Cronjobs

Cornjobs automate routine task to run at specific times triggered by system timer and controlled by cron daemon in the background.

Types of con jobs:

System con jobs

User cron jobs

sudo systemctl status (shows status of cron job daemon)

sudo systemctl stop cron (stops all cron jobs)

sudo systemctl restart cron (refreshes cron daemon)

**Cron job syntax**

6 columns

a – b – c – d – e – f

a: minutes = 0-59

b: hour = 0-23 (military time)

c: day of month = 0-31 (\*=every day of the month)

d: month of year: 0-12 (0-none, 12-dec, \*=every month)

e: day of the week = 0-7 (0-Sun, 1-mon, 2-tue, 3-wed, 4-thu, 5-fri, 6-sat, 7-sun, \*= every day of the week)

f: path of filename (/directory/scriptname.sh)

**cron format**: \*\*\*\*\* /directory/scriptname.sh > /output

e.g. \*20\*4 /root/backup.sh (2am, every month, on Thursdays)

\*2\*\*\* /root/backup.sh (2am, every day, every month)

30\*\*\*\* /root/backup.sh (every 30 mins)

H/2\*\*\*\* /root/backup.sh (every 30 mins)

0\*\*\*\* /root/backup.sh (every minute)

02\*\*\* /root/backup.sh (every 2hrs)

Steps to configure a cronjob:

1. Define job flow by writing a shell script
   1. sudo vi firstcronjob.sh
2. Use an online cron generator to define the timer
   1. \*\*\*\*\* /directory/firstcronjob.sh > tmp (example)
3. On the server, set the crontab table in the crontab file for the current user:

crontab -e <enter>

1. Any number of scheduled tasks can be added to the file, one per line.

* To edit contab for a different user: crontab -u other\_username -e
* To list existing cronjobs set by current user: crontab -l
* To block other user from cronjob:
  + Create a file called cron.allow: sudo vi /etc/cron.allow
  + Add users who should be granted permission to run cronjobs at each line starting with root user.
  + Sudo vi cron.deny (add usernames to be denied access and permission)

HOW TO RUN A SCRIPT IN BACKGROUND MODE

./scriptname.sh&

TO bring a script from background to foreground: fg <process\_id> constant (fg means foreground)

How to run a script in background mode and also ignore network instabilities:

Nohup ./scriptname.sh& (nohup means no hang up so the system ignores connectivity issues)

LINUX COMMAND FOR PROCESS ANALYSIS

sudo apt install atop

top (shows all systems running)

atop (provides more information than top)

htop (prints out easy system analysis health state like CPU, RAM, MEMORY)

ps r prints only running processes in a session

ps T prints all processes running in the terminal