

Working with Git.



Loïc Roldán Waals
Manager Data & AI
+31 6 43 92 62 84
loic.roldanwaals@sia-partners.com

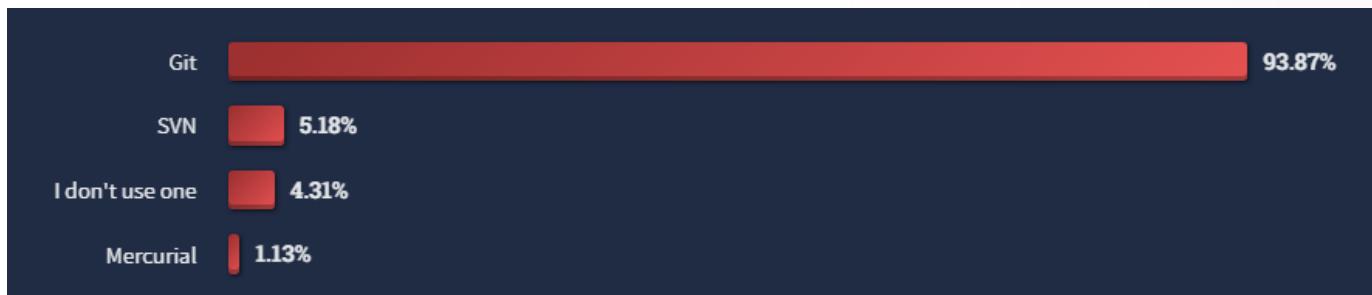
Agenda

1. Context
2. Setup
3. Commands
4. PRs
5. Merge Conflicts
6. Practice



Git was created for version control while working on Linux

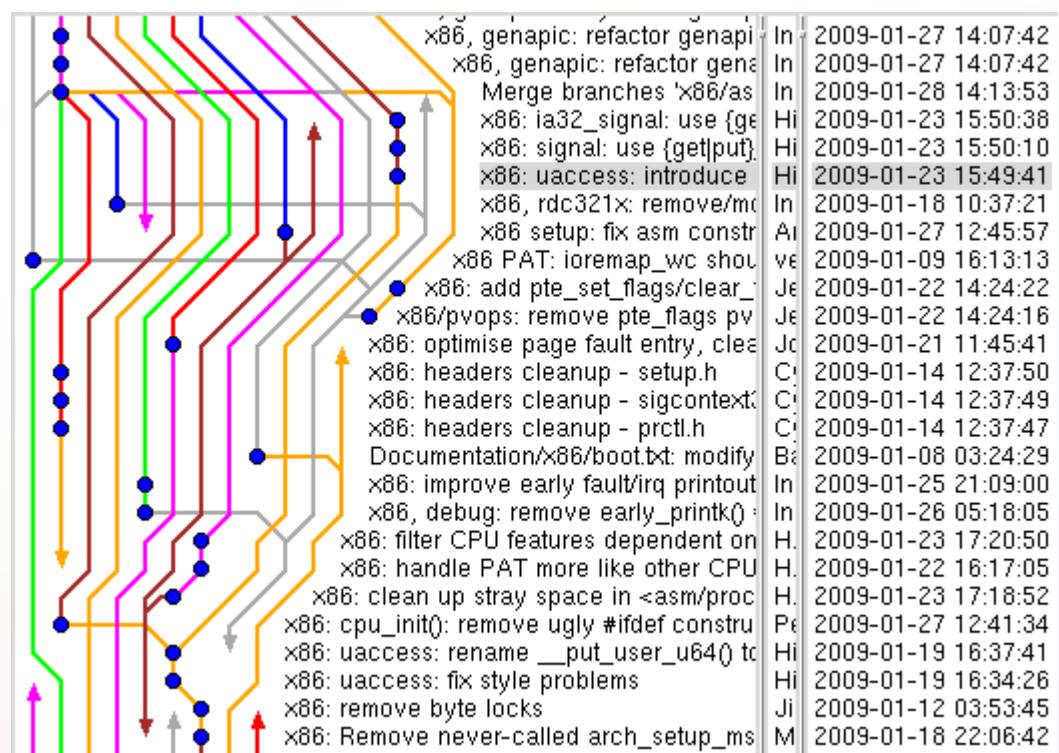
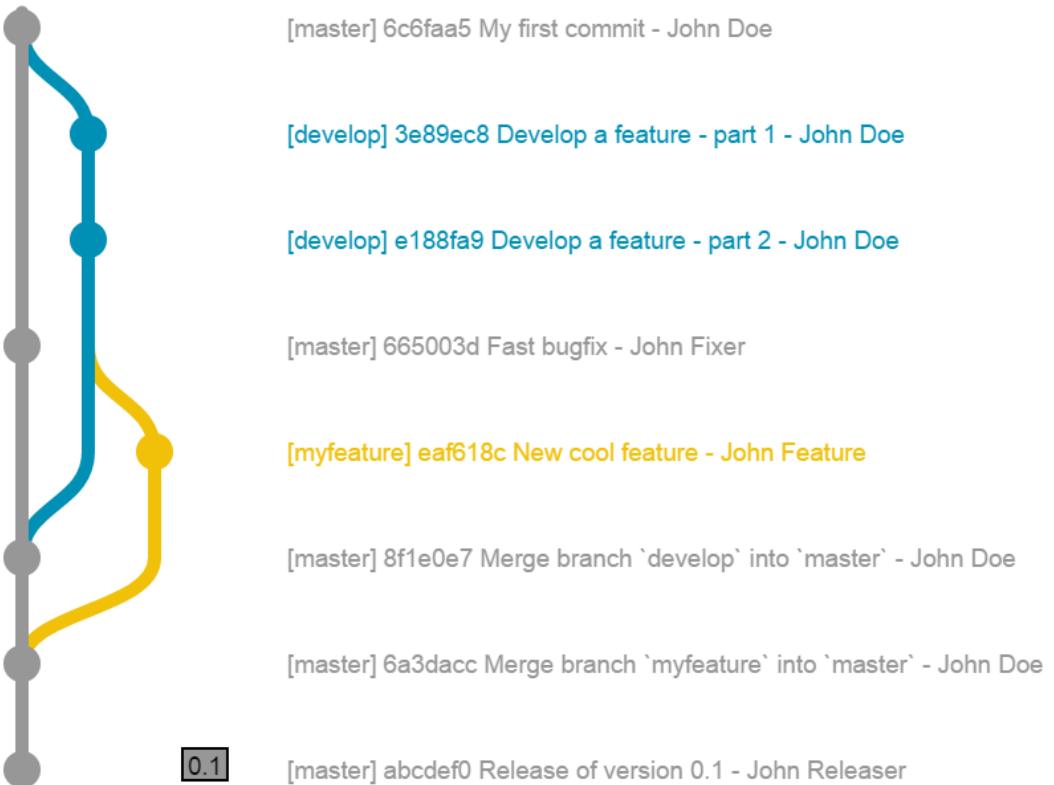
- ✓ We use Git to version control our code in a scalable way.
- ✓ Of the three most common version control systems, Git is (by far) the most popular



Popularity of version control systems in 2023 ([link](#))



Git was created for version control while working on Linux



To use Git, a hosting tool and an access tool is required

Hosting tools

- Central place to host the repository
- Can be accessed by all collaborators
- Typical examples include:



GitHub



Azure DevOps



GitLab



Bitbucket

Access tools

- Way to interact with the host repository and perform local version control
- Often included with an IDE, but can be stand-alone
- Typical examples include:



Command Line



Github Desktop



VS Code



PyCharm

Changing some configurations will make it easier to work with git.



VS Code

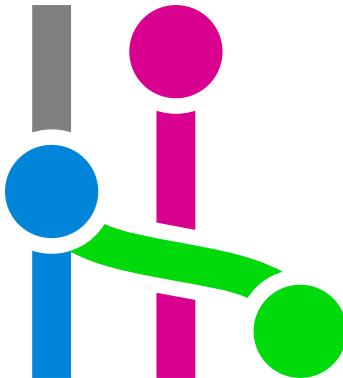
- / For VSCode settings, use Ctrl + Shift + P and write "open default settings (JSON)".
- / Choose the JSON option and paste the provided settings.
- / Install Git Graph and GitLens for nicer GUI



Git

- / Open a terminal
- / Enter the commands:
 - / git config --global core.editor "code -wait"
 - / git config --global -e
- / Then paste the provided example settings.
- / Close the file to save it.

VSCode has nice extensions that allow for a GUI interface.



Git graph

Graph	Description	Date	Author	Commit
	Uncommitted Changes (6)			
	register_device_endpoint origin Validate request parameters. Store device information	19 Nov 2019 11:43	*	*
	Create route for registering a new device	8 Jun 2019 18:22	Michael Hutchison	288b97a5
	master origin R3.0 Merge branch 'build_and_deploy_upgrades'	8 Jun 2019 18:17	Michael Hutchison	a2b39616
	build_and_deploy_upgrades origin Streamline package build scripts	8 Jun 2019 18:15	Michael Hutchison	f8fd600c
	R2.0 Merge branch 'get_status_endpoint'	9 Feb 2019 18:20	Michael Hutchison	f2965593
	Merge branch 'set_status_endpoint'	9 Feb 2019 17:57	Michael Hutchison	22a32a90
	get_status_endpoint origin Retrieve and return device status	9 Feb 2019 17:56	Michael Hutchison	6e3f85b1
	set_status_endpoint origin Documentation of set device status endpoint	9 Feb 2019 17:48	Michael Hutchison	eaddb8a9
	Validate request parameters	9 Feb 2019 17:39	Michael Hutchison	7d8f24b2
	Create get device status route	9 Feb 2019 17:33	Michael Hutchison	7051d755
	Store status, and set response status code	9 Feb 2019 17:24	Michael Hutchison	6e40d17a
	Create set device status route	9 Feb 2019 17:14	Michael Hutchison	d81a61eb
	R1.0 Merge branch 'device_connect_endpoint'	9 Feb 2019 17:07	Michael Hutchison	5cede530
	device_connect_endpoint origin Documentation of device connect and list endpoints	9 Feb 2019 17:04	Michael Hutchison	e430637e
	Merge branch 'device_connect_endpoint'	9 Feb 2019 17:00	Michael Hutchison	f86e7a01
	Store and return device connection session	9 Feb 2019 16:53	Michael Hutchison	6dbd0e99
	Merge branch 'device_list_endpoint'	9 Feb 2019 16:52	Michael Hutchison	222b0bcb
	device_list_endpoint origin Return list of devices	9 Feb 2019 16:40	Michael Hutchison	2b970a0e
	Establish device connection session	9 Feb 2019 16:04	Michael Hutchison	9bb98caf
	Create device list route	9 Feb 2019 15:58	Michael Hutchison	bcd817ec
		9 Feb 2019 15:43	Michael Hutchison	2bcd203d

VSCode has nice extensions that allow for a GUI interface.



GitLens

GitLens Interactive Rebase ➔ main Rebasing 6 commits onto ⚡ f7e60f7

⋮	squash ▾	Adds new capabilities to the new feature	You	3 weeks ago	9536e5f
⋮	pick ▾	Refactors the new feature to improve perf...	You	3 weeks ago	5ba75f1
⋮	drop ▾	Fixes off-by-one in the new feature	You	3 weeks ago	58e6de4
⋮	pick ▾	Adds a new feature	You	3 weeks ago	be6482b
⋮	edit ▾	Adds exports for api	You	a week ago	c41265e
⋮	pick ▾	Supercharged	You	4 years ago	1be008a
⋮		Hello GitLens	You	4 years ago	f7e60f7

p Pick r Reward e Edit s Squash d Drop alt ↑ Move Up alt ↓ Move Down

DISABLE REBASE EDITOR
WILL ABORT REBASE

START REBASE
CTRL+ENTER

ABORT
CTRL+A

Most of your Git workflow can be achieved with 6 commands



pull or fetch

Get changes from the remote. Pull also applies your changes locally.



add and commit

Choose what parts you want to add to the “timeline” of the repo.



(force) push

Update the remote “timeline” with your local “timeline” for a specific branch.



branch and checkout

Create a new branch in the “timeline”. Checkout switches you to your chosen branch.



stash

Save your changes locally without adding them to the “timeline”.



rebase and reset

Allows you to edit the history and which branch your branch is based on.

Let's try to use these commands! Try to complete them all.

1

Clone repo with PAT

2

Create new **branch** called *feature/loics-feature*, based on *fake-main*.

3

Make some simple code changes.

4

Stash them and switch to some other branch (*feature/someones-feature*).

5

Make a new **commit** and **push** changes.

6

Make some more simple changes.

7

Come back to the original branch (*feature/loics-feature*) and **un-stash** them.

8

Wait for some other changes on *fake-main*.

9

Rebase *feature/loics-feature* on the latest changes from *fake-main*.

10

Make multiple **commits** and “accidentally” add some with some unwanted changes.

11

Reset (--mixed) changes and create one new clean **commit**.

12

Force push changes to fix the remote.

Let's try to use these commands!

Clone repo with PAT

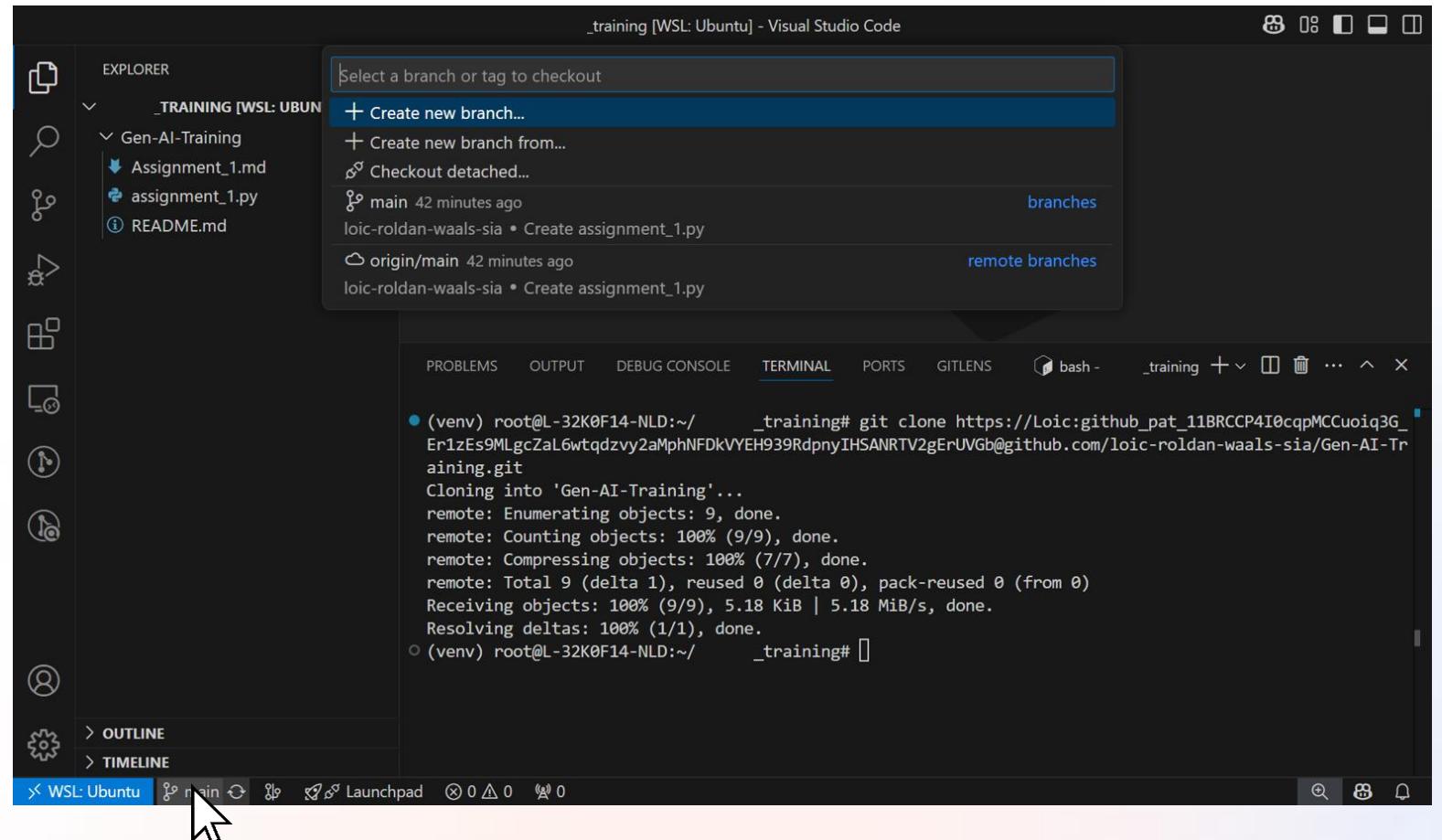
Other methods are also available, this is just the simplest to mange for us.

The screenshot shows a Visual Studio Code interface running in WSL: Ubuntu. The terminal tab is active, displaying the output of a git clone command. The command was run from a root shell (root@L-32K0F14-NLD:~/_training#) and cloned a repository from GitHub. The output shows the progress of the clone, including object enumeration, counting, compressing, and receiving objects, along with a warning about reused objects.

```
(venv) root@L-32K0F14-NLD:~/_training# git clone https://Loic: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@github.com/loic-roldan-waals-sia/Gen-AI-Training.git
Cloning into 'Gen-AI-Training'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 9 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (9/9), 5.18 KiB | 5.18 MiB/s, done.
Resolving deltas: 100% (1/1), done.
(venv) root@L-32K0F14-NLD:~/_training#
```

Let's try to use these commands!

- ✓ Create new **branch** called *feature/loics-feature*, based on *fake-main*.



_training [WSL: Ubuntu] - Visual Studio Code

Select a branch or tag to checkout

- + Create new branch...
- + Create new branch from...
- Checkout detached...
- main 42 minutes ago
loic-roldan-waals-sia • Create assignment_1.py branches
- origin/main 42 minutes ago
loic-roldan-waals-sia • Create assignment_1.py remote branches

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS bash - _training + ×

```
(venv) root@L-32K0F14-NLD:~/_training# git clone https://Loic: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@github.com/loic-roldan-waals-sia/Gen-AI-Training.git
Cloning into 'Gen-AI-Training'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 9 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (9/9), 5.18 KiB | 5.18 MiB/s, done.
Resolving deltas: 100% (1/1), done.
(venv) root@L-32K0F14-NLD:~/_training#
```

WSL: Ubuntu main Launchpad 0 △ 0 0

Let's try to use these commands!

- >Create new **branch** called *feature/loics-feature*, based on *fake-main*.

The new branch will be "feature/loics-feature"

```
(venv) root@L-32K0F14-NLD:~/_training# git clone https://Loic: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@github.com/loic-roldan-waals-sia/Gen-AI-Training.git
Cloning into 'Gen-AI-Training'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 9 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (9/9), 5.18 KiB | 5.18 MiB/s, done.
Resolving deltas: 100% (1/1), done.
(venv) root@L-32K0F14-NLD:~/_training# 
```

Let's try to use these commands!

- / Make some simple code changes.

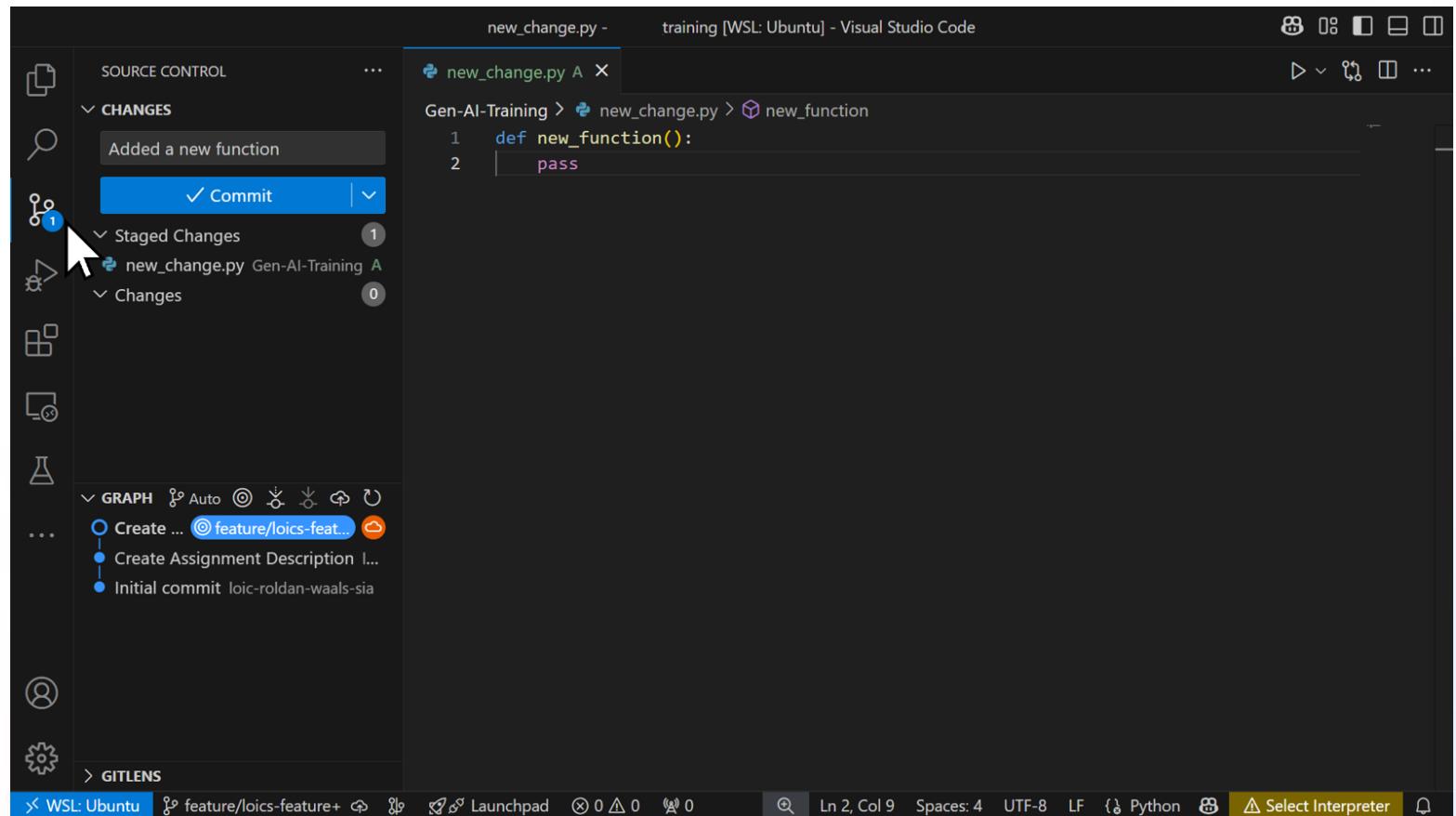
The screenshot shows a Visual Studio Code interface running in a WSL Ubuntu terminal. The Explorer sidebar on the left shows a folder named '_TRAINING [WSL: UBUNTU]' containing files: Gen-AI-Training, Assignment_1.md, assignment_1.py, new_change.py, and README.md. The new_change.py file is currently open in the main editor area. The code in the editor is:

```
1 def new_function():
2     pass
```

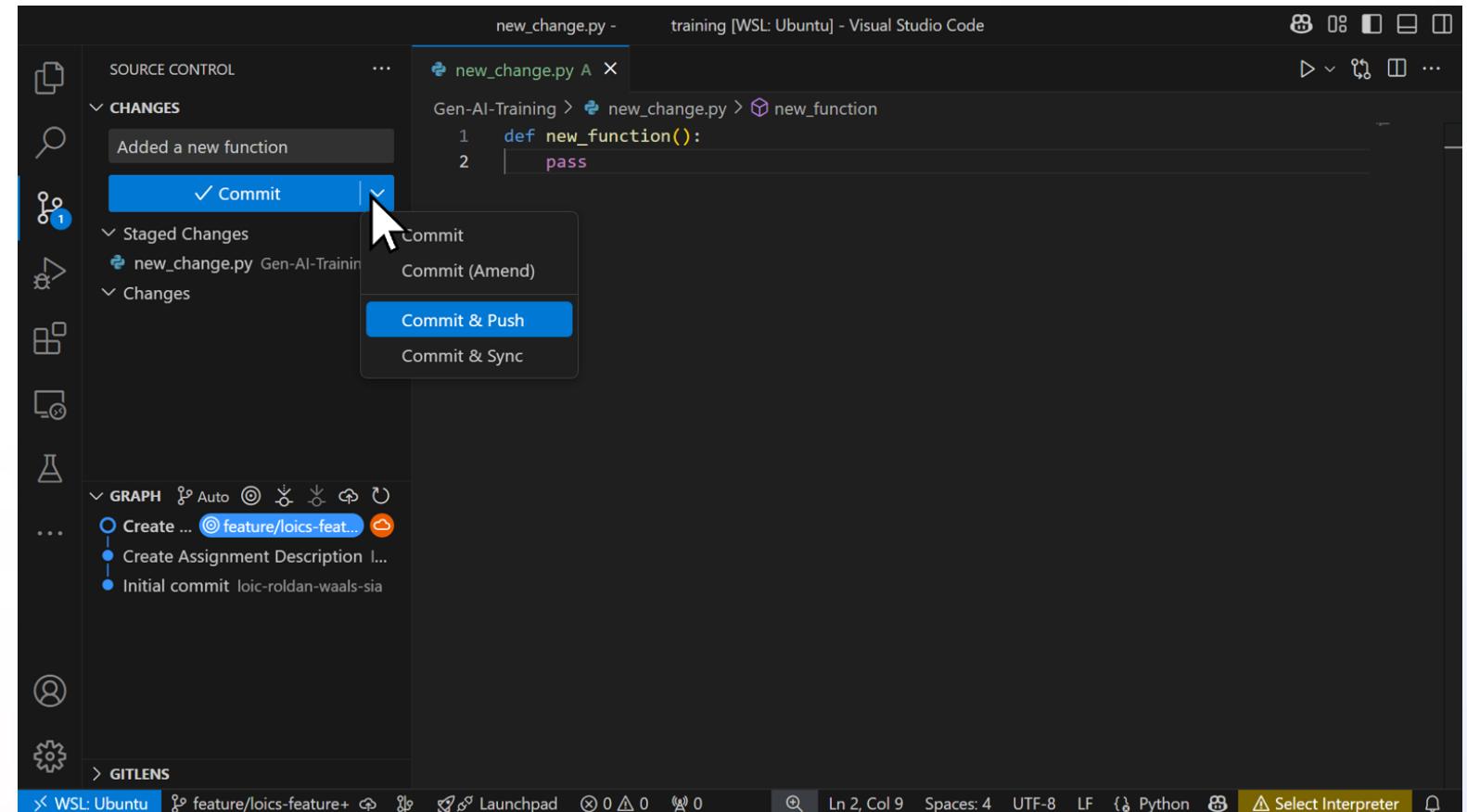
The status bar at the bottom of the screen displays the following information: 'WSL: Ubuntu', 'feature/loics-feature+', 'Launchpad', '0 △ 0', '0', 'Ln 2, Col 9', 'Spaces: 4', 'UTF-8', 'LF', 'Python', 'Select Interpreter', and a refresh icon.

Let's try to use these commands!

- / Make a new **commit** and **push** changes.



Let's try to use these commands!



/ Make a new **commit** and **push** changes.

Let's try to use these commands!

- / Make some more simple changes.

The screenshot shows the Visual Studio Code interface. On the left, the Source Control sidebar is open, showing a commit message: "Message (Ctrl+Enter to commit...)". A blue button labeled "Commit" is highlighted. Below it, there is a "Changes" section with one file listed: "new_change.py Gen-AI-Training... M". The main editor area displays a Python file named "new_change.py" with the following content:

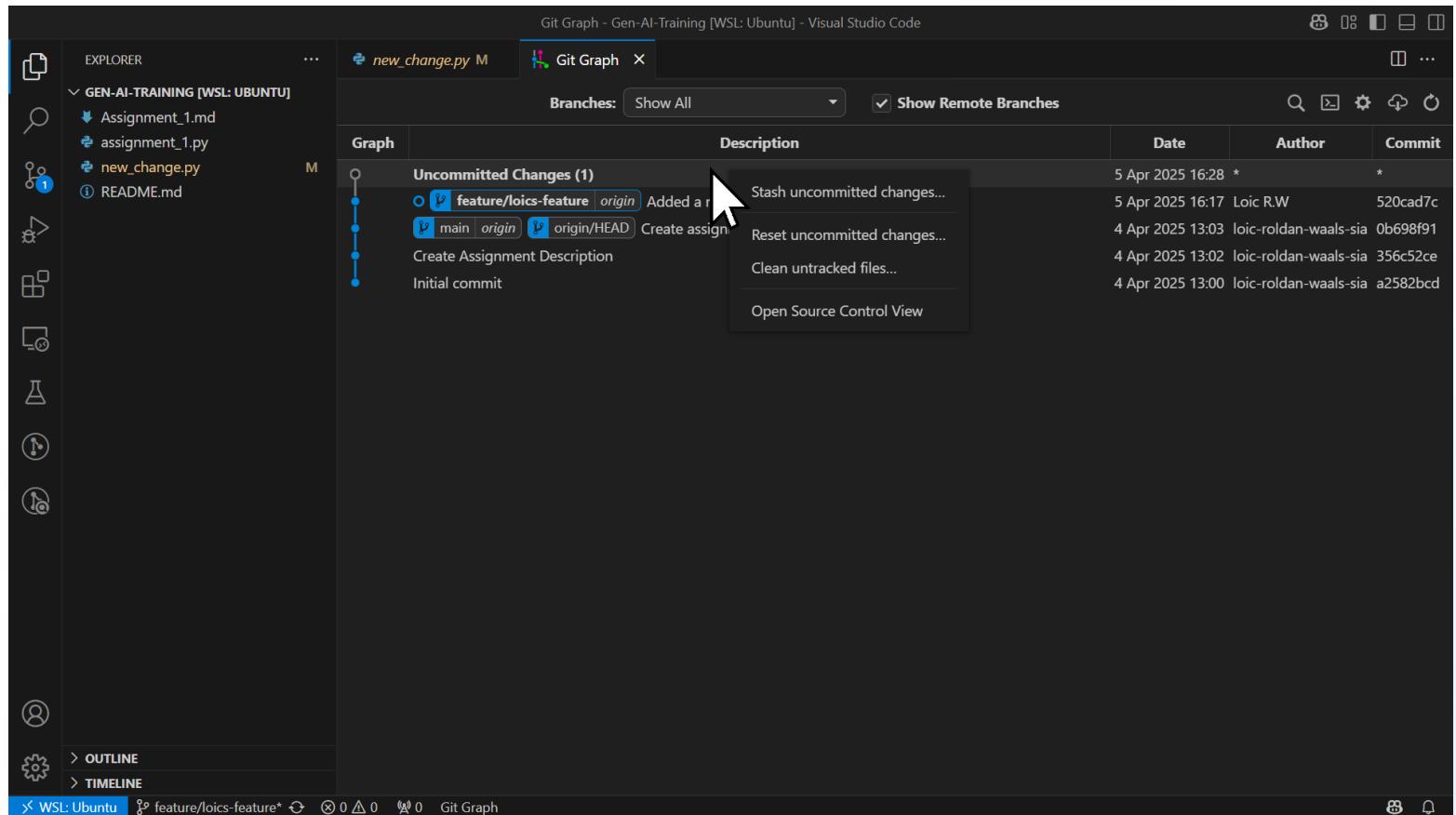
```
new_change.py - training [WSL: Ubuntu] - Visual Studio Code
new_change.py M X

Gen-AI-Training > new_change.py > new_function
You, 19 seconds ago | 2 authors (You and one other)
def new_function():
    """
    Some docstring.
    """
    pass
```

The status bar at the bottom shows the file path "WSL: Ubuntu", the file name "feature/loics-feature*", and various status icons.

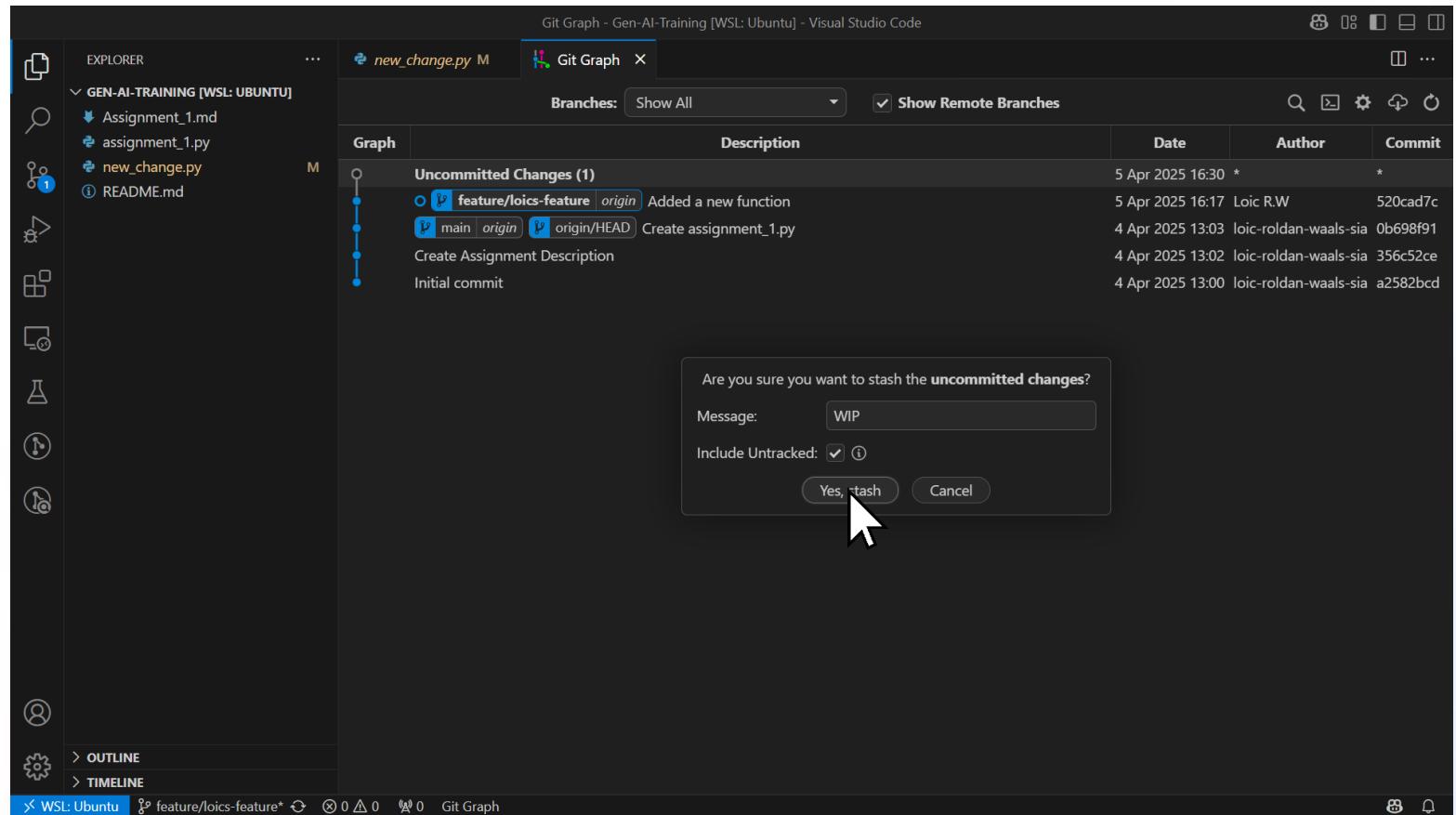
Let's try to use these commands!

- / **Stash** them and switch to some other branch (*feature/someones-feature*).
- / This is common when you are reviewing someone else's work on their branch, but you are not yet finished with your own work.



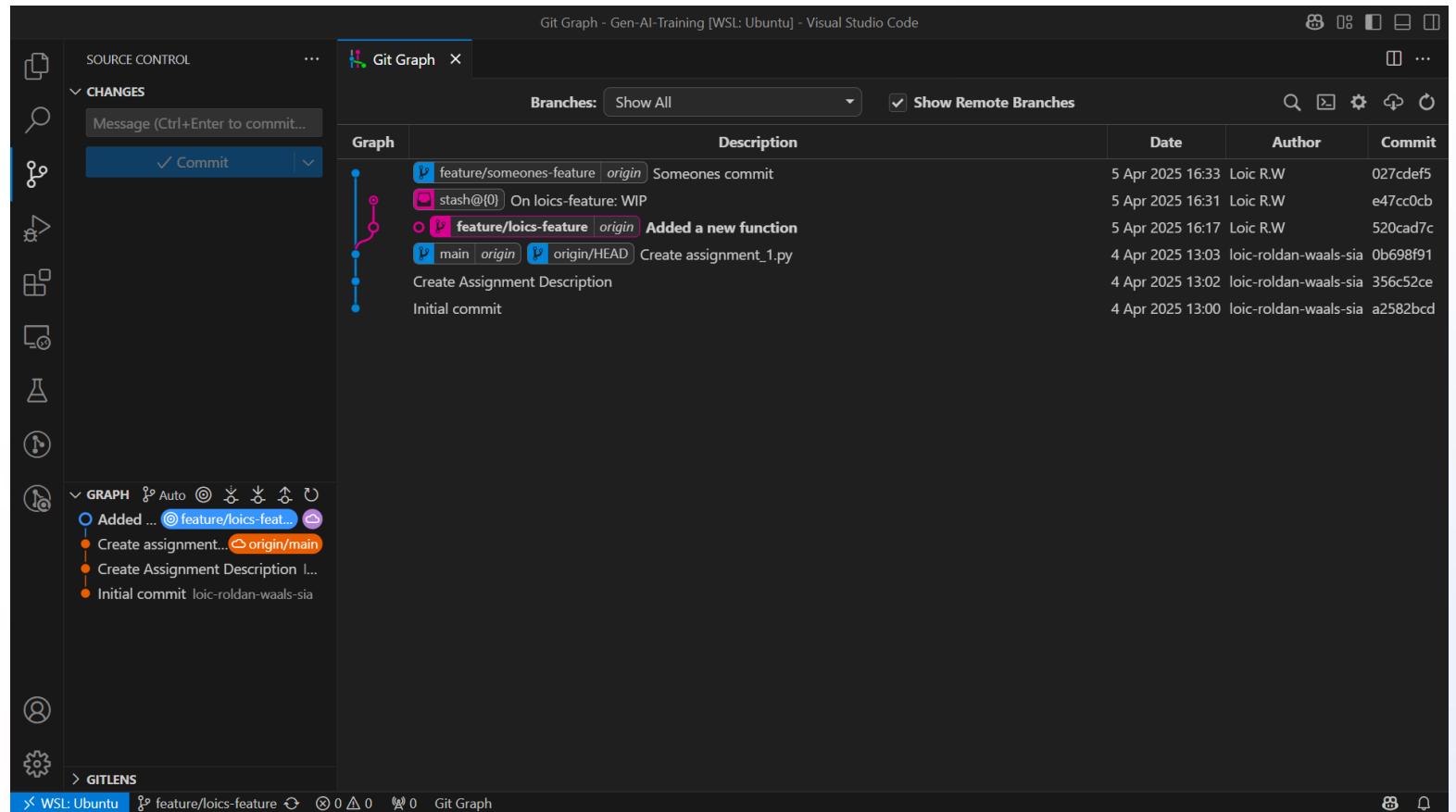
Let's try to use these commands!

- / **Stash** them and switch to some other branch (*feature/someones-feature*).
- / This is common when you are reviewing someone else's work on their branch, but you are not yet finished with your own work.



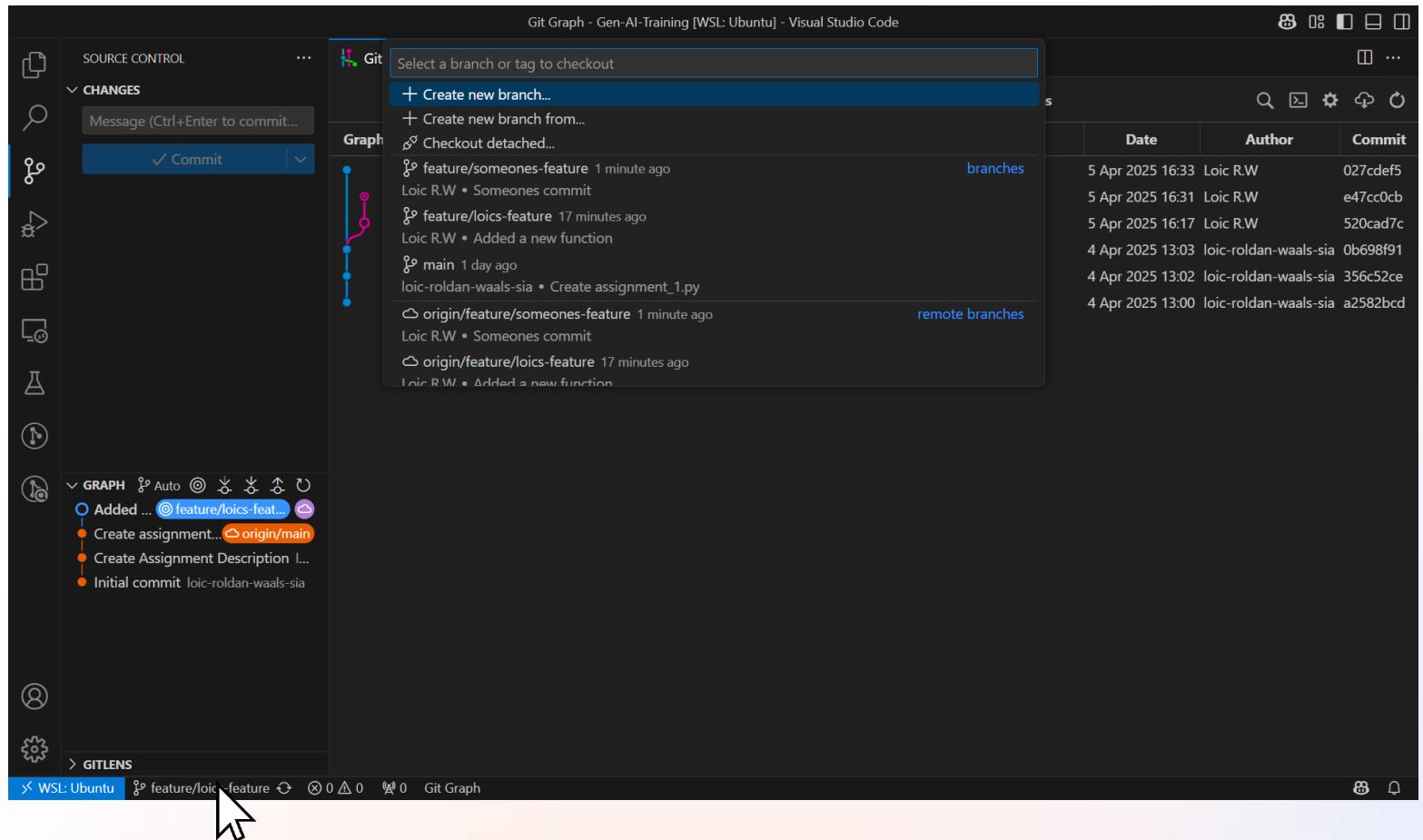
Let's try to use these commands!

- / **Stash** them and switch to some other branch (*feature/someones-feature*).
- / This is common when you are reviewing someone else's work on their branch, but you are not yet finished with your own work.



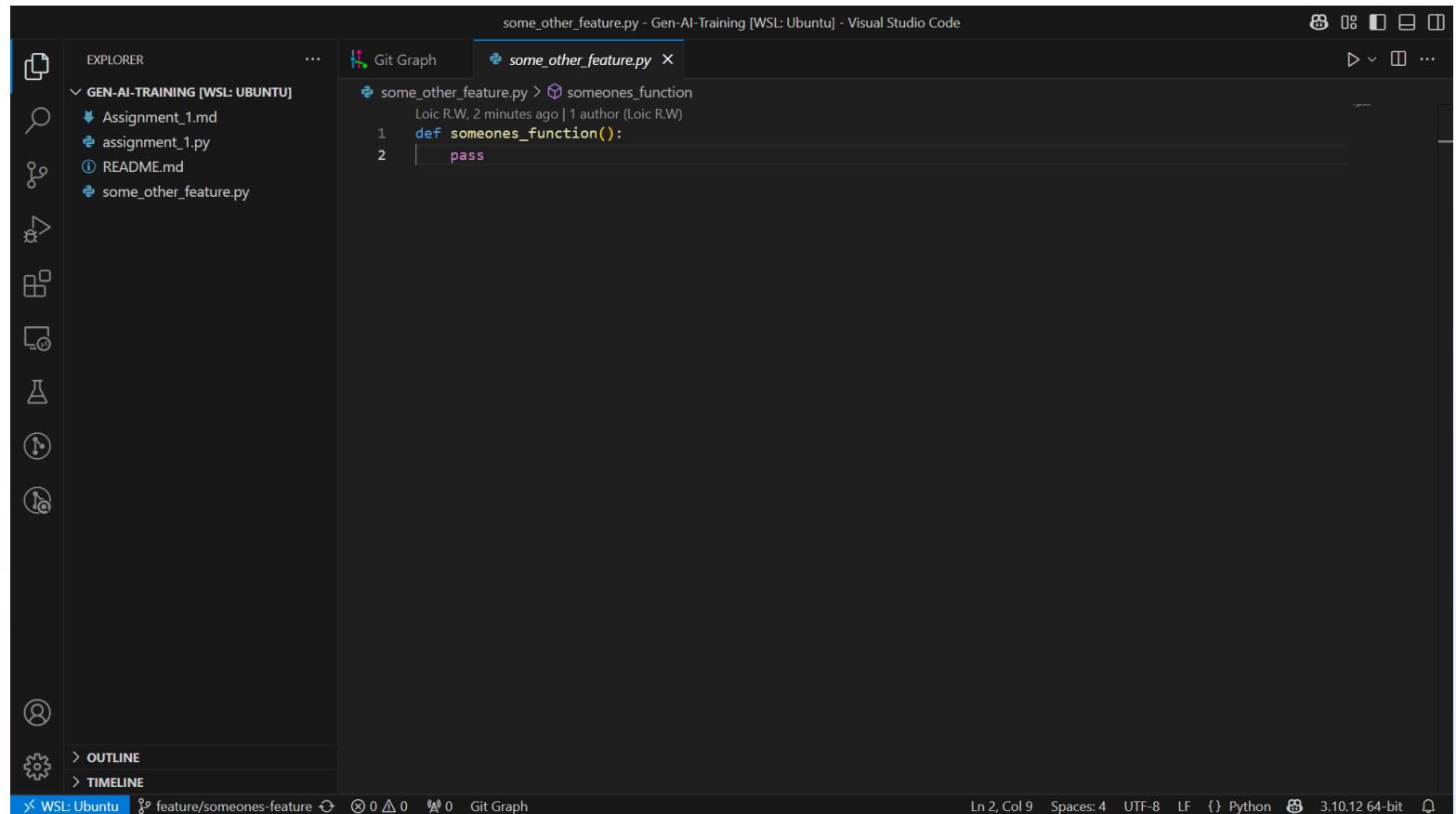
Let's try to use these commands!

- / **Stash** them and switch to some other branch (*feature/someones-feature*).
- / This is common when you are reviewing someone else's work on their branch, but you are not yet finished with your own work.



Let's try to use these commands!

- / **Stash** them and switch to some other branch (*feature/someones-feature*).
- / This is common when you are reviewing someone else's work on their branch, but you are not yet finished with your own work.



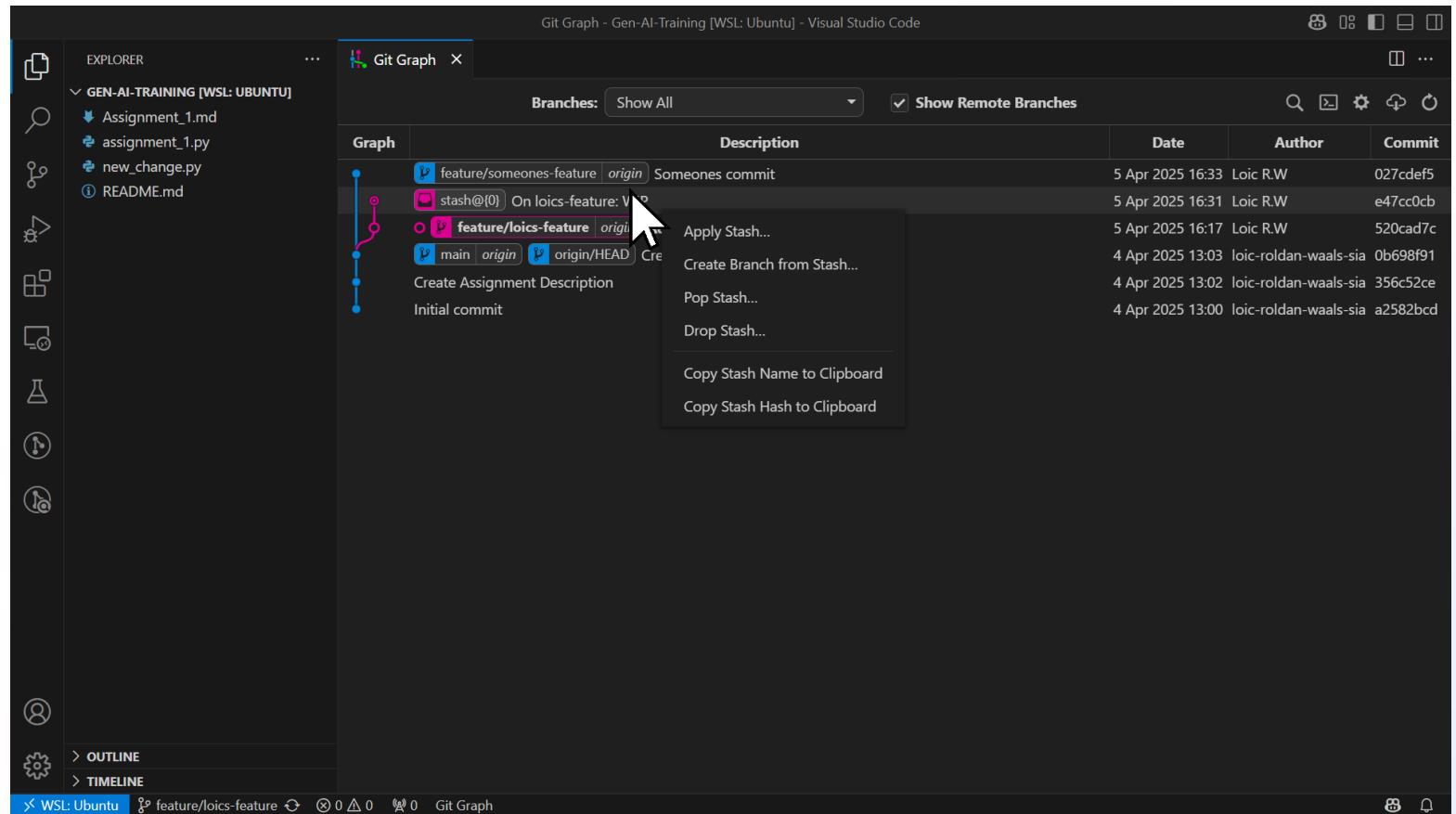
The screenshot shows a Visual Studio Code interface. The Explorer sidebar on the left lists files in the 'GEN-AI-TRAINING [WSL: UBUNTU]' folder: Assignment_1.md, assignment_1.py, README.md, and some_other_feature.py. The Git Graph tab is active in the top bar, showing a commit history for 'some_other_feature.py'. The most recent commit is by Loic R.W., 2 minutes ago, with one author. The commit message is 'someones_function'. The code editor shows the following Python code:

```
def someones_function():
    pass
```

The status bar at the bottom indicates the file is 'feature/someones-feature', has 0 changes, 0 additions, 0 deletions, and 0 unstaged changes. It also shows 'Git Graph' is selected. The bottom right corner shows the version '3.10.12 64-bit'.

Let's try to use these commands!

Come back to the original branch (`feature/loics-feature`) and **un-stash** them.



Let's try to use these commands!

Date	Author	Commit
5 Apr 2025 16:38 *	*	
5 Apr 2025 16:33	Loic R.W	027cdef5
5 Apr 2025 16:17	Loic R.W	520cad7c
4 Apr 2025 13:03	loic-roldan-waals-sia	0b698f91
4 Apr 2025 13:02	loic-roldan-waals-sia	356c52ce
4 Apr 2025 13:00	loic-roldan-waals-sia	a2582bcd

/ Wait for some other changes on *fake-main*.

Let's try to use these commands!

The screenshot shows the Visual Studio Code interface with the Git Graph extension. The Explorer sidebar displays the project structure under 'GEN-AI-TRAINING [WSL: UBUNTU]'. The Git Graph view shows the commit history with the following details:

Date	Author	Commit
5 Apr 2025 16:42	Loic R.W	0bbb5c36
5 Apr 2025 16:33	Loic R.W	027cdef5
5 Apr 2025 16:17	Loic R.W	520cad7c
4 Apr 2025 13:03	loic-roldan-waals-sia	0b698f91
4 Apr 2025 13:02	loic-roldan-waals-sia	356c52ce
4 Apr 2025 13:00	loic-roldan-waals-sia	a2582bcd

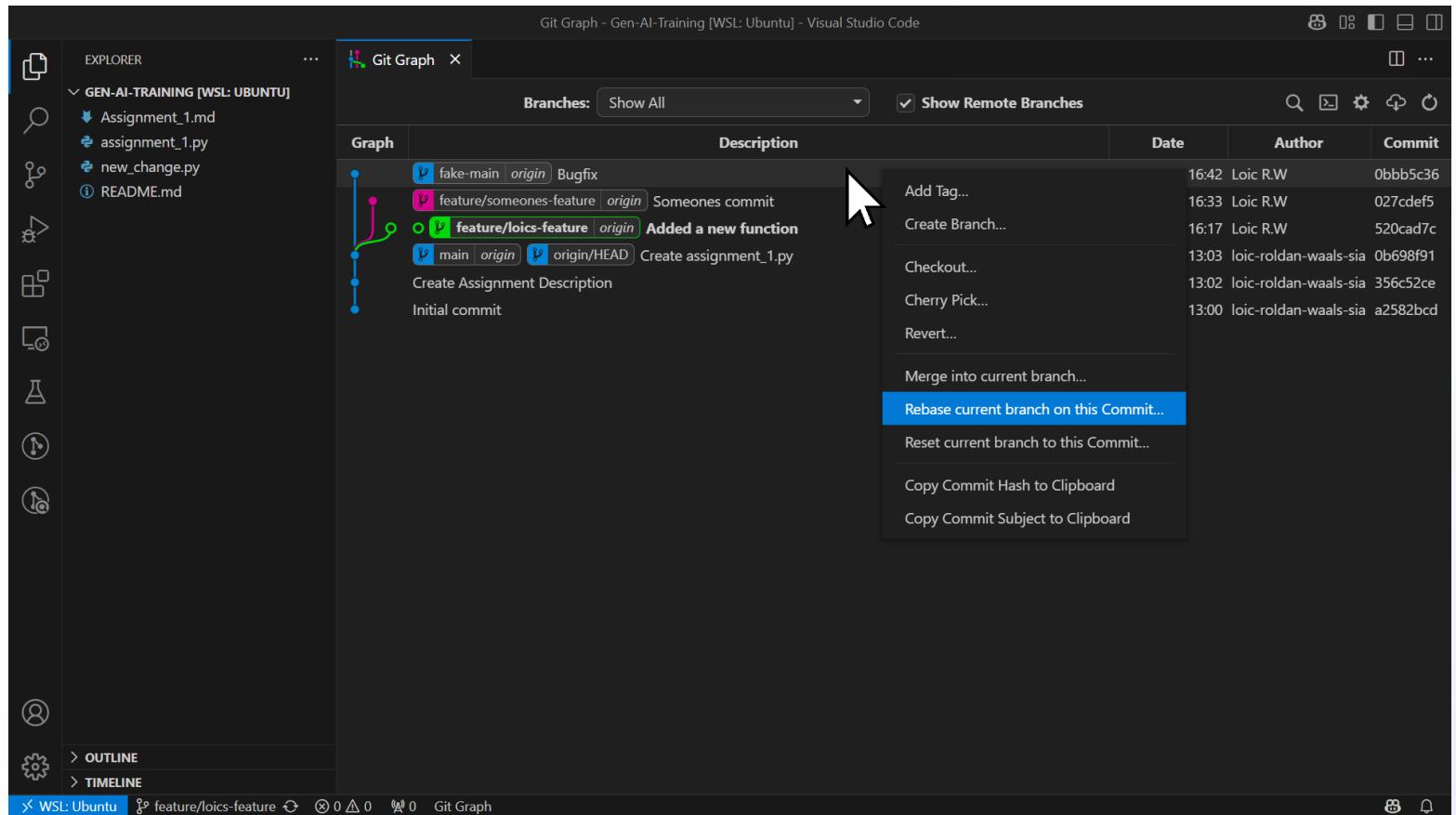
The commit history includes:

- fake-main | origin | Bugfix
- feature/someones-feature | origin | Someones commit
- feature/loics-feature | origin | Added a new function
- main | origin | origin/HEAD | Create assignment_1.py
- Create Assignment Description
- Initial commit

/ Wait for some other changes on *fake-main*.

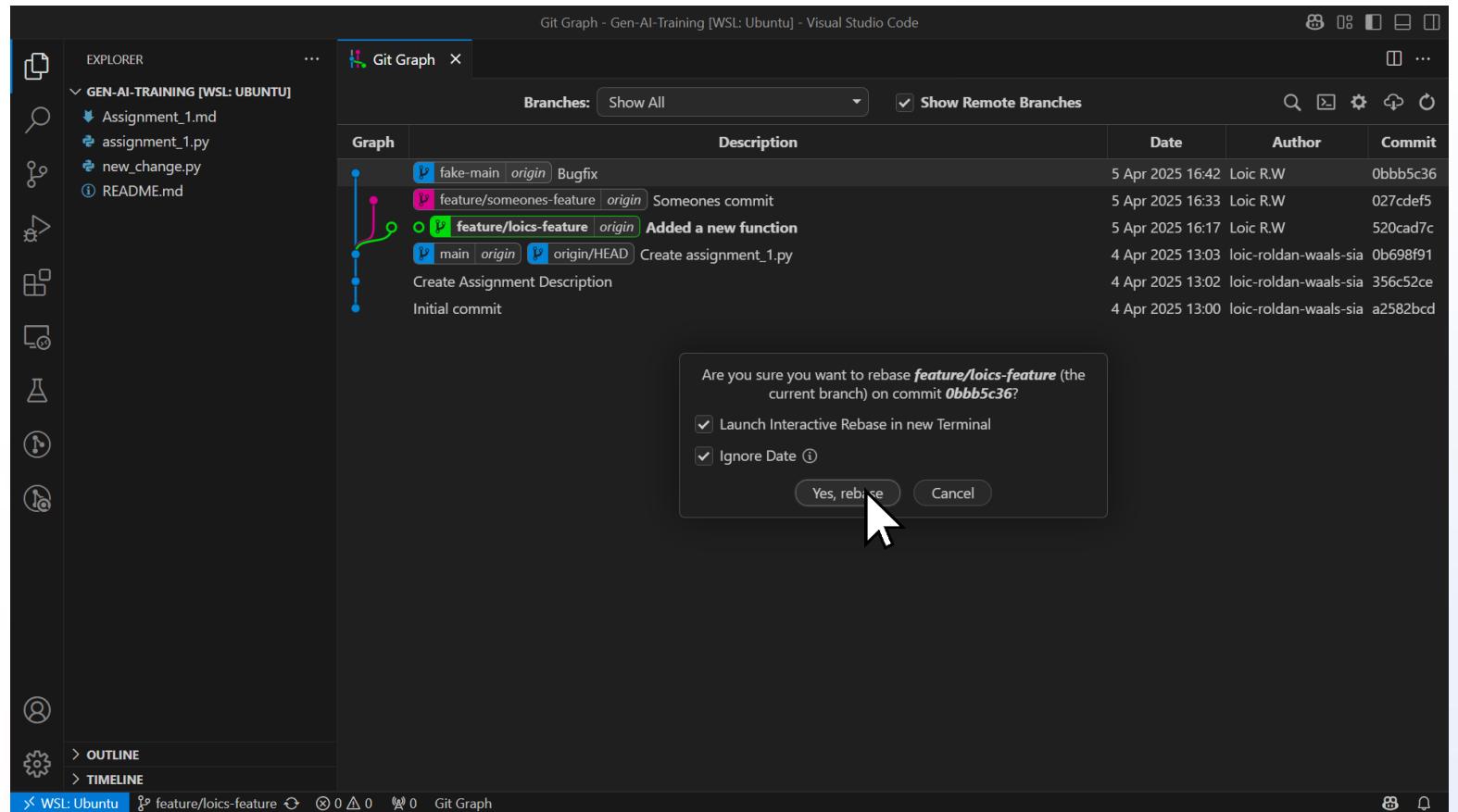
Let's try to use these commands!

- / Rebase *feature/loics-feature* on the latest changes from *fake-main*.



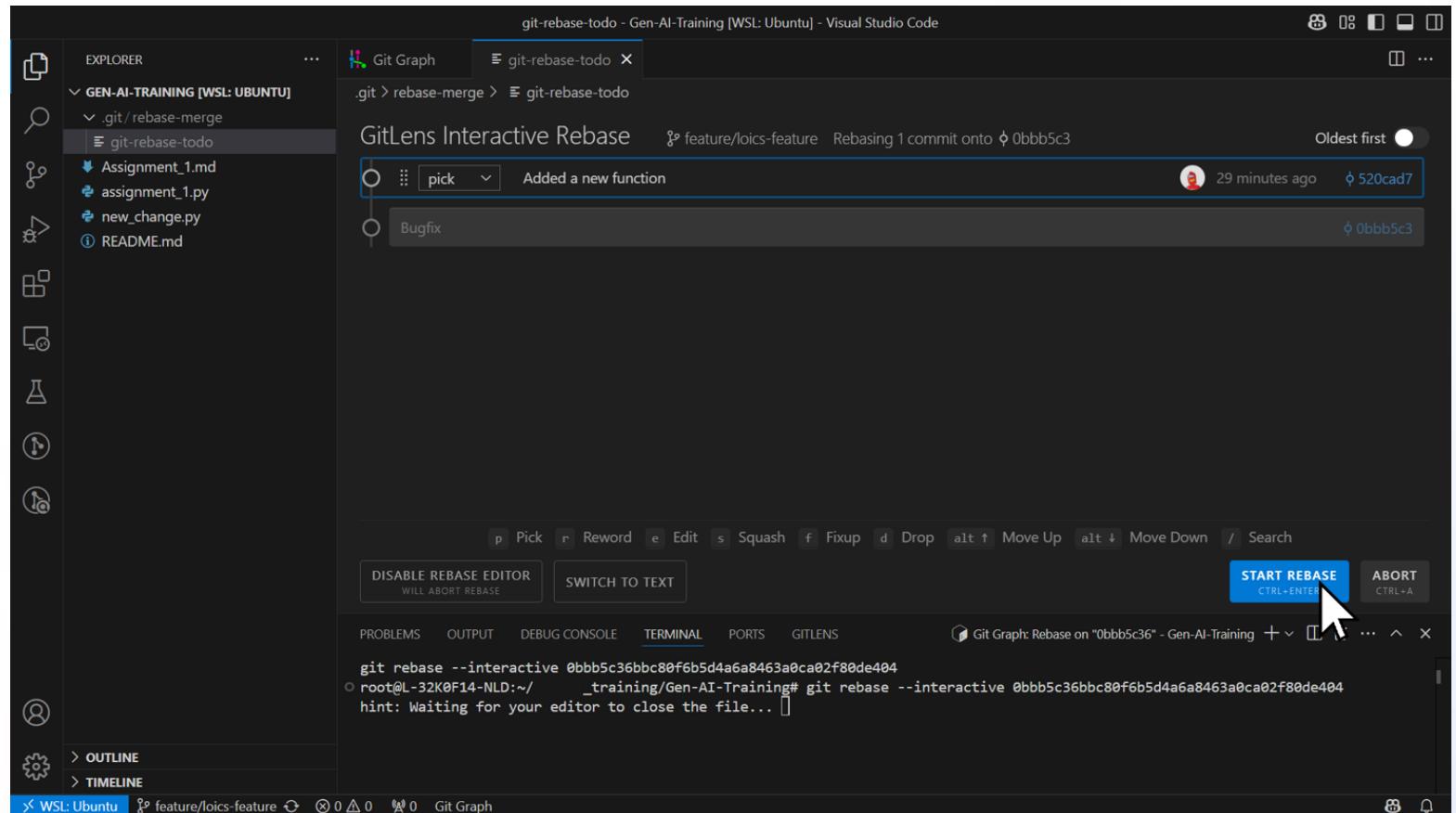
Let's try to use these commands!

/ Rebase *feature/loics-feature* on the latest changes from *fake-main*.



Let's try to use these commands!

- / This is the interactive rebase editor. Here you can:
 - / Reword commits
 - / Reorder commits
 - / Squash (combine) commits (if you want to turn 10 commits into 1)
 - / Drop commits (if you want to delete them from the history)
- / For now, let's just start the rebase.



Let's try to use these commands!

- Now our local is rebased.
- We can continue working on this branch now.

Date	Author	Commit
5 Apr 2025 16:17	Loic R.W	922226d7
5 Apr 2025 16:42	Loic R.W	0bbb5c36
5 Apr 2025 16:33	Loic R.W	027cdef5
5 Apr 2025 16:17	Loic R.W	520cad7c
4 Apr 2025 13:03	loic-roldan-waals-sia	0b698f91
4 Apr 2025 13:02	loic-roldan-waals-sia	356c52ce
4 Apr 2025 13:00	loic-roldan-waals-sia	a2582bcd

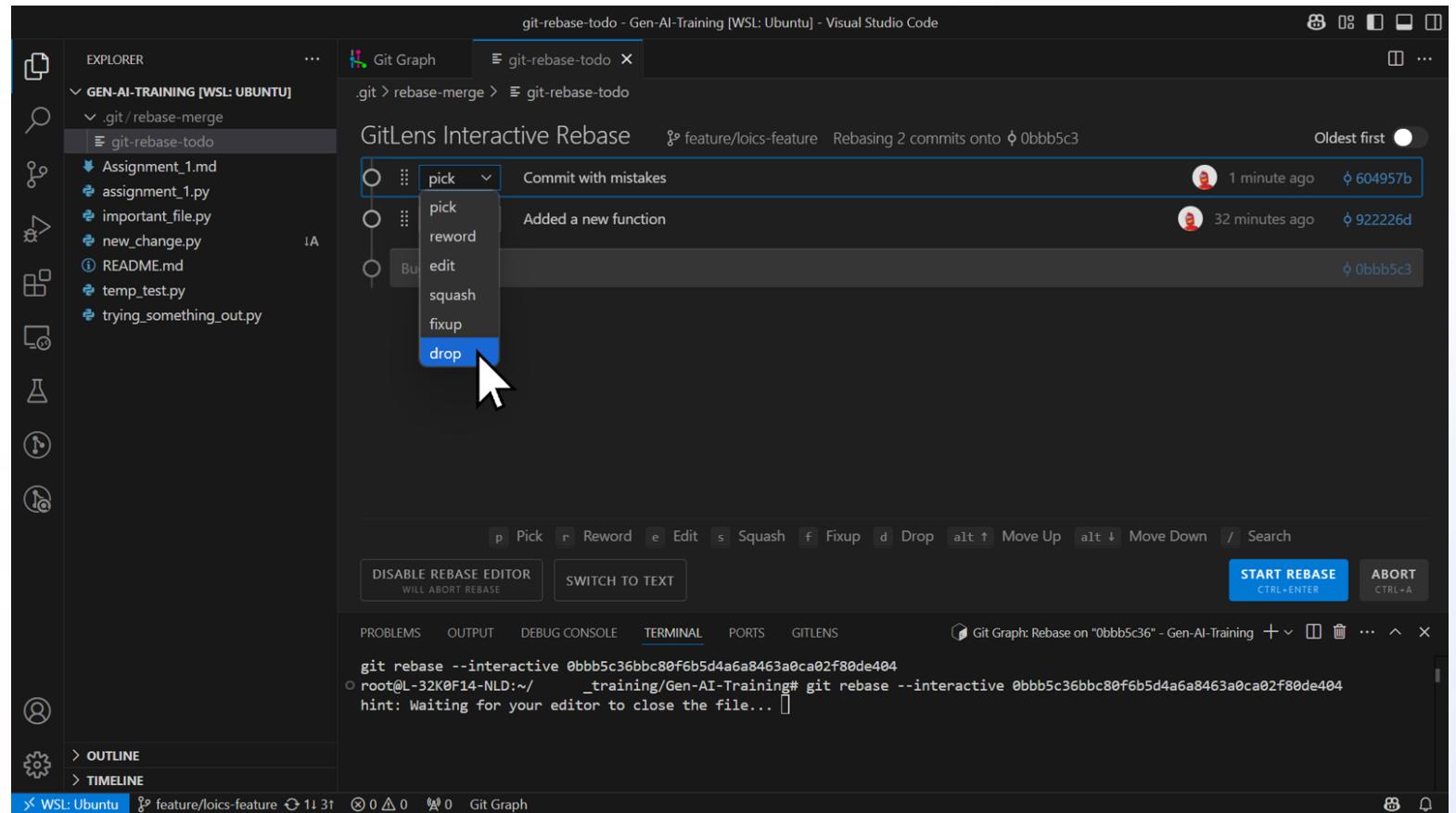
Let's try to use these commands!

- / Oh no! We make multiple **commits** and “accidentally” add some with some unwanted changes.
- / **Reset** (--mixed) or **Rebase** changes to create one new clean **commit**.

Graph	Description	Date	Author	Commit
feature/loics-feature	Commit with mistakes	5 Apr 2025 16:48	Loic R.W	604957b0
fake-main origin	Bugfix	5 Apr 2025 16:17	Loic R.W	922226d7
feature/someones-feature origin	Someones commit	5 Apr 2025 16:42	Loic R.W	0bbb5c36
origin/feature/loics-feature	Added a new function	5 Apr 2025 16:33	Loic R.W	027cdef5
main origin origin/HEAD	Create assignment_1.py	5 Apr 2025 16:17	Loic R.W	520cad7c
	Create Assignment Description	4 Apr 2025 13:03	loic-roldan-waals-sia	0b698f91
	Initial commit	4 Apr 2025 13:02	loic-roldan-waals-sia	356c52ce
		4 Apr 2025 13:00	loic-roldan-waals-sia	a2582bcd

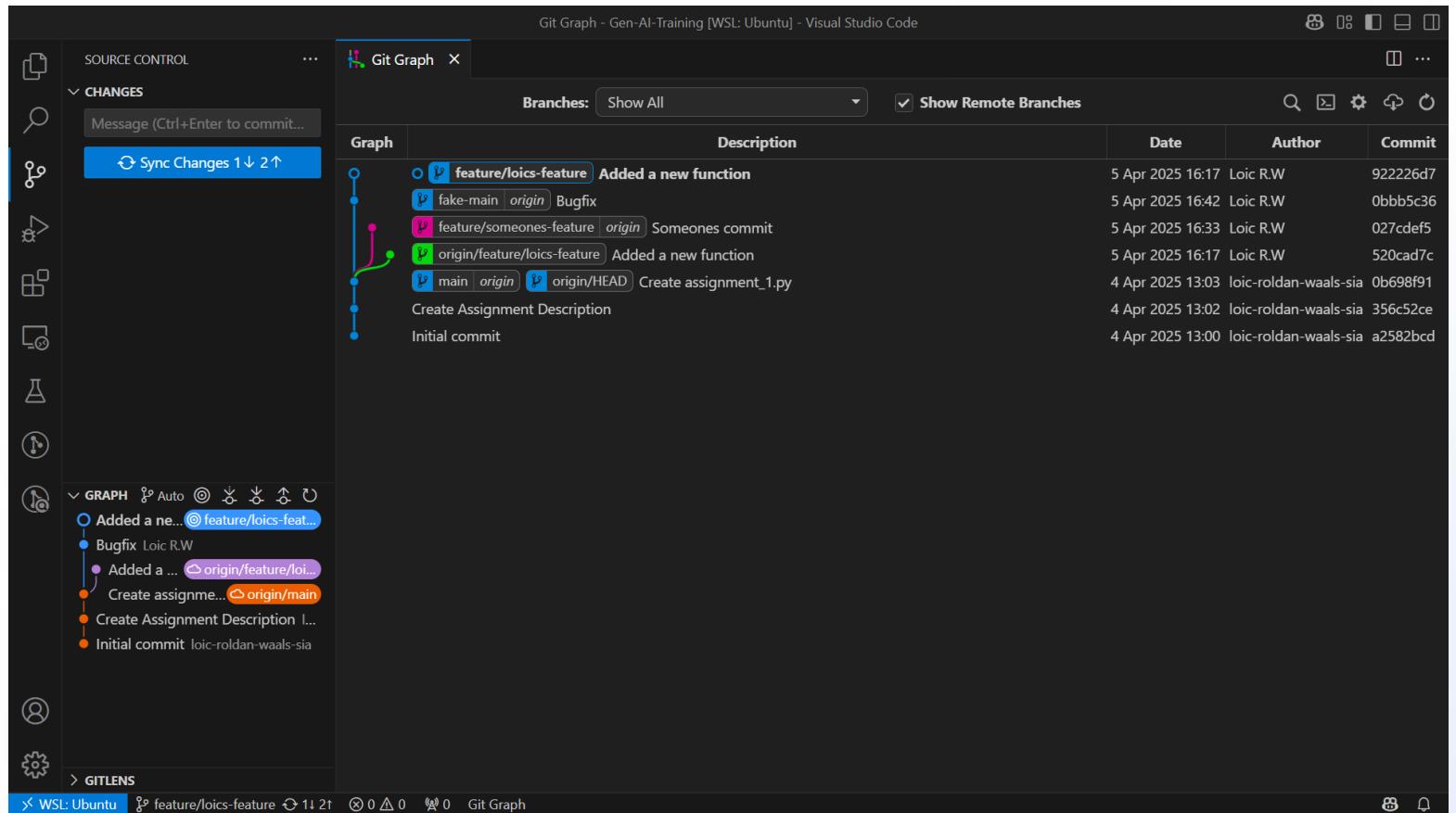
Let's try to use these commands!

- Oh no! We make multiple **commits** and “accidentally” add some with some unwanted changes.
- Reset** (--mixed) or **Rebase** changes to create one new clean **commit**.



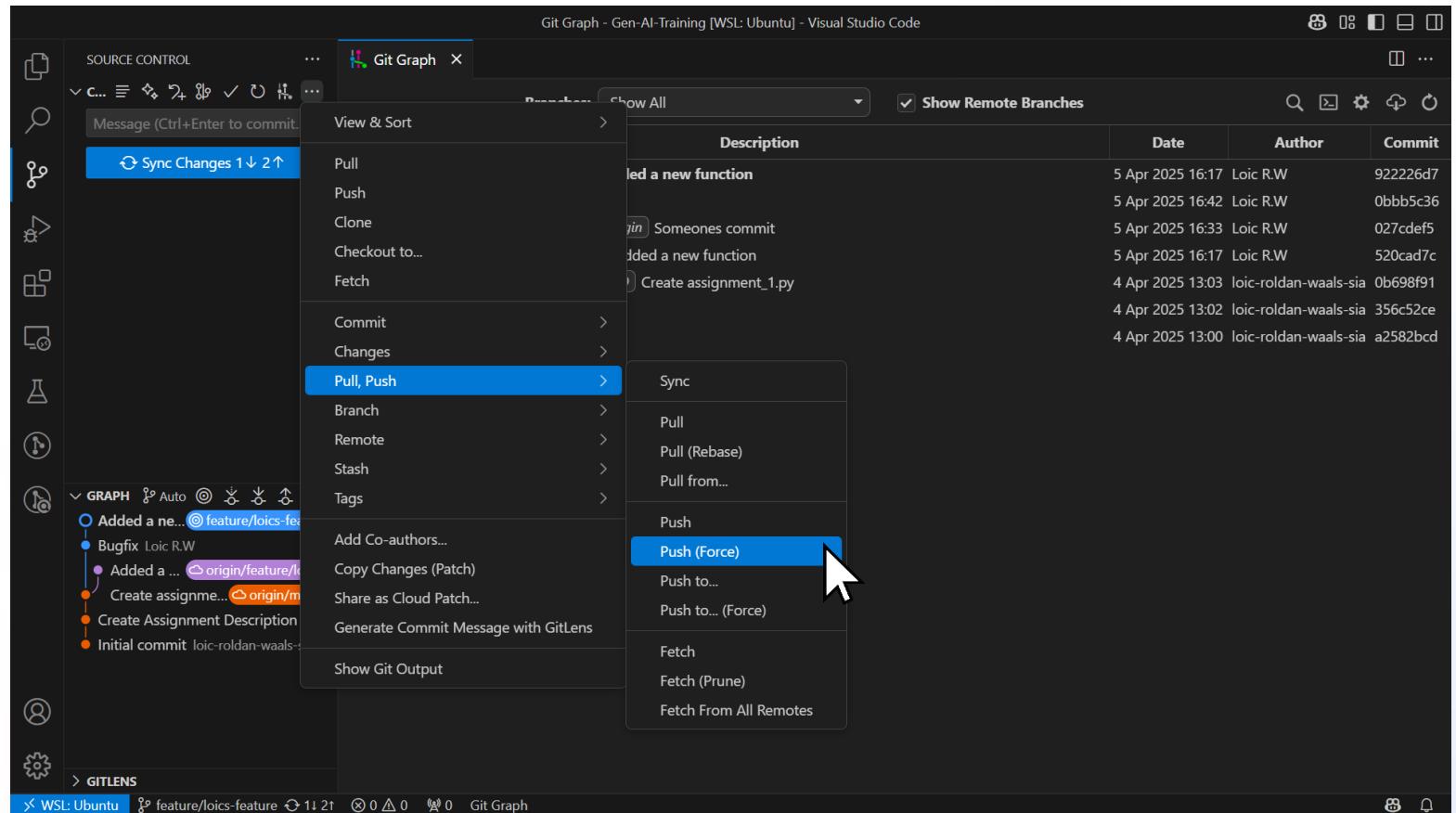
Let's try to use these commands!

/ Force push changes to fix the remote.

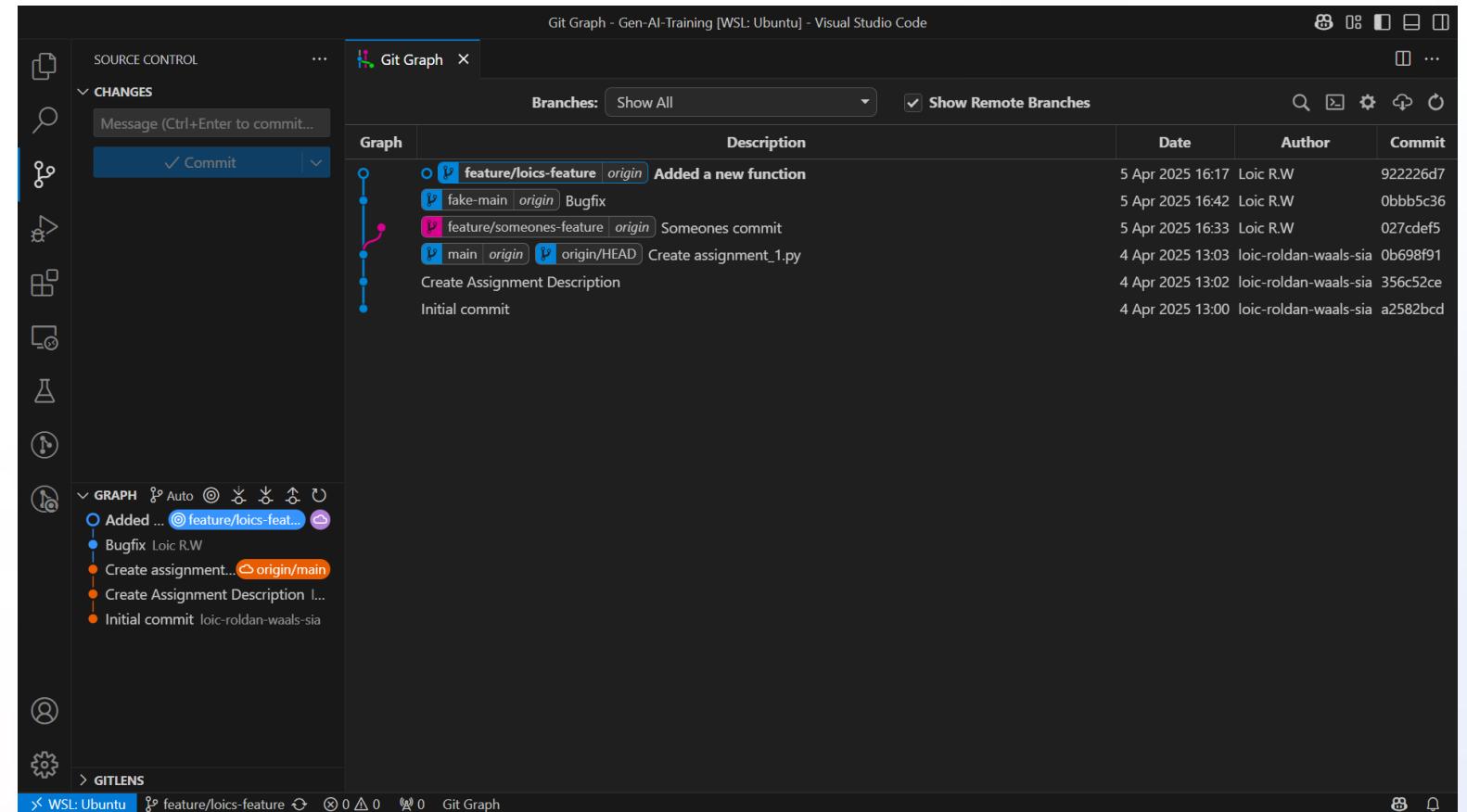


Let's try to use these commands!

/ Force push changes to fix the remote.



Let's try to use these commands!



/ All done!

Let's try to resolve merge conflicts!

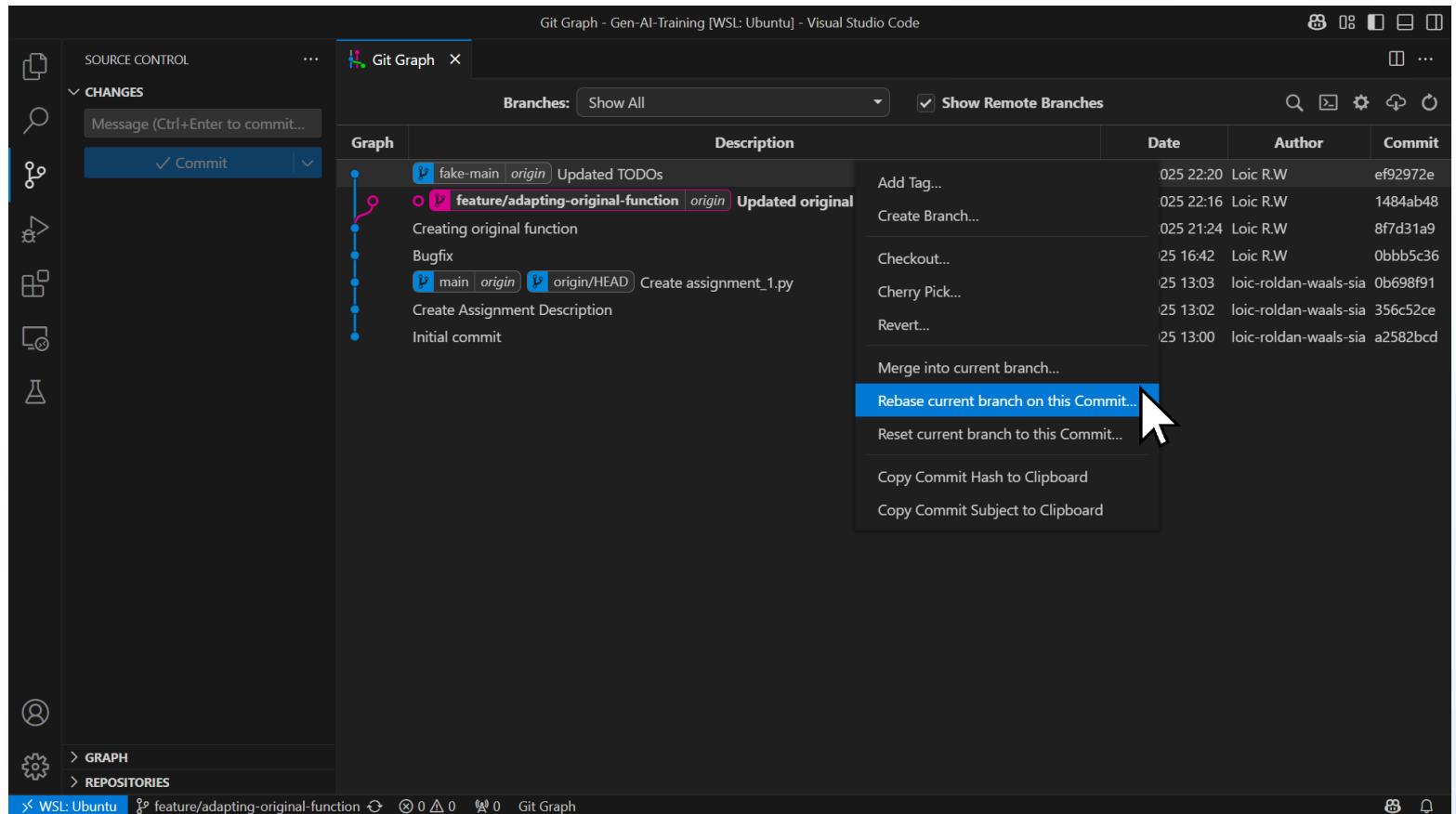
- / *feature/adapting-original-function* is based off *fake-main*.
- / The same file is changed in both branches.
- / When we try to rebase onto *fake-main* we trigger the conflict.

The screenshot shows the Visual Studio Code interface with the "Git Graph" and "Source Control" extensions active. The "Source Control" pane on the left displays a commit message input field and a "Commit" button. The "Git Graph" pane on the right shows a timeline of commits across three branches: *fake-main*, *feature/adapting-original-function*, and *main*. The graph highlights a merge conflict between *fake-main* and *feature/adapting-original-function*. The commit history includes:

Date	Author	Commit
28 Apr 2025 22:20	Loic R.W	ef92972e
28 Apr 2025 22:16	Loic R.W	1484ab48
28 Apr 2025 21:24	Loic R.W	8f7d31a9
5 Apr 2025 16:42	Loic R.W	0bbb5c36
4 Apr 2025 13:03	loic-roldan-waals-sia	0b698f91
4 Apr 2025 13:02	loic-roldan-waals-sia	356c52ce
4 Apr 2025 13:00	loic-roldan-waals-sia	a2582bcd

Let's try to resolve merge conflicts!

- / *feature/adapting-original-function* is based off *fake-main*.
- / The same file is changed in both branches.
- / When we try to rebase onto *fake-main* we trigger the conflict.



Let's try to resolve merge conflicts!

- / *feature/adapting-original-function* is based off *fake-main*.
- / The same file is changed in both branches.
- / When we try to rebase onto *fake-main* we trigger the conflict.

The screenshot shows the Visual Studio Code interface with the 'Git Graph' extension. On the left, the 'SOURCE CONTROL' sidebar shows a 'Changes' list with an item 'Updated original function' and a 'Merge Changes' section containing 'important_file.py'. The 'Git Graph' view in the center displays a timeline of commits across three branches: 'fake-main', 'feature/adapting-original-function', and 'main'. A conflict is indicated by a pink merge node between 'fake-main' and 'feature/adapting-original-function'. The right panel lists individual commits with their details:

Date	Author	Commit
28 Apr 2025 22:20	Loic R.W	ef92972e
28 Apr 2025 22:16	Loic R.W	1484ab48
28 Apr 2025 21:24	Loic R.W	8f7d31a9
5 Apr 2025 16:42	Loic R.W	0bbb5c36
4 Apr 2025 13:03	loic-roldan-waals-sia	0b698f91
4 Apr 2025 13:02	loic-roldan-waals-sia	356c52ce
4 Apr 2025 13:00	loic-roldan-waals-sia	a2582bcd

Let's try to resolve merge conflicts!

- We can fix the changes (file by file) by clicking them in the source control tab and pressing the button on the bottom right that says “Resolve in Merge Editor”

The screenshot shows a Visual Studio Code interface with the following details:

- SOURCE CONTROL** sidebar:
 - A dropdown menu for 'important_file.py' is open, showing 'Updated original function' and a 'Continue' button.
 - 'Merge Changes' section: 1 file, 'important_file.py'.
 - 'Changes' section: 0 files.
- Git Graph** tab is active.
- important_file.py** editor tab is active, showing the following Python code:

```
def original_function():
    # TODO don't forget to add type hint to function
    pass
=====
"""
This does something important
"""

print('beep boop')
```
- Status Bar**: WSL: Ubuntu, ef92972e! (Rebasing), Git Graph, Ln 2, Col 13, Spaces: 4, UTF-8, LF, Python, 3.10.12 64-bit.

Let's try to resolve merge conflicts!

- ✓ Ensure that the code in the “Result” tab on the bottom is what you would like to see.
- ✓ You can use the buttons like “Accept Current” to speed this up

The screenshot shows the Visual Studio Code interface during a merge operation. The title bar indicates "Merging: important_file.py - Gen-AI-Training [WSL: Ubuntu] - Visual Studio Code".

The left sidebar shows the "SOURCE CONTROL" view with a "CHANGES" section containing a single item: "Updated original function" with a "Continue" button. Below it is a "Merge Changes" section for "important_file.py".

The main editor area displays the content of "important_file.py". It contains the following code:

```
def original_function():
    # TODO don't forget to add type hint to function
    pass
```

The "Incoming" file also contains the same code structure:

```
def original_function():
    """
    This does something important
    """
    print('beep boop')
```

A yellow box highlights the conflict resolution area in the "Incoming" file, specifically the multi-line string and the `print` statement. The status bar at the bottom right shows "1 Conflict Remaining".

The "Result" tab at the bottom shows the final merged code:

```
def original_function():
    Loic R.W, 1 hour ago • Creating original function
    No Changes Accepted
    pass
```

A blue "Complete Merge" button is visible in the bottom right corner of the editor area.

Let's try to resolve merge conflicts!

/ Press “Complete Merge” to finish.

The screenshot shows a Visual Studio Code window titled "Merging: important_file.py - Gen-AI-Training [WSL: Ubuntu] - Visual Studio Code". The left sidebar has "SOURCE CONTROL" selected, showing a "CHANGES" section with a tooltip "Updated original function" and a "Continue" button. The main editor area displays code for "important_file.py" with a conflict at line 2. The "Incoming" branch code is:

```
def original_function():
    """
    This does something important
    """
    print('beep beep')
```

The "Current" branch code is:

```
def original_function():
    # TODO don't forget to add type hint to function
    pass
```

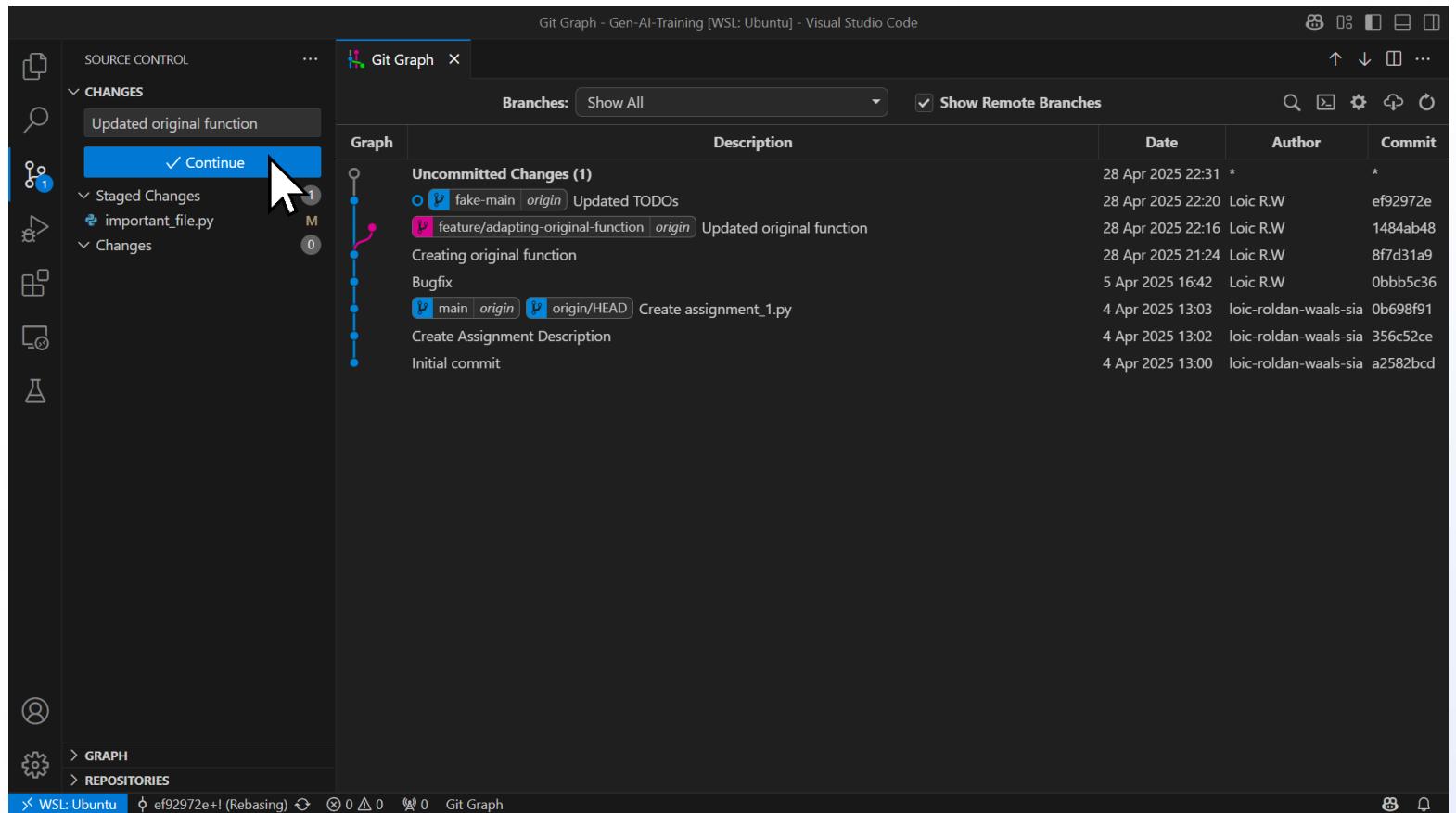
The result of the merge is:

```
def original_function():
    Current + Incoming | Remove Current | Remove Incoming
    # TODO don't forget to add type hint to function
    """
    This does something important
    """
    print('beep beep')
```

A cursor is hovering over the "Complete Merge" button in the bottom right corner.

Let's try to resolve merge conflicts!

- / At this point you can add additional changes to the commit.
- / Press “Continue” to confirm your merges.



Let's try to resolve merge conflicts!

/ Lastly, force push your changes to overwrite the (incorrect) commit on the remote.

The screenshot shows the Visual Studio Code interface with the Git Graph extension open. The left sidebar has 'SOURCE CONTROL' selected, showing a 'CHANGES' section with a message field and a 'Sync Changes 1↓ 2↑' button. The main area displays a 'Git Graph' with a timeline of commits. The graph shows several branches: 'feature/adapting-original-function', 'fake-main', 'origin', 'origin/feature/adapting-original-function', 'main', and 'origin/HEAD'. The commits are listed in the following table:

Date	Author	Commit
28 Apr 2025 22:33	Loic R.W	f9b699a4
28 Apr 2025 22:20	Loic R.W	ef92972e
28 Apr 2025 22:16	Loic R.W	1484ab48
28 Apr 2025 21:24	Loic R.W	8f7d31a9
5 Apr 2025 16:42	Loic R.W	0bbb5c36
4 Apr 2025 13:03	loic-roldan-waals-sia	0b698f91
4 Apr 2025 13:02	loic-roldan-waals-sia	356c52ce
4 Apr 2025 13:00	loic-roldan-waals-sia	a2582bcd

Let's try to resolve merge conflicts!

/ Lastly, force push your changes to overwrite the (incorrect) commit on the remote.

The screenshot shows the Visual Studio Code interface with the Git Graph extension open. The Git Graph view displays a commit history with the following details:

Date	Author	Commit
28 Apr 2025 22:33	Loic R.W	f9b699a4
28 Apr 2025 22:20	Loic R.W	ef92972e
28 Apr 2025 21:24	Loic R.W	8f7d31a9
5 Apr 2025 16:42	Loic R.W	0bbb5c36
4 Apr 2025 13:03	loic-roldan-waals-sia	0b698f91
4 Apr 2025 13:02	loic-roldan-waals-sia	356c52ce
4 Apr 2025 13:00	loic-roldan-waals-sia	a2582bcd

The Source Control pane shows a commit message: "Message (Ctrl+Enter to commit...)". The commit button is highlighted with a blue background and white text. The status bar at the bottom indicates the current repository is "WSL: Ubuntu" and the branch is "feature/adapting-original-function".

To ensure only correct code goes to production we review each other's work

Common PR Challenges

- / Large Pull Requests -> Large pull requests are difficult to review. Break them down into smaller, more manageable chunks. Only include code that actually contributes to the change.
- / Unresolved Conflicts -> Regularly sync your branch with the main branch to avoid complex conflicts later.
- / Stale Pull Requests -> If a pull request becomes outdated, communicate with the team to update or close it.

Good reference guide from google: [Link](#)

PR Best Practices

- / Keep Pull Requests Small.
- / Write Descriptive Titles and Descriptions.
- / Address Feedback Promptly.
- / Communicate Clearly.
- / Branch hygiene.

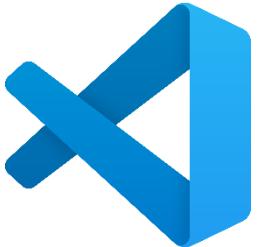
Tips for Merge Conflicts

- / Ensure you have as few commits as possible.
- / Always re-test your code completely.
- / They are always a headache, preventing them (with frequent rebases) is better than fixing them.

Let's practice!



Changing some configurations will make it easier to work with git.



VS Code

- / For VSCode settings, use Ctrl + Shift + P and write "open default settings (JSON)".
- / Choose the JSON option and paste the provided settings.
- / Install Git Graph and GitLens for nicer GUI



Git

- / Open a terminal
- / Enter the commands:
 - / git config --global core.editor "code --wait"
 - / git config --global -e
- / Then paste the provided example settings.
- / Close the file to save it.



We're a next-gen consulting group for a new generation of change.

We're born digital, augmented by data, enhanced by creativity and driven by responsibility.

A new kind of global management consulting group designed to design the future.

We are optimists for change

What you get out of change depends on how you go into it. We believe that optimism is a force multiplier. Our expertise delivers results. Our optimism transforms outcomes.

3,000+ professionals

48 offices

19 countries

25 years of growth



AI and data science is the powerhouse of Sia,
augmenting consulting services and
accelerating productivity.

350+ AI & data experts

12 R&D labs

Advanced GenAI assistants

Positioned as a leader in AI adoption

Our AI x consulting model delivers scalable
business solutions





AI and data science is the powerhouse of Sia, augmenting consulting services and accelerating productivity.

350+ AI & data experts

12 R&D labs

Advanced GenAI assistants

AI consulting services

Building data driven, AI driven organization and agentic AI architectures and covering end-to-end AI value chain, from ideation to industrialization (data science, data marketing, data engineering & AI technologies)

Augmented consulting services

Combining strategy/industry expertise and AI in the same engagements

12 R&D labs coverings all AI techniques

NLP & GenAI, computer vision, time series, operational research, data capture, data duality, engineering, marketing analytics, advanced analytics, climate, asset & liability management, blockchain & cryptocurrencies

In-house AI platform and AI-fueled datasets built

Thanks to unique and distinctive data scraping capabilities and data science expertise, available for consultants and clients

SiaGPT

Set of advanced GenAI assistants, based on evolving multi-LLMs technologies. Available as white label GPT infrastructure and AI/GenAI use cases industrialization platform



Merci

Sia is a next-generation, global management consulting group—born digital, augmented by data, enhanced by creativity, and driven by responsibility. Our 3,000+ professionals partner with clients to resolve challenges and capitalize on opportunities. We believe that in today's world of change and disruption, optimism is a force multiplier.

sia-partners.com | follow us: @SiaPartners



Optimists for change