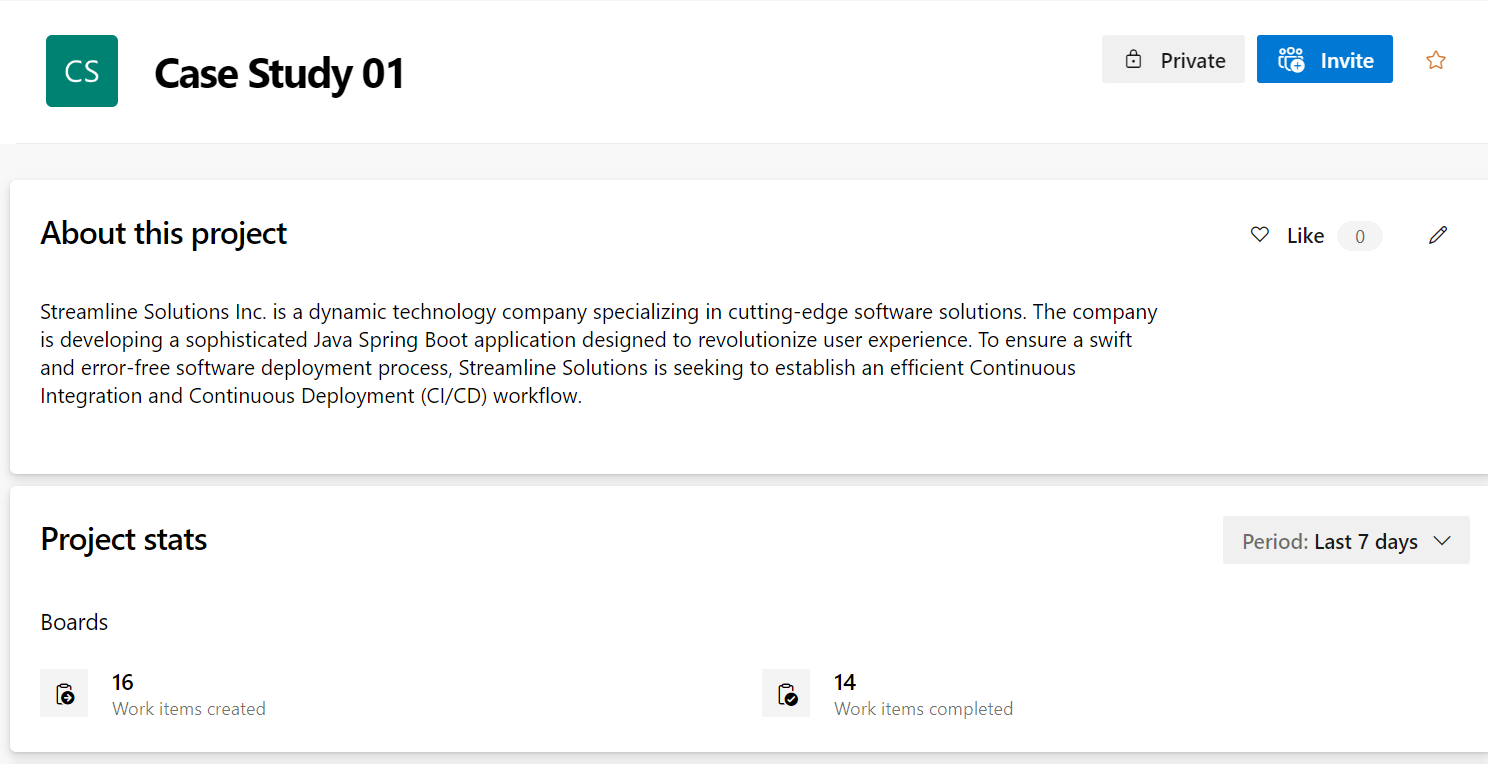
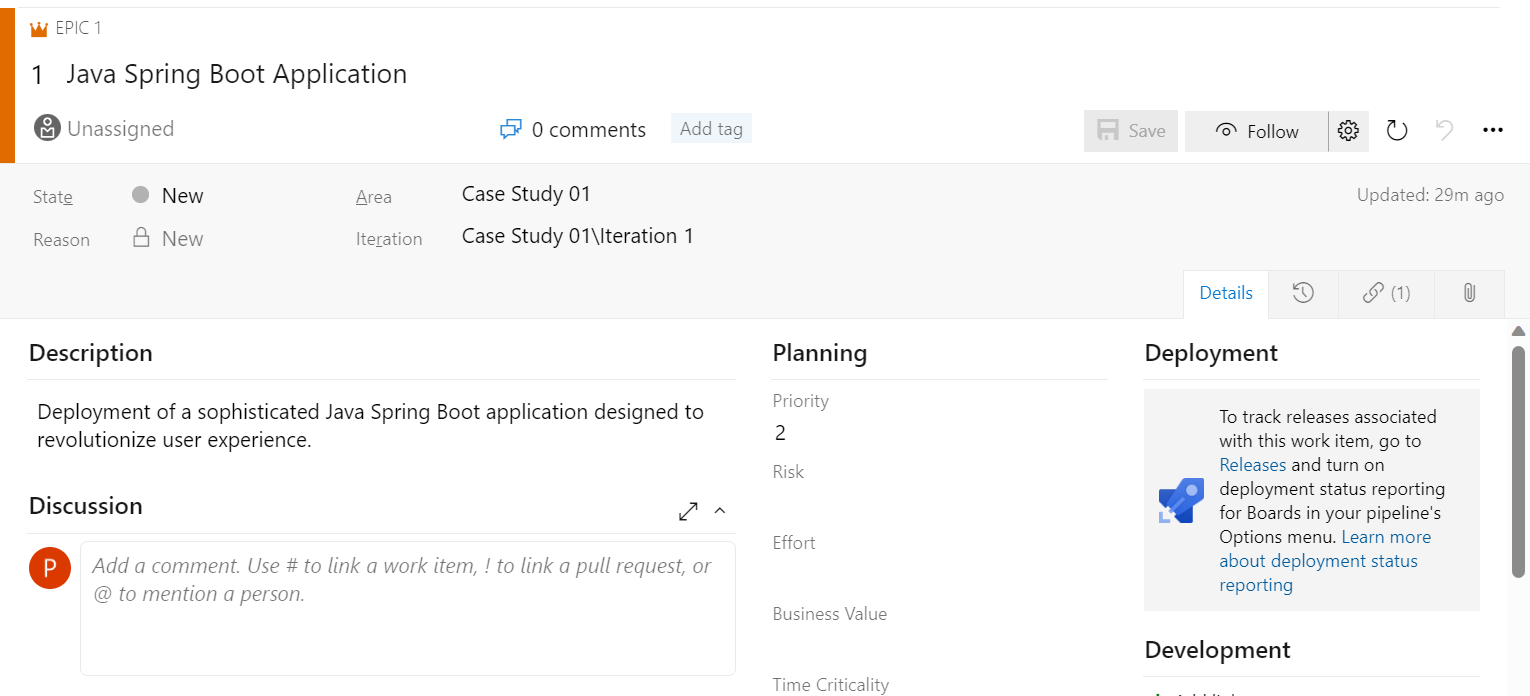
Shell – Final Case Study Assessment

1. Project Management Setup:

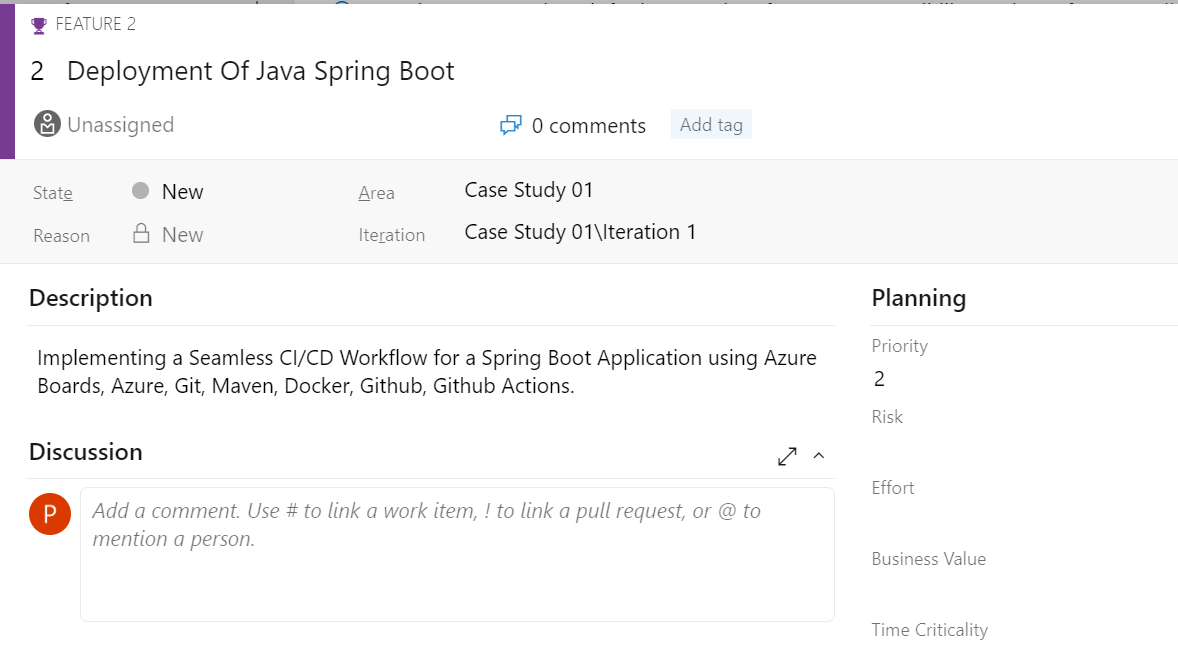
I have created a new project titled Case Study 01 on Azure DevOps. This is a private project using the Agile methodology.



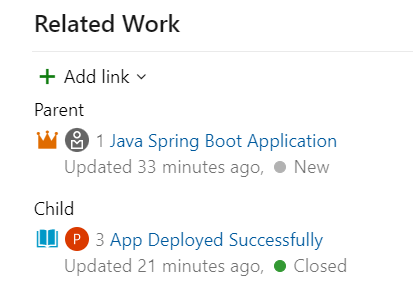
Under this, I have added an Epic for the project, called ‘Java Spring Boot Application’ with the description as ‘Deployment of a sophisticated Java Spring Boot application designed to revolutionize user experience.’ I have set the iteration to iteration 1.



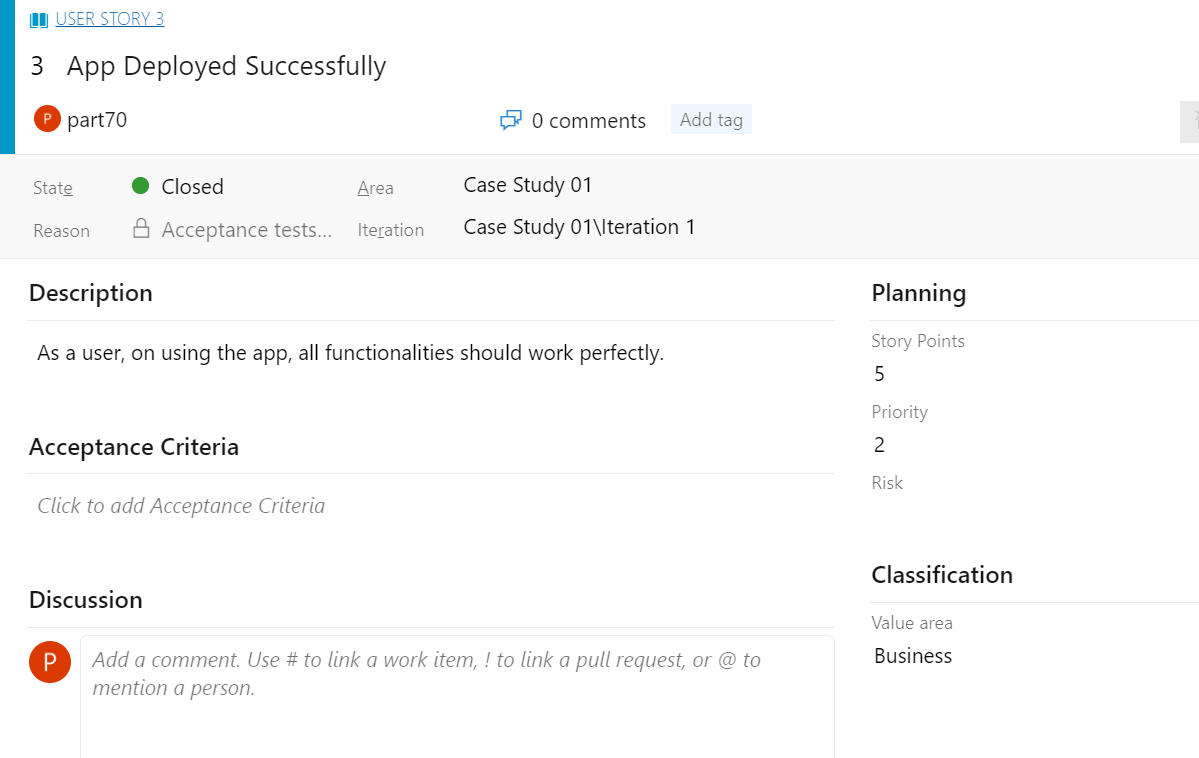
I have added the feature, ‘Deployment of Java Spring Boot’, as a child under the epic.



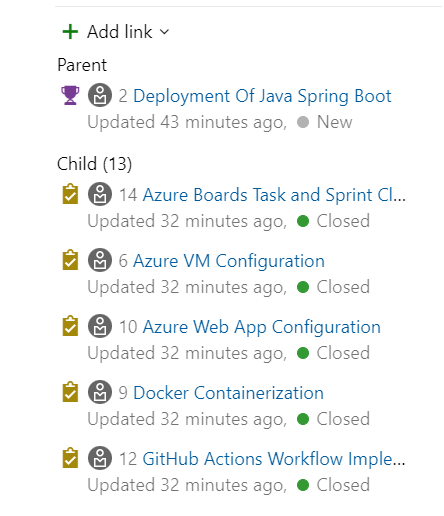
The parent link to the Epic and the child link to the User Story can be seen below.



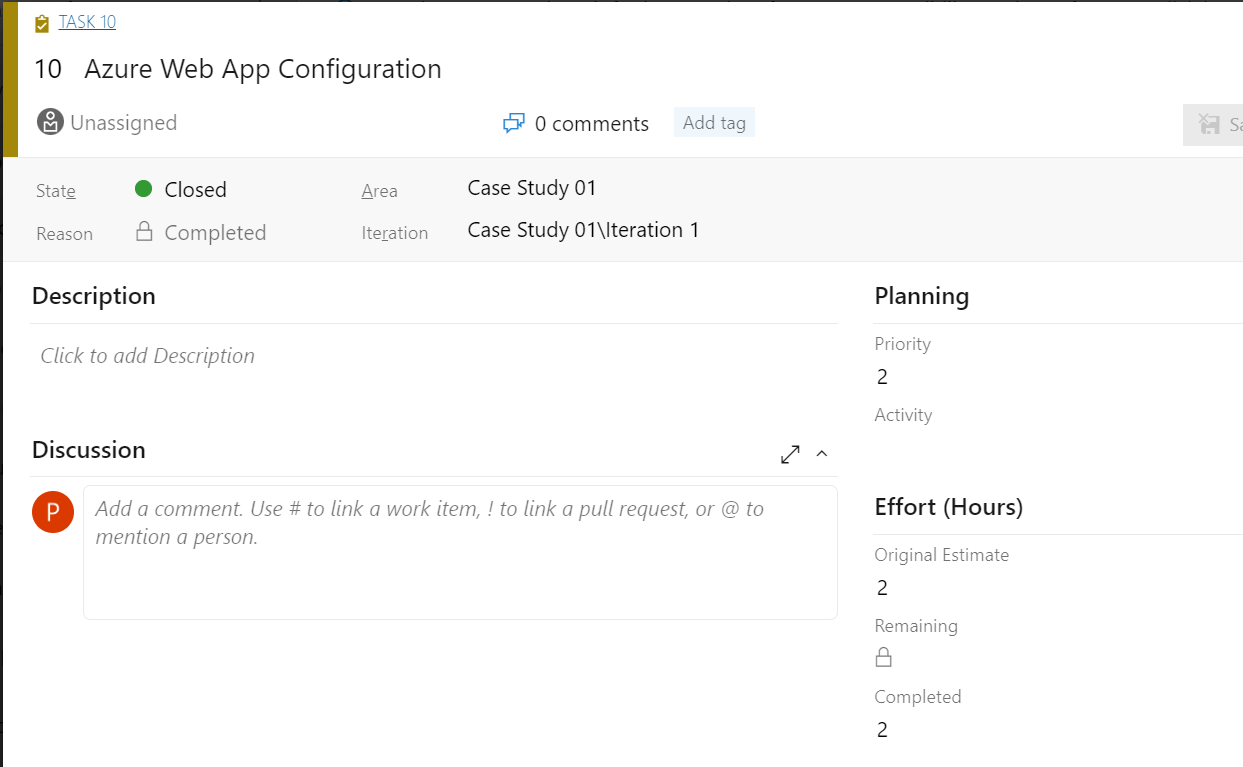
Next, I set the user story and its subtasks, along with their estimated hours and story points.



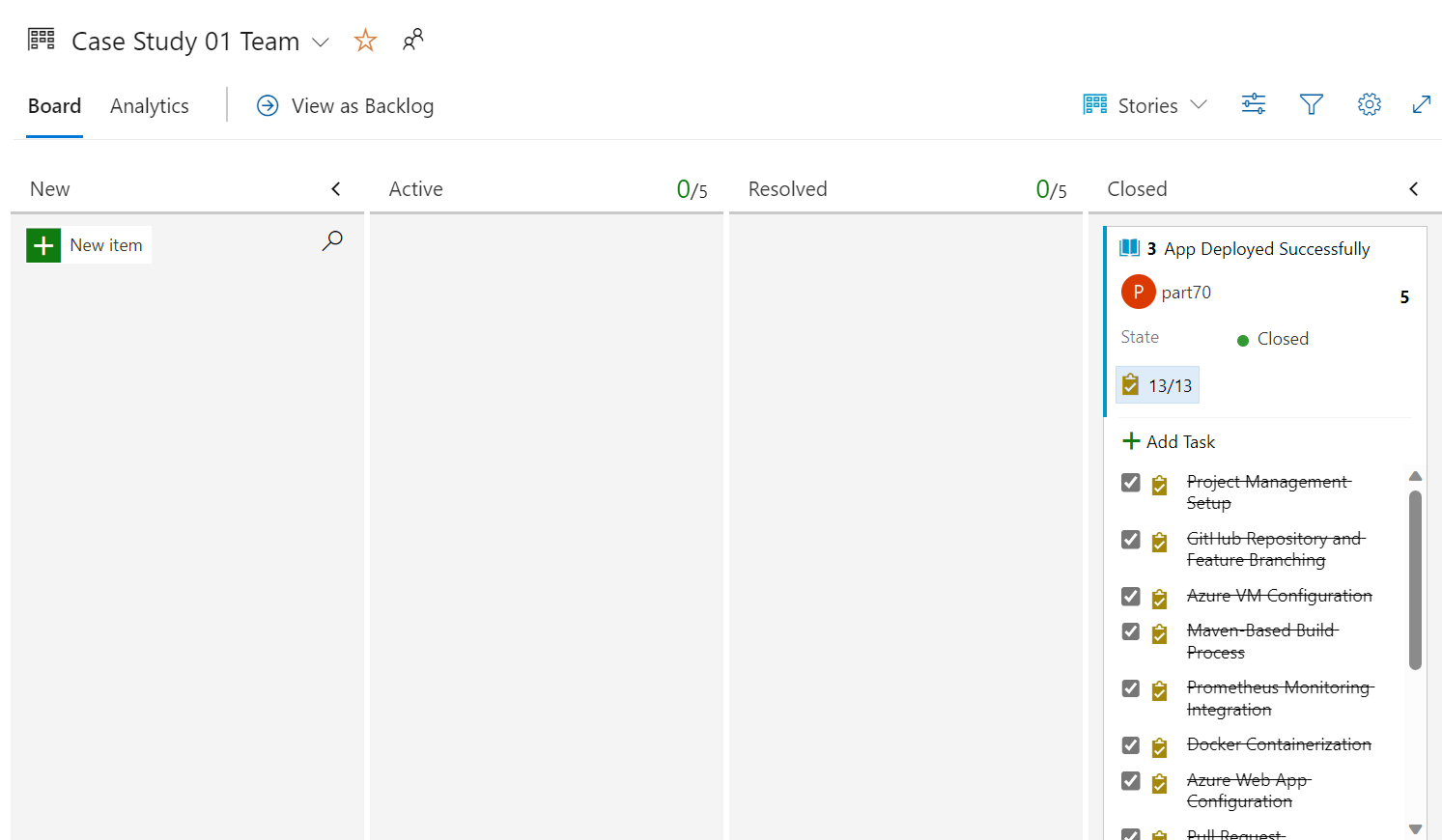
The links to the subtasks can be seen in the below attached screenshot.



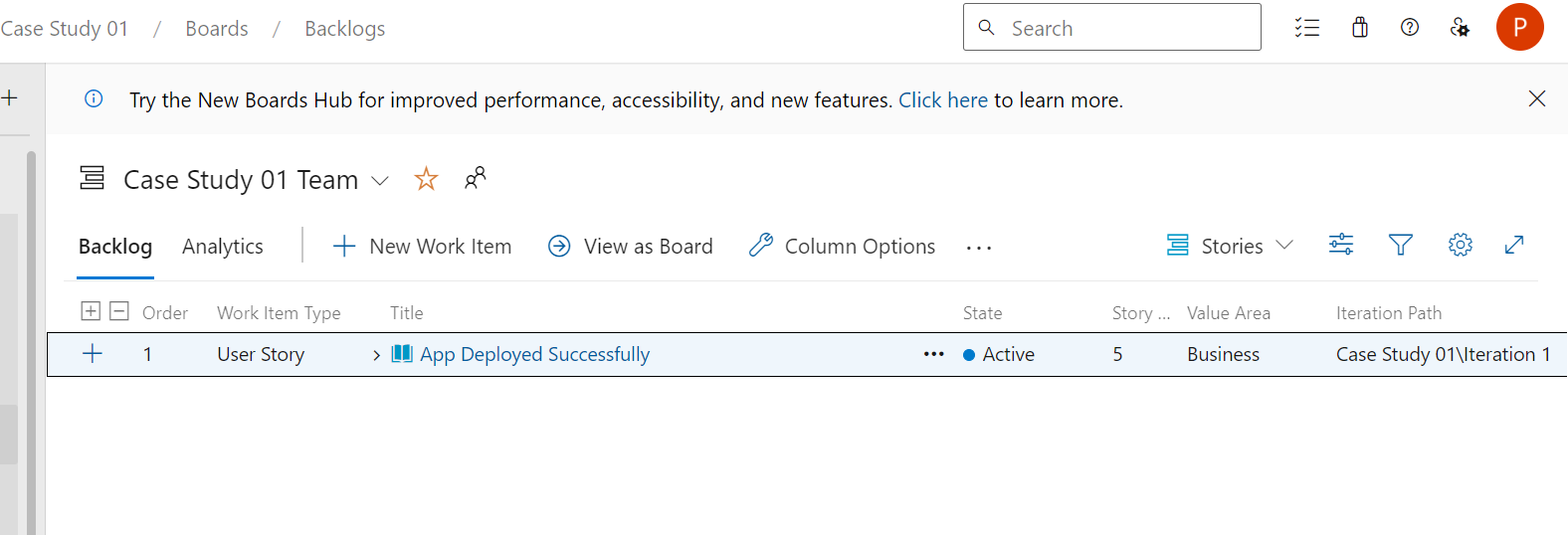
An example of the subtask can be seen below,



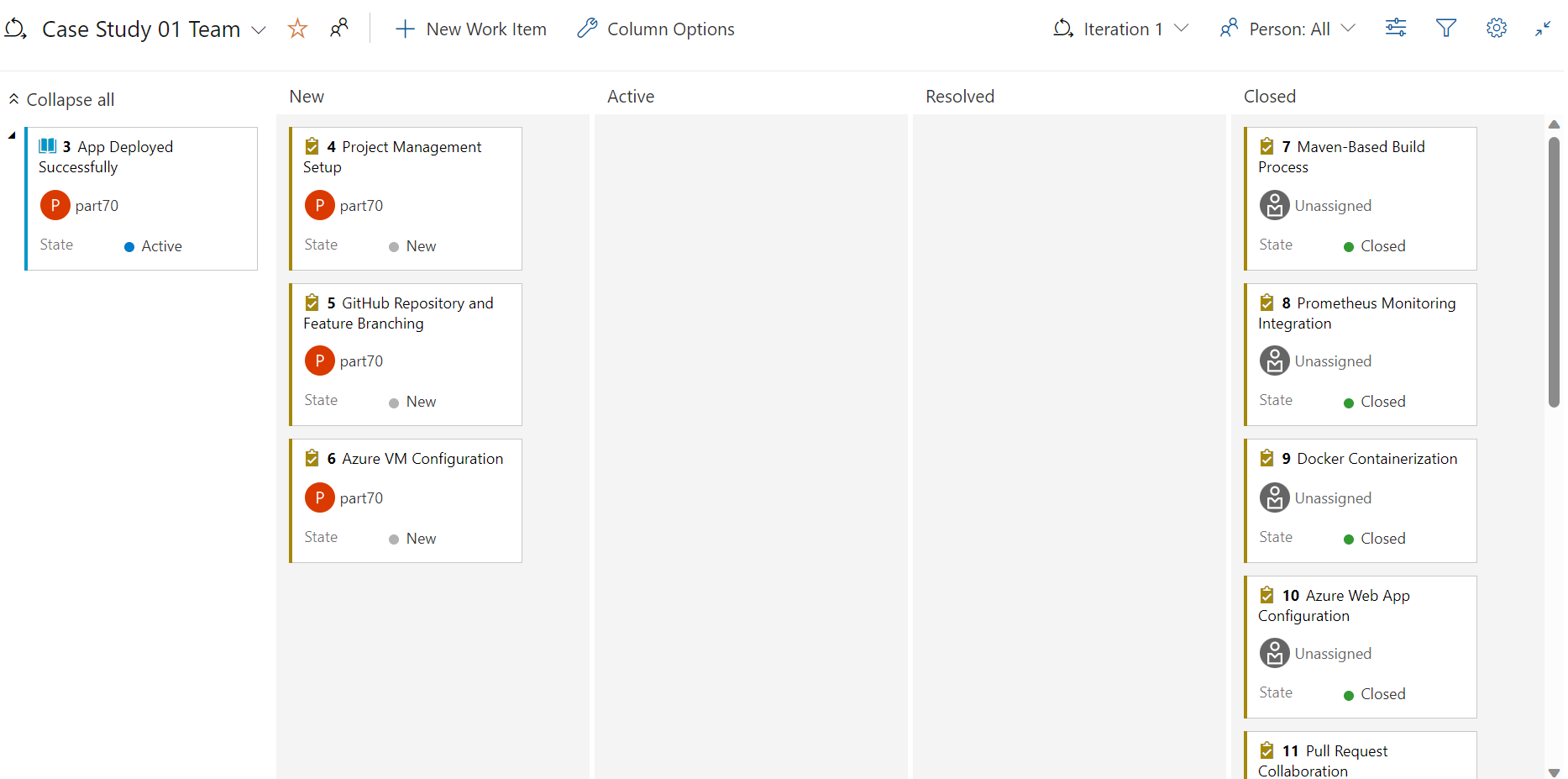
1. Project Board:



1. Project Backlog:

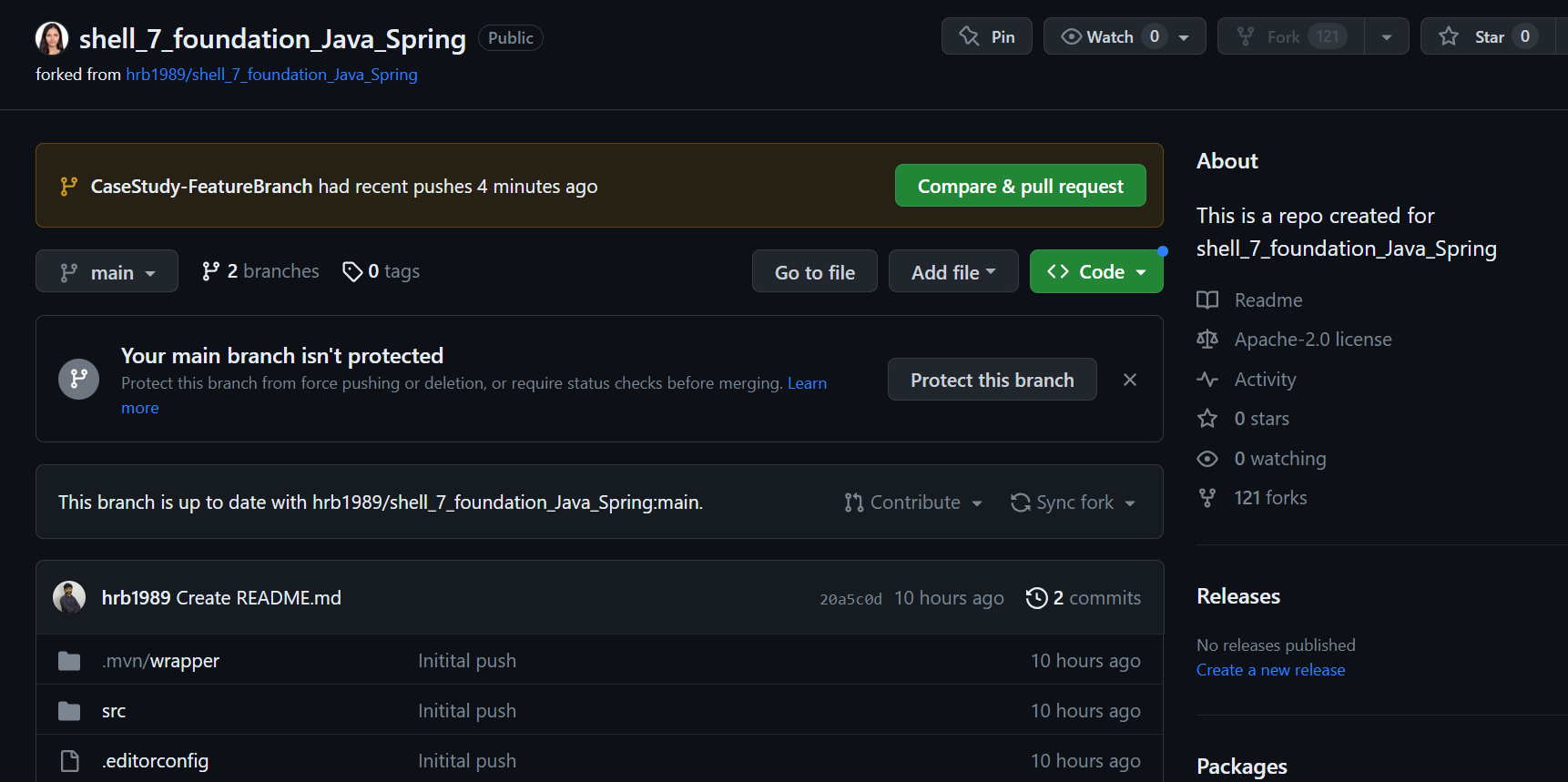


1. Task Board:

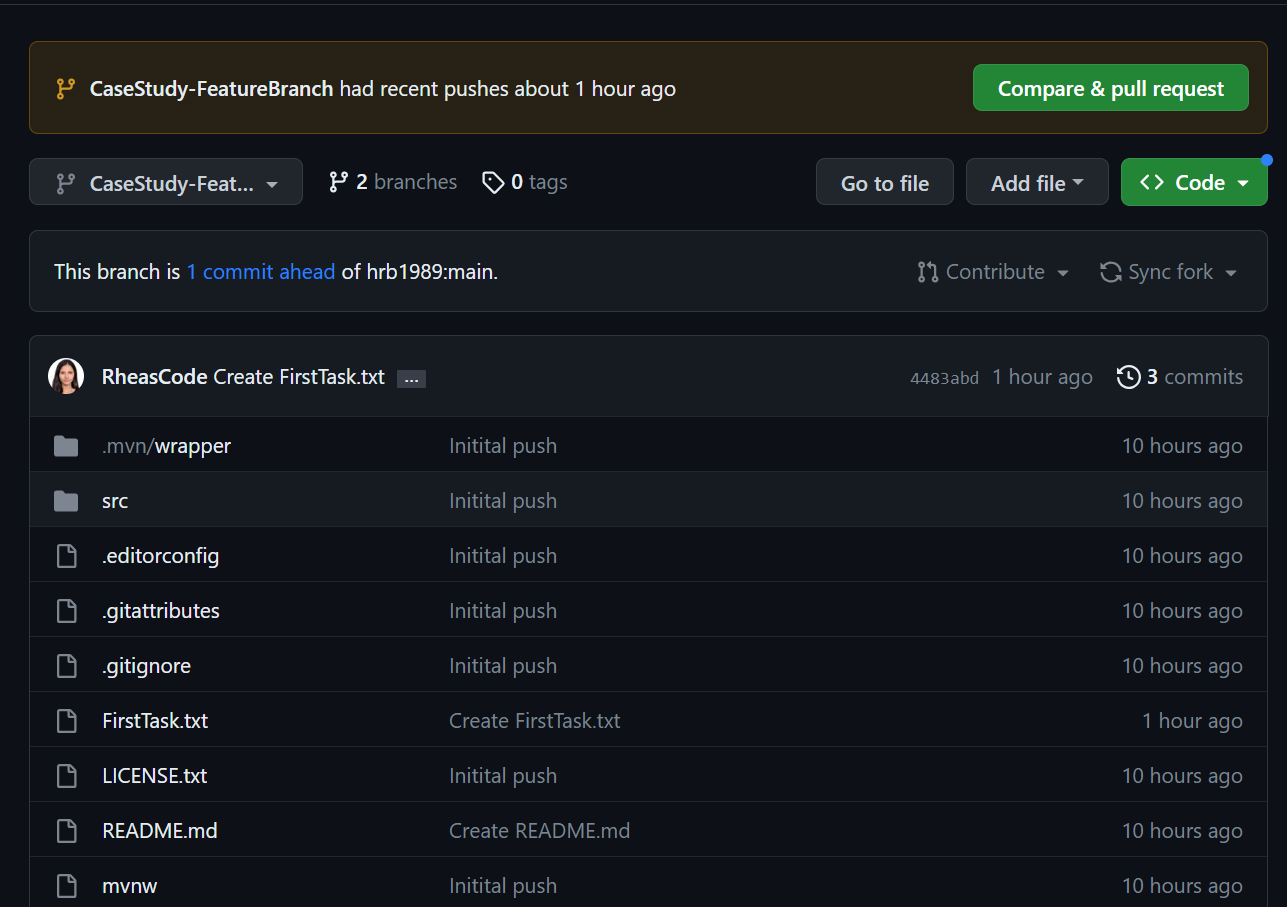


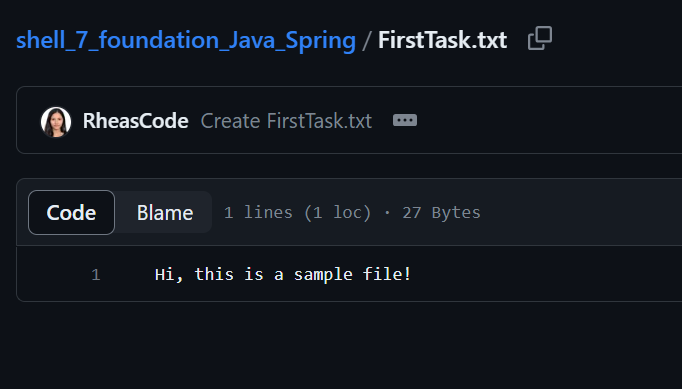
1. GitHub Repository and Feature Branching:

2.1 Forking the repository:

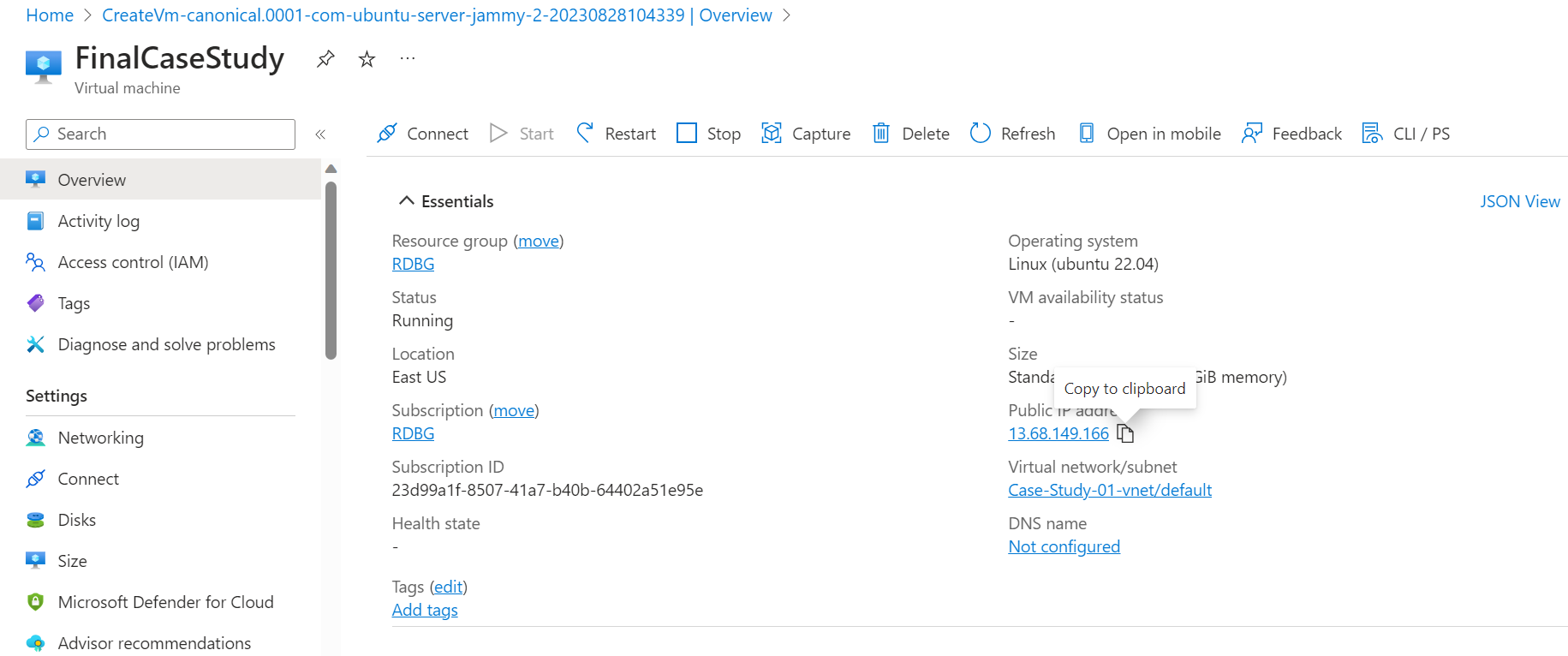


2.2 Creating Feature Branch:

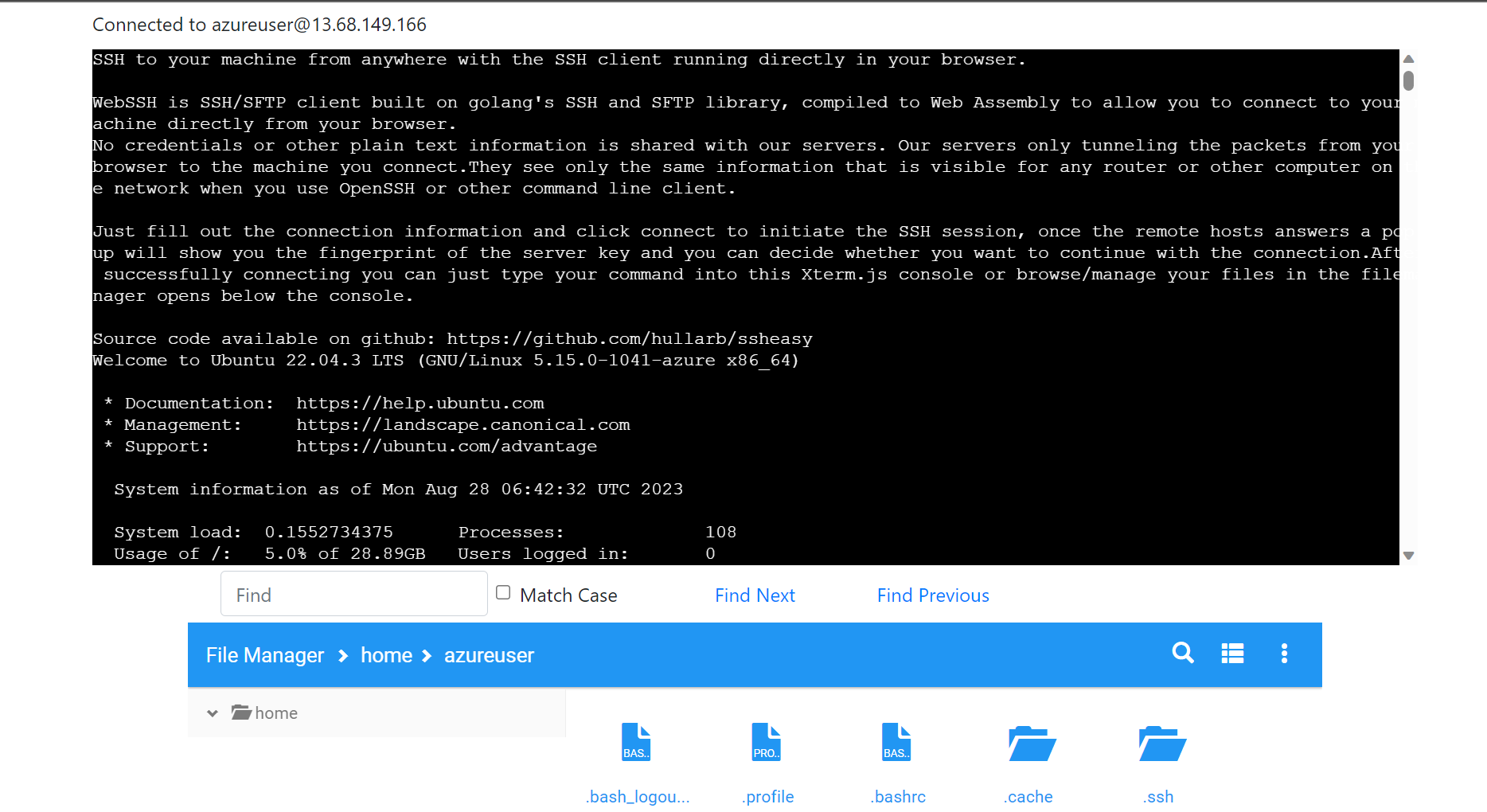
x



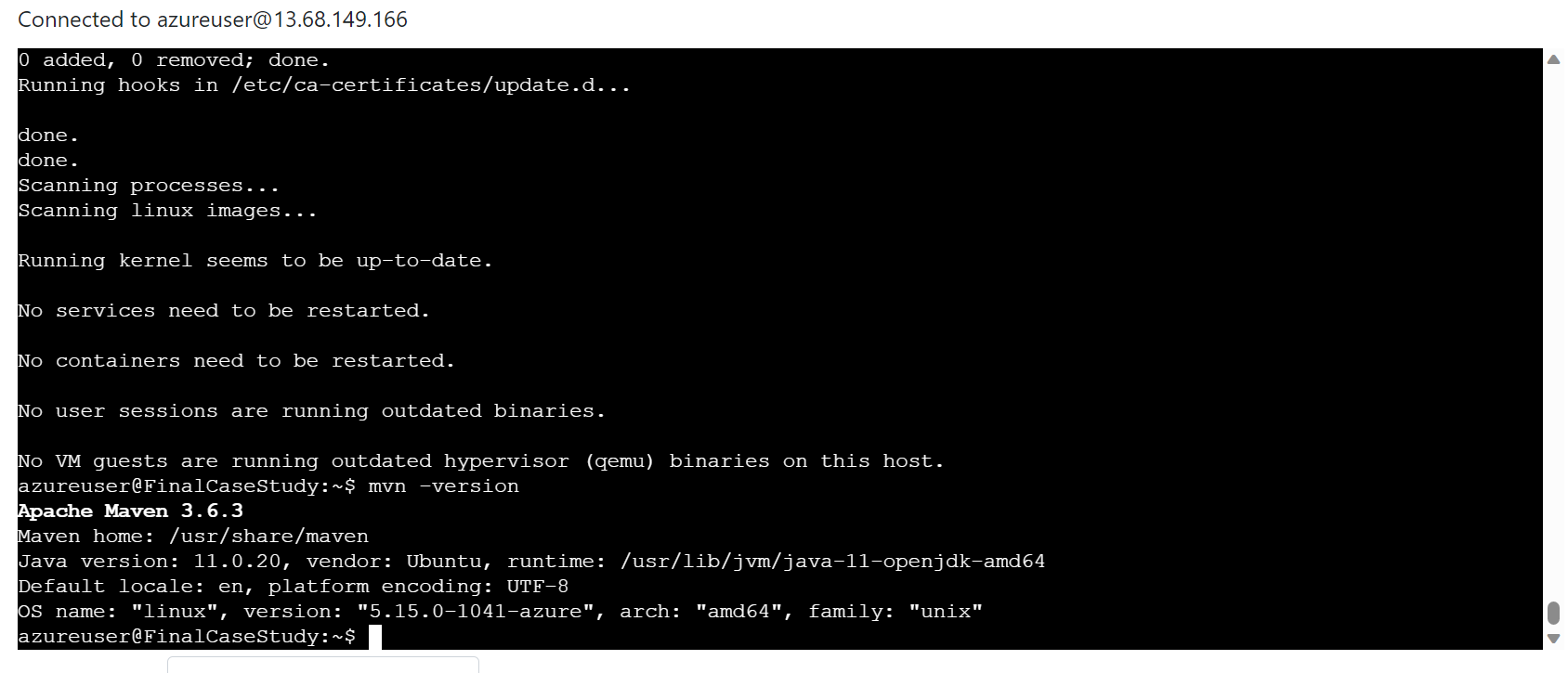
* 1. Connected to VM



3.2 Now, I have connected the VM to ssheasy.com:



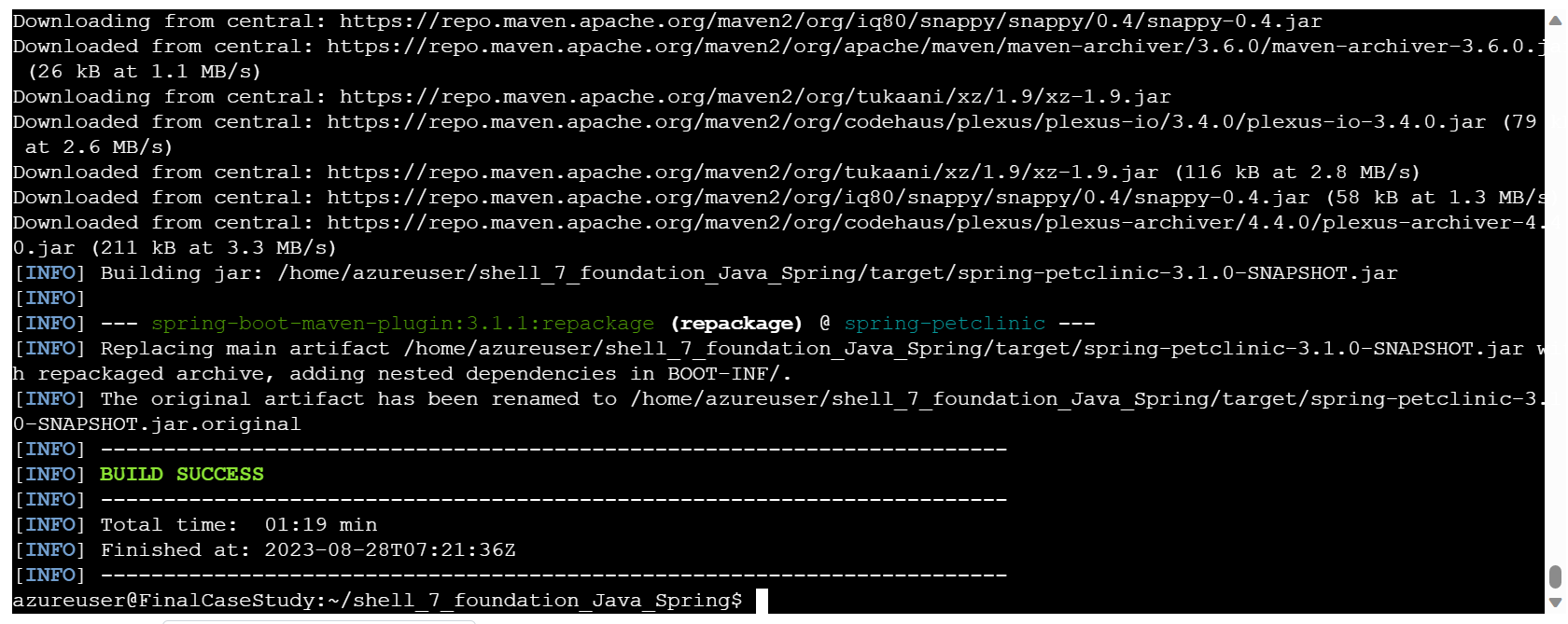
* 1. Installed Maven on the VM:



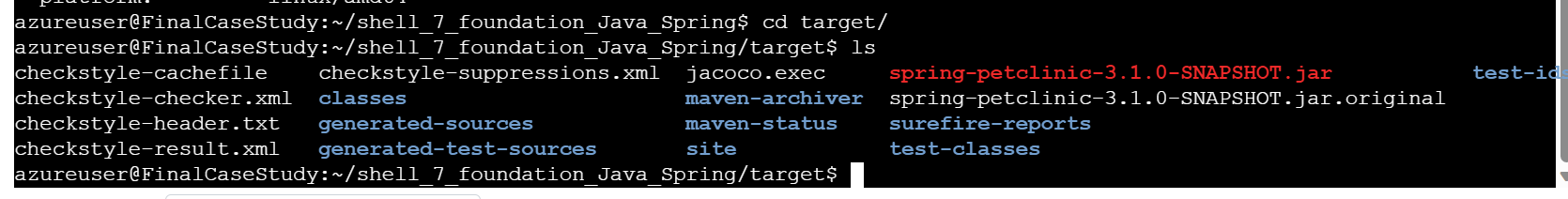
3.3 Updated the Java version to Java 17:



3.4 App built successfully using Maven:



3.5 Jar file from Java Spring Boots codebase can be seen below:

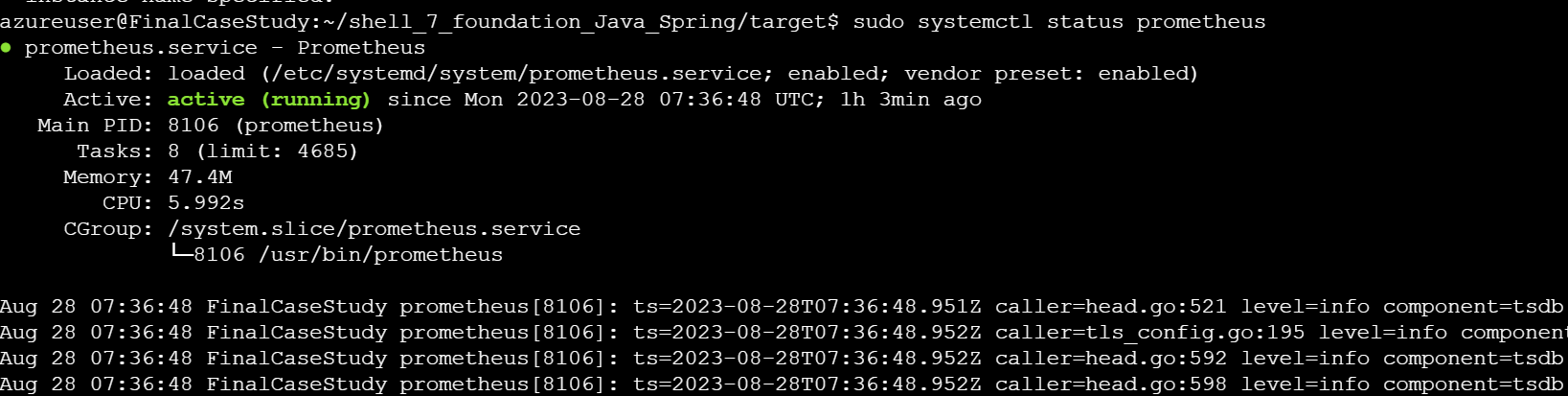


1. Prometheus Monitoring Integration:

4.1 Installed Prometheus successfully:

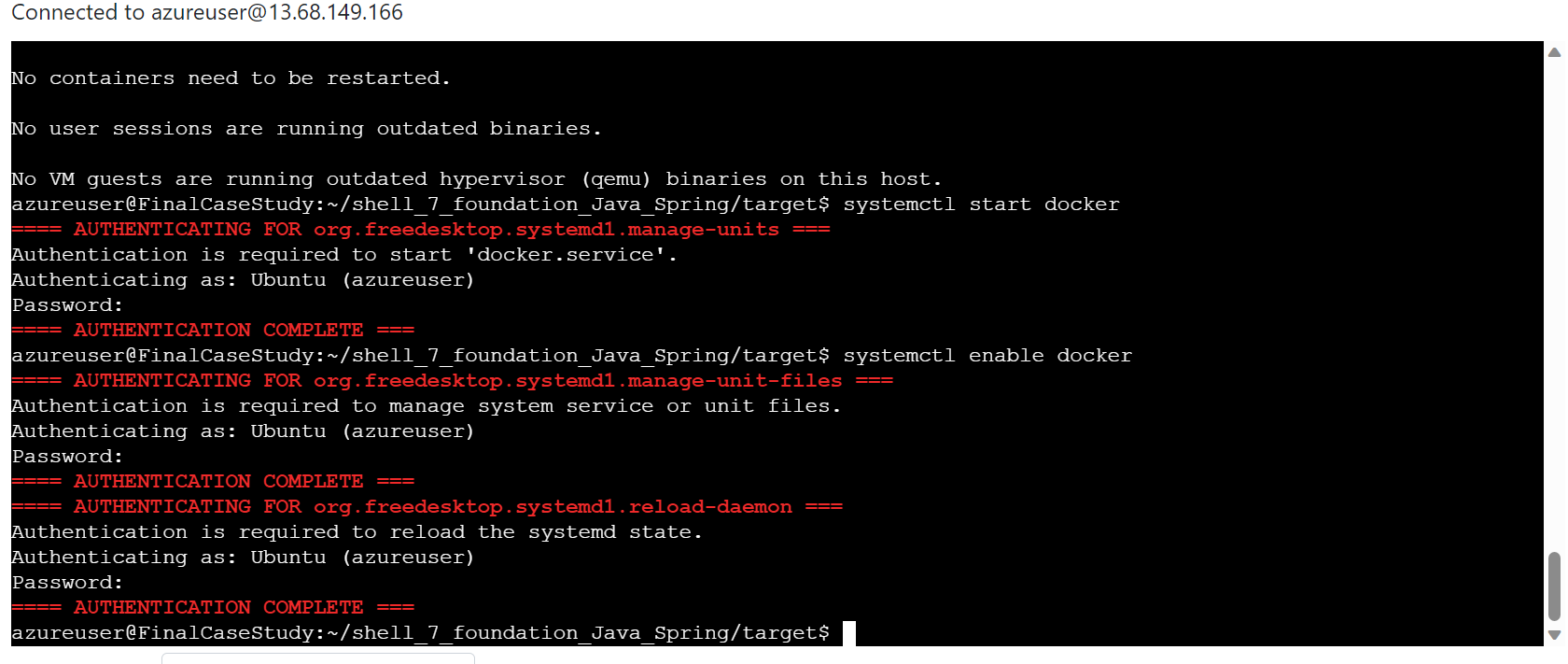


4.2 Configure Prometheus to monitor the system:

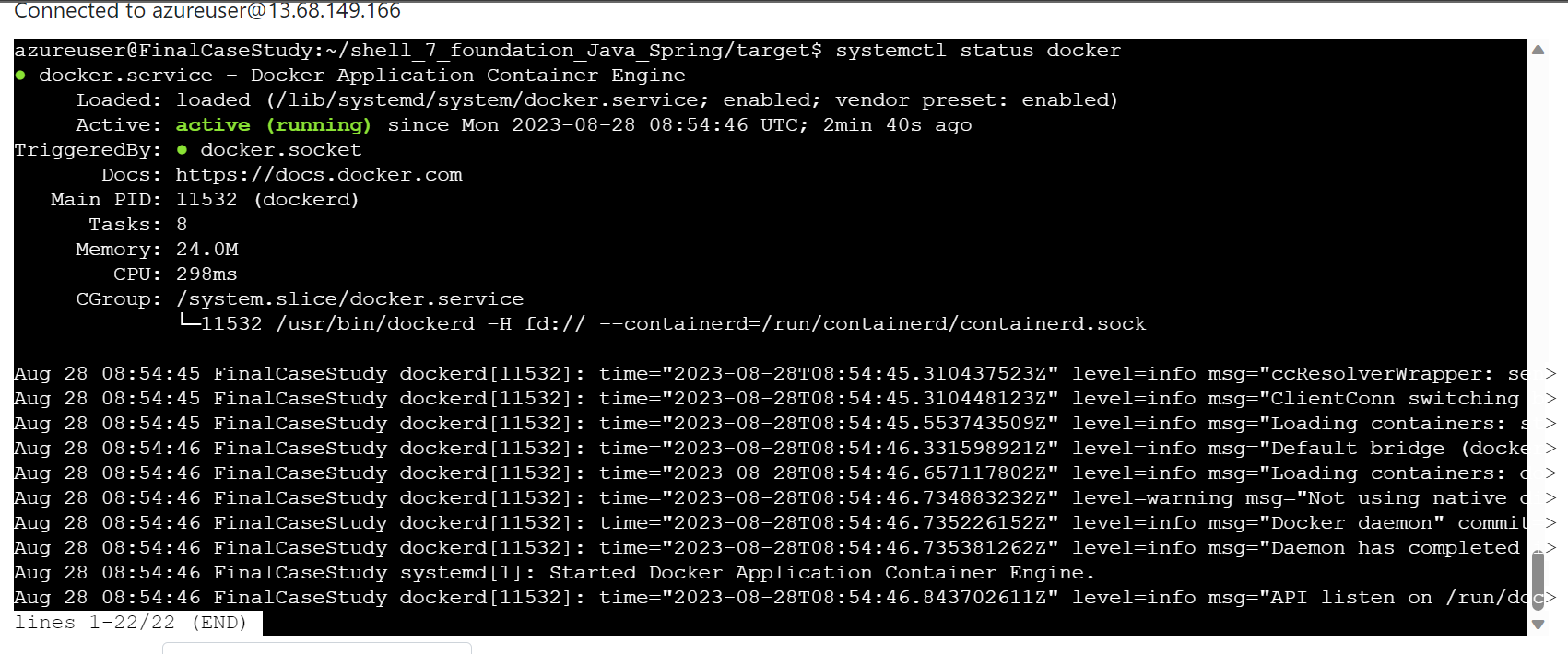


1. Install Docker and run Dockerfile:

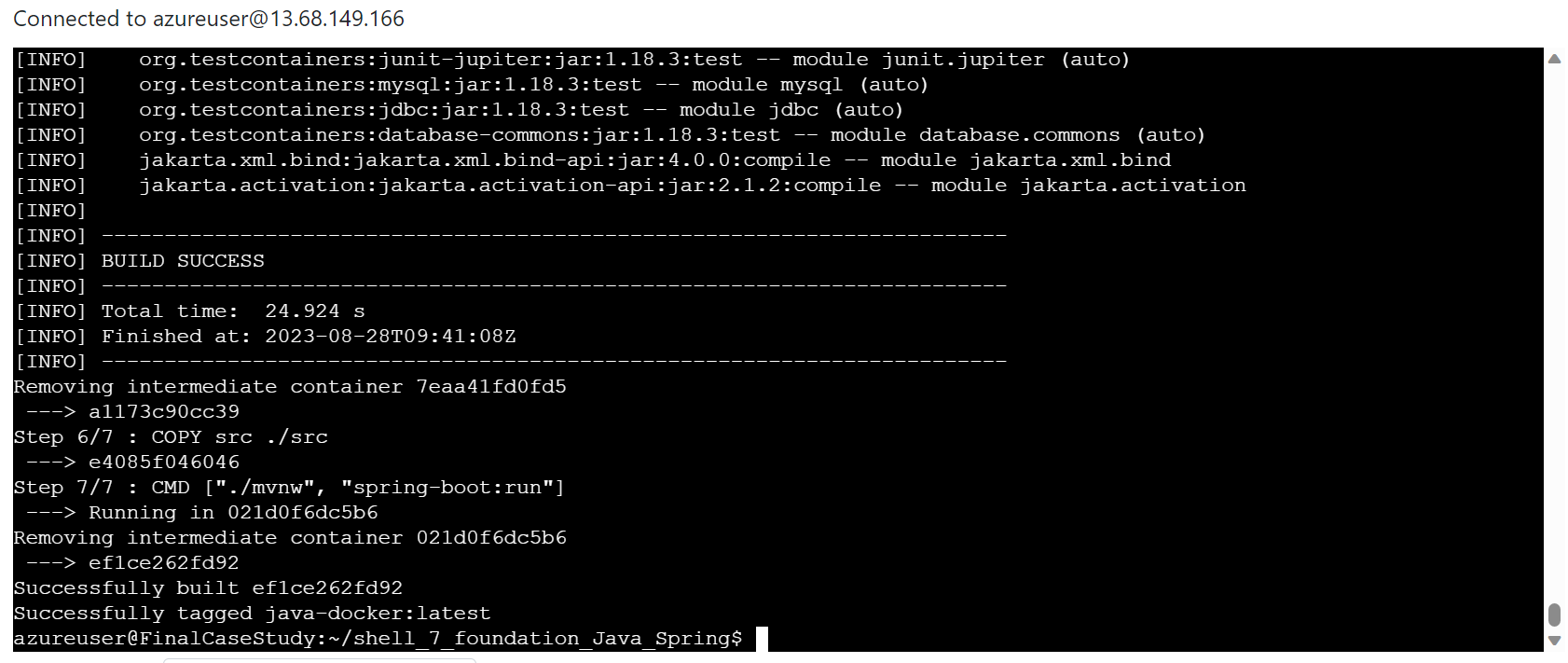
5.1 Installation of docker complete



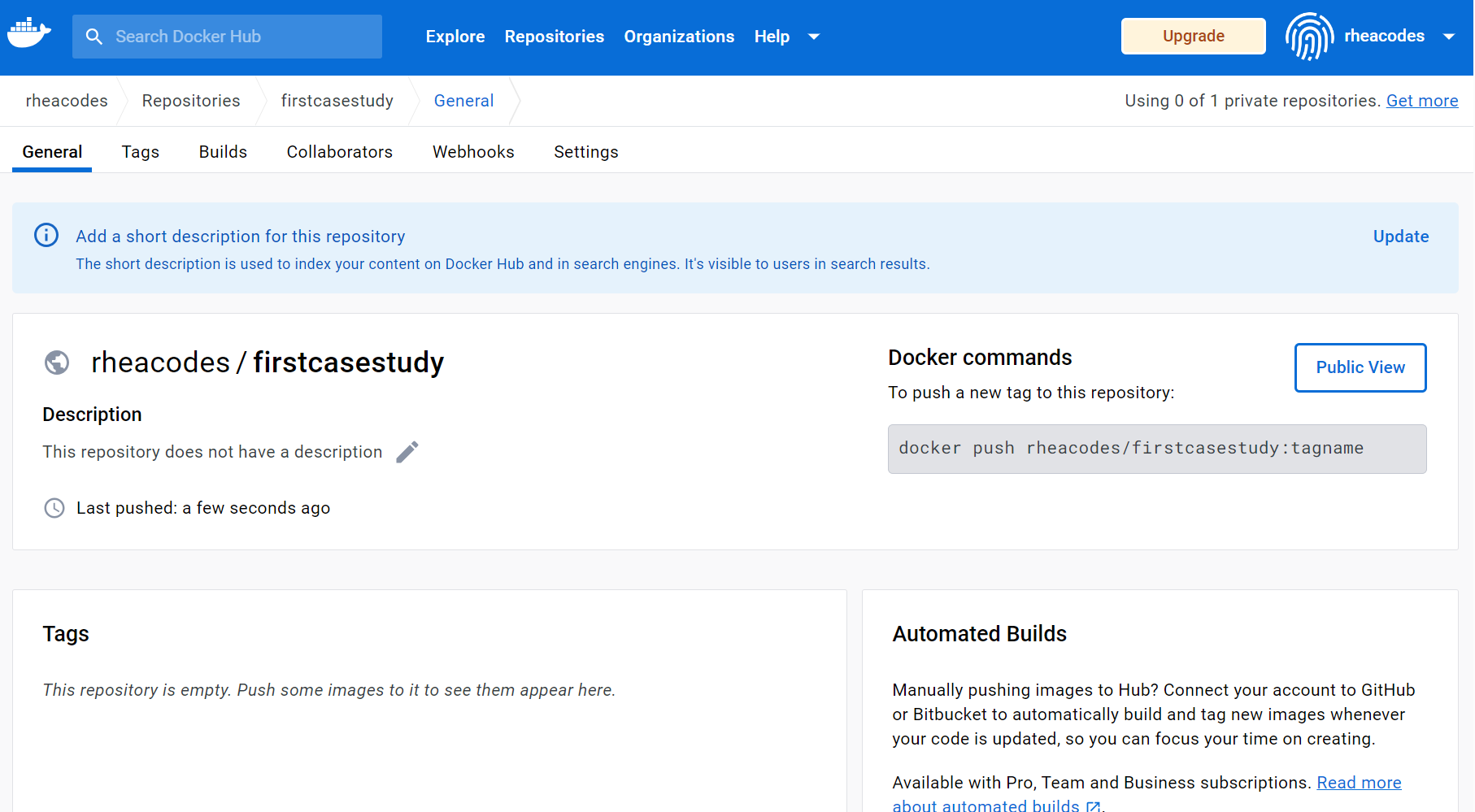
5.2 Docker Active



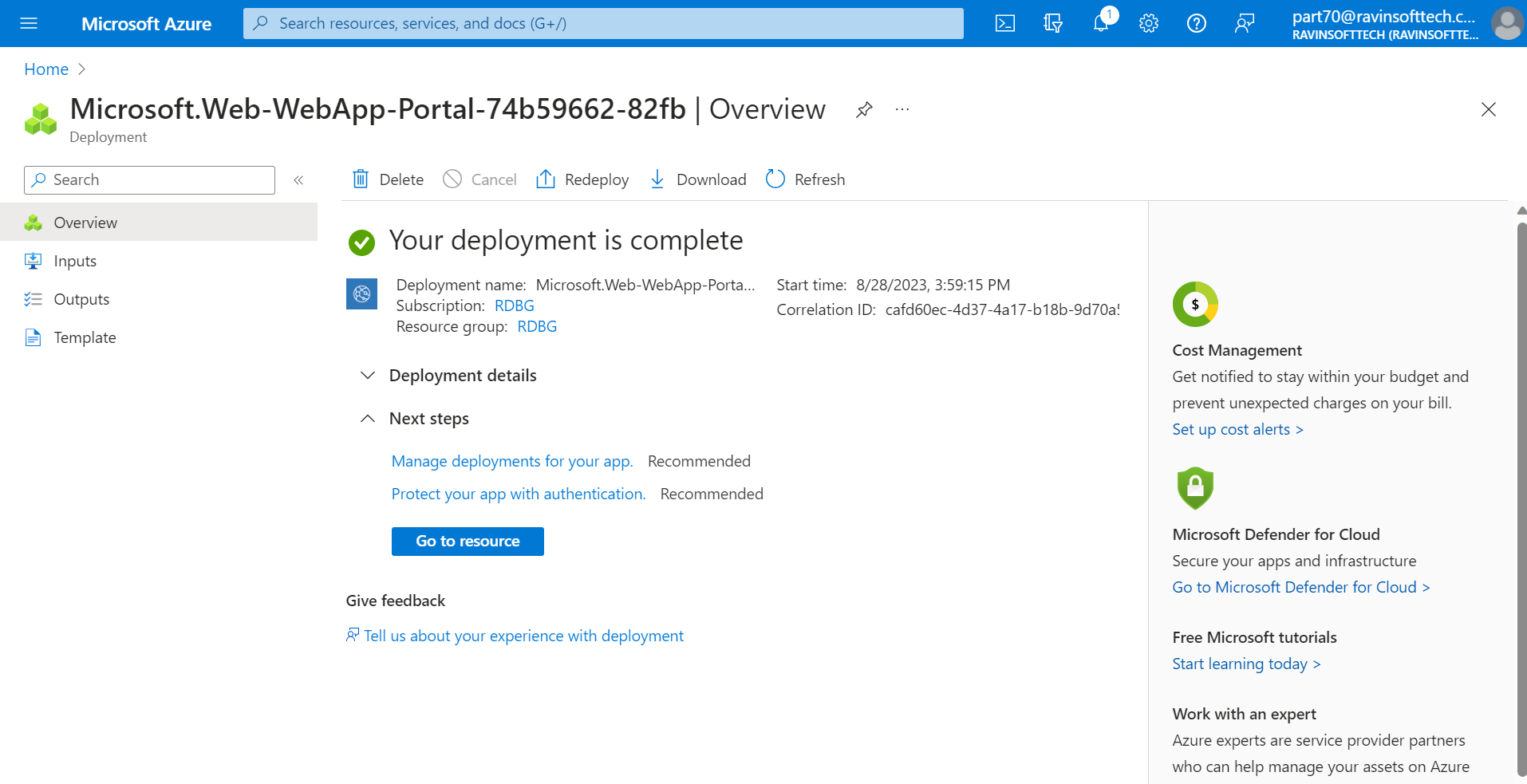
5.3 Dockerfile created:



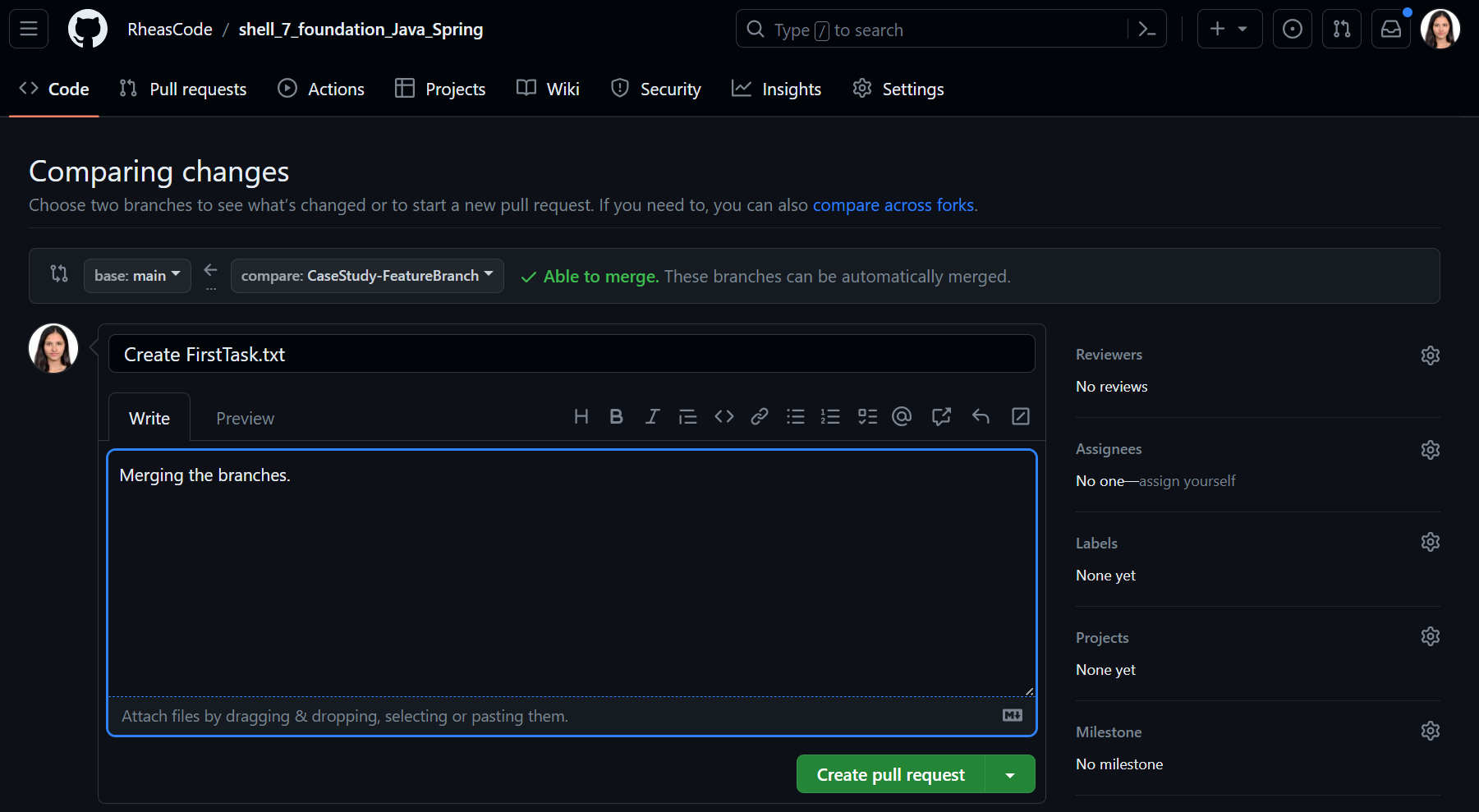
5.4 Docker Image pushed on docker hub:

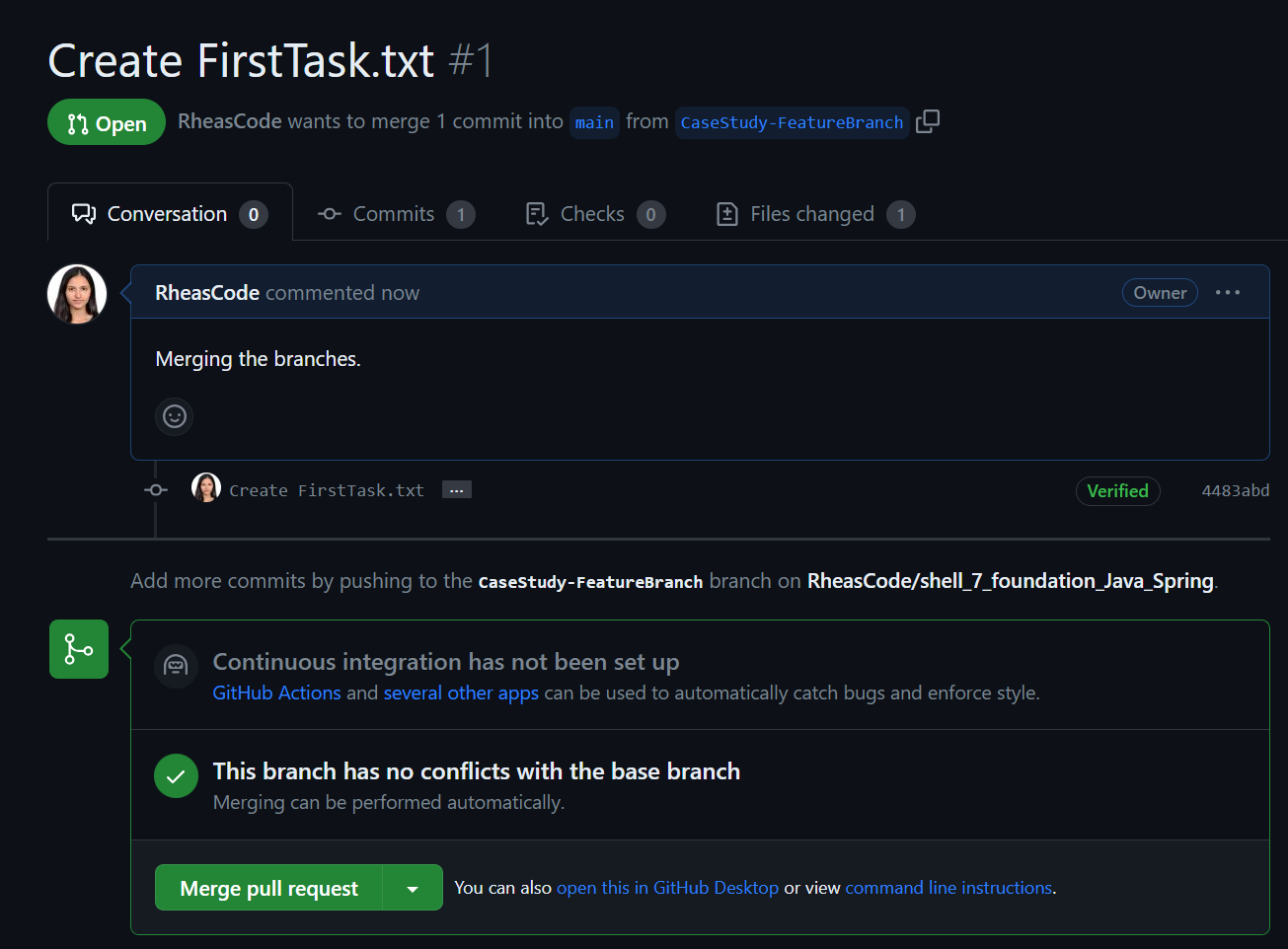


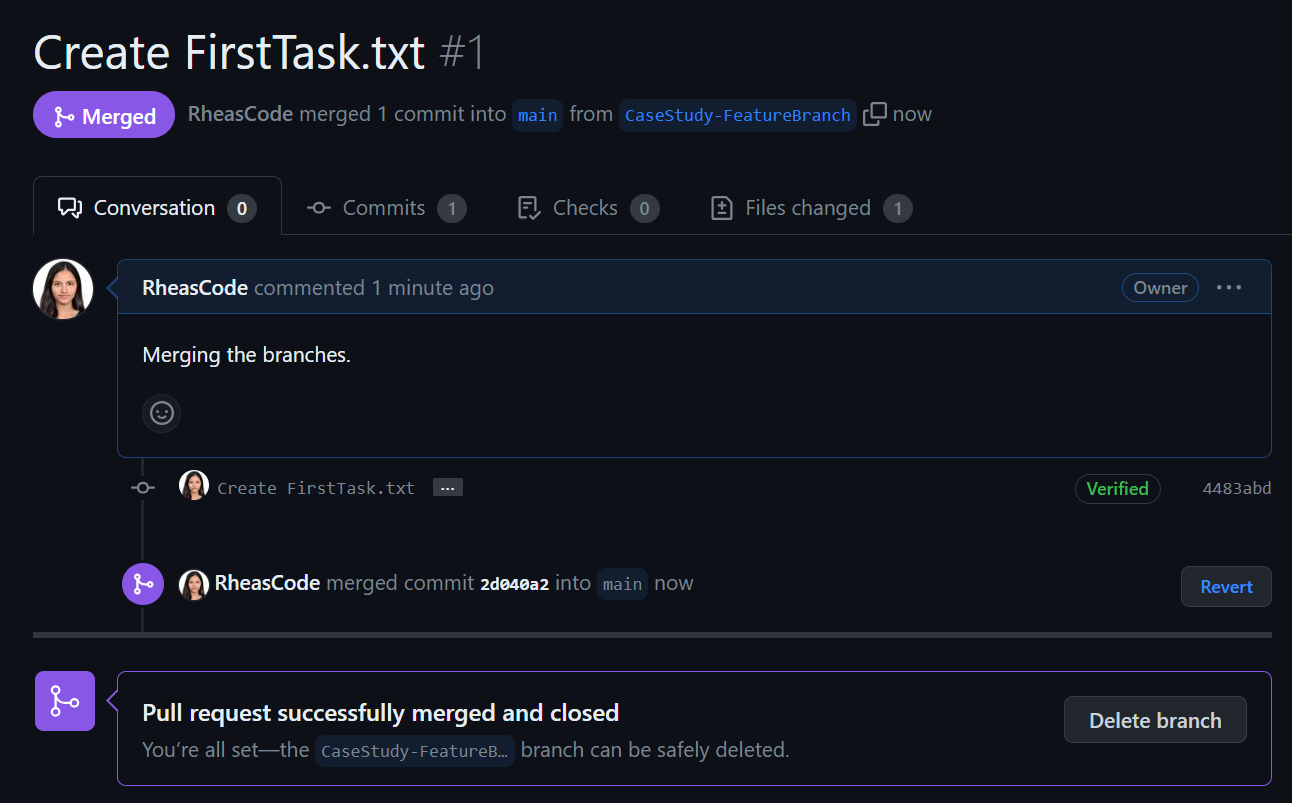
1. Web App Creation:



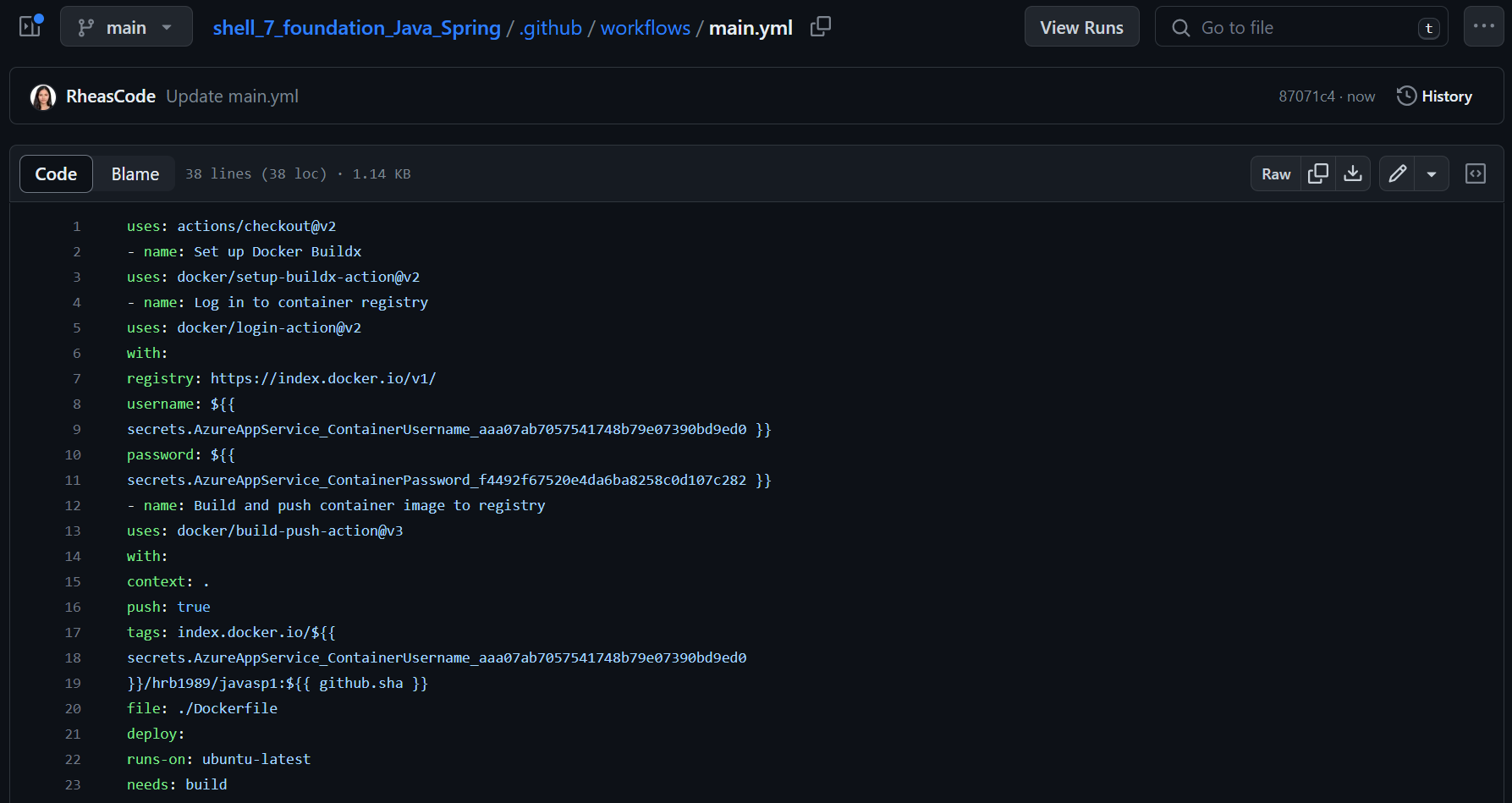
1. Pull request







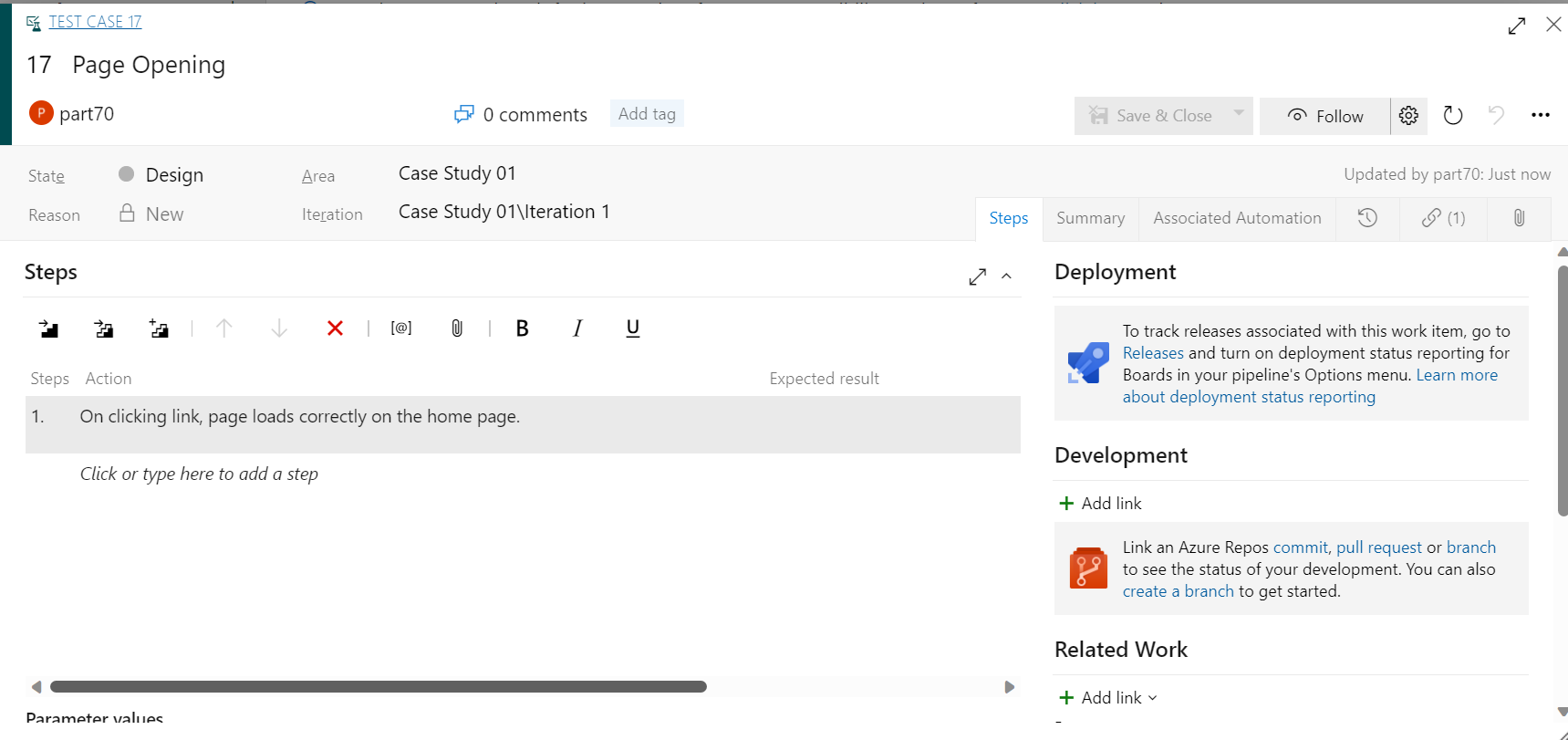
1. Github Actions Workflow:



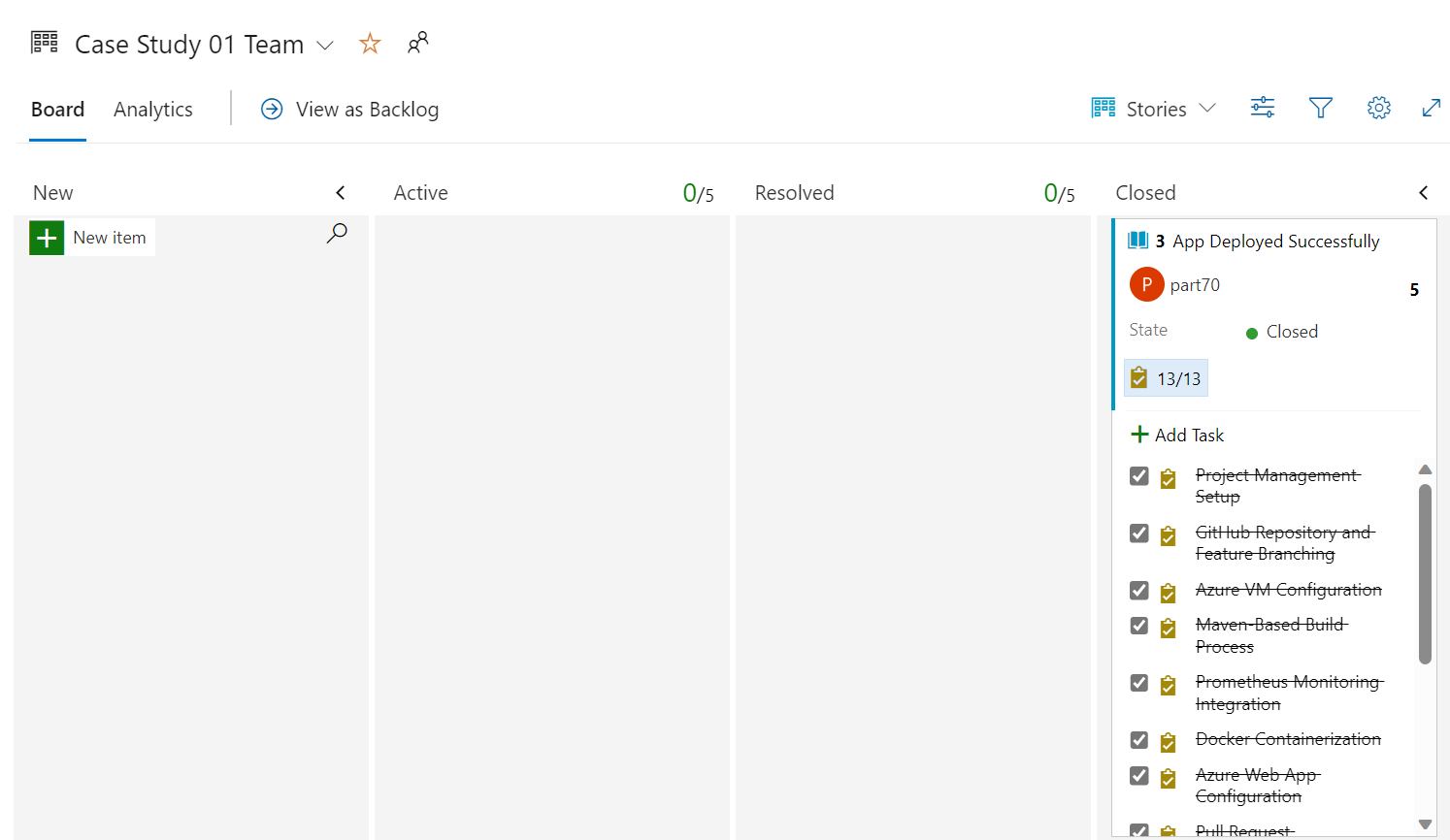
1. Testing:

On testing, the website was working and opening on clicking the link on the VM.

The image was displayed correctly, and the pages menu was clickable and worked as intended.



1. Azure Boards Tasks and Sprint Closure:



1. Project Closure and Reflection:

The website was working well, and in retrospective, there are some things that took a longer time than allotted and could have been handled differently.

A screenshot of a computer

Description automatically generated

1. Outcome

I have learnt about how a sprint works and having implemented the app in a strict time constraint, I have learnt about time management as well as how to work under pressure. I learnt about deploying an app on azure, created a dockerfile and docker image, devops, agile, git, github workflow, etc. Having systematically implemented the CI/CD workflow, Streamline Solutions Inc. aims to achieve rapid and reliable software deployments. The company's Java Spring Boot application will reach customers seamlessly, ensuring a high-quality user experience and setting the stage for future innovation.