Red Hat B.U.I.L.D.

Introduction to GitHub

Learning Version Control and Collaboration

Mikela Jackson
Technical writer 3 for Advanced Cluster Management for Kubernetes (RHACM)

Kimla Lee
Data Portfolio Lead, XE AI and Data

Red Hat B.U.I.L.D.

Red Hat

CONFIDENTIAL designator

Introduction to GitHub

Learning Version Control and Collaboration

Urvashi Mohnani

Principal Software Engineer

OpenShift Containers

Red Hat
B.U.I.L.D.

Red Hat

What is GitHub?

A web-based platform for version control and collaborative software development.

GitHub is built on Git, a distributed version control system created by Linus Torvalds.

Key Features:

- Repository Hosting: Store and manage code repositories in the cloud.
- Version Control: Track and manage changes to the codebase using Git.
- Branching and Merging: Experiment with new features and fix bugs in isolated branches before merging them into the main codebase.
- Pull Requests (PR): Propose changes to the codebase and review them through formal process.
- Issue Tracking: Manage bugs, feature requests, and other project tasks.
- Continuous Integration (CI): Automate testing and deployment workflows.
- Code Review: Facilitate peer reviews to maintain code quality.

Why Learn GitHub?

Unlocking the Power of Collaborative Development

Global Community

A diverse user base with developers from various countries and regions. It supports collaborative efforts on thousands of open-source and enterprise projects worldwide, and accommodates multiple programming languages.

Open Source Contributions

Developers can enhance their skills and reputation by contributing to projects, and they can leverage existing open-source software to improve their own projects, saving time and resources.

Scalability and Integrations

It suitable for both small personal projects and large enterprise applications. It also integrates seamlessly with various tools and services like CI/CD pipelines, code editors, and third-party applications.

Understanding Version Control in Software Development

Version Control in Git

Version Control

Definition – Version control is a system that records changes to files over time, allowing you to recall specific versions later.

Importance – Enables collaboration, tracks changes, facilitates experimentation, and ensures code integrity.

Git: The Foundation of GitHub

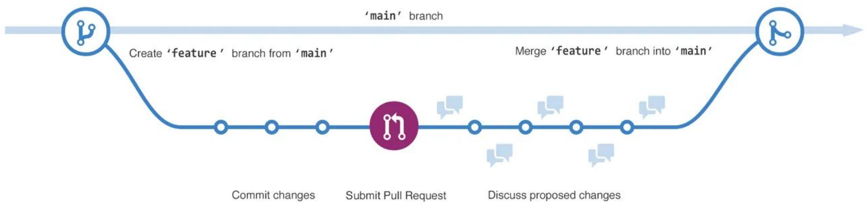
Definition – Git is a distributed version control system (DVCS) designed for speed and efficiency.

Importance – Provides a robust framework for managing code changes, branching, merging, and collaboration.

GitHub – Built on Git, GitHub extends version control with additional features for collaboration, project management, and community engagement

Key GitHub Terms

Essential Concepts for Effective Collaboration



Repository

A project or codebase.

Commit

Saving changes to the project.

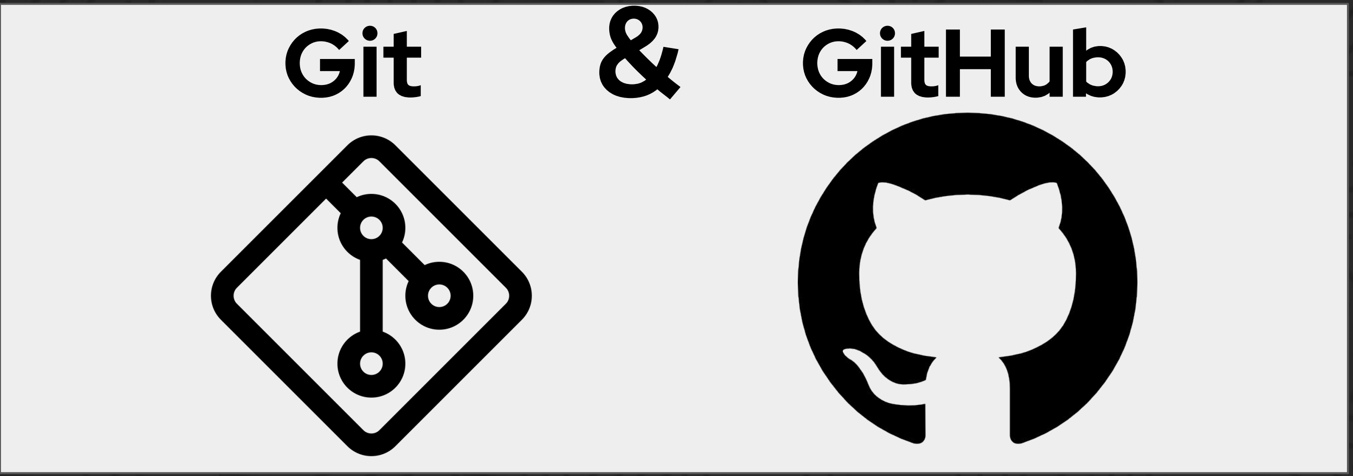
Branch

A copy of the project for separate development

Pull Request

Proposing changes to be merged.

Getting Started with GitHub

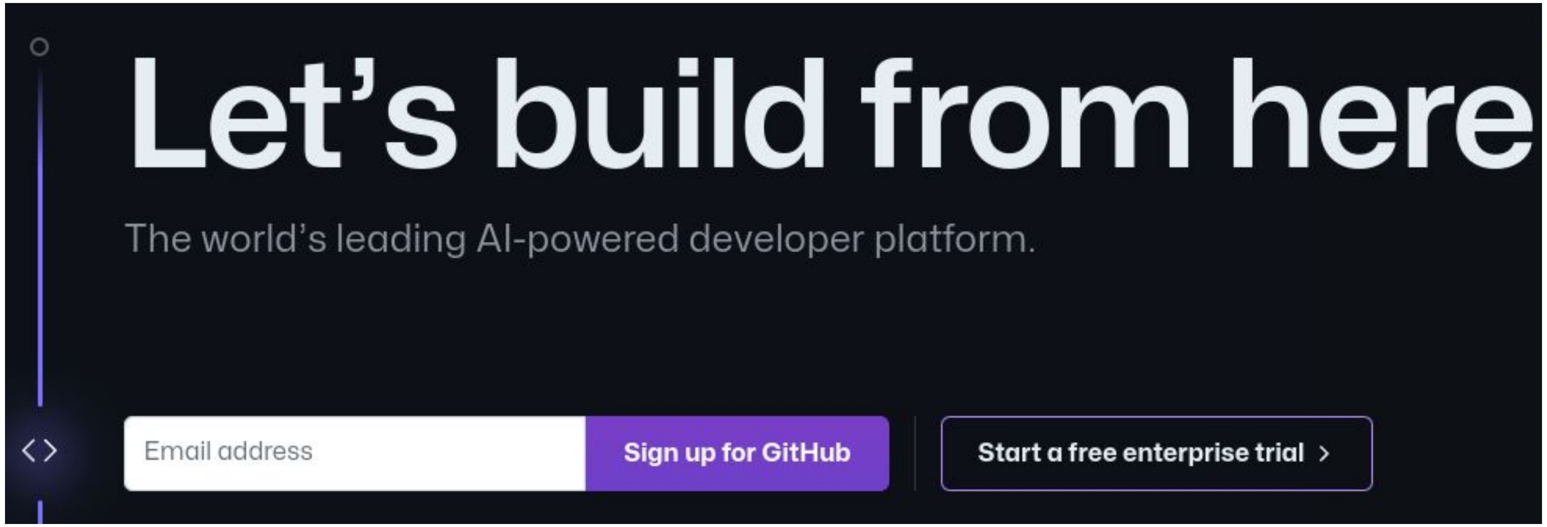


GitHub essentials like repositories, branches, commits, and pull requests.

Creating a GitHub Account

Step By Step Instructions

1. Navigate to https://github.com/
2. If you do not have an account yet, click "Sign Up for GitHub"
3. Follow the prompts to create your personal account



Let's build from here

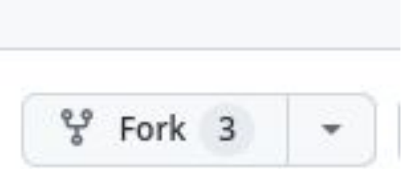
The world's leading AI-powered developer platform.

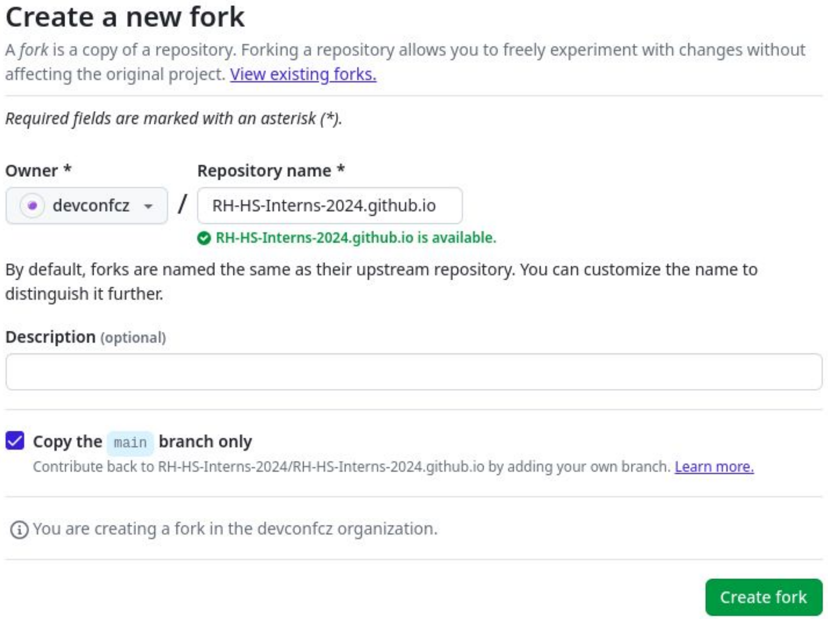
Fork a Repository

Repository is like folder that contains related items, such as files, images, videos, or even other folders.

Forks are often used to iterate on ideas or changes before they are proposed back to the upstream repository

1. Head to https://github.com/RH-HS-Interns-2024/RH-HS-Interns-2024.github.io
2. In the upper-right area, click on "Fork".
3. Click on "Create fork".
4. This will create a copy of the repository in your github account
   a. You can make changes here without affecting the original repository!





Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. View existing forks.

Required fields are marked with an asterisk (*).

Owner * devconfcz / Repository name * RH-HS-Interns-2024.github.io

RH-HS-Interns-2024.github.io is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

☑ Copy the `main` branch only

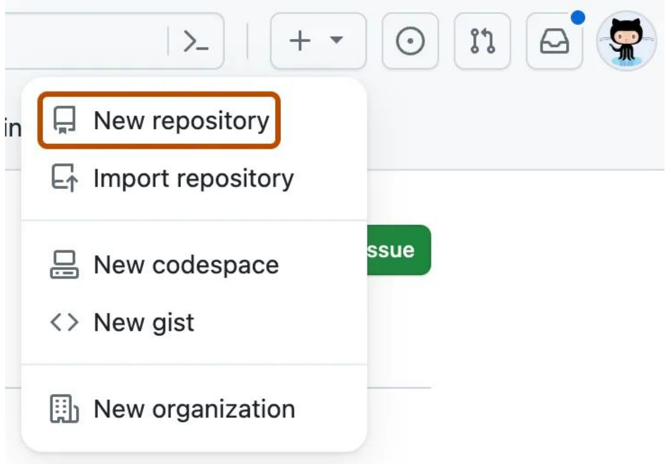
Contribute back to RH-HS-Interns-2024/RH-HS-Interns-2024.github.io by adding your own branch. Learn more.

You are creating a fork in the devconfcz organization.

Create a Repository

Repository is like folder that contains related items, such as files, images, videos, or even other folders.

1. In the upper-right corner of the page, select +, click New repository
2. In the "Repository name" box type hello-world.
3. In the "Description", box, type a short description (i.e. I am learning GitHub.)
4. Select whether repository will be Public or Private. For this tutorial, make it public.
5. Select Add a README file.
6. Click Create repository.

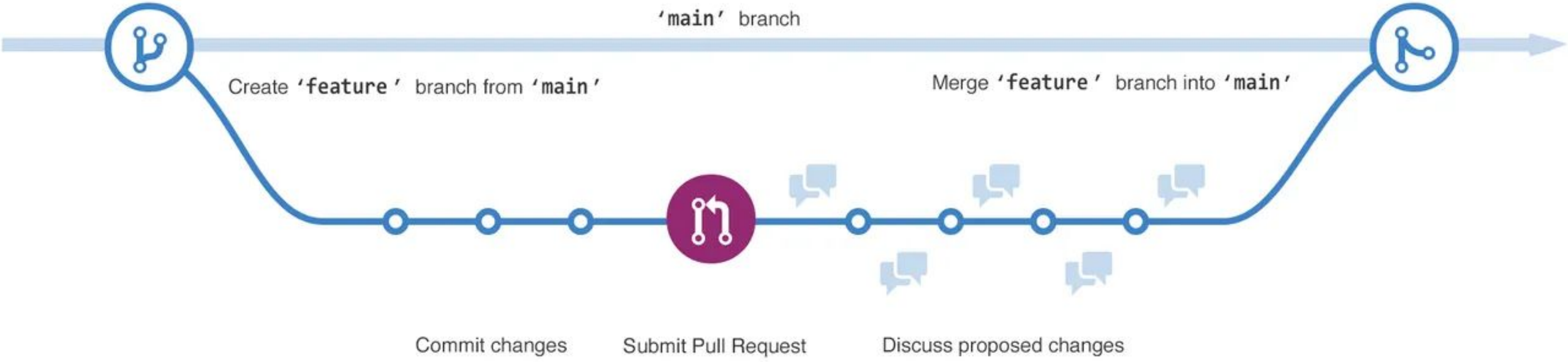


Branching

Branching lets you have different versions of a repository at one time.

This diagram shows:

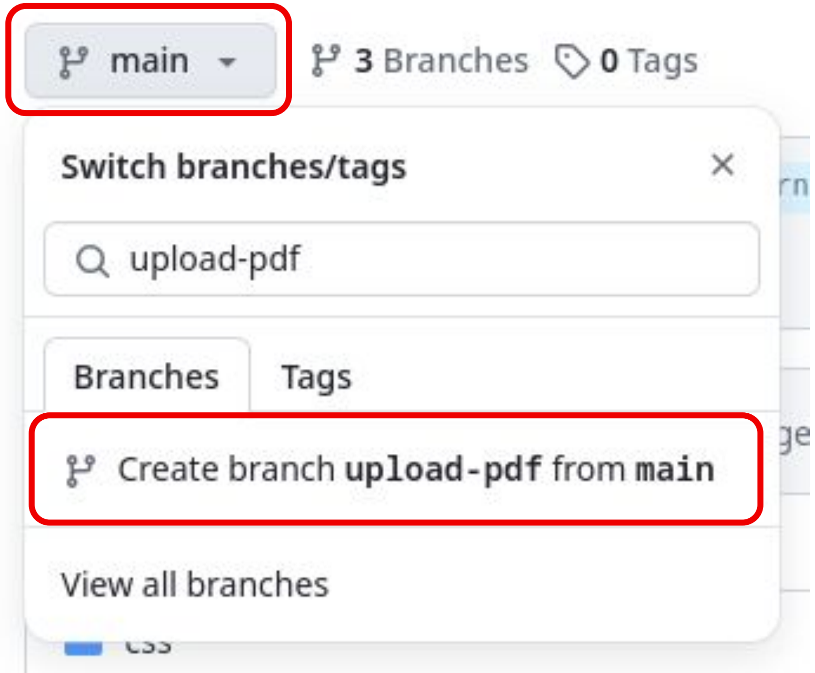
- The main branch
- A new branch called feature
- The journey that feature takes before it's merge into main



Create a Branch

Step by Step Instructions

1. Click the Code tab of the repository you just forked.
2. Above the file list, click drop-down menu that says main.



3. Type a branch name, upload-pdf, into textbox.
4. Click **Create branch: upload-pdf from main**.

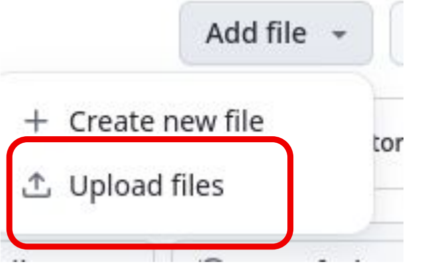
Now you have two branches, main and upload-pdf. Right now, they look exactly the same. Next you'll add changes to the new upload-pdf branch.

Make and commit changes

On GitHub, saved changes are called commits. Each commit has an associated commit message, which is a description explaining why a particular change was made. Commit messages capture the history of your changes so that contributors can understand what you've done and why.

1. Under the upload-pdf branch you created, navigate to interns/YOUR-NAME/Journals and click the README.md file.
2. Download your journal 1.
3. Click on the "Add file" button in the upper-right corner and select "Upload files".
4. Drag and drop your downloaded Journal 1 there, add a commit message and click on

   "commit changes".



5. Also update your README.md file. To edit the file, click the **Edit this file** button.
6. In the editor, write a bit about yourself.
7. Click Commit changes.
8. In the "Commit changes" box, write a commit message that describes the changes.
9. Click Commit changes.

These changes will be made only to the README file on your upload-pdf branch, so now this branch contains content that's different from main.

Make and commit changes

On GitHub, saved changes are called commits. Each commit has an associated commit message, which is a description explaining why a particular change was made. Commit messages capture the history of your changes so that contributors can understand what you've done and why.

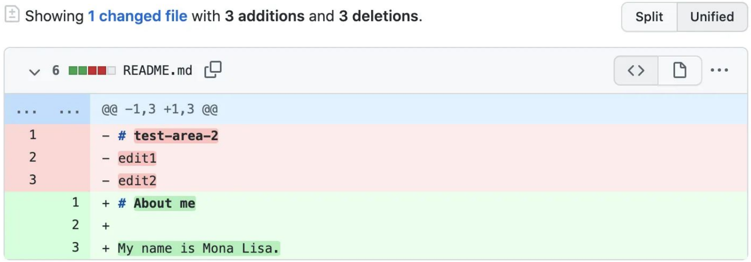
1. Now click on the **index.html** file in the journals folder
2. To edit the file, click ✎ .
3. In the editor, copy the code snippet from your Journals/README.md and add it to the Journals/index.html file. Add the code after the Journals header (`<h1>Journals</h1>`).
4. Click Commit changes.
5. In the “Commit changes” box, write a commit message that describes the changes.
6. Click Commit changes.

These changes will be made only to the README file on your upload-pdf branch, so now this branch contains content that's different from main.

Open a Pull Request

Pull requests are the heart of collaboration on GitHub. When you open a pull request, you're proposing your changes and requesting that someone review and pull in your contribution and merge them into their branch.

1. Click the Pull Requests tab of your RH-HS-Interns-2024.github.io forked repository.
2. Click New pull request.
3. In the Example Comparisons box, select the branch you made, upload-pdf, to compare with main (original).
4. Look over your changes in the diffs on the Compare page, make sure they're wh you want to submit.
5. Click Create pull request.
6. Give your pull request a title and write a brief description of your changes. You can include emojis and drag and drop images and gifs.
7. Click Create pull request.



Showing 1 changed file with 3 additions and 3 deletions.

```
@@ -1,3 +1,3 @@
- # test-area-2
- edit1
- edit2
+ # About me
+
+ My name is Mona Lisa.
```

Reviewing a Pull Request

Reviews allow collaborators to comment on the changes proposed in pull requests, approve the changes, or request further changes before the pull request is merged.

For more information about pull requests, check out this link.

Merge Your Pull Request

A pull request may introduce changes to code that conflict with the existing code on the main. If there are any conflicts, GitHub will alert you about the conflicting code and prevent merging until the conflicts are resolved

1. Add comments on your PR to request review and answer any review questions.
2. Wait for a maintainer to approve and merge your PR. You will receive a message that the request was successfully merged and the request was closed.
3. Click Delete branch. Now that your pull request is merged and your changes are on main, you can safely delete the upload-pdf branch. If you want to make more changes to your project, you can always create a new branch and repeat this process,
4. Click back to the Code tab of the https://github.com/RH-HS-Interns-2024/RH-HS-Interns-2024.github.io repository to see your published changes on main.

What did you learn?

Learned to create a project and make a pull request on GitHub.

Learned how to:

- Fork a repository.
- Start and manage a new branch.
- Change a file and commit those changes to GitHub.
- Open a pull request.
- Request review and merge from maintainers.

Next Steps

Learn more.

- Take a look at your GitHub profile and you'll see your work reflected on your contribution graph.
- If you want to practice the skills you learned in this tutorial, try the GitHub Skills "Introduction to GitHub" course.
- Place your resume on GitHub to share with potential employers by using these steps. Also, they will be able to see all your contributions.

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

in linkedin.com/company/red-hat

▶ youtube.com/user/RedHatVideos

f facebook.com/redhatinc

twitter.com/RedHat