sudo Run as another user

CONCEPTS

sudo runs one command with another identity (root by default) if the sudoers rules allow it, and asks for your own password, not root's.

You run sudo command + options
↓ checks rules
/etc/sudoers who, where, as whom, what
↓ allowed?
Password cached for a few minutes
↓ runs
Command as target user

Redirections happen in your shell before sudo starts, so sudo echo x > /etc/f still writes as you.

RUNNING COMMANDS

`sudo <command>`
Run a command as root

`sudo -u <user> <command>`
Run as another user

`sudo -g <group> <command>`
Run with another primary group

`sudo -u <user> -g <group> <command>`
Set user and group

`sudo -u '#<uid>' <command>`
Target a user by numeric UID

`sudo -n <command>`
Fail instead of asking for a password

`sudo !!`
Repeat the last command with sudo

SHELLS

`sudo -i`
Root login shell with root's environment

`sudo -s`
Root shell keeping your environment

`sudo -iu <user>`
Login shell as another user

`sudo -H <command>`
Set HOME to the target user's home

SHELLS (cont.)

Prefer one-off sudo commands over a root shell: every command stays logged and a typo does less damage.

PRIVILEGES AND PASSWORD CACHE

`sudo -l`
List what you may run here

`sudo -l <command>`
Check if a command is allowed

`sudo -l -U <user>`
List another user's privileges

`sudo -v`
Refresh the cache without running anything

`sudo -k`
Invalidate the cache for this session

`sudo -K`
Remove all cached credentials

`sudo -V`
Show sudo and plugin versions

ENVIRONMENT

`sudo -E <command>`
Keep your environment variables

`sudo <VAR>=<value> <command>`
Set a variable for one command

`sudo env <VAR>=<value> <command>`
Set a variable via env

PATH is replaced by secure_path on many systems, so tools in your own PATH may not be found under sudo.

EDITING FILES

`sudo -e <file>`
Edit a copy and write it back as root

`sudoedit <file>`
Same as sudo -e

`SUDO_EDITOR=<editor> sudoedit <file>`
Pick the editor for sudoedit

sudoedit runs your editor as you, not root, so editor plugins and config are safe to use.

REDIRECTION AND PIPES

`echo '<text>' | sudo tee <file>`
Write a root-owned file

`echo '<text>' | sudo tee -a <file>`
Append to a root-owned file

`echo '<text>' | sudo tee <file> >/dev/null`
Write without echoing to the screen

`sudo sh -c '<command> > <file>'`
Redirect inside a root shell

`sudo bash -c '<command1> && <command2>'`
Run a compound command as root

SUDOERS RULES

`sudo visudo`
Edit /etc/sudoers with a syntax check

`sudo visudo -f /etc/sudoers.d/<name>`
Edit a drop-in file

`sudo visudo -c`
Check the syntax only

Rule: who where = (as whom) what

Part	Example	Meaning
`user or %group`	`alice, %admin`	who may run it
`host`	`ALL`	on which machine
`(runas)`	`(root), (ALL)`	as which user
`tags: commands`	`NOPASSWD: /usr/bin/apt`	what, and options

`alice ALL=(ALL:ALL) ALL`
Full rights for a user

`%wheel ALL=(ALL) ALL`
Full rights for a group

`alice ALL=(root) NOPASSWD: /usr/bin/systemctl restart <svc>`
One command without a password

Never edit /etc/sudoers with a plain editor: one syntax error can lock out every sudo user. Keep a root shell open while testing.