**Task 1: Describe the process of creating a new issue or bug report in the issue tracking system.**

**Here I’m using Jira, so we have to log in to our account. Click on the Create Issue. We must select the project or repository where the bug must be fixed. There we have to give a simple summary of the issue in a clear manner. Then in the issue description, we have to give as much information as possible about the issue, Attach any relevant screenshots. We can add tags explaining the issue. We can assign team members who are the contributors of the project. Simply type @followed by the contributor's username or from the dropdown list of usernames. And click Create the issue is now created.**

**Task 2: Explain how to assign the issue to a specific team member.**

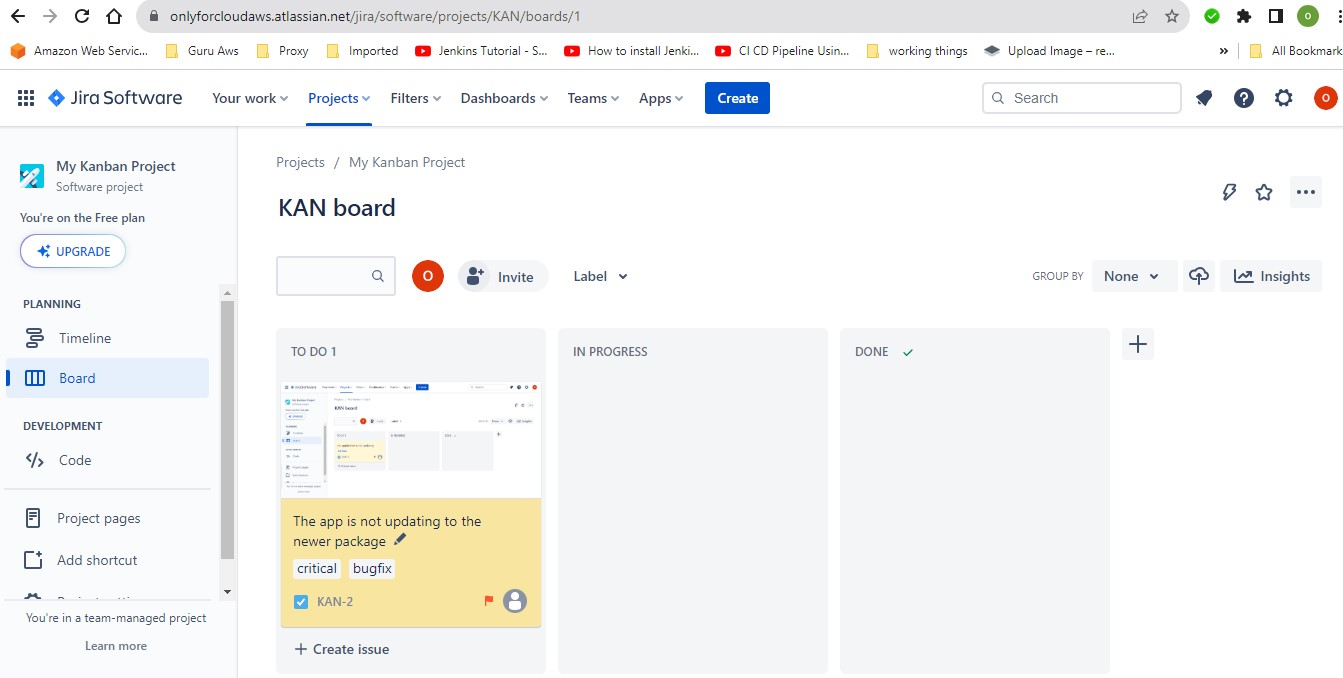
**We can assign team members who are the contributors of the project. Simply type @followed by the contributor's username or from the dropdown list of usernames. The user will receive a notification as an email after their assigning.**

**Task 3: Describe how to use Git branches and pull requests to work on the issue, ensuring that changes are tracked and reviewed.**

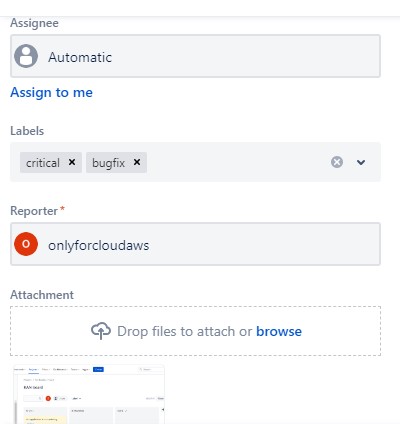
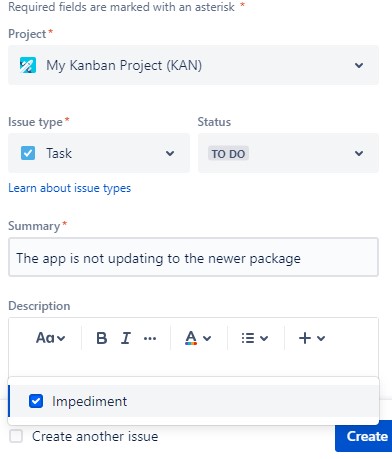
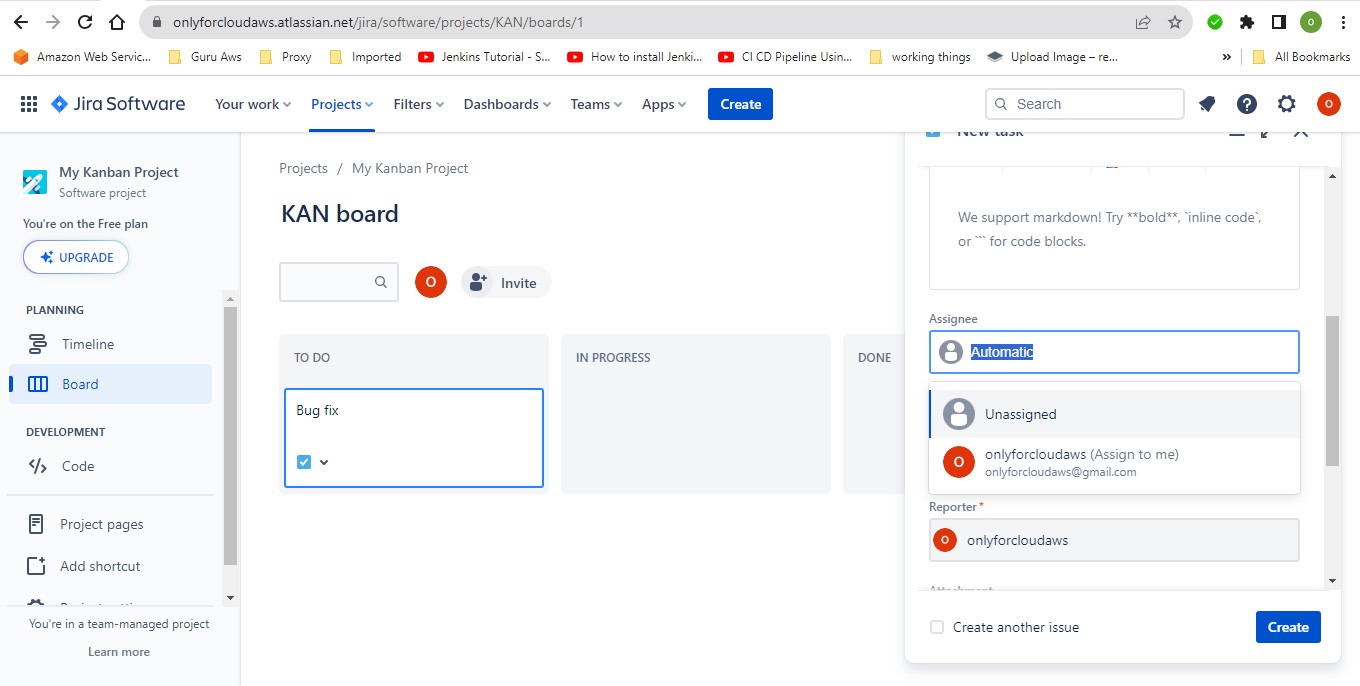
**To work on this exact issue, we can create a branch with the exact issue name so that we can group and work on the changes as a review. Each one working on the issue should give a proper commit message to keep track of the process’s workflow. Once the changes or fix is done, we can push those changes to the remote repo. And now we can raise a pull request, and in the pull request description, we can mention the changes or stages that we have worked for the issue or bug. Once after reviewing the code, we can merge it with the main branch if the code is perfect and the bug is fixed. If else, we can ask the user to work on the code again.**

**Question 1 screenshot for your reference.**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**2. Task Allocation with Jenkins and CI/CD Pipelines**

**Task 1: Explain how to create a new task or user story for the feature.**

**Navigate to the project in Jira or configure and Click on "Create" or the "+" icon after that just Choose the issue type like task, user story and fill in the details like summary, description, priority, etc. later Click "Create" to create the task.**

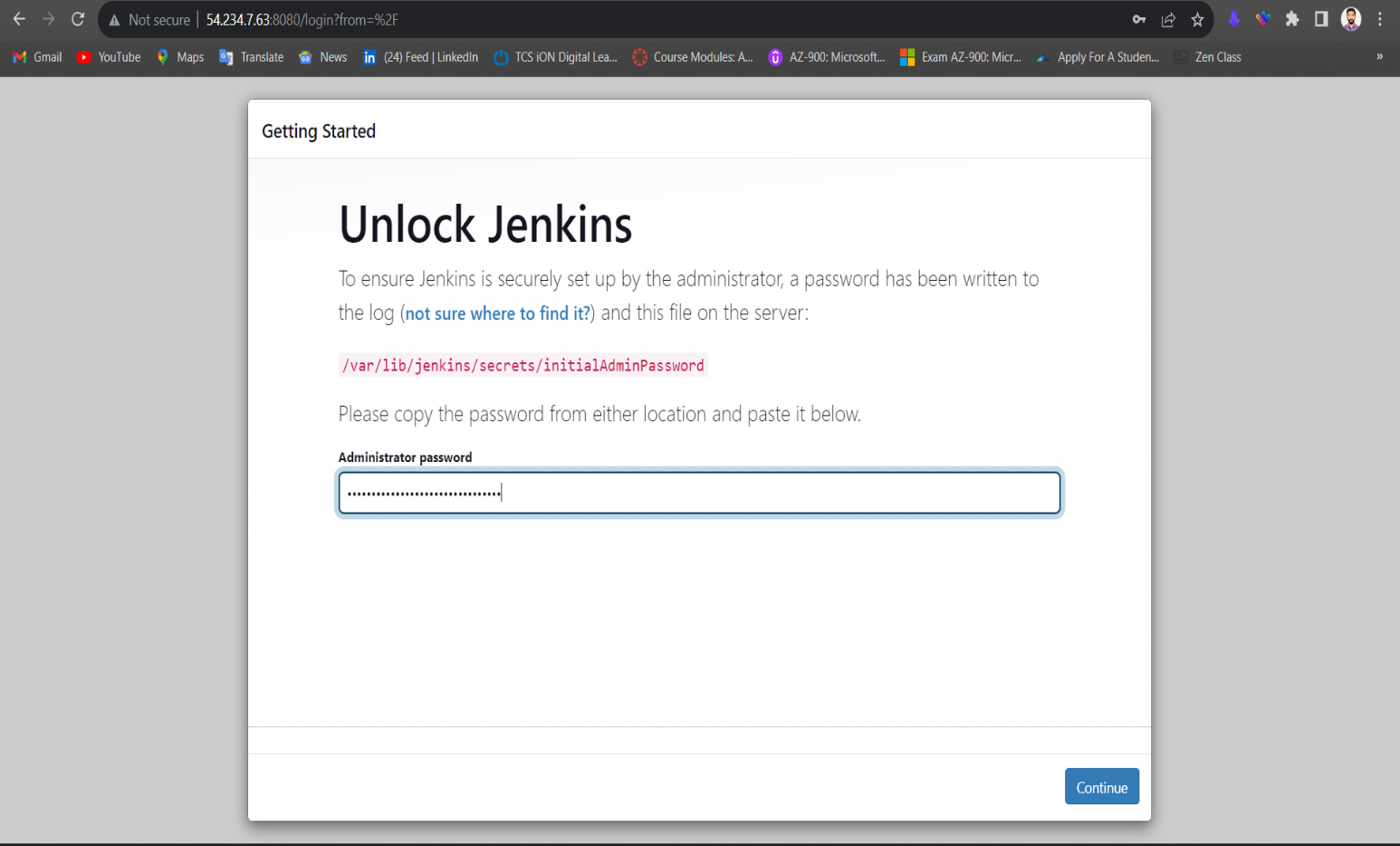
**Task 2: Describe how to assign the task to a developer and communicate the requirements.**

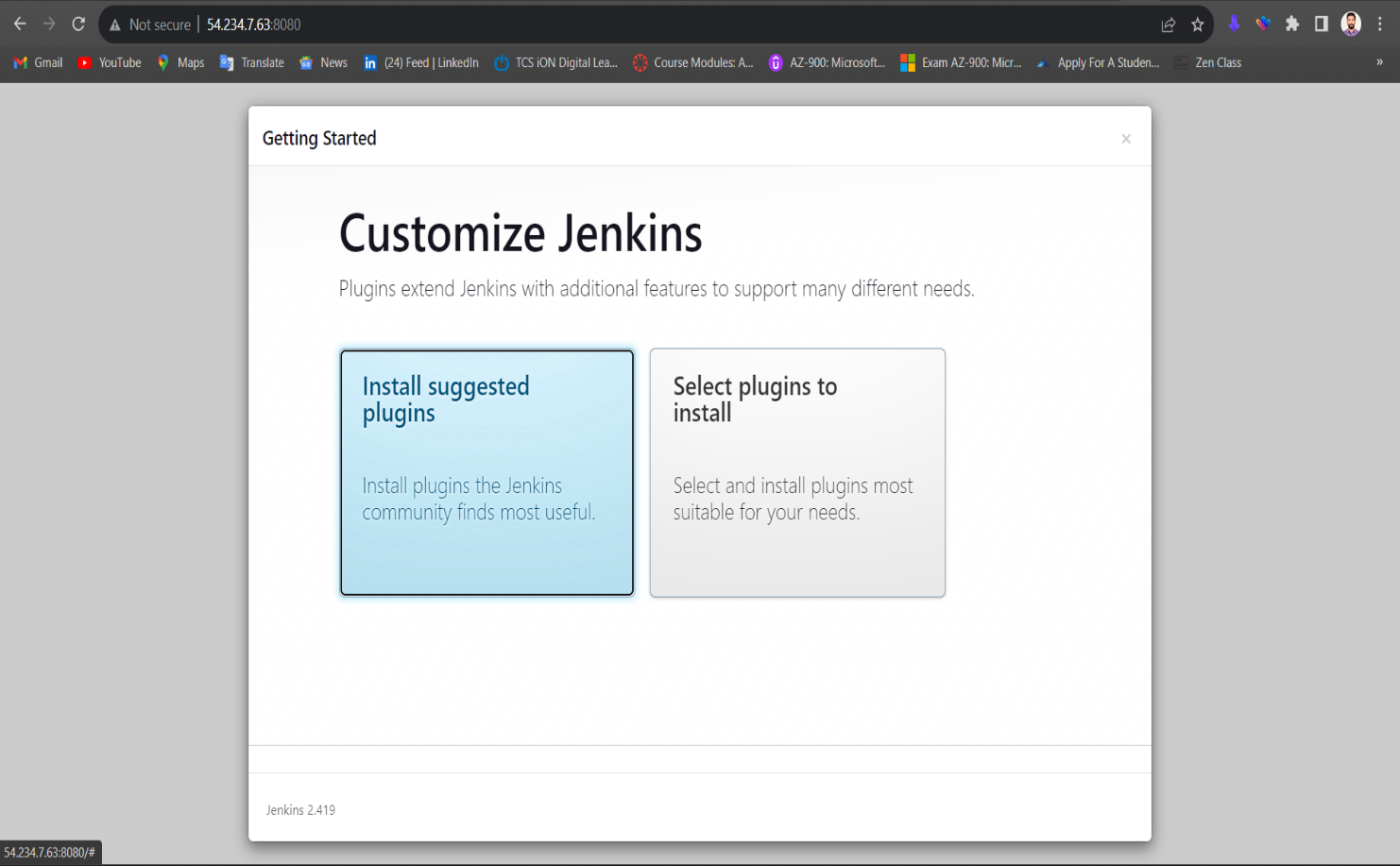
**Open the task you want to assign and Click on the "Assign" button to the team member to assign the task to from the dropdown and Use Jira comments or a project management tool to provide detailed requirements, acceptance criteria, and any other relevant information to the assigned developer.**

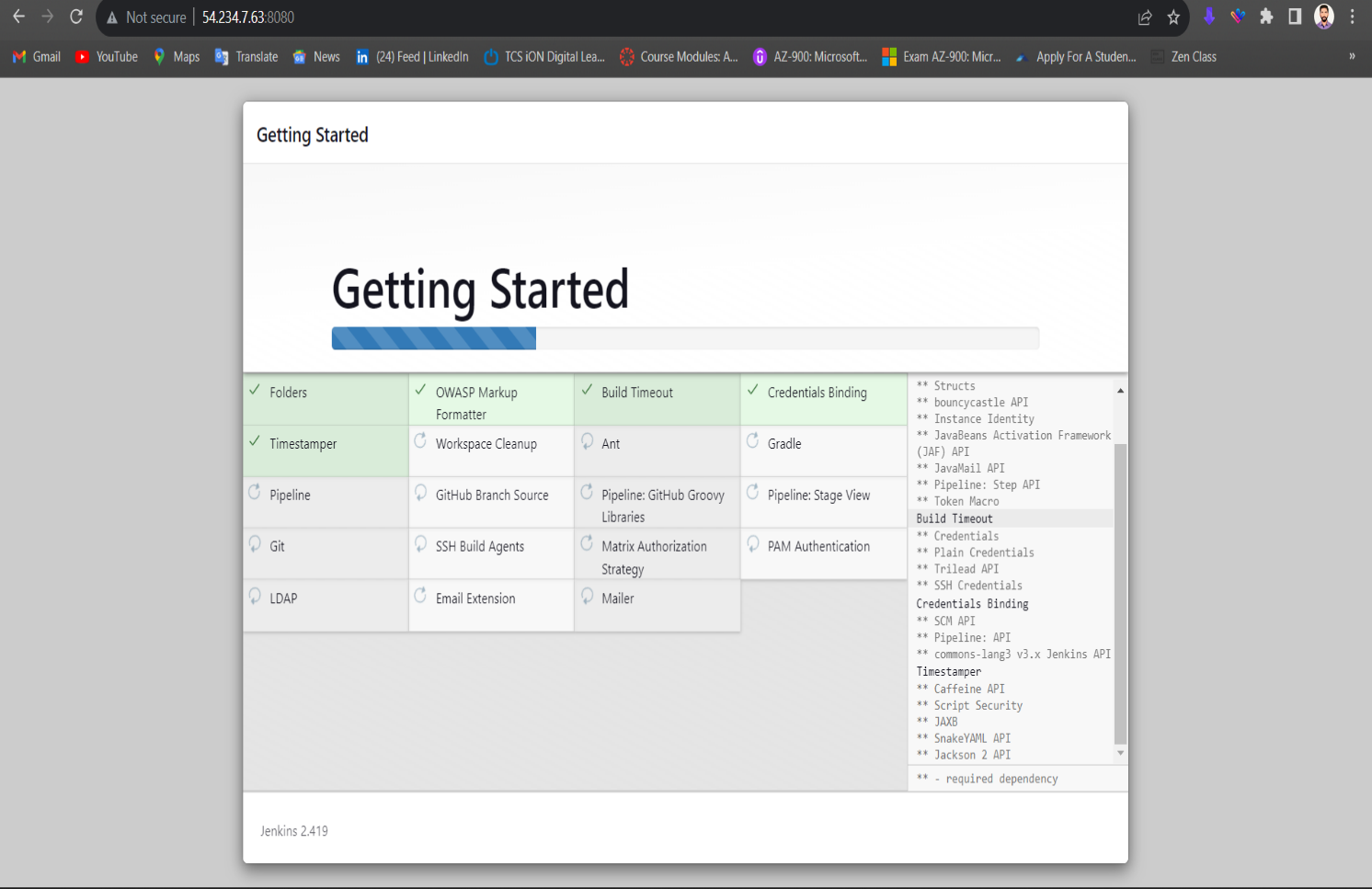
**Task 3: Outline the steps to integrate the new feature into the CI/CD pipeline, including creating a branch, writing tests, and updating the Jenkins pipeline configuration.**

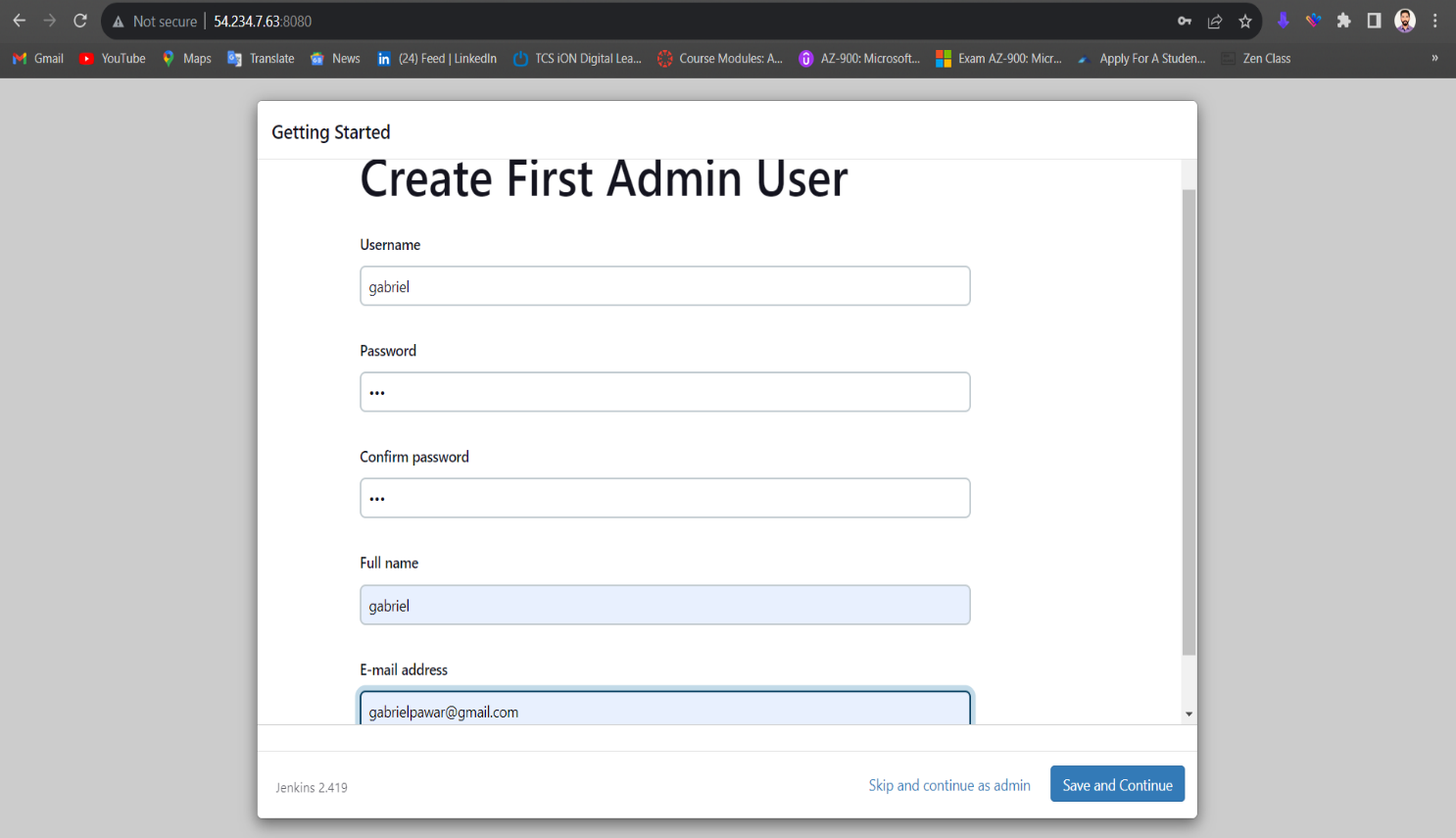
**Create a new branch with any name required with command : git checkout -b abc-branch Make necessary changes for the abc-branch and write appropriate tests. Commit the changes to the branch with command : git add . git commit -m "Adding new abc". After that Pushing the Branch using the command : git push origin abc-branch. Update the Jenkinsfile to include stages for building, testing, and deploying the new feature. Add a stage to run the tests for the new feature and Creating a Pull Request (PR): Go to the repository on GitHub , Click on "New Pull Request." Choose the base branch (e.g., main) and the compare branch (feature-branch). Click "Create pull request." Later you Review and Merge. Team members review the changes in the PR and Address feedback and make additional commits if necessary. Once approved, merge the PR into the main branch. By following these steps, a new feature is successfully integrated into the CI/CD pipeline, ensuring a streamlined development process.**

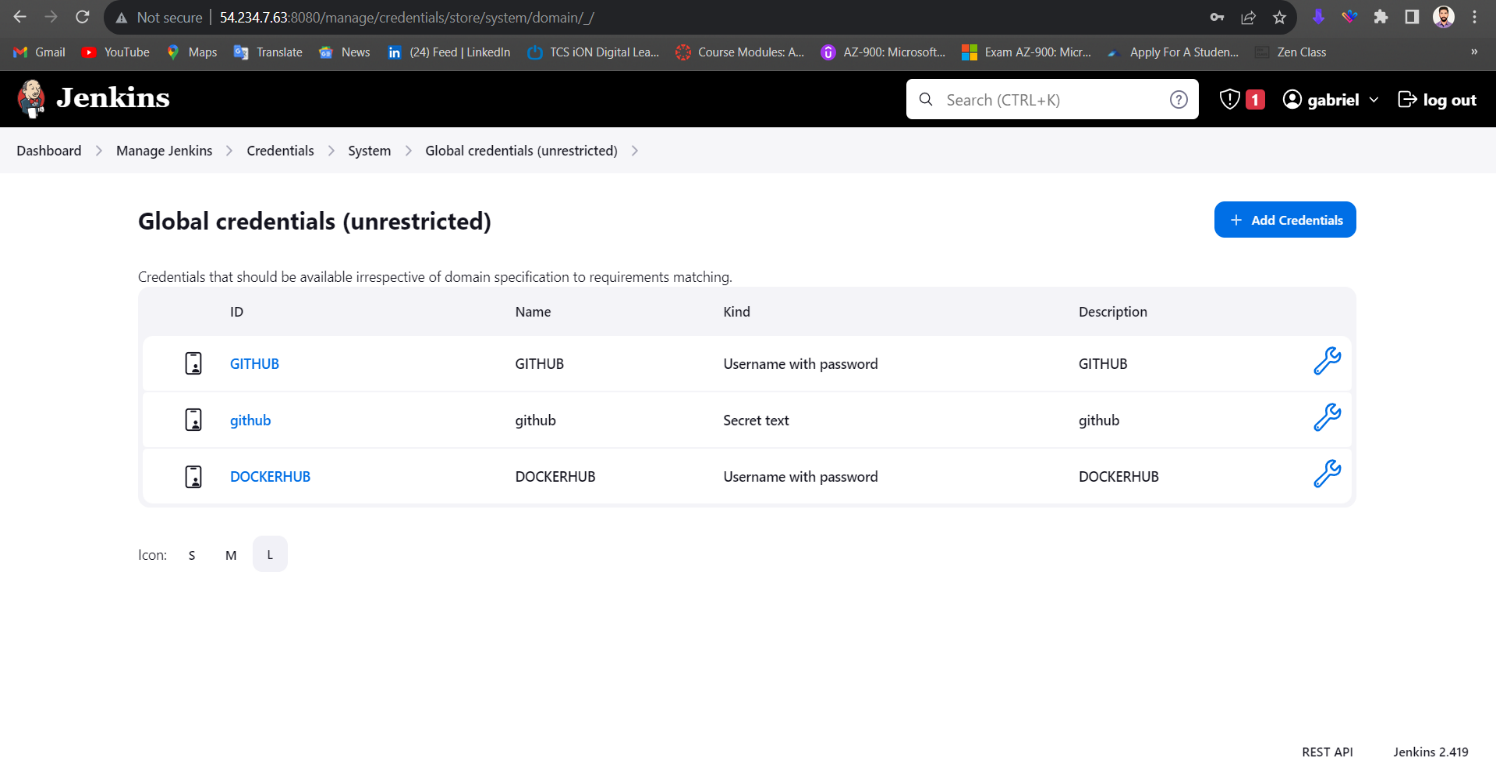
**Question 2 screenshot for your reference.**

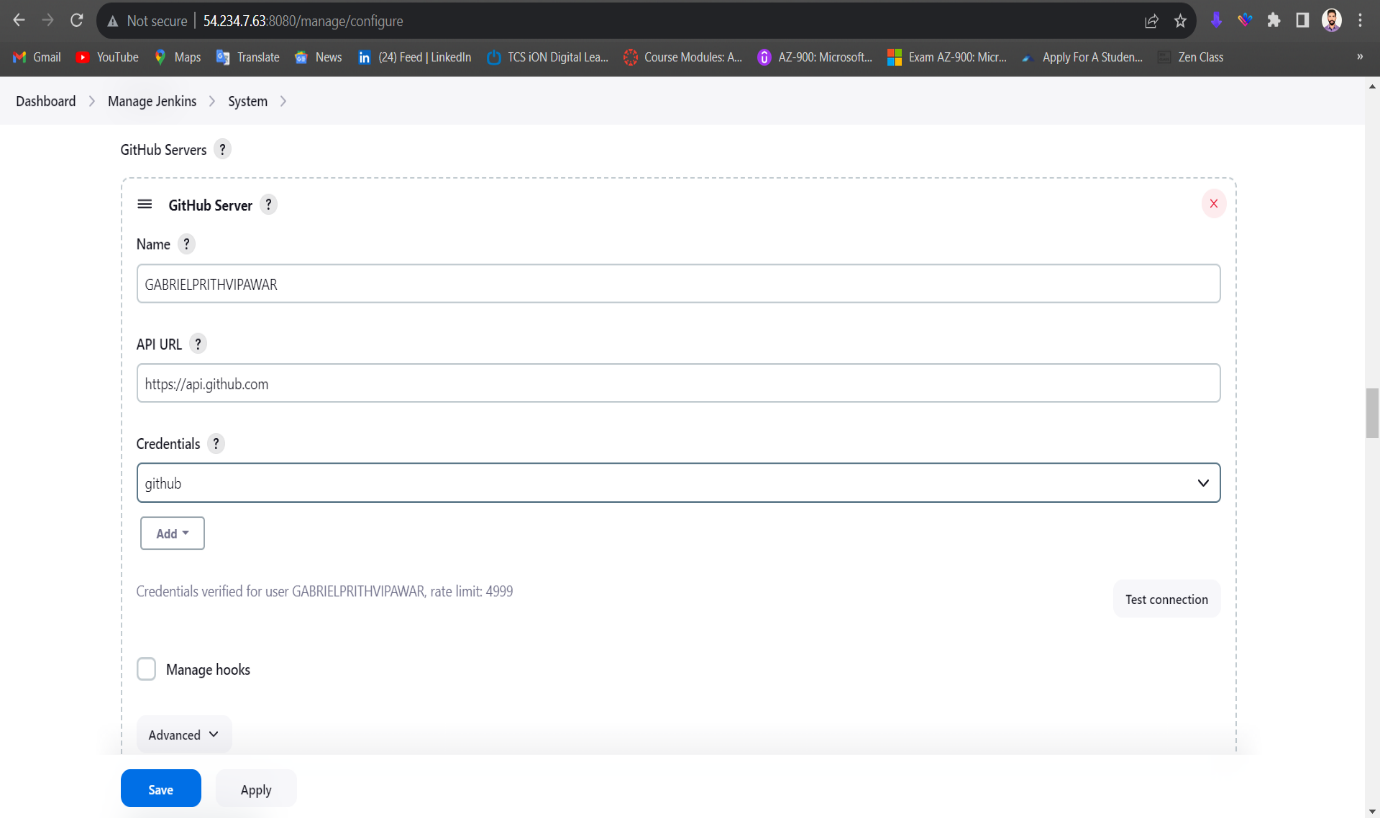
****

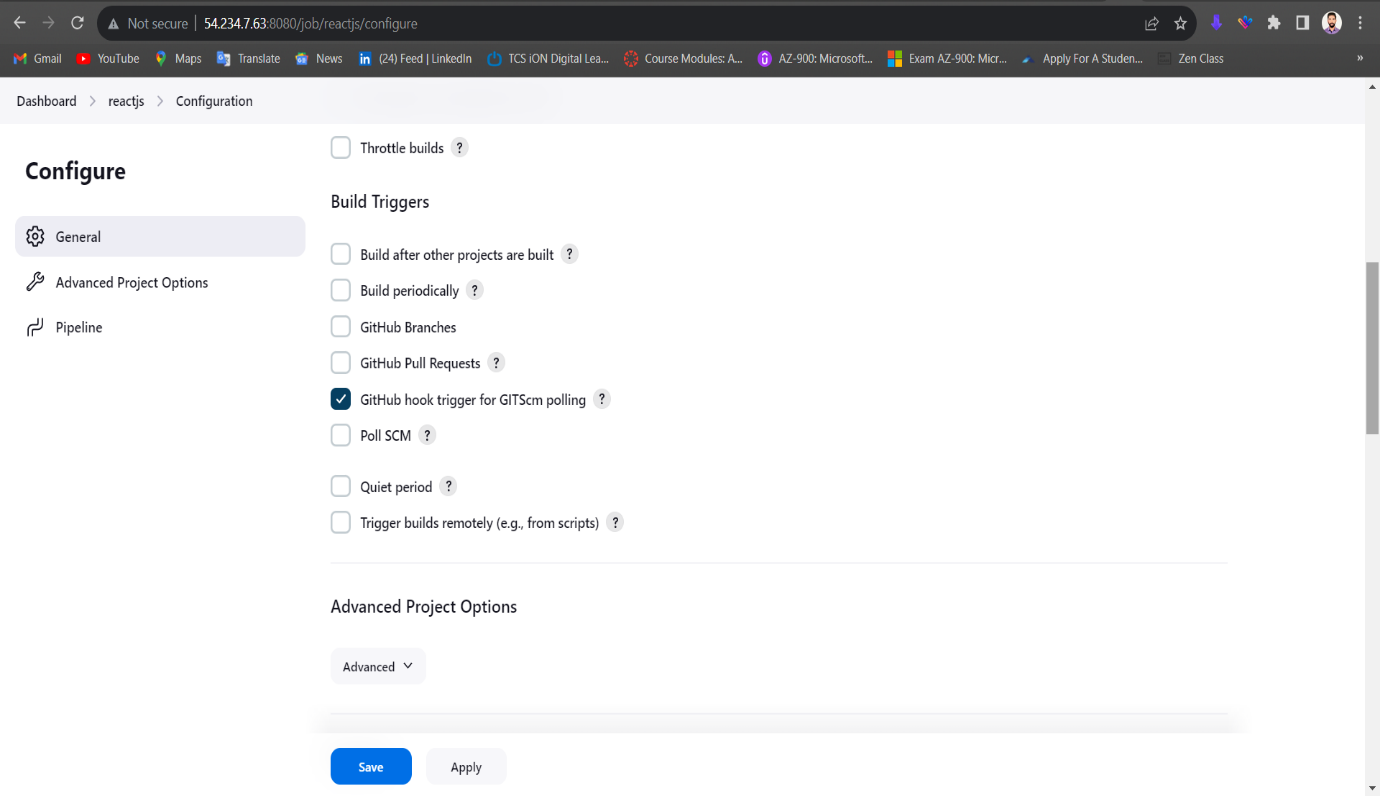
****

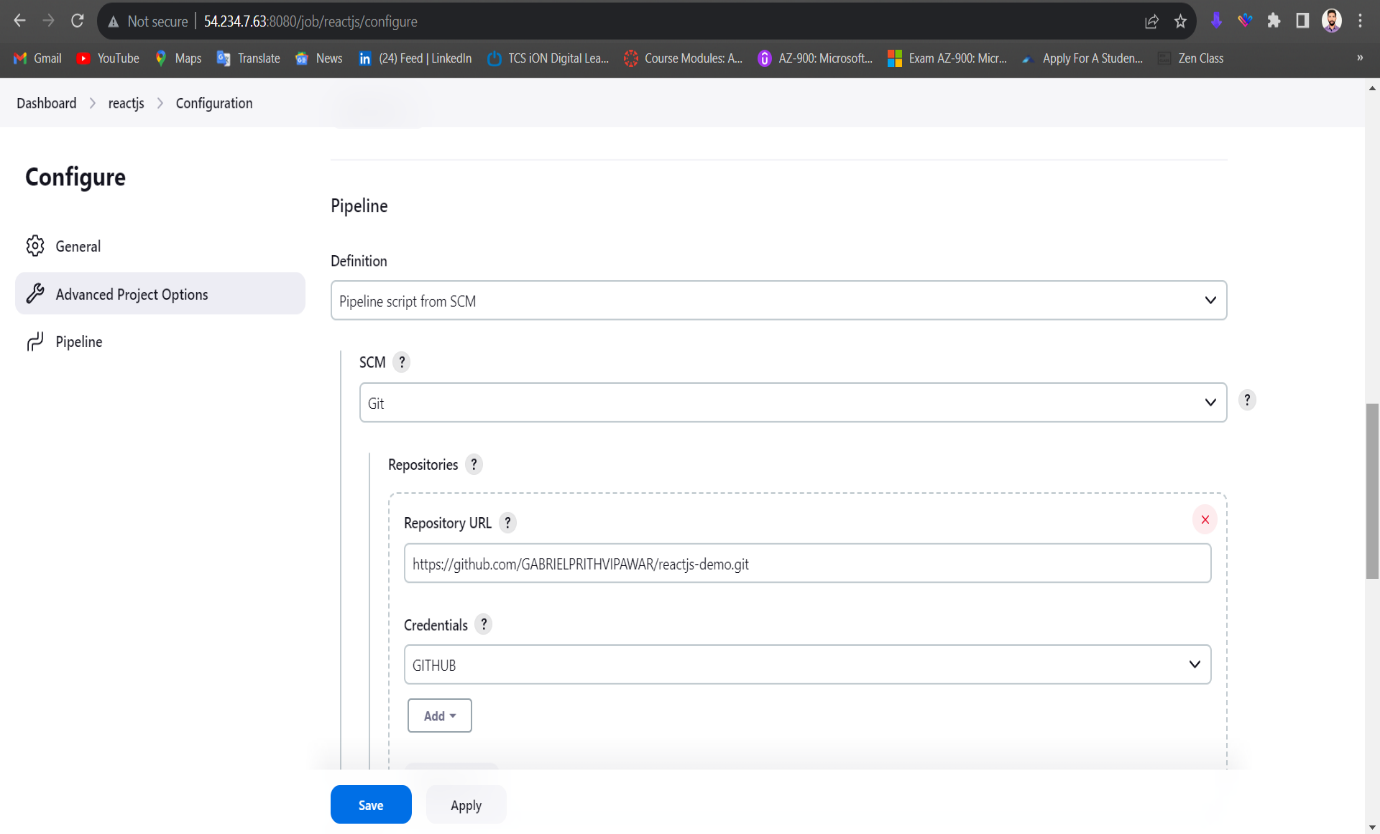
****

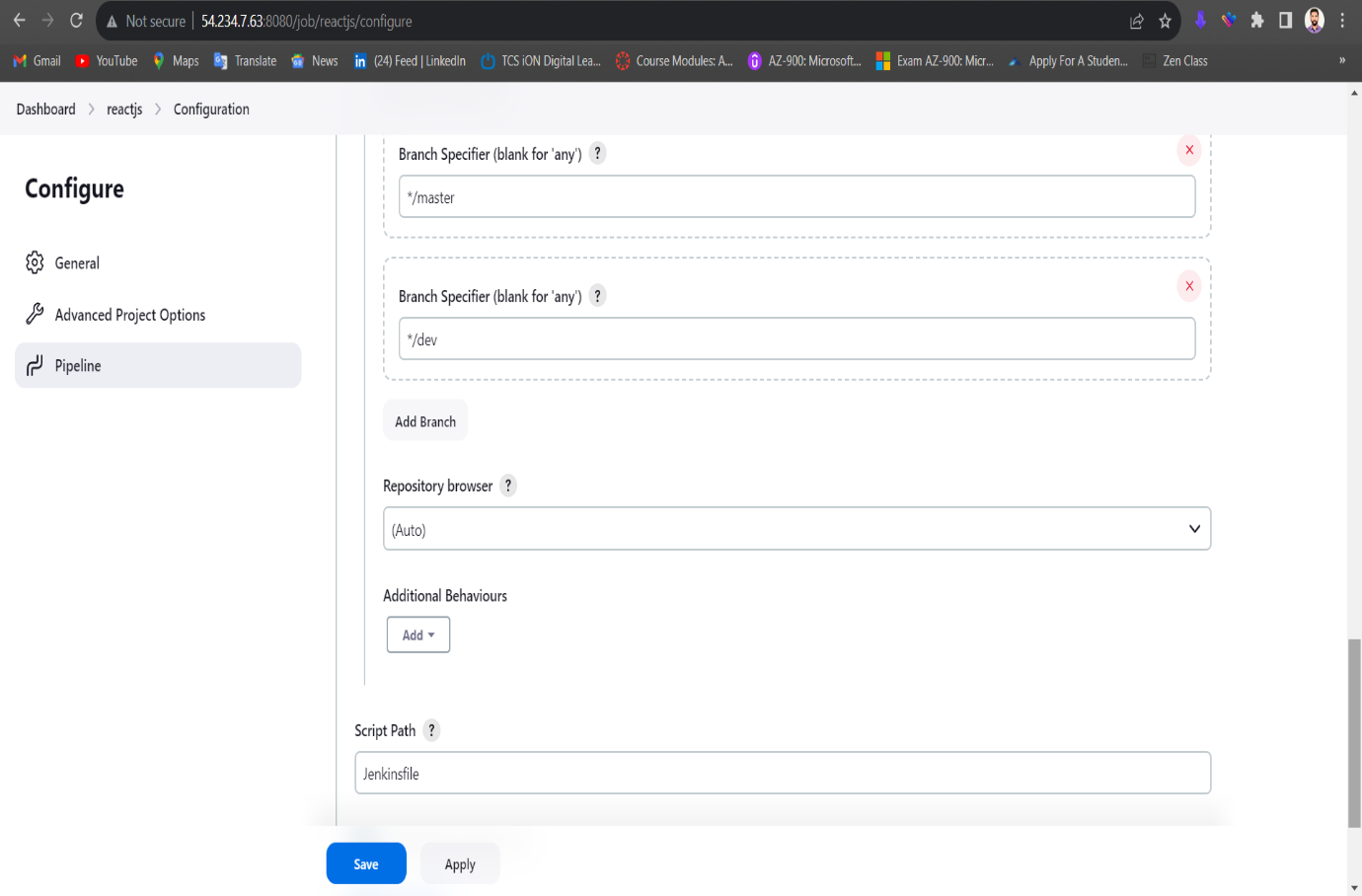
****

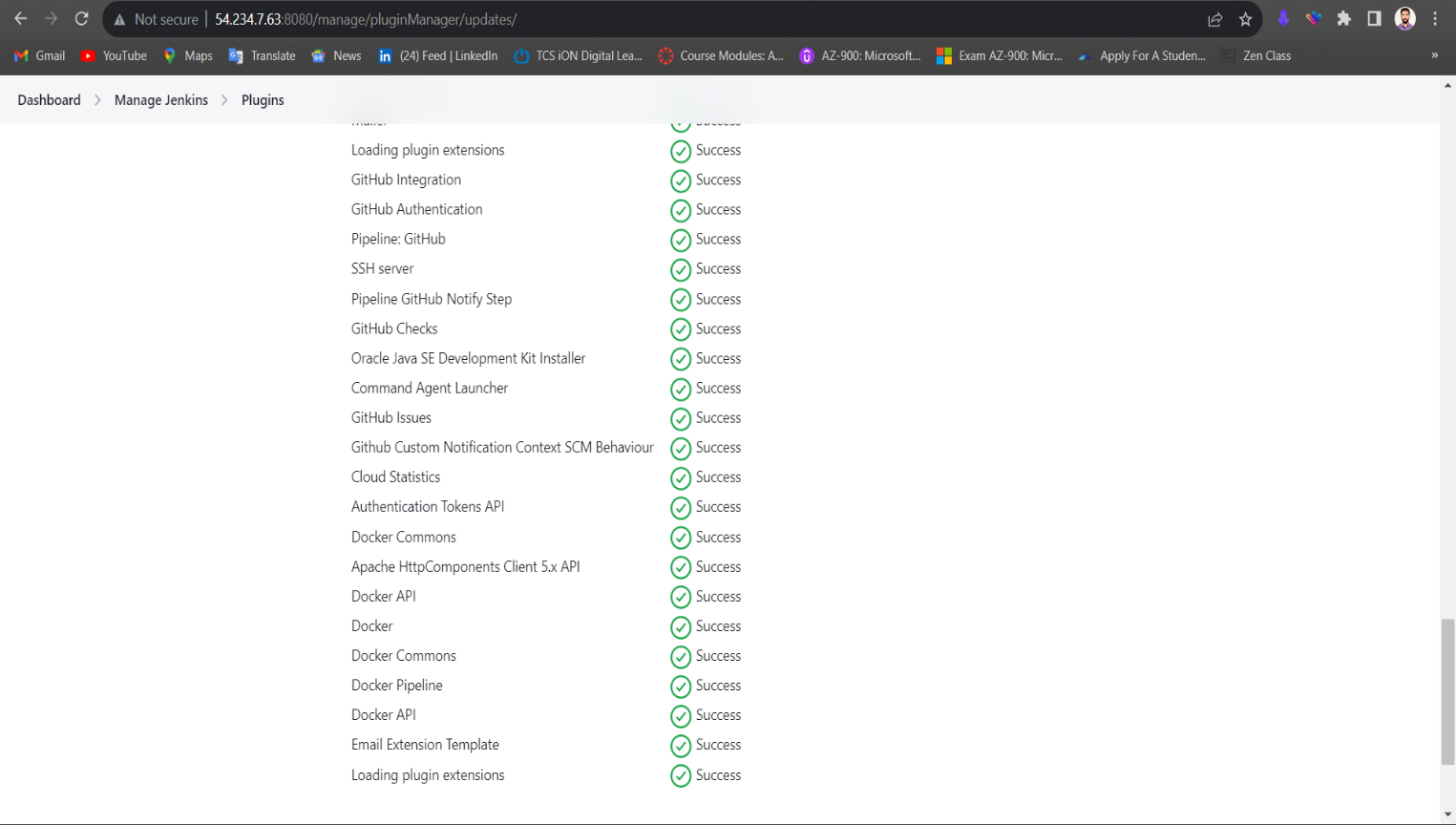
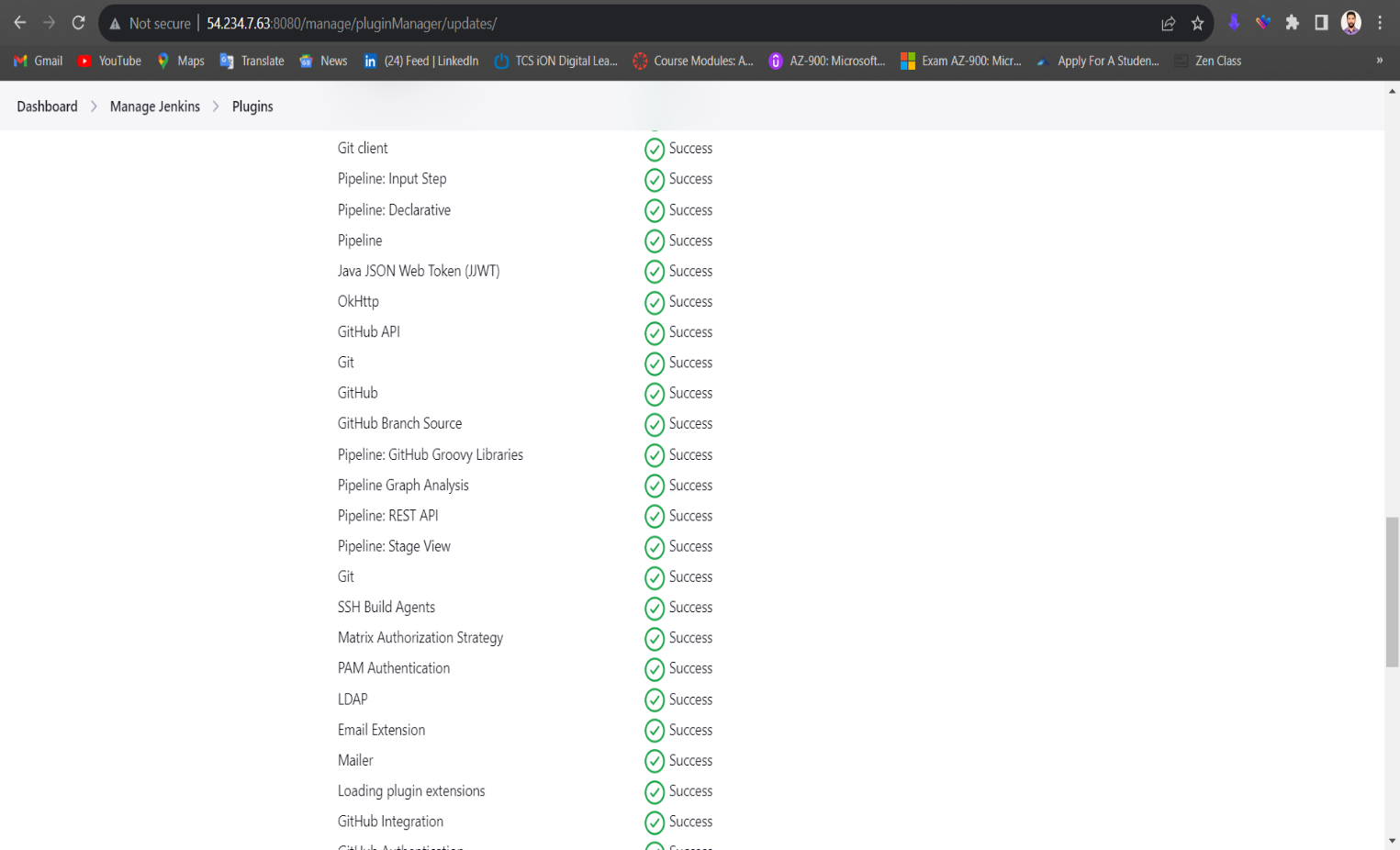
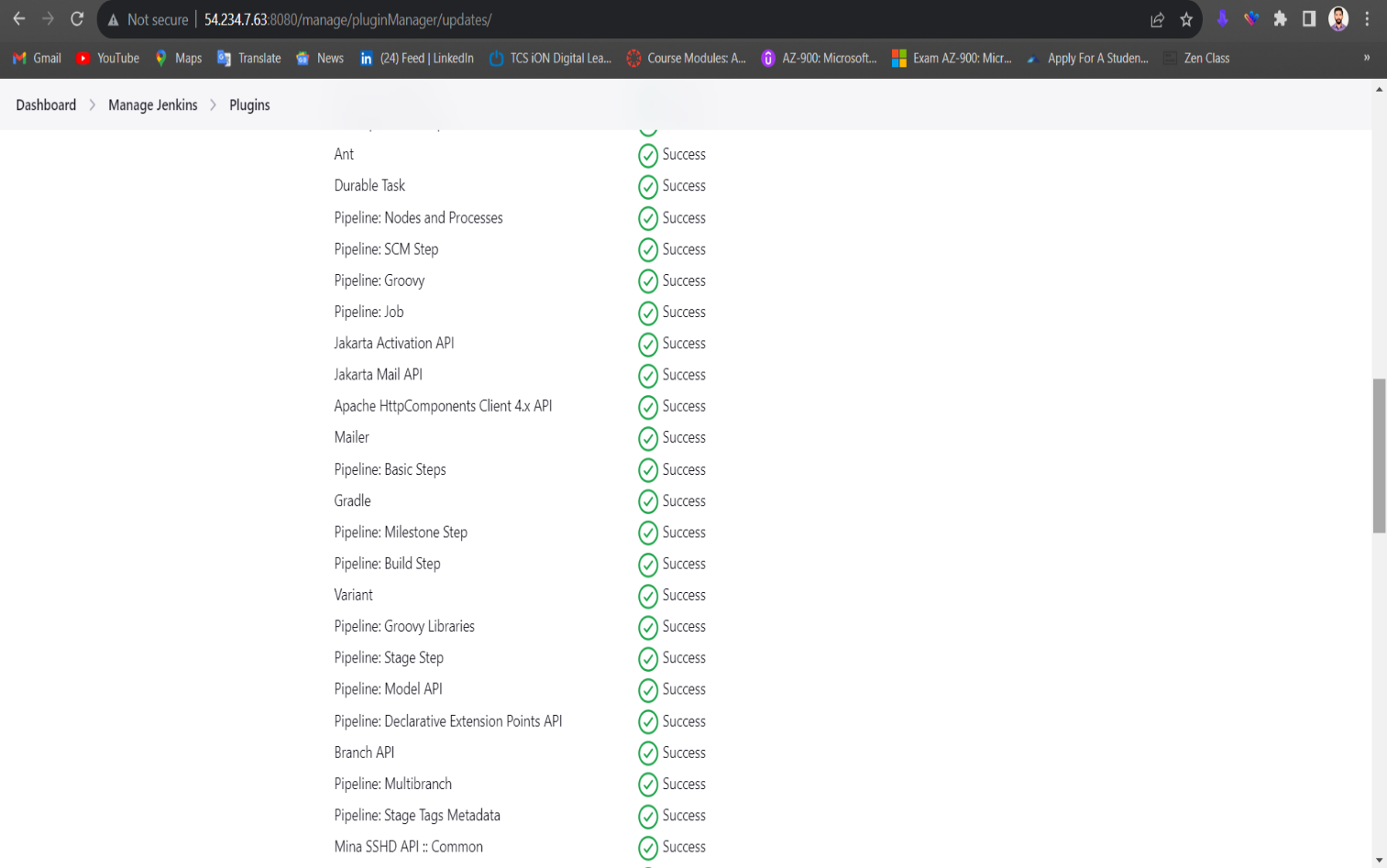
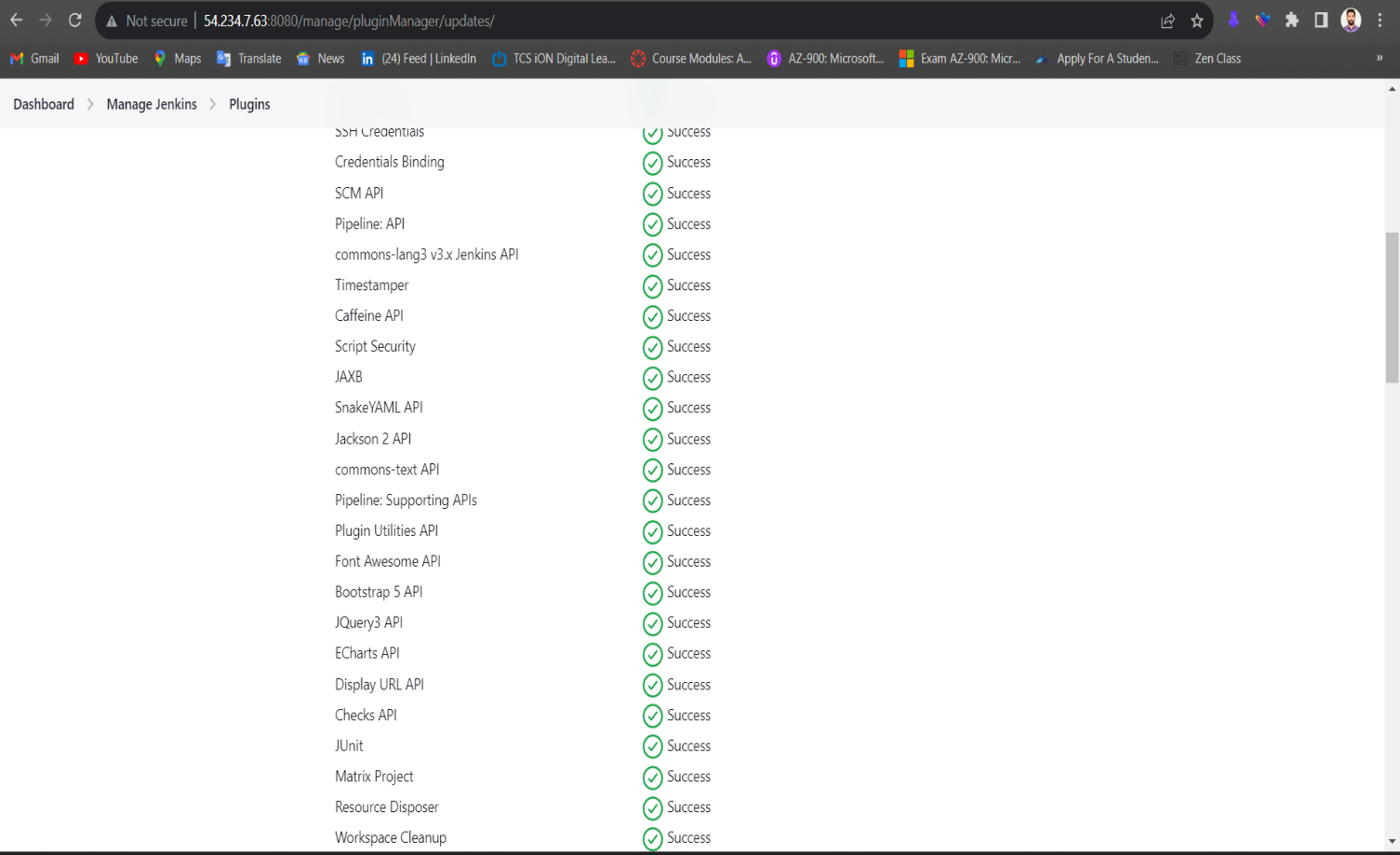
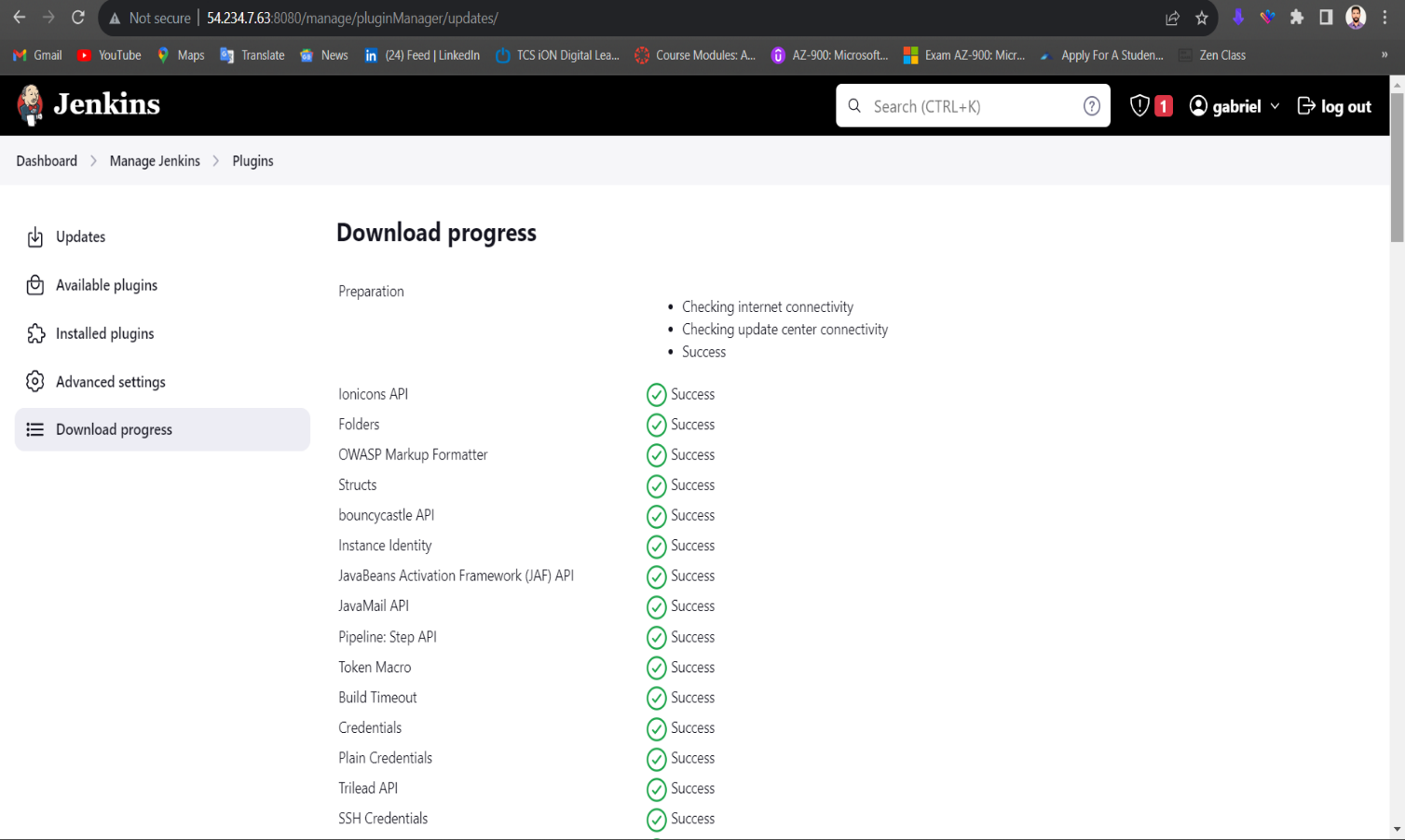
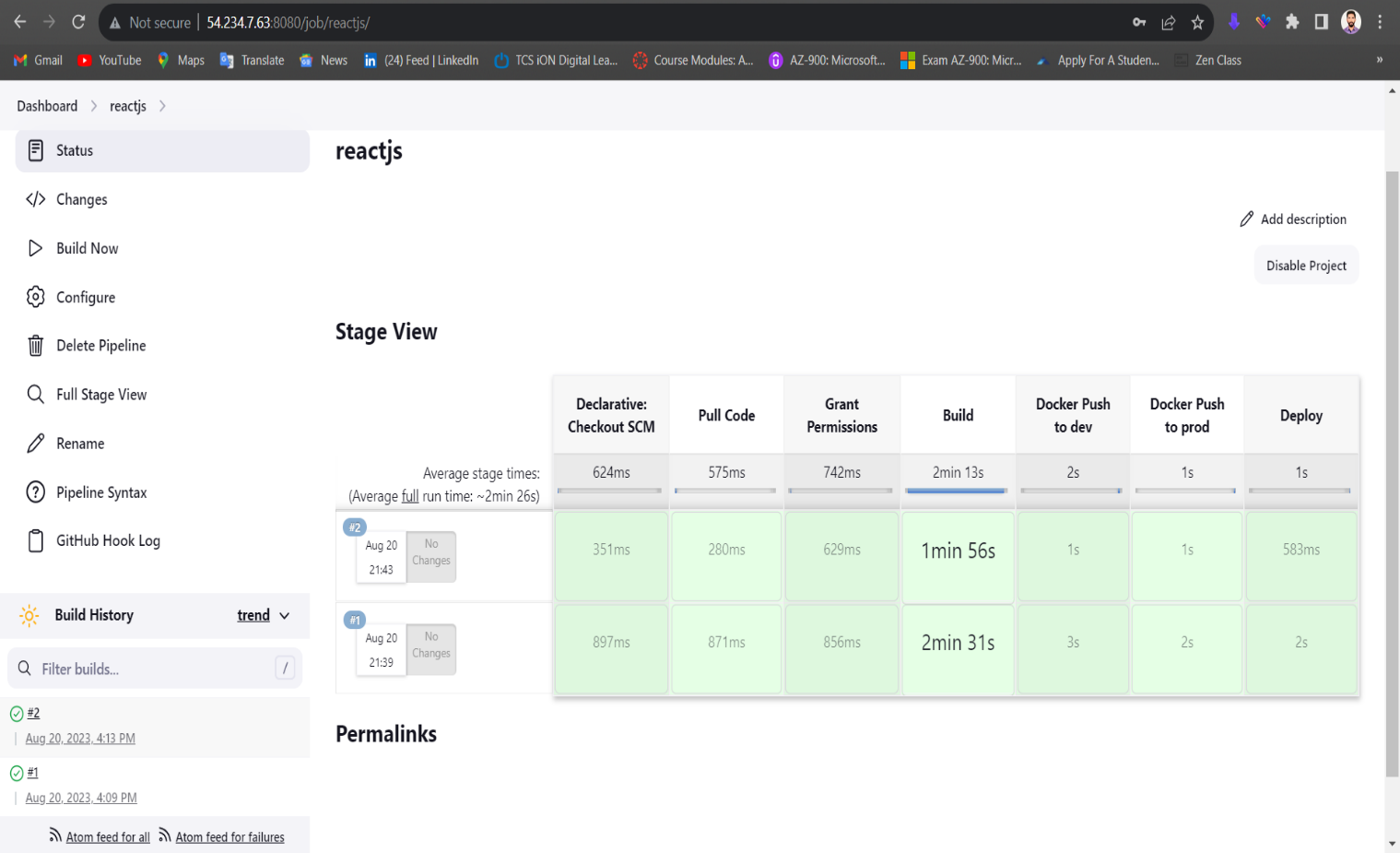
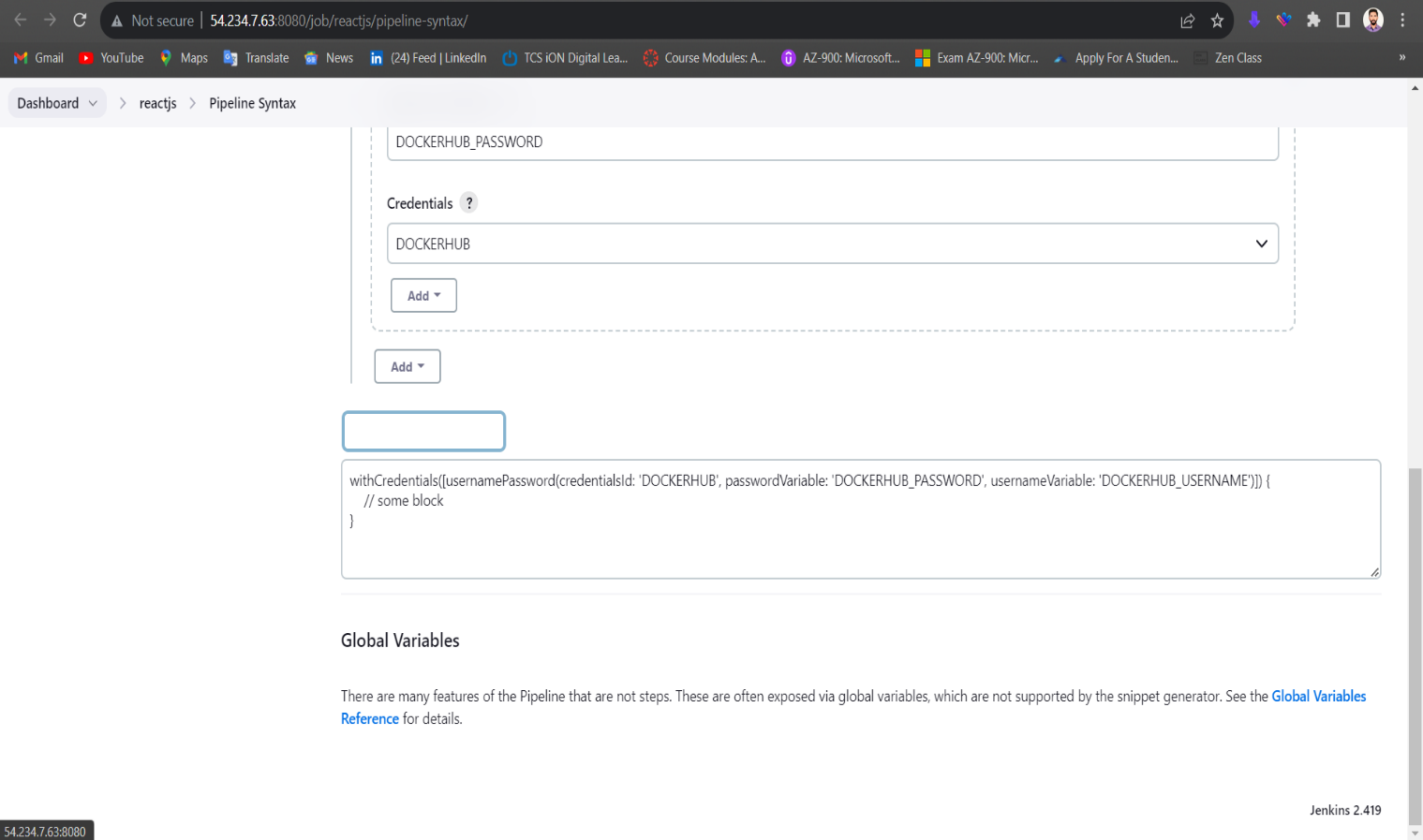
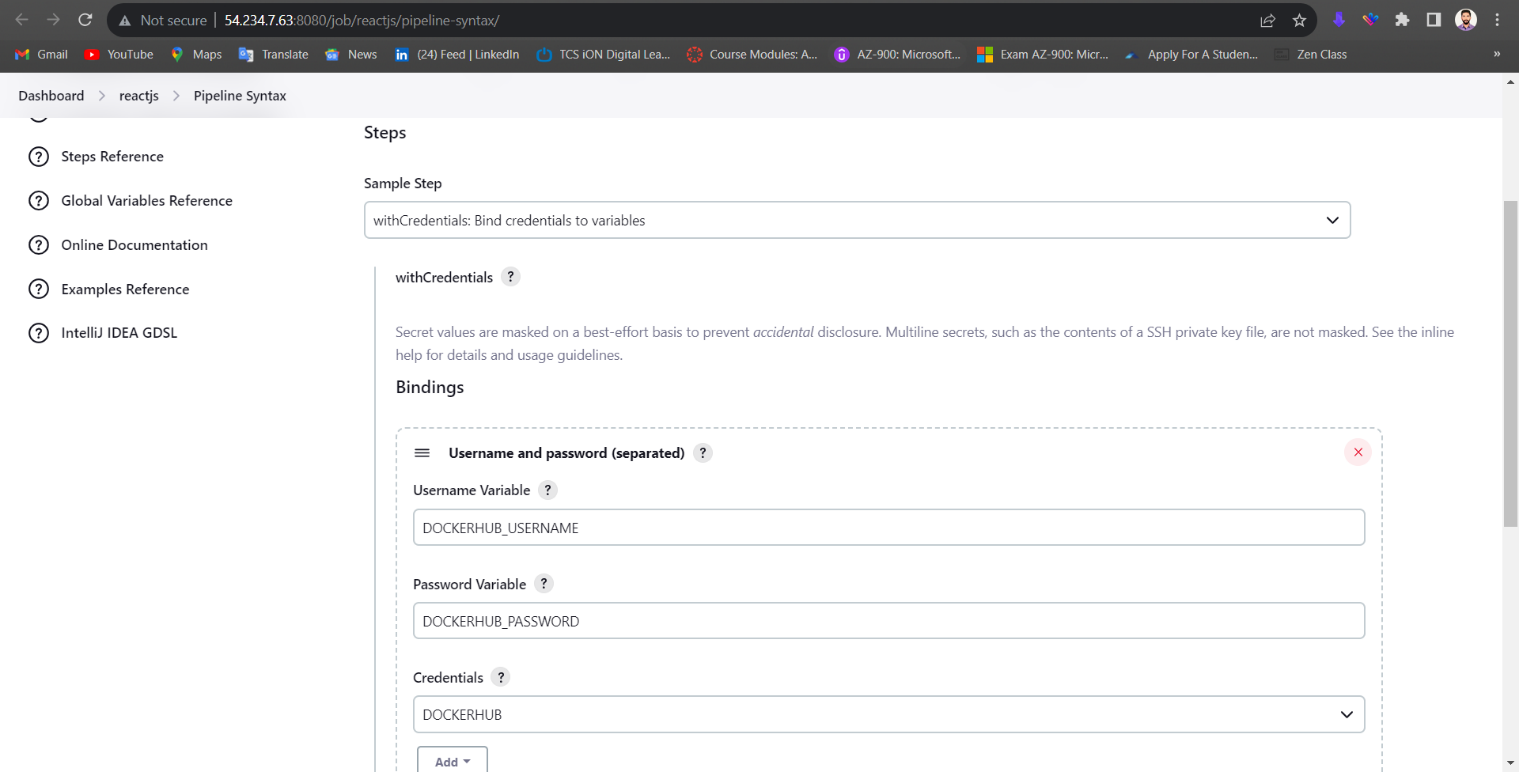
****

****

****

****

****

****