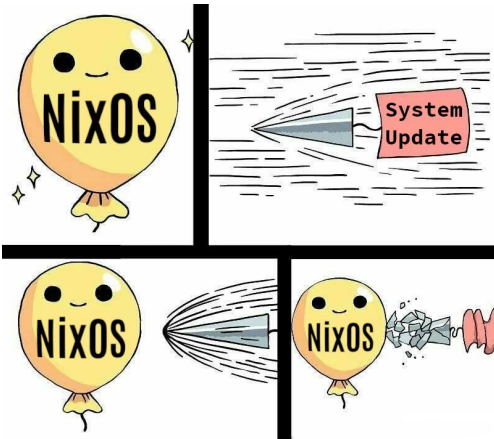


Into the Nixverse



It seems like every time you think you know everything (or just enough) about Linux, there's always something that catches you by surprise...

NixOS and the Nix package manager is software that allows you to configure a Linux system or shell environment in a declarative and therefore reproducible manner. In other words, the state of your system is free from side effects, such as when dependencies of one program are different than another, or when you update a package and the configuration file you've meticulously crafted is out of date. Also, NixOS allows you to rollback to a previous system state, just in case something goes horribly wrong after you update things.

An example system configuration file

```
# Creates a systemd-nspawn container which in turn runs podman containers
{ config, lib, pkgs, ... }:
{
  networking = {
    bridges.br0.interfaces = [ "eth0" ];
    bridges.br1.interfaces = [ "eth0" ];
  };

  containers.baobab = {
    privateNetwork = true;
    hostBridge = "br0"; # Specify the bridge name
    config = { config, pkgs, lib, ... }: {
      [virtualisation.containers.enable = true;
      [virtualization.oci-containers.backend = "podman";
      [virtualisation = {
```

```
□ podman.enable = true;
□ oci-containers.containers = {
□   nextcloud = {
□     image = "docker.io/syncthing/syncthing";
□     autoStart = true;
□     ports = [ "22000:22000" "22000:22000/udp" "8384:8384"];
□     extraOptions = [ "--pull=always" ]
□   };
□ };
□ };
  };
};
}
```

All it takes to rebuild this system is putting the above Nix code into `/etc/nixos/configuration.nix` and executing `nixos-rebuild switch`. Yup, that's it! The Nix packages manager knows to install podman because of lines 13 and 14, and the container's configuration is located in lines 18-22. No need to mess around with package managers or touch any config files.

All of Nixos' available packages can be found in <https://search.nixos.org>, and you can even follow along and see what each option does in the [nixpkgs GitHub repository](#). I have also uploaded my own workstation's super riced-up configuration onto the Grok Zone which you can find [in this chapter](#). All this declarative hubbub also allows you to publish your configurations as something called `flakes`, which allows you to, for example, share your neovim config online and have it run on any other system [with just one command](#).

...We're still not done here.

Frankly, I'm still just a little Nix newbie and haven't even touched things like flakes and nix-writers. But even during my short time learning and using NixOS I've been having a blast. It has honestly been very refreshing to have something else to sink the proverbial teeth into in the realm of Linux. I highly recommend experienced Linux users to give NixOS a whirl; it will take some time to get used to, but your future self that's upgrading or rebuilding your system will thank you. It'll also look nice on your tinder profile.

And remember,



Revision #4

Created 7 September 2024 00:31:26 by GT

Updated 7 September 2024 02:13:38 by GT