

LINUX TERMINAL: PIPES TO CLOUD

CloudComputingTutor
cloudcomputingtutor.com

Part 2 cheat sheet — pipes & redirection, text tools, processes, scripting, SSH

PIPES & REDIRECTION

<code>cmd1 cmd2</code>	chain: stdout feeds stdin (LEGO)
<code>ls wc -l</code>	count files here
<code>> >></code>	to file: overwrite / append
<code>sort < in.txt</code>	file as input
<code>cmd 2> err.log</code>	errors only (stream 2)
<code>cmd > out.txt 2>&1</code>	both streams, one file
<code>cmd > /dev/null 2>&1</code>	the black hole: silence all

TEXT TOOLS: GREP - SED - AWK

<code>grep -i "error" log</code>	search, ignore case
<code>grep -r "TODO" dir/</code>	whole directory tree
<code>grep -v x grep -n x</code>	invert match / line numbers
<code>tail -f log grep err</code>	LIVE filtered error feed
<code>sed 's/old/new/g' f</code>	replace all (prints; > to keep)
<code>awk '{print \$1}' f</code>	pull column 1 (-F, for CSV)

PROCESSES

<code>ps aux grep name</code>	find a process + its PID
<code>top</code>	live view (M mem, P cpu, q quit)
<code>kill PID</code>	polite stop, SIGTERM. Default.
<code>kill -9 PID</code>	fire axe. LAST resort.
<code>cmd & jobs fg</code>	background / list / bring back
<code>nohup cmd &</code>	keeps running after logout

FIRST SCRIPT (backup.sh)

<code>#!/bin/bash</code>	line 1 — the shebang
<code>chmod u+x s.sh; ./s.sh</code>	make runnable, then run
<code>NAME="x"; echo \$NAME</code>	variable — NO spaces at =
<code>TODAY=\$(date)</code>	command output into variable
<code>if [-f file]; then</code>	spaces INSIDE [] required
<code>for i in {1..5}; do</code>	loop; fn(){ }, \$1 = argument

ENVIRONMENT

<code>echo \$HOME</code>	<code>env</code>	one variable / all of them
<code>PATH</code>		why you see 'command not found'
<code>export VAR=value</code>		set for this session
<code>~/.bashrc + source</code>		make permanent, then reload

PACKAGES

<code>sudo apt update</code>	refresh catalog (installs nothing)
<code>sudo apt upgrade / install x</code>	now upgrade / install
<code>brew install x</code>	macOS — get it at brew.sh

SSH — THE BRIDGE TO THE CLOUD

<code>ssh user@server</code>	your terminal, their machine
<code>ssh -p 2222 -i ~/.ssh/k</code>	custom port / choose key
<code>scp f.txt u@s:/path/</code>	copy up (flip order to pull)
<code>ssh-keygen -t ed25519</code>	make key pair (public = padlock)
<code>ssh-copy-id u@s</code>	install padlock: no more passwords

!! PRIVATE KEY DISCIPLINE

`chmod 600 ~/.ssh/id_ed25519`
Never leaves your machine — SSH refuses looser keys.

15-MINUTE PRACTICE DRILL (do this daily for two weeks)

1. `ls /etc | wc -l`, then `ls /etc > list.txt`
2. add alias `ll='ls -la' + source ~/.bashrc`
3. `ps aux | grep bash` (your shell's PID)
4. `sleep 300 &    jobs    kill %1` (politely!)
5. `backup.sh`: 6 lines, `chmod u+x, ./backup.sh`
6. `ssh-keygen -t ed25519, ls -l ~/.ssh` (600!)

Type every command — muscle memory lives in fingers, not eyes. Day 14: your first ssh into a real server.

Interview signal: pipes + a script + SSH keys say "I automate and operate" — the cloud-engineer baseline.

Next: the AWS track — ssh into a server YOU launched · free course at cloudcomputingtutor.com