Terminfo & TERM over SSH

Modern terminal emulators — Ghostty, Kitty, Alacritty — advertise their own terminal type (`TERM=xterm-ghostty`, `xterm-kitty`, and so on). A remote server can only drive a terminal it has a **terminfo** entry for, so the moment you land on a box that has never heard of `xterm-ghostty`, full-screen programs bail out with `Error opening terminal: xterm-ghostty`. The classic tell is that `nano` works but `sudo nano` doesn't — because `sudo` throws away the environment that made it work.

On This Page

- The symptom
- Why it happens
- Quick fix — one-off
- Permanent fix A — copy the terminfo to the server
- Permanent fix B — let sudo keep your TERMINFO
- Permanent fix C — report xterm-256color
- Diagnose it
- Which fix should I use?

The symptom

```
$ sudo nano /etc/hosts
Error opening terminal: xterm-ghostty.
```

You run	Result	Why
`nano file`	works	your login shell has `TERM` **and** a terminfo it can find
`sudo nano file`	**fails**	`sudo` resets the environment; `root` can't find `xterm-ghostty`
`vim`, `top`, `less`, `tmux`	may fail the same way	any curses / full-screen program needs terminfo

Why it happens

Two things line up:

1. `TERM` **names a terminal the server doesn't know.** Ghostty sets `TERM=xterm-ghostty`. The server looks that name up in its **terminfo** database to learn how to move the cursor, clear the screen, and do colour. No entry means `Error opening terminal`.
2. `sudo` **starts a clean environment.** Your login shell probably had a terminfo it *could* find — Ghostty's SSH integration drops one into `~/.terminfo`, and `TERMINFO` / `TERMINFO_DIRS` may point at it. `sudo` strips those variables and runs as `root`, which only searches the system database, where `xterm-ghostty` usually isn't.

Whose lookup	Searches	Has `xterm-ghostty`?
your user	`~/.terminfo`, then `$TERMINFO` / `$TERMINFO_DIRS`, then the system dirs	often yes (injected on connect)
`root` via `sudo`	system dirs only — `/usr/share/terminfo`, `/lib/terminfo`, `/etc/terminfo`	usually **no**

Quick fix — one-off

Override `TERM` for that single command with a type every server understands:

```
sudo TERM=xterm-256color nano /etc/hosts
```

Nothing to install, nothing to configure. You only lose Ghostty-specific niceties (a few colours or key sequences) for that one command.

Permanent fix A — copy the terminfo to the server

Keep the real `xterm-ghostty` features by putting its terminfo where the server — and `root` — can find it. The cleanest approach exports your local entry and compiles it on the remote in one line, with no files to move by hand:

```
# run on your LOCAL machine
infocmp -x xterm-ghostty | ssh user@server -- tic -x -
```

Already SSH'd in, with the entry sitting in your `~/.terminfo`? Copy it into a system location instead:

```
# into root's own database
sudo mkdir -p /root/.terminfo/x
sudo cp -a ~/.terminfo/x/xterm-ghostty /root/.terminfo/x/

# …or system-wide, for every user on the box
sudo cp -a ~/.terminfo/x/xterm-ghostty /usr/share/terminfo/x/
```

> Distros disagree on the system path — it may be `/lib/terminfo/x/` or `/etc/terminfo/x/` instead. Run `infocmp -D` to print the directories *this* server actually searches.

Permanent fix B — let sudo keep your TERMINFO

If the entry is already reachable from your user (for example in `~/.terminfo`), tell `sudo` to carry `TERMINFO` across instead of stripping it:

```
sudo TERM=xterm-256color visudo          # the override just lets visudo open
```

Add one line — ideally as a drop-in like `/etc/sudoers.d/terminfo`, so a package upgrade can't clobber it:

```
Defaults    env_keep += "TERMINFO TERMINFO_DIRS"
```

Now `sudo nano` inherits your terminfo path and finds `xterm-ghostty`.

Permanent fix C — report xterm-256color

Tired of doing this on every new box? Tell the emulator to identify as the universal type. You give up a few bleeding-edge features in exchange for compatibility everywhere.

Terminal	Config file	Line to add
Ghostty	`~/.config/ghostty/config`	`term = xterm-256color`
Kitty	`~/.config/kitty/kitty.conf`	`term xterm-256color`
Alacritty	`~/.config/alacritty/alacritty.toml`	`[env]` → `TERM = "xterm-256color"`

Restart the terminal afterwards. Kitty and Ghostty also ship SSH integration — `kitten ssh user@host`, or Ghostty's SSH feature — that copies the terminfo to the remote for you, so you keep the native `TERM` without the errors.

Diagnose it

```
echo "$TERM"                          # what your terminal claims to be
infocmp "$TERM" >/dev/null && echo ok || echo "no terminfo for $TERM"
infocmp -D                            # the terminfo directories searched, in order
toe | grep -i ghostty                 # is any ghostty entry installed?
sudo sh -c 'echo "$TERM"; infocmp -D' # what root sees (TERM is often reset first)
```

Command	Tells you
`echo $TERM`	the terminal type being advertised
`infocmp $TERM`	dumps the entry — errors if it's missing
`infocmp -D`	the exact terminfo search path on this host
`tic -x file`	compile / install a terminfo source (`-x` keeps extended caps)
`toe`	list every terminfo the system knows about

Which fix should I use?

Situation	Best fix
Just need to edit one file, right now	Quick fix: `sudo TERM=xterm-256color …`
Your own server, want the full terminal	Fix A: copy the terminfo (system-wide or to `/root`)
Entry already in your `~/.terminfo`	Fix B: `env_keep += "TERMINFO"`
You hop across many servers	Fix C: report `xterm-256color`, or use `kitten ssh`