

[Blog](#)[Contact Me](#)

Home IT Dec 13 • Written By Oleg Ovechkin

# Using a Raspberry Pi for Time Machine

Updates:

- April 17th, 2022. I have posted a [series of articles](#) on using Raspberry Pi to implement 3-2-1 backup strategy. Feel free to check the [Introduction](#).
- April 20th, 2022. The note on [Raspberry Pi OS Bullseye April 2022 update](#) is added.

This is a step-by-step, start-to-finish guide to configure a Raspberry Pi as a network storage device for macOS Time Machine backups. A few things to get out of the way before proceeding:

- Since we are talking about Time Machine, I assume you have at least one Mac in your possession with working SSH client.
- You are not afraid of Terminal and comfortable with command line.
- I also assume that the Time Machine backups will be encrypted, so we won't be dealing with encryption on the Raspberry Pi end.

Please note, just backing up your Mac to an external drive or network attached storage without sending those backups also off premises does not constitute a solid data backup and recovery strategy. Here is a [paper from Carnegie Mellon University](#) produced for US-CERT discussing the topic in more details.

## Prerequisites

- **Raspberry Pi 4.** While you can use almost any model of Raspberry Pi, I recommend Pi 4 simply because it features USB 3.0, which significantly improves speed of writing / reading to and from external hard

drives and its Gigabit Ethernet port throughput is not capped at 300Mbps.

- I buy kits made by CanaKit. They come with beefier 3.5 amp power supplies guaranteeing there will be enough juice to power an external 2.5 inch spinning hard drive. I also prefer passive cooling cases, and CanaKit has a few SKUs including those - I believe CanaKit rebrands cases made by Flirc. If you decide to go this route, I highly recommend getting a set of heatsinks as well, for some reason aluminum case CanaKit kits come with no heatsinks for memory or IO chips \\_(`)\_-
- **MicroSD card.** Raspberry Pi boots from a microSD card. Majority of kits include one, otherwise you will need to get it, 4GB or larger capacity.
- **MicroSD card reader** that works with a Mac. We will be preparing bootable microSD media using a Mac.
- **External USB hard drive / SSD.** Any will do as long as it's roomy enough for your backup needs and has USB 3.0 interface. I purchased and use Seagate 5TB USB 3.0 2.5 inch external hard drive. Those are sold at Costco (and frequently go on sale). So far it has served me well without any issues.
- **Ethernet cable.** It is a good idea to keep your Raspberry Pi off Wi-Fi network to ensure the best possible performance during backups.

## Installing OS

Let's get our microSD card ready for the first boot of the Raspberry Pi. The steps in this section are performed on a Mac.

- First, download and install the official [Raspberry Pi Imager](#) software to your Mac.
- Run Raspberry Pi Imager and choose **Raspberry Pi OS (32-bit)** as shown below:



Fig 1. Raspberry Pi Imager Home Screen

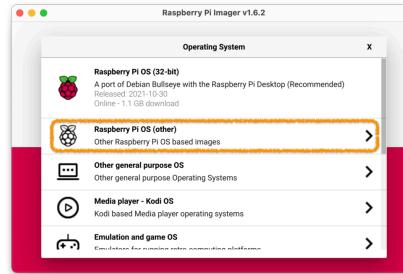


Fig 2. Raspberry Pi Imager OS Selection. Raspberry Pi OS Other

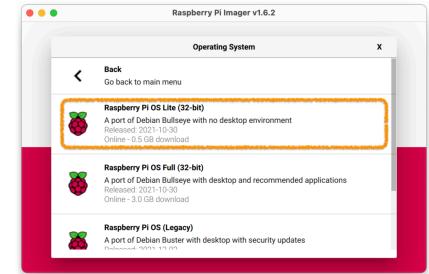


Fig 3. Raspberry Pi Imager OS Selection. Raspberry Pi OS Lite

- Next, click **Choose Storage** (middle button on the Raspberry Pi Imager Home Screen) and choose the

card reader with the microSD in it.

⚠ Note on Raspberry Pi OS Bullseye April 2022 update.



Fig 2022-04-20-Update-1. Interface of Raspberry Pi Imager v1.7.2

In April 2022, Raspberry Pi Foundation introduced several changes, including deprecating the default **pi** user account. Please refer to [Simon Long's blog post](#) for a detailed explanation. Check the version of your Raspberry Pi Imager, and if it is 1.7.2 or later, you have to click on the **Cogwheel** button to bring **Advanced options** dialog - please refer to Fig 2022-04-20-Update-2. The shown values for hostname, username, and password revert the Raspberry Pi's behavior to the pre-April 2022 state. While this makes following this guide easier, I would advise choosing a unique username and a strong password for your default user. Doing so increases the

security of your Raspberry Pi - just remember to substitute **pi** with your default user's username everywhere below.

Note that **Advanced options** dialog now gives an option to enable SSH and provides for setting the hostname and configuring the timezone in advance. Feel free to skip the corresponding steps at the end of this article.

**!** Note that the April 2022 Update has introduced a new behavior enabling automatic console login for the user specified in **Advanced options** dialog. Not turning it OFF might compromise the security of your Raspberry Pi. Please read [this article](#) for more details.

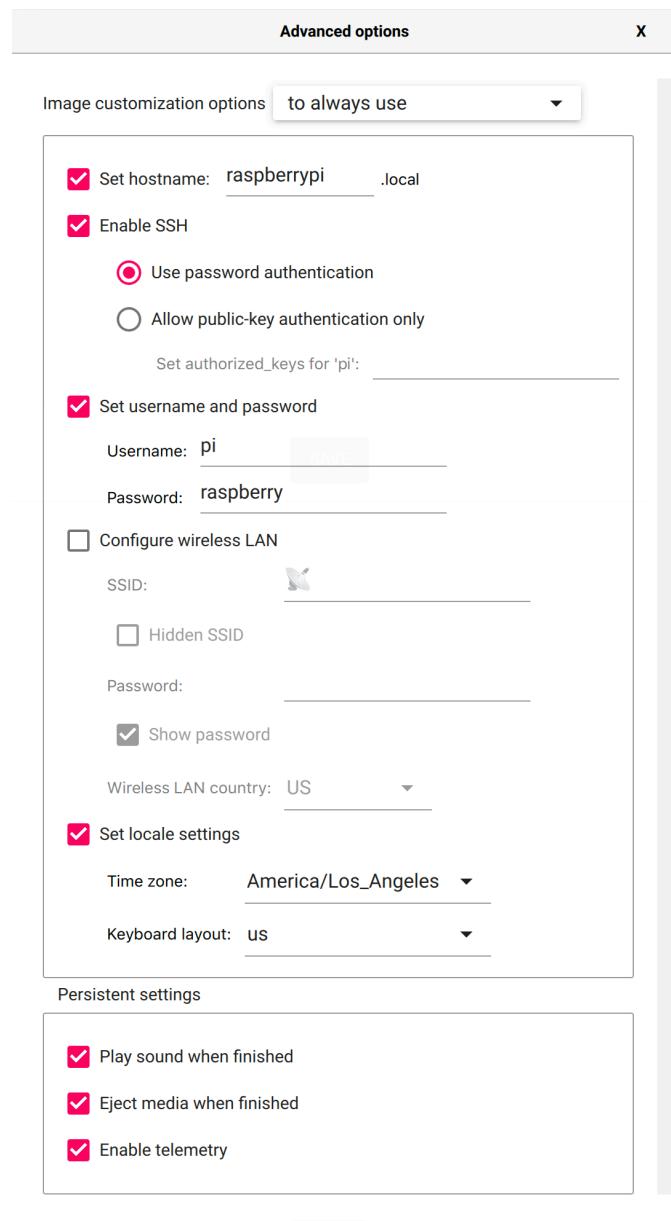


Fig 2022-04-20-Update-2. Advanced Options Dialog of Raspberry Pi Imager v1.7.2

- When everything looks good, proceed with writing the selected Raspberry Pi OS Lite to the chosen storage location of the microSD. The process takes anywhere from one to five minutes depending on whether the OS image needs to be downloaded and your broadband connection speed.
- Once you see **Write Successful** dialog box appear, click on **Continue** button then quit Raspberry Pi Imager.

At this point, the microSD card has been unmounted by Raspberry Pi Imager, but we are not quite done yet.

Let's unplug and then attach the card reader one more time. Then:

- Open Terminal on Mac.
- From the Terminal window execute:  
`$ cd /Volumes/boot && touch ssh && cd`
- Unmount / Eject the card.

This last command creates an empty file named **ssh** in the /boot partition of the microSD card. The presence of this file will make the Raspberry Pi OS enable SSH daemon on the subsequent boot, which is what we want, to avoid plugging in a keyboard and monitor to the Pi.

Finishing up:

- Insert the microSD into your Raspberry Pi.
- Connect Ethernet cable to the ethernet port of your router.
- Connect the power and power the Pi up!

## Connecting to the Raspberry Pi

After a minute or two after powering up, your Raspberry Pi should become available on your home network. In most cases, I suspect, you should be able to SSH to it using **raspberrypi.local** name. If not, then you will need to login to your home router management console to find out the Pi's IP address from the router device list.

Note: Official Raspberry Pi Documentation has an in-depth coverage of the [Remote Access](#) topic.

From here on I assume that the Pi is indeed accessible via **raspberrypi.local**.

- Open Terminal on Mac, enter the following command:  
`$ ssh pi@raspberrypi.local`
- Agree to accept the Raspberry Pi's SSH fingerprint to your Mac's known hosts list.
- When prompted, enter the default password **raspberry**

On the greeting screen, Raspberry Pi OS warns you about the security risk associated with using the default password other SSH. Do not worry, we will take care of it at the end of the article.

```

pi@raspberrypi: ~ -- ssh pi@raspberrypi.local — 96x30
[redacted]
user@apricity ~ % ssh pi@raspberrypi.local
The authenticity of host 'raspberrypi.local (fe80::fd10:c4e:f65c:90d8%en0)' can't be established
.
ECDSA key fingerprint is SHA256:b6uOAb2EZAp62V8qjqZPCVFqufiy3iyOny3C5KQPbKo.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'raspberrypi.local,fe80::fd10:c4e:f65c:90d8%en0' (ECDSA) to the list
of known hosts.
pi@raspberrypi.local's password:
Linux raspberrypi 5.10.63-v7l+ #1459 SMP Wed Oct 6 16:41:57 BST 2021 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

Wi-Fi is currently blocked by rfkill.
Use raspi-config to set the country before use.

pi@raspberrypi:~ $ [redacted]

```

Fig 4. Connecting to Raspberry Pi via SSH

- Before diving into anything else, let's refresh the software packages on our Pi. From the SSH console on your Mac execute:

```
$ sudo apt update && sudo apt upgrade -y
```

## Preparing the Storage

**⚠ Warning:** all data on your external hard drive is about to be permanently deleted. Make sure there is nothing on it that you might regret not having later.

- Connect external hard drive via USB to one of the blue USB 3.0 ports on the Raspberry Pi.
- From SSH console execute:

```
$ sudo lsblk
```

You should see the list of devices and partitions present. Note the device name that does NOT have root (/) or boot (/boot) partitions associated with it. In my case this device is **sda** - refer to Fig 5 below. If your device name is different, you will need to be careful to update all the following commands with the actual name of your external hard drive device.

- So, assuming **sda**, execute

```
$ sudo dd if=/dev/zero of=/dev/sda bs=512 count=10000
```

which zeros out the first ~5MB of the hard drive, erasing any service information like partitions, etc.

- Then follow up with the following duo:

```
$ sudo parted /dev/sda mklabel gpt
```

```
$ sudo parted /dev/sda -a opt mkpart primary 0% 100%
```

- The last command creates a single partition **/sda/sda1** as confirmed by the output from:

```
$ sudo lsblk
```

```
pi@raspberrypi:~ $ sudo lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda      8:0    0  4.5T  0 disk
mmcblk0  179:0   0 59.7G  0 disk
└─mmcblk0p1 179:1   0  256M  0 part /boot
└─mmcblk0p2 179:2   0 59.4G  0 part /
pi@raspberrypi:~ $ sudo dd if=/dev/zero of=/dev/sda bs=512 count=10000
10000+0 records in
10000+0 records out
5120000 bytes (5.1 MB, 4.9 MiB) copied, 0.461501 s, 11.1 MB/s
pi@raspberrypi:~ $ sudo parted /dev/sda mklabel gpt
Information: You may need to update /etc/fstab.

pi@raspberrypi:~ $ sudo parted /dev/sda -a opt mkpart primary 0% 100%
Information: You may need to update /etc/fstab.

pi@raspberrypi:~ $ sudo lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda      8:0    0  4.5T  0 disk
└─sda1    8:1    0  4.5T  0 part
mmcblk0  179:0   0 59.7G  0 disk
└─mmcblk0p1 179:1   0  256M  0 part /boot
└─mmcblk0p2 179:2   0 59.4G  0 part /
pi@raspberrypi:~ $
```

Fig 5. Partitioning External Hard Drive

- To create ext4 file system on **/sda/sda1** giving it label **backups**, proceed with:

```
$ sudo mkfs.ext4 -L backups /dev/sda1
```

## Mounting Backups Partition

- Let's create a directory to serve as the mounting point for the newly created **backups** filesystem:

```
$ sudo mkdir /mnt/backups
```

- To be able to mount this filesystem on our Raspberry Pi and access it like any other directory, we need to make an entry into /etc/fstab system file. Doing so also ensures that the mount persists between reboots. The command below appends an entry to the fstab letting Raspberry Pi know what should be mounted and where:

```
$ echo 'LABEL=backups /mnt/backups ext4 noexec,nodev,noatime,nodiratime 0 0' | sudo tee -a /etc/fstab
```

Note of flags used in the command above:

- **noexec**: do not allow execution of any binaries on the mounted filesystem as it's only going to be used for backup storage;
- **nodev**: do not interpret character or block special devices on the filesystem;
- **noatime**: do not store access times on this filesystem;
- **nodiratime**: do not store directory access times on this filesystem.
- The two numbers at the end of the line instruct your Raspberry Pi to ignore this filesystem when using dump and not to check it with fsck when rebooting.

- At this point we should be able to mount:

```
$ sudo mount /mnt/backups
```

## Optional: Putting the External Hard Drive to Sleep

For spinning disks that do not have to run constantly like in our case, it's not a bad idea to put them to idle when there is no activity. Not only does that prolong disks life, but also reduces carbon footprint.

- **hdparm** is there to help:

```
$ sudo apt install hdparm -y
```

- Then execute:

```
$ sudo hdparm -S 120 /dev/disk/by-label/backups
```

Here we use -S option to specify the period of inactivity before spinning the drive down. I highly recommend reading more on hdparm options (**\$ man hdparm** is your friend) because

according to man pages, “The encoding of the timeout value is somewhat peculiar.” In our case, however, -S 120 instructs hdparm to wait for 10 minutes before spinning down.

- We can make this permanent by adding the following to /etc/hdparm.conf:

```
$ echo -e '\n/dev/disk/by-label/backups {\n\tspindown_time = 120\n}' | sudo tee -a /etc/hdparm.conf
```

## Making the Pi Storage Network-Accessible

In this section we will be configuring two additional packages. Samba will help with accessing the **backups** partition over local network via Server Message Block protocol (or SMB for short). Meanwhile Avahi is used to advertise **backups** network share to Macs as a destination for Time Machine backups.

### Creating Backup User

- Let’s start by creating a separate user for storing backups. I call mine **keeper**:

```
$ sudo adduser --disabled-password --gecos "" keeper
```

- Let’s also create a subdirectory for Samba sharing:

```
$ sudo mkdir /mnt/backups/backups
```

- Then set the ownership for the **backups** partition and everything below to the newly created user **keeper**:

```
$ sudo chown -R keeper: /mnt/backups
```

- And install samba and avahi packages:

```
$ sudo apt install samba avahi-daemon -y
```

### Configuring Samba

In order to make our Mac see the external hard drive attached to the Raspberry Pi over the network, we will add a new share definition for **/mnt/backups/backups** to our Samba configuration in /etc/samba/smb.conf. Here is how the definition looks:

## [backups]

```
comment = Backups
path = /mnt/backups/backups
valid users = keeper
read only = no
vfs objects = catia fruit streams_xattr
fruit:time machine = yes
```

- And this is the command line to achieve that:

```
$ echo -e '\n[backups]\n\tcomment = Backups\n\tpath = /mnt/backups/backups\n\tvalid users =\n\tkeeper\n\tread only = no\n\tvfs objects = catia fruit streams_xattr\n\tfruit:time machine = yes' | sudo\ntee -a /etc/samba/smb.conf
```

- The default /etc/samba/smb.conf has some fluff like definitions for sharing home directories and printers that we really have no use for. To make it clean and nice:

```
$ sudo nano /etc/samba/smb.conf
```

- Scroll down to **Share Definitions** section and comment out [homes], [printers] and [print\$]. Fig 6 and Fig 7 show the end result. Look for lines starting with double semi-column, they indicate the lines needing touches. Then press ^x to save and exit.

```
pi@raspberrypi: ~ -- ssh pi@raspberrypi.local -- 96x30
GNU nano 5.4                               /etc/samba/smb.conf *
=====
# Share Definitions
=====
;; [homes]
;;   comment = Home Directories
;;   browsable = no

# By default, the home directories are exported read-only. Change the
# next parameter to 'no' if you want to be able to write to them.
;;   read only = yes

# File creation mask is set to 0700 for security reasons. If you want to
# create files with group=rw permissions, set next parameter to 0775.
;;   create mask = 0700

# Directory creation mask is set to 0700 for security reasons. If you want to
# create dirs. with group=rw permissions, set next parameter to 0775.
;;   directory mask = 0700

# By default, \\server\username shares can be connected to by anyone
# with access to the samba server.
# The following parameter makes sure that only "username" can connect
# to \\server\username
# This might need tweaking when using external authentication schemes
;;   valid users = %S

# Un-comment the following and create the netlogon directory for Domain Logons

*G Help      *W Write Out    *M Where Is     *K Cut        *E Execute    *L Location
*X Exit      *R Read File    *A Replace     *U Paste      *J Justify    *O Go To Line
```

Fig 6. Editing smb.conf. 1 of 2

```
pi@raspberrypi: ~ -- ssh pi@raspberrypi.local -- 96x30
GNU nano 5.4                               /etc/samba/smb.conf
=====
# [homes]
;;   comment = All Printers
;;   browsable = no
;;   path = /var/spool/samba
;;   printable = yes
;;   guest ok = no
;;   read only = yes
;;   create mask = 0700

# Windows clients look for this share name as a source of downloadable
# printer drivers
;;[print$]
;;   comment = Printer Drivers
;;   path = /var/lib/samba/printers
;;   browsable = yes
;;   read only = yes
;;   guest ok = no
# Uncomment to allow remote administration of Windows print drivers.
# You may need to replace 'lpadmin' with the name of the group your
# admin users are members of.
# Please note that you also need to set appropriate Unix permissions
# to the drivers directory for these users to have write rights in it
;   write list = root, @lpadmin

[backups]
*G Help      *W Write Out    *M Where Is     *K Cut        *E Execute    *L Location
*X Exit      *R Read File    *A Replace     *U Paste      *J Justify    *O Go To Line
```

Fig 7. Editing smb.conf. 2 of 2

- In order to connect to the SMB share as user **keeper**, we need to explicitly add user **keeper** to Samba's password file and set a password with `smbpasswd`:

```
$ sudo smbpasswd -a keeper
```

- A quick way to check your Samba configuration is by executing

```
$ sudo testparm -s
```

Make sure that your output looks exactly like the one in Fig 8.

```
pi@raspberrypi: ~ -- ssh pi@raspberrypi.local — 96x30
Loaded services file OK.
Weak crypto is allowed
Server role: ROLE_STANDALONE

# Global parameters
[global]
    log file = /var/log/samba/log.%m
    logging = file
    map to guest = Bad User
    max log size = 1000
    obey pam restrictions = Yes
    pam password change = Yes
    panic action = /usr/share/samba/panic-action %d
    passwd chat = *Enter\snew\ss*\spassword:*\n\n *Retype\snew\ss*\spassword:*\n\n *password
\supdated\ssuccessfully* .
    passwd program = /usr/bin/passwd %u
    server role = standalone server
    unix password sync = Yes
    usershare allow guests = Yes
    idmap config * : backend = tdb

[backups]
    comment = Backups
    path = /mnt/backups/backups
    read only = No
    valid users = keeper
    vfs objects = catia fruit streams_xattr
    fruit:time machine = yes
pi@raspberrypi:~ $
```

Fig 8. Samba Config Check

- Finally, reload Samba service to pick up the changes:

```
$ sudo service smbd reload
```

## Configuring Avahi (And Making It Pretty)

We will use Avahi to advertise the Samba share to Macs via zero-configuration networking and service discovery protocols. When built with mDNS support, Samba can do it on its own. Unfortunately, the build of Samba

currently available for Raspberry Pi OS Lite does not have this functionality. Avahi will help us to make Macs aware that:

- Raspberry Pi runs SMB server on port 445.
- There is an SMB share point available for Time Machine backups.

As a bonus (this is the pretty part), we are also going to advertise our Raspberry Pi as an Xserve so that it appears sporting Xserve graphics in the Finder.

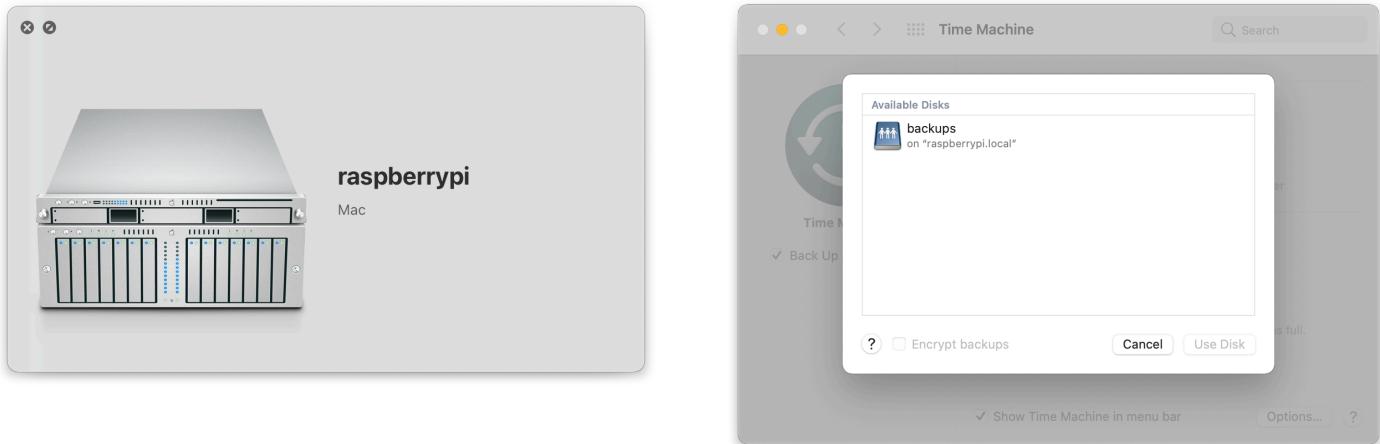
We are going to create file /etc/avahi/services/samba.service and populate it with the following XML:

```
<?xml version="1.0" standalone='no'?><!--*-nxml-*-->
<!DOCTYPE service-group SYSTEM "avahi-service.dtd">
<service-group>
  <name replace-wildcards="yes">%h</name>
  <service>
    <type>_smb._tcp</type>
    <port>445</port>
  </service>
  <service>
    <type>_device-info._tcp</type>
    <port>9</port>
    <txt-record>model=Xserve1,1</txt-record>
  </service>
  <service>
    <type>_adisk._tcp</type>
    <port>9</port>
    <txt-record>dk0=adVN=backups,adVF=0x82</txt-record>
    <txt-record>sys=adVF=0x100</txt-record>
  </service>
</service-group>
```

- Invoke nano text editor:
- ```
$ sudo nano /etc/avahi/services/samba.service
```
- Paste the above XML then ^x to save and exit.
  - Restart Avahi
- ```
$ sudo service avahi-daemon restart
```

## Testing It All

At this point, using your Mac, you should be able to see **raspberrypi** listed in Network section of the Finder. Next, open **Time Machine** from **System Preferences...**, click on **Select Disk...** and confirm that you see **backups on raspberrypi.local** as an option. Do not configure your Mac's Time Machine just yet though.



## Securing the Raspberry Pi

Let's address that security warning on the Raspberry Pi's greeting screen now.

- From SSH console run:
- ```
$ sudo raspi-config
```
- Navigate to **1 System Options -> S3 Password** to change the default password for user **pi**

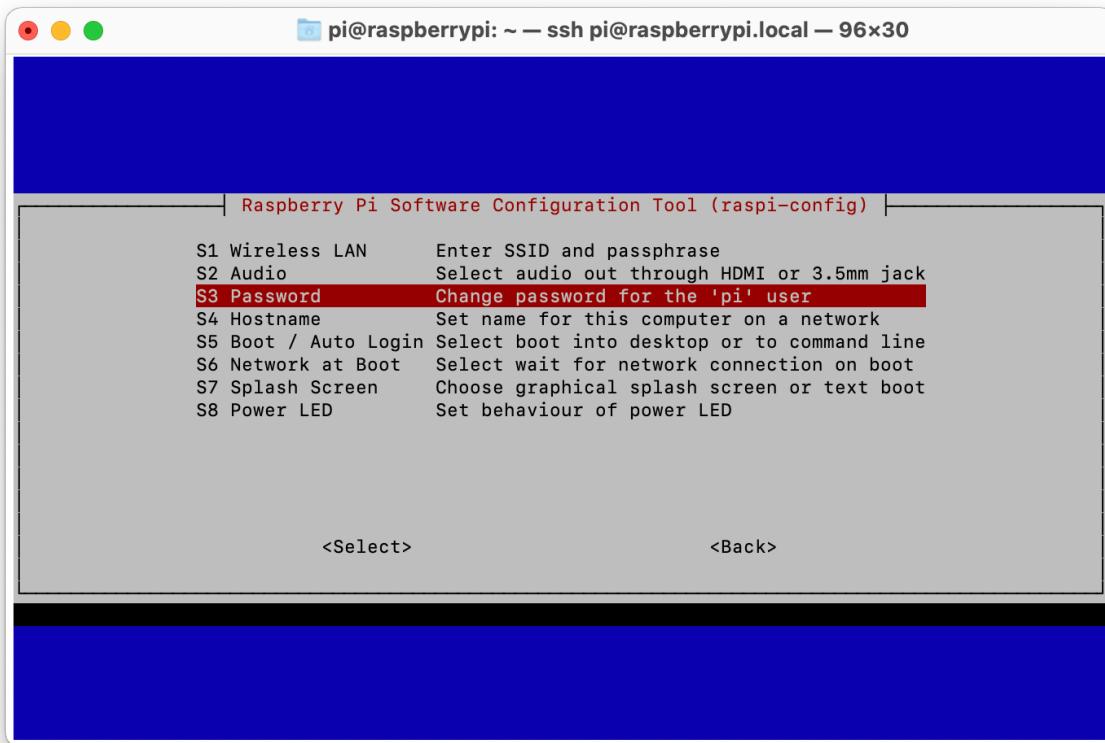


Fig 9. Changing System Password

## Renaming the Raspberry Pi

- From SSH console run:

```
$ sudo raspi-config
```

- Navigate to **1 System Options** -> **S4 Hostname**.
- Enter a new hostname to your liking. I call my Raspberry Pi **keeper**, so this is what you are going to see on the screenshots that follow.
- Click Tab key to select **Ok**,
- Click Tab key twice to **Finish**.
- Choose **Yes** to reboot.

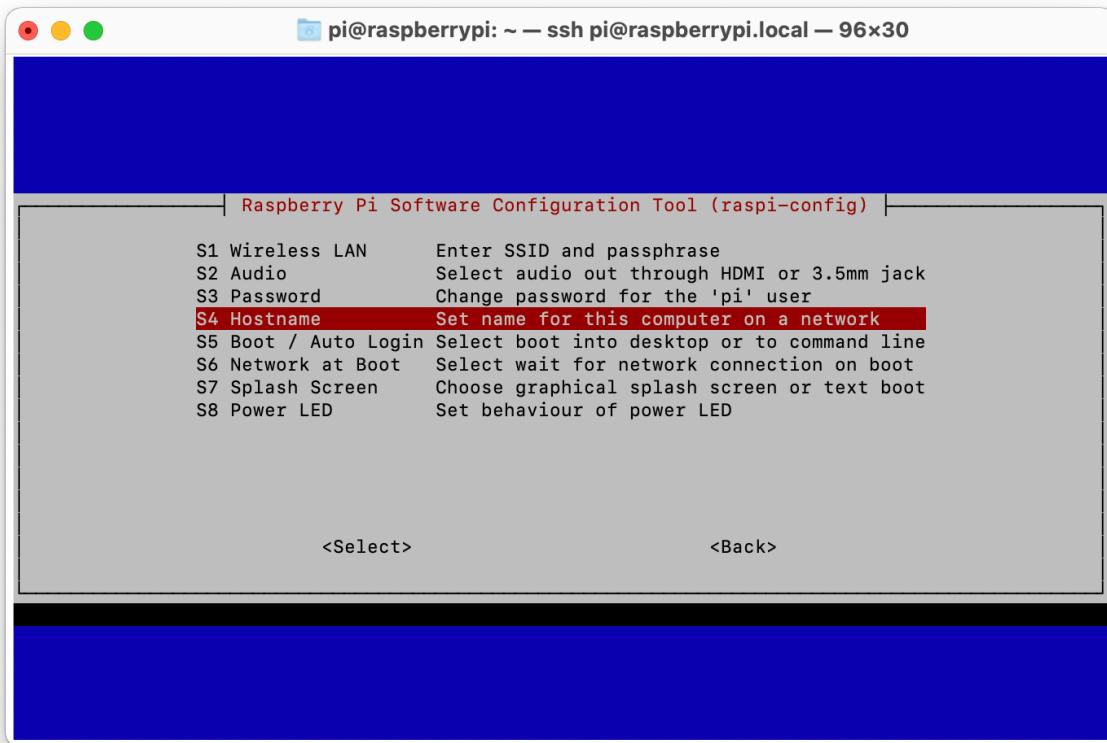
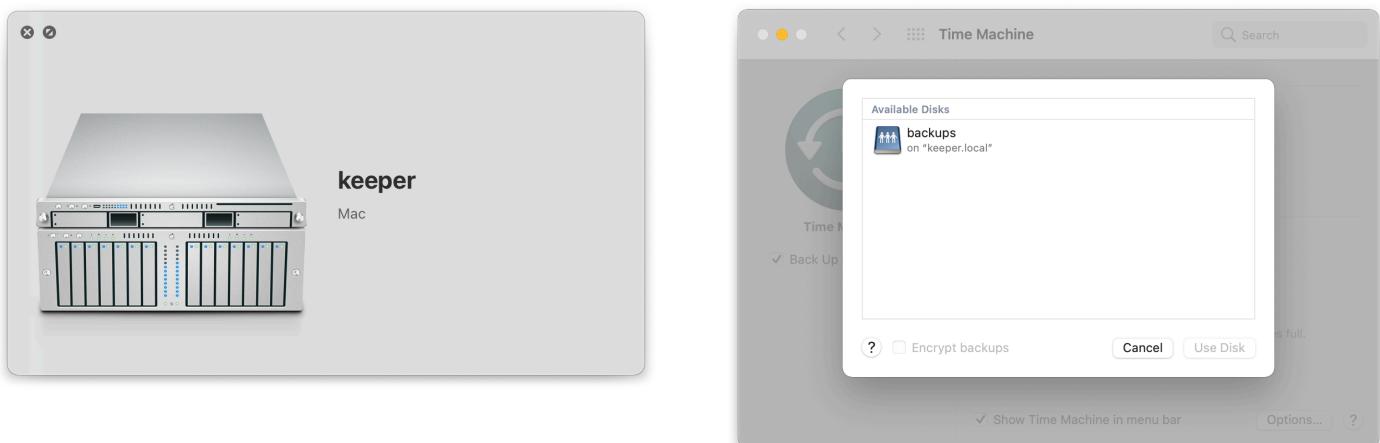


Fig 10. Changing System Name

Once your Raspberry Pi comes back online, it should be accessible by its new name - in my case:

```
$ ssh pi@keeper.local
```

Your Mac's Finder should also reflect the new name.



Your Raspberry Pi is now ready to serve as a network destination for Macs' Time Machine backups. On your Mac, go to **System Preferences...** → **Time Machine**, select **Select Disk...** and pick your Raspberry Pi from the

list. You will need to provide credentials for the Samba user. One more thing: I highly recommend checking **Encrypt backups** option.

Keep it Safe. Back Up!

Backup • Raspberry Pi • macOS • Time Machine



Oleg Ovechkin

<https://ovechkin.xyz>

Comments (40)

Newest First

Preview

Post Comment...



**Jamie Pinheiro** 2 years ago · 0 Likes

Great tutorial, worked well, thank you!



**Yeechang Lee** 2 years ago · 1 Like

My thanks for the guide. One suggestion: Add `nofail` to the `fstab` line, so that the system will still boot if the external drive is not available.



**Oleg Ovechkin** 2 years ago · 0 Likes

Yeechang,

I will update the article to reflect your suggestion.

Thank you!



**Piwero** 2 years ago · 0 Likes

Hello, thanks for this tutorial. I can see the Rasberry Pi from my Mac but when trying to `Add backup disk` I keep asking for login to the server. I've entered correctly, but it seems that I try to do something, then it asks for the login again and so on, I can enter it dozens of times and it keeps requiring login and doesn't create a backup disk. What should I do? Thanks



**Oleg Ovechkin** 2 years ago · 0 Likes

Hi Piwero,

Sorry for your trouble. Your case sounds like a classic password mismatch. I suggest you try the following:

- SSH to your Raspberry Pi as user `pi`;
- as user `pi`, execute `sudo smbpasswd keeper` and re-enter the samba user `keeper` password at the prompt;

That should solve the authentication issues you experience.



**David Lambl** 2 years ago · 0 Likes

Thanks for this! Love how you explained what all of the commands are doing.



**Paul** 2 years ago · 0 Likes

This is such a great walkthrough....I've done it twice now and flawless each time...Many thanks.



**Russell** 2 years ago · 0 Likes

Are there any issues with using the ext4 filesystem for this? For instance, if I wanted to restore a new Mac from a backup which used the ext4 filesystem, would it work?

Presumably the workflow would have to be New Mac (APFS) <-> Router <-> Raspberry Pi <-> Backup Drive (ext4).

I do not think it would be possible to restore files directly, i.e. New Mac (APFS) <-> Backup Drive(ext4)

In other words, I would need the Raspberry Pi in order to access the backups on a Mac?

Many thanks,



**Oleg Ovechkin** 2 years ago · 0 Likes

Hi Russell,

Your assumptions are correct. You won't be able to unplug an ext4-formatted disk from a Raspberry Pi, connect it to your Mac, and expect it to work. In this regard, the described setup is not unique, though. For instance, it would be wrong to assume that one can pull a drive from a NAS and make it work connected directly to a client Mac or Windows box.

Having said that, I did restore Macs on two occasions from Time Machine backups residing on my Raspberry Pi over the network without any issues.

I hope this clarifies 'Y'



**Russell** 2 years ago · 0 Likes

Thank you for clarifying. Hypothetically if I were restoring to a brand new mac using the Apple Migration Assistant, would this pickup the Raspberry Pi backup from the network, and allow me to restore my Mac?

I have successfully followed your tutorial and implemented this today - really easy to follow. I tried to configure Avahi to show the Raspberry Pi as a 'Time Capsule', however I can't see this. It just looks as though it's any other computer. Not the end of the world but the TimeCapsule icon would have been cool.

Thanks again for the tutorial!



**Oleg Ovechkin** 2 years ago · 0 Likes

Hi Russell,

I am glad to hear that everything worked for you!

Regarding your first question... Yes, as I mentioned earlier, I did precisely that using Migration Assistant and restoring from Raspberry Pi twice in the past. I can not guarantee this method will continue to work with every upcoming version of macOS in the future. But it worked flawlessly with macOS 11 (aka Big Sur).

For the second... Try to clear your Mac's DNS cache:

```
sudo dscacheutil -flushcache  
then  
sudo killall -HUP mDNSResponder
```

from the Terminal. Or go ahead and reboot. I have been noticing the same on a few occasions. I suspect that has to do with a Mac "remembering" your Raspberry Pi while it is already configured to run SMB but not Avahi yet. So refreshing Mac's cache should help.

Let me know how it goes!



**Russell** 2 years ago · 0 Likes

Thank you again, I have tried the commands you suggested but this did not resolve anything, but I will try to restart later. My Avahi file looks like:

```
<?xml version="1.0" standalone='no'?><!--*-nxml-*-->
<!DOCTYPE service-group SYSTEM "avahi-service.dtd">
<service-group>
<name replace-wildcards="yes">%h</name>
<service>
<type>_smb._tcp</type>
<port>445</port>
</service>
<service>
<type>_device-info._tcp</type>
<port>9</port>
<txt-record>model=TimeCapsule</txt-record>
</service>
<service>
<type>_adisk._tcp</type>
<port>9</port>
<txt-record>dk0=adVN=backups,adVF=0x82</txt-record>
<txt-record>sys=adVF=0x100</txt-record>
</service>
</service-group>
```

The first backup took about 2 hours to complete, backing up 125gb (encrypted) to an external HDD connected to Raspberry Pi4. Does that sound about right to you? Additionally browsing through files in Time Machine seems fairly slow.

I have also experienced it where the Mac randomly disconnects (not during the backup thankfully, but after a period of idle time). Have you experienced this?



**Oleg Ovechkin** 2 years ago · 0 Likes

Russell,

Your Avahi file looks fine. I tested it, and it works as expected with my Raspberry Pi and a Mac running Big Sur.

Indeed, your initial backup time looks longish - your calculated average throughput was only 17MB/sec. Check your network to make sure it is not the bottleneck here. In my case, I keep the Raspberry Pi plugged in directly to my router with an ethernet cable. My Macs talk to a gen 6, 5GHz Wi-Fi access point. I, too, always encrypt my backups. My transfer data rates are around 35MB/sec. I do not experience random disconnects, either.

To reiterate, I suggest you start looking into the network. All of your remaining issues are most likely caused by its flakiness. If you can, connect all your devices via wired ethernet and see how Time Machine backup and history browsing perform with no Wi-Fi in the picture.



**Russell** 2 years ago · 0 Likes

---

A further update on this. Restarting my Mac running Big Sur resolved the icon issue - the RPi4 is now displaying as a Time Capsule.

Regarding the performance, I may have improved this. It appears that when I created my smb.conf file, it also created a config file in my home directory that was just called 'Share'. I deleted this file, and performance has improved slightly (difficult to judge though because I haven't done a full system backup since).

I'd still be keen to run any diagnostics on the disc, if you can think of anything? Having done some research it seems that poor samba performance is quite consistently reported and there may be settings that can be configured to improve performance.



**Russell** 2 years ago · 0 Likes

Thanks, I'm yet to try to restart but hopefully that gives me the TimeCapsule icon.

The Raspberry Pi is connected to the router directly using an ethernet cable, and the Mac's are connected over 5GHz Wi-Fi. Also I should add that the performance was better backing up from my M2 MacBook Air than my Intel Based MacBook Pro.

I feel like the slow speeds are something to do with the write speed of the HDD. Are you using a HDD or a SSD? Can you think of any checks I can perform on the disc?



**LG** 2 years ago · 0 Likes

Big THX

GREAT tutorial, worked on the first try!



**Martin** 2 years ago · 0 Likes

Which CanaKit would you recommend for this guide?



**Oleg Ovechkin** 2 years ago · 0 Likes

I am partial to CanaKits because of two reasons:

1. CanaKit sells SKUs with passively cooling aluminum cases - they call those "self-cooling." The fewer moving parts, the better, in my book.
2. CanaKit includes higher quality and specs power supplies.

This guide will work fine with any Raspberry Pi having 2GB of memory or more.

Here are a couple of the links for you:

- <https://www.canakit.com/raspberry-pi-4-extreme-aluminum-case-kit.html>
- <https://www.canakit.com/raspberry-pi-4-starter-max-aluminum-kit.html>

Thank you,

Oleg.



**Martin** 2 years ago · 0 Likes

Could you do this with 2 hard drives? How would the instructions change? Also I would literally donate \$ for you to support you for providing an answer to that question :) This article's quality is bomb!



**Oleg Ovechkin** 2 years ago · 0 Likes

Hi Martin,

Yes, since Raspberry Pi runs a Linux variant, one can do it with two or more hard drives. But before answering your question, can you elaborate on what you are trying to achieve or what issues you are trying to work around/solve?

Also, please keep in mind that Raspberry Pi hardware - USB specifically - has a serious limitation. According to the official datasheet (<https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf>) "Downstream USB current is limited to approximately 1.1A in aggregate over the four sockets." In practice, that means that unless the hard drives are independently powered, Raspberry Pi can reliably work with only a single attached external device.



**Michael** 3 years ago · 0 Likes

I guess that's because my hard drive is connected via USB and therefore this command doesn't work.  
Again: Thank you and greetings from Germany!

**LG** 2 years ago · 0 Likes

<https://forums.raspberrypi.com/viewtopic.php?f=28&t=245931>

works 4 me

---

**Michel** 3 years ago · 0 Likes

Hi,

I got an issue with connecting. I see the pi keeper in my network, authentication works but it remains in connecting state. It looks like there is a connection issue somewhere. Do you have an idea?

---

**Miguel** 3 years ago · 0 Likes

Hi Michel,

try allowing the port 445 to "allowed" in the Raspberry Pi firewall configuration. If you have ufw, the following should do the trick: sudo ufw allow 445.

I had the exact same problem and banged my head against the wall for hours. After opening the port and making sure the samba-users had a proper password setup, no problem connecting to the shared drive in time machine nor in Finder.

With kind regards

Miguel

---

**Felipe** 3 years ago · 0 Likes

Hi Oleg, I want to try this one out... I have one external hard drive that I already have 3 partitions, of which one is used for time machine backups directly on my Mac. I'm wondering if it's possible to use the same hard drive, and start doing the backups through the network using the raspberrypi (without

losing my existing backups, and using the same partition). I would like to present a second partition as a storage for a HomeNAS solution.

What do you think? Should that be possible? Or I'll have to start the backups from scratch on a new drive?



**Oleg Ovechkin** 3 years ago · 0 Likes

Hi Felipe! Sorry for the delayed response. You can definitely try that - I see no reason why it would not work. There is the complexity associated with the plan thought. To list just a few points off the top of my head:

1. Based on what you said regarding your hard drive being connected directly to a Mac, I assume you will need to re-format and re-partition it to make it work with Raspberry Pi.
2. In order to continue using the same Mac backup(s), you will need to convert your existing backup to sparse bundle format. As far as I am aware, Mac does not use sparse bundle containers in case of directly connected backup drives. macOS's hdiutil converts between different formats and containers. Having said that, be aware that hdiutil will not help with Time Machine specific metadata (I am referring to com.apple.TimeMachine.MachineID.plist and com.apple.TimeMachine.MachineID.bckup files) necessary for a sparse bundle to be recognized as a valid TM backup - you will need to figure out that part on your own.
3. Whatever will be found while you are trying things out.

As with any experimentation, if you decide to follow along this path, please make sure the existing data is backed up before proceeding!

Good luck,  
Oleg.



**P.** 3 years ago · 0 Likes

Thank you so much, finally a working guide!



**Zach** 3 years ago · 0 Likes

This is an excellent guide. Thank you so much for writing it up. The web is full of outdated or outright incorrect guides (using things like netatalk & hfsprogs (without journalling support!).

I applied it mostly without modification. I did want to share a few things I did differently in case its helpful to anyone

- If you use Ubuntu, Samba has mDNS built in, so there's no need to customize Avahi (I *think* you still need it installed?)

- With Samba's mDNS support you can add the following to [global]:

- To get your Xserve icon:

```
# How should this server appear in macOS finder fruit:model =
Xserve
```

- To make the server name not show up in SHOUTING CASE:

```
# set netbios name to match the case of the hostname rather
than being in all capitals mdns name = mdns
```

Anyway, thank you again for such thorough, well-written guide.



**Oleg Ovechkin** 3 years ago · 0 Likes

Thank you for your kind words, Zach!



**Frank Perkins** 3 years ago · 0 Likes

This has been running great, but I believe after updating to Monterey 12.4, it caused it to stop working. I can still see the piNas in Finder, but when I try to backup it says: Backup Not Completed: Time Machine couldn't backup to 'pi name'

Any ideas? I can telnet to port 445 on the NAS device, I can connect via Finder and create folders. Did something in v12.4 break the compatibility?



**Oleg Ovechkin** 3 years ago · 0 Likes

Hi Frank! Sorry, something got broken with your setup. I can confirm that if a Raspberry Pi is configured as described here, Time Machine continues to work correctly with macOS Monterey

## 12.4.

I recently upgraded one of my Macs from Big Sur 11.6.7 to Monterey 12.4 without any issues. Based on the description, your configuration appears customized, and without knowing the specific details, I am not sure I can be effective in helping you to track down the issue. Having said that, it sounds like your Time Machine sparse bundle (the one on your NAS) might have become corrupted. To verify that hypothesis, try to copy your existing sparse bundle somewhere else or simply rename its top-level directory. Then initiate a new Time Machine backup. If everything goes well on this fresh backup run, then your previous sparse bundle went south, and you will need to look into ways to repair it. Otherwise, it should give you a clue to look / investigate somewhere else.

Good luck!



**Frank Perkins** 3 years ago · 1 Like

Thanks for the response Oleg. You were right, I had something corrupted on my sparse bundle. I just deleted it and it seems to work now. Glad it wasn't a bigger issue! Appreciate the help.



**Nicolás Georger** 3 years ago · 1 Like

Hi Oleg! Thank you very much, I've made only few mods into smb config in order to meet my needs, but the whole is working amazing on a Raspberry Pi 4 and my Macbook Pro M1 with Monterey. Regards from Chile.



**Sasha** 3 years ago · 0 Likes

Hi Oleg, thanks... For some reason I cannot log in via Mac with user Pi and my known password used for SSH (it works fine to run as sudo as well).



**Oleg Ovechkin** 3 years ago · 0 Likes

Hi Sasha,

It sounds like you have access to your Raspberry Pi via the local console. So first, make sure that SSH is indeed enabled. From your Raspberry Pi console, execute:

```
$ sudo raspi-config
```

and from its interface navigate to "3 Interface Options -> I2 SSH -> <Yes>"

You might need to reboot at this point.

Second, make sure you use the correct IP Address to connect from your Mac. For that, on Raspberry Pi run:

```
$ ifconfig
```

and note the address following "inet" for the interface "eth0" That should read something like 192.168.0.xxx

Then from your Mac:

```
$ ssh pi@192.168.0.xxx
```

Hope this helps!



**Frank** 3 years ago · 1 Like

This guide worked perfectly. Thank you so much for sharing!



**steph** 3 years ago · 0 Likes

actually correction, looks like files over 1 GB have issues but 500MB for instance seems to work just fine, cannot get what the difference can be.



**steph** 3 years ago · 0 Likes

hi, thx a lot for that tutorial! just installed my PI and samba works but somehow, downloading from it like movie files get stuck while writing seems ok, very strange, any idea?



**Albert Camacho** 3 years ago · 1 Like

Great tutorial, I followed step by step and everything is working smoothly, I have one question, what are the steps to add another HDD TO backup another Mac in my network? Thanks



**Oleg Ovechkin** 3 years ago · 0 Likes

Hi Albert! I am glad the steps worked for you 😊

You do NOT need another HDD. Your Raspberry Pi can host many backups from several Macs simultaneously as long as the HDD you already have attached is spacious enough. Simply go to your other Mac and repeat the last steps in the article: open "System Preferences... -> Time Machine, select Select Disk... and pick your Raspberry Pi from the list."

I successfully backup three Macs to the same Pi.

Implementing 3-2-1 Backup  
Strategy with Raspberry Pi for  
Time Machine and Sync to  
Cloud

Rethinking Camera Support

Copyright 2021-2024 Oleg Ovechkin.

All Rights Reserved.