

Anleitung

Weitere Online-Werkzeuge

für Lehrveranstaltungen und Konferenzen

Inhalt

1	Übersicht	1
2	Hinweise für Vortragende	2
3	VNC mit SSH-Tunnel	4
4	SSH-Tunnel	6
5	VNC ohne SSH-Tunnel	13

1 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.

Es ist außerdem möglich, auch Gästen (z. B. Studierenden) zu erlauben, ihre Bildschirminhalte auf einfache Weise per VNC freizugeben, allerdings unverschlüsselt. Auch dies wird unten im Detail erklärt.

2 Hinweise für Vortragende

In Präsenzveranstaltungen hat man das Publikum stets vor sich und kann oft am „fragenden Blick“ erkennen, ob die vermittelte Botschaft angekommen ist oder nicht.

Die folgenden Hinweise sollen Vortragenden helfen, auch ohne visuellen Kontakt eine sinnvolle Interaktion mit dem Publikum aufrechtzuerhalten.

- Laut Aussagen von Studierenden im Sommersemester 2020 wird die Einblendung eines Kamerabildes der Lehrperson als hilfreich empfunden, die Atmosphäre einer *genau jetzt* stattfindenden Lehrveranstaltung und damit auch die Konzentration aufrechtzuerhalten.

Um dies auf einfache Weise zu ermöglichen, haben wir eine Schnittstelle entwickelt, die ein per *Jitsi Meet* eingespeistes Kamerabild per VNC an alle überträgt. Wenn gleichzeitig ein VNC-Bildschirm freigegeben ist, erscheint das zusätzliche Kamerabild klein in der unteren, rechten Ecke. Bei ausgeschaltetem VNC erscheint das Kamerabild im Vollbildmodus.

Um dies z. B. für VNC-Kanal 6 zu erreichen, rufe man die URL <https://jitsi.cvh-server.de/VNC6> auf und gebe das Kamerabild frei. (Das Mikrofon sollte man in diesem Fall ausschalten, da ja der Ton über *Mumble* übertragen wird.)

(Auch *OpenMeetings* kann ein Kamerabild an alle übertragen, war dabei jedoch in unseren Tests weniger zuverlässig als die o. a. Kombination von VNC und Jitsi Meet.)

- Es hat sich bewährt, zwischendurch mit einfachen Fragen zu prüfen, ob alle Teilnehmenden noch „dabei“ sind.

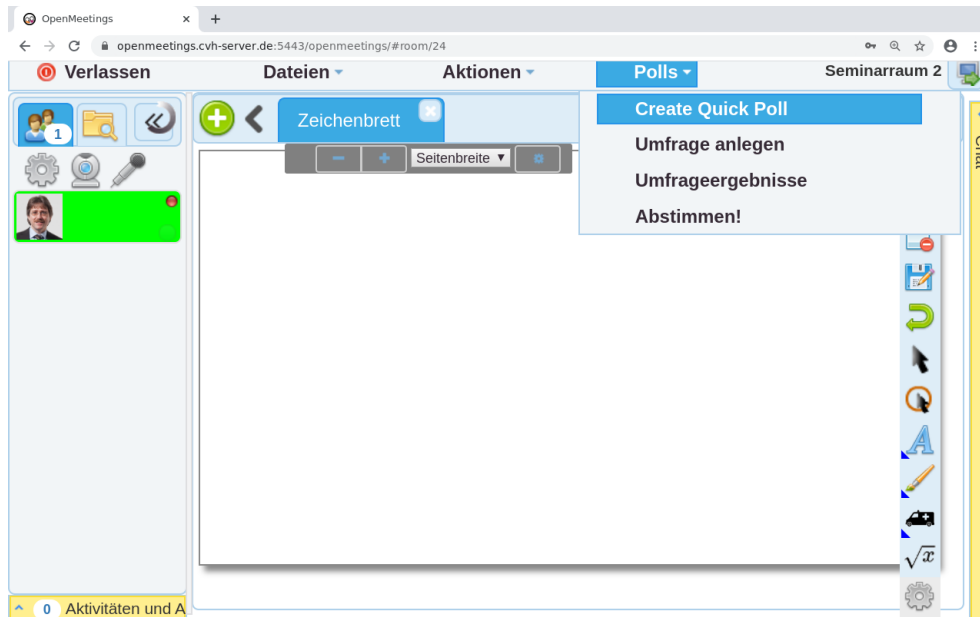
Damit die Aufmerksamkeit der Studierenden erhalten bleibt, sollte die Frage nicht jedesmal gleich lauten („Hat noch jemand Fragen?“), sondern z. B. auch: „Ist es für Sie in Ordnung, jetzt zum nächsten Thema überzugehen? Wer noch Fragen hat, antworte jetzt bitte mit ‚Nein‘.“

Um die Schwelle zum Antworten möglichst niedrig zu legen, haben wir in einem studentischen Projekt im Sommersemester 2020 die Software *CyberVotingHivemind-bot* (*cvh-bot*) entwickelt, mit deren Hilfe man im *Mumble*-Chat „mal eben schnell“ einfache Fragen stellen kann. Der *cvh-bot* zählt dann die eingegangenen Antworten und gibt eine Statistik darüber aus.

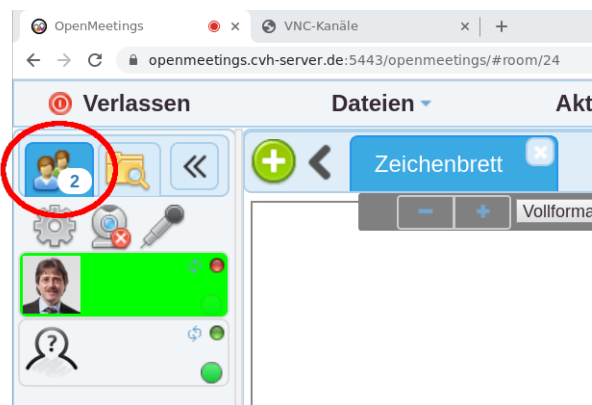
Der *cvh-bot* nimmt Befehle per *Mumble*-Chat entgegen, die mit einem Semikolon (;) beginnen, zum Beispiel:

- `;poll` für eine Ja-Nein-Umfrage,
- `;poll --:++` für eine Bewertungs-Umfrage mit den möglichen Antworten `--`, `-`, `o`, `+` und `++`,
- `;poll Ladung Spannung Strom` für eine Umfrage mit den möglichen Antworten `1` für „Ladung“, `2` für „Spannung“ und `3` für „Strom“
- und schließlich `;endpoll`, um eine Umfrage zu beenden und die Ergebnisse anzeigen zu lassen.

In *OpenMeetings* können Sie auch die Funktion „Quick Poll“ nutzen:



Um die Anzahl der eingegangenen Antworten einordnen zu können, empfiehlt sich ein Blick auf die die Gesamtzahl der Personen im „Raum“. Diese wird in *OpenMeetings* oben links angezeigt:



Ein großer Anteil von fehlenden Stimmen („Enthaltungen“) bei klaren Ja-Nein-Fragen ist ein Indiz dafür, daß noch Unklarheiten bestehen.

- Oft tut sich das Publikum schwer damit, sich per Sprache zu Wort zu melden. In Hörsälen ist dies sogar gar nicht möglich, ohne vorher das Wort erteilt bekommen zu haben. In solchen Situationen haben sich die Chat-Funktionen von *Mumble* und *OpenMeetings* als alternative Kommunikationskanäle bewährt.

In *OpenMeetings* werden neu eintreffende Chat-Botschaften standardmäßig durch ein akustisches Signal angezeigt („Quietscheente“), so daß man sie auch dann bemerken kann, wenn man selbst gerade im Vollbildmodus seine Folien präsentiert.

In *Mumble* muß man die akustische Benachrichtigung explizit aktivieren. Hierzu öffnet man im Menü „Konfiguration“ den Dialog „Einstellungen...“, wählt den Reiter „Nachrichten“, scrollt im Hauptfenster bis ganz unten und aktiviert schließlich für „Textnachricht“ und für „Private Textnachricht“ die Option „Sounddatei“:

Nachricht	Konsole	Benachrichtigung	Hervorheben	Text-zu-Sprache	Sounddatei	Pfad
Benutzer umbenannt	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Selbst stumm gestellt.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	:/SelfMutedDeafened.ogg
Stummstellung aufgehoben.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	:/SelfMutedDeafened.ogg
Selbst taub gestellt.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	:/SelfMutedDeafened.ogg
Taubstellung deaktiviert.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	:/SelfMutedDeafened.ogg
Anderer stumm/taub gestellt	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Benutzer stumm gestellt (du selbst)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	:/UserMutedYouOrByYou.ogg
Benutzer stumm gestellt (durch dich)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	:/UserMutedYouOrByYou.ogg
Benutzer stummgestellt (jemand anderes)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Kanal betreten	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Kanal betreten (verschoben)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Benutzer betrat Kanal	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	:/UserJoinedChannel.ogg
Benutzer verließ den Kanal	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	:/UserLeftChannel.ogg
Benutzer hat sich mit dem Server verbunden ...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	:/UserJoinedChannel.ogg
Benutzer verließ den Kanal und hat den Serve...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Erlaubnis verweigert	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	:/PermissionDenied.ogg
Textnachricht	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	:/TextMessage.ogg
Private Textnachricht	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	:/TextMessage.ogg

Text zu Sprache
Lautstärke
Längen-Schwellwert Eigene Textnachrichten vorlesen

Flüstern
 Flüstern nur von Freunden akzeptieren

Nachrichtenverlauf
Maximale Länge des Nachrichtenverlaufs

Zurücksetzen Voreinstellungen OK Abbrechen Anwenden

Alternativ kann *Mumble* Textnachrichten auch vorlesen. Dies läßt sich am schnellsten über das Menü „Konfiguration“, Menüpunkt „Text-zu-Sprache“ ein- und ausschalten.

- In *Mumble* möchte man gelegentlich das Mikrofon aus- und wieder einschalten. Es empfiehlt sich, dafür ein Tastenkürzel zu definieren: Menü „Konfiguration“, Dialog „Einstellungen. . .“, Reiter „Tastenkürzel“. Diese Taste ist dann **immer aktiv, auch wenn das *Mumble*-Fenster gerade nicht im Vordergrund ist**. Man sollte daher eine Taste wählen, die von anderen Programmen nicht verwendet wird.

Um auch ohne Blick auf das *Mumble*-Fenster zu wissen, ob das eigene Mikrofon gerade an oder aus ist, sollte man dies zusätzlich akustisch durch *verschiedene* Sound-Dateien signalisieren lassen (siehe oben: Menü „Konfiguration“, Dialog „Einstellungen. . .“, Reiter „Nachrichten“). Ich habe mir dafür die Sound-Dateien [mic-off.wav](#) und [mic-on.wav](#) erzeugt, die Sie durch Anklicken oder unter <https://gitlab.cvh-server.de/pgerwinski/ow> herunterladen können.

- Vortragende haben in *Mumble* und *OpenMeetings* gewisse Sonderrechte, z. B. anderen das Wort zu erteilen und zu entziehen oder bestimmte Räume zu betreten. Wenn Sie davon Gebrauch machen möchten, sprechen Sie uns an.
- Wenn Sie in *Mumble* in einem „belebten Campus“ mit möglicherweise Hunderten von Studierenden den Raum wechseln wollen, ist die *Drag-and-drop*-Methode nicht praktikabel. In diesem Fall sollten Sie zu der alternativen Methode greifen und den Ziel-Raum mit der rechten Maustaste anklicken. Es öffnet sich dann ein Menü, über das Sie den Raum („Kanal“) betreten können.

3 VNC mit SSH-Tunnel

Dieser Abschnitt beschreibt die Verwendung von *VNC* in Kombination mit einem SSH-Tunnel, also mit Schutz gegen Abhören und Manipulation. Wenn Sie dieses Maß an

Sicherheit nicht benötigen, können Sie **VNC** auch auf sehr viel einfachere Weise nutzen und direkt in Abschnitt 5 (Seite 13) weiterlesen.

Es gibt verschiedene Anbieter von VNC. Unter GNU/Linux gibt es darüberhinaus verschiedene Methoden, VNC zu nutzen. Am einfachsten installiert man **x11vnc** (siehe Seite 13) und startet es mit **x11vnc**. Damit ist VNC für den Einsatz in Kombination mit einem SSH-Tunnel vorbereitet. (Beenden mit **^C**.)

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

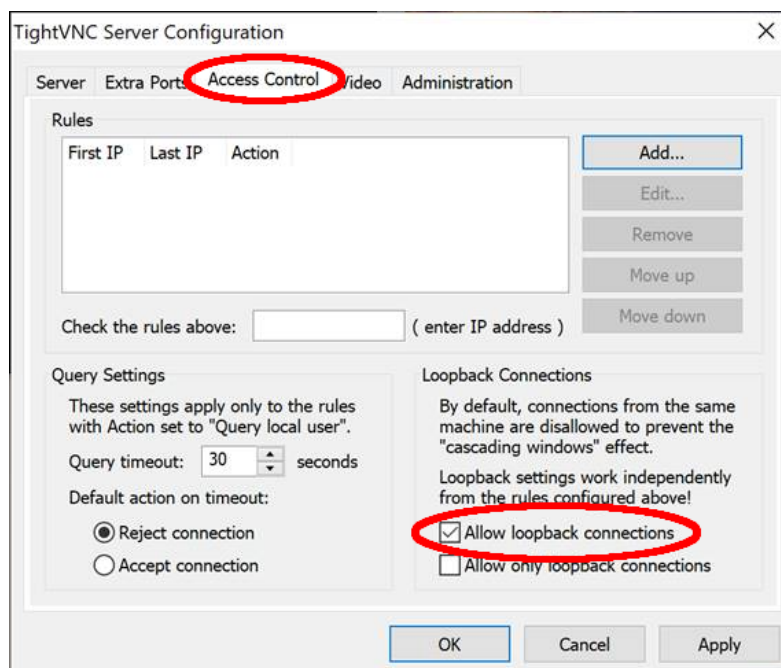
- Laden Sie **TightVNC** von <https://www.tightvnc.com/download.html> herunter und führen Sie eine Standard-Installation durch, abgesehen von einem Punkt: Wir empfehlen, den **automatischen Start** von VNC als Windows-Service zu **deaktivieren** und stattdessen den VNC-Server bei Bedarf manuell zu starten.

(Für den Fall, daß Sie VNC bereits als Service installiert haben und den automatischen Start deaktivieren wollen, geht dies auf folgende Weise: Über Windows-Taste + R das Programm **services.msc** starten, in der Liste den VNC-Service ausfindig machen und von „automatisch“ auf „manuell“ umstellen.)

- 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 Passwörter: ein primäres Passwort für Ihren eigenen Vollzugriff von außen (für Fernwartung, in unserem Fall ungenutzt) sowie ein „View-Only“-Passwort, mit dem der CVH-Server Ihren Bildschirminhalt lesen kann.

Dieses Passwort können wir gleichzeitig für das Web-Interface nutzen, um Ihren Bildschirm für die Betrachtung von außen freizugeben. Beide View-Only-Passwörter können aber auch verschieden sein (z. B. ein sicheres Zufalls-Passwort für den Server und ein leicht zu merkendes für das menschliche Publikum).

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



Damit ist VNC für den Einsatz in Kombination mit einem SSH-Tunnel vorbereitet.

Wenn Sie mehrere Monitore angeschlossen haben, gibt TightVNC diese gemeinsam frei, und sie erscheinen bei der Betrachtung stark verkleinert. Wir haben für MS-Windows 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“.

4 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 XXXX:localhost:59YY ssh-tunnel@main-0.cvh-server.de
```

Das `XXXX` steht hierbei für den von uns für Sie reservierte **Tunnel-Port-Nummer** von `5913` bis `5918`, das `YY` für die Nummer Ihres VNC-Servers, typischerweise `00` bei Verwendung von `x11vnc`.

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

```
1024 SHA256:/KBfmrspi3R4U8Q+tvwUK8HuFcxwK6ej/y1kH+Uupg (DSA)
256 SHA256:kkDVI02KLySNK2cu0kHvDGXqakeIeJ3EWzF3cBL/WU8 (ECDSA)
256 SHA256:S2PJlnABOQNOKj8fnPNWBjoNhEZK/AJ+kwFzKqTtex8 (ED25519)
2048 SHA256:z2hdccP25OE2cOypCrN4V7EUv8AGUY4gnTjHK6g33pE (RSA)
```

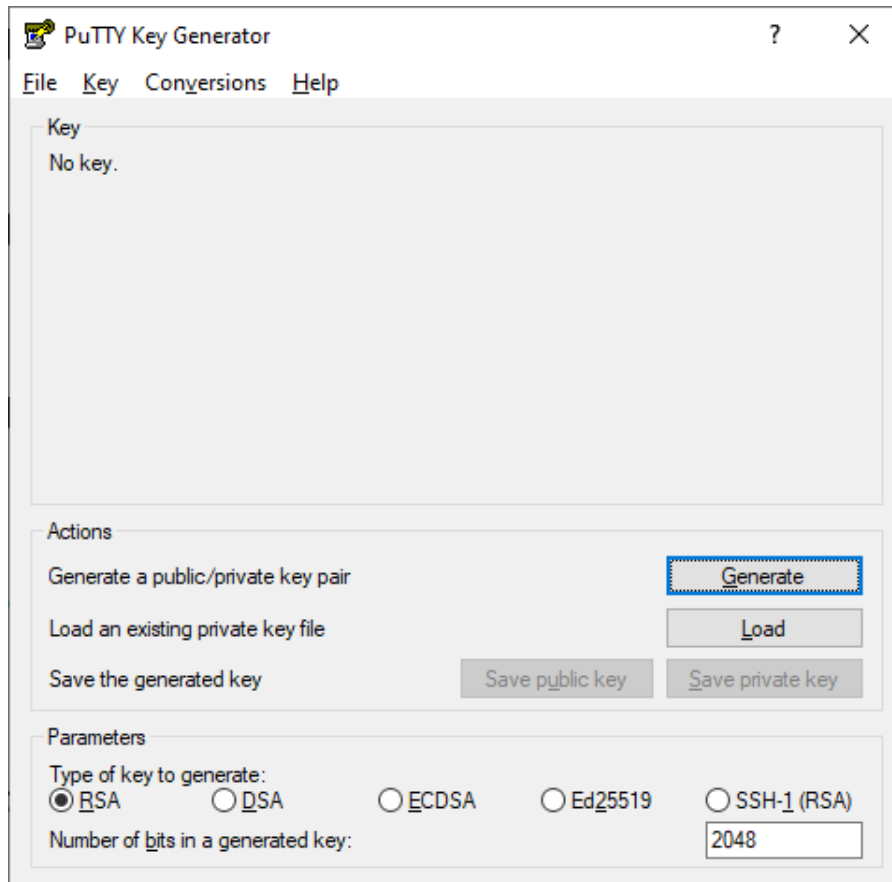
Damit ist der SSH-Tunnel aufgebaut.

- 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.



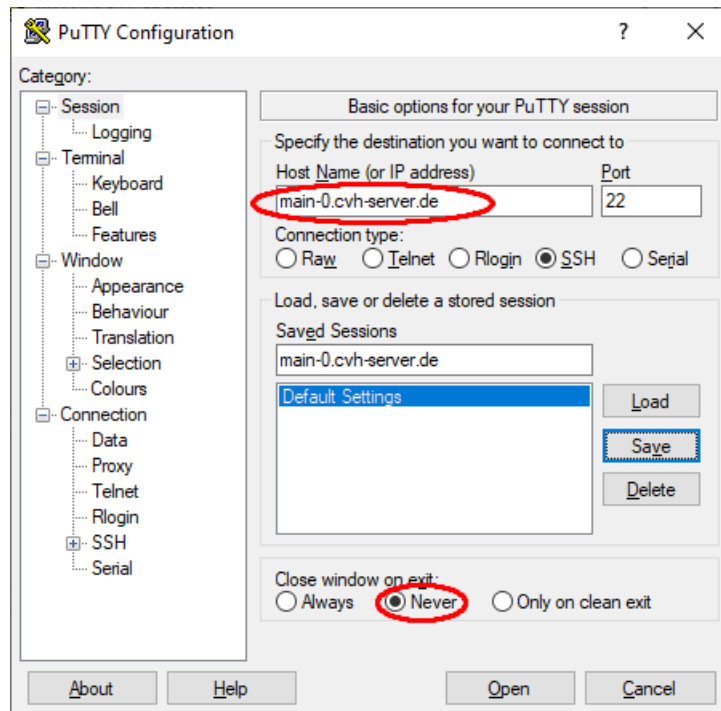
Die Schlüsselerzeugung benötigt kryptographisch sicheren Zufall. Diesen zu erzeugen, kann einige Zeit in Anspruch nehmen. Um diesen Prozeß zu unterstützen, lohnt es sich, z. B. mit der Maus Bewegungen auszuführen, aus denen die Software Zufall extrahieren kann. Die Schlüsselerzeugung wird dadurch wesentlich 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**. Teilen Sie uns bei dieser Gelegenheit auch gleich Ihr View-Only-Passwort für VNC mit, damit wir es auf dem Server hinterlegen können.

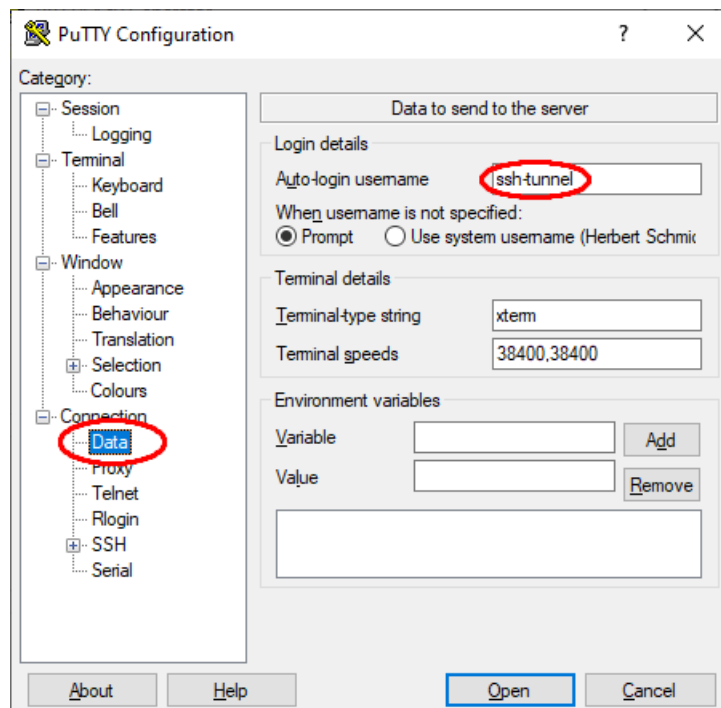
Wer im Besitz des privaten Schlüssels („private key“) ist, kann sich gegenüber dem CVH-Server als Sie ausweisen. Diese Datei sollte daher sicher aufbewahrt werden.

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

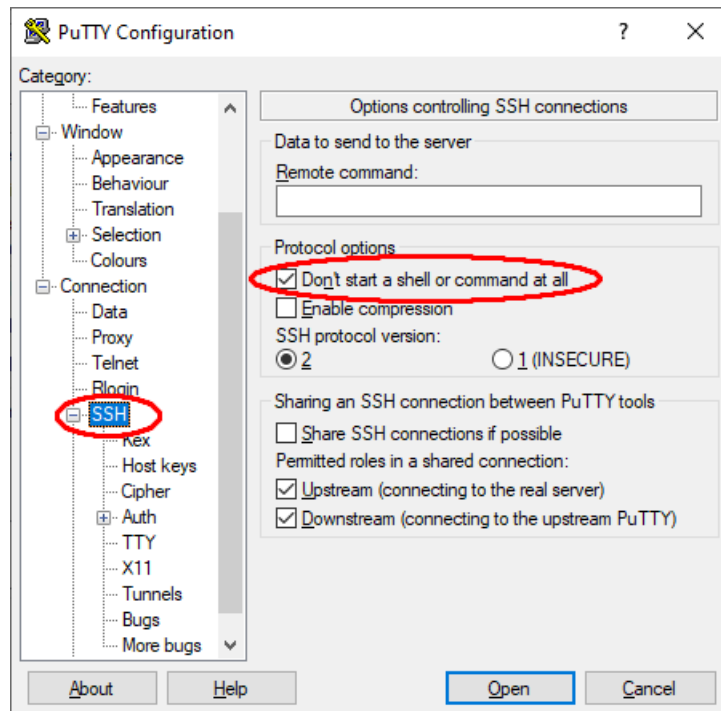


Tragen Sie unter „Host Name“ den Server-Namen main-0.cvh-server.de ein. (Der Standard-Port 22 ist in Ordnung.)

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

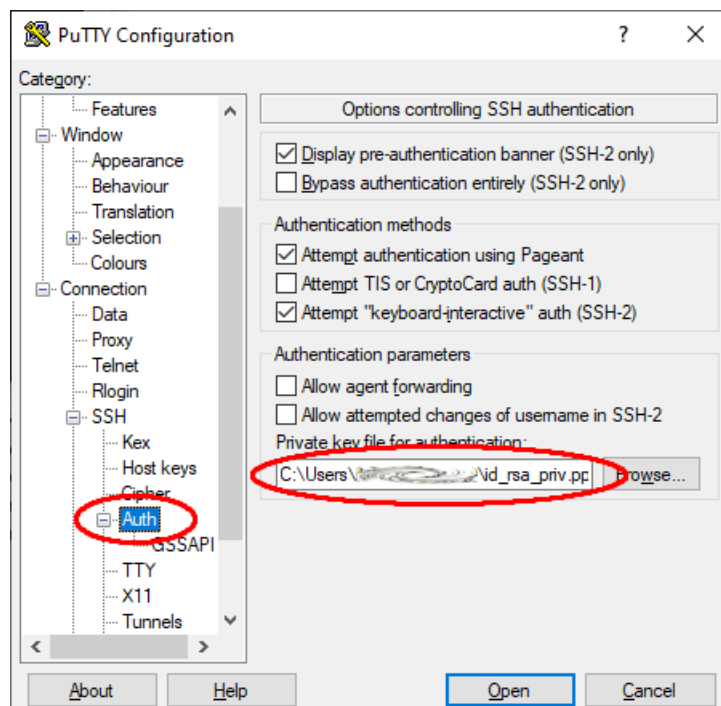


Öffnen Sie links „Connection“ und wählen Sie dort „Data“ aus. Tragen Sie anschließend unter „Auto-login username“ den Benutzernamen ssh-tunnel ein.



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.)

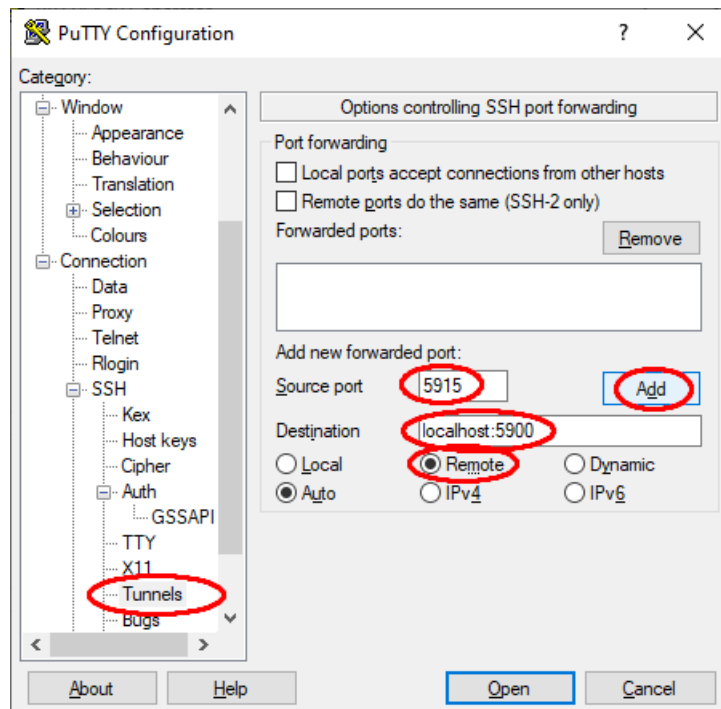


Gehen Sie nun links auf „Auth“ und tragen Sie die Datei ein, in der Sie Ihren **privaten Schlüssel** gespeichert haben.

(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.)

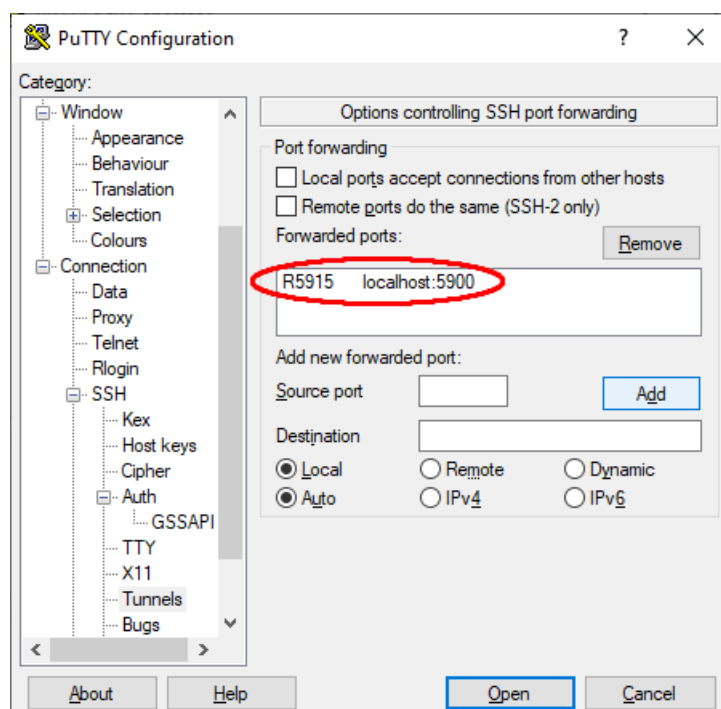
Nun konfigurieren wir den eigentlichen Tunnel.

Sie benötigen dafür eine eindeutige **Tunnel-Port-Nummer** von 5913 bis 5918, die wir für Sie reservieren müssen. (Parallel dazu reservieren wir für Sie auch einen eindeutigen **Kanal** von 1 bis 6, anhand dessen man Ihre Präsentation unter <https://www.cvh-server.de/vnc/> finden kann.)

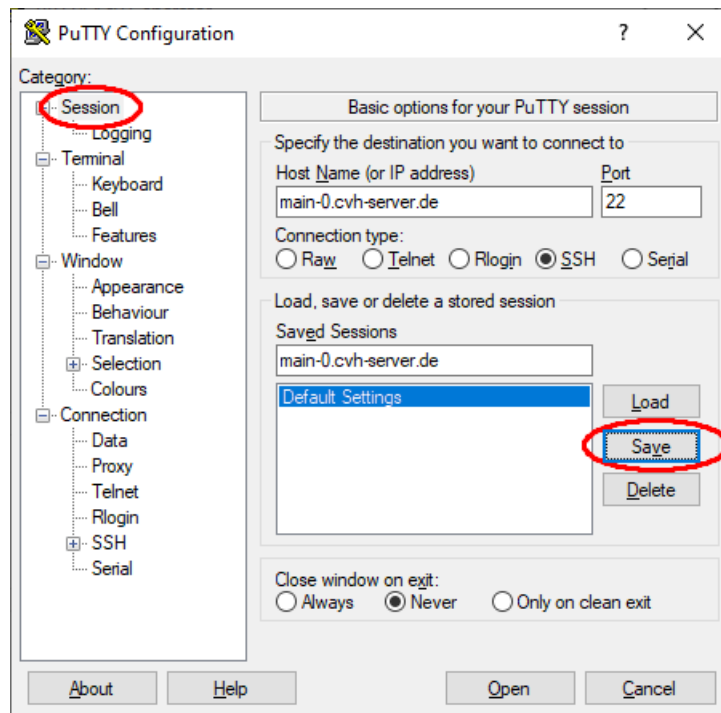


Wählen Sie links „Tunnels“ aus und kreuzen Sie rechts „Remote“ an. Tragen Sie unter „Source port“ Ihre Tunnel-Port-Nummer ein. (In unserem Beispiel ist dies 5915.) Unter „Destination“ tragen Sie unabhängig von **Kanal** und **Tunnel-Port-Nummer** immer die Server-Port-Kombination **localhost:5900** ein. Drücken Sie anschließend auf „Add“.

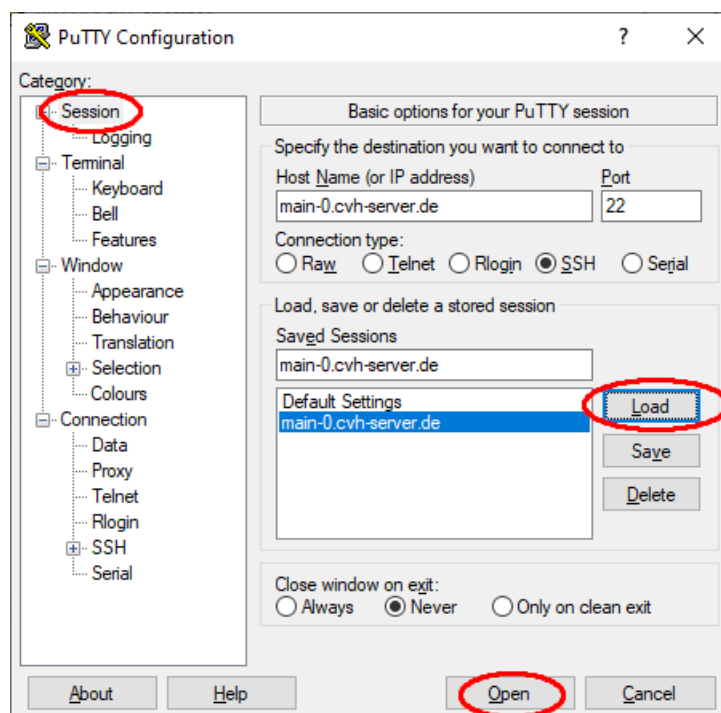
Danach erscheint der neu konfigurierte SSH-Tunnel in der Liste „Forwarded ports“.



(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 einem Tunnel-Port im Bereich von 5913 bis 5918. 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 einzigen Tunnel.)



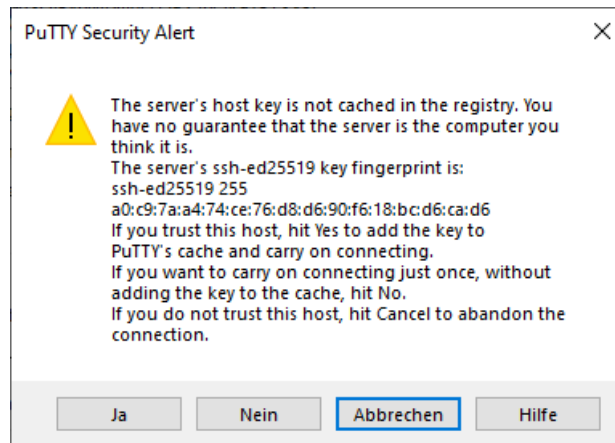
Wir kehren nun zurück zu „Session“ und speichern die Konfiguration. Sie bekommt dabei standardmäßig den Namen des Servers zugewiesen.



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.

Wenn Sie während einer VNC-Sitzung den SSH-Tunnel schließen oder dieser aus irgendeinem Grund abreißt (z. B. infolge eines Netzwerk-Problems), müssen Sie auch den VNC-Server erneut starten, damit das Bild auf dem CVH-Server wieder sichtbar wird. Umgekehrt ist es *nicht* notwendig, den SSH-Tunnel neu zu starten, wenn der VNC-Server neu gestartet wurde.

5 VNC ohne SSH-Tunnel

Für manche interaktiven Lehrformate ist es wünschenswert, daß auch Studierende ihren Desktop-Inhalt präsentieren. Hierfür jedesmal einen SSH-Tunnel einzurichten, wäre zu aufwendig. Tatsächlich ist es aber möglich, auch ohne SSH-Tunnel ein VNC-Bild an den CVH-Server zu senden.

Diese Methode ist unverschlüsselt und daher unsicher. Das Bild kann mit einfachen Mitteln mitgelesen und sogar manipuliert werden. Diese Methode sollte daher nur in Situationen zum Einsatz kommen, in denen es auf Vertraulichkeit nicht ankommt und in denen Sie Authentizität nötigenfalls auf andere Weise gewährleisten können.

Um nun auf diese Weise **unter MS-Windows** den Bildschirminhalt eines Rechners zum CVH-Server zu senden, sind die folgenden Schritte erforderlich:

- Standard-Installation von TightVNC (<https://www.tightvnc.com/download.html>), dabei den VNC-Windows-Service deaktivieren (siehe Seite 5)
- VNC-Server starten
- Sobald der VNC-Server läuft, mit einem Rechtsklick auf das „V“-Symbol in der Statusleiste das Menü öffnen und dort den Punkt „Attach Listening Viewer. . .“ auswählen
- Dort trägt man

`main-0.cvh-server.de:5`

ein und bestätigt den Dialog. (5 ist hier die Nummer des VNC-Kanals.)

Anschließend erscheint das Bild im VNC-Web-Interface des CVH-Servers, in diesem Beispiel Kanal 5, und kann mit dem Passwort des Kanals betrachtet werden.

Unter GNU/Linux gibt es mehrere Methoden, Bildschirmhalte per VNC an den CVH-Server zu exportieren. Die einfachste lautet:

- Standardinstallation von x11vnc (z. B. `sudo apt install x11vnc`)
- `x11vnc -connect main-0.cvh-server.de:6`

Anschließend erscheint der Inhalt des aktuell laufenden X-Servers im VNC-Web-Interface des CVH-Servers, in diesem Beispiel Kanal 6, und kann mit dem Passwort des Kanals betrachtet werden.

Anstelle des gesamten Inhalts aller angeschlossenen Monitore kann `x11vnc` auch einzelne Monitore oder einzelne Fenster per VNC exportieren. Für Details dazu siehe: `man x11vnc`

In allen Fällen ist es insbesondere *nicht* notwendig, ein VNC-Passwort zu vergeben, da man den Bildschirm aktiv exportiert, anstatt anderen Zugriff darauf zu gewähren.

Viel Erfolg!

Stand: 12. November 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>