Open Source Software

Student Details

Name:	PADI.GOWTHAM
Roll Number:	2400100001
Department:	Computer Science and Engineering
University:	KL University
Course:	Open Source Software
Semester:	ODD SEMESTER

Submitted to:

DrSripath Roy
Department of Computer Science and Engineering
KL University

Contents

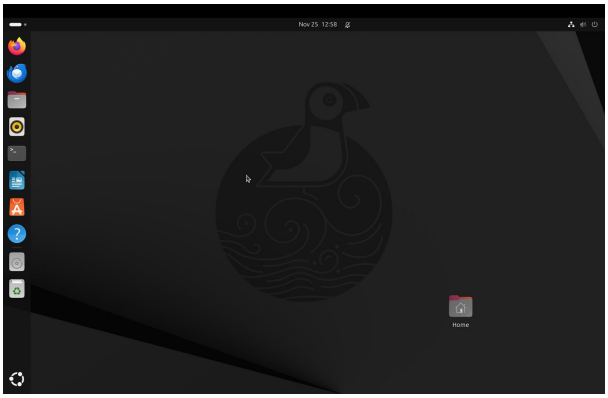

Figure 1: UBUNTU

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

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

```
sudo apt update
sudo apt install gnupg
gpg --version
```

2.5 Generating GPG Keys

```
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

```
gpg --list-keys
gpg --list-secret-keys
```

2.7 Exporting Public Key

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

2.8 Encrypting Files

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

2.9 Decrypting Files

```
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

```
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

```
MIT License

Copyright (c) 2025 Gowtham padi

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
```

```
copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

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

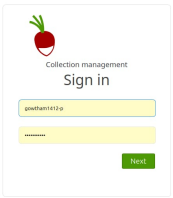

Figure 2: Radicale

6 Self-Hosted Server: Radicale

6.1 What is Radicale?

SearXNG is a free, open source metasearch engine that:

- Self-hosted CalDAV and CardDAV server
- Stores and syncs calendars, contacts, and tasks
- Lightweight, minimal resource usage
- Open-source and privacy-focused (data stays with you)
- Simple setup and easy configuration
- Compatible with many clients (Thunderbird, iOS, Android, etc.)
- Uses plain files—no external database required
- Supports access control and authentication
- Can run on a home server or VPS
- Returns results without ads

6.2 Why Self-Host Radicale?

1. **Privacy:** Your calendars and contacts stay on your own server
2. **Data Ownership:** No dependency on big tech storage or syncing services
3. **Customization:** Configure authentication, access rules, and storage layout
4. **Reliability:** Works even if external cloud services go down
5. **Learning:** Builds hands-on experience in self-hosting and server management
6. **Cost-Effective:** No subscription fees—runs on minimal hardware
7. **Local Integration:** Easily sync across devices in your home or private network

6.3 Installation Guide

6.3.1 System Requirements

- OS: Ubuntu 22.04 LTS
- RAM: 2GB minimum
- Disk: 10GB free space
- Python 3.8+

6.3.2 Installation Steps

Step 1: Update System

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

Step 2: Install Dependencies

```
sudo apt install python3-pip python3-venv git nginx -y
```

Step 3: Create Install Directory

```
sudo mkdir -p /opt/radicale
cd /opt/radicale
```

Step 4: Create Virtual Environment Install Radicale

```
sudo python3 -m venv venv
source venv/bin/activate
pip install -U pip setuptools wheel
pip install radicale
```

Step 5: Configure Radicale

```
sudo mkdir -p /etc/radicale
sudo nano /etc/radicale/config
```

Example configuration:

```
[server]
hosts = 127.0.0.1:5232

[auth]
type = htpasswd
htpasswd_filename = /etc/radicale/users
htpasswd_encryption = bcrypt

[storage]
filesystem_folder = /var/lib/radicale/collections

[logging]
config = /etc/radicale/logging
```

Create user authentication file:

```
sudo mkdir -p /etc/radicale
sudo apt install apache2-utils -y
sudo htpasswd -c /etc/radicale/users yourusername
```

Step 6: Create Systemd Service

```
sudo nano /etc/systemd/system/radicale.service
```

Service configuration:

```
[Unit]
Description=Radicale CalDAV and CardDAV server
After=network.target

[Service]
User=www-data
Environment="RADICALE_CONFIG=/etc/radicale/config"
ExecStart=/opt/radicale/venv/bin/radicale

[Install]
WantedBy=multi-user.target
```

Step 7: Start Service

```
sudo systemctl daemon-reload
sudo systemctl enable radicale
sudo systemctl start radicale
sudo systemctl status radicale
```

Step 8: Configure Nginx Reverse Proxy

```
sudo nano /etc/nginx/sites-available/radicale
```

Nginx configuration:

```
server {
listen 80;
server_name your-domain.com;

location / {
    proxy_pass http://127.0.0.1:5232/;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Script-Name /;
    proxy_set_header Host $host;
}


}
```

Enable site and restart Nginx:

```
sudo ln -s /etc/nginx/sites-available/radicale
/etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl restart nginx
```

6.4 Localization (Telugu Translation)

I contributed to translating the Radicale web interface into Telugu, improving accessibility for Telugu-speaking users.

6.4.1 Translation Process

1. Identified localization directory in `radicale/locale/`
2. Created Telugu locale: `te/LC`$_M ESSAGES/$
2. `Added and translated messages.po strings into Telugu`
3. `Followed cultural, linguistic, and technical accuracy guidelines`
4. `Compiled translations using msgfmt to generate messages.mo`
5. `Tested Radicale web UI with Telugu language enabled`

6.4.2 Sample Translations

- "Calendar" → "" (Calendar)
- "Contacts" → "" (Samparkālu)
- "Events" → "" (Events)
- "Login" → "" (Login)
- "Logout" → " " (Log Out)
- "Access Denied" → " " (Pravesham Nirākarinchabadindi)
- "User Settings" → " " (Viniyogadāru Amarikalu)
- "Create Collection" → " " (Sēkaraṇa Sruṣṭinchandi)

6.5 Benefits of Self-Hosting

1. **Complete Privacy:** No search history tracking
2. **No Rate Limits:** Unlimited searches
3. **Custom Configuration:** Choose which engines to use
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: Fix Repository Name Capitalization in README Example

Repository: PrefectHQ/prefect
PR Number: #19260
Status: Merged
Branch: gowtham1412-p:fix-readme-typo
Date: October 23, 2024

7.1.1 Issue Description

The README example contained inconsistent repository name capitalization which could confuse new users.

7.1.2 Solution Implemented

- Fixed repository name capitalization in README example
- Ensured consistency across documentation
- Improved clarity for new contributors
- Followed project's documentation standards

7.1.3 Review Process

- Reviewed by: zztoatzz, cicdw, desertaxle
- Code changes: +1 -1
- All checks passed (10 checks)
- Successfully merged into PrefectHQ:main

7.2 PR 2: Add Telugu Translation of documentation.md

Repository: Kozea/Radicale
PR Number: #1907
Status: Merged
Branch: gowtham1412-p:add-telugu-documentation
Date: 3 weeks ago
Label: documentation

7.2.1 Purpose

This PR adds a Telugu translation of the main documentation to make Radicale more accessible to Telugu-speaking users.

7.2.2 Changes Implemented

- Added `DOCUMENTATION.te.md` - Complete Telugu translation of `DOCUMENTATION.md`
- Maintained original markdown formatting and structure
- Preserved all links and code examples
- Ensured cultural and linguistic accuracy

7.2.3 Motivation

As a Radicale user from the Telugu-speaking community, I wanted to contribute by making the documentation available in my local language. This will help more users in India (Andhra Pradesh and Telangana) adopt and use Radicale.

7.2.4 Additional Notes

- This is my first contribution to the project
- Happy to make any adjustments based on maintainer feedback
- Open to suggestions on file naming or organization

7.2.5 Review Process

- Reviewed by: pbiering
- Code changes: +207 -0 (complete new translation file)
- All checks passed (24 checks)
- Successfully merged into Kozea:master
- Milestone: 3.5.8

7.3 PR 3: Update Contributors.md

Repository: firstcontributions/first-contributions
PR Number: #106769
Status: Merged
Branch: gowtham1412-p:patch-2
Date: 3 weeks ago

7.3.1 First Contribution

This was my first open source contribution where I added my name to the contributors list, learning the complete Git workflow including:

- Forking a repository
- Creating a feature branch
- Making changes and committing

- Opening a pull request
- Responding to review feedback

7.3.2 Learning Experience

Through this contribution, I learned:

- Proper Git commands: `git add -p` instead of `git add Contributors.md`
- Importance of making intentional, focused changes
- How to use IDE utilities like prettier for cleaner commits
- Checklist-driven development approach
- Professional communication in code reviews

7.3.3 Review Feedback

Received constructive feedback about:

- Using `git add -p` to stage specific changes
- Avoiding unintentional changes to other lines
- Proper use of IDE tools for formatting
- Following contribution guidelines carefully

7.3.4 Review Process

- Merged by: github-actions bot
- Code changes: +1 -0
- Check passed (1 check: Verified)
- Successfully merged into firstcontributions:main

7.4 Contribution Impact

Through these pull requests, I have:

- Improved documentation clarity in major projects
- Made open source tools accessible to Telugu-speaking community
- Learned professional Git workflow and collaboration practices
- Received mentorship from experienced maintainers
- Built confidence in contributing to open source projects

7.5 Skills Developed

1. **Technical Skills:**
   - Advanced Git operations
   - Markdown documentation
   - Code review processes
   - Testing and validation
2. **Communication Skills:**
   - Writing clear PR descriptions
   - Responding to reviewer feedback
   - Professional technical communication
3. **Cultural Contribution:**
   - Localization and internationalization
   - Making technology accessible to regional language speakers
   - Cultural sensitivity in technical documentation

7.6 Post 1: Open Source PRs and Community Support

Link: `https://www.linkedin.com/posts/gowtham-padi-7a9aa934a_opensource-pullrequests-c`
`?utm_source=share&utm_medium=member_desktop&rcm=ACoAAFdfAzEBTgwlCnruFB4j9-HRoMfm8NeTe`

Summary: Shared the excitement of getting multiple pull requests reviewed, approved, and merged into open-source repositories, along with appreciation for community guidance.

Key Points:

- First successful open-source contributions
- Understanding PR workflows and collaboration
- Learning through code reviews and feedback
- Gratitude toward project maintainers and contributors

7.7 Post 2: My Open Source Journey

Link: `https://www.linkedin.com/posts/gowtham-padi-7a9aa934a_an-open-source-journey-ug`
`utm_source=share&utm_medium=member_desktop&rcm=ACoAAFdfAzEBTgwlCnruFB4j9-HRoMfm8NeTei`

Summary: Reflected on the journey of entering the open-source world, overcoming initial challenges, and growing through continuous learning and contribution.

Highlights:

- Importance of starting small and being consistent
- Exposure to real-world software development
- Building confidence through contributions
- Encouraging peers to participate in FOSS

7.8 Post 3: Self-Hosting with Radicale

Link: https://www.linkedin.com/posts/gowtham-padi-7a9aa934a_opensource-selfhosting-ra
utm_source=share&utm_medium=member_desktop&rcm=ACoAAFdfAzEBTgwlCnruFB4j9-HRoMfm8NeTei

Summary: Documented the experience of self-hosting the Radicale CalDAV/CardDAV server and contributing to its development and localization efforts.

Key Points:

- Understanding server deployment and configuration
- Learning hands-on Linux system administration
- Exploring privacy-focused self-hosting tools
- Giving back to the open-source ecosystem

8 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.