自动化测试使用手册

前提：集群中有两个linux节点。

1.拷贝自动化环境：

将已经打包好的自动化环境：myenv，放到集群master节点的/tmp下，拷贝的时候请使用-rp选项，保持自动化环境的目录权限(autotest4/spooler文件夹内的所有文件权限都必须为777，其它文件夹的目录为755)。

注意：

1.每次使用自动之前请务必保证autotest4/文件的属性正确，

1. source自动化环境source /../autotest4/autotest\_prepare.sh

3.source景行管理与调度软件的环境，保证集群节点状态均ok。

4.请勿移动自动化包，如若移动，请重新取包，因为自动化包中有些路径被source之后是无法恢复的。（eg：esub文件中的路径等）

5.必须在master节点上执行case（方便重启集群和reconfig），最好在root下执行所有case，如果不想在root下执行所有case，请将集群管理员用户和user1-user8之间配置免登陆，并且保证用户也可以su自己，eg，用jhadmin执行，su jhadmin必须可以执行成功。

2.执行自动化测试case：

环境安装好以后，使用一下步骤启动自动化测试：

步骤1：在master节点上，（1）.执行： cd /tmp/myenv （2）.执行：. bin/activate

步骤2：将/tmp/myenv/autotest4/autotest\_prepare.sh中的路径和节点名改成当前的路径和节点名。

其中：export AUTOTEST\_TOP="/apps/autotest4/"（自动化测试库所在目录）

export CLUSTER\_ADMIN="jhadmin" （集群的管理员用户）

MASTER="rhelA1" （集群master节点的节点名称）

LINUXA="rhelB" （集群slaver节点的节点名称）

USER1="user1"

USER2="user2"

USER3="user3"

USER4="user4"

USER5="user5"

USER6="user6"

USER7="user7"

USER8="user8"

（user1-user3是自动化case中需要使用，user1-user8是fairshare调度策略中需要使用的用户，如果没有这些用户，部分case将会无法通过。所有集群中出管理员用户外，需要存在8个普通用户）

步骤3：source /tmp/myenv/autotest4/autotest\_prepare.sh

步骤4：执行pybot /tmp/myenv/autotest4/testcase/old/suit-old.txt(要写全路径)

此外如果需要挑选执行一些case来执行，可以参考如下命令，也可以网上查询pybot的选项：

pybot /tmp/myenv/autotest4/testcase/old/suit-old.txt（执行suit-old.txt中的所有case）

pybot –include tag0 /tmp/myenv/autotest4/testcase/ (执行testcase中标签为tag0的suite的所有case)

pybot –include tag1–test case\* /tmp/myenv/autotest4/testcase/old/suit-old.txt（执行suit-old.txt中所有标签为tag1的case）

如果用户想自己安装自动化环境并执行自动化case，则执行以下步骤即可（不含ride.py界面）：

前提：当前节点能访问外网，Python版本在2.6.6-2.7.10之间。

安装步骤：

1.安装pip

(下载和安装步骤见：https://pip.pypa.io/en/stable/installing/，或者使用给出的get-pip.py文件)，然后在节点上执行：

步骤：执行：python get-pip.py

(若使用yum源安装pip，执行以下步骤:1.sudo yum upgrade python-setuptools 2.sudo yum install python-pip python-wheel)

2.安装robotframework

步骤：pip install robotframework（使用2.9-3.0.2之间的版本，为了避免版本不兼容问题）

3.安装robotframework-ride

步骤：pip install robotframework-ride（注：使用1.3-1.5.2之间的版本号，为了避免版本不兼容问题）

执行pybot --version 即可查看版本号

1. pip install psutil

执行自动化测试case：

环境安装好以后，使用一下步骤启动自动化测试：

步骤1：在master节点上，将autotest4拷贝到master节点上，拷贝的时候务必保持目录权限(autotest4/spooler文件夹内的所有文件权限都必须为777，其它文件夹的目录为755)。Eg:拷贝到/tmp下。

步骤2：将/tmp/ autotest4/autotest\_prepare.sh中的路径和节点名改成当前的路径和节点名。

其中：export AUTOTEST\_TOP="/tmp/autotest4/"（自动化测试库所在目录）

export CLUSTER\_ADMIN="jhadmin" （集群的管理员用户）

MASTER="rhelA1" （集群master节点的节点名称）

LINUXA="rhelB" （集群slaver节点的节点名称）

USER1="user1"

USER2="user2"

USER3="user3"

USER4="user4"

USER5="user5"

USER6="user6"

USER7="user7"

USER8="user8"

（user1-user3是自动化case中需要使用，user1-user8是fairshare调度策略中需要使用的用户，如果没有这些用户，部分case将会无法通过。所有集群中出管理员用户外，需要存在8个普通用户）

步骤3：source /tmp/ autotest4/autotest\_prepare.sh

步骤4：执行pybot /tmp/ autotest4/testcase/old/suit-old.txt(要写全路径)

如果用户想使用界面，按照以下步骤可以安装ride。

Ride安装步骤：

1.安装centos虚拟机：

2.安装好后：

挂载光盘：

2.1.mkdir centos7

mount  192.168.0.31:/appstore/ /appstore/

mount -oloop appstore/iso/CentOS/CentOS-7.2.iso  /centos7/

mount: /dev/loop0 is write-protected, mounting read-only

至此yum源已经配置好

3.安装pip

wget https://bootstrap.pypa.io/get-pip.py

然后：python get-pip.py

4.安装robotframework：

pip install robotframework

5.安装ride:

pip install robotframework-ride

执行pybot --version 即可查看版本号

pip install psutil (如果报错： #include <Python.h>

^

compilation terminated.

error: command 'gcc' failed with exit status 1

，安装yum install python-devel.x86\_64 -y)

1. 安装wxPython:

(1)使用下载地址是：<https://sourceforge.net/projects/wxpython/files/wxPython/2.8.12.1/>

下载的包是：wxPython-src-2.8.12.1.tar.bz2

解压：bzip2 -d wxPython-src-2.8.12.1.tar.bz2

解压后切换到wxWidgets-2.8.12目录下，执行./configure

 报错没有gtk2：

 yum search gtk

 gtk2-devel.x86\_64 : Development files for GTK+

yum install gtk2-devel.x86\_64

安装后，重新reconfig。（reconfig成功后会生成make文件的）

 (如果没有g++就安装g++。

yum whatprovides \*/g++

yum install gcc-c++-4.8.5-16.el7.x86\_64

)

重新make

（如果出现./include/wx/defs.h:212:26: error: redeclaration of C++ built-in type ‘bool’ [-fpermissive]

typedef unsigned int bool;

^

以上错误。

则，重新打开一个终端重新config、重新make）

最后，

Make install

8.编写文件：

在/etc/yum.repos.d下写文件：epel.repo，文件内容见下面的cat。

[root@localhost ~]# cd /etc/yum.repos.d/

[root@localhost yum.repos.d]# ls

centos7.repo.bak  CentOS-CR.repo         CentOS-fasttrack.repo  CentOS-Sources.repo  epel.repo

CentOS-Base.repo  CentOS-Debuginfo.repo  CentOS-Media.repo      CentOS-Vault.repo

[root@localhost yum.repos.d]# cat epel.repo

[epel]

name=Extra Packages for Enterprise Linux 7 - $basearch

#baseurl=http://download.fedoraproject.org/pub/epel/7/$basearch

mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-7&arch=$basearch

failovermethod=priority

enabled=1

gpgcheck=0

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7

[epel-debuginfo]

name=Extra Packages for Enterprise Linux 7 - $basearch - Debug

#baseurl=http://download.fedoraproject.org/pub/epel/7/$basearch/debug

mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-debug-7&arch=$basearch

failovermethod=priority

enabled=0

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7

gpgcheck=1

[epel-source]

name=Extra Packages for Enterprise Linux 7 - $basearch - Source

#baseurl=http://download.fedoraproject.org/pub/epel/7/SRPMS

mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-source-7&arch=$

9.yum install wxPython -y

10.执行ride.py即可使用界面。

Python2.7.5安装自动化包中遇到的问题：

1. 需要用python2.7的get-pip。
2. 当pip install安装不上psutils时，按照如下步骤安装：

下载psutils的包，psutil-2.0.0.tar.gz，加压并安装：

tar -zxvf psutil-2.0.0.tar.gz.

Cd psutil-2.0.0

python setup.py install

如果报错：

[root@rhelA1 psutil-2.0.0]# python setup.py install

running install

running bdist\_egg

running egg\_info

writing psutil.egg-info/PKG-INFO

writing top-level names to psutil.egg-info/top\_level.txt

writing dependency\_links to psutil.egg-info/dependency\_links.txt

reading manifest file 'psutil.egg-info/SOURCES.txt'

reading manifest template 'MANIFEST.in'

warning: no previously-included files matching '\*' found under directory 'docs/\_build'

writing manifest file 'psutil.egg-info/SOURCES.txt'

installing library code to build/bdist.linux-x86\_64/egg

running install\_lib

running build\_py

running build\_ext

building '\_psutil\_linux' extension

gcc -pthread -fno-strict-aliasing -O2 -g -pipe -Wall -Wp,-D\_FORTIFY\_SOURCE=2 -fexceptions -fcord-gcc-switches -m64 -mtune=generic -D\_GNU\_SOURCE -fPIC -fwrapv -DNDEBUG -O2 -g -pipe -Waltor-strong --param=ssp-buffer-size=4 -grecord-gcc-switches -m64 -mtune=generic -D\_GNU\_SOURCEil/\_psutil\_linux.c -o build/temp.linux-x86\_64-2.7/psutil/\_psutil\_linux.o

psutil/\_psutil\_linux.c:12:20: fatal error: Python.h: No such file or directory

#include <Python.h>

^

compilation terminated.

error: command 'gcc' failed with exit status 1

需要配置yum源，然后安装yum install python-devel -y。

再次执行python setup.py install安装psutils即可。