12 Linux服务管理

12.1 服务简介与分类

1. 服务的分类

- Linux服务
  - RPM包默认的服务
  - 独立的服务(响应速度快，但是占用资源多)
  - 基于xinetd的服务(占用资源少，但是响应速度慢)
  - 源码包安装的服务

2. **Linux系统服务就是用rpm包安装的。所以rpm包安装的服务可以用管理系统服务的方式进行管理。**
3. 启动与自启动

- 服务启动：就是在当前系统中让服务允许，并提供功能。
- 服务自启动：自启动是指让服务在系统开机或重启动之后，随着系统的启动而自动启动服务。

4. 查询已经安装的服务

- 查询RPM包安装的服务
  - chkconfig --list #查看**服务自启动状态**，可以看到所有RPM包安装的服务
  - ps aux #查看进程
  - netstat -tlun
- 查询源码包安装的服务
  - 查看服务安装位置，一般是/usr/local/下

5. RPM安装服务和源码包安装服务的区别：安装位置不同

- 源码包安装在指定位置，一般是/usr/local/
- RPM包安装在默认位置中(删除需要加-e)

12.2 RPM包安装服务的管理

1. **RPM包安装的默认位置**

- /etc/init.d/ 启动脚本位置
- /etc/sysconfig/ 初始化环境配置文件位置
- /etc/ 配置文件位置
- /etc/xinetd.conf xineted配置文件
- /etc/xinetd.d/ 基于xinetd服务的启动脚本
- /var/lib/ 服务产生的数据放在这里
- /var/log/ 日志

2. **独立服务的管理**

- 启动
  - /etc/init.d/独立服务名 start|stop|status|restart
  - service(redhat专有命令) 独立服务名 start|stop|status|restart
- 自启动
  - chkconfig [--level 运行级别] [独立服务名] [on|off]
  - 示例：chkconfig --level 2345 httpd on
  - 修改/etc/rc.d/rc.local文件
  - 使用ntsysv命令管理自启动

3. **基于xinted的服务**

- 安装xinetd与telnet(基于xinetd的服务)
  - yum -y insatll xinetd
  - yum -t intall telnet server
- xinetd服务的启动
  - vi /etc/xinetd.d/telnet
  - 将disable修改成yes
- xinetd服务的自启动(和启动通用)
  - chkconfig telnet on丨off
  - ntsysv

12.3 源码包安装服务的管理

1. **源码包安装服务的启动**

- 使用绝对路径，调用启动脚本来启动。不同的源码包的启用脚本不同。可以查看源码包的安装说明，查看启动脚本的方法。
- /usr/local/apache2/bin/apachectl start|stop

2. **源码包安装服务的自启动**

- vi /etc/rc.d/rc.local
- 加入 /usr/local/apache2/bin/apachectl start

3. **让源码包服务被服务管理命令识别**

- 让源码包的apache服务能被service命令管理启动
  - ln -s /usr/local/apache2/bin/apachectl /etc/init.d/apache
- 让源码包的apache服务能被chkconfig命令管理自启动
  - vi /etc/init.d/apache
  - # chkconfig: 35 86 76
  - #指定httpd脚本可以被chkconfig命令管理。格式是：chkconfig: 运行级别 启动顺序 关闭顺序
  - # description: source package apache
  - #说明，内容随意
  - # chkconfig --add apache
  - #把源码包apache加入chkconfig命令

12.4 服务管理总结

13 Linux系统管理

13.1 进程管理

1. **进程简介**

- 进程是正在执行的一个程序或者命令，每一个进程都是一个运行的实体，都有自己的地址空间，并占用一定的系统资源。(注：ls也会产生进程，只不过你一回车进程就结束了)

2. **进程管理的作用**

- 判断服务器健康状态
- 查看系统中所有进程
- 杀死进程

3. **进程查看**

- ps aux #查看系统中的所有进程，使用BSD操作系统格式
  - USER：该进程是由哪个用户产生的
  - PID：进程的ID号
  - %CPU：该进程占用CPU资源的百分比，占用越高，进程越耗费资源
  - %MEM：该进程占用物理内存的百分比，占用越高，进程越耗费资源
  - VSZ：该进程占用虚拟内存的大小，单位：KB;
  - RSS：该进程占用实际物理内存的大小，单位KB;
  - TTY：该进程是在哪个终端中运行的。其中tty1-tty7代表本地控制台终端，tty1-tty6是本地的字符界面终端，tty7是图形终端。pts/0-255代表虚拟终端(远程终端)。
  - STAT：进程状态。常见有：R(运行)、S(睡眠)、T(停止)、s(包含子进程)、+(位于后台)
  - START：该进程的启动时间
  - TIME：该进程占用CPU的运算时间，注意不是系统时间
  - COMMAND：产生此进程的命令名
- ps -le #查看系统中所有进程，使用Linux标准命令格式

4. **查看系统健康状态**

- top [选项]
- 选项：
  - -d 秒数 #指定top命令每隔几秒更新。默认是3秒
- 在top命令的交互模式中可以执行的命令(shift+)：
  - ？或h #显示交互模式的帮助

    - P #以CPU使用率排序，默认就是此项
    - M #以内存的使用率排序
    - N #以PID排序
    - q #退出top
- 第一行信息为任务队列信息
    - top - 15:59:13 up 36 days, 4:41, 1 user, load average: 0.00, 0.03, 0.05
    - 系统当前时间 系统运行时间 当前登录用户数 系统在之前1、5、15分钟的**平均负载**(小于1时，负载较小，大于1则系统已经超出了负荷)
- 第二行信息为进程信息
    - zombie 僵尸进程。如果不是0，需要手工检查僵尸进程
- 第三行为CPU信息
    - %Cpu(s): **1.0** us, **0.7** sy, **0.0** ni, **98.3** id, **0.0** wa, **0.0** hi, **0.0** si, **0.0** st
    - 用户模式占用的百分比、系统模式占用的百分比、改变过优先级的用户进程占用的百分比、**空闲CPU的百分比**、等待输入输出进程的用户占用的百分比、硬中断请求服务占用的百分比、软中断请求服务占用的百分比、st(steal time)虚拟时间百分比
- 第四行物理内存信息
- 第五行交换分区信息

5. **查看进程树**

- pstree [选项]
- 选项
    - -p 显示进程的PID
    - -u 显示进程的所属用户

6. **终止进程**

- kill -l#查看可用的进程信号 信号代号 信号名称 说明 1 SIGHUP 该信号让进程立即关闭，然后重新读取配置文件之后重启 2 SIGINT 程序终止信号，用于终止前台进程。相当于输出ctrl+C快捷键 8 SIGFPE 在发生致命的算术运算错误时发出，不仅包含浮点运算错误，还包括溢出及除数为0等其他所有的算术的错误。 9 SIGKILL 用来立即结束程序的运行，本信号不能被阻塞、处理和忽略。一般用于**强制终止进程**。 14 SIGALRM 时钟定时信号，计算的是实际的时间或时钟时间，alarm 函数使用该信号 15 SIGTERM **正常结束进程**的信号，kill命令的默认信号。有时候进程已经发生问题，这个信号无法正常终止进程，我们才尝试用SIGKILL信号 18 SIGCONT 该信号可以让暂停的进程恢复执行，本信号不能被阻断。 19 SIGSTOP 该信号可以暂停前台进程，相当与输入ctrl+Z快捷键，本信号不能被阻断
- 示例：
    - kill -9 22372 #强制杀死进程22372
    - kill -1 22354 #重启进程22354

7. **killall命令**

- killall [选项] [信号] 进程名 #按照进程名杀死进程
- 选项
    - -i #交互式，询问是否需要杀死某个进程
    - -I #忽略进程名的大小写

8. **pkill命令**

- pkill [选项] [信号] 进程名 #按照进程名终止进程
- 选项
    - -t 终端号 #按照终端号踢出用户
- 按照终端号踢出用户示例
    - w #使用w命令查询本机已经登录的用户
    - pkill -t -9 pts/1 #强制杀死从pts/1虚拟终端登录的进程

13.2 工作管理

1. **把进程放入后台**

- tar -zcf etc.tar.gz /etc **&**(用这种方法放入后台的进程还在运行)
- top #在top命令执行过程中，按下**ctrl+z快捷键**(用这种方法放入后台的进程停止运行)

2. **查看后台的工作**

- jobs [-l]
- 选项：
    - -l#显示工作的PID
- 注：“+”号代表最近一个放入后台的工作，也是恢复时第一个恢复的工作，“-”号代表倒数第二个放入后台的工作，恢复时

第二个恢复。

3. **将后台暂停的工作恢复到前台执行**

- fg %工作号
- 参数：
  - %工作号：%号可以省略，但是注意工作号和PID的区别

4. **将后台暂停的工作恢复到后台执行**

- bg %工作号
- 注：后台恢复执行的命令，是不能和前台有交互的，否则不能恢复到后台执行。

13.3 系统资源查看

1. **vmstat命令监控系统资源**

- vmstat [刷新延时 刷新次数]
- 示例：vmstat 1 3

2. **dmesg开机时内核检测信息**

- dmesg
- 示例：dmesg | grep CPU #查看CPU信息

3. **free命令查看内存使用状态**

- free [-b|-k|-m|-g]
- 选项：
  - -b #以字节为单位显示
  - -k #以KB为单位显示(默认KB)
  - -m #以MB为单位显示
  - -g #以GB为单位显示

4. **缓冲和缓存的区别**

- 缓存(cache)是用来加速数据从硬盘中读取的。
- 缓冲(buffer)是用来加速数据“写入”硬盘的。

5. **查看CPU详细信息**

- cat /proc/cpuinfo
  - 一旦断电，该文件的内容丢失
- ls cpu

6. **uptime命令**

- uptime #显示系统的启动时间和平均负载，也就是top命令的第一行。w命令也可以看到这个数据。

7. **查看系统与内核相关信息**

- uname [选项]
- 选项：
  - -a #查看系统所有相关信息
  - -r #查看内核版本
  - -s #查看内核名称

8. **判断当前操作系统的位数**

- file /bin/ls

9. **查询当前Linux的发行版本**

- lsb_release -a

10. **列出进程打开或使用的文件信息**
    - lsof [选项]
    - 选项：
    - -c 字符串 #只列出以字符串开头的进程打开的文件

- -u 用户名 #只列出某个用户的进程打开的文件
- -p pid #列出某个PID进程打开的文件

13.4 系统定时任务(重要)

1. **crond服务管理与访问控制**

- service crond restart #启动crond服务
- chkconfig crond on #自启动crond服务

2. **用户的crontab设置**

- crontab [选项]
- 选项：
  - -e #编辑crontab定时任务
  - -l #查询crontab任务
  - -r #删除当前用户所有的crontab任务
- crontab -e #进入crontab编辑界面。会打开vim编辑你的工作。
- * * * * * 要执行的任务 项目 含义 范围 第一个* 一小时当中的第几分钟 0-59 第二个* 一天当中的第几个小时 0-23 第三个* 一个月当中的第几天 1-31 第四个* 一年当中的第几月 1-12 第五个* 一周当中的星期几 0-7(0和7都表示星期天) 特殊符号 含义 * 代表任何时间，比如第一个*就表示一小时中每分钟执行一次的意思。 , 代表不连续的时间。比如"0 8,12,16 * * *" 表示在每天的8点0分、12点0分、16点0分都执行一次命令 - 代表连续的时间范围。比如"0 5 * * 1-6" 表示在周一到周六的5点0分执行命令 */n 代表每隔多久执行一次。比如"*/10 * * * *"代表每隔10分钟就执行一遍命令
- 示例：5 5 * * 2 /sbin/shutdown -r now

14 日志管理

14.1 日志管理简介

1. **rsyslogd的新特点：**

- 基于TCP网络协议传输日志信息
- 更安全的网络传输方式
- 有日志消息的及时分析框架
- 后台数据库
- 配置文件中可以写简单的逻辑判断
- 与syslog配置文件兼容

2. **确定服务启动**

- ps aux|grep rsyslogd #查看服务是否启动
- chkconfig --list|grep rsyslog #查看服务是否自启动

3. **常见日志的作用** 日志文件 说明 /var/log/cron 记录了系统定时任务相关的日志 /var/log/cups 记录打印信息的日志 /var/log/dmesg 记录了系统在开机时内核自检的信息。也可以使用dmesg命令直接查看内核自检信息 /var/log/btmp 记录错误登录的日志。这个文件是二进制文件，不能直接用vi查看，而要用lastb命令查看 /var/log/lastlog 记录系统中所有用户最后一次的登录时间的日志。这个文件是二进制文件，不能直接用vi查看，而要用lastlog命令查看 /var/log/maillog 记录邮件信息 /var/log/messages 记录系统重要信息的日志。如果系统出现问题，首先要检查的就是这个日志文件。 /var/log/secure 记录验证和授权方面的信息，只要涉及账户和密码的程序都会记录。比如：系统登录、ssh登录、su切换用户、sudo授权，以及添加用户和修改用户密码。 /var/log/wtmp 永久记录所有用户的登录信息、注销信息，同时记录系统的启动、重启、关机事件。同样这个文件是二进制文件，不能直接用vi查看，而要用last命令查看 /var/run/utmp 记录当前已经登录的用户的信息。这个文件会随着用户的登录和注销不断变化，只记录当前登录用户的信息。同样这个文件不能直接vi，而要使用w，who，users等命令来查询。
4. 除了系统默认日志外，采用**RPM方式安装**等系统服务也会默认把日志记录在/var/log/目录中(**源码包安装**的服务日志是在源码包指定目录中)。不过这些日志不是由rsyslogd服务来记录和管理的，而是各个服务使用自己的日志管理文档来记录自身日志。

14.2 rsyslogd日志服务

1. **日志文件格式**

- 基本日志格式包含以下四列：
  - 事件产生的时间
  - 发生事件的服务器的主机名
  - 产生事件的服务器或程序名
  - 事件的具体信息

2. **/etc/rsyslog.conf配置文件**

- authpriv.* /var/log/secure
- 服务名称[连接符号]日志等级 日志所在位置
- 认证相关服务.所有日志等级 记录在/var/log/secure文件中 服务名称 说明 auth 安全和认证相关消息(不推荐使用authpriv替代) authpriv 安全和认证相关消息(私有的) cron 系统定时任务cront和at产生的日志 daemon 和各个守护进程相关的日志 ftp ftp守护进程相关的日志 kern 内核产生的日志(不是用户进程产生的) loca10-loca17 为本地使用预留的日志 lpr 打印产生的日志 mail 邮件收发信息 news 与新闻服务器相关的日志 syslog 由syslogd服务产生的日志信息 user 用户等级类别的日志信息 uucp uucp子系统的日志信息，uucp是早期Linux系统进行数据传递的协议，后来也常用在新闻组服务中 链接符号 说明 * 代表所有的日志等级 . 代表只要比后面的等级高的(包含该等级)日志都记录下来。 .= 代表只记录所需等级的日志，其他等级的都不记录。 .! 代表不等于，除了该等级的日志外，其他等级的日志都记录。 日志等级(从低等级到高等级) 说明 debug 一般的调试信息说明 info 基本的通知信息 notice 普通信息，但是有一定的重要性 warning 警告信息，但是还不会影响到服务或系统的运行 err 错误信息，一般达到err等级的信息已经可以影响到服务或系统的运行了 crit 临界状况信息，比err等级还要严重 alert 警告状态信息，比crit还要严重。必须立即采取行动 emerg 疼痛等级信息，系统已经无法使用了

3. **日志记录位置**

- 日志文件的绝对路径，如/var/log/secure
- 系统设备文件，如/dev/lp0(打印机)
- 转发给远程主机，如@192.168.0.210:514(日志服务器)
- 用户名，如root
- 忽略或丢弃日志，如~

14.3 日志轮替

1. **日志的处理**

- 日志切割：按天保存成不同的文件
- 日志轮换：删除旧的日志

2. **日志的命名规则**

- 如果配置文件中拥有dateext参数，那么日志会用日期来作为日志文件的后缀，如secure-20200218。这样的话日志文件名不会重叠，所以也就不需要日志文件的改名，只需要保存指定的日志个数，删除多余的日志文件即可。
- 如果配置文件中没有dattext参数，那么日志文件就需要进行改名了。当第一次进行日志轮替时，当前的secure日志会自动改名为secure.1，然后新建的secure日志，用来保存新的日志。当第二次进行日志轮替时，日志secure.1会自动改名为secure.2，当前的secure日志会自动改名为secure.1，然后也会新建secure日志，用来保存新的日志，以此类推。
- RPM包安装的服务会自动日志轮替，只有源码包安装的服务需要手动进行日志轮替

3. **logrotate配置文件/etc/logrotate.conf** 参数 参数说明 daily 日志的轮替周期为每天 weekly 日志的轮替周期为每周 monthly 日志的轮替周期为每月 rotate 数字 保留的日志文件的个数。0指没有备份 compress 日志轮替时，旧的日志进行压缩 creat mode owner group 建立新日志，同时指定新日志的权限与所有者和所属组。如create 0600 root utmp mail address 当日志轮替时，输出内容通过邮件发送到指定的邮件地址。如wadefrank@163.com missingok 如果日志不存在，则忽略该日志的警告信息 notifempty 如果日志为空文件，则不进行日志轮替 minsize 大小 日志轮替的最小值。也就是日志一定要达到这个最小值才会轮替，否则就算时间达到也不轮替 size 大小 日志只有大于指定大小才进行日志轮替，而不是按照时间轮替。如size 100k dateext 使用日期作为日志轮替文件的后缀。如secure-20130605
4. **把apache日志加入轮替**

```
vi /etc/logrotate.conf

#加入以下的代码
/usr/local/apache2/logs/access_log{
    daily
    create
    rotate 30
}
```

5. **logrotate命令**

- logrotate [选项] 配置文件名
- 选项：
  - 如果此命令没有选项，则会按照配置文件中的条件进行日志轮替
  - -v 显示日志轮替过程。
  - -f 强制进行日志轮替。不管日志轮替的条件是否已经符合，强制配置文件中的所有的日志进行轮替。

15 启动管理

15.1 CentOS 6.3 启动管理

1. **系统运行级别** 级别 含义 0 关机 1 单用户 2 不完全多用户，不含NFS服务 3 完全多用户 4 未分配 5 图形界面 6 重启

- runlevel #查看运行级别命令
- init 运行级别 #改变运行级别命令
- 修改系统默认运行级别
  - vim /etc/inittab
  - Id:3:initdefault: #系统开机后直接进入哪个运行级别

2. **系统启动过程**

15.2 启动引导程序grub

15.3 系统修复模式

16 备份与恢复

16.1 备份概述

1. **Linux系统需要备份的数据**

- /root/目录
- /home/目录
- /var/spool/mail/目录
- /etc/目录
- 其他目录：日志

2. **安装服务的数据**

- apache需要备份的目录
  - 配置文件
  - 网页主目录
  - 日志文件
- mysql需要备份的数据
  - 源码包安装的mysql：/usr/local/mysql/data/
  - RPM包安装的mysql：/var/lib/mysql/

3. **备份策略**

- 完全备份：完全备份就是指把所有需要备份的数据全部备份，当然完全备份可以备份整块硬盘，整个分区或某个具体的目录。
- 增量备份：只备份每天增加的数据(恢复比较麻烦)
- 差异备份：每次备份备份比原始数据多的数据(折中)

16.2 dump和restore命令(增量备份)

1. **dump命令**

- dump [选项] 备份之后的文件名 原文件或目录
- 选项：
  - -level #就是我们说的0-9十个备份级别，示例-0
  - -f文件名 #指定备份之后的文件名
  - -u #备份成功之后，把备份时间记录在/etc/dumpdates文件
  - -v #显示备份过程中更多的输出信息
  - -j #调用bzlib库压缩备份文件，其实就是把备份文件压缩为.bz2格式
  - -W #显示允许被dump的分区的备份等级及备份时间
- 示例：(备份分区) `dump -0uj -f /root/boot.bak.bz2 /boot/ #`备份命令，先执行一次完全备份，并压缩和更新备份时间 `cat /etc/dumpdates #`查看备份时间文件 `cp intall.log /boot/ #`复制日志文件到/boot分区 `dump -luj -f /root/boot.bak1.bz2 /boot/ #`增量备份/boot分区，并压缩 `dump -W #`查询分区的备份时间及备份级别的 `dump -0j -f /root/etc.dump.bz2 /etc/ #`完全备份/etc/目录，只能使用0级别进行完全备份，而不再支持增量备份

2. **restore命令**

- retore [模式选项] [选项]
- 模式选项：restore命令常用的模式有以下四种，这四个模式不能混用。
  - -C #比较备份数据和实际数据的变化
  - -i #进入交互模式，手工选择需要恢复的文件
  - -t #查看模式，用于查看备份文件中拥有哪些数据
  - -r #还原模式，用于数据还原
- 选项：
  - -f#指定备份文件的文件名
- 示例： #比较备份数据和实际数据的变化 `mv /boot/vmlinuz-2.632-279.el6.i686 /boot//vmlinuz-2.632-279.el6.i686.bak` #把/boot目录中内核镜像文件改个名字 `restore -C -f /root/boot.bak.bz2` #restore发现内核镜像文件丢失 #还原模式 #还原boot.bak.bz2分区备份 #先还原完全备份的数据 `mkdir boot.test cd boot.test/ restore -r -f /root/boot.bak.bz2` #解压缩 `restore -r -f /root/boot.bak1.bz2` #恢复增量备份数据