

Anleitung

Weitere Online-Werkzeuge

für Lehrveranstaltungen und Konferenzen

Übersicht

Im ersten Teil dieser Anleitung ([online-werkzeuge.pdf](#)) wurden die Werkzeuge **mumble**, **VNC** und **OpenMeetings** aus Sicht der Teilnehmenden vorgestellt. Dieser zweite Teil der Anleitung stellt zusätzlich **VNC** aus Sicht Vortragender vor.

Normalerweise verwendet man VNC gezielt zwischen zwei Rechnern oder höchstens einer kleinen Anzahl von Rechnern. Um es auch für Lehrveranstaltungen einsetzen zu können, haben wir auf dem CVH-Server das Web-Interface **noVNC** installiert. Dieses ermöglicht den Zugriff auf den freigegebenen Bildschirm von außen per Web-Browser über eine URL und Eingabe eines Passwortes.

Da **noVNC** auf dem CVH-Server läuft, müssen die Daten des freigegebenen Bildschirminhalts auf sichere Weise vom eigenen Rechner zum CVH-Server übertragen werden. Hierfür gibt es eine etablierte Methode, den **SSH-Tunnel**, der wiederum durch ein spezialisiertes Werkzeug, z. B. **OpenSSH** oder **PuTTY**, realisiert wird.

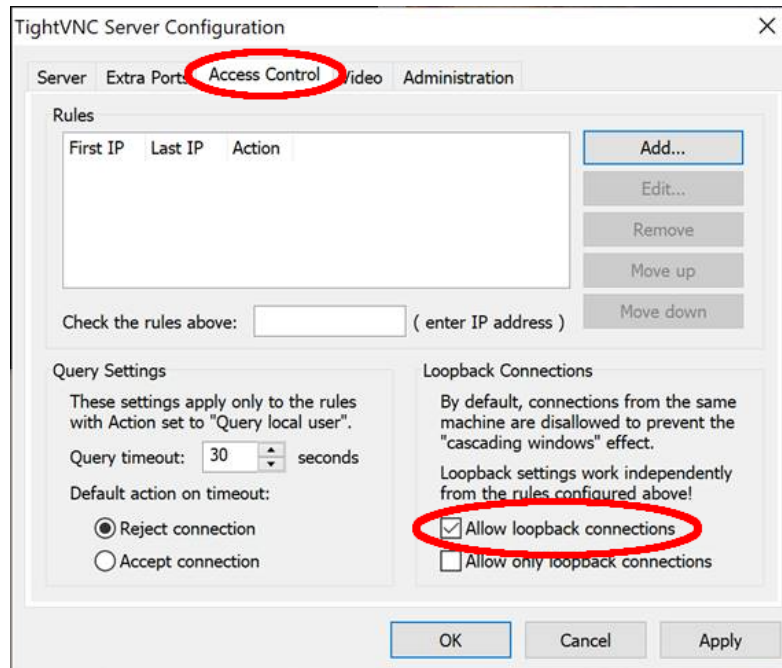
Die folgende Anleitung beschreibt detailliert, wie man mit Hilfe von VNC und einem SSH-Tunnel den eigenen Bildschirminhalt so freigeben kann, daß die Teilnehmenden der Lehrveranstaltung ihn anschließend per Web-Browser betrachten können, während man gleichzeitig z. B. per **mumble** das Gezeigte erklärt.

VNC

Es gibt verschiedene Anbieter von VNC. Diese Anleitung bezieht sich auf **TightVNC** unter MS-Windows. (Unter Unix-artigen Betriebssystemen gibt es sehr unterschiedliche Methoden, VNC zu nutzen, die wir bei Bedarf individuell erklären.)

- Laden Sie **TightVNC** von <https://www.tightvnc.com/download.html> herunter und führen Sie eine Standard-Installation durch.
- Starten Sie nun den neu installierten VNC-Server. Sie erkennen an einem „V“-Symbol in der Statusleiste, daß der Server läuft und damit grundsätzlich die Möglichkeit besteht, von außen auf Ihren Bildschirminhalt zuzugreifen.
- Klicken Sie mit der rechten Maustaste auf das „V“-Symbol und öffnen Sie den Dialog für die Konfiguration von VNC. Vergeben Sie dort zwei Passworte: ein primäres Passwort für Ihren eigenen Vollzugriff von außen (für Fernwartung, in unserem Fall ungenutzt) sowie ein „View-Only“-Passwort, mit dem eingeladene Betrachter Ihren Bildschirm von außen sehen, aber nicht verändern können.

- Gehen Sie nun auf den Reiter „Access Control“ und schalten Sie die Option „Allow loopback connections“ ein.



Damit ist VNC einsatzbereit.

Wenn Sie mehrere Monitore angeschlossen haben, gibt TightVNC diese gemeinsam frei, und diese erscheinen bei der Betrachtung stark verkleinert. Wir haben leider keine Möglichkeit gefunden, nur einen freizugeben, und können daher für den Moment nur empfehlen, den zweiten Monitor für die Dauer der VNC-Sitzung nicht zu verwenden.

Sobald tatsächlich von außen auf Ihren Bildschirminhalt zugegriffen wird, setzt der VNC-Server Ihren Bildschirmhintergrund auf eine einheitliche, dunkle Farbe. Dies dient zum einen als Signal („Achtung, Sie werden beobachtet!“) und vermindert zum anderen die benötigte Bandbreite.

Um VNC zu beenden, klicken Sie wieder mit der rechten Maustaste auf das „V“-Symbol und wählen Sie den Menüpunkt „Beenden“.

SSH-Tunnel

Anleitung für Unix-artige Betriebssysteme (einschließlich *CygWin* unter MS-Windows):

- Installieren Sie *OpenSSH*.
- Rufen Sie `ssh-keygen` auf und lassen Sie uns den Inhalt der neu erzeugten Datei `~/.ssh/id_rsa.pub` auf sichere Weise zukommen. Anhand dieser Datei können wir Ihnen sicheren Zugriff auf den CVH-Server gewähren.
- Geben Sie in einer Shell den folgenden Befehl ein:

```
ssh -N -R 59XX:localhost:59YY ssh-tunnel@main-0.cvh-server.de
```

Das **XX** steht hierbei für den von uns für Sie reservierten Kanal von **01** bis **12**, das **YY** für die Nummer Ihres VNC-Servers, typischerweise **01**.

- Bestätigen Sie die Frage nach dem Fingerprint **bei vollständiger Übereinstimmung** mit einem ausgeschriebenen „yes“. Die korrekten Fingerprints lauten:

```
1024 SHA256:/KBfmrfpI3R4U8Q+tvwUK8HuFcxwK6ej/yIkH+Uupg (DSA)
256 SHA256:kkDVI02KLySNK2cu0kHvDGXqakeIeJ3EWzF3cBL/WU8 (ECDSA)
256 SHA256:S2PJlnABOQNOKj8fnPNWBjoNhEZK/AJ+kwFzKqTtex8 (ED25519)
2048 SHA256:z2hdccP25OE2cOypCrN4V7EUv8AGUY4gnTjHK6g33pE (RSA)
```

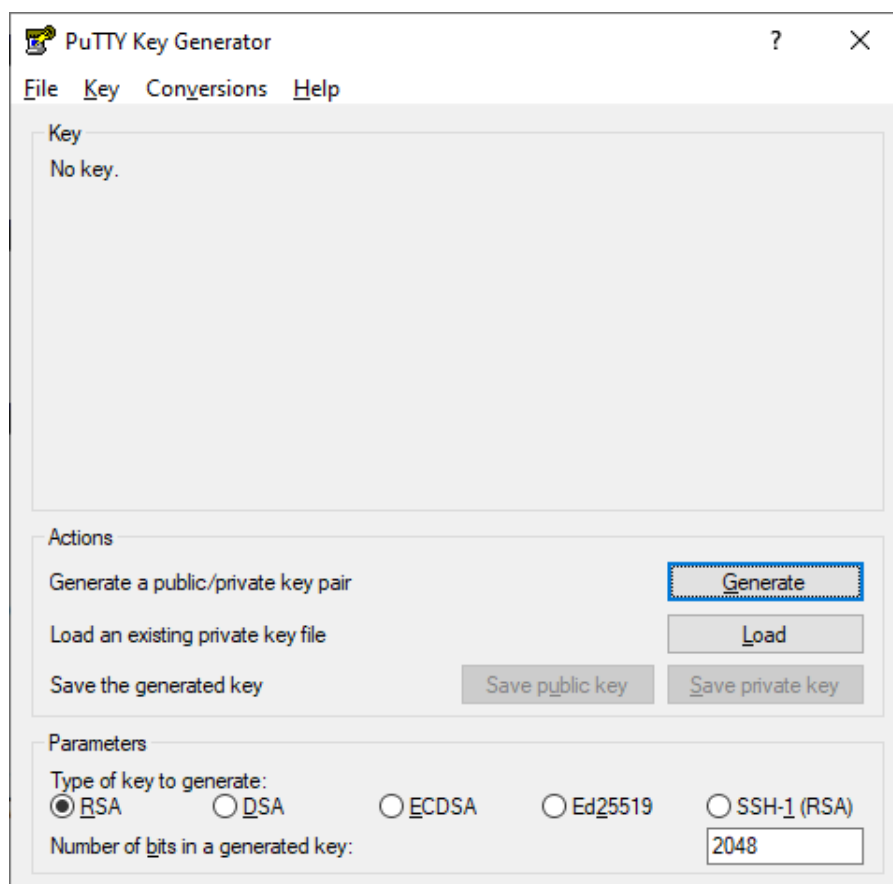
Damit ist der SSH-Tunnel aufgebaut.

- Sobald Sie den `ssh`-Befehl mit `^C` unterbrechen, ist der SSH-Tunnel wieder abgebaut.

Es folgt eine Anleitung für MS-Windows unter Verwendung von *PuTTY*.

Laden Sie *PuTTY* von <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html> herunter und führen Sie eine Standard-Installation durch.

Rufen Sie als nächstes das neu installierte Programm *PuTTYgen* auf und erzeugen Sie ein SSH-Schlüsselpaar. Die Standardeinstellungen (RSA-Schlüssel, 2048 Bits) sind in Ordnung.



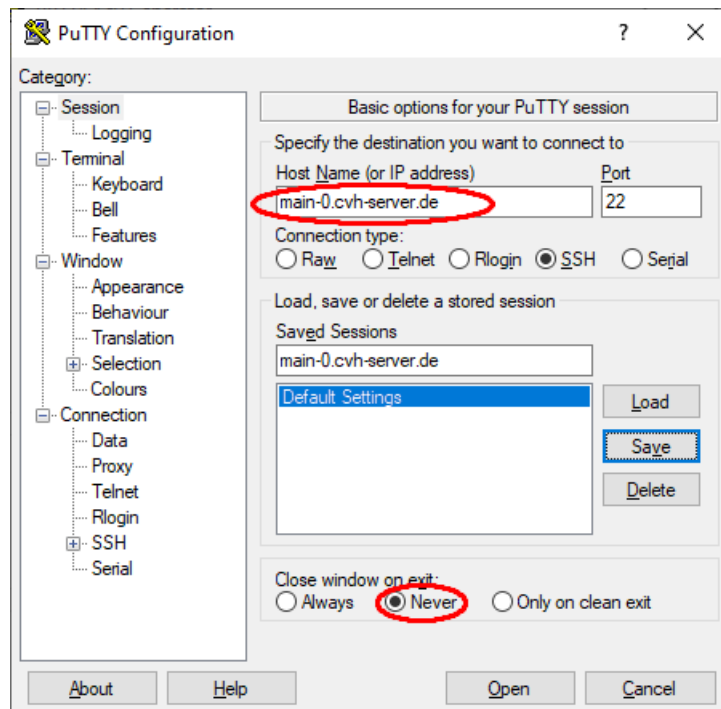
Die Schlüsselerzeugung benötigt kryptographisch sicheren Zufall. Diesen zu erzeugen, kann einige Zeit in Anspruch nehmen. Um diesen Prozeß zu unterstützen, lohnt es sich, z. B. mit der Maus Bewegungen auszuführen, aus denen die Software Zufall extrahieren kann. Die Schlüsselerzeugung wird dadurch wesentlich beschleunigt.

Speichern Sie nun die neu erzeugten Schlüssel.

Lassen Sie uns **den öffentlichen Schlüssel** („public key“) – und nur diesen! – auf sichere Weise zukommen, z. B. per E-Mail **bei gleichzeitiger Kontrolle über eine Sprechverbindung**.

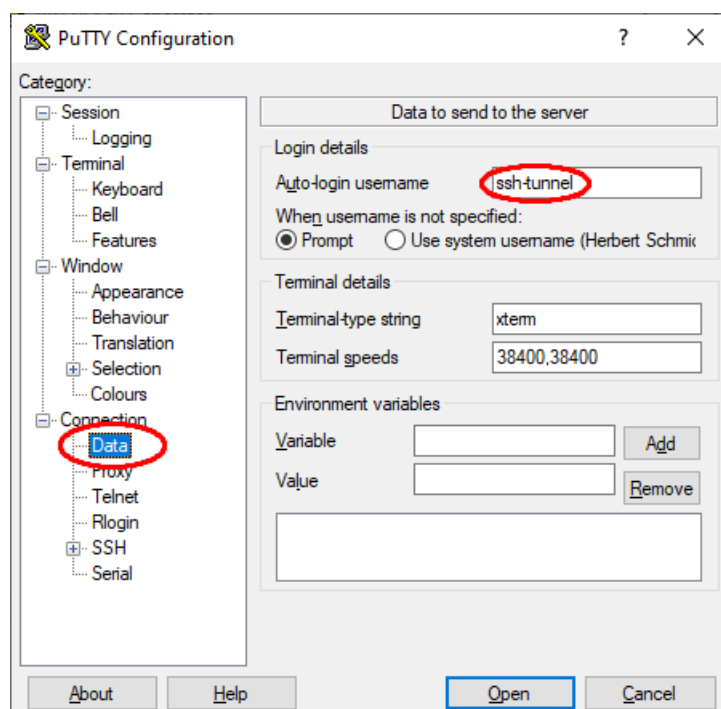
Wer im Besitz des privaten Schlüssels („private key“) ist, kann sich gegenüber dem CVH-Server als Sie ausweisen. Diese Datei sollte daher sicher aufbewahrt werden.

Der nächste Schritt besteht in der Konfiguration von **PuTTY**. Starten Sie dazu das Programm **PuTTY**.

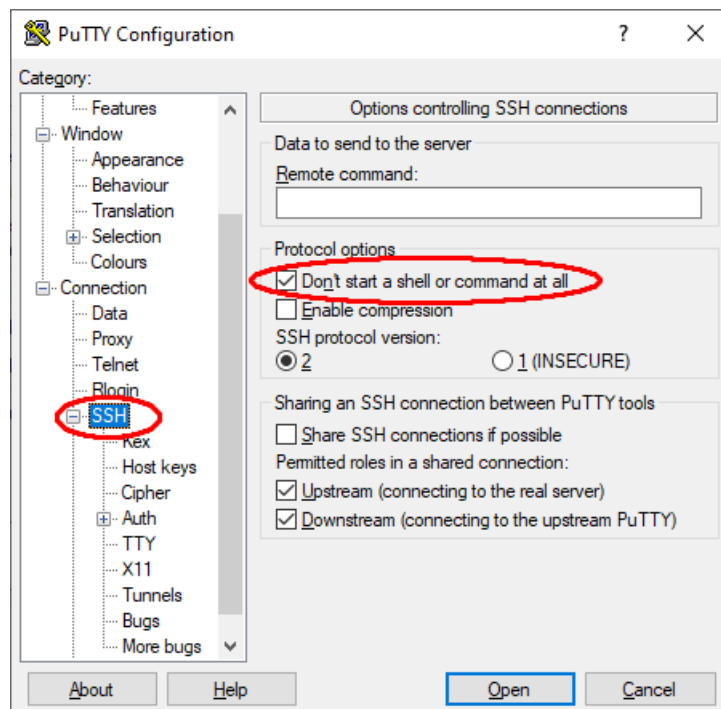


Tragen Sie unter „Host Name“ den Server-Namen **main-0.cvh-server.de** ein. (Der Standard-Port **22** ist in Ordnung.)

Setzen Sie im unteren Teil des Dialogs „Close window on exit“ auf „Never“.

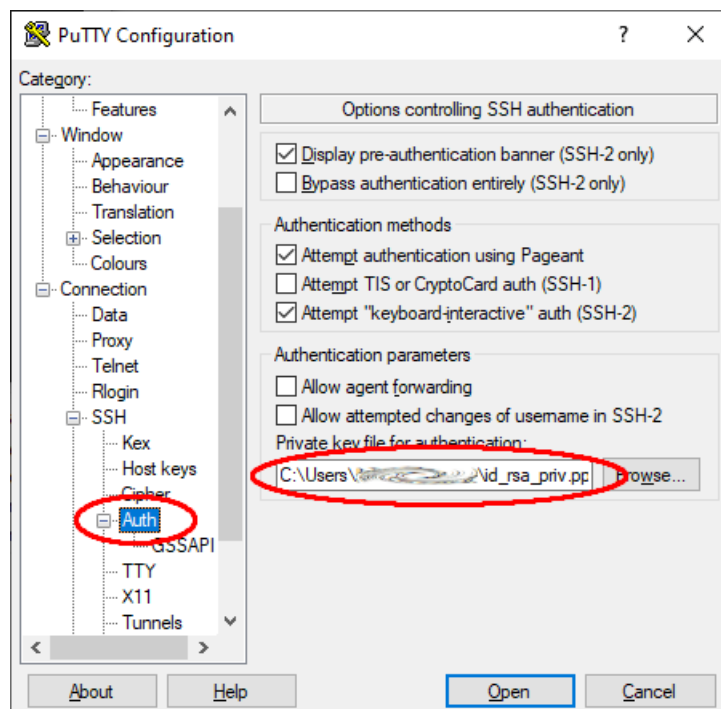


Öffnen Sie links „Connection“ und wählen Sie dort „Data“ aus. Tragen Sie anschließend unter „Auto-login username“ den Benutzernamen **ssh-tunnel** ein.



Wählen Sie nun links unter „Connection“ den Punkt „SSH“ aus und kreuzen Sie unter „Protocol options“ die Option „Don't start a shell or a command at all“ an.

(Hintergrund: Normalerweise dient SSH dazu, auf dem anderen Rechner per Kommandozeile zu arbeiten. In diesem Fall hingegen wollen wir dies gar nicht, sondern nur einen SSH-Tunnel aufbauen.)

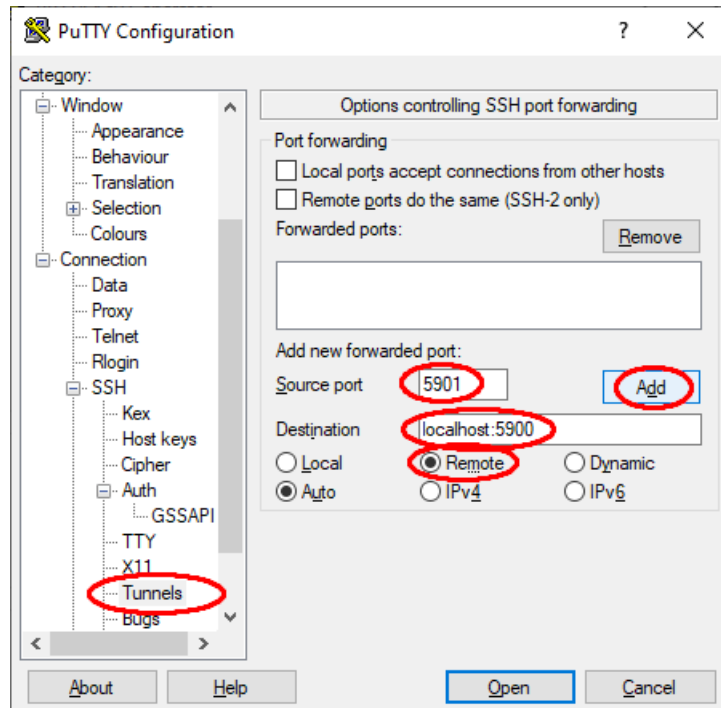


Gehen Sie nun links auf „Auth“ und tragen Sie die Datei ein, in der Sie Ihren **privaten Schlüssel** gespeichert haben.

(Hintergrund: Diese Datei ist ein Zertifikat, mit dem Sie sich gegenüber dem Server, der den zugehörigen *öffentlichen Schlüssel* kennt, authentifizieren können. Diese Art der Authentifikation ist sicherer als eine Passwort-Abfrage.)

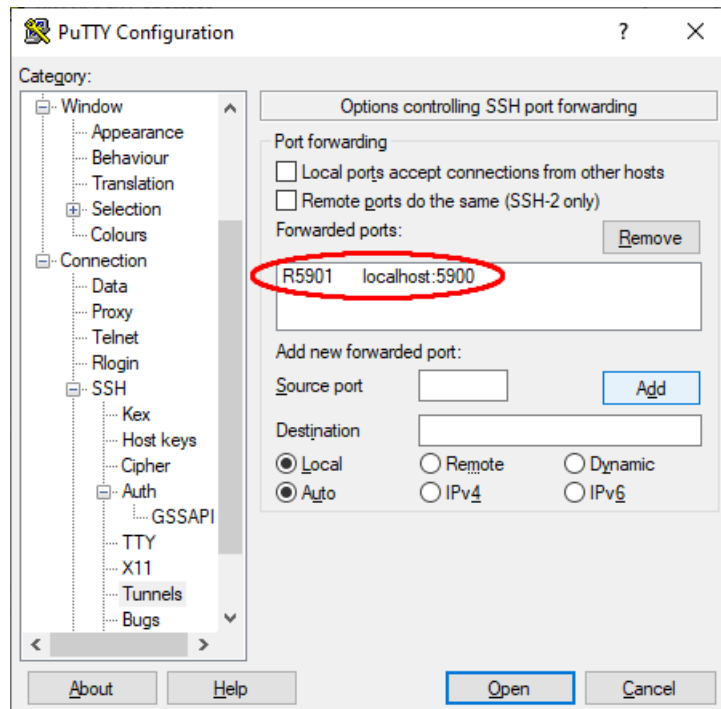
Nun konfigurieren wir den eigentlichen Tunnel.

Sie benötigen dafür einen eindeutigen **Kanal** von 1 bis 12, den wir für Sie reservieren müssen. Anhand dieses Kanals kann man Ihre Präsentation unter <https://www.cvh-server.de/vnc/> finden. (Die Screenshots in dieser Anleitung gelten für Kanal 1.)

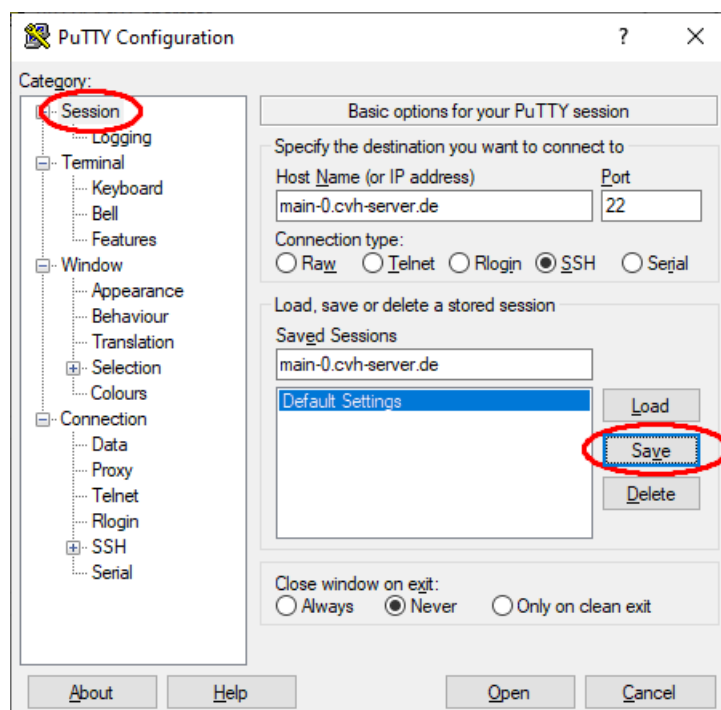


Wählen Sie links „Tunnels“ aus und kreuzen Sie rechts „Remote“ an. Tragen Sie unter „Source port“ die Port-Nummer **5901 für Kanal 1, 5902 für Kanal 2, ..., 5912 für Kanal 12 ein**. Unter „Destination“ tragen Sie unabhängig vom Kanal immer die Server-Port-Kombination **localhost:5900** ein. Drücken Sie anschließend auf „Add“.

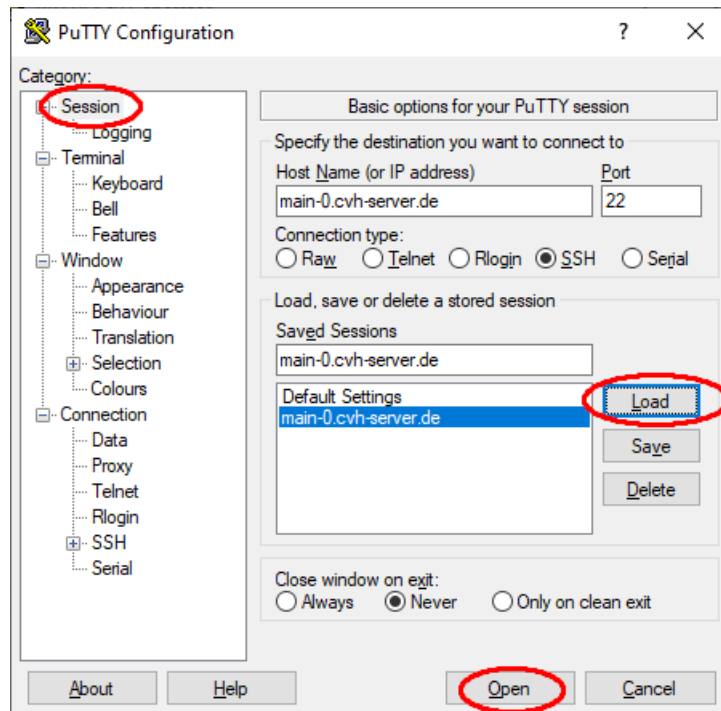
Danach erscheint der neu konfigurierte SSH-Tunnel in der Liste „Forwarded ports“.



(Hintergrund: 5900 ist die Port-Nummer, unter der Ihr VNC-Server Ihren Bildschirminhalt auf Ihrem eigenen Rechner zur Verfügung stellt. Damit *noVNC* auf dem CVH-Server diese Daten sehen kann, müssen wir diese auf dem CVH-Server, also „remote“, anbieten. *noVNC* erwartet diese Daten auf dem Kanal-Port 5901 bis 5912. Durch „Add“ wird dieser Tunnel einer Liste von Tunneln hinzugefügt, die dann zusammen mit SSH alle gleichzeitig aufgebaut werden. Für unseren Anwendungsfall benötigen wir nur einen Tunnel.)



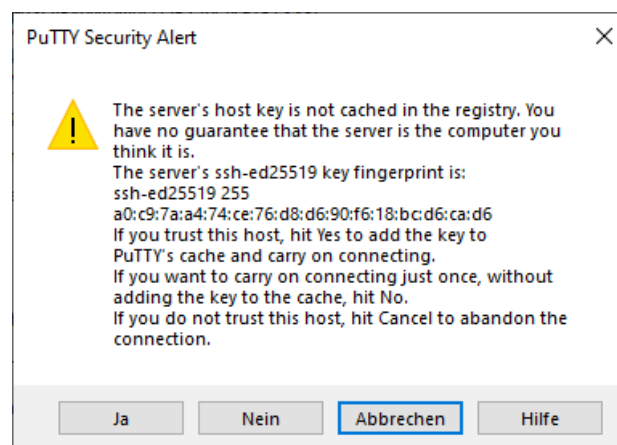
Wir kehren nun zurück zu „Session“ und speichern die Konfiguration. Sie bekommt dabei den Namen des Servers zugewiesen.



Wenn wir dann zu einem späteren Zeitpunkt *PuTTY* erneut starten, müssen wir diese Konfigurationen nicht erneut vornehmen, sondern können sie direkt laden.

Wir können nun mit „Open“ den Verbindungsaufbau starten.

Beim ersten Verbindungsaufbau präsentiert Ihnen *PuTTY* den „Fingerprint“ des öffentlichen Schlüssels des CVH-Servers.



Bestätigen Sie den Dialog **nur dann mit „Ja“, wenn der angezeigte Fingerprint mit dem hier abgedruckten genau übereinstimmt:**

```
ssh-ed25519 255
a0:c9:7a:a4:74:ce:76:d8:d6:90:f6:18:bc:d6:ca:d6
```

Nur diese Prüfung gewährleistet, daß Sie wirklich mit dem CVH-Server verbunden sind und nicht mit dem Rechner eines Angreifers.

Danach erscheint ein dunkles Terminal-Fenster, das während der Verbindung bestehen bleibt. (Hintergrund: Hier würde man normalerweise Befehle eingeben. Da wir nur den SSH-Tunnel brauchen, entfällt diese Funktion.)

Mit dem Start von sowohl *PuTTY* als auch dem VNC-Server ist das System betriebsbereit. Sie können nun Teilnehmende einladen, sich Ihren Bildschirm anzusehen, indem Sie ihnen die URL <https://www.cvh-server.de/vnc/> zusammen mit Ihrer **Kanal-Nr.** und Ihrem „**View-Only**“-**Passwort** zukommen lassen. (Auf Seite der Teilnehmenden ist keine spezielle Konfiguration erforderlich.)

Sobald Sie den SSH-Tunnel wieder schließen und/oder den VNC-Server beenden, wird die Verbindung abgebaut. Alle Teilnehmenden werden dann automatisch ausgeloggt. Sie erkennen dies auch daran, daß ihr normaler Bildschirmhintergrund zurückkehrt, der während der bestehenden VNC-Verbindung abgedunkelt war.

Viel Erfolg!

Stand: 30. März 2020

Copyright © 2020 Peter Gerwinski

Lizenz: CC-by-sa (Version 3.0) oder GNU GPL (Version 3 oder höher)

Sie können diese Anleitung einschließlich \LaTeX -Quelltext herunterladen unter:
<https://gitlab.cvh-server.de/pgerwinski/ow>