Mosh Roaming remote shell

CONCEPTS

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

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

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

mosh -4 *<host>*
IPv4 only

mosh -6 *<host>*
IPv6 only

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

PORTS AND FIREWALL

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

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

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

ESCAPE KEY

Ctrl+^ .
Kill the connection

Ctrl+^ Ctrl+Z
Suspend the client

MOSH_ESCAPE_KEY=*<char>* **mosh** *<host>*
Change the escape character

LOCAL ECHO

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

mosh -a *<host>*
Always predict typing

mosh -n *<host>*
Never predict typing

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

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

SERVER (cont.)

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

pkill -SIGUSR1 mosh-server
Clean up disconnected sessions

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

TROUBLESHOOTING

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

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>
```