### *Requirements:*

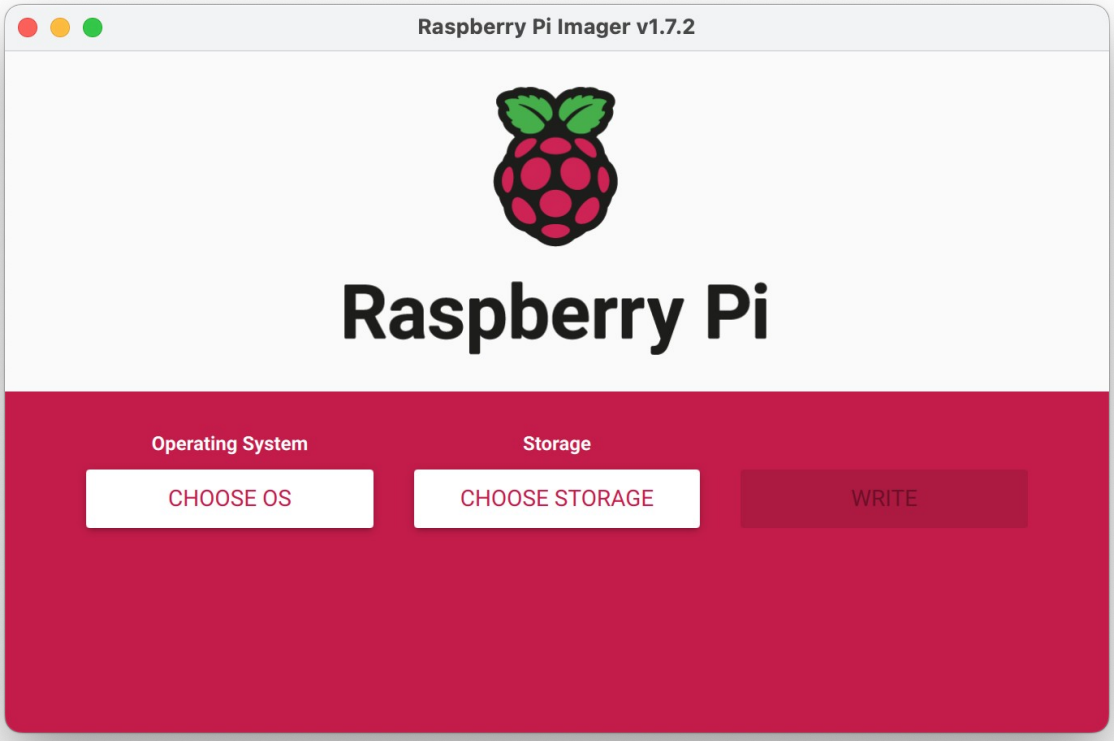
### *What do we need?*

- **RAK2287 Pi HAT WisLink LPWAN Concentrator**
- **RAK2287 Pi HAT**
- **Raspberry Pi 3B+** o **Raspberry Pi 4**
- 16GB SD Card + Card Reader
- 5V USB or USB C (Pi 4) adapter of at least 2A
- A Windows/Mac OS/Linux computer
- **Raspberry Pi Imager**
- The latest RAK2287 Pi HAT firmware although we don't need it in this manual

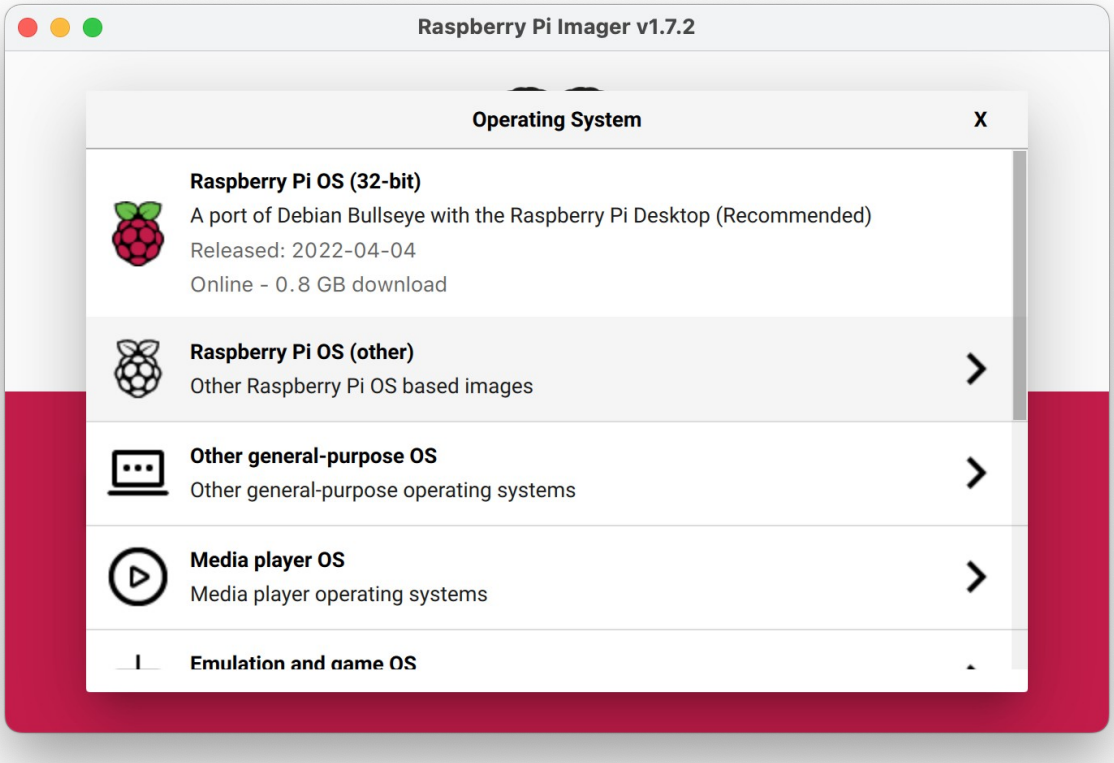
### *Installation on Raspberry Pi:*

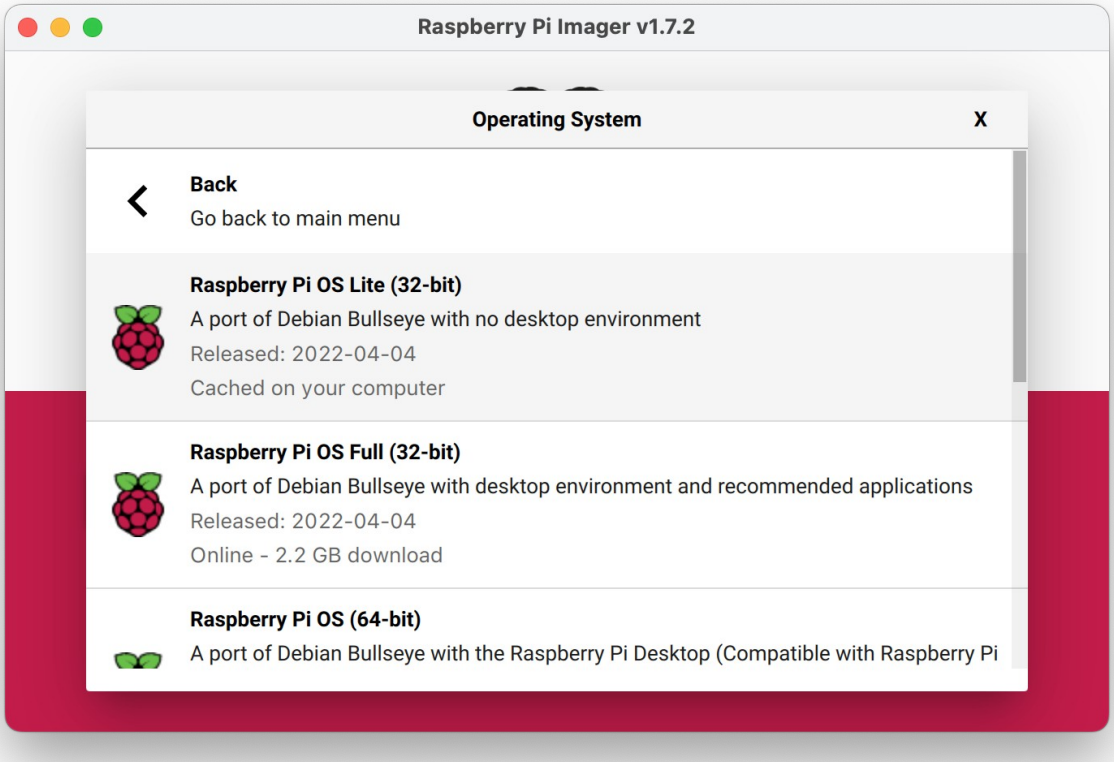
Download ***Raspberry Pi Imager***

Open ***Raspberry Pi Imager*** which can be downloaded from the official Raspberry Pi website. link above.

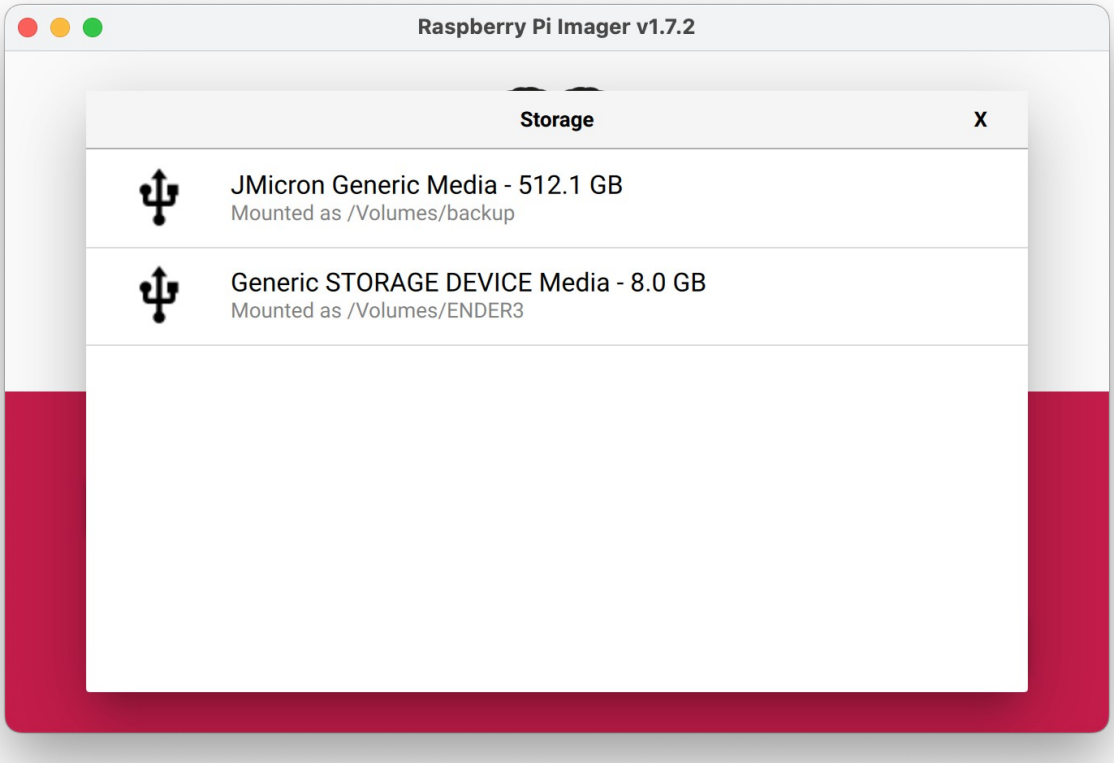



We choose OS Raspberry Pi OS (32-bit)



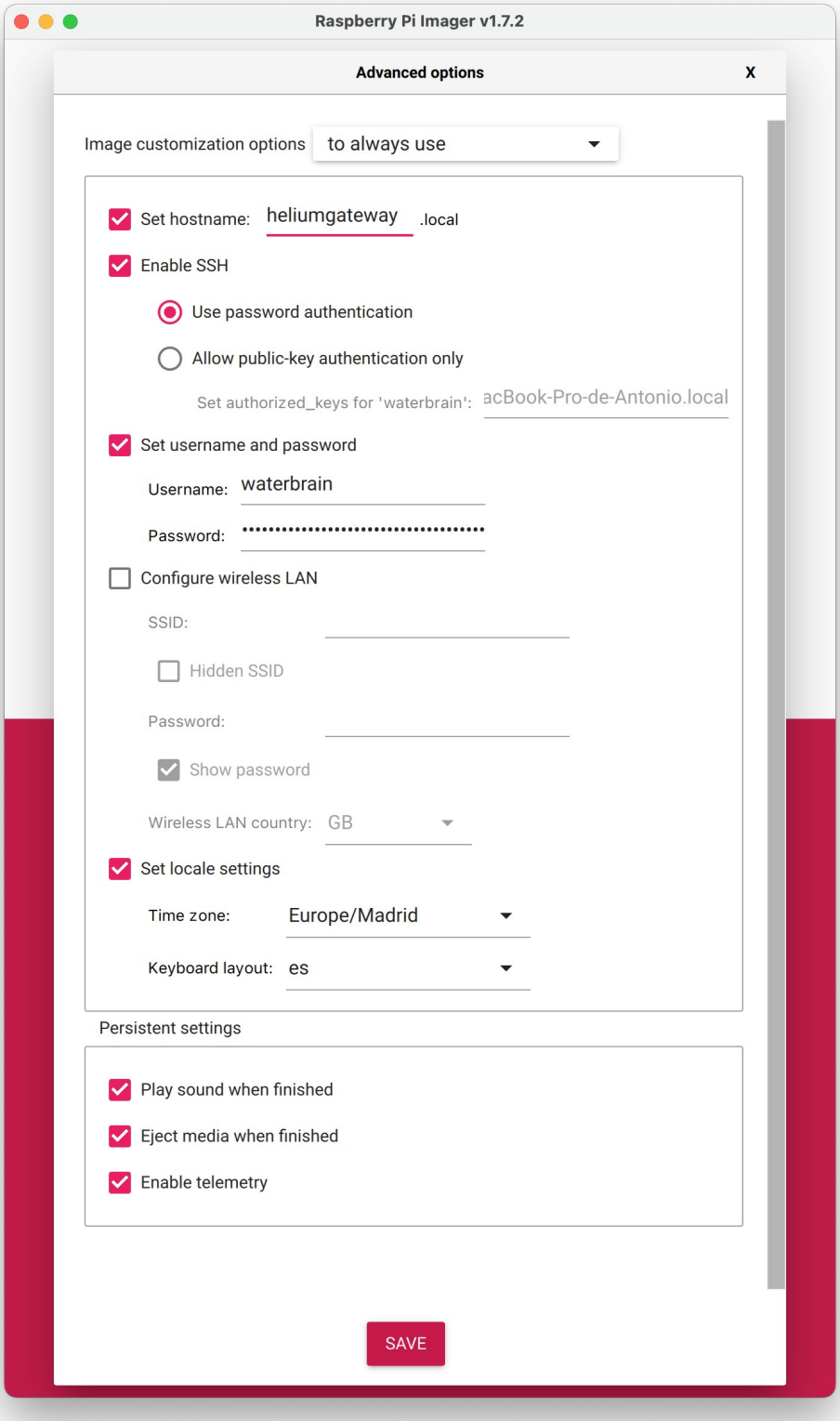




Next we choose the sd card:


Raspberry Pi Imager v1.7.2
Storage
X
JMicron Generic Media - 512.1 GB
Mounted as /Volumes/backup
Generic STORAGE DEVICE Media - 8.0 GB
Mounted as /Volumes/ENDER3

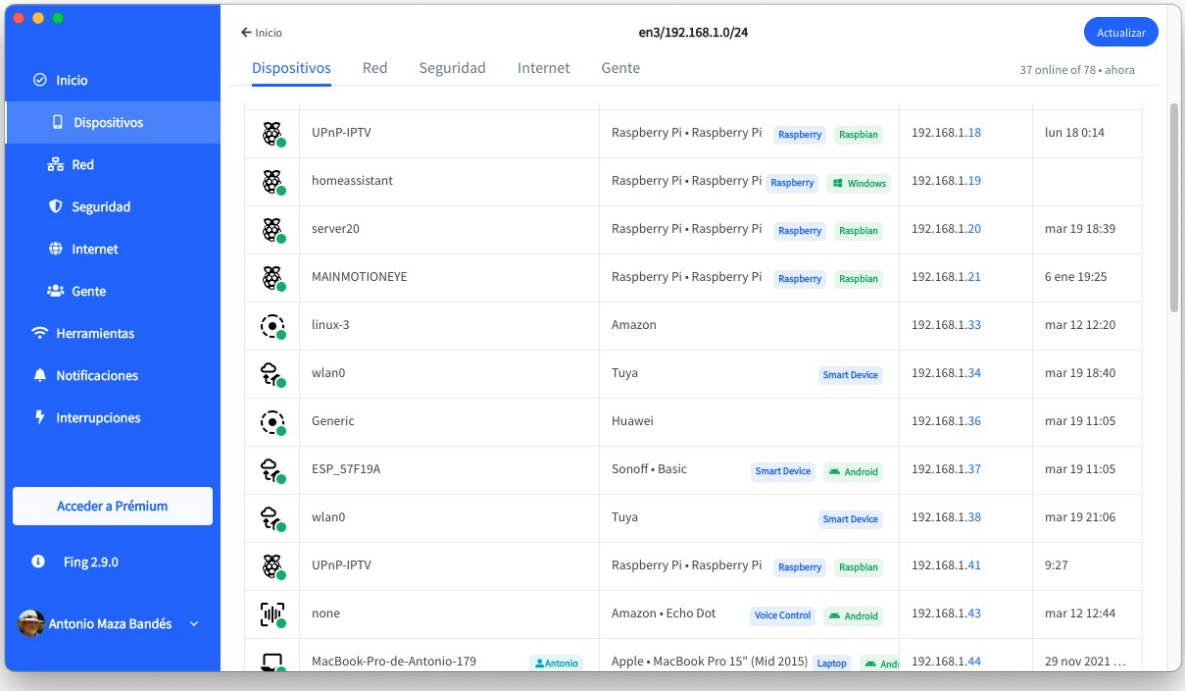

Raspberry Pi Imager v1.7.2
Raspberry Pi
Operating System
Storage
RASPBERRY PI OS LITE (32-BIT)
GENERIC STORA...
WRITE

Here the configuration wheel appears, press it and For legal and security reasons, to avoid attacks with the pi user, I leave it like this:




WRITE and we can now give write. We wait for it to finish. We introduce the image in the Raspberry and we boot.

We look for the IP it is on, I use Fing, which is in the free version for any computer, including iPhone or Android.

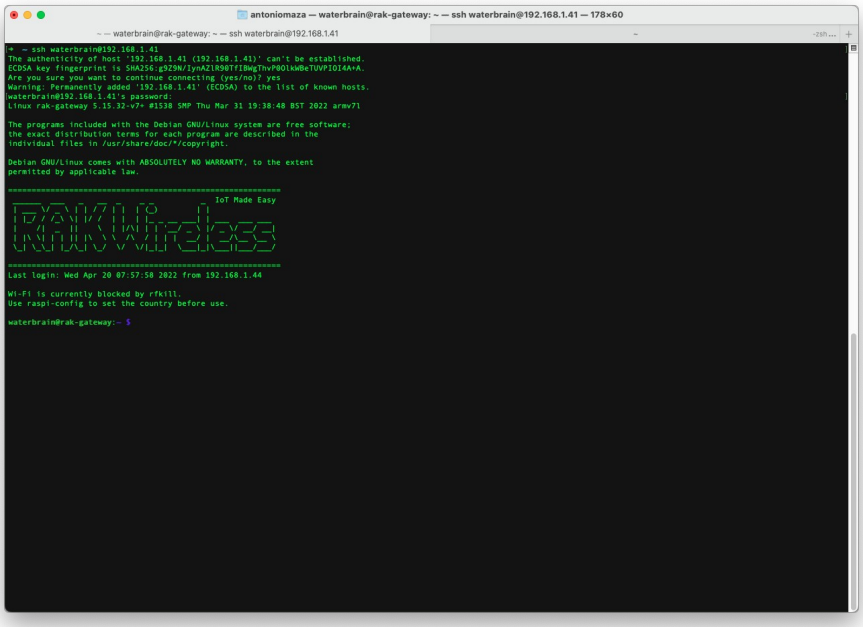



I see that at 9:27 a new device has appeared with the IP 192.168.1.41 so I know where the Rasberry is.

We open a terminal screen:

- **ssh** waterbrain@192.168.1.41

I personally prefer ssh, although you can use any tool like **PuTTY**.Because it is the first time it asks us to add the key of the Raspberry address to its file of known hosts, we say yes and voila:



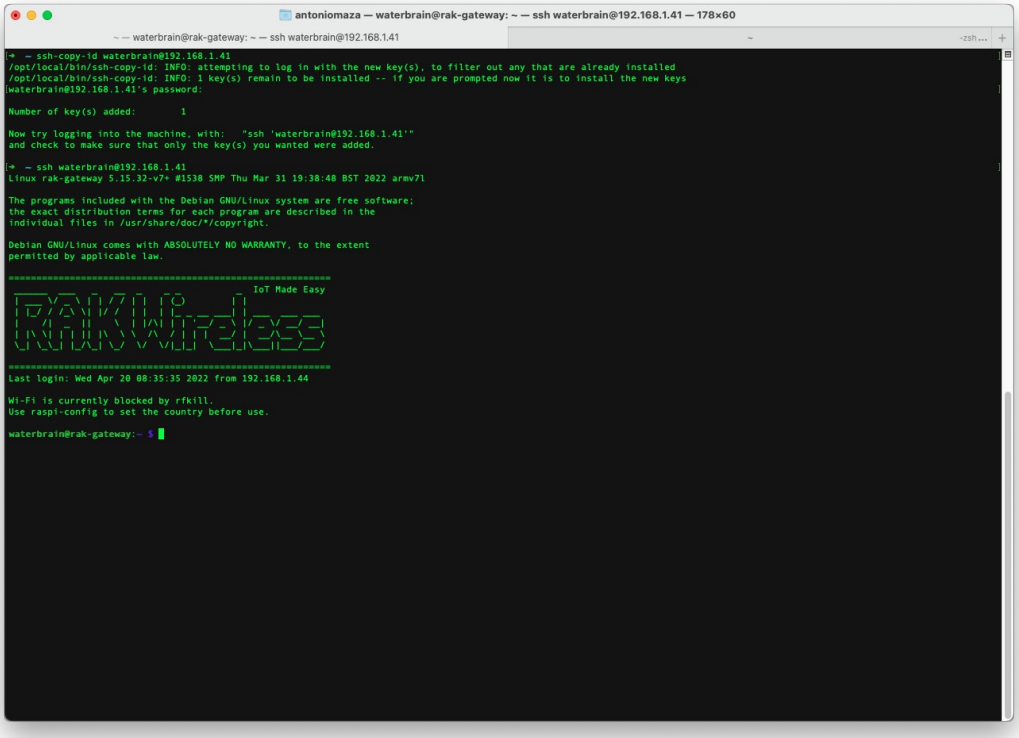

If we want it not to ask us for the password:
We leave the Raspberry (exit) and execute:

- **ssh-copy-id** waterbrain@192.168.1.41

in our terminal, prior to this we must have a key or unique key generated. I do not do it, but just in case we have not done it we will use:

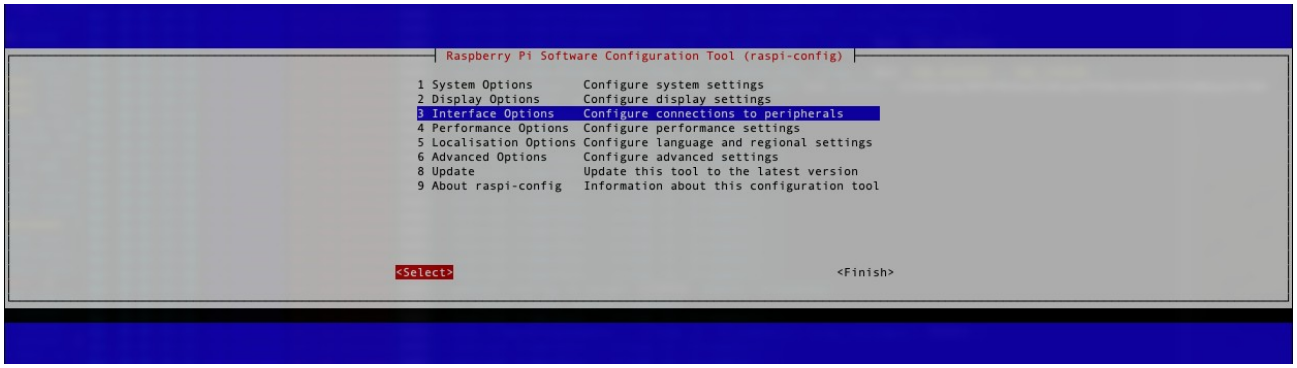
- **ssh-keygen -b 4096 -t rsa**

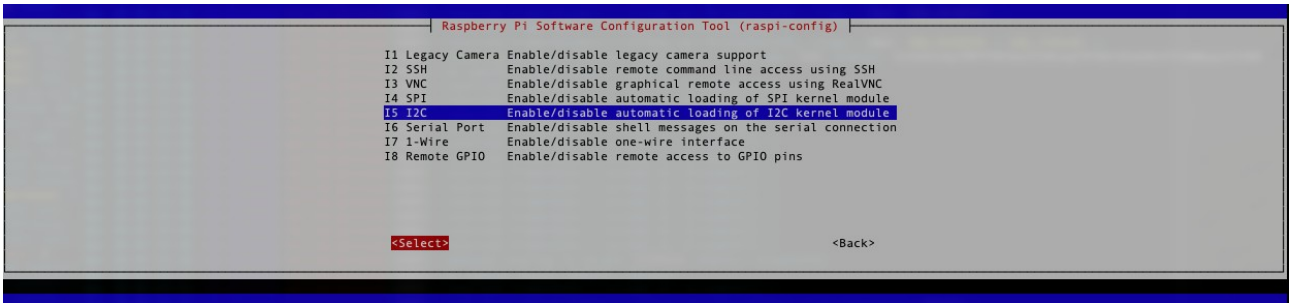
This will generate a public key. For those who are more interested in this topic, you just have to google a little.

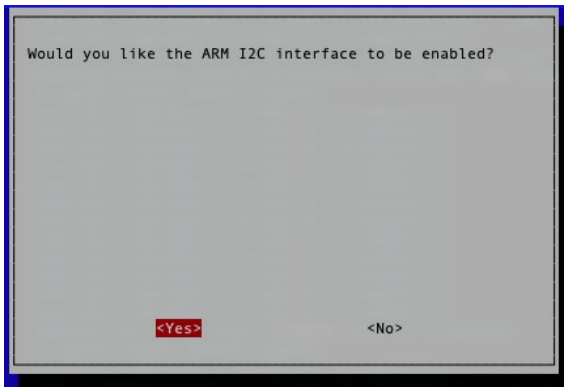



2.-:

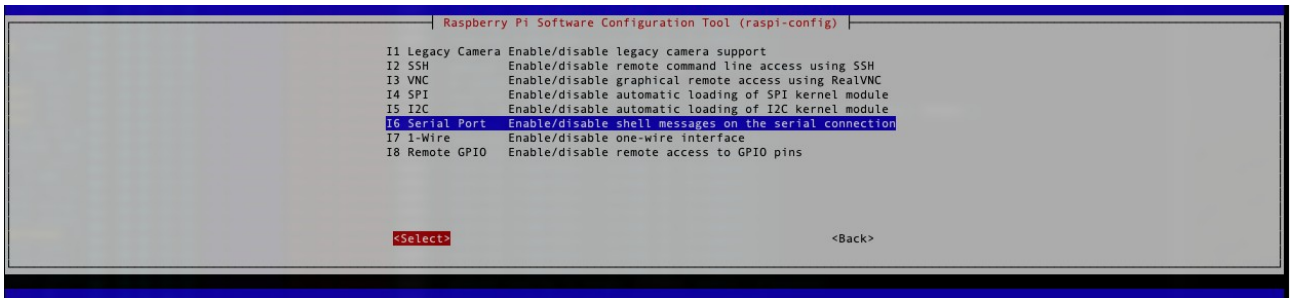
- **sudo raspi-config**

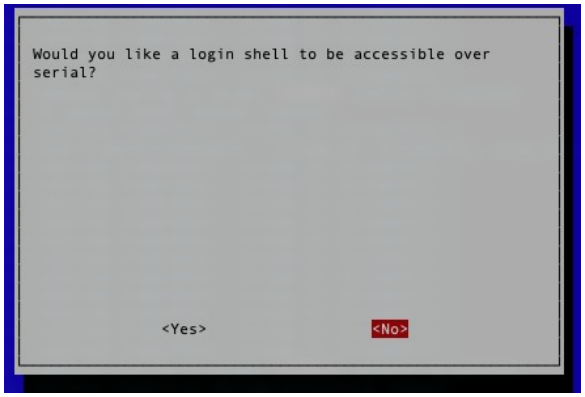


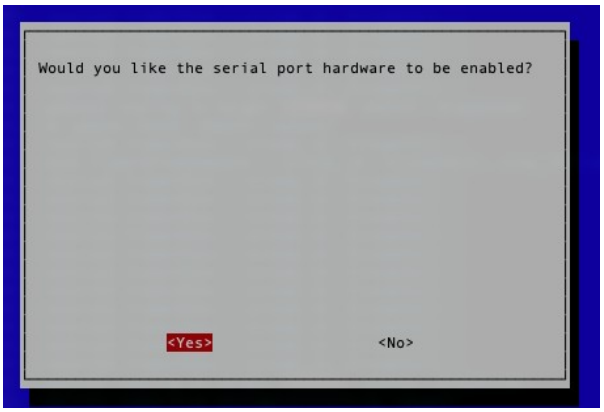



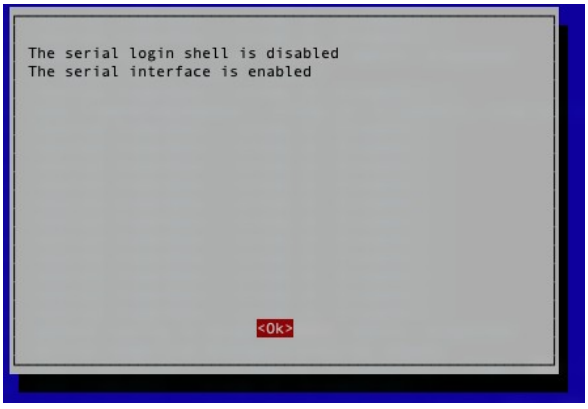

Would you like the ARM I2C interface to be enabled?
<Yes>
<No>

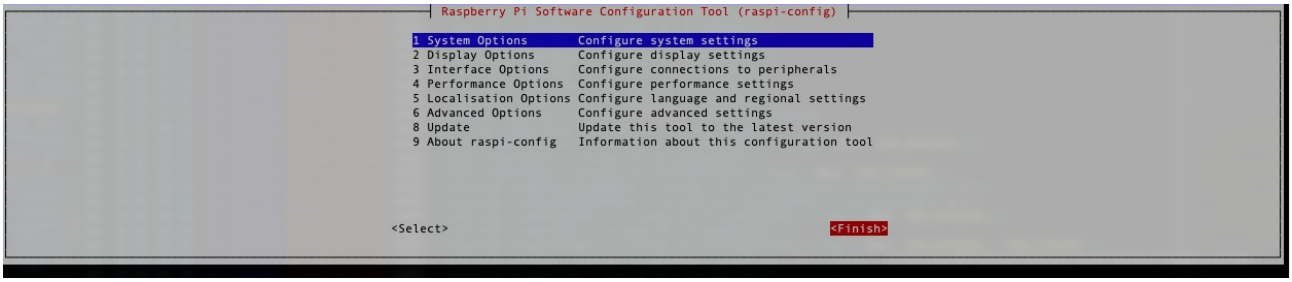

The ARM I2C interface is enabled
<Ok>

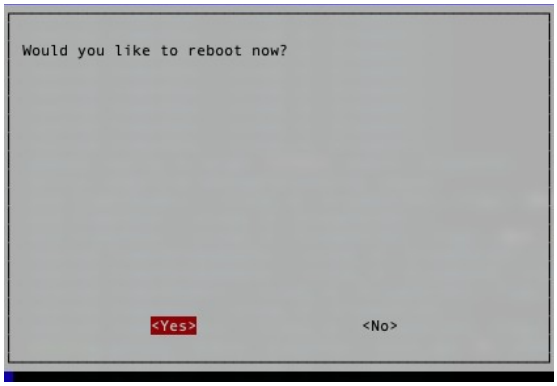

Raspberry Pi Software Configuration Tool (raspi-config)
I1 Legacy Camera Enable/disable legacy camera support
I2 SSH Enable/disable remote command line access using SSH
I3 VNC Enable/disable graphical remote access using RealVNC
I4 SPI Enable/disable automatic loading of SPI kernel module
I5 I2C Enable/disable automatic loading of I2C kernel module
I6 Serial Port Enable/disable shell messages on the serial connection
I7 1-Wire Enable/disable one-wire interface
I8 Remote GPIO Enable/disable remote access to GPIO pins
<Select>
<Back>


Would you like a login shell to be accessible over serial?
<Yes>
<No>


Would you like the serial port hardware to be enabled?
<Yes>
<No>


The serial login shell is disabled
The serial interface is enabled
<Ok>




3.- Edit **.bash_aliases**, I usually use **nano**, and put the following aliases:

- **nano ~/.bash_aliases**

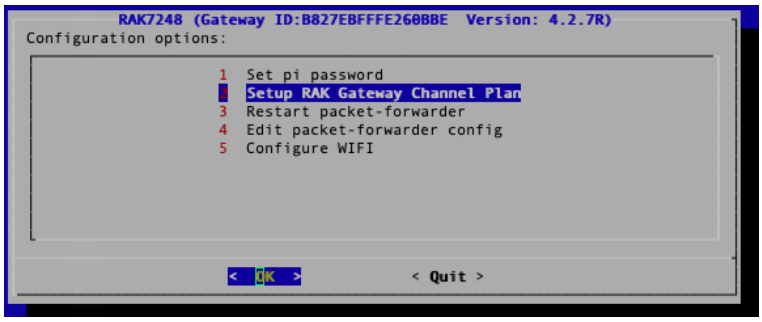
```
alias confrak='sudo nano /opt/ttn-gateway/packet_forwarder/lora_pkt_fwd/global_conf.json'
alias confrakloger='sudo nano /opt/ttn-gateway/lora_gateway/util_pkt_logger/global_conf.json'
alias raklog='sudo tail -f /var/log/syslog | grep ttn-gateway | ccze -A'
alias rakstop='sudo systemctl stop ttn-gateway'
alias rakstart='sudo systemctl start ttn-gateway'
alias heliumlog='sudo tail -f /var/log/syslog | ccze -A | grep helium_gateway'
alias heliumstop='sudo systemctl stop helium_gateway'
alias heliumstart='sudo systemctl start helium_gateway'
alias heliumrestart='sudo systemctl restart helium_gateway'
alias rakrestart='sudo systemctl restart ttn-gateway'
alias heliuminfo='helium_gateway --stdin info | ccze -A'
alias confhelium='sudo nano /etc/helium_gateway/settings.toml'
alias heliumversions='helium_gateway update list'
alias heliumverupdate='helium_gateway update download'
alias nrlog="pm2 logs node-red | ccze -A"
alias nrstop="pm2 stop node-red"
alias nrres="pm2 reload node-red;pm2 logs node-red | ccze -A; pm2 logs node-red | ccze -A"
alias nrstart="pm2 start node-red;pm2 logs node-red | ccze -A"
alias nrrestart="pm2 restart node-red;pm2 logs node-red | ccze -A"
alias nrconf="sudo nano /lib/systemd/system/nodered.service"
alias nract="bash <(curl -sL https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/resources/
update-nodejs-and-nodered)"
alias nrdir="cd .node-red/"
_dir() { [ ! -x "$(command -v ccze)" ] && sudo apt-get install ccze;  \dir "$@" | ccze -A;}; alias dir='_dir'
_ifconfig() { [ ! -x "$(command -v ccze)" ] && sudo apt-get install ccze; \ifconfig "$@" | ccze -A; }; alias
ifconfig='_ifconfig'
upgr() { [ ! -x "$(command -v ccze)" ] && sudo apt-get install ccze; echo "*** Updating"; sudo apt-get update | ccze
-A; echo "*** Upgrading"; sudo apt-get upgrade | ccze -A; }
alias ap2log="sudo journalctl -f -n 50 -u apache2 -o cat | ccze -A"
alias ap2stop="sudo service apache2 stop"
alias ap2start="sudo service apache2 start"
alias ap2restart="sudo service apache2 restart"
alias ap2reload="sudo systemctl reload apache2"
alias ngstop="sudo /etc/init.d/nginx stop"
alias ngstart="sudo /etc/init.d/nginx start"
alias ngstatus="sudo /etc/init.d/nginx status"
alias model='cat /proc/device-tree/model | ccze -A;echo'
alias cputype='cat /proc/device-tree/cpus/cpu@0/compatible | ccze -A;echo'
```

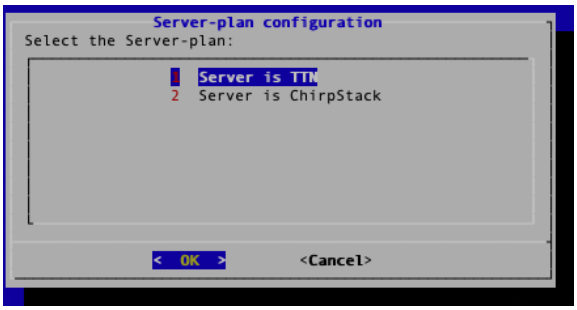
You will already know that to exit **nano** we use **ctr-X** and answer affirmatively when it asks us if we want to save the file.

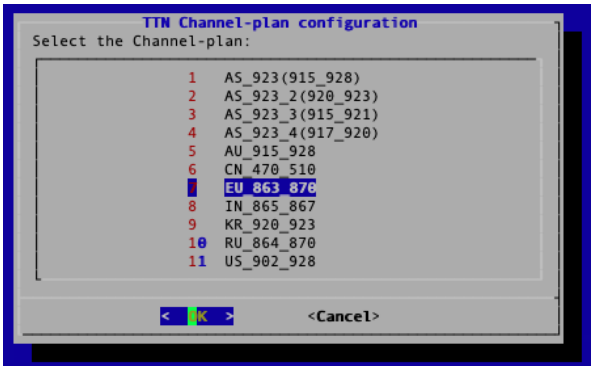
4.- Install the RAK software:

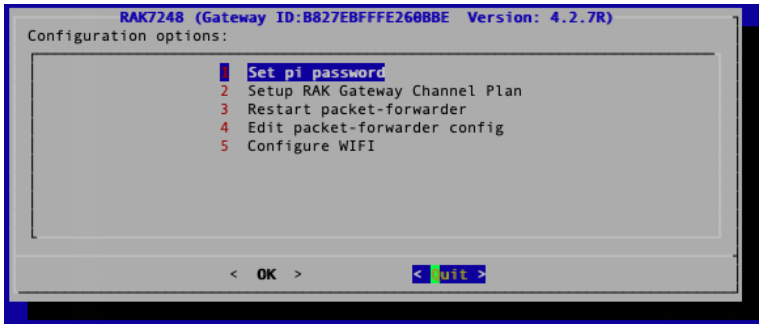
- sudo apt-get install git ccze mc macchanger
- git clone https://github.com/RAKWireless/rak_common_for_gateway.git
- cd ~/rak_common_for_gateway
- sudo ./install.sh
- sudo reboot

5. configure rak gateway:
- **sudo gateway-config**

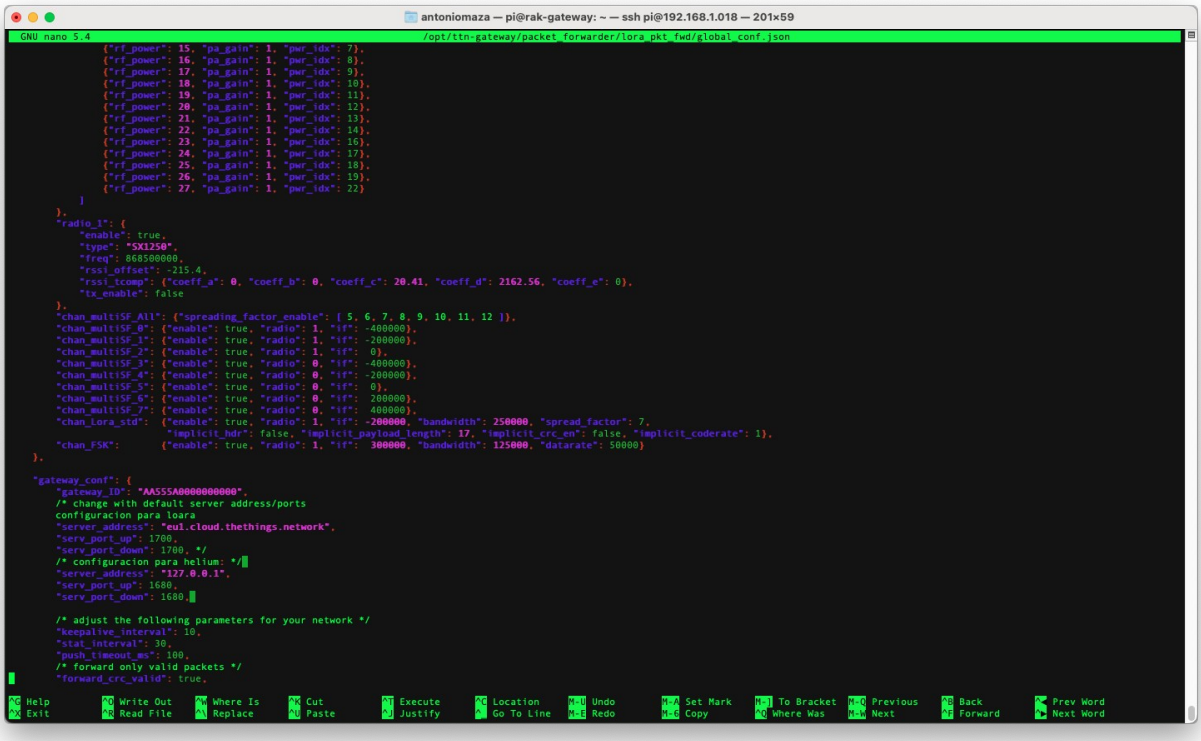





- **confrak**




I leave it like this to be able to play with **LoRa Wan** or **Helium.**

- **rakrestart**

6.- **Install the helium part:**

- **cd /tmp**
- **wget https://github.com/helium/gateway-rs/releases/download/v1.0.0-alpha.22/helium-gateway-v1.0.0-alpha.22-raspi234.deb**
- **sudo dpkg -i helium-gateway-v1.0.0-alpha.22-raspi234.deb**
- **cd**
- **sudo nano /etc/helium_gateway/settings.toml**
- **sudo systemctl restart helium_gateway**

7.- **Configure a fixed ip (optional, but very convenient):**

We connect to the raspberry and execute:

- **sudo nano /etc/dhcpcd.conf**

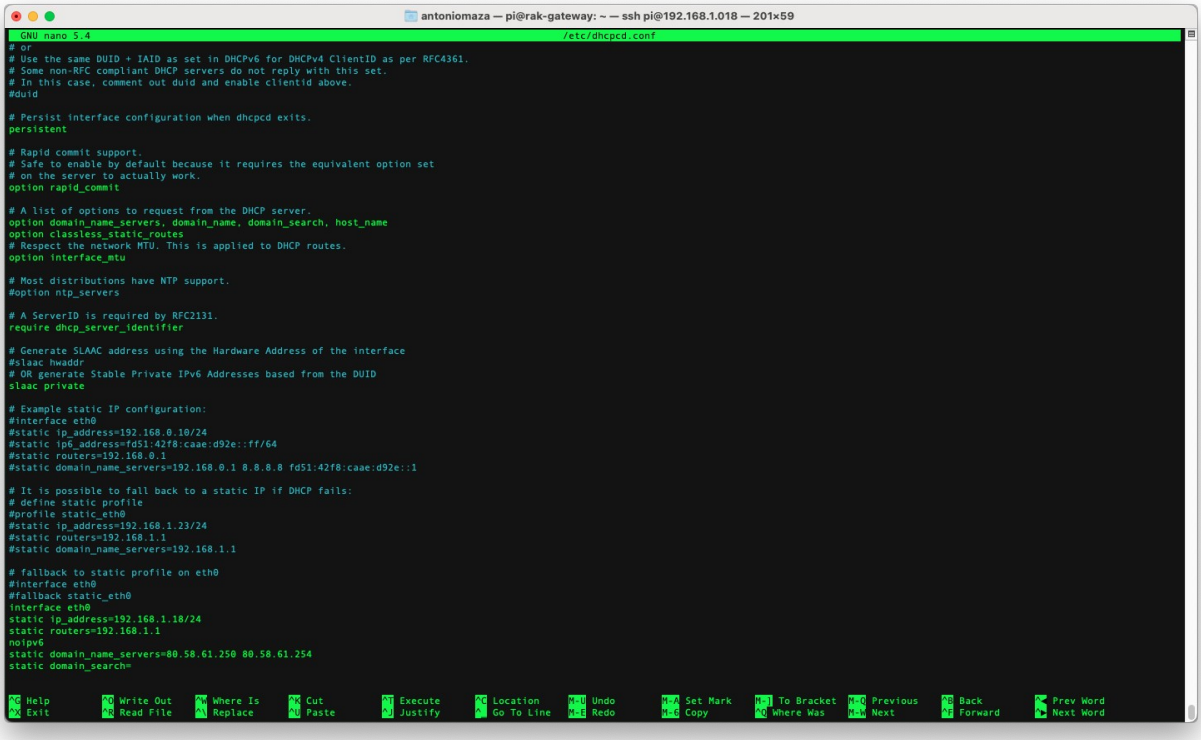
and add the following at the end:

interface eth0

```
static ip_address=192.168.1.18/24
static routers=192.168.1.1
noipv6
static domain_name_servers=80.58.61.250 80.58.61.254
static domain_search=
```

- **sudo nano -m /etc/network/interfaces**

auto lo
allow-hotplug wlan0
iface wlan0 inet static
address 192.168.1.105
netmask 255.255.255.0
broadcast 192.168.1.255
network 192.168.1.0
gateway 192.168.1.1
wpa-ssid "the ssid of your access point”
wpa-psk "your password"
wpa-ap-scan 1
wpa-scan-ssid 1
wpa-proto RSN
wpa-pairwise CCMP
wpa-key-mgmt WPA-PSK
dns-nameservers 80.58.61.250 80.58.61.254# interfaces(5) file used by ifup(8) and ifdown(8)




As I mentioned before, this is done with nano, and you know, once updated, how do we exit by saving the file.

and we give to restart the next time it will already be in that ip.

- **sudo reboot**

We can now use con018 and once we have logged in to play…

- **raklog**
- **heliumlog**
- **heliuminfo**
- **confrak**
- **confhelium**
- **heliuminfo**

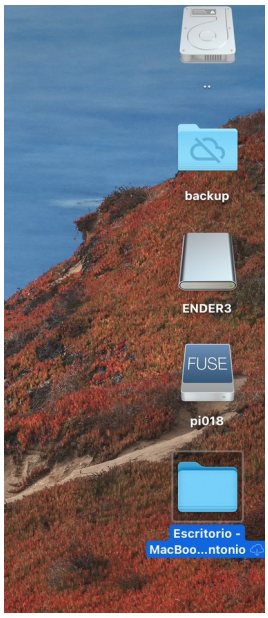
or any of the aliases we added earlier

***Bonus pack:***

In my case I already have the following in my shell environment
**alias conecta018='ssh waterbrain@192.168.1.018'**
and if we have some tool like **fuse** on the mac we can also add **alias connect018='sshfs waterbrain@192.168.1.18:/ pi018 -ovolname=pi018'**
Previously we must have in our home an empty folder with the name pi018

we can now use both an alias and another

If we connect **connect018**, we will see something similar to this on our desktop:




If we click on the pi018 disk:

**!MAGICi** As you can see all the content of the raspberry, we can do something similar with **FileCilla**, which is available for both mac and windows, but this seems more correct and functional to me.