

How to Configure Crontab in Linux?

written by sysadmin | 1 November 2025

As a sysadmin, Crontab is an important tool for running scripts that you want to run at a certain time.

Problem

How to configure crontab in Linux?

Solution

Cron is a command you can use in the shell to set up a task, like a command or a script, to run automatically at certain times, dates, or intervals. It was made by AT&T Bell Laboratories and first came out in May 1975.

A. Format crontab

Cron works based on a crontab file, which tells it what tasks to run and when. Crontab has 2 sections: a time section that has 5 items, where each item has a different parameter, and the command section to be executed. For more details, see the image below:

🎯 A crontab file has five fields for specifying:

```
* * * * * command to be executed
- - - - -
| | | | |
| | | | +----- **DAY OF WEEK** (0-6) (Sunday=0)
| | | +----- **MONTH** (1-12)
| | +----- **DAY OF MONTH** (1-31)
| +----- **HOUR** (0-23)
+--- **MINUTE** (0-59)
```

The syntax of the crontab file (Credit to blog.marcotorres.pe)

B. Crontab commands

To display the contents of the crontab, use the command below:

```
crontab -l
```

To display the crontab for a user, for example, john, use the command below:

```
crontab -u john -l
```

To create or modify a crontab file, use the command below:

```
crontab -e
```

To delete the crontab file, use the command below:

```
crontab -r
```

C. Crontab examples

Here are examples of Crontab to run time.sh file that contains as below, and don't forget to permit (**chmod +x**) so that the file can be run:

```
#!/bin/bash
#
time=`date +"%Y%m%d-%H:%M:%S"`
echo $time >> time.txt
```

1. Once a week

If you want to run a file once a week, you can use the crontab configuration as below:

```
@weekly          cd /home/sysadmin;./time.sh
```

2. Every time you reboot

Use the crontab configuration below if you want to run a file every time the server reboot is completed:

```
@reboot          cd /home/sysadmin;./time.sh
```

3. Every 5 minutes from 1 to 7

Use the crontab configuration below if you want to run a file every 5 minutes from 1 to 7 (i.e., 01:00, 01:05, 01:10, up until 07:55):

```
*/5 * * * *      cd /home/sysadmin;./time.sh
```

4. Every 10:30 on 1,10,20,30

If you run a file every 10:30 on 1,10,20,30, use the crontab configuration as below:

```
30 10 1,10,20,30 * *      cd /home/sysadmin;./time.sh
```

5. Every first Monday of every month, at 7 a.m.

Use the crontab configuration below if you want to run a file every first Monday of every month, at 7 a.m:

```
0 7 1-7 * 1      cd /home/sysadmin;./time.sh
```

6. Every 15 minutes after rebooting

If you run a file every 15 minutes after rebooting, use the crontab configuration as below:

```
@reboot sleep 900 &&    cd /home/sysadmin;./waktu.sh
```

7. Every last Saturday of every month

If you want to run a file every last Saturday of every month, then you have to create a script file first example, last_saturday.sh, as below:

```
cat sabtu.sh
#!/bin/bash
```

```
TODAY=$(date +%Y-%m-%d)
NEXT_SATURDAY_MONTH=$(date -d "next Saturday" +%m)
CURRENT_MONTH=$(date +%m)

# If the next Saturday is in the next month,
# it means that this Saturday is the last Saturday of the month
if [ "$NEXT_SATURDAY_MONTH" != "$CURRENT_MONTH" ]; then
    echo "$(date '+%Y-%m-%d %H:%M:%S') - Running last Saturday of month job"
    >> /home/sysadmin/last_saturday.log
    /home/sysadmin/time.sh
else
    echo "$(date '+%Y-%m-%d %H:%M:%S') - Skipped (not last Saturday)" >>
/home/sysadmin/last_saturday.log
fi
```

And in the crontab, config like below:

```
0 0 * * 6        cd /home/sysadmin;./last_saturday.sh
```

This script will only work on Saturdays in each month, and if there is a Saturday in the following month, then this script will not run. To see the log for this script, go to /home/sysadmin/last_saturday.log, and here is a sample of the log:

```
sysadmin@ubuntu2404:~$ cat last_saturday.log
2025-11-22 00:00:10 - Skipped (not last Saturday)
2025-11-29 00:00:01 - Running last Saturday of month job
2025-12-27 00:00:01 - Running last Saturday of month job
2025-10-04 00:00:08 - Skipped (not last Saturday)
2025-05-24 00:00:01 - Skipped (not last Saturday)
2025-05-31 00:00:23 - Running last Saturday of month job
sysadmin@ubuntu2404:~$
```



last_saturday.log

Note

For crontab to run properly, use absolute paths for files and commands. And don't forget to make sure the script can be executed and preferably in the script file, to include logs so that it can be traced if there are errors. To see the log in Linux whether crontab is running or not on Ubuntu/Debian, you can use the command below:

```
sudo grep CRON /var/log/syslog
```

Use the command below if you are using RHEL/RockyLinux/AlmaLinux:

```
sudo grep CRON /var/log/cron
```

If you can't find the cron log in the file, then open the file **/etc/rsyslog.d/50-default.conf** and search for the word cron. After that, remove the comment mark **#** behind the cron word. Then restart the service using the command:

```
sudo systemctl restart rsyslog
```

References

en.wikipedia.org
crontab.guru
codepolitan.com
askubuntu.com
medium.com