

Ubuntu 22.04 para RaspberryPi 4 64bits no configura la red cableada

RaspberryPi 4B con Ubuntu 22.04

La verdad es que no me lo esperaba. No suelo configurar las raspberry con pantalla, y esta vez me fallo, y anda a ciegas ya que no tenía un cable **hdmi A-D** así que me sonaba raro.

Instalación del S.O.

Habia usado el [RaspberryPi Imager](#) para macos y había elegido como S.O., una **Ubuntu 22.04** de 64 bits, mas que nada porque la voy a usar con MariaDB y Elasticsearch y prefiero que sea con 64 bits.

Cuando instalé el S.O. en las tarjetas configuró con los datos de acceso, mi llave, el nombre de hosts y esas cosas que me hacen la vida más fácil cuando se trata de un serverless sin pantalla.

La cuestión es que no me funcionaba la red de cable, pero si me saltaba la wifi.

Análisis

“NOTA: Muchos sitios usan sudo porque están en modo user. Yo tras muchos años me siento más cómodo como root, porque sé lo que hago, pero no aconsejo el uso de root porque estoy muy habituado a arreglar desaguisados de “expertos” en sistema en producción. (Mis rasp no están en producción)

Tras entrar via wifi, observe que la tarjeta de red estaba reconocida pero desactivada.

```
abkrim@raspdev:~$ sudo su -
root@raspdev:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether d8:3a:dd:34:4d:6f brd ff:ff:ff:ff:ff:ff
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether d8:3a:dd:34:4d:70 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.82/24 metric 600 brd 192.168.1.255 scope global dynamic wlan0
        valid_lft 43061sec preferred_lft 43061sec
    inet6 fe80::da3a:ddff:fe34:4d70/64 scope link
        valid_lft forever preferred_lft forever

root@raspdev:~# networkctl
IDX LINK  TYPE   OPERATIONAL SETUP
  1 lo    loopback carrier   unmanaged
  2 eth0  ether   off        unmanaged
  3 wlan0 wlan    routable   configured

3 links listed.
```

Probar a levantarlo

Simple la prueba que funcionó y levanto la red cableada

```
root@raspdev:~# dhclient -v
Internet Systems Consortium DHCP Client 4.4.1
Copyright 2004-2018 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/wlan0/d8:3a:dd:34:4d:70
```

```
Sending on LPF/wlan0/d8:3a:dd:34:4d:70
Listening on LPF/eth0/d8:3a:dd:34:4d:6f
Sending on LPF/eth0/d8:3a:dd:34:4d:6f
Sending on Socket/fallback
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 3 (xid=0xbe3f7d64)
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 3 (xid=0x98bad722)
DHCPOFFER of 192.168.1.82 from 192.168.1.1
DHCPREQUEST for 192.168.1.82 on wlan0 to 255.255.255.255 port 67 (xid=0x647d3fbe)
DHCPACK of 192.168.1.82 from 192.168.1.1 (xid=0xbe3f7d64)
RTNETLINK answers: File exists
bound to 192.168.1.82 -- renewal in 18230 seconds.
root@raspdev:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether d8:3a:dd:34:4d:6f brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.81/24 brd 192.168.1.255 scope global dynamic eth0
        valid_lft 43195sec preferred_lft 43195sec
    inet6 fe80::da3a:ddff:fe34:4d6f/64 scope link
        valid_lft forever preferred_lft forever
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether d8:3a:dd:34:4d:70 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.82/24 metric 600 brd 192.168.1.255 scope global dynamic wlan0
        valid_lft 42684sec preferred_lft 42684sec
    inet6 fe80::da3a:ddff:fe34:4d70/64 scope link
        valid_lft forever preferred_lft forever
```

Luego lei un tip, [Fix Eth0 rpi ubuntut](#) pero su salida me parecio un poco fea.

Solución

Editar netplan teniendo en cuenta la advertencia de sobre escritura. En mi caso no quiero tenerla desactivada o eliminada de la configuración pero podría eliminar la wifi.

```
root@raspdev:~# cat /etc/netplan/50-cloud-init.yaml
# This file is generated from information provided by the datasource.  Changes
# to it will not persist across an instance reboot.  To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}

network:
  version: 2
  ethernets:
    eth0:
      dhcp4: yes
  wifis:
    renderer: networkd
    wlan0:
      access-points:
        DIWAN:
          password: e717960c10ec0b8216d0HJKHJKHJKHJKHJKH58b60d6f91493837a3a49
      dhcp4: true
      optional: true
```

Es necesario crear el otro fichero con `network: {config: disabled}`

```
root@raspdev:~# cat /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg
network: {config: disabled}
```

Ejecutamos `netplan apply` y voila ya lo tenemos.

Podemos reiniciar nuestra máquina y veremos que tenemos las dos redes.

Desactivación de la wifi sin eliminar de netplan

Bueno, en lugar de script extraños, usaremos el **Systemctl** para esto

Crear el fichero de servicio

`/etc/systemd/system/deactivate-wifi.service` con el contenido

[Unit]

After=network.target

[Service]

ExecStart=/usr/local/bin/off-wifi.sh

[Install]

WantedBy=default.target

Crear el fichero off-wifi.sh

Fichero `/usr/local/bin/off-wifi.sh`

```
#!/bin/bash
```

```
ip link set dev wlan0 down
```

Permisos y recarga del Systemctl

```
root@raspdev:~# chmod 664 /etc/systemd/system/disk-space-check.service
```

```
root@raspdev:~# chmod 664 /etc/systemd/system/deactivate-wifi.service.sh
```

```
root@raspdev:~# systemctl daemon-reload
```

Ya está listo para reiniciar y tras el reinicio eficaz con la red cableada, se apagará la wifi.

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

Revision #1

Created 15 August 2023 05:42:33 by Abkrim

Updated 15 August 2023 05:42:58 by Abkrim