Remote Access to your Server

For security reasons, access to your Observatory server is only possible within the Observatory intranet. Remote access can occur either securing your connection via an intermediate step called *SSH tunneling* (AKA *port forwarding*), through the Observatory VPN service or via the Observatory Remote Workspace.

Following are some examples that demonstrate the concept of SSH tunnelling. For alternative methods of connection, please see the relevant documentation. SSH access to our servers requires you to set up two-factor authentication (2FA) on your account for security reasons.

 The examples below have been tested with OpenSSH v7.3+ on Linux; for MacOS, see example 3 below (which also works on recent Linux versions).

SSH tunneling

By means of an SSH tunnel you can transport any arbitrary data over an encrypted SSH connection. Members of the Observatory can use this technique to gain remote shell access to their servers across our firewall which would prevent access otherwise.

How does it work?

You must have an ssh client installed on your personal device – e.g. laptop, PC – in order to establish a *tunnelled* connection.
The Observatory has a dedicated server (SSH server) ready to listen to any (authenticated) client connections.
Once a client-server connection is established, a given application contacts the SSH client on a chosen port on which the client is listening.
The SSH client in turns forwards all encrypted application data to the server which finally communicates with the actual application server.

For remote ssh connections to your servers, the steps above can be summarized into the following. Establish an ssh client-server to our SSH server and instruct your SSH client to forward any new SSH-connection data that will be sent to an arbitrary port number to go via our SSH server. The server will then relay this information to the SSH server running on your workstation.

Example 1

Establish an SSH connection to a machine called <server>: `SERVER.strw.leidenuniv.nl` via our SSH server `ssh.strw.leidenuniv.nl`

```
ssh -o ProxyCommand="ssh -W %h:%p username@ssh.strw.leidenuniv.nl"
username@SERVER.strw.leidenuniv.nl
```

For connections that will use the DISPLAY environment variable (think of any application with a GUI), add the option `-X` to your SSH commands.

Example 2

As in *Example 1* but this time using your client ssh configuration file usually located at `$HOME/.ssh/config` on GNU/Linux systems

```
# cat $HOME/.ssh/config
Host SERVER.strw.leidenuniv.nl SERVER
     ProxyCommand /usr/bin/ssh -W %h:%p ssh.strw.leidenuniv.nl
     User username
```

Once this configuration is in place, a simple `ssh SERVER` will get you to your workstation. Of course, substitute the name of the server you want to use.

Example 3

More recent versions of ssh (including ssh on MacOS) also have the option `ProxyJump` which has a somewhat easier syntax. When using that option, the examples become: On the commandline:

```
ssh -o ProxyJump="username@ssh.strw.leidenuniv.nl"
username@SERVER.strw.leidenuniv.nl
```

or if your ssh client has the `-J` option:

```
ssh -J username@ssh.strw.leidenuniv.nl username@SERVER.strw.leidenuniv.nl
```

And in the `.ssh/config` file:

```
# cat $HOME/.ssh/config
Host SERVER.strw.leidenuniv.nl SERVER
     Proxyjump ssh.strw.leidenuniv.nl
     User username
```