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.

* * *

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.)

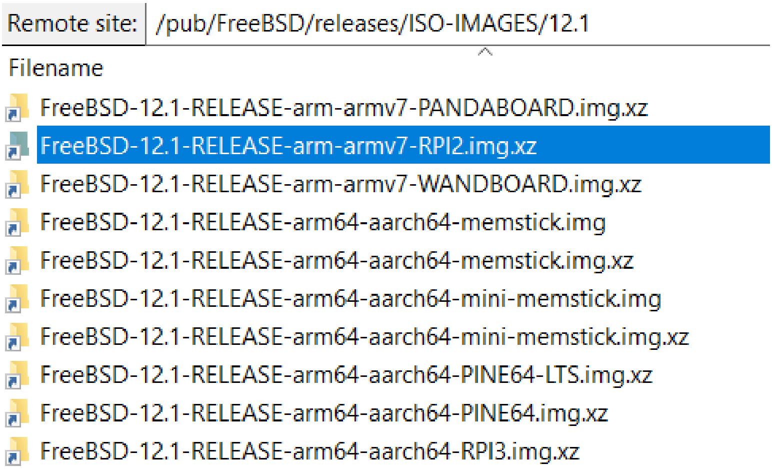
Download FreeBSD

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

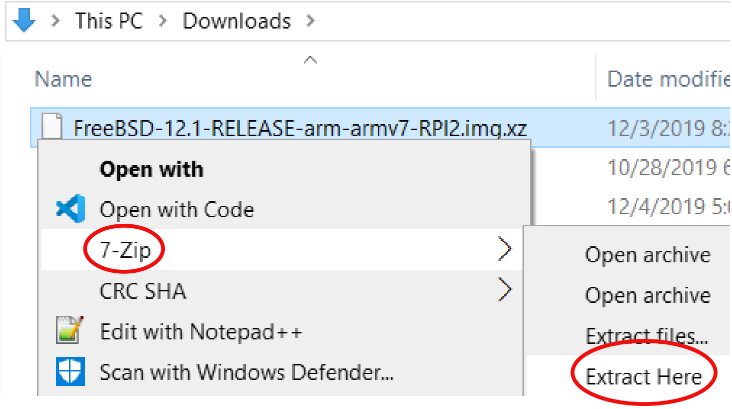
Filename

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.

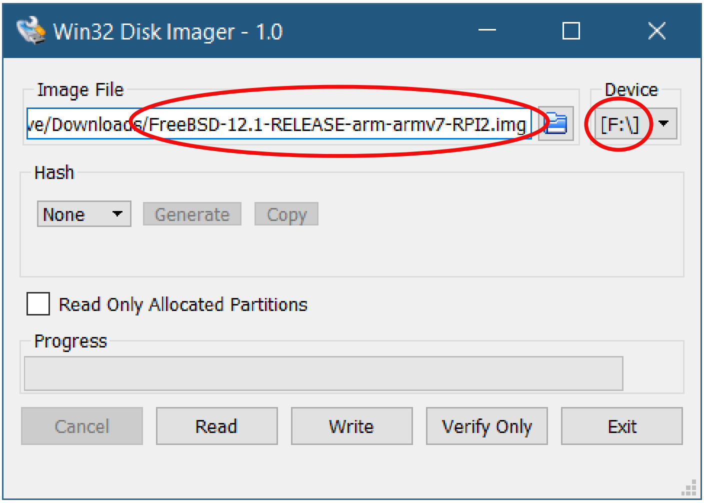


Image File: ve/Downloads/FreeBSD-12.1-RELEASE-arm-armv7-RPI2.img

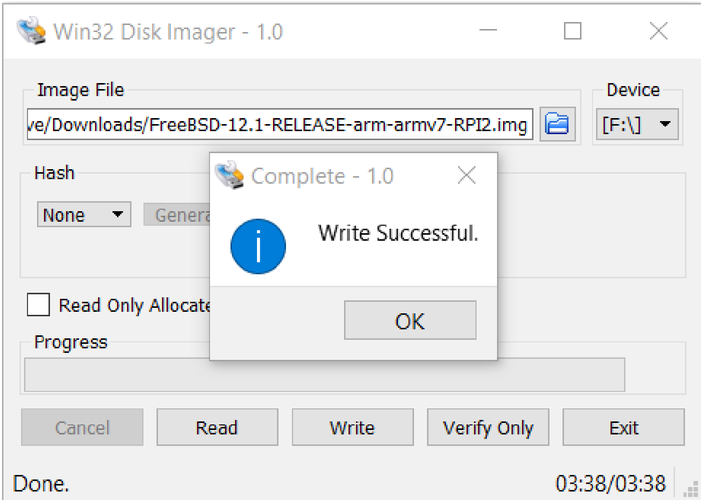
Device: [F:\]

Hash: None

□ Read Only Allocated Partitions

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.



Win32 Disk Imager - 1.0

Image File: ve/Downloads/FreeBSD-12.1-RELEASE-arm-armv7-RPI2.img

Device: [F:\]

Hash: None

□ Read Only Allocate

Complete - 1.0

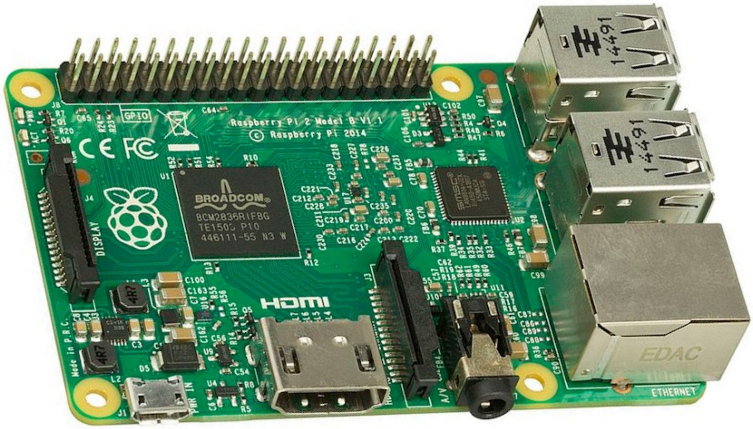
Write Successful.

Done. 03:38/03:38

When complete, eject the media and remove the card.

Booting FreeBSD

Insert the MicroSD card into the Raspberry Pi (left side of picture below.)



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

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.

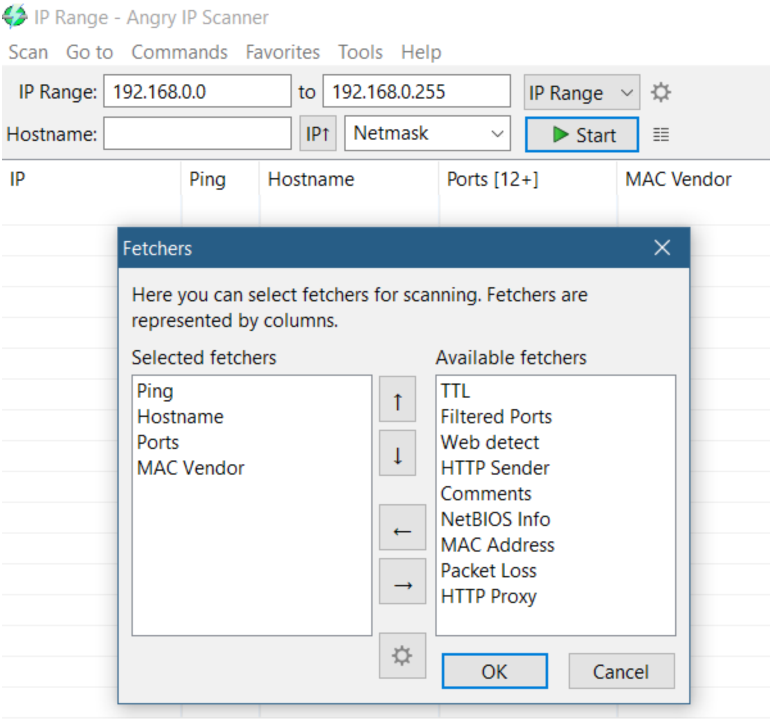
After a short bit of time, the red LED on the MicroSD socket side of the board will turn off. This means FreeBSD has started booting. Normally, this takes a minute at most. If the LED stays on longer than that, there may be a problem with the image written to the MicroSD card or the card itself.

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: 192.168.0.0 to 192.168.0.255

Hostname: Netmask

IP | Ping | Hostname | Ports [12+] | MAC Vendor

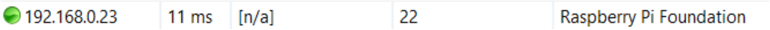
Fetchers

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

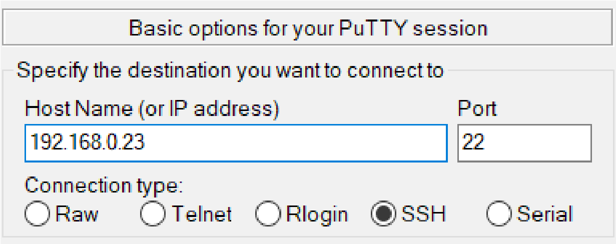
Run a scan of your network and note the IP address found in the output. This will be needed for the first login in the next step.



● 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

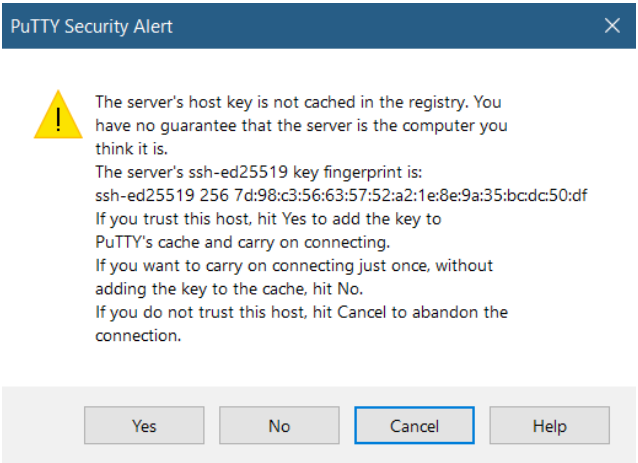
Host Name (or IP address): 192.168.0.23

Port: 22

Connection type:
○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Click the Open button.

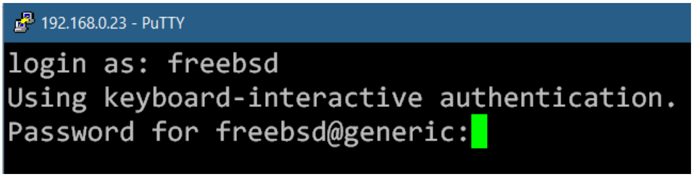
Since this is the first login to this device, you’ll get a security alert. It’s okay to say yes.



PuTTY Security Alert

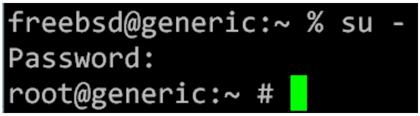
The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.
The server's ssh-ed25519 key fingerprint is:
ssh-ed25519 256 7d:98:c3:56:63:57:52:a2:1e:8e:9a:35:bc:dc:50:df
If you trust this host, hit Yes to add the key to PuTTY's cache and 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.



```
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.



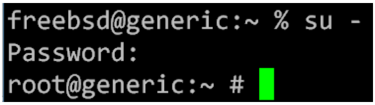
```
freebsd@generic:~ % su -
Password:
root@generic:~ #
```

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

Installing Node.js

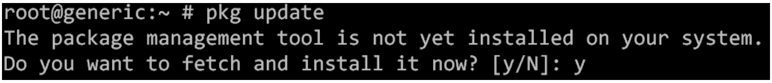
WeeNAS is a Node.js application and needs the Node.js interpreter to be installed first. This is done using FreeBSD’s pkg tool.

First, log in via SSH and su - if you haven’t done so already.



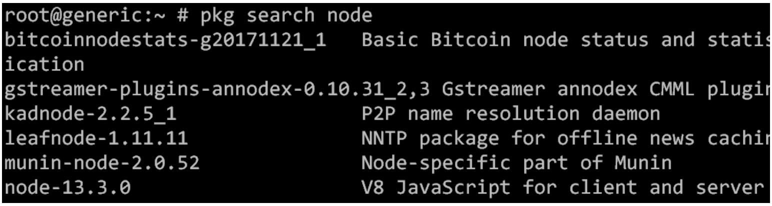
```
freebsd@generic:~ % su -
Password:
root@generic:~ #
```

After becoming superuser, get pkg ready by typing the command ‘pkg update’.



```
root@generic:~ # pkg update
The package management tool is not yet installed on your system.
Do you want to fetch and install it now? [y/N]: y
```

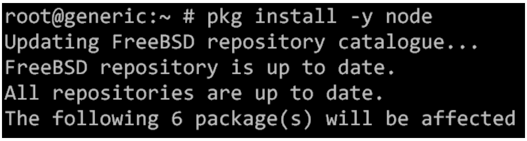
Next, find Node.js using the command ‘pkg search node’.   There will be several matches to sift through.



```
root@generic:~ # pkg search node
bitcoinnodestats-g20171121_1   Basic Bitcoin node status and statis
ication
gstreamer-plugins-annodex-0.10.31_2,3 Gstreamer annodex CMML plugi
kadnode-2.2.5_1                P2P name resolution daemon
leafnode-1.11.11               NNTP package for offline news cachi
munin-node-2.0.52              Node-specific part of Munin
node-13.3.0                    V8 JavaScript for client and server
```

As of this writing, node-13.3.0 was the latest version. Anything 13 or higher should work and it’s usually best to stay current for the latest bug and security fixes.

Finally, use the command ‘pkg install -y node’ to install the latest revision of version 13.3.0.



```
root@generic:~ # pkg install -y node
Updating FreeBSD repository catalogue...
FreeBSD repository is up to date.
All repositories are up to date.
The following 6 package(s) will be affected
```

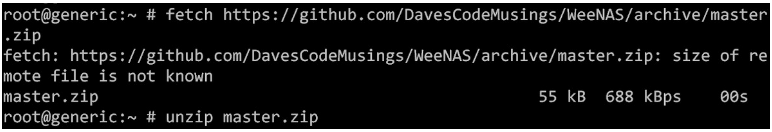
Now that Node.js is installed, WeeNAS can be downloaded and started.

Installing 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.

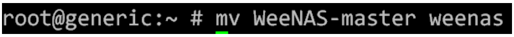
Change directory to /home/root and download the latest version of WeeNAS using the ‘fetch’ program and the link: https://github.com/DavesCodeMusings/WeeNAS/archive/master.zip

Then unzip it.



```
root@generic:~ # fetch https://github.com/DavesCodeMusings/WeeNAS/archive/master
.zip
fetch: https://github.com/DavesCodeMusings/WeeNAS/archive/master.zip: size of re
mote 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.

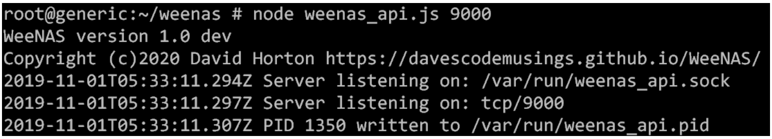


```
root@generic:~ # mv WeeNAS-master weenas
```

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

Starting WeeNAS for the First Time

If you just want to have a quick look at WeeNAS, it can be started on the command-line by changing directory to /root/weenas and running ‘node weenas_api.js 9000’.



```
root@generic:~/weenas # node weenas_api.js 9000
WeeNAS version 1.0 dev
Copyright (c)2020 David Horton https://davescodemusings.github.io/WeeNAS/
2019-11-01T05:33:11.294Z Server listening on: /var/run/weenas_api.sock
2019-11-01T05:33:11.297Z Server listening on: tcp/9000
2019-11-01T05:33:11.307Z PID 1350 written to /var/run/weenas_api.pid
```

This will start WeeNAS and make the web-based administration available via tcp port 9000. For the time being, it can be accessed via web browser like this:

http://<ip address>:9000/

Following the examples so far, that would be:

http://192.168.0.23:9000

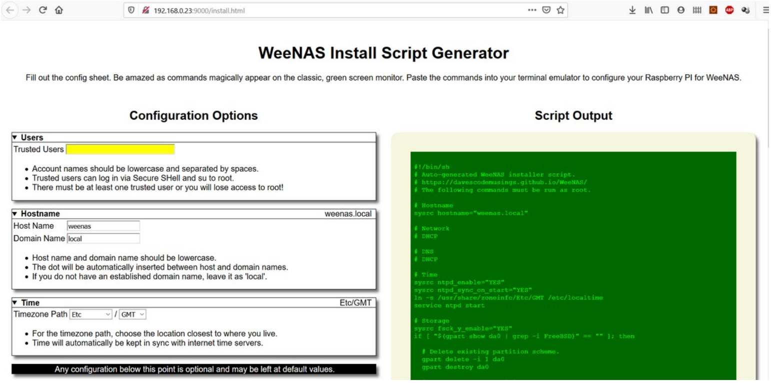
Your IP address will very likely be different.

Note: Instructions for installing WeeNAS so that it can be started and stopped as with FreeBSD’s service command can be found in the section entitled ‘WeeNAS as a Service’.

The WeeNAS Install Script Generator

Rather than making you type a bunch of commands, WeeNAS has the capability of creating a custom install script. This script can be run to carry out the initial configuration.

Access it from: http://<ip address>:9000/



WeeNAS Install Script Generator

Fill out the config sheet. Be amazed as commands magically appear on the classic, green screen monitor. Paste the commands into your terminal emulator to configure your Raspberry PI for WeeNAS.

Configuration Options

▾ **Users**

Trusted Users

- Account names should be lowercase and separated by spaces.
- Trusted users can log in via Secure SHell and su to root.
- There must be at least one trusted user or you will lose access to root!

▾ **Hostname** weenas.local

Host Name weenas

Domain Name local

- Host name and domain name should be lowercase.
- The dot will be automatically inserted between host and domain names.
- If you do not have an established domain name, leave it as 'local'.

▾ **Time** Etc/GMT

Timezone Path Etc / GMT

- For the timezone path, choose the location closest to where you live.
- Time will automatically be kept in sync with internet time servers.

Any configuration below this point is optional and may be left at default values.

Script Output

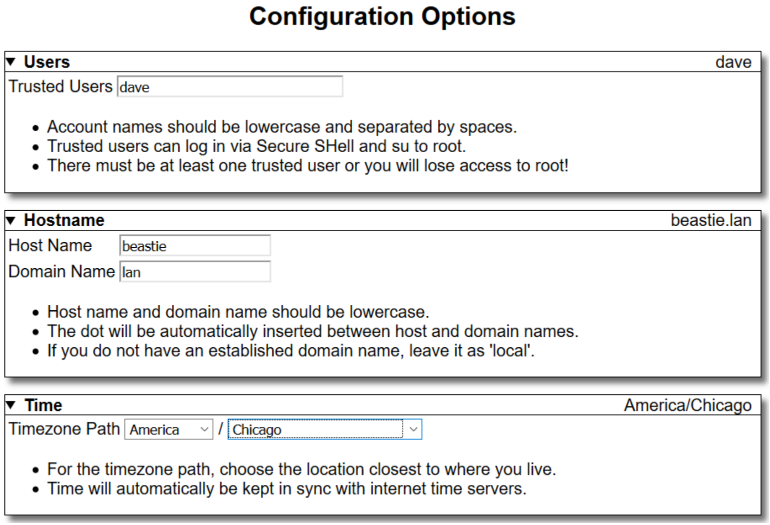
```
#!/bin/sh
# Auto-generated WeeNAS installer script.
# https://davescodemusings.github.io/WeeNAS/
# The following commands must be run as root.

# Hostname
sysrc hostname="weenas.local"

# Network
# DHCP

# DNS
# DHCP

# Time
sysrc ntpd_enable="YES"
sysrc ntpd_sync_on_start="YES"
ln -s /usr/share/zoneinfo/Etc/GMT /etc/localtime
service ntpd start

# Storage
sysrc fsck_y_enable="YES"
if [ "$(gpart show da0 | grep -i FreeBSD)" == "" ]; then

  # Delete existing partition scheme.
  gpart delete -i 1 da0
  gpart destroy da0
```

Note: WeeNAS web pages will adapt to different sized screens. And, while most administration can be carried out on a mobile device, using a large screen device is recommended for the initial install.

The only requirement of the install script generator is to specify one trusted user account. (It is highlighted in yellow so you won’t forget.)

This is the account that will have the ability to log into the underlying FreeBSD operating system via SSH. There must be at least one trusted account since the built-in account of freebsd will be locked as part of the initial configuration.

Other things you may wish to customize are the hostname and timezone.



Configuration Options

▼ **Users** dave

Trusted Users: dave

- Account names should be lowercase and separated by spaces.
- Trusted users can log in via Secure SHell and su to root.
- There must be at least one trusted user or you will lose access to root!

▼ **Hostname** beastie.lan

Host Name: beastie

Domain Name: lan

- Host name and domain name should be lowercase.
- The dot will be automatically inserted between host and domain names.
- If you do not have an established domain name, leave it as 'local'.

▼ **Time** America/Chicago

Timezone Path: America / Chicago

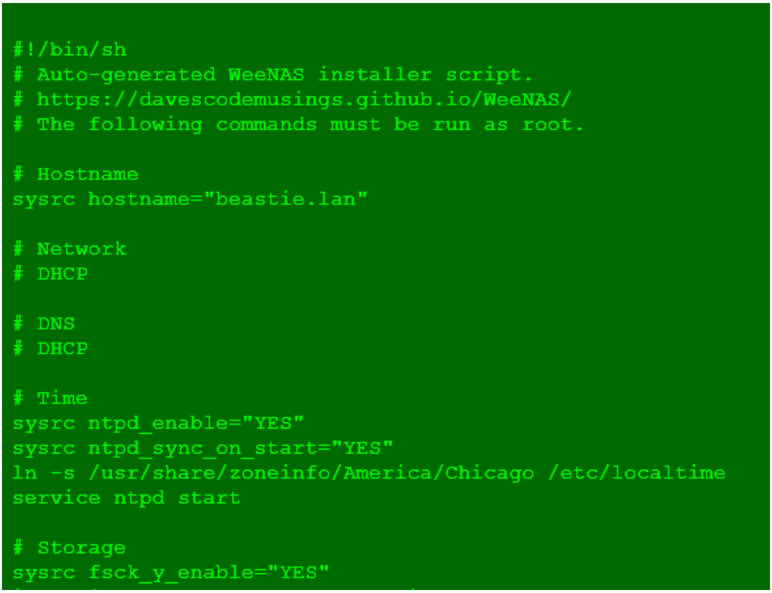
- For the timezone path, choose the location closest to where you live.
- Time will automatically be kept in sync with internet time servers.

As you make changes, the resulting commands will be shown on the simulated green screen monitor. (This will be located on the right side or below depending on the width of your device.)

Running Initial Configuration Commands

When you are satisfied with your configuration, you may copy commands one at a time and paste them into an SSH session or if you are more comfortable, you may copy everything to a file on the host and run them en masse.

A portion of the commands are shown in the screenshot below.



```
#!/bin/sh
# Auto-generated WeeNAS installer script.
# https://davescodemusings.github.io/WeeNAS/
# The following commands must be run as root.

# Hostname
sysrc hostname="beastie.lan"

# Network
# DHCP

# DNS
# DHCP

# Time
sysrc ntpd_enable="YES"
sysrc ntpd_sync_on_start="YES"
ln -s /usr/share/zoneinfo/America/Chicago /etc/localtime
service ntpd start

# Storage
sysrc fsck_y_enable="YES"
```

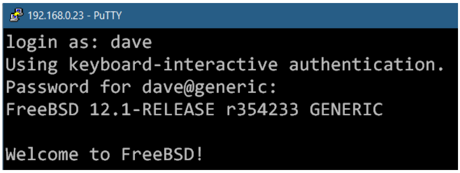
You will need to open a second PuTTY connection and log in as root since the first connection is busy running weenas_api.js as a foreground process. Or, you may shut down the WeeNAS API for now by pressing the CTRL+C combination.

Start with ‘sysrc hostname=…’ and work your way down. Lines that start with # are comments and do not need to be copied. Samba installation will take the longest if you’re looking to plan a break.

Testing Connectivity (SSH)

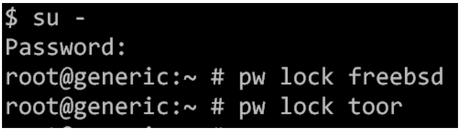
Secure Shell is your way to access the underlying FreeBSD operating system. This test is very important, because if it does not succeed, you will have no way to access the system.

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.



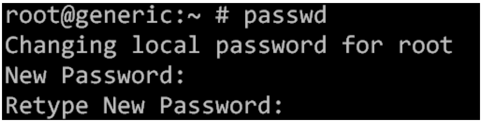
```
login as: dave
Using keyboard-interactive authentication.
Password for dave@generic:
FreeBSD 12.1-RELEASE r354233 GENERIC

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.

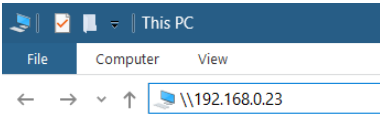


```
root@generic:~ # passwd
Changing local password for root
New Password:
Retype New Password:
```

Testing Connectivity (SMB)

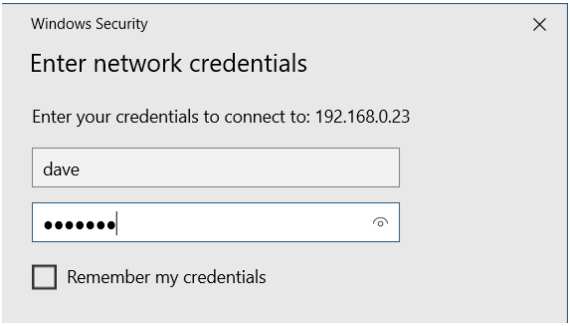
SMB is the Windows way of connecting to network shares. This is how you will access the files stored on the WeeNAS system.

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



\\192.168.0.23

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.


Windows Security

Enter network credentials

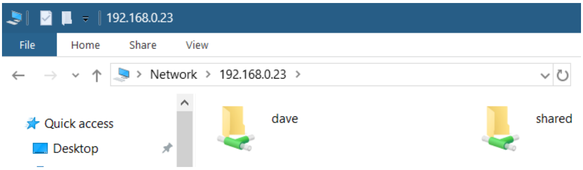
Enter your credentials to connect to: 192.168.0.23

dave

•••••••

□ Remember my credentials

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.

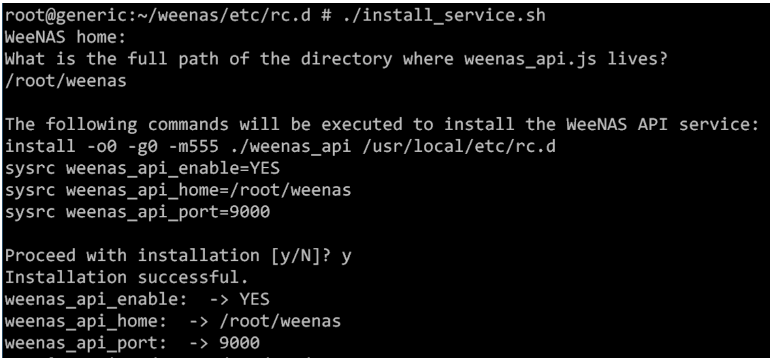
When you’re ready, you can reboot by logging in as a trusted user, ‘su -’, and then enter the command ‘shutdown -r now’.

WeeNAS as a Service

If you’re definitely keeping WeeNAS on your Raspberry Pi, it would be wise to install it as a service. That way, starting and stopping is easier and will be done automatically if ever the system is rebooted.

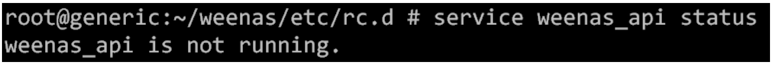
Start by logging in as a trusted user and ‘su -’ to the root user, then change directory to: /root/weenas/etc/rc.d

In this directory is a script named ‘install_service.sh’. Run this script to be guided through the process of configuration and installation.



```
root@generic:~/weenas/etc/rc.d # ./install_service.sh
WeeNAS home:
What is the full path of the directory where weenas_api.js lives?
/root/weenas

The following commands will be executed to install the WeeNAS API service:
install -o0 -g0 -m555 ./weenas_api /usr/local/etc/rc.d
sysrc weenas_api_enable=YES
sysrc weenas_api_home=/root/weenas
sysrc weenas_api_port=9000

Proceed with installation [y/N]? y
Installation successful.
weenas_api_enable:  -> YES
weenas_api_home:  -> /root/weenas
weenas_api_port:  -> 9000
```

That’s all there is to it. You can verify installation by checking the status of the weenas_api service.



```
root@generic:~/weenas/etc/rc.d # service weenas_api status
weenas_api is not running.
```

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.