AI TRAINING SERIES

GitHub

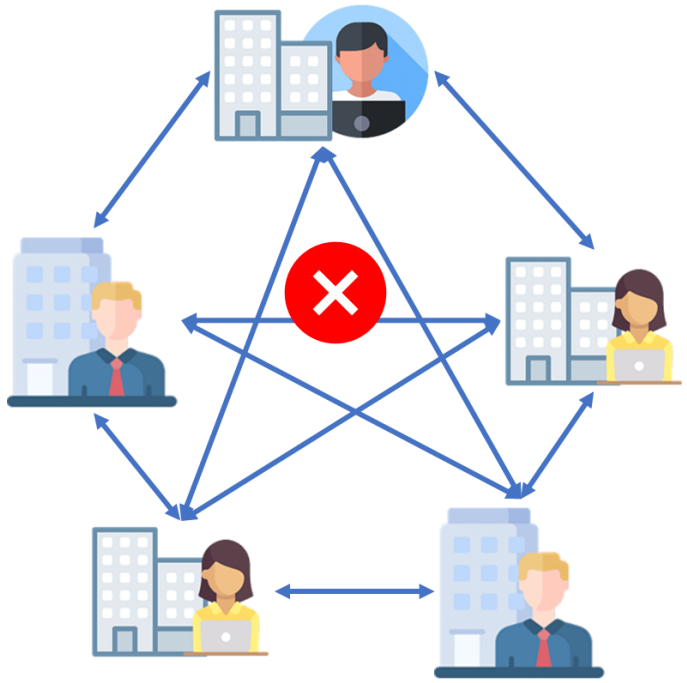
“As more and more artificial intelligence is entering into the world”

CONTENTS

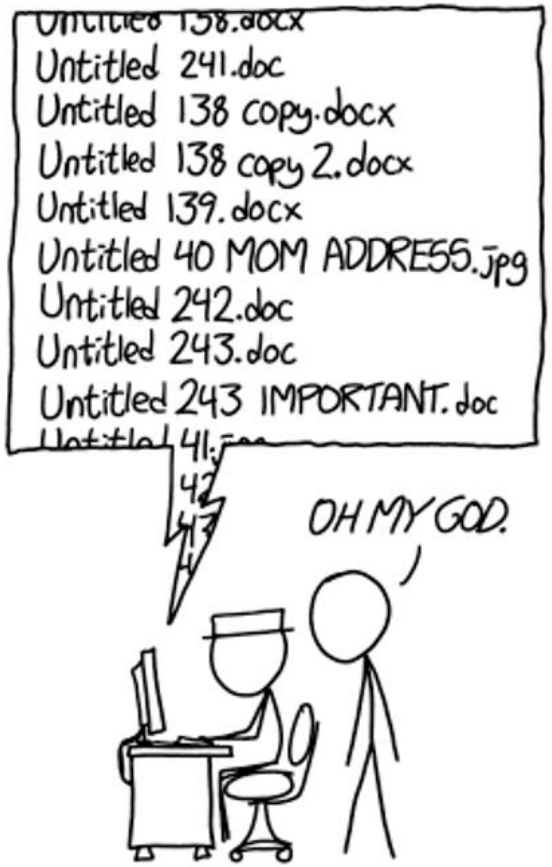
Why GitHub?

Why is Software Management needed?

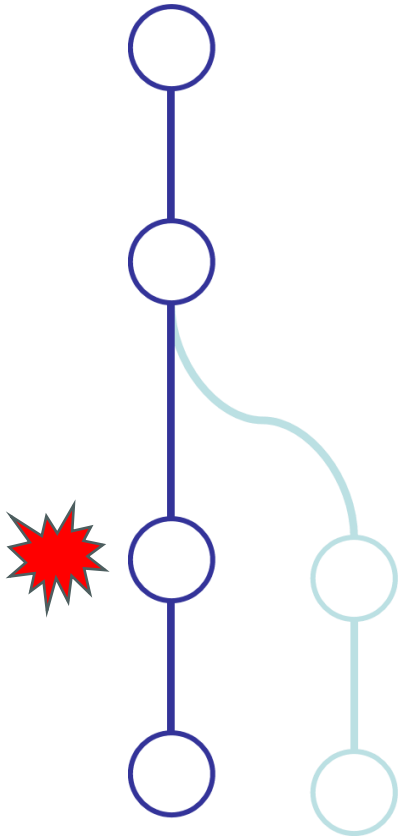
- Collaboration



- Storing/Restoring



- Figure out what happened



Git vs GitHub : What's the difference?

Git

- Used for version control
- Installed locally on computer
- Tracks changes made to a file

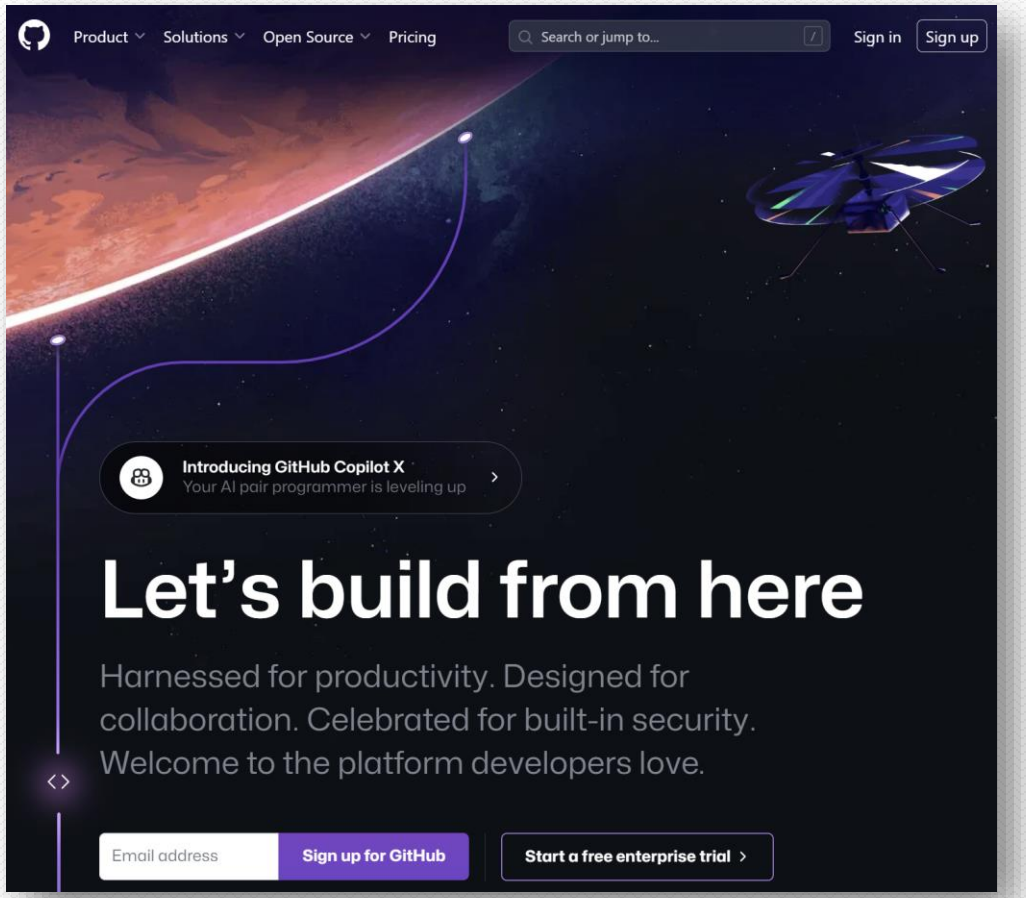
GitHub

- Cloud-based
- Used for hosting Git repository
- Provides a web interface to view file changes

GitHub

What is GitHub?

- A platform and cloud-based service for software development and version control using Git, allowing developers to store and manage code.
- It is commonly used to host open source software development projects.
- Link : https://github.com/



- Without GitHub? You can setup a remote Git repository in your local server for collaboration.

Getting Started

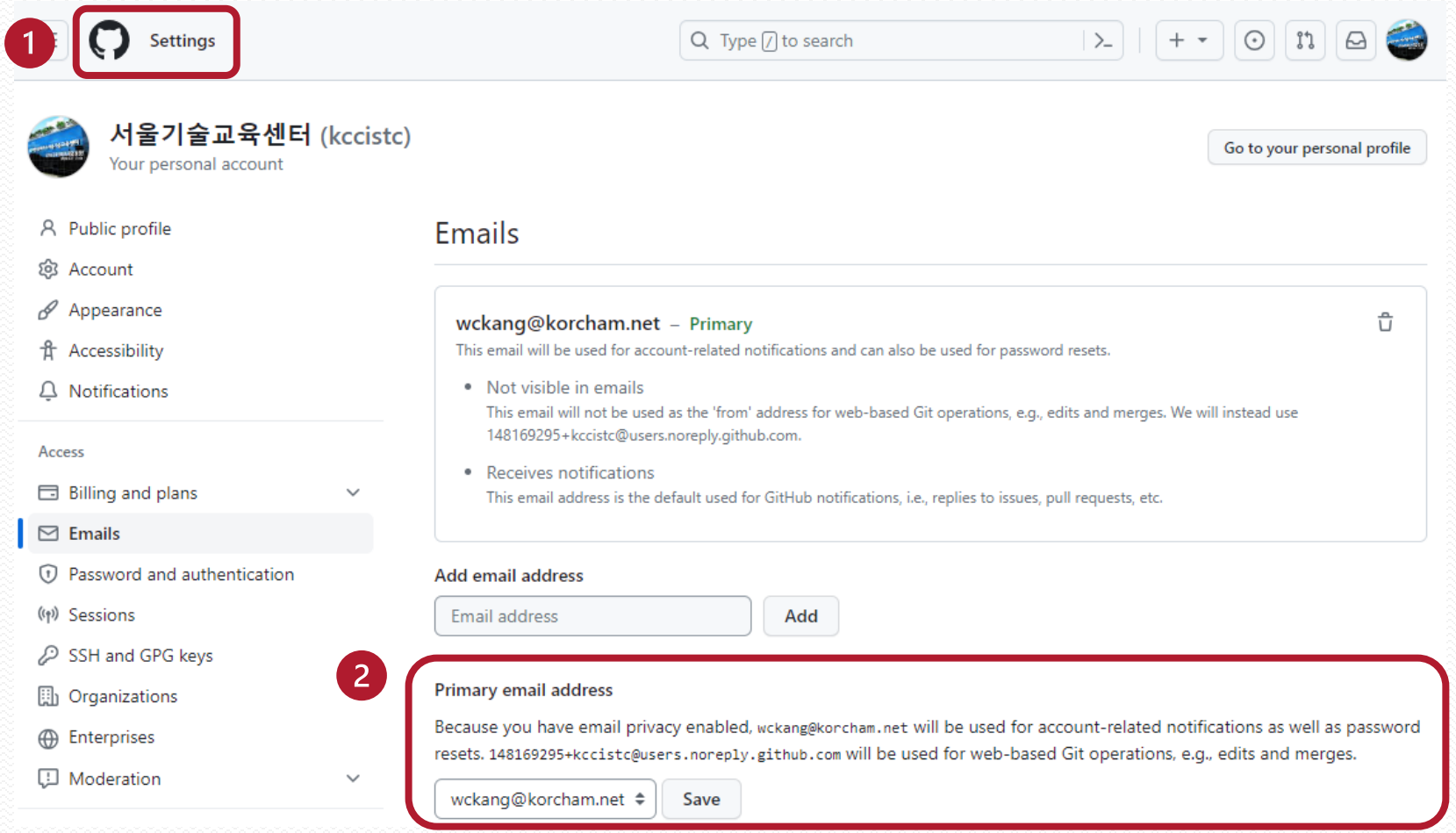
- GitHub Desktop
  - An application that enables to interact with GitHub using a GUI instead of the command line or a web browser.
  - Use GitHub Desktop to complete most Git commands from local machines with visual confirmation of changes.
  - Supported OS
    - Window
    - macOS
  - Link : https://desktop.github.com/

- GitHub CLI
  - An open source tool for using GitHub from your computer's command line.
  - Supported OS
    - Windows
    - macOS
    - Linux
  - Link : https://github.com/cli/cli
  - Command to authenticate

```
$ gh auth login
```

Getting Started

- Ensure the email address that will be associated with the commits you push from the command line as well as web-based Git operations you make



Getting Started

- The first thing you should do when you install Git is to set your user name and email address.

```
$ git config --global user.name "FirstName LastName"
$ git config --global user.email "email@example.com"
```

- Check git setting

```
$ git config --list

user.name=FirstName LastName
user.email=email@example.com
color.status=auto
color.branch=auto
...
```

- Another way to check git setting
  - Open .gitconfig
  - Windows
    - C:\Users\<user name>\.gitconfig
  - Ubuntu
    - /home/<user name>\.gitconfig

Practice : Getting Started

- Goal
  - Create GitHub account
  - Install GitHub CLI : https://github.com/cli/cli

```
$ gh auth login

? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: D8FF-6166
Press Enter to open github.com in your browser...

✓ Authentication complete.
- gh config set -h github.com git_protocol https
✓ Configured git protocol
! Authentication credentials saved in plain text
✓ Logged in as mokiya
```

Practice : Getting Started

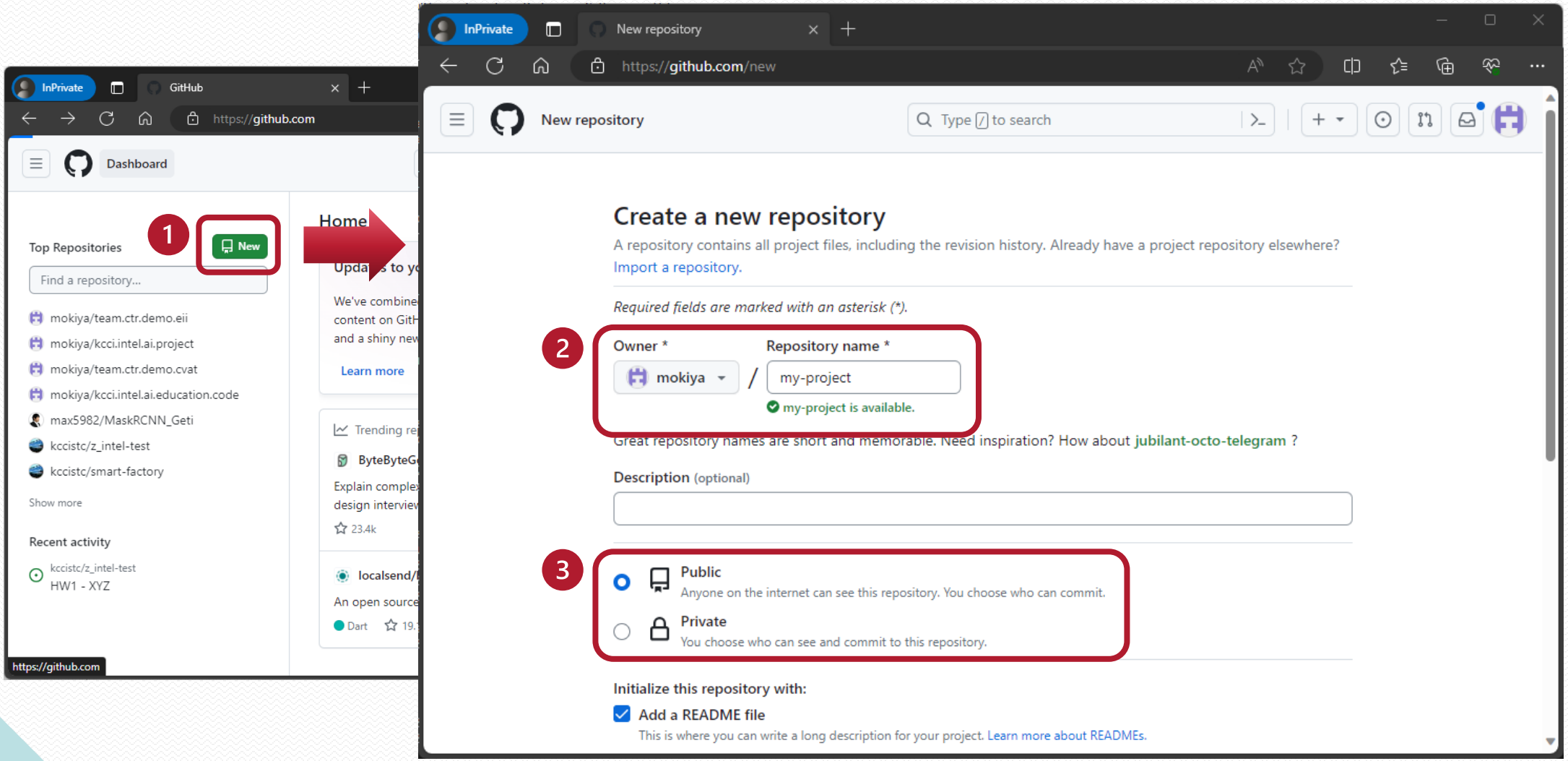
- Goal
  - Set primary email in GitHub
  - Set name & email in local machine

```
$ sudo apt-get update
$ sudo apt-get install git

$ git config --global user.name "FirstName LastName"
$ git config --global user.email "email@example.com"
```

Setting Up a GitHub Repository

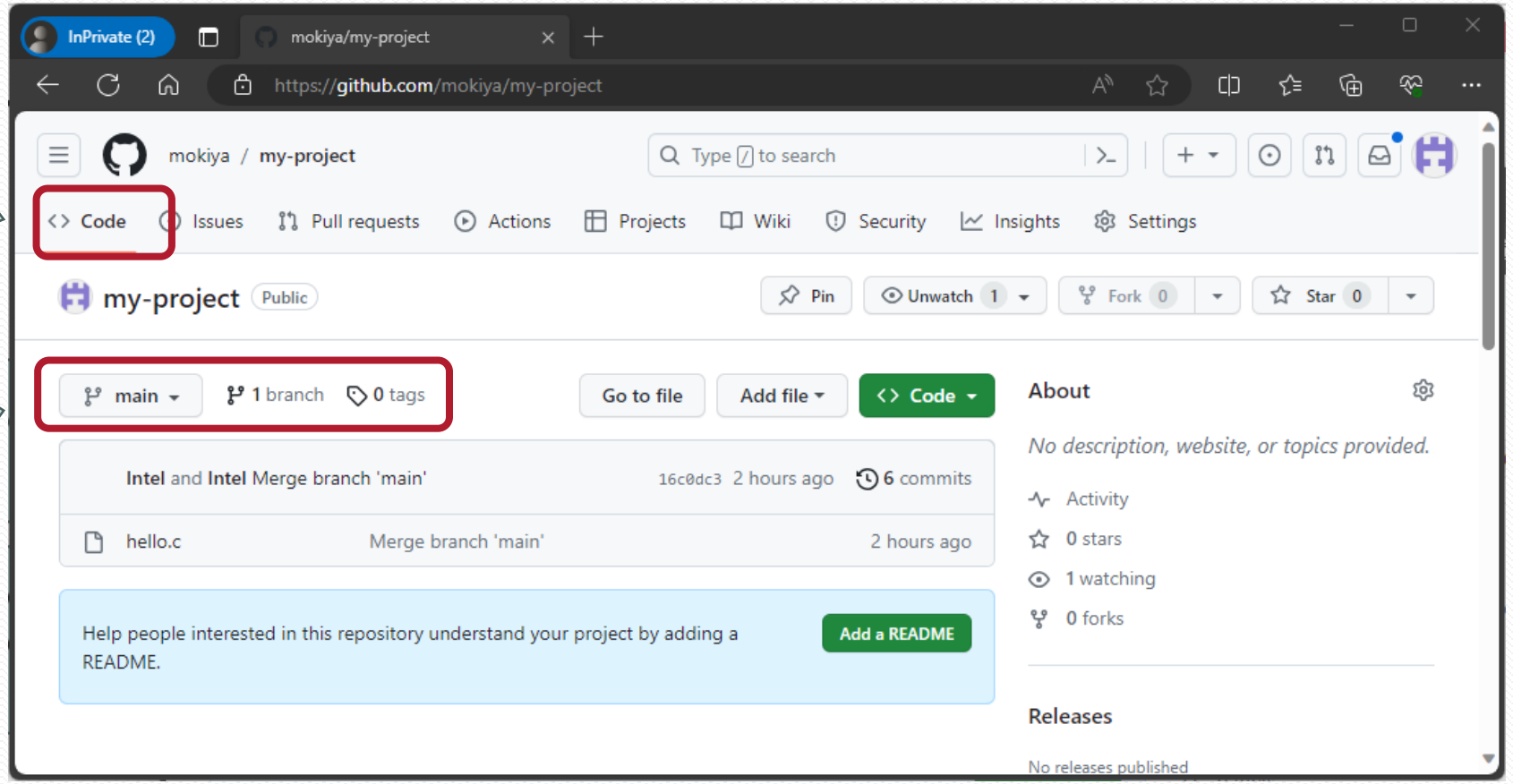
- Create new project in the GitHub



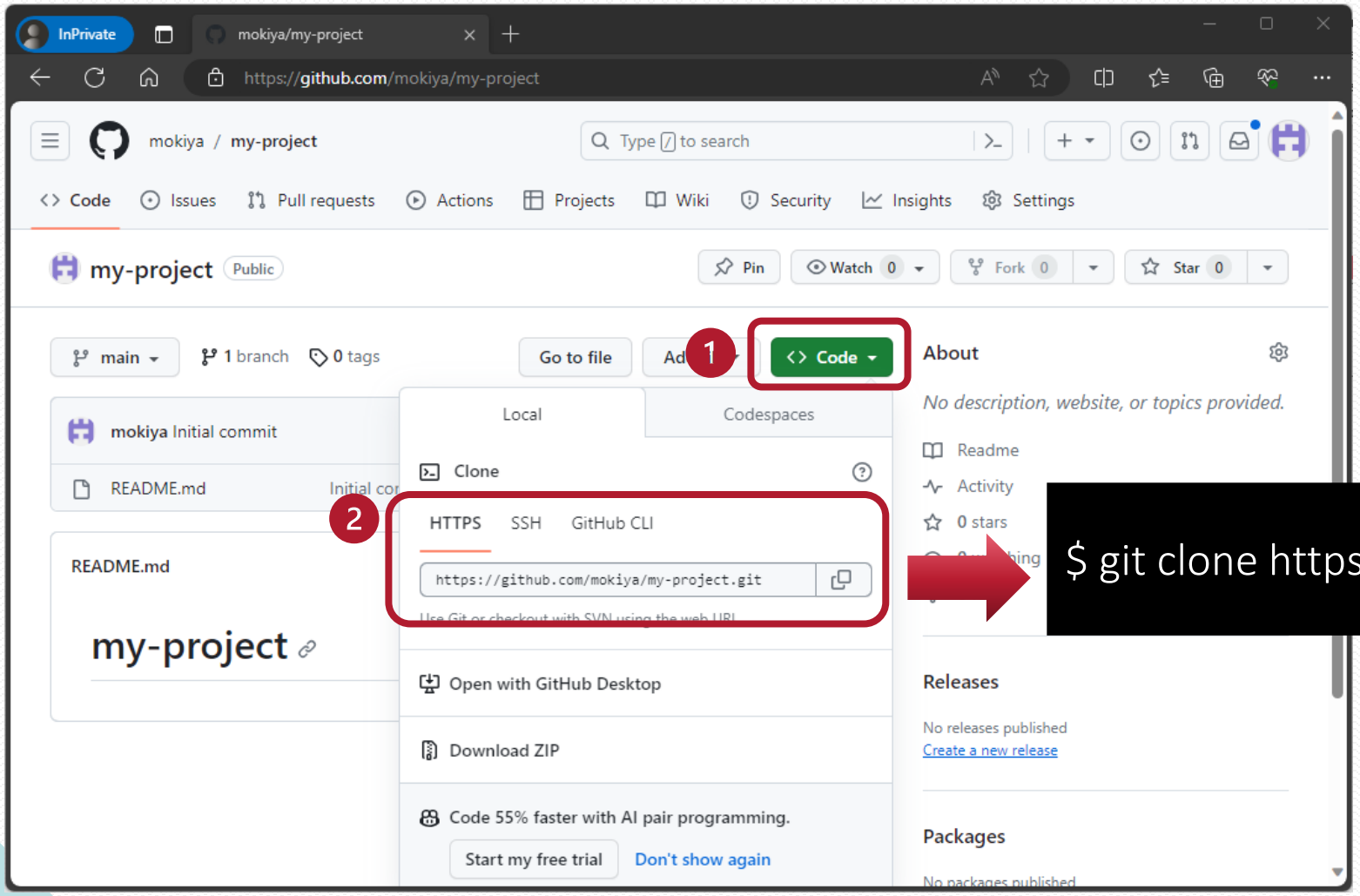
Setting Up a GitHub Repository

Web Interface to see project source codes

Choose branch or tag what you want to a specific snapshot



Setting Up a GitHub Repository



$ git clone https://github.com/mokiya/my-project.git

Setting Up a GitHub Repository

- Once you successfully created GitHub repository, you can start adding commits from scratch and pushing commits to new GitHub repository.

- If you already have an existing repository, you can push an existing repository.

```
$ cd ~/existing-repo

$ git remote add github https://github.com/new-repo-url.git
$ git branch -M main
$ git push -u github main
```

Practice : Setting Up a GitHub Repository

- Goal
  - Create new GitHub repository
  - Practice same way as we did last time, but try to practice using github.
  - Hints)
    - git clone
    - git remote
    - git push

Practice : Setting Up a GitHub Repository

- An existing repository

```
$ cd ~/git-training
$ git clone ~/git-training/my-project.git -b main project-z
$ cd ./projext-z

$ git remote add github https://github.com/mokiya/my-project.git
$ git branch -M main
$ git push -u github main
```

- Clone new Github repository to check, or visit GitHub web.

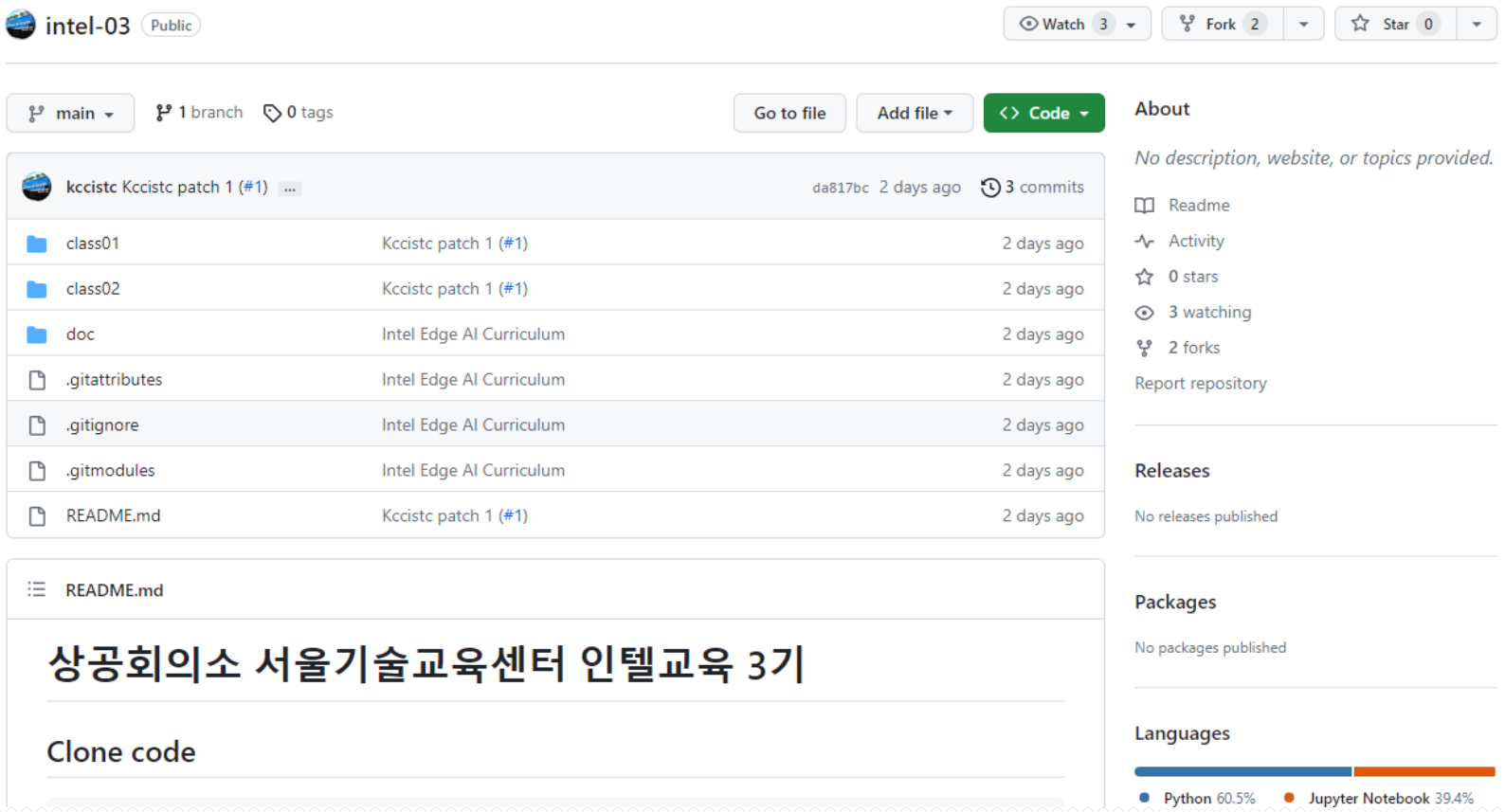
```
$ mkdir ~/github-training

$ cd ~/github-training
$ git clone https://github.com/mokiya/my-project.git -b main
$ cd ./my-project
$ git log
```

Intel Curriculum GitHub Repository

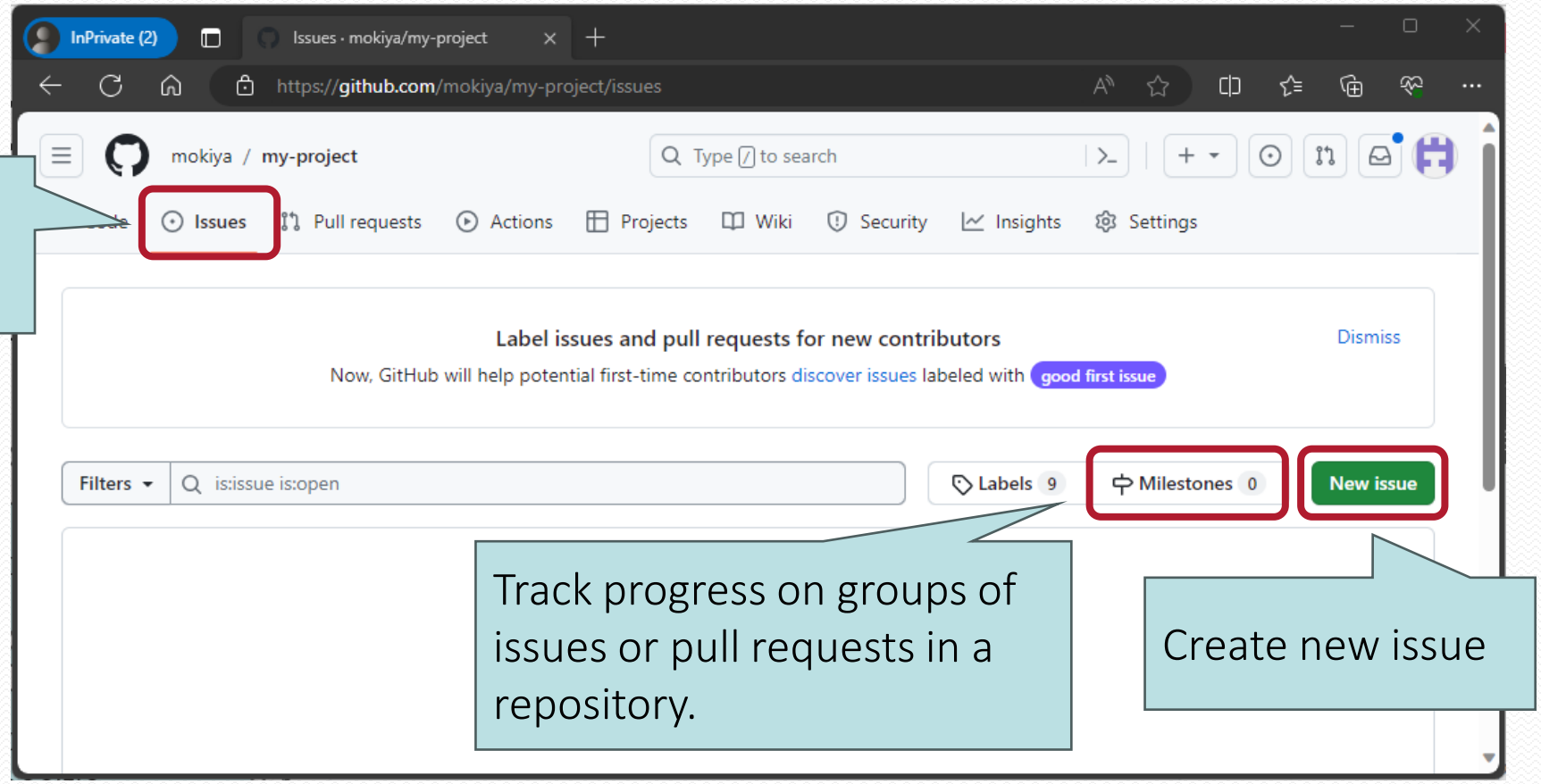
- Materials, sample codes, homework, team projects and so on for Intel curriculum will be managed in Intel Curriculum GitHub Repository.

Intel Curriculum GitHub Repository : https://github.com/kccistc/intel-03/



Creating Issue

- GitHub Issues to track ideas, feedback, tasks, or bugs for work on GitHub.



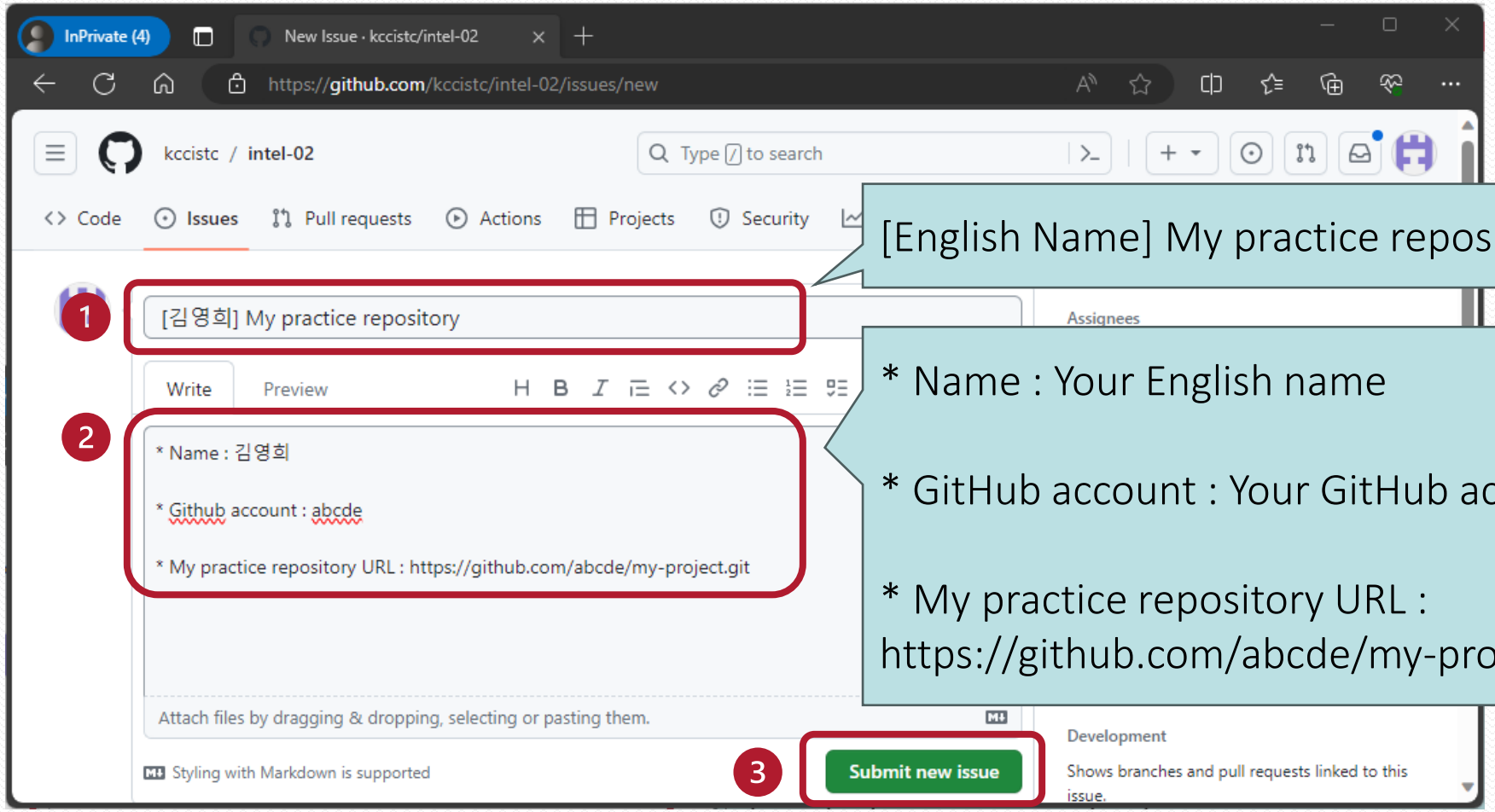
Web Interface to see project source codes

Track progress on groups of issues or pull requests in a repository.

Create new issue

Practice : Creating Issue

- Goal
  - Submit new issue with following contents in Intel Curriculum Repository



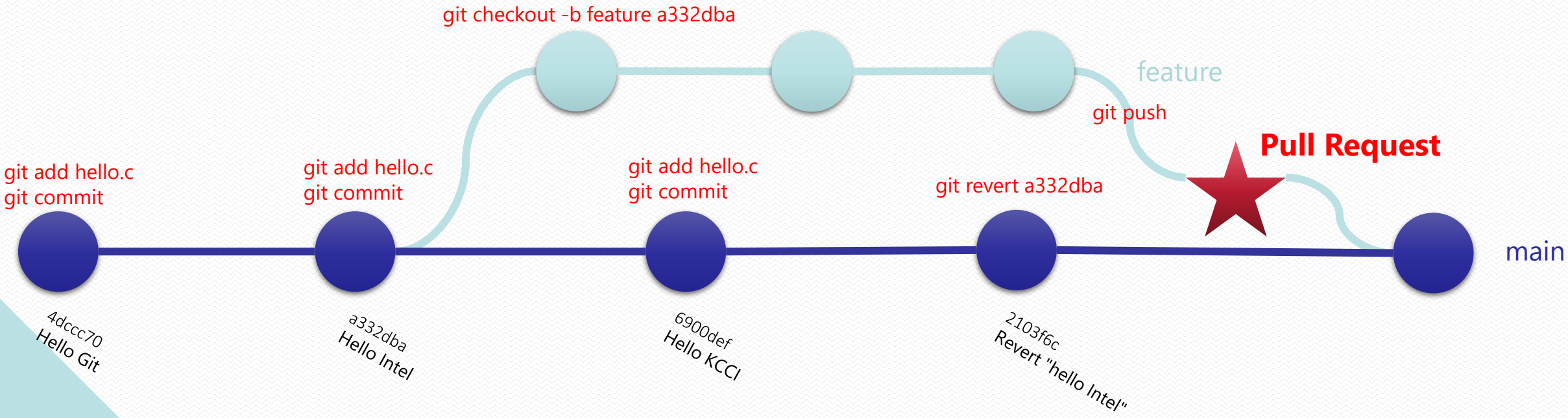
Practice : Creating Issue

- Goal
  - Based on your created issue, teacher will …
    - Give Intel Curriculum Repository permission to your GitHub account
    - Visit your practice repository and review your practice.

  - Please check teacher’s comment in your created issue, then follow up teacher’s feedback.

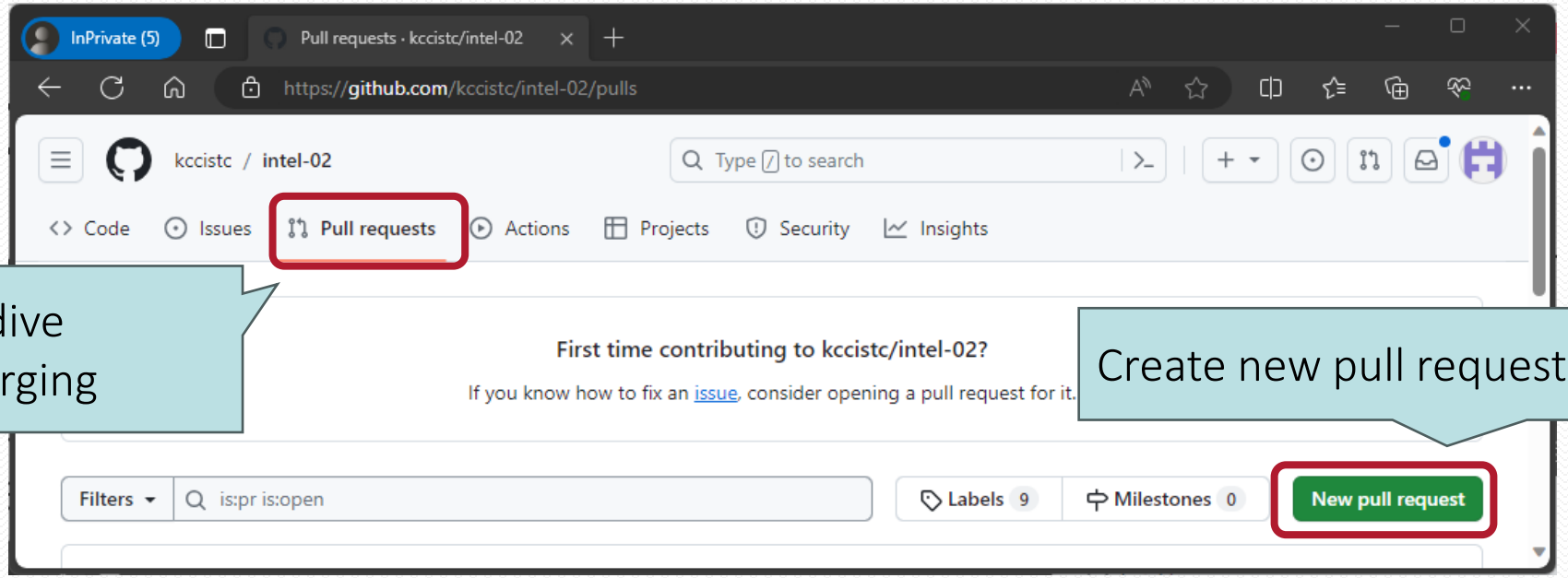
Creating Pull Request

- GitHub is a web-based hosting service for Git repositories.
- In simple terms, you can use Git without GitHub, but you CANNOT use GitHub without Git.
- GitHub offers Pull Request as web interface to discuss changes for merging.



Creating Pull Request

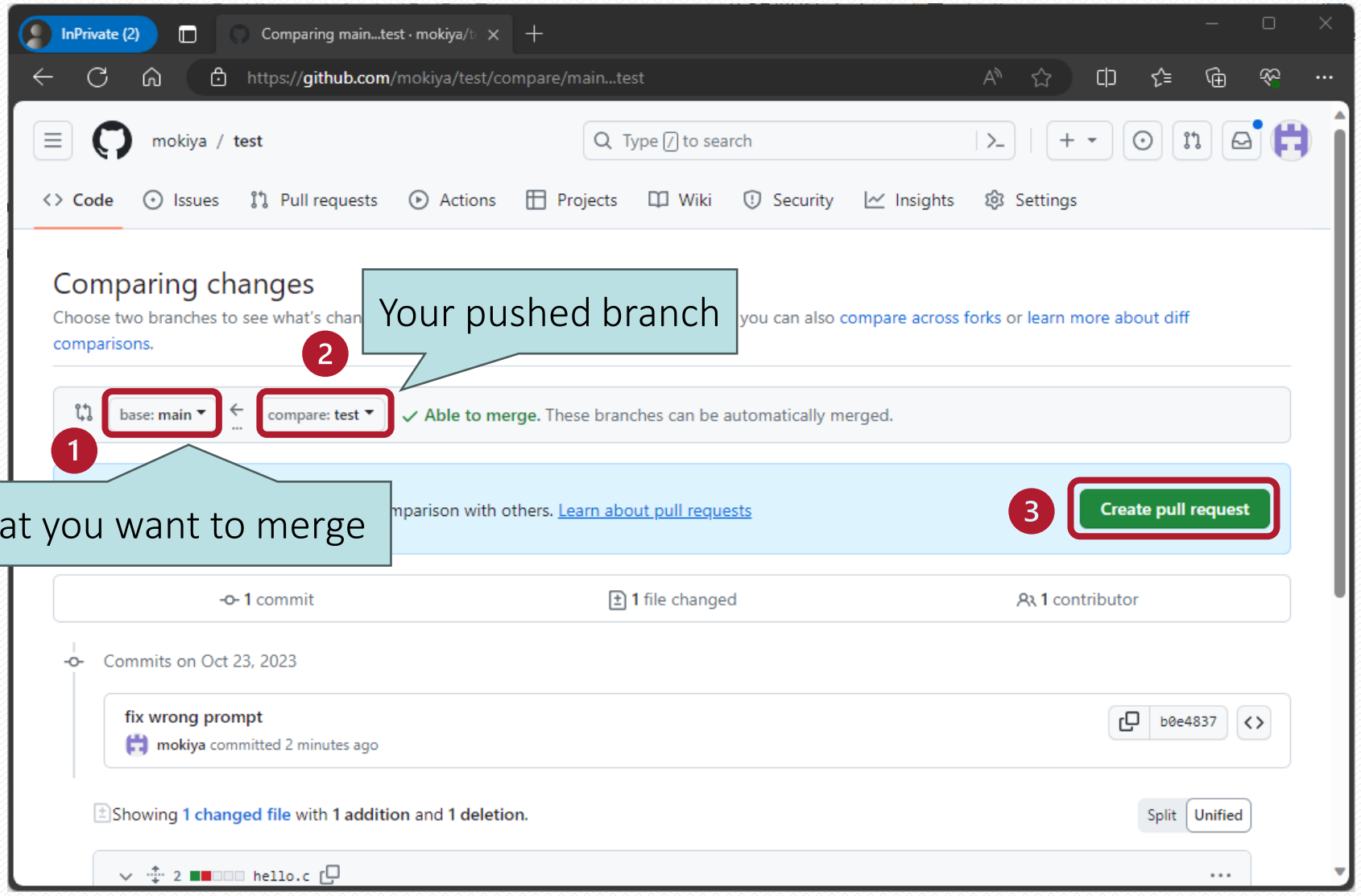
- Pull Request tells others about changes you've pushed to a branch in a repository on GitHub.
- Once a pull request is opened, you can discuss and review the potential changes with collaborators and add follow-up commits before your changes are merged into the base branch.



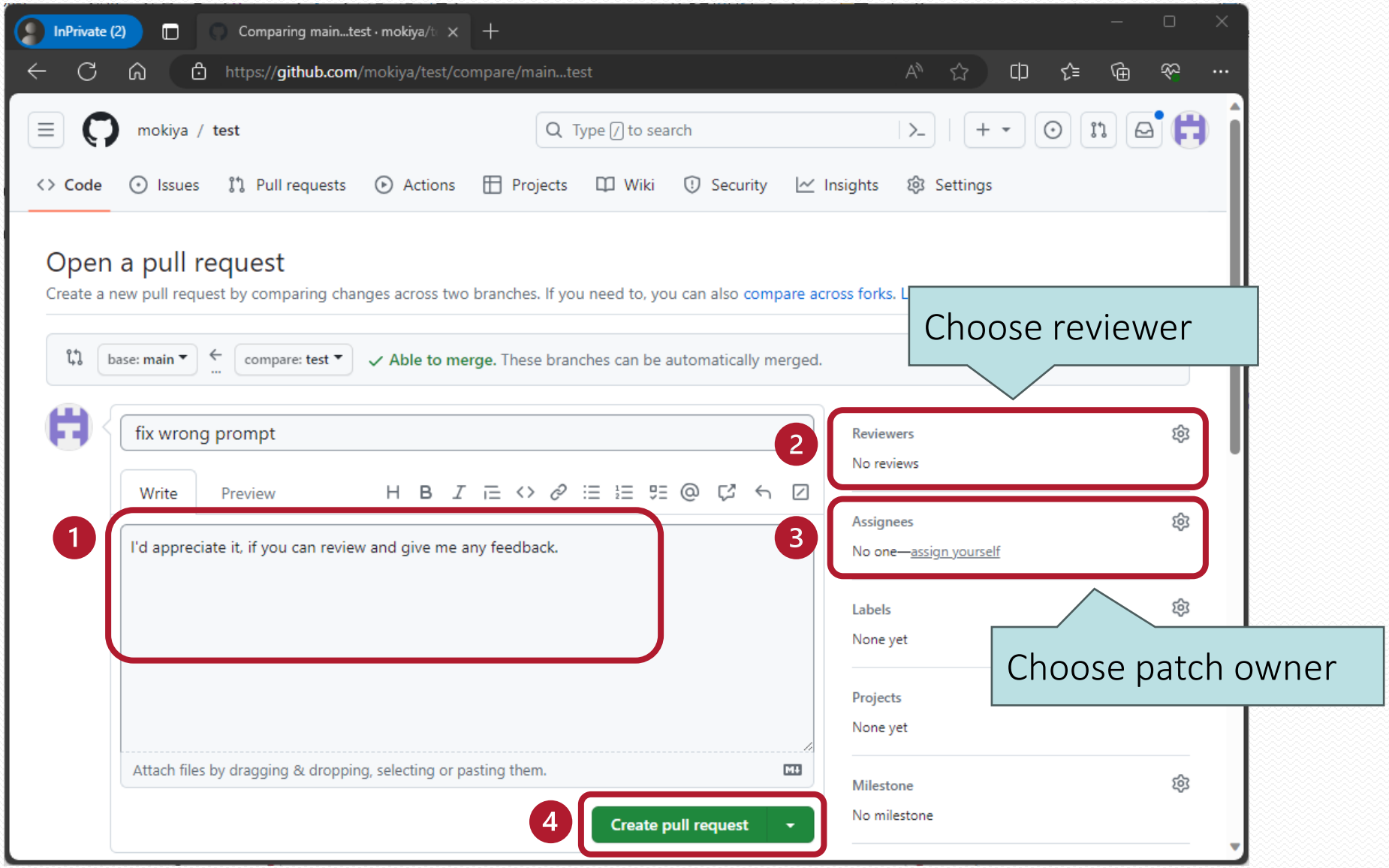
Web Interface to have deep-dive discussion on changes for merging

Create new pull request

Creating Pull Request



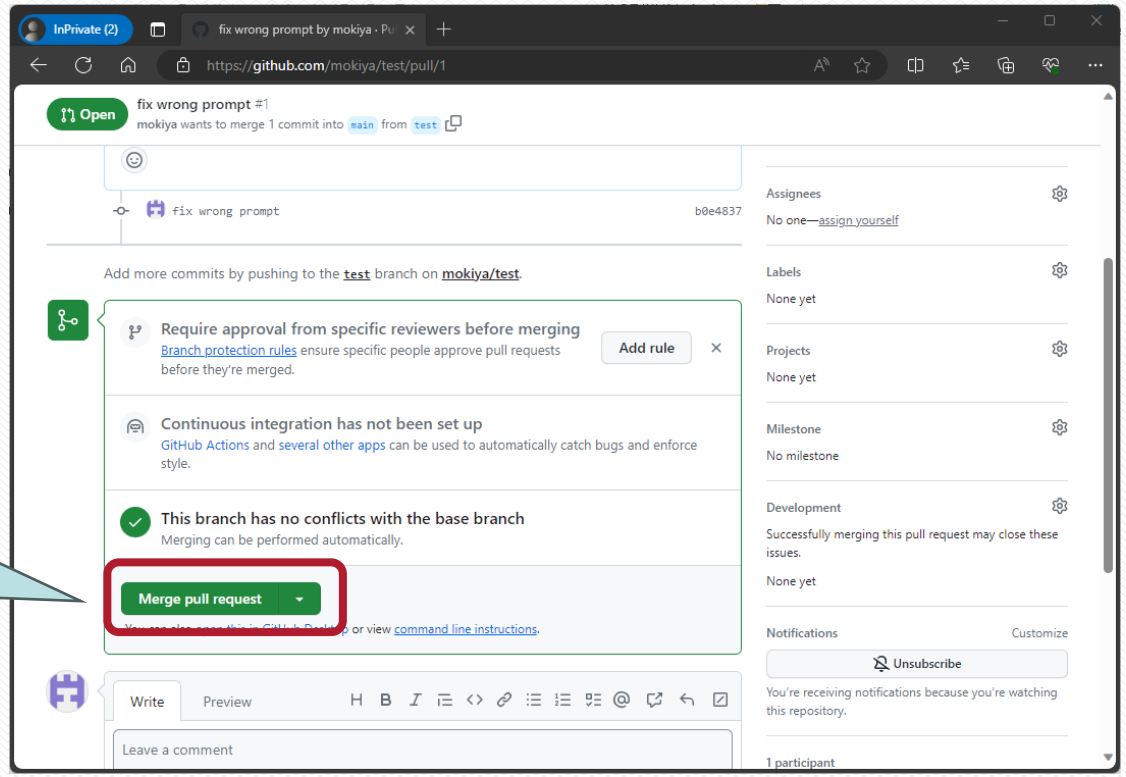
Creating Pull Request



Creating Pull Request

- Once a pull request is opened, issue discussion will start.
- You might need to rework/re-upload changes based on reviewer feedback.
- You might need to rework/re-upload changes, if there are conflicts.

If reviewer approved to merge patches, patch owner can merge changes.



Practice : Creating Pull Request

- Goal
  - Clone Intel Curriculum GitHub.
  - Create commits with following changes
    - Create directories named as your English name under class01/homework/ and class01/homework/
    - Push commits to a new branch following the naming convention below
    - Naming convention : class01-hw2-[github account]
    - Example) if GitHub account is abcdef, branch name should be class01-hw2-abcdef.
  - Create Pull Request
    - Base branch should be “main”
    - Reviewer should be “mokiya”
  - Once reviewer approve your changes, merge your changes in created Pull Request

Ground Rules for Intel Curriculum

Create an Issue with mandatory fields below using GitHub Issue in Intel Curriculum GitHub Repository

- Title / Comment
- Milestone
- Assignee
- Development

Create a Pull Request with mandatory fields in Intel Curriculum GitHub Repository

- Base branch : main
- Reviewer

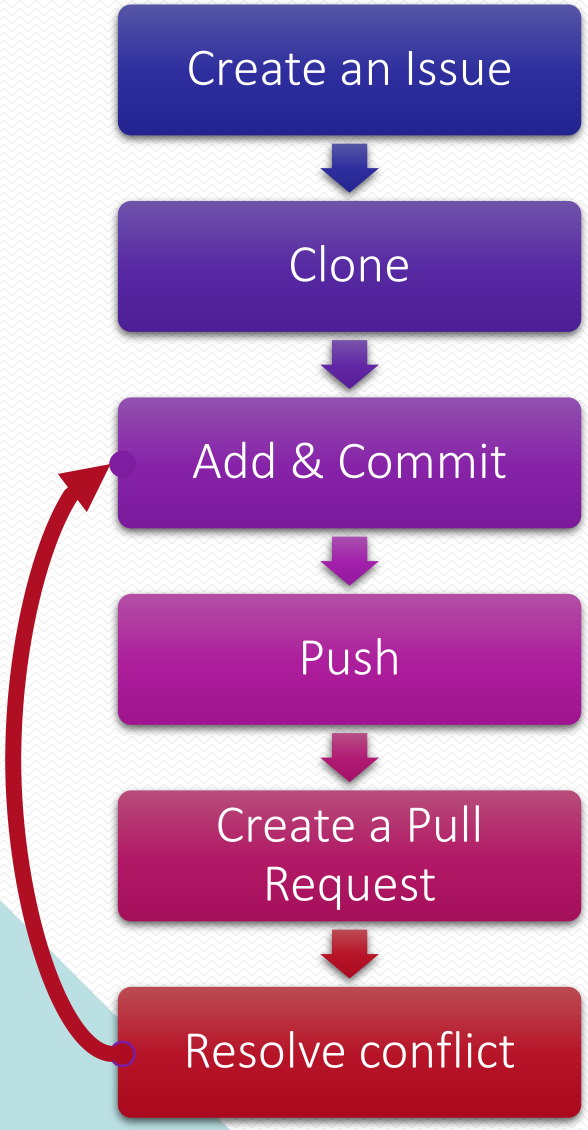
Based on feedback or conflicts, might need to re-work & re-push changes.

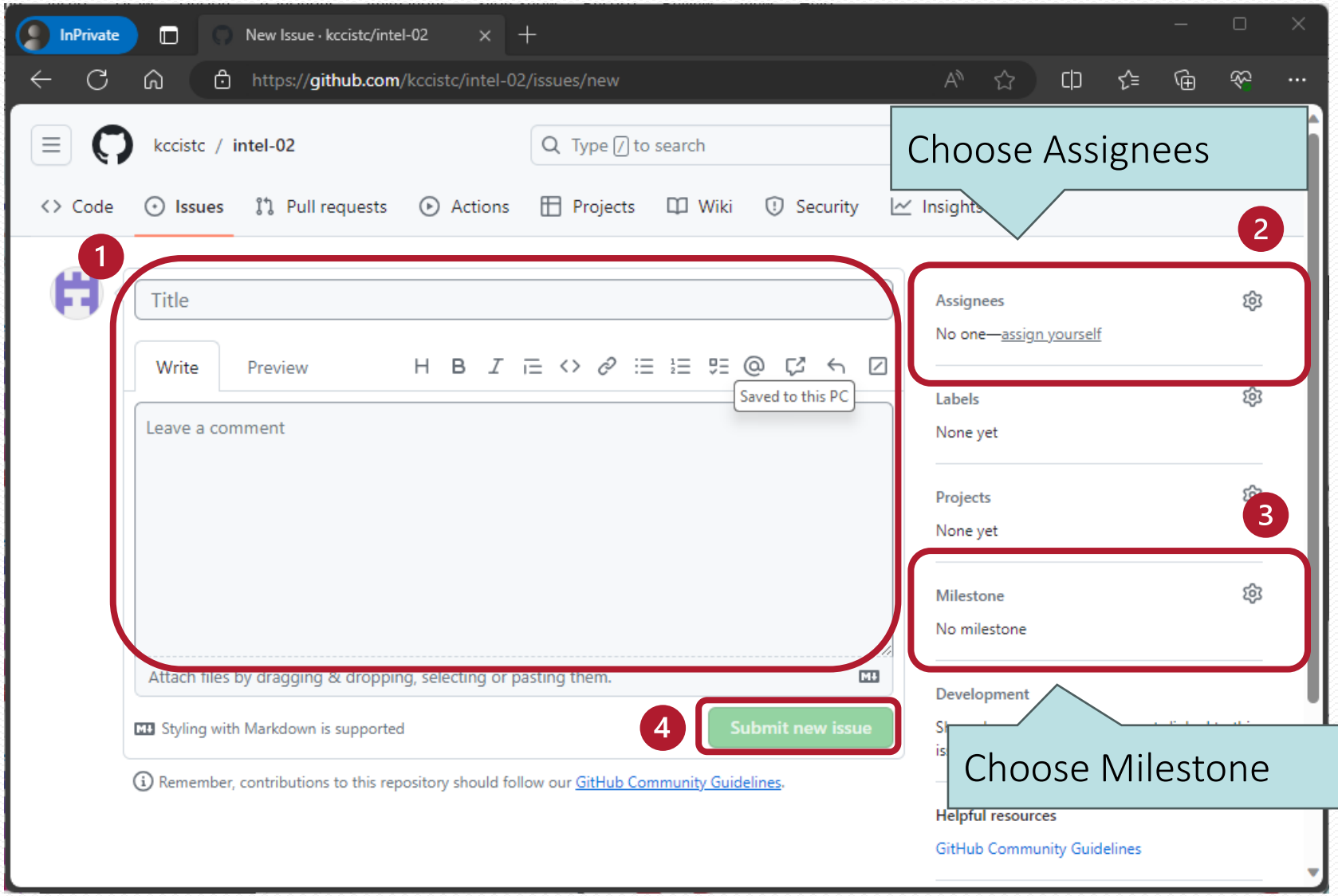
Push commits to a new branch following the naming convention below

- Naming convention : class[xx]-hw[y]-[GitHub account]
- Example) GitHub account : abcdef | Course : class01 | Homework 2
  Branch name should be class01-hw2-abcdef.

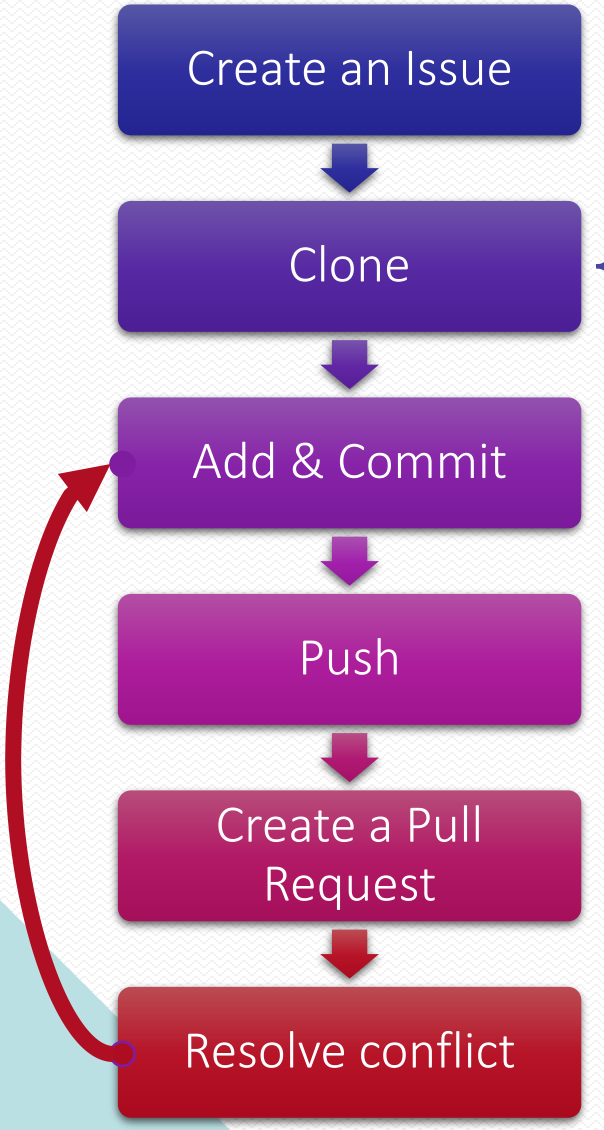
Once reviewer agrees to merge, merge your patches in GitHub

Homework Workflow based Ground Rules





Homework Workflow based Ground Rules

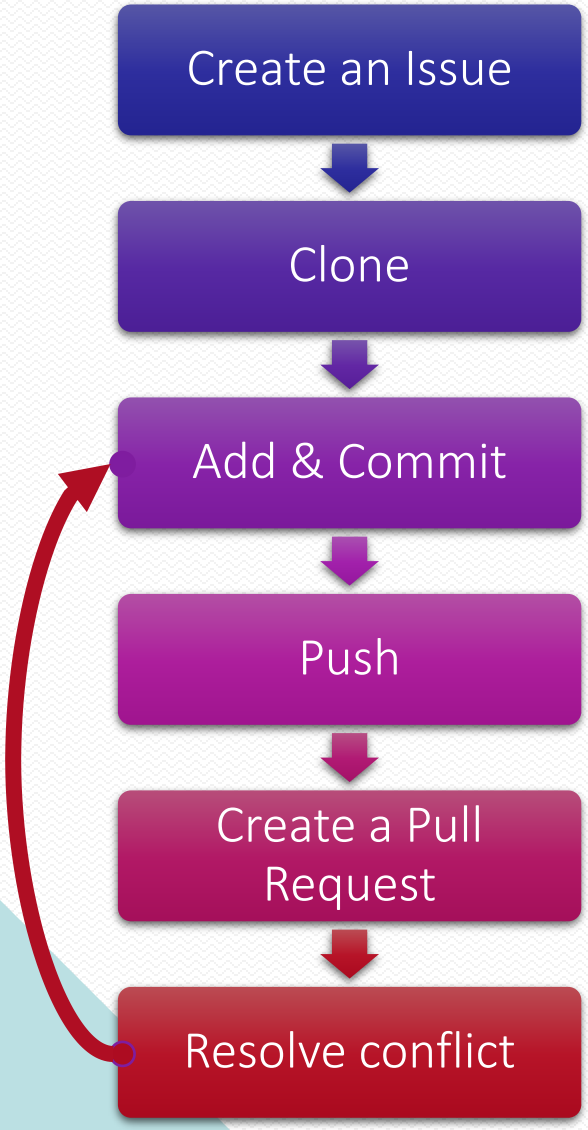


- A Git command line utility which is used to target an existing repository and create a clone, or copy of the target repository.

```
$ git clone <Repository GitHub URL> -b <Branch or Tag Name> <Directory>
```

  - Clone Branch called <Branch or Tag Name> from the repository located at <Repository GitHub URL> into the folder called <Directory> on the local machine.

Homework Workflow based Ground Rules



- A Git command adds a change in the working directory to the staging area.

```
$ git add <File name 1> <File name 2> <Directory 1>
```

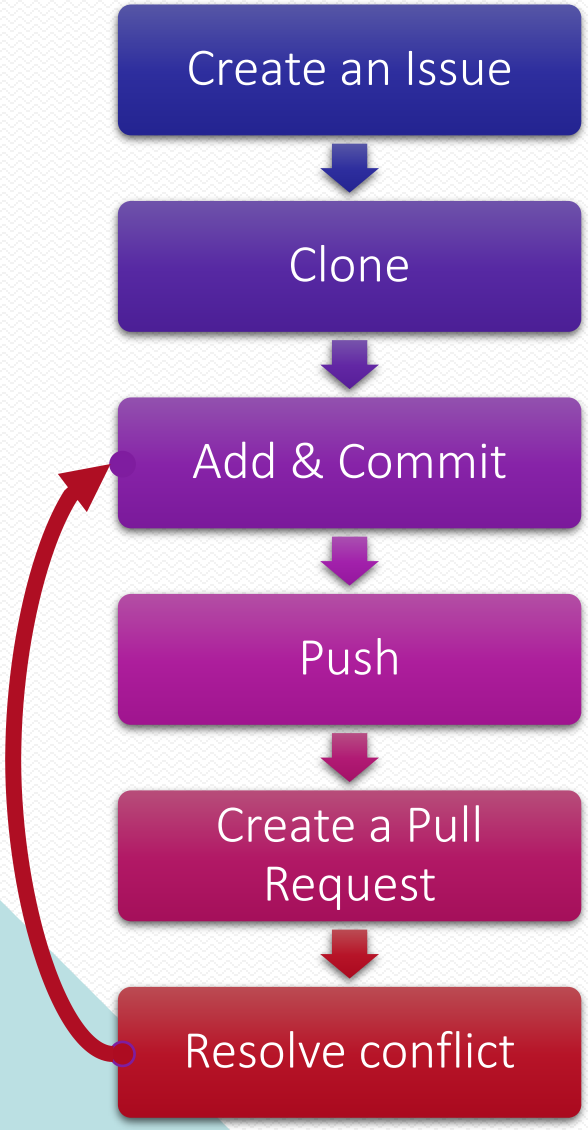
  - Allow <File name 1>, <File name 2>, <Directory> to stage files to be committed.

- A Git command creates a commit, which is like a snapshot of your repository. These commits are snapshots of your entire repository at specific times.

```
$ git commit -m <Comment>
```

  - Without -m option, default text editor will be opened to create the commit message.

Homework Workflow based Ground Rules



- A Git command is used to upload local repository content to a remote repository.

```
$ git push <Remote> HEAD:<Branch Name>
```

  - Push your changes called HEAD: to specific branch (<Branch Name>) in the remote repository (<remote repository>)
  - You can use commit id or local branch name instead of HEAD:
- Naming convention : class[xx]-hw[y]-[GitHub account]

  Ex) GitHub account : abcdef

  Course : class01

  Homework 2 : hw2

  Branch name should be class01-hw2-abcdef

Homework Workflow based Ground Rules

Create an Issue

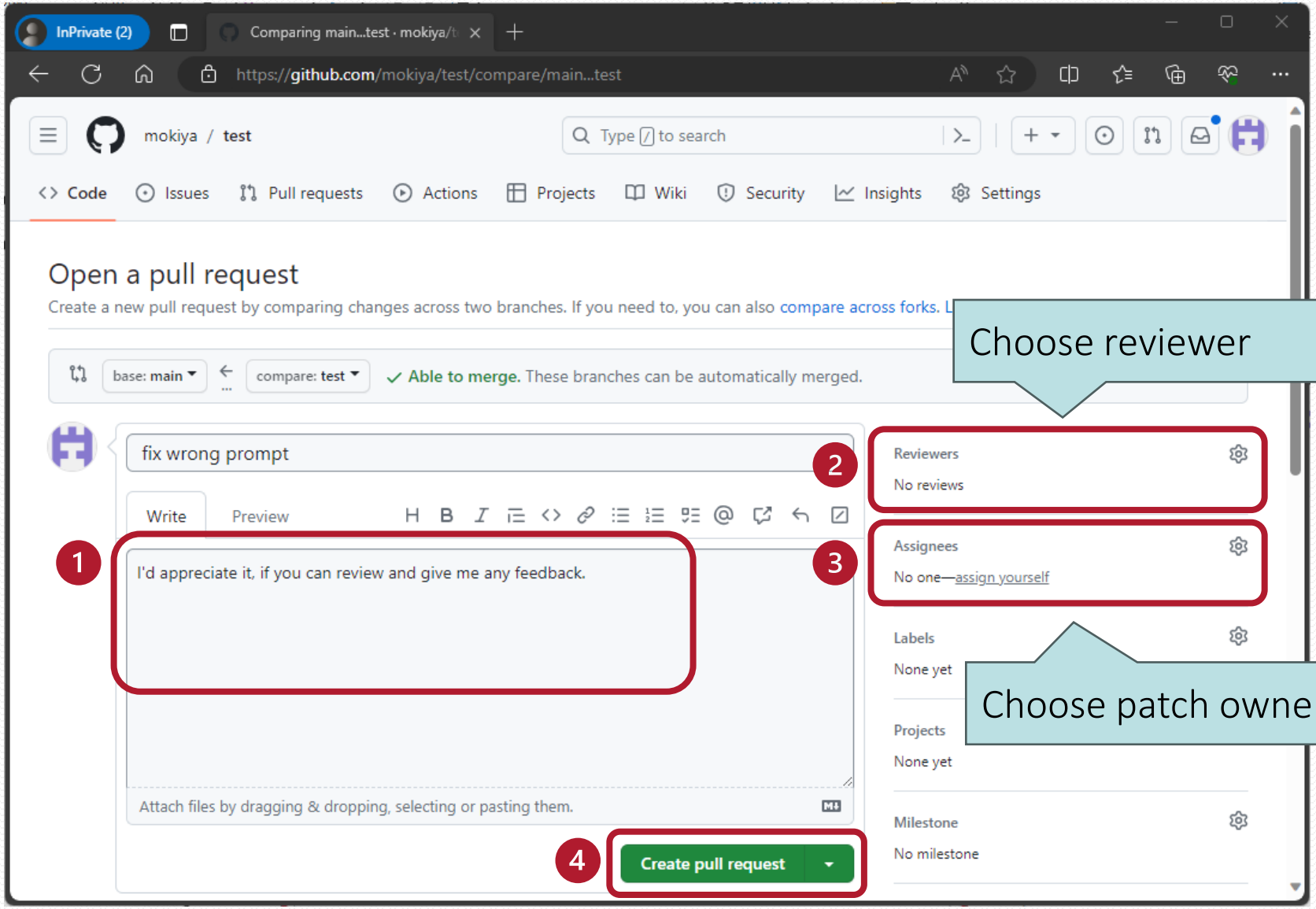
Clone

Add & Commit

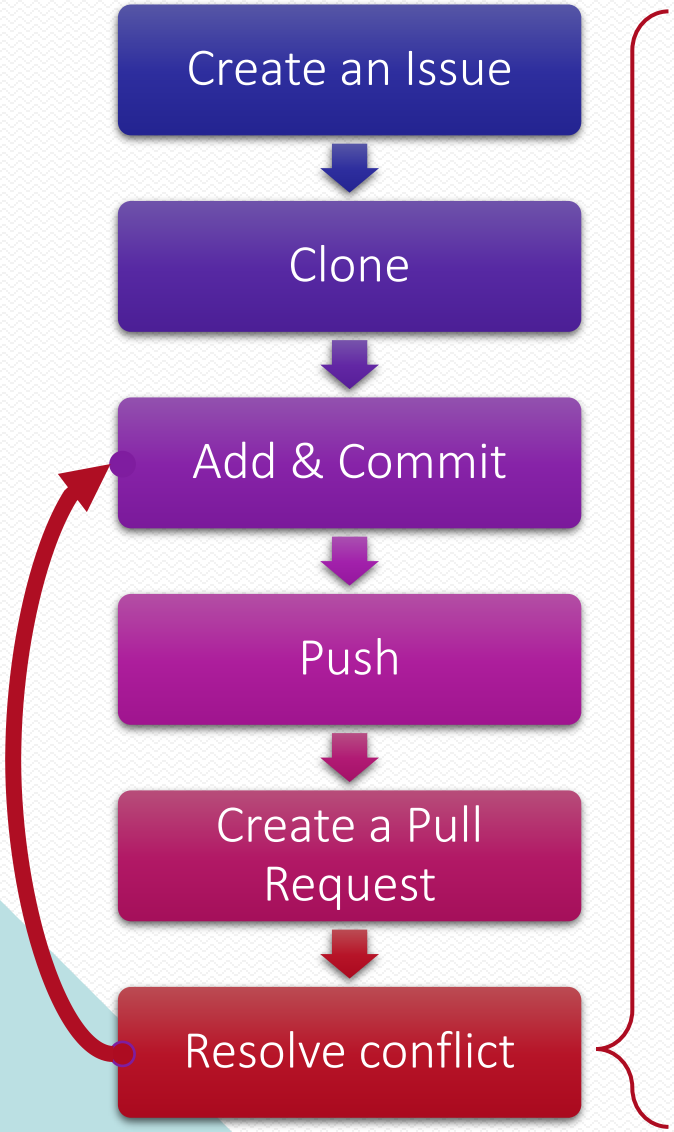
Push

Create a Pull Request

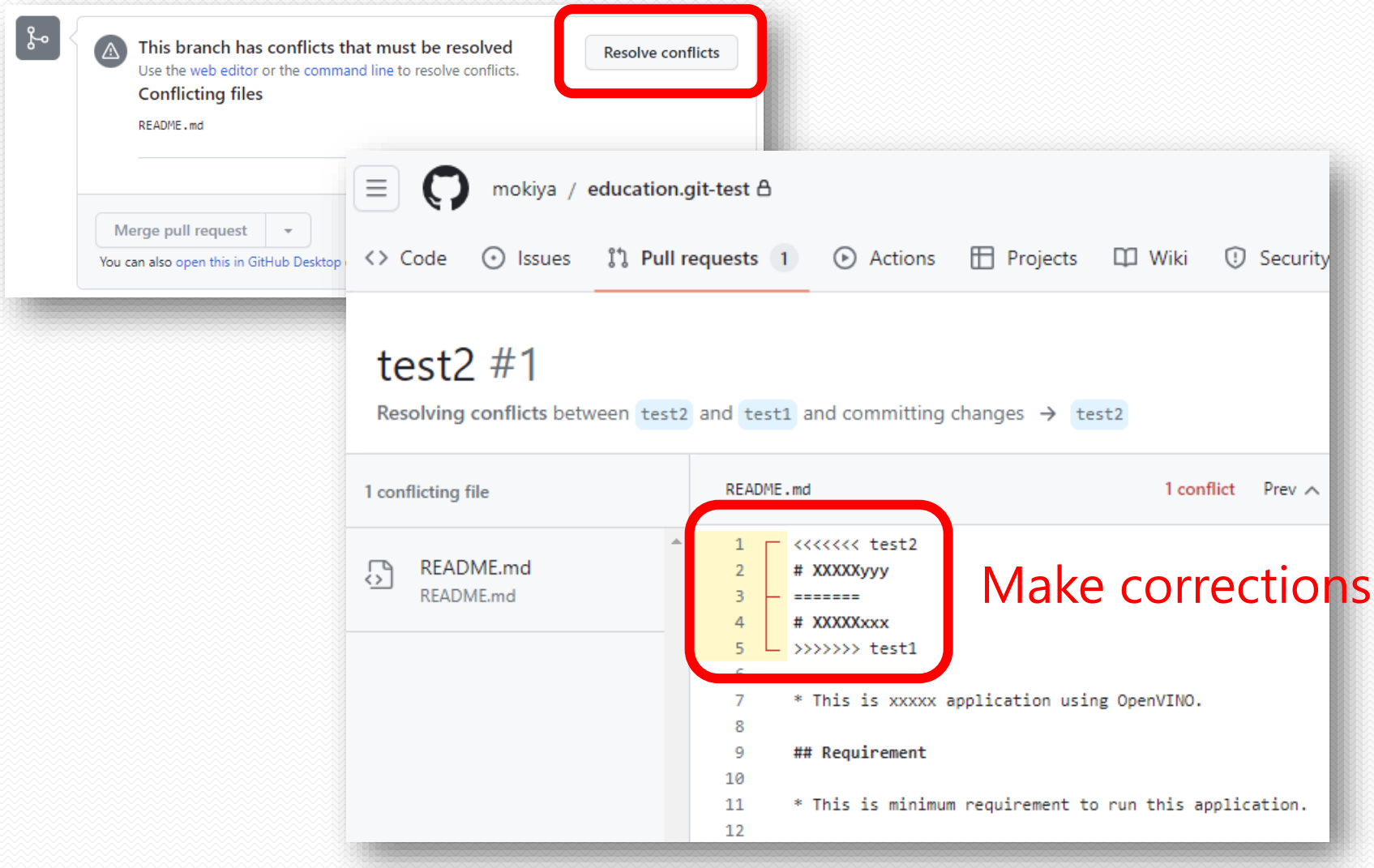
Resolve conflict



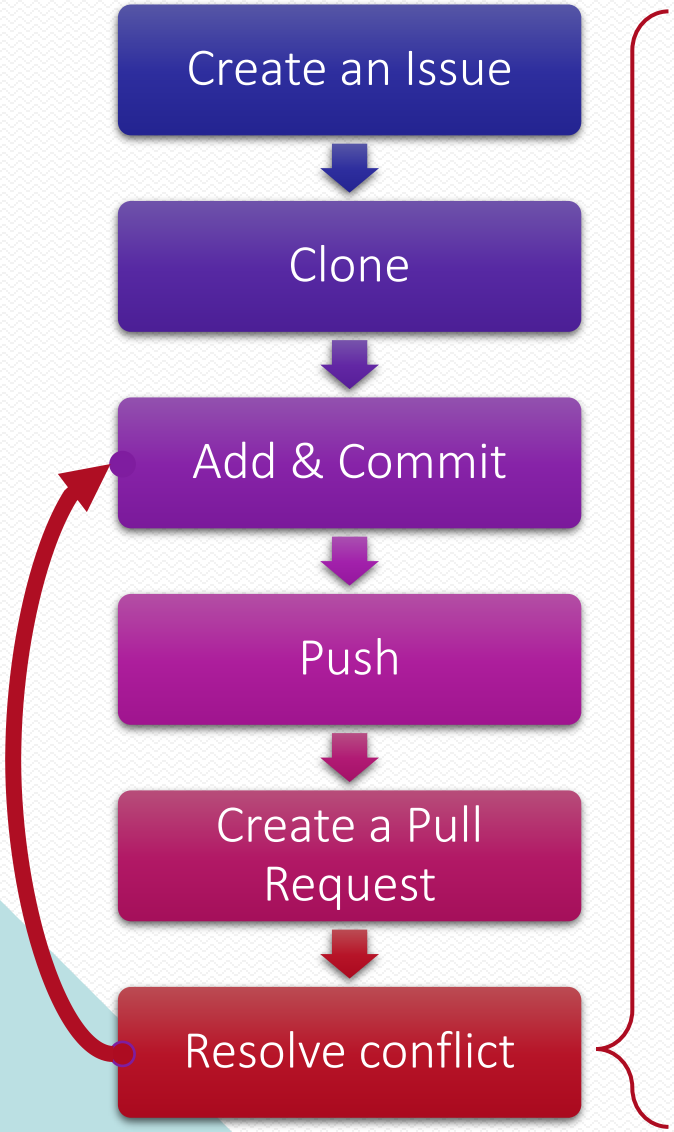
Homework Workflow based Ground Rules



- Resolution #01 : Resolve conflicts in the GitHub



Homework Workflow based Ground Rules



- Resolution #02 : Resolve conflicts in local machine
  - Step 1: Clone the repository or update your local repository with the latest changes.

```
$ git pull origin test1
```

  - Step 2: Switch to the head branch of the pull request.

```
$ git checkout test2
```

  - Step 3: Merge the base branch into the head branch.

```
$ git merge test1
```

  - Step 4: Fix the conflicts and commit the result.
  - Step 5: Push the changes.

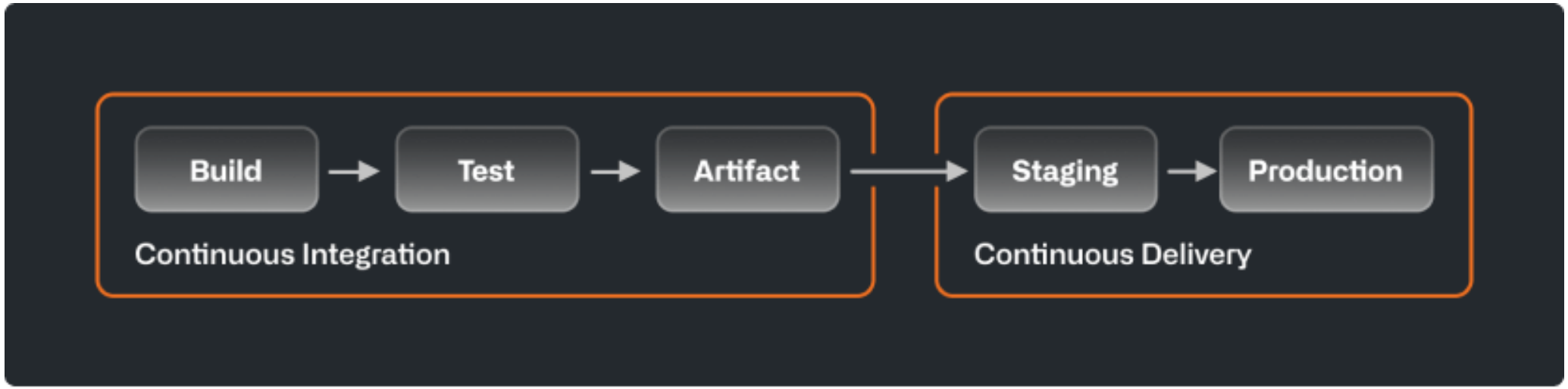
```
$ git push -u origin test2
```

Practice : Homework Workflow

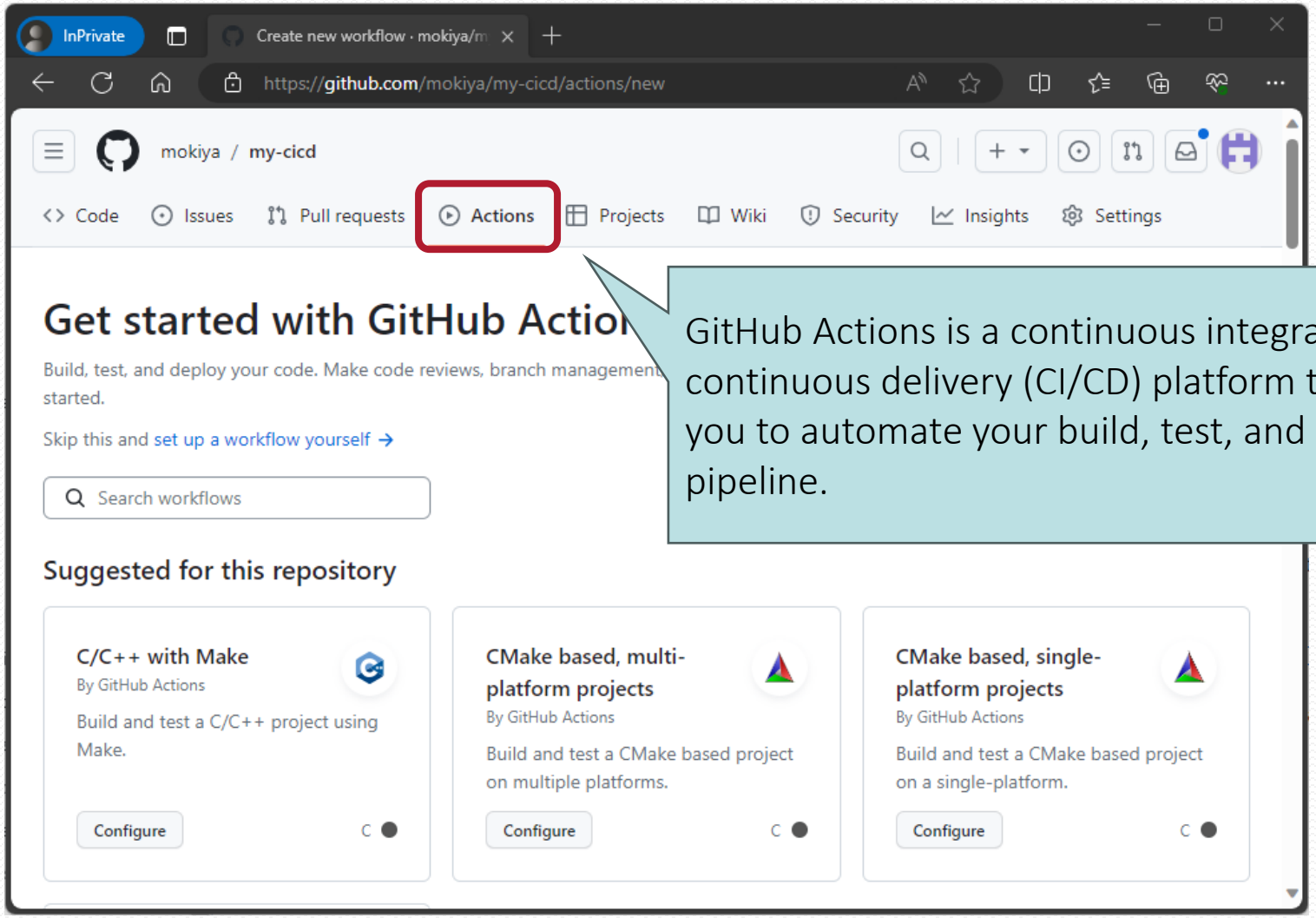
- Goal
  - Enter your name into the table in the two files below.
    - ./class01/README.md
    - ./class02/README.md
  - Be sure to work according to ground rules and create a PR.

CI / CD (Continuous Integration & Delivery)

- CI (Continuous Integration)
  - Automatically builds, tests, and integrates code changes within a shared repositor
- CD (Continuous Delivery)
  - Automatically delivers code changes to production-ready environments for approval, then deploys code changes to customers directly.



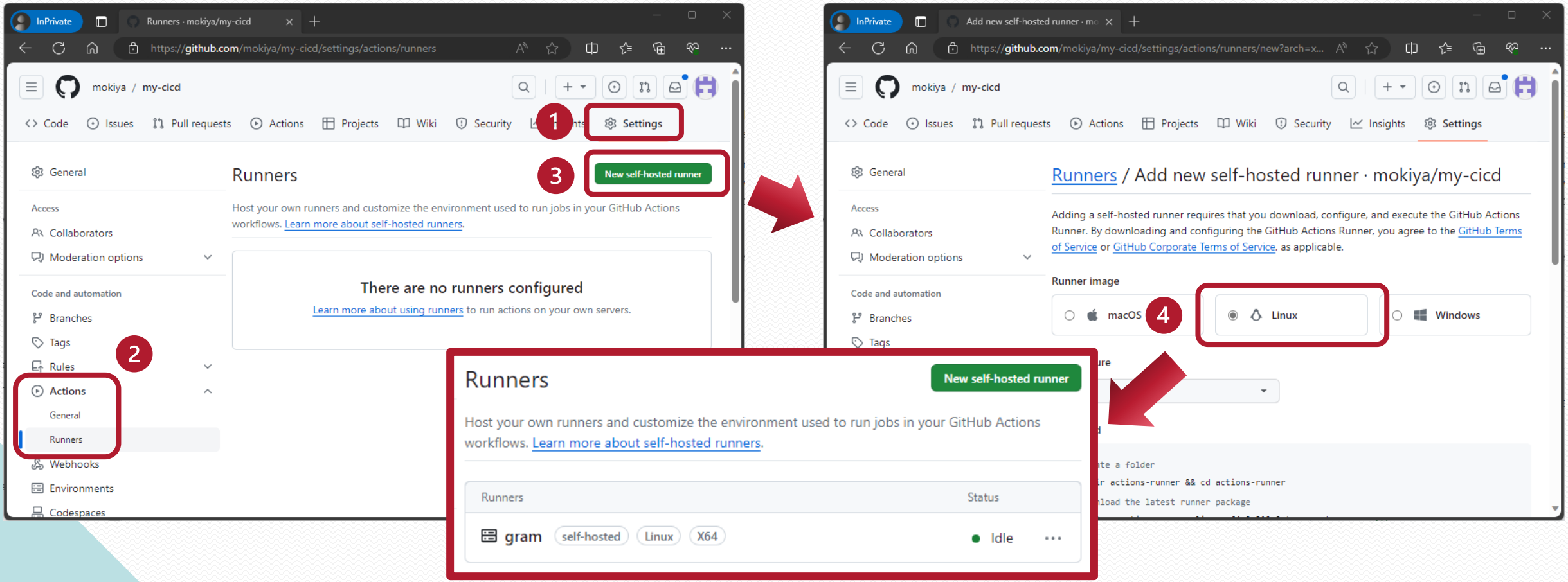
GitHub Actions



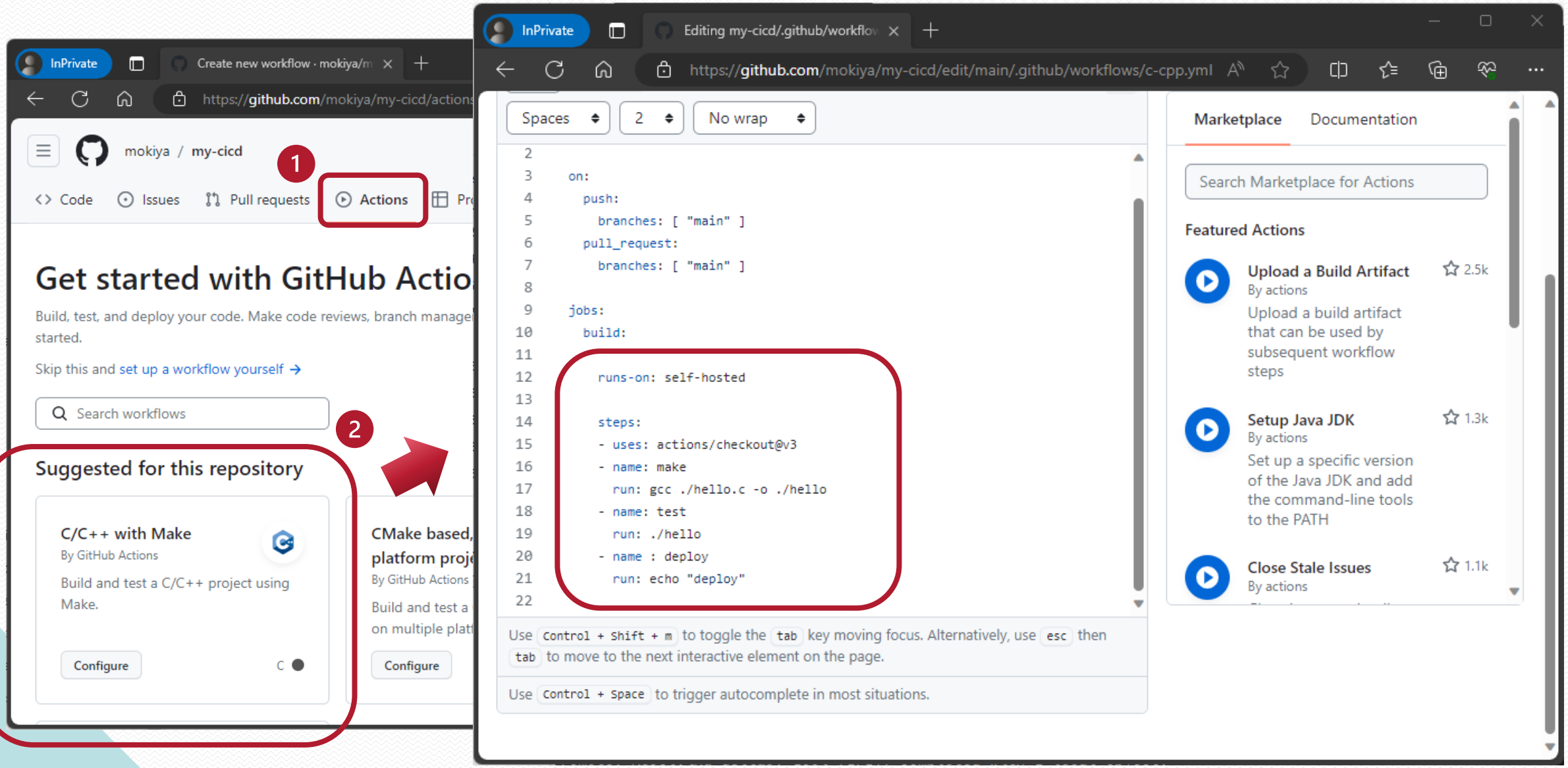
GitHub Actions is a continuous integration and continuous delivery (CI/CD) platform that allows you to automate your build, test, and deployment pipeline.

Self-hosted Runner for GitHub Actions

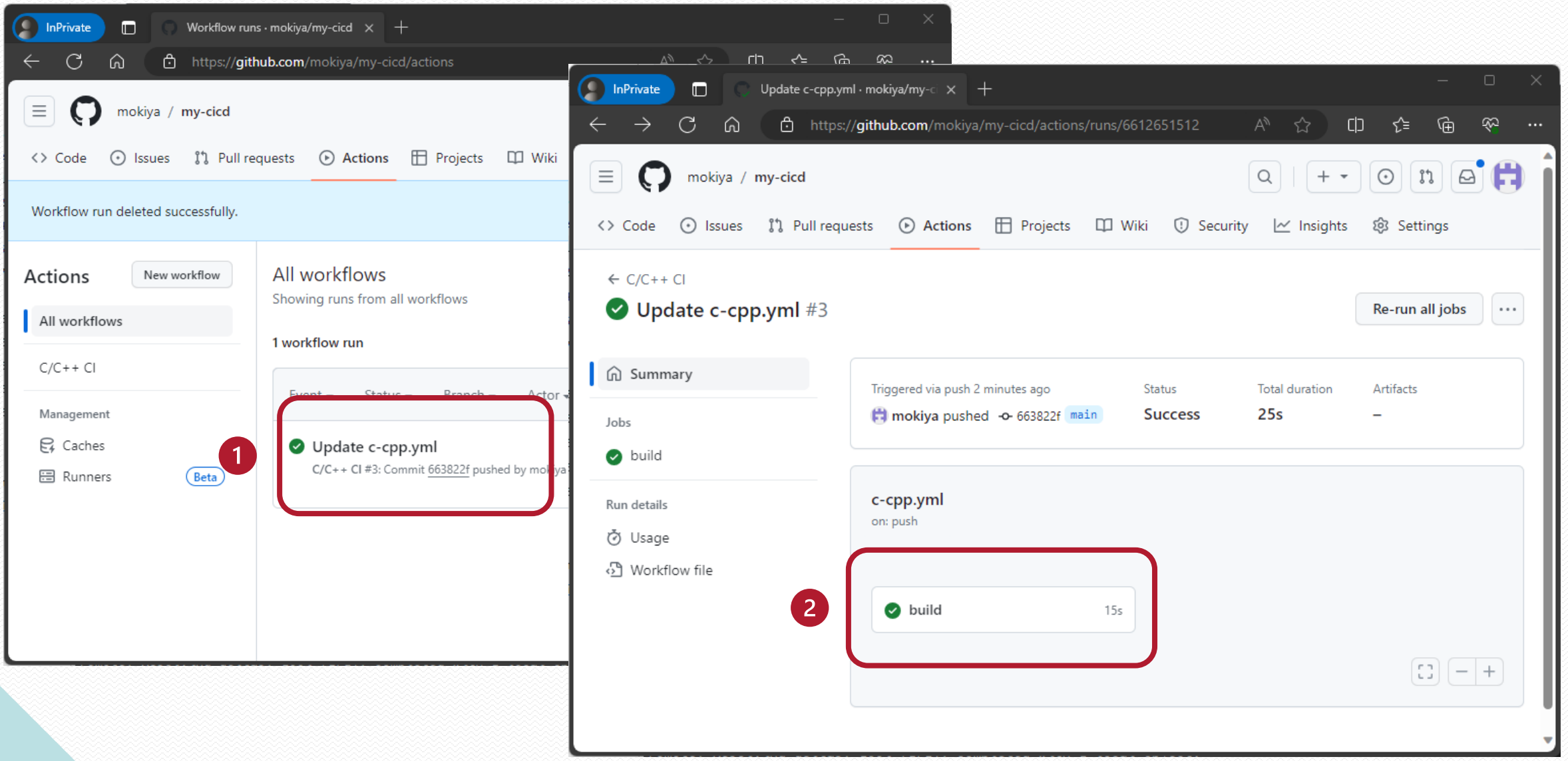
- A self-hosted runner is a system that you deploy and manage to execute jobs from GitHub Actions on GitHub.com



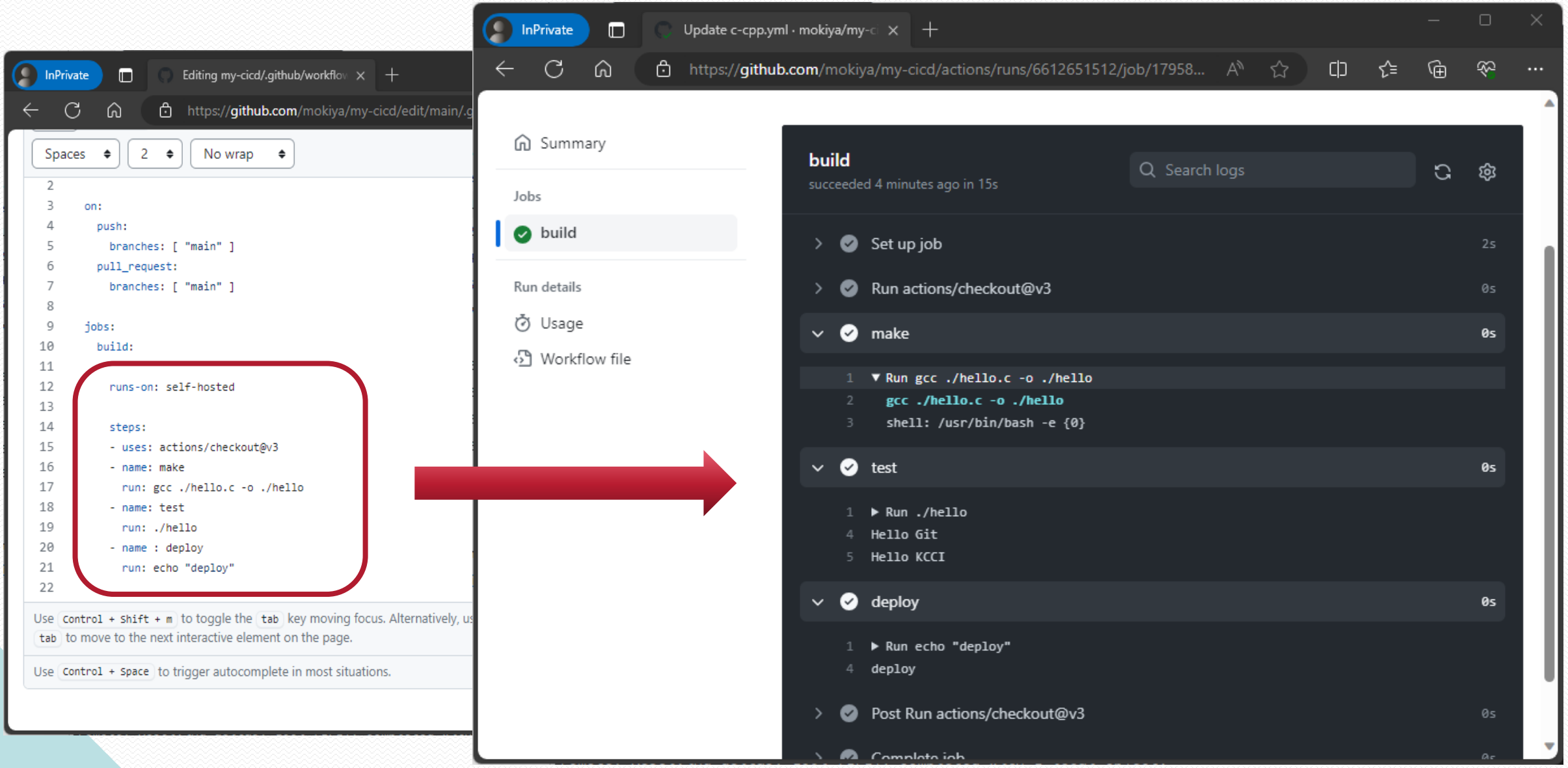
Workflow in GitHub Actions



Demo : CI / CD


InPrivate
Workflow runs · mokiya/my-cicd
https://github.com/mokiya/my-cicd/actions
mokiya / my-cicd
Code
Issues
Pull requests
Actions
Projects
Wiki
Workflow run deleted successfully.
Actions
New workflow
All workflows
C/C++ CI
Management
Caches
Runners
Beta
All workflows
Showing runs from all workflows
1 workflow run
Update c-cpp.yml
C/C++ CI #3: Commit 663822f pushed by mokiya
1
InPrivate
Update c-cpp.yml · mokiya/my-c
https://github.com/mokiya/my-cicd/actions/runs/6612651512
mokiya / my-cicd
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
C/C++ CI
Update c-cpp.yml #3
Re-run all jobs
Summary
Jobs
build
Run details
Usage
Workflow file
Triggered via push 2 minutes ago
mokiya pushed 663822f main
Status
Success
Total duration
25s
Artifacts
–
c-cpp.yml
on: push
build
15s
2

Demo : CI / CD


InPrivate
Editing my-cicd/.github/workflow
https://github.com/mokiya/my-cicd/edit/main/
Spaces
2
No wrap
on:
push:
branches: ["main"]
pull_request:
branches: ["main"]
jobs:
build:
runs-on: self-hosted
steps:
- uses: actions/checkout@v3
- name: make
run: gcc ./hello.c -o ./hello
- name: test
run: ./hello
- name : deploy
run: echo "deploy"
Use Control + Shift + m to toggle the tab key moving focus. Alternatively, us
tab to move to the next interactive element on the page.
Use Control + Space to trigger autocomplete in most situations.
InPrivate
Update c-cpp.yml · mokiya/my-c
https://github.com/mokiya/my-cicd/actions/runs/6612651512/job/17958...
Summary
Jobs
build
Run details
Usage
Workflow file
build
succeeded 4 minutes ago in 15s
Search logs
Set up job
2s
Run actions/checkout@v3
0s
make
0s
Run gcc ./hello.c -o ./hello
gcc ./hello.c -o ./hello
shell: /usr/bin/bash -e {0}
test
0s
Run ./hello
Hello Git
Hello KCCI
deploy
0s
Run echo "deploy"
deploy
Post Run actions/checkout@v3
0s

THANK YOU