

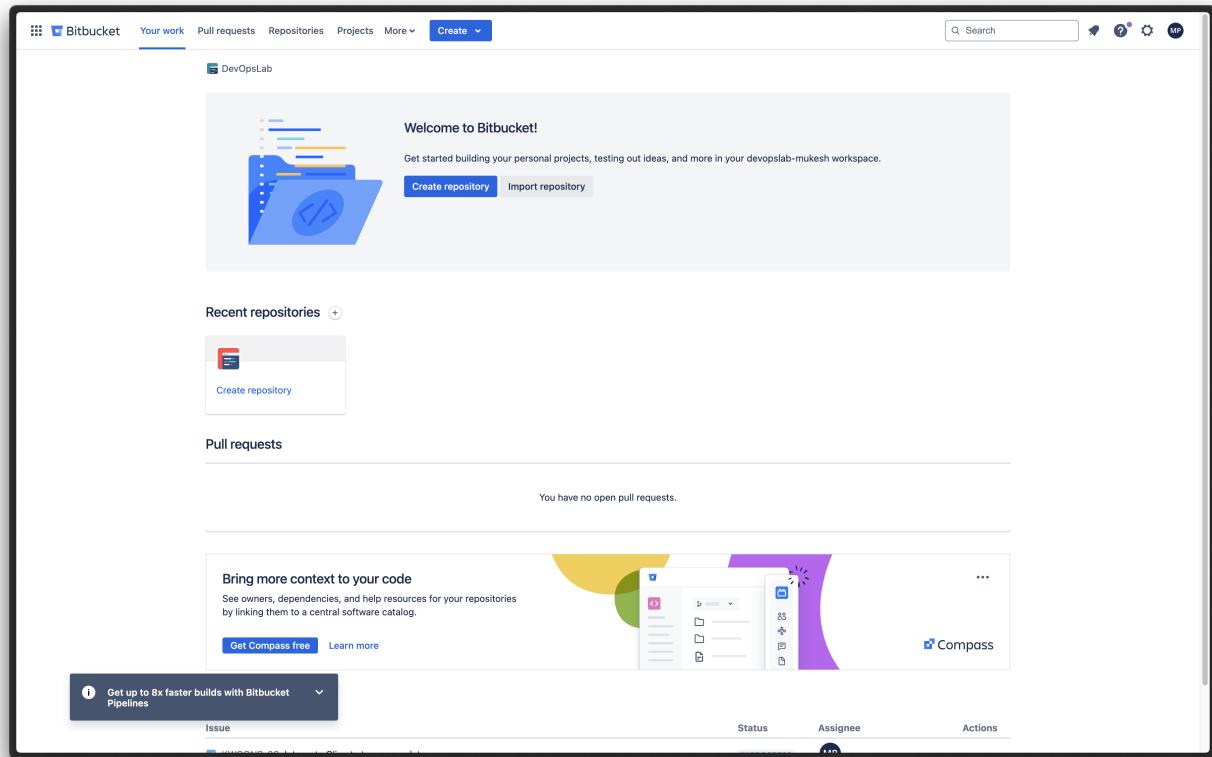
# DevOps Lab - Mukesh T P

## Exercise 3

### Implement Bitbucket Operations using Git

#### 1. Creating a Repository

1. Go to your Bitbucket account.



## 2. Click on the **Repositories** tab in the sidebar.

The screenshot shows the Bitbucket web interface. At the top, there is a navigation bar with links: 'Your work', 'Pull requests', 'Repositories' (which is underlined in blue, indicating it is the active tab), 'Projects', 'More', and a 'Create' button. To the right of the navigation bar is a search bar and some user-specific icons. Below the navigation bar, the page title 'DevOpsLab' and 'Repositories' is displayed. There is a search bar labeled 'Search repositories' with a magnifying glass icon. Below the search bar are filters for 'Projects', 'Privacy', 'Language', and 'Watching'. On the right side of the header, there is a 'Create repository' button. The main content area is titled 'No repositories' and contains the message 'You don't have access to any recently updated repositories.' Below this message are two buttons: 'Import a repository' and 'Create a repository'. At the bottom of the page, there is a dark banner with the text 'Get up to 8x faster builds with Bitbucket Pipelines' and a dropdown arrow.

## 3. Click on the **Create repository** button.

The screenshot shows the 'Create a new repository' dialog box. At the top left is a back arrow. The title of the dialog is 'Create a new repository' with a 'Import repository' link to its right. Inside the dialog, there are several input fields and settings:

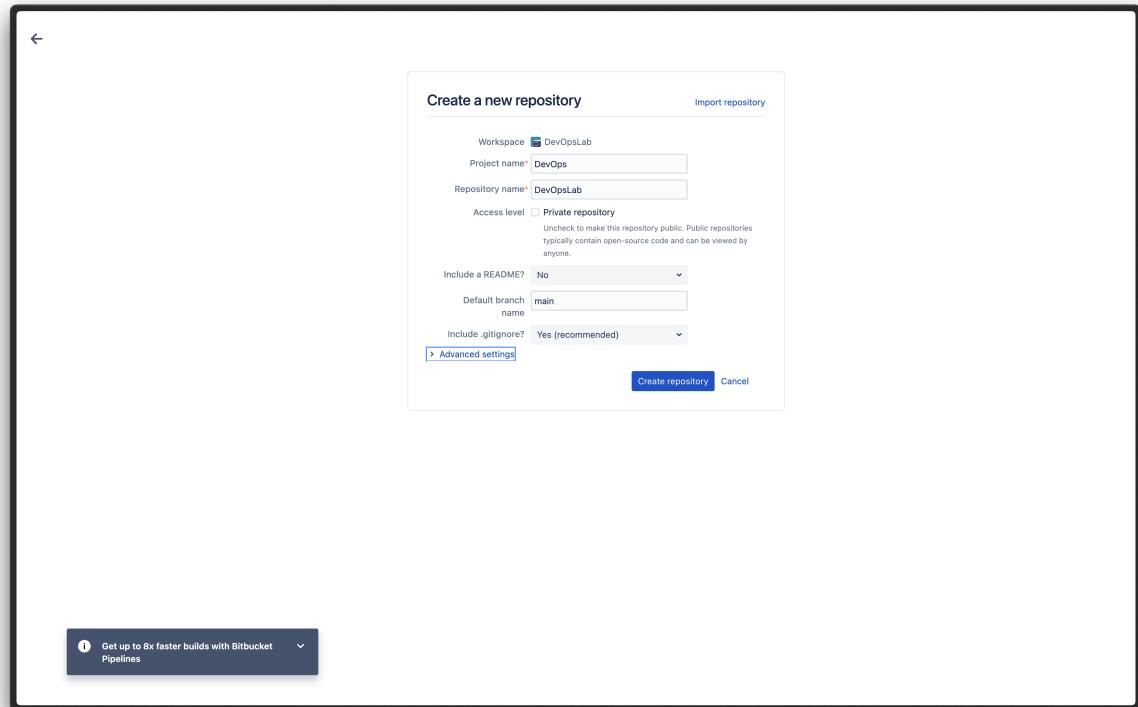
- 'Workspace' dropdown set to 'DevOpsLab'
- 'Project name\*' input field (empty)
- 'Repository name\*' input field (empty)
- 'Access level' dropdown with 'Private repository' selected (indicated by a checked checkbox)
- A tooltip for 'Access level' explains that unchecking makes the repository public, which typically contains open-source code and can be accessed by anyone.
- 'Include a README?' dropdown set to 'No'
- 'Default branch name' input field containing 'e.g., 'main''
- 'Include .gitignore?' dropdown set to 'Yes (recommended)'
- A 'Advanced settings' link
- At the bottom right are 'Create repository' and 'Cancel' buttons

At the very bottom of the dialog is a dark banner with the same 'Get up to 8x faster builds with Bitbucket Pipelines' text and dropdown arrow as the previous screenshot.

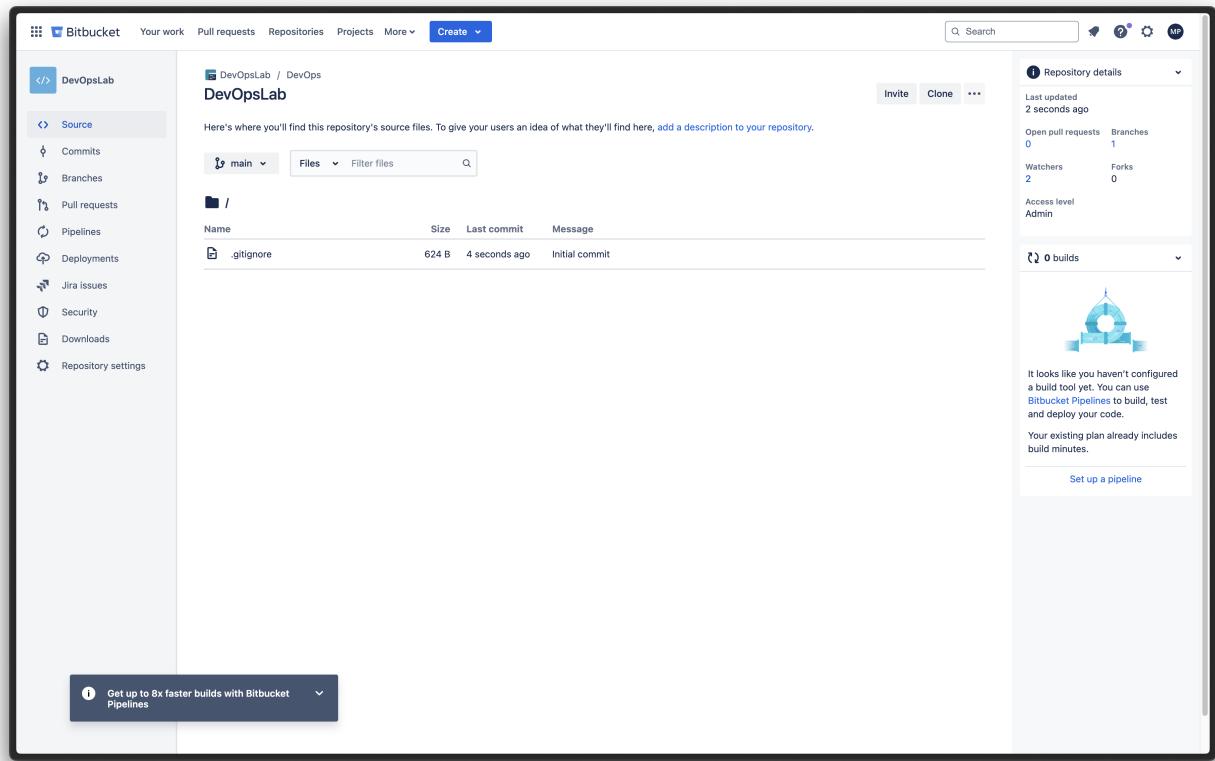
## 4. Fill in the repository details:

- Repository name: example-repo

- Access level: Choose between Private or Public.

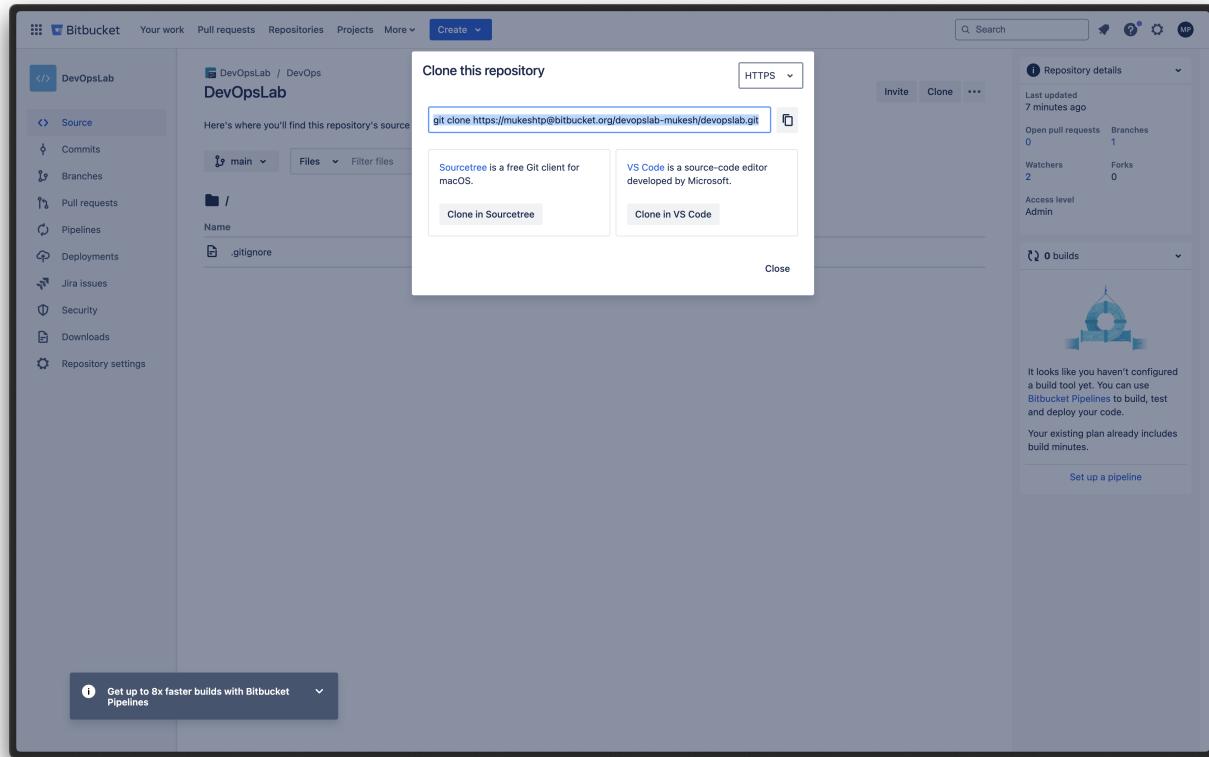


## 5. Click on the **Create repository** button.

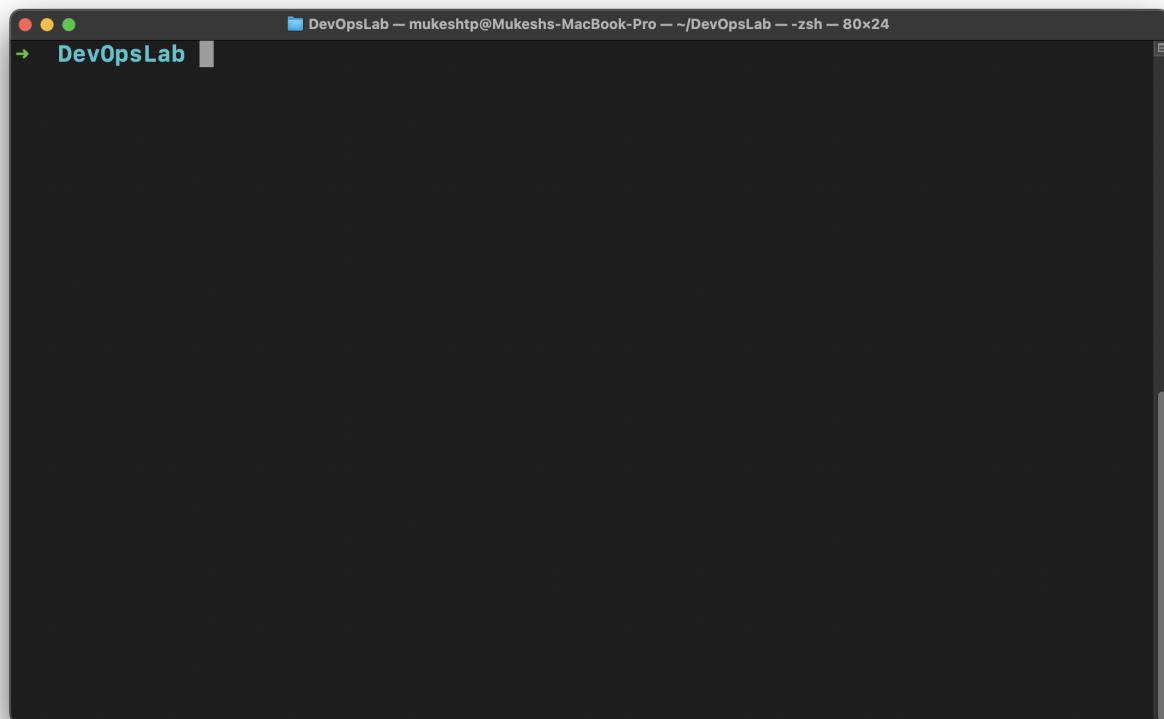


## 2. Cloning a Repository

1. In your Bitbucket repository, find the **Clone** button and copy the repository URL.

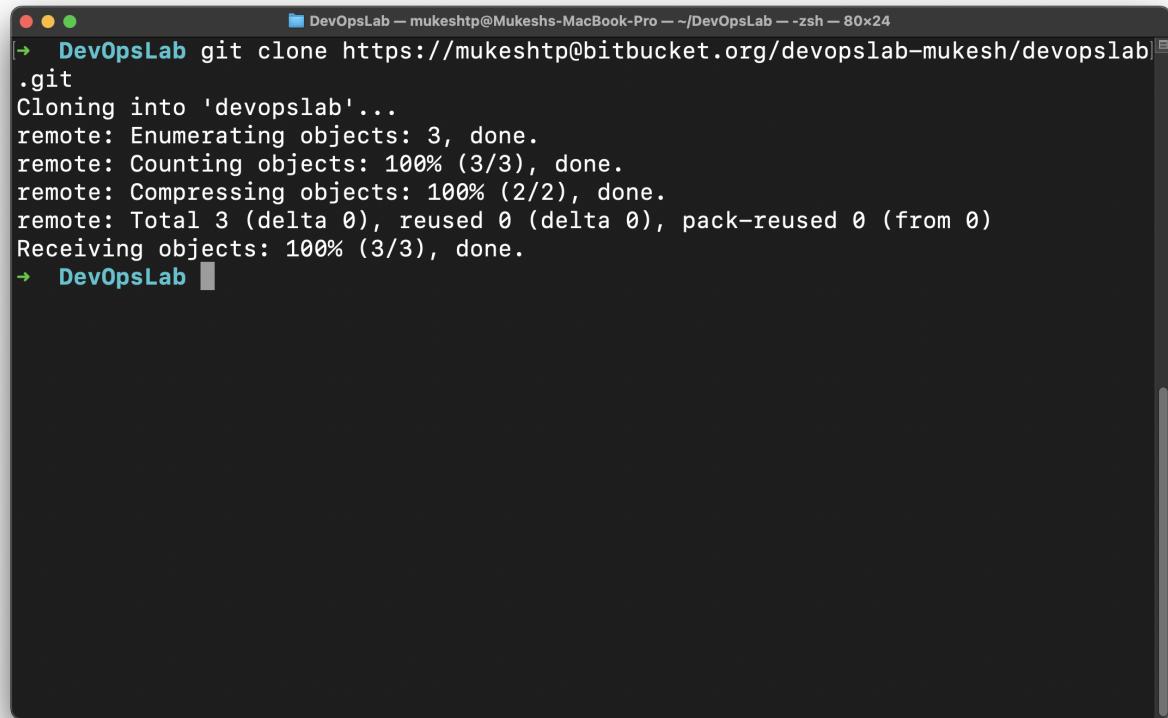


2. Open your terminal.



3. Clone the repository using the following command:

```
git clone https://mukeshtp@bitbucket.org/devopslab-mukesh/devopslab.git
```

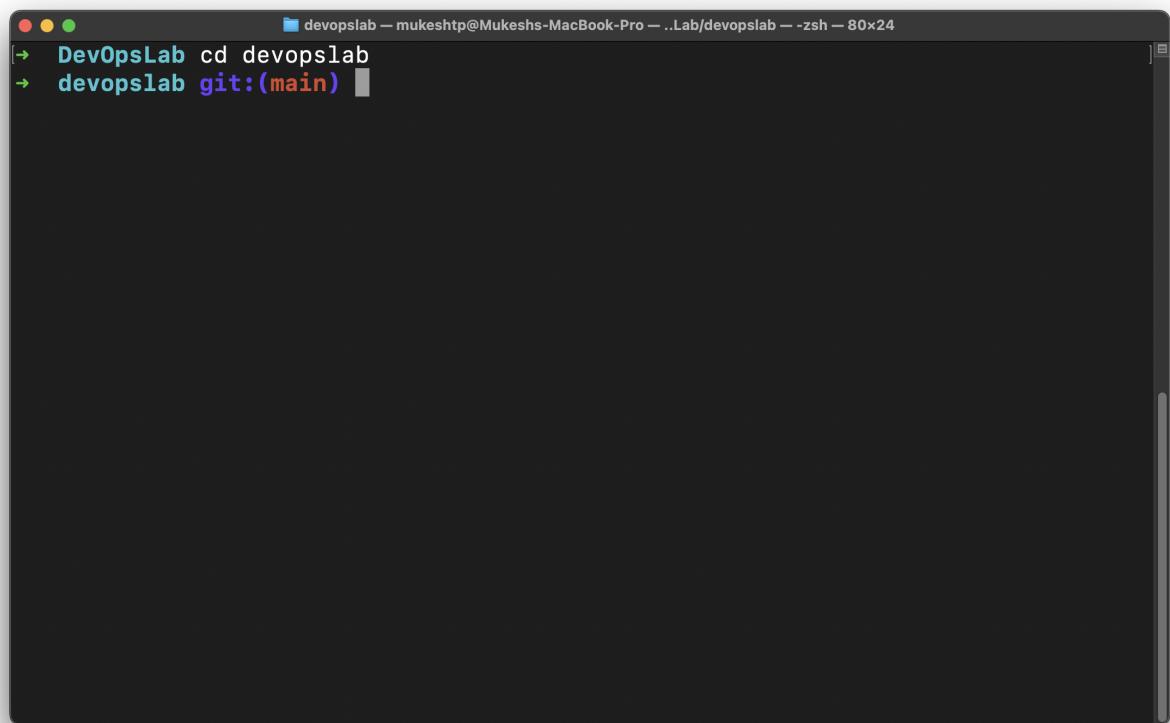


```
DevOpsLab git clone https://mukeshtp@bitbucket.org/devopslab-mukesh/devopslab|.git
Cloning into 'devopslab'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
→ DevOpsLab
```

### 3. Making Changes and Creating a Branch

1. Navigate into the cloned repository:

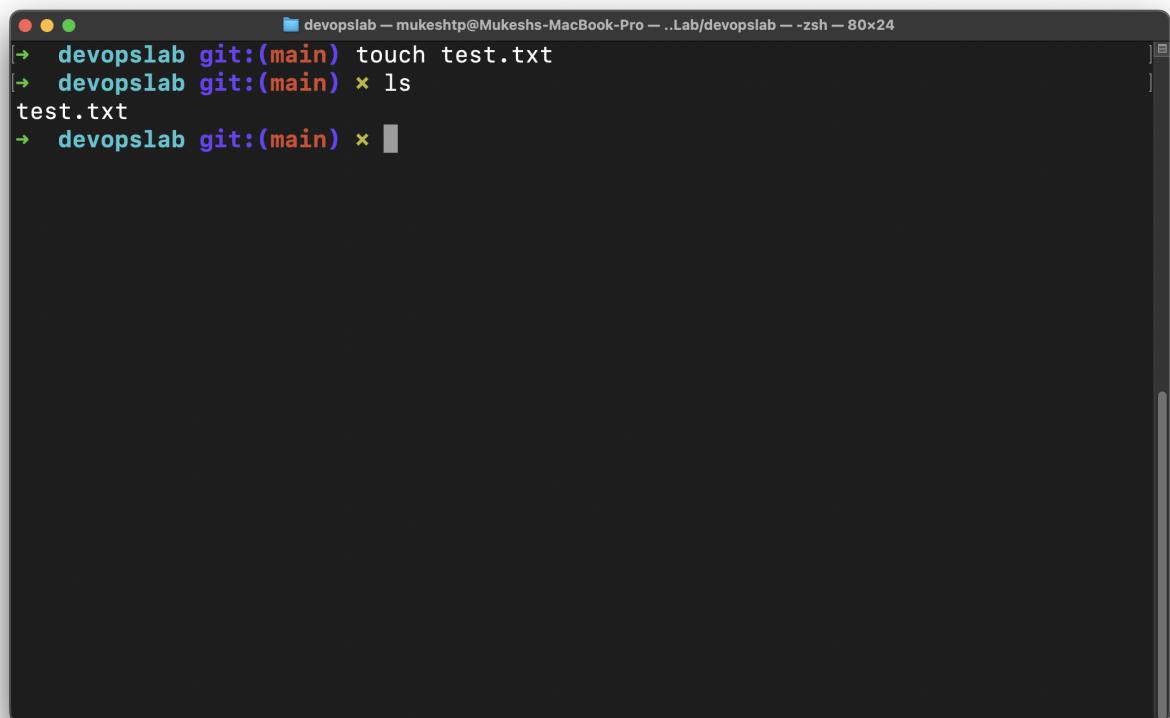
```
cd example-repo
```



```
devopslab -- mukeshtp@Mukeshs-MacBook-Pro -- ..Lab/devopslab -- zsh -- 80x24
[→ DevOpsLab cd devopslab
[→ devopslab git:(main)
```

2. Create a new text file named `test.txt`:

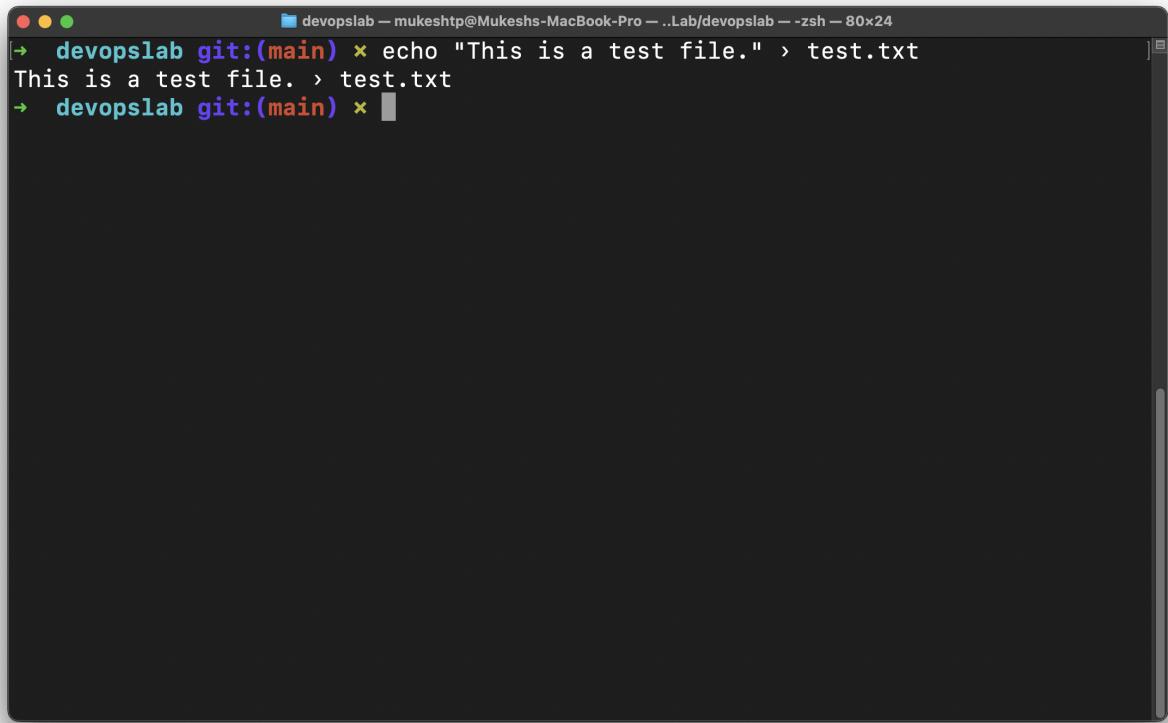
```
touch test.txt
```



```
devopslab -- mukeshtp@Mukeshs-MacBook-Pro -- ..Lab/devopslab -- zsh -- 80x24
[→ devopslab git:(main) touch test.txt
[→ devopslab git:(main) × ls
test.txt
[→ devopslab git:(main) ×
```

3. Add some content to the `test.txt` file.

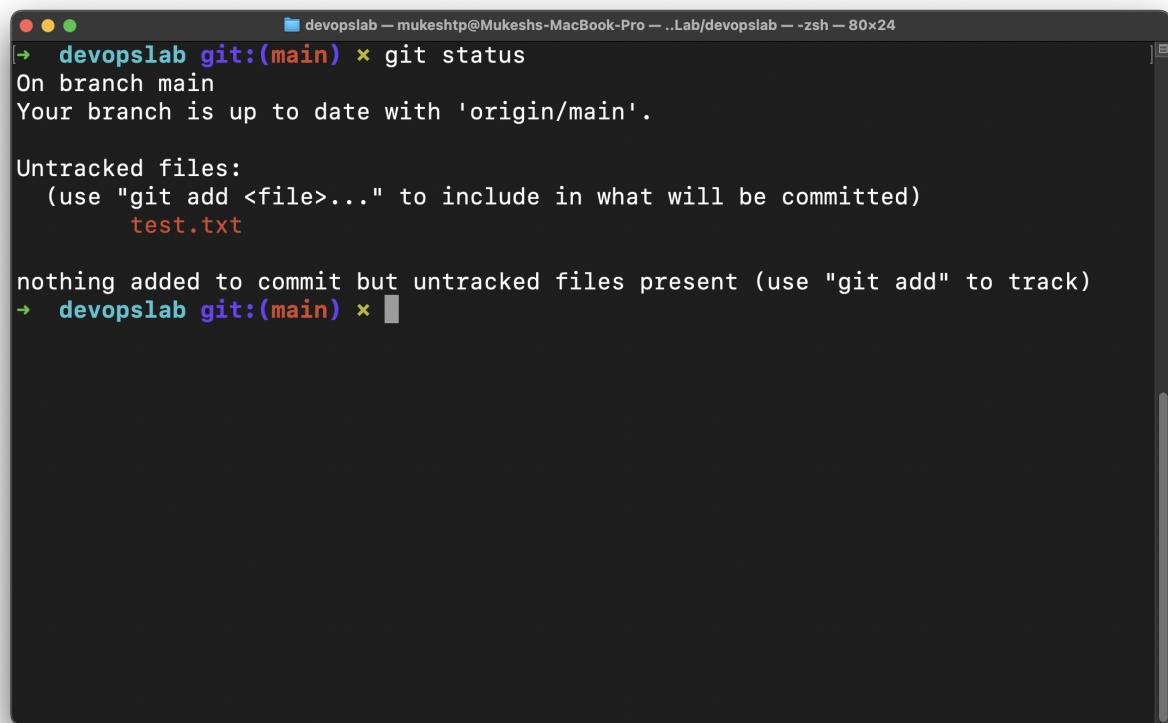
```
echo "This is a test file." > test.txt
```



A screenshot of a macOS terminal window titled 'devopslab'. The window shows the command 'echo "This is a test file." > test.txt' being run and its output. The output shows the text 'This is a test file.' followed by an arrow and 'test.txt'. The terminal has a dark theme with light-colored text.

#### 4. Check the status of the repository:

```
git status
```



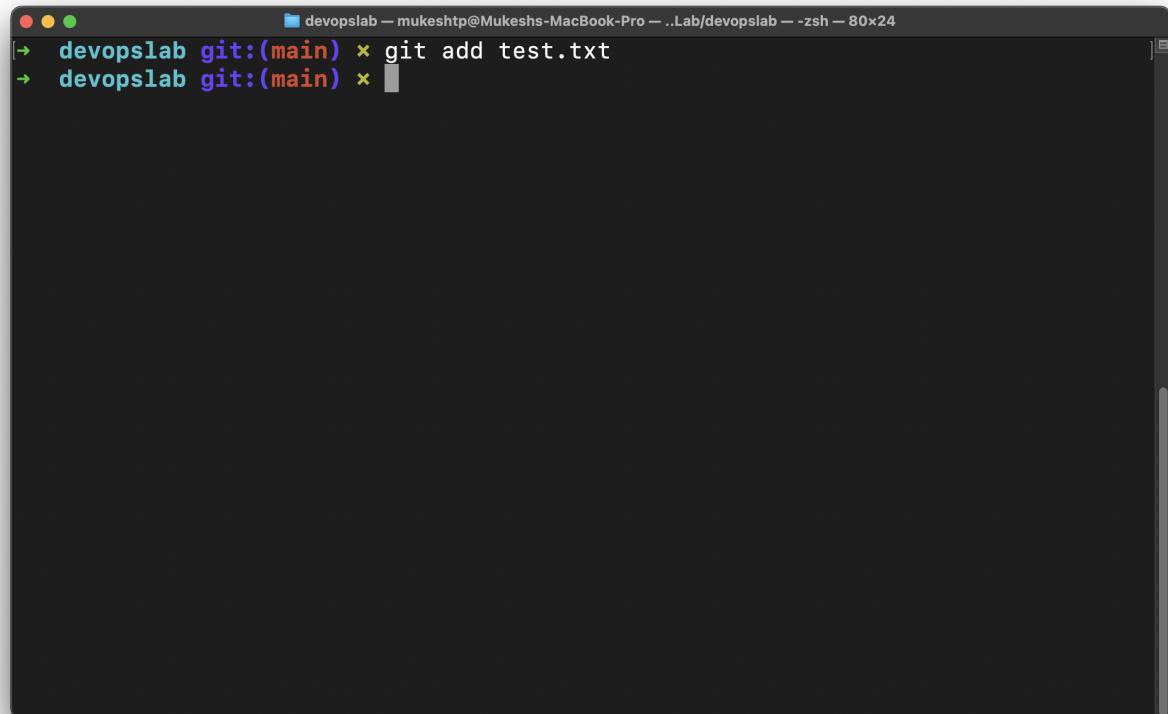
```
devopslab git:(main) ✘ git status
On branch main
Your branch is up to date with 'origin/main'.

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

nothing added to commit but untracked files present (use "git add" to track)
→ devopslab git:(main) ✘
```

5. Stage the changes for commit:

```
git add test.txt
```



```
devopslab git:(main) ✘ git add test.txt
→ devopslab git:(main) ✘
```

6. Commit the changes with a descriptive message:

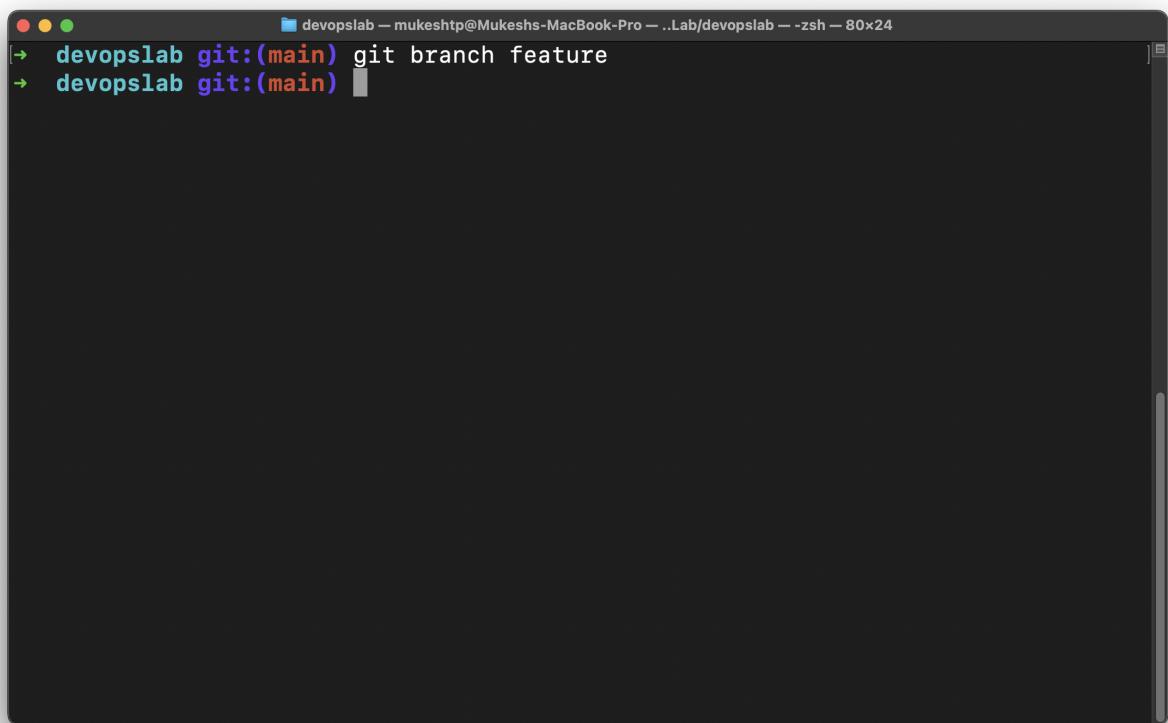
```
git commit -m "Edited test.txt"
```



A screenshot of a macOS terminal window titled "devopslab". The window shows the command "git commit -m \"Edited test.txt\"". The output indicates that one file was changed, with 0 insertions and 0 deletions, and a new file named "test.txt" was created with mode 100644. The prompt "devopslab git:(main)" is visible at the bottom.

7. Create a new branch named **feature**:

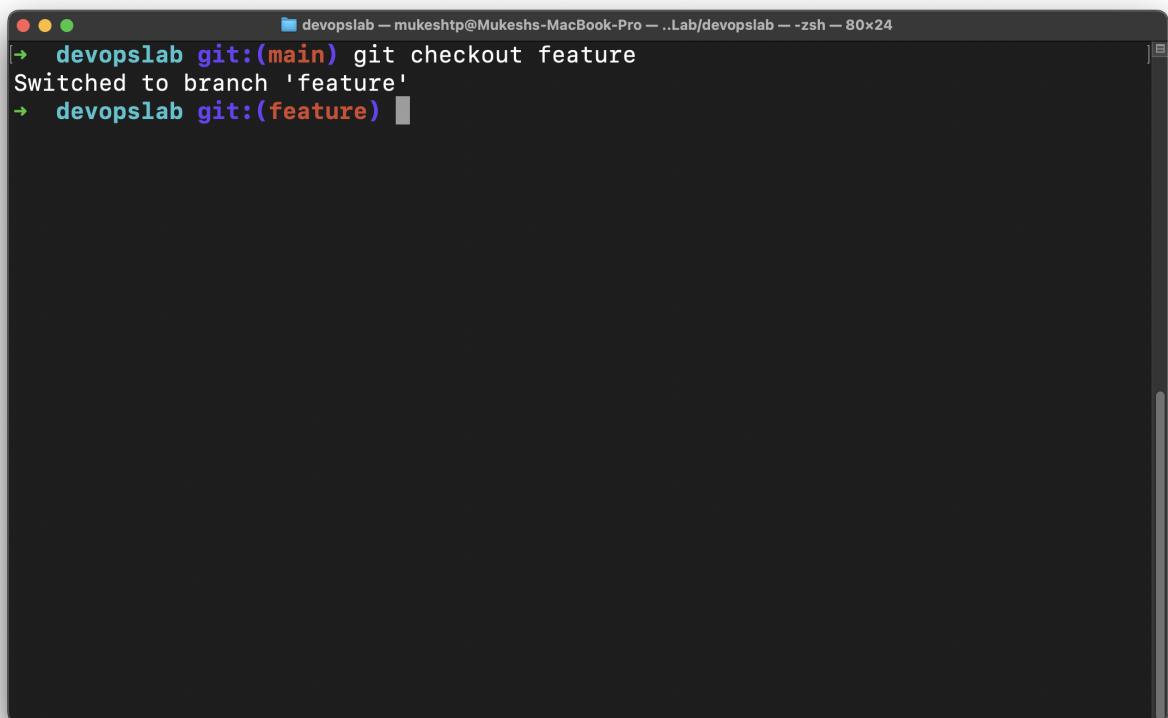
```
git branch feature
```



```
devopslab git:(main) git branch feature
→ devopslab git:(main)
```

8. Switch to the **feature** branch:

```
git checkout feature
```

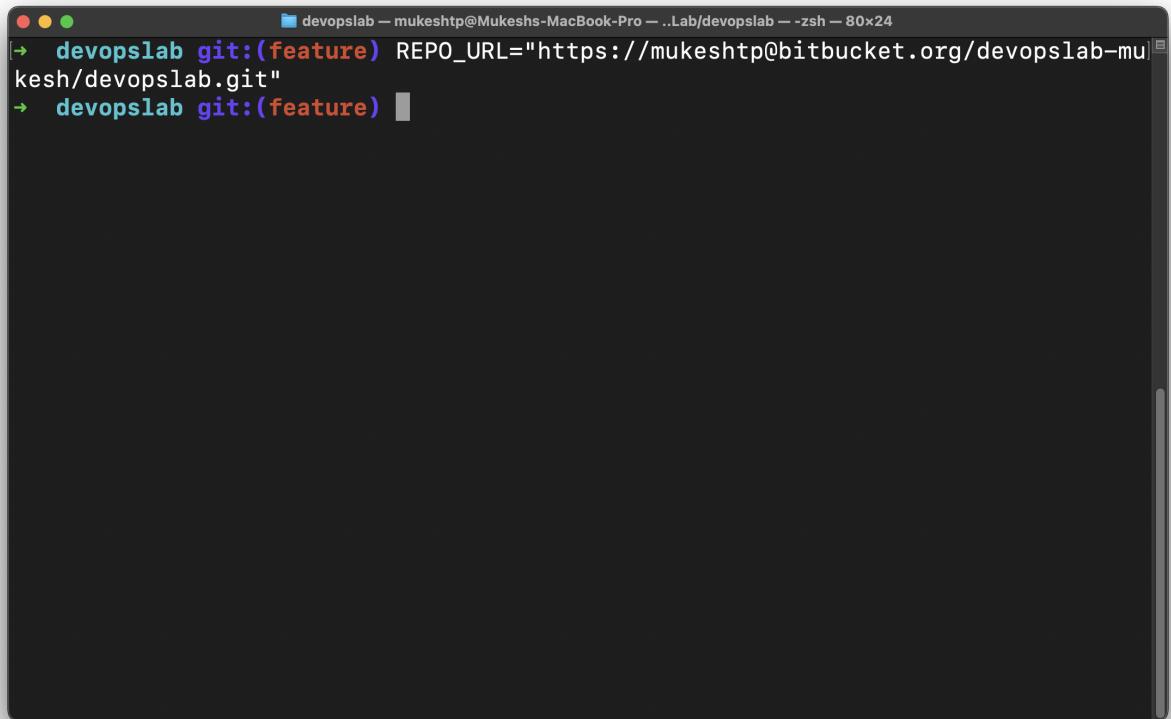


```
devopslab git:(main) git checkout feature
Switched to branch 'feature'
→ devopslab git:(feature)
```

4. Pushing Changes to Bitbucket

1. Add the repository URL in a variable:

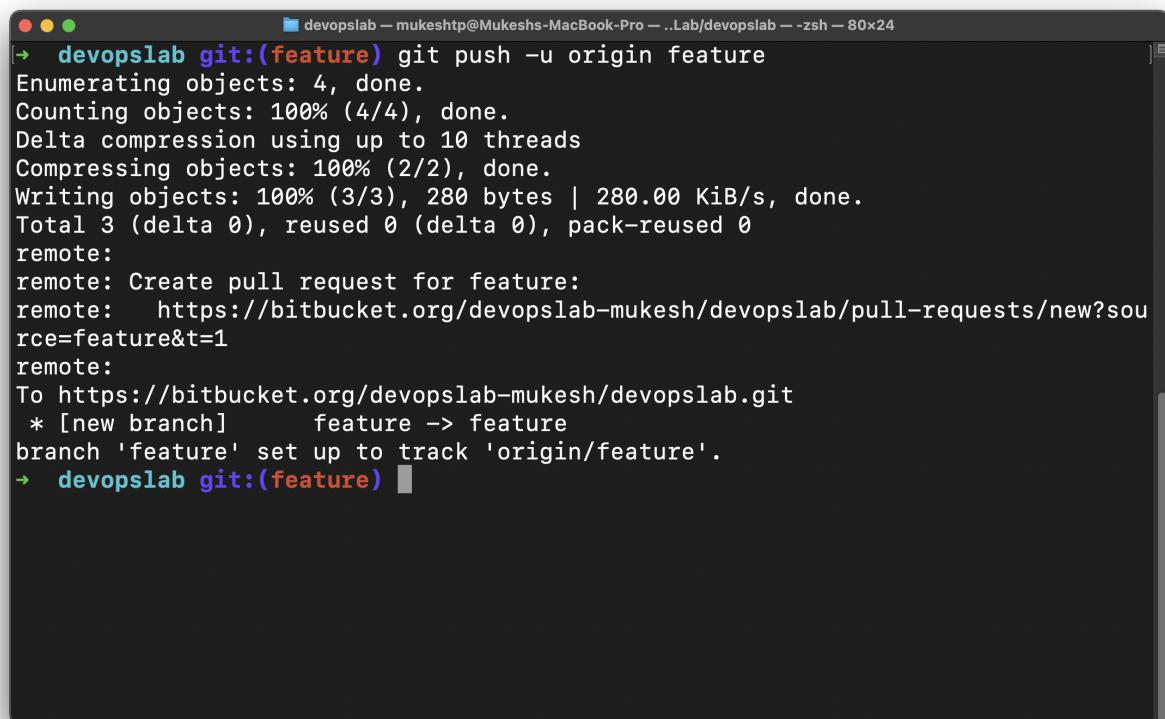
```
REPO_URL=<repository_url>
```



A screenshot of a macOS terminal window titled "devopslab". The window shows the command "git:(feature)" followed by "REPO\_URL='https://mukeshtp@bitbucket.org/devopslab-mukesh/devopslab.git'". The terminal has a dark theme with light-colored text.

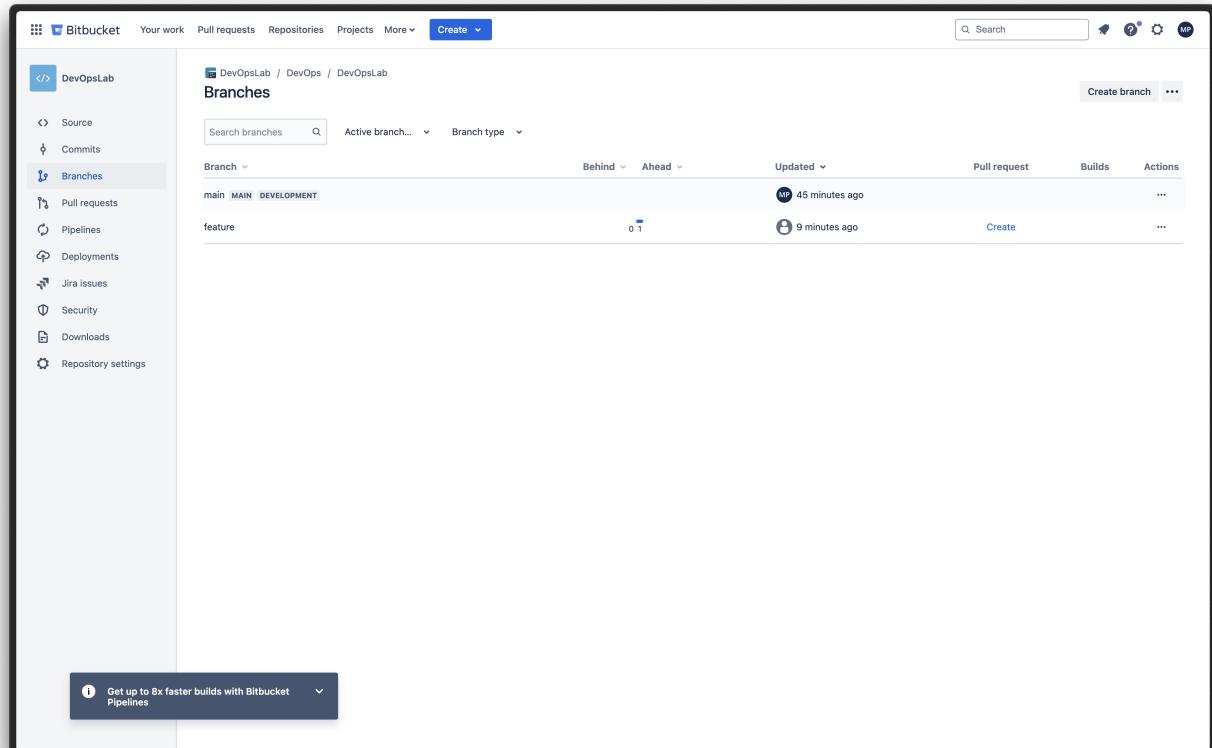
2. Push the **feature** branch to Bitbucket:

```
git push -u origin feature
```



```
devopslab git:(feature) git push -u origin feature
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 10 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 280 bytes | 280.00 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/devopslab-mukesh/devopslab/pull-requests/new?source=feature&t=1
remote:
To https://bitbucket.org/devopslab-mukesh/devopslab.git
 * [new branch]      feature -> feature
branch 'feature' set up to track 'origin/feature'.
→ devopslab git:(feature)
```

3. Check your Bitbucket repository to confirm that the new branch **feature** is available.



Bitbucket

Your work Pull requests Repositories Projects More Create

Search branches

Branches

Create branch

Branches

Search branches Active branch... Branch type

Branch	Behind	Ahead	Updated	Pull request	Builds	Actions
main MAIN DEVELOPMENT	0	1	45 minutes ago			...
feature	0	1	9 minutes ago	Create		...

Get up to 8x faster builds with Bitbucket Pipelines

## 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**.

The screenshot shows the Bitbucket repository page for 'DevOpsLab / DevOps / DevOpsLab'. The left sidebar has 'Source' selected. The main area shows the 'main' branch with one file, '.gitignore', listed. A modal at the bottom left encourages using Bitbucket Pipelines.

The screenshot shows the 'Create a Pull Request' page. The 'Source Branch' is set to 'feature' and the 'Destination Branch' is set to 'main'. The 'Title' field contains 'Edited test.txt'. The 'Description' field is empty. The 'Create pull request' button is visible at the bottom right. A modal at the bottom left encourages using Bitbucket Pipelines.

The screenshot shows the Bitbucket interface for a project named 'DevOpsLab'. On the left sidebar, under the 'Pull requests' section, there is a notification for a new pull request titled 'Edited test.txt'. The main page displays the pull request details, including the source branch ('feature') and target branch ('main'), both of which are currently open. There is a single commit listed. At the top right, there are buttons for 'Approve' and 'Merge'. Below the commit, there is a 'Description' field with placeholder text 'Add a description...', a 'Attachments' section indicating none, and an 'Activity' feed showing a recent update from 'Mukesh T P' opening the pull request. To the right, there is a section for 'Builds' (0 builds), a 'Pipelines' icon, and a note about configuring a build tool. Below that is a 'Merge checks' section with a diagram showing a red circle connected to a blue circle, which then connects to a red square, and finally to a black triangle. A note says 'It looks like you don't have merge checks set up in your target branch yet.' There are also links to 'Set up pipeline', 'Set up merge checks', and 'Set up custom checks'.

## 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.

The screenshot shows the Bitbucket interface for a repository named 'DevOpsLab'. A single pull request titled 'Edited test.txt' is listed under the 'Pull requests' tab. The pull request has been created 2 minutes ago and last updated 2 seconds ago. It is currently in the 'OPEN' state. The source branch is 'feature' and the destination branch is 'main'. The pull request has 1 file changed and 1 commit. There is a description field where 'Add a description...' is written. The activity feed shows three recent events: 'Mukesh T P APPROVED the pull request now', 'Mukesh T P edited the title: Edited-test.txt Edited test.txt 1 second ago', and 'Mukesh T P OPENED the pull request 2 minutes ago'. On the right side, there are sections for 'Builds' (0 builds), 'Pipelines' (a blue circular icon), and 'Merge' buttons. A message box at the bottom left informs about new keyboard shortcuts and syntax highlighting.

This screenshot shows the 'Merge pull request' dialog box overlaid on the Bitbucket interface. The dialog box contains fields for 'Source' (set to 'feature'), 'Destination' (set to 'main'), and a 'Commit message' field containing 'Merged in feature (pull request #1)'. Below the commit message is a dropdown for 'Merge strategy' set to 'Merge commit'. At the bottom of the dialog are 'Merge' and 'Cancel' buttons. The background shows the same repository and pull request details as the first screenshot, with the activity feed and sidebar visible.

The screenshot shows the Bitbucket interface for a pull request titled "Edited test.txt". The pull request has been merged from the "feature" branch into the "main" branch. The status is "MERGED". The activity feed shows several events: "Mukesh T P MERGED the pull request 12 seconds ago", "Mukesh T P APPROVED the pull request 34 seconds ago", "Mukesh T P edited the title: Edited-test.txt Edited test.txt 36 seconds ago", and "Mukesh T P OPENED the pull request 2 minutes ago". There are 0 attachments and 0 builds. A note indicates that no merge checks are set up in the target branch, suggesting the use of Bitbucket Pipelines for CI/CD.

## 6. Syncing Changes

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

```
git checkout main  
git pull origin main
```

```
devopslab git:(feature) 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)
→ devopslab git:(main)
```

```
devopslab git:(main) 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), 233 bytes | 116.00 KiB/s, done.
From https://bitbucket.org/devopslab-mukesh/devopslab
 * branch            main      -> FETCH_HEAD
   5565bd9..62504bc  main      -> origin/main
Updating fc91542..62504bc
Fast-forward
→ devopslab git:(main)
```