

pkill Cheatsheet

By Dejan Panovski • Updated on Mar 4, 2026 • [Download PDF](#)

Quick reference for finding and terminating processes by name, user, and pattern with `pkill` in Linux

The `pkill` command sends signals to processes that match a name, user, terminal, or full command pattern. This cheatsheet covers safe matching patterns, common signals, and practical process-control examples.

Basic Syntax

Core `pkill` command forms.

<code>pkill process_name</code>	Send SIGTERM to matching process names
<code>pkill -f "pattern"</code>	Match against full command line
<code>pkill -u username process_name</code>	Match only processes owned by a user
<code>pkill -x process_name</code>	Match exact process name only
<code>pkill -l process_name</code>	Kill matching processes and print their name and PID

Common Signals

Frequently used signals with `pkill`.

<code>pkill -15 process_name</code>	Graceful stop (SIGTERM , default)
<code>pkill -9 process_name</code>	Force kill (SIGKILL)
<code>pkill -HUP process_name</code>	Reload/reopen config for daemons
<code>pkill -INT process_name</code>	Interrupt process (SIGINT)
<code>pkill -USR1 process_name</code>	Send user-defined signal 1

Match Controls

Limit matches to avoid terminating the wrong process.

<code>killall -x nginx</code>	Kill only exact nginx process name
<code>killall -f "python3 app.py"</code>	Match a specific command string
<code>killall -u deploy -x node</code>	Match exact node only for user deploy
<code>killall -t pts/2</code>	Match processes attached to terminal pts/2
<code>killall -P 1234</code>	Match child processes of PID 1234

Safer Workflow

Preview targets before signaling processes.

<code>pgrep -a nginx</code>	Preview matching processes and command lines
<code>pgrep -afu deploy node</code>	Preview user-scoped full-command matches
<code>pgrep -f "python3 app.py"</code>	Confirm full-pattern matches first
<code>killall -x nginx</code>	Execute only after preview validation
<code>echo \$?</code>	Check exit code (0 match found, 1 none found)

Service and App Examples

Practical process control patterns.

<code>killall -HUP nginx</code>	Ask Nginx master process to reload
<code>killall -u www-data -x php-fpm</code>	Stop php-fpm workers for one user
<code>killall -f "gunicorn: worker"</code>	Signal Gunicorn worker processes
<code>killall -f "node server.js"</code>	Stop a specific Node.js app instance
<code>killall -x firefox</code>	Close all Firefox processes for current user

Troubleshooting

Quick checks for common `pkill` issues.

Nothing happened	Verify matches with <code>pgrep -a</code> using the same pattern
Wrong process was terminated	Use <code>-X</code> or stricter <code>-f</code> pattern matching
Operation not permitted	Use <code>sudo</code> or run as the process owner
Process did not stop	Try <code>SIGTERM</code> first, then escalate to <code>-9</code> only if needed
Script fails when nothing matches	Handle exit code <code>1</code> as a non-match condition

Related Guides

Use these guides for deeper signal and process-management workflows.

pkill Command in Linux	Full <code>pkill</code> guide with examples
kill Command in Linux	Send signals by PID
pgrep Command in Linux	Search processes by name and pattern
ps Command in Linux	Inspect current process list
top Command in Linux	Monitor processes in real time