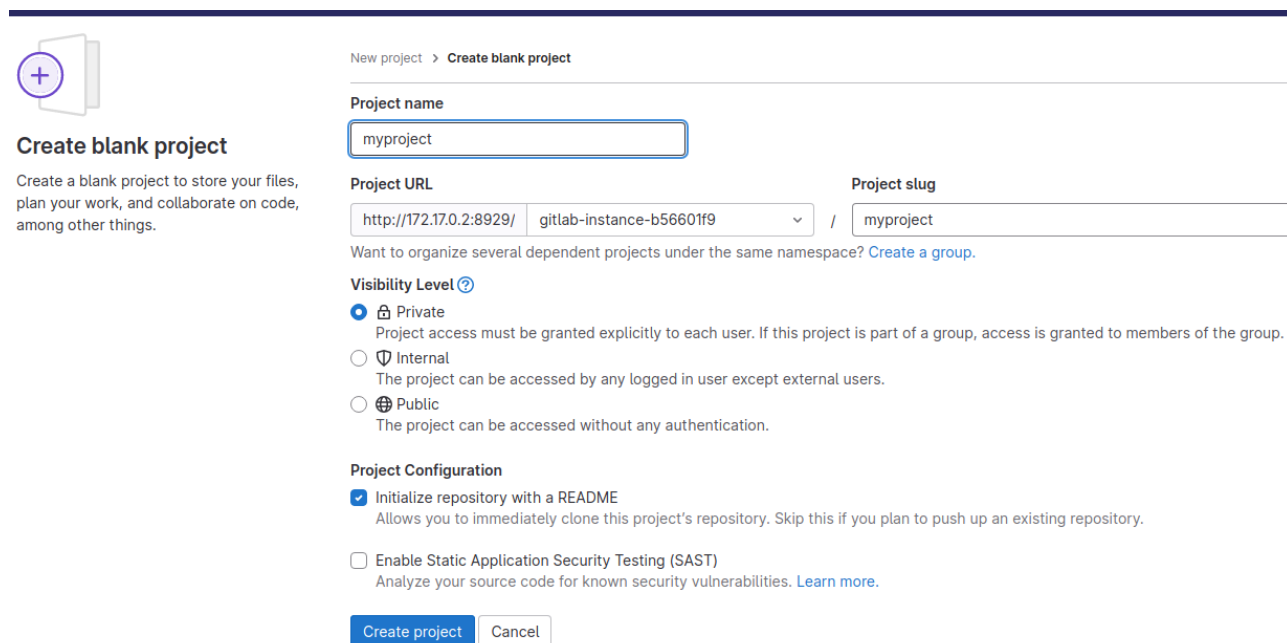


# Упражнение по использованию основных команд git

Для начала работ перейдем в веб-интерфейс Gitlab (его подробная установка с помощью docker-compose описана в соответствующем методическом материале к этому уроку) и выполним создание проекта.

После прохождения аутентификации, в правом верхнем углу нажимает кнопку "New project" - "Create blank project".

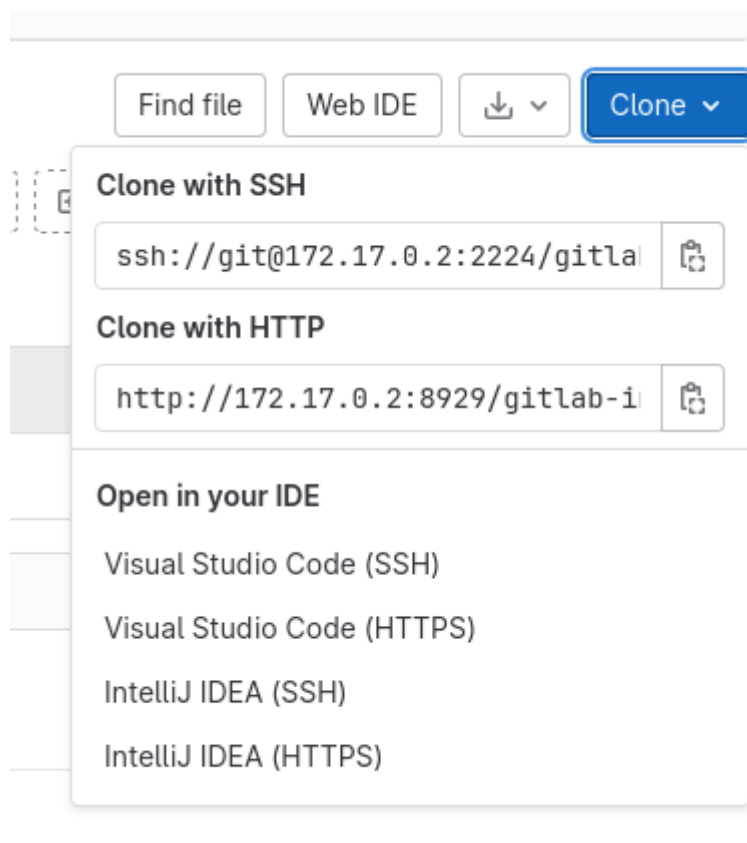
Далее вводим наименование нашего проекта (в нашем примере назовем его "myproject", все остальные настройки для данной практической работы оставим по умолчанию и нажимаем "Create project":



The screenshot shows the 'Create blank project' interface in GitLab. On the left, there is a sidebar with a plus icon and the text 'Create blank project' followed by a description: 'Create a blank project to store your files, plan your work, and collaborate on code, among other things.' The main form area has a breadcrumb 'New project > Create blank project'. It contains several sections: 'Project name' with a text input field containing 'myproject'; 'Project URL' with a dropdown menu showing 'http://172.17.0.2:8929/' and 'gitlab-instance-b56601f9', followed by a slash and a 'Project slug' input field containing 'myproject'; a link 'Want to organize several dependent projects under the same namespace? Create a group.'; 'Visibility Level' with three radio button options: 'Private' (selected), 'Internal', and 'Public', each with a brief description; 'Project Configuration' with two checkboxes: 'Initialize repository with a README' (checked) and 'Enable Static Application Security Testing (SAST)' (unchecked). At the bottom, there are two buttons: 'Create project' (blue) and 'Cancel' (white).

Проект успешно создан.

Скопируем ссылку для клонирования проекта - для этого перейдите в выпадающий список "Clone" и скопируйте ссылку в графе "Clone with HTTP" (в дальнейшем нам понадобится эта ссылка)



Переходим в консоль и выполняем конфигурацию нашего локального пользователя, от имени которого будет происходить работа (подставьте свои значения для переменных):

```
$ git config --global user.name "filipp"
$ git config --global user.email "filipp@fake.com"
```

```
filipp@filipp-notebook:~/Desktop$ git config --global user.name "filipp"
filipp@filipp-notebook:~/Desktop$ git config --global user.email "filipp@fake.com"
filipp@filipp-notebook:~/Desktop$
```

Выполним клонирование удаленного проекта, воспользовавшись ссылкой клонирования. В запросе авторизации вводим пользователя root и пароль от этой учетной записи Gitlab:

```
$ git clone
http://172.17.0.2:8929/gitlab-instance-fe42b4be/myproj.git
```

```
filipp@filipp-notebook:~/Desktop$ git clone http://172.17.0.2:8929/gitlab-instance-b56601f9/myproject.git
Cloning into 'myproject'...
Username for 'http://172.17.0.2:8929': root
Password for 'http://root@172.17.0.2:8929':
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

Переходим в созданную директорию проекта:

```
$ cd myproject  
$ ll
```

```
filipp@filipp-notebook:~/Desktop$ cd myproject/  
filipp@filipp-notebook:~/Desktop/myproject$ ll  
total 20  
drwxrwxr-x  3 filipp filipp 4096 фев  1 18:19 ./  
drwxrwxrwx 14 filipp filipp 4096 фев  1 18:19 ../  
drwxrwxr-x  8 filipp filipp 4096 фев  1 18:19 .git/  
-rw-rw-r--  1 filipp filipp 6238 фев  1 18:19 README.md
```

--

Выполним изменение файла README.md, внося в него изменения:

```
$ nano README.md
```

```
GNU nano 6.2                                README.md *  
  
some changes  
  
# myproject  
  
## Getting started  
  
To make it easy for you to get started with GitLab, here's a list of
```

Сохраняем изменения, вызовем команду для просмотра текущего состояния нашей локальной версии проекта:

```
$ git status
```

```
filipp@filipp-notebook:~/Desktop/myproject$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
```

Видим сообщение, что файл README.md был изменен. Произведем индексацию измененных файлов и повторим вызов команды просмотра изменений в локальной версии нашего проекта:

```
$ git add .
$ git status
```

```
filipp@filipp-notebook:~/Desktop/myproject$ git add .
filipp@filipp-notebook:~/Desktop/myproject$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md
```

Изменения проиндексированы, подготовим к отправке в удаленный репозиторий набор проиндексированных изменений с комментарием (исп. флаг "-m"):

```
$ git commit -m 'change readme'
```

```
filipp@filipp-notebook:~/Desktop/myproject$ git commit -m 'readme changed'
[main 94c2ae0] readme changed
1 file changed, 3 insertions(+)
```

Снова вызовем команду просмотра изменений в локальном репозитории

```
$ git status
```

```
filipp@filipp-notebook:~/Desktop/myproject$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

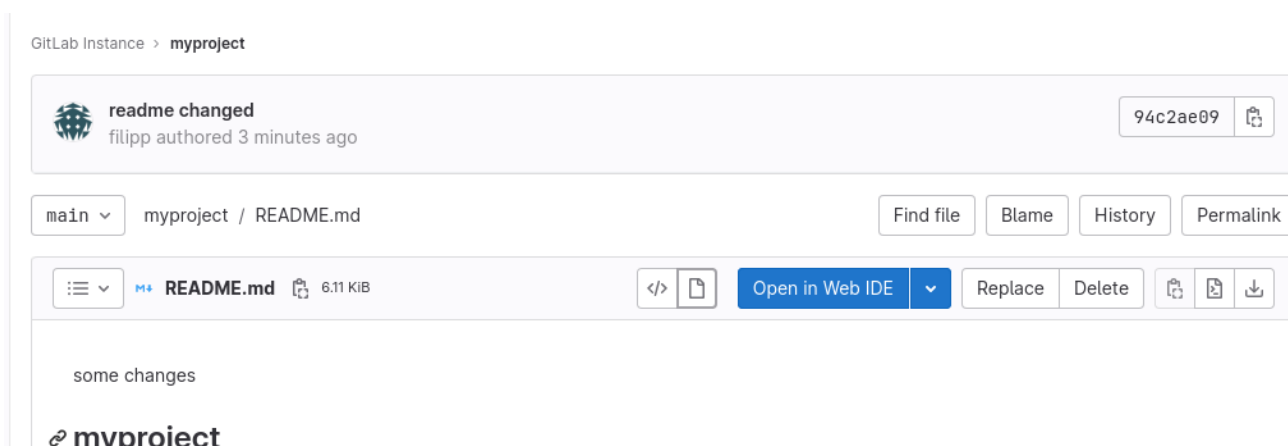
nothing to commit, working tree clean
```

Обратите внимание, что локальный репозиторий опережает удаленный на 1 коммит. Выполним выгрузку локальных изменений в удаленный репозиторий:

```
$ git push
```

```
filipp@filipp-notebook:~/Desktop/myproject$ git push
Username for 'http://172.17.0.2:8929': root
Password for 'http://root@172.17.0.2:8929':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 279 bytes | 279.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
To http://172.17.0.2:8929/gitlab-instance-b56601f9/myproject.git
  90238c0..94c2ae0  main -> main
```

Изменения успешно отправлены. Перейдем в веб-интерфейс Gitlab и убедимся, что они были успешно применены:



--

Выполним команду просмотра текущих веток в локальном проекте:

```
$ git branch
```

```
filipp@filipp-notebook:~/Desktop/myproject$ git branch
* main
```

В данный момент в проекте одна основная ветка `main`. Выполним создание новой ветки, для этого используем флаг `"-b"` в команде переключения веток (будет выполнено создание новой ветки и сразу переключение на нее):

```
$ git checkout -b mybranch
```

```
filipp@filipp-notebook:~/Desktop/myproject$ git checkout -b mybranch
Switched to a new branch 'mybranch'
```

Создание и переключение на новую ветку выполнено. Убедимся в этом, повторим команду просмотра веток:

```
$ git branch
```

```
filipp@filipp-notebook:~/Desktop/myproject$ git branch
main
* mybranch
```

--

Находясь в новой ветке, выполните создание нового файла в проекте с произвольным содержимым:

```
$ nano my.script
```

```
GNU nano 6.2 my.script
myscript content
```

После сохранения файла вызовем команду просмотра состояния локального проекта:

```
$ git status
```



```
filipp@filipp-notebook:~/Desktop/myproject$ git status
On branch mybranch
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    my.script

nothing added to commit but untracked files present (use "git add" to track)
```

Выполним индексирование изменений и подготовку к отправке набора этих изменений:

```
$ git add my.script
$ git commit -m 'my.script added'
```

```
filipp@filipp-notebook:~/Desktop/myproject$ git add my.script
filipp@filipp-notebook:~/Desktop/myproject$ git commit -m 'my.script added'
[mybranch 0a200b9] my.script added
1 file changed, 1 insertion(+)
create mode 100644 my.script
```

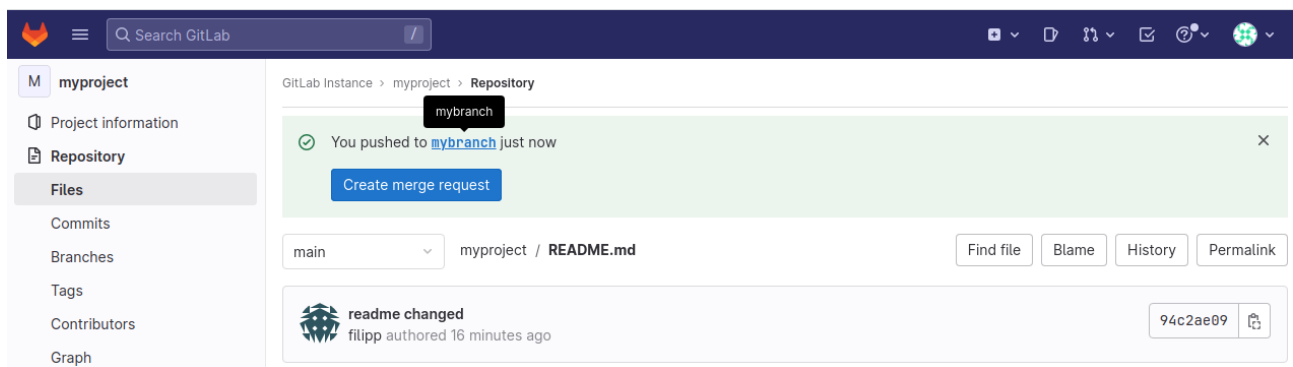
Выполним отправку в удаленный репозиторий набора изменений (в том числе создание новой ветки в удаленном репозитории)

```
$ git push origin mybranch
```

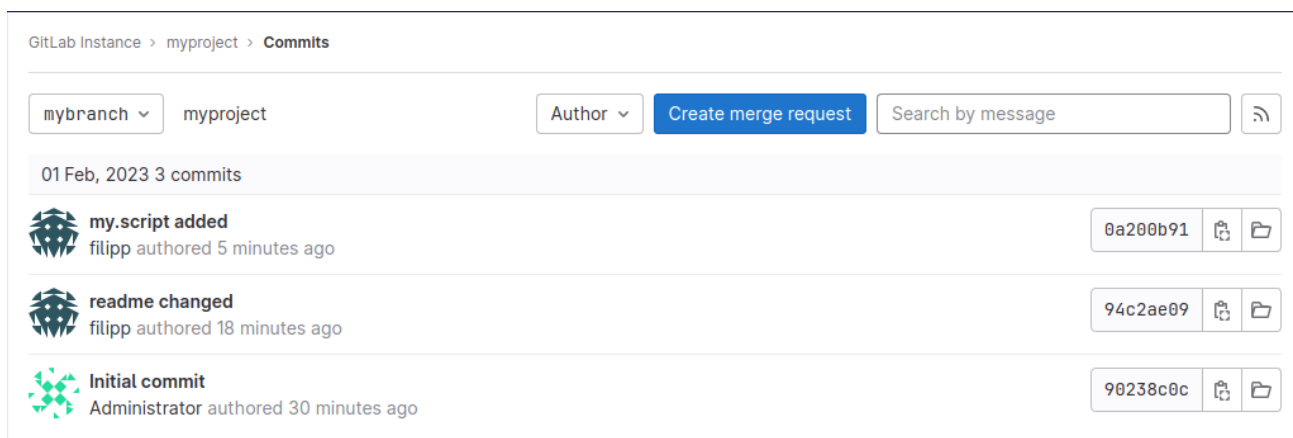
```
filipp@filipp-notebook:~/Desktop/myproject$ git push origin mybranch
Username for 'http://172.17.0.2:8929': root
Password for 'http://root@172.17.0.2:8929':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 288 bytes | 288.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: To create a merge request for mybranch, visit:
remote:   http://172.17.0.2:8929/gitlab-instance-b56601f9/myproject/-/merge_requests/new?merge_request[source_branch%5D=mybranch
remote:
To http://172.17.0.2:8929/gitlab-instance-b56601f9/myproject.git
* [new branch]      mybranch -> mybranch
```

Изменения были успешно отправлены в удаленный репозиторий.

Переходим в веб-интерфейс Gitlab и видим уведомление о создании новой ветки:



Переходим в новую ветку (можно воспользоваться ссылкой в уведомлении о создании ветки)



Нажимаем на кнопку "Create merge request"

В окне создания merge-request описываем свой запрос на слияние (в целях демонстрации назначим себя согласующим лицом merge-request. В реальной жизни это может быть кто-то из ваших коллег по команде):



# New merge request

From `mybranch` into `main` [Change branches](#)

## Title (required)

my.script added

Start the title with `Draft:` to prevent a merge request draft from merging before it's ready.

Add [description templates](#) to help your contributors communicate effectively!

## Description

**Write** Preview

**B** *I* ~~S~~ `code` [link](#)

добавлен новый файл

Supports [Markdown](#). For [quick actions](#), type `/`.

## Assignee

Administrator

## Reviewer

Unassigned

## Milestone

Select milestone

## Labels

Labels

## Merge options

☒ Delete source branch when merge request is accepted.

☐ Squash commits when merge request is accepted. [?](#)

Create merge request

Cancel

Commits 1

Changes 1

Нажимаем на кнопку "Create merge-request".

В открывшемся окне имитируем роль члена команды, выполняющего code-review (поскольку мы были назначены на согласование)

Удовлетворяем запрос на merge-request нажатием на кнопку "merge":

## my.script added

Edit

Code ▾



Open Administrator requested to merge mybranch into main just now

Overview 0

Commits 1

Pipelines 0

Changes 1

добавлен новый файл



Approval is optional

Ready to merge!

☒ Delete source branch ☐ Squash commits ☐ Edit commit message

1 commit and 1 merge commit will be added to main.

Merge

## Activity

Sort or filter ▾

Administrator assigned to @root just now

Write Preview



Mark as done



## Assignee

Edit

Administrator

## 0 Reviewers

Edit

None - assign yourself

## Labels

Edit

None

## Milestone

Edit

None

## Time tracking



No estimate or time spent

## Lock merge request

Edit

Unlocked

## Notifications



## 1 participant



Reference: gitlab-instance-b56...

Source branch: mybranch

Слияние веток выполнено успешно:

## my.script added

Edit

Code ▾



Merged Administrator requested to merge mybranch into main just now

Overview 0

Commits 1

Pipelines 0

Changes 1

добавлен новый файл



Approval is optional

Merged by Administrator just now Revert Cherry-pick

## Merge details

- Changes merged into main with [9818bfbd](#).
- Deleted the source branch.

--

Вернемся на страницу нашего проекта и убедимся в том, что слияние веток произошло и файл был добавлен в ветке `main`:

main ▾

myproject /

+ ▾



History

Find file

Web IDE

⬇ ▾

Clone ▾

Name	Last commit	Last update
 README.md	readme changed	25 minutes ago
 my.script	my.script added	12 minutes ago

Находясь на странице проекта, выполним изменение недавно добавленного файла через редактор в веб-интерфейсе.

Нажимаем на файл, выбираем опцию "Open in Web IDE". Произвольно изменяем содержимое файла и нажимаем кнопку "Create commit":

myproject

gitlab-instance-b56601f9/myproject

main

Edit

📄 README.md

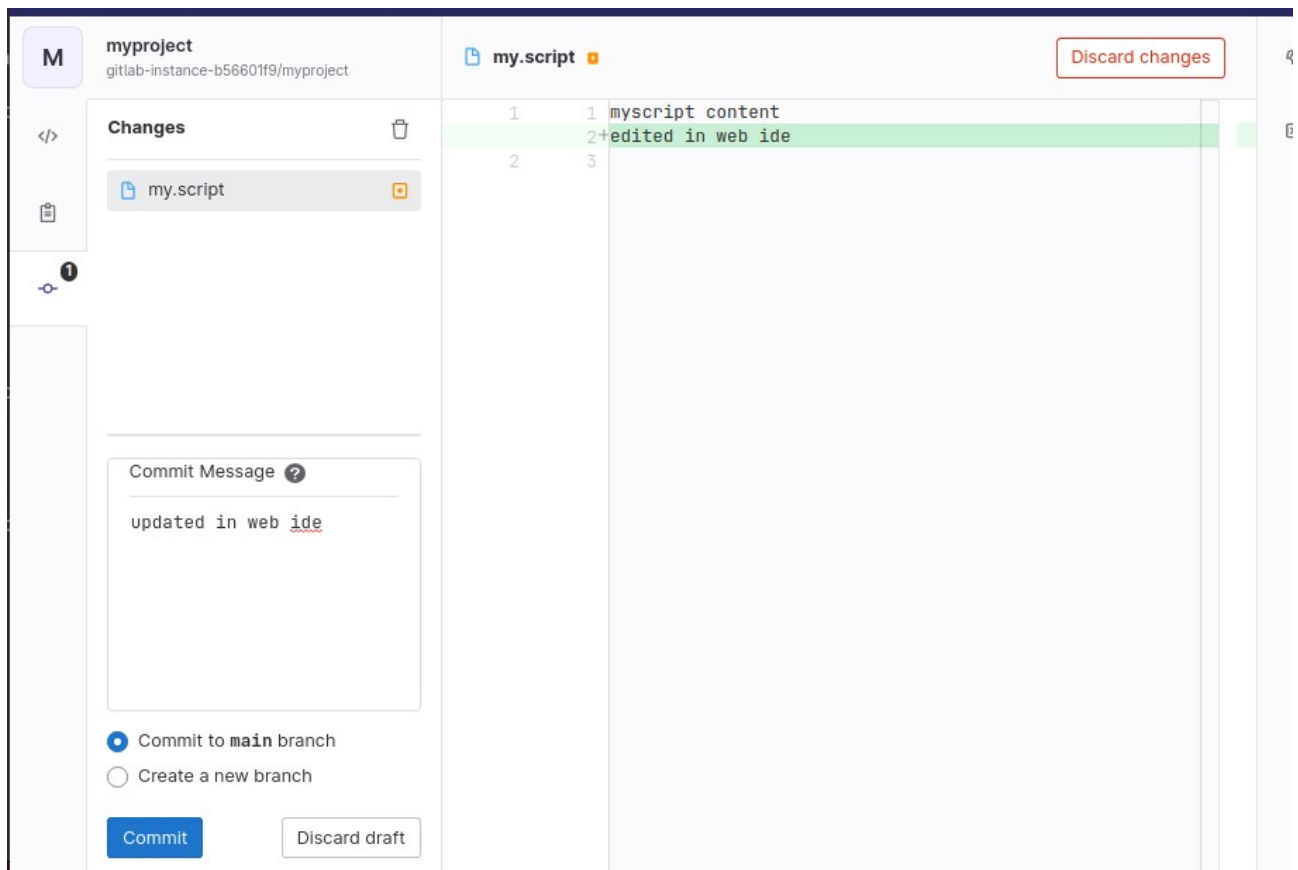
📄 my.script

📄 my.script

1 myscript content

2 edited in web ide

Добавляем комментарий к коммиту, выбираем опцию "commit to main branch" (возможный сценарий в нашей практической работе, в реальной жизни коммиты в мастер запрещены или маловероятны) и нажимаем кнопку "Commit":



Переходим в терминал и выполняем команду переключения на ветку main

```
filipp@filipp-notebook:~/Desktop/myproject$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

Выполняем команду загрузки изменений из удаленного репозитория в нашу локальную копию проекта:

```
$ git pull
```

```
filipp@filipp-notebook:~/Desktop/myproject$ git pull
Username for 'http://172.17.0.2:8929': root
Password for 'http://root@172.17.0.2:8929':
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 542 bytes | 542.00 KiB/s, done.
From http://172.17.0.2:8929/gitlab-instance-b56601f9/myproject
   94c2ae0..0437a3f  main      -> origin/main
Updating 94c2ae0..0437a3f
Fast-forward
 my.script | 2 ++
 1 file changed, 2 insertions(+)
 create mode 100644 my.script
```

Удостоверимся, что мы получили изменения. Вызовем команду просмотра содержимого измененного в web-IDE файла:

```
$ cat my.script
```

```
filipp@filipp-notebook:~/Desktop/myproject$ cat my.script
myscript content
edited in web ide
```

Изменения отображены

--

Изменим снова имеющийся файл:

```
$ nano my.script
```

```
GNU nano 6.2 my.script
edited in terminal

myscript content
edited in web ide
```

Вызовем команду просмотра текущих изменений:

```
$ git status
```

```
filipp@filipp-notebook:~/Desktop/myproject$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   my.script

no changes added to commit (use "git add" and/or "git commit -a")
```

Произведем отмену текущих изменений:

```
$ git restore my.script
$ git status
```

```
filipp@filipp-notebook:~/Desktop/myproject$ git restore my.script
filipp@filipp-notebook:~/Desktop/myproject$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Изменения отменены.

Внесем еще раз изменения в тот же файл:

```
$ nano func.script
```

```
GNU nano 6.2 my.script *
edited in terminal 2
myscript content
edited in web ide
```

Проиндексируем внесенные изменения в локальной версии проекта:

```
$ git add my.script
$ git status
```



```
filipp@filipp-notebook:~/Desktop/myproject$ git add my.script
filipp@filipp-notebook:~/Desktop/myproject$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   my.script
```

Отменим индексацию изменений в локальном проекте:

```
$ git restore --staged my.script
$ git status
```

```
filipp@filipp-notebook:~/Desktop/myproject$ git restore --staged my.script
filipp@filipp-notebook:~/Desktop/myproject$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   my.script

no changes added to commit (use "git add" and/or "git commit -a")
```

Индексация изменений была отменена.

Проиндексируем изменения повторно, подготовим их к отправке:

```
$ git add .
$ git commit -m 'edited in terminal'
$ git status
```

```
filipp@filipp-notebook:~/Desktop/myproject$ git add .
filipp@filipp-notebook:~/Desktop/myproject$ git commit -m 'edited in terminal'
[main 69323b2] edited in terminal
 1 file changed, 2 insertions(+)
filipp@filipp-notebook:~/Desktop/myproject$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

Произведем откат выполненного коммита (воспользуемся хэшем коммита из предыдущего шага):

```
$ git revert 69323b2
```

```
GNU nano 6.2 /home/filipp/Desktop/myproject/.git/COMMIT_EDITMSG
Revert "edited in terminal"

This reverts commit 69323b20a26b9580759564c83d478ae31ff50b1f.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch main
# Your branch is ahead of 'origin/main' by 1 commit.
#   (use "git push" to publish your local commits)
#
# Changes to be committed:
#       modified:   my.script
#
```

Обязательно указываем причину отката коммита, сохраняем изменения:

```
filipp@filipp-notebook:~/Desktop/myproject$ git revert 69323b2
[main 8f0c2da] Revert "edited in terminal"
1 file changed, 2 deletions(-)
```

Вызовем текущий статус изменений в локальном проекте:

```
$ git status
```

```
filipp@filipp-notebook:~/Desktop/myproject$ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

Видим сообщение о том, что ветка `main` в локальном проекте опережает удаленный проект на 2 коммита. Вызовем команду просмотра лога, чтобы визуализировать историю:

```
$ git log --all --graph --decorate --oneline
```

```

filipp@filipp-notebook:~/Desktop/myproject$ git log --all --graph --decorate --oneline
* 8f0c2da (HEAD -> main) Revert "edited in terminal"
* 69323b2 edited in terminal
* 0437a3f (origin/main, origin/HEAD) updated in web ide
* 9818bfb Merge branch 'mybranch' into 'main'
| \
| * 0a200b9 (origin/mybranch, mybranch) my.script added
| /
* 94c2ae0 readme changed
* 90238c0 Initial commit

```

Произведем сброс всех локальных изменений и коммитов, выровняемся с веткой main удаленного репозитория (**все локальные изменения будут утеряны!**)

```
$ git reset -hard origin/main
```

```

filipp@filipp-notebook:~/Desktop/myproject$ git reset --hard origin/main
HEAD is now at 0437a3f updated in web ide

```

Вызовем команду просмотра статуса изменений локального проекта и визуализируем лог изменений:

```

$ git status
$ git log --all --graph --decorate --oneline

```

```

filipp@filipp-notebook:~/Desktop/myproject$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
filipp@filipp-notebook:~/Desktop/myproject$ git log --all --graph --decorate --oneline
* 0437a3f (HEAD -> main, origin/main, origin/HEAD) updated in web ide
* 9818bfb Merge branch 'mybranch' into 'main'
| \
| * 0a200b9 (origin/mybranch, mybranch) my.script added
| /
* 94c2ae0 readme changed
* 90238c0 Initial commit

```

Все локальные изменения и комиты были удалены

--

Выполним создание файла '1.txt' в ветке main, проиндексируем изменения и создадим коммит:

```

$ nano 1.txt
$ git add .
$ git commit -m '1.txt added in main branch'

```

```
filipp@filipp-notebook:~/Desktop/myproject$ nano 1.txt
filipp@filipp-notebook:~/Desktop/myproject$ git add .
filipp@filipp-notebook:~/Desktop/myproject$ git commit -m '1.txt added in main branch'
[main caa48b6] 1.txt added in main branch
1 file changed, 1 insertion(+)
create mode 100644 1.txt
```

Создадим новую ветку и переключимся на нее:

```
$ git checkout -b 'secondary-branch'
```

В созданной ветке создадим файл '2.txt', проиндексируем изменения и создадим коммит:

```
$ nano 2.txt
$ git add .
$ git commit -m '2.txt added in secondary-branch'
```

```
filipp@filipp-notebook:~/Desktop/myproject$ nano 2.txt
filipp@filipp-notebook:~/Desktop/myproject$ git add .
filipp@filipp-notebook:~/Desktop/myproject$ git commit -m '2.txt added in secondary-branch'
[secondary-branch 0b16e1d] 2.txt added in secondary-branch
1 file changed, 1 insertion(+)
create mode 100644 2.txt
```

Переключимся на ветку main и выполним перебазирование:

```
$ git checkout main
```

```
filipp@filipp-notebook:~/Desktop/myproject$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)
```

```
$ git rebase secondary-branch
```

```
filipp@filipp-notebook:~/Desktop/myproject$ git rebase secondary-branch
Successfully rebased and updated refs/heads/main.
```

Убедимся, что оба файла '1.txt' и '2.txt' находятся в ветке main:

```
$ ll
```



```

filipp@filipp-notebook:~/Desktop/myproject$ ll
total 32
drwxrwxr-x  3 filipp filipp 4096 фев  2 16:30 ./
drwxrwxrwx 14 filipp filipp 4096 фев  1 18:19 ../
-rw-rw-r--  1 filipp filipp  14 фев  2 16:13 1.txt
-rw-rw-r--  1 filipp filipp  14 фев  2 16:30 2.txt
drwxrwxr-x  8 filipp filipp 4096 фев  2 16:30 .git/
-rw-rw-r--  1 filipp filipp  35 фев  2 15:48 my.script
-rw-rw-r--  1 filipp filipp 6253 фев  1 18:23 README.md

```

Визуализируем лог изменений:

```
$ git log --all --graph --decorate --oneline
```

```

filipp@filipp-notebook:~/Desktop/myproject$ git log --all --graph --decorate --oneline
* 0b16e1d (HEAD -> main, secondary-branch) 2.txt added in secondary-branch
* caa48b6 1.txt added in main branch
* 0437a3f (origin/main, origin/HEAD) updated in web ide
* 9818bfb Merge branch 'mybranch' into 'main'
| \
| * 0a200b9 (origin/mybranch, mybranch) my.script added
| /
* 94c2ae0 readme changed
* 90238c0 Initial commit
filipp@filipp-notebook:~/Desktop/myproject$ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

```

Перебазирование выполнено успешно, наблюдается разница между предыдущей процедурой слияния git merge (с сохранением веток и истории) и процедурой перебазирования git rebase (линейная история без коммита слияния)

--

Для завершения работы и удаления Gitlab можно выполнить остановку проекта в docker-compose:

```
$ docker-compose down
```

```

filipp@filipp-notebook:~/Desktop/myproject$ docker-compose down
Stopping desktop_web_1 ... done
Removing desktop_web_1 ... done

```

--