

# Working Remote

Christian Wilms

Computer Vision Group  
Universität Hamburg

Sommersemester 2018

04. Juni 2018

# Remote arbeiten

## Was bedeutet remote arbeiten?

- Ihr sitzt Zuhause an eurem Rechner und könnt die Ressourcen am Ikum nutzen.
- Ihr müsst nicht raus nach Stellingen!!!

## Für wen ist das wichtig?

- wenn eurer Rechner zu langsam ist
- wenn ihr mehrere/größere Rechner braucht
- wenn ihr mit der Grafikkarte (Deep Learning, CNNs) arbeiten wollt

# Was braucht ihr, um remote zu arbeiten?

## Verbindung zum entfernten Rechner

- der entfernte Rechner muss eingeschaltet und unter Ubuntu gebootet sein
- mit ssh, bei Windows über PuTTY
- so könnt ihr Befehle auf einem entfernten Rechner ausführen

## Datentransfer zum Ikum-Server

## Terminal Multiplexer

# Was braucht ihr, um remote zu arbeiten?

Verbindung zum entfernten Rechner

Datentransfer zum IkuM-Server

- über scp oder über FileZilla oder oder oder
- so könnt ihr Daten von eurem Rechner in euer Homeverzeichnis (`~/xName`) kopieren und umgekehrt

Terminal Multiplexer

# Was braucht ihr, um remote zu arbeiten?

Verbindung zum entfernten Rechner

Datentransfer zum Ikum-Server

Terminal Multiplexer

- nur relevant, wenn ihr lange auf einer Maschine arbeitet
- sorgt dafür, dass ihr euren Rechner ausschalten könnt, ohne, dass eure Session auf dem entfernten Rechner abbricht

# Welche Rechner mit GPU sind verfügbar?

## RZ-Computer

- die Rechner in den Räumen D-010, D-114 und D-118/119 haben 4 GB GPUs
- ihr habt keine Kontrolle darüber, ob die Rechner angeschaltet sind und welches Betriebssystem läuft
- Namen: rzpc<Nummer>

## KOGS-Computer

Falls Pakete fehlen oder sonstige Fehler auftreten, meldet euch!

# Welche Rechner mit GPU sind verfügbar?

## RZ-Computer

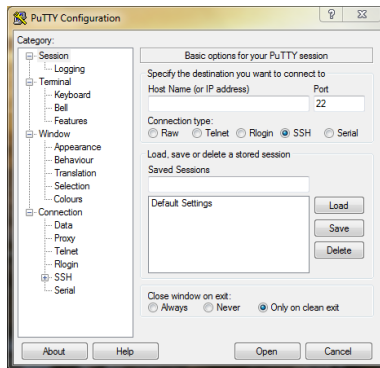
## KOGS-Computer

- die 7 Rechner im Raum R-131 haben 2 GB GPUs
- eigentlich sollten diese stets eingeschaltet und unter Ubuntu gebootet sein
- falls nicht, schreibt mir eine Mail
- Namen: kogspc15, kogspc16, kogspc29, kogspc36, kogspc37, kogspc53, kogspc54

Falls Pakete fehlen oder sonstige Fehler auftreten, meldet euch!

# Windows - Arbeiten mit PuTTY - I

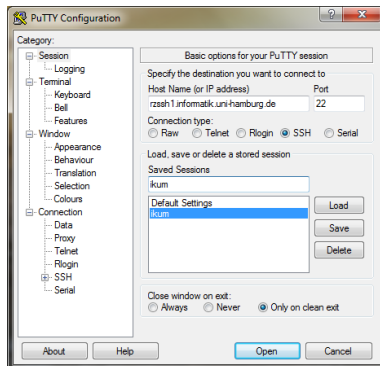
Ladet euch das Programm PuTTY herunter und startet PuTTY.





# Windows - Arbeiten mit PuTTY - II

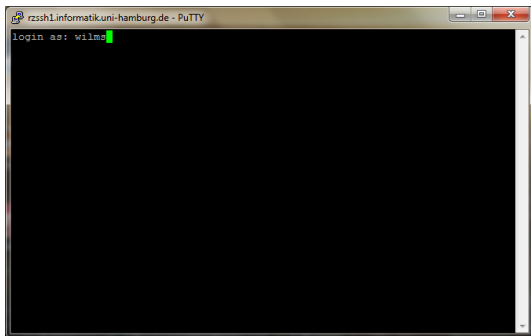
Gebt unter Host Name `rzssh1.informatik.uni-hamburg.de` ein.



Tipp: Wenn ihr bei Saved Sessions einen Namen angebt und Save drückt, werden eure Einstellungen gespeichert.

## Windows - Arbeiten mit PuTTY - III

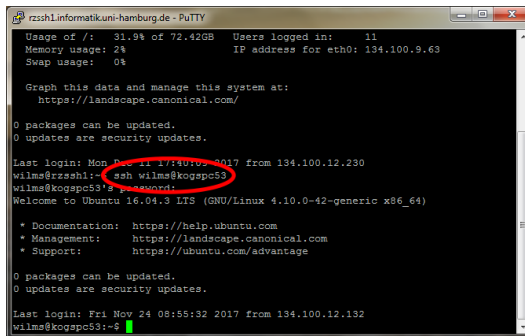
Es öffnet sich eine Konsole, in der ihr zunächst euren Benutzernamen (xNachname) eingibt.



Danach wird euer Passwort abgefragt.

# Zu einem anderen Rechner wechseln

Jetzt seid ihr auf rzssh1 eingeloggt. Um zu einem anderen Rechner zu gelangen, könnt ihr in der Konsole ssh <Benutzernamen>@<Rechnername> eingeben. Beispielweise:



```
rzssh1.informatik.uni-hamburg.de - PuTTY
Usage of /: 31.9% of 72.42GB    Users logged in: 11
Memory usage: 2%              IP address for eth0: 134.100.9.63
Swap usage: 0%

Graph this data and manage this system at:
https://landscape.canonical.com/

0 packages can be updated.
0 updates are security updates.

Last login: Mon Nov 13 17:40:06 2017 from 134.100.12.230
wilms@rzssh1:~$ ssh wilms@kogspc53
wilms@kogspc53's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.10.0-42-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.

Last login: Fri Nov 24 08:55:32 2017 from 134.100.12.132
wilms@kogspc53:~$
```

Ggf. wird euer Passwort erneut abgefragt.

# Basisbefehle in der Konsole

## Navigieren in der Ordnerstruktur

Ihr startet immer in eurem Homeverzeichnis (~/)!

`ls` listet alle Ordner und Dateien im aktuellen Verzeichnis aus

`cd` wechselt das Verzeichnis auf das angegebene (zum Wechsel auf das Homeverzeichnis `cd ~/`)

`mkdir` legt einen neuen Ordner an mit dem übergebenen Namen

## Arbeiten mit Python-Skripten

# Basisbefehle in der Konsole

## Navigieren in der Ordnerstruktur

## Arbeiten mit Python-Skripten

`python` startet das angegebene Python-Skript (bspw. `python 6.3.py`)

`htop` zeigt eine Art Taskmanager an (welche Programme laufen aktuell)

`who` wer ist gerade eingeloggt auf diesem Rechner

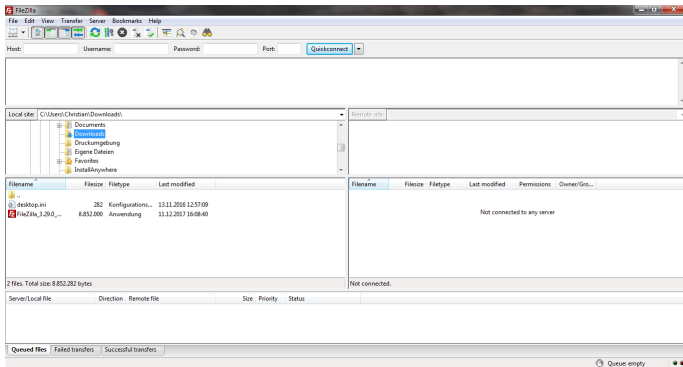
`nvidia-smi` eine Art Taskmanager für die Grafikkarte

# Linux/Mac - ssh-Verbindung aufbauen

- Konsole öffnen
- ssh  
    <Benutzername>@rzssh1.informatik.uni-hamburg.de  
    eingeben
- Passwort korrekt eingeben
- Fertig! Der Rest funktioniert wie vorher beschrieben.

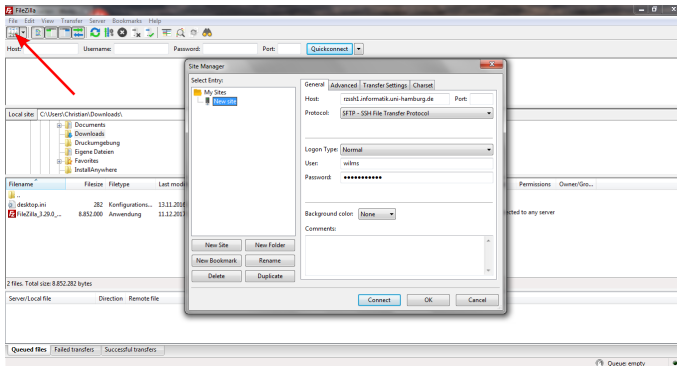
# Windows - Arbeiten mit FileZilla - I

Ladet euch das Programm FileZilla herunter, installiert es ohne weitere Toolbars etc. zu installieren und startet das Programm.



# Windows - Arbeiten mit FileZilla - II

Geht auf das Symbol oben links (Site Manager) und gebt eure Daten dort ein:

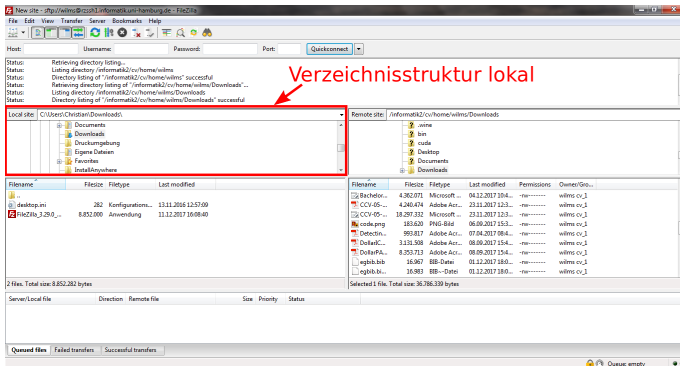


Stellt danach eine Verbindung mit connect her.



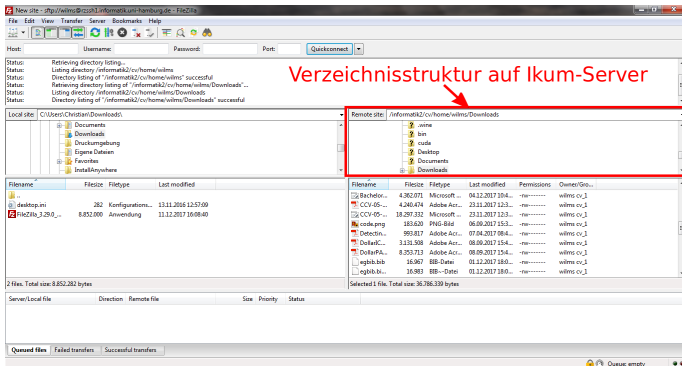
# Windows - Arbeiten mit FileZilla - III

Jetzt seht ihr in der linken Hälfte die Verzeichnisstruktur auf eurem Rechner und auf der rechten Hälfte die Struktur in einem Homeverzeichnis. Jeweils unter den Bäumen sind die Inhalte der aktuell ausgewählten Ordner zu sehen.



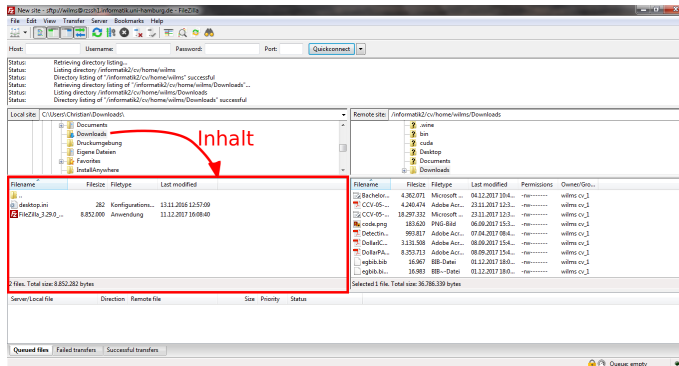
## Windows - Arbeiten mit FileZilla - III

Jetzt seht ihr in der linken Hälfte die Verzeichnisstruktur auf eurem Rechner und auf der rechten Hälfte die Struktur in einem Homeverzeichnis. Jeweils unter den Bäumen sind die Inhalte der aktuell ausgewählten Ordner zu sehen.



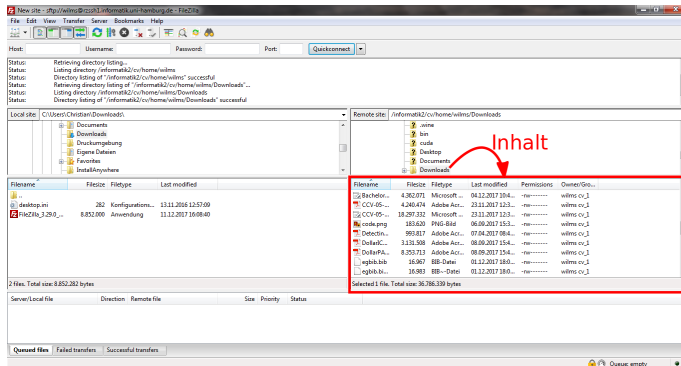
# Windows - Arbeiten mit FileZilla - III

Jetzt seht ihr in der linken Hälfte die Verzeichnisstruktur auf eurem Rechner und auf der rechten Hälfte die Struktur in einem Homeverzeichnis. Jeweils unter den Bäumen sind die Inhalte der aktuell ausgewählten Ordner zu sehen.



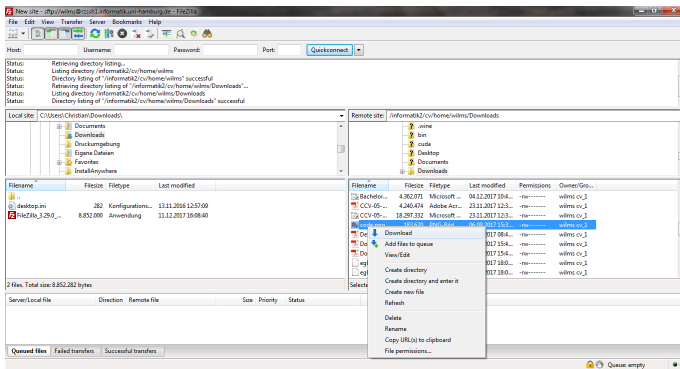
# Windows - Arbeiten mit FileZilla - III

Jetzt seht ihr in der linken Hälfte die Verzeichnisstruktur auf eurem Rechner und auf der rechten Hälfte die Struktur in einem Homeverzeichnis. Jeweils unter den Bäumen sind die Inhalte der aktuell ausgewählten Ordner zu sehen.



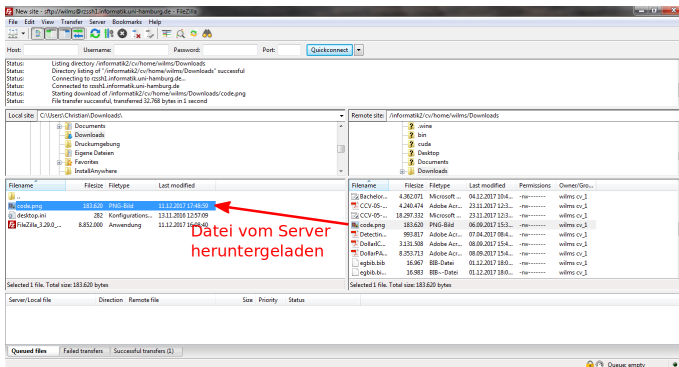
# Windows - Arbeiten mit FileZilla - III

Über das Kontextmenü der Dateien der beiden angezeigten Ordner könnt ihr nun die Dateien hoch- bzw. herunterladen.



# Windows - Arbeiten mit FileZilla - III

Über das Kontextmenü der Dateien der beiden angezeigten Ordner könnt ihr nun die Dateien hoch- bzw. herunterladen.



# Linux/Mac - Datentransfer mit scp

Unter Linux/Mac könnt ihr auch FileZilla verwenden. Alternativ dazu könnt ihr mit scp auch wie folgt vorgehen:

Ihr könnt auch direkt mit scp in der Konsole arbeiten. Bspw.:

```
scp mySkript.py
```

```
<Benutzername>@rzssh1.informatik.uni-hamburg.de:~/cvPrak/  
kopiert die Datei mySkript.py aus dem aktuellen Ordner des  
lokalen Rechners in den Unterordner cvPrak des Homeverzeichnis.
```

# Start mit tmux

Wenn ihr PuTTY oder die ssh-Verbindung schließt, ist eure Session auch auf dem entfernten Rechner beendet! D.h. alle Programme und Python-Skripte, die ihr gestartet habt, werden sofort beendet. tmux verhindert das!

## tmux-Workflow



# Start mit tmux

tmux verhindert das!

## tmux-Workflow

- 1 nach dem Login auf der Zielmaschine (bspw. kogspc42) startet ihr eine tmux-Session mit `tmux new -s <Name>`
- 2 jetzt erscheint quasi ein neues Fenster im Terminal, steuert nun euer Zielverzeichnis an
- 3 startet das Python-Skript bspw. mit `python 6.3.py`
- 4 nun könnt ihr PuTTY einfach schließen und die tmux-Session läuft auf dem entfernten Rechner weiter
- 5 Wenn ihr euch wieder auf dem entfernten Rechner einloggt, könnt ihr mit `tmux attach -t <Name>` die Session wieder aufnehmen und schauen, wie weit die Verarbeitung fortgeschritten ist.

# tmux-Befehle

Vor der Ausführung eines tmux-Befehl muss jeweils `strg+b` gedrückt werden!

- `c` öffnet einen neuen Reiter innerhalb der Session (siehe grüne Leiste unten)
- `<Ziffer>` wechselt auf den Reiter mit der angegebenen Ziffer
- `alt gr+8` ermöglicht das Scrollen mit Bild auf/ab
- `d` von einer Session lösen