

# GITLAB

**NOTE: STEP-1, STEP-2 and STEP-3 are one time process. no need to do everytime**

## STEP-1: Global Configuration (One Time Process)

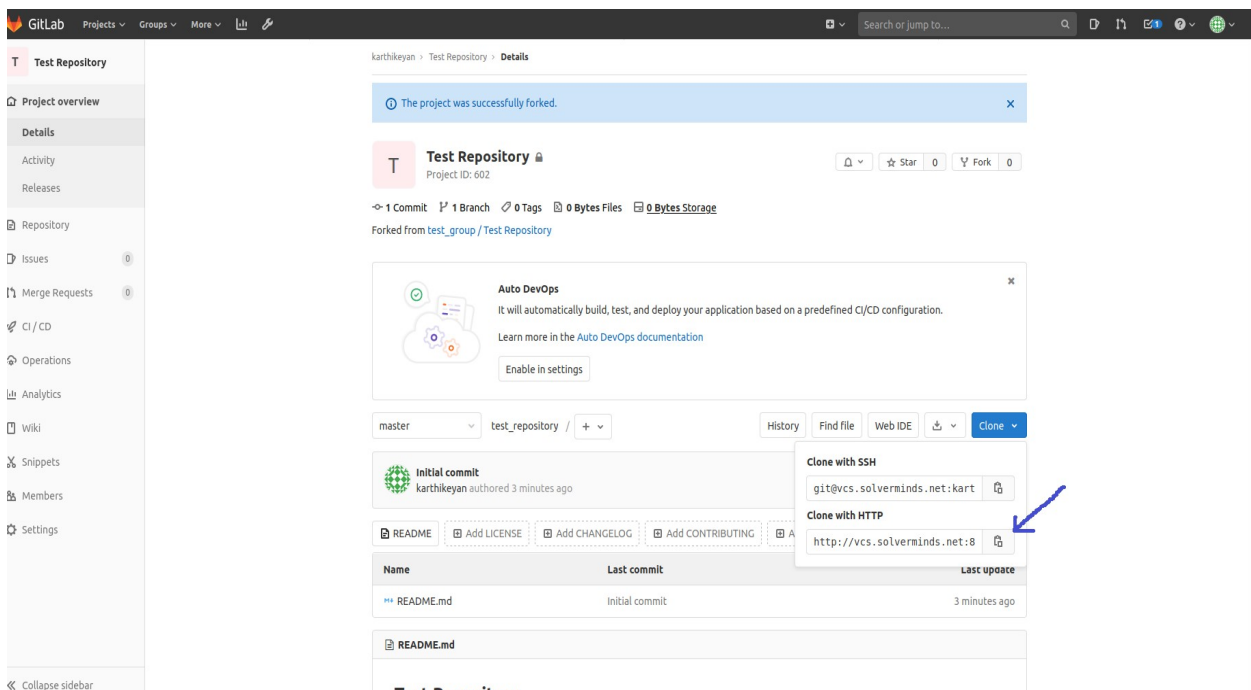
Go to terminal , Make a new directory and check-in to the created directory. Then execute the below commands to configure username and email.

```
# git config --global user.name "USERNAME"  
# git config --global user.email "EMAIL_ADDRESS"
```

```
svmadmin@svmadmin-X556UQK:~/git/test_project/test_repository$ git config --global user.name "karthi"  
svmadmin@svmadmin-X556UQK:~/git/test_project/test_repository$ git config --global user.email "karthikeyan@solverminds.com"
```

## STEP-2: Clone the repository into your local system (One Time Process)

Click the **clone** button and copy the http url,



Open the terminal and execute the below command

```
# git clone "<Copied_http_url>"
```

```
svmadmin@svmadmin-X556UQK:~/git/test_project/test_repository$ git clone http://vcs.solverminds.net:8089/karthi/test_repository.git  
Cloning into 'test_repository'...  
Username for 'http://vcs.solverminds.net:8089': karthi  
Password for 'http://karthi@vcs.solverminds.net:8089':  
remote: Enumerating objects: 3, done.  
remote: Counting objects: 100% (3/3), done.  
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0  
Unpacking objects: 100% (3/3), done.
```

### STEP-3: Create branch and checkout into the created branch (One Time Process)

```
# cd <CLONED_FOLDER>          (Example : Edge_Documentation)
# git branch <BRANCH_NAME>    ( Create a branch)
# git checkout <BRANCH_NAME>  (switch into branch)
```

```
karthi@SOLVERLAPT510:~/test_repository$ git branch new
karthi@SOLVERLAPT510:~/test_repository$
```

```
svmadmin@svmadmin-X556UQK:~/git/test_project/test_repository$ git checkout new
Switched to branch 'new'
```

### STEP-4: Fetching and Merging

Before you make your change in you local repo, first you have to pull the latest commits from Master Source into your local (We always should work on the latest source code).

```
# git pull origin master
```

```
karthi@SOLVERLAPT510:~/test_repository$ git pull origin master
Username for 'http://vcs.solverminds.net:8089': karthi
Password for 'http://karthi@vcs.solverminds.net:8089':
From http://vcs.solverminds.net:8089/test_group/test_repository
 * branch                master      -> FETCH_HEAD
Already up to date.
karthi@SOLVERLAPT510:~/test_repository$
```

### STEP-5: Develop and Push the code

Make your changes in your local repository and push it into the master source through MergeRequest. (Example : adding a file, edit existing files)

```
# git add .
# git commit -m "COMMIT_MESSAGE"
```

```
svmadmin@svmadmin-X556UQK:~/git/test_project/test_repository$ git add .
warning: adding embedded git repository: test_repository
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:   git submodule add <url> test_repository
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:   git rm --cached test_repository
hint: See "git help submodule" for more information.
```

```
svmadmin@svmadmin-X556UQK:~/git/test_project/test_repository$ git commit -m "test_file_created"
[new b767975] test_file_created
2 files changed, 2 insertions(+)
create mode 100644 test_file
create mode 160000 test_repository
```

```
# git push origin <BRANCH_NAME>:<BRANCH_NAME>
```

```
svmadmin@svmadmin-X556UQK:~/git/test_project/test_repository$ git push origin new:new
Username for 'http://vcs.solverminds.net:8089': karthi
Password for 'http://karthi@vcs.solverminds.net:8089':
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 337 bytes | 337.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: To create a merge request for new, visit:
remote:   http://vcs.solverminds.net:8089/karthi/test_repository/-/merge_requests/new?merge_request%5Bsource_branch%5D=new
remote:
To http://vcs.solverminds.net:8089/karthi/test_repository.git
* [new branch]      new -> new
```

After executing the push command, you will see remote URL. **Copy that remote URL and paste it on the browser.** Then you will get a merge request creation page. Please refer the above screenshot.

The screenshot shows the 'New Merge Request' page in GitLab. The page is titled 'New Merge Request' and shows the source branch 'karthi/test\_repository:new' being merged into the target branch 'test\_group/test\_repository:master'. The title of the merge request is 'test\_file\_created'. The description field is empty. The assignee is 'Unassigned' and there is an 'Assign to me' button. The milestone is 'Milestone' and the labels are 'Labels'. The 'Merge options' section has 'Delete source branch when merge request is accepted' checked and 'Squash commits when merge request is accepted' unchecked. The 'Contribution' section has 'Allow commits from members who can merge to the target branch' checked. A blue arrow points to the 'Assign to me' button, and another blue arrow points to the 'Delete source branch' checkbox.

In the above figure, choose assign and Kindly untick the “Delete source branch when merge request is accepted”

Finally, click “**Submit Merge Request**”

**Note:** if you have an write access in this repository then please merge it.

After MergeRequest approved. Please go to the “**Repository Home page**” you will find the files in the repository.

The screenshot shows the GitLab interface for a repository named "Test Repository". The left sidebar contains navigation links: Project overview, Details, Activity, Releases, Repository, Issues (0), Merge Requests (0), CI/CD, Operations, Analytics, Wiki, Snippets, Members, and Settings. The main content area displays the repository details, including the project ID (601), commit count (3), branch count (1), tag count (0), file count (225 KB), and storage usage (225 KB). A banner for Auto DevOps is visible, stating it will automatically build, test, and deploy applications based on a predefined CI/CD configuration. Below the banner, a merge commit is shown: "Merge branch 'new' into 'master'" by karthikeyan, authored 4 minutes ago, with commit hash b42e472b. A table lists the repository's files and their last commit details.

Name	Last commit	Last update
test_repository @ 4029c0f8	test_file_created	7 minutes ago
README.md	Initial commit	28 minutes ago
test_file	test_file_created	7 minutes ago

Below the table, there is a section for the README.md file, titled "Test Repository".