

# **Getting Started with OMV**

## **Addendum A:**

### **Installing OMV5**

### **On Armbian SBC's**



May 26th, 2020 – (Rev1.7)

# Table of Contents

1	Introduction	3
	About this Guide.....	3
2	Not Supported	3
	Desktop versions of Armbian .....	3
3	Supported Devices	4
4	Prerequisites	4
5	Working with the image file	5
	Decompress the Image.....	5
	Verify the image .....	6
	Format and Test Flash Media.....	7
	Flashing Armbian onto an SD-card.....	10
6	The First Boot	11
7	Armbian - First Time Logon	12
8	Finishing the Armbian Install	13
9	Install OMV	14
	First Time GUI Logon.....	14
10	Finishing Up	16
11	Final Notes:	17
	Networking.....	17

# Introduction

Due to the lack of the numerous Single Board Computers (SBC's) required for testing and integration, providing images for all Armbian supported SBC's, with Openmediavault (OMV) pre-installed, will not be possible. However, installing OMV5 on Armbian supported SBC's is a relatively easy task. This document is a guide for that purpose.

## ***About this Guide***

The purpose and intent of this guide is to provide a walk-through to get SBC users up and running as quickly and as easily as possible. This guide assumes that users have a working Windows Client for installing and executing the needed utilities. It is also assumed that Mac and Linux desktop users will be able to find, install, and use utilities equivalent to those called out in [Prerequisites](#).

- This guide contains links to external sources of information and software. It's best used on an Internet connected PC.
  - This is a community document and a work in progress. Input and feedback are welcome and can be sent to: [omvguide@gmail.com](mailto:omvguide@gmail.com)
- 

## Not Supported

### ***Desktop versions of Armbian***

Armbian desktop versions are not supported and are at the user's risk. If a desktop is used, SD-card backup is recommended.

(Continued)

## Supported Devices

To see if an SBC of interest is supported by Armbian, visit the [Armbian Download Page](#).

OMV5 will install on most SBC devices with **Official** Armbian support and a **Buster** image:

- **Suitable For Testing**, “**WIP**” (Work in Progress), and **No Official Support** images may work, but things may be broken. Use of these images is at the user's risk. These images could be considered to be in “BETA” state. Problems are to be expected.
- Use of **End of Support (EOS)** images is highly discouraged. The state of the OS is undetermined and it may not update normally. The install process may fail or, if the installation is successful, OMV may be (or become) unstable. Again, use of **EOS** images is at the user's risk.
- Use of any of the above image classes is not supported.

## Prerequisites

This installation process requires a wired Ethernet connection and Internet access.

Typically, all that is needed to begin the installation is an Ethernet cable, an Armbian supported SBC, a power supply sufficient for the SBC model being used, and one SD-card (two are preferred for backup).

To get started, a few utilities are needed to expand, check, and work with an Armbian image.

- Armbian images are compressed with a .7z extension. Users will need the [7-Zip](#) utility to decompress the image. 7-Zip is installable on a Windows client.
- To check the decompressed image, an [MD5 – SHA Checksum utility](#) is needed. This utility is portable, meaning it's not necessary to install it, but it may require support files. Simply run the executable.
- [SDformatter](#) is a utility for formatting SD-cards, that does a [trim](#) operation on flash media to clear remnants of old files. SDformatter is installable on a Windows client.
- [h2testw\\_1.4](#) is a flash media test program. With a freshly formatted SD-card or USB thumbdrive, it writes files with known content and verifies the content in a read operation, detecting errors in the process. **h2testw\_1.4** downloads as a zip file. By right clicking on the zip file, and using “Extract All”, 7-Zip will expand the zip file to a folder named **h2testw\_1.4**. The executable inside this folder is a portable application. Run the executable.
- To burn an Armbian image to an SD-card, [Etcher](#) is recommended. (It burns the image and verifies it in one process.) Etcher is a portable app. Run the executable.
- [PuTTY](#) is an SSH client that will allow users to connect to their SBC, from a Windows client, and get on the command line. PuTTY is installable.
- While 8GB is the minimum and will work fine, a 16GB card will provide longer life in the role of a boot drive. Users are encouraged to get two SD-cards. One is for the installation and the second is for backing up the OS installation, when configuration is complete.

For the best experience, use only quality **new** SD-cards, such as Samsung or SanDisk, that are rated A1 Class 10.



(Continued)

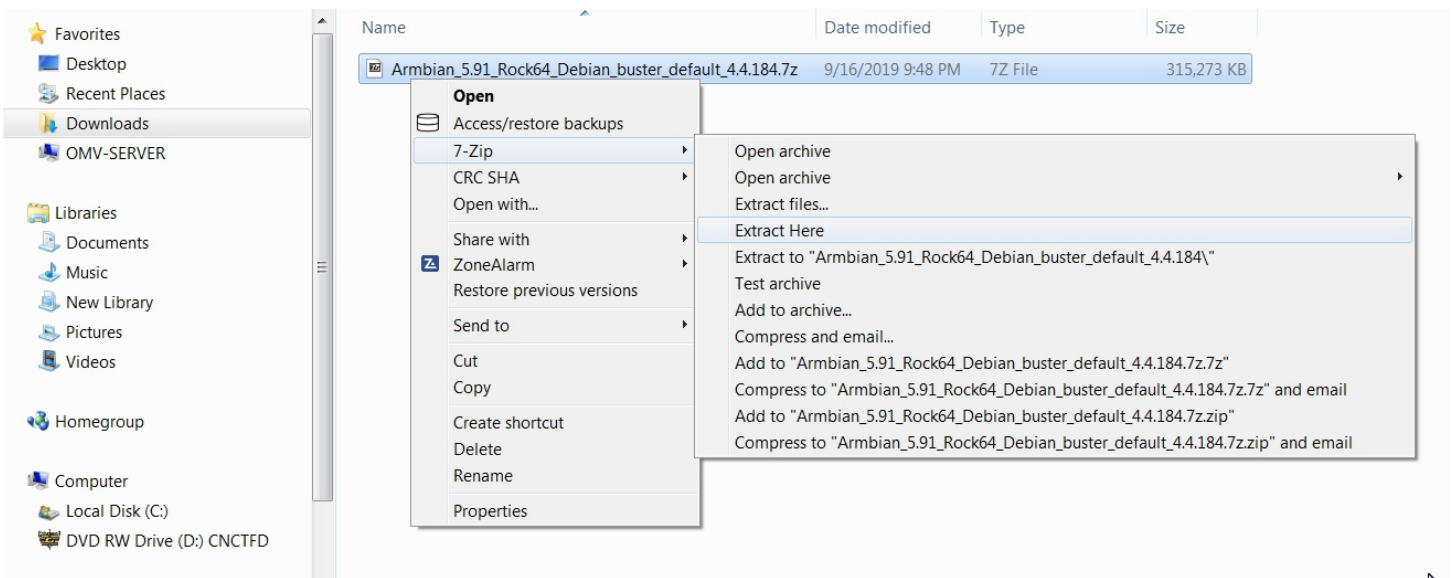
– **\*\*Important\*\***

- When selecting an Armbian image to download, for best possible compatibility with OMV5, select the Armbian **Buster Server** image for your device.
- On the Armbian device page, scroll down to **bottom**, to the “**All download options**” section, and select the **Buster Server** variant. If **Buster Server** is not available, **Buster Minimal** should work. ((Due to the potential for package conflicts, a Buster image with a **desktop** is **not recommended or supported**. Lastly, Bionic or Stretch images will not work with OMV5.))

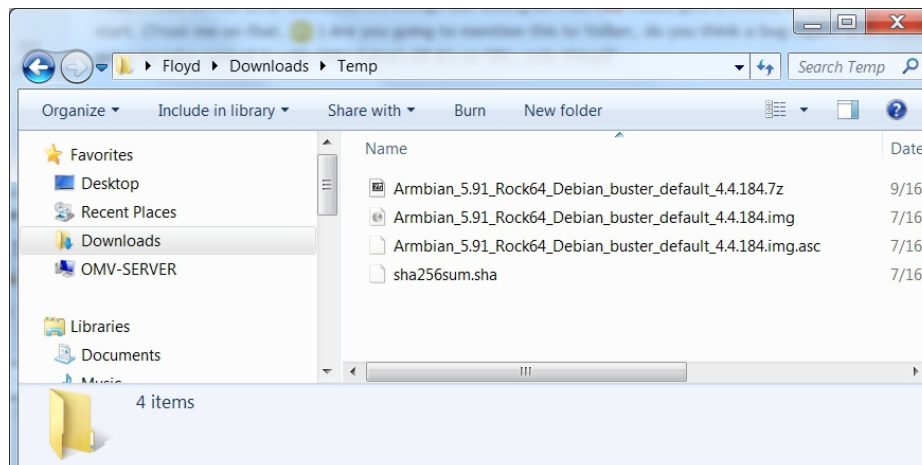
## Working with the image file

### Decompress the Image

Armbian images are compressed and will need to be extracted with 7-Zip. Highlight the compressed file, right click the mouse, and make the menu selections shown below.



The result of the above action is the extraction of files into the current directory. The image file's extension is **.img** (For the purposes of this guide, the file ending with **.asc** can be ignored.)



## Verify the image

MD5 and SHA hashes check for image corruption that may have occurred during the download.

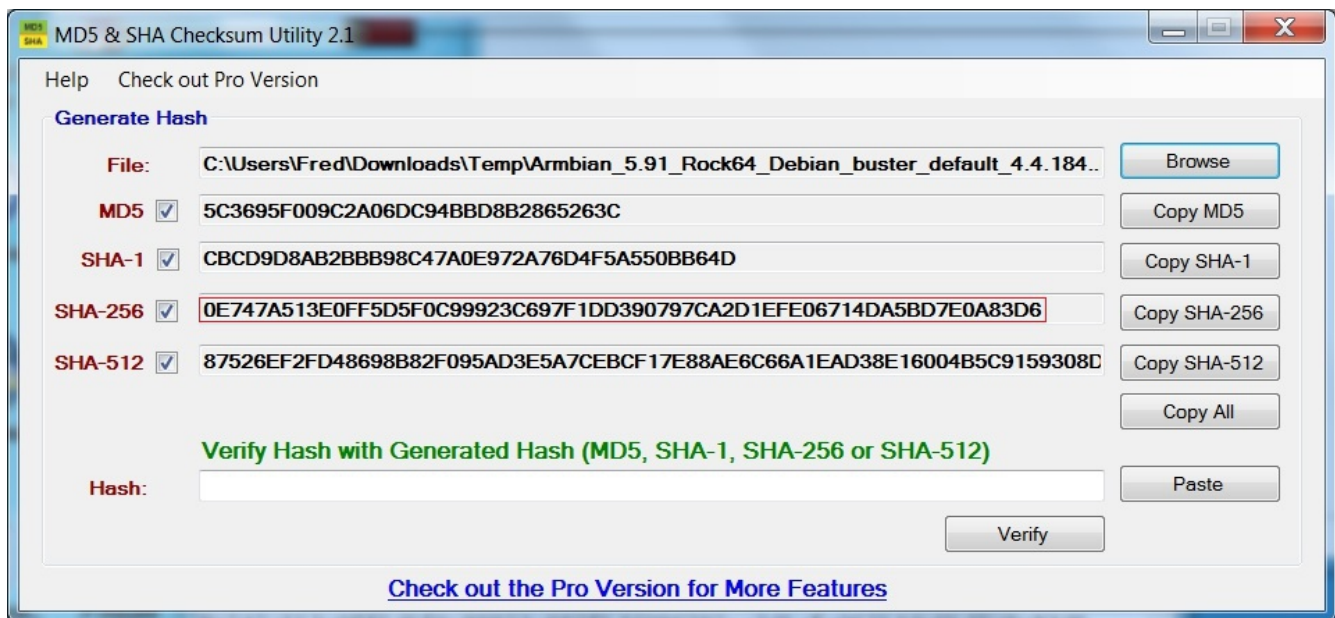
### **\*\*Beginners Note\*\***

**DO NOT SKIP THIS STEP.** The chance of image corruption is highest when downloading and it's pointless to build a server with flawed software. Even the slightest corruption of the image may ruin your installation and the effects may not be noticed until well after your server is built and in use. Headaches can be avoided by checking the image.

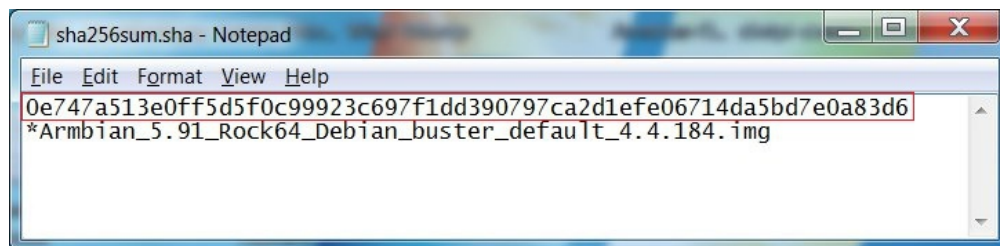
---

Verify the image file, for your SBC, with the [MD5 & SHA checksum utility](#).

(Note that it's possible to “drag and drop” the file name into the utility, on the **File** line. Otherwise, use the **Browse** button and navigate to the image file.)



With the **File, Open**, using the **ALL Files \*.\*** setting, Notepad can be used to open the file **sha256sum.sha**



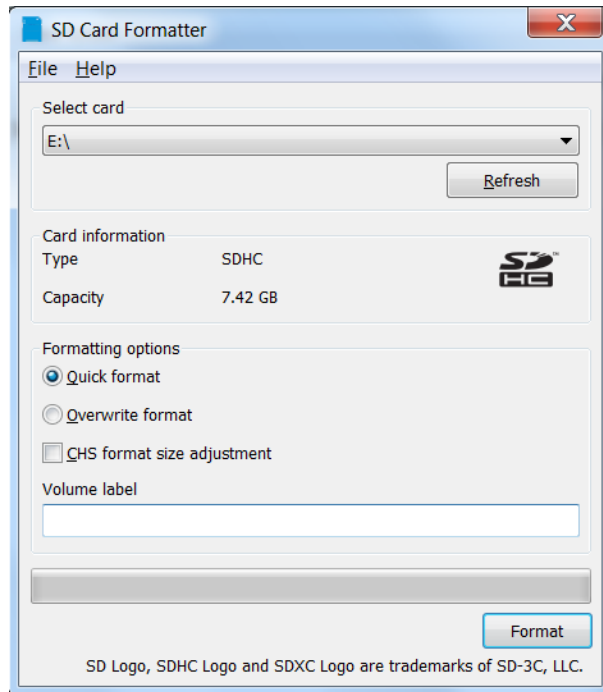
With a sha-256 match, the downloaded image is verified.

## ***Format and Test Flash Media***

Using SDFormatter, do a clean format:

(Note that SDFormatter does a trim operation on the card which cleans up remnants of deleted or previously existing files.)

In most cases, SDFormatter will detect the SD-card or thumb-drive. A volume label is not necessary, at this point, and the default options are fine.



After the SD-card format is completed, open **h2testw** and select your language.

Then, click on **Select target**



Under **Computer**, select the flash media previously formatted.





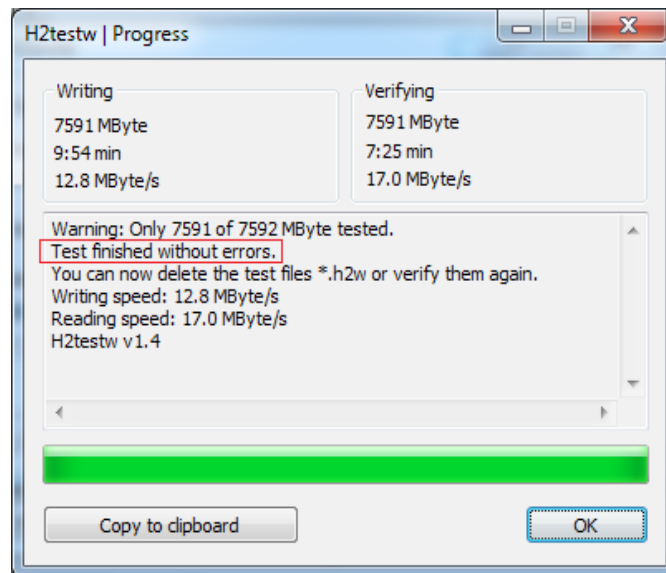
Select **Write+Verify**. (**DO NOT** check the **endless verify** box)



A dialog similar to the following may pop up. Ignore it and click on **OK**.



**“Without errors”** is the desired outcome. If the media tests with errors or is much smaller than is indicated by the SD-card's labeled size, don't use it.



After H2testw verifies the SD-card; **do one more clean format**, using **SDFormatter**, before flashing the card. While optional, at this point, a volume label could be applied.

## Flashing Armbian onto an SD-card

### Start Etcher:

Etcher, in most cases, will auto detect the SD-card or a USB thumb-drive. Click on “**Select Image**” and navigate to the decompressed Armbian image. Then click on **Flash!**

- A windows confirmation dialog may pop up. (Click on **OK**.)
- Etcher will write the image, then verify it in one operation.



A “**Success**” flag will pop up when the job is finished and the Etcher window will display “Flash Another?”. The flash operation is complete.





## The First Boot

**\*\*At this point, to connect to the SBC with PuTTY and to install OMV in a later process, a wired Ethernet connection is required.\*\***

- Insert the SD-card, into the SBC, and apply power.
- Wait 3 to 5 minutes.

The IP address to use for logging into the console is available from your DHCP server. In most cases, your DHCP server will be running on your LAN's router.

(The following is an example.)

DHCP Clients				
Hostname	IP Address	MAC Address	Client Lease Time	Delete
rock64	192.168.1.69	02:31:91:B8:47:3C	0 days 00:02:00	
hotrod	192.168.1.94	10:7B:44:7B:CA:C9	0 days 00:02:00	
Dell-Win10	192.168.1.90	78:2B:CB:A4:1F:4C	Static	
LenovoLaptop	192.168.1.68	50:5B:C2:F8:3A:75	Static	

With the IP address noted, proceed to **First Time Login**.

---

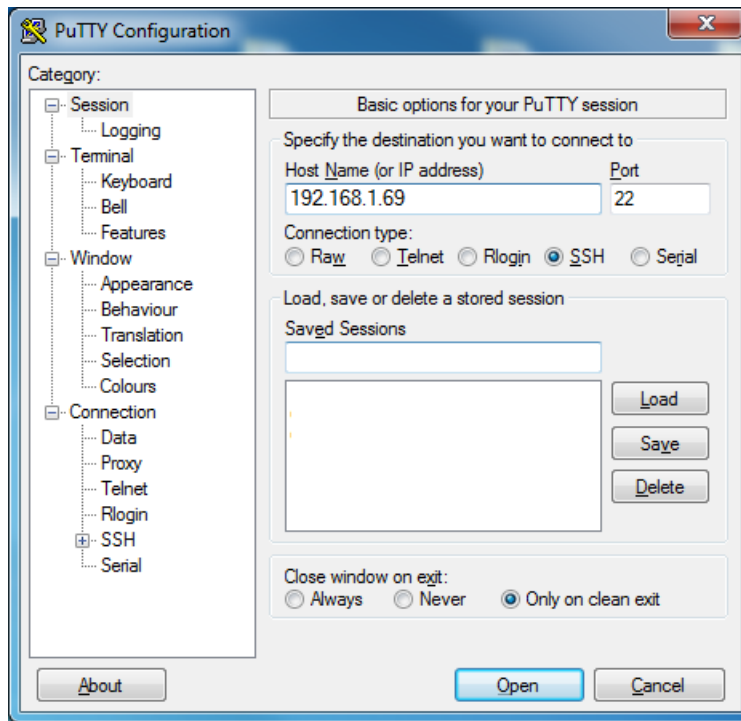
### **\*\*If there's a problem with obtaining a DHCP assigned IP address:\*\***

- In the event that an IP address is not issued to your SBC, check the wired Ethernet connection and reboot the device. In some cases this will mean unplugging and plugging the power supply back in. Allow time for boot up (5 minutes or so) and check the DHCP server again.
- If an address is not issued, or if the user doesn't know how to find the ARM device's IP address on their DHCP server, connect a monitor and keyboard to watch the boot process until it completes. If the IP address is not displayed at the end of the boot cycle, login with the user **root** and password **1234**

Unfortunately, this first time logon may kick off a mandatory root user password change and the addition of a non-root user. Once these processes are complete, type **ip add** on the command line. Note the IP address of the Ethernet interface, in the output, and proceed to First Time Login.

# Armbian - First Time Logon

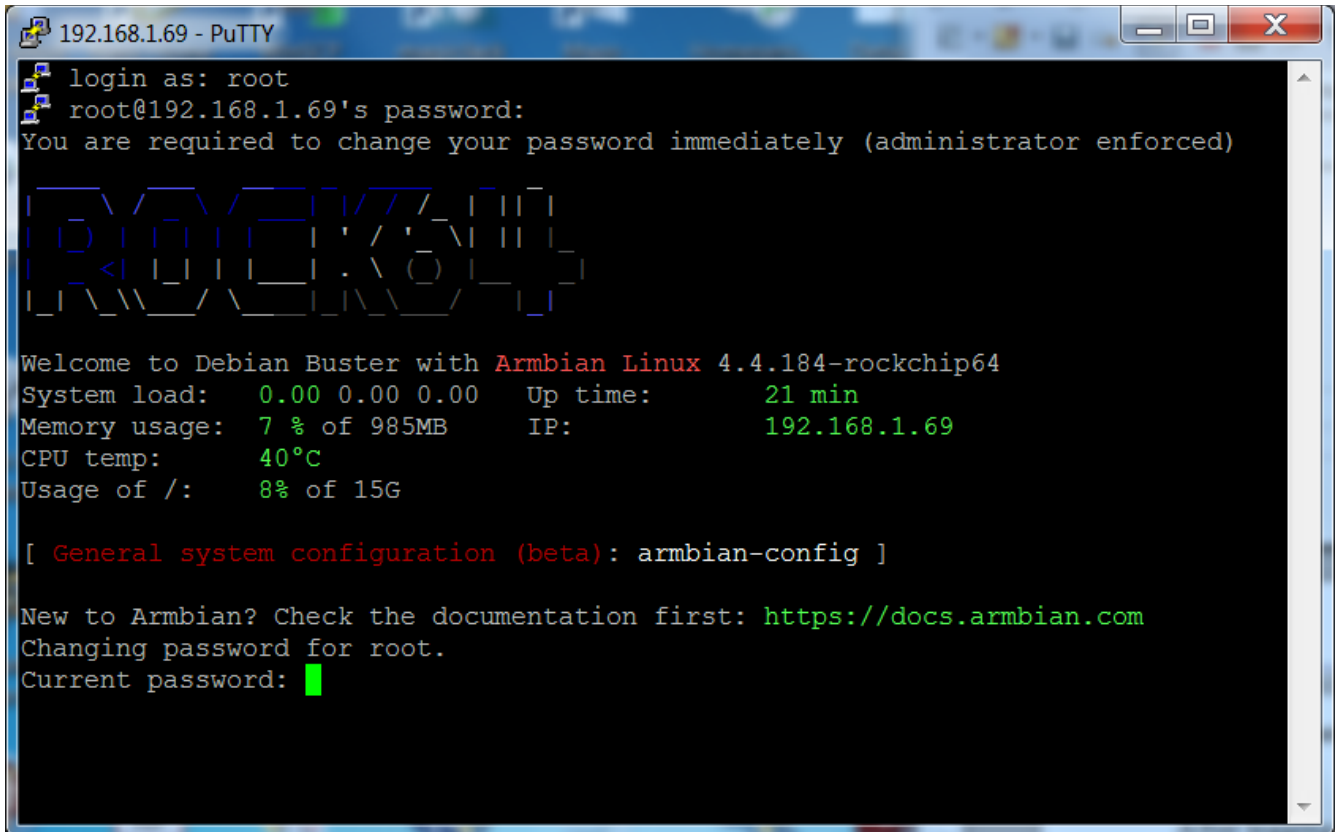
Open PuTTY and type in the OMV IP address



A PuTTY Security Alert will pop up in a first time connection.  
This is normal. Ignore it and select **Yes**.

## When the SSH window opens:

Login as: **root** The Armbian default password is: **1234**



```
192.168.1.69 - PuTTY
login as: root
root@192.168.1.69's password:
You are required to change your password immediately (administrator enforced)

Welcome to Debian Buster with Armbian Linux 4.4.184-rockchip64
System load:  0.00 0.00 0.00   Up time:    21 min
Memory usage:  7 % of 985MB   IP:        192.168.1.69
CPU temp:      40°C
Usage of /:    8% of 15G

[ General system configuration (beta): armbian-config ]

New to Armbian? Check the documentation first: https://docs.armbian.com
Changing password for root.
Current password: 
```

After logging in with the default password, you'll be required to change the root password. Re-enter the current password **1234**, then follow the prompts to enter and confirm a new root password. (Remember this password.)

When the root password change is complete, Armbian will prompt to create a non-root user, with “sudo” admin abilities. Follow the prompts to complete the task. (If desired, this user can be altered, deleted, or replaced later. )

---

## Finishing the Armbian Install

On the command line, copy and paste the following command into the SSH window:  
**apt-get update**

When the command prompt returns copy and paste the following command into the SSH window:  
**apt-get upgrade**  
(This process will take a few to several minutes.)

Finally, reboot the SBC with the following command:  
**reboot**

In a few minutes, open a new PuTTY/SSH window for the installation of OMV.

**\*\* In the event that the SSH client does not respond to the IP address used for the Raspbian installation, look at your DHCP server to see if a “new” IP address has been assigned.\*\***

## Install OMV

Installing OMV on Armbian is very easy, thanks to **Ryecoaaaron** for providing a comprehensive installation script that's executed from a single line.

Copy the following line complete (**Ctrl+C**) and paste it into PuTTY's SSH window, with a right mouse click. Then hit **Enter**.

```
wget -O - https://github.com/OpenMediaVault-Plugin-Developers/installScript/raw/master/install | sudo bash
```

Once the script is running, click out of the SSH window so the script will not be interrupted. Depending on several factors, running this script may take up to 30 minutes.

When the script is complete, the R-PI will automatically reboot.

---

### ***First Time GUI Logon***

After 3 to 5 minutes, OMV can be logged in using the same IP address that was used for the SSH client, entered in a web browser address bar. The web GUI user is **admin** and the default password is **openmediavault**

---

#### **\*\*Note\*\***

After the completion of the script:

In the rare event that the OMV console or SSH client does not respond to the IP address used during the installation, look at your DHCP server to see if a “new” IP address has been assigned.

((Typically, consumer router DHCP leases last 24 hours. In cases where DHCP leases are very short - as it is with some versions of DD-WRT router firmware [10 minutes] - the lease issued to the SBC may time out at the end of the installation. A different address may be issued on reboot.))

(Continued)

## Finishing Up

New users can continue with the setup of OMV using the [Getting Started with Openmediavault 5](#) guide , starting in the section titled **OMV - Initial Configuration**.

All users are encouraged to review the section titled **Operating System Backup** for an easy process to clone your SBC's SD-Card.

## Donate

Openmediavault on Raspberry PI's is the project of OMVextras.org

In addition to enabling the installation of OMV on R-PI's, OMV-extras.org develops plugin's for OMV that make Portainer (Dockers) and other extensions available for your R-PI possible.

Please consider a modest donation to support continuing development and to help offset some of the Developer's costs. [OMV-extras.org](https://omv-extras.org)

## **Final Notes:**

### ***Networking***

The decision was made to depreciate NetworkManager in SBC's. The primary reason for this decision was to better support the installation and use of Dockers.

Supporting Portainer, to allow easy use of Dockers, was a priority add-on for OMV5. Unfortunately, NetworkManager wrecks networking for Portainer and Dockers.

.