**Name: Parmita Patre**

**Class: Final Year CSE Div: B**

**Roll: B\_58**

**Assignment 4**

**Problem Statement:** Set up a GitLab CI/CD Pipeline.

GitLab CI/CD is a powerful tool for automating the software development lifecycle. By utilising a configuration file named .gitlab-ci.yml, developers can define a series of stages and jobs that will be executed in a specific order. This allows for continuous integration (CI) and continuous deployment (CD), which helps streamline the development process and maintain code quality.

**.gitlab-ci.yml file:**

stages:

- build

- test

- deploy

Build job:

stage: build

script:

- echo "The job has started to build"

Test job:

stage: test

script:

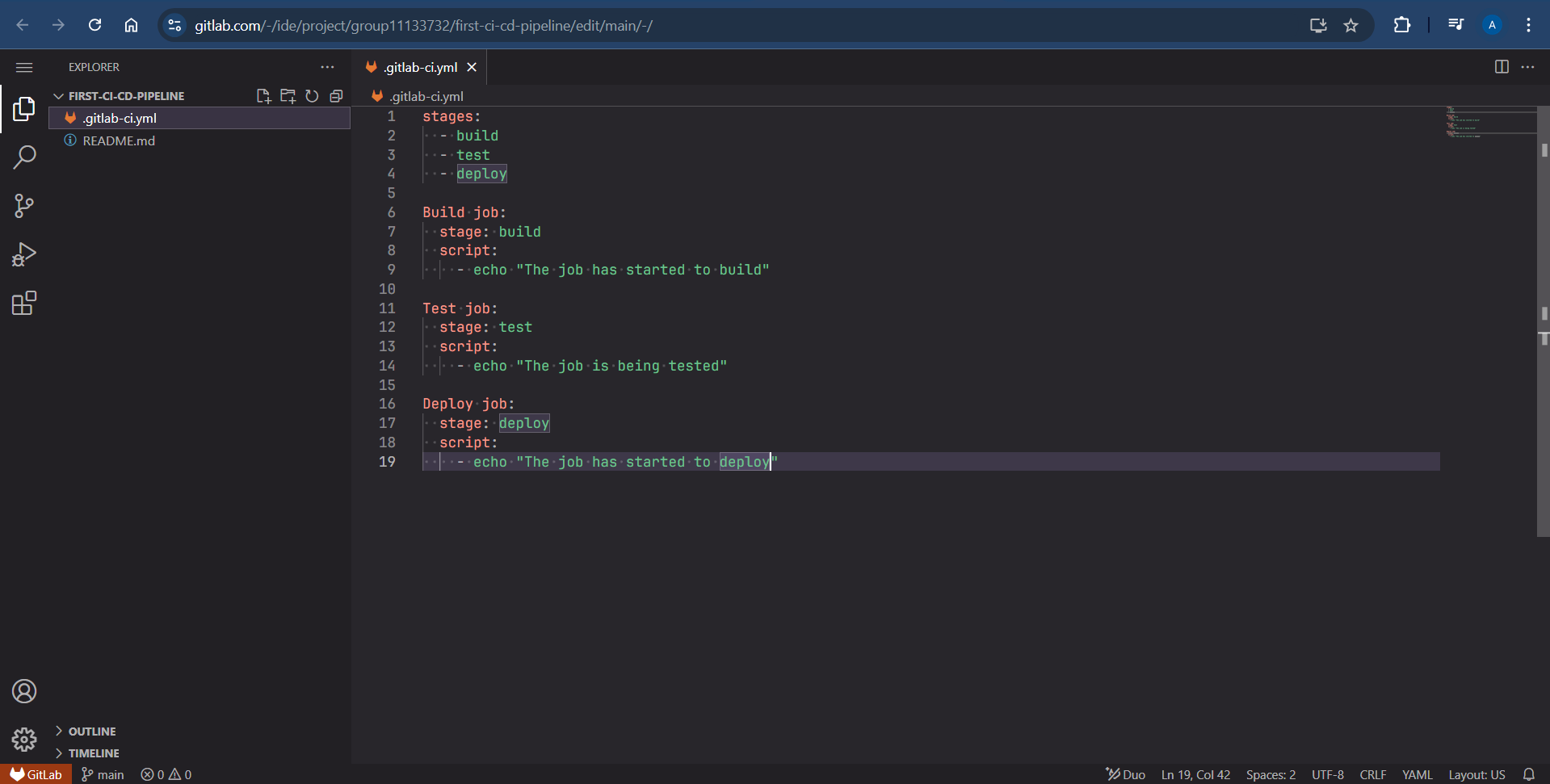
- echo "The job is being tested"

Deploy job:

stage: deploy

script:

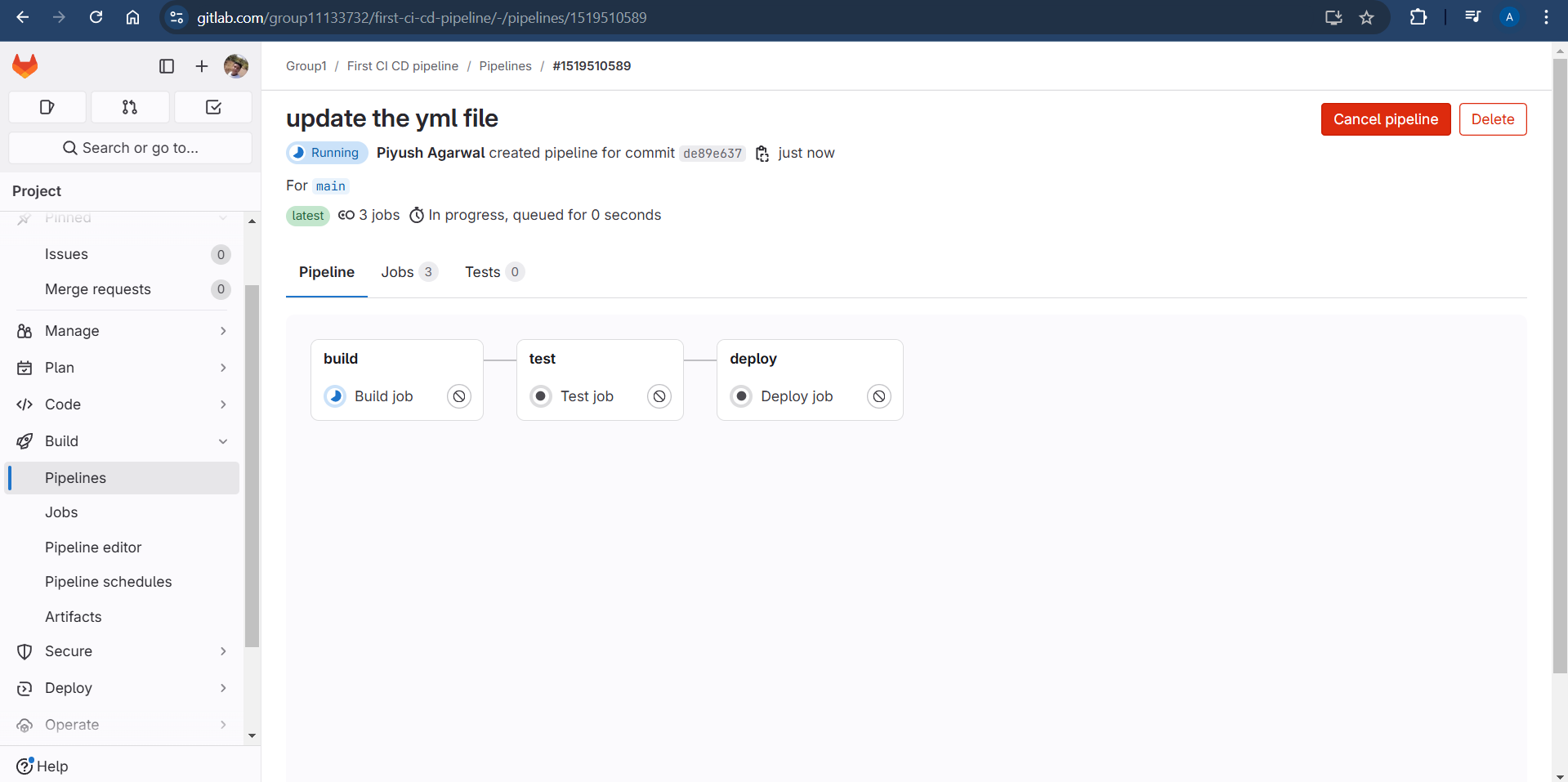
- echo "The job has started to deploy"

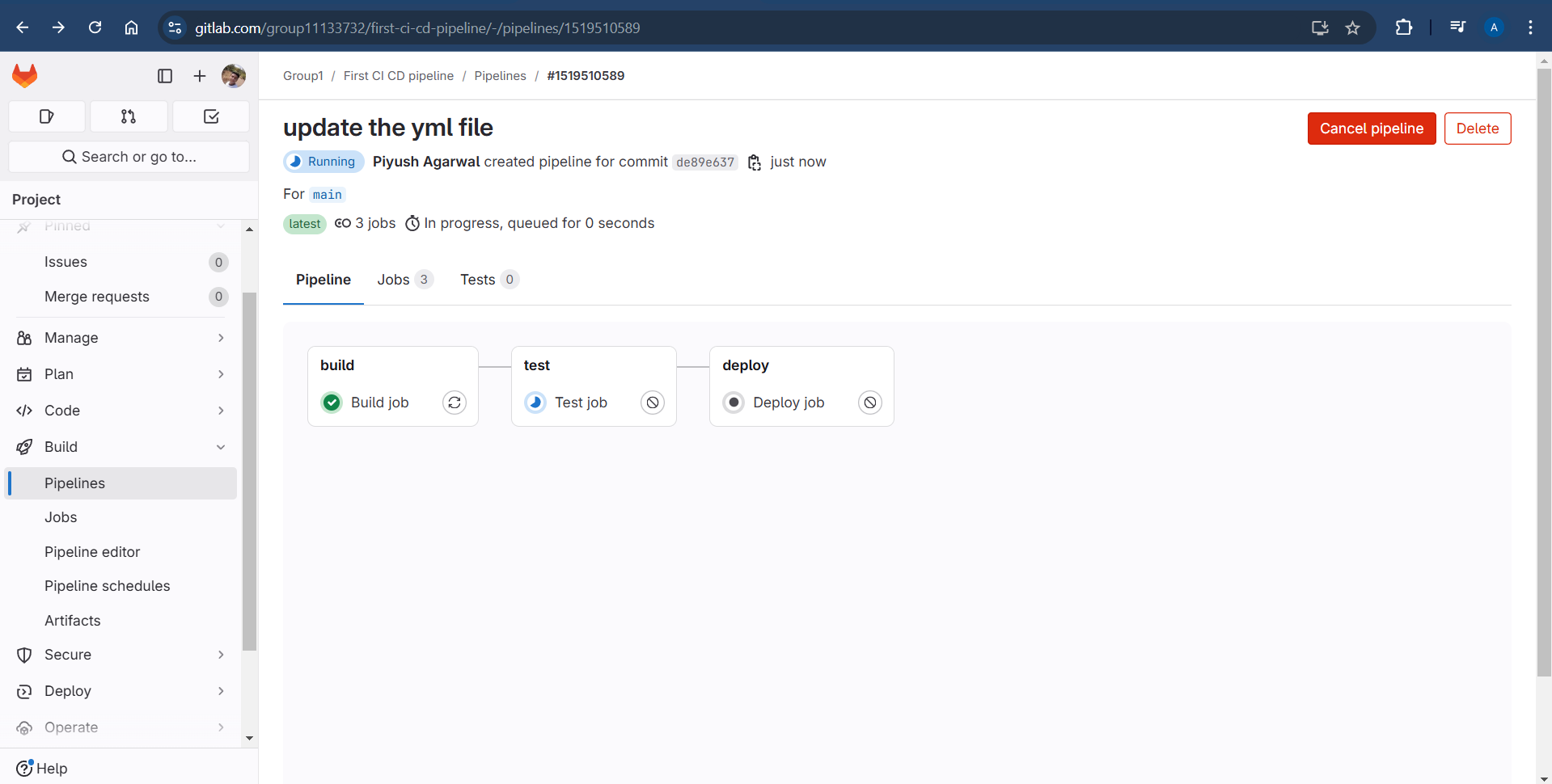


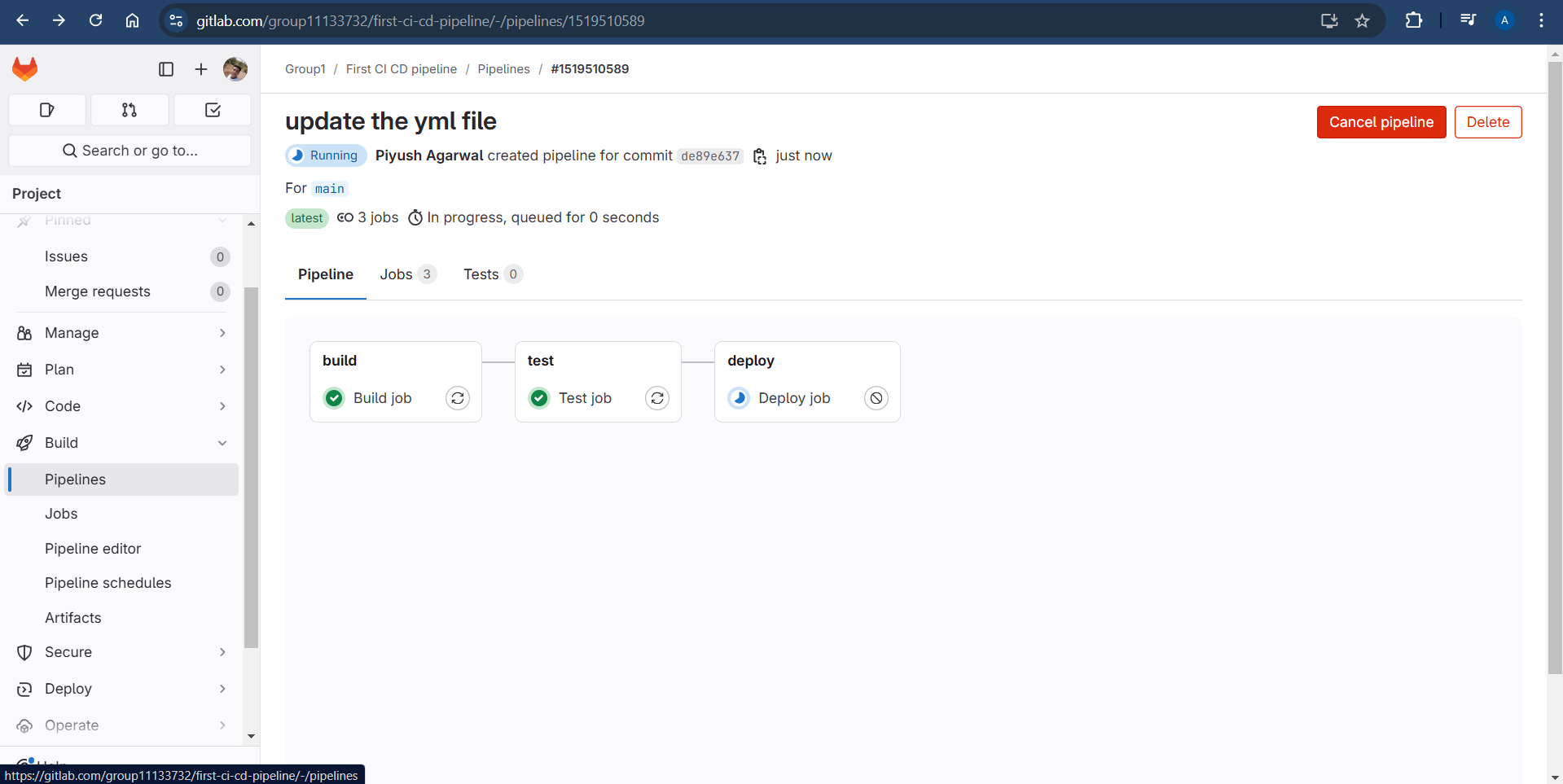
Here,

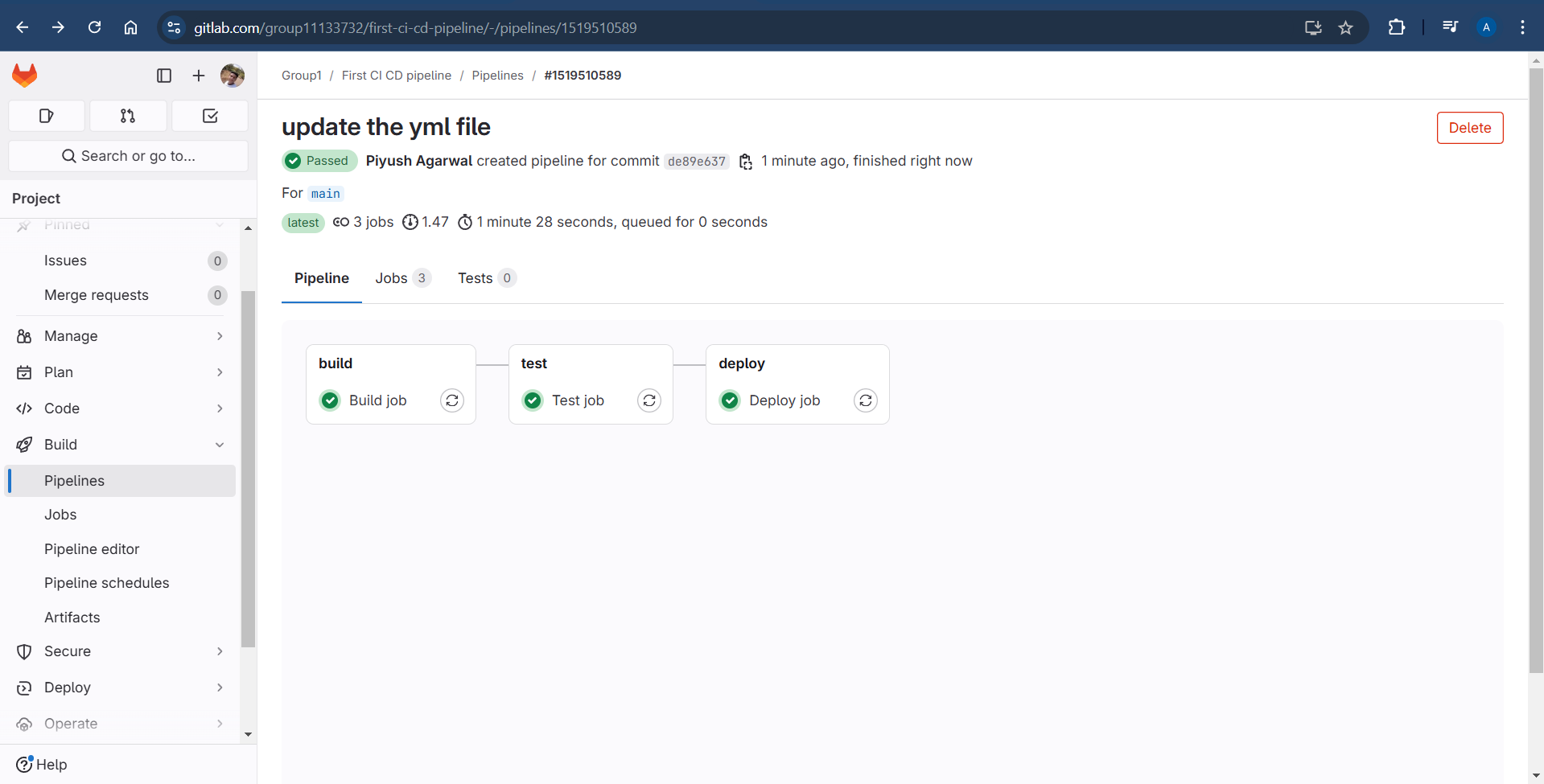
* The “stages” keyword defines the different phases of the pipeline. In this case the stages are: build, test and deploy
* Each job is associated with a stage and contains the “script” keyword, which lists the commands to be executed during that job.

**Screenshots of GitLab:**

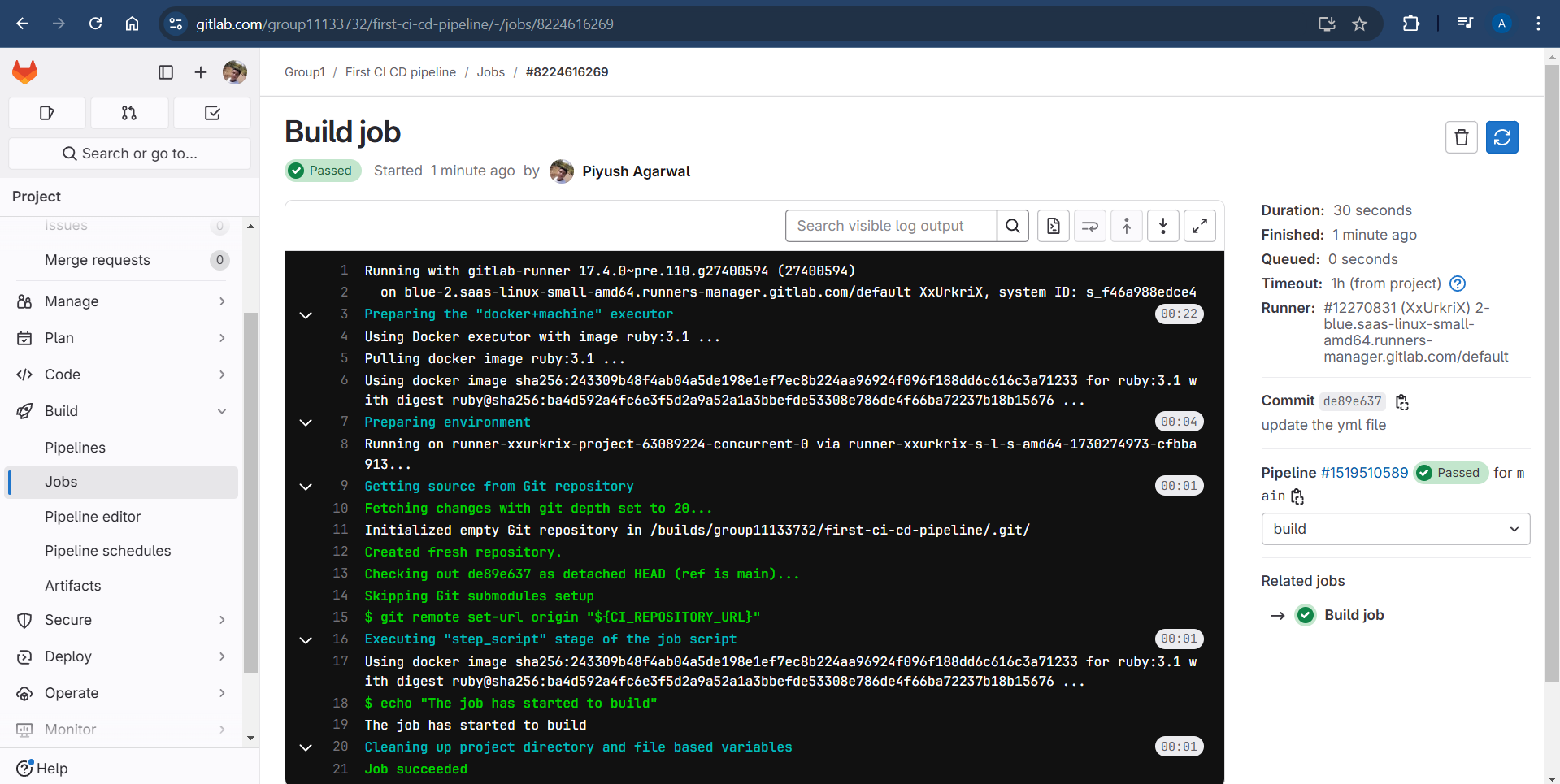




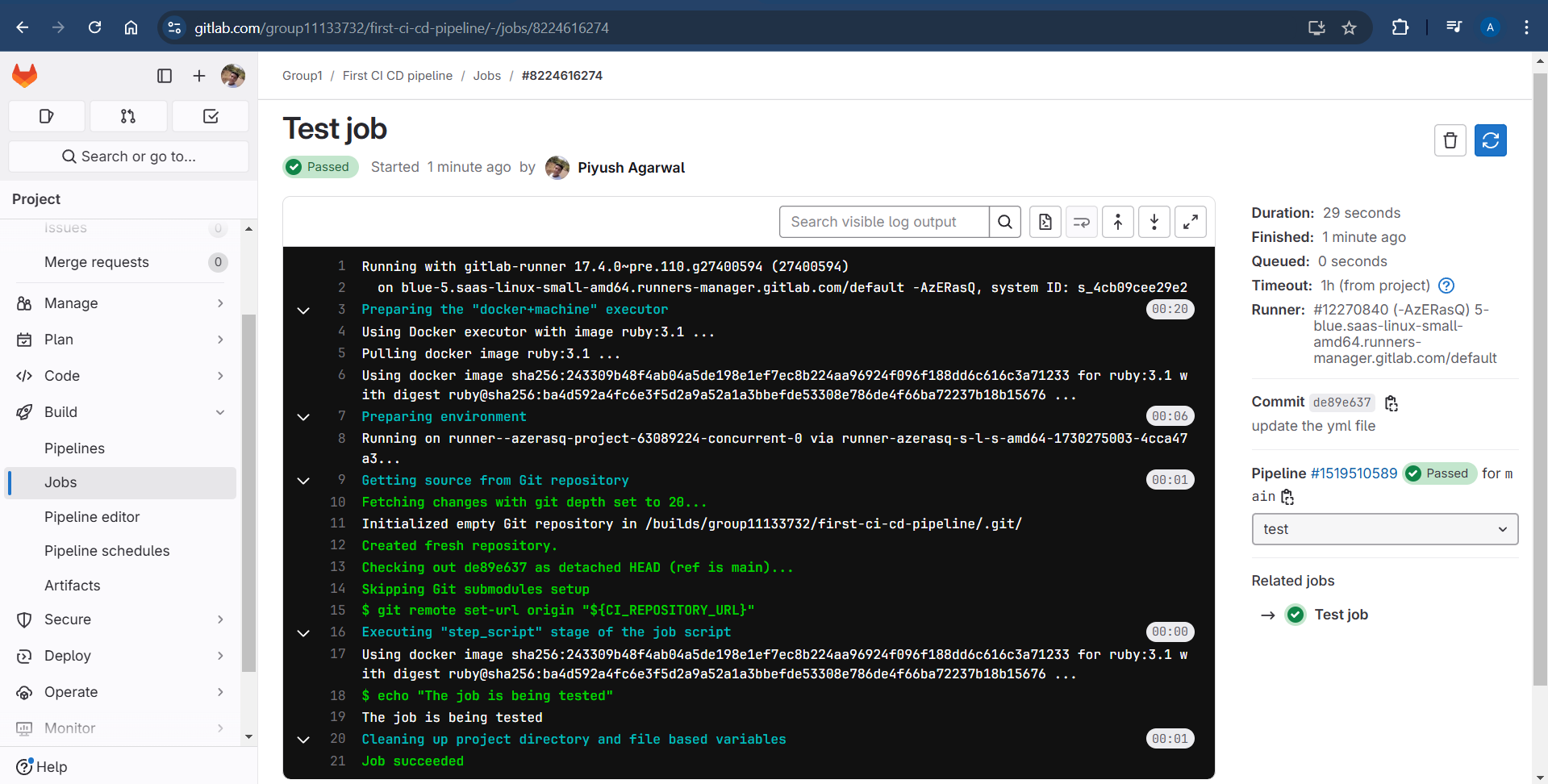




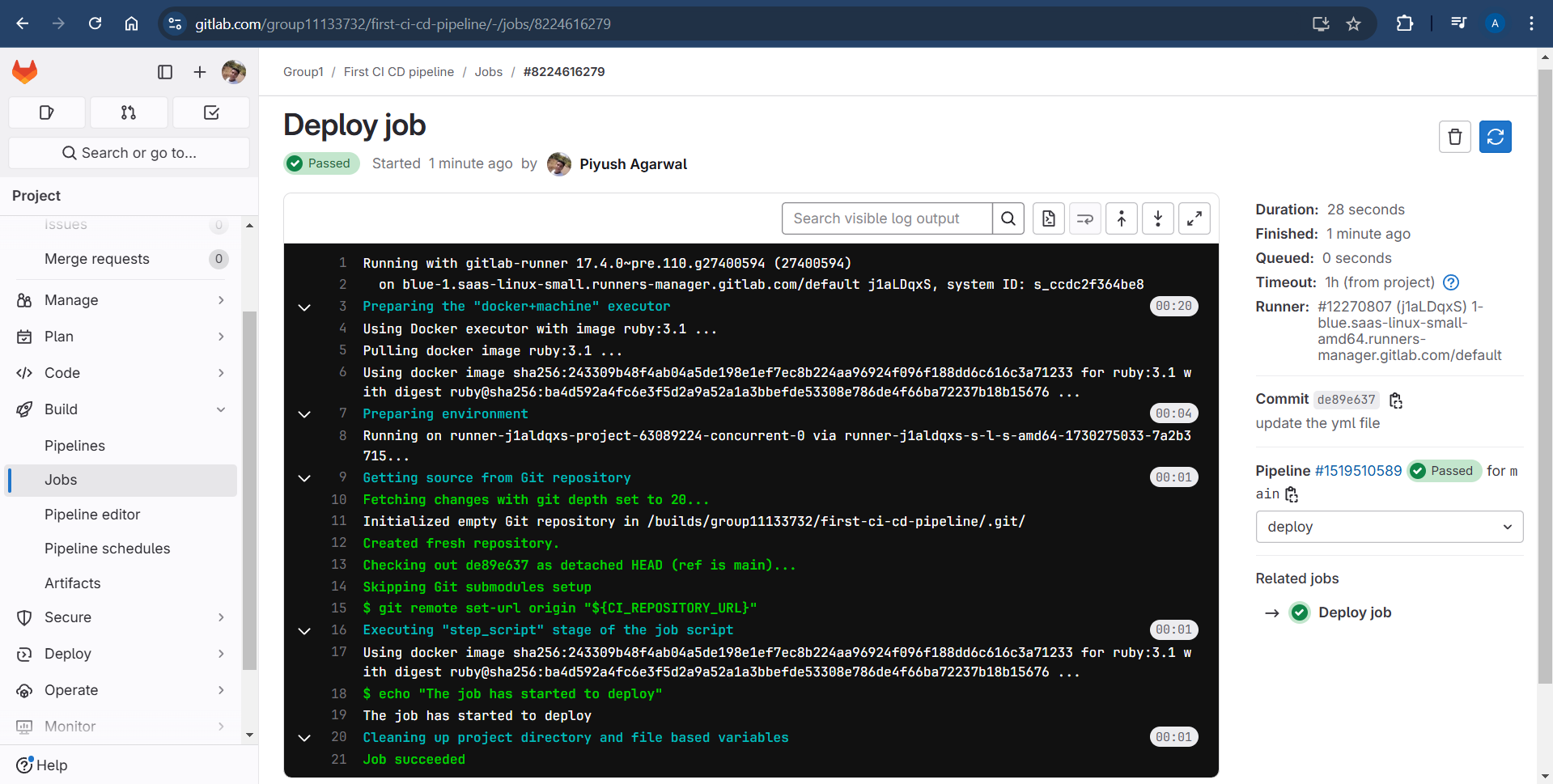
**Build Job:**



**Test Job:**



**Deploy Job:**



**Conclusion:**

In conclusion, this .gitlab-ci.yml file outlines a simple yet effective GitLab CI/CD pipeline with distinct stages for building, testing, and deploying an application. The structure ensures that each phase of the software development lifecycle is executed in a defined order, promoting automated workflows and increasing efficiency. By leveraging GitLab CI/CD, developers can achieve faster delivery times, reduce manual errors, and maintain high-quality code throughout the development process.