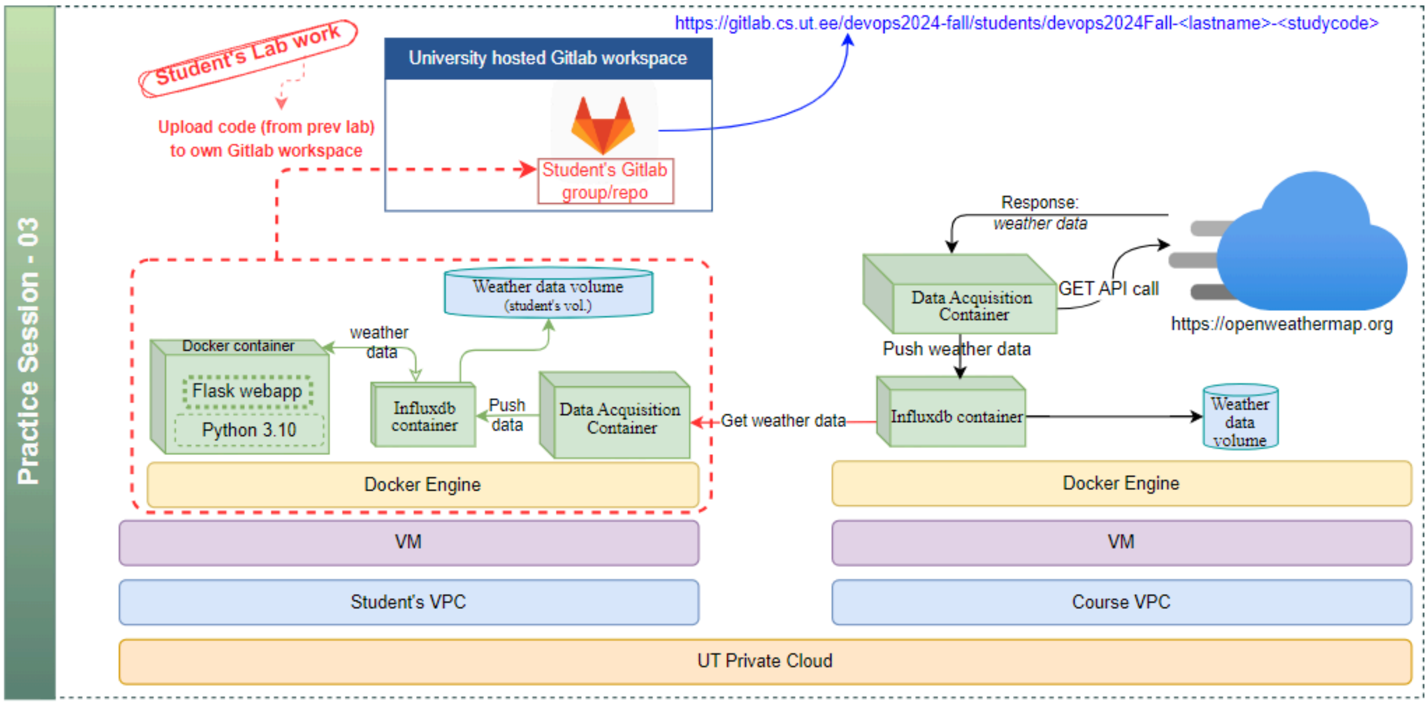
Practice Session 3: Git: Version Control System

Make sure that you have already gone through Lab-02

In this practice session, you will get familiar with the `git` version control system. You will use the Institute's GitLab environment. After this practice session, you should be able to

- create and manage the repository
- create and merge branches
- resolving merge conflicts
- Getting involved with other projects through forking the repo. creating merge requests, etc
- you should be able to perform most of the above tasks through the command line interface

What are we going to do?


Practice Session - 03
Student's Lab work
Upload code (from prev lab) to own Gitlab workspace
University hosted Gitlab workspace
Student's Gitlab group/repo
https://gitlab.cs.ut.ee/devops2024-fall/students/devops2024Fall-<lastname>-<studycode>
Weather data volume (student's vol.)
Docker container
Flask webapp
Python 3.10
weather data
Influxdb container
Push data
Data Acquisition Container
Get weather data
Response: weather data
GET API call
https://openweathermap.org
Data Acquisition Container
Push weather data
Influxdb container
Weather data volume
Docker Engine
Docker Engine
VM
VM
Student's VPC
Course VPC
UT Private Cloud

During this lab session, your tasks will involve familiarising yourself with fundamental Git commands up to Exercise 4. In Exercise 5, you will be required to create a GitLab project for the previous lab materials and subsequently push the corresponding code to the respective project as shown in the above Figure.

Pre-Requisites:

basic bash commands; `ls`, `vim` or `vi`, `cat`, `mkdir`, `cd` etc.

Please Note!!
We are performing this lab on the Linux terminal (bash) of controller VM.

Exercise 1: Creating first repository

Intro: In this exercise you will get familiar with the basic Git commands, including installation and configuration of `git`, creating repositories, cloning repositories etc. You will not use Github rather the GitLab environment provided by the institute.

1.1: Login Institute GitLab environment

In this practice session, we will not use GitHub. We will use the Institute-provided GitLab environment.

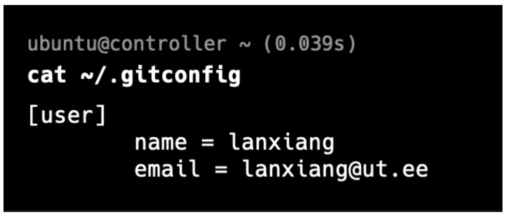
1. Go to https://gitlab.cs.ut.ee/
2. Login using University `username` and `password`

1.2: Connecting to your `controller` VM and check for *git* installation

1. SSH to your `controller` VM
2. Check for installation `git --version`
3. Install git if required.

1.3: Configure Git

1. To review the configuration setting at any time, issue the following command.
   `git config --list`
2. For global settings you may use `--global` option, e.g. `git config --global --list`
3. Configure user information to be used for all the local repositories. Make sure you put your information in quotes " ".
   - `git config --global user.name "Your Name"`
   - `git config --global user.email "your@email.id"`
4. you can find the configuration file at `~/.gitconfig`
   - To see the configuration file enter `cat ~/.gitconfig`



```
ubuntu@controller ~ (0.039s)
cat ~/.gitconfig
[user]
        name = lanxiang
        email = lanxiang@ut.ee
```

1.4: Create the first repo locally

1. Go to your home directory and make a `prac03` folder: `mkdir prac03`
2. Go to `prac03`: `cd prac03`
3. Create a directory with the name `firstrepo` as your repository name.
   - `mkdir firstrepo`
   - `cd firstrepo`

4. Initialize the git repo with
   - `git init` command
   - This will add a `.git` directory with some necessary information
   - You can go inside and check the directory

1.5: Status of the repository

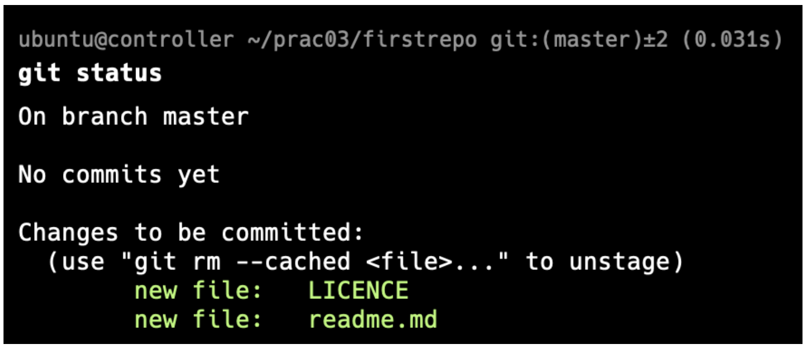
1. This refers to the state of the working directory and the staging area. Enter the command `git status` inside `firstrepo` directory.

```
On branch master
No commits yet
nothing to commit (create/copy files and use "git add" to track)
```

DIY: learn by yourself the meaning of the above lines. You should be able to recall the concepts of branching, commit, and tracking/staging.

1.6: Staging the file

1. What is staging? Staging is an intermediate phase prior to committing a file to the repository with the `git commit` command.
2. Now you are inside `firstrepo` directory.
3. Let's first create two new empty files.
   - `touch LICENCE`
   - `touch readme.md`
4. Now if you enter the `git status` command, you should be able to see that two files are untracked.
5. Add the above files to git tracking system. This is also referred to as *staging the files*.
   - `git add LICENCE`
   - `git add readme.md`
6. Now check the status of your repo using the `git status` command and find the meaning of the output by yourself.
   - `git status`

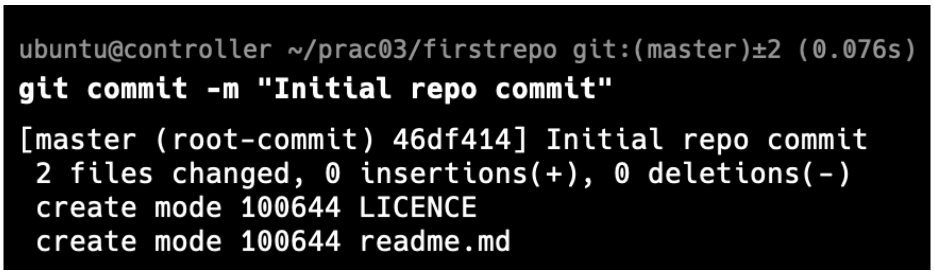


```
ubuntu@controller ~/prac03/firstrepo git:(master)±2 (0.031s)
git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   LICENCE
        new file:   readme.md
```

1.7: Committing the files

1. Now you are inside `firstrepo` directory.
2. Let's make an initial commit and check the status. The `-m` option lets you give a short summary of this commit.

- `git commit -m "Initial repo commit"`



```
ubuntu@controller ~/prac03/firstrepo git:(master)±2 (0.076s)
git commit -m "Initial repo commit"
[master (root-commit) 46df414] Initial repo commit
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 LICENCE
 create mode 100644 readme.md
```

1. Now again check the status with the `git status` command.
2. Modifying the Existing `readme.md` file
   - Open the `readme.md` file add `"Git is cool.."` line: `echo "Git is cool.." >> readme.md`
   - Enter git add command: `git add .`
     Here `.` (dot) at the end of the command represents everything in the current directory. In our case, the command will add only the `readme.md` file.
   - Commit the changes with `git commit` command. e.g. `git commit -m "readme file updated."`
3. Similarly, update the `LICENCE` file with the content available at https://www.apache.org/licenses/LICENSE-2.0.txt and make a commit.
4. To see the history of commits, issue `git log` command.

`DIY: Find the answer: What information can be obtained from the output of git log command?`

1.8: Find the difference between two changes

`git diff` command takes two inputs (e.g. hash of two commits) and reflects the differences between them. Make multiple changes and commits to the readme file.

1. Enter the `git log` command.
2. Choose any two commit hashes, say `<Commit-hash1>` and `<Commit-hash2>`
3. Enter the `git diff` command with both hashes: `git diff <Commit-hash1> <Commit-hash2>`
   `DIY: Find the ans: how to interpret git diff command?`

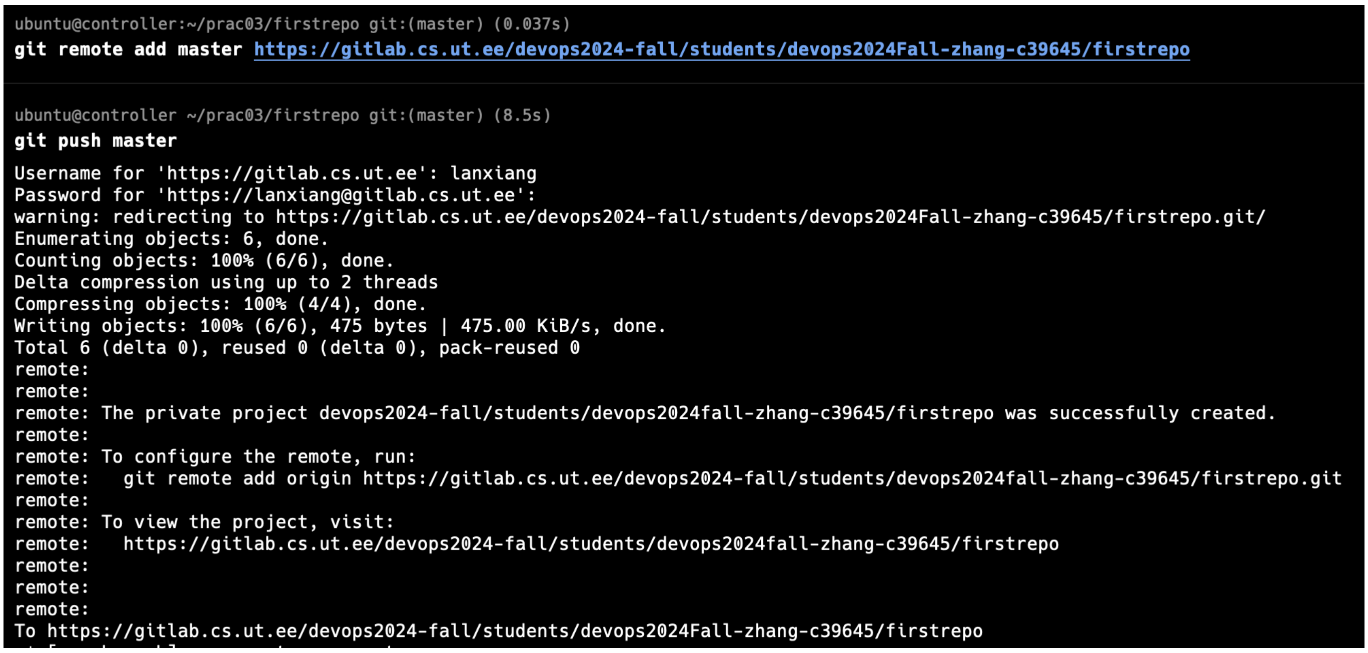
1.9: Pushing the changes to remote repo

So far all the changes are made locally. Now its time to push the code changes to your remote gitlab repo using the `git push` command.

- For this, we have already created the required group and sent you the invitation email. Please check your university email and find the invitation email with the subject “`dehury invited you to join GitLab`”.
- The group name and path should be in the following format:
  - `devops2024-fall/students/devops2024Fall-<lastname>-<studyCode>`
  - e.g. `devops2024-fall/students/devops2024Fall-dehury-xxxxxx`
- You are invited to join the group as a “`Maintainer`”. You are not authorized to access other students’ groups.

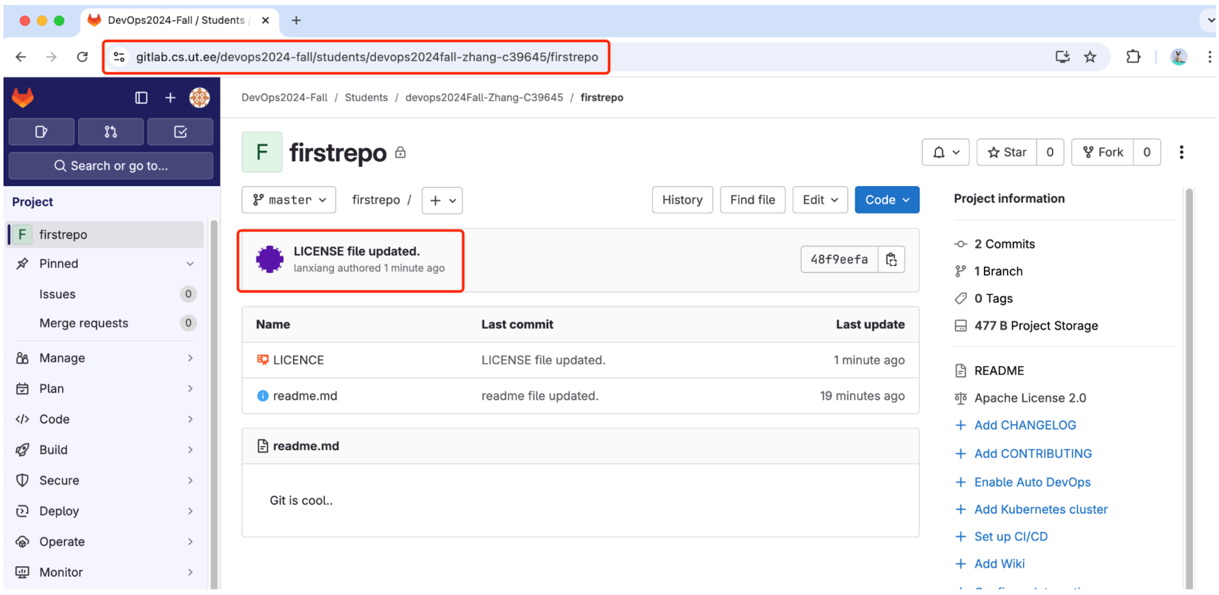
Store your student info in a file using this command: `echo "<your_lastname>-<your_studycode>" > ~/prac03/student_info.txt`

1. push the local repo to your remote gitlab account. You should replace `dehury` with your last name.
   - `git remote add master https://gitlab.cs.ut.ee/devops2024-fall/students/devops2024fall-dehury-xxxxxx/firstrepo`
   - `git push master`
2. The output is similar to below:



```
ubuntu@controller:~/prac03/firstrepo git:(master) (0.037s)
git remote add master https://gitlab.cs.ut.ee/devops2024-fall/students/devops2024Fall-zhang-c39645/firstrepo

ubuntu@controller ~/prac03/firstrepo git:(master) (8.5s)
git push master
Username for 'https://gitlab.cs.ut.ee': lanxiang
Password for 'https://lanxiang@gitlab.cs.ut.ee':
warning: redirecting to https://gitlab.cs.ut.ee/devops2024-fall/students/devops2024Fall-zhang-c39645/firstrepo.git/
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 475 bytes | 475.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote:
remote: The private project devops2024-fall/students/devops2024fall-zhang-c39645/firstrepo was successfully created.
remote:
remote: To configure the remote, run:
remote:   git remote add origin https://gitlab.cs.ut.ee/devops2024-fall/students/devops2024fall-zhang-c39645/firstrepo.git
remote:
remote: To view the project, visit:
remote:   https://gitlab.cs.ut.ee/devops2024-fall/students/devops2024fall-zhang-c39645/firstrepo
remote:
remote:
remote:
To https://gitlab.cs.ut.ee/devops2024-fall/students/devops2024Fall-zhang-c39645/firstrepo
```

3. To verify, go to your remote gitlab account at https://gitlab.cs.ut.ee/ and see if it is present. The repository should be available at `https://gitlab.cs.ut.ee/devops2024-fall/students/devops2024fall-<your lastname>-<your study code>/firstrepo`.
4. The web UI is similar to below:



gitlab.cs.ut.ee/devops2024-fall/students/devops2024fall-zhang-c39645/firstrepo

DevOps2024-Fall / Students / devops2024Fall-Zhang-C39645 / firstrepo

firstrepo

LICENSE file updated.
lanxiang authored 1 minute ago
48f9eefa

Name	Last commit	Last update
LICENCE	LICENSE file updated.	1 minute ago
readme.md	readme file updated.	19 minutes ago

readme.md

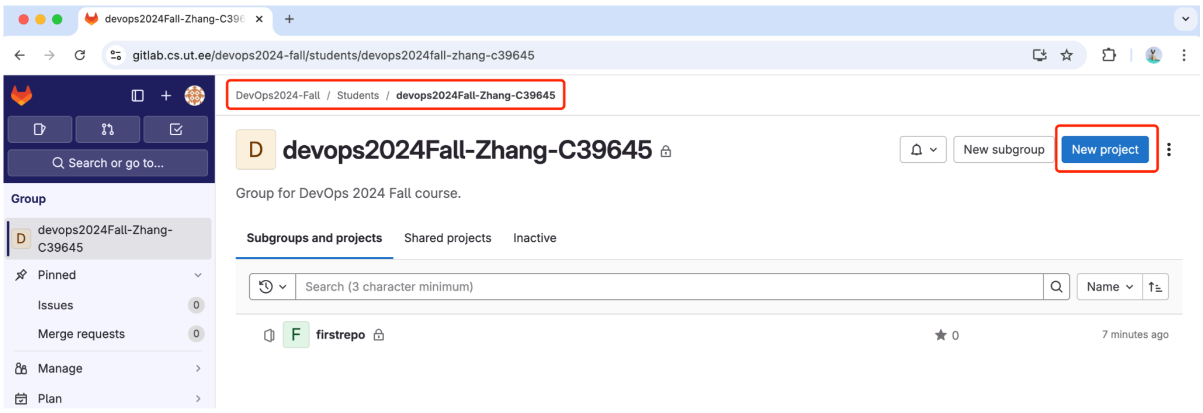
Git is cool..

5. Now in the terminal come outside the `firstrepo` directory with `cd ..` command

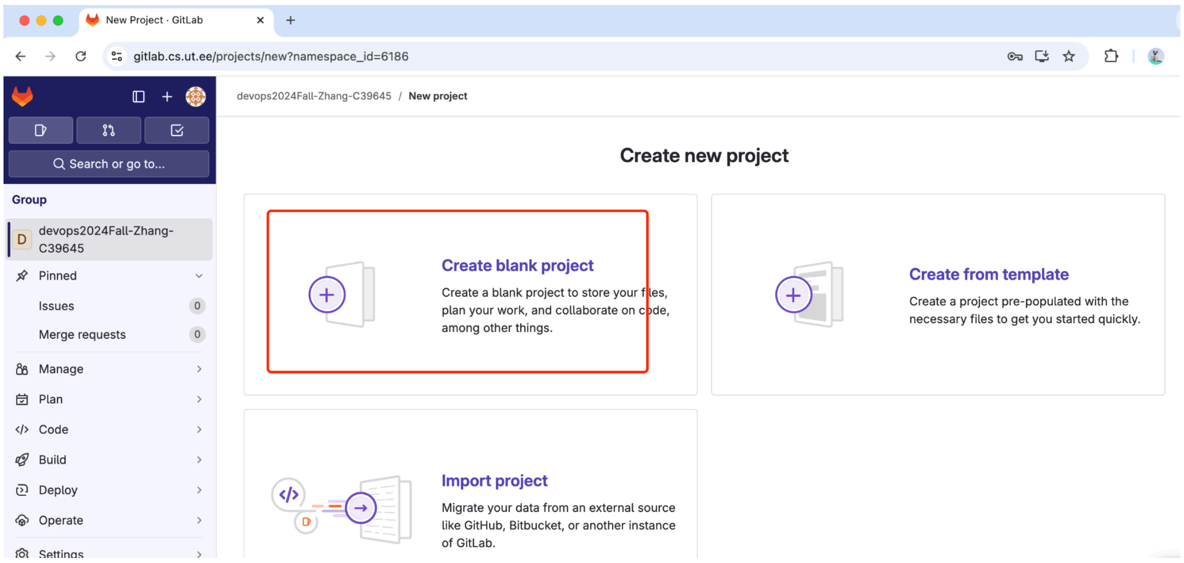
AGS: At this point, Nagios will check if the firstrepo is pushed to gitlab.

1.10: Clone a repo from your remote gitlab account

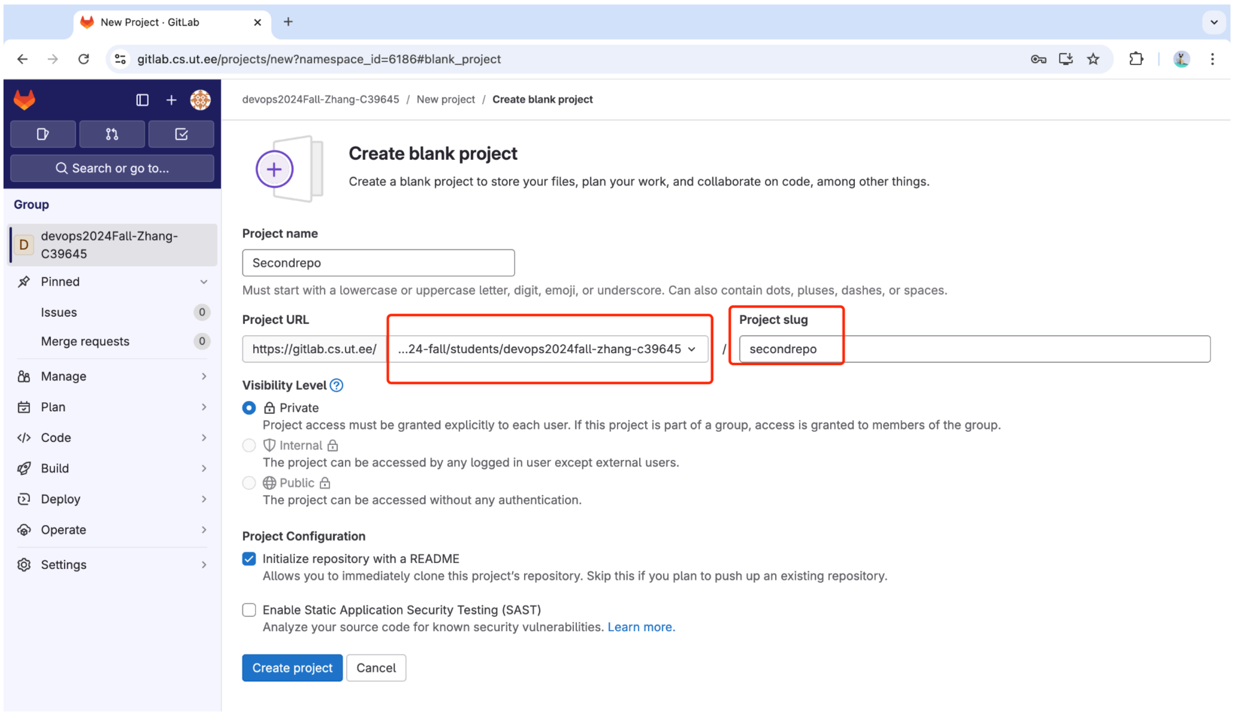
1. login to your gitlab account https://gitlab.cs.ut.ee/
2. Create a new repo inside your group
   - Click on New Project button. Make sure you are inside `https://gitlab.cs.ut.ee/devops2024-fall/students/devops2024fall-<your lastname>-<your study code>` subgroup.


devops2024Fall-Zhang-C39
gitlab.cs.ut.ee/devops2024-fall/students/devops2024fall-zhang-c39645
DevOps2024-Fall / Students / devops2024Fall-Zhang-C39645
Search or go to...
Group
devops2024Fall-Zhang-C39645
Pinned
Issues 0
Merge requests 0
Manage
Plan
devops2024Fall-Zhang-C39645
New subgroup
New project
Group for DevOps 2024 Fall course.
Subgroups and projects
Shared projects
Inactive
Search (3 character minimum)
Name
firstrepo
0
7 minutes ago

   - Select "`Create blank project`"


New Project · GitLab
gitlab.cs.ut.ee/projects/new?namespace_id=6186
devops2024Fall-Zhang-C39645 / New project
Search or go to...
Group
devops2024Fall-Zhang-C39645
Pinned
Issues 0
Merge requests 0
Manage
Plan
Code
Build
Deploy
Operate
Settings
Create new project
Create blank project
Create a blank project to store your files, plan your work, and collaborate on code, among other things.
Create from template
Create a project pre-populated with the necessary files to get you started quickly.
Import project
Migrate your data from an external source like GitHub, Bitbucket, or another instance of GitLab.

   - Enter the project name as `secondrepo` in `Project slug` field and group should be your group name(ex: devops2024fall-zhang-c39645)
   - Leave everything else to its default and click on `Create project`.


New Project · GitLab
gitlab.cs.ut.ee/projects/new?namespace_id=6186#blank_project
Search or go to...
Group
devops2024Fall-Zhang-C39645
Pinned
Issues 0
Merge requests 0
Manage
Plan
Code
Build
Deploy
Operate
Settings
devops2024Fall-Zhang-C39645 / New project / Create blank project

Create blank project

Create a blank project to store your files, plan your work, and collaborate on code, among other things.

Project name: Secondrepo

Must start with a lowercase or uppercase letter, digit, emoji, or underscore. Can also contain dots, pluses, dashes, or spaces.

Project URL: https://gitlab.cs.ut.ee/ ...24-fall/students/devops2024fall-zhang-c39645 / Project slug: secondrepo

Visibility Level
- [✓ Private] Project access must be granted explicitly to each user. If this project is part of a group, access is granted to members of the group.
- Internal: The project can be accessed by any logged in user except external users.
- Public: The project can be accessed without any authentication.

Project Configuration
- [✓] Initialize repository with a README. Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.
- [ ] Enable Static Application Security Testing (SAST). Analyze your source code for known security vulnerabilities. Learn more.

3. Now go to your git terminal and make sure you are **not** inside `firstrepo` directory.
   - enter clone command: `git clone <url of the second repo>`
     - e.g. git clone https://gitlab.cs.ut.ee/devops2024-fall/students/devops2024fall-dehury-xxxxxx/secondrepo.git
     - You should replace `dehury` with your Gitlab username and xxxxxx with your study code.
     - Your terminal output is similar to below:

```
ubuntu@controller ~/prac03 (8.121s)
git clone https://gitlab.cs.ut.ee/devops2024-fall/students/devops2024fall-zhang-c39645/secondrepo.git

Cloning into 'secondrepo'...
Username for 'https://gitlab.cs.ut.ee': lanxiang
Password for 'https://lanxiang@gitlab.cs.ut.ee':
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 (from 0)
Receiving objects: 100% (3/3), done.
```

   - At this point, this may ask you for the university's username and password.
   - Now you should see the `secondrepo` directory available in currect directory. Change the current directory to `secondrepo` directory, using `cd secondrepo` command.
   - Here you can see the only default `README.md` file.

Exercise 2: Branches and Merging

Intro: Branching means you diverge from the main line of development and continue to do work without messing with that main line. The default branch name in Git is master. As you start making commits, you're given a master branch that points to the last commit you made. git init command creates a master branch by default and most people don't bother to change it.

We will use the `firstrepo` repository in this exercise.

Go to the Terminal, `cd firstrepo`. To recap: in the current directory, you have two files: `readme.md` and `LICENSE`.

2.1. Creating a New Branch

Create a new branch called `branch-ex2` using `git branch branch-ex2` command. This creates a new pointer to the same commit you're currently on. It is a good practice to use branches rather than the master branch. This allows you to not mess with the main code.

2.2. Listing branches

`git branch <options>` command allows you to list, create, or delete branches. List all the branches using the `git branch` command. Here you should see following two branches
`$ git branch`

```
  branch-ex2
* master
```

The * indicates the current branch.

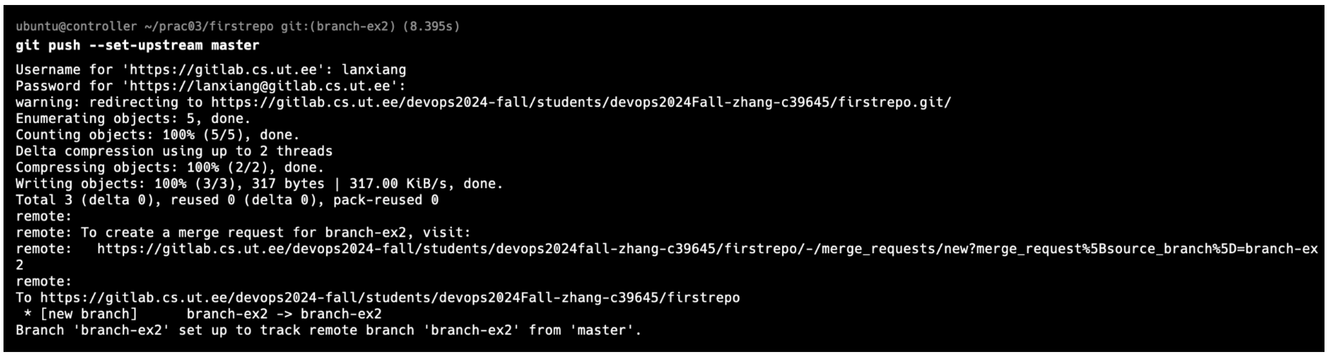
2.3. Change the current branch

1. `git checkout <branch name>` is used to switch to another branch. Let's switch to the newly created branch:
   - `git checkout branch-ex2`
2. Now enter git branch command to see the current active branch. The output changes to following:
   `$ git branch`

```
* branch-ex2
  master
```

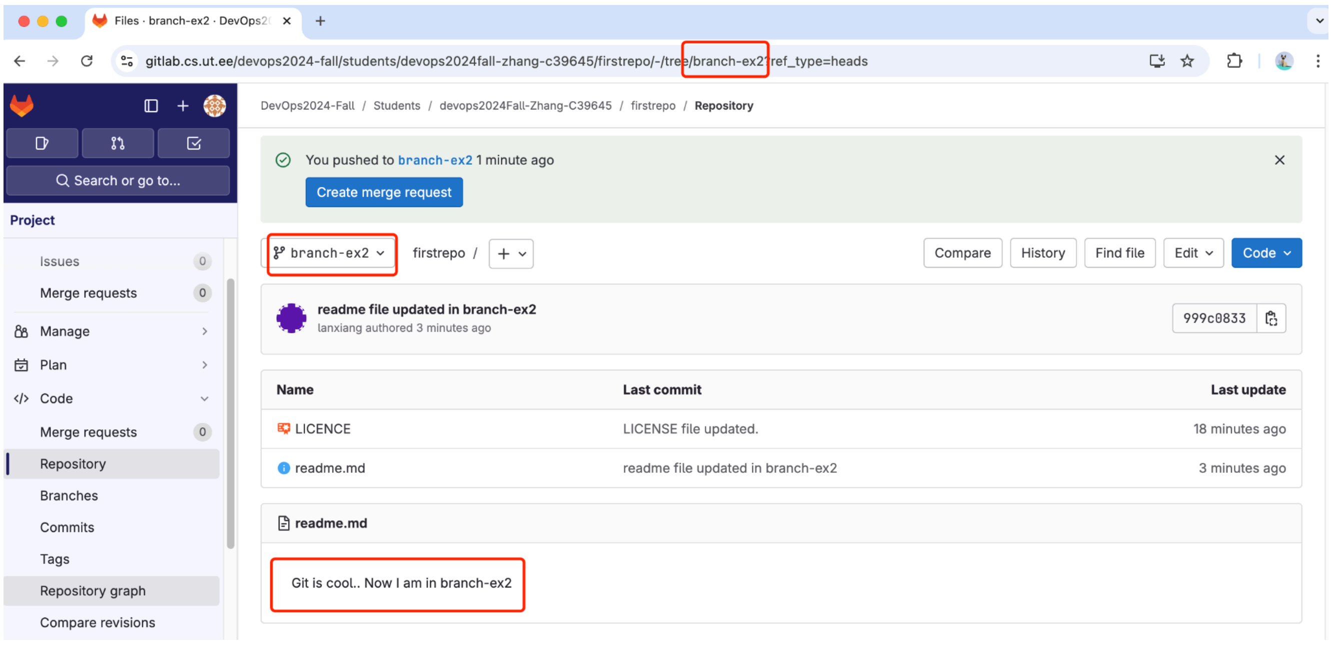
2.4. Modify new branch content

1. Now let's append a new line to the `readme.md` file.
   - `echo "Now I am in branch-ex2" >> readme.md`
2. Stage the `readme.md` changes : `git add readme.md`
3. Commit the changes on this branch: `git commit -m "readme file updated in branch-ex2"`
4. Check the repository status using the `git log` command. Here, you should see that the commits are on branch `branch-ex2`.
5. Now check the content of the `readme.md` file. Here you should see the line saying `Now I am in branch-ex2`.
6. Using `git push --set-upstream master` command push all the changes of this branch to your remote gitlab account. Sample output given below:



```
ubuntu@controller ~/prac03/firstrepo git:(branch-ex2) (8.395s)
git push --set-upstream master

Username for 'https://gitlab.cs.ut.ee': lanxiang
Password for 'https://lanxiang@gitlab.cs.ut.ee':
warning: redirecting to https://gitlab.cs.ut.ee/devops2024-fall/students/devops2024Fall-zhang-c39645/firstrepo.git/
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 317 bytes | 317.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: To create a merge request for branch-ex2, visit:
remote:   https://gitlab.cs.ut.ee/devops2024-fall/students/devops2024fall-zhang-c39645/firstrepo/-/merge_requests/new?merge_request%5Bsource_branch%5D=branch-ex
2
remote:
To https://gitlab.cs.ut.ee/devops2024-fall/students/devops2024Fall-zhang-c39645/firstrepo
 * [new branch]      branch-ex2 -> branch-ex2
Branch 'branch-ex2' set up to track remote branch 'branch-ex2' from 'master'.
```

7. Using your browser, go to your remote Gitlab repo and check the update status. Sample output given below:



Files · branch-ex2 · DevOps2

gitlab.cs.ut.ee/devops2024-fall/students/devops2024fall-zhang-c39645/firstrepo/-/tree/branch-ex2?ref_type=heads

DevOps2024-Fall / Students / devops2024Fall-Zhang-C39645 / firstrepo / **Repository**

You pushed to branch-ex2 1 minute ago

Create merge request

branch-ex2 firstrepo /

readme file updated in branch-ex2
lanxiang authored 3 minutes ago
999c0833

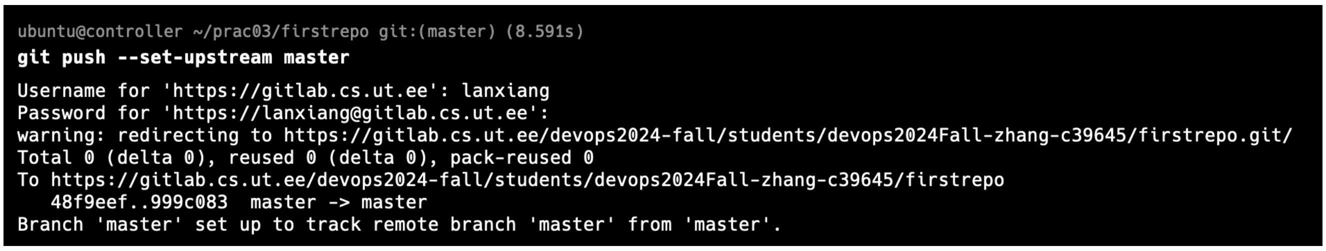
Name	Last commit	Last update
LICENCE	LICENSE file updated.	18 minutes ago
readme.md	readme file updated in branch-ex2	3 minutes ago

readme.md

Git is cool.. Now I am in branch-ex2

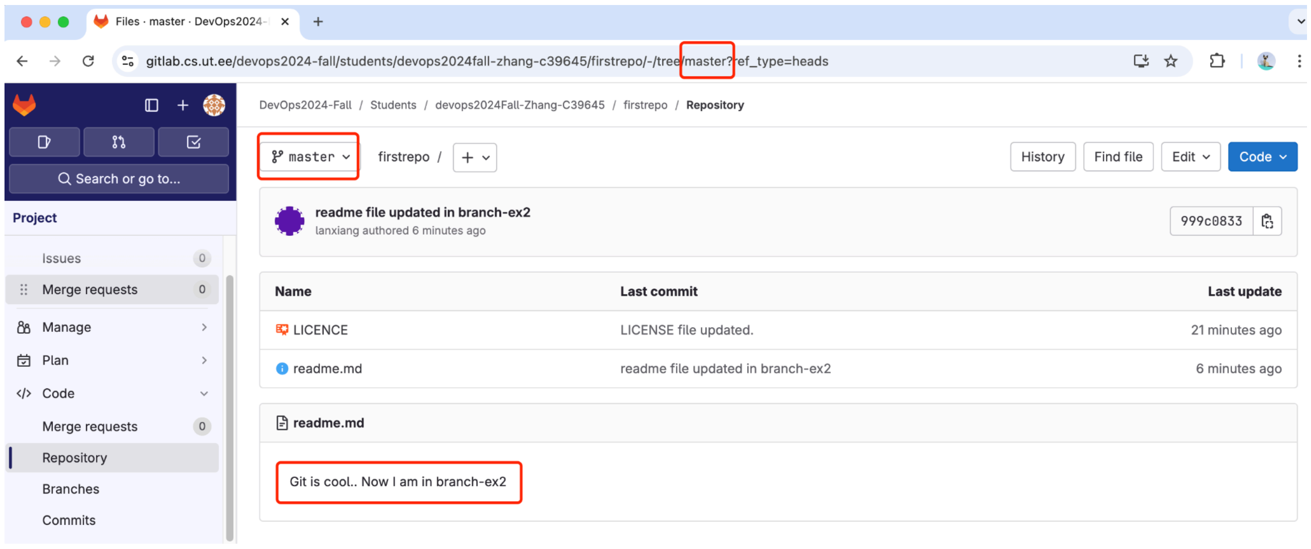
2.5. Merge the content of multiple branches

1. Here we will merge the content of the `readme.md` file from `branch-ex2` to master branch using the `git merge <source branch-name>` command.
2. Lets first checkout the master branch: `git checkout master`
3. We are now in the `master` branch. Lets first verify if the line `Now I am in branch-ex2` is present using `cat readme.md` command. As expected, that line should not be present in this branch.
4. Now issue `git merge` command: `git merge branch-ex2 -m "merging readme file to master"`
5. From the terminal push all the changes of this branch to your remote gitlab account. Sample output given below:



```
ubuntu@controller ~/prac03/firstrepo git:(master) (8.591s)
git push --set-upstream master

Username for 'https://gitlab.cs.ut.ee': lanxiang
Password for 'https://lanxiang@gitlab.cs.ut.ee':
warning: redirecting to https://gitlab.cs.ut.ee/devops2024-fall/students/devops2024Fall-zhang-c39645/firstrepo.git/
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://gitlab.cs.ut.ee/devops2024-fall/students/devops2024Fall-zhang-c39645/firstrepo
   48f9eef..999c083  master -> master
Branch 'master' set up to track remote branch 'master' from 'master'.
```

6. Using your browser, go to your remote Gitlab repo and check the update status. Sample output given below:


Files · master · DevOps2024-
gitlab.cs.ut.ee/devops2024-fall/students/devops2024fall-zhang-c39645/firstrepo/-/tree/master?ref_type=heads
Search or go to...
Project
Issues 0
Merge requests 0
Manage
Plan
Code
Merge requests 0
Repository
Branches
Commits
DevOps2024-Fall / Students / devops2024Fall-Zhang-C39645 / firstrepo / Repository
master
firstrepo /
History
Find file
Edit
Code

readme file updated in branch-ex2
lanxiang authored 6 minutes ago
999c0833

Name	Last commit	Last update
LICENCE	LICENSE file updated.	21 minutes ago
readme.md	readme file updated in branch-ex2	6 minutes ago

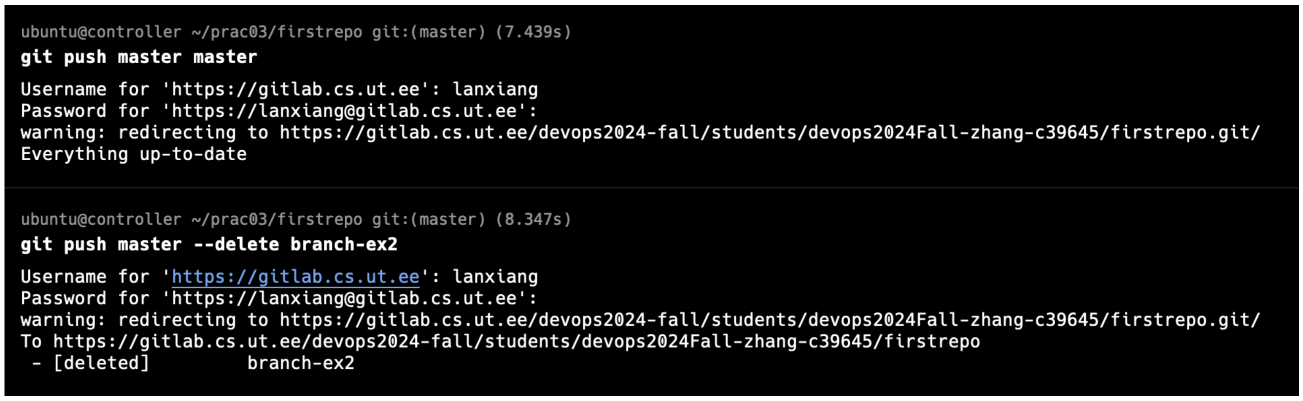
readme.md

Git is cool.. Now I am in branch-ex2

7. Now check the content of the `readme.md` file using `cat readme.md`. The changes from the `branch-ex2` branch should be available in the current `master` branch.

2.6. Deleting a branch

1. Make sure that `branch-ex2` and `master` is available using the `git branch` command.
2. Checkout the `master` branch: `git checkout master`
3. Delete the `branch-ex2` branch using following command:
   - `git branch -d branch-ex2`
4. Push all the changes to your remote gitlab account.



```
ubuntu@controller ~/prac03/firstrepo git:(master) (7.439s)
git push master master
Username for 'https://gitlab.cs.ut.ee': lanxiang
Password for 'https://lanxiang@gitlab.cs.ut.ee':
warning: redirecting to https://gitlab.cs.ut.ee/devops2024-fall/students/devops2024Fall-zhang-c39645/firstrepo.git/
Everything up-to-date

ubuntu@controller ~/prac03/firstrepo git:(master) (8.347s)
git push master --delete branch-ex2
Username for 'https://gitlab.cs.ut.ee': lanxiang
Password for 'https://lanxiang@gitlab.cs.ut.ee':
warning: redirecting to https://gitlab.cs.ut.ee/devops2024-fall/students/devops2024Fall-zhang-c39645/firstrepo.git/
To https://gitlab.cs.ut.ee/devops2024-fall/students/devops2024Fall-zhang-c39645/firstrepo
 - [deleted]         branch-ex2
```

AGS: At this point, Nagios will check if the branch-ex2 is merged and deleted.

5. Question: what will happen if you are checked out at `branch-ex2` while deleting `branch-ex2`?
   You will get the error similar to below:
   `$ git branch -d branch-ex2`

```
error: Cannot delete branch 'branch-ex2' checked out at 'D:/firstrepo'
```

6. Question: what will happen, if the `branch-ex2` is not fully merged with the master branch?
   You will get following error:
   `$ git branch -d master`

```
error: The branch 'master' is not fully merged.
If you are sure you want to delete it, run 'git branch -D master'.
```

Exercise 3: Handling Merge Conflicts

Conflict arises when you may have made overlapping changes to a file, and Git cannot automatically merge the changes. In this exercise we will see how to handle the conflicts.

3.1. Update the readme file with some extra lines.

Lets append some new lines to the `readme.md` file. Make sure that you are inside `firstrepo` in your git terminal.

1. Append two new lines using following commands:
   - `echo "This line is added in master branch." >> readme.md`
   - `echo "This is just an extra line inserted while in master branch." >> readme.md`
2. Stage the file: `git add .`
3. Commit the staged content: `git commit -m "added some extra lines to readme file"`
4. At this point the content of `readme.md` file should be:

```
    Git is cool..
    Now I am in branch-ex2
    This line is added in master branch.
    This is just an extra line inserted while in master branch.
```

3.2. Create and checkout new branch

1. Lets first create a new branch called `branch-ex3` using the `git branch branch-ex3` command.
2. Switch to or checkout the new branch: `git checkout branch-ex3`
3. Open `readme.md` file and update the third line:
   From: This line is added in master branch.
   To: This line is MODIFIED in BRANCH-EX3 branch.
4. Stage and commit the changes:
   - `git add .`
   - `git commit -m "readme file modified in branch-ex3"`

5. Here, the content of the readme.md file should be:

```
      Git is cool..
      Now I am in branch-ex2
      This line is MODIFIED in BRANCH-EX3 branch.
      This is just an extra line inserted while in master branch.
```

3.3. Checkout master branch and modify the `readme.md` file

Now lets again modify the same file in `master` branch:

1. First checkout `master` branch: `git checkout master`
2. Verify the content of the `readme.md` file. The content should be as below:

```
Git is cool..
Now I am in branch-ex2
This line is added in master branch.
This is just an extra line inserted while in master branch.
```

3. Lets update again the third line of `readme.md` file.
   From: This line is added in master branch.
   To: This line is RE-MODIFIED in MASTER branch.
4. Stage and commit the changes
   a. `git add .`
   b. `git commit -m "readme file re-modified in master"`
5. Now the content of `readme.md` file should look like:

```
Git is cool..
Now I am in branch-ex2
This line is RE-MODIFIED in MASTER branch.
This is just an extra line inserted while in master branch.
```

3.4. Merge to master branch

Now at this point we will merge the `branch-ex3` branch to `master` branch.

1. Checkout `master` branch: `git checkout master`
2. Merge the `branch-ex3` branch using `git merge branch-ex3` command.
3. Here you will get following similar error:

```
Auto-merging readme.md
CONFLICT (content): Merge conflict in readme.md
Automatic merge failed; fix conflicts and then commit the result.
```

4. If you now see the content of `readme.md` file, this should look like:
   `$ cat readme.md`

```
Git is cool..
Now I am in branch-ex2
<<<<<<< HEAD
This line is RE-MODIFIED in MASTER branch.
=======
This line is MODIFIED in BRANCH-EX3 branch.
>>>>>>> branch-ex3
```

```
This is just an extra line inserted while in master branch.
```

5. The `readme.md` file now contains information to help you find the conflict. The line between `<<<<<<< HEAD` and `=======` represents the line from the `master` branch and the line between `=======` and `>>>>>>> branch-ex3` represents the line from `branch-ex3` branch.
6. Lets remove the line from the `master` branch and keep the line from `branch-ex3` branch. For this remove following lines from the `readme.md` file:

```
<<<<<<< HEAD
This line is RE-MODIFIED in MASTER branch.
=======
```

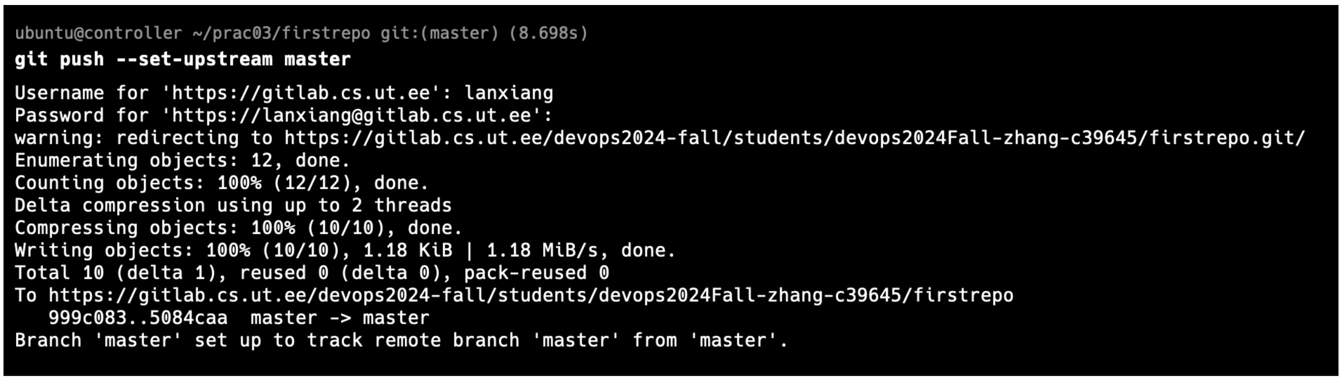
   and

```
>>>>>>> branch-ex3
```

7. Now the `readme.md` file should look like:

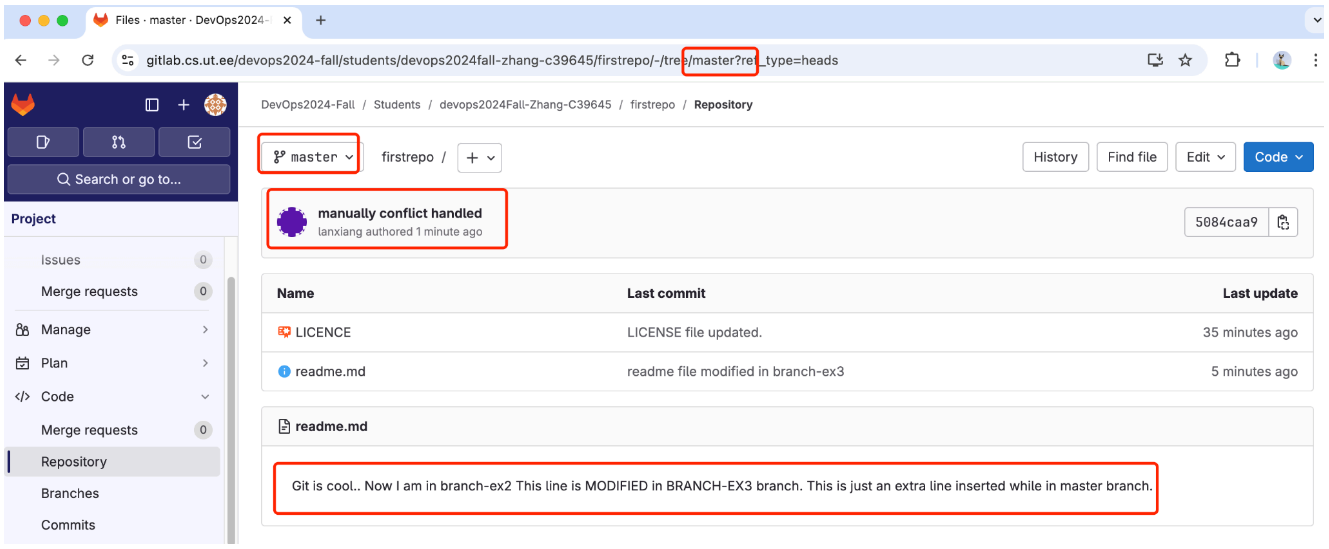
```
Git is cool..
Now I am in branch-ex2
This line is MODIFIED in BRANCH-EX3 branch.
This is just an extra line inserted while in master branch.
```

8. Now Stage and commit the changes
   a. `git add .`
   b. `git commit -m "manually conflict handled"`
9. From the terminal push all the changes of this branch to your remote gitlab account. Sample output given below:



```
ubuntu@controller ~/prac03/firstrepo git:(master) (8.698s)
git push --set-upstream master
Username for 'https://gitlab.cs.ut.ee': lanxiang
Password for 'https://lanxiang@gitlab.cs.ut.ee':
warning: redirecting to https://gitlab.cs.ut.ee/devops2024-fall/students/devops2024Fall-zhang-c39645/firstrepo.git/
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 2 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 1.18 KiB | 1.18 MiB/s, done.
Total 10 (delta 1), reused 0 (delta 0), pack-reused 0
To https://gitlab.cs.ut.ee/devops2024-fall/students/devops2024Fall-zhang-c39645/firstrepo
   999c083..5084caa  master -> master
Branch 'master' set up to track remote branch 'master' from 'master'.
```

10. Using your browser, go to your remote Gitlab repo and check the update status. Sample output given below:


gitlab.cs.ut.ee/devops2024-fall/students/devops2024fall-zhang-c39645/firstrepo/-/tree/master?ref_type=heads

DevOps2024-Fall / Students / devops2024Fall-Zhang-C39645 / firstrepo / Repository

master firstrepo /

manually conflict handled
lanxiang authored 1 minute ago 5084caa9

Name	Last commit	Last update
LICENCE	LICENSE file updated.	35 minutes ago
readme.md	readme file modified in branch-ex3	5 minutes ago

readme.md

Git is cool.. Now I am in branch-ex2 This line is MODIFIED in BRANCH-EX3 branch. This is just an extra line inserted while in master branch.

AGS: At this point, Nagios will check if the branch-ex3 is merged into and conflicts are resolved.

Exercise 4: Forking and merging a branch

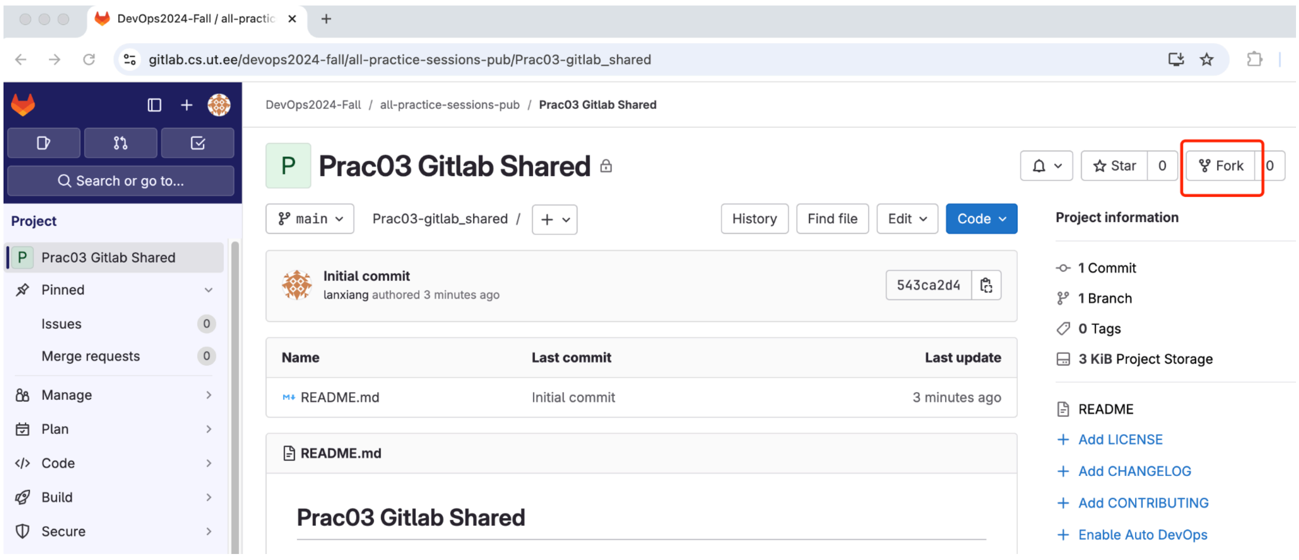
Intro: A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

In this exercise you are going to complete the following tasks:

- Forking the main repository (https://gitlab.cs.ut.ee/devops2024-fall/all-practice-sessions/prac03-gitlab_shared) to your gitlab account.
- You will clone the forked repository and create a branch, modify it by adding your files and merge it with your forked repository.
- Finally, you will send merge request to owner of the main repository (https://gitlab.cs.ut.ee/devops2024-fall/all-practice-sessions/prac03-gitlab_shared)

4.1. Forking a project

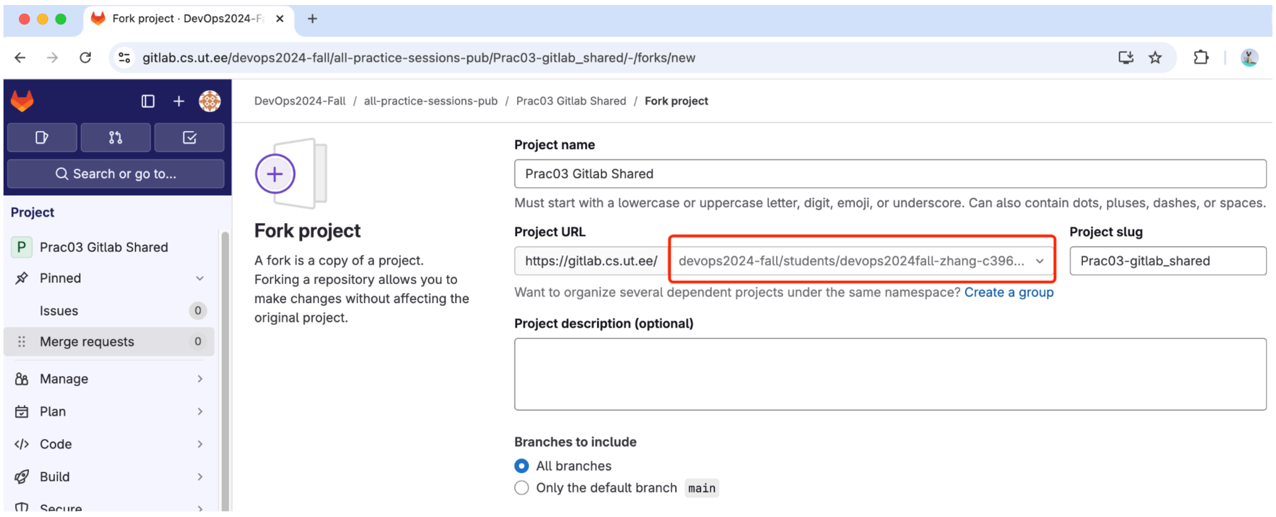
- Login to your remote GitLab account: https://gitlab.cs.ut.ee
- Go to https://gitlab.cs.ut.ee/devops2024-fall/all-practice-sessions/prac03-gitlab_shared
- Fork this repository (GUI)



Name	Last commit	Last update
README.md	Initial commit	3 minutes ago

README.md

Prac03 Gitlab Shared


Fork project · DevOps2024-F
gitlab.cs.ut.ee/devops2024-fall/all-practice-sessions-pub/Prac03-gitlab_shared/-/forks/new
DevOps2024-Fall / all-practice-sessions-pub / Prac03 Gitlab Shared / Fork project
Project
Prac03 Gitlab Shared
Pinned
Issues 0
Merge requests 0
Manage
Plan
Code
Build
Secure

Fork project

A fork is a copy of a project. Forking a repository allows you to make changes without affecting the original project.

Project name: Prac03 Gitlab Shared

Must start with a lowercase or uppercase letter, digit, emoji, or underscore. Can also contain dots, pluses, dashes, or spaces.

Project URL: https://gitlab.cs.ut.ee/ devops2024-fall/students/devops2024fall-zhang-c396...

Project slug: Prac03-gitlab_shared

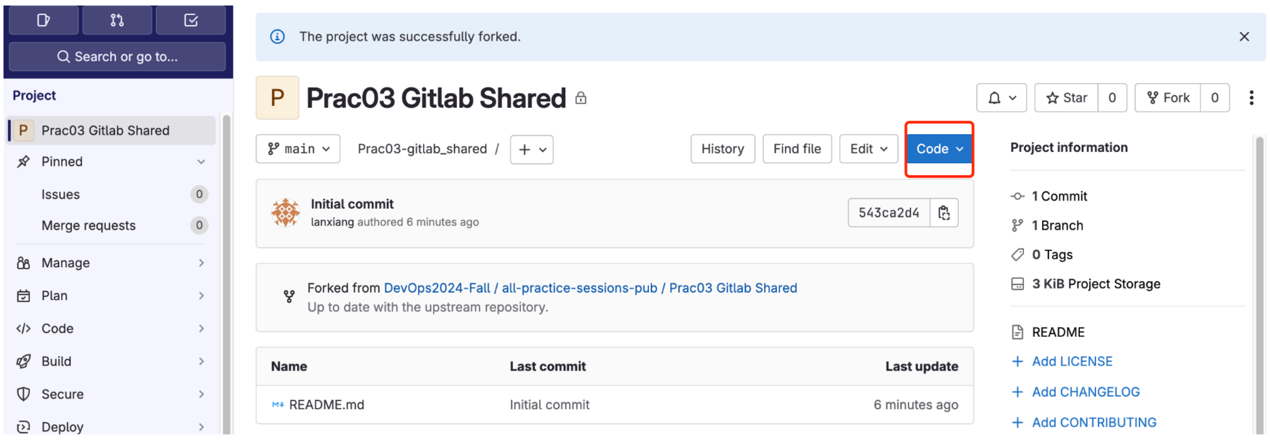
Want to organize several dependent projects under the same namespace? Create a group

Project description (optional)

Branches to include
[✓ All branches]
Only the default branch main

4.2. Clone your forked project, create and add your files and merging

- Clone that repository to your local `prac03` folder using the `git clone <url>` command. The url repository for cloning can be found here



The project was successfully forked.

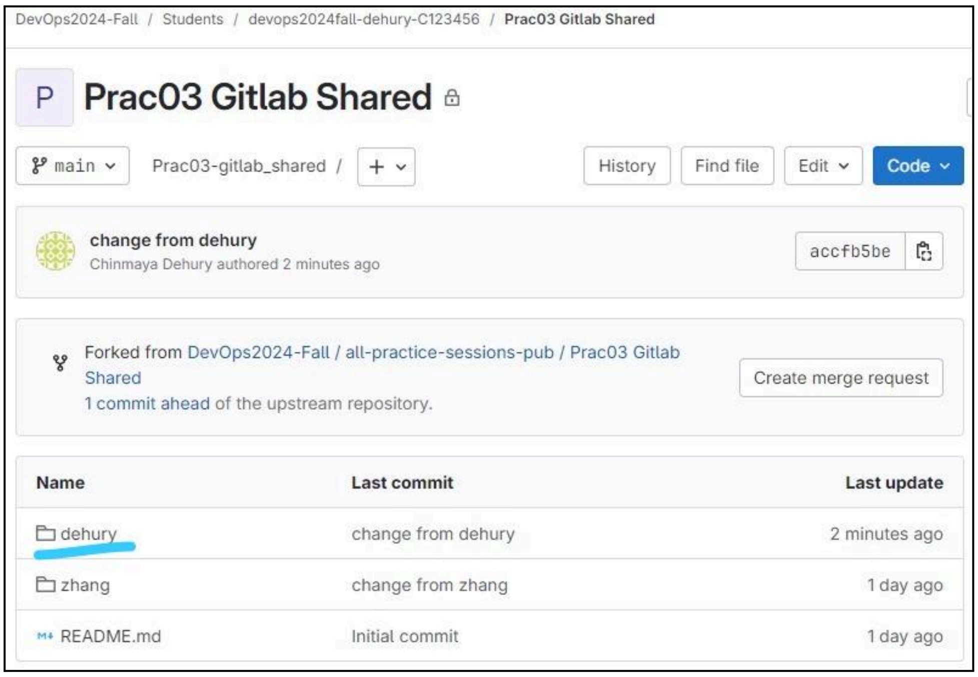
Initial commit
lanxiang authored 6 minutes ago 543ca2d4

Forked from DevOps2024-Fall / all-practice-sessions-pub / Prac03 Gitlab Shared
Up to date with the upstream repository.

Name	Last commit	Last update
README.md	Initial commit	6 minutes ago

- Make a new branch `ex4-dehury` using `git branch ex4-dehury`. You should replace `dehury` with your last name.

- Checkout the newly created `ex4-dehury` branch using `git checkout ex4-dehury`. Replace `dehury` with your last name.
- Create a directory: `mkdir dehury && cd dehury`. Replace `dehury` with your last name.
- Create a file `hello.txt` inside the `dehury` directory using the following command.
  `echo "Helloooo, Its <your_last_name> speaking from DevOps course." > hello.txt`
- Change to parent directory `cd ..`
- Stage parent directory `git add .`
- Commit the changes locally using `git commit -m "change from <your__last_name>"`
- See the list of existing remotes `git remote`
- Add a remote with the name `main` using the command `git remote add main` https://gitlab.cs.ut.ee/devops2024-fall/students/devops2024fall-xxxxxx-xxxxxx/Prac03-gitlab_shared. Replace `devops2024fall-xxxxxx-xxxxxx` with your group name
- Push the changes `git push main`
- Check out `git checkout main` and Merge the changes to your repository `git merge ex4-dehury`. Replace `dehury` with your name.
- Push the final changes `git push main`
- After this you should see the following changes in your repo


DevOps2024-Fall / Students / devops2024fall-dehury-C123456 / Prac03 Gitlab Shared

Prac03 Gitlab Shared

main ⌄ Prac03-gitlab_shared / + ⌄ History Find file Edit ⌄ Code ⌄

change from dehury
Chinmaya Dehury authored 2 minutes ago
accfb5be

Forked from DevOps2024-Fall / all-practice-sessions-pub / Prac03 Gitlab Shared
1 commit ahead of the upstream repository.
Create merge request

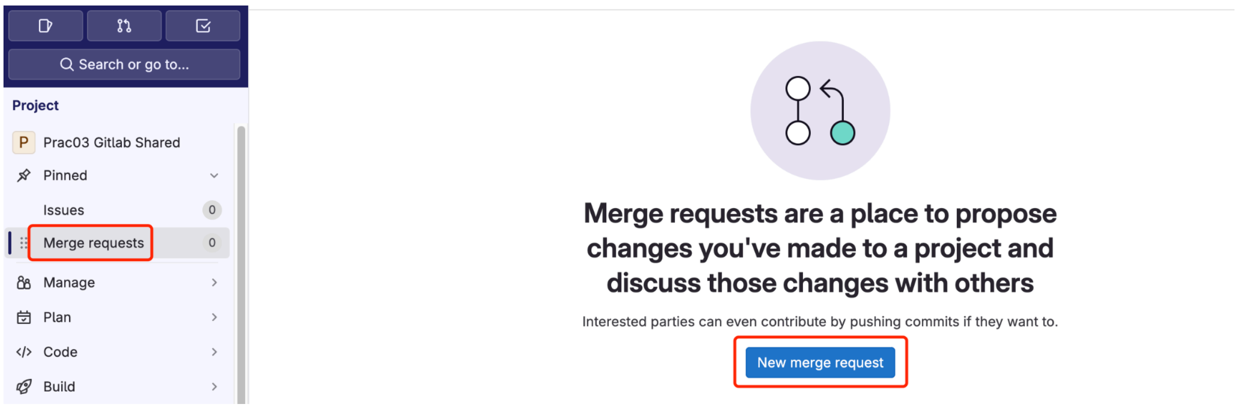
Name	Last commit	Last update
dehury	change from dehury	2 minutes ago
zhang	change from zhang	1 day ago
README.md	Initial commit	1 day ago

AGS: At this point, Nagios will check if the ex4-<student_lastname> branch exists and the hello.txt file has the expected content.

4.3. Sending merge request to owner of the main repository

Here, you can send a merge request in two ways: 1)Gitlab GUI 2) git cmd line interface

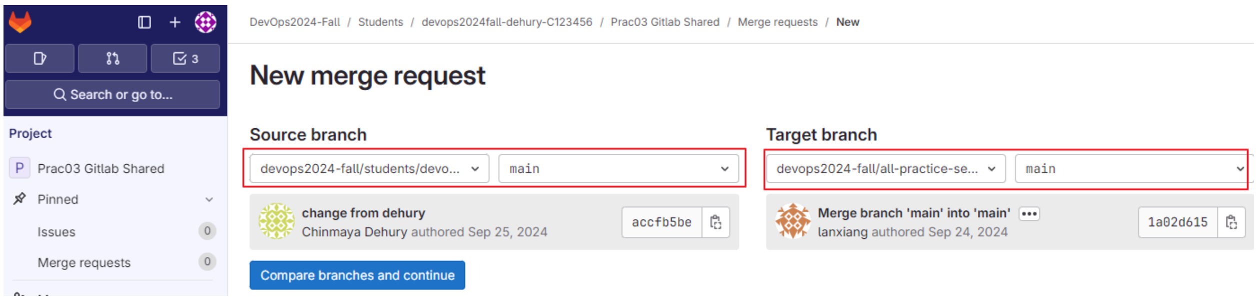
- Create a merge request using GUI, for this go to your gitlab project and create as shown below:



Merge requests are a place to propose changes you've made to a project and discuss those changes with others

Interested parties can even contribute by pushing commits if they want to.

- Make sure, you added the correct projects as shown below


Search or go to...
Project
Prac03 Gitlab Shared
Pinned
Issues 0
Merge requests 0
DevOps2024-Fall / Students / devops2024fall-dehury-C123456 / Prac03 Gitlab Shared / Merge requests / New
Compare branches and continue

New merge request

Source branch: devops2024-fall/students/devo... | main

change from dehury
Chinmaya Dehury authored Sep 25, 2024 accfb5be

Target branch: devops2024-fall/all-practice-se... | main

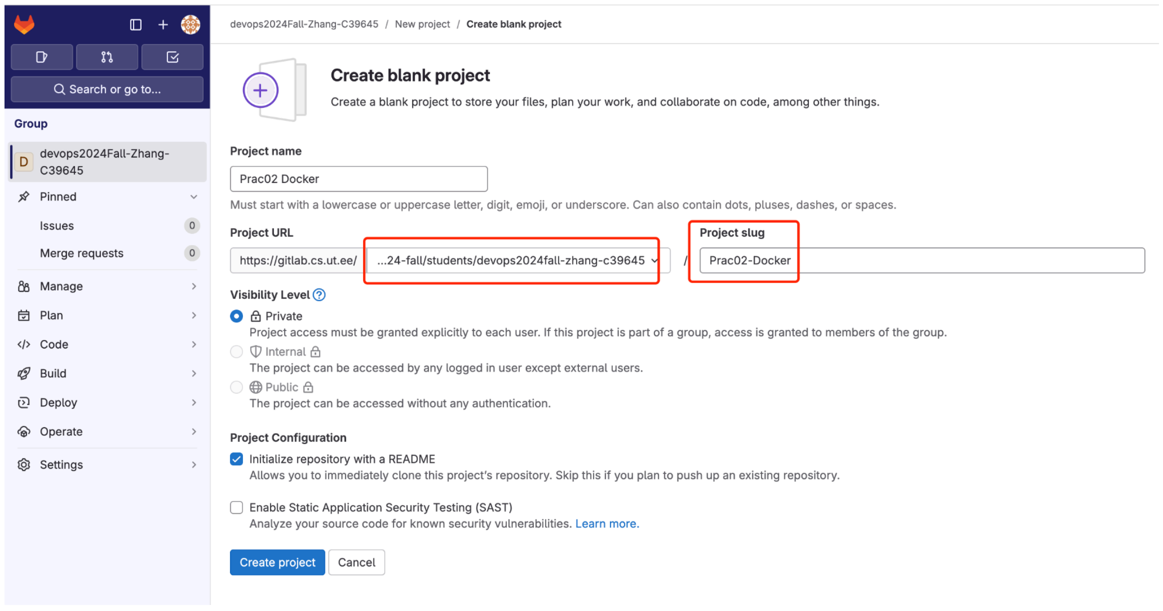
Merge branch 'main' into 'main'
lanxiang authored Sep 24, 2024 1a02d615

- Click on `Compare branches and continue option`.
- Leave everything to its default and click on "`Create merge request`" option.
- If everything goes well, PI and TAs will get a merge request notification.

Exercise 5: Uploading Lab02 content

In this exercise you will create a project and upload the content/materials of Lab02.

- Login to your remote GitLab account: https://gitlab.cs.ut.ee
- Create a project in your group with the name "`Prac02-Docker`"


devops2024Fall-Zhang-C39645 / New project / Create blank project

Create blank project

Create a blank project to store your files, plan your work, and collaborate on code, among other things.

Project name: Prac02 Docker

Must start with a lowercase or uppercase letter, digit, emoji, or underscore. Can also contain dots, pluses, dashes, or spaces.

Project URL: https://gitlab.cs.ut.ee/ ...24-fall/students/devops2024fall-zhang-c39645 / Project slug: Prac02-Docker

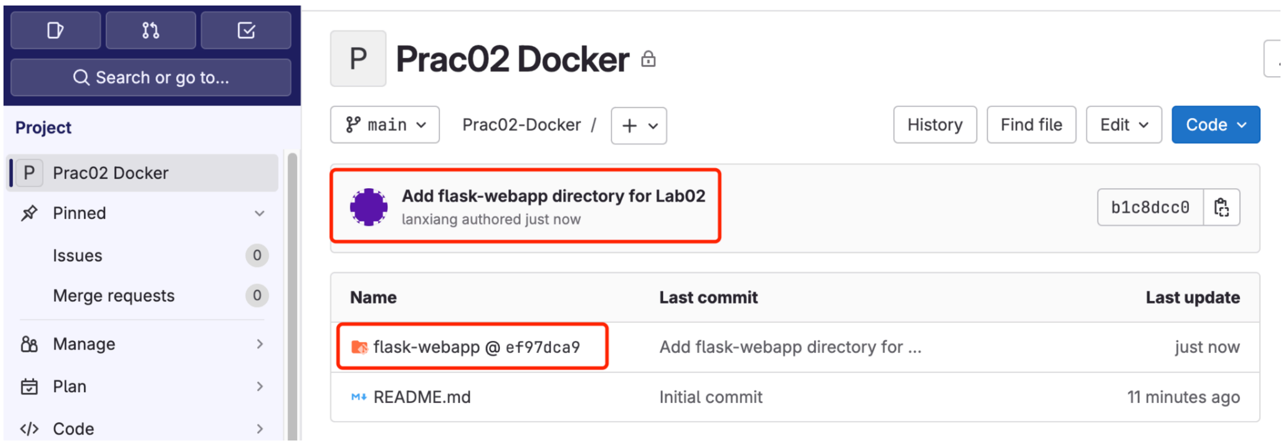
Visibility Level
- [✓ Private] Project access must be granted explicitly to each user. If this project is part of a group, access is granted to members of the group.
- Internal: The project can be accessed by any logged in user except external users.
- Public: The project can be accessed without any authentication.

Project Configuration
- [✓] Initialize repository with a README — Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.
- [ ] Enable Static Application Security Testing (SAST) — Analyze your source code for known security vulnerabilities. Learn more.

- Clone the new project repository into your local `prac03` folder using `git clone https://gitlab.cs.ut.ee/devops2024-fall/students/devops2024fall-xxx-xxxxxx/Prac02-Docker.git`
  Replace `devops2024fall-xxxxxx-xxxxxx` with your group name.
- Navigate to the cloned repository: `cd Prac02-Docker`
- Copy the "`flask-webapp`" directory to the cloned repository: `cp -r /home/ubuntu/flask-webapp .`
- Create a `.gitignore` file inside `Prac02-Docker` directory with the following lines. This will tell git to ignore the file named "`env.list`" and others in the staging and commit process.

```None
env.list
flask-webapp/flask-app/.env
venv
```

- Add, Commit, and Push the Changes:
  `git add .`
  `git commit -m "Add flask-webapp directory for Lab02"`
- Push the changes to the remote repository: `git push origin main`



Add flask-webapp directory for Lab02
lanxiang authored just now

b1c8dcc0

Name	Last commit	Last update
flask-webapp @ ef97dca9	Add flask-webapp directory for ...	just now
README.md	Initial commit	11 minutes ago

AGS: At this point, Nagios will check if the Prac02-Docker project and the flask-webapp directory are correctly uploaded to the GitLab repository.

Deliverables

- Go to https://scoring.devops.cs.ut.ee/nagios/
- Find your host by your pseudonym
- Make sure that All the services are in OK state.
- UPDATE: Even if you see all OK (green color), still please keep your application running, if any.

Deadline: 2nd Oct 2024, 2 PM EET