

How To Set Up Your Research Project with VSCode on the SAFE Datacenter Infrastructure

Philip Schwedler
SAFE Research Datacenter*

Last Update: September 8, 2025

We encourage all SAFE researchers to actively use the SAFE Data Center research infrastructure for their projects. Compared to traditional use of a local computer, the SAFE Database Cluster offers greater resource availability, improved computing power, and easier sharing of large projects among SAFE researchers. In this short tutorial we will show you how to set up VSCode. VSCode is a source code editor, e.g. for Python code or $\text{\LaTeX}$, with a very convenient in-built feature, the Visual Studio Code Remote - SSH extension, which allows you to open a remote folder on any remote machine, virtual machine or container with a running SSH server (e.g. the SAFE server cluster). Once connected to the server, you can interact with files and folders anywhere on the remote file system and run your files directly on the server.

Contents

1	Contact	2
2	Installation Requirements	2
3	SAFE Datacenter Infrastructure	2
4	Connecting to the Server Cluster	3
5	Set Up the Project Environment & Using Git	6
5.1	Create Project Directory	6
5.2	Activate Git	9
6	Optional: Set Up Virtual Environment to Freeze Coding Environment	13
7	Optional: Include $\text{\LaTeX}$	14
8	Optional: Use Git Submodules to combine different projects	15

*Original guide by Philip in 2022. Since then, updated by others from the Data Center.

1 Contact

If you encounter any difficulties or just want general information, do not hesitate to contact us.

SAFE Research Datacenter: datacenter@safe-frankfurt.de

More information about the SAFE Research Datacenter, and further guides can be found [here](#), and [here](#).

2 Installation Requirements

For security reasons, a connection to the SAFE database cluster can only be established from the university network. From home, you must use Cisco Anyconnect VPN to connect to the SAFE computer cluster. Find instructions for this [here](#). For the VPN use your HRZ credentials. The credentials for authentication to the database cluster are provided by the SAFE Data Center. We ask you to install the following program:

- Installation of [Visual Studio Code](#) (VSCode)

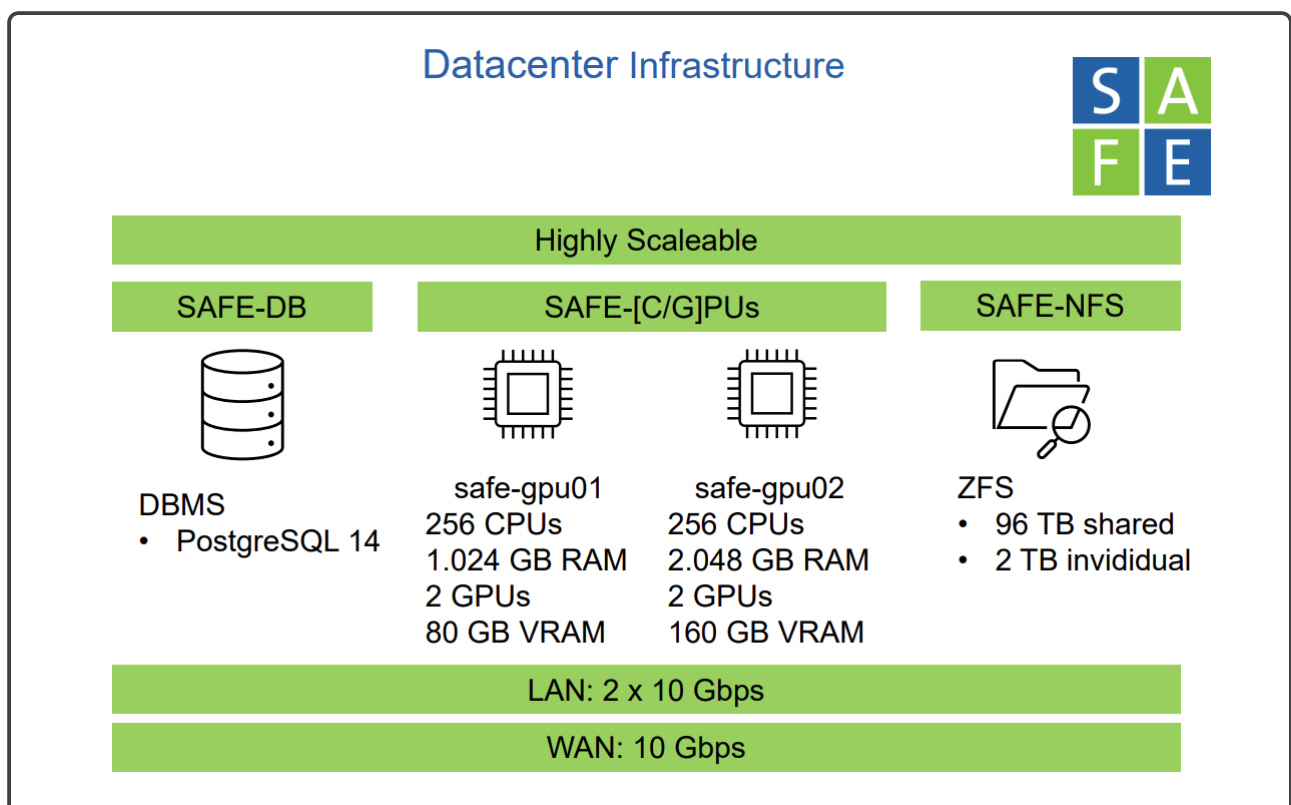
If you want to use VSCode and its features described below locally as well (i.e. if you are not connected to the SAFE server cluster), we recommend that you also install the following:

- Installation of [Git](#)
- Optional for $\text{\LaTeX}$ Integration: Installation of Tex distribution, e.g. [TeX Live](#)

For some of these installations administrator rights are required, for these contact the IT support.

IT Administrator: it_support@safe-frankfurt.de

3 SAFE Datacenter Infrastructure



SAFE is in the comfortable position to host three computing nodes. They can be reached via [safe-gpu0\[1,2,3\].safe-frankfurt.de](#). The following guide assumes that you want to access [safe-gpu01.safe-frankfurt.de](#). The access to e.g., [safe-gpu02.safe-frankfurt.de](#) is completely analogous by just exchanging [safe-gpu01](#) with [safe-gpu02](#). More information about the SAFE Research Infrastructure can be found [here](#).

4 Connecting to the Server Cluster

Every SAFE user has access to a personal drive share on the server cluster, depending on the machine you are using, this can be accessed through

On your Windows machine

- `\\storage.safe-frankfurt.de\<hrz_username>`

On your MacBook / iMac

- `smb://storage.safe-frankfurt.de/<hrz_username>`

On the compute node

- `homedir, <hrz_user>@safe-gpu01:~ $`

Every other shared drive on the server cluster can be accessed through

On your Windows machine

- `\\storage.safe-frankfurt.de\<share>`

On your MacBook / iMac

- `smb://storage.safe-frankfurt.de/<share>`

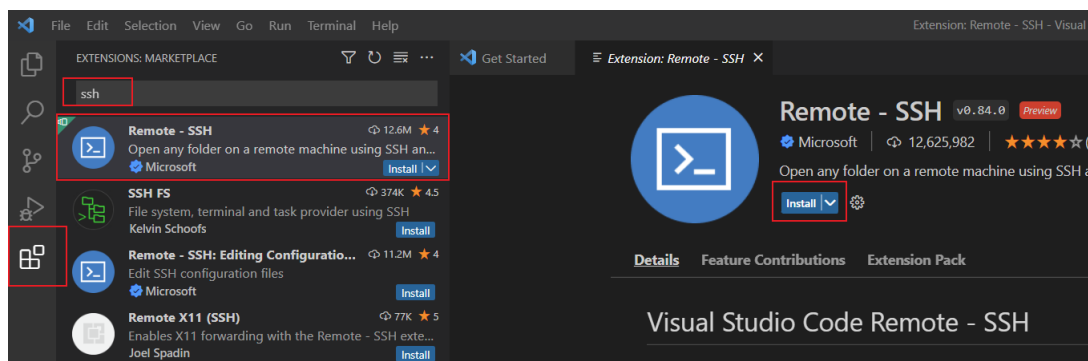
On the compute node

- `/SAFE_NAS`

In the following we describe how you can access either your personal or shared drive on the SAFE server cluster from your local machine through VS Code. The following guide is adjusted to the SAFE server environment. For a general short introduction, we recommend [this](#) video.

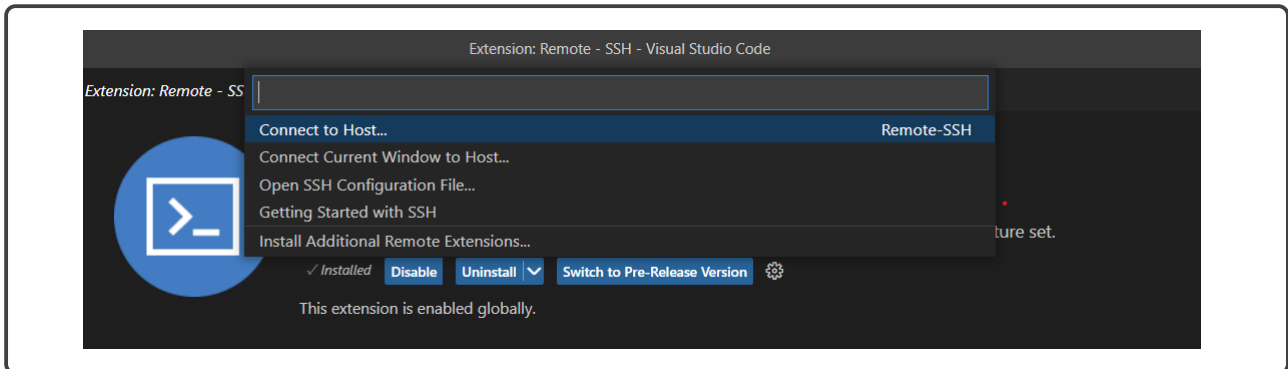
1. Install Remote - SSH Connection

Go to Extensions on the left and search for ssh. Then install Remote - SSH.

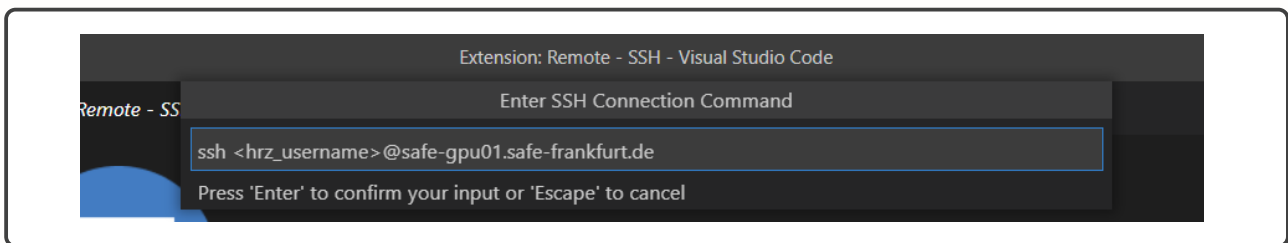


2. Connect to the Remote for the First Time

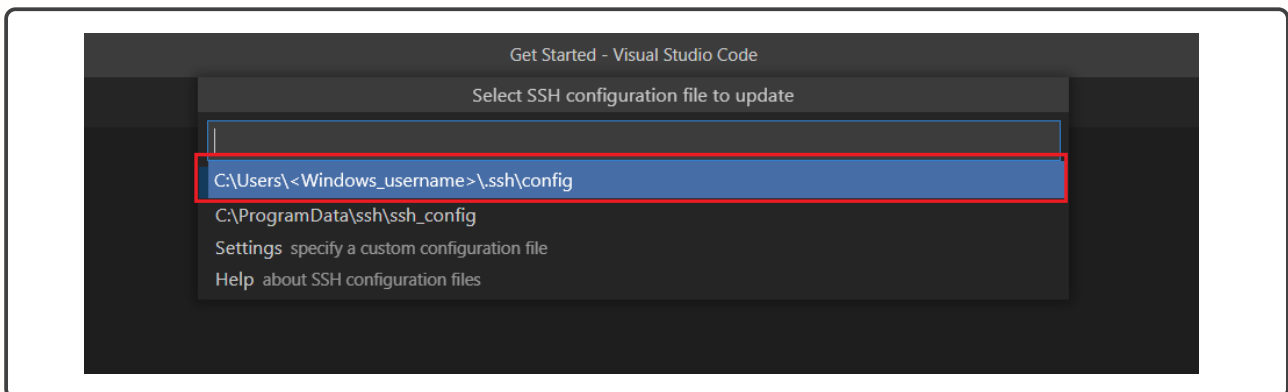
Click on the green button in the lower left corner (just below the gear). Then the following window will open:



After you click **Connect to Host...**, click **Add New SSH Host...** and enter the following:

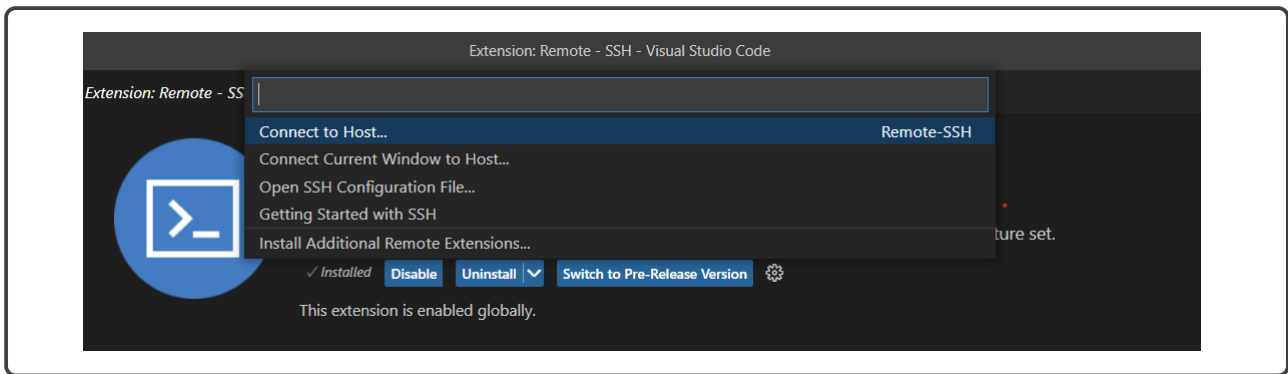


Confirm your input by pressing **Enter** on the keyboard and then click:

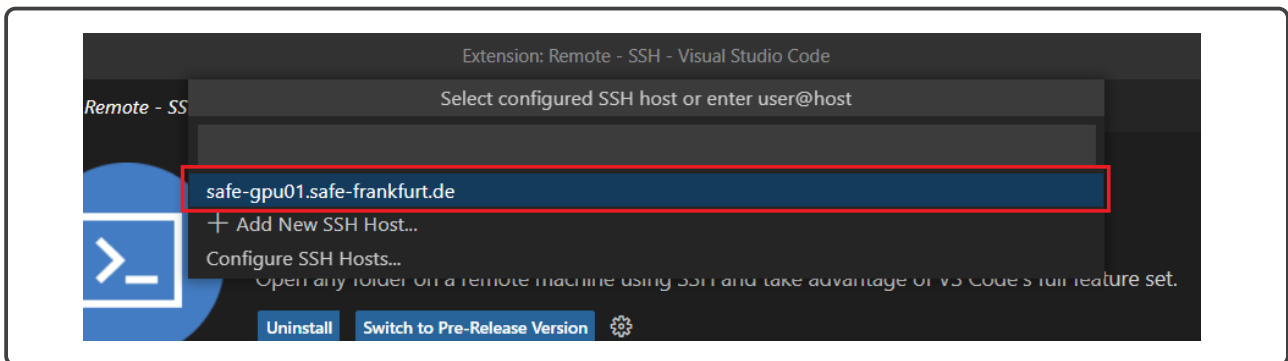


3. Connect to the Remote

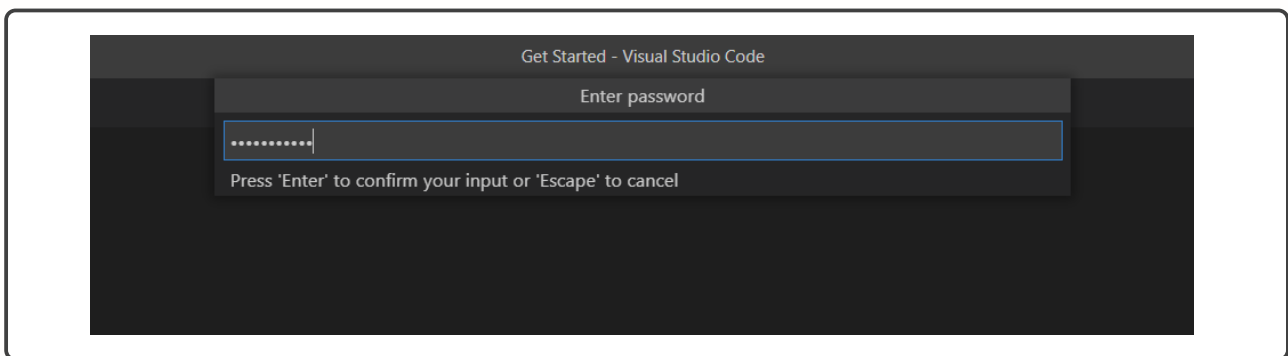
Click on the green button in the lower left corner (just below the gear). Then the following window will open:



After clicking **Connect to Host...** click on **safe-gpu01.safe-frankfurt.de**



Then a new VSCode window will open where you have to enter your HRZ password.



There might also appear a pop-up regarding the operating system. Choose "linux" here. When the connection is successfully established, you will see the green box with **SSH:safe-gpu01.safe-frankfurt.de** in the lower left corner.

4. General Remarks

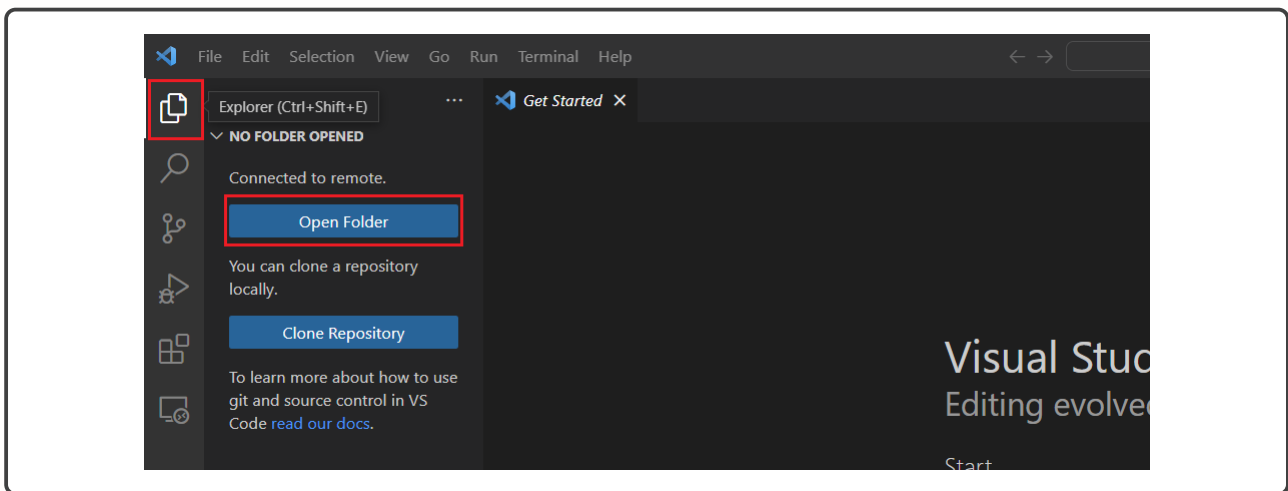
It is not necessary to connect to the server cluster. You can also use VSCode locally on your computer as a common IDE (Integrated Development Environment) such as Spyder, Notepad++, etc. But then you are limited to the computing power of the local system (laptop).

5 Set Up the Project Environment & Using Git

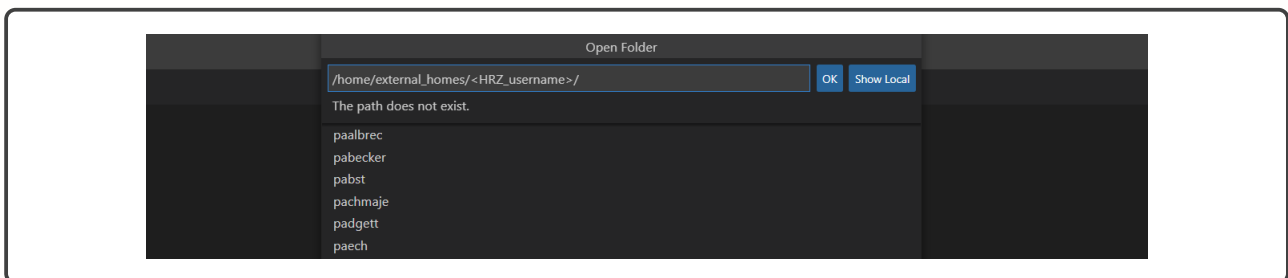
To clarify, Visual Studio Code (VSCode) is a code editor for developing, running, and debugging code, while Git is the tool that lets you create a local repository (on your PC or server) and manage versions of your files. As an optional feature, we provide below a short tutorial on how to connect to Github. Github is an free online service that will host your Git repositories (in the cloud). This allows you to access, track, and share your code with people who don't have access to the SAFE server cluster. This is of particular interest to you if you are collaborating with researchers outside of SAFE.

5.1 Create Project Directory

In this section we show how to create a working directory for your new project. When you have open the VSCode window, click on **Explorer** and then click on **Open Folder**.



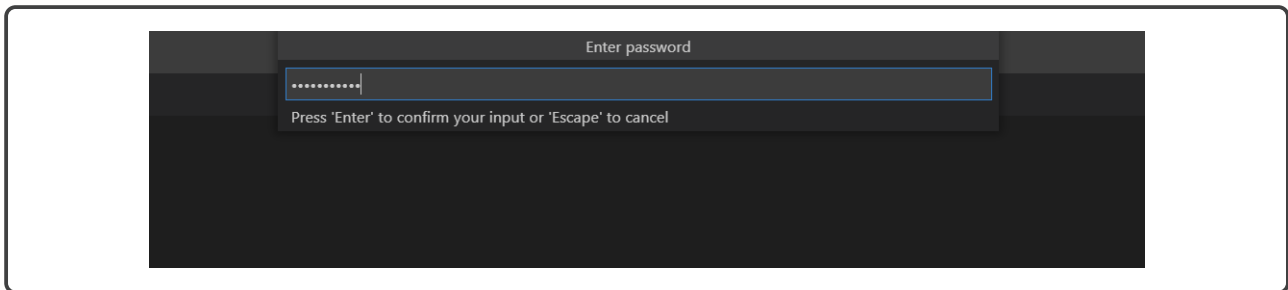
After that, a small window will appear where you can select the folder. If you are using VSCode only to access files locally, you can also use the **Show Local** button to select the folder you want to open. If you want to open a folder on the server cluster, you must actually navigate using the VSCode window (otherwise VSCode will close the connection to the server cluster after you click **Show Local**, as you can then see in the green box in the lower left corner).



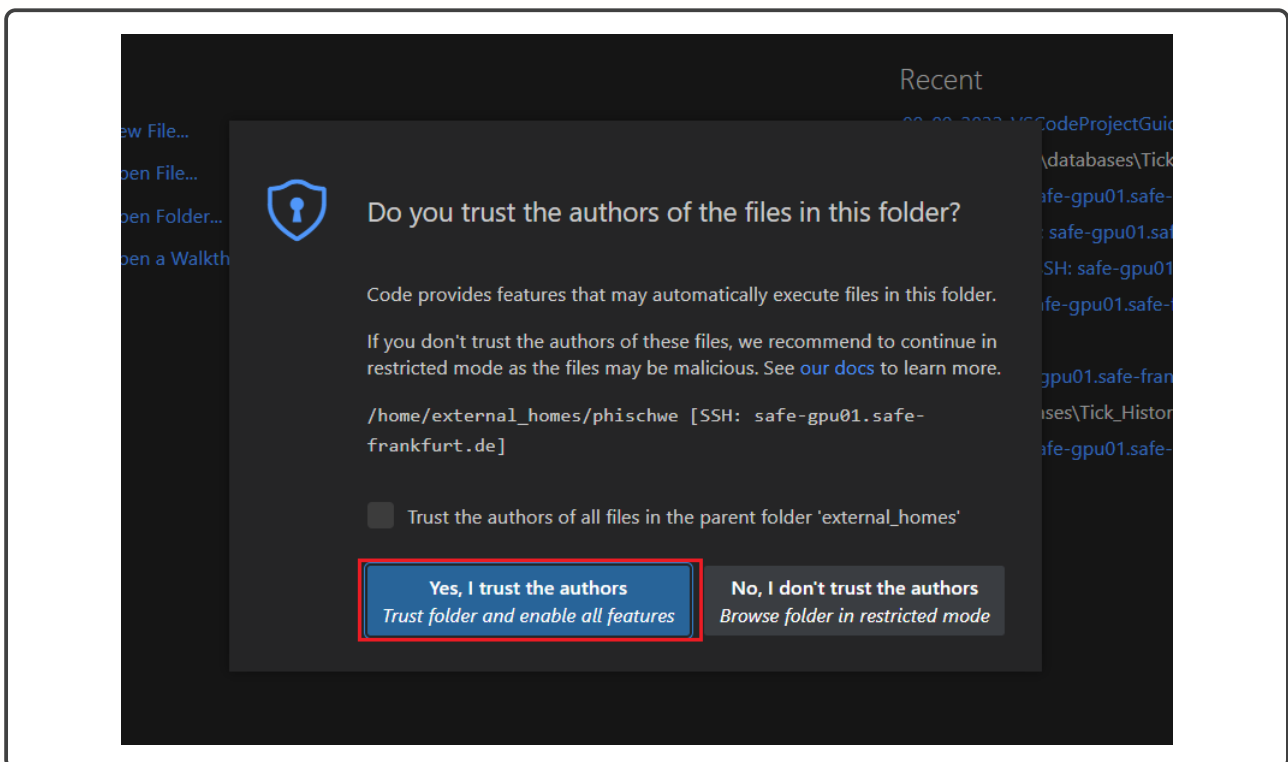
After selecting the folder `/home/external_homes/<hrz_username>` which is a relative path on the server cluster equivalent to

- path on your local Windows machine: `\\storage.safe-frankfurt.de\\<hrz_username>`
- path on your local Linux machine: `smb://storage.safe-frankfurt.de/<hrz_username>`

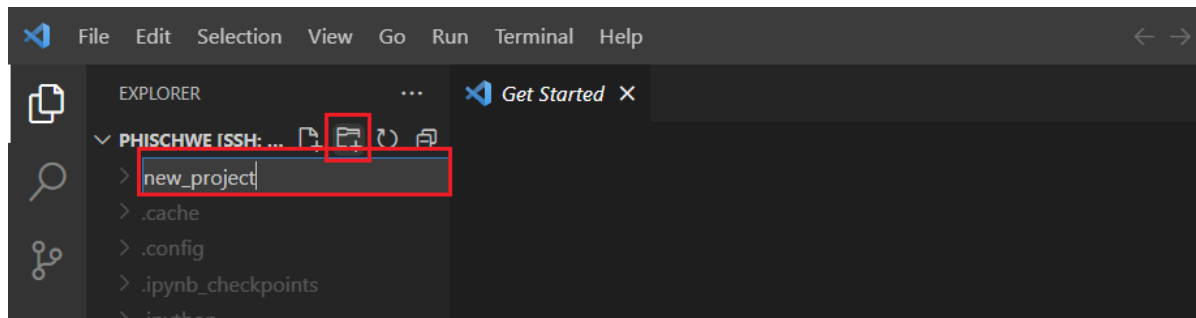
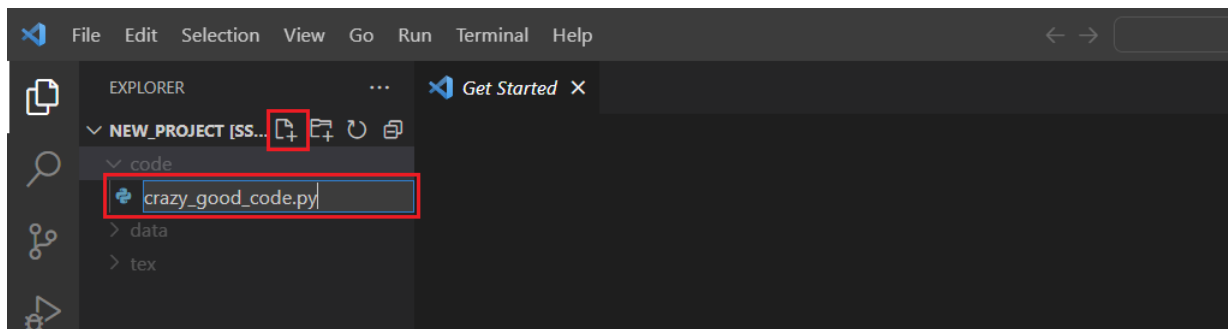
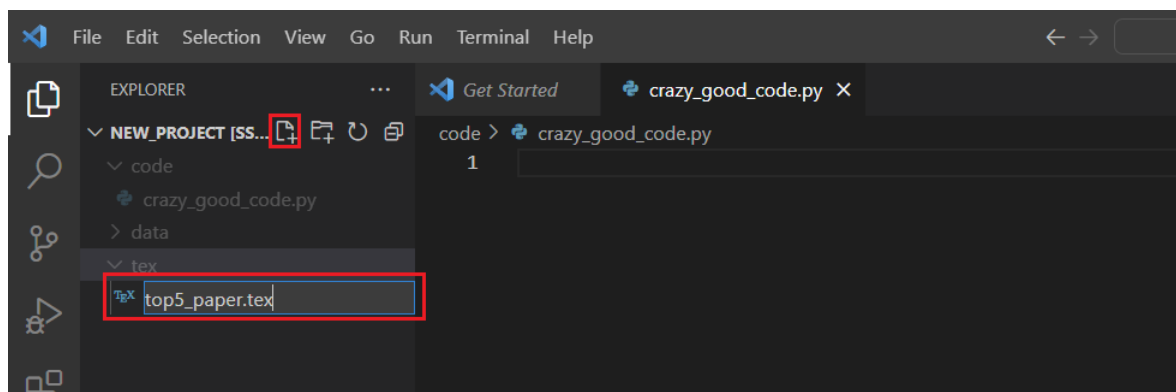
you have to enter your HRZ password again to access the selected folder.



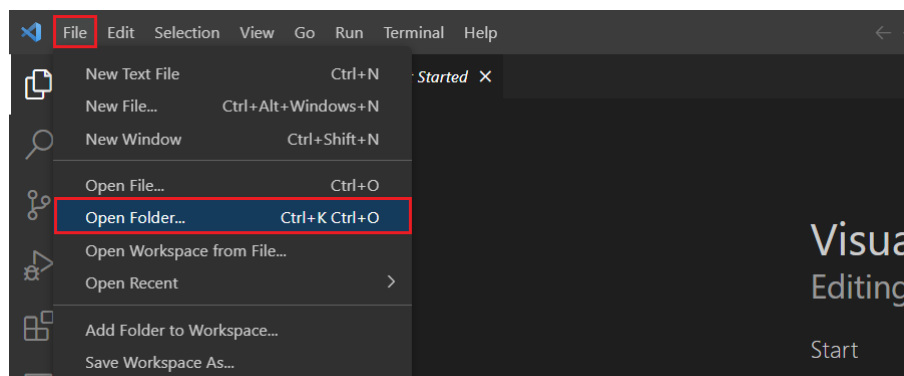
Click on **Yes, I trust the authors** and the selected folder `/home/external_homes/<hrz_username>` will be opened in the VSCode environment. The content of the folder is displayed on the left tab.

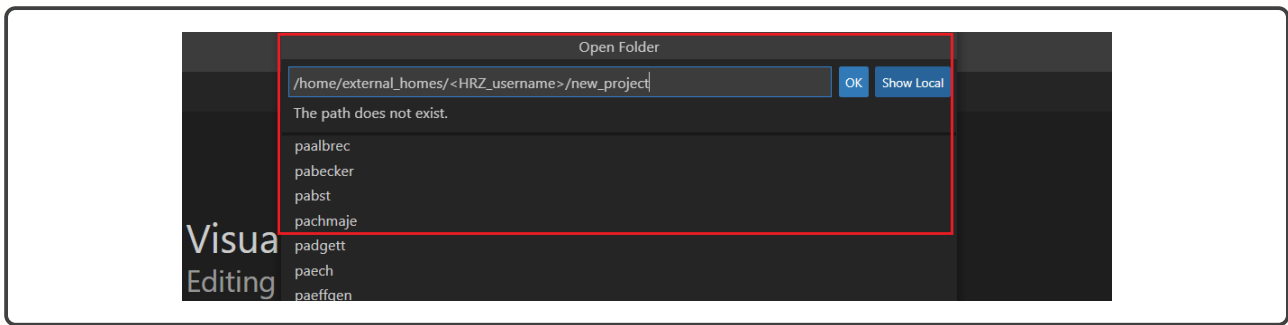


If you click on the Folder or File button, you can create new folders and files directly in the VSCode environment. Alternatively, you can always open your (Windows) Explorer and navigate in the appropriate direction to create your folders and files. When you click the Folder button, simply enter the name of your new folder (here `new_project`). If you click the File button, enter the name of the new file and the appropriate extension (here `crazy_good_code.py` or `top5_paper.tex` for a Python or Tex file).



If you want to change your folder, go to **File > Open Folder...** and then select the folder in the window that opens.





5.2 Activate Git

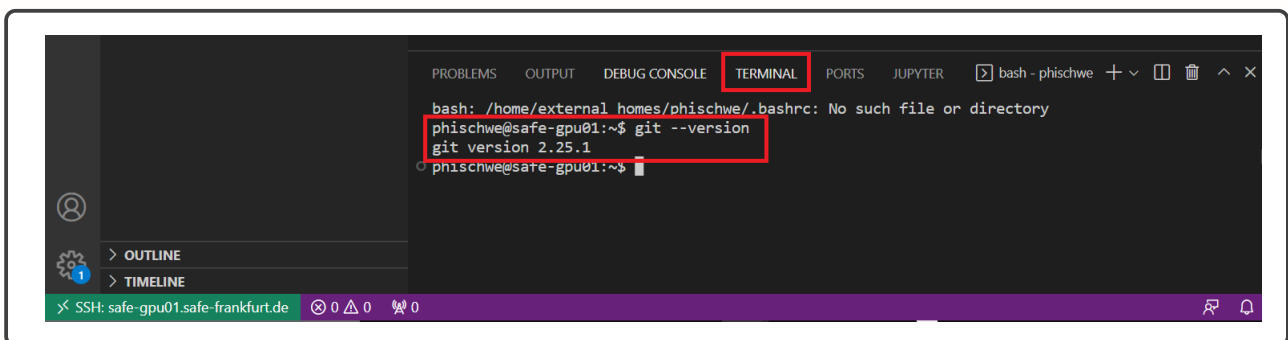
1. Install and Enable Git

Git is already installed on the server. Only if you want to use Git locally, you need to install Git on your device before you can access from VSCode. To see if Git is already installed, you can this by typing this command in the terminal

```
1 $ git --version
```

and you should see an output similar to

```
1 git version 2.37.3.windows.1
```



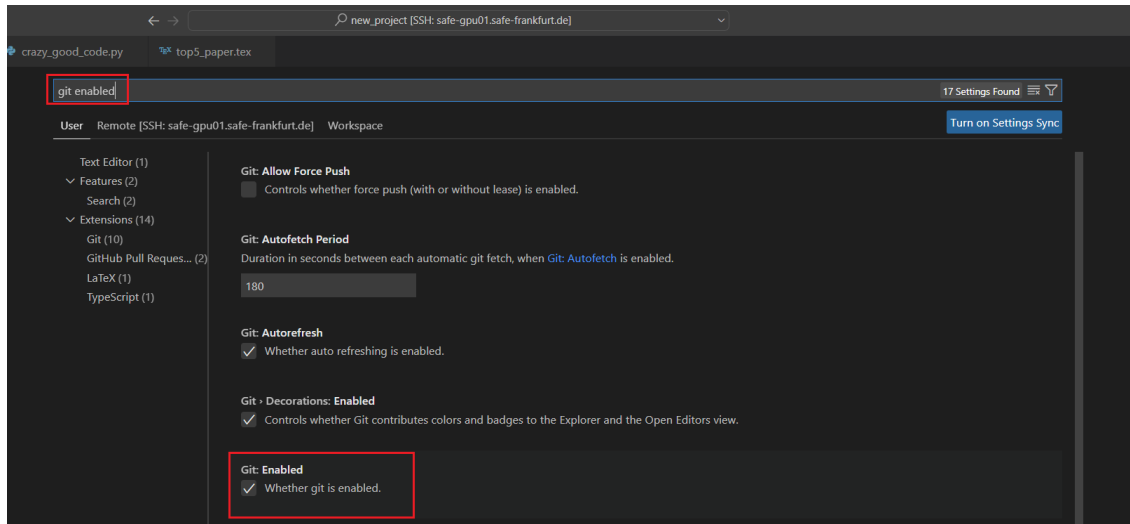
If it is not installed however, just go to [Git](#) and download the executable file for your machine.

To enable Git in VS Code on Windows:

- Go to **File > Preferences**
- Go to **Settings**
- Type **Git: Enabled** in the search bar
- Make sure that the box is ticked

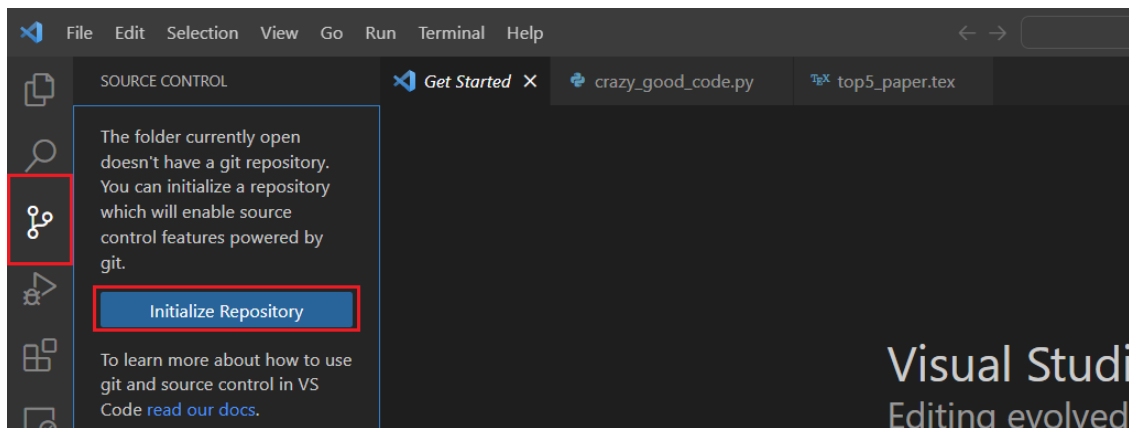
To enable Git in VS Code on Mac:

- Got to **Code > Preferences**
- Go to **Settings**
- Type **Git: Enabled** in the search bar
- Make sure that the box is ticked

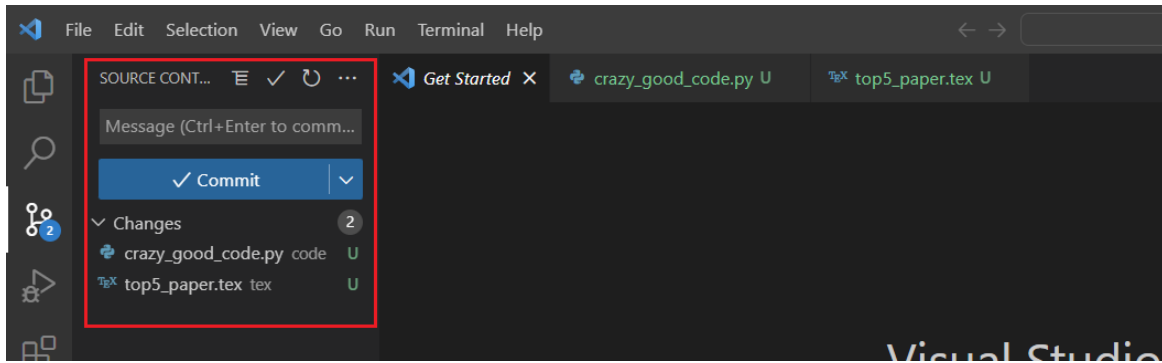


2. Activate on your Project Folder

If you have a project directory that is not currently under version control and you want to control it with Git under VSCode, you must first open the appropriate folder (see section 5.1). Then click **Source Control** in the left pane and then click **Initialize Repository**.



Then you will see all current changes in the directory on the left side. After initialization, you usually see all files in the current directory. After an initial commit, at this point you see all changes (e.g. deleting a line of code in a Python file) that happened compared to your last commit.



3. Committing Your Project

The first time you commit, you have set up your commit email address and username in Git. To do this, open the terminal in VSCode. Set up the user name in Git by

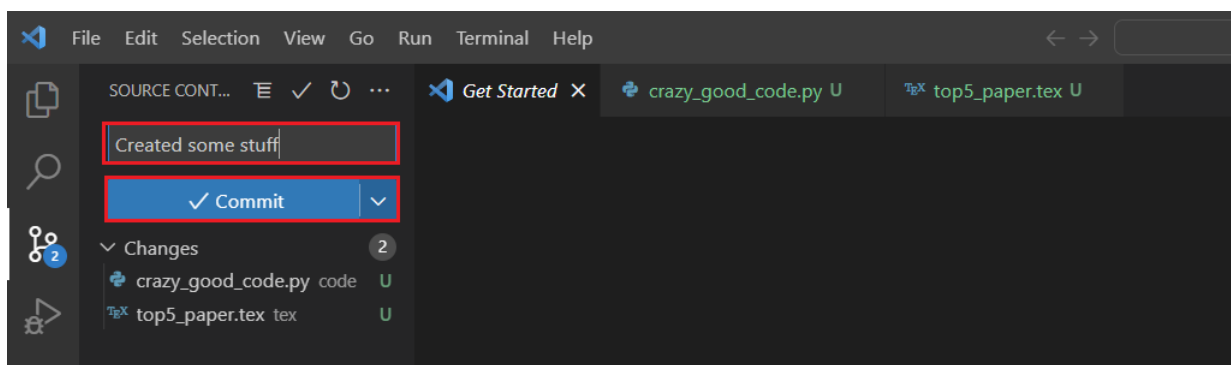
```
1 $ git config --global user.name yourusername
```

and set an email address in Git by.

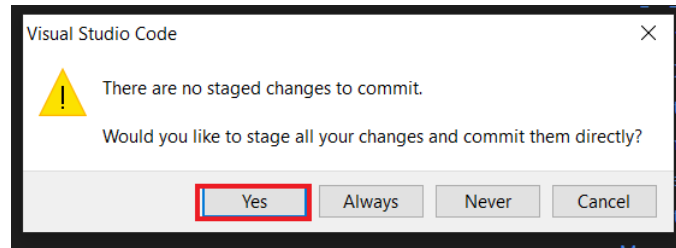
```
1 $ git config --global user.name email@yourmail.com
```



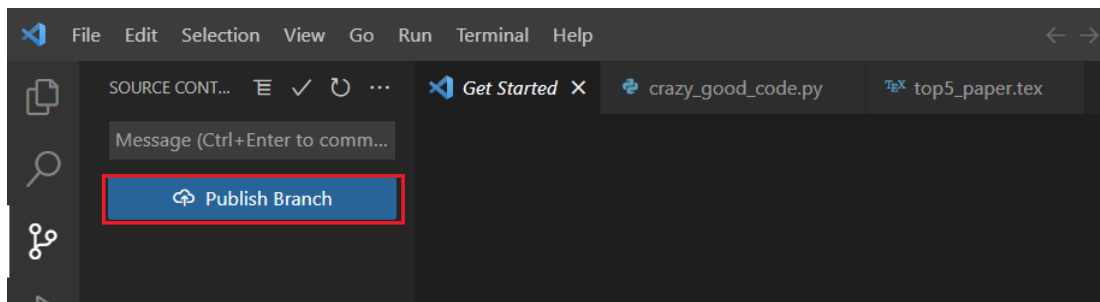
If you want to commit/save your changes, you need to enter a short note about the changes and click the **Commit** button.



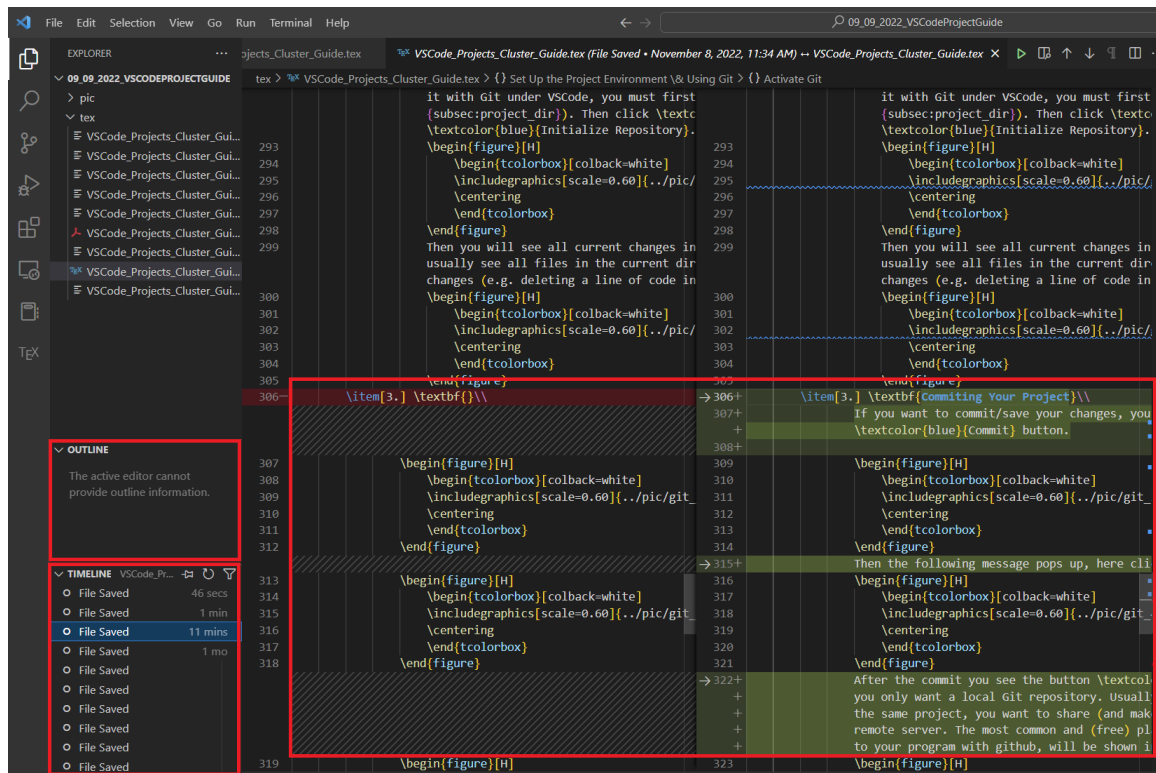
Then the following message appears, click **Yes** here.



After the commit, you will see the **Publish Branch** button. This is irrelevant to you if you only want a local Git repository. Usually, especially if you are collaborating with others (outside of SAFE) on a project, you will want to share your project on a remote server that is available outside of your organization's infrastructure (and take an additional backup of your project). The most common and (free) platform for this is [Github](#). How to connect your local repository to Github will be described in an upcoming guide. In order not to depend on an external third party code hosting platform like Github (nevertheless free and open source), Goethe University also provides the same functionality via the platform [Gitlab](#) (also described in the upcoming guide).



In general, you can compare various changes to your file in the **TIMELINE** section of VSCode, and a file outline in the **OUTLINE** section.



6 Optional: Set Up Virtual Environment to Freeze Coding Environment

Coding a project sometimes takes a lot of time, or you have a new research idea based on an old project? If so, you may often face the problem that your code from that time no longer works with newer versions of your chosen programming language. You can freeze your code at the time you start your new program and save it. Whenever you go back to your old code, you can then switch to your old programming version. We will give you an example here with Python code.

Go to the terminal at the bottom of the VSCode window and enter the following commands.

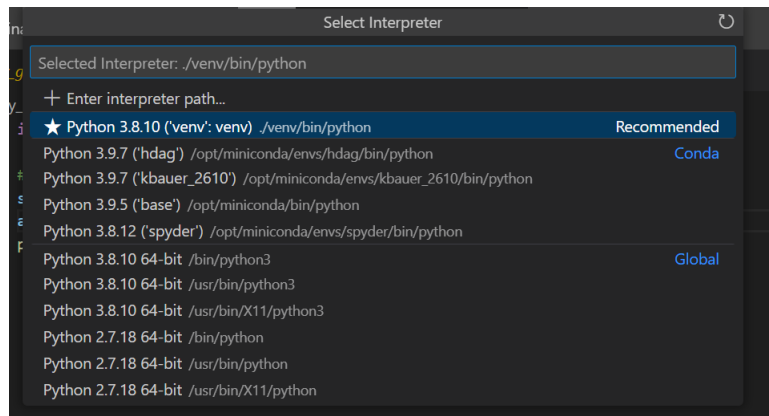
```
1 $ python3 -m venv venv
2 $ touch .gitignore
```

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER

bash: /home/external_homes/phischwe/.bashrc: No such file or directory
● phischwe@safe-gpu01:~/new_project/code$ python3 -m venv venv
● phischwe@safe-gpu01:~/new_project/code$ touch .gitignore
○ phischwe@safe-gpu01:~/new_project/code$
```

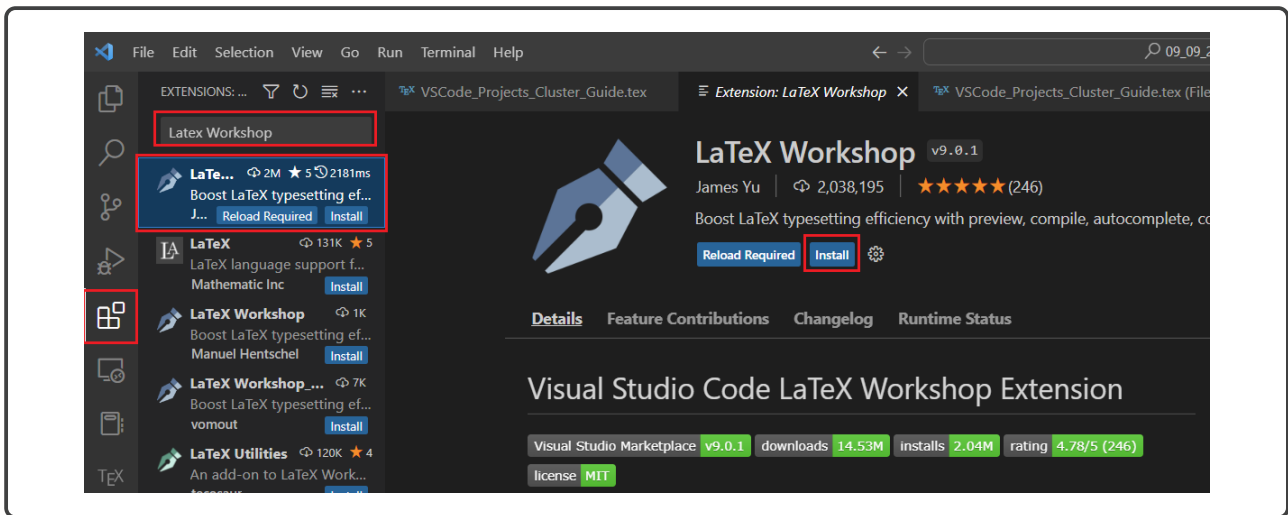
This piece of code creates what is called a virtual environment (venv), which is also called venv here. By working in this virtual environment, you decouple your current code from the running Python version on the server. For example, if the Python version on the server is upgraded (which happens from time to time without notification from the Data Center team), say to

Python 3.9, you still have the previous version available within the virtual environment, say Python 3.8. Using the touch `.gitignore` command, you create an empty file called `.gitignore`. In this file, you can add any paths that you do not want Git to consider, for example, if you do not want to track a particular folder. You should always add the following path `venv\` to the file `.gitignore`. Now you can change the Python version in the lower right corner and the following menu will be displayed. Note that the Python button at the bottom is only displayed in VSCode if you have a Python (`.py`) file open.



7 Optional: Include $\text{\LaTeX}$

To ensure that Latex works on your local machine, you must first install a Tex distribution, e.g. install [TeX Live](#). If you are creating a Tex file on the server cluster, you do not need to install a Tex distribution; this is already done by the datacenter team on the server itself. What you need to install in both cases is the Latex Workshop extension. By installing this extension, VSCode knows how to handle latex code and how to display it. To install the LaTeX Workshop extension, go to [Extensions](#) in the left panel, then search for [LaTeX Workshop](#) and click [Install](#).



8 Optional: Use Git Submodules to combine different projects

Submodules are repositories within other Git repositories. As an example, let us assume that we have 2 repositories named *skeleton* and *core*. As the names suggest, we assume that the module *skeleton* the (sub)module *core* contains. Each parent repository contains a so-called *.gitmodules* file. In this file we need to define the path to the submodule folder, the URL to the submodule git file, the submodule repository and the branch. It could look like the following:

```
1 [submodule app/styles/core]
2 path = app/styles/core
3 url = ../../repo/core.git
4 branch = master
```

We need to make sure that this configuration file is created and initialized. For this we can use the following commands in the VSCode terminal:

```
1 git submodule add -b master [URL to the git repository].
2 git submodule init
```

Always remember that submodules are not automatically updated when you update your master repository. For example, if you pull the changes in *skeleton*, you need to update the submodule separately.

To update the submodules in VSCode, switch to version control in your project and you will see the repositories. In our example, we would see the *skeleton* (main repository) and also the *core* submodule. This only works if the *.gitmodules* file has already been created, so VS Code will recognize this. Since the submodule is just another repository pointing to a specific commit, you can click the sync icon next to branches (master) and the repositories will be updated. You can also quickly check out another branch by simply clicking on the branch name. However, if you are tracking branches in your submodules and need to update another branch that has not yet been checked out, the following command might be useful:

```
1 git submodule update --remote --merge
```