

# MACHINE LEARNING OPERATIONS



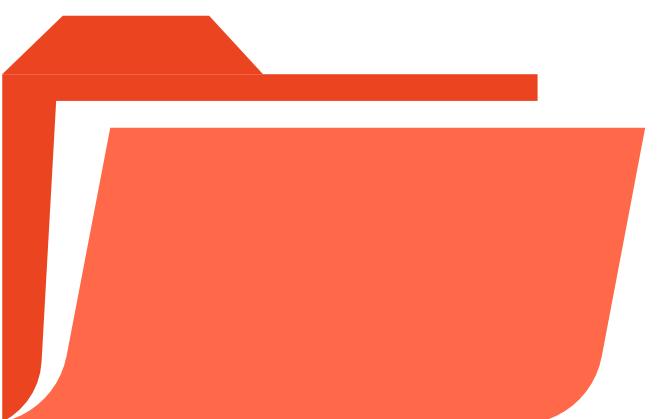
Presented by **Asst. Prof. Dr. Tuchsanai Ploysuwan**

WEEK 1



- 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!



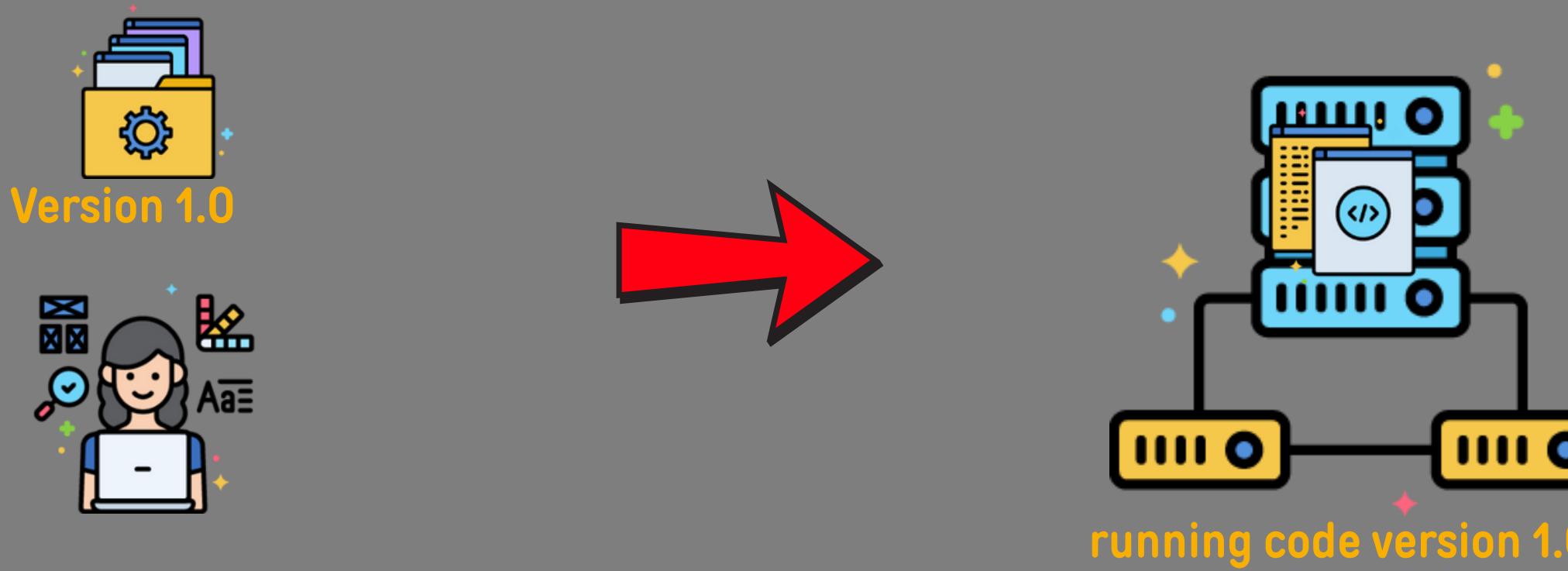
# Before Version Control System



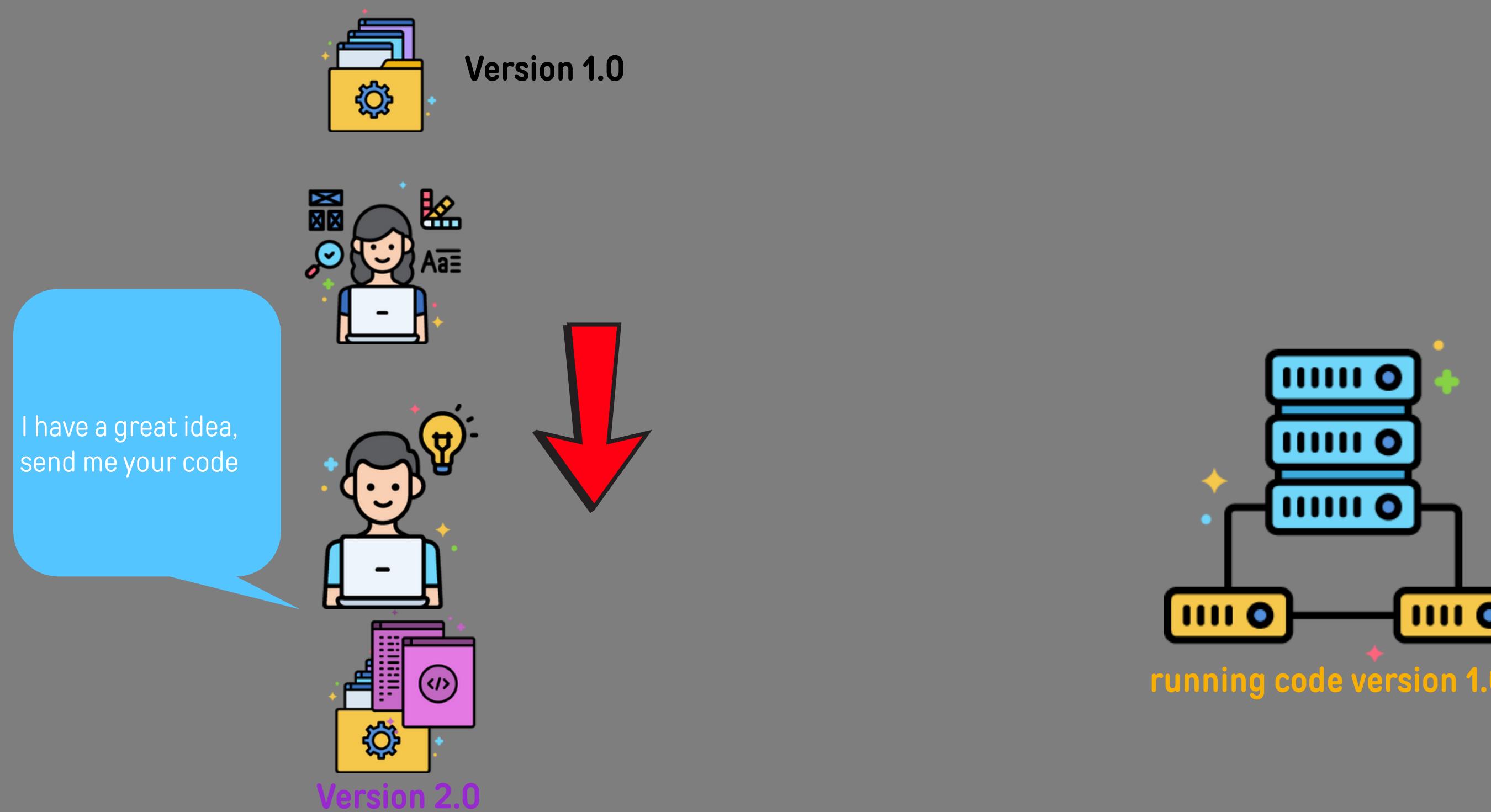
# Before Version Control System



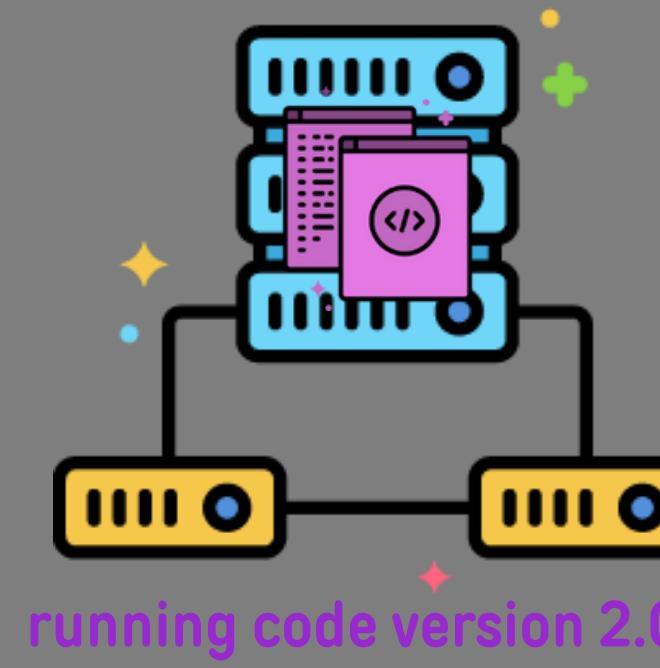
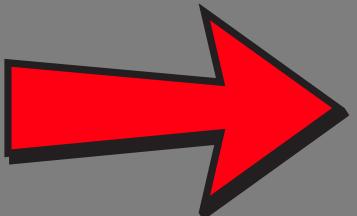
# Before Version Control System



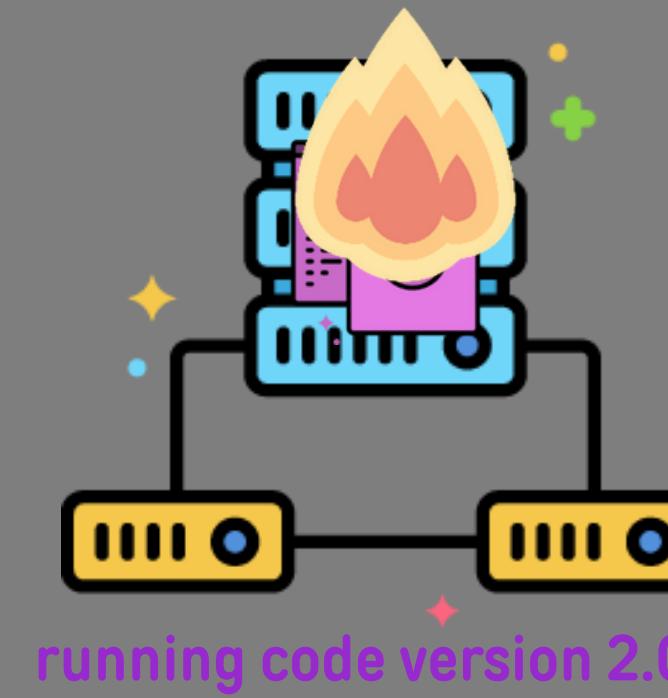
# Before Version Control System



# Before Version Control System



# Before Version Control System



# Before Version Control System



Version 1.0

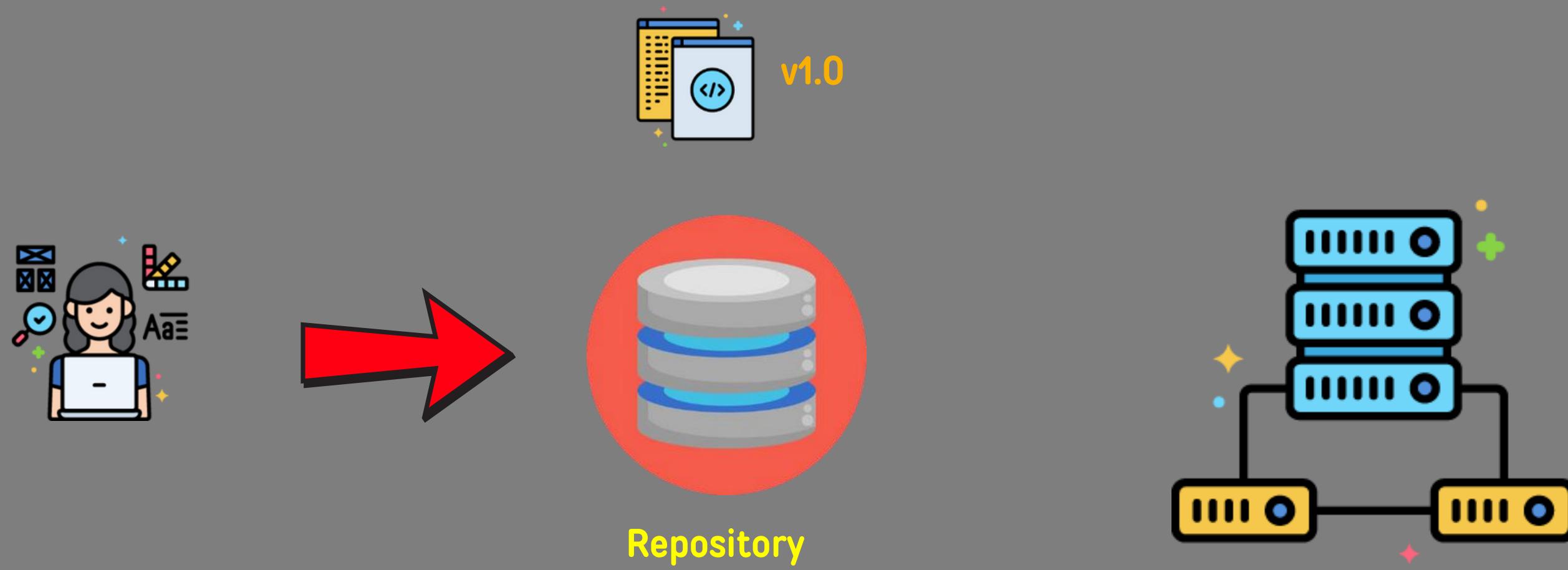


Version 2.0

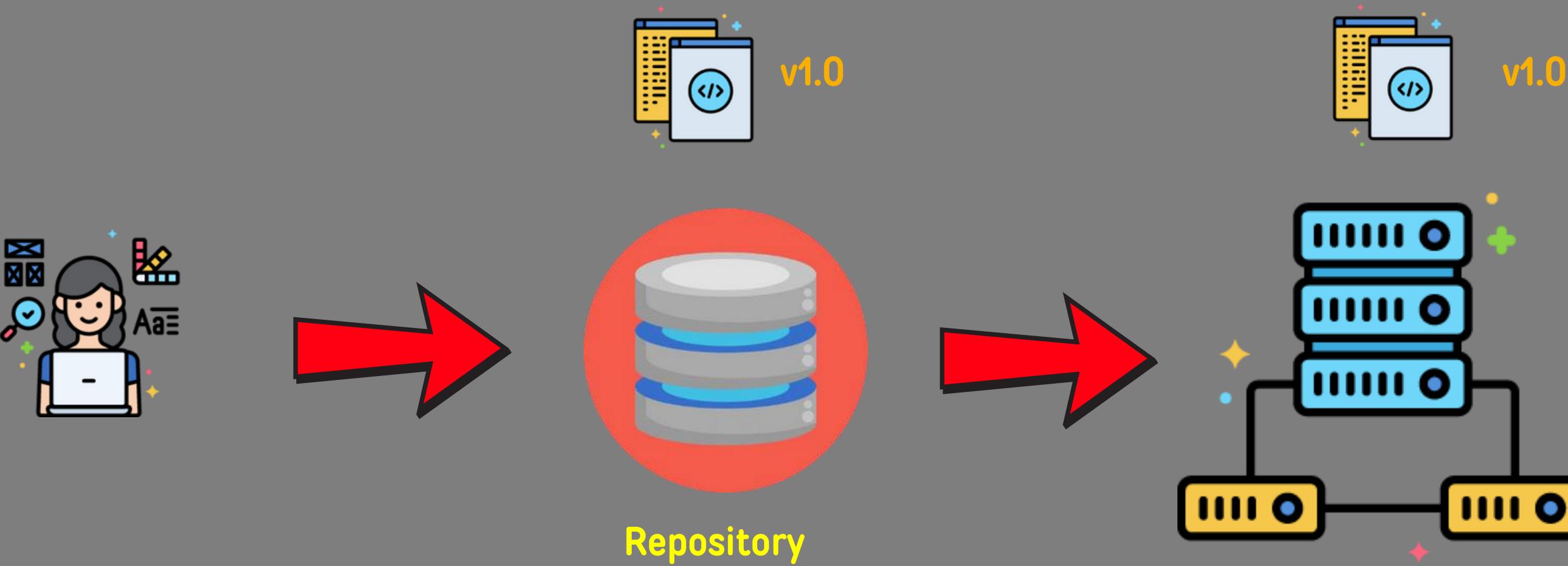


- Rollback is time consuming
- No audit tracking
- Not scalable for large teams

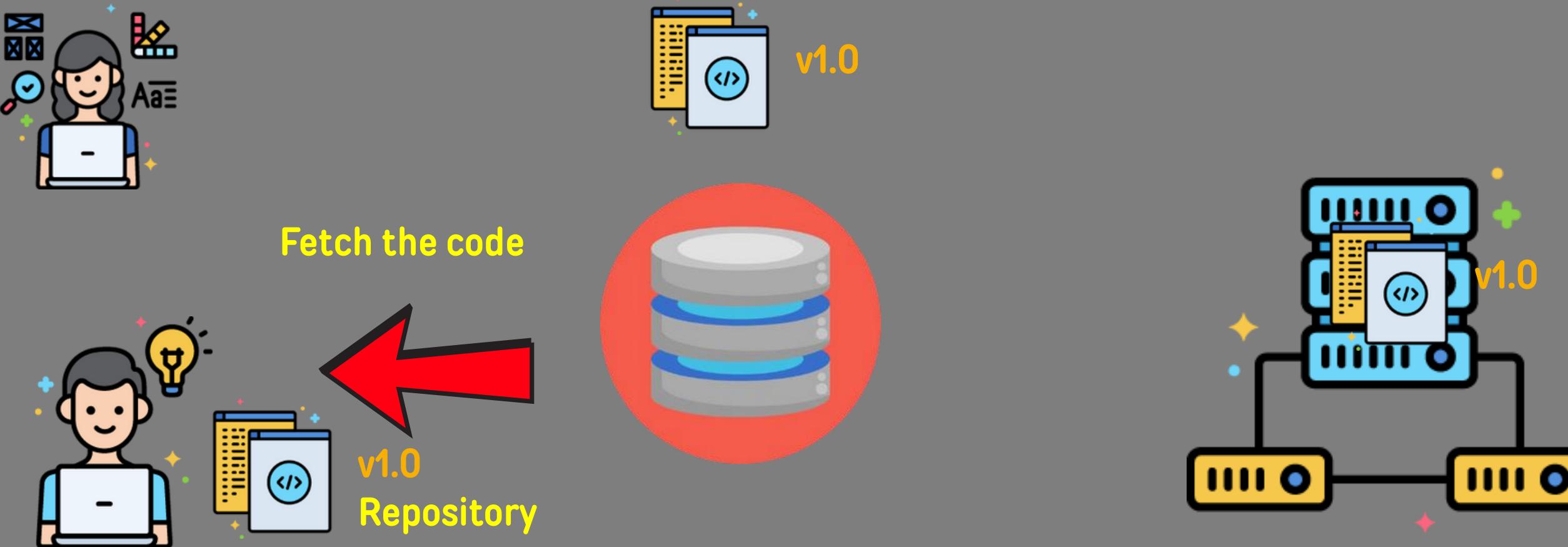
# Version Control System



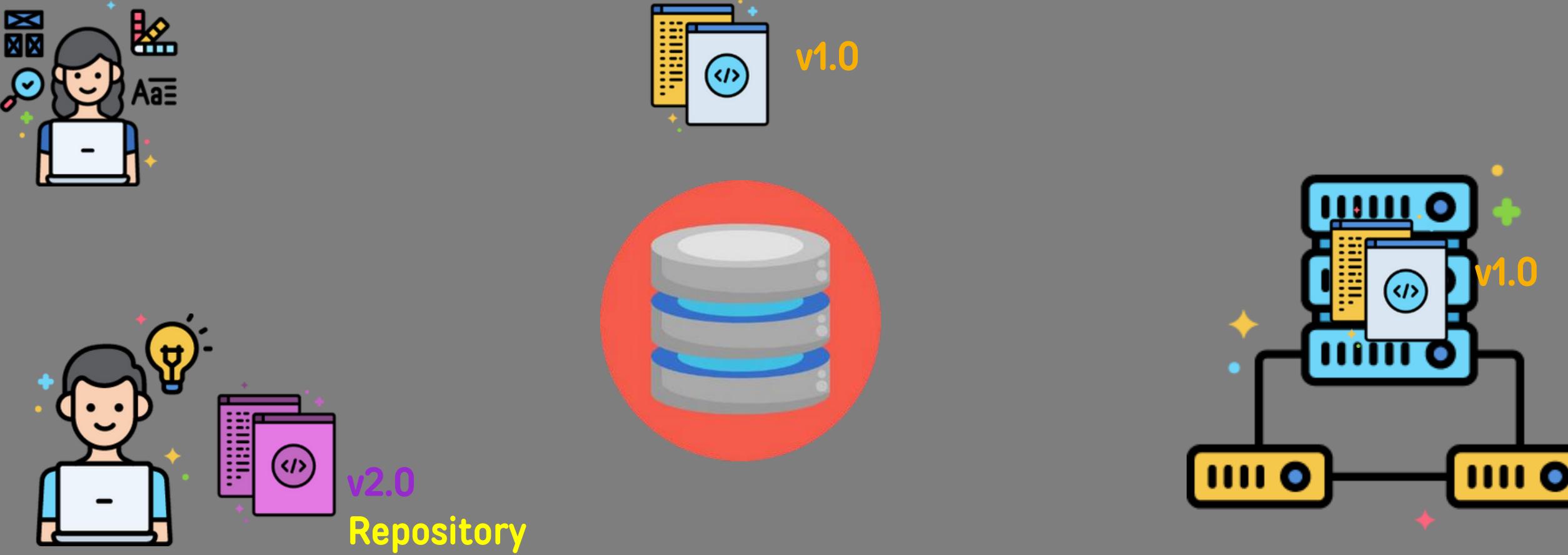
# Version Control System



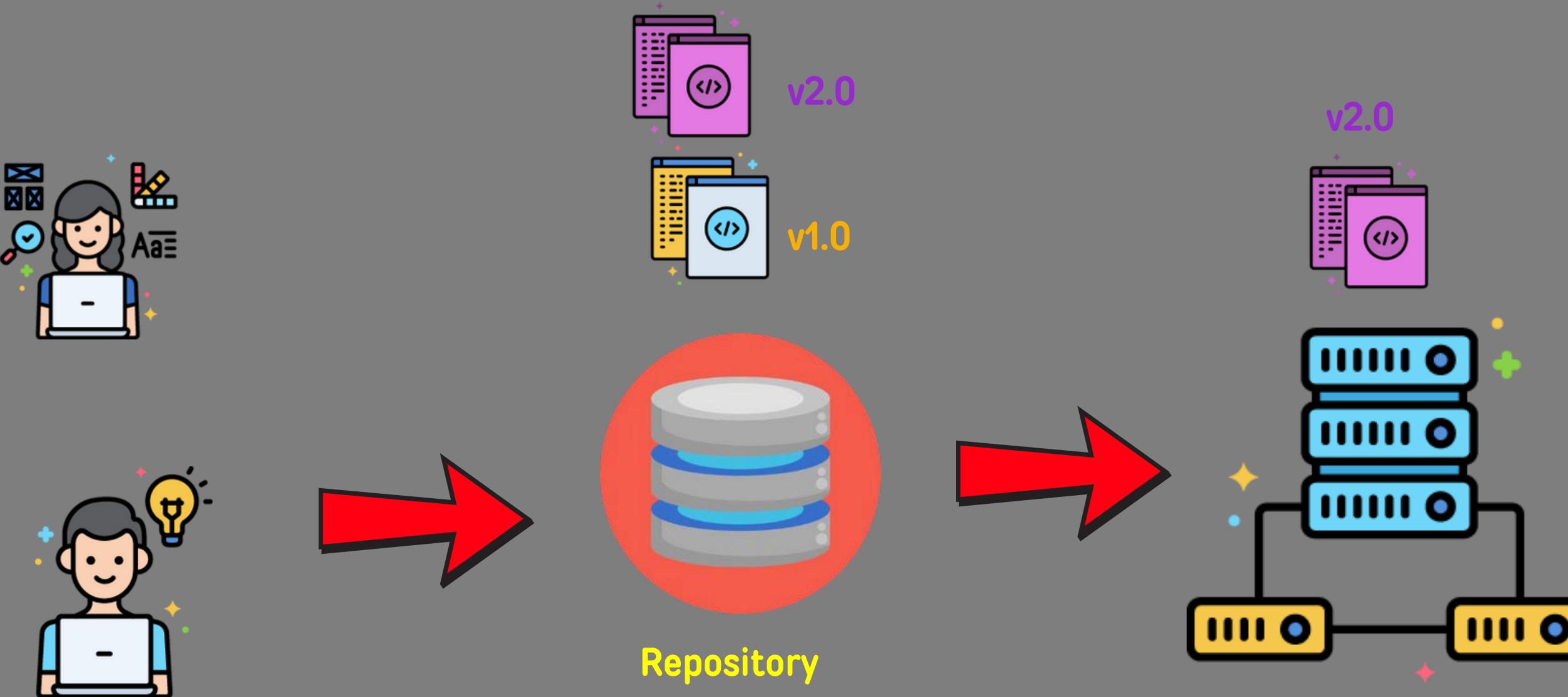
# Version Control System



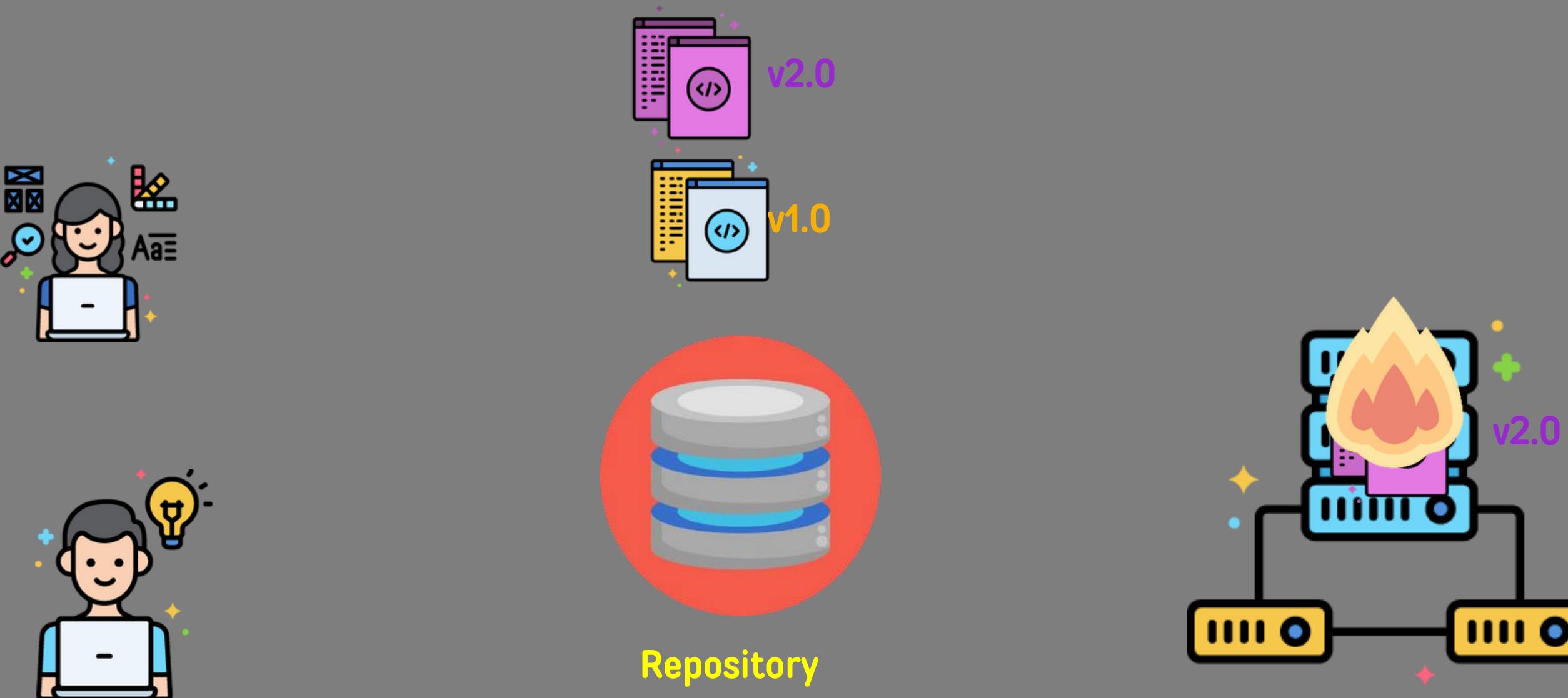
# Version Control System



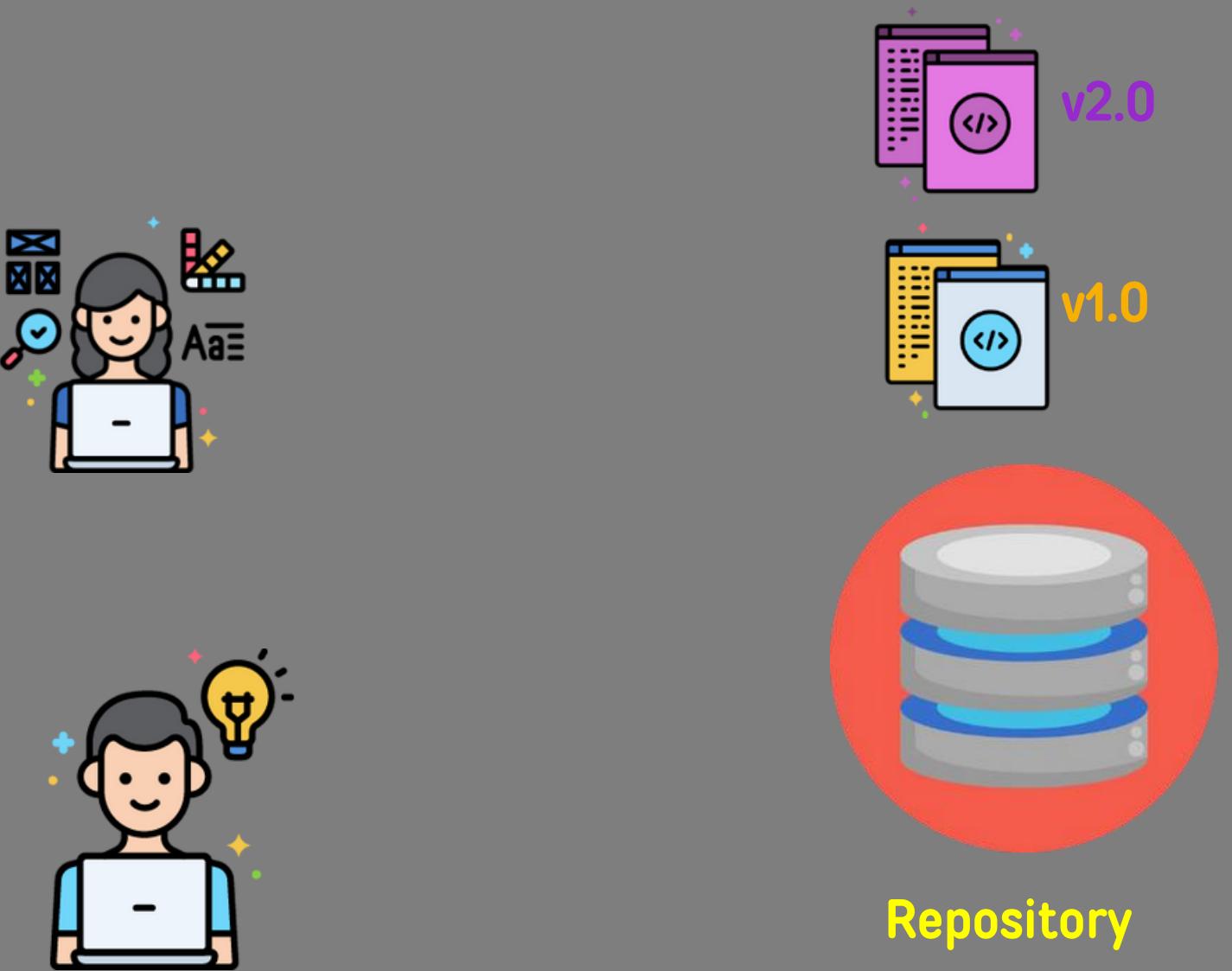
# Version Control System



# Version Control System

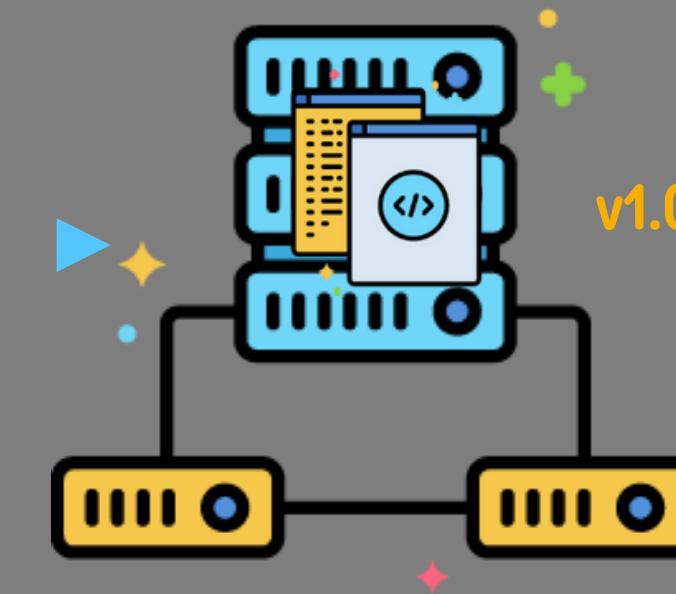


# Version Control System

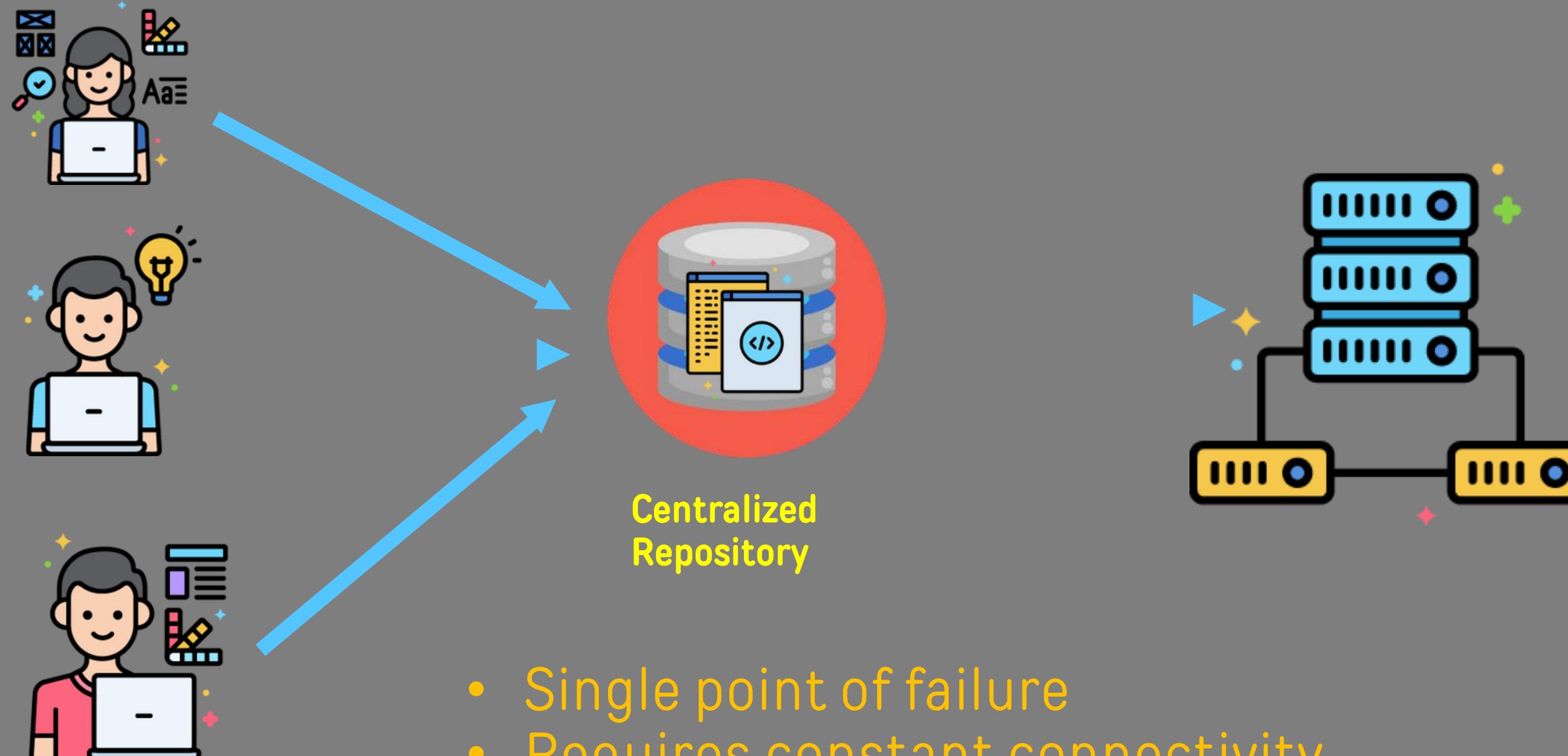


## Why Git?

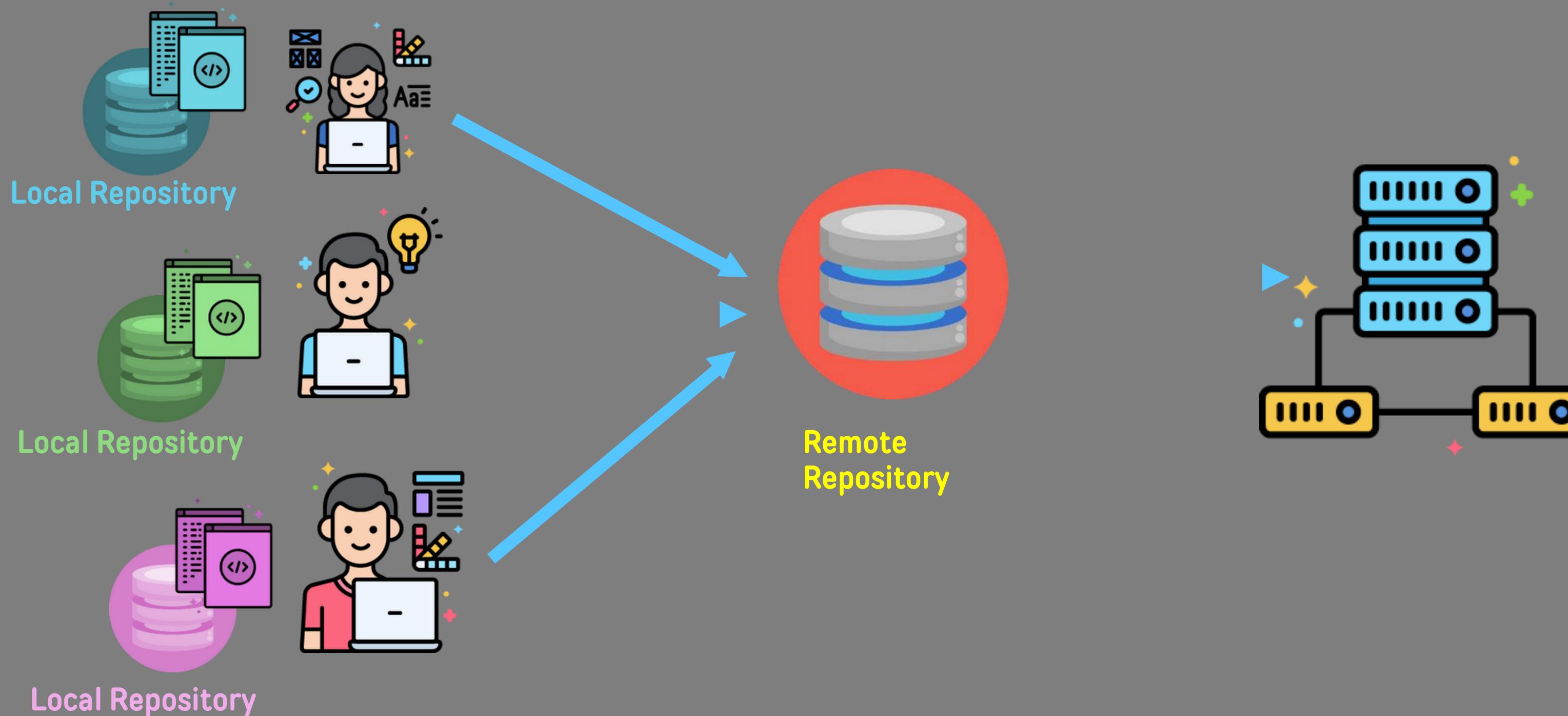
- Distributed



# Centralized Version Control System

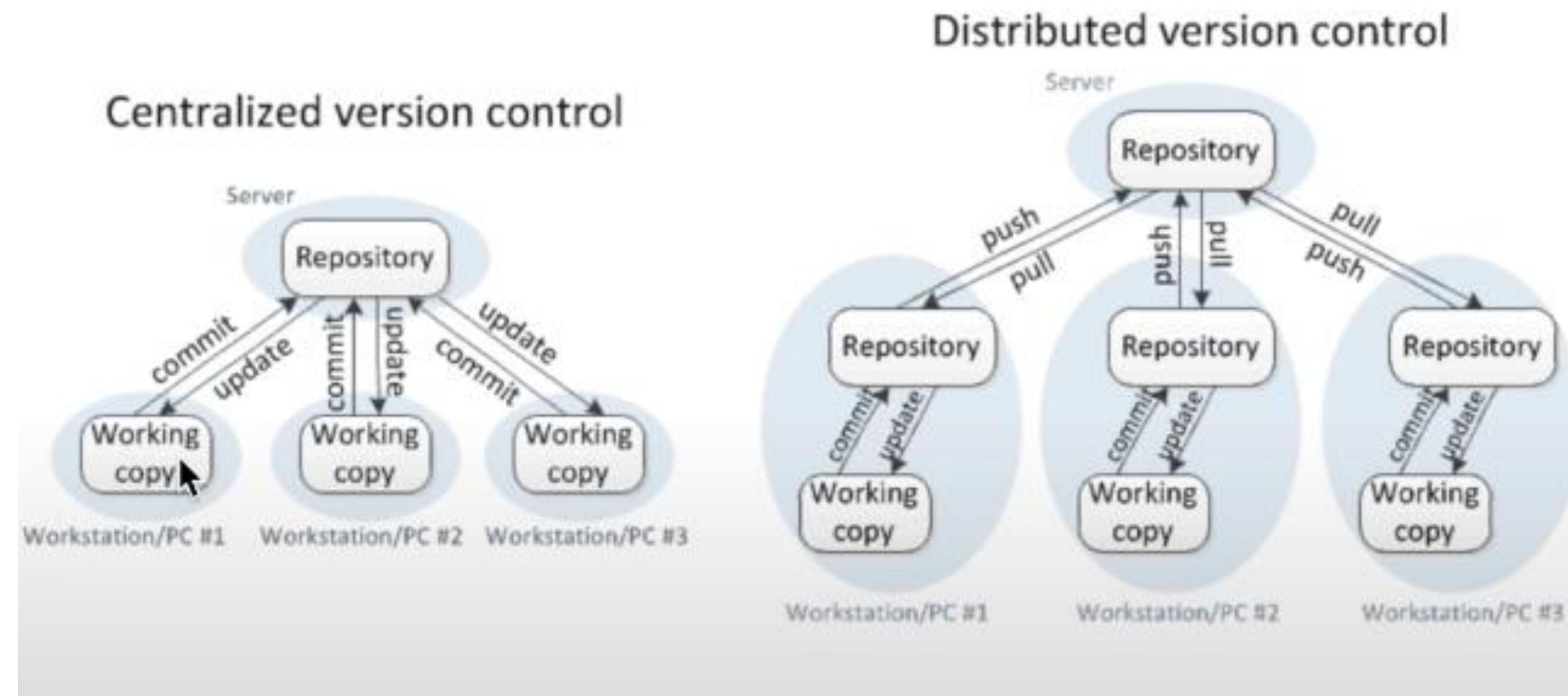


# Distributed Version Control System

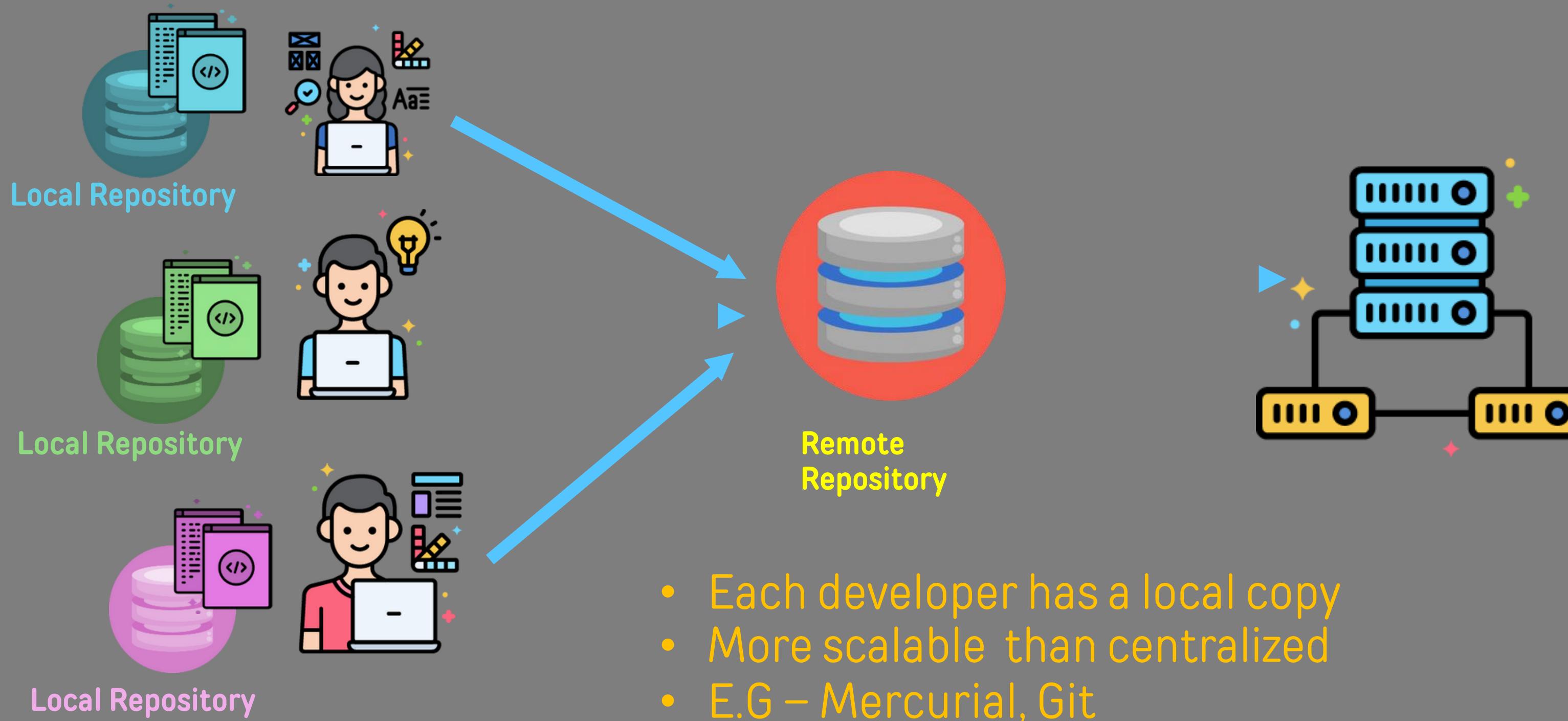


# Centralized vs Distributed

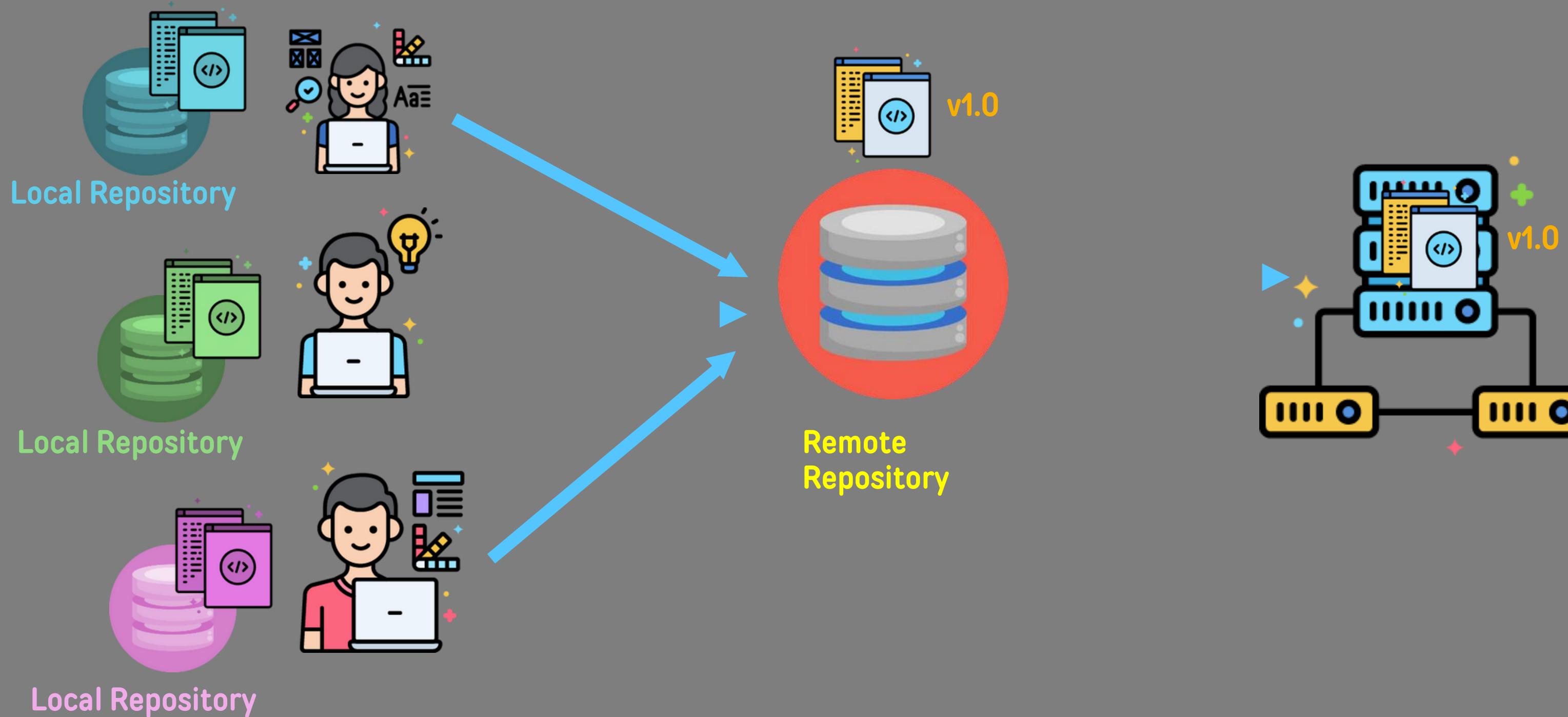
## Version Control



# Distributed Version Control System



# Distributed Version Control System



# Version Control System

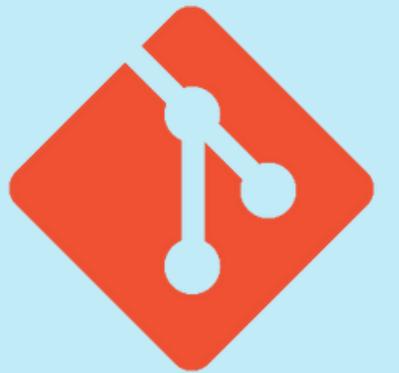


## Why Git?

- 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

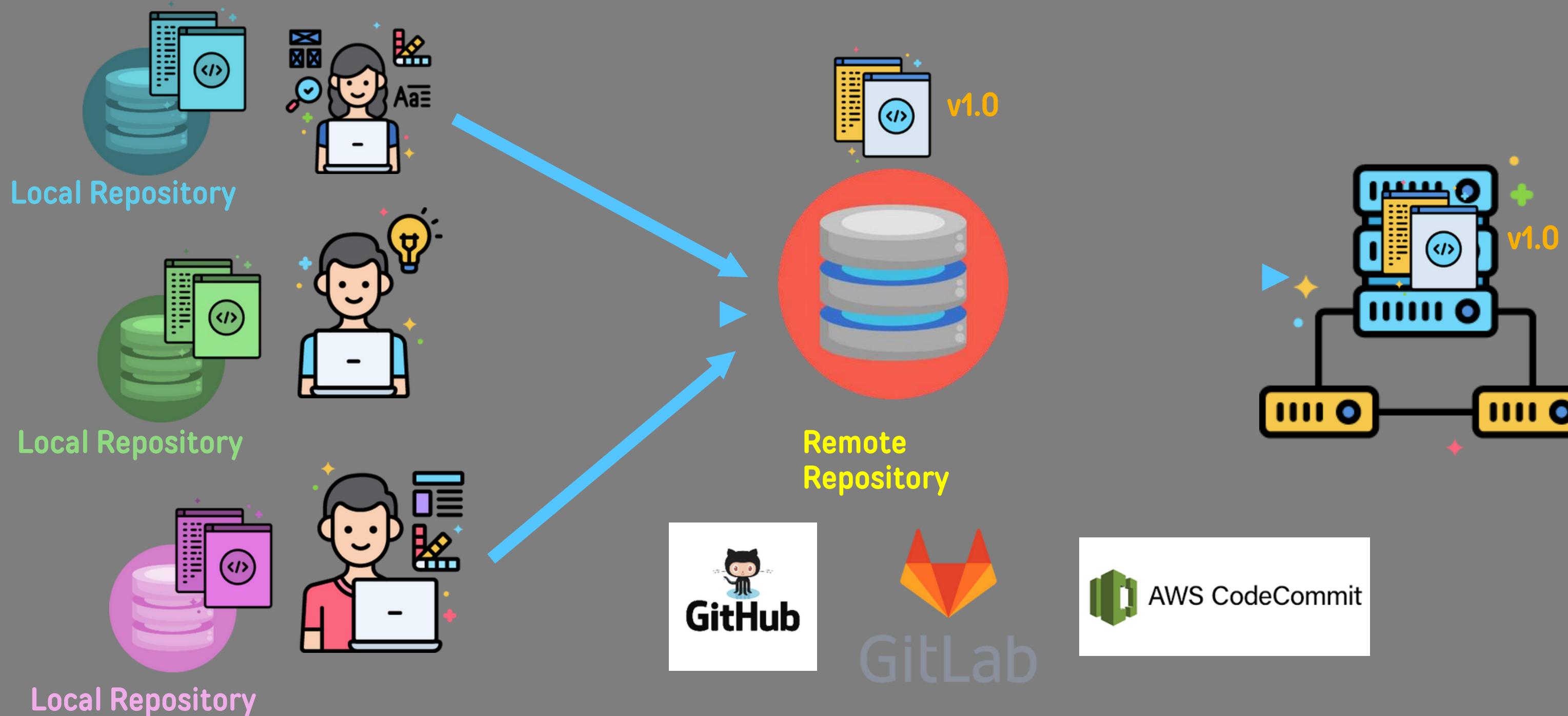
# Git vs GitHub

# Git Vs. GitHub



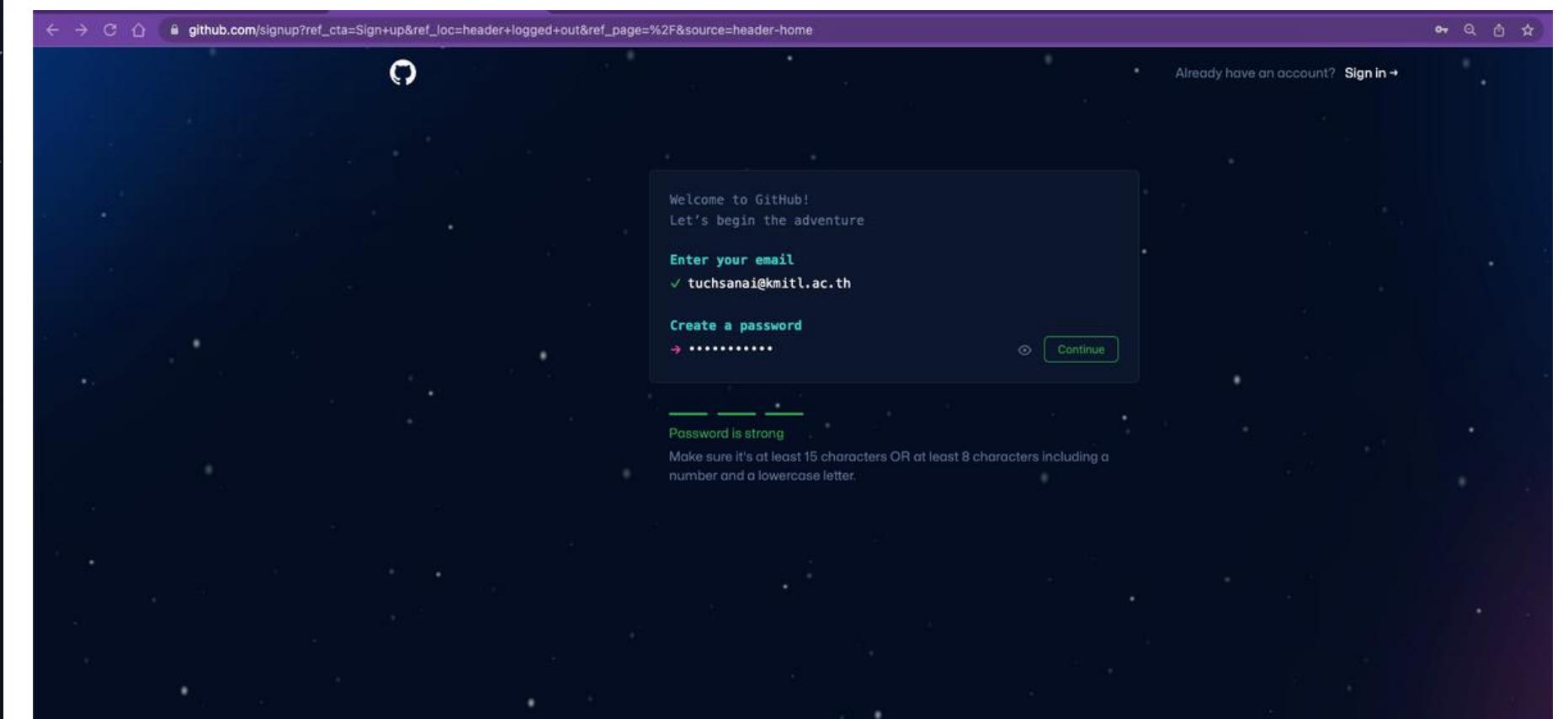
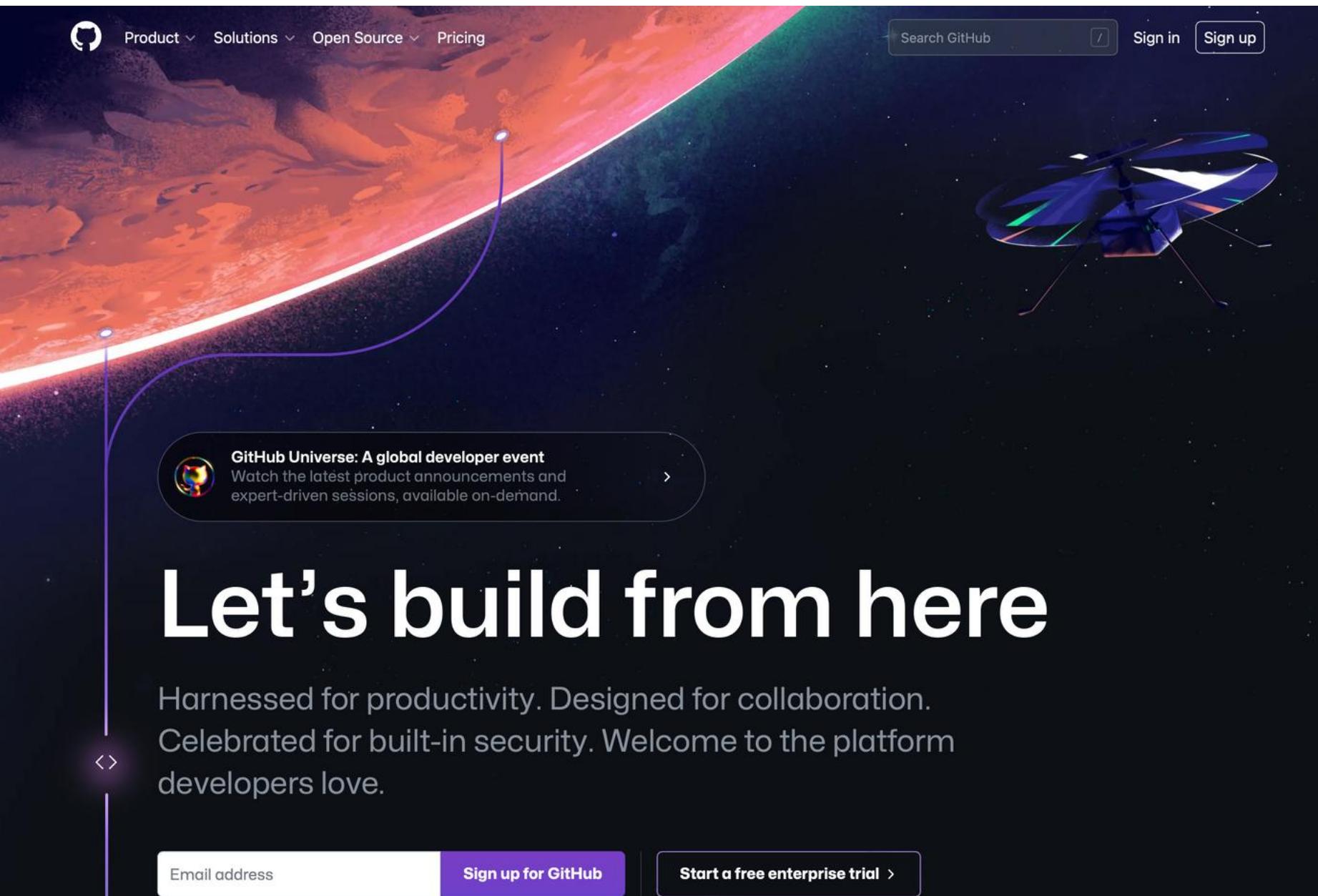
- Version Control System
- Installed locally on the system
- Created in 2005, by Linus Torvalds
- Open source, and used in multiple cloud repository services
- Git repository hosting services with other features
- Runs on the cloud
- Created in 2008, currently owned by Microsoft
- Not open source, have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



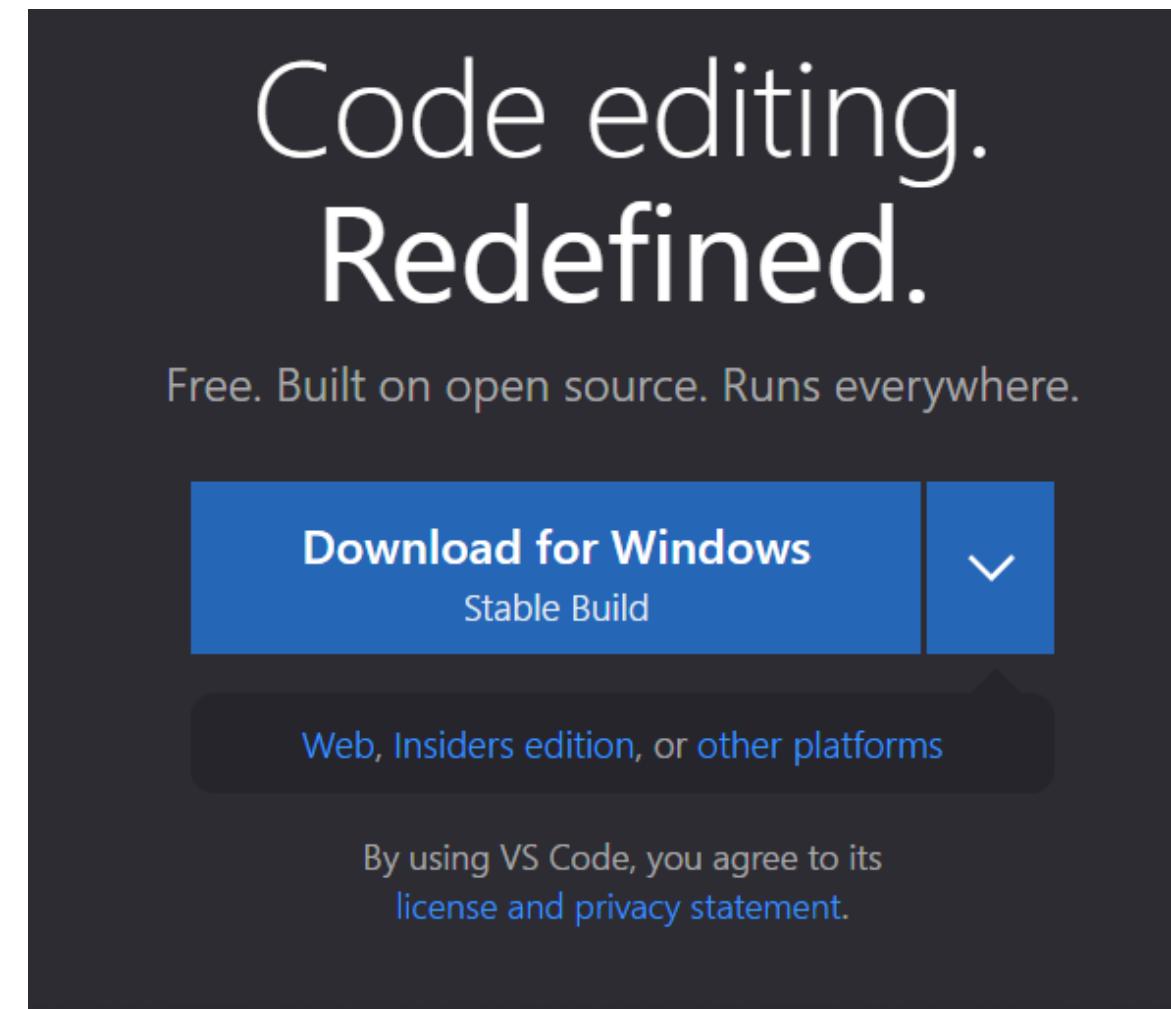
# Week 1 - Starting with Git

- **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 computer monitor displaying the latest source release '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 of the monitor, there's a section for 'Logos' with a link 'View Logos →'. At the bottom, there's a section for 'Git via Git' with a command line example: `git clone https://github.com/git/git`. The footer contains links for 'About this site' and 'Git is a member of Software Freedom Conservancy'.

# **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”**

- -- ลบ config เก่า --
  - git config --global --unset user.name
  - git config --global --unset user.email
  - git config --global --unset credential.username
- 
- -- ลบ origin คนเก่า 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.



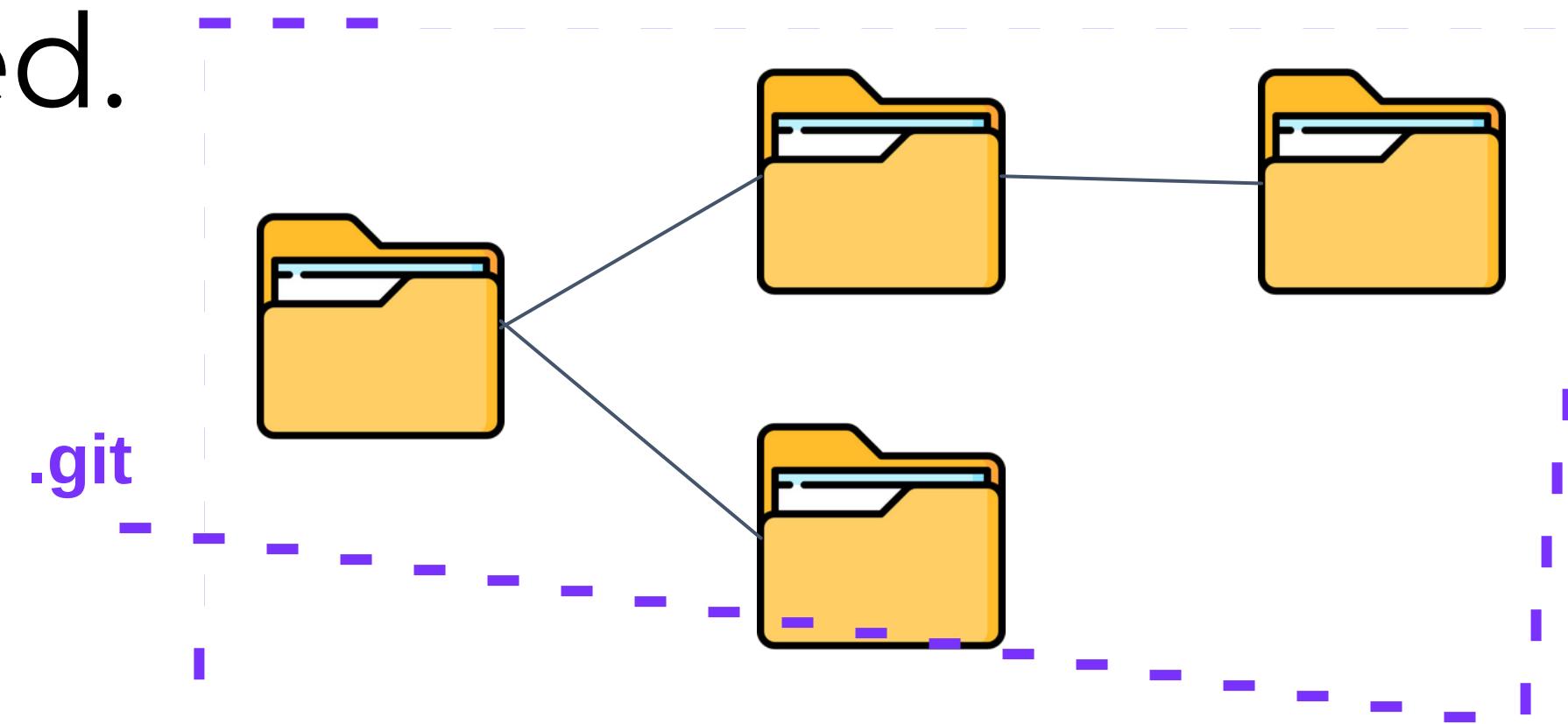
# Day 1 - Starting with Git

- 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.



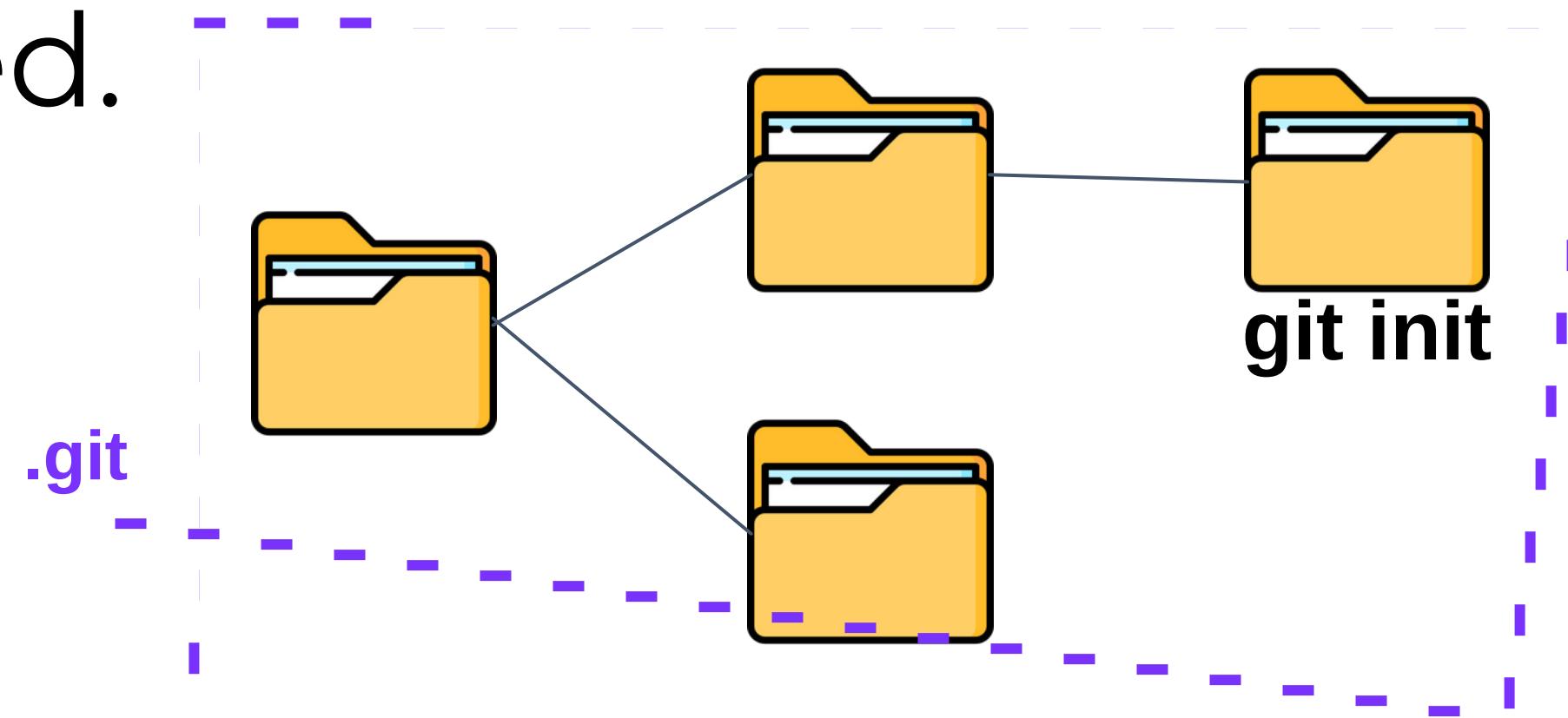
# Day 1 - Starting with Git

- 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.



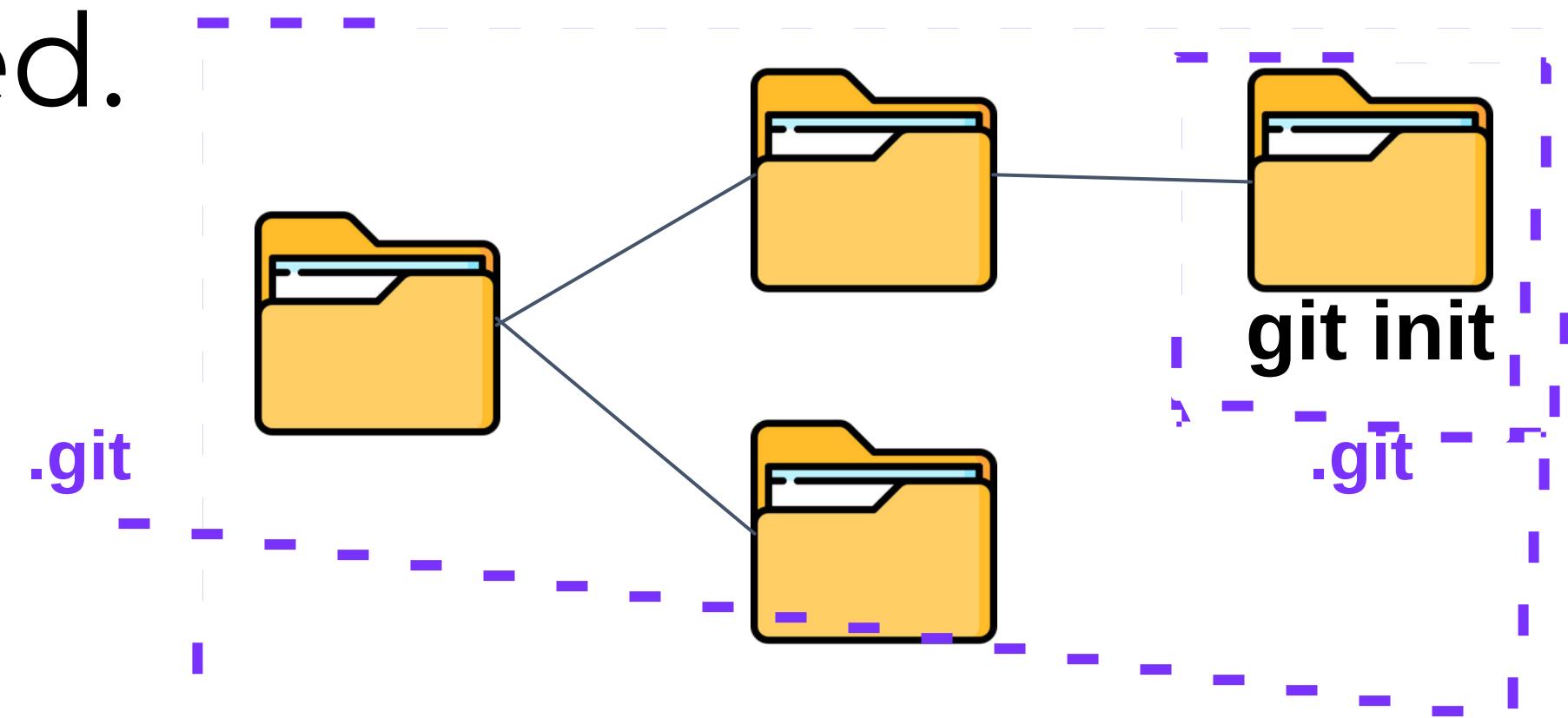
# Day 1 - Starting with Git

- 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.



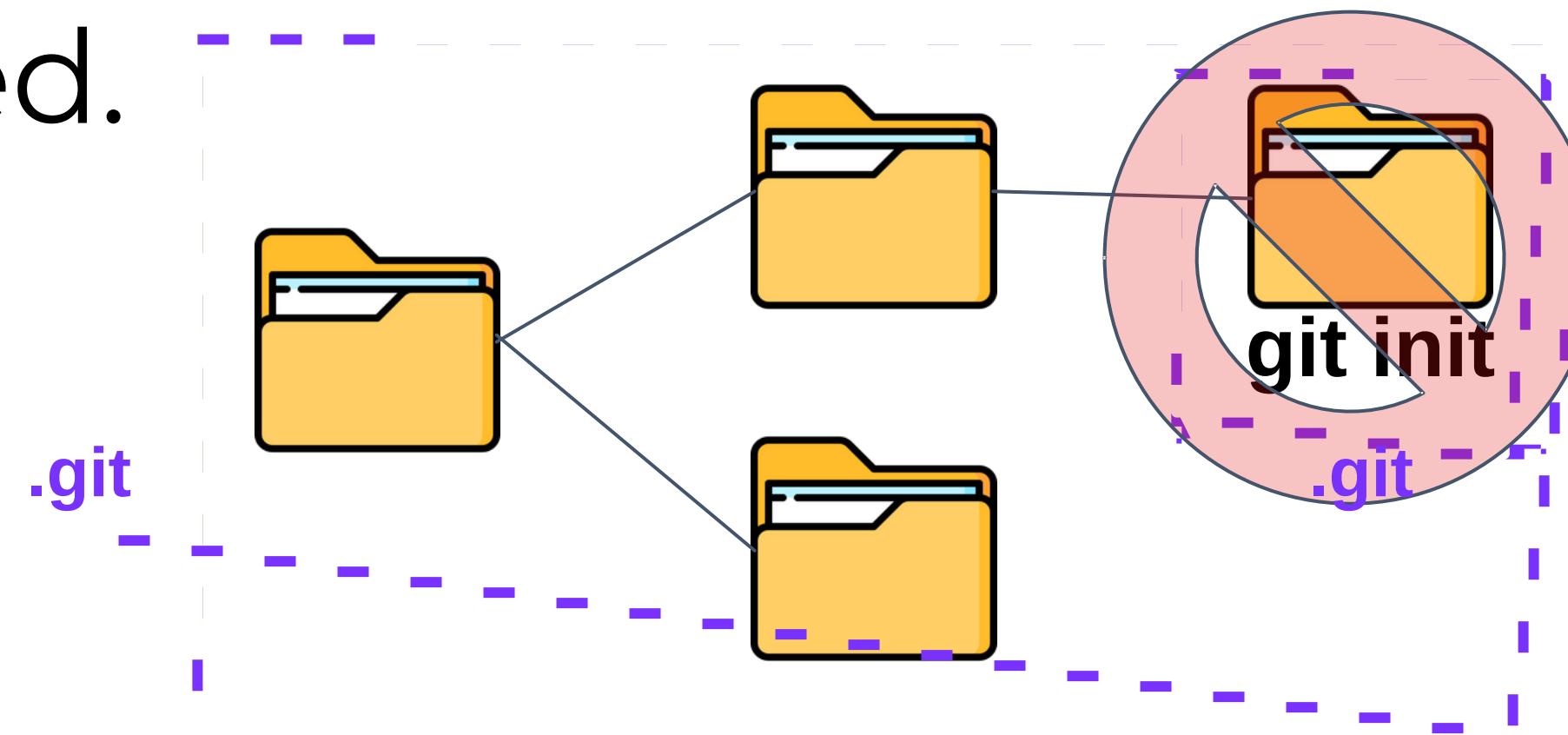
# Day 1 - Starting with Git

- 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.



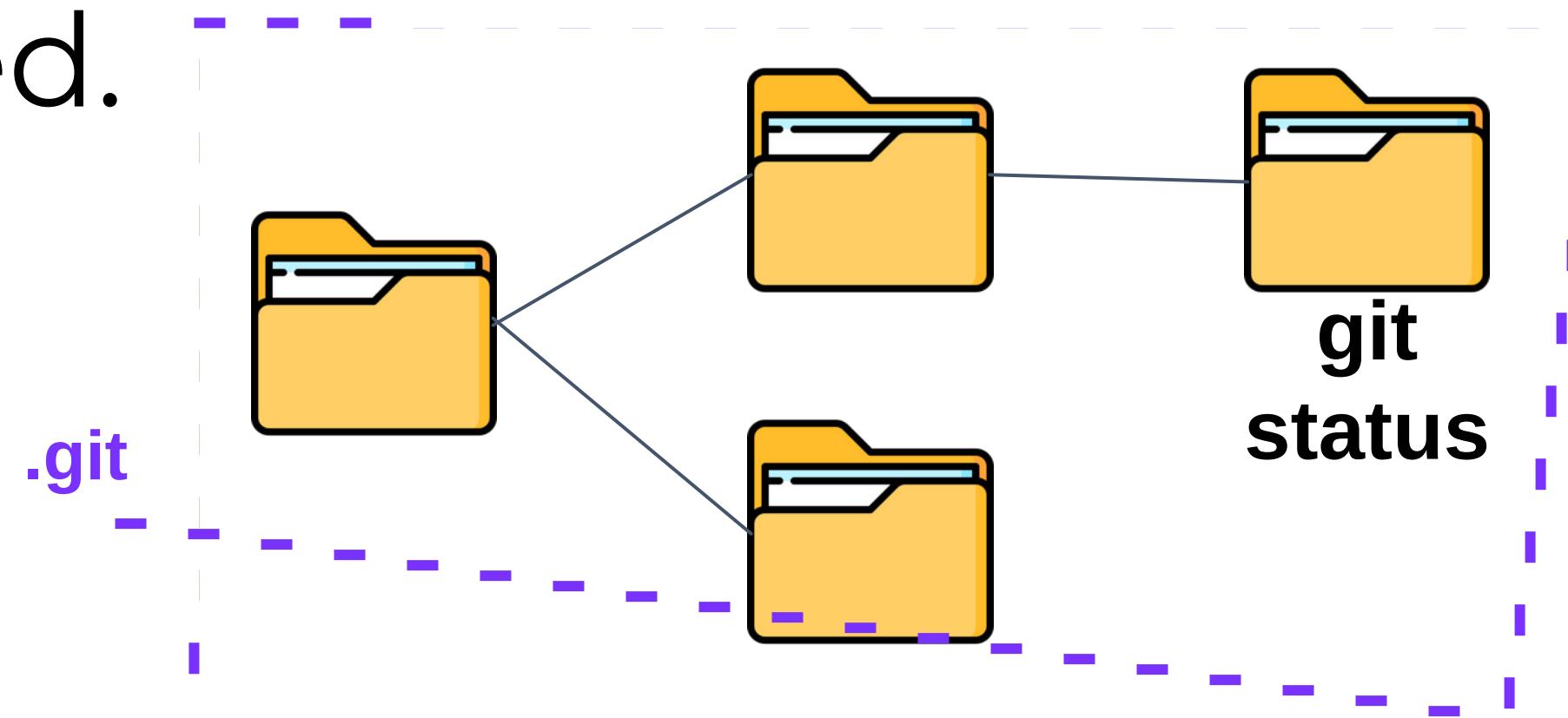
# Day 1 - Starting with Git

- 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

ติดตั้งบน Mac, Windows, Linux, หรือแม้กระทั่ง IDE หรือภาษาการเขียนโปรแกรม

สร้าง

ชอร์สโค้ด | ลองอ่านเอกสาร คำสั่งของ คอมมานต์ไลน์ คุณ!



**DAY 1**

# **Private Repositories and Tokens**

# Day 1 - Starting with Git

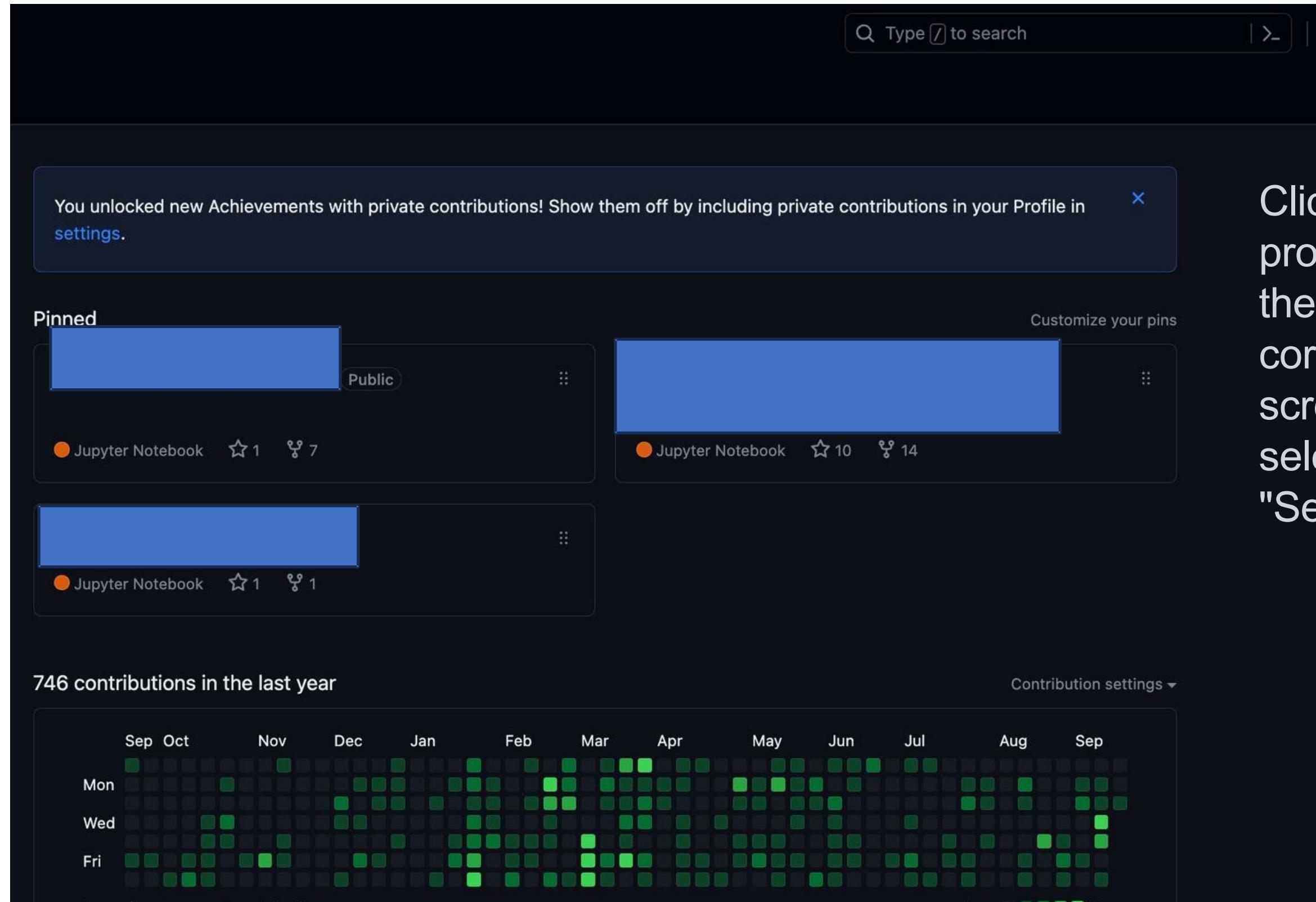
- Clone Syntax with PAT:

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

- Previously we used:

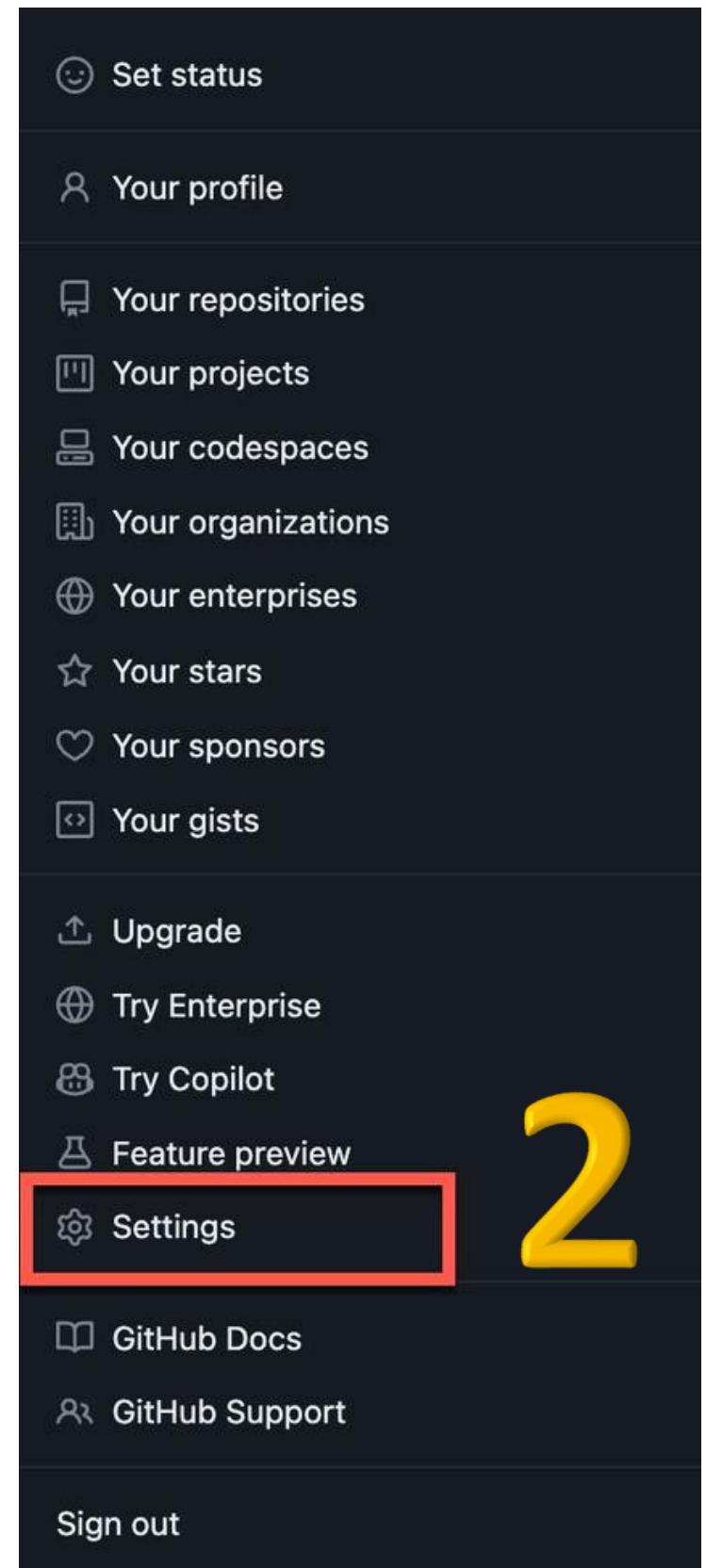
```
git clone https://github.com/account/repo.git
```

# Create a Personal Access Token



1

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



**3** In the left sidebar, select "Developer settings."

**Public profile**

Name: tuchsanai

Profile picture: A circular image of a white cat with blue eyes.

Public email: Select a verified email to display

Bio: Tell us a little bit about yourself

Pronouns: Don't specify

URL: [Input field]

Social accounts: Link to social profile (repeated four times)

Company: [Input field]

Location: [Input field]

Display current local time:  Other users will see the time difference from their local time.

**In the left sidebar, select "Developer settings."**

Please see our [privacy statement](#) to learn more about how we use this information.

**Update profile**

Navigation sidebar:

- Your personal account
- Public profile
- Account
- Appearance
- Accessibility
- Notifications
- Billing and plans
- Emails
- Password and authentication
- Sessions
- SSH and GPG keys
- Organizations
- Enterprises
- Moderation
- Planning, automation
- Repositories
- Codespaces
- Packages
- Copilot
- Pages
- Saved replies
- Code security and analysis
- Applications
- Scheduled reminders
- Security log
- Sponsorship log
- Developer settings

**4**

**Settings / Developer Settings**

GitHub Apps

OAuth Apps

Personal access tokens

**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](#).

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

**5**

**Settings / Developer Settings**

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens

Tokens (classic)

**Personal access tokens (classic)**

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

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

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

**6**

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"/> <b>repo</b>	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"/> <b>repo</b>	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"/> <b>workflow</b>	Update GitHub Action workflows
<input type="checkbox"/> <b>write:packages</b>	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input type="checkbox"/> <b>delete:packages</b>	Delete packages from GitHub Package Registry
<input type="checkbox"/> <b>admin:org</b>	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"/> <b>admin:public_key</b>	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"/> <b>admin:repo_hook</b>	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"/> <b>admin:org_hook</b>	Full control of organization hooks
<input type="checkbox"/> <b>gist</b>	Create gists
<input type="checkbox"/> <b>notifications</b>	Access notifications
<input type="checkbox"/> <b>user</b>	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)

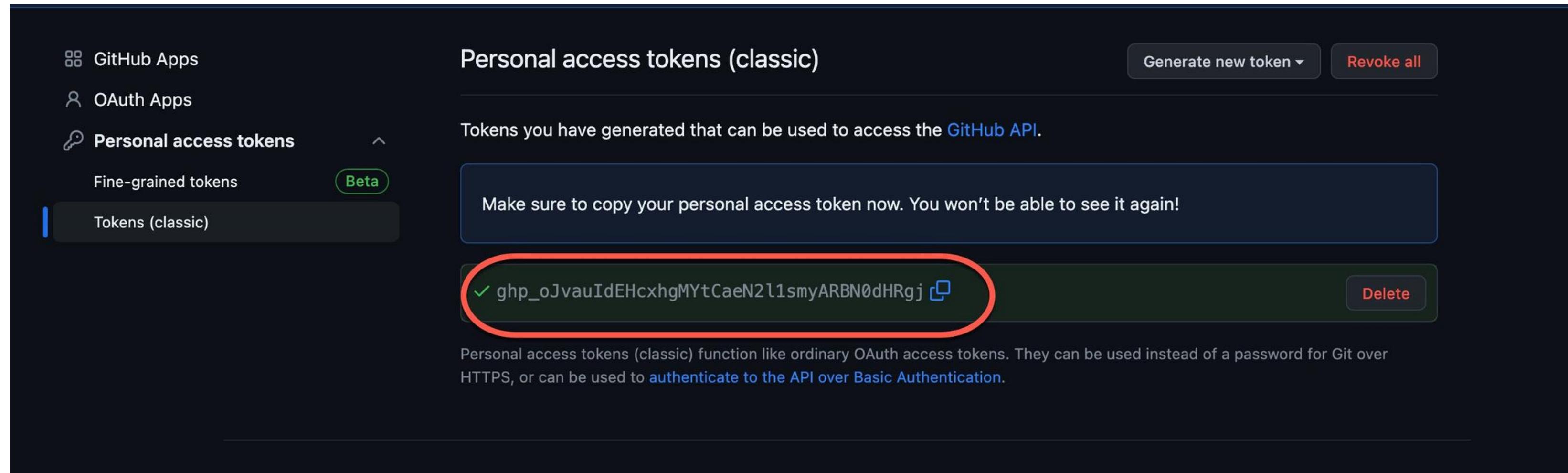
Generate new token ▾ Revoke all

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!

✓ ghp\_oJvauIdEHcxhgMYtCaeN2l1smyARBN0dHRgj ⚡ 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](#).



# **LAB 1.0**

# 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 .

The image shows two side-by-side screenshots of the GitHub Dashboard. A large red '1' is drawn over the top right corner of the left screenshot, and a large red '2' is drawn over the top right corner of the right screenshot.

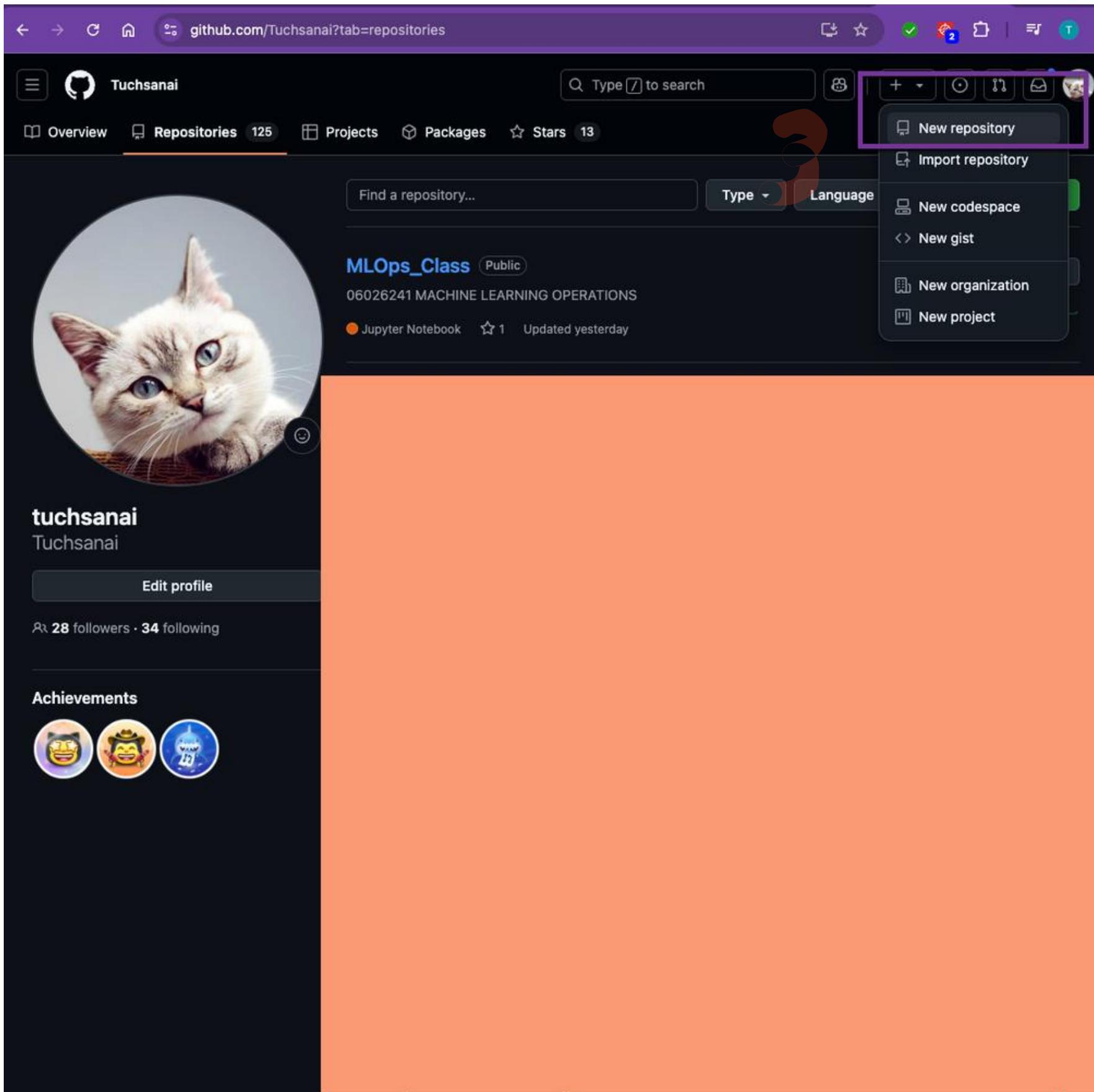
**Left Screenshot (User: llSourcell):**

- Top Bar:** Shows the GitHub logo, dashboard link, search bar, and user profile icon.
- Dashboard Area:**
  - Top repositories:** Shows a list of repositories including `Tuchsanai/DevTools`, `Tuchsanai/AIMaster-seagate-training-2024`, `Tuchsanai/DL-FOR-COMPUTER-VISION`, `Tuchsanai/DGX_2024`, `Tuchsanai/tp_idea1_2024`, `Tuchsanai/Machine_Learning`, and `Tuchsanai/pytorch_docker`.
  - Join GitHub Education!** Pop-up window:
    - Text:** 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.
    - Section:** Free and discounted services for teachers and students.
      - Copilot:** Turn natural language prompts into coding suggestions.
      - Heroku:** Build, run, and operate applications entirely in the cloud.
      - Microsoft Azure:** Access to Microsoft Azure cloud services and learning resources.
    - Join GitHub Education** button.
  - Home Feed:** Shows activity from `llSourcell` and `ALucek`, including repository publicizations and code contributions.

**Right Screenshot (User: Tuchsanai):**

- Top Bar:** Shows the GitHub logo, dashboard link, search bar, and user profile icon.
- Dashboard Area:**
  - Top repositories:** Shows a list of repositories including `Tuchsanai/DevTools`, `Tuchsanai/AIMaster-seagate-training-2024`, `Tuchsanai/DL-FOR-COMPUTER-VISION`, `Tuchsanai/DGX_2024`, `Tuchsanai/tp_idea1_2024`, `Tuchsanai/Machine_Learning`, and `Tuchsanai/pytorch_docker`.
  - Join GitHub Education!** Pop-up window:
    - Text:** 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.
    - Section:** Free and discounted services for teachers and students.
      - Copilot:** Turn natural language prompts into coding suggestions.
      - Heroku:** Build, run, and operate applications entirely in the cloud.
      - Microsoft Azure:** Access to Microsoft Azure cloud services and learning resources.
    - Join GitHub Education** button.
  - Home Feed:** Shows activity from `llSourcell` and `ALucek`, including repository publicizations and code contributions.
- Right Sidebar:** Shows the user profile information and a list of links:
  - User Profile:** Tuchsanai, tuchsanai
  - Links:**
    - Set status
    - Your profile (highlighted with a green border)
    - Your repositories (highlighted with a green border)
    - 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 \*

Repository name \*  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

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

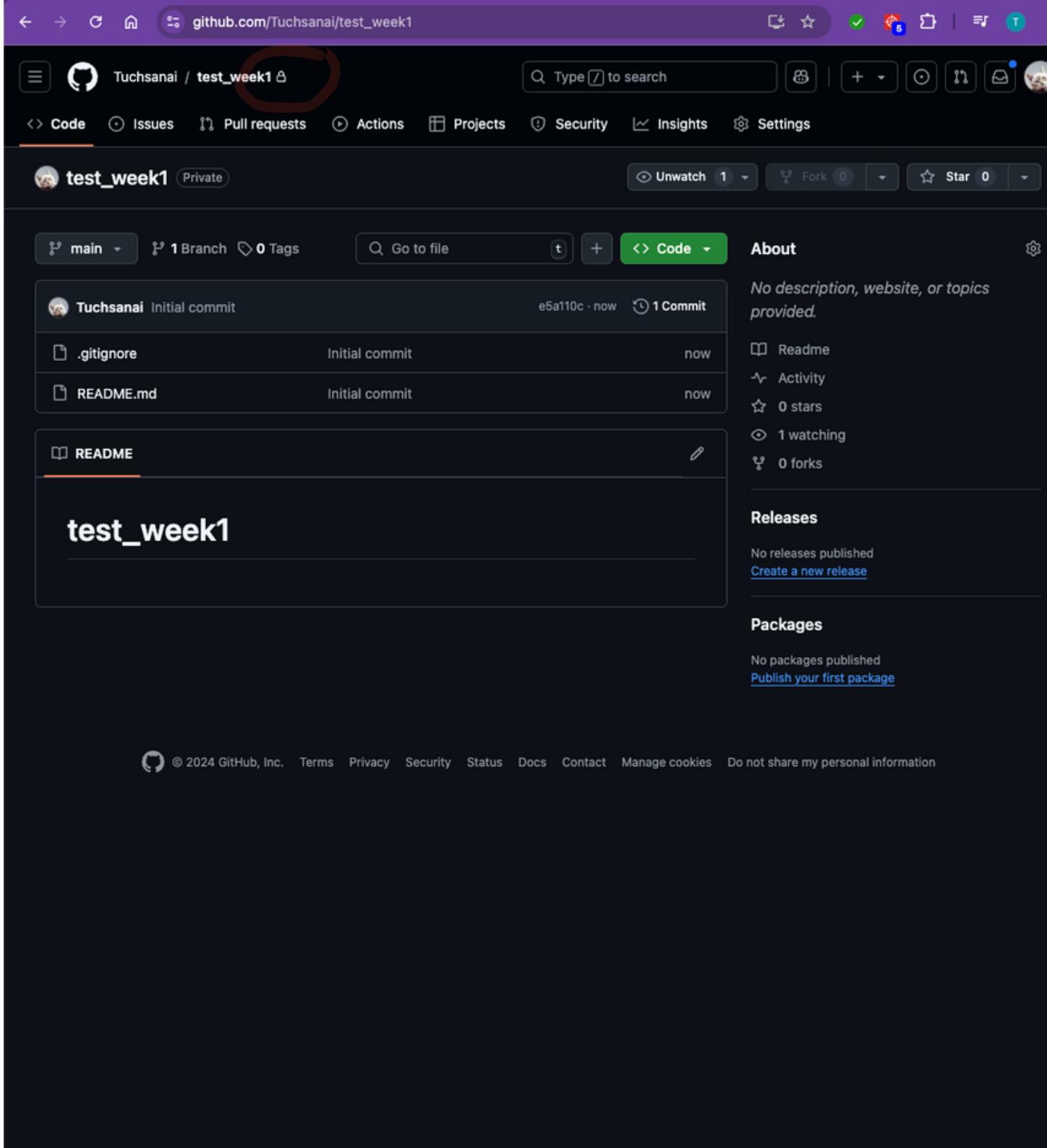
Choose a license

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

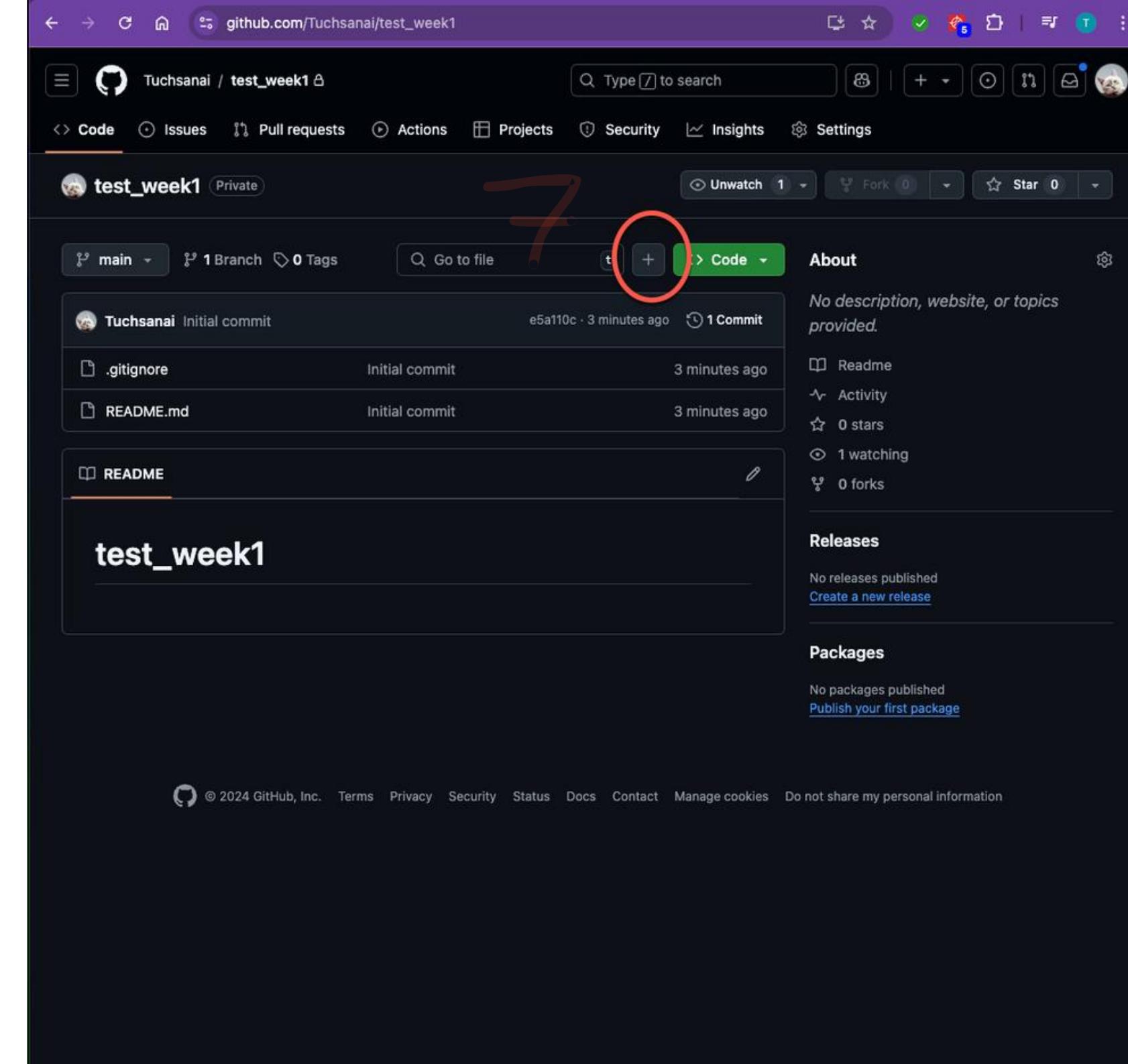
This will set `main` as the default branch. Change the default name in your [settings](#).

ⓘ You are creating a private repository in your personal account.

# Privated repository



A screenshot of a GitHub repository page for 'test\_week1'. The repository is private, as indicated by the 'Private' badge. The main navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. The repository name 'test\_week1' is circled in red at the top left. The page displays a single commit from 'Tuchsanai' with the message 'Initial commit' and timestamp 'e5a110c · now'. It also shows files like '.gitignore', 'README.md', and 'README'. The 'Code' tab is selected. The footer contains standard GitHub links and copyright information.



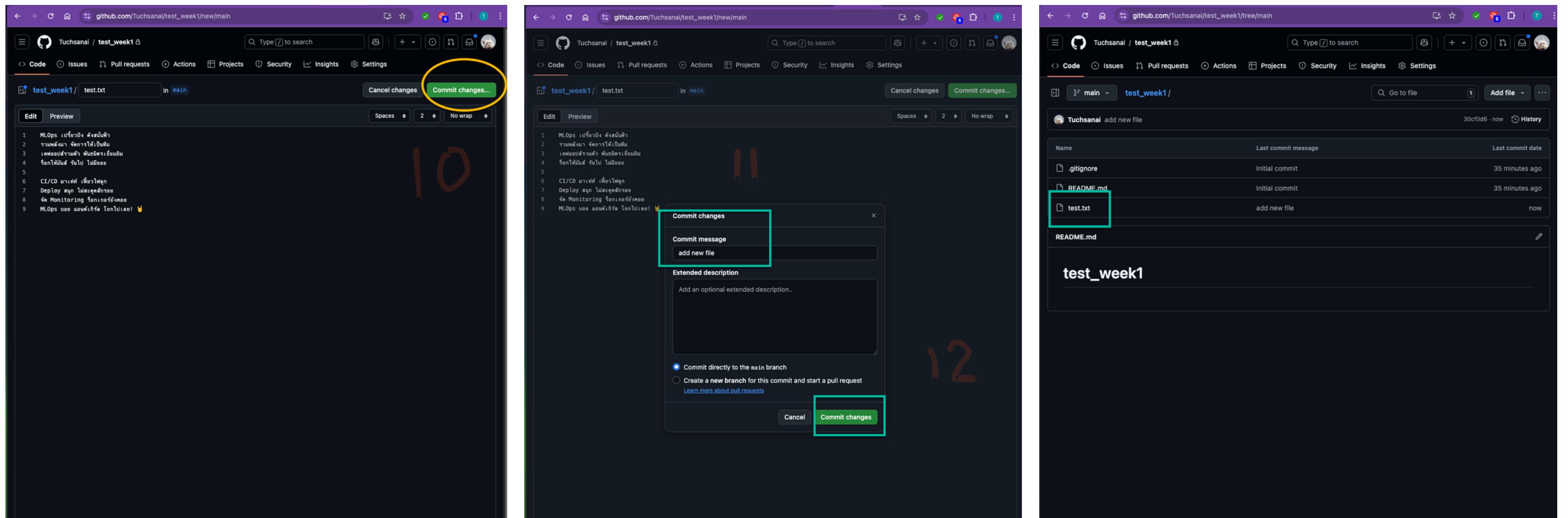
A screenshot of the same GitHub repository page for 'test\_week1' after a refresh. The repository name is circled in red at the top left. The commit from 'Tuchsanai' is now dated 'e5a110c · 3 minutes ago'. The 'Code' tab is highlighted with a red circle. The rest of the page content, including the file list and footer, remains identical to the first screenshot.

# สร้าง file : test.txt

A screenshot of a GitHub repository page for 'test\_week1'. The repository is private and has one branch named 'main'. The 'Code' tab is selected. A tooltip for the '+ Create new file' button is displayed, containing the text: 'description, website, or topics provided.' Below the repository details, there are sections for 'Releases' and 'Packages', both of which are currently empty.

A screenshot of the 'test.txt' file editor in GitHub. The file contains the following text:

```
1 MLOps เป็นวิชาที่ต้องนั่งฟัง  
2 รวมเพลิงมา จัดการให้เป็นพื้น  
3 เดฟโอล์ฟาร์มด้วย พัฒนา มีตัวชี้วัด  
4 รีโอกให้มันดี วนไป ไม่มีผลอย  
5  
6 CI/CD มาเพื่อ เพื่อไฟล์  
7 Deploy สมูก ไม่แพคูลส์กรอย  
8 จัด Monitoring รีโอกเกอร์ยังคงอย  
9 MLOps ของ แอนด์เกิร์ด โยกไปเลย!
```



- Clone private repository:

**git clone https://username:YOUR\_TOKEN@github.com/username/repo.git**