Table of Contents

Part 2 of 2 - Explore all 232+ Linux commands at dargslan.com/learn/linux-commands
Each command includes syntax, options, practical examples with output, and pro tips.

$ userdel

Intermediate

Delete a user account

userdel removes a user account from the system. It deletes the user entry from /etc/passwd and /etc/shadow, and optionally removes the home directory and mail spool.

userdel requires root privileges. Without -r, it only removes the user account, leaving home directory and files intact. With -r, ...

Options & Flags

`-r`	Remove home directory and mail spool
`-f`	Force removal (even if user is logged in)
`-z`	Remove SELinux user mapping

Practical Examples

Example: Remove user only

```
$ sudo userdel olduser
```

Removes the account but keeps home directory and files.

Example: Remove user and files

```
$ sudo userdel -r olduser
```

Removes the account, home directory, and mail spool.

Example: Force remove logged-in user

```
$ sudo userdel -f baduser
```

Force removes even if the user is currently logged in or has running processes.

Example: Full cleanup procedure

```
$ sudo killall -u olduser; sudo userdel -r olduser; sudo find /tmp -user olduser -delete
```

Kills processes, removes user, and cleans up temp files.

Example: Check before deleting

```
$ ps -u olduser; find /var/www -user olduser
```

Check for running processes and owned files before deletion.

Tips & Best Practices

Warning: Kill processes first: userdel may fail if the user has running processes. Kill them first: sudo killall -u username.

Pro Tip: Backup before -r: userdel -r permanently deletes the home directory. Back up important data first: tar czf /backup/user.tar.gz /home/user.

Note: Files remain without -r: Without -r, home directory files remain as orphaned (owned by deleted UID). Clean up with find / -nouser.

$ usermod

Intermediate

Modify a user account

usermod modifies an existing user account. It can change the username, home directory, shell, groups, expiration date, and other account properties.

usermod requires root privileges. The most common uses are adding users to groups (-aG), changing the login shell (-s), locking/unlocking accounts ...

Options & Flags

`-aG`	Append user to supplementary groups
`-s`	Change login shell
`-d`	Change home directory
`-l`	Change login name
`-L`	Lock account (disable login)
`-U`	Unlock account
`-e`	Set account expiration date
`-g`	Change primary group
`-c`	Change comment (full name)

Practical Examples

Example: Add to group

```
$ sudo usermod -aG sudo jdoe
```

Adds user to sudo group while keeping all existing groups.

Example: Add to multiple groups

```
$ sudo usermod -aG docker,www-data jdoe
```

Adds user to docker and www-data groups.

Example: Change shell

```
$ sudo usermod -s /bin/zsh jdoe
```

Changes the default login shell to zsh.

Example: Lock account

```
$ sudo usermod -L baduser
```

Locks the account - user cannot log in. Does not kill existing sessions.

Example: Change home directory

```
$ sudo usermod -d /home/newhome -m jdoe
```

Changes home directory and moves existing files (-m) to the new location.

Tips & Best Practices

Warning: Always use -a with -G: usermod -G sudo user REPLACES all groups with just sudo. Always use -aG to APPEND: usermod -aG sudo user.

Pro Tip: Verify group changes: After usermod -aG, verify with groups username or id username. User must log out/in for changes to take effect.

Note: Lock vs delete: usermod -L locks the account (reversible with -U). userdel removes it permanently. Lock accounts for temporary disabling.

$ visudo

Intermediate

Safely edit the sudoers file

visudo safely edits the /etc/sudoers file. It locks the file against simultaneous edits, validates syntax before saving, and prevents saving a broken sudoers file that could lock everyone out of sudo.

Never edit /etc/sudoers directly with a regular text editor - a syntax error can prevent all su...

Options & Flags

`-c`	Check syntax only (do not edit)
`-f`	Edit a specific sudoers file
`-s`	Strict mode (more rigorous checking)

Practical Examples

Example: Edit sudoers

```
$ sudo visudo
```

Opens /etc/sudoers for safe editing with syntax validation.

Example: Check syntax

```
$ sudo visudo -c
/etc/sudoers: parsed OK
```

Validates the sudoers file without opening for editing.

Example: Edit custom sudoers file

```
$ sudo visudo -f /etc/sudoers.d/developers
```

Edits a drop-in sudoers file with syntax checking.

Example: Grant sudo to user

```
$ # Add this line in visudo:\njdoe ALL=(ALL:ALL) ALL
```

Grants full sudo access to user jdoe.

Example: Allow specific command

```
$ # Add this line:\ndeployer ALL=(ALL) NOPASSWD: /usr/bin/systemctl restart nginx
```

Allows deployer to restart nginx without password.

Tips & Best Practices

Warning: Never use nano/vim directly: Always use visudo to edit sudoers. A syntax error in sudoers can permanently lock out sudo access.

Pro Tip: Use /etc/sudoers.d/ directory: Create files in /etc/sudoers.d/ instead of editing /etc/sudoers directly. Easier to manage and less risky.

Note: Change editor: Set your editor: export EDITOR=nano, then visudo uses nano instead of vim.

$ w

Beginner

Show who is logged in and what they are doing

w displays who is logged in and what they are doing. It shows the username, terminal, remote host, login time, idle time, CPU usage, and the current command for each logged-in user.

w combines the functionality of who (who is logged in) and uptime (system load). It is a quick way to see system a...

Options & Flags

`-h`	Do not print header
`-s`	Short format
`-f`	Toggle showing FROM field
`-u`	Ignore current process username
`username`	Show only specific user

Practical Examples

Example: Show all users

```
$ w
USER   TTY   FROM            IDLE   WHAT
admin  pts/0  192.168.1.50    0.00s  w
jdoe   pts/1  10.0.0.100      5:30   vim deploy.sh
```

Shows who is logged in, their terminal, source, idle time, and current command.

Example: Short format

```
$ w -s
```

Shows abbreviated output without login time and CPU columns.

Example: Specific user

```
$ w admin
```

Shows sessions only for the admin user.

Example: Without header

```
$ w -h
```

Shows logged-in users without the header line - useful in scripts.

Example: Count logged-in users

```
$ w -h | wc -l
3
```

Counts the number of active user sessions.

Tips & Best Practices

Pro Tip: Quick system overview: w shows users, load average, and uptime in one command - perfect for a quick check when connecting to a server.

Note: IDLE column: IDLE shows how long since the user last typed. High idle times may indicate forgotten sessions or hanging connections.

Warning: FROM may be empty: The FROM field shows the source IP/hostname. It may be empty for console logins or local sessions.

$ whoami

Beginner

Print the current username

whoami prints the effective username of the current user. It is the simplest way to check what user you are currently operating as, especially after su or sudo.

whoami is equivalent to id -un but more memorable. It is commonly used in scripts to verify the running user, check for root, and build...

Options & Flags

`(no options)`	Print current effective username

Practical Examples

Example: Check current user

```
$ whoami
user
```

Shows the effective username.

Example: After sudo

```
$ sudo whoami
root
```

Shows that sudo runs commands as root.

Example: In a script

```
$ if [ "$(whoami)" != "root" ]; then echo "Run as root!"; exit 1; fi
```

Checks that a script is running as root.

Example: Build user path

```
$ CONFIG_DIR="/home/$(whoami)/.config/myapp"
```

Creates a user-specific configuration path.

Example: Compare with logname

```
$ echo "Effective: $(whoami), Login: $(logname)"
Effective: root, Login: admin
```

Shows both effective user and original login user.

Tips & Best Practices

Pro Tip: Script root check: Use [[$(whoami) == "root"]] or [[$EUID -eq 0]] to check for root in scripts. $EUID is faster.

Note: whoami vs who am i: whoami shows effective user. who am i shows the original login user (from utmp). They differ after su/sudo.

Warning: EUID is more reliable: In scripts, $EUID (effective UID) is more reliable and faster than calling whoami.