

CASE STUDY- RESOLVING MERGE CONFLICTS

Problem Statement:

You work for Zendrix Software & Co. You have been assigned the task of updating the Master branch of their Git repository with all the features from the feature branches.

Following is the GitHub account, <https://github.com/devops-intellipaat/merge-conflict.git> Consider,

- Feature1 branch to be a public branch
- Feature2 branch to be a private branch

The company relies on a monolithic architecture, and for now all the code resides in one file “main.c”.

The respective features have been added in the feature branches for main.c. Meanwhile, a security patch was made to the master branch, and now feature1 and feature2 branches are behind from master by 1 commit.

Following tasks have to be done:

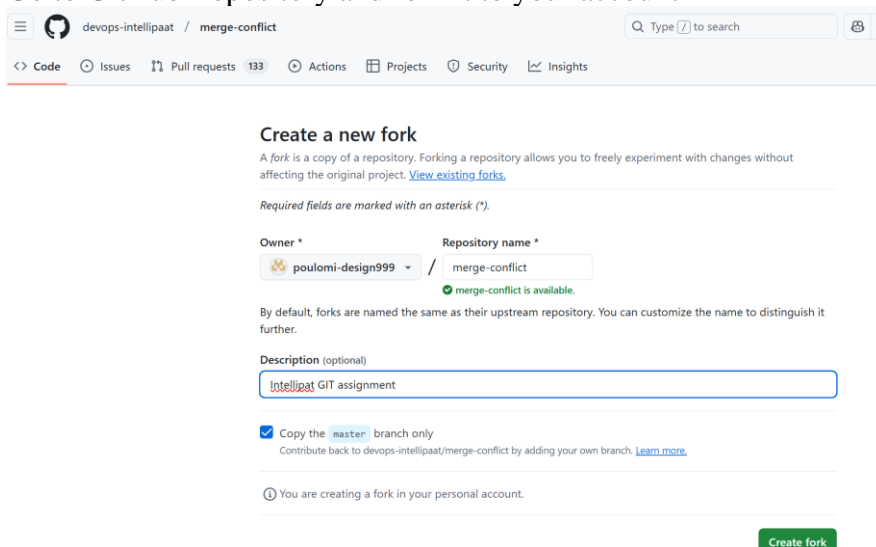
1. Update Feature1 and Feature2 branch with the Security Patch
2. Apply changes of Feature1 and Feature2 branch on master
3. Finally push all the branches to GitHub

For Solving this, please fork the repository to your Github account and then work. As a solution, please submit your GitHub’s repository link.

Solution:

Fork the Repository:

- Go to GitHub Repository and fork it to your account



The screenshot shows the GitHub interface for creating a new fork of the repository 'devops-intellipaat / merge-conflict'. The page includes a search bar, navigation tabs (Code, Issues, Pull requests, Actions, Projects, Security, Insights), and a 'Create a new fork' section. The 'Owner' is set to 'poulomi-design999' and the 'Repository name' is 'merge-conflict'. A green checkmark indicates 'merge-conflict is available'. The 'Description' field contains 'Intellipaat GIT assignment'. The 'Copy the master branch only' checkbox is checked. A green 'Create fork' button is at the bottom right.

devops-intellipaat / merge-conflict

Q Type / to search

<> Code Issues Pull requests 133 Actions Projects Security Insights

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 * Repository name *

poulomi-design999 / merge-conflict

merge-conflict is available.

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

Description (optional)

Intellipaat GIT assignment

☒ Copy the master branch only

Contribute back to devops-intellipaat/merge-conflict by adding your own branch. [Learn more.](#)

① You are creating a fork in your personal account.

Create fork

Clone the Forked Repository:

- Clone the forked repository to your local machine:

`git clone https://github.com/poulomi-design999/merge-conflict.git`

`cd merge-conflict`

```
GIT $git clone https://github.com/poulomi-design999/merge-conflict.git
Cloning into 'merge-conflict'...
remote: Enumerating objects: 9, done.
remote: Total 9 (delta 0), reused 0 (delta 0), pack-reused 9 (from 1)
Receiving objects: 100% (9/9), done.
GIT $
```

```
GIT $cd merge-conflict
GIT $ll
total 1
-rw-r--r-- 1 bhats 197609 112 Feb  9 22:10 main.c
GIT $
```

Add the Original Repository as a Remote:

- Add the original repository as a remote to keep your fork up-to-date:

`git remote add upstream https://github.com/devops-intellipaat/merge-conflict.git`

```
GIT $git remote add upstream https://github.com/devops-intellipaat/merge-conflict.git
GIT $
```

Fetch the Latest Changes from the Original Repository:

- Fetch the latest changes from the original repository:

`git fetch upstream`

```
GIT $git fetch upstream
remote: Enumerating objects: 6, done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 6 (from 1)
Unpacking objects: 100% (6/6), 582 bytes | 38.00 KiB/s, done.
From https://github.com/devops-intellipaat/merge-conflict
* [new branch]      feature1 -> upstream/feature1
* [new branch]      feature2 -> upstream/feature2
* [new branch]      master -> upstream/master
GIT $
```

Update Feature1 Branch with Security Patch:

- Checkout the `feature1` branch and merge the latest changes from `master`:

`git checkout feature1`

```
GIT $git checkout feature1
branch 'feature1' set up to track 'upstream/feature1'.
Switched to a new branch 'feature1'
GIT $|
```

`git merge upstream/master`

```
GIT $git merge upstream/master
Auto-merging main.c
CONFLICT (content): Merge conflict in main.c
Automatic merge failed; fix conflicts and then commit the result.
GIT $
```

Resolve Any Merge Conflicts (If Any):

- Open `main.c` and resolve any merge conflicts. After resolving conflicts:

`git add main.c`

`git commit -m "Resolve merge conflicts and apply security patch to feature1"`

```
GIT $git add main.c
GIT $git commit -m "Resolve merge conflicts and apply security patch to feature1"
[feature1 dbc5b51] Resolve merge conflicts and apply security patch to feature1
GIT $
```

Update Feature2 Branch with Security Patch:

- Checkout the `feature2` branch and merge the latest changes from `master`:

`git checkout feature2`

```
GIT $git checkout feature2
branch 'feature2' set up to track 'upstream/feature2'.
Switched to a new branch 'feature2'
GIT $|
```

`git merge upstream/master`

```
Switched to a new branch 'feature2'
GIT $git merge upstream/master
Auto-merging main.c
CONFLICT (content): Merge conflict in main.c
Automatic merge failed; fix conflicts and then commit the result.
GIT $
```

```
GIT $git add main.c
GIT $git commit -m "Resolve merge conflicts and apply security patch to feature2"
[feature2 4396576] Resolve merge conflicts and apply security patch to feature2
GIT $
```

Apply Changes of Feature1 and Feature2 Branch on Master:

- Checkout the master branch and merge feature1:

`git checkout master`

```
GIT $git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
GIT $
```

`git merge feature1`

```
GIT $git merge feature1
Updating f71ccdc..dbc5b51
Fast-forward
 main.c | 6 ++++++
 1 file changed, 6 insertions(+)
GIT $|
```

Merge feature2 branch into master:

`git merge feature2`

```

GIT $git merge feature2
Auto-merging main.c
CONFLICT (content): Merge conflict in main.c
Automatic merge failed; fix conflicts and then commit the result.
GIT $git add main.c
GIT $git commit -m "Merge feature1 into master"
[master e3ea839] Merge feature1 into master
GIT $git commit -m "Merge feature2 into master"
On branch master
Your branch is ahead of 'origin/master' by 5 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
GIT $git merge feature2
Already up to date.

```

Push All Changes to GitHub:

- Push changes of `feature1`, `feature2`, and `master` branches to your GitHub repository:

`git push origin feature1`

```

GIT $git push origin feature1
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 420 bytes | 210.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature1' on GitHub by visiting:
remote:   https://github.com/poulomi-design999/merge-conflict/pull/new/feature1
remote:
To https://github.com/poulomi-design999/merge-conflict.git
 * [new branch]      feature1 -> feature1
GIT $

```

`git push origin feature2`

```

GIT $git push origin feature2
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 420 bytes | 420.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature2' on GitHub by visiting:
remote:   https://github.com/poulomi-design999/merge-conflict/pull/new/feature2
remote:
To https://github.com/poulomi-design999/merge-conflict.git
 * [new branch]      feature2 -> feature2
GIT $

```

`git push origin master`

```
GIT $git push origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 360 bytes | 360.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/poulomi-design999/merge-conflict.git
    f71ccdc..e3ea839  master -> master
GIT $
```

Github link : <https://github.com/poulomi-design999/merge-conflict/tree/master>