



Remote Connections

SSH en VNC

**DE HOGESCHOOL
MET HET NETWERK**

Hogeschool PXL – Dep. PXL-IT – Elfde-Liniestraat 26 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook



SSH

- OpenSSH
 - Secure SHell
 - OpenSSH-client
 - is standaard geïnstalleerd op Ubuntu Server en Ubuntu Desktop
 - OpenSSH-server
 - dient geïnstalleerd te zijn op de PC die we vanop afstand willen managen
 - Bv. Een Ubuntu-server(serverrack) managen vanaf je Ubuntu Desktop(laptop)
 - is standaard geïnstalleerd
 - Op Ubuntu Server 18.04 (cloud edition)
 - is niet standaard geïnstalleerd
 - Op Ubuntu Server 18.04 (standard edition)
 - maar kan wel aangevinkt worden tijdens installatie
 - Op Ubuntu Desktop 18.04



SSH-Server

- SSH-server

- Installatie

- `sudo apt install openssh-server`

- Configuratie

- `sudo vi /etc/ssh/sshd_config`

ListenAddress - indien we op een bepaalde NIC willen luisteren

MaxSessions - Hoeveel gelijktijdige connecties toegelaten worden

PermitRootLogin - op "no" voor security (na login sudo...)

DenyUsers - Deze gebruikers mogen niet inloggen over ssh

DenyGroups - De gebruikers van deze groepen mogen niet inloggen



Meer opties voor sshd_config vind je hier terug:
[man sshd_config](#)

SSH-server

- SSH gebruikt poort 22 op de Server
 - `grep ssh /etc/services`
 - toont poort 22 over TCP
 - `ss -ln 'sport = ssh'` → toont dat er enkel geluisterd wordt via TCP op Port 22
 - `ss -lt4` → toont listening Port ssh
 - `ss -lt4n` → toont listening Port 22
 - `ss -at4` → toont zowel de listening, als de established
 - `ss -o state established '(dport = ssh or sport = ssh)'`
 - toont alle verbonden connecties van enkel Port 22



SSH-client

- SSH-client

- Installatie

- ssh-client is automatisch geïnstalleerd
 - `sudo apt install openssh-client`

- Configuratie

- `/etc/ssh/ssh_config`
 - staat standaard goed
 - Je kan bvb wel de `VisualHostKey` op `yes` zetten om telkens de ASCII Art te zien bij het connecteren naar een server

```
student@ubdesk:~$ sudo apt install openssh-client
[sudo] password for student:
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-client is already the newest version (1:7.2
0 upgraded, 0 newly installed, 0 to remove and 138
```

```
student@ubdesk1804:~$ ssh student@192.168.87.128
Host key fingerprint is SHA256:9y6dCz03rmVnjXTnNP+W
+---[ECDSA 256]---+
|      . .      |
|    . + .      |
|      + +      |
|    . . O . .  |
| . . . S O . . .+|
| O . . . +=+E. ==|
|   + .O +.O=.X *|
| . . . . .oooX *.|
|   .O. . .+OO .|
+-----[SHA256]-----+
student@192.168.87.128's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-65-
```



SSH Server Authentication

- Server Authentication

- Een Public Key van de server wordt gebruikt om zich te authenticeren bij de client
 - `/etc/ssh/ssh_host_rsa_key.pub` OF `/etc/ssh/ssh_host_dsa_key.pub` OF `/etc/ssh/ssh_host_ecdsa_key.pub` OF `/etc/ssh/ssh_host_ed25519_key.pub`
 - Via de setting `StrictHostKeyChecking` bij de client (`/etc/ssh/ssh_config`)
 - Standaard op Ask
 - Elke eerste verbinding naar een nieuwe host wordt er gevraagd of je dit wil en zo ja wordt de public key opgeslagen op de client in de known-hosts-file (`~/.ssh/known_hosts`)
 - Indien de public key van een bestaande server wijzigt, zal de client niet kunnen connecteren naar deze host
 - Op te lossen door de “oude” public key van de server te verwijderen uit de file (op de client) en opnieuw te connecteren



SSH-connecties met username/pwd

- SSH-connectie
 - `ssh <gebruikersnaam>@<serverip>`
 - De eerste maal wordt gevraagd of je wel wilt connecteren met deze onbekende server
 - Indien je bevestigd wordt de public key van de server opgeslaan op de client in `~/.ssh/known_hosts` (homefolder van de user)
 - kan ook server-wide ingesteld worden door handmatig de public key(s) van de ssh-server(s) op te slaan in `/etc/ssh/ssh_known_hosts`
 - `ssh <serverip>`
 - indien je geen naam opgeeft voor de connectie zal er getracht worden om in te loggen met de gebruiker die het commando uitvoert



```

student@ubdesk:~$ ssh student@172.16.110.128
The authenticity of host '172.16.110.128 (172.16.110.128)' can't be established.
ECDSA key fingerprint is SHA256:d+zh1iKQscfWXoinWbUvRCll3rigjThlXAk1xtiWZ8s.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.110.128' (ECDSA) to the list of known hosts.
student@172.16.110.128's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-87-generic x86_64)

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

109 packages can be updated.
57 updates are security updates.

Last login: Wed Nov 22 10:43:03 2017
student@ubuntuServer:~$ exit
logout
Connection to 172.16.110.128 closed.
student@ubdesk:~$ cat .ssh/known_hosts
|1|JoHqkCrSYvE+rLGqJR5UIXBJNi0=|0LPbtaE/M4qKysOWrTkTt4/57uE= ecdsa-sha2-nistp256
zbJ+mIYxijBPEpnlVeAb2TPx5GKpvYqNTenIXs5lB+43KHU=

```

Eerste keer aanloggen op een nog onbekende ssh-server

Je kan deze fingerprint controleren op de server door het commando:

```
ssh-keygen -l -f /etc/ssh/ssh_host_ecdsa_key.pub
```

```
256 SHA256: d+zh1iKQscfWXoinWbUvRCll3figjThlXAk1xtiWZ8s. root@ubserv (ECDSA)
```



Nadien nogmaals aanloggen op een reeds gekende ssh-server

```

student@ubdesk:~$ ssh student@172.16.110.128
student@172.16.110.128's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-87-generic x86_64)

 * Documentation:  https://help.ubuntu.com

```


SSH-connecties met keys

- SSH met keys
 - ook passwordless ssh genoemd
 - er wordt een private/public-keypair gemaakt op de client (Desktop)
 - private-key blijft op de client (Desktop) en is persoonlijk
 - public-key wordt gekopieerd in de homedir van de persoon op server waarmee en waarnaar we willen connecteren over ssh
 - de public-key kan hergebruikt worden om tegelijk de mogelijkheid te hebben met meerdere servers passwordless te connecteren over ssh
 - met dezelfde private key dus



SSH-connecties met keys

- SSH keypair

- Aanmaken

- ssh-keygen -t ecdsa
 - met eventueel een -b 256 (default), 384 of 521 voor hogere encryptie
 - private-key kan extra beveiligd worden met een passphrase
 - Het keypair staat nu in ~/.ssh
 - private-key: id_ecdsa
 - public-key: id_ecdsa.pub

```
student@ubdesk:~$ ssh-keygen -t ecdsa
Generating public/private ecdsa key pair.
Enter file in which to save the key (/home/student/.ssh/id_ecdsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/student/.ssh/id_ecdsa.
Your public key has been saved in /home/student/.ssh/id_ecdsa.pub.
The key fingerprint is:
SHA256:sickdmQAt46ivtaPBNeI+QNs6jH4Fv6lge0sVlv901o student@ubdesk
The key's randomart image is:
+----[ECDSA 256]----+
| ..o                |
| . o                |
| . o                |
| . oooo            |
| .B.o+. = S        |
| =.*+ = +          |
| ++==+ + oE        |
| o**+o+ o.o        |
| +*=o+. ...o       |
+----[SHA256]-----+
```

```
student@ubdesk:~$ ls -l ~/.ssh/
total 12
-rw----- 1 student student 227 okt 27 10:14 id_ecdsa
-rw-r--r-- 1 student student 176 okt 27 10:14 id_ecdsa.pub
-rw-r--r-- 1 student student 222 okt 27 06:15 known_hosts
```



SSH-connecties met keys

- SSH keypair
 - Public-key naar de server kopiëren
 - onder de gebruiker waarmee je wil inloggen over ssh
 - `ssh-copy-id [-i ~/.ssh/id_rsa.pub] <gebruiker>@<serverip>`
 - om te mogen kopiëren naar de homefolder van deze gebruiker moeten we het wachtwoord opgeven van deze gebruiker
 - `-i ~/.ssh/id_rsa.pub` moet je niet meegeven als je de default bestandsnaam (en pad) gebruikt

```
student@ubdesk:~$ ssh-copy-id student@172.16.110.128
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
student@172.16.110.128's password:
```

```
Number of key(s) added: 1
```

```
Now try logging into the machine, with: "ssh 'student@172.16.110.128'"
and check to make sure that only the key(s) you wanted were added.
```

De public key komt op de **server** in **authorized_keys**
in de **homefolder** van de te connecteren user

```
student@ubserv:~$ cat ~/.ssh/authorized_keys
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTY=
W4z+Zmv+DMNVTzSS+R4UYvP009R9GoyP2U+ovpZy4/1I4BLMAGvk0xV1G54= student
```

SSH-connecties met keys

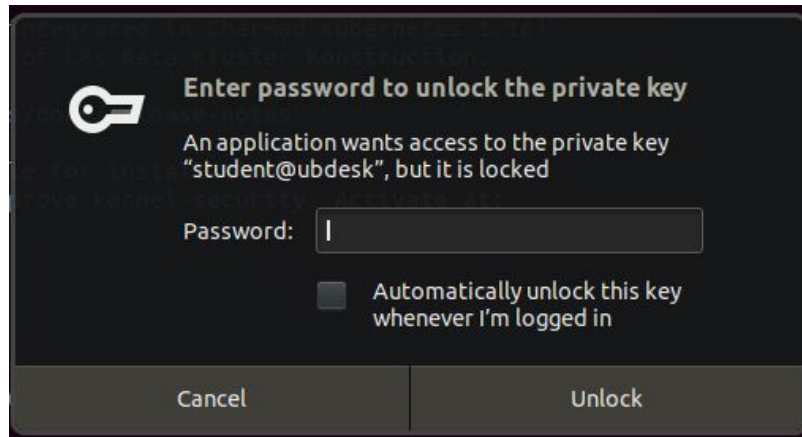
- SSH keypair
 - Verder beveiligen van de ssh-server
 - aanpassen van `/etc/ssh/sshd_config`
 - `PasswordAuthentication no`
 - geeft aan of er met een paswoord mag worden ingelogd
 - Reloaden van de sshd-configuratie
 - `sudo systemctl reload ssh`



SSH-connecties met keys

- SSH keypair
 - Passwordless connecting over ssh vanaf een ubuntu Desktop

```
student@ubdesk:~$ ssh student@172.16.110.128
```



De passphrase om de private key te unlocken wordt gevraagd. Je kan aanvinken dat de private key in de toekomst automatisch wordt ge-unlocked als je succesvol ingelogd bent.



```
student@ubdesk:~$ ssh student@172.16.110.128
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-87-generic x86_64)

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

Dan wordt er geen wachtwoord meer gevraagd

SSH-connecties met keys

- SSH keypair
 - Passwordless connecting over ssh zonder X (X = GUI = Ubuntu Desktop)
 - Als je hier ook slechts éénmaal je private-key wilt unlocken en vervolgens meerdere malen gebruiken voor verscheidene ssh-connecties
 - `ssh-agent bash` - start een nieuwe shell met de agent running
 - `ssh-add ~/.ssh/id_ecdsa` - houdt de private key(s) in het geheugen
 - We moeten dus niet telkens opnieuw de passphrase opgeven als we een nieuwe ssh-connectie starten

```
student@ubuntuserver02:~$ ssh-agent bash
student@ubuntuserver02:~$ ssh-add ~/.ssh/id_ecdsa
Enter passphrase for /home/student/.ssh/id_ecdsa: 
Identity added: /home/student/.ssh/id_ecdsa (/home/student/.ssh/id_ecdsa)
student@ubuntuserver02:~$ ssh student@192.168.202.128
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-39-generic x86_64)
```

* Documentation: <https://help.ubuntu.com>



SSH-connecties debuggen

- Indien een bepaalde connectie niet werkt
 - kan je gaan troubleshooten door te debuggen
 - je krijgt dan veel meer informatie op de server te zien als je met een client connectie begint te maken
 - Eerst moet je de huidige ssh-server stoppen
 - `sudo systemctl stop ssh`
 - Hierna kan je de versie met debugging starten
 - `sudo /usr/sbin/sshd -ddd`
 - Indien je een foutmelding krijgt moet je eerst nog een directory aanmaken
 - `mkdir /run/sshd`
 - Connecteer nu vanaf de client en kijk naar de meldingen in het terminal-venster van de server



SSH - extra security

- SSH-server: extra security
 - AllowUsers (of DenyUsers)
 - aan te duiden in /etc/ssh/sshd_config op server
 - om verbindingen van bepaalde gebruikers toe te laten of te verbieden
 - Al wie niet in de AllowUsers is opgenomen, is dan wel automatisch geweigerd!
 - Indien DenyUsers en AllowUsers beiden bestaan, worden ze in deze volgorde verwerkt!
 - Men kan ook werken met AllowGroups of DenyGroups
 - Dan is de volgorde
DenyUsers, AllowUsers, DenyGroups, AllowGroups

```
AllowUsers gert guy@webserver.pxl.be tom@172.16.110.55  
            bart@*.kinepolis.be
```



SSH - extra security

- SSH-server: extra security
 - hosts.allow en hosts.deny
 - om verbindingen vanaf andere PCs toe te laten of niet
 - PCs en/of subnets toelaten
 - /etc/hosts.allow
 - sshd: 172.16.231.0/255.255.255.0
 - Alle andere PCs en subnets verbieden
 - /etc/hosts.deny
 - sshd: ALL
 - De volgorde waarop deze bestanden verwerkt worden:
 - indien een match in hosts.allow dan connectie toegelaten
 - indien in match in hosts.deny dan connectie verboden
 - indien geen match in één van deze files dan toegelaten



SSH - motd en nologin

- SSH-server: Overige bestanden uit /etc
 - /etc/motd
 - inhoud wordt afgedrukt als in sshd_config PrintMotd op yes staat
 - /etc/nologin
 - indien deze file bestaat, kan niemand inloggen, behalve root, en wordt de tekst in dit bestand getoond
 - touch nologin is al voldoende
 - maar de tekst die in dit bestand staat wordt ook getoond



SSH - commando's sturen

- Commando's sturen over ssh
 - in plaats van een interactieve sessie te starten met ssh, kan je ook onmiddellijk een commando meegeven aan je connectie
 - `ssh <gebruiker>@<ssh-server> '<commando>'`
 - vb: `ssh student@172.16.110.128 'echo $HOSTNAME; ip a'`
 - na het uitvoeren van het commando stopt de connectie
 - het commando wordt remote uitgevoerd, maar de output wordt lokaal getoond
 - Gebruik optie t om een interactieve sessie te starten
 - `ssh -t student@172.16.110.128 'vi test.sh'`
 - om programma's te runnen die een tty (pseudo-terminal) nodig hebben



SSH - files kopiëren met scp

- Files kopiëren over ssh met scp
 - scp
 - secure copy (over ssh) tussen twee PCs, waarvan één de lokale PC moet zijn
 - scp <lokaal bestand> <user>@<serverip>:<doelmap>
 - doelmap start in de homefolder van de gebruiker waarmee geconnecteerd wordt, of er moet een absoluut pad gebruikt worden (beginnend met /)
 - scp ~/Desktop/oef10_1.sh student@172.16.110.128:oefeningen/
 - De doeldirectory (oefeningen) moet wel bestaan



SSH - files kopiëren met scp

- Files kopiëren over ssh met scp
 - scp
 - je kan ook een bestand kopiëren van de server naar client
 - `scp student@172.16.110.128:oefeningen/oef11_2.sh Desktop/`
 - je kan een bestand tijdens het kopiëren ook hernoemen
 - `scp oef11_3.sh student@172.16.110.128:oefeningen/oef11_3.oud`

Opgelet: Het scp-commando wordt enkel uitgevoerd op de client!



SSH - files kopiëren met scp

- Een map kopiëren over ssh met scp
 - `scp -r`
 - kopieert recursief de inhoud van de map en submappen
 - `scp -r <lokale map> <user>@<serverip>:<doelmap>`
 - doelmap start in de homefolder van de gebruiker waarmee geconnecteerd wordt, of er moet een absoluut pad gebruikt worden (beginnend met /)
 - `ssh student@172.16.110.128 'mkdir CDR'`
 - `scp -r /media/cdrom/ student@172.16.110.128:CDR/`



SSH - secure ftp

- Files kopiëren over ssh met sftp
 - sftp
 - Secure File Transfer Protocol (FTP over SSH)
 - werkt indien ssh werkt
 - sftp <gebruiker>@<serverip>
 - help
 - ls/l ls
 - cd/lcd
 - pwd/lpwd
 - get/put
 - mkdir/lmkdir
 - rm/rmdir
 - !<localcommand>
 - bye/quit
 - kan ook mee geconnecteerd worden vanuit Filezilla



SSH - sshfs

- Een filesystem mounten over ssh
 - sshfs installeren op de client
 - `sudo apt install sshfs`
 - directory aanmaken onder je homedir
 - `mkdir sshmount`
 - directory over ssh lokaal mounten
 - `sshfs student@172.16.110.128: sshmount/`

VIA `/etc/fstab` (genereer, als zijnde root (onder zijn homefolder) een rsa-keypair zonder passphrase en kopieer de public key naar de andere server)

```
sshfs#root@172.16.110.128: /home/student/sshmount fuse comment=sshfs,defaults,users,allow_other,_netdev 0 0
```

`allow_other` → dan mogen anderen, dan diegene die de mount uitgevoerd heeft, ook aan de inhoud van de mount



SSH - sshfs

- Een filesystem mounten over ssh
 - zorgen dat de connectie behouden blijft
 - `sudo vi /etc/ssh/ssh_config`
 - `ServerAliveInterval 120`
 - `sudo systemctl reload ssh`
 - unmounten van een sshfs-mount
 - `sudo fusermount -u <mountpoint>`
OF
 - `sudo umount <mountpoint>`



SSH-X11 forwarding

- X11
 - Client-Server architectuur
 - Normale toestand op een Ubuntu-desktop
 - Een grafische applicatie is de X-client
 - Vraagt aan de server om een beeld te renderen
 - De X-server maakt het beeld en brengt het naar het beeldscherm



SSH-X11 forwarding

- X11-Forwarding
 - Er wordt via SSH aangelogd op een ssh-server
 - Op deze ssh-server wordt een grafische applicatie gestart
 - Deze applicatie vraagt aan de X-server om het beeld te renderen
 - De vraag van de client wordt nu gesteld aan de X-server die draait op de SSH-client
 - Dus de connectie tussen de X-client en X-server wordt gelegd over de ssh-verbinding van ssh-server naar ssh-client



SSH-X11 forwarding

```
student@ubserv:~$ ip a s ens33 | grep 'inet ' | tr -s ' ' | cut -d ' ' -f3
192.168.110.128/24
student@ubserv:~$ sudo apt install x11-apps
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
```

```
student@ubserv:~$ xclock
Error: Can't open display:
```

```
student@ubdesk:~$ ssh -X student@192.168.110.128 xclock
Warning: locale not supported by C library, locale unchanged
```



Op de Server installeren we een GUI-applicatie (bvb om een FW te managen). Maar deze kan natuurlijk niet gestart worden op deze server, omdat er geen X-server aanwezig is.

Vanaf de Desktop kunnen we via SSH toch de applicatie draaien die op de Server wordt opgestart.



SSH - connecties vanuit windows

- SSH connecties maken vanuit windows
 - Putty
 - www.putty.org => klik op download
 - download putty.zip en pak uit op het bureaublad
 - putty.exe
 - ip-adres instellen en connectie maken
 - Je kan ook met een keypair werken
 - met puttygen
 - om een keypair te maken in windows
 - met pageant
 - zodanig dat je de passphrase niet telkens opnieuw moet opgeven

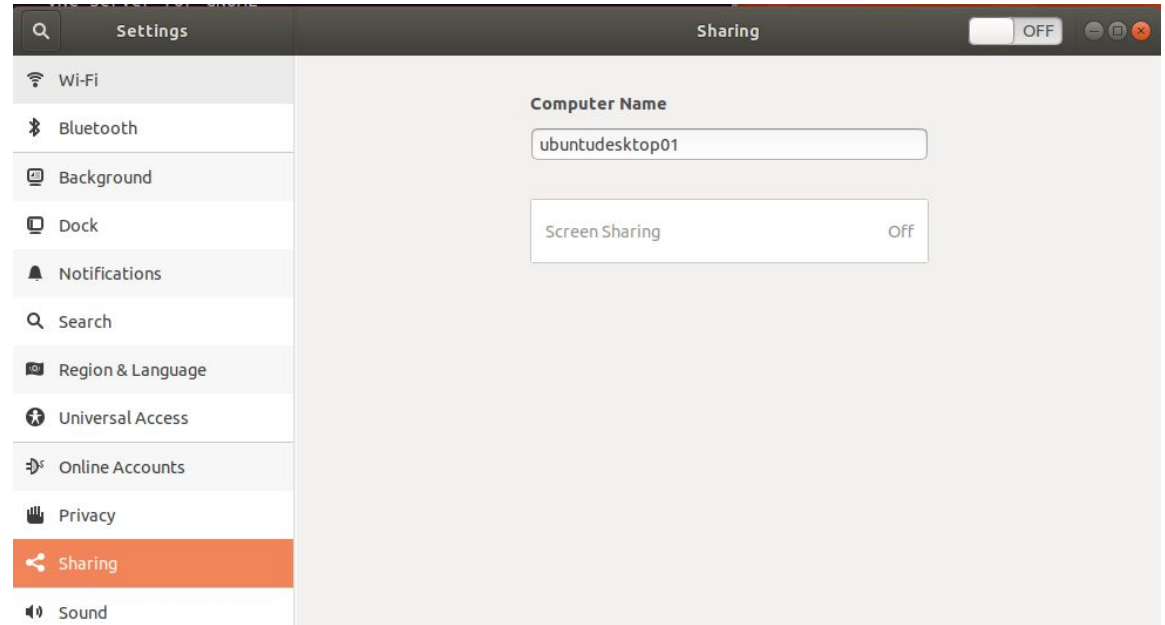
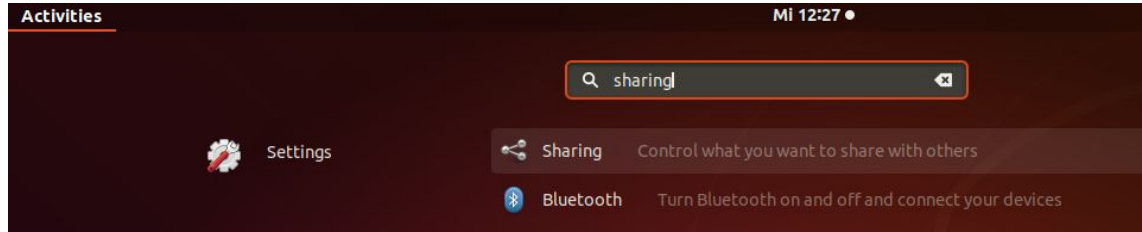


VNC

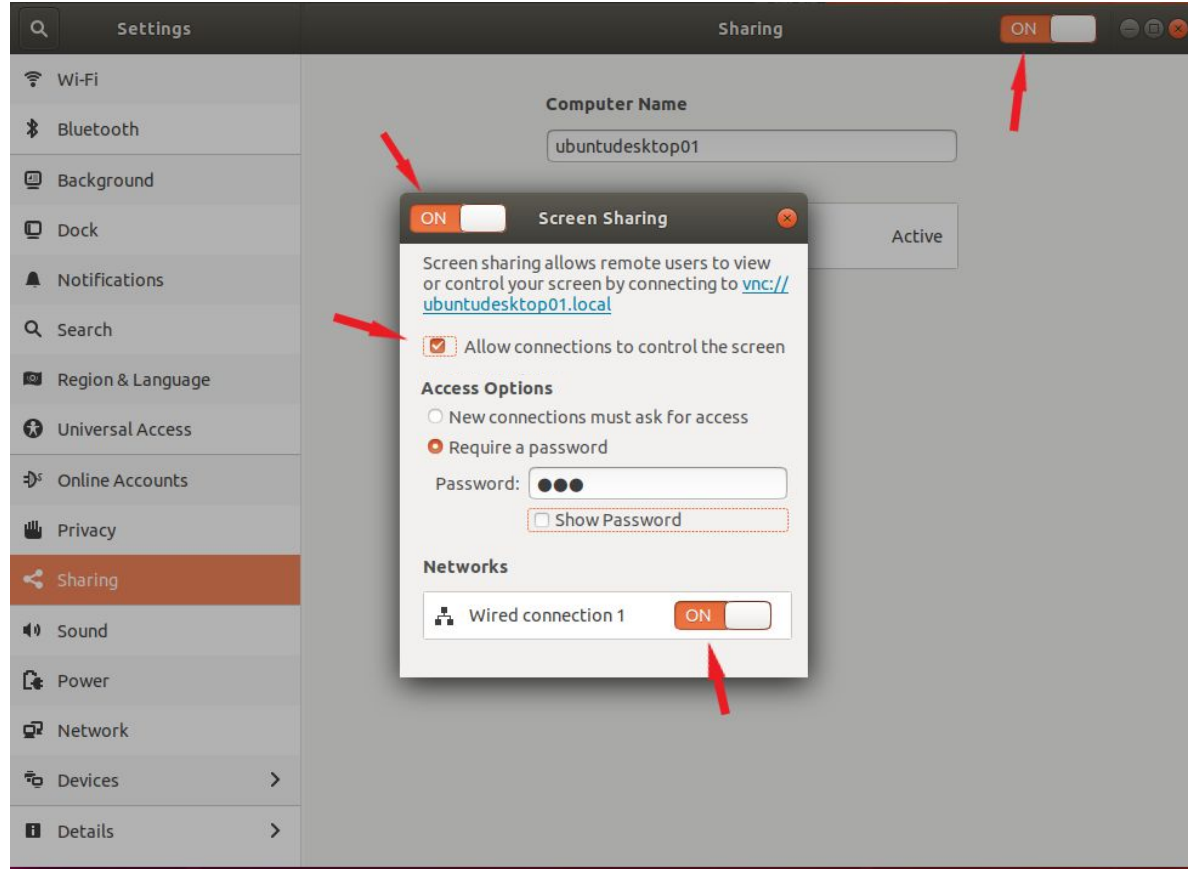
- VNC
 - Virtual Network Computing
 - Remote Control via Desktop Sharing
 - Ingebakken in Ubuntu Desktop
 - Server via “Screen Sharing”
 - Client via “Remmina Remote Desktop Client”



VNC - Vino Remote Desktop Server



VNC - Vino Remote Desktop Server



VNC - Remmina Remote Desktop Client

