tufantunc / **ssh-mcp** Public

Code | Issues 11 | Pull requests 5 | Actions | Projects | Security and quality | Insights

main | 1 Branch | 17 Tags | Go to file | Go to file | Code

**tufantunc** Feature/command description (#26) 463d25f · 3 months ago

| | | |
|---|---|---|
| .github | feat: improve sudo implementation and add… | 5 months ago |
| src | Feature/command description (#26) | 3 months ago |
| test | Feature/command description (#26) | 3 months ago |
| .gitattributes | init project | last year |
| .gitignore | init project | last year |
| CODE_OF_CONDUCT.md | Refactor Code of Conduct for clarity and co… | 9 months ago |
| CONTRIBUTING.md | init project | last year |
| LICENSE | Initial commit | last year |
| README.md | Feature/command description (#26) | 3 months ago |
| docker-compose.yml | feat: improve sudo implementation and add… | 5 months ago |
| opencode.jsonc | Feature/command description (#26) | 3 months ago |
| package-lock.json | feat: improve sudo implementation and add… | 5 months ago |
| package.json | Feature/command description (#26) | 3 months ago |
| tsconfig.json | init project | last year |

README | Code of conduct | Contributing | MIT license

# SSH MCP Server

npm v1.5.0 | downloads 4.9k/month | node >=18 | license MIT | Stars 411 | Forks 64 | Publish to npm passing | issues 11 open

Archestra Score Good

**SSH MCP Server** is a local Model Context Protocol (MCP) server that exposes SSH control for Linux and Windows systems, enabling LLMs and other MCP clients to execute shell commands securely via SSH.

## Contents

## Quick Start

- Install SSH MCP Server
- Configure SSH MCP Server
- Set up your MCP Client (e.g. Claude Desktop, Cursor, etc)
- Execute remote shell commands on your Linux or Windows server via natural language

## Features

- MCP-compliant server exposing SSH capabilities
- Execute shell commands on remote Linux and Windows systems
- Secure authentication via password or SSH key
- Built with TypeScript and the official MCP SDK
- **Configurable timeout protection** with automatic process abortion
- **Graceful timeout handling** - attempts to kill hanging processes before closing connections

### Tools

- `exec`: Execute a shell command on the remote server
  - **Parameters:**
    - `command` (required): Shell command to execute on the remote SSH server
    - `description` (optional): Optional description of what this command will do (appended as a comment)
  - **Timeout Configuration:**
- `sudo-exec`: Execute a shell command with sudo elevation
  - **Parameters:**
    - `command` (required): Shell command to execute as root using sudo
    - `description` (optional): Optional description of what this command will do (appended as a comment)
  - **Notes:**
    - Requires `--sudoPassword` to be set for password-protected sudo
    - Can be disabled by passing the `--disableSudo` flag at startup if sudo access is not needed or not available
    - For persistent root access, consider using `--suPassword` instead which establishes a root shell
    - Tool will not be available at all if server is started with `--disableSudo`
  - **Timeout Configuration:**
    - Timeout is configured via command line argument `--timeout` (in milliseconds)
    - Default timeout: 60000ms (1 minute)
    - When a command times out, the server automatically attempts to abort the running process before closing the connection
  - **Max Command Length Configuration:**
    - Max command characters are configured via `--maxChars`
    - Default: `1000`
    - No-limit mode: set `--maxChars=none` or any `<= 0` value (e.g. `--maxChars=0`)

## Installation

1. **Clone the repository:**

```
git clone https://github.com/tufantunc/ssh-mcp.git
cd ssh-mcp
```

2. **Install dependencies:**

```
npm install
```

## Client Setup

You can configure your IDE or LLM like Cursor, Windsurf, Claude Desktop to use this MCP Server.

**Required Parameters:**

- `host`: Hostname or IP of the Linux or Windows server
- `user`: SSH username

**Optional Parameters:**

- `port`: SSH port (default: 22)
- `password`: SSH password (or use `key` for key-based auth)
- `key`: Path to private SSH key
- `sudoPassword`: Password for sudo elevation (when executing commands with sudo)
- `suPassword`: Password for su elevation (when you need a persistent root shell)
- `timeout`: Command execution timeout in milliseconds (default: 60000ms = 1 minute)
- `maxChars`: Maximum allowed characters for the `command` input (default: 1000). Use `none` or `0` to disable the limit.
- `disableSudo`: Flag to disable the `sudo-exec` tool completely. Useful when sudo access is not needed or not available.

```
{
    "mcpServers": {
        "ssh-mcp": {
            "command": "npx",
            "args": [
                "ssh-mcp",
                "-y",
                "--",
                "--host=1.2.3.4",
                "--port=22",
                "--user=root",
                "--password=pass",
                "--key=path/to/key",
                "--timeout=30000",
                "--maxChars=none"
            ]
        }
    }
}
```

### Claude Code

You can add this MCP server to Claude Code using the `claude mcp add` command. This is the recommended method for Claude Code.

**Basic Installation:**

```
claude mcp add --transport stdio ssh-mcp -- npx -y ssh-mcp -- --host=YOUR_HOST --user=YOUR_USER --password=YOUR_PASSW
```

**Installation Examples:**

**With Password Authentication:**

```
claude mcp add --transport stdio ssh-mcp -- npx -y ssh-mcp -- --host=192.168.1.100 --port=22 --user=admin --password=
```

**With SSH Key Authentication:**

```
claude mcp add --transport stdio ssh-mcp -- npx -y ssh-mcp -- --host=example.com --user=root --key=/path/to/private/k
```

**With Custom Timeout and No Character Limit:**

```
claude mcp add --transport stdio ssh-mcp -- npx -y ssh-mcp -- --host=192.168.1.100 --user=admin --password=your_passw
```

**With Sudo and Su Support:**

```
claude mcp add --transport stdio ssh-mcp -- npx -y ssh-mcp -- --host=192.168.1.100 --user=admin --password=your_passw
```

**Installation Scopes:**

You can specify the scope when adding the server:

- **Local scope** (default): For personal use in the current project

```
claude mcp add --transport stdio ssh-mcp --scope local -- npx -y ssh-mcp -- --host=YOUR_HOST --user=YOUR_USER --p
```

- **Project scope**: Share with your team via `.mcp.json` file

```
claude mcp add --transport stdio ssh-mcp --scope project -- npx -y ssh-mcp -- --host=YOUR_HOST --user=YOUR_USER -
```

- **User scope**: Available across all your projects

```
claude mcp add --transport stdio ssh-mcp --scope user -- npx -y ssh-mcp -- --host=YOUR_HOST --user=YOUR_USER --pa
```

**Verify Installation:**

After adding the server, restart Claude Code and ask Cascade to execute a command:

```
"Can you run 'ls -la' on the remote server?"
```

For more information about MCP in Claude Code, see the official documentation.

## Testing

You can use the MCP Inspector for visual debugging of this MCP Server.

```
npm run inspect
```

## Disclaimer

SSH MCP Server is provided under the MIT License. Use at your own risk. This project is not affiliated with or endorsed by any SSH or MCP provider.

## Contributing

We welcome contributions! Please see our Contributing Guidelines for more information.

## Code of Conduct

This project follows a Code of Conduct to ensure a welcoming environment for everyone.

## Releases 17

**1.5.0** Latest
on Jan 3

+ 16 releases

## Sponsor this project


**tufantunc** Tufan Tunç

Sponsor

Learn more about GitHub Sponsors

## Packages

No packages published

## Contributors 3


**tufantunc** Tufan Tunç


**msexxeta** Manuel Schweigert


**Matvey-Kuk** Matvey Kukuy

## Languages

**TypeScript** 55.1% **JavaScript** 44.9%