

# How To Use Top, Netstat, Du, & Other Tools to Monitor Server Resources

Authored by: **ASPHostServer Administrator** [asphostserver@gmail.com]

Saved From: <http://faq.asphosthelpdesk.com/article.php?id=279>

---

## How Do I Monitor Process Utilization?

---

### top

---

One of the most common tools for checking the resource utilization of processes is "**top**".

Top provides a simple, real-time table of your processes, with the largest consumers on top:

top

```
top - 14:45:52 up 29 min,  1 user,  load average: 0.10, 0.09, 0.06
Tasks:  56 total,   1 running,  55 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.0%us,  0.3%sy,  0.0%ni, 99.7%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   1019600k total,  393756k used,  625844k free,   11136k buffers
Swap:      0k total,    0k used,    0k free,   316748k cached

  PID %MEM  VIRT  SWAP  RES  CODE  DATA  SHR  nFLT  nDRT  S  PR  NI  %CPU  COMMAND
  832  1.3 32364  18m  12m  896   11m 1688    1    0  S  20  0  0.0  bash
  820  0.4 89456  83m 4008  488   948 3040   12    0  S  20  0  0.0  sshd
  812  0.3 49948  46m 2828  488   616 2216    0    0  S  20  0  0.0  sshd
    1  0.2 24192  21m 2108  152   868 1300   23    0  S  20  0  0.0  init
  400  0.1  243m 242m 1420  344  216m 1084    0    0  S  20  0  0.0  rsyslogd
```

The top portion has some system statistics, including load averages for the past minute, 5 minutes, and 15 minutes. It also shows memory and swap usage, and the count of various process states.

The bottom portion has every process on the system, organized by the top users of resources. This list is updated in real-time.

### htop

---

Although "top" is included in almost every distribution by default, an improved version, called "**htop**" is available for download from most repositories.

To install htop on Ubuntu, type the following:

```
sudo apt-get install htop
```

Running htop, we can see that it has a similar output, but is colored, and is more interactive:

```
htop

CPU[ |                               0.7%]      Tasks: 21, 3 thr; 1 running
  Mem[ ||||| | | | | | | | | | | | | 64/995MB]    Load average: 0.00 0.02 0.05
  Swp[                               0/0MB]      Uptime: 00:37:37
  PID USER      PRI  NI  VIRT   RES   SHR  S CPU% MEM%   TIME+  Command
 2752 root         20   0 25660  1876  1364 R  0.0  0.2  0:00.06 htop
    1 root         20   0 24192  2108  1300 S  0.0  0.2  0:00.55 /sbin/init
  312 root         20   0 17224   640   444 S  0.0  0.1  0:00.04 upstart-udev-brid
  314 root         20   0 21592  1360   760 S  0.0  0.1  0:00.04 /sbin/udev --dae
  394 messagebu  20   0 23808   688   436 S  0.0  0.1  0:00.01 dbus-daemon --sys
  401 syslog      20   0 243M  1420  1084 S  0.0  0.1  0:00.07 rsyslogd -c5
  402 syslog      20   0 243M  1420  1084 S  0.0  0.1  0:00.00 rsyslogd -c5
```

The top portion is much easier to read and the bottom portion is organized in a more clear fashion.

Here are some keys that will help you use htop more effectively:

- **M**: Sort processes by memory usage
- **P**: Sort processes by processor usage
- **?**: Access help
- **k**: Kill current/tagged process
- **F2**: Setup htop. You can choose display options here.
- **/**: Search processes

There are plenty of more options you that you can access through help or setup. These should be your first stops in exploring htop's functionality.

## How Do I Find Out Which Program Is Using My Bandwidth?

### nethogs

If your network connection seems saturated and you are unsure which application is the culprit, a program called "**nethogs**" is a good choice for finding out.

On Ubuntu, you can install nethogs with the following command:

```
sudo apt-get install nethogs
```

We can run it by simply typing:

nethogs

NetHogs version 0.8.0

PID	USER	PROGRAM	DEV	SENT	RECEIVED
3379	root	/usr/sbin/sshd	eth0	0.485	0.182 KB/sec
820	root	sshd: root@pts/0	eth0	0.427	0.052 KB/sec
?	root	unknown TCP		0.000	0.000 KB/sec
TOTAL				0.912	0.233 KB/sec

As you can see, above all, nethogs output is simple. It associates each application with its associated network traffic.

There are only a few commands that you can use to control nethogs:

- **m**: Change displays between "kb/s", "kb", "b", and "mb".
- **r**: Sort by traffic received.
- **s**: Sort by traffic sent.
- **q**: quit

Although this is a simple tool, nethogs is a great way to associate traffic with a specific applications.

## IPTraff

**IPTraff** is another great way to monitor network traffic. It provides a number of different interactive monitoring interfaces.

On Ubuntu, you can install IPTraff with the following command:

```
sudo apt-get install iptraf
```

To run the program, simply call it from the command line with root privileges:

```
sudo iptraf
```

```
????????????????????????????????????????????????????????????
? IP traffic monitor ?
? General interface statistics ?
? Detailed interface statistics ?
? Statistical breakdowns... ?
? LAN station monitor ?
????????????????????????????????????????????????????????????
? Filters... ?
????????????????????????????????????????????????????????????
? Configure... ?
????????????????????????????????????????????????????????????
```



The **"netstat"** command is a versatile tool for gathering network information. It is extremely flexible and powerful.

By default, netstat prints a list of open sockets:

```
netstat
```

```
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 192.241.187.204:ssh    ip223.hichina.com:50324 ESTABLISHED
tcp      0      0 192.241.187.204:ssh    rrcs-72-43-115-18:50615 ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags           Type           State           I-Node   Path
unix   5      [ ]            DGRAM          State           6559     /dev/log
unix   3      [ ]            STREAM         CONNECTED       9386
unix   3      [ ]            STREAM         CONNECTED       9385
. . .
```

If we add an **"-a"** option, it will list all ports, listening and non-listening:

```
netstat -a
```

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 *:ssh                   *:*                     LISTEN
tcp      0      0 192.241.187.204:ssh    rrcs-72-43-115-18:50615 ESTABLISHED
tcp6     0      0 [::]:ssh                [::]:*                 LISTEN
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags           Type           State           I-Node   Path
unix   2      [ ACC ]           STREAM         LISTENING       6195     @/com/ubuntu/upstart
unix   2      [ ACC ]           STREAM         LISTENING       7762     /var/run/acpid.socket
unix   2      [ ACC ]           STREAM         LISTENING       6503     /var/run/dbus/system_bus_socket
. . .
```

If you'd like to filter to see only TCP or UDP connections, use the **"-t"** or **"-u"** flags respectively:

```
netstat -at
```

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 *:ssh                   *:*                     LISTEN
tcp      0      0 192.241.187.204:ssh    rrcs-72-43-115-18:50615 ESTABLISHED
tcp6     0      0 [::]:ssh                [::]:*                 LISTEN
```

See statistics by passing the **"-s"** flag:

```
netstat -s
```

```
Ip:
  13500 total packets received
  0 forwarded
  0 incoming packets discarded
  13500 incoming packets delivered
  3078 requests sent out
  16 dropped because of missing route
Icmp:
  41 ICMP messages received
  0 input ICMP message failed.
  ICMP input histogram:
    echo requests: 1
    echo replies: 40
. . .
```

If you would like to continuously update the output, you can use the "-c" flag.

There are many other options that can change the output. Explore the man pages for more ideas.

## How Do I Find Out How Much Disk Space I Have Left?

---

### df

---

For a quick overview of how much disk space you have left on your drives, you can use the "df" program.

Without any options, its output looks like this:

```
df

Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/vda         31383196 1228936  28581396   5% /
udev              505152         4    505148   1% /dev
tmpfs            203920         204    203716   1% /run
none              5120           0       5120   0% /run/lock
none            509800         0    509800   0% /run/shm
```

This outputs disk usage in bytes, which may be a bit hard to read.

To fix this problem, we can specify to output in a human-readable format:

```
Filesystem      Size  Used Avail Use% Mounted on
/dev/vda         30G  1.2G  28G   5% /
udev             494M  4.0K  494M   1% /dev
tmpfs            200M  204K  199M   1% /run
none             5.0M    0  5.0M   0% /run/lock
none            498M    0  498M   0% /run/shm
```

If we want to see the total disk space available on all filesystems, we can pass the "--total" option. This will add a row at the bottom with summary information:

```
df -h --total
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/vda	30G	1.2G	28G	5%	/
udev	494M	4.0K	494M	1%	/dev
tmpfs	200M	204K	199M	1%	/run
none	5.0M	0	5.0M	0%	/run/lock
none	498M	0	498M	0%	/run/shm
total	32G	1.2G	29G	4%	

## du

---

While df is an easy way of getting an overview, "**du**" gives a better picture of what is taking up space on your system.

The command will analyze usage for the current directory and any subdirectories. The default output of du looks like this:

```
du
```

```
4 ./cache
8 ./ssh
28 .
```

Once again, we can specify human-readable output by passing it "-h":

```
du -h
```

```
4.0K ./cache
8.0K ./ssh
28K .
```

To see file sizes as well as directories, type the following:

```
du -a
```

```
0 ./cache/motd.legal-displayed
4 ./cache
4 ./ssh/authorized_keys
8 ./ssh
4 ./profile
4 ./bashrc
4 ./bash_history
28 .
```

For a total at the bottom, you can add the "-c" option:

```
du -c

4 ./cache
8 ./ssh
28 .
28 total
```

If you are only interested in the total and not the specifics, you can issue:

```
du -s

28 .
```

## Improvements

---

These two tools have improved versions that can be installed on Ubuntu.

An improved version of df is "**pydf**". It can be installed with this command:

```
sudo apt-get install pydf
```

The pydf command organizes everything in neat charts with colored output. It shows disk usage graphically with usage bars:

```
pydf -a

dev/vda      30G 1200M   27G  3.9 [.....] /
udev        493M 4096B   493M  0.0 [.....] /dev
devpts              0    0    0   - [.....] /dev/pts
proc              0    0    0   - [.....] /proc
tmpfs        199M  204k   199M  0.1 [.....] /run
none         5120k    0 5120k  0.0 [.....] /run/lock
none         498M    0  498M  0.0 [.....] /run/shm
. . .
```

An improvement on du is "**ncdu**". This command can be installed by typing:

```
sudo apt-get install ncdu
```

This command uses an interactive ncurses display to graphically represent your disk usage:

```
ncdu

--- /root -----
8.0KiB [#####] /.ssh
4.0KiB [#####] /.cache
4.0KiB [#####] .bashrc
```



```
4.0KiB [##### ] .profile
4.0KiB [##### ] .bash_history
```

You can step through the filesystem by using the up and down arrows and pressing "enter" on any directory entry.

## How Do I Find Out How Much of my Memory Is In Use?

---

### free

---

The easiest way of finding out the current memory usage on your system is using the "free" command.

When used without options, the output looks like this:

```
free

              total        used         free       shared    buffers     cached
Mem:           507620      408172      99448           0       123672      248224
-/+ buffers/cache:      36276      471344
Swap:              0           0           0
```

To display in a more readable format, we can pass the "-m" option to display the output in megabytes:

```
free -m

              total        used         free       shared    buffers     cached
Mem:              495          398           97           0          120          242
-/+ buffers/cache:           35          460
Swap:              0           0           0
```

The middle line, marked "-/+ buffers/cache", will show the actual memory used by applications.

The "Mem" line includes the memory used for buffering and caching, which is freed up as soon as needed for other purposes.

### vmstat

---

The "vmstat" command can output various information about your system, including memory, swap, disk io, and cpu information.

We will use the command to get another view into memory usage:

```
vmstat

procs -----memory----- ---swap-- -----io----- -system-- ----cpu----
 r  b  swpd  free  buff  cache  si  so  bi  bo  in  cs  us  sy  id  wa
```

```
1 0 0 99340 123712 248296 0 0 0 1 9 3 0 0 100 0
```

We can see this in megabytes by choosing our unit with the "-S" flag:

```
vmstat -S M
```

```
procs -----memory----- ---swap-- -----io----- -system-- -----cpu-----
 r  b   swpd   free   buff   cache   si   so   bi   bo   in   cs  us  sy  id  wa
 1  0     0    96    120    242    0    0    0    1    9    3  0  0 100  0
```

As you can see, this tool helps us break down the "-/+ buffers/cache" category of the "free" command. We get to see how much of that block is used for buffering and how much for cache.

To get some general statistics about memory usage, type:

```
vmstat -s -S M
```

```
495 M total memory
398 M used memory
252 M active memory
119 M inactive memory
 96 M free memory
120 M buffer memory
242 M swap cache
  0 M total swap
  0 M used swap
  0 M free swap
```

. . . .

To get information about individual system processes' cache usage, type:

```
vmstat -m -S M
```

Cache	Num	Total	Size	Pages
ext4_groupinfo_4k	195	195	104	39
UDPLITEv6	0	0	768	10
UDPv6	10	10	768	10
tw_sock_TCPv6	0	0	256	16
TCPv6	11	11	1408	11
kcopyd_job	0	0	2344	13
dm_uevent	0	0	2464	13
bsg_cmd	0	0	288	14

. . . .

This will give you details about what kind of information is stored in the cache.

## Conclusion

Using these tools, you should begin to be able to monitor your server from the command line. There are many other utilities that perform simple monitoring operations, but these are a good starting point.