## 案例三：批量添加多个用户

批量创建多个用户的需求在生产环境中用的倒是不多，但是我们经常会在面试时的笔试题里看到。本案例也是取材于某企业笔试题里的一道shell脚本题，但我又增加了一些特殊需求。具体要求如下：

1）批量增加100个用户

2）用户名为user01 - user100

3）给所有用户都设定一个随机密码，密码长度12位，需要包含大小写字母以及数字

4）并把用户的密码保存到/data/user\_passwd文件中

5）这100个用户同属于users组

知识点一：系统用户管理

虽然在生产环境中对系统用户的管理的情况并不多，但是作为一个合格的系统管理员，是需要懂得如何增加、删除、更改用户操作的。

1）增加用户

useradd user1

这样会增加user1这个用户，如果不指定它所属的组，则会创建和用户名同名的一个组。要想指定组，那么该组首先已经存在，增加组的命令是：

groupadd grp1

然后增加用户的时候再指定组：

useradd user2 -g grp1

除了指定组外，还可以指定用户的id，如：

useradd -u 1100 user3

指定用户的家目录：

useradd -h /data/123 user4

指定用户的shell：

useradd -s /sbin/nologin user5

2）删除用户

最简单的方法：

userdel user5

但这样会保留user5的家目录以及邮件目录，要想连这些目录也删除的做法是：

userdel -r user5

3）更改用户信息

命令是usermod，它的用法和useradd几乎是一样的，命令的选项以及代表的含义都是相同的。如，更改用户uid：

usermod -u 1108 user3

更改用户家目录：

usermod -h /home/user4 user4

更改所属组

usermod -g grp2 user2

还有一个-G选项，它的含义是指定用户的扩展组，可以有多个

usermod -G grp1,users user1

知识点二：seq命令

先来看一下seq命令的最简单用法：

seq 1 5

1

2

3

4

5

seq可以打印一个序列出来，我们通常在shell脚本中把它作为循环遍历的对象。如：

# for i in `seq 1 5`; do echo "the number is $i"; done

the number is 1

the number is 2

the number is 3

the number is 4

the number is 5

seq后面有两个数，第一个是起始数字，后面一个是结束数字，规范写法应该在中间加上步长数值，表示每次增加多少，如果不写步长，默认就是1，再来看下面的例子：

seq 1 2 10

1

3

5

7

9

以上步长为2，下面的步长为-3，也就是说它可以倒序：

seq 10 -3 1

10

7

4

1

另外还有一个常用的选项-w，如下所示：

seq -w 5 10

05

06

07

08

09

10

本案例中，要求我们增加的用户名并不是简单的user1, user2, user3...，而是user01, user02, user03...，所以在使用for循环遍历的时候，需要加上-w。

知识点三：随机字符串

给用户设置密码，有一个命令叫做mkpasswd，需要安装expect包，如下：

yum install -y expect

直接运行mkpasswd命令会生成一个9位的随机字符串，包含大小写字母、数字以及特殊符号，例如：

# mkpasswd

04SQ|kfuc

当然，还可以给它增加一些选项，生成自定义的随机字符串，如，生产一个12位，包含3个数字、两个大写字母、0个特殊符号的随机串，可以这样做：

mkpasswd -l 12 -d 3 -s 0 -C 2

2i8marXHd0uy

知识点四：给用户设定或修改密码

系统添加完账户后，默认账户是没有设置密码的，所以该账户也就无法登录。给账户设定密码的命令叫做passwd，后面跟用户名，如果省略掉用户名就是给用户自己设定或更改密码。passwd命令修改密码需要和用户交互，如下所示：

passwd user1

更改用户 user1 的密码 。

新的 密码：

重新输入新的 密码：

passwd：所有的身份验证令牌已经成功更新。

这种形式在shell脚本中是不可取的，因为没有人为我们输入密码，解决这个问题有两个方案。

1）模拟用户输入两次密码

echo -e "your-passwd\nyour-passwd\n" |passwd

说明：echo -e可以把&quot;\n&quot;识别为回车，这样就相当于连续输入了两次&quot;your-passwd&quot;字符串。

2）使用passwd命令的--stdin选项

echo -e "your-passwd" |passwd --stdin

--stdin可以从标准输入读取令牌(只有根用户才能进行此操作)。

本案例参考脚本

学完以上储备知识，下面是最终的shell脚本内容，以此作为参考，希望你一定要亲自动手写一下。

#!/bin/bash

##批量添加100个用户，user01-user100

##作者：

##日期：2018-09-18

##版本：v0.1

#检查是否有mkpasswd命令

#如果没有该命令，需要安装expect包

if ! which mkpasswd &>;/dev/null

then

echo "没有mkpasswd命令，安装该命令："

yum install -y expect

fi

#判断/data/user\_passwd文件是否已经存在

#若存在，应该先删除掉

[ -f /data/user\_passwd ] && rm -f /data/user\_passwd

#因为100为三位数，所以只能遍历到99

for n in `seq -w 1 99`

do

pass=`mkpasswd -l 12 -s 0`

echo "添加用户user\_$n"

useradd -g users user\_$n

echo "给用户user\_$n设定密码"

echo $pass |passwd --stdin user\_$n

echo "user\_$n $pass" >>/data/user\_passwd

done

pass=`mkpasswd -l 12 -s 0`

echo "添加用户user\_100"

useradd -g users user\_100

echo "给用户user100设定密码"

echo $pass|passwd --stdin user\_100

echo "user\_100 $pass" >>/data/user\_passwd

说明： [ -f /data/user\_passwd ] && rm -f /data/user\_passwd 这条命令，相当于：

if [ -f /data/user\_passwd ]

then

rm -f /data/user\_passwd

fi

这里的&&表示，当它前面的命令执行成功才会执行后面的命令。和它相对应的还有一个||符号，它表示当前面的命令执行不成功时，才会执行后面的命令，在shell脚本中往往替代if判断