)

图 13-5-3 输入主机权限信息

步骤 3:集中查看主机的运行情况。

单击左侧导航中"仪表盘"选项,可集中查看管理主机的运行情况,主要包括 CPU、内存、 网络和磁盘 IO 等情况,通过集中对比可直观查看主机运行性能是否存在异常情况,如图 13-5-4、 图 13-5-5、图 13-5-6、图 13-5-7 所示。

项目十三

93

- Charles



图 13-5-4 集中查看主机 CPU 运行情况











图 13-5-7 集中查看主机磁盘 IO 情况

步骤4:集中管理时主机切换。

94

项目十三

在左侧导航"服务器"选项中,单击主机后的下拉列表,可查看被控主机的列表。根据需要选

使用图形界面管理 openEuler

项目十三

% of 1 CPU core 硬件 inr VirtualBox 产标记 0 • 机器编号 729446a797 操作系统 openEuler 22.03 (LTS Proi SP2) MiB 内存 System Up To Date 显示描印 384 256 128 主机名 Project-13-Tas 884 她 加入她 间 2024-01-06 16: 电源选项 重启 🖌 MiB/s 花泉 達/画 C Enable : Kbps 网络流量 800

择相应主机的名称后单击,进行被控主机的切换,如图 13-5-8 所示,切换后如图 13-5-9 所示。

图 13-5-8 管理主机切换



图 13-5-9 切换后的管理主机

步骤 5: 集中对主机进行运维管理。

(1)为任务一主机添加用户。将默认管理主机切换为任务一主机(Project-13-Task-01),根据 任务四中 Cockpit 管理工具的运维操作方法,对任务一主机添加账户信息,如图 13-5-10、图 13-5-11 所示。

(2)为任务二主机添加防火墙规则。将默认管理主机切换为任务二主机 (Project-13-Task-02), 根据任务四中 Cockpit 管理工具的运维操作方法,对任务二主机添加"允许访问 Cockpit 服务通过 防火墙"规则信息,如图 13-5-12、图 13-5-13 所示。 项目十三

95

Concernant and

项目十三

96

соскріт			
-	🖹 Project-13-Ta 🗸	的建筑资产	
		openeuler root root	
	存储		
	网络		
	账户		
	服务		
	内核转储		
	软件更新		
	终端		

图 13-5-10 查看系统默认账户

COCKPI	т		🛓 root 🗸
-	Project-13-Ta 🗸	展户 ≥ task1	
æ	系统	hites htes	会话 翻除
	日志 开経 発売 第59 かはH線 比代更新 広明 作38	全 な set	
	Diagnostic Reports SELinux	(神代公共 55H 卷明) 存在未能成本小教氏的公明。	•

图 13-5-11 为任务一主机添加用户

оскріт					🛓 root -
<u></u>	🗐 Project-13-Ta 🗸	网络 » Firewall			
	系统	Firewall #			
	日志	Allowed Services			Add Service
	存储	服务	TCP	UDP	
	网络	DHCPv6 Client		546	
	账 丹	Multicast DNS (mDNS)		5353	
	内核转储	SSH	22		
	软件更新				
	应用				
	终端				
	Diagnostic Reports				
	SELinux				

图 13-5-12 查看系统默认防火墙规则

			使用图形界	面管理 openEuler	项目十三
соскріт	r				≛ root∽
	🗏 Project-13-Ta 🗸	网络 > Firewall			
		Firewall #			
		Allowed Services			Add Services
	存储	服务	TCP	UDP	
	网络 新白	✓ Cockpit	9090		
		评值			
	内核转储	cocipicies you access and configure your server remotely.			
	软件更新	DHCPv6 Client		546	1
		Multicast DNS (mDNS)		5353	
	终端 Diagnostic Reports	SSH	22		

图 13-5-13 为任务二主机添加防火墙规则

