

21-12-24

Day 5

## At, batch And cronlab

→ Let's continue with same instance.

### At command:

\* can be used to schedule the job once.

\* name of the service "atd".

### Syntax:

\* at HH:MM

\* atq - to list the jobs executed.

\* atrm - to remove the job.

### Putty:

```
[root@ip-172-31-32-76 ~]# date
```

```
Tue Aug 20 04:26:01 UTC 2024
```

```
[root@ip-172-31-32-76 ~]# at 04:30
```

warning: commands will be executed using /bin/sh

[∴ scheduling a job at  
UTC "04:30"]

```
at> echo > lululush
```

```
at> echo > lrrrrr
```

```
at> echo hello world welcome > ll
```

```
at>
```

∴ [CTRL + D = exit]

```
Job 1 at Tue Aug 20 04:30:00 2024
```

```
[root@ip-172-31-32-76 ~]# ls
```

```
[root@ip-172-31-32-76 ~]# atq
```

[∴ Shows jobs which are scheduled]

```
1 Tue Aug 20 04:30:00 2024 a root
```

root@ip-172-31-32-76 ~# 15

Free  
(after the desired time)

root@ip-172-31-32-76 ~# atq

root@ip-172-31-32-76 ~# 15

ll llllll lululu.sh

root@ip-172-31-32-76 ~# cat ll

hello world welcome

root@ip-172-31-32-76 ~# cat lululu.sh

[∴ blank, because we didn't added anything]

root@ip-172-31-32-76 ~# at 04:34

warning: ..

at > echo hp > screen

at >

∴ [CTRL + D = exit]

job 2 at .. 04:34:00 2024

root@ip-172-31-32-76 ~# atq

2 ... 04:34:00 2024 at last

(∴ after desired time)

root@ip-172-31-32-76 ~# 15

ll llllll lululu.sh screen

root@ip-172-31-32-76 ~# cat screen

hp

root@ip-172-31-32-76 ~# at 04:37

warn--

at > date >> me

[∴ after desired time]

[∴ the job won't be executed until the desired time]

[∴ once it reaches the time, the job will be scheduled and it won't show in "atq"]

[∴ let's schedule another "job"]

[root@ip-172-31-32-76 ~]# ls

ll llllll lulu.sh me screen

[root@ip-172-31-32-76 ~]# at me

Thu Aug 20 04:37:00 UTC 2024

[root@ip-172-31-32-76 ~]# at 04:39

w..

at mkdir Syed-Ibrahim

root

job 8 at ... 04:37:23 2024 2 hat

[root@ip-172-31-32-76 ~]# at 15

ll llll lulu.sh me screen

[root@ip-172-31-32-76 ~]# atq

8 ... 04:39 ...

[root@ip-172-31-32-76 ~]# atrm 8

[root@ip-172-31-32-76 ~]# atq

[root@ip-172-31-32-76 ~]#

[∴ cmd to remove a job]

→ Let's remove all the previous files and folders

Batch:

\* batch command used to execute the set of jobs whenever the system load is low.

\* atq - to check the queued batch jobs.

\* htop - view the cpu utilization.

\* load - below 1.05

To increase cpu load:

```
sudo stress --cpu 8 --timeout 20
```

(increasing cpu load for 20 seconds)

Path:

```
cont
```

```
[root@ip-172-31-34-30 ~]# top
```

[ $\therefore$  to view the cpu utilization]

```
[root@ip-172-31-34-30 ~]# htop
```

```
-bash: htop: command not found
```

```
[root@ip-172-31-34-30 ~]# yum install htop
```

```
⋮  
y  
⋮
```

complete!

```
[root@ip-172-31-34-30 ~]# htop
```

```
cpu [      ] tasks: 39, --  
MEM [|||||] Load Average: 0.00  
SWP [      ] uptime: 00:08:45
```

```
[root@ip-172-31-34-30 ~]# ls
```

```
[root@ip-172-31-34-30 ~]# cat
```

warning: commands will be executed using /bin/sh

```
at> date > banner
```

```
at>
```

[ $\therefore$  CTRL + D = exit]

Job 5 At wed Aug 21 04:08:00 2024

```
[root@ip-172-31-34-30 ~]# ls
```

bsant

```
[root@ip-172-31-34-30 ~]# cat bsant
```

```
wed Aug 21 04:08:11 UTC 2024
```

[∴ This job is scheduled immediately, because our "cpu load" is less than "1.05"]

[∴ If the "cpu load" is greater than "1.05", the job will be in hold and will be executed after only the "cpu load" goes less than "1.05"]

→ gonna increase the "cpu load" by using a command.

```
[root@ip-172-31-34-30 ~]# sudo stress --cpu 8 --timeout 30
```

```
sudo: stress: command not found
```

```
[root@ip-172-31-34-30 ~]# yum install stress
```

```
...  
y  
...
```

complete!

```
[root@ip-172-31-34-30 ~]# htop
```

```
cpu [      ] tasks: 39,...
```

```
MEM [||||| ] load average: 0.00
```

```
swp [      ] uptime:...
```

[∴ Now "cpu load" is "0.00"]

```
[root@ip-172-31-34-30 ~]# sudo stress --cpu 8 --timeout 30
```

```
stress: info: [26526] dispatching hogs: 8 cpu, 0 io, 0 vm, 0 hdd
```

```
stress: info: [26526] successful run completed in 30s
```

```
[root@ip-172-31-34-30 ~]# htop
```

```
...  
load average: 1.78
```

[∴ "cpu load" after running the cmd is "1.78"]

```
[root@ip-172-31-34-30 ~]# batch
```

warning: ---

```
at> date > vbsant1
```

```
at>
```

[∴ CTRL + D = exit]

Job 6 at wed Aug 21 04:11:00 2024

[root@ip-172-31-34-30 ~]# Atq

6 Wed Aug 21 04:11:00 2024 h root

[root@ip-172-31-34-30 ~]# ls

bsant

[~~best~~ root@ip-172-31-34-30 ~]# htop

Load Average: (0.3)

[∴ 'Load Average' is loaded down]

[root@ip-172-31-34-30 ~]# ls

bsant ✓ bsant1

[root@ip-172-31-34-30 ~]#

---

### crontab:

1. `crontab -e` - used to edit system crontabs..
2. `crontab -l` - used to view crontab entries (cron jobs) and display system crontab file contents.
3. `crontab -r` - will remove the current crontab file.
4. `crontab -i` - will show a prompt before removing a user's crontab

### Steps:

1. `crontab -l` (first check any existing crontab is present)
2. `crontab -e` (to edit the crontab)
3. Assign jobs in editor.
4. Execute it

[∴ note: "crontab" only works in "ubuntu"]

→ create a "ubuntu" instance

Path: (crontab-practice)

root@ip-172-31-40-238:~# crontab -e

no crontab for root - using an empty one

select an editor, to change later, run 'select-editor'.

1. /bin/nano ←--- easiest
2. /usr/bin/vim.basic
3. /usr/bin/vim.tiny
4. /bin/ed

choose 1-4 [1]: 2

```
# --
# --
*** date > stop
~
:wq
```

[∴ wq - write and quit]

crontab: installing new crontab

root@ip-172-31-40-238:~# crontab -l

```
# --
# --
```

```
*** date > stop
```

[∴ to view the cron]

root@ip-172-31-40-238:~# ls

snaps

root@ip-172-31-40-238:~# touch stop

root@ip-172-31-40-238:~# ls

snaps stop

root@ip-172-31-40-238:~# cat stop

wed Aug 21 04:31:01 UTC 2024

root@ip-172-31-40-238:~# crontab -e

```
# ...  
# ...  
* * * * * date > stop  
* * * * * date >> stop1  
  
:wq
```

crontab: installing new crontab

root@ip-172-31-40-238:~# date

wed Aug 21 04:33:01 ...

root@ip-172-31-40-238:~# cat stop

wed Aug 21 04:33:01 ...

root@ip-172-31-40-238:~# cat stop1

wed Aug 21 04:33:01  
.. .. 04:34:01  
.. .. 04:35:01

} after a time interval of  
1 minute.

[∴ Here in "crontab" after assigning a cron, if we cat "stop", it will

Show the time which it was assigned, after 1 minute passes if u give cat for  
"stop" the time will change. (simple logic)]

[∴ if u give cat for "stop" again and again in same minute, it cron  
change.]

[∴ same thing applies to "stop1", but it has ">>" so it appends.]