

Setting up your Raspberry Pi Zero W (Wireless)

Instructions below are

- specific to the **Raspberry Pi Zero W** target board
- valid as of August 2022.

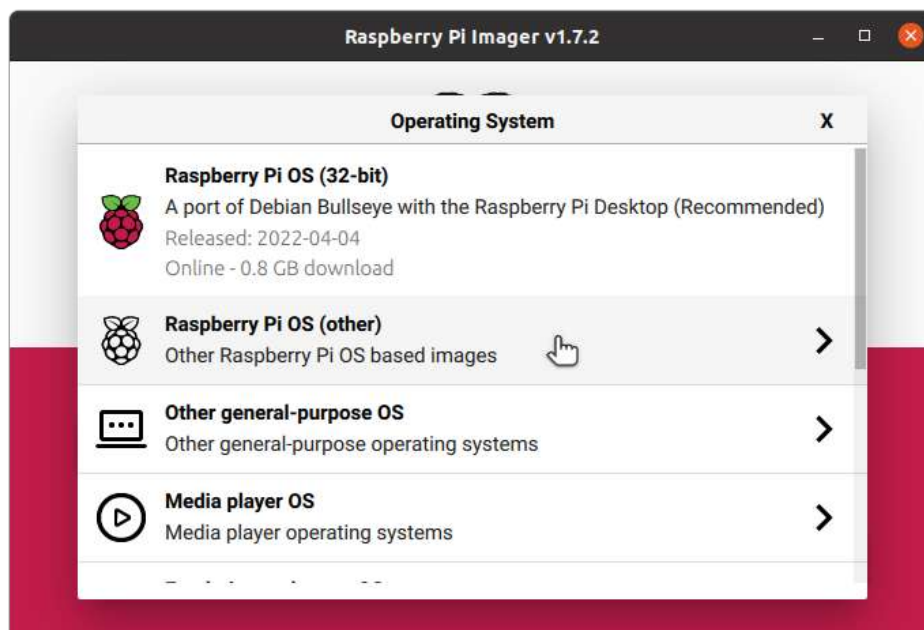


Table of Contents

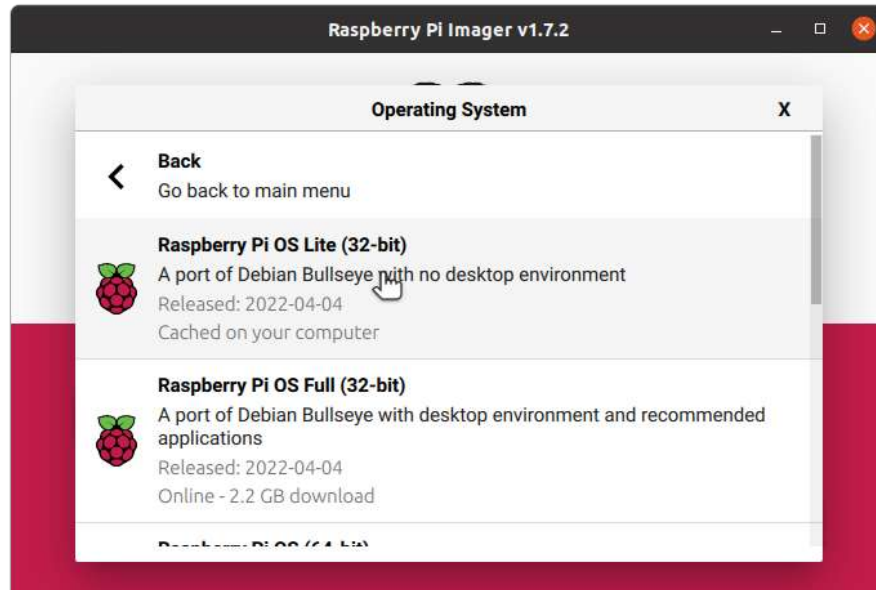
I Installing the Raspberry Pi OS on an SD card.....	2
II Login to the board over SSH.....	5
Obtain the device's IP address.....	5
Headless login over SSH.....	7
What to do once logged in to the Raspberry Pi.....	8

I Installing the Raspberry Pi OS on an SD card

1. Burn the microSD card with a recent and stable Raspberry Pi OS image; simultaneously have the Raspberry Pi *Imager* app setup the network, username, and so on...
 - i. Download the Raspberry Pi Imager app
Navigate to <https://www.raspberrypi.com/software/> , download and install it on your host system (you can even install it onto a Linux guest (that supports graphics) and run it from there)
 - ii. Run it, thus installing a recent stable Raspberry Pi OS image onto your microSD card!
 - a) A 45 second video demonstrating how: <https://youtu.be/ntaXWS8Lk34> ; take a look!
 - iii. Select the OS:
 - a) click the 'CHOOSE OS' button
 - b) To optimize, and for the Raspberry Pi Zero (W) target, click on the second choice: Raspberry Pi OS (other) [below]



- c) then click on the first one:
Raspberry Pi OS Lite (32-bit) – no desktop environment

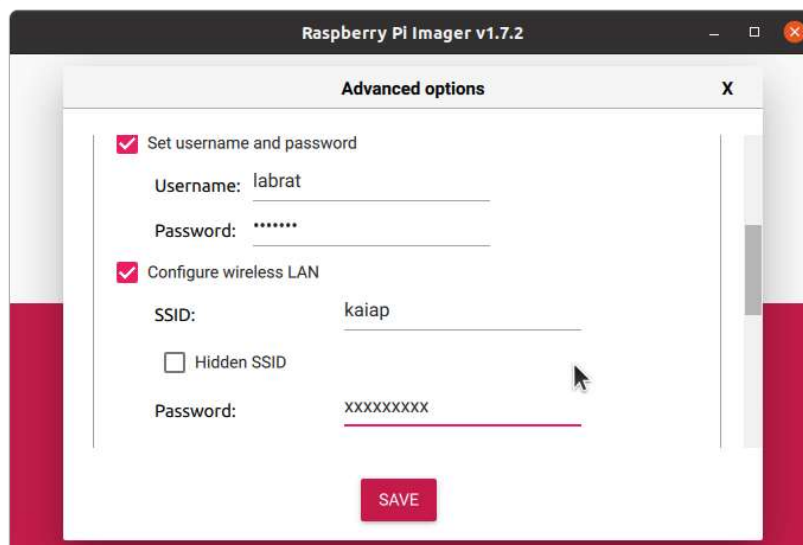


iv. Then select the microSD card via the STORAGE button

v. **NOTE: Very IMPORTANT -**

Next, before writing the image, DO make use of the Imager app's **Settings gearwheel** button! It's *very useful* to use the Imager app to pre-configure the R Pi OS – you can select:

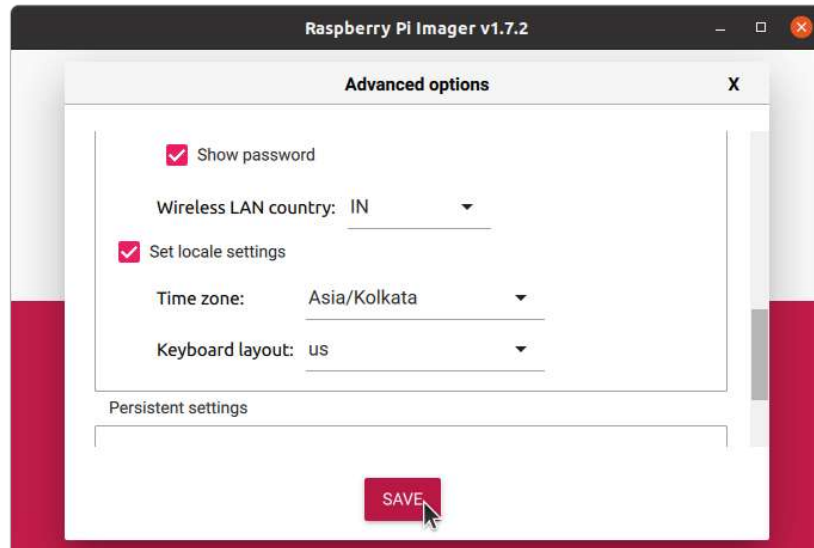
- a) the board hostname ('rpi0w_<myname>' perhaps)
- b) do Enable SSH
 - i. to keep it simple, you can initially use password authentication, but, as you learn more, DO switch to authentication via SSH keys only – its far more secure!
- c) Security: be sure to choose a username other than the default one (pi); f.e., labrat ; supply a password
- d) **Very Important:** Configure the wireless LAN - the WiFi AP; specify the SSID and password. We're going to use the Raspberry Pi Zero W in 'headless' mode – no HDMI monitor and all that jazz! You'll need to configure the WiFi AP in order to login over SSH.



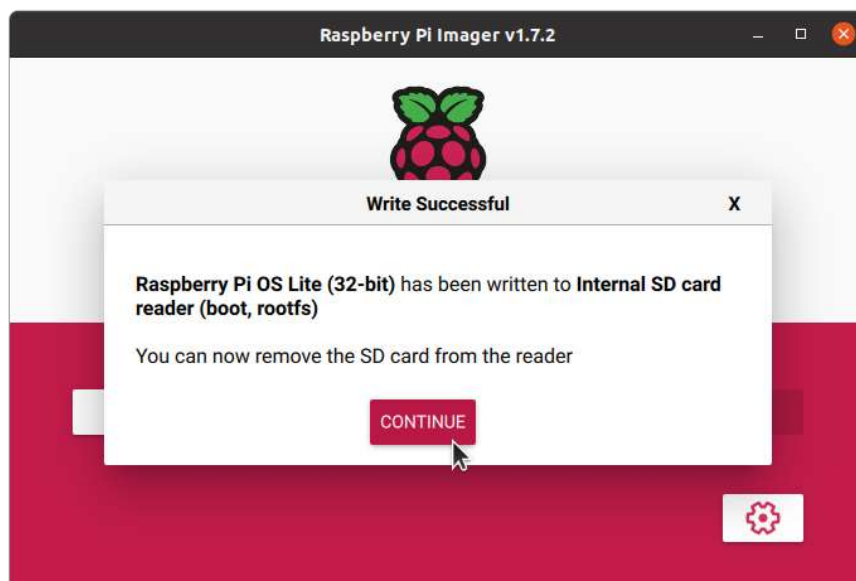
i. Don't forget to scroll down, set the:

1. Country for Wireless to India (IN)
2. Locale (Time zone as Asia/Kolkata, keyboard as 'us')

e) Click on the **SAVE** button when all set...



vi. Now click on the **WRITE** button; once you confirm, the image gets written and verified.



It can take a while, be patient... When done, eject the SD card, click on the **CONTINUE** button.

2. On a Linux (or MacOS) host (*or guest!*), I reinsert the SD card and do this to verify the OS image is written:

```

~ $ df -h | grep "^/dev/mmcblk"
/dev/mmcblk0p2 1.6G 1.2G 296M 81% /media/kaiwan/rootfs
/dev/mmcblk0p1 253M 50M 203M 20% /media/kaiwan/boot
~ $ ls /media/kaiwan/boot
bcm2708-rpi-b.dtb      bcm2710-rpi-3-b-plus.dtb  cmdline.txt      fixup.dat      overlays/
bcm2708-rpi-b-plus.dtb  bcm2710-rpi-cm3.dtb      config.txt      fixup_db.dat    start4cd.elf
bcm2708-rpi-b-rev1.dtb  bcm2710-rpi-zero-2.dtb   COPYING.linux   fixup_x.dat     start4db.elf
bcm2708-rpi-cm.dtb      bcm2710-rpi-zero-2-w.dtb  firstrun.sh     issue.txt      start4.elf
bcm2708-rpi-zero.dtb    bcm2711-rpi-400.dtb      fixup4cd.dat    kernel7l.img   start4x.elf
bcm2708-rpi-zero-w.dtb  bcm2711-rpi-4-b.dtb      fixup4.dat      kernel7l.img   start_cd.elf
bcm2709-rpi-2-b.dtb     bcm2711-rpi-cm4.dtb      fixup4db.dat    kernel8.img    start_db.elf
bcm2710-rpi-2-b.dtb     bcm2711-rpi-cm4s.dtb     fixup4x.dat     kernel.img     start.elf
bcm2710-rpi-3-b.dtb     bootcode.bin             fixup_cd.dat    LICENCE.broadcom start_x.elf
~ $ ls /media/kaiwan/rootfs/
bin@ dev/ home/ lost+found/ mnt/ proc/ run/ srv/ tmp/ var/
boot/ etc/ lib@ media/      opt/ root/ sbin@ sys/ usr/
~ $ cat /media/kaiwan/rootfs/etc/issue
Raspbian GNU/Linux 11 \n \l
~ $
~ $ █

```

On a Windows host you can always lookup the SD card content with the File explorer app.

Don't forget to unmount (eject) the SD card before physically removing it from it's slot.

II Login to the board over SSH

To login over SSH, we obviously **require the system's (DHCP-assigned) IP address**. We're assuming you've installed the Raspberry Pi OS and configured the wireless network – as the previous section described.

Obtain the device's IP address

- Power off the Raspberry Pi board
- Insert the microSD card containing the installed Raspberry Pi OS into your board, and apply power
- It should boot up ... give it a minute or two and then proceed to the next step...
- Ensure you're *on the same local network* as the device (f.e. 192.168.1.x or 10.20.1.x). To find it's IP address:
 - On a Windows host, via a browser:
 - navigate to the router gateway (f.e. 192.168.1.1 or 10.20.1.1)
 - login if required
 - Lookup the DHCP client list in the web browser app; the Raspberry Pi should show up; look for the hostname you gave it when installing the image. Once located, note it's allocated IP address.
 - On a Linux/Mac host:
 - Here too you can of course use a browser as in the previous paragraph (assuming you're on a Linux system running a graphical desktop)

- Via the Linux CLI, there are several ways to figure the device's IP address; among them are:
 - Install the nmap and arp-scan utils:
`sudo apt install nmap arp-scan -y`
 - `sudo arp-scan --localnet | egrep -i "Raspberry"`
 - `sudo nmap -sn -PR <domain-to-scan>/24` # domain f.e. 10.20.1.0
 - The 'nmap' way tends to be best; use my [localnet_discov.sh](#) helper script to make it easy!

An example of using it to discover the board's IP address:

```
~ $ localnet_discov.sh 10.20.1.0 | grep -i rpi0wlabrat
B8:27:EB: [redacted], rpi0wlabrat.wlan, 10.20.1.63
~ $
```

You can see the board hostname (highlighted in red colour) clearly.

Advanced: how to setup the Wireless network on the CLI

Just in case the Raspberry Pi Imager app does not succeed in setting up the wireless network...

1. On your Linux host, run `wpa_passphrase` and obtain the required snippet for a given SSID:

```
~ $ wpa_passphrase
usage: wpa_passphrase <ssid> [passphrase]

If passphrase is left out, it will be read from stdin
~ $
~ $ wpa_passphrase OfficeNet01 WelCome2WF0@Last
network={
    ssid="OfficeNet01"
    #psk="WelCome2WF0@Last"
    psk=b03af193ebd2ad742ea6f397f9da0357a005c644bcc91329b7f61b17d0046c3d
}
~ $
```

2. Copy-paste the output into the `wpa_supplicant.conf` file on the SD card:

```
sudo nano <rootfs_mountpoint>/etc/wpa_supplicant/wpa_supplicant.conf
<copy-paste>
<save & exit>
```

Obviously, delete the commented-out cleartext password line for security.

Headless login over SSH

Once you obtain the board IP address, login over SSH:

```
ssh <username>@<IPAddr>
```

F.e.

```
ssh myname@192.168.1.108
```

If it fails with a message like:

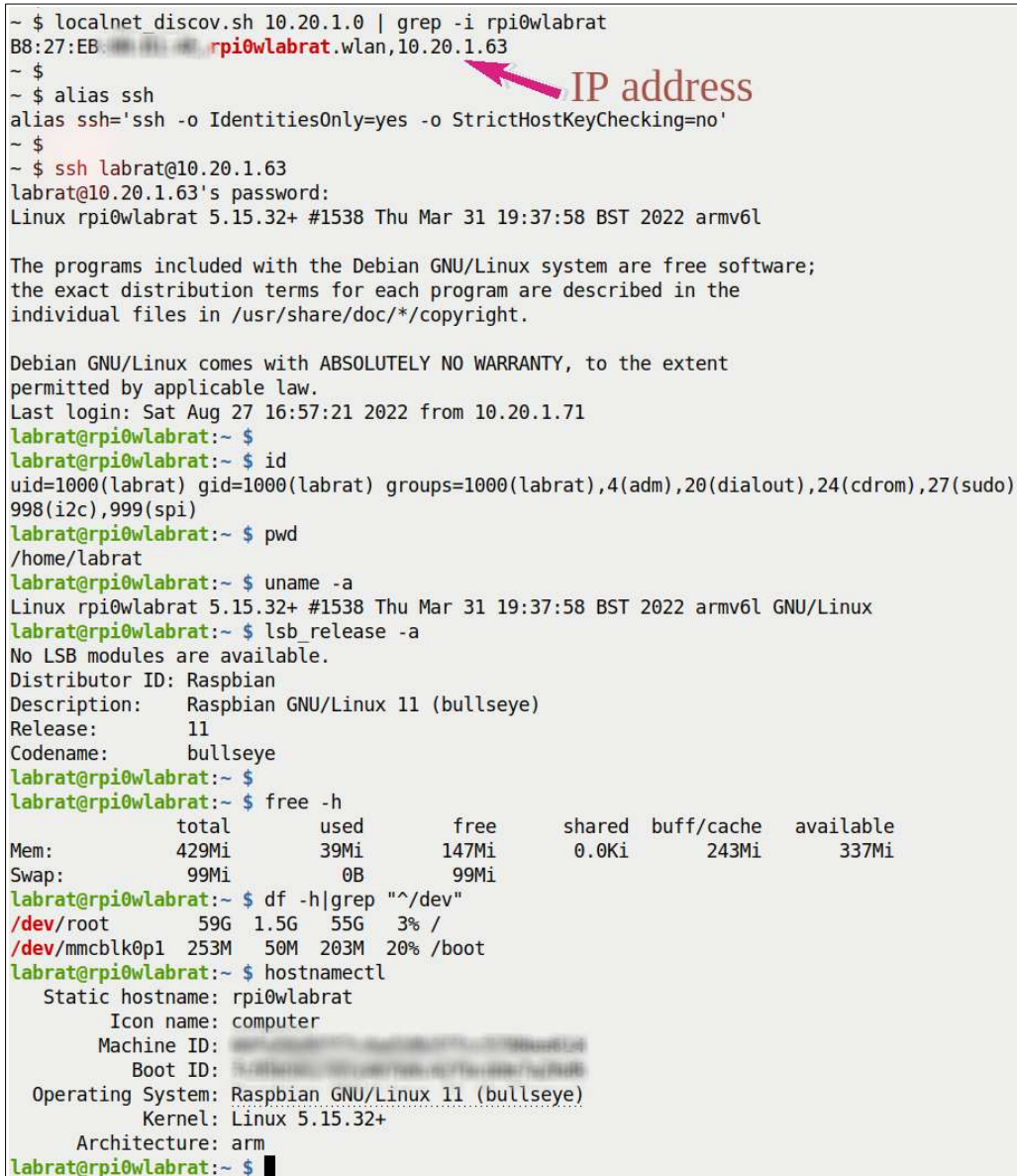
“... Too many authentication failures ...”

try this:

```
alias ssh='ssh -o IdentitiesOnly=yes -o StrictHostKeyChecking=no'
alias scp='scp -o IdentitiesOnly=yes -o StrictHostKeyChecking=no'
```

and then rerun the `ssh` command shown above! It should succeed.

A sample screenshot:



```
~ $ localnet discov.sh 10.20.1.0 | grep -i rpi0wlabrat
B8:27:EB:00:00:00 rpi0wlabrat.wlan,10.20.1.63
~ $
~ $ alias ssh
alias ssh='ssh -o IdentitiesOnly=yes -o StrictHostKeyChecking=no'
~ $
~ $ ssh labrat@10.20.1.63
labrat@10.20.1.63's password:
Linux rpi0wlabrat 5.15.32+ #1538 Thu Mar 31 19:37:58 BST 2022 armv6l

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.
Last login: Sat Aug 27 16:57:21 2022 from 10.20.1.71
labrat@rpi0wlabrat:~ $
labrat@rpi0wlabrat:~ $ id
uid=1000(labrat) gid=1000(labrat) groups=1000(labrat),4(adm),20(dialout),24(cdrom),27(sudo),
998(i2c),999(spi)
labrat@rpi0wlabrat:~ $ pwd
/home/labrat
labrat@rpi0wlabrat:~ $ uname -a
Linux rpi0wlabrat 5.15.32+ #1538 Thu Mar 31 19:37:58 BST 2022 armv6l GNU/Linux
labrat@rpi0wlabrat:~ $ lsb_release -a
No LSB modules are available.
Distributor ID: Raspbian
Description: Raspbian GNU/Linux 11 (bullseye)
Release: 11
Codename: bullseye
labrat@rpi0wlabrat:~ $
labrat@rpi0wlabrat:~ $ free -h
              total        used        free      shared  buff/cache   available
Mem:           429Mi       39Mi       147Mi          0.0Ki        243Mi        337Mi
Swap:            99Mi          0B          99Mi
labrat@rpi0wlabrat:~ $ df -h|grep "^/dev"
/dev/root           59G   1.5G   55G    3% /
/dev/mmcblk0p1    253M    50M  203M   20% /boot
labrat@rpi0wlabrat:~ $ hostnamectl
  Static hostname: rpi0wlabrat
            Icon name: computer
        Machine ID: 75-4883a3c0-1000-4000-8000-000000000000
           Boot ID: 75-4883a3c0-1000-4000-8000-000000000000
  Operating System: Raspbian GNU/Linux 11 (bullseye)
         Kernel: Linux 5.15.32+
    Architecture: arm
labrat@rpi0wlabrat:~ $
```

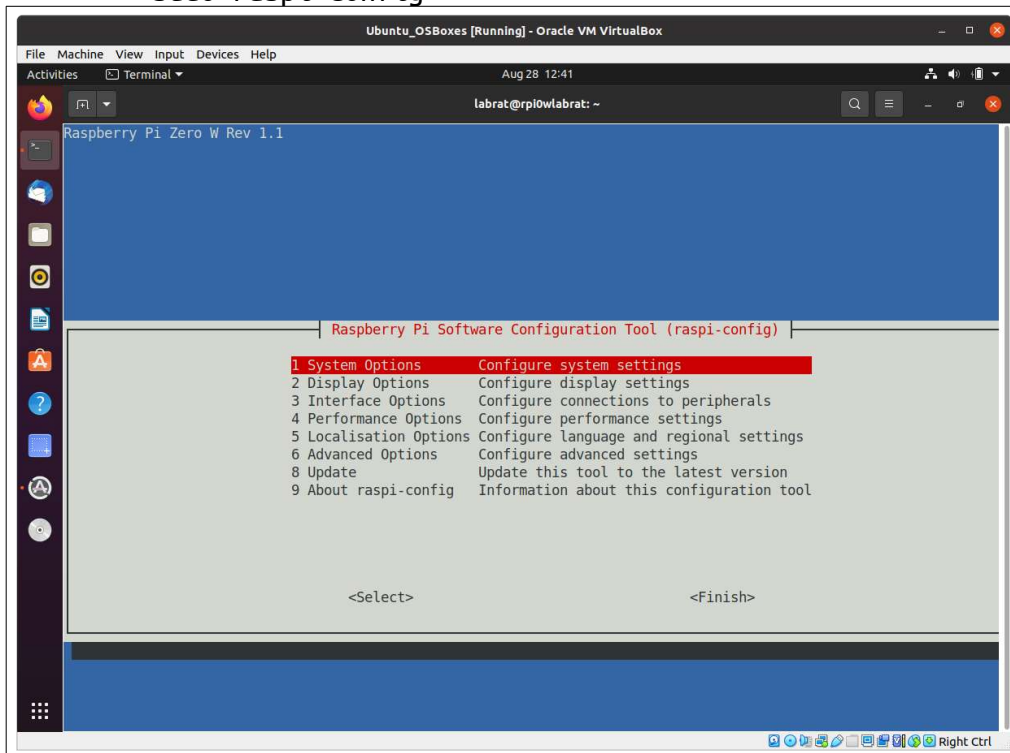
Tips:

- On a Windows host:
 - if required, setup the SSH client (eg. Putty) to use these parameters:
-o IdentitiesOnly=yes -o StrictHostKeyChecking=no
to ssh and scp
- On a Linux/Mac host:
 - put the ssh and scp aliases into a startup script
- *Security: use password-less logins by setting up SSH keys!*
Ref: <https://www.raspberrypi-spy.co.uk/2019/02/setting-up-ssh-keys-on-the-raspberry-pi/>
- **If all else fails when trying to ssh in**, power off the R Pi, attach a USB-to-serial dongle, setup a terminal emulator (Hyperterminal / Putty on Windows, minicom on Linux), login over it and get the IP address (ip a or ifconfig).
Ref: [*WORKING ON THE CONSOLE WITH THE RASPBERRY PI, kaiwanTECH, Dec 2018.*](#)

What to do once logged in to the Raspberry Pi

Once logged in to the R Pi board:

- First run
`sudo raspi-config`



and set things up as required...

- Next, install minimally required packages first:

```
sudo apt install -y git perl
```

- Then - optional, useful – to install common development + other tooling and utils:

```
git clone https://github.com/kaiwan/init
```

The partially truncated screenshot shows cloning the ‘init’ repo and running the `apt_install_common_stuff.sh` to install common tooling...

```
labrat@rpi0wlabrat:~ $ git clone https://github.com/kaiwan/init
Cloning into 'init'...
remote: Enumerating objects: 67, done.
remote: Counting objects: 100% (67/67), done.
remote: Compressing objects: 100% (50/50), done.
remote: Total 67 (delta 32), reused 40 (delta 17), pack-reused 0
Receiving objects: 100% (67/67), 16.77 KiB | 151.00 KiB/s, done.
Resolving deltas: 100% (32/32), done.
labrat@rpi0wlabrat:~ $
labrat@rpi0wlabrat:~ $ ls
init
labrat@rpi0wlabrat:~ $ cd init/
labrat@rpi0wlabrat:~/init $ ls
0setup_rpi.bash      dot_vimrc  README.md      source_repos.txt  ssh2rpi.sh
apt_install_common_stuff.sh  LICENSE   rpi_static_ip.txt  ssh2rpi          wpa_supplicant.conf
labrat@rpi0wlabrat:~/init $
labrat@rpi0wlabrat:~/init $ ./apt_install_common_stuff.sh
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  raspberrypi-kernel-headers
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 28.6 MB of archives.
After this operation, 182 MB of additional disk space will be used.
Get:1 http://archive.raspberrypi.org/debian bullseye/main armhf raspberrypi-kernel-headers armhf
.6 MB]
Fetched 28.6 MB in 9s (3,070 kB/s)
Selecting previously unselected package raspberrypi-kernel-headers.
(Reading database ... 42376 files and directories currently installed.)
Preparing to unpack .../raspberrypi-kernel-headers_1%3a1.20220328-1_armhf.deb ...
Unpacking raspberrypi-kernel-headers (1:1.20220328-1) ...
Progress: [ 20%] [#####.....]
```

(This script even sets up a useful startup script that will auto-execute whenever you spawn a shell!).

There, you're all set!