Tutorials

This section of our documentation contains step-by-step tutorials to help outline what Ubuntu Server is capable of while helping you achieve specific aims.

- Basic installation
- How to operate the Server installer
- Screen-by-screen installer guide
- Configuring storage in the Server installer
- □□ How to Mount NFS Storage on Linux (with Proper Permissions)
- How to delete a folder with contents on ubuntu cli
- □□ Expanding Root Filesystem on Ubuntu with LVM (Virtual Machine)
- □□ NFS Permissions Fix -Boot Script

Basic installation

This chapter provides an overview of how to install Ubuntu Server Edition. You can also refer to this guide on how to operate the installer for more information on using the installer, and to this screen-by-screen reference guide for more information about each of the installer screens.

Preparing to install

This section explains various aspects to consider before starting the installation.

System requirements

Ubuntu Server Edition provides a common, minimalist base for a variety of server applications, such as file/print services, web hosting, email hosting, etc. This version supports four 64-bit architectures:

- amd64 (Intel/AMD 64-bit)
- arm64 (64-bit ARM)
- ppc64el (POWER8 and POWER9)
- s390x (IBM Z and LinuxONE)

The recommended system requirements are:

- CPU: 1 gigahertz or better
- RAM: 1 gigabyte or more
- Disk: a minimum of 2.5 gigabytes

Perform a system back up

Before installing Ubuntu Server Edition you should make sure all data on the system is backed up.

If this is not the first time an operating system has been installed on your computer, it is likely you will need to re-partition your disk to make room for Ubuntu.

Any time you partition your disk, you should be prepared to lose everything on the disk should you make a mistake or something goes wrong during partitioning. The programs used in installation are quite reliable, most have seen years of use, but they also perform destructive actions.

Download the server ISO

You can obtain the amd64 server download from https://releases.ubuntu.com/. Select the version you wish to install and select the “server install image” download. Note that the server download includes the installer.

There are platform specific how-to guides for installations on:

- s390x LPAR
- z/VM
- ppc64el

Create a bootable USB

There are many ways to boot the installer but the simplest and most common way is to create a bootable USB stick to boot the system to be installed with (tutorials for other operating systems are also available).

Boot the installer

Plug the USB stick into the system to be installed and start it.

Most computers will automatically boot from USB or DVD, though in some cases this is disabled to improve boot times. If you don’t see the boot message and the “Welcome” screen which should appear after it, you will need to set your computer to boot from the install media.

There should be an on-screen message when the computer starts telling you what key to press for settings or a boot menu. Depending on the manufacturer, this could be `Escape`, `F2`, `F10` or `F12`. Simply restart your computer and hold down this key until the boot menu appears, then select the drive with the Ubuntu install media.

If you are still having problems, check out the Ubuntu Community documentation on booting from CD/DVD.

After a few moments, the installer will start in its language selection screen.

Welcome screen of the Server installer showing the language selection options

Using the installer

The installer is designed to be easy to use and have sensible defaults so for a first install you can mostly just accept the defaults for the most straightforward install:

- Choose your language
- Update the installer (if offered)
- Select your keyboard layout
- Do not configure networking (the installer attempts to configure wired network interfaces via DHCP, but you can continue without networking if this fails)
- Do not configure a proxy or custom mirror unless you have to in your network
- For storage, leave “use an entire disk” checked, and choose a disk to install to, then select “Done” on the configuration screen and confirm the install
- Enter a username, hostname and password
- On the SSH and snap screens, select “Done”
- You will now see log messages as the install is completed
- Select restart when this is complete, and log in using the username and password provided

How to operate the Server installer

How to operate the Server installer

This document explains how to use the installer in general terms. For a step-by-step guide through the screens of the installer, you can use our screen-by-screen reference guide.

Get the installer

Installer images are made (approximately) daily and are available from https://cdimage.ubuntu.com/ubuntu-server/daily-live/current/. These are not tested as extensively as the images from release day, but they contain the latest packages and installer, so fewer updates will be required during or after installation.

You can download the server installer for amd64 from https://ubuntu.com/download/server and other architectures from http://cdimage.ubuntu.com/releases/20.04/release/.

Installer UI navigation

In general, the installer can be used with the `up` and `down` arrows and `space` or `Enter` keys and a little typing.

`Tab` and `Shift + Tab` move the focus down and up respectively. `Home` / `End` / `Page Up` / `Page Down` can be used to navigate through long lists more quickly in the usual way.

Running the installer over serial

By default, the installer runs on the first virtual terminal, `tty1`. This is what is displayed on any connected monitor by default. However, servers do not always have a monitor. Some out-of-band management systems provide a remote virtual terminal, but some times it is necessary to run the

installer on the serial port. To do this, the kernel command line needs to have an appropriate console specified on it – a common value is `console=ttyS0` but this is not something that can be generically documented.

When running on serial, the installer starts in a basic mode that does using only the ASCII character set and black and white colours. If you are connecting from a terminal emulator such as gnome-terminal that supports Unicode and rich colours you can switch to "rich mode" which uses Unicode, colours and supports many languages.

Connecting to the installer over SSH

If the only available terminal is very basic, an alternative is to connect via SSH. If the network is up by the time the installer starts, instructions are offered on the initial screen in basic mode. Otherwise, instructions are available from the help menu once networking is configured.

In addition, connecting via SSH is assumed to be capable of displaying all Unicode characters, enabling more translations to be used than can be displayed on a virtual terminal.

Help menu

The help menu is always in the top right of the screen. It contains help – both general and for the currently displayed screen – and some general actions.

Switching to a shell prompt

You can switch to a shell at any time by selecting "Enter shell" from the help menu, or pressing `Control + Z` or `F2`.

If you are accessing the installer via `tty1`, you can also access a shell by switching to a different virtual terminal (`Control + Alt` + arrow, or `Control + Alt` + number keys, move between virtual terminals).

Global keys

There are some global keys you can press at any time:

KEY	ACTION

`ESC`	Go back
`F1`	Open help menu
`Control + Z`, `F2`	Switch to shell
`Control + L`, `F3`	Redraw screen
`Control + T`, `F4`	Toggle rich mode (colour, Unicode) on and off

Screen-by-screen installer guide

The installer is designed to be easy to use without the need to refer to documentation. However, this reference guide provides more information for each of the screens of the installer.

Language selection

welcome_c

This screen selects the language for the installer and the default language for the installed system.

More languages can be displayed if you connect via SSH.

Refresh

refresh

This screen is shown if there is an update for the installer available. This allows you to get any improvements and bug fixes made since release.

If you choose to update, the new version will be downloaded and the installer will restart at the same point of the installation.

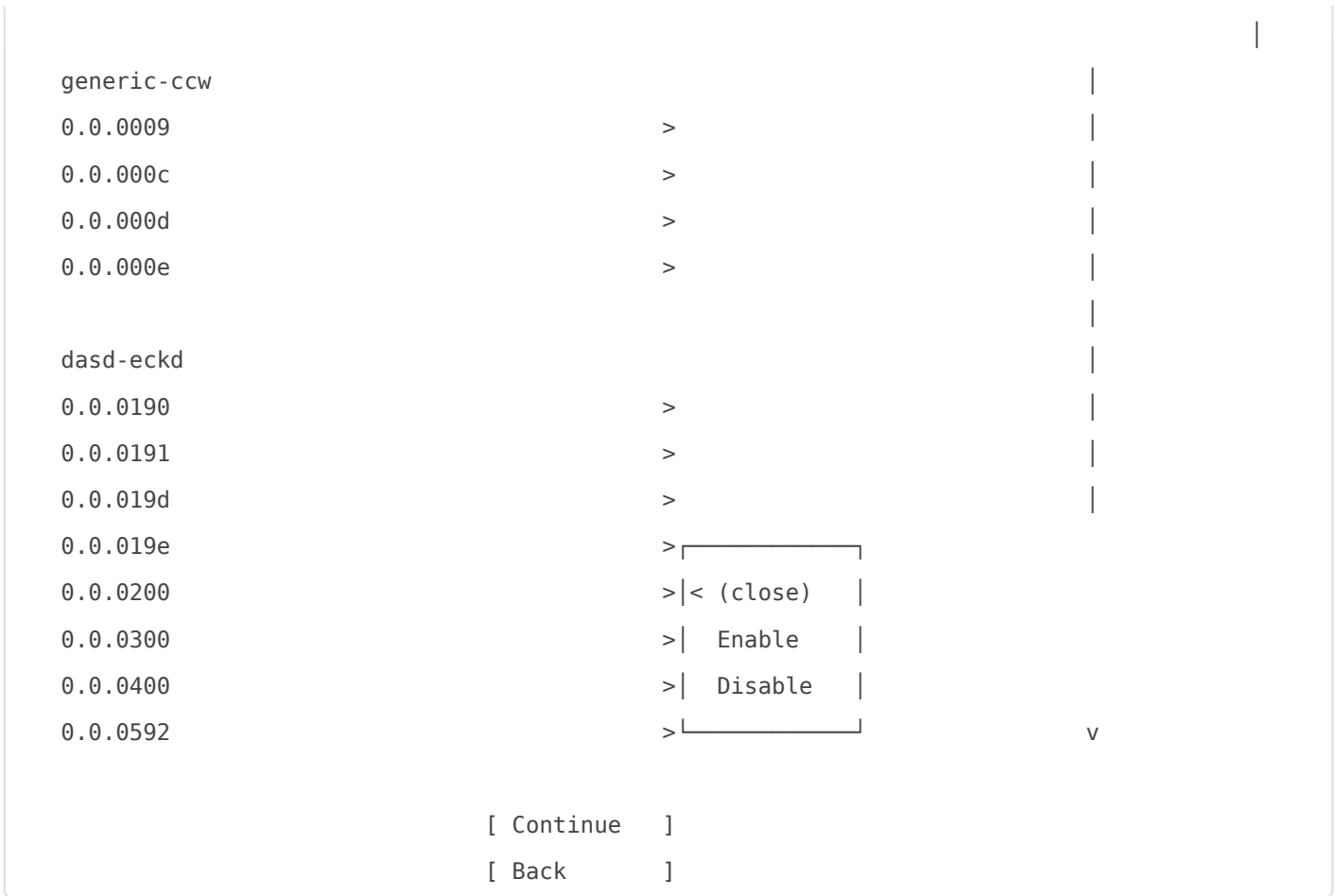
Keyboard

keyboard

Choose the layout and variant of keyboard attached to the system, if any. When running in a virtual terminal, it is possible to guess the layout and variant by answering questions about the keyboard.

Zdev (s390x only)

```
================================================================================
  Zdev setup
================================================================================
  ID                           ONLINE  NAMES                                    ^
```



This screen is only shown on s390x and allows z-specific configuration of devices.

The list of devices can be long. `Home` / `End` / `Page Up` / `Page Down` can be used to navigate through the list more quickly.

Network

network

This screen allows the configuration of the network. Ubuntu Server uses NetPlan to configure networking and the UI of the installer can configure a subset of NetPlan's capabilities. In particular it can configure DHCP or static addressing, VLANs and bonds.

If networking is present (defined as "at least one interface has a default route") then the installer will install updates from the archive at the end of installation.

Proxy

proxy

The proxy configured on this screen is used for accessing the package repository and the snap store both in the installer environment and in the installed system.

Mirror

mirror

The installer will attempt to use `geoip` to look up an appropriate default package mirror for your location. If you want or need to use a different mirror, enter its URL here.

Storage

storage_config

Storage configuration is a complicated topic and has its own page for documentation.

storage_confirm

Once the storage configuration is confirmed, the install begins in the background.

Identity

identity

The default user will be an administrator, able to use `sudo` (this is why a password is needed, even if SSH public key access is enabled on the next screen).

SSH

ssh

A default Ubuntu install has no open ports. It is very common to administer servers via SSH so the installer allows it to be installed with the click of a button.

You can import keys for the default user from GitHub or Launchpad.

If you import a key, then password authentication is disabled by default but it can be re-enabled again if you wish.

Snaps

snaps

If a network connection is enabled, a selection of snaps that are useful in a server environment are presented and can be selected for installation.

Installation logs

install_progress

The final screen of the installer shows the progress of the installer and allows viewing of the full log file. Once the install has completed and security updates installed, the installer waits for confirmation before restarting.

install_done

Configuring storage in the Server installer

Guided options

storage_guided

Selecting “Use an entire disk” on the Guided storage configuration screen will install Ubuntu onto the selected disk, replacing any partitions or data already there.

You can choose whether or not to set up LVM, and if you do, whether or not to encrypt the volume with LUKS. If you encrypt the volume, you need to choose a passphrase that will need to be entered each time the system boots.

If you select “Custom storage layout”, no configuration will be applied to the disks.

In either case, the installer moves onto the main storage customisation screen.

The main storage screen

storage_manual

This screen presents a summary of the current storage configuration. Each device or partition of a device corresponds to a different row (which can be selected), and pressing `Enter` or `space` while a device is selected opens a menu of actions that apply to that device.

Partitions

add_partition_menu

To add a partition to a device, select “Add GPT Partition” for that device.

add_dialog

You can leave “Size” blank to use all the remaining space on the device.

RAID

add_raid

Linux software RAID (RAID stands for “Redundant Array of Inexpensive Disks”) can be used to combine several disks into a single device that is (usually) tolerant to any one disk failure.

A software RAID device can be created out of entire disks or unformatted partitions. Select the “Create software RAID (“MD”)” button to open the creation dialog.

The server installer supports creating devices with RAID level 0, 1, 5, 6 or 10. It does not allow customising other options such as metadata format or RAID10 layout at this time. See the Linux RAID documentation for more details.

A software RAID device can be formatted and mounted directly, can be partitioned into several partitions, or even be used as part of another RAID device or LVM volume group.

Logical Volume Manager (LVM)

add_lvm

The LVM is a system of managing logical volumes, or filesystems, that is much more advanced and flexible than the traditional method of partitioning a disk into one or more segments and formatting that partition with a filesystem. It can be used to combine several disks into one larger pool of storage but it offers advantages even in a single disk system, such as snapshots and easy resizing of logical volumes.

As with RAID, a LVM volume group can be created out of entire disks or unformatted partitions. Select the “Create LVM volume group” button to open the creation dialog.

Once a volume group has been created, it can be divided into named logical volumes which can then be formatted and mounted. It generally makes sense to leave some space in the volume group for storage of snapshots and creation of more logical volumes as needed.

The server installer does not supported configuring any of the many, many options that LVM supports when creating volume groups and logical volumes.

Selecting boot devices

add_boot_device

On all architectures other than s390x, the bootloader needs to be installed to a disk in such a way that the system firmware can find it on boot. By default, the first device to have a partition created on it is selected as a boot device but this can be changed later.

On amd64 and arm64 systems, multiple disks can be selected as boot devices, which means a system can be configured so that it will continue to boot after a failure of any one drive (assuming the root filesystem is placed on a RAID). The bootloader will be installed to each of these drives, and the operating system configured to install new versions of GRUB to each drive as it is updated.

amd64 systems use GRUB as the bootloader. amd64 systems can boot in either UEFI or legacy (sometimes called “BIOS”) mode (many systems can be configured to boot in either mode) and the bootloader is located completely differently in the two modes.

In legacy mode, the bootloader is read from the first “sector” of a hard drive (exactly which hard drive is up to the system firmware, which can usually be configured in a vendor-specific way). The installer will write GRUB to the start of all disks selected as a boot devices. As GRUB does not entirely fit in one sector, a small unformatted partition is needed at the start of the disk, which will automatically be created when a disk is selected as a boot device (a disk with an existing GPT partition table can only be used as a boot device if it has this partition).

In UEFI mode, the bootloader loaded from a “EFI System Partition” (ESP), which is a partition with a particular type GUID. The installer automatically creates a 512MiB ESP on a disk when it is selected as a boot device and will install GRUB there (a disk with an existing partition table can only be used as a boot device if it has an ESP – bootloaders for multiple operating systems can be installed into a single ESP). UEFI defines a standard way to configure the way in which the operating system is chosen on boot, and the installer uses this to configure the system to boot the just-installed operating system. One of the ESPs must be mounted at `/boot/efi`.

Supported arm64 servers boot using UEFI, and are configured the same way as an UEFI-booting amd64 system.

ppc64el systems also load their bootloader (Petitboot, a small linux kernel) from a “PReP” partition with a special flag, so in most ways they are similar to a UEFI system. The installer only supports one PReP partition at this time.

Limitations and workarounds

Currently, the installer cannot *edit* partition tables. You can use existing partitions or reformat a drive entirely but you cannot, for example, remove a large partition and replace it with two smaller ones.

The installer allows the creation of LVM volume groups and logical volumes and MD raid devices, but does not allow tweaking of the parameters – for example, all logical volumes are linear and all MD raid devices use the default metadata format (1.2).

These limits can both be worked around in the same way: drop to a shell and use the usual shell commands to edit the partition table or create the LV or RAID with desired parameters, and then select these partitions or devices as mount points in the installer. Any changes you make while the installer is running but before altering the storage configuration will reflected in the installer.

The installer cannot yet configure iSCSI mounts, ZFS at all, or btrfs subvolumes.

? How to Mount NFS Storage on Linux (with Proper Permissions)

Purpose

Mount a remote NFS share on a Linux server for use by **Docker containers**, ensuring stable operation, correct permissions, and automatic remounting.

---

1. Install NFS Client on the Server

```
sudo apt update
sudo apt install nfs-common -y
```

---

2. Create a Local Mount Directory

Create a local directory where the NFS share will be mounted:

```
sudo mkdir -p /srv/nfs-mount
sudo chown $(whoami):$(whoami) /srv/nfs-mount
```

> *(You can replace `/srv/nfs-mount` with your preferred path.)*

---

3. Mount the NFS Share (Manual Test)

Example:

```
sudo mount -t nfs4 192.168.100.11:/mnt/hdd-storage/my-nfs-share /srv/nfs-mount
```

- `nfs4`: Use NFS version 4 for better performance and locking.
- `proto=tcp`: Reliable transport protocol.
- `hard`: Wait for server recovery instead of failing immediately.
- `timeo=600`: Timeout setting for NFS operations.
- `retrans=2`: Retry failed operations twice.
- `sec=sys`: Default authentication method.
- `_netdev`: Ensure mount occurs only after network is ready.

---

4. Verify That the Mount Worked

```
mount | grep nfs
```

You should see output like:

```
192.168.100.11:/mnt/hdd-storage/my-nfs-share on /srv/nfs-mount type nfs4 (...)
```

---

5. Make the Mount Persistent (Auto-Mount on Boot)

Edit your `/etc/fstab` file:

```
sudo nano /etc/fstab
```

Add the following line at the bottom:

```
192.168.100.11:/mnt/hdd-storage/my-nfs-share /srv/nfs-mount nfs4
rw,relatime,hard,proto=tcp,timeo=600,retrans=2,sec=sys,_netdev 0 0
```

Save and exit (`Ctrl+O`, `Enter`, `Ctrl+X`).

---

6. Test the fstab Entry Without Rebooting

```
sudo mount -a
```

No errors = success!

---

7. Using the NFS Mount with Docker

When running your containers, bind-mount the NFS storage into the container:

```
docker run -d \
  --name my-container \
  -v /srv/nfs-mount:/app/data \
  my-docker-image
```

This will allow your Docker containers to directly access the NFS storage.

? Summary

- **Use NFSv4** (`nfs4`) whenever possible.
- **Always** include `_netdev` in your `/etc/fstab` entries.
- **Use** `hard` **mounts** to protect container file operations during NFS issues.
- **Bind-mount** NFS paths carefully into containers.
- **Avoid** using NFS for database storage unless network latency is extremely low.

?? Useful Commands

- Check NFS mounts: `mount | grep nfs`
- Manual remount all: `sudo mount -a`
- Test connectivity: `ping nfs-server-ip`

? Additional Notes

- If the NFS server reboots, containers **may pause** temporarily.
- If using `docker-compose.yml`, you can map volumes to `/srv/nfs-mount`.
- For production, consider creating a **systemd mount unit** for better recovery behavior.

How to delete a folder with contents on ubuntu cli

To delete a folder and its contents in Ubuntu 23.04 using the command line interface (CLI), you can use the `rm` command with the `-r` (recursive) option. Be very careful when using the `rm` command, as it will permanently delete files and folders.

Here's the basic syntax:

```
rm -r /path/to/your/folder
```

Replace "/path/to/your/folder" with the actual path to the folder you want to delete.

For example, if you want to delete a folder named "myfolder" located in your home directory, you can use the following command:

```
rm -r ~/myfolder
```

Again, be cautious when using the `rm -r` command, as it will not prompt you for confirmation, and the deletion is irreversible. Double-check the path to the folder before running the command to avoid accidental data loss.

If you want to confirm each file deletion, you can use the `-i` option:

```
rm -ri /path/to/your/folder
```

This will prompt you for confirmation before deleting each file and subdirectory.

Always exercise caution when using the `rm` command, especially with the `-r` option, to avoid unintentional data loss.

? Expanding Root Filesystem on Ubuntu with LVM (Virtual Machine)

? Use Case

When a virtual machine runs out of space on the root (`/`) partition, and the underlying disk has already been expanded via the hypervisor or cloud platform.

This guide applies to systems using:

- Ubuntu Server (e.g., 22.04 LTS)
- LVM-managed disks
- A single-root disk layout (e.g., `/dev/mapper/ubuntu--vg-ubuntu--lv`)

? Prerequisites

- A snapshot or backup of the VM (highly recommended)
- Root/sudo access
- Disk already expanded in the hypervisor (e.g., from 72GB to 250GB)

? Step-by-Step Instructions

1. **Check Current Disk Usage**

bash
Copy Edit

```
df -h /
```

2. **List Disks and Partitions**

bash
Copy Edit

```
lsblk
```

Look for:

- The disk (e.g., `xvda`)
- The root LVM volume (e.g., `/dev/mapper/ubuntu--vg-ubuntu--lv`)
- Confirm that a partition (e.g., `xvda3`) is larger than the mounted root volume

3. **Extend the Logical Volume**

bash
Copy Edit

```
sudo lvextend -l +100%FREE /dev/ubuntu-vg/ubuntu-lv
```

4. **Resize the Filesystem**

Assuming you're using `ext4`:

bash
Copy Edit

```
sudo resize2fs /dev/ubuntu-vg/ubuntu-lv
```

> “ □□ To confirm the filesystem type:

bash
Copy Edit

```
df -T /
```

5. **Verify Expansion**

bash
Copy Edit

```
df -h /
```

You should now see the full size available (e.g., ~146GB instead of 72GB).

? Optional: Clean Up Old Snapshots & Logs

Free up even more space:

bash
Copy Edit

```
sudo journalctl --vacuum-time=10d
sudo apt autoremove
sudo apt clean
```

? Outcome

The root filesystem is now successfully extended. The server will run normally with more disk space, avoiding future outages caused by full disks.

? NFS Permissions Fix -Boot Script

Purpose

Ensure that a mounted NFS share has correct ownership and permissions for Docker containers every time the server boots.

---

1. Create a Permission Fix Script

Create a simple script that will reset ownership and permissions on the NFS mount point after boot.

bash

```
sudo nano /usr/local/bin/fix-nfs-permissions.sh
```

Paste inside the script:

```
#!/bin/bash
# Fix NFS ownership
chown -R youruser:yourgroup /srv/nfs-mount
chmod -R 775 /srv/nfs-mount
```

(Replace `youruser` and `yourgroup` with your actual username and group.)

---

2. Make the Script Executable

```
sudo chmod +x /usr/local/bin/fix-nfs-permissions.sh
```

---

3. Create a systemd Service

Create a small service that runs the script automatically at boot:

```
sudo nano /etc/systemd/system/fix-nfs-permissions.service
```

Add this content:

```
[Unit]
Description=Fix NFS Permissions at Boot
After=network.target nfs-client.target remote-fs.target

[Service]
Type=oneshot
ExecStart=/usr/local/bin/fix-nfs-permissions.sh

[Install]
WantedBy=multi-user.target
```

---

4. Enable and Start the Service

```
# Reload systemd to recognize the new service
sudo systemctl daemon-reload

# Enable it to start at every boot
sudo systemctl enable fix-nfs-permissions.service

# Run it now without rebooting (optional)
sudo systemctl start fix-nfs-permissions.service
```

? Summary

- Creates a simple fix script for NFS permissions
- Automates it via systemd on every reboot
- Useful for Docker setups that rely on consistent NFS access

---

?? Useful Commands

```
# Check service status
sudo systemctl status fix-nfs-permissions.service

# Manually trigger the script
sudo /usr/local/bin/fix-nfs-permissions.sh
```

Tip: You can combine this technique with your Docker container volumes to ensure permissions stay stable even after a server or NFS reboot! □□