GIT LAB CICD PIPE-LINE

Dockerhub:

userName: [pusamsaikumar302@gmail.com](mailto:pusamsaikumar302@gmail.com)

password: VK\_SE5kK\*Gjb2F&

GitLab:

Username: [pusamsaikumar302@gmail.com](mailto:pusamsaikumar302@gmail.com)

Password: saikumarP123

Prequisites:

1. Create gitlab account
2. Create docker file in asp.net core web api project
3. Setting up variable for the docker connection

cd existing\_repo

git remote add origin https://gitlab.com/gitlabtest2562326/testdockerapp.git

git branch -M main

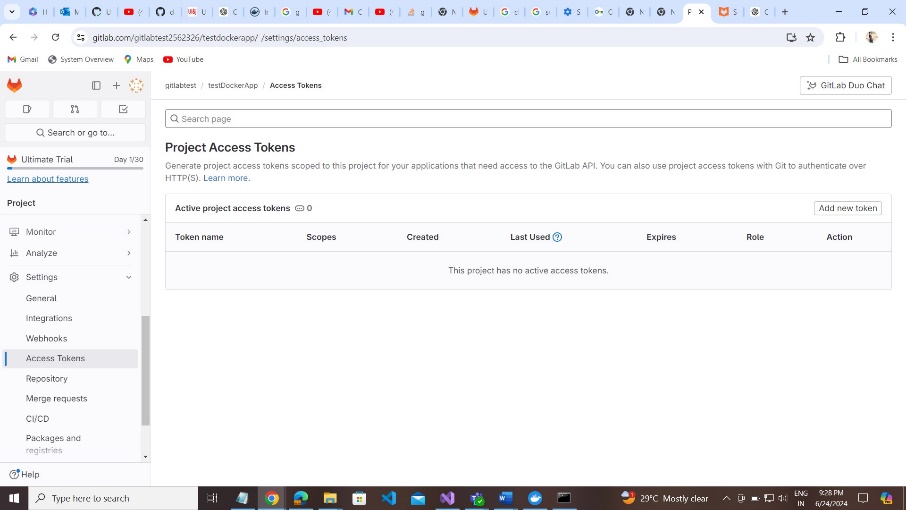
git push -uf origin main

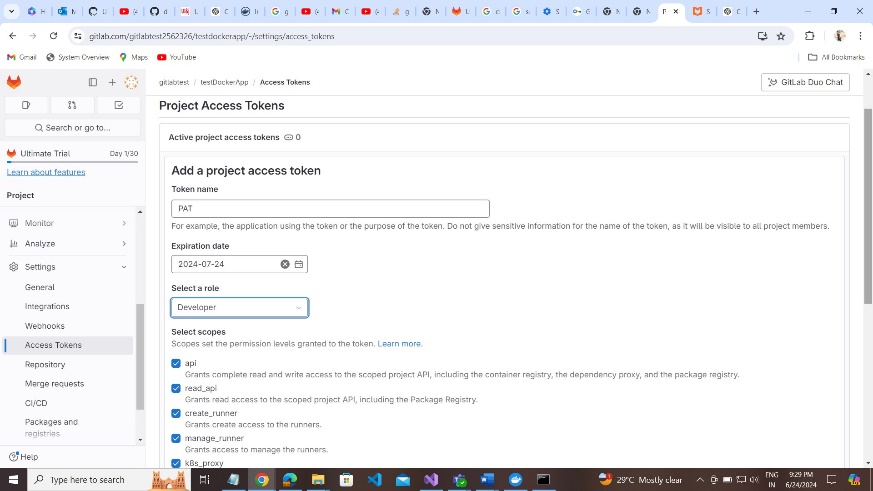
4.Create Personal access token:

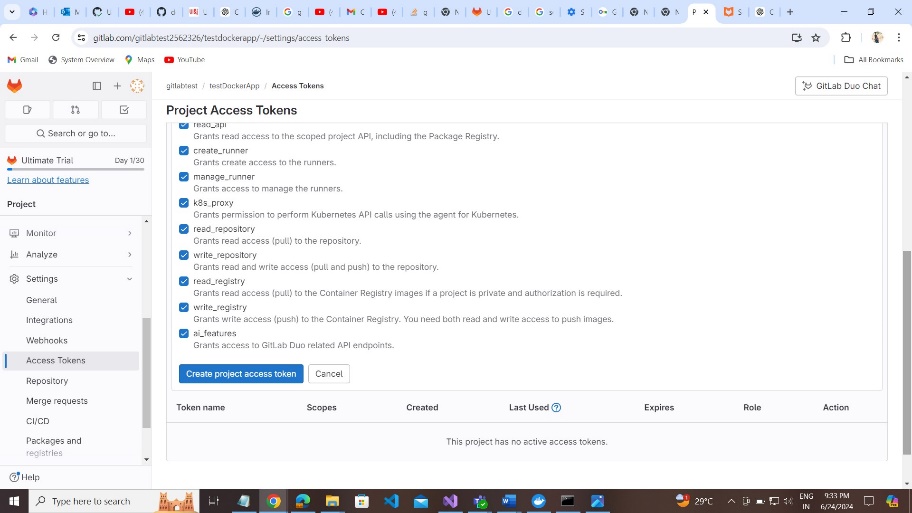
 Go to your GitLab profile.

 Navigate to Settings > Access Tokens.

 Create a new personal access token with the read\_registry scope.

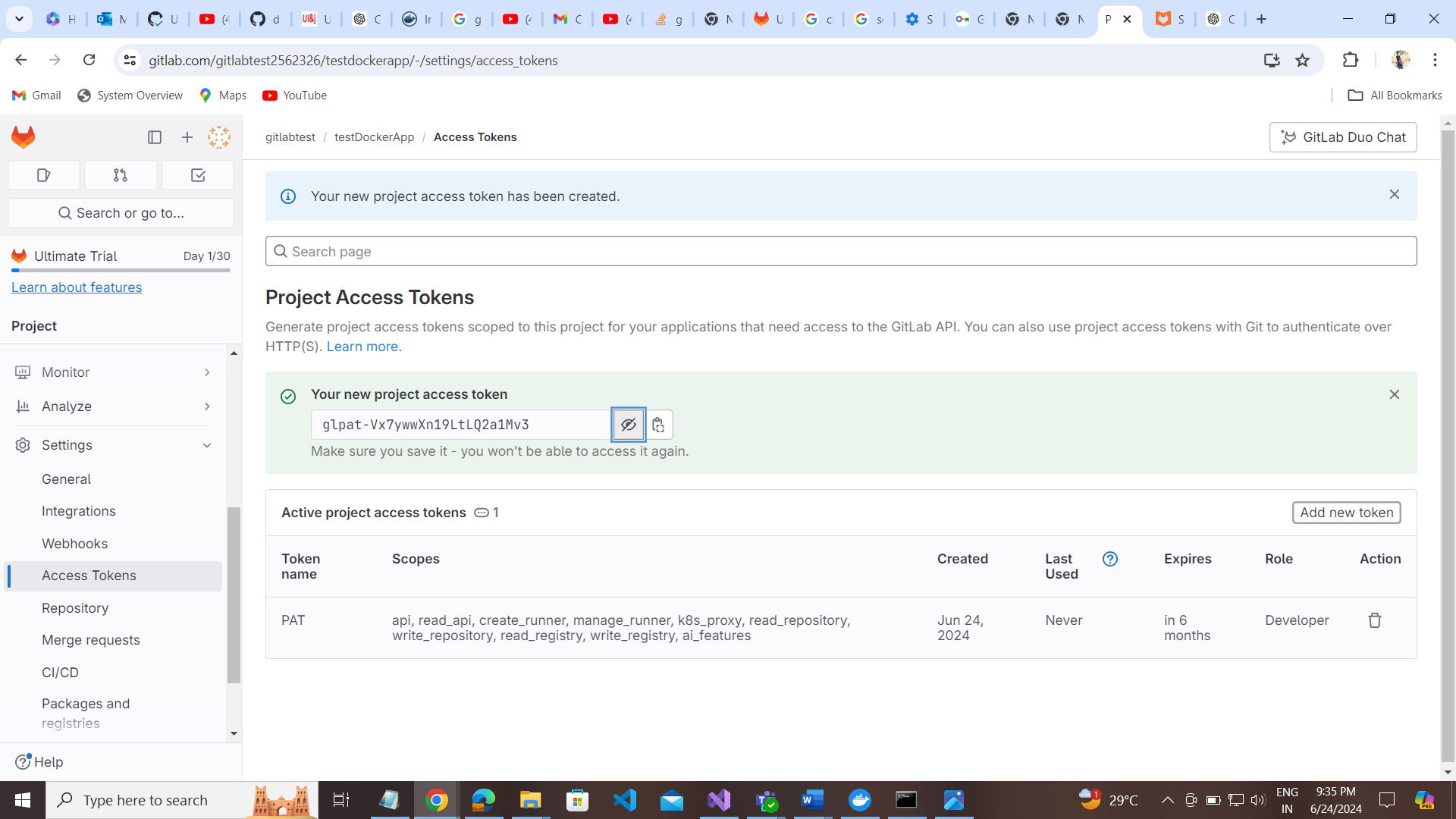






Generated access token:

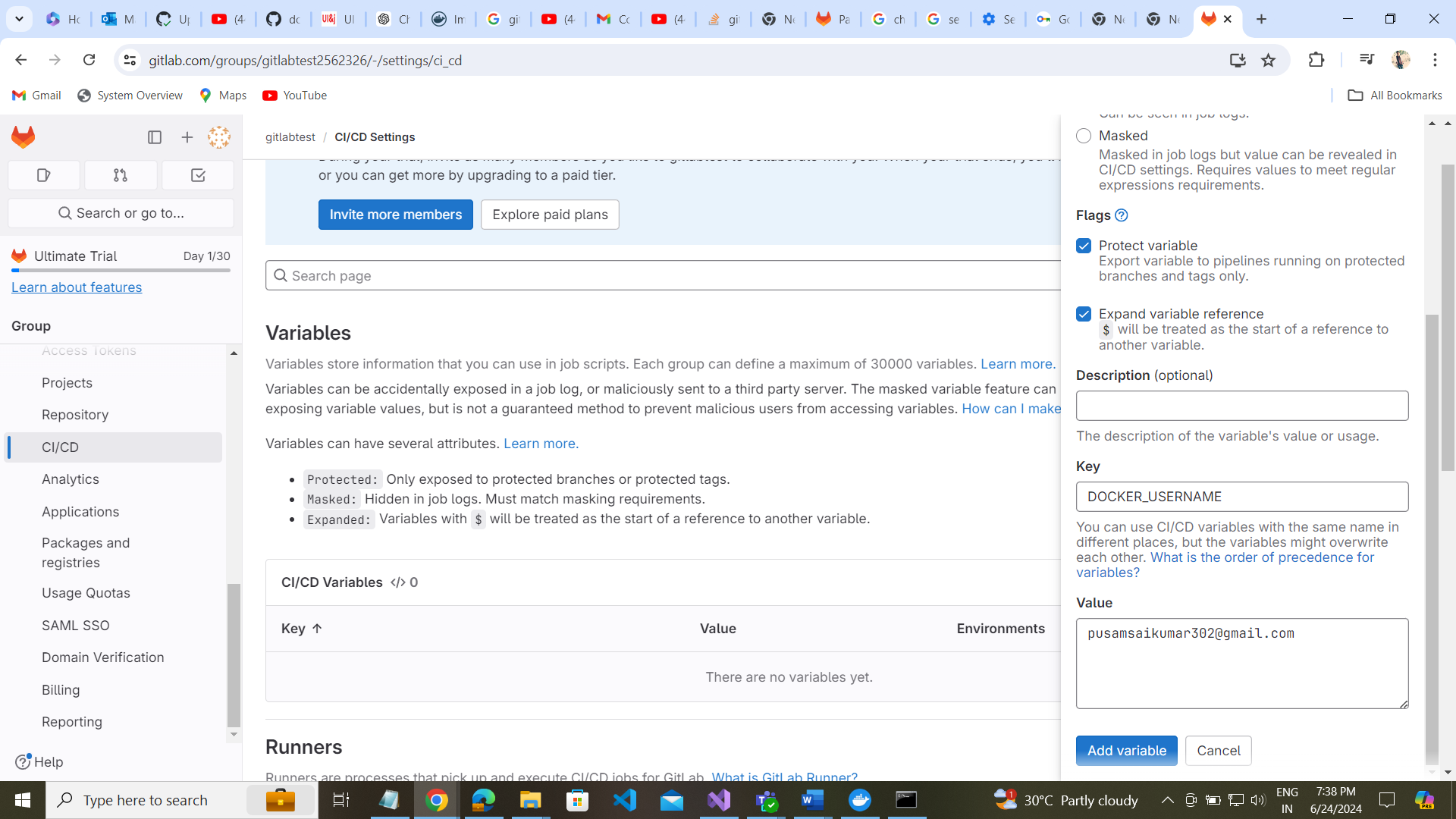
PAT: glpat-Vx7ywwXn19LtLQ2a1Mv3

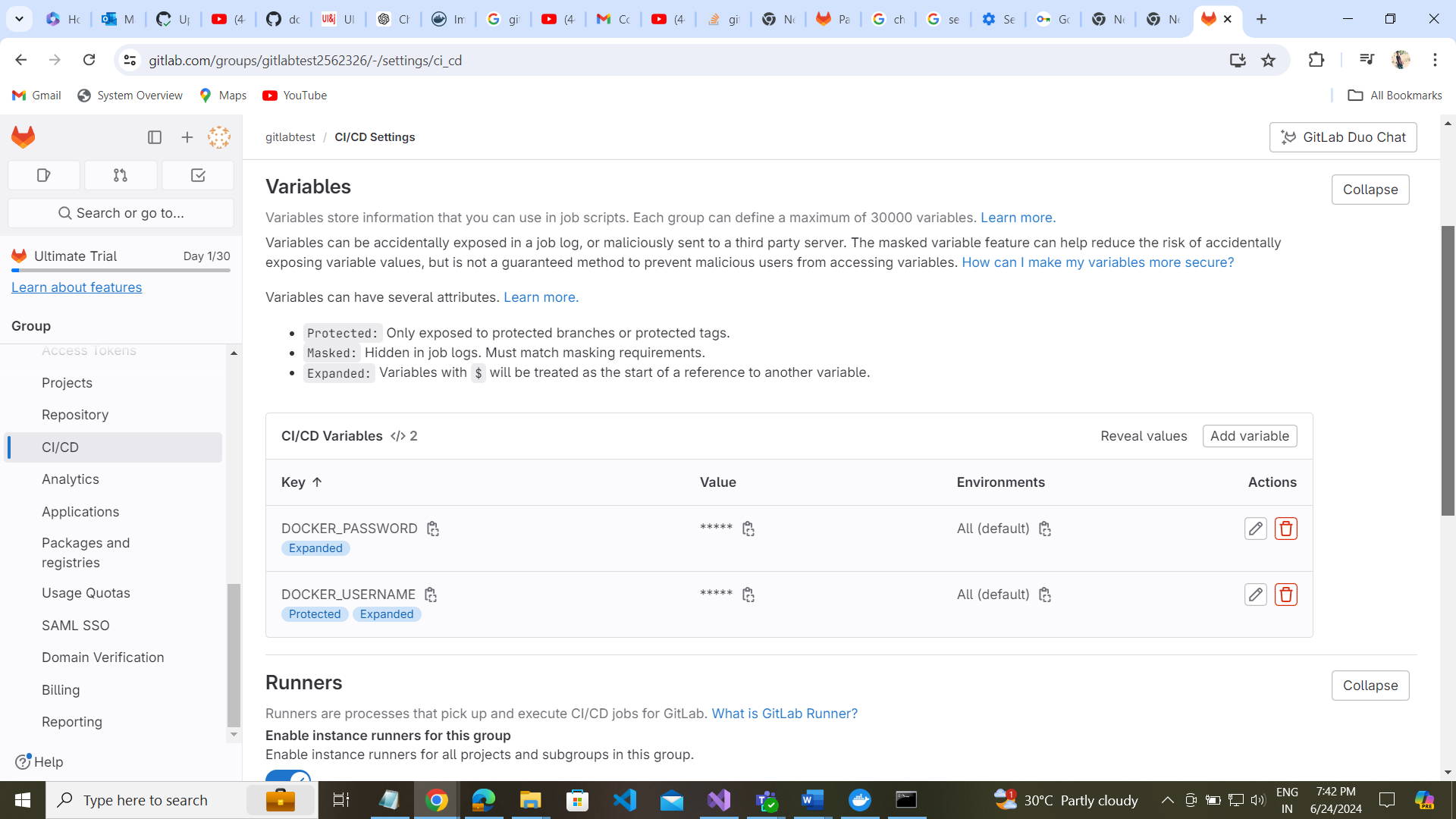


https://gitlab.com/gitlabtest2562326/testdockerapp.git

Go to settings ---- > CICD ---- > Variables  
add username and password of docker hub.

Container-Registry (registry.gitlab.com)

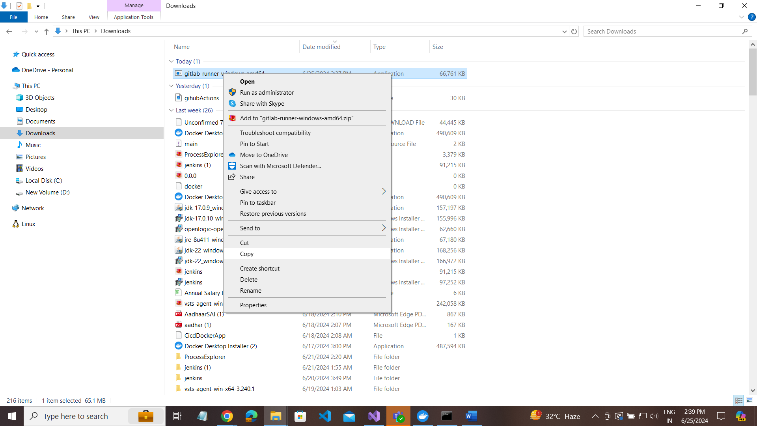


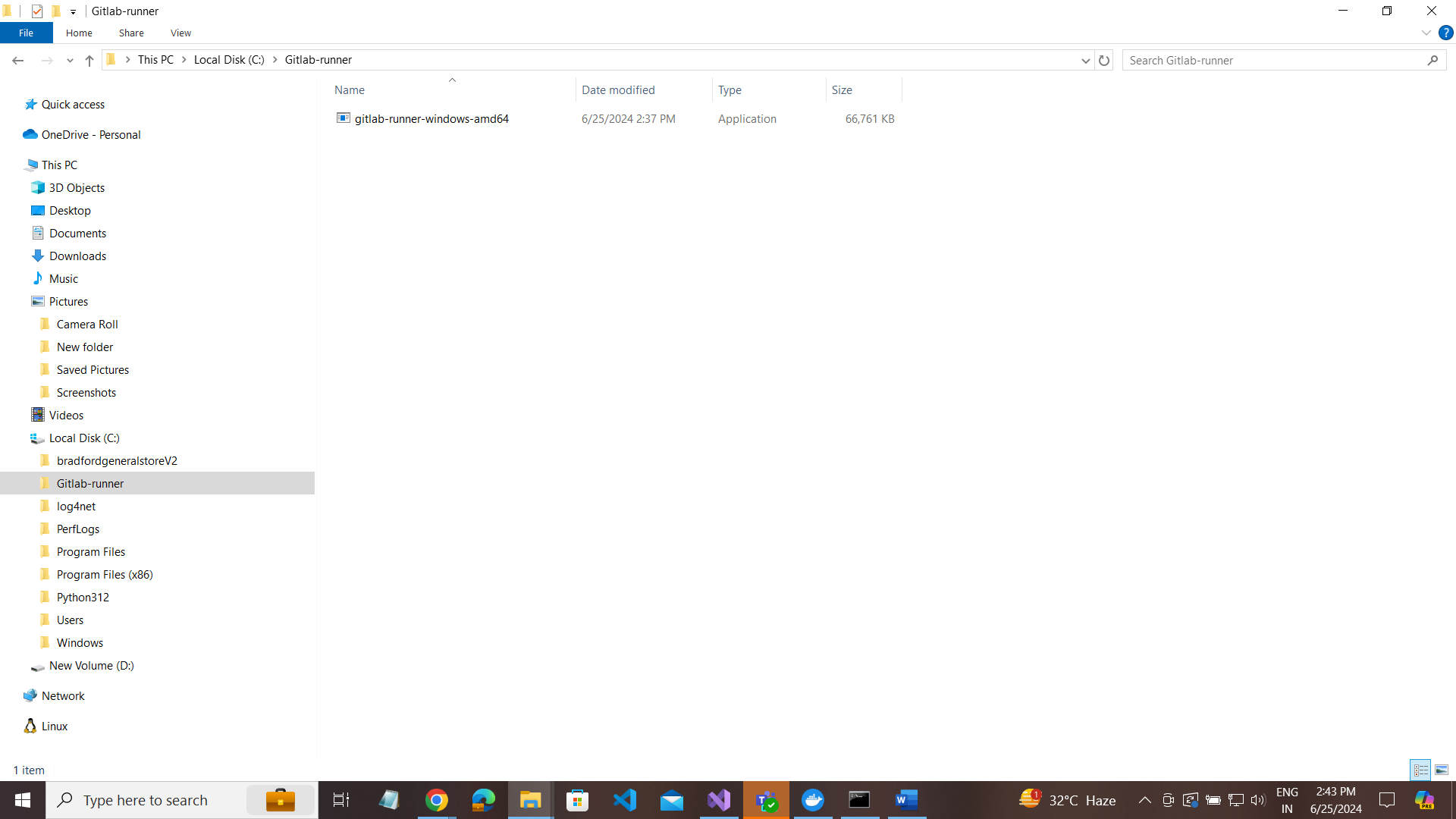
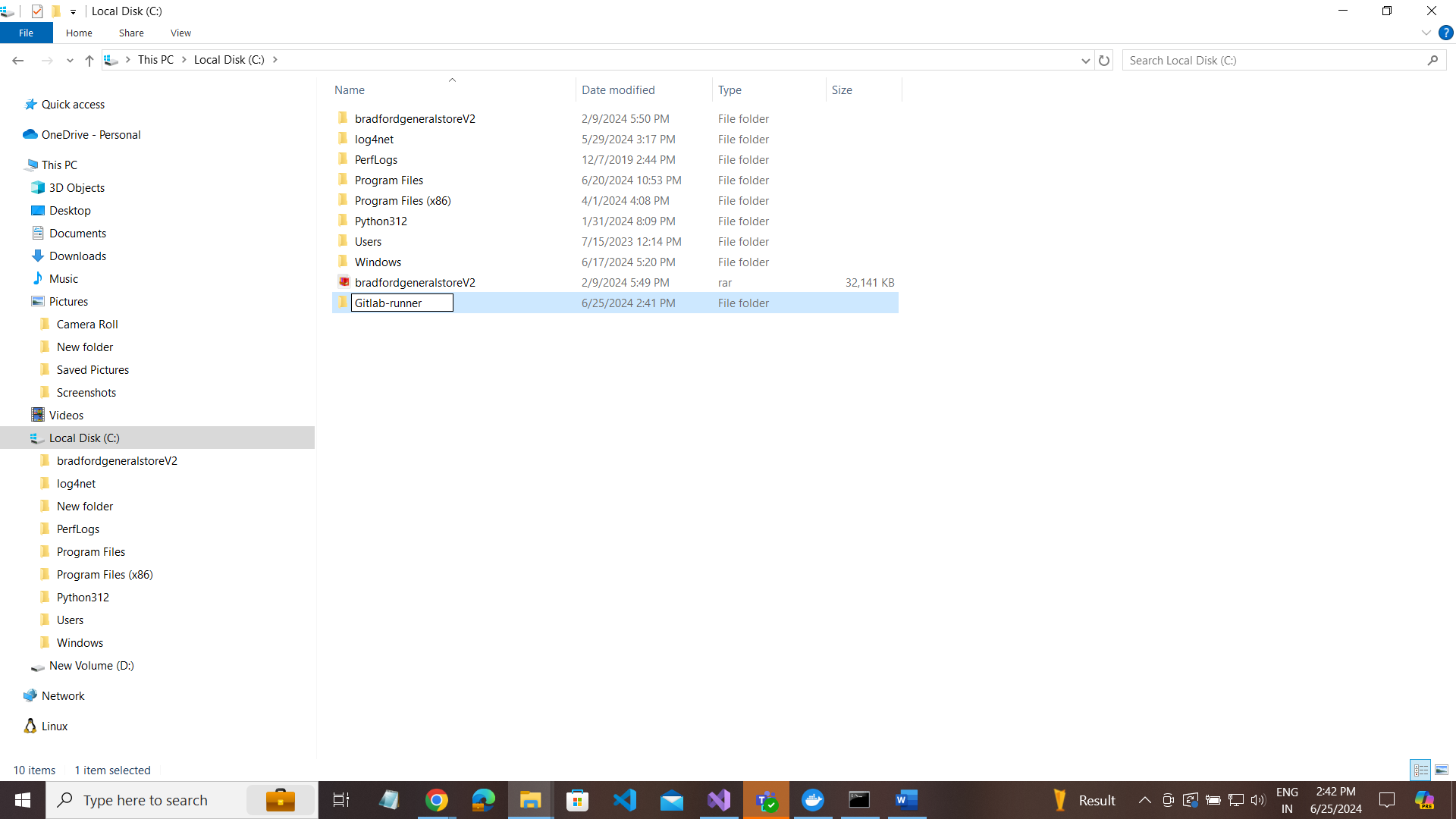


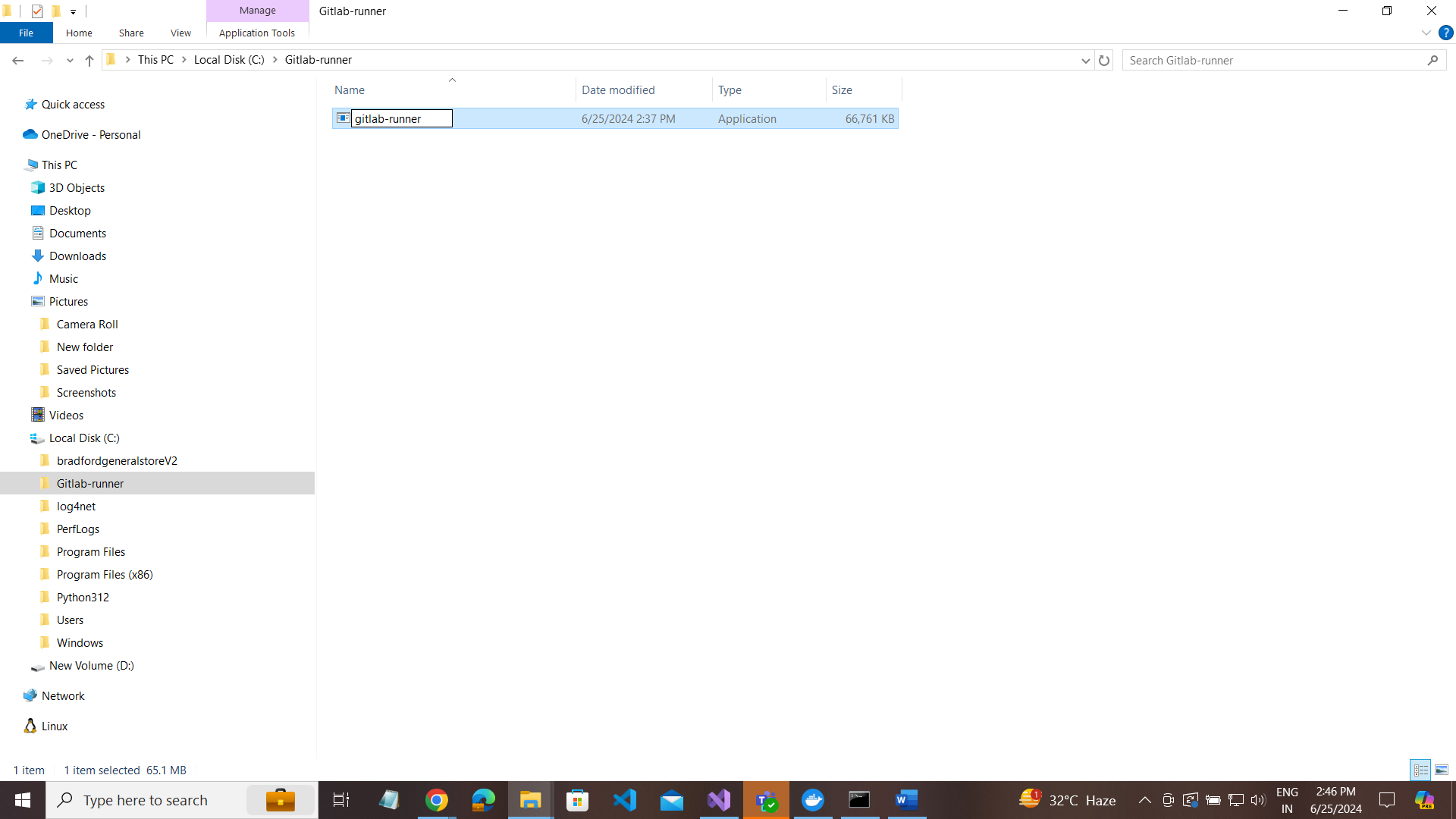
Install git lab runnder on windows :

url: <https://docs.gitlab.com/runner/install/windows.html>

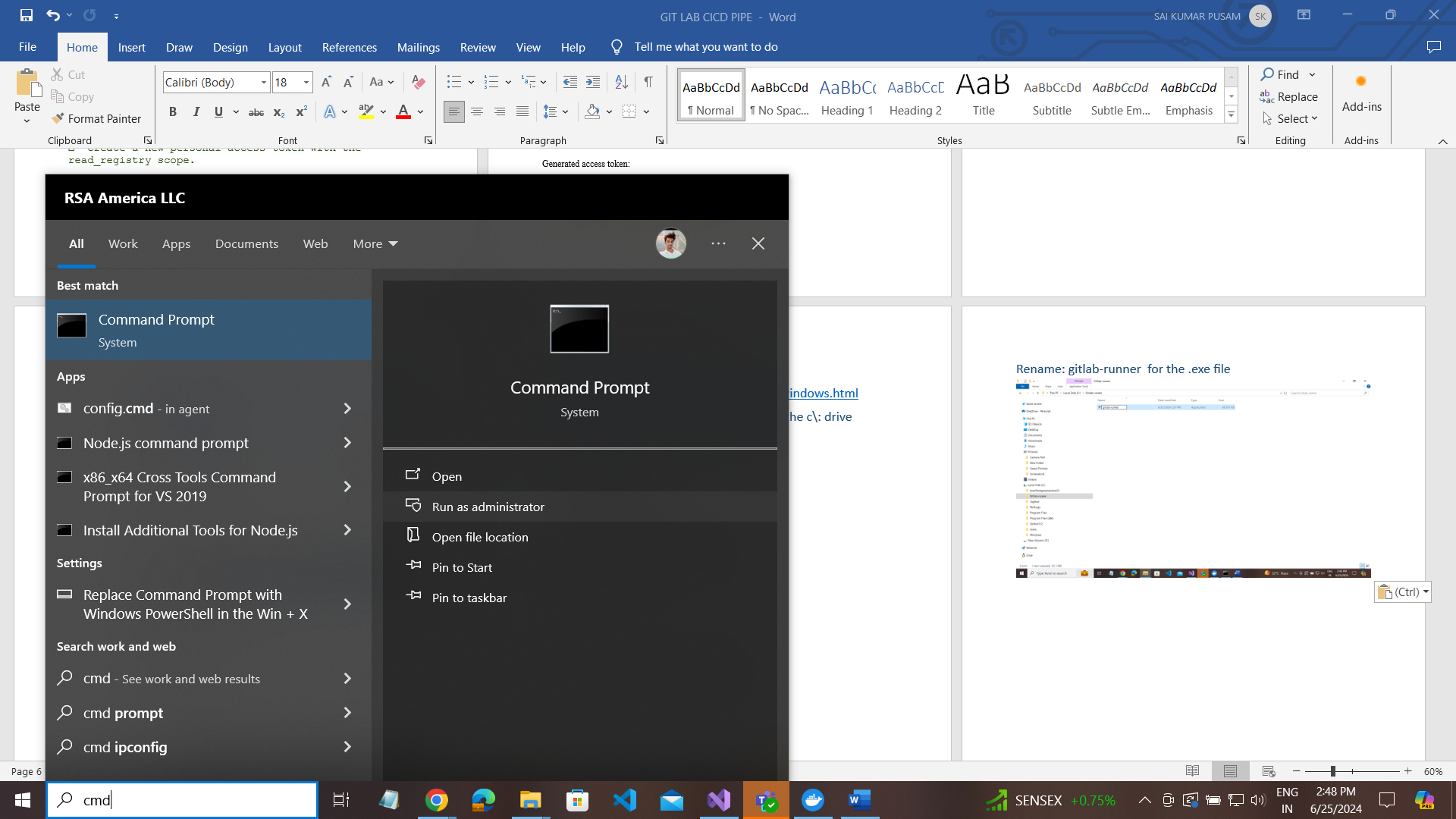
copy the gitlab runner exe file and paste on the c\: drive

with specific folder-name Gitlab-runner  




Rename: gitlab-runner for the .exe file

Open command prompt as Administrator:



Run the following commands to install gitlab-runner

C:\Windows\system32>cd c:\

c:\>cd c:\gitlab-runner

c:\Gitlab-runner>.\gitlab-runner.exe install

Runtime platform arch=amd64 os=windows pid=17088 revision=fe451d5a version=17.1.0

c:\Gitlab-runner>.\gitlab-runner.exe install

Runtime platform arch=amd64 os=windows pid=17088 revision=fe451d5a version=17.1.0

Register the Gitlab-runner:  
c:\Gitlab-runner>.\gitlab-runner.exe register

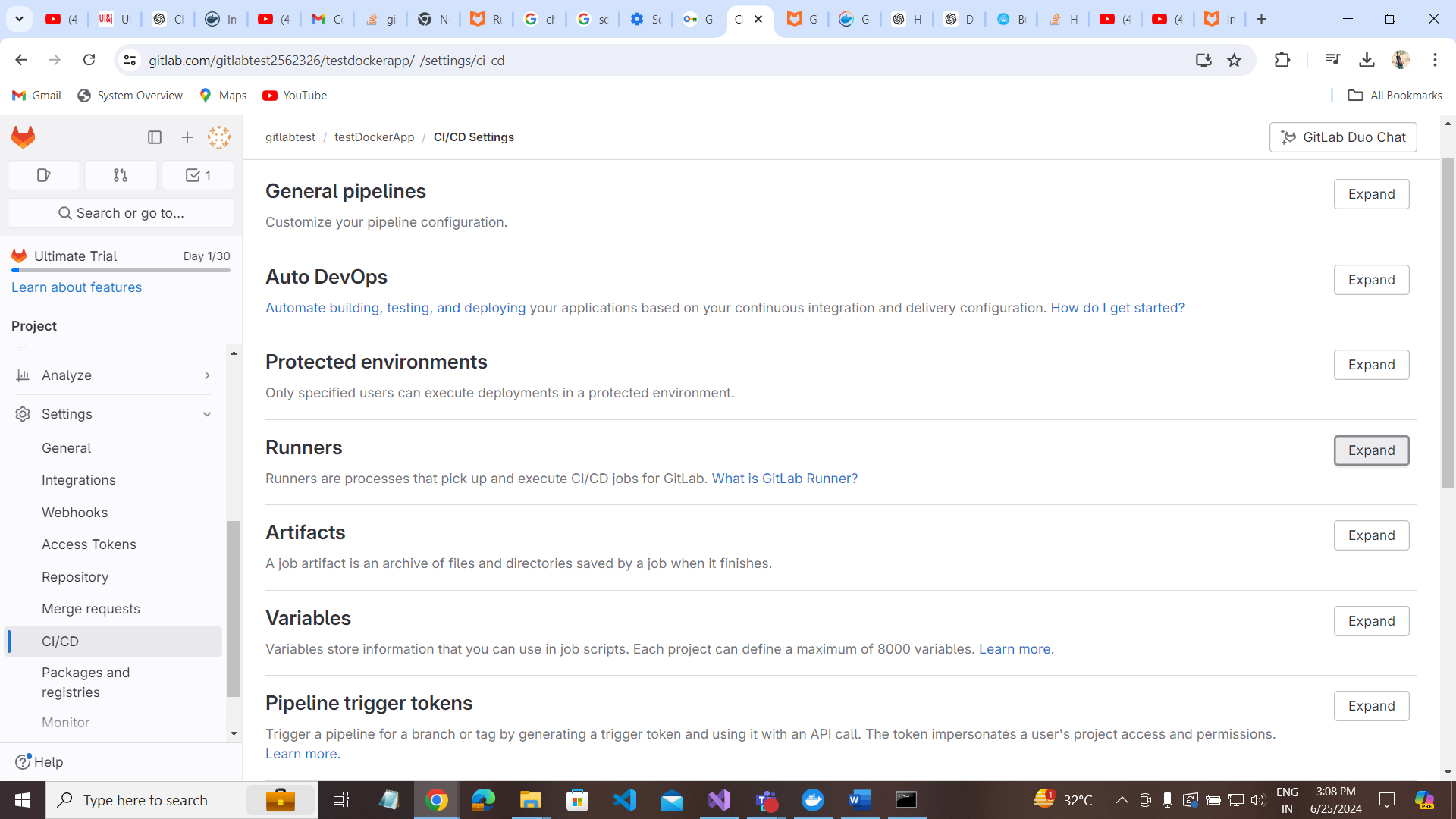
Runtime platform arch=amd64 os=windows pid=8844 revision=fe451d5a version=17.1.0

Enter the GitLab instance URL (for example, https://gitlab.com/):

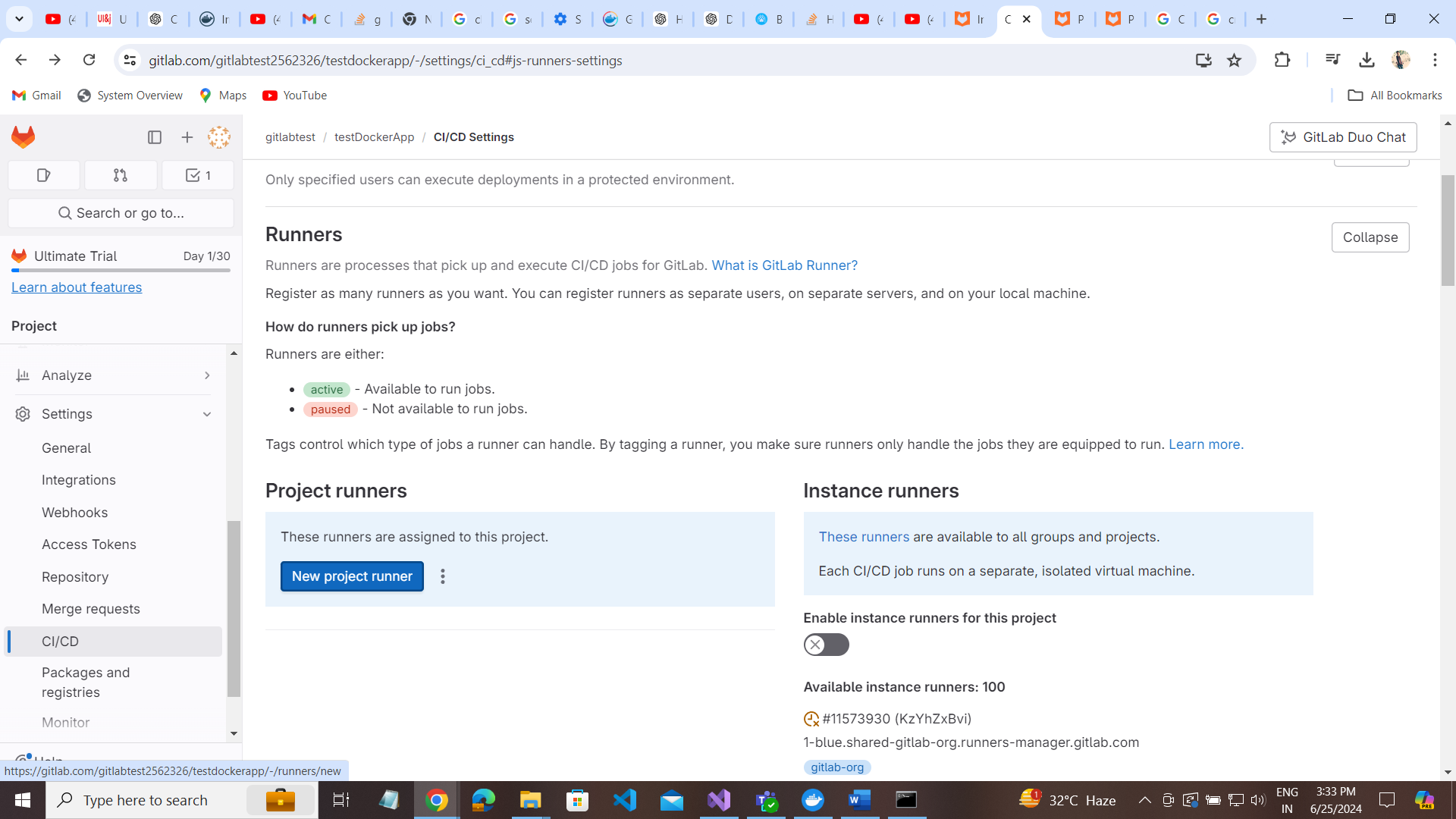
https://gitlab.com/

Enter the registration token:

For registration token , we have to login to GitLab accout  
copy from the your project using CICD Settings.  
we have go to settings --- > CICD --- > Runner - --- > Copy registration token by clicking dot lines.



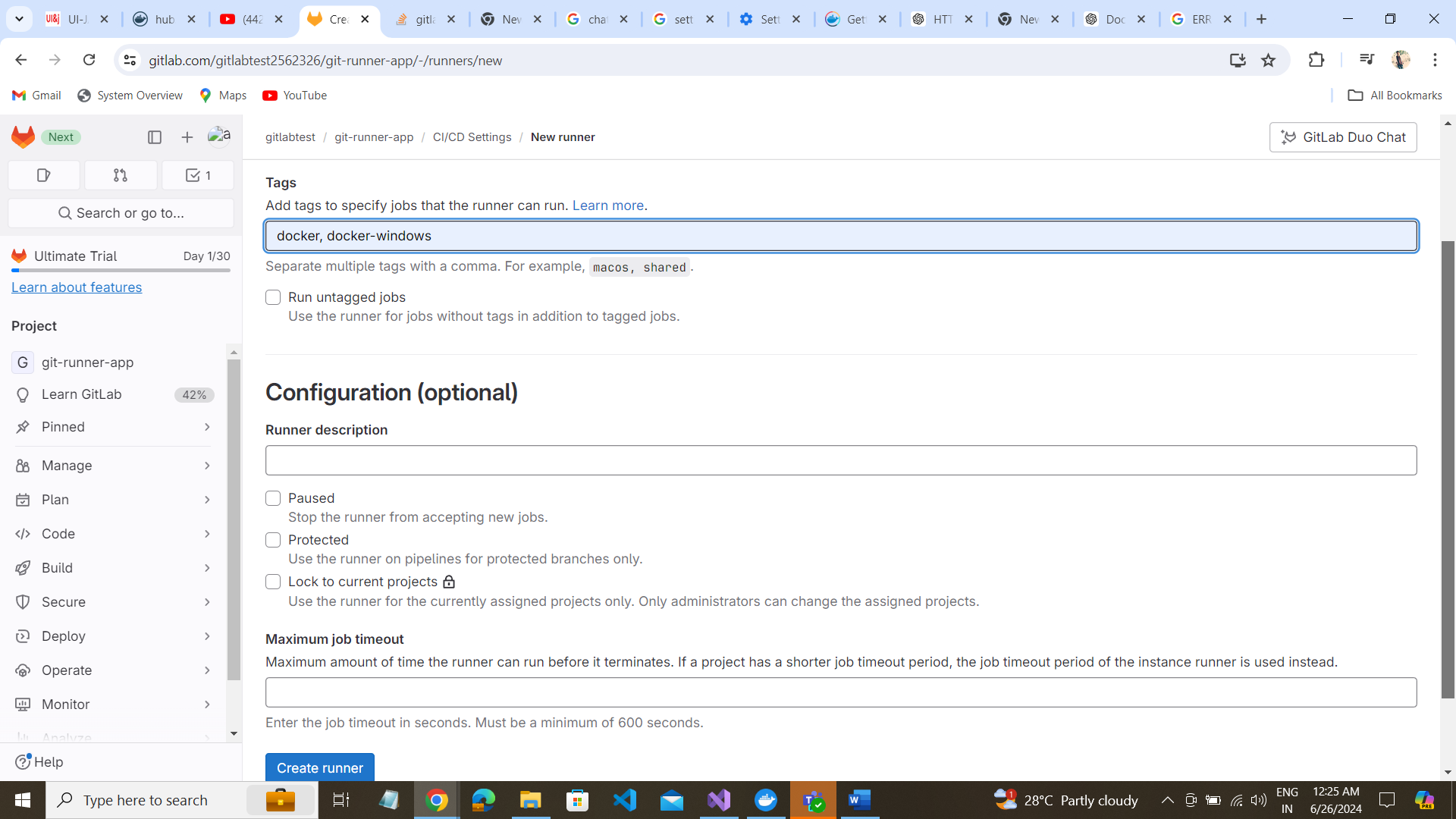
Go to runner page click on the New Project Runner

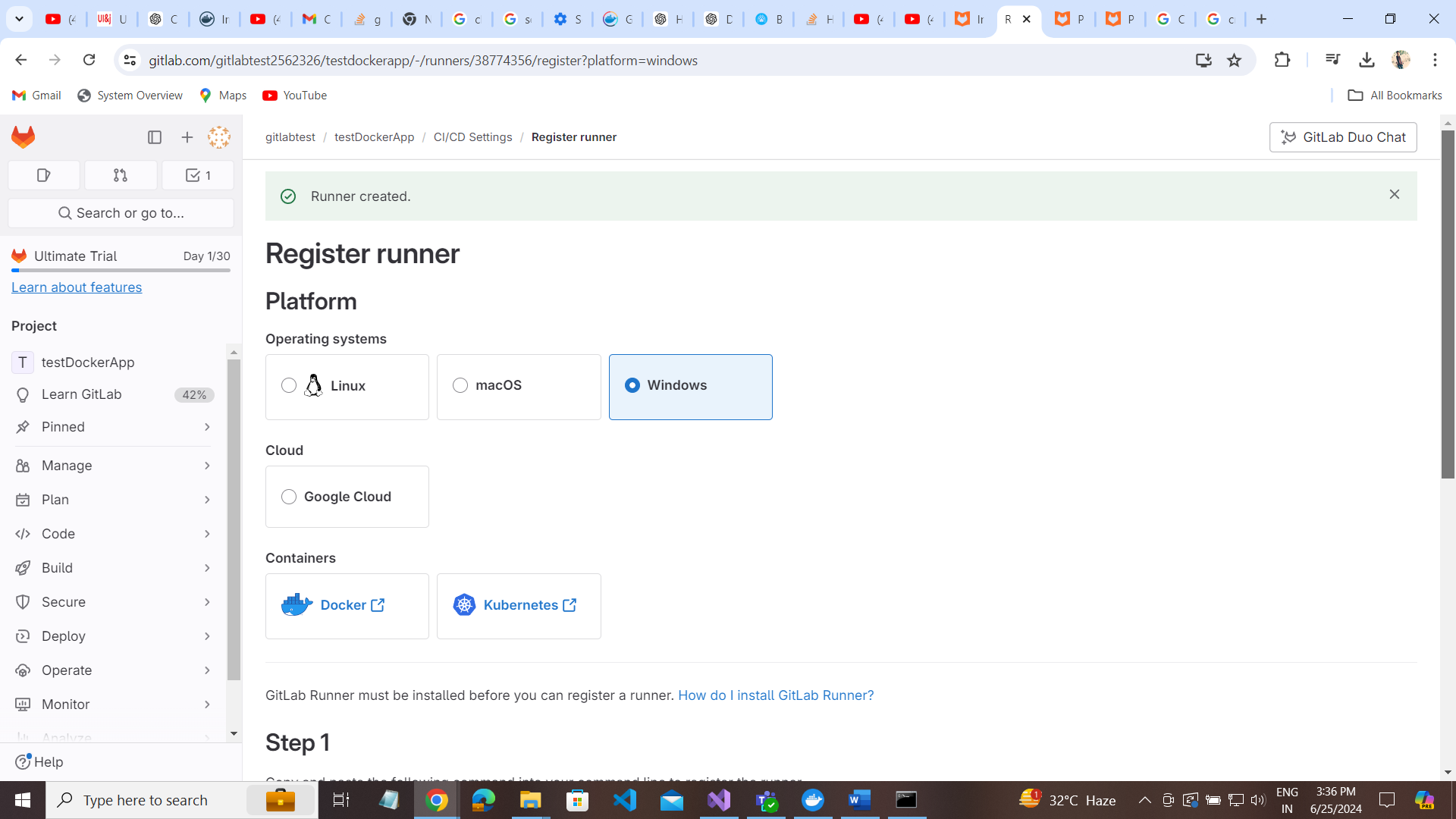


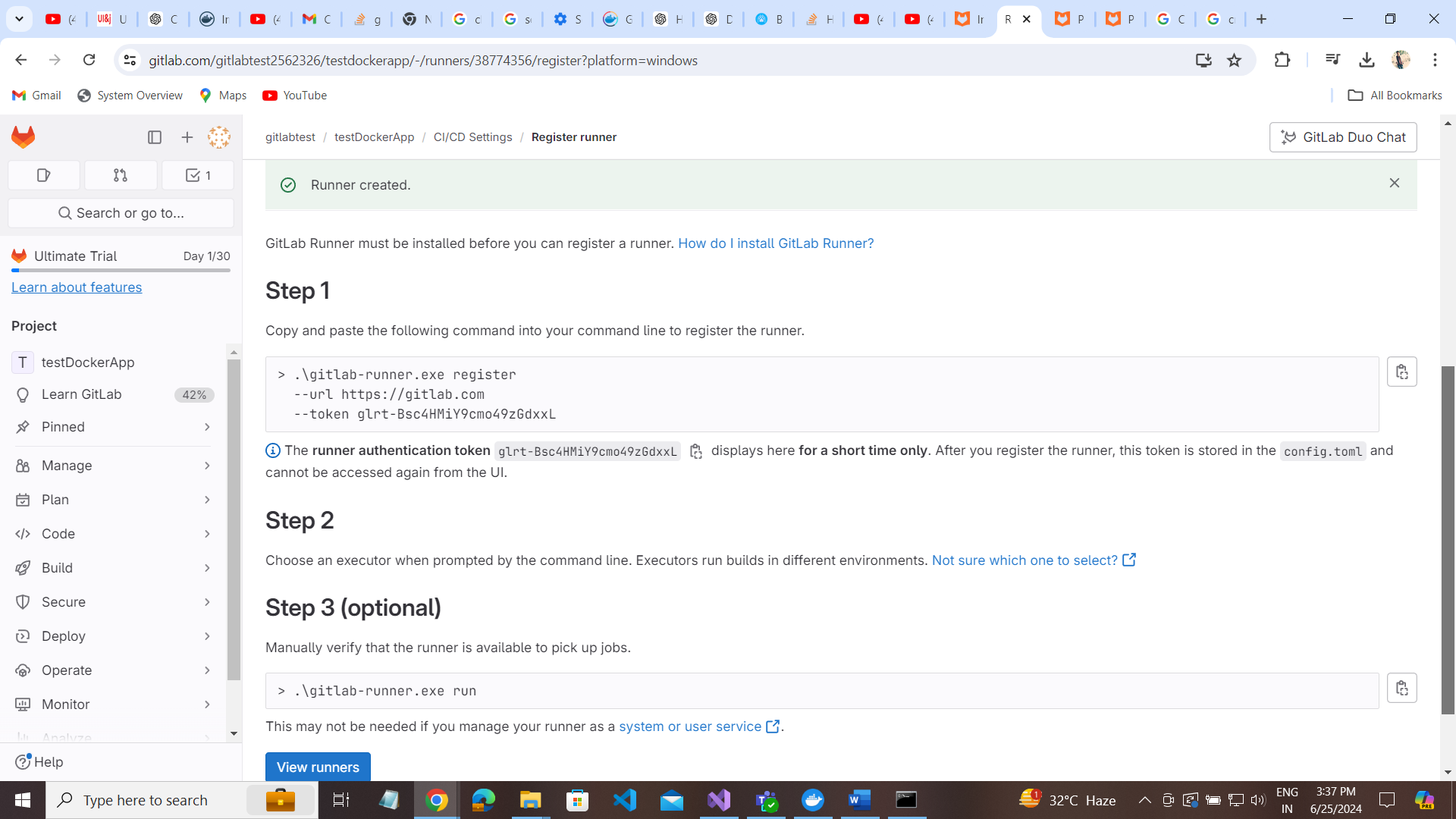
Add some tags name like windows , windows-2019, windows-2022, powershell, docker-windows, test, (comma-seperated)

docker+machine

add tags from New Project runner







install on windows 10:

c:\Gitlab-runner>.\gitlab-runner.exe register

Runtime platform arch=amd64 os=windows pid=26004 revision=fe451d5a version=17.1.0

Enter the GitLab instance URL (for example, https://gitlab.com/):

https://gitlab.com/

Enter the registration token:

glrt-NfzDJZyBk5vszPxfyypy

WARNING: A runner with this system ID and token has already been registered.

Verifying runner... is valid runner=NfzDJZyBk

Enter a name for the runner. This is stored only in the local config.toml file:

[DESKTOP-1JNARAI]: Project runner used

Enter an executor: ssh, parallels, docker, docker+machine, kubernetes, docker-autoscaler, instance, custom, shell, virtualbox, docker-windows:

docker-windows or docker+machine

Enter the default Docker image (for example, mcr.microsoft.com/windows/servercore:1809):

mcr.microsoft.com/windows/servercore:1809

Runner registered successfully. Feel free to start it, but if it's running already the config should be automatically reloaded!

Configuration (with the authentication token) was saved in "c:\\Gitlab-runner\\config.toml"

Execute runner

c:\Gitlab-runner>.\gitlab-runner.exe run

Runtime platform arch=amd64 os=windows pid=16436 revision=fe451d5a version=17.1.0

Starting multi-runner from c:\Gitlab-runner\config.toml... builds=0 max\_builds=0

Configuration loaded builds=0 max\_builds=1

listen\_address not defined, metrics & debug endpoints disabled builds=0 max\_builds=1

[session\_server].listen\_address not defined, session endpoints disabled builds=0 max\_builds=1

Initializing executor providers builds=0 max\_builds=1

WARNING: Failed to process runner builds=0 error=failed to update executor: initializing taskscaler: executor requires autoscaler config executor=instance max\_builds=1 runner=NfzDJZyBk

WARNING: Failed to process runner builds=0 error=failed to update executor: initializing taskscaler: executor requires autoscaler config executor=instance max\_builds=1 runner=NfzDJZyBk

Update : .gitlab-ci.yml file

# Define stages

stages:

- build

- deploy

# Variables

variables:

DOCKER\_DRIVER: overlay2

CI\_REGISTRY: registry.gitlab.com

CI\_REGISTRY\_IMAGE: registry.gitlab.com/gitlabtest2562326/git-runner-app

# Job to build and push the Docker image

build:

stage: build

image: docker:latest

services:

- docker:dind

variables:

DOCKER\_TLS\_CERTDIR: "/certs"

before\_script:

- docker info

script:

# Log in to the GitLab container registry

- docker login -u "pusamsaikumar302@gmail.com" -p "saikumarP123" $CI\_REGISTRY

- echo "Current Directory:saikumarP123"

- docker build -t registry.gitlab.com/gitlabtest2562326/git-runner-app:gitrunnerapp -f GitRunnerApp/Dockerfile .

# Push the Docker image to the GitLab container registry

- docker push registry.gitlab.com/gitlabtest2562326/git-runner-app:gitrunnerapp

only:

- main

# Job to deploy the Docker image (example deployment)

deploy:

stage: deploy

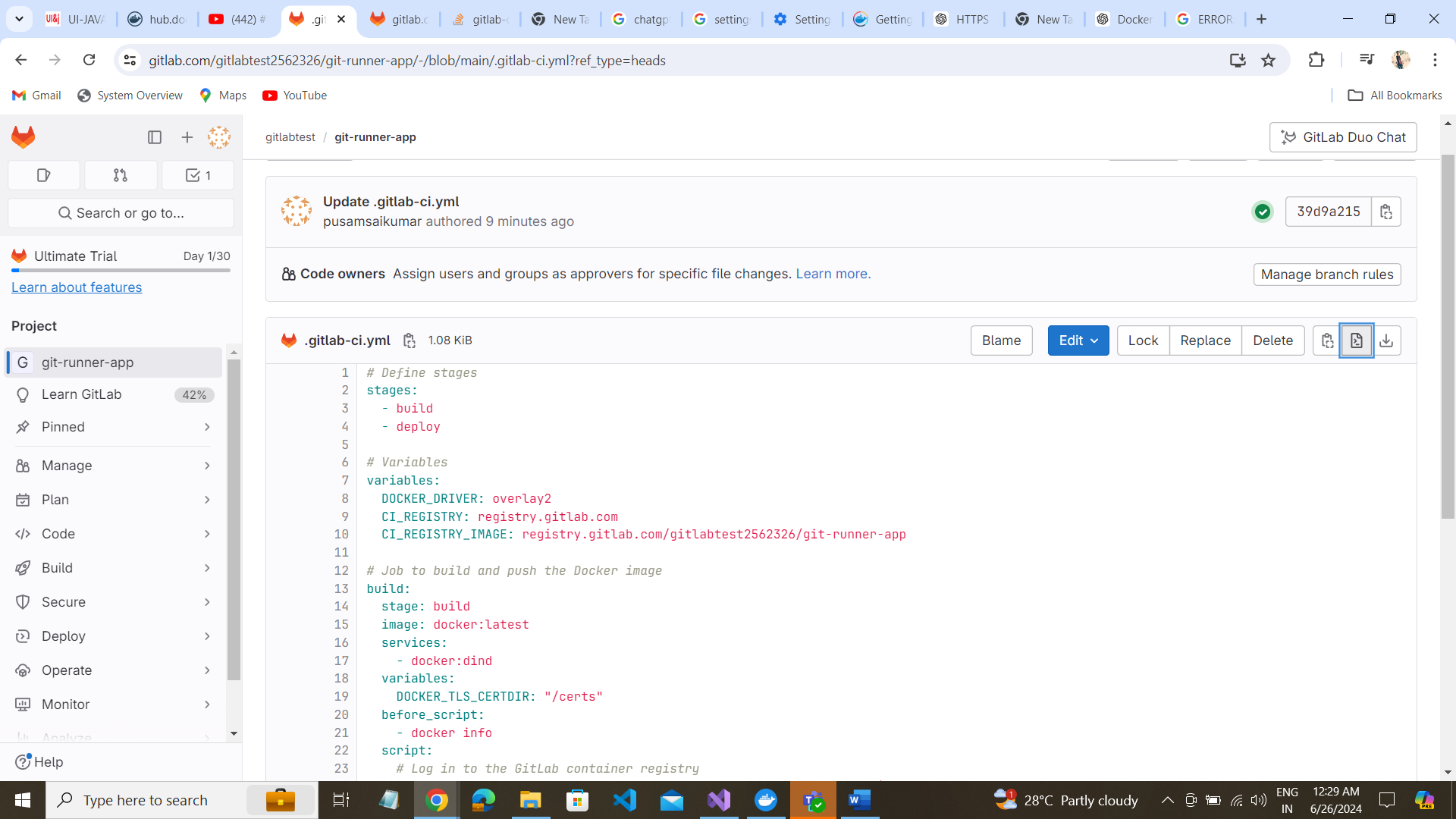
image: alpine:latest

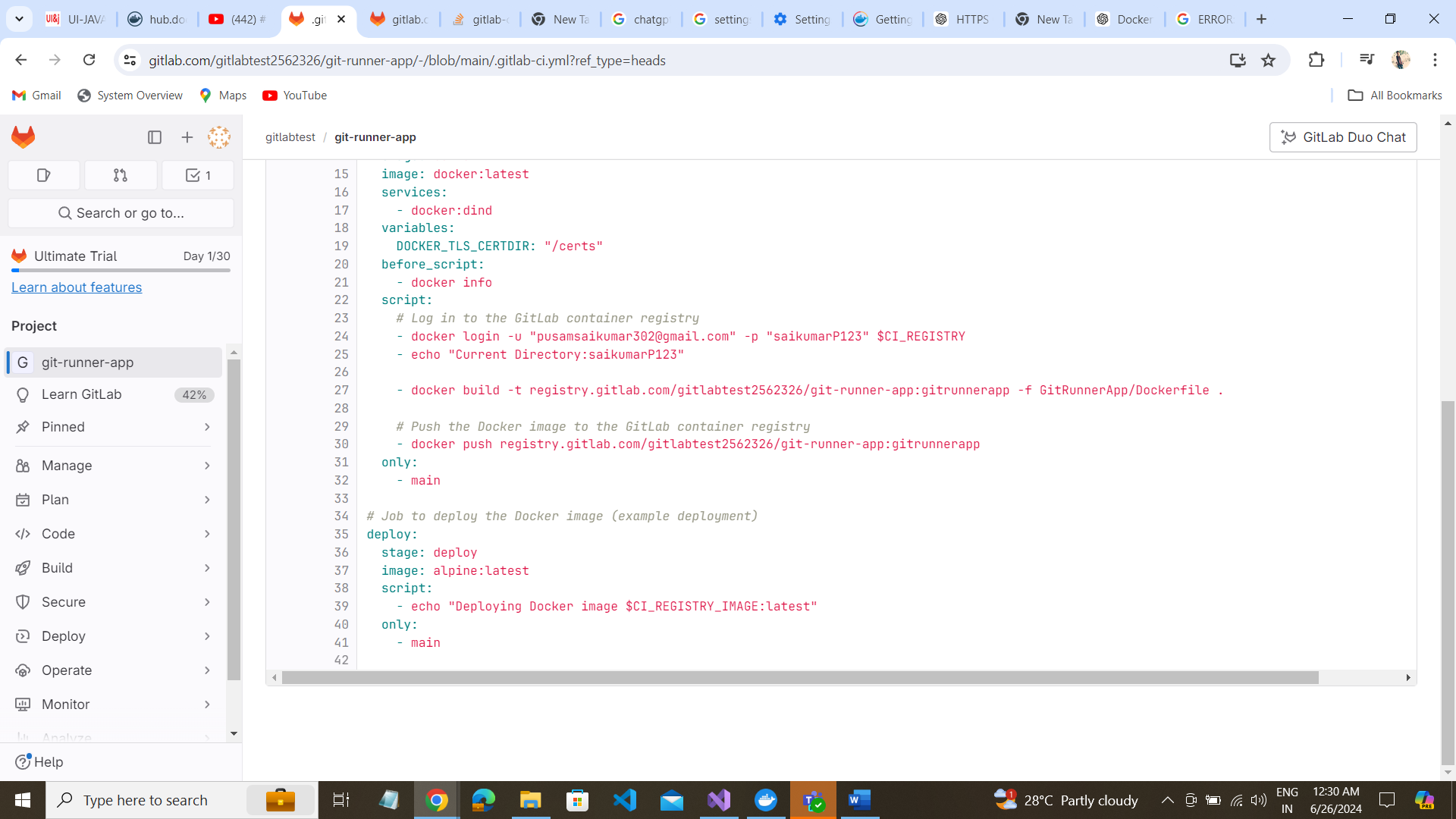
script:

- echo "Deploying Docker image $CI\_REGISTRY\_IMAGE:latest"

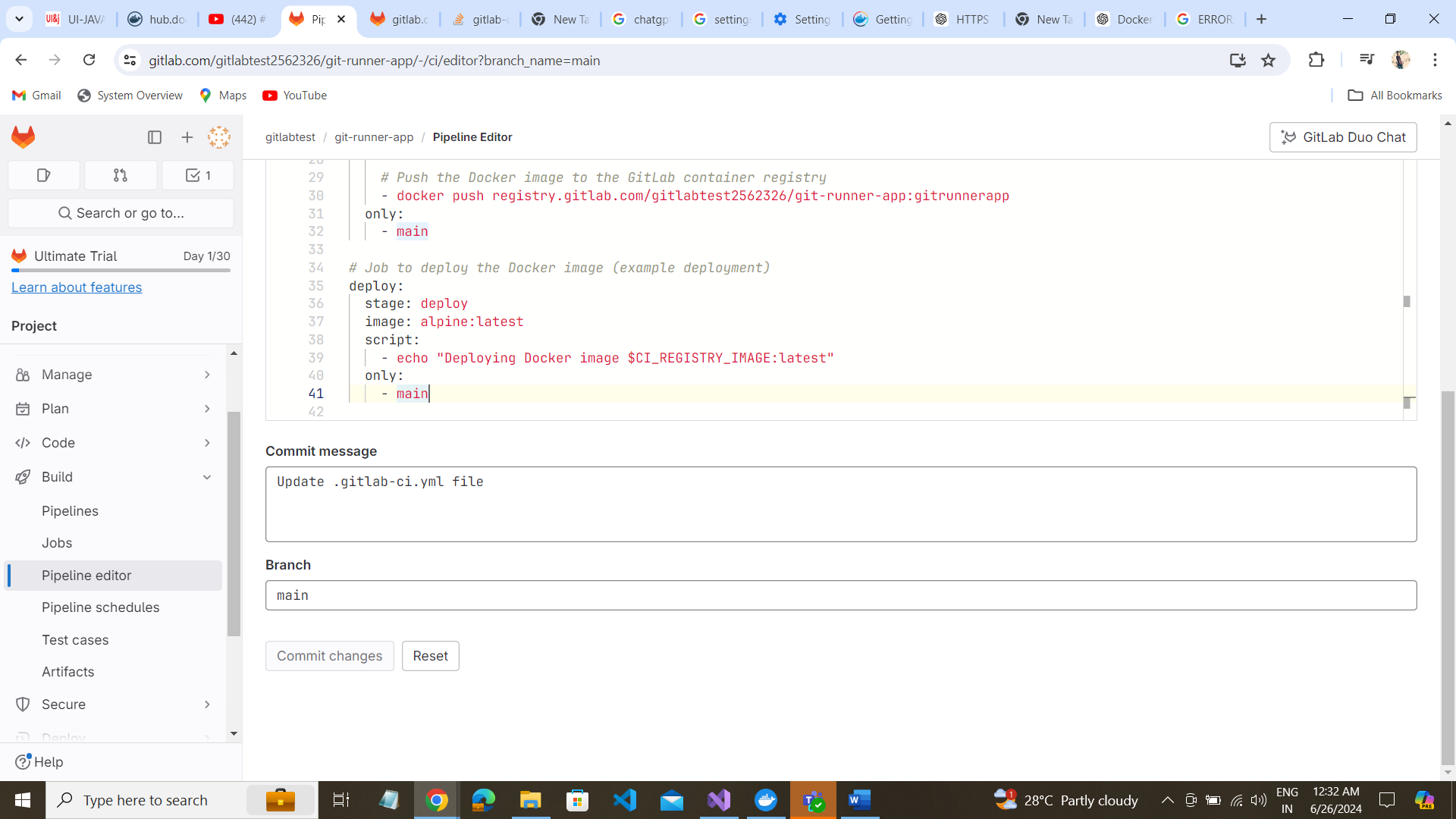
only:

- main

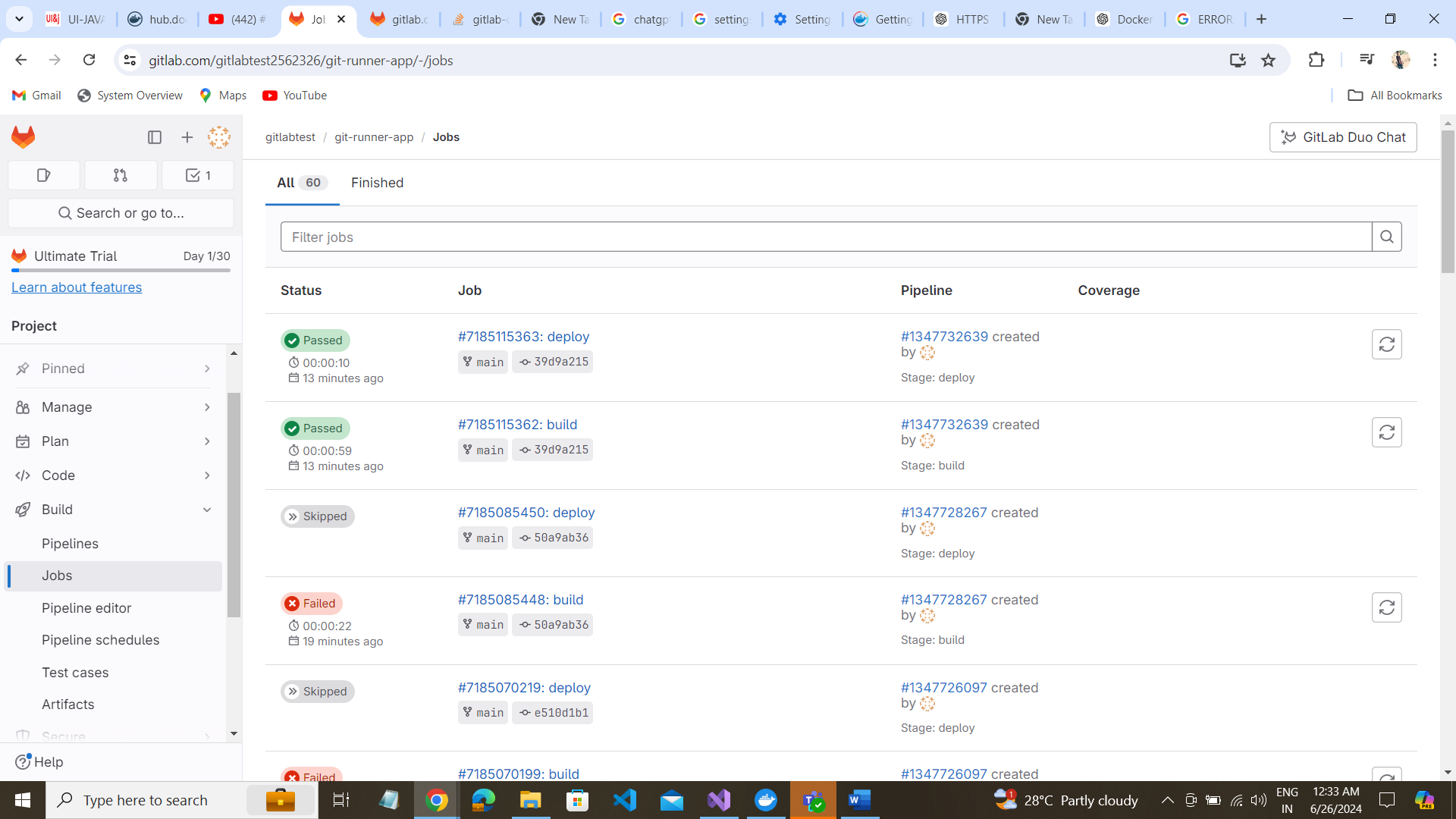




Commit changes



Check jobs: build and deploy



Example: stages like build test deploy build-docker:

# Define stages

stages:

- build

- test

- deploy

# Variables

variables:

DOCKER\_DRIVER: overlay2

CI\_REGISTRY: registry.gitlab.com

CI\_REGISTRY\_IMAGE: registry.gitlab.com/gitlabtest2562326/git-runner-app

# Job to build the application

build:

stage: build

image: mcr.microsoft.com/dotnet/sdk:6.0

script:

# Restore dependencies

- dotnet restore ./GitRunnerApp.sln

# Build the solution

- dotnet build ./GitRunnerApp.sln -c Release

artifacts:

paths:

- GitRunnerApp/bin

only:

- main

# Job to run tests

test:

stage: test

image: mcr.microsoft.com/dotnet/sdk:6.0

script:

# Run tests

- dotnet test ./GitRunnerApp.sln

only:

- main

# Job to build and push the Docker image

docker\_build:

stage: deploy

image: docker:latest

services:

- docker:dind

variables:

DOCKER\_TLS\_CERTDIR: "/certs"

before\_script:

- docker info

script:

# Log in to the GitLab container registry

- echo "$CI\_JOB\_TOKEN" | docker login -u "$CI\_REGISTRY\_USER" --password-stdin $CI\_REGISTRY

# Build the Docker image

- docker build -t $CI\_REGISTRY\_IMAGE:gitrunnerapp -f GitRunnerApp/Dockerfile .

# Push the Docker image to the GitLab container registry

- docker push $CI\_REGISTRY\_IMAGE:gitrunnerapp

only:

- main

# Job to deploy the Docker image (example deployment)

deploy:

stage: deploy

image: alpine:latest

script:

- echo "Deploying Docker image $CI\_REGISTRY\_IMAGE:gitrunnerap"

only:

- main