

Practice Project: Part 2 - Git CLI



Estimated Time: 45 minutes

Overview

Congratulations on starting your company's journey with Git and GitHub by open-sourcing the Shipping Cost Calculator Python script. All the developers have contributed to the repository, and their changes have been accepted and merged into a new global repository. Now, you have been asked to edit some of the code and also add few more files. For this, you will fork this repository, make the necessary edits, add files using Git CLI in the provided lab environment, and open a pull request.

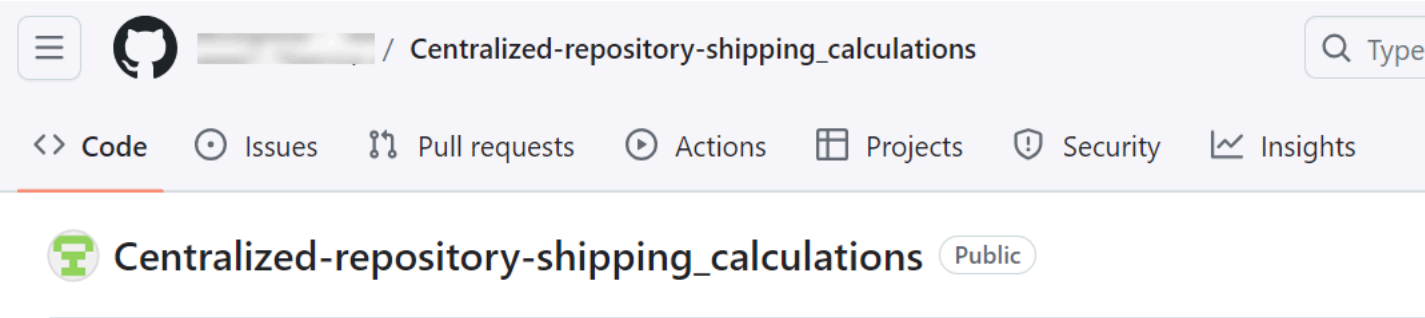
Objectives

After completing this lab, you will be able to demonstrate that you can:

- 1. Fork the upstream repository into your own account.
- 2. Clone the code locally in the lab environment.
- 3. Create a branch in the repository.
- 4. Merge the branch back into the main branch.
- 5. Create a pull request from the forked repository to the upstream repository.
- 6. Revert a change that you made earlier.

Task 1: Fork the repository

- 1. Log in to GitHub and go to the [Central Repository](#).
- 2. Click on Fork to create a personal copy of a GitHub repository in your account. This allows you to make changes independently.



- 3. Now click on Create Fork.

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.

Required fields are marked with an asterisk (*).

Owner *

Repository name *

Centralized-repository-shi

✔ Centralized-repository-shipping_calculations 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 Centralized-repository-shipping_calculations by adding your own branch. [Learn more.](#)

i

You are creating a fork in your personal account.

Create fork

NOTE: In real-life scenarios you will have the flexibility to change the repository name. However, in this Practice Project, it's recommended to keep the same repository name to ensure you complete the Project correctly and enhance your understanding.

Task 2: Create a new branch and submit the first pull request

1. Open the terminal and clone your forked repository in the lab environment. Make sure to replace <your GitHub username> with your actual GitHub username.

```
1. 1
1. git clone https://github.com/<your GitHub username>/Centralized-repository-shipping_calculations.git
```

Copied!

2. Switch to the **Centralized-repository-shipping_calculations** directory by using the `cd` command.

```
1. 1
1. cd Centralized-repository-shipping_calculations
```

Copied!

3. Create and switch to a new branch named **shipping_calculator_fixes** using the `git checkout -b` command.

```
1. 1
1. git checkout -b shipping_calculator_fixes
```

Copied!

- Verify the current branch and view all branches with the command `git branch`.

```
theia@theia-ksundararaja:/home/project$ cd Centralized-repository-shipping_calculations
theia@theia-ksundararaja:/home/project/Centralized-repository-shipping_calculations$ git checkout -b shipping_calculator_fixes
Switched to a new branch 'shipping_calculator_fixes'
theia@theia-ksundararaja:/home/project/Centralized-repository-shipping_calculations$ git branch
main
* shipping_calculator_fixes
theia@theia-ksundararaja:/home/project/Centralized-repository-shipping_calculations$
```

4. Insert the following line into the `Shipping_Cost_Calculator.py` file, replacing <your GitHub username> with your real GitHub username.

```
1. 1
1. # Here is a new update by <your GitHub username>
```

Copied!

5. To configure your Git user information, execute the given commands:

```
1. 1
1. git config --global user.email "your-email@example.com"
```

Copied!

1. 1

1. `git config --global user.name "YourName"`

Copied!

NOTE: Please ensure that you replace "[your-email@example.com](#)" with your actual GitHub email ID and "YourName" with your actual GitHub username.

6. Add and commit the code changes you have made, and check the commit status:

1. 1
2. 2
3. 3

1. `git add .`
2. `git commit -m "added a new line to Shipping_Cost_Calculator.py"`
3. `git status`

Copied!

- `git add`: Stages changes made to `Shipping_Cost_Calculator.py` for the next commit."
 - `git commit`: `git commit` records the staged changes in `Shipping_Cost_Calculator.py` along with a specified message ('added a new line to `Shipping_Cost_Calculator.py`').
 - `git status`: `git status` displays the current status and changes of `Shipping_Cost_Calculator.py` within the repository.
7. Use the command below to push the changes you made above in the lab environment or your local system IDE to the `shipping_calculator_fixes` branch in your Github repository, which is tracked as the upstream branch.

1. 1

1. `git push --set-upstream origin shipping_calculator_fixes`

Copied!

-set-upstream:

This flag is used to set up a tracking relationship between the local branch and the remote branch allowing you to use *git push* and *git pull* without specifying the remote branch every time.

origin:

This is the default name given to the remote repository from which you originally cloned your local repository.

shipping_calculator_fixes:

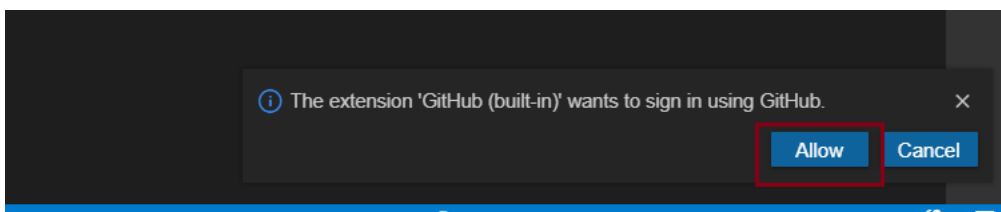
This is the name of the local branch that you want to push to the origin (ie. remote) repository. It represents the set of changes you have made locally and want to update on the remote server.

- In the provided command, you're pushing the changes from the *shipping_calculator_fixes* branch (created in your lab environment or local system IDE) to the remote repository (https://github.com/<your GitHub username>/Centralized-repository-shipping_calculations.git) which you forked in Task 1.
- The `--set-upstream` flag in the command establishes a tracking link between the *shipping_calculator_fixes* branch (in the lab environment or your local system) and the remote branch with the same name (*shipping_calculator_fixes*) in the origin repository. This streamlines future operations by eliminating the requirement to explicitly mention the name of the remote branch (ie. *shipping_calculator_fixes*) every time you run `git pull` or `git push`.

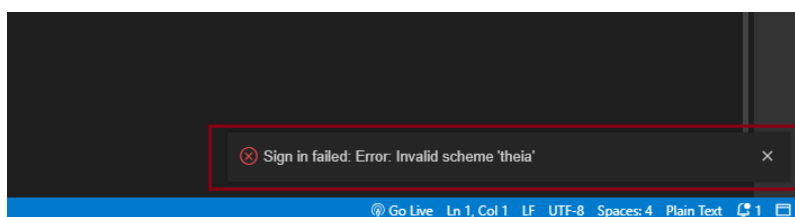
▼ Please follow the appropriate process, as mentioned here:

Scenario 1: If you encounter the 'Allow' prompt, kindly proceed with the following steps:

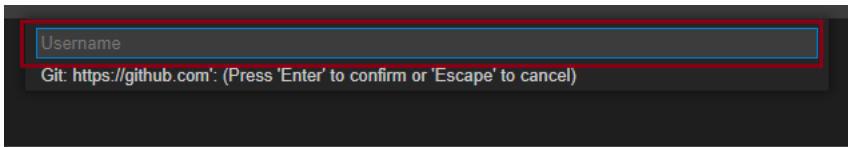
1. You will see the below pop-up in the Theia environment. Click on Allow.



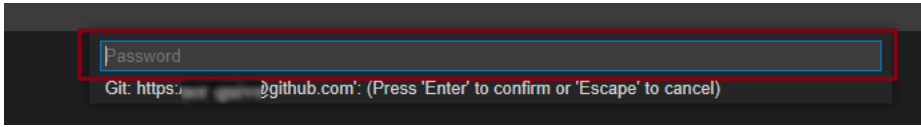
Please disregard this message as it is unrelated to the Git commands and has no impact on the process.



2. At the top, you'll receive a prompt. Enter your GitHub username and then press 'Enter'.

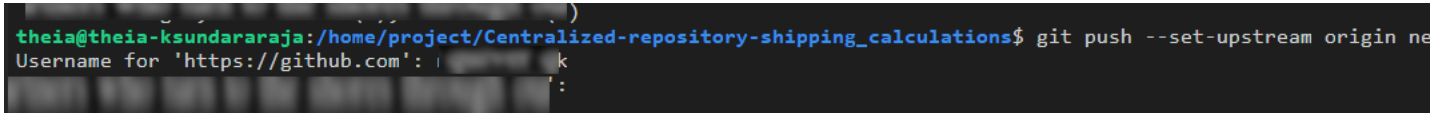


3. Another prompt will appear at the top. Enter your GitHub PAT (Personal Access Token) as password and then press 'Enter'.

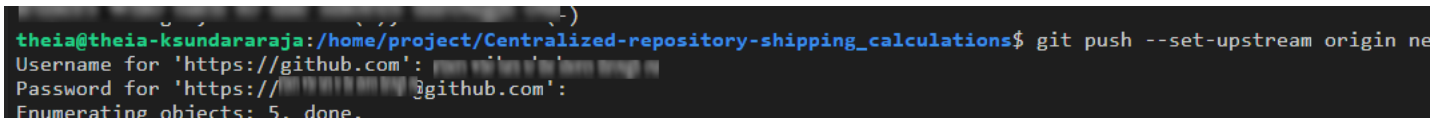


Scenario 2: You will get the prompts in the terminal:

1. In the terminal, a prompt will appear. Enter your GitHub username and press 'Enter'.

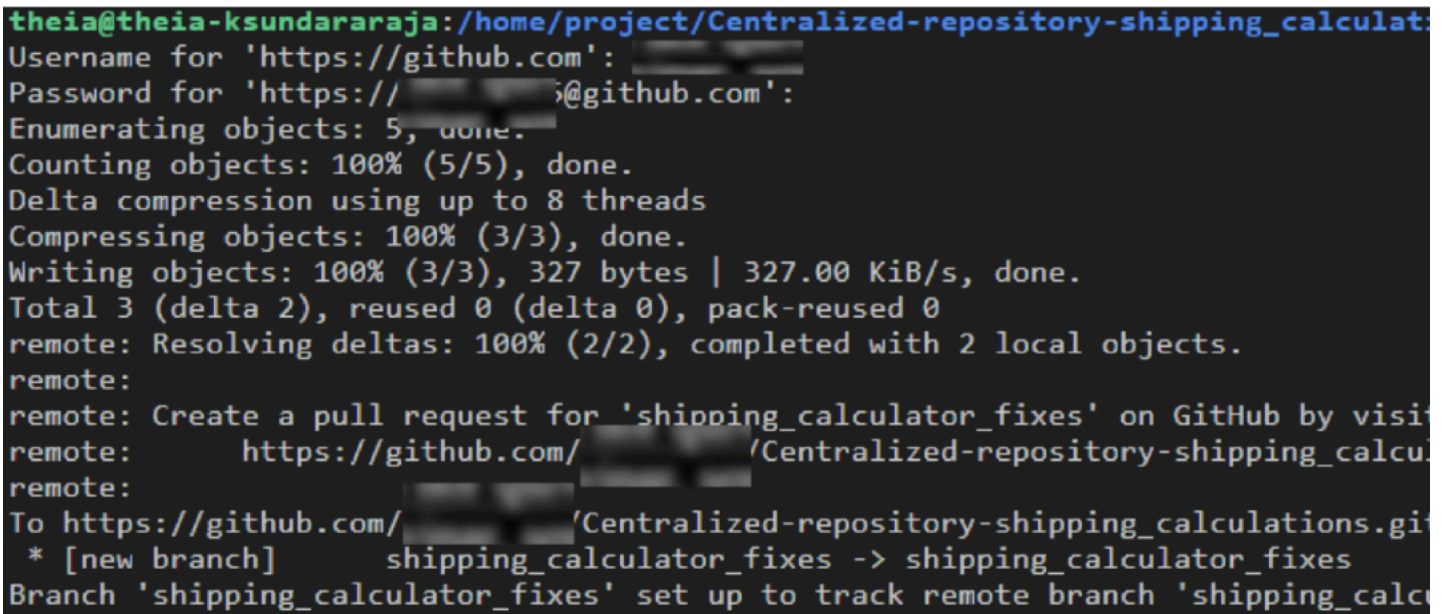


2. In the terminal, you'll receive another prompt. Enter your GitHub PAT (Personal Access Token) and press 'Enter'.



When entering or pasting your PAT (Personal Access Token) in the terminal, the token won't be visible for security reasons. Therefore, ensure accurate entry or pasting before pressing 'Enter'.

This ensures the successful pushing of your changes to the remote origin repository.



8. Generate and merge a pull request to incorporate the changes from the shipping_calculator_fixes branch into your main branch.

NOTE: In the past, the default branch in your GitHub repo used the name master. Effective Oct 1, 2020, all new GitHub repositories use the more inclusive term main as the name of the default branch instead of master.

- Please refer to Tasks 7 and 8 of [Practice Project Part 1](#) to complete this process.

▼ Your pull request should look like the example shown here:

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: shipping_calculator_fixes

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

Add a title

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

Reviewers

No reviews

Assignees

No one—[assign yourself](#)

Labels

None yet

Projects

None yet

Milestone

No milestone

Helpful resources

[GitHub Community Guide](#)

Task 3: Commit another update and revert it

1. Add the following comment to the `Shipping_Cost_Calculator.py` file.

```
1. 1
```

```
1. # Here is another update by <your GitHub username>
```

Copied!

2. Add and commit the new changes:

▼ Click here to view the solution

```
1. 1
```

```
1. git add .
```

Copied!

```
1. 1
```

```
1. git commit -m "added additional line to Shipping_Cost_Calculator.py"
```

Copied!

However, you now believe that this recently added line is unnecessary, and you can include it later.

- One method to achieve this is to remove the line you added, and then include and commit the changes again.
- **Instead, you will now utilize the `git revert` command to accomplish the same functionality.**

3. Execute the `git revert` command as below:

```
1. 1
```

```
1. git revert HEAD --no-edit
```

Copied!

- `git revert`: is a Git command used to undo a previous commit by creating a new commit that represents the inverse of the specified commit.
- `HEAD`: This refers to the latest commit in the current branch.
- `--no-edit`: This option is used to perform the revert without opening the default text editor to edit the commit message.

After running this command, Git will automatically generate a commit message indicating that it's a revert and will create a new commit with the changes undone. The `--no-edit` option prevents Git from opening an editor for you to modify the commit message; instead, it uses the default revert message.

4. This will result in the following output:

```

theia@theia-ksundararaja:/home/project/Centralized-repository-shipping_calculations$ g
theia@theia-ksundararaja:/home/project/Centralized-repository-shipping_calculations$ g
[shipping_calculator_fixes f71c81b] added additional line to Shipping_Cost_Calculator.
1 file changed, 2 insertions(+)
theia@theia-ksundararaja:/home/project/Centralized-repository-shipping_calculations$ g
[shipping_calculator_fixes 56509c5] Revert "added additional line to Shipping_Cost_Cal
Date: Tue Nov 14 00:59:26 2023 -0500
1 file changed, 2 deletions(-)
theia@theia-ksundararaja:/home/project/Centralized-repository-shipping_calculations$

```

- You will observe that the extra commit has been effectively undone. If you inspect the `Shipping_Calculator.py` file, the extra line you added will no longer be present.
- Now, you are back to your initial commit and can proceed to push it to your repository.

5. Switch to the main branch using the `git checkout` command:

- 1
1. `git checkout main`

Copied!

6. Execute the `git merge` command to push the updates from the `shipping_calculator_fixes` branch into your main branch.

- 1
1. `git merge shipping_calculator_fixes`

Copied!

```

theia@theia-ksundararaja:/home/project/Centralized-repository-shipping_calculations$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
theia@theia-ksundararaja:/home/project/Centralized-repository-shipping_calculations$ git merge shipping_calculator_fixes
Updating 45a5b23..56509c5
Fast-forward
 Shipping_Cost_Calculator.py | 4 +++-
1 file changed, 3 insertions(+), 1 deletion(-)

```

7. Run the `git log` command to obtain a chronological record of all commits. This will display information about the line that was added and subsequently undone through the `git revert` command.

- 1
1. `git log`

Copied!

```

theia@theia-ksundararaja:/home/project/Centralized-repository-shipping_calculations$ git log
commit 56509c5e384426de3d41b76d5708ad03e6c2e6e7 (HEAD -> main, shipping_calculator_fixes)
Author: 
Date: Tue Nov 14 00:59:26 2023 -0500

    Revert "added additional line to Shipping_Cost_Calculator.py"

    This reverts commit f71c81b67c7a3e12e15f465be2442375ac9f7e0d.

commit f71c81b67c7a3e12e15f465be2442375ac9f7e0d
Author: 
Date: Tue Nov 14 00:59:19 2023 -0500

    added additional line to Shipping_Cost_Calculator.py

```

Note: To exit the `git log` command, simply press the “Q” key. This action will close the log view and bring you back to the command prompt.

With this action, all the commits from the `shipping_calculator_fixes` branch have been successfully pushed to the main branch in your forked repository.

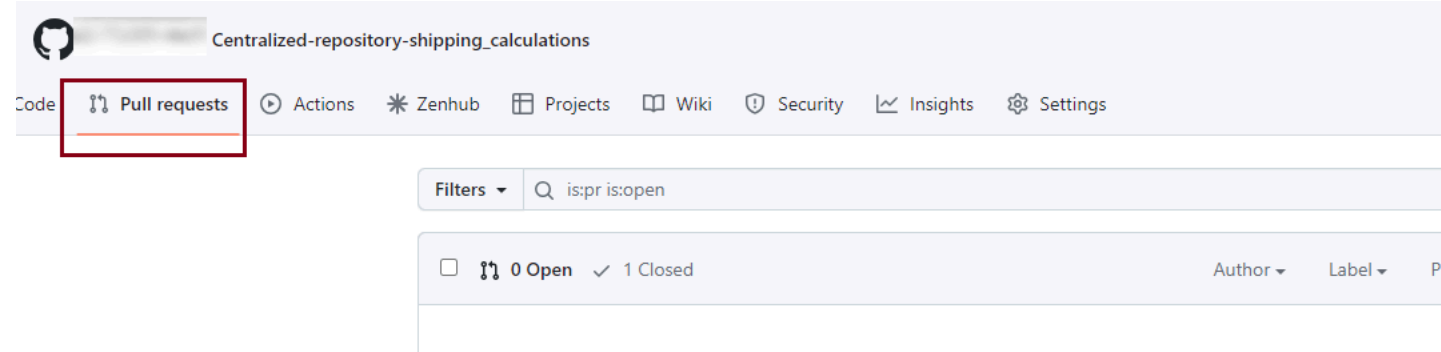
Task 4: Raise a pull request from your repository to the centralized collaboration repository

Now, you will contribute your project to the centralized collaboration repository for consideration and acceptance by the company authorities.

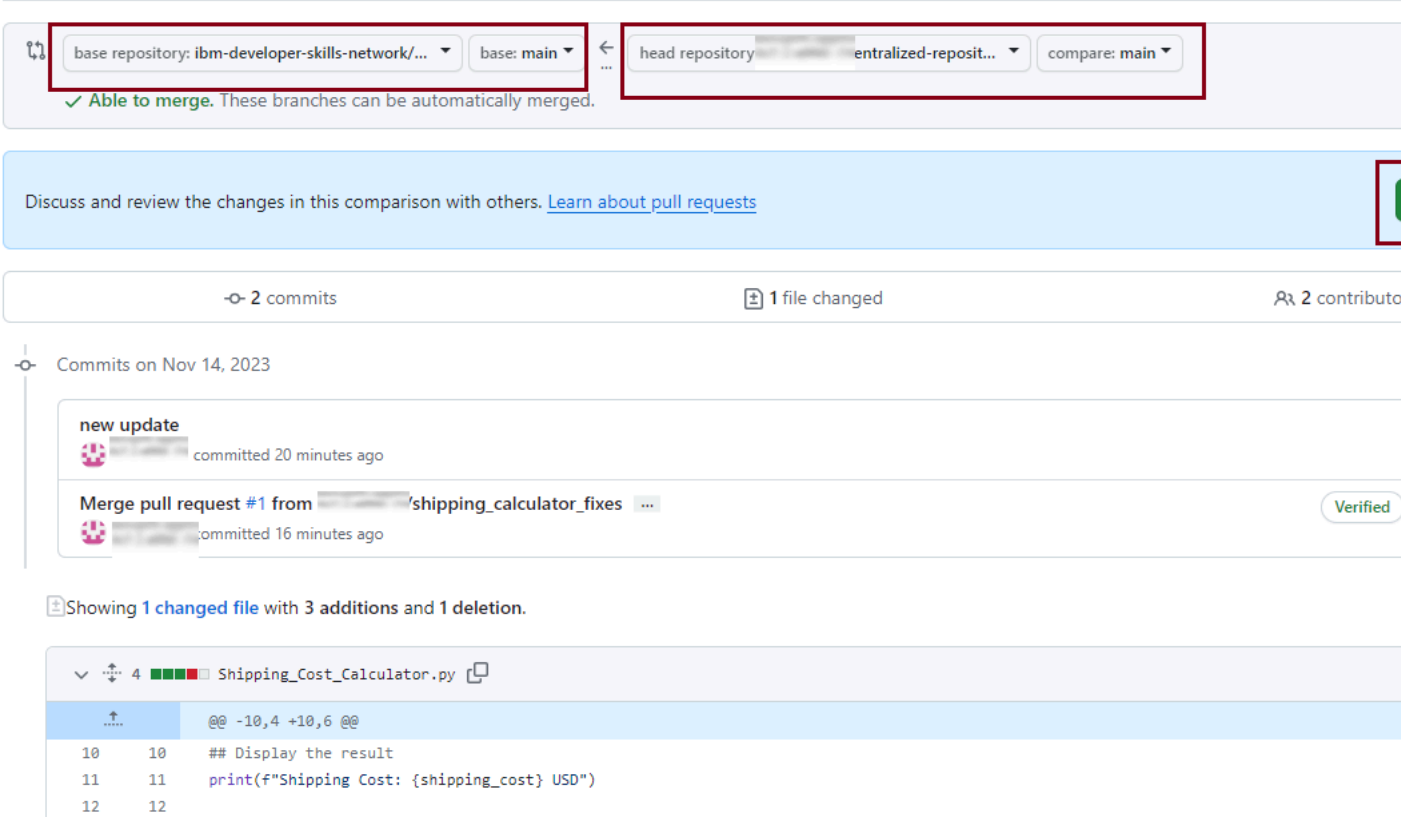
Since this is the centralized repo, only the company authorities have write and Admin access to it.

You can only raise pull requests to this and do not have access to merge it.

1. Navigate to Pull requests and click on New pull request in your GitHub repository.



2. Select compare across forks, pick the relevant fields from the dropdown, and then click on Create pull request.



- The head repository and the compare fields refer to the repository URL and branch, respectively, from where you want to initiate the pull request. In this instance, it is your GitHub repository.
- The base repository and the base fields refer to the repository URL and branch, respectively, of the upstream repository where you intend to submit a pull request. In this instance, it is the centralized repository that you forked during Task 1.

3. Include an appropriate comment and proceed to click on Create pull request.

base repository: ibm-developer-skills-network/... base: main head repository: Centralized-reposit... compare: main

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

Add a title

Pull request to centralized repository

Add a description

WritePreview

HBIIE<>|:|:|@|↩|

Add your description here...

☒ Allow edits by maintainers

Create pull request

Remember, contributions to this repository should follow its contributing guidelines and code of conduct.

4. The displayed interface indicates that you have successfully submitted your latest update to your company’s centralized repository for collaboration.

Pull request to centralized repository #4

Open

wants to merge 2 commits into ibm-developer-skills-network:main from :main

Conversation0

Commits2

Checks0

Files changed1

commented now

No description provided.

and others added 2 commits 37 minutes ago

new update

Merge pull request #1 from shipping_calculator_fixes Verified

Add more commits by pushing to the main branch on Centralized-repository-shipping_calculations.

This branch has no conflicts with the base branch

Only those with [write access](#) to this repository can merge pull requests.

5. The pull request will undergo review and subsequently be merged by the administrators of the repository.

Summary

In this lab, you forked the repository to your account and cloned it locally. You then created a new branch for making changes. After completing your modifications, you merged the branch back into main. You also learned to create pull requests for collaboration. Finally, you explored the process of reverting changes when needed.

Author(s)

Nikesh Kumar
K Sundararajan

about:blank

8/8