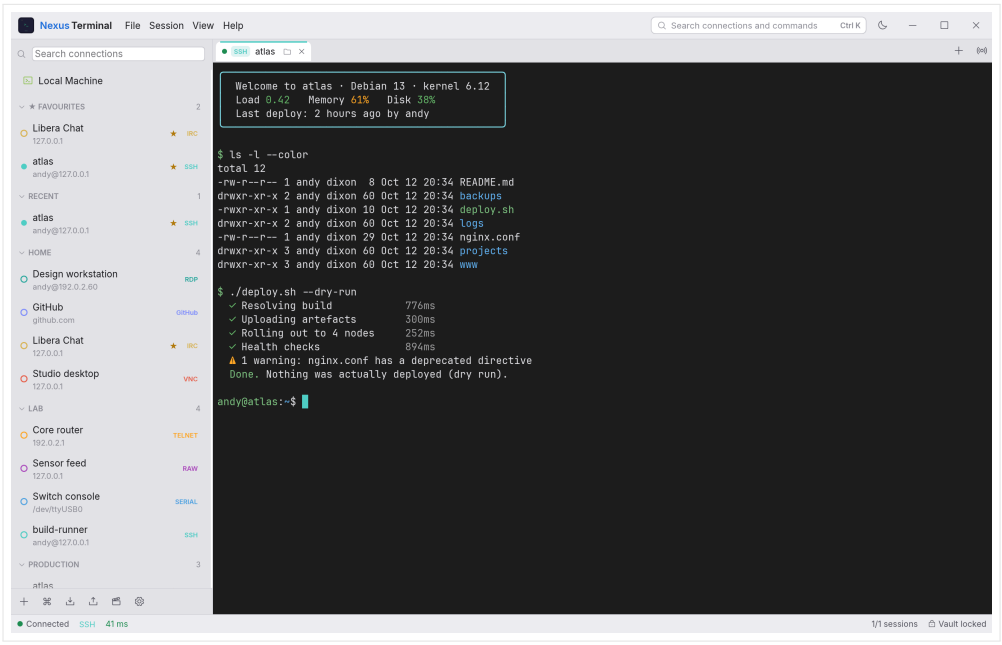
nexus terminal.

The User Guide

Every terminal you need. One window.
SSH, Telnet, RAW, Serial, VNC, RDP, IRC and GitHub.



Nexus Terminal puts every remote session in one native window: SSH with keys, jump hosts and 2FA, serial consoles, remote desktops, IRC and GitHub, with your passwords kept in the OS keychain or an encrypted vault. **Read in any order:** every chapter opens with a **TL;DR** and ends with a **Done when** check, so you always know when you can stop.

How to use this guide

TL;DR

You do not need to read this in order. Every chapter starts with a box like this one that says what the chapter is for, and how long it takes. Read the box; if it is what you need, carry on; if not, skip the chapter.

I just want to…

…get connected to something *right now*	Five-minute start, p. 8
…work out what all the buttons do	The grand tour, p. 13
…SSH into a server (keys, jump hosts, 2FA)	SSH, p. 19
…talk to a router or Arduino over a USB cable	Serial, p. 22
…copy files to and from a server	Files, p. 44
…type the same thing into ten servers at once	Never type it twice, p. 41
…keep my passwords safe	Passwords & the vault, p. 49
…keep two computers in step	Moving house, p. 60
…fix something that’s broken	When things go wrong, p. 64
…just see the keyboard shortcuts	The cheat sheet, p. 68

The key, so you can scan

Marker	Means
`Ctrl / ⌘` + `K`	A key to press.
Settings > Security	A menu path, or a button to click.
`~/.ssh/config`	Something to type, or a file name.
TL;DR	The chapter in a few lines. Read only this if you are in a hurry.
	Roughly how long the chapter takes, and what you can do afterwards.
Quick win	Something you can try in under a minute.
Pro tip	A trick that saves you time later.
Watch out	A thing that might trip you up.
Careful	Could lose data or weaken security. Read every one.
Side quest	Interesting, but not needed to get the job done.
Done when	How you know you have finished the chapter.

Words like **New Connection** are labels you will see in Nexus Terminal. `Ctrl / ⌘` means `Ctrl` on Windows and Linux, `⌘` (Command) on a Mac, so `Ctrl / ⌘` + `K` is `Ctrl`+`K` or `⌘`+`K` depending on your machine.

Contents

How to use this guide 1

Get going

1 **Five-minute start** 8

Install it 8

Open it 8

Make your first connection 9

Two shortcuts for impatient people 10

The one shortcut to rule them all 11

2 **The grand tour** 13

Three looks 14

3 **Profiles: your address book** 15

Create, edit, copy 15

Favourites, groups and Recent 16

Do lots at once 16

“It’s already open…” 17

Connect to all the things

4 **SSH: the big one** 19

The two-minute version 19

How it logs in 20

Jump hosts: getting through the bastion 20

Proxies 20

2FA without the phone faff 20

Do things automatically on login 21

Keeping it alive 21

5 Serial: cables and consoles 22

Setting it up 23

While you're connected 23

6 Telnet and RAW TCP 24

Telnet 24

RAW TCP 24

7 VNC remote desktops 26

The toolbar 27

Tunnel it 27

8 RDP: Windows desktops 28

Setting it up 29

While you're connected 29

Getting through 29

9 IRC 30

10 GitHub 32

11 Your own computer 34

Work faster

12 Tabs and split panes 36

Tabs 36

Split panes 36

13 Terminal superpowers 38

Keyboard and mouse 38

The right-click menu 38

Shell integration 39

Pictures in the terminal 39

Find 40

When the connection drops 40

14 Never type it twice **41**

Snippets: `Ctrl / ⌘` + `Shift` + `S` 41

Broadcast: `Ctrl / ⌘` + `Shift` + `B` 42

Triggers: let it watch for you 42

15 Files: SFTP and drag-and-drop **44**

The SFTP browser 44

Drag and drop 45

16 Port forwards **46**

17 Recordings **47**

Stay safe

18 Passwords and the vault **49**

Where secrets live 49

Setting up the vault 49

Three ways to handle each password 50

Seeing and changing what's saved 50

19 Host keys: is it really that server? **52**

Certificate authorities 53

20 SSH keys **54**

Make it yours

21 Settings **56**

Updates 57

22 Import and export **58**

23 Moving house **60**

Profile sync 60

One-off moves 61

Where everything lives 61

When things go wrong

24 Troubleshooting, by symptom 64

Connecting . . . 64
Serial . . . 65
Remote desktops . . . 65
Files . . . 65
The terminal itself . . . 66
Still stuck? . . . 66

The extras

25 Keyboard cheat sheet 68

26 Command-line options 69

27 Glossary 70

PART 1

Get going

From zero to connected before your tea goes cold.

- Five-minute start
- The grand tour
- Profiles: your address book

CHAPTER 1

Five-minute start

TL;DR

Install it, open it, press `Ctrl / ⌘` + `N`, fill in a host, hit **Create & Connect**. Done. Lost? Press `Ctrl / ⌘` + `K` and type what you want.

5 minutes · You'll be connected to something and know the one shortcut that matters

1.1 Install it

Grab the zip for your machine from `uncomplex.co.uk/nexus`. They're named `nexus-terminal-<version>-<os>-<arch>.zip`, so a modern Mac wants `macos-aarch64` and most PCs want `windows-x86_64`.

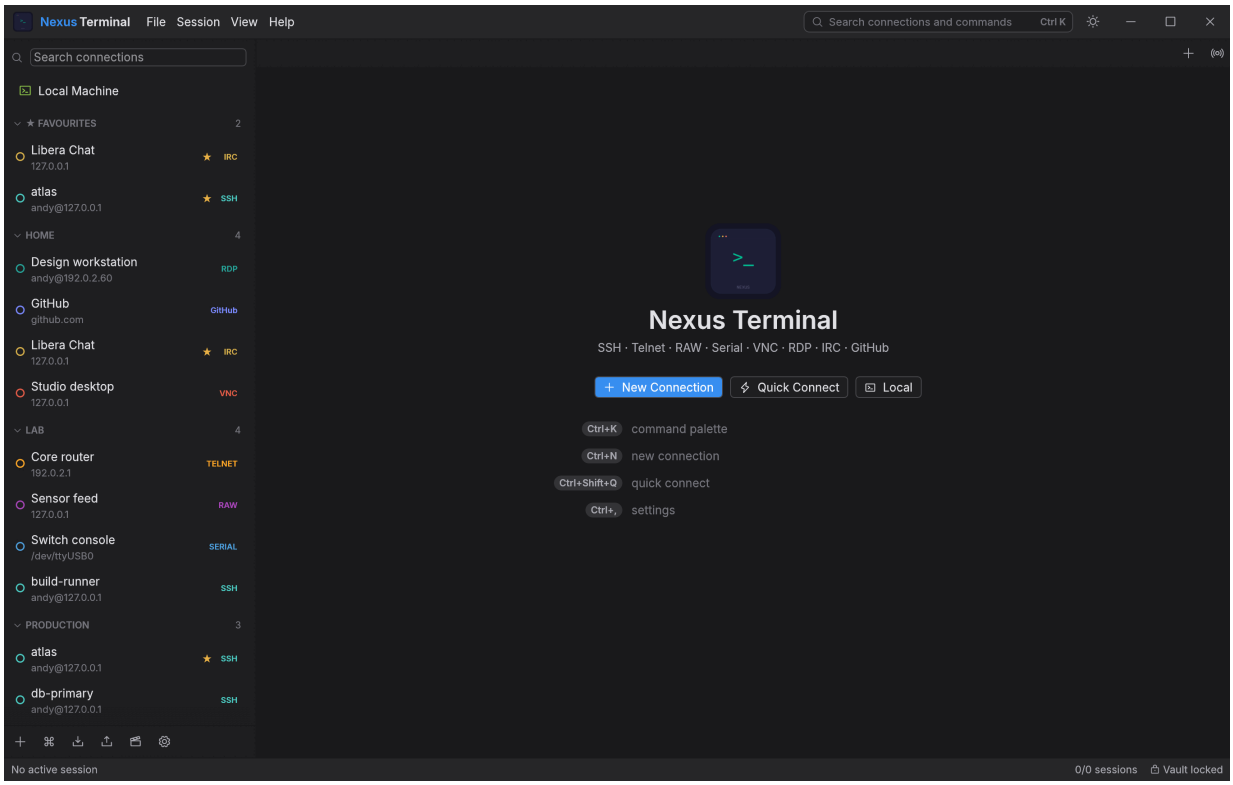
- **Windows** Unzip it and run `nexus-terminal.exe`. No installer, no admin rights, nothing in the registry. Pop it somewhere sensible like `C:\Tools` and pin it to the taskbar.
- **macOS** Unzip it and drag `NexusTerminal.app` into **Applications**. If macOS grumbles that it can't check the developer, right-click the app, choose **Open**, then **Open** again. You only have to do that once.
- **Linux** Unzip it and run `./install.sh`. It copies `nexus-terminal` into `/.local/bin` and adds a launcher entry with the icon, so it turns up in your app menu. (Make sure `/.local/bin` is on your `PATH`.)

Build it yourself · optional; skip it if you are in a hurry

It's a single Rust program. With Rust installed: `cd rust && cargo run –release`. The finished binary lands in `rust/target/release/nexus-terminal`. The build guide in `docs/build-and-release.md` has the full story.

1.2 Open it

The first time you open Nexus Terminal you get a calm, empty welcome screen. Three buttons, four shortcuts. That's it.



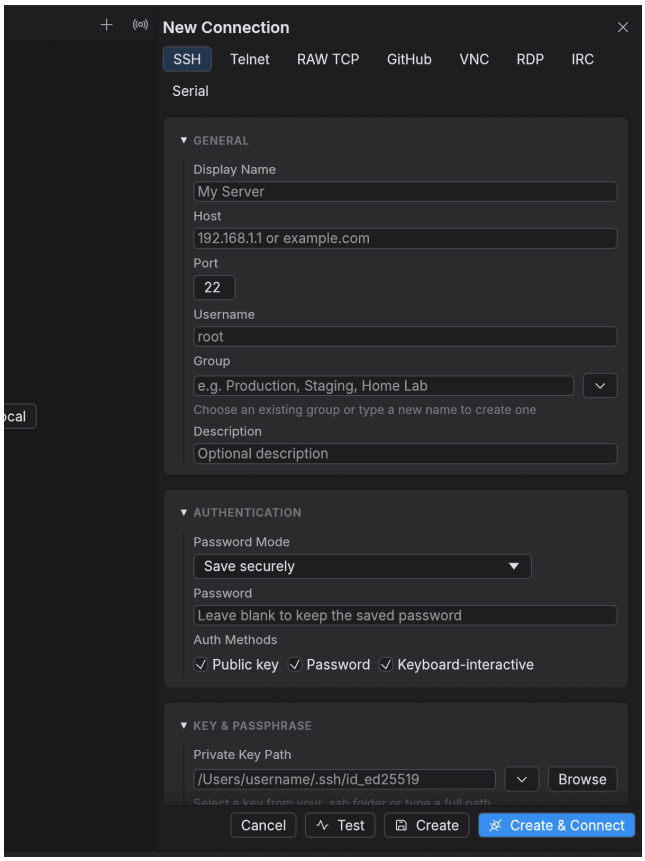
The welcome screen. Your saved connections will live down the left; here there are a few already.

> ⚠ **Got more than one computer?** Point **Profile sync** at a shared folder and your connections follow you around. See Moving house on p. 60.

1.3 Make your first connection

1. Click **New Connection**, or press `Ctrl` / `⌘` + `N`.
2. Pick the protocol along the top: `SSH`, `Telnet`, `RAW TCP`, `GitHub`, `VNC`, `RDP`, `IRC` or `Serial`.
3. Fill in the **General** bit: a **Display Name** you'll recognise, the **Host**, the **Port** and your **Username**.
4. Click **Create & Connect**.

That's genuinely it. A tab opens, you log in, and next time it's one click in the sidebar.



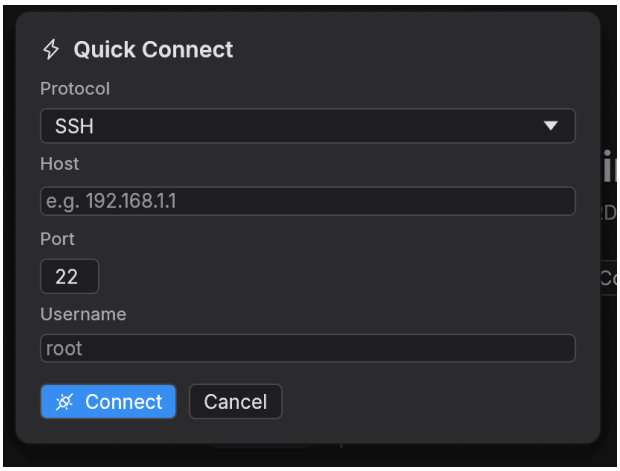
A brand-new SSH profile. Only the top few fields are needed to get going.

> ▷ **Quick win**
>
> Click **Test** before you save. Nexus Terminal checks the host and port are reachable and tells you the round-trip time. If that fails, no amount of password-fiddling will help, so it's a great first check.

1.4 Two shortcuts for impatient people

Quick Connect

Press `Ctrl / ⌘` + `Shift` + `Q`. For a one-off session you don't want to save. Pick SSH, Telnet, RAW, IRC or Serial, type the host (or serial device), hit `Enter`. Nothing gets saved.



Quick Connect. In, out, nobody gets hurt.

Local terminal

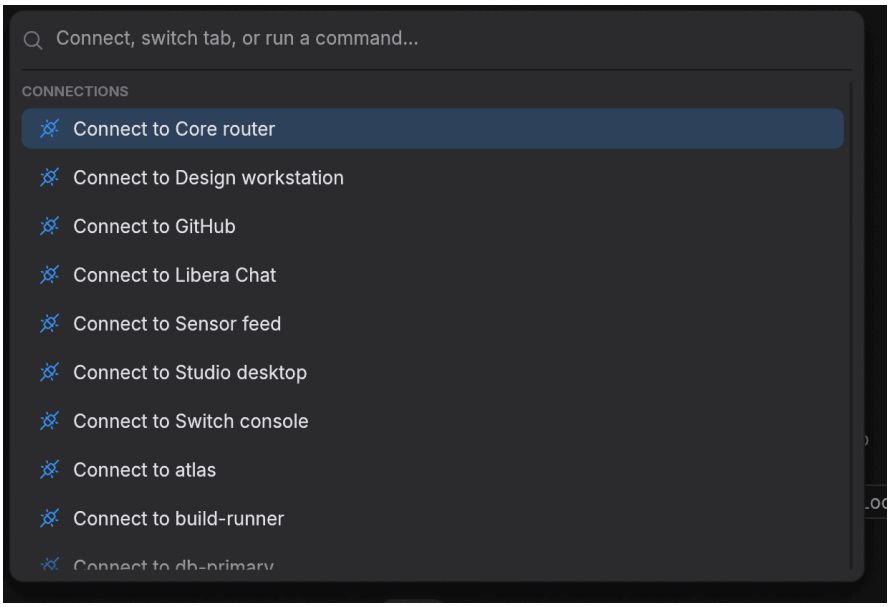
Click **Local Machine** at the top of the sidebar (or **Local** on the welcome screen) for a shell on *this* computer. When you type `exit`, the tab says `[Exited - press enter to restart]`. Press `Enter` and you've got a fresh shell in the same tab.

1.5 The one shortcut to rule them all

`Ctrl / ⌘` + `K`

The command palette. If you remember nothing else, remember this.

Type a few letters of *anything*: a connection name, a tab, "theme", "split", "lock", "snippets". Use `↑` `↓` to pick, `Enter` to go, `Esc` to bail. It's fuzzy, so "atl" finds *atlas* and "sft" finds the *SFTP Browser*.



The command palette: every connection, tab and action in one searchable list.

Your five-minute checklist

- ☐ Installed and opened Nexus Terminal
- ☐ Opened a **Local Machine** tab and typed `exit`, then `Enter` to get it back
- ☐ Made a profile with `Ctrl / ⌘` + `N` and clicked **Test**
- ☐ Connected to it
- ☐ Opened the palette with `Ctrl / ⌘` + `K` and found something by typing three letters

All ticked? You're officially more productive than you were five minutes ago.

Done when you are connected to something, and `Ctrl / ⌘` + `K` finds whatever you type.

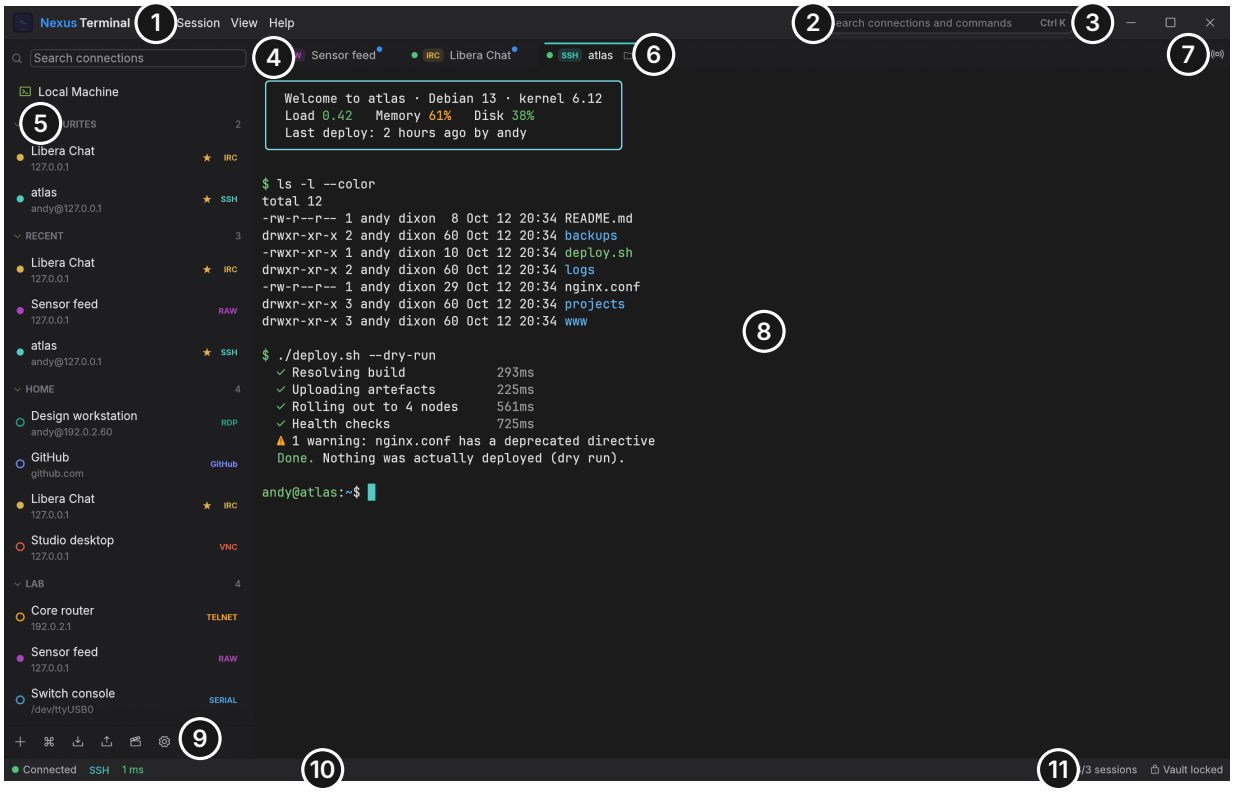
CHAPTER 2

The grand tour

TL;DR

Connections on the left. Sessions in tabs across the top. The terminal in the middle. Status along the bottom. Everything else is one `Ctrl / ⌘` + `K` away.

4 minutes · You'll know what every bit of the window is for



1 Menus **File**, **Session**, **View** and **Help**. Drag the title bar to move the window; double-click it to maximise.

2 Palette The search box is the command palette. Click it or press `Ctrl / ⌘` + `K`.

3 Light/dark Flip between the dark and light themes in one click.

4 Search Fuzzy-search your connections by name, host, tag or protocol.

5 Sections **Local Machine**, **Favourites ★**, **Recent** (your last five), then one section per group. Click a header to fold it away.

6 Tabs One per session, with a coloured state dot, protocol badge and close button. SSH tabs get a little 🗁 for files. A blue dot means "new output while you weren't looking".

7 New & broadcast **+** opens a new connection; (((•))) toggles broadcast mode.

8 Workspace Your terminal (or IRC chat, VNC desktop, GitHub list…). Split it into panes, too.

9 Footer New connection, palette, import, export, recordings and settings.

10 Status State, protocol and live latency for SSH. Shows **REC** while recording and **BROAD-**

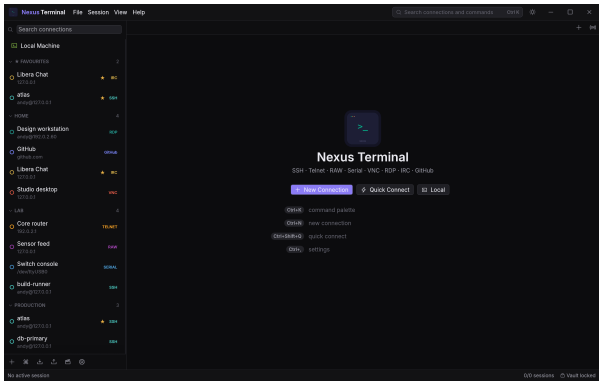
CAST when broadcasting.

⓫ **Sessions & vault** How many sessions are connected and whether the vault is locked. Click the vault to open security settings.

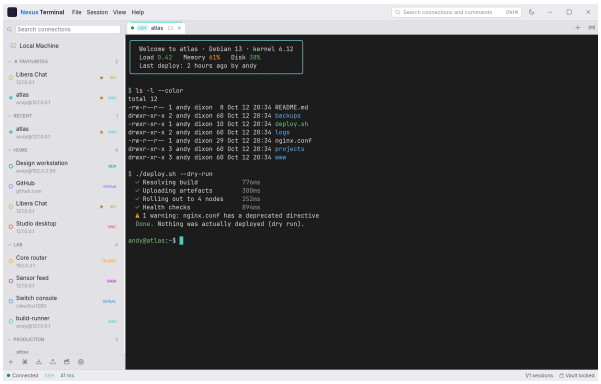
> **Pro tip.** Drag the right-hand edge of the sidebar to make it wider or narrower. Toasts (the little pop-up messages in the top-right) tidy themselves away after a few seconds; click × if you're impatient.

2.1 Three looks

Nexus Terminal comes in **Pro** (the default, a soft charcoal), **Midnight** (darker, bluer, for 2 a.m. deploys) and **Light** (for sunny days and projectors). Switch with the button in the title bar, the **View** menu, or type "theme" into the palette.



Midnight



Light, with a session open

The *terminal* has its own colour schemes too: eleven of them, including **Dracula**, **Nord**, **Tokyo Night**, **Catppuccin Mocha** and **Rosé Pine**. Those live in **Settings > Terminal** and the **View** menu.

✓ **Done when** you can point at any part of the window and say what it is for.

CHAPTER 3

Profiles: your address book

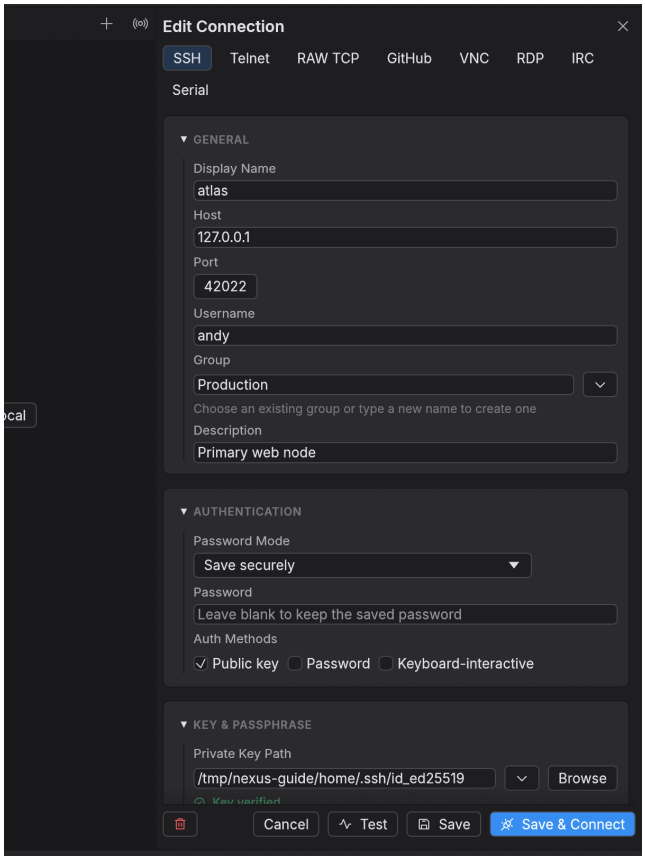
TL;DR

A *profile* is a saved connection. Give it a name and a group, star the ones you use daily, and right-click for everything else. Ctrl-click to select lots at once.

6 minutes · You'll keep dozens of connections tidy without thinking about it

3.1 Create, edit, copy

- **New:** **+** in the sidebar footer or tab strip, `Ctrl / ⌘` + `N`, or right-click a group header and choose **New connection in …** (it lands in that group automatically).
- **Edit:** double-click a profile, or right-click > **Edit**. The editor slides in on the right.
- **Copy:** right-click > **Duplicate**. Great for "same as web-01 but web-02".
- **Delete:** right-click > **Delete**. This also deletes the saved passwords for that profile.



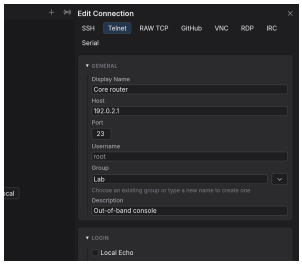
Editing an existing SSH profile. The editor scrolls; sections are grouped so you can ignore most of them.

What goes in a profile

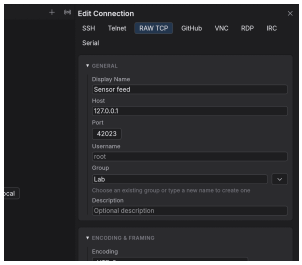
Every profile has the same **General** section, then sections specific to its protocol (Part 2 covers each one). The common bits are:

Display Name	What you'll see in the sidebar. Make it obvious: "db-primary", not "server 3".
Host / Port	Where to connect. Serial profiles have a **Device** instead.
Group	Pick an existing group or type a new name to create one.
Description	One line shown on hover.
Terminal	Font, size (8–36), scrollback (0–100 000 lines) and cursor style, just for this profile.
Tags & Notes	Comma-separated tags (searchable!) and free-form notes for future-you.

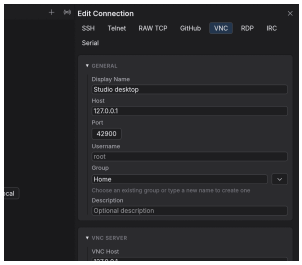
Each protocol adds its own sections underneath. Here are four of them side by side; Part 2 goes through each one.



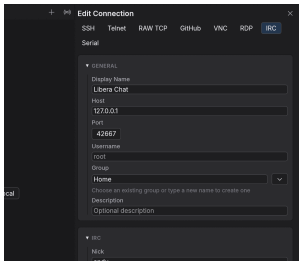
Telnet



RAW TCP



VNC



IRC

> **Tags are secretly brilliant** Tag things by where they live and what they do (`eu-west`, `db`, `customer-x`). The sidebar search matches tags, so typing "db" instantly narrows hundreds of profiles to the database servers.

3.2 Favourites, groups and Recent

- Right-click > **Add to Favourites** pins a profile into the ★ **Favourites** section at the top.
- **Recent** always shows the last five things you connected to. You don't have to do anything.
- Groups are just the **Group** field. No group? It goes in **Ungrouped**.

3.3 Do lots at once

Hold `Ctrl` (`⌘` on a Mac) and click several profiles. A bar appears at the bottom of the sidebar:

Connect all	opens them all in tabs	**Move group**	moves them (blank = Ungrouped)
Disconnect	closes their sessions	**Delete**	deletes them (asks first)
Tags	add or replace tags	**Clear**	forget the selection

▷ **Quick win**

Select every web server, choose **Connect all**, then turn on broadcast (`Ctrl` / `⌘` + `Shift` + `B`). You're now typing into all of them at once. Chapter 14 has the details, and the safety advice.

3.4 "It's already open…"

Click a profile that already has a live session and Nexus Terminal asks whether to **Use Existing** (jump to that tab), open a **New Connection** alongside it, or **Cancel**. No more accidental duplicate sessions.

✓ **Done when** every connection you use has a profile in a group, and the ones you use daily are favourites.

PART 2

Connect to all the things

Nine ways in. Pick the one you need; skip the rest.

- SSH: the big one
- Serial: cables and consoles
- Telnet and RAW TCP
- VNC remote desktops
- RDP: Windows desktops
- IRC
- GitHub

CHAPTER 4

SSH: the big one

TL;DR

Host, username, key, done. Everything else (jump hosts, proxies, forwards, 2FA, login scripts) is optional and lives in its own fold-away section of the editor. Open only the ones you need.

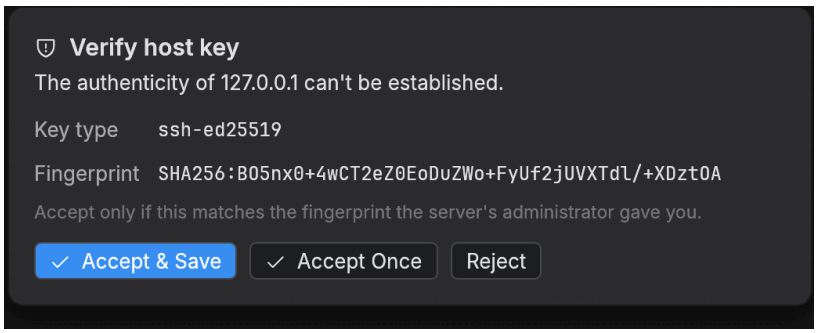
10 minutes (or 2 for the basics) · You'll log in with keys, hop through bastions and never type a 2FA code again

SSH AT A GLANCE

Default port	22
Logs in with	Password, public key, keyboard-interactive, SSH agent
Extras	SFTP file browser, port forwards, live latency, agent forwarding, TOTP 2FA
Good for	Servers, VMs, Raspberry Pis, anything with `sshd`

4.1 The two-minute version

1. `Ctrl` / `⌘` + `N`, leave the protocol on `SSH`.
2. **Host**, **Username**. Leave the port on 22 unless you know otherwise.
3. Under **Key & Passphrase**, pick your key from the suggestions (Nexus Terminal lists the keys in your `~/.ssh` folder) and check for a little green **Key verified**.
4. **Create & Connect**. The first time, you'll be asked to trust the server's host key. Say yes if you were expecting it.



First time connecting? You'll be shown the server's fingerprint. **Accept & Save** if it's the server you meant.

⚠ **Watch out.** Type the full path to a key if you're not using the suggestions. Nexus Terminal doesn't expand `~` in that box, so write `/home/you/.ssh/id_ed25519` (or `C:\Users\you\.ssh\id_ed25519`).

4.2 How it logs in

Under **Authentication**, tick which **Auth Methods** to try: **Public key**, **Password** and **Keyboard-interactive**. Under **SSH Agent**, tick **Use SSH Agent for Authentication** to use keys already loaded into `ssh-agent` (or Pageant, or the Windows OpenSSH agent).

Got an SSH *user certificate*? Put it next to the key with the usual name (`id_ed25519-cert.pub` beside `id_ed25519`) and Nexus Terminal presents it automatically. Servers that hand out *host* certificates are covered in Host keys (p. 52).

Every password and passphrase has a *mode*: save it securely, remember it until you quit, or ask every time. That's all explained in Passwords & the vault (p. 49).

💡 **Agent forwarding** Tick **Forward Agent to Remote** (it's `ssh -A`) and you can `git pull` or hop onwards from the server using the keys on *your* machine. Only do it on servers you trust: anyone with root there can use your agent while you're connected.

4.3 Jump hosts: getting through the bastion

Lots of servers hide behind a bastion. Add one or more hops under **Jump Hosts**, each with its own user, host, port and optional key. Nexus Terminal tunnels through them in order, so the chain *you > bastion > internal-box > db-primary* is just three rows.

4.4 Proxies

Stuck behind a corporate proxy? Under **Proxy**, choose **SOCKS5** or **HTTP CONNECT** and fill in the host, port and (optional) username and password. The proxy is used to reach the SSH server, or the first jump host if you have some.

4.5 2FA without the phone faff

If your server asks for a one-time code, paste the Base32 TOTP secret (the text version of the QR code) into **TOTP Secret** and hit **Save**. From then on, Nexus Terminal answers the "Verification code:" prompt itself. **Test** shows today's code so you can check it matches your phone.

⚠ **Watch out.** Save the profile first, then save the TOTP secret. And make sure your computer's clock is right: TOTP codes are time-based, so a clock that's a minute out will produce wrong codes.

4.6 Do things automatically on login

Startup A **Startup Command** to run once you're in (say, `tmux attach`), plus **Environment Variables** as `KEY=VALUE`, one per line.

Login Script For old-school boxes that ask questions. Add steps: **Wait for** some text, then **Send** a reply. Repeat as needed.

4.7 Keeping it alive

Under **Advanced**: the **Terminal Type**, the **Keepalive Interval** and **Max Missed Keepalives**, compression, and **Auto-Reconnect** with a maximum number of attempts (1–20). Reconnection backs off politely: 1 s, 2 s, 4 s… up to 30 s.

⏩ **What's that number in the status bar?** · optional; skip it if you are in a hurry

For SSH sessions the status bar shows live latency, the round trip to the server. Green is snappy. If it's in the hundreds and your typing feels sticky, it's the network, not you.

✓ **Done when** you log in with a key, and any bastion or 2FA code is handled for you.

CHAPTER 5

Serial: cables and consoles

TL;DR

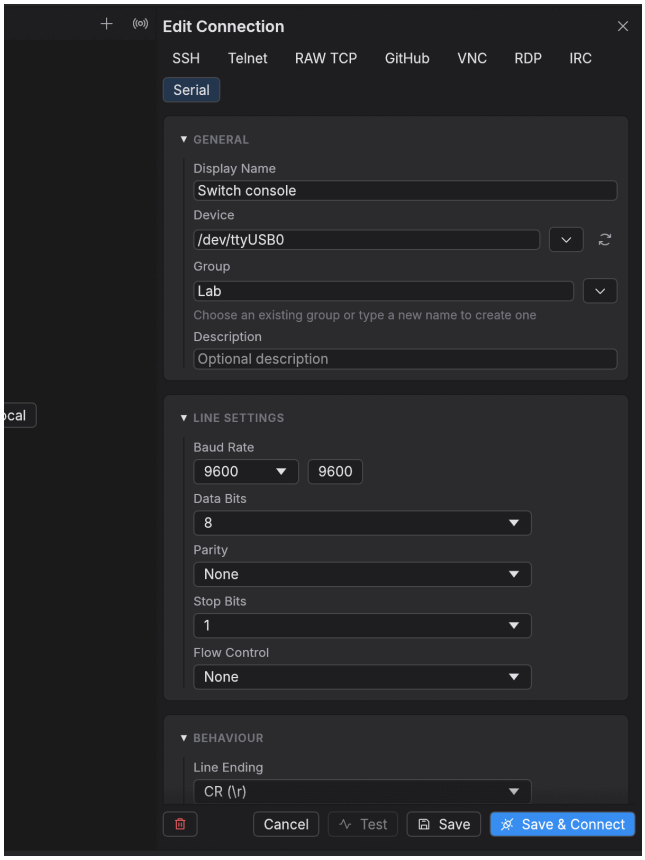
Plug in the adapter, pick the device, match the baud rate. 115200 8N1 is the default and usually right. Right-click the tab to send a break or wiggle DTR/RTS.

4 minutes · You'll be talking to a switch, router or microcontroller over USB

Serial

AT A GLANCE

Device	`/dev/ttyUSB0`, `/dev/cu.usbserial-…` or `COM3`
Defaults	115200 baud, 8 data bits, no parity, 1 stop bit, no flow control, CR line endings, DTR and RTS on
Extras	Send Break, toggle DTR/RTS, auto-reconnect when the cable comes back
Good for	Network gear consoles, Arduinos, ESP32s, Raspberry Pi UARTs, 3D printers




A serial profile for a switch console at 9600 baud. Click refresh after plugging in an adapter.

5.1 Setting it up

Device Pick it from the list (hit refresh after plugging in) or type the path.

Line Settings **Baud Rate** (a preset from 300 to 1 000 000, or **Custom**), **Data Bits** (5–8), **Parity** (none, odd, even), **Stop Bits** (1 or 2) and **Flow Control** (none, XON/XOFF, RTS/CTS).

Behaviour **Line Ending**, **Encoding**, **Local Echo** (for devices that don't echo what you type), **Assert DTR on connect**, **Assert RTS on connect** and **Auto-reconnect**.

5.2 While you're connected

Right-click the tab for **Send Break** (handy for getting into a router's ROMMON) and to toggle **DTR** and **RTS** (handy for resetting boards that wire those to reset). Pull the cable out and the session disconnects; with auto-reconnect on, it picks up again when you plug it back in.

Serial tabs are proper terminals, so recording, triggers, broadcast, snippets, find and split panes all work.

⚠ **Gibberish on screen?** Garbled characters nearly always mean the baud rate, parity or stop bits don't match the device. Check the device's manual. 9600 and 115200 are the usual suspects.

Linux and macOS quirks On Linux, add yourself to the `dialout` group (`uucp` on some distributions) and log out and in again. On a Mac, use the `/dev/cu.*` device, not `/dev/tty.*`.

Done when the device's console answers when you press `Enter`.

CHAPTER 6

Telnet and RAW TCP

TL;DR

Telnet is a terminal for old devices. RAW is a packet-level conversation with any TCP port, with a hex view.

4 minutes · You'll poke at old kit and raw sockets without reaching for netcat

6.1 Telnet

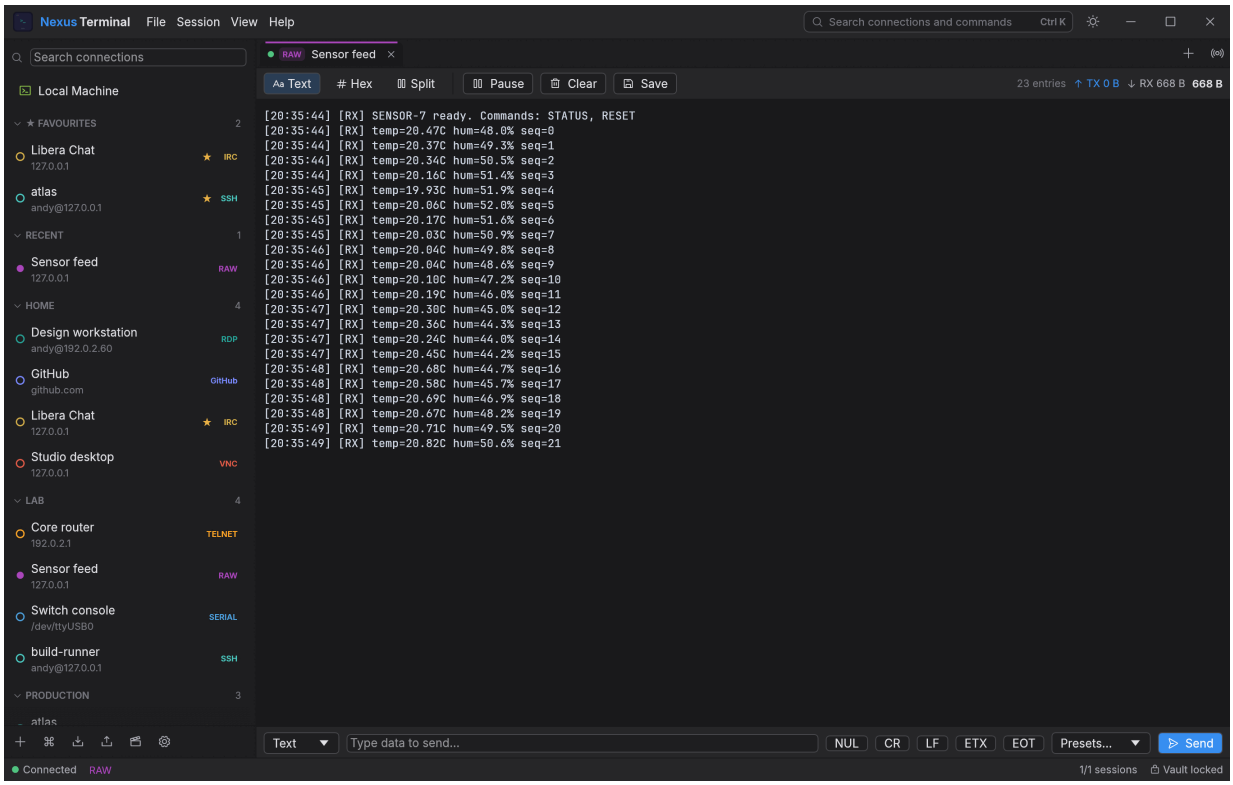
Telnet AT A GLANCE

Default port	23
Options	Local echo, line mode, terminal type, encoding (UTF-8, ASCII, ISO 8859-1), line endings, login script
Extras	Send Break from the tab menu, **Debug Negotiation** for when options go wrong

Telnet is plain text Everything, including your password, crosses the network unencrypted. Fine on a lab bench, not fine over the internet or café Wi-Fi.

6.2 RAW TCP

RAW connects to any TCP port and shows you exactly what goes back and forth. Instead of a terminal you get a traffic log with **Text**, **Hex** and **Split** views.



A RAW session to a chatty sensor, with timestamps and direction markers turned on.

- **Send bar:** pick **Text**, **Hex** or **Escaped** (`\r\n` and friends), type, press `Enter`. One-click buttons send **NUL**, **CR**, **LF**, **ETX** and **EOT**.
- **Presets:** save payloads you send often in the profile under **Payload Presets**, then pick them from **Presets...**.
- **Capture:** **Pause**, **Clear**, and **Save** (writes the whole conversation as JSON).
- **Profile options:** encoding (UTF-8, ASCII, Hex), line ending, timestamps, direction, timeouts and auto-reconnect.

> ▷ **Quick win**
>
> Point a RAW profile at a web server on port 80 and send `HEAD / HTTP/1.0\r\n\r\n` in **Escaped** mode. Instant HTTP headers, no browser required.

✓ **Done when** the device or port you were poking at answers you.

CHAPTER 7

VNC remote desktops

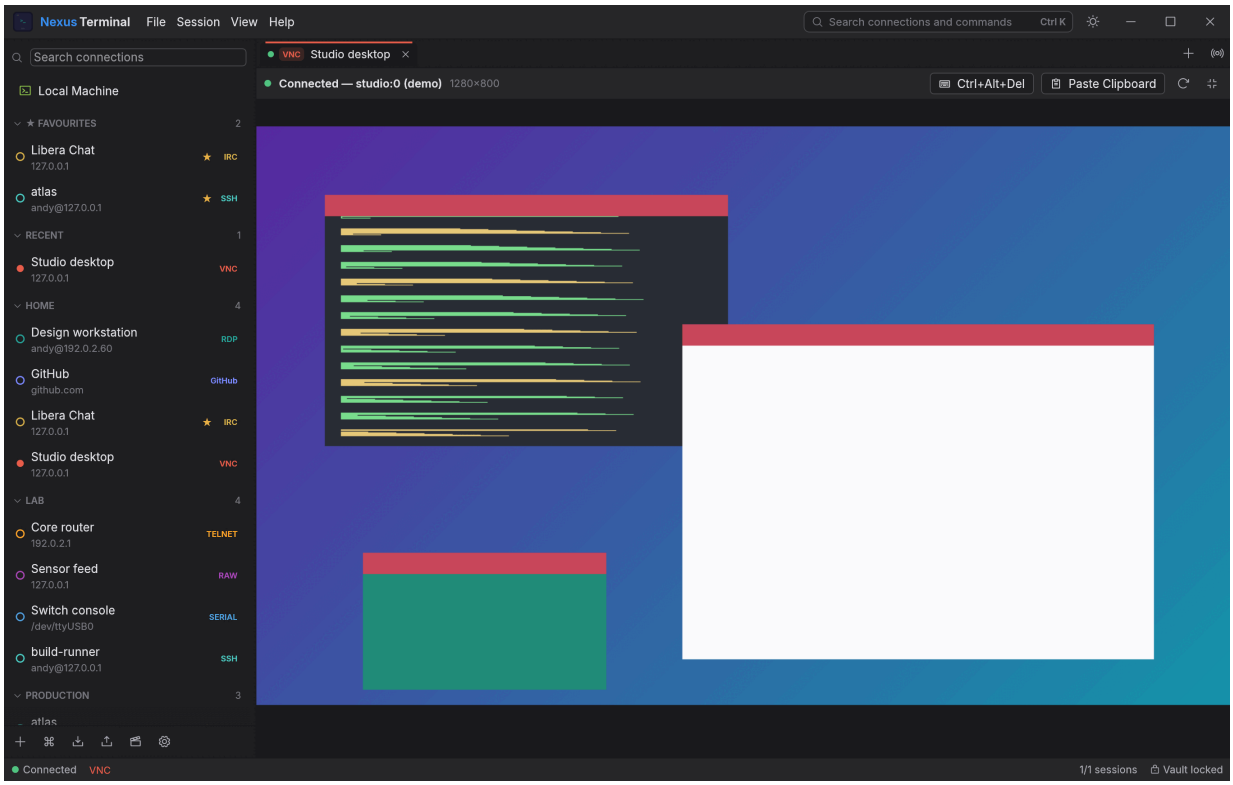
TL;DR

VNC shows you another computer's screen. Tunnel it through SSH (tick a box), because plain VNC isn't encrypted.

4 minutes · You'll see and drive a remote desktop, safely

VNC AT A GLANCE

Default port	5900 (plus the display number: display `:1` is 5901)
Display	**View Only**, **Scale to Fit**, **Shared Session**, **Quality (0-9)**
Security	Optional SSH tunnel with password, key or agent login



A remote desktop, scaled to fit the tab.

7.1 The toolbar

Connection state and desktop size on the left. On the right: switch between scaled and 1:1, refresh, **Paste Clipboard** (sends your clipboard to the remote machine) and **Ctrl+Alt+Del**. Click the desktop to give it your keyboard. Anything the remote machine copies lands on your clipboard automatically.

7.2 Tunnel it

Tick **Enable SSH Tunnel** and fill in the SSH host, port, username and login method. The VNC traffic then travels inside an encrypted SSH connection. If you connect to a VNC profile *without* a tunnel, Nexus Terminal warns you and offers **Connect Anyway**, **Edit Profile** or **Cancel**.

> ⚠ **Connecting to a Mac?** Turn on **VNC viewers may control screen with password** in macOS Screen Sharing settings. Nexus Terminal uses standard VNC password authentication, not Apple's own login method.

✓ **Done when** you can see the remote desktop and move its pointer.

CHAPTER 8

RDP: Windows desktops

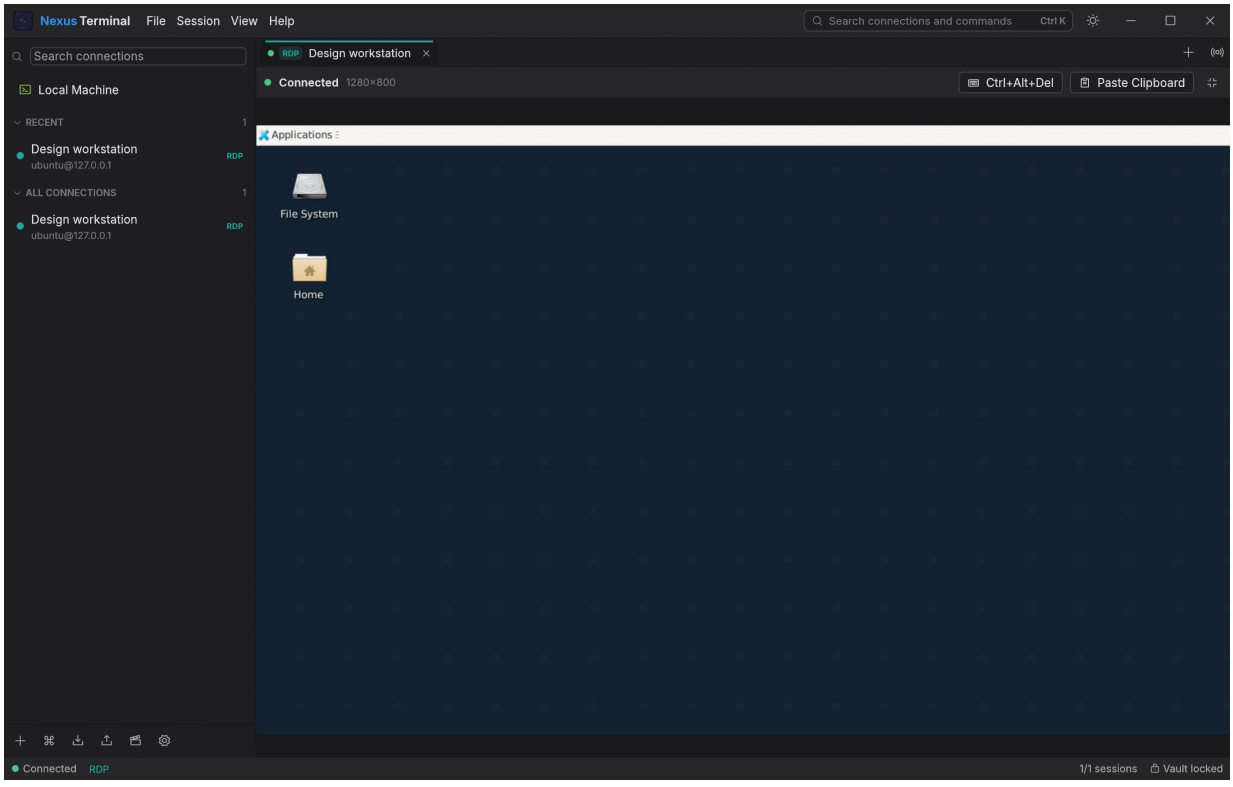
TL;DR

Host, username, password. Nexus Terminal logs on with NLA, pins the server's certificate like an SSH host key, and can go through a proxy, an SSH tunnel or an RD Gateway.

4 minutes · You'll be on a Windows (or xrdp) desktop without leaving Nexus Terminal

RDP AT A GLANCE

Default port	3389
Logs in with	NLA (NTLM, or Kerberos with a KDC); **Allow servers without NLA** for xrdp
Display	**Scale to Fit**, fixed size, **Resize With Tab**
Extras	Clipboard both ways, remote audio, printer, shared folder
Getting there	SOCKS5 or HTTP proxy, SSH tunnel, RD Gateway (experimental)



An xrdp desktop in a tab. Ctrl+Alt+Del and Paste Clipboard sit on the toolbar.

8.1 Setting it up

1. `Ctrl / ⌘` + `N`, protocol `RDP`.
2. **Host** and **Username**; add a **Domain** if your account has one.
3. Pick how the password is kept (save it, remember it until you quit, or ask every time).
4. **Create & Connect**. The first time, you're shown the server's certificate fingerprint. Accept it if it's the machine you meant.

Windows checks your password *before* it builds a desktop (Network Level Authentication), so a typo fails fast. Linux machines running xrdp don't speak NLA: tick **Allow servers without NLA** for those. In an Active Directory shop, set a **Kerberos KDC** (`host` or `host:port`) and use your realm as the domain.

> ⚠ **A changed certificate** RDP servers mostly use self-signed certificates, so Nexus Terminal remembers the fingerprint in **Known Hosts**. If it changes you get **Reject**, **Accept Once** or **Accept & Update**, and nothing (not even your password) is sent until you choose.

8.2 While you're connected

It works like the VNC viewer: click the desktop to give it your keyboard, **Ctrl+Alt+Del** sends exactly that, and text you copy on either side turns up on the other. With **Resize With Tab** the remote desktop follows the tab's size (Windows does this; other servers are scaled).

Under **Devices**: **Play Remote Audio**; **Redirect Printer**, which adds a *Nexus Terminal* printer whose jobs land as PostScript files in your Downloads folder; and (Linux and macOS) a **Shared Folder** that shows up remotely as a drive called Nexus. The server can read and change that folder, and nothing outside it.

8.3 Getting through

Pick *one* of: a **Proxy** (SOCKS5 or HTTP CONNECT), an **SSH Tunnel** (same as VNC), or an **RD Gateway** (experimental: the gateway needs a proper public certificate and Basic authentication over HTTPS).

✓ **Done when** the Windows desktop is open in a tab.

CHAPTER 9

IRC

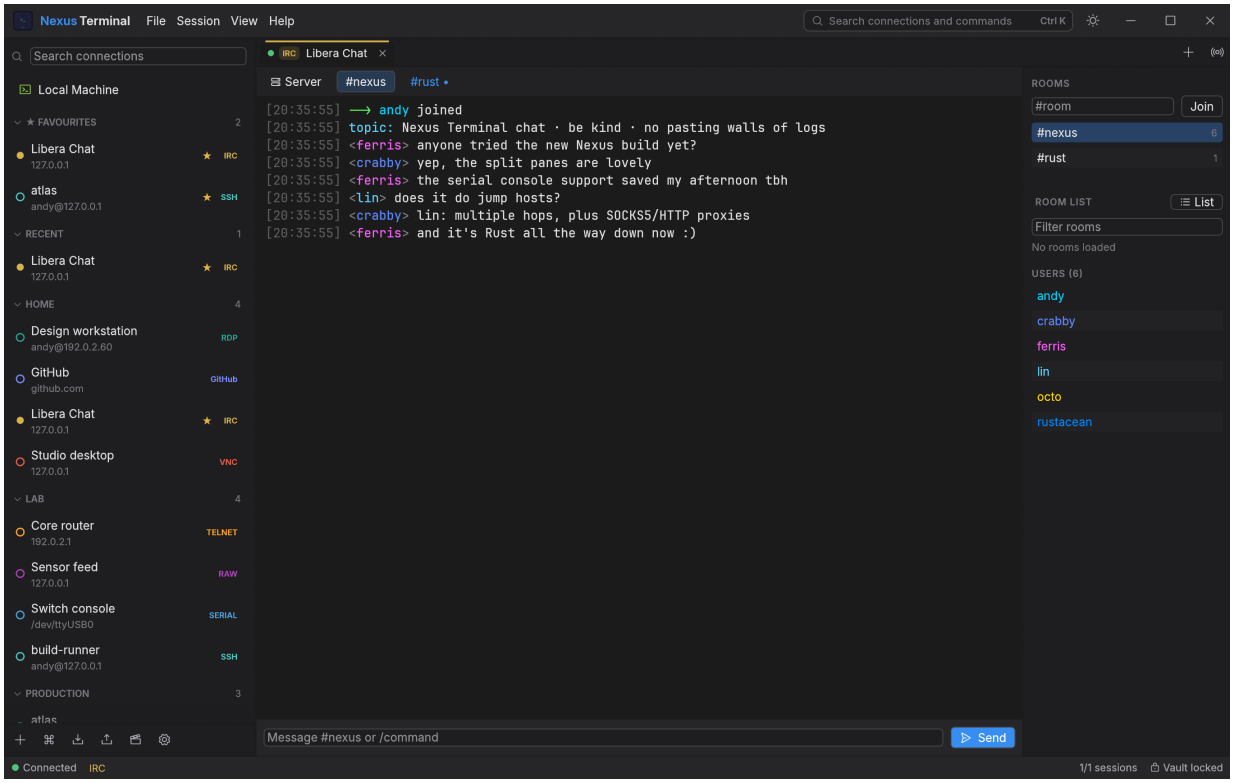
TL;DR

Nick, server, rooms to join. Tabs for each room, `/commands` as you'd expect, `Tab` completes nicks.

3 minutes · You'll be chatting in your project's channels from the same window as your servers

IRC AT A GLANCE

Default port	6667, or 6697 with **Use TLS** (use TLS!)
Logs in with	SASL PLAIN (most networks want this now), or NickServ identify
Extras	Auto-join rooms, room browser, real message times (IRCv3 server-time), SSH tunnel



Chatting in #nexus. Rooms down the right, users in the room below them.

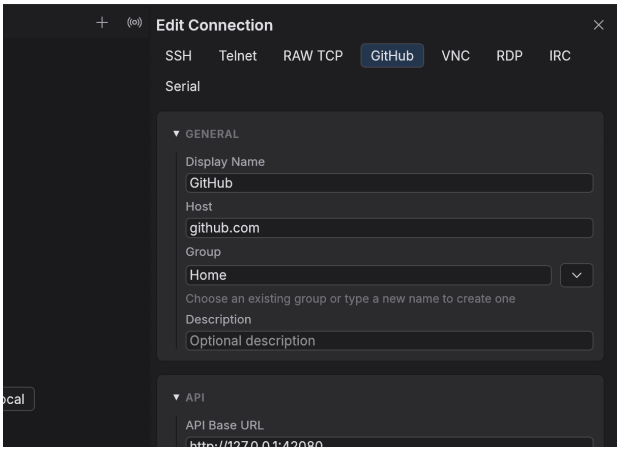
- A **Server** tab plus one per room or private conversation. A dot means unread messages.
- Commands: `/join`, `/part`, `/msg`, `/me`, `/notice`, `/nick`, `/topic`, `/list`, `/names`, `/whois`, `/who`, `/raw`, `/quit` and `/help`.

- `Tab` completes a nick; click a nick to mention them.
- The side panel lists your rooms, a room browser with a filter (click to join) and who’s in the current room.

Done when you are in your channels and can see the conversation.

CHAPTER 10

GitHub

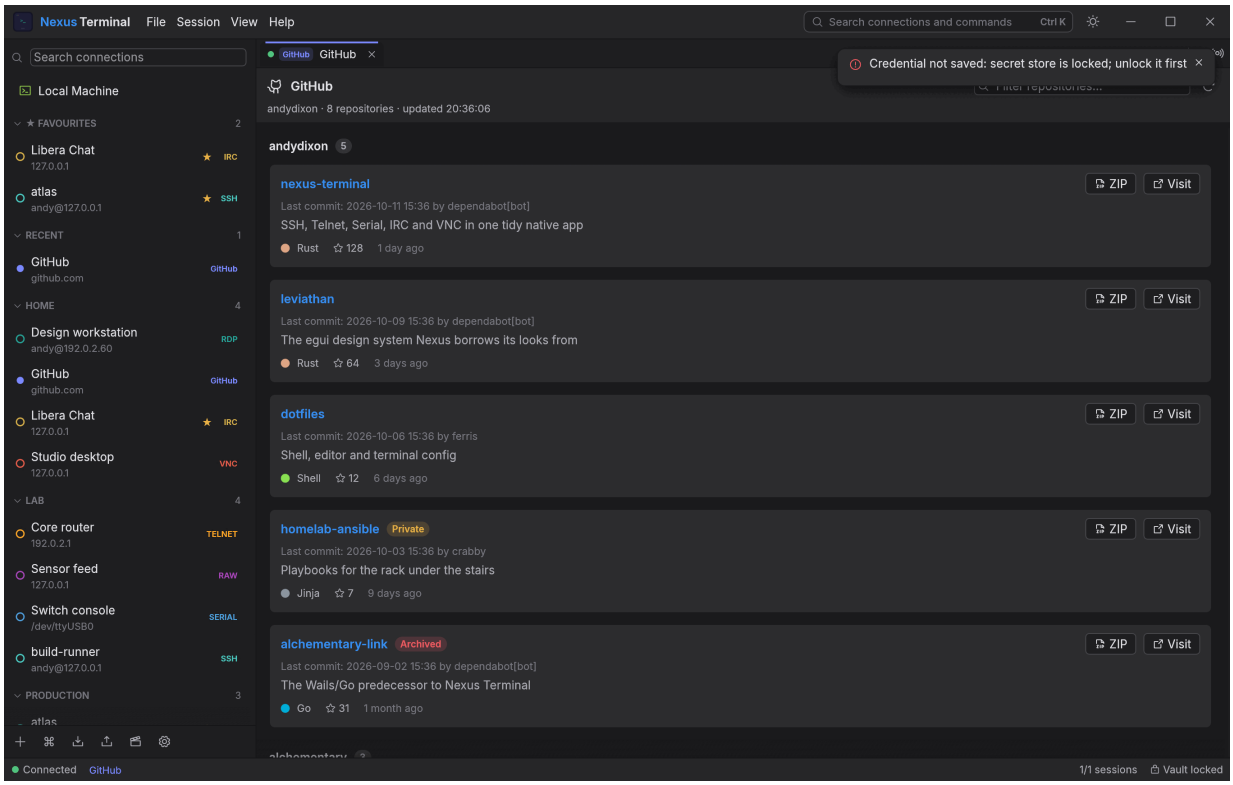


A GitHub profile: just the API address and a token.

TL;DR

Paste a personal access token with the `repo` scope. Get a tidy, filterable list of every repository you can see.

2 minutes · You'll see all your repositories, freshest first



Repositories grouped by owner, with the latest commit and who made it.

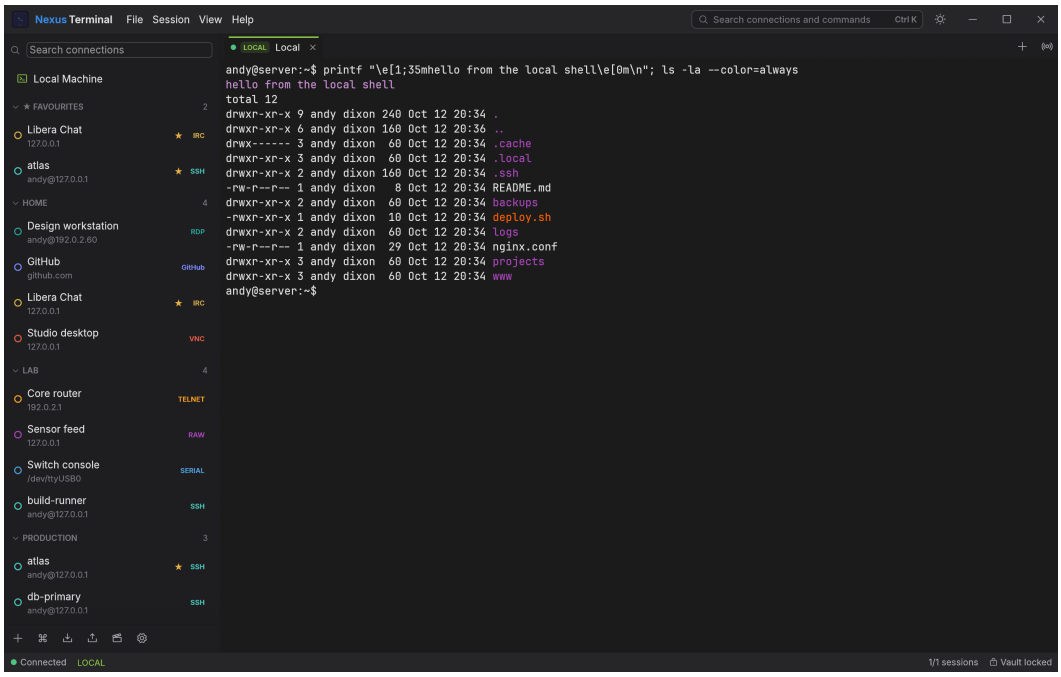
- Grouped by owner or organisation, sorted by most recent activity.
- Badges for `Private`, `Fork` and `Archived`, plus language and stars.
- Filter box, **Refresh now**, **Visit** (opens in your browser) and **ZIP** (downloads an archive).
- Works with GitHub Enterprise: set **API Base URL** to your server's API address. Refreshes every 60–3600 seconds.

Done when your repositories are listed in a tab, freshest first.

CHAPTER 11

Your own computer

Click **Local Machine** for a shell on this computer. It uses your system's default shell unless you choose another in **Settings > General > Local shell** (`pwsh`, `zsh`, `fish`, whatever you fancy).



A local shell, with all the same terminal powers as SSH.

Done when a **Local Machine** tab is open with your usual shell.

PART 3

Work faster

Fewer clicks, less typing, more getting on with it.

- Tabs and split panes
- Terminal superpowers
- Never type it twice
- Files: SFTP and drag-and-drop
- Port forwards
- Recordings

CHAPTER 12

Tabs and split panes

TL;DR

Right-click a tab for everything. `Ctrl / ⌘` + `Shift` + `D` splits right, `Ctrl / ⌘` + `Shift` + `E` splits down. `Ctrl / ⌘` + `1` to `Ctrl / ⌘` + `9` jump straight to a tab.

4 minutes · You'll juggle a dozen sessions without losing any of them

12.1 Tabs

Move Drag tabs about, or right-click > **Move Left** / **Move Right**.

Pin Right-click > **Pin**. Pinned tabs stay at the front.

Rename Right-click > **Rename**. "prod DB, DON'T TOUCH" is a perfectly valid tab name.

Duplicate Opens another session to the same profile.

Reconnect Does what it says.

Peek Hover a tab for its name, host, state and when it connected.

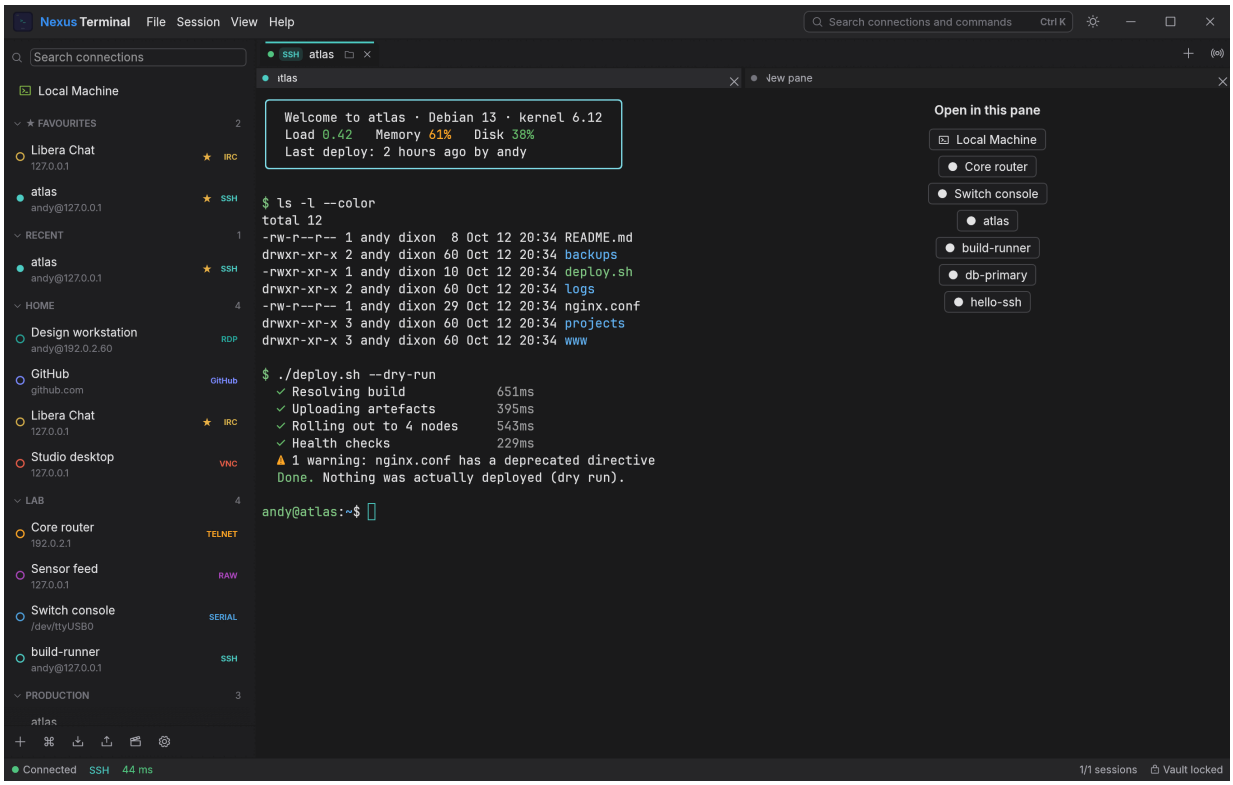
Close The ×, `Ctrl / ⌘` + `W`, or right-click > **Close**. Connected tabs ask first (you can turn that off in Settings).

Switch `Ctrl / ⌘` + `Tab` and `Ctrl / ⌘` + `Shift` + `Tab`, or `Ctrl / ⌘` + `1` to `Ctrl / ⌘` + `9`.

Lots of tabs? Scroll the tab strip with your mouse wheel or trackpad. Quit with a dozen open and they're all back next time: **Restore tabs at launch** (on by default) reconnects saved profiles and reopens local shells, splits and all, in the folder they were in.

12.2 Split panes

Want two sessions side by side? Split the focused pane right with `Ctrl / ⌘` + `Shift` + `D` or down with `Ctrl / ⌘` + `Shift` + `E` (also in the **Session** menu, the right-click menu and the palette). The new pane asks what to open: **Local Machine** or any of your terminal profiles.



A fresh split. Pick what goes in the new pane.

Drag a divider to resize. Click **×** on a pane's header to close it and its neighbour takes the space. Closing the tab closes every pane in it.

> ▷ **Quick win**
>
> Split right, open `htop` on one side and tail a log on the other. Now you look like a hacker in a film.

✓ **Done when** your sessions are in tabs and panes you can move between from the keyboard.

CHAPTER 13

Terminal superpowers

TL;DR

Select to copy, `Ctrl / ⌘` + `F` to search, Ctrl/⌘-click links. Right-click for the rest.

4 minutes · You'll copy, search and scroll like you've used it for years

13.1 Keyboard and mouse

Copy	Select text, then `Ctrl / ⌘` + `C`. Nothing selected? `Ctrl` + `C` interrupts the program as usual.
Paste	`Ctrl / ⌘` + `V`. Pastes are bracketed when the program asks for it, so editors won't auto-indent your paste into a staircase. Pasting several lines into a shell asks first (turn that off in Settings).
Select	Drag for a range, double-click a word, triple-click a line, `Shift`-click to extend.
Links	Hold `Ctrl` (`⌘`) and click a URL to open it in your browser. Real hyperlinks that programs print (`ls --hyperlink`, compilers) get a dotted underline; hover to see where they really go.
Scroll	Mouse wheel, or `Shift` + `PageUp`, `Shift` + `PageDown`, `Shift` + `Home`, `Shift` + `End`.

Why does the scroll wheel move the cursor in less? · optional; skip it if you are in a hurry

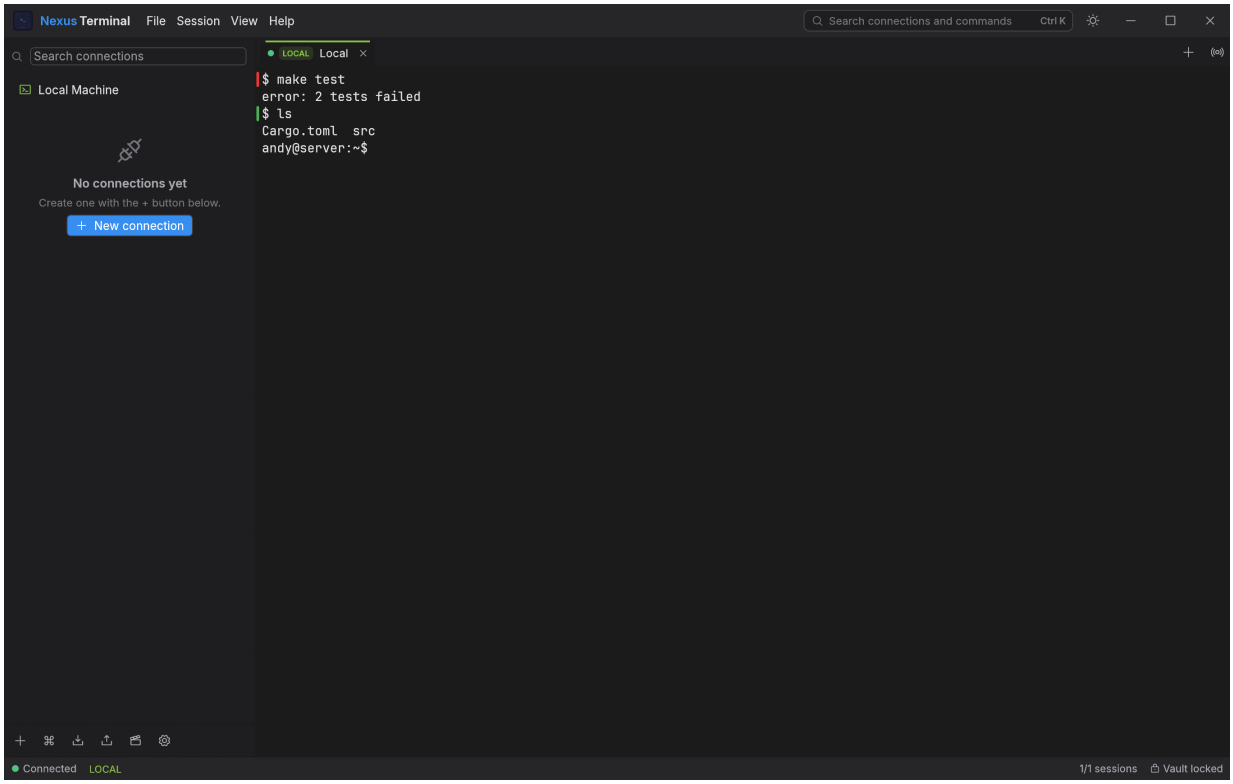
Full-screen programs (`less`, `vim`, `tmux`, `byobu`) use the terminal's "alternate screen", where the wheel sends arrow keys instead of scrolling. Programs with mouse support (`htop`, `vim` with `mouse=a`, `tmux` mouse mode) get your clicks directly. That's normal terminal behaviour, not a bug.

13.2 The right-click menu

Copy, **Paste**, **Select All**, **Find**, **Previous Prompt** / **Next Prompt**, **Copy Last Command Output**, **Send Snippet**, **Port Forwards** (SSH), **Split Right**, **Split Down**, **Export Output** (saves the whole scrollback as a text file), **Clear** and **Reset**.

13.3 Shell integration

If your shell marks its prompts (fish 4, Starship and Powerlevel10k do already), Nexus Terminal knows where every command starts and stops. `Ctrl / ⌘` + `Shift` + `Up` and `Ctrl / ⌘` + `Shift` + `Down` jump from prompt to prompt, **Copy Last Command Output** grabs just what the last command printed, and a thin bar beside each prompt shows how it went: green for success, red for failure. Hover the bar for the exit code and how long it took.



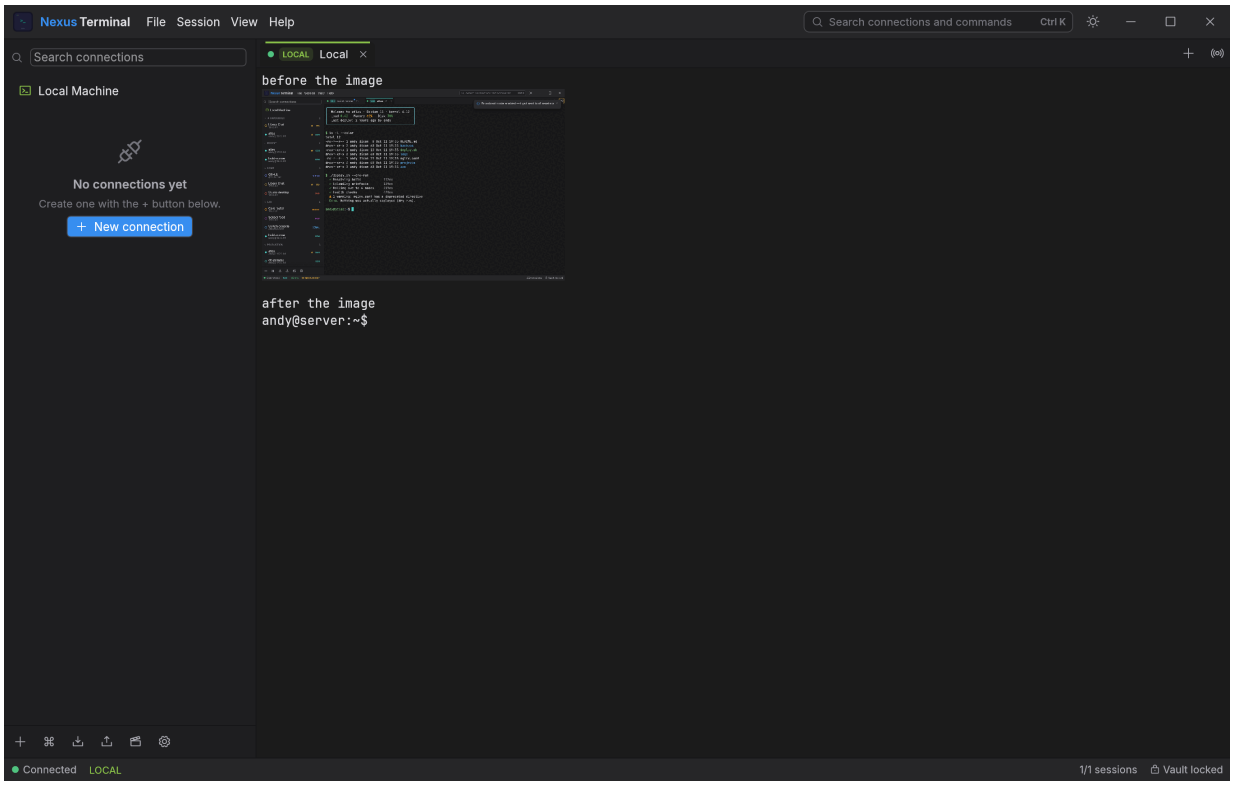
`make test` failed (red bar), `ls` worked (green bar).

For bash, add this to `~/.bashrc` on the machine the shell runs on (the user manual has the zsh version):

```
PS0='\[\e]133;C\a\]'
PROMPT_COMMAND='printf "\033]133;D;%s\007" "$?"; '"${PROMPT_COMMAND}"
PS1='\[\e]133;A\a\]'"$PS1"'\[\e]133;B\a\]'
```

13.4 Pictures in the terminal

Programs that print images the iTerm2 way (`imgcat`, `wezterm imgcat`, `chafa -f iterm`) show PNG, JPEG and GIF images right in the scrollback. They scroll with the text and never spill past the edge.



An image printed by a script, with the text carrying on underneath.

13.5 Find

`Ctrl` / `⌘` + `F` opens the find bar. Toggle **Aa** for case-sensitive and **.*** for regular expressions. `Enter` for the next match, `Shift` + `Enter` for the previous one, `Esc` to close. It shows “3 of 17” so you know where you are.

13.6 When the connection drops

The pane dims and shows a **Disconnected** card with **Reconnect** and **Close**. If auto-reconnect is on, it says **Reconnecting…** with a **Cancel** button, and a toast tells you which attempt it’s on.

If the problem was a password or host key, you get a friendly dialog explaining what happened, with **Retry**, **Edit Profile**, **Known Hosts** (for host-key trouble) and **Close**.

Done when you have copied, searched and scrolled back through a session’s output.

CHAPTER 14

Never type it twice

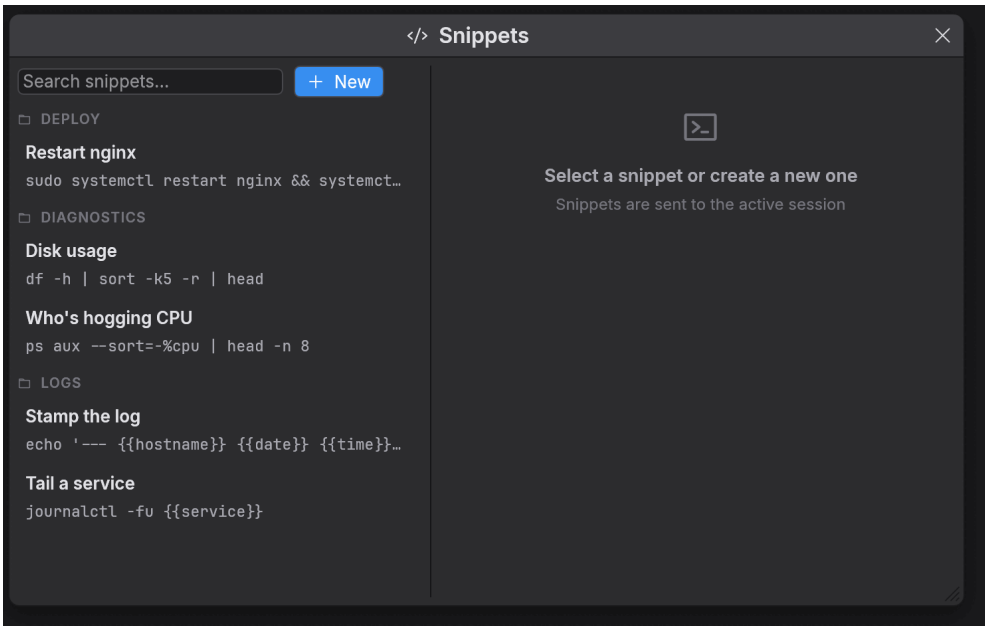
TL;DR

Snippets store commands. **Broadcast** types into many sessions at once. **Triggers** watch output and tap you on the shoulder.

6 minutes · You'll stop retyping the same commands, and let Nexus Terminal watch the logs for you

14.1 Snippets: `Ctrl` / ⌘ + `Shift` + `S`

Save commands you use a lot, with a name, description and folder. Search them, click **Send to Session**, and the command is typed into the active session followed by `Enter`.



The snippet library, organised into folders.

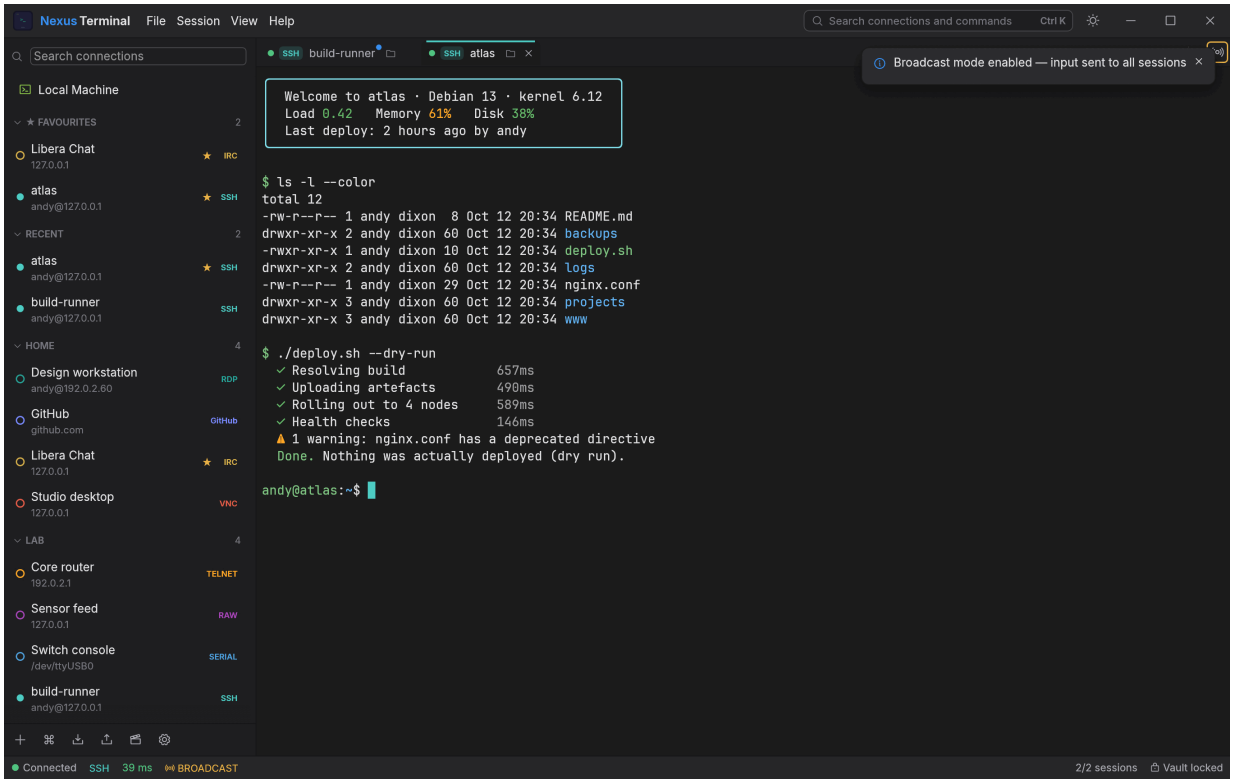
Snippets can have blanks to fill in:

- `{{hostname}}`, `{{date}}`, `{{time}}` and `{{timestamp}}` are filled in for you.
- Anything else, like `{{service}}`, is asked for when you send it.

So `journalctl -fu {{service}}` asks "service?" and tails whichever one you type.

14.2 Broadcast: `Ctrl / ⌘` + `Shift` + `B`

Turn broadcast on and everything you type goes to *every* connected terminal. The status bar shows **BROADCAST** so you can't forget it's on.



Broadcast is on: the status bar says so in no uncertain terms.

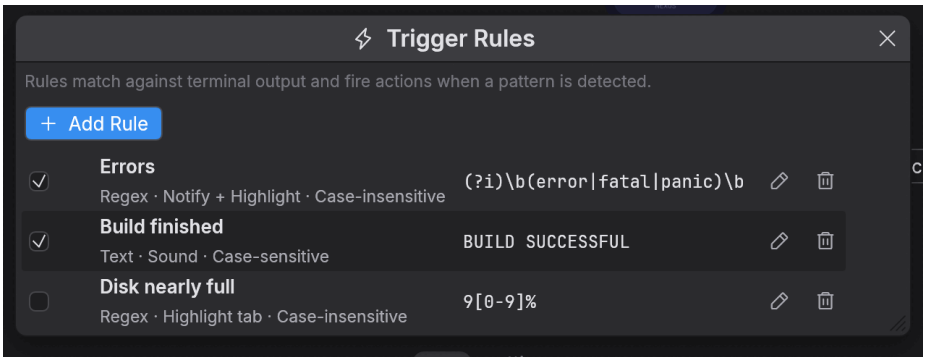
To broadcast to only some sessions, right-click the (((•))) button (or **Session > Broadcast targets...**), tick the ones you want and click **Apply & Enable**.

> **With great power...** Broadcast doesn't care which tab you *meant*. Double-check the target list before you type anything destructive. `sudo reboot` into twenty servers is very efficient and very memorable.

14.3 Triggers: let it watch for you

Session > Trigger Rules (or the palette). Each rule has a pattern (plain text or a regular expression), a case-sensitivity switch, an on/off switch, and an action:

Notify	shows a toast
Highlight tab	briefly outlines the pane
Notify + Highlight	both
Sound	asks your OS for attention (a bouncing dock icon or flashing taskbar button) and shows a to



Three rules: shout about errors, ping when a build finishes, and one switched off for now.

Rules check each complete line of output, with colours stripped out first.

> **Quick win**
>
> Add a rule for `BUILD SUCCESSFUL` with the **Sound** action, start a long build, and go and make a cup of tea. Nexus Terminal will tell you when it's done. (Or `(?i)error|fatal` for the opposite.)

Done when the commands you type most are snippets, and anything that goes to several servers goes by broadcast.

CHAPTER 15

Files: SFTP and drag-and-drop

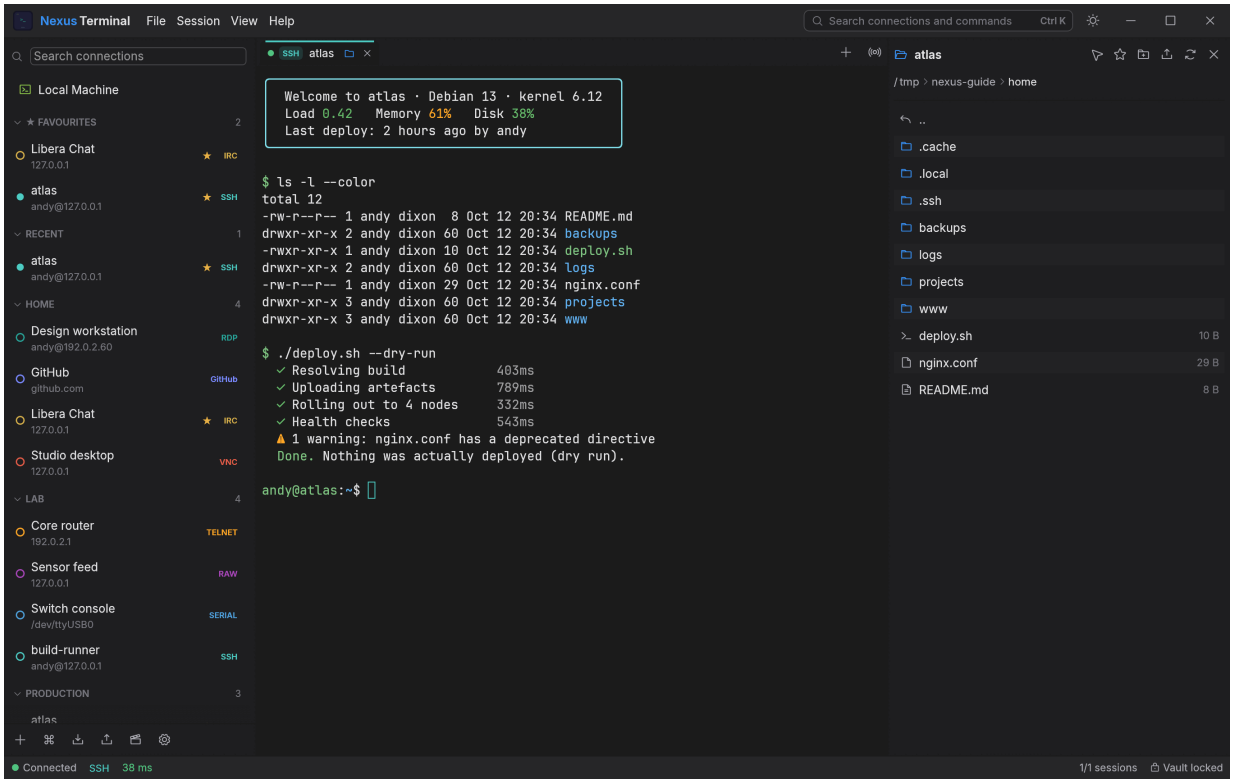
TL;DR

Click the 🗁 on an SSH tab. Or drop a file onto the terminal and choose **Upload**.

4 minutes · You'll move files to and from servers without leaving the terminal

15.1 The SFTP browser

Click the 🗁 folder icon on an SSH tab, **Session > SFTP browser**, or type "sftp" in the palette. It opens in the side dock next to your terminal, in the folder your shell is in if the shell reports it (fish does; bash and zsh need a one-line hook, see the user manual).



Browsing the server's files next to the terminal.

- Move around with the breadcrumb trail, the `..` entry or **Go to path**.
- Upload files; download one or many to a folder you choose.
- Rename, delete (it asks first), make folders, refresh.
- ★ bookmarks the current folder, per profile.

- Ctrl-click (⌘-click) or Shift-click to select several things. Transfers show progress as they go.

15.2 Drag and drop

Drop files onto a terminal and choose:

Upload to current directory Pick the remote folder (SSH only).

Paste contents Types the file's contents into the terminal, as if you'd typed it.

The upload picker looks like the SFTP browser but shows folders only, labelled `READ ONLY` if you can't write there and `NO READ` if you can't look inside. Keyboard fans: `↑` `↓` to move, `Enter` to choose, `Backspace` to go up, `Esc` to cancel. If the SFTP browser is open, the picker starts in the same folder.

Done when you have copied a file to a server and back again.

CHAPTER 16

Port forwards

TL;DR

Local forwards bring a remote port to your machine. *Remote* forwards do the opposite.

3 minutes · You'll reach a database or web app that only listens on the server

Forwards saved in an SSH profile (under **Port Forwarding**, with **Active** ticked) start when you connect. To manage them on a live session, right-click the terminal > **Port Forwards**, or **Session > Port forwards**.

Local → Listen on your computer, forward to somewhere the server can reach. Example: `127.0.0.1:15432` > `127.0.0.1:5432` lets your desktop database app talk to Postgres on the server as if it were local.

Remote ← Ask the server to listen and send connections back to somewhere *your* computer can reach. Great for showing a colleague the dev server running on your laptop.

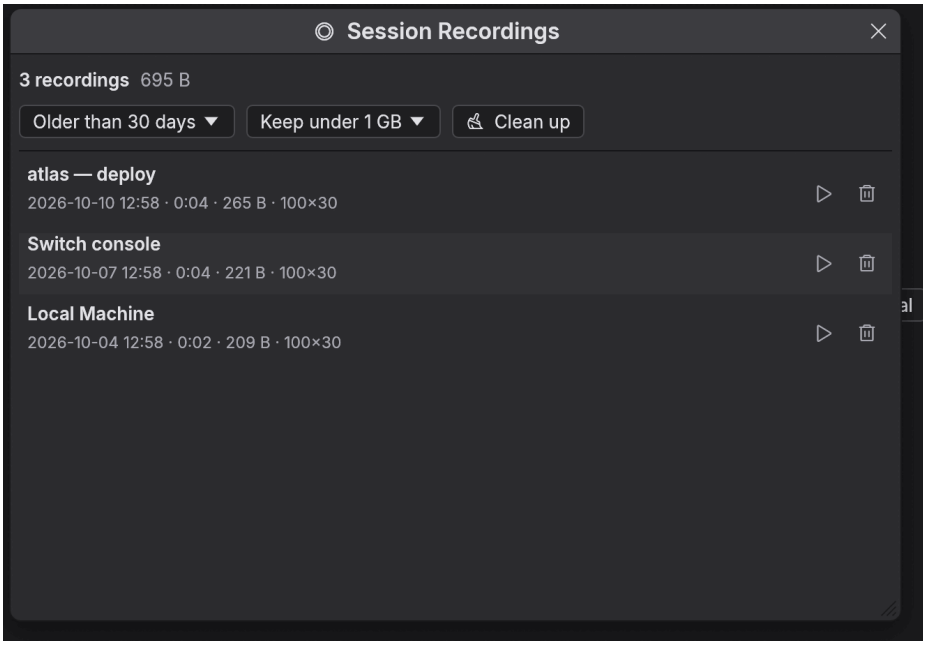
Active forwards are listed with an L or R badge and a stop button. Toasts tell you when one starts, stops or fails.

Done when the remote service opens on a port on your own computer.

CHAPTER 17

Recordings

`Ctrl` / `⌘` + `Shift` + `R` starts and stops recording the active session. A red dot appears on the tab and **REC** in the status bar. Recordings are standard asciicast v2 files, so other tools (like `asciinema`) can play them too.



The recordings browser. Play, delete, or tidy up in bulk.

The browser (sidebar footer, **Session** menu or palette) shows how many recordings you have and how much space they take. Play or delete each one, or use **Clean up** to remove recordings older than 7, 30 or 90 days, or to keep the total under 1 GB or 5 GB. Playback has play/pause, restart, a seek bar and speed control.

> **Pro tip.** Record the next time you do something fiddly and rare (a database failover, a certificate rotation). Six months later, "how did we do that again?" has an answer you can literally watch.

Done when you have replayed a session you recorded.

PART 4

Stay safe

Passwords, keys and host checks, minus the jargon headache.

- Passwords and the vault
- Host keys: is it really that server?
- SSH keys

CHAPTER 18

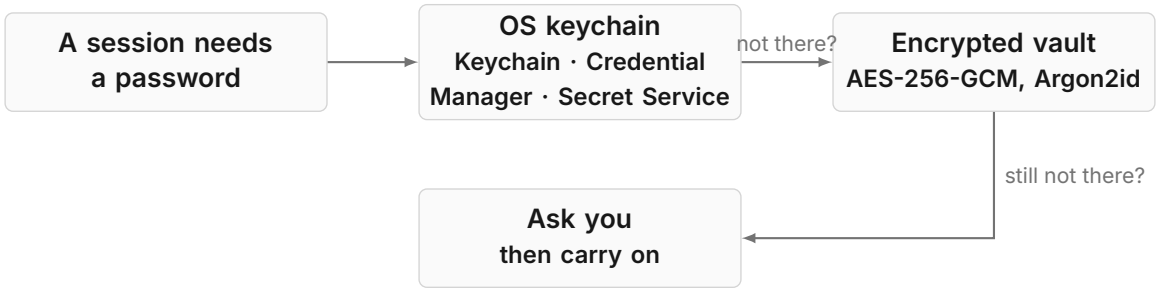
Passwords and the vault

TL;DR

Passwords go in your OS keychain. If there isn't one, they go in an encrypted vault protected by a master password. For each password you choose: save it, remember it until you quit, or ask every time.

5 minutes · You'll know exactly where your passwords live and who can read them

18.1 Where secrets live

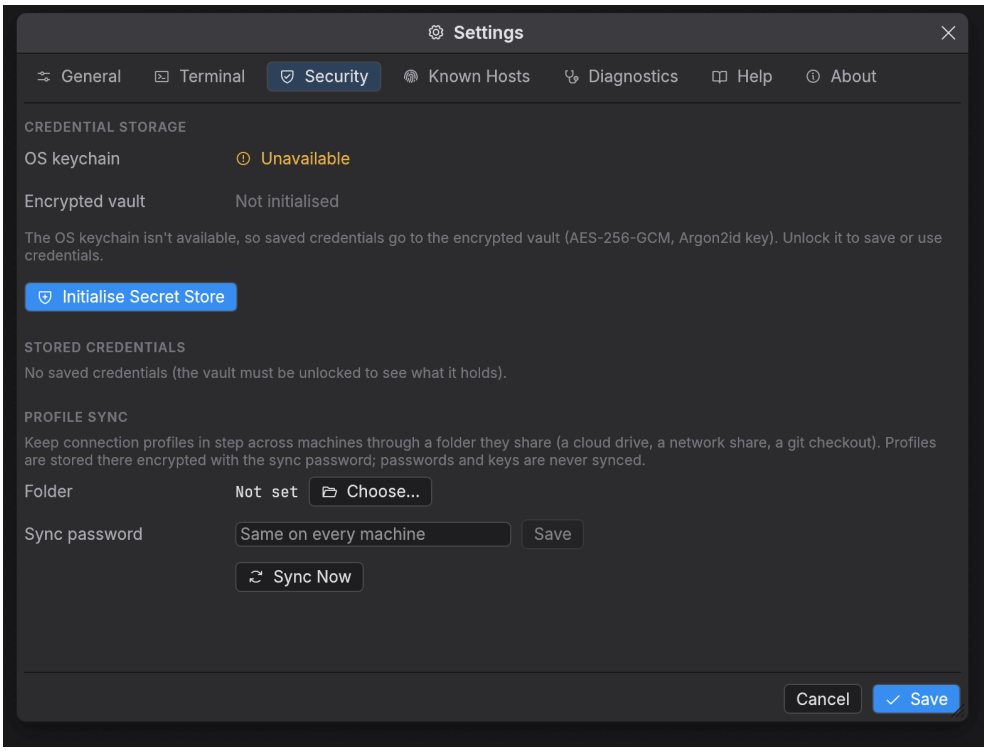


OS keychain first macOS Keychain, Windows Credential Manager, or the Linux Secret Service (GNOME Keyring, KWallet and friends).

Encrypted vault as backup If there's no keychain, or it refuses, secrets go into Nexus Terminal's own vault: AES-256-GCM with a key derived from your master password by Argon2id using 128 MiB of memory. In plain English: very, very expensive to brute-force.

18.2 Setting up the vault

Settings > Security > Initialise Secret Store. Choose a master password; the strength meter goes from `Weak` through `Fair` and `Good` to `Strong`. Unlock and lock it from the same page, from the palette, or by clicking the vault badge in the status bar. It locks itself when you quit.



Security settings on a machine without an OS keychain: the vault does the job instead.

There's no "forgot password" link Lose the master password and the secrets in the vault are gone for good. That's the point of encryption. Put it in your password manager.

18.3 Three ways to handle each password

Every password, passphrase, token or TOTP secret has its own setting:

Save securely
Stored in the keychain or vault and used automatically. Most convenient.

Remember for this session
Asked once, kept in memory, never written to disk.

Ask every time
Asked on every single connection. Most paranoid.

When Nexus Terminal needs a secret it doesn't have, it asks. Press `Esc` or **Cancel** to give up on that connection.

18.4 Seeing and changing what's saved

Stored credentials on the Security page lists every saved password, passphrase and token by profile, and whether it's in the keychain or the vault. **Forget** deletes one. **Change Master Password...** re-encrypts the whole vault under a new password in one go.

What gets written down? · optional; skip it if you are in a hurry

Connections, disconnections, profile changes, credential access, imports and exports go into an audit log in the database. Secrets themselves never do. TOTP secrets live in the keychain or vault too, never in the profile.

Done when you know whether your passwords are in the OS keychain or the vault, and any master password is in your password manager.

CHAPTER 19

Host keys: is it really that server?

TL;DR

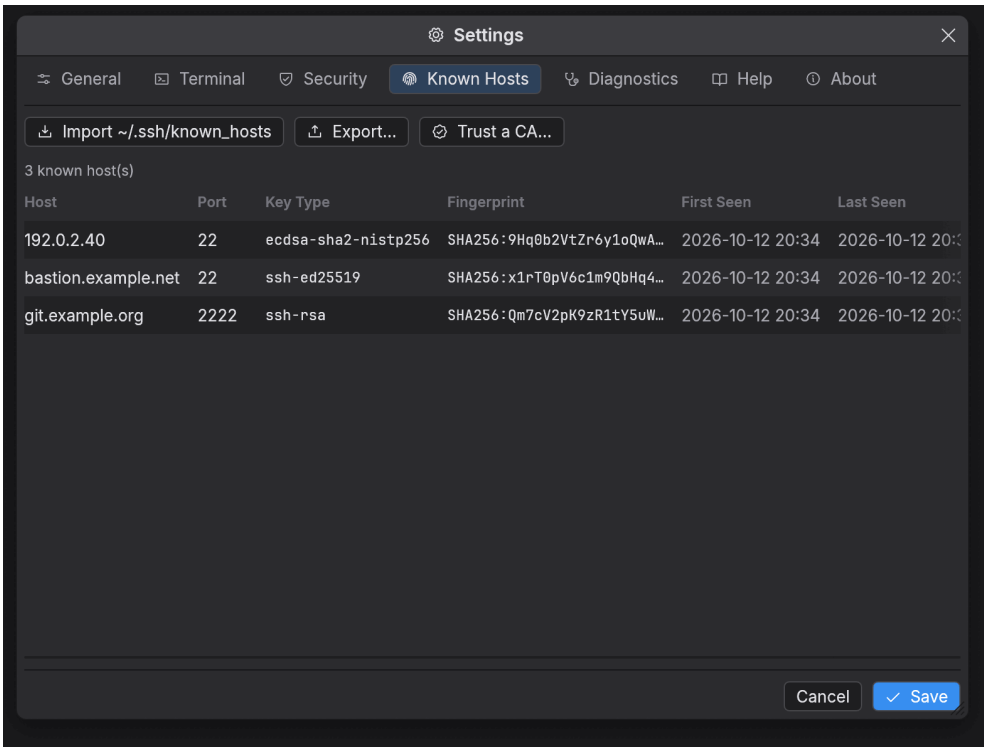
Every SSH server (and every RDP server, via its certificate) has a fingerprint. Nexus Terminal remembers it the first time and shouts if it ever changes. A changed fingerprint means a reinstalled server, or someone in the middle. Find out which before carrying on.

3 minutes · You'll understand the scary fingerprint dialog and what to do about it

Pick a policy per SSH profile under **Host Verification**:

Strict (known hosts only)	Only connects to servers you've already trusted.
Ask on new or changed keys	Shows the fingerprint and asks. New: **Accept & Save** or **Reject**. Changed: **Reject**, **Accept Once** or **Accept & Update**. The default.
Accept new, reject changed	Trusts first-time servers silently, still refuses changed keys.
Insecure (accept anything)	Exactly what it says. Lab use only.

Under *Strict* and *Accept new*, a key that differs from the saved one is rejected outright.



Known hosts. Click a fingerprint to copy it; **Remove** to forget a server.

> **Quick win**
>
> Nexus Terminal keeps its own list, but **Import ~/.ssh/known_hosts** brings in everything your command-line `ssh` already trusts (hashed names can't be read back, so those are skipped). **Export...** goes the other way, in OpenSSH format.

19.1 Certificate authorities

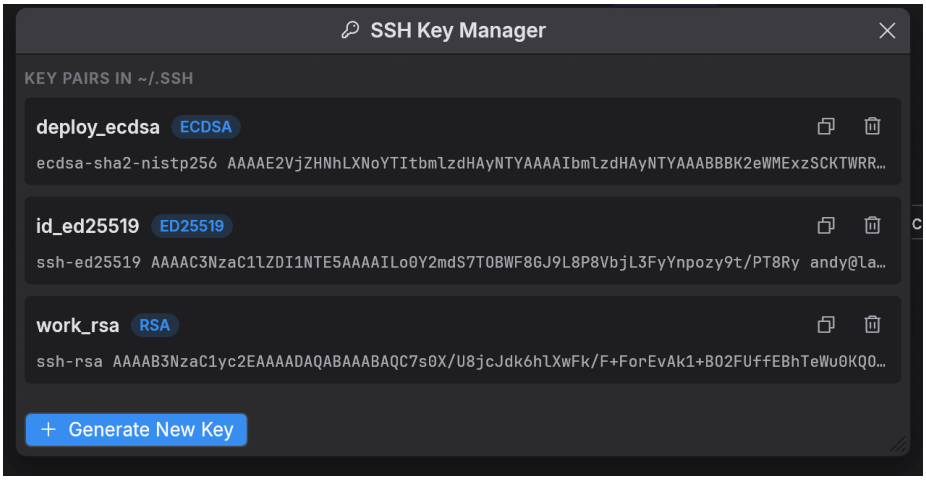
Big estates sign host keys with an SSH certificate authority instead of handing out fingerprints. **Trust a CA...** adds the CA's public key for a host pattern in OpenSSH style (`*.corp.example`, `[jump.example]:2222`, `!old.corp.example`); `@cert-authority` lines in an imported `known_hosts` do the same. A server the CA covers must then show a certificate the CA signed, naming that host and in date. It's trusted without a prompt, and anything else is refused.

Done when you have checked a fingerprint and trusted the server, or refused one you could not vouch for.

CHAPTER 20

SSH keys

Session > SSH key manager (or the palette) lists the key pairs in `~/.ssh` with their type. Copy a public key to the clipboard (ready to paste into GitHub or `authorized_keys`), delete a pair, or generate a new `Ed25519`, `RSA` (2048 or 4096 bits) or `ECDSA` key with an optional passphrase.



The SSH key manager.

Pro tip. Not sure which type? Pick Ed25519. It's small, fast and secure. Keys are generated inside Nexus Terminal and saved in the usual OpenSSH format, so `ssh` and `git` can use them too.

Done when you have a key pair and its public half is on the server.

PART 5

Make it yours

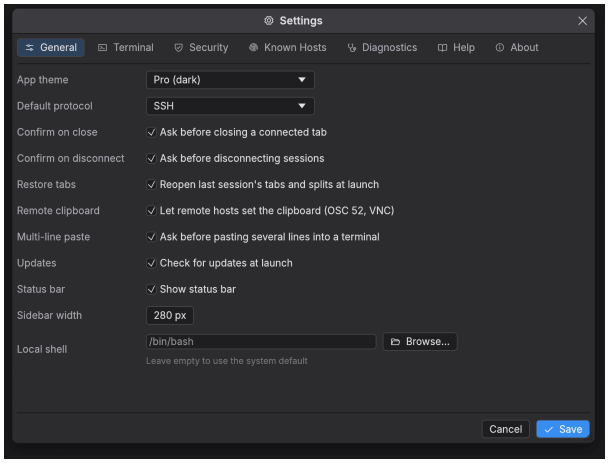
Settings, backups, and keeping several computers in step.

- Settings
- Import and export
- Moving house

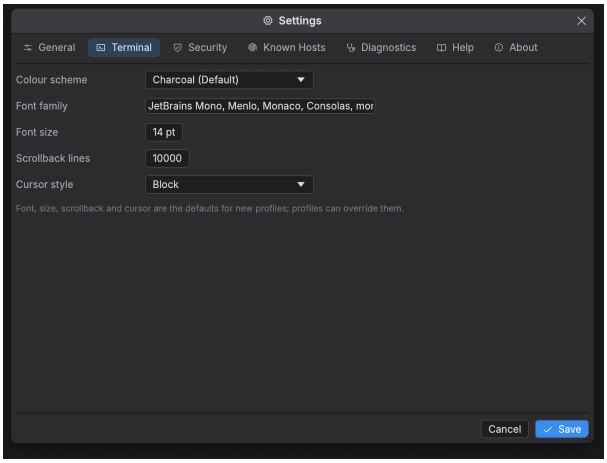
CHAPTER 21

Settings

`Ctrl` / `⌘` + `,` opens settings (also the ⚙ in the sidebar footer and **File > Settings**). Change things, then click **Save**.



General



Terminal

General App theme (Pro, Midnight, Light), default protocol for new profiles, **Restore tabs at launch** (saved connections reconnect, local shells reopen in the folder they were in), whether to ask before closing or disconnecting, whether remote programs may write to your clipboard, a confirmation before pasting several lines, the update check, status bar, sidebar width, and the local shell (empty for the system default, or **Browse...**).

Terminal Colour scheme, font, size, scrollback and cursor style. These are the defaults for new profiles; each profile can override them.

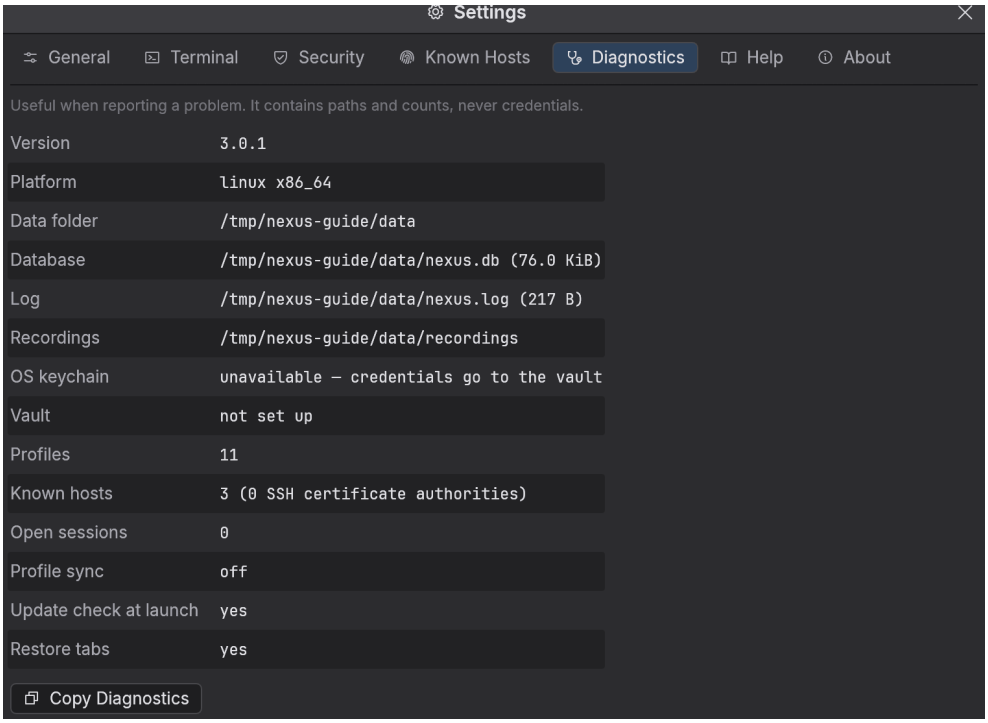
Security Keychain and vault status; set up, unlock or lock the vault; change the master password; your stored credentials; profile sync.

Known Hosts Saved server fingerprints, with first-seen and last-seen times; import and export `known_hosts`; trusted SSH certificate authorities.

Diagnostics Version, platform, where your data and log live, and the state of the keychain, vault and sync. **Copy Diagnostics** for bug reports (never includes credentials).

Help Built-in help: connection types, shortcuts and troubleshooting, when this guide isn't to hand.

About Version, **Check for Updates** and **Import ~/.ssh/config**.



Diagnostics: everything a bug report needs, nothing secret.

3 minutes · You'll know where every setting lives

21.1 Updates

Nexus Terminal checks once at launch. If there's a newer version the status bar says **Update available**; click it to see the release notes and a **View release** button. Nothing downloads or installs behind your back.

Done when you know which page of **Settings** holds the thing you wanted to change.

CHAPTER 22

Import and export

TL;DR

File > Export profiles... makes a password-protected `.nexus` file. **File > Import profiles...** reads it back. Already have an `~/.ssh/config`? Import that too.

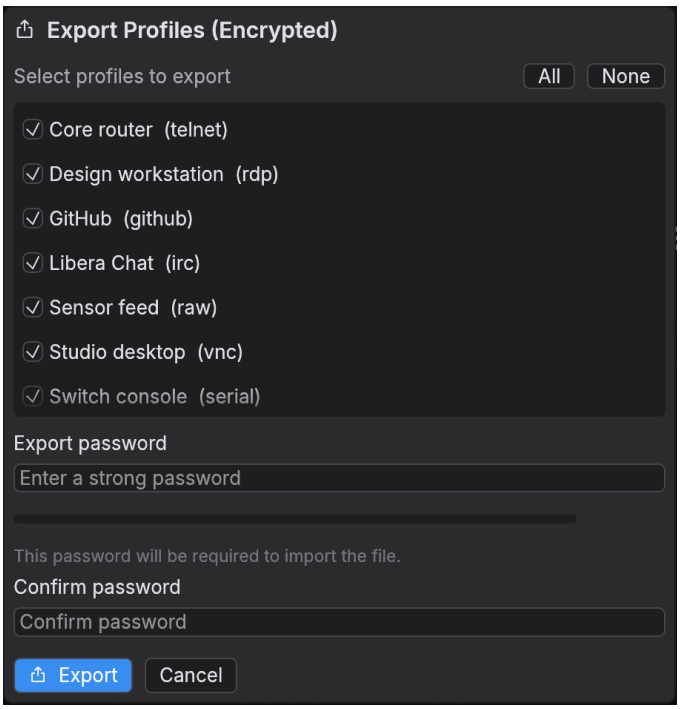
3 minutes · You'll back up your profiles, or copy them to another computer

Encrypted export Choose the profiles, set a password, save a `.nexus` file. Saved passwords aren't included, so the file is safe to email to yourself.

Encrypted import Pick a `.nexus` file, enter its password and decide what happens to profiles that already exist: **Skip existing**, **Overwrite existing** or **Create duplicates**.

JSON **Export Profiles (JSON)** and **Import Profiles (JSON)** are in the palette. Tick **Include secrets** to keep passwords typed by login scripts, which are otherwise blanked. Proxy passwords and TOTP secrets live in the keychain or vault, so no export carries them.

OpenSSH config **File > Import ~/.ssh/config** turns your `Host` blocks into profiles, with host names, users, ports, identity files and jump hosts.



Exporting profiles. Everything's ticked to start with; untick what you don't want.

JSON with secrets is not encrypted A JSON export with **Include secrets** ticked has those login-script passwords in plain text. Use the encrypted `.nexus` export to move things between machines.

Done when an export of your profiles is somewhere safe.

CHAPTER 23

Moving house

TL;DR

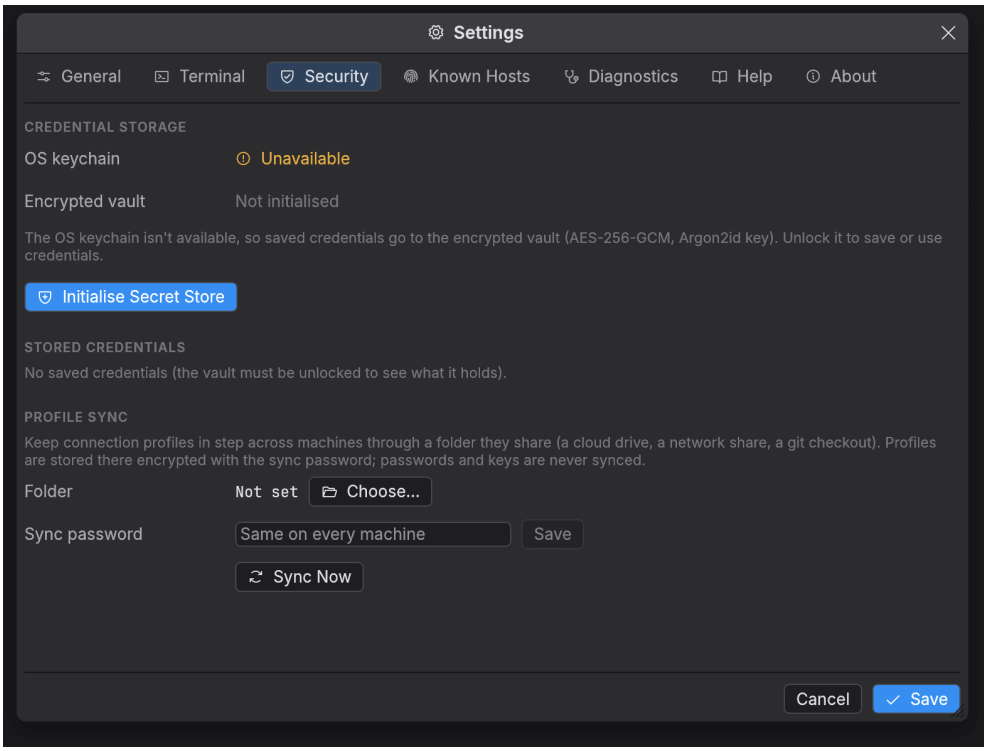
Pick a folder every machine can see, set the same sync password on each, done. Profiles stay in step; passwords stay put on each machine.

3 minutes · Your connections will follow you from laptop to desktop to work machine

23.1 Profile sync

Settings > Security > Profile sync. **Choose...** a folder your machines share (Dropbox, iCloud Drive, OneDrive, Nextcloud, a network share, or a git checkout you commit) and set the same **Sync password** everywhere. Nexus Terminal keeps one encrypted file there, `nexus-profiles.nexus`, and merges it at launch, a few seconds after you change a profile, every ten minutes and whenever you click **Sync Now**.

- ✓ Edited on two machines? The newer change wins, profile by profile.
- ✓ Deleted on one machine? It goes from the others too (unless someone edited it afterwards).
- ✗ Saved passwords, keys and tokens never sync. Each machine keeps its own in its keychain or vault. (Anything a login script *sends* is part of the profile, so that does travel.)



Profile sync lives on the Security page, under your stored credentials.

⚠ **Watch out.** Lose the sync password and the sync file can't be read, just like an encrypted export. Your local profiles are fine.

23.2 One-off moves

Not worth setting up sync? **File > Export profiles...** on the old machine, **File > Import profiles...** on the new one (see Import and export). Known hosts travel the same way through **Export...** and **Import** on the Known Hosts page.

23.3 Where everything lives

macOS	`~/Library/Application Support/NexusTerminal`
Windows	`%APPDATA%\NexusTerminal`
Linux	`$XDG_CONFIG_HOME/NexusTerminal`, usually `~/.config/NexusTerminal`

Inside: `nexus.db` (profiles, known hosts, preferences, snippets, IRC state and the audit log), `vault/vault.json`, `recordings/`, `triggers.json`, `sftp-bookmarks.json` and `nexus.log`, the diagnostic log. Keychain entries are stored under the service name `NexusTerminal`.

Try things without touching your real data Start Nexus Terminal with `--data-dir /some/other/folder` and it uses that folder instead. Brilliant for experiments, demos and a portable install on a USB stick.

Done when a profile you add on one computer turns up on the other.

PART 6

When things go wrong

Symptom first, fix second. No essays.

Troubleshooting, by symptom

CHAPTER 24

Troubleshooting, by symptom

TL;DR

Find the heading that matches what you're seeing. Try the fixes in order. Most problems are the network, the password, or the baud rate.

as long as it takes · You'll find your symptom and the fix in one place

24.1 Connecting

"Authentication failed"

- Check the username, password, key path and the ticked **Auth Methods**.
- If the key has a passphrase, check how the passphrase is stored (**Passphrase Mode**).
- Two-factor server? Click **Test** next to the TOTP secret and compare with your phone. Check your clock.
- Saved password not being used? The vault might be locked: click **Vault locked** in the status bar.

"Host key verification failed"

The server's fingerprint changed since you last connected. That happens when a server is rebuilt, but it's also exactly what an attack looks like. **Check with whoever runs the server.** If it's legitimate: choose **Accept & Update** in the warning (or **Accept Once** to defer the decision). Under a Strict or Accept-new policy there's no prompt: **Settings > Known Hosts**, **Remove** the entry and reconnect.

It just says "Connecting…" forever

- Click **Test** in the profile editor. If that fails too, it's the network, a firewall or the wrong port.
- Behind a corporate proxy? Set one up under **Proxy** (SSH profiles).
- Going through a bastion? Check each jump host on its own first.

The session keeps reconnecting

Check the network and the credentials, then the profile's auto-reconnect settings. Reconnection backs off (1 s, 2 s, 4 s… up to 30 s) and stops after the maximum number of attempts.

24.2 Serial

The port won't open

- Is the adapter plugged in? Hit refresh in the device list.
- **Permission denied** on Linux: add yourself to `dialout` (or `uucp`), then log out and back in.
- On a Mac, use `/dev/cu.*`, not `/dev/tty.*`.
- **Port is in use by another program**: close `screen`, `minicom`, the Arduino IDE serial monitor or whatever else has it open.

Gibberish on screen

The baud rate, parity or stop bits don't match the device. Check its manual.

24.3 Remote desktops

VNC won't connect

- VNC ports are 5900 plus the display number: `:0` is 5900, `:1` is 5901.
- Check the password and how it's stored.
- Using an SSH tunnel? Check the tunnel's host, user, login method and key, and the host key policy.
- Connecting to a Mac? Turn on password access in macOS Screen Sharing settings (see p. 26).

RDP won't connect

- Linux box running xrdp? Tick **Allow servers without NLA**.
- "Logon failed"? Check the domain. Kerberos users: the KDC must be reachable directly, not through the proxy.
- Certificate warning after a server rebuild? That's expected once; compare the fingerprint, then **Accept & Update**.
- Going through an RD Gateway? It needs a certificate from a public CA and Basic authentication over HTTPS.

24.4 Files

Upload is greyed out or fails

The folder is probably READ ONLY for your user. Pick another, or check permissions on the server. Uploads need an SSH session; other protocols can't do it.

“NO READ” on a folder

You don’t have permission to look inside. Try the parent folder, or ask the server’s admin.

24.5 The terminal itself

The terminal is the wrong size

Resize the window or nudge a split divider; the terminal re-fits itself. Still wrong? Reconnect. Very old servers may prefer a different **Terminal Type** in the profile (try `vt100`).

The scroll wheel moves the cursor instead of scrolling

You’re in a full-screen program (`less`, `vim`, `tmux`). That’s how terminals work: the wheel sends arrow keys there. Quit the program, or use its own scrolling (in `tmux`: `Ctrl` + `B` then `[`).

Emoji are in black and white

Known limitation: colour emoji show as monochrome glyphs. Everything else renders normally.

▷ Still stuck?

Start Nexus Terminal with `–data-dir` pointing at an empty folder and try the same connection there. If it works, something in your profile is off; duplicate the working one and compare.

24.6 Still stuck?

Something odd that isn’t on this list

Look at `nexus.log` in your data folder (**Settings > Diagnostics** shows where): connections, reconnects, host-key checks and errors, never passwords or what’s on screen. Start Nexus Terminal with `RUST_LOG=debug` for more detail. Reporting a bug? **Copy Diagnostics** and paste it in.

✓ **Done when** the thing that was broken works again, or the diagnostic log is ready to send.

PART 7

The extras

The cheat sheet, the command line, and words you might trip over.

- Keyboard cheat sheet
- Command-line options
- Glossary

CHAPTER 25

Keyboard cheat sheet

Everywhere

Command palette	`Ctrl` / `⌘` + `K`
New connection	`Ctrl` / `⌘` + `N`
Quick Connect	`Ctrl` / `⌘` + `Shift` + `Q`
Settings	`Ctrl` / `⌘` + `,`
Full screen	`F11`

Tabs & panes

Close tab	`Ctrl` / `⌘` + `W`
Next tab	`Ctrl` / `⌘` + `Tab`
Previous tab	`Ctrl` / `⌘` + `Shift` + `Tab`
Go to tab 1–9	`Ctrl` / `⌘` + `1` … `9`
Split right	`Ctrl` / `⌘` + `Shift` + `D`
Split down	`Ctrl` / `⌘` + `Shift` + `E`

Power tools

Broadcast on/off	`Ctrl` / `⌘` + `Shift` + `B`
Snippets	`Ctrl` / `⌘` + `Shift` + `S`
Record on/off	`Ctrl` / `⌘` + `Shift` + `R`

In the terminal

Copy selection	`Ctrl` / `⌘` + `C`
Paste	`Ctrl` / `⌘` + `V`
Find	`Ctrl` / `⌘` + `F`
Next / previous match	`Enter` / `Shift` + `Enter`
Scroll a page	`Shift` + `PageUp`
Top / bottom	`Shift` + `Home` / `End`
Previous / next prompt	`Ctrl` / `⌘` + `Shift` + `Up` / `Down`
Open link	`Ctrl` / `⌘` + click

Mouse moves

Select word / line	double / triple-click
Extend selection	`Shift` + click
Select several profiles	`Ctrl` / `⌘` + click
Everything else	right-click it

Done when the shortcuts you use most are ones you no longer look up.

CHAPTER 26

Command-line options

`–data-dir DIR`	Use another data folder (portable installs, demos, testing).
`–theme pro|midnight|light`	Start with this theme.
`–connect NAME`	Connect to a saved profile by name at startup. Repeat it for several.
`–open WINDOW`	Open a window at startup: `settings`, `settings:security`, `settings:diagnostics`, `snippets`, `keys`, `recordings`, `palette`, `quick`…
`–run-local CMD`	Open a local shell and run a command in it.
`RUST_LOG=debug`	(environment) More detail in `nexus.log`, and on the terminal you started from.
`–version`	Print the version and exit.

How the screenshots in this guide were made · optional; skip it if you are in a hurry

Every screenshot here is the real app, driven by `docs/guide/tools/screenshots.sh`: it starts throwaway SSH, IRC, VNC and GitHub-ish servers, fills a demo data folder, and runs Nexus Terminal headless with `–screenshot`, `–do` (run a palette command) and `–answer` (fill a password prompt; only accepted together with `–screenshot`). Change the app, rerun the script, rebuild the guide.

Done when you can start Nexus Terminal from a script with the flags you need.

CHAPTER 27

Glossary

Agent A background program holding your unlocked SSH keys, so you type the passphrase once.

Asciicast The file format for terminal recordings. Plays in Nexus Terminal and `asciinema`.

Bastion / jump host A server you go *through* to reach the ones behind it.

Baud rate How fast a serial line talks. Both ends must agree.

Broadcast Typing into several sessions at once.

Certificate authority A key that vouches for other keys, so servers signed by it are trusted without a prompt.

DTR / RTS Serial control lines. Some boards use them for reset or flashing.

Fingerprint A short summary of a server's host key, used to spot imposters.

Host key A server's identity. Changes when it's rebuilt, or when someone's pretending.

Keychain Your operating system's password store.

NLA Network Level Authentication: Windows checks your password before building a desktop.

OSC 133 The escape codes a shell uses to mark its prompts, which powers shell integration.

Port forward A tunnel carrying one TCP port through SSH.

Profile A saved connection.

RAW TCP A plain socket: bytes in, bytes out, no terminal rules.

RDP Microsoft's Remote Desktop Protocol, for Windows (and xrdp) desktops.

SASL How IRC networks log you in while you connect, instead of messaging NickServ afterwards.

SFTP File transfer over SSH.

Snippet A saved command, optionally with blanks to fill in.

TOTP Time-based one-time passwords: the six digits from your authenticator app.

Trigger A rule that watches output and reacts to a pattern.

Vault Nexus Terminal's encrypted password store, for when there's no keychain.

VNC A way of seeing and controlling another computer's desktop.

✓ **Done when** none of the words in this guide trip you up.



That's the lot.

Go and connect to something. 🚀

Every terminal you need. One window.
Go and connect to something.

uncomplex. Complicated things. Thoughtfully simplified.