1. 青龙已带python3（忽略）
2. 青龙安装相关依赖pip3 install beautifulsoup4 requests

新建定时任务

命令pip3 install beautifulsoup4 requests

其他任意

1. 运行定时任务查看日志pip3

成功即可，失败则先运行pip3 install requests方法如2

1. 新建脚本main.py

输入以下内容：（https://github.com/gold-huiyun/EUserv\_extend）

# -\*- coding: utf8 -\*-

import re

import json

import time

import requests

from bs4 import BeautifulSoup

# 强烈建议部署在非大陆区域，例如HK、SG等

# 常量命名使用全部大写的方式，可以使用下划线。

USERNAME = '6666'  # 这里填用户名，邮箱也可

PASSWORD = '8888'  # 这里填密码

# Server酱 http://sc.ftqq.com/?c=code

SCKEY = 'SCT57393TUDObY86gBrcBkQ7WSmW3AWmE'  # 这里填Server酱的key，无需推送可不填 示例: SCU646xxxxxxxxdacd6a5dc3f6

# 酷推 https://cp.xuthus.cc

COOL\_PUSH\_SKEY = ''

# 通知类型 CoolPush\_MODE的可选项有（默认send）：send[QQ私聊]、group[QQ群聊]、wx[个微]、ww[企微]

COOL\_PUSH\_MODE = 'send'

# PushPlus https://pushplus.hxtrip.com/message

PUSH\_PLUS\_TOKEN = ''

# Telegram Bot Push https://core.telegram.org/bots/api#authorizing-your-bot

TG\_BOT\_TOKEN = ''  # 通过 @BotFather 申请获得，示例：1077xxx4424:AAFjv0FcqxxxxxxgEMGfi22B4yh15R5uw

TG\_USER\_ID = ''  # 用户、群组或频道 ID，示例：129xxx206

TG\_API\_HOST = 'api.telegram.org'  # 自建 API 反代地址，供网络环境无法访问时使用，网络正常则保持默认

# wecomchan https://github.com/easychen/wecomchan

WECOMCHAN\_DOMAIN = ''  # http(s)://example.com/

WECOMCHAN\_SEND\_KEY = ''

WECOMCHAN\_TO\_USER = '@all'  # 默认全部推送, 对个别人推送可用 User1|User2

# 变量命名使用全部小写的方式，可以使用下划线。

desp = ''  # 不用动

# 函数命名使用全部小写的方式，可以使用下划线。

def print\_(info):

    print(info)

    global desp

    desp = desp + info + '\n\n'

def login(username, password) -> (str, requests.session):

    headers = {

        "user-agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) "

                      "Chrome/83.0.4103.116 Safari/537.36",

        "origin": "https://www.euserv.com",

    }

    url = "https://support.euserv.com/index.iphp"

    session = requests.Session()

    sess = session.get(url, headers=headers)

    sess\_id = re.findall("PHPSESSID=(\\w{10,100});", str(sess.headers))[0]

    # 访问png

    png\_url = "https://support.euserv.com/pic/logo\_small.png"

    session.get(png\_url, headers=headers)

    login\_data = {

        "email": username,

        "password": password,

        "form\_selected\_language": "en",

        "Submit": "Login",

        "subaction": "login",

        "sess\_id": sess\_id

    }

    f = session.post(url, headers=headers, data=login\_data)

    f.raise\_for\_status()

    if f.text.find('Hello') == -1:

        return '-1', session

    return sess\_id, session

def get\_servers(sess\_id: str, session: requests.session) -> {}:

    d = {}

    url = "https://support.euserv.com/index.iphp?sess\_id=" + sess\_id

    headers = {

        "user-agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) "

                      "Chrome/83.0.4103.116 Safari/537.36",

        "origin": "https://www.euserv.com"

    }

    f = session.get(url=url, headers=headers)

    f.raise\_for\_status()

    soup = BeautifulSoup(f.text, 'html.parser')

    for tr in soup.select('#kc2\_order\_customer\_orders\_tab\_content\_1 .kc2\_order\_table.kc2\_content\_table tr'):

        server\_id = tr.select('.td-z1-sp1-kc')

        if not len(server\_id) == 1:

            continue

        flag = True if tr.select('.td-z1-sp2-kc .kc2\_order\_action\_container')[

                           0].get\_text().find('Contract extension possible from') == -1 else False

        d[server\_id[0].get\_text()] = flag

    return d

def renew(sess\_id: str, session: requests.session, password: str, order\_id: str) -> bool:

    url = "https://support.euserv.com/index.iphp"

    headers = {

        "user-agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) "

                      "Chrome/83.0.4103.116 Safari/537.36",

        "Host": "support.euserv.com",

        "origin": "https://support.euserv.com",

        "Referer": "https://support.euserv.com/index.iphp"

    }

    data = {

        "Submit": "Extend contract",

        "sess\_id": sess\_id,

        "ord\_no": order\_id,

        "subaction": "choose\_order",

        "choose\_order\_subaction": "show\_contract\_details"

    }

    session.post(url, headers=headers, data=data)

    data = {

        "sess\_id": sess\_id,

        "subaction": "kc2\_security\_password\_get\_token",

        "prefix": "kc2\_customer\_contract\_details\_extend\_contract\_",

        "password": password

    }

    f = session.post(url, headers=headers, data=data)

    f.raise\_for\_status()

    if not json.loads(f.text)["rs"] == "success":

        return False

    token = json.loads(f.text)["token"]["value"]

    data = {

        "sess\_id": sess\_id,

        "ord\_id": order\_id,

        "subaction": "kc2\_customer\_contract\_details\_extend\_contract\_term",

        "token": token

    }

    session.post(url, headers=headers, data=data)

    time.sleep(5)

    return True

def check(sess\_id: str, session: requests.session):

    print("Checking.......")

    d = get\_servers(sess\_id, session)

    flag = True

    for key, val in d.items():

        if val:

            flag = False

            print\_("ServerID: %s Renew Failed!" % key)

    if flag:

        print\_("ALL Work Done! Enjoy")

# Server酱 http://sc.ftqq.com/?c=code

def server\_chan():

    data = (

        ('text', 'EUserv续费日志'),

        ('desp', desp)

    )

    response = requests.post('https://sc.ftqq.com/' + SCKEY + '.send', data=data)

    if response.status\_code != 200:

        print('Server酱 推送失败')

    else:

        print('Server酱 推送成功')

# 酷推 https://cp.xuthus.cc/

def coolpush():

    c = 'EUserv续费日志\n\n' + desp

    data = json.dumps({'c': c})

    url = 'https://push.xuthus.cc/' + COOL\_PUSH\_MODE + '/' + COOL\_PUSH\_SKEY

    response = requests.post(url, data=data)

    if response.status\_code != 200:

        print('酷推 推送失败')

    else:

        print('酷推 推送成功')

# PushPlus https://pushplus.hxtrip.com/message

def push\_plus():

    data = (

        ('token', PUSH\_PLUS\_TOKEN),

        ('title', 'EUserv续费日志'),

        ('content', desp)

    )

    url = 'https://pushplus.hxtrip.com/send'

    response = requests.post(url, data=data)

    if response.status\_code != 200:

        print('PushPlus 推送失败')

    else:

        print('PushPlus 推送成功')

# Telegram Bot Push https://core.telegram.org/bots/api#authorizing-your-bot

def telegram():

    data = (

        ('chat\_id', TG\_USER\_ID),

        ('text', 'EUserv续费日志\n\n' + desp)

    )

    response = requests.post('https://' + TG\_API\_HOST + '/bot' + TG\_BOT\_TOKEN + '/sendMessage', data=data)

    if response.status\_code != 200:

        print('Telegram Bot 推送失败')

    else:

        print('Telegram Bot 推送成功')

# wecomchan https://github.com/easychen/wecomchan

def wecomchan():

    response = requests.get(WECOMCHAN\_DOMAIN + 'wecomchan?sendkey=' + WECOMCHAN\_SEND\_KEY + '&msg\_type=text' + '&to\_user=' +

                            WECOMCHAN\_TO\_USER + '&msg=' + 'EUserv续费日志\n\n' + desp)

    if response.status\_code != 200:

        print('wecomchan 推送失败')

    else:

        print('wecomchan 推送成功')

def main\_handler(event, context):

    if not USERNAME or not PASSWORD:

        print\_("你没有添加任何账户")

        exit(1)

    user\_list = USERNAME.strip().split()

    passwd\_list = PASSWORD.strip().split()

    if len(user\_list) != len(passwd\_list):

        print\_("The number of usernames and passwords do not match!")

        exit(1)

    for i in range(len(user\_list)):

        print('\*' \* 30)

        print\_("正在续费第 %d 个账号" % (i + 1))

        sessid, s = login(user\_list[i], passwd\_list[i])

        if sessid == '-1':

            print\_("第 %d 个账号登陆失败，请检查登录信息" % (i + 1))

            continue

        servers = get\_servers(sessid, s)

        print\_("检测到第 {} 个账号有 {} 台VPS，正在尝试续期".format(i + 1, len(servers)))

        for k, v in servers.items():

            if v:

                if not renew(sessid, s, passwd\_list[i], k):

                    print\_("ServerID: %s Renew Error!" % k)

                else:

                    print\_("ServerID: %s has been successfully renewed!" % k)

            else:

                print\_("ServerID: %s does not need to be renewed" % k)

        time.sleep(15)

        check(sessid, s)

        time.sleep(5)

    # 五个通知渠道至少选取一个

    SCKEY and server\_chan()

    COOL\_PUSH\_MODE and COOL\_PUSH\_SKEY and coolpush()

    PUSH\_PLUS\_TOKEN and push\_plus()

    TG\_BOT\_TOKEN and TG\_USER\_ID and TG\_API\_HOST and telegram()

    WECOMCHAN\_DOMAIN and WECOMCHAN\_SEND\_KEY and WECOMCHAN\_TO\_USER and wecomchan()

    print('\*' \* 30)

if \_\_name\_\_ == '\_\_main\_\_':  # 方便我本地调试

    main\_handler(None, None)

1. 新建定时任务

命令task main.py

其他按需填写

1. 填写main.py内的参数
2. 完成