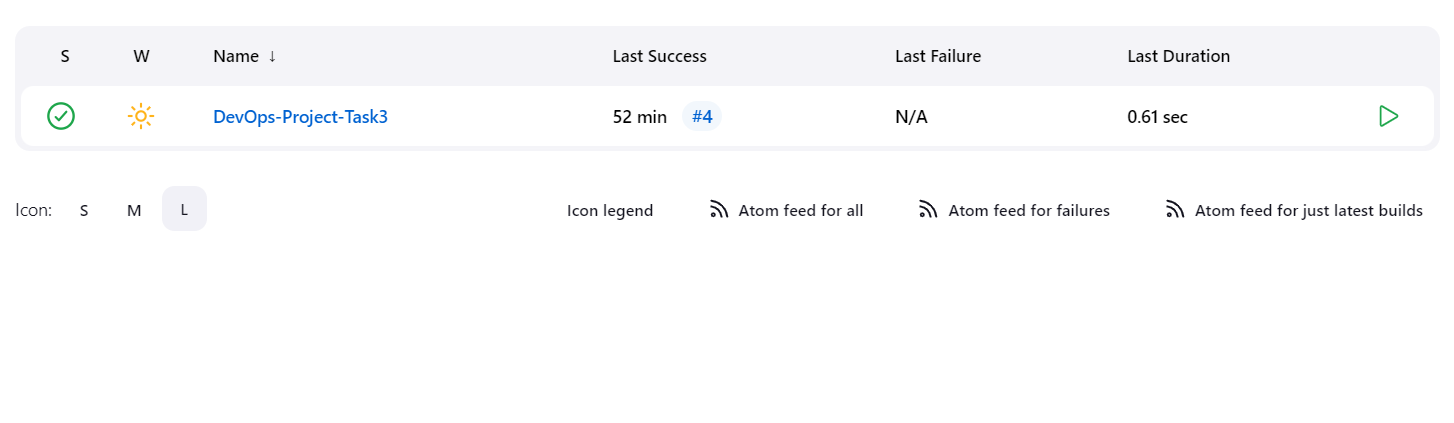


**Tool: Jenkins**

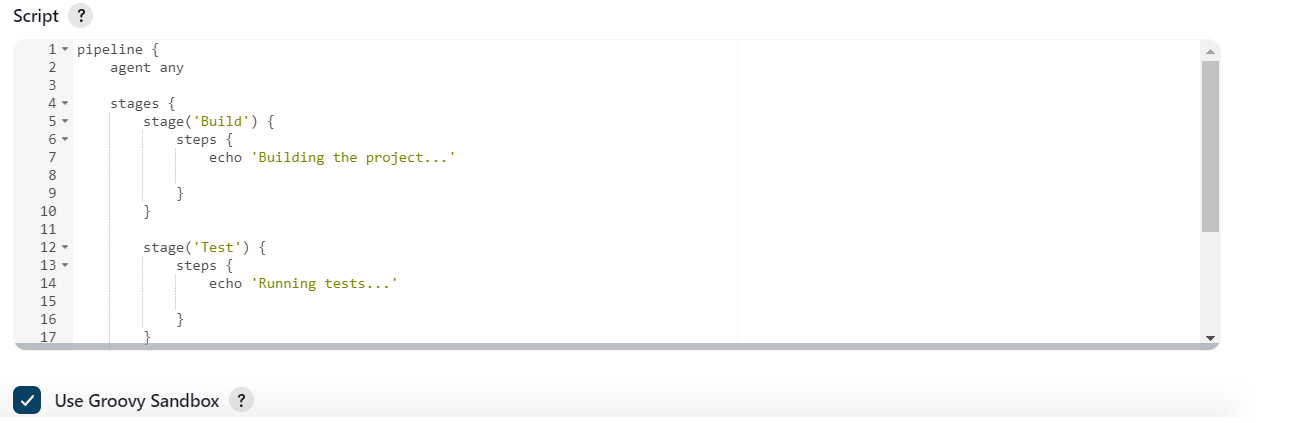
# Findings

Jenkins is a cloud server that provides a facility to build, test, and deploy applications in one go. We don’t have to use different platforms for building, testing, and deploying code, just create a job in Jenkins then it will build, test and deploy it on the server.

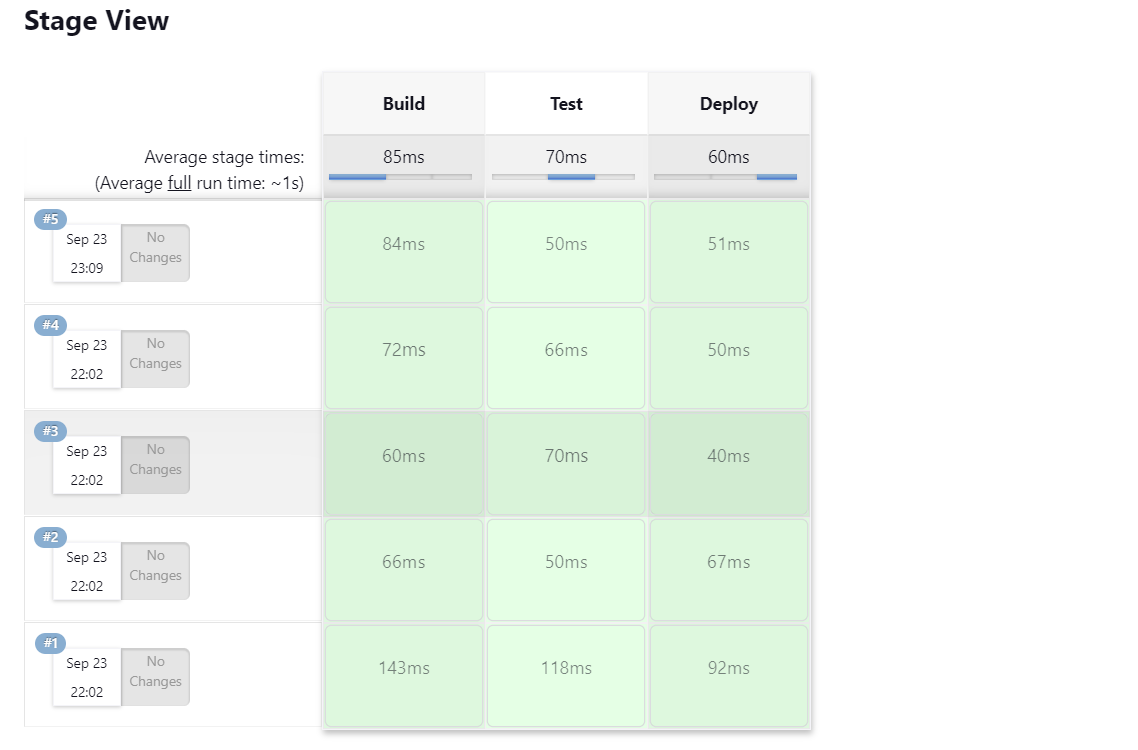
# Screenshots



In this screenshot, a job is created in a Jenkins dashboard which is then used to build, test, and deploy the code.

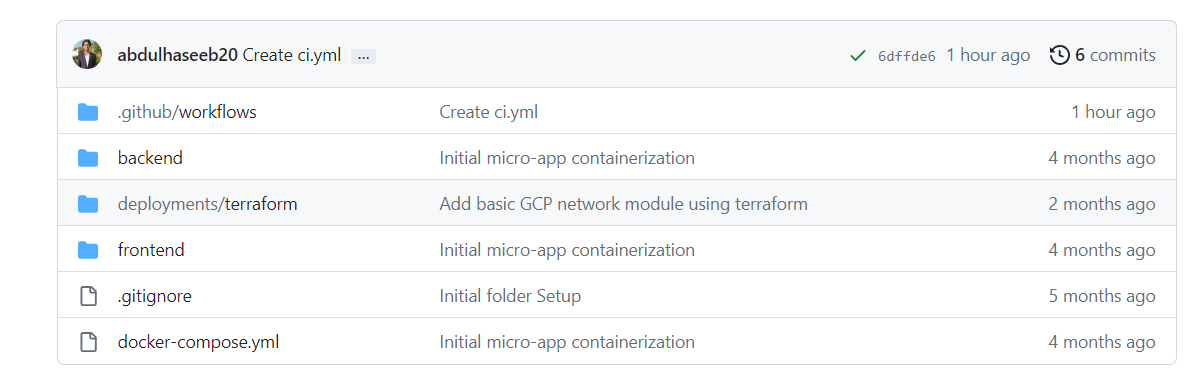


This is a simple pipeline written in groovy language. In this pipeline, simple print statements are coded to check whether the pipeline is built, tested, and deployed successfully or not.



In this screenshot, job is built, then tested, and then finally deployed on the server.

Jenkins is a reliable tool for deploying small to large scale applications as shown as we can easily build, test, and deploy within no time. Also, it is more secure than other deployment servers. For this reason, Jenkins is preferred in most enterprises.

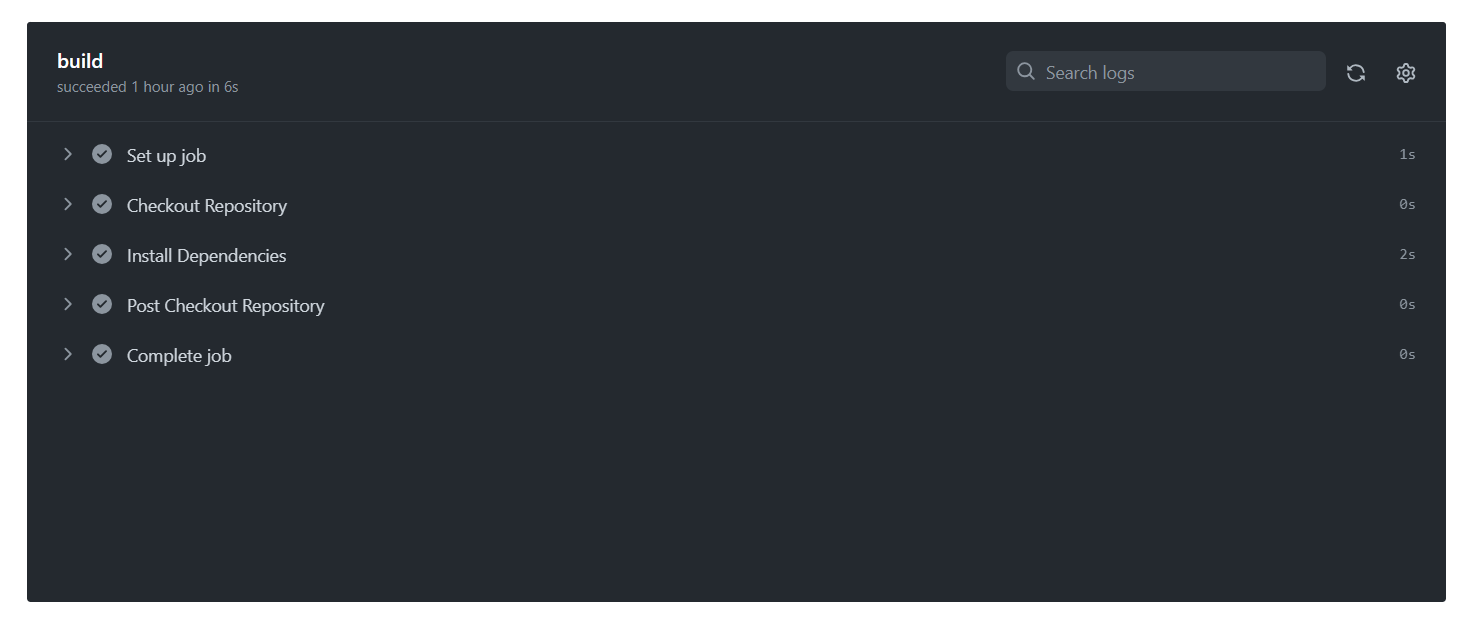


Then, I created a workflow action in the repository provided by the instructor, [Questra-Digital/ts-micro-app](https://github.com/Questra-Digital/ts-micro-app).

A screenshot of a computer

Description automatically generated

Then I created a ci.yml file in which all depenedencies, push, pull requests and jobs are created. Now, just need to run this build.



Now as shown above, job is built and all dependencies are created and job is completed.

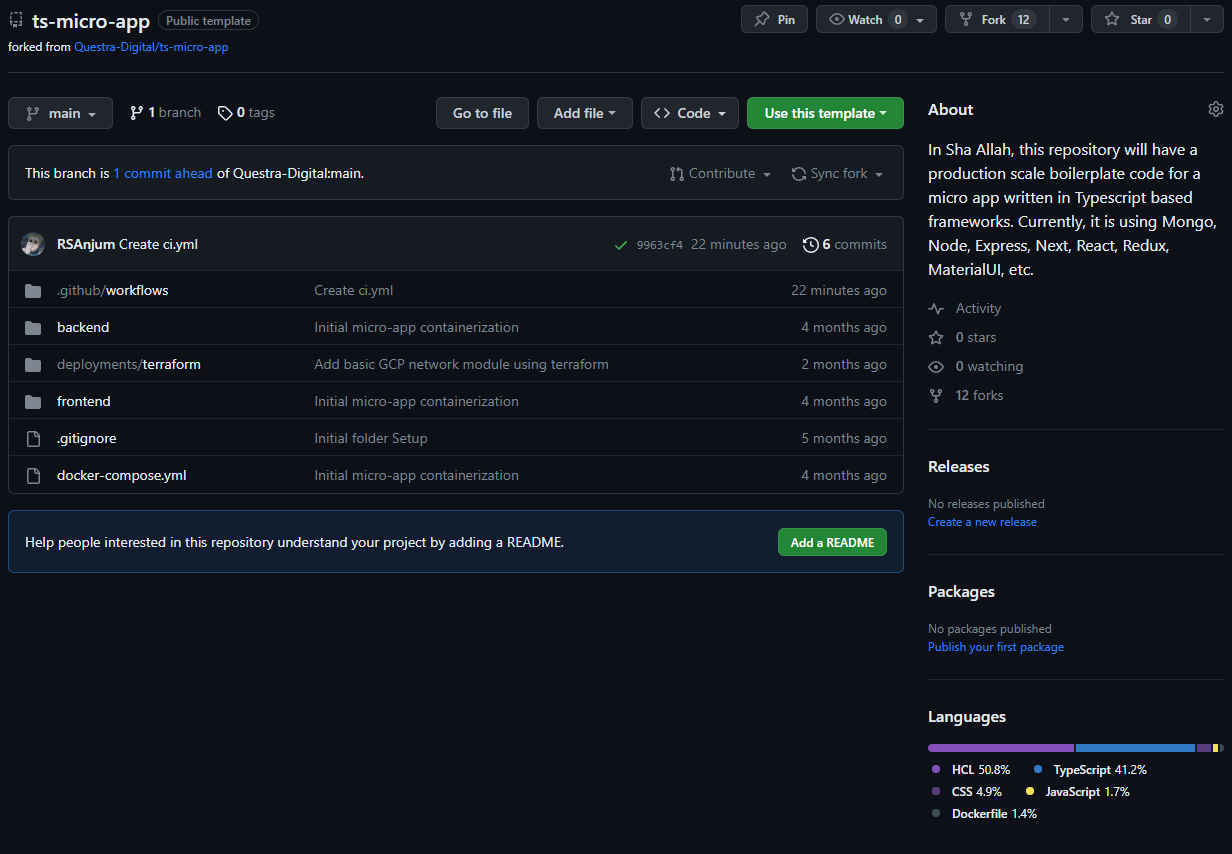
**Tool: GitHub Actions**

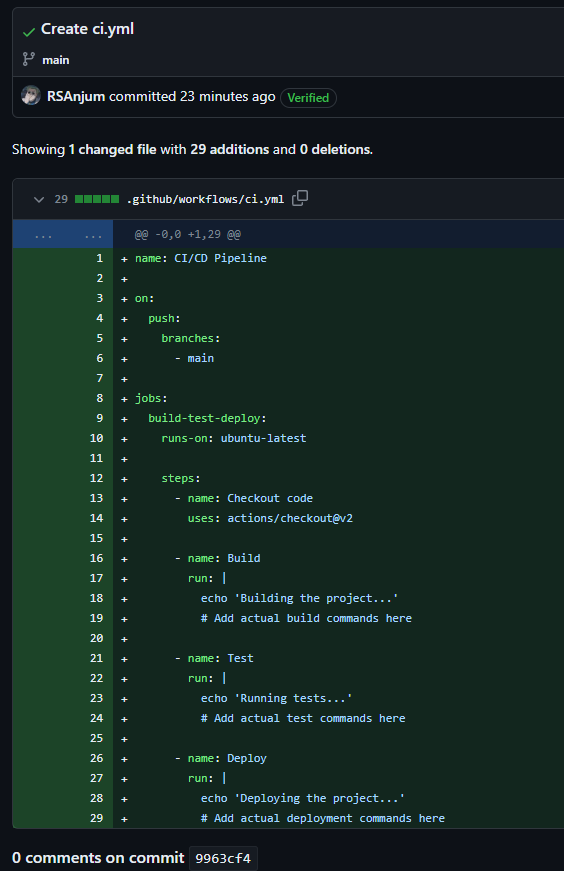
GitHub Actions:

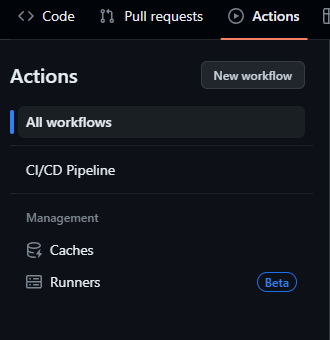
GitHub Actions is an integrated automation platform tightly integrated with GitHub repositories. It empowers developers to automate workflows, including building, testing, and deploying code, directly within their GitHub repositories. GitHub Actions offers a native and code-centric approach to automation, allowing you to define workflows using YAML files stored in your repository, making it easy to version and collaborate on your automation processes.

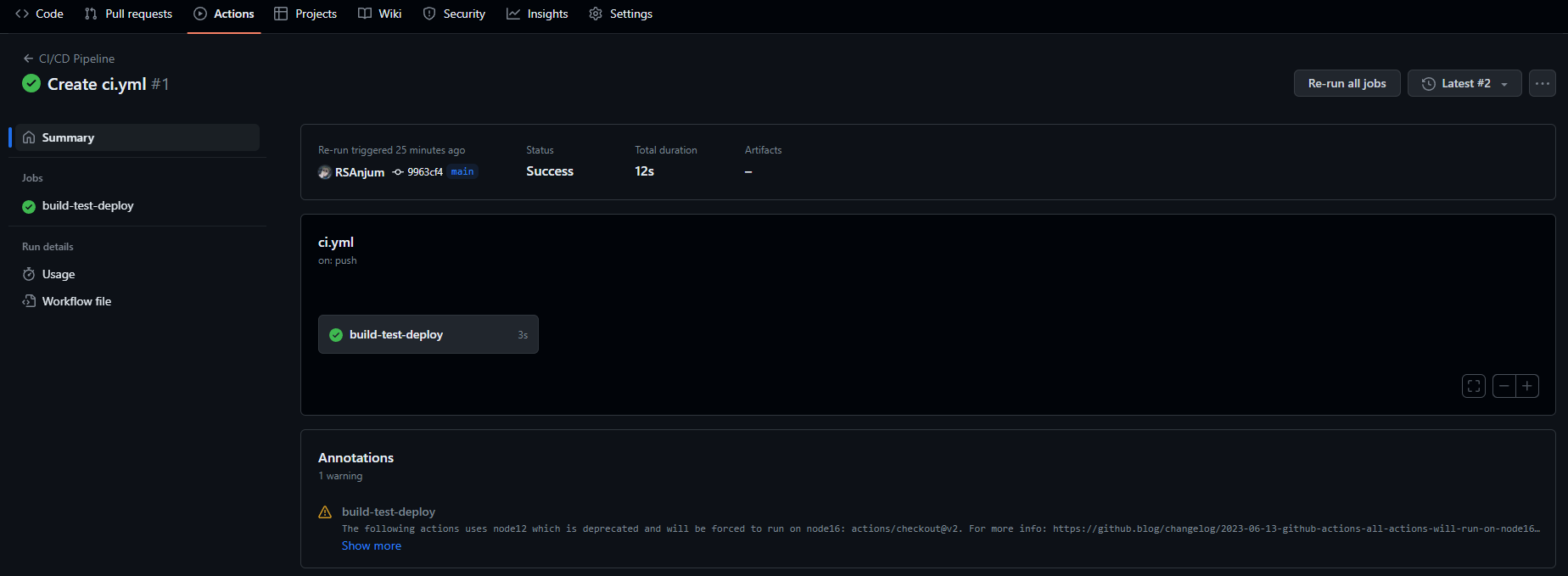
Comparison with Jenkins:

While Jenkins offers a comprehensive cloud server solution for building, testing, and deploying applications, GitHub Actions provides a more streamlined and tightly integrated approach within the GitHub ecosystem. Jenkins is a standalone cloud server, whereas GitHub Actions operates seamlessly within the GitHub repository, making it convenient for teams already using GitHub for version control. Both tools are powerful in their own right, with Jenkins offering more extensive customization options, while GitHub Actions excels in simplicity and integration. The choice between them depends on your specific project requirements and your team's familiarity with each platform.



I forked the main branch as instructed by the instructor and created the ci.yml file in .github/workflows directory.  
  


I added the code for the pipeline and saved it in the yml file. I committed all the changes to the recently forked branch.  


Then, under the Actions tab, i could see github recognising my yml file and displaying it as an independent workflow.  
  


As we can see in the screenshot, the creation of the pipeline, and the building/running was also a success as tested by the Github, thus completing the assigned task.