WeeNAS
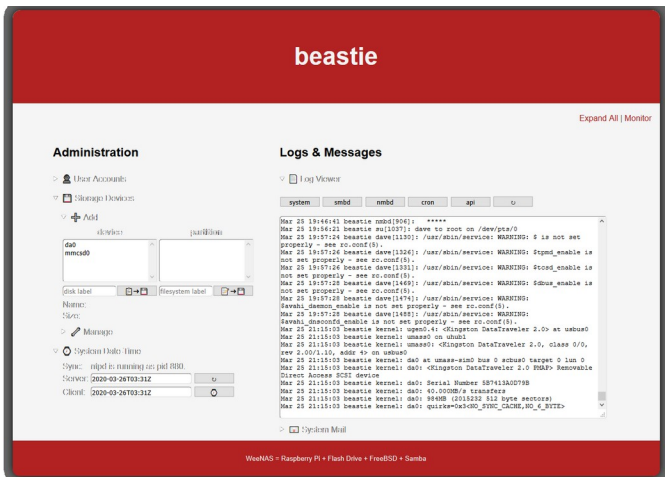

Installation Guide

Everything you need to know to get your
WeeNAS system up and running.

Installing WeeNAS on FreeBSD 12.1

This guide will help you get started with WeeNAS by outlining the procedure for installing FreeBSD, the operating system that is the foundation of WeeNAS.

To be successful, you should be familiar with the Raspberry Pi, know the basics of home network configuration, and also how to use open-source network utilities. Most of the installation and configuration is scripted, but familiarity with using the command-line is required to get the web-based administration system up and running.

If you are an experienced Raspberry Pi tinkerer, you should be fine, but if this is your first RPi project, you may find it easier to start with one of the official Raspberry Pi distributions.

<p style="text-align:center">* * *</p>

What You Will Need

- A PC or laptop with access to the internet and a MicroSD card slot.

- A 32G Class 10 or faster MicroSD card that is compatible with Raspberry Pi.

- A Raspberry Pi 2B with power supply.

- An internet router with a wired connection for the Raspberry Pi.

- 7-Zip software to uncompress the FreeBSD image.

- Win32DiskImager to write the FreeBSD image to the MicroSD card.

- AngryIP Scanner to find your device's DHCP address.

- PuTTY Secure Shell (SSH) client for initial setup.

- Mozilla Firefox for web-based administration. (Firefox is best supported. Other browsers may work depending on their support for HTML5.)

Note:

All of the software listed above is open source licensed and costs nothing to use. However, if you find it useful, donations to these projects help keep them going.

Download FreeBSD

Use an FTP client to visit: ftp.freebsd.org

Or use a browser and go the HTTP equivalent:
http://ftp.freebsd.org/

Browse to the directory for the 12.1 ISO images:
/pub/FreeBSD/releases/ISO-IMAGES/12.1

Find the .img.xz for your model of Raspberry Pi.

Remote site:	/pub/FreeBSD/releases/ISO-IMAGES/12.1

Filename
- FreeBSD-12.1-RELEASE-arm-armv7-PANDABOARD.img.xz
- FreeBSD-12.1-RELEASE-arm-armv7-RPI2.img.xz
- FreeBSD-12.1-RELEASE-arm-armv7-WANDBOARD.img.xz
- FreeBSD-12.1-RELEASE-arm64-aarch64-memstick.img
- FreeBSD-12.1-RELEASE-arm64-aarch64-memstick.img.xz
- FreeBSD-12.1-RELEASE-arm64-aarch64-mini-memstick.img
- FreeBSD-12.1-RELEASE-arm64-aarch64-mini-memstick.img.xz
- FreeBSD-12.1-RELEASE-arm64-aarch64-PINE64-LTS.img.xz
- FreeBSD-12.1-RELEASE-arm64-aarch64-PINE64.img.xz
- FreeBSD-12.1-RELEASE-arm64-aarch64-RPI3.img.xz

This guide was written using the older Raspberry Pi 2B. Your experience may be different if you have a later revision board.

Write the Image to the MicroSD Card

First, uncompress the .xz image with 7-Zip by right-clicking the file and using the Extract Here option from the context menu.
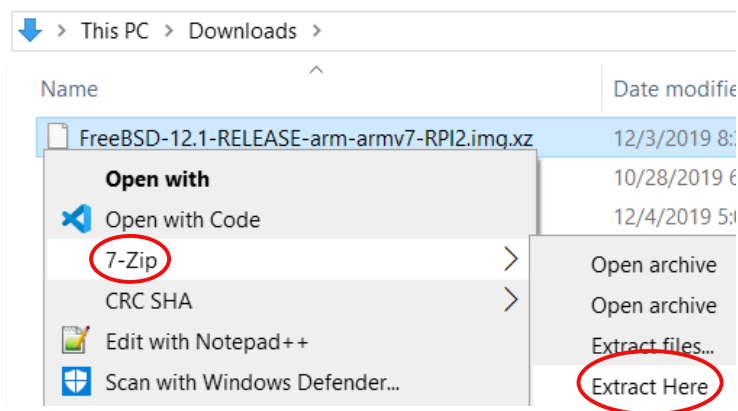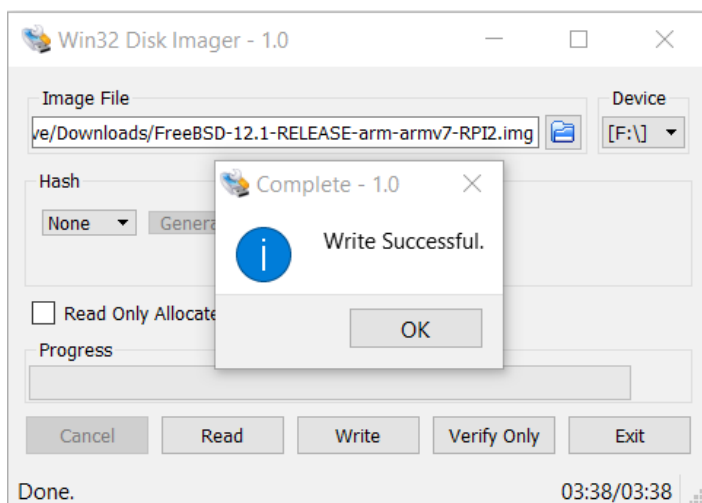
Next, use Win32DiskImager to copy the image onto the MicroSD card.
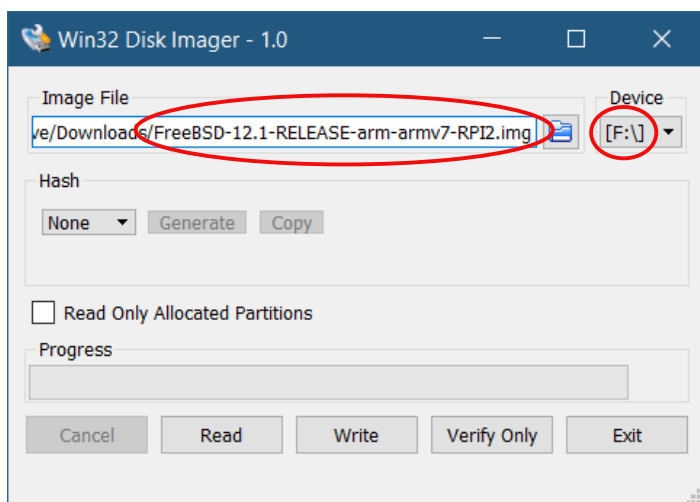
Verify that the correct .img file is selected (not the .xz file.) Also, verify that the drive letter of the MicroSD card is correct.
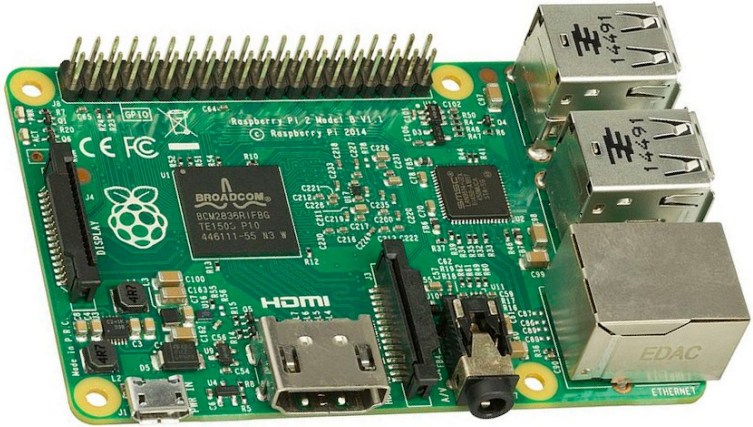
If everything looks good, click the Write button to begin. The process takes about four minutes.

When complete, eject the media and remove the card.

Booting FreeBSD

Insert the MicroSD card into the Raspberry Pi (left side of picture below, on the underside of the board.)

Next, attach a CAT5 Ethernet cable between the Raspberry Pi's RJ-45 jack (lower right side of the picture) to a port on your internet router.

No USB devices should be plugged in at this time.

Finally, plug in the power cable (lower, left corner) and power up the Raspberry Pi.

The initial boot process takes some time and with no monitor attached, it's difficult to see how things are progressing. You can watch the LEDs on the Raspberry Pi to get a rough idea of the status.

After a short bit of time, the red LED on the MicroSD socket side of the board will turn off. This means

FreeBSD has started booting. Normally, this takes a minute at most. If the LED stays on longer than that, there may be a problem with the image written to the MicroSD card or the card itself.

Next in the boot process, the network link light and traffic indicator LEDs will illuminate, turn off, and come back on. This means the network subsystem is starting and is a good indication that FreeBSD is nearly ready for you to log in. Wait for the link light to remain on for a while before proceeding.

Finding Your IP Address

Before you can log into FreeBSD on the Raspberry Pi, you have to know the IP address. If your internet router shows a table of connected devices, look for it there under the name of 'generic'. Otherwise, you can use Angry IP Scanner to find it.

Under the Angry IP Scanner menu, Tools > Fetchers, you can configure the columns of information that will be shown about each device. Add MAC Vendor to the default list.
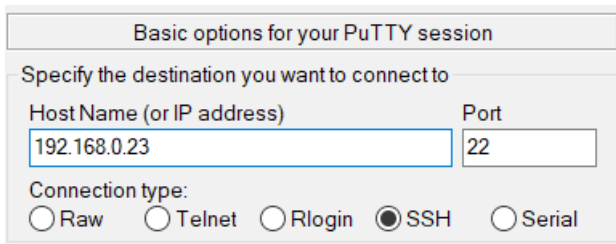
Run a scan of your network and look for "Raspberry Pi Foundation" in the MAC Vendor column. Note the IP address found in the output.

🟢 192.168.0.23	11 ms	[n/a]	22	Raspberry Pi Foundation

Logging in via SecureSHell (SSH)

Open up PuTTY and enter the IP address you found into the field labeled 'Host Name (or IP address)'.
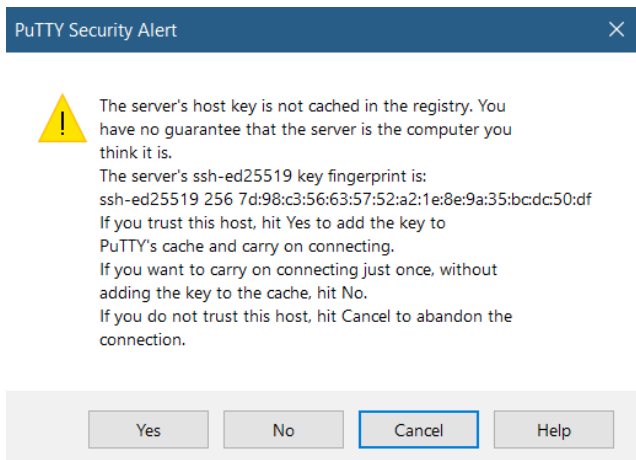
Click the Open button.

Since this is the first login to this device, you'll get a security alert. It's okay to say yes.
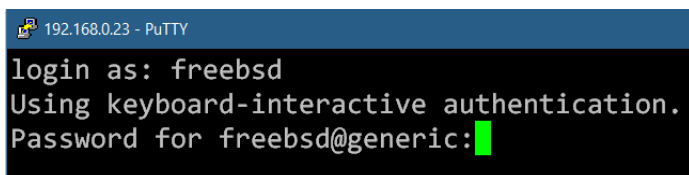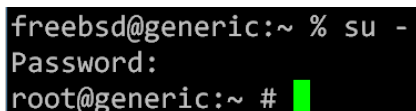
After that, a login prompt will appear. Log in with the default username/password combination of freebsd/freebsd.


```
192.168.0.23 - PuTTY
login as: freebsd
Using keyboard-interactive authentication.
Password for freebsd@generic:
```

You'll be treated to some welcome messages and be left at a command prompt. Type 'su -' and enter 'root' when prompted for a password.


```
freebsd@generic:~ % su -
Password:
root@generic:~ #
```

This will be the procedure to follow any time you need gain superuser access via SSH (though the passwords will be different after configuration.)

Downloading WeeNAS

In this guide, the WeeNAS package will be installed into /root/. It can be installed in other locations, but this is what the guide will use.

Change directory to /root and download the latest version of WeeNAS using the 'fetch' program and the following link: https://github.com/DavesCodeMusings/WeeNAS/archive/master.zip

fetch --no-verify-peer [github link]

The --no-verify-peer option avoids "Certificate verification failed" errors that crop up.

After downloading, unzip it with 'unzip master.zip'.

```
root@generic:~ # fetch --no-verify-peer https://github.com/DavesCodeMusings/WeeN
AS/archive/master.zip
fetch: https://github.com/DavesCodeMusings/WeeNAS/archive/master.zip: size of re
mote file is not known
master.zip                                      2243 kB 2952 kBps    00s
root@generic:~ # unzip master.zip
Archive:  master.zip
   creating: WeeNAS-master/
```

The directory name WeeNAS-master comes from the GitHub platform. It must be renamed to weenas before use. The command 'mv WeeNAS-master weenas' will change the directory name to weenas.

```
root@generic:~ # mv WeeNAS-master weenas
```

The remainder of this guide will refer to /root/weenas as the installation location.

Installing WeeNAS

WeeNAS comes bundled with a script called install.sh. This script will detect and configure the USB storage device used for home drives as well as install the Samba and Node.js packages it requires.

First, plug in the USB storage device you want to use for home drives. No other USB flash drives should be plugged in at this time.

Then, run the script from the /root/weenas directory, like this:

cd /root/weenas
sh ./install.sh

Your USB flash drive will be detected and information displayed to help you identify it. You will be asked to confirm overwriting the USB flash drive.

```
root@generic:~/weenas # sh ./install.sh

The following USB mass storage device was detected:
   Name: da0
   Mediasize: 1031798784 (984M)
   descr: Kingston DataTraveler 2.0

   There is an existing FreeBSD partition on this device.

Do you wish to format this device and use it for WeeNAS home drives?
ALL DATA CURRENTLY ON THE DEVICE WILL BE LOST [y/N]?
```
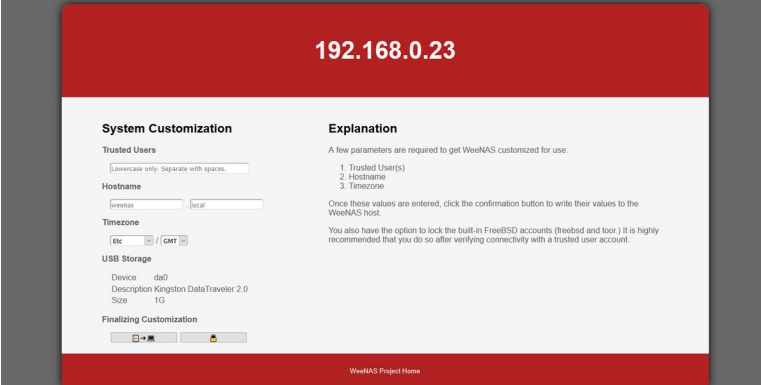
If you say yes, the USB flash drive will be formatted exclusively for use by WeeNAS. All data on the device will be lost and it will no longer be recognizable to the Windows operating system without re-formatting.

The install script is automatic from here on out. Depending on your internet speed, it could take several minutes to download and install the necessary packages.

In the end, you will see a message directing you to open a web browser to customize the system. When you open the browser, it'll look like this:

Notice that the USB storage device has been detected and displayed for you. This is where users' home directories will be stored. Make sure it is the correct device.

Your final tasks are to fill in the fields for Trusted Users, Hostname, and Timezone.

Trusted Users are the accounts that can log in via SSH and su to root. There needs to be at least one trusted user account, but there can be more. Separate multiple user accounts with spaces.

The hostname is divided into two fields, host and domain. The host portion can be any name you want, but is limited to fifteen characters or less for NetBIOS compatibility. If you have a registered domain, enter it in the field after the dot. If you don't, just enter local. Both fields should be limited to lowercase letters.

Setting the timezone is a matter of selecting your area from the left and location from the right. Choose the area first and the available locations will be filled automatically.

Here's an example:

Trusted Users

dave

Hostname

beastie . local
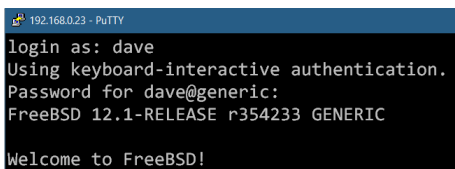
Timezone

America / Chicago

That's all there is to system customization. Review your choices and click the button to write the values to your WeeNAS device.
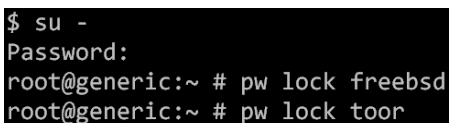
Testing Connectivity (SSH)

Secure Shell is your way to access the underlying FreeBSD operating system. This test is very important, because if it does not succeed, you will have no way to access the system.
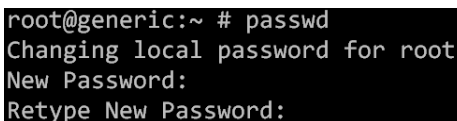
Open another PuTTY window and enter the IP address of your WeeNAS Raspberry Pi. Log in with the trusted user account and the password given to the 'passwd' command.

```
192.168.0.23 - PuTTY
login as: dave
Using keyboard-interactive authentication.
Password for dave@generic:
FreeBSD 12.1-RELEASE r354233 GENERIC

Welcome to FreeBSD!
```

Enter the command 'su -' to switch to the root user. If all is successful, go ahead and lock the built-in accounts of freebsd and toor.

```
$ su -
Password:
root@generic:~ # pw lock freebsd
root@generic:~ # pw lock toor
```

This is also a good time to change the root password to something else. Do that with the 'passwd' command.

```
root@generic:~ # passwd
Changing local password for root
New Password:
Retype New Password:
```

Testing Connectivity (SMB)

SMB is the Windows way of connecting to network shares. This is how you will access the files stored on the WeeNAS system.

Open up Windows Explorer and enter the IP address of the WeeNAS server preceded by a double backslash.
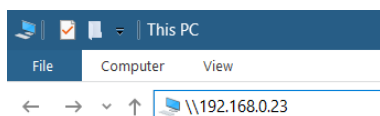
You should be prompted for a username and password. Enter the trusted username and the password given to the 'smbpasswd' command. You do not need to check the box labeled 'Remember my credentials' at this time.
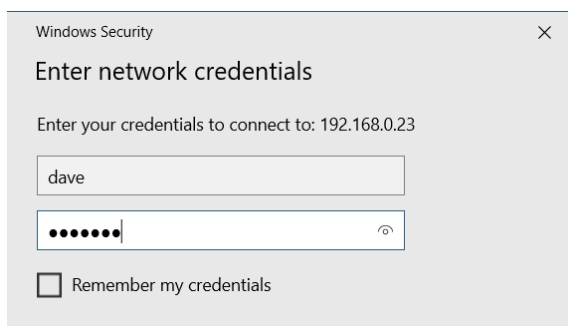
If all goes well, you should see a network folder with your user account name and possibly a shared folder if you elected to include that in the configuration.
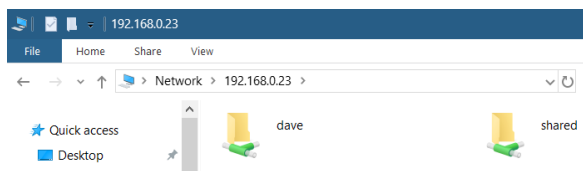
Reboot for Config Changes

Congratulations! You've installed, configured and tested the installation of your WeeNAS system. The last thing to do is reboot. This will ensure that changes to the hostname or network configuration take effect. It's also good way to find out if there are any problems with the system.

But, before you reboot, you may want to configure the WeeNAS API as a service that will start automatically when the system comes up. That's covered in the upcoming section.

When you're ready, you can reboot by logging in as a trusted user, 'su -' to root, and then enter the command 'shutdown -r now'.

Next Steps

Now that the initial installation is complete, regular day to day maintenance can be performed using the WeeNAS web-based admin page. To access it, simply open a web browser to:

http://[weenas host]:9000/admin.html

Replace [weenas host] with the hostname or IP address of your WeeNAS FreeBSD Raspberry Pi.