

**Subscribe now**

X

Get the highlights in your
inbox every week.

**Subscribe****LOG IN****SIGN UP****M**
[Art](#)
[Op](#)[sAbout](#)[Privacy Statement](#)

20 Linux commands every sysadmin should know

If your application isn't working—or you're just looking for more information—these 20 commands will come in handy.

14 Oct 2020 | [Rosemary Wang_\(/users/joatmon08\)](#) | 795 | [31 comments](#)

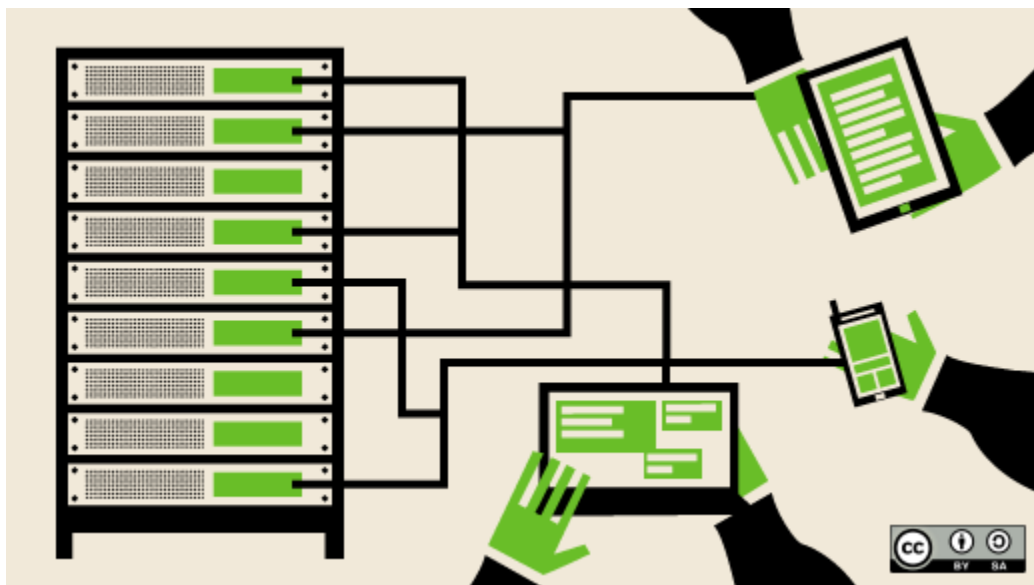


Image by : Opensource.com

In a world bursting with new tools and diverse development environments, it's

Subscribe now

Get the highlights in your inbox every week.

X developer or engineer to learn some basic commands and packages can help developers timize their applications and—when things go › information to operators and sysadmins.

per or want to manage your own application, the ›mmands can help you better understand your lp you describe problems to sysadmins ›tion might work locally but not on a remote host. ›x development environments, containers, virtual ›al.

1. curl

curl transfers a URL. [Use this command \(https://opensource.com/article/20/5/curl-cheat-sheet\)](https://opensource.com/article/20/5/curl-cheat-sheet) to test an application's endpoint or connectivity to an upstream service endpoint. **curl** can be useful for determining if your application can reach another service, such as a database, or checking if your service is healthy.

As an example, imagine your application throws an HTTP 500 error indicating it can't reach a MongoDB database:

```
$ curl -I -s myapplication:5000
HTTP/1.0 500 INTERNAL SERVER ERROR
```

The **-I** option shows the header information and the **-s** option silences the response body. Checking the endpoint of your database from your local desktop:

```
$ curl -I -s database:27017
HTTP/1.0 200 OK
```

So what could be the problem? Check if your application can get to other places besides the database from the application host:

Subscribe now

X [source.com](https://opensource.com)

Get the highlights in your
inbox every week.

y to reach the database from the application host.
atabase's hostname, so try that first:

```
host 'database'
```

ition cannot resolve the database because the
able or the host (container or VM) does not have a
ve the hostname.

2. python -m json.tool / jq

After you issue **curl**, the output of the API call may be difficult to read.

Sometimes, you want to [pretty-print](https://en.wikipedia.org/wiki/Prettyprint) (<https://en.wikipedia.org/wiki/Prettyprint>) the JSON output to find a specific entry. Python has a built-in JSON library that can help with this. You use **python -m json.tool** to indent and organize the JSON. To use Python's JSON module, pipe the output of a JSON file into the **python -m json.tool** command.

```
$ cat test.json
{"title": "Person", "type": "object", "properties": {"firstName": {"type": "string"
```

To use the Python library, pipe the output to Python with the **-m** (module) option.

```
$ cat test.json | python -m json.tool
{
  "properties": {
    "age": {
      "description": "Age in years",
```

```

        "minimum": 0,
        "type": "integer"
    },
    "firstName": {

```

Subscribe now

X

Get the highlights in your
inbox every week.

...sing, you can [install jq](https://stedolan.github.io/jq/download/) (<https://stedolan.github.io/jq/download/>). **jq** provides some [options](https://stedolan.github.io/jq/manual/v1.5/) (<https://stedolan.github.io/jq/manual/v1.5/>) that extract specific values from the JSON input. To pretty-print like the Python module above, simply apply **jq** to the output.

```

$ cat test.json | jq
{
  "title": "Person",
  "type": "object",
  "properties": {
    "firstName": {
      "type": "string"
    },
    "lastName": {
      "type": "string"
    },
    "age": {
      "description": "Age in years",
      "type": "integer",
      "minimum": 0
    }
  },
  "required": [
    "firstName",
    "lastName"

```

```
]
}
```

Subscribe now

X

Get the highlights in your inbox every week.

admins and developers issue [this command](#) [/19/7/master-ls-command](#)) quite often. In the world of containers, `ls` can help determine your container image's working directory. If you're picking up your files, `ls` can help you examine your file system. If you're looking up your files, `ls` can help you examine your file system. If you're looking up your files, you can't run myapp because of a permission error. To check the permissions using `ls -l`, you realize that the permissions are `-rw-r--r--`, which are read and write only.

```
$ ls -l myapp
-rw-r--r--. 1 root root 33 Jul 21 18:36 myapp
```

4. tail

tail displays the last part of a file. You usually don't need every log line to troubleshoot. Instead, you want to check what your logs say about the most recent request to your application. For example, you can use **tail** to check what happens in the logs when you make a request to your [Apache HTTP server](https://httpd.apache.org/) (<https://httpd.apache.org/>).

```
[root@localhost ~]# tail -f /var/log/httpd/access_log
::1 - - [21/Jul/2017:18:46:58 +0000] "GET / HTTP/1.1" 403 4897 "-" "curl/7.29.0"
::1 - - [21/Jul/2017:18:47:00 +0000] "GET / HTTP/1.1" 403 4897 "-" "curl/7.29.0"
::1 - - [21/Jul/2017:18:47:02 +0000] "GET / HTTP/1.1" 403 4897 "-" "curl/7.29.0"
+0000] "GET / HTTP/1.1" 403 4897 "-" "curl/7.29.0"
X +0000] "GET / HTTP/1.1" 403 4897 "-" "curl/7.29.0"
+0000] "GET / HTTP/1.1" 403 4897 "-" "curl/7.29.0"
+0000] "GET / HTTP/1.1" 403 4897 "-" "curl/7.29.0"
+0000] "GET / HTTP/1.1" 403 4897 "-" "curl/7.29.0"
+0000] "GET / HTTP/1.1" 403 4897 "-" "curl/7.29.0"
+0000] "GET / HTTP/1.1" 403 4897 "-" "curl/7.29.0"
+0000] "GET / HTTP/1.1" 403 4897 "-" "curl/7.29.0"
```

Subscribe now

Get the highlights in your
inbox every week.

HTTP logs and see the requests as they happen.

ow" option, which outputs the log lines as they are
has a background script that accesses the
snapshot every few seconds and the log records the request. Instead of following
the log in real time, you can also use **tail** to see the last 100 lines of the file with
the **-n** option.

```
$ tail -n 100 /var/log/httpd/access_log
```

5. cat

cat concatenates and prints files. [You might issue cat \(https://opensource.com/article/19/2/getting-started-cat-command\)](https://opensource.com/article/19/2/getting-started-cat-command) to check the contents of your dependencies file or to confirm the version of the application that you have already built locally.

```
$ cat requirements.txt
flask
flask_pymongo
```

The example above checks whether your Python Flask application has Flask listed as a dependency.

6. grep

grep searches file patterns. If you are looking for a specific pattern in the output

Subscribe now

Get the highlights in your inbox every week.

X highlights the relevant lines. Use this command for processes, and more. If you want to see if Apache come overwhelmed by the number of lines. By **grep** command, you isolate the lines that indicate server

```
apache.catalina.startup.Catalina.start
^O [main] org.apache.catalina.startup.Catalina.s
```

the **ps** command, part of the **procps-ng** package which provides useful commands for investigating process IDs, shows the status of a running process. Use this command to determine a running application or confirm an expected process. For example, if you want to check for a running Tomcat web server, you use **ps** with its options to obtain the process ID of Tomcat.

```
$ ps -ef
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0	2	18:55	?	00:00:02	/docker-java-home/jre/bi
root	59	0	0	18:55	pts/0	00:00:00	/bin/sh
root	75	59	0	18:57	pts/0	00:00:00	ps -ef

For even more legibility, use **ps** and pipe it to **grep**.

```
$ ps -ef | grep tomcat
```

root	1	0	1	18:55	?	00:00:02	/docker-java-home/jre/bi
------	---	---	---	-------	---	----------	--------------------------

8. env

env allows you to set or print the environment variables. During troubleshooting, you may find it useful for checking if the wrong environment variable prevents

your application from starting. In the example below, this command is used to check the environment variables set on your application's host.

Subscribe now

X

Get the highlights in your
inbox every week.

al/sbin

017/test

sing Python and has environment variables to
se.

9. top

top displays and updates sorted process information. Use this [monitoring tool](https://opensource.com/life/16/2/open-source-tools-system-monitoring) (<https://opensource.com/life/16/2/open-source-tools-system-monitoring>) to determine which processes are running and how much memory and CPU they consume. A common case occurs when you run an application and it dies a minute later. First, you check the application's return error, which is a memory error.

```
$ tail myapp.log
Traceback (most recent call last):
MemoryError
```

Is your application *really* out of memory? To confirm, use **top** to determine how much CPU and memory your application consumes. When issuing **top**, you notice a Python application using most of the CPU, with its memory usage climbing, and suspect it is your application. While it runs, you hit the "C" key to see the full command and reverse-engineer if the process is your application. It turns out to be your memory-intensive application (**memeater.py**). When your application has run out of memory, the system kills it with an out-of-memory

(OOM) error.

Subscribe now

Get the highlights in your
inbox every week.

```
top - 19:03:48 up 43 min, 2 users, load average: 0.47, 0.13, 0.08
Tasks: 99 total, 1 running, 98 sleeping, 0 stopped, 0 zombie
      0 running, 0 sleeping, 0 stopped, 0 zombie
      0 id, 23.2 wa, 0.0 hi, 12.7 si, 0.0 st
      0 free, 1735948 used, 52628 buff/cache
      0 free, 786420 used. 33364 avail Mem
```

RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
291g	120	R	72.8	71.8	0:06.87	python
0	0	S	10.7	0.0	0:02.04	kswapd0
0	0	S	3.2	0.0	0:00.27	kworker/0:2H
272	180	S	1.9	0.0	0:00.15	httpd
520	64	S	0.6	0.0	0:00.46	tuned
0	0	R	0.3	0.0	0:00.31	rcu_sched
600	360	S	0.3	0.0	0:00.32	dbus-daemon
772	480	S	0.0	0.0	0:01.33	systemd
0	0	S	0.0	0.0	0:00.00	kthreadd

the application increases, eventually being OOM-killed.

```
top - 19:03:48 up 43 min, 2 users, load average: 0.50, 0.18, 0.10
Tasks: 99 total, 1 running, 98 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.8 us, 60.5 sy, 0.0 ni, 0.0 id, 23.4 wa, 0.0 hi, 15.2 si, 0.0 st
KiB Mem : 1883804 total, 70284 free, 1763616 used, 49904 buff/cache
KiB Swap: 1572860 total, 912480 free, 660380 used. 6828 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
18770	root	20	0	2049280	1.491g	72	D	70.4	83.0	0:06.84	python memeater.py
25	root	20	0	0	0	0	S	13.7	0.0	0:03.63	[kswapd0]
18766	root	0	-20	0	0	0	S	3.2	0.0	0:00.21	[kworker/0:0H]
16464	root	20	0	338484	0	0	S	1.9	0.0	0:00.62	/usr/sbin/VBoxService --pidfile /var/run/vboxadd+
3	root	20	0	0	0	0	S	0.3	0.0	0:00.16	[ksoftirqd/0]
555	root	20	0	24192	140	60	S	0.3	0.0	0:00.07	/usr/lib/systemd/systemd-logind
556	dbus	20	0	32924	520	208	S	0.3	0.0	0:00.34	/bin/dbus-daemon --system --address=systemd: --n+
1	root	20	0	127864	776	488	S	0.0	0.0	0:01.34	/usr/lib/systemd/systemd --system --deserialize +
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	[kthreadd]
6	root	20	0	0	0	0	S	0.0	0.0	0:00.04	[kworker/u2:0]
7	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	[migration/0]
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	[rcu bh]

By hitting the "C" key, you can see the full command that started the application.

In addition to checking your own application, you can use **top** to debug other processes that utilize CPU or memory.

10. netstat

netstat shows the network status. This command shows network ports in use and their incoming connections. However, **netstat** does not come out-of-the-box on Linux. If you need to install it, you can find it in the [net-tools](#)

(<https://wiki.linuxfoundation.org/networking/net-tools>) package. As a developer who experiments locally or pushes an application to a host, you may receive an error that a port is already allocated or an address is already in use. Using

Subscribe now

Get the highlights in your
inbox every week.

X and port options demonstrates that Apache HTTP
the below host.

Foreign Address	State	PID/Program name
0.0.0.0:*	LISTEN	1/systemd
0.0.0.0:*	LISTEN	823/sshd
0.0.0.0:*	LISTEN	1053/master
:::*	LISTEN	1/systemd
:::*	LISTEN	18310/httpd
:::*	LISTEN	823/sshd
:::*	LISTEN	1053/master
0.0.0.0:*		559/chronyd
0.0.0.0:*		644/dhclient
0.0.0.0:*		644/dhclient
:::*		559/chronyd
:::*		644/dhclient

Using **netstat -tulpn** shows that Apache already uses port 80 on this machine.

11. ip

If **ip** address does not work on your host, it must be installed with the [iproute2](https://wiki.linuxfoundation.org/networking/iproute2) (<https://wiki.linuxfoundation.org/networking/iproute2>) package. The subcommand **address** (or just **ip a** for short) shows the interfaces and IP addresses of your application's host. You use **ip address** to verify your container or host's IP address. For example, when your container is attached to two networks, **ip address** can show which interface connects to which network. For a simple check, you can always use the **ip address** command to get the IP address of the host. The example below shows that the web tier container has an IP address of 172.17.0.2 on interface eth0.

```
$ ip address show eth0
```

```
3: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
    link/ether d4:3b:04:9e:b2:c2 brd ff:ff:ff:ff:ff:ff
    inet 10.1.1.3/27 brd 10.1.1.31 scope global dynamic noprefixroute eth0
        valid_lft 52072sec preferred_lft 52072sec
```

12. Lsof

Lsof lists the open files associated with your application. On some Linux machine

Subscribe now

Get the highlights in your
inbox every week.

X f with the [lsof \(http://www.linuxfromscratch.org\)](http://www.linuxfromscratch.org) package. In Linux, almost any interaction with . As a result, if your application writes to a file or **lsof** will reflect that interaction as a file. Similar to check for listening ports. For example, if you want to use **Lsof** to check which process is using it. Below, **lsof** listens on port 80. You can also use **Lsof** to examining where the web server's binary resides

```

COMMAND  PID  USER  FD   TYPE    DEVICE  SIZE/OFF      NODE NAME
httpd    18312 apache  4u    IPv6  32945    0t0  TCP *:http (LISTEN)
httpd    18313 apache  4u    IPv6  32945    0t0  TCP *:http (LISTEN)
httpd    18314 apache  4u    IPv6  32945    0t0  TCP *:http (LISTEN)
httpd    18315 apache  4u    IPv6  32945    0t0  TCP *:http (LISTEN)
[root@localhost ~]# lsof -p 18311
COMMAND  PID  USER  FD   TYPE    DEVICE  SIZE/OFF      NODE NAME
httpd    18311 apache  cwd    DIR      253,0    239        64 /
httpd    18311 apache  rtd    DIR      253,0    239        64 /
httpd    18311 apache  txt    REG      253,0  519432  34542018 /usr/sbin/httpd
httpd    18311 apache  mem    REG      253,0   62184  34380923 /usr/lib64/libnss_files-2.17.so
httpd    18311 apache  mem    REG      253,0   27704  34380905 /usr/lib64/httpd/modules/mod_cgi.so

```

Lsof shows that httpd listens on port 80. Examining httpd's process ID also shows all the files httpd needs in order to run.

The name of the open file in the list of open files helps pinpoint the origin of the process, specifically Apache.

13. df

You can use **df** (display free disk space) to troubleshoot disk space issues. When you run your application on a container orchestrator, you might receive an error message signaling a lack of free space on the container host. While disk space should be managed and optimized by a sysadmin, you can use **df** to figure out the existing space in a directory and confirm if you are indeed out of space.

```
$ df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
devtmpfs	7.7G	0	7.7G	0%	/dev
tmpfs	16G	35G	31%	/	
X	246M	2.8G	9%	/boot	
	88G	13G	88%	/home	
	9.4M	191M	5%	/boot/efi	
	55G	54G	51%	/run/media/tux/red	

Subscribe now

Get the highlights in your
inbox every week.

ormation in human-readable format. By default, as
sults for everything under the root directory, but
roviding a directory as part of your command (such

To retrieve more detailed information about which files use the disk space in a directory, you can use the **du** command. If you wanted to find out which log takes up the most space in the **/var/log** directory, for example, you can use **du** with the **-h** (human-readable) option and the **-s** option for the total size.

```
$ du -sh /var/log/*
1.8M  /var/log/anaconda
384K  /var/log/audit
4.0K  /var/log/boot.log
0     /var/log/chrony
4.0K  /var/log/cron
4.0K  /var/log/maillog
64K   /var/log/messages
```

The example above reveals the largest directory under **/var/log** to be **/var/log/audit**. You can use **du** in conjunction with **df** to determine what utilizes the disk space on your application's host.

15. id

To check the user running the application, use the **id** command to return the user

identity. The example below uses [Vagrant \(https://www.vagrantup.com/\)](https://www.vagrantup.com/) to test the application and isolate its development environment. After you log into the Vagrant box, if you try to install Apache HTTP Server (a dependency) the system

Subscribe now

Get the highlights in your inbox every week.

X the command as root. To check your user and and notice that you are running as the "vagrant"

```

r
rm this command.

```

```

agrant) groups=1000 (vagrant) context=unconfined

```

ie command as a superuser, which provides

16. chmod

When you run your application binary for the first time on your host, you may receive the error message "permission denied." As seen in the example for **ls**, you can check the permissions of your application binary.

```

$ ls -l
total 4
-rw-rw-r--. 1 vagrant vagrant 34 Jul 11 02:17 test.sh

```

This shows that you don't have execution rights (no "x") to run the binary. **chmod** can correct the permissions to enable your user to run the binary.

```

$ chmod +x test.sh
[vagrant@localhost ~]$ ls -l
total 4
-rwxrwxr-x. 1 vagrant vagrant 34 Jul 11 02:17 test.sh

```

As demonstrated in the example, this updates the permissions with execution rights. Now when you try to execute your binary, the application doesn't throw a

permission-denied error. **Chmod** may be useful when you load a binary into a container as well. It ensures that your container has the correct permissions to execute your binary.

Subscribe now

X

Get the highlights in your
inbox every week.

helps resolve a URL to a set of application servers. JRL does not resolve, which causes a connectivity example, say you attempt to access your database your application's host. Instead, you receive a message, you try using **dig** (DNS lookup utility) or (the servers) to figure out why the application can't

```
Server: 10.0.2.3
Address: 10.0.2.3#53
```

```
** server can't find mydatabase: NXDOMAIN
```

Using **nslookup** shows that **mydatabase** can't be resolved. Trying to resolve with **dig** yields the same result.

```
$ dig mydatabase
```

```
; <<>> DiG 9.9.4-RedHat-9.9.4-50.el7_3.1 <<>> mydatabase
;; global options: +cmd
;; connection timed out; no servers could be reached
```

These errors could be caused by many different issues. If you can't debug the root cause, reach out to your sysadmin for more investigation. For local testing, this issue may indicate that your host's nameservers aren't configured appropriately. To use these commands, you will need to install the [BIND Utilities](http://www.linuxfromscratch.org/blfs/view/svn/basicnet/bind-utils.html) (<http://www.linuxfromscratch.org/blfs/view/svn/basicnet/bind-utils.html>) package.

18. firewall-cmd

Traditionally, firewalls were configured on Linux with the **iptables** command, and

Subscribe now

Get the highlights in your
inbox every week.

X s actually been largely [replaced by nftables](https://opensource.com/blog/2016/10/28/what-comes-after-iptables-its-friend)
. A friendly front-end for [nftables](https://nftables.org/documentation/en-us/red_hat_enterprise_linux/8/html/networking/getting-started-with-configuring-networking)
[documentation/en-us/red_hat_enterprise_linux/8/html/networking/getting-started-with-configuring-networking](https://nftables.org/documentation/en-us/red_hat_enterprise_linux/8/html/networking/getting-started-with-configuring-networking)), and the one that ships with
s **firewall-cmd**. This command helps you set up
traffic, both outgoing and incoming, your computer
up into *zones*, so you can quickly and easily
another, depending on your requirements.

itforward. You use the command and some
number of options, all of which are named in ways that help you almost construct
a human-readable sentence. For instance, to see what zone you're currently in:

```
$ sudo firewall-cmd --get-active-zones`
corp
  interfaces: ens0
dmz
  interfaces: ens1
```

In this example, your computer has two network devices, and one is assigned to
the *corp* zone, while the other is assigned to the *dmz* zone.

To see what each zone permits, you can use the `--list-all` option:

```
$ sudo firewall-cmd --zone corp --list-all
corp
  target: default
  interfaces: ens0
  services: cockpit dhcpv6-client ssh
  ports:
  protocols:
[...]
```

Adding services is just as easy:

```
$ sudo firewall-cmd --add-service http --permanent
```

Subscribe now

X |

Get the highlights in your
inbox every week.

s designed to be intuitive, and it has an extensive
s, plus the ability to write **nft** rules directly. Once
you can [download our firewall-cmd cheat sheet](#)
[et](#)) to help you remember its most important

[:://opensource.com/article/18/7/sysadmin-guide-](https://opensource.com/article/18/7/sysadmin-guide-)

le) enforced on an application host managed by
an enterprise. SELinux provides least-privilege access to processes running on
the host, preventing potentially malicious processes from accessing important
files on the system. In some situations, an application needs to access a specific
file but may throw an error. To check if SELinux blocks the application, use **tail**
and **grep** to look for a "denied" message in the **/var/log/audit** logging.
Otherwise, you can check to see if the box has SELinux enabled by using
sestatus.

```
$ sestatus
SELinux status:                enabled
SELinuxfs mount:              /sys/fs/selinux
SELinux root directory:       /etc/selinux
Loaded policy name:            targeted
Current mode:                  enforcing
Mode from config file:         enforcing
Policy MLS status:             enabled
Policy deny_unknown status:    allowed
Max kernel policy version:     28
```

The output above indicates that the application's host has SELinux enabled. On
your local development environment, you can update SELinux to be more

permissive. If you need help with a remote host, your sysadmin can help you determine the best practice for allowing your application to access the file it needs. If you're interacting with SELinux frequently, [download our SELinux cheat sheet](#) X [-selinux](#)) for quick reference.

Subscribe now

Get the highlights in your
inbox every week.

For testing and debugging, you may forget a variant of the [history command](#) ([/18/6/history-command](#)). It shows the history of commands since the start of the session. You can use **history** to troubleshoot your application. For example, in the course of this article, it shows the various commands used and learned.

```
$ history
 1 clear
 2 df -h
 3 du
```

What if you want to execute a command in your previous history, but you don't want to retype it? Use **!** before the command number to re-execute.

```
95 netstat -tulpn | grep 80
96 clear
97 history
[root@localhost vagrant]# !95
netstat -tulpn | grep 80
tcp6      0      0 :::80          :::*           LISTEN     18310/httpd
```

Adding **!** before the command number you want to execute issues the command again.

Basic commands can enhance your troubleshooting expertise when determining why your application works in one development environment but perhaps not in another. Many sysadmins leverage these commands to debug problems with systems. Understanding some of these useful troubleshooting commands can help you communicate with sysadmins and resolve issues with your application.

This article was originally published in July 2017 and has been updated by the editor

Subscribe now

X

Get the highlights in your
inbox every week.

[admin](#))

author

- As a developer, engineer, and enthusiast of cloud computing automation, Rosemary Wang bridges the technical and cultural frastructure engineers and application developers. Whether it is a structure engineer learn to code or an application developer structure failures, she has fascination for solving intractable e. She interfaces with vendors, clients, startups, and open find creative...

• [More about me \(/users/joatmon08\)](#)

Contributors



[\(/users](#)

[/seth\)](#)

[Seth](#)

[Kenlon](#)

[\(/users](#)

[/seth\)](#)

Recommended reading



Subscribe now

Get the highlights in your inbox every week.

X



[10 lessons from sysadmins adapting to change in 2020](#)

(/article/20/12/



[Manage multiple service instances with systemctl](#)

(/article/20/12/multiple-service-instances-



[Why failure should be normalized and how to do it](#)

(/article/20/11/normalize-

failure?utm_source=opensource&utm_medium=article&utm_campaign=20/11/normalize-failure

[How I use Cockpit for my home's Linux server management](#)

(/article/20/11/cockpit-server-management

[Load balance network traffic with HAProxy](#)

(/article/20/11/load-balancing

31 Comments, [Register \(/user/register\)](#) or [Log in \(/user/login?destination=node/37791\)](#) to post a comment.



[Don Watkins \(/users/don-watkins\)](#) on 26 Jul 2017

Great list. I learned a few more thanks to your wonderful article. Thank you for sharing.



[David Both \(/users/dboth\)](#) on 26 Jul 2017

Excellent article. I learned some new things. Thanks!



Marcos Alano on 26 Jul 2017

I can't figure out the difference between:
`$ curl -I -s database:27017`

Subscribe now

X

Get the highlights in your
inbox every week.

[bcotton](#) on 26 Jul 2017

[ource.com \(https://opensource.com\)](https://opensource.com)` just gives header
the HTML from the page.

| resolve ok not just the first.



Vegan Garo on 26 Jul 2017

I use `chmod u+x`. yep, one can never be too careful



[Ben Cotton \(/users/bcotton\)](/users/bcotton) on 26 Jul 2017

Great article. One `tail` trick that comes in really handy for me is to use `-F`. This follows log files as they get rotated and waits for the file to exist if it doesn't. This makes it really useful for watching daemon logs.



[Nayiotus \(/users/nayiotus\)](/users/nayiotus) on 01 Aug 2017

For the lazier folk (myself included) and if your OS supports it, you can just use `tailf` :)



Howard on 26 Jul 2017

Wonderful article! Very helpful and thoughtful.



[japa \(/users/jsenjali\)](/users/jsenjali) on 26 Jul 2017

you still amaze me.. This is really helpful.

Subscribe now

X

Get the highlights in your
inbox every week.

il 2017

ysadmin' who doesn't know these by finger muscle memory
d.

2017

on 01 Aug 2017



Totally right ! These are all basic "entry level" commands.



Sergio Araujo on 27 Jul 2017

Even for Desktop users, GNU has hundreds of good programs.

youtube-dl can download an entire youtube channel at once

and curl can dowload a range of images/audios at once

curl -O [http://example.com/file\[1-100\].txt](http://example.com/file[1-100].txt) ([http://example.com/file\[1-100\].txt](http://example.com/file[1-100].txt))

I think rsync deserves a mention.

Great article.



Bartosz Fenski on 27 Jul 2017

Instead of netstat (deprecated) sysadmin nowadays should rather know ss command.



Tomas on 28 Jul 2017

How do get the output of "netstat -s" with ss? Please don't say "use ss -s".



dean linux on 28 Jul 2017

... solve DNS if the first did though.

Subscribe now

X

Get the highlights in your
inbox every week.

[ers/joatmon08](#)) on 01 Aug 2017

it out! It seems that the formatting cut out the difference out. The
d from a local desktop while the second is issued from within the
clarification was added to the text.

xample 6. Abusing cats is wrong! :D



Mike Schwager on 28 Sep 2017

Totally right! Demonstrating poor use of command on a tutorial for commands is bad
form! Sad! :-P



[Jose Delarosa \(/users/jdelaros1\)](#) on 31 Jul 2017

Not sure iptables belongs here, but if it does, then firewall-cmd does as well.



[Tamwyn \(/users/tamwyn\)](#) on 28 Sep 2017

Maybe also a mention about nftables would have been useful.



[Maxim Stewart \(/users/raveyoda\)](#) on 01 Aug 2017

I recently resolved an issue with WordPress not installing themes, plugins, etc and it was due
to SELinux.... Good times...good times.



Brad Allison on 01 Aug 2017

I'm surprised no one gave the perl answer to #2:

```
cat test.json | perl -MData::Dumper -MJSON=from_json -ne 'print Dumper(from_json($_))'
```

Subscribe now

X

Get the highlights in your
inbox every week.

ip.

I to install iproute2 to find your IP address instead of just using

like ls, cat, grep, and ps on the list? It makes the list sound like it's
sysadmins but could never get hired as such.



I would add awk to the list. I couldn't live without it. It's much more powerful than most of the basic commands such as grep, if you take the time to learn it as a programming language. In this day and age of heavier languages such as Perl, Python and Ruby, most folks have forgotten how small, fast and powerful awk is. And as a bonus, you will find some version of awk on just about every Unix/Linux variant, including minimal and hardened installs. Perhaps awk deserves another dedicated article. And not just one liners. :-)

Furthermore, every "sysadmin" should know the native package managers such as rpm (yum/zypper) and dpkg (apt). Once they enlighten themselves with those tools, installing software directly from Github or using "configure/make/make install" to blather software all over a system with no governance will seem like a joke. And better yet, making packages graduates you to the next level and impresses your colleagues. :-)



[Jesús Juárez \(/users/jjayala\)](#) on 28 Sep 2017

Absolutely agree.

I use awk a lot combined with grep, cat, ls, find, etc.

Once I wrote a program in awk to parse a binary file, it was easier and ran faster than use a program language. I was impressed.

And like others I think the list has several basic commands, not just sysadmin commands.

I'd add

ssh, scp, sftp

ftp

Subscribe now

X is advanced options)

Get the highlights in your
inbox every week.

ware

: by CURL.



M Matos on 22 Sep 2017

Missing some useful ones like ln, rsync, strace, tac, rev, sed, awk, cut, watch, diff, more/less.
Great beginners list indeed. Cheers



Amar on 22 Oct 2017

Nice list, Rosemary. Commands like 'cat', 'tail', 'ls' are so basic they probably belong under the
category of commands that any "Linux user" should know.
Commands that every sysadmin (even beginners) should know: find, awk, traceroute, tar.
There are others but these come to mind right away. Very useful article, nonetheless.



Thong Ly on 24 Oct 2017

I would add 'less' to the list

Subscribe now

Get the highlights in your inbox every week.

Subscribe to our weekly newsletter

X

Subscribe

[element](#)

Get the highlights in your inbox every week.

Find us:

[Privacy Policy](#) | [Terms of Use](#) | [Contact](#) | [Meet the Team](#) | [Visit opensource.org](#)