

Hands-on Lab: Get familiar with fork and pull requests



Estimated time needed: 30 mins

Objectives

After completing this lab, you will be able to:

1. Use git commands to manage upstream repositories
2. Create a personal access token
3. Fork existing repository using the UI
4. Clone forked repository in the lab environment
5. Create a new branch
6. Make changes locally
7. Add and commit to the local branch
8. Push changes to the forked repository
9. Create a pull request to the upstream repository

Pre-requisites

This lab is designed to run on Skills Network - Cloud IDE which runs on a Linux system in the cloud and already has git installed. If you intend to run this lab on your own system, please ensure you have git (on Linux or macOS) or GitBash (on Windows) installed.

Note: While the lab allows you to copy-paste the commands, the best way to learn is to type the command yourselves. The instructors highly recommend the same.

Exercise 1: Generate personal access token

The first step is to generate an access token from GitHub.com. Follow the lab named [Generate GitHub personal access token](#) and copy the access token to use as a password in the upcoming exercises.

Exercise 2: Fork the repository

To fork a source repository, complete the following steps:

1. Log in to GitHub and go to this project's [sample source repository](#). This is the upstream repository for your project.
2. At the top right of the screen, click Fork and select your own GitHub account as the destination for the fork.

The screenshot shows the GitHub repository page for `ibm-developer-skills-network / gkpbt-css-circle`. The repository is public and has 2 watchers and 1 fork. The `Fork` button is highlighted with a red box. Below the repository name, there is a navigation bar with links to Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The main content area shows the repository's structure with files and folders, including `.github/workflows`, `.gitignore`, `LICENSE`, `README.md`, `circle.html`, and `style.css`. The `README.md` file is selected, and its content is displayed below. The right sidebar shows the repository's metadata, including the name `css-circle`, a README link, Apache-2.0 License, 0 stars, 2 watchers, and 1 fork. There are also sections for Releases and Packages, both indicating no published items.

ibm-developer-skills-network / gkpbt-css-circle Public

generated from ibm-developer-skills-network/coding-project-template

Unwatch 2 Fork 1

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

upkarlidderr Automatically close PRs 729ceb2 2 days ago 5 commits

.github/workflows	Automatically close PRs	2 days ago
.gitignore	Initial commit	3 days ago
LICENSE	Initial commit	3 days ago
README.md	Update README.md	3 days ago
circle.html	Create circle.html	3 days ago
style.css	Create style.css	3 days ago

README.md

Readme

css-circle

Readme

Apache-2.0 License

0 stars

2 watching

1 fork

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

A copy of the source repository has now been added as one of your GitHub repositories. This is the origin repository.

upkarlidderr / gkpbt-css-circlePublicforked from ibm-developer-skills-network/gkpbt-css-circle

<> Code

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

main1 branch0 tags

Go to fileAdd fileCode

This branch is up to date with ibm-developer-skills-network:main.

ContributeFetch upstream

upkarlidderr Automatically close PRs729ceb2 2 days ago5 commits

.github/workflows	Automatically close PRs	2 days ago
.gitignore	Initial commit	3 days ago
LICENSE	Initial commit	3 days ago
README.md	Update README.md	3 days ago
circle.html	Create circle.html	3 days ago
style.css	Create style.css	3 days ago

README.md

Readme

About

css-circle

Readme

Apache-2.0 License

0 stars

0 watching

1 fork

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Languages

HTML69.9%

CSS30.1

Exercise 3: Clone the forked repository

A clone is a local copy of a repository. Before you can clone the forked repository, you first need its HTTPS URL, which provides secure access to it.

To clone the forked repository, complete the following steps:

1. In your list of repositories, click the forked repository. On the repository's main page, click the **Code** button.
2. Click the clipboard icon to copy the URL. Make sure the HTTPS tab is active.

upkarliddler / gkpbt-css-circle Public
forked from ibm-developer-skills-network/gkpbt-css-circle

Code Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code

This branch is up to date with ibm-developer-skills-network:main

upkarliddler Automatically close PRs

.github/workflows	Automatically close PRs	
.gitignore	Initial commit	
LICENSE	Initial commit	3 days ago
README.md	Update README.md	3 days ago
circle.html	Create circle.html	3 days ago
style.css	Create style.css	3 days ago

Clone
HTTPS SSH GitHub CLI
https://github.com/upkarliddler/gkpbt-css-circle
Use Git or checkout with SVN using the web URL.
Download ZIP

README.md

Readme

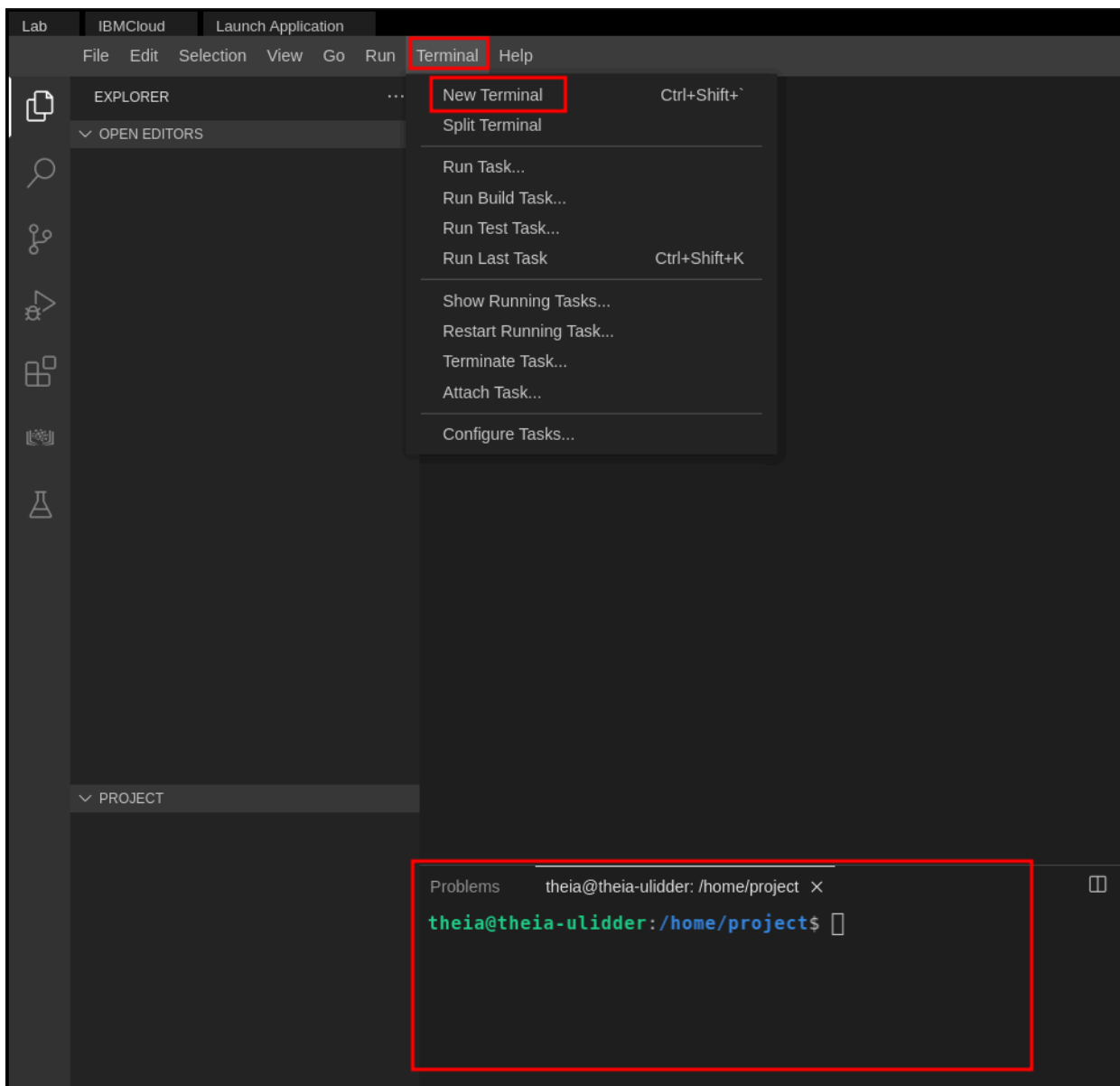
About
css-circle
Readme
Apache-2.0 License
0 stars
0 watching
1 fork

Releases
No releases published
[Create a new release](#)

Packages
No packages published
[Publish your first package](#)

Languages
HTML 69.9%

3. Open the terminal in the lab environment by using the menu in the editor: Terminal > New Terminal.



4. Let's export the copied URL in an environment variable so it's available for us to use in the later steps, run the following command in terminal:

```
export ORIGIN=<your repository HTTPS URL>
```

Replace <your repository HTTPS URL> with the URL you copied in step 2.

5. Run the following command with the HTTPS URL you copied earlier:

```
git clone $ORIGIN
```

```
Problems theia@theia-ulidder: /home/project X

theia@theia-ulidder:/home/project$ export ORIGIN=https://github.com/upkarlidder/gkpbt-css-circle.git
theia@theia-ulidder:/home/project$ git clone $ORIGIN

Cloning into 'gkpbt-css-circle'...
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 22 (delta 5), reused 7 (delta 1), pack-reused 0
Unpacking objects: 100% (22/22), done.

theia@theia-ulidder:/home/project$ ls -la

total 16

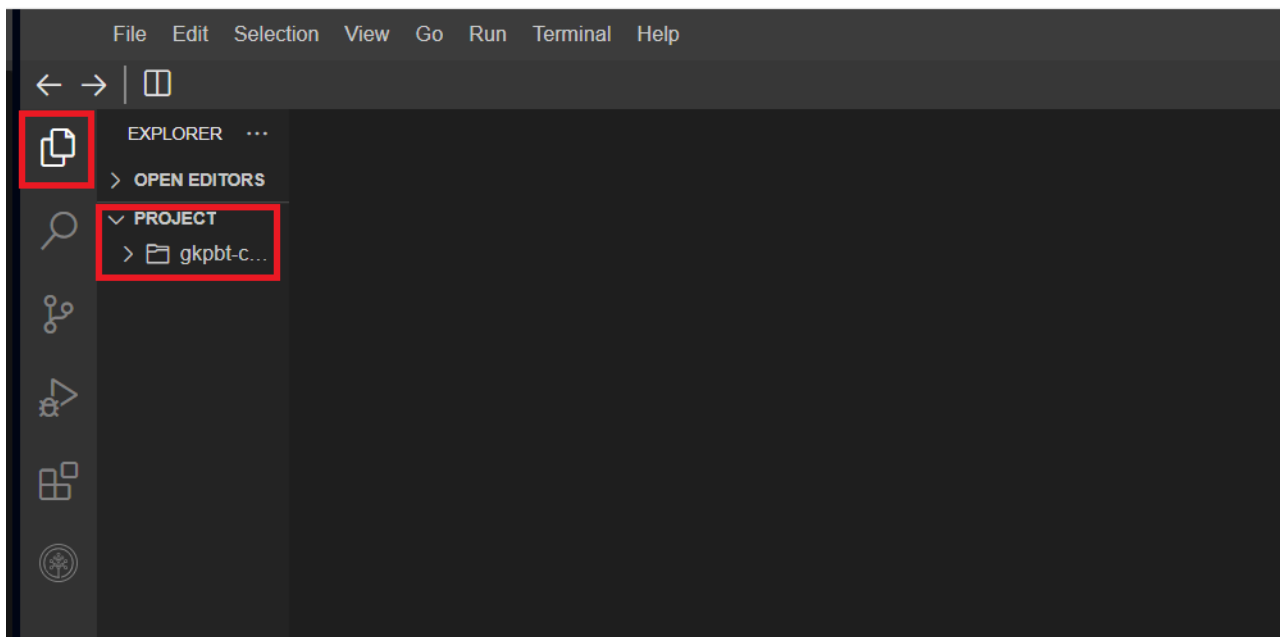
drwxrwsrwx 3 root  users 4096 Jan 18 16:41 
drwxrwxr-x 1 root  root  4096 Jan 10 21:57 ..
drwxr-sr-x 4 theia users 4096 Jan 18 16:41 gkpbt-css-circle
```

The command clones the repository that is on GitHub into your current directory.

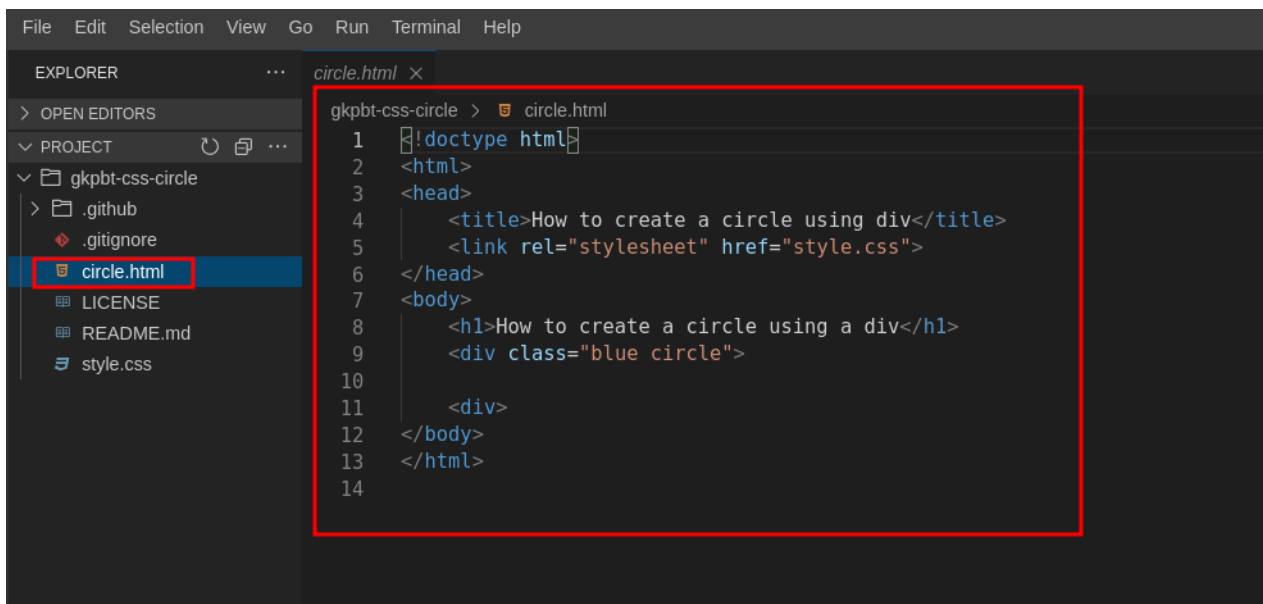
Exercise 4: Explore the cloned repo

To become familiar with the cloned repo, complete the following steps:

1. Click the Explorer icon as shown in the following image:



2. Click Project and expand the folder of the project you just cloned. You can open the files in the editor, on the right side, by clicking on the file name.



Exercise 5: Create the feature-circle-500 branch

We will now add a new feature to the source code. We will increase the circle's size to 500x500 pixels. Before we make this change, we will create a new branch.

1. Navigate to our repository using this command `cd gkpbt-css-circle`
2. Create a new branch using the `git checkout -b feature-circle-500` command. Notice that we used a single command instead of creating a branch and then checking it out. The `-b` flag creates the branch if it does not already exist.
3. You can check that you are in the new branch by using the `git branch` command.

```
Problems theia@theia-ulidder: /home/project/gkpbt-css-circle X
theia@theia-ulidder:/home/project/gkpbt-css-circle$ git checkout -b feature-circle-500
Switched to a new branch 'feature-circle-500'
theia@theia-ulidder:/home/project/gkpbt-css-circle$ git branch
* feature-circle-500
main
```

Exercise 6: Make required code changes

1. Let's change the width and height to 500px each. Open the `style.css` file from the file explorer and change the code as follows:

```
.blue {
  background-color:blue
}
.circle{
  border-radius:50%;
  width:500px;
  height:500px;
}
```

2. If you do a `git status` at this point, you will see a change is shown. This change is not staged at this point, but Git is aware of it.

```
git status
```

```

theia@theia-ulidder:/home/project/gkpbt-css-circle$ git status
On branch feature-circle-500
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

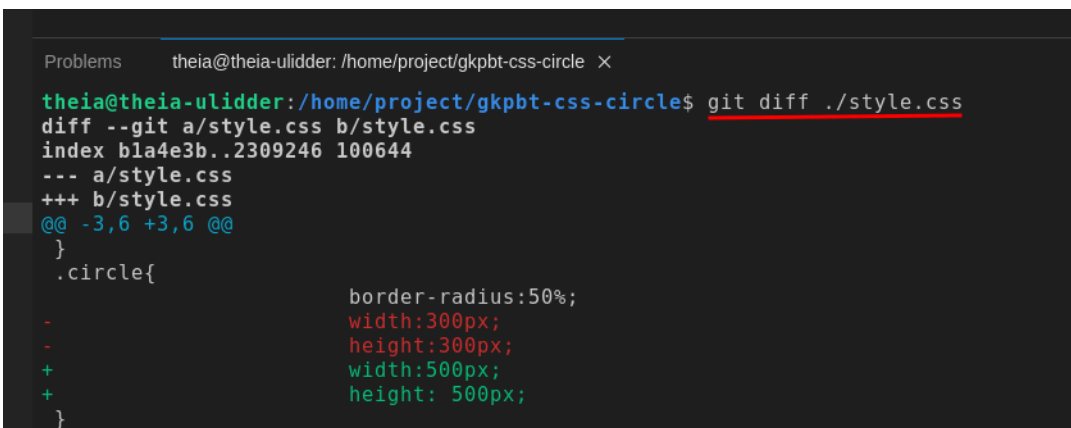
        modified:   style.css

no changes added to commit (use "git add" and/or "git commit -a")

```

- Optionally, you can use the `git diff` command to see the detailed changes:

```
git diff ./style.css
```



```

theia@theia-ulidder:/home/project/gkpbt-css-circle$ git diff ./style.css
diff --git a/style.css b/style.css
index bla4e3b..2309246 100644
--- a/style.css
+++ b/style.css
@@ -3,6 +3,6 @@
 }
 .circle{
-        border-radius:50%;
-        width:300px;
-        height:300px;
+        border-radius:50%;
+        width:500px;
+        height: 500px;
 }

```

Notice the text in red was deleted and the text in green was added. Essentially, we changed the height and width from 300px to 500px each.

Note: To exit the `git diff` command, simply press the "Q" key.

Exercise 7: Add and commit your changes

A commit is Git's way of recording your file changes, similar to how you might save an edited document. To commit the change that you made in the previous exercise, you first need to add it to a staging area. Git will then take the staged snapshot of changes and commit them to the project. Remember, Git will never change files unless you explicitly ask it to.

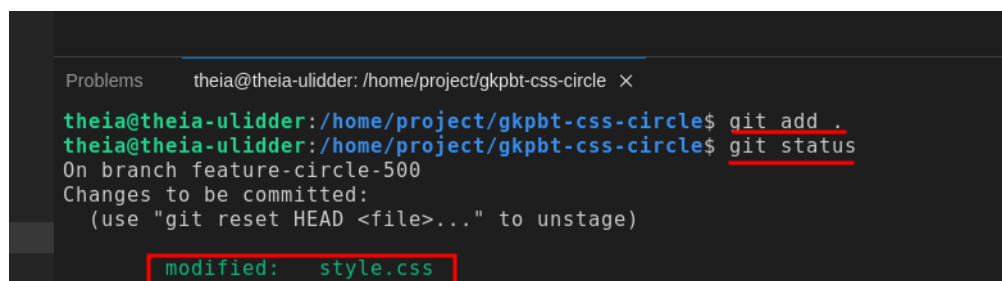
To commit your new file, complete the following steps:

- To move the changes from your working project directory to the staging area, type the following command in the Terminal window:

```
git add .
```

The `git add` command has several options. The single `.` adds all untracked files in the current directory and subdirectories to the staging area. Alternatively, you can add the single file you created by using the `git add style.css` command. Finally, you can use `git add -A` to recursively add all files from the top level git folder.

- If you check the status at this point, you will see the file has changed from Untracked to Changes to be committed:



```

theia@theia-ulidder:/home/project/gkpbt-css-circle$ git add .
theia@theia-ulidder:/home/project/gkpbt-css-circle$ git status
On branch feature-circle-500
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   style.css

```

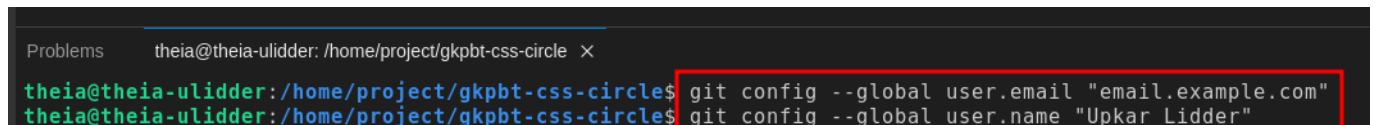
- To commit the new file to the local repository, you need to first tell git who you are. Type in the following commands to set your email and username. The email should be the same as your GitHub email.

Set your email:

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

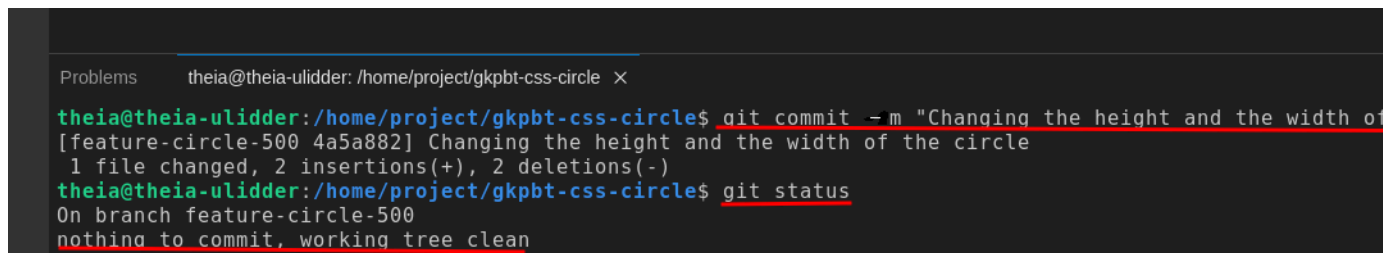
Set your name:

```
git config --global user.name "Your Name"
```

A terminal window with a dark background. The title bar shows 'Problems' and 'theia@theia-ulidder: /home/project/gkpbt-css-circle X'. The terminal shows two lines of command input: 'theia@theia-ulidder: /home/project/gkpbt-css-circle\$ git config --global user.email "email@example.com"' and 'theia@theia-ulidder: /home/project/gkpbt-css-circle\$ git config --global user.name "Upkar Lidder"'. Both lines are enclosed in a red rectangular box.

```
theia@theia-ulidder: /home/project/gkpbt-css-circle$ git config --global user.email "email@example.com"
theia@theia-ulidder: /home/project/gkpbt-css-circle$ git config --global user.name "Upkar Lidder"
```

4. Type the following command in the Terminal window to commit the file.
Note: It's always a good practice to add a description for the commit so you can remember what the change was if you have to refer to it later.
- **-m flag:** It is used in Git commit commands to specify the commit message directly in the command line, allowing you to provide a brief description of the changes you are committing.
- ```
git commit -m "Changing the height and the width of the circle"
```

A terminal window with a dark background. The title bar shows 'Problems' and 'theia@theia-ulidder: /home/project/gkpbt-css-circle X'. The terminal shows the following commands and output: 'theia@theia-ulidder: /home/project/gkpbt-css-circle\$ git commit -m "Changing the height and the width of the circle"', followed by '[feature-circle-500 4a5a882] Changing the height and the width of the circle', '1 file changed, 2 insertions(+), 2 deletions(-)', 'theia@theia-ulidder: /home/project/gkpbt-css-circle\$ git status', 'On branch feature-circle-500', and 'nothing to commit, working tree clean'. The last three lines of output are enclosed in a red rectangular box.

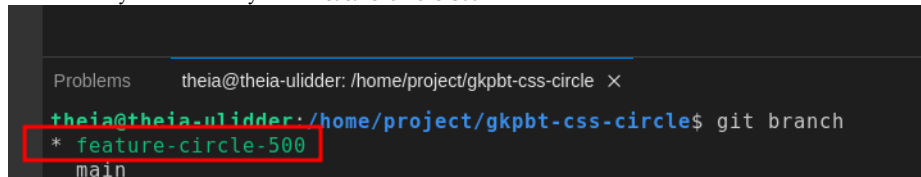
```
theia@theia-ulidder: /home/project/gkpbt-css-circle$ git commit -m "Changing the height and the width of the circle"
[feature-circle-500 4a5a882] Changing the height and the width of the circle
1 file changed, 2 insertions(+), 2 deletions(-)
theia@theia-ulidder: /home/project/gkpbt-css-circle$ git status
On branch feature-circle-500
nothing to commit, working tree clean
```

As you can see, git status now says there is nothing to commit and the working tree is clean. The new file is now ready to be pushed from your local system to origin on GitHub.

## Exercise 8: Merge your branch back into main branch

If you are happy with your changes in the feature-circle-500 branch, you can now merge it back into your local main branch by following these steps:

1. Confirm that you are currently in the feature-circle-500 branch.

A terminal window with a dark background. The title bar shows 'Problems' and 'theia@theia-ulidder: /home/project/gkpbt-css-circle X'. The terminal shows the command 'theia@theia-ulidder: /home/project/gkpbt-css-circle\$ git branch'. The output is '\* feature-circle-500' and 'main'. The line '\* feature-circle-500' is enclosed in a red rectangular box.

```
theia@theia-ulidder: /home/project/gkpbt-css-circle$ git branch
* feature-circle-500
main
```

2. Check out the main branch

```
git checkout main
```

If you run git branch again, you should see the \* against the main branch.



```
Problems theia@theia-ullder: /home/project/gkpbt-css-circle X
theia@theia-ullder:/home/project/gkpbt-css-circle$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
theia@theia-ullder:/home/project/gkpbt-css-circle$ git branch
feature-circle-500
* main
```

3. Merge the feature-circle-500 branch into main.

```
git merge feature-circle-500
```

```
theia@theia-ullder:/home/project/gkpbt-css-circle$ git merge feature-circle-500
Updating 729ceb2..4a5a882
Fast-forward
 style.css | 4 ++--
1 file changed, 2 insertions(+), 2 deletions(-)
```

4. Confirm the change was merged by using the `git log` command. We are using `--oneline` flag to display logs more concisely.

```
Problems theia@theia-ullder: /home/project/gkpbt-css-circle X
theia@theia-ullder:/home/project/gkpbt-css-circle$ git log --oneline
4a5a882 (HEAD -> main, origin/feature-circle-500, feature-circle-500) Changing the height and the width of the circle
729ceb2 (origin/main, origin/HEAD) Automatically close PRs
0169944 Update README.md
8f09fd1 Create style.css
2d31fb1 Create circle.html
11bec50 Initial commit
```

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.

## Exercise 9: Delete the feature-circle-500 branch

Since you are done making the change, let's delete the feature-circle-500 branch by following these steps:

1. Ensure you are on the main branch. If not, check it out first

```
git checkout main
```

2. Delete the feature-circle-500 branch, the common flag used is `-d` (lowercase), which stands for "delete"

```
git branch -d feature-circle-500
```

3. You can confirm the branch was deleted by listing all branches

```
git branch
```

```
Problems theia@theia-ulidder: /home/project/gkpbt-css-circle X
theia@theia-ulidder:/home/project/gkpbt-css-circle$ git branch -d feature-circle-500
Deleted branch feature-circle-500 (was 4a5a882).
theia@theia-ulidder:/home/project/gkpbt-css-circle$ git branch
* main
```

## Exercise 10: Push your changes to origin

This push will synchronize all the changes you made on your local system with your fork repository on GitHub.

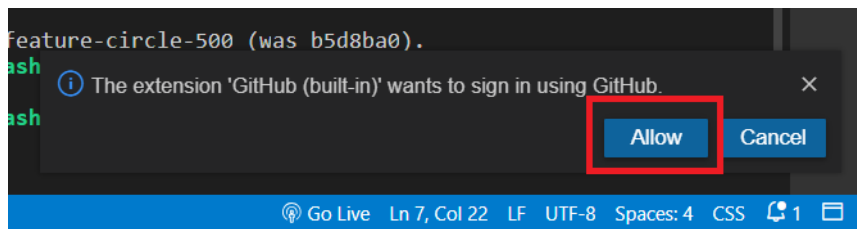
To push your update to GitHub, complete the following steps:

1. In the Terminal window, run the following command:

```
git push origin main
```

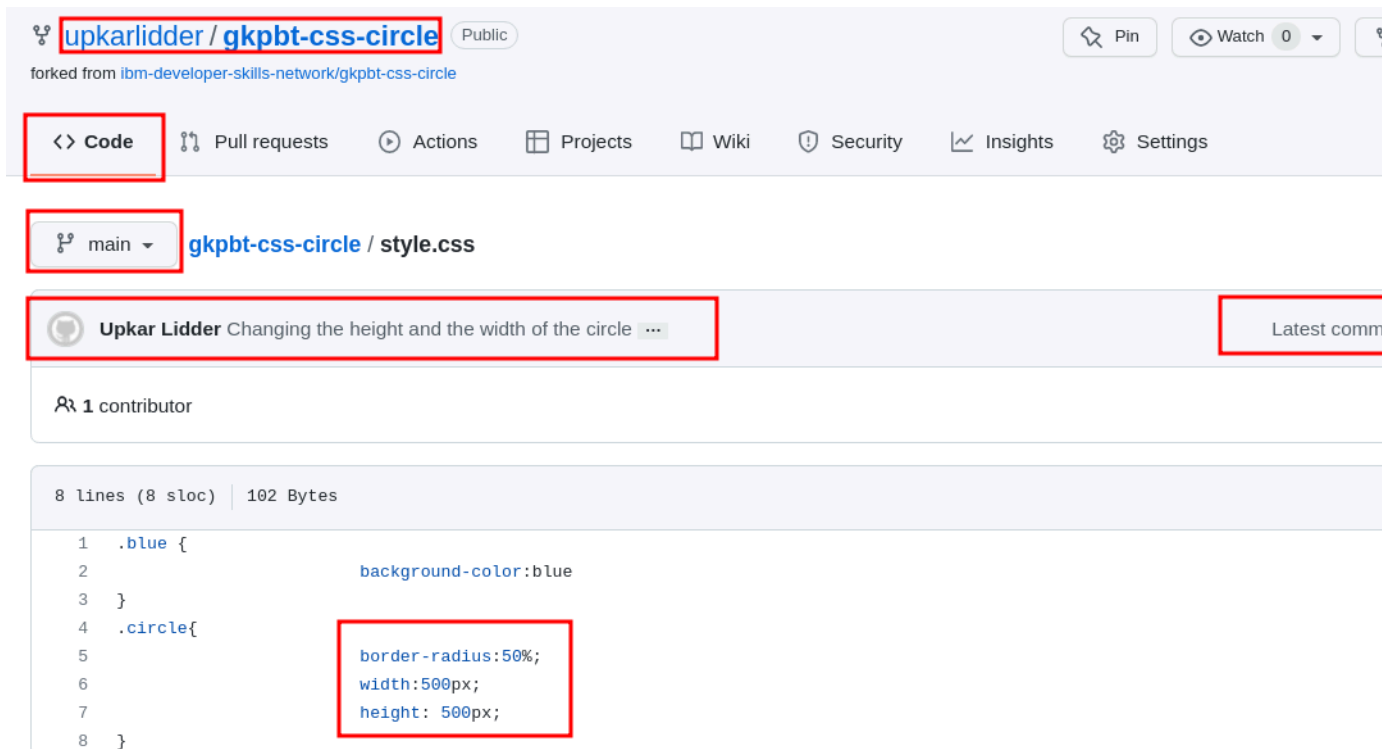
Once you submit that command, Cloud IDE will display a dialog in the lower right corner, requesting permission to sign in using GitHub. Click "Allow"

Note : If you don't see the dialog box below, you will be asked to enter your GitHub username and password in the terminal. Your PAT (Personal Access Token) will be hidden when you type or paste it in the terminal for security reasons. So, make sure you enter or paste it correctly before hitting 'Enter'.



### ► Note on Warning Messages

2. Go to the fork repository in your GitHub account and verify that the local changes have now been added to the main branch.



## Exercise 11: Create a pull request

The final step is to request the original project pull in the changes you've made to your fork. To merge your changes to the original repository, you need to create a pull request.

To create a pull request, complete the following steps:

1. Ensure you are on the **Code** tab. Click **Contribute** and then **Open pull request**.

The screenshot shows the GitHub interface for a repository named 'upkarliddler / gkpbt-css-circle', which is a public fork of 'ibm-developer-skills-network/gkpbt-css-circle'. The 'Code' tab is selected, and the 'main' branch is active. A dropdown menu is open from the 'Contribute' button, showing the message 'This branch is 1 commit ahead of ibm-developer-skills-network:main.' and a green 'Open pull request' button. The repository's commit history is visible, showing a series of commits to files like '.github/workflows', '.gitignore', 'LICENSE', 'README.md', 'circle.html', and 'style.css'. The right sidebar shows the repository's 'About' section, including a description, a list of dependencies, and a 'Release' section.

2. In the "Comparing changes" panel, GitHub shows you that it is comparing the main branch of your fork to that of the original repository, and that your changes can be merged. Click the **Create pull request** button.

**ibm-developer-skills-network / gkpbt-css-circle** Public

generated from [ibm-developer-skills-network/coding-project-template](#)

<> **Code**

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

## 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](#).

base repository: **ibm-developer-skills-network/gk...** base: **main**

←

head repository: **upkarliddar/gkpbt-css-circle** compare: **main**

✓ **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

Commits on Jan 17, 2022

Changing the height and the width of the circle

Upkar Liddar committed 1 hour ago

Showing **1 changed file** with **2 additions** and **2 deletions**.

style.css

@@ -3,6 +3,6 @@

3 }  
4 .circle{  
5 border-radius:50%;  
6 - width:300px;  
7 - height:300px;  
8 }

3 }  
4 .circle{  
5 border-radius:50%;  
6 + width:500px;  
7 + height: 500px;  
8 }

3. You are taken to the **Open pull request** screen. Notice that your commit message appears as the title of the pull request. Click the **Create pull request** button.



### Add a title

Changing the height and the width of the circle

### Add a description

Write

Preview

H B I | | @

Add your description here...

Markdown is supported

Paste, drop, or click to add files

☒ Allow edits by maintainers

Create pull request

**Note:** For the purposes of this lab, your pull request will be processed and closed automatically.

You should see the following message in your pull request after a few minutes:

ibm-developer-skills-network / gkpbt-css-circlePublic

Unwatch2Fork

generated from ibm-developer-skills-network/coding-project-template

<> Code

Issues

**Pull requests1**

Actions

Projects

Wiki

Security

Insights

Settings

## Changing the height and the width of the circle #2

Closed


upkarlidderr wants to merge 1 commit into ibm-developer-skills-network:main from upkarlidderr:main

Conversation0

Commits1

Checks0

Files changed1



upkarlidderr commented 14 seconds ago


Collaborator

...

Signed-off-by: Upkar Lidder <email.example.com>

Changing the height and the width of the circle


4a5a882



github-actions bot commented now

...

Congratulations! You have completed the lab. Closing for maintainence purpose.











github-actions bot closed this now

Pull request closed

If you wish, you can delete this fork of ibm-developer-skills-network/gkpbt-css-circle in the [settings](#).

WritePreview

H B I  <>       

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Record a Loom

Reopen pull request

Comment

Reviewers

No reviews

Assignees

No one—assign your

Labels

None yet

Projects

None yet

Milestone

No milestone

Linked issues


Successfully merging issues.

None yet

Notifications

You're receiving notif this repository.

1 participant



## Exercise 12: Practice on your own

1. Create a new branch called `feature-add-color`.
  - ▶ [Click here for the solution](#)
2. Make `feature-add-color` the active branch.
  - ▶ [Click here for the solution](#)
3. Add another css rule as follows:

```
.red {
 background-color:red
}
```

4. Stage this change.

► [Click here for the solution](#)

5. Commit the changes in your `feature-add-color`.

► [Click here for the solution](#)

6. Merge the changes in `feature-add-color` into `main`.

► [Click here for the solution](#)

7. Delete the `feature-add-color` branch.

► [Click here for the solution](#)

8. Push your changes to origin.

► [Click here for the solution](#)

9. Create a new pull request for this feature in the upstream repository using the GitHub UI.

## Summary

In this lab, you have learned how to fork an upstream repository into your own account and then clone it locally in the lab environment. You then learned how to synchronize changes in your local repository with remote GitHub repositories using pull requests.

## Author(s)

**Upkar Lidder**

## Other Contributor(s)

**Richard Ye**

© IBM Corporation. All rights reserved.