

## **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.

\* \* \*

The text of this document are copyright (c)2020 David Horton and are released under the Creative Commons Attribution Share-Alike (BY-SA) Version 4.0 license.

<https://creativecommons.org/licenses/by-sa/4.0/>

## **What You Will Need**

- A PC or laptop with access to the internet and a MicroSD card slot.
- A 32G Class 10 or better MicroSD card.
- 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 client for initial setup.
- Mozilla Firefox or Google Chrome for web-based administration. (Firefox is best supported. Internet Explorer and Edge are not supported at all.)

## 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`

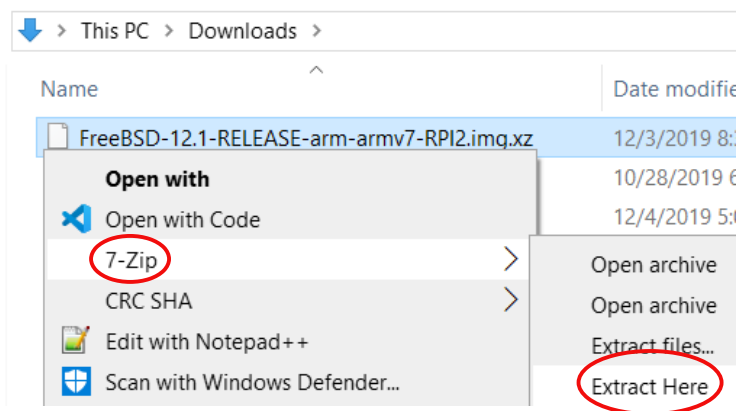
Find the .img.xz for your model of Raspberry Pi.

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

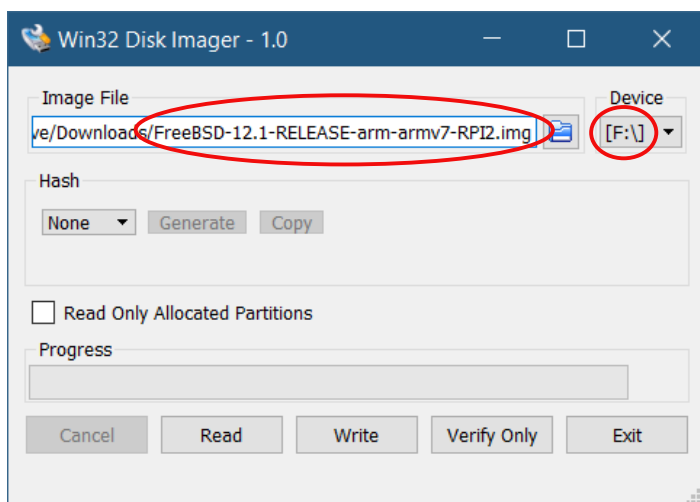
This guide was written using the older Raspberry Pi 2B. Your experience may be different if you have a later revision board.

## 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.

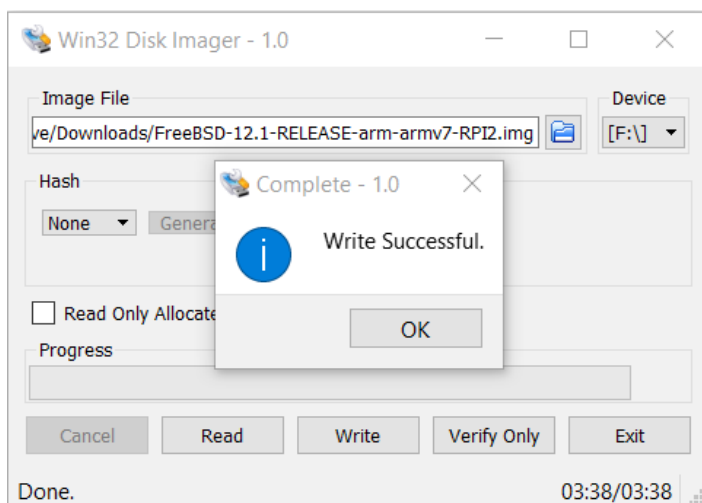


Next, use Win32DiskImager to copy the image onto the MicroSD card.



Verify that the correct .img file is selected (not the .xz file.) Also, verify that the drive letter of the MicroSD card is correct.

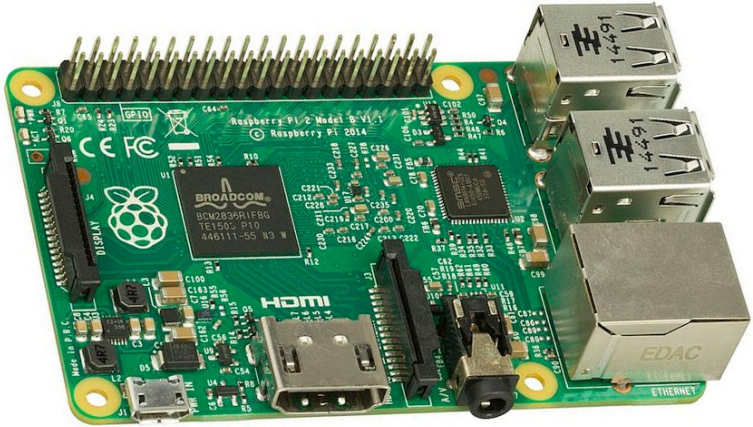
If everything looks good, click the Write button to begin. The process takes about four minutes.



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.

 IP Range - Angry IP Scanner

Scan Go to Commands Favorites Tools Help

IP Range:	192.168.0.0	to	192.168.0.255	IP Range ▾	⚙
Hostname:		IP↑	Netmask ▾	<b>Start</b>	☰
IP	Ping	Hostname	Ports [12+]	MAC Vendor	

**Fetchers** ✕

Here you can select fetchers for scanning. Fetchers are represented by columns.

**Selected fetchers**  
Ping  
Hostname  
Ports  
MAC Vendor

↑  
↓  
←  
→  
⚙

**Available fetchers**  
TTL  
Filtered Ports  
Web detect  
HTTP Sender  
Comments  
NetBIOS Info  
MAC Address  
Packet Loss  
HTTP Proxy

**OK** **Cancel**

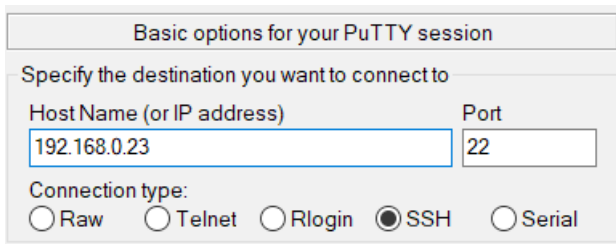


Run a scan of your network and look for “Raspberry Pi Foundation” in the MAC Vendor column. Note the IP address found in the output.

 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)'.



Basic options for your PuTTY session

Specify the destination you want to connect to

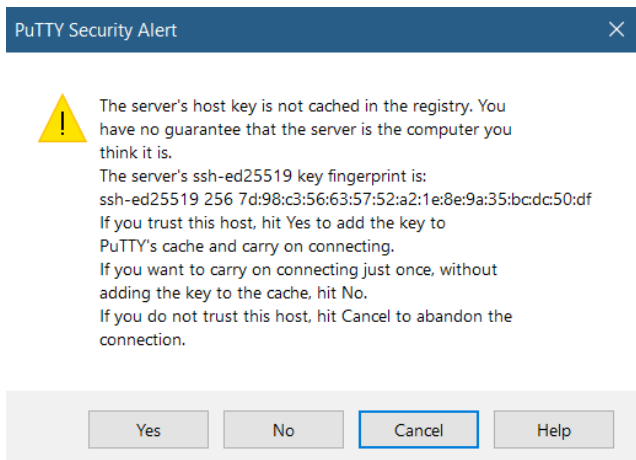
Host Name (or IP address)  Port

Connection type:


☐ Raw ☐ Telnet ☐ Rlogin ☒ SSH ☐ Serial

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.



PuTTY Security Alert

 The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.

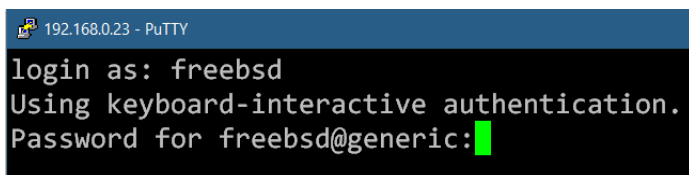
The server's ssh-ed25519 key fingerprint is:  
ssh-ed25519 256 7d:98:c3:56:63:57:52:a2:1e:8e:9a:35:bcdc:50:df

If you trust this host, hit Yes to add the key to PuTTY's cache and carry on connecting.

If you want to carry on connecting just once, without adding the key to the cache, hit No.

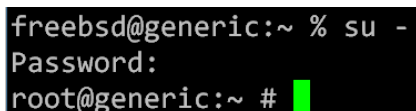
If you do not trust this host, hit Cancel to abandon the connection.

After that, a login prompt will appear. Log in with the default username/password combination of freebsd/freebsd.

A terminal window with a blue title bar containing a small icon and the text '192.168.0.23 - PuTTY'. The terminal text shows a login prompt for 'freebsd', followed by a message about keyboard-interactive authentication, and then a password prompt for 'freebsd@generic:' with a red cursor.

```
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.

A terminal window showing the execution of the 'su -' command. It prompts for a password, and after entering 'root', the prompt changes from '~ #' to '~ #', indicating root access.

```
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 /home/root/. It can be installed in other locations, but this is what the guide will use.

Change directory to /home/root and download the latest version of WeeNAS using the 'fetch' program and the following link:

<https://github.com/DavesCodeMusings/WeeNAS/archive/master.zip>

Note: If you get errors saying "Certificate verification failed," you can use the --no-verify-peer option to get around it.

```
root@generic:~ # fetch --no-verify-peer https://github.com/DavesCodeMusings/WeeNAS/archive/master.zip
```

After downloading, unzip it:

```
root@generic:~ # fetch https://github.com/DavesCodeMusings/WeeNAS/archive/master.zip
fetch: https://github.com/DavesCodeMusings/WeeNAS/archive/master.zip: size of remote file is not known
master.zip                                     55 kB  688 kBps   00s
root@generic:~ # unzip master.zip
```

The directory name WeeNAS-master comes from the GitHub platform. It can be renamed or have a symbolic link pointed to it. For example, '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

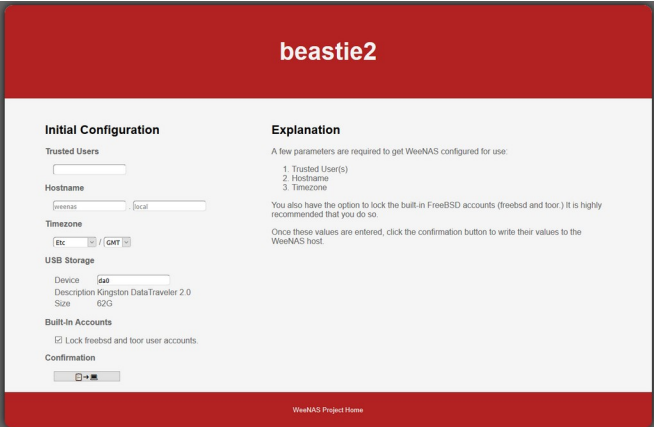
WeeNAS comes bundled with a script called `install.sh` that is used to download and configure the Samba and Node.js packages it needs to run. The script should be run from the `/root/weenas` directory, like this:

```
cd /root/weenas
```

```
sh ./install.sh
```

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 configure the system. When you do, it'll look like this:



The screenshot shows the 'beastie2' web interface for WeeNAS initial configuration. The interface is divided into two main columns: 'Initial Configuration' and 'Explanation'.

**Initial Configuration**

- Trusted Users:** A text input field with the value 'weenas'.
- Hostname:** A text input field with the value 'local'.
- Timezone:** A dropdown menu showing 'ETC' and a button labeled 'GMT'.
- USB Storage:**
  - Device:** A text input field with the value 'sda'.
  - Description:** 'Kingston DataTraveler 2.0'.
  - Size:** '62G'.
- Built-in Accounts:** A checkbox labeled 'Lock freebsd and toor user accounts.' which is checked.
- Confirmation:** A button with a right-pointing arrow.

**Explanation**

A few parameters are required to get WeeNAS configured for use:

1. Trusted User(s)
2. Hostname
3. Timezone

You also have the option to lock the built-in FreeBSD accounts (freebsd and toor.) It is highly recommended that you do so.

Once these values are entered, click the confirmation button to write their values to the WeeNAS host.

At the bottom of the interface, there is a link labeled 'WeeNAS Project Home'.

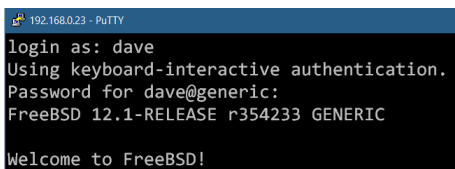
Notice that the USB storage device has been detected and displayed for you.

(More to come.)

## 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.

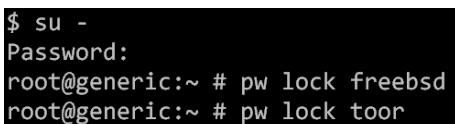
Open another PuTTY window and enter the IP address of your WeeNAS Raspberry Pi. Log in with the trusted user account and the password given to the 'passwd' command.



```
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.

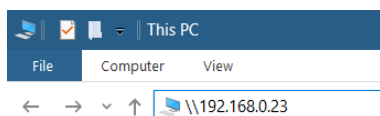


```
root@generic:~ # passwd
Changing local password for root
New Password:
Retype New Password:
```

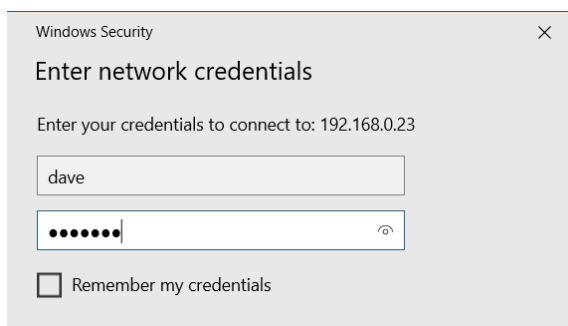
## Testing Connectivity (SMB)

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

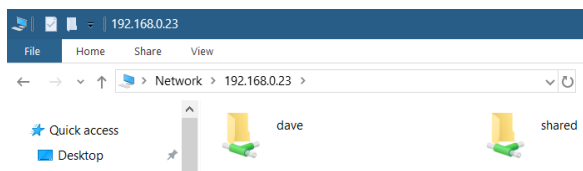
Open up Windows Explorer and enter the IP address of the WeeNAS server preceded by a double backslash.



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 -', 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.