Jelly Concepcion

Aspiring Cyber Defense Incident Responder | Future Chief Information Security Officer (CISO) | Passionate About Cybersecurity & Threat Detection

Email: jellydizonconcepcion@gmail.com | GitHub: https://github.com/jellyconcepcion/blue-team-homelab | LinkedIn: https://www.linkedin.com/in/jellyconcepcion/

Introduction:

> This document is a step-by-step guide to quickly recreate the Ubuntu Server VM used in my Blue Team lab environment (Month 1). It covers the full installation of Ubuntu 24.04, network configuration, Wazuh SIEM deployment, and verification of all services. Screenshots and evidence files are included to ensure reproducibility.
>
> **Goal:** Build a fully functional Ubuntu-based SIEM host with static IPs and Wazuh installed, ready for SOC lab testing.
>
> **Audience:** Cybersecurity learners or SOC engineers who want to quickly replicate the Ubuntu VM setup without trial-and-error.

Preparation Stage.

Prep: create a Host-Only network in VirtualBox

🔹 Steps to Open Host Network Manager in VirtualBox 7.1.12

1. Open **Oracle VM VirtualBox**.

2. VirtualBox menu → Click **File** → **Tools** → **Network Manager** → this opens the **Host Network Manager**.

---

1) Use the existing adapter (reuse it)

If you already have one Host-Only network named "**VirtualBox Host-Only Ethernet Adapter**" with IPv4 `192.168.56.1`. No need to create another adapter. Creating a second one only adds noise (#2) and will complicate screenshots and documentation. Reuse the existing adapter ("VirtualBox Host-Only Ethernet Adapter").

2) Turn OFF the DHCP server (so we control IPs)

1. VirtualBox → **File** → **Tools** → **Network Manager** → **Host-only Networks** tab.
2. Select the existing adapter named "**VirtualBox Host-Only Ethernet Adapter**" (the one showing `192.168.56.1`).
3. Click the **DHCP Server** tab.
4. Uncheck **Enable Server**. Click **Apply** / **OK**.

Why: disabling DHCP forces you to set static IPs inside each VM and guarantees the addresses won't change. It also looks more professional in documentation.

Screenshots to take now:

Take two separate full-window screenshots (one for each pane).

- Filename: `blue-team-homelab/01-lab-setup/01-ubuntu-siem/screenshots/vb-hostonly-adapter.png` — show the **Adapter** tab with IPv4 address visible.
- Filename: `blue-team-homelab/01-lab-setup/01-ubuntu-siem/screenshots/vb-hostonly-dhcp-disabled.png` — show the **DHCP** tab with "Enable Server" **unchecked**.
- Why separate? It's clearer in documentation and prints well in README or PDF reports.

Create an Ubuntu VM using ubuntu-24.04.3-live-server-amd64.iso.

1) Create Ubuntu Server VM — LAB-SIEM-WAZUH-01 (first)

Goal: Install Ubuntu Server 24.04, set hostname `wazuh-lab-01`, assign host-only IP `192.168.56.10`, and prepare for Wazuh install.

VirtualBox GUI settings

- **New** → Name: `LAB-SIEM-WAZUH-01` → Type: Linux → Version: `Ubuntu (64-bit)`. Screenshot: blue-team-homelab/01-lab-setup/01-ubuntu-siem/screenshots/ubuntu-create-vm-name-os.png
- Username: labadmin. Password: Dream!110100 Hostname: LAB-SIEM-WAZUH-01. Domain Name: myguest.virtualbox.org. Screenshot: ubuntu-create-vm-unattended-install.png

Note: The system should meet the recommended minimum hardware requirements of 4Gb of RAM and 2 CPU cores for installing wazuh.

- **Memory: 4096** MB.
- **Processors:** 2 CPU cores. Screenshot: ubuntu-create-vm-hardware.png
- **Hard disk:** Create VDI, dynamically allocated, **40 GB** (Wazuh/Indexing can take space). Screenshot: ubuntu-create-vm-hard-disk.png
- **System → Motherboard**
  - Uncheck EFI (no EFI necessary for Ubuntu server).
  - **Boot Order: For installation, it's cleaner and more professional to set Optical first, then Hard Disk:**

- **Optical (checked, top priority)**
- **Hard Disk (checked, second)**
- **Floppy (unchecked)**
- **Network (unchecked)**

When the VM boots the first time, it will read the ISO (Ubuntu installer). After installation finishes, Ubuntu will boot from the virtual hard disk automatically (since it's second in the list). Screenshot: ubuntu-vm-settings-system.png

- **Network** → **Uncheck Adapter 1** NAT, Check **Adapter 2** Host-Only Adapter. (To solve the problem on the Installation System where you got stuck on "Installing kernel" for more than 1 hour, **force install without mirror by disconnecting network**

(disable NAT adapter) so installer is forced to use the ISO pool instead of pulling from the net. This sometimes lets "Installing kernel" finish, since it doesn't wait for a failed download.) Screenshots: ubuntu-vm-settings-adapter1.png and ubuntu-vm-settings-adapter2.png

- To enable VT-x/AMD-V:

A. Check virtualization in Task Manager (fast)

1. Press `Ctrl+Shift+Esc` → Performance → CPU.
2. Look for: **Virtualization: Enabled. Screenshot:** vb-create-ubuntu-host-virtualization-enabled.png

B. Check with `systeminfo` (detailed)

1. **Open Command Prompt (cmd) as Administrator.**
2. **Run: systeminfo**
3. **Look for lines near the bottom: Virtualization Enabled In Firmware: Yes**

If Task Manager shows Enabled and `systeminfo` says virtualization enabled in firmware, then VT-x is present. If VirtualBox still doesn't expose an explicit checkbox, you're fine — VirtualBox will use VT-x automatically. Screenshot: vb-create-ubuntu-systeminfo-virtualization.png

- **Attaching the correct Ubuntu ISO to the VM:**

1. **In VirtualBox, select your VM `LAB-SIEM-WAZUH-01`.**
2. **Click Settings → Storage.**
3. **In the Storage Tree click the optical device row (Controller: IDE → Optical Drive: IDE Secondary Device 0). If the entry currently shows `Unattended-...aux-iso.viso`, that means an auto-generated ISO has been attached.**
4. **To remove it, right click the "`Unattended-...aux-iso.viso`" then click the "Remove Attachment." You need to remove the "`Unattended-...aux-iso.viso`" from IDE before adding the iso to SATA. You can't have the same ISO mounted in two controllers (IDE + SATA) as it can confuse the VM firmware/boot order. Best practice is only one optical drive, either on IDE *or* SATA.**
5. **Now under Controller: SATA, click the CD icon with + → *Add Optical Drive*.**
6. **When "Optical Disk Selector" prompt showed, click add then browse to: C:\Users\Concepcion\Documents\blue-team-homelab\iso\ubuntu-24.04.3-live-server-amd64.iso → choose ubuntu-24.04.3-live-server-amd64.iso.**
7. **You'll see both your `.vdi` and the ISO listed under SATA. Now you're running *all-SATA*, which is usually more stable in VirtualBox. Screenshot:** ubuntu-vm-settings-storage.png

Before starting the Ubuntu VM, screenshot the summary of Ubuntu VM on vbox VMs list: ubuntu-vm-overall-setup-summary.png

---

Start ubuntu vm.

During the installation process

1. Boot menu / installer started. Action: Choose **Install Ubuntu Server**.
2. Language / Keyboard / Installation Type (optional). Skip screenshots for this to keep README clean.)
3. Network Configuration

What to do:

a. **Highlight enp0s8** (use arrow keys).
b. Press **Enter** → you'll see options.
c. Choose **Edit IPv4** → then select **Manual**.
d. Enter the following values:
   - **Address** → `192.168.56.10`
   - **Subnet** → `192.168.56.0/24 [should be in CIDR form (xx.xx.xx.xx/yy)]`
   - **Gateway** → *(leave blank)*
   - **Name servers** → `8.8.8.8`

e. Save and go back → enp0s8 should now show "Static" with your IP. Highlight "Continue without network" then press enter. Screenshot: ubuntu-network-config.png

4. Proxy Configuration (Skip screenshots for this to keep README clean.)

5. Ubuntu archive mirror configuration (take screenshot). Since initially when I enabled the network adapter 1 NAT and I got stuck on "Reading package lists…" for more than an hour, i just disabled the network adapter 1 NAT and clicked done. Screenshot: ubuntu-archive-mirror.png

6. Guided storage configuration (take screenshot). Enable "Use an entire disk" and disable "set up this disk as an LVM group." Screenshot: ubuntu-guided-storage-config.png

7. Storage Configuration: File System Summary (take screenshot). Screenshot: ubuntu-storage-config.png

8. Profile Configuration

Values:

- **Your name** → `Jelly Concepcion`
- **`Your server’s name (hostname)`** `→ wazuh-lab-01`
- **`Pick a username`** `→ labadmin`
- **`Choose a password`** `→ Dream!110100 (Screenshot:` ubuntu-profile-config.png)

10. Upgrade to Ubuntu Pro (Skip screenshots for this to keep README clean.)

11. SSH Configuration (take screenshot). Screenshot: ubuntu-ssh-config.png

12. Installation System (If stuck on “Installing kernel” for more than 1 hour, disable network adapter 1 NAT and make sure to attach iso to SATA instead in IDE.) Click “Reboot Now” after installation complete.

After clicking the “Reboot Now”, you will encounter:

“[FAILED] Failed unmounting cdrom.mount - /cdrom.

Please remove the installation medium, then press ENTER:
[FAILED] Failed unmounting cdrom.mount - /cdrom.”

The “Failed unmounting cdrom.mount - /cdrom” error

- This happens a lot with Ubuntu Server inside VirtualBox.

- It just means the OS tried to eject the ISO but VirtualBox had already unmounted it.

- It’s harmless. What you need to do is checked VM → Settings → Storage then if you see the optical drive “Empty,” that’s VirtualBox automatically cleaning up the ISO. It’s expected after install. Going back to the VM and pressing **Enter** to continue was the correct move.

After the failed message, this will show up:

“Ubuntu 24.04.3 LTS wazuh-lab-01 tty1

wazuh-lab-01 login:”. Screenshot: **First login prompt** (`wazuh-lab-01 login:` screen before typing anything): `ubuntu-login-screen.png`

Just type:
“wazuh-lab-01 login: labadmin
Password: Dream!110100”
When typing the password, Linux hides your typing when you enter a password.

On the console:

- You won’t see `*` or dots.
- You won’t see the cursor move.
- It looks like “nothing is happening.”

But your keystrokes are still being read.

Just carefully type your password (`Dream!110100`) and hit **Enter**.
If it’s correct, you’ll get a shell prompt like:
labadmin@wazuh-lab-01:~$
Screenshot: **Successful shell prompt after login:** (`labadmin@wazuh-lab-01:~$`)
`ubuntu-shell-prompt.png`

Enable the Network Adapter 1 = NAT

Do these commands **now** from `labadmin@wazuh-lab-01:~$`.

Quick bring-up & DHCP:

```
# bring the NAT interface up
sudo ip link set enp0s3 up
```

Your NAT NIC `enp0s3` is still **down** because `dhclient` isn’t installed on Ubuntu 24.04 by default.

Ubuntu 24.04 uses **systemd-networkd / netplan** and `systemd-networkd` instead of `dhclient`. Fix it the modern way.

Step 1: Bring the interface up with `systemd-networkd`
Run:

```
sudo systemctl restart systemd-networkd
```

Then check:
ip a show enp0s3 (Screenshot: **Checking NAT interface after restart** (`ip a show enp0s3`): `ubuntu-ip-a-enp0s3.png)`
ip route (Screenshot: **Routing table check** (`ip route`) before netplan edit (shows missing default route): `ubuntu-ip-route-before.png)`

Problem:

- ✅ `enp0s3` (NAT adapter) is **UP**,
- ❌ but it only got **IPv6**, no IPv4 (we need a 10.0.2.x from VirtualBox NAT),
- ❌ routing table has only the host-only `192.168.56.0/24` → no default route,
- That’s why `ping 8.8.8.8` = *network unreachable*.

So the VM still has **no internet**. The next step is to fix the network configuration so `enp0s3` gets IPv4 via DHCP.

Step 1: Edit netplan config
Open the netplan file:

```
sudo nano /etc/netplan/50-cloud-init.yaml
```

Replace its contents with this (indentation is important — 2 spaces per level):

```
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: true
      dhcp6: false
    enp0s8:
      addresses:
        - 192.168.56.10/24
      dhcp4: false
      dhcp6: false
```

Screenshot: **Editing netplan file** (`sudo nano /etc/netplan/50-cloud-init.yaml` before save): `ubuntu-netplan-edit.png`
Save (`CTRL+O`, `ENTER`), then exit (`CTRL+X`).

Step 2: Apply changes
sudo netplan apply

If you see a warning, that warning means your **netplan config file permissions are too open** (anyone can read/write it). Netplan refuses to fully trust it until we fix the permissions.

Step 1: Fix permissions
Run this:
sudo chmod 600 /etc/netplan/50-cloud-init.yaml
That makes it **read/write only by root** (secure).

Step 2: Re-apply config
sudo netplan apply (Screenshot: **Permissions fix** (`sudo chmod 600 /etc/netplan/50-cloud-init.yaml`) + reapply: `ubuntu-netplan-permissions.png)`

Step 3: Verify
Check if NAT got an IPv4:
ip a show enp0s3 (Screenshot: **Verify NAT IPv4 assigned** (`ip a show enp0s3` after netplan apply): `ubuntu-ip-a-after.png)`
ip route (Screenshot: **Routing table with default route** (`ip route` after fix): `ubuntu-ip-route-after.png)`

You should see something like:

- `enp0s3` has IPv4: `10.0.2.15/24`

- `ip route` has: `default via 10.0.2.2 dev enp0s3`

Step 4: Test connectivity
ping -c 3 8.8.8.8
ping -c 3 google.com

Screenshot: **Connectivity test** (`ping -c 3 8.8.8.8` and `ping -c 3 google.com`): `ubuntu-connectivity-test.png`

If both work → internet is fixed.
Then run:
sudo apt update && sudo apt upgrade -y (Screenshot: **Update & upgrade run** (`sudo apt update && sudo apt upgrade -y`): `ubuntu-system-update.png)`
sudo apt install -y curl gnupg apt-transport-https ca-certificates (Screenshot: **Install dependencies (curl, gnupg, etc.):** `ubuntu-install-dependencies.png)`

After that, your Ubuntu VM will be fully updated and ready for the **Wazuh SIEM installation**.

You're basically at the point where you can start building your SOC lab.

Run the Wazuh quick installer
Once updates finish, run the **official Wazuh quick installer**:

1. Fetch the versioned installer explicitly:
   curl -sO https://packages.wazuh.com/4.12/wazuh-install.sh
   Screenshot: **Download installer script** (`curl -sO https://packages.wazuh.com/4.12/wazuh-install.sh`): `ubuntu-wazuh-download.png`

2. **Inspect the file:**

   file wazuh-install.sh

   You should see something like "ASCII text executable."

3. **Make the script executable:**

   chmod +x wazuh-install.sh

   `chmod +x` ensures it's executable. Screenshot: Make script executable (`chmod +x wazuh-install.sh`): `ubuntu-wazuh-chmod.png`

4. **Run it:**

   sudo ./wazuh-install.sh -a

   Screenshot: Running the installer (`sudo ./wazuh-install.sh -a`) — capture mid-install: `ubuntu-wazuh-install.png`

The `-a` flag installs **all-in-one** (manager + indexer + dashboard).

The installer will print the **dashboard URL** and **admin password** at the end.

You can access the web interface: https://<wazuh-dashboard-ip>:443
User: admin
Password: rWR7wFg50LsU.ilo1KTo?Kdp5ENw?C05
Screenshot: **Installer completion screen** (shows dashboard URL + admin password): `ubuntu-wazuh-install-complete.png`

Since your Wazuh all-in-one installer finished successfully and gave you the dashboard URL + password, now you need to:

1) Verify services

Run:

```
sudo systemctl status wazuh-manager wazuh-indexer wazuh-dashboard
```

All 3 should be **active (running)**.
Screenshot: **Service status check** (`sudo systemctl status wazuh-manager wazuh-indexer wazuh-dashboard`): `ubuntu-wazuh-services.png`

2) Access the dashboard

- From your **host PC's browser** (not the VM), go to:
  https://192.168.56.10
  (this works because `192.168.56.10` is your host-only adapter address).

- Accept the self-signed certificate warning. Screenshot: **First login to dashboard** (browser on host → `https://192.168.56.10`, accept cert warning): `ubuntu-wazuh-dashboard-login.png`

- Login with:

Username: admin
Password: rWR7wFg50LsU.ilo1KTo?Kdp5ENw?C05

Screenshot: for the **actual Wazuh login form**, you'll take another screenshot: `ubuntu-wazuh-dashboard-auth.png`

3) Confirm login

Once you log in successfully, you should land on the Wazuh dashboard (default workspace).

Screenshot: **Successful Wazuh dashboard landing page:** `ubuntu-wazuh-dashboard.png`

Create the evidence file on Ubuntu

(Note: You first need to cd "C:/Users/Concepcion/Documents/blue-team-homelab on gitbash then run git log --oneline and find the SHA you need.)

Run:

```
# Ubuntu / wazuh-lab-01
echo "blue-team-homelab commit: b3d2839 | $(date -u '+%Y-%m-%d %H:%M
UTC')" | sudo tee /etc/lab-evidence-ubuntu.txt
cat /etc/lab-evidence-ubuntu.txt
```

Screenshot: ubuntu-evidence-file.png

Take a VirtualBox snapshot (preserve state)

GUI method (VirtualBox)

1. Open VirtualBox, select the VM (e.g., `LAB-SIEM-WAZUH-01`).
2. Click **Snapshots** (right pane) → **Take** (camera icon).
3. Name: `M1-Ubuntu-Evidence-b3d2839`
   Description: `Ubuntu lab at commit b3d2839 (Wazuh installed, evidence file added)`

Screenshot: vb-ubuntu-snapshot.png