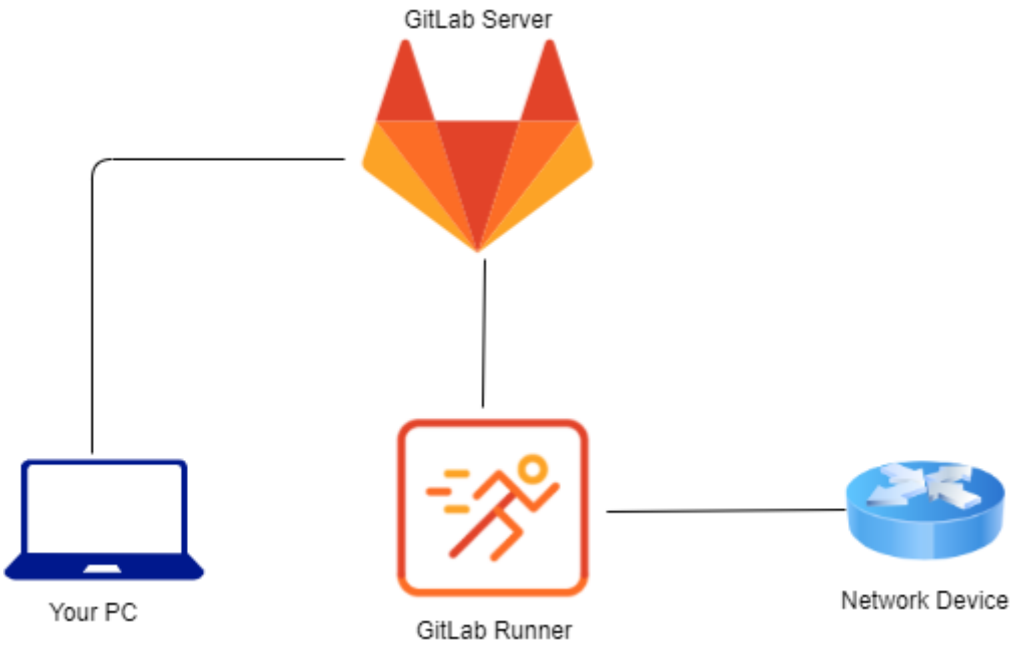
Lab - CI/CD cơ bản (Hello World)

1. Mô hình LAB



Hình 1: Mô hình LAB sử dụng để thực thi.

2. Yêu cầu

Thực hiện xây dựng luồng CI/CD cơ bản để thực hiện deploy thành công in ra " Hello World" với gitlab server và Gitlab Runner.

3. Các bước thực hiện

a) Thực hiện cài đặt Gitlab Server và Gitlab Runner theo file hướng dẫn hoặc có thể làm theo các bước sau đối với máy ảo Centos

- Các bước cài Gitlab Server

sudo yum -y update

sudo yum -y install curl vim policycoreutils python3-policycoreutils

sudo yum -y install postfix

sudo systemctl enable postfix && sudo systemctl start postfix

curl -s https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.rpm.sh | sudo bash

sudo yum install gitlab-ce

sudo vi /etc/gitlab/gitlab.rb

external_url 'http://gitlab.example.com' # Lưu ý: chỉnh sửa file thành địa chỉ card mạng để phù hợp bằng lệnh trên Centos: ifconfig

sudo gitlab-ctl reconfigure

sudo firewall-cmd --permanent --add-service={ssh,http,https} --permanent

sudo firewall-cmd --reload

Sau đó đăng nhập với username mặc định thường là root và password tại vị trí :**/etc/gitlab/initial_root_password** theo địa chỉ đã tạo card mạng ở trên.

- Các bước cài Gitlab Runner

curl -L "https://packages.gitlab.com/install/repositories/runner/gitlab-runner/script.rpm.sh" | sudo bash

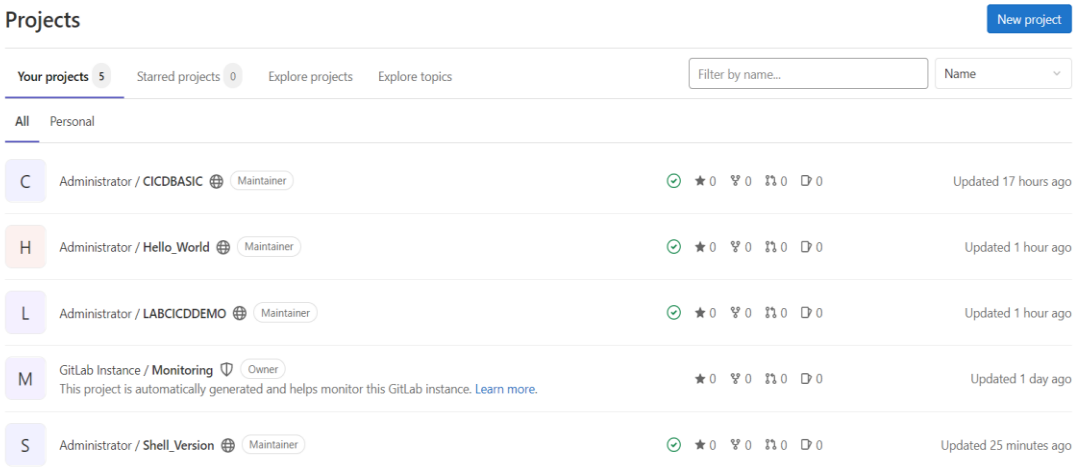
sudo yum install gitlab-runner -y

- Sau khi cài đặt xong thực hiện các bước để register Gitlab Runner với Gitlab Server

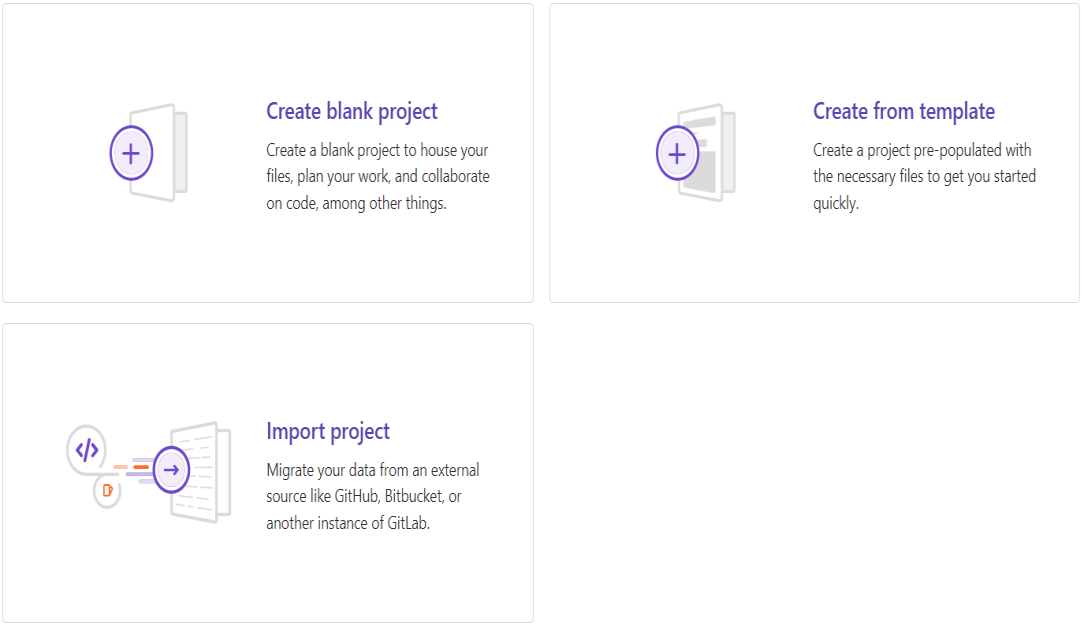
b) Tạo project mới trên Gitlab Server để register Gitlab Runner

- Tạo project trên Gitlab Server

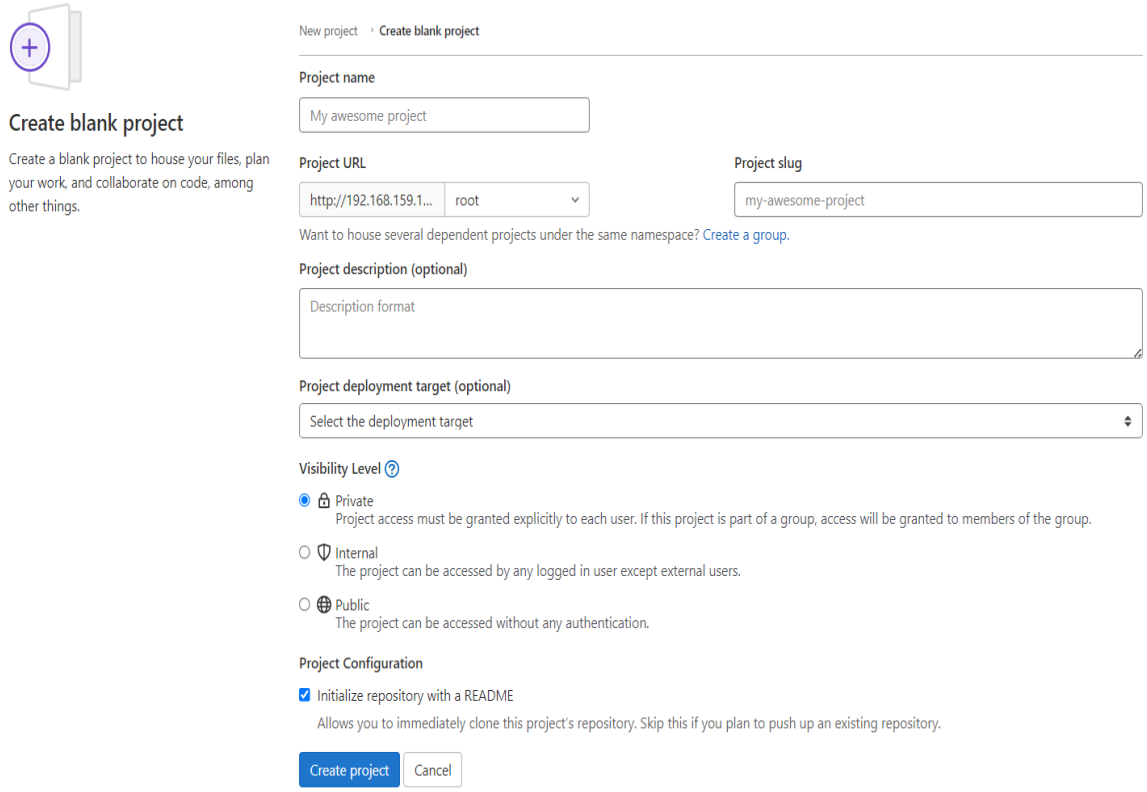
Đăng nhập vào Gitlab Server sau khi có username và password chọn "New project":



Sau đó chọn "Create blank project"



Sau đó điền tiên cũng như Description (nếu có) và chọn chế độ "Private, Internal, Public cần thiết và nhấn "Create project" phía cuối trang.

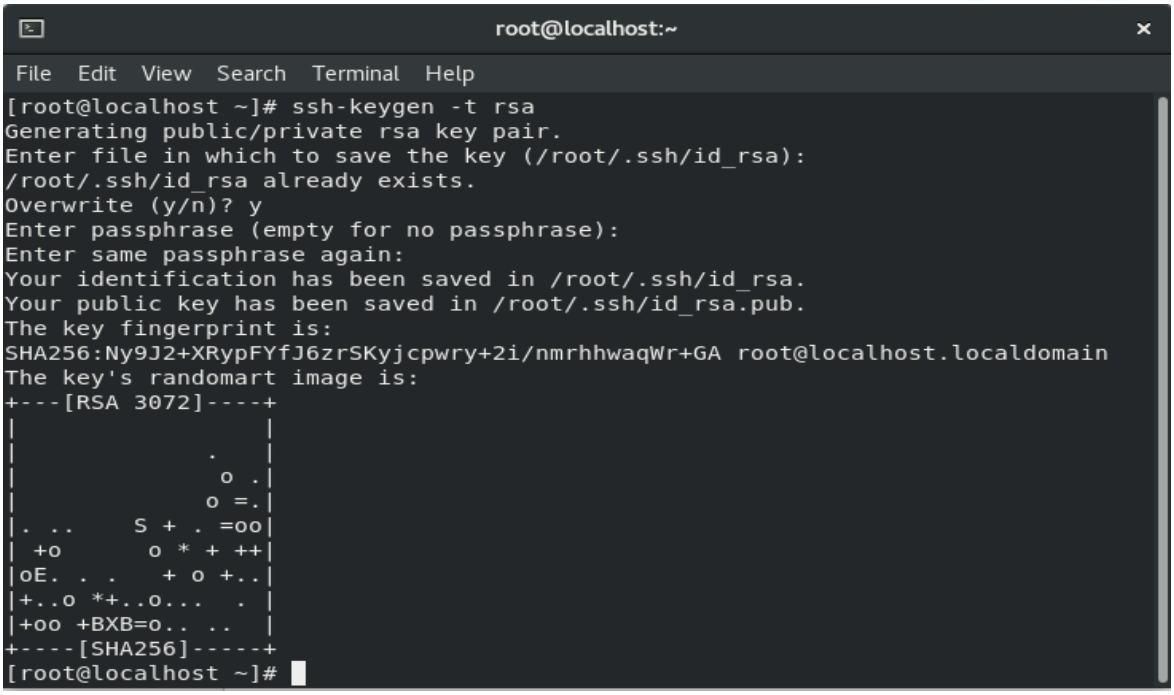


- Cấu hình SSH-Key

+ Trên cửa sổ Terminal của Centos nhập các lệnh sau để tạo SSH-Key:

ssh-keygen -t rsa

+ Một cửa sổ hiện ra như bên dưới bao gồm chọn nơi lưu Key cũng như mật khẩu nếu không, có thể bỏ qua.



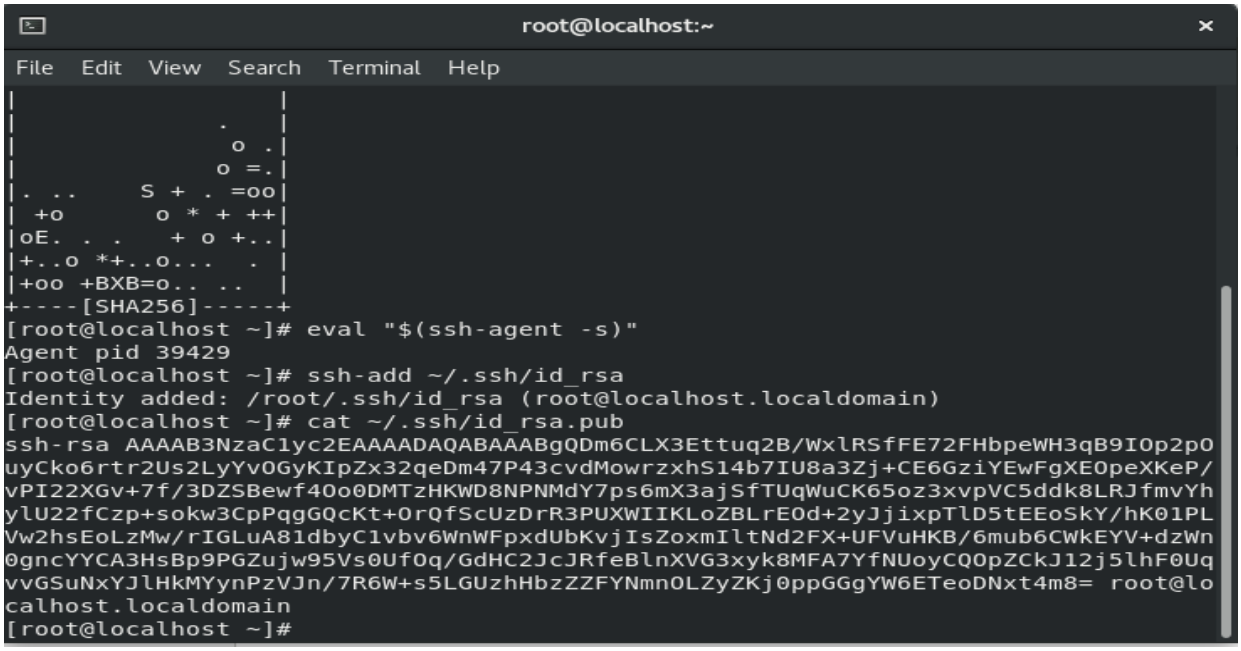
- Tiếp theo thêm Key vào ssh-agent

  eval "$(ssh-agent -s)"

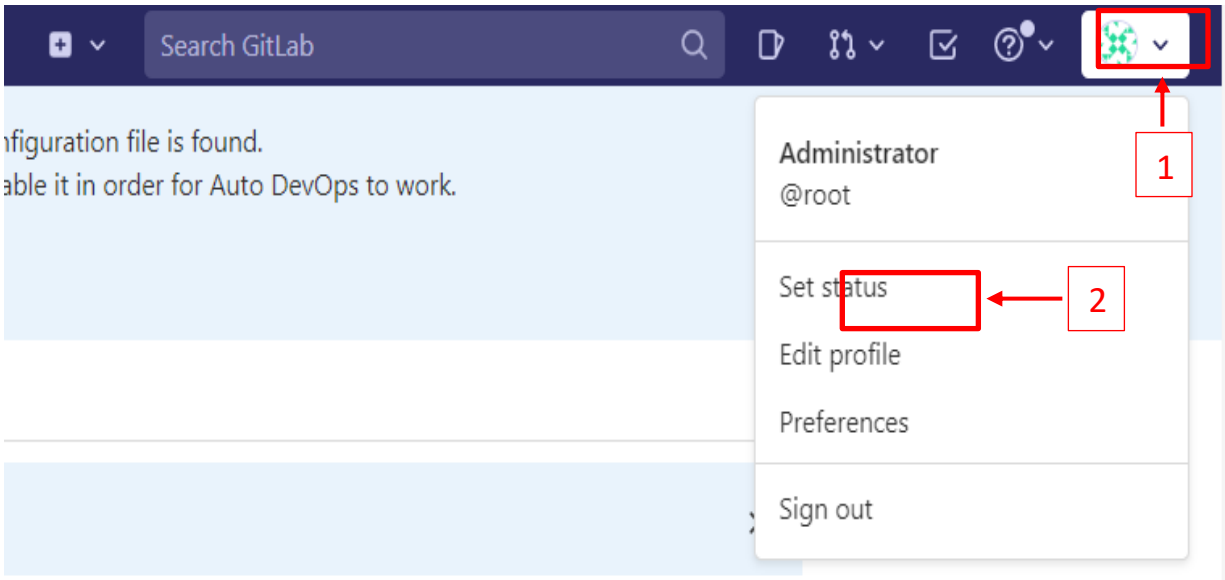
  ssh-add ~/.ssh/id_rsa

  cat ~/.ssh/id_rsa.pub

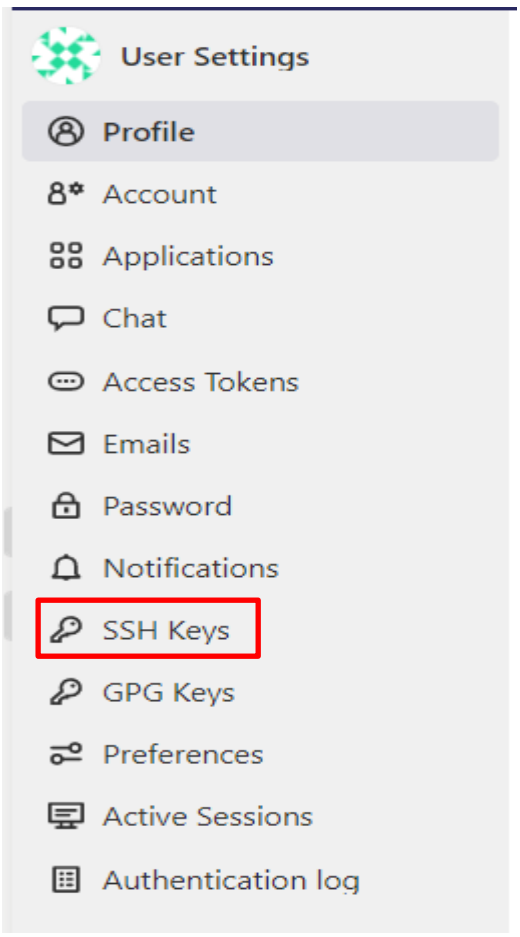
- Sau đó đoạn mã Key hiện ra và thực hiện copy:



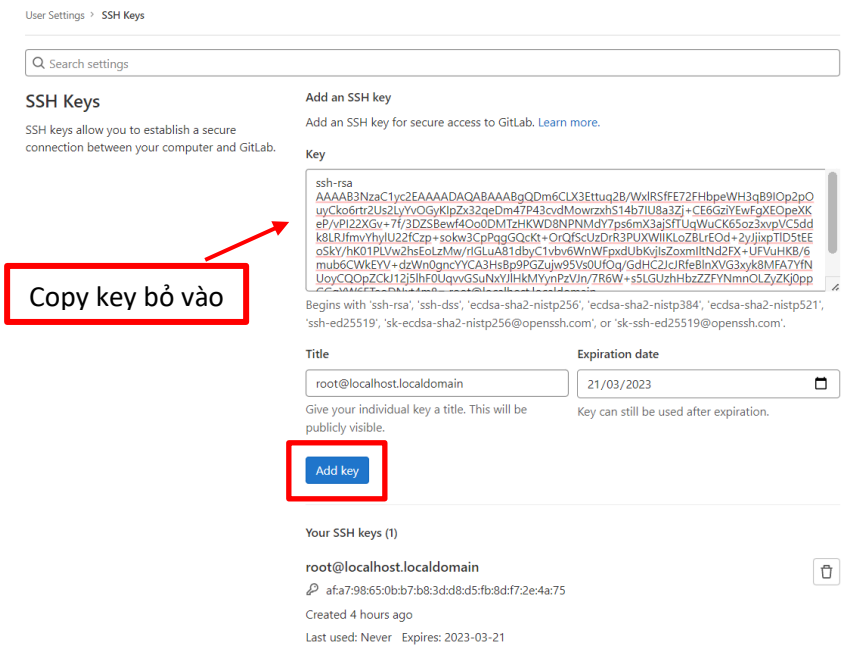
- Tiếp theo mở gitlab Server để thêm SSH-Key. Tạo cửa sổ nhấp vào biểu tượng hình bên góc phải và nhấn chọn **Edit profile**



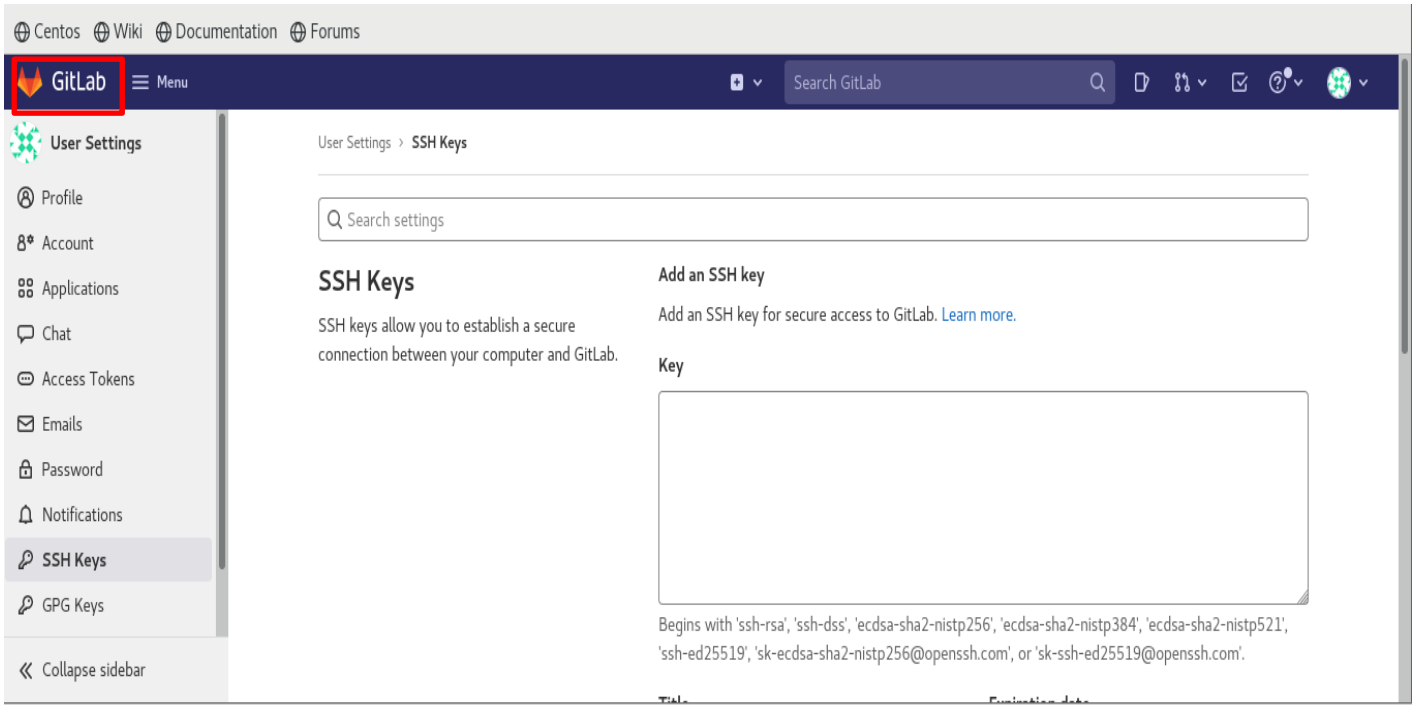
- Sau đó nhấn chọn **SSH Keys** bên góc trái:



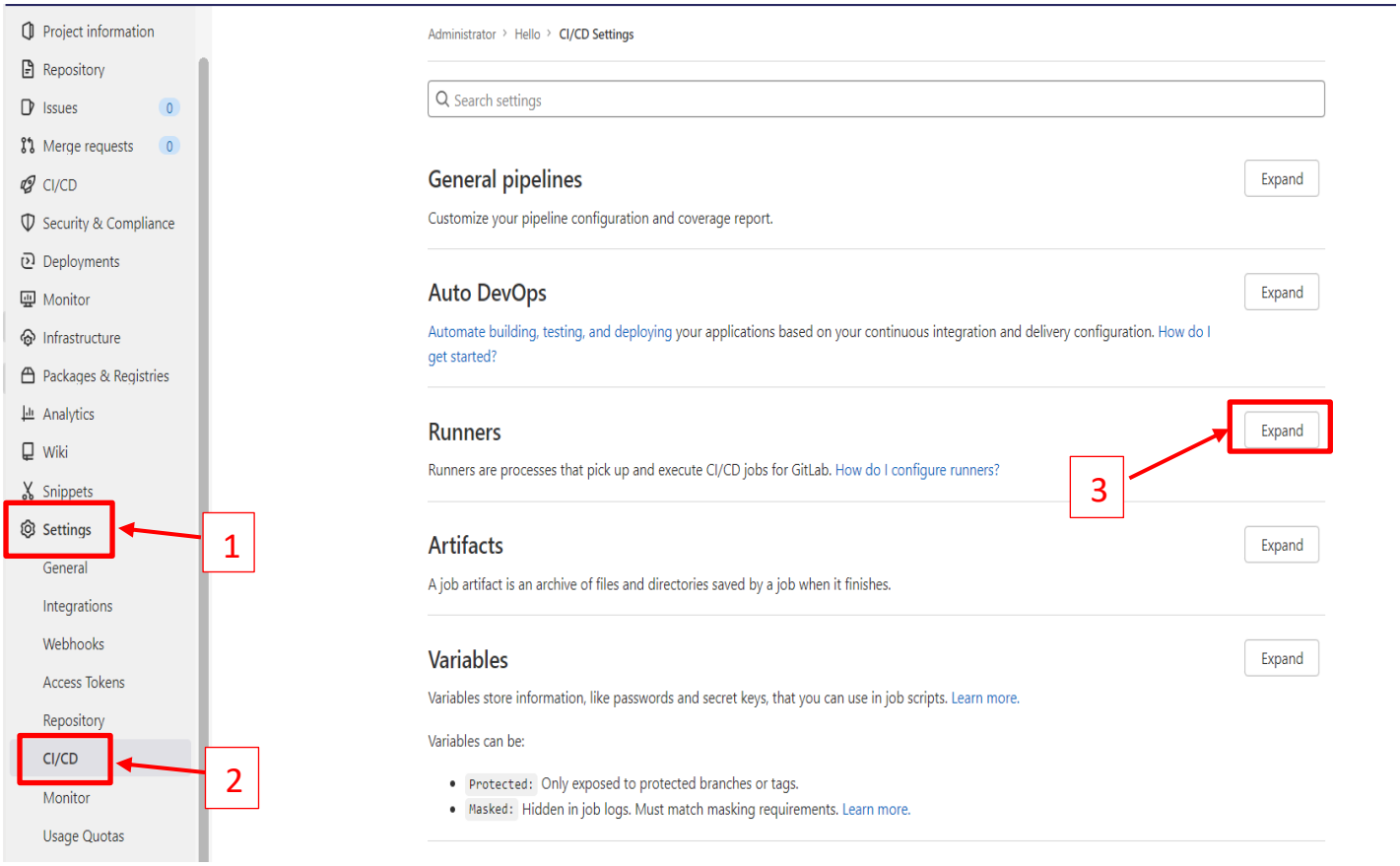
- Sau đó nhập **SSH_Key** đã copy và nhập các thông tin cần thiết sau đó nhấn **Add Key.**



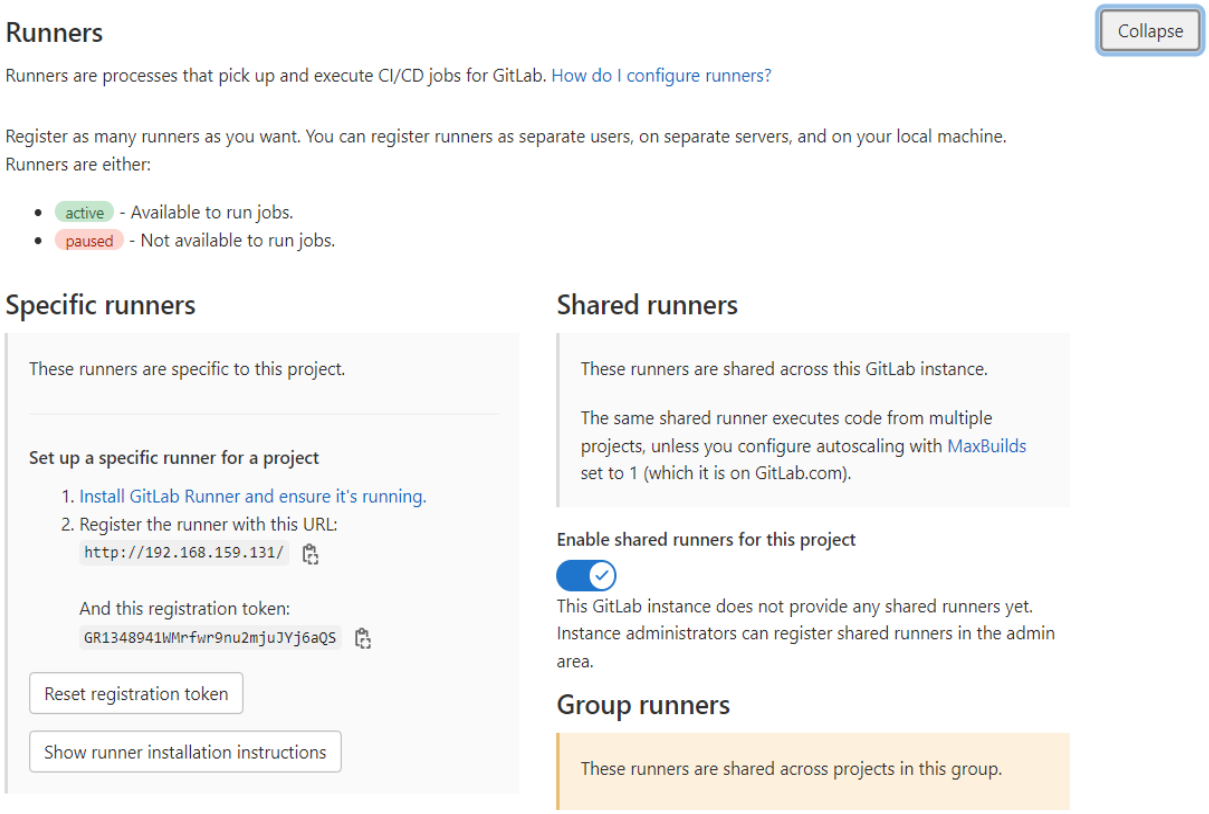
- Sau đó cũng tại gitlab Server từ project đã tạo mới để nhận các thông tin cần thiết để register cho Gitlab Runner.



- Tiếp theo nhấn **Settings** và chọn vào **CI/CD** và mục bên phải ngay Runners chọn **Expand**



- Ta sẽ nhận được link **URL** và **Registeration** để register cho Gitlab Runner:



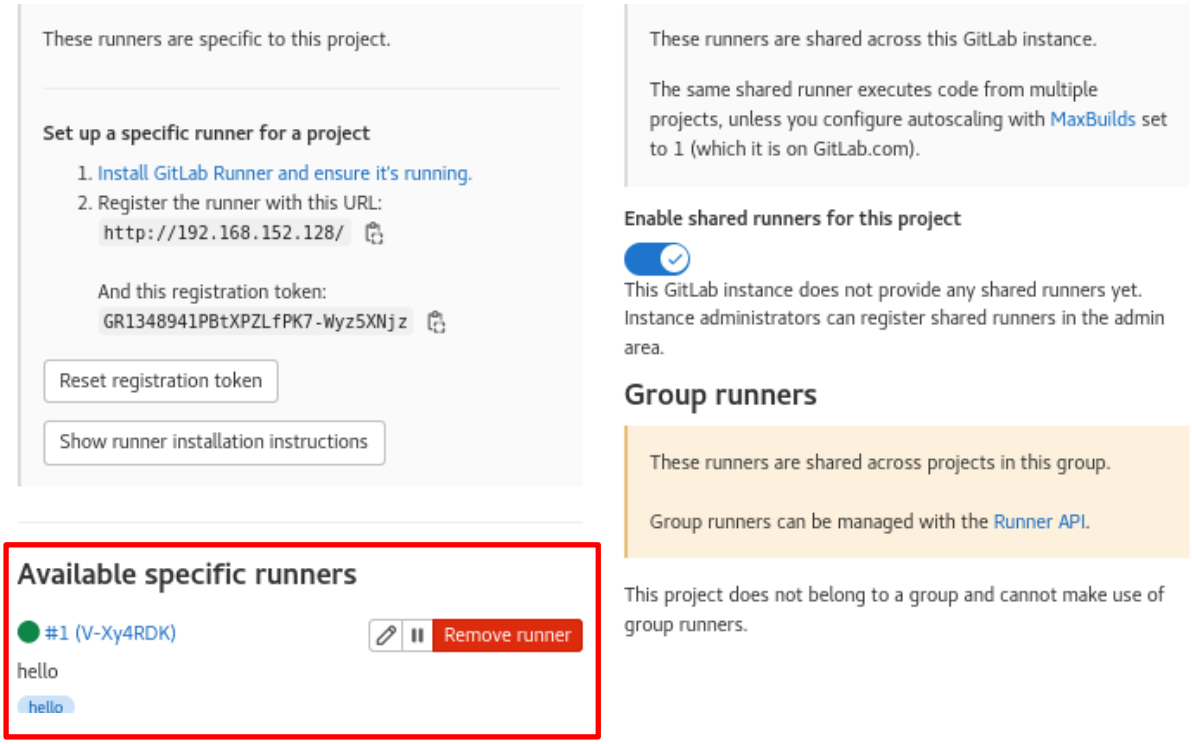
- Tiếp theo ta mở Gitlab Runner tại cửa sổ Terminal của Centos:

  Nhập lệnh: **# sudo gitlab-runner register**

- Và nhập các URL và Token như hướng dẫn trên. Ngoài ra các thông số còn lại có thể khai báo khác nhưng ở đây Excutor mình chọn là thực hiện với **docker** và có thể chọn **shell**

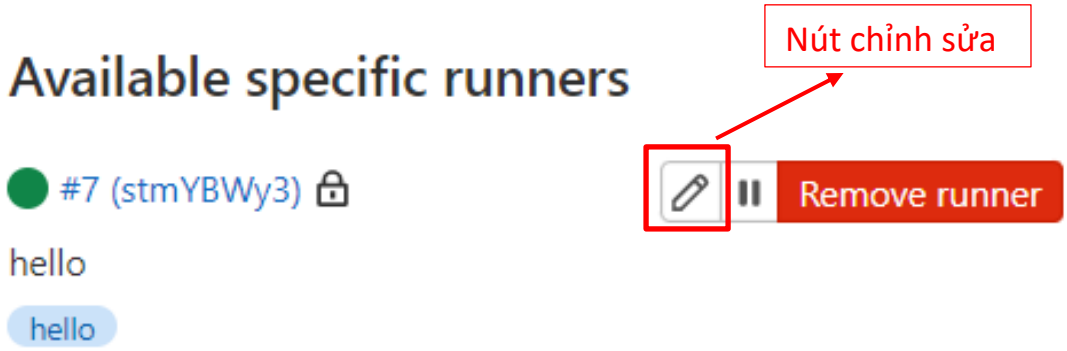
```
[root@localhost ~]# sudo gitlab-runner register
Runtime platform                                    arch=amd64 os=linux pid=4196
6 revision=c6e7e194 version=14.8.2
Running in system-mode.

Enter the GitLab instance URL (for example, https://gitlab.com/):
http://192.168.159.131/
Enter the registration token:
GR1348941WMrfwr9nu2mjuJYj6aQS
Enter a description for the runner:
[localhost.localdomain]: hello
Enter tags for the runner (comma-separated):
hello
Enter optional maintenance note for the runner:
hello
Registering runner... succeeded                     runner=GR134894
Enter an executor: docker, parallels, docker+machine, docker-ssh+machine, virtua
lbox, kubernetes, custom, docker-ssh, shell, ssh:
docker
Enter the default Docker image (for example, ruby:2.7):
centos:8
Runner registered successfully. Feel free to start it, but if it's running alrea
dy the config should be automatically reloaded!
[root@localhost ~]#
```

- Sau đó mở gitlab server và reload lại một Gitlab Runner đã được tạo.

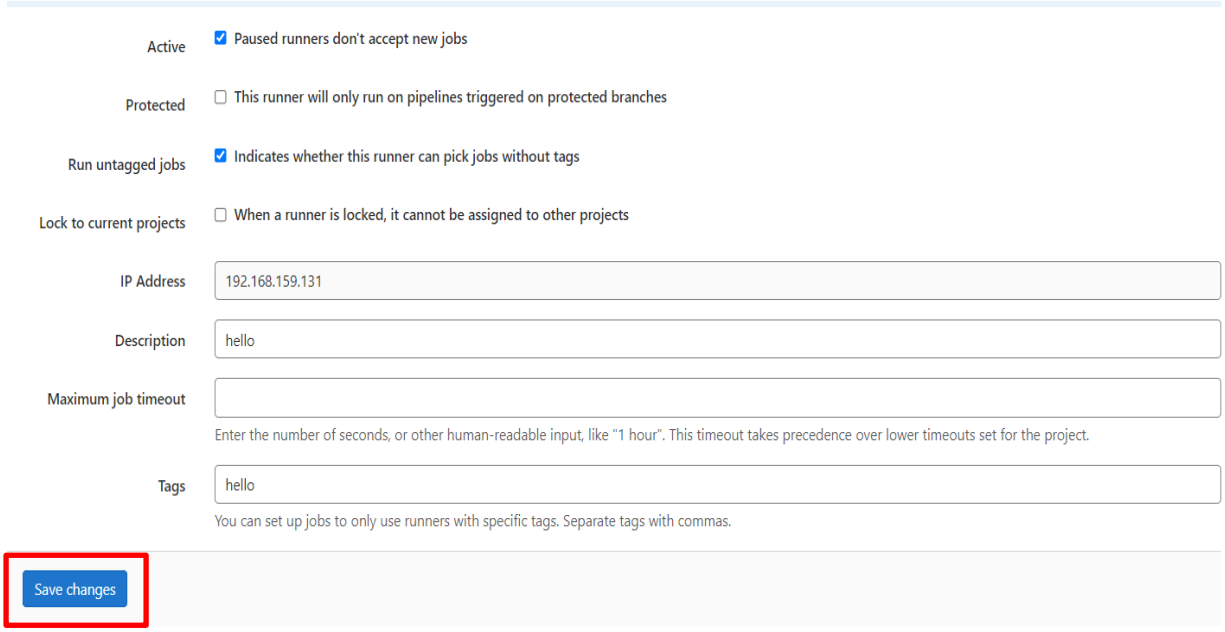


- Sau khi tạo xong sẽ xuất hiện Avaiable specific runners như trong hình.

- Sau đó nhấn vào nút chỉnh sửa để thay đổi các thông tin cần thiết như hình bên dưới tích chọn **Indicates**… và bỏ chọn **When**……và nhấn

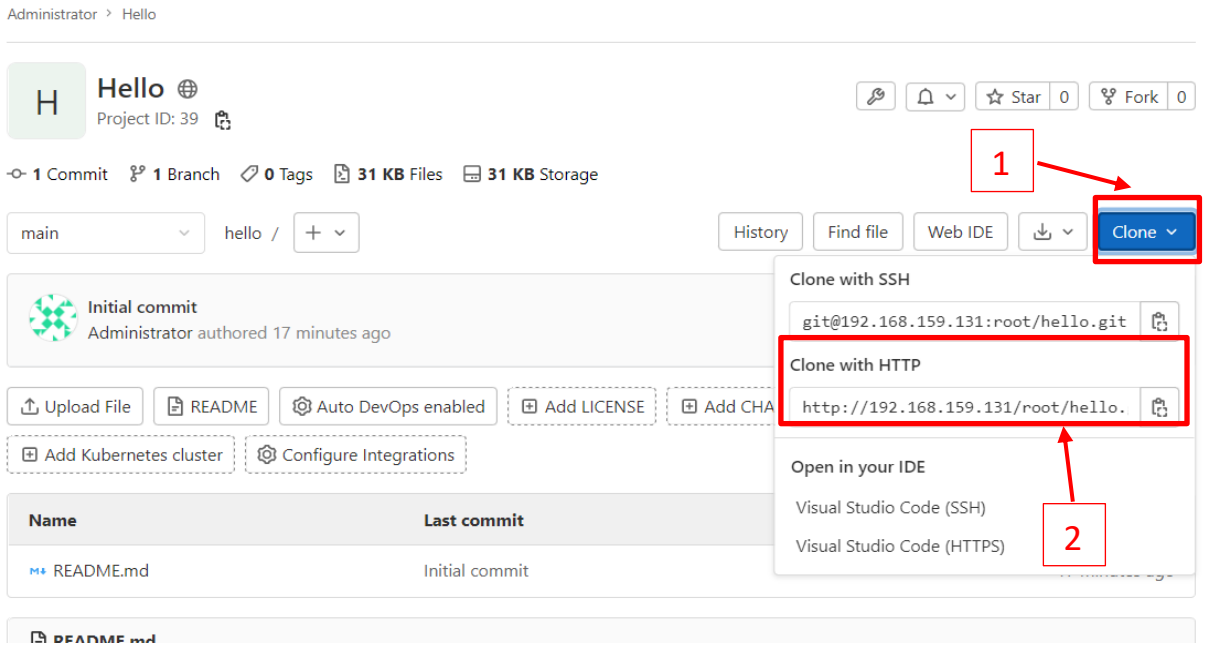


Save changes



c) Thực hiện clone project đã tạo về trên máy Centos

- Đầu tiên nhấn vào project và nhấn vào mũi tên bên cạnh Clone sẽ hiển thị link để chúng ta clone về. Ở đây mình chọn là HTTP.



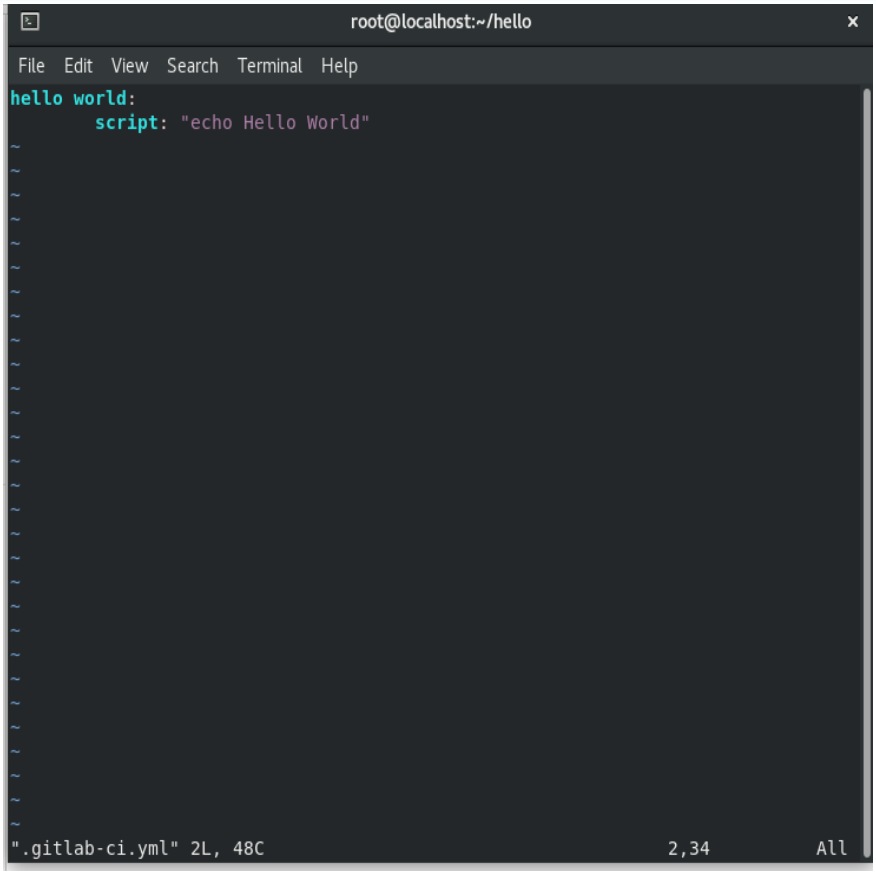
- Tiếp theo, ta thực hiện clone project về bằng cách lệnh sau trên cửa sổ Terminal của Centos:

```
[root@localhost ~]# git clone http://192.168.159.131/root/hello.git
Cloning into 'hello'...
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
Unpacking objects: 100% (3/3), 2.78 KiB | 2.78 MiB/s, done.
```

- Tiếp theo tạo file mới với tên là **.gitlab-ci.yml** trên máy Centos bằng các lệnh sau:
- Bằng cách trỏ đến thư mục bằng lệnh: **cd <tên_thư_mục>**
- Tiếp theo thêm file mới bằng cách: **vim .gitlab-ci.yml**

```
[root@localhost ~]# cd hello
[root@localhost hello]# vim .gitlab-ci.yml
```

- Sau đó nhấn chữ "I" để điền câu lệnh như hình và nhấn ESC sau đó nhập **:wq** để lưu và thoát.



- Sau đó thực hiện các lệnh sau như hình bên dưới:

git add .

git commit -m "Lời nhắn"

```
[root@localhost hello]# git add .
[root@localhost hello]# git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   .gitlab-ci.yml

[root@localhost hello]# git commit -m "add file"
[main 33d9e52] add file
 Committer: root <root@localhost.localdomain>
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, 2 insertions(+)
 create mode 100644 .gitlab-ci.yml
```

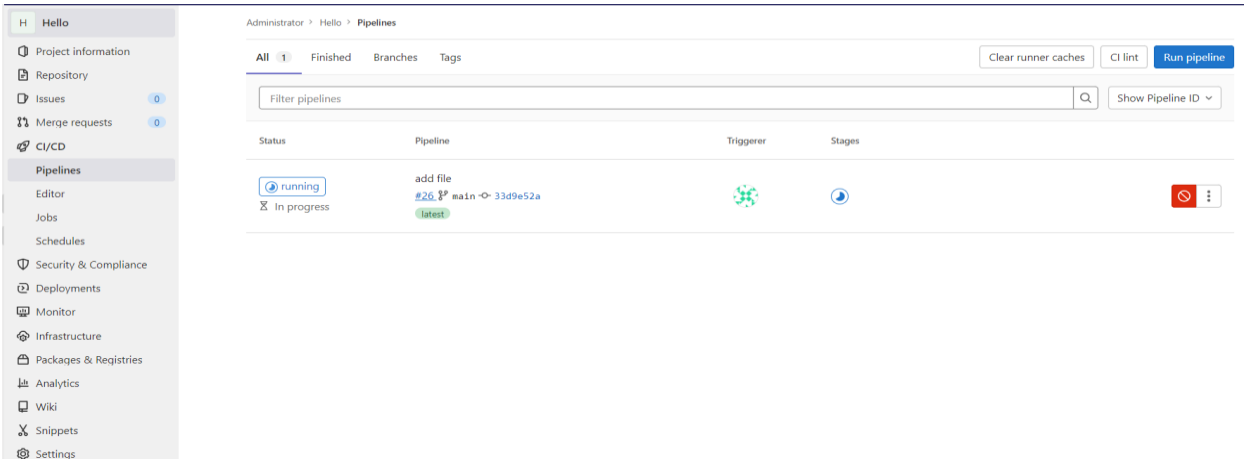
- Sau đó nhấn **git push** và nhập thông tin username và password để đẩy file vừa tạo lên project thành công

```
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 313 bytes | 313.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To http://192.168.159.131/root/hello.git
   b8758f8..33d9e52  main -> main
```

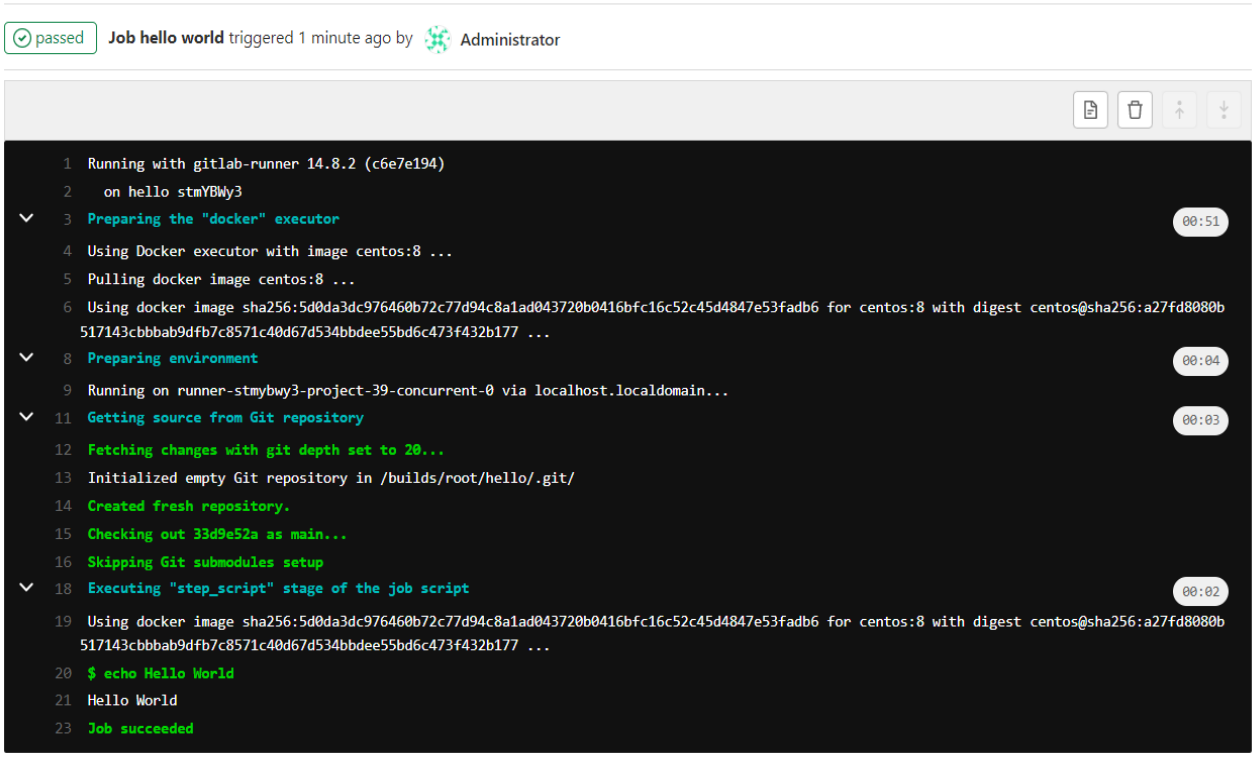
- Sau đó kiểm tra lại file trên project đã tạo tên gitlab Server bằng cách reload lại trang:

Name	Last commit	Last update
.gitlab-ci.yml	add file	3 minutes ago
README.md	Initial commit	30 minutes ago

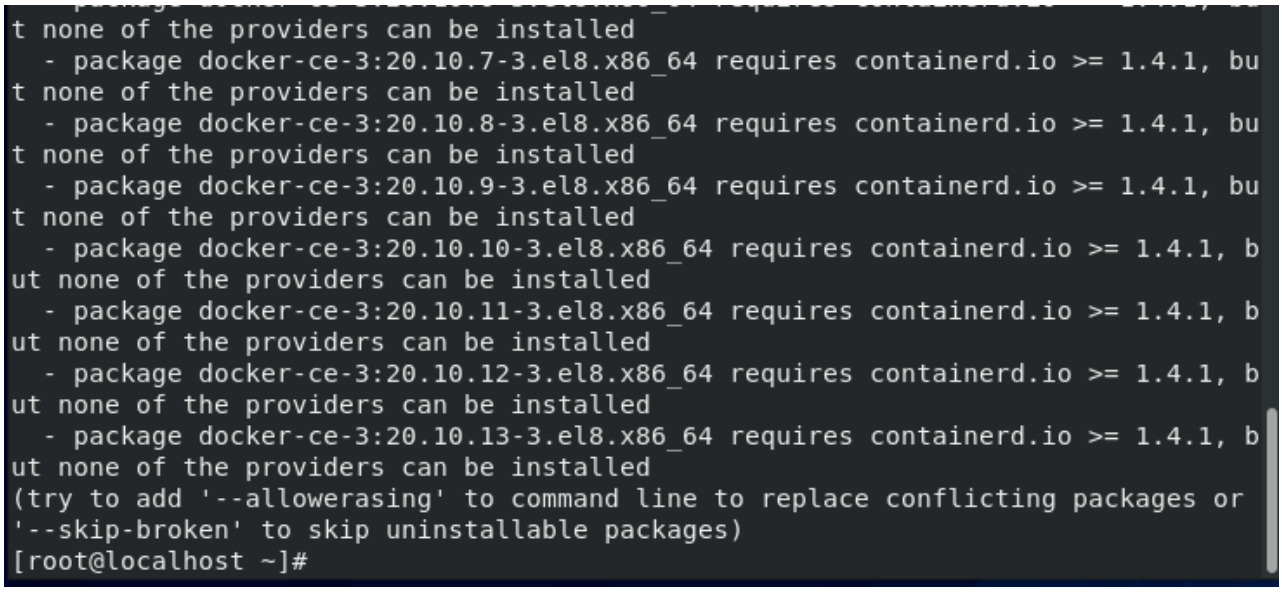
- File mới đã được tạo.
- Cuối cùng kiểm tra kết quả bằng cách nhấn vào CI/CD và chọn pipelines kết quả như hình bên dưới.



- Cuối cùng nhấn vào Jobs để xem các bước chạy và in thành công “Hello World”



Chú ý: ở đâu sử dụng docker excutor để thực thi nếu bạn chưa tải nó sẽ báo lỗi bạn cũng có thể thay đổi thành shell ở bước register gitlab-runner để không gặp lỗi hoặc nếu muốn cài docker để sử dụng về sau có thể tham khảo các bước cài sau: https://www.liquidweb.com/kb/how-to-install-docker-on-centos-8/



- Nếu gặp lỗi ở bước này chỉ cần nhập lại:

  dnf install docker-ce --nobest –allowerasing

- Khi đó, Gitlab sẽ bỏ qua các lỗi và tiếp tục cài đặt docker:

```
 Verifying        : docker-ce-rootless-extras-20.10.13-3.el8.x86_64       6/12
 Verifying        : docker-scan-plugin-0.17.0-3.el8.x86_64                7/12
 Verifying        : buildah-1.22.3-2.module_el8.5.0+911+f19012f9.x86_6    8/12
 Verifying        : cockpit-podman-33-1.module_el8.5.0+890+6b136101.no    9/12
 Verifying        : containers-common-2:1-2.module_el8.5.0+890+6b13610   10/12
 Verifying        : podman-3.3.1-9.module_el8.5.0+988+b1f0b741.x86_64    11/12
 Verifying        : podman-catatonit-3.3.1-9.module_el8.5.0+988+b1f0b7   12/12

nstalled:
 containerd.io-1.5.10-3.1.el8.x86_64
 docker-ce-3:20.10.13-3.el8.x86_64
 docker-ce-cli-1:20.10.13-3.el8.x86_64
 docker-ce-rootless-extras-20.10.13-3.el8.x86_64
 docker-scan-plugin-0.17.0-3.el8.x86_64
 libcgroup-0.41-19.el8.x86_64
emoved:
 buildah-1.22.3-2.module_el8.5.0+911+f19012f9.x86_64
 cockpit-podman-33-1.module_el8.5.0+890+6b136101.noarch
 containers-common-2:1-2.module_el8.5.0+890+6b136101.noarch
 podman-3.3.1-9.module_el8.5.0+988+b1f0b741.x86_64
 podman-catatonit-3.3.1-9.module_el8.5.0+988+b1f0b741.x86_64

omplete!
root@localhost ~]#
```

- Sau đó chúng ta nhập: **systemctl enable --now docker** để bật docker lên.
- Tiếp tục nhập để confirm docker: **systemctl status docker.**
- Sau đó, add user là root để cấp quyền thực thi docker: **usermod -aG docker root.**