

# NixOS configuration

## What is this ?

This git repository contains my NixOS configuration files.

## How to use ?

You can use it as a premade configuration or as an example to do yours.

## Installation :

Install NixOS following the wiki instruction : NixOS manual

Change configuration.nix file to enable git

`git clone` this repository where you want to store the config folder

Replace `hosts/hardware-configuration.nix` by yours

Finally rebuild with

```
sudo nixos-rebuild switch --flake path_to_the_config_folder#system_you_want
```

This configuration provide 3 different hosts (currently only medium):

- light : no graphical interface, just what you need to code
- medium : basic graphical interface, intended for laptop usage
- large : full configuration, intended for pc usage

## User managment

Users are handled in `users`

### Add a user

Create a new folder `users/newuser`.

Copy `users/skwyx/home.nix` and `users/skwyx/user.nix` files under this new folder.

Add this line to `users/user_declaration.nix` :

```
users.newuser.enable = lib.mkEnableOption "Enable newuser";
```

And this one to `users/user_activation.nix` in imports :

```
./newuser/user.nix
```

To enable it you can now add it in the hosts file you use, for exemple `hosts/medium.nix` :

```
users = {  
    newuser.enable = true;  
};
```

To change the user configuration go back to `users/newuser` and modify `home.nix` and `user.nix`

In `user.nix` you will modify permissions and shell that the user will have and his groups.

In `home.nix` you will modify components for the user (i.e. app and tools), username and home directory

### Remove a user

Simply disable it by turning it off in `hosts/system_you_use.nix` in `users = { user.enable = false }`

## Modifying, Adding, Removing contents

You should only need to change `modules` folder to modify or add packages.

If it's a package which is used system wide, or need to be at system level, then modify `modules/system`.

If it's user level, then modify `modules/user`

Keep in mind that if you want to keep the components system you will need to modify `components.nix` files and add your new components to activation files like in `users/youruser/home.nix`

You should not need to remove any configuration piece thanks to the components system, just disable it in your components activation and it's done.

### Dotfiles

To handle dotfiles I use flakes so the dotfiles are mine and if you don't want to use it you will need to modify more things.

There is many means to handle dotfiles with NixOS.

I will cover the one I know but I can miss some.

### Using flakes

That's what i'm using currently and it's the best way to me.

So if you want to change the flake you use for your dotfiles.

First go in `flake.nix` and change the url for the one you want for instance :

```
nvim-dotfiles.url = "github:akSkwYX/neovim-dotfiles";
```

become

```
nvim-dotfiles.url = "github:BestUser/neovim-flake";
```

Then, go where you import this flake (by default for neovim it's in `users/home.nix`) and replace the import line by what's indicated in the new dotfiles repository.

Rebuild and your configuration should have changed (maybe you will need to update flakes by doing `nix flake update --flake path_to_flake`)

### Using home-manager

Remove flakes entirely (see precedent section but suppress instead of modify)  
Then in your user folder add a file for your configuration (or a folder if you want to split your configuration)  
Put your configuration in it and import it from `users/youruser/home.nix`  
Rebuild and you are done

### Using non-declarative dotfiles

This should not be used  
First, suppress flake usage  
Don't forget to install the package somewhere (you probably want it to be in your user configuration)  
Put your dotfiles in `.config/`

### TODO

- Testing
  - Deactivation of a once activated package
  - Installation
- Modules
  - Add more modules with premade configurat