Jira插件自动申请

脚本：

[root@vm-10-183-138-248-centos jira-confluenceLicense]# cat license\_opra.sh

#申请license执行脚本，调用py申请，放在Jenkins master工作空间下

ssh -n saicapp@10.183.138.244 "mysql -uself\_check\_svc -pScm@@123 self\_check\_svc -e \"SELECT \* from jira\_confluence\_license where expiryDate<=(select DATE\_SUB(curdate(),INTERVAL -1 DAY))\"" > licenseList.txt

sed -i '1d' licenseList.txt

sed -i 's/\t/,/g' licenseList.txt

scp -rp licenseList.txt saicapp@172.16.8.252:/opt/wyx/jiraLicense

ssh saicapp@172.16.8.252 "cd /opt/wyx/jiraLicense;sh process.sh init"

if [ $? == 1 ];then

echo "没有需要申请的插件" > email.html

exit 0

else

ssh saicapp@172.16.8.252 "cat /opt/wyx/jiraLicense/updateSql.sql" > updateSql.sql

scp -rp updateSql.sql saicapp@10.183.138.244:/tmp/updateSql.sql

ssh -n saicapp@10.183.138.244 "mysql -uself\_check\_svc -pScm@@123 self\_check\_svc < /tmp/updateSql.sql"

ssh saicapp@172.16.8.252 "cat /opt/wyx/jiraLicense/email.html" > email.html

fi

数据库表结构



[root@scm-node01 jiraLicense]# cat process.sh

#!/bin/bash

#set -x

opra=$1

type=$2

function get\_info(){

# tomorrow=`date -d tomorrow +%Y-%m-%d`

# echo $tomorrow

# cat $type.txt|grep $tomorrow > licenseList.txt

ssh -n saicapp@10.183.138.244 "mysql -uself\_check\_svc -pScm@@123 self\_check\_svc -e \"SELECT \* from jira\_confluence\_license where expiryDate=(select DATE\_SUB(curdate(),INTERVAL -1 DAY))\"" > licenseList.txt

sed -i '1d' licenseList.txt

sed -i 's/\t/,/g' licenseList.txt

}

function confluence(){

cat licenseList.txt |grep "^confluence"

if [ $?==0 ];then

add=`cat licenseList.txt |grep "^confluence"|awk -F ',' '{print $4}'`

echo "confluence 需要额外申请license！！地址为："$add >> email.html

fi

if [ -s licenseList.txt ];then

echo "confluence 没有需要申请的插件" >> email.html

else

echo "confluence 插件" >> email.html

cat tem >> email.html

fi

}

function sed\_template(){

filename="email.html"

while read line

do

line=`echo $line`

plugName=`echo $line|awk -F ',' '{print $1}'`

license=`echo $line|awk -F ',' '{print $2}'`

type=`echo $line|awk -F ',' '{print $3}'`

plugInfo="<td>"$plugName"<\/td>"

typeInfo="<td>"$type"<\/td>"

licenseInfo="<td style=\"word-wrap:break-word;word-break:break-all;\" width=\"400px\">"$license"</td>"

sed -i '/<\/table>/i\<tr>' $filename

sed -i '/<\/table/i\'"$plugInfo" $filename

sed -i '/<\/table/i\'"$typeInfo" $filename

sed -i '/<\/table>/i\'"$licenseInfo" $filename

sed -i '/<\/table>/i\<\/tr>' $filename

done < new\_license.txt

}

function end(){

echo "</body>" >> email.html

echo "</html>" >> email.html

expireDate=`date +"%Y-%m-%d" -d "+ 29days"`

tomorrow=`date -d tomorrow +%Y-%m-%d`

sed -i 's/'$tomorrow'/'$expireDate'/g' jira.txt

sed -i 's/'$tomorrow'/'$expireDate'/g' confluence.txt

#ssh -n saicapp@10.183.138.244 "mysql -uself\_check\_svc -pScm@@123 self\_check\_svc -e \"UPDATE jira\_confluence\_license set expiryDate=#(select DATE\_SUB(curdate(),INTERVAL -29 DAY))

# WHERE expiryDate=(select DATE\_SUB(curdate(),INTERVAL -1 DAY))\""

}

if [ X$type == X'confluence' ];then

confluence

fi

if [ X$opra == X'get\_info' ];then

get\_info

fi

if [ X$opra == X'sed\_template' ];then

sed\_template

fi

if [ X$opra == X'end' ];then

end

fi

if [ X$opra == X'-h' ];then

echo "get\_info 获取需要申请的jira或者confluence插件列表"

echo "confluence 参数2为confluence时执行，对于confluence这个插件，单独列出来"

echo "sed\_template 许可证获取到以后，插入邮件"

echo "end 邮件结尾"

fi

if [ X$opra == X'init' ];then

> new\_license.txt

> updateSql.sql

> error.log

> email.html

#在10.24操作

#get\_info

#confluence

if [ ! -s licenseList.txt ];then

echo "没有需要申请的插件"

rm -rf email.html

exit 1

else

python3 jiraLicense.py

cp email.html.template email.html

sed\_template

add=`cat licenseList.txt |grep "^confluence"|awk -F ',' '{print $4}'`

if [ ! -z "$add" ];then

echo "confluence 需要额外申请license！！地址为："$add >> email.html

fi

end

fi

#scp -rp updateSql.sql saicapp@10.183.138.244:/tmp/updateSql.sql

#ssh -n saicapp@10.183.138.244 "mysql -uself\_check\_svc -pScm@@123 self\_check\_svc < /tmp/updateSql.sql"

fi

[root@scm-node01 jiraLicense]# cat jiraLicense.py

# -\*- coding: utf-8 -\*-

import time

import requests

import re

from urllib import parse

import json

import datetime

from requests.auth import HTTPBasicAuth

import os

from retrying import retry

def log\_output(massage):

log\_time = datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S') + "\t"

print(log\_time, massage)

# 判断前者时间是否大于后者时间

def compareDate(now\_date, new\_date):

if datetime.datetime.strptime(now\_date, '%Y-%m-%d') <= datetime.datetime.strptime(new\_date, '%Y-%m-%d'):

return True

else:

return False

class ApplyLicense:

def \_\_init\_\_(self):

self.header = {

'user-agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36'}

self.cookie = ''

self.cookie2 = ''

with open('cookie', 'r', encoding='utf-8') as f:

self.cookie1 = f.read().replace('\n', '')

with open('cookie2', 'r', encoding='utf-8') as f:

self.cookie2 = f.read().replace('\n', '')

self.cookies1 = {'cookie': self.cookie1}

self.cookies2 = {'cookie': self.cookie2}

self.cookies = self.cookies1

self.envcard\_name = ''

self.envcard\_key = ''

# get\_atlassian\_sum=0

# 访问申请的页面

@retry(stop\_max\_attempt\_number=5, wait\_random\_min=1000, wait\_random\_max=5000)

def atlassian(self, url, license\_type):

header = {

'user-agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36'}

log\_output(self.cookies)

req = requests.get(url, cookies=self.cookies,

headers=header, timeout=3)

content = req.content.decode()

print(url)

print(self.cookies)

data = {

'organisation\_name': 'jira',

'key': 'release-management',

'addOnName': 'Release Management for Jira',

'callback': '',

'licensefieldname': '',

'referrer': 'pac',

'binaryURL': 'binaryURL',

'csrfToken': 'csrfToken',

'marketplaceTermsConfirm': 'true',

'marketplaceTerms': 'true',

'usersubmit': 'true',

'submit': ''

}

for key in data:

charMatch = key + "\"\svalue=\"([\s\S]\*?)\""

# charMatch = key + "\"\svalue=\"[0-9A-Za-z-. //:]\*\""

# print(charMatch)

open('sss.html', 'w', encoding='utf-8').write(content)

try:

value = re.findall(charMatch, content)

# print(value)

except IndexError:

log\_output("没有获取到" + str(key) + "对应的值 ： ")

else:

data[key] = value

# print("新值为 "+key +":" + data[key])

data['marketplaceTermsConfirm'] = 'true'

log\_output(data)

self.envcard\_name = data['addOnName']

self.envcard\_key = data['key']

cookies = requests.utils.dict\_from\_cookiejar(req.cookies)

print(cookies)

log\_output('获取申请license相关信息，返回code '+str(req.status\_code))

if ' There appears to be an error with the page that' in content:

log\_output(content)

log\_output(" ERROR：请求license地址失败，确定账号是否还有次数" + url)

with open('error.log', 'a+', encoding='utf-8') as f:

f.writelines(" ERROR：请求license地址失败，确定账号是否还有次数" + url + '\n')

if self.cookies != self.cookies2:

self.cookies = self.cookies2

log\_output(" ERROR:切换cookie，尝试重新获取！" +

self.cookies['cookie'])

self.atlassian(url, license\_type)

else:

log\_output(" ERROR: 两个账号都没有次数了，需要手动申请")

with open('error.log', 'a+', encoding='utf-8') as f:

f.writelines(

" ERROR：请求license地址失败，确定账号是否还有次数" + url + '\n')

return None

else:

licence, expireDate = self.licence(

cookies, data, url, license\_type)

print(licence, expireDate)

self.cookies = self.cookies1

return licence, expireDate

# 调用接口申请license

@retry(stop\_max\_attempt\_number=5, wait\_random\_min=1000, wait\_random\_max=5000)

def licence(self, cookies, data, url, license\_type):

# csrfToken = atlassian()

# binaryURLencode = url.split('binaryURL=')[1]

# binaryURL = parse.unquote(binaryURLencode).encode('gbk', 'replace').decode('utf-8', 'replace')

# print(binaryURL)

log\_output(data)

req\_url = "https://my.atlassian.com/addon/try"

header = {

'user-agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36',

'origin': 'https://my.atlassian.com',

'referer': url,

}

req = requests.post(req\_url, data=data,

cookies=cookies, headers=header)

print(req.status\_code)

print(req\_url, data, cookies, header)

print("cookie")

print(cookies)

# print(req.text)

try:

licence = re.findall(

'<textarea class="textarea" name="">[\w\W]\*</textarea>', req.text)[0]

except IndexError:

print("提交License申请的响应:" + '\n' + req.text)

if 'Log in to Jira, Confluence, and all other Atlass' in req.text:

print('cookies失效，请确认；如果没问题，请重试，可能是网络原因导致的！')

exit(1)

else:

print('ERROR:申请license失败，请查找问题，重试')

licence = licence.split('>')[1].split('</')[0].replace('\n', '')

expireDate = self.update\_license(

licence, data['key'][0], license\_type)

return licence, expireDate

@retry(stop\_max\_attempt\_number=5, wait\_random\_min=1000, wait\_random\_max=5000)

def update\_license(self, license, pluginkey, license\_type):

auth = HTTPBasicAuth("cctu", "1qaz2wsx")

if license\_type == 'jira':

log\_output('当前更新的插件类型为：' + license\_type)

updateUrl = "http://jira.saicmobility.com:8080/rest/plugins/1.0/" + \

pluginkey + "-key/license"

else:

log\_output('当前更新的插件类型为：' + license\_type)

updateUrl = "http://jira.saicmobility.com:8090/rest/plugins/1.0/" + \

pluginkey + "-key/license"

log\_output("更新插件的地址为："+updateUrl)

# req=requests.put(url=updateUrl,headers=headers,data=data,auth=auth)

# print(req.url,req.headers)

# print(req.content.decode())

payload = "{\"rawLicense\" : \"%s\"}" % (license)

log\_output(str(payload))

headers = {

'Content-Type': 'application/vnd.atl.plugins+json',

}

response = requests.request(

"PUT", updateUrl, headers=headers, data=payload, auth=auth)

log\_output("更新插件的返回值："+str(response.status\_code))

if license not in str(response.status\_code):

response = requests.request(

"PUT", updateUrl, headers=headers, data=payload, auth=auth)

if response.status\_code == 404:

with open('error.log', 'a+', encoding='utf-8') as f:

f.writelines(datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')+'\t' +

"更新License返回值: " + str(response.status\_code) + "插件已不存在，无需更新" + url)

return '4040-04-04'

else:

log\_output(payload)

log\_output(response.text.encode('utf8'))

try:

expireDate = json.loads(

response.content.decode()).get('expiryDateString')

expireDate = datetime.datetime.strptime(

expireDate, '%b %d, %Y')

log\_output("confluence")

except ValueError:

log\_output("jira")

except TypeError:

subCode = json.loads(response.content.decode()).get('subCode')

if subCode == 'upm.plugin.license.error.invalid.update':

log\_output('插件已经更新！')

# 获取插件到期时间

expireDate\_respon = requests.get(

updateUrl, headers=headers, auth=auth)

expireDate\_json = json.loads(

expireDate\_respon.content.decode())

expireDate = expireDate\_json.get('expiryDateString')

if ',' in expireDate:

expireDate = datetime.datetime.strptime(

expireDate, '%b %d, %Y')

# print(expireDate\_json)

except json.decoder.JSONDecodeError:

print('更新失败')

return None

expireDate = str(expireDate).split(' ')[0]

log\_output("插件到期时间是：" + expireDate)

if license\_type == 'confluence-other':

updateUrl = "http://operating.saicm.local:8090/rest/plugins/1.0/" + \

pluginkey + "-key/license"

response = requests.request(

"PUT", updateUrl, headers=headers, data=payload, auth=auth)

try:

expireDate = json.loads(

response.content.decode()).get('expiryDateString')

expireDate = datetime.datetime.strptime(

expireDate, "%b %d, %Y")

expireDate = str(expireDate).split(' ')[0]

except TypeError:

subCode = json.loads(

response.content.decode()).get('subCode')

if subCode == 'upm.plugin.license.error.invalid.update':

log\_output('插件已经更新！')

return expireDate

# 到自己的账户查看已经申请过的license

@retry(stop\_max\_attempt\_number=5, wait\_random\_min=1000, wait\_random\_max=5000)

def get\_atlassian\_index(self, evcard\_plugin\_name):

result = 'True'

license\_index\_url = 'https://my.atlassian.com/products/index'

# log\_output(self.cookies)

respon\_data = requests.get(

license\_index\_url, headers=self.header, cookies=self.cookies)

log\_output(evcard\_plugin\_name)

if self.cookies == self.cookies2:

result = 'False'

log\_output("切换账号，尝试使用cookie2查看，cookie更新为："+self.cookie2)

with open('index-cookies2.html', 'w', encoding='utf-8') as f:

f.write(respon\_data.content.decode())

elif self.cookies == self.cookies1:

log\_output("当前账号使用的是通用cookie")

with open('index-cookies1.html', 'w', encoding='utf-8') as f:

f.write(respon\_data.content.decode())

if 'Log in to Jira, Confluence, and all other Atlass' in respon\_data:

log\_output("没有访问到页面，cookie失效：", respon\_data.text)

return None

# print(respon\_data.content)

license\_sum = re.findall(

'license\_[0-9]{1,100}', respon\_data.content.decode())

license\_expireDate = re.findall(

'[0-9]{2}\s[A-Z]{1}[a-z]\*\s[0-9]{4}', respon\_data.content.decode())

license\_name = re.findall(

'<span class="desc"><strong>([\s\S]\*?)</strong>', respon\_data.content.decode())

for index in range(len(license\_sum)):

# 插件名称等于index界面的插件，并且index界面过期时间小于当前时间，则通过index界面拿到插件

# print(evcard\_plugin\_name,license\_name[index],datetime.datetime.today().date(),datetime.datetime.strptime(

# license\_expireDate[index], '%d %b %Y').date() )

if self.cookies == self.cookie2:

log\_output(evcard\_plugin\_name +

license\_name[index].split(': Trial')[0])

#

# datetime.datetime.strptime('2021-03-28','%Y-%m-%d').date()

if evcard\_plugin\_name == license\_name[index].split(': Trial')[0] and datetime.datetime.today().date() < datetime.datetime.strptime(

license\_expireDate[index], '%d %b %Y').date():

log\_output(datetime.datetime.today().date())

log\_output(license\_expireDate[index])

url = 'https://my.atlassian.com/products/licensedetail?licenseId=' + \

license\_sum[index].split('\_')[1]

response\_data = requests.get(

url, headers=self.header, cookies=self.cookies)

if self.cookies == self.cookies2:

log\_output("当前账号为cookie2")

elif self.cookies == self.cookies1:

log\_output("当前账号为cookie1")

log\_output(url+'\t')

try:

license\_content = \

re.findall('readonly>[\w\W]\*</textarea>', response\_data.content.decode())[0].split('>')[1].split(

'<')[0].replace('\n', '')

return license\_content

except IndexError:

log\_output("没有找到对应的信息，licensekey或者license：" +

response\_data.content.decode())

log\_output(self.cookies)

return None

# print(license\_content,license\_key,license\_expireDate[index], license\_name[index])

elif evcard\_plugin\_name == license\_name[index].split(': Trial')[0]:

log\_output(evcard\_plugin\_name+'\t'+datetime.date.strftime(

datetime.datetime.strptime(license\_expireDate[index], '%d %b %Y'), '%Y-%m-%d'))

if result != 'False':

self.cookies = self.cookies2

license\_content = self.get\_atlassian\_index(

evcard\_plugin\_name)

if license\_content is not None:

return license\_content

return None

# 获取atlssian网站上插件的license-key或者name

@retry(stop\_max\_attempt\_number=5, wait\_random\_min=1000, wait\_random\_max=5000)

def get\_license\_name(self, atlassian\_license\_url, param):

header = {

'user-agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36'}

with open('cookie2', 'r', encoding='utf-8') as f:

cookie2 = f.read().replace('\n', '')

cookies = {'cookie': cookie2}

# print(cookie)

req = requests.get(atlassian\_license\_url,

cookies=cookies, headers=header)

data = {

'organisation\_name': 'jira',

'key': 'release-management',

'addOnName': 'Release Management for Jira',

'callback': '',

'licensefieldname': '',

'referrer': 'pac',

'binaryURL': 'binaryURL',

'csrfToken': 'csrfToken',

'marketplaceTermsConfirm': 'true',

'marketplaceTerms': 'true',

'usersubmit': 'true',

'submit': ''

}

# csrfToken = re.findall('csrfToken"\svalue="[a-zA-Z0-9-]\*"',content)[0].split('value="')[1].split('"')[0]

for key in data:

charMatch = key + "\"\svalue=\"([\s\S]\*?)\""

try:

# print(re.findall(charMatch, req.text)[0])

value = re.findall(charMatch, req.text)[0]

# value = re.findall(charMatch, req.text)[0].split('value="')[1].split('"')[0]

except IndexError:

# print("没有获取到" + key + "对应的值 ： ")

oo = 1

else:

data[key] = value

if data['addOnName'] == 'Release Management for Jira':

with open('sss.html', 'w', encoding='utf-8') as u:

u.write(req.text)

# print(charMatch)

# elif '&amp;' in data['addOnName'] :

# data['addOnName']=data['addOnName'].replace('&amp;','&')

log\_output(str(data))

return data[param]

if \_\_name\_\_ == '\_\_main\_\_':

# atlassian("https://my.atlassian.com/addon/try/com.xiplink.jira.git.jira\_git\_plugin?referrer=pac&binaryURL=https%3A%2F%2Fmarketplace.atlassian.com%2Fdownload%2Fapps%2F4984%2Fversion%2F1004751","jira")

f = open("licenseList.txt")

for line in f:

line = line.replace('\n', '')

expireDate = line.split(',')[2]

#tomorrow = datetime.date.today() + datetime.timedelta(days=1)

# if str(tomorrow) == expireDate:

url = line.split(',')[3]

license\_type = line.split(',')[4]

name = line.split(',')[0]

log\_output('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_' + name + ' 到期时间：' +

expireDate + ' 开始申请插件' + '\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_')

log\_output('插件申请的地址为：' + url)

if license\_type == "confluence-main":

log\_output("confluence插件，需要单独申请！！！")

else:

try:

license\_apply = ApplyLicense()

# pluginName = ApplyLicense.get\_license\_name(

# license\_apply, url, 'addOnName')

pluginName = license\_apply.get\_license\_name(url, 'addOnName')

license\_key = url.split('?')[0].split('/')[-1]

# license\_content = ApplyLicense.get\_atlassian\_index(

# ApplyLicense, pluginName)

license\_content = license\_apply.get\_atlassian\_index(pluginName)

license\_apply.cookies = license\_apply.cookies1

# ApplyLicense.cookies = ApplyLicense.cookies1

log\_output(license\_content)

if license\_content is not None:

# expireDate = ApplyLicense.update\_license(ApplyLicense,

# license\_content, license\_key, license\_type)

expireDate = license\_apply.update\_license(

license\_content, license\_key, license\_type)

else:

log\_output('当前账号已申请的插件中，未找到符合条件的，重新获取name，再查找')

# pluginName = ApplyLicense.get\_license\_name(

# ApplyLicense, url, 'addOnName')

pluginName = license\_apply.get\_license\_name(

url, 'addOnName')

# print(license\_name)

# license\_content = ApplyLicense.get\_atlassian\_index(

# ApplyLicense, pluginName)

license\_content = license\_apply.get\_atlassian\_index(

pluginName)

# ApplyLicense.cookies = ApplyLicense.cookies1

license\_apply.cookies = license\_apply.cookies1

if license\_content is not None:

# expireDate = ApplyLicense.update\_license(ApplyLicense,

# license\_content, license\_key, license\_type)

expireDate = license\_apply.update\_license(

license\_content, license\_key, license\_type)

else:

log\_output('当前账号已申请的插件中，未找到符合条件的,尝试重新申请')

# license\_content, expireDate = ApplyLicense.atlassian(

# ApplyLicense, url, license\_type)

license\_content, expireDate = license\_apply.atlassian(

url, license\_type)

if expireDate is None:

log\_output('插件申请失败！！！，' + url)

with open('error.log', 'a+', encoding='utf-8') as fs:

fs.write(datetime.datetime.now().strftime(

'%Y-%m-%d %H:%M:%S') + "\t" + '插件申请失败！！！，' + url + '\t' + line + '\n')

with open('new\_license.txt', 'a+', encoding='utf-8') as f:

f.write(pluginName + ',' +

license\_content + ',' + license\_type + '\n')

except IndexError:

with open('error.log', 'a+', encoding='utf-8') as fs:

fs.write(datetime.datetime.now().strftime(

'%Y-%m-%d %H:%M:%S') + "\t" + "没有插件名称，无法查询" + '\t' + line + '\n')

if expireDate is None:

print("需要排查问题以后重新申请")

else:

with open('updateSql.sql', 'a+', encoding='utf-8') as f2:

sql = "UPDATE jira\_confluence\_license set expiryDate='%s' WHERE pluginName='%s'" % (

expireDate, name)

print(sql)

f2.write(sql + ';' + '\n')