**Linux常见服务：**



**crontab详情查看：https://www.cnblogs.com/intval/p/5763929.html**

**进入编辑器进行编辑：**

**[root@localhost bin]# cd /data/www**

**[root@localhost www]# vim cron.php**

<?php

Echo time().”\r\n”;

**[root@localhost bin]# crontab -e**

\*/1 \* \* \* \* php /data/www/cron.php >> /tmp/cron.log 每隔一分钟执行一次

注释：>>：追加型输入

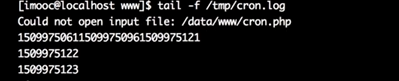
>: 覆盖写入

查询定义的crontab列表

**[root@localhost tmp]# crontab -l**

\*/1 \* \* \* \* php /data/www/cron.php >> /tmp/cron.log

查询执行效果：





此时服务器上的时间和电脑上面的不是一样的，需要使用到日期定时同步

（ntpdate 和 crontab结合使用）

**[root@localhost www]# ntpdate**

-bash: ntpdate: command not found 服务器上没有这个服务，，需要安装

[**root@localhost www]# yum install ntp 安装**

**[root@localhost www]# ntpdate -v**

11 Jan 21:45:17 ntpdate[2166]: ntpdate 4.2.6p5@1.2349-o Wed Apr **12 21:24:06** UTC 2017 (1)

**时间显示明显不对**

**[root@localhost www]# ntpdate cn.pool.ntp.org 同步日期时间**

**[root@localhost www]# ll /etc/localtime 查看系统时间配置（关于时区）**

lrwxrwxrwx. 1 root root 38 Jan 6 06:55 /etc/localtime -> ../usr/share/zoneinfo/America/New\_York ##############包含一个软链

**[root@localhost www]# rm /etc/localtime** ##########删除软链

rm: remove symbolic link ‘/etc/localtime’? y

**[root@localhost www]# date**

Fri Jan 12 02:52:33 UTC 2018 ###########小时还是存在问题

设置上海时区，重新建立软连接

**[root@localhost Asia]# ln -s /usr/share/zoneinfo/Asia/Shanghai /etc/localtime**

**再次查看软链**

**[root@localhost Asia]# ll /etc/localtime**

lrwxrwxrwx. 1 root root 33 Jan 12 10:57 /etc/localtime -> /usr/share/zoneinfo/Asia/Shanghai

**查看时间**

**[root@localhost ~]# date**

**Fri Jan 12 10:59:31 CST 2018**



现在时间就是对的了

做一个定时时间同步任务

**[root@localhost ~]# crontab -e**

\*/30 \* \* \* \* ntpdate cn.pool.ntp.org 每半个小时同步一次

验证定时任务成不成？

**root@localhost ~]# crontab -l**

\*/30 \* \* \* \* ntpdate cn.pool.ntp.org

**[root@localhost ~]# ntpdate cn.pool.ntp.org ##########执行时间同步**

12 Jan 11:06:31 ntpdate[2250]: adjust time server 120.25.115.19 offset -0.006625 sec

**正确执行脚本：成功!!!!!!!!!!!!!!!!!!!!!!!!!!!**

**<<<<<<<<<每个账号下面都有一个crontab,互相不会影响>>>>>>>**

**Linux常见服务--Logrotate&&Supervisor**

**logrotate:**

用于管理各种服务的日志

查看logrotate.d下各种服务的日志管理：

**[root@localhost ~]# cd /etc/logrotate.d**

**[root@localhost logrotate.d]# ls**

bootlog httpd nginx wpa\_supplicant

chrony mysql syslog yum

**[root@localhost logrotate.d]# vim nginx**

/var/log/nginx/\*.log {

daily ####按照每天进行切割

missingok ###

rotate 52 ###最多保留52份log文件

compress ###压缩

delaycompress ###延迟压缩

notifempty

create 640 nginx adm ###创建640权限的文件；可读写

sharedscripts

postrotate

if [ -f /var/run/nginx.pid ]; then

kill -USR1 `cat /var/run/nginx.pid`

1,1 Top

####执行完之后的脚本运行情况

查看nginx服务下的日志：

**[root@localhost logrotate.d]# ls /var/log/nginx/\*.log ###使用通配符**

/var/log/nginx/access.log /var/log/nginx/error.log

[root@localhost logrotate.d]# cd /var/log/nginx

查看nginx的切割日志：

**[root@localhost logrotate.d]# cd /var/log/nginx**

**[root@localhost nginx]# ls**

access.log error.log

access.log-20180110.gz error.log-20180110.gz ######以前的日志压缩，并且以日期的命名方式进行命名。这种比较好

access.log-20180111.gz error.log-20180112

access.log-20180112

应用：更改服务的日志切割：

EG:

**[root@localhost logrotate.d]# vim nginx**

增加某服务（xxxx）的日志切割

**[root@localhost logrotate.d]# vim xxxx**

Logrotate详细情况了解：

**<https://www.cnblogs.com/kevingrace/p/6307298.html>**

**Supervisor：**

supervisor就是用Python开发的一套通用的进程管理程序，能将一个普通的命令行进程变为后台daemon，并监控进程状态，异常退出时能自动重启。

1. 安装supervisor

安装pip

1. wget https://bootstrap.pypa.io/get-pip.py
2. python get-pip.py

**[root@localhost ~]# pip install supervisor**

2.生成配置文件

[root@localhost ~]# mkdir /etc/supervisor

[root@localhost ~]# echo\_supervisord\_conf 自动生成配置文件

[root@localhost tmp]# echo\_supervisord\_conf > /etc/supervisor/supervisor.conf

将生成的配置文件写入/etc/supervisor/supervisor.conf

3.**使用supervisor控制redis随着supervisor的启动而启动**

修改文件

**[root@localhost ~]# vim /etc/supervisor/supervisor.conf**

[include]

files = /etc/supervisor/conf.d/\*.ini ;

**[root@localhost ~]# cd /etc/supervisor/**

**[root@localhost supervisor]# mkdir conf.d**

**[root@localhost supervisor]# cd conf.d/**

**[root@localhost conf.d]# touch redis.ini**

**查看redi-server在哪里**

**[root@localhost conf.d]# which redis-server**

/usr/local/bin/redis-server

编辑控制redis

**[root@localhost conf.d]# vim redis.ini**

[program:redis]

command=/usr/local/bin/redis-server

outostart=true

outorestart=true

starsecs=3

启动supervisor服务

**[root@localhost conf.d]# supervisord -c /etc/supervisor/supervisor.conf**

**[root@localhost conf.d]# ps -ef | grep sup**

root 3591 1 0 14:18 ? 00:00:00 /usr/bin/python /usr/bin/supervisord -c /etc/supervisor/supervisor.conf

root 3597 1158 0 14:18 pts/0 00:00:00 grep --color=auto sup

**服务成功启动！！！！！**

**进入supervisor的命令端：**

**[root@localhost conf.d]# supervisorctl**

http://localhost:9001 refused connection

supervisor>

发现服务被拒绝:

查看redis是否已经启动

**[root@localhost conf.d]# supervisorctl -c /etc/supervisor/supervisor.conf**

redis RUNNING pid 3592, uptime 0:05:43

Redis已经成功启动！！！！！

**解决上述绿色标记问题（supervisor.conf换成supervisord.conf(系统默认搜索)）**

**[root@localhost supervisor]# pwd**

/etc/supervisor

**[root@localhost supervisor]# cp supervisor.conf supervisord.conf**

**[root@localhost supervisor]# ps -ef | grep sup**

root 3591 1 0 14:18 ? 00:00:00 /usr/bin/python /usr/bin/supervisord -c /etc/supervisor/supervisor.conf

root 3634 1158 0 14:30 pts/0 00:00:00 grep --color=auto sup

**[root@localhost supervisor]# ps -ef | grep sup**

root 3591 1 0 14:18 ? 00:00:00 /usr/bin/python /usr/bin/supervisord -c /etc/supervisor/supervisor.conf

root 3636 1158 0 14:30 pts/0 00:00:00 grep --color=auto sup

**[root@localhost supervisor]# kill 3591**

**[root@localhost supervisor]# supervisord -c /etc/supervisor/supervisord.conf**

**[root@localhost supervisor]# ps -ef | grep sup**

root 3640 1 0 14:32 ? 00:00:00 /usr/bin/python

/usr/bin/supervisord -c /etc/supervisor/supervisord.conf

root 3646 1158 0 14:33 pts/0 00:00:00 grep --color=auto sup

**[root@localhost supervisor]# supervisorctl**

redis RUNNING pid 3641, uptime 0:01:07

supervisor>

**现在直接输入：[root@localhost supervisor]# supervisorctl就可以显示supervisor控制的服务启动情况；并且进入supervisor的命令端：**

**[root@localhost supervisor]# supervisorctl**

redis RUNNING pid 3641, uptime 0:01:07

supervisor> status ##############查看状态

redis RUNNING pid 3641, uptime 0:07:12

supervisor> stop redis ############停止redis

redis: stopped

supervisor> start redis ############启动redis

redis: started

supervisor> exit