St. Francis Institute of Technology, Mumbai-400 103

**Department Of Information Technology**

A.Y. 2024-2025

Class: TE-ITA/B, Semester: V

Subject: **DevOps Lab**

**Experiment – 5: To implement continuous integration with Jenkins**

1. **Aim:** To implement continuous integration with Jenkins
2. **Objectives:** Aim of this experiment is that, the students will be able

* To Integrate and deploy tools like Jenkins and Maven, which is used to build

applications in DevOps environment

1. **Outcomes:** After study of this experiment, the students will be able

* To understand the importance of Jenkins to Build and deploy Software

Applications on server environment.

* Learn about Jenkins (With Architecture)
* To have introduction to Maven / Gradle / Ant

1. **Prerequisite:** Knowledge of software engineering concept of integration
2. **Requirements:** Jenkins,JDK, python, ANT,Personal Computer, Windows operating system, browser, Internet Connection, Microsoft Word.
3. **Pre-Experiment Exercise:**

**Brief Theory:** Refer shared material

1. **Laboratory Exercise**
   * + 1. **Procedure:**

**a. Answer the following:**

* Explain continuous integration

Continuous Integration (CI) is a development practice where developers frequently integrate their code changes into a shared repository, usually several times a day. Each integration is automatically verified by building the project and running tests against it to detect integration errors as quickly as possible.

In Jenkins, Continuous Integration (CI) involves:

* **Monitoring Code Changes:** Jenkins watches a shared repository for new code commits.
* **Automated Builds:** When changes are detected, Jenkins automatically builds the project.
* **Running Tests:** Jenkins runs tests to check for issues.
* **Feedback:** Jenkins provides immediate feedback on the build and test results.
* Why Jenkins is popular? Mention advantages.

Jenkins is popular for several reasons and offers many advantages:

1.Jenkins is free to use and has a large, active community contributing plugins and support.

2.It supports a vast array of plugins (over 1,800) for integrating with various tools and technologies, making it highly customizable.

3.Jenkins is straightforward to install and configure, with options for running it as a standalone application or as a server.

4.It integrates with various source control management systems like Git, Subversion, and Mercurial.

5.Jenkins supports both Continuous Integration and Continuous Delivery (CD), allowing for automated deployment.

6.Jenkins pipelines can be defined as code (using Jenkinsfile), making it easier to manage and version control the build process.

7.Jenkins supports distributed builds, allowing it to scale across multiple machines to handle large projects and concurrent builds.

8.The extensive ecosystem of plugins and integrations provides functionality for monitoring, notifications, and reporting.

9.The large Jenkins community offers extensive documentation, forums, and user groups for support and troubleshooting.

**b**. **Execute following (Refer the shared material) and attach screenshots:**

* Build jobs in Jenkins

1. **Post-Experiments Exercise**
2. **Extended Theory:**

Nil

1. **Questions:**

* How is continuous integration achieved using Jenkins?
* Have you created a build job in Jenkins? Explain how to do it.
* What are the types of jobs or projects in Jenkins?

1. **Conclusion:**

* Write what was performed in the experiment.
* Write the significance of the topic studied in the experiment.

1. **References:**

<https://jenkins.io/doc/>

<https://www.cloudbees.com/jenkins/what-is-jenkins>

<https://vmokshagroup.com/blog/what-is-jenkins/>

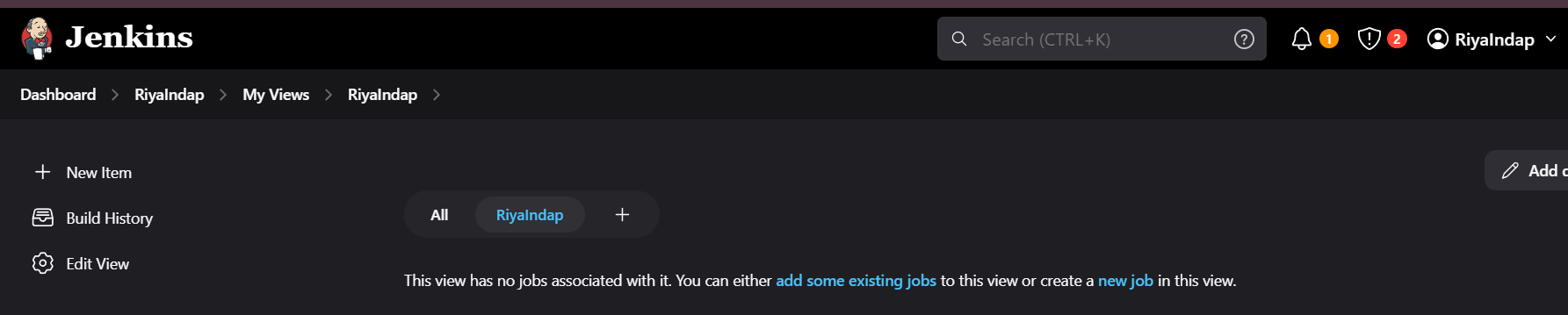
<https://www.infoworld.com/article/3239666/what-is-jenkins-the-ci-server-explained.html>

<https://hackr.io/blog/jenkins-interview-questions>

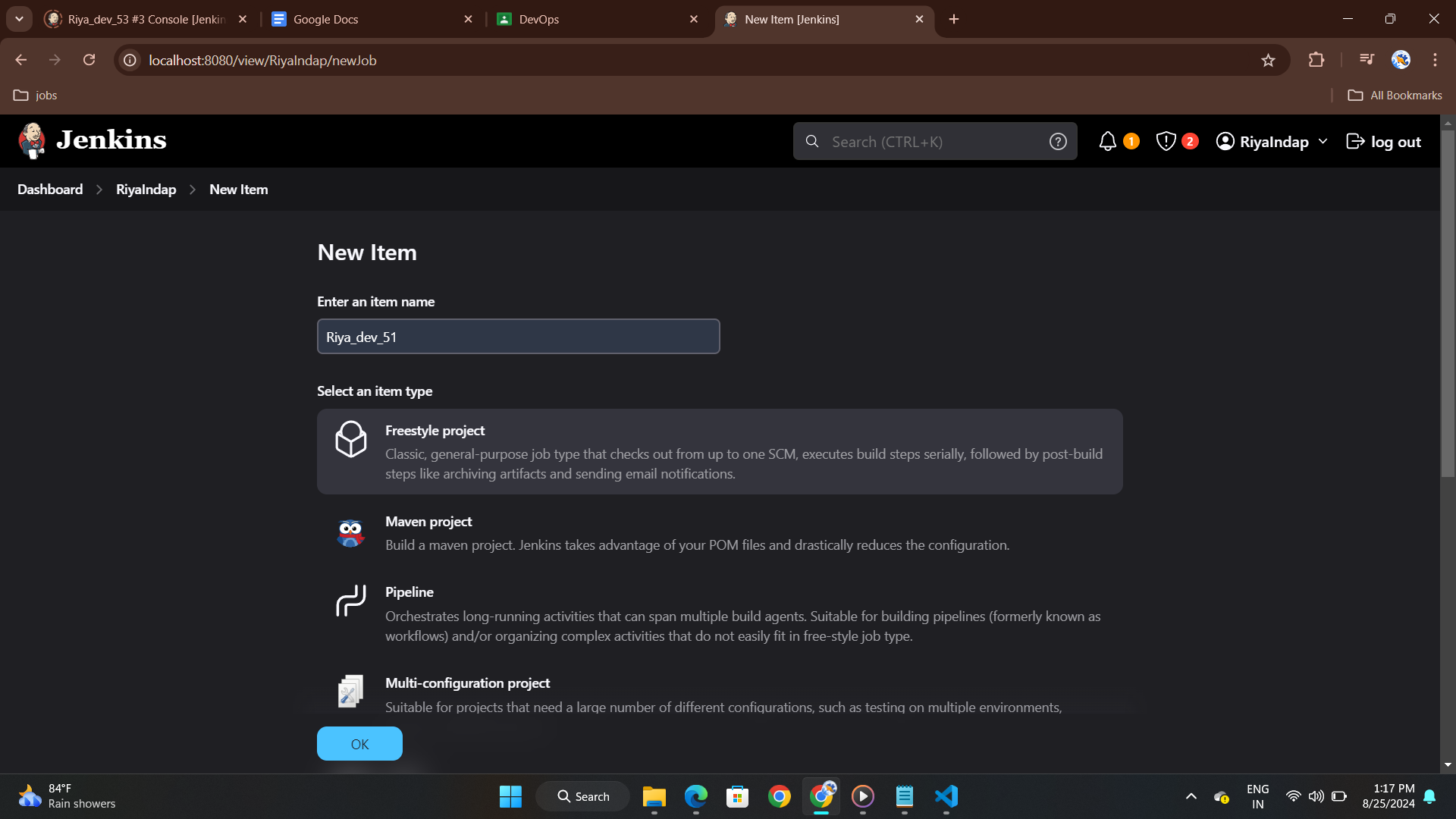
<https://www.edureka.co/blog/interview-questions/jenkins-interview-questions/>

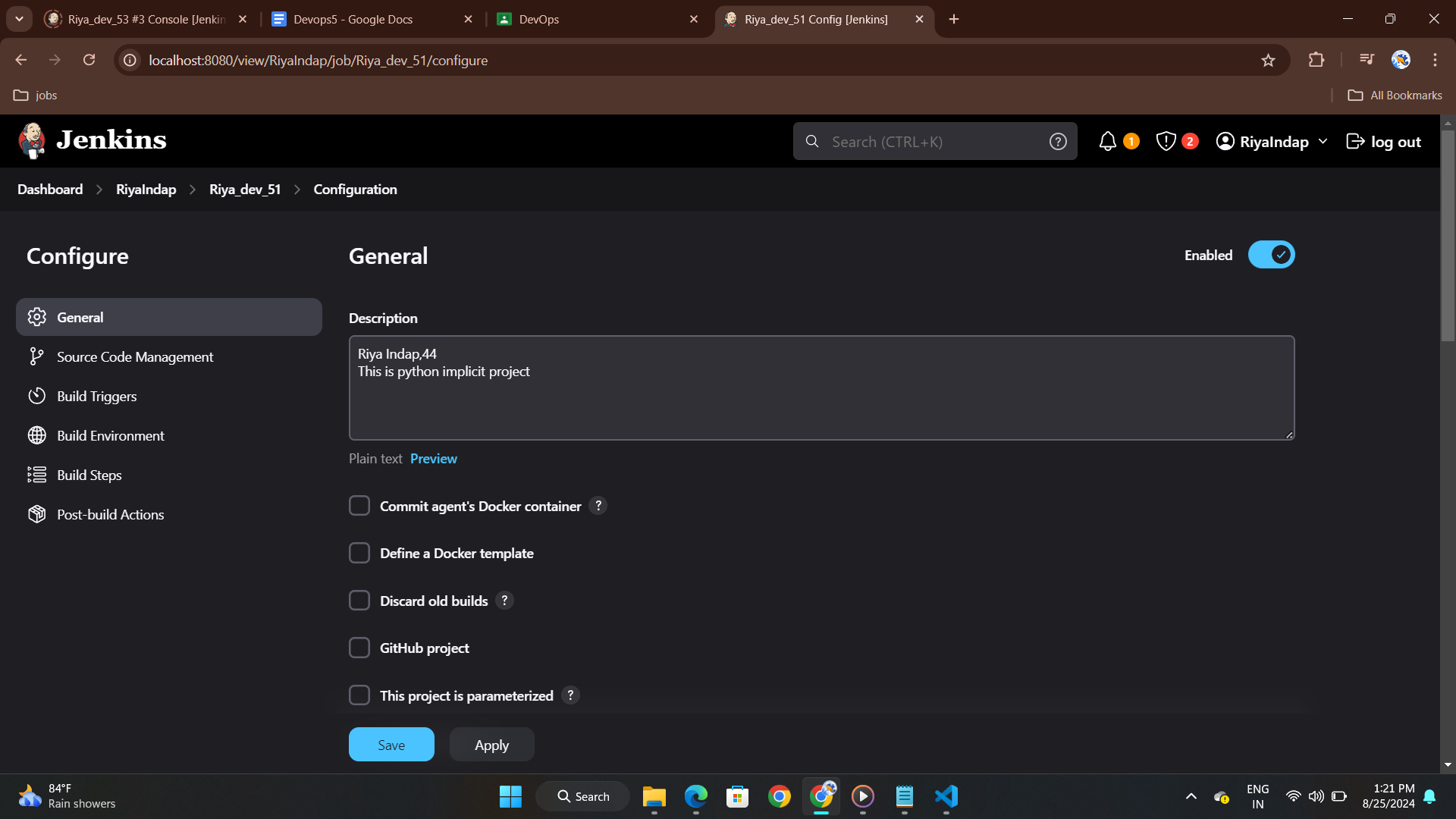


**Creating View:**

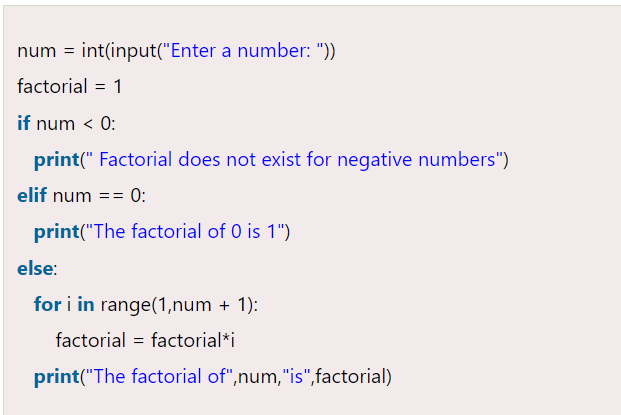


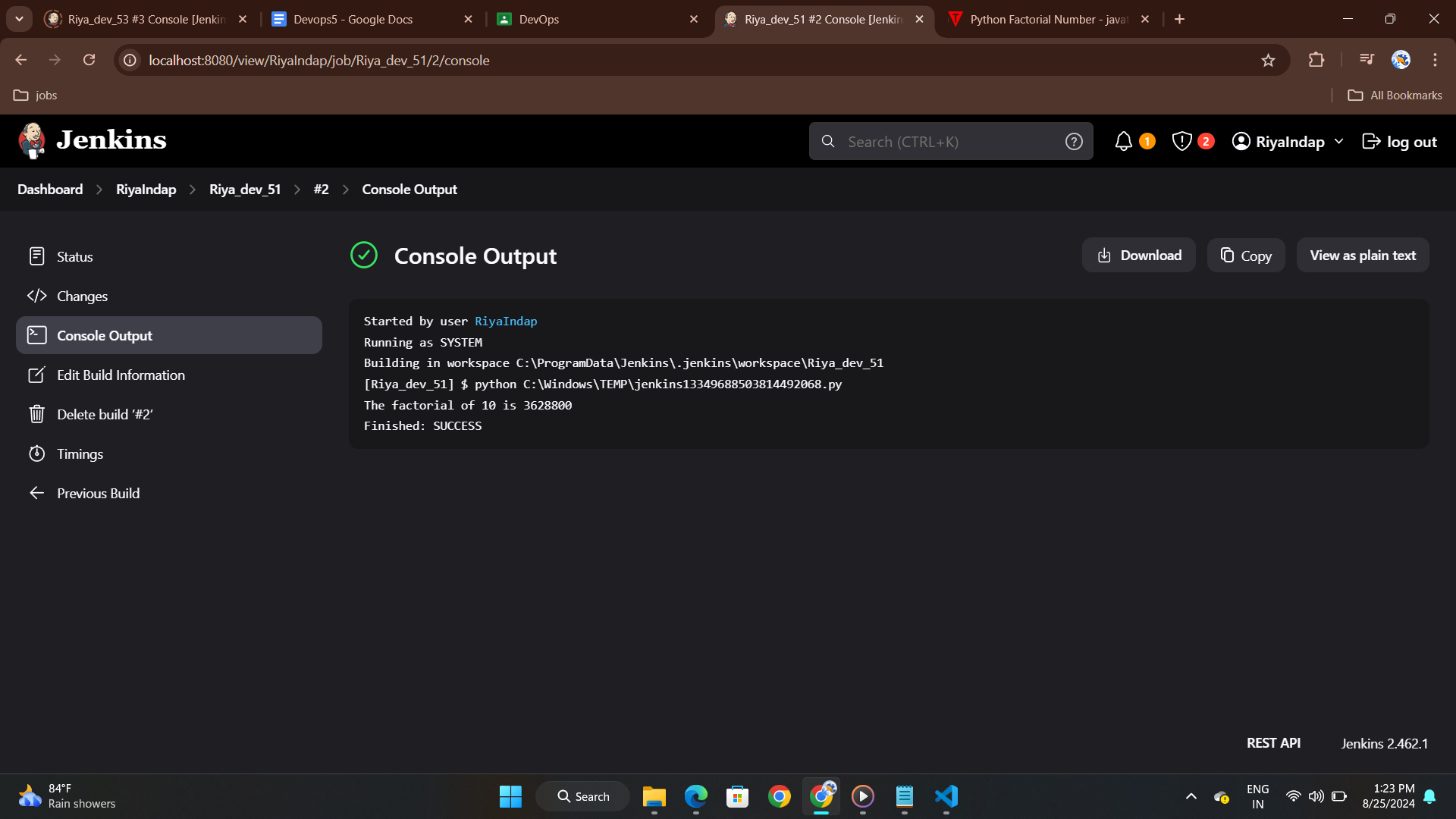
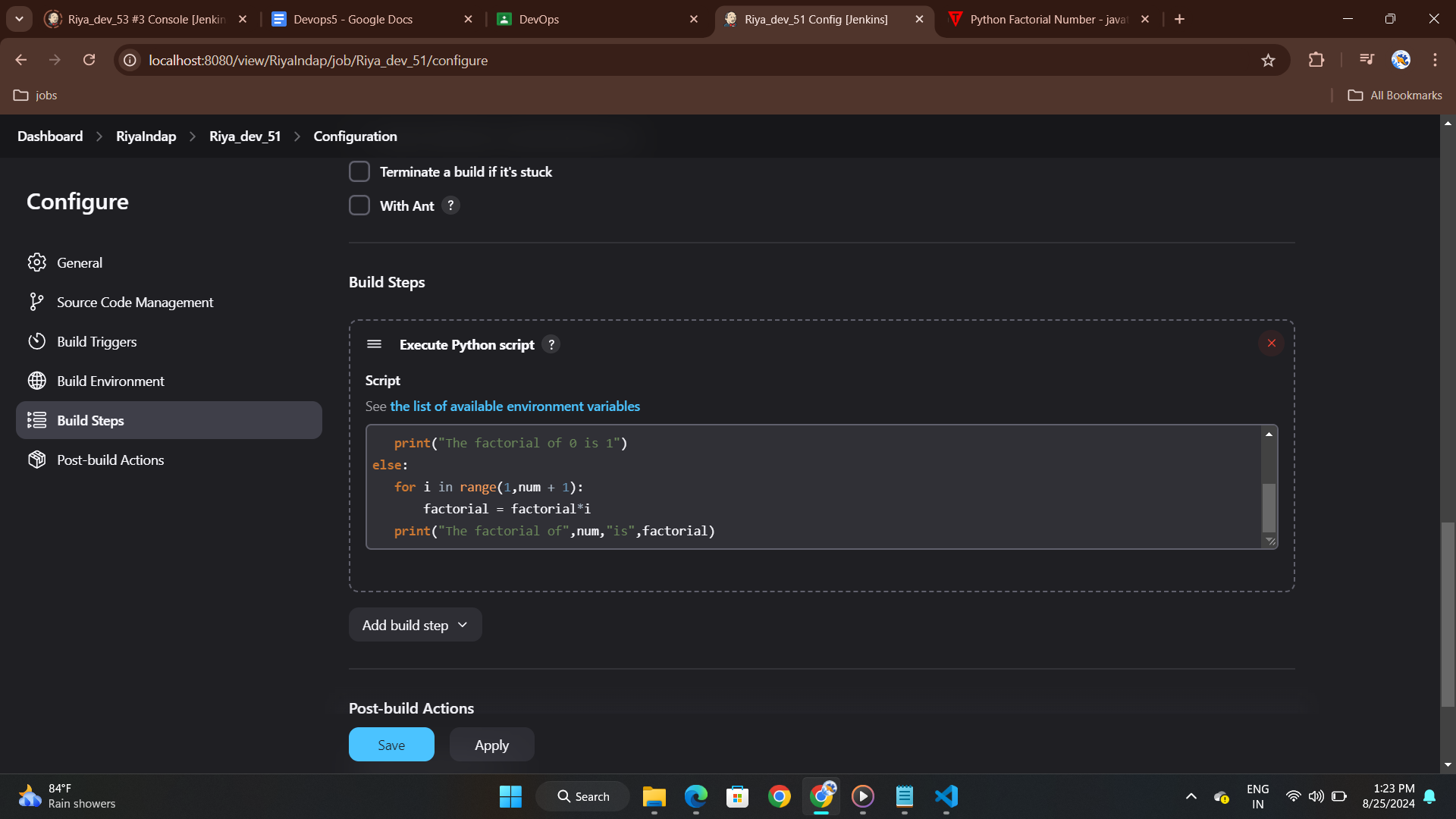
**Project 1.**



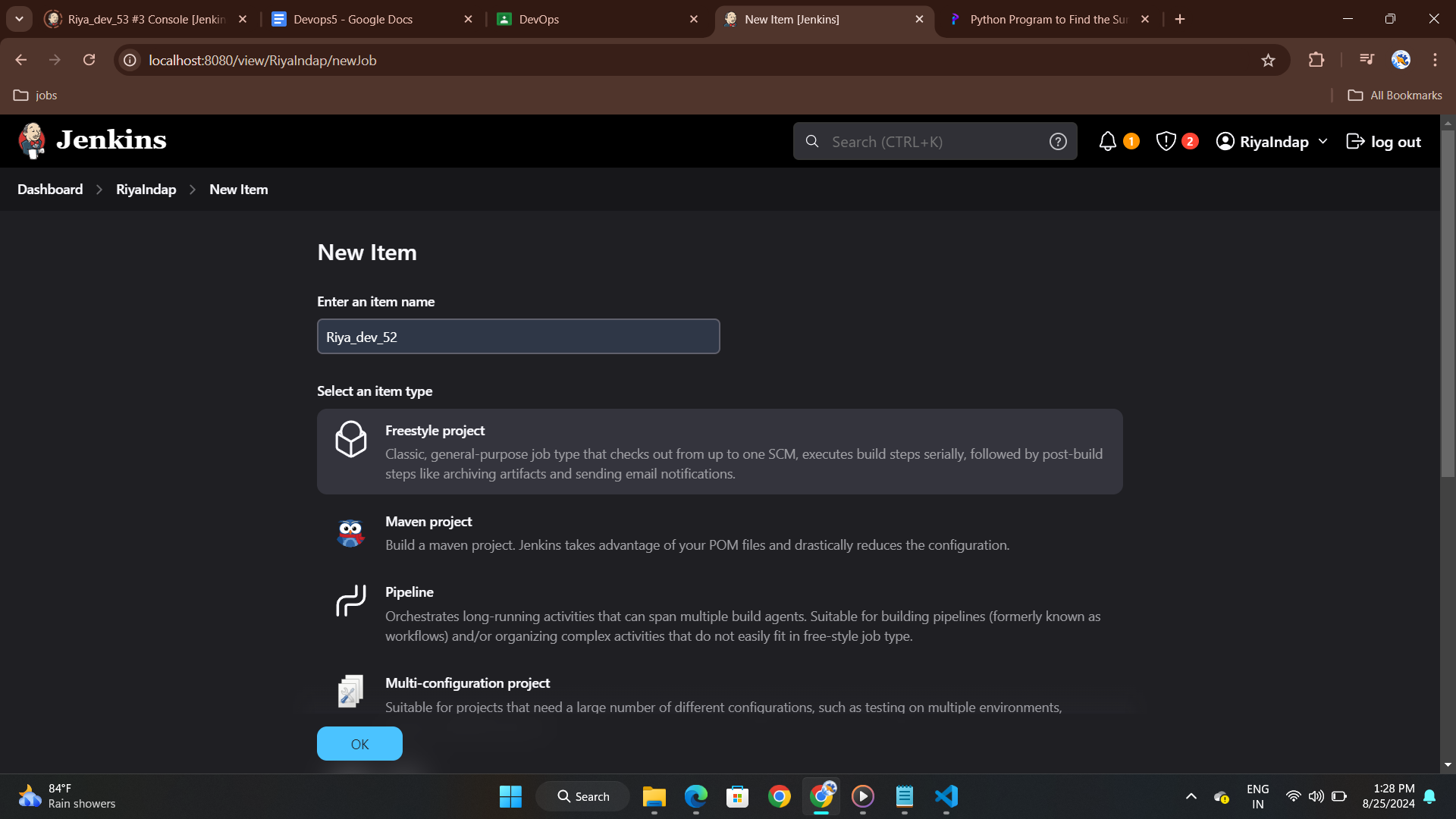
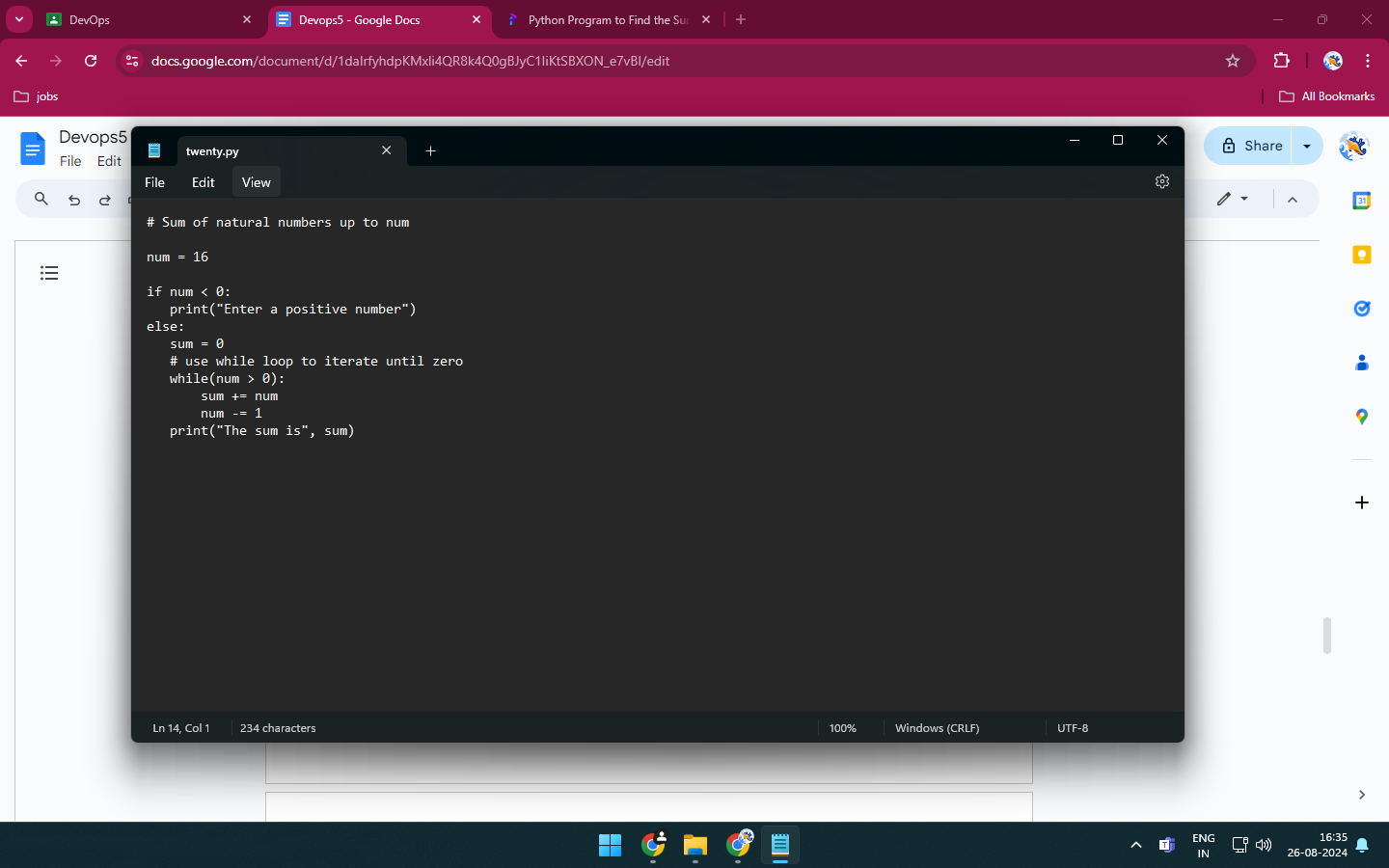


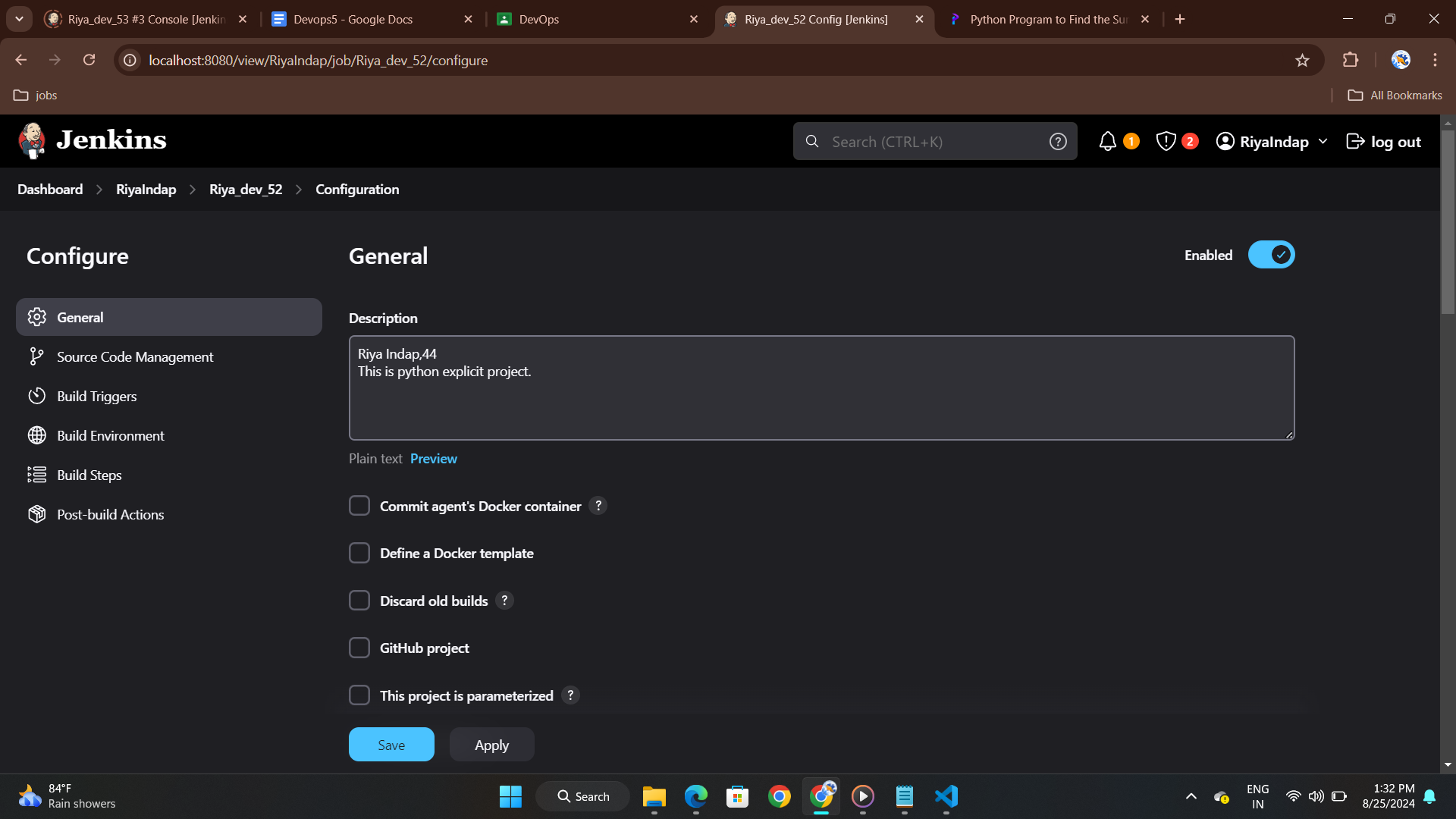
code:

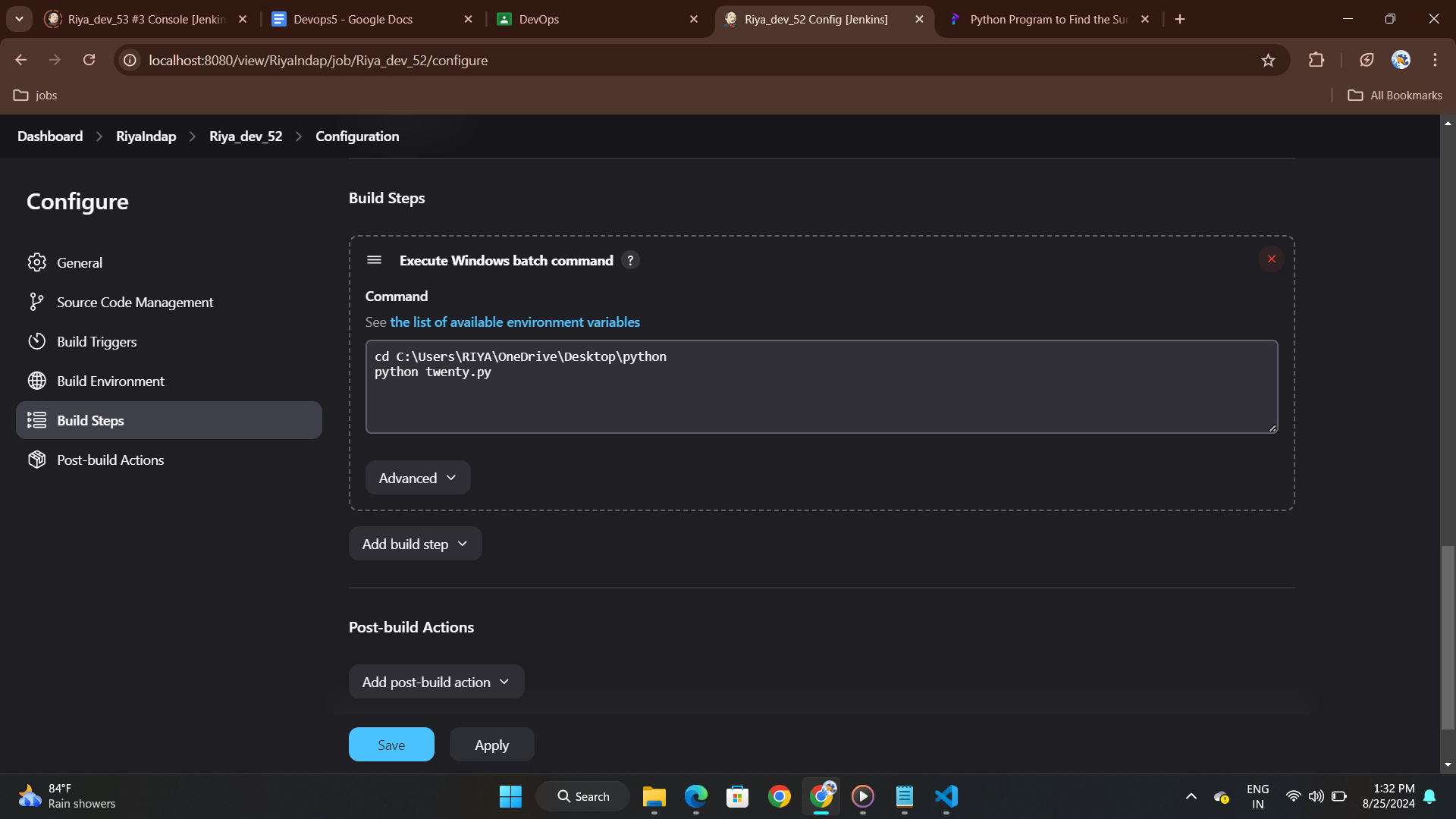


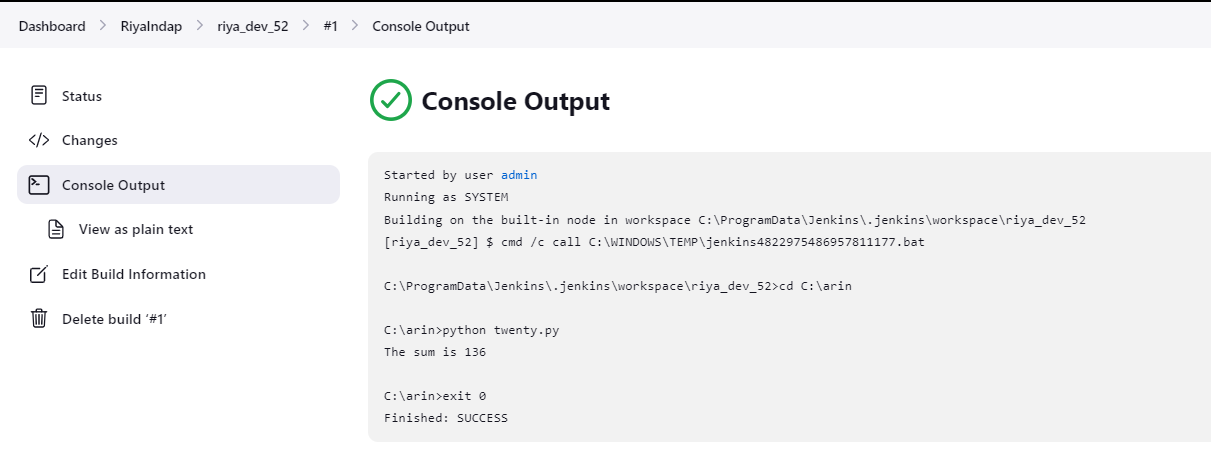


**Project 2:**

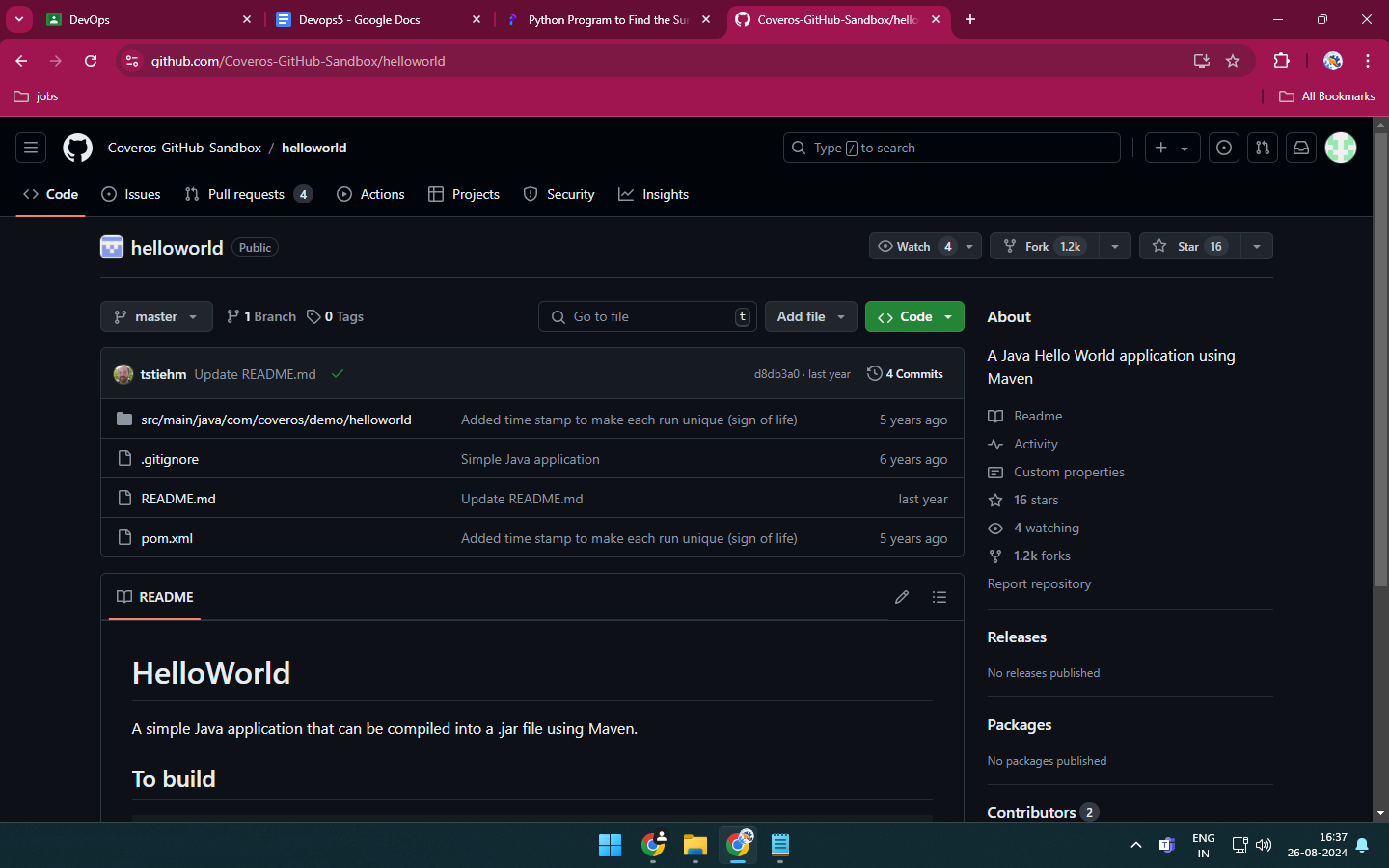


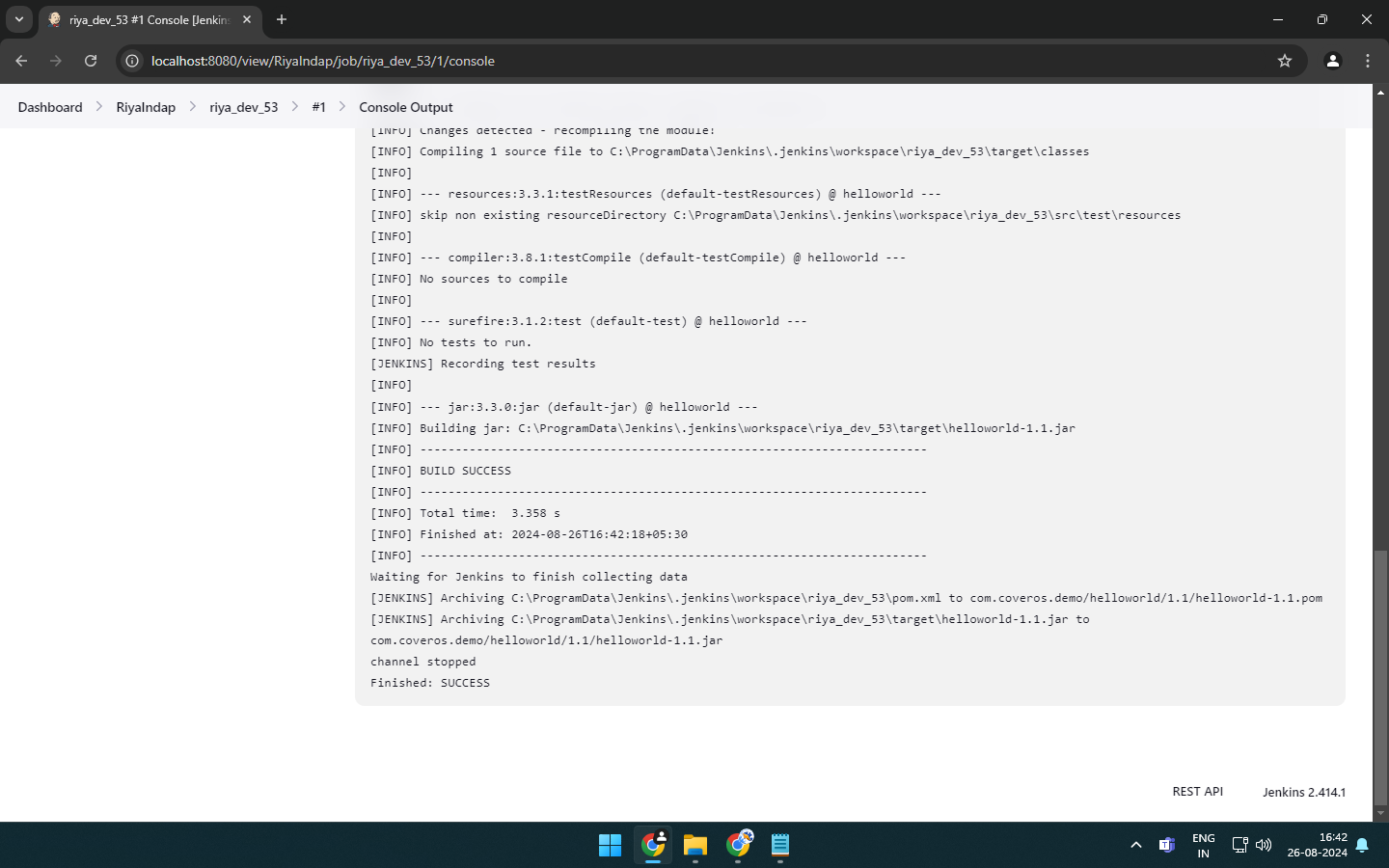
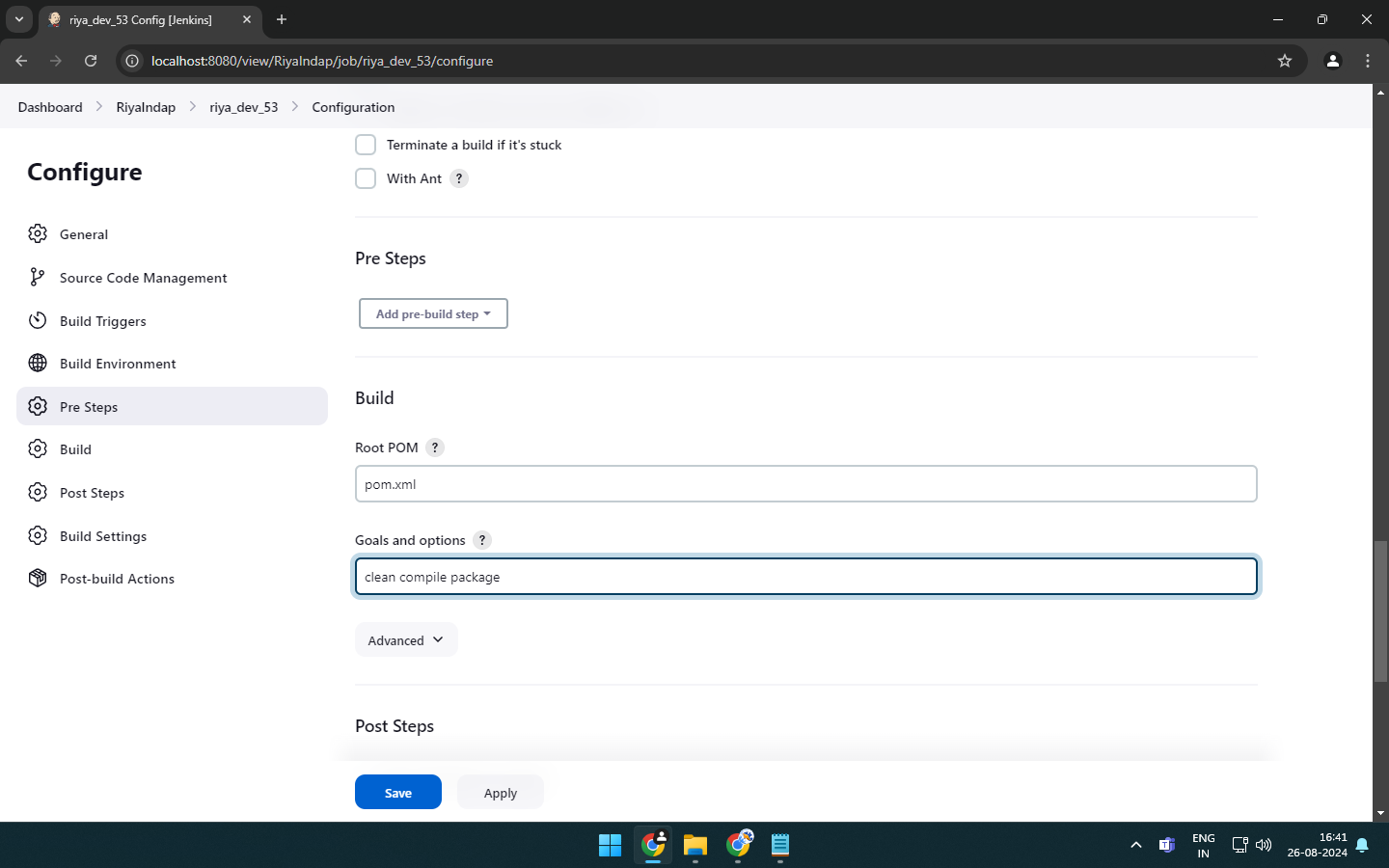
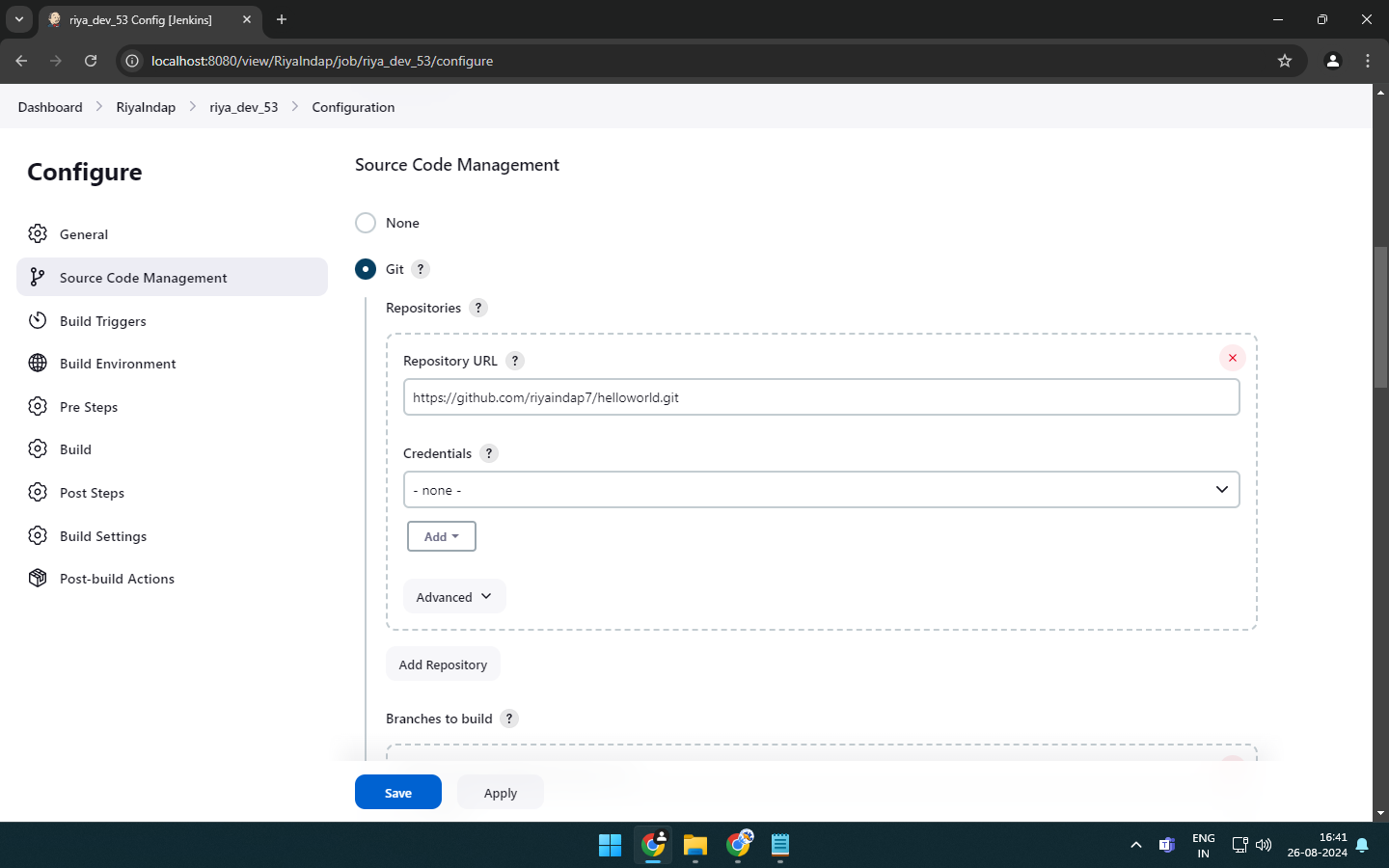
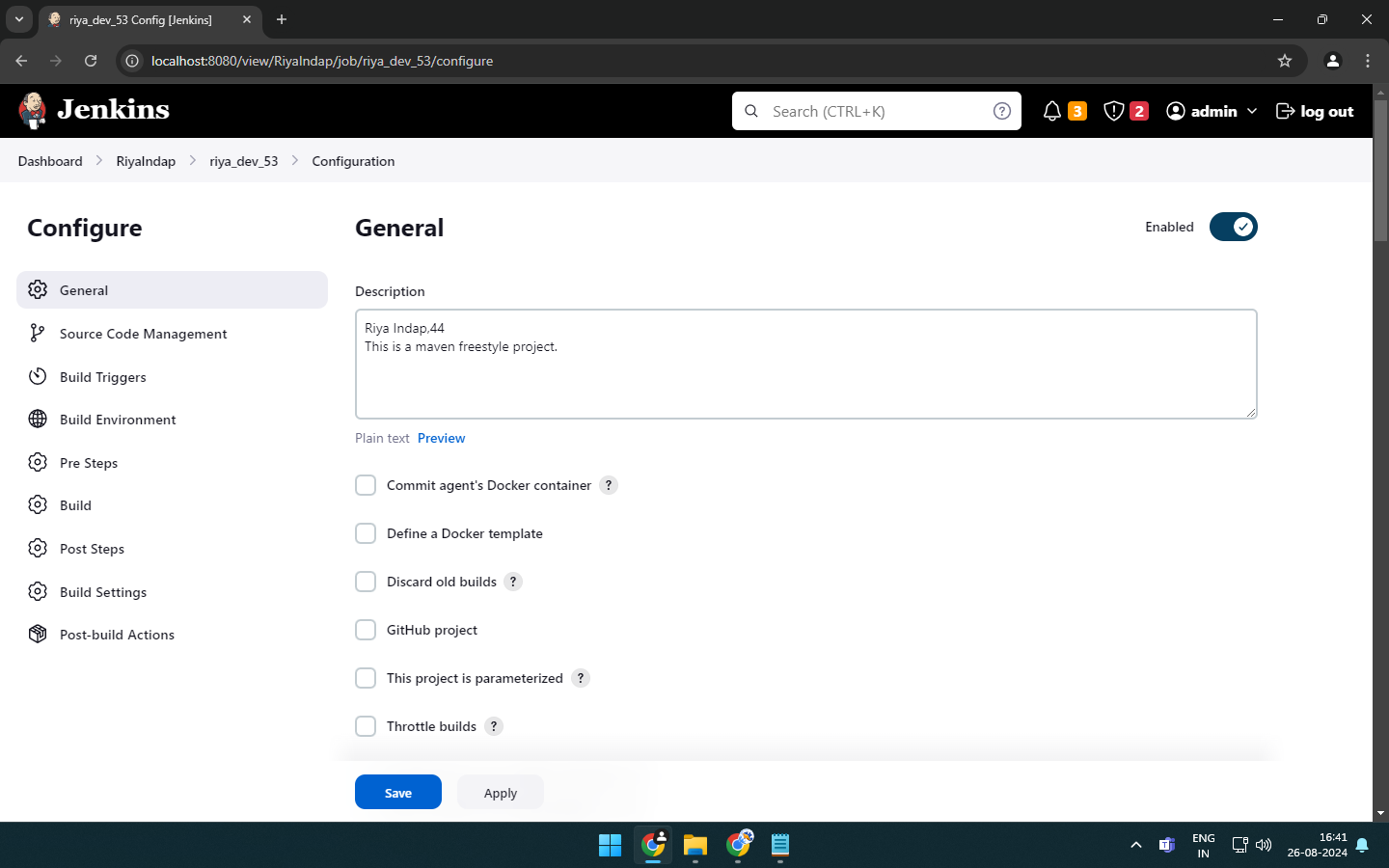
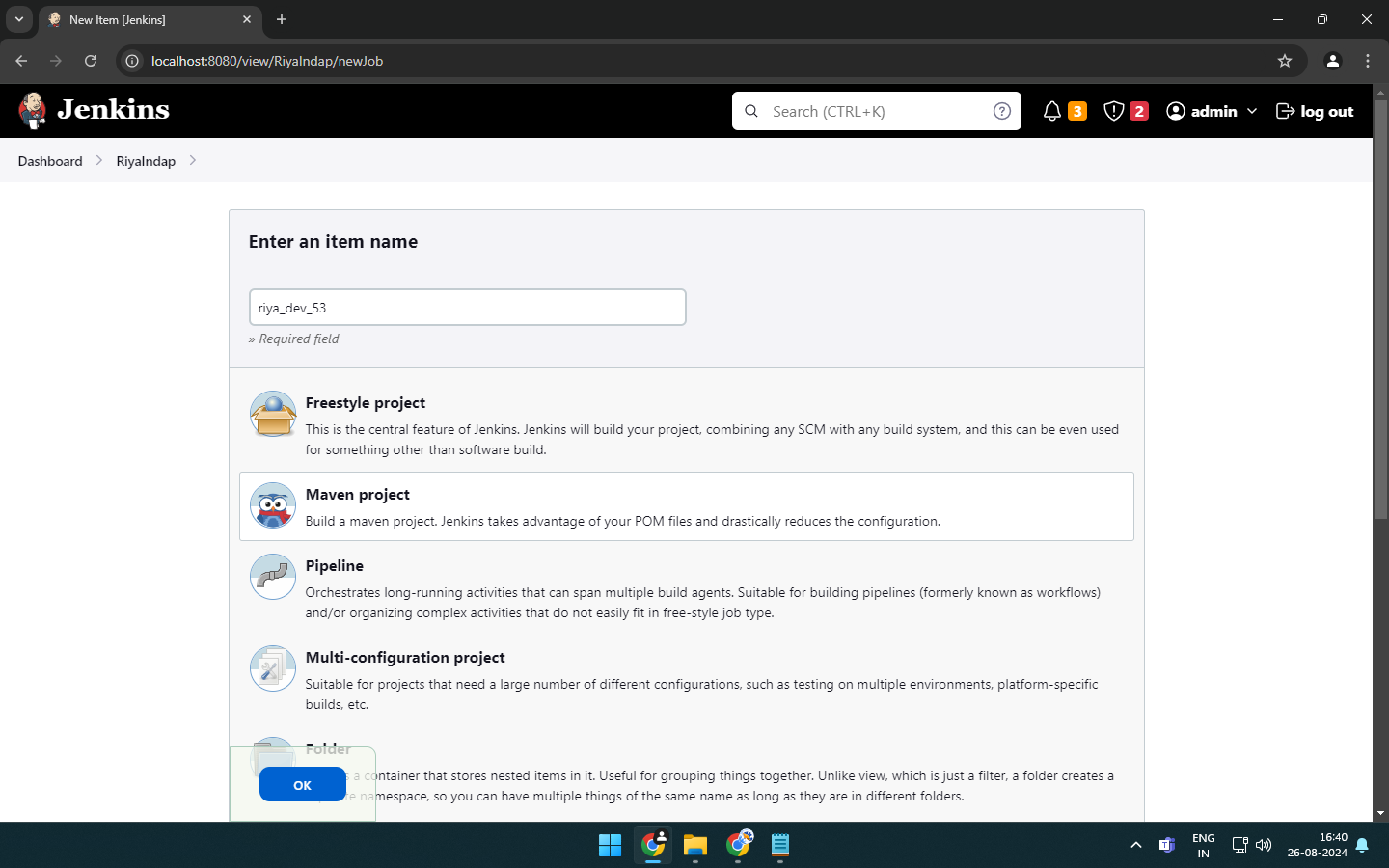
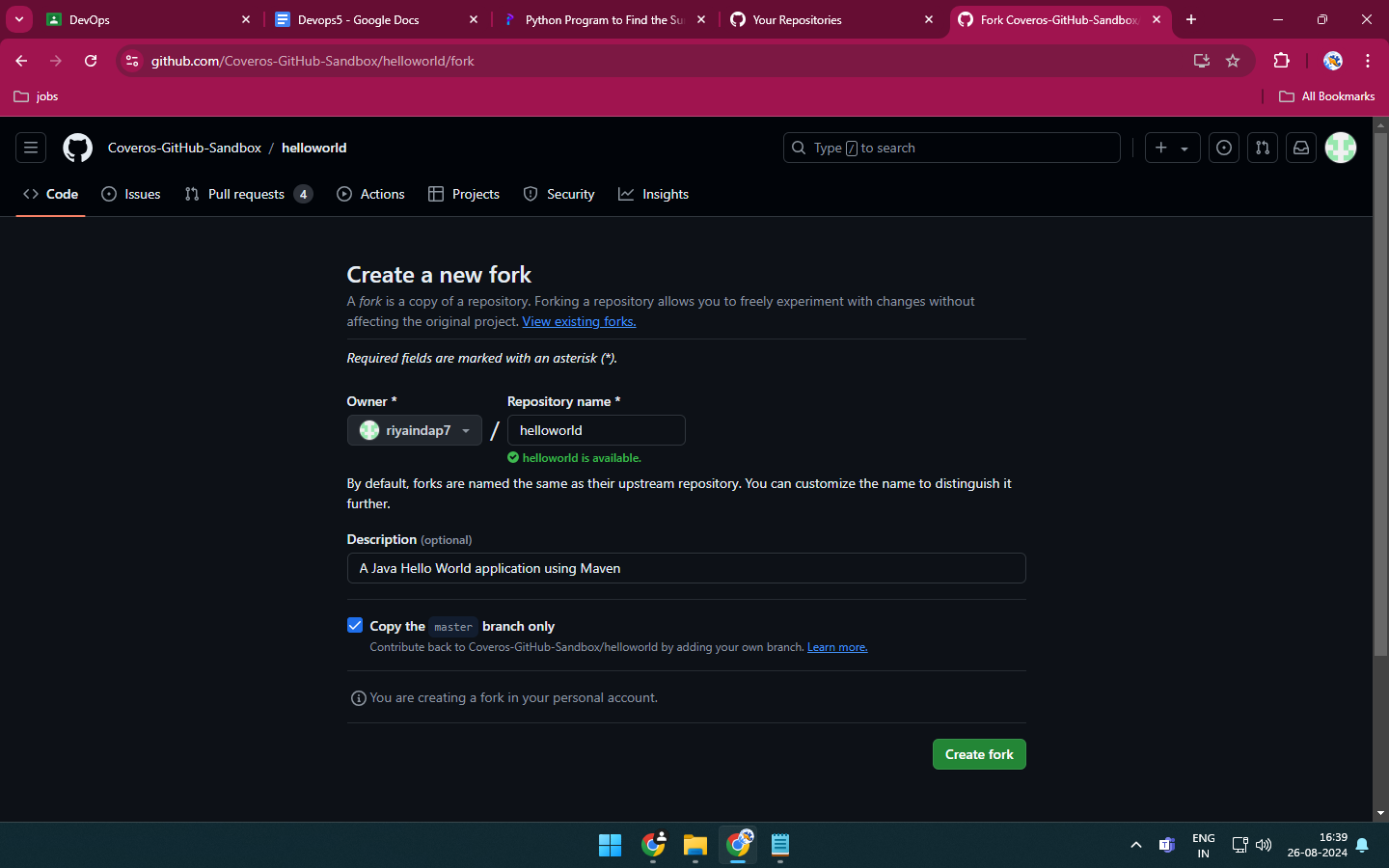


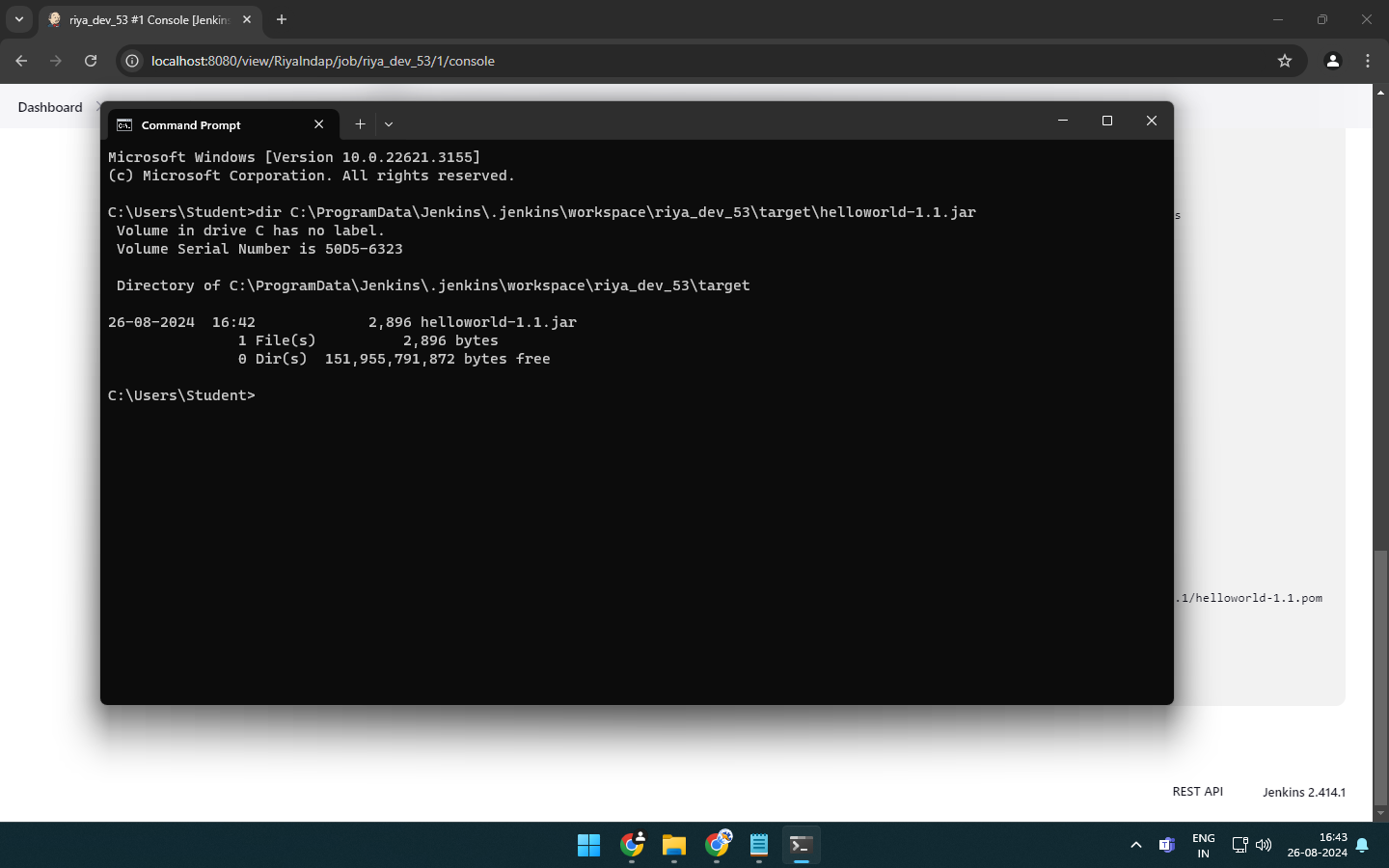




**Project3:**







**Project 4:**

