

## JOB AUTOMATION

Automation with cron and at:

With cronjob we can schedule a job without human interaction.

In any operating system, it is possible to create jobs that you want to reoccur. This process, known as job scheduling, is usually done based on user-defined jobs. For Red Hat or any other Linux, this process is handled by the cron service or a daemon called crond, which can be used to schedule tasks (also called jobs). By default, Red Hat comes with a set of predefined jobs that occur on the system (hourly, daily, weekly, monthly, and with arbitrary periodicity). As an administrator, however, you can define your own jobs and allow your users to create them as well.

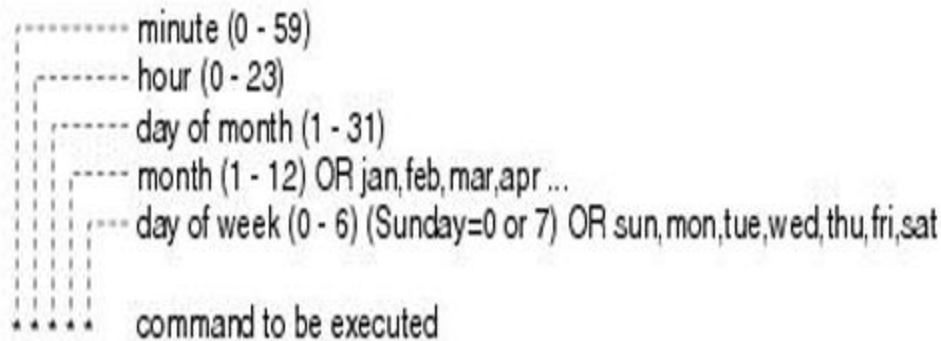
The importance of the job scheduling is that the critical tasks like taking backups, which the clients usually wants to be taken in nights, can easily be performed without the intervention of the administrator by scheduling a cron job. If the cron job is scheduled carefully than the backup will be taken at any given time of the client and there will be no need for the administrator to remain back at nights to take the backup.

Important Files related to cron and at:

- ➔ **/etc/crontab** is the file which stores all scheduled jobs
- ➔ **/etc/cron.deny** is the file used to restrict the users from using cron jobs.
- ➔ **/etc/cron.allow** is used to allow only users whose names are mentioned in this file to use cron jobs. (this file does not exist by default)
- ➔ **/etc/at.deny** same as cron.deny for restricting at jobs
- ➔ **/etc/at.allow** same as cron.allow for allowing user to use at jobs.

Crontab format:

- ➔ To assign a job in the Crontab file the format used is the following



Options	Explanation
*	Is treated as a wild card. Meaning any possible value.
* / 5	Is treated as ever 5 minutes, hours, days, or months. Replacing the 5 with another numerical value will change this option.
2,4,6	Treated as an OR, so if placed in the hours, this could mean at 2, 4, or 6 o'clock.
9-17	Treats for any value between 9 and 17. So if placed in day of month this would be days 9 through 17. Or if put in hours it would be between 9 and 5.

### Crontab Commands

Command	Explanation
<b>crontab -e</b>	Edit your crontab file, or create one if it doesn't already exist.
<b>crontab -l</b>	Display your crontab file.
<b>crontab -r</b>	Remove your crontab file.
<b>crontab -u</b>	If combined with <b>-e</b> , edit a particular user's Crontab file and if combined with <b>-l</b> , display a particular user's crontab file. If combined with <b>-r</b> , deletes a particular user's Crontab file

### CRON JOBS:

To check the assigned cron jobs of currently logged in user

To check the cron jobs the command is `#crontab -l`

root@master-server:~

```
[root@master-server ~]# crontab -l  
no crontab for root  
[root@master-server ~]# whoami  
root  
[root@master-server ~]#
```

To check the cron jobs of a particular user

To check a user's cron jobs, the syntax is `#crontab -l -u <username>`

`#crontab -l -u yallareddy`

root@master-server:~

```
[root@master-server ~]# crontab -l -u yallareddy  
no crontab for yallareddy  
[root@master-server ~]#
```

Setting a job to display the current date for every minute on present console

- ➔ To set the above job the steps are
- ➔ Create a file to store the output of the cronjob.
- ➔ Touch `/root/cronjoboutput`
- ➔ Schedule the task as shown below
- ➔ `#crontab -e` and enter the field as shown below and save it as in VI editor
- ➔ Put this entry's and save the file. `* * * * * date > /root/cronjoboutput`
- ➔ Note: > this means only one output will store. If you want to append then give >> the output will append everytime.
- ➔ Then read the file `# cat cronjoboutput` → you can see output of every minute as per cron job schedule.

root@master-server:~

```
[root@master-server ~]# cat cronjoboutput  
Thu May 8 12:58:01 IST 2025  
[root@master-server ~]#
```

Note: if you want to append the output just give >> like below in `crontab -e`

```
root@master-server:~  
[root@master-server ~]# crontab -l  
* * * * * date >> /root/cronjoboutput  
[root@master-server ~]#
```

Like this the data will be append

```
root@master-server:~  
[root@master-server ~]# cat cronjoboutput  
Thu May 8 12:59:02 IST 2025  
Thu May 8 13:00:01 IST 2025  
[root@master-server ~]#
```

Note: make sure crond service must be run.

```
root@master-server:~  
[root@master-server ~]# service crond status  
crond (pid 1648) is running...  
[root@master-server ~]#
```

Note: Note: you can use 0 or 7 for Sunday. Check whether it got created or not on scheduled day, if it created you can see the directory otherwise a error mail will be generated to your mail.

Note: in google you can check crontab generater url you will select options easily.

<https://crontab-generator.org/>

AT JOBS:

- ➔ “at” is used to schedule the job for a particular time or interval, in other words it is used only for one time or only for one interval.
- ➔ At jobs cannot be reused.