Cron Job Cheatsheet

A comprehensive cheat sheet for Cron jobs, covering syntax, operators, special strings, and practical examples. Learn how to schedule tasks effectively with this handy reference.

Cron Syntax and Structure

Basic Cron Syntax

Cron entries follow a specific format to define when and how a command should be executed.

`* * * * * command`

Each asterisk represents a time unit, in the following order:

`minute hour day month weekday`

Time Unit Values

`Minute`	Values range from 0 to 59.
`Hour`	Values range from 0 to 23.
`Day`	Values range from 1 to 31.
`Month`	Values range from 1 to 12 (or names like `Jan`, `Feb`, etc.).
`Weekday`	Values range from 0 to 6 (0 is Sunday, or names like `Sun`, `Mon`, etc.).

Understanding the Fields

Each field in a cron entry specifies a time unit. Understanding these fields is crucial for scheduling tasks accurately.

Example: `30 2 * * 1-5` - This will run a command at 2:30 AM on every weekday (Monday to Friday).

Cron Operators and Special Characters

Cron Operators

`*` (Asterisk)	Represents 'all values'. For example, `*` in the month field means every month.
`,` (Comma)	Specifies a list of values. Example: `1,15` in the day field means the 1st and 15th of the month.
`-` (Hyphen)	Defines a range of values. Example: `1-5` in the weekday field means Monday to Friday.
`/` (Slash)	Specifies step values. Example: `*/15` in the minute field means every 15 minutes.

Special Strings

`@reboot`	Runs the command every time the system reboots.
`@hourly`	Equivalent to `0 * * * *`, runs the command at the beginning of every hour.
`@daily`	Equivalent to `0 0 * * *`, runs the command at midnight every day.
`@weekly`	Equivalent to `0 0 * * 0`, runs the command at midnight every Sunday.
`@monthly`	Equivalent to `0 0 1 * *`, runs the command at midnight on the first day of every month.
`@yearly`	Equivalent to `0 0 1 1 *`, runs the command at midnight on January 1st every year.

Practical Cron Examples

Common Scheduling Examples

These examples demonstrate how to schedule various tasks using cron syntax.

`0 * * * * /path/to/script.sh` - Runs `script.sh` at the beginning of every hour.

`*/5 * * * * /path/to/script.sh` - Runs `script.sh` every 5 minutes.

`0 22 * * 1-5 /path/to/backup.sh` - Runs `backup.sh` at 10 PM on weekdays only.

`30 01 1 * * /path/to/monthly_report.sh` - Runs `monthly_report.sh` at 1:30 AM on the 1st of every month.

`0 0 1 1 mon /path/to/yearly_cleanup.sh` - Runs `yearly_cleanup.sh` at midnight on the first day of year.

Combining Operators

Cron operators can be combined to create more complex schedules. Here are a few examples

`0 9-17 * * mon-fri /path/to/business_hours.sh` - runs the given script every hour from 9 am to 5 pm on weekdays.

`0 0,12 * * sat,sun /path/to/weekend_tasks.sh` - runs the given script at midnight and noon on weekends.

Managing Crontab and Troubleshooting

Crontab Commands

`crontab -e`	Opens the crontab file in a text editor to add or modify cron jobs.
`crontab -l`	Lists the current cron jobs for the user.
`crontab -r`	Removes the current crontab file. Use with caution!
`crontab -u user -e`	Opens the crontab file for a specific user (requires appropriate permissions).

Troubleshooting Cron Jobs

If your cron jobs are not running as expected, consider these troubleshooting steps:

1. **Check Cron Daemon Status:** Ensure the cron daemon is running. Use `systemctl status cron` or `service cron status`.
2. **Examine Cron Logs:** Check the cron logs for errors. Logs are typically located in `/var/log/syslog` or `/var/log/cron`.
3. **Verify Script Permissions:** Make sure the script is executable. Use `chmod +x /path/to/script.sh`.
4. **Use Absolute Paths:** Always use absolute paths to commands and scripts in cron jobs.
5. **Check Environment Variables:** Cron jobs run in a minimal environment. Set any required environment variables in the script or crontab.