

# Remotely - Remote Desktop Support

An open source, self hosted remote desktop support tool for Window, Linux, and some MacOS connections.

- Bare Metal (on host) Install
  - Bare Metal Installation
- Docker Install
  - Docker Install of Remotely
- Installing the Client
  - Installing the Remotely Client on Linux
  - Installing the Client on Windows Machines

# Bare Metal (on host) Install

Installation of Remotely on a Host Server (non-docker).

# Bare Metal Installation

<https://www.youtube.com/embed/fZY8fsq6AJ8>

## Introduction

Remotely is an amazing tool. It's open source, self hosted, and free to use. The developer puts in a ton of time and effort to make this a really nice remote support tool, so if you have a few bucks, and you are saving hundreds, or thousands by utilizing this software, think about donating a bit of cash to him over at GitHub.

<https://github.com/lucent-sea/Remotely>

This application is a great replacement for tools like Team Viewer, AnyDesk, Go To Assist, Bomgar, and so many more. If you want to support your family, or friends, or even business clients, in a safe way, with open source software that doesn't rely on third party controlled servers, then this is very likely the software you've been looking for.

## Server Install on "Bare Metal"

I put "bare metal" in quotes, because I actually use a Digital Ocean VPS for it, but the process is the same regardless.

First, figure out where you want to run this server, and make sure you've setup a Ubuntu 18.04 or later, or a Windows Server (I would guess 2016 or later - but I'm not a Windows user, so no real sure on that one).

Next, jump over to the GitHub page (linked above), and go to the Releases section. Find the latest release (usually at the top of that page), and expand it. You'll find downloadable binaries and installer scripts there. Grab the installer script for your distro (OS) of choice, and get ready to run it.

We'll be doing this on Linux, but for Windows I think it's an executable or a power shell script.

On Linux, just open your terminal (shell) application on your server, and type

```
wget <link to the script you want to download>
```

For instance I ran

```
wget https://github.com/lucent-sea/Remotely/releases/download/v2021.05.19.1433/Remotely_Server_Installer
```

Let it download, then run it using the command

```
sudo ./Remotely_Server_Installer
```

Now just answer the questions as they are presented.

You'll want to make sure to answer "Yes" to the first question unless you have a Github account setup, know how to make personal access tokens, have forked the repo, and can setup GitHub Actions.

Next, provide the folder to host the remotely server from. I used

```
/var/www/remotely/
```

Next, provide the Fully Qualified Domain Name you want to use for your server. This needs to be a domain or sub-domain that you own.

I used `https://remote.osia.me` because I own the domain `osia.me`.

Finally, answer the questions for LetsEncrypt and pick the server type you want to run. I prefer NGinX, so that's what I selected.

Once the script completes with no errors, you should be able to access your new Remotely server at the domain you entered during install.

Navigate to that page in your browser (Firefox, Chrome, Edge, etc), and create your first / admin account.

Check out the video for more settings and UI features.

# Docker Install

Installation of Remotely with Docker

# Docker Install of Remotely

## Install Remotely using Docker and NGinX Proxy Manager

First you'll need to have Docker-CE installed, on your desired host. If you want a fairly easy / safe way to also access your remotely server from outside your local network, you'll want to install and setup NGinX Proxy Manager in Docker as well.

See the following articles / Videos for instructions on setting thoe up.

## Installing Docker-CE and Docker-Compose

You can easily install Docker-CE, Docker-Compose, Portainer-CE, and NGinX Proxy manager by using this quick install script I created and maintain on Github. Just use the command:

```
wget https://gitlab.com/bmcgonag/docker_installs/-/raw/main/install_docker_nproxyman.sh
```

To download the script to your desired host.

Change the permissions to make the script executable:

```
chmod +x ./install_docker_nproxyman.sh
```

and then run the script with the command:

```
./install_docker_nproxyman.sh
```

When run, the script will prompt you to select your host operating system, then will ask you which bits of software you want to install.

Simply enter 'y' for each thing you want to install.

At some point, you may be asked for your super user (sudo) password as well.

Allow the script to complete installation.

At this point, you might want to log out and back in, as this will allow you to use the `docker` and `docker-compose` commands without the need of `sudo` in front of them.

Once you've setup Docker and NGinX Proxy Manager (known as NPM from here on), you're ready to install Remotely via Docker.

Again in a terminal window enter the following command:

```
docker run -d --name remotely --restart unless-stopped -p 5000:5000 -v /remotely-data:/remotely-data  
translucency/remotely:latest
```

In the above command, you'll be running the container as a daemon (-d) so it doesn't stop running when you exit the command line interface (terminal / shell). You're telling the container to restart automatically unless you specifically stop it with the `docker stop remotely` command, and you are mapping port 5000 on your host machine to the port 5000 on the docker container. Finally, you're setting up a volume mapping on the host from `/remotely-data` to the container folder `/remotely-data`. This mapping will allow you to keep backups of your remotely information by simply copying the folder to an external drive or NAS as desired.

Once you have the command entered, press the Return / Enter key and let the server start up. As long as you don't get any errors in the terminal, you should be able to access your install at the IP address of your host machine on port 5000. If you have port 5000 already in use on the host, you can change the port on the left side of the colon ":" in the command to any free port on your machine. I believe I used 8282.

So I then went to <http://192.168.7.51:8282> in my browser to make sure the UI came up.

## Proxy the local address to a domain

I want to access my install, and make it accessible to clients through the internet, so I need a way a domain name ([remotely.routemehome.org](http://remotely.routemehome.org)) to route to my home, and then to my docker container.

For this we use NPM. Presuming you already have it setup and accessible, log into NPM, and create a new Host entry. You need to make sure the domain you'll use points to your home IP. For this you can register a domain and create a wildcard A record (\*) to point to your home's public IP, or you can use a service like DuckDNS to get a [duckdns.org](http://duckdns.org) domain and have it update your IP as it changes over time. Other options also exist like CloudFlare's dynamic DNS options, but choose whichever options works best for you.

Here's a video on using DuckDNS for getting a URL that will get you back to your homelab setup.

<https://www.youtube.com/embed/Dm5MyuUdq2s?feature=oembed>

Now, in the host entry, enter the domain or subdomain that you'll use for your Remotely install. I chose remotely.routemehome.org.

Next, enter the IP address of the docker-gateway for your Remotely install. If the Remotely container and the NPM containers are on different host machines, then use the IP address of the host that the Remotely container is running on.

Next, enter the port of the "host". 5000 if you didn't change it.

Enable Websockets, and Block common exploits, then save.

Click on the URL in the list, and make sure you get the Remotely Registration page. Once you've got that set, we just need to edit our entry, and move to the SSL tab.

On the SSL tab, select to "Request a New Certificate" from the drop down, enable Force SSL, enter your email address, and agree to the LetsEncrypt terms of service.

Click 'Save'. As long as no errors are shown, you should now be able to open your Remotely site, and be automatically sent to the SSL page.

Register your first user / admin account, and get to supporting people.



# Installing the Client

Install the Remotely client on client machines.

# Installing the Remotely Client on Linux

## Install the client on Linux

Your clients, family, friends, or anyone you are doing remote support for, can go to your remotely server URL, then click on the Downloads tab. Here they'll see a list of available downloadable clients for the various operating systems and architectures Remotely can support.

## Dependencies Needed.

You'll need a few files installed in order for the Remotely Client to function properly on Linux (only tested on Ubuntu 18.04 and later).

```
sudo apt update
```

```
sudo apt install libx11-dev libxrandr-dev libc6-dev libgdiplus libxtst-dev xclip
```

## Make Remotely Client Executable

```
chmod +x Remotely_Desktop
```

## Run the Remotely Client

```
sudo ./Remotely_Desktop
```

# Installing the Client on Windows Machines

## Client Software Download

Your clients, family, friends, or anyone you are doing remote support for, can go to your remotely server URL, then click on the Downloads tab. Here they'll see a list of available downloadable clients for the various operating systems and architectures Remotely can support.

Have them download the correct one for their system (for Windows this is just an .exe file).

Note: the .exe will generate a warning in Windows 10 saying "We Protected Your Computer". There should be a link near the bottom of this message. Have the user click the link, and select "Install Anyway". The "Install Anyway" option will not appear until they click the link for more options.

## First Run of the Client

The user may be asked to enter the server address the first time they run the client. Just have them type in your Remotely Server URL. They'll then receive a code from your server. They give you the code, and you enter it to connect to their machine.

You can build remotely yourself using GitHub pipelines to have your Remotely URL predefined if you wish.