Self Study : ghada emad bhloul os

Comparison between At Vs Crontab Vs Systemd Timers Vs Anacron

At : is a command that allows the users to schedule one-time tasks or recurring jobs at a specific time and date . It is mainly useful for automating the system maintenance , backups, software updates and various administrative tasks.
Used it when scheduling one-time tasks to run at a specific time in the future .

Options with at command
-l : list the at jobs in the queue
-d : delete the at job specified by the job number.
-b : submit a batch job and this is the default behavior.
-v : display vervose information about the job

Crontab : is a command that is used for scheduling and automationg tasks . It facilitates the users to run the script or commands at specified times and intervals . It is ideal for repetitive tasks such as system maintenance, backups and updates
each user can have their crontab file where they can define the scheduling commands that are to be executed .
Format for crontab
Min Hour Dom Mon Dow CMD
Min : specify the minute when the command will run
Hour : specify the hour when the command is scheduled to executed
Dom : specifies the day of the month for the task
Mon : indicates the month during which the command will be executed
Dow : specifies the day of the week for the task

Using of Crontab??
1. scheduling a job for a specific time
2. to view crontab entries using -l option
3. to edit crontab entries edit the current logged-in user’s crontab entries using -e option
4. to schedule a job for every minute using cron
5. to schedule a job for more than one time
6. to schedule a job for a within certain range of time

Systemd Timers : systemd timer units provide a mechanism for scheduling jobs on Linux. The execution time of these jobs can be based on the time and date or on events. systemd timer units are identified by the .timer file name extension. Each timer file requires a corresponding service file it controls. In other words, a timer file activates and manages the corresponding service file. Systemd timers support the following features:

- Jobs scheduled using a timer unit can depend on other systemd services. Timer units are treated as regular systemd services, so can be managed with systemctl
- Timers can be real-time (being triggered on calendar events) or monotonic (being triggered at a specified time elapsed from a certain starting point).
- Time units are logged to the system journal, which makes it easier to monitor and troubleshoot them.
- Timers use the centralized systemd management services.

- If the system is off during the expected execution time, the timer is executed once the system is running again.

Anacron : is used to execute commands periodically with a frequency specified in days . Its main advantage over cron is that it can be used on a machine which is not running continuously. In cron if a machine is not running on time of a scheduled job then it will skip it , but anacron is a bit different as it first checks for timestamp of the job then decides whether to run it or not and if its timestamp is greater than or equal to numbers of days , then runs it after a specified time delay

it mainly constitutes of two import files .
1 . /etc/anacrontab : it contains specifications of jobs
2. /var/spool/anacron : this directory is used for anancron for storing timestamp files.

```
[ghada@localhost etc]$ cd /var/spool
[ghada@localhost spool]$ ls
anacron  at  cron  cups  lpd  mail  plymouth  rhsm
[ghada@localhost spool]$ cd anacron
[ghada@localhost anacron]$ sudo cat cron.daily
[sudo] password for ghada:
20250211
[ghada@localhost anacron]$ sudo cat cron.weekly
[ghada@localhost anacron]$ ls
cron.daily  cron.monthly  cron.weekly
[ghada@localhost anacron]$ sudo cat cron.weekly
[ghada@localhost anacron]$ sudo cat cron.monthly
[ghada@localhost anacron]$ sudo cat cron.daily
20250211
[ghada@localhost anacron]$ sudo cat cron.weekly
[ghada@localhost anacron]$ sudo anacron -d ou
Anacron started on 2025-02-12
Normal exit (0 jobs run)
[ghada@localhost anacron]$ sudo anacron -d -u
Updated timestamp for job `cron.daily' to 2025-02-12
Updated timestamp for job `cron.weekly' to 2025-02-12
Updated timestamp for job `cron.monthly' to 2025-02-12
[ghada@localhost anacron]$ sudo anacron -d -f
Anacron started on 2025-02-12
Will run job `cron.daily' in 24 min.
Will run job `cron.weekly' in 44 min.
Will run job `cron.monthly' in 64 min.
^C
```

Options used within anacron :
-f : used to force execution of the jobs , ignoring the timestamps
-u : only update the timestamps of the jobs , to the current date
-s : serialize execution of jobs . Anacron will not start a new job before the previous one finished
-n : run jobs now , ignore any delay
-d : don't fork to the background , in this mode , anacron will output informational messages to standard error
-q : supppress messages to standard error
-v : print version information and exit
-h : print short usage message and exit

You can add any script to **etc/cron.daily** or **etc/cron.weekly** or **cron.monthly** directory.but remember script should be sh not bash.