



WEEK 1

SOFTWARE DEVELOPMENT TOOLS AND ENVIRONMENTS

UNLIMITED

Presented by

Asst. Prof. Dr. Tuchsanai Ploysuwan



 DevTools Public

Unpin Unwatch 1

main 2 branches 0 tags

Go to file Add file Code

Your main branch isn't protected
Protect this branch

Tuchsanai Create README.md bb20146 now 121 commits

File / Commit	Author / Message	Date
Docker	ff	3 weeks ago
GIT	Update Week1.pdf	13 hours ago
Jenkins	j	last month
kubernetes	Create README.md	now
.gitignore	Update .gitignore	3 months ago
README.md	d	13 hours ago

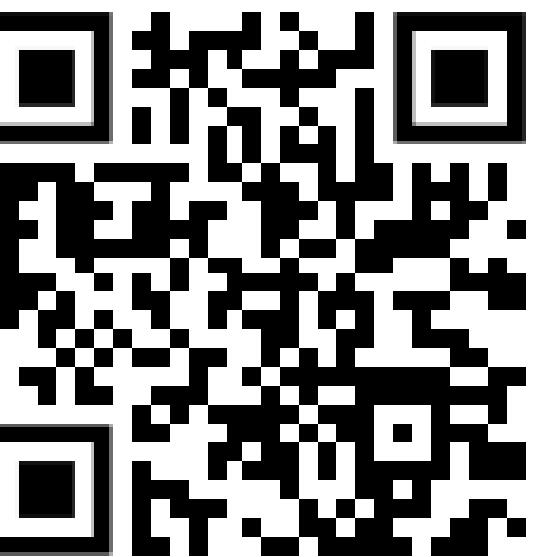
☰ README.md

SOFTWARE-DEVELOPMENT-TOOLS-AND-ENVIRONMENTS

เนื้อหาตามหลักสูตร :

หลักการเพื่อเป็นผู้เชี่ยวชาญด้านซอฟต์แวร์ บทบาทของแอพพลิเคชันในงานด้านวิศวกรรมซอฟต์แวร์ เครื่องมือการพัฒนาซอฟต์แวร์แบบօเจล์ การติดตามความคืบหน้าของการพัฒนาผลิตภัณฑ์ การจัดการเวอร์ชันและการกำหนดค่า เครื่องสำหรับสร้างและการบูรณาการอย่างต่อเนื่อง เครื่องสำหรับแก้จุดบกพร่องและการรวมข้อมูลเชิงประสิทธิภาพของโปรแกรม สภาพแวดล้อมแบบร่วมนิโอพัฒนา เครื่องมือสำหรับการควบรวมรวมและติดตั้ง

Principles to Software Professionals, Roles of Applications in Software Engineering Tasks, Agile Software Development Tools, Product Development Tracking, Version and Configuration Management, Build and Continuous Integration Tools, Program Debugging and Profiling Tools, Collaborative Development Environments, Packaging and Deployment



Document



กลุ่ม facebook

การให้คะแนน

การให้คะแนน	
Midterm	35
Final	35
Homework and Exercise LAB	15
Mini project	15

แผนการสอน	
สัปดาห์ที่	หัวข้อ
1	<p>Git and GitHub Configure Git Creating and Cloning Repositories Private Repositories and Token</p> <p>Lab week 1</p>
2	<p>Understanding Git Usage and Workflow Add and Commit Git Log Git Remote and Git Push Fetch and Pull</p> <p>Lab week 2</p>
3	<p>Understanding Branches Understanding HEAD Git Branch Commands Delete and Rename Branches Merging Branches - Theory and Concepts Merging Branches in Practice Git Diff</p> <p>Lab week 3</p>
4	<p>Git with Going back and Undoing Changes -Git Checkout and Detached HEAD -Git Restore, Git Reset, Git Revert Undoing Changes - Exercise and Solution</p>

5	<p>Docker</p> <p>Docker Overview Basic Docker Commands Docker Run</p> <p>Docker Images Environment Variables Command vs Entrypoint</p>
6	<p>Labs - Docker Images Labs - Environment Variables Labs - Command vs Entrypoint</p> <p>Docker Compose Docker Registry</p> <p>Lab: Docker Registry Labs: Docker Compose</p>
7	<p>Docker Engine Docker Storage Docker Networking</p> <p>Labs - Docker Storage Labs - Docker Networking</p>
8	<p>Docker Swarm Docker Service Docker Stacks CI/CD - Docker Integration</p> <p>Lab</p>

5	<p>Docker</p> <p>Docker Overview</p> <p>Basic Docker Commands</p> <p>Docker Run</p> <p>Docker Images</p> <p>Environment Variables</p> <p>Command vs Entrypoint</p> <p>Labs - Docker Images</p> <p>Labs - Environment Variables</p> <p>Labs - Command vs Entrypoint</p>	9	<p>Kubernetes 1</p> <p>Container Orchestration</p> <p>Kubernetes Architecture</p> <p>PODs</p>
6	<p>Docker Compose</p> <p>Docker Registry</p> <p>Lab: Docker Registry</p> <p>Labs: Docker Compose</p>	10	<p>Kubernetes 2</p> <p>Basics of Networking in Kubernetes</p> <p>ReplicaSets and Deployments</p>
7	<p>Docker Engine</p> <p>Docker Storage</p> <p>Docker Networking</p> <p>Labs - Docker Storage</p> <p>Labs - Docker Networking</p>	11	<p>Micro service</p>
8	<p>Docker Swarm</p> <p>Docker Service</p> <p>Docker Stacks</p> <p>CI/CD - Docker Integration</p> <p>Lab</p>	12	<p>Jenkins</p>
		13	<p>Mini Project Pitching</p>
		14	<p>Mini Project Pitching</p>
		15	<p>Mini Project Pitching</p>
		16	<p>Mini Project Pitching</p>

- Git was developed in 2005 by **Linus Torvalds**
- Git is **Version control system** is a system that records changes to a file or set file over time so that you. can restore specific version later
- Git is a **Distributed Version Control System**







Git – What and Why

Oh boy, I sure do
love playing my
video games!



I'm going to save
my game now in
case I die soon!



Oh jeez, this is
going to be a
difficult fight!



ughhhh I died!





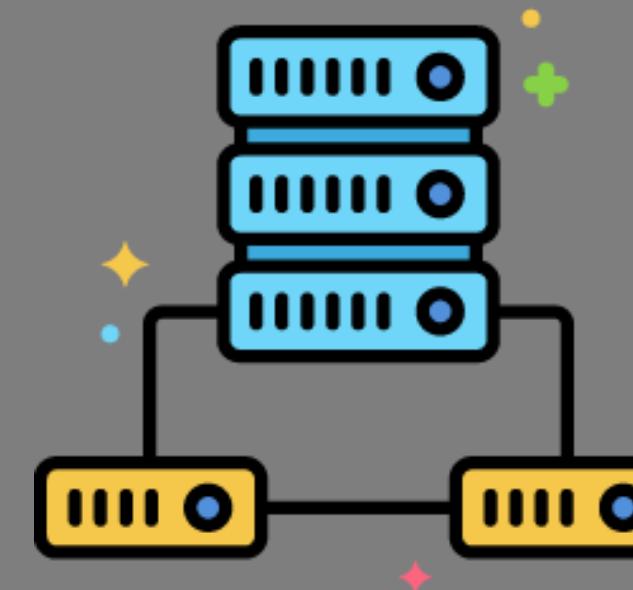
Thank heavens I
saved my game! I
can just revert!



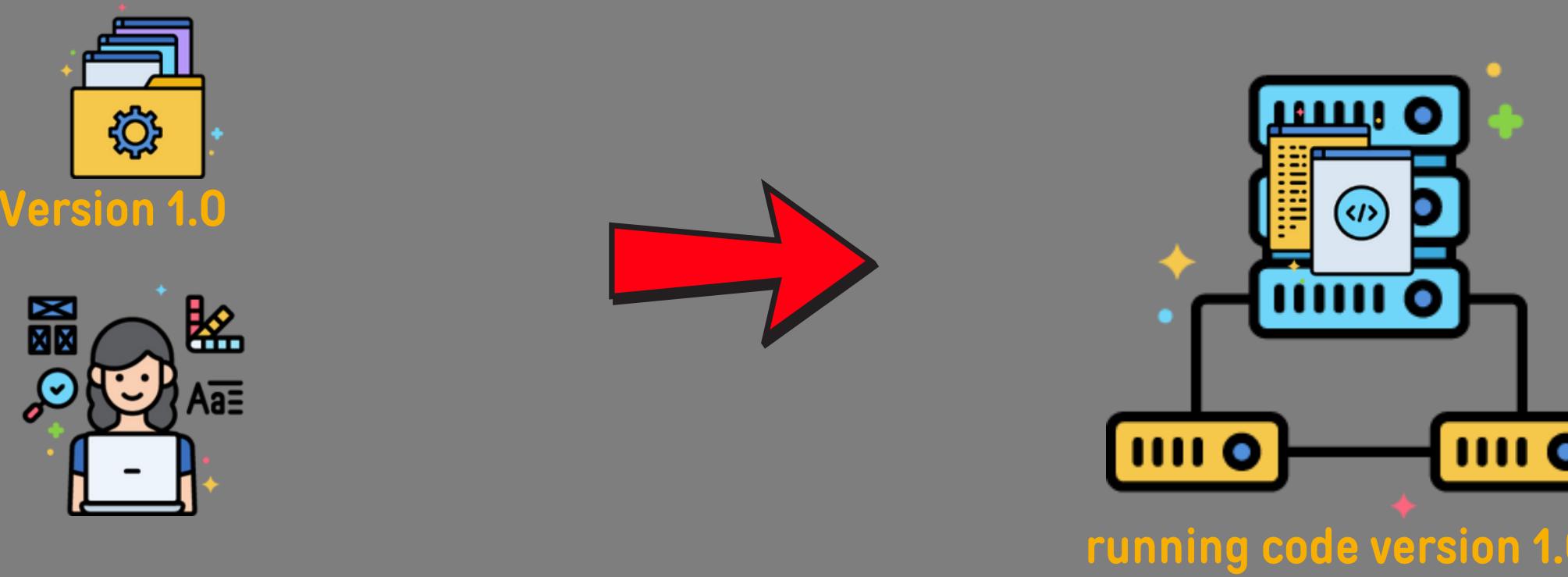
Before Version Control System



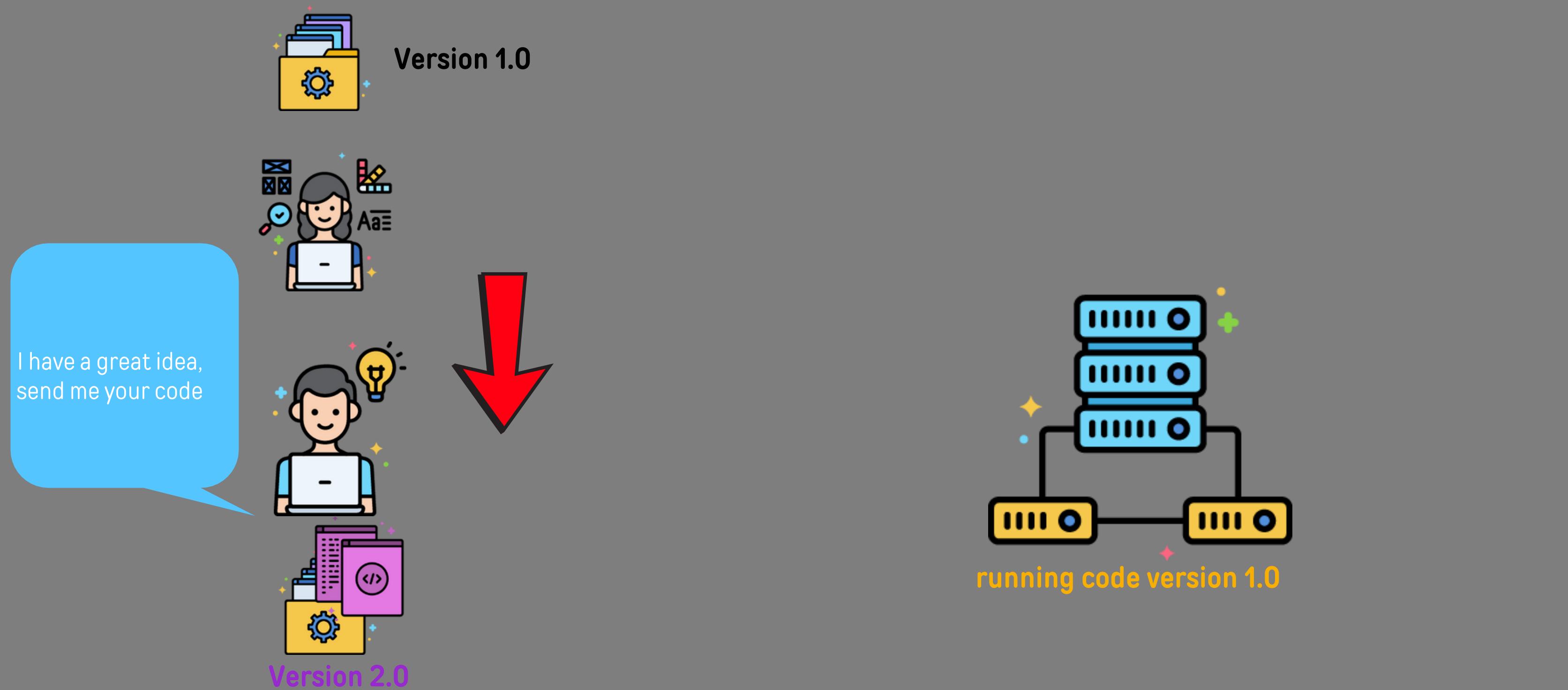
Before Version Control System



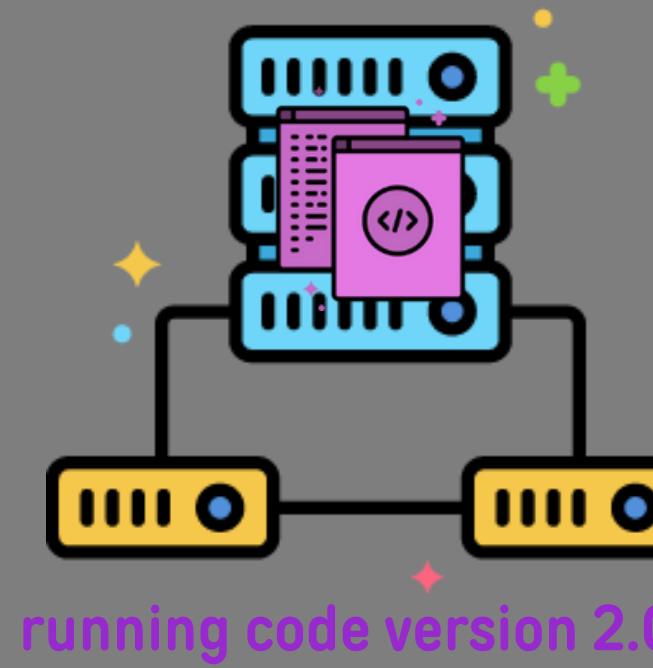
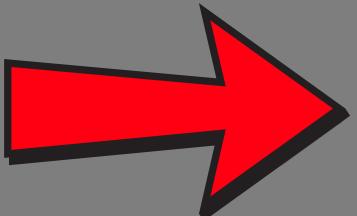
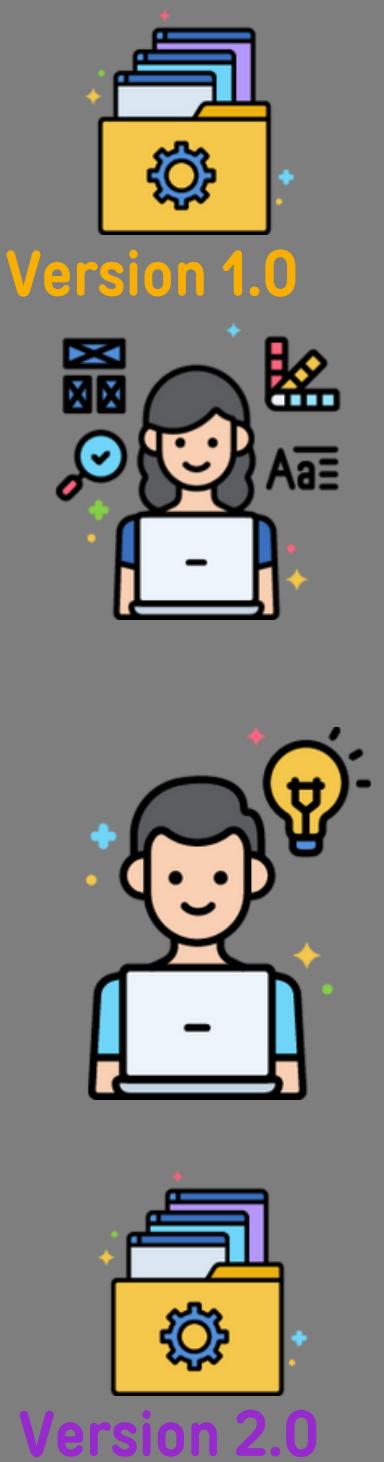
Before Version Control System



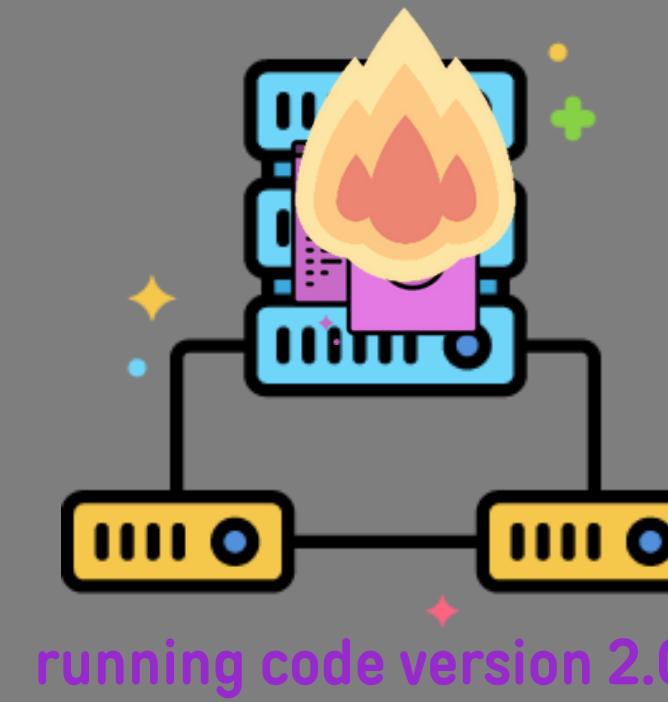
Before Version Control System



Before Version Control System



Before Version Control System



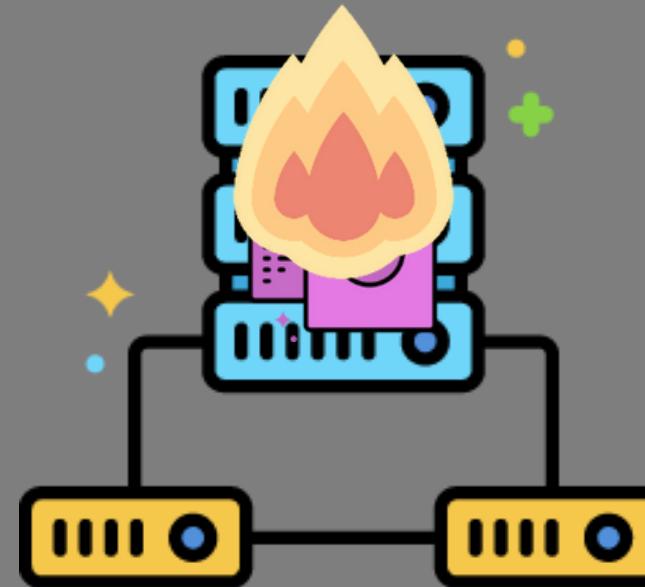
Before Version Control System



Version 1.0



Version 2.0



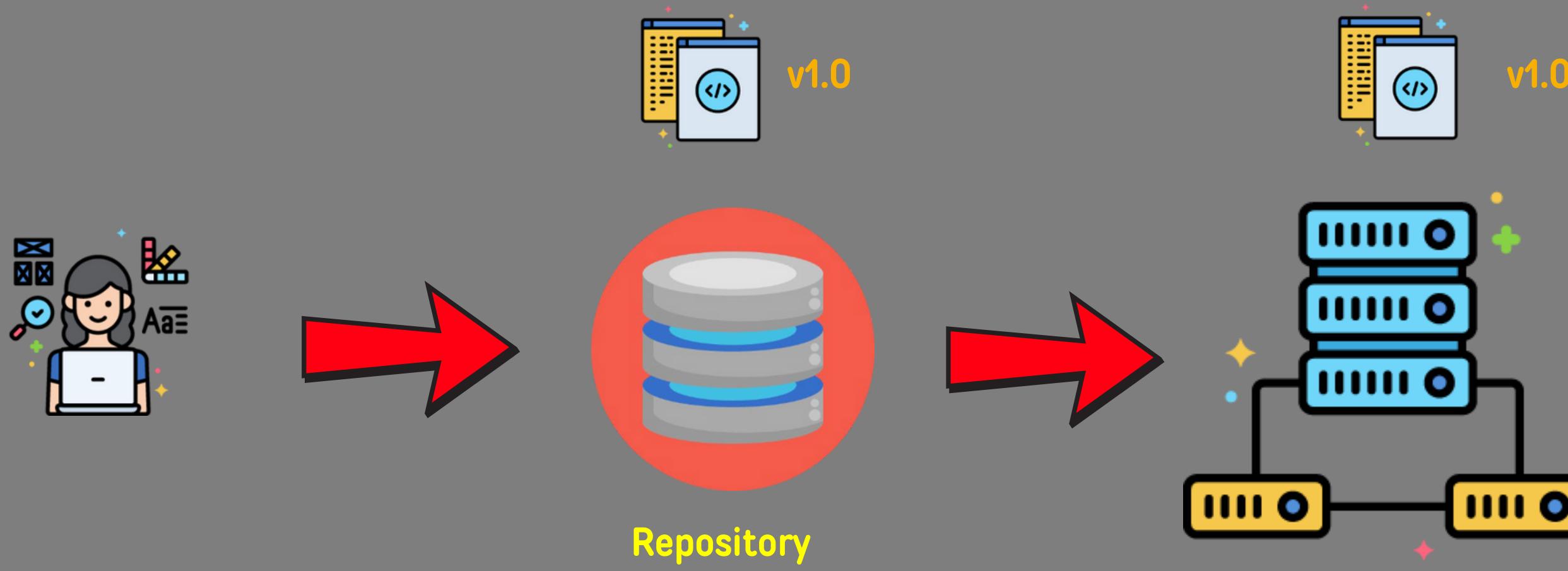
running code version 2.0

- Rollback is time consuming
- No audit tracking
- Not scalable for large teams

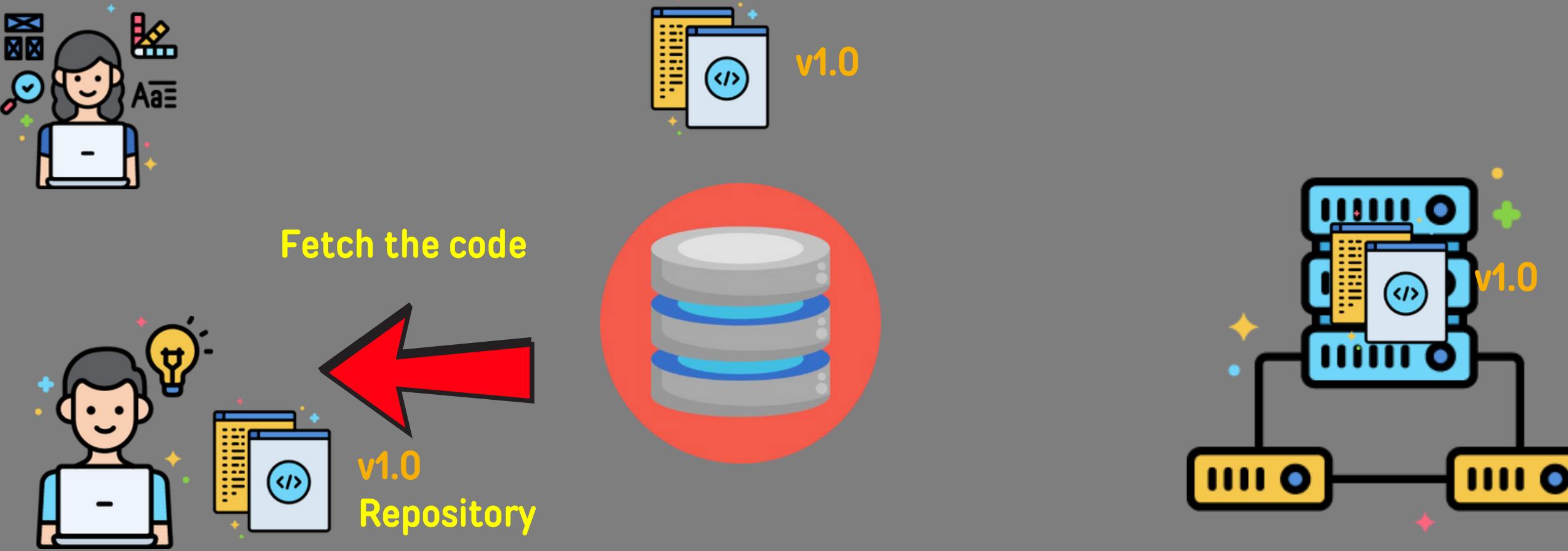
Version Control System



Version Control System



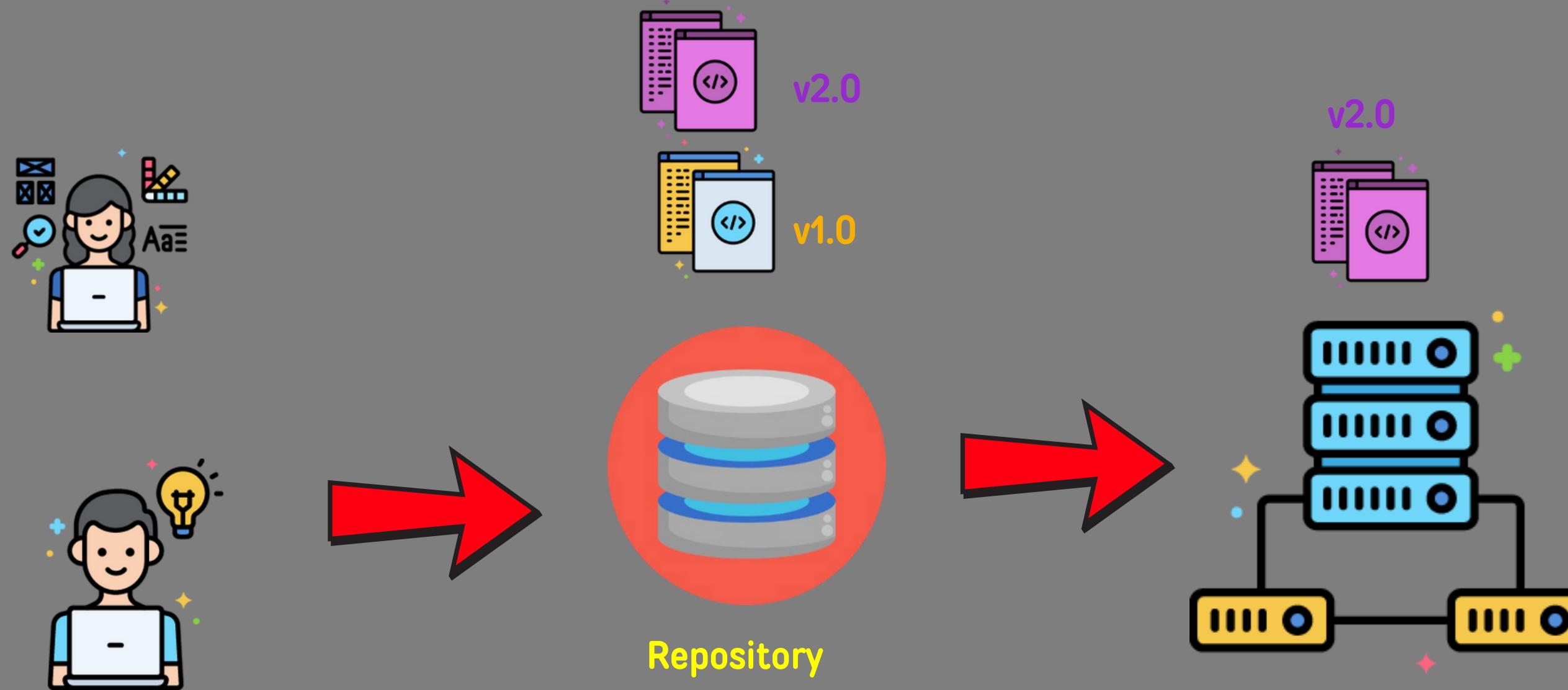
Version Control System



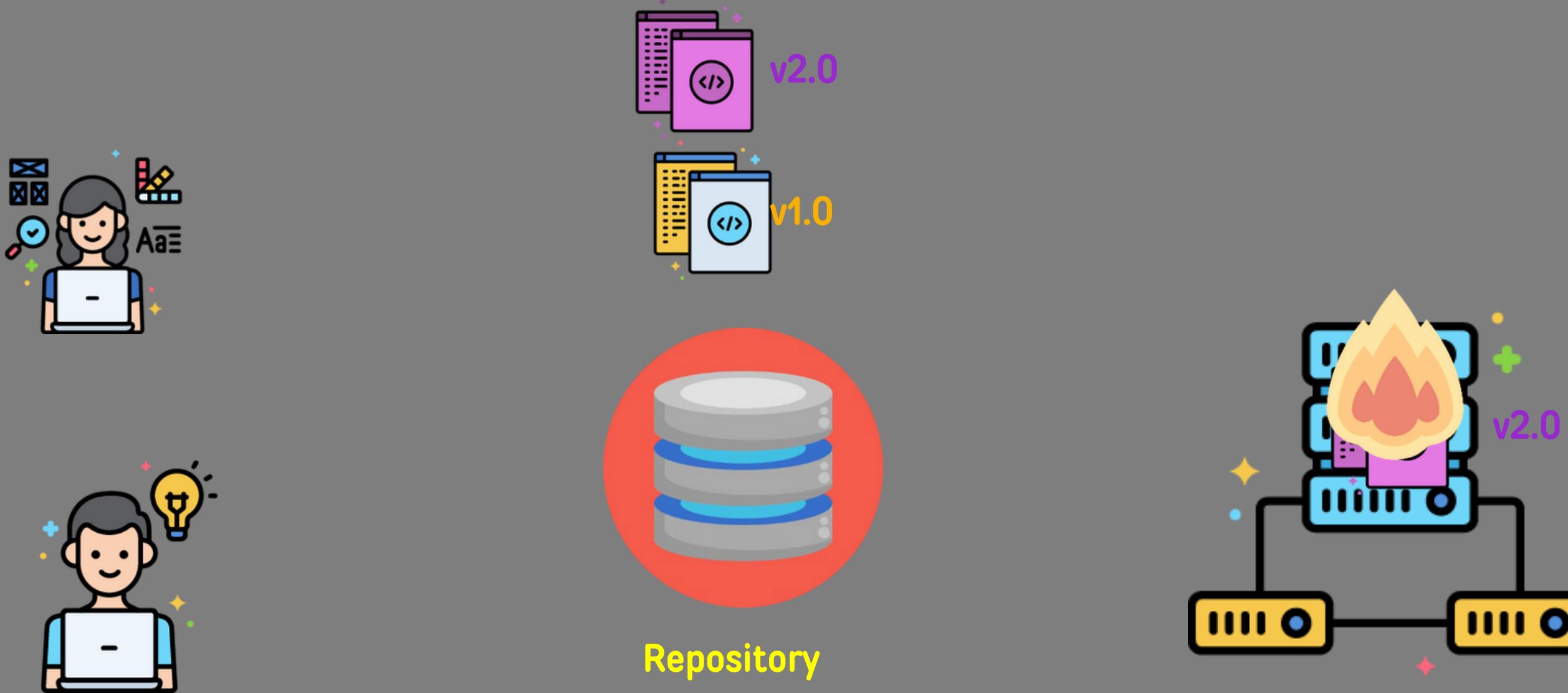
Version Control System



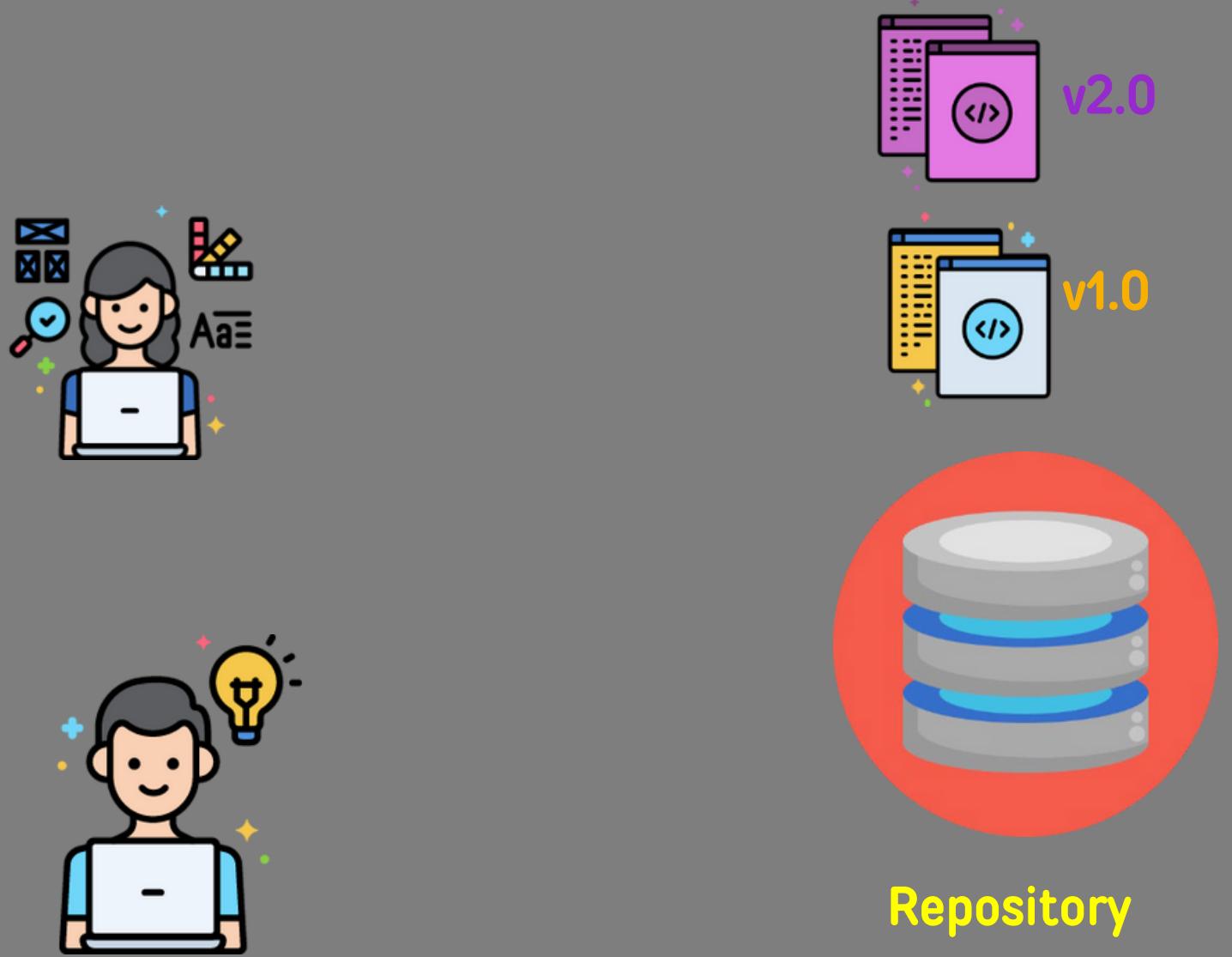
Version Control System



Version Control System

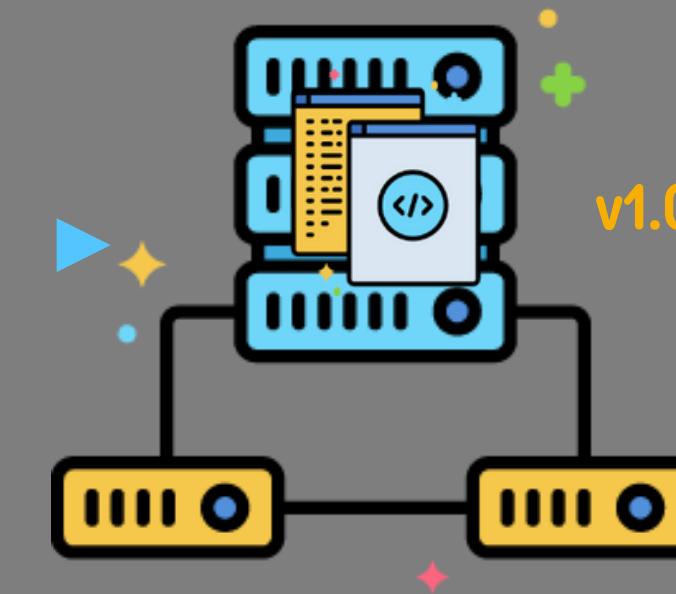


Version Control System

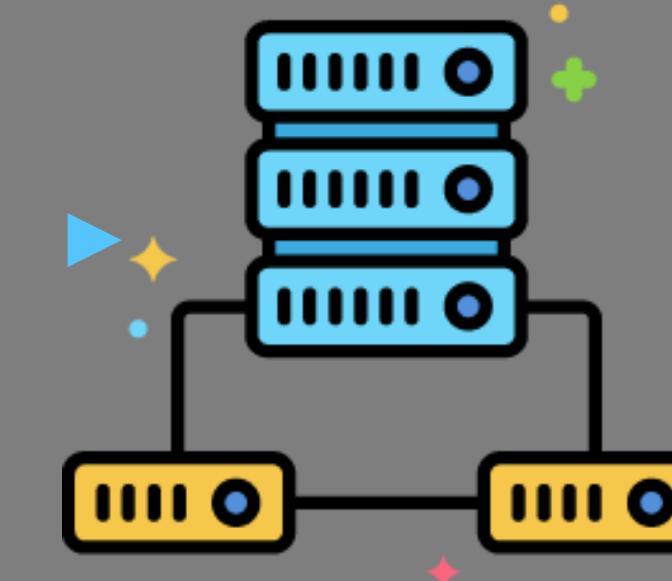
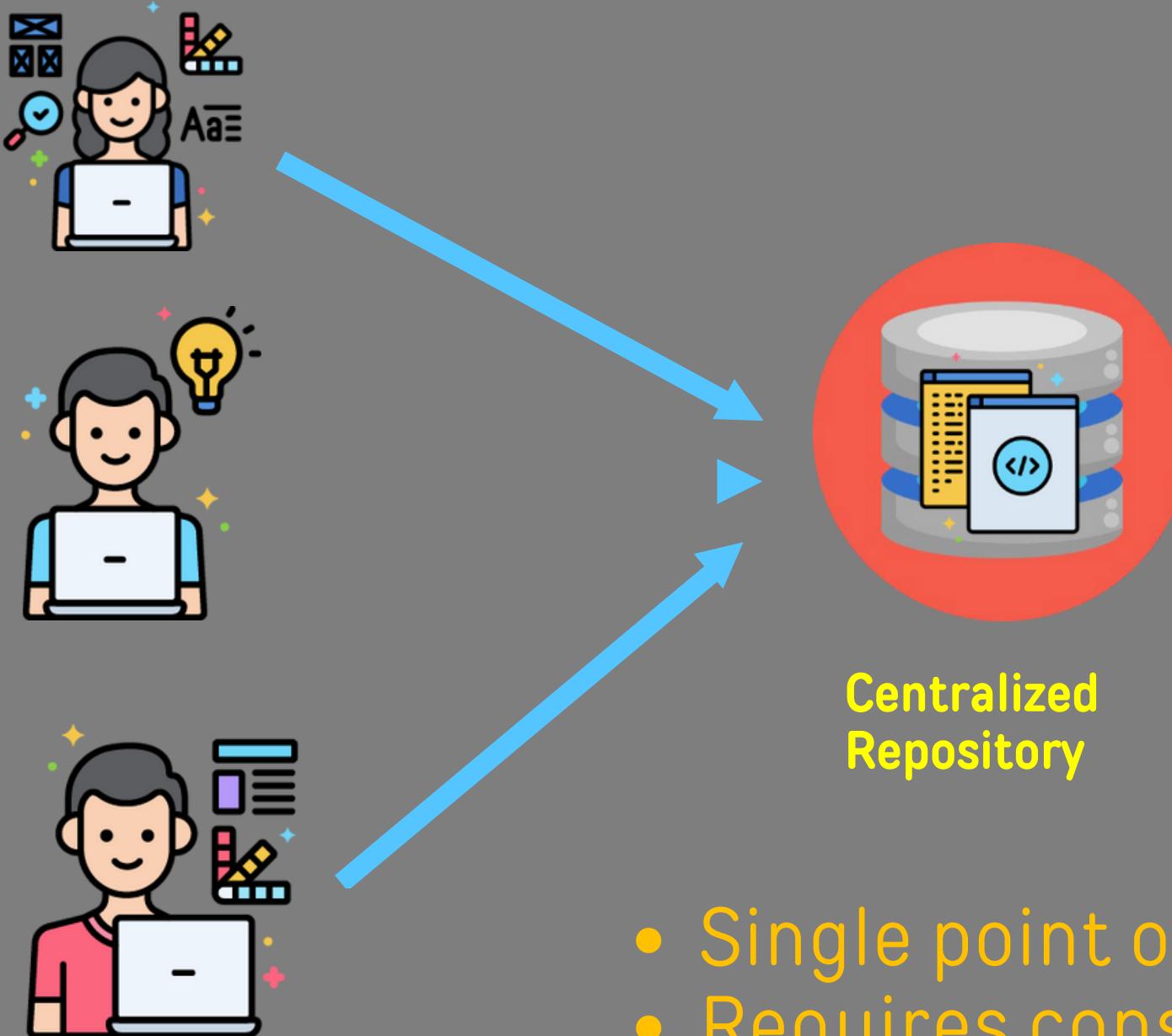


Why Git?

- Distributed

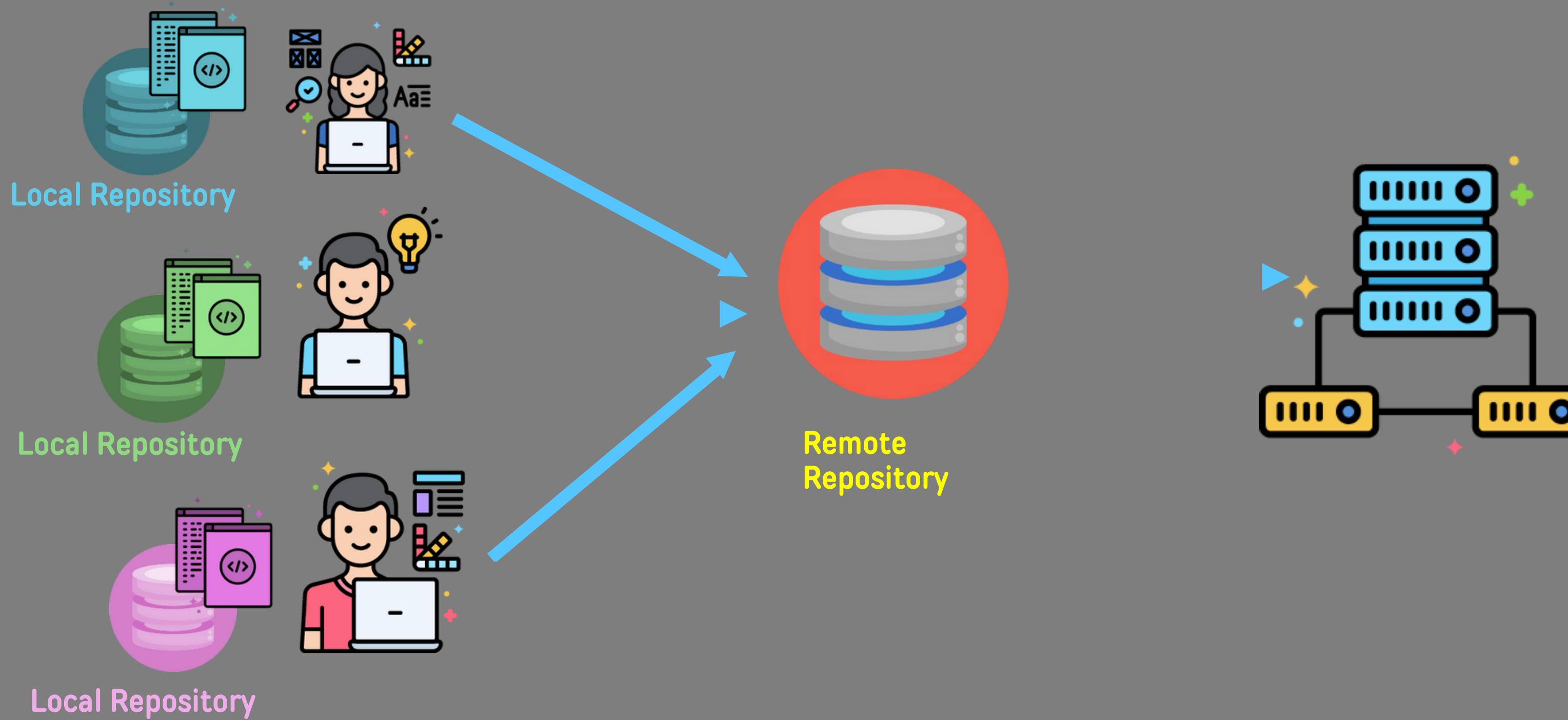


Centralized Version Control System



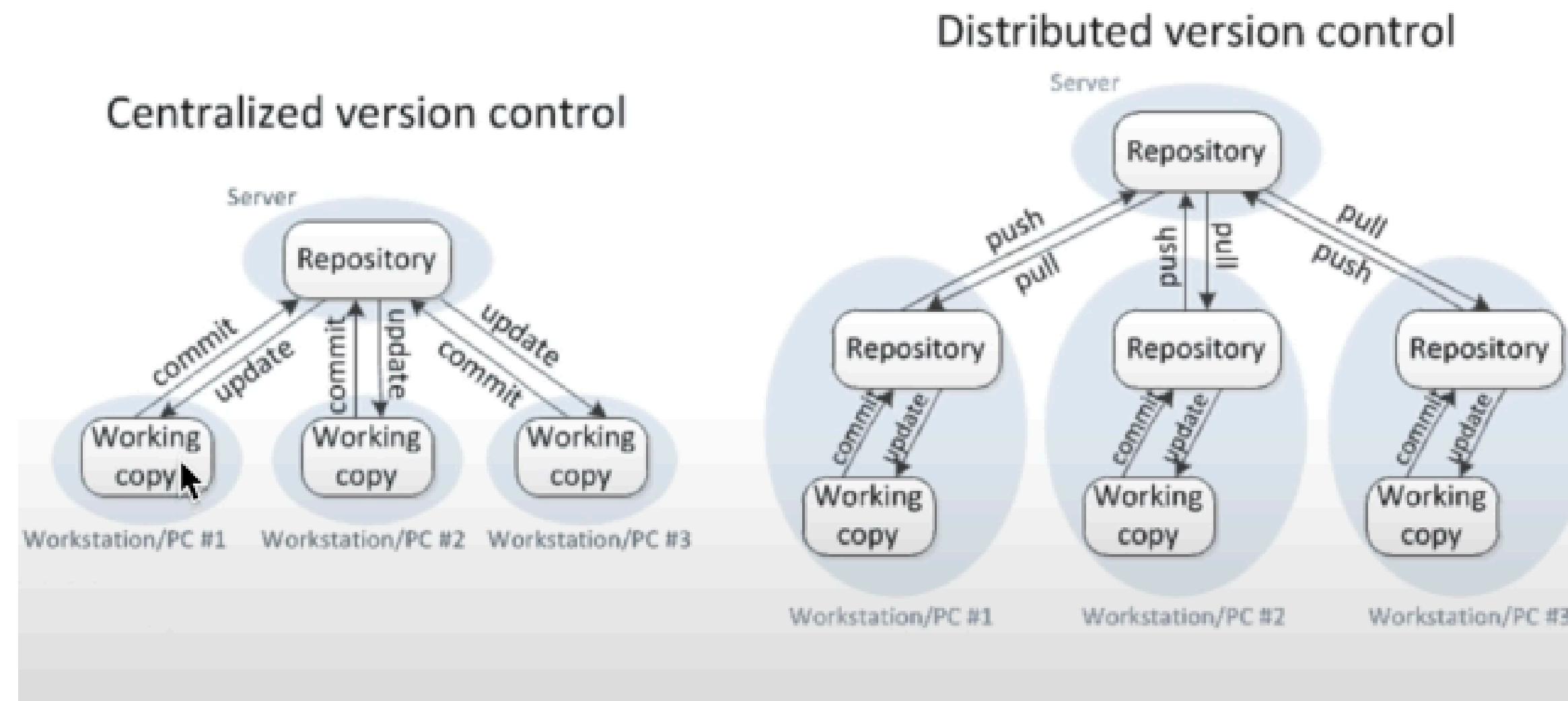
- Single point of failure
- Requires constant connectivity
- E.G – Subversion, Endevor

Distributed Version Control System

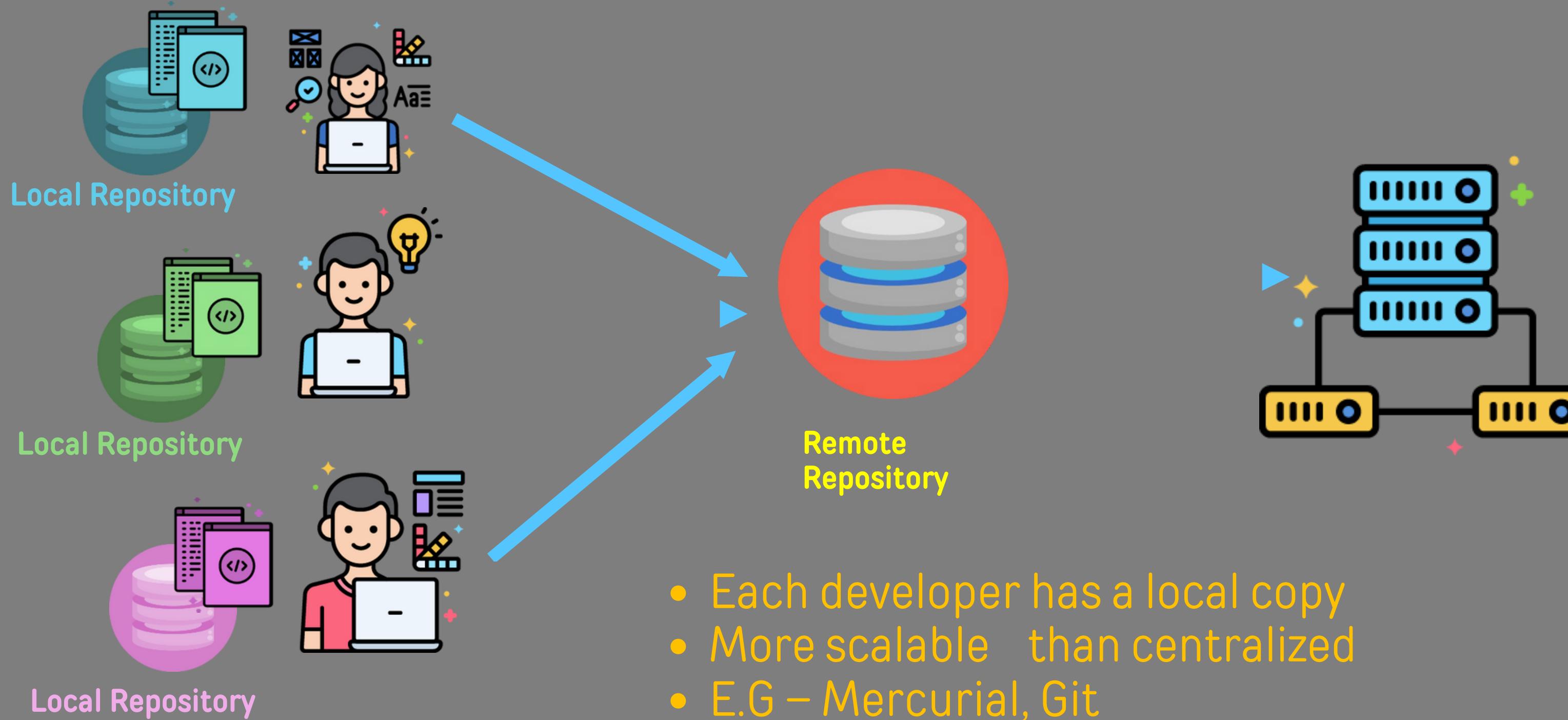


Centralized vs Distributed

Version Control



Distributed Version Control System



Version Control System

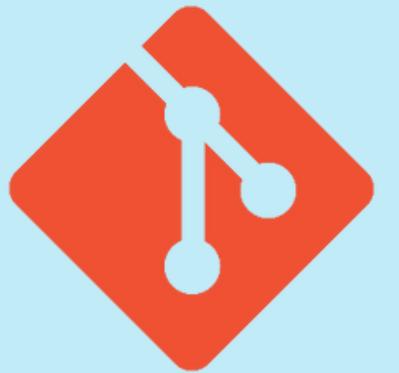


Why Git?

- Distributed
- Performant
- Detailed audit tracking
- Open source
 - Free!
 - Implemented with Kubernetes GitOps, integration with Jenkins and other DevOps tools
 - GitHub, GitLab, Code Commit are all based on Git

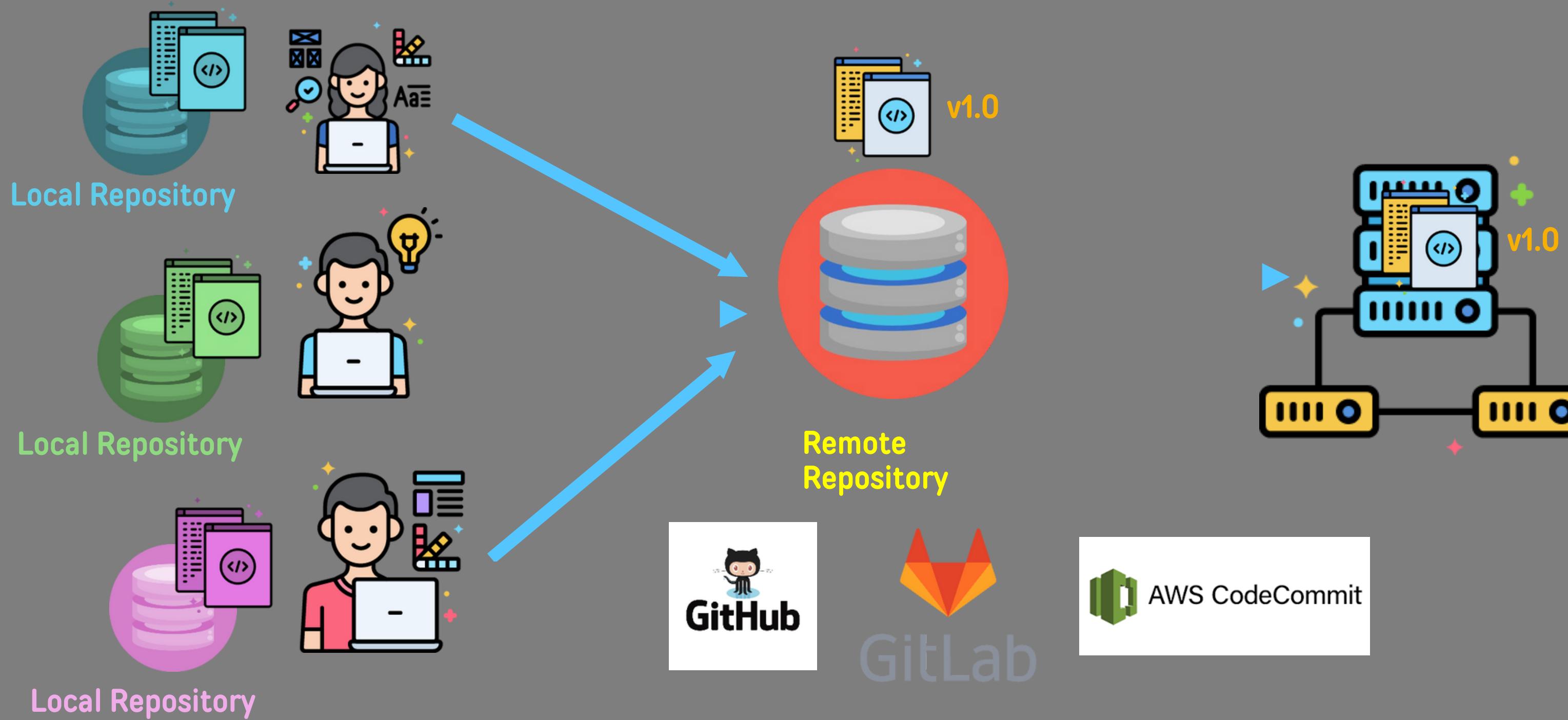
Git vs GitHub

Git Vs. GitHub



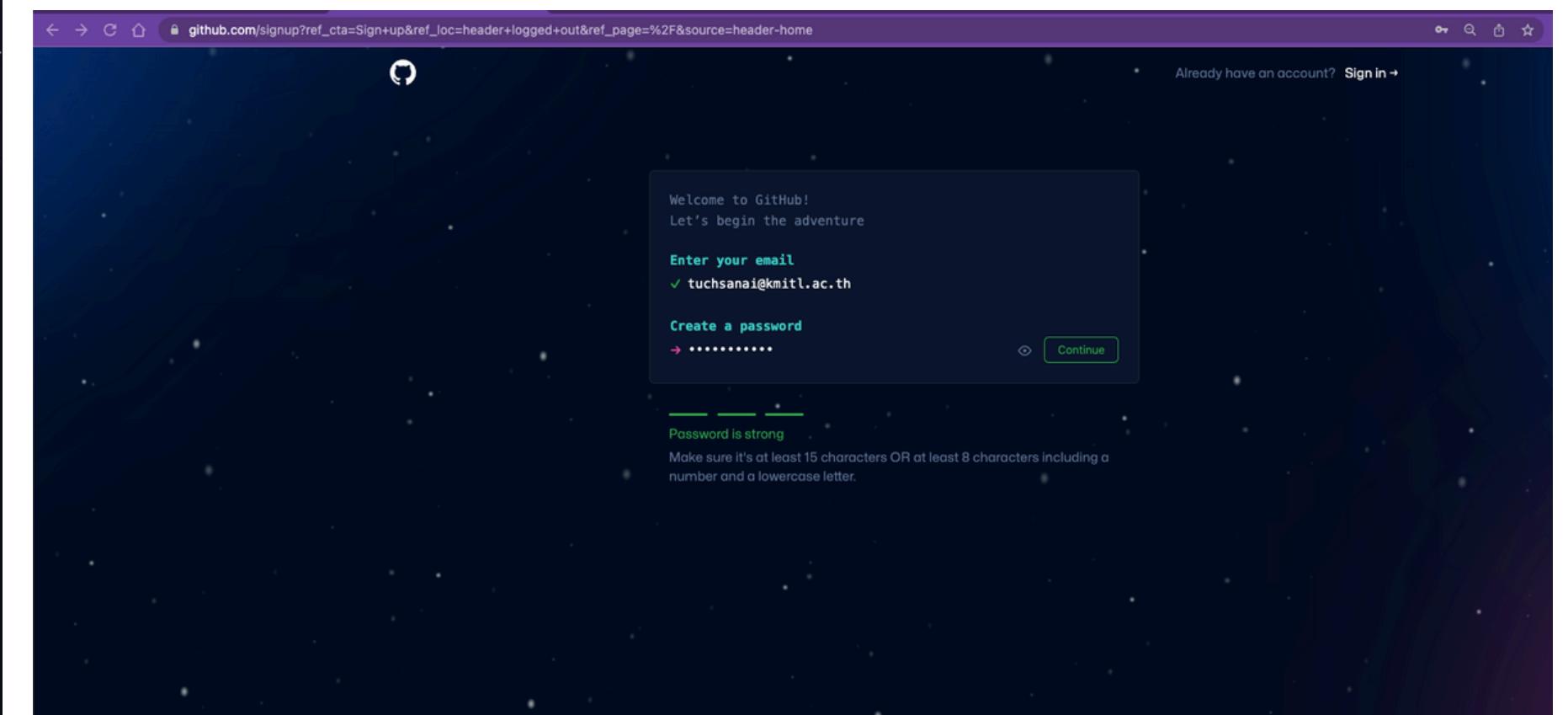
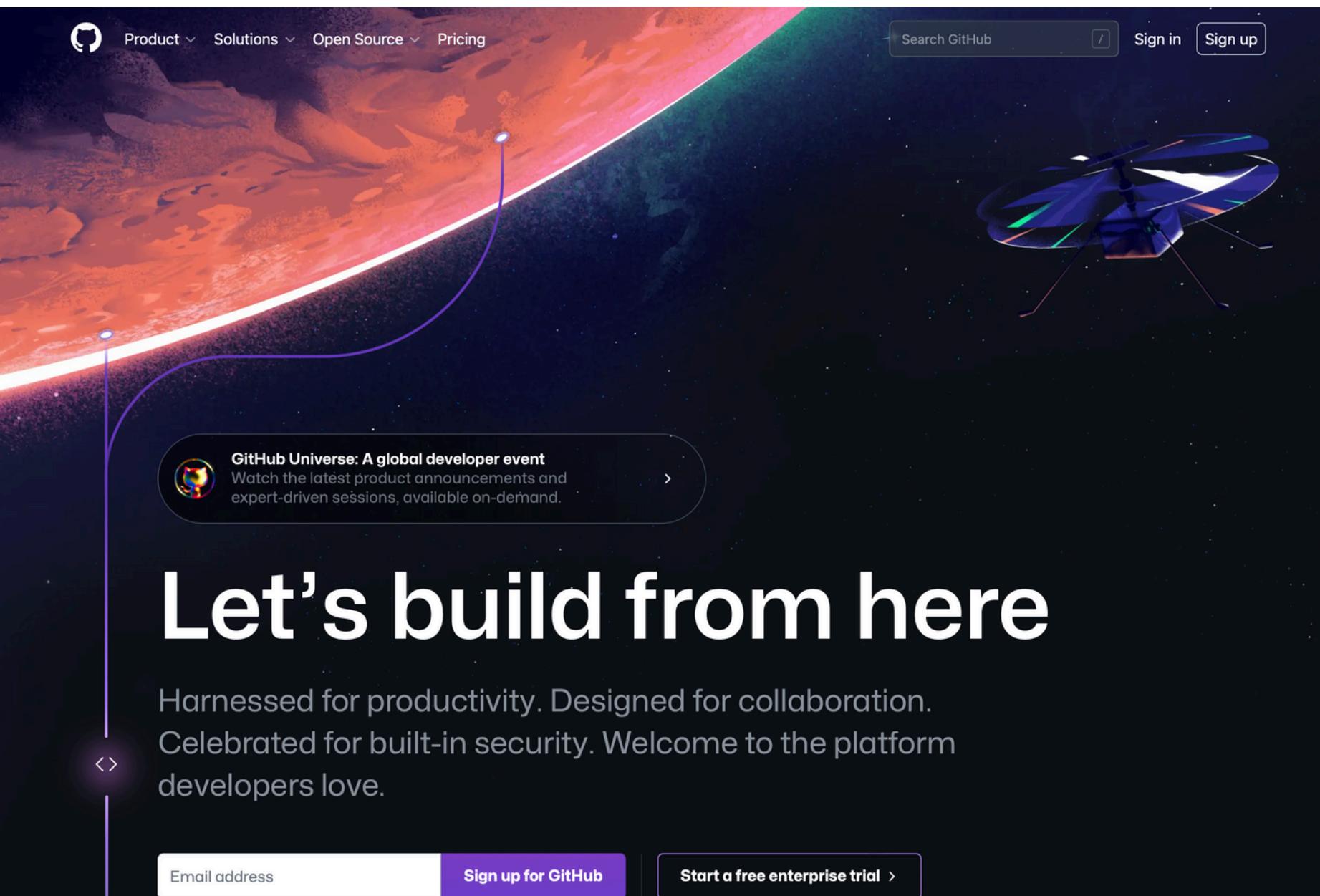
- Version Control System
- Installed locally on the system
- Created in 2005, by Linus Torvalds
- Open source, and used in multiple cloud repository services
- Git repository hosting services with other features
- Runs on the cloud
- Created in 2008, currently owned by Microsoft
- Not open source, have free and paid tiers

Distributed Version Control System



Week 1 - Starting with Git

Sign Up : <https://github.com>



<https://github.com>

Installing Git

Week 1 - Starting with Git

- **MacOS or Linux Users:**
 - Congrats! You already have Git installed on your machine since it comes pre-installed as part of your OS.
 - To confirm this, open up a terminal and type:
 - **git --version**
 - **>> git version 2.25.1 (Apple Git-128)**

Week 1 - Starting with Git

- **MacOS or Linux Users:**
 - If you wish to update or re-install git, you can do this by simply selecting the MacOS or Linux links on the official git website:
 - **<https://git-scm.com/downloads>**

Week 1 - Starting with Git

- **MacOS or Linux Users:**
 - Our suggested text editor for this course is VS Code:
 - <https://code.visualstudio.com/>
 - Its created by Microsoft and has direct integrations with GitHub and is one of the most popular text editors today.
 - You can follow along with any text editor you prefer however.



Week 1 - Starting with Git

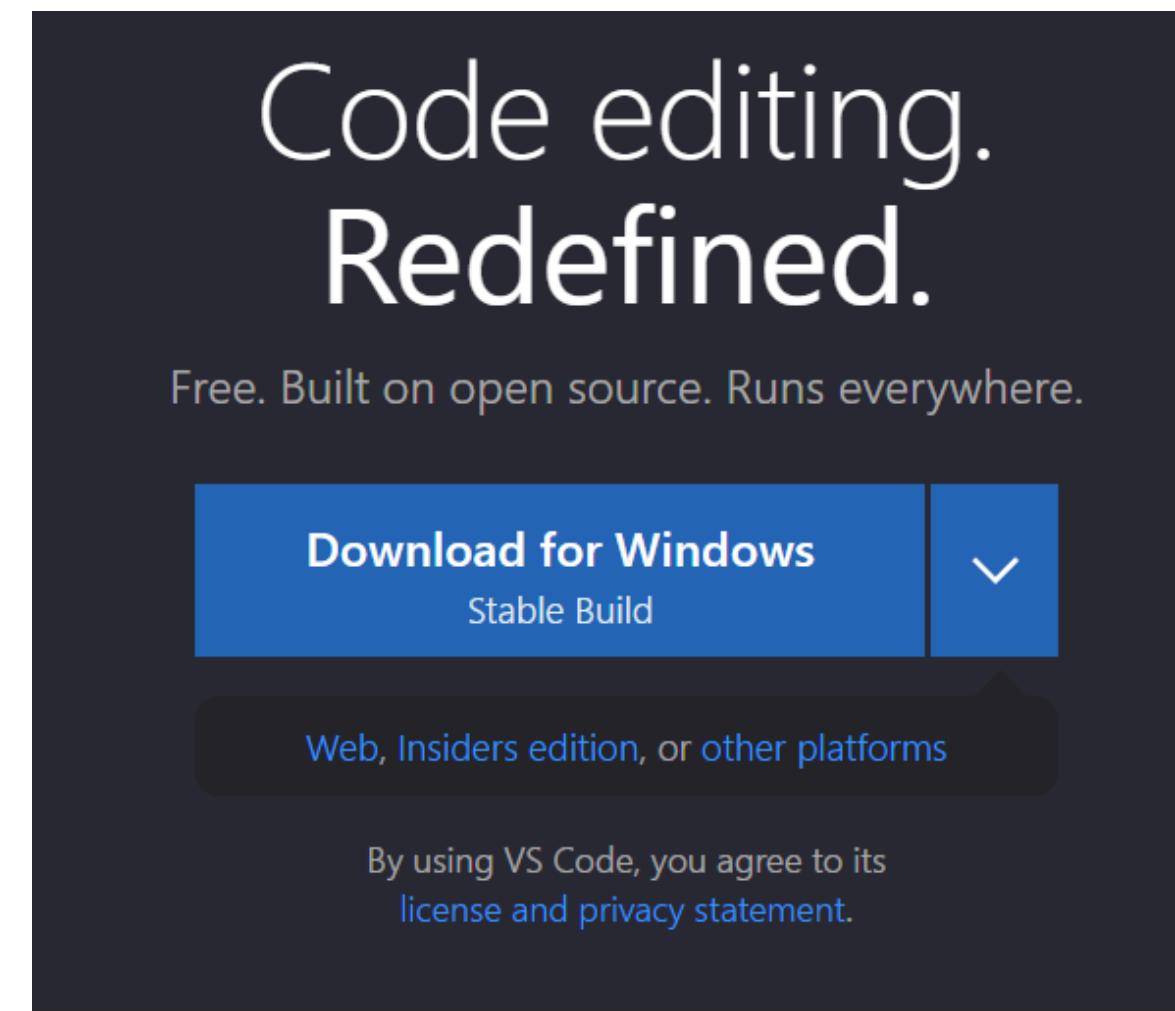
- **Windows Users:**

- Our *HIGHLY recommend* text editor for this course is VS Code:
 - **<https://code.visualstudio.com/>**
- Why *HIGHLY recommended*?
 - Windows + VS Code + GitHub
 - Upon installing git you will be asked to select a default editor, you'll need VS Code installed to select it as default



Week 1 - Starting with Git

- **Windows Users:**
 - Go to:
 - <https://code.visualstudio.com/>
 - Download with Default Settings:





Week 1 - Starting with Git

- **Windows Users:**
 - Next we'll download git, go to:
 - **<https://git-scm.com/>**

The screenshot shows the 'Downloads' section of the official Git website. At the top, there's a navigation bar with links for 'About', 'Documentation', 'Downloads', 'Community', and 'Logos'. The 'Downloads' link is highlighted in red. Below the navigation, there's a sidebar with links for 'macOS', 'Windows', and 'Linux/Unix'. The main content area features a large image of a Mac monitor displaying the latest source release version '2.38.1'. Text next to the image says 'Latest source Release 2.38.1' and 'Release Notes (2022-10-07)'. A 'Download for Mac' button is visible. To the right, there are sections for 'GUI Clients' and 'Logos', both with descriptive text and links. At the bottom, there's a section for 'Git via Git' with instructions for cloning the repository.

DAY 1

Configure Git

Week 1 - Starting with Git

- You can check the current configuration with the commands:
- The configuration commands will be:
 - **git config --global user.name “user”**
 - **git config --global user.email “email”**
- If switch with another github account
 - **git config --global user.name “user”**
 - **git config --global user.email “email”**
 - **git config --global credential.username “user”**

- -- au config เก่า --
 - git config --global --unset user.name
 - git config --global --unset user.email
 - git config --global --unset credential.username
-
- -- au origin ลบเก่า ถ้า commit ไม่ได้ --
 - git remote remove origin

Show global Git configuration?

```
git config --list or git config -l
```

or look at your `~/.gitconfig` file. The local configuration will be in your repository's `.git/config` file.

```
git config --list --show-origin
```

Week 1 - Starting with Git

- Let's head over to our command line interface to set-up our Git configuration:
 - Git Bash
 - Terminal
 - Command Prompt

DAY 1

Creating a Git Repository

Day 1 - Starting with Git

- How can we create a Git Repository?
 - **git init**
 - This command initializes a Git Repository on your local machine.
 - You only need to run this command once per project.
 - **git status**
 - This command will report back the status of your Git repository.

Day 1 - Starting with Git

- How can we create a Git Repository?
 - Upon creating a repository with **git init** you will create a hidden .git file.
 - The .git file is a hidden file that manages the versioning of the files inside the Git repository.

Day 1 - Starting with Git

- Git inside a Folder/Directory:
 - Upon creating a Git Repository, all the folders/directories inside the top level Git Repository will also be part of that Repository, meaning all the changes are tracked.

Day 1 - Starting with Git

- Git inside a Folder/Directory:
 - Upon creating a Git Repository, all the folders/directories inside the top level Git Repository will also be part of that Repository, meaning all the changes are tracked.



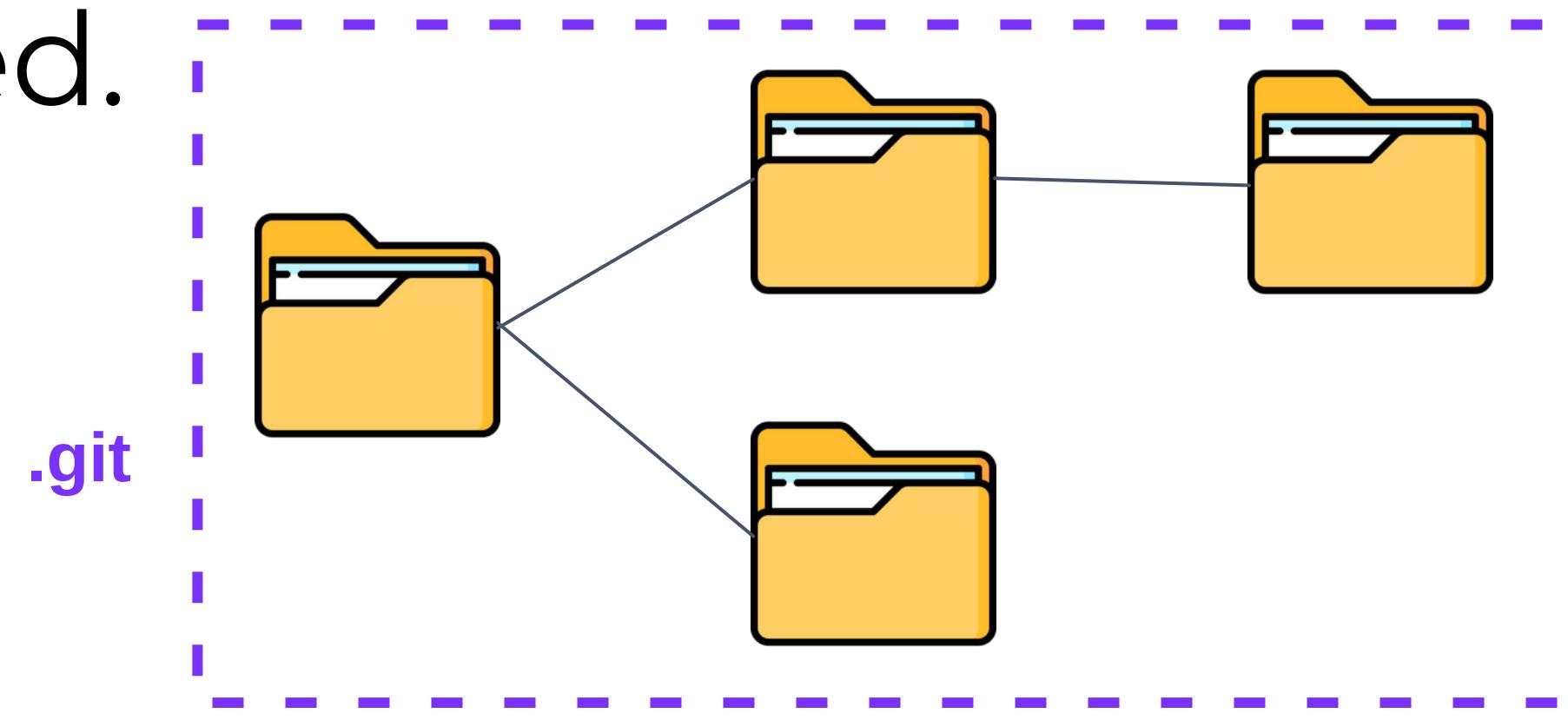
Day 1 - Starting with Git

- Git inside a Folder/Directory:
 - Upon creating a Git Repository, all the folders/directories inside the top level Git Repository will also be part of that Repository, meaning all the changes are tracked.



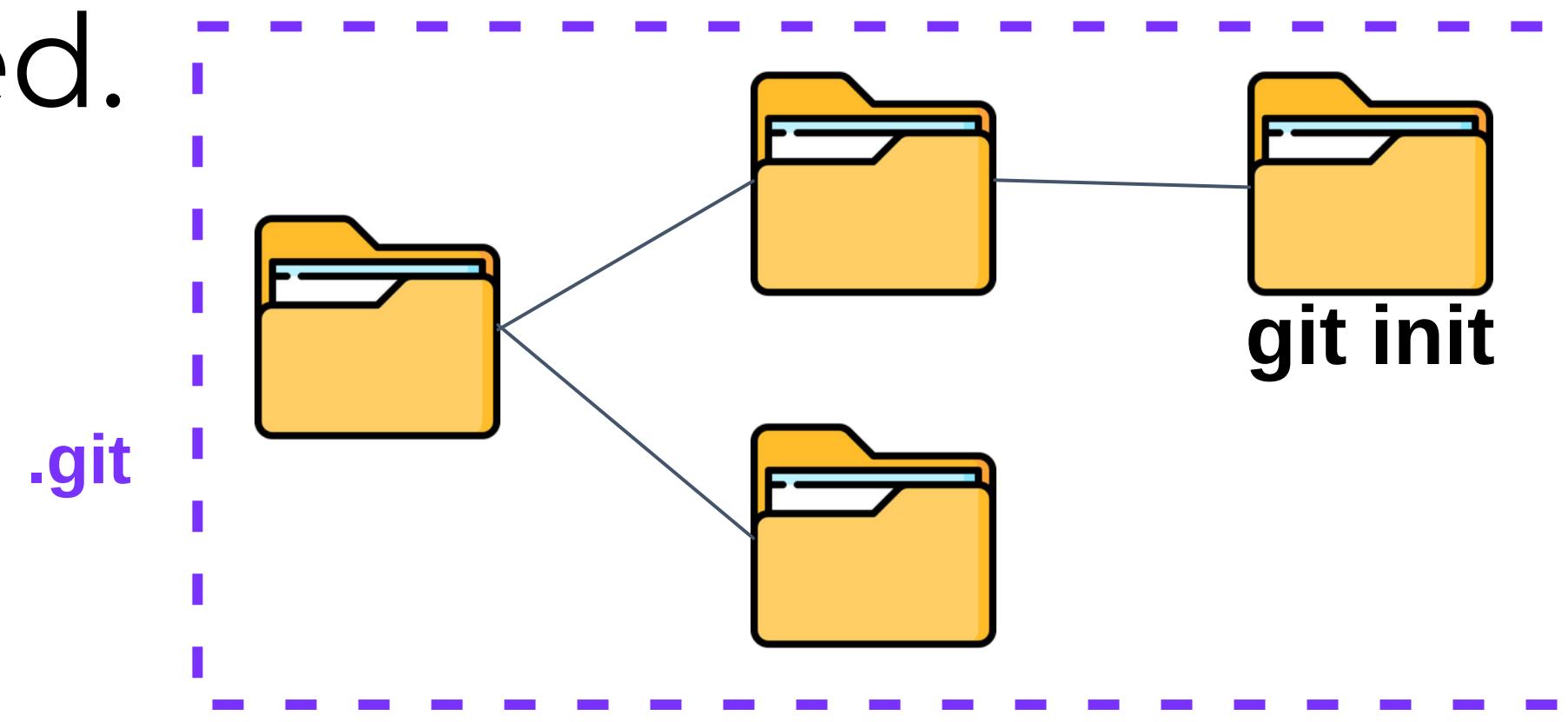
Day 1 - Starting with Git

- Git inside a Folder/Directory:
 - Upon creating a Git Repository, all the folders/directories inside the top level Git Repository will also be part of that Repository, meaning all the changes are tracked.



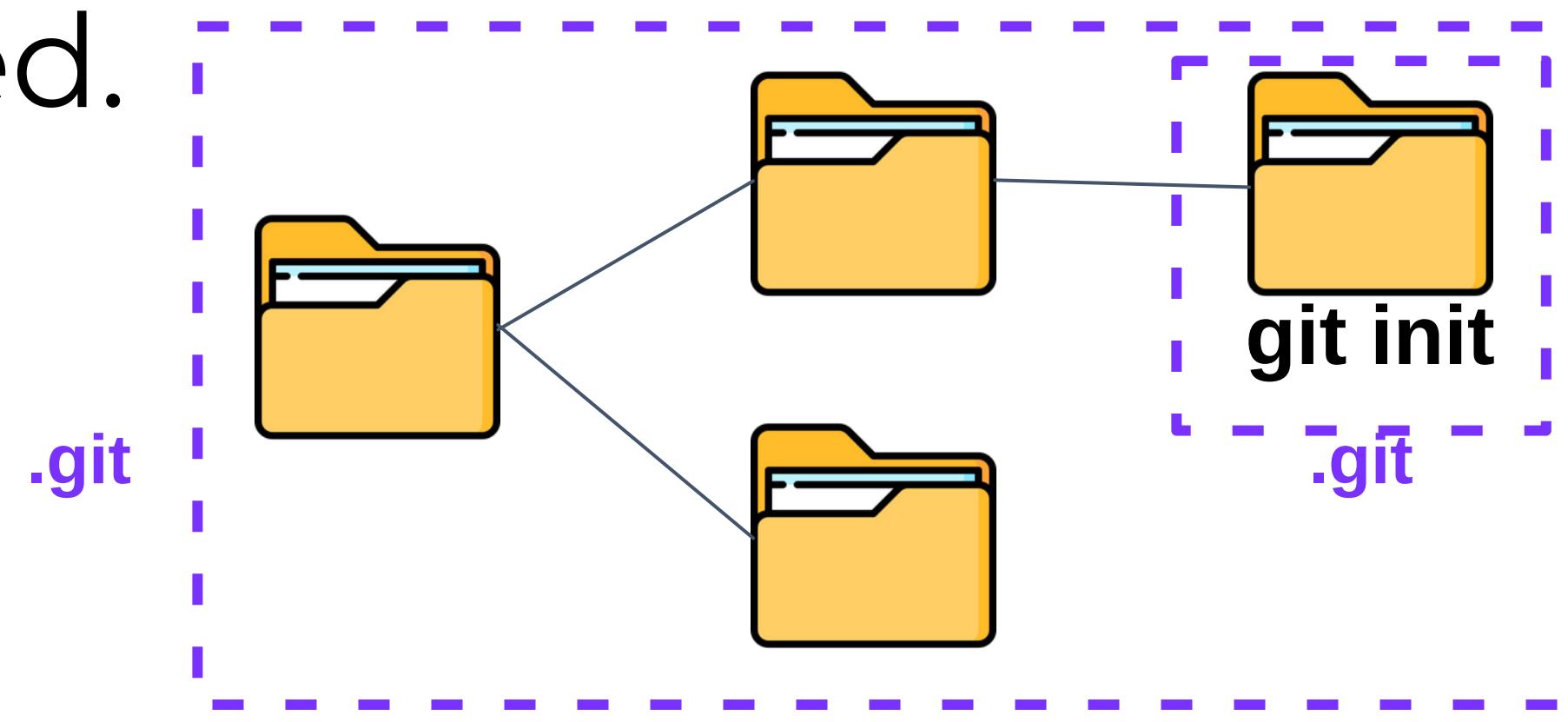
Day 1 - Starting with Git

- Git inside a Folder/Directory:
 - Upon creating a Git Repository, all the folders/directories inside the top level Git Repository will also be part of that Repository, meaning all the changes are tracked.



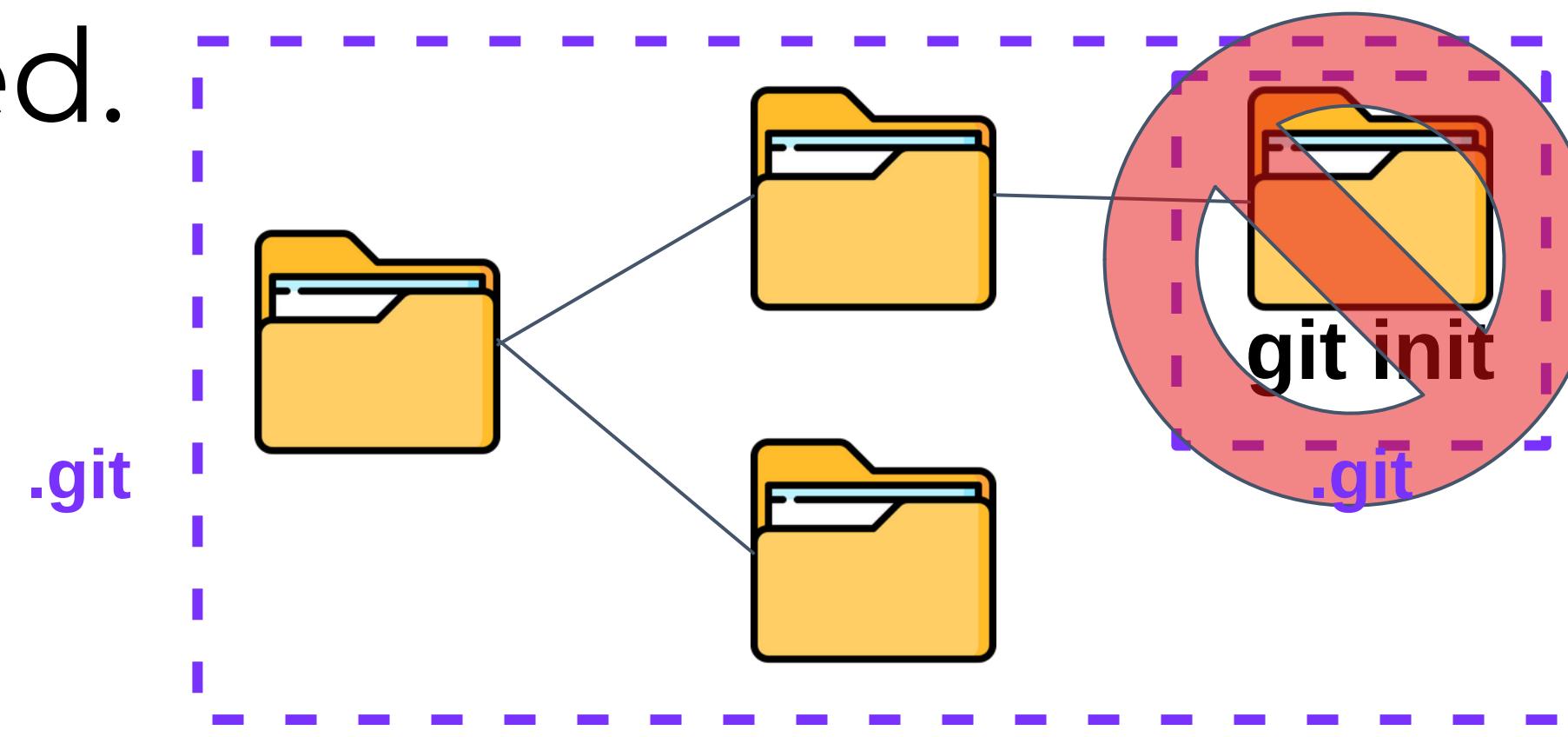
Day 1 - Starting with Git

- Git inside a Folder/Directory:
 - Upon creating a Git Repository, all the folders/directories inside the top level Git Repository will also be part of that Repository, meaning all the changes are tracked.



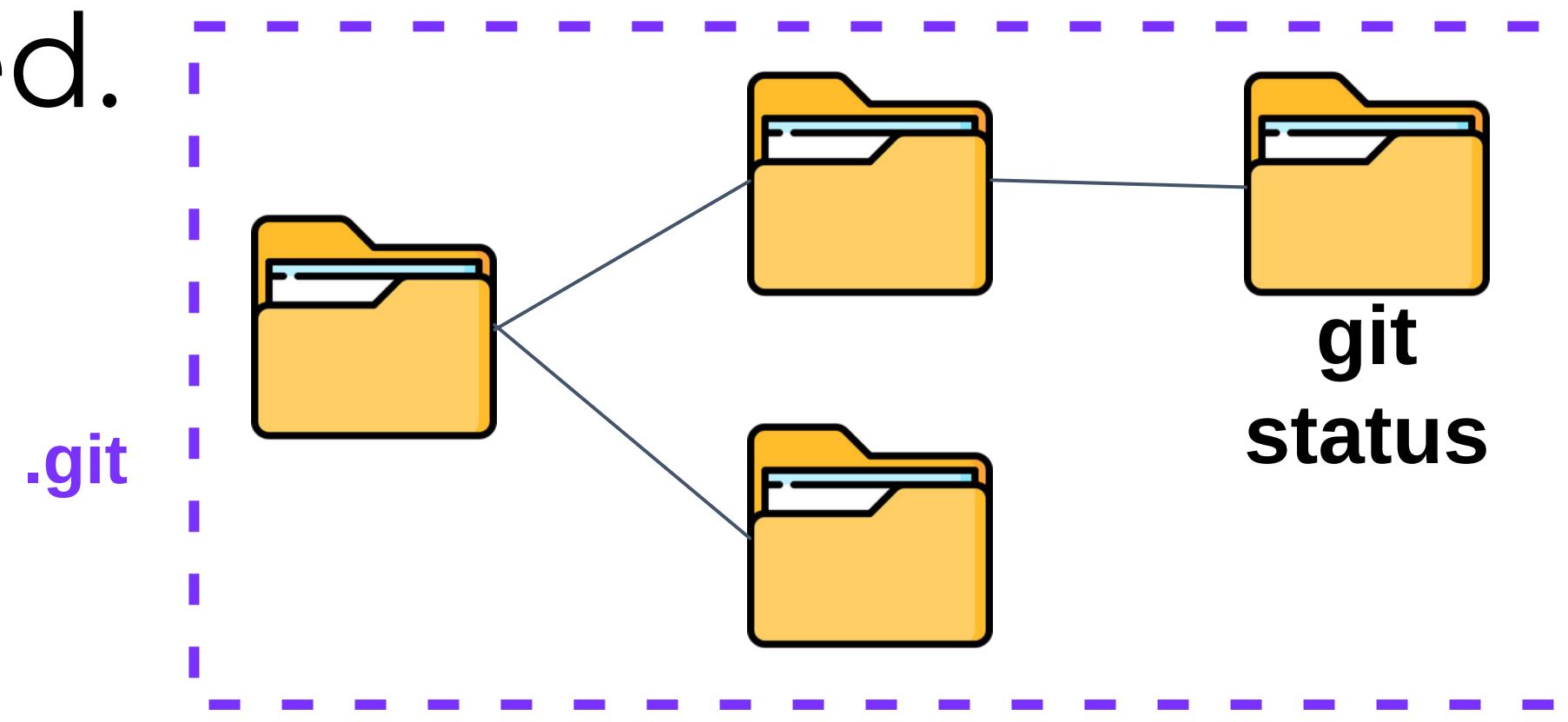
Day 1 - Starting with Git

- Git inside a Folder/Directory:
 - Upon creating a Git Repository, all the folders/directories inside the top level Git Repository will also be part of that Repository, meaning all the changes are tracked.



Day 1 - Starting with Git

- Git inside a Folder/Directory:
 - Upon creating a Git Repository, all the folders/directories inside the top level Git Repository will also be part of that Repository, meaning all the changes are tracked.



Ignoring Files

We can tell Git which files and directories to ignore in a given repository, using a `.gitignore` file. This is useful for files you know you NEVER want to commit, including:

- Secrets, API keys, credentials, etc.
- Operating System files
- (`.DS_Store` on Mac)
- Log files
- Dependencies & packages



.gitignore

Create a file called `.gitignore` in the root of a repository. Inside the file, we can write patterns to tell Git which files & folders to ignore:

- `.DS_Store` will ignore files named `.DS_Store`
- `folderName/` will ignore an entire directory
- `*.log` will ignore any files with the `.log` extension

<https://www.toptal.com/developers/gitignore>



gitignore.io

สร้างไฟล์ .gitignore ที่มีประโยชน์สำหรับโปรเจกต์ของคุณ

ดันพารามบ์ไปติด IDE หรือภาษาการเขียนโปรแกรม

สร้าง

ชื่อรสโตค | ลองอ่านเอกสารคำสั่งของ คอมมานต์ไลน์ คุณ!



DAY 1

Private Repositories and Tokens

Day 1 - Starting with Git

- Clone Syntax with PAT:

```
git clone https://username:YOUR_TOKEN@github.com/username/repo.git
```

- Previously we used:

```
git clone https://github.com/account/repo.git
```

Create a Personal Access Token

You unlocked new Achievements with private contributions! Show them off by including private contributions in your Profile in settings.

Pinned

Customize your pins

Public

Jupyter Notebook ⭐ 1 ⚡ 7

Jupyter Notebook ⭐ 10 ⚡ 14

Jupyter Notebook ⭐ 1 ⚡ 1

746 contributions in the last year

Contribution settings ▾

	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep
Mon	■			■	■	■	■	■	■	■	■	■	■
Wed	■	■	■	■	■	■	■	■	■	■	■	■	■
Fri	■	■	■	■	■	■	■	■	■	■	■	■	■

1

Click on your profile picture in the upper-right corner of the screen and select "Settings."

Set status

Your profile

Your repositories

Your projects

Your codespaces

Your organizations

Your enterprises

Your stars

Your sponsors

Your gists

Upgrade

Try Enterprise

Try Copilot

Feature preview

Settings

GitHub Docs

GitHub Support

Sign out

2

1 Your personal account

2 Public profile

3 Account

Appearance

Accessibility

Notifications

4 Billing and plans

5 Emails

6 Password and authentication

Sessions

SSH and GPG keys

Organizations

Enterprises

Moderation

7 Planning, automation

8 Repositories

Codespaces

Packages

Copilot

Pages

Saved replies

9 Security

Code security and analysis

10 Applications

Scheduled reminders

11 Lives

Security log

Sponsorship log

12 Developer settings

13 Update profile

In the left sidebar, select "Developer settings."

This screenshot shows the GitHub Public profile settings page. The sidebar on the left has 'Developer settings' highlighted with a yellow box and a large blue arrow pointing to it. A callout box with a yellow border contains the text 'In the left sidebar, select "Developer settings."'. The main content area shows various profile settings like Name, Bio, and Social accounts.

1 GitHub Apps

2 OAuth Apps

3 Personal access tokens

4 GitHub Apps

Want to build something that integrates with and extends GitHub? [Register a new GitHub App](#) to get started developing on the GitHub API. You can also read more about building GitHub Apps in our [developer documentation](#).

5 Personal access tokens (classic)

6 Generate new token

This screenshot shows the GitHub Developer Settings page. The 'Personal access tokens' section is highlighted with a red box. A large yellow number '4' is overlaid on the page. Below it, another screenshot shows the 'Personal access tokens (classic)' section with a red box around the 'Generate new token' button. A large yellow number '5' is overlaid on the page. At the bottom right, a large yellow number '6' is displayed.

7 Personal access tokens (classic)

8 Need an API token for scripts or testing? [Generate a personal access token](#) for quick access to the GitHub API.

9 Personal access tokens (classic) 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](#).

10 2023 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

11 5

12 6

This screenshot shows the GitHub Developer Settings page focusing on the 'Personal access tokens (classic)' section. The 'Tokens (classic)' tab is selected. A large yellow number '7' is overlaid on the page. Below it, another screenshot shows the same section with a red box around the 'Generate new token' button. A large yellow number '8' is overlaid on the page. At the bottom right, a large yellow number '9' is displayed.

New personal access token (classic)

Personal access tokens (classic) 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
test

What's this token for?

Expiration *
7 days The token will expire on Tue, Sep 26 2023

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 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
<input type="checkbox"/> manage_runners:org	Manage org runners and runner groups
<input type="checkbox"/> admin:public_key	Full control of user public keys
<input type="checkbox"/> write:public_key	Write user public keys
<input type="checkbox"/> read:public_key	Read user public keys
<input checked="" type="checkbox"/> admin:repo_hook	Full control of repository hooks
<input checked="" type="checkbox"/> write:repo_hook	Write repository hooks
<input checked="" type="checkbox"/> read:repo_hook	Read repository hooks
<input type="checkbox"/> admin:org_hook	Full control of organization hooks
<input type="checkbox"/> gist	Create gists
<input type="checkbox"/> notifications	Access notifications
<input type="checkbox"/> user	Update ALL user data
<input type="checkbox"/> read:user	Read ALL user profile data
<input type="checkbox"/> user:email	Access user email addresses (read-only)
<input type="checkbox"/> user:follow	Follow and unfollow users

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens Beta

Tokens (classic)

Personal access tokens (classic)

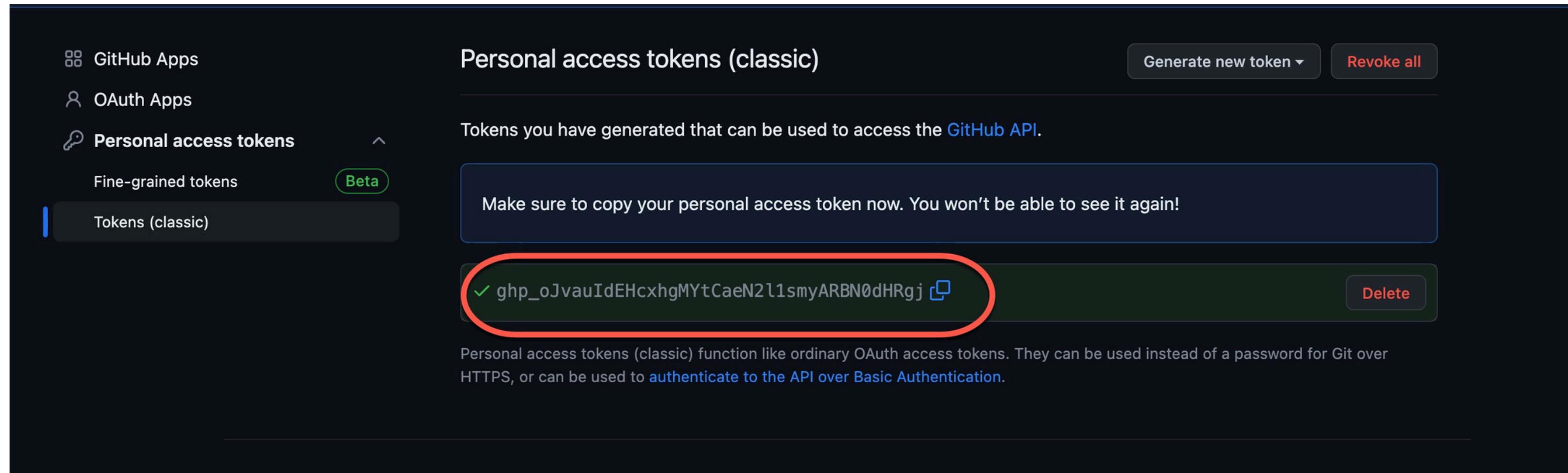
Generate new token ▾ Revoke all

Tokens you have generated that can be used to access the GitHub API.

Make sure to copy your personal access token now. You won't be able to see it again!

✓ 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 ⚡ Delete

Personal access tokens (classic) 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](#).



DAY 1

Exercise

Day 1 - Starting with Git

- **Exercise Tasks:**

- Create a new **Private Repository** on GitHub.
- Initialize your repository with README, license and gitignore.
- create new test.txt with random data
- Clone your Repository using the Command Line .

1

2

Dashboard

Type / to search

Top repositories

New

Find a repository...

Join GitHub Education!

GitHub Education opens doors to new skills, tools, and a collaborative community eager to drive innovation. Join us and build a foundation for your future in technology.

Free and discounted services for teachers and students.

Copilot
Heroku
Microsoft Azure

Join GitHub Education

Home

Give feedback

Filter

...

IISourceLL made this repository public 12 hours ago

IISourceLL/DualStrike_AI_For_Lymphoma Python ⭐ 1

...

ALucek made this repository public 20 hours ago

ALucek/chunking-strategies Jupyter Notebook

...

Dashboard

Type / to search

Top repositories

New

Find a repository...

Tuchsanai/DevTools

Tuchsanai/AIMaster-seagate-training-2024

Tuchsanai/DL-FOR-COMPUTER-VISION

Tuchsanai/DGX_2024

Tuchsanai/tp_idea1_2024

Tuchsanai/Machine_Learning

Tuchsanai/pytorch_docker

Show more

Join GitHub Education

Home

IISourceLL made this repository public 12 hours ago

IISourceLL/DualStrike_AI_For_Lymphoma Python ⭐ 1

...

ALucek made this repository public 20 hours ago

ALucek/chunking-strategies Jupyter Notebook

...

Tuchsanai

tuchsanai

Set status

Your profile

Your repositories **2**

Your Copilot

Your projects

Your stars

Your gists

Your organizations

Your enterprises

Your sponsors

Try Enterprise

Feature preview

Settings

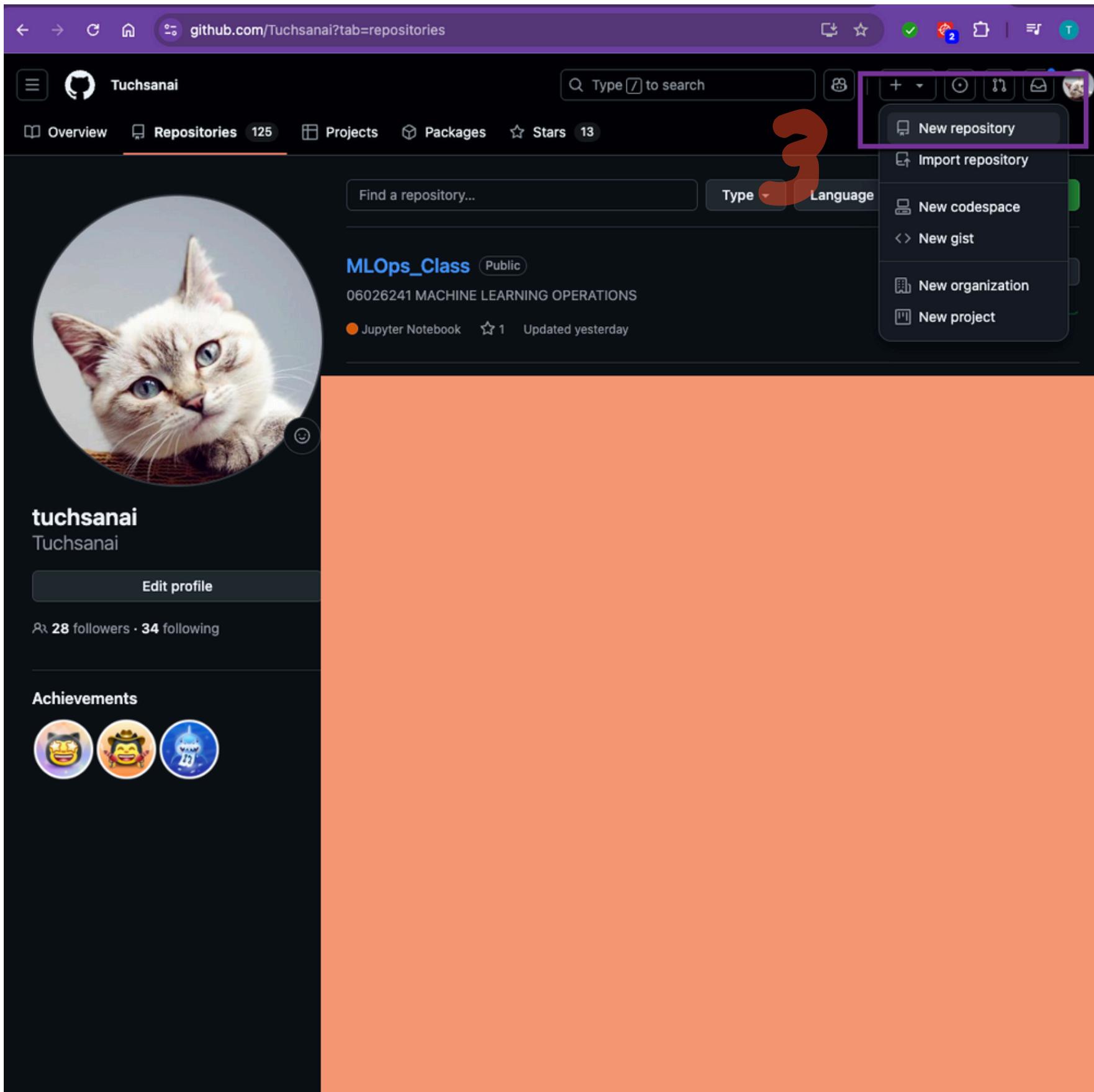
GitHub Docs

GitHub Support

GitHub Community

Sign out

Click New repository



Create a new repository

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

Required fields are marked with an asterisk (*).

Owner * Tuchsanai

Repository name * test_week1
test_week1 is available.

Great repository names are short and memorable. Need inspiration? How about [super-duper-disco](#) ?

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:

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

Add .gitignore

.gitignore template: Python

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

Choose a license

License: None

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

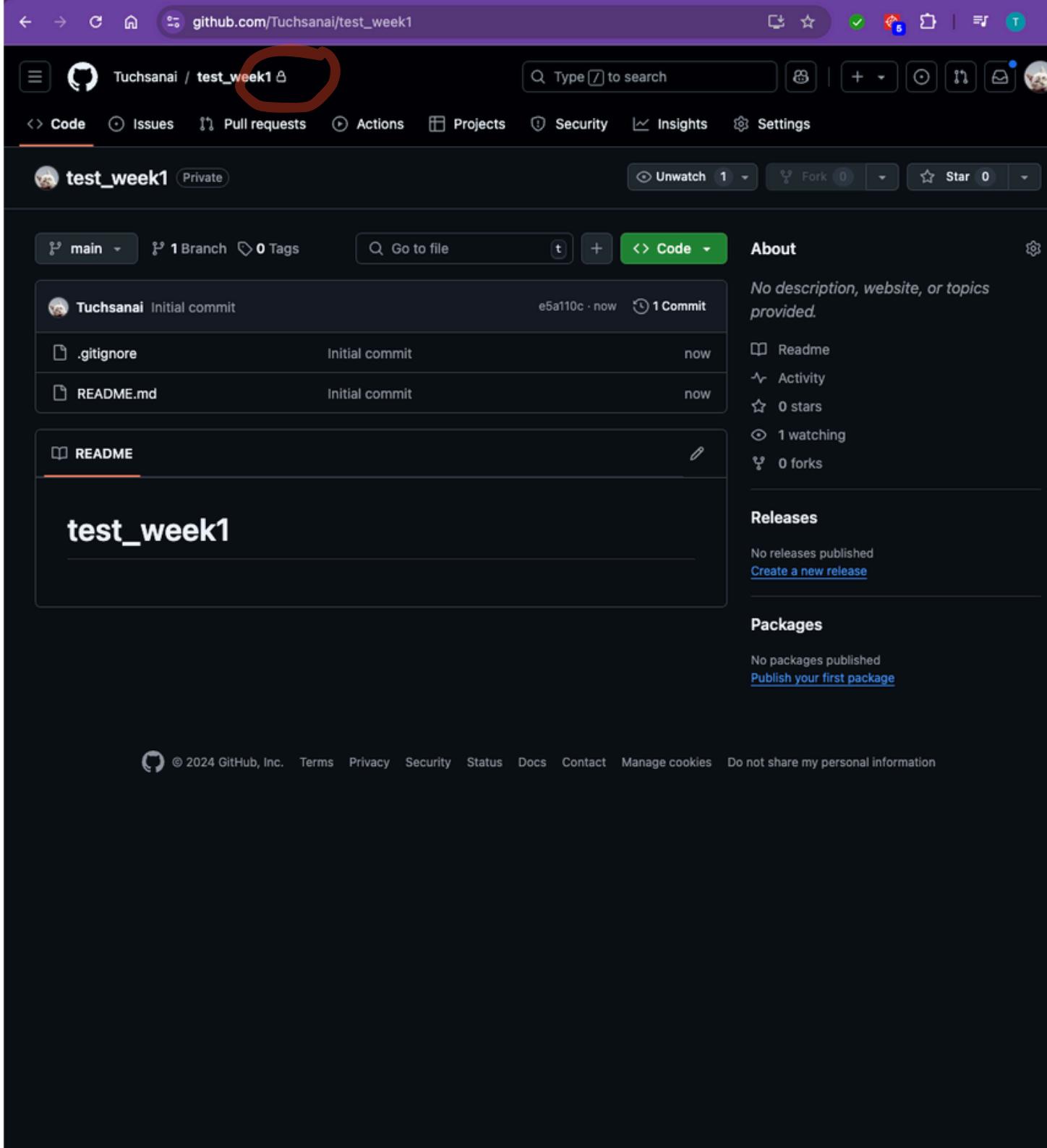
This will set `main` as the default branch. Change the default name in your [settings](#).

You are creating a private repository in your personal account.

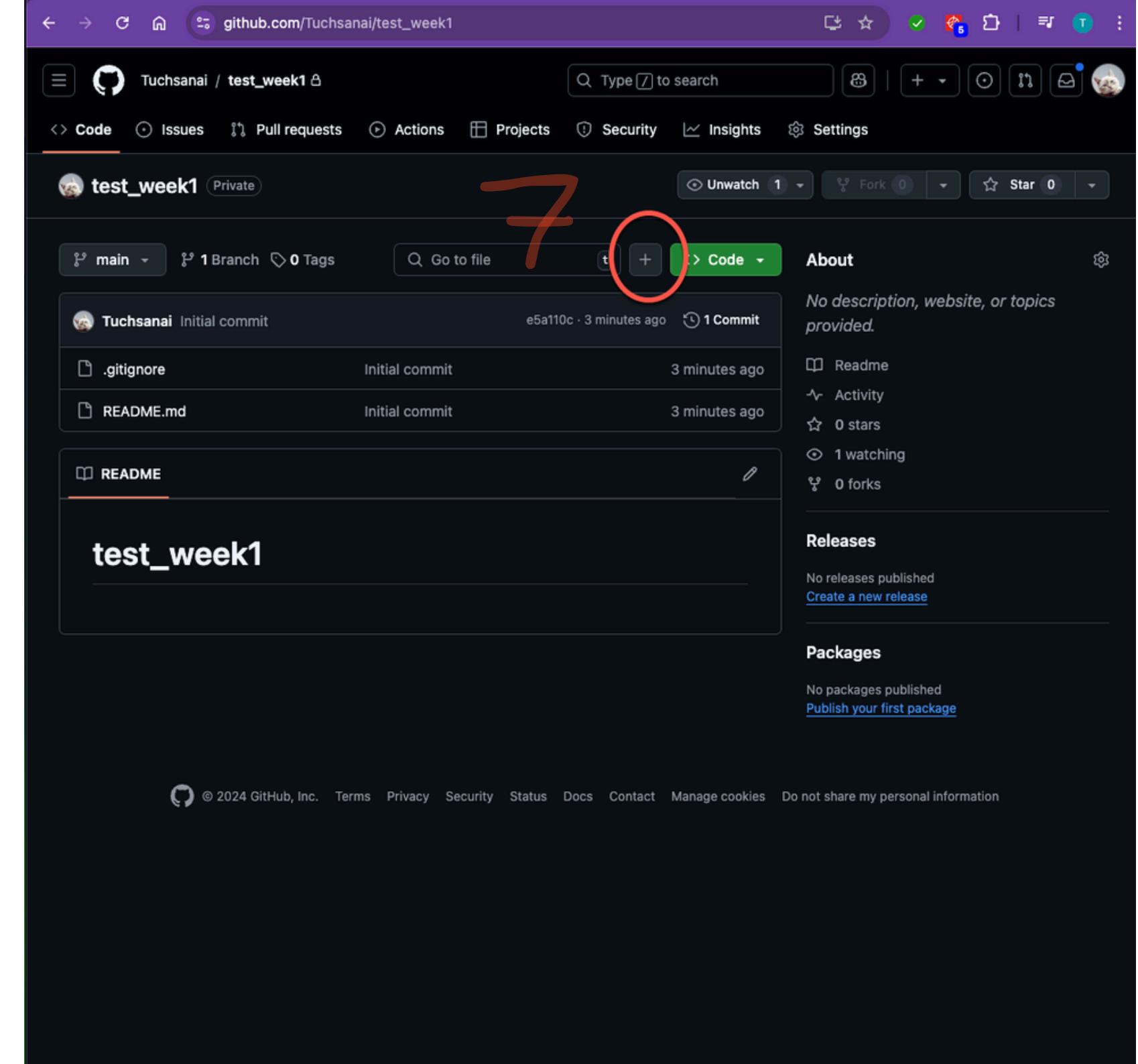
Create repository

A screenshot of the 'Create a new repository' form on GitHub. The 'Repository name' field contains 'test_week1' and is circled in red. The 'Private' radio button is selected and circled in red. The 'Add a README file' checkbox is checked and circled in red. The '.gitignore template: Python' dropdown is open and circled in red. A large red number '4' is overlaid on the right side of the form.

Privated repository

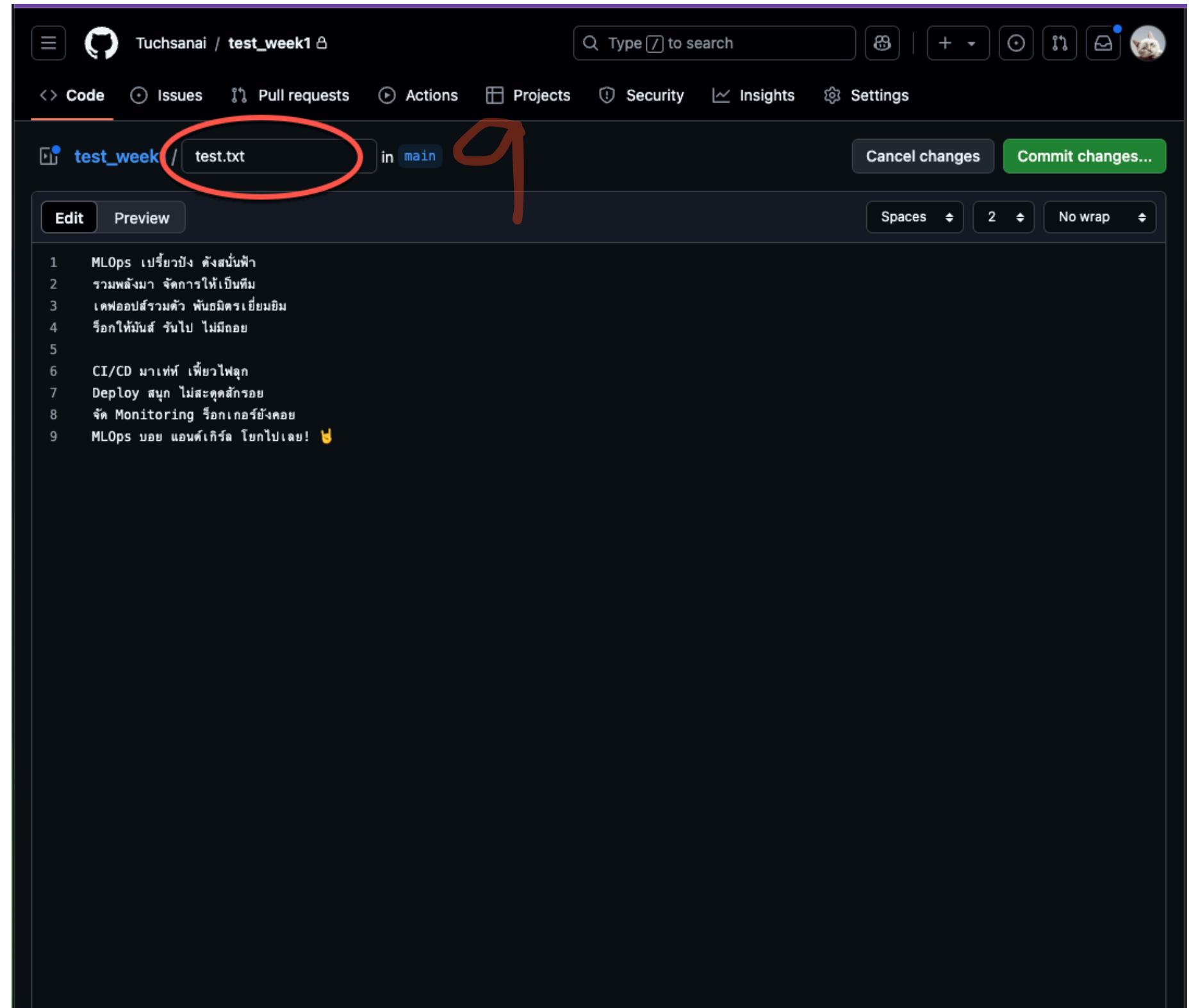
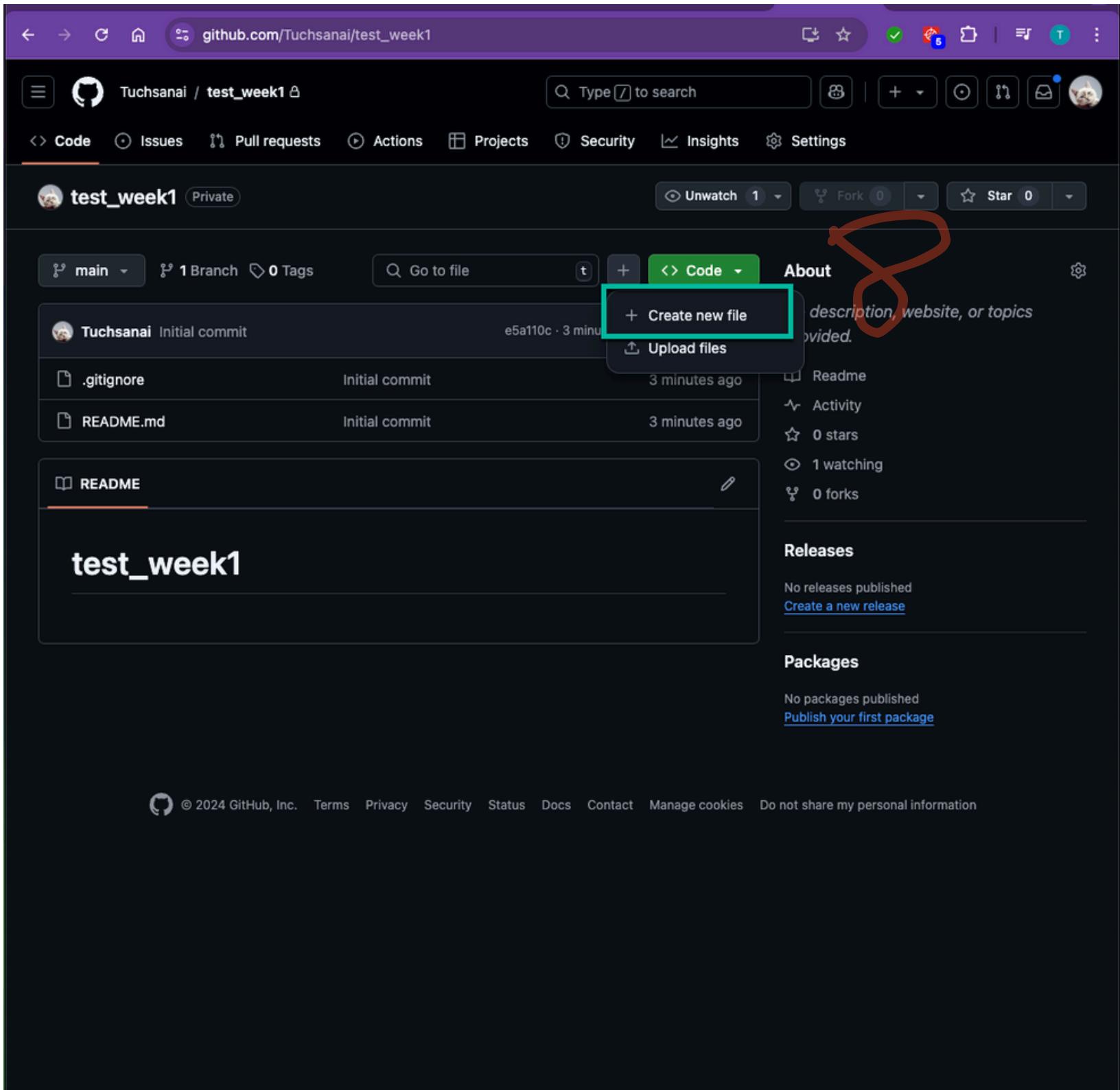


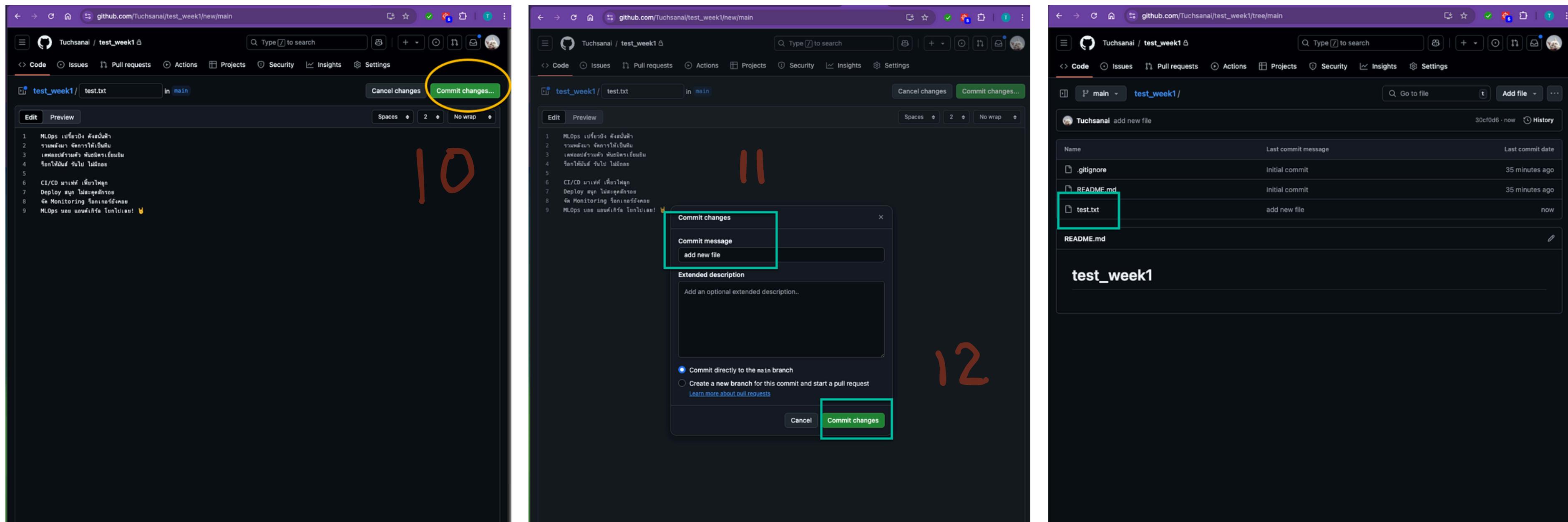
A screenshot of a GitHub repository page for 'test_week1'. The repository is private, as indicated by the 'Private' badge. The URL is github.com/Tuchsanai/test_week1. The repository has 1 commit, 1 branch, and 0 tags. The README file is visible. The 'Code' tab is selected. A red circle highlights the repository name 'test_week1' in the header.



A screenshot of the same GitHub repository page for 'test_week1' after a refresh. The repository is still private. The URL is github.com/Tuchsanai/test_week1. The repository has 1 commit, 1 branch, and 0 tags. The README file is visible. A large red '7' is drawn over the repository name. A red circle highlights the 'Code' tab. The commit information shows it was updated 3 minutes ago.

สร้าง file : test.txt





• Clone private repository:

git clone https://username:YOUR_TOKEN@github.com/username/repo.git