一、ansible原理：

控制端主机自带很多模块（模块就是脚本）；

ansible通过ssh远程被管理主机，将控制端的模块（脚本）或命令传输到被管理主机；

在被管理端主机执行模块（脚本）或命令，执行不同的模块或命令可以实现不同的功能；

最后ansible退出ssh远程。

绝大多数模块（脚本）都需要参数才能执行成功！！！类似于shell脚本的位置变量！

二、准备如表-1所示的实验环境，操作系统为RHEL8，配置主机名称、IP地址、YUM源、

1）Control控制节点

1. 修改/etc/hosts，在文件中手动添加如下内容，修改该文件的目的是做域名解析。**[**root@control **~]**# vim **/**etc**/**hosts        #修改文件，手动添加如下内容（不要删除文件原来的内容）
2. **192.168.4.253**    control
3. **192.168.4.11**        node1
4. **192.168.4.12**        node2
5. **192.168.4.13**        node3
6. **192.168.4.14**        node4
7. **192.168.4.15**        node5

**2.**如何验证？

1. **[**root@control **~]**# ping node1 #可以使用ping命令依次ping所有域名

3、配置SSH密钥实现免密码登录（非常重要）

Ansible是基于SSH远程的原理实现远程控制，如果控制端主机无法免密登录被管理端主机，后续的所有试验都会失

1. **[**root@control **~]**# ssh**-**keygen **-**f **/**root**/.**ssh**/**id\_rsa **-**N '' #生成ssh密钥
2. #**-**f指定密钥文件存放在哪个目录，文件叫什么名字，**-**N ''设置密钥的密码为空
3. **[**root@control **~]**# **for** i **in** node1 node2 node3 node4 node5
4. **do**
5. ssh**-**copy**-**id $i
6. done
7. #拷贝密钥到远程主机
8. #提示：拷贝密钥到远程主机时需要输入对方电脑的账户密码才可以！！
9. #拷贝密钥到node1就需要输入node1对应账户的密码，拷贝密钥到node2就需要输入node2对应的密码

如何验证？

警告：如果有任何一台主机远程还需要密码，就不要往下继续操作，后面实验都会失败！！！

ssh node1 #使用ssh命令依次远程所有主机都可以免密码登录

2)部署Ansible软件（仅Control主机操作，软件包在ansible\_soft目录）

1. **[**root@control **~]**# tar **-**xf ansible\_soft**.**tar**.**gz
2. **[**root@control **~]**# cd ansible\_soft
3. **[**root@control ansible\_soft**]**# dnf **-**y install **\***

**步骤二：修改配置文件**

1. 主配置文件说明：
2. 主配置文件ansible.cfg（主配置文件的内容可以参考/etc/ansible/ansible.cfg）
3. ansible配置文件查找顺序
4. 首先检测ANSIBLE\_CONFIG变量定义的配置文件（默认没有这个变量）
5. 其次检查当前目录下的./ansible.cfg文件
6. 再次检查当前用户家目录下~/ansible.cfg文件
7. 最后检查/etc/ansible/ansible.cfg文件
8. 1) 修改主配置文件。
9. **[**root@control **~]**# mkdir **~/**ansible
10. **[**root@control **~]**# vim **~**/ansible/ansible**.**cfg
11. **[**defaults**]**
12. inventory **=** **~**/ansible/inventory
13. #主机清单配置文件（inventory可以是任意文件名），英语词汇：inventory（清单、财产清单）
14. #forks **=** **5** #ssh并发数量
15. #ask\_pass **=** True #使用密钥还是密码远程，True代表使用密码
16. #host\_key\_checking **=** False #是否校验密钥（第一次ssh时是否提示yes**/**no）注意#可以不用写

2) 修改主机清单文件（清单文件名必须与主配置文件inventory定义的一致）。

1. **[**root@control **~]**# vim **~**/ansible/inventory
2. **[**test**]**                    #定义主机组（组名称任意）
3. node1                    #定义组中的具体主机，组中包括一台主机node1
4. **[**proxy**]** #定义主机组（组名称任意），英语词汇：proxy（代理人，委托人）
5. node2 #proxy组中包括一台主机node2
6. **[**webserver**]**
7. node**[3:4]** #这里的node**[3:4]**等同于node3和node4
8. **[**database**]**
9. node5
10. **[**cluster**:**children**]**        #嵌套组（children为关键字），不需要也可以不创建嵌套组
11. webserver #嵌套组可以在组中包含其他组
12. database

三、ansible应用

Ansible ad-hoc是一种通过命令行批量管理的方式，命令基本格式如下：

格式：ansible 主机集合 -m 模块名 -a "参数"

1）查看主机列表

1. **[**root@control **~]**# cd **~/**ansible                            #非常重要
2. **[**root@control ansible**]**# ansible all **--**list**-**hosts #查看所有主机列表
3. # **--**list**-**hosts是ansible这个命令的固定选项，如同ls **-**a一样（**-**a是ls命令的固定选项）
4. #英语词汇：list（列表，清单）、host（主机、主办、主人）

2) 测试远程主机是否能ping通。

当需要远程多个主机或者多个组时，中间使用逗号分隔！！！

1. **[**root@control ansible**]**# ansible node1 **-**m ping     #调用ping模块
2. **[**root@control ansible**]**# ansible node1**,**webserver **-**m ping

常见报错（有问题可以参考，没问题可以忽略）：

1. node1 **|** UNREACHABLE**!** **=>** **{**
2. "changed"**:** **false,**
3. "msg"**:** "Failed to connect to the host via ssh: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password)."**,**
4. "unreachable"**:** **true**
5. **}**
6. 问题分析：
7. 英语词汇：Failed（失败），connect（连接），to（到），host（主机），via（通过）
8. permission（权限），denied（被拒绝）
9. Failed to connect to host via ssh（通过ssh远程连接到主机失败）
10. Permission denied（因为无法连接，所以报错说权限被拒绝）
11. 解决办法：手动ssh其他主机（如node1），看看是否可以实现免密码登录。
12. Ansible的原理是基于ssh远程管理，如果无法实现免密码登录，后面的实验无法成功！
13. 如何实现免密码登录，可以参考案例上面的命令，或者第一阶段知识。

3）快速入门。

模块就是脚本（多数为Python脚本），多数脚本都支持参数，默认模块为command。

1. **[**root@control ansible**]**# ansible node1 **-**m command **-**a "uptime" #查看CPU负载
2. **[**root@control ansible**]**# ansible node1 **-**m command **-**a "uname -r" #查看内核版本
3. **[**root@control ansible**]**# ansible node1 **-**a "ip a s" #查看网卡信息
4. **[**root@control ansible**]**# ansible all **-**a "date" #查看时间

通过ansible-doc获取帮助。

1. **[**root@control ansible**]**# ansible**-**doc **-**l     #列出所有模块
2. **[**root@control ansible**]**# ansible**-**doc **-**l **|** grep yum            #在所有模块中过滤关键词
3. **[**root@control ansible**]**# ansible**-**doc yum         #查看模块帮助

4）Shell模块。

command和shell模块的区别，command不支持bash的特性（bash有哪些特性可以参考Shell课程第一天的PPT），如管道和重定向等功能，但是shell模块可以支持。

不可以使用shell模块执行交互命令，如vim、top等。

1. **[**root@control ansible**]**# ansible test **-**m command **-**a "ps | wc -l" #报错
2. **[**root@control ansible**]**# ansible test **-**m command **-**a "ls &" #报错
3. **[**root@control ansible**]**# ansible test **-**m shell **-**a "ps aux | wc -l" #进程数量
4. **[**root@control ansible**]**# ansible test **-**m shell **-**a "who" #登陆信息
5. **[**root@control ansible**]**# ansible test **-**m shell **-**a "touch /tmp/txt.txt"
6. #使用shell模块创建文件会有Warning警告提示，正常！！！

5）script模块

script模块会把-a后面的脚本拷贝到被管理端主机，然后执行这个脚本。

1. **[**root@control ansible**]**# vim **~**/ansible/test**.**sh
2. #**!**/bin/bash
3. dnf **-**y install httpd
4. systemctl start httpd
5. **[**root@control ansible**]**# ansible test **-**m script **-**a "./test.sh"
6. #test是主机组的名称，**-**m调用script模块，**-**a后面的**./**test**.**sh是上面创建脚本的相对路径和文件名
7. #**./**是当前目录的意思，在当前目录下有个脚本叫test**.**sh

如何验证？

因为ansible远程的是node1，所以打开node1这台电脑，查看下是否安装了httpd软件、是否启动了服务。

1. **[**root@node1 **~]**# rpm **-**q httpd
2. **[**root@node1 **~]**# systemctl status httpd

四、很多ansible模块都具有幂等性的特征。

幂等性：任意次执行所产生的影响均与一次执行的影响相同。

**file模块**

file模块可以创建文件、目录、链接；修改权限与属性等（ansible-doc file）

1. **[**root@control ansible**]**# ansible test **-**m file **-**a "path=/tmp/file.txt state=touch" #远程test组中所有主机，新建文件，path后面指定要创建的文件或目录的名称
2. #state**=**touch是创建文件，state**=**directory是创建目录
3. ## 验证： 到node1主机，使用ls **/**tmp**/**file**.**txt看看文件是否被创建成功 ##

常见报错（有问题可以参考，没问题可以忽略）：

1. node1 **|** FAILED**!** **=>** **{**
2. … …
3. "changed"**:** **false,**
4. "msg"**:** "value of state must be one of: absent, directory, file, hard, link, touch, got: touc"
5. **}**
6. 英语词汇：value（值），must（必须），be（是），of（…的），one（一个）
7. value of state must be one of**:**【state的值必须是后面给出的其中一个值】
8. 解决办法：检查state的值是否有字母错误**,**上面报错例子中输入的是touc，不是touch。

常见错误（有问题可以参考，没问题可以忽略）：

1. node1 **|** FAILED**!** **=>** **{**
2. … …
3. "msg"**:** "Unsupported parameters for (file) module: nmae Supported parameters include: \_diff\_peek, \_original\_basename, access\_time,
4. access\_time\_format, attributes, backup, content, delimiter, directory\_mode,
5. follow, force, group, mode, modification\_time, modification\_time\_format, owner,
6. path, recurse, regexp, remote\_src, selevel, serole, setype, seuser, src, state,
7. unsafe\_writes"
8. **}**
9. 英语词汇：unsupported（不支持的），parameters（参数），supported（支持的）include**(**包括**)**
10. 问题分析：file模块不支持nmae这个参数，它支持的参数包括哪些，后面有提示**.**
11. 解决办法：检查模块的参数是否有字母错误，上面错误案例将name错写为nmae。

更多file模块的案例：

1. **[**root@control ansible**]**# ansible test **-**m file **\**
2. **-**a "path=/tmp/mydir state=directory"
3. #远程test组中所有主机，创建目录，path后面指定要创建的文件或目录的名称
4. ## 验证：到node1主机，使用ls **/**tmp**/**看看tmp目录下是否有mydir子目录
5. **[**root@control ansible**]**# ansible test **-**m file **\**
6. **-**a "path=/tmp/file.txt owner=sshd group=adm mode=0777"
7. #修改文件或目录权限，path后面指定要修改的文件名或目录名称，owner后面指定用户，group后面指定组，mode后面指定要修改的权限（**0777**中第一个**0**代表的是无特殊权限，如SUID、SGID等）
8. ## 验证：到node1主机，使用ls **-**l **/**tmp**/**file**.**txt查看文件的详细信息是否正确
9. **[**root@control ansible**]**# ansible test **-**m file **-**a "path=/tmp/mydir state=absent"
10. #state**=**absent代表删除（删除目录）
11. **[**root@control ansible**]**# ansible test **-**m file **-**a "path=/tmp/file.txt state=absent"
12. # state**=**absent代表删除（删除文件）
13. **[**root@control ansible**]**# ansible test **-**m file **\**
14. **-**a "src=/etc/hosts path=/tmp/host.txt state=link"
15. #给/etc/hosts文件创建一个链接文件/tmp/host**.**txt（src指定源文件，path是软链接文件名）
16. #相当于执行命令 ln **-**s **/**etc**/**hosts **/**tmp**/**host**.**txt
17. ## 验证：到node1主机使用ls **-**l **/**tmp**/**hosts查看文件是否为软链接

**步骤二：copy模块**

copy模块可以将文件拷贝到远程主机 (ansible-doc copy)。

1. **[**root@control ansible**]**# echo AAA **>** **~/**a3**.**txt #新建测试文件
2. **[**root@control ansible**]**# ansible test **-**m copy **-**a "src=~/a3.txt dest=/root/"
3. #把管理端本机的a3**.**txt文件，拷贝到test组中所有主机的/root/目录
4. #src代表源文件，dest代表目标文件
5. ## 验证：到node1主机使用ls **/**root**/**a3**.**txt查看是否有该文件

**步骤三：fetch模块**

fetch模块与copy类似，但是作用相反,可以将其他主机的文件拷贝到本地(ansible-doc fetch)。

1. **[**root@control ansible**]**# ansible test **-**m fetch **-**a "src=/etc/hostname dest=~/"
2. #将远程test组中所有主机的hostname文件下载到本地家目录
3. #src代表源文件，dest代表目标文件
4. **[**root@control ansible**]**# ls **~/** #使用ls查看下是否下载成功
5. #不能下载目录，如果需要下载目录，可以先打包后再下载

**步骤四：lineinfile|replace模块**

在修改单个文件的单行内容时可以使用lineinfile模块(ansible-doc lineinfile)。

1. **[**root@control ansible**]**# ansible test **-**m lineinfile **\**
2. **-**a "path=/etc/issue line='hello world'"
3. #在/etc/issue文件中添加一行内容hello world，默认添加到最后，line后面跟的是需要添加的文件内容
4. ## 验证：到node1主机执行命令cat **/**etc**/**issue查看文件内容是否正确
5. **[**root@control ansible**]**# ansible test **-**m lineinfile **\**
6. **-**a "path=/etc/issue line='hello world'"
7. #基于幂等原则，重复执行，不会创建多行内容
8. **[**root@control ansible**]**# ansible test **-**m lineinfile **\**
9. **-**a "path=/etc/issue line='insert' insertafter='Kernel'"
10. #将line后面的内容插入到/etc/issue文件中Kernel行的后面
11. #英语词汇：insert（插入），after（在…后面）
12. ## 验证：到node1主机执行命令cat **/**etc**/**issue查看文件内容是否正确

lineinfile会替换一整行，replace可以替换关键词(ansible-doc replace)。

1. **[**root@control ansible**]**# ansible test **-**m replace **\**
2. **-**a "path=/etc/issue.net regexp=Kernel replace=Ocean"
3. #将node1主机中/etc/issue**.**net文件全文所有的Kernel替换为Ocean
4. #regexp后面是需要替换的旧内容；replace后面是需要替换的新内容
5. ## 验证：到node1主机执行命令cat **/**etc**/**issue**.**net查看文件内容是否正确

## 四、4 案例4：Ansible ad-hoc应用三

**4.1 问题**

沿用练习三，继续练习Ansible ad-hoc应用案例，具体要求如下：

* 远程目标主机创建、删除系统账户；设置系统账户属性、修改账户密码
* 为目标主机创建、删除yum源配置文件；远程目标主机安装、卸载软件包
* 使用service模块管理远程主机的服务
* 创建、删除逻辑卷

**步骤一：user模块**

user模块可以实现Linux系统账户管理(ansible-doc user)。

1. **[**root@control ansible**]**# ansible test **-**m user **-**a "name=tuser1"
2. #远程test组中的所有主机并创建系统账户tuser1，默认state的值为present，代表创建用户
3. ## 验证：到node1主机执行命令id tuser1查看是否有该用户
4. **[**root@control ansible**]**# ansible test **-**m user **-**a **\**
5. "name=tuser2 uid=1010 group=adm groups=daemon,root home=/home/tuser2"
6. #创建账户并设置对应的账户属性，uid指定用户ID号，group指定用户属于哪个基本组
7. #groups指定用户属于哪些附加组，home指定用户的家目录
8. ## 验证： 到node1主机执行命令id tuser2查看是否有该用户
9. **[**root@control ansible**]**# ansible test **-**m user **\**
10. **-**a "name=tuser1 password={{'abc'| password\_hash('sha512')}}"
11. #修改账户密码，用户名是tuser1，密码是abc，密码经过sha512加密
12. **[**root@control ansible**]**# ansible test **-**m user **\**
13. **-**a "name=tuser1 state=absent"
14. #删除账户tuser1，state**=**absent代表删除账户的意思，name指定要删除的用户名是什么
15. #账户的家目录不会被删除，相当于执行userdel tuser1
16. **[**root@control ansible**]**# ansible test **-**m user **\**
17. **-**a "name=tuser2 state=absent remove=true"
18. #删除tuser2账户同时删除家目录、邮箱，相当于执行userdel **-**r tuser2

**步骤二：yum\_repository模块**

使用yum\_repository可以创建或修改yum源配置文件（ansible-doc yum\_repository）。

1. **[**root@control ansible**]**# ansible test **-**m yum\_repository **\**
2. **-**a "name=myyum description=hello baseurl=ftp://192.168.4.254/centos gpgcheck=no"
3. #新建一个yum源配置文件/etc/yum**.**repos**.**d**/**myyum**.**repo
4. #yum源文件名为myyum，该文件的内容如下：
5. **[**myyum**]**
6. baseurl **=** ftp**:**//192.168.4.254/centos
7. gpgcheck **=** **0**
8. name **=** hello
9. ## 验证：到node1主机ls **/**etc**/**yum**.**repos**.**d**/**查看该目录下是否有新的yum文件
10. **[**root@control ansible**]**# ansible test **-**m yum\_repository **\**
11. **-**a "name=myyum description=test baseurl=ftp://192.168.4.254/centos gpgcheck=yes gpgkey=…"
12. #修改yum源文件内容
13. **[**root@control ansible**]**# ansible test **-**m yum\_repository **-**a "name=myyum state=absent"
14. #删除yum源文件myyum

**步骤三：yum模块**

使用yum模块可以安装、卸载、升级软件包（ansible-doc yum），

state: present(安装)|absent(卸载)|latest(升级)。

1. **[**root@control ansible**]**# ansible test **-**m yum **-**a "name=unzip state=present"
2. #安装unzip软件包，state默认为present，也可以不写
3. ## 验证：到node1主机执行命令rpm **-**q unzip查看是否有该软件
4. **[**root@control ansible**]**# ansible test **-**m yum **-**a "name=unzip state=latest"
5. #升级unzip软件包，软件名称可以是**\***，代表升级所有软件包
6. **[**root@control ansible**]**# ansible test **-**m yum **-**a "name=unzip state=absent"
7. #调用yum模块，卸载unzip软件包，state**=**absent代表卸载软件
8. ## 验证：到node1主机执行命令rpm **-**q unzip查看该软件是否已经被卸载

**步骤四：service模块（ansible-doc service）**

service为服务管理模块（启动、关闭、重启服务等），

state:started|stopped|restarted，

enabled:yes设置开机启动。

1. **[**root@control ansible**]**# ansible test **-**m yum **-**a "name=httpd"
2. #调用yum模块，安装httpd软件包
3. ## 验证：到node1主机执行命令rpm **-**q httpd查看该软件是否被安装
4. **[**root@control ansible**]**# ansible test **-**m service **-**a "name=httpd state=started"
5. #调用service模块，启动httpd服务
6. ## 验证：到node1主机执行命令systemctl status httpd查看服务状态
7. **[**root@control ansible**]**# ansible test **-**m service **-**a "name=httpd state=stopped"
8. #调用service模块，关闭httpd服务
9. ## 验证：到node1主机执行命令systemctl status httpd查看服务状态
10. **[**root@control ansible**]**# ansible test **-**m service **-**a "name=httpd state=restarted"
11. #调用service模块，重启httpd服务
12. **[**root@control ansible**]**# ansible test **-**m service **-**a "name=httpd enabled=yes"
13. #调用service模块，设置httpd服务开机自启

**步骤五：逻辑卷相关模块（ansible-doc lvg、ansible-doc lvol）**

提示：做实验之前需要给对应的虚拟机添加额外磁盘，并创建磁盘2个分区

提示：可以使用前面学习过的parted或fdisk命令给磁盘创建分区

提示：这里的磁盘名称仅供参考，不要照抄！！！

lvg模块:创建、删除卷组(VG)，修改卷组大小，

state:present(创建)|absent(删除)。

1. **[**root@control ansible**]**# ansible test **-**m yum **-**a "name=lvm2"
2. #安装lvm2软件包，安装了lvm2软件后，才有pvcreate、vgcreate、lvcreate等命令
3. **[**root@control ansible**]**# ansible test **-**m lvg **-**a "vg=myvg pvs=/dev/sdb1"
4. #创建名称为myvg的卷组，该卷组由/dev/sdb1组成
5. #注意：这里的磁盘名称要根据实际情况填写
6. ## 验证：到node1主机执行命令pvs和vgs查看是否有对应的PV和VG
7. **[**root@control ansible**]**# ansible test **-**m lvg **-**a "vg=myvg pvs=/dev/sdb1,/dev/sdb2"
8. #修改卷组大小，往卷组中添加一个设备/dev/sdb2

lvol模块:创建、删除逻辑卷(LV)，修改逻辑卷大小，

state:present(创建)|absent(删除)。

1. **[**root@control ansible**]**# ansible test **-**m lvol **-**a "lv=mylv vg=myvg size=2G"
2. #使用myvg这个卷组创建一个名称为mylv的逻辑卷，大小为2G
3. ## 验证：到node1主机执行命令lvs查看是否有对应的LV逻辑卷
4. **[**root@control ansible**]**# ansible test **-**m lvol **-**a "lv=mylv vg=myvg size=4G"
5. #修改LV逻辑卷大小
6. **[**root@control ansible**]**# ansible test **-**m lvol **-**a "lv=mylv vg=myvg state=absent force=yes"
7. #删除逻辑卷，force**=**yes是强制删除
8. **[**root@control ansible**]**# ansible test **-**m lvg **-**a "vg=myvg state=absent"
9. #删除卷组myvg

附加信息（知识总结）：

ansible原理：

控制端主机自带很多模块（模块就是脚本）；

ansible通过ssh远程被管理主机，将控制端的模块（脚本）拷贝到被管理主机；

在被管理端主机执行模块（脚本），执行不同的模块可以实现不同的功能；

最后ansible退出ssh远程。

绝大多数模块（脚本）都需要参数才能执行成功！！！类似于shell脚本的位置变量！

拓扑结构如图-10所示。

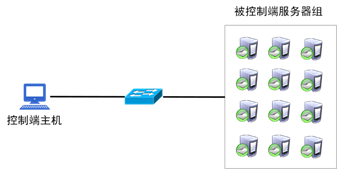


图-10

准备环境：

1）配置IP地址、主机名、系统光盘的YUM源、拷贝ansible软件包到控制端主机。

2）控制端主机修改/etc/hosts，实现所有主机的域名解析。

3）在控制端主机安装ansible软件：yum –y install \*

4）在控制端主机生产ssh密钥并传给所有其他主机：ssh-keygen、ssh-copy-id

5）控制端主机创建配置文件(没有这些文件需要手动创建)：

a）mkdir /root/ansible

b) /root/ansible/ansible.conf【inventory=/root/ansible/inventory】

c）/root/ansible/inventory

ansible命令的语法格式（需要背诵）：

ansible 主机集合 -m 模块名 -a "参数"

ansible常用模块（模块名称和作用需要背诵）：

command模块 ： 执行命令

shell模块： 执行命令（支持管道、重定向等）

script模块： 拷贝任意脚本到远程，再执行该脚本

file模块： 创建删除文件、目录、链接，修改文件、目录的权限等

copy模块： 将控制端主机的文件拷贝到远程其他主机

fetch模块： 将远程其他主机的文件拷贝到本地

lineinfile模块： 修改文件内容（按行修改）

replace模块： 修改文件内容（按关键词修改）

user模块： 创建、删除、修改账户

yum\_repository模块：创建、修改、删除YUM源配置文件

yum模块： 安装、卸载、升级软件，设置服务开机自启

lvg模块： 创建、删除、修改VG

lvol模块： 创建、删除、修改LV

提醒：全天的实验不需要死记硬背每个模块的每个参数，所有参数都可以查看帮助！

附加思考题（假设在没有创建ssh密钥的情况下）：

在没有创建ssh密钥的情况下，如果node1需要ssh远程node2是否需要输入密码？

如果node1需要ssh远程node2，应该输入谁的用户名和密码？

node1使用自己系统的账户和密码能否ssh远程node2，node2可以被随便登录还安全吗？

路人甲能否使用自己家的钥匙，去路人乙家（开路人乙家的门）？

node1执行命令ssh root@192.168.4.12，这里的root是谁的账户名？输入谁的密码？

如果使用真机windows的Xshell去ssh远程node2虚拟机，需要在windows也有一个root用户吗？