systemd-ssh-proxy

- Description
- Exit status
- Examples
- See Also

systemd

systemd-ssh-proxy

1

systemd-ssh-proxy

SSH client plugin for connecting to

AF_VSOCK

and

AF_UNIX

sockets

```
Host unix/* unix%* vsock/* vsock%* vsock-mux/* vsock-mux%*
    ProxyCommand /usr/lib/systemd/systemd-ssh-proxy %h %p
    ProxyUseFdpass yes
```

/usr/lib/systemd/systemd-ssh-proxy

ADDRESS

PORT

Description

`systemd-ssh-proxy` is a small "proxy" plugin for the `ssh(1)` tool that allows connecting to `AF_UNIX` and `AF_VSOCK` sockets. It implements the interface defined by `ssh`'s `ProxyCommand` configuration option. It's supposed to be used with an `ssh_config(5)` configuration fragment like the following:

```
Host unix/* unix%* vsock/* vsock%* vsock-mux/* vsock-mux%*
    ProxyCommand /usr/lib/systemd/systemd-ssh-proxy %h %p
    ProxyUseFdpass yes
    CheckHostIP no

Host .host
    ProxyCommand /usr/lib/systemd/systemd-ssh-proxy unix/run/ssh-unix-local/socket %p
    ProxyUseFdpass yes
    CheckHostIP no
```

A configuration fragment along these lines is by default installed into `/etc/ssh/ssh_config.d/20-systemd-ssh-proxy.conf`.

With this in place, SSH connections to host string `unix/` followed by an absolute `AF_UNIX` file system path to a socket will be directed to the specified socket, which must be of type `SOCK_STREAM`. Similar, SSH connections to `vsock/` followed by an `AF_VSOCK` CID will result in an SSH connection made to that CID. `vsock-mux/` followed by an absolute `AF_UNIX` file system path to a socket is similar but for cloud-hypervisor/firecracker which do not allow direct `AF_VSOCK` communication between the host and guests, and provide their own multiplexer over `AF_UNIX` sockets. See cloud-hypervisor VSOCK support and Using the Firecracker Virtio-vsock Device. Note that `%` can be used as a separator instead of `/` to be compatible with tools like `scp` and `rsync`.

Moreover, connecting to `.host` will connect to the local host via SSH, without involving networking.

This tool is supposed to be used together with `systemd-ssh-generator(8)` which when run inside a VM or container will bind SSH to suitable addresses. `systemd-ssh-generator` is supposed to run in the container or VM guest, and `systemd-ssh-proxy` is run on the host, in order to connect to the container or VM guest.

Exit status

On success, 0 is returned, a non-zero failure code otherwise.

Examples

Talk to a local VM with CID 4711

```
ssh vsock/4711
```

Talk to a VM guest hosted with cloud-hypervisor/firecracker

```
ssh vsock-mux/run/vm-1234.sock
```

Talk to the local host via ssh

```
ssh .host
```

or equivalent:

```
ssh unix/run/ssh-unix-local/socket
```

Copy local 'foo' file to a local VM with CID 1348

```
scp foo vsock%1348:
```

See Also

`systemd(1)`, `systemd-ssh-generator(8)`, `vsock(7)`, `unix(7)`, `ssh(1)`, `sshd(8)`

□□□ #1
□ □□□ □□□ 2026-10-31 15:01:56 UTC
□ □□□ □□□ 2026-10-31 15:01:56 UTC