## 案例十：检测域名是否到期

任何一个网站对外提供服务，都需要有一个域名，就像每个人都有人名，网站的名字就是这个域名。如果你想做一个网站，首先需要去购买一个域名，与其说是购买不如说是租用，因为你要想使用这个域名每年都需要缴纳一笔费用。既然是租用，所以就有期限这一说。假如你在2018-10-10购买了一个域名，期限为一年，则到2019-10-10该域名就会过期，在到期前你没有续费，则该域名就会被收回，其他人可以注册并使用这个域名。

当我们名下域名数量非常多的时候，则很容易忘记为这些域名续费从而导致过期并回收。本案例的需求是，写一个shell脚本来监控指定的域名是否到期，具体要求如下：

1）写一个函数，域名以参数的形式传递给这个函数

2）域名到期前的一周和到期后的一周（两周时间），每天都要发告警邮件

3）脚本每天执行一次

知识点一：whois

一个域名的信息，比如所有者邮箱、电话、地址和啥时候过期等都是公开的，你可以在浏览器里访问https://www.whois.net 进行查询。国内也有不少类似的网站可以查询域名信息。那么在Linux命令行下如何查询呢？

# whois aminglinux.com

[Querying whois.verisign-grs.com]

[whois.verisign-grs.com]

Domain Name: AMINGLINUX.COM

Registry Domain ID: 1800256822\_DOMAIN\_COM-VRSN

Registrar WHOIS Server: whois.55hl.com

Registrar URL: http://www.55hl.com

Updated Date: 2018-05-04T22:57:37Z

Creation Date: 2013-05-10T06:02:05Z

Registry Expiry Date: 2021-05-10T06:02:05Z

Registrar: JIANGSU BANGNING SCIENCE & TECHNOLOGY CO. LTD

Registrar IANA ID: 1469

Registrar Abuse Contact Email: abuse@55hl.com

Registrar Abuse Contact Phone: +86 025 86883426 1009

Domain Status: ok https://icann.org/epp#ok

Name Server: F1G1NS1.DNSPOD.NET

Name Server: F1G1NS2.DNSPOD.NET

DNSSEC: unsigned

URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/

>>> Last update of whois database: 2018-10-09T14:46:17Z <<<

默认系统并没有这个命令，需要安装jwhois包，CentOS系统这样安装：

# yum install -y jwhois

本案例需要监控域名过期时间，所以我们要关注的行是'Expiry Date'。还有一个问题我们不得不考虑，不同的域（.com, .cn）查询到的结果有所不同，比如.cn的结果是这样的：

# whois aminglinux.cn

[Querying whois.cnnic.cn]

[whois.cnnic.cn]

Domain Name: aminglinux.cn

ROID: 20160322s10001s82727381-cn

Domain Status: serverHold

Registrant ID: zl40fvi7lfo56d

Registrant: 个人用户

Registrant Contact Email: 306798658@qq.com

Sponsoring Registrar: 北京新网数码信息技术有限公司

Name Server: ns11.xincache.com

Name Server: ns12.xincache.com

Registration Time: 2016-03-22 17:42:01

Expiration Time: 2019-03-22 17:42:01

DNSSEC: unsigned

知识点二：cut命令

其实在案例四和案例八中已经出现过cut命令，在这里我再总结一下它的用法。

语法： cut -d '分隔字符' [-cf] n 这里的n是数字

-d ：后面跟分隔字符，分隔字符要用单引号括起来

-c ：后面接的是第几个字符

-f ：后面接的是第几个区块

# cat /etc/passwd |cut -d ':' -f 1 |head -n5

root

bin

daemon

adm

lp

说明：-d 后面跟分隔字符，这里使用冒号作为分割字符，-f 1 就是截取第一段，-f和1之间的空格可有可无。

# head -n2 /etc/passwd|cut -c2

o

i

# head -n2 /etc/passwd|cut -c1

r

b

# head -n2 /etc/passwd|cut -c1-10

root:x:0:0

bin:x:1:1:

# head -n2 /etc/passwd|cut -c5-10

:x:0:0

x:1:1:

-c 后面可以是1个数字n，也可以是一个区间n1-n2，还可以是多个数字n1,n2,n3

# head -n2 /etc/passwd|cut -c1,3,10

ro0

bn:

知识点三：进程控制

当运行一个进程时，你可以使它暂停（按Ctrl+z），然后使用fg命令恢复它，利用bg命令使他到后台运行，你也可以使它终止（按Ctrl+c组合键）。

# vi test1.txt

testtestsstststst

使用”vi”编辑test1.txt, 随便输入一些内容，按 “ESC” 后， 使用”Ctrl + z”组合键使任务暂停：

# vi test1.txt

[1]+ Stopped vi test1.txt

可以看到提示”vi test1.txt”已经停止了，然后使用fg命令恢复它，此时又进入刚才的”vi”窗口了。再次使其暂停，然后输入 jobs, 可以看到在被暂停或者在后台运行的任务:

# jobs

[1]+ Stopped vi test1.txt

如果想把暂停的任务丢在后台跑起来，就使用bg命令:

# bg

[1]+ vi test1.txt &

[1]+ Stopped vi test1.txt

但是 vi 似乎并不支持在后台运行，那换一个其他的命令：

# sar 1 > /tmp/1.log

^Z //这里按了Ctrl + z组合键

[2]+ Stopped sar 1 > /tmp/1.log

# jobs

[1]- Stopped vi test1.txt

[2]+ Stopped sar 1 > /tmp/1.log

# bg 2

[2]+ sar 1 > /tmp/1.log &

在上面的例子中，又有一个新的知识点需要你知道，那就是多个被暂停的任务会有编号，使用jobs命令可以看到两个任务，使用bg或者fg的时候，就需要在后面加一个编号了，上面例子中使用”bg 2”把第二个被暂停的任务丢到后台跑起来了，丢入后台需要使用在命令后边加一个'&'符号，中间有个空格。

丢到后台的任务如何关掉呢？如果你没有退出刚才的shell, 那么先使用”fg 编号”把任务调到前台，然后使用 “Ctrl + c” 结束任务：

# fg 2

sar 1 > /tmp/1.log

^C //这里使用的Ctrl + c组合键

另一种情况则是，关闭到当前的shell, 再次打开另一个shell时，使用jobs命令并不会显示在后台运行或者被暂停的任务，要想停掉它的话，则需要先知道其pid, 然后使用kill命令杀死那个进程。

# sar 1 > /tmp/1.log &

[1] 9433

# ps aux |grep sar

root 9433 0.0 0.0 6180 516 pts/2 S 09:57 0:00 sar 1

root 9435 0.0 0.0 103308 848 pts/2 S+ 09:58 0:00 grep sar

在shell脚本中，多条指令执行是有先后顺序的，也就是说只有前面的指令执行完（不管成功与否）后面的指令才会执行。如果有一条指令运行时间比较久，则会阻碍后续指令执行。若不想让这条执行慢的指令影响到后面指令就可以在该指令后面加一个'&'，即把它丢到后台去。使用'&'符号把任务丢入后台运行，会显示pid信息，如果忘记这个pid，我们还可以使用'ps aux'命令找到那个进程。想结束掉该进程，需要使用 kill 命令：

# kill 9433

[1]+ 已终止 sar 1 > /tmp/1.log

kill命令语法很简单，直接在后面加pid即可，如果遇到杀不死的进程时，可以在kill 后面加一个选项： kill -9 [pid]

知识点四：环境变量PATH

在讲环境变量之前，先介绍一下命令which，它用于查找某个命令的绝对路径。示例命令如下：

# which rmdir

/usr/bin/rmdir

# which rm

alias rm='rm -i'

/usr/bin/rm

# which ls

alias ls='ls --color=auto'

/usr/bin/ls

其中 rm 和 ls 是两个特殊的命令，它们使用alias命令做了别名。我们用的rm命令实际上是 rm -i ，加上 -i 选项后，删除文件或者命令时都会询问是否确定要删除，这样做比较安全。

命令which不常使用，平时只用来查询某个命令的绝对路径或者判断某个命令是否存在。在上面的示例中，用which查到rm命令的绝对路径为/usr/bin/rm。那么你是否会问：“为什么我们使用命令时，只是直接打出了命令，而没有使用这些命令的绝对路径呢？”这是环境变量 PATH 在起作用。请输入如下命令：

# echo $PATH

/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin

这里的echo用来输出$PATH的值。因为/bin目录在PATH的设定中，所以自然可以找到ls 。但值得注意的是，由于PATH里有/root目录，如果你将ls移到/root目录下，当执行ls命令时，系统自然就找不到可执行文件了，它会提示'command notfound!'。示例命令如下：

# mv /usr/bin/ls /root/

# ls

-bash: /usr/bin/ls: command notfound!

那么，该如何解决上面的这种问题呢？有两种方法，一种方法是直接将/root这个路径加入到 $PATH 当中，命令如下：

# PATH=$PATH:/root

# echo $PATH

/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin:/root

# ls

anaconda-ks.cfg ls

另一种方法是使用绝对路径，命令如下：

# /root/ls

anaconda-ks.cfg ls

知识点五：判断一个变量的值是否为空

在shell脚本中，如果一个变量没有成功赋值就被引用了，则会影响到脚本的正常执行，判断一个变量的值是否为空有两种方法：

1）用-z（zero的意思）

# b= //给变量b赋值空

# if [ -z "$b" ]; then echo "The value of b is null.";else echo "The value of b is $b"; fi

The value of b is null.

# b=1

# if [ -z "$b" ]; then echo "the value of b is null.";else echo "The value of b is $b."; fi

The value of b is 1.

2）用-n（not null的意思）

# a=1

# if [ -n "$a" ]; then echo "The value of a is $a.";else echo "The value of a is null."; fi

The value of a is 1.

# a=

# if [ -n "$a" ]; then echo "The value of a is $a.";else echo "The value of a is null."; fi

The value of a is null.

知识点六：判断某个进程是否存在

前面我们介绍过ps查看进程，但是需要结合grep，而且需要统计行数，其实还有一个更简单的用法：

# sleep 10 &

[1] 32742

# pgrep sleep

32742

知识点七：杀死进程

在知识点三中，我们用到了kill命令杀死进程，但是需要知道进程的pid。下面再来介绍一个简单的方法：

# sar 1 >/tmp/1.log &

# killall sar

[1]+ 已终止 sar 1 > /tmp/sar.log</code></pre>

说明：killall和kill不同的地方在于killall可以直接跟进程名。killall也支持-9选项，有时候用killall杀死进程不好用，需要带上-9。

本案例参考脚本

#!/bin/bash

#检测域名是否过期

#作者：

#日期：2018-10-10

#版本：v0.2

mail\_u=admin@admin.com

#当前日期时间戳，用于和域名的到期时间做比较

t1=`date +%s`

#检测whois命令是否存在，不存在则安装jwhois包

is\_install\_whois()

{

which whois >/dev/null 2>/dev/null

if [ $? -ne 0 ]

then

yum install -y jwhois

fi

}

notify()

{

e\_d=`whois $1|grep 'Expiry Date'|awk '{print $4}'|cut -d 'T' -f 1`

#如果e\_d的值为空，则过滤关键词'Expiration Time'

if [ -z "$e\_d" ]

then

e\_d=`whois $1|grep 'Expiration Time'|awk '{print $3}'`

fi

#将域名过期的日期转化为时间戳

e\_t=`date -d "$e\_d" +%s`

#计算一周一共有多少秒

n=`echo "86400\*7"|bc`

e\_t1=$[$e\_t-$n]

e\_t2=$[$e\_t+$n]

if [ $t1 -ge $e\_t1 ] && [ $t1 -lt $e\_t ]

then

python mail.py $mail\_u "Domain $1 will to be expired." "Domain $1 expire date is $e\_d."

fi

if [ $t1 -ge $e\_t ] && [ $t1 -lt $e\_t2 ]

then

python mail.py $mail\_u "Domain $1 has been expired" "Domain $1 expire date is $e\_d."

fi

}

#检测上次运行的whois查询进程是否存在

#若存在，需要杀死进程，以免影响本次脚本执行

if pgrep whois &>/dev/null

then

killall -9 whois

fi

is\_install\_whois

for d in aaa.net aaa.com bbb.com aaa.cn ccc.com

do

notify $d &

done