SSH Usage Guide

Table of Contents

1. Basic SSH Commands
2. SSH Key Management
3. SSH Configuration
4. Port Forwarding
5. File Transfer
6. Security Best Practices
7. Troubleshooting
8. Advanced Usage

Basic SSH Commands

Connecting to a Remote Server

```bash
# Basic connection
ssh username@hostname

# Connect to specific port
ssh -p 2222 username@hostname

# Connect with verbose output (for debugging)
ssh -v username@hostname

# Execute a command remotely
ssh username@hostname 'ls -la'

# Connect and run interactive command
ssh -t username@hostname 'sudo tail -f /var/log/messages'
```

Connection Options

```bash
```

```
# Disable password authentication
ssh -o PasswordAuthentication=no username@hostname

# Disable host key checking (use cautiously)
ssh -o StrictHostKeyChecking=no username@hostname

# Connect with compression
ssh -C username@hostname

# Force IPv4 or IPv6
ssh -4 username@hostname  # IPv4
ssh -6 username@hostname  # IPv6
```

SSH Key Management

Generating SSH Keys

bash

```
# Generate RSA key (4096-bit recommended)
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"

# Generate Ed25519 key (modern, recommended)
ssh-keygen -t ed25519 -C "your_email@example.com"

# Generate key with custom filename
ssh-keygen -t ed25519 -f ~/.ssh/custom_key_name

# Generate key without passphrase (automated systems)
ssh-keygen -t ed25519 -f ~/.ssh/key_name -N ""
```

Managing SSH Keys

bash

```bash
# Copy public key to remote server
ssh-copy-id username@hostname

# Copy specific key
ssh-copy-id -i ~/.ssh/custom_key.pub username@hostname

# Manually add key to authorized_keys
cat ~/.ssh/id_rsa.pub | ssh username@hostname 'mkdir -p ~/.ssh && cat >> ~/.ssh/authorized_keys'

# List loaded keys in ssh-agent
ssh-add -l

# Add key to ssh-agent
ssh-add ~/.ssh/private_key

# Remove key from ssh-agent
ssh-add -d ~/.ssh/private_key

# Remove all keys from ssh-agent
ssh-add -D
```

SSH Agent

```bash
# Start ssh-agent
eval "$(ssh-agent -s)"

# Add key to agent
ssh-add ~/.ssh/id_rsa

# Add key with timeout (1 hour)
ssh-add -t 3600 ~/.ssh/id_rsa

# Forward agent connection
ssh -A username@hostname
```

SSH Configuration

Client Configuration (~/.ssh/config)

```bash
```

```
# Basic host configuration
Host myserver
    HostName 192.168.1.100
    User myusername
    Port 22
    IdentityFile ~/.ssh/myserver_key

# Multiple hosts with shared settings
Host web1 web2 web3
    User admin
    Port 2222
    IdentityFile ~/.ssh/webserver_key

# Wildcard matching
Host *.example.com
    User admin
    IdentityFile ~/.ssh/example_key

# Jump host/bastion configuration
Host private-server
    HostName 10.0.0.5
    User admin
    ProxyJump bastion-host

# Full example with common options
Host production
    HostName prod.example.com
    User deploy
    Port 22
    IdentityFile ~/.ssh/prod_key
    ForwardAgent yes
    Compression yes
    ServerAliveInterval 60
    ServerAliveCountMax 3
    StrictHostKeyChecking yes
    UserKnownHostsFile ~/.ssh/known_hosts
```

Server Configuration (/etc/ssh/sshd_config)

bash

```
# Change default port
Port 2222

# Disable root login
PermitRootLogin no

# Disable password authentication
PasswordAuthentication no
PubkeyAuthentication yes

# Limit users
AllowUsers user1 user2
DenyUsers baduser

# Connection limits
MaxAuthTries 3
MaxSessions 10
ClientAliveInterval 300
ClientAliveCountMax 2

# Disable empty passwords
PermitEmptyPasswords no

# Disable X11 forwarding if not needed
X11Forwarding no

# Use protocol 2 only
Protocol 2
```

Port Forwarding

Local Port Forwarding

bash

```
# Forward local port 8080 to remote port 80
ssh -L 8080:localhost:80 username@hostname

# Forward to different host through SSH server
ssh -L 8080:database.internal:3306 username@gateway

# Multiple port forwards
ssh -L 8080:web:80 -L 3306:db:3306 username@hostname

# Background process
ssh -fN -L 8080:localhost:80 username@hostname
```

Remote Port Forwarding

bash

```
# Forward remote port 8080 to local port 80
ssh -R 8080:localhost:80 username@hostname

# Allow remote connections to forwarded port
ssh -R 0.0.0.0:8080:localhost:80 username@hostname

# Background process
ssh -fN -R 8080:localhost:80 username@hostname
```

Dynamic Port Forwarding (SOCKS Proxy)

bash

```
# Create SOCKS proxy on port 8080
ssh -D 8080 username@hostname

# Background SOCKS proxy
ssh -fN -D 8080 username@hostname
```

File Transfer

SCP (Secure Copy)

bash

```bash
# Copy file to remote server
scp file.txt username@hostname:/path/to/destination/

# Copy file from remote server
scp username@hostname:/path/to/file.txt ./

# Copy directory recursively
scp -r directory/ username@hostname:/path/to/destination/

# Copy with specific SSH key
scp -i ~/.ssh/custom_key file.txt username@hostname:/path/

# Copy through jump host
scp -J bastion-host file.txt username@target-host:/path/

# Preserve timestamps and permissions
scp -p file.txt username@hostname:/path/

# Copy with compression
scp -C large_file.tar.gz username@hostname:/path/
```

SFTP (SSH File Transfer Protocol)

```bash
# Connect to SFTP server
sftp username@hostname

# SFTP commands (once connected)
ls              # List remote directory
lls             # List local directory
pwd             # Show remote working directory
lpwd            # Show local working directory
cd remote_dir   # Change remote directory
lcd local_dir   # Change local directory
get remote_file # Download file
put local_file  # Upload file
mkdir new_dir   # Create remote directory
rmdir dir_name  # Remove remote directory
rm file_name    # Delete remote file
chmod 755 file  # Change remote file permissions
exit            # Close connection
```

rsync over SSH

```bash
# Sync directories
rsync -avz local_dir/ username@hostname:/remote/dir/

# Sync with progress
rsync -avz --progress local_dir/ username@hostname:/remote/dir/

# Dry run (preview changes)
rsync -avzn local_dir/ username@hostname:/remote/dir/

# Exclude files
rsync -avz --exclude='*.log' local_dir/ username@hostname:/remote/dir/

# Delete files on destination that don't exist locally
rsync -avz --delete local_dir/ username@hostname:/remote/dir/
```

Security Best Practices

Key Security

- Use Ed25519 or RSA 4096-bit keys
- Protect private keys with strong passphrases
- Regularly rotate SSH keys
- Use different keys for different purposes
- Store keys securely (encrypted storage)

Server Hardening

```bash
```

```
# Disable root login
PermitRootLogin no

# Use key-based authentication only
PasswordAuthentication no
PubkeyAuthentication yes

# Change default port
Port 2222

# Limit login attempts
MaxAuthTries 3

# Use fail2ban for intrusion prevention
sudo apt install fail2ban

# Configure firewall (UFW example)
sudo ufw allow 2222/tcp
sudo ufw enable
```

Connection Security

bash

```
# Verify host key fingerprint
ssh-keygen -l -f /etc/ssh/ssh_host_rsa_key.pub

# Always verify host keys on first connection
# Compare with known good fingerprint

# Use known_hosts for host verification
StrictHostKeyChecking yes
```

Troubleshooting

Common Issues and Solutions

Connection Refused

bash

```
# Check if SSH service is running
sudo systemctl status ssh

# Check listening ports
sudo netstat -tlnp | grep :22

# Check firewall rules
sudo ufw status
```

Permission Denied

bash

```
# Check SSH key permissions
chmod 600 ~/.ssh/id_rsa
chmod 644 ~/.ssh/id_rsa.pub
chmod 700 ~/.ssh
chmod 600 ~/.ssh/authorized_keys

# Check server-side permissions
# authorized_keys: 600
# ~/.ssh directory: 700
# Home directory: 755 or more restrictive
```

Debug Connection Issues

bash

```
# Verbose SSH output
ssh -vvv username@hostname

# Check SSH logs on server
sudo tail -f /var/log/auth.log

# Test specific key
ssh -i ~/.ssh/specific_key username@hostname
```

Performance Issues

bash

```
# Disable DNS lookup on server
UseDNS no

# Enable compression
ssh -C username@hostname

# Use faster cipher (less secure)
ssh -c aes128-ctr username@hostname
```

Advanced Usage

SSH Tunneling

bash

```
# Create persistent tunnel
autossh -M 20000 -f -N -L 8080:localhost:80 username@hostname

# VPN-like tunnel with tun/tap
ssh -w 0:0 username@hostname
```

SSH Multiplexing

bash

```
# Configuration for connection sharing
Host *
  ControlMaster auto
  ControlPath ~/.ssh/connections/%r@%h:%p
  ControlPersist 10m

# Create master connection
ssh -M username@hostname

# Use existing connection
ssh username@hostname
```

SSH Escape Sequences

```
# While connected, press Enter then:
~.   # Disconnect
~^Z  # Suspend SSH
```

```
~#   # List forwarded connections
~~   # Send literal ~
~?   # Help
```

ProxyCommand Usage

```bash
# Connect through HTTP proxy
ssh -o ProxyCommand='nc -X connect -x proxy:8080 %h %p' username@hostname

# Chain SSH connections
ssh -o ProxyCommand='ssh gateway nc %h %p' username@internal-host
```

SSH with Docker

```bash
# SSH into Docker container
docker exec -it container_name /bin/bash

# SSH tunnel to containerized service
ssh -L 5432:localhost:5432 docker-host
```

Quick Reference Commands

```bash
```

```
# Generate key
ssh-keygen -t ed25519

# Copy key to server
ssh-copy-id user@host

# Connect with key
ssh -i ~/.ssh/key user@host

# Port forward
ssh -L 8080:localhost:80 user@host

# Copy files
scp file.txt user@host:/path/

# SOCKS proxy
ssh -D 8080 user@host

# Execute remote command
ssh user@host 'command'

# Mount remote filesystem
sshfs user@host:/path /local/mount
```

This guide covers the most common SSH usage patterns. For additional information, consult the SSH manual pages: `man ssh`, `man ssh-keygen`, `man scp`, and `man sftp`.