

Basic SSH Usage

Connecting to a Remote Server

<code>ssh user@host</code>	Connects to the specified host as the given user. <b>Example:</b> <code>ssh john.doe@example.com</code>
<code>ssh -p port user@host</code>	Connects to the host on a specific port. <b>Example:</b> <code>ssh -p 2222 john.doe@example.com</code>
<code>ssh -i private_key user@host</code>	Connects using a specific private key file. <b>Example:</b> <code>ssh -i ~/.ssh/id_rsa john.doe@example.com</code>
<code>ssh -v user@host</code>	Verbose mode, useful for debugging connection issues.
<code>ssh -T user@host command</code>	Execute a single command on the remote host without opening a shell. <b>Example:</b> <code>ssh -T john.doe@example.com uptime</code>
<code>ssh -q user@host</code>	Quiet mode, suppresses most warning and diagnostic messages.

SSH Configuration File (~/.ssh/config)

The <code>~/.ssh/config</code> file allows you to define settings for SSH connections.
<b>Example:</b> <pre>Host example   HostName example.com   User john.doe   Port 2222   IdentityFile ~/.ssh/id_rsa</pre>
Now you can simply use <code>ssh example</code> to connect.

Common SSH Options

<code>HostName</code>	The actual hostname or IP address of the server.
<code>User</code>	The username to use for the connection.
<code>Port</code>	The port number to connect to (default is 22).
<code>IdentityFile</code>	Specifies the private key file for authentication.
<code>StrictHostKeyChecking</code>	Controls how SSH handles unknown host keys ( <code>yes</code> , <code>no</code> , <code>ask</code> ).
<code>ProxyCommand</code>	Command to use to connect to the server.

Key Management

Generating SSH Keys

<code>ssh-keygen</code>	Generates a new SSH key pair (private and public key).  <b>Example:</b> <code>ssh-keygen -t rsa -b 4096 -C "your_email@example.com"</code>
<code>ssh-keygen -t ed25519</code>	Generates a new Ed25519 SSH key pair (private and public key).  <b>Example:</b> <code>ssh-keygen -t ed25519 -C "your_email@example.com"</code>
<code>ssh-keygen -t rsa -b 4096</code>	Generates a new RSA SSH key pair with 4096 bits.
<code>ssh-keygen -f keyfile</code>	Creates a key without prompting.

Copying Keys to Remote Servers

<code>ssh-copy-id user@host</code>	Copies your public key to the remote server's <code>~/.ssh/authorized_keys</code> file.  <b>Example:</b> <code>ssh-copy-id john.doe@example.com</code>
<code>cat ~/.ssh/id_rsa.pub   ssh user@host 'mkdir -p ~/.ssh &amp;&amp; cat &gt;&gt; ~/.ssh/authorized_keys'</code>	Alternative method to copy the public key manually.
<code>pbcopy &lt; ~/.ssh/id_rsa.pub</code>	Copy the public key to clipboard.

Key Security

Always protect your private key. Ensure it has appropriate permissions (e.g., <code>chmod 600 ~/.ssh/id_rsa</code> ). Never share your private key with anyone.
Use a strong passphrase when generating your SSH key. This adds an extra layer of security.

## Port Forwarding

### Local Port Forwarding

<code>ssh -L local_port :host:remote_port user@ssh_server</code>	Forwards traffic from <code>local_port</code> on your machine to <code>remote_port</code> on <code>host</code> as seen from <code>ssh_server</code> .  <b>Example:</b> <code>ssh -L 8080:localhost:80 john.doe@example.com</code> (Access the web server on <code>example.com</code> via <code>localhost:8080</code> on your machine).
<code>ssh -L 8080:192.168.1.10:80 john.doe@example.com</code>	Access the web server on <code>192.168.1.10</code> on your machine.

### Remote Port Forwarding

<code>ssh -R remote_port :host:local_port user@ssh_server</code>	Forwards traffic from <code>remote_port</code> on <code>ssh_server</code> to <code>local_port</code> on <code>host</code> as seen from your machine.  <b>Example:</b> <code>ssh -R 9000:localhost:3000 john.doe@example.com</code> (Someone connecting to <code>example.com:9000</code> will be forwarded to your machine's port 3000).
------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### Dynamic Port Forwarding (SOCKS Proxy)

<code>ssh -D local_port user@ssh_server</code>	Creates a SOCKS proxy on <code>local_port</code> on your machine, routing all traffic through <code>ssh_server</code> .  <b>Example:</b> <code>ssh -D 1080 john.doe@example.com</code> (Configure your browser to use <code>localhost:1080</code> as a SOCKS proxy).
<code>ssh -N -D 1080 user@ssh_server</code>	Background the process and don't execute a remote command.

### Common Options

- `-N`: Do not execute a remote command. Useful for port forwarding only.
- `-f`: Requests ssh to go to background after authentication.

## Advanced SSH Usage

### Executing Commands Remotely

<code>ssh user@host 'command'</code>	Executes a single command on the remote host.  <b>Example:</b> <code>ssh john.doe@example.com 'df -h'</code> (Shows disk space usage on the remote server).
<code>ssh user@host &lt;&lt; EOF command1 command2 EOF</code>	Executes multiple commands using a 'here document'.  <b>Example:</b> <pre>ssh john.doe@example.com &lt;&lt; EOF mkdir test_dir cd test_dir pwd EOF</pre>
<code>ssh user@host bash -s &lt; script.sh</code>	Execute a shell script.  <b>Example:</b> <code>ssh john.doe@example.com bash -s &lt; script.sh</code>

### SCP (Secure Copy)

<code>scp file user@host:destination</code>	Copies a file to a remote host.  <b>Example:</b> <code>scp myfile.txt john.doe@example.com:/home/john.doe/</code>
<code>scp user@host:file destination</code>	Copies a file from a remote host.  <b>Example:</b> <code>scp john.doe@example.com:/home/john.doe/myfile.txt .</code>
<code>scp -r directory user@host:destination</code>	Copies a directory recursively to a remote host.  <b>Example:</b> <code>scp -r mydirectory john.doe@example.com:/home/john.doe/</code>
<code>scp -P port user@host:file destination</code>	Copies a file from a remote host on a specific port.  <b>Example:</b> <code>scp -P 2222 john.doe@example.com:/home/john.doe/myfile.txt .</code>

### SSH Agent Forwarding

<code>ssh -A user@host</code>	Enables agent forwarding, allowing you to use your local SSH keys on the remote server for further connections. Use with caution, as it can pose a security risk.  <b>Note:</b> Ensure <code>ForwardAgent yes</code> is in your <code>~/.ssh/config</code> or the server's <code>/etc/ssh/ssh_config</code> .
<code>ssh -o ForwardAgent=yes user@host</code>	Enables agent forwarding, allowing you to use your local SSH keys on the remote server for further connections. Use with caution, as it can pose a security risk.

### Mosh (Mobile Shell)

Mosh is a mobile shell that provides a more robust and responsive connection, especially over unreliable networks. It tolerates intermittent connectivity and IP address changes.
<b>Basic Usage:</b> <ol style="list-style-type: none"><li>Install mosh on both your local machine and the remote server.</li><li><code>mosh user@host</code></li></ol>