Shell Commands

We've put together some of the more frequently used linux shell commands, and organized them by name so you can easily find a command, their description and how to use it. This guide will continue to be updated and should not be considered a complete list of linux shell commands, but commands, we found, often used. If you would like to add to this guide, please email us and let us know. We know that these themselves are bash commands and not actually SSH commands but it is what most Linux newbies are looking for when searching for 'SSH commands'.

Common Linux Shell Commands

ls

List files/directories in a directory, comparable to dir in windows/dos.

```
ls -al
#shows all files (including ones that start with a period), directories, and
details attributes for each file.
```

cd

Change directory ·· cd /usr/local/apache : go to /usr/local/apache/ directory

```
cd ~  #go to your home directory
cd -  #go to the last directory you were in
cd .. #go up a directory cat : print file contents to the screen
```

chmod

Changes file access permissions.

The set of 3 go in this order from left to right: **USER - GROUP - EVERYONE**

```
0 = --- No permission
1 = --x Execute only
2 = -w- Write only
3 = -wx Write and execute
4 = r-- Read only
5 = r-x Read and execute
6 = rw- Read and write
7 = rwx Read, write and execute
```

Usage: chmod numberpermissions filename/directory

```
chmod 000  #No one can access
```

```
chmod 644  #Usually for HTML pages
chmod 755  #Usually for CGI scripts
```

chown

Changes file ownership permissions

The set of 2 go in this order from left to right: **USER - GROUP**

```
chown root myfile.txt      # Changes the owner of the file to root
chown root.root myfile.txt # Changes the owner and group of the file to
root
```

tail

Like cat, but only reads the end of the file.

```
tail /var/log/messages      # see the last 20 (by default) lines of
/var/log/messages
tail -f /var/log/messages   # watch the file continuously, while it's being
updated
tail -200 /var/log/messages # print the last 200 lines of the file to the
screen
```

more/less

like cat, but opens the file one screen at a time rather than all at once

```
more /etc/userdomains # browse through the userdomains file. hit Space to go
to the next page, q to quit
```

vi

```
vi /home/burst/public_html/index.html # edit the index page for the user's
website
```

While in the vi program you can use the following useful commands, you will need to hit SHIFT + : to go into command mode

```
:q! # This force quits the file without saving and exits vi
:w  # This writes the file to disk, saves it
:wq # This saves the file to disk and exists vi
:LINENUMBER : EG :25 # Takes you to line 25 within the file
:$ # Takes you to the last line of the file
```

```
:0 # Takes you to the first line of the file
```

- http://www.tutorialspoint.com/unix/unix-vi-editor.htm
- interactive tutorial

grep

Looks for patterns in files

```
grep root /etc/passwd    # shows all matches of root in /etc/passwd
grep -v root /etc/passwd # shows all lines that do not match root
```

ln

Create's "links" between files and directories

```
ln -s /usr/local/apache/conf/httpd.conf /etc/httpd.conf # Now you can edit
/etc/httpd.conf rather than the original. changes will affect the original,
however you can delete the link and it will not delete the original.
```

last

Shows who logged in and when

```
last -20    # shows only the last 20 logins
last -20 -a # shows last 20 logins, with the hostname in the last field
```

w

shows who is currently logged in and where they are logged in from.

```
who # This also shows who is on the server in an shell
```

netstat

shows all current network connections.

```
netstat -an # shows all connections to the server, the source and
destination ips and ports.
netstat -rn # shows routing table for all ips bound to the server.
```

top

shows live system processes in a nice table, memory information, uptime and other useful info. This is excellent for managing your system processes, resources and ensure everything is working fine and your server isn't bogged down. top then type Shift + M to sort by memory usage or Shift + P to sort by CPU usage

ps

ps is short for process status, which is similar to the top command. It's used to show currently running processes and their PID. A process ID is a unique number that identifies a process, with that you can kill or terminate a running program on your server (see kill command).

```
ps U username # shows processes for a certain user
ps aux # shows all system processes
ps aux --forest # shows all system processes like the above but organizes in
a hierarchy that's very useful!
```

touch

creates an empty file

```
touch /home/burst/public_html/404.html # create an empty file called
404.html in the directory /home/burst/public_html/
```

file

attempts to guess what type of file a file is by looking at it's content.

```
file * # prints out a list of all files/directories in a directory
```

du

shows disk usage.

```
du -sh # shows a summary, in human-readble form, of total disk space used in
the current directory, including subdirectories.
du -sh * # same thing, but for each file and directory. helpful when finding
large files taking up space.
```

wc

word count

```
wc -l filename.txt # tells how many lines are in filename.txt
```

cp

copy a file

```
cp filename filename.backup # copies filename to filename.backup
cp -a /home/burst/new_design/* /home/burst/public_html/ # copies all files,
retaining permissions form one directory to another.
cp -av * ../newdir # Copies all files and directories recurrsively in the
current directory INTO newdir
```

mv

move a file command

```
mv -v oldfilename newfilename # Move a file or directory from oldfilename to
newfilename in a verbose mode
```

rm

delete a file

```
rm -vi filename.txt # deletes filename.txt, will more than likely ask if you
really want to delete it
rm -fv filename.txt # deletes filename.txt, will not ask for confirmation
before deleting.
rm -rfv tmp/ # recursively deletes the directory tmp, and all files in it,
including subdirectories. BE VERY CAREFULL WITH THIS COMMAND!!!
```

tar

creating and Extracting .tar.gz and .tar files

```
tar -zxvf file.tar.gz         # Extracts the file that have been compressed
tar -xvf file.tar             # Extracts the file
tar -cf archive.tar contents/ # Takes everything from contents/ and puts it
into archive.tar
gzip -d filename.gz           # Decompress the file, extract it
```