Como instalar o Debian 13 com Btrfs, criptografia completa de disco, snapshots e suporte a rollback para um sistema Linux resiliente.

Este guia passo a passo aborda subvolumes, Snapper, GRUB-Btrfs e recuperação segura em caso de falha nas atualizações.

O Debin GNU/Linux é conhecido por sua estabilidade e confiança. Com o Debian 13 "Trixie", você terá pacotes atualizados, melhor suporte do hardware e 5 anos de atualizações de segurança, fazendo-o ideal para estações de trabalho, servidores e ambientes de desenvolvimento.

Mas e se você pudesse adicionar uma camada extra de proteção nessa distribuição já tão consolidade e confiável? Você consegue isso com o sistema de arquivos BTRFS, que possibilita fazer snapshots instantâneos e restaurar todo o sistema de forma tão simples como o apertar de

um botão. Uma atualização quebrou algo? Restaure em segundos! Quer testar uma alteração ariscada? Faça o snapshot e teste sem medo.

A Instalação gráfica do Debian ainda não dá suporte para a personalização dos subvolumes do Btrfs, então, nesse tutorial vamos usar o método do debootstrap para construir manualmente o sistema com:

- Sistema de arquivos Btrfs otimizado para a estrutura de subvolumes
- Snapper para automatização e gerenciamento dos snapshots
- Integração GRUB-Btrfs com o sistema de boot
- Swap com suporte para hibernação
- Restauração facilitada com o assistente Btrfs

Seja você um usuário em busca de um sistema operacional robusto para uso diário, um administrador de sistemas que deseja opções confiáveis de rollback ou um desenvolvedor que precisa fazer testes com frequência, este guia o ajudará a criar um sistema Debian resiliente.

Prerequisitos:

- **Debian 13 (Trixie) Live ISO**: Faça o download da Live ISO (Não é a instalação ISO padrão) do site official Debian website.
- **Sistema destino**: Uma máquina física ou virtual para instalar o Debian.
- **Conexão com a internet**: An active and stable internet connection.
- **Espaço em disco**: Minimo de 20 GB para o sistema base; É recomendado 50 GB ou mais para o ambiente Desktop e o armazenamento adequado os snapshots.

Sumário:

- Passo 1: Boot com o Live CD (escolhi o XFCE, mas você pode escolher qualquer outro)
- Passo 2: Formatar o disco e criar as partições GPT

Preparativos

```
sudo su


lsblk -p
```

Definições das variáveis

```
export DISK=/dev/sda          # Ajuste para o seu disco (ex: /dev/nvme0n1)
export MAPPER=cryptroot       # Nome do mapeamento LUKS
```

Atualizações dos pacotes e instalação do particionador

```
apt update && apt install gdisk -y
```

Particionamento e Criptografia

```
# Limpeza e Partições
sgdisk -Z $DISK
sgdisk -og $DISK
sgdisk -n 1::+512M -t 1:ef00 -c 1:'ESP' $DISK
sgdisk -n 2::+1G    -t 2:8300 -c 2:'BOOT' $DISK
sgdisk -n 3::       -t 3:8300 -c 3:'LINUX' $DISK

# Formatação inicial
mkfs.fat -F32 -n EFI ${DISK}1
mkfs.ext4 -L BOOT ${DISK}2

# LUKS
cryptsetup luksFormat --type luks2 ${DISK}3
cryptsetup open ${DISK}3 $MAPPER
```

```
# Criar Filesystem no dispositivo mapeado
mkfs.btrfs -L DEBIAN /dev/mapper/$MAPPER
mount /dev/mapper/$MAPPER /mnt
```

Btrfs e Subvolumes

```
# Create essential subvolumes
btrfs subvolume create /mnt/@          # Root filesystem
btrfs subvolume create /mnt/@home      # User home data
btrfs subvolume create /mnt/@opt       # Optional software
btrfs subvolume create /mnt/@cache     # Cache data
btrfs subvolume create /mnt/@lightdm   # Display manager data (XFCE)
btrfs subvolume create /mnt/@libvirt   # Virtual machines
btrfs subvolume create /mnt/@log       # Log files
btrfs subvolume create /mnt/@spool     # Spool data
btrfs subvolume create /mnt/@tmp       # Temporary files
btrfs subvolume create /mnt/@swap      # Swap file location

# Unmount when done
umount -v /mnt
```

> **Note:** Create the subvolume matching your display manager.
> For GNOME use **`@gdm3`**, for KDE use **`@sddm`**, and for XFCE use **`@lightdm`**. Only one is required depending on your desktop environment.

Montagem Final para Instalação

```
# Opções de montagem comuns
export BTRFS_OPTS="compress=zstd:3,noatime,space_cache=v2"

# Montar Raiz
mount -o subvol=@,$BTRFS_OPTS /dev/mapper/$MAPPER /mnt
```

```
# Criar diretórios
mkdir -vp /mnt/{home,opt,boot/efi,var/{cache,lib/{lightdm,libvirt},log,spool,tmp,swap}}

# Montar Partições Físicas
mount -vo ${DISK}1 /mnt/boot/efi
mount -vo ${DISK}2 /mnt/boot

# Montar Subvolumes restantes
mount -vo $BTRFS_OPTS,subvol=@home /dev/mapper/$MAPPER /mnt/home
mount -vo $BTRFS_OPTS,subvol=@opt /dev/mapper/$MAPPER /mnt/opt
mount -vo $BTRFS_OPTS,subvol=@cache /dev/mapper/$MAPPER /mnt/var/cache
mount -vo $BTRFS_OPTS,subvol=@lightdm /dev/mapper/$MAPPER /mnt/var/lib/lightdm
mount -vo $BTRFS_OPTS,subvol=@libvirt /dev/mapper/$MAPPER /mnt/var/lib/libvirt
mount -vo $BTRFS_OPTS,subvol=@log /dev/mapper/$MAPPER /mnt/var/log
mount -vo $BTRFS_OPTS,subvol=@spool /dev/mapper/$MAPPER /mnt/var/spool
mount -vo $BTRFS_OPTS,subvol=@tmp /dev/mapper/$MAPPER /mnt/var/tmp

# Mount swap subvolume without compression or CoW for reliability
mount -vo defaults,noatime,subvol=@swap /dev/mapper/$MAPPER /mnt/var/swap

# Verify the mounts
lsblk -po name,size,fstype,uuid,mountpoints $DISK
```

Install the Debian 13 Base System with debootstrap

```
# Install debootstrap if not already installed
apt install -y debootstrap

# Install base Debian 13 (Trixie) system into /mnt
debootstrap --arch=amd64 trixie /mnt http://deb.debian.org/debian

# Mount necessary filesystems for chroot environment
```

```
for dir in dev proc sys run; do
    mount -v --rbind "/${dir}" "/mnt/${dir}"
    mount -v --make-rslave "/mnt/${dir}"
done

# Mount EFI variables (for UEFI systems)
mount -v -t efivarfs efivarfs /mnt/sys/firmware/efi/efivars
```

Configure fstab + crypttab

```
# 1. Obter UUIDs das partições físicas
BTRFS_UUID=$(blkid -s UUID -o value /dev/mapper/$MAPPER) ; echo "Btrfs UUID: $BTRFS_UUID"
BOOT_UUID=$(blkid -s UUID -o value ${DISK}2) ; echo "Boot UUID: $BOOT_UUID"
EFI_UUID=$(blkid -s UUID -o value ${DISK}1) ; echo "EFI UUID: $EFI_UUID"
LUKS_UUID=$(blkid -s UUID -o value ${DISK}3) ; echo "LUKS UUID: $LUKS_UUID"

# 2. Criar /etc/fstab (Sistema de Arquivos)
cat > /mnt/etc/fstab << EOF
# <file system>  <mount point>     <type>
<options>                                                  <dump> <pass>
UUID=$BTRFS_UUID /                 btrfs
defaults,noatime,space_cache=v2,compress=zstd:3,subvol=@          0  0
UUID=$BTRFS_UUID /home             btrfs
defaults,noatime,space_cache=v2,compress=zstd:3,subvol=@home      0  0
UUID=$BTRFS_UUID /opt              btrfs
defaults,noatime,space_cache=v2,compress=zstd:3,subvol=@opt       0  0
UUID=$BTRFS_UUID /var/cache        btrfs
defaults,noatime,space_cache=v2,compress=zstd:3,subvol=@cache     0  0
UUID=$BTRFS_UUID /var/lib/lightdm  btrfs
defaults,noatime,space_cache=v2,compress=zstd:3,subvol=@lightdm   0  0
UUID=$BTRFS_UUID /var/lib/libvirt  btrfs
defaults,noatime,space_cache=v2,compress=zstd:3,subvol=@libvirt   0  0
UUID=$BTRFS_UUID /var/log          btrfs
defaults,noatime,space_cache=v2,compress=zstd:3,subvol=@log       0  0
UUID=$BTRFS_UUID /var/spool        btrfs
defaults,noatime,space_cache=v2,compress=zstd:3,subvol=@spool     0  0
UUID=$BTRFS_UUID /var/tmp          btrfs
```

```
defaults,noatime,space_cache=v2,compress=zstd:3,subvol=@tmp        0  0
UUID=$BTRFS_UUID /var/swap         btrfs
defaults,noatime,subvol=@swap                                      0  0
UUID=$BOOT_UUID  /boot             ext4
defaults,noatime                                                   0  2
UUID=$EFI_UUID   /boot/efi         vfat
defaults,noatime                                                   0  2
EOF

# 3. Criar /etc/crypttab (Criptografia)
cat > /mnt/etc/crypttab << EOF
# <target name>  <source device>         <key file>   <options>
$MAPPER          UUID=$LUKS_UUID          none        luks,discard
EOF

# 4. Verificar os arquivos criados
echo "=== /etc/fstab ==="
cat /mnt/etc/fstab
echo ""
echo "=== /etc/crypttab ==="
cat /mnt/etc/crypttab
```

Chroot into the Installed System

```
chroot /mnt /bin/bash
```

Configure Base System Settings

```
# Set the system hostname
echo "debian" > /etc/hostname

# Configure /etc/hosts
cat > /etc/hosts << EOF
127.0.0.1       localhost
127.0.1.1       $(cat /etc/hostname)
```

```
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
EOF

# Set the timezone (adjust to your region)
ln -sf /usr/share/zoneinfo/America/New_York /etc/localtime

# Install and configure locales
apt install -y locales
dpkg-reconfigure locales
```

Configure Repositories and Install Base Packages

```
# Configure APT sources for Debian 13 (Trixie)
cat > /etc/apt/sources.list << EOF
deb http://deb.debian.org/debian trixie main contrib non-free non-free-firmware
deb-src http://deb.debian.org/debian trixie main contrib non-free non-free-firmware

deb http://security.debian.org/debian-security trixie-security main contrib non-free non-free-firmware
deb-src http://security.debian.org/debian-security trixie-security main contrib non-free non-free-firmware

deb http://deb.debian.org/debian trixie-updates main contrib non-free non-free-firmware
deb-src http://deb.debian.org/debian trixie-updates main contrib non-free non-free-firmware
EOF

# Update package lists
apt update

# Install kernel, system tools, and essential utilities
apt install -y linux-image-amd64 linux-headers-amd64 \
    firmware-linux firmware-linux-nonfree \
```

```
    grub-efi-amd64 efibootmgr network-manager \
    btrfs-progs sudo vim bash-completion
```

Create Swap with Hibernation Support

```
# Prepare swap file
truncate -s 0 /var/swap/swapfile
chattr +C /var/swap/swapfile                    # Disable COW
btrfs property set /var/swap compression none    # Disable compression

# My system has 4 GB RAM, so I create 6 GB swap for hibernation (1.5× of RAM)
dd if=/dev/zero of=/var/swap/swapfile bs=1M count=6144 status=progress
chmod 600 /var/swap/swapfile
mkswap -L SWAP /var/swap/swapfile

# Add swap to fstab and enable it
echo "/var/swap/swapfile none swap defaults 0 0" >> /etc/fstab
swapon /var/swap/swapfile
swapon -v

# Configure GRUB for hibernation (CORRECTED for LUKS)
SWAP_OFFSET=$(btrfs inspect-internal map-swapfile -r /var/swap/swapfile)
LUKS_UUID=$(blkid -s UUID -o value ${DISK}3)
GRUB_CMD="quiet resume=UUID=$LUKS_UUID resume_offset=$SWAP_OFFSET"
sed -i "s/^GRUB_CMDLINE_LINUX_DEFAULT=.*/GRUB_CMDLINE_LINUX_DEFAULT=\"$GRUB_CMD\"/"
/etc/default/grub

# Update GRUB configuration with new kernel parameters
update-grub

# Configure initramfs for hibernation (CORRECTED for LUKS)
cat > /etc/initramfs-tools/conf.d/resume << EOF
RESUME=UUID=$LUKS_UUID
RESUME_OFFSET=$SWAP_OFFSET
EOF

# Update initramfs to include hibernation support
```

```
update-initramfs -u -k all
```

Como instalar o Debian 13 com Btrfs, criptografia completa, snapshots e rollback

Este guia mostra como instalar o **Debian 13 “Trixie”** usando:

- □□ Criptografia completa de disco (LUKS2)
- □□ Sistema de arquivos **Btrfs**
- □□ Subvolumes bem definidos
- ♻ Snapshots e rollback (base para Snapper + GRUB-Btrfs)
- □□ Swap com suporte a **hibernação**
- □□ Integração correta com **UEFI + GRUB**

O objetivo é criar um sistema **resiliente, auditável e recuperável**, adequado tanto para **estações de trabalho** quanto para **servidores Linux modernos**.

⚠ AVISO IMPORTANTE

> **Este procedimento APAGA COMPLETAMENTE o disco selecionado.**
> Certifique-se de escolher corretamente o dispositivo (`/dev/sda`, `/dev/nvme0n1`, etc).

□□ Pré-requisitos

- Debian 13 **Live ISO** (não é a ISO de instalação padrão)
- Sistema em modo **UEFI**
- Conexão com a internet

- Disco com pelo menos **20 GB** (recomendado: 50 GB ou mais)

---

1 Preparação inicial (Live ISO)

Entre como root:

```
sudo su
```

Instale ferramentas básicas:

```
apt update && apt install -y gdisk debootstrap openssh-server
```

Identifique os discos:

```
lsblk -p
```

Definição de variáveis

```
export DISK=/dev/sda        # ajuste conforme necessário
export MAPPER=cryptroot
```

---

2 Particionamento GPT + UEFI

```
sgdisk -Z $DISK
sgdisk -og $DISK

sgdisk -n 1::+512M -t 1:ef00 -c 1:'ESP'   $DISK
sgdisk -n 2::+1G    -t 2:8300 -c 2:'BOOT' $DISK
```

```
sgdisk -n 3::       -t 3:8300 -c 3:'LINUX' $DISK

# Atualiza as particoes do disco no kernel
partprobe $DISK
```

Formatação

```
mkfs.fat -F32 -n EFI  ${DISK}1
mkfs.ext4    -L BOOT ${DISK}2
```

3□ Criptografia LUKS2

```
# formata a particao encriptada
cryptsetup luksFormat ${DISK}3

# se quiser verificar a particao luks
 cryptsetup luksDump ${DISK}3

# Descriptografa a particao para criar os volumes
cryptsetup open ${DISK}3 $MAPPER
```

Criar filesystem Btrfs:

```
mkfs.btrfs -L DEBIAN /dev/mapper/$MAPPER
mount /dev/mapper/$MAPPER /mnt
```

4□ Criação de subvolumes Btrfs

```
btrfs subvolume create /mnt/@
btrfs subvolume create /mnt/@home
btrfs subvolume create /mnt/@opt
btrfs subvolume create /mnt/@cache
btrfs subvolume create /mnt/@lightdm
btrfs subvolume create /mnt/@libvirt
```

```
btrfs subvolume create /mnt/@log
btrfs subvolume create /mnt/@spool
btrfs subvolume create /mnt/@tmp
btrfs subvolume create /mnt/@swap
```

Desmonte:

```
umount /mnt
```

---

5 Montagem final do sistema

Opções comuns

```
export BTRFS_OPTS="noatime,compress=zstd:1,space_cache=v2"
```

Montar raiz

```
mount -vo subvol=@,$BTRFS_OPTS /dev/mapper/$MAPPER /mnt
```

Criar diretórios:

```
mkdir -vp /mnt/{home,opt,boot,var/{cache,lib/{lightdm,libvirt},log,spool,tmp,swap}}
```

Montar partições físicas (**IMPORTANTE**)

```
mount ${DISK}2 /mnt/boot
mkdir -vp /mnt/boot/efi
mount ${DISK}1 /mnt/boot/efi
```

Montar subvolumes

```
mount -vo $BTRFS_OPTS,subvol=@home     /dev/mapper/$MAPPER /mnt/home
mount -vo $BTRFS_OPTS,subvol=@opt      /dev/mapper/$MAPPER /mnt/opt
mount -vo $BTRFS_OPTS,subvol=@cache    /dev/mapper/$MAPPER /mnt/var/cache
mount -vo $BTRFS_OPTS,subvol=@lightdm  /dev/mapper/$MAPPER /mnt/var/lib/lightdm
mount -vo $BTRFS_OPTS,subvol=@libvirt  /dev/mapper/$MAPPER /mnt/var/lib/libvirt
mount -vo $BTRFS_OPTS,subvol=@log      /dev/mapper/$MAPPER /mnt/var/log
mount -vo $BTRFS_OPTS,subvol=@spool    /dev/mapper/$MAPPER /mnt/var/spool
mount -vo $BTRFS_OPTS,subvol=@tmp      /dev/mapper/$MAPPER /mnt/var/tmp
```

Subvolume swap (sem COW)

```
mount -vo noatime,nodatacow,nodatasum,subvol=@swap /dev/mapper/$MAPPER /mnt/var/swap
```

Validação obrigatória

```
findmnt /mnt
findmnt /mnt/boot/
findmnt /mnt/boot/efi

# Verify the mounts
lsblk -po name,size,fstype,uuid,mountpoints $DISK
```

---

6 Bootstrap do Debian

```
debootstrap --arch=amd64 trixie /mnt http://deb.debian.org/debian
```

---

7 Preparação do chroot

```
for dir in dev proc sys run; do
  mount -v --rbind /$dir /mnt/$dir
  mount -v --make-rslave /mnt/$dir
done
```

```
mount -v -t efivarfs efivarfs /mnt/sys/firmware/efi/efivars
```

9 fstab e crypttab

```
# 1. Obter UUIDs das partições físicas
BTRFS_UUID=$(blkid -s UUID -o value /dev/mapper/$MAPPER) ; echo "Btrfs UUID: $BTRFS_UUID"
BOOT_UUID=$(blkid -s UUID -o value ${DISK}2) ; echo "Boot UUID: $BOOT_UUID"
EFI_UUID=$(blkid -s UUID -o value ${DISK}1) ; echo "EFI UUID: $EFI_UUID"
LUKS_UUID=$(blkid -s UUID -o value ${DISK}3) ; echo "LUKS UUID: $LUKS_UUID"
```

fstab

```
cat > /mnt/etc/fstab << EOF
UUID=$BTRFS_UUID  /                    btrfs
defaults,noatime,space_cache=v2,compress=zstd:1,subvol=@          0  0
UUID=$BTRFS_UUID  /home                btrfs
defaults,noatime,space_cache=v2,compress=zstd:1,subvol=@home      0  0
UUID=$BTRFS_UUID  /opt                 btrfs
defaults,noatime,space_cache=v2,compress=zstd:1,subvol=@opt       0  0
UUID=$BTRFS_UUID  /var/cache           btrfs
```

```
defaults,noatime,space_cache=v2,compress=zstd:1,subvol=@cache     0  0
UUID=$BTRFS_UUID  /var/lib/lightdm     btrfs
defaults,noatime,space_cache=v2,compress=zstd:1,subvol=@lightdm   0  0
UUID=$BTRFS_UUID  /var/lib/libvirt     btrfs
defaults,noatime,space_cache=v2,compress=zstd:1,subvol=@libvirt   0  0
UUID=$BTRFS_UUID  /var/log             btrfs
defaults,noatime,space_cache=v2,compress=zstd:1,subvol=@log       0  0
UUID=$BTRFS_UUID  /var/spool           btrfs
defaults,noatime,space_cache=v2,compress=zstd:1,subvol=@spool     0  0
UUID=$BTRFS_UUID  /var/tmp             btrfs
defaults,noatime,space_cache=v2,compress=zstd:1,subvol=@tmp       0  0
UUID=$BTRFS_UUID  /var/swap            btrfs
defaults,noatime,nodatacow,nodatasum,subvol=@swap                 0  0
UUID=$BOOT_UUID   /boot                ext4
defaults,noatime                                                  0  2
UUID=$EFI_UUID    /boot/efi            vfat
umask=0077,noatime                                                0  1
EOF

##### As informações abaixo são as mesmas instruções acima, apenas foram adicionados
comentários para facilitar o entendimento.

#####  ESSAS INSTRUÇÕES ABAIXO NÃO DEVER SER EXECUTADAS NOVAMENTE
# 2. Criar /etc/fstab (Sistema de Arquivos)
cat > /mnt/etc/fstab << EOF
# ============================================================
# /etc/fstab — Debian 13 (Trixie)
# Btrfs + LUKS + subvolumes
# ============================================================

# <file system>  <mount point>        <type>
<options>                                                  <dump> <pass>

# ---- BTRFS ROOT & SUBVOLUMES ----
UUID=$BTRFS_UUID  /                    btrfs
defaults,noatime,space_cache=v2,compress=zstd:3,subvol=@          0  0
UUID=$BTRFS_UUID  /home                btrfs
defaults,noatime,space_cache=v2,compress=zstd:3,subvol=@home      0  0
UUID=$BTRFS_UUID  /opt                 btrfs
```

```
defaults,noatime,space_cache=v2,compress=zstd:3,subvol=@opt       0  0

# ---- VAR SPLIT ----
UUID=$BTRFS_UUID  /var/cache           btrfs
defaults,noatime,space_cache=v2,compress=zstd:3,subvol=@cache     0  0
UUID=$BTRFS_UUID  /var/lib/lightdm     btrfs
defaults,noatime,space_cache=v2,compress=zstd:3,subvol=@lightdm   0  0
UUID=$BTRFS_UUID  /var/lib/libvirt     btrfs
defaults,noatime,space_cache=v2,compress=zstd:3,subvol=@libvirt   0  0
UUID=$BTRFS_UUID  /var/log             btrfs
defaults,noatime,space_cache=v2,compress=zstd:3,subvol=@log       0  0
UUID=$BTRFS_UUID  /var/spool           btrfs
defaults,noatime,space_cache=v2,compress=zstd:3,subvol=@spool     0  0
UUID=$BTRFS_UUID  /var/tmp             btrfs
defaults,noatime,space_cache=v2,compress=zstd:3,subvol=@tmp       0  0

# ---- SWAP (subvolume dedicado, sem compressão) ----
UUID=$BTRFS_UUID  /var/swap            btrfs
defaults,noatime,nodatacow,nodatasum,subvol=@swap                 0  0

# ---- PARTIÇÕES FÍSICAS ----
UUID=$BOOT_UUID   /boot                ext4
defaults,noatime                                                    0  2
UUID=$EFI_UUID    /boot/efi            vfat
umask=0077,noatime                                                  0  1
EOF

# Verify the fstab file content
cat /mnt/etc/fstab
```

crypttab

```
# 3. Criar /etc/crypttab (Criptografia)
# <target name>  <source device>         <key file>   <options>
# $MAPPER          UUID=$LUKS_UUID          none        luks,discard

cat > /mnt/etc/crypttab << EOF
$MAPPER          UUID=$LUKS_UUID          none        luks,discard
EOF
```

```
# 4. Verificar crypttab
echo "=== /etc/crypttab ==="
cat /mnt/etc/crypttab
```

Entrar no chroot:

```
chroot /mnt /bin/bash
```

---

8 Configuração básica do sistema

Hostname:

```
echo debian > /etc/hostname
```

```
# Configure /etc/hosts
cat > /etc/hosts << EOF
127.0.0.1       localhost
127.0.1.1       $(cat /etc/hostname)

::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
EOF
```

Timezone:

```
ln -sf /usr/share/zoneinfo/America/Porto_Velho /etc/localtime
```

Locales:

```
apt install -y locales
dpkg-reconfigure locales
```

Configure Repositories and Install Base Packages

```
# Configure APT sources for Debian 13 (Trixie)
cat > /etc/apt/sources.list << EOF
deb http://deb.debian.org/debian trixie main contrib non-free non-free-firmware
deb-src http://deb.debian.org/debian trixie main contrib non-free non-free-firmware

deb http://security.debian.org/debian-security trixie-security main contrib non-free non-free-firmware
deb-src http://security.debian.org/debian-security trixie-security main contrib non-free non-free-firmware

deb http://deb.debian.org/debian trixie-updates main contrib non-free non-free-firmware
deb-src http://deb.debian.org/debian trixie-updates main contrib non-free non-free-firmware
EOF
```

```
# Update package lists
apt update
```

□□ Kernel, GRUB e swap com hibernação

Instalar pacotes essenciais:

```
apt install -y linux-image-amd64 linux-headers-amd64 \
    firmware-linux firmware-linux-nonfree \
    grub-efi-amd64 efibootmgr network-manager \
```

```
    btrfs-progs sudo vim bash-completion \
    fzf man-db atop gnupg2 sysstat lsof hwinfo chrony ncdu \
    btop htop iotop wget nmap tcpdump whois bind9-dnsutils dnstop \
    iftop locate traceroute bmon netdiscover fastfetch tmux grc \
    bat entr eza exuberant-ctags silversearcher-ag fd-find ripgrep \
    fonts-cascadia-code fonts-firacode psmisc \
    cryptsetup cryptsetup-initramfs
```

Criar swapfile:

```
# Prepare swap file
truncate -s 0 /var/swap/swapfile
chattr +C /var/swap/swapfile                    # Disable COW
btrfs property set /var/swap compression none    # Disable compression

# My system has 4 GB RAM, so I create 6 GB swap for hibernation (1.5× of RAM)
dd if=/dev/zero of=/var/swap/swapfile bs=1M count=10240 status=progress
chmod 600 /var/swap/swapfile
mkswap -L SWAP /var/swap/swapfile

# Add swap to fstab and enable it
echo "/var/swap/swapfile none swap defaults 0 0" >> /etc/fstab
swapon /var/swap/swapfile
swapon -v

# Configure GRUB for hibernation
SWAP_OFFSET=$(btrfs inspect-internal map-swapfile -r /var/swap/swapfile)
BTRFS_UUID=$(blkid -s UUID -o value /dev/mapper/$MAPPER)
GRUB_CMD="quiet resume=UUID=$BTRFS_UUID resume_offset=$SWAP_OFFSET"
echo "GRUB_CMDLINE_LINUX_DEFAULT=\"$GRUB_CMD\"" >> /etc/default/grub

# Update GRUB configuration with new kernel parameters
update-grub

# Configure initramfs for hibernation (using swap file)
cat > /etc/initramfs-tools/conf.d/resume << EOF
RESUME=UUID=$BTRFS_UUID
RESUME_OFFSET=$SWAP_OFFSET
```

```
EOF

# Update initramfs to include hibernation support
update-initramfs -u -k all
```

1□ 1□ Finalização

Criar usuário:

```
# Create a new user (replace with your username and name)
useradd -m -G sudo,adm,systemd-journal -s /bin/bash -c "Krisofferson Marini" marini

# Set the user password
passwd marini

# Verify the user creation
id marini
```

Install and Configure GRUB Bootloader

```
# Install GRUB for UEFI
grub-install \
  --target=x86_64-efi \
  --efi-directory=/boot/efi \
  --bootloader-id=debian \
  --recheck

# Generate GRUB configuration
update-grub
```

Sair e reiniciar:

```
exit
umount -vR /mnt
```

```
reboot
```

Revision #43
Created 2026-02-05 19:03:24 UTC by Krisofferson Marini
Updated 2026-02-09 16:04:29 UTC by Krisofferson Marini