

# 项目十一

---

## 运维管理

### 任务七 使用 Linux-Dash 实现可视化实时监控



扫码看视频

#### 【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

**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。



**提醒**

(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 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。

表 11-7-2 Linux-Dash 软件 PHP 环境部署基本条件表

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)

(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. # 使用 yum 工具安装 Apache
2. [root@Project-11-Task-07 ~]# 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/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/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.

```

```
15. # 使用 yum 工具安装 PHP 解析器（本次安装 PHP 版本为 8.0）
16. [root@Project-11-Task-07 ~]# yum install -y php php-devel
17.
18. # 设置 SELinux 模式（关闭状态）
19. [root@Project-11-Task-07 ~]# setenforce 0
20.
21. # 配置防火墙规则（允许 http 协议访问）
22. [root@Project-11-Task-07 ~]# firewall-cmd --add-service=http --permanent
23. success
24. # 使防火墙规则配置生效
25. [root@Project-11-Task-07 ~]# firewall-cmd --reload
26. success
27.
28. # 使用 php -v 命令验证已安装的 PHP 版本
29. [root@Project-11-Task-07 ~]# php -v
30. PHP 8.0.28 (cli) (built: Feb 14 2023 11:05:44) ( NTS )
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 Apache 版本
35. [root@Project-11-Task-07 ~]# httpd -v
36. Server version: Apache/2.4.51 (Unix)
37. Server built:   Mar 13 2023 16:30:49
38.
39. # 使用 sestatus 验证 SELinux 模式
40. [root@Project-11-Task-07 ~]# sestatus
41. SELinux status:                enabled
42. SELinuxfs mount:              /sys/fs/selinux
43. SELinux root directory:       /etc/selinux
44. Loaded policy name:            targeted
45. # 查看当前 SELinux 的状态为 permissive（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 firewall-cmd 命令查看防火墙允许通过服务规则
54. [root@Project-11-Task-07 ~]# firewall-cmd --list-services
55. dhcpv6-client http mdns ssh
```

操作命令+配置文件+脚本程序+结束



小贴士

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/tariqbuidis/linux-dash/archive/master.zip
3. --2023-07-11 14:59:02-- https://github.com/tariqbuidis/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/tariqbuidis/linux-dash/zip/refs/heads/master [跟随至新的 URL]
8. --2023-07-11 14:59:02-- https://codeload.github.com/tariqbuidis/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" >
```

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。



提醒

本步骤修改 Apache 配置文件以发布监控程序，修改配置文件具体方法参见项目四。

(3) 重新启动 httpd 服务，使 Apache 服务配置生效。

操作命令：

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。

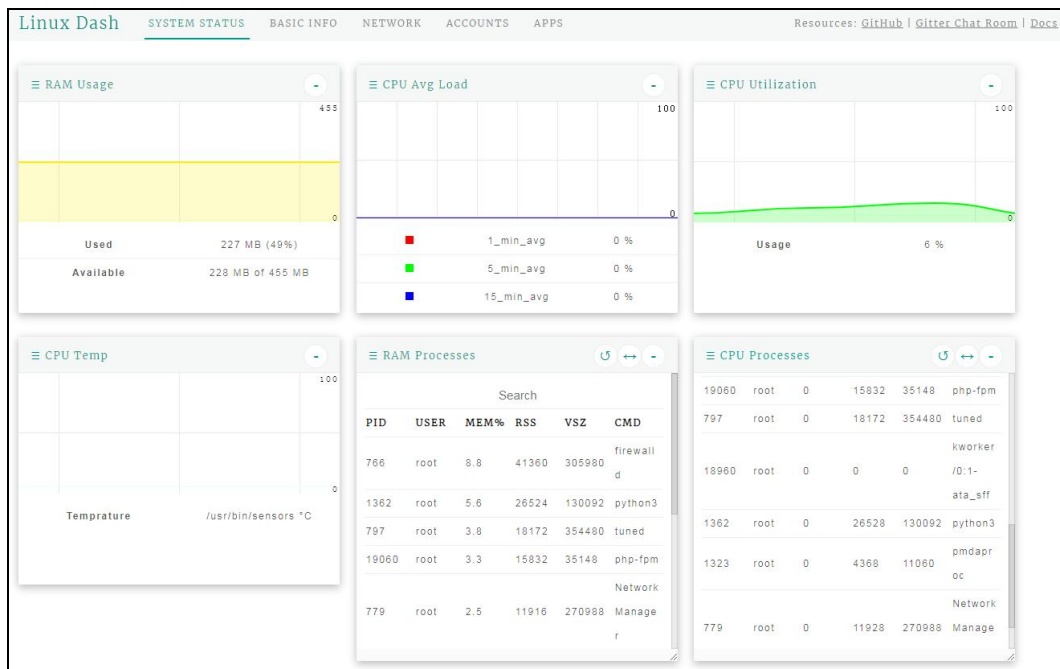


图 11-7-1 Linux-Dash 主界面

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。

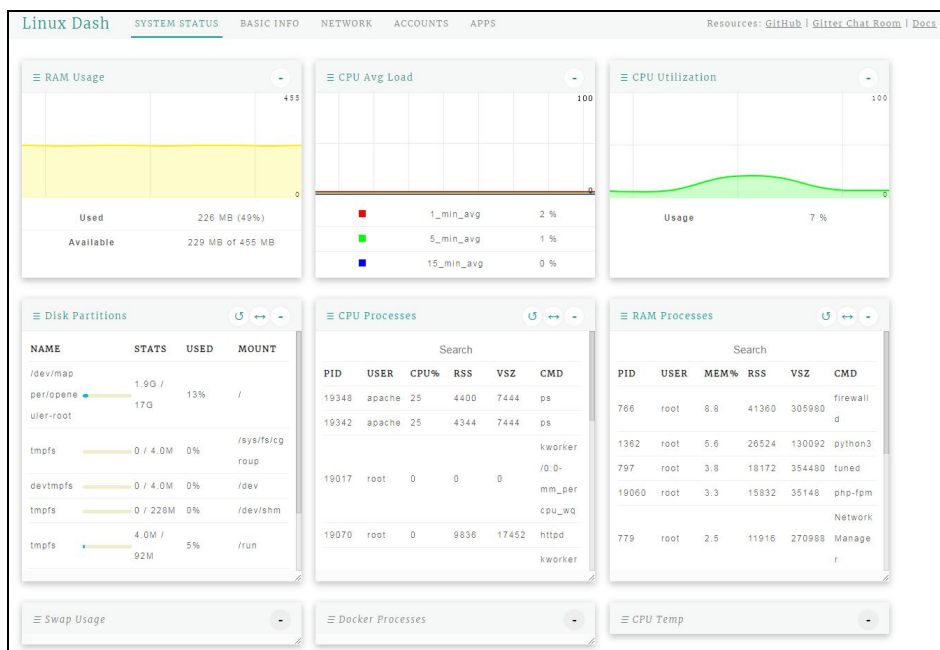


图 11-7-2 系统状态监控界面

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

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。

表 11-7-4 Linux-Dash 监控主机基本信息的字段

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

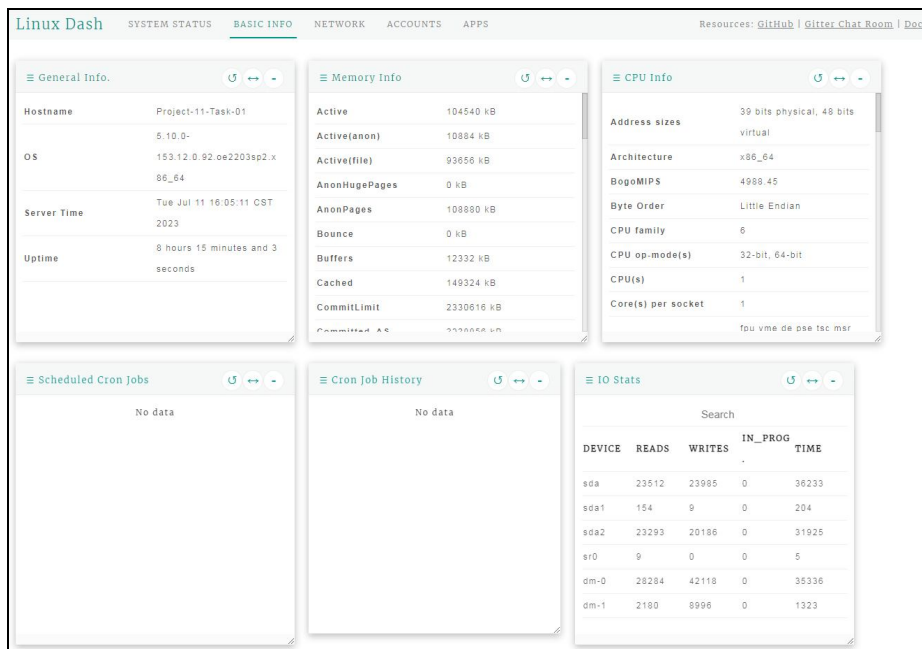


图 11-7-3 基本信息监控界面

(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。

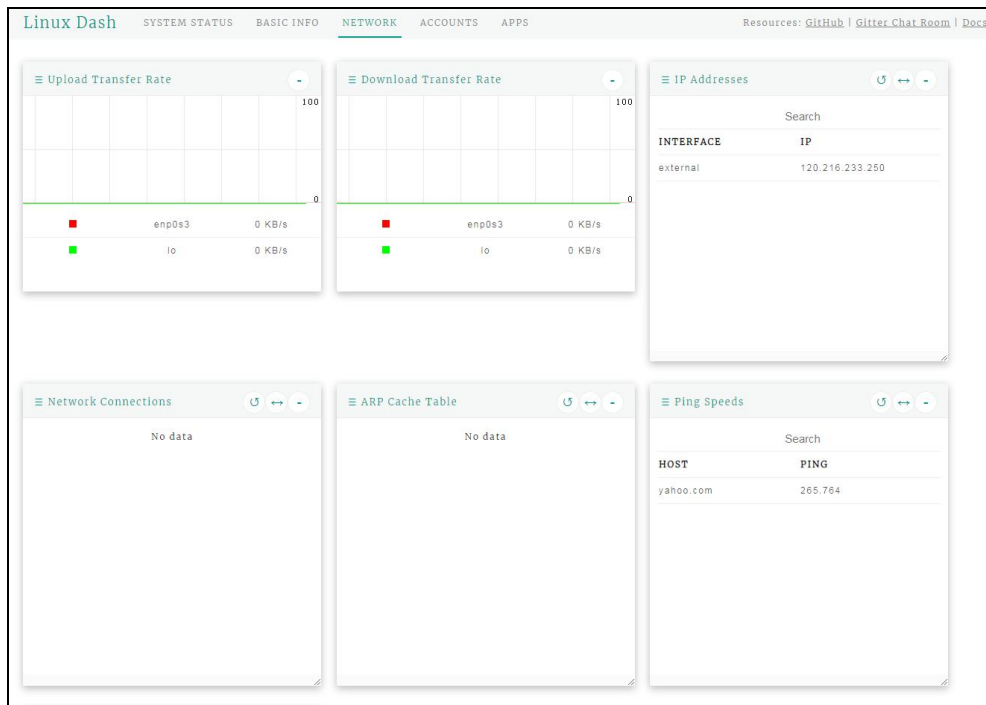


图 11-7-4 网络信息监控界面

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。



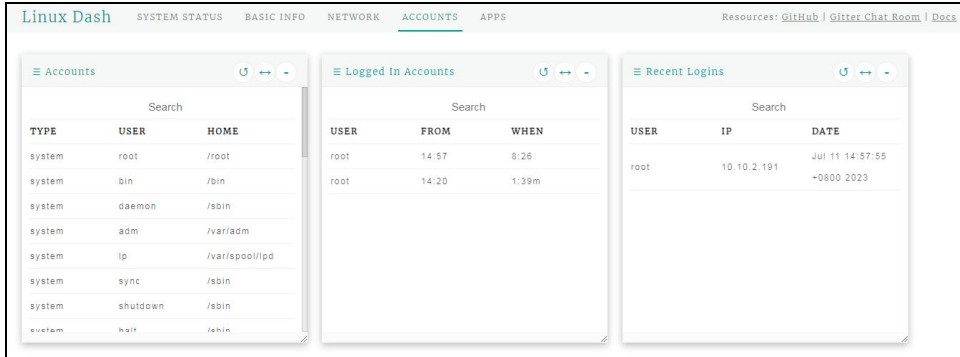


图 11-7-5 系统用户访问监控

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。

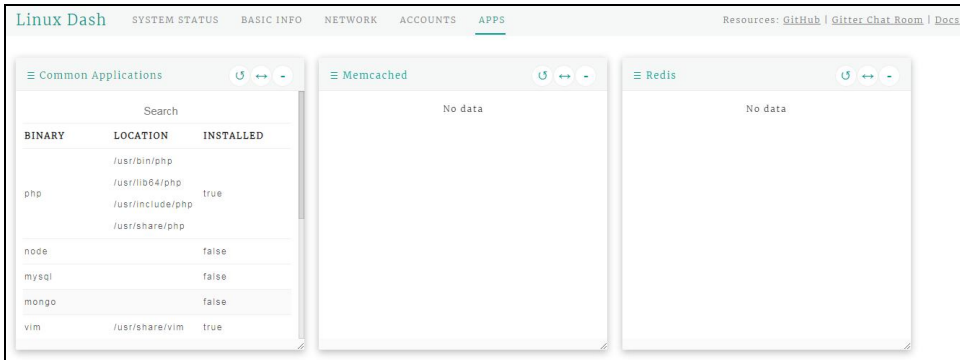


图 11-7-6 系统应用程序监控

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

续表

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。

操作命令：

```

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.
5. # 第二步：配置防火墙，允许 TCP 80 端口通过
6. [root@Project-11-Task-07 ~]# firewall-cmd --permanent --add-port=80/tcp
7. [root@Project-11-Task-07 ~]# firewall-cmd --reload
8.
9. # 第三步：进入 Linux-Dash 的目录文件中
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

```

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



提醒

(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



扫码看视频

### 【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】

**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 11-8-1 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-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

**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。