

Experiment No. 3

Title:

Implement GitLab Operations using Git.

Objective:

The objective of this experiment is to guide you through the process of using Git commands to interact with GitLab, from creating a repository to collaborating with others through merge requests.

Introduction:

GitLab is a web-based platform that offers a complete DevOps lifecycle toolset, including version control, continuous integration/continuous deployment (CI/CD), project management, code review, and collaboration features. It provides a centralized place for software development teams to work together efficiently and manage the entire development process in a single platform.

Key Features of GitLab:

- **Version Control:** GitLab provides version control capabilities using Git, allowing developers to track changes to source code over time. This enables collaboration, change tracking, and code history maintenance.
- **Repositories:** Repositories on GitLab are collections of files, code, documentation, and assets related to a project. Each repository can have multiple branches and tags, allowing developers to work on different features simultaneously.
- **Continuous Integration/Continuous Deployment (CI/CD):** GitLab offers robust

CI/CD capabilities. It automates the building, testing, and deployment of code changes, ensuring that software is delivered rapidly and reliably.

- **Merge Requests:** Merge requests in GitLab are similar to pull requests in other platforms. They enable developers to propose code changes, collaborate, and get code reviewed before merging it into the main codebase.
- **Issues and Project Management:** GitLab includes tools for managing project tasks, bugs, and enhancements. Issues can be assigned, labeled, and tracked, while project boards help visualize and manage work.
- **Container Registry:** GitLab includes a container registry that allows users to store and manage Docker images for applications.
- **Code Review and Collaboration:** Built-in code review tools facilitate collaboration among team members. Inline comments, code discussions, and code snippets are integral parts of this process.
- **Wiki and Documentation:** GitLab provides a space for creating project wikis and documentation, ensuring that project information is easily accessible and well-documented.
- **Security and Compliance:** GitLab offers security scanning, code analysis, and compliance features to help identify and address security vulnerabilities and ensure code meets compliance standards.
- **GitLab Pages:** Similar to GitHub Pages, GitLab Pages lets users publish static websites directly from a GitLab repository.

Benefits of Using GitLab:

- **End-to-End DevOps:** GitLab offers an integrated platform for the entire software development and delivery process, from code writing to deployment.
- **Simplicity:** GitLab provides a unified interface for version control, CI/CD, and project management, reducing the need to use multiple tools.
- **Customizability:** GitLab can be self-hosted on-premises or used through GitLab's cloud service. This flexibility allows organizations to choose the hosting option that best suits their needs.
- **Security:** GitLab places a strong emphasis on security, with features like role-based access control (RBAC), security scanning, and compliance checks.
- **Open Source and Enterprise Versions:** GitLab offers both a free, open-source Community Edition and a paid, feature-rich Enterprise Edition, making it suitable for individual developers and large enterprises alike.

Prerequisites:


- Computer with Git installed (<https://git-scm.com/downloads>)
- GitLab account (<https://gitlab.com/>)
- Internet connection

Experiment Steps:

Step 1: Creating a Repository


- Sign in to your GitLab account.
- Click the "New" button to create a new project.
- Choose a project name, visibility level (public, private), and other settings.

- Click "Create project."



Create or import your first project

Projects help you organize your work. They contain your file repository, issues, merge requests, and so much more.

| Create | Import |
|--|--------|
| <div><p>Group name</p><input type="text" value="sandy.devops.stuffs-Group1"/></div> <div><p>Project name</p><input type="text" value="sandy.devops.stuffs-Experiment3"/></div> <div><p>Select a template (optional)</p><p>Get started with one of our popular project templates. ?</p><div><div>Select</div><div>▼</div></div></div> <div><p>Your project will be created at:</p><p><code>https://gitlab.com/sandy.devops.stuffs-group1/sandy.devops.stuffs-experiment3</code></p><p>You can always change your URL later</p></div> <div><p><input checked="" type="checkbox"/> Include a Getting Started README</p><p>Recommended if you're new to GitLab</p></div> <div><div>Create project</div></div> | |

Step 2: Cloning a Repository

- Open your terminal or command prompt.

- Add SSH Key (if already you have an SSH key pair in your local machine). Else you run `ssh-keygen` command to create an SSH key pair. Then follow the below steps.

```
ls -la ~/.ssh
```

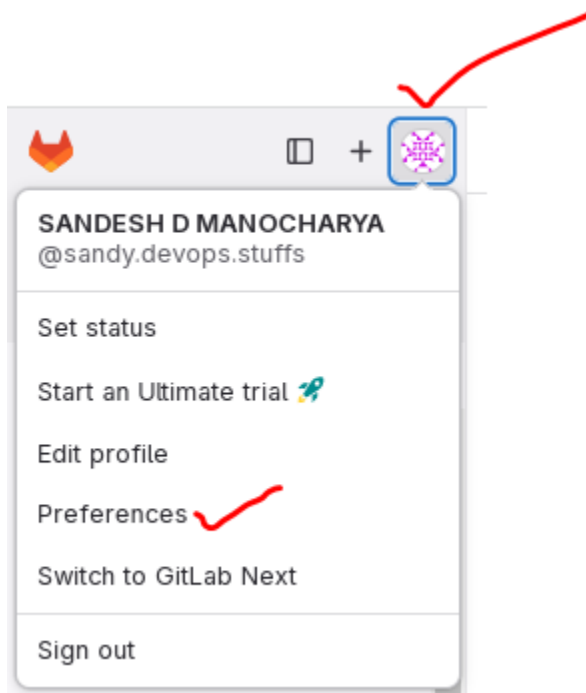
```
ubuntu@ip-172-31-5-122:~$ ls -la ~/.ssh
total 28
drwx-----  2 ubuntu ubuntu 4096 May 11 18:44 .
drwxr-x--- 11 ubuntu ubuntu 4096 May 19 04:13 ..
-rw-----  1 ubuntu ubuntu  386 Apr  9 08:47 authorized_keys
-rw-----  1 ubuntu ubuntu  419 May 11 18:29 id_ed25519
-rw-r--r--  1 ubuntu ubuntu  111 May 11 18:29 id_ed25519.pub
-rw-----  1 ubuntu ubuntu 1120 May 19 04:13 known_hosts
-rw-r--r--  1 ubuntu ubuntu  142 May 11 18:44 known_hosts.old
ubuntu@ip-172-31-5-122:~$ |
```

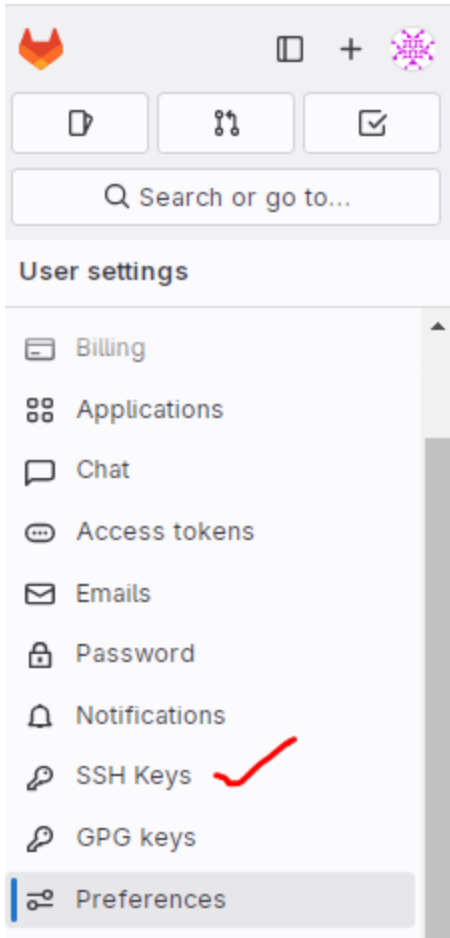
```
cat ~/.ssh/id_ed25519.pub
```

```
ubuntu@ip-172-31-5-122:~$ cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIjM4sJT4T1WE2hz0lX99MawwR/1Nq2s1XetDJwjEaUfP sandy.devops.stuffs@gmail.com
ubuntu@ip-172-31-5-122:~$ |
```

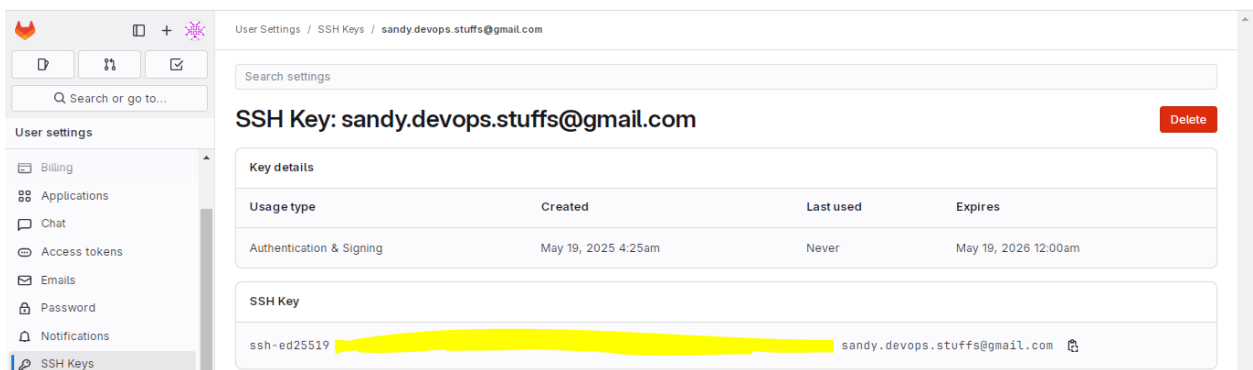
Go to **GitLab** → **Preferences** → **SSH Keys**

(or open: <https://gitlab.com/-/profile/keys>)





Paste the key and click **Add key**.



- Navigate to the directory where you want to clone the repository.

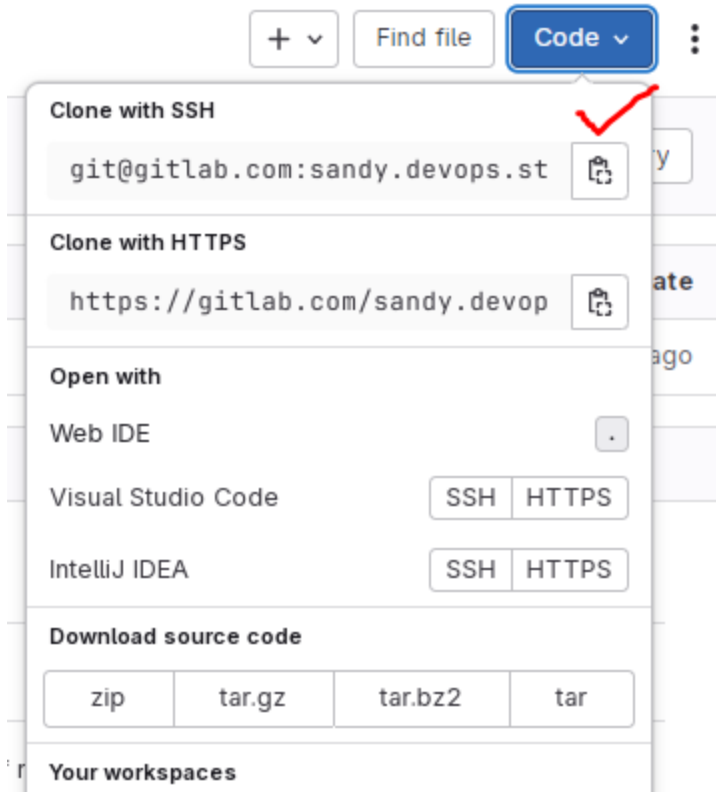
```

ubuntu@ip-172-31-5-122:~$ ls -la
total 100868
drwxr-x--- 11 ubuntu ubuntu 4096 May 14 05:45 .
drwxr-xr-x  3 root  root  4096 Apr  9 08:47 ..
drwxrwxr-x  2 ubuntu docker 4096 Apr  9 09:20 .aws
-rw-----  1 ubuntu ubuntu 7312 May 14 05:53 .bash_history
-rw-r--r--  1 ubuntu ubuntu 220 Mar 31 2024 .bash_logout
-rw-r--r--  1 ubuntu ubuntu 3771 Mar 31 2024 .bashrc
drwx-----  3 ubuntu ubuntu 4096 Apr  9 09:26 .cache
drwx-----  2 ubuntu ubuntu 4096 Apr  9 11:36 .docker
drwxr-xr-x  3 ubuntu ubuntu 4096 Apr 13 11:11 .kube
-rw-----  1 ubuntu ubuntu  20 May 14 05:45 .lessht
-rw-r--r--  1 ubuntu ubuntu 807 Mar 31 2024 .profile
drwx-----  2 ubuntu ubuntu 4096 May 11 18:44 .ssh
-rw-r--r--  1 ubuntu ubuntu  0 Apr  9 08:53 .sudo_as_admin_successful
-rw-----  1 ubuntu docker 10512 May 14 05:24 .viminfo
drwxrwxr-x  3 ubuntu ubuntu 4096 May 11 18:57 Experiment1
drwxrwxr-x  3 ubuntu ubuntu 4096 May 14 05:45 Experiment2
drwxr-xr-x  3 ubuntu docker 4096 Apr  8 18:42 aws
-rw-rw-r--  1 ubuntu docker 68286133 Apr  9 09:09 awscli2.zip
-rw-rw-r--  1 ubuntu docker 34916991 Apr  9 09:05 eksctl.tar.gz
drwxrwxr-x  3 ubuntu docker 4096 Apr 13 09:41 flask-app
ubuntu@ip-172-31-5-122:~$
ubuntu@ip-172-31-5-122:~$
ubuntu@ip-172-31-5-122:~$ |

```

- Copy the repository URL from GitLab.

The screenshot shows the GitLab web interface for a repository named 'sandy.devops.stuffs-Experiment3'. The repository is owned by 'sandy.devops.stuffs-Group1' and is currently on the 'main' branch. The 'Code' dropdown menu is open, showing options to clone the repository using SSH or HTTPS, or to open it with a web IDE, Visual Studio Code, or IntelliJ IDEA. The 'Clone with SSH' option is selected, and the URL 'git@gitlab.com:sandy.devops.stuffs-Experiment3.git' is displayed. The 'Clone with HTTPS' option is also visible, with the URL 'https://gitlab.com/sandy.devops.stuffs-Experiment3.git'. The 'Open with' section shows options for Web IDE, Visual Studio Code, and IntelliJ IDEA. The 'Download source code' section shows options for zip, tar.gz, tar.bz2, and tar. The 'Your workspaces' section is also visible, indicating that no agents are available to create workspaces.



- Run the following command:

```
git clone <repository_url>
```

- Replace <repository_url> with the URL you copied from GitLab.
- This will clone the repository to your local machine.

```
ubuntu@ip-172-31-5-122:~$ git clone git@gitlab.com:sandy.devops.stuffs-group1/sandy.devops.stuffs-Experiment3.git
Cloning into 'sandy.devops.stuffs-Experiment3'...
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.
ubuntu@ip-172-31-5-122:~$
```

```
ls -ls
```



```
ubuntu@ip-172-31-5-122:~$ ls -ls
total 100808
  4 drwxrwxr-x 3 ubuntu ubuntu    4096 May 11 18:57 Experiment1
  4 drwxrwxr-x 3 ubuntu ubuntu    4096 May 14 05:45 Experiment2
  4 drwxr-xr-x 3 ubuntu docker    4096 Apr  8 18:42 aws
66688 -rw-rw-r-- 1 ubuntu docker 68286133 Apr  9 09:09 awscli2.zip
34100 -rw-rw-r-- 1 ubuntu docker 34916991 Apr  9 09:05 eksctl.tar.gz
  4 drwxrwxr-x 3 ubuntu docker    4096 Apr 13 09:41 flask-app
  4 drwxrwxr-x 3 ubuntu ubuntu    4096 May 19 04:27 sandy.devops.stuffs-Experiment3
ubuntu@ip-172-31-5-122:~$ |
```

Step 3: Making Changes and Creating a Branch

- Navigate into the cloned repository:

Syntax: `cd <repository_name>`

```
cd sandy.devops.stuffs-Experiment3
```

```
ls -la
```

```
ubuntu@ip-172-31-5-122:~$ cd sandy.devops.stuffs-Experiment3
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ ls -la
total 20
drwxrwxr-x  3 ubuntu ubuntu 4096 May 19 04:27 .
drwxr-x--- 12 ubuntu ubuntu 4096 May 19 04:27 ..
drwxrwxr-x  8 ubuntu ubuntu 4096 May 19 04:27 .git
-rw-rw-r--  1 ubuntu ubuntu 6223 May 19 04:27 README.md
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ |
```

- Create a new text file named "example.txt" using a text editor.
- Add some content to the "example.txt" file.
- Save the file and return to the command line.

```
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ vi example.txt
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ cat example.txt
Welcome to experiment no 3
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ |
```

- Check the status of the repository:

```
git status
```

```
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
   example.txt

nothing added to commit but untracked files present (use "git add" to track)
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ |
```

- Stage the changes for commit:

```
git add example.txt
```

```
git status
```

```
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ git add example.txt
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ 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)
   new file:   example.txt

ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ |
```

- Commit the changes with a descriptive message:

```
git commit -m "Added content to example.txt"
```

```
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ git commit -m "Added content to example.txt"
[main 3elfd1] Added content to example.txt
Committer: Ubuntu <ubuntu@ip-172-31-5-122.ap-south-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 1 insertion(+)
create mode 100644 example.txt
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ |
```

```
git status
```

```
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ 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
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ |
```

- Create a new branch named "feature":

```
git branch
```

```
git branch feature
```

```
git branch
```

```
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ git branch
* main
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ git branch feature
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ git branch
feature
* main
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ |
```

- Switch to the "feature" branch:

```
git checkout feature
```

```
git branch
```

```
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ git checkout feature
Switched to branch 'feature'
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ git branch
* feature
main
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ |
```

Step 4: Pushing Changes to GitLab

- Add Repository URL in a variable

```
git remote add origin <repository_url>
```

- Replace <repository_url> with the URL you copied from GitLab.

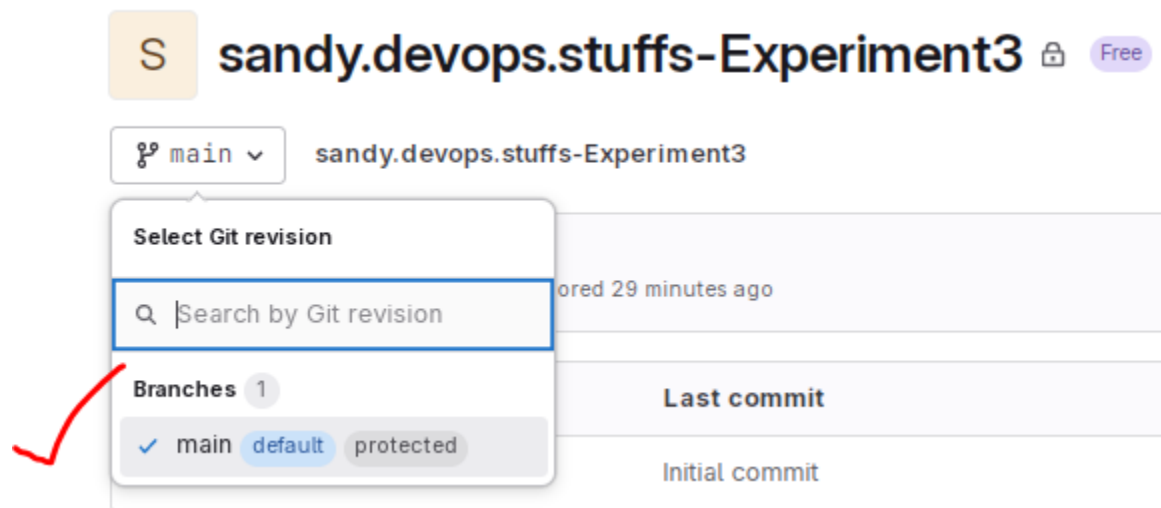
```
git remote add origin git@gitlab.com:sandy.devops.stuffs-group1/sandy.devops.stuffs-Experiment3.git
```

```
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ git remote add origin git@gitlab.com:sandy.devops.stuffs-group1/sandy.devops.stuffs-Experiment3.git
error: remote origin already exists.
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ |
```

Note: Since we already cloned the repo we got this error saying the “error: remote origin already exists.”

- Push the "feature" branch to GitLab:

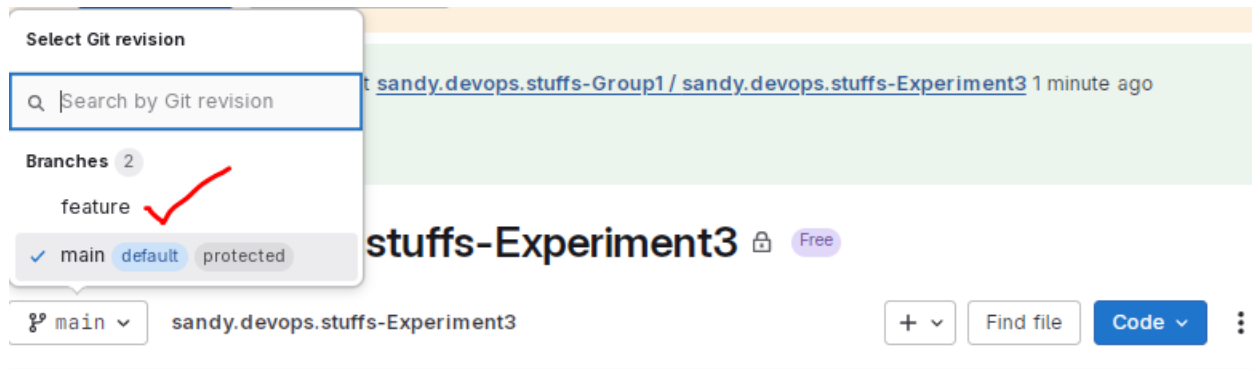
Right now there is only `main` branch in GitLab.



```
git push origin feature
```

```
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ git push origin feature
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 339 bytes | 339.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: To create a merge request for feature, visit:
remote:   https://gitlab.com/sandy.devops.stuffs-group1/sandy.devops.stuffs-Experiment3/-/merge_requests/new?merge_request%5Bsource_branch%5D=feature
remote:
To gitlab.com:sandy.devops.stuffs-group1/sandy.devops.stuffs-Experiment3.git
 * [new branch]   feature -> feature
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ |
```

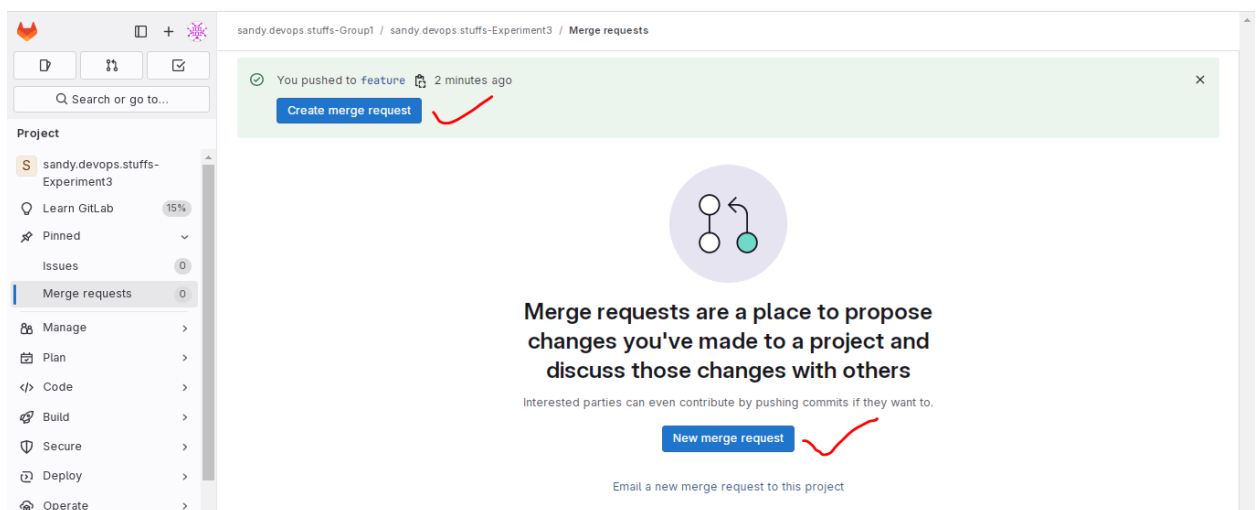
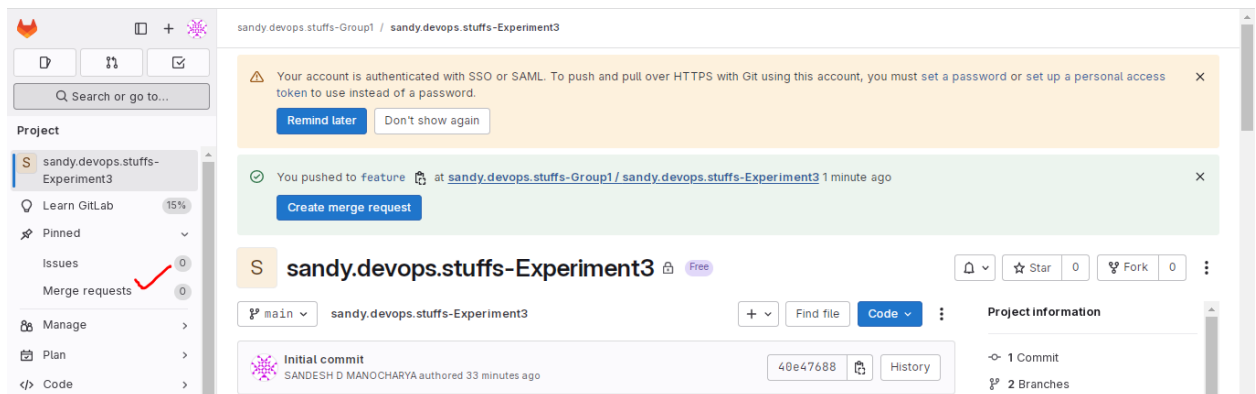
- Check your GitLab repository to confirm that the new branch "feature" is available.



Step 5: Collaborating through Merge Requests

1. Create a merge request on GitLab:

- Go to the repository on GitLab.
- Click on "Merge Requests" and then "New Merge Request."



- Choose the source branch ("feature") and the target branch ("main" or "master").

New merge request

Source branch

sandy.devops.stuffs-group1/sandy... feature

Added content to example.txt
 Ubuntu authored May 19, 2025
 3e1fd18

Compare branches and continue

Target branch

sandy.devops.stuffs-group1/sandy... main

Initial commit
 SANDESH D MANOCHARYA authored May 19, 2025
 40e47688

- Review the changes and click "Submit merge request."

2. Review and merge the merge request:

- Add a title and description for the merge request.
- Assign reviewers if needed.
- Once the merge request is approved, merge it into the target branch.

sandy.devops.stuffs-Group1 / sandy.devops.stuffs-Experiment3 / Merge requests / 1

Added content to example.txt

Open
 SANDESH D MANOCHARYA requested to merge feature into main 25 seconds ago

Overview 0 Commits - Pipelines 0 Changes 1

0
 0
 0

Before you can run pipelines, we need to verify your account.

We won't ask you for this information again. It will never be used for marketing purposes.

[Verify my account](#)

8 ✓ [Approve](#) 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](#)

0 Assignees Edit
 None - assign yourself

0 Reviewers Edit
 None - assign yourself

Labels Edit
 None

Milestone Edit
 None

Time tracking +
 No estimate or time spent

1 Participant

Merged by [SANDESH D MANOCHARYA](#) 25 seconds ago
 [Revert](#) [Cherry-pick](#) [Delete source branch](#)

Merge details

- Changes merged into main with [7ae85fd4](#).
- Did not delete the source branch.

Step 6: Syncing Changes

- After the merge request is merged, update your local repository:

```
git branch
```

```
git checkout main
```

```
git branch
```

```
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ git branch
* feature
  main
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ 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)
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ git branch
  feature
* main
```

```
git pull origin main
```

```
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ git pull origin main
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (1/1), 291 bytes | 291.00 KiB/s, done.
From gitlab.com:sandy.devops.stuffs-group1/sandy.devops.stuffs-Experiment3
 * branch            main       -> FETCH_HEAD
   40e4768..7ae85fd  main       -> origin/main
Updating 3e1f1d1..7ae85fd
Fast-forward
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ |
```

```
cat example.txt
```

```
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ cat example.txt
Welcome to experiment no 3
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ |
```

```
git checkout feature
```

```
git branch
```

```
cat example.txt
```

```
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ git checkout feature
Switched to branch 'feature'
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ git branch
* feature
  main
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ cat example.txt
Welcome to experiment no 3
ubuntu@ip-172-31-5-122:~/sandy.devops.stuffs-Experiment3$ |
```

Conclusion:

This experiment provided you with practical experience in performing GitLab operations using Git commands. You learned how to create repositories, clone them to your local machine, make changes, create branches, push changes to GitLab, collaborate through merge requests, and synchronize changes with remote repositories. These skills are crucial for effective collaboration and version control in software development projects using GitLab and Git.

Questions/Exercises:

1. What is GitLab, and how does it differ from other version control platforms?
2. Explain the significance of a GitLab repository. What can a repository contain?
3. What is a merge request in GitLab? How does it facilitate the code review process?
4. Describe the steps involved in creating and submitting a merge request on GitLab.
5. What are GitLab issues, and how are they used in project management?
6. Explain the concept of a GitLab project board and its purpose in organizing tasks.
7. How does GitLab address security concerns in software development? Mention some security-related features.
8. Describe the role of compliance checks in GitLab and how they contribute to maintaining software quality.