

# 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 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
01	<a href="#">scripts/01-prepare-gns3-host.sh</a>	root	<input checked="" type="checkbox"/> YES
02	<a href="#">scripts/02-install-docker.sh</a>	root	<input checked="" type="checkbox"/> YES
03	<a href="#">scripts/03-install-gns3-server.sh</a>	root	<input checked="" type="checkbox"/> YES
04	<a href="#">scripts/04-bridge-tap-provision.sh</a>	root	<input checked="" type="checkbox"/> YES
05	<a href="#">scripts/05-expand-root-lvm-ubuntu.sh</a>	root	<input checked="" type="checkbox"/> NO
06	Connect from GNS3 GUI	user <a href="#">gns3</a>	—
07	Collect logs (optional)	root	—
08	Verify host readiness	root	—

## Step 01 — Prepare Host

```
sudo bash scripts/01-prepare-gns3-host.sh
sudo reboot
```

## What this step does

- Prompts for the primary NIC and configures **static IPv4**
  - Writes Netplan configuration:
    - `/etc/netplan/01-static-ip.yaml`
  - Sets system timezone and enables NTP
  - Installs baseline administrative and network utilities
  - Installs and enables OpenSSH Server
  - Creates a dedicated runtime user: `gns3`
  - Optionally enables passwordless sudo (lab mode)
  - Installs KVM / libvirt virtualization baseline
  - Loads required kernel modules:
    - `tun`
    - `br_netfilter`
  - Applies sysctl tuning for routing and bridging
- 

## Step 02 — Install Docker CE

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

## What this step does

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

A reboot (or full logout/login) is required for group membership to apply.

---

## Step 03 — Install GNS3 Server

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

## What this step does

- Installs GNS3 Server from the official Ubuntu PPA
- Installs required components:
  - `ubridge`
  - QEMU / KVM
  - libvirt
  - console tools (VNC / SPICE)
- Writes authoritative configuration:
  - `~/.config/GNS3/2.2/gns3_server.conf`
- Installs and enables:
  - `gns3server.service`
- Performs hard verification:
  - KVM acceleration available
  - `ubridge` executable by user `gns3`

---

## Step 04 — Bridge and TAP Provisioning

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

### What this step does

- Converts the physical NIC into a bridge port
- Creates Linux bridge:
  - `br0` (owns the IP address)
- Creates persistent TAP interfaces:
  - `tap0`
  - `tap1`
- Assigns TAP ownership to user `gns3`
- Installs and enables:
  - `gns3-taps.service`
- Ensures TAP interfaces persist across reboot

After this step, **the bridge (`br0`) owns the IP address**, not the physical NIC.

---

## Step 05 — Expand Root Filesystem (Ubuntu Default LVM)

### **⚠ Optional but strongly recommended**

Ubuntu Server commonly installs with a **small root logical volume** (often ~100 GB), even when the disk is much larger.

This step safely expands the root filesystem to use **all remaining free disk space**.

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

### What this step does

- Detects Ubuntu's default logical volume:

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

- If the LV exists:
  - Extends it to consume **100% of free space**
  - Grows the filesystem using **resize2fs**
- If the LV does not exist:
  - Script exits safely
  - No changes are made

### Important notes

- Non-destructive
- No reboot required
- Applies only to Ubuntu's default LVM layout

You can confirm results with:

```
df -h /
```

## Step 06 — Connect from GNS3 GUI

From your workstation:

### 1. Add a **Remote GNS3 Server**

- Host: server IP address
- Port: default (**3080**)

## 2. Add a **Cloud node**

### 3. Bind interfaces to:

- `tap0`
- `tap1`

These TAP interfaces provide direct Layer-2 access to the physical lab network.

---

## Step 07 — Collect Logs (Optional)

If troubleshooting or submitting logs:

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

This generates a compressed archive containing all installation and verification logs.

Default log location:

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

---

## Step 08 — Verify Host Readiness

After completing Step 04 (and rebooting), run:

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

What this verifies

- KVM acceleration (`/dev/kvm`)
- Docker installed and service active
- `gns3server` systemd service running
- `ubridge` executable by user `gns3`
- Linux bridge (`br0`) exists and has IPv4
- TAP interfaces (`tap0`, `tap1`) exist and are UP
- TAPs attached to bridge
- `gns3-taps.service` active

Exit codes

Code	Meaning
0	<input checked="" type="checkbox"/> Host READY

---

Code	Meaning
------	---------

---

1	✗ Host 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/
```

Each execution creates a timestamped log file:

```
03-install-gns3-server-2026-01-23_21-14-10.log
```

Logs are written to both:

- console output
  - persistent log files
- 

## Dry-run Mode (Advanced)

Preview what a script would do **without making changes**:

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

Dry-run mode:

- prints intended commands
  - skips package installs
  - skips file writes
  - skips service changes
  - skips network modifications
- 

## 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/>
-