

Login via SSH and VNC

Secure Shell and Virtual Network Computing

→ Client-Server-System for a **secure** (encrypted) connection and login to a remote host.

Info box

- Requirements: name of the remote machine (host name + domain name OR IP-address) as well as user name plus corresponding password.
- During the FIRST secure login to a new remote host, the “**fingerprint**” of the host is shown and the user is queried whether to trust that specific host machine: → **Reply: yes** (Unless of course you have reasons not to . . .)

SSH with Linux and macOS

The SSH-client is already installed for Linux/Unix and macOS (also for Win10, see further down).

- 1 Linux: open terminal/console;
macOS: Applications → Utilities → Terminal
- 2 Login via Username (e.g. htodt) and Hostname (e.g. bell)

```
ssh htodt@bell.stud.physik.uni-potsdam.de -Y
```

IMPORTANT! There is always at least once space character between a command and its options. Everything is case sensitive! The common option `-Y` activates X11-forwarding (graphics and window-widgets). An alternative to providing the hostname is the direct usage of the IP address, e.g. `ssh htodt@141.89.178.71 -Y`

- 3 Verify the success of the connection via the simple command `hostname` (should now yield e.g. "bell"). The application `xeyes` opens up a graphical element (the eponymous eyes) and thereby demonstrates the successful X11-forwarding (to terminate it, press `CTRL` + `c` or `STRG` + `c`)

MacOS users may have to set up the support for X11 first: <https://www.xquartz.org>

The integrated **PowerShell** of Windows 10 behaves pretty much like Linux with regard to SSH connections. It comes with its own SSH-Client and a very similar syntax (without a X11 environment, however):

```
ssh username@weber.stud.physik.uni-potsdam.de
```

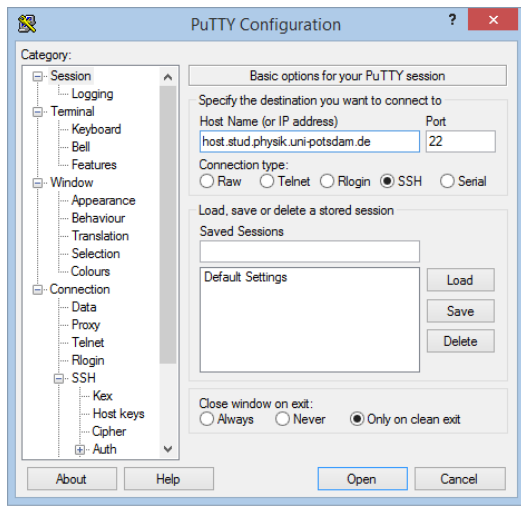
Older Windows versions (such as Window 8.1, etc) require the installation of a dedicated SSH-client, such as:

- PuTTY: <https://www.putty.org>
- MobaXterm: <http://mobaxterm.mobatek.net>
→ SSH-Client with X11-support, (recommended). Connect after launch with

```
ssh username@weber.stud.physik.uni-potsdam.de -Y
```

(use your username) to connect to a remote machine with X11-forwarding (in the given example line: weber). Password required.

PuTTY also allows for SSH-logins, no X11-forwarding for graphics, though.



- 1 Category → “Sessions”
- 2 Host Name[†] : Name or IP of a computer pool host, e.g. mahler.stud.physik.uni-potsdam.de
Port: 22
- 3 → Open
- 4 Enter Username[†] and Password

[†]The Username can also be given directly in the command line before the @ “Host Name”:
user@host.stud.physik.uni-potsdam.de

The SecureShell after Login

After a successful login you execute commands ON THE host machine. The following **commands** should work for you:

hostname

→ shows (prints) the name of the host you are currently logged into. Call this also when you are unsure where you are currently typing on (local or remote machine).

whoami

→ shows your current user name (will likely differ from your own machine to your university pool account!)

w

→ shows user names of logged in users

logout (or exit)

→ end current SSH-session (not required, but nice)

Virtual Network Computing enables us to mirror the complete (Desktop) screen of the remote machine to our local machine via (inter)net connections. The connection works in both directions, such that your keyboard and mouse inputs (NOT your private desktop screen) are transmitted back to the server.

This means that you have full control over the (graphical) user interface of the remote machine as if your screen, mouse and keyboard were connected to it.

This requires a VNC-server to run on the remote machine. That service is a dedicated software that provides the necessary interfaces to the remote machine for a external user.

First, you log in to that remote machine via SSH connection. In your remote shell (check?) execute the VNC server once(!). This is a SERVICE that will not be terminated when the (SSH) connection ends, unlike simple applications that you may have started (such as xeyes). Launch `vncserver` with:

```
htodt@mahler:~> vncserver
```

This service asks you (at first launch) to set a new password specifically to access this service, for simplicity you may use your user password here but it is independent of your user account! BEWARE: This service password is stored locally on the server with a considerably weak protection (simple hash) in `~/ .vnc/passwd`. The VNC server also asks you for the need of a View-Only-Password during initial setup. Deny that with “no”.

So you have two sets of accesses:

- Your username + password to access one of the machines at the University PC pool
- A password to access the VNC Service for the user already being logged in (see above)

Display number

The VNC-Server will create a dedicated *display number* at launch:

New 'weber:3 (htodt)' desktop is weber:3

Please note this number down, it is needed later on since it impacts the *port number* you will have to use!

Other relevant options of the `vncserver` command (remember the space before the options):

```
vncserver -list
```

→ shows running `vncserver`-processes and their *display number* of the user (other IDs may be used by other users on the same host!):

```
weber/htodt> vncserver -list
```

```
TigerVNC server sessions:
```

```
X DISPLAY # PROCESS ID
```

```
:7 11856
```

```
vncserver -kill :display
```

→ ends the `vncserver`-process for the corresponding “display” (*display number*)

```
weber/htodt> vncserver -kill :7
```

Under Linux the corresponding VNC client is installed as package, e.g., under openSUSE:

```
user:~> sudo zypper install tigervnc
```

or under Ubuntu respectively:

```
user:~> sudo apt-get install xtightvncviewer
```

Under macOS one has the the following options:

- one can install the client analogously via **MacPorts** (<https://www.macports.org>, **must be installed before**) in the shell (terminal):

```
user:~> sudo port install tigervnc
```

- one can install the **realVNC Viewer** for macOS
<https://www.realvnc.com/de/connect/download/viewer/>
- one can use the onboard VNC client called **Screen Sharing** or **Bildschirmfreigabe** (German version), which can be found with help of *Spotlight* or directly at
`/System/Library/CoreServices/Applications/Screen Sharing.app/Contents/MacOS/Screen Sharing`

Screen Sharing (Bildschirmfreigabe) and German keyboard layout

The onboard macOS-VNC-client does not support the third level of key assignments (via **ALT GR** key), therefore many relevant special characters, e.g., | (pipe) or \ (backslash) cannot be used with German keyboard layout in a *Screen Sharing* session.

The VNC session can be launched as follows:

- 1 start the `vncserver` on the remote host (see p. 8)
- 2 from the shell: open a so-called SSH tunnel to the remote host, on which `vncserver` has already been launched, here with `socket=5903` for VNC server display `:3` → `5900` + VNC display number (see p. 9):

```
ssh -L 5903:localhost:5903 username@mahler.stud.physik.uni-potsdam.de
```

Remarks:

- instead of the IP address also the *hostname* of the remote host can be used, here: `mahler.stud.physik.uni-potsdam.de`
- additional options: `-C` → compression (not useful for VNC); `-N -f` → start tunnel in background (no remote-shell)
- syntax of the `-L` option actually: *local_socket*:*IP*:*remote_port*, hence *socket number* in front of colon is arbitrary, e.g., `-L 5555:localhost:5903`, if multiple VNC connections are required to different hosts with the same display number!

- ③ the local VNC client can now access the SSH tunnel (IP of the local host: 127.0.0.1 = localhost) with the socket 5903[†] (→ VNC server display :3):

```
vncviewer localhost:5903
```

For the RealVNC Viewer and macOS *Screen Sharing* the input localhost:5903 is done in the address field for VNC server in the graphical user interface of these programs (cf. p. 20).

A window will open, which might ask for the VNC server password that was set when starting `vncserver` (see p. 8). Eventually the desktop environment of the remote host should appear (see p. 22).

[†] cf. remark regarding socket number above, on p. 13

If the connection is refused, terminate/kill the corresponding SSH process from step 2 and restart the SSH tunnel.

If an error occurs with the text that the connection to the server is not possible, there might be the problem that `vncserver` is not really running on the remote host.

- ④ Quit: just close the window of the VNC session (session on remote host will persist) or use “Logout” on the remote host (terminates session).

The option `-via`

Some versions of `vncviewer` (in terminal/shell) support the option `-via`, which opens an SSH tunnel on the fly, hence the steps 2 & 3 can be combined in a single program call:

```
vncviewer -via username@mahler.stud.physik.uni-potsdam.de mahler.stud.physik.uni-potsdam.de:1
```

where `:1` is again the display number (see p. 9).

Under Windows there are, e.g., the following VNC clients available:

- TightVNC jviewer

<https://www.tightvnc.com/download/2.8.3/tvnjviewer-2.8.3-bin-gnugpl.zip>

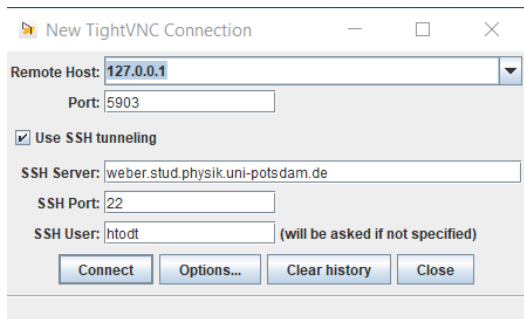
→ JAVA-Applet, **requires JAVA!**: <https://www.java.com/de/download/manual.jsp>

- RealVNC <https://www.realvnc.com/de/connect/download/viewer/>

Both require that the program `vncserver` has been already started on the remote host (see p. 8).

There are many more VNC viewers/clients, just use an internet search engine.

The TightVNC jviewer also start the necessary SSH tunnel on the fly:



New TightVNC Connection

Remote Host: 127.0.0.1

Port: 5903

Use SSH tunneling

SSH Server: weber.stud.physik.uni-potsdam.de

SSH Port: 22

SSH User: htodt (will be asked if not specified)

Connect Options... Clear history Close

Example configuration for vncserver on “weber” with Display :3 and user “htodt”

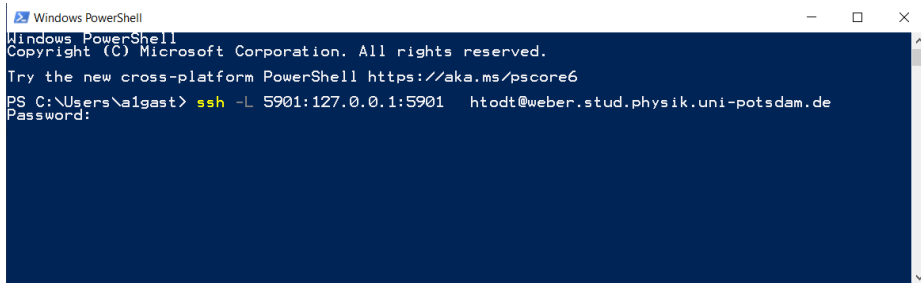
- 1 Remote Host: 127.0.0.1 (localhost)
- 2 Port = 5900 + display number (e.g., 5903, if VNC server uses Display :3)
- 3 Use SSH tunneling : ✓
- 4 SSH Server:
weber.stud.physik.uni-potsdam.de
- 5 SSH Port: 22
- 6 SSH User: username (here: htodt)
- 7 → Connect

Caution: Under Windows the TightVNC jviewer does not support the third level of key assignment via **ALT GR**, i.e. with a German keyboard layout one cannot type a backslash \ and so on.

VNC clients for Windows III

For the RealVNC Viewer one must first open an SSH tunnel.

Method I: SSH tunnel using PowerShell of Windows 10:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

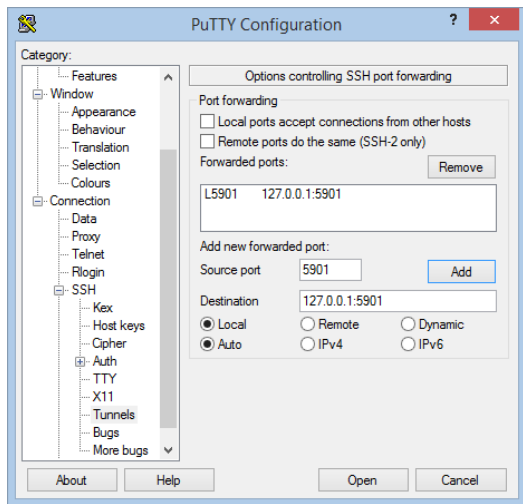
PS C:\Users\algast> ssh -L 5901:127.0.0.1:5901 htdot@weber.stud.physik.uni-potsdam.de
Password:
```

syntax very similar to Linux or macOS shell (cf. p. 13), here using the IP address and VNC server Display :1 :

```
ssh -L 5901:localhost:5901 username@mahler.stud.physik.uni-potsdam.de
```

→ ssh will ask for the password of the remote account
(if using the options `-N` `-f` the PowerShell will be blocked then)

Method II: SSH tunnel using PuTTY (e.g., also for older versions of Windows)



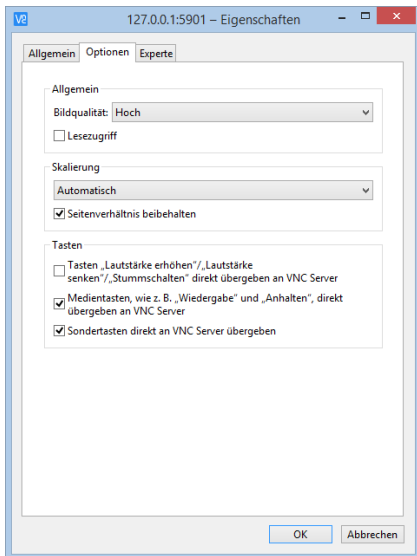
- 1 as on page 5:
Category → “Sessions”
Host Name : name or IP of a host from the stud cluster, e.g., mahler.stud.physik.uni-potsdam.de
Port: 22
- 2 Category → Connection → SSH
→ Tunnels
Source Port: 5901 (socket number, see p. 13)
Destination: 127.0.0.1:5901 (if VNC Display :1)
→ Add → Open



Input of user name and password

as soon as the SSH tunnel works, one can start, e.g., the **RealVNC Client**:

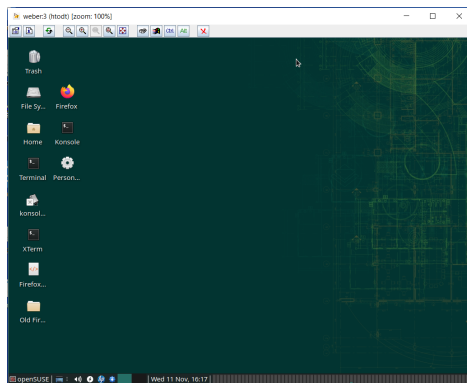


→ address: 127.0.0.1:5901, if VNC server uses Display :1,
so again → 5900 + VNC server display number (here :1), or the socket number, see
corresponding remark on p. 13



- Warnings due to a unencrypted connection can be ignored (refer to the local connection to the SSH tunnel).
- It is recommended to set the Picture quality to “high” (in menu item “Properties”). This can be also done later in the menu of the VNC window on the top  by clicking on “Properties” .
- The RealVNC Viewer is also available for many other operating systems (e.g., Linux, macOS).

Stud cluster desktop in the VNC Viewer



After successful login you can see the desktop of the Linux computer (with “openSUSE” menu button lower left) in a new window.

Heads up! Important note about closing the session

End you VNC session just by closing the window of the VNC viewer. Do **not** logout, as you might see a black screen when you login the next time (s. p. 32).

By creating an RSA key pair and depositing the public key on the remote host one can login via SSH without the password:

- 1 In a terminal (Linux, macOS) or in the PowerShell (Windows) create a key pair using:

```
ssh-keygen
```

You will be asked where to save the key pair and for a passphrase. Just confirm both default settings with (i.e. no passphrase).

- 2 Then **change** to the directory `.ssh` in your home directory (in doubt: `cd ; cd .ssh`).
- 3 From there copy your public key via `scp` to the the remote host:

```
scp id_rsa.pub user@host.stud.physik.uni-potsdam.de:~/.ssh/id_rsa.pub_laptop
```

(assuming that there is already `.ssh` in your home directory on the remote host, if not, just execute `ssh-keygen` also once on the remote host)

- ④ Login via ssh on the remote host, change there to `.ssh` (is directly in your home directory) and *append* the previously copied public key `id_rsa.pub_laptop` to the file `authorized_keys`:

```
cat id_rsa.pub_laptop >> authorized_keys
```

Then, you should be able to login to the remote host without typing your password. **Important note:**

- By using `cat ... >> authorized_keys` you can append further public keys to the file without overwriting the key(s) already in that file
- The directory `.ssh` must always have the following permissions:

```
drwx----- 1 htodt astro      1112 Dec  9 15:48 .ssh
```

i.e. not readable for other users

The file `.ssh/config` for more comfort |

The file `config` in the directory `.ssh` allows, e.g., to shorten the host name of the remote host(s):

- 1 Create the file `config` in the directory `.ssh` with help of a *text editor*. Under Windows pay attention that the editor (e.g., Notepad) *does not* append the file extension `.txt` automatically. If necessary rename the file (e.g., `mv config.txt config`).
- 2 e.g., for the host `bell` and the user `htodt` create the following entry:

```
Host bell
Hostname bell.stud.physik.uni-potsdam.de
User htodt
```

- 3 subsequently, it is sufficient only to use the short name (here: `bell`), when starting a connection/tunnel with SSH in the terminal or PowerShell:

```
ssh bell
ssh -L 5903:localhost:5903 bell
```

The file `.ssh/config` for more comfort II

Remarks:

- You can create further entries for other remote hosts, insert a blank line for better readability between the entries for each host.
- Under Linux/macOS you can switch on automatic X11 forwarding (otherwise with the option `+Y`) with the following lines:

```
ForwardX11 yes  
ForwardX11Trusted yes
```

- A frequent problem of SSH connections (at least in the past) is the interruption because of time-out. You may try to keep the connection “alive” by some kind of ping every 60s with the following lines:

```
Host *  
ServerAliveInterval 60
```

(The asterisk `*` means that the following line(s) apply to all remote hosts.)

The file `.ssh/config` for more comfort III

Example for a config file

```
Host *
ServerAliveInterval 60

Host bell
Hostname bell.stud.physik.uni-potsdam.de
User htodt
ForwardX11 yes
ForwardX11Trusted yes

Host joule
Hostname 141.89.178.77
User htodt
ForwardX11 yes
ForwardX11Trusted yes
```

Screen resolution in a running VNC session

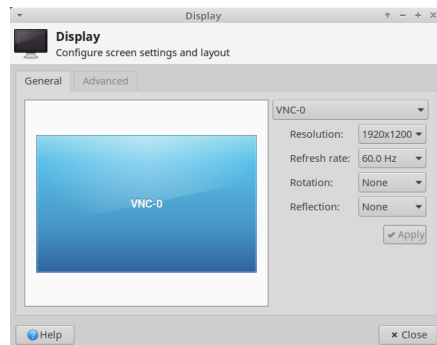
Modern desktop environments (Xfce, LXQT, ...) usually allow for changing the “screen resolution” of the running VNC session by the user.

e.g. under Xfce

→ openSUSE → Settings

→ Display

Pay attention that there is really the VNC display shown (here: VNC-0)!



→ this will also change the current size of the VNC session window

Alternatives: The tiger VNC viewer allows to change (permanently) the resolution by resizing the window of the viewer.

Also the command `xrandr` in your VNC session, e.g., `xrandr -s 1920x1080`, can set the resolution, but only to available resolutions.

If your desktop environment doesn't support setting of the VNC-display resolution, you can alternatively also set a fixed resolution when starting the *VNC server*:

```
vncserver -geometry 1920x1200
```

→ sets the initial default resolution to 1920×1200 pixel (width \times height) for the VNC session. Take into account that your own physical screen should have a sufficient resolution to display the VNC viewer window.

In case you are experiencing problems with your VNC access, please check the following:

- 1 The `vncserver` is actually running on the remote host and that you use the correct display number as well as the proper password for the VNC service
- 2 You have a proper SSH-tunnel to the remote host to the correct port number.
- 3 The VNC-Viewer on your machine uses the proper port number as well and connects to the `localhost` (127.0.0.1)

Regarding Point 1:

- you can change the password of a running VNC-server via `vncpasswd`
- you can delete all VNC-server settings by removing `~/.vnc` on the remote host. The files will be initialized at the next start of the VNC-server.
- terminate the running `vncserver` and restart it (`vncserver -kill :display`)

Regarding Point 2:

- you need to start the SSH-tunnel on your local machine

- pay attention to error messages during the SSH-tunnel initialization, e.g.: `bind [127.0.0.1]:5903: Address already in use` → use a different socket number (see p. 13)
- make sure that no other (former?) SSH-tunnel is already running on that socket (Unix: `ps aux | grep ssh`, Windows Powershell: `Get-Process`)
- if required, kill all SSH-connections and try starting the SSH-tunnel again

Regarding Point 3:

- the proper address usually is *localhost:socket number* (from SSH-tunnel), so for example `127.0.0.1:5902`. This assumes that the SSH-tunnel to the remote host is already operational and that the VNC-server is already running on the remote host
- Real VNC Server will ask for the VNC-server password; in case you use Tight VNC jviewer to also set up the SSH-tunnel on its own, it will also request a username and password (naturally)

Black screen after VNC login

If you only see a black screen in the VNC viewer, the following issued could be the reason:

- The virtual display of the VNC-server has activated a screensaver or some energy-saving mode (unnecessary). **Solution:** Click into the window of the VNC viewer and disable the screensaver or energy-saving settings of the VNC display for the future.
- You have logged out from your last session. The virtual VNC display would show now the login screen, however this process is not owned by the user and therefore the VNC viewer cannot access it. **Solution:** Kill and restart the VNC server. Do not logout next time (just close the viewer).
- The VNC server or the VNC display have crashed. **Solution:** Kill and restart the VNC server.

Links (clickable)

<https://www.xquartz.org>

<https://www.putty.org>

<http://mobaxterm.mobatek.net>

<https://www.macports.org>

<https://www.java.com/de/download/manual.jsp>

<https://www.tightvnc.com/download.php>

<https://www.realvnc.com/de/connect/download/viewer/>