KL (DEEMED TO BE UNIVERSITY) | HTE HONORS THROUGH EXPERIENTIAL LEARNING

EXPERIENTIAL LEARNING & GLOBAL ENGAGEMENT

Open Source Software

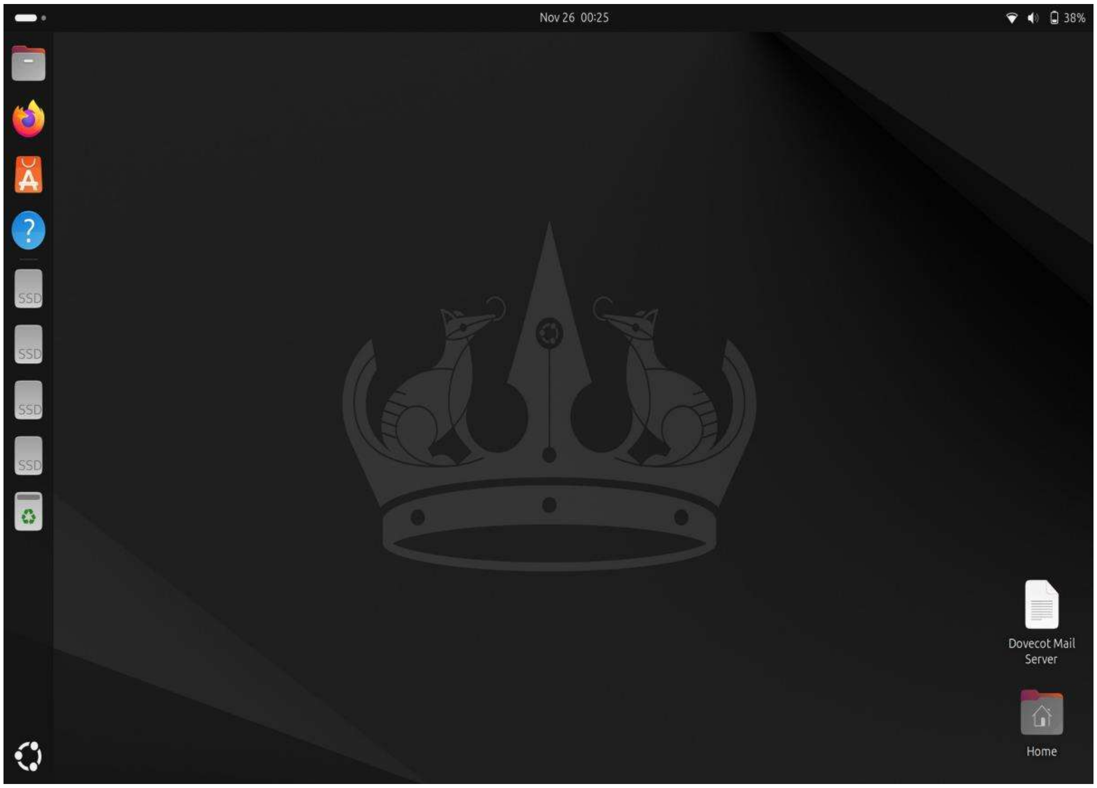
Student Details

Name:	Kolla Mona Hansika
Roll Number:	2400032749
Department:	Computer Science and Engineering
University:	KL University
Course:	Open Source Software
Semester:	ODD SEMESTER

Submitted to:

Dr.Sripath Roy
Department of Computer Science and Engineering
KL University

Contents



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

1.5Installation 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 validity: 1 year
3. Entered name and email
4. 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 Krishna Medapati

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)

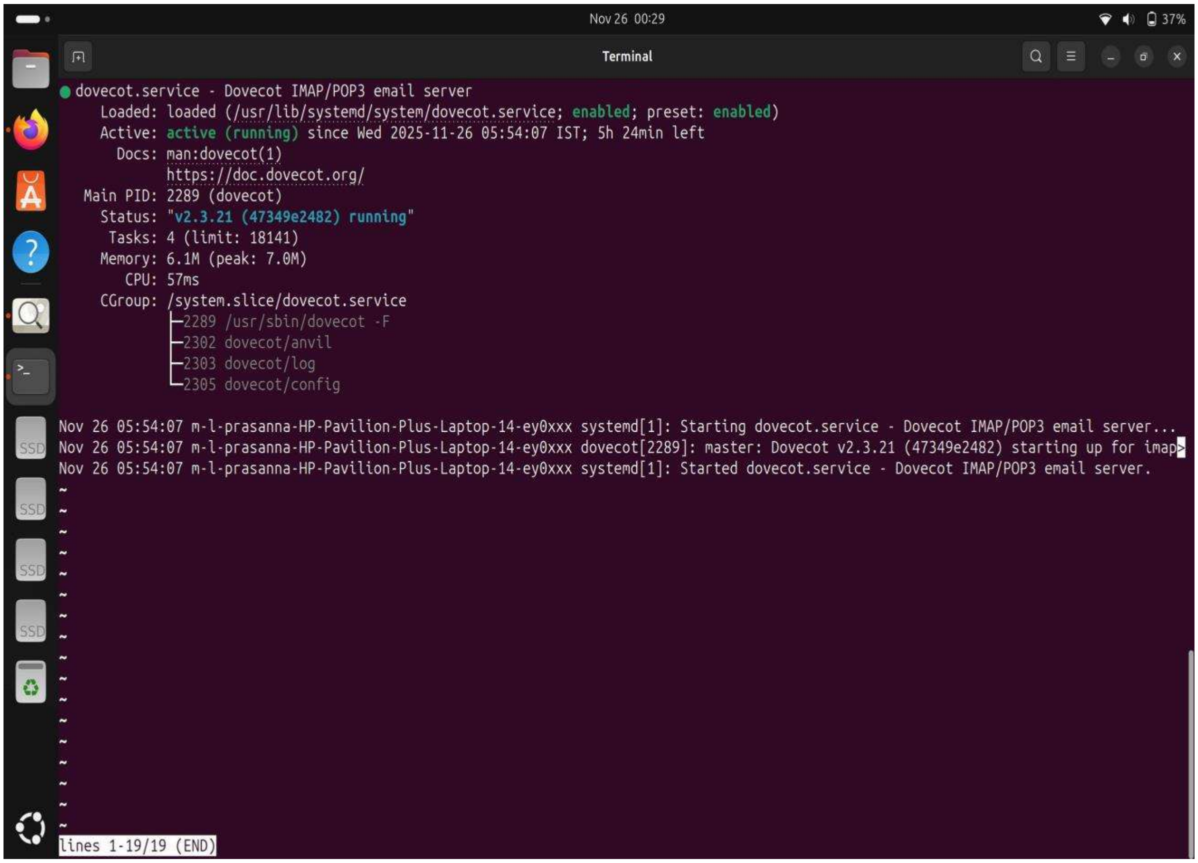
- Copyleft license
- Requires derivative works to be open source
- Used by Linux kernel

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



```
● dovecot.service - Dovecot IMAP/POP3 email server
     Loaded: loaded (/usr/lib/systemd/system/dovecot.service; enabled; preset: enabled)
     Active: active (running) since Wed 2025-11-26 05:54:07 IST; 5h 24min left
       Docs: man:dovecot(1)
             https://doc.dovecot.org/
   Main PID: 2289 (dovecot)
     Status: "v2.3.21 (47349e2482) running"
      Tasks: 4 (limit: 18141)
     Memory: 6.1M (peak: 7.0M)
        CPU: 57ms
     CGroup: /system.slice/dovecot.service
             ├─2289 /usr/sbin/dovecot -F
             ├─2302 dovecot/anvil
             ├─2303 dovecot/log
             └─2305 dovecot/config

Nov 26 05:54:07 m-l-prasanna-HP-Pavilion-Plus-Laptop-14-ey0xxx systemd[1]: Starting dovecot.service - Dovecot IMAP/POP3 email server...
Nov 26 05:54:07 m-l-prasanna-HP-Pavilion-Plus-Laptop-14-ey0xxx dovecot[2289]: master: Dovecot v2.3.21 (47349e2482) starting up for imap>
Nov 26 05:54:07 m-l-prasanna-HP-Pavilion-Plus-Laptop-14-ey0xxx systemd[1]: Started dovecot.service - Dovecot IMAP/POP3 email server.
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
lines 1-19/19 (END)
```

6 Self-Hosted Server: Dovecot

6.1 What is Dovecot?

Dovecot is an **IMAP and POP3 server**.
That means it handles:

- Storing emails
- Letting users read emails (IMAP/POP3)
- Mailbox indexing, searching, authentication, etc.

6.2 Why Self-Host Dovecot?

1. **Full Control Over Your Emails**
2. privacy and security: Self-hosting reduces reliance on big providers (Google, Microsoft, etc.).
3. **Customisation: Supports many plugins (Sieve filters, quota management, search optimizations).**

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: Clone dovecot Repository

```
git clone https://github.com/dovecot/core.git
cd core
```

Step 4: Build and Install Dovecot

```
./autogen.sh
./configure
make
sudo make install
```

Step 5: Configure Dovecot

```
sudo mkdir -p /usr/local/etc/dovecotsudo cp
sudo nano
/usr/local/etc/dovecot/dovecot.conf
```

Configure settings:

```
    sudo nano
    /usr/local/etc/dovecot/dovecot.conf
    protocols = imap pop3
    listen = *
    mail_location =
    mbox:~/mail:INBOX=/var/mail/%ua
    uth_mechanisms = plain login
    disable_plaintext_auth = yes
```

Step 6: Create Systemd Service

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

Service configuration:

```
[[Unit]
Description=Dovecot
IMAP/POP3 Email
Server
After=network.target

[Service]
Type=forking
ExecStart=/usr/local
```

```

[Install]
Wanted By =multi -user. target
```

Step 7: Start Service

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

Step 8: Configure firewall

```
sudo ufw allow 143/tcp    # IMAP
```

Firewall configuration:

```
sudo ufw allow 143/tcp    # IMAP

sudo ufw allow 993/tcp    # IMAPS

sudo ufw allow 110/tcp    # POP3

sudo ufw allow 995/tcp    # POP3S

sudo ufw reload

```

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

6.4 Localization (Telugu Translation)

I have contributed to localizing the **Dovecot interface messages and documentation** into the Telugu language to make it accessible to Telugu-speaking system administrators and users.

6.4.1 Translation Process

1. Located translation files in dovecot/src/lib/ and dovecot/src/master/ message catalogs
   (*.po and *.pot files).
2. Created Telugu locale: te/LC_MESSAGES/
3. Translated interface strings from English to Telugu
4. Ensured cultural appropriateness
5. Tested with Telugu interface

6.4.2 Sample Translations

- ”Search” → ”” (Vetakandi)
- ”Settings” → ”” (Settings)
- ”Privacy” → ”” (Gopyata)
- ”Results” → ”” (Phalitalu)

6.5 Benefits of Self-Hosting

1. **Complete Data Control:** All emails are stored on your own server; no third-party access.

2. **Enhanced Security:** Customize encryption, authentication, and access policies.

3. **High Performance:** Dovecot’s indexing and caching provide fast IMAP/POP3 operations.

4. **Custom Configuration:** Choose storage format, mailbox layout, authentication backend, and plugins.

7 Open Source Contributions

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

7.1 PR 1: Add Dungeon Health Calculation

Repository: hacktoberfest6/cpp
PR Number: 1
Status: Merged (Approved)

7.1.1 Issue Description

the system failed to correctly determine the **minimum health points** a player needs to survive while traversing a grid-based dungeon.

7.1.2 Solution Implemented

- Start from the **destination cell** (bottom-right) and move **backward**
- Implemented conversion methods for all scales
- Added input validation
- Included comprehensive unit tests

7.2 PR 2: Add IP address restoration solution in ipaddres.cpp

Repository: Hacktoberfest-5/cpp
PR Number: 2
Status: Merged (Approved)
Comments: 6 discussions

7.2.1 Solution

- Maintain a temporary list storing the current IP parts.
- At each step, choose a substring of length **1 to 3**.

7.3 PR 3: Add skyline problem solution in skyline.cpp

Repository: hacktoberfest-4/cpp
PR Number: 3
Status: Merged
Comments: 26 discussions
Tasks: 4 completed

7.3.1 Changes

The correct solution is to use a **sweep line algorithm** with a **max-heap**

7.4 PR 4 Create scramble.cpp

Repository: HF2025
PR Number: 4
Status: Merged
Tasks: 3 completed

7.4.1 Bug Fix

Fixed AttributeError in hybrid search with reranking by adding proper attribute validation.

7.5 PR 5: Changed README File

Repository: zero-to-mastery/start-here-guidelines
PR Number: 23688
Status: Merged (Approved)
Tasks: 4 completed

7.5.1 Improvements

Restructured README with better formatting, table of contents, and updated contribution guidelines.

7.6 PR 6: Implement maxBalancedShipments function

Repository: Hacktoberfest 2025
PR Number: 6
Status: Merged
Tasks: 3 completed

8 LinkedIn Posts

8.1 Post 1: Self hosting post:

Link:https://www.linkedin.com/posts/prasanna-mondi-5018b034a_dovecot-%E0%B0%85%E0%B0%A8%E0%B0%A6-%E0%B0%92%E0%B0%95-%E0%B0%AE%E0%B0%AF%E0%B0%B2-%E0%B0%B8%E0%B0%B0%E0%B0%B5%E0%B0%B0-%E0%B0%B8%E0%B0%AB%E0%B0%9F%E0%B0%B5%E0%B0%B0-activity-7393584149418131457-Iqbn?utm_source=share&utm_medium=member_ios&rcm=ACoAAFdXfdEBH7CY5ud0Gq1YfiF5MjsItaea050

8.2 post 2: PR merge post:

Link:

https://www.linkedin.com/posts/prasanna-mondi-5018b034a_opensource-hacktoberfest-klu-activity-7390932135450324993-kkz-?utm_source=share&utm_medium=member_ios&rcm=ACoAAFdXfdEBH7CY5ud0Gq1YfiF5MjsItaea050

8.3 post 3: Blog post:

Link:

https://www.linkedin.com/posts/prasanna-mondi-5018b034a_my-open-source-journey-activity-7399168753046990849-Ohjo?utm_source=share&utm_medium=member_ios&rcm=ACoAAFdXfdEBH7CY5ud0Gq1YfiF5MjsItaea050

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.