Install Guide (Ubuntu 24.04)

> ⚠ **Important**
>
> Steps **01** and **04** modify network configuration using Netplan. Run from a **local console or out-of-band access (iDRAC / iLO)** whenever possible. SSH connectivity may briefly drop during these steps.

Prerequisites

Before starting, ensure:

- Ubuntu Server **24.04 LTS** is installed
- You can log in using the **initial administrative user** created during installation
- The logged-in user has `sudo` privileges
- You have console access (strongly recommended)
- You know your lab network settings:
  - Static IPv4 address
  - CIDR prefix (e.g. `/24`)
  - Default gateway
  - DNS server(s)

Execution Order (Do Not Deviate)

Step	Script	Must Run As	Reboot After
00	Copy install files from USB	logged-in user	—
01	`scripts/01-prepare-gns3-host.sh`	root	☑ YES
02	`scripts/02-install-docker.sh`	root	☑ YES
03	`scripts/03-install-gns3-server.sh`	root	☑ YES
04	`scripts/04-bridge-tap-provision.sh`	root	☑ YES
05	`scripts/05-expand-root-lvm-ubuntu.sh`	root	❌ NO
06	Connect from GNS3 GUI	user `gns3`	—
07	Collect logs (optional)	root	—
08	Verify host readiness	root	—

Step 00 — Copy Installation Files from USB

This project is designed to be deployed from a **USB drive**.

Because the server will reboot multiple times during installation, all files must be copied to the **local filesystem** before beginning.

At this stage:

- the `gns3` runtime user **does not yet exist**
- you are logged in as the **initial administrative user**

All work in this step is performed as the **currently logged-in user**.

00.1 Insert the USB drive

Insert the USB drive containing the project.

00.2 Identify the USB block device

List block devices:

```
lsblk
```

Example:

```
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sda      8:0    0   512G  0 disk
├─sda1   8:1    0   512M  0 part /boot/efi
├─sda2   8:2    0     2G  0 part /boot
└─sda3   8:3    0 509.5G  0 part
sdb      8:16   1    16G  0 disk
└─sdb1   8:17   1    16G  0 part
```

In this example:

- `sda` → internal system disk
- `sdb` → USB drive

⚠ **Do not guess.** Confirm using disk size and the removable (`RM`) flag.

00.3 Create a mount point

```
sudo mkdir -p /mnt/usb
```

00.4 Mount the USB drive

Replace `/dev/sdb1` with your actual USB partition.

```
sudo mount /dev/sdb1 /mnt/usb
```

Verify:

```
ls /mnt/usb
```

You should see:

```
gns3-bare-metal-kit/
```

00.5 Copy project files to the local system

Copy the entire project into the **home directory of the currently logged-in user**.

```
cp -a /mnt/usb/gns3-bare-metal-kit ~/
```

This results in:

```
/home/<logged-in-user>/gns3-bare-metal-kit
```

Examples:

```
/home/student/gns3-bare-metal-kit
/home/admin/gns3-bare-metal-kit
/home/itadmin/gns3-bare-metal-kit
```

00.6 Verify local copy

```
cd ~/gns3-bare-metal-kit
ls
```

Expected contents:

```
scripts/
docs/
README.md
LICENSE
CHANGELOG.md
```

From this point forward, **all installation commands must be run from this local directory**, not from the USB drive.

00.7 Unmount the USB drive

```
sudo umount /mnt/usb
```

The USB drive may now be safely removed.

Why Step 00 is required

This approach ensures:

- scripts remain available after every reboot
- installation does not depend on removable media
- consistent working directory across all steps
- reduced risk of accidental execution from `/mnt`
- alignment with real-world server deployment practices

Step 01 — Prepare Host

```
cd ~/gns3-bare-metal-kit
sudo bash scripts/01-prepare-gns3-host.sh
sudo reboot
```

What this step does

- Configures static IPv4 using Netplan
- Sets timezone and enables NTP
- Installs administrative utilities
- Installs and enables OpenSSH Server
- Creates runtime user `gns3`
- Installs KVM / libvirt virtualization baseline
- Loads kernel modules and sysctl tuning

Step 02 — Install Docker CE

```
sudo bash scripts/02-install-docker.sh
sudo reboot
```

What this step does

- Installs Docker CE from the official repository
- Enables Docker service
- Adds `gns3` user to `docker` group

Step 03 — Install GNS3 Server

```
sudo bash scripts/03-install-gns3-server.sh
sudo reboot
```

What this step does

- Installs GNS3 Server from Ubuntu PPA
- Installs ubridge, QEMU, libvirt, console tools
- Writes authoritative GNS3 configuration
- Installs and enables `gns3server.service`
- Verifies KVM and ubridge permissions

Step 04 — Bridge and TAP Provisioning

```
sudo bash scripts/04-bridge-tap-provision.sh
sudo reboot
```

What this step does

- Creates Linux bridge `br0`
- Moves IP ownership from NIC to bridge
- Creates persistent TAP interfaces `tap0` and `tap1`
- Installs and enables `gns3-taps.service`

Step 05 — Expand Root Filesystem (Optional)

```
sudo bash scripts/05-expand-root-lvm-ubuntu.sh
```

What this step does

- Extends Ubuntu's default root logical volume:

  ```
  /dev/mapper/ubuntu--vg-ubuntu--lv
  ```

- Consumes all remaining free disk space
- Grows the filesystem using `resize2fs`

No reboot is required.

Step 06 — Connect from GNS3 GUI

- Add the remote GNS3 server (host IP)
- Add a Cloud node
- Bind to `tap0` / `tap1`

Step 07 — Collect Logs (Optional)

```
sudo bash scripts/06-collect-logs.sh
```

Logs are stored in:

```
/var/log/gns3-bare-metal/
```

Step 08 — Verify Host Readiness

```
sudo bash scripts/07-verify-host.sh
```

Checks:

- KVM acceleration
- Docker service
- gns3server service
- ubridge permissions
- bridge + TAP integrity

Exit code:

- `0` → READY
- `1` → NOT READY

This script is **non-mutating** and safe to run at any time.

Logging

All scripts automatically log to:

```
/var/log/gns3-bare-metal/
```

Logs are written to both console and file.

Dry-Run Mode (Advanced)

Preview script behavior without making changes:

```
sudo bash scripts/02-install-docker.sh --dry-run
```

References

- Docker Engine (Ubuntu): https://docs.docker.com/engine/install/ubuntu/
- Docker Engine (Debian): https://docs.docker.com/engine/install/debian/
- GNS3 Linux Installation: https://docs.gns3.com/docs/getting-started/installation/linux/
