

usermod Cheatsheet

By Dejan Panovski • Updated on Oct 11, 2026

Quick reference for modifying Linux user accounts with `usermod` and managing groups with `groupadd`, `gpasswd`, and `groupdel`

The `usermod` command changes existing user accounts in Linux, from group membership and login shell to home directory, UID, and account locking. This cheatsheet also covers the group commands you use alongside it.

Basic Syntax

Core `usermod` command forms.

<code>sudo usermod [options] username</code>	General syntax
<code>sudo usermod -c "Full Name" username</code>	Change the GECOS/comment field
<code>id username</code>	Show UID, GID, and group membership
<code>getent passwd username</code>	Show the user's account entry
<code>usermod --help</code>	List all available options

Supplementary Groups

Add or remove secondary group membership.

<code>sudo usermod -aG docker username</code>	Add user to one group
<code>sudo usermod -aG docker,sudo username</code>	Add user to multiple groups
<code>sudo usermod -G docker username</code>	Replace all supplementary groups with <code>docker</code>
<code>sudo usermod -rG docker username</code>	Remove user from one group
<code>sudo gpasswd -d username docker</code>	Remove user from one group (alternative)

Primary Group

Change the group assigned to new files the user creates.

<code>sudo usermod -g developers username</code>	Set primary group to <code>developers</code>
<code>id -gn username</code>	Show the user's primary group
<code>newgrp developers</code>	Start a shell with <code>developers</code> as the active group
<code>sudo chgrp developers file</code>	Change group ownership of a file
<code>sudo chgrp -R developers dir/</code>	Change group ownership recursively

Home Directory and Shell

Move the home directory or change the login shell.

<code>sudo usermod -d /srv/username -m username</code>	Change home path and move its contents
<code>sudo usermod -d /srv/username username</code>	Change home path only (old files stay in place)
<code>sudo usermod -s /bin/zsh username</code>	Change login shell
<code>sudo usermod -s /usr/sbin/nologin username</code>	Disable interactive login
<code>cat /etc/shells</code>	List valid login shells

Rename and UID

Change the login name or numeric user ID. Record the old UID before using `-u` , and review the `find` results before running `chown` .

<code>sudo usermod -l newname oldname</code>	Rename the user account
<code>sudo usermod -l newname -d /home/newname -m oldname</code>	Rename user and move home directory
<code>sudo groupmod -n newname oldname</code>	Rename the matching user private group
<code>old_uid=\$(id -u username)</code>	Record the current UID before changing it
<code>sudo usermod -u 1050 username</code>	Change UID (files in home are updated)
<code>sudo find /srv -xdev -uid "\$old_uid" -print</code>	Preview files still owned by the old UID under <code>/srv</code>
<code>sudo find /srv -xdev -uid "\$old_uid" -exec chown -h username {} +</code>	Reassign old-UID files under <code>/srv</code> to the user

Replace `/srv` with the directory or mount point you need to repair. The `-xdev` option stays on that filesystem, so scan other mounts separately. Use the same shell session so `$old_uid` remains available.

Lock, Unlock, and Expiry

Lock or unlock password authentication and set an account expiry date.

<code>sudo usermod -L username</code>	Lock the account password
<code>sudo usermod -U username</code>	Unlock the account password
<code>sudo usermod -e 2026-12-31 username</code>	Expire the account on a date
<code>sudo usermod -e "" username</code>	Remove the expiry date
<code>sudo chage -l username</code>	Show expiry and password aging details

The `-L` option blocks password authentication only; SSH keys may still work. The `-U` option does not remove an account expiry date; use `-e ""` separately when needed.

Manage Groups

Create, rename, and delete groups.

<code>sudo groupadd developers</code>	Create a group
<code>sudo groupadd -g 1500 developers</code>	Create a group with a specific GID
<code>sudo groupadd -r appgroup</code>	Create a system group
<code>sudo groupmod -n devs developers</code>	Rename a group
<code>sudo groupdel developers</code>	Delete a group

List Users and Groups

Check who belongs where.

<code>groups username</code>	List groups a user belongs to
<code>getent group developers</code>	Show a group entry and explicitly listed members
<code>getent group</code>	List all groups
<code>getent passwd</code>	List all user accounts
<code>cut -d: -f1 /etc/group</code>	List local group names only

Primary group membership is recorded in the `passwd` database, so primary members may be absent from `getent group` output. Use `id username` to check a user's primary and supplementary groups.

Troubleshooting

Quick checks for common `usermod` problems.

<code>user username</code> is currently used by <code>process</code>	Log the user out and stop their processes before <code>-l</code> , <code>-u</code> , or <code>-d</code>
New group not active	Log out and back in, or run <code>newgrp groupname</code>
User lost other groups	<code>-G</code> without <code>-a</code> replaces the list; re-add the missing groups with <code>-aG</code>
group 'name' does not exist	Create it first with <code>groupadd</code>
unlocking the user's password would result in a passwordless account	Set a password with <code>passwd</code> instead of using <code>-U</code>

Related Guides

Use these guides for full user and group management tasks.

[usermod Command in Linux](#)

Full usermod tutorial with examples

[How to Add a User to a Group in Linux](#)

Manage supplementary and primary groups

[How to Create Groups in Linux](#)

Create groups with groupadd

[groupdel Command in Linux: Delete a Group](#)

Remove groups with groupdel

[How to List Groups in Linux](#)

Show group membership

[useradd Cheatsheet](#)

Create new user accounts