

Hands-on Lab: Getting Started with GitHub



Effort: 20 min

In this lab, you will get started with GitHub by creating a GitHub account and project and adding a file to it using its Web interface.

Objectives

After completing this lab, you will be able to:

1. Describe GitHub
2. Create a GitHub account
3. Add a Project / Repo
4. Edit / Create a file
5. Upload a file & Commit

GitHub Overview

First, let us introduce to GitHub. GitHub in simple words is a collection of folders and files. It is a Git repository hosting service, but it adds many of its own features. While Git is a command-line tool and a server needs to be hosted and maintained via command line as well, GitHub provides this Git server for you and a Web-based graphical interface. It also provides access control and several collaboration features, such as wikis and basic task management tools for every project. GitHub provides cloud storage for source code, supports all popular programming languages, and streamlines the iteration process. GitHub includes a free plan for individual developers and for hosting open source projects.

Exercise 1: Creating a GitHub Account

Please follow the steps given below to create an account in GitHub:

Step 1: Create an account: <https://github.com/join>

NOTE: If you already have a GitHub account, you can skip this step and simply login to your account.

Step 2: Provide the necessary details to create an account as shown below:

Join GitHub

Create your account

Username *

Email address *

Password *

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter.
[Learn more.](#)

Email preferences

☒ Send me occasional product updates, announcements, and offers.

Verify your account

Please solve this puzzle so we
know you are a real person

Verify



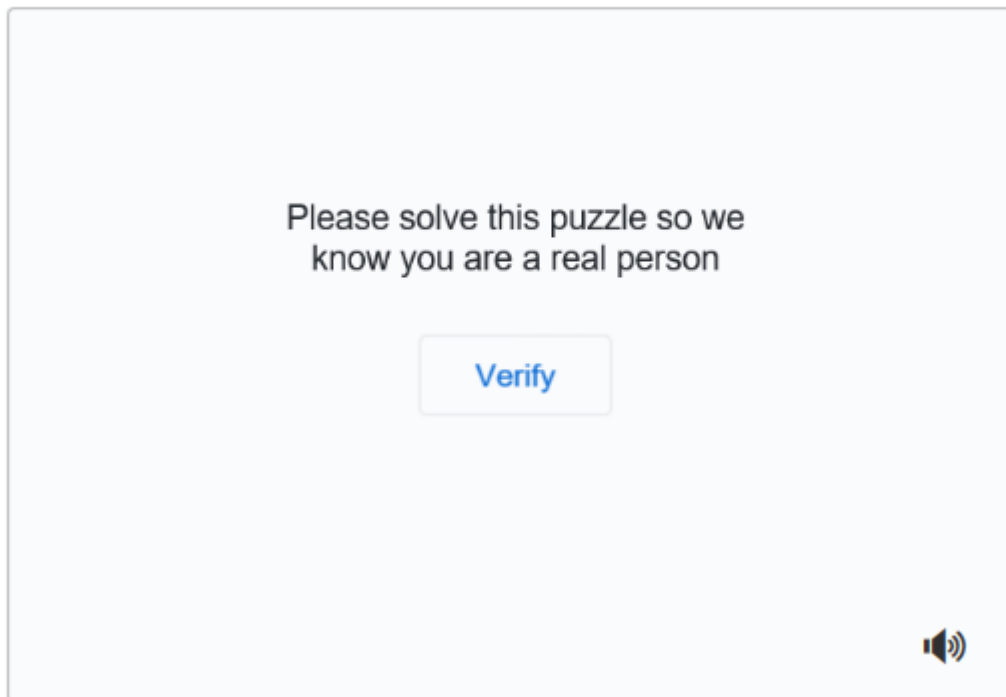
Create account

By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#). We'll occasionally send you account-related emails.

and click **Create account**.

Step 3: Click **Verify** to verify the account and click **Done**

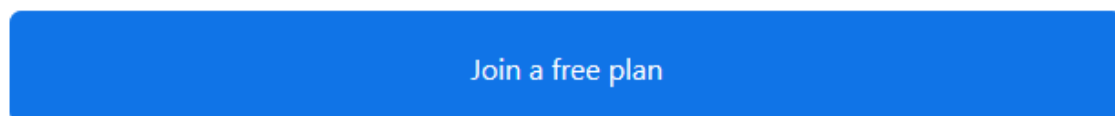
Verify your account



Step 4: After verification, click [Join a Free Plan](#)

Email preferences










☒ Send me occasional product updates, announcements, and offers.



Step 5: Select the details as shown below and click [Complete Setup](#)

What do you plan to use GitHub for?

(Select up to 3)

 Learn to code	 Learn Git and GitHub	 Host a project (repository)
 Create a website with GitHub Pages	 Collaborating with my team	 Find and contribute to open source
 School work and student projects	 Use the GitHub API	 Other

I am interested in:

We'll connect you with communities and projects that fit your interests.

For example: `zeplin` `elm` `apn`

Complete setup

Step 6: Go to your email, find the verification email from GitHub, and click on the link/button in that email to verify your email.

NOTE: If you do not receive verification email, click [Resend verification email](#).



Please verify your email address

Before you can contribute on GitHub, we need you to verify your email address.

An email containing verification instructions was sent to **Your email address**

[Resend verification email](#)[Change your email settings](#)

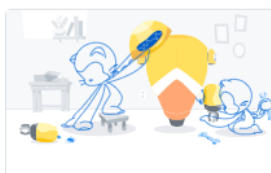
Email is verified

Your email was verified.




What do you want to do first?

Every developer needs to configure their environment, so let's get your GitHub experience optimized for you.



Start a new project
Start a new repository or bring over an existing repository to keep contributing to it.

[Create a repository](#)



Collaborate with your team
Improve the way your team works together and get access to more features with an organization.

[Create an organization](#)



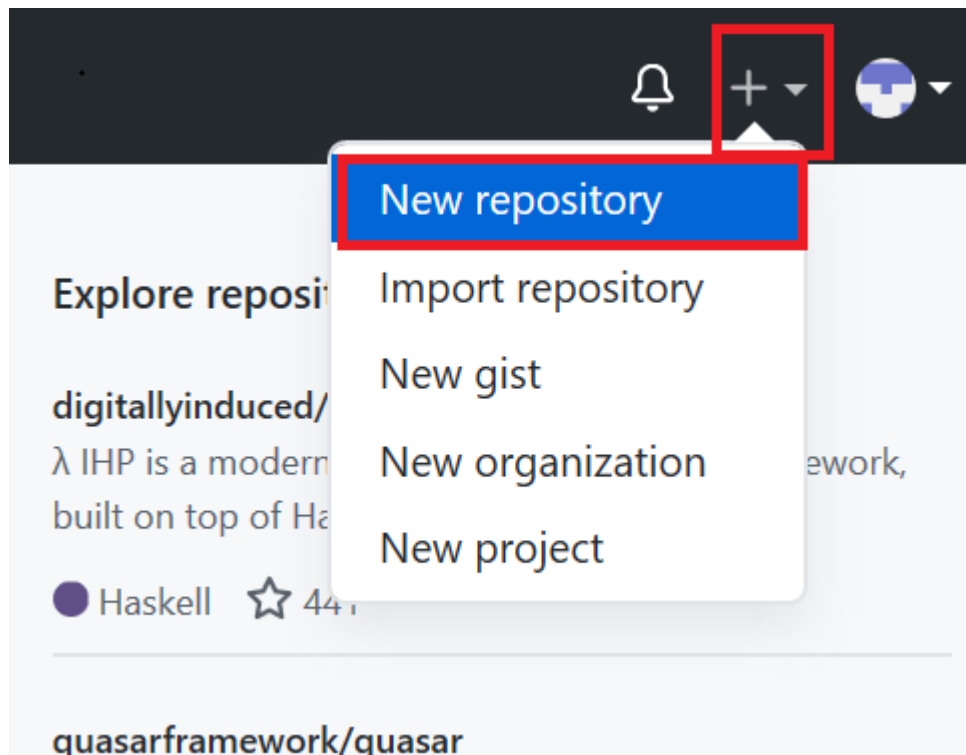
Learn how to use GitHub
Get started with an "Introduction to GitHub" course in our Learning Lab.

[Start Learning](#)

[Skip this for now >](#)

Exercise 2: Adding a Project / Repo

Step 1: Click on the + symbol and click **New repository**.




Step 2: Provide a repository a name and initialize with the empty `README.md` file.

Create a new repository


A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *

Repository name *


 Malika-s ▾

 /


testrepo 

Great repository names are short and memorable. Need inspiration? How about [urban-octo-waffle?](#)

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.

Skip this step if you're importing an existing repository.


☒ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer.

Add .gitignore: None ▾

 |

Add a license: None ▾



Create repository

and click [Create repository](#).

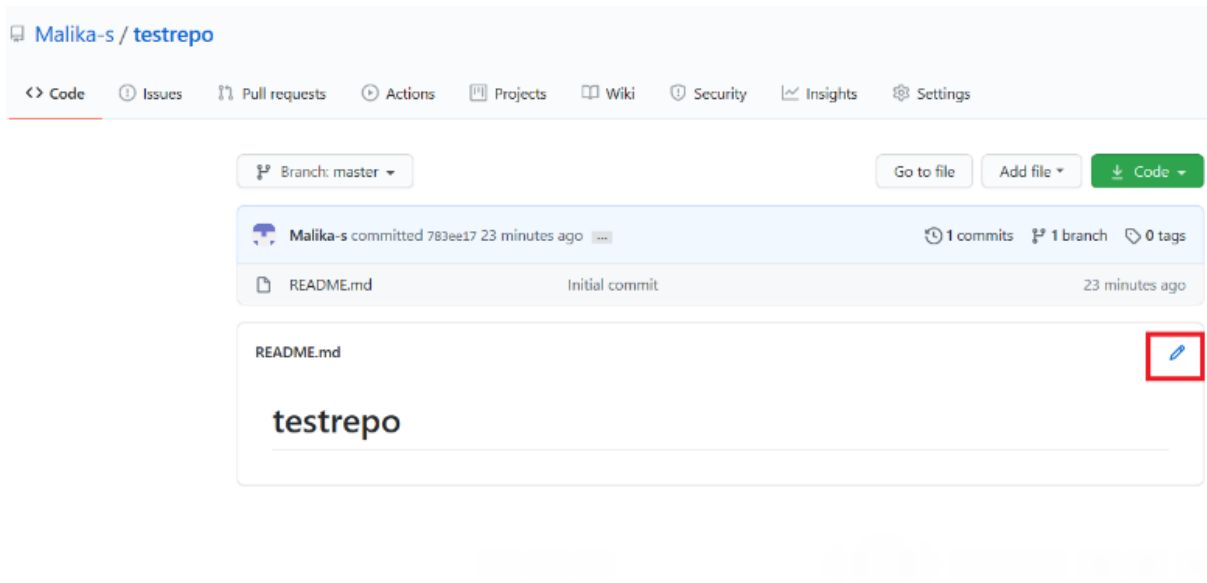
Now, you will be redirected to the repository you have created.

Let's start editing the repository.

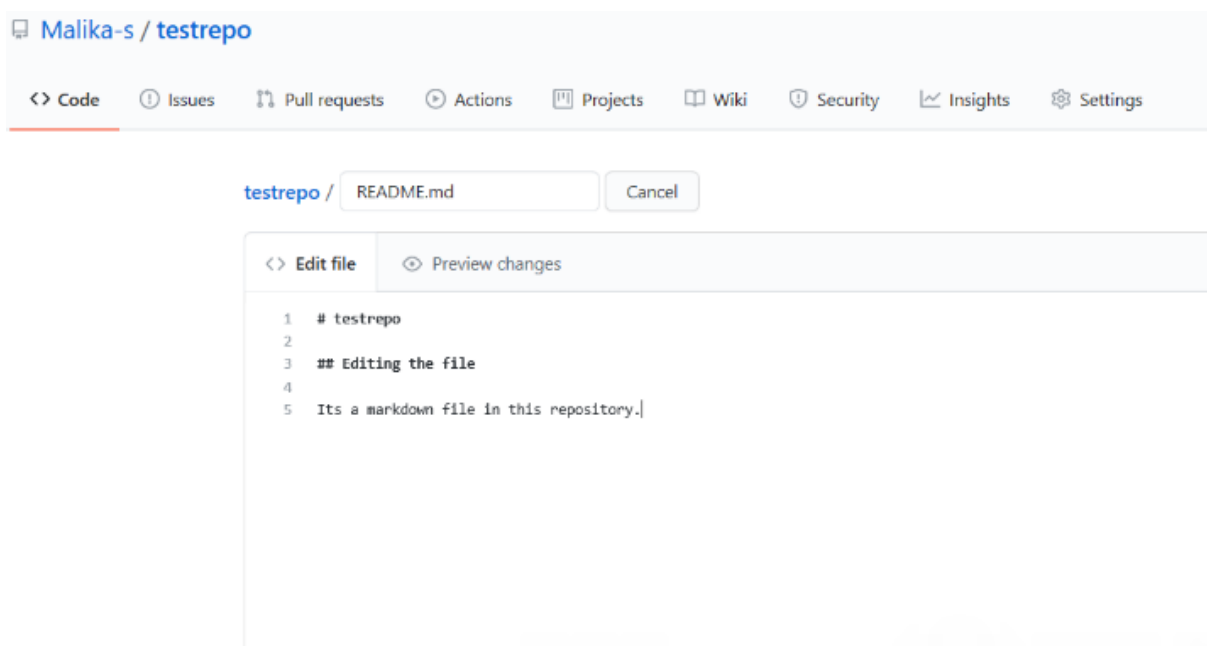
Exercise 3: Create / edit a file

Exercise 3a: Edit a file

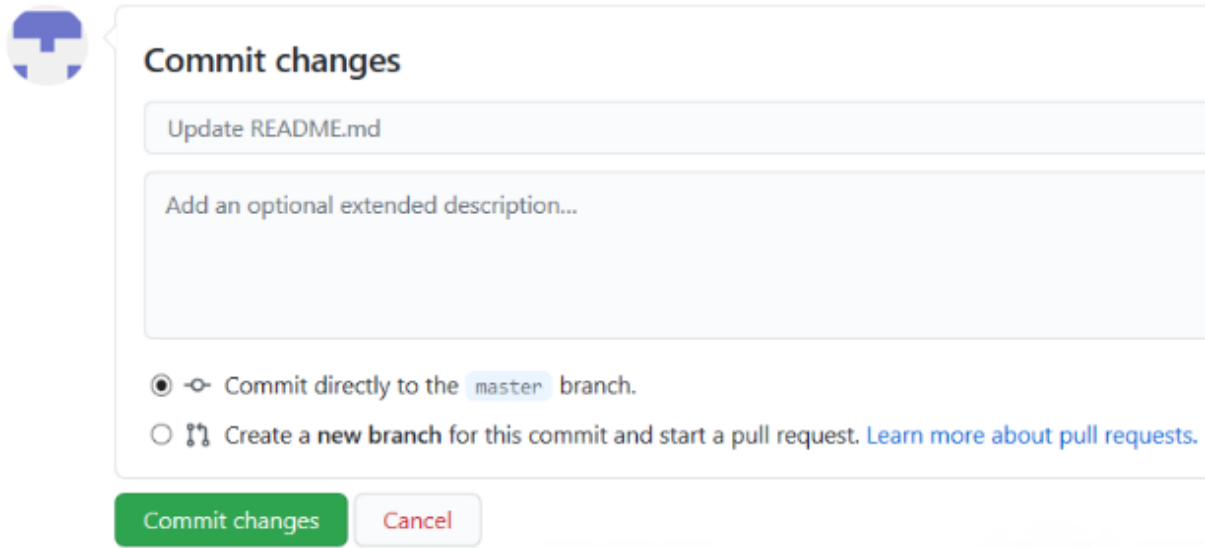
Step 1: Once the repository is created, the root folder of your repository is listed by default and it has just one file [ReadMe.md](#). Click on the pencil icon to edit the file.



Step 2: Add text to file.



Step 3: Scroll down the page after adding the text and click **Commit Changes**.



The image shows a GitHub 'Commit changes' dialog box. At the top left is a GitHub logo. The title 'Commit changes' is in bold. Below it is a text input field containing 'Update README.md'. Underneath is a larger text area with the placeholder 'Add an optional extended description...'. Below the text area are two radio button options: the first is selected and says 'Commit directly to the master branch.', the second is unselected and says 'Create a new branch for this commit and start a pull request. Learn more about pull requests.' At the bottom are two buttons: a green 'Commit changes' button and a grey 'Cancel' button.

Commit changes

Update README.md

Add an optional extended description...

☒ Commit directly to the `master` branch.

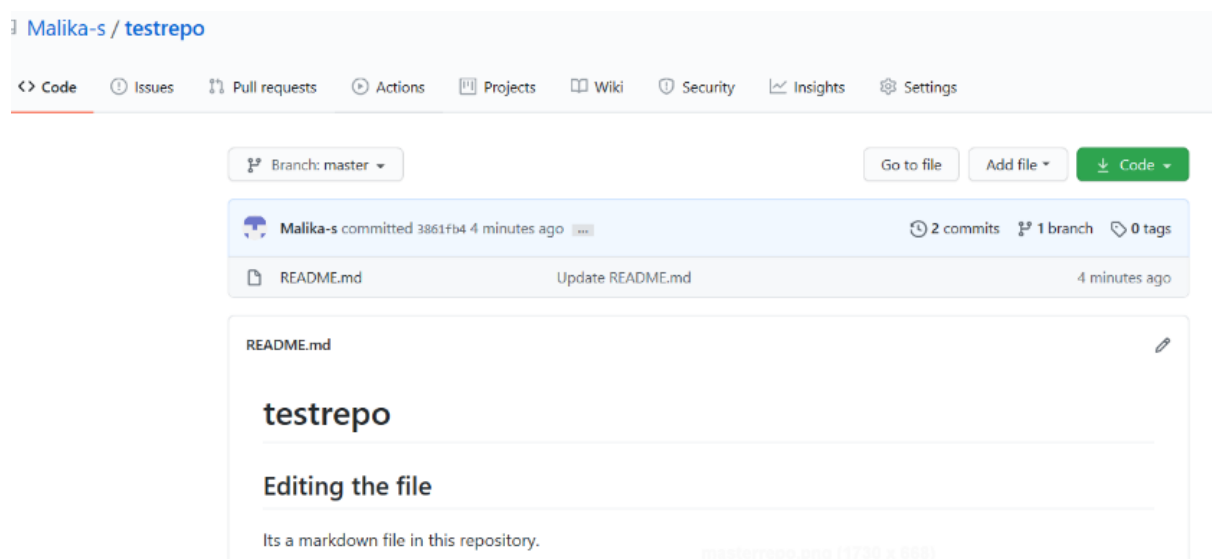
☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes Cancel

Now, check your file is edited with the new text.

Exercise 3b: Create a new file

Step 1: Click on the repository name to go back to the master branch like in this testrepo.



The image shows the GitHub repository page for 'Malika-s / testrepo'. The repository name is at the top left. Below it is a navigation bar with links: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation bar is a section for the current branch, 'Branch: master', with buttons for 'Go to file', 'Add file', and 'Code'. Below this is a commit history section showing a commit by 'Malika-s' 4 minutes ago. Below the commit history is a file list showing 'README.md' with the action 'Update README.md' 4 minutes ago. Below the file list is the content of the 'README.md' file, which contains the text 'testrepo' and 'Editing the file'. At the bottom of the file content is a note: 'Its a markdown file in this repository.'

Malika-s / testrepo

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

Branch: master Go to file Add file Code

Malika-s committed 3861fb4 4 minutes ago 2 commits 1 branch 0 tags

README.md Update README.md 4 minutes ago

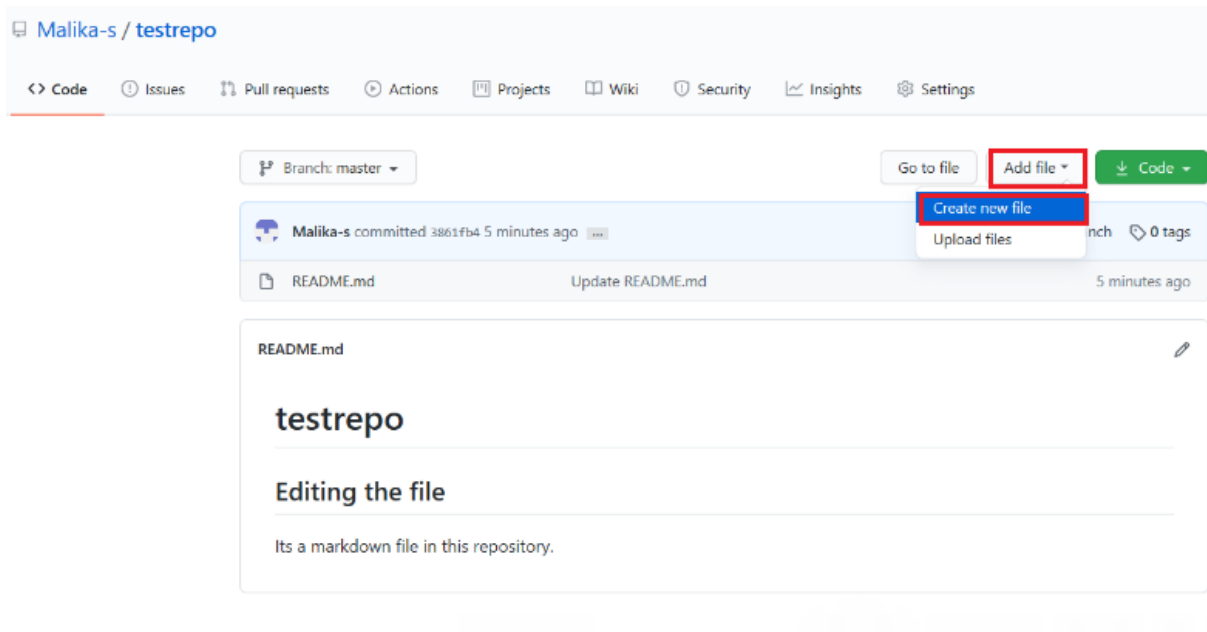
README.md

testrepo

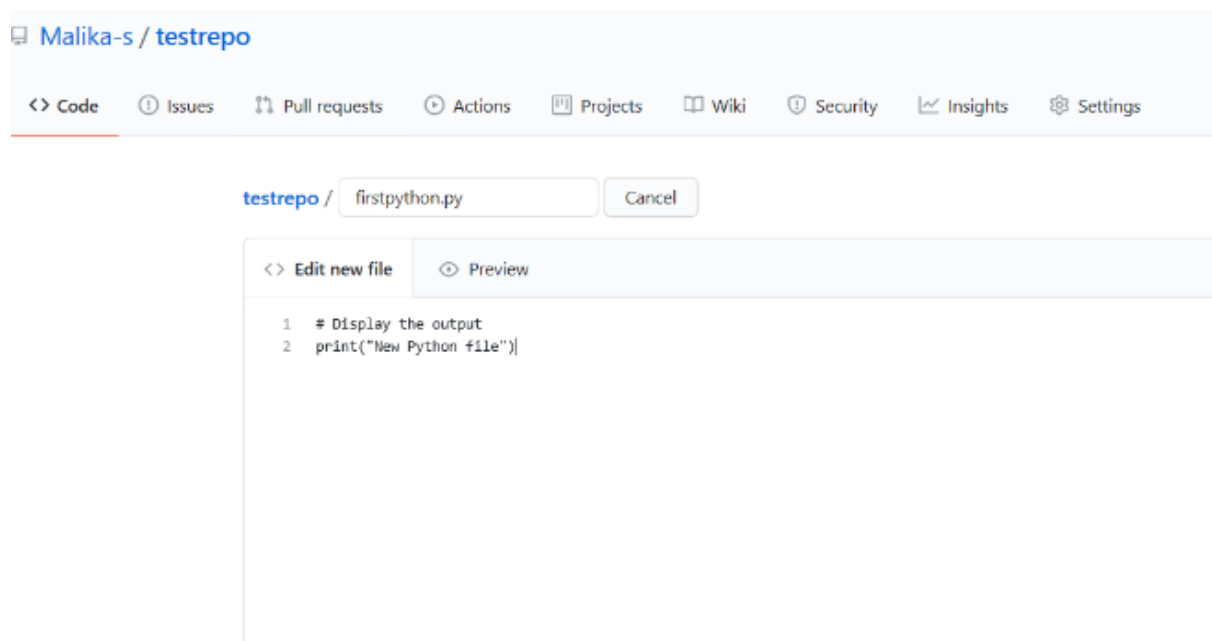
Editing the file

Its a markdown file in this repository.

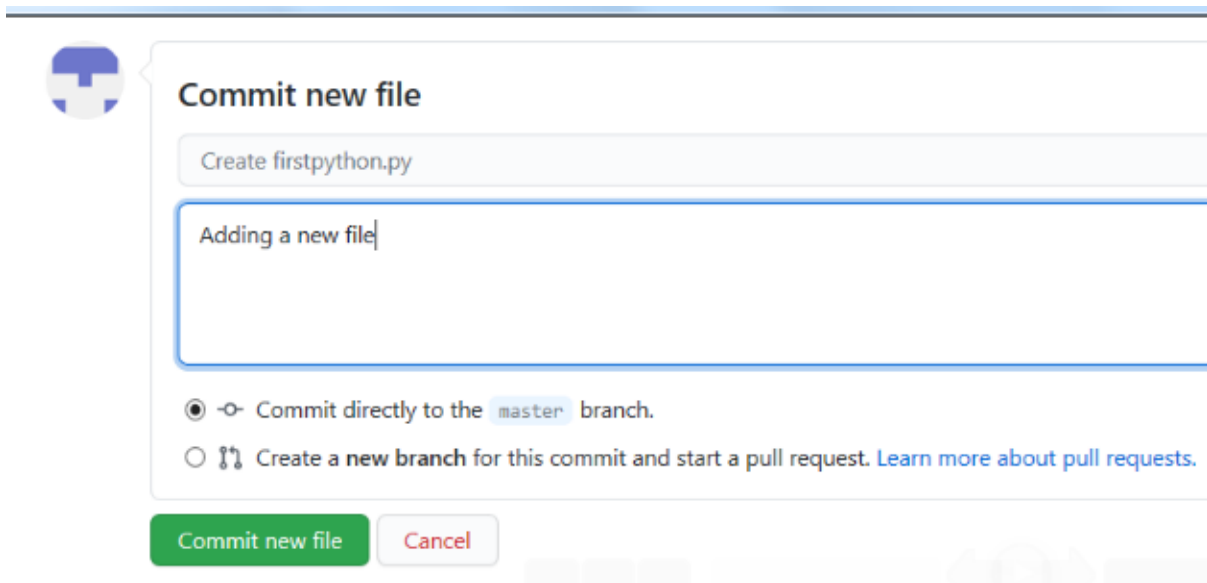
Step 2: Click **Add file** and select **Create New file** to create a file in the repository.



Step 3: Provide the file name and the extension of the file. For example, firstpython.py and add the lines.



Step 4: Scroll down the page after adding the text. Add description of the file (optional) and click Commit new file.



Commit new file

Create firstpython.py

Adding a new file

☒ Commit directly to the `master` branch.

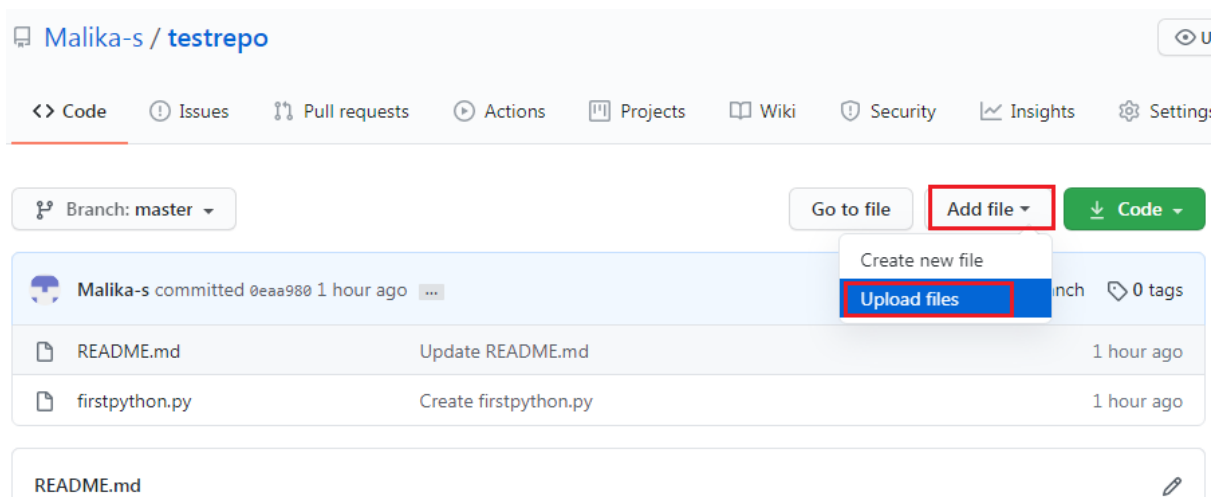
☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit new file Cancel

Step 5: Your file is now added to your repository and the repository listing shows when the file was added/changed.

Exercise 4: Upload a file & Commit

Step 1: Click **Add file** and select **Upload files** to upload a file (Upload any .txt,.ipynb, .png file) in the repository from the local computer.



Malika-s / testrepo

<> Code ⓘ Issues 🔗 Pull requests ▶ Actions 📁 Projects 📖 Wiki ⚠ Security 📈 Insights ⚙ Settings

Branch: master

Go to file Add file Code

Create new file

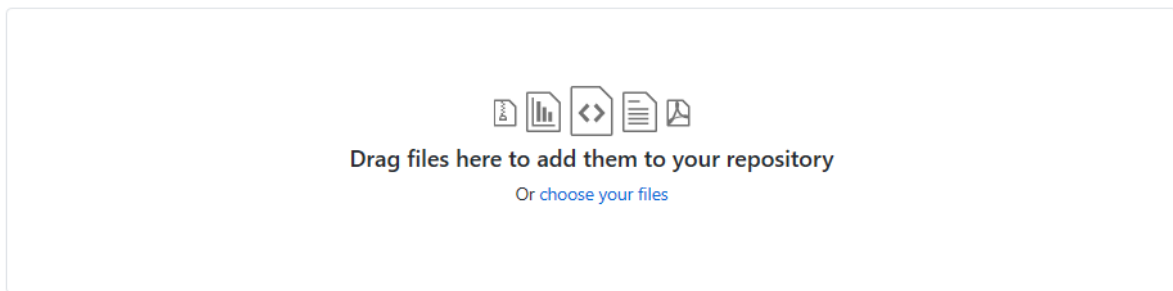
Upload files

Malika-s	committed 0eaa980	1 hour ago	...
README.md	Update README.md	1 hour ago	
firstpython.py	Create firstpython.py	1 hour ago	

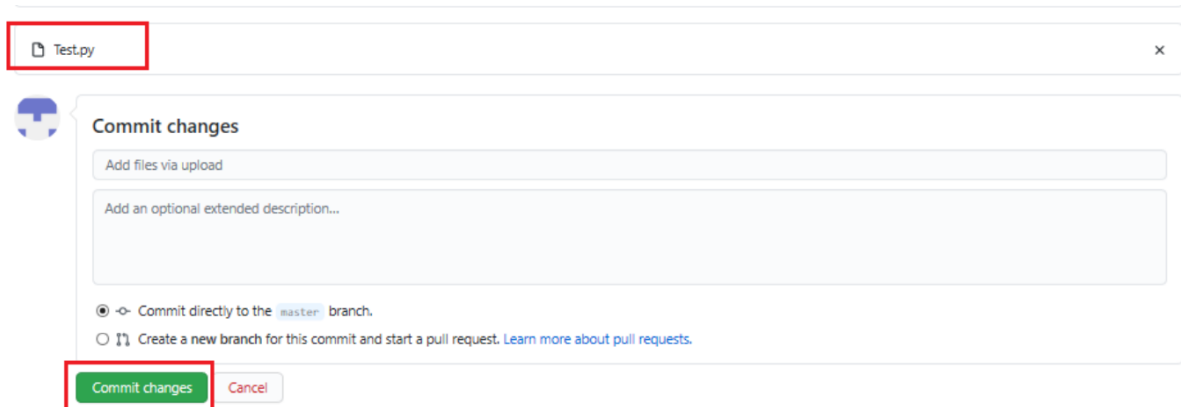
README.md

Step 2: Click on choose your files and choose any files from your computer.

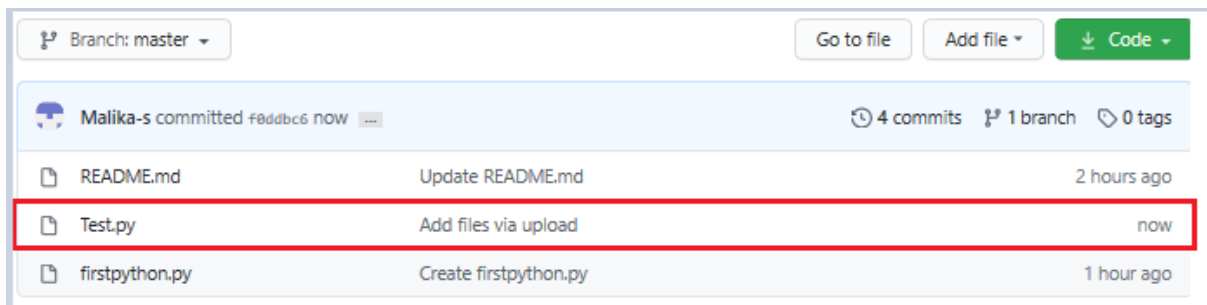
testrepo /



Step 3: Once the file finishes uploading, click on **Commit Changes**



Step 4: Now, your file is uploaded in the repository.



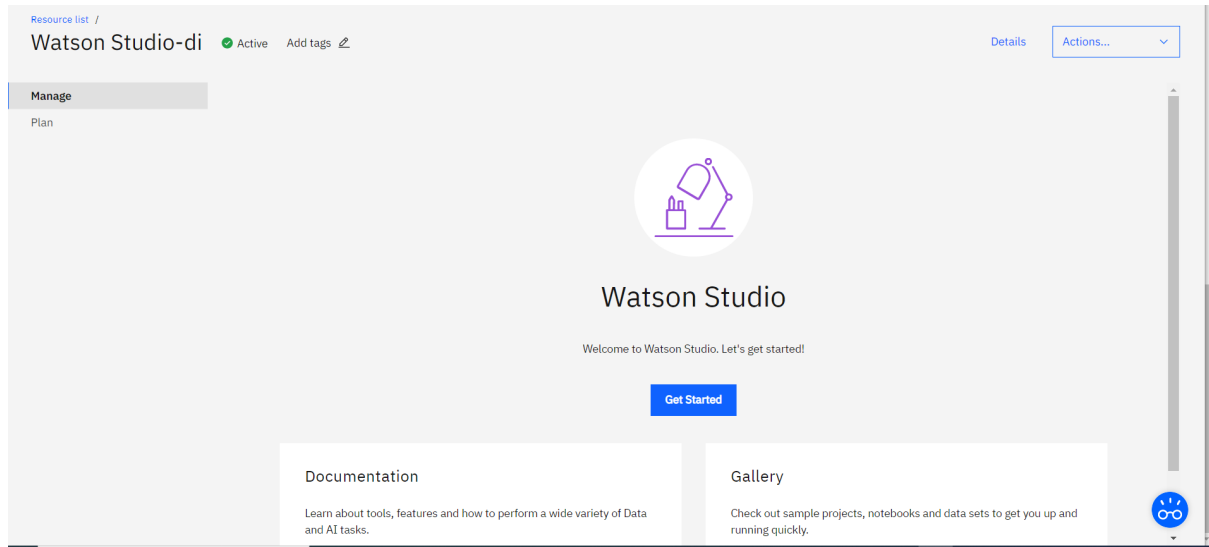
Summary

In this document, you have learned how to create a new repository, adding a new file, editing a file, and uploading a file in a repository and commit the changes.

Hands-on Lab: Publishing notebooks from Watson to Github

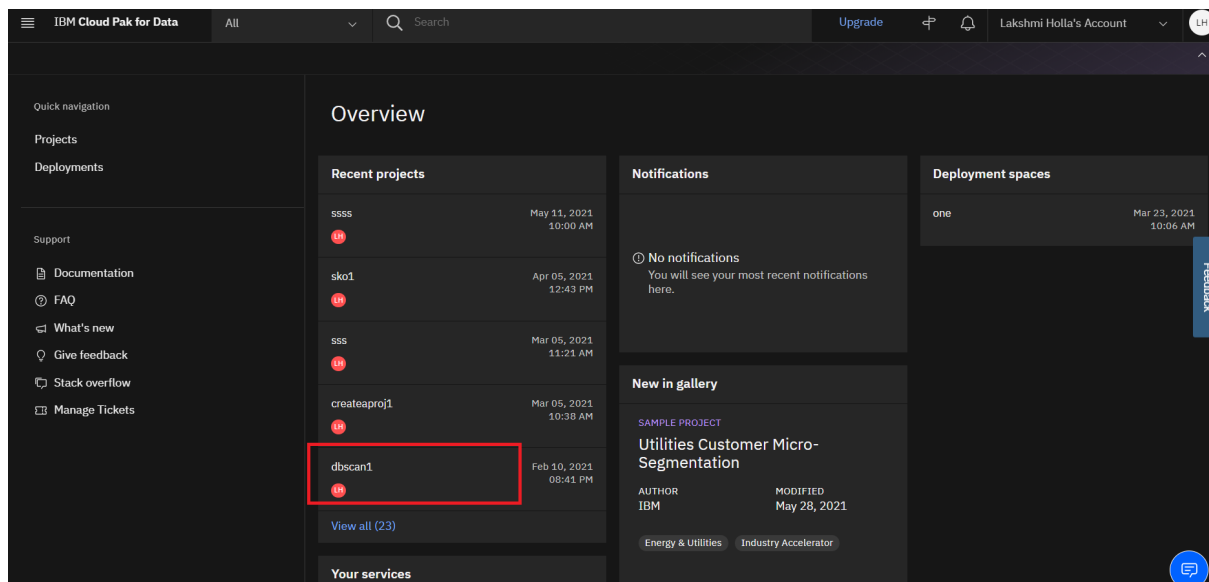
Effort: 20 mins

Step 1: Click on the Get Started Link of Watson Studio to navigate to the page where projects are listed.



Step 2: Next click on the project which you want to integrate with github. This is the project containing your notebooks to be published on Github.

Note: This is just an example of a project. The project names will differ according to what you have created.



Step 3: Next click on the Settings Tab and scroll down to check the Integrations section.

Projects / dbscan1

Overview Assets Environments Jobs Access Control Settings

What assets are you looking for?

▼ Data assets

0 assets selected.

<input type="checkbox"/>	Name	Type	Created by	Last modified
<input type="checkbox"/>	XL5 Canada.xlsx	Data Asset	Lakshmi Holla	Feb 15, 2021, 03:55 PM

▼ Notebooks

New Notebook +

Name	Shared	Scheduled	Status	Language	Last editor	Last modified	
<input type="checkbox"/> Segmenting-Clustering Neighborhoods-Washington DC				Python 3.7	Lakshmi Holla	Feb 24, 2021	
<input type="checkbox"/> Clus-DBSCN-weather				Python 3.7	Lakshmi Holla	Jun 02, 2021	
<input type="checkbox"/> createnew				Python 3.7	Lakshmi Holla	Feb 15, 2021	
<input type="checkbox"/> Peer graded ETL				Python 3.7	Lakshmi Holla	Apr 14, 2021	

Integrations

Github repository	Not connected
Figure Eight <small>BETA</small>	Setup required
DefinedCrowd <small>BETA</small>	Setup required

Step 4: Click on the GitHub repository on the integrations section. When you click on the textbox under the Repository url you will get the error as shown below:

Integrations

Github repository	Not connected
-------------------	---------------

A GitHub personal access token has not been setup. Configure settings.

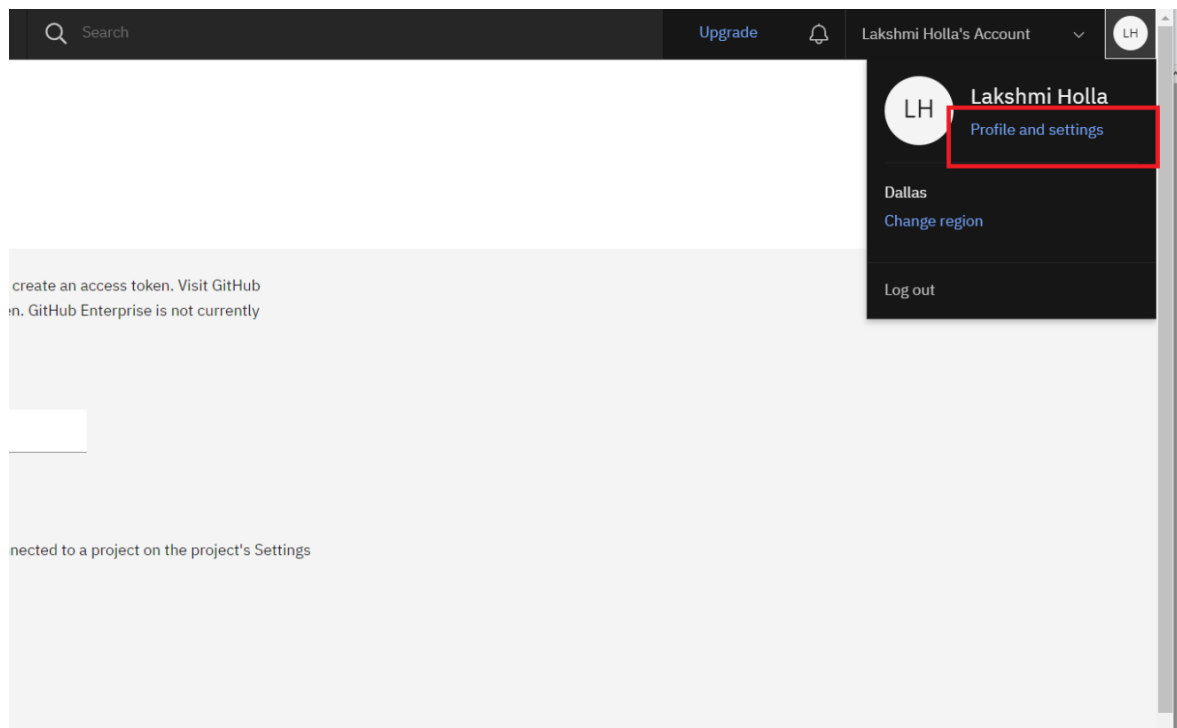
Repository URL

This field is required.

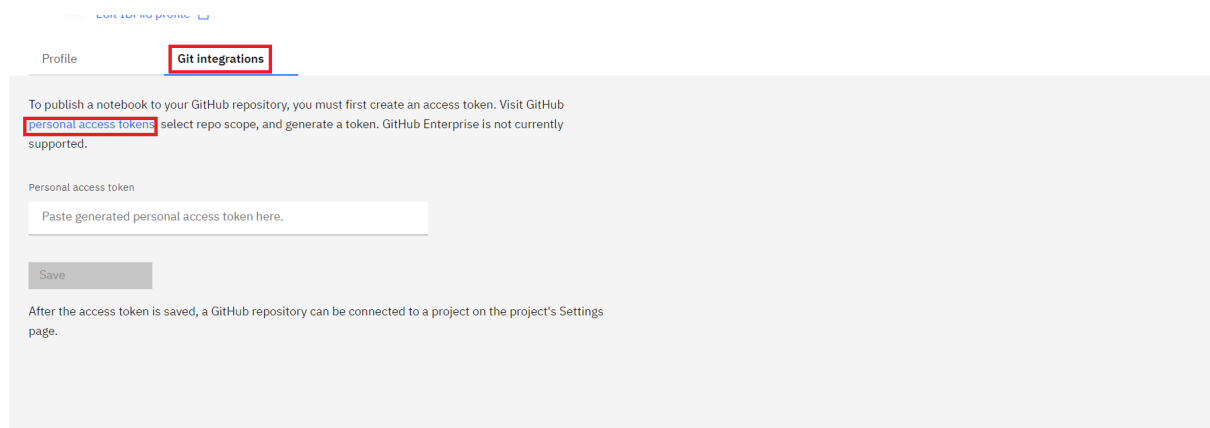
Update

To create and configure the personal access token follow the subsequent steps.

Step 5: Next click on Profiles and Settings right below your initials on the top right screen.



Step 6: Click on the gitIntegrations tab.



Step 7: Click on personal access token. This will prompt you to enter your github password and login to github.

Later create a token by specifying the name in the Note and choose repo scope.

- GitHub Apps
- OAuth Apps
- Personal access tokens

New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

IBM Watson Studio

What's this token for?


Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry
<input type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input type="checkbox"/> read:org	Read org and team membership, read org projects

Scroll down and click 'Generate Token'.

Step 8: Copy the token generated.

8wzewDE0pifZLrPx6sUgQeYk4BRoCw 

Step 9: Paste it under the Personal Access token in the Git Integrations tab of Watson and click on save.

[Edit IBMid profile](#)

Profile **Git integrations**

To publish a notebook to your GitHub repository, you must first create an access token. Visit [GitHub personal access tokens](#), select repo scope, and generate a token. GitHub Enterprise is not currently supported.

Personal access token

Save

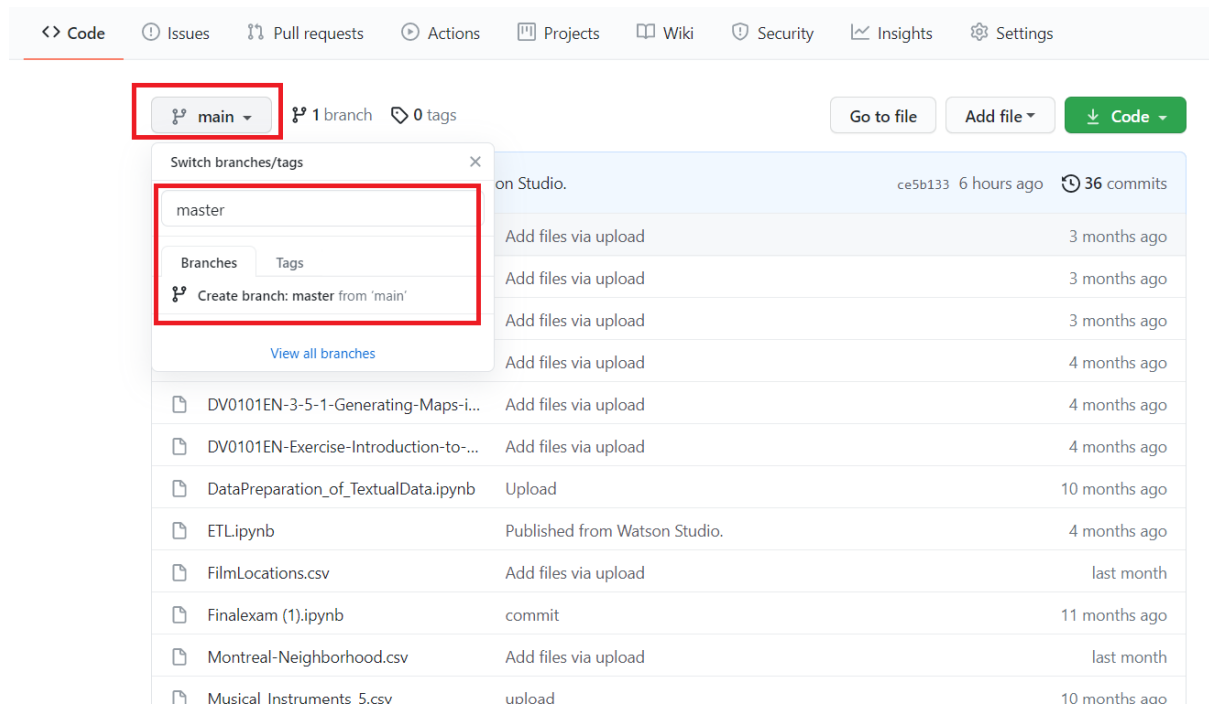
After the access token is saved, a GitHub repository can be connected to a project on the project's Settings page.

Step 10: Go back to your github repository by clicking this [link](#).
Select your repository on the left pane.

Create a branch called master under your repository.

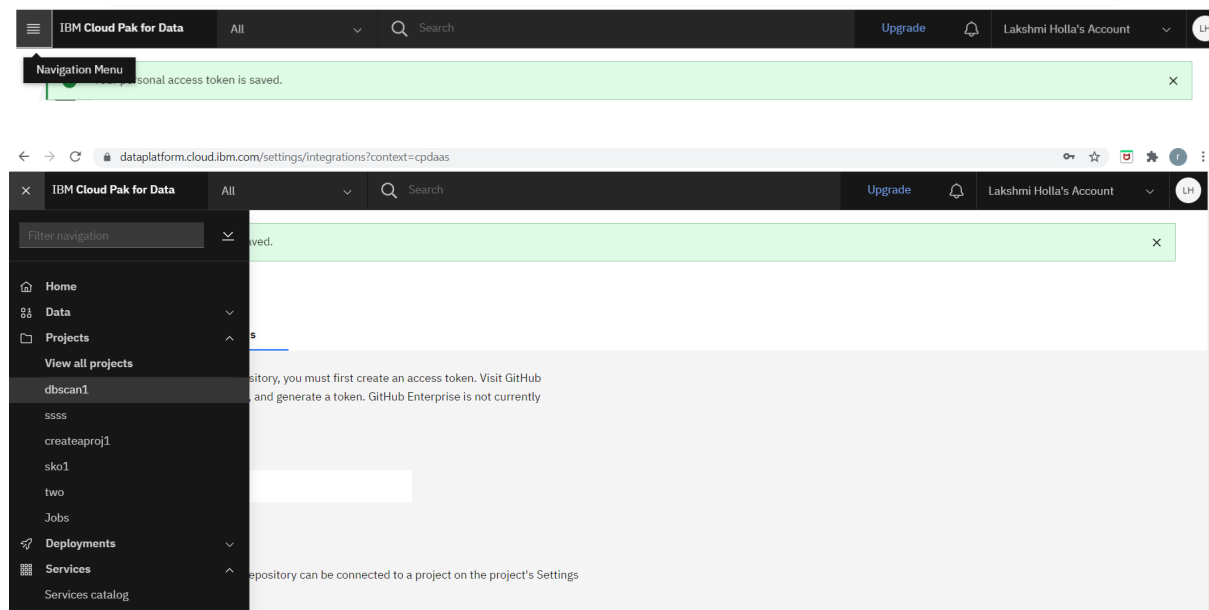
Step 11: Click the button on the left side of your screen that shows the current branch as "main" and has a dropdown arrow.

Step 12: Create a new branch called "master"



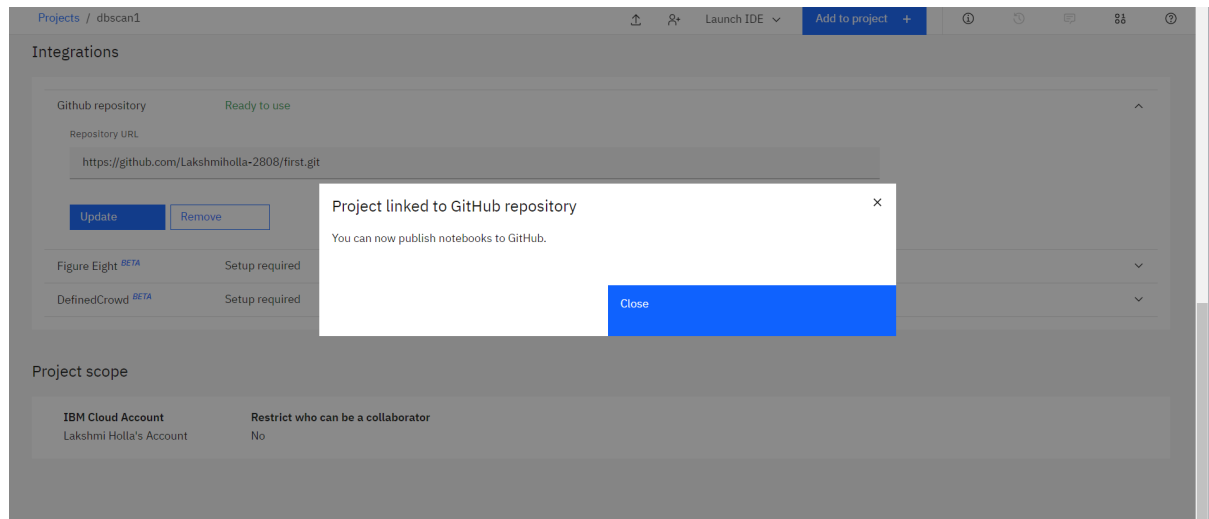
This is necessary because when you connect a repository to a project in Watson Studio and try to publish a notebook to GitHub, it automatically pushes your notebook to the master branch. If you only have a main branch in your repository and not one titled "master," Watson will not have anywhere to push your notebook.

Step 13: Once done with this step navigate to your project on watson by clicking the Navigation menu.



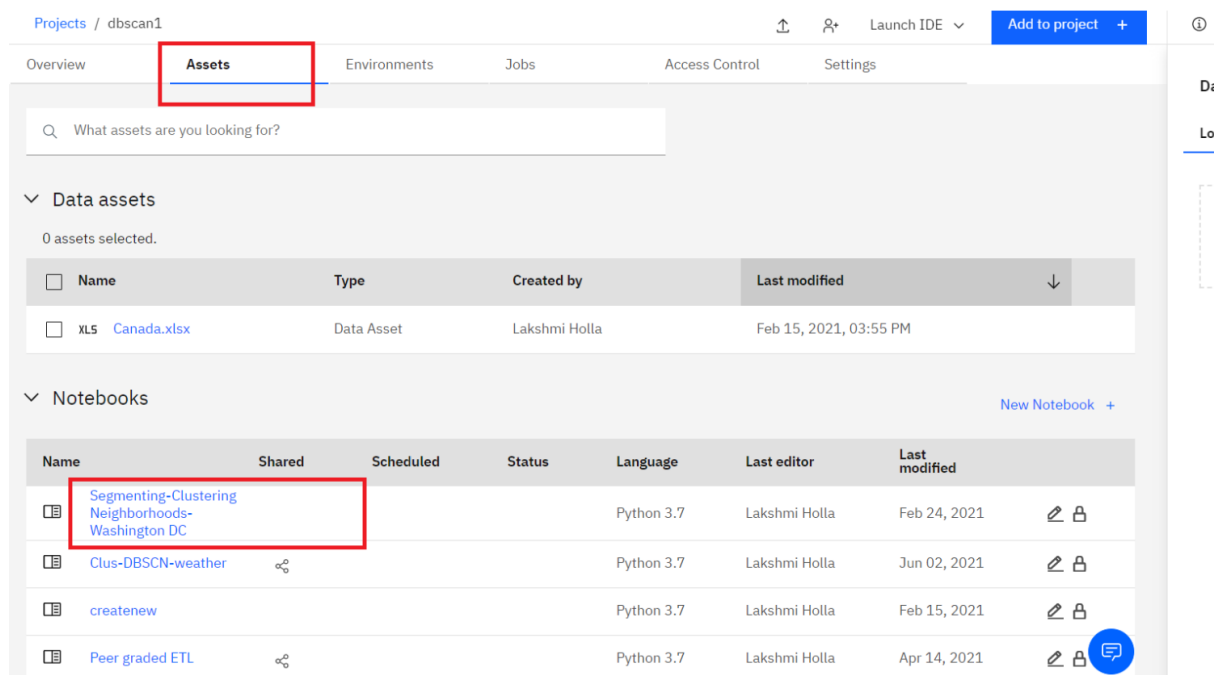
Step 14: Go to the settings tab and provide the github link of your repository.

If you have not yet created the repository follow the steps given in the link [Github Repository Creation](#) for creating a repository.



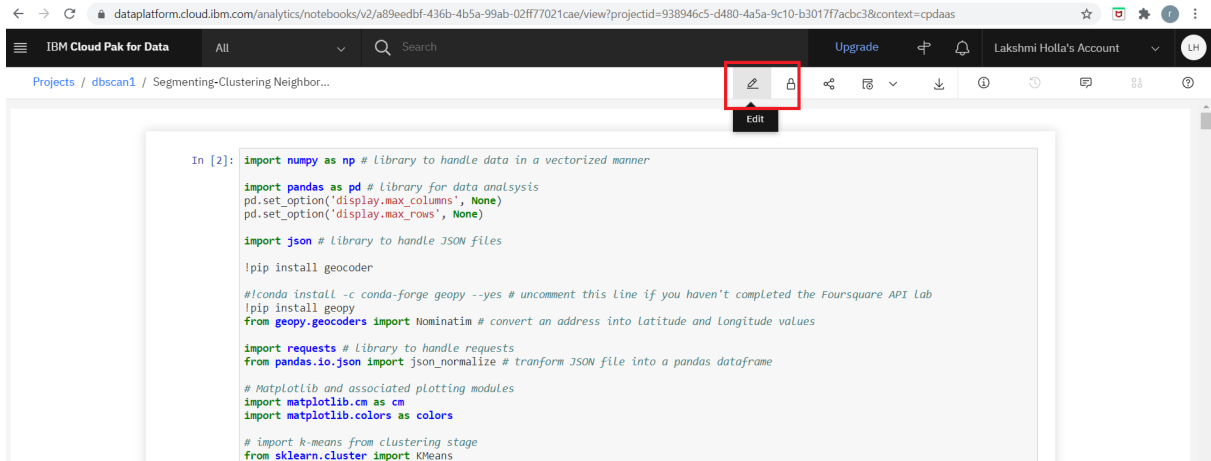
If update is not enabled, click outside text box which will enable it and then click 'Update'.

Step 15: Navigate to the Assets tab of your project and then click on the notebook which you want to publish.

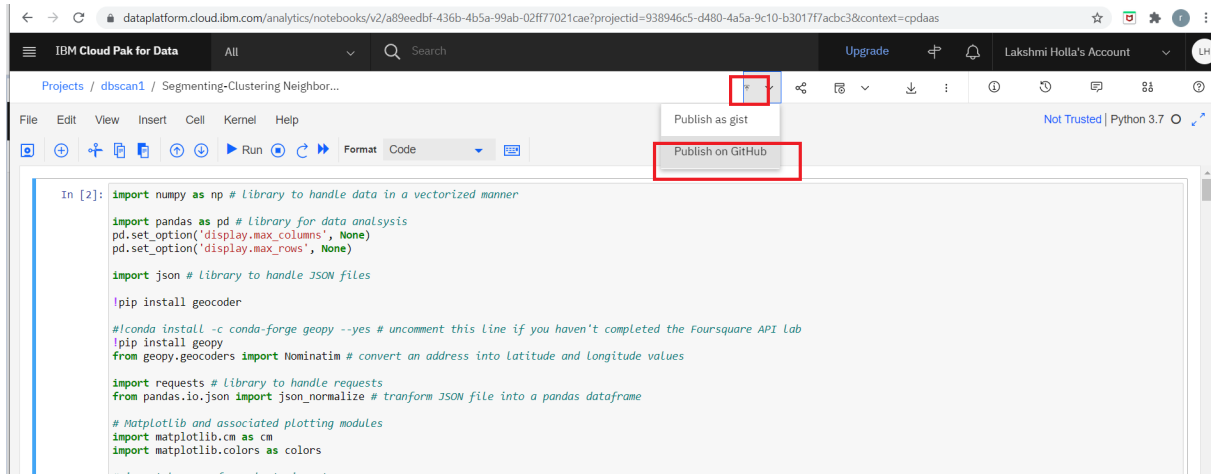


This will open your notebook.

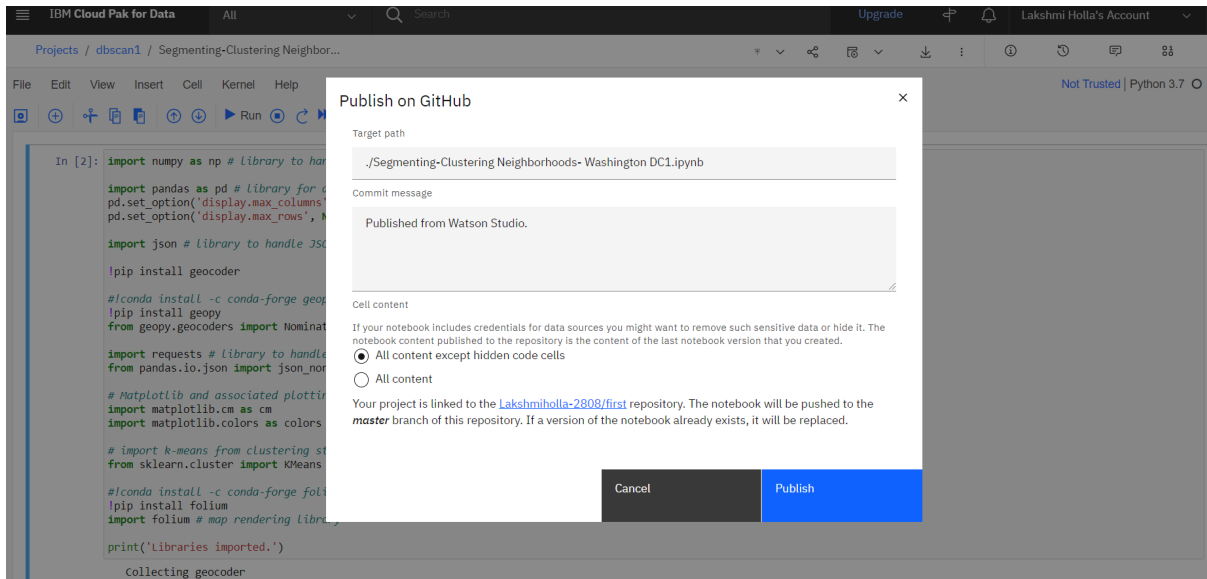
Step 16: Click on the edit option to edit your notebook.



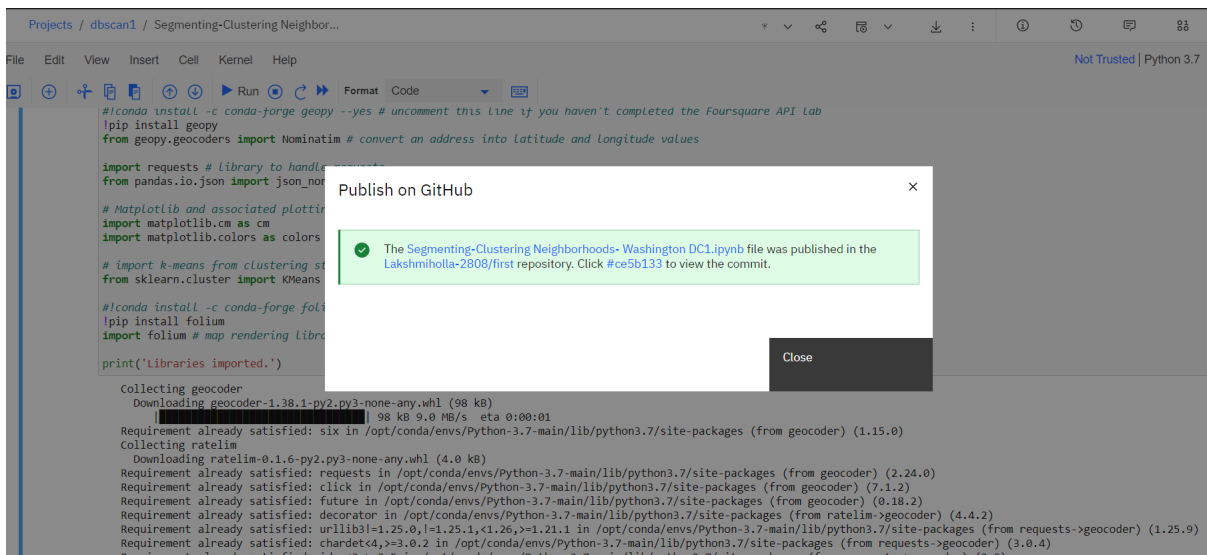
Step 17: Select Publish on github.



Step 18: Next click on Publish to publish your notebook



Step 19: This will publish your notebook to github.



Author(s)