JELLE DE LOECKE

CONFIGURE WIFI ON A MINIMAL DEBIAN 11 SERVER

By Jelle De Loecker on Jan 30, 2023

TLDR

configuring wifi on a minimal Debian server can be achieved by accessing the terminal and installing necessary packages, bringing up the wifi interface, configuring wpa-supplicant with the correct ESSID and credentials, and creating the necesary systemd services. This guide provides a step-by-step process to complete the task.

Warning

This guide was actually performed on a brand new *Proxmox* install on an old *Intel NUC* I had lying around, which is basically a minimal Debian 11 operating system.

I can't use a wired connection on my little server because there's no way to lay down a wire to where I'm planning on running the system. And for my use case, wifi should be enough.

This minimal install has no desktop environment (and it won't get one) and it is not using `NetworkManager` , so how do I get it to automatically connect to Wifi on boot? Let's go.

Access the terminal

First of all: get access to the terminal, and become root. Either by using a monitor and keyboard connected to the server, or by connecting to the server via SSH (on a temporary wired connection).

Then you'll need to make sure some packages have been installed:

```
apt install wpasupplicant wireless-tools
```

Bring the interface up

You can take a look at all the networks that have been enabled by issuing the following command:

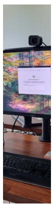
```
ip address
```

I get the following output:

```
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: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master vmbr0 state UP gr
    link/ether d4:5d:df:09:a7:4f brd ff:ff:ff:ff:ff:ff
```

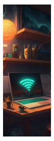
RECENT BLOGPOSTS

MAKE ELECTRON APPLICATI USE THE NATIVE WAYLAND RENDERER
Jun 05, 2024


THE ULTIMATE MULTI-MONITOR SETUP WIT THE FRAMEWO 16
Jun 01, 2024

PROMPTIN ABOUT YOURSELF
Nov 05, 2023

SETTING U A PROXMO VM TO ACT AS A VPN GATEWAY FOR OTHEI CONTAINEI
Feb 13, 2023


CONFIGUR WIFI ON A MINIMAL DEBIAN 11 SERVER
Jan 30, 2023

```
    altname enp0s31f6
3: wlp3s0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 00:1c:bf:fd:2a:aa brd ff:ff:ff:ff:ff:ff
4: vmbr0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qle
    link/ether d4:5d:df:09:a7:4f brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.238/24 scope global vmbr0
       valid_lft forever preferred_lft forever
    inet6 fe80::d65d:dfff:fe09:a74f/64 scope link
       valid_lft forever preferred_lft forever
```

`wlp3s0` is my wifi device. If your wifi device isn't there, you'll have to look for another tutorial that actually makes your wifi device work on Debian. (Maybe it's a driver issue, maybe there's a physical wifi kill switch on the system?)

Anyway, let's bring the interface up:

```
ip link set wlp3s0 up
```

When I do `ip address`, the lines regarding the wifi device will have changed a bit:

```
3: wlp3s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group defaul
    link/ether 00:1c:bf:fd:2a:aa brd ff:ff:ff:ff:ff:ff
```

You'll see **UP** is now mentioned in between the angled brackets, but the state is still **DOWN**.

The **UP** between the angled brackets means that the device's link layer is operational. The **DOWN** after the `state` keyword means the device's *network* layer is down. Which is correct, because it's not connected to anything yet.

Configure WPA supplicant

We need to configure the actual connection.

If you don't know your ESSID (wifi network name), you can scan for it:

```
iwlist wlp3s0 scan | grep ESSID
```

At my office, I get this output:

```
ESSID:"telenet-3A41B"
ESSID:"telenet-2618A45"
ESSID:"HP-Print-9D-Photosmart 5520"
ESSID:"CT"
ESSID:"telenet-FC33D"
ESSID:"HP-Print-CB-Officejet Pro 8610"
ESSID:"DIRECT-c4-HP M252 LaserJet"
ESSID:"Eleven Ways"
ESSID:"Eleven Ways"
ESSID:"CT"
ESSID:"telenet-ap-8248214"
```

In this example **"Eleven Ways"** is double because there are 2 APs at my office.

Let's configure the connection, and save it to a file.

```
wpa_passphrase "Eleven Ways" wifipassword | tee /etc/wpa_supplicant/wpa_supplicant.conf
```

You can now test this configuration file to see if it works:

```
wpa_supplicant -c /etc/wpa_supplicant/wpa_supplicant.conf -i wlp3s0
```

If something is **wrong**, like a wrong password, you'll see something like this:

```
Successfully initialized wpa_supplicant
wlp3s0: SME: Trying to authenticate with f0:2f:74:c8:d5:54 (SSID='Eleven Ways' freq=5500 MHz)
wlp3s0: Trying to associate with f0:2f:74:c8:d5:54 (SSID='Eleven Ways' freq=5500 MHz)
wlp3s0: Associated with f0:2f:74:c8:d5:54
wlp3s0: CTRL-EVENT-SUBNET-STATUS-UPDATE status=0
wlp3s0: CTRL-EVENT-DISCONNECTED bssid=f0:2f:74:c8:d5:54 reason=2
wlp3s0: WPA: 4-Way Handshake failed - pre-shared key may be incorrect
```

If it does work, you'll get this:

```
Successfully initialized wpa_supplicant
wlp3s0: SME: Trying to authenticate with f0:2f:74:c8:d5:54 (SSID='Eleven Ways' freq=5500 MHz)
wlp3s0: Trying to associate with f0:2f:74:c8:d5:54 (SSID='Eleven Ways' freq=5500 MHz)
wlp3s0: Associated with f0:2f:74:c8:d5:54
wlp3s0: CTRL-EVENT-SUBNET-STATUS-UPDATE status=0
wlp3s0: WPA: Key negotiation completed with f0:2f:74:c8:d5:54 [PTK=CCMP GTK=CCMP]
wlp3s0: CTRL-EVENT-CONNECTED - Connection to f0:2f:74:c8:d5:54 completed [id=0 id_str=]
```

The difference is subtle. The first fails, because the `Handshake failed` , the second one is correct because `Key negotiation completed`

Now we have to edit the `wpa_supplicant.conf` file:

```
nano /etc/wpa_supplicant/wpa_supplicant.conf
```

Make it look like the following (adding the `ctrl_interface` and `update_config` lines)

```
ctrl_interface=/run/wpa_supplicant
update_config=1

network={
        ssid="Eleven Ways"
        #psk="wifipassword"
        psk=GENERATEDCODE
}
```

If your network is *hidden*, you'll have to add `scan_ssid=1` to the `network` block (paste it under the `psk=CODE` line)

Finally: create a symlink to the configuration file with the name of your wireless device added to it:

```
ln -s /etc/wpa_supplicant/wpa_supplicant.conf /etc/wpa_supplicant/wpa_supplicant-wlp3s0.conf
```

This is needed because the systemd service we're going to create will look for a `wpa_supplicant` file ending with the name of the interface.

Add Systemd services

This is where I ran into problems with existing guides. They claim you can use the normal `wpa_supplicant` service, but that's not correct: that service is meant to be used with `NetworkManager` , which is not going to be used here.

So instead we instruct systemd to enable the `wpa_supplicant@` service especially for our interface, like this:

```
systemctl enable wpa_supplicant@wlp3s0
```

We're not there yet. This might allow us to connect to the wifi network, we still need an IP address.

If you want it to get an IP address automatically, you'll have to add the following DHCP service file manually.

```
nano /etc/systemd/system/dhclient.service
```

And paste in the following:

> ⚠ **Warning**
>
> Don't forget to use the name of your own interface! In my case this was `wlp3s0`

```
[Unit]
Description=DHCP Client
Before=network.target
After=wpa_supplicant.service

[Service]
Type=forking
ExecStart=/sbin/dhclient wlp3s0 -v
ExecStop=/sbin/dhclient wlp3s0 -r
Restart=always

[Install]
WantedBy=multi-user.target
```

And enable it:

```
systemctl enable dhclient
```

Now we need to restart the services to get them working:

```
systemctl restart wpa_supplicant
systemctl restart networking
```

Now your wifi will be up and have an address!

```
ip a
```

Results in:

```
3: wlp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default ql
    link/ether 00:1c:bf:fd:2a:aa brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.122/24 brd 192.168.50.255 scope global dynamic wlp3s0
       valid_lft 86274sec preferred_lft 86274sec
    inet6 2a02:1812:1606:e300:21c:bfff:fefd:2aaa/64 scope global dynamic mngtmpaddr
       valid_lft 252442sec preferred_lft 79642sec
    inet6 fe80::21c:bfff:fefd:2aaa/64 scope link
       valid_lft forever preferred_lft forever
```

Let me know if it works, or if you ran into any problems.

PREVIOUS **NEXT**

COMMENTS

Name

Email

Website

Body