

# Configuration Manual

MSc Research Project  
MSc in Cybersecurity

**Preeti Bhardwaj**  
Student ID: x21139351

School of Computing  
National College of Ireland

Supervisor: Arghir Nicolae Moldovan

**National College of Ireland**  
**MSc Project Submission Sheet**  
**School of Computing**



**Student Name:** Preeti Bhardwaj  
**Student ID:** X21139351  
**Programme:** MSc Cybersecurity **Year:** 2022-2023  
**Module:** MSc Research Project  
**Lecturer:** Arghir Nicolae Moldovan  
**Submission Due Date:** 15-12-2022  
**Project Title:** Detecting container vulnerabilities leveraging the CICD pipeline  
**Word Count:** **Page Count:**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature:** Preeti Bhardwaj

**Date:** 15-12-2022

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
<b>Attach a Moodle submission receipt of the online project submission,</b> to each project (including multiple copies).	<input type="checkbox"/>
<b>You must ensure that you retain a HARD COPY of the project,</b> both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

<b>Office Use Only</b>	
Signature:	
Date:	
Penalty Applied (if applicable):	

# Configuration Manual

## Introduction:

In the research project we explored how docker vulnerabilities can be detected in an automated pipeline to strengthen the security of vulnerable docker images.. The research methodology and design followed consisted of an implementation that was setup to achieve the research outputs. We set up a CICD pipeline in Gitlab and incorporated a vulnerability scanning tool to scan docker images. The configuration manual is a step-by-step guide to install, setup and implement the research.

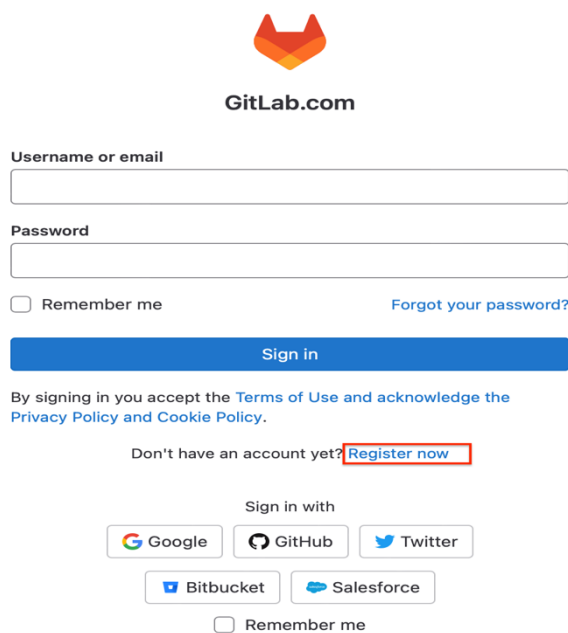
## Configurations:

<b>Gitlab Enterprise edition</b>	<b>15.7.0</b>
<b>Trivy tool</b>	<b>Version : v0.35.0</b>
<b>Clair tool</b>	<b>Version : v8.0</b>
<b>Docker Image #1</b>	<b>cern/cc7-base</b>
<b>Docker Image #2</b>	<b>centos/httpd-24-centos7</b>

## Implementation:

For implementation, the initial goal is to set up a CI CD pipeline:

**Step 1.** Setup a new account on Gitlab by Registering a new account.



GitLab.com

Username or email

Password

Remember me [Forgot your password?](#)

Sign in

By signing in you accept the [Terms of Use](#) and acknowledge the [Privacy Policy](#) and [Cookie Policy](#).

Don't have an account yet? [Register now](#)

Sign in with

Google GitHub Twitter

Bitbucket Salesforce

Remember me

**Step 2.** Once registered, users need to set up the details and after that there will be an option to Create or import a project. We will create a new project for the research. Enter the project and group.

- **Click on Create Project**

## Create or import your first project

Projects help you organize your work. They contain your file repository, issues, merge requests, and so much more.

CreateImport

**Group name**

**Project name**

Your project will be created at:

<https://gitlab.com/research180/research-project>

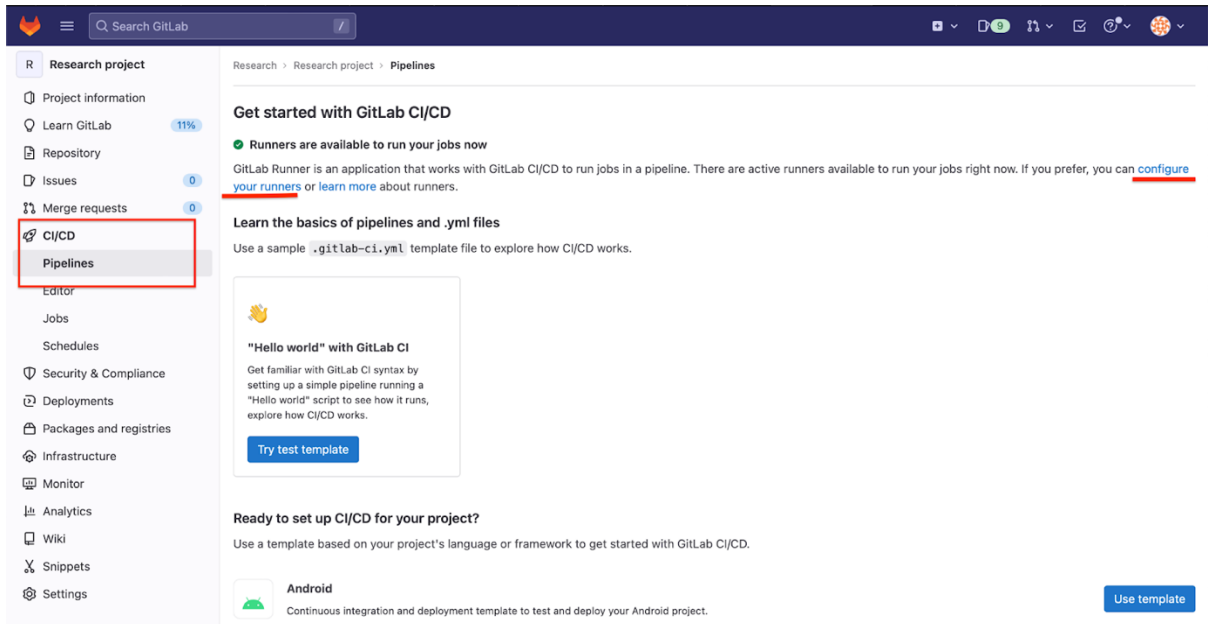
You can always change your URL later

Include a Getting Started README  
Recommended if you're new to GitLab

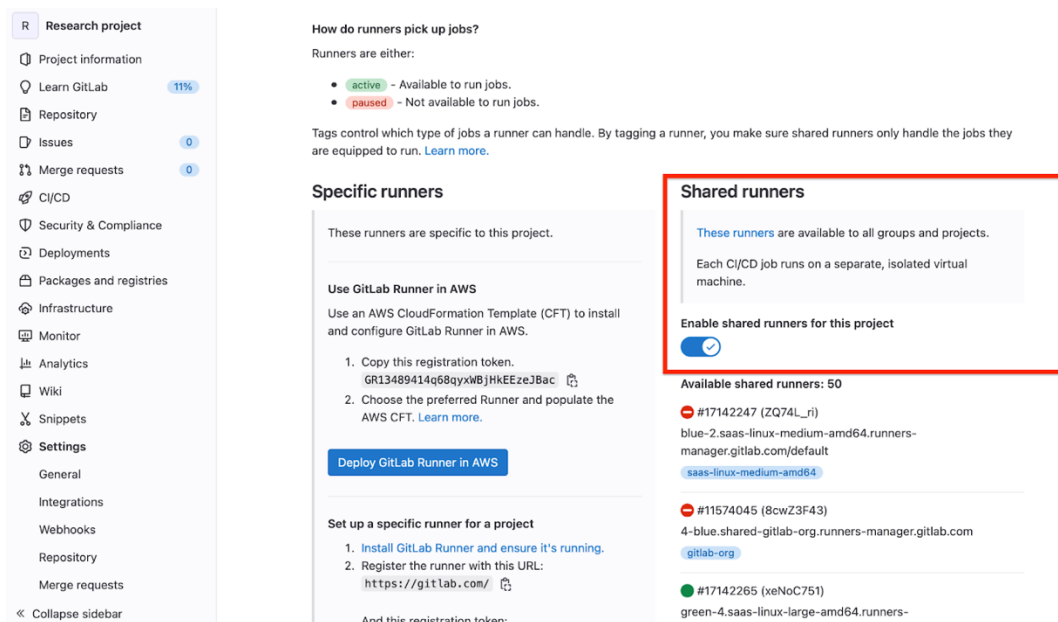
Create project

**Step 3.** After saving the project, the user will be directed to the dashboard. We will configure the Gitlab Runner [1]

- **Click on CICD.**
- **Under CICD, Click on Pipeline**
- **Click on Configure your runner**



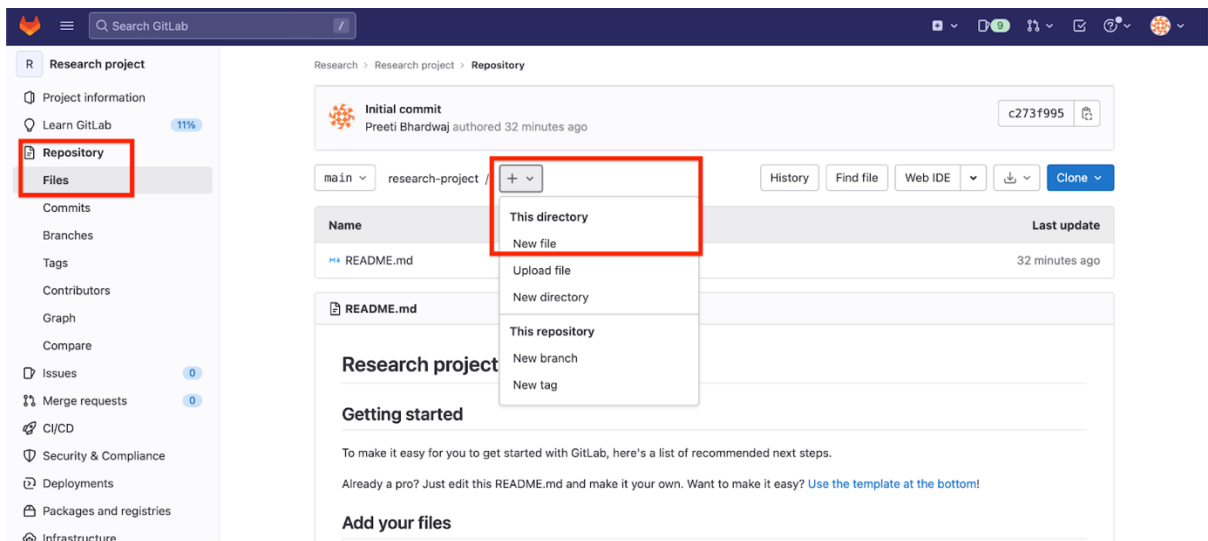
**Step 4.** Enable the shared runner. Runner can be used locally as well following the official documentation by gitlab [2]. However, for our research project it is convenient to use the shared runner without any hassle.



**Step 5.** We will now create the files that will define CICD jobs. We will first create the [.gitlab-ci.yml](#) at the root repository.

- Click on Repository on dashboard
- Click on Files

- Click on + symbol (beside the project name)

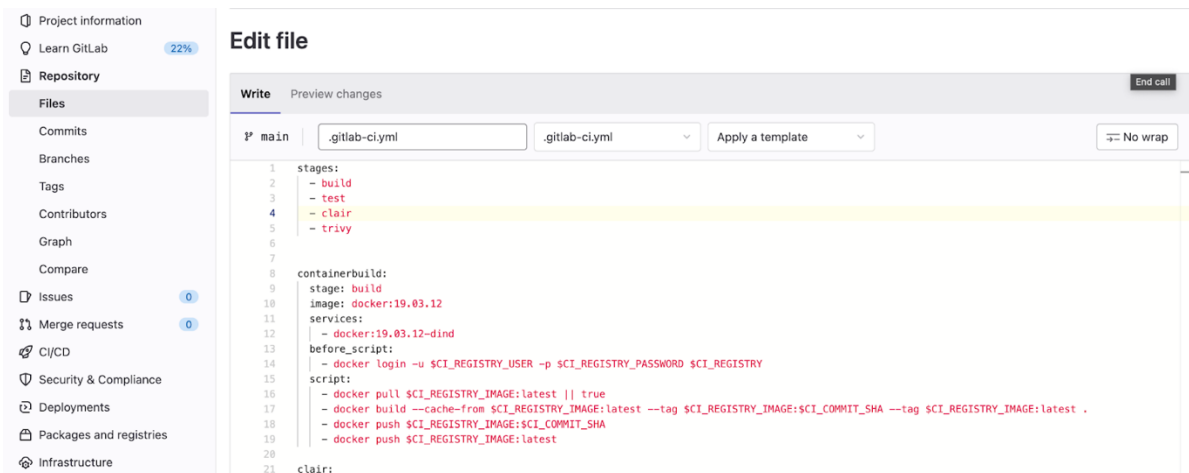


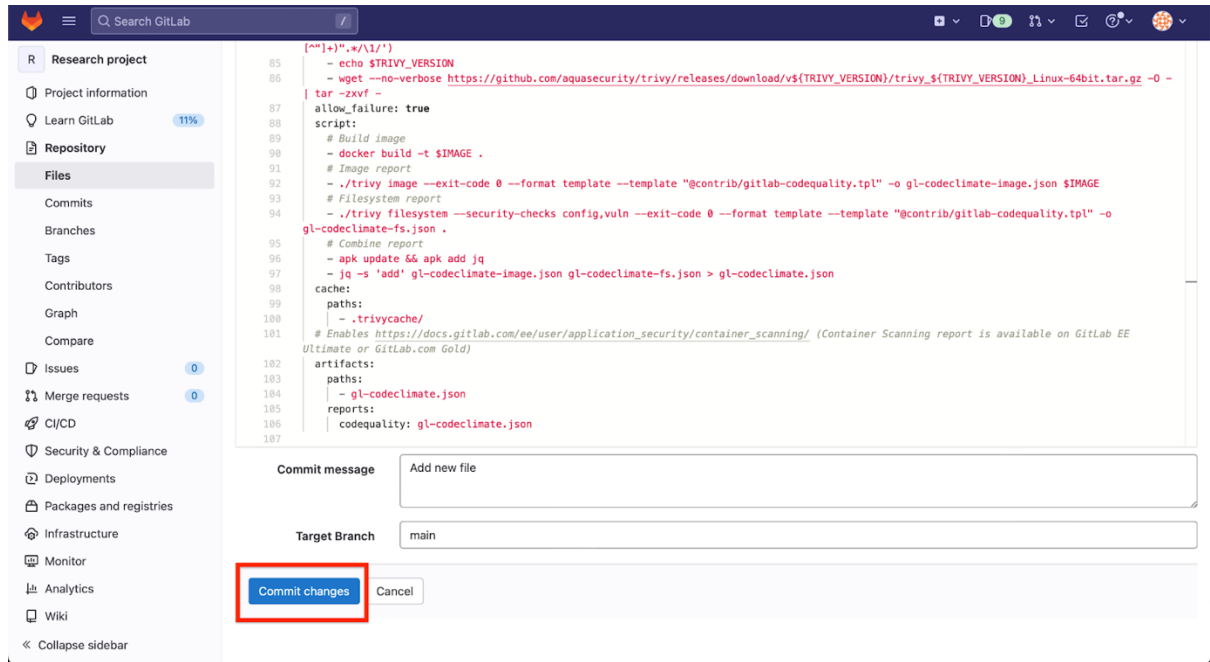
**Step6.** We will create the pipeline file with stages as:

- **build**
- **test**
- **clair**
- **trivy**

*#Code of the pipeline and tools(entire file code) has been shared in the artefacts and code zip file.*

- Use the **.gitlab-ci.yml** pipeline code. (Shared in artefacts and code zip)
- Save file extensions as: **.gitlab-ci.yml**
- Click on **Commit changes**

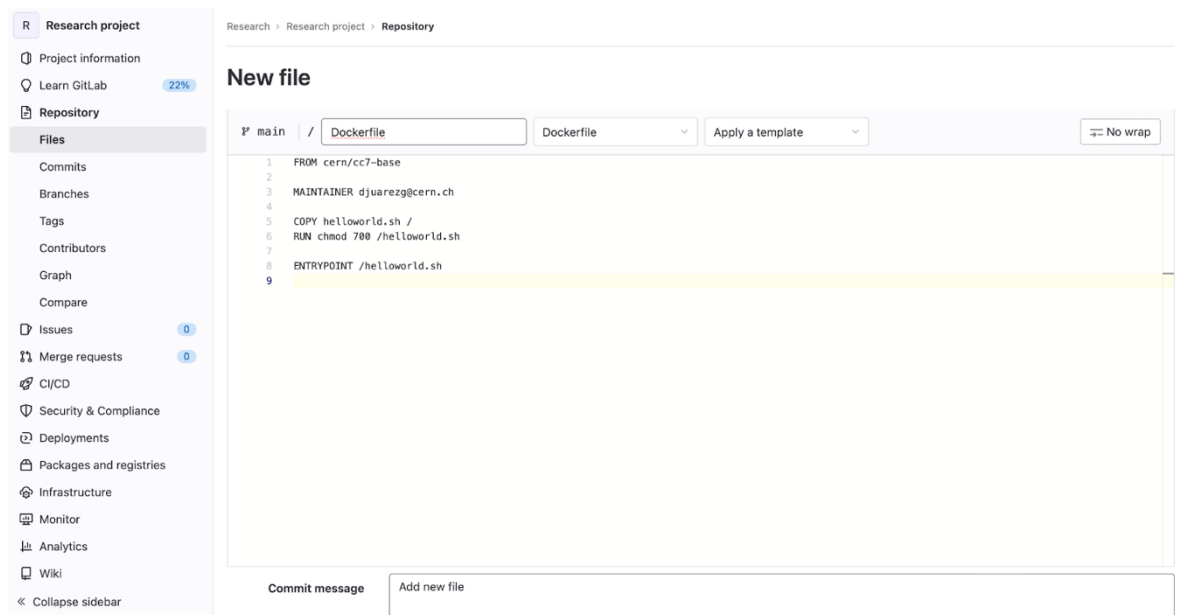




**Step.7 We will now create the docker file, which will contain the docker image.**

- Click on Repository on dashboard
- Click on Files
- Click on + symbol (beside the project name)
- Use Docker Image file code.
- Name the file as: Dockerfile
- Click on Commit changes.

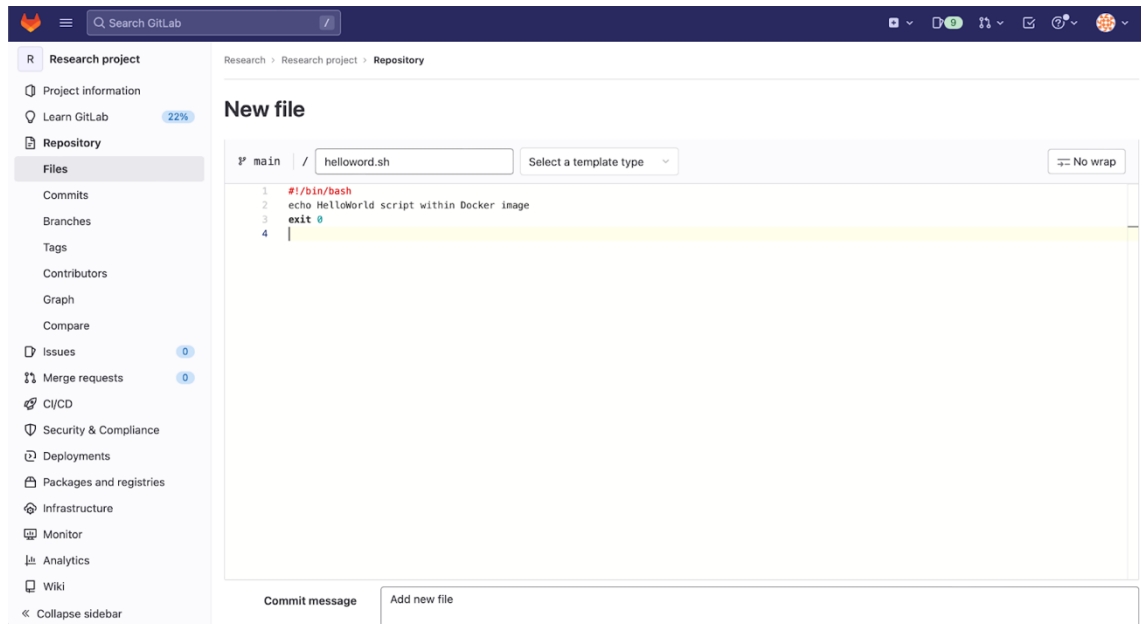
*#Code of the Docker images : Image #1 : cern/cc7-base, Image #2 :centos/httpd-24-centos7 has been shared in the artefacts and code zip file.*



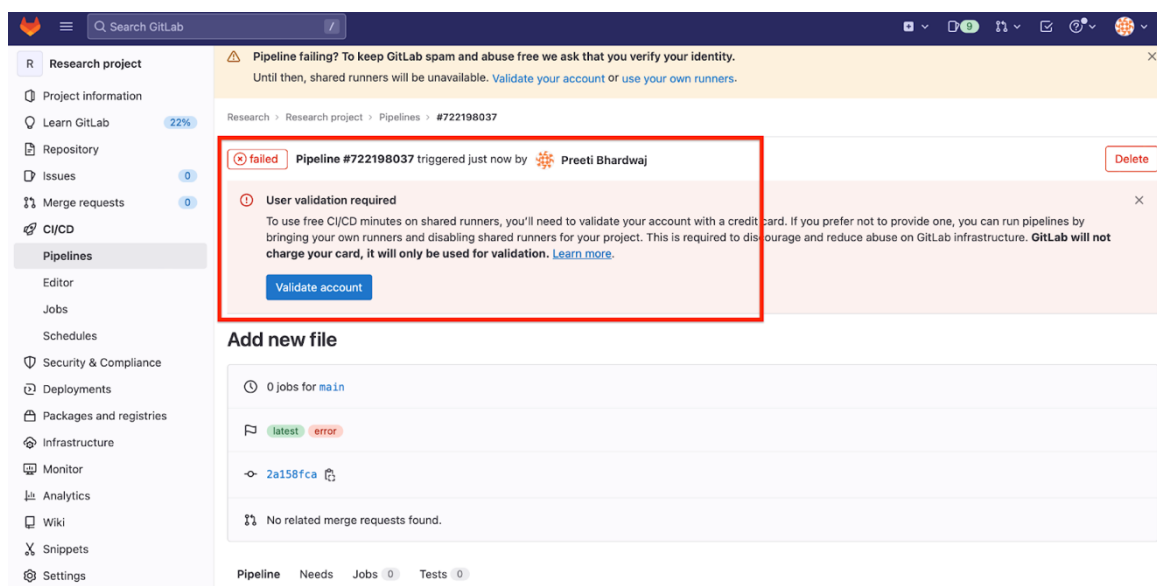
**Step8. We will also create a small Hello word code file that will sit on the top of the Docker image.**

- **Click on Repository on dashboard**
- **Click on Files**
- **Click on + symbol (beside the project name)**
- **Use helloworld.sh file**
- **Click on Commit changes**

*#Code of the helloworld.sh file has been shared in the artefacts and code zip file.*



**Important note:** One thing to note here is if the user has not provided Credit card details to validate the user account. The pipeline will give an **error**:



Once the card details are saved, the account will be validated and CICD pipeline will be ready to run jobs.



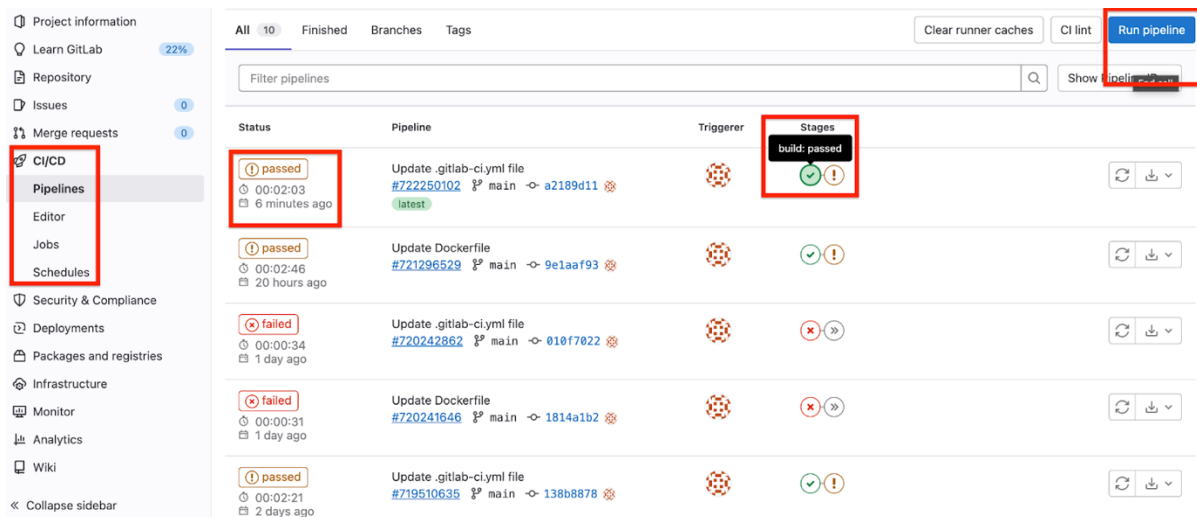
Running the CICD Pipeline: Once the implementation is done , we will now run our pipeline to scan the docker images.

### Step1.

- Click on CICD on the dashboard
- Click on Run pipeline

All the status of the running pipeline will show on the screen, hover above the stages and click to check the running jobs. The status of the running job will show on Pipeline page as well as the status of the running job can be checked from the stages and clicking on each running job under the stage.

- **Note:** While scanning with Clair tool pipeline gets terminated whenever the errors are encountered and detected in the docker image.



The screenshot displays the GitLab CI/CD dashboard. On the left sidebar, the 'CI/CD' section is expanded, with 'Pipelines' highlighted. The main area shows a list of pipelines under the 'All' tab. The pipelines are as follows:

Status	Pipeline	Triggerer	Stages
passed 00:02:03 6 minutes ago	Update .gitlab-ci.yml file #722250102 latest	a2189d11	build: passed
passed 00:02:46 20 hours ago	Update Dockerfile #721296529	9e1aaf93	build: failed
failed 00:00:34 1 day ago	Update .gitlab-ci.yml file #720242862	010f7022	build: failed
failed 00:00:31 1 day ago	Update Dockerfile #720241646	1814a1b2	build: failed
passed 00:02:21 2 days ago	Update .gitlab-ci.yml file #719510635	138b8878	build: passed

## References:

[1] *Gitlab Runner* (no date) *GitLab*. Available at: <https://docs.gitlab.com/runner/> (Accessed: December 15, 2022).

[2] *Install gitlab runner* (no date) *GitLab*. Available at: <https://docs.gitlab.com/runner/install/> (Accessed: December 15, 2022).