

Get Started with NixOS

Now that we have learned the basics of the Nix language, we can start using it to configure our NixOS system. The default configuration file for NixOS is located at `/etc/nixos/configuration.nix`. This file contains all the declarative configuration for the system, including settings for the time zone, language, keyboard layout, network, users, file system, and boot options.

To modify the system state in a reproducible manner (which is highly recommended), we need to manually edit the `/etc/nixos/configuration.nix` file and then execute `sudo nixos-rebuild switch` to apply the modified configuration. This command generates a new system environment based on the modified configuration file, sets the new environment as the default one, and preserves the previous environment in the boot options of grub/systemd-boot. This ensures that we can always roll back to the old environment even if the new one fails to start.

While `/etc/nixos/configuration.nix` is the classic method for configuring NixOS, it relies on data sources configured by `nix-channel` and lacks a version-locking mechanism, making it challenging to ensure the reproducibility of the system. A better approach is to use Flakes, which provides reproducibility and facilitates configuration management.

In this section, we will first learn how to manage NixOS using the classic method (`/etc/nixos/configuration.nix`), and then we will explore the more advanced Flakes.

Configuring the System using `/etc/nixos/configuration.nix`

The `/etc/nixos/configuration.nix` file is the default and classic method for configuring NixOS. While it lacks some of the advanced features of Flakes, it is still widely used and provides flexibility in system configuration.

To illustrate how to use `/etc/nixos/configuration.nix`, let's consider an example where we enable SSH and add a user named `ryan` to the system. We can achieve this by adding the following content to `/etc/nixos/configuration.nix`:

```
1  # Edit this configuration file to define what should be installed on
2  # your system. Help is available in the configuration.nix(5) man page
3  # and in the NixOS manual (accessible by running 'nixos-help').
4  { config, pkgs, ... }:
5
6  {
7      imports =
8          [ # Include the results of the hardware scan.
9            ./hardware-configuration.nix
10          ];
11
12      # Omit previous configuration settings...
13
14      # Add user 'ryan'
15      users.users.ryan = {
16          isNormalUser = true;
17          description = "ryan";
18          extraGroups = [ "networkmanager" "wheel" ];
19          openssh.authorizedKeys.keys = [
20              # Replace with your own public key
21              "ssh-ed25519 <some-public-key> ryan@ryan-pc"
22          ];
23          packages = with pkgs; [
24              firefox
25              # thunderbird
26          ];
27      };
28
29      # Enable the OpenSSH daemon.
30      services.openssh = {
31          enable = true;
32          settings = {
33              X11Forwarding = true;
34              PermitRootLogin = "no"; # disable root login
35              PasswordAuthentication = false; # disable password login
36          };
37          openFirewall = true;
38      };
39
40      # Omit the rest of the configuration...
41  }
```

In this configuration, we declare our intention to enable the openssh service, add an SSH public key for the user 'ryan', and disable password login.

To deploy the modified configuration, run `sudo nixos-rebuild switch`. This command will apply the changes, generate a new system environment, and set it as the default. You can now log in to the system using SSH with the configured SSH keys.

You can always try to add `--show-trace --print-build-logs --verbose` to the `nixos-rebuild` command to get the detailed error message if you encounter any errors during the deployment.

Remember that any reproducible changes to the system can be made by modifying the `/etc/nixos/configuration.nix` file and deploying the changes with `sudo nixos-rebuild switch`.

To find configuration options and documentation:

- Use search engines like Google, e.g., search for [Chrome NixOS](#) to find NixOS-related information about Chrome. The NixOS Wiki and the source code of Nixpkgs are usually among the top results.
- Utilize the [NixOS Options Search](#) to search for keywords.
- Refer to the [Configuration section](#) in the NixOS Manual for system-level configuration documentation.
- Search for keywords directly in the source code of [nixpkgs](#) on GitHub.

References

- [Overview of the NixOS Linux distribution](#)

11 reactions



[0 comments](#) – powered by [giscus](#)

