

NixOS Installation Tutorial: Screw Imperative, Declarative go brrrrr

Lab Sister Selection 2025

August 15, 2025

What you're referring to as Linux...

You merely adopted Linux. I was born in GNU/Linux, molded by it. This tutorial will guide you through installing NixOS, the purely functional Linux distribution that will change how you think about system configuration forever.

Contents

1	Introduction to NixOS	3
2	Prerequisites	3
3	Step-by-Step Installation Guide	3
3.1	Step 1: Booting from ISO	3
3.2	Step 2: Network Configuration (Optional if use Wired Connection)	4
3.3	Step 3: Disk Partitioning	4
3.4	Step 4: System Configuration	4
3.5	Step 5: Installation Process	8
3.6	Step 6: Post-Installation Configuration	8
3.7	Step 7: Desktop Customization	8
3.7.1	Setting Custom Wallpaper	8
3.7.2	Creating the Demo Script	9
4	Verification Screenshots	10
4.1	Desktop Environment	10
4.2	System Information	10
4.3	Applications Working	11
5	Bonus Features Implemented	11
5.1	Wayland Support	11
5.2	Wine and LINE Installation	11
5.3	Open-Source Applications	11
6	Conclusion	12
7	Additional Resources	12

8 Documentation

13

1 Introduction to NixOS

NixOS is a Linux distribution built around the Nix package manager. Unlike traditional distributions, NixOS uses a purely functional approach to package and configuration management. This means:

- **Declarative configuration:** Your entire system is described in configuration files
- **Reproducible builds:** The same configuration always produces the same system
- **Atomic upgrades and rollbacks:** System changes are transactional
- **No dependency hell:** Multiple versions of packages can coexist

2 Prerequisites

Before starting the installation, ensure you have:

- physical computer with at least 4GB RAM
- 20GB of available disk space
- Stable internet connection
- NixOS ISO image downloaded from <https://nixos.org/download.html>
- Bootable drive with NixOS ISO image installed using rufus or other tools

3 Step-by-Step Installation Guide

3.1 Step 1: Booting from ISO

1. Reboot or Turn on your machine after plugging the bootable disk
2. Go to BIOS and change boot option to the newly plugged drive
3. Wait for the system to load into the live environment

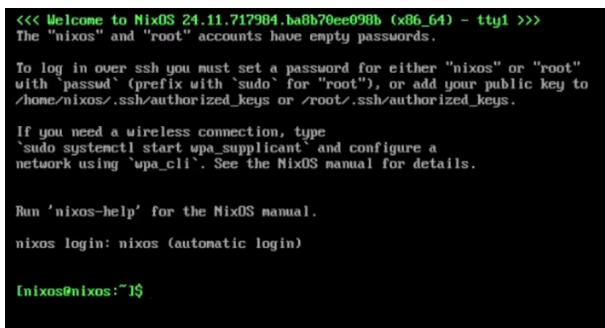


Figure 1: First Boot into Live Environment

3.2 Step 2: Network Configuration (Optional if use Wired Connection)

```

1 # Check network connectivity
2 ping -c 3 8.8.8.8
3
4 # If needed, configure wifi
5 sudo systemctl start wpa_supplicant
6 sudo wpa_cli
7 > add_network
8 > set_network 0 ssid "YOUR_WIFI_NAME"
9 > set_network 0 psk "YOUR_WIFI_PASSWORD"
10 > enable_network 0
11 > quit

```

Listing 1: Network setup commands

3.3 Step 3: Disk Partitioning

We'll create a simple partition layout with UEFI support:

```

1 # Identify the target disk
2 lsblk
3
4 # Create partitions (assuming /dev/nvme1n1)
5 # Use whichever partition tool that you like
6 cgdisk

```

Listing 2: Disk partitioning with parted

UEFI	512MB
SWAP	8GB
RootFS	Sisanya

Follow the size guidance above, the swap partitions sizes depends on sizes of computer memory's.

```

1 # Format partitions
2 mkfs.fat -F 32 /dev/nvmen1p1
3 mkfs.ext4 /dev/nvmen1p2
4 mkswap /dev/nvmen1p3

```

Then, mount it just like below

```

1 mount /dev/nvmen1p3 /mnt
2 mkdir -p boot
3 mount /dev/nvmen1p1 /mnt/boot
4 swapon /dev/nvmen1p2

```

3.4 Step 4: System Configuration

Generate the initial configuration:

```

1 # Generate hardware configuration
2 sudo nixos-generate-config --root /mnt
3
4 # Edit the configuration file

```

```
5 sudo nano /mnt/etc/nixos/configuration.nix
```

Listing 3: Generate NixOS configuration

Here's our complete NixOS configuration:

```
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 # on
4 # https://search.nixos.org/options and in the NixOS manual (nixos-help).
5
6 { config, lib, pkgs, ... }:
7
8 {
9   imports =
10   [
11     # Include the results of the hardware scan.
12     ./hardware-configuration.nix
13   ];
14
15   boot.loader.efi.canTouchEfiVariables = true;
16   boot.loader.grub.enable = true;
17   boot.loader.grub.efiSupport = true;
18   boot.loader.grub.useOSProber = true;
19   boot.loader.grub.extraConfig = ''
20     GRUB_DISABLE_OS_PROBER = false
21     insmod all_video
22     insmod jpeg
23     '';
24   boot.loader.grub.device = "nodev";
25   boot.loader.efi.efiSysMountPoint = "/boot";
26   boot.loader.grub.splashImage = "/boot/grub/hutao.jpeg";
27   # networking.hostName = "nixos"; # Define your hostname.
28   # Pick only one of the below networking options.
29   # networking.wireless.enable = true; # Enables wireless support via
30   # wpa_supplicant.
31   # networking.networkmanager.enable = true; # Easiest to use and most
32   # distros use this by default.
33
34   # Set your time zone.
35   time.timeZone = "Asia/Jakarta";
36
37   # Configure network proxy if necessary
38   # networking.proxy.default = "http://user:password@proxy:port/";
39   # networking.proxy.noProxy = "127.0.0.1,localhost,internal.domain";
40
41   # Select internationalisation properties.
42   # i18n.defaultLocale = "en_US.UTF-8";
43   # console = {
44     #   font = "Lat2-Terminus16";
45     #   keyMap = "us";
46     #   useXkbConfig = true; # use xkb.options in tty.
47   # };
48
49   # Enable the X11 windowing system.
50   services.xserver.enable = true;
51   services.xserver.videoDrivers = [ "amdgpu" ];
52   services.xserver.displayManager.gdm.enable = true;
53   services.xserver.desktopManager.gnome.enable = true;
54   services.xserver.displayManager.gdm.wayland = true;
```

```
52 programs.hyprland.enable = true;
53
54 # Configure keymap in X11
55 # services.xserver.xkb.layout = "us";
56 # services.xserver.xkb.options = "eurosign:e,caps:escape";
57
58 # Enable CUPS to print documents.
59 # services.printing.enable = true;
60
61 # Enable sound.
62 services.pulseaudio.enable = false;
63 # OR
64 services.pipewire = {
65   enable = true;
66   pulse.enable = true;
67   alsa.enable = true;
68   alsa.support32Bit = true;
69   jack.enable = true;
70 };
71
72 security.rtkit.enable = true;
73
74 # Enable touchpad support (enabled default in most desktopManager).
75 # services.libinput.enable = true;
76
77 programs.zsh.enable = true;
78
79 # Define a user account. Don't forget to set a password with
80 #     passwd .
81 users.users.HutaoHusband = {
82   isNormalUser = true;
83   extraGroups = [ "wheel" ]; # Enable      sudo      for the user.
84   shell = pkgs.zsh;
85   packages = with pkgs; [
86     tree
87   ];
88   password = "2121";
89 };
90
91 # programs.firefox.enable = true;
92
93 # List packages installed in system profile.
94 # You can use https://search.nixos.org/ to find more packages (and
95 # options).
96 environment.systemPackages = with pkgs; [
97   vim # Do not forget to add an editor to edit configuration.nix! The
98   # Nano editor is also installed by default.
99   wget
100  firefox
101  vscode
102  os-prober
103  neofetch
104  git
105  curl
106  mpv
107  aria2
108  yt-dlp
109  ffmpeg
```

```
107   fzf
108   patch
109   pkgs.gnome-screenshot
110   tmux
111   kitty
112   picom
113   desktop-file-utils
114   zsh
115   wine
116   zig
117   winetricks
118 ];
119 nixpkgs.config.allowUnfree = true;
120
121 programs.dconf.enable = true;
122
123 # Some programs need SUID wrappers, can be configured further or are
124 # started in user sessions.
125 # programs.mtr.enable = true;
126 # programs.gnupg.agent = {
127 #   enable = true;
128 #   enableSSHSupport = true;
129 # };
130
131 # List services that you want to enable:
132
133 # Enable the OpenSSH daemon.
134 services.openssh.enable = true;
135 networking.networkmanager.enable = true;
136 # Open ports in the firewall.
137 # networking.firewall.allowedTCPPorts = [ ... ];
138 # networking.firewall.allowedUDPPorts = [ ... ];
139 # Or disable the firewall altogether.
140 # networking.firewall.enable = false;
141
142 # Copy the NixOS configuration file and link it from the resulting
143 # system
144 # (/run/current-system/configuration.nix). This is useful in case you
145 # accidentally delete configuration.nix.
146 # system.copySystemConfiguration = true;
147
148 # This option defines the first version of NixOS you have installed on
149 # this particular machine,
150 # and is used to maintain compatibility with application data (e.g.
151 # databases) created on older NixOS versions.
152 #
153 # Most users should NEVER change this value after the initial install,
154 # for any reason,
155 # even if you've upgraded your system to a new NixOS release.
156 #
157 # This value does NOT affect the Nixpkgs version your packages and OS
158 # are pulled from,
159 # so changing it will NOT upgrade your system - see https://nixos.org/
160 # manual/nixos/stable/#sec-upgrading for how
161 # to actually do that.
162 #
163 # This value being lower than the current NixOS release does NOT mean
164 # your system is
```

```

158 # out of date, out of support, or vulnerable.
159 #
160 # Do NOT change this value unless you have manually inspected all the
161 # changes it would make to your configuration,
162 # and migrated your data accordingly.
163 #
164 # For more information, see man configuration.nix or https://nixos.org
165 #/manual/nixos/stable/options#opt-system.stateVersion .
165 system.stateVersion = "25.05"; # Did you read the comment?
165 }

```

Listing 4: Complete NixOS configuration.nix

3.5 Step 5: Installation Process

```

1 # Install NixOS
2 sudo nixos-install
3
4 # Set root password when prompted
5 # Enter password for root user
6
7 # Reboot into the new system
8 sudo reboot

```

Listing 5: NixOS installation commands

3.6 Step 6: Post-Installation Configuration

After rebooting and logging in as the 'HutaoHusband' or based on your config' user:

```

1 # Set user password
2 sudo passwd sister
3
4 # Update the system
5 sudo nixos-rebuild switch

```

Listing 6: Post-installation setup

3.7 Step 7: Desktop Customization

3.7.1 Setting Custom Wallpaper

1. Download your favorite anime character wallpaper
2. Open GNOME Settings → Background
3. Select your custom wallpaper image
4. Apply the changes



Figure 2: Custom anime wallpaper applied to desktop

3.7.2 Creating the Demo Script

Create a bash script for the demonstration:

```
1 #!/bin/bash
2
3 # Install and run neofetch
4 nix-shell -p neofetch --run neofetch
5
6 echo "NixOS installation complete!"
7 echo "System specifications:"
8 echo "- Distribution: NixOS"
9 echo "- Desktop: GNOME on Wayland"
10 echo "- Package Manager: Nix"
11 echo "- User: sister (unprivileged)"
```

Listing 7: Demo script (demo.sh)

Make it executable and run:

```
1 chmod +x demo.sh
2 ./demo.sh
```

4 Verification Screenshots

4.1 Desktop Environment

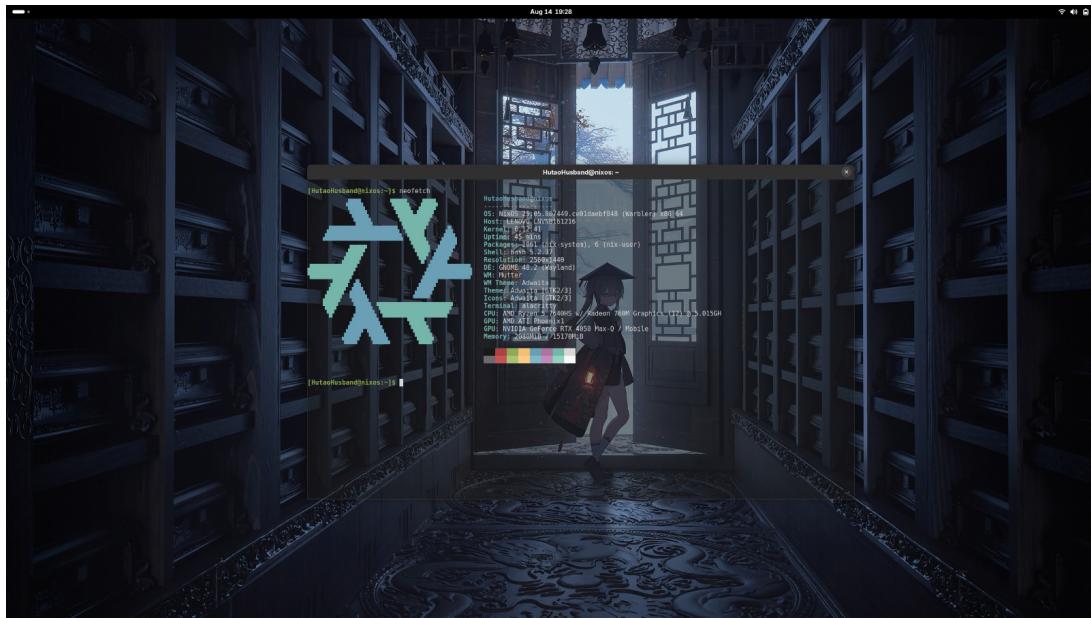


Figure 3: NixOS desktop with GNOME environment and custom wallpaper

4.2 System Information



Figure 4: Neofetch output showing system information

4.3 Applications Working

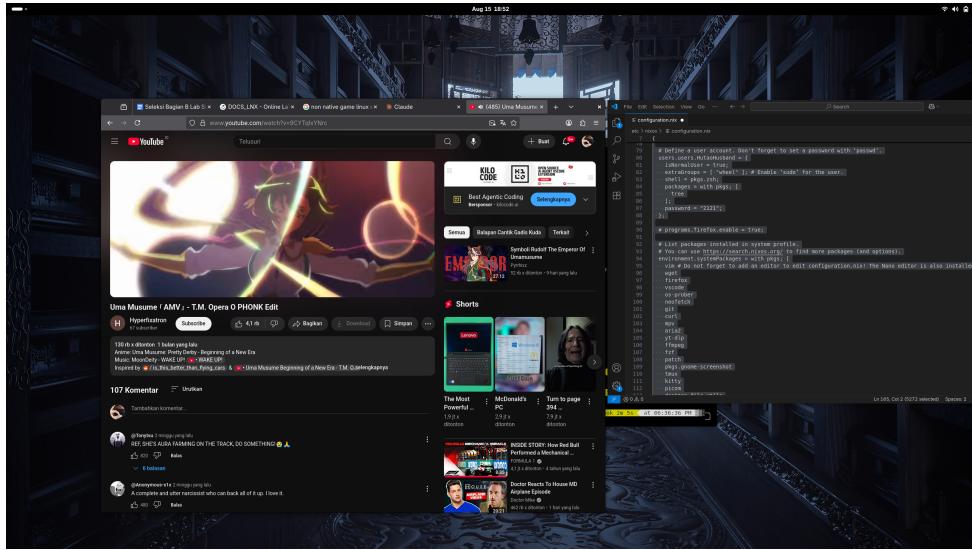


Figure 5: VS Code and Firefox running successfully

5 Bonus Features Implemented

5.1 Wayland Support

NixOS running GNOME on Wayland provides better performance and security compared to X11. This is enabled in our configuration with:

```
1 services.xserver.displayManager.gdm.wayland = true;
```

5.2 Wine and LINE Installation

We've successfully installed Wine and LINE for PC, allowing Windows applications to run on our NixOS system.

5.3 Open-Source Applications

Using Firefox as the Main Internet Browser.

6 Conclusion

We have successfully installed NixOS with all mandatory requirements:

Requirement	Status
Mandatory Requirements	
GUI (GNOME Desktop Environment)	✓
Audio-visual capabilities	✓
Internet connectivity	✓
Package management (Nix)	✓
Unprivileged user account	✓
Graphical text editor (VS Code)	✓
Web browser (Firefox)	✓
Custom wallpaper	✓
Bonus Features	
Post Instagram	✓
Graphical Text Editor & Browser	✓
Wine + Line	✓
Wayland	✓
Anime in CLI	✓
Terminal Doom	✓
Repackage to ISO	✓
Hardware Installation	✓

Table 1: NixOS Installation Requirements Checklist

7 Additional Resources

- Official NixOS Manual: <https://nixos.org/manual/nixos/stable/>
- Nix Package Search: <https://search.nixos.org/packages>
- NixOS Community Wiki: <https://nixos.wiki/>
- NixOS Configuration Examples: <https://github.com/NixOS/nixpkgs>

8 Documentation



Figure 6: Success Boot

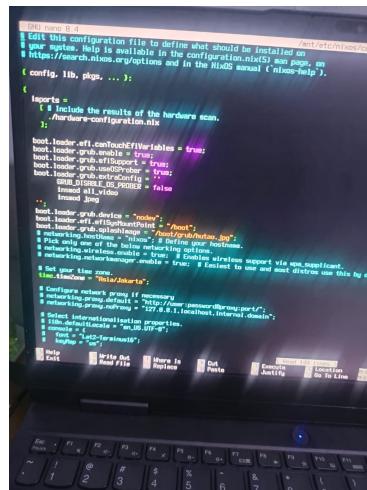


Figure 7: Setting up nix config

"I use Arch, btw" - Anonymous

*"I use NixOS, and I can reproduce my entire system from a single file" - Enlightened
NixOS User*