ORACLE Certified Professional

ORACLE E-Business Suite 11*i* Certified Professional

Author – *A.Kishore/Sachin*
http://appsdba.info

Crontab : Everything You Want To Know.

Linux has a great program called cron. It allows tasks to be automatically run in the background at regular intervals. We could also use it to automatically create backups, synchronize files, schedule updates, and much more.Lets Know about crontab.

Crontab

The crontab (cron derives from chronos, Greek for time; tab stands for table) command, available in Unix and Unix-like operating systems, is used to schedule commands to be executed periodically. To see what crontabs are currently running on our system, we can open a terminal and run:

sudo crontab –l
The above command displays the entries in crontab if any.

To edit the list of cronjobs you can run:

sudo crontab –e

This will open a default editor (could be vi or pico, if we want to change the default editor) to let us manipulate the crontab. If you save and exit the editor, all our cronjobs are saved into crontab. Cronjobs are written in the following format:

* * * * * /bin/myscript.sh

Scheduling explanation:

As we can see there are 5 stars. The stars represent different date parts in the following order:

1. minute (from 0 to 59)
2. hour (from 0 to 23)
3. day of month (from 1 to 31)
4. month (from 1 to 12)
5. day of week (from 0 to 6) (0=Sunday)

- **Execute something every minute:**

If we want to execute the file "myscript.sh"every minute, then we need to do the following crontab entry.

* * * * * /bin/myscript.sh

Please Note: If we leave the star, or asterisk, it means every. There are all asterisks. So this means execute /bin/myscript.sh:

1.every minute
2.every hour
3.every day of the month
4.every month
5.every day in the week.

So the above script will be executed every minute without exception.

- **Execute every Friday 2AM**

So if we want to schedule the script to run at 2AM every Friday, we would need to do the following cronjob entry:

0 2 * * 5 /bin/myscript.sh

So, the script will be executed when the system clock hits the following:

1.minute: 0
2.hour: 2
3.day: * (every day of month)
4.month: * (every month)
5.weekday: 5 (=Friday)

- **Execute on workdays 1AM**

If we want to schedule the script to Monday till Friday at 2 AM, we would need the following cronjob entry:

0 2 * * 1-5 /bin/myscript.sh

The script will be executed when the system clock hits the following:

1.minute: 0
2.hour: 2
3.day: * (every day of month)
4.month: * (every month)
5.weekday: 1-5 (=Monday till Friday)

- **Execute 10 past after every hour on the first of every month**

If we need to execute 10 pasts after every hour on the first of every month, then we would need the following cronjob entry:

10 * 1 * * /bin/myscript.sh

- **Execute something every 10 minutes:**

0,10,20,30,40,50 * * * * /bin/myscript.sh

- **Storing the crontab output:**

By default cron saves the output of /bin/myscript.sh in the user's mailbox (root in this case). But it's better if the output is saved in a separate logfile. Lets check it as shown below.

*/10 * * * * /bin/myscript.sh 2>&1 >> /var/log/myscript_output.log

Explanation
Linux can report on different levels. There's standard output (STDOUT) and standard errors (STDERR). STDOUT is marked 1, STDERR is marked 2. So the following statement tells Linux to store STDERR in STDOUT as well, creating one data stream for messages & errors:
2>&1
Now that we have 1 output stream, we can pour it into a file. Where > will overwrite the file, >> will append to the file. In this case we'd like to append:
>> /var/log/myscript_output.log

- **Mailing the crontab output:**

By default cron saves the output in the user's mailbox (root in this case) on the local system. But we can also configure crontab to forward all output to a real email address by starting our crontab with the following line:
MAILTO=yourname@yourdomain.com

- **Mailing the crontab output of just one cronjob**

If we receive only one cronjob's output in our mail, make sure this package is installed:
aptitude install mailx

And change the cronjob like this:

*/10 * * * * /bin/myscript.sh 2>&1 | mail -s "Cronjob ouput" yourname@yourdomain.com

- **Trashing the crontab output**

To remove crontab out ,we need to do the following:

*/10 * * * * /bin/myscript.sh 2>&1 > /dev/null

Just pipe all the output to the null device. On Unix-like operating systems, /dev/null is a special file that discards all data written to it.