参考文档：https://www.cnblogs.com/kunpengit/p/4045343.html

定时任务顾名思义就是在某一时间点自动进行任务操作。
在做Pgsql的备份利用crontab进行定时操作， 使用起来比较方便。故分享
具体的定时编辑命令：crontab -e
首先从crontab的文件分析使用策略，root用户下，在/etc下有一个文件crontab，其内容如下
[root@myzk ~]# cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# 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
该 文件下的前四行内容为crontab的环境变量，SHELL变量的值指定系统使用的SHELL环境（该样例为bash shell），PATH变量定义了执行命令的路径。Cron的输出以电子邮件的形式发给MAILTO变量定义的用户名。如果MAILTO变量定义为空字符 串（MAILTO=""），电子邮件不会被发送。执行命令或脚本时HOME变量可用来设置基目录。
注：以上系统会默认可以不用修改任何!

在root用户下，可以直接vim /etc/crontab文件进行脚本的添加定时任务脚本，而在其他普通用户下可以通过crontab –e 进行脚本的添加

编辑完成，可以用crontab –l进行查看脚本信息

定时命令脚本解析：

从表格中可以看出脚本格式如下：

minute hour day month week user-name command

minute---分钟 （0-59）

hour------小时（0-23）

day--------日期（1-31）

month---月份（1-12）

week----星期（0-6） //0代表星期天

除了数字还有几个个特殊的符号就是"*"、"/"和"-"、","，*代表所有的取值范围内的数字，"/"代表每的意思，"*/5"表示每5个单位，"-"代表从某个数字到某个数字，","分开几个离散的数字。以下举几个例子说明问题：

以具体例子进行分析定时脚本：

将date命令和crontab命令结合使用：

*/1 * * * * touch /home/shiyanlou/$(date "+现在时间是\%Y\%m\%d\%H\%M\%S")

pgslq.sh为需要执行的脚本，内容为需要进行的备份操作或者其他任务脚本

1 * * * * /home/postgres/pgsql.sh

表示的是每小时的第一分钟执行该脚本

2 3 * * * /home/postgres/pgsql.sh

表示每天的3点零2分执行该脚本

1 1 * * 0 / home/postgres/pgsql.sh

表示的是每周的1点1分进行脚本的执行

1 1 1 * * / home/postgres/pgsql.sh

表示的是每月的1点1分进行脚本的执行

比较容易犯的错误是通常会把每小时的第一分钟按做每分钟执行一次，这点要注意两者的区别：

1 * * * * /home/postgres/pgsql.sh

表示的是每小时的第一分钟执行该脚本

*/1 * * * * /home/postgres/pgsql.sh

表示的是每一分钟执行该脚本

因此这里要记住”/”这个符号带来的区别

“-”的用法：

0 10 * * 1-3 / home/postgres/pgsql.sh
表示的是每个周一到周三的早上10点执行该脚本
0 10 * * 1、3、5 / home/postgres/pgsql.sh
表示的是每周的周一、周三、周五的早上10点执行该脚本
这些大致就是定时任务的几乎常见的可能性定时脚本。
在表格中看到user-name这个表示的用户该脚本所在的用户，一般情况下在做项目不可能会有root用户进行编写，所以我们可能直接在普通用户下进行定时脚本的编写，直接执行crontab –e，编写定时任务。
每 次编辑完某个用户的cron设置后，cron自动在/var/spool/cron下生成一个与此用户同名的文件，此用户的cron信息都记录在这个文件 中，这个文件是不可以直接编辑的，只可以用crontab -e 来编辑。cron启动后每过一分钟读一次这个文件，检查是否要执行里面的命令。因此此文件修改后不需要重新启动cron服务。