

Network Access

We need to make sure we have network access to our server even if we don't have a display.

- [Remote Terminal](#)
- [Remote Desktop](#)
- [File Sharing](#)

Remote Terminal

SSH – or Secure Shell – allows us to connect to our server over our local network to run commands remotely.

This is very helpful for running an always-on server that doesn't have a display attached, often called a "headless" system. During the Debian install process, we opted to enable SSH by support.

When connecting to a new host, SSH will create a digital fingerprint for it. When we connect to this server in the future, it will compare the server's fingerprint to the one currently on file. In the event they don't match in the future, SSH will warn us that there is a mismatch and someone might be impersonating your server.

There are many different ways to connect to your server using SSH. We will highlight some common methods for connecting to our server and controlling it remotely.

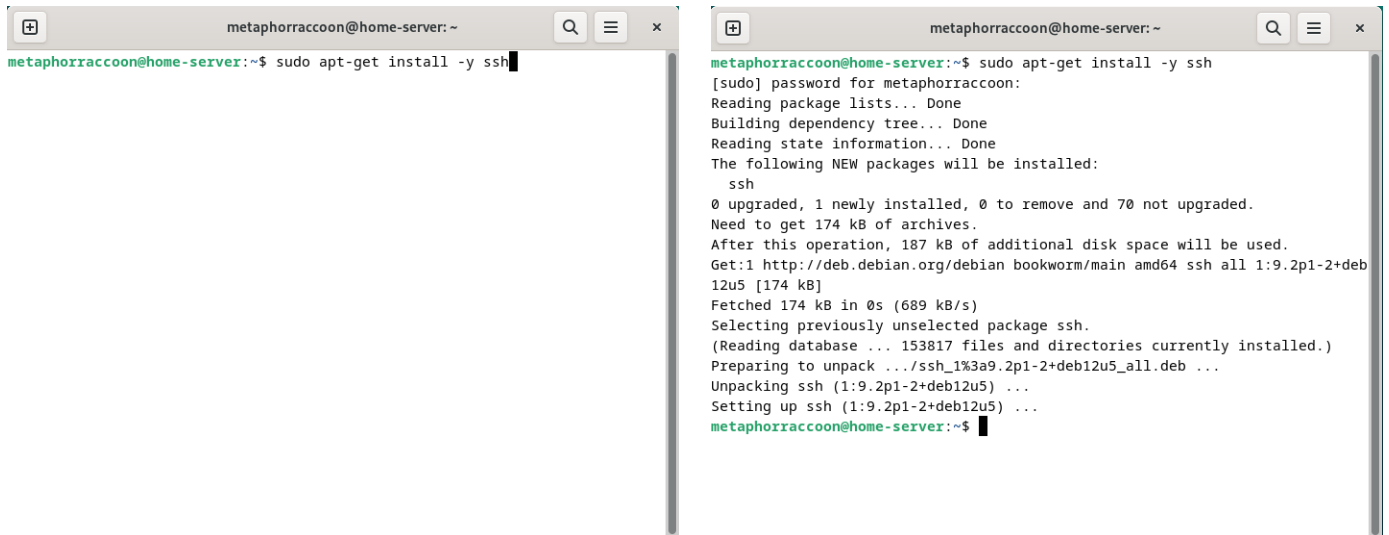
Linux

You can use the terminal available on most Distributions of Linux – such as Debian, Ubuntu, and Linux Mint – to remotely connect to your server.

You will need to know your [username set during the installation process](#) and the IP Address you set during [network configuration](#).

You can open a terminal and ensure that SSH is installed by running the command:

```
sudo apt-get install -y ssh
```



```
metaphorraccoon@home-server: ~  
metaphorraccoon@home-server:~$ sudo apt-get install -y ssh  
  
metaphorraccoon@home-server:~$ sudo apt-get install -y ssh  
[sudo] password for metaphorraccoon:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following NEW packages will be installed:  
  ssh  
0 upgraded, 1 newly installed, 0 to remove and 70 not upgraded.  
Need to get 174 kB of archives.  
After this operation, 187 kB of additional disk space will be used.  
Get:1 http://deb.debian.org/debian bookworm/main amd64 ssh all 1:9.2p1-2+deb12u5 [174 kB]  
Fetched 174 kB in 0s (689 kB/s)  
Selecting previously unselected package ssh.  
(Reading database ... 153817 files and directories currently installed.)  
Preparing to unpack .../ssh_1%3a9.2p1-2+deb12u5_all.deb ...  
Unpacking ssh (1:9.2p1-2+deb12u5) ...  
Setting up ssh (1:9.2p1-2+deb12u5) ...  
metaphorraccoon@home-server:~$
```

Once we are sure this is installed, we can run connect to our server through the terminal using the *ssh* command. We will be connecting to our user account what is at our server's IP address:

```
ssh username@192.168.68.100
```



```
metaphorraccoon@home-server: ~  
metaphorraccoon@home-server:~$ ssh metaphorraccoon@192.168.68.100
```

You will be alerted that the authenticity of the host server could not be verified. This happens because we have never connected to the server before and it can't ensure the server's fingerprint.

```
metaphorraccoon@home-server: ~  
metaphorraccoon@home-server:~$ ssh metaphorraccoon@192.168.68.100  
The authenticity of host '192.168.68.100 (192.168.68.100)' can't be established.  
ED25519 key fingerprint is SHA256:kWh1o1ifp1FwFvNHiwWW/FueuUk/2BZEPooUHBqLd0g.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

We can type 'yes' then hit 'enter' to accept the newly generated host fingerprint and remember it for future use.

```
metaphorraccoon@home-server: ~  
metaphorraccoon@home-server:~$ ssh metaphorraccoon@192.168.68.100  
The authenticity of host '192.168.68.100 (192.168.68.100)' can't be established.  
ED25519 key fingerprint is SHA256:kWh1o1ifp1FwFvNHiwWW/FueuUk/2BZEPooUHBqLd0g.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '192.168.68.100' (ED25519) to the list of known hosts.  
metaphorraccoon@192.168.68.100's password: █
```

Now, you will be prompted for your password. After hitting 'enter', you will be connected to your server and can run commands.

```
metaphorraccoon@development: ~  
metaphorraccoon@home-server:~$ ssh metaphorraccoon@192.168.68.100  
The authenticity of host '192.168.68.100 (192.168.68.100)' can't be established.  
ED25519 key fingerprint is SHA256:kWh1o1ifp1FwFvNHiWW/FueuUk/2BZEPooUHBQld  
0g.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '192.168.68.100' (ED25519) to the list of known  
hosts.  
metaphorraccoon@192.168.68.100's password:  
Linux development 6.1.0-32-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.129-1 (2  
025-03-06) x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Sun Mar 30 13:27:17 2025 from ::ffff:127.0.0.1  
metaphorraccoon@development:~$ █
```

You can disconnect from the SSH connection with the following command:

```
exit
```

Windows

Windows 10 and Windows 11 come with a client pre-installed so you can use SSH to connect to your server remotely.

You will need to know your username set during the installation process and the IP Address you set during network configuration.

You can open a terminal or PowerShell and run the following command:

```
ssh username@192.168.68.100
```

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

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

PS C:\Users\Josh> ssh metaphorraccoon@192.168.68.100
```

You will be alerted that the authenticity of the host server could not be verified. This happens because we have never connected to the server before and it can't ensure the server's fingerprint.

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

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

PS C:\Users\Josh> ssh metaphorraccoon@192.168.68.100
The authenticity of host '192.168.68.100 (192.168.68.100)' can't be established.
ED25519 key fingerprint is SHA256:kwH1o1ifp1FwFvNHiwWw/FueuUk/2BZEPOoUHBqLd0g.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? █
```

We can type 'yes' then hit 'enter' to accept the newly generated host fingerprint and remember it for future use.

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

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

PS C:\Users\Josh> ssh metaphorraccoon@192.168.68.100
The authenticity of host '192.168.68.100 (192.168.68.100)' can't be established.
ED25519 key fingerprint is SHA256:kWh1o11fp1FwFvNHwWw/FueuUk/2BZEPOoUHBqLd0g.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.68.100' (ED25519) to the list of known hosts.
metaphorraccoon@192.168.68.100's password: █
```

Now, you will be prompted for your password. After hitting 'enter', you will be connected to your server and can run commands.

```
metaphorraccoon@development ~
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\Users\Josh> ssh metaphorraccoon@192.168.68.100
The authenticity of host '192.168.68.100 (192.168.68.100)' can't be established.
ED25519 key fingerprint is SHA256:kWh1o11fp1FwFvNHwWw/FueuUk/2BZEPOoUHBqLd0g.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.68.100' (ED25519) to the list of known hosts.
metaphorraccoon@192.168.68.100's password: █
Linux development 6.1.0-32-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.129-1 (2025-03-06) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

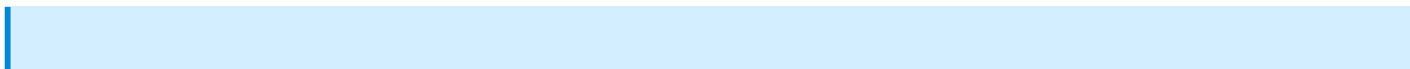
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Apr 10 16:10:59 2025 from 192.168.68.67
metaphorraccoon@development:~$ █
```

You can disconnect from the SSH connection with the following command:

```
exit
```

MacOS

MacOS, and the legacy OS X, come with SSH already installed so you can remotely connect to your server.



You will need to know your [username set during the installation process](#) and the IP Address you set during [network configuration](#).

We can connect to our server through the terminal using the `ssh` command. We will be connecting to our user account what is at our server's IP address:

```
ssh username@192.168.68.100
```

You will be alerted that the authenticity of the host server could not be verified. This happens because we have never connected to the server before and it can't ensure the server's fingerprint.

We can type 'yes' then hit 'enter' to accept the newly generated host fingerprint and remember it for future use.

Now, you will be prompted for your password. After hitting 'enter', you will be connected to your server and can run commands.

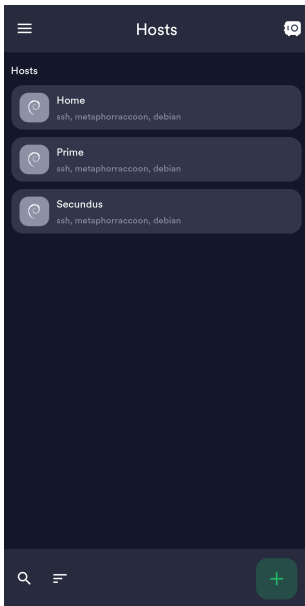
You can disconnect from the SSH connection with the following command:

```
exit
```

Android & iOS

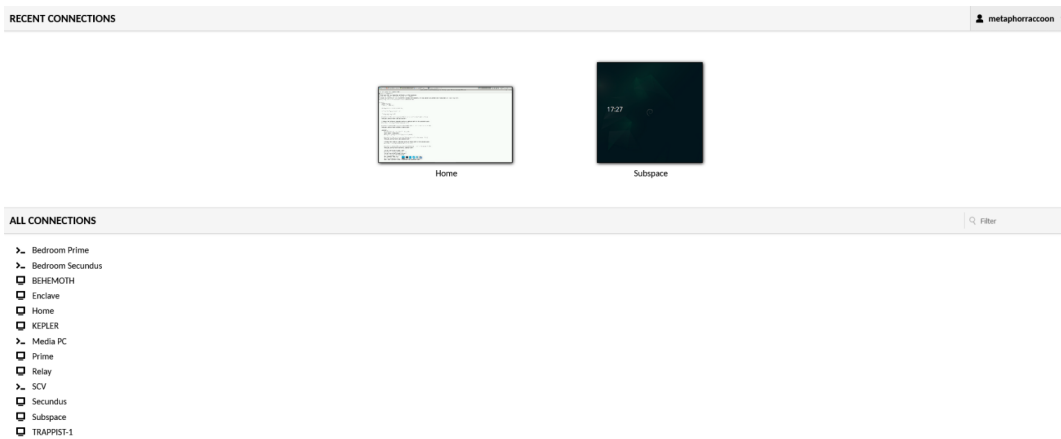
There are apps available through the Google Play Store and Apple App Store that allow you to remotely connect to your server using SSH.

We recommend [Termius](#), a freeware application for Android 7 and iOS 16 or newer. You can create a profile for your server with the credentials pre-saved for quick connection. As a premium paid feature, you can also sync these hosts between the mobile and desktop clients.



Web Access

SSH is a great option for connecting to your server over your local network, but is not available through the world wide web. The easiest way to accomplish this is to use our home server to host [Guacamole](#), a web application that facilitates remote connection to RDP, VNC and SSH through your web browser.



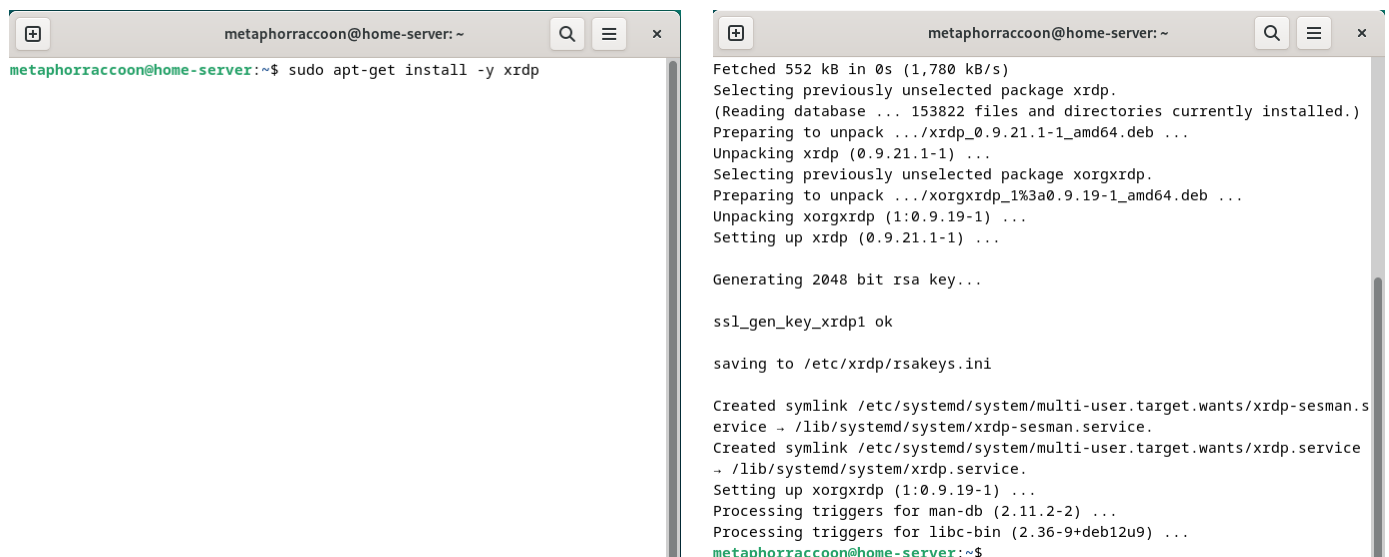
Remote Desktop

We will be installing [xrdp](#), a service that allows us to remotely access our computer over the local network. This is an open-source implementation of Microsoft's [Remote Desktop Protocol](#) and will work with any software that supports RDP.

Installing Remote Desktop

Run this command in a terminal to install it:

```
sudo apt-get install -y xrdp
```



```
metaphorraccoon@home-server: ~  
metaphorraccoon@home-server:~$ sudo apt-get install -y xrdp  
  
Fetched 552 kB in 0s (1,780 kB/s)  
Selecting previously unselected package xrdp.  
(Reading database ... 153822 files and directories currently installed.)  
Preparing to unpack .../xrdp_0.9.21.1-1_amd64.deb ...  
Unpacking xrdp (0.9.21.1-1) ...  
Selecting previously unselected package xorgxrdp.  
Preparing to unpack .../xorgxrdp_1%3a0.9.19-1_amd64.deb ...  
Unpacking xorgxrdp (1:0.9.19-1) ...  
Setting up xrdp (0.9.21.1-1) ...  
  
Generating 2048 bit rsa key...  
  
ssl_gen_key_xrdp1 ok  
  
saving to /etc/xrdp/rsakeys.ini  
  
Created symlink /etc/systemd/system/multi-user.target.wants/xrdp-sesman.s  
ervice -> /lib/systemd/system/xrdp-sesman.service.  
Created symlink /etc/systemd/system/multi-user.target.wants/xrdp.service  
-> /lib/systemd/system/xrdp.service.  
Setting up xorgxrdp (1:0.9.19-1) ...  
Processing triggers for man-db (2.11.2-2) ...  
Processing triggers for libc-bin (2.36-9+deb12u9) ...  
metaphorraccoon@home-server:~$
```

Once installed, we can verify that the service is running with the command:

```
sudo systemctl status xrdp
```

```
metaphorraccoon@home-server: ~
• xrdp.service - xrdp daemon
  Loaded: loaded (/lib/systemd/system/xrdp.service; enabled; preset: enabled)
  Active: active (running) since Thu 2025-04-10 16:24:35 PDT; 1min 0s ago
  Docs: man:xrdp(8)
        man:xrdp.ini(5)
  Process: 25209 ExecStartPre=/bin/sh /usr/share/xrdp/socksetup (code=exited, stat>
  Process: 25217 ExecStart=/usr/sbin/xrdp $XRDPOPTIONS (code=exited, status=0/SUC>
  Main PID: 25218 (xrdp)
  Tasks: 1 (limit: 4599)
  Memory: 904.0K
  CPU: 17ms
  CGroup: /system.slice/xrdp.service
          └─25218 /usr/sbin/xrdp

Apr 10 16:24:34 home-server systemd[1]: Starting xrdp.service - xrdp daemon...
Apr 10 16:24:34 home-server xrdp[25217]: [INFO ] address [0.0.0.0] port [3389] mode 1
Apr 10 16:24:34 home-server xrdp[25217]: [INFO ] listening to port 3389 on 0.0.0.0
Apr 10 16:24:34 home-server xrdp[25217]: [INFO ] xrdp_listen_pp done
Apr 10 16:24:34 home-server systemd[1]: xrdp.service: Can't open PID file /run/xrdp/>
Apr 10 16:24:35 home-server systemd[1]: Started xrdp.service - xrdp daemon.
Apr 10 16:24:36 home-server xrdp[25218]: [INFO ] starting xrdp with pid 25218
Apr 10 16:24:36 home-server xrdp[25218]: [INFO ] address [0.0.0.0] port [3389] mode 1
Apr 10 16:24:36 home-server xrdp[25218]: [INFO ] listening to port 3389 on 0.0.0.0
lines 1-23
```

Accessing Remote Desktop

xrdp is an open-source implementation of the Microsoft Windows RDP protocol that is widely supported by most operating systems. Here, we can explore some of the ways to remotely access your server desktop for maintenance.

You cannot remote desktop into your account while you are logged in locally through the desktop.

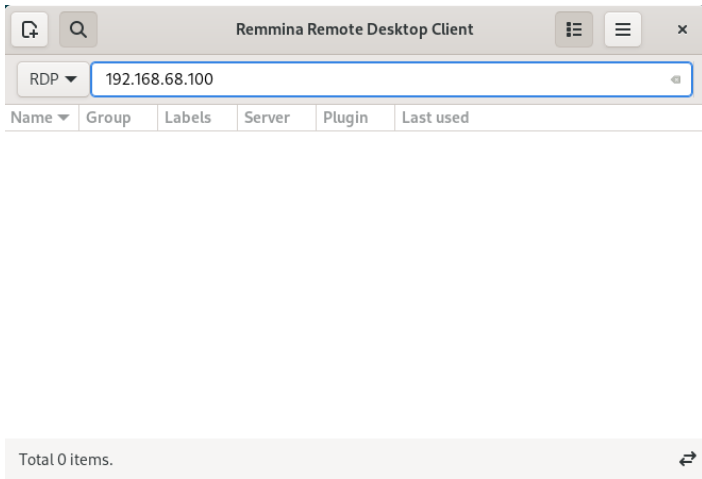
Linux

We will be installing [Remmina](#), an open-source client for accessing computers over the network using protocols like RDP, SSH, and VNC. We will need to open the terminal and run the following commands to install the software.

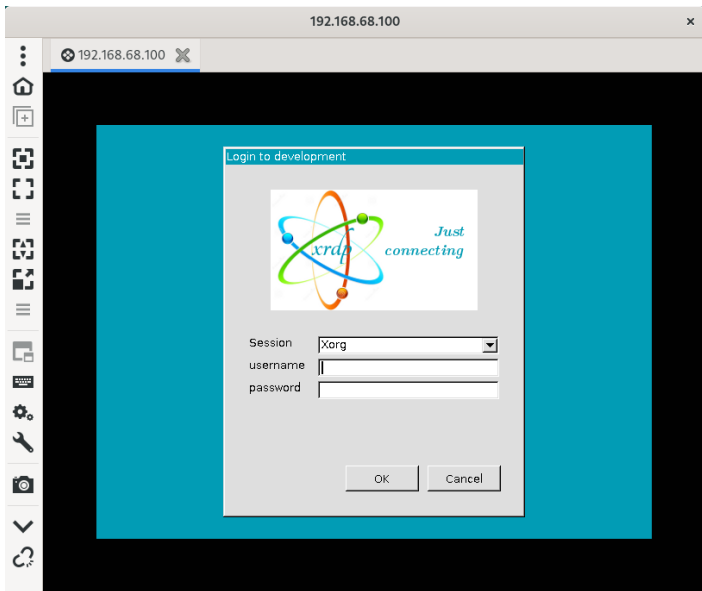
```
sudo apt update
sudo apt-get install -y remmina remmina-plugin-rdp remmina-plugin-vnc
```



Once we have finished installing the software, we can open it from the start menu.

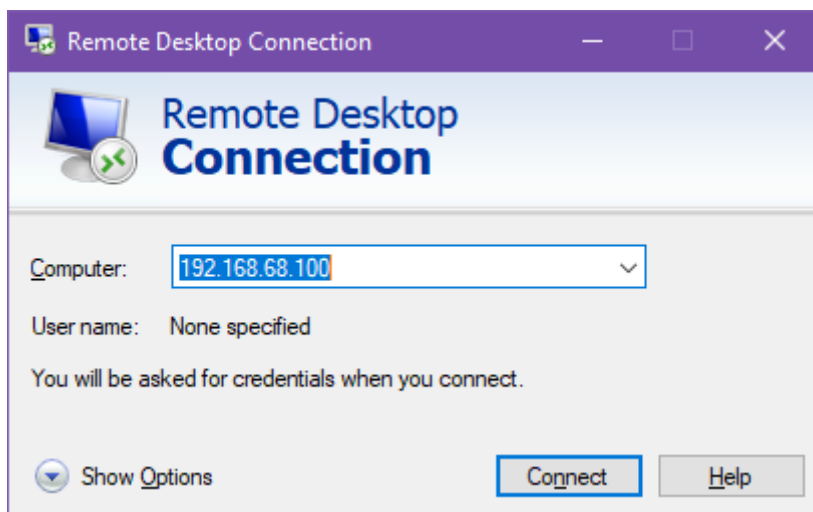


You can connect to your server by entering the IP address into the bar. Make sure to use the RDP protocol.

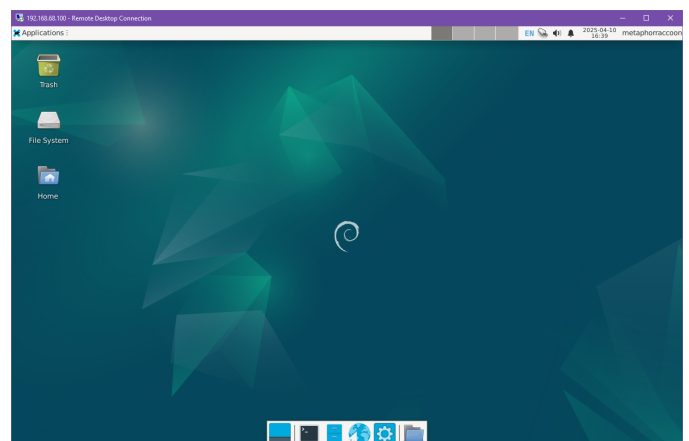
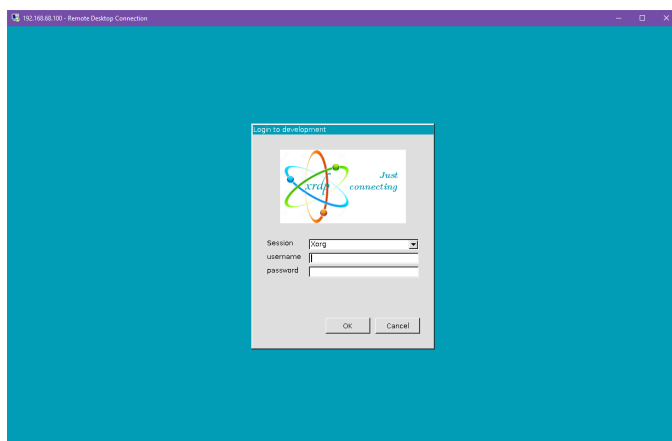


Windows

Most versions of Windows come with the Remote Desktop Connection program installed that can be used to connect to your server. You will need to enter the IP address of your server on the local network.



Once connected, you can enter your username and password to connect to the desktop.



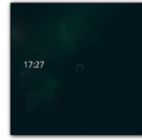
MacOS

Android & iOS

Web Access



Home




Subspace

- > Bedroom Prime
- > Bedroom Secundus
- BEHEMOTH
- Enclave
- Home
- KEPLER
- > Media PC
- Prime
- Relay
- > SCV
- Secundus
- Subspace
- TRAPPIST-1

Clipboard

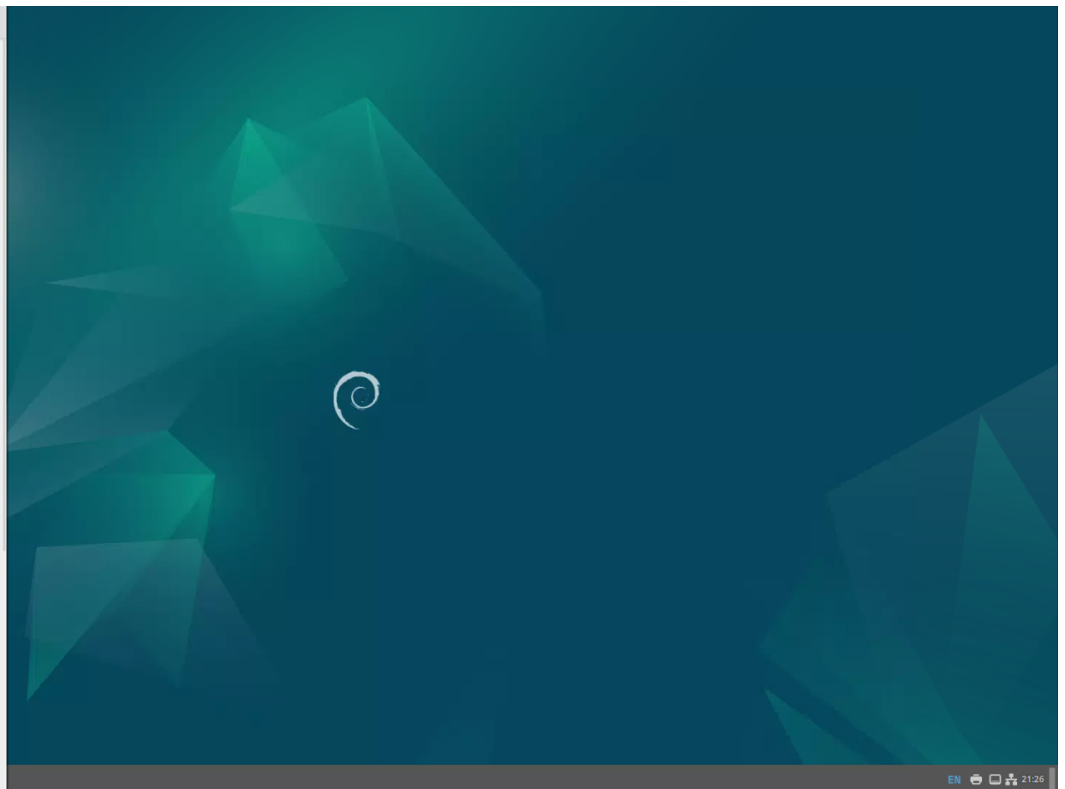
Text copied/cut within Guacamole will appear here. Changes to the text below will affect the remote clipboard.

Input method

- None
No input method is used. Keyboard input is accepted from a connected, physical keyboard.
- Text input
Allow typing of text, and emulate keyboard events based on the typed text. This is necessary for devices such as mobile phones that lack a physical keyboard. 
- On-screen keyboard
Display and accept input from the built-in Guacamole on-screen keyboard. The on-screen keyboard allows typing of key combinations that may otherwise be impossible (such as Ctrl+Alt+Del).

Mouse emulation mode

Determines how the remote mouse behaves with respect to touches.



File Sharing

We will be installing [Samba](#), a protocol that allows us to share your files over the local network. This is open-source implementation of Microsoft's [SMB](#) protocol.

Installation

We can install it by entering the following command:

```
sudo apt-get install -y samba samba-common-bin smbclient
```

Now that Samba is installed, we can ensure it's running by using the following command:

```
sudo systemctl status samba
```

Now that we know it's installed and running, we can set up our storage drives for sharing.

Setting Up Shares

Before we make any changes to the Samba configuration, we should back up the default. We can do this by copying the file to a backup:

```
sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.backup
```

Now, we will use a terminal-based text editor known as *nano*. We will edit the Samba configuration file that was just backed up:

```
sudo nano /etc/samba/smb.conf
```

Samba comes with default sharing options, but we are going to modifying the configuration file to include the hard drives we mounted earlier. Using the arrows keys, navigate to the very bottom of the file.

For our Storage drive, we will be sharing it across our local network so anyone with the password can access the files on it.

```
[Storage]
  path = /mnt/storage
  writable = yes
  guest ok = no
  valid users = @sambashare
```

Once we've made our edits, we can hit Ctrl-O to save, then enter to confirm the file name, and finally Ctrl-X to close the nano editor.

Create a Samba User

Next, we will provide our user account with access to the Samba share we just made.

Change 'username' to your account's username.

```
sudo adduser username sambashare
```

Next, we will need to set the password we'll use to access our files.

Change 'username' to your account's username.

```
sudo smbpasswd -a username
```

You will be prompted to enter and confirm your password. If you wish, this can be the same as your account password.

Once that is completed, we can restart the Samba service using the following command:

```
sudo systemctl restart smb
```

Now, we can verify that our Samba shares are working by verifying the output of the following command:

```
smbclient -L localhost -U %
```

This program lists all available Samba shares on the local computer.