

## Git assignment

- First of all clone repository and create index.html file in it manually and push that file to gitlab in main branch.

```
vaghaaks@L10-VAGHAAKS-1 MINGW64 ~  
$ git clone "https://gitlab.com/roima_2025_intern/gitlab_assignment.git"  
Cloning into 'gitlab_assignment'...  
info: please complete authentication in your browser...  
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.
```

```
vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/gitlab_assignment (main)  
$ git branch  
* main
```

```
vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/gitlab_assignment (main)  
$ touch index.html
```

```
vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/gitlab_assignment (main)  
$ ls  
README.md  index.html
```

```
vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/gitlab_assignment (main)  
$ git add .
```

```
vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/gitlab_assignment (main)  
$ git commit -m "Add index.html file"  
[main 0117f4e] Add index.html file  
Committer: Vaghasiya <akshay.vaghasiya@dialogos.fi>  
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:
```

```
vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/gitlab_assignment (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 22 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 320 bytes | 160.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://gitlab.com/roima_2025_intern/gitlab_assignment.git
abfe11f..0117f4e  main -> main
```

- Make some changes in index.html file in personal branch and push that code to remote repository.

```
<html>

<head>

</head>

<body>

<p>dishant</p>
<p>prashant</p>
<p>nikhil</p>


</body>
</html>
```

```
vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/gitlab_assignment (akshay_branch)
$ git add .

vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/gitlab_assignment (akshay_branch)
$ git commit -m "add 3 intern name into index.html"
```

- Create pull request and see that conflict occur.

## add 3 intern name into index.html

 Open Akshay-vaghasiya requested to merge `akshay_branch` into `main` 1 minute ago


Overview 0 Commits - Pipelines 0 Changes 1



8  [Approve](#) Approval is optional 



**Merge blocked:** 1 check failed

 Merge conflicts must be resolved.

[Resolve locally](#)

[Resolve conflicts](#)

### Merge details

- The source branch is [2 commits behind](#) the target branch.
- 1 commit and 1 merge commit will be added to `main`.
- Source branch will be deleted.

- To solve this conflict I have done some changes in code means I have take both code.

# add 3 intern name into index.html

Akshay-vaghasiya requested to merge `akshay_branch` into `main` 6 minutes ago

Showing **1 conflict**

index.html

```
4
5     </head>
6
7     <body>
8
9     <p>dishant</p>
10    <p>prashant</p>
11    <p>nikhil</p>
12
13    <p>zoha</p>
14    <p>ramesh</p>
15    <p>raju</p>
16    <p>suresh</p>
17    <p>karishma</p>
18
19    </body>
20    </html>
21
```

- After push you can see that conflict is solve.



Merged by  [Akshay-vaghasiya](#) 2 minutes ago

[Revert](#)

[Cherry-pick](#)

[Delete source branch](#)

#### Merge details


- Changes merged into main with [7513aad1](#).
- Did not delete the source branch.

## Activity

All activity ▾



- **Akshay-vaghasiya** added 4 commits 3 minutes ago
  - [93cd9ab5...d20cac96](#) - 3 commits from branch `main`
  - [defe84bc](#) - Merge branch 'main' into 'akshay\_branch'

[Compare with previous version](#)
-  **Akshay-vaghasiya** merged 2 minutes ago
- **Akshay-vaghasiya** mentioned in commit [7513aad1](#) 2 minutes ago

- For second conflict I have create one project because I get some error in first project and create one branch and modify code and also push that code in remote repository.

```
vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/assignment (akshay2)
$ git checkout -b "akshay_branch1"
Switched to a new branch 'akshay_branch1'

vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/assignment (akshay_branch1)
$ git add .

vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/assignment (akshay_branch1)
$ git status
On branch akshay_branch1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.html

vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/assignment (akshay_branch1)
$ git commit -m "update intern details"
[akshay_branch1 45e9c10] update intern details
Committer: Vaghasiya <akshay.vaghasiya@dialogos.fi>
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:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

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

    git commit --amend --reset-author

1 file changed, 5 insertions(+), 7 deletions(-)

vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/assignment (akshay_branch1)
$ git push -u origin akshay_branch1
Enumerating objects: 24, done.
Counting objects: 100% (24/24), done.
Delta compression using up to 22 threads
Compressing objects: 100% (18/18), done.
Writing objects: 100% (20/20), 2.44 KiB | 226.00 KiB/s, done.
Total 20 (delta 5), reused 13 (delta 2), pack-reused 0 (from 0)
remote:
remote: To create a merge request for akshay_branch1, visit:
remote:   https://gitlab.com/roima_2025_intern/assignment/-/merge_requests/new?m
merge_request%5Bsource_branch%5D=akshay_branch1
remote:
To https://gitlab.com/roima_2025_intern/assignment.git
* [new branch]      akshay_branch1 -> akshay_branch1
```

```

vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/assignment (akshay_branch1)
$ git pull origin main
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 18 (delta 9), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (18/18), 2.14 KiB | 24.00 KiB/s, done.
From https://gitlab.com/roima_2025_intern/assignment
 * branch          main          -> FETCH_HEAD
 * b38691e..6b1a4f3 main        -> origin/main
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.

vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/assignment (akshay_branch1|MERGING)
$ git add .

vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/assignment (akshay_branch1|MERGING)
$ git commit -m "merge akshay_branch1 to main"
[akshay_branch1 dcd4752] merge akshay_branch1 to main
Committer: Vaghasiya <akshay.vaghasiya@dialogos.fi>
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:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

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


    git commit --amend --reset-author

vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/assignment (akshay_branch1)
$ git push origin akshay_branch1
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 22 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 364 bytes | 182.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: View merge request for akshay_branch1:
remote:   https://gitlab.com/roima_2025_intern/assignment/-/merge_requests/29
remote:
To https://gitlab.com/roima_2025_intern/assignment.git
   45e9c10..dcd4752 akshay_branch1 -> akshay_branch1

```

- Create one pull request from frontend and see that conflict occur.

## update intern details

 Open Akshay-vaghasiya requested to merge `akshay_branch1` into `main` right now

Overview 0 Commits 1 Pipelines 0 Changes 1



 Looks like there's no pipeline here. ×

GitLab CI/CD can automatically build, test, and deploy your application. It only takes a few minutes to get started, and we can help you create a pipeline configuration file.

[Try out GitLab Pipelines](#)

8  [Approve](#) Approval is optional  ▼

 Merge blocked: 1 check failed ▼

 Merge conflicts must be resolved. [Resolve locally](#) [Resolve conflicts](#)



- Now see that part what I change and what is present in main branch and for resolve this conflict I take some changes from both the files.


index.html

Interactive modeEdit inlineView file @ 45e9c10

<div>@@ -11,44 +11,31 @@</div> <div>11 &lt;p&gt;Ankit    hobbies : read books&lt;/p&gt;</div> <div>12</div> <div>13 &lt;head&gt;</div> <div>&gt;&gt;&gt;&gt;&gt;&gt; 45e9c10d: update intern details (our changes)</div> <div>14</div> <div>15 &lt;title&gt; intern 2025 &lt;/title&gt;</div> <div>16 &lt;/head&gt;</div> <div>17</div> <div>18 &lt;body style="background-color:yellow;"&gt;</div> <div>19</div> <div>20 &lt;p style="color:red;"&gt;manav    hobbies:playing&lt;/p&gt;</div> <div>21 &lt;p style="color:blue;"&gt;raj    hobbies:football&lt;/p&gt;</div> <div>22</div> <div>23 &lt;p&gt;harsh, Hobbies:chess&lt;/p&gt;</div> <div>24 &lt;p&gt;shiv, Hobbies:chess, cricket&lt;/p&gt;</div> <div>25</div> <div>26 &lt;p&gt;Zoha Surti&lt;/p&gt;</div> <div>27 &lt;p&gt;Darshan Patel&lt;/p&gt;</div> <div>28 &lt;p&gt;nikhil vaghasiya&lt;/p&gt;</div>	<div>@@ -11,44 +11,31 @@</div> <div>11 &lt;p&gt;Ankit    hobbies : read books&lt;/p&gt;</div> <div>12</div> <div>13 &lt;head&gt;</div> <div>&lt;&lt;&lt;&lt;&lt;&lt; HEAD: main (their changes)</div> <div>14 &lt;title&gt;</div> <div>15 Intern Details</div> <div>16 &lt;/title&gt;</div> <div>17 &lt;style&gt;</div> <div>18 bodd {</div> <div>19 font-size:12px</div> <div>20 }</div> <div>21 &lt;/style&gt;</div> <div>22 &lt;/head&gt;</div> <div>23</div> <div>24 &lt;body style="background-color:lightblue;"&gt;</div> <div>25</div> <div>26 &lt;p style="color:red;"&gt;manav    hobbies:playing&lt;/p&gt;</div> <div>27 &lt;p style="color:red;"&gt;raghu    hobbies:basketball&lt;/p&gt;</div> <div>28</div> <div>29</div> <div>30 &lt;p&gt;harsh    Hobbies:chess&lt;/p&gt;</div> <div>31</div> <div>32 &lt;p&gt;shiv    Hobbies:footbal&lt;/p&gt;</div> <div>33</div>
--	---

- Now you can see conflict is solve and merge.

## update intern details

 Merged **Akshay-vaghasiya** requested to merge **akshay\_branch1**  into **main** 8 minutes ago

**Overview** 0 Commits 2 Pipelines 0 Changes 1



0



0



Looks like there's no pipeline here.



GitLab CI/CD can automatically build, test, and deploy your application. It only takes a few minutes to get started, and we can help you create a pipeline configuration file.

[Try out GitLab Pipelines](#)

8✓ Approval is optional



 Merged by  **Akshay-vaghasiya** just now

[Revert](#)

[Cherry-pick](#)

### Merge details

- Changes merged into **main** with [790a4ad8](#).
- Deleted the source branch.

- This is the output of what I made changes in above file.

zoha

ramesh

suresh

parimal

Divyang

Akshay Vaghasiya || hobbies : play cricket

Parth Patel || hobbies : listen music

Ankit || hobbies : read books

manav || hobbies:playing

raj || hobbies:football

harsh, Hobbies:chess

shiv, Hobbies:chess, cricket

Zoha Surti

Darshan Patel

nikhil vaghasiya

Divyesh

## Intern and Hobbies

Aryan || Hobbies:chess

Viraj || Hobbies:gaming

- How to delete a branch?

```
vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/gitlab_assignment (main|MERGING)
$ git branch -D akshay_branch
Deleted branch akshay_branch (was 43b4958).

vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/gitlab_assignment (main|MERGING)
$ git push origin --delete akshay_branch
To https://gitlab.com/roima_2025_intern/gitlab_assignment.git
- [deleted]          akshay_branch
```

- What needs to be done when the remote repository declines a push?

- First pull data from the origin.

```
vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/gitlab_assignment (akshay_branch1)
$ git pull origin
remote: Enumerating objects: 42, done.
remote: Counting objects: 100% (42/42), done.
remote: Compressing objects: 100% (34/34), done.
remote: Total 34 (delta 21), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (34/34), 3.51 KiB | 22.00 KiB/s, done.
From https://gitlab.com/roima_2025_intern/gitlab_assignment
* [new branch]      aryan12      -> origin/aryan12
+ 1488286...b7bdfe1 branch_aryan -> origin/branch_aryan (forced update)
* [new branch]      branch_darshan -> origin/branch_darshan
+ ba6841a..78f2ad4 darshan_branch -> origin/darshan_branch
+ 3d1ddd7..3b6ebfa main          -> origin/main
* [new branch]      zoha_branch2 -> origin/zoha_branch2
You asked to pull from the remote 'origin', but did not specify
a branch. Because this is not the default configured remote
for your current branch, you must specify a branch on the command line.
```

- Make changes and push changes

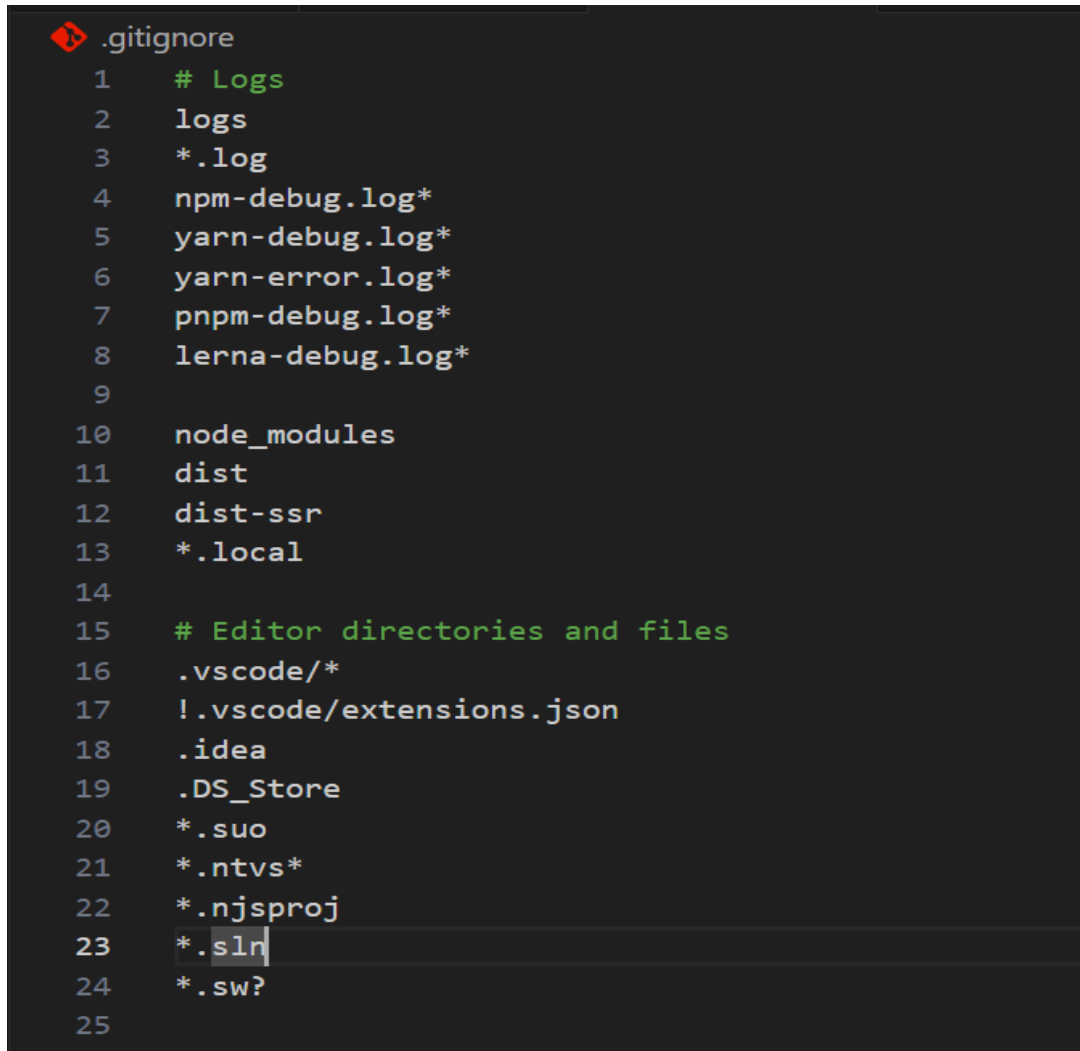
```
vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/gitlab_assignment (akshay_branch1)
$ git push origin
fatal: The current branch akshay_branch1 has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin akshay_branch1

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.
```

- **What is .gitignore?**

- In this file we can specify which file to ignore when we push code to remote repositories.

A screenshot of a code editor showing a .gitignore file. The file contains a list of patterns to ignore, including logs, node\_modules, dist, dist-ssr, \*.local, editor directories like .vscode, .idea, and .DS\_Store, and various IDE-specific files like \*.suo, \*.ntvs\*, \*.njsproj, \*.sln, and \*.sw?. The file is titled ".gitignore" with a red icon. The content is as follows:

```
.gitignore
1  # Logs
2  logs
3  *.log
4  npm-debug.log*
5  yarn-debug.log*
6  yarn-error.log*
7  pnpm-debug.log*
8  lerna-debug.log*
9
10 node_modules
11 dist
12 dist-ssr
13 *.local
14
15 # Editor directories and files
16 .vscode/*
17 !.vscode/extensions.json
18 .idea
19 .DS_Store
20 *.suo
21 *.ntvs*
22 *.njsproj
23 *.sln
24 *.sw?
25
```

- **Git alias what, why is it need?**



- When we want to set shortcut for git commands.

```
vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/gitlab_assignment (main)
$ git config --global alias.lol "log"

vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/gitlab_assignment (main)
$ git lol
commit 3d1ddd791633a4b372bbb4b961cdb78450ff076b (HEAD -> main)
Merge: 8ad43af bf26372
Author: Zoha Surti <zoha.surati@roimaint.com>
Date: Mon Jan 27 11:39:02 2025 +0000
```

- **What is the difference between a git merge and a git rebase?**

- Merge is used for merging branches and preserving history.

 **Merged by**  [Akshay-vaghasiya](#) 2 minutes ago

[Revert](#) [Cherry-pick](#) [Delete source branch](#)


Merge details

- Changes merged into main with [7513aad1](#).
- Did not delete the source branch.

## Activity

All activity ▾



- **Akshay-vaghasiya** added 4 commits 3 minutes ago
  - [93cd9ab5...d20cac96](#) - 3 commits from branch `main`
  - [defe84bc](#) - Merge branch 'main' into 'akshay\_branch'[Compare with previous version](#)
-  **Akshay-vaghasiya** merged 2 minutes ago
- **Akshay-vaghasiya** mentioned in commit [7513aad1](#) 2 minutes ago

- Rebase is used for rewrites commit history

```
vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/assignment2 (main)
$ git pull origin main --rebase
From https://gitlab.com/roima_2025_intern/assignment2
* branch          main          -> FETCH_HEAD
Already up to date.
```

- What is the difference between clone with SSH and HTTP/S.

```
vaghaaks@L10-VAGHAAKS-1 MINGW64 ~
$ git clone "https://gitlab.com/roima_2025_intern/gitlab_assignment.git"
Cloning into 'gitlab_assignment'...
info: please complete authentication in your browser...
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.
```

```
vaghaaks@L10-VAGHAAKS-1 MINGW64 ~
$ git clone "git@gitlab.com:zohasurti/versioncontrolassignment.git"
Cloning into 'versioncontrolassignment'...
remote: Enumerating objects: 29, done.
remote: Counting objects: 100% (16/16), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 29 (delta 7), reused 0 (delta 0), pack-reused 13 (from 1)
Receiving objects: 100% (29/29), 5.68 KiB | 387.00 KiB/s, done.
Resolving deltas: 100% (7/7), done.
```

- What are ssh keys?

```
vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/gitlab_assignment (main)
$ ssh-keygen -t rsa -b 4096 -C "akshayvaghasiya3636@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/vaghaaks/.ssh/id_rsa):
Created directory '/c/Users/vaghaaks/.ssh'.
Enter passphrase for "/c/Users/vaghaaks/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/vaghaaks/.ssh/id_rsa
Your public key has been saved in /c/Users/vaghaaks/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:58u4sxKtH5W9LZmZjzslYemHeqPWeHZfTcNbZf68DUS akshayvaghasiya3636@gmail.com
The key's randomart image is:
+---[RSA 4096]-----+
|
|          .  o
|        o+ .o.
|       .S +o.o o+
|      . .+  +Boo*
|      o.  .+B+E.=
|      o .++ .B=.o+
```

- How to configure ssh keys in git account.

```
vaghaaks@L10-VAGHAAKS-1 MINGW64 ~
$ cd .ssh/

vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/.ssh
$ ls
id_rsa  id_rsa.pub


vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/.ssh
$ eval "$(ssh-agent -s)"
Agent pid 2119

vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/.ssh
$ ssh-add ~/.ssh/id_rsa
Identity added: /c/Users/vaghaaks/.ssh/id_rsa (akshayvaghasiya3636@gmail.com)
```

```
vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/.ssh
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQC1N6R1RPoOyIuuZONAhOVLMO1ljk0WA7gJcX1rXs+0
s+fGLwKS/yFRiRo+L5LvSjklDwwDIgcrQI066SwFbrGrQ6kxkeni9uD73l/5hxEwVV+Aq+vvddM96ZZ
UqByG2uMvT05G3qtqsQLJwvYGci6dvyHmn9niiPA1WQo6I9V4LNRHmt/7wNHRwh1NyR70K1wHrB6JOr
QZzdUQ/a5AUj34SwpjQCASrJtwelzfZltyNuXlhyGF9W2Cp/QZOz0tq7Q0Td/Ytuzl/MFQDznYuTK3hM
wkCKj9IBnw480TYfe5G/WyZA0D1+PCcrLm/YCLp6iryIB0dJFYqzr83m2JGNotV+mORxtRmkgxhQnZSg
VGTasx4i8C8Cd1xUkOPXEdyR/Qu5kbbk62raE9C9BA1L0jxHp0moAXmrc9Vr1uEoqgPdn40ihOgZqvD0c
Mq8uEewUfpdOTpkDBqYyc/txF5rGWLJDanqZDxJ9DhwP6PsrUk9EVfG9oFhkXLD9fFowPhmp2Jbwe5Ux
J+PLPcRiOCSJnHbFGV1CiKAJpctiGr50oU9BvdCJpOGAOSmcSx2Z//z9qjVDPKjntG/q0afLQiK+h1vq
nsihzCK7L1IcfBbx5KFHitAxYqN9t0NfWw57VLCldzPst0hYQe9qTJrys6gkRv8bjY1n2yJMs5dAp1a3
aQ== akshayvaghasiya3636@gmail.com
```

## SSH Keys

SSH keys allow you to establish a secure connection between your computer and GitLab. SSH fingerprints verify that the client is connecting to the correct host. Check the [current instance configuration](#).

Your SSH keys <span>1</span>					<button>Add new key</button>
Title	Key	Usage type	Created	Last used	
<a href="#">akshayvaghasiya3636@gmail.com</a>	 a0:eb:c0:32:3b:09:63:28:93:32:4d:49:1f:e6:b6:2d	Authentication & Signing	9 minutes ago	5 minutes ago	

- SSH doesn't require repetitive authentication for every repository action.

- What is forking? Demonstrate by creating a fork.

- Forking is the process of creating a copy of someone else's repository into your GitHub account. It allows you to make changes without affecting the original repository.



```

vaghaaks@L10-VAGHAAKS-1 MINGW64 ~
$ git clone "https://gitlab.com/Akshay-vaghasiya/versioncontrolassignment.git"
Cloning into 'versioncontrolassignment'...
remote: Enumerating objects: 29, done.
remote: Counting objects: 100% (16/16), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 29 (delta 7), reused 0 (delta 0), pack-reused 13 (from 1)
Receiving objects: 100% (29/29), 5.68 KiB | 969.00 KiB/s, done.
Resolving deltas: 100% (7/7), done.

vaghaaks@L10-VAGHAAKS-1 MINGW64 ~
$ cd versioncontrolassignment/

vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/versioncontrolassignment (main)
$ git remote add upstream https://gitlab.com/zohasurti/versioncontrolassignment.git

vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/versioncontrolassignment (main)
$ git fetch upstream
From https://gitlab.com/zohasurti/versioncontrolassignment
 * [new branch]      aryan_branch -> upstream/aryan_branch
 * [new branch]      darshan_branch -> upstream/darshan_branch
 * [new branch]      main -> upstream/main
 * [new branch]      zoha_branch -> upstream/zoha_branch

```

- Your forked repository is connected to your remote origin. To stay updated with the original repository, you need to add a new remote pointing to the original repository.

- **What is a force push, when should it be used and why would you need to?**

```

vaghaaks@L10-VAGHAAKS-1 MINGW64 ~/versioncontrolassignment (main)
$ git push origin main --force
Everything up-to-date

```

- When you explicitly need to overwrite the remote branch.
- It is useful when cleaning up messy commits. It allows you to rewrite the history and correct push commits with error.