[Top](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/DEVOPS/DAY02/CASE/01/index.html" \l "page_top_case)

# NSD Devops DAY02

1. [案例1：通过本机发送邮件](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/DEVOPS/DAY02/CASE/01/index.html" \l "case1)
2. [案例2：通过互联网服务器发送邮件](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/DEVOPS/DAY02/CASE/01/index.html" \l "case2)
3. [案例3：天气预报查询](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/DEVOPS/DAY02/CASE/01/index.html" \l "case3)
4. [案例4：钉钉机器人](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/DEVOPS/DAY02/CASE/01/index.html" \l "case4)
5. [案例5：获取zabbix版本信息](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/DEVOPS/DAY02/CASE/01/index.html" \l "case5)
6. [案例6：获取令牌](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/DEVOPS/DAY02/CASE/01/index.html" \l "case6)
7. [案例7：创建主机](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/DEVOPS/DAY02/CASE/01/index.html" \l "case7)

## 1 案例1：通过本机发送邮件

### 1.1 问题

编写一个send\_mail.py脚本，实现以下功能：

1. 创建bob和alice帐户
2. 编写发送邮件件程序，发件人为root，收件人是本机的bob和alice帐户

### 1.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：创建bob和alice帐户

1. [root@ localhost day12]# ls /home/
2. Student zabbix zhangsan
3. [root@localhost day12]# useradd bob
4. [root@localhost day12]# useradd alice
5. [root@localhost day12]# ls /home/
6. alice bob Student zabbix zhangsan

步骤二：编写发送邮件件程序，发件人为root，收件人是本机的bob和alice帐户

1. [root@ localhost day12]# vim send\_mail.py
2. import smtplib
3. from email.mime.text import MIMEText
4. from email.header import Header
5. #邮件正文有三个参数：第一个为文本内容，第二个设置文本格式plain，第三个utf-8设置编码保证多语言兼容性
6. message = MIMEText('Python邮件发送测试\n', 'plain', 'utf8')
7. 标准邮件需要三个头部信息： From, To, 和 Subject
8. #发送者信息
9. message['From'] = Header('root@localhost', 'utf8')
10. #接收者信息
11. message['To'] = Header('bob@localhost', 'utf8')
12. #主题信息
13. message['Subject'] = Header('mail test', 'utf8')
14. sender = 'root@redhat.com'        #发送方
15. receivers = ['bob@localhost', 'alice@126.com']    #收件方
16. smtp\_obj = smtplib.SMTP('localhost')    #用localhost发邮件
17. # smtplib负责发送邮件
18. smtp\_obj.sendmail(sender, receivers, message.as\_string())

SMTP是发送邮件的协议，Python内置对SMTP的支持，可以发送纯文本邮件、HTML邮件以及带附件的邮件。

Python对SMTP支持有smtplib和email两个模块，email负责构造邮件，smtplib负责发送邮件。

Python SMTP 对象使用 sendmail 方法发送邮件：

1. smtp\_obj.sendmail(sender, receivers, message.as\_string())

参数说明：

sender: 邮件发送者地址。

receivers: 字符串列表，邮件发送地址。

message.as\_string(): 发送消息 ，str模式

由于可以一次发给多个人，所以recives传入一个列表，邮件正文是一个str，as\_string()把MIMEText对象变成str。

步骤三：测试脚本执行

1. [root@ localhost day12]# python3 send\_mail.py
2. [root@ localhost day12]# mail –u bob
3. Heirloom Mail version 12.5 7/5/10. Type ? for help.
4. “/var/mail/bob”: 1 message 1 new
5. >N 1 =?utf8?q?root=4Oloca Mon Jul 30 09:36 18?663 “”
6. & 1
7. From root@redhat.com Mon Jul 30 09:36:44 2018
8. Return- Path: <root@redhat.com>
9. X- Original- To: bob@localhost.tedu.cn
10. Content- Type: text/plain; charset=“utf8”
11. From: root@localhost@room8pc16.tedu.cn
12. To: bob@localhost@room8pc16.tedu.cn
13. Subject: mail test
14. Date: Mon, 30 Jul 2018 09:36:44 +0800 (CST)
15. Status: R
16. Python邮件发送测试
17. &

## 2 案例2：通过互联网服务器发送邮件

### 2.1 问题

编写一个mail\_client.py脚本，实现以下功能：

1. 通过自己互联网注册的邮箱，为其他同学互联网邮箱发邮件

### 2.2 方案

导入sys模块，用sys.argv方法获取get\_web函数实参，让用户在命令行上提供http://www.tedu.cn和/tmp/tedu.html两个参数，调用get\_web函数实现如下功能：

1)导入urllib模块，使用urllib模块的urlopen函数打开url（即网址），赋值给html

2)以写方式打开/tmp/tedu.html文件

3)以循环方式：

读html获取的数据，保存到data

将data写入/tmp/tedu.html

4)关闭html

### 2.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：环境准备

使用SMTP协议发送的邮件，需要先查看您的发件人邮箱是否有开启SMTP协议，如没有需要开启，测试使用的是126.com的邮箱作为发信人邮箱，开启SMTP协议如下

1. 先登录到126.com邮箱，如图-1所示：

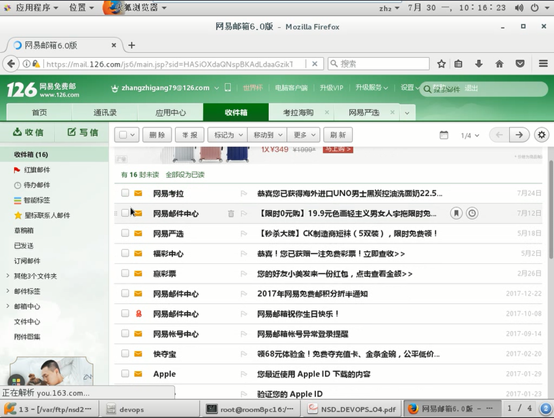


图-1

2. 看到邮箱上面的功能栏中有一个“设置”的选项，单击该选项，然后选择下拉菜单的“POP3/SMTP/IMAP”，如图-2所示：

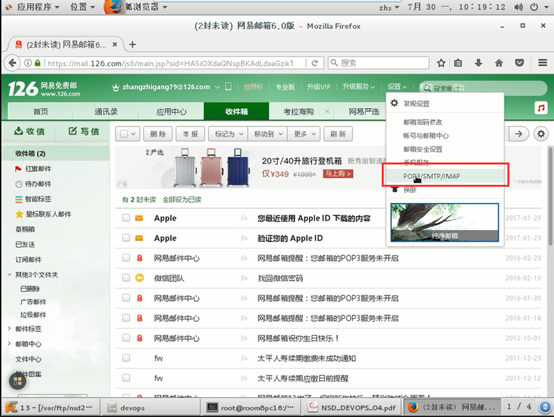


图-2

3. 如图-3所示，上面红框的两个必须勾选上，如没有勾选，要选择开启就可以勾选上了：

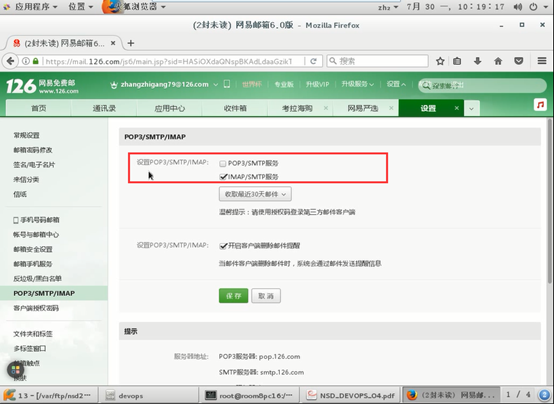


图-3

4.页面向下可以可以看到下图-4红框里是：SMTP服务器是:smtp.126.com：

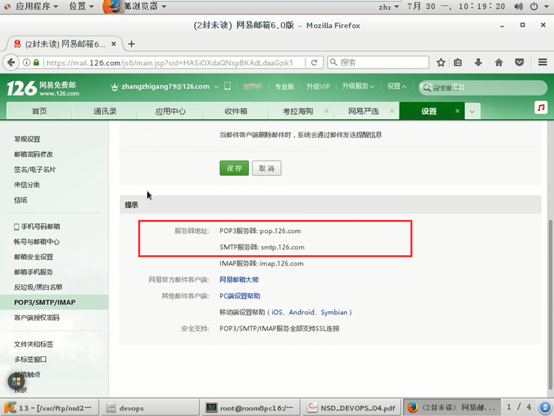


图-4

步骤二：编写脚本

1. [root@ localhost day12]# vim mail\_client.py
2. #!/usr/bin/env python3
3. import smtplib
4. from getpass import getpass
5. from email.mime.text import MIMEText
6. from email.header import Header
7. mail\_host = 'smtp.126.com'        #发件人邮箱账号
8. mail\_user = 'zhangzhigang79@126.com'        #收件人邮箱账号
9. mail\_pwd = getpass()        #获取密码
10. #邮件正文有三个参数：第一个为文本内容，第二个设置文本格式plain，第三个utf-8设置编码保证多语言兼容性
11. message = MIMEText('Python邮件发送测试\n', 'plain', 'utf8')
12. #发送者信息
13. message['From'] = Header('zhangzhigang79@126.com', 'utf8')
14. #接收者信息
15. message['To'] = Header('zhangzhigang79@126.com', 'utf8')
16. #主题信息
17. message['Subject'] = Header('python 1802 mail test', 'utf8')
18. sender = 'zhangzhigang79@126.com'        #发送方
19. receivers = ['zhangzhigang79@126.com']        #接收方
20. smtp\_obj = smtplib.SMTP()        #创建SMTP对象
21. smtp\_obj.connect(mail\_host)    #将SMTP对象与发送人邮件简历连接建立连接
22. smtp\_obj.login(mail\_user, mail\_pwd)        #登录用户名密码
23. # SMTP 对象使用 sendmail 方法发送邮件
24. smtp\_obj.sendmail(sender, receivers, message.as\_string())

步骤三：测试脚本执行

1. [root@ localhost day12]# python3 mail\_client.py
2. Password：

如果发送成功，结果显示如图-5所示：

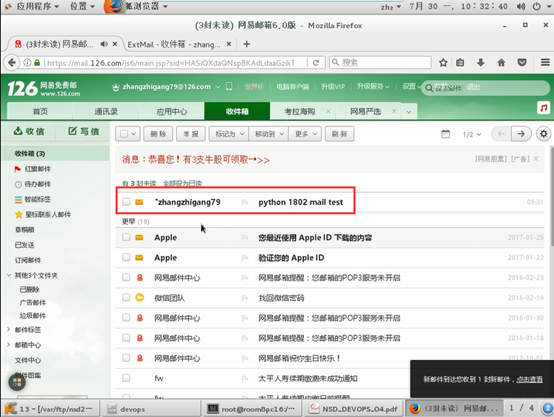


图-5

## 3 案例3：天气预报查询

### 3.1 问题

编写一个display\_weather.py脚本，实现以下功能：

1. 运行程序时，屏幕将出现你所在城市各区县名字
2. 用户指定查询某区县，屏幕上将出现该区县当前的气温、湿度、风向、风速等

### 3.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：找到天气信息规律

1.首先我们想要实现的功能是天气预报，从哪获取天气这是一个问题，在这里可以使用http://www.weather.com.cn/data/sk/101051301.html这个应用程序编程接口，101051301是城市的ID，可以到http://www.weather.com.cn/查看，替换后浏览器打开，如图-6所示，图示为json格式：



图-6

注意：图中看不懂的文字是编码问题

2.从图-1中可以看出cityid就是城市ID，temp是温度，SD是湿度，我们编写代码可以直接获取到网站相应信息，编写脚本wather.py：

1. [root@ localhost day12]# vim weather.py
2. #!/usr/bin/python #这里是python的目录
3. from urllib.request import urlopen
4. import json
5. #打开网页，使用urllib模块的urlopen函数打开url，赋值给html
6. html = urlopen('http://www.weather.com.cn/data/sk/101010100.html')
7. #读html获取的数据，保存到data
8. data = html.read()
9. #从data中获取我们想要的信息，json.loads()是将json格式数据转换为字典
10. #（可以理解为json.loads()函数是将字符串转化为字典）
11. print(json.loads(data))
12. #关闭html
13. html.close()

3.执行脚本结果如下：

1. [root@ localhost day12]# python3 weather.py
2. {‘weatherinfo’:{‘city’:‘北京’,‘cityid’:‘101010100’,‘temp’:‘27.9’,‘WD’:‘南风’,‘WS’:‘小于3级’,‘SD’:‘28%’,‘AP’:‘1002hPa’,‘njd’:‘暂无实况’,‘WSE’:‘<3’,‘time’:‘17:55’,‘sm’:‘2.1’,‘isRadar’:‘1’,‘Radar’:‘JC\_RADAR\_AZ9010\_JB’}}

从以上脚本执行结果中我们可以看到，从网站中获取到的数据是以字典形式显示，显示信息有城市、城市id，温度、风向等等，根据这种规律，编写下面代码

步骤二：编写代码实现如下功能

1.定义一个字典，该字典中键‘0’和‘1’对应的值为天气网址中城市对应的id，城市不同id则不同

2.运行程序时，屏幕将出现你所在城市名字

3.当用户指定查询某城市（即输入0或1时）

4.调用get\_weather函数，函数的实际参数为city\_codes字典对应值（即对应的城市id）

5. 打开天气网页，使用urllib模块的urlopen函数打开url，赋值给html

6.读html获取的数据，用json.loads()获取天气信息，获取到的信息为字典形式

7.从获取到的字典数据中提取气温、湿度、风向、风速等信息，保存在output变量中

8.将output变量作为get\_weather函数的返回值，打印在屏幕上

1. [root@ localhost day12]# vim display\_weather.py
2. #!/usr/bin/python
3. from urllib.request import urlopen
4. import json
5. def get\_weather(city\_code):        #定义一个输入城市id的函数
6. 5.打开天气网页，使用urllib模块的urlopen函数打开url，赋值给html
7. url = 'http://www.weather.com.cn/data/sk/%s.html' % city\_code
8. html = urlopen(url)
9. 6.读html获取的数据，用json.loads()获取我们想要的信息
10. #json.loads()是将json格式数据转换为字典
11. #（可以理解为json.loads()函数是将字符串转化为字典）
12. data = json.loads(html.read())
13. 7.output为返回值，即最终屏幕显示的信息
14. output = '风向：%s, 风力: %s， 温度：%s, 湿度：%s' % (
15. #data获取到的天气信息为字典，该字典中weatherinfo键对应的值还是一个字典，这个字典中‘WD’键对应的值是风向，‘WS’键对应的值是风力，'temp'键对应的值是温度，'SD'键对应的值是湿度，利用键值对关系将相应数据显示出来即可
16. data['weatherinfo']['WD'],
17. data['weatherinfo']['WS'],
18. data['weatherinfo']['temp'],
19. data['weatherinfo']['SD']
20. )
21. return output
22. if \_\_name\_\_ == '\_\_main\_\_':
23. 1.定义字典：键对应的值为天气网站网址接口中城市ID
24. city\_codes = { '0': '101010100', '1': '101121404'}
25. 2．代码执行后，屏幕给出的提示信息
26. prompt = """(0) 北京
27. (1) 台儿庄
28. 请选择(0/1): """
29. 3．根据提示信息，输入0或1
30. choice = input(prompt)
31. 4.调用get\_weather函数，其实际参数为city\_codes字典对应值
32. 8.打印调用get\_weather函数返回值
33. print(get\_weather(city\_codes[choice]))

步骤三：测试脚本执行

1. [root@ localhost day12]# python3 display\_weather.py
2. (0) 北京
3. (1) 台儿庄
4. 请选择(0/1): 0
5. 风向：南风， 风力： 小于3级， 温度：27.9， 湿度：28%
6. [root@ localhost day12]# python3 display\_weather.py
7. (0) 北京
8. (1) 台儿庄
9. 请选择(0/1): 1
10. 风向：东北风， 风力： 小于3级， 温度：22.3， 湿度：64%

## 4 案例4：钉钉机器人

### 4.1 问题

通过钉钉软件创建一个群聊机器人，要求：

1. 编写代码，通过python脚本实现钉钉机器人在群中发送消息
2. 使用json和requests模块

### 4.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：在钉钉中创建群聊机器人

/

图-7

/

图-8

/

图-9

/

图-10

/

图-11

步骤二：编写代码

1. [root@ localhost day12]# vim dingtalk.py
2. import json
3. import requests
4. import sys
5. def send\_msg(url, reminders, msg):
6. headers = {'Content-Type': 'application/json;charset=utf-8'}
7. data = {
8. "msgtype": "text", # 发送消息类型为文本
9. "at": {
10. "atMobiles": reminders,
11. "isAtAll": False, # 不@所有人
12. },
13. "text": {
14. "content": msg, # 消息正文
15. }
16. }
17. r = requests.post(url, data=json.dumps(data), headers=headers)
18. return r.text
19. if \_\_name\_\_ == '\_\_main\_\_':
20. msg = sys.argv[1]
21. reminders = ['15055667788'] # 特殊提醒要查看的人,就是@某人一下
22. url = 此处填写上面webhook的内容
23. print(send\_msg(url, reminders, msg))

步骤三：测试脚本执行

1. [root@ localhost day12]# python3 dingtalk.py "这只是一个测试而已"

/

图-12

## 5 案例5：获取zabbix版本信息

### 5.1 问题

编写一个zabbix\_apiversion.py脚本，实现以下功能：

1. 安装zabbix服务器
2. 获取zabbix api的url
3. 编写python程序，访问zabbix api，取得zabbix版本号

### 5.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：安装zabbix服务器

安装zabbix服务器具体步骤请参照NSD SECURITY DAY06 cookbook中案例2操作

步骤二：获取zabbix api的url

1)设置前端后，你就可以使用远程HTTP请求来调用API。为此，需要向api\_jsonrpc.php位于前端目录中的文件发送HTTP POST请求。如果你的Zabbix前端安装在http://192.168.4.2/zabbix，那么用HTTP请求来调用apiinfo.version方法就如下面这样：

1. POST 192.168.4.2/zabbix/api\_jsonrpc.php HTTP/1.1

2)从zabbix官方文档中获取 Zabbix API 版本，如图-7所示：

官方文档地址如下：

https://www.zabbix.com/documentation/3.4/zh/manual/api/reference/apiinfo/version



图-13

步骤三：编写脚本

在HTTP协议中，post提交的数据必须放在消息主体中，但是协议中并没有规定必须使用什么编码方式，从而导致了提交方式的不同。服务端根据请求头中的Content-Type字段来获知请求中的消息主体是用何种方式进行编码，再对消息主体进行解析。

请求的 Content-Type 头部必须设置为以下值之一：

application/json-rpc

application/json

application/jsonrequest

1. [root@localhost day12]# vim zabbix\_apiversion.py
2. #!/usr/bin/env python3
3. import requests
4. import json    # python中的dict类型要转换为json格式的数据需要用到json库
5. #要访问的网址
6. url = 'http://192.168.4.2/zabbix/api\_jsonrpc.php'
7. #请求头部信息
8. headers = {'Content-Type': 'application/json-rpc'}
9. # data是从官方文档处获得的
10. data = {
11. "jsonrpc": "2.0",    #jsonrpc协议的版本号，固定的
12. "method": "apiinfo.version",        #在zabbix手册上查到的，查询zabbix版本
13. "params": [],        #没有额外参数
14. "id": 1    #随便写个数字
15. }
16. #使用requests发送请求，访问指定网站，向url发送data请求，r收到的返回响应为json格式
17. #将data转成json格式，zabbix要求提交的数据是json格式
18. r = requests.post(url, headers=headers, data=json.dumps(data))
19. #将json格式解码，zabbix返回的数据都是json格式
20. print(r.json()))

需要注意的是python中并没有json类型这一说法，通过json.dumps(data)转换的字典对象，最后得到的是一个字符串对象，也就是说，在python中json格式的数据实际上就是一个字符串

步骤四：测试脚本执行

1. [root@localhost day12]# python3 zabbix\_apiversion.py
2. {"jsonrpc": "2.0", "result": "2.4.0", "id": 1}

## 6 案例6：获取令牌

### 6.1 问题

编写一个get\_token.py脚本，实现以下功能：

1. 编写get\_token函数
2. 该函数接受zabbix服务器url、用户名和密码作为参数
3. 函数返回值为用户令牌token

### 6.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：编写脚本get\_token.py，获取身份令牌

1)从zabbix官方文档中使用 user.login 方法登录并获取身份验证令牌请求，如图-8所示：

官方文档地址如下：

https://www.zabbix.com/documentation/3.4/zh/manual/api/reference/user/login



图-14

2)编写get\_token.py文件

1. [root@localhost day12]# vim get\_token.py
2. #!/usr/bin/env python3
3. import requests
4. import json    # python中的dict类型要转换为json格式的数据需要用到json库
5. #要访问的网址
6. url = 'http://192.168.4.2/zabbix/api\_jsonrpc.php'
7. #请求头部信息
8. headers = {'Content-Type': 'application/json-rpc'}
9. # data是从官方文档处获得的
10. data = {
11. # API使用的JSON-RPC协议的版本; Zabbix API实现JSON-RPC版本2.0
12. "jsonrpc": "2.0",
13. "method": "user.login",    #调用的API方法
14. # params将被传递给API方法的参数
15. "params": {
16. "user": "Admin",
17. "password": "zabbix"
18. },
19. "id": 1    #请求的任意标识符
20. }
21. #使用requests发送请求，访问指定网站，向url发送data请求，r收到的返回响应为json格式
22. #将data转成json格式
23. r = requests.post(url, headers=headers, data=json.dumps(data))
24. #将json格式解码
25. print(r.json()))

步骤四：测试脚本执行，获取用户令牌token

1. [root@localhost day12]# python3 get\_token.py
2. {"jsonrpc": "2.0", "result": "0424bd59b807674191e7d77572075f33", "id": 1}

## 7 案例7：创建主机

### 7.1 问题

编写一个remote\_comm.py脚本，实现以下功能：

1. 主机192.168.4.10已安装zabbix\_agent
2. 将该主机填加到zabbix监控的主机中
3. 主机属于Linux Servers组

### 7.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：编写脚本，检索组

1)从zabbix官方文档中使用hostgroup.get 方法获取主机组请求，如图-9所示：

官方文档地址如下：

https://www.zabbix.com/documentation/3.4/zh/manual/api/reference/hostgroup/get



图-15

2)编写get\_hostgroup.py文件，检索组

1. [root@localhost day12]# vim get\_hostgroup.py
2. #!/usr/bin/env python3
3. import requests
4. import json    # python中的dict类型要转换为json格式的数据需要用到json库
5. #要访问的网址
6. url = 'http://192.168.4.2/zabbix/api\_jsonrpc.php'
7. #请求头部信息
8. headers = {'Content-Type': 'application/json-rpc'}
9. # data是从官方文档处获得的
10. data = {
11. # API使用的JSON-RPC协议的版本; Zabbix API实现JSON-RPC版本2.0
12. "jsonrpc": "2.0",
13. "method": " hostgroup.get ",    #调用的API方法
14. # params将被传递给API方法的参数
15. "params": {
16. "output": "extend",
17. },
18. "auth": "0424bd59b807674191e7d77572075f33",    #之前获取到的令牌
19. "id": 1    #请求的任意标识符
20. }
21. #使用requests发送请求，访问指定网站，向url发送data请求，r收到的返回响应为json格式
22. #将data转成json格式
23. r = requests.post(url, headers=headers, data=json.dumps(data))
24. ginfo = r.json()
25. print(ginfo['result'])    #打印主机组信息
26. for item in ginfo['result']:
27. print(item['groupid'], item['name'])    #打印主机组id和名称

3)测试脚本执行，获取了主机名称及id，如图-10所示：

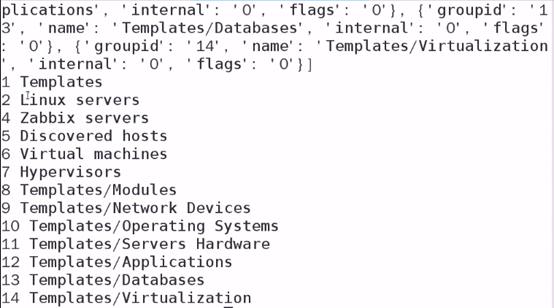


图-16

步骤二：编写脚本，检索模板

1)从zabbix官方文档中使用template.get 方法获取模板请求，如图-11所示：

官方文档地址如下：

https://www.zabbix.com/documentation/3.4/zh/manual/api/reference/template/get



图-17

2)编写get\_template.py文件

1. [root@localhost day12]# vim get\_template.py
2. #!/usr/bin/env python3
3. import requests
4. import json    # python中的dict类型要转换为json格式的数据需要用到json库
5. #要访问的网址
6. url = 'http://192.168.4.2/zabbix/api\_jsonrpc.php'
7. #请求头部信息
8. headers = {'Content-Type': 'application/json-rpc'}
9. # data是从官方文档处获得的
10. data = {
11. # API使用的JSON-RPC协议的版本; Zabbix API实现JSON-RPC版本2.0
12. "jsonrpc": "2.0",
13. "method": "template.get",    #调用的API方法
14. # params将被传递给API方法的参数
15. "params": {
16. "output": "extend",
17. },
18. "auth": "0424bd59b807674191e7d77572075f33",    #之前获取到的令牌
19. "id": 1    #请求的任意标识符
20. }
21. #使用requests发送请求，访问指定网站，向url发送data请求，r收到的返回响应为json格式
22. #将data转成json格式
23. r = requests.post(url, headers=headers, data=json.dumps(data))
24. #将json格式解码
25. tinfo = r.json()
26. #print(tinfo)        #打印模板信息
27. for item in tinfo['result']:
28. print(item['templateid'], item['host'])        #打印模板名及id

3)测试脚本执行，获取了模板名称及id，如图-12所示：



图-18

步骤三：编写脚本，创建主机

1)从zabbix官方文档中使用host.create方法获取创建主机请求，如图-13所示：

官方文档地址如下：

https://www.zabbix.com/documentation/3.4/zh/manual/api/reference/host/create



图-19

2)编写get\_host.py文件

1. [root@localhost day12]# vim get\_host.py
2. #!/usr/bin/env python3
3. import requests
4. import json    # python中的dict类型要转换为json格式的数据需要用到json库
5. #要访问的网址
6. url = 'http://192.168.4.2/zabbix/api\_jsonrpc.php'
7. #请求头部信息
8. headers = {'Content-Type': 'application/json-rpc'}
9. # data是从官方文档处获得的
10. data = {
11. # API使用的JSON-RPC协议的版本; Zabbix API实现JSON-RPC版本2.0
12. "jsonrpc": "2.0",
13. "method": "user.login",    #调用的API方法
14. # params将被传递给API方法的参数
15. "params": {
16. "host": "mylinux",        #要创建的主机的名称
17. "interfaces": [
18. {
19. "type": 1, # 1 agent; 2 SNMP; 3 IPMI; 4 JMX
20. "main": 1, # 该接口是否在主机上用作默认接口。1 默认
21. "useip": 1, # 是否应通过IP进行连接
22. "ip": "192.168.4.3",
23. "dns": "",
24. "port": "10050"        #ip已装zabbix\_agent，端口号为10050
25. }
26. ],
27. "groups": [
28. {
29. "groupid": "2"        #之前检索到的主机组id
30. }
31. ],
32. "templates": [
33. {
34. "templateid": "10001"        #之前检索到的模板id
35. }
36. ],
37. "inventory\_mode": 0,        #资产信息，0为停用
38. },
39. "auth": "0424bd59b807674191e7d77572075f33",        #之前获取到的令牌
40. "id": 1    #请求的任意标识符
41. }
42. #使用requests发送请求，访问指定网站，向url发送data请求，r收到的返回响应为json格式
43. #将data转成json格式
44. r = requests.post(url, headers=headers, data=json.dumps(data))

3)测试脚本执行，运行脚本成功后，主机创建成功，如图-20所示：



图-20