SSH Quick Reference

Connections, keys, config, tunnels, SCP, SFTP

Connecting

Basic Connection

```
ssh user@host                   # connect to host
ssh -p 2222 user@host           # custom port
ssh user@host command            # run remote command
ssh -t user@host "top"          # force TTY allocation
```

Connection Flags

-p port	Connect to specific port
-i key	Use specific identity (private key)
-t	Force pseudo-terminal allocation
-v / -vv / -vvv	Verbose debugging (increasing detail)
-q	Quiet mode (suppress warnings)
-N	No remote command (for tunnels)
-f	Go to background before command
-J jump	Jump host (ProxyJump)

Key Management

Generate Keys

```
ssh-keygen -t ed25519 -C "you@example.com"
ssh-keygen -t rsa -b 4096 -C "you@example.com"
ssh-keygen -t ed25519 -f ~/.ssh/mykey
ssh-keygen -p -f ~/.ssh/id_ed25519  # change passphrase
```

Deploy Public Key

```
ssh-copy-id user@host
ssh-copy-id -i ~/.ssh/mykey.pub user@host
# Manual: append .pub to remote authorized_keys
cat ~/.ssh/id_ed25519.pub | ssh user@host \
  "mkdir -p ~/.ssh && cat >> ~/.ssh/authorized_keys"
```

Key Files

~/.ssh/id_ed25519	Private key (keep secret)
~/.ssh/id_ed25519.pub	Public key (share freely)
~/.ssh/authorized_keys	Remote: accepted public keys
~/.ssh/known_hosts	Known host fingerprints

Config File

~/.ssh/config Basics

```
Host myserver
  HostName 192.168.1.100
  User deploy
  Port 2222
  IdentityFile ~/.ssh/deploy_key

# Then connect with just:
# ssh myserver
```

Useful Config Options

```
Host *
  ServerAliveInterval 60
  ServerAliveCountMax 3
  AddKeysToAgent yes
  IdentitiesOnly yes

Host bastion
  HostName bastion.example.com
  User admin
```

Config Directives

Host	Alias pattern for the entry
HostName	Actual hostname or IP
User	Login username
Port	Remote port (default 22)
IdentityFile	Path to private key
ProxyJump	Jump through another host
ServerAliveInterval	Keep-alive interval (seconds)
IdentitiesOnly	Only use specified keys

Port Forwarding

Local Forwarding (-L)

```
# Access remote port 5432 via local port 5432
ssh -L 5432:localhost:5432 user@host
# Access remote-db:3306 through ssh host
ssh -L 3306:remote-db:3306 user@host
# Bind to all interfaces
ssh -L 0.0.0.0:8080:localhost:80 user@host
```

Remote Forwarding (-R)

```
# Expose local port 3000 on remote port 8080
ssh -R 8080:localhost:3000 user@host
# Allow remote connections from any interface
ssh -R 0.0.0.0:8080:localhost:3000 user@host
```

Dynamic Forwarding (-D)

```
# SOCKS5 proxy on local port 1080
ssh -D 1080 user@host
# Background SOCKS proxy
ssh -D 1080 -fN user@host
```

SCP & SFTP

SCP (Secure Copy)

```
scp file.txt user@host:/remote/path/
scp user@host:/remote/file.txt ./local/
scp -r dir/ user@host:/remote/path/
scp -P 2222 file.txt user@host:/path/
```

SFTP (Interactive Transfer)

```
sftp user@host
# Inside sftp session:
# put local.txt    — upload file
# get remote.txt   — download file
# ls / lcd / cd    — list / change directory
```

Transfer Flags

-r	Recursive (copy directories)
-P port	Specify port (SCP uses -P, not -p)
-C	Enable compression
-l limit	Bandwidth limit in Kbit/s
-i key	Use specific identity file

Agent Forwarding

SSH Agent

```
eval "$(ssh-agent -s)"          # start agent
ssh-add ~/.ssh/id_ed25519       # add key to agent
ssh-add -l                      # list loaded keys
ssh-add -D                      # remove all keys
```

Forwarding Agent

```
ssh -A user@host                # forward agent
# Or in ~/.ssh/config:
# Host myserver
#   ForwardAgent yes
```

Agent Notes

> Agent forwarding lets the remote host use your local keys without copying them. Use only with trusted hosts. Prefer ProxyJump over agent forwarding when possible.

Tunnels

Persistent Tunnel

```
# Background tunnel that stays open
ssh -fNT -L 5432:localhost:5432 user@host
# Auto-reconnecting tunnel (with autossh)
autossh -M 0 -fNT -L 5432:localhost:5432 user@host
```

Jump Hosts / Bastion

```
ssh -J bastion user@internal-host
ssh -J user1@hop1,user2@hop2 user@target
# Config equivalent:
# Host internal
#   HostName 10.0.0.5
#   ProxyJump bastion
```

Tunnel Management

-f	Background after authentication
-N	No remote command
-T	Disable pseudo-terminal
~.	Kill stuck SSH session (escape)
~C	Open command line for forwarding
~#	List forwarded connections

Troubleshooting

Debugging Connection

```
ssh -vvv user@host              # max verbosity
ssh -G user@host                # dump config (dry run)
ssh-keyscan host                # fetch host keys
ssh-keygen -R host              # remove from known_hosts
```

Common Issues

Permission denied	Wrong key, user, or ~/.ssh perms (700/600)
Host key changed	ssh-keygen -R host, then reconnect
Connection timed out	Check firewall, port, and host reachability
Too many auth failures	Use -i to specify key or IdentitiesOnly
Broken pipe	Add ServerAliveInterval to config

File Permissions

```
chmod 700 ~/.ssh
chmod 600 ~/.ssh/id_ed25519     # private key
chmod 644 ~/.ssh/id_ed25519.pub # public key
chmod 600 ~/.ssh/authorized_keys
chmod 644 ~/.ssh/known_hosts
```

Security Best Practices

Server Hardening

PasswordAuthentication no	Disable password login
PermitRootLogin no	Disable root SSH access
AllowUsers deploy	Whitelist allowed users
Port 2222	Non-default port (avoid scanners)
MaxAuthTries 3	Limit auth attempts

SSH Quick Reference

Key Practices

Prefer Ed25519 keys (smaller, faster, more secure).
Always set a passphrase on private keys.
Use ssh-agent to avoid retyping passphrases.
Rotate keys periodically; revoke unused keys.
Use IdentitiesOnly to control which key is offered.

Multiplexing

Connection Sharing

```
# In ~/.ssh/config
Host *
  ControlMaster auto
  ControlPath ~/.ssh/sockets/%r@%h-%p
  ControlPersist 600

# Create socket directory
mkdir -p ~/.ssh/sockets
```

Multiplexing Benefits

Reuses a single TCP connection for multiple SSH sessions to the same host. Eliminates repeated handshakes — faster connects and lower overhead. ControlPersist keeps the master alive (seconds).

Escape Sequences

SSH Escape Commands

~.	Terminate connection (kill stuck session)
~^Z	Suspend SSH session
~C	Open command line (add forwarding)
~#	List forwarded connections
~&	Background SSH (waiting for connections)
~?	Show escape help
~~	Send literal tilde