Part 74
-
Headless Pi & Remote Access

Version: 2021-02-20

Setting up a Raspberry Pi headless

If you do not use a monitor or keyboard to run your Pi, known as headless, but you still need to do some wireless setup, there is a facility to enable wireless networking and SSH when creating a image.
Once an image is created on an SD card, by inserting it into a card reader on a Linux or Windows machines the boot folder can be accessed. Adding certain files to this folder will activate certain setup features on the first boot of the Pi itself.

Setting up wireless networking

You will need to define a `wpa_supplicant.conf` file for your particular wireless network. Put this file in the `boot` folder, and when the Pi first boots, it will copy that file into the correct location in the Linux `root` file system and use those settings to start up wireless networking.
After the Pi is connected to power, make sure to wait up to 5 minutes for it to boot up and register on the network. The Pi's IP address will not be visible immediately after power on, so this step is crucial to connect to it headlessly.
Note: Depending on the OS and editor you are creating this on, the file could have incorrect newlines or the wrong file extension. So make sure you use an editor that accounts for this. Linux expects the line feed (LF) newline character.

`wpa_supplicant.conf` file example

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=BE

network={
      ssid="MYWIFI-SSID"
      psk="MYWIFI-PASSWORD"
}
```

See “Raspberry Pi - Part 07 - Activating WiFi” for more detailed information on WiFi connections

Remote Access

With no keyboard or monitor, you will need some way of accessing the headless Raspberry Pi.

IP Address

Any device connected to a Local Area Network is assigned an IP address. In order to connect to your Raspberry Pi from another machine using SSH, you need to know the Pi's IP address. This is easy if you have a display connected, and there are a number of methods for finding it remotely from another machine on the network.

Router devices list

In a web browser navigate to your router's, log in and browse to the list of connected devices.

Resolving raspberrypi.local with mDNS

On Raspberry Pi OS, multicast DNS is supported out-of-the-box by the Avahi service.
If your device supports mDNS, you can reach your Raspberry Pi by using its hostname and the `.local` suffix. The default hostname on a fresh Raspberry Pi OS install is `raspberrypi`, so by default any Raspberry Pi running Raspberry Pi OS responds to

```
ping raspberrypi.local
```

If the Raspberry Pi is reachable, ping will show its IP address

```
raspberrypi.local (192.168.1.131): 56 data bytes
64 bytes from 192.168.1.131: icmp_seq=0 ttl=255 time=2.618 ms
```

If you change the system hostname of the Raspberry Pi (e.g., by editing `/etc/hostname`), Avahi will also change the `.local` mDNS address.

nmap command

The nmap command (Network Mapper) is a free and open-source tool for network discovery, available for Linux, macOS, and Windows https://nmap.org/download.html.
To install on Windows, download the self-installer as it install also the npcap utility that is neededsee the nmap.org download page.
To use nmap to scan the devices on your network, you need to know the subnet you are connected to. First find your own IP address, in other words the one of the computer you're using to find your Pi's IP address: go to the Control Panel, then under Network and Sharing Center, click View network connections, select your active network connection and click View status of this connection to view the IP address
Now you have the IP address of your computer, you will scan the whole subnet for other devices. For example, if your IP address is `192.168.1.5`, other devices will be at addresses like `192.168.1.2`, `192.168.1.3`, `192.168.1.4`, etc. The notation of this subnet range is `192.168.1.0/24` (this covers `192.168.1.0 to 192.168.1.255`).
Now open a command prompt and use the nmap command with the `-sn` flag (ping scan) on the whole subnet range.

```
nmap -sn 192.168.1.0/24
```

Ping scan just pings all the IP addresses to see if they respond. For each device that responds to the ping, the output shows the hostname and IP address like so

```
Starting Nmap 6.40 ( http://nmap.org ) at 2014-03-10 12:46 GMT
Nmap scan report for hpprinter (192.168.1.2)
Host is up (0.00044s latency).
Nmap scan report for Gordons-MBP (192.168.1.4)
Host is up (0.0010s latency).
Nmap scan report for ubuntu (192.168.1.5)
Host is up (0.0010s latency).
Nmap scan report for raspberrypi (192.168.1.8)
Host is up (0.0030s latency).
```

```
Nmap done: 256 IP addresses (4 hosts up) scanned in 2.41 seconds
```

Here you can see a device with hostname `raspberrypi` has IP address `192.168.1.8`.

Getting the IP address of a Pi using your smartphone

The `Fing` app is a free network scanner for smartphones. It is available for Android and iOS. Your phone and your Raspberry Pi have to be on the same network, so connect your phone to the correct wireless network.

When you open the `Fing` app, touch the refresh button in the upper right-hand corner of the screen. After a few seconds you will get a list with all the devices connected to your network. Scroll down to the entry with the manufacturer "`Raspberry Pi`". You will see the IP address in the bottom left-hand corner, and the MAC address in the bottom right-hand corner of the entry.

SSH (Secure Shell)
You can access the command line of a Raspberry Pi remotely from another computer or device on the same network using SSH.
The Raspberry Pi will act as a remote device. You can connect to it using a client on another machine. You only have access to the command line, not the full desktop environment.

Enable SSH
As of the November 2016 release, Raspberry Pi OS has the SSH server disabled by default. For headless setup, SSH can be enabled by placing a file named `ssh`, without any extension, onto the `boot` partition of the SD card from another computer. When the Pi boots, it looks for the `ssh` file. If it is found, SSH is enabled and the file is deleted. The content of the file does not matter; it could contain text, or nothing at all.
If you have loaded Raspberry Pi OS onto a blank SD card, you will have two partitions. The first one, which is the smaller one, is the `boot` partition. Place the file into this one.

Set up your client
You can use SSH to connect to your Raspberry Pi from a Windows 10 computer that is using October 2018 Update or later without having to use third-party clients.
If you haven't already done so, go to `Settings > Apps > Apps & features > Manage optional features > Add a feature`, and choose to `install OpenSSH Client`.
You will need to know your Raspberry Pi's IP address to connect to it.
To connect to your Pi using following command from a command prompt but replace <IP> with the IP address of the Raspberry Pi.

```
ssh pi@<IP>
```

If you have set up another user on the Raspberry Pi, you can connect to it in the same way, replacing the default pi user with your own username, e.g.

```
ssh eben@192.168.1.5
```

If you receive a connection timed out error, it is likely that you have entered the wrong IP address for the Raspberry Pi.
When the connection works you will see a security/authenticity warning. Type `yes` to continue. You will only see this warning the first time you connect.
In the event that your Pi has taken the IP address of a device to which your computer has connected before (even if this was on another network), you may be given a warning and asked to clear the record from your list of known devices. Following this instruction and trying the ssh command again should be successful.
Next you will be prompted for the password for the user as which you are trying to connect: the default password for the `pi` user on Raspberry Pi OS is `raspberry`. For security reasons it is highly recommended to change the default password on the Raspberry Pi. You should now be able to see the Raspberry Pi prompt, which will be identical to the one found on the Raspberry Pi itself.

```
pi@raspberrypi ~ $
```

You are now connected to the Raspberry Pi remotely, and can execute commands.

Note: There are quite a few SSH client available free to use. To name some of them:
Putty, KiTTY (based on Putty), Solar PuTTY, SuperPuTTY, PuTTY Tray, ExtraPuTTY, MobaXterm, SmarTTY, Bitvise SSH Client, mRemoteNG

Passwordless SSH access
It is possible to configure your Raspberry Pi to allow access from another computer without needing to provide a password each time you connect. To do this, you need to use an SSH key instead of a password. To generate an SSH key

- Check for existing SSH keys
  First, check whether there are already keys on the Windows 10 computer you are using to connect to the Raspberry Pi. Replace <user> with your login username

  ```
  dir C:\Users\<user>\.ssh
  ```

  If you see files named `id_rsa.pub` or `id_dsa.pub` then you have keys set up already, so you can skip the 'Generate new SSH keys' step below.

- Generate new SSH keys
  To generate new SSH keys enter the following command

  ```
  ssh-keygen
  ```

  Upon entering this command, you will be asked where to save the key. We suggest saving it in the default location `(~/.ssh/id_rsa)` by pressing Enter.
  You will also be asked to enter a passphrase, which is optional. The passphrase is used to encrypt the private SSH key, so that if someone else copied the key, they could not impersonate you to gain access. If you choose to use a passphrase, type it here and press Enter, then type it again when prompted. Leave the field empty for no passphrase.
  Now look inside your .ssh directory

  ```
  dir C:\Users\<user>\.ssh
  ```

  and you should see the files `id_rsa` and `id_rsa.pub`
  The `id_rsa` file is your private key. Keep this on your computer.
  The `id_rsa.pub` file is your public key. This is what you share with machines that you connect to: in this case your Raspberry Pi. When the machine you try to connect to matches up your public and private key, it will allow you to connect.
  Take a look at your public key to see what it looks like

  ```
  type C:\Users\<user>\.ssh\id_rsa.pub
  ```

  It should be in the form:

  ```
  ssh-rsa <REALLY LONG STRING OF RANDOM CHARACTERS> user@host
  ```

- Copy your public key to your Raspberry Pi
  Copy the file manually over SSH

  ```
  type C:\Users\<user>\.ssh\id_rsa.pub | ssh <USERNAME>@<IP-ADDRESS>
  'mkdir -p ~/.ssh && cat >> ~/.ssh/authorized_keys'
  ```

  If you see the message `ssh: connect to host <IP-ADDRESS> port 22: Connection refused` and you know the `IP-ADDRESS` is correct, then you may not have enabled SSH on your Raspberry Pi.
  Note: you can also send files over SSH using the scp command (secure copy).

- Adjust permissions for your `home` and `.ssh` directories
  If you can't establish a connection after following the steps above there might be a problem with your directory permissions on the Raspberry Pi. First, you want to check the logs for any errors:

  ```
  tail -f /var/log/secure
  # might return:
  Nov 23 12:31:26 raspberrypi sshd[9146]: Authentication refused: bad
  ownership or modes for directory /home/pi
  ```

  If the log says `Authentication refused: bad ownership or modes for directory /home/pi` there is a permission problem regarding your home directory. SSH needs your `home` and `~/.ssh` directory to not have group write access. You can adjust the permissions using `chmod`

  ```
  chmod g-w $HOME
  chmod 700 $HOME/.ssh
  chmod 600 $HOME/.ssh/authorized_keys
  ```

  Now only the user itself has access to `.ssh` and `.ssh/authorized_keys` in which the public keys of your remote machines are stored.

SCP (Secure Copy)
We recommend using the WinSCP client. Follow the instructions on the WinSCP website https://winscp.net/eng/download.php to install the client, then follow the WinSCP Quick Start instructions.