# SCP Cheatsheet

A comprehensive guide to using Secure Copy (SCP) for transferring files between systems, covering essential options, usage examples, and security considerations.

## SCP Basics

### SCP Overview

SCP (Secure Copy) is a command-line utility that allows you to securely transfer files and directories between two locations. It uses SSH for data transfer, providing confidentiality and integrity of the data.

SCP is commonly used for:

- Securely transferring files between local and remote systems.
- Copying files between two remote servers.
- Backing up files to a remote server.
- Moving files from one server to another during migrations.

### Basic Syntax

The basic syntax for SCP is:

```
scp [options] source destination
```

Where:

- `options` are command-line flags that modify the behavior of SCP.
- `source` is the file or directory to be copied.
- `destination` is the location where the file or directory will be copied.

### Source and Destination Paths

| | |
|---|---|
| Local to Remote | `scp /local/path/file user@remote_host:/remote/path` |
| Remote to Local | `scp user@remote_host:/remote/path/file /local/path` |
| Remote to Remote | `scp user1@remote_host1:/path/file user2@remote_host2:/path` |

## Common SCP Options

### Essential Options

| | |
|---|---|
| `-r` | Recursively copy entire directories. |
| `-v` | Verbose mode; displays debugging messages. |
| `-C` | Enable compression during transfer. |
| `-p` | Preserves modification times, access times, and modes from the original file. |
| `-q` | Quiet mode; suppresses warning and diagnostic messages. |

### Advanced Options

| | |
|---|---|
| `-P port` | Specifies the port to connect to on the remote host. Default is 22. |
| `-l limit` | Limits the bandwidth used by SCP, specified in Kbit/s. |
| `-i identity_file` | Selects the file from which the identity (private key) for public key authentication is read. |

## SCP Examples

### Basic File Transfers

Copy a local file to a remote directory:

```
scp /path/to/local/file
user@remote_host:/path/to/remote/directory
```

Copy a remote file to a local directory:

```
scp user@remote_host:/path/to/remote/file
/path/to/local/directory
```

### Advanced Examples

Copy a directory recursively to a remote system:

```
scp -r /path/to/local/directory
user@remote_host:/path/to/remote/directory
```

Copy a file with compression:

```
scp -C /path/to/local/file
user@remote_host:/path/to/remote/directory
```

Limit bandwidth during file transfer:

```
scp -l 200 /path/to/local/file
user@remote_host:/path/to/remote/directory
```

Copying a file using a specific identity file:

```
scp -i ~/.ssh/id_rsa /path/to/local/file
user@remote_host:/path/to/remote/directory
```

## Security Considerations

### Secure Authentication

Always use strong passwords or SSH keys for authentication.

- **SSH Keys:** SSH keys provide a more secure way to authenticate compared to passwords. Generate a key pair using `ssh-keygen` and copy the public key to the remote server using `ssh-copy-id`.
- **Password Policies:** If using passwords, enforce strong password policies and regularly change passwords.

### File Permissions

Ensure proper file permissions are set on both the source and destination systems.

- **Source Permissions:** Only allow authorized users to read the source files.
- **Destination Permissions:** Set appropriate permissions on the destination directory to prevent unauthorized access or modification.

### Network Security

Secure the network connection between the local and remote systems.

- **Firewalls:** Configure firewalls to allow SSH traffic only from trusted networks.
- **VPNs:** Use VPNs for secure file transfer over untrusted networks.
- **SSH Configuration:** Review SSH server configuration (`/etc/ssh/sshd_config`) to disable insecure options and enforce security best practices.