Installing NixOS on a Raspberry Pi

Contents

This tutorial assumes you have a Raspberry Pi 4 Model B with 4GB RAM.

Before starting this tutorial, make sure you have all the necessary hardware:

- HDMI cable/adapter.
- 8GB+ SD card.
- SD card reader (in case your machine doesn't have an SD slot).
- Power cable for your Raspberry Pi.
- USB keyboard.

> **ℹ Note**
>
> This tutorial was written for the Raspberry Pi 4B. Using a previously supported model like the 3B or 3B+ is possible with some modifications to this tutorial.

Skip to main content

Booting NixOS live image

> **ℹ Note**
>
> Booting from USB may require an EEPROM firmware upgrade. This tutorial boots from
> an SD card to avoid such hiccups.

To prepare the AArch64 image on another device with Nix, run the following commands:

```
$ nix-shell -p wget zstd

[nix-shell:~]$ wget https://hydra.nixos.org/build/226381178/download/1/nixos-sd-image-
[nix-shell:~]$ unzstd -d nixos-sd-image-23.11pre500597.0fbe93c5a7c-aarch64-linux.img.z
[nix-shell:~]$ dmesg --follow
```

> **ℹ Note**
>
> You can download a recent image from Hydra, clicking on the latest successful build
> (marked with a green checkmark), and copying the link to the build product image.

> **ℹ Note**
>
> It may be more convenient to use a software like Etcher to flash the image to your SD
> card if you are on a system where it's available.

Your terminal should be printing kernel messages as they come in.

Plug in your SD card and your terminal should print what device it got assigned, for example
`/dev/sdX`.

Press `Ctrl` + `C` to stop `dmesg --follow`.

Copy NixOS to your SD card by replacing `sdX` with the name of your device in the following
command:

```
[nix-shell:~]$ sudo dd if=nixos-sd-image-23.11pre500597.0fbe93c5a7c-aarch64-linux.img
```

Skip to main content

Once that command exits, **move the SD card into your Raspberry Pi and power it on**.

You should be greeted with a fresh shell!

In case the image doesn't boot, it's worth updating the firmware and booting the image again.

Getting internet connection

Run `sudo -i` to get a root shell for the rest of the tutorial.

At this point you'll need an internet connection. If you can use an ethernet cable, plug it in and skip to the next section.

If you're connecting to wifi, run `iwconfig` to find the name of your wireless network interface. If it's `wlan0`, replace `SSID` and `passphrase` with your data and run:

```
# wpa_supplicant -B -i wlan0 -c <(wpa_passphrase 'SSID' 'passphrase') &
```

Once you see in your terminal that connection is established, run `host nixos.org` to check that the DNS resolves correctly.

In case you've made a typo, run `pkill wpa_supplicant` and start over.

Updating firmware

To benefit from updates and bug fixes from the vendor, we'll start by updating Raspberry Pi firmware:

```
# nix-shell -p raspberrypi-eeprom
# mount /dev/disk/by-label/FIRMWARE /mnt
# BOOTFS=/mnt FIRMWARE_RELEASE_STATUS=stable rpi-eeprom-update -d -a
```

Installing and configuring NixOS

Now we'll install NixOS with our own configuration, here creating a `guest` user and enabling

Skip to main content

In the `let` binding below, change the value of the `SSID` and `SSIDpassword` variables to the `SSID` and `passphrase` values you used previously:

```
1 { config, pkgs, lib, ... }:
2
3 let
4   user = "guest";
5   password = "guest";
6   SSID = "mywifi";
7   SSIDpassword = "mypassword";
8   interface = "wlan0";
9   hostname = "myhostname";
10 in {
11
12   boot = {
13     kernelPackages = pkgs.linuxKernel.packages.linux_rpi4;
14     initrd.availableKernelModules = [ "xhci_pci" "usbhid" "usb_storage" ];
15     loader = {
16       grub.enable = false;
17       generic-extlinux-compatible.enable = true;
18     };
19   };
20
21   fileSystems = {
22     "/" = {
23       device = "/dev/disk/by-label/NIXOS_SD";
24       fsType = "ext4";
25       options = [ "noatime" ];
26     };
27   };
28
29   networking = {
30     hostName = hostname;
31     wireless = {
32       enable = true;
33       networks."${SSID}".psk = SSIDpassword;
34       interfaces = [ interface ];
35     };
36   };
37
38   environment.systemPackages = with pkgs; [ vim ];
39
40   services.openssh.enable = true;
41
42   users = {
43     mutableUsers = false;
44     users."${user}" = {
45       isNormalUser = true;
46       password = password;
47       extraGroups = [ "wheel" ];
48     };
49   };
```

Skip to main content

```
52    system.stateVersion = "23.11";
53 }
```

To save time on typing the whole configuration, download it:

```
# curl -L https://tinyurl.com/tutorial-nixos-install-rpi4 > /etc/nixos/configuration.r
```

> **ⓘ Note**
>
> Credentials you write into a NixOS configuration will be stored in plain text in your
> `/nix/store` when that configuration is built.
>
> If you **don't** want this to happen, you can enter your credentials at a console or use
> one of the community's solutions for encrypted secrets.

Due to the way the `nixos-sd-image` is designed, NixOS is actually *already installed* at this point,
so we only need to `nixos-rebuild` with our new configuration:

```
# nixos-rebuild boot
# reboot
```

If your system doesn't boot, select the oldest configuration in the bootloader menu to get back
to the live image and start over.

Making changes

It booted, congratulations!

To make further changes to the configuration, search through NixOS options, edit
`/etc/nixos/configuration.nix`, and update your system:

```
$ sudo -i
# nixos-rebuild switch
```

Skip to main content

Next steps

- Once you have a working OS, try upgrading it with `nixos-rebuild switch --upgrade` to install more recent package versions, and reboot to the old configuration if something broke.

- To enable hardware acceleration for a nice graphical desktop experience, add the `nixos-hardware` module to your configuration:

```
1 imports = [
2    "${fetchTarball "https://github.com/NixOS/nixos-hardware/tarball/master"}/ras
3 ];
```

  We recommend pinning the reference to `nixos-hardware`: Pinning Nixpkgs

- To tweak bootloader options affecting hardware, see `config.txt` options. You can change these options by running `mount /dev/disk/by-label/FIRMWARE /mnt` and opening `/mnt/config.txt`.