



Open Source Software

Student Details

Name: ADIREDDY LOKESH
Roll Number: 2400032210
Department: Computer Science and Engineering
University: KL University
Course: Open Source
Semester: ODD SEMESTER

Submitted to:

Dr Sripath Roy

Department of Computer Science and Engineering
KL University

Contents

1	Linux Distribution	3
1.1	Distribution Used: Ubuntu 22.04 LTS	3
1.2	Why Ubuntu?	3
1.3	Key Features of Ubuntu 22.04 LTS	3
1.4	System Specifications	3
1.5	Installation Process	3
2	Encryption and GPG	4
2.1	What is Encryption?	4
2.2	Types of Encryption	4
2.2.1	Symmetric Encryption	4
2.2.2	Asymmetric Encryption	4
2.3	GNU Privacy Guard (GPG)	4
2.4	Installing GPG	4
2.5	Generating GPG Keys	4
2.6	Listing Keys	5
2.7	Exporting Public Key	5
2.8	Encrypting Files	5
2.9	Decrypting Files	5
3	Sending Encrypted Email	5
3.1	Email Encryption Overview	5
3.2	Tools Used	5
3.3	Setting up Thunderbird with GPG	6
3.3.1	Installation	6
3.3.2	Configuring OpenPGP	6
3.4	Sending Encrypted Email	6
3.5	Receiving Encrypted Email	6
3.6	Best Practices	6
4	Privacy Tools from prism-break.org	7
4.1	What is PRISM-Break?	7
4.2	Tool 1: Signal - Encrypted Messaging	7
4.3	Tool 2: Firefox - Web Browser	7
4.4	Tool 3: ProtonMail - Encrypted Email	8
4.5	Tool 4: Tor Browser - Anonymous Browsing	8
4.6	Tool 5: VeraCrypt - Disk Encryption	9
5	Open Source License	9
5.1	License Used: MIT License	9
5.2	What is the MIT License?	9
5.3	MIT License Text	9
5.4	Why Choose MIT License?	10
5.5	Other Common Open Source Licenses	10
5.5.1	GPL (GNU General Public License)	10
5.5.2	Apache License 2.0	10

5.5.3	BSD License	10
6	Self-Hosted Server: Dashy	11
6.1	What is Dashy?	11
6.2	Why Self-Host Dashy?	11
6.3	Installation Guide.....	12
6.3.1	System Requirements	12
6.3.2	Installation Steps.....	12
6.4	Localization (Telugu Translation)	14
6.4.1	Translation Process.....	14
6.4.2	Sample Translations	14
6.5	Benefits of Self-Hosting.....	14
7	Open Source Contributions	15
7.1	PR 1: Roshanjossey/code-contributions	15
7.1.1	Issue Description	15
7.1.2	Solution Implemented	15
7.2	PR 2: firstcontributions/first-contributions	15
7.2.1	Issue Description	15
7.2.2	Solution Implemented	15
7.3	PR 3: KLGLUG/Y24OpenSourceEngineering	16
7.3.1	Issue Description	16
7.3.2	Solution Implemented	16
7.4	PR 4: iib0011/omni-tools	16
7.4.1	Issue Description	16
7.4.2	Solution Implemented	16
7.5	PR 5: PrismJS/prism	17
7.5.1	Issue Description	17
7.5.2	Solution Implemented	17
7.6	PR 6: topoteretes/cognee	17
7.6.1	Issue Description	17
7.6.2	Solution Implemented	17
7.7	PR 7: recloudstream/cloudstream	18
7.7.1	Issue Description	18
7.7.2	Solution Implemented	18
7.8	PR 8: MUNJETIJAYANTHKRISHNA/stage1	18
7.8.1	Issue Description	18
7.8.2	Solution Implemented	18
8	LinkedIn Posts	19
8.1	Post 2: Expo at KL University	19
8.2	Post 3: Blog Link to My Open Source Journey	19
9	Conclusion	20

1 Linux Distribution

1.1 Distribution Used: Ubuntu 22.04 LTS

For this project, I have used **Ubuntu 22.04 LTS** as my primary operating system.

1.2 Why Ubuntu?

Ubuntu is one of the most popular Linux distributions for several reasons:

- **User-Friendly:** Ubuntu has an intuitive interface suitable for beginners
- **Long Term Support:** LTS versions receive 5 years of security updates
- **Large Community:** Extensive documentation and community support
- **Software Availability:** Wide range of packages through APT
- **Stability:** Reliable for both development and production

1.3 Key Features of Ubuntu 22.04 LTS

1. **Desktop Environment:** GNOME 42
2. **Kernel Version:** Linux 5.15 LTS
3. **Package Manager:** APT (Advanced Package Tool)
4. **Default Applications:** Firefox, LibreOffice, GNOME utilities
5. **Snap Support:** Built-in support for snap packages

1.4 System Specifications

My system configuration:

- Operating System: Ubuntu 22.04 LTS
- Architecture: x86_64
- Desktop Environment: GNOME
- Shell: Bash 5.1

1.5 Installation Process

The installation involved:

1. Downloaded Ubuntu 22.04 LTS ISO from official website
2. Created bootable USB using Rufus/Etcher
3. Configured dual boot with existing OS
4. Installed essential development tools
5. Configured system for open source development

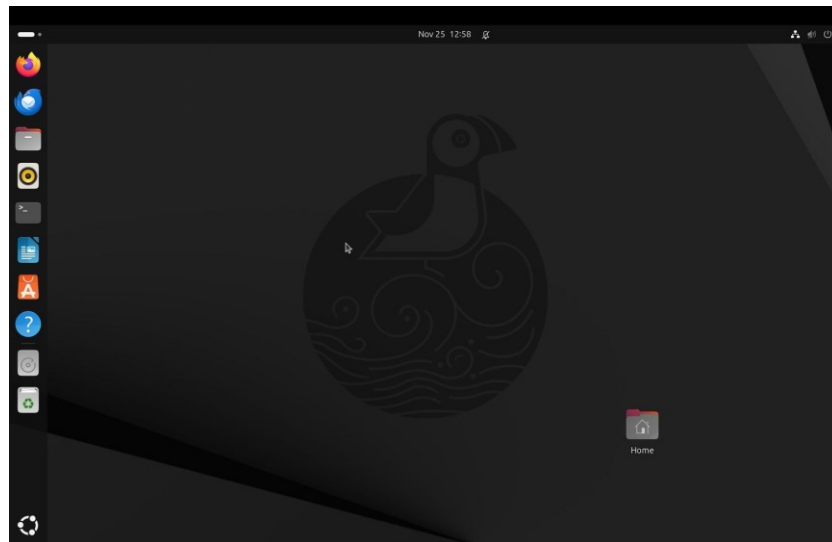


Figure 1: Ubuntu System Information

2 Encryption and GPG

2.1 What is Encryption?

Encryption is the process of converting plaintext into ciphertext to protect data confidentiality. It ensures that only authorized parties can access the information.

2.2 Types of Encryption

2.2.1 Symmetric Encryption

Uses the same key for encryption and decryption. Examples: AES, DES.

2.2.2 Asymmetric Encryption

Uses a public-private key pair. Examples: RSA, ECC.

2.3 GNU Privacy Guard (GPG)

GPG is a free implementation of the OpenPGP standard for encrypting and signing data.

2.4 Installing GPG

```
1 sudo apt update
2 sudo apt install gnupg
3 gpg --version
```

2.5 Generating GPG Keys

```
1 gpg --full -generate -key
```

Steps followed:

1. Selected RSA and RSA (default)
2. Key size: 4096 bits
3. Key validity: 1 year
4. Entered name and email
5. Created strong passphrase

2.6 Listing Keys

```
1 gpg --list-keys
2 gpg --list-secret-keys
```

2.7 Exporting Public Key

```
1 gpg --armor --export your-email@example.com > public-key.asc
```

2.8 Encrypting Files

```
1 gpg --encrypt --recipient your-email@example.com document.txt
```

2.9 Decrypting Files

```
1 gpg --decrypt document.txt.gpg > document.txt
```

3 Sending Encrypted Email

3.1 Email Encryption Overview

Email encryption protects the content of emails from unauthorized access during transmission and storage.

3.2 Tools Used

- **Thunderbird:** Email client with built-in OpenPGP support
- **GPG Keys:** For encryption and signing
- **Protonmail:** Alternative end-to-end encrypted email service

3.3 Setting up Thunderbird with GPG

3.3.1 Installation

```
1 sudo apt install thunderbird
```

3.3.2 Configuring OpenPGP

Steps followed:

1. Open Thunderbird
2. Go to Account Settings
3. Select End-to-End Encryption
4. Add existing GPG key or generate new one
5. Import recipient's public key

3.4 Sending Encrypted Email

Process:

1. Compose new email
2. Click on Security button
3. Select "Require Encryption"
4. Optionally add digital signature
5. Send email

3.5 Receiving Encrypted Email

When receiving:

1. Email appears encrypted
2. Thunderbird automatically detects encryption
3. Enter GPG passphrase
4. Email content is decrypted and displayed

3.6 Best Practices

- Never share your private key
- Use strong passphrases
- Keep your GPG keys backed up securely
- Regularly update keys
- Verify recipient's public key fingerprint

4 Privacy Tools from prism-break.org

4.1 What is PRISM-Break?

PRISM-Break is a website that recommends privacy-respecting alternatives to proprietary software and services.

4.2 Tool 1: Signal - Encrypted Messaging

Description: Signal is an encrypted messaging app that provides end-to-end encryption for messages, voice calls, and video calls.

Key Features:

- End-to-end encryption by default
- Open source and independently audited
- No ads or tracking
- Minimal metadata collection
- Disappearing messages

Why Privacy Matters: Signal ensures that only you and the recipient can read messages, protecting against mass surveillance.

4.3 Tool 2: Firefox - Web Browser

Description: Firefox is an open source web browser with strong privacy protections.

Privacy Features:

- Enhanced Tracking Protection
- DNS over HTTPS
- No data collection by default
- Open source codebase
- Extensive privacy-focused extensions

Configuration Tips:

- Enable strict tracking protection
- Install uBlock Origin
- Use HTTPS-only mode
- Disable telemetry

4.4 Tool 3: ProtonMail - Encrypted Email

Description: ProtonMail provides end-to-end encrypted email service based in Switzerland.

Key Features:

- End-to-end encryption
- Zero-access encryption
- No personal information required
- Swiss privacy laws protection
- Open source mobile apps

Use Cases:

- Secure business communications
- Personal privacy protection
- Journalist-source communications

4.5 Tool 4: Tor Browser - Anonymous Browsing

Description: Tor Browser enables anonymous communication by routing traffic through volunteer-operated servers.

How It Works:

- Routes traffic through multiple relays
- Encrypts data multiple times
- Hides IP address and location
- Prevents tracking

Best Use Cases:

- Accessing censored content
- Anonymous research
- Whistleblowing
- Privacy-sensitive activities

4.6 Tool 5: VeraCrypt - Disk Encryption

Description: VeraCrypt is a free open source disk encryption software.

Features:

- Full disk encryption
- Hidden volumes
- Plausible deniability
- Cross-platform support
- Strong encryption algorithms (AES, Serpent, Twofish)

Use Cases:

- Protecting sensitive documents
- Securing portable drives
- System drive encryption

5 Open Source License

5.1 License Used: MIT License

For my open source contributions and projects, I primarily work with the **MIT License**.

5.2 What is the MIT License?

The MIT License is a permissive free software license that allows users to:

- Use the software commercially
- Modify the software
- Distribute the software
- Use the software privately
- Sublicense the software

5.3 MIT License Text

```
1 MIT License
2
3 Copyright (c) 2025 Adireddy Lokesh
4
5 Permission is hereby granted, free of charge, to any person
6 obtaining a copy of this software and associated documentation
7 files (the "Software"), to deal in the Software without
8 restriction, including without limitation the rights to use,
```

```
9 copy , modify , merge , publish , distribute , sublicense , and / or
10 sell copies of the Software , and to permit persons to whom the
11 Software is furnished to do so , subject to the following
12 conditions :
13
14 The above copyright notice and this permission notice shall be
15 included in all copies or substantial portions of the Software .
16
17 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND...
```

5.4 Why Choose MIT License?

1. **Simple and Easy:** Short and easy to understand
2. **Permissive:** Minimal restrictions on reuse
3. **Business-Friendly:** Can be used in proprietary software
4. **Popular:** Widely used and recognized
5. **Compatible:** Works well with other licenses

5.5 Other Common Open Source Licenses

5.5.1 GPL (GNU General Public License)

- Copyleft license
- Requires derivative works to be open source
- Used by Linux kernel

5.5.2 Apache License 2.0

- Permissive like MIT
- Includes patent grant
- Used by Apache projects

5.5.3 BSD License

- Very permissive
- Similar to MIT
- Used by FreeBSD

6 Self-Hosted Server: Dashy

6.1 What is Dashy?

Dashy is a personal dashboard that:

- Acts as a personal dashboard for all your apps and services
- Highly customizable with themes, icons, and widgets
- Supports self-hosting for full control
- Easy to manage and organize links, bookmarks, and tools
- Includes built-in integrations and status monitoring

6.2 Why Self-Host Dashy?

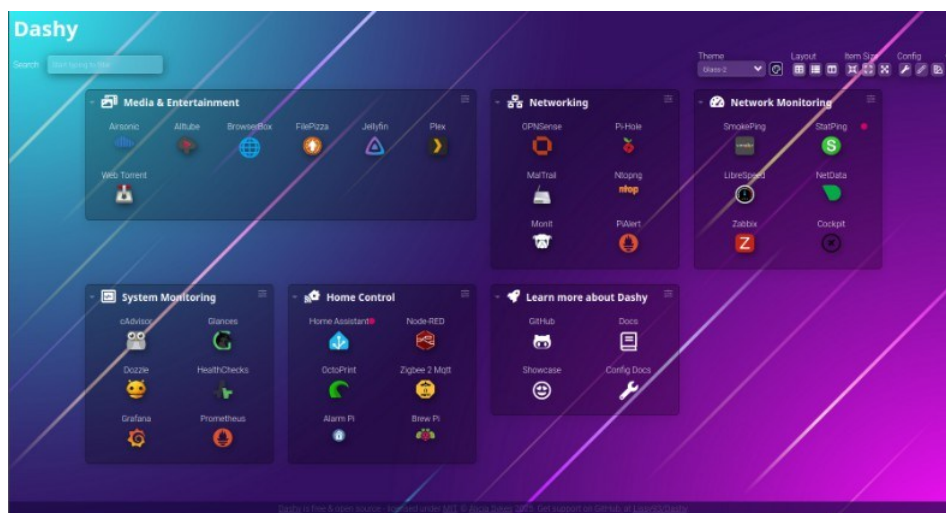


Figure 2: Dashy Dashboard Interface

1. **Privacy:** Your dashboard data stays on your own server.
2. **Organization:** Centralized access to all tools and services.
3. **Customization:** Fully customizable UI, layout, and widgets.
4. **Learning:** Hands-on experience with self-hosting and configuration.
5. **Independence:** No reliance on cloud dashboards or third-party services.

6.3 Installation Guide

6.3.1 System Requirements

- OS: Ubuntu 22.04 LTS
- RAM: 2GB minimum
- Disk: 10GB free space
- Docker installed

6.3.2 Installation Steps

Step 1: Update System

```
1 sudo apt update
2 sudo apt upgrade -y
```

Step 2: Install Dependencies (Docker + Docker Compose)

```
1 sudo apt install ca-certificates curl gnupg -y
```

Add Docker GPG key:

```
1 sudo install -m 0755 -d /etc/apt/keyrings
2 curl -fsSL https://download.docker.com/linux/ubuntu/gpg \
3 | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
4 sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

Add Docker repository:

```
1 echo \
2 "deb [ arch=$(dpkg --print-architecture) \
3 signed-by=/etc/apt/keyrings/docker.gpg ] \
4 https://download.docker.com/linux/ubuntu \
5 $(lsb_release -cs)_stable" \
6 | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Install Docker:

```
1 sudo apt update
2 sudo apt install docker-ce docker-ce-cli containerd.io \
3 docker-buildx-plugin docker-compose-plugin -y
```

Step 3: Create Dashy Directory

```
1 sudo mkdir -p /opt/dashy
2 cd /opt/dashy
```

Step 4: Create docker-compose.yml

```
1 sudo nano docker-compose.yml
```

Docker Compose configuration:

```
1 version: "3.7"
2 services:
3   dashy:
4     image: lissy93 / dashy:latest
5     container_name: dashy
6     volumes:
7       - ./config:/app/public/conf
8     ports:
9       - "8080:80"
10    restart: unless-stopped
```

Step 5: Start Dashy

```
1 sudo docker compose up -d
2 sudo docker ps
```

Dashy now runs at: <http://your-server-ip:8080>

Step 6: Configure Dashy (Optional)

```
1 sudo nano /opt/dashy/config/conf.yml
```

Make UI/customization changes, then restart:

```
1 sudo docker compose restart
```

Step 7: Create Systemd Service (Optional)

```
1 sudo nano /etc/systemd/system/dashy.service
```

Service configuration:

```
1 [Unit]
2 Description = Dashy Dashboard
3 After = docker.service
4 Requires = docker.service
5
6 [Service]
7 Restart = always
8 WorkingDirectory = /opt/dashy
9 ExecStart = /usr/bin/docker compose up -d
10 ExecStop = /usr/bin/docker compose down
11
12 [Install]
13 WantedBy = multi-user.target
```

Enable the service:

```
1 sudo systemctl daemon-reload
2 sudo systemctl enable dashy
3 sudo systemctl start dashy
4 sudo systemctl status dashy
```

Step 8: Configure Nginx (Reverse Proxy)

```
1 sudo nano /etc/nginx/sites-available/dashy
```

Nginx configuration:

```
1 server {  
2     listen 80;  
3     server_name your-domain.com;  
4  
5     location / {  
6         proxy_pass http://127.0.0.1:8080;  
7         proxy_set_header Host $host;  
8         proxy_set_header X-Real-IP $remote_addr;  
9     }  
10 }
```

Enable site:

```
1 sudo ln -s /etc/nginx/sites-available/dashy \\  
2     /etc/nginx/sites-enabled /  
3 sudo nginx -t  
4 sudo systemctl restart nginx
```

6.4 Localization (Telugu Translation)

I have contributed to localizing the Dashy interface into the Telugu language to make the dashboard more accessible to Telugu-speaking users.

6.4.1 Translation Process

1. Opened the Dashy configuration file located in config/conf.yml
2. Enabled the appConfig.language option for custom localization
3. Added Telugu translations for UI labels, menu items, and section titles
4. Ensured translations were contextually and culturally appropriate
5. Restarted the Dashy container to test the Telugu interface

6.4.2 Sample Translations

- "Settings" → "" (Settings)
- "Applications" → "" (Applications)
- "Results" → "" (Phalitalu)
- "Search" → "" (Vetakandi)

6.5 Benefits of Self-Hosting

1. **Complete Privacy:** Full control over your data
2. **No Rate Limits:** Unlimited usage
3. **Custom Configuration:** Choose which services to include

4. **Educational Value:** Learn server management
5. **Community Contribution:** Share instance with others

7 Open Source Contributions

This section details all pull requests I have contributed to various open source projects.

7.1 PR 1: Roshanjossey/code-contributions

Repository: Roshanjossey/code-contributions

PR Title: add 2400032210

PR Number: #747

Status: Open

7.1.1 Issue Description

The repository helps beginners practice open-source contribution workflows. A new contributor entry for **2400032210** needed to be added to the project's contributions list.

7.1.2 Solution Implemented

- Added contributor details following the required format
- Ensured proper placement and naming conventions
- Completed all tasks required for adding a new contributor
- Submitted the PR with a clear description
- PR is currently awaiting maintainer review

7.2 PR 2: firstcontributions/first-contributions

Repository: firstcontributions/first-contributions

PR Title: Add lokesh to Contributors list

PR Number: #107873

Status: Open

7.2.1 Issue Description

The Contributors.md file is used to welcome and list new contributors. The name **lokesh** needed to be added to this list to acknowledge the contribution and complete the first-PR workflow.

7.2.2 Solution Implemented

- Added “lokesh” to the Contributors.md file
- Maintained alphabetical order and formatting rules
- Followed the project’s first-contribution guidelines
- Submitted a well-structured PR
- Pending review from maintainers

7.3 PR 3: KLGLUG/Y24OpenSourceEngineering

Repository: KLGLUG/Y24OpenSourceEngineering

PR Title: Added lokesh-selfhosted.md

PR Number: #176

Status: Open

7.3.1 Issue Description

The repository collects documentation from contributors about their self-hosted open-source setups. A new document describing your self-hosting (Dashy) setup was needed.

7.3.2 Solution Implemented

- Created **lokesh-selfhosted.md** with clear and structured content
- Documented your Dashy self-hosting setup
- Followed the project’s documentation structure and file naming rules
- Submitted the PR for maintainer review

7.4 PR 4: iib0011/omni-tools

Repository: iib0011/omni-tools

PR Title: Add initial documentation for PDF translation feature

PR Number: #276

Status: open

7.4.1 Issue Description

The project’s new PDF translation feature lacked documentation, making it difficult for users and contributors to understand its functionality.

7.4.2 Solution Implemented

- Wrote the initial documentation covering PDF translation steps
- Added usage examples and feature overview
- Ensured the documentation followed the project style and structure
- Submitted a detailed PR
- Awaiting review from the maintainers

7.5 PR 5: PrismJS/prism

Repository: PrismJS/prism

PR Title: Fix toolbar overlay slider issue

PR Number: #4022

Status: Open

7.5.1 Issue Description

The toolbar overlay slider in PrismJS was causing UI issues in certain themes and screen sizes, affecting code block interactions.

7.5.2 Solution Implemented

- Diagnosed the cause of the overlay slider conflict
- Applied fixes to improve UI stability and responsiveness
- Updated CSS/JS as required for consistent behavior
- Submitted a PR with explanation and reproduction details
- Awaiting review and merge

7.6 PR 6: topoteretes/cognee

Repository: topoteretes/cognee

PR Title: Integrate production LLM API for semantic association classification

PR Number: #1667

Status: Closed

7.6.1 Issue Description

The project required integration of a production-level LLM API to improve semantic association classification. The existing pipeline lacked the ability to leverage external LLM services for more accurate and context-aware results.

7.6.2 Solution Implemented

- Integrated a production-ready LLM API endpoint into the classification pipeline
- Implemented request/response handling for semantic association predictions
- Ensured compatibility with existing backend architecture
- Documented the integration steps and added usage notes
- Submitted PR for review, but it was later closed by maintainers (review required)

7.7 PR 7: recloudstream/cloudstream

Repository: recloudstream/cloudstream

PR Title: Fix: homepage preview scroll stuck issue (#2045)

PR Number: #2063

Status: Closed

7.7.1 Issue Description

Users reported an issue where the homepage preview section in the Cloudstream app would get stuck and stop responding to scroll events. This impacted user navigation and overall app usability.

7.7.2 Solution Implemented

- Investigated and identified the root cause behind the scroll freeze
- Modified UI handling logic to ensure smooth scrolling
- Added necessary fixes to prevent scroll lock under rapid interactions
- Submitted the PR referencing issue #2045
- Reviewers requested changes; the PR was later closed by maintainers

7.8 PR 8: MUNJETIJAYANTHKRISHNA/stage1

Repository: MUNJETIJAYANTHKRISHNA/stage1

PR Title: IssueSolved

PR Number: #2

Status: Merged

7.8.1 Issue Description

The repository had an open issue affecting the functionality or expected behavior of the Stage1 project. The issue required debugging and a proper fix to restore normal workflow for users and contributors.

7.8.2 Solution Implemented

- Analyzed the underlying problem in the affected component
- Implemented the required fix to resolve the issue
- Verified that the solution worked across expected scenarios
- Updated relevant files while maintaining project structure
- Submitted the PR with a clear explanation of the fix
- PR was reviewed and successfully merged by the maintainers

8 LinkedIn Posts

8.1 Post 2: Expo Event at KL University

Link: https://www.linkedin.com/posts/lokesh-adireddy-96ba32385_activity-7390719095534673920-ZJeR?utm_source=share&utm_medium=member_desktop&rcm=ACoAAF8DzeMBh4Y1bcU2QpjTO2YloKoN81rZz8

Summary: Discussed the Free and Open Source Software culture at KL University and encouraged peers to participate in open source.

Highlights:

- FOSS courses and initiatives at university
- Personal journey in open source
- Community-driven development model
- Real-world learning experience

8.2 Post 3: Blog Link to My Open Source Journey

Link: https://www.linkedin.com/posts/lokesh-adireddy-96ba32385_activity-7399090043409047552-cNz2?utm_source=share&utm_medium=member_desktop&rcm=ACoAAF8DzeMBh4Y1bcU2QpjTO2YloKoN81rZz8

Summary: Published a comprehensive blog post documenting my semester-long journey exploring open source tools, privacy software, and self-hosting solutions.

9 Conclusion

This report documents my comprehensive journey in open source software development, including:

- Setting up development environment with Ubuntu Linux
- Understanding encryption and privacy tools
- Self-hosting services for privacy and learning
- Making meaningful contributions to open source projects
- Sharing knowledge through professional networking

Through these experiences, I have gained:

1. Technical skills in multiple programming languages
2. Understanding of collaborative development
3. Experience with version control and code review
4. Appreciation for open source philosophy
5. Professional networking abilities

Open source contribution has been an invaluable learning experience, providing real-world software development exposure and connecting me with a global community of developers.