

Experiment No. 2

Title:

Implement GitHub Operations using Git.

Objective:

The objective of this experiment is to guide you through the process of using Git commands to interact with GitHub, from cloning a repository to collaborating with others through pull requests.

Introduction:

GitHub is a web-based platform that offers version control and collaboration services for software development projects. It provides a way for developers to work together, manage code, track changes, and collaborate on projects efficiently. GitHub is built on top of the Git version control system, which allows for distributed and decentralised development.

Key Features of GitHub:

- **Version Control:** GitHub uses Git, a distributed version control system, to track changes to source code over time. This allows developers to collaborate on projects while maintaining a history of changes and versions.
- **Repositories:** A repository (or repo) is a collection of files, folders, and the entire history of a project. Repositories on GitHub serve as the central place where code and project-related assets are stored.
- **Collaboration:** GitHub provides tools for team collaboration. Developers can work together on the same project, propose changes, review code, and discuss issues within the context of the project.
- **Pull Requests:** Pull requests (PRs) are proposals for changes to a repository. They allow developers to submit their changes for review, discuss the changes, and collaboratively improve the code before merging it into the main codebase.
- **Issues and Projects:** GitHub allows users to track and manage project-related issues, enhancements, and bugs. Projects and boards help organize tasks, track progress, and manage workflows.
- **Forks and Clones:** Developers can create copies (forks) of repositories to work on their own versions of a project. Cloning a repository allows developers to create a local copy of the project on their machine.
- **Branching and Merging:** GitHub supports branching, where developers can create separate lines of development for features or bug fixes. Changes made in branches can be merged back into the main codebase.

- **Actions and Workflows:** GitHub Actions enable developers to automate workflows, such as building, testing, and deploying applications, based on triggers like code pushes or pull requests.
- **GitHub Pages:** This feature allows users to publish web content directly from a GitHub repository, making it easy to create websites and documentation for projects.

Benefits of Using GitHub:

- **Collaboration:** GitHub facilitates collaborative development by providing tools for code review, commenting, and discussion on changes.
- **Version Control:** Git's version control features help track changes, revert to previous versions, and manage different branches of development.
- **Open Source:** GitHub is widely used for hosting open-source projects, making it easier for developers to contribute and for users to find and use software.
- **Community Engagement:** GitHub fosters a community around projects, enabling interaction between project maintainers and contributors.
- **Code Quality:** The code review process on GitHub helps maintain code quality and encourages best practices.
- **Documentation:** GitHub provides a platform to host project documentation and wikis, making it easier to document codebases and project processes.
- **Continuous Integration/Continuous Deployment (CI/CD):** GitHub Actions allows for automating testing, building, and deploying applications, streamlining the development process.

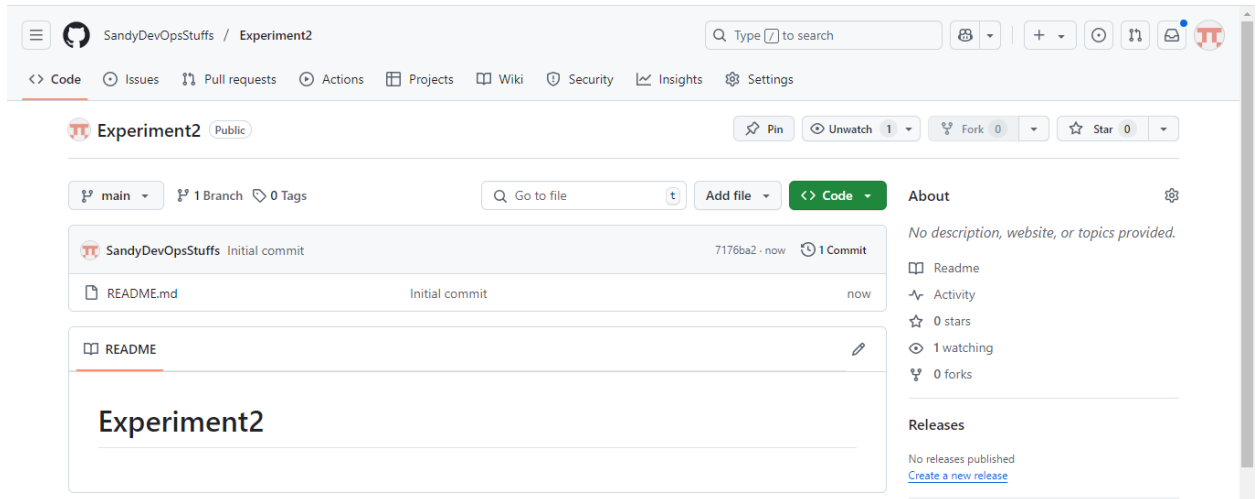
Prerequisites:

- Computer with Git installed: <https://git-scm.com/downloads>
- GitHub account: <https://github.com/>
- Internet connection

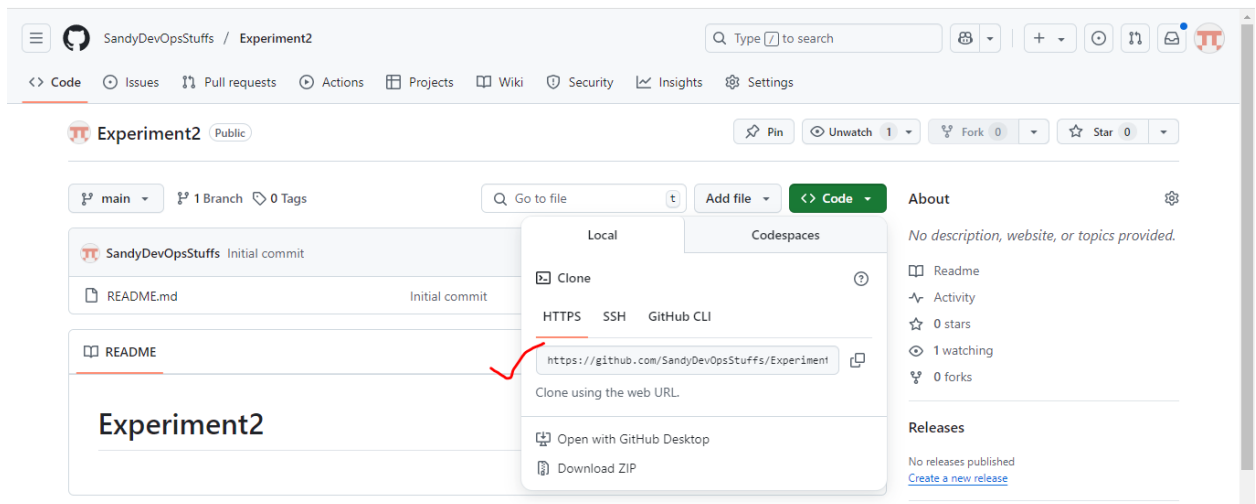
Experiment Steps:

Step 1: Cloning a Repository

- Sign in to your GitHub account.
- Create a public repository `Experiment2`.



- Click the "Code" button and copy the repository URL (HTTP or SSH).



- Open your terminal or command prompt.
- Navigate to the directory where you want to clone the repository.

```
ls -la
```

```

ubuntu@ip-172-31-5-122:~$ ls -la
total 100864
drwxr-x--- 10 ubuntu ubuntu    4096 May 11 18:53 .
drwxr-xr-x  3 root   root      4096 Apr  9 08:47 ..
drwxrwxr-x  2 ubuntu docker    4096 Apr  9 09:20 .aws
-rw-----  1 ubuntu ubuntu    5333 May 11 19:12 .bash_history
-rw-r--r--  1 ubuntu ubuntu     220 Mar 31 2024 .bash_logout
-rw-r--r--  1 ubuntu ubuntu    3771 Mar 31 2024 .bashrc
drwx-----  3 ubuntu ubuntu    4096 Apr  9 09:26 .cache
drwx-----  2 ubuntu ubuntu    4096 Apr  9 11:36 .docker
drwxr-xr-x  3 ubuntu ubuntu    4096 Apr 13 11:11 .kube
-rw-----  1 ubuntu ubuntu      20 May 11 18:53 .lessht
-rw-r--r--  1 ubuntu ubuntu     807 Mar 31 2024 .profile
drwx-----  2 ubuntu ubuntu    4096 May 11 18:44 .ssh
-rw-r--r--  1 ubuntu ubuntu      0 Apr  9 08:53 .sudo_as_admin_successful
-rw-----  1 ubuntu docker   10681 May 11 18:07 .viminfo
drwxrwxr-x  3 ubuntu ubuntu    4096 May 11 18:57 Experiment1
drwxrwxr-x  3 ubuntu docker    4096 Apr  8 18:42 aws
-rw-rw-r--  1 ubuntu docker 68286133 Apr  9 09:09 awscli2.zip
-rw-rw-r--  1 ubuntu docker 34916991 Apr  9 09:05 eksctl.tar.gz
drwxrwxr-x  3 ubuntu docker    4096 Apr 13 09:41 flask-app
ubuntu@ip-172-31-5-122:~$
ubuntu@ip-172-31-5-122:~$
ubuntu@ip-172-31-5-122:~$ |

```

- Run the following command:

```
git clone <repository_url>
```

- Replace <repository_url> with the URL you copied from GitHub.
- This will clone the repository to your local machine.

```

ubuntu@ip-172-31-5-122:~$ git clone https://github.com/SandyDevOpsStuffs/Experiment2.git
Cloning into 'Experiment2'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
ubuntu@ip-172-31-5-122:~$

```

```
ls -la
```

```

ubuntu@ip-172-31-5-122:~$ ls -la
total 100868
drwxr-x--- 11 ubuntu ubuntu 4096 May 13 06:54 .
drwxr-xr-x  3 root   root   4096 Apr  9 08:47 ..
drwxrwxr-x  2 ubuntu docker 4096 Apr  9 09:20 .aws
-rw-----  1 ubuntu ubuntu 5333 May 11 19:12 .bash_history
-rw-r--r--  1 ubuntu ubuntu 220  Mar 31 2024 .bash_logout
-rw-r--r--  1 ubuntu ubuntu 3771 Mar 31 2024 .bashrc
drwx-----  3 ubuntu ubuntu 4096 Apr  9 09:26 .cache
drwx-----  2 ubuntu ubuntu 4096 Apr  9 11:36 .docker
drwxr-xr-x  3 ubuntu ubuntu 4096 Apr 13 11:11 .kube
-rw-----  1 ubuntu ubuntu  20  May 11 18:53 .lessht
-rw-r--r--  1 ubuntu ubuntu 807  Mar 31 2024 .profile
drwx-----  2 ubuntu ubuntu 4096 May 11 18:44 .ssh
-rw-r--r--  1 ubuntu ubuntu  0  Apr  9 08:53 .sudo_as_admin_successful
-rw-----  1 ubuntu docker 10681 May 11 18:07 .viminfo
drwxrwxr-x  3 ubuntu ubuntu 4096 May 11 18:57 Experiment1
drwxrwxr-x  3 ubuntu ubuntu 4096 May 13 06:54 Experiment2 ✓
drwxr-xr-x  3 ubuntu docker 4096 Apr  8 18:42 aws
-rw-rw-r--  1 ubuntu docker 68286133 Apr  9 09:09 awscli2.zip
-rw-rw-r--  1 ubuntu docker 34916991 Apr  9 09:05 eksctl.tar.gz
drwxrwxr-x  3 ubuntu docker 4096 Apr 13 09:41 flask-app
ubuntu@ip-172-31-5-122:~$ |

```

Step 2: Making Changes and Creating a Branch

- Navigate into the cloned repository:

```
cd Experiment2
```

```
ls -la
```

```

ubuntu@ip-172-31-5-122:~$ cd Experiment2
ubuntu@ip-172-31-5-122:~/Experiment2$ ls -la
total 16
drwxrwxr-x  3 ubuntu ubuntu 4096 May 13 06:54 .
drwxr-x--- 11 ubuntu ubuntu 4096 May 13 06:54 ..
drwxrwxr-x  8 ubuntu ubuntu 4096 May 13 06:54 .git
-rw-rw-r--  1 ubuntu ubuntu  13  May 13 06:54 README.md
ubuntu@ip-172-31-5-122:~/Experiment2$ |

```

- Create a new text file named "example.txt" using a text editor.
- Add some content to the "example.txt" file.

```
vi example.txt
```

- Save the file and return to the command line.

```
cat example.txt
```

```
ubuntu@ip-172-31-5-122:~/Experiment2$ vi example.txt
ubuntu@ip-172-31-5-122:~/Experiment2$ cat example.txt
Welcome to NHCE
ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

- Check the status of the repository:

```
git status
```

```
ubuntu@ip-172-31-5-122:~/Experiment2$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  example.txt

nothing added to commit but untracked files present (use "git add" to track)
ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

- Stage the changes for commit:

```
git add example.txt
```

```
git status
```

```
ubuntu@ip-172-31-5-122:~/Experiment2$ git add example.txt
ubuntu@ip-172-31-5-122:~/Experiment2$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
  new file:   example.txt

ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

- Commit the changes with a descriptive message:

```
git commit -m "Added content to example.txt"
```

```

ubuntu@ip-172-31-5-122:~/Experiment2$ git commit -m "Added content to example.txt"
[main 78f544b] Added content to example.txt
Committer: Ubuntu <ubuntu@ip-172-31-5-122.ap-south-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 1 insertion(+)
create mode 100644 example.txt
ubuntu@ip-172-31-5-122:~/Experiment2$ |

```

git status

```

ubuntu@ip-172-31-5-122:~/Experiment2$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
ubuntu@ip-172-31-5-122:~/Experiment2$ |

```

- Create a new branch named feature:

git branch

git branch feature

git branch

```

ubuntu@ip-172-31-5-122:~/Experiment2$ git branch feature
ubuntu@ip-172-31-5-122:~/Experiment2$ git branch
  feature
* main
ubuntu@ip-172-31-5-122:~/Experiment2$ |

```

- Switch to the feature branch:

git checkout feature

git branch

```
ubuntu@ip-172-31-5-122:~/Experiment2$ git checkout feature
Switched to branch 'feature'
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$ git branch
* feature
  main
ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

Step 3: Pushing Changes to GitHub

- Add Repository URL in a variable

```
git remote add origin <repository_url>
```

Note: Preferably SSH URL, not HTTP URL. Because GitHub no longer supports HTTP password authentication.

- Replace <repository_url> with the URL you copied from GitHub.

```
ubuntu@ip-172-31-5-122:~/Experiment2$ git remote add origin git@github.com:SandyDevOpsStuffs/Experiment2.git
error: remote origin already exists.
```

Note: It throws an error “remote origin already exists”. Because already we cloned the repo from GitHub. So no need to add it back to GitHub.

- Check or verify the current remote URL.

```
git remote -v
```

```
ubuntu@ip-172-31-5-122:~/Experiment2$ git remote -v
origin https://github.com/SandyDevOpsStuffs/Experiment2.git (fetch)
origin https://github.com/SandyDevOpsStuffs/Experiment2.git (push)
ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

Note: By default GitHub has taken HTTP URL itself.

- Change remote from HTTPS to SSH:

```
git remote set-url origin
git@github.com:SandyDevOpsStuffs/Experiment2.git
```

```
git remote -v
```



```
ubuntu@ip-172-31-5-122:~/Experiment2$ git remote set-url origin git@github.com:SandyDevOpsStuffs/Experiment2.git
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$ git remote -v
origin  git@github.com:SandyDevOpsStuffs/Experiment2.git (fetch)
origin  git@github.com:SandyDevOpsStuffs/Experiment2.git (push)
ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

- Test or verify the SSH connection to GitHub.

```
ssh -T git@github.com
```

```
ubuntu@ip-172-31-5-122:~/Experiment2$ ssh -T git@github.com
Hi SandyDevOpsStuffs! You've successfully authenticated, but GitHub does not provide shell access.
ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

- Push the feature branch to GitHub using SSH:

```
git push origin feature
```

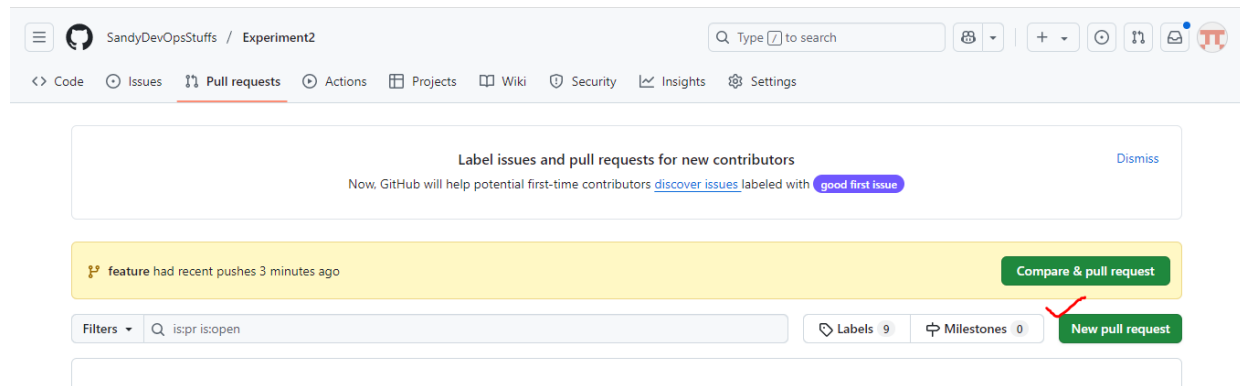
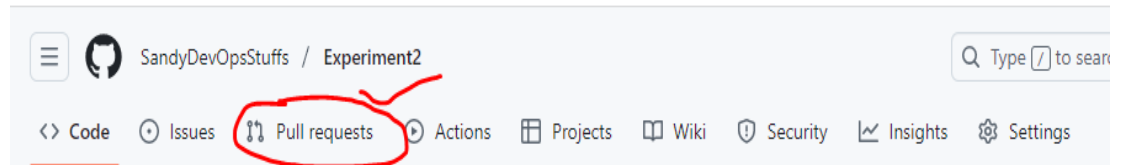
```
ubuntu@ip-172-31-5-122:~/Experiment2$ git push origin feature
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 327 bytes | 327.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature' on GitHub by visiting:
remote:   https://github.com/SandyDevOpsStuffs/Experiment2/pull/new/feature
remote:
To github.com:SandyDevOpsStuffs/Experiment2.git
 * [new branch]      feature -> feature
ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

- Check your GitHub repository to confirm that the new branch feature is available.

The screenshot displays the GitHub interface for the repository 'SandyDevOpsStuffs / Experiment2'. At the top, a yellow banner indicates that the 'feature' branch has recent pushes. Below this, the repository's main navigation bar shows the 'Code' tab selected. A 'Switch branches/tags' dropdown menu is open, showing a list of branches: 'main' (selected) and 'feature' (highlighted with a red checkmark). The right sidebar provides repository statistics, including 0 stars, 1 watching, and 0 forks. The 'About' section on the right states 'No description, website, or topics provided.'

Step 4: Collaborating through Pull Requests

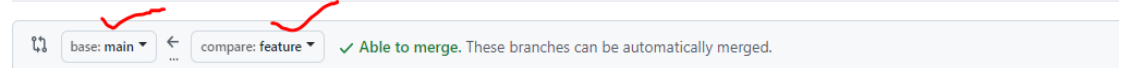
- Create a pull request on GitHub:
 - Go to the repository on GitHub.
 - Click on "Pull Requests" and then "New Pull Request."



- Choose the base branch (usually main or master) and the compare branch (feature).

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#) or [learn more about diff comparisons](#).



- Review the changes and click "Create Pull Request."

base: main ← compare: feature ✓ Able to merge. These branches can be automatically merged.

Discuss and review the changes in this comparison with others. [Learn about pull requests](#) [Create pull request](#)

1 commit 1 file changed 1 contributor

Commits on May 13, 2025

Added content to example.txt
Ubuntu committed 26 minutes ago

Showing 1 changed file with 1 addition and 0 deletions. Split Unified

example.txt

```
... @@ -0,0 +1 @@
1 + Welcome to NHCE
```

SandyDevOpsStuffs / Experiment2

Type to search

<> Code Issues Pull requests 1 Actions Projects Wiki Security Insights Settings

Added content to example.txt #1

Open SandyDevOpsStuffs wants to merge 1 commit into main from feature

Conversation 0 Commits 1 Checks 0 Files changed 1 +1 -0

SandyDevOpsStuffs commented now

No description provided.

Added content to example.txt 78f544b

No conflicts with base branch
Merging can be performed automatically.

Merge pull request You can also merge this with the command line. [View command line instructions](#).

Reviews
No reviews
Still in progress? [Convert to draft](#)

Assignees
No one—[assign yourself](#)

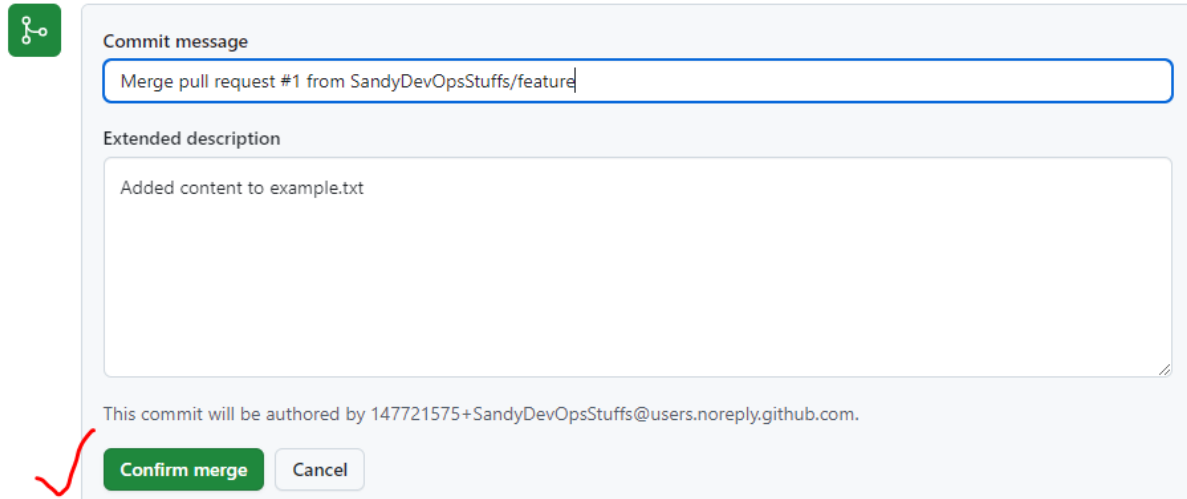
Labels
None yet

Projects
None yet

- Review and merge the pull request:
 - Add a title and description for the pull request.
 - Assign reviewers if needed.
 - Once the pull request is approved, merge it into the base branch.

No conflicts with base branch
Merging can be performed automatically.

Merge pull request You can also merge this with the command line. [View command line instructions](#).



A screenshot of the GitHub 'Commit message' dialog. The 'Commit message' field contains the text 'Merge pull request #1 from SandyDevOpsStuffs/feature'. The 'Extended description' field contains the text 'Added content to example.txt'. Below the fields, a message states 'This commit will be authored by 147721575+SandyDevOpsStuffs@users.noreply.github.com.' At the bottom, there are two buttons: 'Confirm merge' (highlighted with a red checkmark) and 'Cancel'.

Commit message

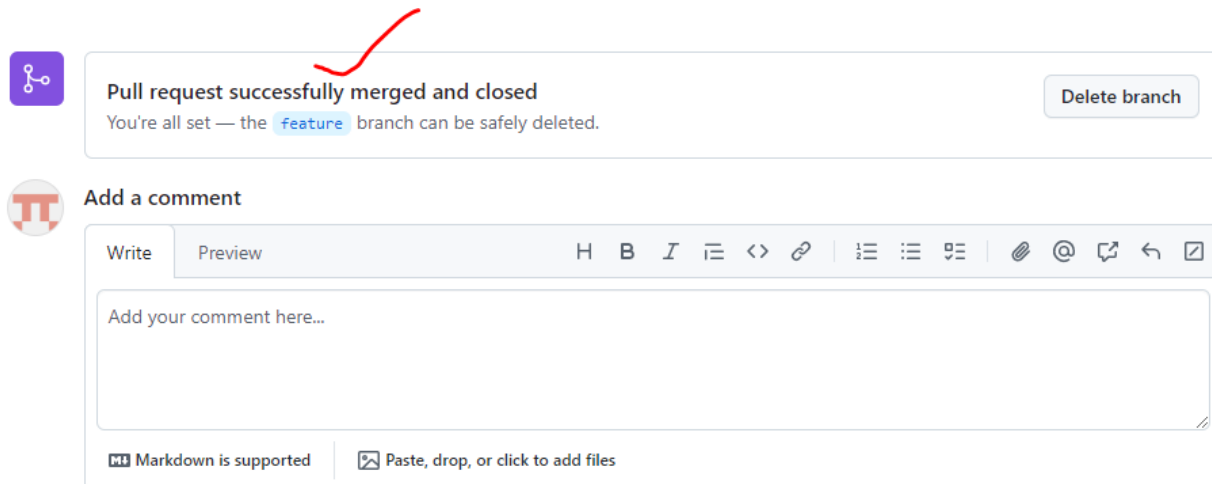
Merge pull request #1 from SandyDevOpsStuffs/feature

Extended description

Added content to example.txt

This commit will be authored by 147721575+SandyDevOpsStuffs@users.noreply.github.com.

Confirm merge Cancel



A screenshot of a GitHub notification box titled 'Pull request successfully merged and closed'. The text inside says 'You're all set — the feature branch can be safely deleted.' and there is a 'Delete branch' button. Below this is a section for 'Add a comment' with a 'Write' tab selected, a rich text editor with a toolbar, and a text area for the comment. The notification box is marked with a red checkmark.












Pull request successfully merged and closed

You're all set — the feature branch can be safely deleted.

Delete branch

Add a comment

Write Preview

H B I           

Add your comment here...

Markdown is supported Paste, drop, or click to add files

Step 5: Resolving Merge Conflicts

To simulate a conflict, repeat the following:

1. **Create two branches** from main:

```
git branch
```

```
git checkout main
```

```
git branch
```

```

ubuntu@ip-172-31-5-122:~/Experiment2$ git branch
* feature
  main
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$ git branch
  feature
* main
ubuntu@ip-172-31-5-122:~/Experiment2$

```

```
git checkout -b feature-1
```

```
git branch
```

```

ubuntu@ip-172-31-5-122:~/Experiment2$ git checkout -b feature-1
Switched to a new branch 'feature-1'
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$ git branch
  feature
* feature-1
  main
ubuntu@ip-172-31-5-122:~/Experiment2$ |

```

```
cat example.txt
```

```

ubuntu@ip-172-31-5-122:~/Experiment2$ cat example.txt
Welcome to NHCE
ubuntu@ip-172-31-5-122:~/Experiment2$

```

```
echo "Line from feature-1" > example.txt
```

```
cat example.txt
```

```

ubuntu@ip-172-31-5-122:~/Experiment2$ echo "Line from feature-1" > example.txt
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$ cat example.txt
Line from feature-1
ubuntu@ip-172-31-5-122:~/Experiment2$ |

```

```
git status
```

```

ubuntu@ip-172-31-5-122:~/Experiment2$ git status
On branch feature-1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   example.txt

no changes added to commit (use "git add" and/or "git commit -a")
ubuntu@ip-172-31-5-122:~/Experiment2$ |

```

git add example.txt

git status

```

ubuntu@ip-172-31-5-122:~/Experiment2$ git add example.txt
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$ git status
On branch feature-1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   example.txt

ubuntu@ip-172-31-5-122:~/Experiment2$ |

```

git commit -m "Updated example.txt in feature-1"

git status

```

ubuntu@ip-172-31-5-122:~/Experiment2$ git commit -m "Updated example.txt in feature-1"
[feature-1 c44d57d] Updated example.txt in feature-1
  Committer: Ubuntu <ubuntu@ip-172-31-5-122.ap-south-1.compute.internal>
  Your name and email address were configured automatically based
  on your username and hostname. Please check that they are accurate.
  You can suppress this message by setting them explicitly. Run the
  following command and follow the instructions in your editor to edit
  your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 1 insertion(+), 1 deletion(-)
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$ git status
On branch feature-1
nothing to commit, working tree clean
ubuntu@ip-172-31-5-122:~/Experiment2$ |

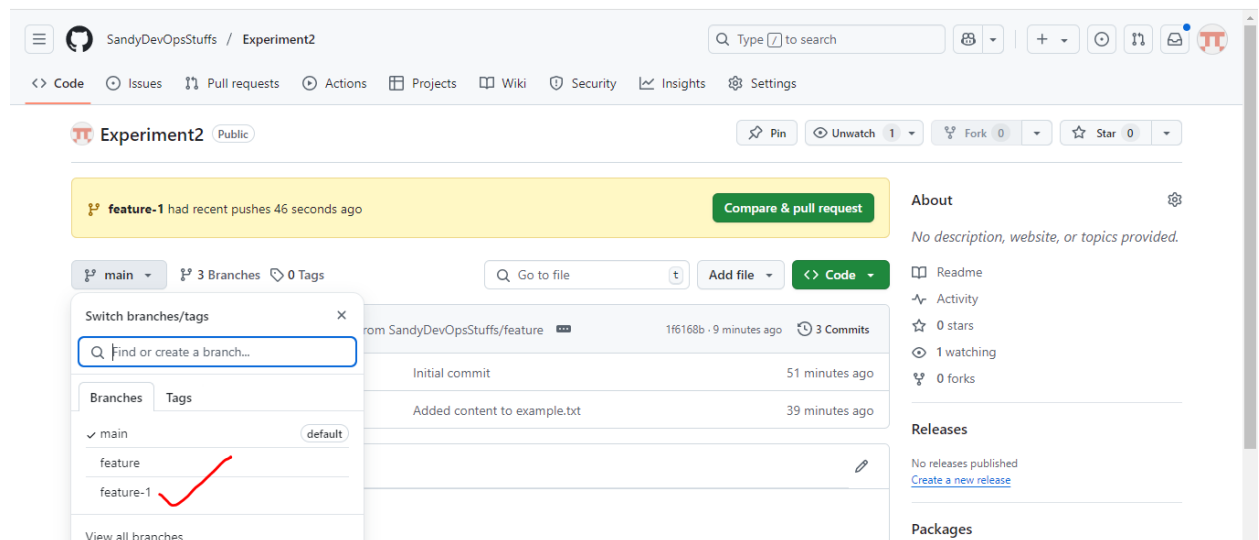
```

git push origin feature-1

git status

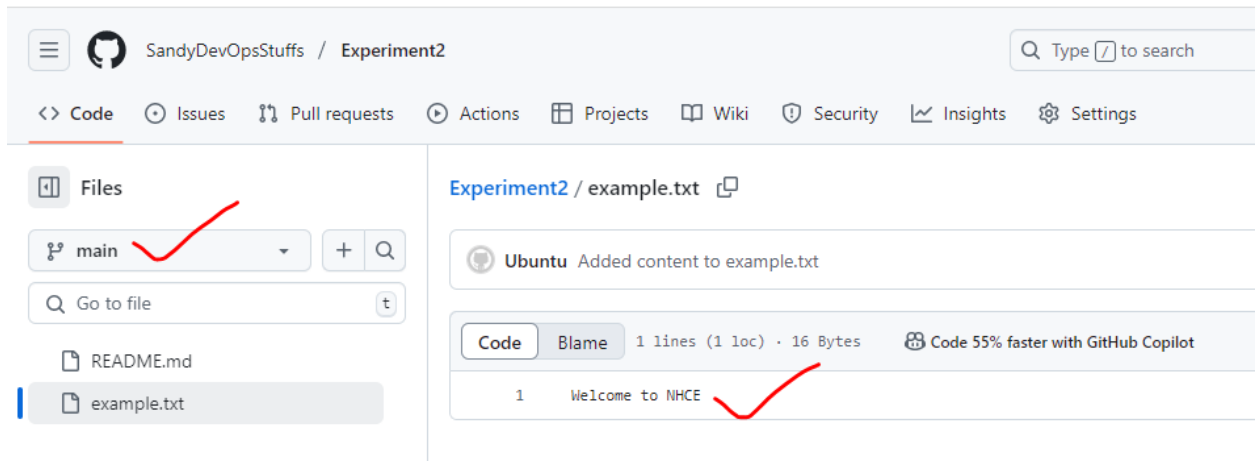
```
ubuntu@ip-172-31-5-122:~/Experiment2$ git push origin feature-1
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 332 bytes | 332.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature-1' on GitHub by visiting:
remote:   https://github.com/SandyDevOpsStuffs/Experiment2/pull/new/feature-1
remote:
To github.com:SandyDevOpsStuffs/Experiment2.git
 * [new branch]      feature-1 -> feature-1
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$ git status
On branch feature-1
nothing to commit, working tree clean
ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

- Confirm the feature-1 branch creation in remote repo in GitHub.



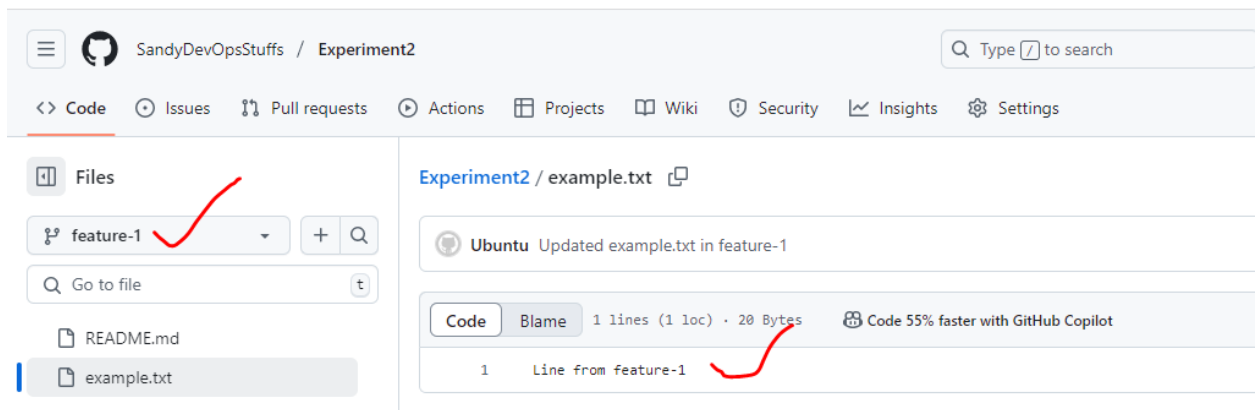
Being in the main branch check the file content which should be as follows:

Welcome to NHCE



Now change the branch from main to feature-1 and check the file content which should be as follows:

Line from feature-1



```
git checkout main
```

```
git branch
```



```
ubuntu@ip-172-31-5-122:~/Experiment2$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$ git branch
  feature
  feature-1
* main
ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

```
git checkout -b feature-2
```

```
git branch
```

```
ubuntu@ip-172-31-5-122:~/Experiment2$ git checkout -b feature-2
Switched to a new branch 'feature-2'
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$ git branch
  feature
  feature-1
* feature-2
  main
ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

```
cat example.txt
```

```
ubuntu@ip-172-31-5-122:~/Experiment2$ cat example.txt
Welcome to NHCE
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

Note: Even though when I am already in feature-2 branch it still shows the content of the file from main branch. Because from the feature-2 branch I have not modified the file yet.

```
echo "Line from feature-2" > example.txt
```

```
cat example.txt
```

```
ubuntu@ip-172-31-5-122:~/Experiment2$ echo "Line from feature-2" > example.txt
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$ cat example.txt
Line from feature-2
ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

git status

```
ubuntu@ip-172-31-5-122:~/Experiment2$ git status
On branch feature-2
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   example.txt

no changes added to commit (use "git add" and/or "git commit -a")
ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

git add example.txt

git status

```
ubuntu@ip-172-31-5-122:~/Experiment2$ git add example.txt
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$ git status
On branch feature-2
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   example.txt

ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

git commit -m "Updated example.txt in feature-2"

git status

```

ubuntu@ip-172-31-5-122:~/Experiment2$ git commit -am "Updated example.txt in feature-2"
[feature-2 b11fb1f] Updated example.txt in feature-2
Committer: Ubuntu <ubuntu@ip-172-31-5-122.ap-south-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 1 insertion(+), 1 deletion(-)
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$ git status
On branch feature-2
nothing to commit, working tree clean
ubuntu@ip-172-31-5-122:~/Experiment2$ |

```

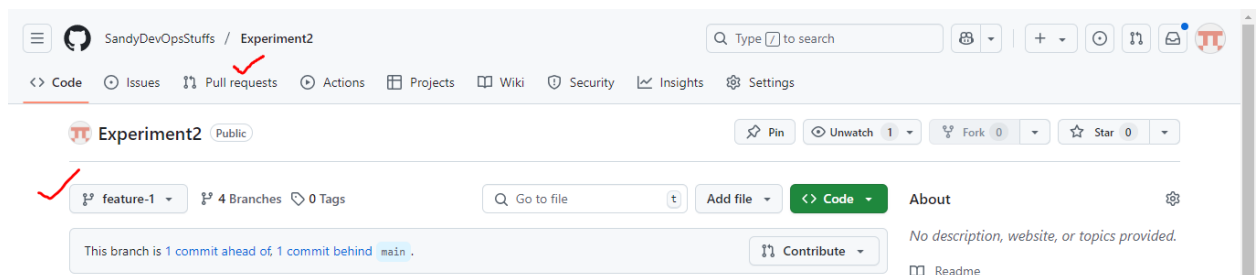
git push origin feature-2

```

ubuntu@ip-172-31-5-122:~/Experiment2$ git push origin feature-2
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 332 bytes | 332.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature-2' on GitHub by visiting:
remote:   https://github.com/SandyDevOpsStuffs/Experiment2/pull/new/feature-2
remote:
To github.com:SandyDevOpsStuffs/Experiment2.git
 * [new branch]      feature-2 -> feature-2
ubuntu@ip-172-31-5-122:~/Experiment2$ |

```

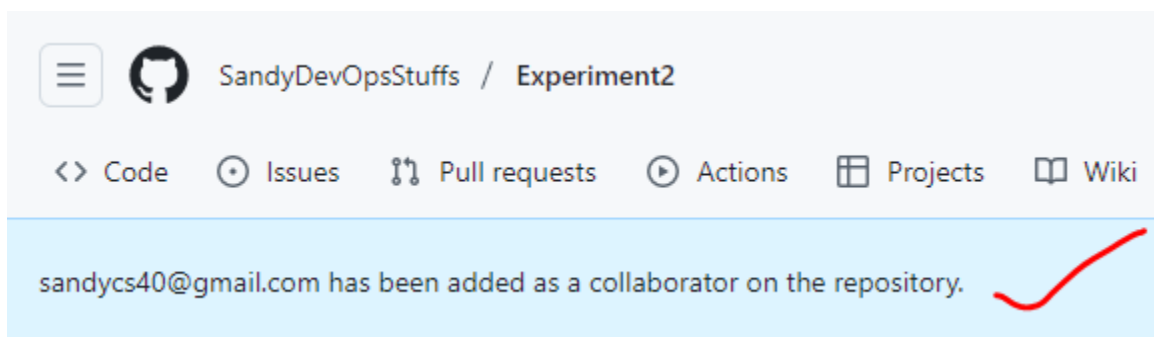
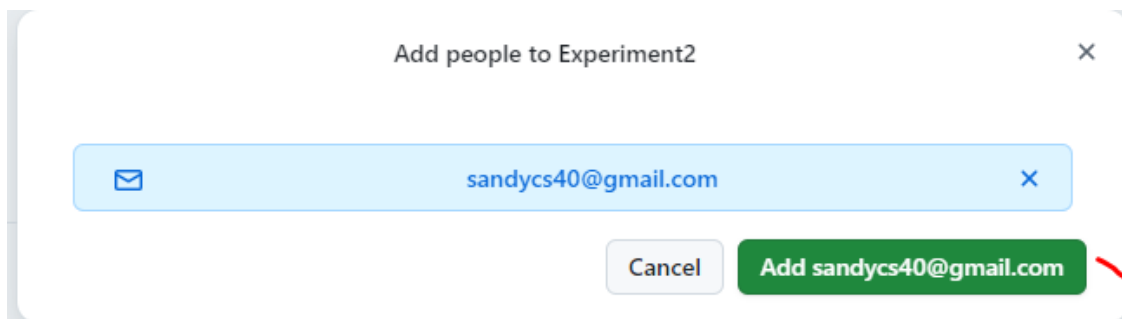
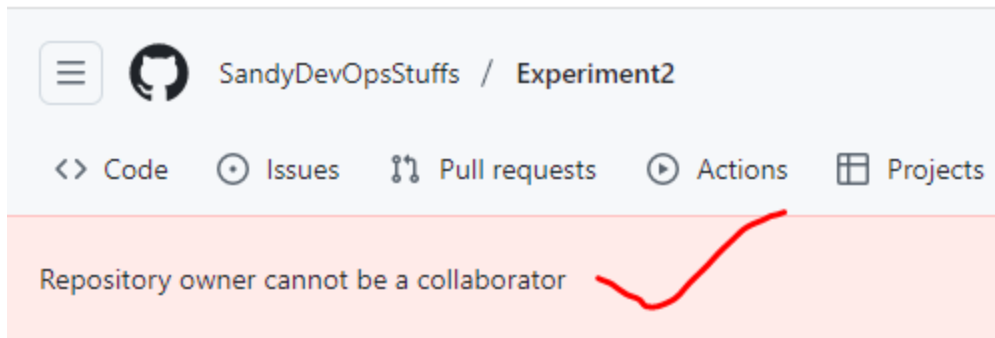
2. Navigate to the branch `feature-1` on GitHub by selecting it from the drop down, go to **Settings** of the repo, add **collaborators** by typing their mail IDs, inform them to accept the invitation sent from the GitHub to their mail IDs and create a **pull request (PR)** for `feature-1` by selecting the collaborators as **reviewers** and merge it on GitHub.



The screenshot shows the GitHub repository settings for 'Experiment2'. The 'Settings' tab is selected, indicated by a red checkmark. The left sidebar contains a menu with 'Collaborators' highlighted. The main content area is titled 'Collaborators and teams' and shows 'Public repository' status. Below this, a 'Direct access' box states '0 collaborators have access to this repository. Only you can contribute to this repository.' The 'Manage access' section below it displays a message 'You haven't invited any collaborators yet' with an 'Add people' button highlighted by a red checkmark. The left sidebar also includes sections for 'Access', 'Code and automation', and 'Security'.

The screenshot shows a dialog box titled 'Add people to Experiment2'. It prompts the user to 'Search by username, full name, or email'. A search bar contains the text 'sandycs40@gmail.com'. Below the search bar, a list of results shows an email icon next to 'sandycs40@gmail.com' with the text 'Invite to Experiment2'. At the bottom right, there are two buttons: 'Cancel' and 'Add to repository'.

Note: If you type your own mail ID with which you have created the GitHub account then you will get below error. So I have entered my other mail ID as a reviewer for this demo.




Manage access

Add people


☐ Select all


Type ▾

☐



SANDESH-D-MANOCHARYA
Awaiting SANDESH-D-MANOCHARYA's response

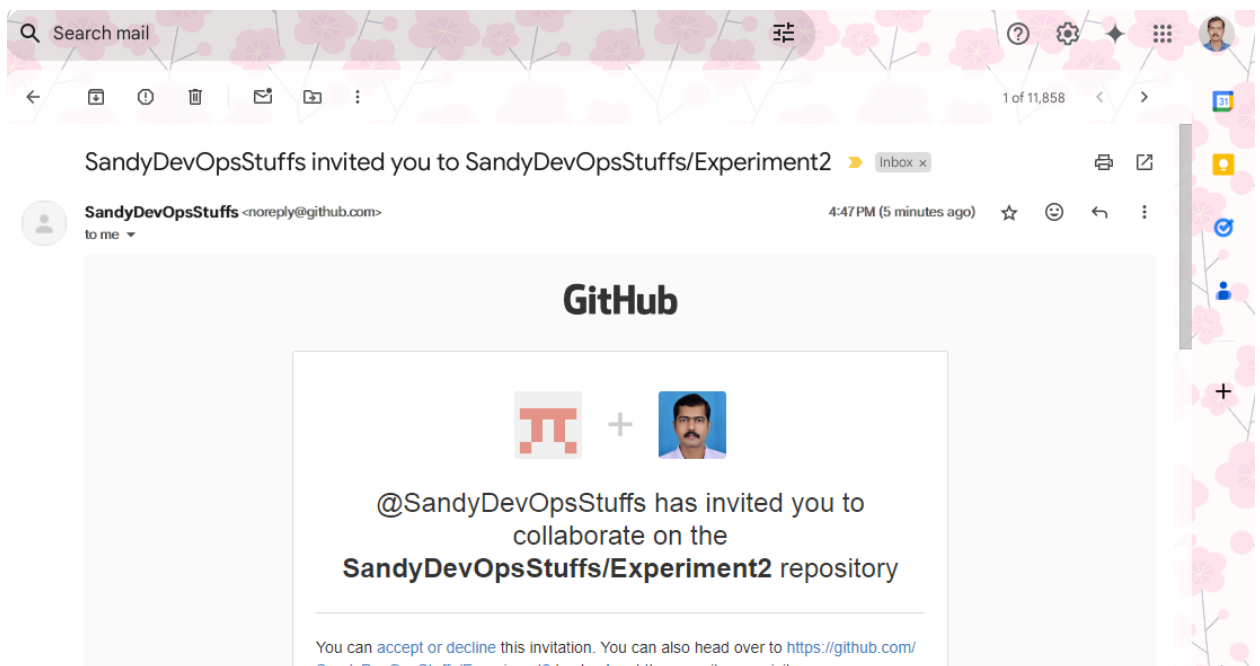
Pending Invite 



< Previous Next >

Note: Now you will see the collaborator status as “Pending”.

- Collaborator will receive a mail as below:





@SandyDevOpsStuffs has invited you to collaborate on the **SandyDevOpsStuffs/Experiment2** repository

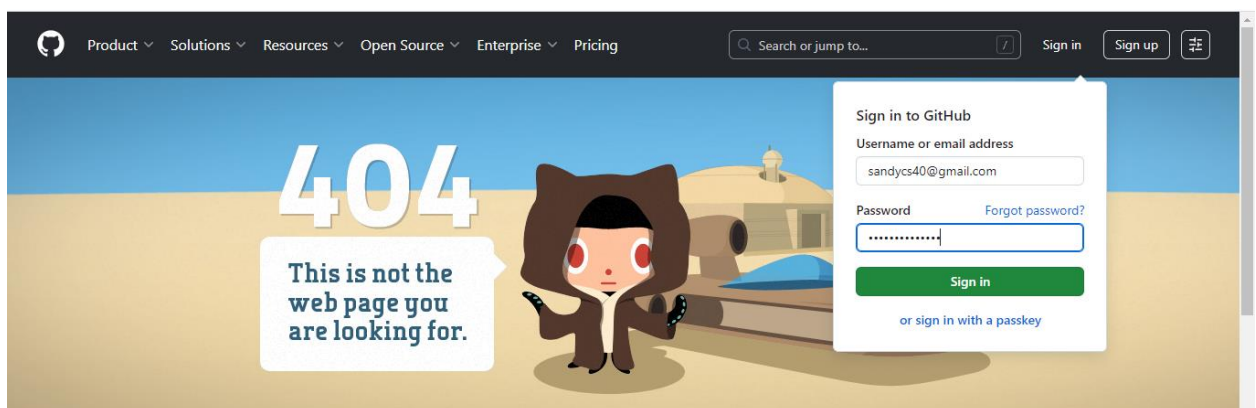
You can [accept or decline](#) this invitation. You can also head over to <https://github.com/SandyDevOpsStuffs/Experiment2> to check out the repository or visit [@SandyDevOpsStuffs](#) to learn a bit more about them.

This invitation will expire in 7 days.

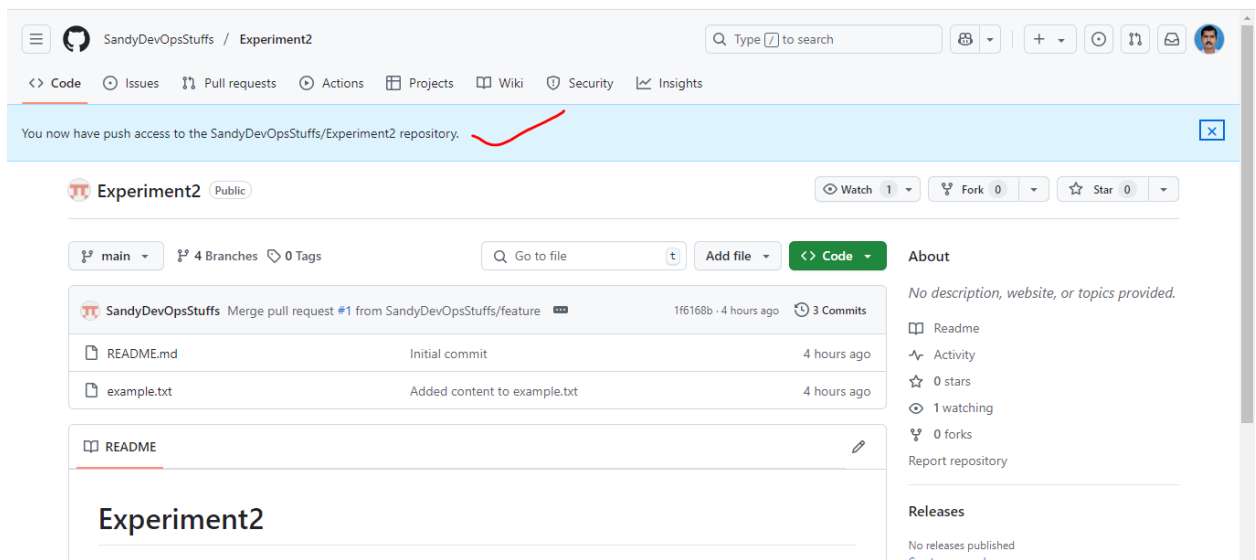
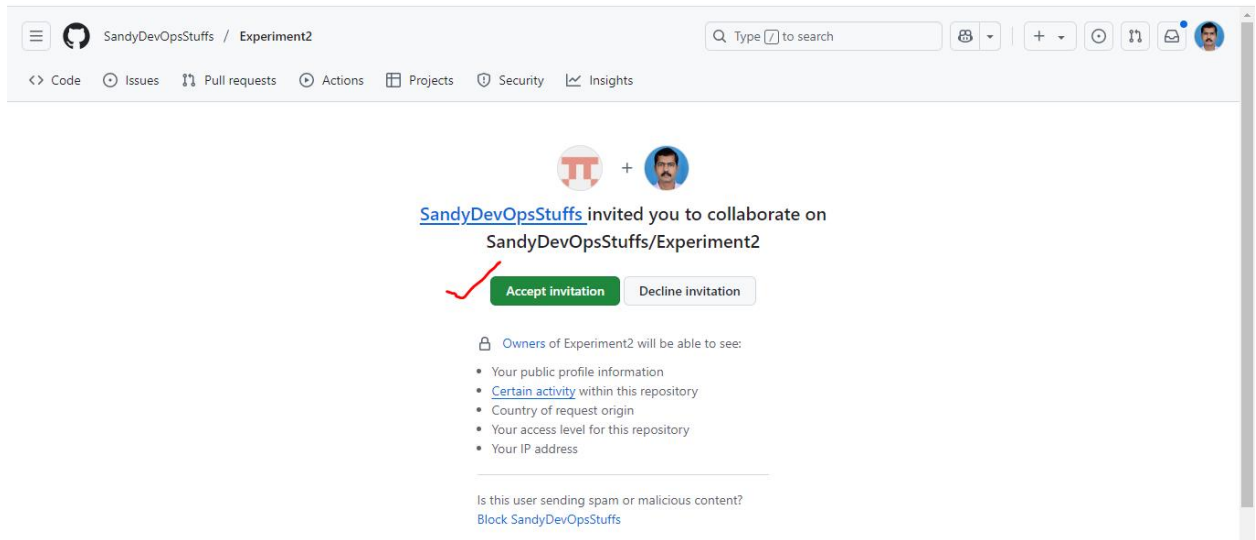
[View invitation](#)

- Collaborator should click on “View invitation” in his mail inbox.
- It will take him to GitHub Login page.

Note: It is better to open GitHub for collaborator in another browser.



- After login the collaborator should click on “Accept Invitation”.



- Now you go to your GitHub repo to create a PR. Now you will **not** see the collaborator status as “Pending”.


Manage access


Add people

☐ Select all

Type ▾


☐






**SANDESH-D-MANOCHARYA**
Collaborator



< Previous Next >

- Now create a PR by adding the collaborator as reviewer.

 SandyDevOpsStuffs / Experiment2

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

Label issues and pull requests for new contributors

Dismiss

Now, GitHub will help potential first-time contributors [discover issues](#) labeled with [good first issue](#)

Filters ▾


Labels 9

Milestones 0

New pull request

☐ 0 Open ✓ 1 Closed

Author ▾ Label ▾ Projects ▾ Milestones ▾ Reviews ▾ Assignee ▾ Sort ▾



There aren't any open pull requests.

You could search [all of GitHub](#) or try an [advanced search](#).

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#) or [learn more about diff comparisons](#).

base: main ← compare: feature-1 ✓ Able to merge. These branches can be automatically merged.

Discuss and review the changes in this comparison with others. [Learn about pull requests](#) [Create pull request](#)

1 commit 1 file changed 1 contributor

Commits on May 13, 2025

Updated example.txt in feature-1
Ubuntu committed 4 hours ago

Showing 1 changed file with 1 addition and 1 deletion. Split Unified

example.txt

```
... @@ -1 +1 @@  
1 - Welcome to NHCE  
1 + Line from feature-1
```

- Click on “Reviewers” on right side and you will be able to see a collaborator. Select that collaborator.

SandyDevOpsStuffs / Experiment2

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#). [Learn more about diff comparisons here](#).

base: main ← compare: feature-1 ✓ Able to merge. These branches can be automatically merged.

Add a title

Updated example.txt in feature-1

Add a description

Write Preview

Add your description here...

Reviewers

Request up to 15 reviewers

Type or choose a user

SANDESH-D-MANOCHARYA

Labels

None yet

Projects

None yet

Milestone

- Click on “Create pull request”.

base: main ← compare: feature-1 ✓ Able to merge. These branches can be automatically merged.

Add a title

Updated example.txt in feature-1

Write

Preview

H B I

Add your description here...

Markdown is supported

Paste, drop, or click to add files

Create pull request

Reviewers

SANDESH-D-MANOCHARYA

Assignees

No one—[assign yourself](#)

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Use [Closing keywords](#) in the description to automatically close issues

Helpful resources

[GitHub Community Guidelines](#)

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

SandyDevOpsStuffs / Experiment2

Type to search

<> Code Issues Pull requests 1 Actions Projects Wiki Security Insights Settings

Updated example.txt in feature-1 #2

Open

SandyDevOpsStuffs wants to merge 1 commit into main from feature-1

Conversation 0 Commits 1 Checks 0 Files changed 1

SandyDevOpsStuffs commented now

No description provided.

Updated example.txt in feature-1

c44d57d

SandyDevOpsStuffs requested a review from SANDESH-D-MANOCHARYA now

Review requested

Review has been requested on this pull request. It is not required to merge. [Learn more about requesting a pull request](#)

Reviewers

SANDESH-D-MANOCHARYA

Still in progress? [Convert to draft](#)

Assignees

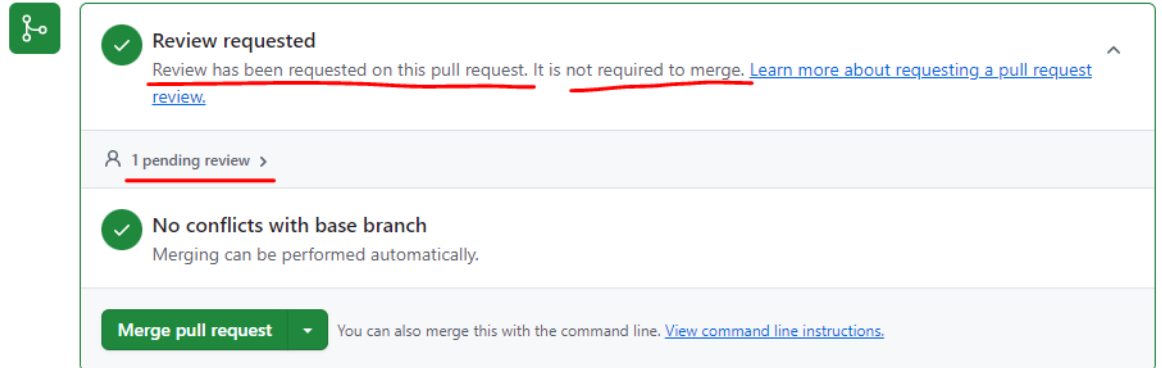
No one—[assign yourself](#)

Labels

None yet

Projects

None yet



A screenshot of a GitHub Pull Request status summary. It features a green checkmark icon and the text "Review requested". Below this, it states "Review has been requested on this pull request. It is not required to merge." followed by a link "Learn more about requesting a pull request review." A second line shows "1 pending review" with a person icon. Below that, another green checkmark icon is followed by "No conflicts with base branch" and "Merging can be performed automatically." At the bottom, there is a green button labeled "Merge pull request" and a link "View command line instructions."

Review requested

Review has been requested on this pull request. It is not required to merge. [Learn more about requesting a pull request review.](#)

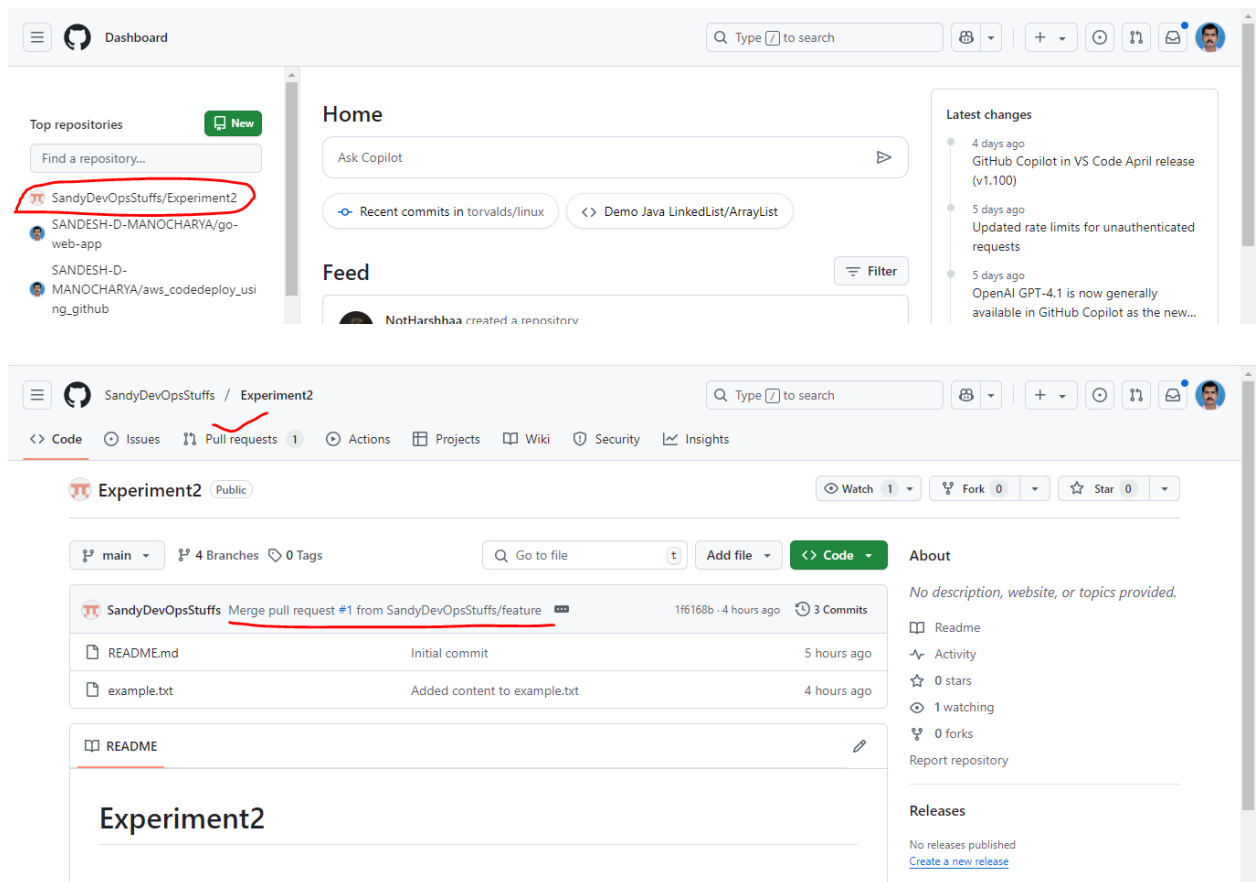
1 pending review >

No conflicts with base branch

Merging can be performed automatically.

Merge pull request You can also merge this with the command line. [View command line instructions.](#)

- Now collaborator or reviewer has to review the code and approve the PR.
- Collaborator will see a repo in left panel.



A screenshot of the GitHub interface. The top section shows the "Dashboard" with a search bar and navigation icons. The left sidebar lists "Top repositories" with a search bar and a list of repositories, including "SandyDevOpsStuffs/Experiment2" which is highlighted with a red circle. The main content area shows the "Home" page with a search bar, a "Ask Copilot" button, and a "Feed" section. The bottom section shows the "Experiment2" repository page, which includes a "Merge pull request #1 from SandyDevOpsStuffs/feature" button, a list of files (README.md, example.txt), and a "README" section with the title "Experiment2".

Dashboard

Find a repository...

SandyDevOpsStuffs/Experiment2

Home

Ask Copilot

Recent commits in torvalds/linux

Demo Java LinkedList/ArrayList

Feed

NotHarshhaa created a repository

Latest changes

- 4 days ago GitHub Copilot in VS Code April release (v1.100)
- 5 days ago Updated rate limits for unauthenticated requests
- 5 days ago OpenAI GPT-4.1 is now generally available in GitHub Copilot as the new...

Experiment2

main 4 Branches 0 Tags

Go to file Add file Code

SandyDevOpsStuffs Merge pull request #1 from SandyDevOpsStuffs/feature 1f6168b · 4 hours ago 3 Commits

- README.md Initial commit 5 hours ago
- example.txt Added content to example.txt 4 hours ago

README

Experiment2

About

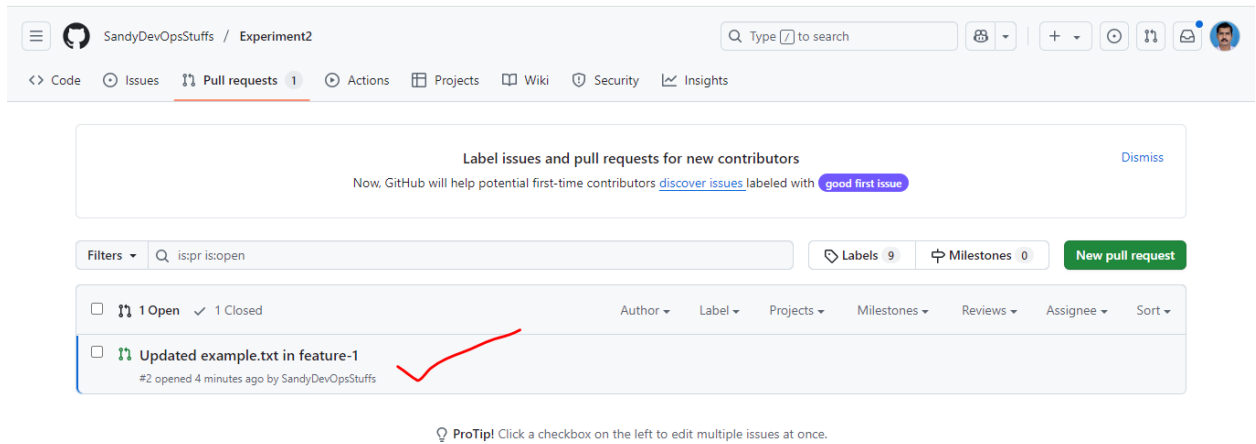
No description, website, or topics provided.

- Readme
- Activity
- 0 stars
- 1 watching
- 0 forks
- Report repository

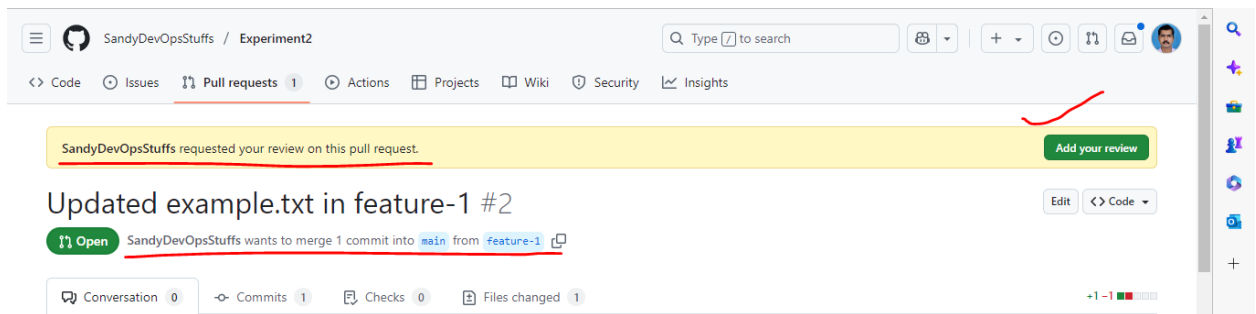
Releases

No releases published

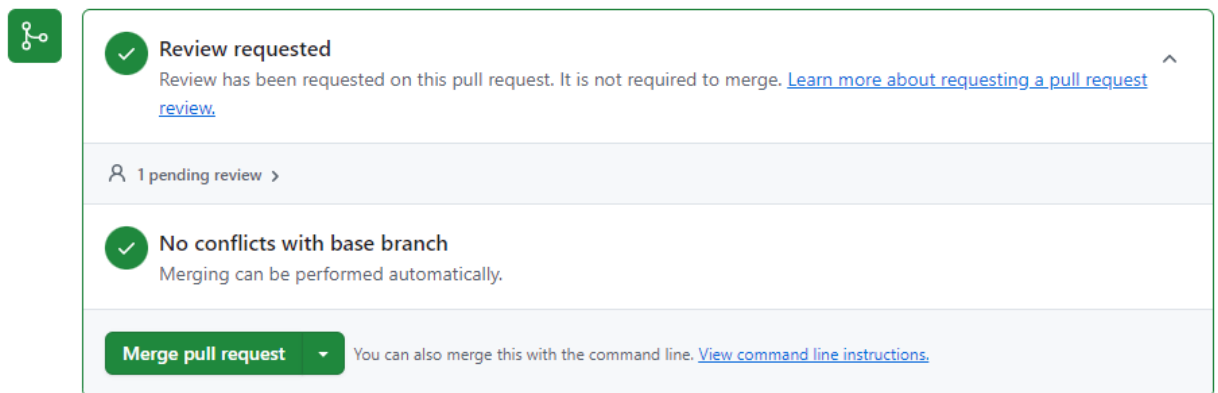
[Create a new release](#)



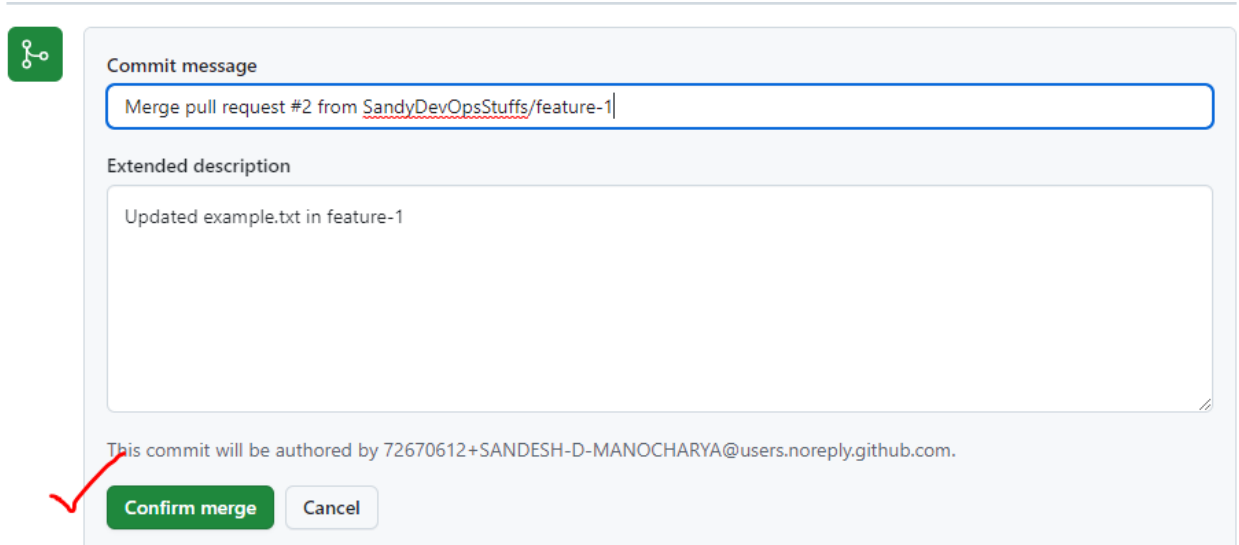
- Reviewer will click on the PR and he can add his reviews.



- Reviewer has to scroll down and click on “Merge pull request”.



- Reviewer has to click on “Confirm merge”.



Commit message

Merge pull request #2 from SandyDevOpsStuffs/feature-1

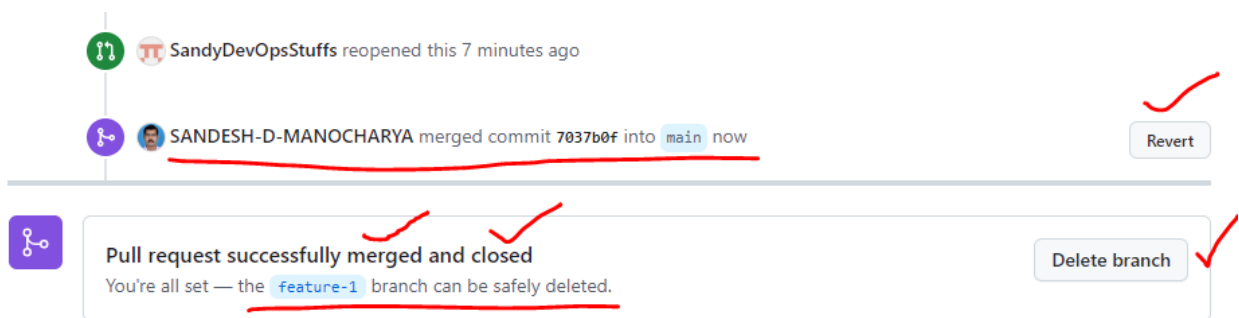
Extended description

Updated example.txt in feature-1

This commit will be authored by 72670612+SANDESH-D-MANOCHARYA@users.noreply.github.com.

☒

- Both you and the reviewer will get below same screen on your respective browsers:



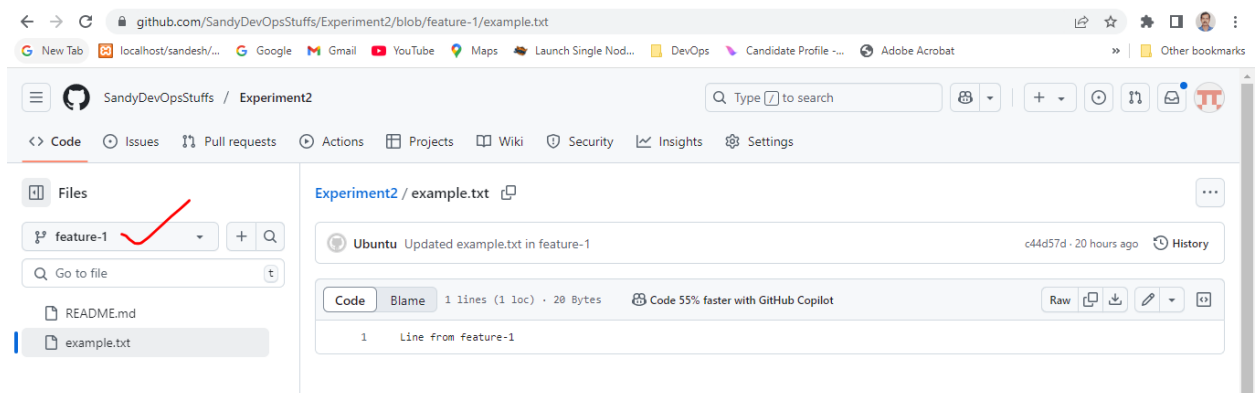
SandyDevOpsStuffs reopened this 7 minutes ago

SANDESH-D-MANOCHARYA merged commit 7037b0f into main now

Pull request successfully merged and closed

You're all set — the feature-1 branch can be safely deleted.

- Now you go to your GitHub repo, go to `feature-1` branch and see the content of `example.txt`. Similarly go to the branch `main` and see the content of `example.txt`.



github.com/SandyDevOpsStuffs/Experiment2/blob/feature-1/example.txt

SandyDevOpsStuffs / Experiment2

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Files

feature-1 ☒ +

Go to file

README.md

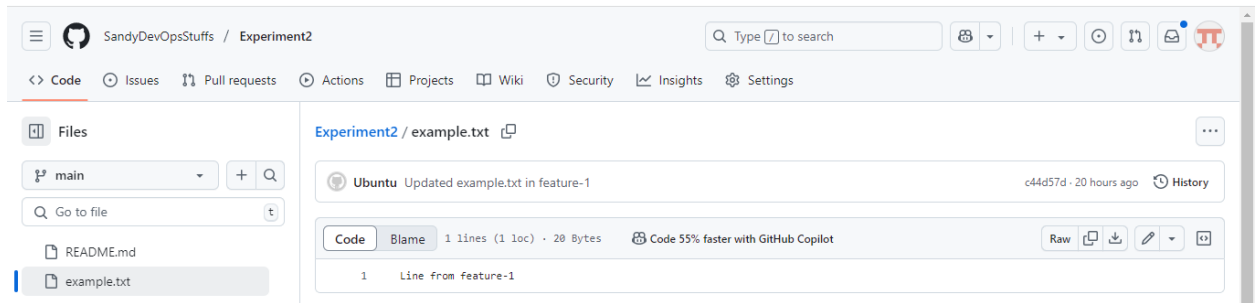
example.txt

Experiment2 / example.txt

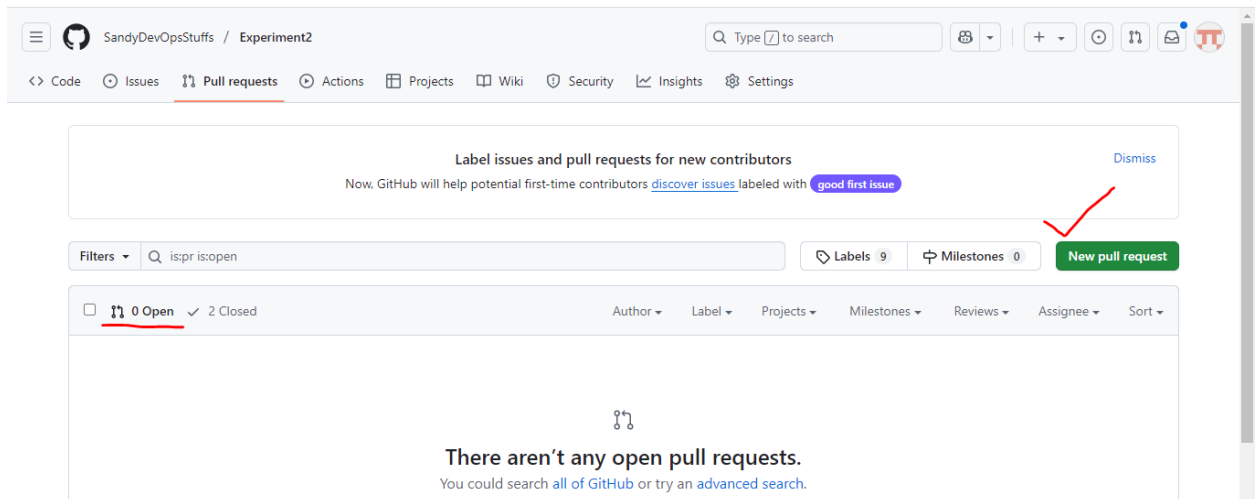
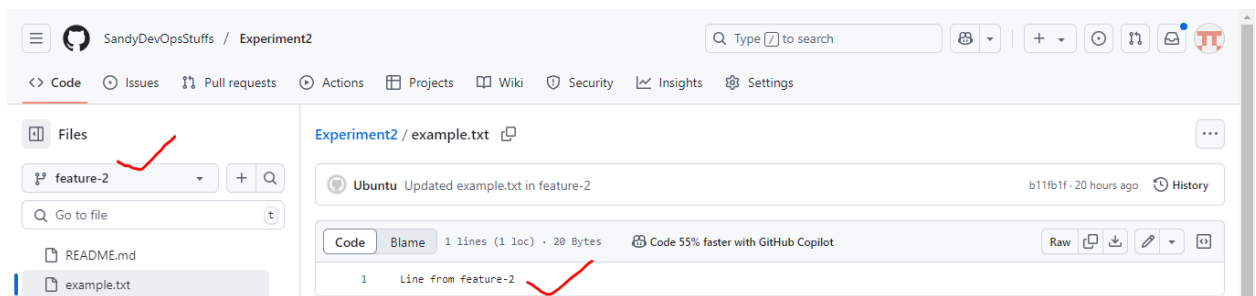
Ubuntu Updated example.txt in feature-1 c44d57d · 20 hours ago History

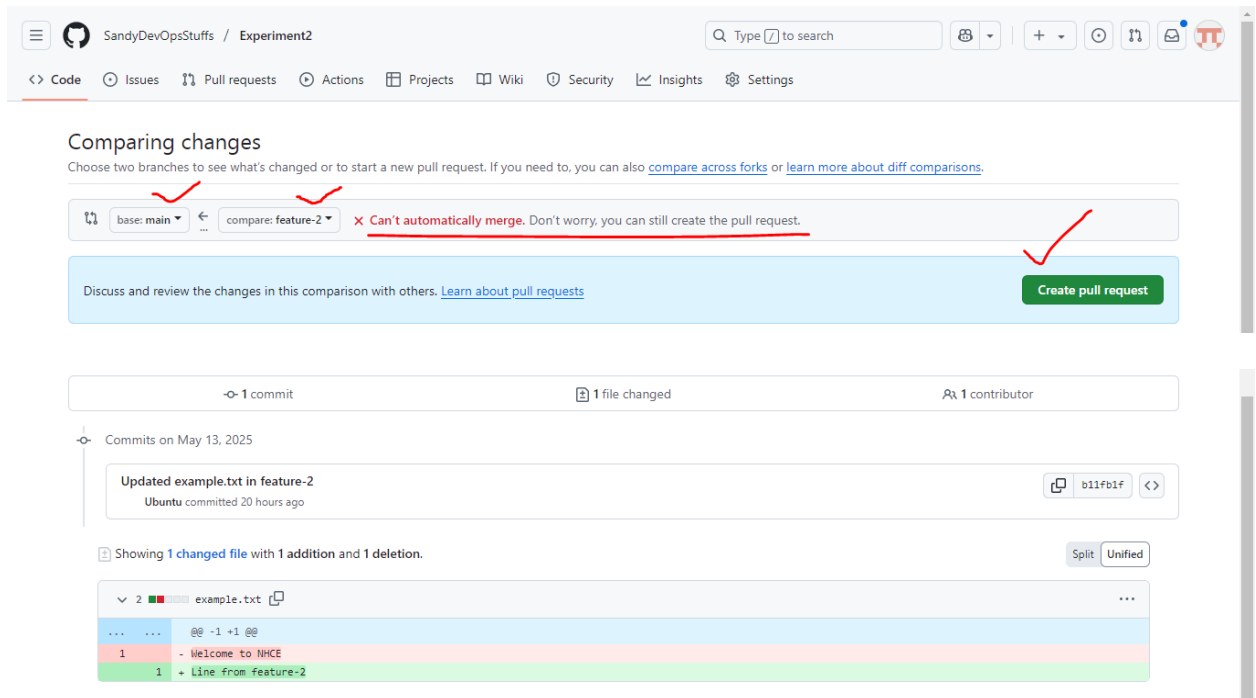
Code Blame 1 lines (1 loc) · 20 Bytes Code 55% faster with GitHub Copilot Raw Download Edit

1 Line from feature-1



3. Create a pull request for feature-2 – it will show a conflict.





- Scroll down and click on “Create pull request” and select a reviewer from the right side panel.

Add a description

Write Preview

H B I

Add your description here...

Markdown is supported Paste, drop, or click to add files

Create pull request

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Updated example.txt in feature-2 #3

Edit <> Code

[Open](#) SandyDevOpsStuffs wants to merge 1 commit into `main` from `feature-2`

Conversation 0 Commits 1 Checks 0 Files changed 1 +1 -1



SandyDevOpsStuffs commented 2 minutes ago

Owner ...

No description provided.



Updated example.txt in feature-2

b11fb1f

SandyDevOpsStuffs requested a review from SANDESH-D-MANOCHARYA 1 minute ago

Reviewers

SANDESH-D-MANOCHARYA

Still in progress? [Convert to draft](#)

Assignees

No one—[assign yourself](#)

Labels

- Scroll down to see the error message of conflict.



This branch has conflicts that must be resolved

[Resolve conflicts](#)

Use the [web editor](#) or the command line to resolve conflicts before continuing.

example.txt

Merge pull request



You can also merge this with the command line. [View command line instructions.](#)

4. Resolve conflict locally:

```
git checkout feature-2
```

```
git branch
```

```
ubuntu@ip-172-31-5-122:~/Experiment2$ git branch
feature
feature-1
* feature-2
main
ubuntu@ip-172-31-5-122:~/Experiment2$
```

```
cat example.txt
```

```
ubuntu@ip-172-31-5-122:~/Experiment2$ cat example.txt
Line from feature-2
```

```
git pull origin main # triggers merge conflict
```

```
ubuntu@ip-172-31-5-122:~/Experiment2$ git pull origin main
remote: Enumerating objects: 2, done.
remote: Counting objects: 100% (2/2), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (2/2), 1.78 KiB | 1.78 MiB/s, done.
From github.com:SandyDevOpsStuffs/Experiment2
 * branch          main      -> FETCH_HEAD
    7176ba2..7037b0f  main      -> origin/main
hint: You have divergent branches and need to specify how to reconcile them.
hint: You can do so by running one of the following commands sometime before
hint: your next pull:
hint:
hint:   git config pull.rebase false  # merge
hint:   git config pull.rebase true   # rebase
hint:   git config pull.ff only       # fast-forward only
hint:
hint: You can replace "git config" with "git config --global" to set a default
hint: preference for all repositories. You can also pass --rebase, --no-rebase,
hint: or --ff-only on the command line to override the configured default per
hint: invocation.
fatal: Need to specify how to reconcile divergent branches. ✓
ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

But this will not add conflict marks to the file `example.txt`. Because you are on `feature-2` and you ran:

```
git pull origin main
```

But Git **didn't know how to handle divergent branches**, so it aborted the merge.

That's why you **didn't get conflict markers** (<<<<<<) in `example.txt`.

So run the following command.

```
git pull --no-rebase origin main
```

```
ubuntu@ip-172-31-5-122:~/Experiment2$ git pull --no-rebase origin main
From github.com:SandyDevOpsStuffs/Experiment2
 * branch          main      -> FETCH_HEAD
Auto-merging example.txt
CONFLICT (content): Merge conflict in example.txt
Automatic merge failed; fix conflicts and then commit the result.
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

```
cat example.txt
```

```
ubuntu@ip-172-31-5-122:~/Experiment2$ cat example.txt
<<<<<<< HEAD
Line from feature-2
=====
Line from feature-1
>>>>>>> 7037b0fffb95137d3db043f42114473a03d252cd
ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

5. Open `example.txt` and resolve the conflict:

```
<<<<<<< HEAD
Line from feature-2
=====
Line from feature-1
>>>>>>> main
```

Note: When Git cannot automatically merge changes, it marks the conflict in the file as shown above.

This format means:

Marker	Meaning
<<<<<<< HEAD	This is your current branch (feature-2)
=====	Separator between conflicting changes
>>>>>>> main	This is the incoming change from the branch you pulled (main)

Why the Conflict?

Because **`example.txt`** was changed in both branches:

- feature-1 added: `Line from feature-1`
- feature-2 added: `Line from feature-2`

Git doesn't know which one to keep, so **you decide**.

What Should You Do?

You **manually edit the file** and choose what makes sense:

Option 1: Combine both lines

Line from both feature-1 and feature-2

This is what we usually do in collaborative teams — preserve both changes.

Option 2: Choose one version

Keep only feature-1's version:

Line from feature-1

Or only feature-2's version:

Line from feature-2

Note: I have file content as follows:

```
<<<<<<< HEAD
Line from feature-2
=====
Line from feature-1
>>>>>>> 7037b0fffb95137d3db043f42114473a03d252cd
```

Instead of:

```
<<<<<<< HEAD
Line from feature-2
=====
Line from feature-1
>>>>>>> main
```

When you saw this in your local file:

```
<<<<<<< HEAD
Line from feature-2
=====
Line from feature-1
>>>>>>> 7037b0fffb95137d3db043f42114473a03d252cd
```

That 7037b0f... is a **local commit hash** (on your machine) that represents the tip of main at the time Git attempted the merge. It **might not match** GitHub's **web UI** because:

- Your local main and remote main could be slightly different.
- You pulled or merged before pushing some changes.
- GitHub often shows the abbreviated hash of the **merge commit** (not always the one being merged **into**).

Your GitHub Shows:

- main has a commit: 78f544b (from yesterday)
- That commit added: Welcome to NHCE in example.txt
- The parent of that commit: 7176ba2

Here's What's Likely Happening:

- Your **local** main had the commit 7037b0f... (perhaps a new one made while resolving feature-1 merge locally).
- GitHub's main still reflects the last visible pushed commit 78f544b (perhaps from before or after a rebase or squash merge).

This **discrepancy is normal** in Git — hashes can differ locally vs. GitHub because of:

- Merge commits
- Rebase operations
- Squash merges
- History rewrites

6. I prefer to replace the content with :

Line from both feature-1 and feature-2

Ensure that you are in feature-2 branch.

```
git branch
```

```
ubuntu@ip-172-31-5-122:~/Experiment2$ git branch
feature
feature-1
* feature-2
main
ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

```
vi example.txt
cat example.txt
```

```
ubuntu@ip-172-31-5-122:~/Experiment2$ vi example.txt
ubuntu@ip-172-31-5-122:~/Experiment2$ cat example.txt
Line from both feature-1 and feature-2
ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

7. Finalize the resolution:

```
git add example.txt
```

```
git commit -m "Resolved merge conflict in example.txt"
```

```
ubuntu@ip-172-31-5-122:~/Experiment2$ git add example.txt
ubuntu@ip-172-31-5-122:~/Experiment2$ git commit -m "Resolved merge conflict in example.txt"
[feature-2 71489e2] Resolved merge conflict in example.txt
Committer: Ubuntu <ubuntu@ip-172-31-5-122.ap-south-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

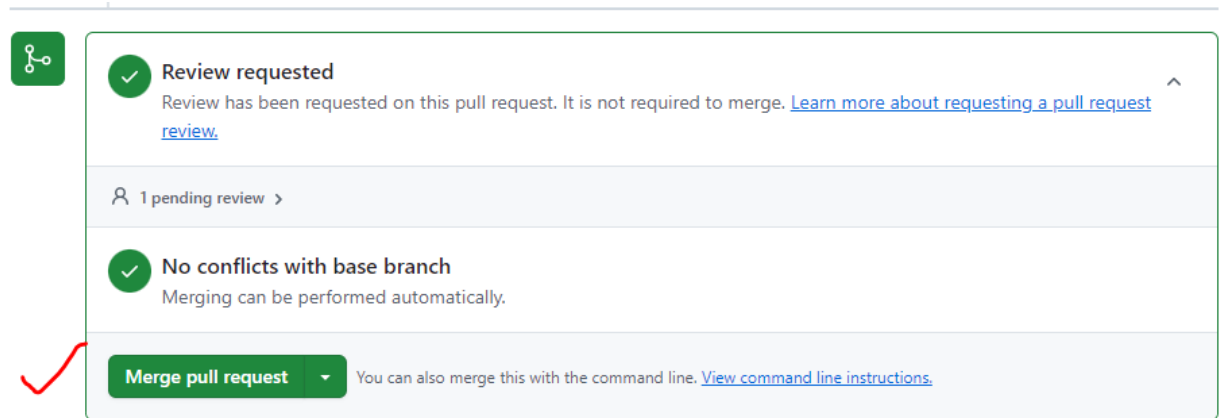
After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author
ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

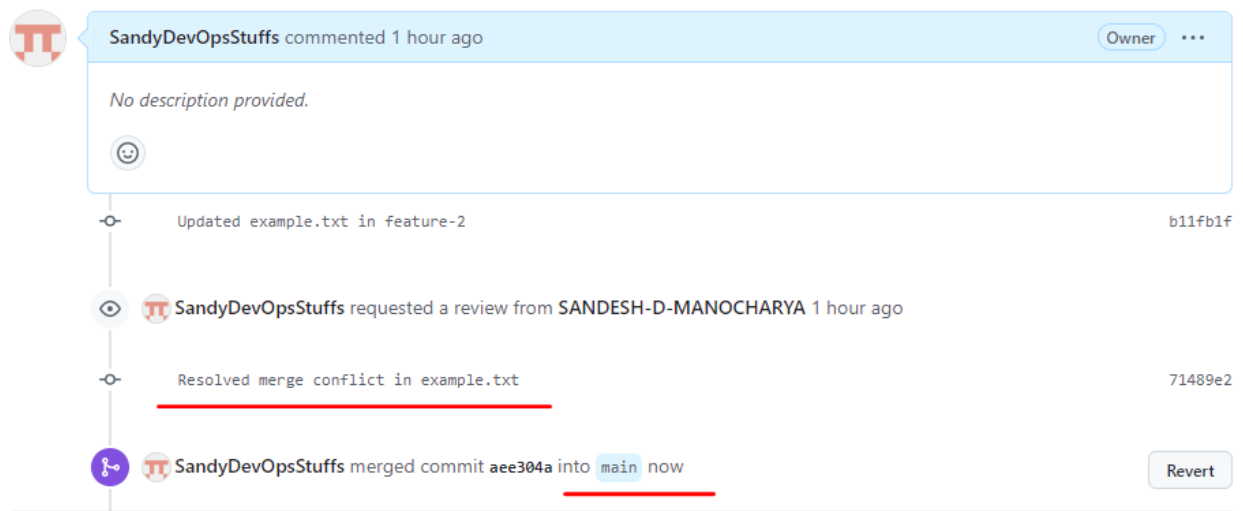
```
git push origin feature-2
```

```
ubuntu@ip-172-31-5-122:~/Experiment2$ git push origin feature-2
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 376 bytes | 376.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:SandyDevOpsStuffs/Experiment2.git
    b11fb1f..71489e2  feature-2 -> feature-2
ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

- Go back to GitHub, PR and complete the merge of `feature-2`. Because there is nothing to do by the reviewer since we got merge conflict during the process of PR creation itself. So first resolve it and merge it yourself.



- Click on "Confirm merge".



- Now go to `feature-2` and `main` and observe the content of `example.txt`.

SandyDevOpsStuffs / Experiment2

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Files

feature-2

Go to file

README.md

example.txt

Experiment2 / example.txt

Ubuntu Resolved merge conflict in example.txt

Code Blame 1 lines (1 loc) · 39 Bytes Code 55% faster with GitHub Copilot

1 Line from both feature-1 and feature-2

SandyDevOpsStuffs / Experiment2

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Files

main

Go to file

README.md

example.txt

Experiment2 / example.txt

Ubuntu Resolved merge conflict in example.txt

Code Blame 1 lines (1 loc) · 39 Bytes Code 55% faster with GitHub Copilot

1 Line from both feature-1 and feature-2

Step 5: Syncing Changes

After the pull request is merged, update your local repository:

But before that first let us see the content of file in each branch:

```
git branch
```

```
cat example.txt
```



```
ubuntu@ip-172-31-5-122:~/Experiment2$ git branch
  feature
  feature-1
* feature-2
  main
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$ cat example.txt
Line from both feature-1 and feature-2
ubuntu@ip-172-31-5-122:~/Experiment2$
```

git checkout feature

cat example.txt

```
ubuntu@ip-172-31-5-122:~/Experiment2$ git checkout feature
Switched to branch 'feature'
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$ cat example.txt
Welcome to NHCE
ubuntu@ip-172-31-5-122:~/Experiment2$
```

git checkout feature-1

cat example.txt

```
ubuntu@ip-172-31-5-122:~/Experiment2$ git checkout feature-1
Switched to branch 'feature-1'
ubuntu@ip-172-31-5-122:~/Experiment2$ git branch
  feature
* feature-1
  feature-2
  main
ubuntu@ip-172-31-5-122:~/Experiment2$ cat example.txt
Line from feature-1
ubuntu@ip-172-31-5-122:~/Experiment2$
```

git checkout main

cat example.txt

```
ubuntu@ip-172-31-5-122:~/Experiment2$ git checkout main
Switched to branch 'main'
Your branch is behind 'origin/main' by 3 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)
ubuntu@ip-172-31-5-122:~/Experiment2$
ubuntu@ip-172-31-5-122:~/Experiment2$ cat example.txt
Welcome to NHCE
ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

Now update your local repository.

```
git checkout main
```

```
ubuntu@ip-172-31-5-122:~/Experiment2$ git checkout main
Already on 'main'
Your branch is behind 'origin/main' by 3 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)
ubuntu@ip-172-31-5-122:~/Experiment2$
```

```
git branch
```

```
ubuntu@ip-172-31-5-122:~/Experiment2$ git branch
  feature
  feature-1
  feature-2
* main
ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

```
git pull origin main
```

```
ubuntu@ip-172-31-5-122:~/Experiment2$ git pull origin main
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (1/1), 910 bytes | 910.00 KiB/s, done.
From github.com:SandyDevOpsStuffs/Experiment2
 * branch                main          -> FETCH_HEAD
  7037b0f..aee304a main      -> origin/main
Updating 78f544b..aee304a
Fast-forward
 example.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

```
cat example.txt
```

```
ubuntu@ip-172-31-5-122:~/Experiment2$ cat example.txt
Line from both feature-1 and feature-2
ubuntu@ip-172-31-5-122:~/Experiment2$ |
```

Now your `main` branch is updated with the latest code.

Conclusion:

This experiment provided you with practical experience in performing GitHub operations using Git commands. You learned how to clone repositories, make changes, create branches, push changes to GitHub, collaborate through pull requests, and synchronize changes with remote repositories. You also explored how to resolve merge conflicts, a common challenge in collaborative development.

Questions:

1. Explain the difference between Git and GitHub.
2. What is a GitHub repository? How is it different from a Git repository?
3. Describe the purpose of a README.md file in a GitHub repository.
4. How do you create a new repository on GitHub? What information is required during the creation process?
5. Define what a pull request (PR) is on GitHub. How does it facilitate collaboration among developers?
6. Describe the typical workflow of creating a pull request and having it merged into the main branch.
7. How can you address and resolve merge conflicts in a pull request?
8. Explain the concept of forking a repository on GitHub. How does it differ from cloning a repository?
9. What is the purpose of creating a local clone of a repository on your machine? How is it done using Git commands?
10. Describe the role of GitHub Issues and Projects in managing a software development project. How can they be used to track tasks, bugs, and enhancements?