

01

Fedora Classroom

Git 101 with Pagure

02



Topics

Version Control Systems
Why Git?
VCS Hosting Sites
Fedora Pagure
Exploring Pagure
Git Fundamentals

03

Been in situations like these?

CODE BACKUP

Now what should I name the latest version of the code?

Project_TEST.py

Project_SOLVED_PROB.py

Project_BACKUP_1.py

Project_BACKUP_2.py

CODE COLLABORATION

Mahij, I have mailed you the code. Please check!

I have got that mail, Mohit but where are the dependencies?

Oh no! Wait, I will send them as attachment to you

I have got them but not the code!

Version Control Systems



YOUR COMPANION IN BUILDING COMPLEX PROJECTS

- Seamlessly manage changes made to your codebase
- Handle concurrent additions made by multiple people
- Track edits and progress made per requested feature
- Backup state at timestamp and rollback if and when needed
- Experimenting features safely and adding them to release

Version Control Tools

05

Git 101 with Pagure

GIT

- Most widely used version control system in development
- Distributed version control system for tracking changes
- Full repository based in a directory with all changes
- Does not primarily depend upon network access
- Created by Linus Torvalds

APACHE SUBVERSION

- Software versioning and revision control system
- Capable of keeping history of code, docs and web pages
- Aims to be a capable successor of the widely used CVS
- Used in projects like ASF, FreePascal, FreeBSD etc.
- Created by CollabNet Inc.

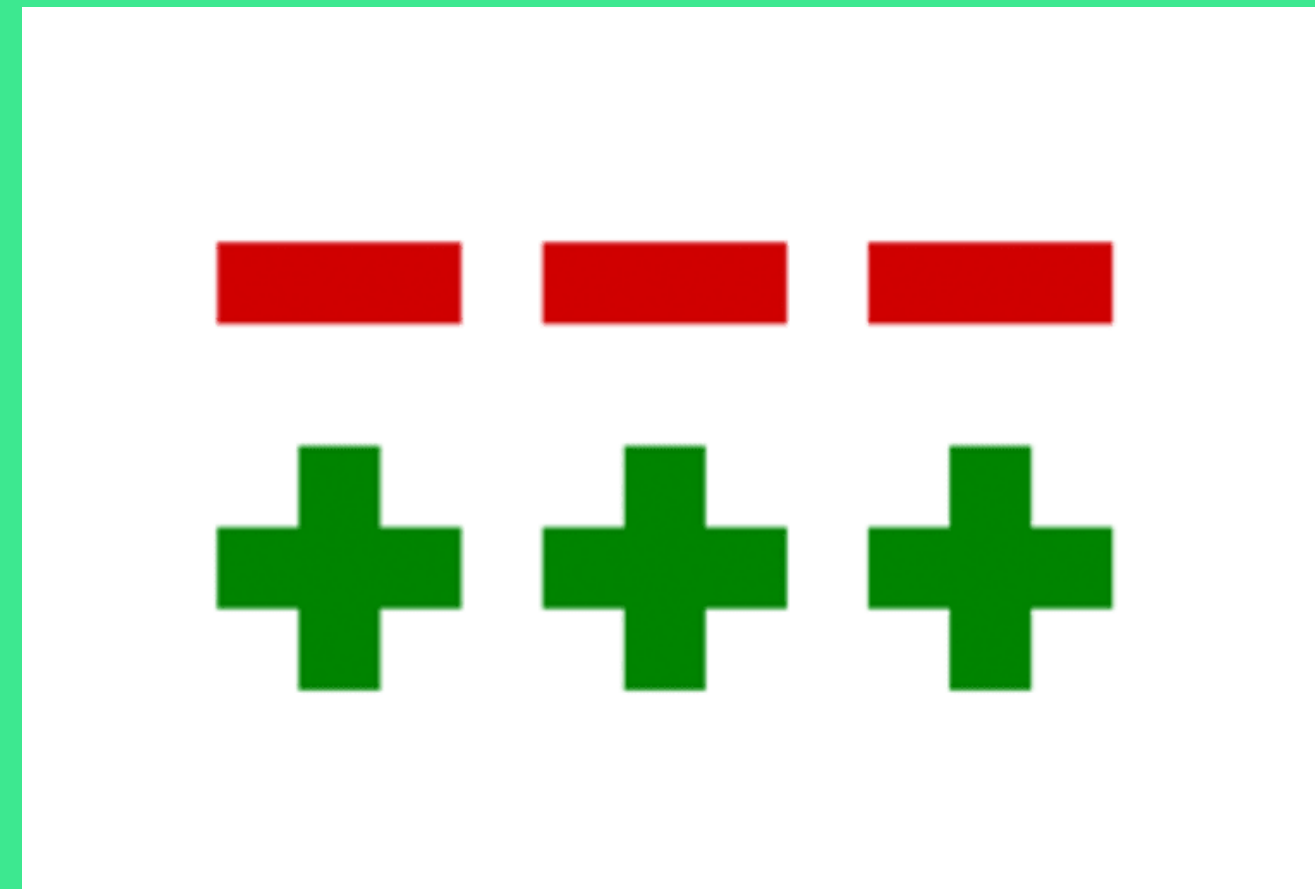
MERCURIAL

- Distributed revision control tool for software developers
- Capable of robust handling of plain text and binary files
- Aims to be high performing, scalable and decentralized
- Mainly implemented in Python with some parts in C
- Created by Matt Mackall

Why Git?

Why not something else?

- Uses SHA compression for changes which makes it fast and robust
- Conveniently handle conflicts when same file is worked upon by multiple people
- Safe experimentation using sandbox created by branches for quick feedback
- Immutable commits can easily facilitate rollbacks and revert changes
- Documentation is widely available on Git with fellow contributors to help you



Fedora Pagure

07

pagure.io

A lightweight Git web-interface built with
and for open-source tools based on
PyGit2 hosting over 2000 projects as of
July 7, 2020

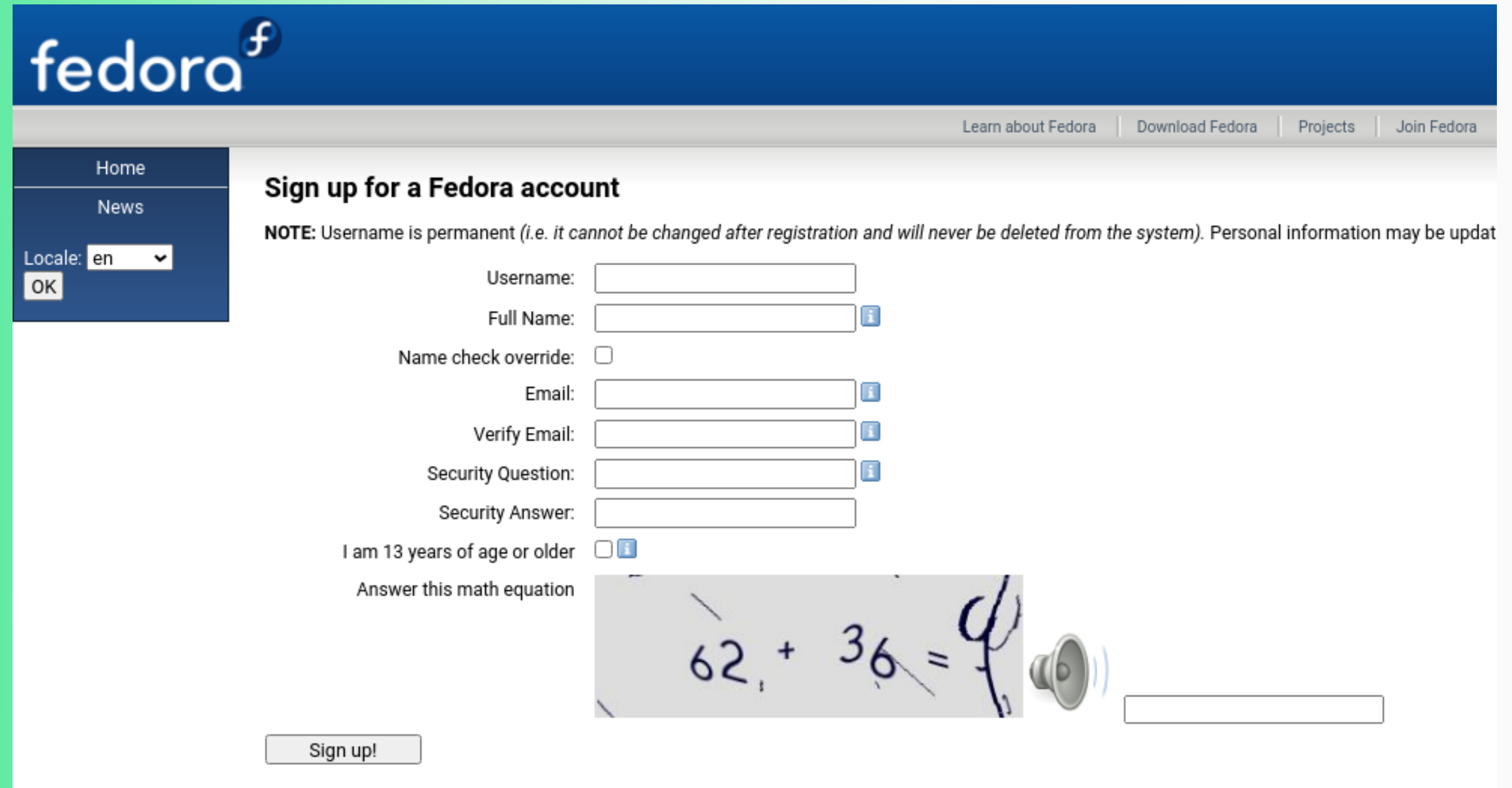
The screenshot shows the Fedora Pagure website interface. At the top, there's a header with the Fedora logo and the word "PAGURE" in blue. A "Log In" button is in the top right corner. Below the header, a large heading says "Welcome to Fedora Pagure" followed by the text "Pagure is an Open Source software code hosting system." Below this, there are tabs for "Project", "Users", and "Groups". The main content area is titled "All Projects 2158" and includes a search bar and a "Sort" dropdown. The projects are displayed in a grid of nine items, each with a repository icon, a title, a description, and creation information.

Project Name	Description	Created	Stars
neuro-sig/20190909-OSB-workshop-presentation	NeuroFedora presentation for the OSB workshop	created 10 months ago	1
neuro-sig/2019-flock-neurofedora	Presentation slides for NeuroFedora talk at Flock	created a year ago	0
neuro-sig/2019-NeuroFedora-poster-CNS	The NeuroFedora poster at CNS*2019 Barcelona	created a year ago	0
neuro-sig/2020-CNS-Abstract	Abstract to submit to CNS*2020	created 5 months ago	0
2DadsWithBaggage	2 Dads with Baggage is a LGBT family travel blog. Global adventures in gay parenthood. We are two dads with two teen girls, sharing our global experience of traveling, lifestyle, parenthood and gay travel with kids. Read our blog as we travel, navigate, experience and grow. 2 dads with 2 teen daughters explore. Travel reviews, travel tips, parenting humor, gay parenting, gay travel, lifestyle, art, culture. Feel free to contact us at (+1)619-865-3382 and mail us on jon@2dadswithbaggage.com .	created a month ago	0
389-admin	389 Administration Server	created 3 years ago	6
389-admin-console	389 Java Administration Server	created 3 years ago	1
389-adminutil	389 Administration Server Utility Library	created 3 years ago	0
389-console	389 Java Console	created 3 years ago	1

Create an account

08

In order to contribute to the projects on Pagure and to create your own repositories, you need to have an FAS account under your name.



The screenshot shows the Fedora account registration page. The header features the Fedora logo and navigation links: "Learn about Fedora", "Download Fedora", "Projects", and "Join Fedora". A sidebar on the left contains links for "Home" and "News", a locale selector set to "en", and an "OK" button. The main content area is titled "Sign up for a Fedora account" and includes a note: "NOTE: Username is permanent (i.e. it cannot be changed after registration and will never be deleted from the system). Personal information may be updated." The registration form consists of the following fields and options:

- Username:
- Full Name: ⓘ
- Name check override: ☐
- Email: ⓘ
- Verify Email: ⓘ
- Security Question: ⓘ
- Security Answer:
- I am 13 years of age or older: ☐ ⓘ
- Answer this math equation:  ⓘ

A "Sign up!" button is located at the bottom left of the form area.

09

Exploring Pagure

Fedora Pagure

GIT 101 WITH PAGURE

LOGGING INTO YOUR ACCOUNT

Signup and login to your account in Pagure

GENERATING AND ADDING SSH KEYS

Using RSA to authenticate Git accesses

CREATING YOUR FIRST REPOSITORY

Make your first repository on Pagure

10

Open up Firefox

Or any other browser of your choice

Head to

<https://admin.fedoraproject.org/accounts/>

11

Git Fundamentals

PART 1 of 4

GIT 101 WITH PAGURE

INSTALLING GIT

Installing Git on Fedora (if not already)

INITIALIZING A REPOSITORY

Turn your project folder into a Git repository

CLONING A REPOSITORY

Create a copy of a specific repository

Installing Git

12

STEP #1

Open up a terminal

STEP #2

Type in

```
sudo dnf install git-all -y
```

and press ENTER

STEP #3

Type in

```
git version
```

and press ENTER

Initializing a repository

13

STEP #1

Navigate to your project directory with Nautilus (or a file manager of your choice)

STEP #2

Right click on an empty location and open a terminal

STEP #3

Type in
git init
and press ENTER

Cloning a repository

14

STEP #1

Pick a project of your choice from any Git hosting site

STEP #2

Click on the clone button and copy the HTTPS URL

STEP #3

Type in
git clone <CLONE-URL>
and press ENTER

15

Git Fundamentals

PART 2 of 4

GIT 101 WITH PAGURE

ADDING FILES

Staging modifications before committing

COMMITTING CHANGES

Save all the changes you have just staged

EXPERIMENTING WITH CODE

Create and make changes exclusive to a branch

Adding files

16

STEP #1

Navigate inside a local Git repository folder with Nautilus (or a file manager of your choice)

STEP #2

Right click on an empty location and open a terminal

STEP #3

Type in
git add <FILENAME>
and press ENTER

Committing Changes

17

STEP #1

Navigate inside a local Git repository folder with Nautilus (or a file manager of your choice)

STEP #2

Right click on an empty location and open a terminal

STEP #3

Type in
git commit -m "<COMMIT-MESSAGE>"
and press ENTER

Experimenting with code

18

STEP #1

In a terminal with your Git repository folder as your current working directory, type in

```
git branch --all
```

and press ENTER

STEP #2

Type in

```
git branch <BRANCH-NAME>
```

and press ENTER

STEP #3

Type in

```
git checkout <BRANCH-NAME>
```

and press ENTER

Experimenting with code

19

STEP #4

In a checked out branch, add and commit changes that you want to make.

STEP #5

Type in
git checkout master
and press ENTER

STEP #6

Type in
git merge <BRANCH-NAME>
and press ENTER to merge to master.

20

Git Fundamentals

PART 3 of 4

GIT 101 WITH PAGURE

PUSH CHANGES TO REMOTE

Upload all local branch commits to the remote

FETCH CHANGES FROM REMOTE

Download changes made in the remote

PULL CHANGES FROM REMOTE

Update your current branch with remote changes

Push changes to remote

21

STEP #1

Navigate inside a local Git repository folder with Nautilus (or a file manager of your choice)

STEP #2

Right click on an empty location and open a terminal.
Add and commit changes.

STEP #3

Type in
git push
and press ENTER

Fetch changes from remote

22

STEP #1

Open up a terminal with your Git repository folder as your current working directory.

STEP #2

Type in
git fetch
and press ENTER

STEP #3

Find fetched changes in
refs/remotes/<REMOTE>/

Pull changes from remote

23

STEP #1

Open up a terminal with your Git repository folder as your current working directory.

STEP #2

Type in
git pull
and press ENTER

STEP #3

Find updated changes made in your current branch, pulled from the remote repository.

24

Git Fundamentals

PART 4 of 4

GIT 101 WITH PAGURE

CHECKING STATUS

Current state of your Git directory and staging area

COMMON ISSUES

Problems you might face when working with Git

FURTHER READING

Remember these are just fundamentals

Checking status

25

STEP #1

Navigate to your project directory with Nautilus (or a file manager of your choice)

STEP #2

Right click on an empty location and open a terminal

STEP #3

Type in
git status
and press ENTER

Common issues

26

I CANNOT PUSH TO A REPOSITORY

You cannot push to somebody else's repository unless you have an access to it. Make a fork of it first.

REPOSITORY IS CLUTTERED

Either selectively add files to the repository or add a .gitignore file to let Git know which files to avoid.

CHANGES TO MASTER BROKE MY CODE

It is generally a good practice to push your code to a branch first and merge it to the master after testing.

Further reading

27

THE ORIGINAL DOCUMENTATION

Visit
<https://git-scm.com/>
for learning more

SEARCH ENGINES ARE YOUR BEST FRIEND

Look up your issues on Google or DuckDuckGo. There is always someone who had the questions you have now.

FEDORA JOIN SIG

Ask a question or just drop by to say hello!
<https://docs.fedoraproject.org/en-US/fedora-join>



Thank you!

*Feel free to message us on Join SIG in case
of doubts and queries*