Preparing Your VPS

This page makes the following assumptions:

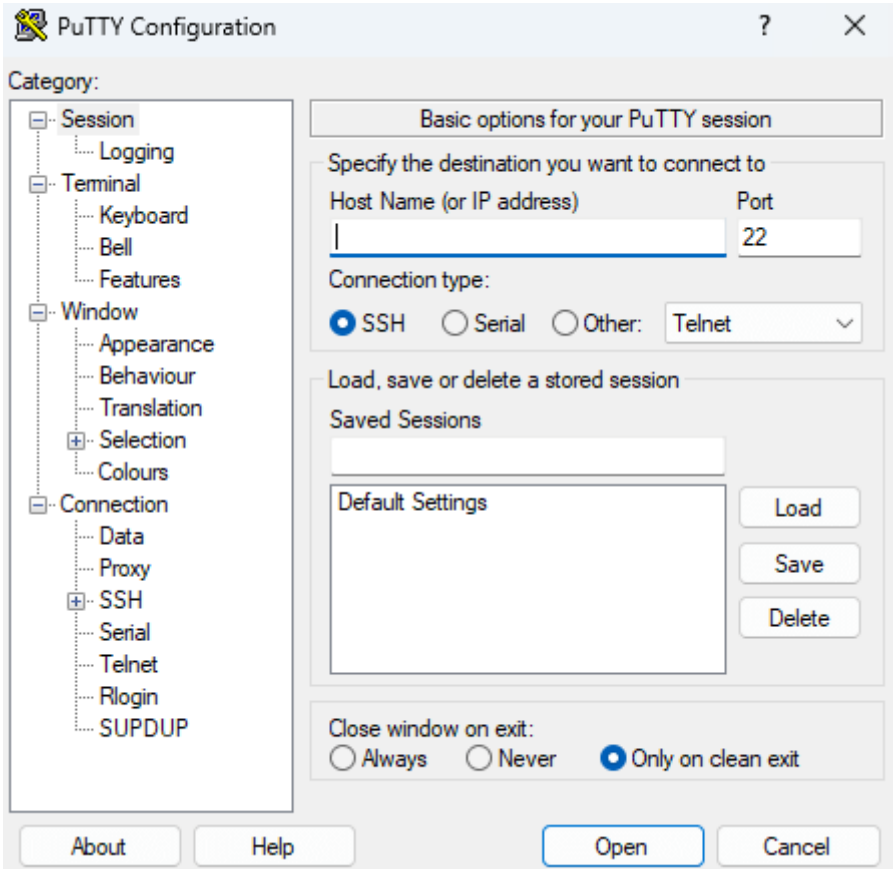
- You have a running VPS.
- You have a domain name that's pointed to your VPS.

If this is not the case, go through the first two pages, Introduction and Planning a Node and Setting Up your Domain Name, then come back here.

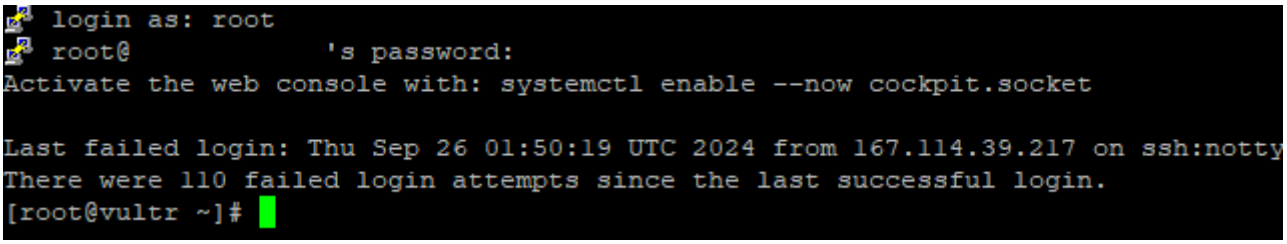
Logging Into Your VPS

Your host should have given you a username and password - the user is likely root. If the user is not root, it's likely it has "sudo" permissions - meaning you can use it to become root once logged in (and you can disregard the "Creating a User Account" steps later in this guide.) You likely also have a console you can log in through on your host - and this is useful, especially if you goof something up and can't log in any other way - but it's more ideal to be able to log in through something called SSH. This basically gives you a command prompt directly into your server, and, among other things, it's likely much easier to copy-paste info to and from it.

Assuming you're running Windows, one program you can use for logging into SSH is called PuTTY. You'll want to download the MSI "Windows Installer" - likely 64-bit x86, or 32-bit x86 if your PC is *really* old. All the defaults should be fine. Once you start up PuTTY, you'll be presented with a window like this:



You'll put your VPS's IP address under "Host Name" and click Open, then press Accept. It'll then ask you for your username and password. Type in the details that your hosting provider gave you. Note that it won't show you typing any characters when entering your password - don't panic, that's by design. Once you do that, you should be in!



Before you continue, change the root password real quick by running this command:

```
passwd
```

It will ask you for a password. Type in whichever password you'd like, hit enter, and then type it in again. It should say "**passwd: all authentication tokens updated successfully.**" If not, try again.

Let's also make sure nano is installed. This is a console text editor that's probably one of the friendlier options out there. (If you know you'd prefer vim/emacs, you can go ahead and use that instead!) Run this command:

```
yum install -y nano
```

It should either install nano, or basically say "is already installed, nothing to do." Either way, you're good!

Securing Your VPS

At this point, you only have a root account with a password. While this *technically* works, this is not a good idea. The root account is one that has ultimate power over the server, and it's not one you typically want to be able to log directly into; instead, it's best to have a user account that you can then promote yourself to root through.

Creating a New User with Sudo Permissions

We're going to disable logging into the system as root over the internet. Instead, you're going to log in as a new user you're about to create, that can then be promoted to root using a command called "sudo". Run the following commands, replacing USERNAME_HERE with your desired username:

```
adduser USERNAME_HERE
passwd USERNAME_HERE
```

After running the passwd command, it will once again ask you for a password, this time for your new user. Type in whichever password you'd like, hit enter, and then type it in again. It should say "**passwd: all authentication tokens updated successfully.**" If not, try again.

Next, run this command to add them to the wheel group, which is a special group that can be "promoted" to root using sudo:

```
usermod -aG wheel USERNAME_HERE
```

> This works on Rocky Linux 9, and may also work on other Red Hat-based distros. If this doesn't work, try replacing "wheel" with "sudo". If that doesn't work either, you can simply give your specific user sudo permissions by running the command **EDITOR=nano visudo**, then adding the following line:

```
USERNAME_HERE ALL=(ALL:ALL) ALL
```

At this point, try logging out and logging back in as your new user. Then, run the following command to promote yourself to sudo:

```
sudo su -
```

It'll ask for your password - type it in. It should take you to a command prompt as root. If it says something along the lines of "This user is not in the sudoers file. This incident will be reported", then you'll need to log back in as root and check the sudoers file. If you need help with this, there are folks in the Discord!

If it did work, though, it's time to lock out remote root login.

Disabling Root Login

> Before proceeding, make absolutely certain you can both log in as your new user **and** you can run **sudo su** - successfully. If you proceed, and you find you are unable to do either of these things, you will be locked out of SSH access to root altogether, and will need to use your webhost's console to re-enable root login.

Run the following command:

```
sudo nano /etc/ssh/sshd_config
```

In this file, look for a line that says "PermitRootLogin yes". You will want to change this line to instead say

```
PermitRootLogin no
```

> To use Nano, you can move the text cursor with the arrow keys, and type as you normally would. When you're finished, press Ctrl+X. If you've made any changes, you'll be asked whether you want to save them or not - hit Y for Yes, N for No, then Enter to write the changes to the file you were writing (or change the name if you wish to write it to a new/different file.)

Save the file, and then restart SSH:

```
sudo systemctl restart sshd
```

From now on, you will only log into your server with the username and password you gave it. Trying to log in with the root credentials over SSH will not work. They will still work when logging in via your webhost's console, however, so keep your root password somewhere safe just in case you need it.

> While not included here, you may also want to consider setting up login using a private key. This is more secure than a password. Steps to do this will be included at a later date, but it should be something you can find online or ask for help with.

Setting Up a Firewall

We're now going to set up the firewall so it allows you to log in to SSH and allows it to act as a webserver, but doesn't allow anyone else to connect to it otherwise.

> These instructions should work on Rocky Linux 9 and any other recent RHEL-based Linux distro. If you're using a different distro, instructions to set up the firewall may be different.

First, make sure firewalld is installed and running:

```
sudo yum install -y firewalld
sudo systemctl enable --now firewalld
```

Then, check to see what your default zone is, then list all the rules in your current zone:

```
sudo firewall-cmd --get-default-zone
```

```
# Replace public in all following commands with whatever zone your default is
sudo firewall-cmd --zone=public --list-all
```

If ssh isn't in services, add it:

```
sudo firewall-cmd --zone=public --add-service=ssh
```

Then, add HTTP and HTTPS:

```
sudo firewall-cmd --zone=public --add-service=http
sudo firewall-cmd --zone=public --add-service=https
```

Finally, run this command to make this configuration permanent:

```
sudo firewall-cmd --runtime-to-permanent
```

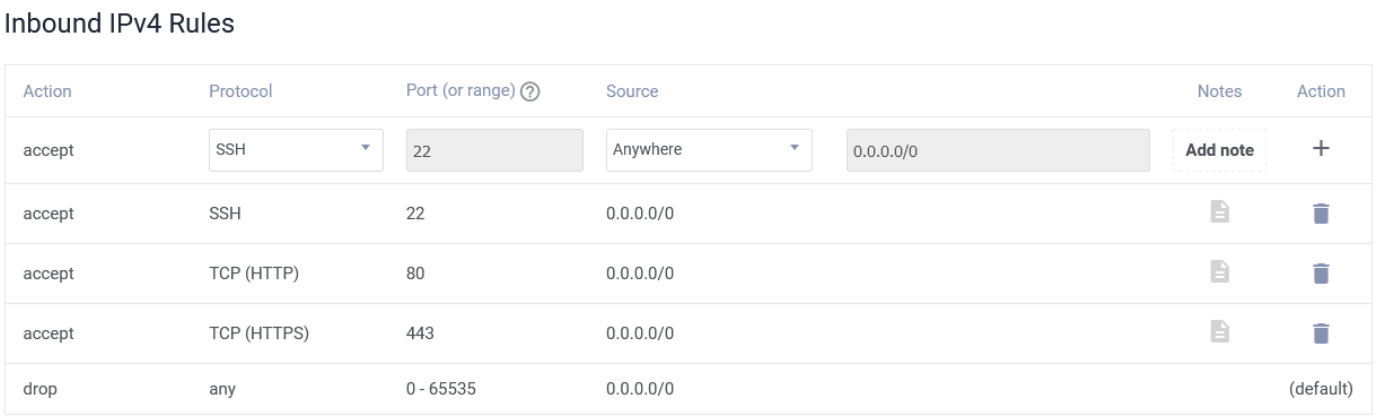
> Hopefully, running these commands did not result in you being disconnected from the server. In the event this happens, however, don't panic - use your webhost's console to log in and add SSH to the currently running firewall config.

Bonus - Setting Up Your Webhost's Firewall

Some (not all) webhosts offer a configurable firewall in front of your VPS. If you are using one such webhost, you can configure it to only permit SSH and HTTP(S) as well. The benefit of this is that it blocks this traffic from even getting to your VPS in the first place. This guide will go over setting up

such a firewall on Vultr.

In Vultr, go to Network > Firewall, then click "Add Firewall Group". Give it a name, and then under Inbound IPv4 Rules, add a few rules so that the list looks like this:



> If you want to be extra secure, you can set up the SSH line to only accept your IP address - this will stop the waves of attempts from bots aimlessly trying to log in with default credentials. You'll need to keep it updated with your IP, though, or you'll lose access!

Do the same under IPv6 Rules. Finally, under Linked Instances, add your current VPS.

Update

Now that we've got those bits out of the way, let's make sure your OS is up to date. Run the following command:

```
sudo yum update -y
```

It should automatically download and update all your OS packages. You may want to restart afterwards, just to make sure all the updates are in effect:

```
sudo shutdown -r now
```

Relatively easy!

> These are only a few of the things you can do to increase the security of your system - this is only scratching the surface. You can write an entire book with all the ways you can further secure a system - such as installing and configuring something like fail2ban to monitor failed logins, auditing applications to monitor operating system changes, tweaking kernel settings - although, each security improvement also comes with its own caveats and associated difficulties. Consider what makes sense for your system and your risk level.

Once you've got all that out of the way, it's time for the moment we've all been waiting for - installing GoToSocial!

> At this point, if you're planning on installing Akkoma instead, you may want to refer to something such as the Akkoma documentation for continuing. A guide for Akkoma will be made in the future.

Revision #9
Created 26 September 2024 00:40:04 by Tenna Lotor
Updated 28 September 2024 01:13:15 by Tenna Lotor