QMS formation

Mathieu Sambuq Guillaume Serriere

Kratos Communications SAS

11 February 2020

Table of Contents

1. QMS structure
2. Quality improvement management
3. Markdown
4. Git
5. Gitlab

Table of Contents

1. QMS structure
2. Quality improvement management
3. Markdown
4. Git
5. Gitlab

General Idea of new QMS

Seminar during Q4 2019 point the needs to:

- Improve the processes
- Improve the structure of the quality system
- Define a way to make collaborative and continuous improvement on processes

General Idea of new QMS

Seminar during Q4 2019 point the needs to:

- Improve the processes
- Improve the structure of the quality system
- Define a way to make collaborative and continuous improvement on processes

New version of QMS based on new architecture:

- Website to host QMS
- Git + Gitlab
- Markdown + Pandoc

QMS site

- URL http://192.168.48.22:8082/repository/qa/QMS.html
- All existing process are hosted in html format and keep to last version
- Up to date template are hosted

QMS site

- URL http://192.168.48.22:8082/repository/qa/QMS.html
- All existing process are hosted in html format and keep to last version
- Up to date template are hosted

Futur plan:

- Proper url name for the site
- Pdf version
- Better information about versioning

Gitlab project

Gitlab project: https://gitlab-toulouse.kratos.us/qms/kratos-qms

Contains information on:

- Markdown version of process
- Template
- Tools to generate qms website

Table of Contents

1 QMS structure

2 Quality improvement management

3 Markdown

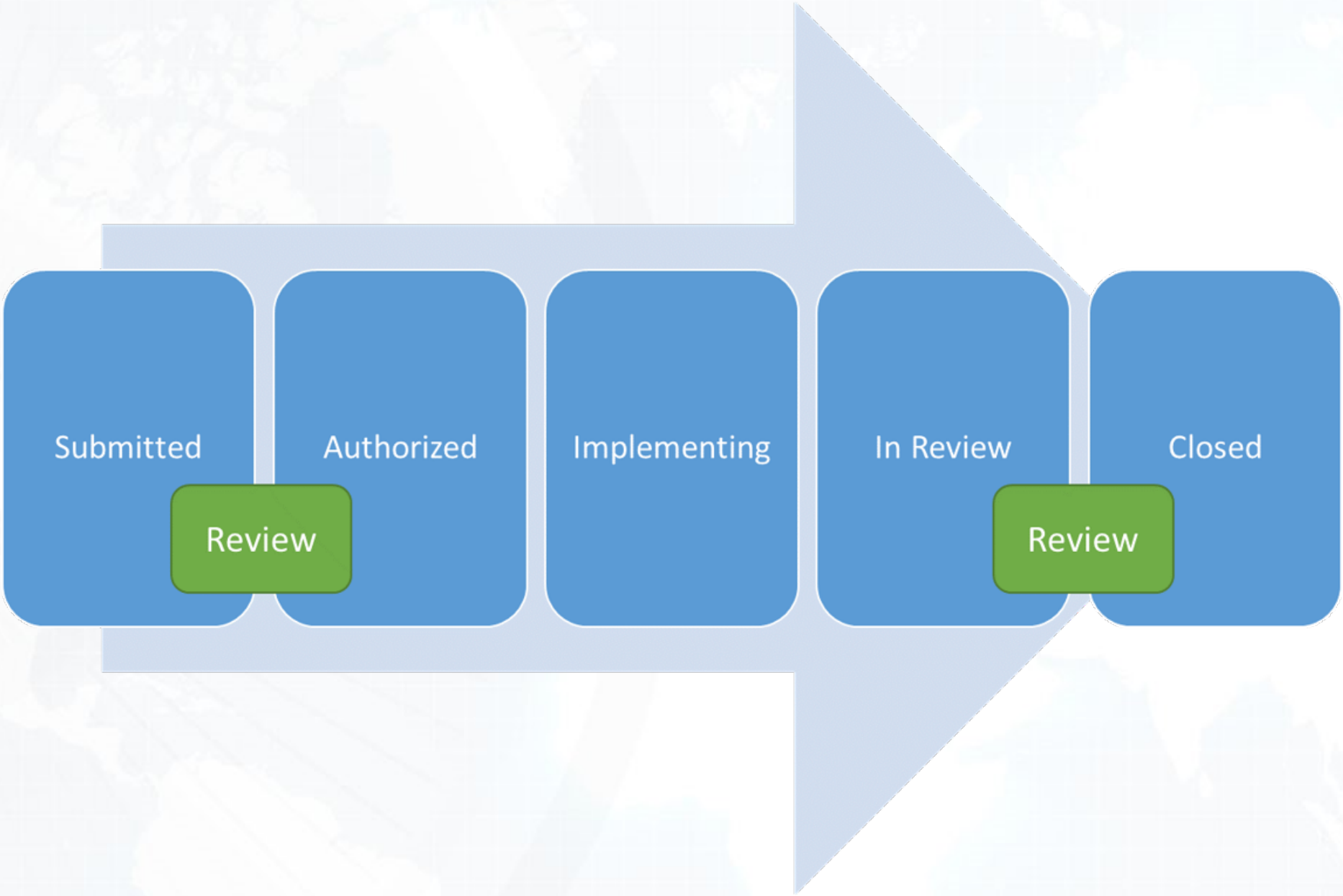
4 Git

5 Gitlab

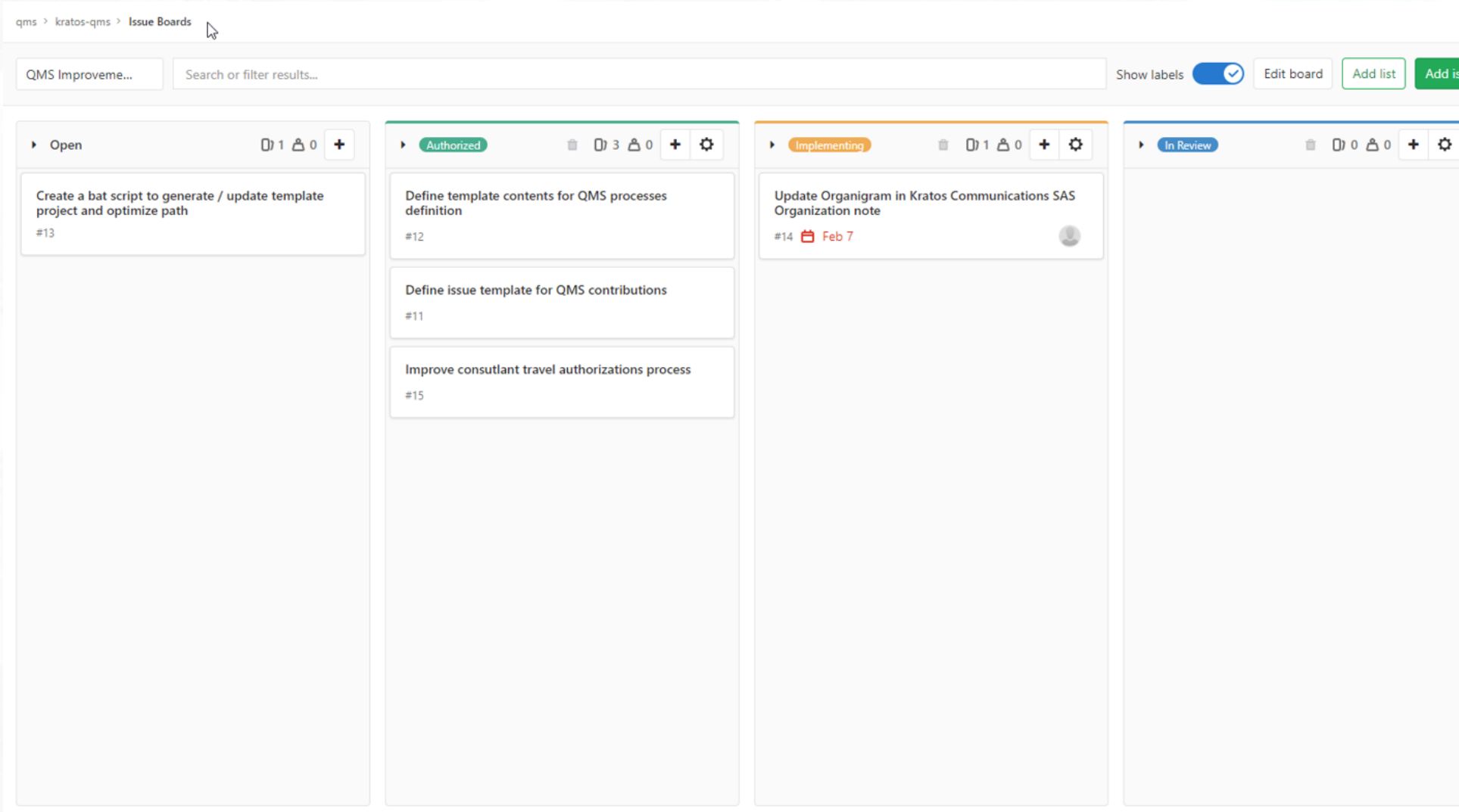
Why use Gitlab

- Collaboration at several levels:
  - Issue Submission
  - Issue Implementation (Update of a process or template)
  - Merge Request Review
- Version Control of processes and templates
- Flexible workflow using Issues
- Processes written in markdown language: managed like source code

Gitlab issue workflow



Gitlab issue workflow



Gitlab issue workflow

- Everyone can:
  - Create a new Issue.
  - Comment on other Issues. Like, Dislike.
  - Review, comment and approve merge requests.
- Volunteers for Issue implementation can:
  - Move an authorized issue to Implementing Status.
  - Move an Implemented issue to in Review Status.
- Nacera, Bruno and Mathieu can:
  - Authorize or close new Issues.
  - Close reviewed Issues.

Merge request

Merge request is an essential part of the process. It helps to avoid mistake and to improve quality of contribution.

Merge request

Merge request is an essential part of the process. It helps to avoid mistake and to improve quality of contribution.

But it can be source of frustration for all actors and every one needs to be willing to do is share of effort.

Merge request - Reviewer

A reviewer must follow these rules:

- Be benevolent
- Do not expect people to write like you
- Accept that the originator might not apply some of your suggestion for good reason
- Review the subsequent modification if not trivial

Merge request - Originator

The originator must follow these rules:

- Be benevolent
- Assume good faith
- Accept that you can have made a mistake
- Apply modification that are improving the quality of your contribution
- Do not accept proposition that doesn't improve the quality

Table of Contents

1 QMS structure

2 Quality improvement management

3 Markdown

4 Git

5 Gitlab

What is markdown

Definition

Markdown is a lightweight markup language that aims to provide basic text formatting in plain text format.

What is markdown

Definition

Markdown is a lightweight markup language that aims to provide basic text formatting in plain text format.

It is used as an intermediate language that is then translated in other format.

Example

```
# Section title
Text

## Sub-section title
Text
```

1 Section title

Text

1.1 Sub-section title

Text

Example

```
First␣paragraph.

Second␣paragraph.
Not␣a␣line␣break.␣␣
Line␣break.

First␣line.<br>Second␣line.


Third␣paragraph.
```

First paragraph.

Second paragraph. Not a line break.
Line break.

First line.
Second line.

Third paragraph.

Note: <br> is a tag from HTLM language and is interpreted by markdown as a line break.

Example: Text Emphasis

```
*This text will be italic*

_This will also be italic_

**This text will be bold**

__This will also be bold__

_You **can** combine them_
```

This text will be italic

This will also be italic

This text will be bold

This will also be bold

*You **can** combine them*

Example: Bullet list

```
+ first item
+ second item

-- Or --

- first item
- second item

-- Or --

* first item
* second item
```

- first item
- second item

Example: Enumerate list

```
1. first item
2. second item
5. third item
5. fourth item

a. first item
b. second item
b. third item
```

1. first item
2. second item
3. third item
4. fourth item

a. first item
b. second item
c. third item

Value on enumeration doesn't matter.

Example: Nested list

```
1. first item
    1. nested element
    2. nested element
2. second item
    + first point
    + second point
1. third point
    a. first letter
    a. second letter
        – first point
1. fourth point
```

1. first item
    1. nested element
    2. nested element
2. second item
    - first point
    - second point
3. third point
    a. first letter
    b. second letter
        - first point
4. fourth point

Example: Table

```
| Header 1 | Header 2 |
| --- | --- |
| Text | Longer text as you can see |
| Text | Text with <br>line return |
```

Header 1	Header 2
Text	Longer text as you can see
Text	Text with line return

Note: <br> is a tag from HTLM language and is interpreted by markdown as a line break.

Example: Image

```
![Kratos Logo](Images/Kratos_Cover.png)
```

Kratos Logo

Link are can be with a relative path (Images/kratos.png) or absolute (C:\Users\guillaume.serriere\Documents\qms formation\kratos.png)

Example: Link

```
[Link 1](document.md)

[Link **with emphasis**](document.md)
```

Link 1

Link **with emphasis**

- Link are can be with a relative path (Images/kratos.png) or absolute (C:\Users\guillaume.serriere\Documents\qms formation\kratos.png).
- Link to *.md file will be replaced with link to *.html file in our system.

Example: Special Character

Some characters are part of markdown language and need to be escaped (put a \in front) in order to be usable normally.

- '>'
- '-', '+', '*' when use at beginning of line
- '\'
- '|'

Definition

Pandoc is an open-source document conversion software first released in 2006.
It allows to convert from its own markdown format to html format.

Definition

Pandoc is an open-source document conversion software first released in 2006.
It allows to convert from its own markdown format to html format.

The pandoc version can be extended and we added several extensions:

- Automatic link conversion from *.md to *.html
- Automatic toc and header numeration
- Header information (kratos logo and information) inclusion

Table of Contents

1 QMS structure

2 Quality improvement management

3 Markdown

4 Git

5 Gitlab

What is git ?

Definition

Git is a distributed Version Controlling System originally developed by Linus Torvalds first released in 2005.
It allows to track and to share the change of a set of file.

What is git ?

Definition

Git is a distributed Version Controlling System originally developed by Linus Torvalds first released in 2005.
It allows to track and to share the change of a set of file.

Note: Git is a lot more than this.

Terminology: Repository

Repository definition

A repository is the name of the structure that handle the code and track the change.

Terminology: Commit

Commit definition

Git is working by taking snapshot at some point in time of the state of every file, creating a new version of all the files each time. Each snapshot is recorded incrementally with the difference with the last recorded snapshot.
Such snapshot is called a commit.

Terminology: Commit

Commit definition

Git is working by taking snapshot at some point in time of the state of every file, creating a new version of all the files each time. Each snapshot is recorded incrementally with the difference with the last recorded snapshot.
Such snapshot is called a commit.

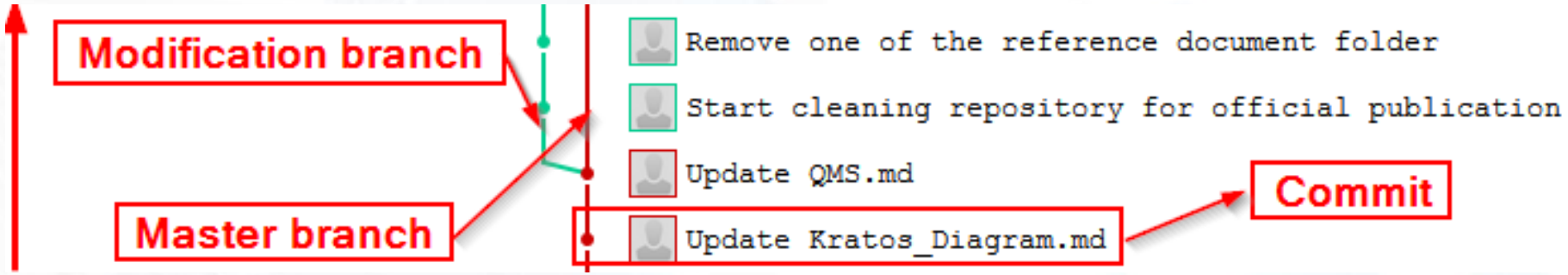
All commit are organized in a graph and every version recorded in git can be constructed by applying all commit in order.

Terminology: Branch

Branch definition

A branch is a succession of commit. It has a base, and a HEAD which is the most recent commit of said branch and commits between.

A branch can be created from any commit, and a repository can have multiple branches.



Master

Master is by convention the name given to the main branch of a repository.

Merge and Master

Master

Master is by convention the name given to the main branch of a repository.

Merge

A merge is the operation use to bring modification of a branch in another one.

Merge explication

A merge is done by:

1. taking the most recent common commit
2. computing the cumulative difference since this commit
3. applying it to the other branch

Merge explication

A merge is done by:

1. taking the most recent common commit
2. computing the cumulative difference since this commit
3. applying it to the other branch

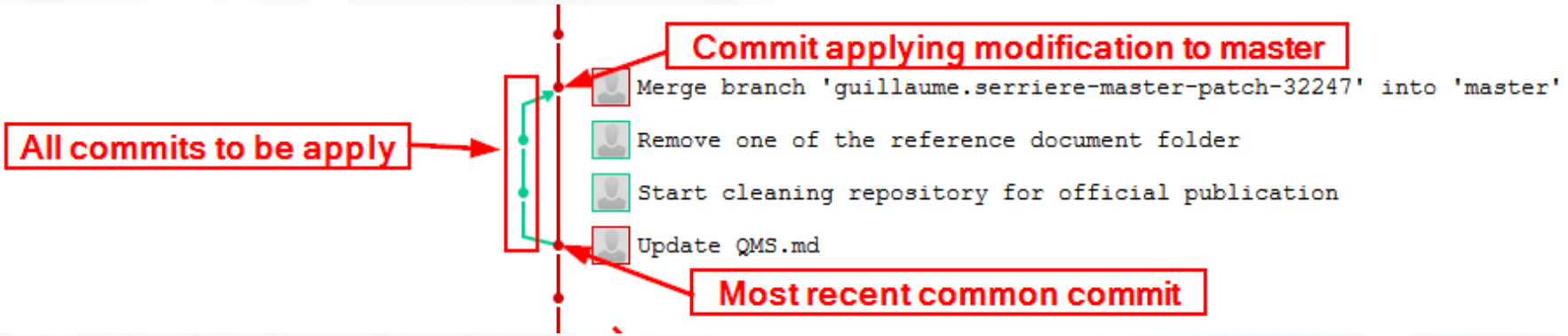


Table of Contents

1. QMS structure
2. Quality improvement management
3. Markdown
4. Git
5. Gitlab

What is Gitlab ?

Gitlab definition

Gitlab is a web server which provides a set of feature to git repositories.

Gitlab definition

Gitlab is a web server which provides a set of feature to git repositories.

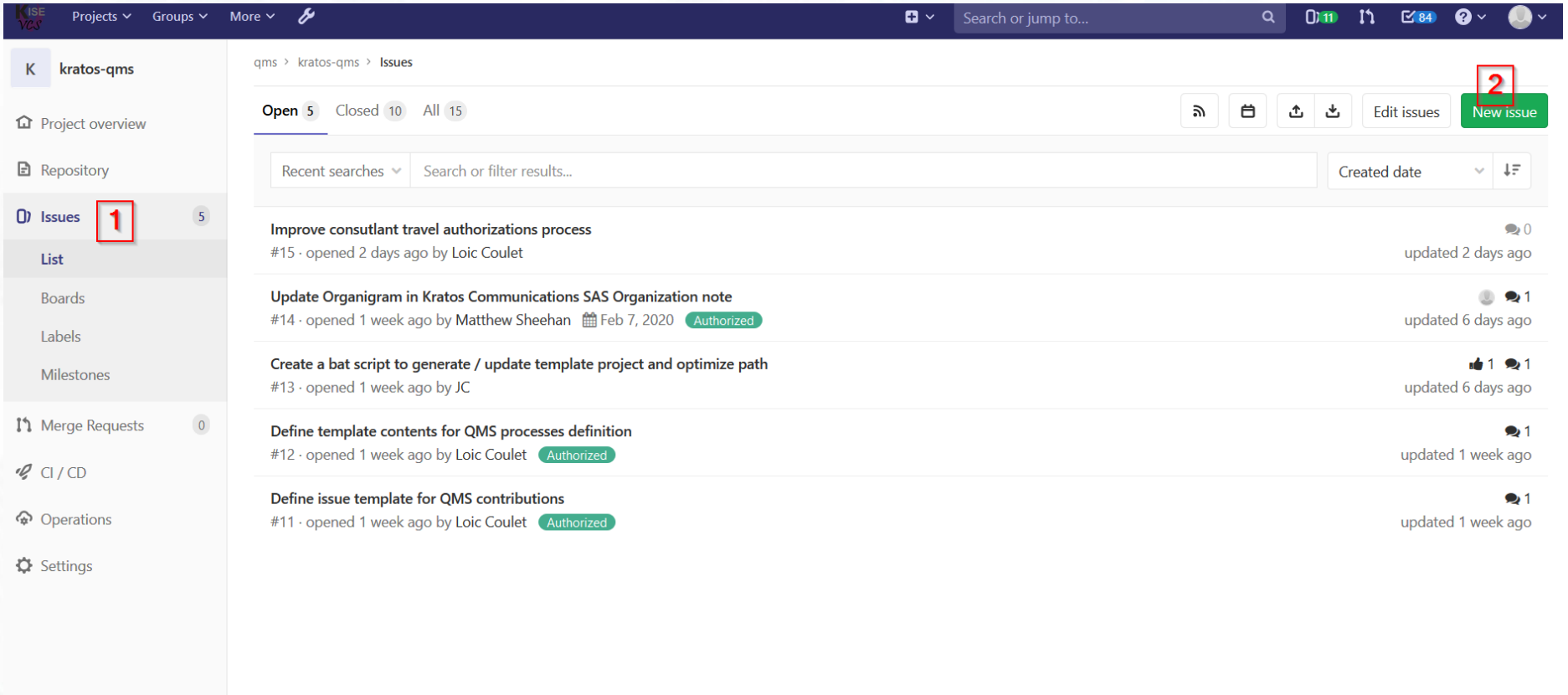
Set of features useful for QMS:

- Issue creation and commenting
- Branch merging tool
- Continuous integration

First step

The first step of the process is to create an issue describing what needs to be done.

Issue creation



Issue creation

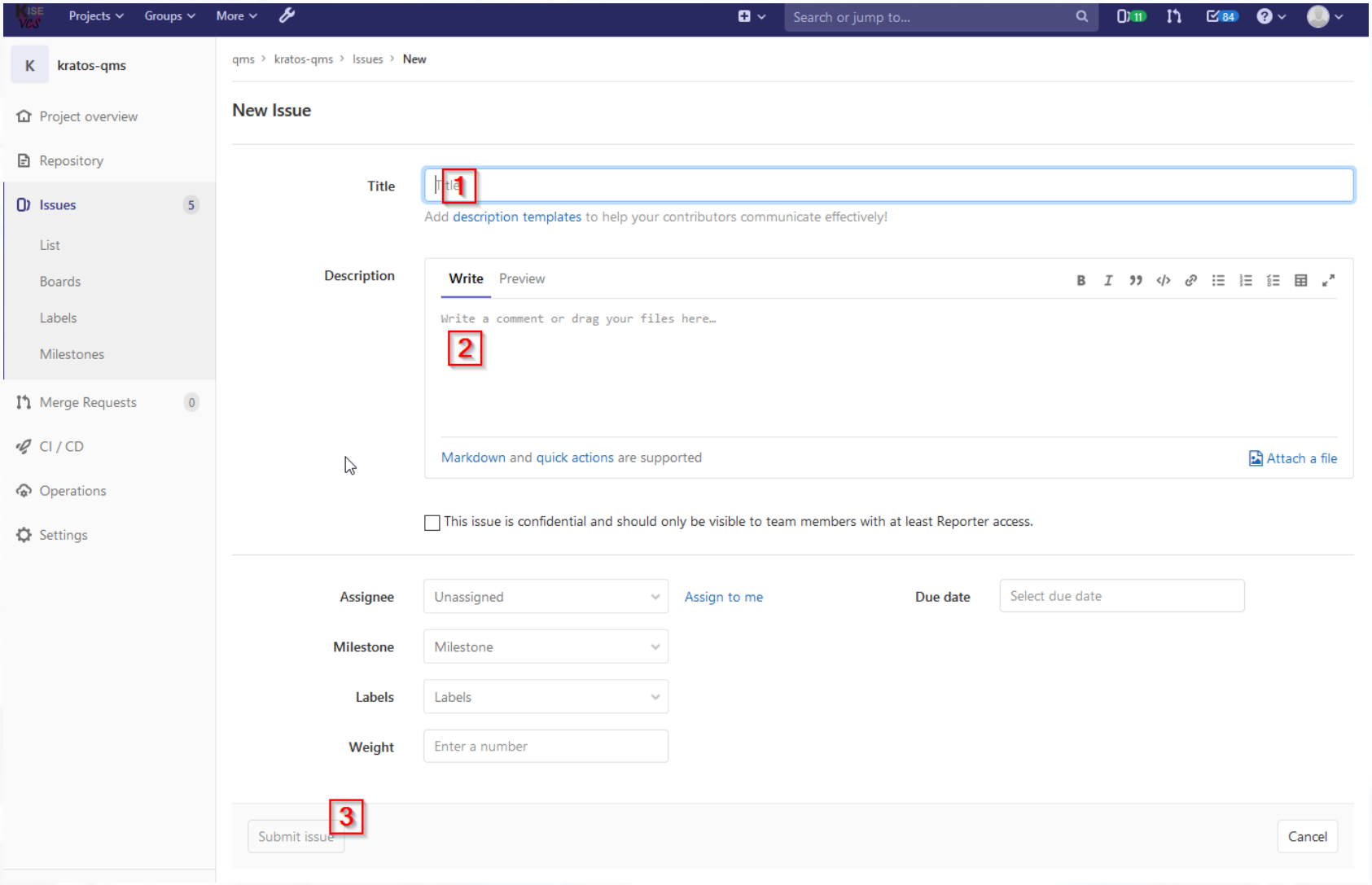
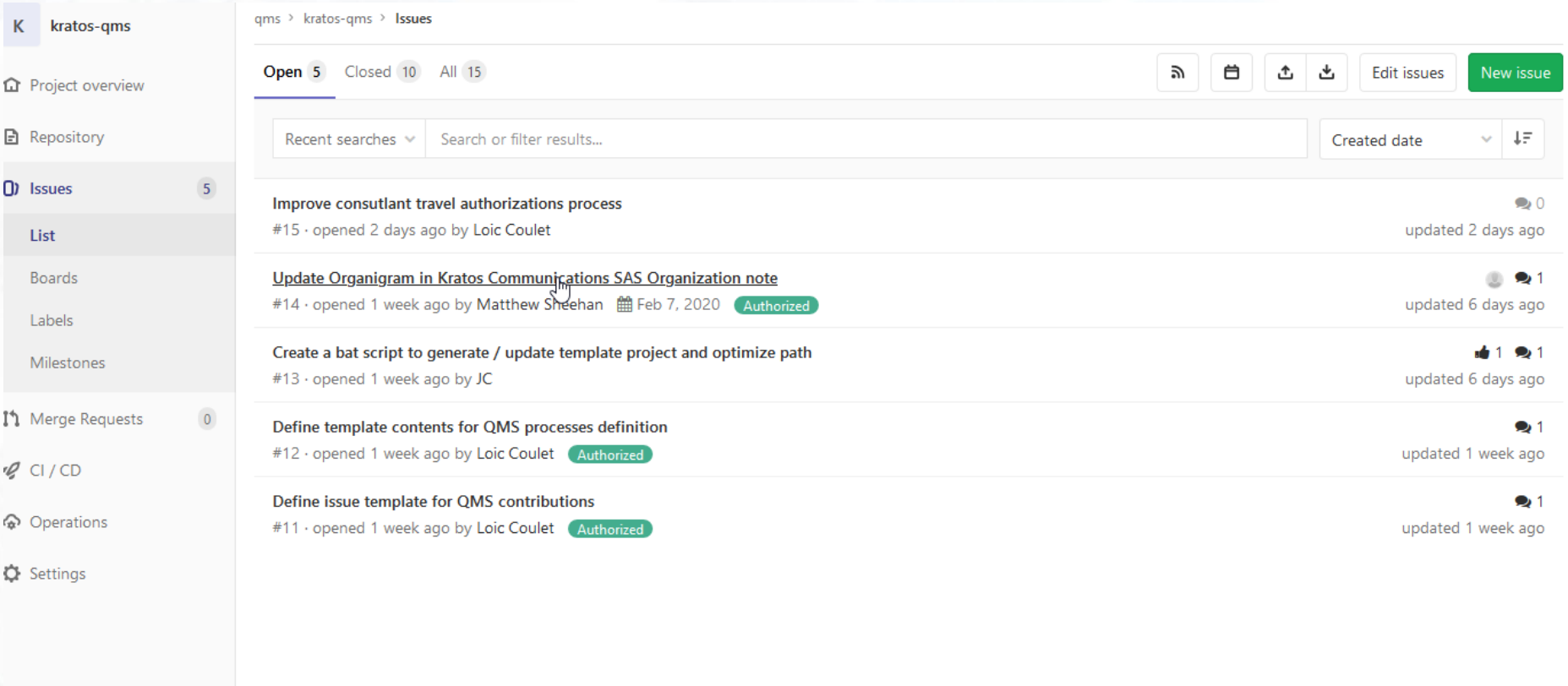
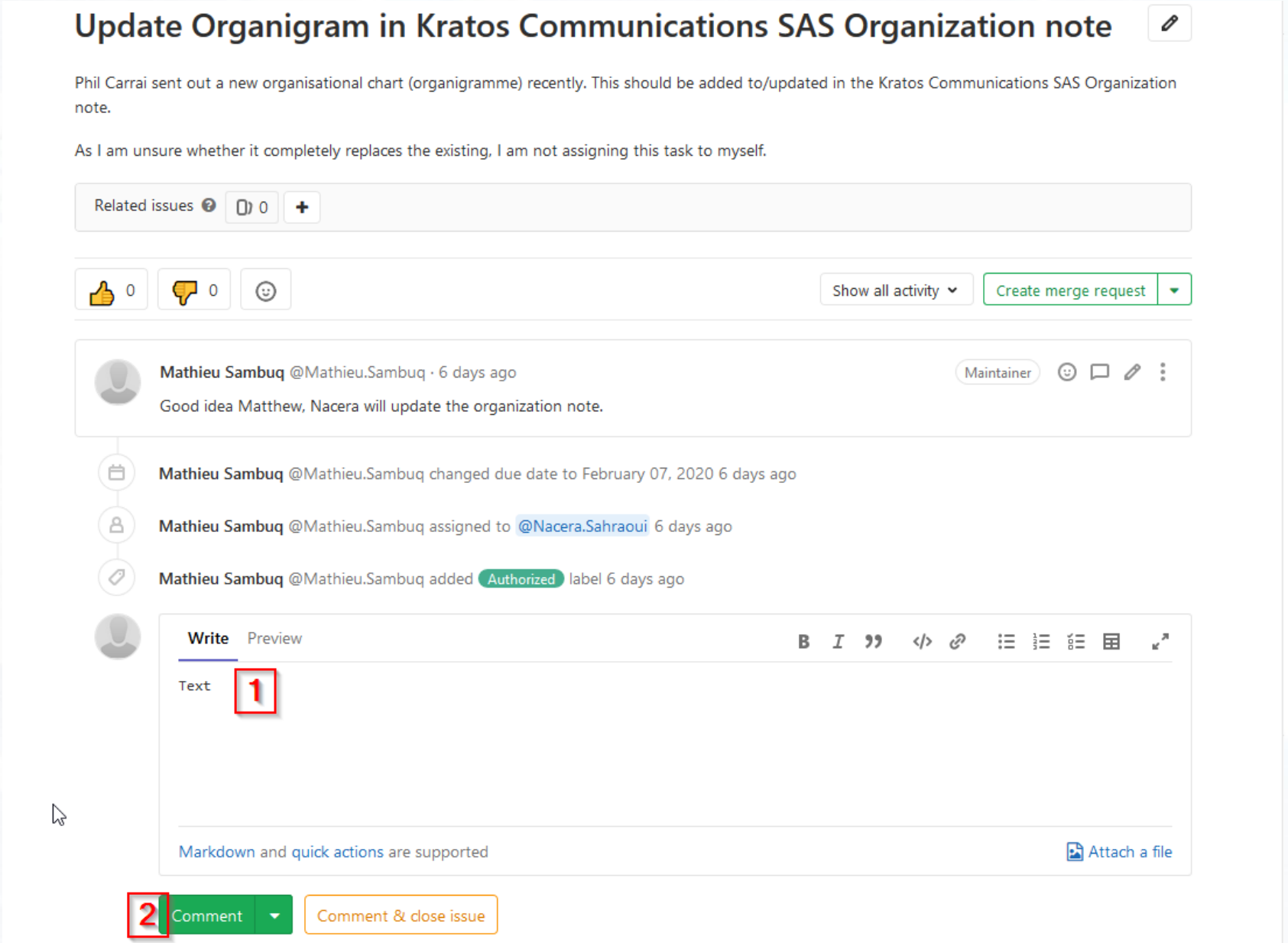


Figure: 1. Title, 2. Description, 3. Submit

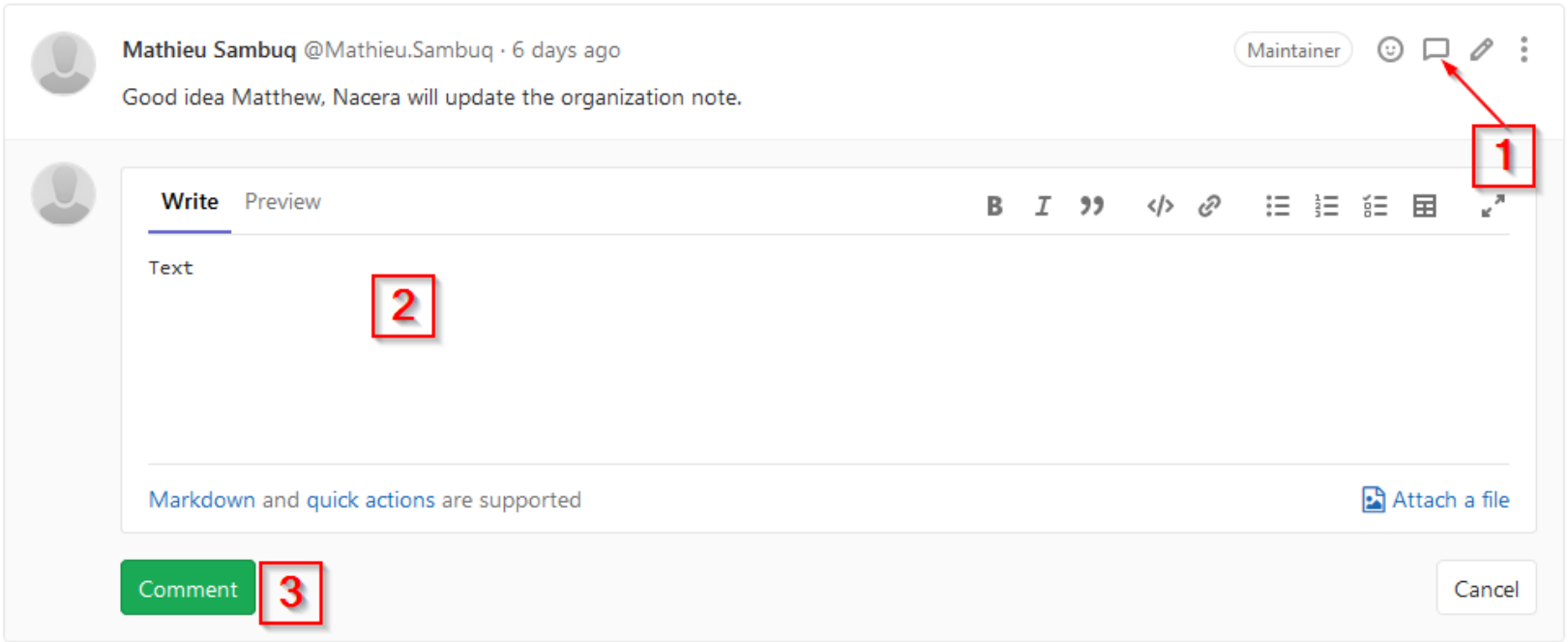
Commenting on an issue



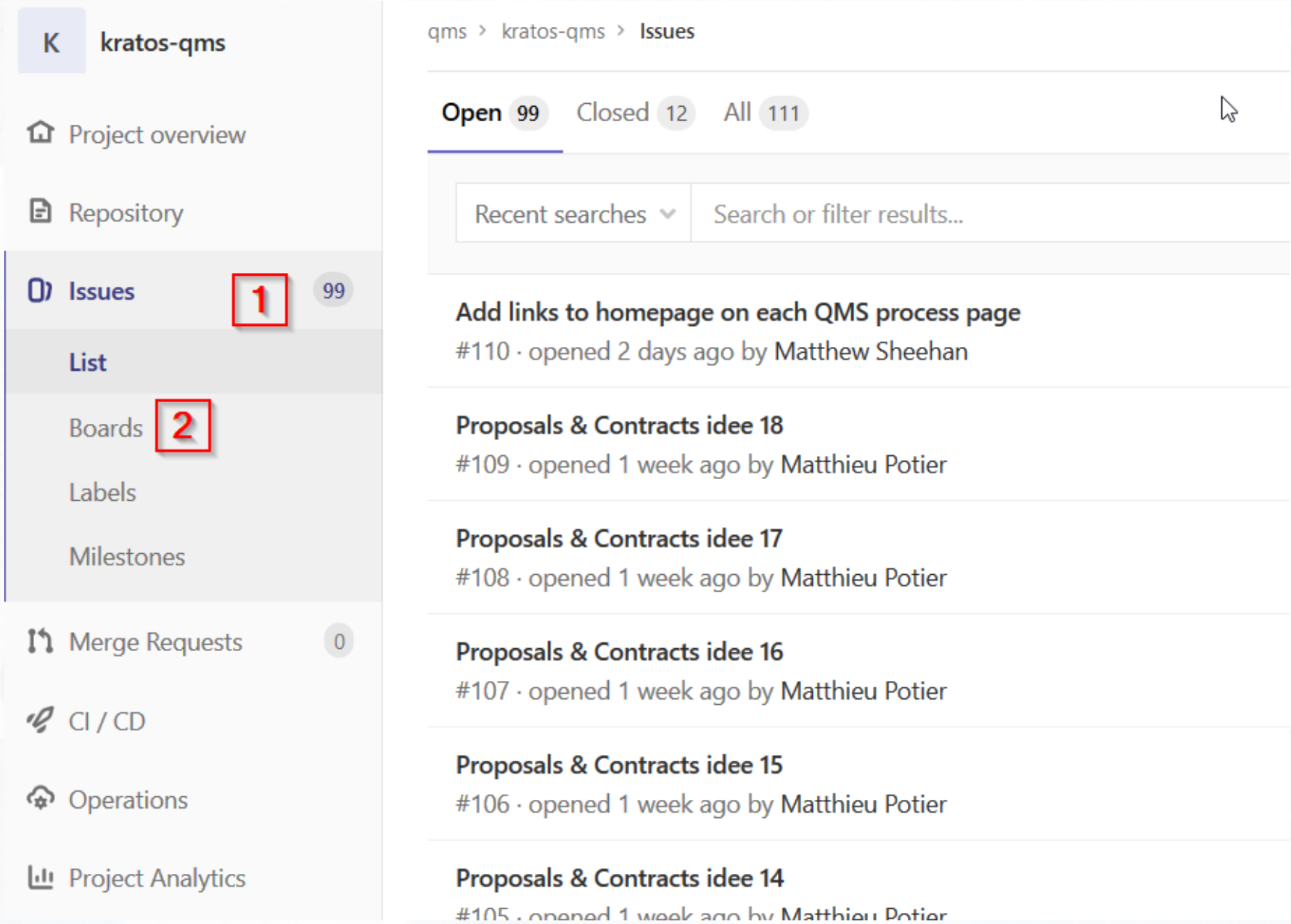
Commenting on an issue



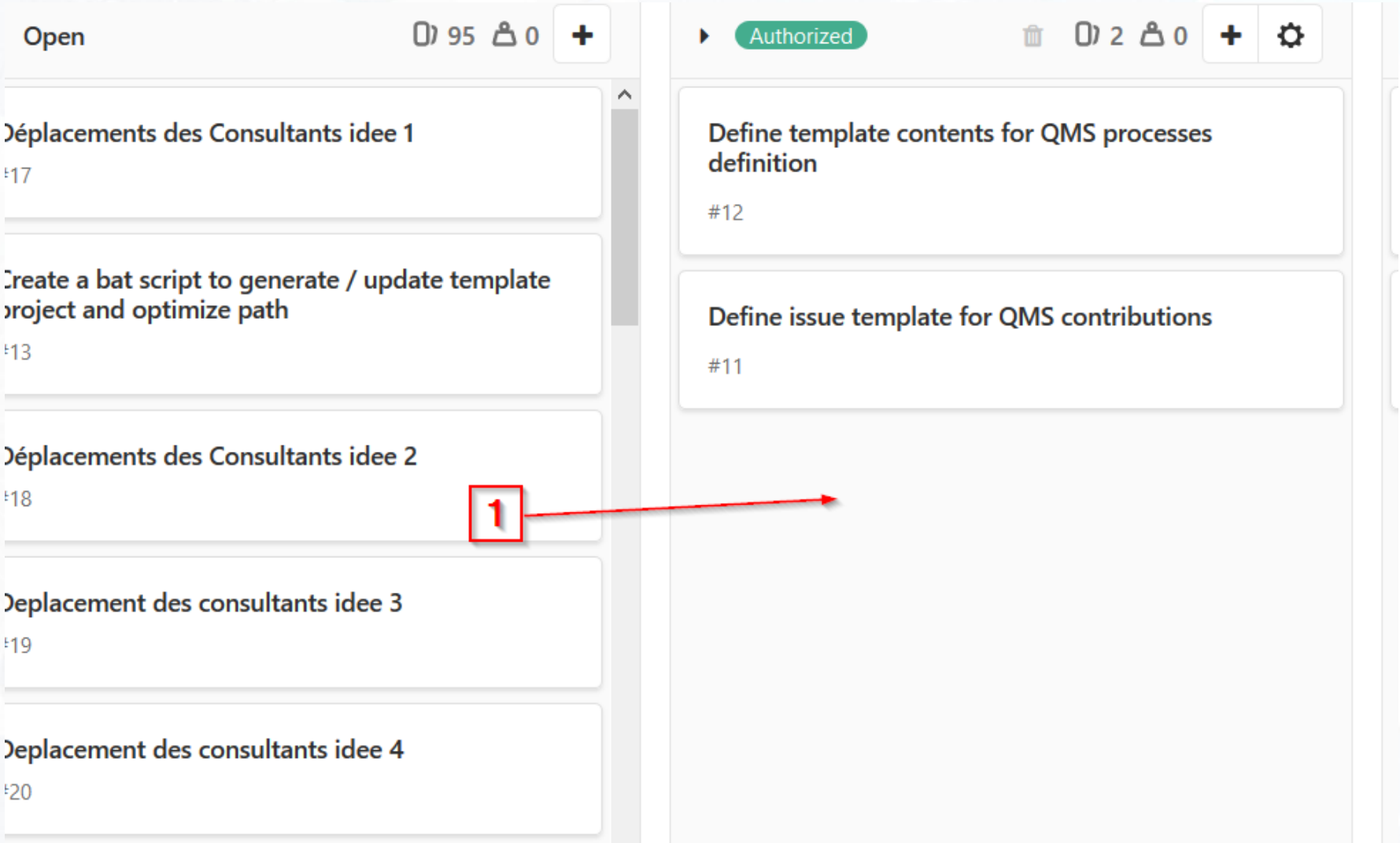
Commenting a comment



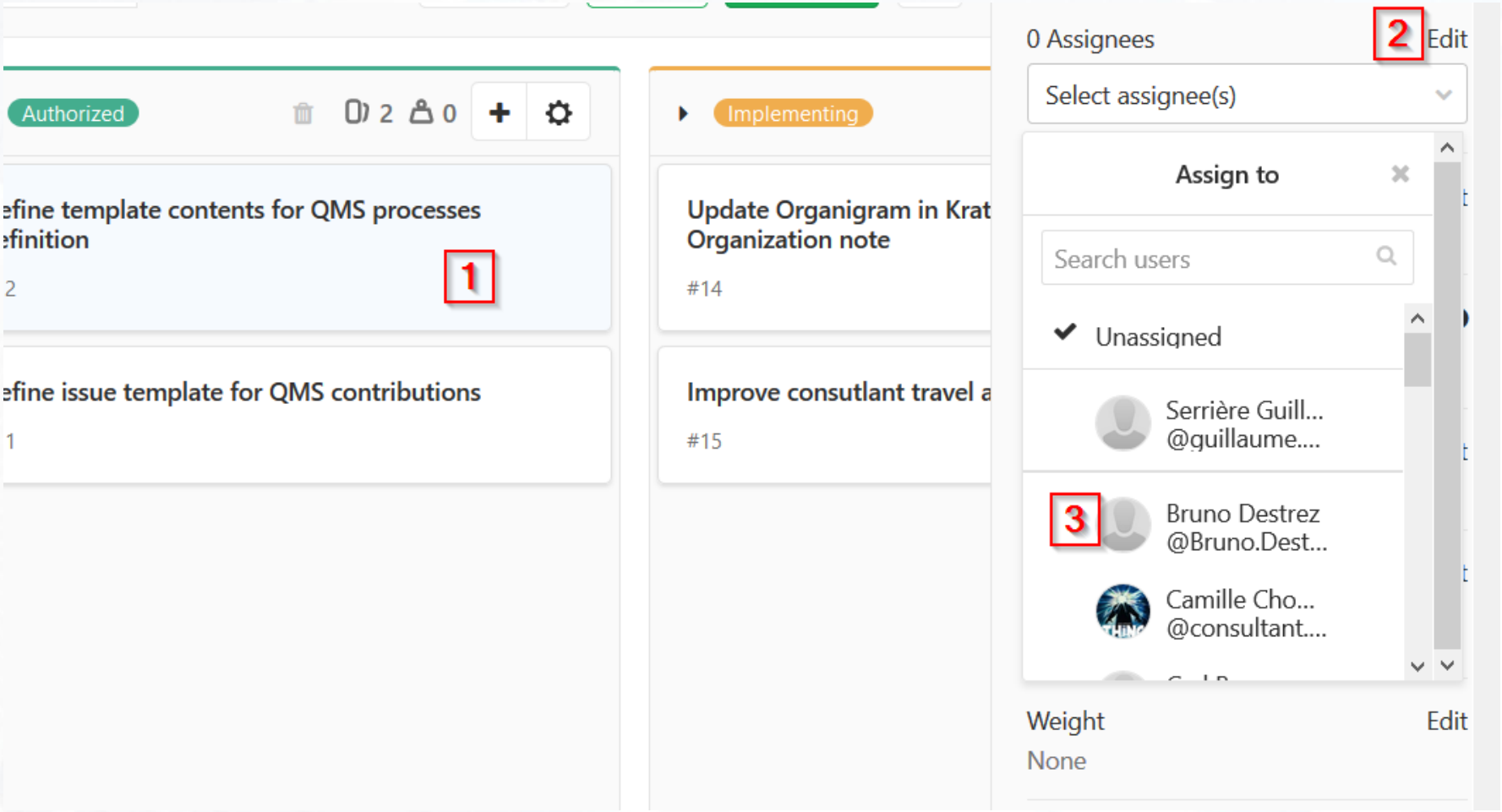
Issue manipulation



Issue manipulation



Issue manipulation



1. Click in the issue
2. Click on Edit in Assignees section
3. Select people

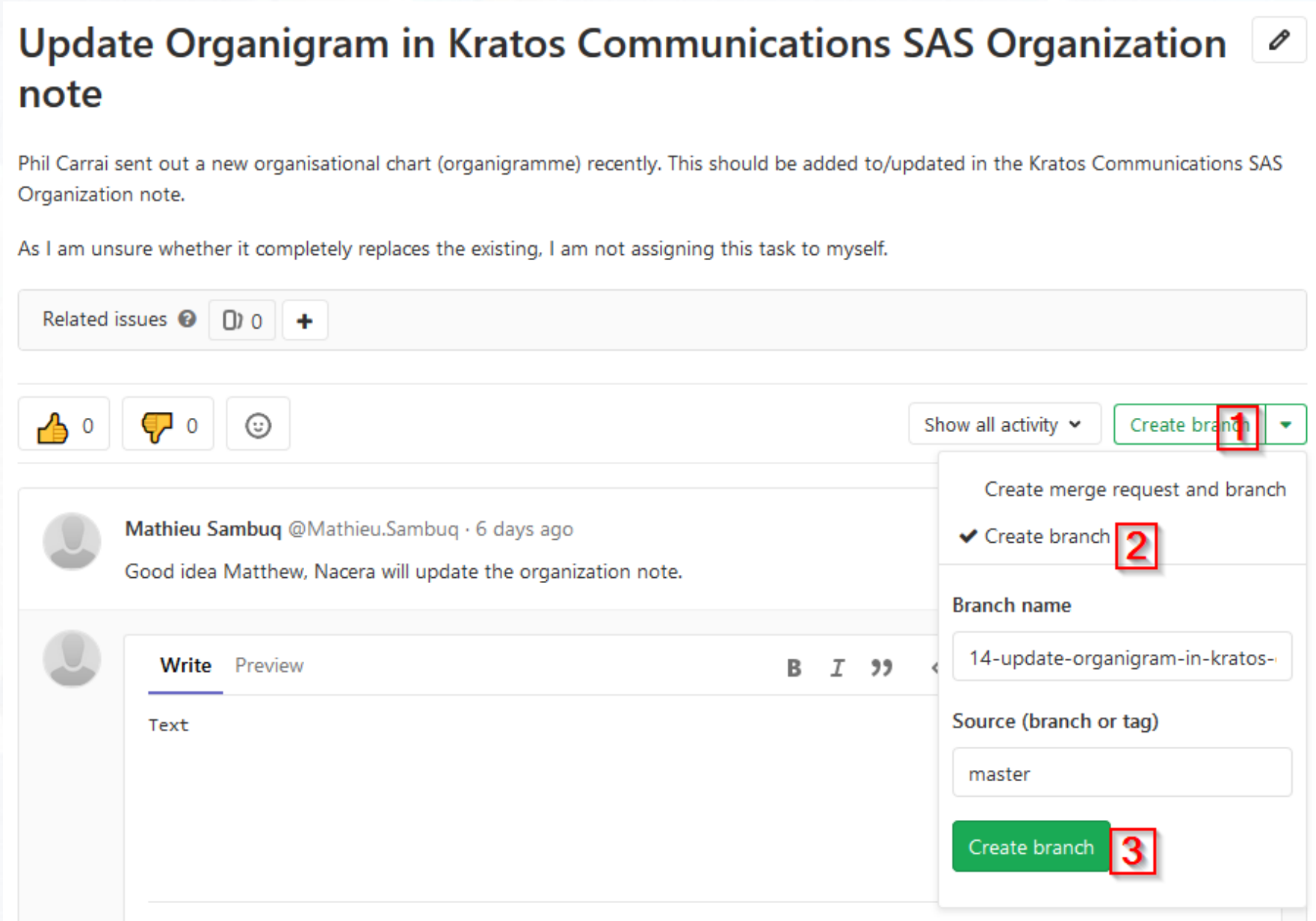
Second step

The second step is to edit what needs to be change.

For this there is two solution presented in:

1. Editing one file at time (easiest)
2. Using the gitlab Web IDE (hardest)

Branch creation

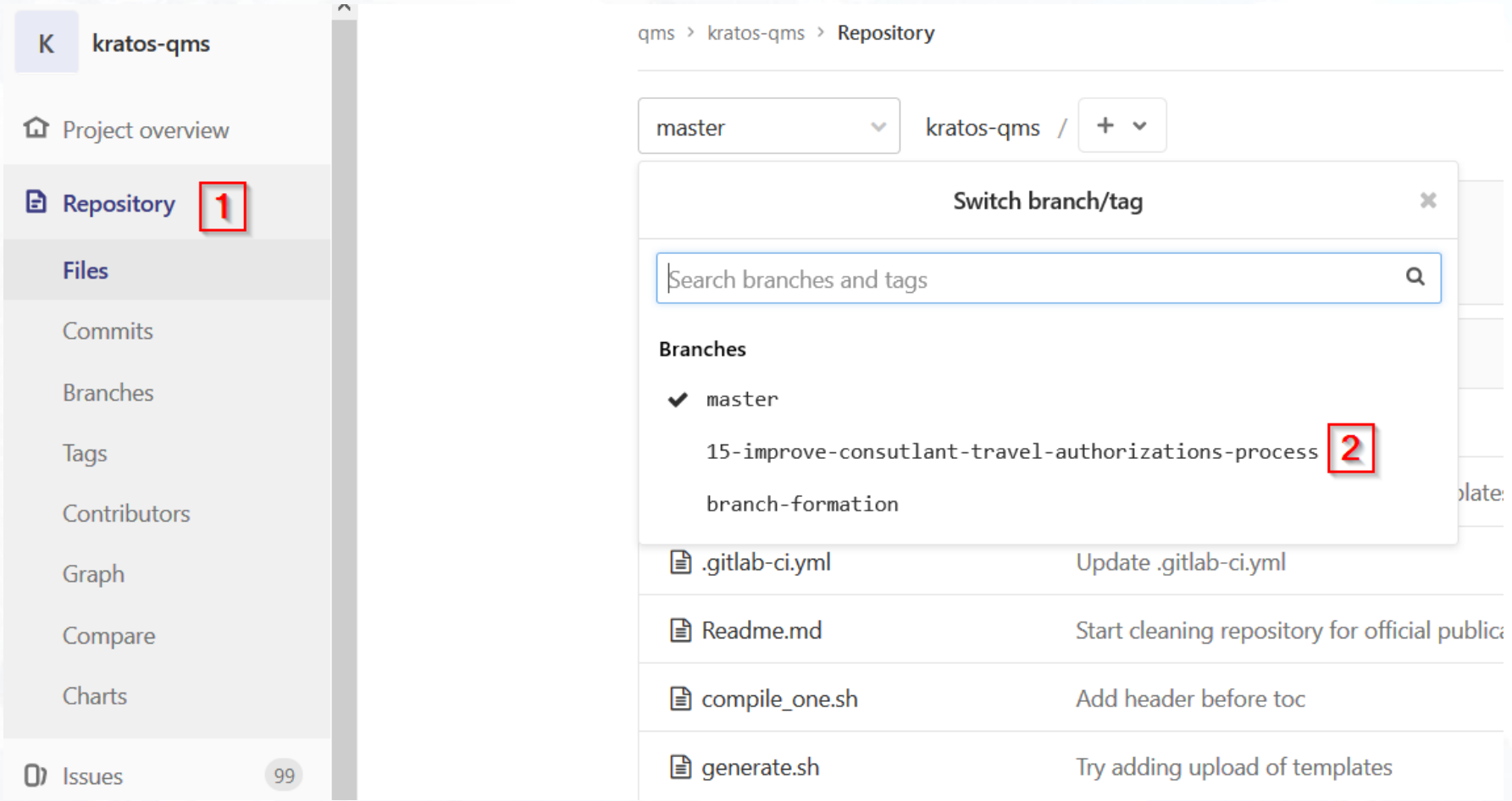


Editing one file

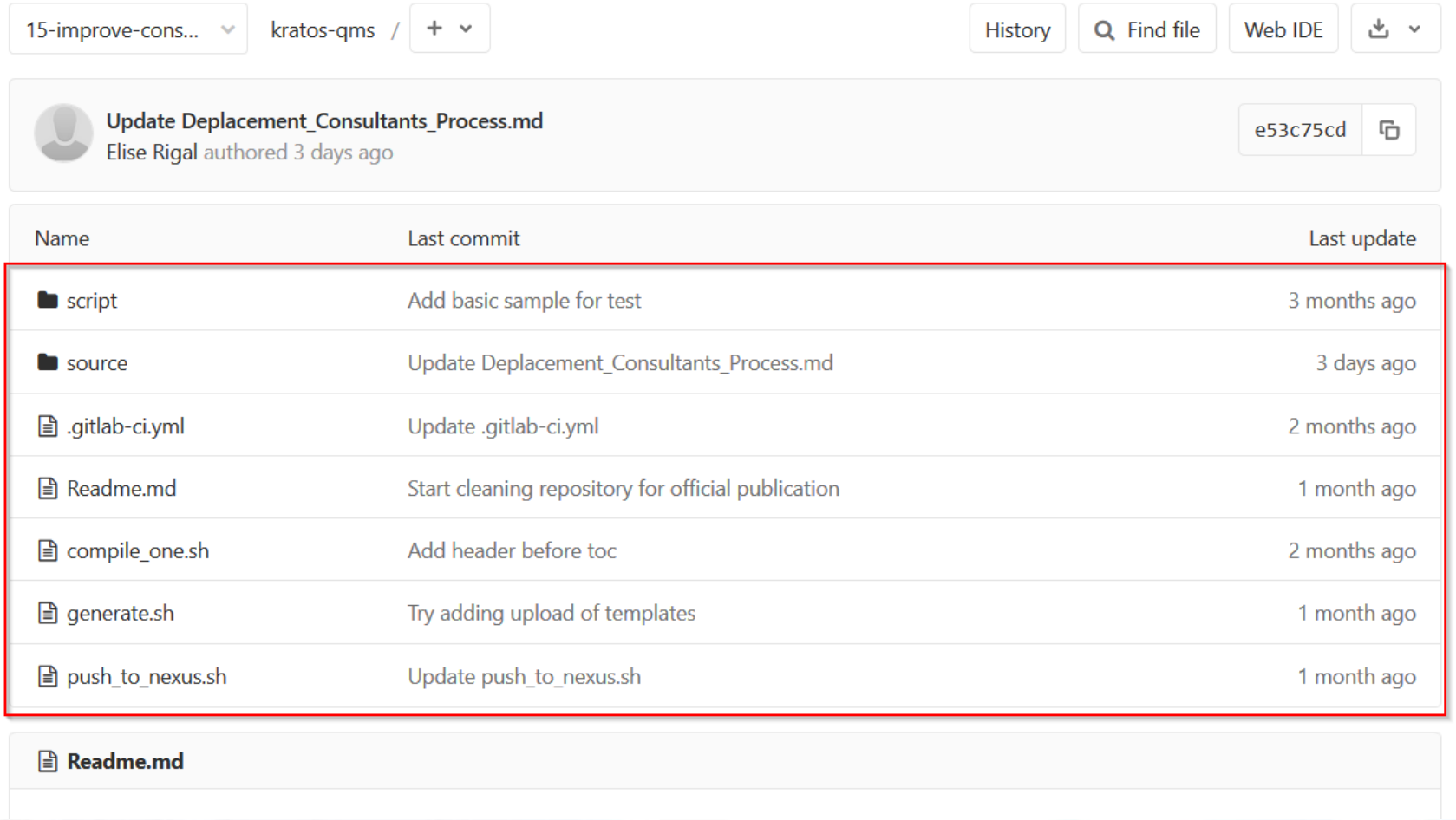
Gitlab let you the possibility to edit the file one at a time.

In this configuration, every time a file is saved, a commit is generated.

Editing one file - Branch creation



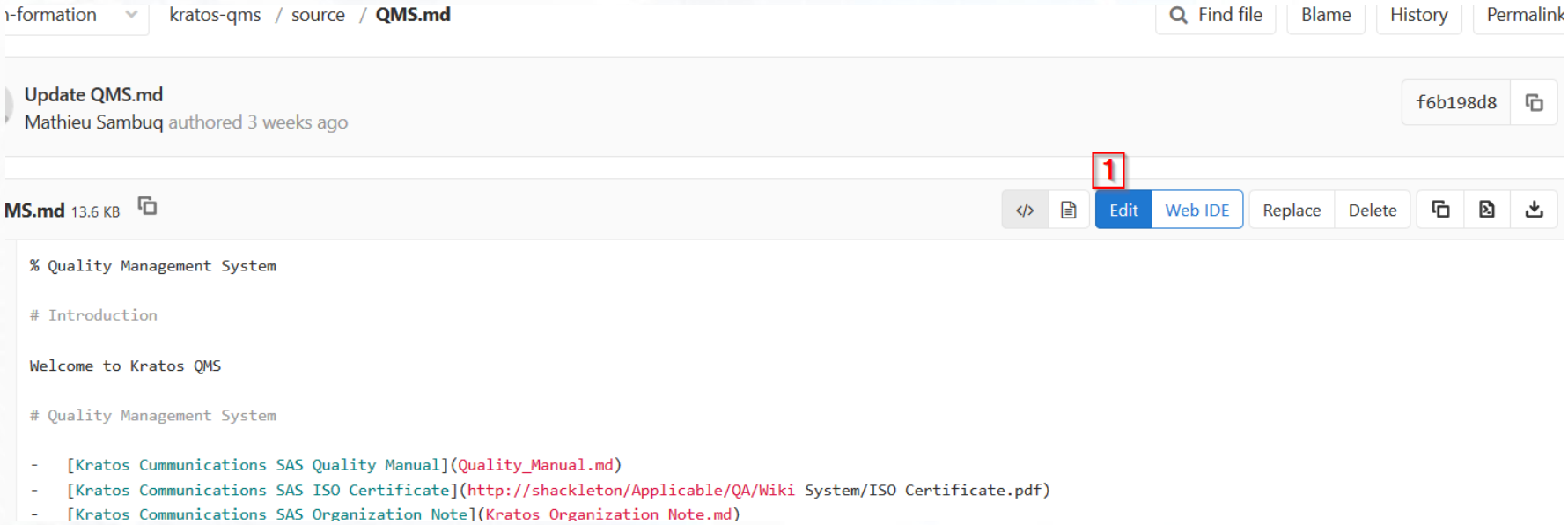
Editing one file - File selection



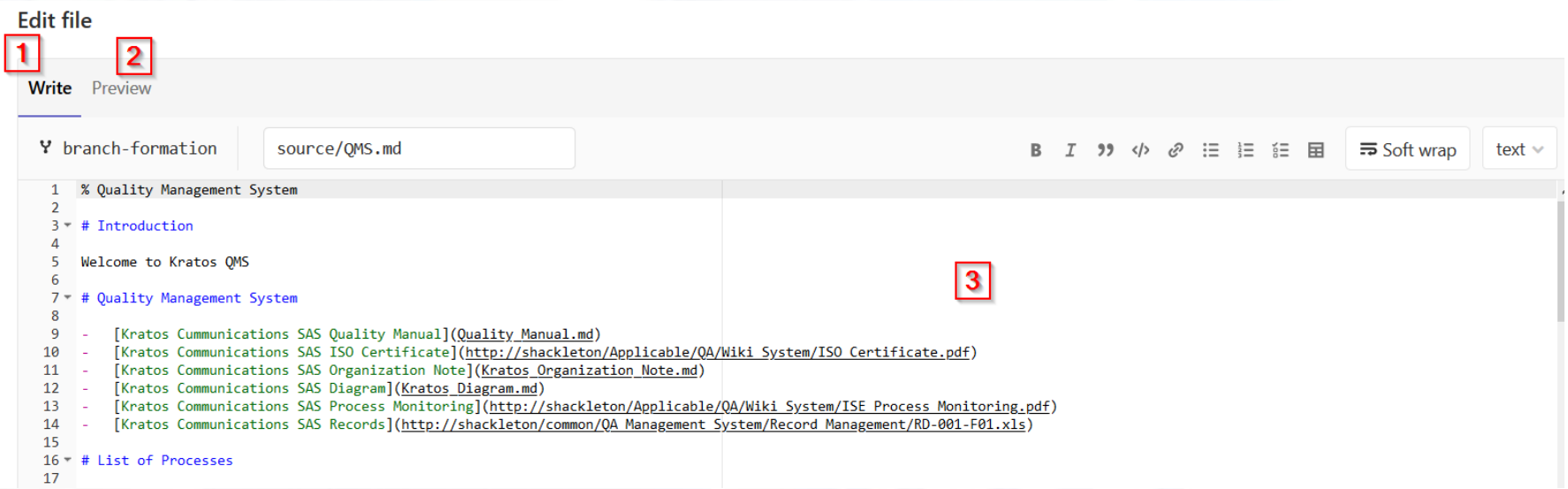
Browse to find the file you want to edit.

Note: All processes are under source. All image are under source/Images.

Editing one file - File selection

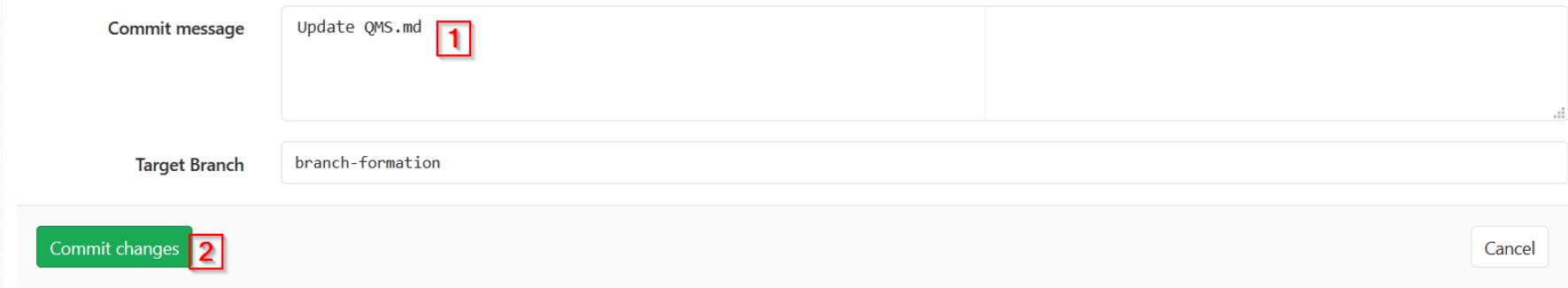


Editing one file - File edition

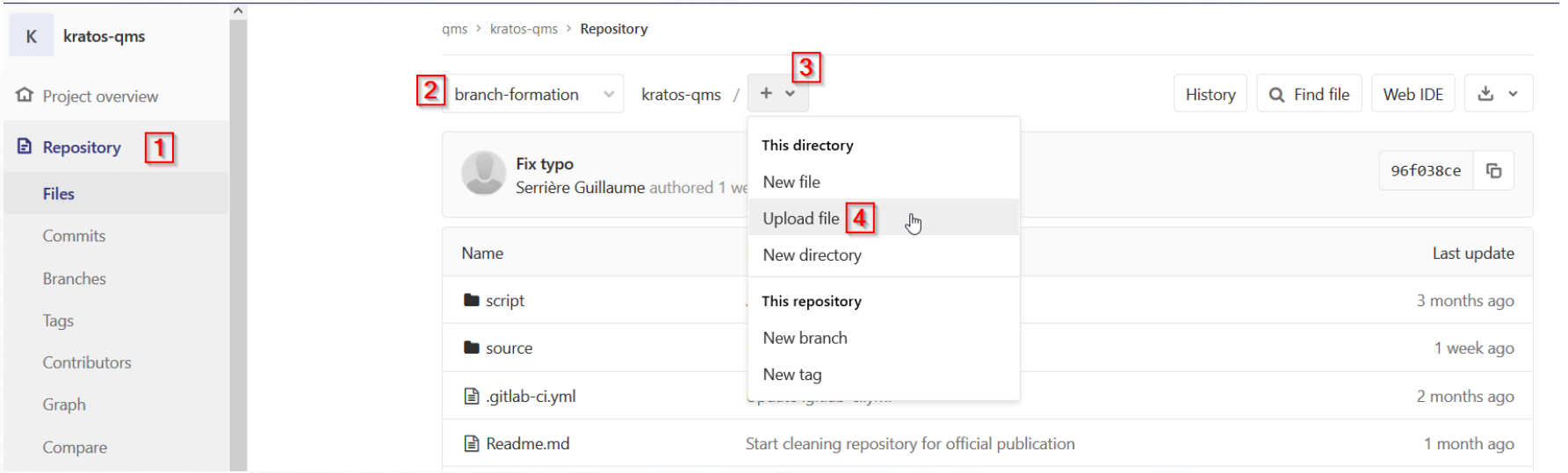


1. Code edition mode
2. Preview mode
3. Edition zone

Editing one file - File commit

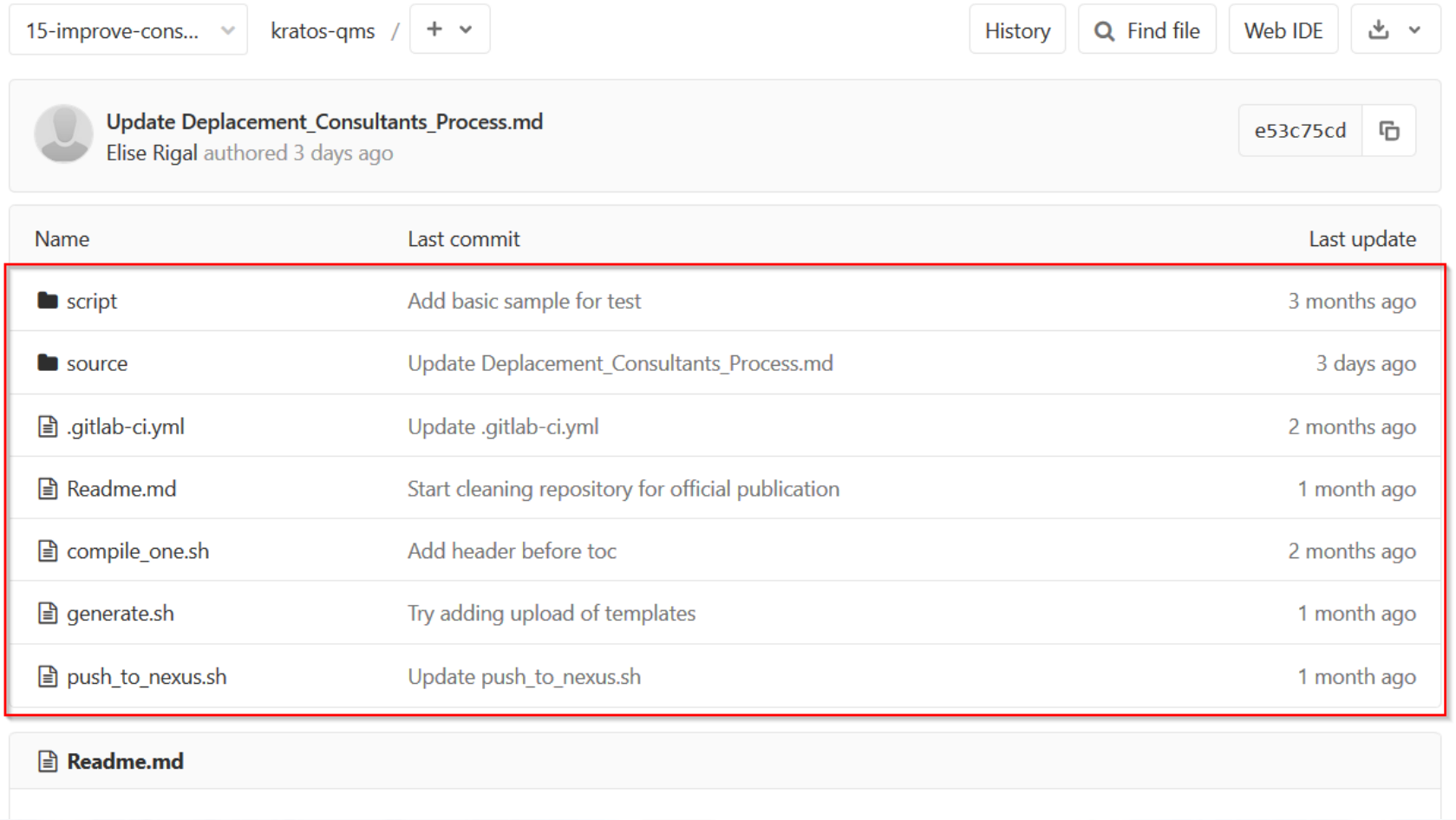


Editing one file - Add file



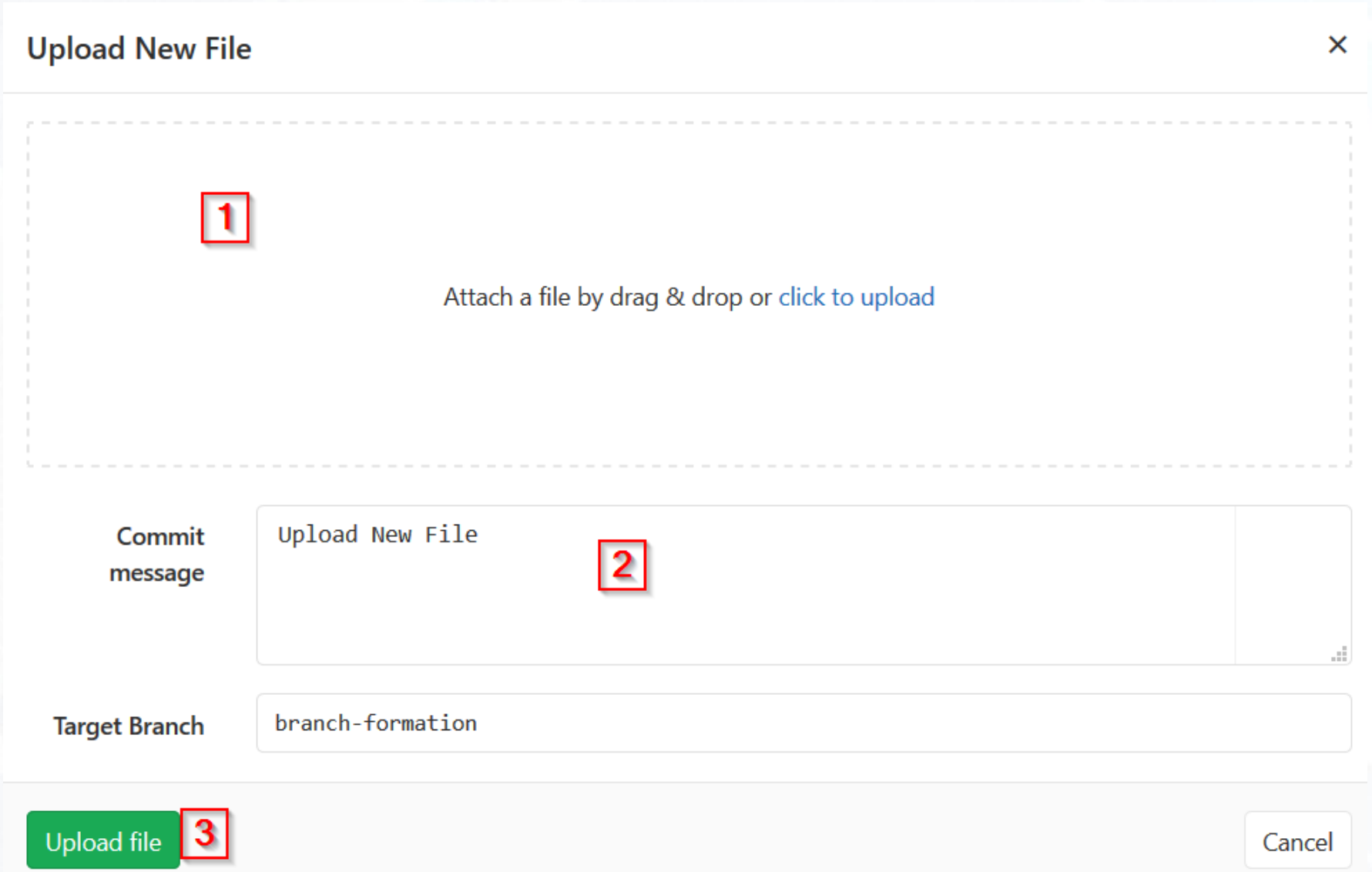
Note: Select the right branch (2) before uploading the file.

Editing one file - Add file



Browse to the directory you want to add the file to.

Editing one file - Add file

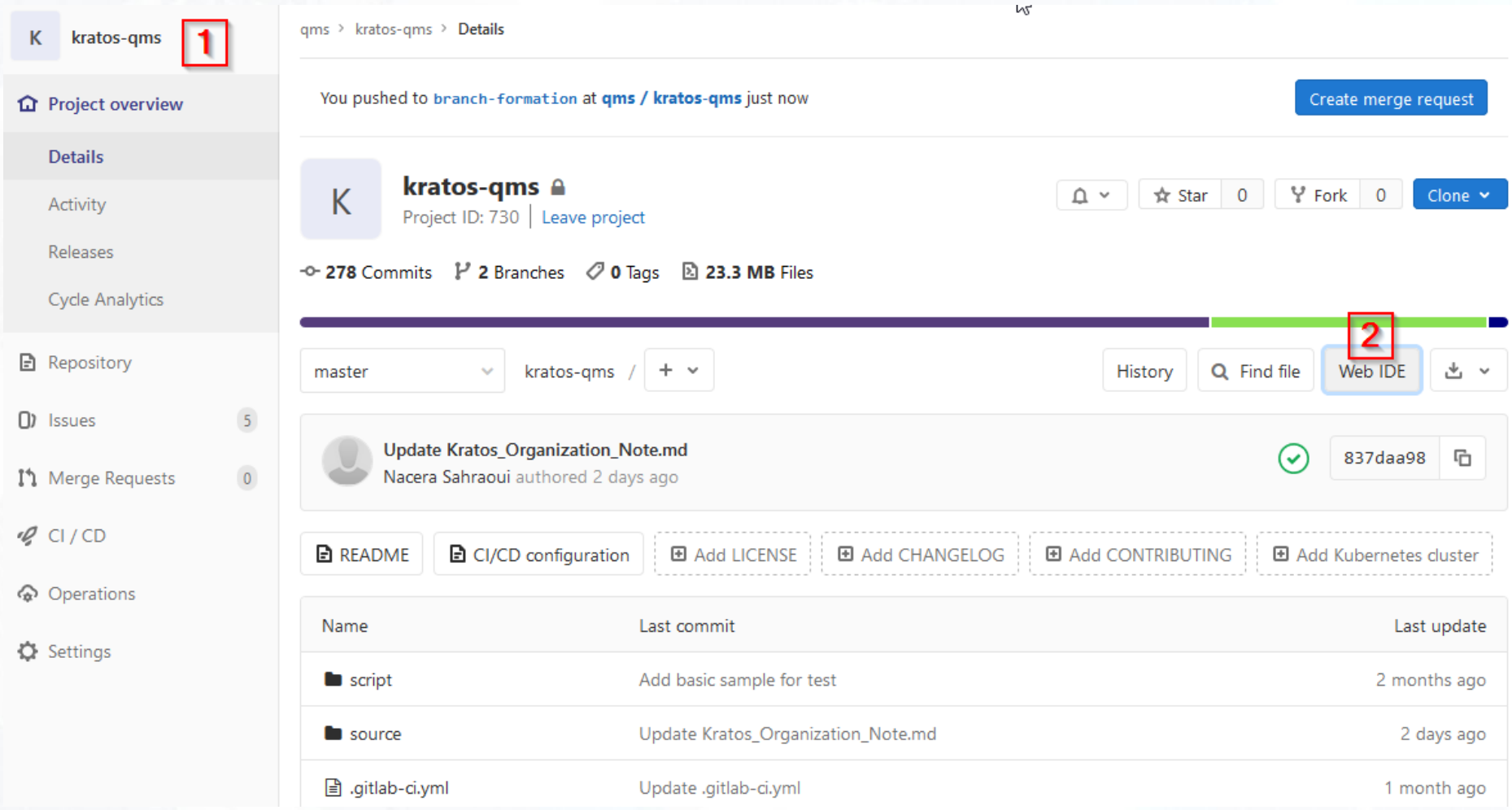


Note: Drag and drop from file explorer to the zone 1.

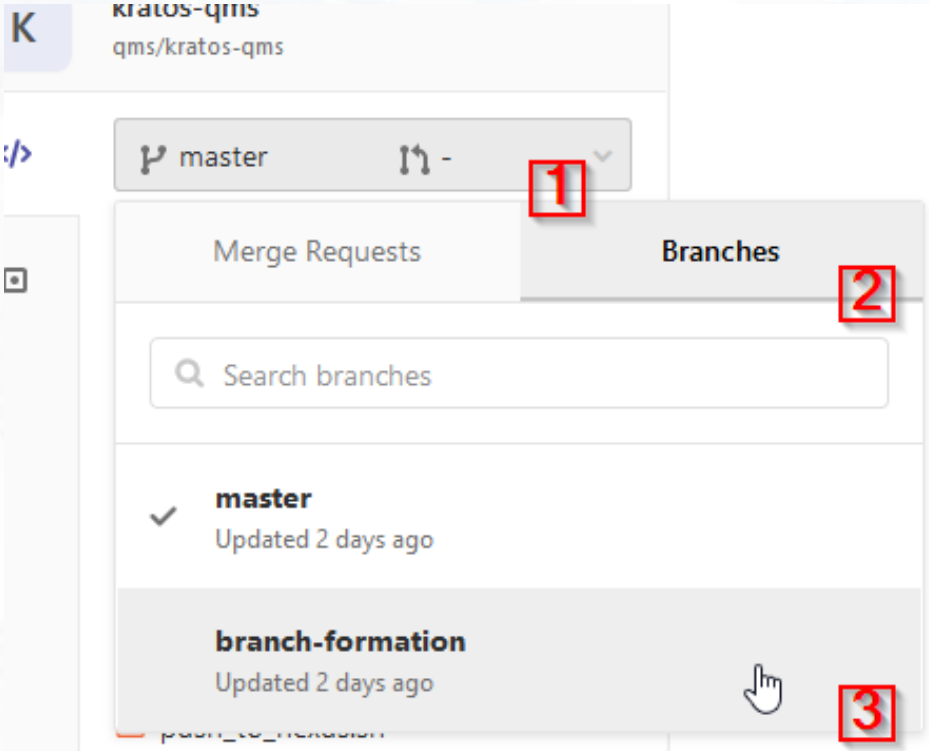
Gitlab Web IDE

Gitlab provides with its web interface an Integrated Development Environment that allows to modify several files at the same time and to make more precise commit.

Gitlab Web IDE

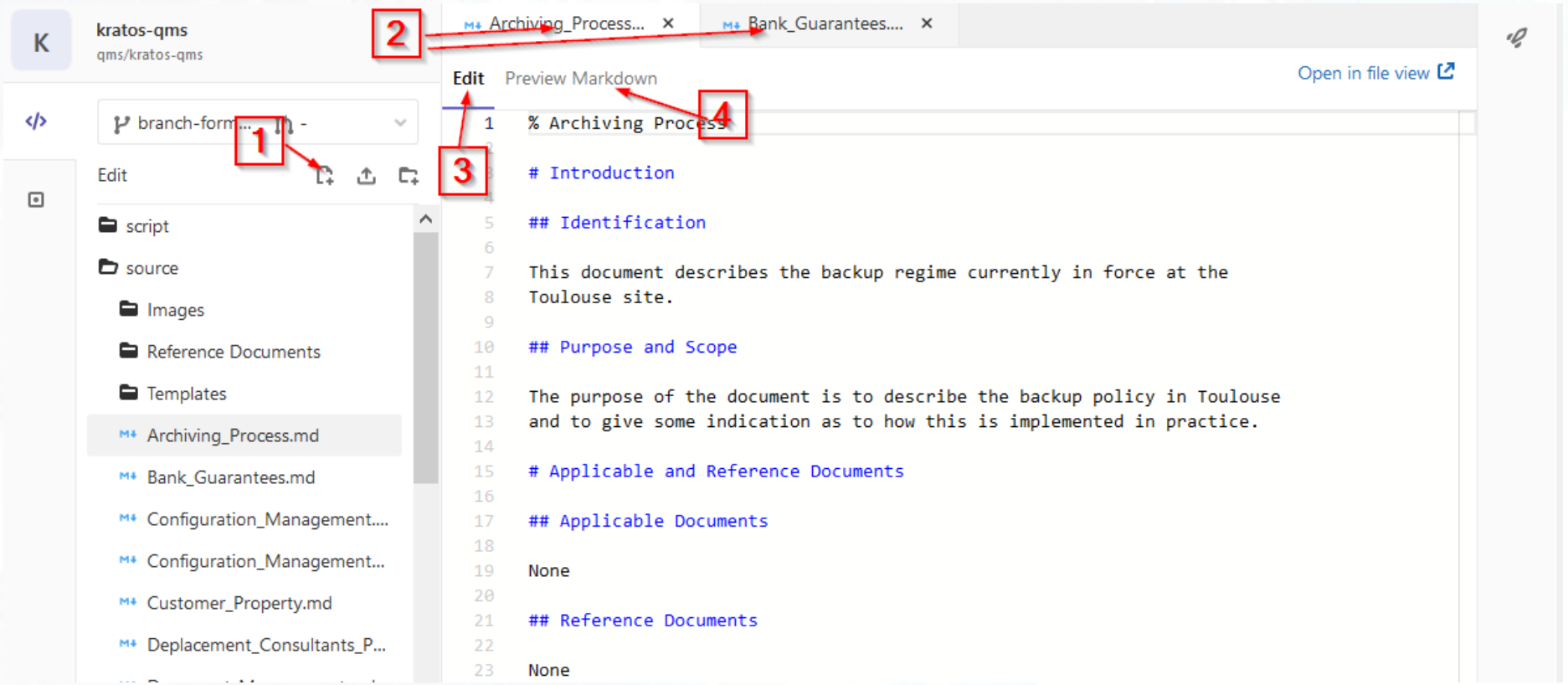


Gitlab Web IDE



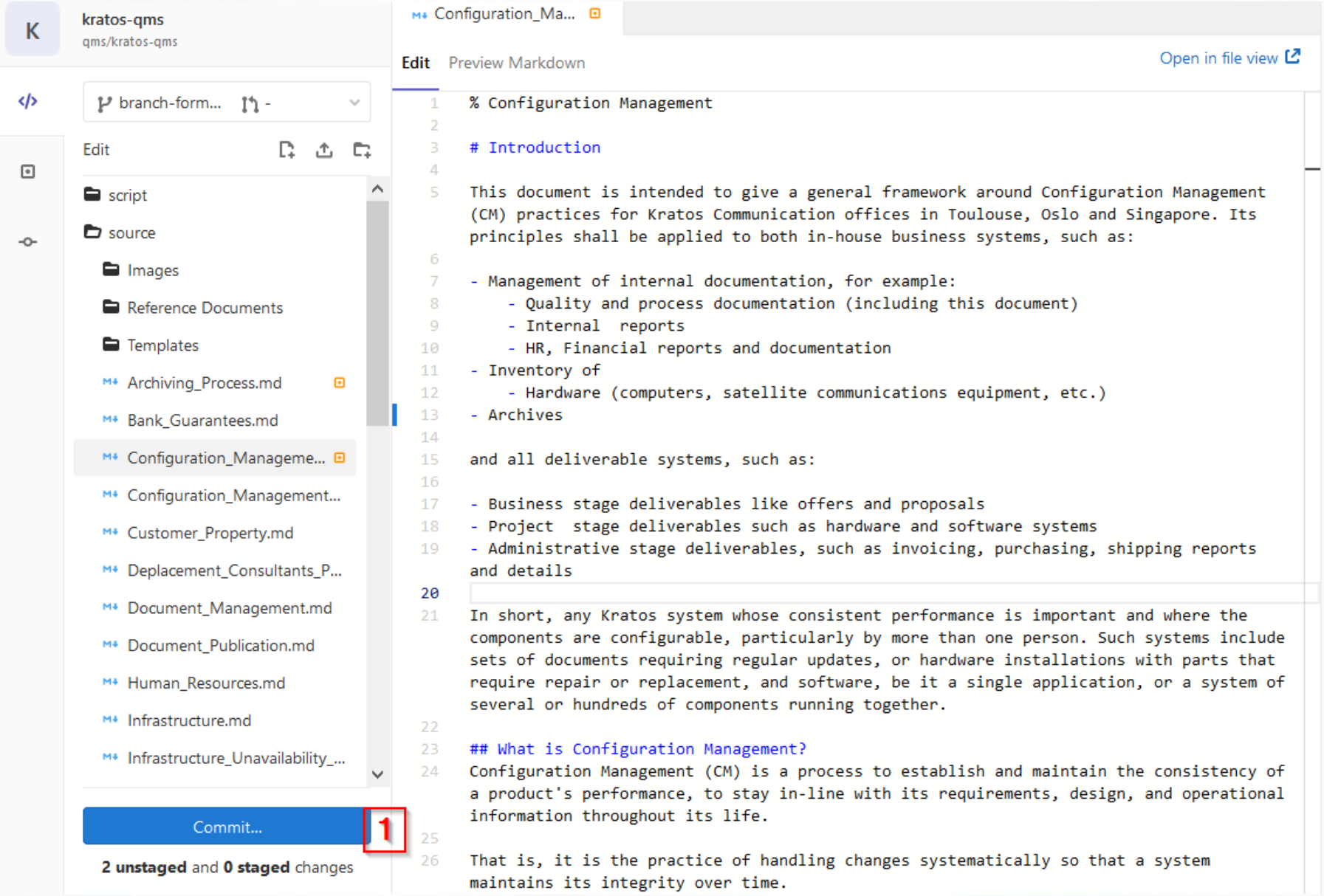
Make and review changes in the browser with t

Gitlab Web IDE - Modify the code

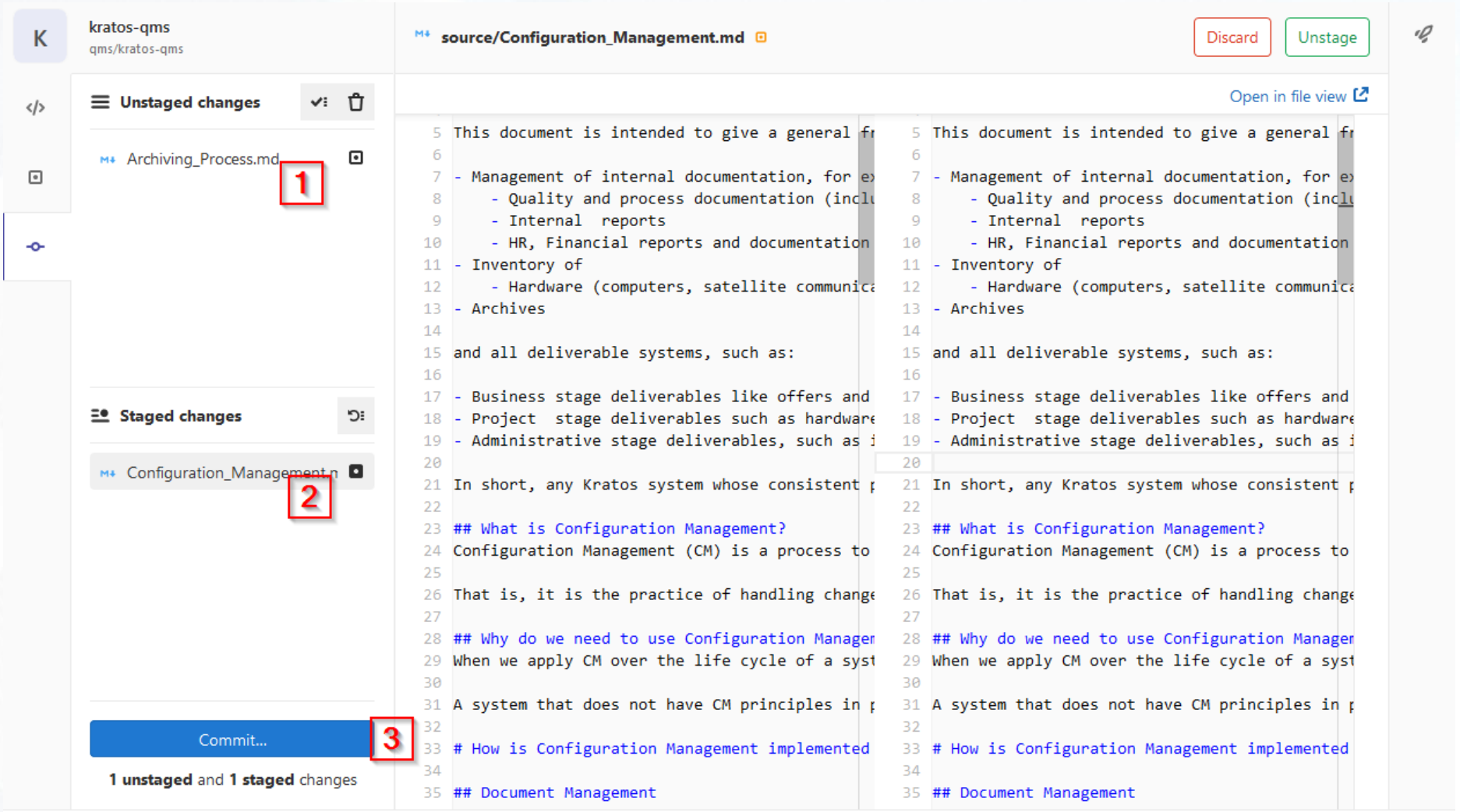


1. New file
2. Tab navigation
3. Code edit view
4. Preview

Gitlab Web IDE - Commit code on branch

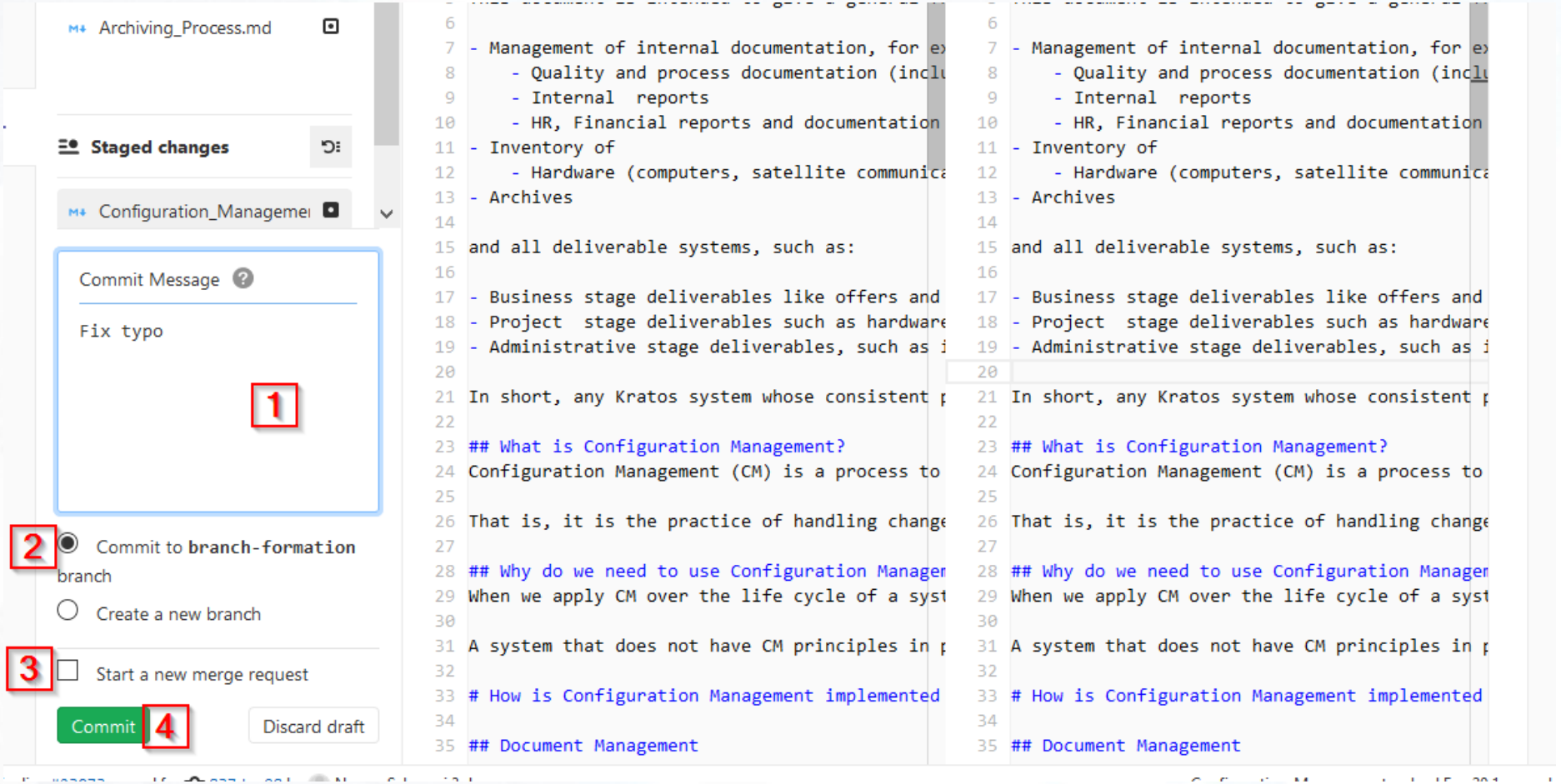


Gitlab Web IDE - Commit code on branch



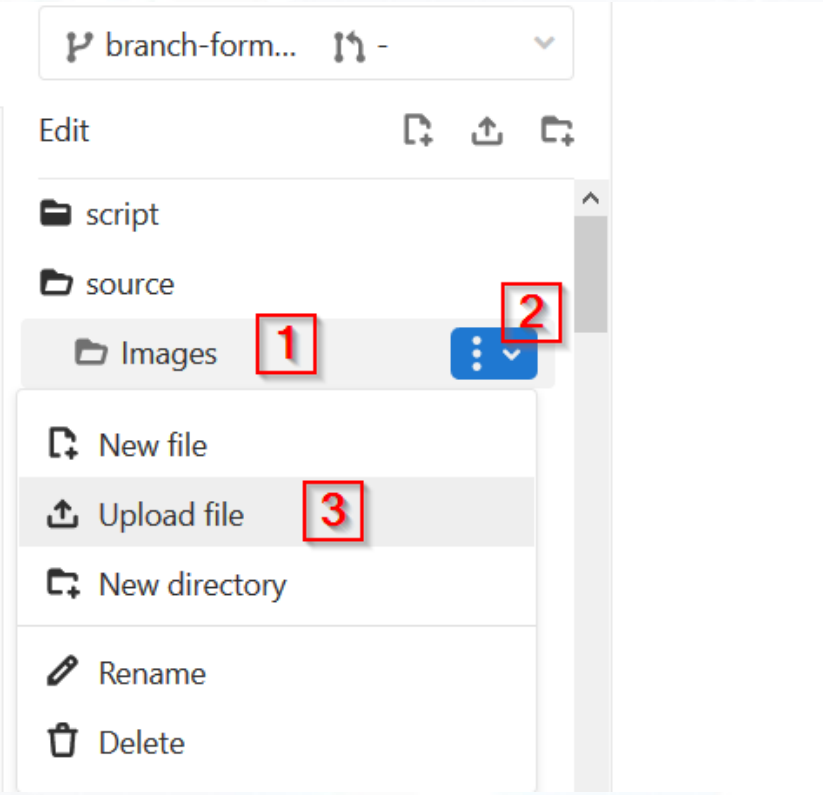
1. Add file to commit
2. File in commit
3. Commit

Gitlab Web IDE - Commit code on branch



1. Enter commit message
2. Select commit on your branch
3. Unclick start merge request
4. Click commit

Gitlab Web IDE - Upload file

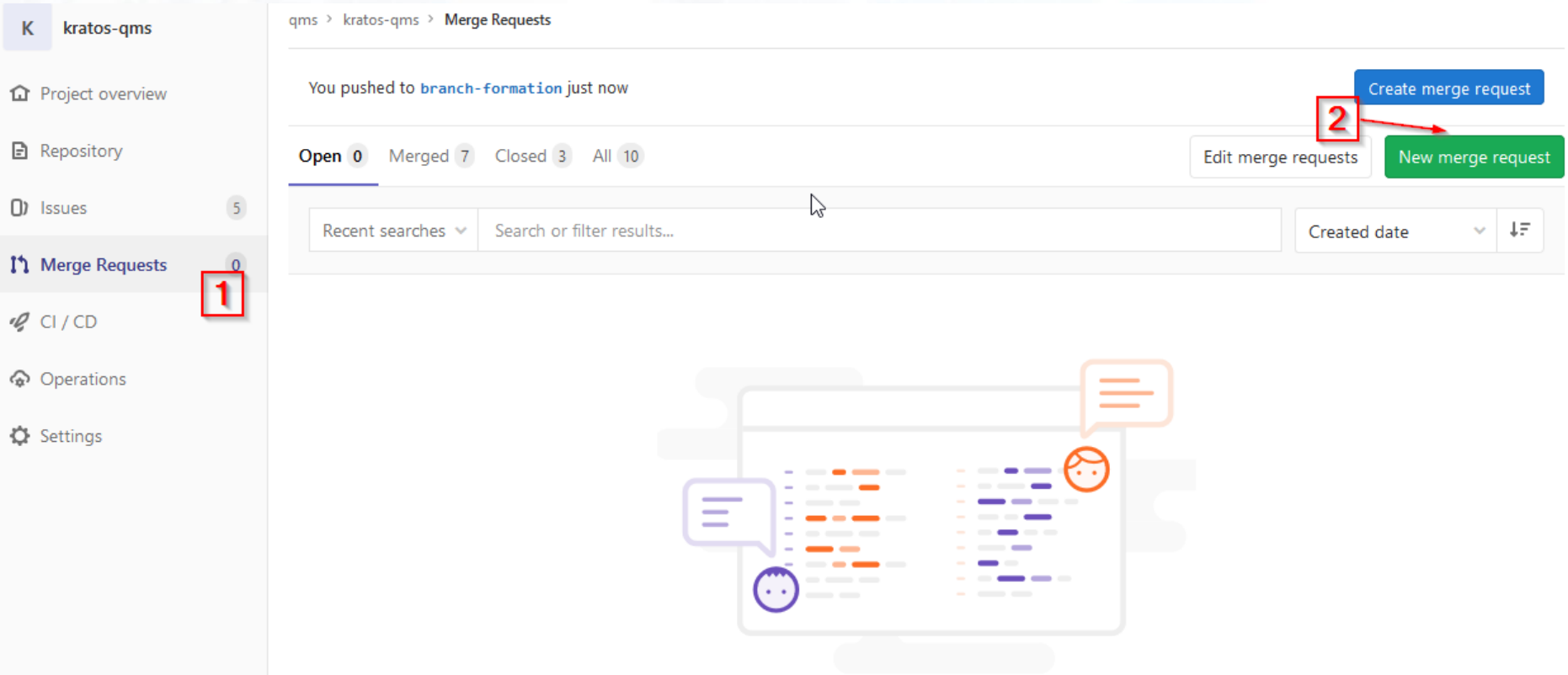


1. Select folder
2. Click on arrow
3. Select upload file
4. Select the file you want to upload

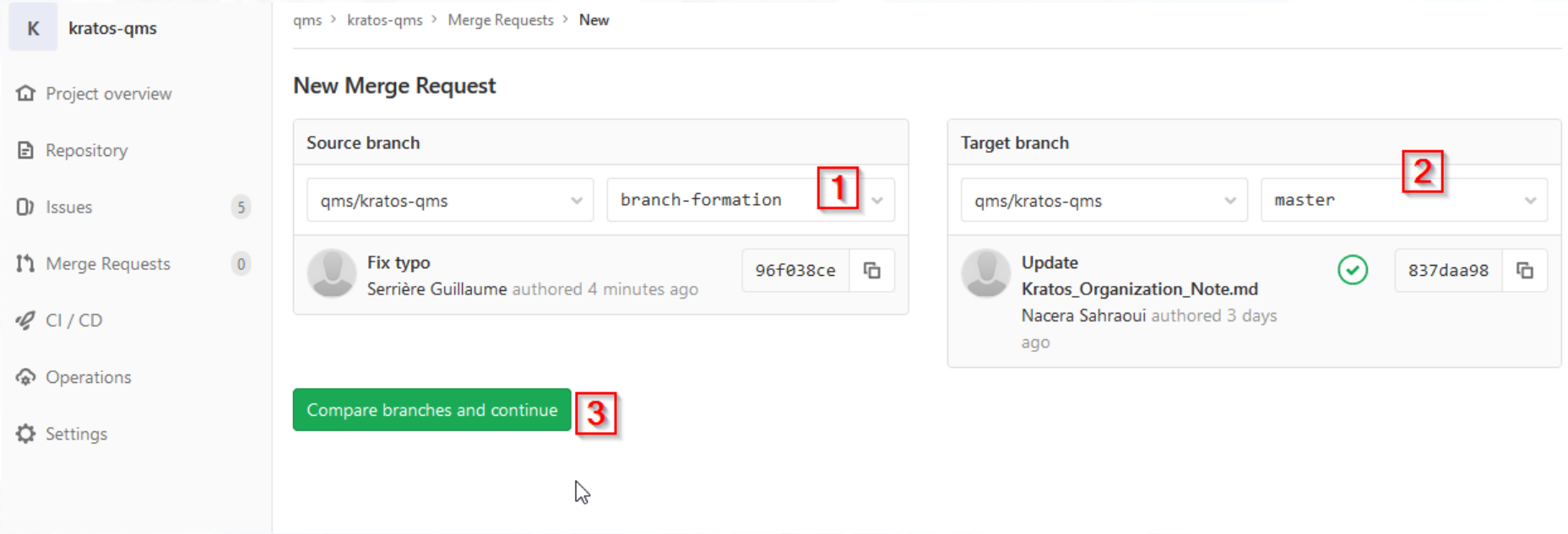
Third step

The third step is to submit a merge request for review.

Start merge request

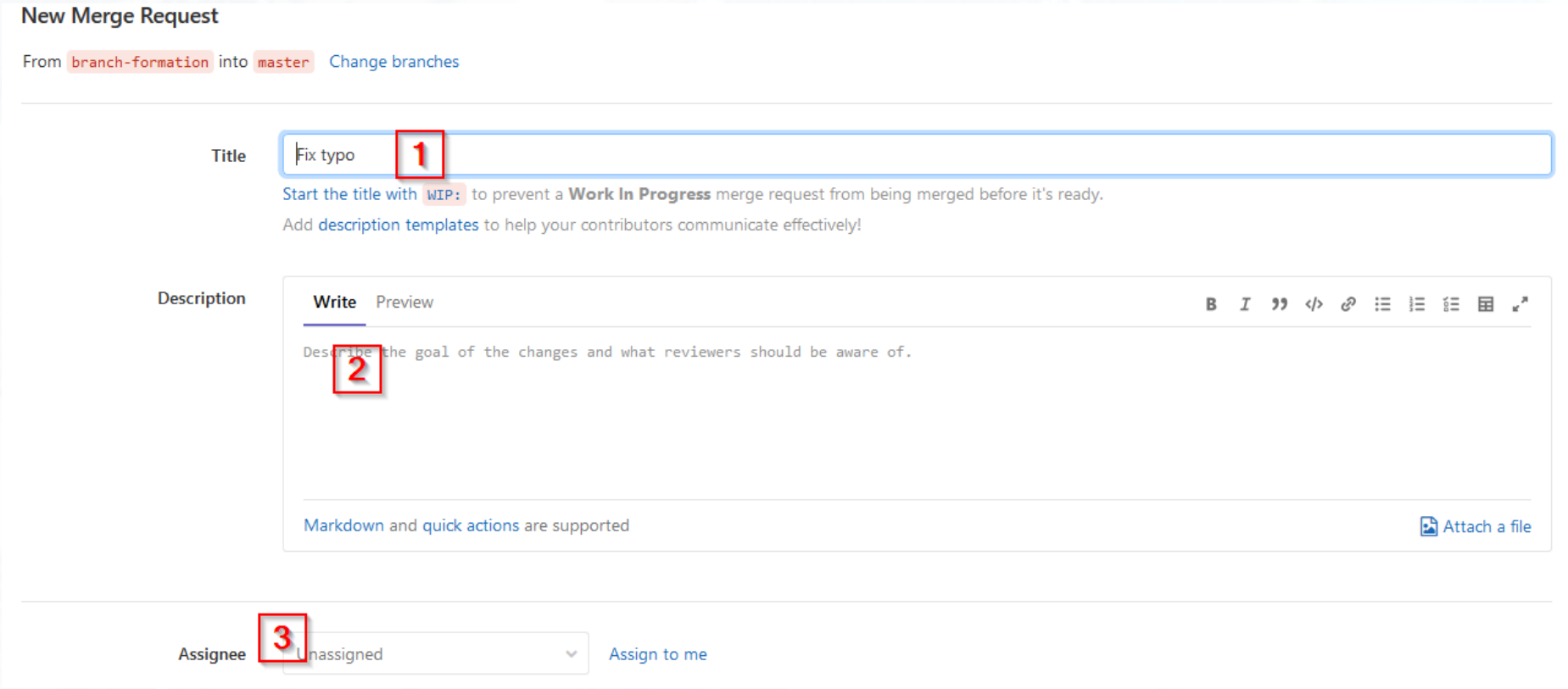


Start merge request



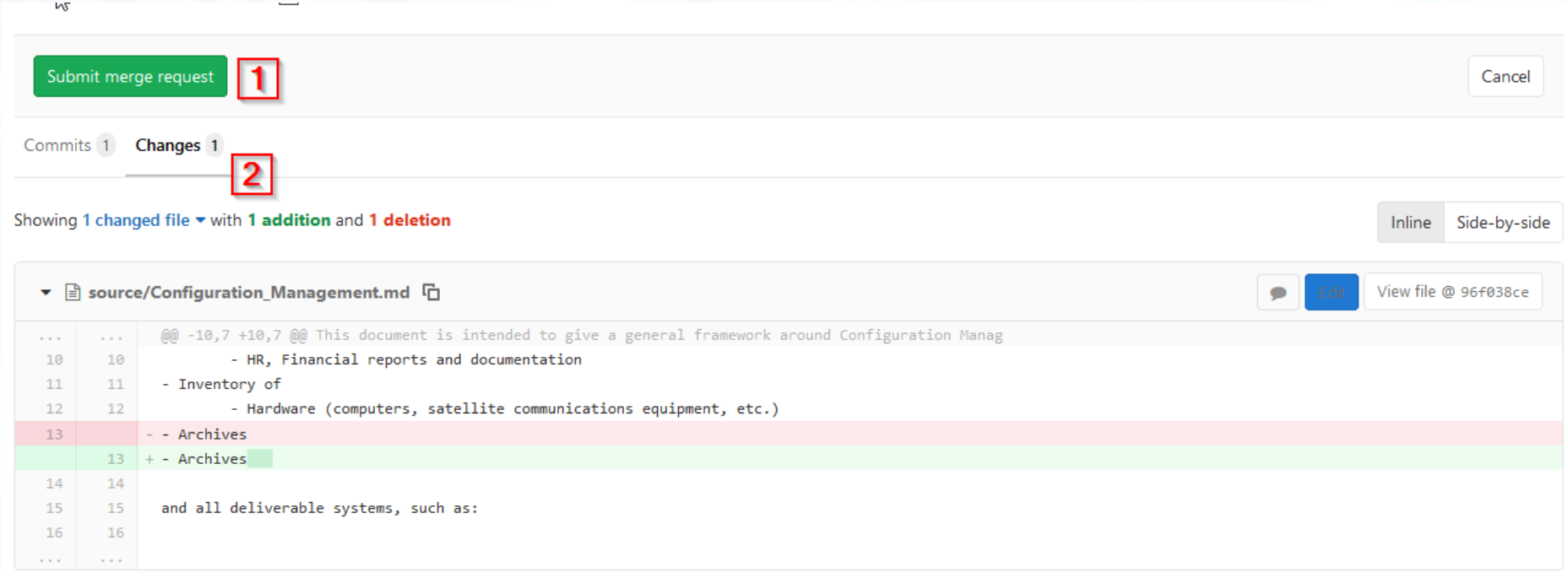
1. Select your branch
2. Select master
3. Continue

Start merge request



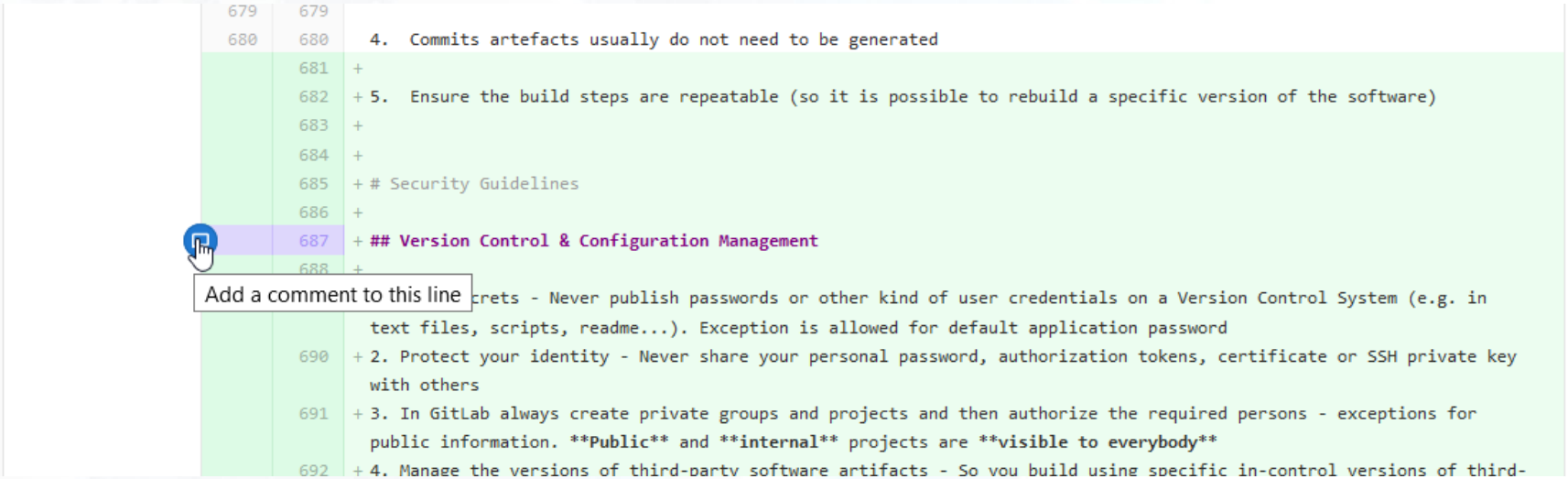
1. Merge request title
2. Description
3. Assign people (for notification)

Start merge request

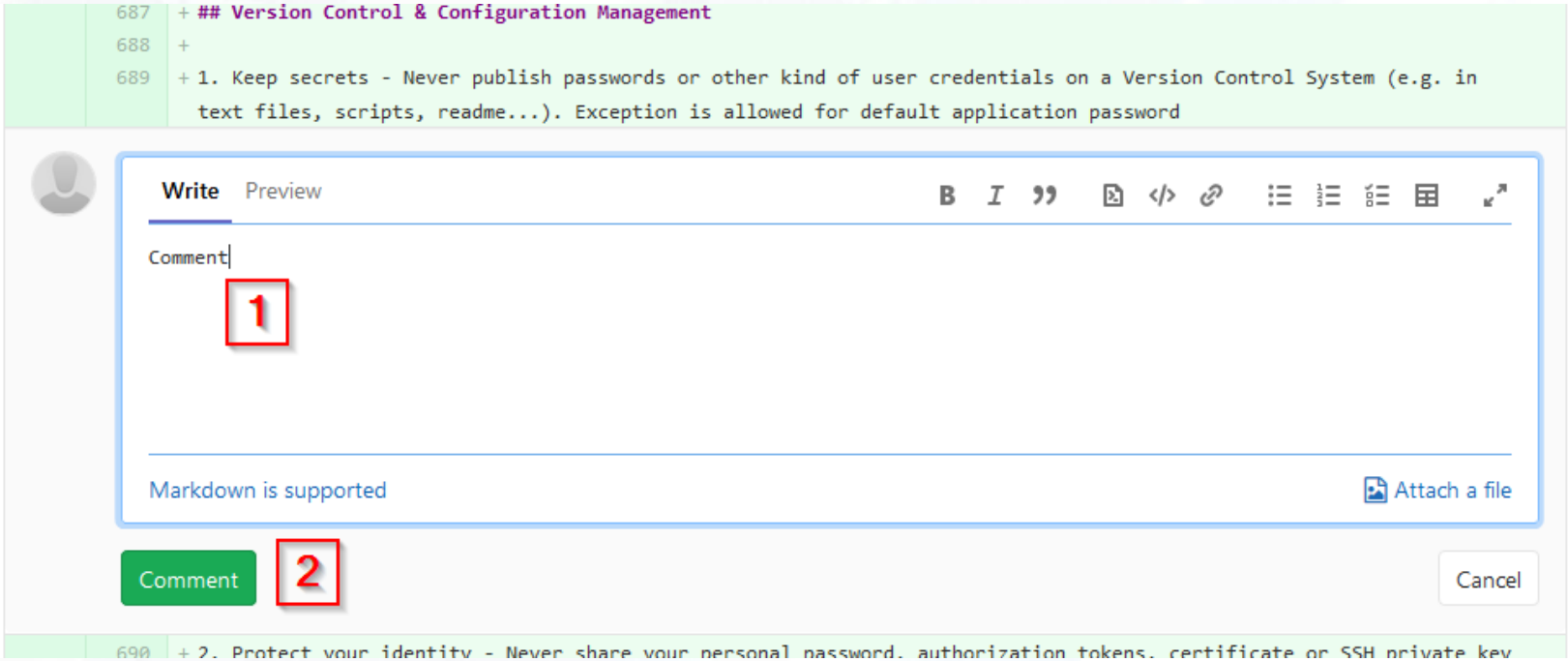


1. Create merge request
2. Change review

Merge request comment

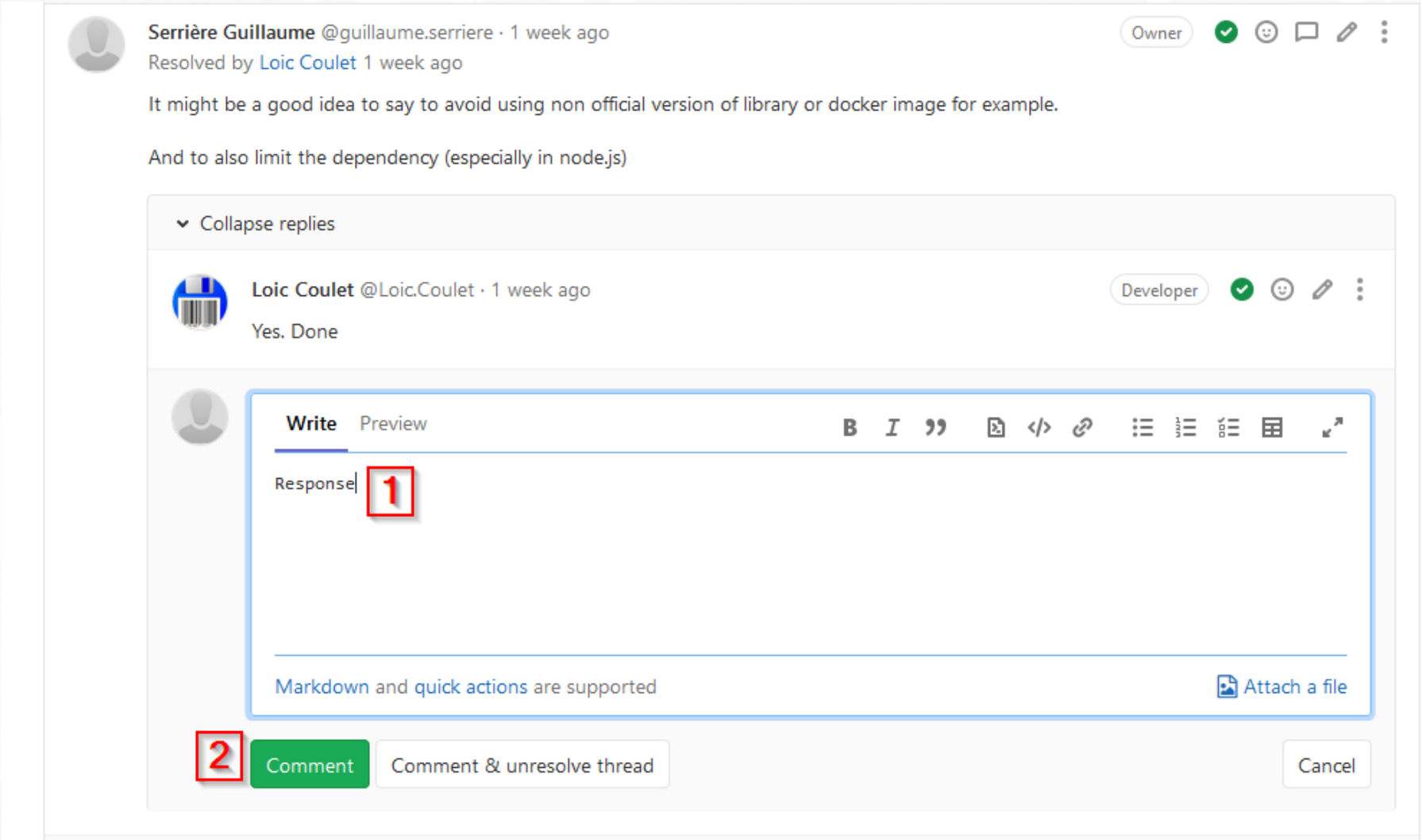


Merge request comment



- Comment zone
- Post comment

Merge request comment



- Reply zone
- Post reply