Linux 云计算集群架构师

学神 IT 教育：从零基础到实战，从入门到精通！

联系方式：

学神 IT 教育官方网站: http://www.xuegod.cn

Linux 云计算架构师进阶学习群 QQ 群: 1072932914

学习顾问：小语老师　　学习顾问：边边老师　　学神微信公众号

微信扫码添加学习顾问微信，同时扫码关注学神公众号了解最新行业动态，获取更多学习资料及答疑就业服务！

第十六章　Linux 计划任务与日志的管理

本节所讲内容：

16.1　计划任务-at-cron-计划任务使用方法

大家平常都会有一些比如说：你每天固定几点起床？每天按时上班打卡、每月 15 号准时开工资、每年 2 月 14 你俩口子某某纪念日等这些诸如此类，这些都是定时发生的。或者说是通俗点说：例行公事；还比如说我们还会遇到一些突发事件，临时几点过来加个班？刚好晚上几点聚个餐？

像上面这些情况，如果事少的话你大脑可以记住，如果事很多，像老板经理董事长每天的工作安排，通常都是记在一些本上，或者闹铃提醒等。

那么，咱们的 LINUX 系统和上面的情况也很类似，我们也可以通过一些设置。来让电脑定时提醒我们该做什么事了。或者我们提前设置好，告诉电脑你几点做什么几点做什么，这种我们就叫它定时任务。而遇到一些需要执行的事情或任务。我们也可以通过命令来告诉电脑一会临时把这个工作给做一下

总结：在我们 LINUX 中，我们可以通过 crontab 和 at 这两个东西来实现这些功能的

计划任务的作用：是做一些周期性的任务，在生产中的主要用来定期备份数据

CROND：这个守护进程是为了周期性执行任务或处理等待事件而存在

任务调度分两种：系统任务调度，用户任务调度

计划任务的安排方式分两种:

一种是定时性的，也就是例行。就是每隔一定的周期就要重复来做这个事情

一种是突发性的，就是这次做完了这个事，就没有下一次了，临时决定，只执行一次的任务

at 和 crontab 这两个命令：

at：它是一个可以处理仅执行一次就结束的指令

crontab：它是会把你指定的工作或任务，比如：脚本等，按照你设定的周期一直循环执行下去

16.1　Linux 计划任务管理

16.1.1　at 计划任务的使用

语法格式：　at　时间　；服务：atd

```
[root@xuegod63 ~]# yum -y install at
[root@xuegod63 ~]# systemctl start atd          #开启 atd 服务
[root@xuegod63 ~]# systemctl status atd         #查看 atd 服务状态
[root@xuegod63 ~]# systemctl is-enabled atd
#查看是否开始开机启动服务，如果弹出 enabled，说明开机启动此服务
```

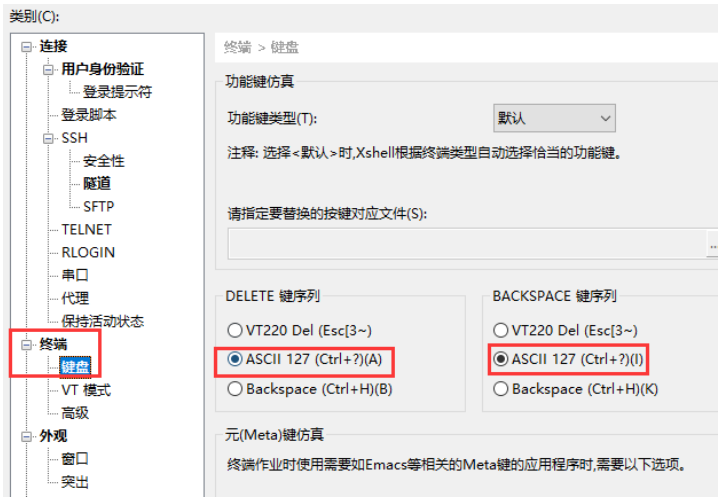
在 Centos6 查看开机启动服务：

```
[root@xuegod63 ~]# chkconfig --list  |  grep atd          #此命令在 centos7 上不能执行
```

在 Centos7 之后的系统查看是否开机启动：

```
[root@xuegod63 ~]# systemctl list-unit-files
```

实战-使用 at 创建计划任务



[root@xuegod63 ~]# date　　#查看系统时间

2018 年 05 月 21 日 星期一 20:43:29 CST

[root@xuegod63 ~]# at 20:46　　#注意：如果是上午时间，后面加上 am，9:20am

at> mkdir /tmp/xuegod　　#输入你要执行的命令

at> touch /tmp/xuegod/a.txt

#结束：ctrl+d

[root@xuegod63 ~]# at -l　　#查看计划任务

[root@xuegod63 ~]# atq　　#查看计划任务

检查 at 计划任务运行结果:

[root@xuegod63 ~]# ls /tmp/xuegod/

a.txt

互动：如果正在执行命令，ctrl+D ，按成 ctrl+S 会怎么样？ 尤其是使用 vim 保存，按成 ctrl+s

解决： ctrl+s 在 linux 下是锁定屏幕显示的意思，这时整个界面被锁定，不能进行正常输入。使用 ctrl+q 来解除锁定,

16.1.2 查看和删除 at 将要执行的计划任务

这个查看，只能看到还没有执行的。如果这个任务已经开始执行或者执行完成了，是看不到的

[root@xuegod63 ~]# at -l

5　　Sat Aug 19 20:50:00 2017 a root

任务编号　执行的时间　　队列 执行者

5　　Fri Oct 28 20:55:00 2016　　a　　root

[root@xuegod63 ~]# at -c 5

#-c:打印任务的内容到标准输出，查看 5 号计划任务具体内容

查看定时任务内容

[root@xuegod63 ~]# ls /var/spool/at/

a00003018452cb　a0000501845084　spool

[root@xuegod63 ~]# tail -10 /var/spool/at/a0000501845084

at 计划任务的特殊写法

[root@ xuegod63 ~]# at 20:00 2030-12-29　　在某天

[root@ xuegod63 ~]# at now +10min　　在 10 分钟后执行
[root@ xuegod63 ~]# at 17:00 tomorrow　　明天下午 5 点执行
[root@xuegod63 ~]# at 6:00 pm +3 days　在 3 天以后的下午 6 点执行
[root@xuegod63 ~]# at 23:00 < /root/a.txt　　把 a.txt 的内容输入给他也可以
vim a.txt
mkdir /opt/test
touch /opt/test/test.txt

删除 at 计划任务
语法：　atrm　任务编号
[root@xuegod63 ~]# at -l
3　　Tue May 22 08:43:00 2018 a root
5　　Mon May 21 23:00:00 2018 a root
[root@xuegod63 ~]# atrm 3 5
[root@xuegod63 ~]# at -l
3　　Tue May 22 08:43:00 2018 a root

16.1.3　crontab 定时任务的使用

crond 命令定期检查是否有要执行的工作，如果有要执行的工作便会自动执行该工作
cron 是一个 linux 下的定时执行工具，可以在无需人工干预的情况下运行作业。
linux 任务调度的工作主要分为以下两类：
系统执行的工作：系统周期性所要执行的工作，如更新 whatis 数据库　updatedb 数据库，日志定期切割，收集系统状态信息，/tmp 定期清理
启动 crond 服务
[root@xuegod63 at]# systemctl start crond
[root@xuegod63 at]# systemctl enable crond

16.1.4　cron 命令参数介绍

crontab 的参数：
crontab -l　　#列出当前用户下的 cron 服务的详细内容
crontab -u user1 -l　　#列出指定用户 user1 下的 cron 服务的详细内容
crontab -r　　#删除 cron 服务
crontab -e　　#编辑 cron 服务
例如:
crontab -u root -l　　# root 查看自己的 cron 计划任务
crontab -u user1 -r　　# root 想删除 user1 的 cron 计划任务
crontab -e 编辑时的语法

星期日用 0 或 7 表示
一行对应一个任务，特殊符号的含义：
*　　代表取值范围内的数字　(任意/每)
/　　指定时间的间隔频率 */10　0-23/2　放在小时下（在 0-23 点之间，每隔 2 小时执行一次)
-　　代表从某个数字到某个数字　　8-17　　8 到 17 之间执行

，　分开几个离散的数字 6,10-13,20　　6 执行，10 到 13 之间执行，20 执行

16.1.5　创建计划任务

例 1：每天凌晨 2 点 1 分开始备份数据

[root@xuegod63 spool]# crontab -e　　#添加计划任务

1 2 * * *　tar zcvf /opt/grub2.tar.gz　/boot/grub2

[root@xuegod63 ~]# crontab　-l　　#查看

例 2：黑客：以非 root 用户添加计划任务。　最好使用已经存在系统用户添加。这里使用 bin 用户来添加

[root@xuegod63 ~]# crontab -u bin -e

*/1 * * * * echo "aaaaaaa"　>> /tmp/bin.txt

1 * * * * 每小时第 1 分钟

*/1 * * * * 每分钟

排查：

[root@xuegod63 ~]# crontab　-u bin -l

*/1 * * * * echo "aaaaaaa"　>> /tmp/bin.txt

互动：如何排查所有用户的计划任务？　不会：1　有思路：6

做黑客要有一个很扎实的基础，还要有很好的思维

注：所有用户的计划任务，都会在/var/spool/cron/下产生对应的文件

[root@xuegod63 ~]# ll /var/spool/cron/

total 8

-rw------- 1 root root 42 Nov 12 10:11 bin

-rw------- 1 root root 19 Nov 12 10:06 root

所以后期可以使用这一招排查，黑客是否在你的机器中安装了定时任务

16.1.6　系统级别的计划任务

系统级别的计划任务

[root@xuegod63 etc]# ll /etc/crontab

-rw-r--r--. 1 root root 451 Dec 28　2013 /etc/crontab

这个是系统任务调度的配置文件

[root@xuegod63 etc]# vim /etc/crontab

SHELL=/bin/bash　　#指定操作系统使用哪个 shell

PATH=/sbin:/bin:/usr/sbin:/usr/bin　　#系统执行命令的搜索路径

MAILTO=root　　#将执行任务的信息通过邮件发送给 xx 用户

For details see man 4 crontabs

Example of job definition:

.---------------- minute (0 - 59)

|　.------------- hour (0 - 23)

|　|　.---------- day of month (1 - 31)

|　|　|　.------- month (1 - 12) OR jan,feb,mar,apr ...

|　|　|　|　.---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat

|　|　|　|　|

* * * * * user-name command to be executed

也可以直接在/etc/crontab 中添加计划任务

使用 crontab 命令的注意事项：

环境变量的问题

清理你的邮件日志 ，比如使用重定向 >/dev/null 2>&1

[root@xuegod63 bin]# ls /etc/cron #按两下 tab 键

cron.d/ cron.deny cron.monthly/ cron.weekly/

cron.daily/ cron.hourly/ crontab

注： cron.d/ #是系统自动定期需要做的任务，但是又不是按小时，按天，按星期，按月来执行的，那么就放在这个目录下面。

cron.deny #控制用户是否能做计划任务的文件;

cron.monthly/ #每月执行的脚本;

cron.weekly/ #每周执行的脚本;

cron.daily/ #每天执行的脚本;

cron.hourly/ #每小时执行的脚本;

crontab #主配置文件 也可添加任务;

16.1.7 实战-常见的计划任务写法和案例

常见写法：

每天晚上 21:00 重启 apache

```
0 21 * * * /etc/init.d/httpd restart
```

每月 1、10、22 日的 4 : 45 重启 apache。

```
45 4 1,10,22 * * /etc/init.d/httpd restart
```

每月 1 到 10 日的 4 : 45 重启 apache。

```
45 4 1-10 * * /etc/init.d/httpd restart
```

每隔两天的上午 8 点到 11 点的第 3 和第 15 分钟重启 apache

```
3,15 8-11 */2 * * /etc/init.d/httpd restart
```

晚上 11 点到早上 7 点之间，每隔一小时重启 apach

```
0 23-7/1 * * * /etc/init.d/apach restart
```

周一到周五每天晚上 21:15 寄一封信给 root@xuegod:

```
15 21 * * 1-5 mail -s "hi" root@xuegod < /etc/fstab
```

https://tool.lu/crontab

可以查看接下来 7 次的执行日期

互动：crontab 不支持每秒。 每 2 秒执行一次脚本，怎么写?

在脚本的循环中，添加命令 sleep 2 ，执行 10 次自动退出，然后添加，计划任务：

vim sh.sh

```
#!/bin/bash
for (( i=1;i<=10;i++ ))
do
        echo /tmp/`date "+%Y%m%d%H%M%S"`.txt
        touch /tmp/`date "+%Y%m%d%H%M%S"`.txt
```

```
        sleep 2
done
或者
#!/bin/bash
for i in {1..10}
do
        echo /tmp/`date "+%Y%m%d%H%M%S"`.txt
        touch /tmp/`date "+%Y%m%d%H%M%S"`.txt
        sleep 2
done

* * * * *   sh /root/sh.sh
 案例要求：
每天 22：00 备份/etc/目录到/tmp/backup 下面
将备份命令写入一个脚本中
每天备份文件名要求格式：  2017-08-19_etc.tar.gz
在执行计划任务时，不要输出任务信息
存放备份内容的目录要求只保留三天的数据
[root@xuegod63 ~]# cat backup.sh

#!/bin/bash
[ -d /tmp/backup ] || mkdir -p /tmp/backup
[ -f /tmp/backup/`date +%F`_etc.tar.gz ] || tar czf /tmp/backup/`date
+%Y-%m-%d`_etc.tar.gz /etc
find /tmp/backup -name "*.tar.gz" -mtime +3 -exec rm -rf {} \;
#find /tmp/backup -name "*.tar.gz" -mtime -3 |xargs ls -lh


-mtime -1，当前时间为 2021-04-10 22:31，2021-04-09 22:31~ 2021-04-10 22:31 之间
修改的文件
-mtime 1，当前时间为 2021-04-10 22:31，2021-04-08 22:31~ 2021-04-09 22:31 之间修
改的文件
-mtime +1，当前时间为 2021-04-10 22:31，2021-04-08 22:21 之前修改的文件
[root@xuegod63 ~]# crontab -l
13 21 * * * echo "xuegod1707" > /tmp/a.txt
0 22 * * * /root/backup.sh &   >/dev/null
注：工作中备份的文件不要放到/tmp,因为过一段时间，系统会清空备/tmp 目录
```

16.2　日志的种类和记录的方式自定义 ssh 服务日志类型存储位置

在 centos8 中，系统日志消息由两个服务负责处理：systemd-journald 和 rsyslog

16.2.1　常见日志文件的作用

系统日志文件概述：/var/log 目录保管由 rsyslog 维护的，里面存放的一些特定于系统和服务的日志文件

日志文件　用途

/var/log/message　大多数系统日志消息记录在此处。有也例外的：如与身份验证，电子邮件处理相关的定期作业任务等

/var/log/secure　　安全和身份验证相关的消息和登录失败的日志文件。　ssh 远程连接产生的日志

/var/log/secure　　安全和身份验证相关的消息和错误的日志文件

/var/log/maillog　与邮件服务器相关的消息日志文件

/var/log/cron 与定期执行任务相关的日志文件

/var/log/boot.log　与系统启动相关的消息记录

例 1：查看哪个 IP 地址经常暴力破解系统用户密码

[root@xuegod63 ~]# ssh root@192.168.1.63　　#故意输错 3 次密码

[root@xuegod63 log]# grep Failed /var/log/secure

Aug 19 21:55:42 panda sshd[84029]: Failed password for root from 10.10.30.130 port 50916 ssh2

Aug 19 21:55:44 panda sshd[84029]: Failed password for root from 10.10.30.130 port 50916 ssh2

Aug 19 21:55:47 panda sshd[84029]: Failed password for root from 10.10.30.130 port 50916 ssh2

Aug 19 21:55:52 panda sshd[84034]: Failed password for root from 10.10.30.130 port 50917 ssh2

[root@xuegod63 log]# grep Failed /var/log/secure|awk '{print $11}'

192.168.1.63

192.168.1.63

192.168.1.63

[root@xuegod63 log]# grep Failed /var/log/secure|awk '{print $11}'|uniq -c

3 192.168.1.63

注：awk '{print $11}'　　#以空格做为分隔符，打印第 11 列的数据

uniq 命令用于报告或忽略文件中的重复行，-c 或——count：在每列旁边显示该行重复出现的次数；

例 2：/var/log/wtmp 文件的作用

/var/log/wtmp 也是一个二进制文件，记录每个用户的登录次数和持续时间等信息。

可以用 last 命令输出 wtmp 中内容：　last　显示到目前为止，成功登录系统的记录

[root@xuegod63 ~]# last

root　　pts/2　　192.168.1.8　　Tue May 22 00:35　　still logged in

root　　pts/2　　192.168.1.8　　Mon May 21 20:42 - 00:35　(03:53)

或：

[root@xuegod63 ~]# last -f /var/log/wtmp

例 3：使用 /var/log/btmp 文件查看暴力破解系统的用户

/var/log/btmp 文件是记录错误登录系统的日志。如果发现/var/log/btmp 日志文件比较大，大于 1M，就算大了，就说明很多人在暴力破解 ssh 服务，此日志需要使用 lastb 程序查看

[root@xuegod63 ~]# lastb

root　　ssh:notty　　xuegod63.cn　　Mon May 21 21:49 - 21:49　(00:00)

root ssh:notty xuegod63.cn Mon May 21 21:49 - 21:49 (00:00)

发现后，使用防火墙，拒绝掉：命令如下：

iptables -A INPUT -i ens33 -s 192.168.1.63（暴力破解地址） -j DROP

#将新规则追加于尾部入站请求 ens33 网卡，地址是 192.168.1.63 的 IP，被丢弃。

查看恶意 ip 试图登录次数：

lastb | awk '{ print $3}' | uniq -c | sort -n

打印第三列 去重并显示复次数 按字符串数值大小排序

清空日志：

方法 1：[root@xuegod63 ~]# > /var/log/btmp

方法 2： rm -rf /var/log/btmp && touch /var/log/btmp

两者的区别?

使用方法 2，因为创建了新的文件，而正在运行的服务，还用着原来文件的 inode 号和文件描述码，所需要重启一下 rsyslog 服务。建议使用方法 1 > /var/log/btmp

16.2.2 日志的记录方式

分类 级别

日志的分类:

daemon 后台进程相关

kern 内核产生的信息

lpr 打印系统产生的

authpriv 安全认证

cron 定时相关

mail 邮件相关

syslog 日志服务本身的

news 新闻系统

local0~7 自定义的日志设备

local0-local7 8 个系统保留的类， 供其它的程序使用或者是用户自定义

日志的级别: 轻 重

编码	优先级	严重性
7	debug	信息对开发人员调试应用程序有用，在操作过程中无用
6	info	正常的操作信息，可以收集报告，测量吞吐量等
5	notice	注意，正常但重要的事件，
4	warning	警告，提示如果不采取行动。将会发生错误。比如文件系统使用 90%
3	err	错误，阻止某个模块或程序的功能不能正常使用
2	crit	关键的错误，已经影响了整个系统或软件不能正常工作的信息
1	alert	警报,需要立刻修改的信息
0	emerg	紧急，内核崩溃等严重信息

16.2.3 rsyslog 日志服务

rhel5 ->服务名称 syslog ->配置文件 /etc/syslog.conf

rhel6-7 ->服务名称 rsyslog ->配置文件 /etc/rsyslog.conf

我们来查看一下日志的配置文件信息：

编辑配置文件 vim /etc/rsyslog.conf

*.info;mail.none;authpriv.none;cron.none /var/log/messages
authpriv.* /var/log/secure
mail.* -/var/log/maillog
cron.* /var/log/cron
.emerg :omusrmsg:
uucp,news.crit /var/log/spooler
local7.* /var/log/boot.log
注释：
#$UDPServerRun 514 #允许 514 端口接收使用 UDP 协议转发过来的日志
#$InputTCPServerRun 514 ##允许 514 端口接收使用 TCP 协议转发过来的日志
Centos8 为
#input(type="imudp" port="514") #允许 514 端口接收使用 UDP 协议转发过来的日志
#input(type="imtcp" port="514") ##允许 514 端口接收使用 TCP 协议转发过来的日志
#kern.* 内核类型的所有级别日志 -->存放到--> /dev/console
*.info;mail.none;authpriv.none;cron.none /var/log/messages
所有的类别并且级别是 info 以上 除了 mail,authpriv,cron (产生的日志太多,不易于查看)
类别.级别
authpriv.* 认证的信息-->存放--> /var/log/secure
mail.* 邮件相关的信息--> 存放--> -/var/log/maillog
cron.* 计划任务相关的信息-->存放--> /var/log/cron
local7.* 开机时显示的信息-->存放--> /var/log/boot.log
注：
“- ”号： 邮件的信息比较多,先将数据存储到内存,达到一定大小,全部写到硬盘.有利于减少 I/O 进程的开销
数据存储在内存,如果关机不当数据消失

16.2.4 日志输入的规则
. info 大于等于 info 级别的信息全部记录到某个文件
.=级别 仅记录等于某个级别的日志
例:.=info 只记录 info 级别的日志
.! 级别 除了某个级别意外,记录所有的级别信息
例.!err 除了 err 外记录所有
.none 指的是排除某个类别 例： mail.none 所有 mail 类别的日志都不记录

16.2.5 实战-自定义 ssh 服务的日志类型和存储位置
[root@xuegod63 ~]# vim /etc/rsyslog.conf #以 73 行下，插入以下红色标记内容
65 local7.* /var/log/boot.log
66 local0.* /var/log/sshd.log
注：把自定义 local0 类别的日志，保存到 /var/log/sshd.log 路径

定义 ssh 服务的日志类别为 local0，编辑 sshd 服务的主配置文件
[root@xuegod63 log]# vim /etc/ssh/sshd_config #插入
SyslogFacility local0

改：40　SyslogFacility AUTHPRIV

为：40　SyslogFacility local0

#把 sshd 服务日志 默认的安全认证类别 改成我们的 自定义 local0 类别

先重启 rsyslog 服务(生效配置)

[root@xuegod63 log]# systemctl restart rsyslog

再重启 sshd 服务.生成日志

[root@xuegod63 log]# systemctl restart sshd

验证是否生成日志并查看其中的内容，

[root@xuegod63 ~]# cat　/var/log/sshd.log　　#说明修改成功

May 22 00:19:54 xuegod63 sshd[44737]: Server listening on 0.0.0.0 port 22.

May 22 00:19:54 xuegod63 sshd[44737]: Server listening on :: port 22.

上面对就的信息：时间　　主机　服务　　进程 ID　　相关的信息

互动：如何防止日志删除?

[root@xuegod63 ~]# chattr +a /var/log/sshd.log

[root@xuegod63 ~]# lsattr /var/log/sshd.log

-----a---------- /var/log/sshd.log

[root@xuegod63 ~]# systemctl restart sshd

[root@xuegod63 ~]# cat /var/log/sshd.log　#重启服务，查看日志有所增加

注：这个功能看着很强大，其实不实用，因这样会让系统日志切割时报错，最主的是，黑客可以取消这个属性。

[root@xuegod63 ~]# chattr -a /var/log/sshd.log　　#取消，这里一定要取消，不然后面做日志切割报错

互动：当日志太多，导致日志很文件大怎么办?

16.3　实战-日志切割-搭建远程日志收集服务器

16.3.1　日志的切割

在 linux 下的日志会定期进行滚动增加，我们可以在线对正在进行回滚的日志进行指定大小的切割（动态），如果这个日志是静态的。比如没有应用向里面写内容。那么我们也可以用 split 工具进行切割；其中 Logrotate 支持按时间和大小来自动切分,以防止日志文件太大。

logrotate 配置文件主要有：

/etc/logrotate.conf 以及 /etc/logrotate.d/ 这个子目录下的明细配置文件。

logrotate 的执行由 crond 服务调用的。

[root@xuegod63 ~]# vim /etc/cron.daily/logrotate　　#查看 logrotate 脚本内容

logrotate 程序每天由 cron 定时任务在指定的时间启动

日志是很大的,如果让日志无限制的记录下去 是一件很可怕的事情，日积月累就有几百兆占用磁盘的空间，

如果你要找出某一条可用信息：　海底捞针

日志切割:

当日志达到某个特定的大小,我们将日志分类,之前的日志保留一个备份,再产生的日志创建一个同名的文件保存新的日志.

16.3.2　logrotate 配置文件详解 centos7 系统

编辑配置文件

[root@xuegod63 log]# vim /etc/logrotate.conf

说明：(全局参数)

weekly ： 每周执行回滚，或者说每周执行一次日志回滚

rotate： 表示日志切分后历史文件最多保存离现在最近的多少份 [rəʊˈteɪt] 旋转

（rotate 4 保留最近 4 份日志，以前的，第 5,6,7 等等都删掉）

create ： 指定新创建的文件的权限与所属主与所属组

dateext ： 使用日期为后缀的回滚文件 #可以去/var/log 目录下看看

单独配置信息

/var/log/btmp { 指定的日志文件的名字和路径

missingok 如果文件丢失，将不报错

monthly 每月轮换一次

create 0664 root utmp 设置 btmp 这个日志文件的权限，属主，属组

minsize 1M 文件超过 1M 进行回滚（分割），所以大家要知道它不一定每个月都会进行分割，要看这个文件大小来定

rotate 1 日志切分后历史文件最多保存 1 份，不含当前使用的日志

其它参数说明：

monthly: 日志文件将按月轮循。其它可用值为‘daily’，‘weekly’或者‘yearly’。

rotate 5: 一次将存储 5 个归档日志。对于第六个归档，时间最久的归档将被删除。

compress: 在轮循任务完成后，已轮循的归档将使用 gzip 进行压缩。

delaycompress: 总是与 compress 选项一起用，delaycompress 选项指示 logrotate 不要将最近的归档压缩，压缩将在下一次轮循周期进行。这在你或任何软件仍然需要读取最新归档时很有用。

missingok: 在日志轮循期间，任何错误将被忽略，例如“文件无法找到”之类的错误。

notifempty: 如果日志文件为空，轮循不会进行。

create 644 root root: 以指定的权限创建全新的日志文件，同时 logrotate 也会重命名原始日志文件。

postrotate/endscript: 在所有其它指令完成后，postrotate 和 endscript 里面指定的命令将被执行。在这种情况下，rsyslogd 进程将立即再次读取其配置并继续运行。

/var/lib/logrotate/status 中默认记录 logrotate 上次轮换日志文件的时间。

16.3.3 实战-使用 logrotate 进行 ssh 日志分割

定义了 ssh 日志存储在/var/log/sshd 的基础上执行：

[root@xuegod63 ~]# vim /etc/logrotate.d/sshd #创建一个 sshd 配置文件，插入内容：

```
/var/log/sshd.log {
    missingok
    weekly
    create 0600 root root
    minsize 1M
    rotate 3
}
```

那有同学说我不想每周，或每月，我想一分钟分割一次日志，日志当然没有这样分割的，但你可以用计划任务调用这个脚本就行了。

```
crontab -e
    */1 * * * *  logrotate -vf /etc/logrotate.d/sshd
```

[root@xuegod63 ~]#systemctl restart rsyslog

[root@xuegod63 ~]# logrotate -d /etc/logrotate.d/sshd #预演，不实际轮询（切割）

[root@xuegod63 ~]# logrotate -vf /etc/logrotate.d/sshd #强制轮询（切割），也就是说即使轮循条件没有满足，也可以通过加-f 强制让 logrotate 轮循日志文件

-v 显示指令执行过程

-f 强制执行

[root@xuegod63 ~]# ls /var/log/sshd*

/var/log/sshd.log /var/log/sshd.log.1 /var/log/sshd.log.2 /var/log/sshd.log.3

再次查看日志文件大小，已经为 0

[root@xuegod63 ~]# ll -h /var/log/sshd.log

-rw------- 1 root root 0 5 月 22 00:49 /var/log/sshd.log

例 2：实战-使用 logrotate 进行 nginx 日志分割

前提已经搭建好 nginx，大家了解一下，后期讲了 nginx 后你在练习这个

[root@xuegod63 nginx]# vim /etc/logrotate.d/nginx

/usr/local/nginx/logs/*.log { #指定日志文件位置，可用正则匹配

daily #调用频率，有：daily，weekly，monthly 可选

rotate 5 #一次将存储 5 个归档日志。对于第六个归档，时间最久的归档将被删除。

sharedscripts #所有的日志文件都轮转完毕后统一执行一次脚本，

postrotate #执行命令的开始标志

if [-f /usr/local/nginx/logs/nginx.pid]; then #判断 nginx 是否启动

/usr/local/nginx/sbin/nginx -s reload

#让 nginx 重新加载配置文件，生成新的日志文件，如果 nginx 没启动不做操作

fi

endscript #执行命令的结束标志

}

没有切割日志： 日志 150G 了。。。

16.3.4 配置远程日志服务器-实现日志集中的管理

实验拓扑图：

server 端配置

[root@xuegod63 ~]# vim /etc/rsyslog.conf #使用 TCP 协议方式，收集日志

改：19 #$ModLoad imtcp

20 #$InputTCPServerRun 514

为：

19 $ModLoad imtcp

20 $InputTCPServerRun 514

Centos8 把下面 2 行的注释去掉

24 #module(load="imtcp") # needs to be done just once

25 #input(type="imtcp" port="514")

注：使用 UDP 协议 速度快 不保证数据的完整，使用 TCP 协议 可靠.完整

[root@xuegod63 ~]# systemctl restart rsyslog #重新启动 rsyslog

查看服务监听的状态：

[root@xuegod63 ~]# netstat -anlpt| grep 514

tcp 0 0 0.0.0.0:514 0.0.0.0:* LISTEN
45631/rsyslogd

tcp6 0 0 :::514 :::* LISTEN
45631/rsyslogd

服务端验证:

在服务端关闭 selinux 和防火墙

[root@xuegod63 ~]# getenforce

Enforcing

[root@xuegod63 ~]# setenforce 0 #关闭 selinux 功能

[root@xuegod63 ~]#getenforce

Permissive

[root@xuegod63 ~]# systemctl stop firewalld

[root@xuegod63 ~]# systemctl status firewalld

[root@xuegod63 ~]# iptables -F #清空防火墙规则

client 端配置:

登录 xuegod64 客户端

[root@xuegod64 ~]# vim /etc/rsyslog.conf　#在 90 行之后，插入
.　@@192.168.1.63:514　　#写入服务端的 ip 地址

注：　*.* 所有类别和级别的日志；　@@192.168.1.63:514 运端 tcp 协议的日志服务端的 IP 和端口

重启 rsyslog 服务
[root@xuegod64 ~]# systemctl restart rsyslog.service

服务端查看日志：
[root@xuegod63 ~]# tail -f /var/log/messages | grep xuegod64 --color　#动态查看日志

在客户端 xuegod64 进行测试
语法：　logger　要模拟发送的日志
[root@xuegod64 ~]# logger　"aaaaa"

在服务端查看日志
[root@xuegod63 ~]# tail -f /var/log/messages | grep xuegod64 --color
#服务器端到查看消息
May 21 16:32:16 xuegod64 root: aaaaa
注：
总结：服务器使用 udp 协议，客户端使用的配置文件中这一行只能有一个@
.　@192.168.1.64:514
服务器使用 tcp 协议，客户端使用的配置文件中这一行必须有两个@@
.　@@192.168.1.64:514

16.4　实战-配置公司内网服务器每天定时自动开关机

实战场景：为了节约公司开销，需要你设置公司的 svn 版本管理服务器，每天晚上 23:00 开机，每天早上 9:00 自动开机。

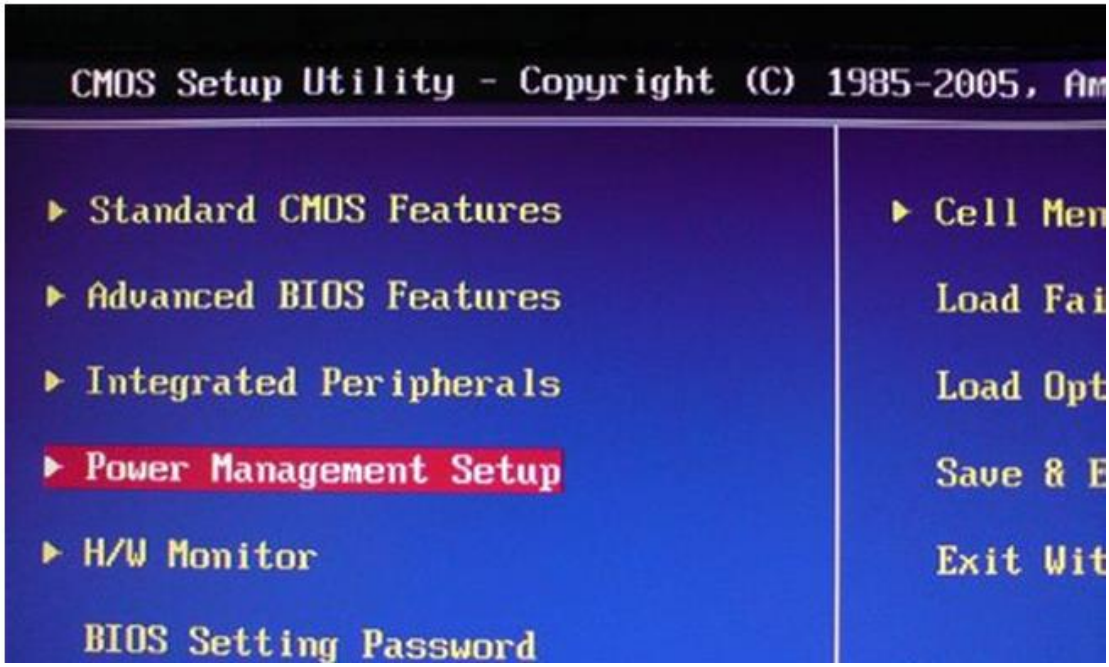
16.4.1　定时关机

[root@xuegod63 ~]# crontab -e　#写入以下内容
0 23 * * *　/usr/sbin/shutdown　-h now

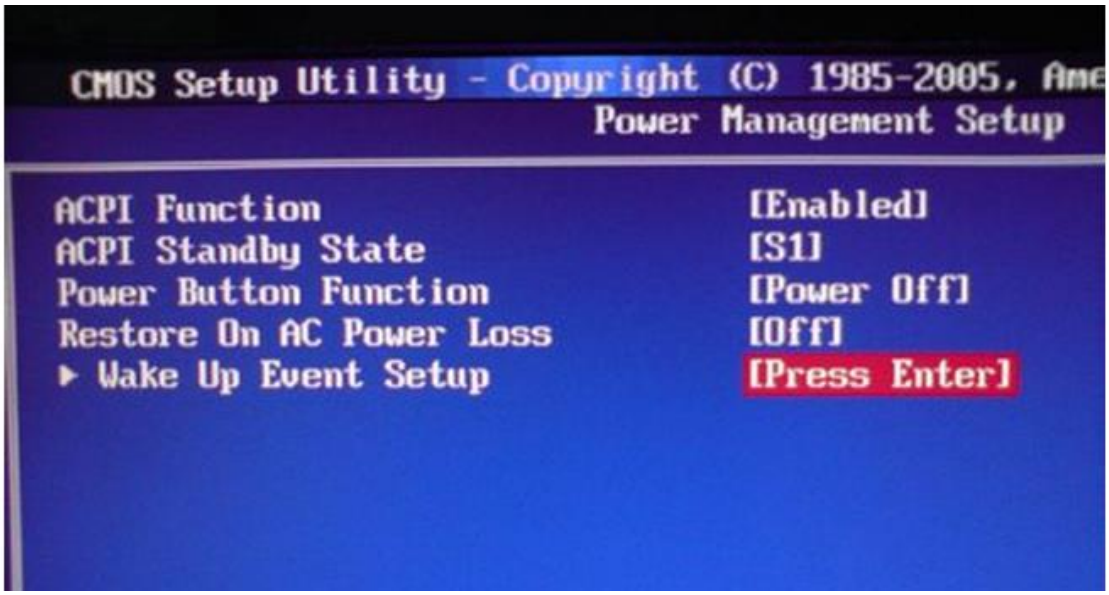
16.4.2　定时开机

这个可以通过设置 bios（位于主板中的最底层控制系统）来实现，**前提是 bios 支持电源管理。**

进入 bios，一般是在开机后出现主板画面是按 Delete 这个键，部分品牌机可能按 F2，进入 bios 设置界面了。然后通过键盘上的箭头选择 Power Management Setup，就进入电源管理设置了。

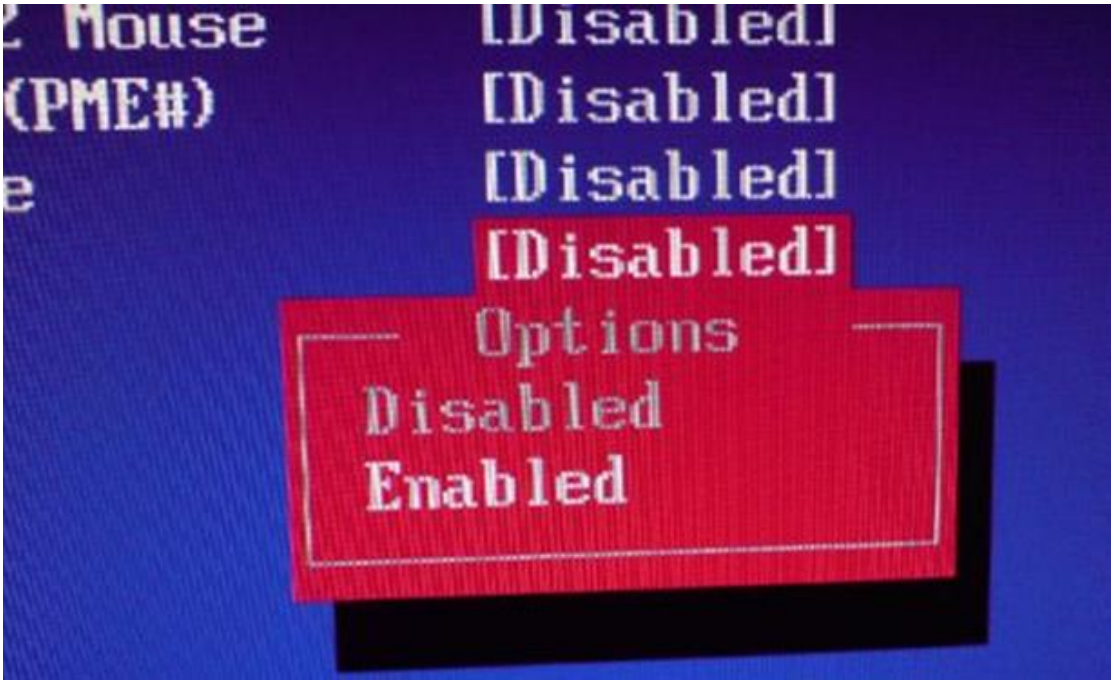


通过回车进入这个设置后，选择 Wake Up Event Setup，回车选择 Press Enter。

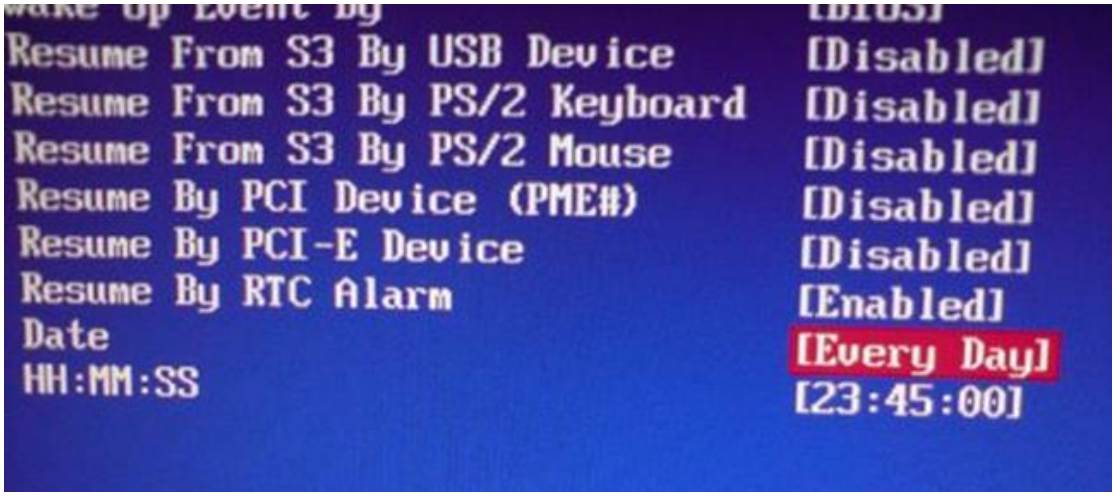


最后，在这个界面内继续找到 Resume By RTC Alarm，回车选择一下。

继续回车选择，将 Disabied 更改为 Enabled，然后继续回车确定。然后再继续设置时间点和日期。



然后选择日期，并且选择你需要电脑每天需要在几点开机，当然，要保证你的主板时间是准确的。



假如你需要每天都定时开机，就选择 Every Day，，你如果想要在每天 6:45 开机，就通过数字键输入 06：15:00，最后，一般按 F10 进行保存，重启电脑后生效。

总结：

16.1 计划任务-at-cron-计划任务使用方法

16.2 日志的种类和记录的方式-自定义 ssh 服务日志类型和存储位置

16.3 实战-日志切割-搭建远程日志收集服务器

16.4 实战-配置公司内网服务器每天定时自动开关机