cron cheat sheet

Read and write crontab schedules: the fields, special characters, common schedules and gotchas.

The five fields

```
┌──────────── minute        0–59
│ ┌────────── hour          0–23
│ │ ┌──────── day of month  1–31
│ │ │ ┌────── month         1–12 or JAN–DEC
│ │ │ │ ┌──── day of week   0–7 or SUN–SAT
│ │ │ │ │
* * * * *  /path/to/command
```

0 and 7 are both Sunday. In `/etc/crontab` and `/etc/cron.d/`, the user to run as goes between the schedule and the command. Spring and Quartz put a sixth field, seconds, in front; Quartz numbers weekdays 1–7 from Sunday.

Special characters

`*`	Every value
`5`	One value
`1–5`	A range: 1 to 5
`1,15`	A list
`*/15`	A step: 0, 15, 30, 45
`9–17/2`	A step within a range: 9, 11 … 17
`MON, JAN`	Names. In ranges, use numbers (1-5): the Debian and macOS manuals don't allow MON-FRI
`?`	Any day (Spring, Quartz; day fields only)

Nicknames

`@yearly`	`0 0 1 1 *`
`@annually`	`0 0 1 1 *`
`@monthly`	`0 0 1 * *`
`@weekly`	`0 0 * * 0`
`@daily`	`0 0 * * *`
`@midnight`	`0 0 * * *`
`@hourly`	`0 * * * *`
`@reboot`	Once, when cron starts (at boot)

The crontab command

`crontab -e`	Edit your crontab; saving installs it
`crontab -l`	Show it
`crontab -l > crontab.bak`	Back it up
`crontab -r`	Delete all of it, without asking

Common schedules

`* * * * *`	Every minute
`*/5 * * * *`	Every 5 minutes
`*/10 * * * *`	Every 10 minutes
`*/15 * * * *`	Every 15 minutes
`*/30 * * * *`	Every 30 minutes
`0 * * * *`	Every hour
`0 */2 * * *`	Every 2 hours
`0 0,12 * * *`	Twice a day
`0 0 * * *`	Every day at midnight
`0 9 * * 1–5`	Every weekday at 9:00
`0 9 * * 1`	Every Monday at 9:00
`0 0 1 * *`	On the 1st of every month
`0 0 1 */3 *`	Every quarter
`0 0 1 1 *`	Once a year

Two tricks

Every 30 seconds

```
* * * * * /path/to/job.sh
* * * * * sleep 30; /path/to/job.sh
```

Last day of the month (one line; GNU date)

```
0 23 28-31 * * [ "$(date -d tomorrow +\%d)" = 01 ] &&
/path/to/job.sh
```

Gotchas

- **Both day fields set?** Cron runs when *either* matches: `0 0 13 * 5` is every 13th plus every Friday. A field starting with `*` (like `*/2`) doesn't count as set.
- **Steps start over** each hour, day or month: `*/7` in the minutes runs at :56, then :00.
- **`%` is a line break** in a crontab command. Write `\%`, as in `date +\%F`.
- **Short PATH, no `~/.bashrc`.** Use full paths, or a full `PATH=` line at the top.
- **The server's time zone** applies (cronie: `CRON_TZ=`). Avoid 01:00–03:00: daylight saving can skip or repeat a run.
- **Output:** `>> /path/job.log 2>&1`, or cron mails it; `MAILTO=""` turns that off.
- **Did it run at all?** `journalctl -u cron` (Debian, Ubuntu) or `journalctl -u crond` (Fedora, RHEL).
- **End the file with a newline,** or cron may treat the last line as broken.