OPEN SOURCE ENGINEERING

,

Student ID: 2400031705

Name: PEDDINENI VENKATA SUSMITHA

KL University

1 Understanding the Core Ubuntu Linux Distribution

1.0.1 1. Overview and Philosophy

Ubuntu is a powerful, free, and open-source operating system built upon the stable foundation of Debian Linux. It stands as the world's most popular Linux distribution for desktop use, successfully blending cutting-edge features with unparalleled user-friendliness. Developed and maintained by Canonical Ltd., Ubuntu's guiding principle is "Linux for human beings."

1.0.2 2. The Desktop Experience (GNOME)

The standard Ubuntu desktop utilizes the **GNOME** desktop environment, which presents a modern, clean, and highly efficient graphical interface. The key design elements include a permanent dock (launcher) on the left side for quick access to essential applications, and the **Activities Overview**. This view, easily accessed by pressing the Super (Windows) key, provides a centralized hub for managing all open windows, workspaces, and system-wide searching.

1.0.3 3. Software Management and Packaging

Ubuntu employs a robust dual-system for software management. The traditional and reliable **Advanced Packaging Tool (APT)** manages **DEB** packages, handling core system utilities and standard applications sourced from official repositories. Complementing this is the use of **Snaps**, a modern, containerized package format pioneered by Canonical. Snaps bundle an application with all its required dependencies, guaranteeing consistent performance across different Ubuntu versions.

2 Encryption and GPG

2.1 Definition

Encryption is a security technique used to convert readable data into an unreadable or scrambled format. This protects information from unauthorized access. Only a person who has the correct **key** can convert the encrypted data back to its original, readable form. Encryption plays a major role in ensuring:

- **Privacy** – keeping information confidential
- **Security** – protecting sensitive data from hackers
- **Integrity** – preventing unauthorized modifications
- **Safe communication** – especially over the internet

Encryption is widely used in email communication, banking systems, cloud storage, messaging apps, and secure file sharing.

2.2 GPG (GNU Privacy Guard) Explained

GPG is the GNU implementation of the **OpenPGP** standard (originally Pretty Good Privacy - PGP). It is essential for protecting individual files and ensuring secure, authenticated communication.

2.2.1 Core GPG Concepts

GPG relies on **asymmetric cryptography**, which uses a pair of mathematically linked keys:

- **Public Key:** This key is shared with everyone. It can be used to **encrypt** a message that only you can read, or to **verify** a signature you created.
- **Private (Secret) Key:** This key is kept **secret** and is protected by a strong passphrase. It is used to **decrypt** messages sent to you, or to **digitally sign** files to prove they came from you.

2.2.2 Basic GPG Command-Line Usage

GPG is usually pre-installed on Ubuntu and is primarily used through the command line (Terminal).

A. Generating a Key Pair The first step is to create your public and private key pair:
Bash

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

You will be prompted to select the key type (RSA and RSA is common), keysize (4096 is recommended), expiration date, and your Real Name, Email, and a strong **passphrase** to protect your private key.

B. Encrypting a File for Yourself (Symmetric Encryption) To quickly encrypt a file using a single passphrase (like a standard password), use symmetric encryption:
Bash

```
gpg -c myfile.txt
```

This command will prompt you for a passphrase and create an encrypted file named `myfile.txt.gpg`.

C. Encrypting a File for Someone Else (Asymmetric Encryption) To securely send a file, you must use the recipient's **Public Key** (which you must have previously imported into your keyring with `gpg --import`):
Bash

```
gpg --encrypt --recipient "recipient@example.com" mysecretfile.doc
```

This creates `mysecretfile.doc.gpg`. Only the recipient, who holds the corresponding Private Key, can decrypt it.

D. Decrypting a File To decrypt a file that was encrypted for you:
Bash

```
gpg --decrypt mysecretfile.doc.gpg
```

You will be prompted for the passphrase that protects your Private Key. You can use the `--output` option to specify the decrypted

3 Sending Encrypted Email

Sending encrypted email is a secure method of communication that ensures only the intended recipient can read the message. Ubuntu commonly uses **GPG (GNU Privacy Guard)** for this purpose, which relies on public-key cryptography. The sender encrypts the email using the recipient's **public key**, and the recipient decrypts it using their **private key**.

The process involves several steps:

- **Public Key Exchange:** Both users generate a GPG key pair and share their public keys with each other. Public keys are safe to share and can even be uploaded to key servers.
- **Importing the Recipient's Key:** Before sending an encrypted email, the sender must import the recipient's public key into their keyring.
- **Encrypting the Message:** Once the key is imported, the email content or attached file is encrypted using GPG. The encrypted text appears unreadable to anyone without the corresponding private key.
- **Sending the Email:** The encrypted message can be pasted into an email client or sent as an encrypted attachment.
- **Decryption:** The recipient uses their private key to decrypt the email and restore it to its original readable form.

Encrypted email ensures **confidentiality**, **authentication** through digital signatures, and **integrity** of the message. It protects sensitive communication from unauthorized access, interception, and tampering during transmission.

4 Privacy Tools from PRISM-Break

Below are five recommended privacy-focused tools listed on **prism-break.org**. These tools help protect user data, communication, and online activity.

4.1 Tor Browser

Tor Browser allows anonymous web browsing by routing internet traffic through multiple encrypted nodes. It hides the user's IP address and prevents websites or trackers from monitoring activity.

4.2 KeePassXC

KeePassXC is a secure, open-source password manager that stores all passwords in an encrypted local database. It avoids cloud storage and ensures complete control over sensitive credentials.

4.3 Signal

Signal is an end-to-end encrypted messaging application used for private chats, voice calls, and video calls. It does not store metadata and prioritizes user privacy.

4.4 Syncthing

Syncthing is a peer-to-peer file synchronization tool that allows devices to share files directly without using third-party cloud servers. All transfers are encrypted and decentralized.

4.5 ProtonMail

ProtonMail is an encrypted email service based in Switzerland. It provides end-to-end encryption, meaning even the service provider cannot read user emails.

5 Open Source License

Certainly. Here is the information about the **MIT License** organized into clear, descriptive headings, strictly maintaining a paragraph-only format within each section.

5.1 The Core Purpose and Classification

The MIT License is renowned as one of the most permissive and concise open-source licenses currently in use. Originating from the Massachusetts Institute of Technology, its primary goal is to encourage maximum adoption and reuse of software with minimal legal friction. It is formally classified as a **permissive license**, meaning it grants users broad rights to use, modify, and distribute the software without imposing .

5.2 Granted Rights and Permissions

The license grants blanket permission to any individual or entity obtaining a copy of the software and its associated documentation to deal with the Software without restriction. Specifically, users are granted explicit rights to **use, copy, modify, merge, publish, distribute, sublicense, and/or sell** copies of the software.

5.3 The Only Two Conditions for Distribution

Unlike licenses that enforce reciprocal sharing, the MIT License has only two critical requirements that must be met when the software is distributed or included in a larger work.

5.4 Disclaimer of Warranty and Liability

A key component of the MIT License is its comprehensive liability disclaimer, which serves to protect the original authors. The license emphatically states that the software is provided **"AS IS,"** meaning it comes without any guarantee or warranty of any kind, whether express or implied, including warranties of merchantability or fitness for a particular purpose.

6 Self Hosted Server

6.1 About

FlatPress is a simple, lightweight, and database-free content management system designed mainly for blogging and small websites. What makes FlatPress unique is that it uses **flat files** to store all data instead of relying on traditional databases like MySQL. This design choice makes it extremely easy to install, maintain, and move between servers. Since it only requires PHP, FlatPress can run on almost any hosting environment, even very low-cost or limited ones. This simplicity also reduces the chances of database-related security risks and errors, making FlatPress a stable option for users who want a minimal system without too much technical complexity.

FlatPress offers all the essential features expected from a modern blogging platform. Users can create posts and pages directly from a clean and simple admin panel. Themes allow customization of the site's appearance, and plugins can extend the functionality as needed, such as adding contact forms, enhancing SEO, or improving the editor experience. The built-in comment system allows readers to interact, and additional plugins can add moderation or spam protection features. Because all content is stored in text files, backups are extremely easy—copying the site directory is usually enough to secure everything. This also makes FlatPress highly portable: you can move the entire site simply by uploading the folder to another server.

6.2 Installation Process (Flatpress)

To install FlatPress, first ensure your server supports PHP 5.2 or higher and has a web server such as Apache or Nginx. Unlike many CMS platforms, FlatPress does not require a database, which makes it lightweight and easy to set up. Begin by downloading the latest version of FlatPress from the official website and extracting the ZIP file on your computer. The extracted folder contains key directories like `fp-admin` for the admin panel, `fp-content` for themes, plugins, posts, and pages, as well as files like `index.php` and `fp-config.php` for configuration. Upload all these files to your web server using an FTP client or your hosting file manager, either to the root directory or a subfolder. After uploading, open the `fp-config.php` file in a text editor to configure your site settings, including the site title, admin username and password, and admin email. Make sure the `fp-content` folder and its subdirectories are writable so that FlatPress can store content. Once uploaded and configured, access the admin panel by navigating to `http://yourdomain.com/fp-admin/` in your browser and log in using your admin credentials. From the admin panel, you can create posts and pages, install themes and plugins, and manage comments. Because FlatPress stores all content in plain text files, it is highly portable and easy to back up—simply copy the folder to another server to migrate the site.

FlatPress

7 Open Source Contribution

7.1 PR 1 : First Contribution

7.1.1 Goal

The project's objective is to simplify the standard open-source contribution workflow, allowing beginners to easily add their name to the project's `Contributors.md` file.

7.1.2 The Contribution Workflow

The tutorial details the standard **fork - clone - edit - pull request** sequence, essential for collaborative coding.

7.1.3 1. Setup

- **Fork:** Create a copy of the repository in your personal GitHub account.
- **Clone:** Download the forked repository to your local machine using the `git clone` command and the SSH URL.
- **Prerequisites:** Ensure **Git** is installed; alternatives for users uncomfortable with the command line (GUI tools) are provided.

7.1.4 2. Making Changes

- **Branch:** Create a new isolated branch for your changes using `git switch -c your-new-branch-name`.
- **Edit:** Add your name to the `Contributors.md` file using a text editor.
- **Commit:** Stage the changes with `git add Contributors.md` and save them locally with `git commit -m "Add your-name to Contributors list"`.

7.1.5 3. Submission

- **Push:** Upload your local branch to your GitHub fork using `git push -u origin your-branch-name`.
- **Pull Request (PR):** Go to your GitHub repository and submit a PR via the "Compare & pull request" button for review by the project maintainers.

7.1.6 Next Steps

Upon merging the PR, the user is encouraged to celebrate their first contribution and seek out other beginner-friendly issues on the project list.

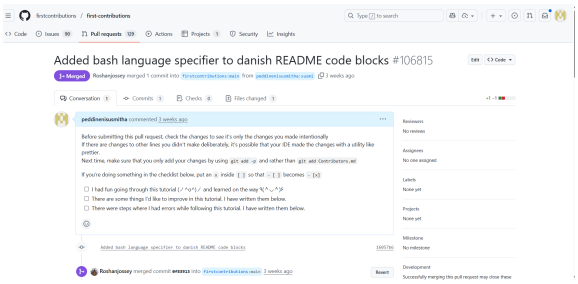


Figure 1:

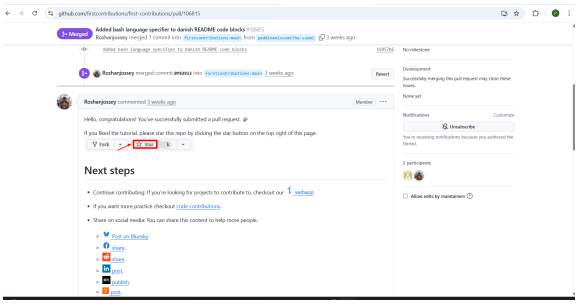


Figure 2:

7.2 PR 2 : Zero-to-mastery

Zero-to-Mastery (ZTM) is a popular, community-driven open-source organization built to help learners develop real-world programming skills. It focuses on practical learning through project-based resources, curated guides, and contributions from developers around the world. The ZTM repositories act like a collaborative support system where new developers can practice GitHub workflows, contribute to open-source projects, and explore structured learning materials that align with industry needs. It offers a friendly environment that encourages beginners to learn, contribute, and grow along with the community.

7.2.1 Licensing and Self-Hosting Options

Most Zero-to-Mastery repositories are licensed under the \textbf{MIT License}, a highly permissive open-source license. This allows anyone to freely use, modify, and distribute the code with very few restrictions. The MIT License makes ZTM resources easy to integrate into personal projects, learning materials, or community contributions.

7.2.2 Community and Support

ZTM maintains an active and encouraging global community. Learners and contributors can participate through GitHub discussions, issues, and pull requests. The organization is beginner-friendly and provides clear contribution guidelines, making it easy for first-time contributors to get involved.

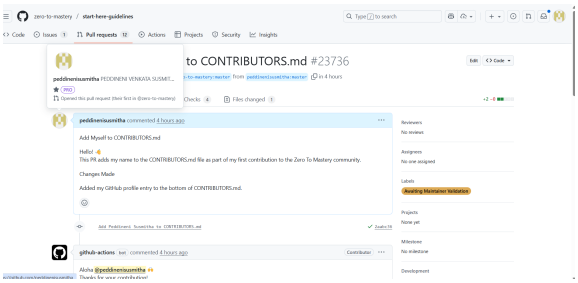


Figure 3: Enter Caption

7.3 PR 3 : Y24 Open Source Engineering

7.3.1 Introduction and Purpose

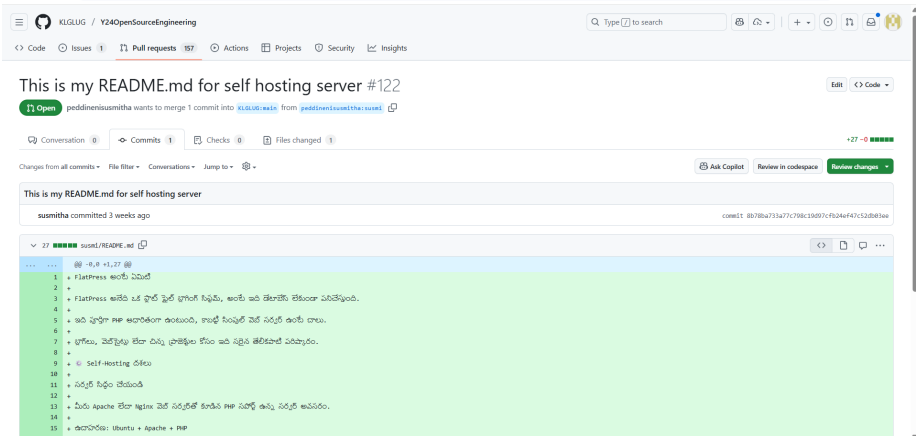
FlatPress is a **lightweight, open-source blogging platform** that works **without any database**. Instead of MySQL or MongoDB, it stores everything (posts, settings, plugins, themes) in simple **text files**. This makes it **very easy to install, move, and back up**.

7.3.2 Technical Components

FlatPress is built using a lightweight and modular architecture that focuses on simplicity and speed. It does not rely on any database system; instead, it uses a **flat-file storage mechanism**, where all data is stored in text files. The core system is written in **PHP**, making it compatible with almost all standard web hosting environments.

7.3.3 Operation and Usage

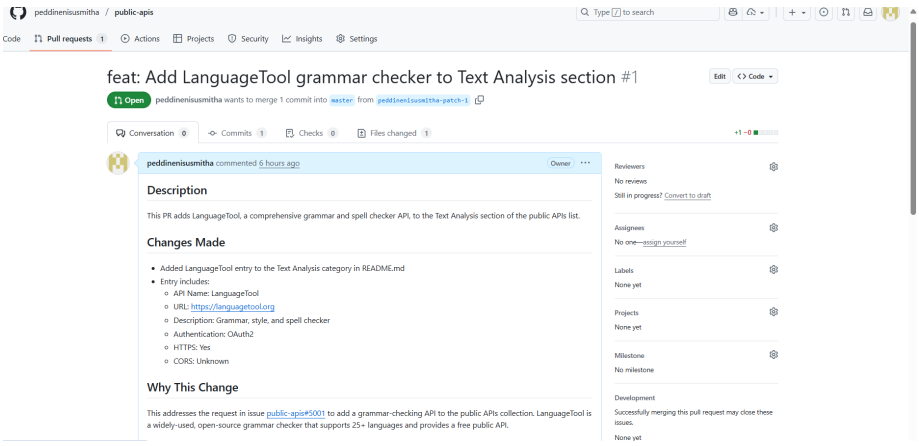
FlatPress operates as a simple, file-based blogging system where all actions are handled through lightweight PHP scripts. Users interact with the platform mainly through a clean and minimal **administration panel**, which allows them to perform essential blogging operations such as creating posts, editing pages, managing comments, and customizing the appearance of their site.



7.4 PR 4: public-apis

Public APIs (Application Programming Interfaces) are openly accessible interfaces that allow developers to interact with external software services, platforms, or data sources. They are designed to be publicly available so that anyone can use them without needing special authorization beyond basic registration or an API key. Public APIs act as a communication bridge, enabling applications to request information, send data, or perform operations on remote servers. These APIs power many everyday features such as weather forecasts, payment gateways, maps, social media integrations, and machine learning services.

Public APIs typically follow standard communication protocols like **REST**, **GraphQL**, or **gRPC**, and use data formats such as **JSON** or **XML** for requests and responses. Developers can easily integrate these APIs into websites, mobile apps, or backend systems to extend functionality without building everything from scratch. They come with documentation that explains endpoints, methods, parameters, authentication types, and usage limits.



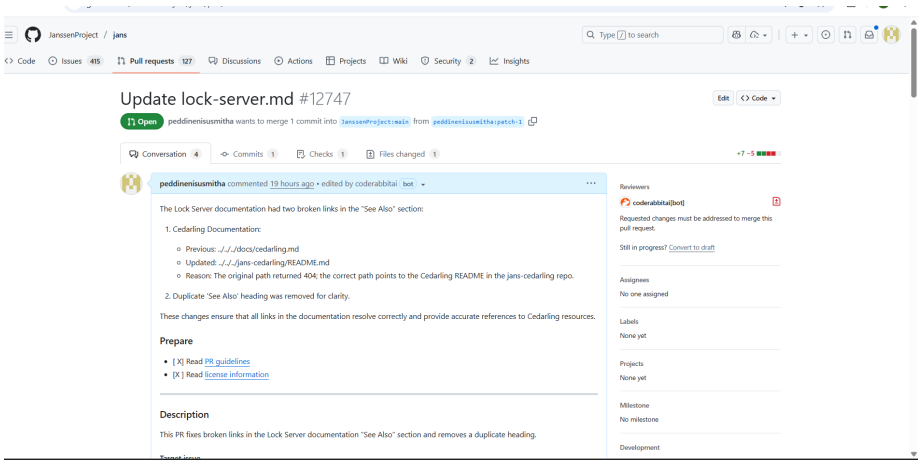
7.5 PR 5 : Update lock server

7.5.1 The Issue (What was Missing)

The **"Update Lock Server"** change refers to improvements made to the documentation of the *Janssen Lock Server*, which is a component responsible for coordinating distributed locking within the Janssen identity platform. The Lock Server helps different services avoid conflicts by controlling access to shared resources, so accurate documentation is important for developers working with this system.

In my update, I focused on fixing errors in the **See Also** section of the file `lock-server.md`. The previous documentation contained **broken links**, especially the one pointing to Cedarling documentation, which redirected to a non-existing path and resulted in a 404 error. I corrected this link so users can now access the Cedarling README in the correct repository path. Additionally, the documentation had a **duplicate "See Also" heading**, which was unnecessary and also caused markdown linting violations. I removed the duplicate heading to make the document cleaner and more readable.

These changes make the documentation more accurate, maintainable, and user-friendly. After submitting the pull request, automated tools reviewed it. The Snyk and DCO checks passed, but CodeRabbit requested a small fix—one duplicate heading was still present and needs to be removed. Once this correction is applied, the update will be ready for approval and merging.



8 Linkedin Post Links

8.1 PR :

`https://www.linkedin.com/posts/peddneni-venkata-susmitha-86797b343_excited-to-share-my-first-open-s`
`utm_source=share&utm_medium=member_desktop&rcm=ACoAAFYbYg8B4itUMUxbmmOu-1GRtEjbaQgqtk0`

8.2 Journey Of Open Source :

`https://www.linkedin.com/posts/peddneni-venkata-susmitha-86797b343_my-experience-on-open-source-eng`
`utm_source=share&utm_medium=member_desktop&rcm=ACoAAFYbYg8B4itUMUxbmmOu-1GRtEjbaQgqtk0`

8.3 Self Hosted Project :

https://www.linkedin.com/posts/dharani-mukkera-252125344_projectexpo-flatpress-opensource-activity-
utm_source=share&utm_medium=member_desktop&rcm=ACoAAFYo0R4BtHkcuF4LkoCc9Sl-SaUmgPcH-Ak