Experiment No. 4

Title:

Implement BitBucket Operations using Git.

Objective:

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

Introduction:

Bitbucket is a web-based platform designed to provide version control, source code management, and collaboration tools for software development projects. It is widely used by teams and individuals to track changes in code, collaborate on projects, and streamline the development process. Bitbucket offers Git and Mercurial as version control systems and provides features to support code collaboration, continuous integration/continuous deployment (CI/CD), and project management.

Key Features of BitBucket:

● Version Control: Bitbucket supports both Git and Mercurial version control systems, allowing developers to track changes, manage code history, and work collaboratively on projects.

● Repositories: In Bitbucket, a repository is a container for code, documentation, and other project assets. It houses different branches, tags, and commits that represent different versions of the project.

● Collaboration: Bitbucket enables team collaboration through features like pull requests, code reviews, inline commenting, and team permissions. These tools help streamline the process of merging code changes.

● Pull Requests: Pull requests in Bitbucket allow developers to propose and review code changes before they are merged into the main codebase. This process helps ensure code quality and encourages collaboration.

● Code Review: Bitbucket provides tools for efficient code review, allowing team members to comment on specific lines of code and discuss changes within the context of the code itself.

● Continuous Integration/Continuous Deployment (CI/CD): Bitbucket integrates with CI/CD pipelines, automating processes such as building, testing, and deploying code changes to various environments.

● Project Management: Bitbucket offers project boards and issue tracking to help manage tasks, track progress, and plan project milestones effectively.

● Bitbucket Pipelines: This feature allows teams to define and automate CI/CD pipelines directly within Bitbucket, ensuring code quality and rapid delivery.

● Access Control and Permissions: Bitbucket allows administrators to define user roles, permissions, and access control settings to ensure the security of repositories and project assets.

Benefits of Using BitBucket:

- Version Control: Bitbucket's integration with Git and Mercurial provides efficient version control and code history tracking.

● Collaboration: The platform's collaboration tools, including pull requests and code reviews, improve code quality and facilitate team interaction.

● CI/CD Integration: Bitbucket's integration with CI/CD pipelines automates testing and deployment, resulting in faster and more reliable software delivery.

● Project Management: Bitbucket's project management features help teams organize tasks, track progress, and manage milestones.

● Flexibility: Bitbucket offers both cloud-based and self-hosted options, providing flexibility to choose the deployment method that suits the organization's needs.

● Integration: Bitbucket integrates with various third-party tools, services, and extensions, enhancing its functionality and extending its capabilities.

Prerequisites:

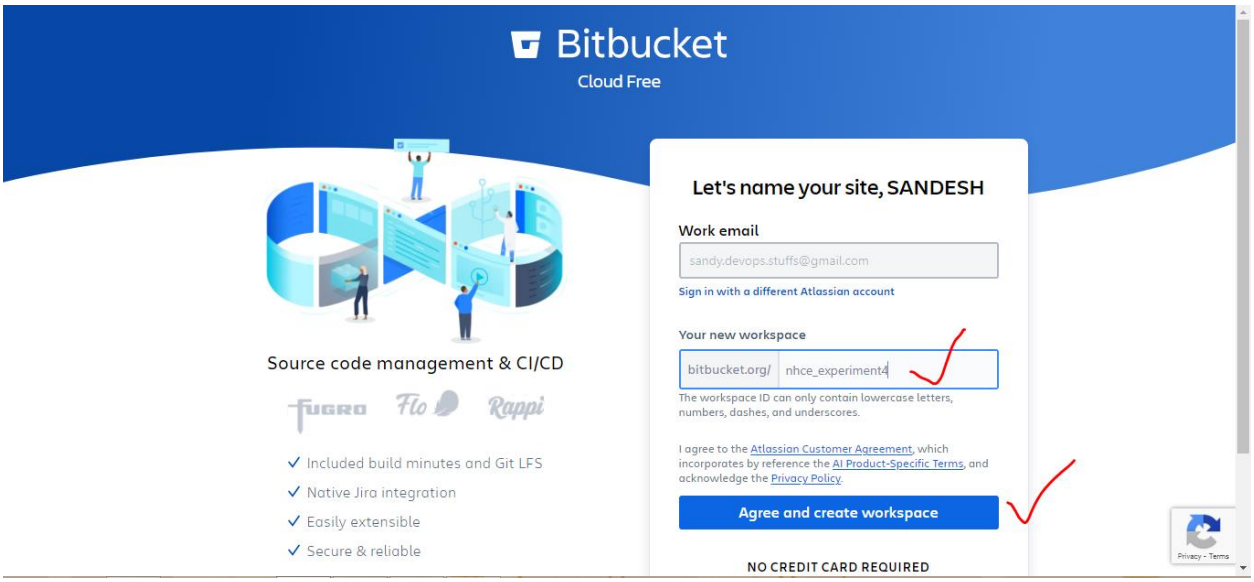
● Computer with Git installed (https://git-scm.com/downloads)

- Bitbucket account (https://bitbucket.org/)
- Internet connection

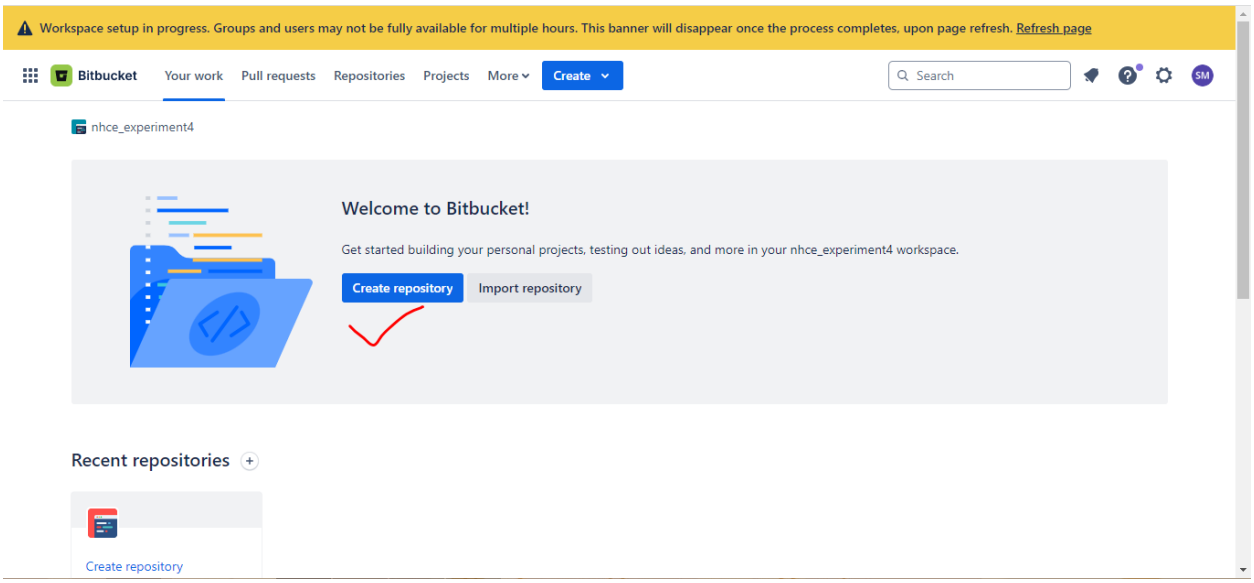
Experiment Steps:

Step 1: Creating a Repository

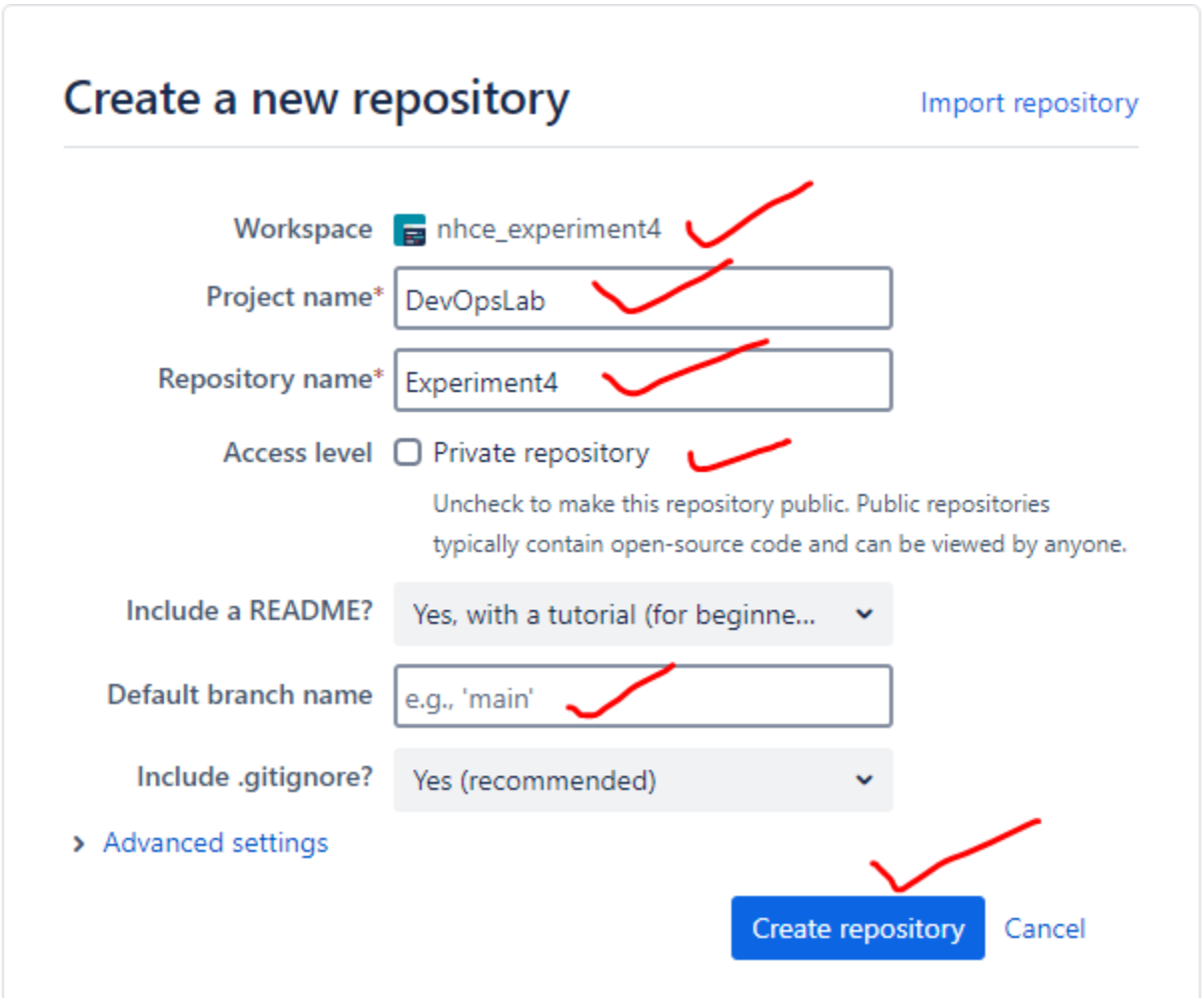
- Sign in to your Bitbucket account.



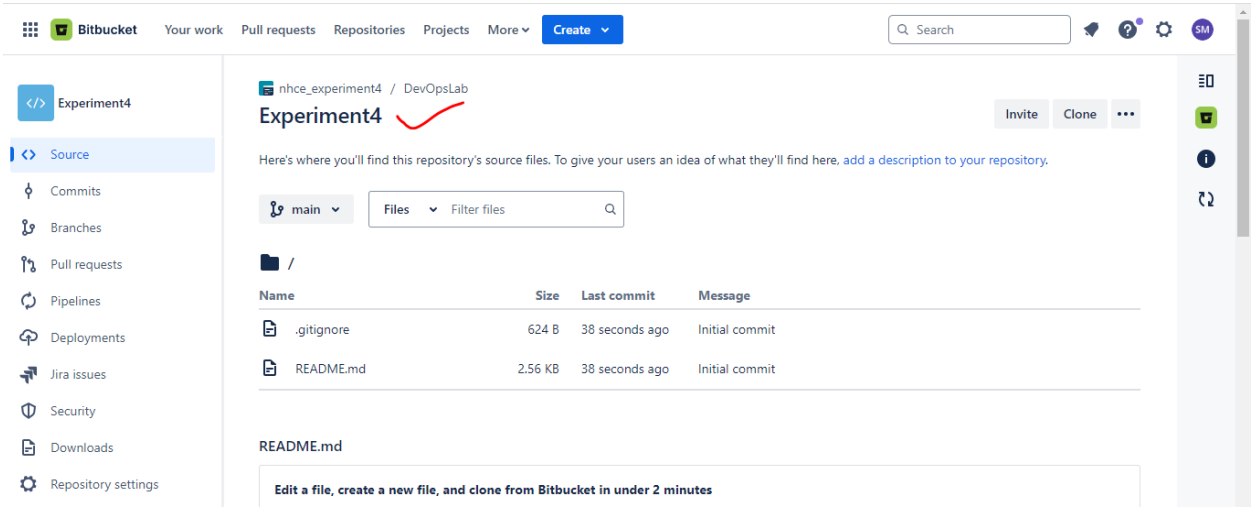
- Click the "Create" button to create a new repository.



● Choose a repository name, visibility (public or private), and other settings.



● Click "Create repository."



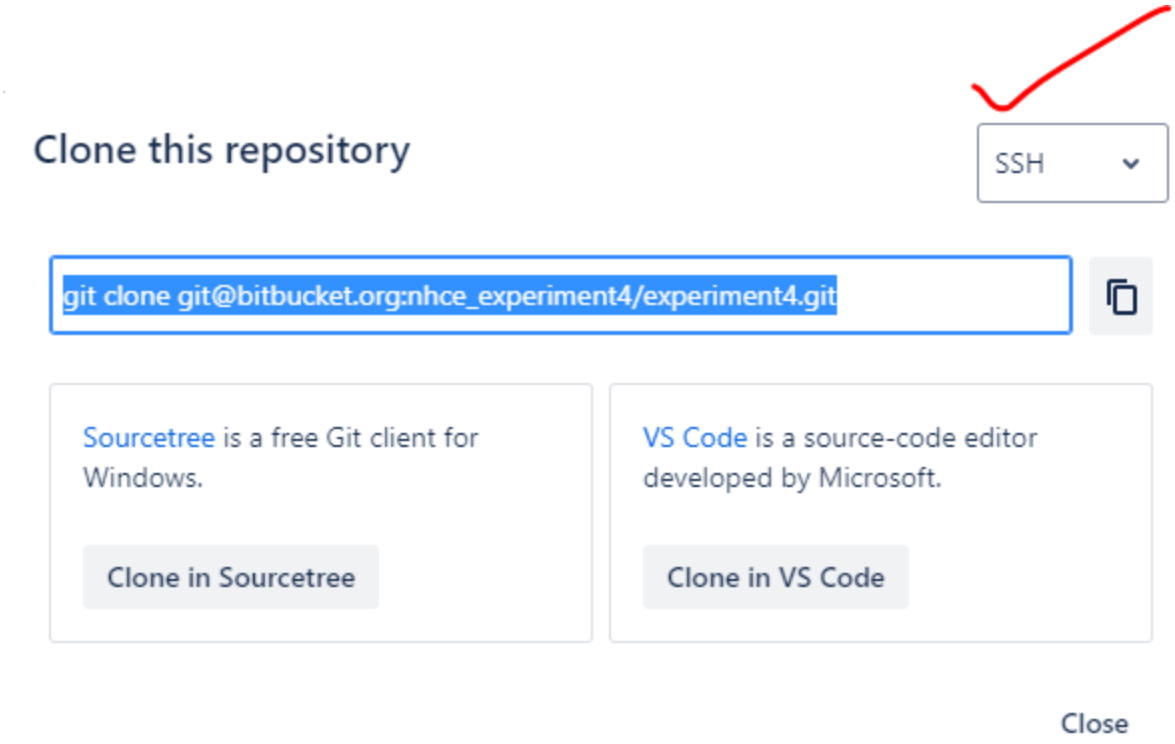
Step 2: Cloning a Repository

● Open your terminal or command prompt.

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

```
ubuntu@ip-172-31-5-122:~$ ls -la
total 100872
drwxr-x--- 12 ubuntu ubuntu     4096 May 19 04:52 .
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     8134 May 19 04:58 .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 19 04:52 .lesshst
-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    10731 May 19 04:31 .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 awscliv2.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
drwxrwxr-x  3 ubuntu ubuntu     4096 May 19 04:31 sandy.devops.stuffs-Experiment3
ubuntu@ip-172-31-5-122:~$
```

● Copy the repository URL from Bitbucket.



- Run the following command:

```
git clone <ssh_repository_url>
```

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

```
ubuntu@ip-172-31-5-122:~$ git clone https://nhce_experiment4-admin@bitbucket.org/nhce_experiment4/experiment4.git
Cloning into 'experiment4'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (4/4), done.
ubuntu@ip-172-31-5-122:~$
```

- This will clone the repository to your local machine.

```
ubuntu@ip-172-31-5-122:~$ ls -la
total 100876
drwxr-x--- 13 ubuntu ubuntu     4096 May 23 11:34 .
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     8134 May 19 04:58 .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 19 04:52 .lesshst
-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    10731 May 19 04:31 .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 awscliv2.zip
-rw-rw-r--  1 ubuntu docker 34916991 Apr  9 09:05 eksctl.tar.gz
drwxrwxr-x  3 ubuntu ubuntu     4096 May 23 11:34 experiment4
drwxrwxr-x  3 ubuntu docker     4096 Apr 13 09:41 flask-app
drwxrwxr-x  3 ubuntu ubuntu     4096 May 19 04:31 sandy.devops.stuffs-Experiment3
ubuntu@ip-172-31-5-122:~$
```

Step 3: Making Changes and Creating a Branch

● Navigate into the cloned repository:

```
cd <repository_name>
```

```
ls -la
```

```
ubuntu@ip-172-31-5-122:~$ cd experiment4/
ubuntu@ip-172-31-5-122:~/experiment4$ ls -la
total 20
drwxrwxr-x  3 ubuntu ubuntu 4096 May 23 11:34 .
drwxr-x--- 13 ubuntu ubuntu 4096 May 23 11:34 ..
drwxrwxr-x  8 ubuntu ubuntu 4096 May 23 11:34 .git
-rw-rw-r--  1 ubuntu ubuntu  624 May 23 11:34 .gitignore
-rw-rw-r--  1 ubuntu ubuntu 2622 May 23 11:34 README.md
ubuntu@ip-172-31-5-122:~/experiment4$
```

● 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:~/experiment4$ vi example.txt
ubuntu@ip-172-31-5-122:~/experiment4$ cat example.txt
Welcome to experiment4
ubuntu@ip-172-31-5-122:~/experiment4$
```

● Check the status of the repository:

```
git status
```

```
ubuntu@ip-172-31-5-122:~/experiment4$ 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:~/experiment4$
```

● Stage the changes for commit:

```
git add example.txt

git status
```

```
ubuntu@ip-172-31-5-122:~/experiment4$ git add example.txt
ubuntu@ip-172-31-5-122:~/experiment4$ 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:~/experiment4$
```

● Commit the changes with a descriptive message:

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

```
ubuntu@ip-172-31-5-122:~/experiment4$ git commit -m "Added content to example.txt"
[main 8112097] 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:~/experiment4$
```

```
git status
```

```
ubuntu@ip-172-31-5-122:~/experiment4$ 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:~/experiment4$
```

● Create a new branch named "feature":

```
git branch

git branch feature
```

```
ubuntu@ip-172-31-5-122:~/experiment4$ git branch
* main
ubuntu@ip-172-31-5-122:~/experiment4$
ubuntu@ip-172-31-5-122:~/experiment4$
ubuntu@ip-172-31-5-122:~/experiment4$ git branch feature
ubuntu@ip-172-31-5-122:~/experiment4$
ubuntu@ip-172-31-5-122:~/experiment4$ git branch
  feature
* main
ubuntu@ip-172-31-5-122:~/experiment4$
```

● Switch to the "feature" branch:

```
git checkout feature

git branch
```

```
ubuntu@ip-172-31-5-122:~/experiment4$ git checkout feature
Switched to branch 'feature'
ubuntu@ip-172-31-5-122:~/experiment4$
ubuntu@ip-172-31-5-122:~/experiment4$ git branch
* feature
  main
ubuntu@ip-172-31-5-122:~/experiment4$
```

Step 4: Pushing Changes to Bitbucket

● Add Repository URL in a variable

```
git remote add origin <ssh_repository_url>
```

● Replace <repository_url> with the URL you copied from Bitbucket.

```
ubuntu@ip-172-31-5-122:~/experiment4$ git remote add origin git@bitbucket.org:nhce_experiment4/experiment4.git
error: remote origin already exists.
ubuntu@ip-172-31-5-122:~/experiment4$
```

NOTE: We got the error because already we have cloned the repo from Bitbucket.

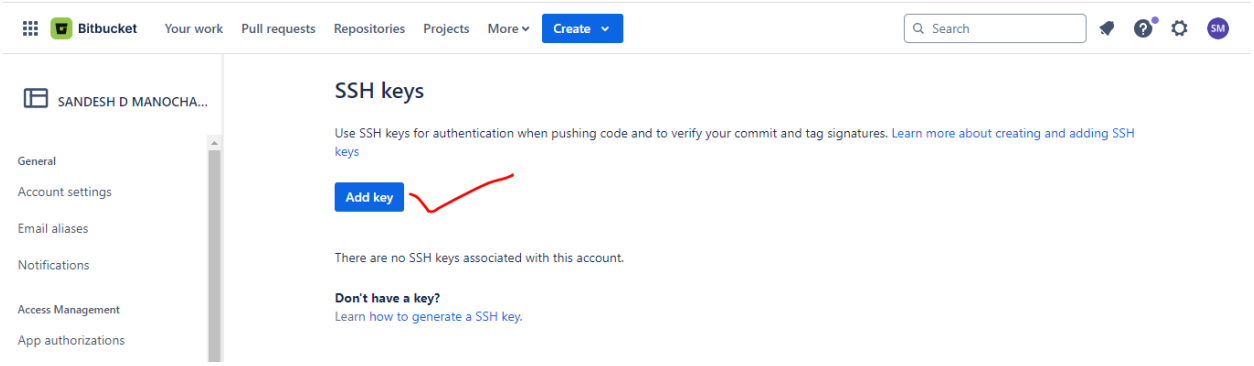
- Add SSH key to Bitbucket.

```
cd

cd .ssh

ls -la

cat id_ed25519.pub
```

```
ubuntu@ip-172-31-5-122:~/experiment4$ cd
ubuntu@ip-172-31-5-122:~$ cd .ssh
ubuntu@ip-172-31-5-122:~/.ssh$ ls -la
total 28
drwx------  2 ubuntu ubuntu 4096 May 11 18:44 .
drwxr-x--- 13 ubuntu ubuntu 4096 May 23 11:53 ..
-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:~/.ssh$ cat id_ed25519.pub
ssh-ed25519 [redacted] sandy.devops.stuffs@gmail.com
ubuntu@ip-172-31-5-122:~/.ssh$
```

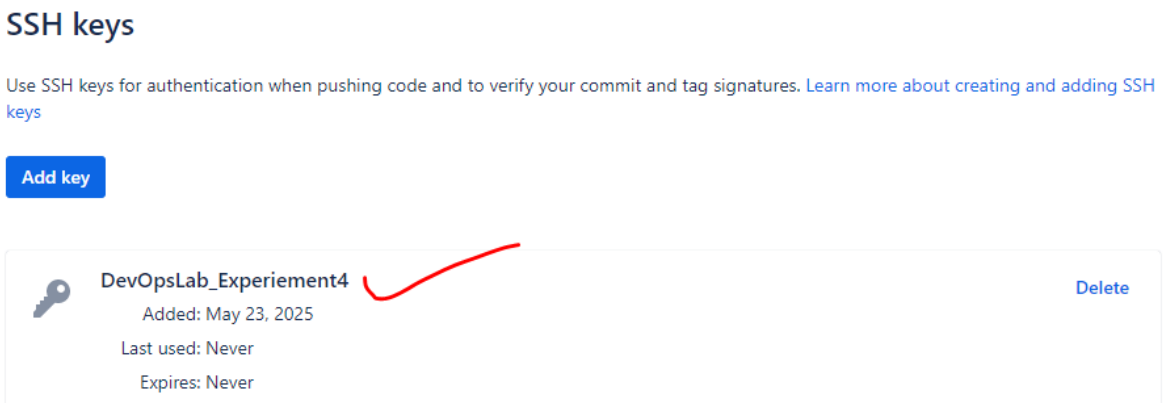
- Copy the content of `id_ed25519.pub` and go to https://bitbucket.org/account/settings/ssh-keys/

- Click **"Add key"**



- Paste your public key (copied from the previous step)
- Give it a meaningful **Label** (e.g., Ubuntu-Experiment4)

- Save



- Test the conection

```
ssh -T git@bitbucket.org
```

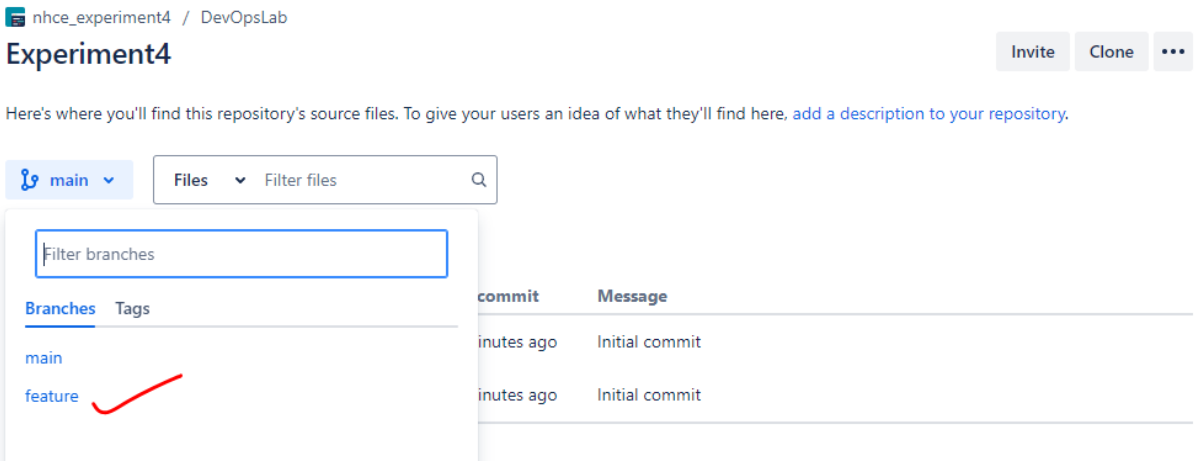
```
ubuntu@ip-172-31-5-122:~/experiment4$ ssh -T git@bitbucket.org
The authenticity of host 'bitbucket.org (13.200.41.136)' can't be established.
ED25519 key fingerprint is SHA256:ybgmFkzwOSotHTHLJgHOOQN8LOxErw6vd0VhFA9m3SM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'bitbucket.org' (ED25519) to the list of known hosts.
authenticated via ssh key.

You can use git to connect to Bitbucket. Shell access is disabled
ubuntu@ip-172-31-5-122:~/experiment4$
```

- Push the "feature" branch to Bitbucket:

```
git push origin feature
```

```
ubuntu@ip-172-31-5-122:~/experiment4$ 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), 368 bytes | 368.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create pull request for feature:
remote:   https://bitbucket.org/nhce_experiment4/experiment4/pull-requests/new?source=feature&t=1
remote:
To bitbucket.org:nhce_experiment4/experiment4.git
 * [new branch]      feature -> feature
ubuntu@ip-172-31-5-122:~/experiment4$
```

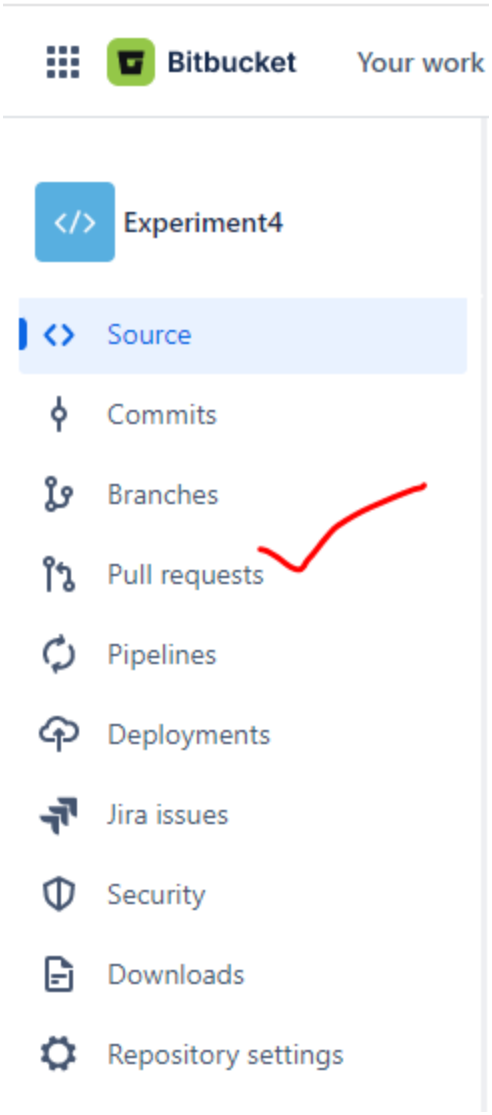
- Check your Bitbucket repository to confirm that the new branch "feature" is available.



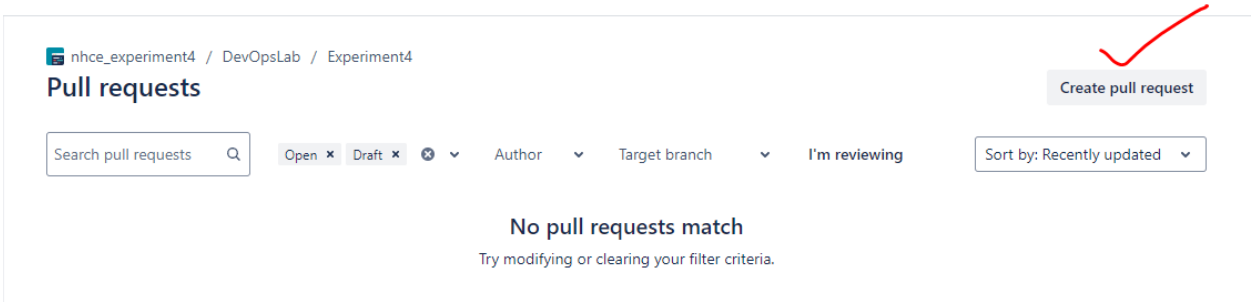
Step 5: Collaborating through Pull Requests

1. Create a pull request on Bitbucket:

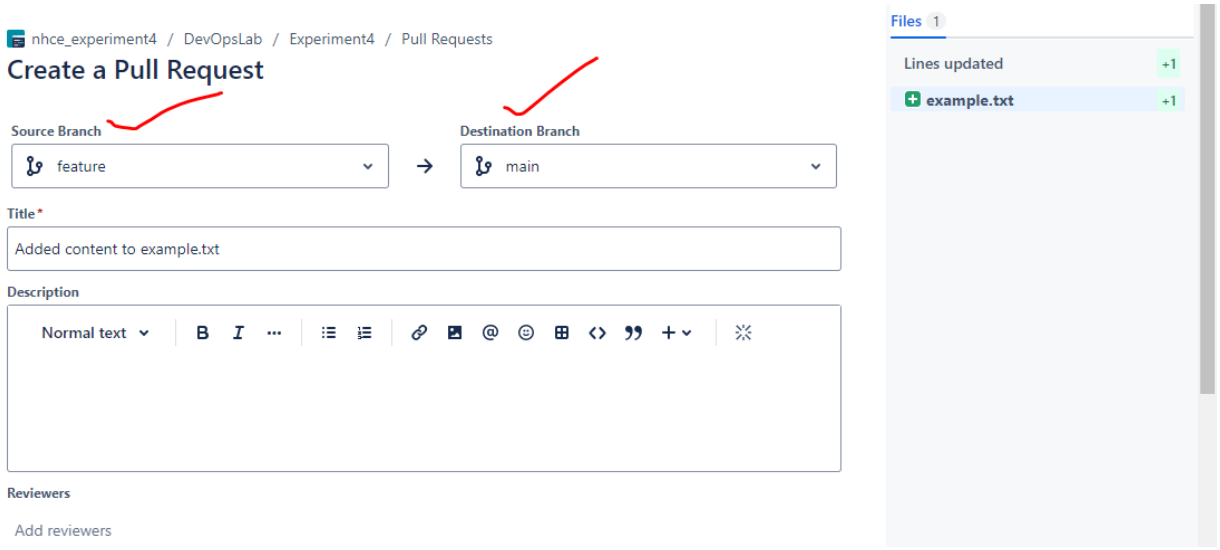
- Go to the repository on Bitbucket.



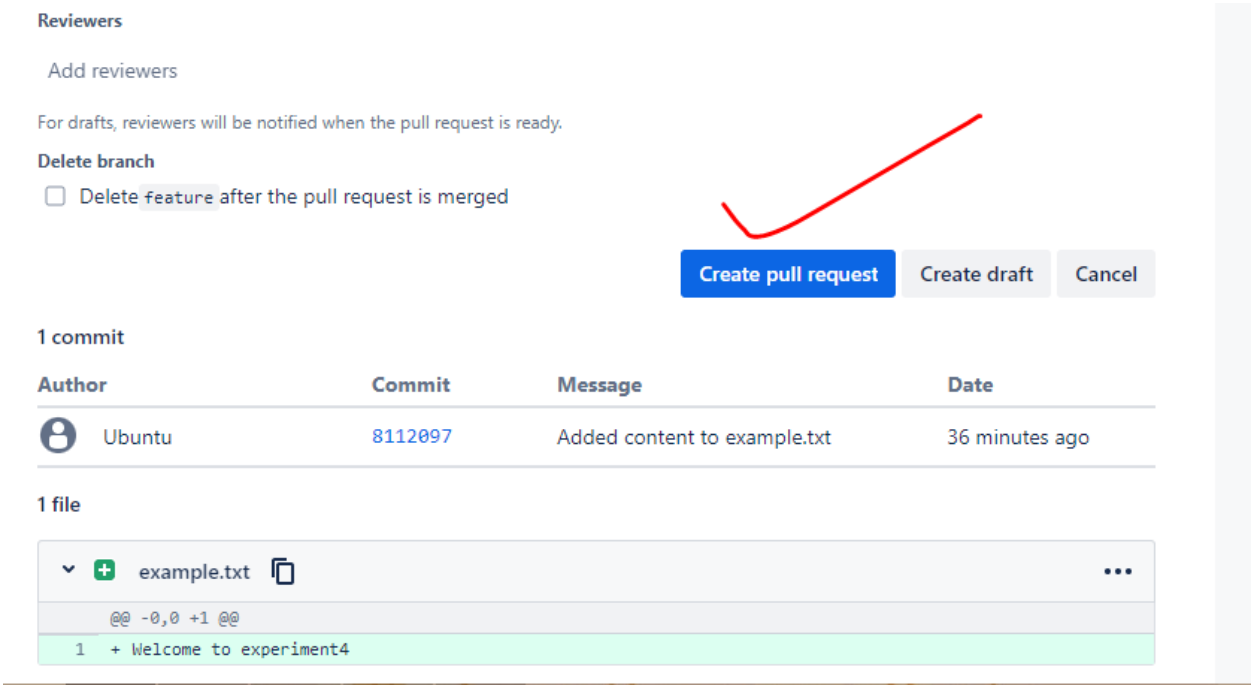
- Click on "Create pull request."

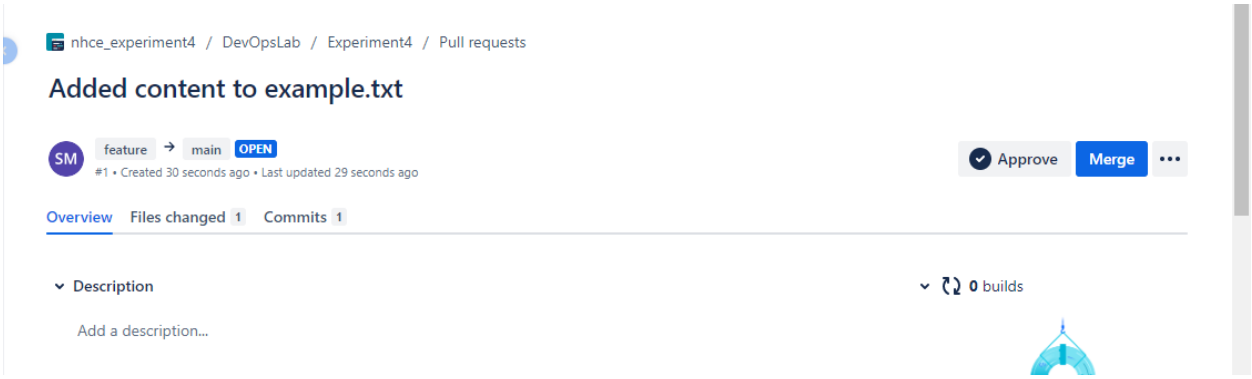


- Choose the source branch ("feature") and the target branch ("main" or "master").



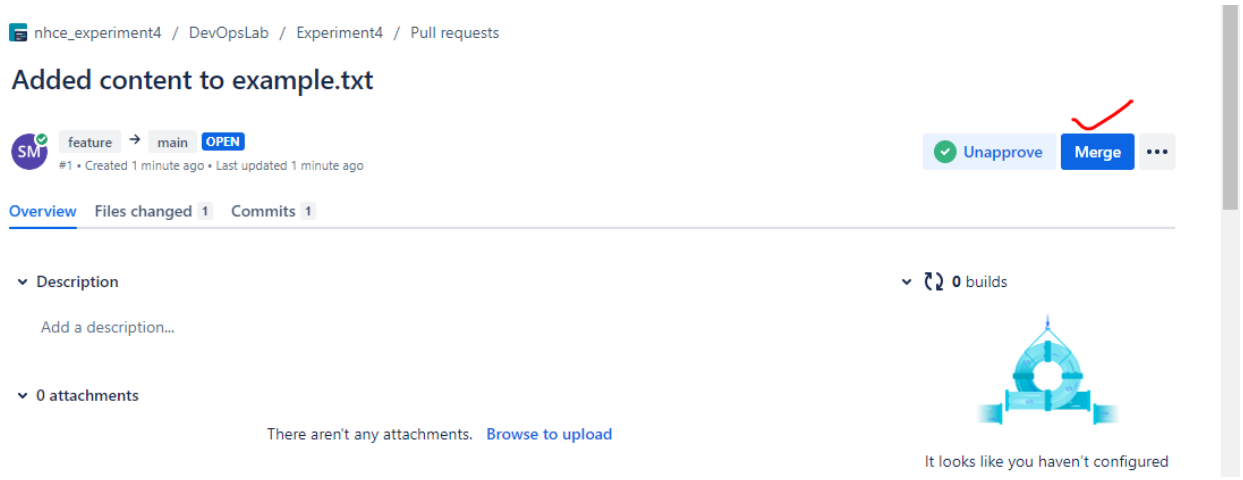
- Review the changes and click "Create pull request."

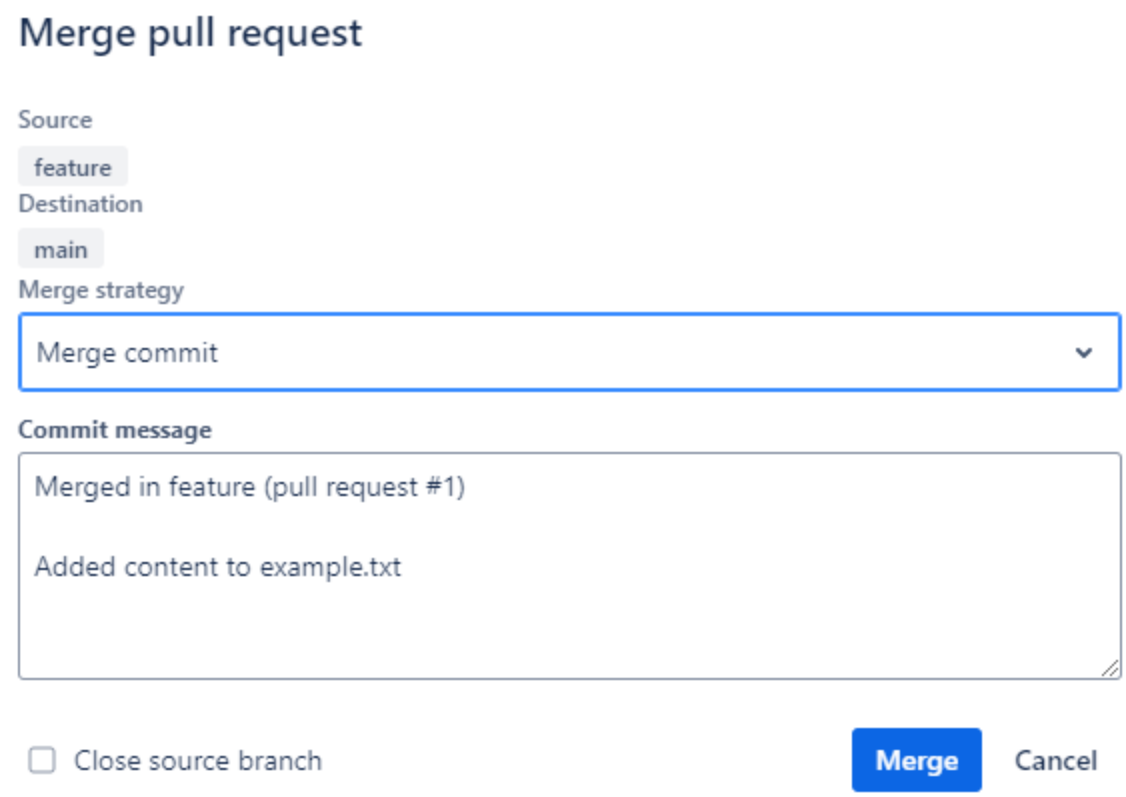


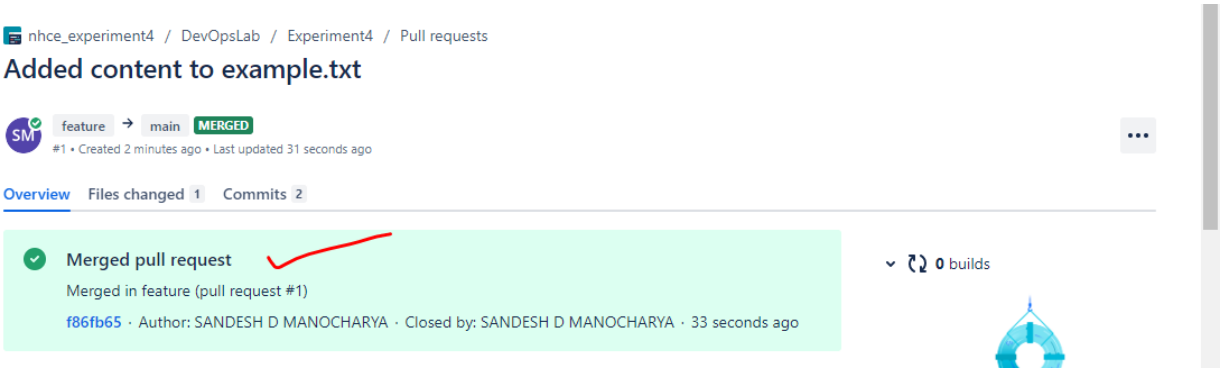


2. Review and merge the pull request:

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







Step 6: Syncing Changes

● After the pull request is merged, update your local repository:

```
git branch
```

```
git checkout main
```

`git branch`

```
ubuntu@ip-172-31-5-122:~/experiment4$ git branch
* feature
  main
ubuntu@ip-172-31-5-122:~/experiment4$
ubuntu@ip-172-31-5-122:~/experiment4$
ubuntu@ip-172-31-5-122:~/experiment4$ 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:~/experiment4$
ubuntu@ip-172-31-5-122:~/experiment4$
ubuntu@ip-172-31-5-122:~/experiment4$ git branch
  feature
* main
ubuntu@ip-172-31-5-122:~/experiment4$
```

`git pull origin main`

```
ubuntu@ip-172-31-5-122:~/experiment4$ 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), 259 bytes | 259.00 KiB/s, done.
From bitbucket.org:nhce_experiment4/experiment4
 * branch            main       -> FETCH_HEAD
   1f83a4f..f86fb65  main       -> origin/main
Updating 8112097..f86fb65
Fast-forward
ubuntu@ip-172-31-5-122:~/experiment4$
```

- Now go to Bitbucket to see the merged code.

Conclusion:

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

Questions/Exercises:

Q.1 What is Bitbucket, and how does it fit into the DevOps landscape?

Q.2 Explain the concept of branching in Bitbucket and its significance in collaborative development.

Q.3 What are pull requests in Bitbucket, and how do they facilitate code review and collaboration?

Q.4 How can you integrate code quality analysis and security scanning tools into Bitbucket's CI/CD pipelines?

Q.5 What are merge strategies in Bitbucket, and how do they affect the merging process during pull requests?