

Getting started with Fedora-Infra

by Dhriti Shikhar

FAS username: dhrish20

Twitter handle: DhritiShikhar

Fedora-Infra

- Channels:
 - #fedora
 - #fedora-appson Freenode server
- Github Repository:
<https://github.com/fedora-infra>

Fedora-Infra

- The Fedora Infrastructure team develops, deploys, and maintains various **services** for the Fedora Project.

Fedora-Infra services (1)

- Bodhi:
 - a web-system that facilitates the process of publishing updates for a Fedora-based software distribution.
 -
- Fedora-tagger:
 - This application allows users to tag and rate packages in Fedora

Fedora-Infra Services (2)

- Koji
- Pagure
- Pkgdb2
- FAS

Core members of Fedora-Infra

- Pingou <pingou>
- Ralph Bean <ralph>
- Kevin Fenzi <nirik>

Pagure

- Link to the site:
 - pagure.io
- Link to the repository:
 - <https://pagure.io/pagure/>

Pagure

- maintained by pingou
- A **light weight** git centered forge project
- based on pygit

Features of Pagure

- a web-interface for git repositories
- a ticket system
- system to create new projects
- fork existing ones
- create/merge pull-requests across or within projects

Pre-requisites

- CLI
- git
- Python

1. **Apply for FAS account**

- Link: <https://admin.fedoraproject.org/accounts/user/new>

2. Set your public key

- Generate your ssh key using:
 - `ssh-key gen -t rsa`
- Copy the public key from `~/.ssh/id_rsa.pub` to your pagure account

3. Fork the pagure repository

- Click on the fork button on the upper right side

4. Clone the pagure repository

- SSH Clone:

`ssh://git@pagure.io/pagure.git`

- Http Clone:

`https://pagure.io/pagure.git`

5. Pagure: Issues

- All issues:
 - <https://pagure.io/pagure/issues>
- Easyfix:
 - fedoraproject.org/easyfix/

Finding and understanding a bug

- look for keywords
- Read the comments

Where to make the changes?

- Use tools such as:
 - grep
 - pdb

6. Creating a branch

- Create a new branch using the command:
`git checkout -b <branch_name>`
- No special characters in the branch name

Making the changes

- Check your changes using:
git diff

7. Add the changes to staging area

- Use the command:
`git add <filename>`

8. Commit the changes

- Use the command:
`git commit -m "<commit message>"`
- To amend the commit message:
`git commit -a`
`git commit --amend`

9. Adding a remote

Use the command:

```
git remote add <remote_name> <url>
```

10. Pushing the changes to remote

Use the command:

```
git push <remote_name> <branch_name>
```

11. Sending a PR

- Go to your repository
- Go to the branch

12. Follow up after sending a PR

- Keep a track of your PR
- Reply to the comments
- Make appropriate changes and send a PR again

End