OpenSSH Secure remote shell

CONCEPTS

Put per-host settings in ~/.ssh/config so a long command becomes just: ssh <alias>.

Client ssh + ssh-agent
↓ encrypted
Jump host optional bastion
↓ forwarded
Server sshd, authorized_keys

Keep the private key secret; only the .pub file ever leaves your machine.

CONNECT

ssh *<user>*@*<host>*
Log in to a host

ssh *<host>*
Log in as your local user name

ssh -p *<port>* *<user>*@*<host>*
Use a non-default port

ssh *<host>* *<command>*
Run a command and return

ssh -i *<keyfile>* *<host>*
Pick a specific private key

ssh -t *<host>* *<command>*
Force a terminal (sudo, top, tmux)

ssh -v *<host>*
Show debug output (up to -vvv)

KEYS

ssh-keygen -t ed25519 -C "*<comment>***"**
Generate an Ed25519 key pair

ssh-copy-id *<user>*@*<host>*
Install your public key on a host

ssh-keygen -R *<host>*
Forget a host in known_hosts

AGENT

eval "$(ssh-agent)"
Start an agent in this shell

ssh-add
Add the default key to the agent

ssh-add *<keyfile>*
Add a specific key

AGENT (cont.)

ssh-add -l
List loaded keys

COPY FILES

scp *<file>* *<user>*@*<host>*:*<path>*
Upload a file

scp *<user>*@*<host>*:*<path>* *<dir>*
Download a file

scp -r *<dir>* *<host>*:*<path>*
Copy a directory

TUNNELS

ssh -L *<lport>*:*<dest>*:*<dport>* *<host>*
Local forward: reach dest via host

ssh -R *<rport>*:*<dest>*:*<dport>* *<host>*
Remote forward: expose a local port on host

ssh -D *<port>* *<host>*
Open a local SOCKS proxy

ssh -N *<host>*
Forward only, run no command

ssh -f -N -L *<lport>*:*<dest>*:*<dport>* *<host>*
Tunnel in the background

ssh -J *<jump>* *<host>*
Connect through a jump host

CONFIG

Host *<alias>*
Start a config block

HostName *<name-or-ip>*
Real address of the alias

User *<user>*
Default login name

IdentityFile *<path>*
Key to use for this host

ProxyJump *<jump>*
Always hop via this host

ESCAPE KEYS

Enter ~.
Kill a hung session