

TRAINING CONTENT

Continuous Integration

Session-1

YOUR NEXT DESTINATION
OF SOFTWARE OUTSOURCING

Lecture Outline

- ☐ What CI/CD Means
- ☐ How GitLab CI/CD works
- ☐ CI/CD Tools comparison
- ☐ GitLab CI/CD Architecture
- ☐ Installation of GitLab
- ☐ Create project and clone Repository
- ☐ Setup GitLab Runner
- ☐ Configure GitLab Runner
- ☐ Create CI/CD pipeline
- ☐ RUN CI/CD Pipeline
- ☐ References

What CI/CD Means

CI/CD

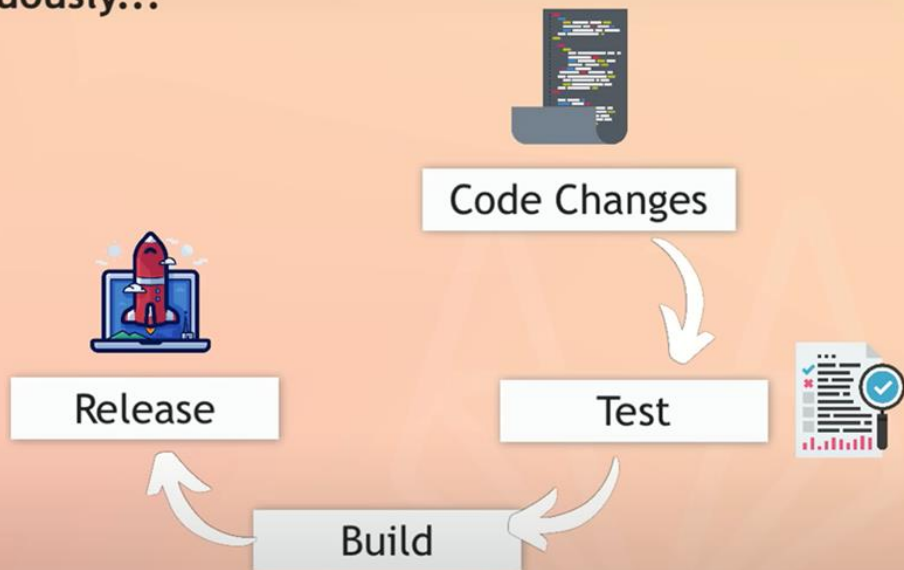
CI → Continuous Integration

CD → Continuous Deployment
or

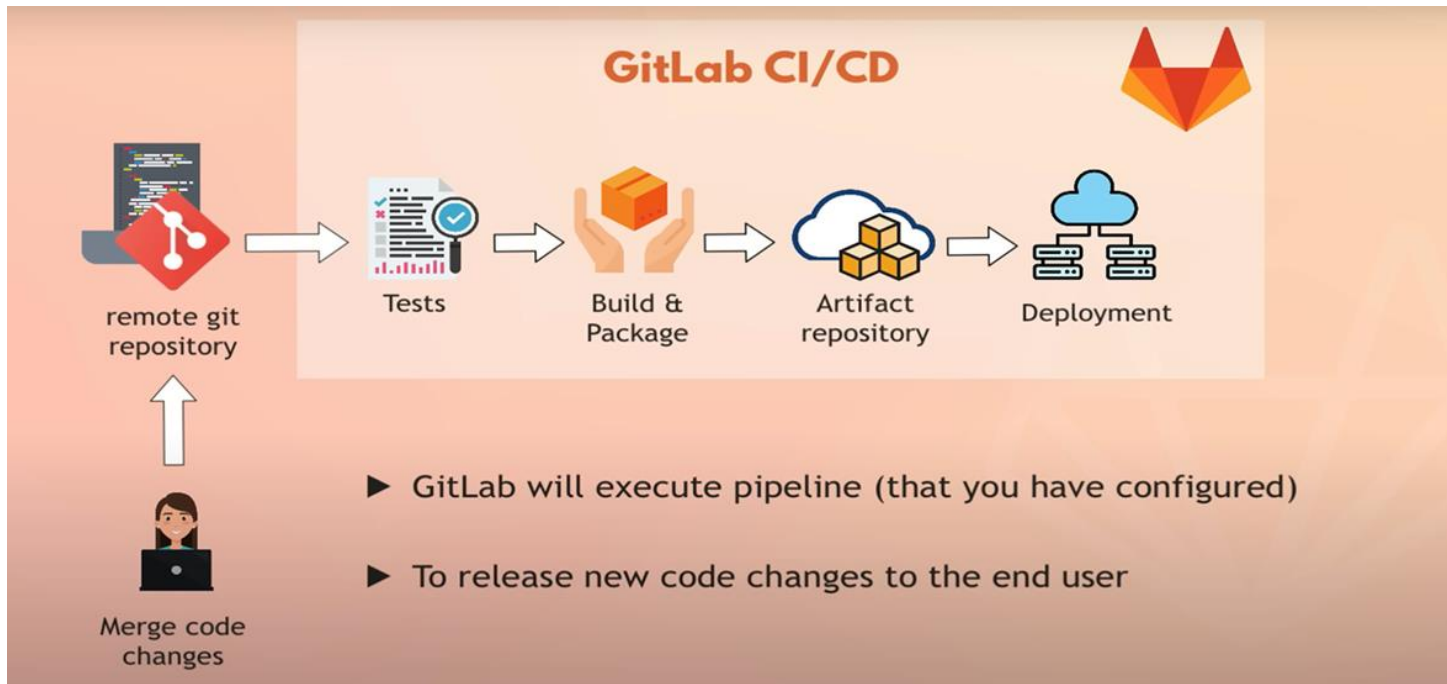
CD → Continuous Delivery

Work Flow Of CI/CD

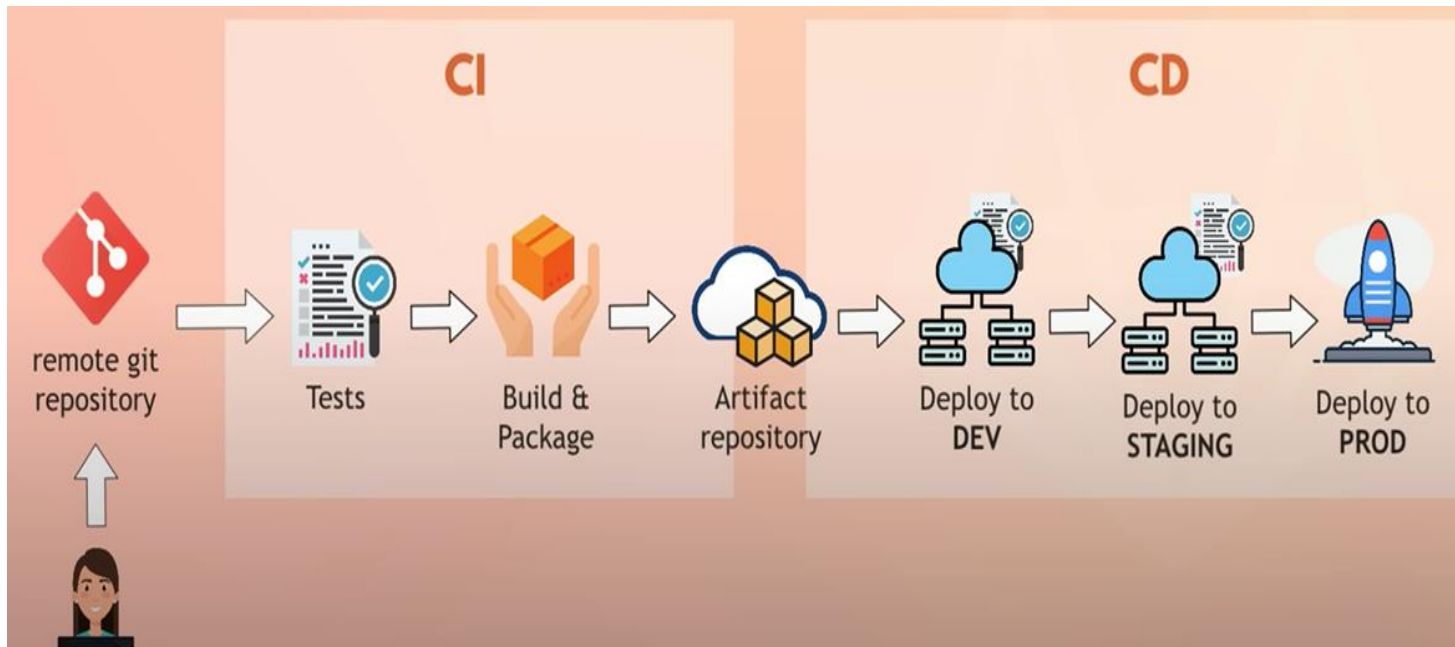
Automatically and continuously...



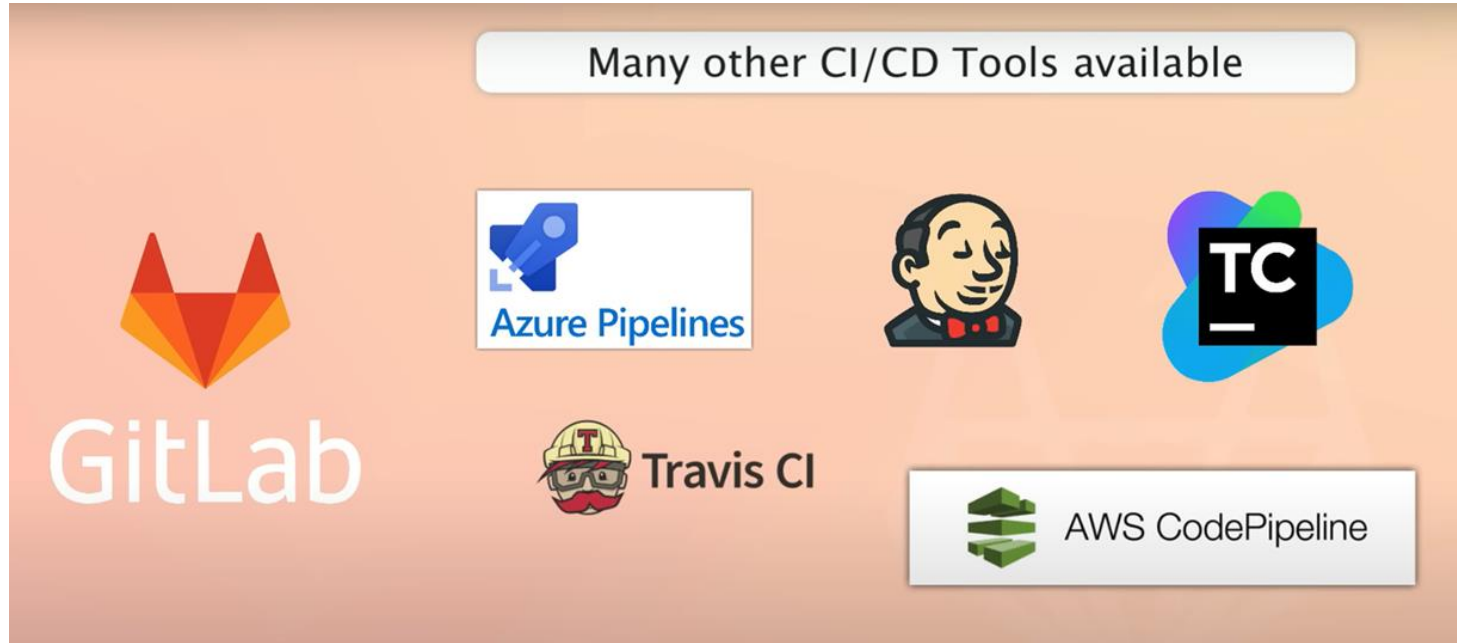
Work Flow Of CI/CD



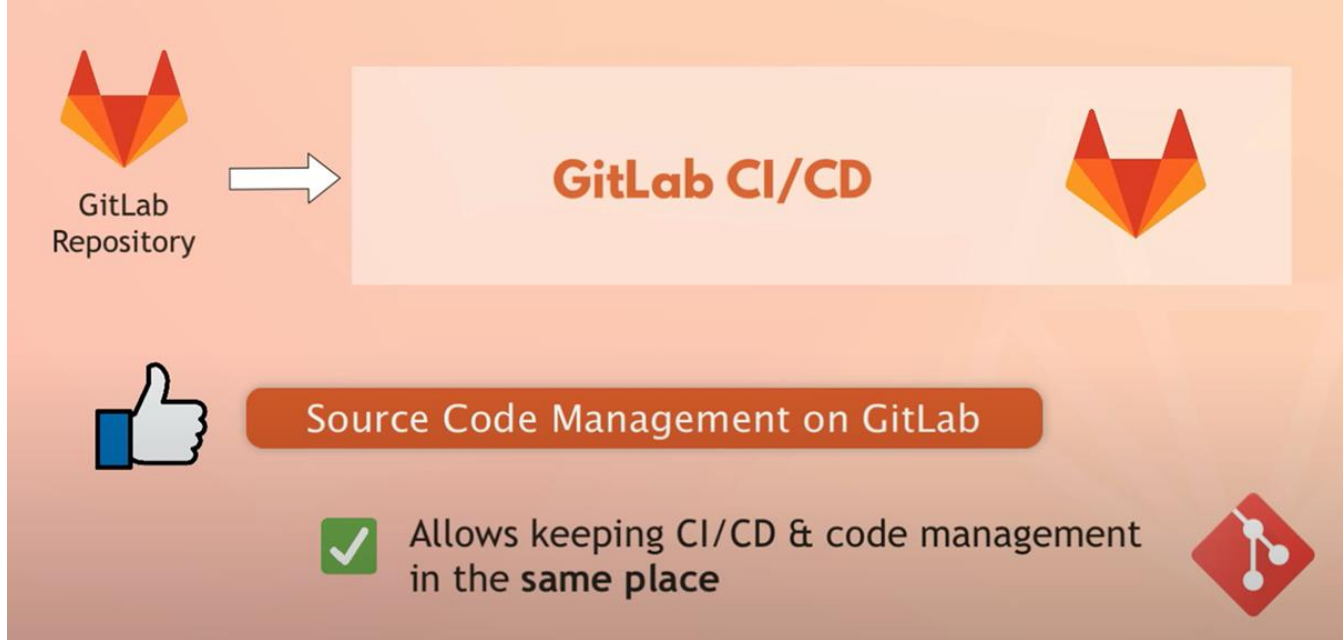
Work Flow Of CI/CD



GitLab In Comparison With Other CI/CD Platforms



Benefits Of Using GitLab CI/CD



GitLab Comparison With Jenkins

Benefits of using GitLab CI/CD



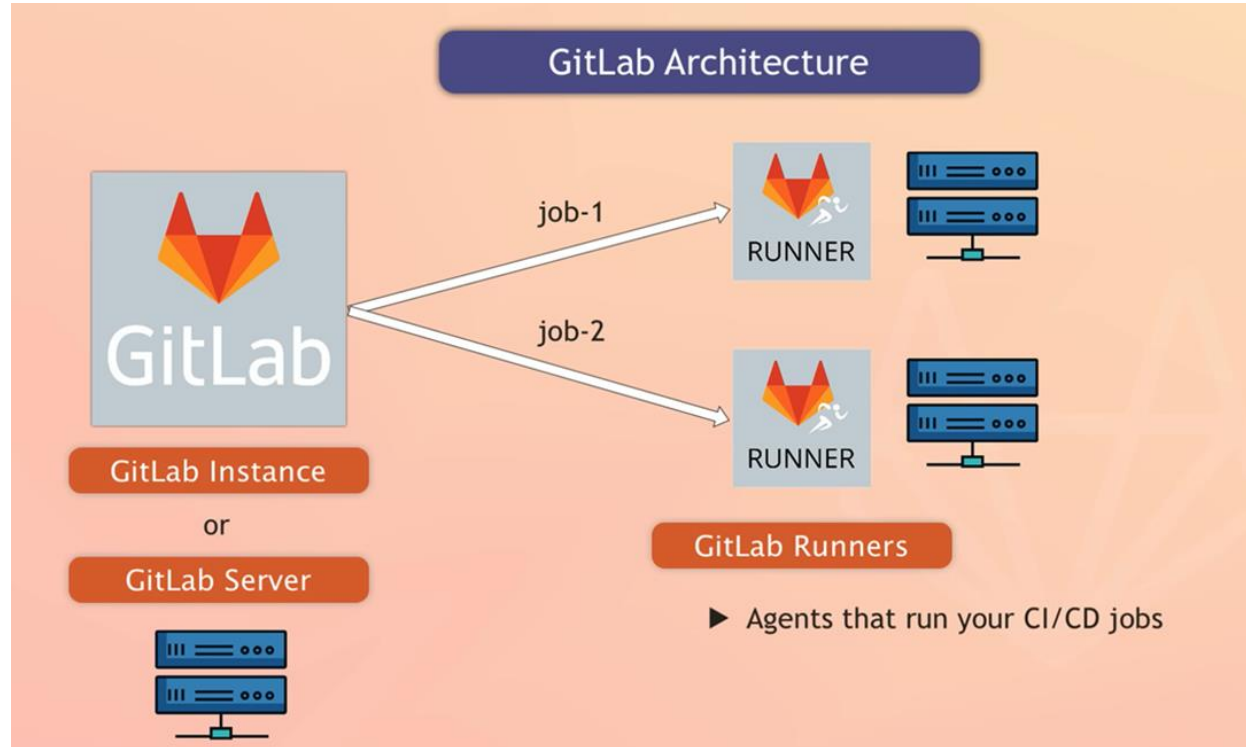
- ✓ Seamless integration into code repository
- ✓ Using CI/CD without overhead of setting it up yourself
- ✓ Pipeline configuration as part of your application code
- ✓ Self-Hosted or SaaS (managed)



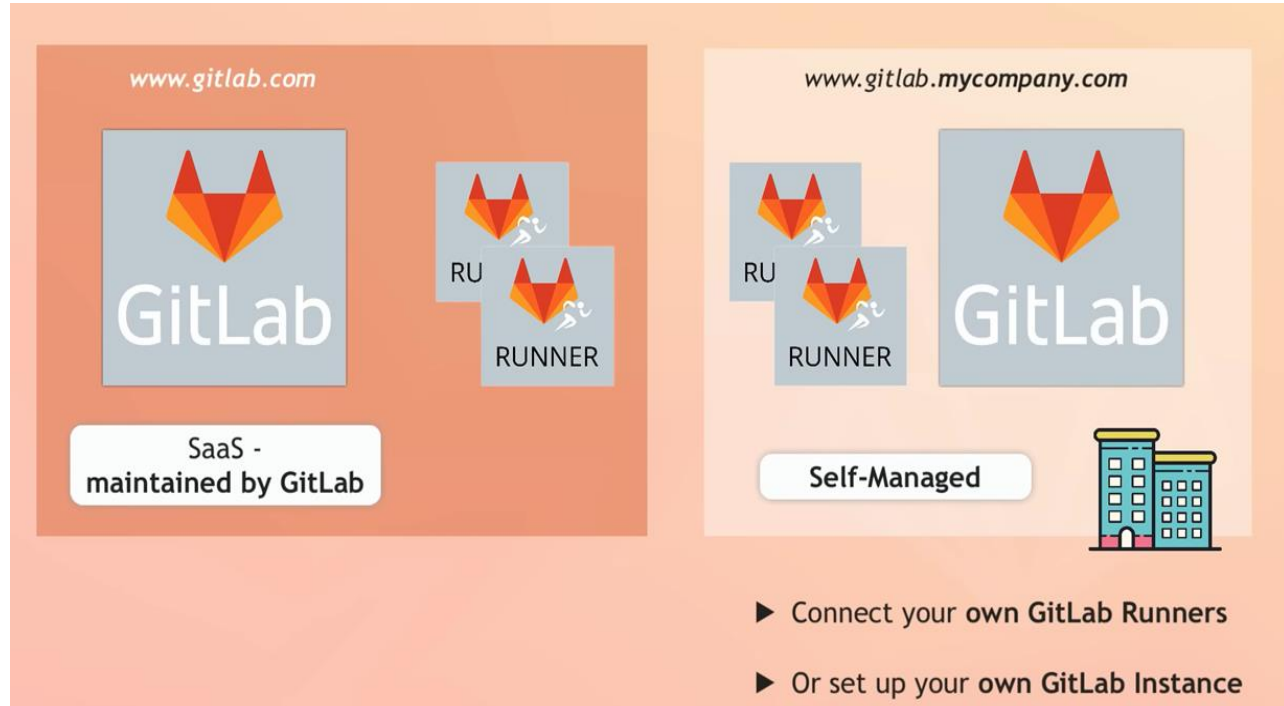
- ⊖ Only CI/CD tool
- ✗ Self-hosting is the only option



GitLab Architecture



GitLab Architecture



GitLab Installation



Prerequisites

- An Ubuntu server 20.04 or other
- A root user or a user with sudo privileges

Step1- Getting Started

```
# apt update -y  
# apt upgrade -y
```

GitLab Installation



Step 2- Add GitLab Repository

```
# apt install curl debian-archive-keyring lsb-release ca-  
certificates apt-transport-https software-properties-common -y
```

Add GPG key with following command

```
gpg_key_url="https://packages.gitlab.com/gitlab/gitlab-ce/gpgkey"  
curl -fsSL $gpg_key_url | gpg --dearmor -o /etc/apt/trusted.gpg.d/gitlab.gpg
```

Add the GitLab repo to APT with following command

```
# nano /etc/apt/sources.list.d/gitlab_gitlab-ce.list
```

Add the following line and save and close

```
deb https://packages.gitlab.com/gitlab/gitlab-ce/ubuntu/ focal main  
deb-src https://packages.gitlab.com/gitlab/gitlab-ce/ubuntu/ focal main
```

Update the repository with following command

```
# apt update -y
```

GitLab Installation



Step 3- Install GitLab

```
# apt install gitlab-ce -y
```

Add GPG key with following command

```
gpg_key_url="https://packages.gitlab.com/gitlab/gitlab-ce/gpgkey"  
curl -fsSL $gpg_key_url | gpg --dearmor -o /etc/apt/trusted.gpg.d/gitlab.gpg
```

Step 4- Configure GitLab

```
# nano /etc/gitlab/gitlab.rb
```

Change the following line and save and close

```
external_url 'http://test-gitlab.local.com'
```

Now reconfigure GitLab

```
# gitlab-ctl reconfigure
```

Verify the GitLab configuration

```
# gitlab-rake gitlab:check
```

GitLab Installation

Step 5- To Access GitLab copy the password

```
# cat /etc/gitlab/initial_root_password
```

To access the Gitlab web UI type following in your web browser

`http://test-gitlab.local.com`

user: root

pass: Password you have copied

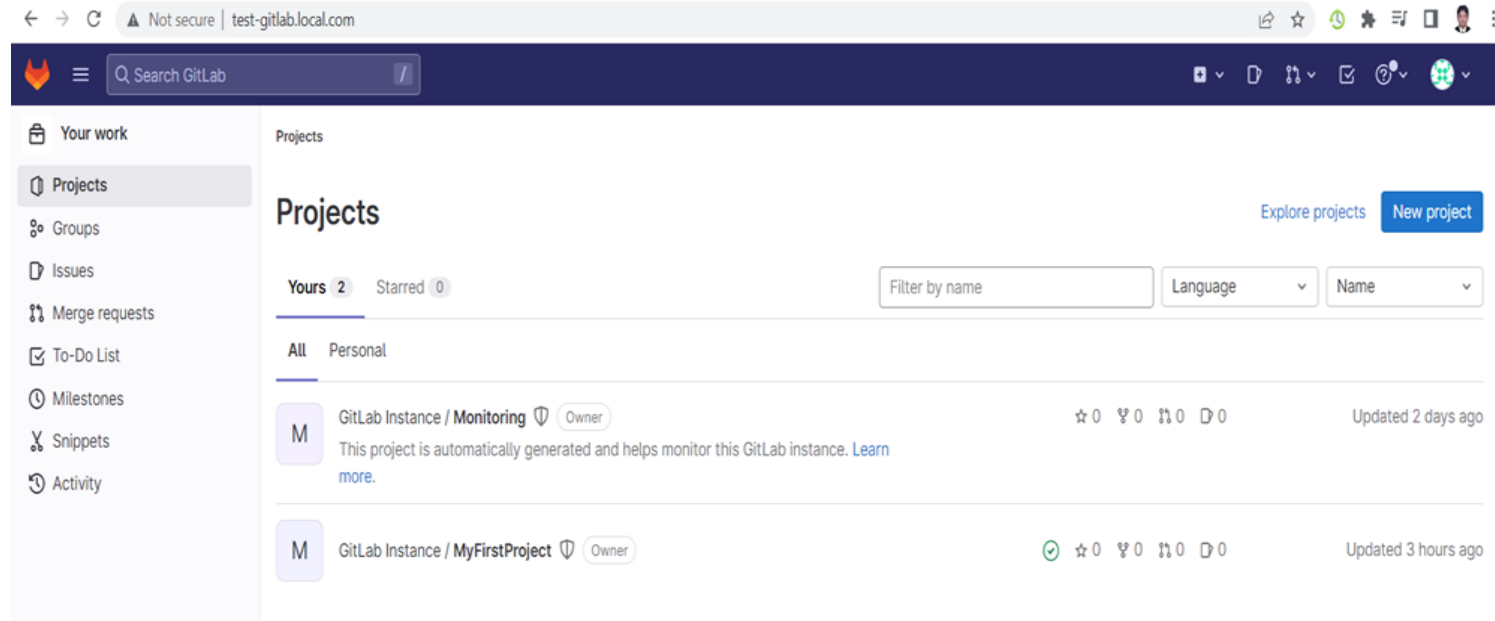
Step 6– Manage GitLab Instance

```
# gitlab-rake 'gitlab:password:reset[root]'
```

To verify the status of GitLab

```
# gitlab-ctl status
```

Create Project



Create Project



test-gitlab.local.com/projects/new#blank_project

Search GitLab

Your work

- Projects
- Groups
- Issues
- Merge requests
- To-Do List
- Milestones
- Snippets
- Activity

Create blank project

Create a blank project to store your files, plan your work, and collaborate on code, among other things.

Project name

Mytestproject

Must start with a lowercase or uppercase letter, digit, emoji, or underscore. Can also contain dots, pluses, dashes, or spaces.

Project URL

http://test-gitlab.local.com/ gitlab-instance-148b6b38

Project slug

mytestproject

Want to organize several dependent projects under the same namespace? [Create a group.](#)

Visibility Level

- ☐ Private
Project access must be granted explicitly to each user. If this project is part of a group, access is granted to members of the group.
- ☐ Internal
The project can be accessed by any logged in user except external users.
- ☒ Public
The project can be accessed without any authentication.

Project Configuration

- ☒ Initialize repository with a README
Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.
- ☐ Enable Static Application Security Testing (SAST)
Analyze your source code for known security vulnerabilities. [Learn more.](#)

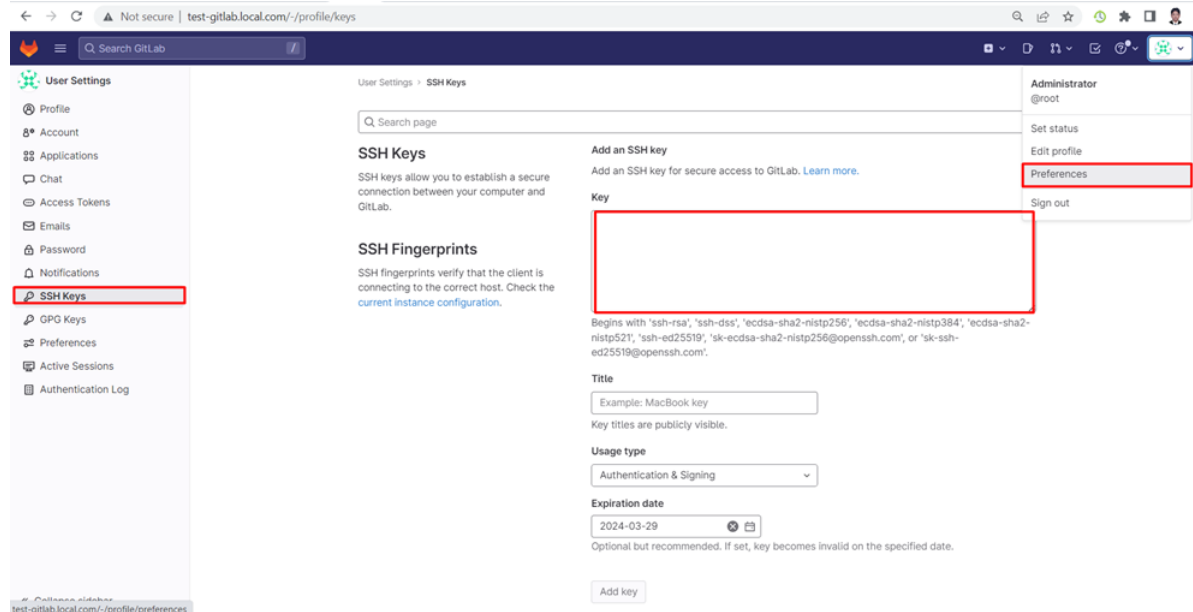
Create project Cancel

Clone Repository



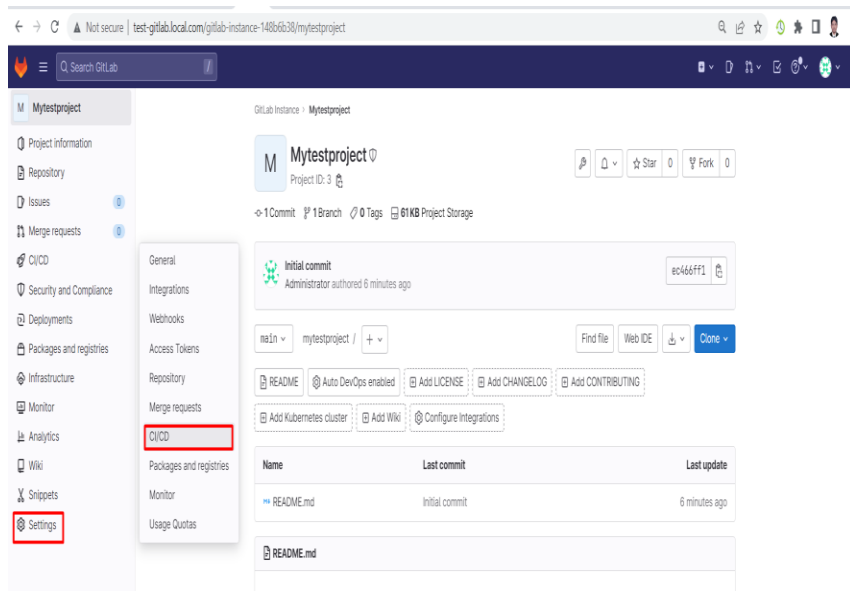
A screenshot of the GitLab web interface. The browser address bar shows 'test-gitlab.local.com/gitlab-instance-148b6b38/mytestproject'. The left sidebar contains a navigation menu with options like 'Project Information', 'Repository', 'Issues', 'Merge requests', 'CI/CD', 'Security and Compliance', 'Deployments', 'Packages and registries', 'Infrastructure', 'Monitor', 'Analytics', 'Wiki', 'Snippets', and 'Settings'. The main content area shows a notification that 'Project 'Mytestproject' was successfully created.' Below this, it displays 'Project ID: 3', '1 Commit', '1 Branch', '0 Tags', and '61 KB Project Storage'. An 'Initial commit' by 'Administrator' is shown with a commit hash 'ec466ff1'. A 'Clone' button is visible, which has been clicked, opening a dropdown menu. This menu is highlighted with a red rectangle and contains three sections: 'Clone with SSH' (showing 'git@test-gitlab.local.com:gitlab'), 'Clone with HTTP' (showing 'http://test-gitlab.local.com/gitlab'), and 'Open in your IDE' (listing Visual Studio Code and IntelliJ IDEA for both SSH and HTTPS).

Add ssh Key To Clone the Repository



The screenshot shows the GitLab web interface. On the left, the 'User Settings' sidebar is visible, with 'SSH Keys' highlighted. The main content area is titled 'SSH Keys' and includes a search bar. Below the title, there are two sections: 'SSH Keys' (describing the purpose of SSH keys) and 'SSH Fingerprints' (describing how fingerprints verify connections). To the right of the main content, there is a sidebar with user information and a list of actions: 'Administrator', '@root', 'Set status', 'Edit profile', 'Preferences' (highlighted with a red box), and 'Sign out'. The 'Add an SSH key' section is also visible, with a large red box highlighting the 'Key' input field. Below this, there are fields for 'Title', 'Usage type' (set to 'Authentication & Signing'), and 'Expiration date' (set to '2024-03-29'). An 'Add key' button is at the bottom of this section.

Runner Installation



GitLab Instance > Mytestproject

Mytestproject
Project ID: 3

1 Commit 1 Branch 0 Tags 61 KB Project Storage

Initial commit
Administrator authored 6 minutes ago

naIn mytestproject / +

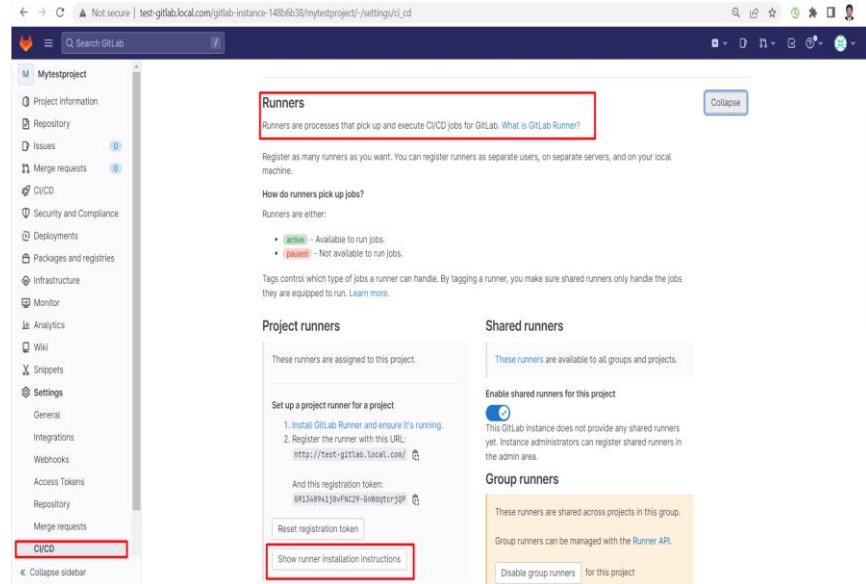
Find file Web IDE Clone

README Auto DevOps enabled Add LICENSE Add CHANGELOG Add CONTRIBUTING

Add Kubernetes cluster Add Wiki Configure Integrations

Name	Last commit	Last update
README.md	Initial commit	6 minutes ago

README.md



Runners
Runners are processes that pick up and execute CI/CD jobs for GitLab. [What is GitLab Runner?](#)

Register as many runners as you want. You can register runners as separate users, on separate servers, and on your local machine.

How do runners pick up jobs?

Runners are either:

- active** - Available to run jobs.
- paused** - Not available to run jobs.

Tags control which type of jobs a runner can handle. By tagging a runner, you make sure shared runners only handle the jobs they are equipped to run. [Learn more.](#)

Project runners

These runners are assigned to this project.

Set up a project runner for a project

1. Install GitLab Runner and ensure it's running.
2. Register the runner with this URL:
<http://test-gitlab.local.com/>

And this registration token:
981348W4318F8C29-6m8qterj0P

Reset registration token

Show runner installation instructions

Shared runners

These runners are available to all groups and projects.

Enable shared runners for this project

This GitLab instance does not provide any shared runners yet. Instance administrators can register shared runners in the admin area.

Group runners

These runners are shared across projects in this group.

Group runners can be managed with the [Runner API](#).

Disable group runners for this project

Runner Installation



Install a runner

Environment

Linux

macOS

Windows

Docker

Kubernetes

AWS

Architecture

amd64

Download and install binary

Download latest binary

```
# Download the binary for your system
sudo curl -L --output /usr/local/bin/gitlab-runner https://gitlab-runner-downloa
ds.s3.amazonaws.com/latest/binaries/gitlab-runner-linux-amd64

# Give it permission to execute
sudo chmod +x /usr/local/bin/gitlab-runner

# Create a GitLab Runner user
sudo useradd --comment 'GitLab Runner' --create-home gitlab-runner --shell /bin/
bash

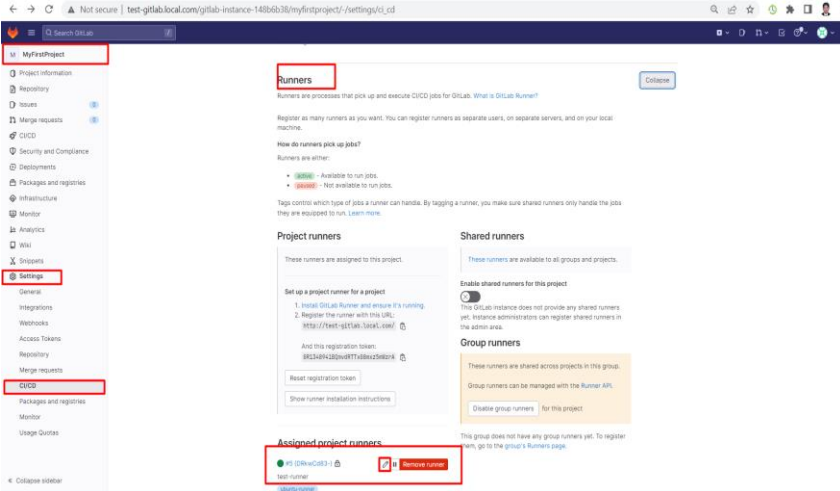
# Install and run as a service
sudo gitlab-runner install --user=gitlab-runner --working-directory=/home/gitlab
-runner
sudo gitlab-runner start
```

Command to register runner

```
sudo gitlab-runner register --url http://test-gitlab.local.com/ --registration-t
oken $REGISTRATION_TOKEN
```

Close

Runner Configuration



Runners

Runners are processes that pick up and execute CI/CD jobs for GitHub. What is GitHub Runner?

Register as many runners as you want. You can register runners as separate users, on separate servers, and on your local machine.

How do runners pick up jobs?

Runners are either:

- **Available** - Available to run jobs.
- **Unavailable** - Not available to run jobs.

Tags control which type of jobs a runner can handle. By tagging a runner, you make sure shared runners only handle the jobs they are equipped to run. [Learn more.](#)

Project runners

These runners are assigned to this project.

Set up a project runner for a project

1. Install GitHub Runner and ensure it's running.
2. Register the runner with this URL: <https://test-github.local/>

And this registration token: [892349x226w9f7Tt8u292928](#)

[Reset registration token](#)

[Show runner installation instructions](#)

Shared runners

These runners are available to all groups and projects.

Enable shared runners for this project

☐ This GitHub instance does not provide any shared runners yet. Instance administrators can register shared runners in the admin area.

Group runners

These runners are shared across projects in this group.

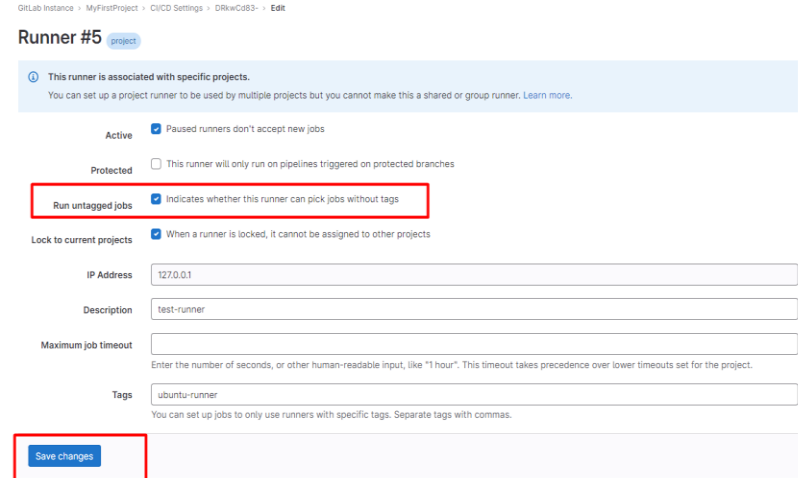
Group runners can be managed with the Runner API.

[Create group runners](#) for this project

Assigned project runners

Runner	Status	Tags
test-runner	Available	

[Remove runner](#)



Runner #5

This runner is associated with specific projects.

You can set up a project runner to be used by multiple projects but you cannot make this a shared or group runner. [Learn more.](#)

Active ☒ Paused runners don't accept new jobs

Protected ☐ This runner will only run on pipelines triggered on protected branches

Run untagged jobs ☒ Indicates whether this runner can pick jobs without tags

Lock to current projects ☒ When a runner is locked, it cannot be assigned to other projects

IP Address 127.0.0.1

Description test-runner

Maximum job timeout

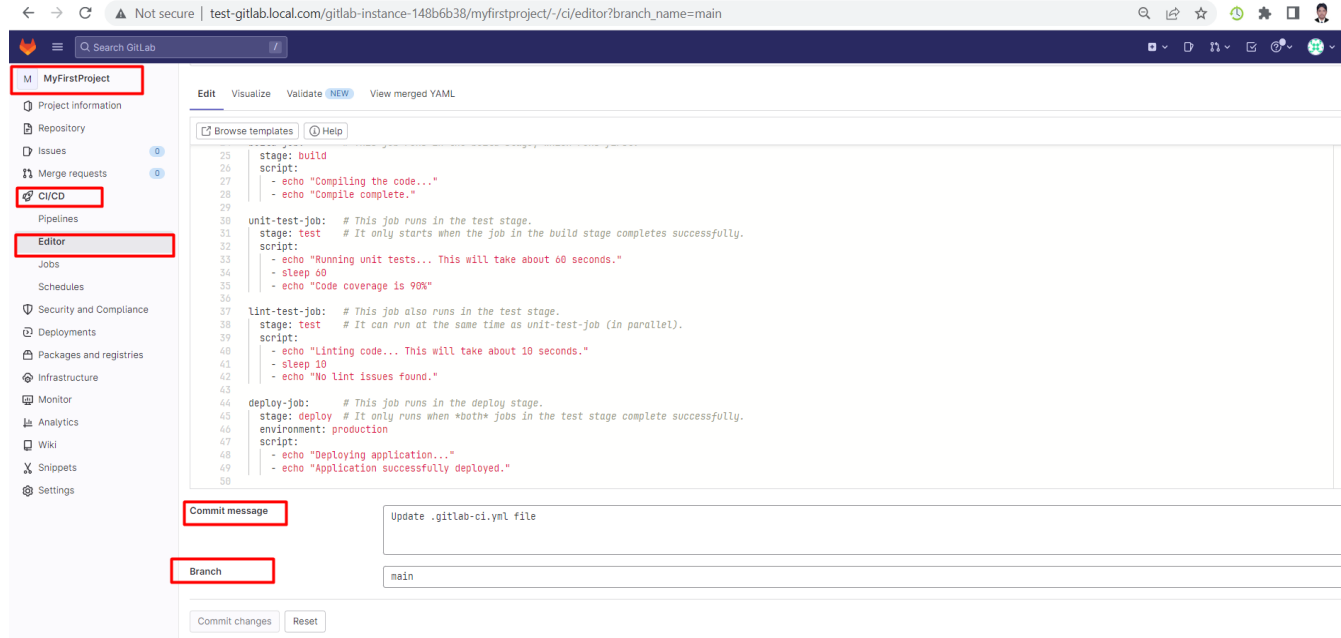
Enter the number of seconds, or other human-readable input, like "1 hour". This timeout takes precedence over lower timeouts set for the project.

Tags ubuntu-runner

You can set up jobs to only use runners with specific tags. Separate tags with commas.

[Save changes](#)

Create CI/CD Pipeline



test-gitlab.local.com/gitlab-instance-148b6b38/myfirstproject/-/ci/editor?branch_name=main

Search GitLab

MyFirstProject

Project information

Repository

Issues 0

Merge requests 0

CI/CD

Pipelines

Editor

Jobs

Schedules

Security and Compliance

Deployments

Packages and registries

Infrastructure

Monitor

Analytics

Wiki

Snippets

Settings

Edit Visualize Validate (NEW) View merged YAML

Browse templates Help

```

25 stage: build
26 script:
27   - echo "Compiling the code..."
28   - echo "Compile complete."
29
30 unit-test-job: # This job runs in the test stage.
31 stage: test # It only starts when the job in the build stage completes successfully.
32 script:
33   - echo "Running unit tests... This will take about 60 seconds."
34   - sleep 60
35   - echo "Code coverage is 90%"
36
37 lint-test-job: # This job also runs in the test stage.
38 stage: test # It can run at the same time as unit-test-job (in parallel).
39 script:
40   - echo "Linting code... This will take about 10 seconds."
41   - sleep 10
42   - echo "No lint issues found."
43
44 deploy-job: # This job runs in the deploy stage.
45 stage: deploy # It only runs when both jobs in the test stage complete successfully.
46 environment: production
47 script:
48   - echo "Deploying application..."
49   - echo "Application successfully deployed."
50

```

Commit message

Update .gitlab-ci.yml file

Branch

main

Commit changes Reset

stages: *# List of stages for jobs, and their order of execution*

- build
- test
- deploy

build-job: *# This job runs in the build stage, which runs first.*

stage: build

script:

- echo "Compiling the code..."
- echo "Compile complete."

unit-test-job: *# This job runs in the test stage.*

stage: test *# It only starts when the job in the build stage completes successfully.*

script:

- echo "Running unit tests... This will take about 60 seconds."
- sleep 60
- echo "Code coverage is 90%"

deploy-job: *# This job runs in the deploy stage.*

stage: deploy *# It only runs when *both* jobs in the test stage complete successfully.*

environment: production

script:

Run CI/CD Pipeline



MyFirstProject

- Project information
- Repository
- Issues
- Merge requests
- CI/CD
- Pipelines**
- Editor
- Jobs
- Schedules
- Security and Compliance
- Deployments
- Packages and registries
- Infrastructure
- Monitor
- Analytics
- Wiki
- Snippets
- Settings

GitLab Instance > MyFirstProject > Pipelines

All 12 Finished Branches Tags

Clear runner caches CI lint **Run pipeline**

Filter pipelines

Status	Pipeline	Triggerer	Stages
passed 0 00:01:14 3 hours ago	test commit #12 main 9bd64cf9 latest		✓✓✓
passed 0 00:01:13 21 hours ago	Update .gitlab-ci.yml file #11 main 9db78248		✓✓✓
passed 0 00:01:12 21 hours ago	Update .gitlab-ci.yml file #10 main 9db78248		✓✓✓
passed 0 00:01:12 21 hours ago	Update .gitlab-ci.yml file #9 main 9db78248		✓✓✓
canceled 21 hours ago	Update .gitlab-ci.yml file #8 main 9db78248		✗✗✗
passed 21 hours ago	Update .gitlab-ci.yml file #7 main 9db78248		✓✓✓
failed 0 00:00:09 1 day ago	Update .gitlab-ci.yml file #6 main 9db78248		✗✗✗
failed 0 00:00:09	Update .gitlab-ci.yml file #5 main 9db78248		✗✗✗

« Collapse sidebar

References



Gitlab Installation

<https://cloudinfrastructureservices.co.uk/how-to-install-gitlab-on-ubuntu-22-04/>

GitLab official documentation

<https://docs.gitlab.com/ee/ci/>

A white L-shaped line is positioned to the left of the "Thank You" text.

Thank You

A white L-shaped line is positioned to the right of the "Thank You" text.

Get in touch with us:

www.bjitgroup.com