

# Git

How to use GIT

- [Gitea](#)
  - [Sign-In to gitea](#)
  - [Create your first Repository](#)
  - [Configure Workspace](#)
  - [Your first steps with git](#)

# Gitea

How to work with gitea

Gitea

# Sign-In to gitea

Navigate to your providers gitea website and sign-in with your credentials.

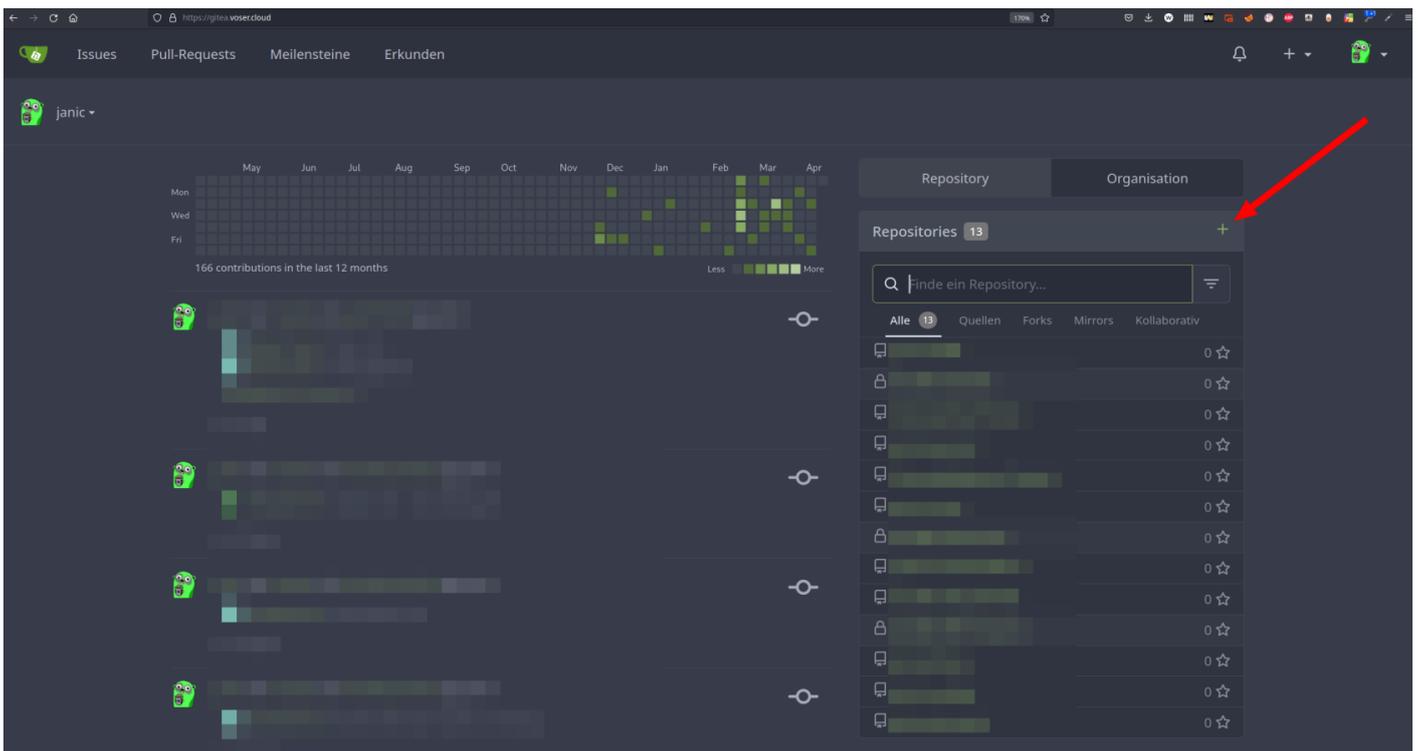
For example: <https://gitea.voser.cloud/user/login>

You may be prompted to change your password when you sign-in the first time.

# Create your first Repository

## Create a new Repository

When you are logged in you can just click on the "+" sign on the right side of repositories.



## Configure Repository

Here you find all the settings described.

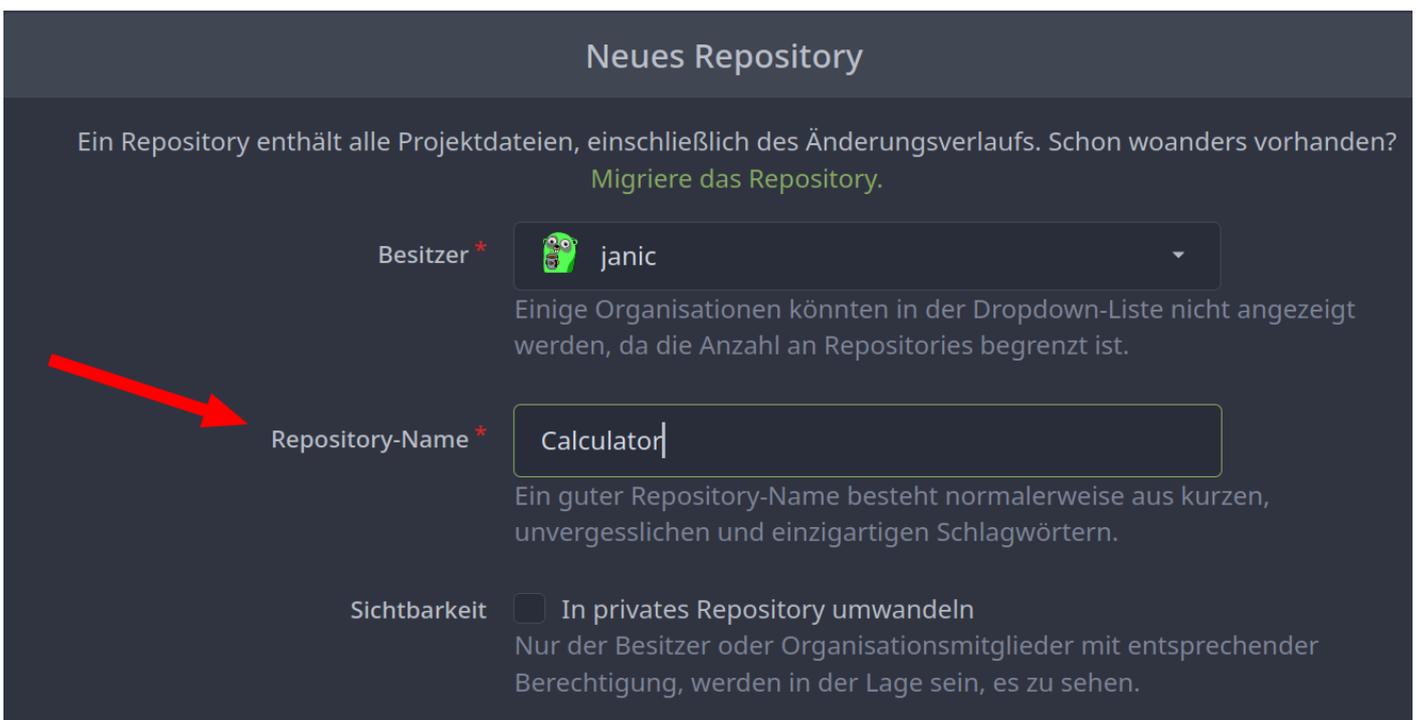
### Owner

Select the owner of the Repository, you can either select your own user or a organization, where you have the rights to create a repository.



## Repository-Name

Select a name, this can be anything.



## Private Repository

If your repository contains sensitive data, you should set the checkbox for private repository. If this checkbox is not set, everybody with access to this website could read your code. If you set this checkbox, only you and administrators can see this repository.

Repository-Name <sup>\*</sup>

Calculator

Ein guter Repository-Name besteht normalerweise aus kurzen, unvergesslichen und einzigartigen Schlagwörtern.

Sichtbarkeit

In privates Repository umwandeln

Nur der Besitzer oder Organisationsmitglieder mit entsprechender Berechtigung, werden in der Lage sein, es zu sehen.

Beschreibung

Gib eine kurze Beschreibung an (optional)



## Description

This description gets written in the README.md, this can be changed late, type a descriptive sentence.

Sichtbarkeit

In privates Repository umwandeln

Nur der Besitzer oder Organisationsmitglieder mit entsprechender Berechtigung, werden in der Lage sein, es zu sehen.

Beschreibung

This project is used to create a simple calculator which fits my needs.

Template

Vorlage auswählen



# Template

If you would have a template repository, you could use this as base. If you have none, just the default will be used.

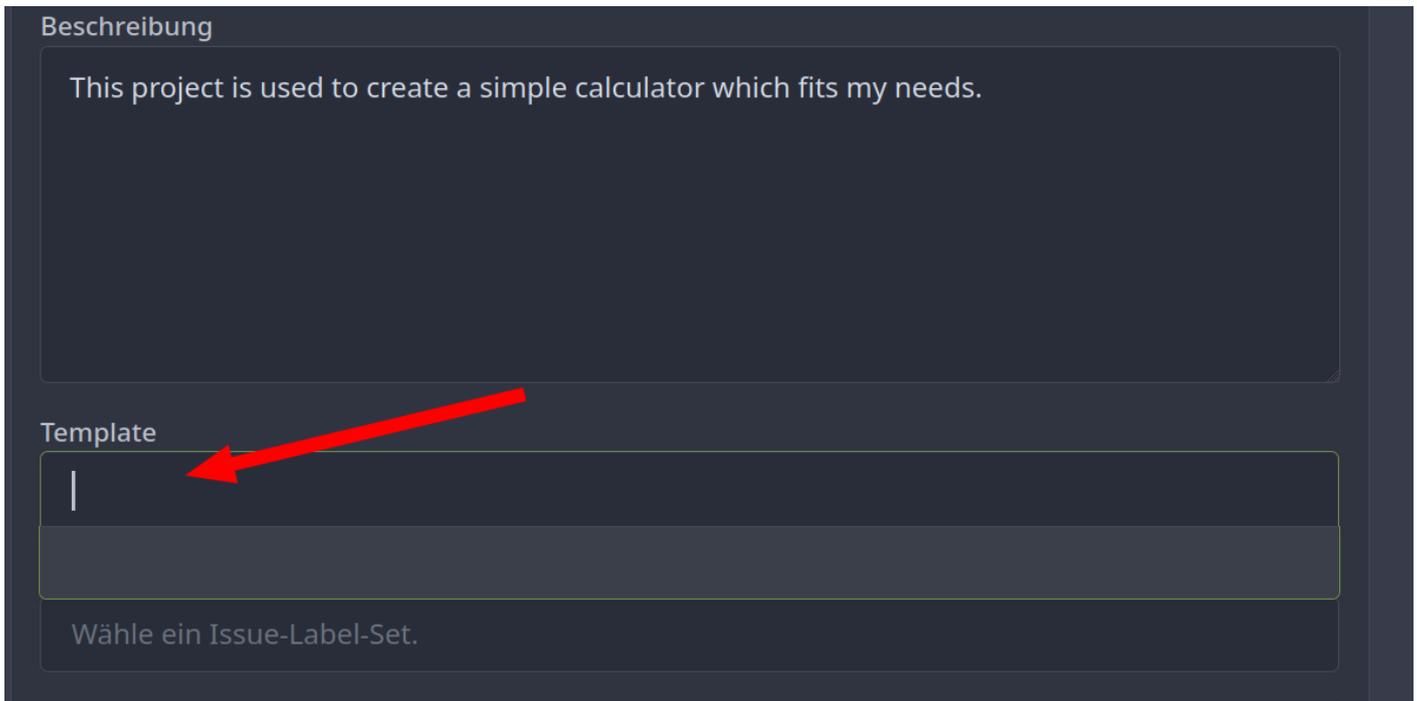
Beschreibung

This project is used to create a simple calculator which fits my needs.

Template

|

Wähle ein Issue-Label-Set.



# Issue Label

The Issue-Label-Set is used when creating a Issue, you can categorize the issues with the following tags. Just select the default.

Template

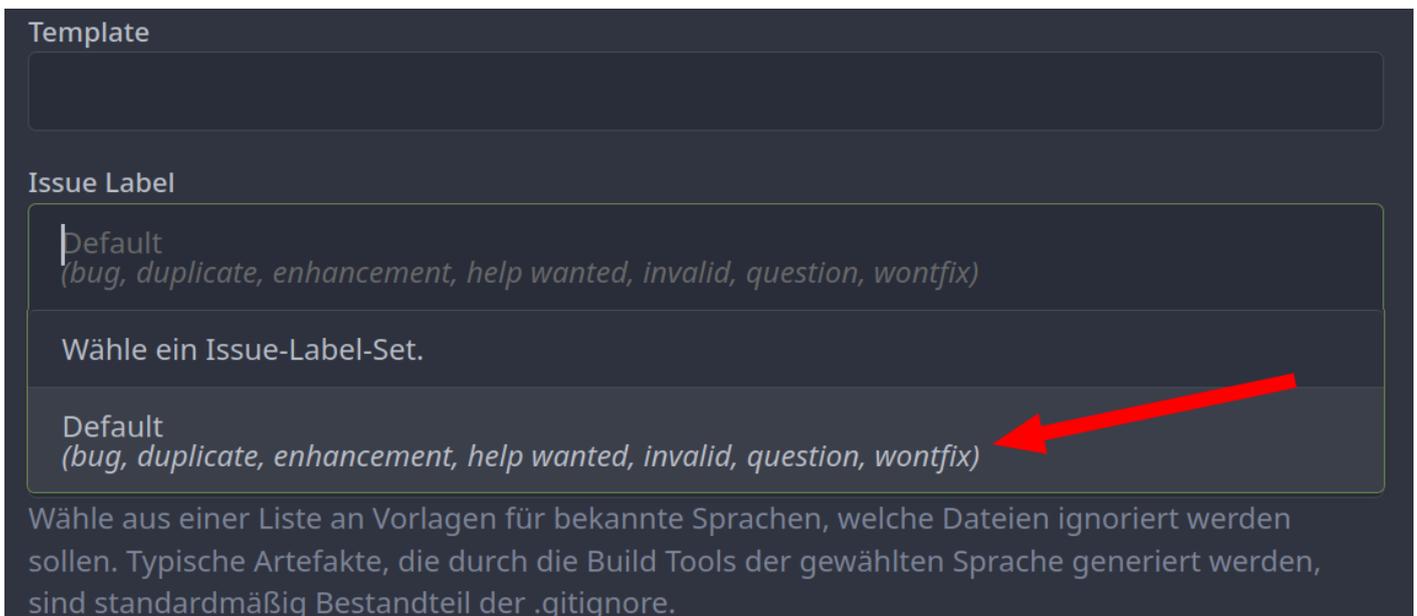
Issue Label

Default  
*(bug, duplicate, enhancement, help wanted, invalid, question, wontfix)*

Wähle ein Issue-Label-Set.

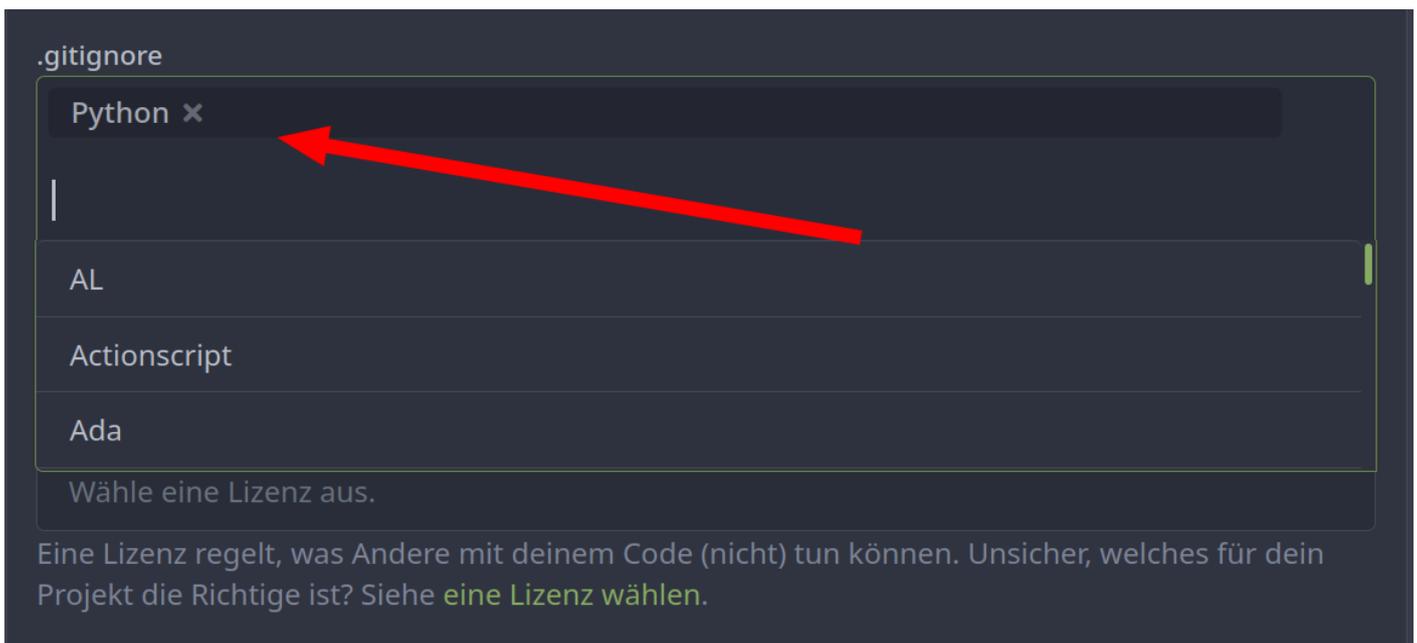
Default  
*(bug, duplicate, enhancement, help wanted, invalid, question, wontfix)*

Wähle aus einer Liste an Vorlagen für bekannte Sprachen, welche Dateien ignoriert werden sollen. Typische Artefakte, die durch die Build Tools der gewählten Sprache generiert werden, sind standardmäßig Bestandteil der .gitignore.



# Gitignore File

The .gitignore file is used to ignore some files like build binaries and cache or sensitive data. All the files defined in this file do not get uploaded to your git repository. My recommendation is to just select the default for your programming language, you can adjust this on every commit.



## License

To protect your intellectual Property you can select a license for your repository, my recommendation is to use Apache-2.0.

.gitignore

Python ✕

Wähle aus einer Liste an Vorlagen für bekannte Sprachen, welche Dateien ignoriert werden sollen. Typische Artefakte, die durch die Build Tools der gewählten Sprache generiert werden, sind standardmäßig Bestandteil der .gitignore.

Lizenz

Apache-2.0

Apache-2.0

App-s2p

Artistic-1.0

Hier kannst du eine komplette Beschreibung für dein Projekt schreiben.

Repository initialisieren (Fügt .gitignore, License und README-Dateien hinzu)

## README

The README file is a markdown file in your repository, which has the description in it. you can customize this to your needs.

Additionally check the checkbox to initialize your repository, otherwise it will not create the files.

Lizenz

Apache-2.0

Eine Lizenz regelt, was Andere mit deinem Code (nicht) tun können. Unsicher, welches für dein Projekt die Richtige ist? Siehe [eine Lizenz wählen](#).

README

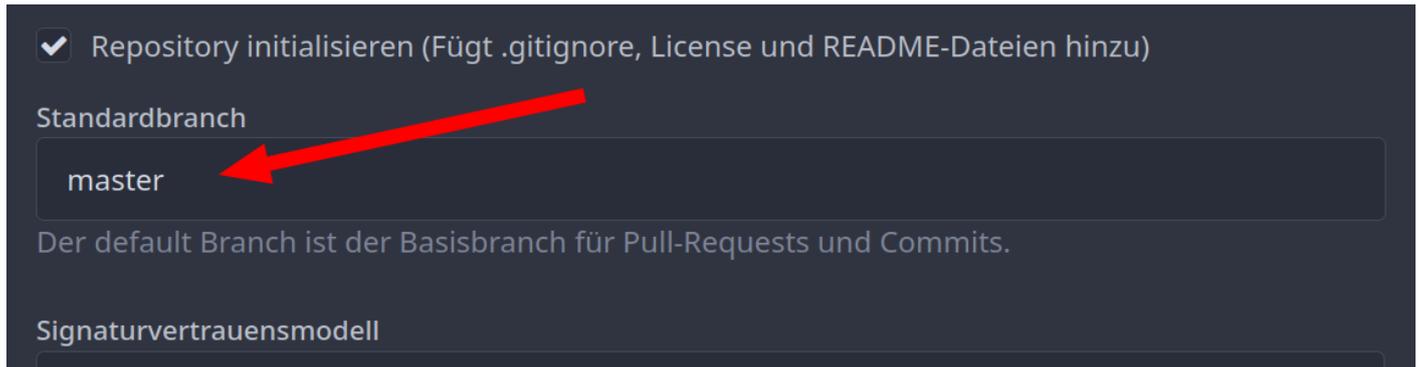
Default

Hier kannst du eine komplette Beschreibung für dein Projekt schreiben.

Repository initialisieren (Fügt .gitignore, License und README-Dateien hinzu)

## Defaultbranch

The default branch describes to which branch the push request are getting routed. Use master for default.



Repository initialisieren (Fügt .gitignore, License und README-Dateien hinzu)

Standardbranch

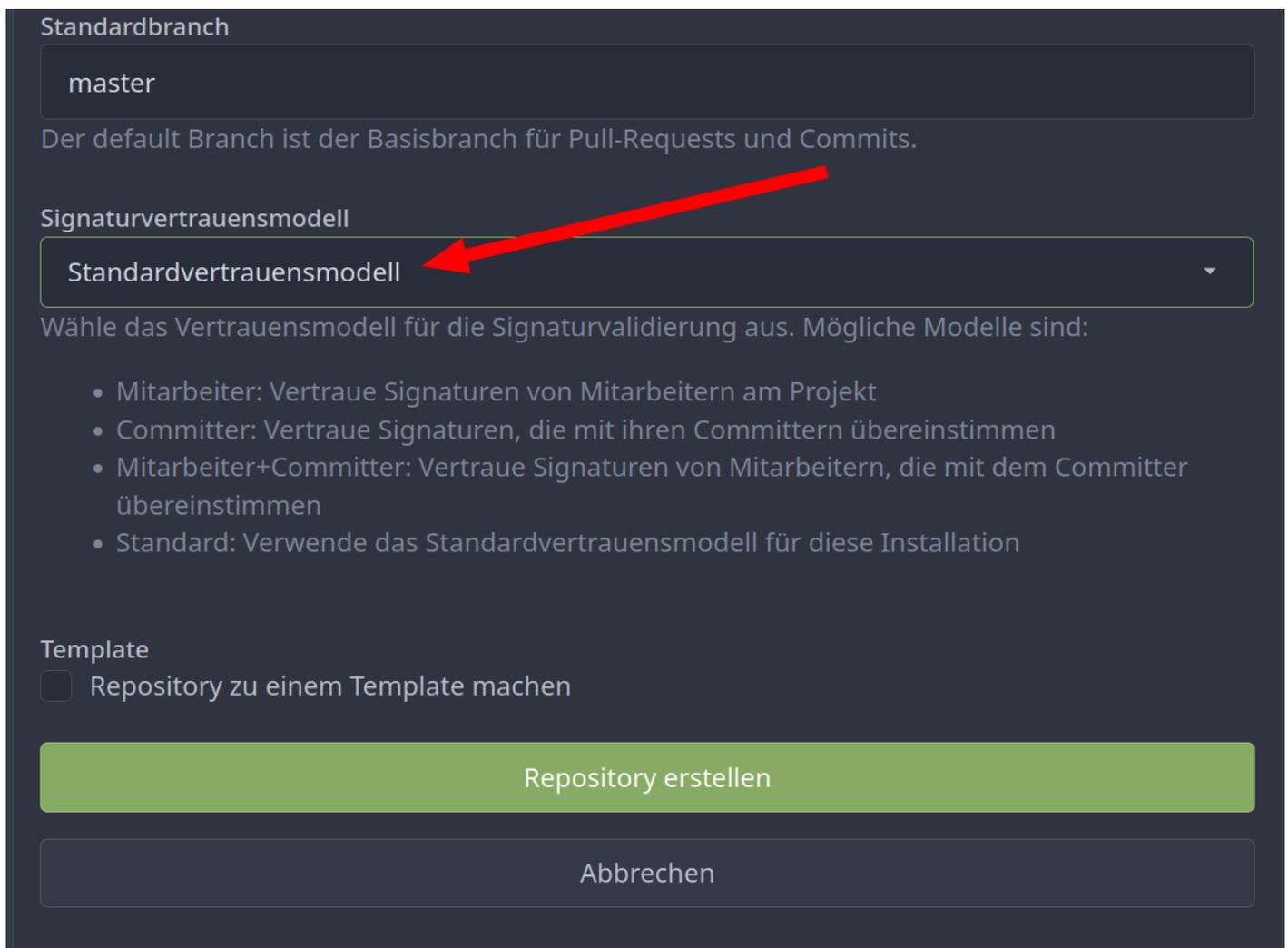
master

Der default Branch ist der Basisbranch für Pull-Requests und Commits.

Signaturvertrauensmodell

## Default trust model

The default trust model describes how the trusts are related. Use the default.



Standardbranch

master

Der default Branch ist der Basisbranch für Pull-Requests und Commits.

Signaturvertrauensmodell

Standardvertrauensmodell

Wähle das Vertrauensmodell für die Signaturvalidierung aus. Mögliche Modelle sind:

- Mitarbeiter: Vertraue Signaturen von Mitarbeitern am Projekt
- Committer: Vertraue Signaturen, die mit ihren Committern übereinstimmen
- Mitarbeiter+Committer: Vertraue Signaturen von Mitarbeitern, die mit dem Committer übereinstimmen
- Standard: Verwende das Standardvertrauensmodell für diese Installation

Template

Repository zu einem Template machen

Repository erstellen

Abbrechen

## Template

You could convert this repository settings to a template, but remember a template can not be edited. If you just want a Project to work on do not check this box.

Signaturvertrauensmodell

Standardvertrauensmodell

Wähle das Vertrauensmodell für die Signaturvalidierung aus. Mögliche Modelle sind:

- Mitarbeiter: Vertraue Signaturen von Mitarbeitern am Projekt
- Committer: Vertraue Signaturen, die mit ihren Committern übereinstimmen
- Mitarbeiter+Committer: Vertraue Signaturen von Mitarbeitern, die mit dem Committer übereinstimmen
- Standard: Verwende das Standardvertrauensmodell für diese Installation

Template

Repository zu einem Template machen

Repository erstellen

Abbrechen

Afterwards click on "create repository".

## The final result

The final result looks something like this.

janic / Calculator

Beobachten beenden 1 Favorisieren 0 Fork 0

<> Code Issues Pull-Requests Projekte Releases Wiki Aktivität Einstellungen

This project is used to create a simple calculator which fits my needs.

Themen verwalten

1 Commit 1 Branch 0 Tags 102 KiB

Branch: master Neuer Pull-Request

Neue Datei Datei hochladen HTTPS SSH ssh://git@gitea.voser.cloud:22

Janic Voser 09db48bb58 Initial commit vor 1 Sekunde

.gitignore Initial commit vor 1 Sekunde

LICENSE Initial commit vor 1 Sekunde

README.md Initial commit vor 1 Sekunde

README.md

### Calculator

This project is used to create a simple calculator which fits my needs.

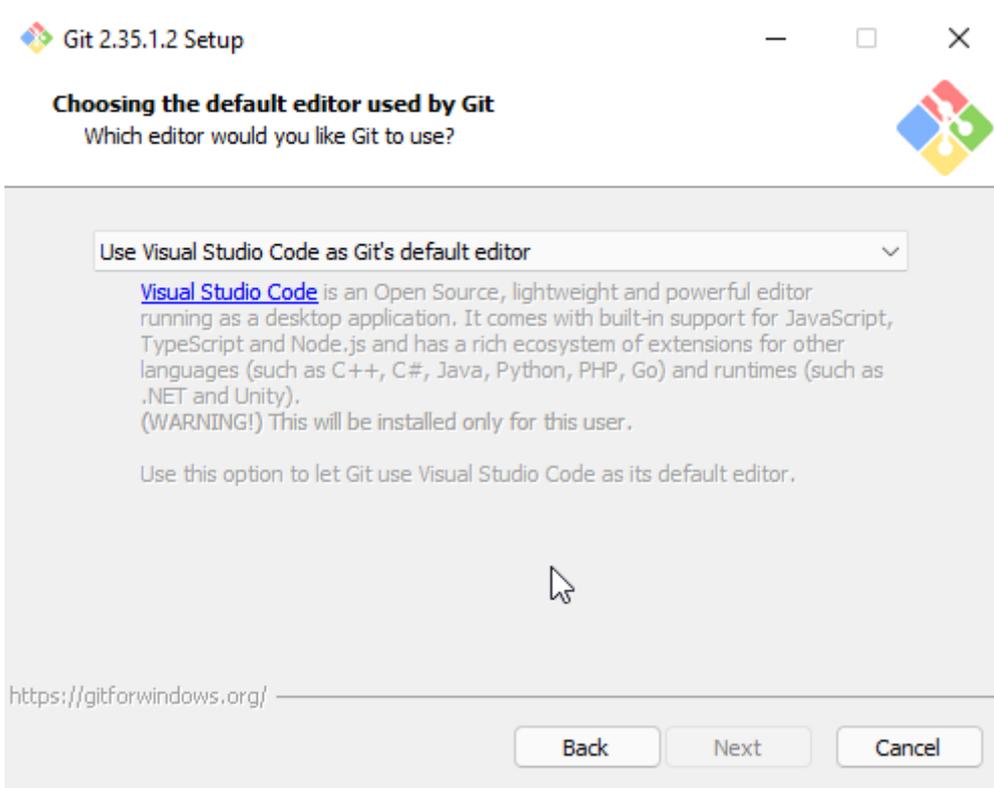


# Configure Workspace

## Install git for windows (Git Bash)

<https://gitforwindows.org/>

You can use everywhere the default settings except for the default editor, select there "Use Visual Studio code as Git's default editor"



## Generate an ssh-key

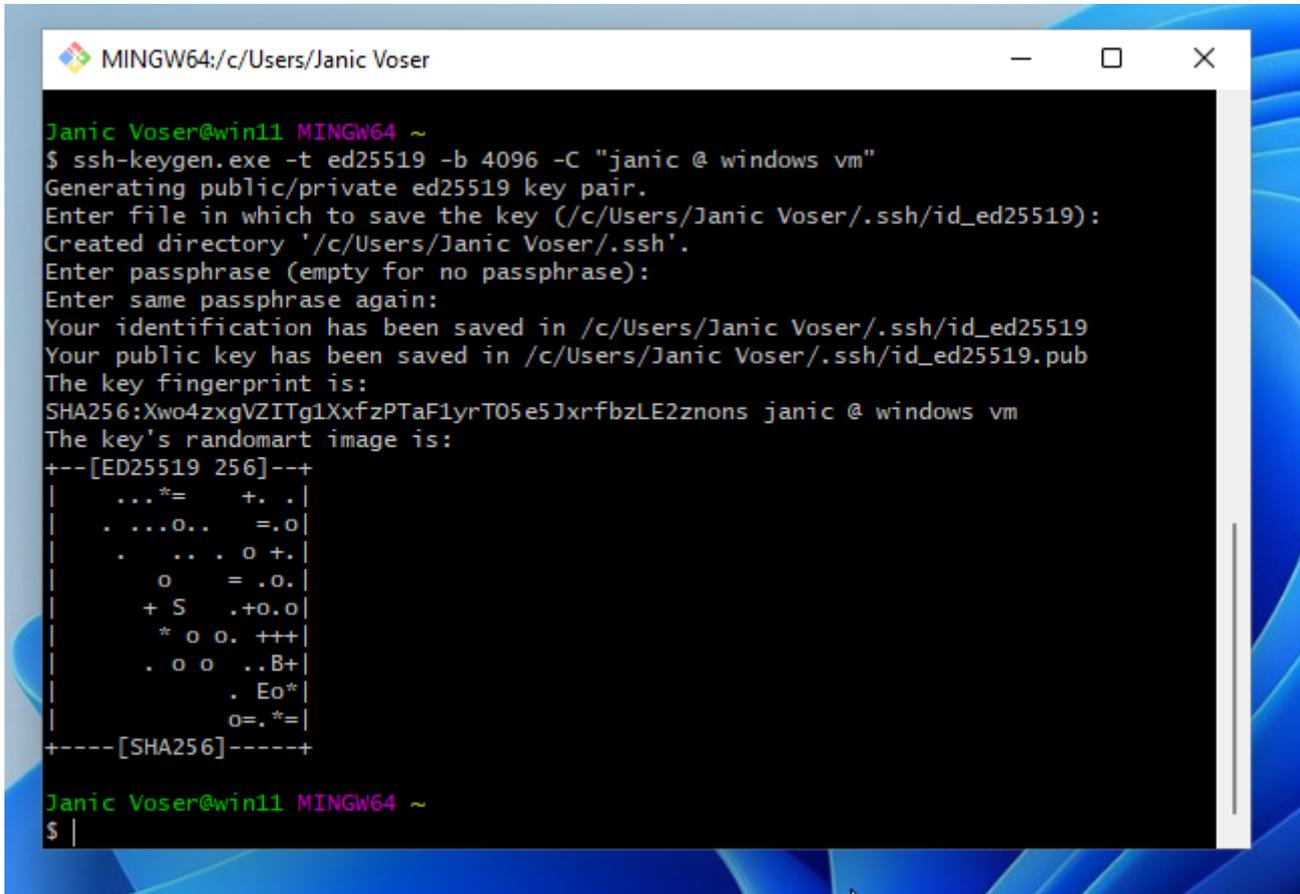
1. Open "Git Bash" and type the following command to create an SSH-Key

```
ssh-keygen.exe -t <type> -b <bit ammount> -c <Your identifying comment>
```

# For example

```
ssh-keygen.exe -t ed25519 -b 4096 -C "janic @ windows pc"
```

2. Now you are prompted to enter the path to the key. You can use the default. (just press enter and do not enter anything)
3. Now you are prompted to enter a password for your key. The default is no password. (just press enter and do not enter anything)
4. Now you are prompted to confirm your password, if you have entered none, just hit enter again.



```
MINGW64:/c/Users/Janic Voser
Janic Voser@win11 MINGW64 ~
$ ssh-keygen.exe -t ed25519 -b 4096 -C "janic @ windows vm"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/Janic Voser/.ssh/id_ed25519):
Created directory '/c/Users/Janic Voser/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Janic Voser/.ssh/id_ed25519
Your public key has been saved in /c/Users/Janic Voser/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:Xwo4zxcgVZITg1XxfzPTaF1yrT05e5JxrfbzLE2znons janic @ windows vm
The key's randomart image is:
+--[ED25519 256]--+
|    ..*=    +. .|
| . ...0..   =.0|
| .  .. . 0 +.|
|    o    = .o.|
| + S    .+0.o|
| * o o.  +++|
| . o o  ..B+|
| . Eo*|
|    o=. *|=|
+-----[SHA256]-----+
Janic Voser@win11 MINGW64 ~
$
```

# Always use your custom ssh-key

1. Navigate to your home directory with the command `cd`
2. Navigate in your ssh config directory with the command `cd .ssh`
3. List the content of this directory with the command `ls` you should now see you custom key in here.
4. Create a config file which defines to always use your custom ssh-key you can use the command below

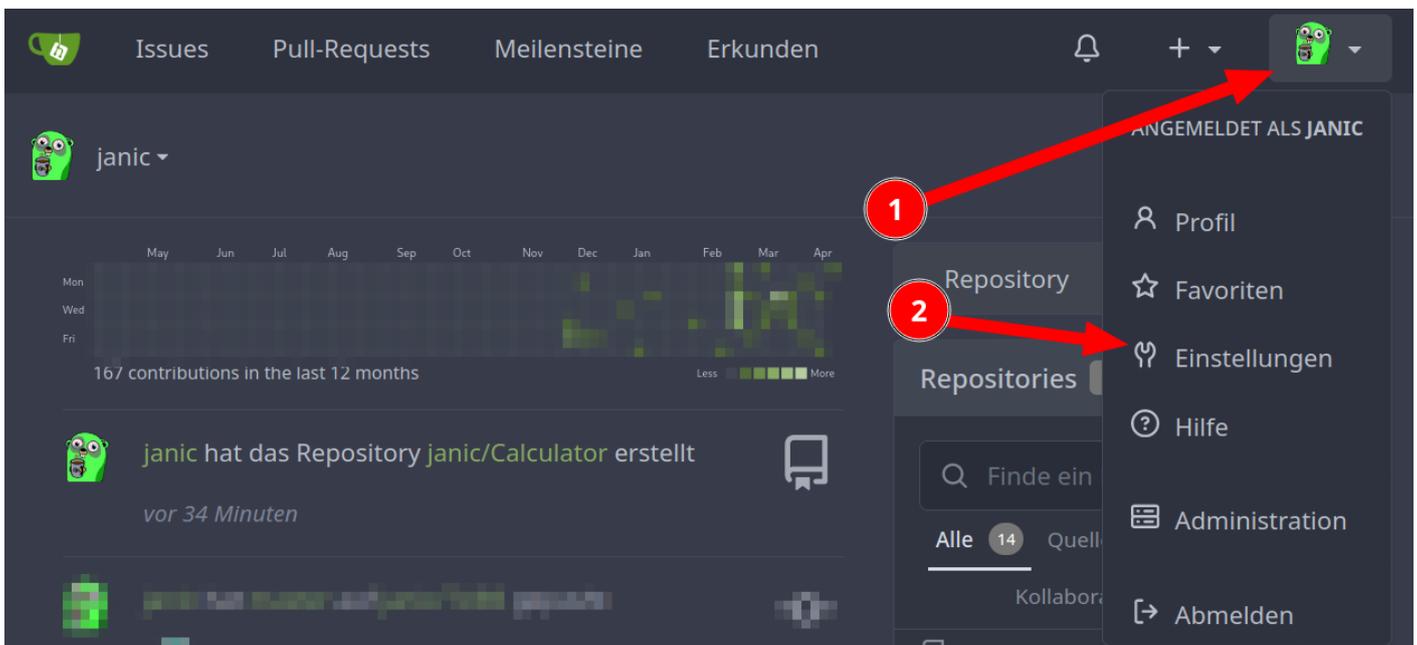
```
echo "IdentityFile ~/.ssh/<your ssh key>" > config
# For example:
```

```
echo "IdentityFile ~/.ssh/id_ed25519" > config
```

```
Janic Voser@win11 MINGW64 ~  
$ cd .ssh  
  
Janic Voser@win11 MINGW64 ~/.ssh  
$ ls  
id_ed25519 id_ed25519.pub  
  
Janic Voser@win11 MINGW64 ~/.ssh  
$ echo "IdentityFile ~/.ssh/id_ed25519" > config  
  
Janic Voser@win11 MINGW64 ~/.ssh
```

# Add your Public SSH-Key to Gitea

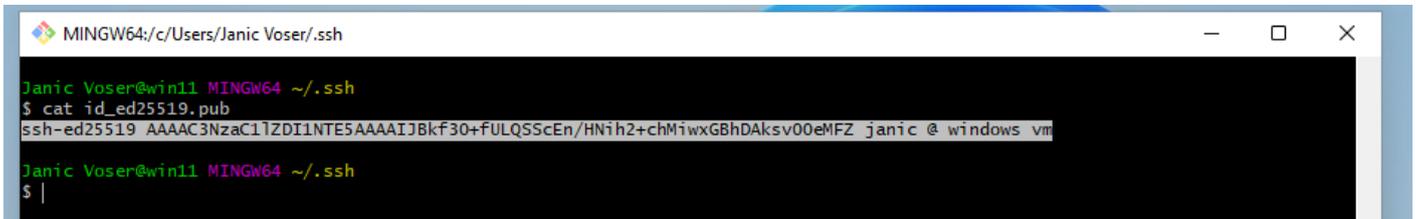
1. Login to your gitea account
2. Click in the top right on your profile and there on "Settings"



3. Get the content of your public key

You can use the cat command to open the content of the public key.

```
cat <your public key>  
# For example:  
cat id_ed25519.pub
```



```
MINGW64:/c/Users/Janic Voser/.ssh
Janic Voser@win11 MINGW64 ~/.ssh
$ cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC11ZDI1NTE5AAAAIJBkf30+fULQSScEn/HNih2+chMiwxGBhDAksv00eMFZ janic @ windows vm
Janic Voser@win11 MINGW64 ~/.ssh
$ |
```

4. Follow the steps below

Go to "SSH- / GPG-Key"

Click on "add key"

Enter the name of this key

Enter the content of the public key

Click on "Add Key"

The screenshot shows the GitHub 'SSH / GPG-Schlüssel' management page. It features a navigation bar at the top with links for 'Issues', 'Pull-Requests', 'Meilensteine', and 'Erkunden'. Below this is a secondary navigation bar with 'Profil', 'Account', 'Erscheinung', 'Sicherheit', 'Anwendungen', 'SSH- / GPG-Schlüssel', 'Organisationen', and 'Repositories'. The main content area is titled 'SSH-Schlüssel verwalten' and includes a 'Schlüssel hinzufügen' button. A form for adding a new key is visible, with fields for 'Schlüsselname' (containing 'janic @ windows vm') and 'Inhalt' (containing an SSH key). Below the form are 'Schlüssel hinzufügen' and 'Abbrechen' buttons. A list of existing keys is shown below, each with 'Entfernen' and 'Verifizieren' buttons. At the bottom, there are links for help and a 'GPG-Schlüssel verwalten' section.

1. Click on the 'SSH- / GPG-Schlüssel' tab in the top navigation bar.

2. Click on the 'Schlüssel hinzufügen' button in the top right of the main content area.

3. Click on the 'Schlüsselname' input field.

4. Click on the 'Inhalt' text area.

5. Click on the 'Schlüssel hinzufügen' button at the bottom of the form.

# Configure Git

You need to set a default username and email for git. You can do this with the following commands

```
git config --global user.name "Name"  
git config --global user.email "e@mail.com"  
#For example:  
git config --global user.name "Janic Voser"
```

```
git config --global user.email "janic@voser.cloud"
```

```
Janic Voser@win11 MINGW64 ~/.ssh
$ git config --global user.name "Janic Voser"

Janic Voser@win11 MINGW64 ~/.ssh
$ git config --global user.email "janic@voser.cloud"

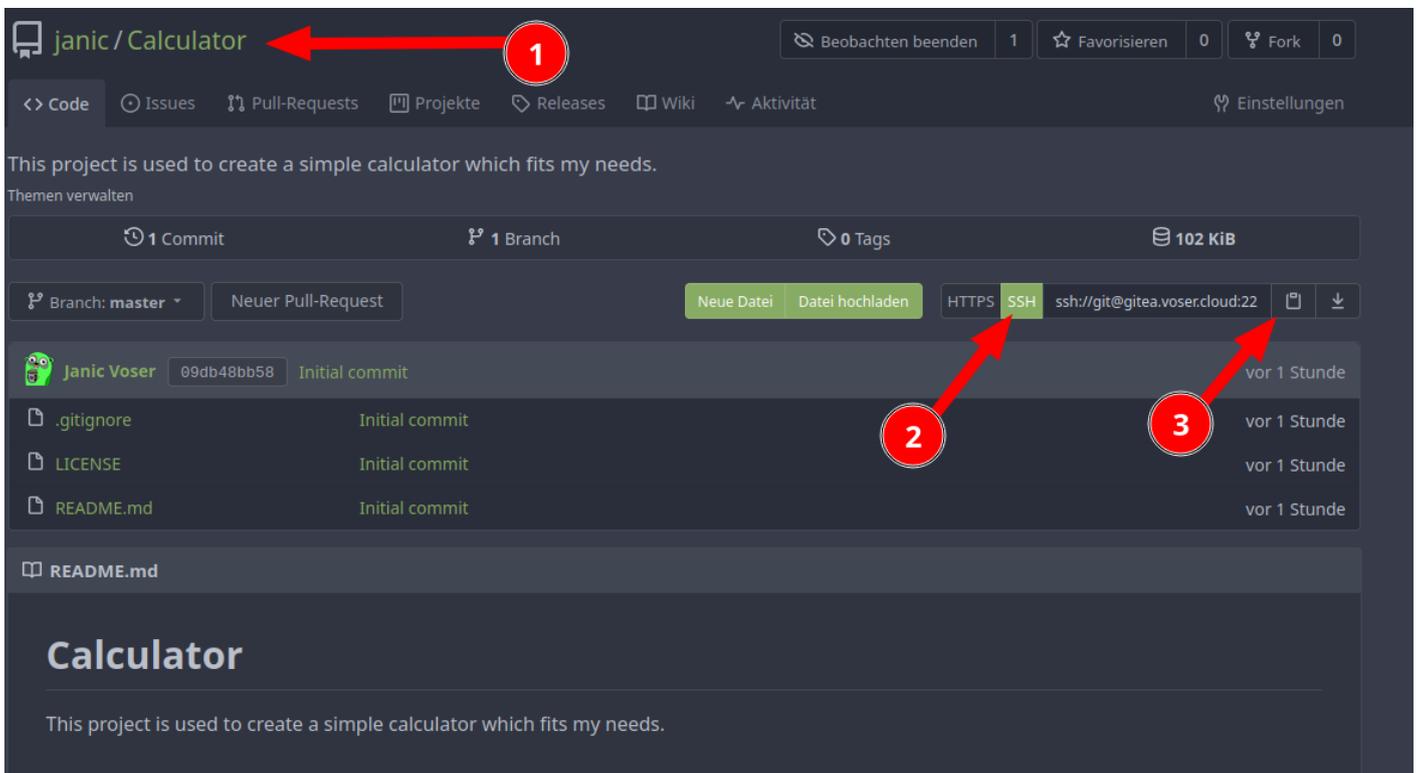
Janic Voser@win11 MINGW64 ~/.ssh
$ |
```

# Your first steps with git

## Clone your git repository

### Get the git url

Navigate to your git repository and clone the git url



## Navigate to your workspace

Navigate to your workspace directory.

```
MINGW64:/c/Users/Janic Voser/workspace
Janic Voser@win11 MINGW64 ~
$ cd
Janic Voser@win11 MINGW64 ~
$ cd workspace/
Janic Voser@win11 MINGW64 ~/workspace
$ |
```

## Clone the repository

You can clone the repository by using the git clone command.

```
git clone <git url>
# For example:
git clone ssh://git@gitea.voser.cloud:2222/janic/Calculator.git
```

If you are prompted to accept to ssh-fingerprint type yes and hit enter

```
Janic Voser@win11 MINGW64 ~/workspace
$ git clone ssh://git@gitea.voser.cloud:2222/janic/Calculator.git
Cloning into 'Calculator'...
The authenticity of host '[gitea.voser.cloud]:2222 ([172.22.0.32]:2222)' can't be established.
ED25519 key fingerprint is SHA256:X1Cehj/DamEowBwC3ppqhe3ioGCjNtNZxoHizj2omGU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
warning: Permanently added [gitea.voser.cloud]:2222 (ED25519) to the list of known hosts.
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), 5.33 KiB | 5.33 MiB/s, done.
Janic Voser@win11 MINGW64 ~/workspace
$ |
```

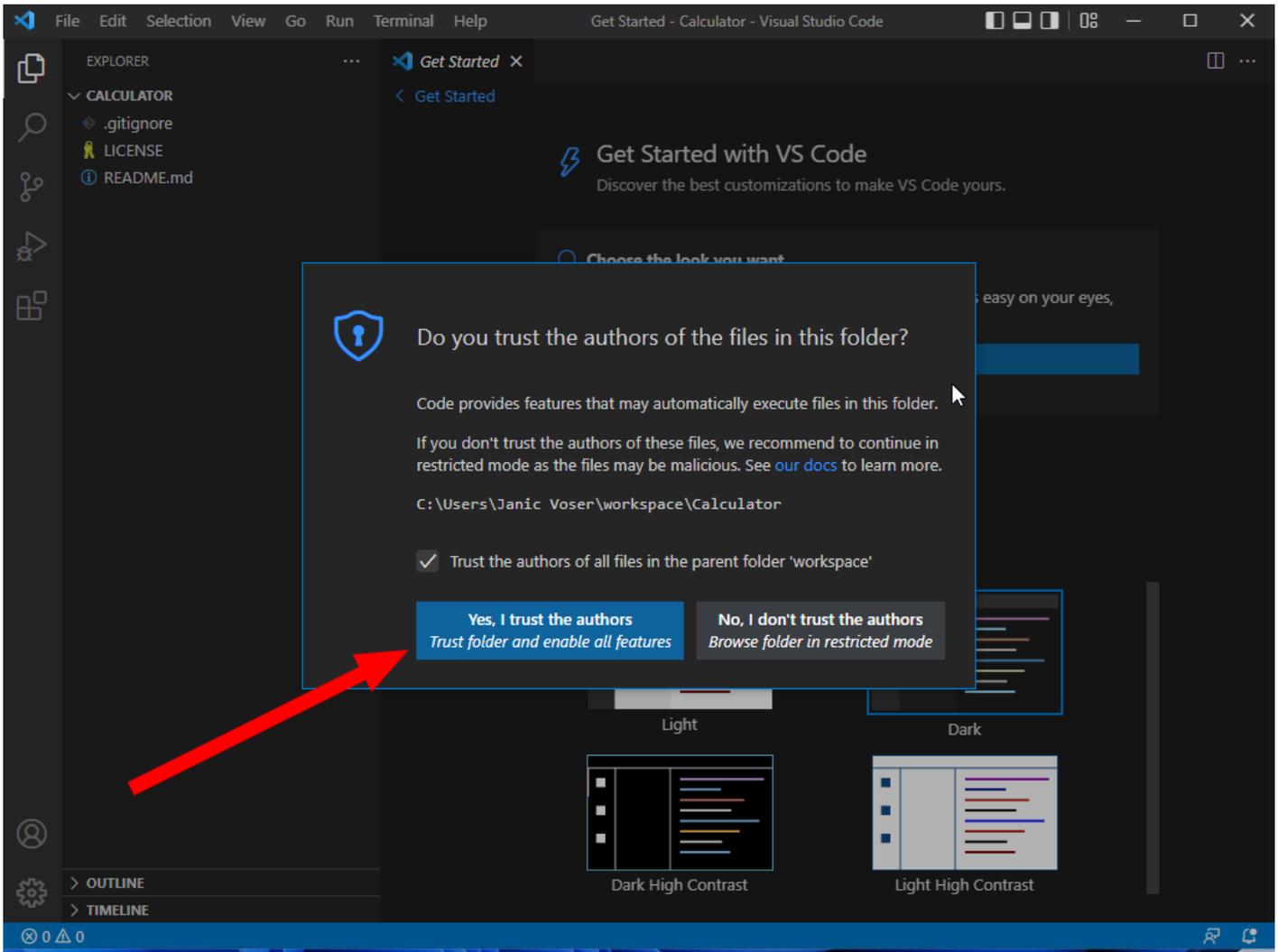
## Open repository in your editor

Navigate into your repository ( `cd "your repository name"` / `cd Calculator` )

Open this directory in VScode with the following command `code .`

```
Janic Voser@win11 MINGW64 ~/workspace
$ cd Calculator/
Janic Voser@win11 MINGW64 ~/workspace/Calculator (master)
$ code .
```

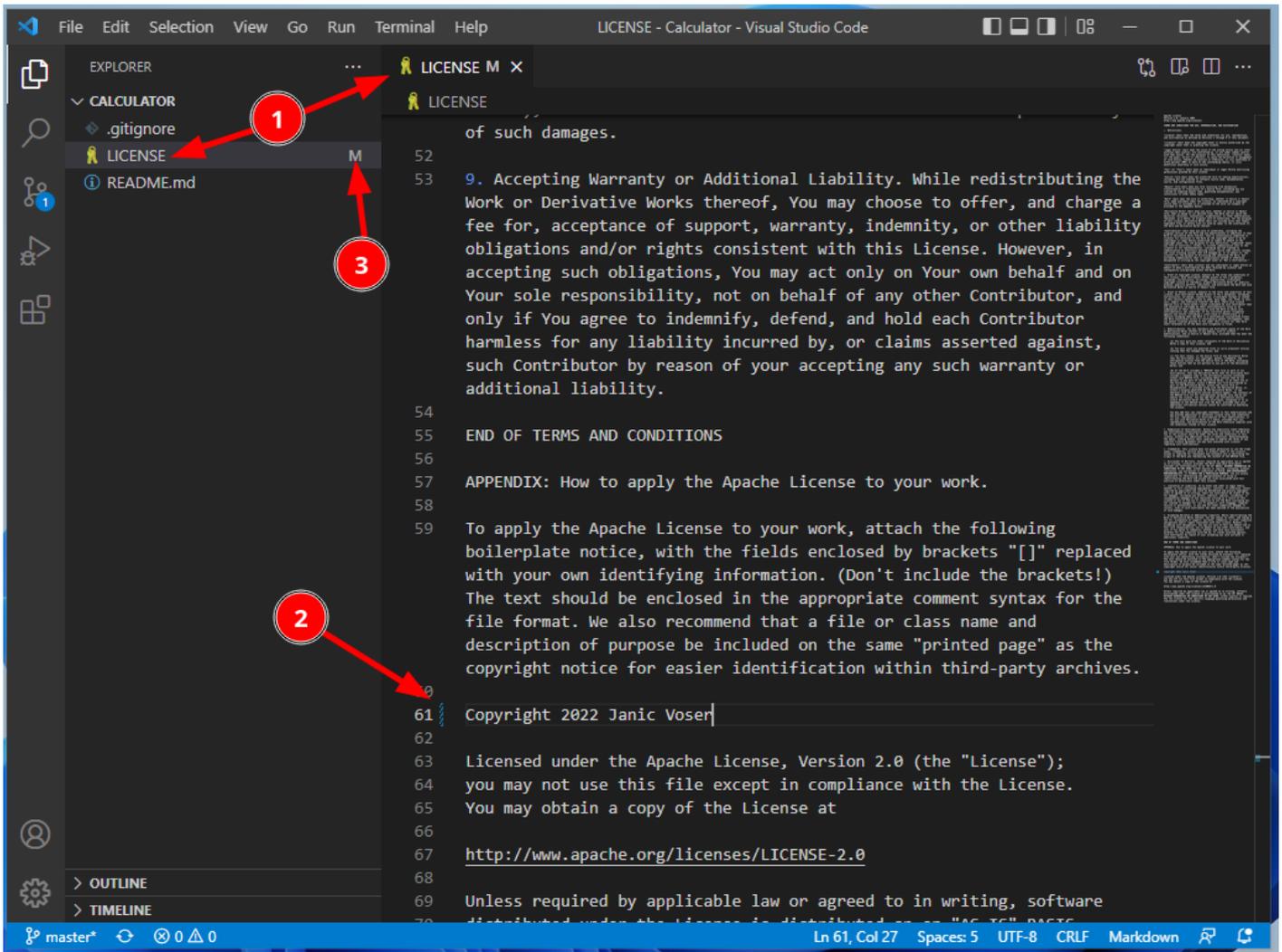
In Visual Studio Code you need to trust the author of this repository.



Afterwards you can see the Project folder on the left side. You can press `<ctrl>+<b>` to toggle this explorer view.

## Edit a file

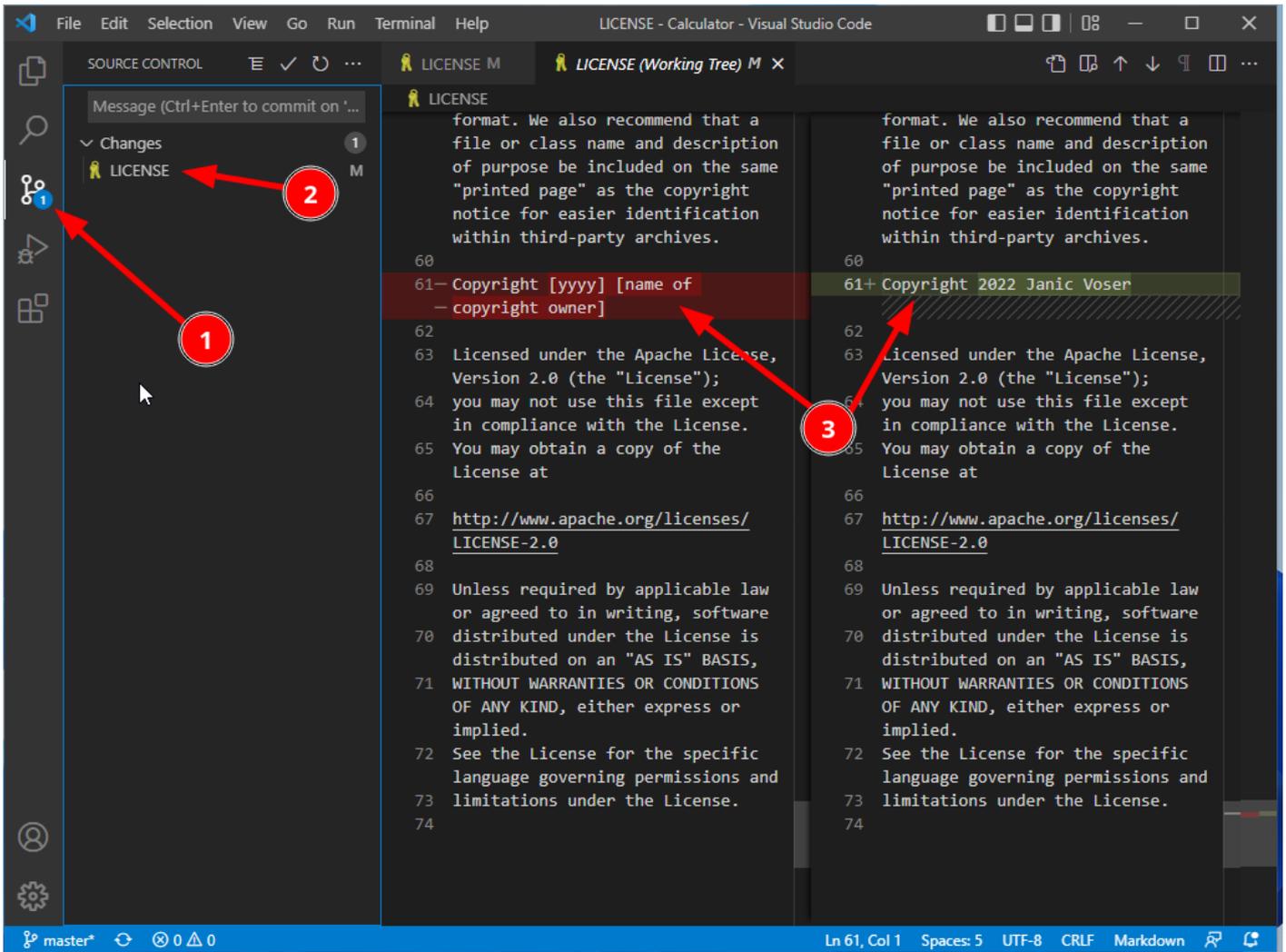
If you now edit a file you can see which lines you have modified.



1. Click on the file you want to edit.
2. Edit the line, you will see a blue mark in the beginning of the line, if the line was modified.
3. If you save the file you will see an M on the end of the file, this is a visual indicator that the file got edited.

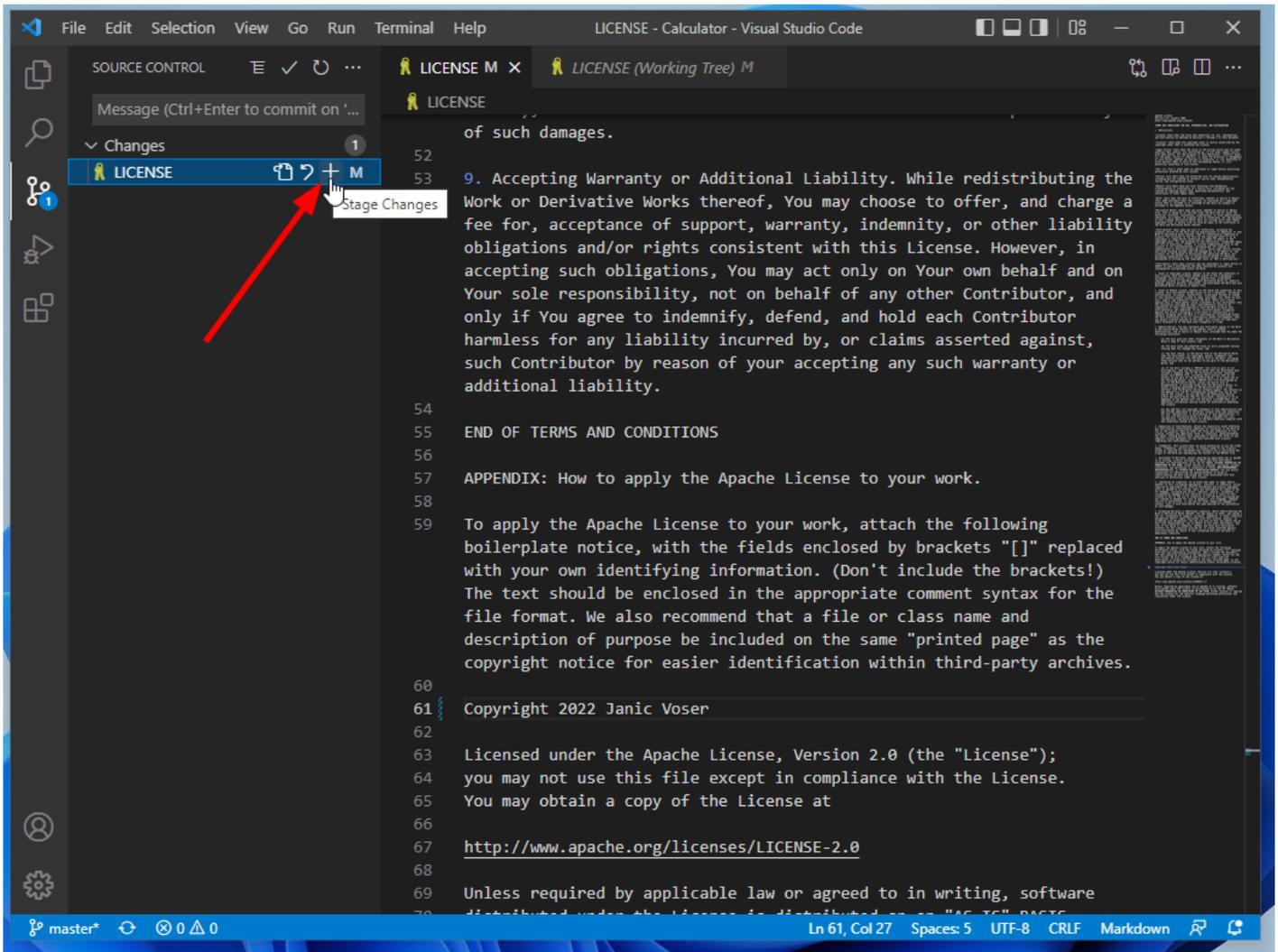
# GIT Gui VScode

## Changes



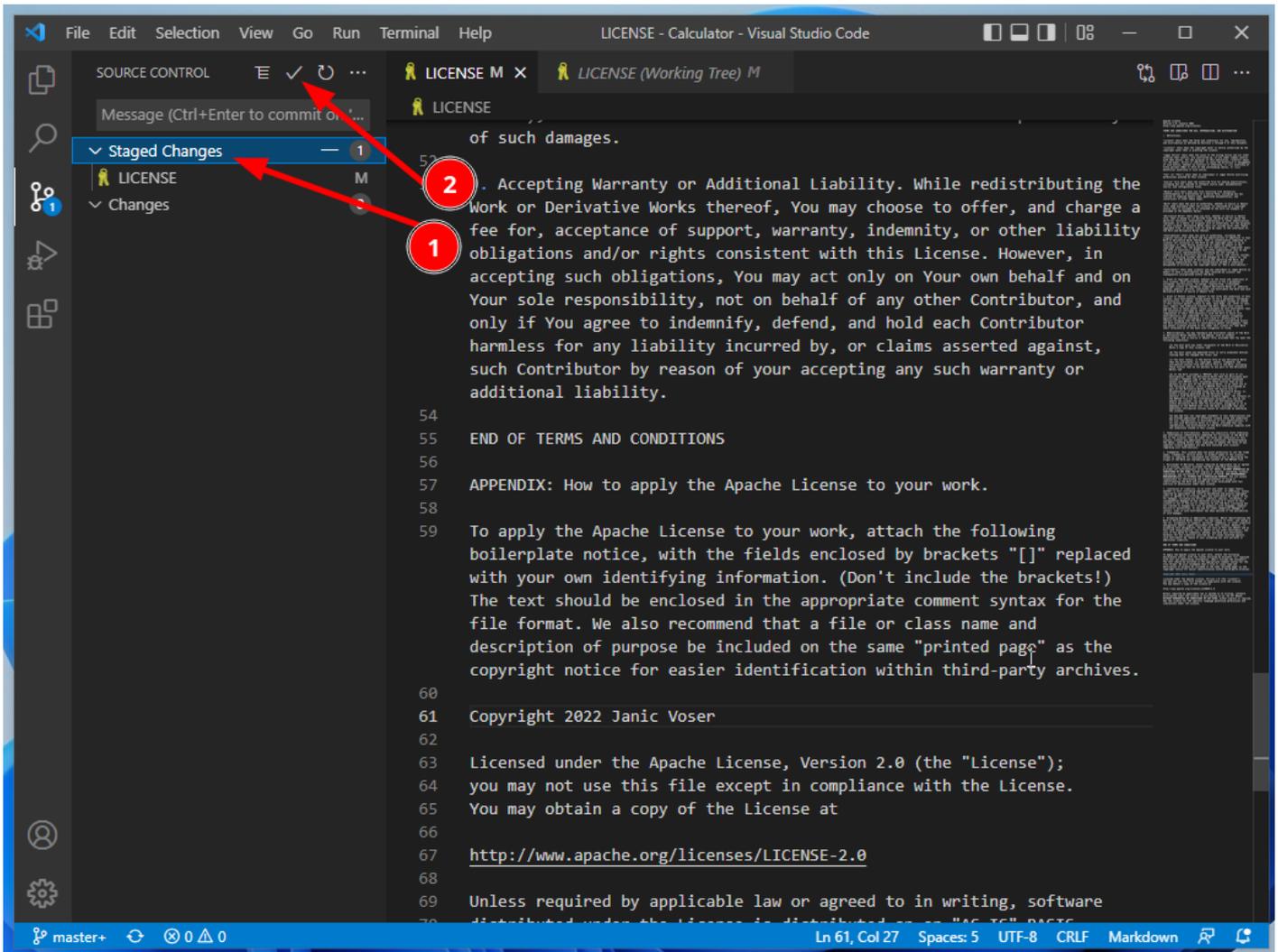
1. Navigate to the git section of VS-Code
2. Click on the modified file
3. Here you can see the changes

## Add files to commit

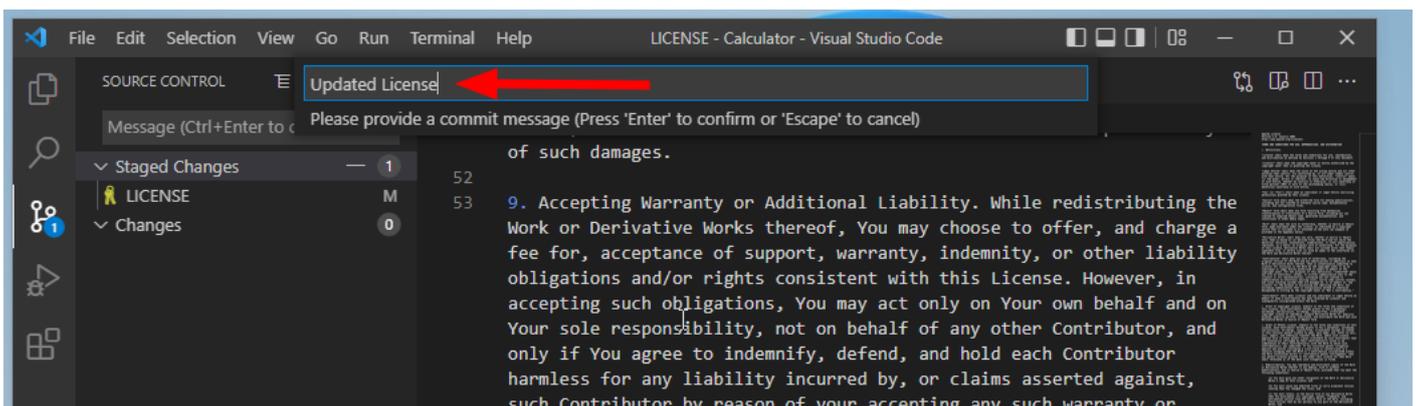


Click on the plus sign of the modified file.

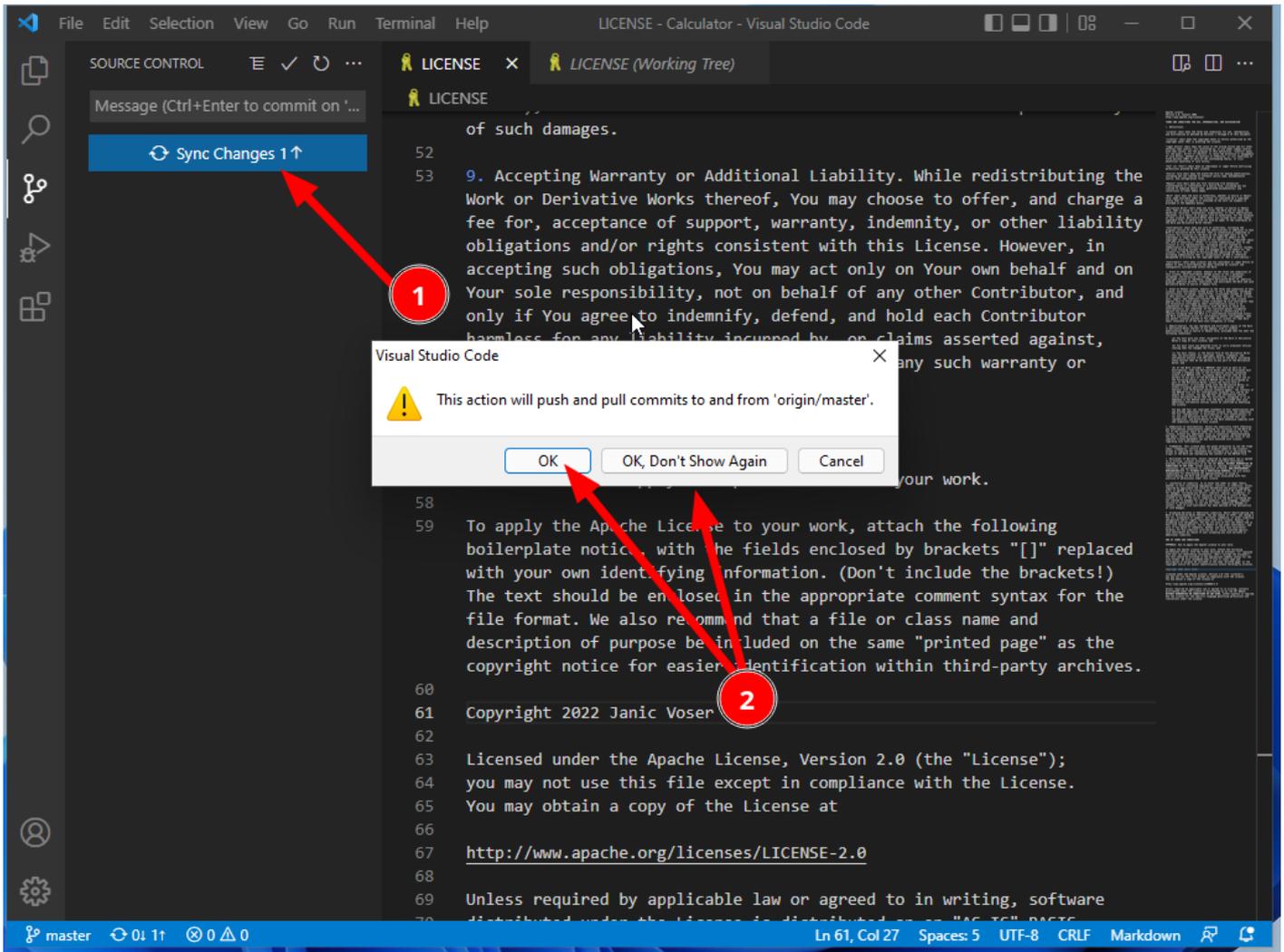
## Commit



1. Every file in the Staged changes gets committed with the same commit comment.
2. Click on Commit to add the changes

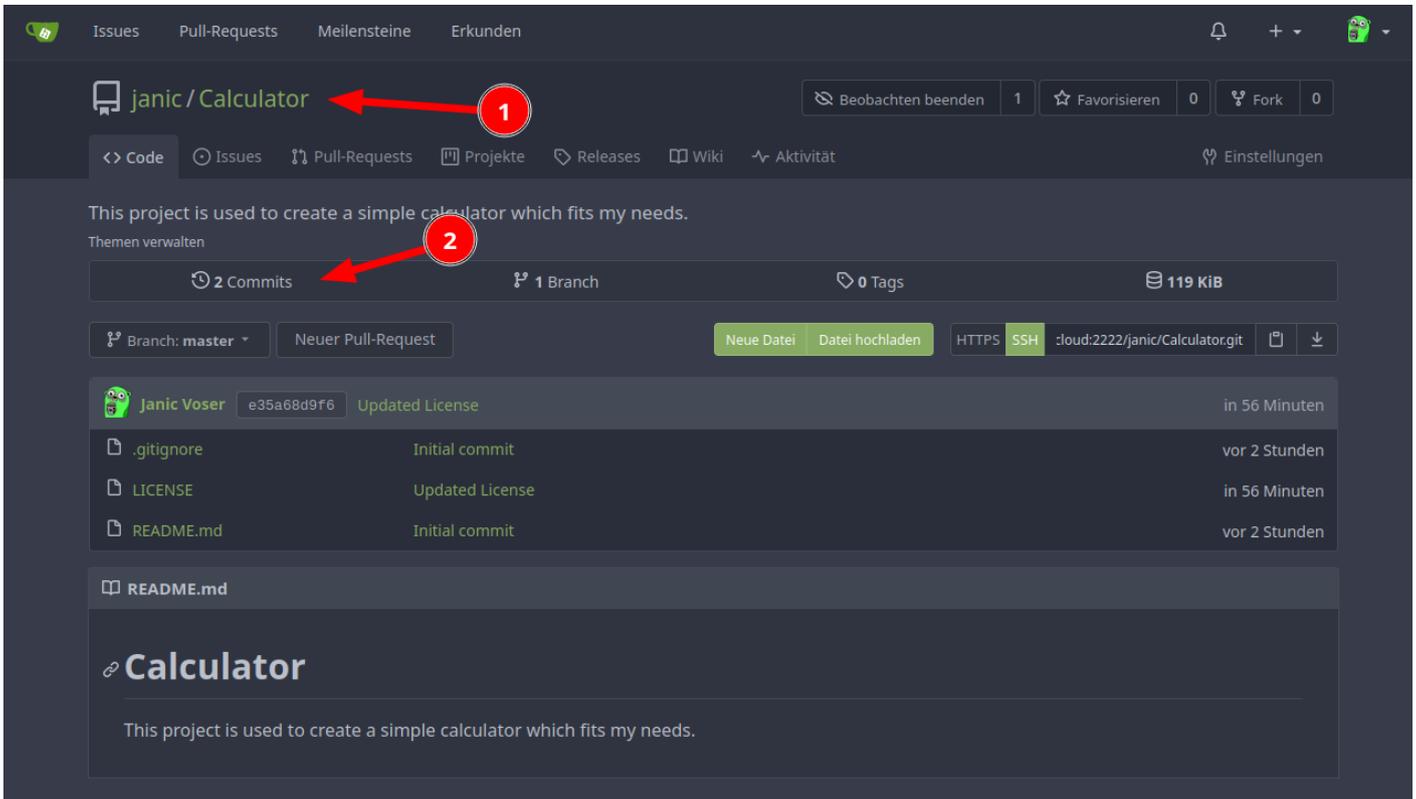


Enter a commit message which describes the changes

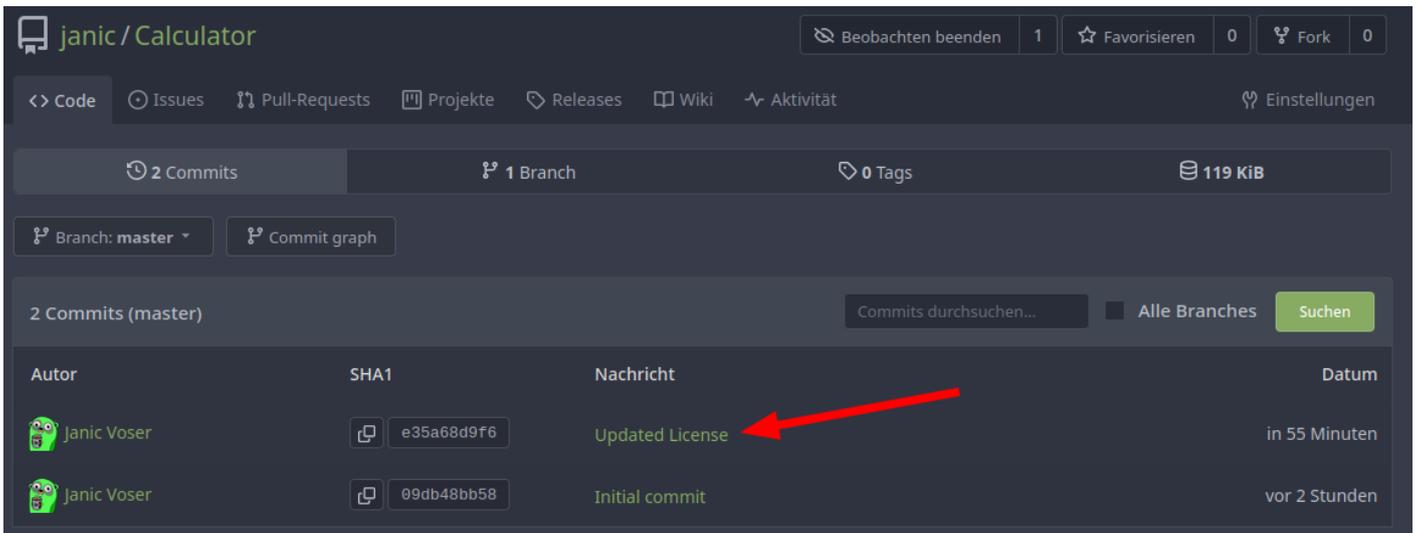


1. Click on Sync Changes to push the changes to your remote git repository (gitea.voser.cloud or github.com or ...)
2. Click ok to confirm the push.

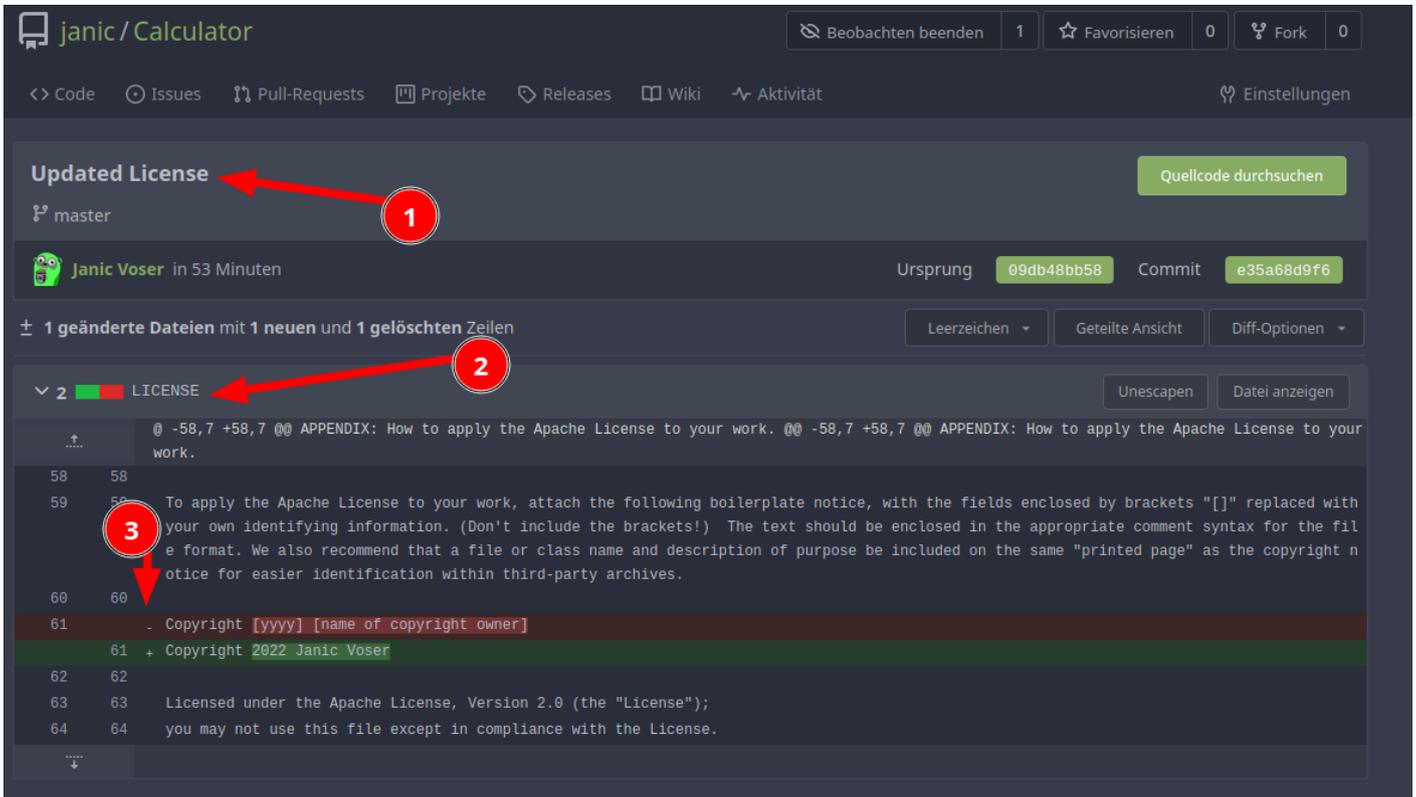
## Counter Check on website



1. Go to your Repository
2. Click on Commits



Now you can click on the commit



1. Here you see the commit name
2. Here are all edited files listed
3. Here you can see the changed lines a plus(green) or minus(red) in the beginning of the line. Plus means this line was added, minus means this line was removed