Cron and Crontab

Cron is a Linux utility that schedules tasks to run automatically at specified intervals. The tasks are defined in a crontab file, which is a simple text file containing a list of commands meant to be run at specified times. Each line of the file represents a single cron job and follows a particular syntax.

For example, you could set a cron job to automate repetitive tasks such as **backing up databases** or data, **updating the system** with the latest security patches, **checking the disk space usage** , **sending emails**, and so on.

The cron jobs can be scheduled to run by a minute, hour, day of the month, month, day of the week, or any combination of these.

Crontab Syntax and Operators

Each line in the user crontab file contains six fields separated by a space followed by the command to be run.

```
* * * * * command(s)
- - - - -
| | | | |
| | | | ----- Day of week (0 - 7) (Sunday=0 or 7)
| | | ------- Month (1 - 12)
| | --------- Day of month (1 - 31)
| ----------- Hour (0 - 23)
------------- Minute (0 - 59)
```

The first five fields may contain one or more values, separated by a comma or a range of values separated by a hyphen.

- ✓ * - The asterisk operator means any value or always. If you have the asterisk symbol in the Hour field, it means the task will be performed each hour.

- ✓ , - The comma operator allows you to specify a list of values for repetition. For example, if you have 1,3,5 in the Hour field, the task will run at 1 am, 3 am and 5 am.
- ✓ - - The hyphen operator allows you to specify a range of values. If you have 1-5 in the Day of week field, the task will run every weekday (From Monday to Friday).
- ✓ / - The slash operator allows you to specify values that will be repeated over a certain interval between them. For example, if you have */4 in the Hour field, it means the action will be performed every four hours. It is same as specifying 0,4,8,12,16,20. Instead of asterisk before the slash operator, you can also use a range of values, 1-30/10 means the same as 1,11,21.

- ✓ @yearly (or @annually) - Run the specified task once a year at midnight (12:00 am) of the 1st of January. Equivalent to 0 0 1 1 *.
- ✓ @monthly - Run the specified task once a month at midnight on the first day of the month. Equivalent to 0 0 1 * *.
- ✓ @weekly - Run the specified task once a week at midnight on Sunday. Equivalent to 0 0 * * 0.
- ✓ @daily - Run the specified task once a day at midnight. Equivalent to 0 0 * * *.
- ✓ @hourly - Run the specified task once an hour at the beginning of the hour. Equivalent to 0 * * * *.

Linux Crontab Command

The crontab command allows you to install, view , or open a crontab file for editing:

- ✓ **crontab –e** Edit crontab file, or create one if it doesn't already exist.
- ✓ **crontab –l** Display crontab file contents.
- ✓ **crontab –r** Remove your current crontab file.
- ✓ **crontab –I** Remove your current crontab file with a prompt before removal.
- ✓ **crontab -u <username>** Edit other user crontab file. This option requires system administrator privileges.

Cron Jobs Examples

1. **Run a command at 15:00 on every day from Monday through Friday:**

   0 15 * * 1-5 command

2. **Run a script every 5 minutes and redirected the standard output to dev null, only the standard error will be sent to the specified e-mail address:**

   MAILTO=email@example.com

   */5 * * * * /path/to/script.sh > /dev/null

3. **Run two commands every Monday at 3 PM (use the operator && between the commands):**

   0 15 * * Mon command1 && command2

4. **Run a PHP script every 2 minutes and write the output to a file :**

   */2 * * * * /usr/bin/php /path/to/script.php >> /var/log/script.log

5. **Run the a script at 9:15pm, on the 1st and 15th of every month:**

   15 9 1,15 * * /path/to/script.sh

6. **Set custom HOME, PATH, SHELL and MAILTO variables and run a command every minute.**

   HOME=/opt

   PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

   SHELL=/usr/bin/zsh

   MAILTO=email@example.com

   */1 * * * * command