

CMSC388T

# Using Git For Collaboration

# Contents

1

## Remotes and Branches

Using the Git Command-line to work with branches on your local repository

2

## Using The Github UI

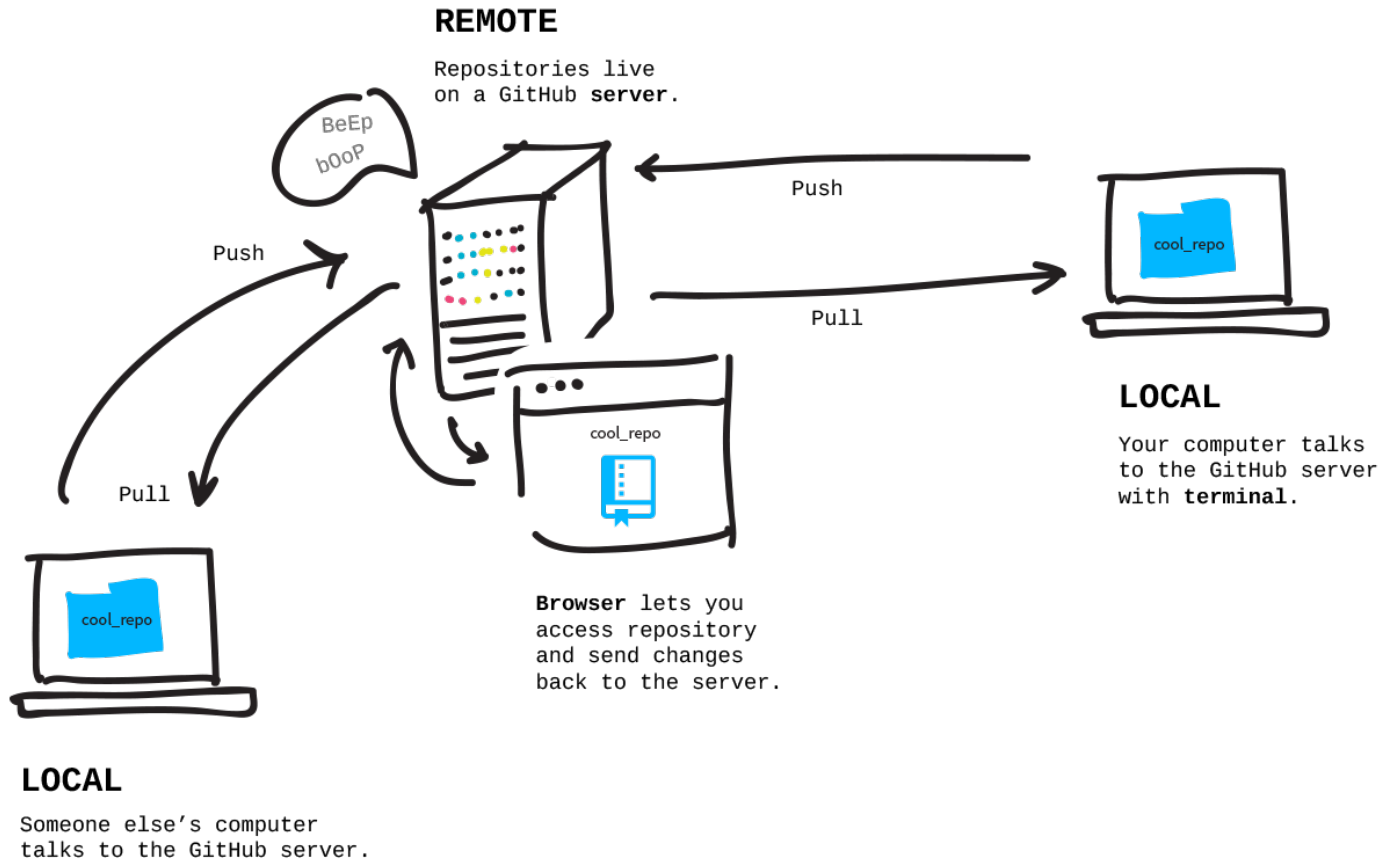
Working with collaborators using the Github UI

3

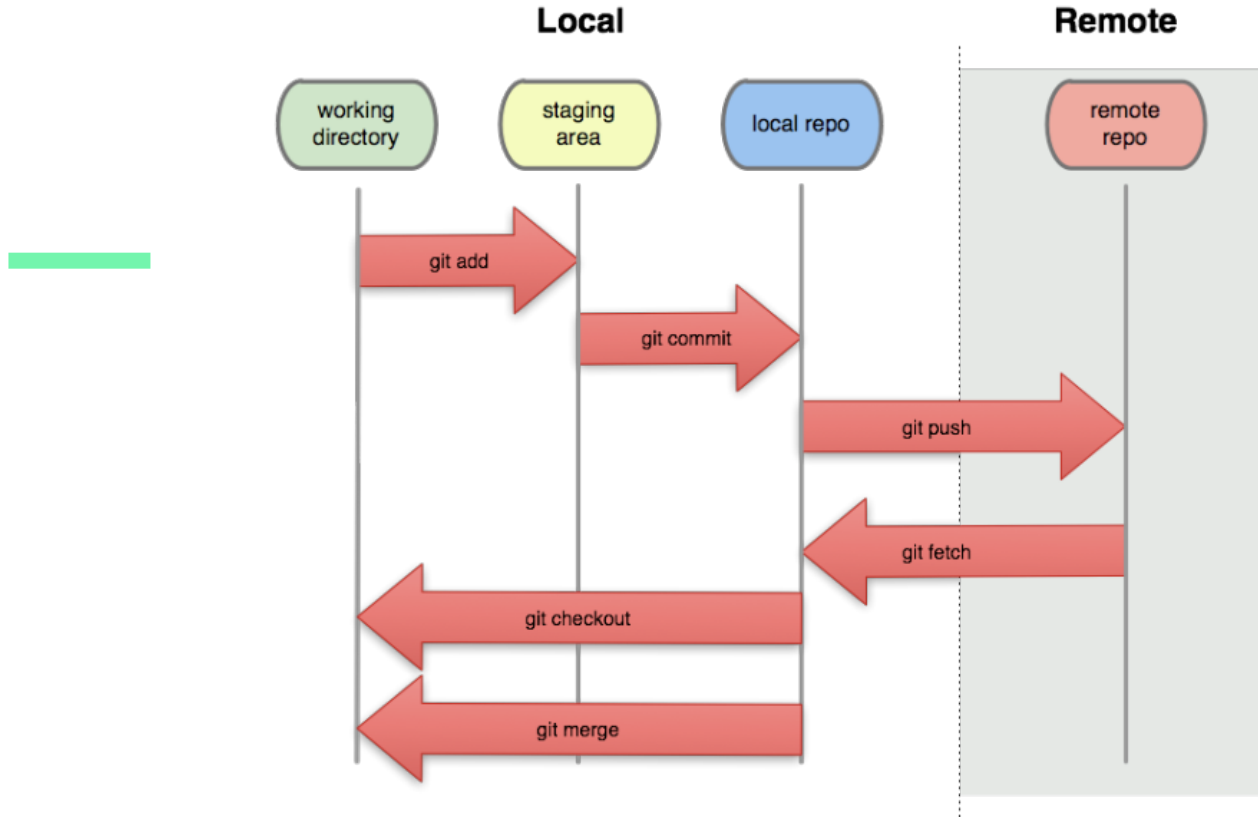
## CI/CD

Understanding Continuous Integration and Continuous Deployment/Delivery

# Remotes



# Git Local vs Remote





# Some Basic Terminology

## Repository (Repo)

A folder for your project that contains all of your project's files along with their revision history.

## Local repository (Workspace)

Your local copy of a repository on your computer.

## origin

The remote, original copy of a repository.

## Git Command Line – 2

### **git status**

Gives you a current overview of your repository. It telling you which files have or haven't been saved and what changes are in staging.

### **git push**

Pushes changes in your locally committed repository to the remote repository.

### **git pull**

Downloads or updates the contents of a remote repository to your local device.

# SSH Keys

- ssh keys are special, unique files that allow the user to access secure data (in this case our code)
- Git implements a similar level of security and requires user to ssh with ssh keys.
- If a user does not have a key, they will not gain access to anything

## Generate your SSH keys

Enter the following command with the email associated with your GitHub account:

```
ssh-keygen
```

```
→ ~ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/sanjay/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/sanjay/.ssh/id_rsa.
Your public key has been saved in /Users/sanjay/.ssh/id_rsa.pub.
```

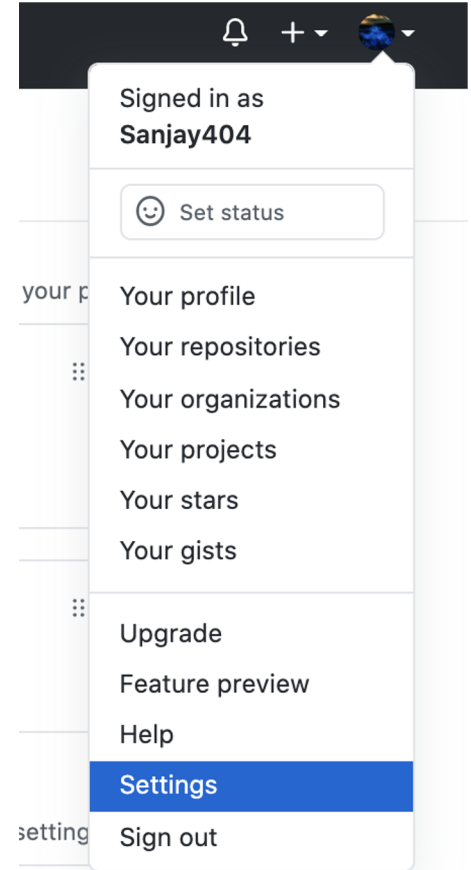
Accept the default save location by pressing enter and continue by answering the prompts and run the following command:

```
pbcopy < ~/.ssh/id_rsa.pub
```

The above command copies the ssh key to your clipboard

# SSH keys continued ...

1. Navigate and sign into GitHub in your browser
1. In the top right corner, press on your icon and then in the menu bar, select Settings



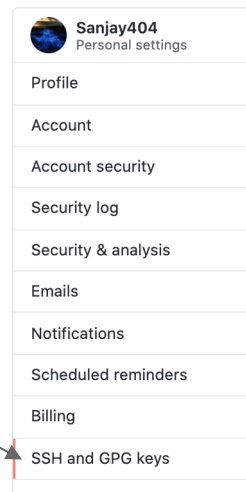
# SSH keys continued ...

3. Once in settings, navigate to SSH and GPG keys.

4. Navigate to the button that is labeled “new SSH key”

and then paste what is in your clipboard into “Key” [SSH keys](#) / Add new

5. Press “Add SSH key” when finished



Title

Key

Begins with 'ssh-rsa', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', or 'ecdsa-sha2-nistp521'

Add SSH key

# Making our repository

Use the “git init” command to make a new repository called Test\_Repo

```
→ git_examples git init Test_Repo  
Initialized empty Git repository in /Users/sanjay/Desktop/classes/git_examples/Test_Repo/.git/
```

Now simply add any files you want to the directory

```
→ Test_Repo git:(master) touch test.txt  
→ Test_Repo git:(master) × ls  
test.txt
```

## Moving test.txt to the Staging Area

run "git add test.txt" to move  
the file to the staging area

```
[→ Test_Repo git:(master) ✕ git add test.txt
```

# Committing Our Changes

---

run “git commit” to move the file from the staging area. We add the “-m” option to include a message

```
[→ Test_Repo git:(master) ✕ git commit -m "V1"  
[master (root-commit) 1a7c555] V1  
1 file changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 test.txt
```



# Checking on Our Progress

run “git status” to check what is  
in our staging area and in our  
master branch

```
[→ Test_Repo git:(master) git status  
On branch master  
nothing to commit, working tree clean
```

You can also run “git log” to see  
the all your commit history

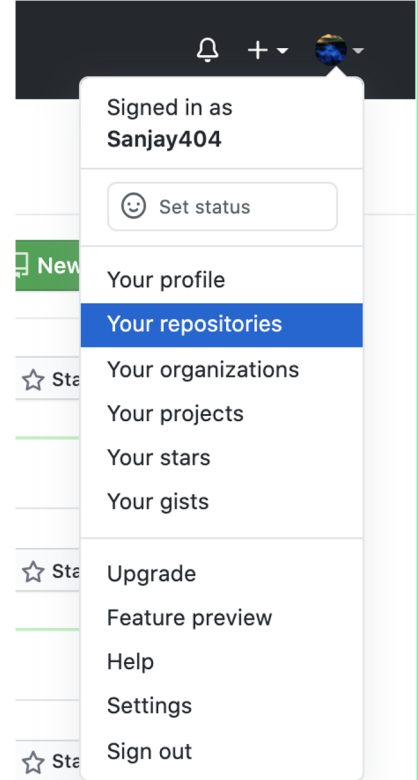
```
commit 1a7c555f1a720a00d80778861d464624dc43b315 (HEAD -> master)  
Author: Sanjay S <srikumar.sanjay@gmail.com>  
Date: Mon Nov 16 23:17:50 2020 -0500
```

V1

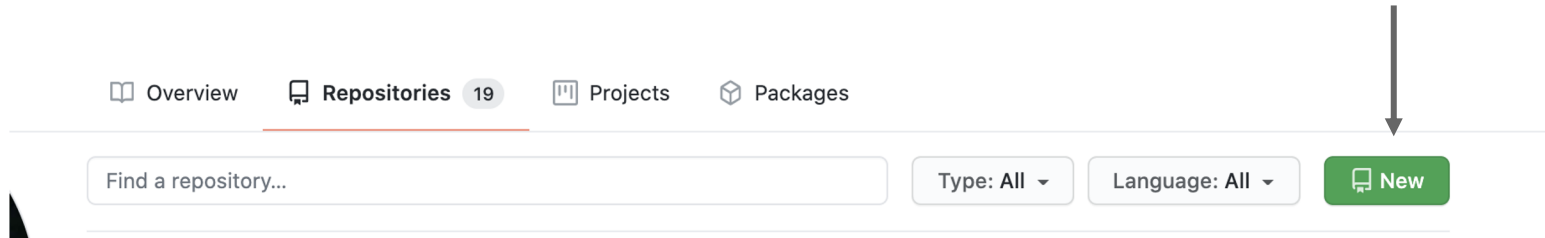
(END)

# Making our remote repository

1. Navigate and sign into GitHub in your browser
1. In the top right corner, press on your icon and then in the menu bar, select Your repositories



# Remote Repository continued ...



1. Find the above search bar, and look for the button labeled new
1. In the top right corner, press on your icon and then in the menu bar, select Your repositories

# Remote Repository continued ...

## Create a new repository

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

Owner \*

Sanjay404

Repository name \*

Create repository names are short and memorable. Need inspiration? How about [ideal-winner](#)?

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:

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

☐ Add a README file

This is where you can write a long description for your project. [Learn more](#).

☐ Add .gitignore

Choose which files not to track from a list of templates. [Learn more](#).

☐ Choose a license

A license tells others what they can and can't do with your code. [Learn more](#).

Create repository

Give your repository a name and then select  
Create Repository

# Remote Repository continued ...

GitHub will then give you a ssh url to use, along with some commands we've gone over to get you started

**Quick setup — if you've done this kind of thing before**

Set up in Desktop

 or 

HTTPS

SSH

git@github.com:Sanjay404/hi.git

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

**...or create a new repository on the command line**

```
echo "# hi" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:Sanjay404/hi.git
git push -u origin main
```

**...or push an existing repository from the command line**

```
git remote add origin git@github.com:Sanjay404/hi.git
git branch -M main
git push -u origin main
```

**...or import code from another repository**

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

# Pushing To Remote Repository

“git remote **add** (origin remote repository URL) creates a remote connection called origin from your local repository

“git push origin master” simply pushes any changes from your local master branch to the remote master branch

(don't worry about branching just yet)

```
→ Test_Repo git:(main) git remote add origin git@github.com:Sanjay404/Test_Repo.git
→ Test_Repo git:(main) git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 209 bytes | 209.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:Sanjay404/Test_Repo.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

# Viewing Repository on Github

The screenshot shows the GitHub interface for a repository named 'Test\_Repo' by user 'Sanjay404'. The top navigation bar includes a search field, links for Pull requests, Issues, Marketplace, and Explore, and a user profile icon. The repository header shows the name 'Sanjay404 / Test\_Repo', a dropdown menu for 'Unwatch' (set to 'Unwatch'), 1 star, 0 forks, and 0 watchers. Below the header is a tabbed interface with 'Code' selected, and other tabs for Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The main content area displays the 'main' branch with 1 branch and 0 tags. A commit by 'Sanjay404' is shown with the message 'V1', commit hash '1a7c555', and '11 minutes ago'. A file 'test.txt' is listed under this commit. A green 'Code' button is visible. A blue box prompts the user to 'Add a README'. On the right, the 'About' section states 'No description, website, or topics provided.' Below it, the 'Releases' section says 'No releases published' with a link to 'Create a new release'. The 'Packages' section also states 'No packages published' with a link to 'Publish your first package'. The footer contains the GitHub logo, copyright notice '© 2020 GitHub, Inc.', and links for Terms, Privacy, Security, Status, Help, Contact GitHub, Pricing, API, Training, Blog, and About.

Search or jump to...

Pull requests Issues Marketplace Explore

Sanjay404 / Test\_Repo

Unwatch 1 Star 0 Fork 0

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

main 1 branch 0 tags

Go to file Add file Code

Sanjay404 V1 1a7c555 11 minutes ago 1 commits

test.txt V1 11 minutes ago

Help people interested in this repository understand your project by adding a README. Add a README

About

No description, website, or topics provided.

Releases

No releases published  
Create a new release

Packages

No packages published  
Publish your first package

© 2020 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About



## 2. Working With Branches

---

Using the Git CLI to work with branches on your local repository



# What are Branches?

---

- A version of the repository that you can edit that has been separated from the master branch
- Allows you to create edits on a new feature you are developing without changing the main code
- These branches can be merged later with the master branch

# Branches

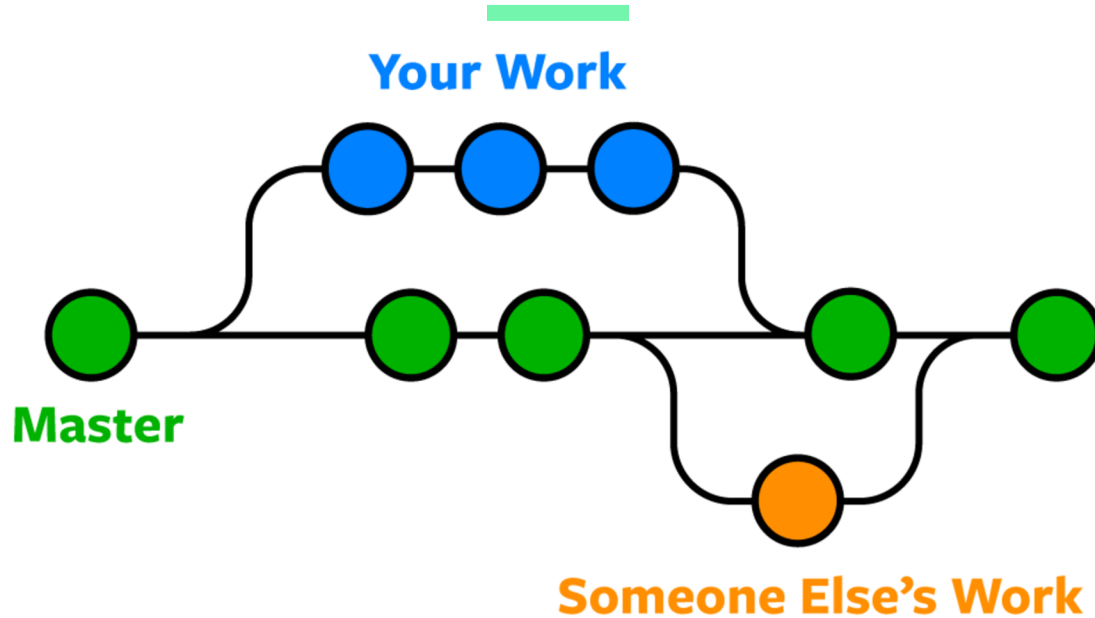


Image Source: [nobledesktop.com](http://nobledesktop.com)

# Checking Your Current Branch

---

To check the branch you are currently on, you can run the **git status** command within your repository:

```
[TestRepo]$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
[TestRepo]$
```

# Getting a List of Branches

---

To get a list of branches that currently exist, you can use the **git branch** command within your repository:

```
[TestRepo $]git branch
* master
[TestRepo $]
```

# Creating a New Branch

---

To create a new branch you can use the **git branch branch\_name** command within your repository:

```
TestRepo $git branch test_branch
TestRepo $git branch
* master
  test_branch
TestRepo $
```

# Switching Branches

---

To create a new branch you can use the **git checkout branch\_name** command within your repository:

```
[TestRepo $git checkout test_branch
Switched to branch 'test_branch'
[TestRepo $git branch
  master
  * test_branch
[TestRepo $
```

# Merge Branch

---

To merge a branch to an existing branch, you need to first checkout the branch you want to merge to with **git checkout dest\_branch** and merge with **git merge --no-ff src\_branch** command within your repository:

```
[TestRepo $]git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
[TestRepo $]git merge --no-ff test_branch
Merge made by the 'recursive' strategy.
 README.md | 2 ++
1 file changed, 2 insertions(+)
[TestRepo $]
```

# Push A Branch

To push a branch to a remote repository, you can push your new branch with **git push -u origin your\_branch** command within your repository.

**Caution:** you will need to pull changes before pushing new changes.

```
TestRepo $git push -u origin test_branch
Enumerating objects: 44, done.
Counting objects: 100% (44/44), done.
Delta compression using up to 8 threads
Compressing objects: 100% (38/38), done.
Writing objects: 100% (44/44), 5.28 KiB | 1.76 MiB/s, done.
Total 44 (delta 10), reused 7 (delta 1)
remote: Resolving deltas: 100% (10/10), done.
remote:
remote: Create a pull request for 'test_branch' on GitHub by visiting:
remote:   https://github.com/sagars729/TestRepo/pull/new/test_branch
remote:
To github.com:sagars729/TestRepo.git
 * [new branch]      test_branch -> test_branch
Branch 'test_branch' set up to track remote branch 'test_branch' from 'origin'.
TestRepo $
```



# Pulling New Changes

To pull changes from a remote repository, you can use the **git pull** command within your repository. **Caution:** The **git status** command will not show you if the remote branch has new commits

```
TestRepo $git status
On branch test_branch
Your branch is up to date with 'origin/test_branch'.

nothing to commit, working tree clean
TestRepo $git pull
remote: Enumerating objects: 17, done.
remote: Counting objects: 100% (16/16), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 11 (delta 2), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (11/11), done.
From github.com:sagars729/TestRepo
   22f6593..b560609  test_branch -> origin/test_branch
   4bfc5eb..81544f5  master      -> origin/master
Updating 22f6593..b560609
Fast-forward
 README.md | 2 ++
 1 file changed, 2 insertions(+)
TestRepo $
```



## 3. Using The Github UI

---

Working with collaborators using the Github UI

# Viewing The Repository

Repository Owner and Name

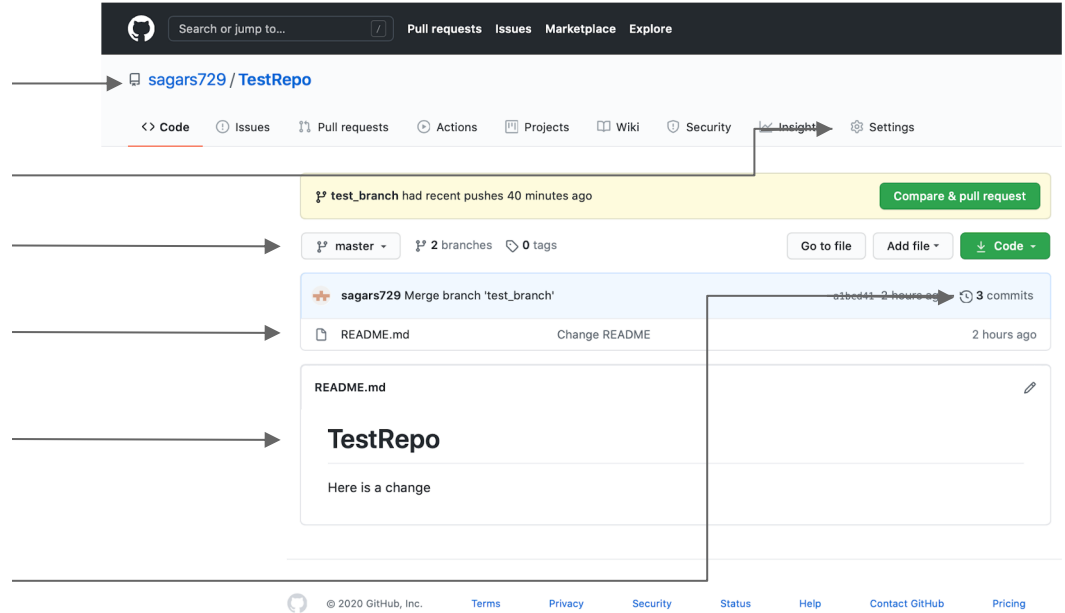
Settings

Current Branch

Files

Project README

Commits



# Adding Collaborators

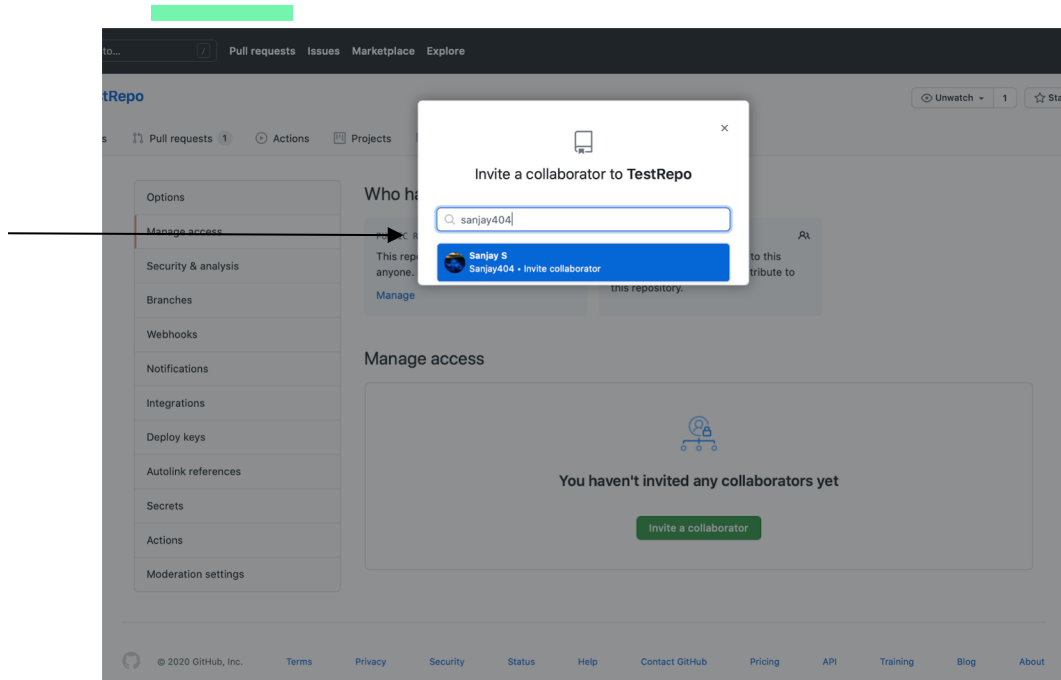
Manage Access Tab

Invite Collaborator

The screenshot shows the GitHub repository settings page. The 'Settings' tab is selected in the top navigation bar. On the left sidebar, the 'Manage access' tab is highlighted with a red vertical line and an arrow pointing to it from the text 'Manage Access Tab'. Below this, the 'Invite Collaborator' text has an arrow pointing to the 'Invite a collaborator' button in the main content area. The main content area shows 'Who has access' with two cards: 'PUBLIC REPOSITORY' and 'DIRECT ACCESS'. The 'DIRECT ACCESS' card indicates that 0 collaborators have access. Below this, the 'Manage access' section displays a message: 'You haven't invited any collaborators yet' with an icon of a person and a key, and a green button labeled 'Invite a collaborator'.

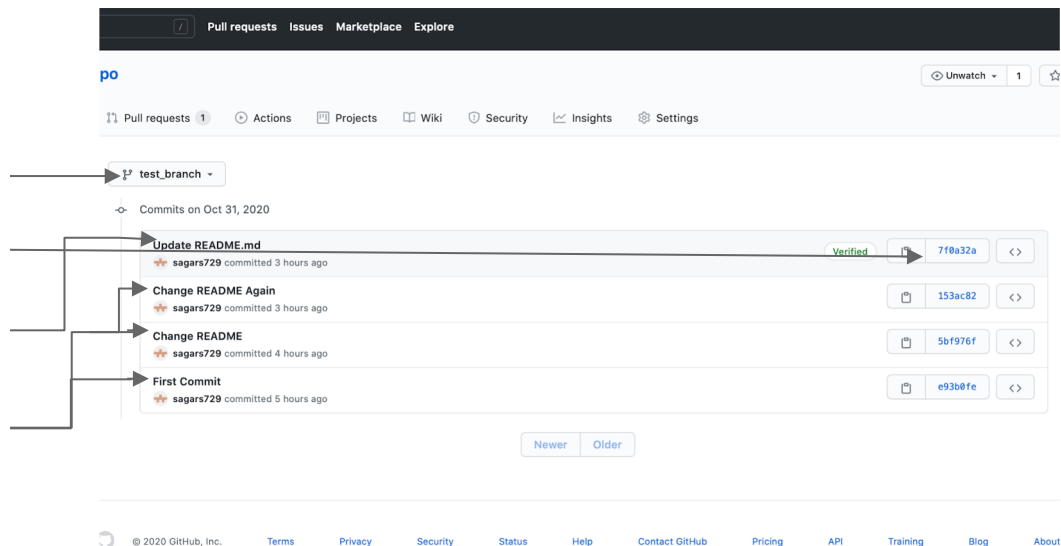
# Adding Collaborators

Add A Collaborator



# Viewing Commits and Older Versions

Current Branch  
Commit Hash  
Commit Message  
Old Versions



# Viewing Changes

Commit Message

Commit Hash

Changed Lines

The screenshot displays a GitHub pull request interface. At the top, a dark navigation bar contains links for Pull requests, Issues, Marketplace, and Explore. Below this, a light gray header shows the repository name 'epo' and various icons for Pull requests (1), Actions, Projects, Wiki, Security, Insights, and Settings. The main content area features a blue header for the pull request 'Update README.md' with a 'Browse files' button. Below the header, the commit information is shown: 'sagars729 committed 3 hours ago' with a 'Verified' badge. A commit hash '7f0a32a96e74d648d1ce32cd8de3ee1c2cd814f9' is displayed, with a link to the parent commit '1 parent: 1623e82'. The diff section shows 'Showing 1 changed file with 3 additions and 0 deletions.' The file 'README.md' is expanded, showing a diff with line numbers. The changes are highlighted in green: line 4 has a '+' sign, line 5 has a '+' sign, and line 6 has a '+' sign. The commit message 'Let's add one more change' is visible. Below the diff, there are '0 comments on commit 7f0a32a' and a 'Lock conversation' button. At the bottom, there is a comment input area with a 'Write' tab, a 'Preview' tab, and a 'Comment on this commit' button.

Update README.md

test\_branch (#1)

sagars729 committed 3 hours ago Verified

1 parent: 1623e82 commit 7f0a32a96e74d648d1ce32cd8de3ee1c2cd814f9

Showing 1 changed file with 3 additions and 0 deletions.

Unified Split

3 README.md

@@ -1,4 +1,7 @@

1 # TestRepo

2

3 Here is a change

4 +

5 Here is another change

6 +

+ Let's add one more change

0 comments on commit 7f0a32a

Lock conversation

Write Preview

Leave a comment

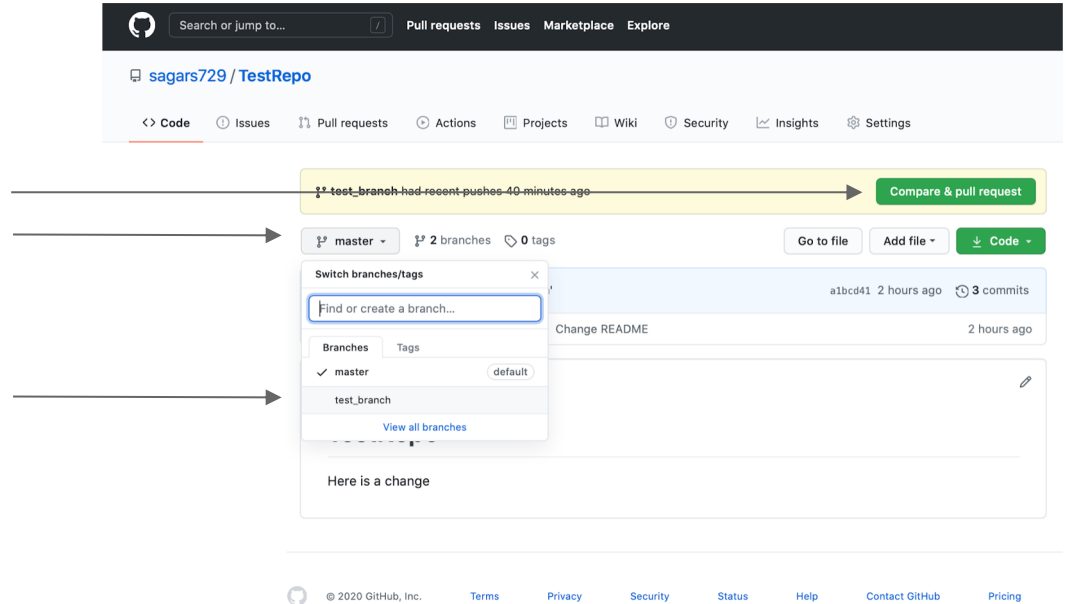
Attach files by dragging & dropping, selecting or pasting them.

Comment on this commit

# Switch Branch

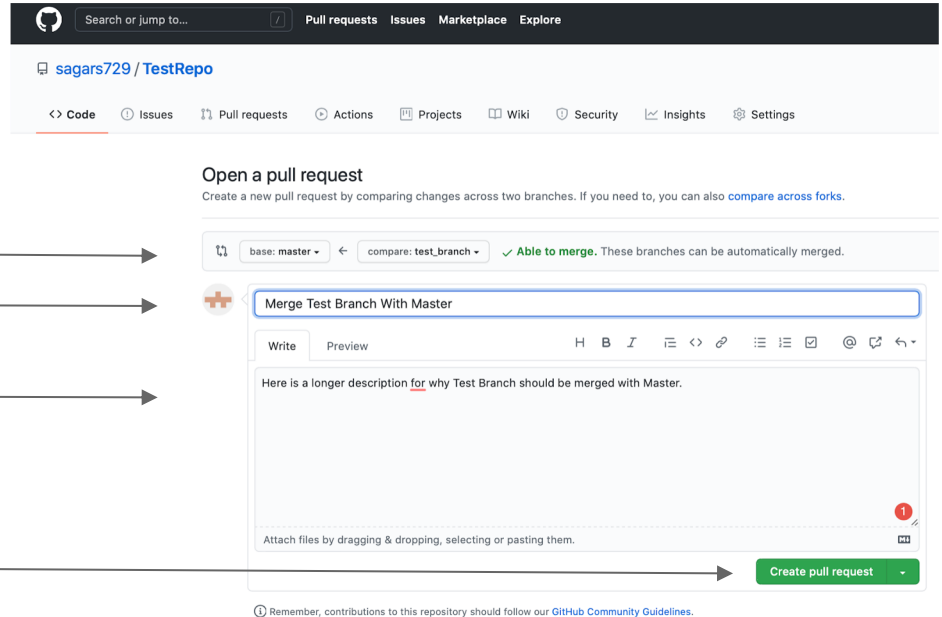
Create A Pull Request  
Switch Branch

Choose Branch





# Creating A Pull Request



The screenshot shows the GitHub web interface for creating a pull request. At the top, there's a navigation bar with the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. Below this is the repository header for 'sagars729 / TestRepo' with tabs for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The main section is titled 'Open a pull request' with a subtitle: 'Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).' Below this, there's a comparison bar showing 'base: master' and 'compare: test\_branch' with a green checkmark and the text 'Able to merge. These branches can be automatically merged.' The main content area has a title 'Merge Test Branch With Master' and a 'Write' tab. The text area contains the message: 'Here is a longer description for why Test Branch should be merged with Master.' At the bottom, there's a dashed line for attaching files and a green 'Create pull request' button. A small red circle with the number '1' is next to the button. A footnote at the bottom says: 'Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).'

Search or jump to...

Pull requests Issues Marketplace Explore

sagars729 / TestRepo

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

### Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: master ← compare: test\_branch ✓ Able to merge. These branches can be automatically merged.

#### Merge Test Branch With Master

Write Preview H B I ≡ <> ⌂ ☑ @ ↻ ↶

Here is a longer description for why Test Branch should be merged with Master.

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

Create pull request

① Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Select Branch To Merge To  
Commit Message For Merge

Longer Merge Description

Create Pull Request

\_\_\_\_\_

## Select a Team Member



# Adding Reviewer

Click To Choose Reviewer

Select a Team Member

Merge is Blocked Until Review

The screenshot displays a GitHub Pull Request titled "Merge Test Branch With Master #1". The interface includes a navigation bar with links for Pull requests, Issues, Marketplace, and Explore. Below the navigation bar, the pull request details are shown, including the status "Open" and the description "sagars729 wants to merge 2 commits into master from test\_branch". The pull request is currently blocked, as indicated by the "Review required" and "Merging is blocked" messages. A dropdown menu is open, showing a list of reviewers: "nrishnan19 Nandhini Krishnan" and "Sanjay404 Sanjay S". The "Merging is blocked" message states: "Merging can be performed automatically with 1 approving review. As an administrator, you may still merge this pull request. Merge pull request You can also open this in GitHub Desktop or view command line instructions."

# Adding Review

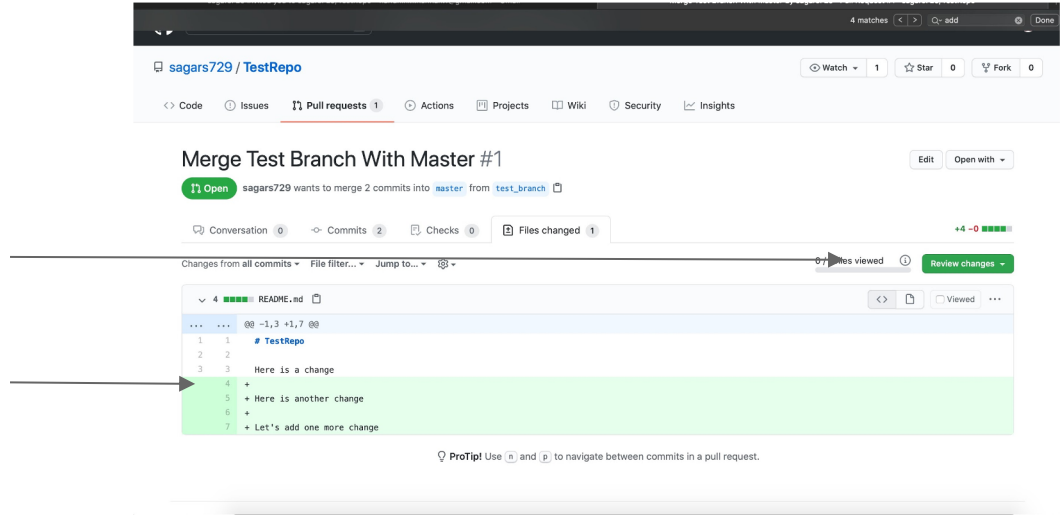
Click To Add Your Review

The screenshot shows a GitHub pull request page for the repository 'sagars729 / TestRepo'. The page title is 'Merge Test Branch With Master #1'. A yellow banner at the top states 'sagars729 requested your review on this pull request.' with a green 'Add your review' button. Below the title, it says 'sagars729 wants to merge 2 commits into master from test\_branch'. The 'Conversation' tab is selected, showing a comment from 'sagars729' that says 'Here is a longer description for why Test Branch should be merged with Master.' and a commit history with two commits: 'Change README Again' and 'Update README.ad'. On the right, the 'Reviewers' section shows 'nkrishnan19' with a note 'At least 1 approving review is required to merge this pull request.' and the 'Assignees' section shows 'sagars729'.

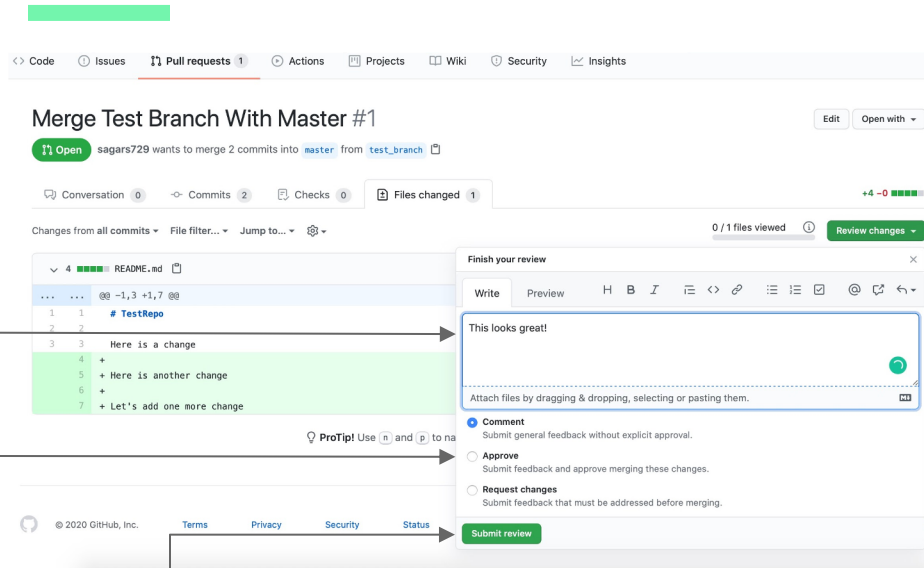
# Reviewing Changes

Click To Add Your Review

View Changes In Pull Request



# Approving Changes



Add Your Review

Choose To Comment, Approve, or Request Changes

Submit Review

# Merge Your Pull Request

See The Review

Merge Pull Request

The screenshot displays a GitHub Pull Request titled "Merge Test Branch With Master #1". The interface includes a header with "Open" and "Edit" buttons, and a summary bar showing "sagars729 wants to merge 2 commits into master from test\_branch". The main content area shows a conversation with comments from "sagars729" and "nkrishnan19". A review section at the bottom shows "Changes approved" with 1 approval, a note about GitHub Actions, and a confirmation that the branch has no conflicts. A green "Merge pull request" button is visible at the bottom of the review section. On the right side, there are sections for Reviewers, Assignees, Labels, Projects, Milestone, Linked issues, and Notifications.

Merge Test Branch With Master #1

Open Edit Open with

sagars729 wants to merge 2 commits into master from test\_branch

Conversation 0 Commits 2 Checks 0 Files changed 1 +4 -0

sagars729 commented 4 days ago

Here is a longer description for why Test Branch should be merged with Master.

sagars729 added 2 commits 4 days ago

Change README Again 153ac82

Update README.md Verified 7f8a32a

sagars729 self-assigned this 4 days ago

sagars729 requested a review from nkrishnan19 32 minutes ago

nkrishnan19 approved these changes 27 minutes ago

nkrishnan19 left a comment

This looks great!

View changes

Collaborator

Add more commits by pushing to the test\_branch branch on sagars729/TestRepo.

Changes approved

1 approving review by reviewers with write access. Learn more. Show all reviewers

1 approval

Continuous integration has not been set up

GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request You can also open this in GitHub Desktop or view command line instructions.

Reviewers

nkrishnan19 Still in progress? Convert to draft

Assignees

sagars729

Labels

None yet

Projects

None yet

Milestone

No milestone

Linked issues

Successfully merging this pull request may close these issues. None yet

Notifications Customize

Unsubscribe You're receiving notifications because you're watching this repository.

2 participants

Lock conversation

# CI/CD



Understanding Continuous Integration  
and Continuous Deployment/Delivery



# What is Continuous Integration?

---

- The coding practice of implementing small changes and frequently committing those changes
- Allows for big projects to be divided into smaller more manageable sub-tasks
- Enables large teams to work collaboratively on a single project by having each team member pick up a sub-task
- Frequently committing allows for each team member's contributions to be used by the rest of the team without much delay

# What is Continuous Delivery?

---

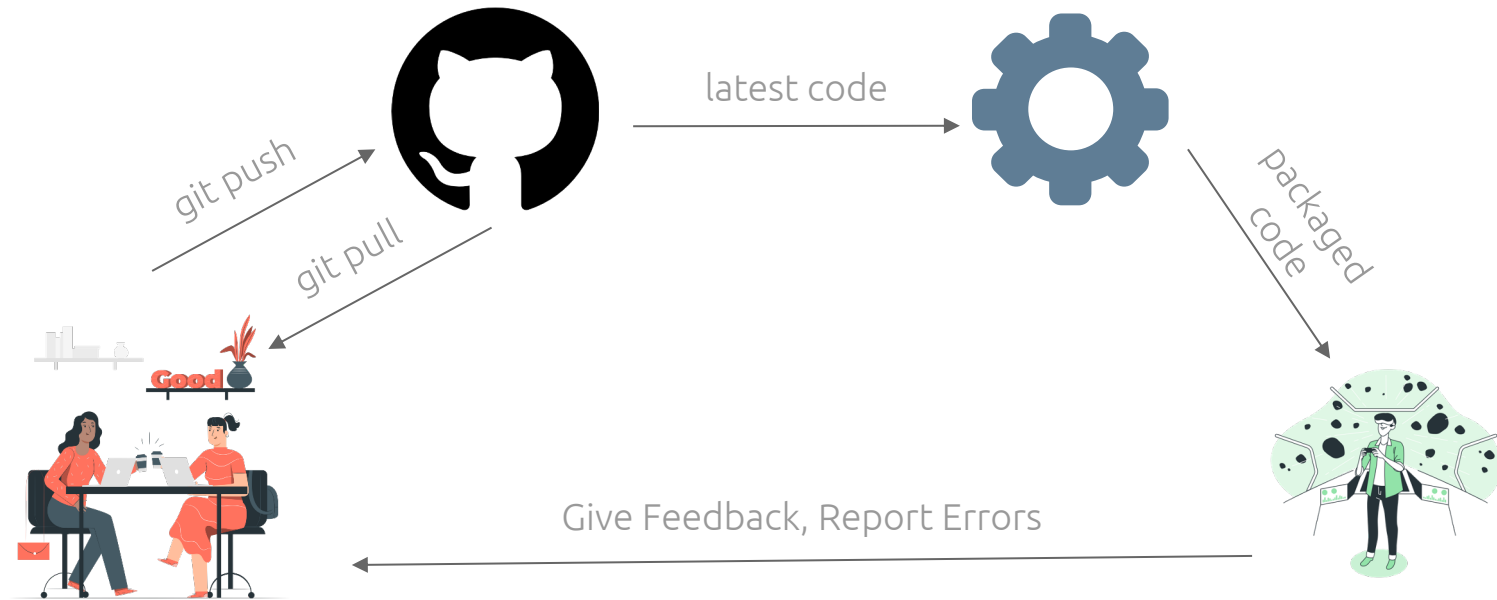
- The coding practice of automating the code release process
- As changes are continuously integrated into a system by developers, they are also continuously delivered to clients
- Requires at least one manual step of approving the deployment of code once code is ready to be delivered

# Continuous Delivery vs Continuous Deployment

---

- Continuous Delivery requires at least one step of manually approving the deployment of code once it is ready
- Continuous Deployment automates the step of approving the deployment of code
- Requires rigorous testing, monitoring, and ability to scale back

# A Quick Scenario



# CI/CD

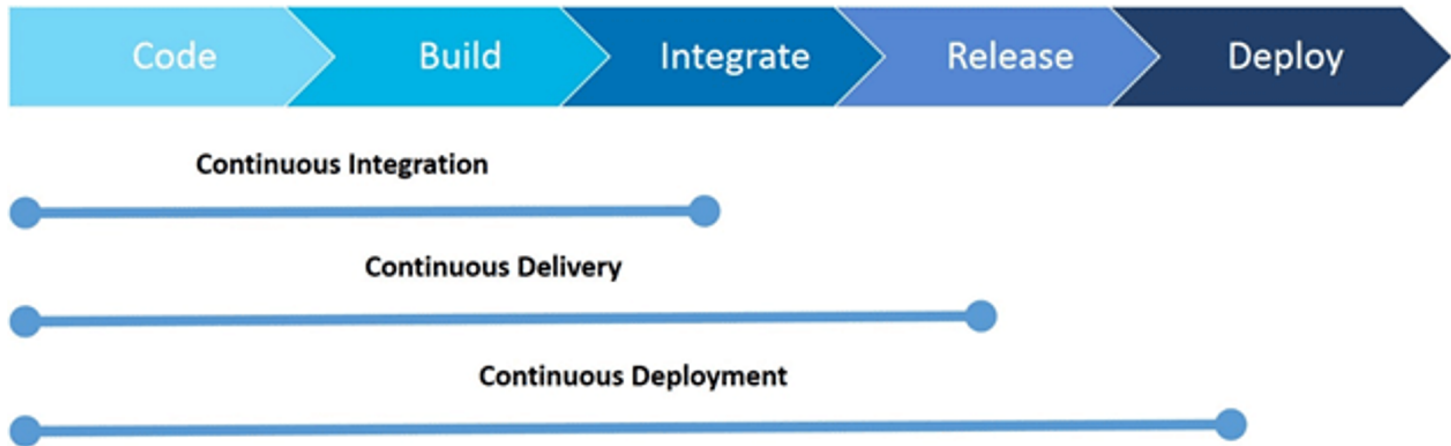


Image Source: [saviantconsulting.com](https://www.saviantconsulting.com)