Mosh Roaming remote shell

START HERE

`mosh <host>`
Connect

`mosh <user>@<host>`
Connect as a user

`mosh --ssh="ssh -p <port>" <host>`
Custom SSH port

`mosh -p <port> <host>`
Pin the UDP port

`mosh <host> -- tmux a`
Attach tmux

`Ctrl+^ .`
Kill a hung session

`sudo apt install mosh`
Install (both ends)

`locale | grep UTF-8`
Check the locale

mosh client local → SSH (TCP 22) → **Login** starts mosh-server → UDP 60000-61000 → **mosh-server** your shell → roams with you → **Session** survives IP changes

CONCEPTS

SSH only logs in and starts mosh-server; the session itself runs over UDP, so it survives sleep, Wi-Fi changes and roaming.

Only the visible screen is synced, so there is no scrollback. Run tmux or screen on the server for history.

Both client and server need mosh installed, and UDP must get through. Windows has no native client (use WSL or Cygwin).

CONNECT

`mosh <host>`
Connect to a host

`mosh <user>@<host>`
Connect as another user

`mosh <host> -- <command>`
Run a command instead of the shell

`mosh --ssh="ssh -p <port>" <host>`
Use a non-default SSH port

`mosh --ssh="ssh -i <keyfile>" <host>`
Pass SSH options such as a key

`mosh <host> -- tmux new -As <name>`
Attach to or create a tmux session

`mosh --server=<path> <host>`
Path to mosh-server on the remote
Needed when mosh-server is not on the login PATH, e.g. in a home directory or on macOS and BSD servers.

`mosh -4 <host>`
IPv4 only

`mosh -6 <host>`
IPv6 only

`mosh --family=<mode> <host>`
IPv4/IPv6 selection (auto, all, prefer-inet6)

`mosh --no-ssh-pty <host>`
Run SSH without a pty

`mosh --client=<path> <host>`
Path to the local mosh-client

PORTS AND FIREWALL

`mosh -p <port> <host>`
Use one server UDP port

`mosh -p <low>:<high> <host>`
Use a UDP port range

`mosh -p 0 <host>`
Let the OS pick a UDP port

`sudo ufw allow 60000:61000/udp`
Open the mosh range in ufw

`sudo firewall-cmd --add-port=60000-61000/udp --permanent`
Open the range in firewalld

`sudo iptables -A INPUT -p udp --dport 60000:61000 -j ACCEPT`
Open the range in iptables

`mosh --bind-server=<ssh|any|IP> <host>`
Which address mosh-server binds to
ssh (default) answers from the address SSH arrived on; any helps behind sslh-style setups.

`mosh --experimental-remote-ip=<mode> <host>`
How the client finds the IP (proxy, local, remote)

ESCAPE KEY

`Ctrl+^ .`
Kill the connection
Typed as Ctrl+Shift+6 on a US keyboard.

`Ctrl+^ Ctrl+Z`
Suspend the client

`Ctrl+^ ^`
Send a literal Ctrl+^

`MOSH_ESCAPE_KEY=<char> mosh <host>`
Change the escape character
One literal ASCII character; useful on keyboards where Ctrl+^ is awkward.

Escape char
Escape key sequences work at any time. A printable escape char (like ~) must follow a newline.

LOCAL ECHO

`mosh --predict=adaptive <host>`
Echo locally only on slow links (default)

LOCAL ECHO (cont.)

`mosh -a <host>`
Always predict typing

`mosh -n <host>`
Never predict typing

`mosh -o <host>`
Overwrite text instead of inserting

`MOSH_PREDICTION_DISPLAY=<mode>`
Set the prediction mode by env var

`MOSH_TITLE_NOPREFIX=1`
Drop the [mosh] window title prefix

`mosh --no-init <host>`
Keep the alternate screen out of use
Skips the terminal's enter/exit alternate-screen strings so output stays in your terminal's scrollback.

SERVER

`sudo apt install mosh`
Install on Debian or Ubuntu

`sudo dnf install mosh`
Install on Fedora

`sudo pacman -S mosh`
Install on Arch

`brew install mosh`
Install on macOS

`mosh-server --version`
Check mosh-server is found

`locale`
Check a UTF-8 locale is set
Mosh refuses to start without a UTF-8 locale on both sides.

`export LC_ALL=en_US.UTF-8`
Set a UTF-8 locale

`pgrep -a mosh-server`
List running sessions on the server

`pkill -SIGUSR1 mosh-server`
Clean up disconnected sessions
With MOSH_SERVER_SIGNAL_TMOUT set, sessions with a live client ignore it.

`export MOSH_SERVER_NETWORK_TMOUT=<seconds>`
Exit after no client contact for that long

`mosh-server new -p <port>`
Start a server by hand on a port

SERVER (cont.)

`mosh-server new -c 256`
Advertise 256 colours

TROUBLESHOOTING

`mosh-server new`
Test it starts and prints a port and key

`mosh --local 127.0.0.1`
Test mosh without SSH

`ss -lunp | grep mosh`
Check the server's UDP socket

Common errors

Message	Likely cause
Nothing received from server on UDP port	Firewall blocks UDP
mosh-server: command not found	Not on remote PATH, use --server
needs a UTF-8 native locale	Set LC_ALL on both ends
Connection to server lost	Wait, it resumes on its own

No UDP on the network (some hotels, corporate Wi-Fi) means no mosh. Fall back to plain SSH inside tmux.

RECIPES

Roaming tmux session

```
# survives laptop sleep and network changes
mosh <user>@<host> -- tmux new -As main
```

Open the firewall (ufw)

```
sudo ufw allow OpenSSH
sudo ufw allow 60000:61000/udp
sudo ufw status
```

Smaller port range

```
# fewer sessions, fewer open ports
sudo ufw allow 60000:60010/udp
mosh -p 60000:60010 <host>
```

Config via ssh_config alias

```
# ~/.ssh/config
Host box
    HostName <host>
    Port 2222
    User <user>

mosh --ssh="ssh -p 2222" box
```