FTP Configuration

Local : https://ubuntu.com/server/docs/service-ftp

NIS Configuration

NFS Configuration

https://ubuntu.com/server/docs/service-nfs
We need to update /etc/exports file to tell server which client devices are allowed to connect to this NFS server.

```
/nfsshare 10.10.13.184(rw,sync,no_root_squash,no_subtree_check)
```

To apply this config changes command: sudo exportfs -a

TELNET Configuration

https://linuxways.net/ubuntu/how-to-install-telnet-server-and-client-on-ubuntu/

Mantis Installation

https://computingforgeeks.com/install-and-configure-mantis-bug-tracker-on-ubuntu/

Kernel Compilation

https://phoenixnap.com/kb/build-linux-kernel

Centos Installation

https://www.tecmint.com/install-centos-7-alongside-windows-10-dual-boot/

Server World

https://www.server-world.info/en/

Wordpress

https://www.digitalocean.com/community/tutorials/how-to-install-wordpress-on-ubuntu-22-04-with-a-lamp-stack

sudo systemctl restart apache2
sudo apachectl configtest

Git

If you're working with Git on a local machine with multiple users and you want to set up a repository for collaborative development offline, you can follow these steps. In this example, let's consider two users, `user1` and `user2`.

### Initial Setup:

1. **Create a New Git Repository:**
   - Open a terminal and navigate to the desired directory:
     ```bash
     cd /path/to/your/project
     ```
   - Initialize a new Git repository:
     ```bash
     git init
     ```

2. **Configure User Information:**
   - Set the user information for `user1` and `user2`:
     ```bash
     # For user1
     git config user.name "User1 Name"
     git config user.email "user1@example.com"

     # For user2
     ```

git config user.name "User2 Name"
git config user.email "user2@example.com"

### Collaborative Development:

3. **Create and Switch Branches:**
   - Create a new branch for each user to work on:
```bash
# User1 creates and switches to a new branch
git checkout -b user1_branch

# User2 creates and switches to a new branch
git checkout -b user2_branch
```

4. **Work on Respective Branches:**
   - Users can independently make changes in their branches.
```bash
# User1 makes changes and commits
git add .
git commit -m "User1's changes"

# User2 makes changes and commits
git add .
git commit -m "User2's changes"
```

5. **Switch Between Branches:**
   - Users can switch between branches to work on different features.
```bash
# User1 switches to User2's branch
git checkout user2_branch
```

### Merging Changes:

6. **Merge Changes:**
   - After completing their work, users can merge changes back to the main branch:
```bash
# User1 merges changes to the main branch
git checkout main
git merge user1_branch
```

```
# User2 merges changes to the main branch
git checkout main
git merge user2_branch
```

### Handling Conflicts:

7. **Resolve Conflicts (if any):**
   - If there are conflicts during the merge, Git will prompt users to resolve them manually.

8. **Continue Development:**
   - Users can continue working on their respective branches or create new branches as needed.

### Note:

- Ensure proper communication between users to avoid conflicts and coordinate development efforts.
- Remember that this approach is for collaborative development on a local machine. If you plan to work across different machines or want a backup of your repository, consider using a remote repository (e.g., GitHub, GitLab) for more robust version control.

These steps provide a basic workflow for collaborative Git development on a local machine. Adjustments might be necessary based on the specific requirements and collaboration patterns of your project.

SVN

https://meetawaiszafar.medium.com/install-configure-svn-server-on-ubuntu-20-04-with-apache2-6dcd7d9a49e9

tutorial : Basic SVN Tutorial

Hg Mercurial

Not supported

Debian Package Manager

https://earthly.dev/blog/creating-and-hosting-your-own-deb-packages-and-apt-repo/

Bugzilla

https://bugzilla.readthedocs.io/en/latest/installing/quick-start.html

Drupal

https://www.rosehosting.com/blog/how-to-install-drupal-on-ubuntu-22-04/

Joomla

https://hostadvice.com/how-to/website-builders/joomla/how-to-install-joomla-on-an-ubuntu/

SonarQube

https://www.digitalocean.com/community/tutorials/how-to-ensure-code-quality-with-sonarqube-on-ubuntu-18-04

SonarCloud

https://sonarcloud.io

Asanaa

ASANA