

# OS commands you should know

- [Ubuntu Firewall](#)
  - [sudo ufw status](#)
- [df - disk space monitoring](#)
- [watch - continuously run a command and watches for changes.](#)
- [htop - Advanced version of top](#)
- [Screen - Run an terminal in the background and if you lose connectivity it still runs](#)
- [crontab - Scheduler](#)

# Ubuntu Firewall

# sudo ufw status

## UFW STATUS

-----

Shows you the status of your firewall and what ports are being allow from where. Below you see the L0 and L1 ports are opened and SSH. Please remember that you have Ubuntu OS firewall rules and then you have cloud based rules you should look into.

#sudo ufw status

**Status: active**

To	Action	From
--	-----	----
22/tcp	ALLOW	Anywhere
9000/tcp	ALLOW	Anywhere
9001/tcp	ALLOW	Anywhere
9010/tcp	ALLOW	Anywhere
9011/tcp	ALLOW	Anywhere

# df - disk space monitoring

Very simple command of df -h (h means human readable...aka GB sizes).

df -h

Filesystem	Size	Used	Avail	Use%	Mounted on
tmpfs	1.6G	1.8M	1.6G	1%	/run
/dev/sda3	<b>450G</b>	<b>220G</b>	<b>208G</b>	<b>52%</b>	<b>/</b>
tmpfs	7.8G	0	7.8G	0%	/dev/shm
tmpfs	5.0M	4.0K	5.0M	1%	/run/lock
/dev/sda1	3.9G	6.1M	3.9G	1%	/boot/efi
tmpfs	1.0M	0	1.0M	0%	/mnt/nodectlsecure
tmpfs	1.6G	104K	1.6G	1%	/run/user/1002

You are interested in the / (ROOT) volume. You can see I have 208GB available which should handle my node for quite some time. If you have disk space issues please see [Stararchiver KB](#) under the Node Operator shelf.

# watch - continuously run a command and watches for changes.

This command does not run for every application so if you notice issues then its not compatible. So a good example of how watch works is to use it with df command.

```
watch -d df -h
```

So below I'm watching my filesystem and the watch command highlights in white what has changed. So if you are in DIP status then you should see disk space decreasing. It updates every 2 seconds (which is changeable)

```
Every 2.0s: df
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
tmpfs	1626040	1788	1624252	1%	/run
/dev/sda3	471722200	230300564	217385952	52%	/
tmpfs	8130184	0	8130184	0%	/dev/shm
tmpfs	5120	4	5116	1%	/run/lock
/dev/sda1	3990888	6228	3984660	1%	/boot/efi
tmpfs	1024	0	1024	0%	/mnt/nodectlsecure
tmpfs	1626036	104	1625932	1%	/run/user/1002

CTRL-C to exit out of watch.

# htop - Advanced version of top

## HTOP

All Linux OS's come with top (shows top processes and CPU/Memory details). It still does the job but its suggested you install and use HTOP. Do the following....

1. Login to your node
2. run "apt-get install htop"
3. HTOP is now installed. Just run it and it will look like this..

```
0[|||||] 15.8% 4[|||||] 10.6%
1[|||||] 43.7% 5[|||||] 37.9%
2[|||||] 16.7% 6[|||||] 39.3%
3[|||||] 27.2% 7[|||||] 50.3%
Mem[|||||] 4.94G/15.5G Tasks: 134, 356 thr; 5 running
Swp[|||||] 480M/8.00G Load average: 0.68 0.96 1.11
                                Uptime: 3 days, 22:19:31

0-8 are the processors/cores
Memory usage

PID USER VIRT CPU% MEM% TIME+ Command
1 root 164M 0.0 0.1 2:20.89 /sbin/init splash
291 root 287M 0.0 1.2 1:18.53 /lib/systemd/systemd-journald
340 root 27052 0.0 0.0 0:00.66 /lib/systemd/systemd-udevd
504 root 8584 0.0 0.0 0:05.50 /usr/sbin/haveged --Foreground --verbose=1
505 systemd-o 14836 0.0 0.0 5:32.39 /lib/systemd/systemd-oomd
506 systemd-r 26464 0.0 0.1 1:19.40 /lib/systemd/systemd-resolved
754 root 234M 0.0 0.0 0:17.93 /usr/libexec/accounts-daemon
827 root 234M 0.0 0.0 0:17.80 /usr/libexec/accounts-daemon
876 root 234M 0.0 0.0 0:00.00 /usr/libexec/accounts-daemon
758 root 2816 0.0 0.0 0:00.00 /usr/sbin/acpid
765 avahi 7712 0.0 0.0 0:42.78 avahi-daemon: running [meeseeks.local]
818 avahi 7444 0.0 0.0 0:00.00 avahi-daemon: chroot helper
769 root 9496 0.0 0.0 0:00.44 /usr/sbin/cron -f -P
770 messagebu 11148 0.0 0.0 0:14.44 @dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-act
786 root 82772 0.0 0.0 0:15.70 /usr/sbin/irqbalance --foreground
817 root 82772 0.0 0.0 0:00.00 /usr/sbin/irqbalance --foreground
790 root 41196 0.0 0.1 0:00.11 /usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-triggers
791 root 237M 0.0 0.1 0:02.40 /usr/libexec/polkitd --no-debug
811 root 237M 0.0 0.1 0:00.00 /usr/libexec/polkitd --no-debug
881 root 237M 0.0 0.1 0:00.55 /usr/libexec/polkitd --no-debug
794 root 234M 0.0 0.0 0:00.03 /usr/libexec/power-profiles-daemon
824 root 234M 0.0 0.0 0:00.00 /usr/libexec/power-profiles-daemon
877 root 234M 0.0 0.0 0:00.00 /usr/libexec/power-profiles-daemon
795 syslog 217M 0.0 0.0 0:16.71 /usr/sbin/rsyslogd -n -iNONE
829 syslog 217M 0.0 0.0 0:06.04 /usr/sbin/rsyslogd -n -iNONE
```

If you press F4 (to filter) and type java and press enter it will filter the DAG related processes.

```
0[||||| 4.0%] 4[||||| 5.9%]
1[||||| 4.0%] 5[||||| 3.3%]
2[||||| 3.9%] 6[||||| 2.0%]
3[||||| 5.2%] 7[||||| 4.7%]
Mem[||||| 5.06G/15.5G] Tasks: 131, 336 thr; 1 running
Swp[||||| 480M/8.00G] Load average: 0.83 0.97 1.09
Uptime: 3 days, 22:22:28
```

PID	USER	VIRT	CPU%	MEM%	TIME+	Command
757195	root	11.5G	17.1	15.5	16:40.25	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl-no
757197	root	11.5G	0.0	15.5	0:01.30	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl
757198	root	11.5G	0.0	15.5	0:08.25	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl
757199	root	11.5G	0.0	15.5	0:00.00	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl
757200	root	11.5G	0.0	15.5	0:00.52	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl
757201	root	11.5G	0.0	15.5	0:00.04	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl
757202	root	11.5G	0.0	15.5	0:01.14	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl
757203	root	11.5G	0.0	15.5	0:00.76	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl
757204	root	11.5G	0.0	15.5	0:00.87	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl
757206	root	11.5G	0.0	15.5	0:01.04	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl
757207	root	11.5G	0.0	15.5	0:00.00	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl
757208	root	11.5G	0.0	15.5	0:00.33	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl
757209	root	11.5G	0.0	15.5	1:00.02	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl
757210	root	11.5G	0.0	15.5	0:10.82	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl
757211	root	11.5G	0.0	15.5	0:00.21	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl
757213	root	11.5G	0.0	15.5	0:01.43	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl
757214	root	11.5G	0.0	15.5	0:00.01	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl
757220	root	11.5G	0.0	15.5	0:08.11	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl
757221	root	11.5G	0.0	15.5	0:08.09	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl
757222	root	11.5G	0.0	15.5	0:08.27	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl
757223	root	11.5G	0.0	15.5	0:07.96	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl
757224	root	11.5G	0.0	15.5	0:08.27	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl
757225	root	11.5G	0.0	15.5	0:08.06	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl
757226	root	11.5G	0.0	15.5	0:07.98	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl
757227	root	11.5G	0.0	15.5	0:00.53	/usr/bin/java -jar -Xms1024M -Xmx7G -Xss256K /var/tessellation/dag-10/cl

# Screen - Run an terminal in the background and if you lose connectivity it still runs

is a utility that lets you run an application/script/terminal and then place it in the background so you can come back to it later. If you were to lose your network connection to your SSH terminal then you can go back to where you were.

1. run " " "
2. You are now in the first screen. Run whatever you need to run and press CTRL-A then D to detach.
3. You have now dethatched from your terminal. If you want to return to that terminal all you do is type screen -r (return

If you have multiple screens launched you just return to the proper screen. It will list them for you automatically like so...

```
nodeadmin@Constellation-Node:/scripts$
```

There are screens on:

832022.pts-0.meeseeks	(05/27/2025 03:45:36 AM)	(Attached)
831693.pts-0.meeseeks	(05/27/2025 03:42:36 AM)	(Attached)

Now you can launch your process in screen, detach, put your computer to sleep/power off and come back later.



# crontab - Scheduler

**Crontab** - Crontab is scheduler so you can launch scripts hourly/daily/weekly/monthly.

1. type "sudo crontab -e" (Edit the crontab)
2. Pick 1. nano unless you like other editors.

```
nodeadmin@Constellation-Node:/scripts$ crontab -e
no crontab for nodeadmin - 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]: █
```

Add the following to the bottom of the crontab to run the script at 10AM Sunday

**0 10 0 /scripts/<starchiver script> -options**

To run it every Sunday at, for example, 10:00 AM, add:

bash

Copy

Edit

0 10 \* \* 0 /scripts/myscript.sh



## Explanation:

- 0 → minute
- 10 → hour (10 AM)
- \* → every day of the month
- \* → every month
- 0 → Sunday (can also use 7)

It should look like this...