Report on GitFlow Assignment

JUNE 25

Authored by: Deya’ Aldeen AL-Bettar

Table Content

Introduction

1. What is GitFlow

   - GitFlow is a branching model for Git, created by Vincent Driessen, which provides a structured process for managing feature development, release preparation, and maintenance of production code.

2. Why use GitFlow

   1. Structured Workflow
      - GitFlow provides a clear structure for managing development, which is particularly beneficial for teams working on complex projects. The separation of different types of work (e.g., new features, releases, and hotfixes) into specific branches helps organize the workflow.

   2. Parallel Development
      - GitFlow supports parallel development by allowing multiple feature branches to be worked on simultaneously. This is especially useful in larger teams where different team members are responsible for different features.

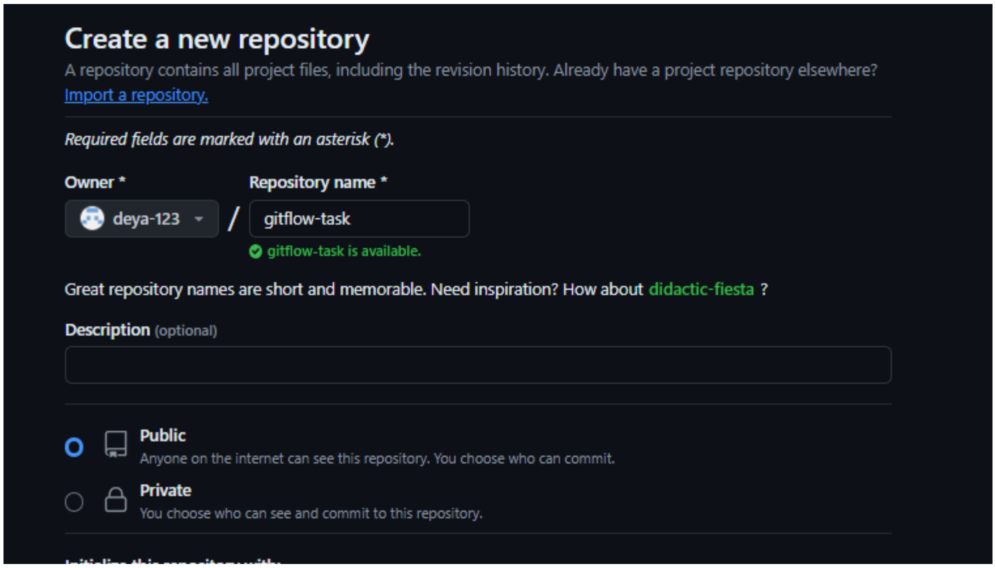
   3. Collaboration and Code Review
      - GitFlow branching strategy encourages the use of pull requests and code reviews. Feature branches are typically merged back into develop only after review, improving code quality and collaboration.

Description my project

The project involves creating a simple project using text files, designed to simulate a collaborative development environment.

The implementation of GitFlow

1. Repository Creation:
    - Start by creating a new Git repository on GitHub or another Git hosting service. Name it "gitflow-task" for our project.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner * deya-123 / Repository name * gitflow-task

gitflow-task is available.

Great repository names are short and memorable. Need inspiration? How about didactic-fiesta ?

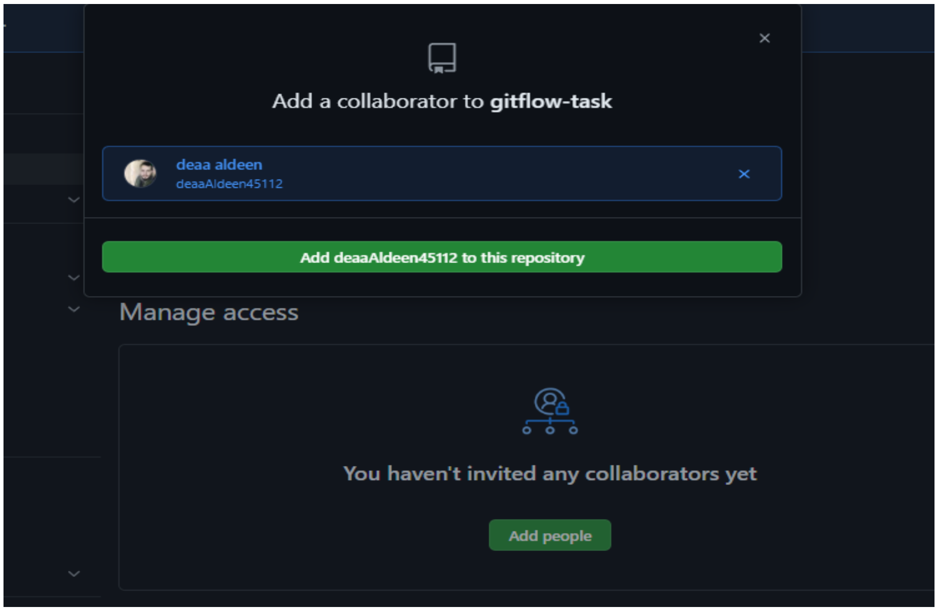
Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:

2. Inviting Collaborators:
    - Invite team members or simulated collaborators to join the repository. This step involves granting them appropriate access (e.g., read, write, or admin) to facilitate collaboration.



Add a collaborator to **gitflow-task**

deaa aldeen
deaaAldeen45112

Add deaaAldeen45112 to this repository

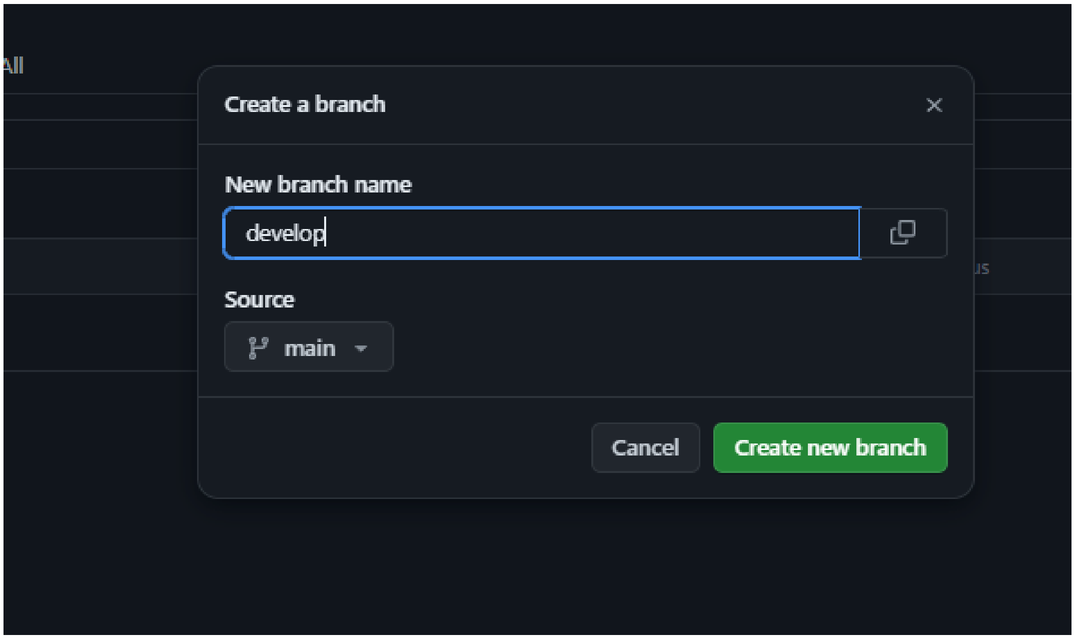
Manage access

You haven't invited any collaborators yet

Add people

3. Creating the Development Branch:

- Begin by establishing the `develop` branch as the main integration branch for ongoing development work. This branch serves as the foundation for incorporating new features and enhancements.



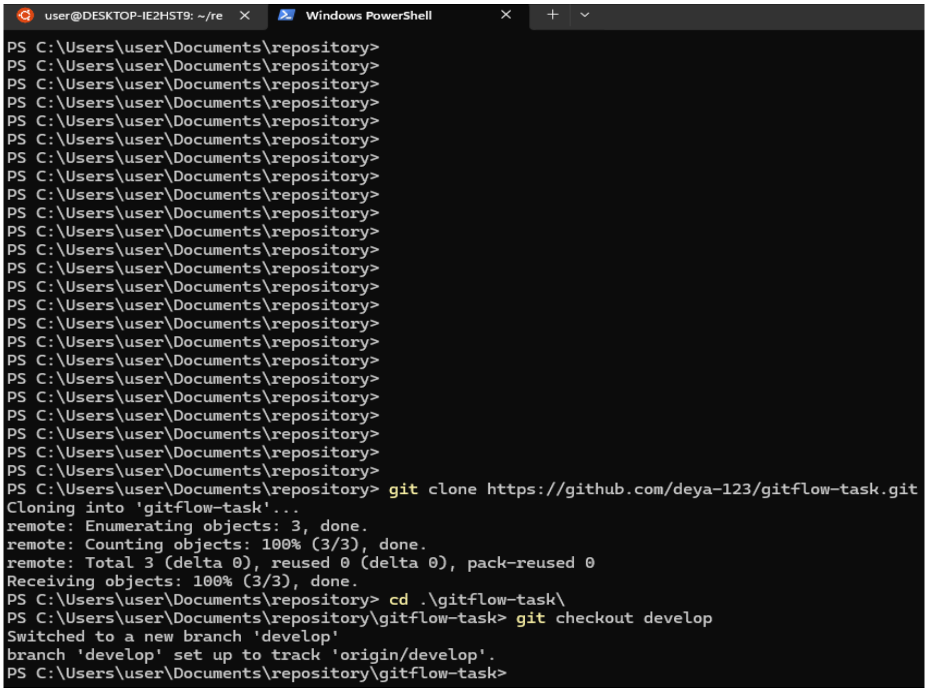
Create a branch

New branch name: develop

Source: main

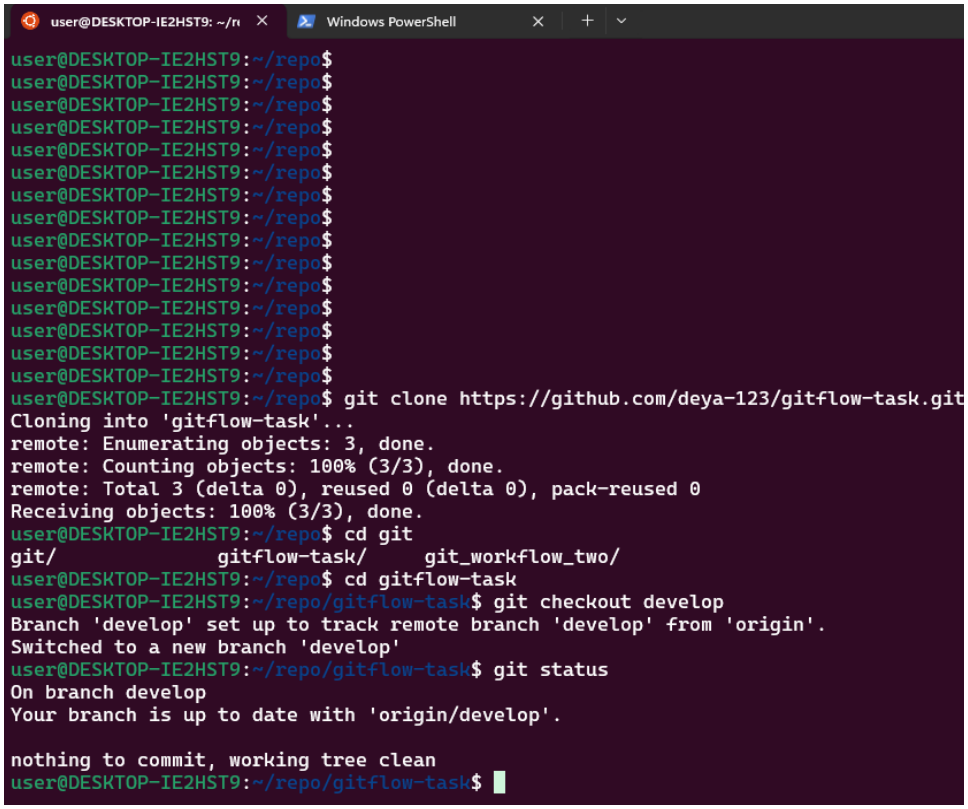
4. Cloning the Repository:

   - I have two developers, the first developer is a Linux developer (deaaaldeen45112) and the second is a Windows developer (deaa-123), each developer cloned the repository on their local machine and then switched to the `develop` branch:
   - Windows developer:



```
PS C:\Users\user\Documents\repository>
PS C:\Users\user\Documents\repository>
PS C:\Users\user\Documents\repository>
PS C:\Users\user\Documents\repository>
PS C:\Users\user\Documents\repository>
PS C:\Users\user\Documents\repository>
PS C:\Users\user\Documents\repository>
PS C:\Users\user\Documents\repository>
PS C:\Users\user\Documents\repository>
PS C:\Users\user\Documents\repository>
PS C:\Users\user\Documents\repository>
PS C:\Users\user\Documents\repository>
PS C:\Users\user\Documents\repository>
PS C:\Users\user\Documents\repository>
PS C:\Users\user\Documents\repository>
PS C:\Users\user\Documents\repository>
PS C:\Users\user\Documents\repository>
PS C:\Users\user\Documents\repository>
PS C:\Users\user\Documents\repository>
PS C:\Users\user\Documents\repository>
PS C:\Users\user\Documents\repository>
PS C:\Users\user\Documents\repository>
PS C:\Users\user\Documents\repository>
PS C:\Users\user\Documents\repository>
PS C:\Users\user\Documents\repository> git clone https://github.com/deya-123/gitflow-task.git
Cloning into 'gitflow-task'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
PS C:\Users\user\Documents\repository> cd .\gitflow-task\
PS C:\Users\user\Documents\repository\gitflow-task> git checkout develop
Switched to a new branch 'develop'
branch 'develop' set up to track 'origin/develop'.
PS C:\Users\user\Documents\repository\gitflow-task>
```

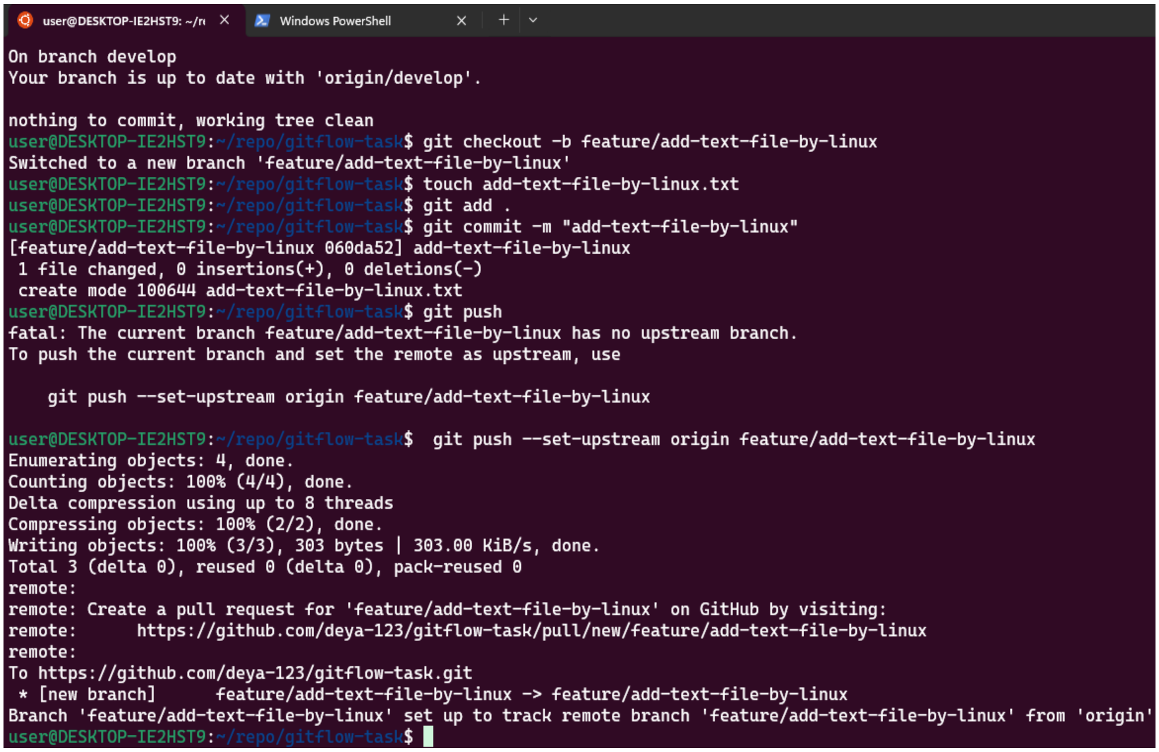
- Linux developer:



```
user@DESKTOP-IE2HST9:~/repo$
user@DESKTOP-IE2HST9:~/repo$
user@DESKTOP-IE2HST9:~/repo$
user@DESKTOP-IE2HST9:~/repo$
user@DESKTOP-IE2HST9:~/repo$
user@DESKTOP-IE2HST9:~/repo$
user@DESKTOP-IE2HST9:~/repo$
user@DESKTOP-IE2HST9:~/repo$
user@DESKTOP-IE2HST9:~/repo$
user@DESKTOP-IE2HST9:~/repo$
user@DESKTOP-IE2HST9:~/repo$
user@DESKTOP-IE2HST9:~/repo$
user@DESKTOP-IE2HST9:~/repo$
user@DESKTOP-IE2HST9:~/repo$
user@DESKTOP-IE2HST9:~/repo$
user@DESKTOP-IE2HST9:~/repo$ git clone https://github.com/deya-123/gitflow-task.git
Cloning into 'gitflow-task'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
user@DESKTOP-IE2HST9:~/repo$ cd git
git/              gitflow-task/     git_workflow_two/
user@DESKTOP-IE2HST9:~/repo$ cd gitflow-task
user@DESKTOP-IE2HST9:~/repo/gitflow-task$ git checkout develop
Branch 'develop' set up to track remote branch 'develop' from 'origin'.
Switched to a new branch 'develop'
user@DESKTOP-IE2HST9:~/repo/gitflow-task$ git status
On branch develop
Your branch is up to date with 'origin/develop'.

nothing to commit, working tree clean
user@DESKTOP-IE2HST9:~/repo/gitflow-task$
```

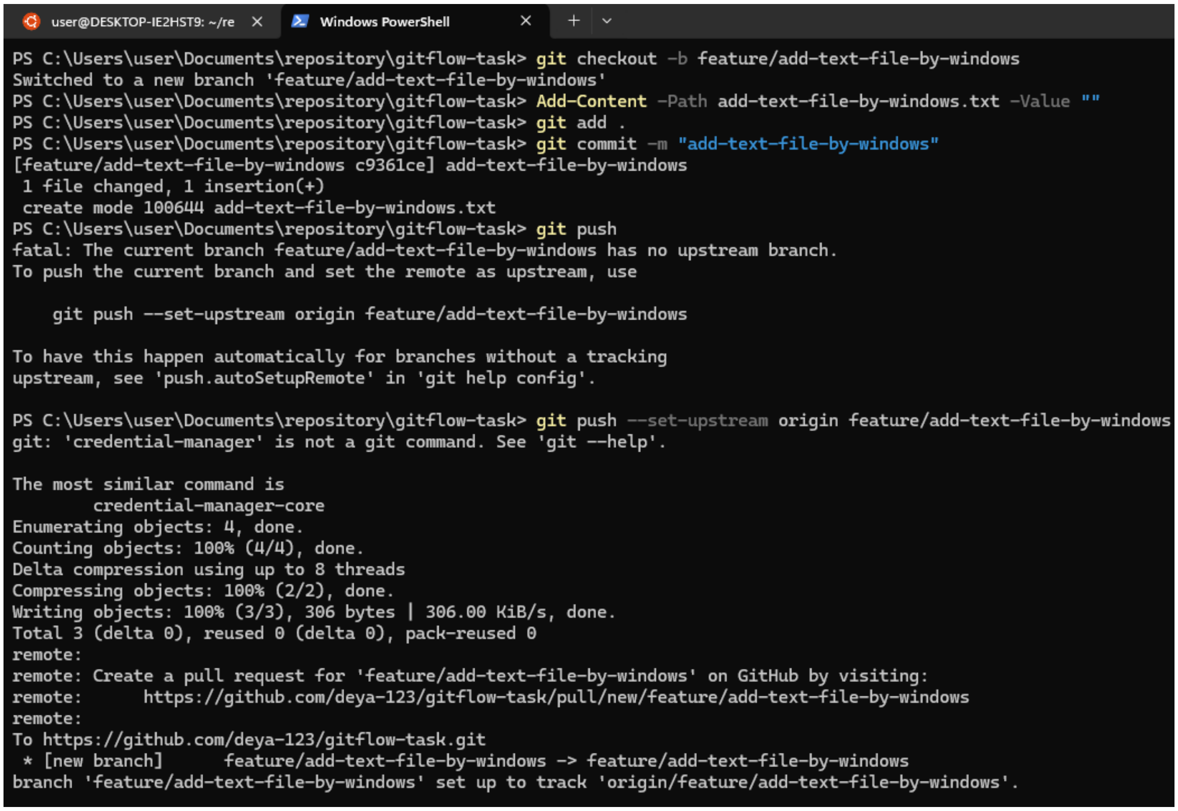
5. Features:
    - Adding Features
        - Collaborators create feature branches from `develop` to independently develop specific functionalities

- Linux Developer (Feature: add-text-file-by-linux):



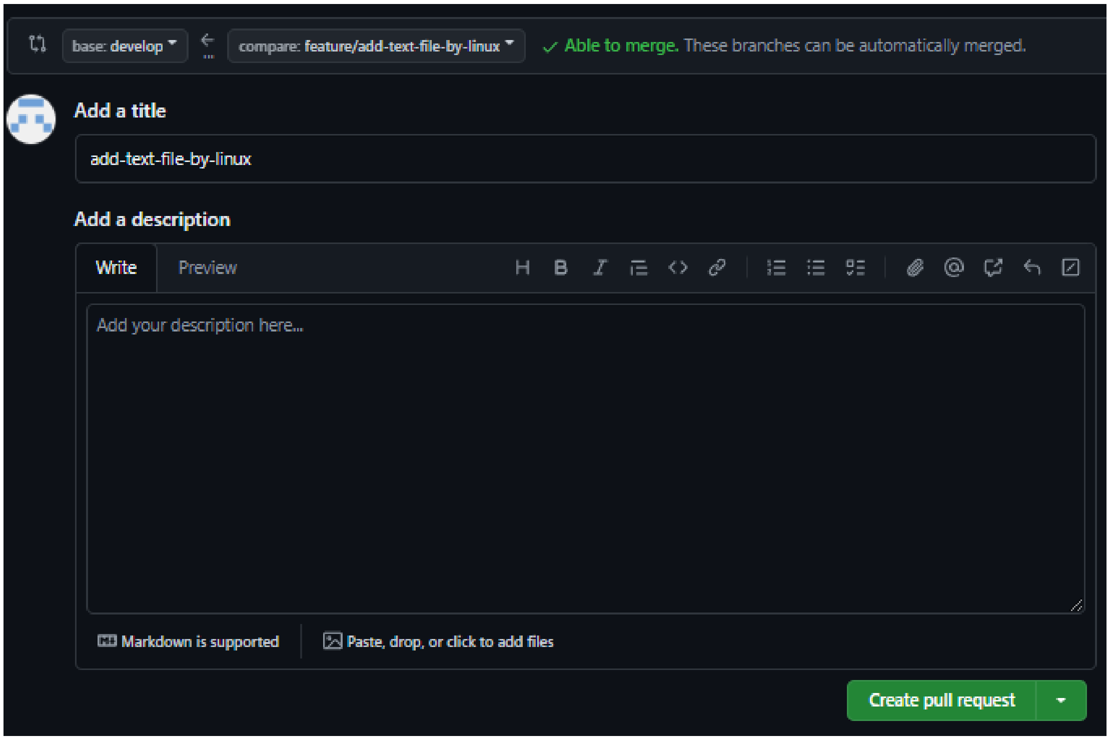
```
On branch develop
Your branch is up to date with 'origin/develop'.

nothing to commit, working tree clean
user@DESKTOP-IE2HST9:~/repo/gitflow-task$ git checkout -b feature/add-text-file-by-linux
Switched to a new branch 'feature/add-text-file-by-linux'
user@DESKTOP-IE2HST9:~/repo/gitflow-task$ touch add-text-file-by-linux.txt
user@DESKTOP-IE2HST9:~/repo/gitflow-task$ git add .
user@DESKTOP-IE2HST9:~/repo/gitflow-task$ git commit -m "add-text-file-by-linux"
[feature/add-text-file-by-linux 060da52] add-text-file-by-linux
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 add-text-file-by-linux.txt
user@DESKTOP-IE2HST9:~/repo/gitflow-task$ git push
fatal: The current branch feature/add-text-file-by-linux has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin feature/add-text-file-by-linux

user@DESKTOP-IE2HST9:~/repo/gitflow-task$  git push --set-upstream origin feature/add-text-file-by-linux
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 303 bytes | 303.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature/add-text-file-by-linux' on GitHub by visiting:
remote:      https://github.com/deya-123/gitflow-task/pull/new/feature/add-text-file-by-linux
remote:
To https://github.com/deya-123/gitflow-task.git
 * [new branch]      feature/add-text-file-by-linux -> feature/add-text-file-by-linux
Branch 'feature/add-text-file-by-linux' set up to track remote branch 'feature/add-text-file-by-linux' from 'origin'
user@DESKTOP-IE2HST9:~/repo/gitflow-task$ 
```

- Windows Developer (Feature: add-text-file-by-windows):



```
PS C:\Users\user\Documents\repository\gitflow-task> git checkout -b feature/add-text-file-by-windows
Switched to a new branch 'feature/add-text-file-by-windows'
PS C:\Users\user\Documents\repository\gitflow-task> Add-Content -Path add-text-file-by-windows.txt -Value ""
PS C:\Users\user\Documents\repository\gitflow-task> git add .
PS C:\Users\user\Documents\repository\gitflow-task> git commit -m "add-text-file-by-windows"
[feature/add-text-file-by-windows c9361ce] add-text-file-by-windows
 1 file changed, 1 insertion(+)
 create mode 100644 add-text-file-by-windows.txt
PS C:\Users\user\Documents\repository\gitflow-task> git push
fatal: The current branch feature/add-text-file-by-windows has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin feature/add-text-file-by-windows

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.

PS C:\Users\user\Documents\repository\gitflow-task> git push --set-upstream origin feature/add-text-file-by-windows
git: 'credential-manager' is not a git command. See 'git --help'.

The most similar command is
        credential-manager-core
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 306 bytes | 306.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature/add-text-file-by-windows' on GitHub by visiting:
remote:      https://github.com/deya-123/gitflow-task/pull/new/feature/add-text-file-by-windows
remote:
To https://github.com/deya-123/gitflow-task.git
 * [new branch]      feature/add-text-file-by-windows -> feature/add-text-file-by-windows
branch 'feature/add-text-file-by-windows' set up to track 'origin/feature/add-text-file-by-windows'.
```

- Merging Features
  - Once features are complete and tested within their respective branches, they are merged back into `develop` using pull requests (PRs). This process includes conducting thorough code reviews to address any feedback or conflicts, ensuring high code quality and stable integration.
  - Linux Developer (Feature: add-text-file-by-linux):



base: develop ← compare: feature/add-text-file-by-linux

✓ Able to merge. These branches can be automatically merged.

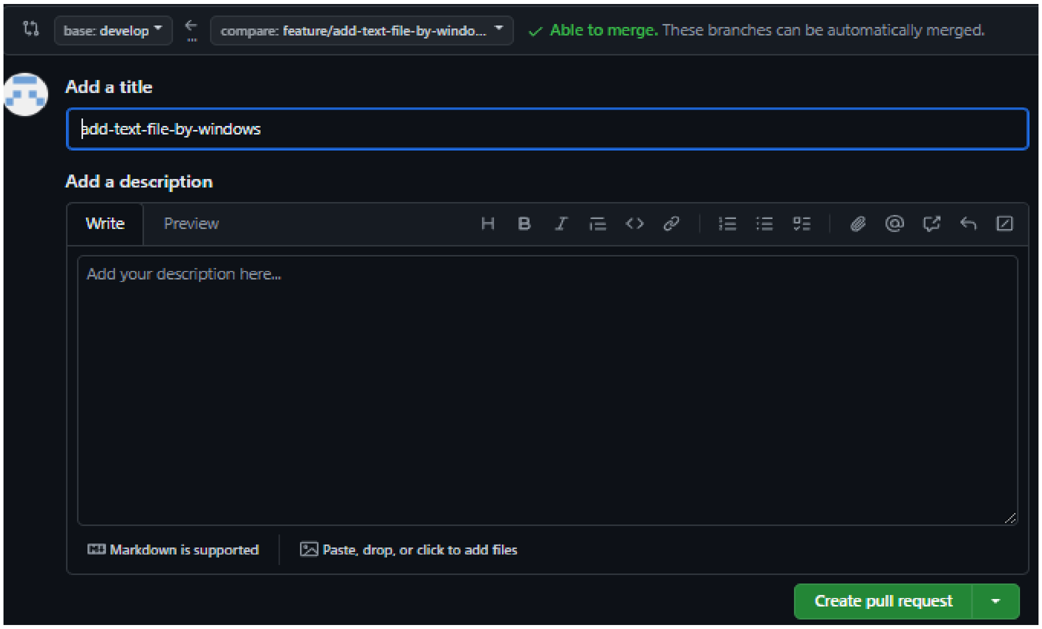
Add a title

add-text-file-by-linux

Add a description

Write
Preview
Add your description here...
Markdown is supported
Paste, drop, or click to add files
Create pull request

  - Windows Developer (Feature: add-text-file-by-windows):



base: develop ← compare: feature/add-text-file-by-windo...

✓ Able to merge. These branches can be automatically merged.

Add a title

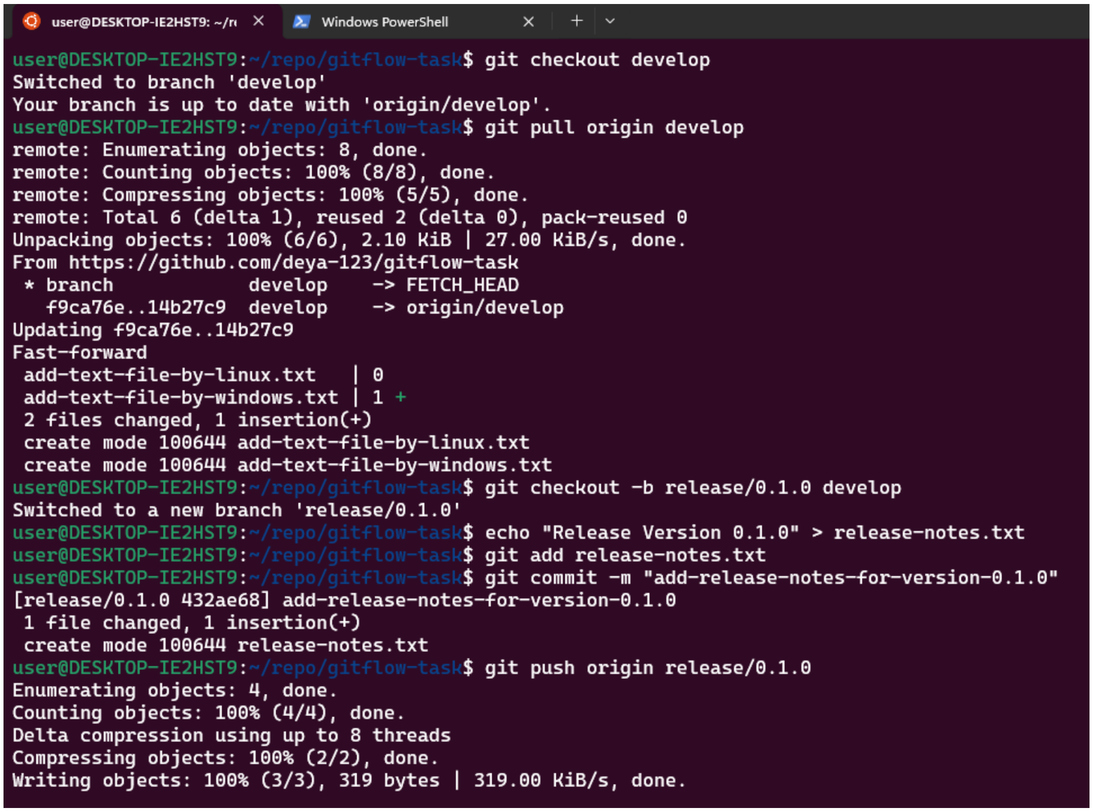
add-text-file-by-windows

Add a description

Write
Preview
Add your description here...
Markdown is supported
Paste, drop, or click to add files
Create pull request

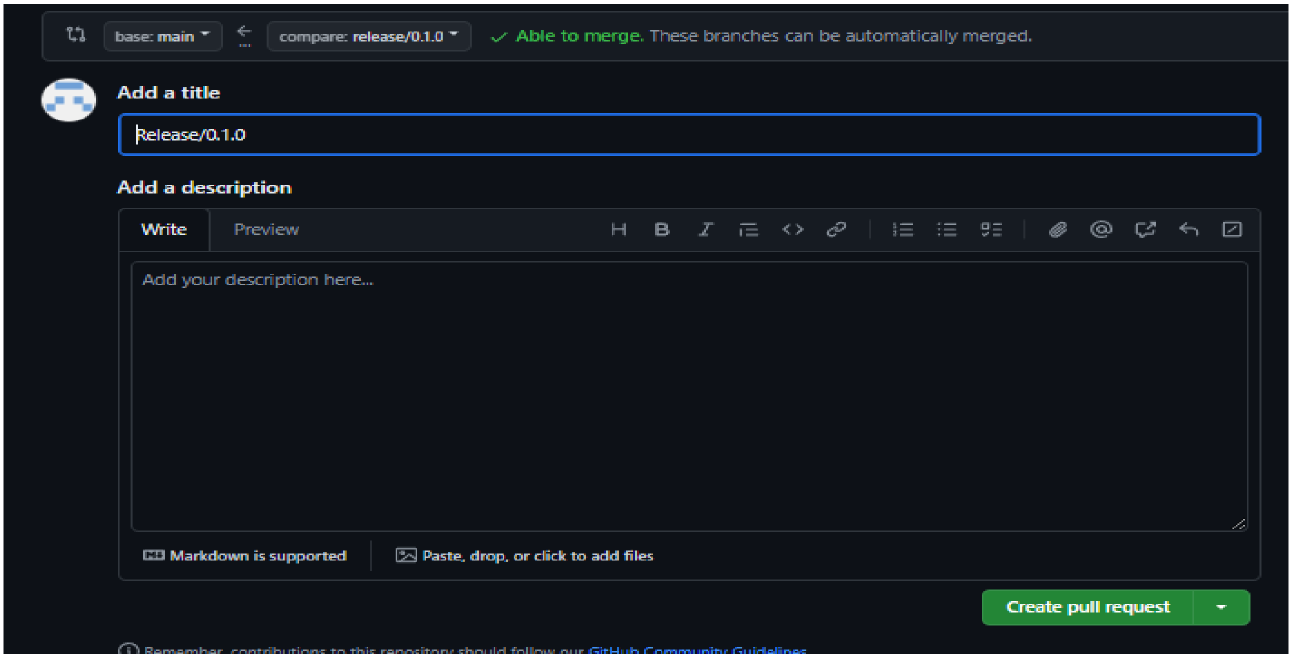
6. Release:

- Adding a Release
    - The Linux developer pulls the latest changes from the `develop` branch to ensure their local repository is up-to-date, Prepare for a release by branching off develop into a release branch, and then creating a new file (release-notes.txt) to document the release



```
user@DESKTOP-IE2HST9:~/repo/gitflow-task$ git checkout develop
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
user@DESKTOP-IE2HST9:~/repo/gitflow-task$ git pull origin develop
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 6 (delta 1), reused 2 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), 2.10 KiB | 27.00 KiB/s, done.
From https://github.com/deya-123/gitflow-task
 * branch            develop    -> FETCH_HEAD
   f9ca76e..14b27c9  develop    -> origin/develop
Updating f9ca76e..14b27c9
Fast-forward
 add-text-file-by-linux.txt   | 0
 add-text-file-by-windows.txt | 1 +
 2 files changed, 1 insertion(+)
 create mode 100644 add-text-file-by-linux.txt
 create mode 100644 add-text-file-by-windows.txt
user@DESKTOP-IE2HST9:~/repo/gitflow-task$ git checkout -b release/0.1.0 develop
Switched to a new branch 'release/0.1.0'
user@DESKTOP-IE2HST9:~/repo/gitflow-task$ echo "Release Version 0.1.0" > release-notes.txt
user@DESKTOP-IE2HST9:~/repo/gitflow-task$ git add release-notes.txt
user@DESKTOP-IE2HST9:~/repo/gitflow-task$ git commit -m "add-release-notes-for-version-0.1.0"
[release/0.1.0 432ae68] add-release-notes-for-version-0.1.0
 1 file changed, 1 insertion(+)
 create mode 100644 release-notes.txt
user@DESKTOP-IE2HST9:~/repo/gitflow-task$ git push origin release/0.1.0
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 319 bytes | 319.00 KiB/s, done.
```

- Merging a Release
    - ○ Merge the release branch into `main` and `develop`



✓ Able to merge. These branches can be automatically merged.

Add a title

Release/0.1.0

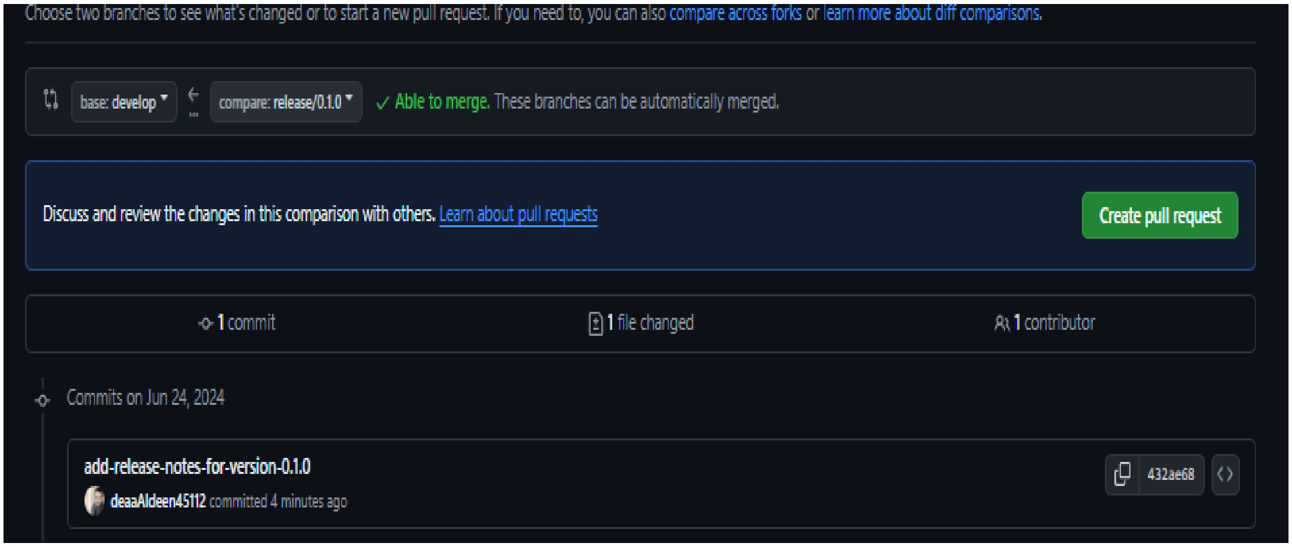
Add a description

Add your description here...

Markdown is supported

Paste, drop, or click to add files

Remember, contributions to this repository should follow our GitHub Community Guidelines



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.

✓ Able to merge. These branches can be automatically merged.

Discuss and review the changes in this comparison with others. Learn about pull requests

1 commit

1 file changed

1 contributor

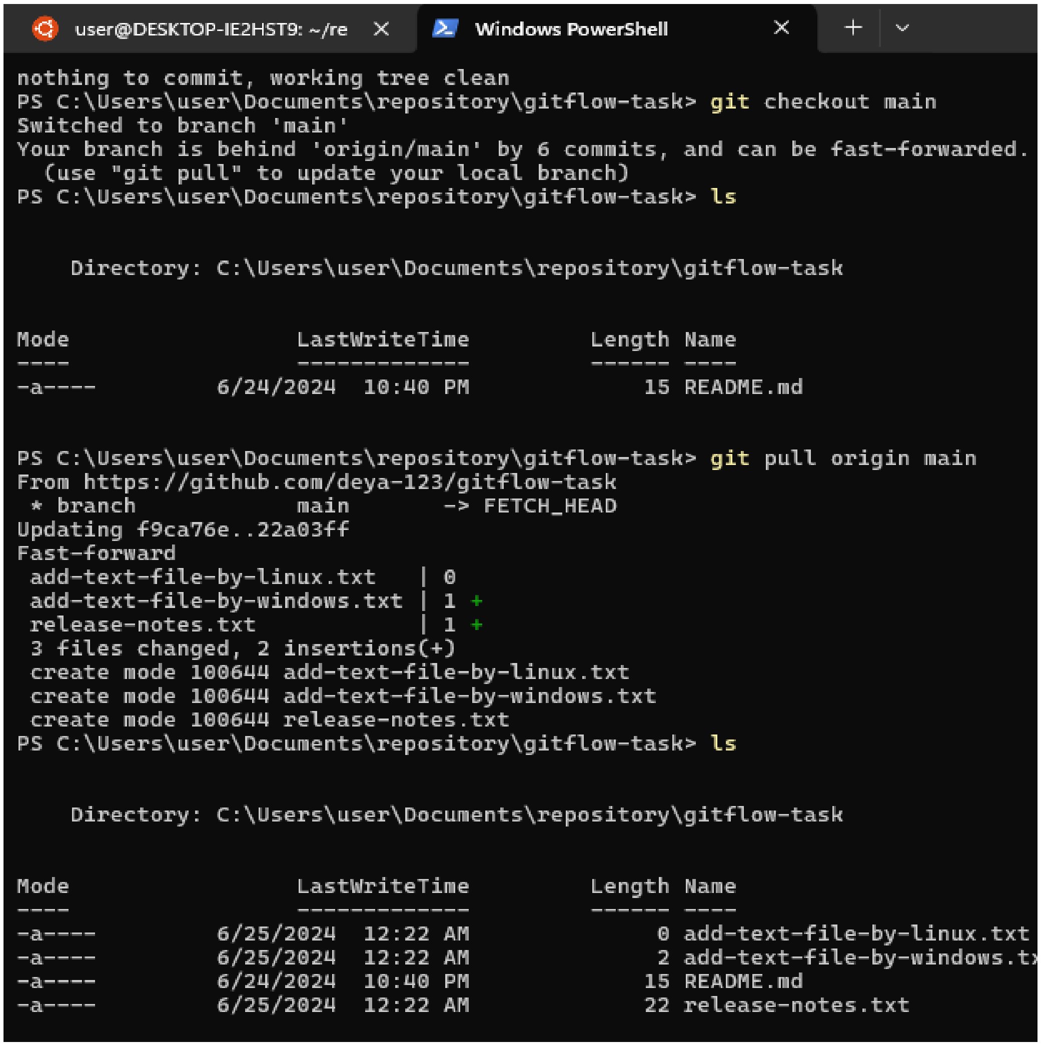
Commits on Jun 24, 2024

add-release-notes-for-version-0.1.0

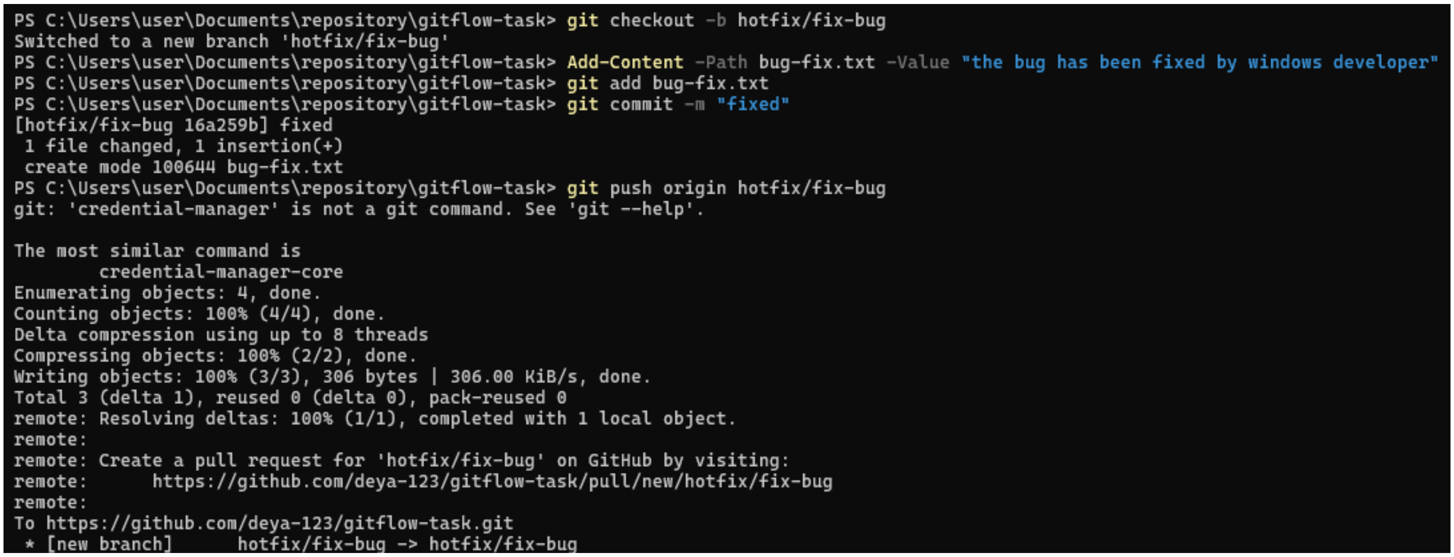
deaaAldeen45112 committed 4 minutes ago

7. Hotfix:

- Adding a Hotfix
    - ○ Address urgent issues or bugs discovered in the production environment by creating a `hotfix` branch directly from `master`.
    - ○ The Windows developer prepares for a hotfix by pulling the latest changes from the `main` branch, and then addressing an urgent bug fix by creating a `hotfix` branch from `master`, the bug is just adding sentence "the bug has been fixed by windows developer" in the bug-fix.txt file

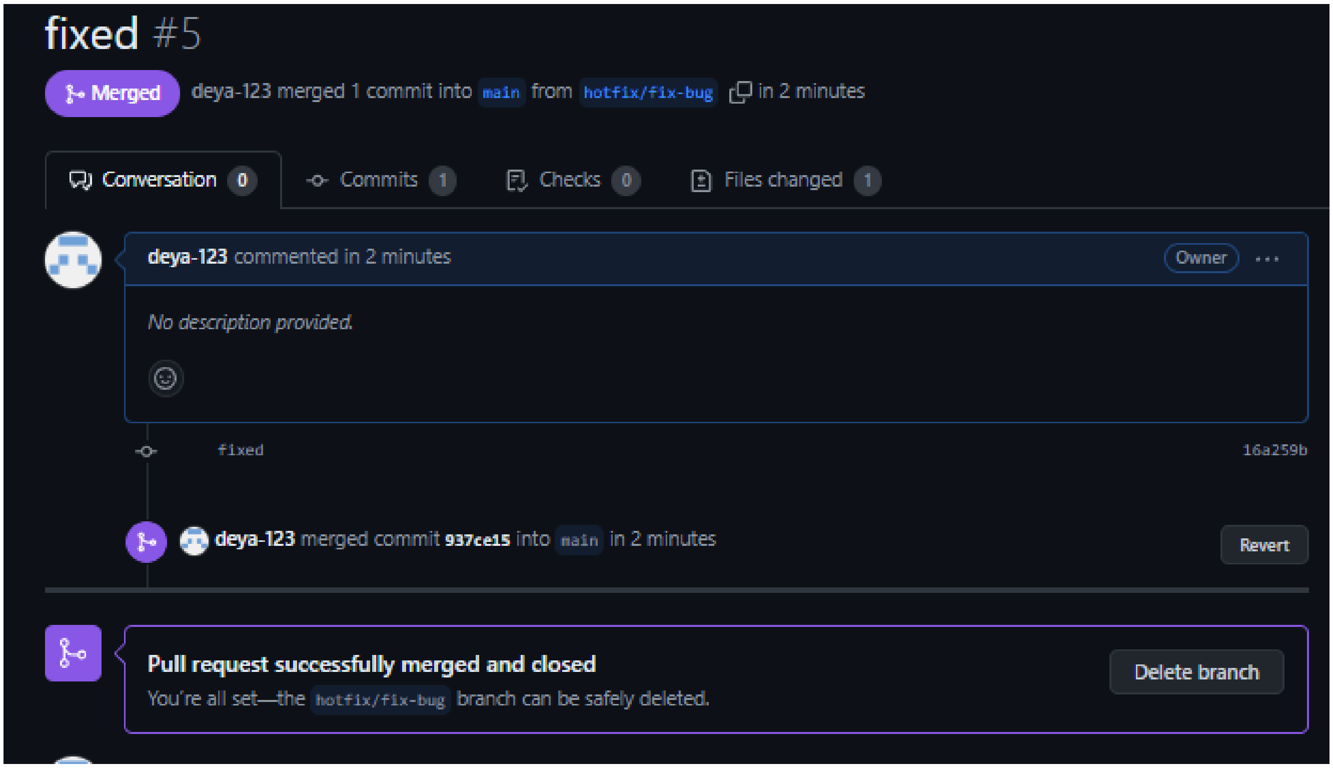


```
nothing to commit, working tree clean
PS C:\Users\user\Documents\repository\gitflow-task> git checkout main
Switched to branch 'main'
Your branch is behind 'origin/main' by 6 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)
PS C:\Users\user\Documents\repository\gitflow-task> ls


    Directory: C:\Users\user\Documents\repository\gitflow-task


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         6/24/2024  10:40 PM             15 README.md


PS C:\Users\user\Documents\repository\gitflow-task> git pull origin main
From https://github.com/deya-123/gitflow-task
 * branch            main       -> FETCH_HEAD
Updating f9ca76e..22a03ff
Fast-forward
 add-text-file-by-linux.txt   | 0
 add-text-file-by-windows.txt | 1 +
 release-notes.txt            | 1 +
 3 files changed, 2 insertions(+)
 create mode 100644 add-text-file-by-linux.txt
 create mode 100644 add-text-file-by-windows.txt
 create mode 100644 release-notes.txt
PS C:\Users\user\Documents\repository\gitflow-task> ls


    Directory: C:\Users\user\Documents\repository\gitflow-task


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         6/25/2024  12:22 AM              0 add-text-file-by-linux.txt
-a----         6/25/2024  12:22 AM              2 add-text-file-by-windows.tx
-a----         6/24/2024  10:40 PM             15 README.md
-a----         6/25/2024  12:22 AM             22 release-notes.txt
```



```
PS C:\Users\user\Documents\repository\gitflow-task> git checkout -b hotfix/fix-bug
Switched to a new branch 'hotfix/fix-bug'
PS C:\Users\user\Documents\repository\gitflow-task> Add-Content -Path bug-fix.txt -Value "the bug has been fixed by windows developer"
PS C:\Users\user\Documents\repository\gitflow-task> git add bug-fix.txt
PS C:\Users\user\Documents\repository\gitflow-task> git commit -m "fixed"
[hotfix/fix-bug 16a259b] fixed
 1 file changed, 1 insertion(+)
 create mode 100644 bug-fix.txt
PS C:\Users\user\Documents\repository\gitflow-task> git push origin hotfix/fix-bug
git: 'credential-manager' is not a git command. See 'git --help'.

The most similar command is
        credential-manager-core
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 306 bytes | 306.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'hotfix/fix-bug' on GitHub by visiting:
remote:      https://github.com/deya-123/gitflow-task/pull/new/hotfix/fix-bug
remote:
To https://github.com/deya-123/gitflow-task.git
 * [new branch]      hotfix/fix-bug -> hotfix/fix-bug
```

- Merging a Hotfix
    - Merge the hotfix/fig-bug branch into `main` and `develop`



fixed #5

Merged deya-123 merged 1 commit into `main` from `hotfix/fix-bug` in 2 minutes

deya-123 commented in 2 minutes

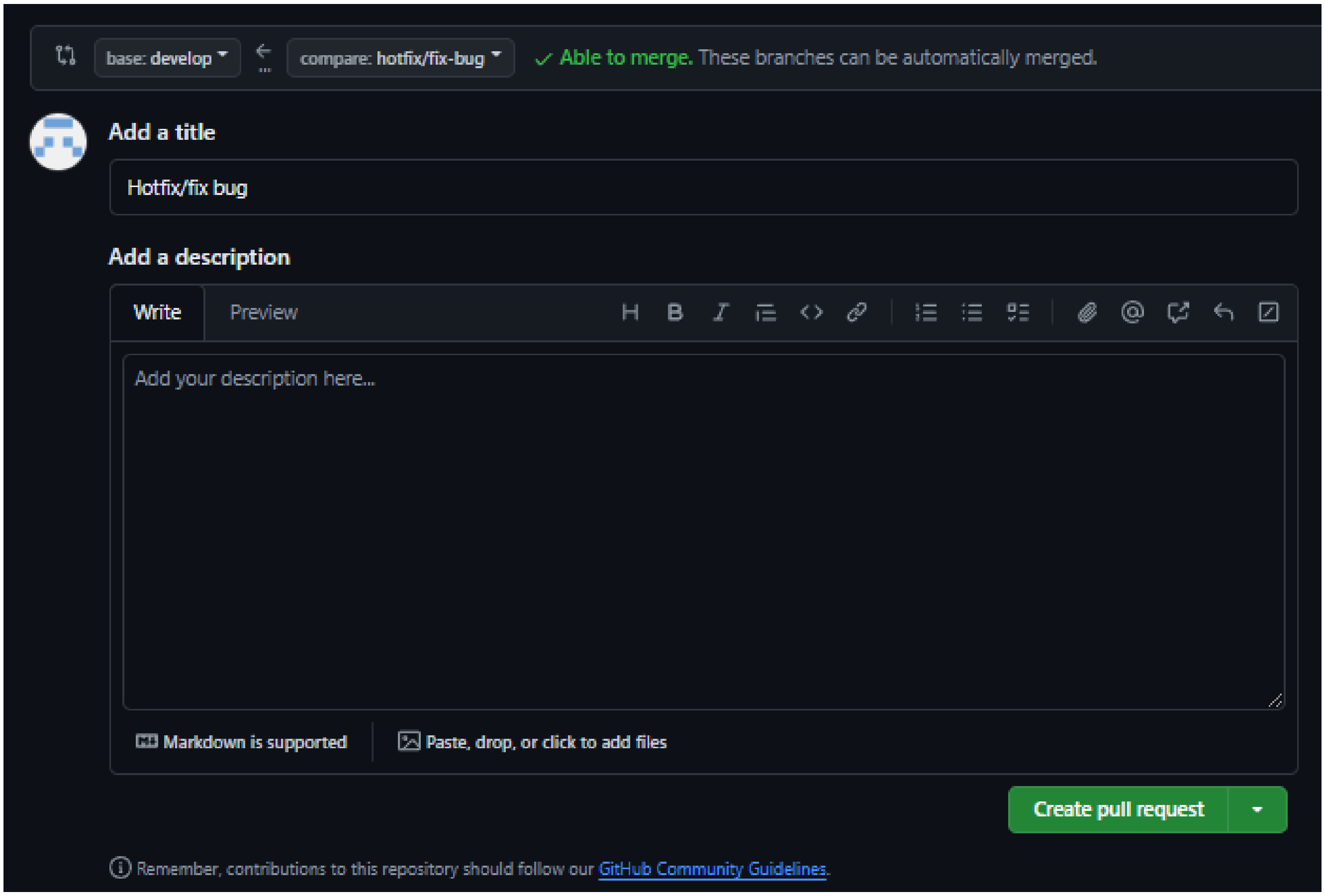
No description provided.

fixed 16a259b

deya-123 merged commit **937ce15** into `main` in 2 minutes

Pull request successfully merged and closed

You're all set—the `hotfix/fix-bug` branch can be safely deleted.



✓ Able to merge. These branches can be automatically merged.

Add a title

Hotfix/fix bug

Add a description

Add your description here...

Remember, contributions to this repository should follow our GitHub Community Guidelines.

GitFlow Challenges and Solutions

Challenges and Solutions

- Challenge 1: Merge Conflicts
  - Problem: Frequent merge conflicts when multiple feature branches are integrated.
  - Solution: Encourage frequent pulling and merging of develop into feature branches.
- Challenge 2: Keeping Branches Up-to-date
  - Problem: Feature branches getting outdated and causing integration problems.
  - Solution: Implement a policy for developers to regularly rebase their feature branches with the latest develop changes.

My learning Experience with GitFlow

Learning GitFlow has been eye-opening, especially in real-world situations where teamwork and precise release management are critical. GitFlow's branching model provides a structured approach to managing larger projects, offering distinct branches for features, releases, and hotfixes.

One key insight is its ability to maintain a clear separation between ongoing development (in the develop branch) and stable releases (in main). This separation ensures that new features can be developed without disrupting the main stable codebase. Moreover, the use of feature branches (feature) allows for isolated development of new features, which can be independently tested and reviewed before merging into develop.