**Practical 1**

**Aim: Git and Github, Basic Git Commands**

**In git bash :**

ADMIN@DESKTOP-MGB9JRR MINGW64 ~

$ git config --global user.name "Aradhya Pendurkar"

ADMIN@DESKTOP-MGB9JRR MINGW64 ~

$ git config --global user.email "2403z0nf8waradhya@viva-technology.org"

ADMIN@DESKTOP-MGB9JRR MINGW64 ~

$ cd coding

bash: cd: coding: No such file or directory

ADMIN@DESKTOP-MGB9JRR MINGW64 ~

$ mkdir git-lab-1

ADMIN@DESKTOP-MGB9JRR MINGW64 ~

$ cd git-lab-1

ADMIN@DESKTOP-MGB9JRR MINGW64 ~/git-lab-1

$ code

**In vscode terminal :**

**Download extension “Git graph”**

**Open git folder in vscode**

**Create index.js file**

PS C:\Program Files\Git> git init

Initialized empty Git repository in C:/Program Files/Git/.git/

**Now we can see hidden .git folder in file explorer(show hidden files from view)**

**We can see untracked besides the index.js file**

**index .js :**

**const age = 20;**

**const name = "Aradhya Pendurkar ";**

**const dob = "19/03/2003";**

PS C:\Program Files\Git> git add index2.js

**Now its added**

**Add new lines to index.js:**

**const age2 = 20;**

**const name2 = "Aradhya";**

**const dob2 = "19/03/2003";**

PS C:\Program Files\Git> git diff

diff --git a/index.js b/index.js

index 77a6eac..fb1d22b 100644

--- a/index.js

+++ b/index.js

@@ -1,3 +1,9 @@

const age = 20;

-const name = "Aradhya Pendurkar";

-const dob = "19/03/2003";

\ No newline at end of file

+const name = "Aradhya Pendurkar ";

+const dob = "19/03/2003";

+

+

+

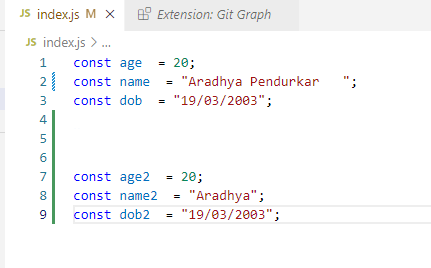
+const age2 = 20;

+const name2 = "Aradhya";

+const dob2 = "19/03/2003";

\ No newline at end of file

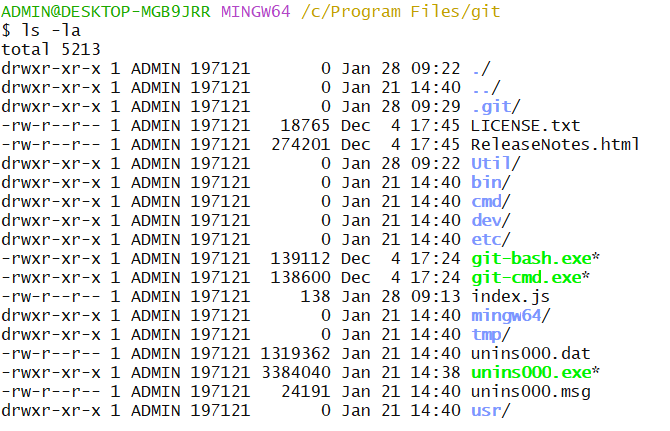
:



****

**git add .**

(adds all in one command )

**** 

**1. git config --global user.name "Aradhya Pendurkar"**

Set your name for commits.



**2. git config --global user.email "your\_email@example.com"**

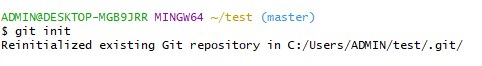
Set your email for commits.



**3. git init**

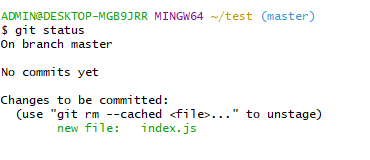
Initialize a new repository in your project folder.

click view→ hidden items enabled $



**4. git status**

Check the status of files in the working directory (untracked, modified, etc.).



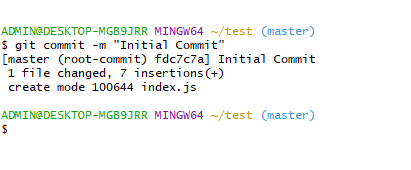
**5. git add .**

Stage all changes (new, modified, or deleted files).

Any folder can be added in staging area. If we put add.\_\_\_ it will add all the files to staging area. Repeat step 4.



**6. git commit -m "Initial commit"**



**Practical 2**

**Aim:**

**3.Push and Pull request**

**4. Git fork**

**5. Git merge**

**6. Git Cloning**

**7. Git Collaborations**

Github.com

You required your email id

First time then signup

If already created account then sign in

Now at the right top side of github icon is there is Your Profile menu explain and

To create Repository on GitHub

New :

create repository>>

Give repository name as githubBatch1(do not give space while creating repository)

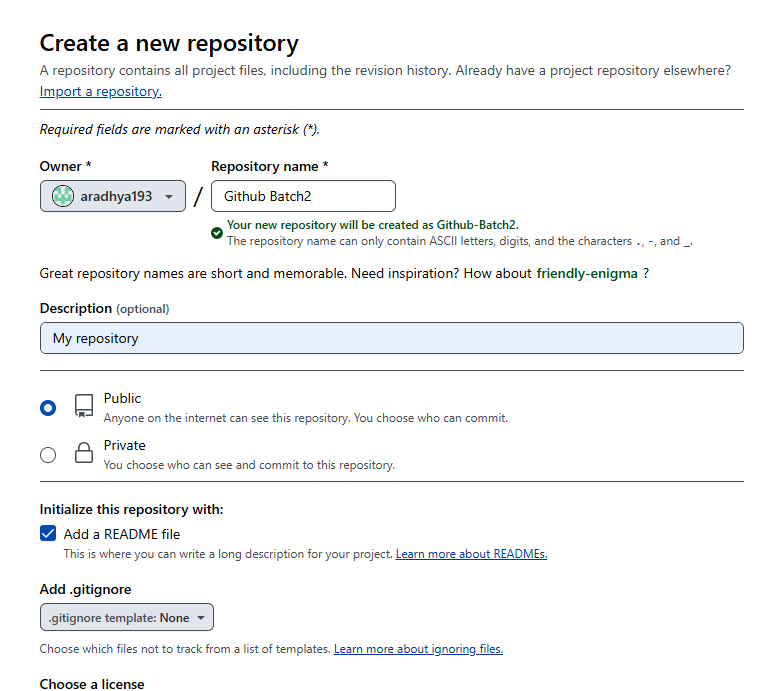
you can write any description

take private

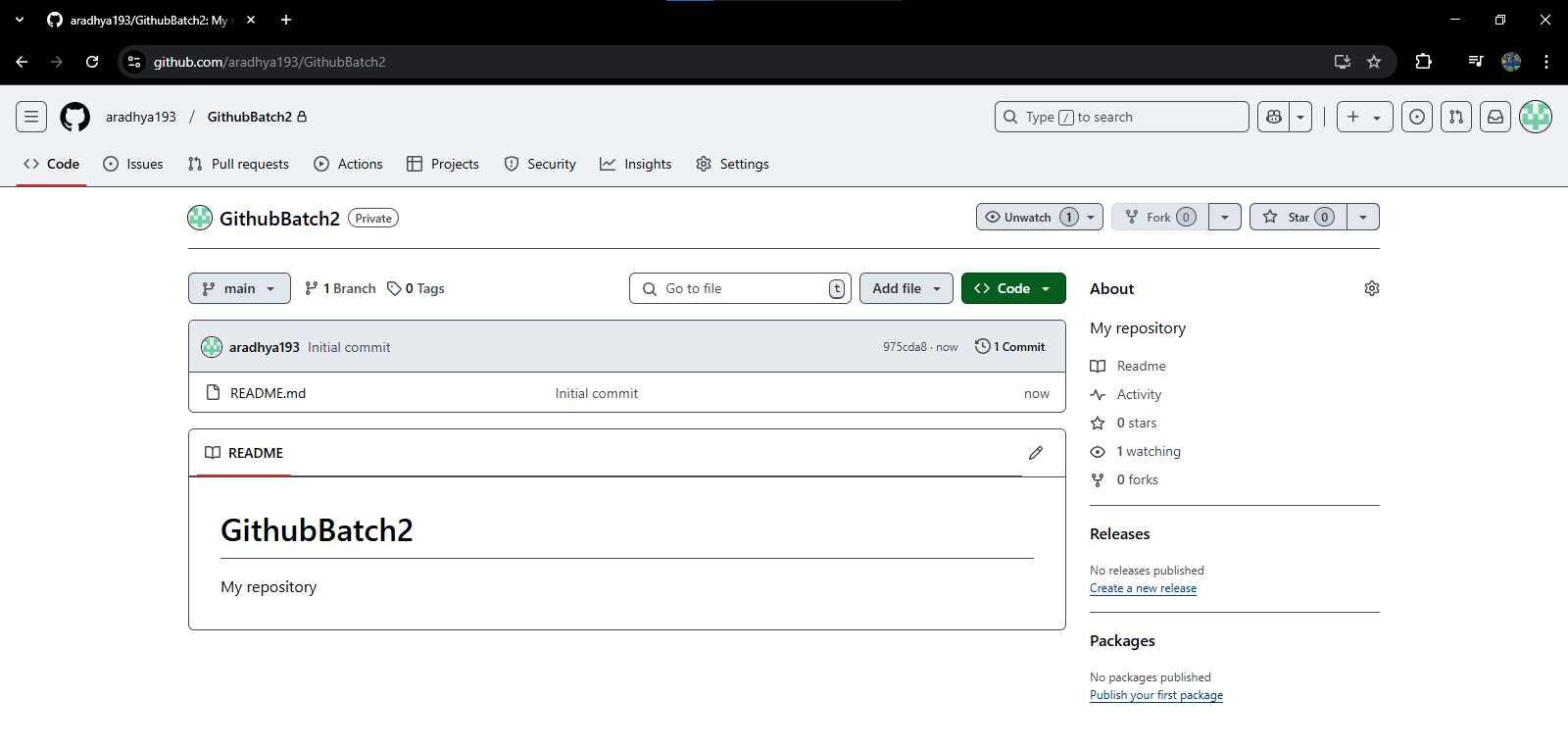
Now Add a README file (Compulsory add otherwise it will not add because no any file

there)

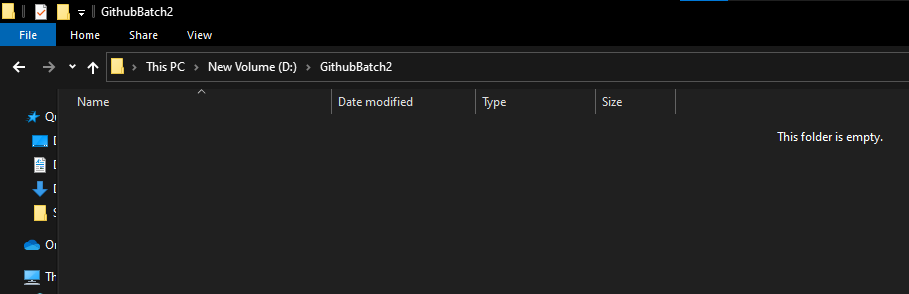
by default main branches directly created

****

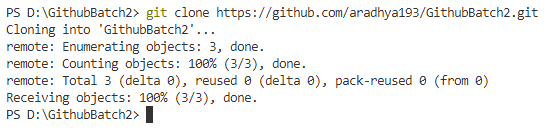
**Repository Created**

****

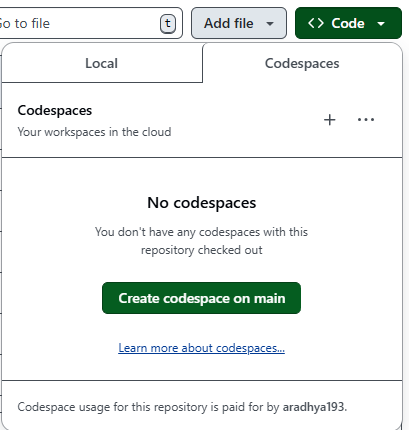
**Create a Folder named after the repository name**

****

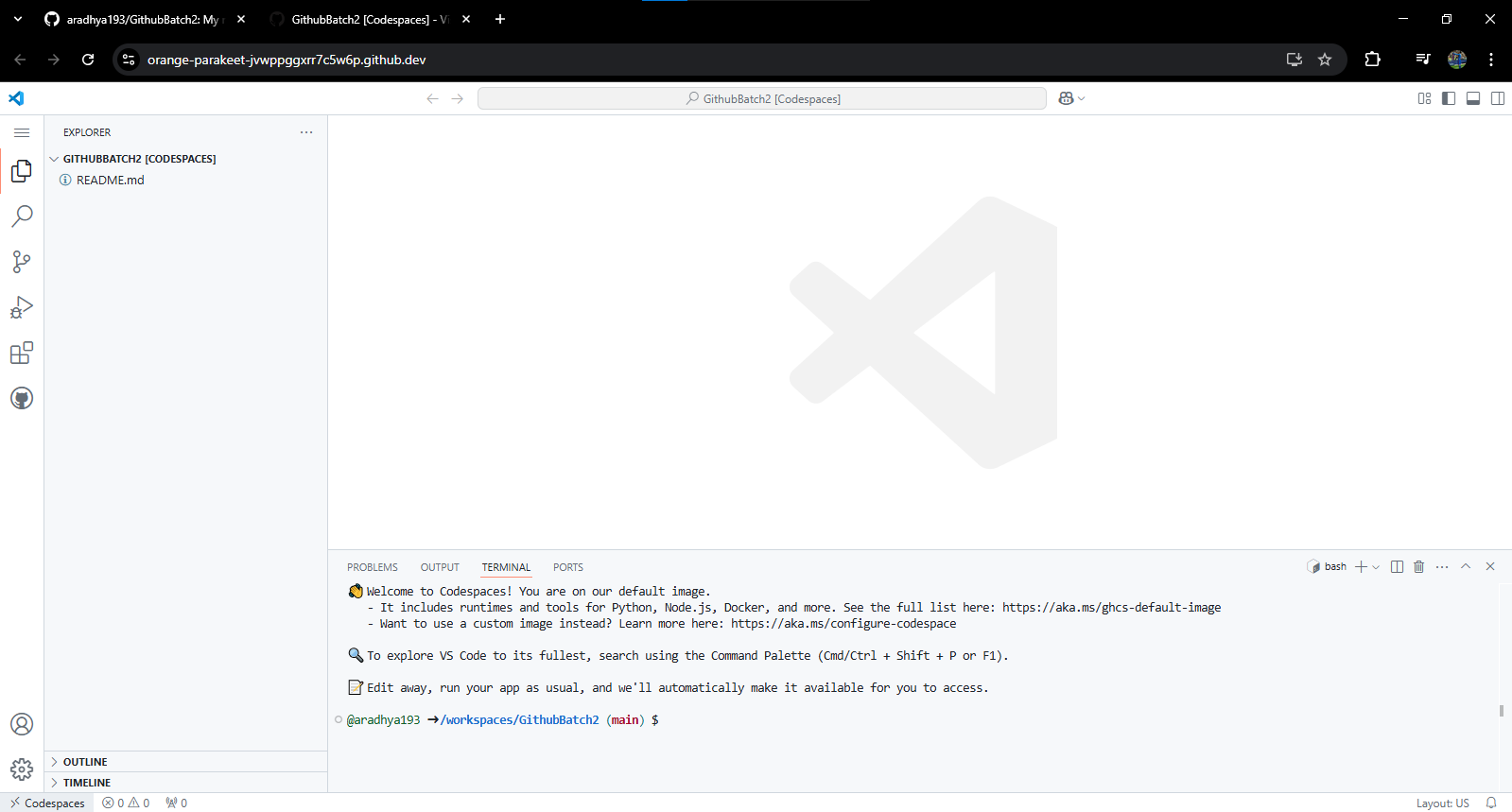
**Go to folder and right click → Open with VSCode and type in terminal→ git clone and your repo link**

****

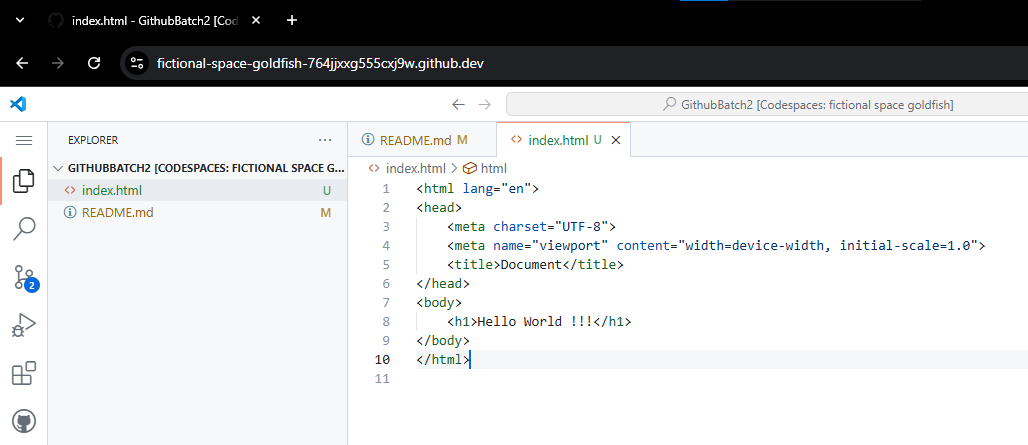
**Now Click On code and create Codespace on main**

****

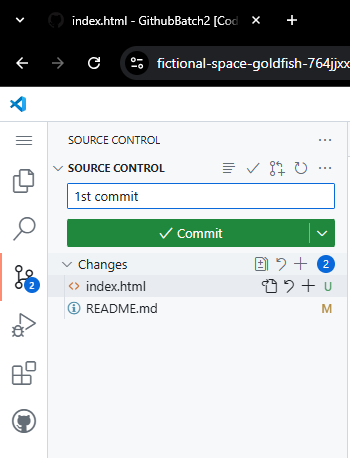
**Codespace created**

****

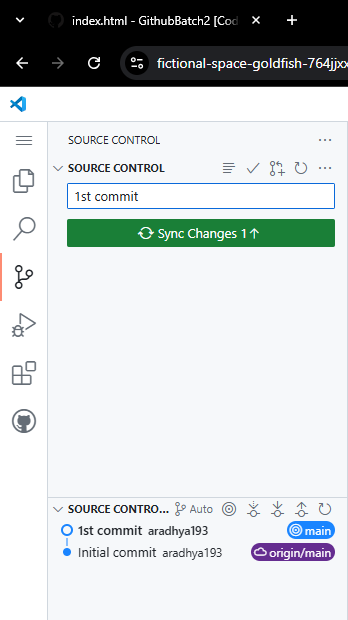
**Create a new file index.html and write doc and press enter.**

****

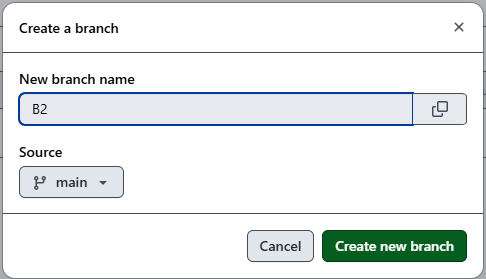
**Click on branch and write message and click on commit**

****

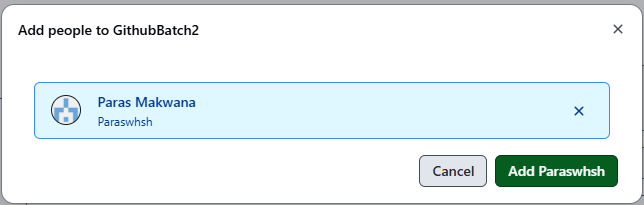
**Now write message and sync changes**

****

**Creating a branch**

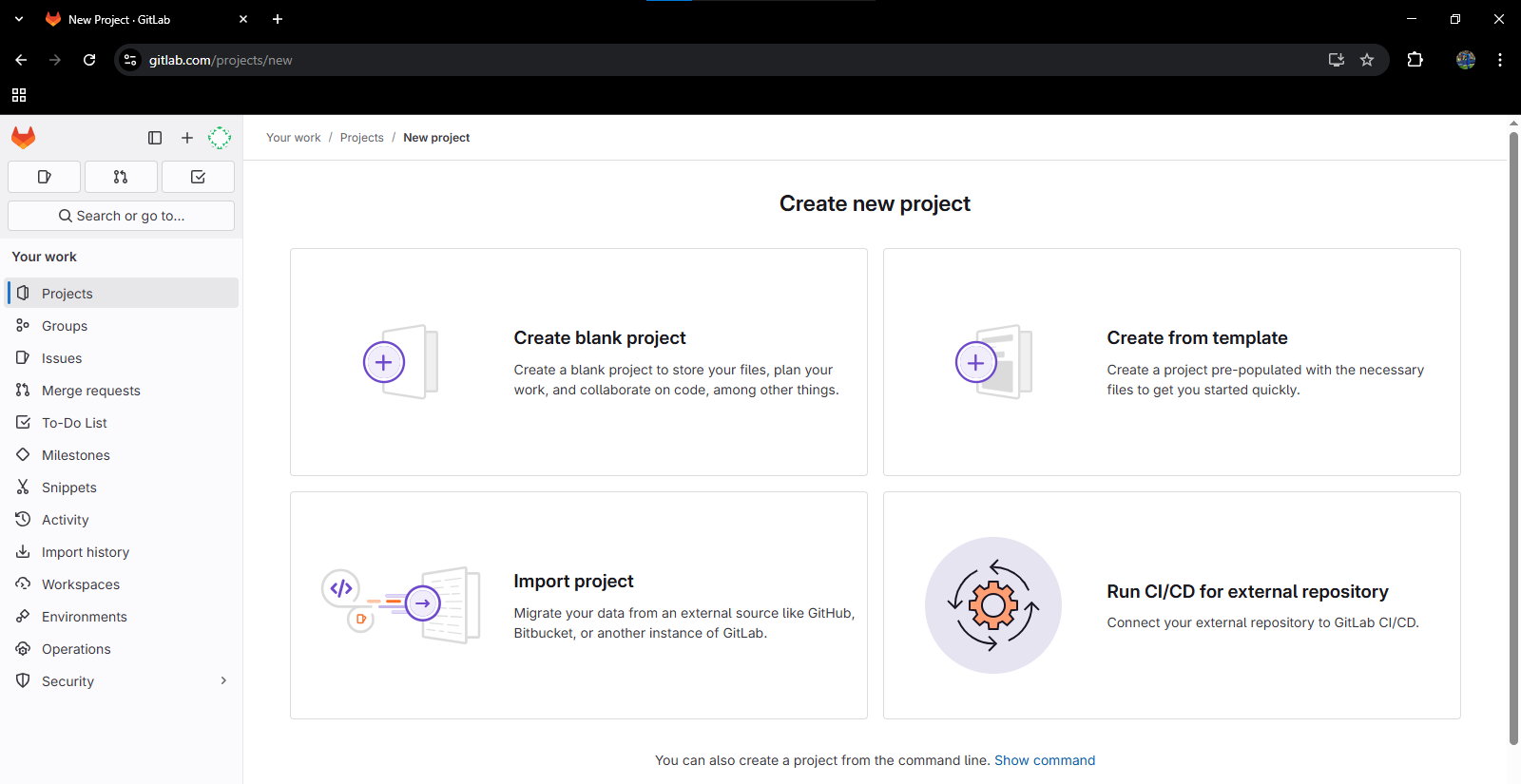
****

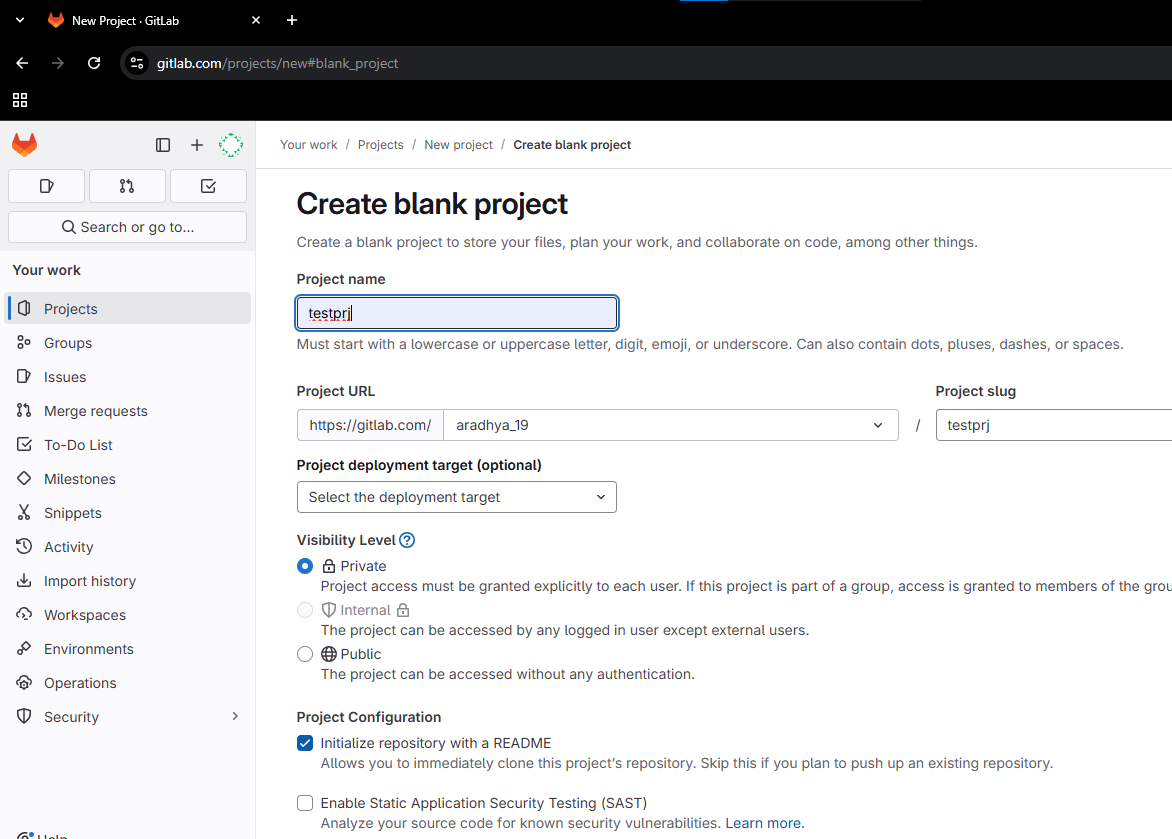
**Adding people**

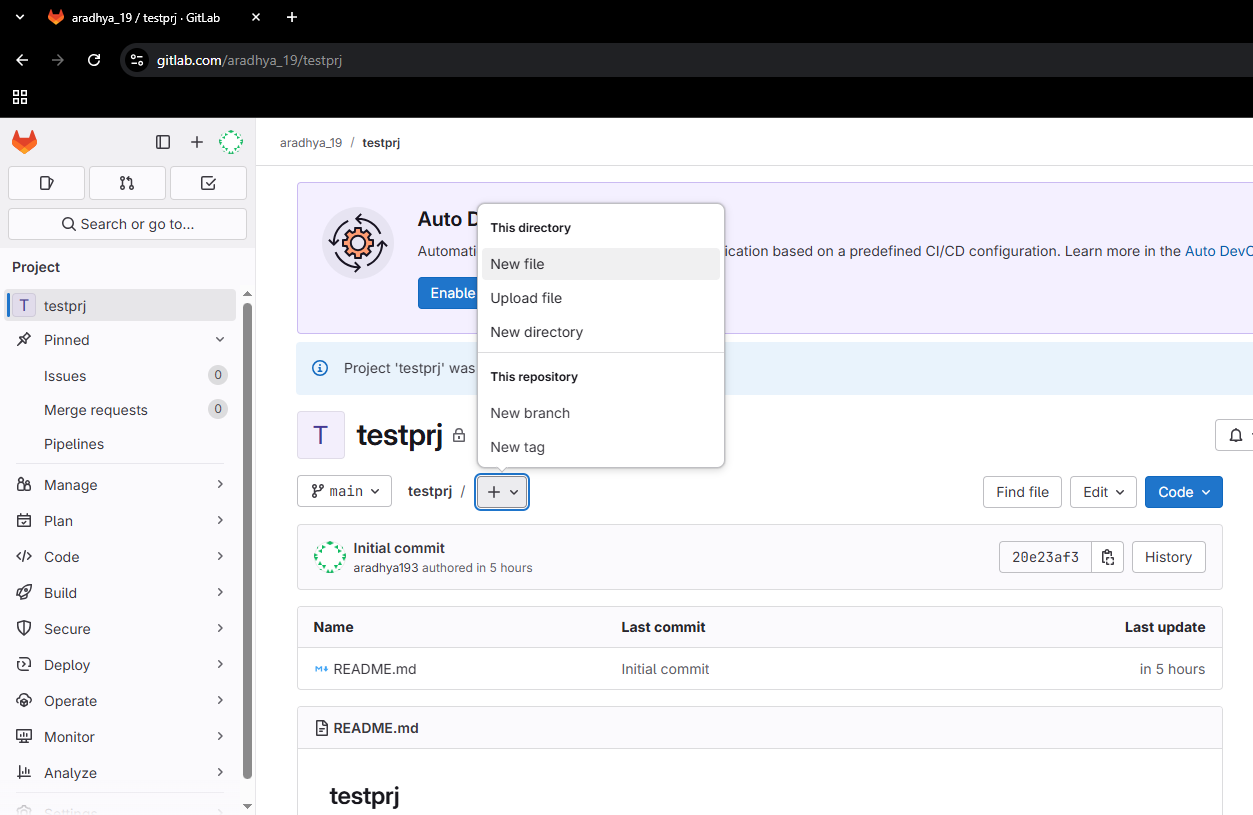
****

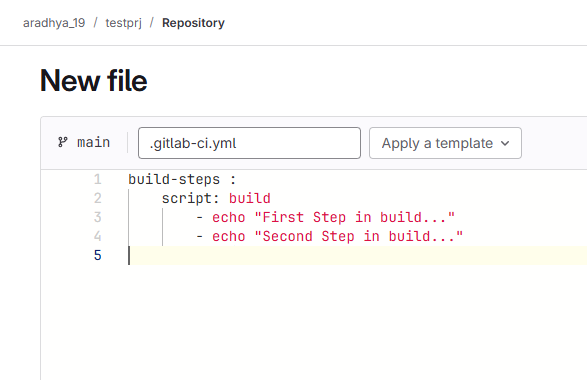
**Practical 3**

**Create a blank project :**

****

****

****

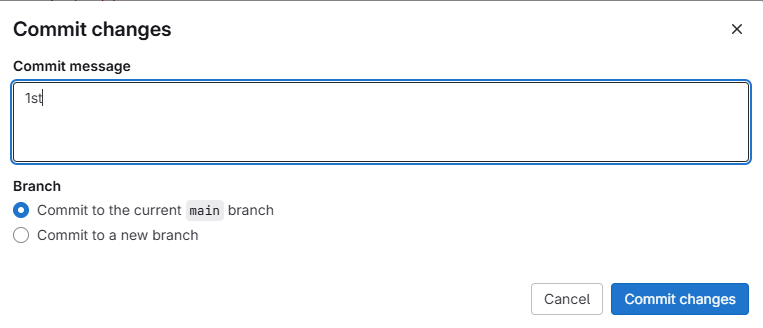
****

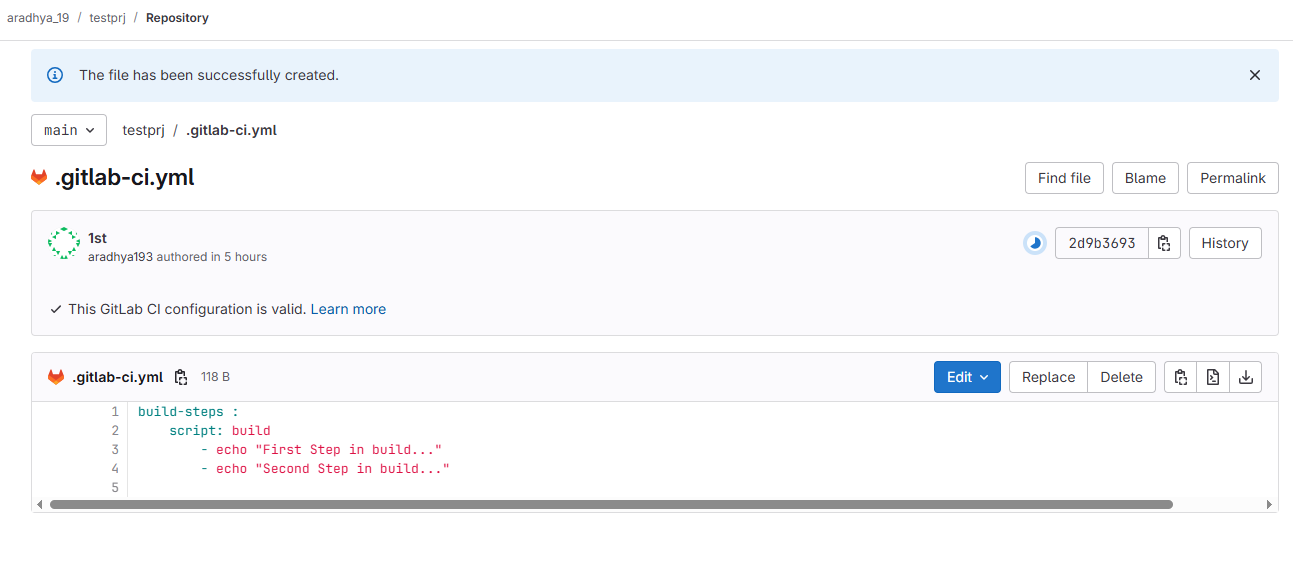
**build-steps :**

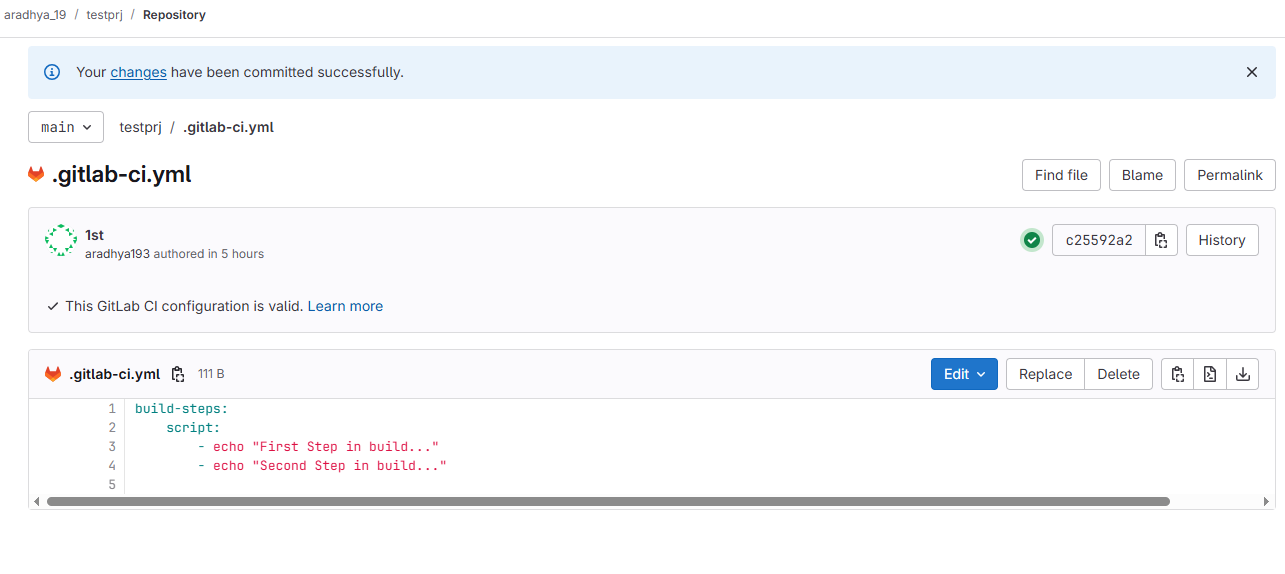
**script:**

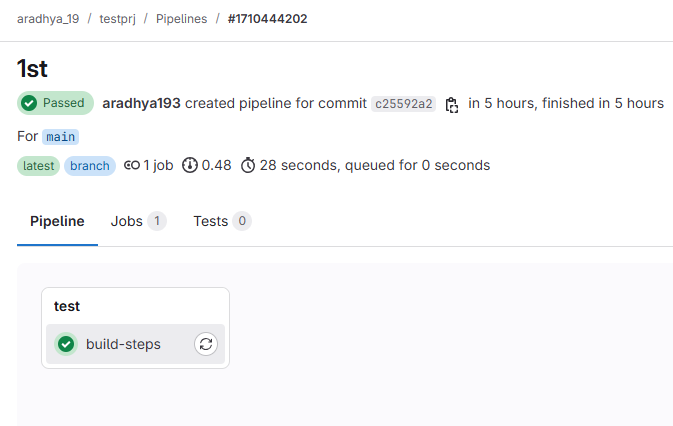
**- echo "First Step in build..."**

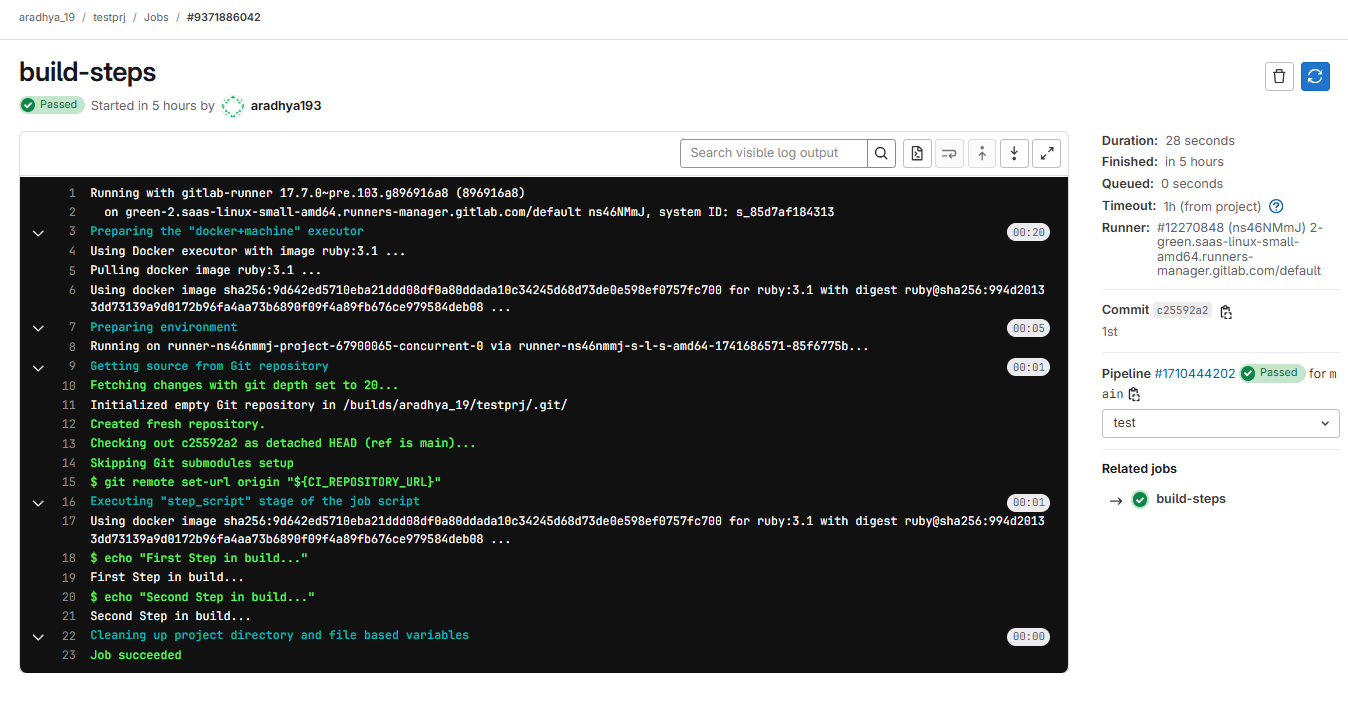
**- echo "Second Step in build..."**

****

****

****

****

****

**stages:**

**- build**

**- test**

**- deploy**

**build-steps :**

**stage: build**

**script:**

**- echo "First Step in build..."**

**- echo "Second Step in build..."**

**testing-job :**

**stage: test**

**script:**

**- echo "First Step in testing..."**

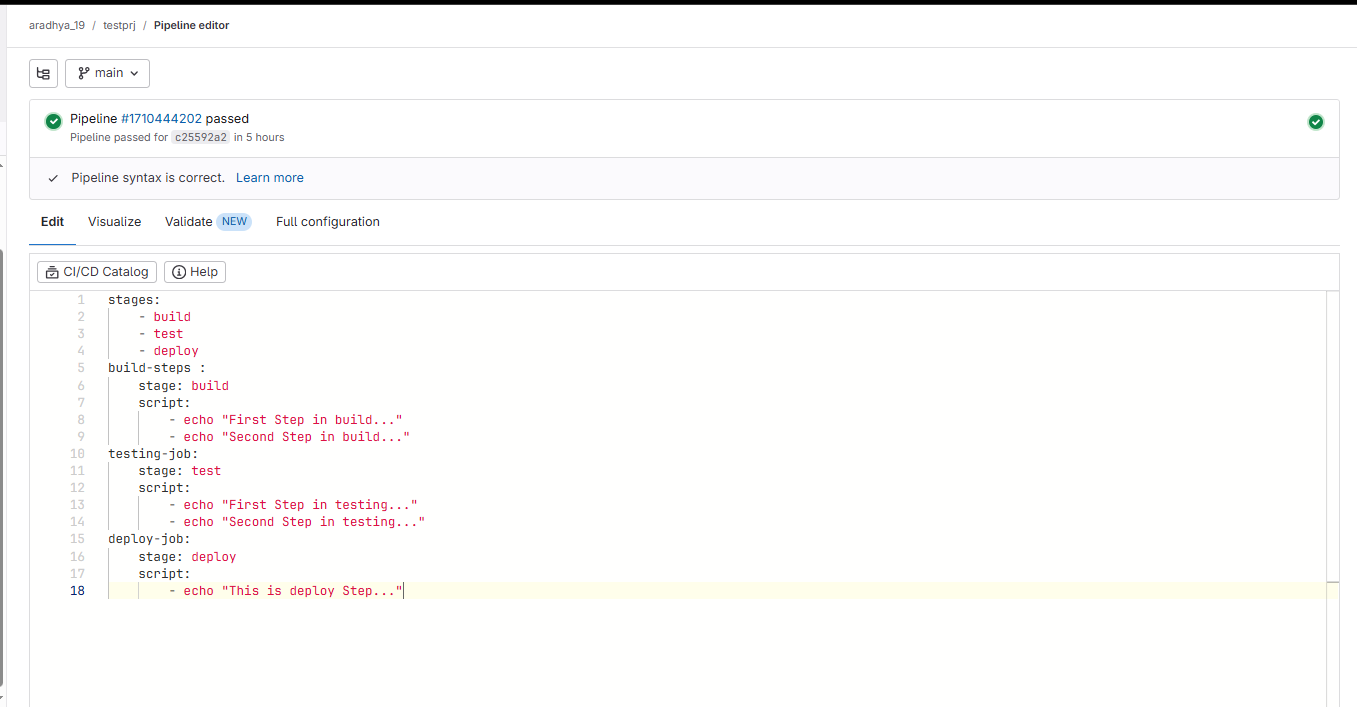
**- echo "Second Step in testing..."**

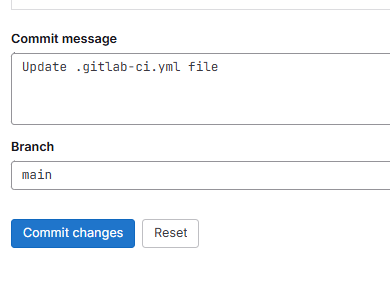
**deploy-job :**

**stage: deploy**

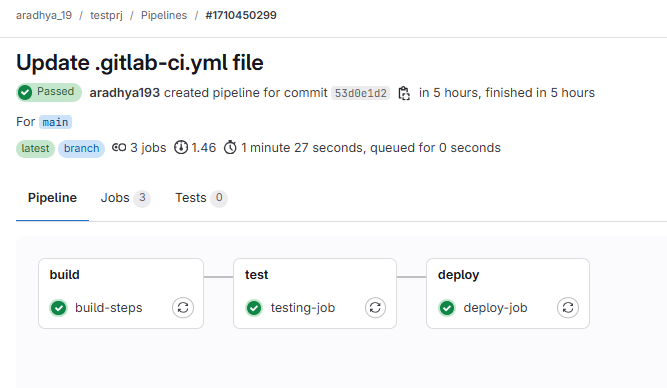
**script:**

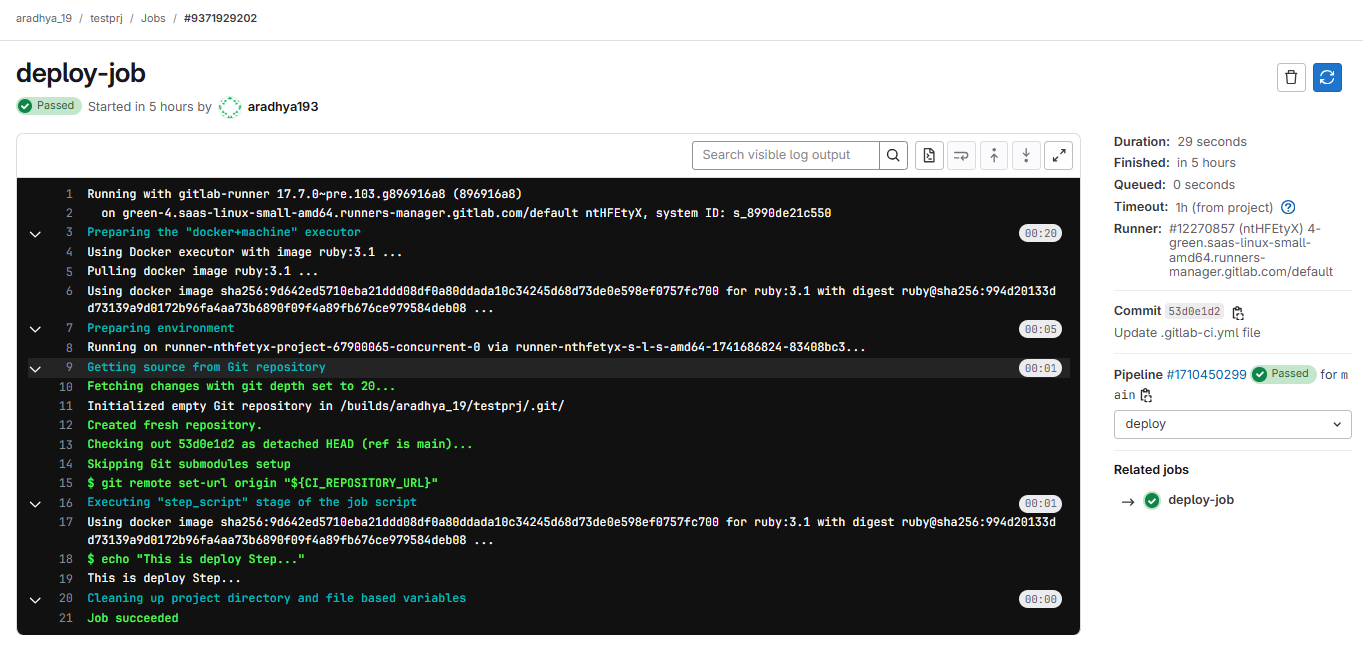
**- echo "This is deploy step..."**

****

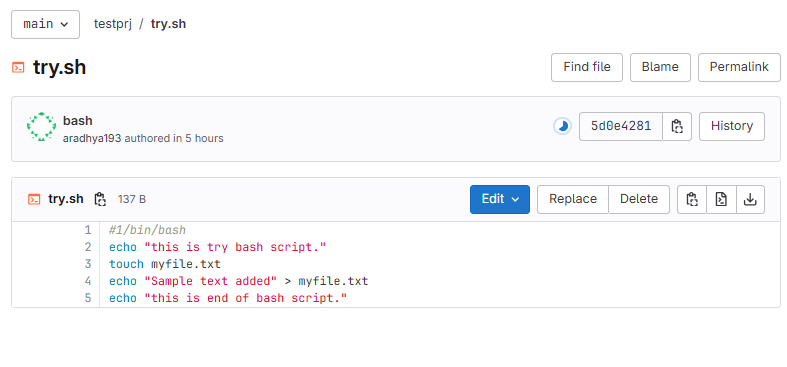
****

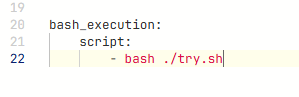
****

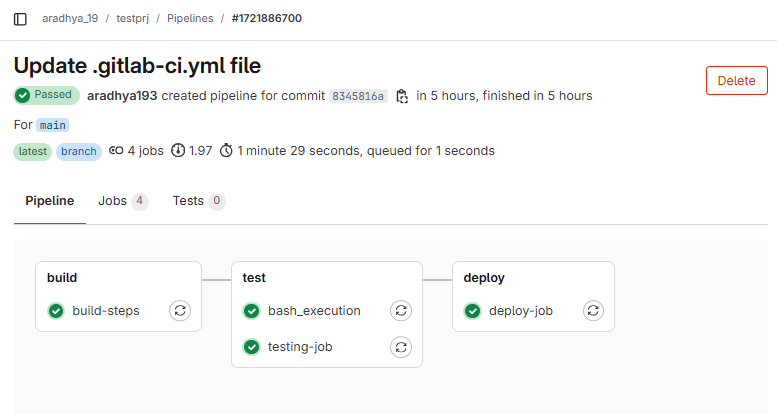
****

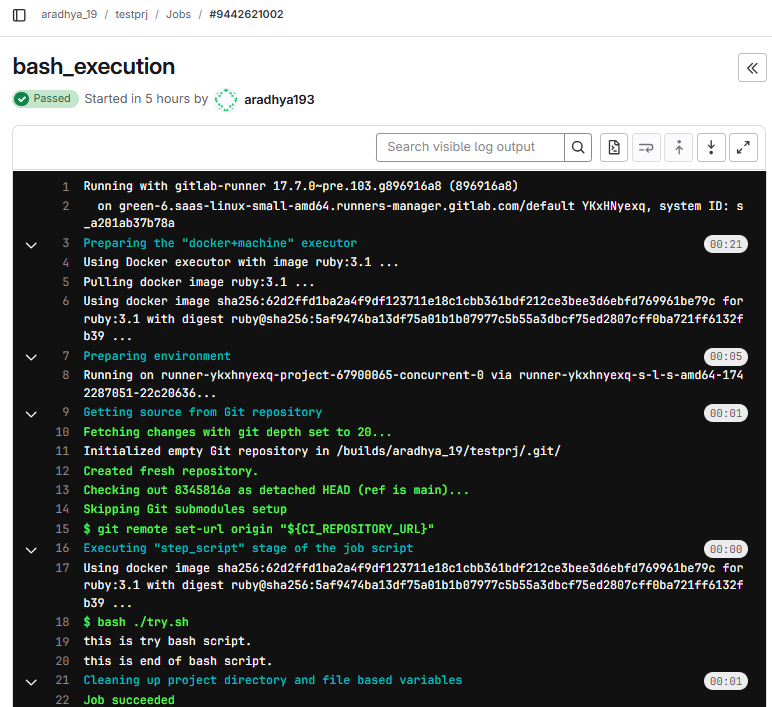


Running a bash script:

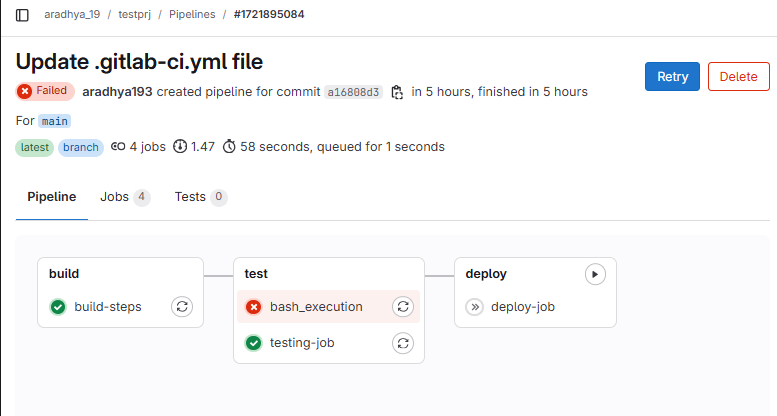


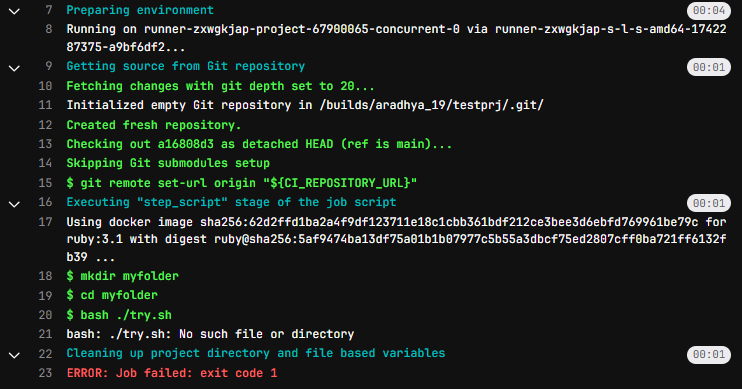




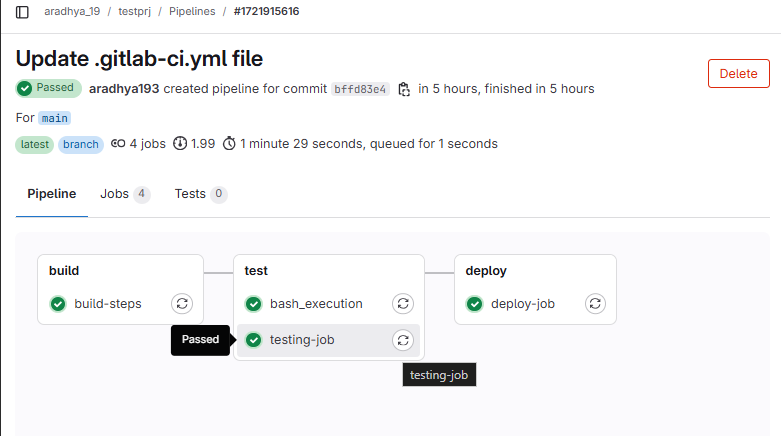












stages:

- build

- test

- deploy

- bash\_execution

build-steps :

stage: build

script:

- echo "First Step in build..."

- echo "Second Step in build..."

testing-job :

stage: test

script:

- echo "First Step in testing..."

- echo "Second Step in testing..."

deploy-job :

stage: deploy

script:

- echo "This is deploy step..."

bash\_execution:

script:

- mkdir myfolder

- cp ./try.sh myfolder

- cd myfolder

- bash ./try.sh

- touch newtext.txt

artifacts:

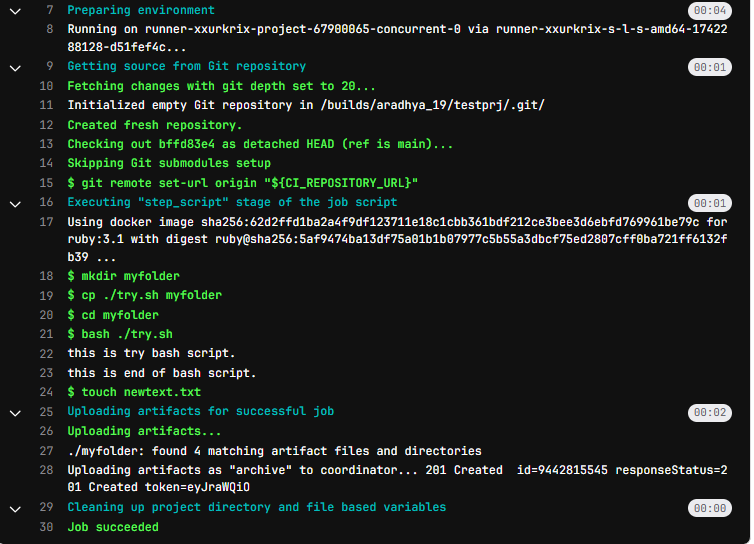
paths:

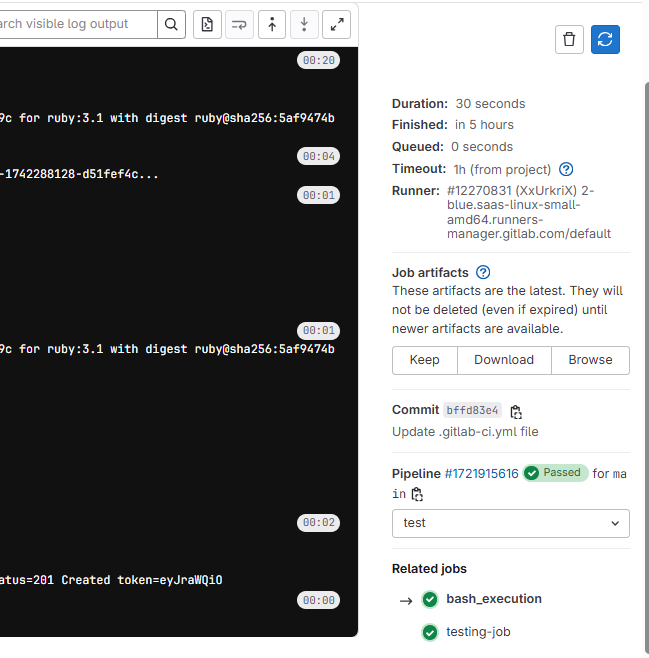
- "./myfolder"

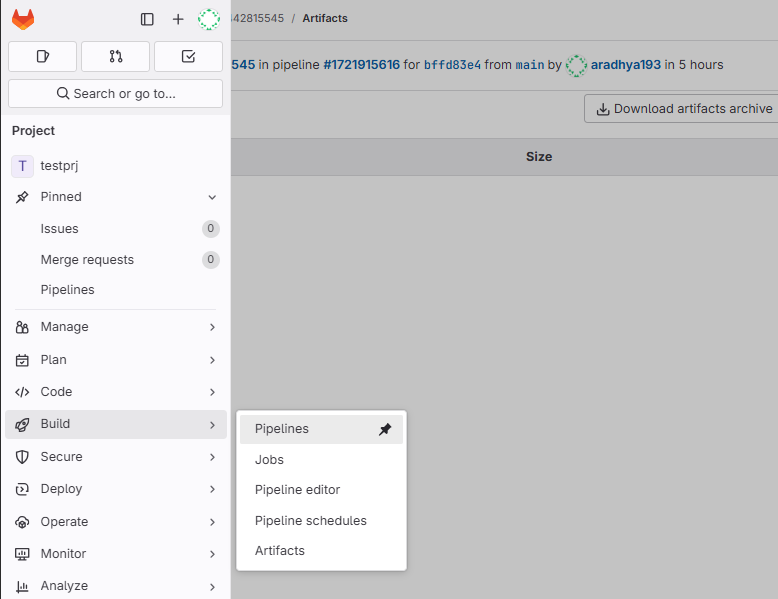
when: on\_success

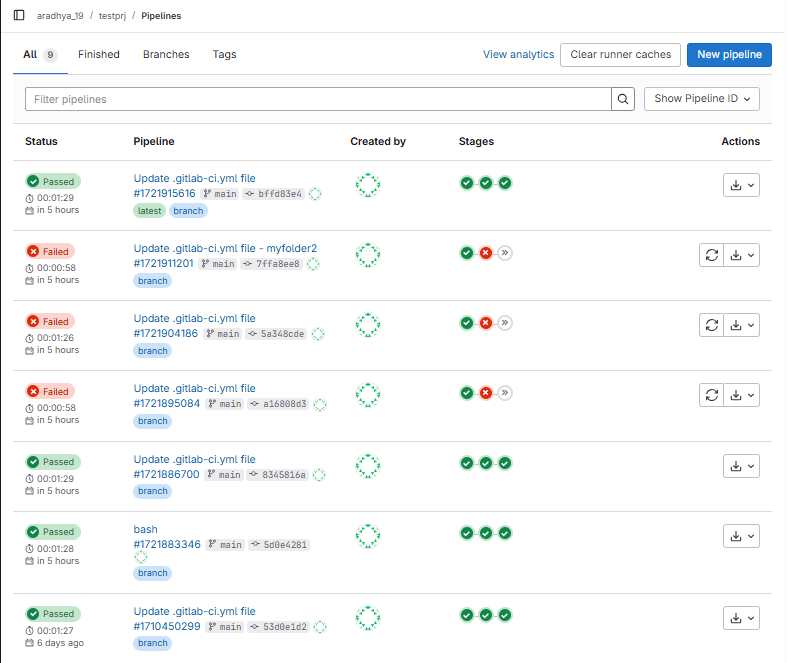
access: all

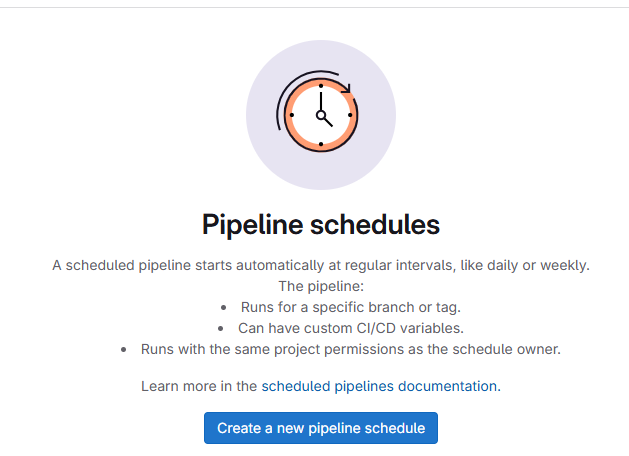
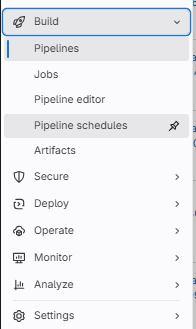
expire\_in: 30 days

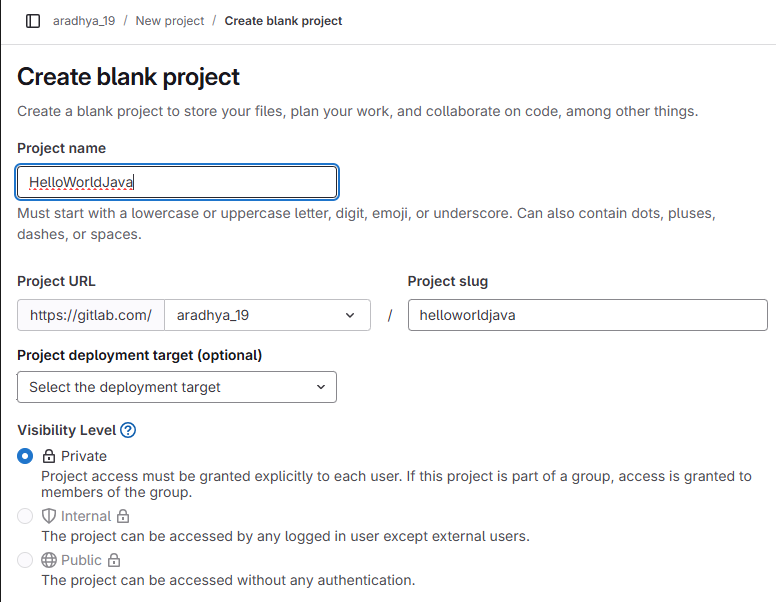


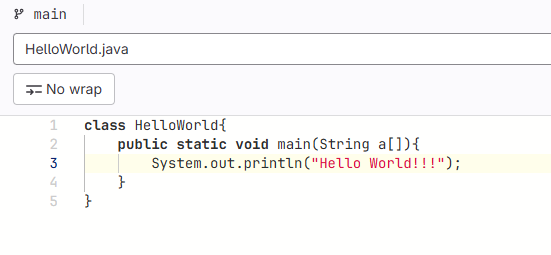












stages:

- build

- test

before\_script:

- apt-get update && apt-get install -y openjdk-17-jdk

build:

stage: build

script:

- javac HelloWorld.java

- ls -l

artifacts:

paths:

- "HelloWorld.class"

only:

- main

test :

stage: test

when: manual

script:

- ls -l

- java HelloWorld

only:

- main

