Manual

Basic use

ssh -R 80:localhost:3000 serveo.net

The -R option instructs your SSH client to request port forwarding from the server and proxy requests to the specified host and port (usually localhost). A subdomain of serveo.net will be assigned to forward HTTP traffic.

Request multiple tunnels at once

ssh -R 80:localhost:8888 **-R 80:localhost:9999** serveo.net

The target server doesn't have to be on localhost

 ssh -R 80:**example.com**:80 serveo.net

Request a particular subdomain

The subdomain is chosen deterministically based on your IP address, the provided SSH username, and subdomain availability, so you'll often get the same subdomain between restarts. You can also request a particular subdomain:

ssh -R **incubo:**80:localhost:8888 serveo.net  
ssh -R **incubo.serveo.net:**80:localhost:8888 serveo.net

Change the SSH username to get assigned a different subdomain:

ssh -R 80:localhost:8888 **foo@**serveo.net  
ssh -R 80:localhost:8888 **-l foo** serveo.net

SSH forwarding

Request forwarding for port 22, and serveo.net will act as a jump host, allowing you to conveniently SSH into your machine (a unique alias is required):

ssh -R **myalias:22**:localhost:22 serveo.net

Then you can establish an SSH connection using serveo.net as an intermediary like this:

ssh -J serveo.net user@myalias

The -J option was introduced in the OpenSSH client version 7.3. If you have an older client, you can use the ProxyCommand option instead:

ssh **-o ProxyCommand="ssh -W myalias:22 serveo.net"** user@myalias

Generic TCP forwarding

If you request a port other than 80, 443, or 22, raw TCP traffic will be forwarded. (In this case, there's no way to route connections based on hostname, and the host, if specifed, will be ignored.)

ssh -R **1492**:localhost:1492 serveo.net

If port 0 is requested, a random TCP port will be forwarded:

ssh -R **0**:localhost:1492 serveo.net

Connect on port 443

In some environments, outbound port 22 connections are blocked. For this reason, you can also connect on port 443.

ssh **-p 443** -R 80:localhost:8888 serveo.net

Keep the connection alive

Use ServerAliveInterval to prevent an idle connection from timing out:

ssh **-o ServerAliveInterval=60** -R 80:localhost:8888 serveo.net

Automatically reconnect

Use [autossh](http://www.harding.motd.ca/autossh/" \t "_blank) for more persistent tunnels. Use "-M 0" to disable autossh's connectivity checking (rely on ServerAliveInterval and ServerAliveCountMax instead):

**autossh -M 0** -R 80:localhost:8888 serveo.net

See <https://www.everythingcli.org/ssh-tunnelling-for-fun-and-profit-autossh/> for more about autossh.

GUI

Pressing g will start a GUI session. Use up and down arrow keys to inspect requests/responses, and left and right arrow keys to switch between request and response views. Use j, k, u, d, p, and n to scroll the lower inspector pane. Press r to replay the selected request. Press h or ? at any time for help.

Custom Domain

To use your own domain or subdomain, you'll first need an SSH key pair. Use the **ssh-keygen** program to generate a key pair if you don't already have one.

Next, use **ssh-keygen -l** and note your key's fingerprint. Here's an example output:

2048 SHA256:pmc7ZRv7ymCmghUwHoJWEm5ToSTd33ryeDeps5RnfRY no comment (RSA)

In this example, the fingerprint is **SHA256:pmc7ZRv7ymCmghUwHoJWEm5ToSTd33ryeDeps5RnfRY**.

Now you need to add two DNS records for the domain or subdomain you'd like to use:

1. An A record pointing to **159.89.214.31**.
2. For each SSH key to allow, a TXT record in the form of authkeyfp=[fingerprint]. For the example key above, the DNS record would be **authkeyfp=SHA256:pmc7ZRv7ymCmghUwHoJWEm5ToSTd33ryeDeps5RnfRY**.

Once your DNS records are in place, you can request your subdomain/domain from Serveo:

ssh -R subdomain.example.com:80:localhost:3000 serveo.net

When you request port forwarding for subdomain.example.com, Serveo will fetch the TXT records from your DNS server and only allow forwarding if you've provided a public key with the same fingerprint as specified in TXT records.