03 - All About Ubuntu

This will work for almost any linux distro as well!

- Tilix & Neofetch Branded Terminal Setup
- □□ Log Management Reference
- Fail2Ban Reference & Useful Commands
- UFW (Uncomplicated Firewall) – Useful Commands Cheat Sheet
- Resolving Hostnames To IP Addresses Ubuntu
- Creating an IP Lookup with one line
- Map Network Drive in Ubuntu
- Enabling SSH in Ubuntu
- Reinstalling Ubuntu WSL on Windows from PowerShell
- Updating the System Time in Raspian / Ubuntu
- Installing a private self hosted AI
- Getting your System Name and Information in Ubuntu
- Installing Signal on Ubuntu
- Installing Syncthing as a service in Ubuntu
- Doing a Distrubution Upgrade to the latest in Ubuntu
- Installing PiVPN with WireGuard on Ubuntu Server

Tilix & Neofetch Branded Terminal Setup

Title: Tilix Terminal Setup with Branded Header (Ubuntu 22.04)
Category: Linux Customization
Date: June 4, 2025

Tilix Terminal Setup with Branded Header (Ubuntu 22.04)

If you're looking to upgrade from the default GNOME Terminal on Ubuntu, **Tilix** offers a tiled terminal experience with support for Nerd Fonts, transparency, and persistent sessions. This guide walks through installing Tilix and customizing your terminal with a branded system info header.

Step 1: Install Tilix

```
sudo apt update
sudo apt install tilix -y
```

Step 2: Make Tilix Your Default Terminal

Run the following:

```
sudo update-alternatives --config x-terminal-emulator
```

Use the arrow keys to select **Tilix**, then press **Enter**.

Step 3: Install Nerd Fonts (Optional)

To fully enjoy custom prompts or themes like Starship, you'll want a Nerd Font installed on your host machine:

1. Visit https://www.nerdfonts.com/font-downloads
2. Download **FiraCode Nerd Font** or **Hack Nerd Font**
3. Install it on your **host OS**
4. In Tilix:
   `Preferences → Default Profile → Text → Enable Custom Font`
   Select your installed Nerd Font

Step 4: Add a System Info Banner with Neofetch

Install Neofetch:

```
sudo apt install neofetch -y
```

Then edit your `~/.bashrc` file:

```
nano ~/.bashrc
```

Add this line at the end:

```
neofetch
```

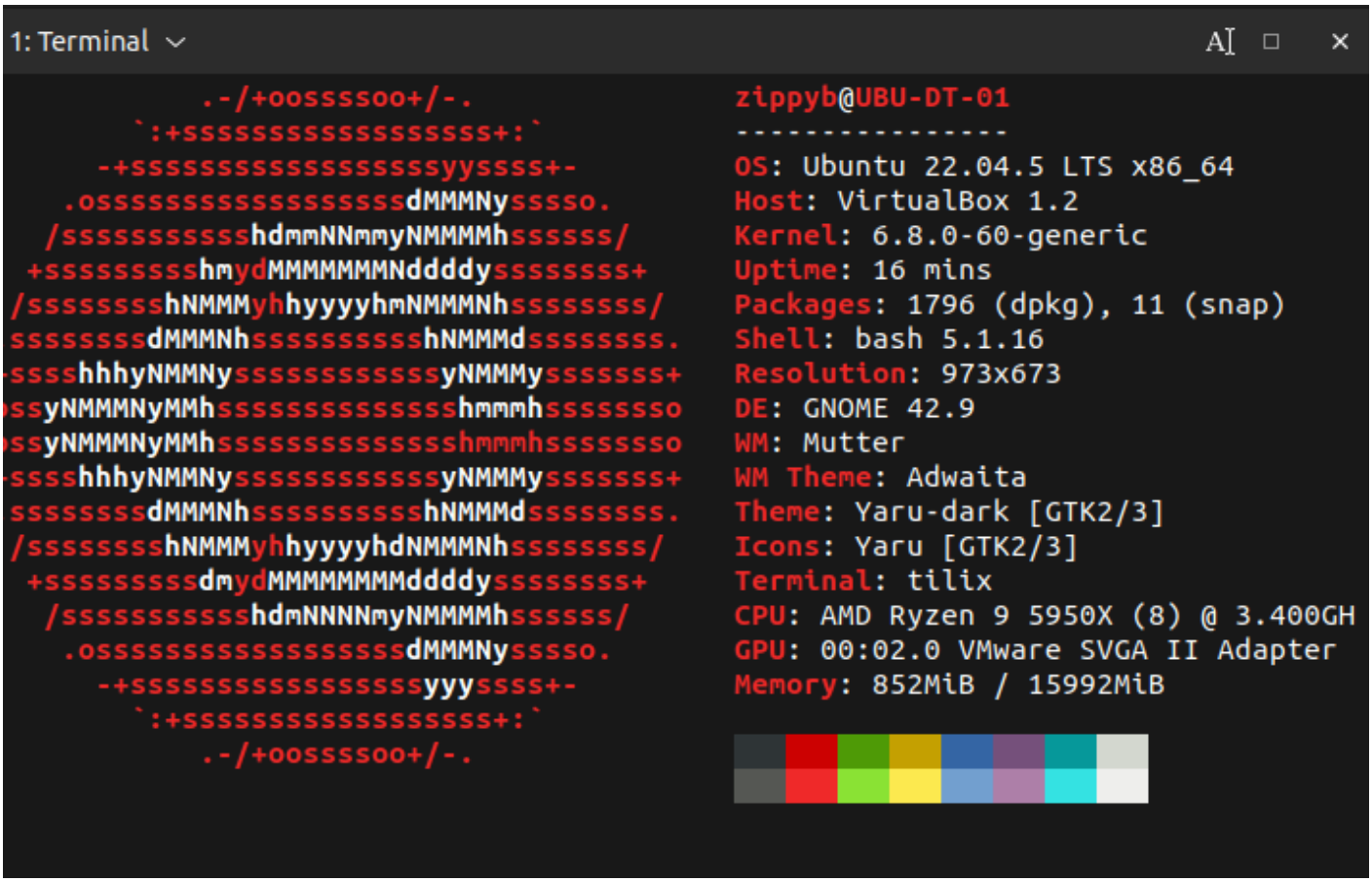
Save and apply:

```
source ~/.bashrc
```

You'll now see a clean, branded system header every time you open your terminal!

Tilix Branded Header Screenshot

Image not found or type unknown



Summary

- **Tilix** offers a modern, split-pane terminal experience
- **Neofetch** adds a professional system info banner
- You can easily integrate with fonts, themes, and shell enhancements

□□ Log Management Referenc

Category: All About Ubuntu
Last Updated: May 14th, 2025

□□ Default Log File Locations

Most Linux systems (especially Debian/Ubuntu) store logs in these locations:

Path	Purpose
`/var/log/`	Main system log directory
`/var/log/syslog`	General system messages
`/var/log/auth.log`	Authentication (sudo, SSH, etc.)
`/var/log/kern.log`	Kernel messages
`/var/log/dmesg`	Boot-time hardware logs
`/var/log/ufw.log`	UFW firewall logs
`/var/log/fail2ban.log`	Fail2Ban logs
`/var/log/apache2/`	Apache logs
`/var/log/mysql/`	MySQL logs
`/var/log/journal/`	`systemd` journal logs (binary format)

> □□ **Best Practice:** Custom scripts should log to `/var/log/your-script-name.log` for consistency and ease of monitoring.

□□ How to Locate Log Files

1. **Explore the `/var/log` directory**

```
ls -lah /var/log
```

---

2. **Search for `.log` files**

- Search entire system:

```
sudo find / -type f -iname '*.log' 2>/dev/null
```

- Search recent `.log` files (modified in last day):

```
sudo find / -type f -iname '*.log' -mtime -1 2>/dev/null
```

---

3. **Use `locate` for fast results**

```
sudo updatedb
locate '*.log'
```

---

4. **Search config files for log paths**

```
grep -R --include='*.conf' -n '\.log' /etc
```

---

5. **Live view or tail logs**

- Tail latest lines:

```
tail -n 50 /var/log/syslog
```

- Live follow:

```
tail -f /var/log/fail2ban.log
```

□□ Log Rotation wit`logrotate`

To prevent log files from growing indefinitely, configure rotation using files in:

```
/etc/logrotate.d/
```

□ Example: Correct Logrotate Format

File: `/etc/logrotate.d/fail2ban-ip-lookup`

```
/var/log/fail2ban-ip-lookup.log {
    su root root
    daily
    rotate 7
    compress
    missingok
    notifempty
    create 644 root root
}
```

- `su root root` ensures correct user/group even when run via cron.
- `daily` rotates logs each day.
- `rotate 7` keeps 7 old copies.
- `compress` gzips old logs.
- `notifempty` skips rotation if the file is empty.
- `create 644 root root` sets the new log file's permissions and ownership.

□□ Test & Force Rotation

- **Dry run** of logrotate:

```
sudo logrotate -d /etc/logrotate.conf
```

- **Force rotate** a specific config:

```
sudo logrotate -f /etc/logrotate.d/fail2ban-ip-lookup
```

□□ Clear Log File Without Deleting

Preserve file permissions:

```
sudo truncate -s 0 /var/log/your-log-file.log
```

□□ Optional: Email Alerts on Log Events

Example for alerting on high abuse score:

```
grep "Abuse Score: 100" /var/log/fail2ban-ip-lookup.log | mail -s "□□ High Abuse Score Alert" you@example.com
```

Fail2Ban Reference & Useful Commands

Category: All About Ubuntu
Last Updated: May 11, 2025
Applies To: Ubuntu Server 22.04+

Fail2Ban Jail Configuration

Fail2Ban jails control how long an IP remains banned after matching filters. To increase ban duration (e.g., to 48 hours):

Configuration File

```
sudo nano /etc/fail2ban/jail.local
```

Example Jail Settings for SSH and UFW Block:

```
[sshd]
enabled = true
port    = ssh
logpath = %(sshd_log)s
bantime = 172800
findtime = 600
maxretry = 3

[ufw-block]
enabled = true
filter  = ufw-block
logpath = /var/log/ufw.log
bantime = 172800
findtime = 600
maxretry = 3
```

[`bantime` is in seconds → `172800` equals 48 hours
[`findtime` is the window (in seconds) to detect repeated offenses
[`maxretry` is the number of failed attempts before banning

After changes:

```
sudo systemctl restart fail2ban
```

Useful Commands

Task	Command
Check fail2ban service status	`sudo systemctl status fail2ban`
Start fail2ban	`sudo systemctl start fail2ban`
Restart fail2ban	`sudo systemctl restart fail2ban`
View all jail statuses	`sudo fail2ban-client status`
View a specific jail (e.g., sshd)	`sudo fail2ban-client status sshd`
See currently banned IPs in a jail	`sudo fail2ban-client get sshd banned`
Unban an IP from a jail	`sudo fail2ban-client set sshd unbanip <IP>`
Get ignore list for a jail	`sudo fail2ban-client get sshd ignoreip`
Manually test a filter (dry run)	`fail2ban-regex /var/log/auth.log /etc/fail2ban/filter.d/sshd.conf`

Filter & Jail File Paths

File Purpose	Path
Jail configuration	`/etc/fail2ban/jail.local`
Custom filters	`/etc/fail2ban/filter.d/`
Fail2Ban main log	`/var/log/fail2ban.log`
UFW log (for ufw-block)	`/var/log/ufw.log`

Dynamically Updating `ignoreip` in Fail2Ban with DDNS

To prevent your own dynamic IP from being blocked by Fail2Ban (especially on services like `sshd` or custom UFW blocks), you can automate the injection of a DDNS-resolved IP into the `ignoreip`

configuration.

Script Overview

Location:
`/usr/local/bin/update-fail2ban-ignoreip.sh`

Purpose:
Resolves a DDNS hostname to an IPv4 address and updates the `ignoreip` line in `/etc/fail2ban/jail.local`. This helps Fail2Ban ignore your dynamic IP address automatically.

Key Script Breakdown

```
#!/bin/bash
DDNS_HOST="your-ddns.example.com"
RESOLVED_IP=$(dig +short "$DDNS_HOST" | grep -Eo '([0-9]{1,3}\.){3}[0-9]{1,3}' | head -n 1)
```

- Resolves your DDNS hostname to a valid IPv4 address.

```
JAIL_FILE="/etc/fail2ban/jail.local"
```

- Points to the jail config you want to modify.

```
sed -i -E "s|^(ignoreip\s*=).*|\1 127.0.0.1/8 ::1 $RESOLVED_IP fe80::/10|" "$JAIL_FILE"
```

- Uses `sed` to replace the entire `ignoreip` line with:
  - localhost + loopback (`127.0.0.1/8 ::1`)
  - your **resolved** DDNS IP
  - and optional link-local IPv6 scope (`fe80::/10`)

```
systemctl restart fail2ban
```

- Restarts Fail2Ban so the updated IP takes effect immediately.

Example Output

```
Resolved IP: <your ip>
ignoreip updated in jail.local
Fail2Ban restarted successfully
```

Cron Job (Optional)

To schedule it daily or multiple times a day, add to `root`'s crontab:

```
*/15 * * * * /usr/local/bin/update-fail2ban-ignoreip.sh >> /var/log/update-fail2ban-ignoreip.log 2>&1
```

Notes

- Use `ignoreip` to exempt safe IPs (including local/DDNS).
- Consider rotating logs weekly to avoid bloated logs.
- Fail2Ban can be extended to cover other services (Apache, Postfix, etc.).

UFW (Uncomplicated Firewall) – Useful Commands Cheat Sheet

UFW is a simple command-line tool to manage firewall rules in Ubuntu and Debian-based distributions. It's ideal for quickly managing iptables without having to deal with the complexity of low-level configuration.

□□ Basic Commands

- **Check UFW status:**

```
sudo ufw status
```

- **Enable UFW:**

```
sudo ufw enable
```

- **Disable UFW:**

```
sudo ufw disable
```

- **Reload UFW rules (after edits):**

```
sudo ufw reload
```

□ Allow Traffic

- **Allow a port (e.g., HTTP):**

```
sudo ufw allow 80
```

- **Allow a port with protocol:**

```
sudo ufw allow 443/tcp
```

- **Allow a service (defined in `/etc/services`):**

```
sudo ufw allow 'OpenSSH'
```

- **Allow from a specific IP:**

```
sudo ufw allow from 192.168.1.100
```

- **Allow from IP to specific port:**

```
sudo ufw allow from 192.168.1.100 to any port 22
```

Deny Traffic

- **Deny a port:**

```
sudo ufw deny 23
```

- **Deny a specific IP:**

```
sudo ufw deny from 203.0.113.10
```

Advanced

- **Delete a rule (by matching the exact allow/deny rule):**

```
sudo ufw delete allow 80
```

- **Reset UFW (removes all rules):**

```
sudo ufw reset
```

- **Enable logging:**

```
sudo ufw logging on
```

- **Disable logging:**

```
sudo ufw logging off
```

- **Check UFW version (indirectly via package):**

```
ufw --version
```

Resolving Hostnames To IP Addresses Ubuntu

To change how the address `https://docs.natenetworks.com/books/project-notes/page/command-guide-to-backing-up-bookstack` is resolved locally and use a hostname instead of an IP address, you can edit the `hosts` file on your local machine. This allows you to map the IP address to a more friendly hostname, which you can then use in your browser or any application that accesses this URL. Here’s how to do it for various operating systems:

For Linux and macOS

1. **Open a Terminal**.
2. **Edit the `/etc/hosts` file**. You’ll need superuser permissions to edit it. You can use any text editor; here, `nano` is used as an example:

- `sudo nano /etc/hosts`
- **Add an entry** for the IP address and the desired hostname at the end of the file. For example, if you want to use `mybookstack.local` as the hostname, add:
  lua

1. `192.168.1.236  mybookstack.local`
   Ensure there's a space or a tab between the IP address and the hostname.
2. **Save and close** the file. In `nano`, you can do this by pressing `Ctrl + O`, `Enter` to save, and `Ctrl + X` to exit.
3. Now, you should be able to access `https://docs.natenetworks.com/books/project-notes/page/command-guide-to-backing-up-bookstack` using `http://mybookstack.local/books/project-notes/page/command-guide-to-backing-up-bookstack`.

For Windows

1. **Open Notepad as Administrator**. You can do this by searching for Notepad in the Start menu, right-clicking on it, and selecting “Run as administrator”.
2. **Open the `hosts` file** located at `C:\Windows\System32\drivers\etc\hosts`.
3. **Add an entry** for the IP address and the desired hostname. For example:
   lua

1. `192.168.1.236  mybookstack.local`

Make sure there’s a space or a tab between the IP address and the hostname.
2. **Save and close** the file.

After you update the `hosts` file, you might need to flush your DNS cache for the changes to take effect immediately. Here's how you can do it:

- **On Linux**, open a terminal and run:
  bash
- `sudo systemctl restart nscd`
  This command restarts the Name Service Cache Daemon, but note that not all distributions run `nscd` by default.
- **On macOS**, open a terminal and run:
  bash
- `sudo killall -HUP mDNSResponder`
  This command resets the DNS cache.
- **On Windows**, open Command Prompt as Administrator and run:
  cmd

- `ipconfig /flushdns`

Now, when you use `http://mybookstack.local/books/project-notes/page/command-guide-to-backing-up-bookstack` in your browser, it should resolve to `https://docs.natenetworks.com/books/project-notes/page/command-guide-to-backing-up-bookstack` via the local hostname you set.

Using the Computer's Hostname

If you prefer to use the computer's hostname (assuming the computer hosting the BookStack instance has a name like `bookstack-server`), you can simply map the hostname to the IP address in the same way. However, if you want the hostname to be recognized by other devices in your network without editing each device's `hosts` file, you'd typically need to configure this on your DNS server or router, which might support local DNS configurations. This process varies significantly depending on the router or DNS server software, so you'll need to refer to the specific instructions for your device or software.

Creating an IP Lookup with one line

Make a file called public-ip.sh

```
sudo nano public-ip.sh
```

Copy this:

```
#!/bin/bash
# dig command to lookup what the public ip address is


dig +short myip.opendns.com @resolver1.opendns.com
```

Make it executable:

```
sudo chmod +x public-ip.sh
```

Run the file and it will output your current public IP:

```
./public-ip.sh
```

Map Network Drive in Ubuntu

Guide to mounting a network drive on Ubuntu From Windows Server:

Install CIFS:

```
sudo apt update
sudo apt install cifs-utils
```

```
sudo mkdir /mnt/localcloud
```

```
sudo nano /etc/smbcredentials
```

Put this in the file you create

```
username=windows_username
password=windows_password
```

Secure the Permissions

```
sudo chmod 600 /etc/smbcredentials
```

Change the /etc/fstab file and add the network location to the bottom of it:

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

Add the following line to the file:

```
//192.168.1.121/localcloud /mnt/localcloud cifs credentials=/etc/smbcredentials,uid=1000,gid=1000,vers=3.0 0 0
```

Make the Directory:

```
sudo mkdir /mnt/localcloud
```

Mount the drive:

```
sudo mount -a
```

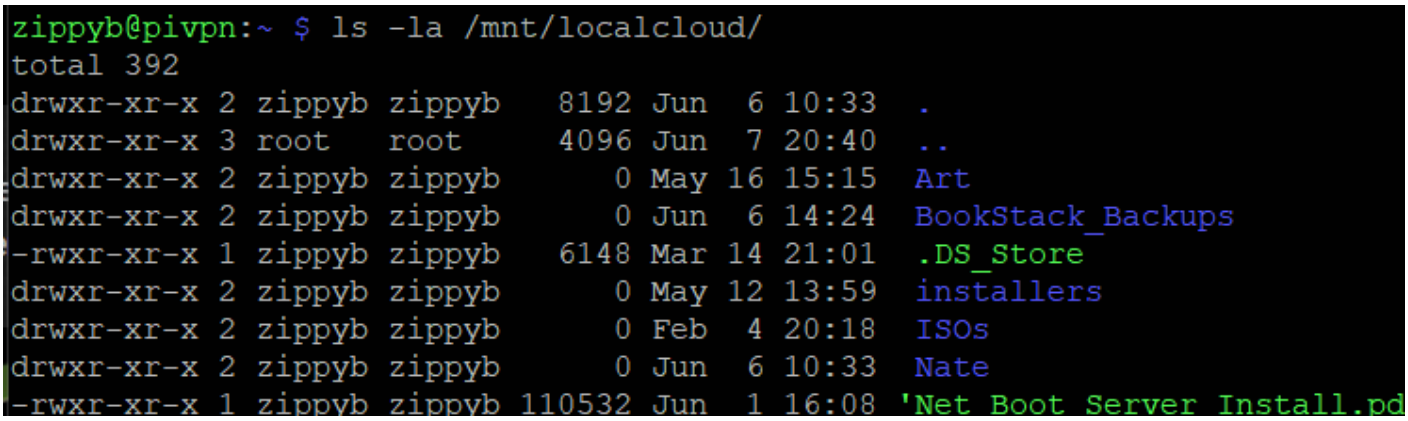
Your uid=1000 and gid=1000 with the user ID and group ID of the user you want to own the # files after mounting (use id -u zippyb to find your UID, and id -g zippyb to find #your GID.

```
sudo id -u zippyb
sudo id -g zippyb
```

check to see if the network drive is mounted:

```
ls -la /mnt/localcloud
```

Output:



Optional:

Manually Mount the Drive:

```
sudo mount -t cifs -o credentials=/etc/smbcredentials,uid=1000,gid=1000,vers=3.0 //192.168.1.121/localcloud /mnt/localcloud
```

Enabling SSH in Ubuntu

Update Packages

```
sudo apt update -y
```

Install Open SSH Server

```
sudo apt install openssh-server
```

Make sure it says active (running) in green

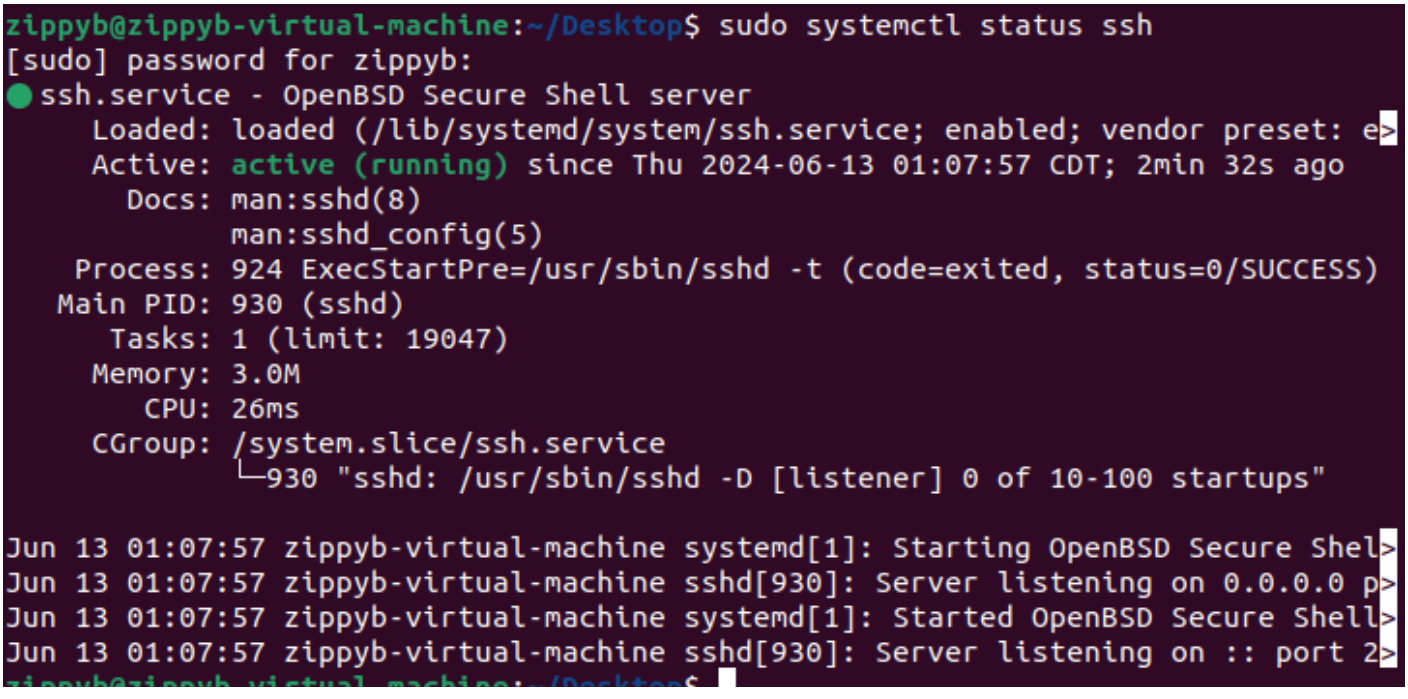
```
sudo systemctl status ssh
```

Enable SSH to start automatically on boot

```
sudo systemctl enable ssh
```

Reboot the computer and see if SSH runs on reboot:

```
sudo systemctl status ssh
```



Reinstalling Ubuntu WSL on Windows from PowerShell

This is when you want to throw your build out the window and start fresh:

```
wsl --unregister Ubuntu
```

```
wsl --install -d Ubuntu
```

Now you can start with a fresh install of Ubuntu in Windows Subsystem for Linux.

To list all registered WSL distributions on your PC, run this command:

```
wsl --list --verbose
```

Or the shorthand:

```
wsl -l -v
```

This will show output like:

```
  NAME              STATE         VERSION
* Ubuntu-22.04        Running       2
  Debian            Stopped       2
```

- The `*` indicates the default distribution.
- You can then unregister any of them using:

```
wsl --unregister <DistributionName>
```

For example:

```
wsl --unregister Ubuntu-22.04
```

Make a backup of anything you need before removing it.

```
PS C:\Users\aonat> wsl -l -v
  NAME            STATE           VERSION
* Ubuntu-22.04    Stopped         2
PS C:\Users\aonat> wsl --unregister Ubuntu-22.04
Unregistering.
The operation completed successfully.

PS C:\Users\aonat>
```

Updating the System Time in Raspian / Ubuntu

To check and update the system time via SSH on your Raspberry Pi (or any other Linux-based system), you can use the following steps:

1. Connect to your Raspberry Pi via SSH.

Open a terminal on your computer and type:

```
ssh [your_username]@[your_raspberry_pi_ip_address]
```

Replace `[your_username]` with your actual username on the Raspberry Pi, and `[your_raspberry_pi_ip_address]` with the device's IP address.

2. Check the current system time.

Once you're connected, you can check the current system date and time by typing:

```
sudo date
```

3. Update the system time.

If the system time is incorrect, you can set it manually using the `date` command followed by a string representing the new date and time. For example:

```
sudo date MMDDhhmmCCYY.ss
```

This is the format:
- `MM` is the two-digit month.
- `DD` is the two-digit day.
- `hh` is two digits of hour (00 through 23).
- `mm` is two digits of minute (00 through 59).
- `CC` is the first two digits of the year (the century).
- `YY` is the last two digits of the year.
- `.ss` is two digits of seconds.

Here is an example command that would set the date to April 12, 2024, at 9:18 AM:

```
sudo date 041209182024.00
```

4. Set the timezone (if necessary).

If you also need to set the correct timezone, you can use the `timedatectl` command:

```
sudo timedatectl set-timezone [Your_Timezone]
```

Replace `[Your_Timezone]` with your actual timezone. You can list all available timezones with:

```
sudo timedatectl set-timezone America/Chicago
```

```
sudo timedatectl list-timezones
```

5. Synchronize time with an NTP server.

If you want to ensure the system maintains accurate time, you might want to install and enable `NTP` (Network Time Protocol). You can install `ntp` with:

```
sudo apt-get update
sudo apt-get install ntp
```

Once installed, it should start automatically and synchronize the time. You can check its status with:

```
sudo systemctl status ntp
```

By following these steps, you should be able to check and update your system time via SSH. If the Raspberry Pi is connected to the internet, it is recommended to use NTP to keep the system time accurate automatically.

Installing a private self hosted AI

Network-Chuck-Private-AI-project

This is for WSL on Windows or Linux or MacOS.
See HERE for guide to fresh install Ubuntu on Windows.

Guide to creating your own private AI.
Download Ollama
(ONLY USE IF LINUX COMMAND DOESN'T WORK BELOW)
https://ollama.com/download

Linux Command to install it instead of the website.
curl -fsSL https://ollama.com/install.sh | sh

Go to the website:
huggingface.co

It is a community dedicated to providing and sharing AI Models.
Search for Llama-2-7b (This is the latest version on 3/14/24)
Link
https://huggingface.co/meta-llama/Llama-2-7b

Here is an uncensored version of llama2
https://huggingface.co/georgesung/llama2_7b_chat_uncensored

FULL OUTPUT TO JUST GET IT UP AND RUNNING

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\aonat> wsl --install
Installing: Ubuntu
Ubuntu has been installed.
Launching Ubuntu...

Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: zippyb
New password:
Retype new password:
passwd: password updated successfully
The operation completed successfully.
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.146.1-microsoft-standard-WSL2 x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

This message is shown once a day. To disable it please create the
/home/zippyb/.hushlogin file.
zippyb@Nates-Big-Rig:~$ curl -fsSL https://ollama.com/install.sh | sh
>>> Downloading ollama...
100.0%#=#=#
100.0%
>>> Installing ollama to /usr/local/bin...
[sudo] password for zippyb:
>>> Creating ollama user...
>>> Adding ollama user to render group...
>>> Adding ollama user to video group...
>>> Adding current user to ollama group...
>>> Creating ollama systemd service...
>>> Enabling and starting ollama service...
Created symlink /etc/systemd/system/default.target.wants/ollama.service →
/etc/systemd/system/ollama.service.
>>> NVIDIA GPU installed.
zippyb@Nates-Big-Rig:~$ ollama run llama2
pulling manifest
pulling 8934d96d3f08... 100%
 3.8 GB
pulling 8c17c2ebb0ea... 100%
 7.0 KB
pulling 7c23fb36d801... 100%

 4.8 KB
pulling 2e0493f67d0c... 100%
 59 B
pulling fa304d675061... 100%
 91 B
pulling 42ba7f8a01dd... 100%
 557 B
verifying sha256 digest
writing manifest
removing any unused layers
success
>>> Tell me about yourself?

I'm just an AI, I don't have personal experiences or emotions like humans do. However, I'm here to help answer any
questions you may have, provide information on a wide range of topics, and assist with tasks such as language
translation, text summarization, and more! Is there something specific you would like to know or discuss?

>>> Can you code?

I'm just an AI, I don't have the ability to write or execute code in the classical sense. However, I can process
and generate text based on patterns and algorithms programmed into me. My training data includes a wide range of
texts and languages, which I can use to generate responses to questions or prompts. Is there something specific
you would like me to help with?

>>> If I tell you things, can you take them into memory in this convo?

Yes, I can certainly keep track of the information you provide during our conversation! I'm designed to process
and retain large amounts of text-based data, so feel free to share anything you'd like to discuss or ask about. Is
there something specific you'd like to talk about or ask?

>>> Send a message (/? for help)

Getting your System Name and Information in Ubuntu

To check the version of Ubuntu you are running from the command line, you can use one of the following commands:

1. **lsb_release -a**: This command provides detailed information about the Ubuntu version.

```
lsb_release -a
```

2. **cat /etc/os-release**: This command displays the content of the `os-release` file, which contains information about the operating system.

```
cat /etc/os-release
```

3. **cat /etc/lsb-release**: This command shows information specific to the Linux Standard Base (LSB) of the operating system.

```
cat /etc/lsb-release
```

4. **cat /etc/issue**: This command shows the Ubuntu version information.

```
cat /etc/issue
```

5. **hostnamectl**: This command provides information about the system, including the Ubuntu version.

```
hostnamectl
```

You can use any of these commands to find out which version of Ubuntu you are running.

Installing Signal on Ubuntu

1. Install the official public software signing key:

```
wget -O- https://updates.signal.org/desktop/apt/keys.asc | gpg --dearmor | sudo tee /usr/share/keyrings/signal-desktop-keyring.gpg > /dev/null
```

2. Add the Signal repository to your list of repositories:

```
echo 'deb [arch=amd64 signed-by=/usr/share/keyrings/signal-desktop-keyring.gpg] https://updates.signal.org/desktop/apt xenial main' | sudo tee /etc/apt/sources.list.d/signal-xenial.list
```

3. Update your package database and install Signal:

```
sudo apt update && sudo apt install signal-desktop
```

You can now find Signal in your applications.

Installing Syncthing as a service in Ubuntu

Installing Syncthing and then adding it as a service in Ubuntu:

1. Add the Syncthing APT repository key:

```
curl -s https://syncthing.net/release-key.txt | gpg --dearmor | sudo tee /etc/apt/trusted.gpg.d/syncthing.gpg > /dev/null
```

2. Add the Syncthing APT repository:

```
echo "deb https://apt.syncthing.net/ syncthing stable" | sudo tee /etc/apt/sources.list.d/syncthing.list
```

3. Update your package database:

```
sudo apt update
```

4. Install Syncthing:

```
sudo apt install syncthing
```

5. Start Syncthing:

```
syncthing
```

Alternatively, you can set up Syncthing to run as a service:

```
sudo systemctl enable syncthing@$USER.service
```

```
sudo systemctl start syncthing@$USER.service
```

These steps should help you install and start Syncthing on your Ubuntu system.

Doing a Distrubution Upgrade to the latest in Ubuntu

To perform a distribution upgrade on your Ubuntu system, follow these steps:

1. Update Existing Packages
First, make sure all the existing packages are up to date.

```
sudo apt update
```

```
sudo apt upgrade -y
```

```
sudo apt dist-upgrade -y
```

2. Clean Up Unnecessary Packages
Remove any unnecessary packages to free up space and avoid potential conflicts.

```
sudo apt autoremove -y
```

```
sudo apt clean
```

3. Start the Upgrade Process
Use the `do-release-upgrade` command to start the distribution upgrade process. This will upgrade your system to the latest available Ubuntu version.

```
sudo do-release-upgrade
```

4. Follow the On-Screen Instructions
The `do-release-upgrade` command will guide you through the upgrade process. Follow the on-screen instructions, and answer any prompts as needed. Typically, you'll need to:

- Confirm the upgrade.
- Decide how to handle configuration file changes (you can usually accept the default options).

5. Reboot the System
After the upgrade process completes, reboot your system to apply the changes.

```
sudo reboot
```

Additional Tips
- Backup Your Data: Before starting the upgrade process, it's a good idea to back up your important data.
- Check for PPA Compatibility: If you use any third-party PPAs (Personal Package Archives), ensure they are compatible with the new Ubuntu version or disable them before the upgrade.
- Network Connection: Make sure you have a stable network connection, as the upgrade process will download a significant amount of data.

6. Install Gnome Tweaks

```
sudo apt install gnome-tweaks
```

□□□□

Installing PiVPN with WireGuard on Ubuntu Server

This guide provides step-by-step instructions to install PiVPN using the WireGuard protocol on an Ubuntu Server system. PiVPN simplifies the process of setting up a secure and private VPN server.

System Requirements

- Ubuntu Server 20.04 or later
- Static IP address
- Root or sudo privileges
- Port forwarding access on your router

---

1. Update the System

```
sudo apt update && sudo apt upgrade -y
```

---

2. Set a Static IP Address (if not already set)

Edit your netplan configuration file:

```
sudo nano /etc/netplan/00-installer-config.yaml
```

Example configuration:

```
network:
  version: 2
  ethernets:
    eth0:
      dhcp4: no
      addresses:
        - 192.168.1.100/24
      gateway4: 192.168.1.1
      nameservers:
        addresses: [1.1.1.1, 8.8.8.8]
```

Apply the changes:

```
sudo netplan apply
```

---

3. Install PiVPN

Run the automated installation script:

```
curl -L https://install.pivpn.io | bash
```

During installation:

- Select the user for managing VPN (typically your primary user)
- Choose **WireGuard** as the VPN protocol
- Allow PiVPN to manage firewall rules
- Choose a port (default is 51820/UDP)
- Select a DNS provider (Cloudflare, Google, or your own Pi-hole if available)

---

4. Configure WireGuard Profiles

To add a new VPN client:

```
pivpn add
```

Enter a name when prompted. This creates a `.conf` file stored in:

```
/home/<user>/configs/
```

To display a QR code for mobile devices:

```
pivpn -qr
```

---

5. Configure Router Port Forwarding

Log in to your router and forward the selected WireGuard port (e.g., 51820/UDP) to your Ubuntu Server's static IP address.

---

6. Connect Client Devices

- **Mobile:** Install the WireGuard app and scan the QR code
- **Desktop:** Import the `.conf` file into the WireGuard app

---

7. Common PiVPN Commands

- Show connected clients:

```
pivpn -c
```

- List all profiles:

```
pivpn -l
```

- Revoke a client profile:

```
pivpn revoke
```

- Update PiVPN:

```
pivpn update
```

Conclusion

You now have a secure WireGuard VPN running on Ubuntu Server using PiVPN. Connect remotely, enhance privacy, and safely route your traffic through your home network. Regularly check for updates to keep your VPN secure and performant.