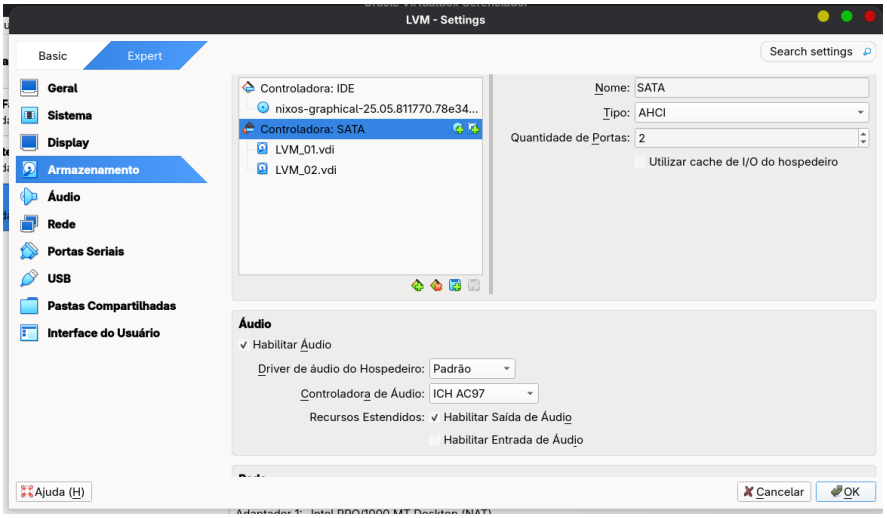
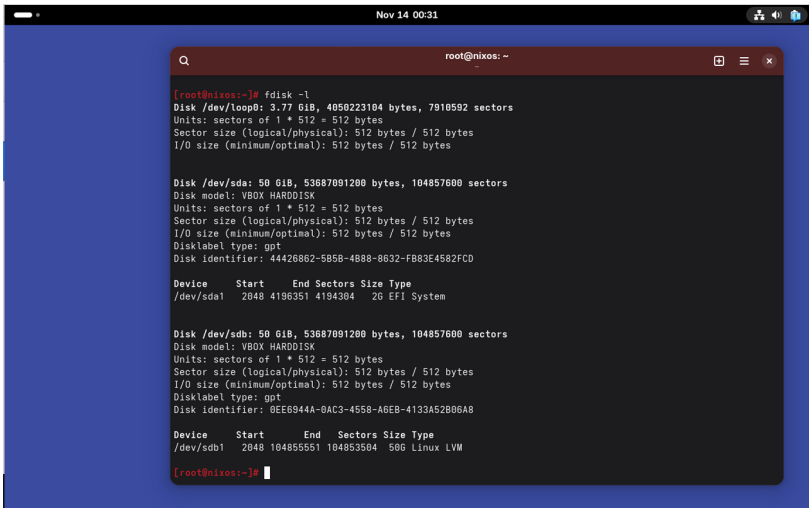
Instalando Nix OS manualmente.

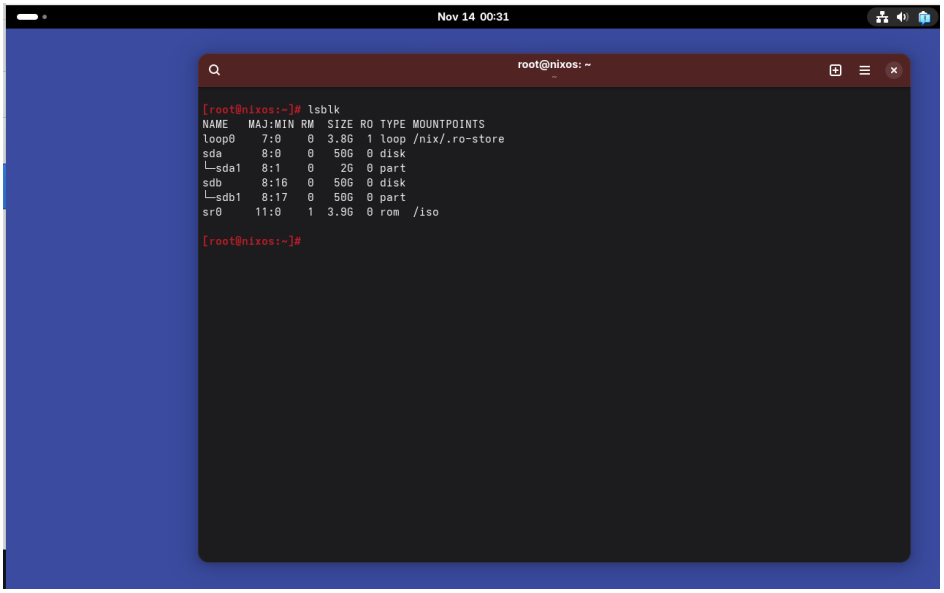
Nix OS é uma distribuição Linux independente que possui pacotes próprios, uma linguagem de programação própria, o Nix, e um sistema de gerenciamento de pacotes declarativo, isto é, você instala os pacotes através de declarações num local específico do arquivo de configuração na linguagem nix, o /etc/nixos/configuration.nix. O Nix não é uma distribuição que se baseia na distribuição de arquivos no modelo padrão do Linux, e possui um diretório próprio, o nix, onde há uma pasta chamada store onde todos os pacotes são alocados. Por ser atômico, muitos podem pensar que não é possível configurar de formar adequada os discos as partições do sistema, mas pode sim e, isso é feito adequadamente na instalação manual. Os procedimentos a seguir foram feitos numa máquina virtual contendo dois discos de 50GB cada. O sistema de formatação de arquivos a ser utilizado será o padrão do Linux EXT4 que poderia ser substituído por um BTRFS ou qualquer outro que desejar. A formatação se dará em cima da camada do LVM, pois a dupla EXT4 e LVM, permite flexibilização de capacidade de armazenamento, aumentando ou diminuindo o tamanho das partições e ou, numa possibilidade diferente, adicionar mais discos sem grandes problemas, além de também poder tirar snapshots e realizar backups. Aliás, grande parte do trabalho de configuração fica em torno destas tarefas em específico. O Nix OS para esta tarefa é muito prático e acredito que isto faz com que seja bom para aqueles que gostam de configurar seu equipamento sem muitos passos.



Conforme imagem acima, a ISO usada foi a de modo gráfica, pois ela ajuda a navegar na rede, se for preciso e manipular certos arquivos mais eficiente.

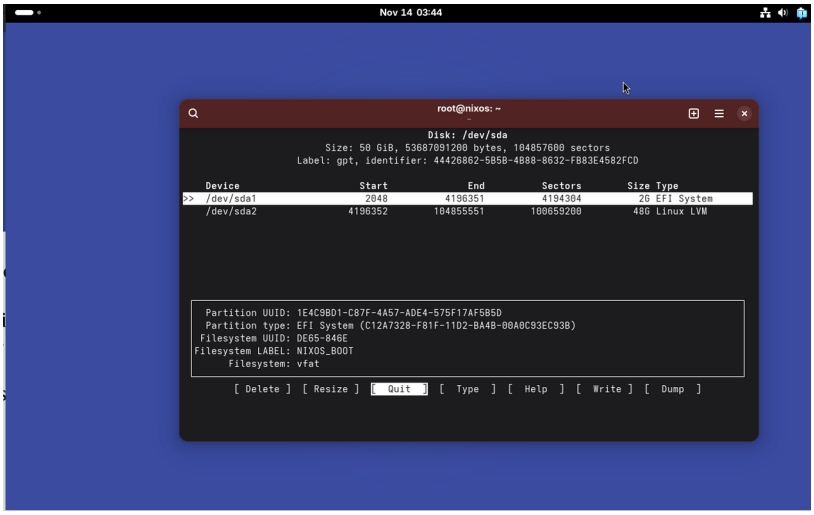


Conforme a segunda imagem, foi aberto o Terminal, (no KDE é o Konsole) e o controle foi passado para o administrador root através do comando sudo -i aplicado diretamente no terminal.



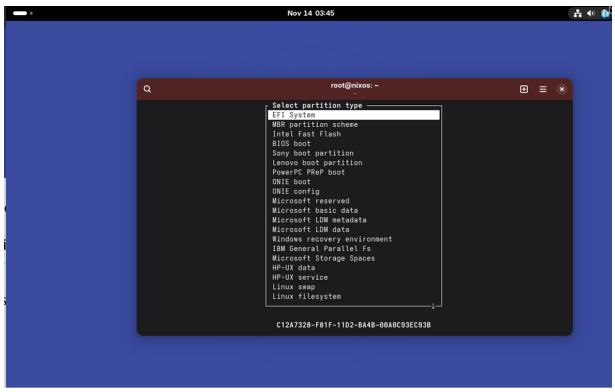
No exemplo, através do comando cfdisk, ou outro como o próprio fdisk, podemos criar as seguintes partições:

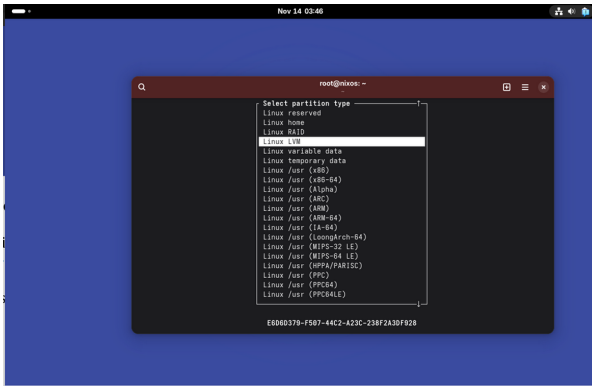
- No primeiro disco sda: a partição de inicialização do sistema que vai ser do tipo EFI de tamanho 2GB; a partição restante (48GB) que será utilizada como parte do grupo de volume geral;
- No segundo disco, apenas o disco inteiro como parte do volume geral.



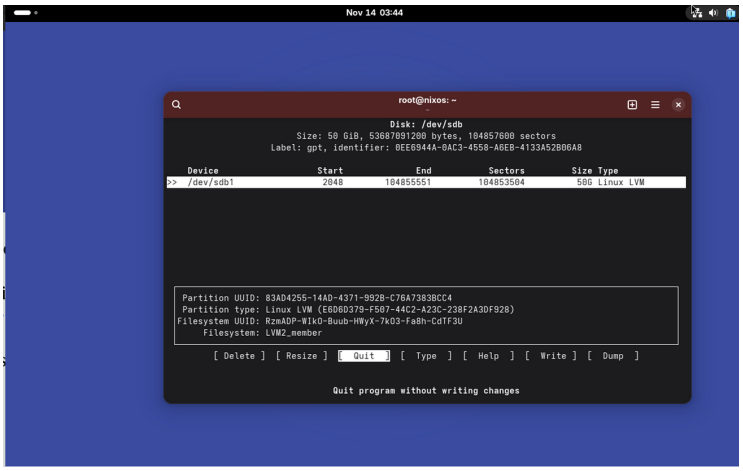
Ao iniciar o cfdisk, com o comando <cfdisk /dev/sda>, o mesmo pergunta se a tabela de partição a ser montada é a GPT, bastando apenas clicar na seleção. Acima o esquema do primeiro disco já montado, com o sistema de inicialização e tamanho de partição já definidos.

As figuras abaixo mostram as seleções para as duas partições.





Para o segundo disco, o esquema de partição ficará assim:



LVM

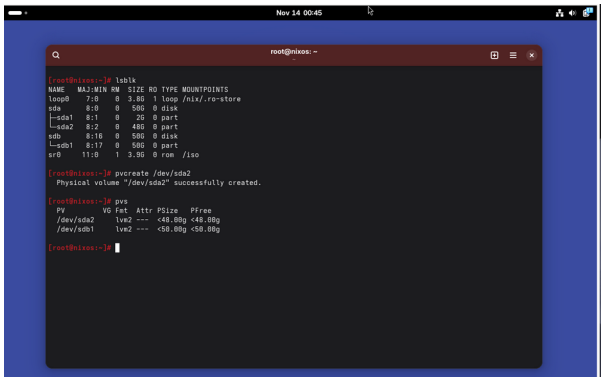
Passando esta etapa, os discos estão prontos para a configuração do LVM.

Para o particionamento com o LVM, há três etapas a serem seguidas:

1. A criação dos volumes físicos (/dev/sda2 e dev/sdb1);
2. A criação do grupo de volumes (nixlvm – único);
3. A criação dos volumes lógicos (root, swap, home).

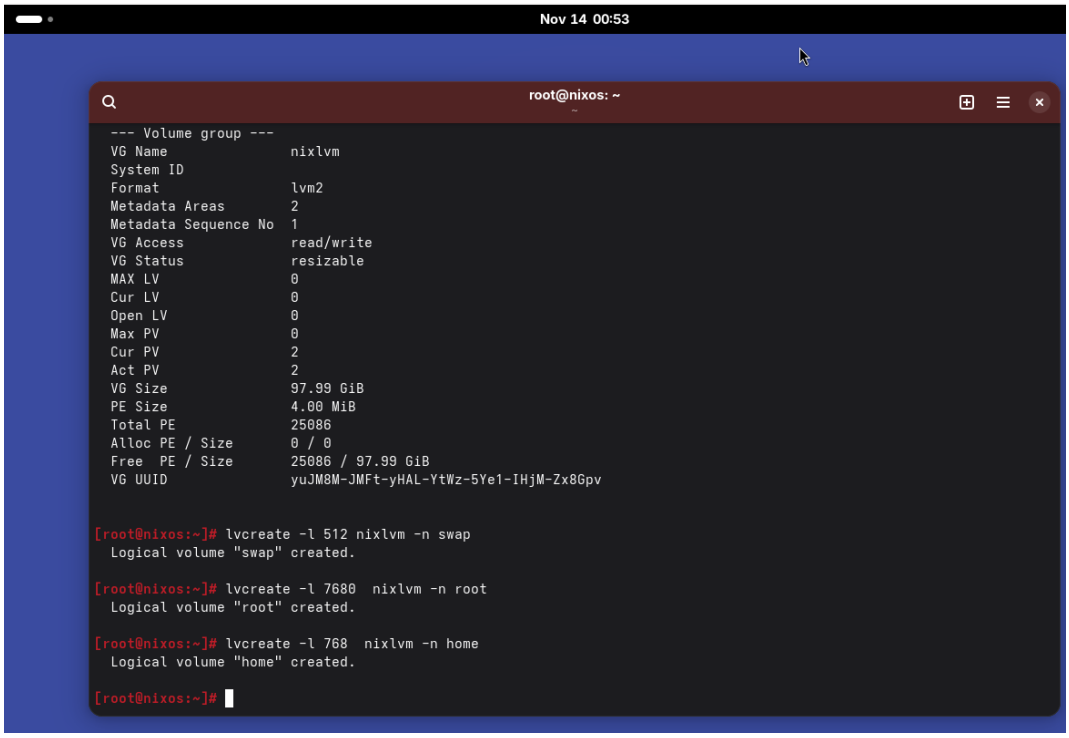
Para a criação dos volumes físicos basta digitar o seguinte comando:

pvcreate /dev/sda2 /dev/sdb1



Para verificar a entrada dos dados basta digitar:
pvs ou pvdisplay

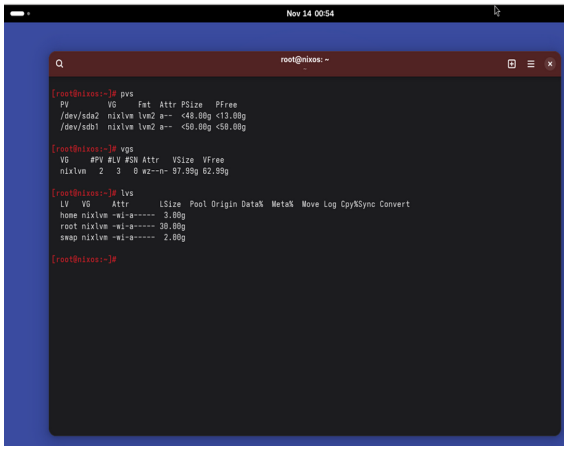
Para a criação do Grupo de Volume, será necessário digitar:
vgcreate nixlvm /dev/sda2 /dev/sdb1
Da mesma forma que o comando pvs e pvdisplay, o comando vgs e vgdisplay se encrregam da tarefa de mostrar os detalhes deste Grupo de Volume o nixlvm. Abaixo, a imagem dos detalhes deste comando e os próximos.



De acordo com os detalhes desta imagem podemos notar o valor total do grupo (97,99 GiB) e também o mesmo cálculo considerando o volume mínimo estendido de 4MiB, mostrando o valor 25086 PE. Este valor ou valores abaixo dele serão importantes para a criação dos volumes lógicos. No exemplo acima, foi utilizado o valor em torno de PEs para criar as partições. Note que o comando <lvcreate -l 512 nixlvm -n swap> é igual ao comando <lvcreate -L 2G nixlvm -n swap>, onde:

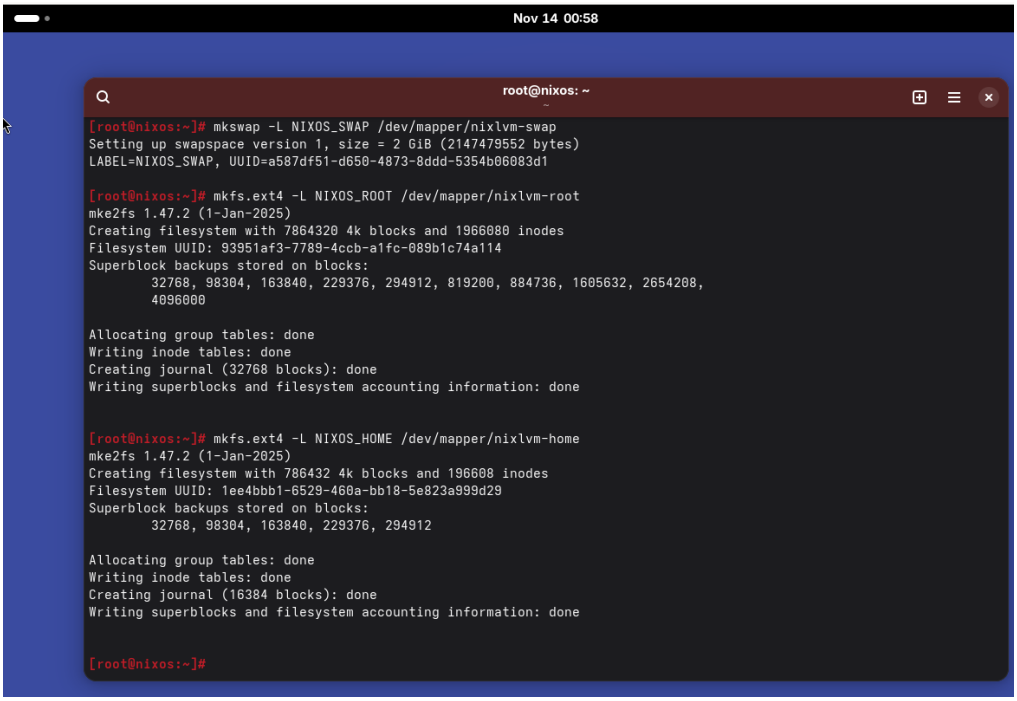
- -l para múltiplos de PEs;
- -L para valores diretos GiB ou similares;
- -n para os nomes dos volumes lógicos.

Assim como os comandos anteriores, o comando lvs e lvdisplay são equivalentes.



Tipo de Sistema de Arquivos

Após a criação dos volumes lógicos, será necessário criar o sistema de arquivos FAT32, EXT4 e SWAP para a montagem do sistema, conforme mostra a imagem abaixo.



Os comandos são:

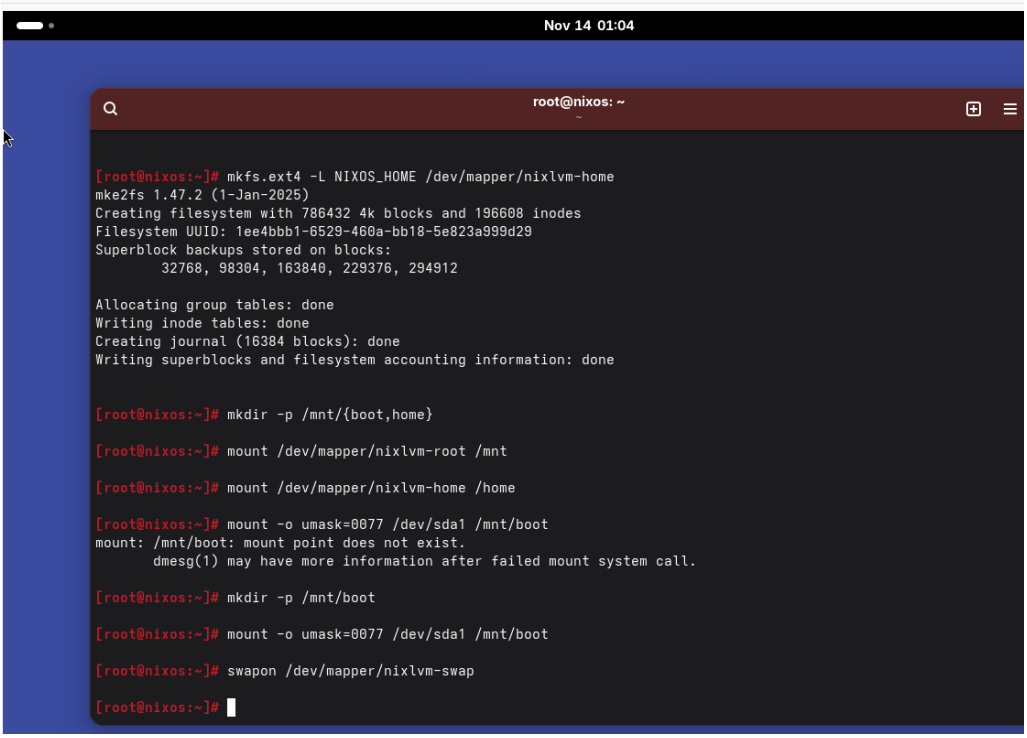
- mkfs.ext4 -L NIXOS_ROOT /dev/mapper/nixlvm-root;
- mkfs.ext4 -L NIXOS_HOME /dev/mapper/nixlvm-home;
- mkswap -L NIXOS_SWAP /dev/mapper/nixlvm-swap;
- mkfs.fat -n NIXOS_BOOT -F32 /dev/sda1.

Montagem dos sistemas de partições

Com os sistemas de arquivos definidos, agora basta finalizar a montagem dos mesmos para seguir com a instalação em si. Os comando utilizados são:

- mkdir -p /mnt/{boot,home};
- mount /dev/mapper/nixlvm-root /mnt
- mount /dev/mapper/nixlvm-home /mnt/home
- mount -o umask=0077 /dev/sda1 /mnt/boot
- swapon /dev/mapper/nixlvm-swap

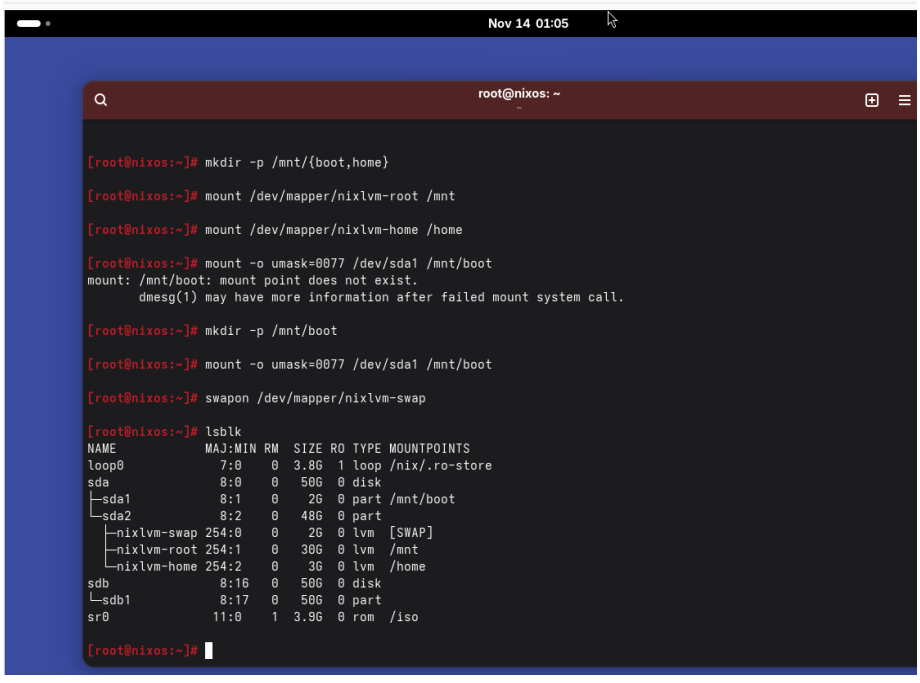
A seguir, imagem com os detalhes da montagem

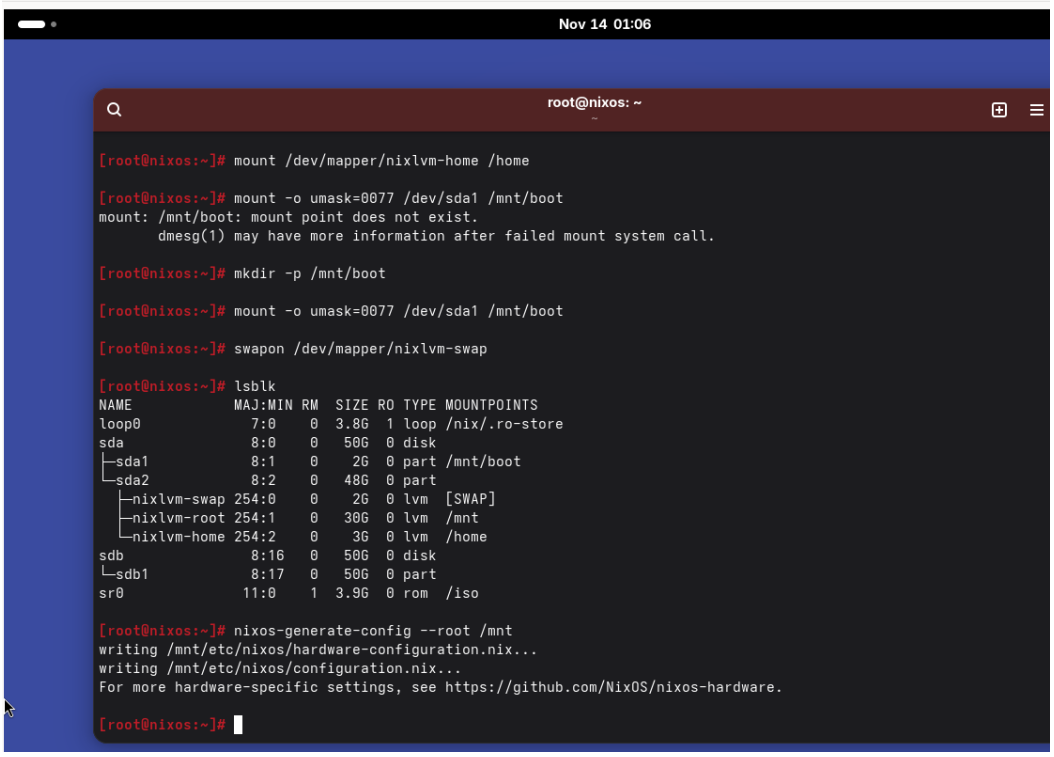


Na imagem abaixo, atento a instrução para a montagem do diretório home. O comando correto é esse:

- mount /dev/mapper/nixlvm-home /mnt/home

Caso haja algum problema, basta corrigir digitação e realizar a geração da configuração do nix novamente <nixos-generate-config - - root /mnt> , como foi o meu caso.





Com o comando <nixos-generate-config - - root /mnt>, há a geração dos arquivos /mnt/etc/nixos/hardware-configuration.nix e /mnt/etc/nixos/configuration.nix, e o sistema nix está com quase tudo pronto para ser instalado, faltando apenas detalhar o que será instalado e a suas configurações pessoais.

Configurando os arquivos do Nix

No arquivo harware-configuration.nix, confira o esquema criado com o do comando lsblk. Foi assim que corrigi o meu problema anterior de montagem do /mnt/home. Desmontei o /home criado (umount /dev/mapper/nixlvm-home) para montá-lo no lugar certo (/mnt/home).
Dois pontos a considerar no arquivo configuration.nix:

- detalhes do tamanho de swap também é colocado nele;
- lembre-se de colocar um HASH da senha para o usuário logar inicialmente.

Para o segundo item acima, abra outro terminal (estando um já ocupado com a configuração) e digite:

- sudo -i (tecle enter)
- # nix-shell --run 'mkpasswd -m SHA-512 -s' -p mkpasswd (tecle enter)
- coloque a seguinte linha no arquivo configuration.nix:

users.users.SEU_NOME.initialHashedPassword = "O_CÓDIGO_GERADO_NO COMANDO_ANTERIOR";

- Crie a senha temporária quando no prompt aparecer password:
- Copie a senha em hash para o comando passado no arquivo configuration.nix.

Para o arquivo de trocas swap, basta acrescentar o seguinte no mesmo arquivo configuration.nix:

```
# Creating Swap Space
  swapDevices = [{
```

```
    device = "/swapfile";
    size = 8 * 1024; # 8GB
  }];
```

Segue abaixo os arquivos configuration.nix e hardware-configuration.nix na íntegra.

hardware-configuration.nix

```
# Do not modify this file!  It was generated by ‘nixos-generate-config’
# and may be overwritten by future invocations.  Please make changes
# to /etc/nixos/configuration.nix instead.
{ config, lib, pkgs, modulesPath, ... }:

{
  imports = [ ];

     boot.initrd.availableKernelModules = [ "ata_piix" "ohci_pci" "ehci_pci" "ahci" "sd_mod"
"sr_mod" ];
  boot.initrd.kernelModules = [ "dm-snapshot" ];
  boot.kernelModules = [ "kvm-intel" ];
  boot.extraModulePackages = [ ];

  fileSystems."/" =
    { device = "/dev/disk/by-uuid/93951af3-7789-4ccb-a1fc-089b1c74a114";
      fsType = "ext4";
    };

  fileSystems."/boot" =
    { device = "/dev/disk/by-uuid/DE65-846E";
      fsType = "vfat";
      options = [ "fmask=0077" "dmask=0077" ];
    };

  fileSystems."/home" =
    { device = "/dev/disk/by-uuid/1ee4bbb1-6529-460a-bb18-5e823a999d29";
      fsType = "ext4";
    };

  swapDevices =
    [ { device = "/dev/disk/by-uuid/a587df51-d650-4873-8ddd-5354b06083d1"; }
    ];

  # Enables DHCP on each ethernet and wireless interface. In case of scripted networking
  # (the default) this is the recommended approach. When using systemd-networkd it's
  # still possible to use this option, but it's recommended to use it in conjunction
  # with explicit per-interface declarations with `networking.interfaces.<interface>.useDHCP`.
  networking.useDHCP = lib.mkDefault true;
  # networking.interfaces.enp0s3.useDHCP = lib.mkDefault true;

  nixpkgs.hostPlatform = lib.mkDefault "x86_64-linux";
  virtualisation.virtualbox.guest.enable = true;
}
```

Configuration.nix

```
# Edit this configuration file to define what should be installed on
# your system. Help is available in the configuration.nix(5) man page, on
# https://search.nixos.org/options and in the NixOS manual (`nixos-help`).

{ config, lib, pkgs, ... }:

{
  imports =
    [ # Include the results of the hardware scan.
      ./hardware-configuration.nix
    ];

  # Use the systemd-boot EFI boot loader.
  boot.loader.systemd-boot.enable = true;
  boot.loader.efi.canTouchEfiVariables = true;

  networking.hostName = "nixos"; # Define your hostname.
  # Pick only one of the below networking options.
  # networking.wireless.enable = true;  # Enables wireless support via wpa_supplicant.
  networking.networkmanager.enable = true;  # Easiest to use and most distros use this by default.

  # Set your time zone.
  time.timeZone = "America/Sao_Paulo";

  # Configure network proxy if necessary
  # networking.proxy.default = "http://user:password@proxy:port/";
  # networking.proxy.noProxy = "127.0.0.1,localhost,internal.domain";

  # Select internationalisation properties.
  # i18n.defaultLocale = "en_US.UTF-8";
  # console = {
  #   font = "Lat2-Terminus16";
  #   keyMap = "us";
  #   useXkbConfig = true; # use xkb.options in tty.
  # };

  i18n.defaultLocale = "pt_BR.UTF-8";

  i18n.extraLocaleSettings = {
    LC_ADDRESS = "pt_BR.UTF-8";
    LC_IDENTIFICATION = "pt_BR.UTF-8";
    LC_MEASUREMENT = "pt_BR.UTF-8";
    LC_MONETARY = "pt_BR.UTF-8";
    LC_NAME = "pt_BR.UTF-8";
    LC_NUMERIC = "pt_BR.UTF-8";
    LC_PAPER = "pt_BR.UTF-8";
    LC_TELEPHONE = "pt_BR.UTF-8";
    LC_TIME = "pt_BR.UTF-8";
```

```
};

# Enable the X11 windowing system.
services.xserver.enable = true;


# Enable the GNOME Desktop Environment.
services.xserver.displayManager.gdm.enable = true;
services.xserver.desktopManager.gnome.enable = true;


# Configure keymap in X11
services.xserver.xkb.layout = "br";
console.keyMap = "br-abnt2";
# services.xserver.xkb.options = "eurosign:e,caps:escape";

# Enable CUPS to print documents.
# services.printing.enable = true;

# Enable sound.
# services.pulseaudio.enable = true;
# OR
 services.pipewire = {
   enable = true;
   alsa.enable = true;
   alsa.support32Bit = true;
   pulse.enable = true;
 };

# Enable touchpad support (enabled default in most desktopManager).
# services.libinput.enable = true;

# Define a user account. Don't forget to set a password with ‘passwd’.
 users.users.SEUNOME_AQUI = {
   isNormalUser = true;
   extraGroups = [ "wheel" "libvirtd"]; # Enable ‘sudo’ for the user.
   packages = with pkgs; [
#     tree
   ];
   initialHashedPassword = "SUA_HASHED_SENHA_AQUI";
 };



 nixpkgs.config.allowUnfree = true;
 # Installing Firefox
 programs.firefox.enable = true;

 # Installing Virt Manager
 virtualisation.libvirtd = {
       enable = true;
```

```
        qemu.vhostUserPackages = with pkgs; [ virtiofsd ];
    };
     programs.virt-manager.enable = true;

    # Install Flatpak
    services.flatpak.enable = true;

    systemd.services.flatpak-repo = {
      wantedBy = [ "multi-user.target" ];
      path = [ pkgs.flatpak ];
      script = "flatpa remote-add --if-not-exits flathub https://dl.flathub.org/repo/flathub.flatpakrepo";

    };
    # List packages installed in system profile.
    # You can use https://search.nixos.org/ to find more packages (and options).
     environment.systemPackages = with pkgs; [
        vim # Do not forget to add an editor to edit configuration.nix! The Nano editor is also installed by default.
        wget
        git
        curl
        nixos-generators
      ];

    swapDevices = [{
      device = "/swapfile";
      size = 2 * 1024; # 2GB
    }];

    # Some programs need SUID wrappers, can be configured further or are
    # started in user sessions.
    # programs.mtr.enable = true;
    # programs.gnupg.agent = {
    #   enable = true;
    #   enableSSHSupport = true;
    # };

    # List services that you want to enable:

    # Enable the OpenSSH daemon.
    # services.openssh.enable = true;

    # Open ports in the firewall.
    # networking.firewall.allowedTCPPorts = [ ... ];
    # networking.firewall.allowedUDPPorts = [ ... ];
    # Or disable the firewall altogether.
    # networking.firewall.enable = false;

    # Copy the NixOS configuration file and link it from the resulting system
    # (/run/current-system/configuration.nix). This is useful in case you
    # accidentally delete configuration.nix.
    # system.copySystemConfiguration = true;
```

This option defines the first version of NixOS you have installed on this particular machine,
and is used to maintain compatibility with application data (e.g. databases) created on older NixOS versions.
#
Most users should NEVER change this value after the initial install, for any reason,
even if you've upgraded your system to a new NixOS release.
#
This value does NOT affect the Nixpkgs version your packages and OS are pulled from,
so changing it will NOT upgrade your system - see https://nixos.org/manual/nixos/stable/#sec-upgrading for how
to actually do that.
#
This value being lower than the current NixOS release does NOT mean your system is
out of date, out of support, or vulnerable.
#
Do NOT change this value unless you have manually inspected all the changes it would make to your configuration,
and migrated your data accordingly.
#
# For more information, see `man configuration.nix` or https://nixos.org/manual/nixos/stable/options#opt-system.stateVersion .
system.stateVersion = "25.05"; # Did you read the comment?

}

Instalação do sistema
Para incluir pacotes e configurações específicas, procure:

- Manual Nix, no cd da instalação;
- na web, Nix Package Search (https://search.nixos.org/packages?channel=25.05);
- na web, NixOS Wiki, (https://wiki.nixos.org/wiki/NixOS_Wiki).

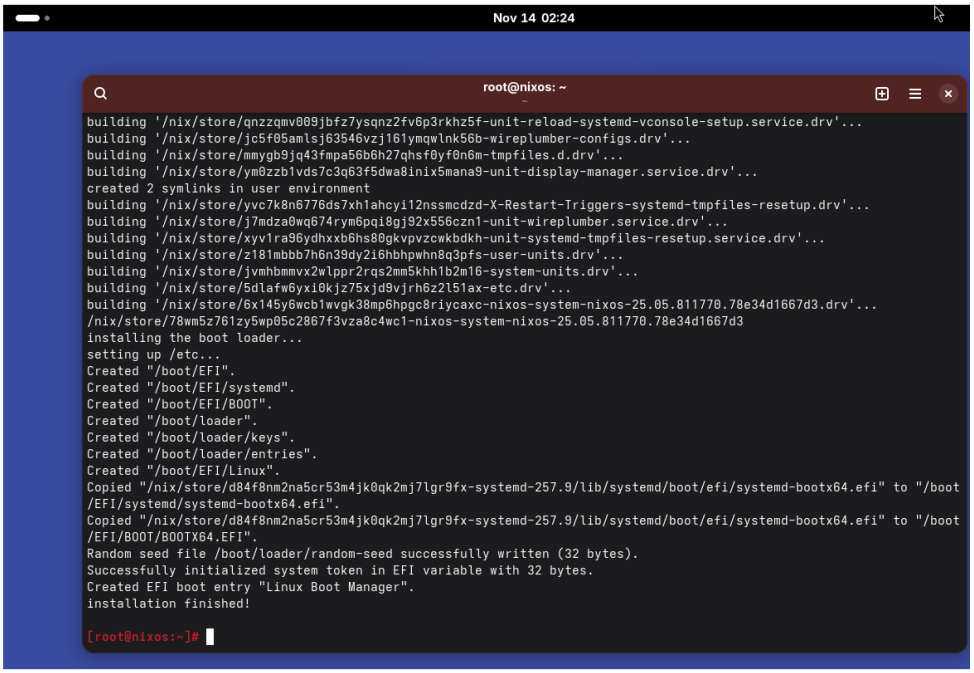
Com os arquivos de configuração já modificados, e com os pacotes necessários, podemos prosseguir com a instalação do sistema. Esse passo é possível com o seguinte comando:

- # nixos-install –no-root-passwd
- # reboot

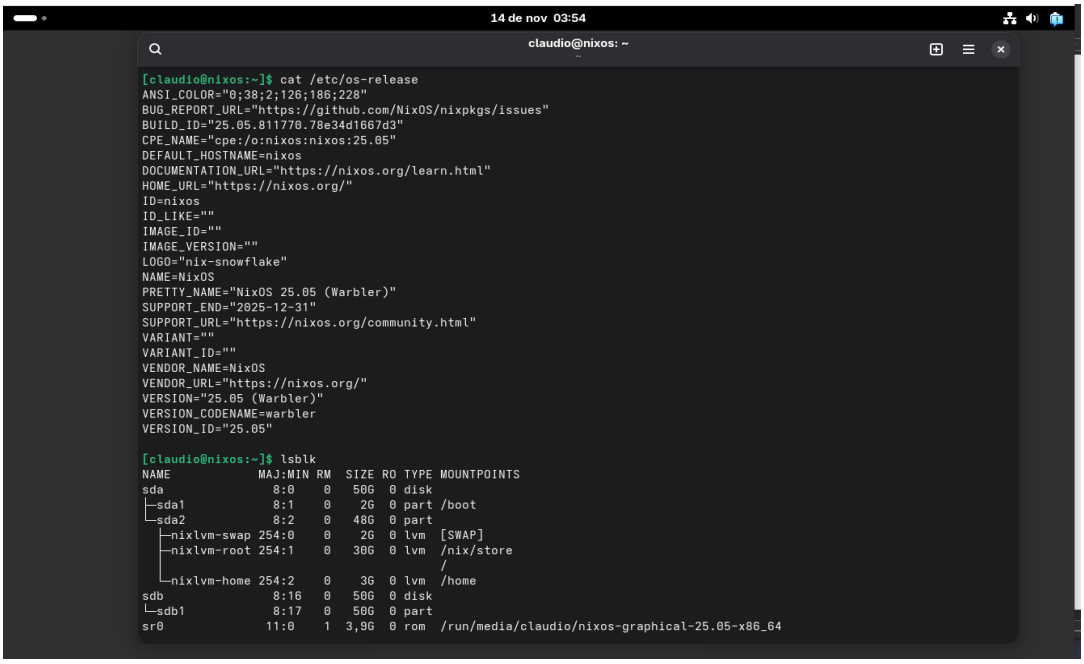
Conforme dito no começo do arquivo, a parte mais trabalhosa da instalação manual do NixOSvem antes, com o processo de montagem e escolha das partições. Este trabalho com o LVM e EXT4 valerá bastante, pois daqui para frente, será necessário só aumentar, reduzir ou anexar volumes lógicos. Abaixo, segue exemplos de como aumentar e diminuir os volumes lógicos.

- lvextend -l +768 /dev/nixlvm-root
- resize2fs /dev/mapper/nixlvm-root

- umount /dev/mapper/nixlvm-root
- e2fsck -f /dev/mapper/nixlvm-root
- resize2fs -p /dev/mapper/nixlvm-root 4G ou lvreduce -l -1024 (-1G) /dev/mapper/nixlvm-root
- lvreduce -l 4136 (4G) /dev/mapper/nixlvm-root (TOTAL de %G para 4G)
- resize2fs /dev/mapper/nixlvm-root
- e2fsck -f /dev/mapper/nixlvm-root
- mount /dev/mapper/nixlvm-root



Com a instalação finalizada, reinicie o sistema (reboot).



Instalação concluída com sucesso.
Acredito que digitar este tutorial demorou bem mais que instalar o sistema em si!
Bons estudos e até mais.
Cláudio