[Tech,Knowledge].toStream()

# Beginner's Guide to Installing Node.js on a Raspberry Pi

> *This article has been updated to cover the installation of both Rasperry Pi OS Bullseye and the latest version of Node at the time of this writing which is Node 20.x.*

In this installment of my LTM (Learning through Making) series of Node.js tutorials, I will show you how to get Node.js up and running on a Raspberry Pi 4 (or Pi 3, or Pi 2). I'm focusing on the Raspberry Pi 4/ Pi 3 / Pi 2 rather than older versions such as the Raspberry Pi Model B+ since these are the latest models at the time of this writing. The Raspberry Pi 4, for example, sports a 1.5 GHz quad-core ARMv8 chip with RAM options of 8 GB, 4 GB, 2 GB, or 1 GB versus the Raspberry Pi 1 Model B+'s 700 MHz single-core ARMv6 chip with 512 MB RAM.

A majority of this installation guide should still prove useful for other Raspberry Pi systems besides the Pi 4/ Pi 3/ Pi 2; however, the final steps focused on the installation of Node.js will not work for these systems based on the older ARMv6 architecture.

> *If you are interested in creating a slimmed down Raspberry Pi system without a graphical user interface, see my article on creating a lightweight Raspberry Pi with Raspbian Lite.*

This tutorial is useful for anyone wishing to successfully install a Raspberry Pi 4/Pi 3/Pi 2 system, even if they are not interested in Node.js since the Node.js installation happens in the final steps of the tutorial. But why would you not want to install Node.js? 😃 Let's get started!

## Table of Contents


- Hardware Needed
- Write Raspberry Pi OS Image to SD Card
- Prep the Hardware
- Configure the Pi
- Apply Raspberry Pi OS Updates
- Prepare for Remote VNC Connections
- Create Windows File Share on the Pi (Optional)
- Install Node.js
- Continue your Node.js Journey


## Hardware Needed

If you don't currently own a Raspberry Pi, the easiest way to get started is to buy a starter kit such as this one on Amazon:

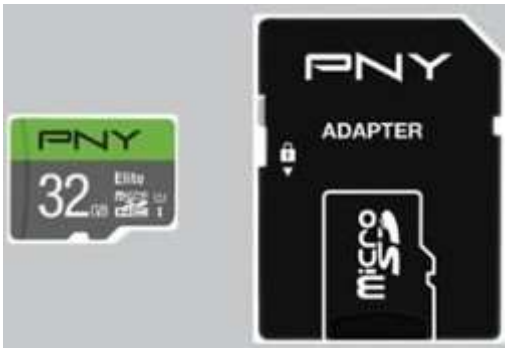
- Vilros Raspberry Pi 4 4GB Complete Desktop Kit (or purchase Vilros Raspberry Pi 4 8GB Basic Starter Kit or CanaKit Raspberry Pi 4 Extreme Kit – 1GB RAM)
  Note: my affiliate info is included on Amazon links so you can, if you desire, support these tutorials while paying nothing extra.
  Other note: due to supply chain issues, it is hard to find Raspberry Pi systems at reasonably low prices. The Raspberry Pi Amazon links may have a higher price than before due to these issues.

You will need the following items to get started:

- Raspberry Pi 4 Model B, Pi 3 Model B+, Pi 3 Model B, or P i 2 Model B (these are the official names for the Raspberry Pi 4, Pi 3, and Pi 2. **Don't let**

the "B" confuse you since there is also an older Raspberry Pi Model B which is now effectively the "Raspberry Pi 1 Model B".)

- MicroSD card (I recommend a class 10 card that is 16 GB or higher. I like this PNY 32GB Elite Class 10 Card).
- MicroSD to SD memory card adapter (so you can use the SD card reader on your laptop/desktop to write to the microSD card. Many microSD card vendors include an SD adapter with the purchase as shown below. You do not need to use an adapter if your computer has a microSD card slot built in. )




- (Pi 4 only) USB C power supply (3A or greater recommended) to power the Pi. For example, this CanaKit 3.5A Raspberry Pi 4 Power Supply.
- (Pi 2 and 3 only) Micro USB power supply (2.5A or greater recommended) to power the Pi. For example, this CanaKit 5V 2.5A Raspberry a Pi 3 B+ Power Supply.
- (Pi 4 only) Micro HDMI to HDMI adapter since the Raspberry Pi 4 doesn't have a full-sized HDMI port. I like this Cable Matters Micro HDMI to HDMI Adapter. This is not needed for the Pi 2 and Pi 3 since these models have full-sized HDMI Ports.
- (Pi 2 only) USB WiFi adapter (or Ethernet cable if preferred. I have tested this Edimax USB WiFi adapter on Raspberry Pi 2 and it works great out of the box.) This is not needed for the Pi 4/Pi 3 since these systems include built-in WiFi.
- Windows laptop/desktop (Linux or Mac works great too, but this tutorial is geared more toward Windows machines so some readers will need to adapt.)

- Monitor with HDMI cable/USB keyboard/mouse (You can borrow these from another system temporarily and then run your Pi in a headless mode as I will explain later.)

Once you have gathered the needed hardware, we are ready to bring our Pi machine to life!

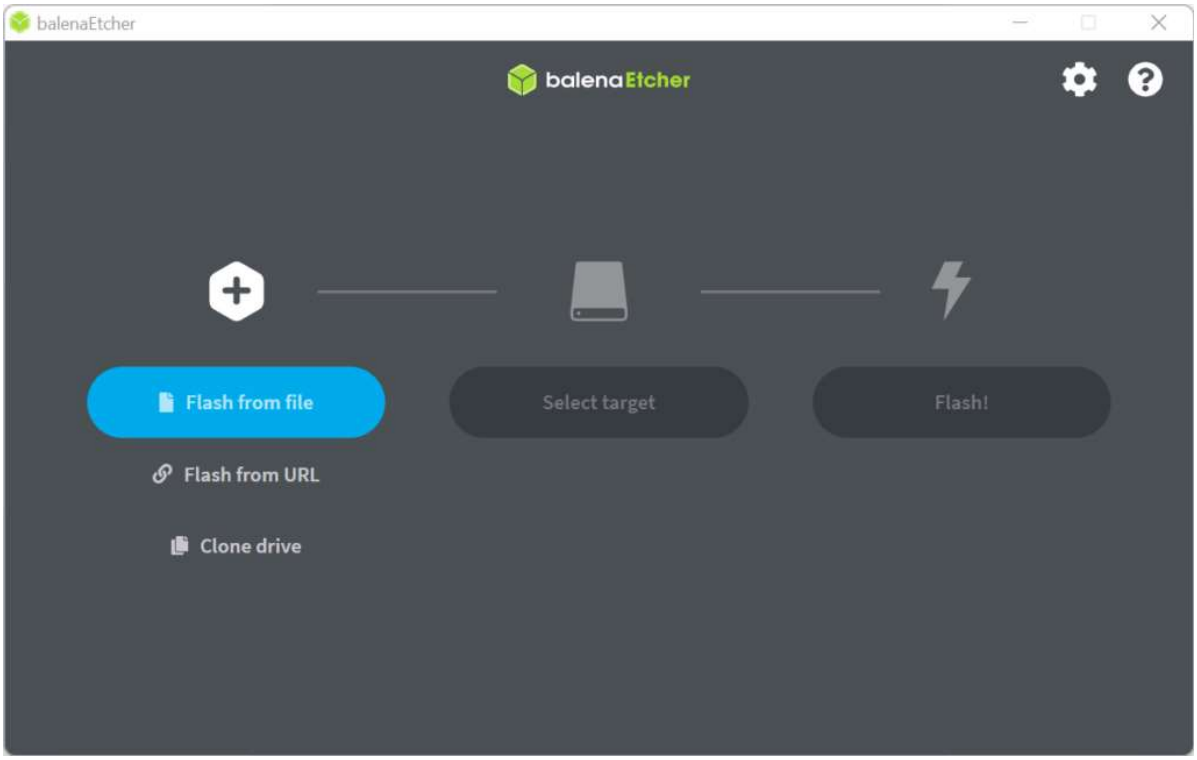
## Write Raspberry Pi OS Image to SD Card

We will be running Raspberry Pi OS which is a free operating system based on Debian Linux and optimized for the Pi. Here we go:

- Download the latest Raspberry Pi OS image from https://www.raspberrypi.com/software/operating-systems. Download the 64-bit option if you are using a Pi 4 or Pi 3. Otherwise, if you are using a Pi 2, download the 32-bit option. Be sure to download one of the desktop versions ("Raspberry Pi OS Bullseye with desktop" or "Raspberry Pi OS Bullseye with desktop and recommended software") rather than the "lite" version. I recommend downloading the smaller version ("Raspberry Pi OS Bullseye with desktop") since you can always add other software packages later if needed. The download for the smaller desktop version is a zip file that is approximately 760 MB for the 64-bit version, and 840 MB for the 32-bit version. If you have a slow connection, it may take some time to download.
- Insert the microSD card into the SD adapter as shown in the Samsung sample picture above.
- Insert the SD adapter in the SD card reader on your laptop/desktop. Make sure it is seated well in the connection.
- Launch Windows Explorer and verify that the SD card registers as a drive on your system. Make a note of the drive letter as you will need it soon.
- Download and install Balena Etcher. Balena Etcher is a cross-platform (Windows, OS X, Linux) tool for flashing images to SD cards. It has many nice features including a simple user interface, the ability to flash an image

without unzipping the zip file, and a validation step to verify the SD card image was written correctly to the SD card.

  - (As an alternative for Windows users, you can use Win32 Disk Imager since it also provides the ability to read an SD card and create an image file which is handy for creating snapshots of Raspberry Pi systems that can flashed back to an SD card later if needed. For Win32 Disk Imager, you will need to first unzip the downloaded zip file before burning the .img file to the appropriate drive letter containing your SD card.)
- Launch Etcher. Be patient as Etcher may take a few seconds to launch. When launched, the interface will look something like this:




- Click the *Flash from file* button and navigate to the Raspberry Pi OS zip file that you downloaded. (It does NOT need to be unzipped before flashing.)
- Click the *Select drive* button to specify the target device location of the SD card.

  WARNING: Be sure you choose the right target device, or you will write the image to another drive on your system, and this will not be good.
- Click the *Flash!* button to write the image to your SD card.

- If you are running Windows, you may be presented with a security prompt regarding making changes to your device. If so, click *Yes* to proceed.
- After the image is written to the SD card and verified, Etcher will automatically dismount your SD card so it can be safely removed.
- Remove the SD card and or adapter from your laptop/desktop and remove the microSD card from the SD adapter.

## Prep the Hardware

See the Raspberry Pi 4 Tech Specs page for a diagram showing the various parts of the Pi 4. Also, you will find a diagram of the Pi 2 here (the Pi 3 looks very similar). These diagrams may help you in this section. Let's get this little SBC (single board computer) ready to boot! Here are the steps:

- Insert the microSD card into the Pi. (Turn the Pi upside down. The microSD card slot is located on the side opposite of the USB ports. This short video explains the process well.)
- Connect the USB keyboard and mouse
- Connect the HDMI cable
- Connect the USB Wi-Fi adapter (or Ethernet connection if you are not using Wi-Fi) This step is not necessary if you are using a Raspberry Pi 4/Pi 3 model which has on-board Wi-Fi.
- Connect the power supply (You will see the Pi lights come on and power up.)
- You should now see Raspberry Pi OS booting on the screen. Hooray!

## Configure the Pi

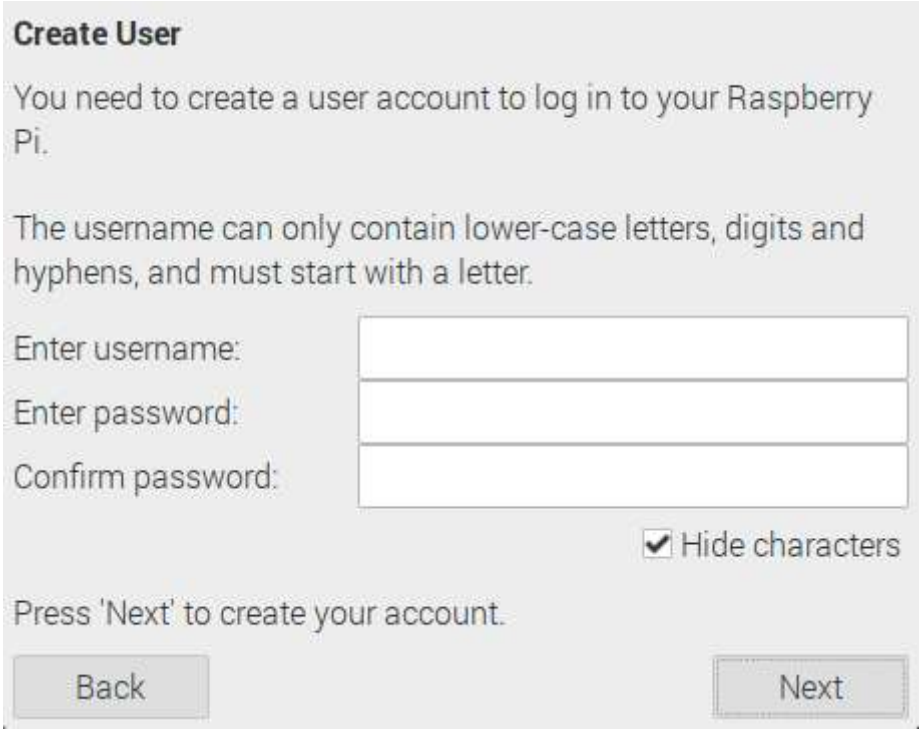
Raspberry Pi OS includes a setup wizard that launches on the first boot, making it easier than ever to set up our Pi. After booting Raspberry Pi OS for the first time, you will be greeted with the following dialog box:

> *Note: If you need to run the setup wizard at a future time (since it only launches on first boot), issue the following command from the terminal (I will show you how to launch the terminal later in the Apply Raspberry Pi OS Updates section of this tutorial):*
>
> ```
> $ sudo -E piwiz
> ```

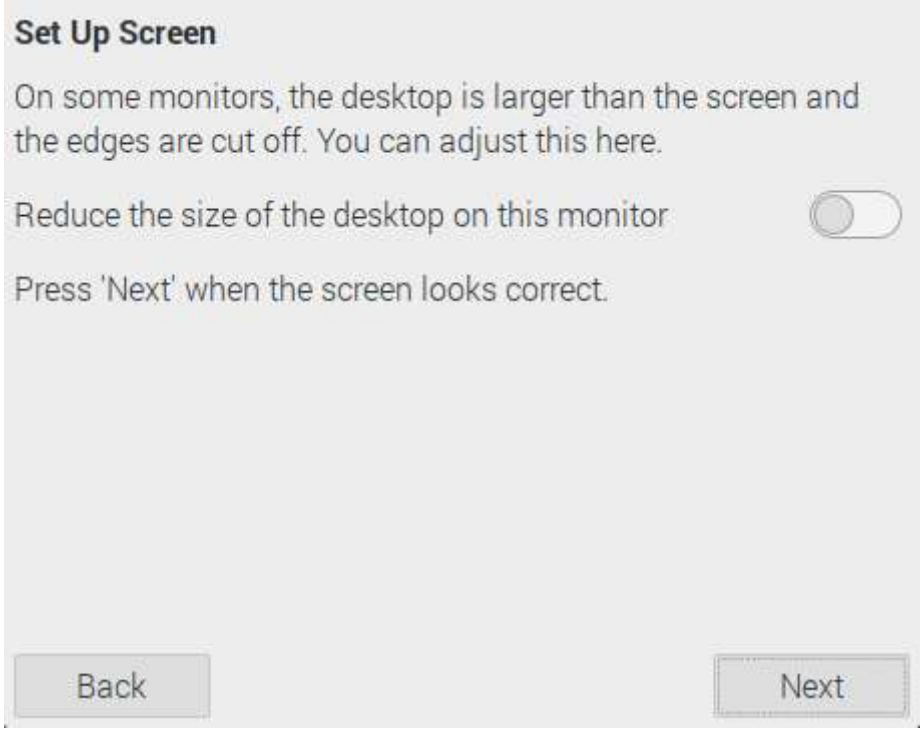
Press the *Next* button. I won't keep telling you to press the Next button, but it is implied as I show each screen in the setup wizard.

Enter your country, language, and timezone to ensure your keyboard and other localization settings work as expected for you:

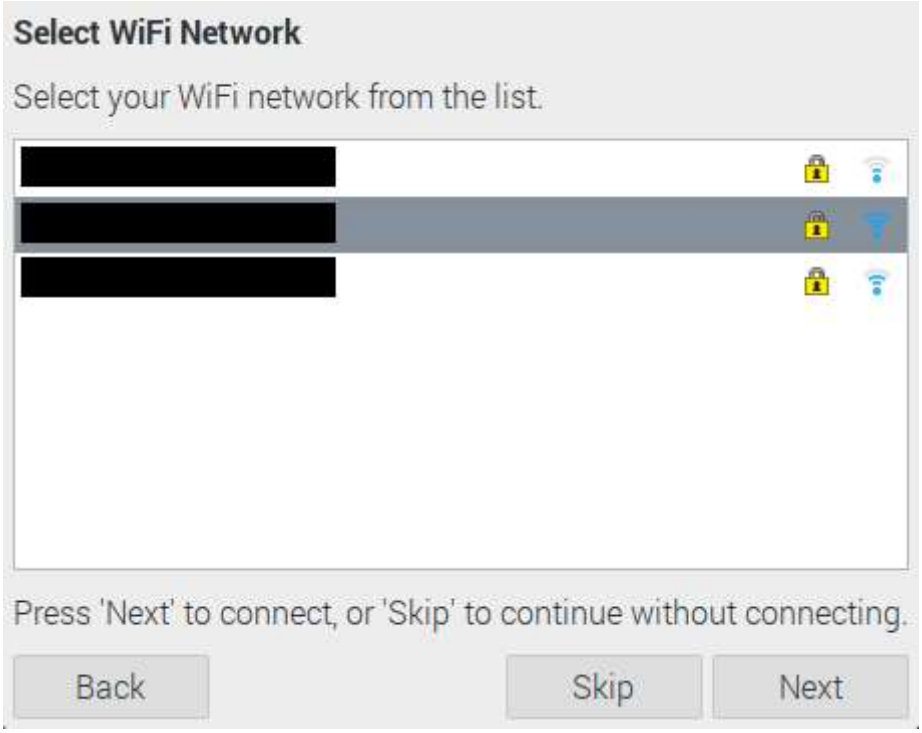
Select a user name to use such as your first name. I do not recommend the user name of “pi” since this was the default user name for many years and it is considered a security risk to use this common name. Also, enter a password. Just make sure you don’t forget it. 😉



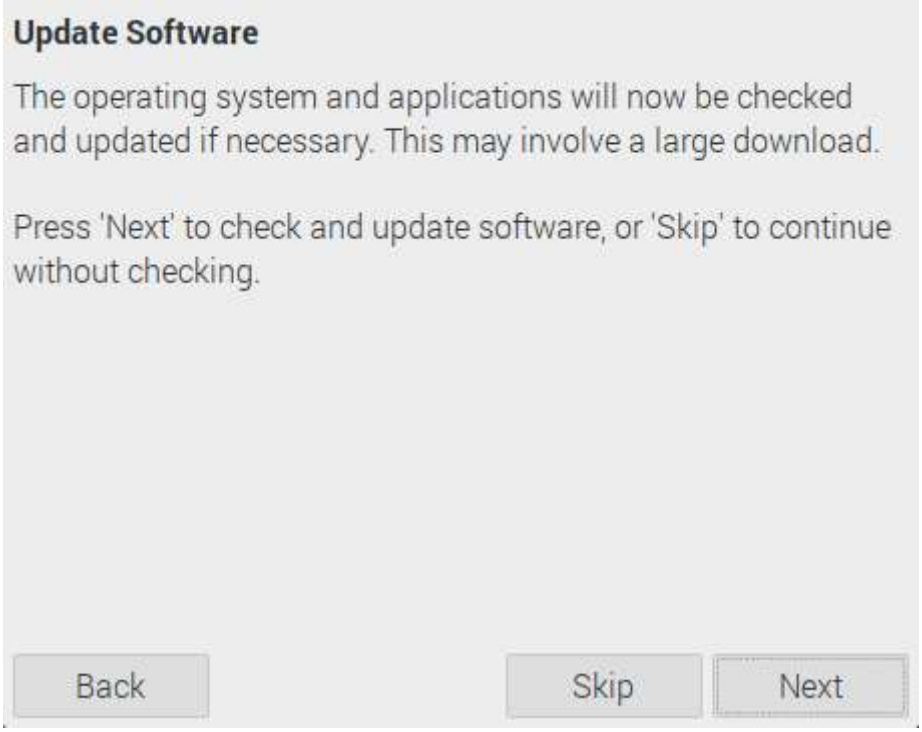
In this tutorial, I'm showing you how to build a "headless" Pi system that you can remote into without a dedicated monitor so just press the *Next* button unless you will be connecting it to a monitor and you have a need to check the box based on observations:




Select your WiFi network from the list.  When you press *Next*, you will be prompted to enter your WiFi network password:




Press the *Next* button to update your software (or see the alternative step immediately below). It is a good idea to ensure you have the latest versions of software including security updates.

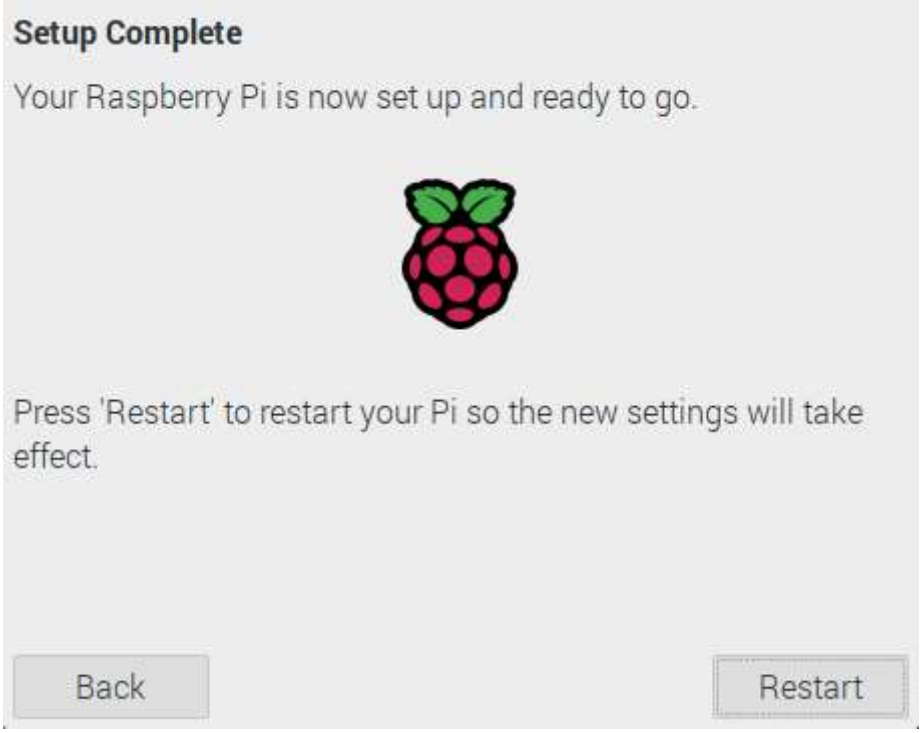


### Alternative step for updating software

The "Update Software" dialog box is handy, but it does not provide detailed feedback (i.e. bytes remaining to download) for one to ascertain how long the update process will take. This is especially problematic for those who have slow Internet connections. As an alternative step for updating software, press the *Skip* button and follow the directions in the Apply Raspberry Pi OS Updates section further down in the tutorial.

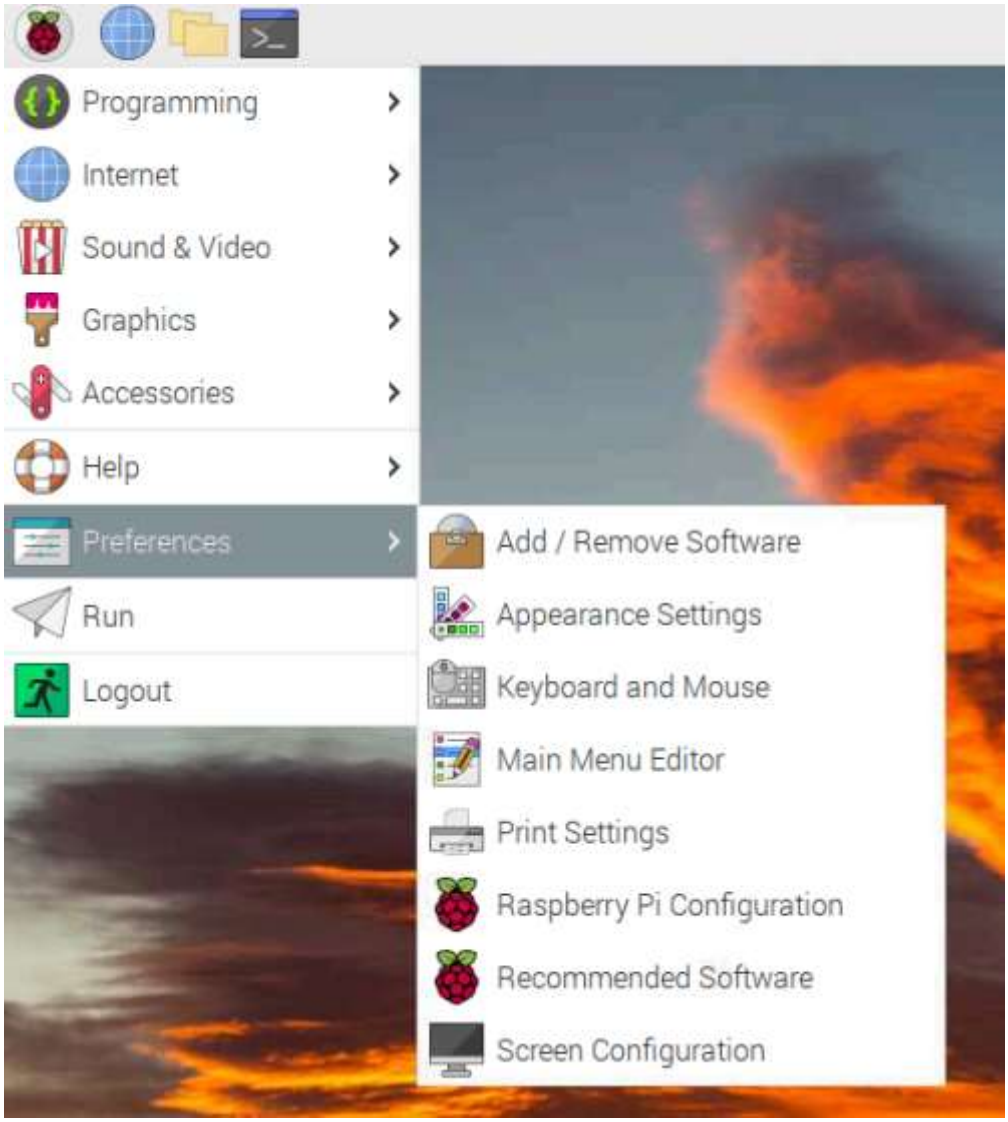
### Back to our regularly scheduled program...

The initial setup is complete. Press the *Restart* button as prompted:

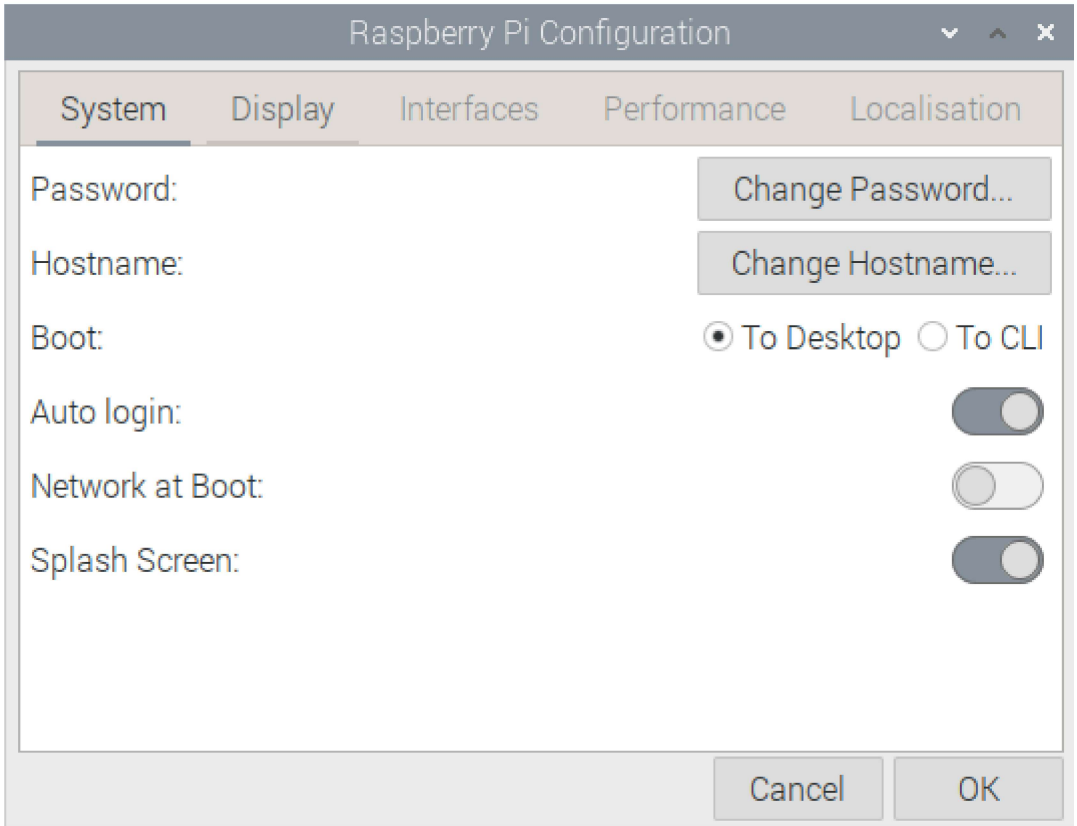



After the system is restarted, we need to make a couple more changes to the Pi for the purpose of personalizing it for our use. To get started, launch

*Raspberry Pi Configuration* which is located in the Menu under *Preferences.*



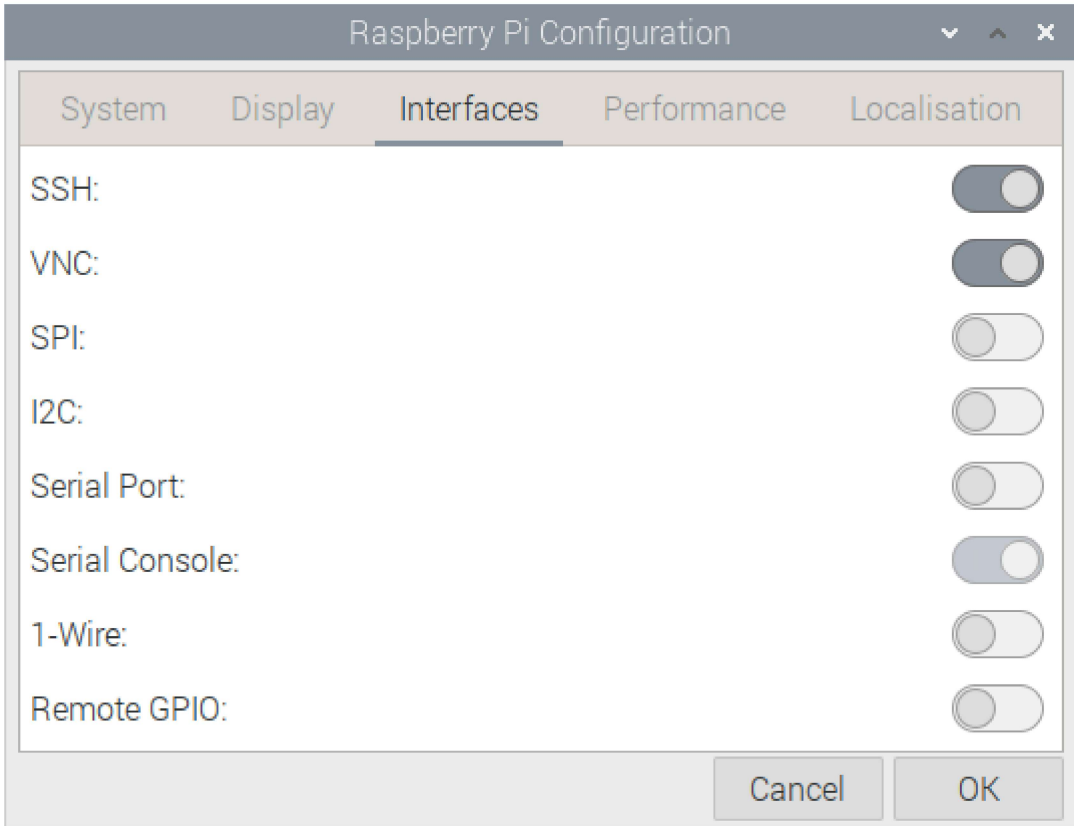

This will launch the Raspberry Pi Configuration utility, a handy graphical version of the console-based *raspi-config* program that was used for many years to configure the Pi (and is actually still used behind the scenes by this graphical version). The following window will appear:




Change the host name from the default of “raspberrypi” to something shorter such as “raspi” or something fun and personal to you by.

Next, select the *Interfaces* tab.

Click the VNC toggle button on the right to enable VNC. A little further down in this tutorial, we will configure the Pi so we can remote into it with VNC. While you are in the Raspberry Pi Configuration tool, you can also enable SSH as another avenue to connect to your Pi:

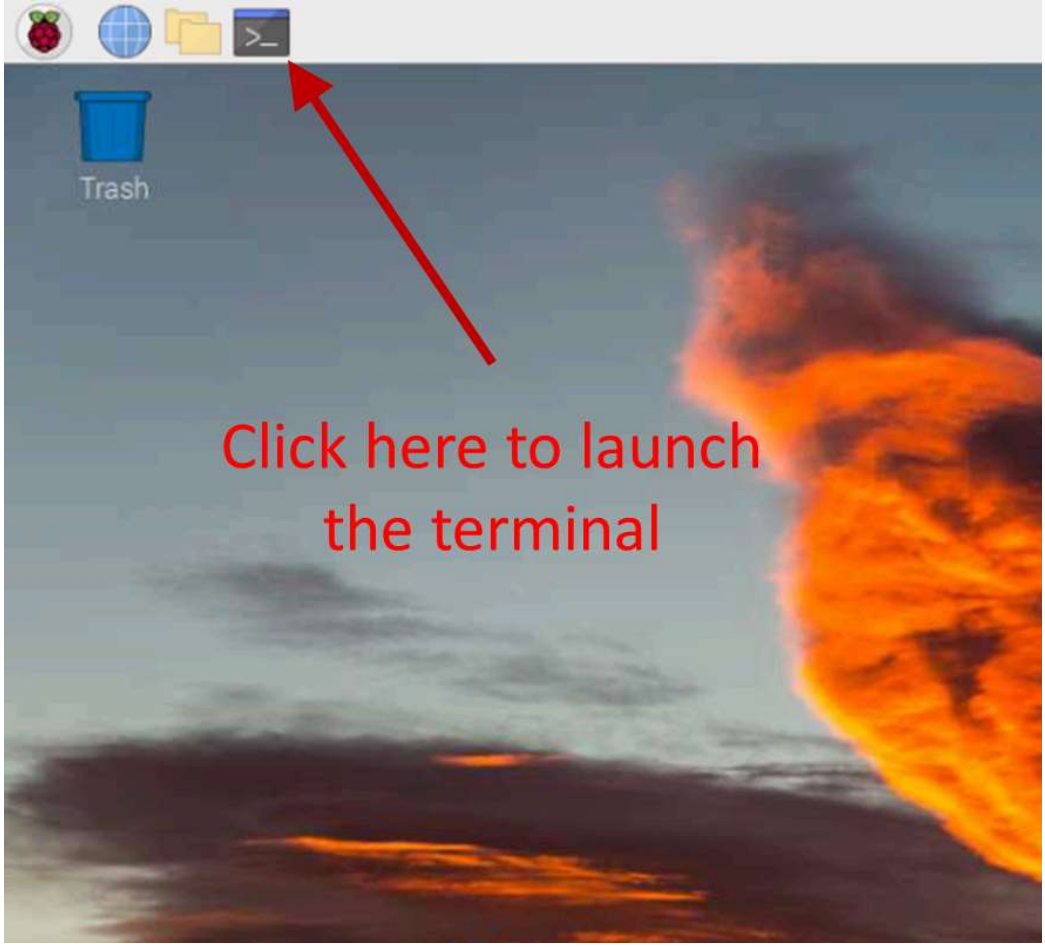



Press the *OK* button and select *Yes* to reboot your Pi when prompted.

## Apply Raspberry Pi OS Updates

If you did not already apply the Raspberry Pi OS updates using the setup wizard above (or you are visiting this tutorial later and want to know how to keep your Pi up to date), follow the instructions in this section.

First, launch the terminal:




From the terminal, issue a ping command to verify your network connection is working:

```
$ ping -c 5 google.com
```

We're now ready to ensure our system is up to date with security patches.

We will first run the apt "update" command.  This command will not actually update any software on the system but will download the latest package lists from the software repositories so that Raspberry Pi OS will be aware of all new software available along with dependencies.  Issue the following command at the "$" prompt:

```
$ sudo apt update
```

Next, run the following command to upgrade any packages installed on your system that need upgrades:

```
$ sudo apt upgrade -y
```

This is important to keep your Pi system synchronized with security updates, etc. These two commands should be issued together and run periodically.

## Prepare for Remote VNC Connections

We want to run our Pi remotely without requiring a dedicated HDMI monitor and USB keyboard/mouse. This is known as headless mode. We will be using the RealVNC server that Raspberry Pi OS ships with to accomplish this goal. To prepare for this, we need to install the samba package so we will be able to access the Pi by its host name from Windows machines on the network rather than by its IP address which can change since the Pi receives its IP address via DHCP. (For OS X users, you can install Bonjour and then access your Raspberry Pi by host name. For example, if your host name is "raspi", you can access it on OS X as "raspi.local".) OK, let's install Samba:

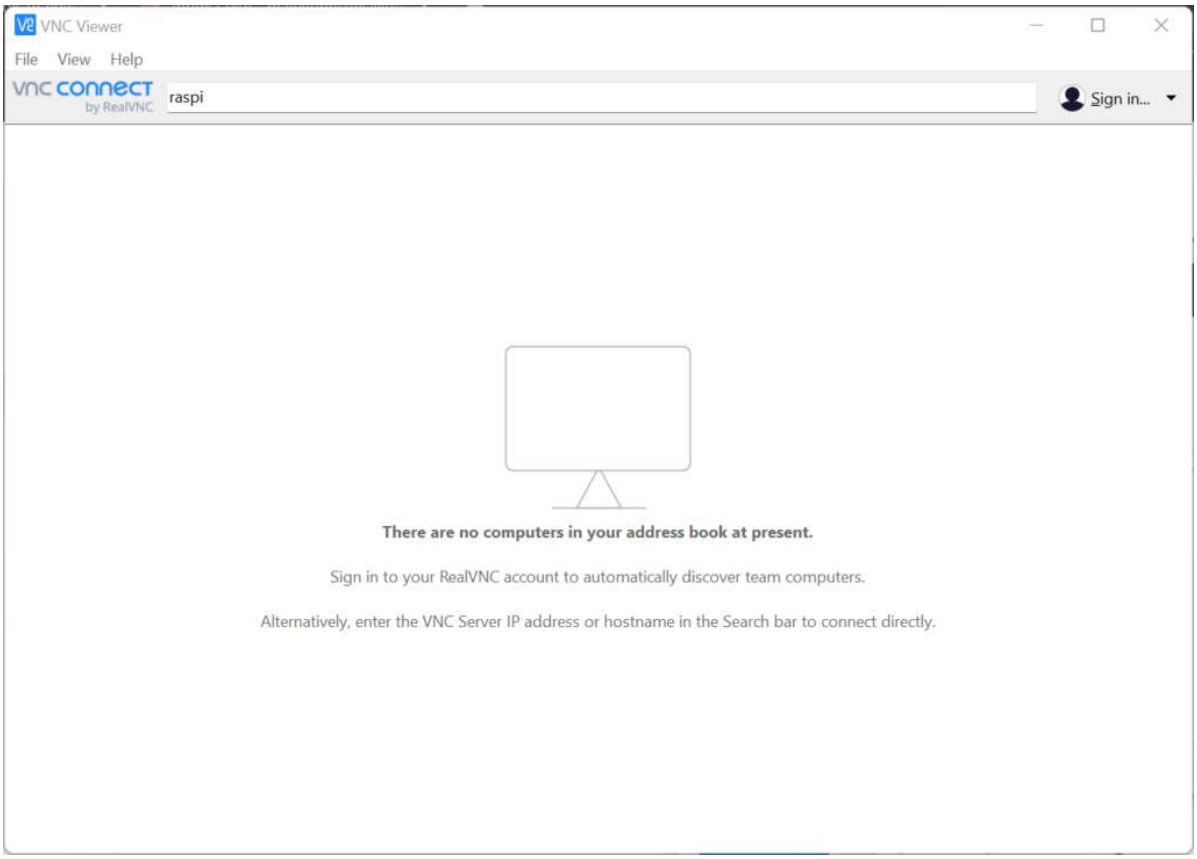
```
$ sudo apt install samba -y
```

- After this installation completes, you should be able to ping the Pi Hostname (configured in the *Raspberry Pi Configuration* program above) from a Windows machine:

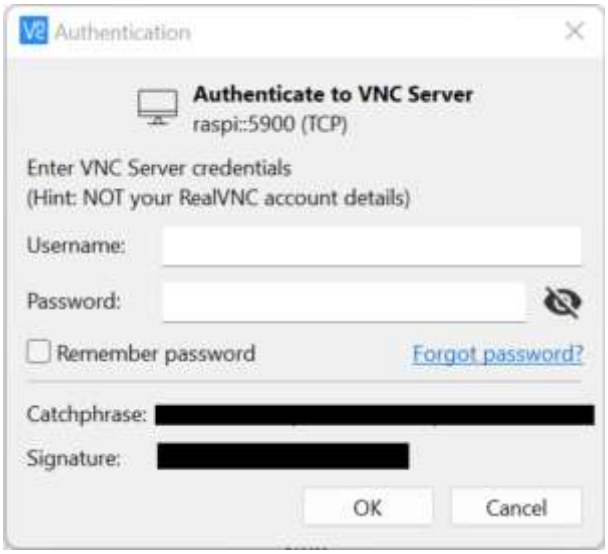
  ```
  C:\> ping raspi
  ```

- Next, install VNC Viewer on your Windows machine from here and accept all defaults in the installation process.

- You are now ready to launch a remote connection!
- From your Windows machine, hit the *Windows* key, type "VNC Viewer" and select to invoke it. (VNC Viewer is also available for macOS and Linux.)
- When VNC Viewer starts up for the first time, it may ask you if you want to create an account. An account is not mandatory, so you can decline and continue this tutorial.
- Type the host name of your Pi ("raspi" in my case) in the textbox and press the Enter key on your keyboard to launch the remote connection.




- You may be presented with a warning dialog box if you have never connected to this Raspberry Pi before. Accept this warning and continue.
- When prompted for credentials, enter the username and password you configured during the installation and press *OK*.




Amazing! We are connected remotely to the Pi and no longer need the dedicated HDMI monitor and USB keyboard/mouse. Let's go ahead and shut down the Pi for a minute so we can free up our monitor and keyboard/mouse:

- Launch a terminal session and enter the following command to cleanly shut down your system:

```
$ sudo poweroff
```

- Wait a minute for the Pi to fully power down.
- Unplug the power cable, HDMI cable, and the USB keyboard and mouse.
- Plug the power cable back in and the Pi should be on its way back up without all of those extra cables!
- After giving the Pi a minute or so to boot, connect to it once again using VNC Viewer.

## Create Windows File Share on the Pi (Optional)

We can create a Windows file share on the Pi so that we can copy files from our Windows machine directly onto the Pi. This can come in handy in many contexts. This is also useful for OS X users since you will be able to connect to your Raspberry Pi file share from Finder under "Shared". Here's how we do it:

- Launch a terminal session.
- Issue the following command to launch the Mousepad editor.

```
$ sudo -E mousepad /etc/samba/smb.conf &
```

- We could have also used the console-based "nano" editor, but Mousepad provides a GUI which is more suitable for beginners. The ampersand on the end of the command launches Mousepad as a background process so we are returned to a shell prompt in the terminal. Also, the smb.conf is part of the Samba package we installed earlier and this is why it exists on our system.
- Scroll down to the very bottom of the smb.conf configuration file and add the following (or copy and paste from this tutorial), replacing USERNAME with the user name you chose in the setup process:

```
[PiShare]
 comment=Pi Share
 path=/home/USERNAME
 browseable=yes
 writeable=yes
 only guest=no
 create mask=0740
 directory mask=0750
 public=no
```

- Save the changes to the smb.conf configuration file and exit the Mousepad editor. (Raspberry Pi OS will automagically restart the SMB (Samba) services based on the changes you made to the configuration file.)
- Back in a terminal session, we need to create an SMB (Samba) user so we can connect securely from Windows. We will create a user with the same name as the username you chose, but this will be an SMB (Windows user) rather than the one you assigned in the setup wizard. Use the following command, but replace USERNAME with your user name in lower case:

```
$ sudo smbpasswd -a USERNAME
```

You will be prompted to enter and re-enter a password. Document this password somewhere for future reference.

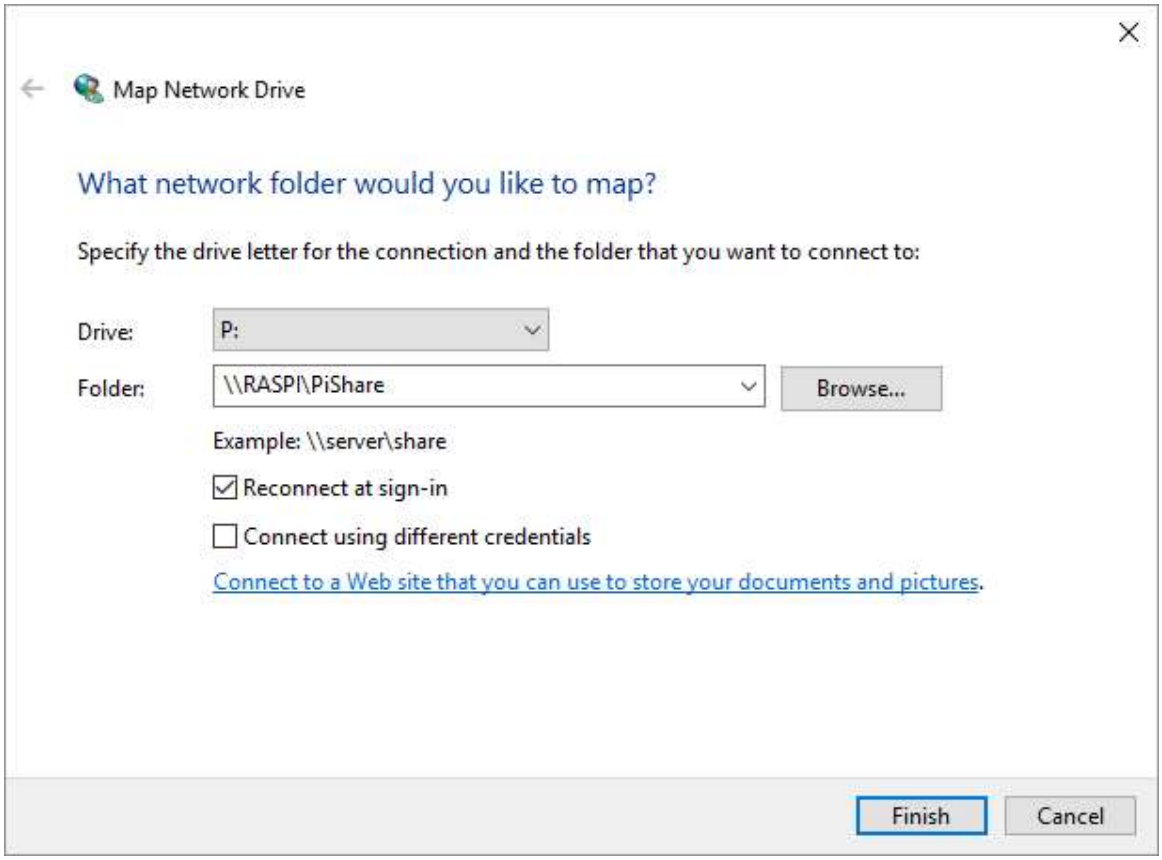
We are ready to connect to our newly created Pi file share from Windows!

- Back on your Windows machine, type the Windows key and "R" (Win-R) to launch a run box for typing in commands.
- Enter two backslashes followed by the name of your Pi machine (i.e. your Hostname) and hit enter. For example:

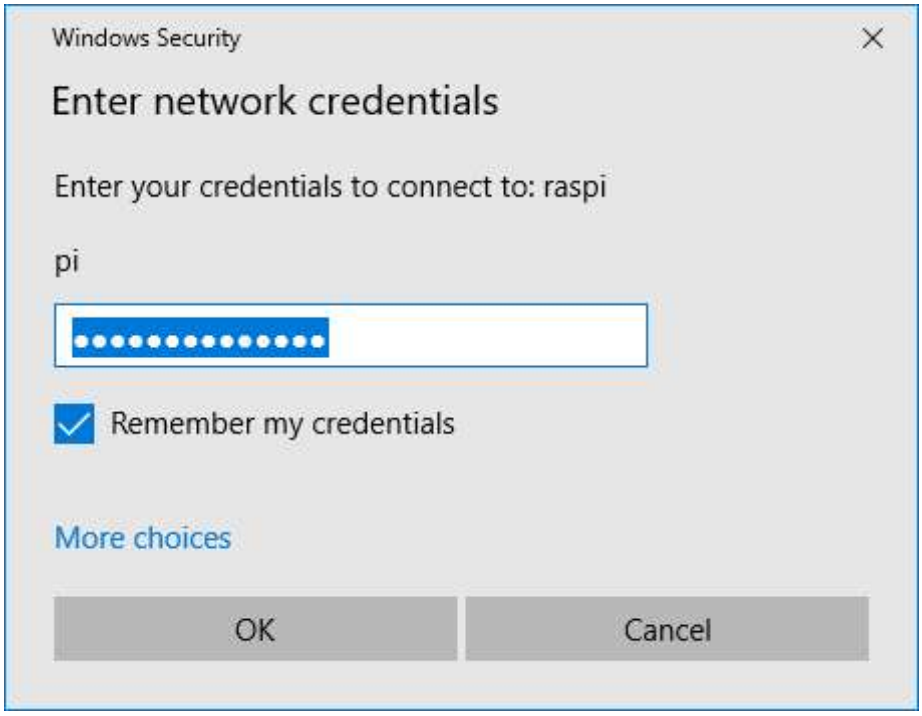
```
\\RASPI
```

- (Please note that when connecting to the file shares through Windows 10, the host name here can be entered in lower case or upper case.)
- When prompted for credentials, enter the following:
  - Username: The format to enter here in Windows parlance is the domain\username. For us, our domain will be our Pi hostname and our user name will be "pi". Therefore, we will enter the following with USERNAME replaced with your user name in lower case:
    raspi\USERNAME
    If your Hostname is not raspi, you will obviously enter that instead.
  - Password: (enter the password you created with the smbpasswd command above)
- Voilà! We should now be connected. Double click the *PiShare* folder to launch the file share you just created. You will also see a folder called *pi* that provides read-only access to your entire home directory.
- You can test your newly created file share by creating a text file on the Windows side and verifying that it shows up in the */home/USERNAME* directory on the Pi side. USERNAME should be replaced with your user name in lower case.

- In Windows, you can also map a drive to the file share you just created as follows:
  - Launch Windows Explorer.
  - Click on the *Map network drive* button in the ribbon menu at the top and select *Map network drive*.
  - Enter the drive letter (for example "P" for "pi" drive), the share folder path, and select *Reconnect at sign-in*:




  - Enter the password you created with the smbpasswd command above (if prompted) and select *Remember my credentials*:




- You should now see a “P” drive (or other drive letter you selected) when viewing through Windows Explorer!

## Install Node.js

You now have an amazing general purpose Raspberry Pi system that can be used for a variety of tasks and inter-operates well in the Windows world (it even looks like a Windows machine to the other Windows machines!) – and can play nicely in the Mac and Linux world too. Let’s go ahead and install Node.js so we will be ready to do some fun projects in the future. Here are the steps:

Our friends at NodeSource host and maintain some excellent Node.js binary distributions. We will leverage a command they have written to add another package repository to our Pi so that we will be able to “apt install” a modern version of Node.js from their repository. This is beneficial since the Debian/Raspberry Pi OS versions may not always be up to date. By adding the NodeSource repository, we will also be able to receive updates rather than just installing a standalone (.deb) file version of Node. that cannot be updated easily.

> *Note: As described at the beginning of this article, this final section of the tutorial related to the installation of Node.js requires a Pi system based on the newer ARMv7 or ARMv8 chip such as the Pi 4/Pi 3/Pi 2. NodeSource provides Node.js binaries for these newer ARMv7+ architectures, but not for Raspberry Pi systems based on the older ARMv6 architecture such as the Raspberry Pi Model B/B+ or the older Raspberry Pi Zero. The Raspberry Pi Zero 2, however, is based on ARMv7 and thus works with the NodeSource Node.js binaries*
>
> *Read the writing carefully on your Raspberry Pi circuit board to confirm it indicates something like "Raspberry Pi 4 Model B" or "Raspberry Pi 2 Model B". If in doubt, run the following command in the terminal:*
>
> ```
> $ uname -m
> ```
>
> *If the result returned starts with "armv6", you are running a Raspberry Pi based on the older ARMv6 chipset and the next Node.js installation step will not work; otherwise, you are ready for the next step.*

Here we go in the final stretch! Let's proceed with an installation of the latest version of Node at the moment which is Node 20.5.1.

```
$ curl -sL https://deb.nodesource.com/setup_20.x | sudo -E bash -
```

The previous command updates our Debian apt package repository to include the NodeSource packages.

> *Note: It's generally a good idea from a security perspective to know what commands you are invoking on your system, especially since the command above invokes the script as the root user. If you want the technical details behind this script, you can launch the URL (https://deb.nodesource.com/setup_20.x) in your browser and review*

> *the code. The script carries out some ceremony to determine the Linux distribution you are running and instructs the Debian apt package system to add the NodeSource package repository as a trusted source for obtaining Debian packages. This enables us to install Node.js now and upgrade to more recent versions of Node.js when they become available.*

Now that we have added the NodeSource package repository, we can move on and install Node.js!

```
$ sudo apt install nodejs -y
```

We can then test and see what version of Node.js we are running and launch the Node REPL as we discussed in one of my previous articles as a quick test to confirm the installation was successful.

```
$ node -v
v20.5.1
$ node
> 1 + 3
4
<
> # We can hit Ctrl-C twice to exit the REPL and get back to the
```

## Continue your Node.js Journey

To continue your learning-through making journey with Node.js, follow my tutorial to leverage Node.js for creating a functional webserver without any code. This can be accomplished using the Raspberry Pi system you just created here or on a Windows/Linux/Mac machine as I outlined in one of my previous articles. I hope you have enjoyed this article!

Follow @thisDaveJ (Dave Johnson) on Twitter and subscribe to my RSS feed to stay up to date on the latest tutorials and tech articles.

## Related Articles

Create a Lightweight Raspberry Pi with Raspbian Lite

Learning through Making – Getting Started with Node.js

Upgrading to more recent versions of Node.js on the Raspberry Pi

Create a Web Server in Node without any Code

Make Your Raspberry Pi Web Server Available on the Internet with Node.js

Controlling a Raspberry Pi from a Mobile Device with Bonus Menu Too

Tagged as: javascript, networking, node.js, raspi, windows

Categorized in: Node.js, Raspberry Pi

September 6, 2022 Dave Johnson 154 Comments

Follow @thisDaveJ

Support these free tutorials by buying something on Amazon with this link.

Copyright ©2024 Dave Johnson - using Decode theme