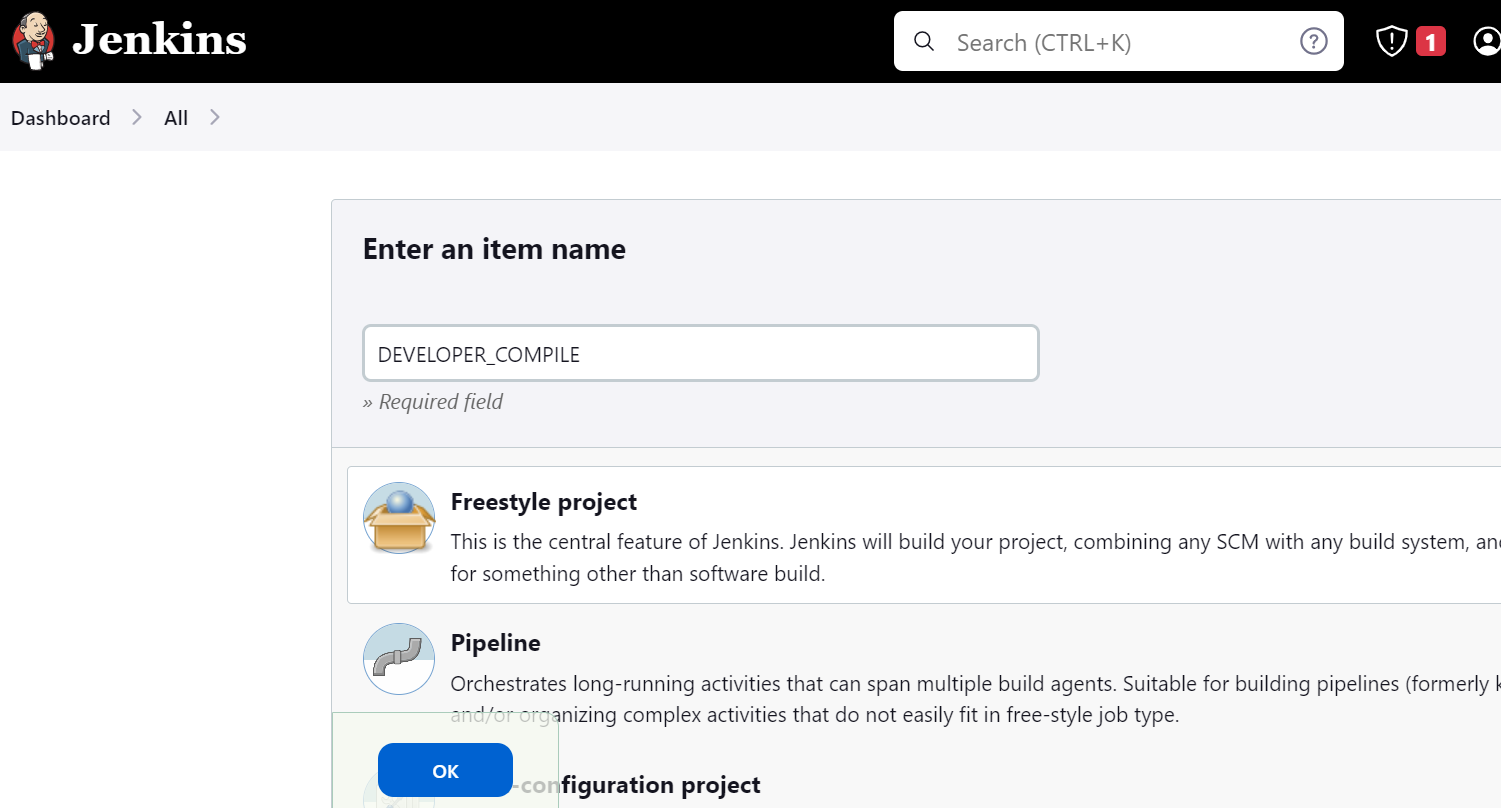
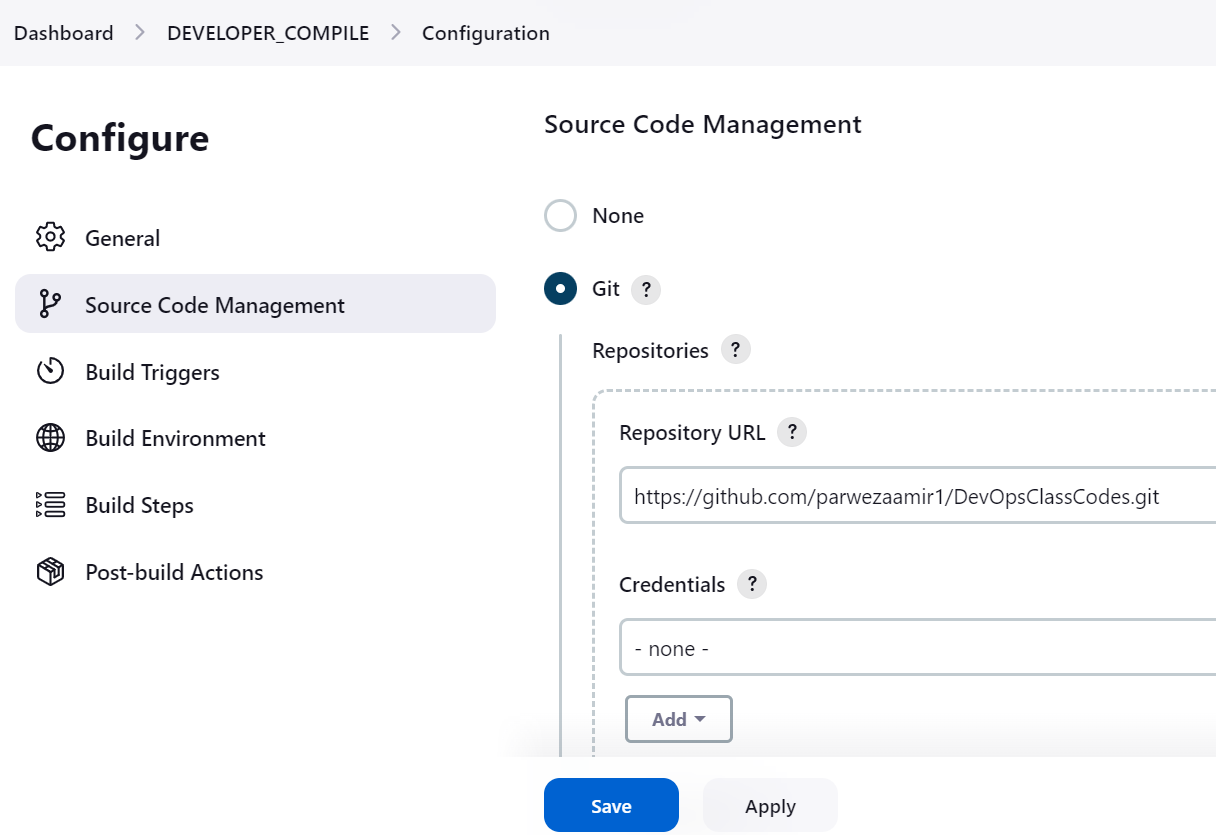
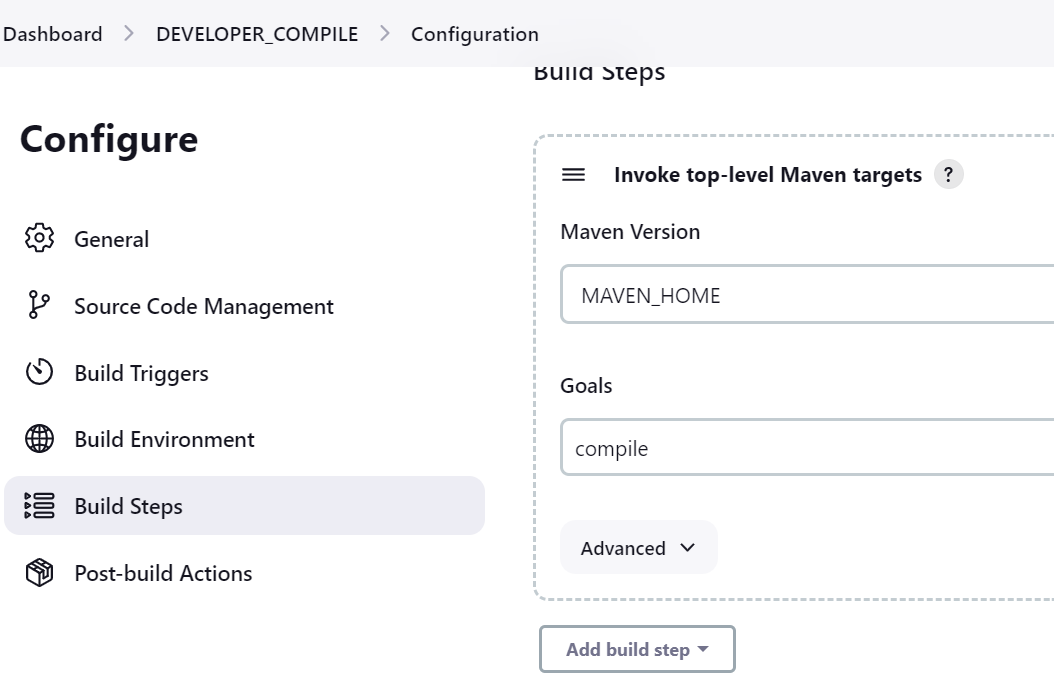
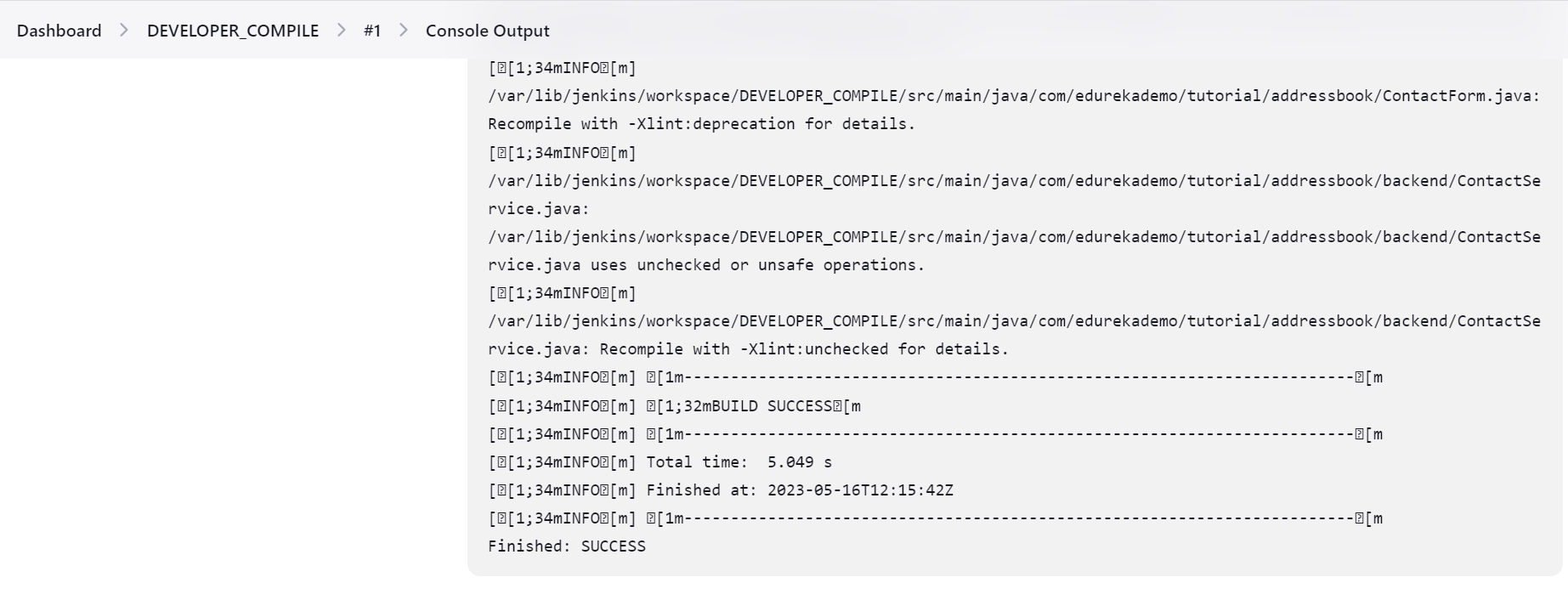
1. **DEVELOPER\_COMPILE**

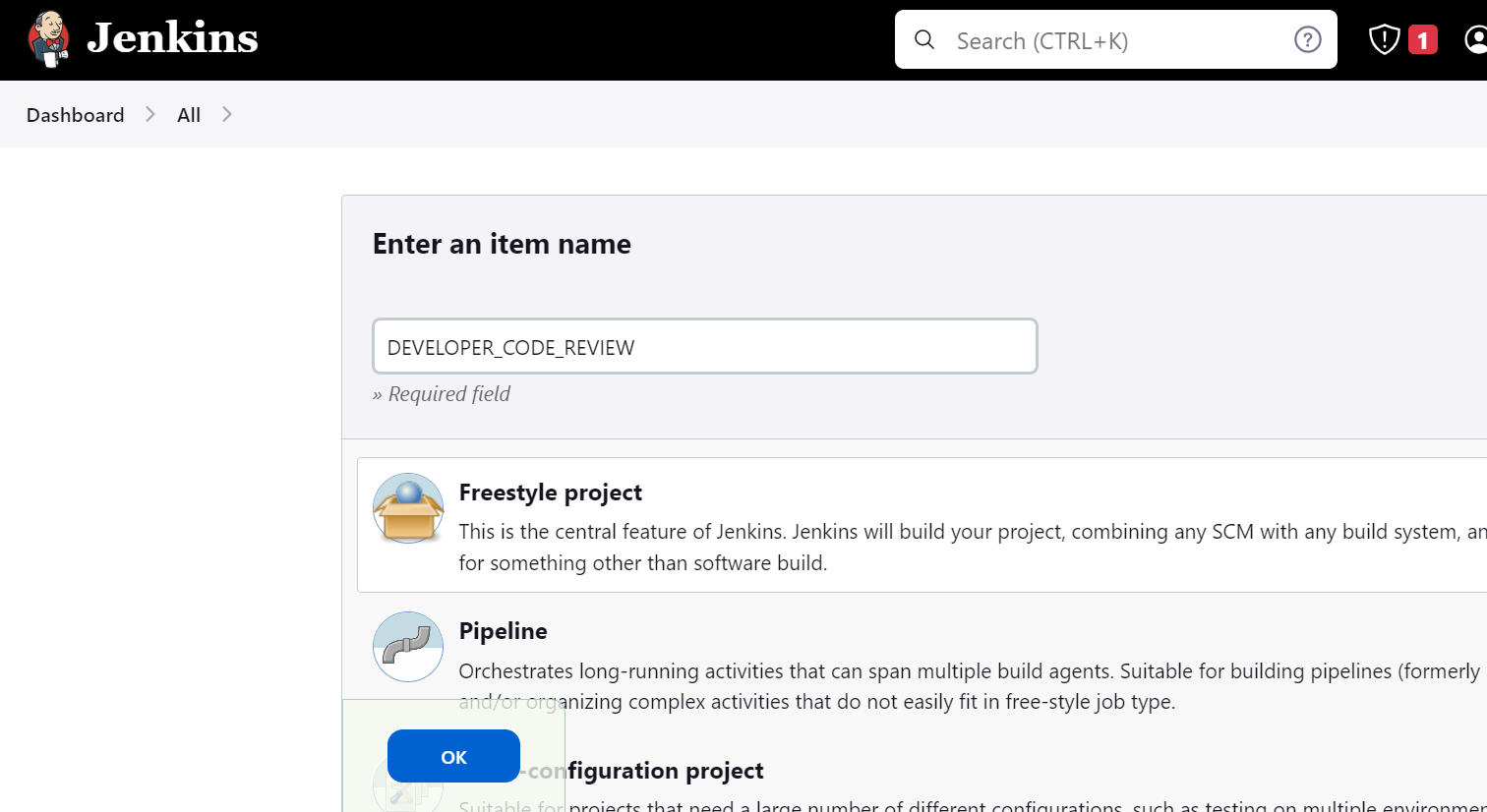
****

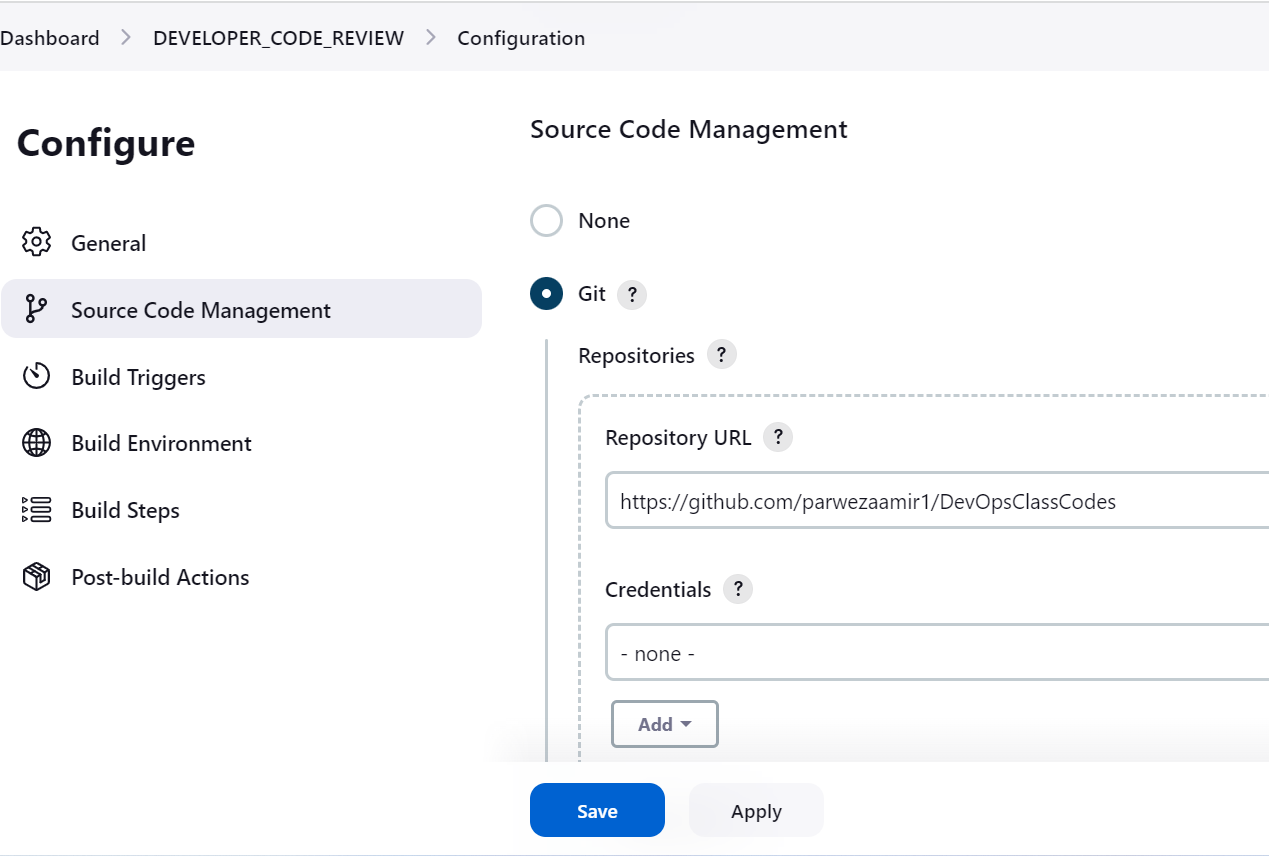
****

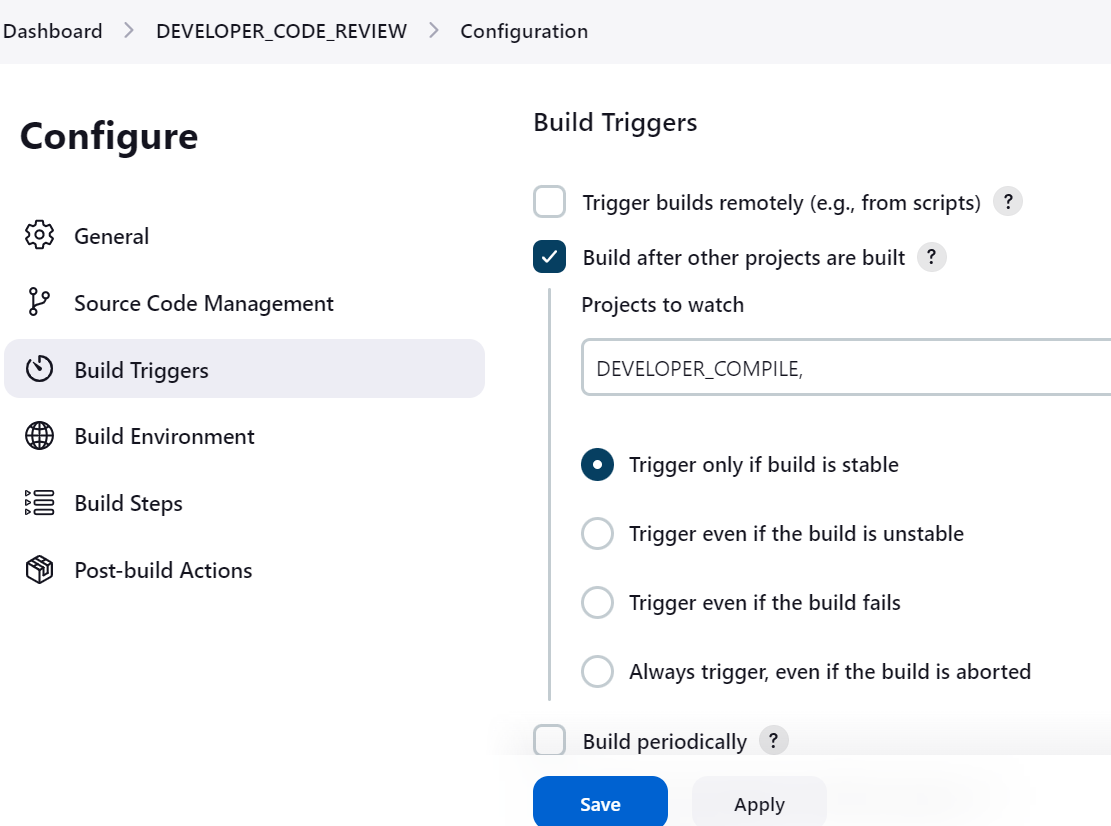
****

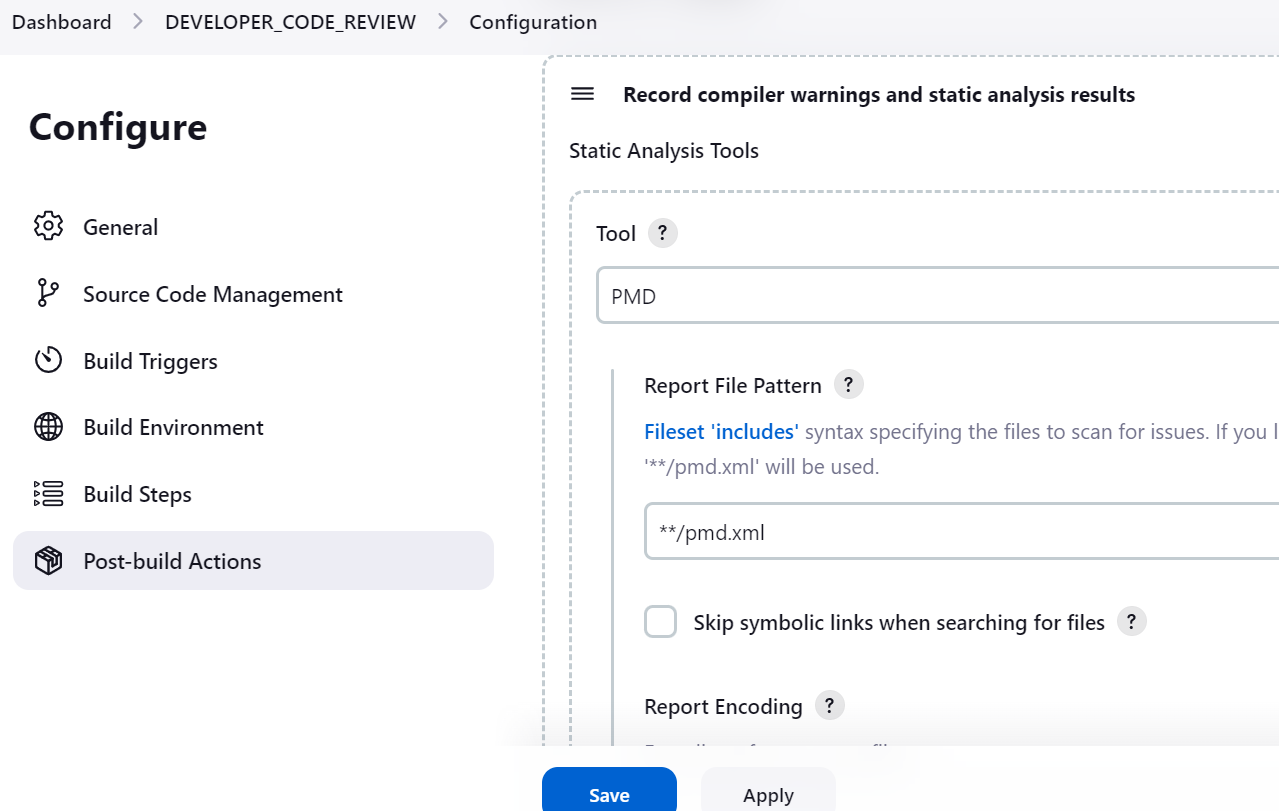
****

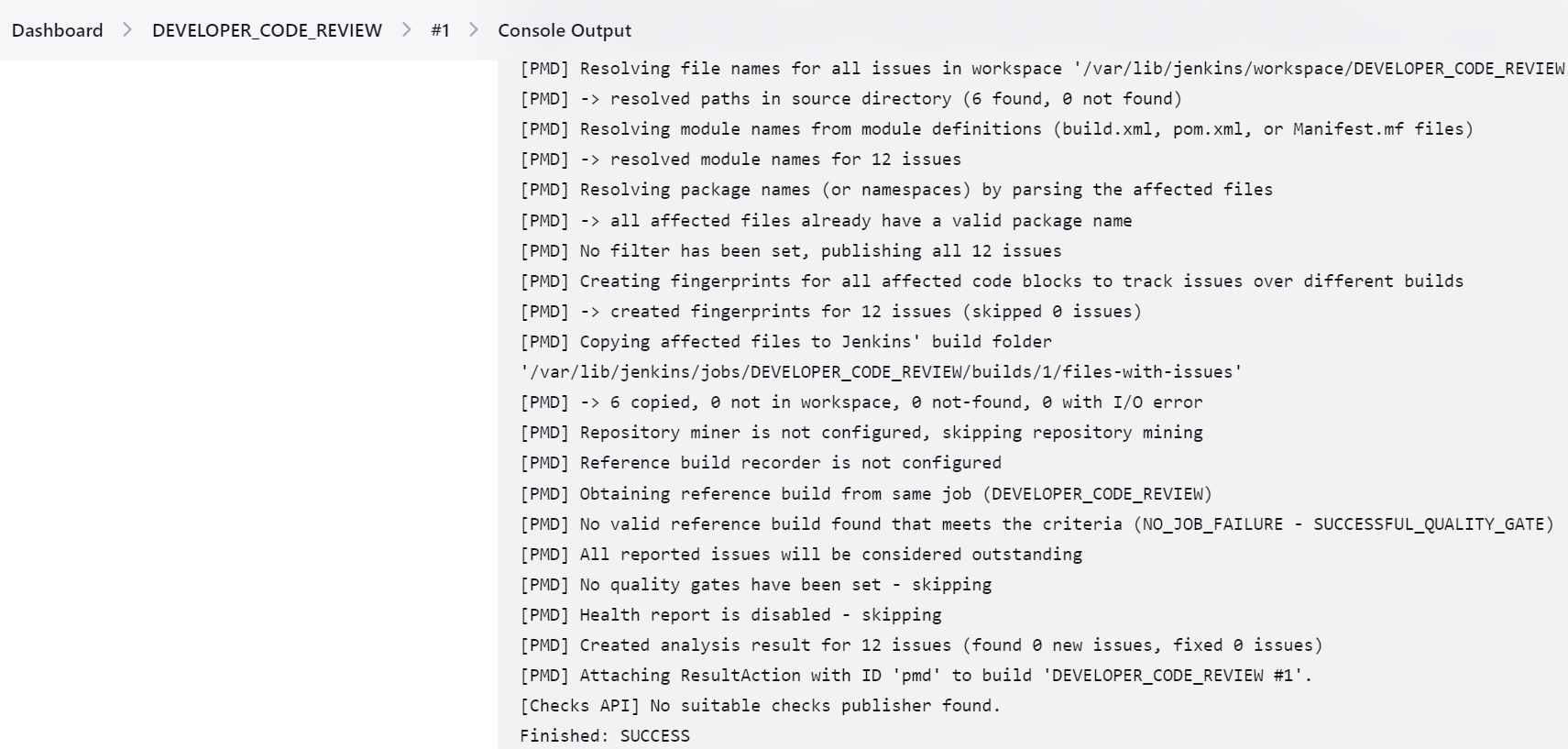
1. **DEVELOPER\_CODE\_REVIEW**

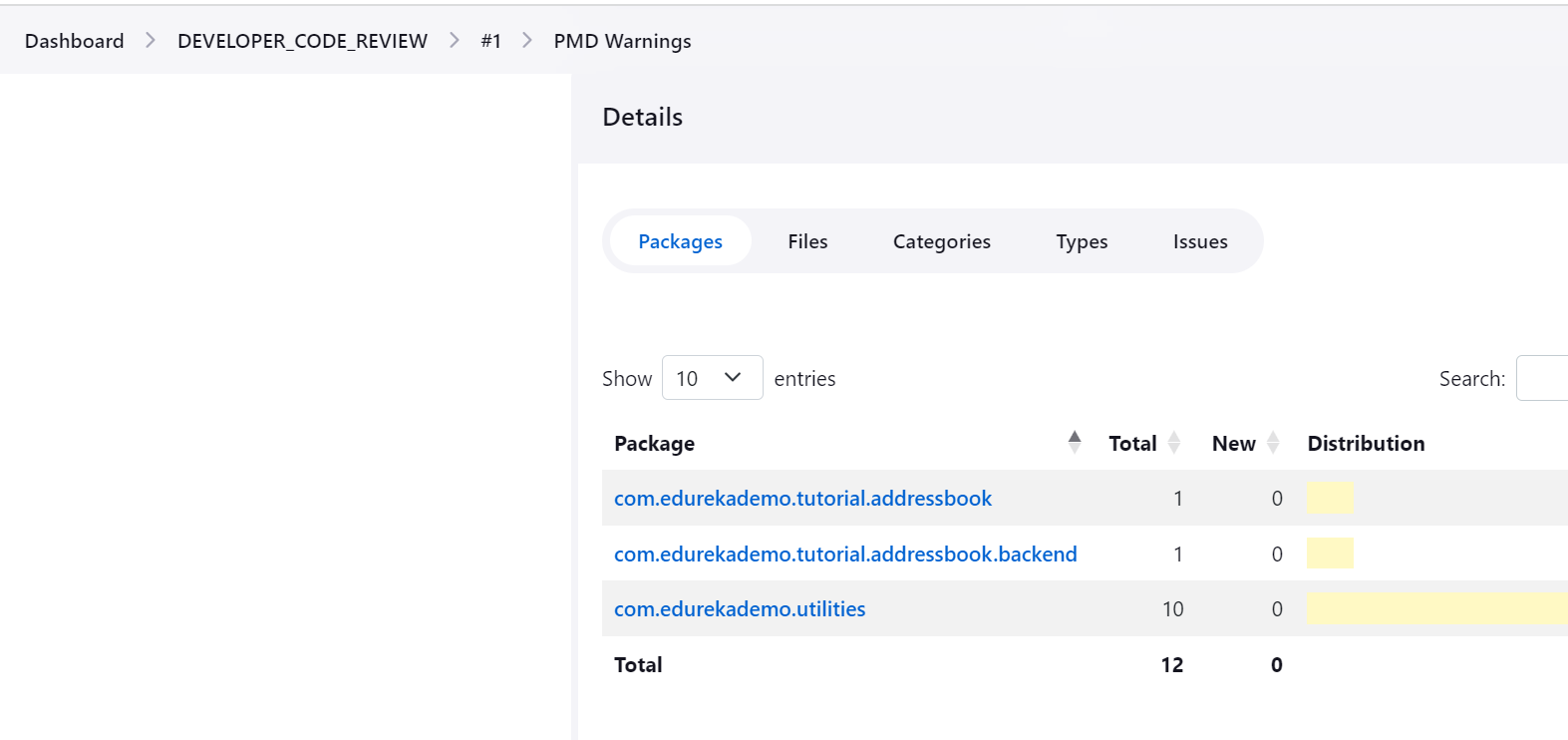
****

****

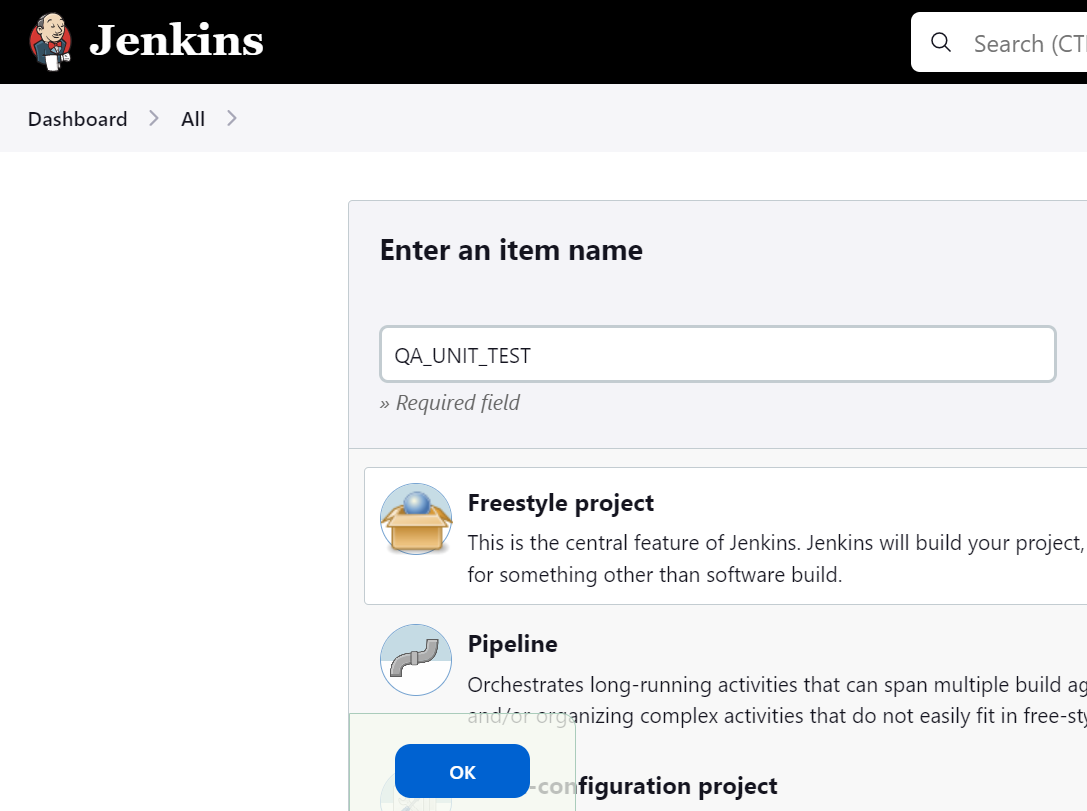
****

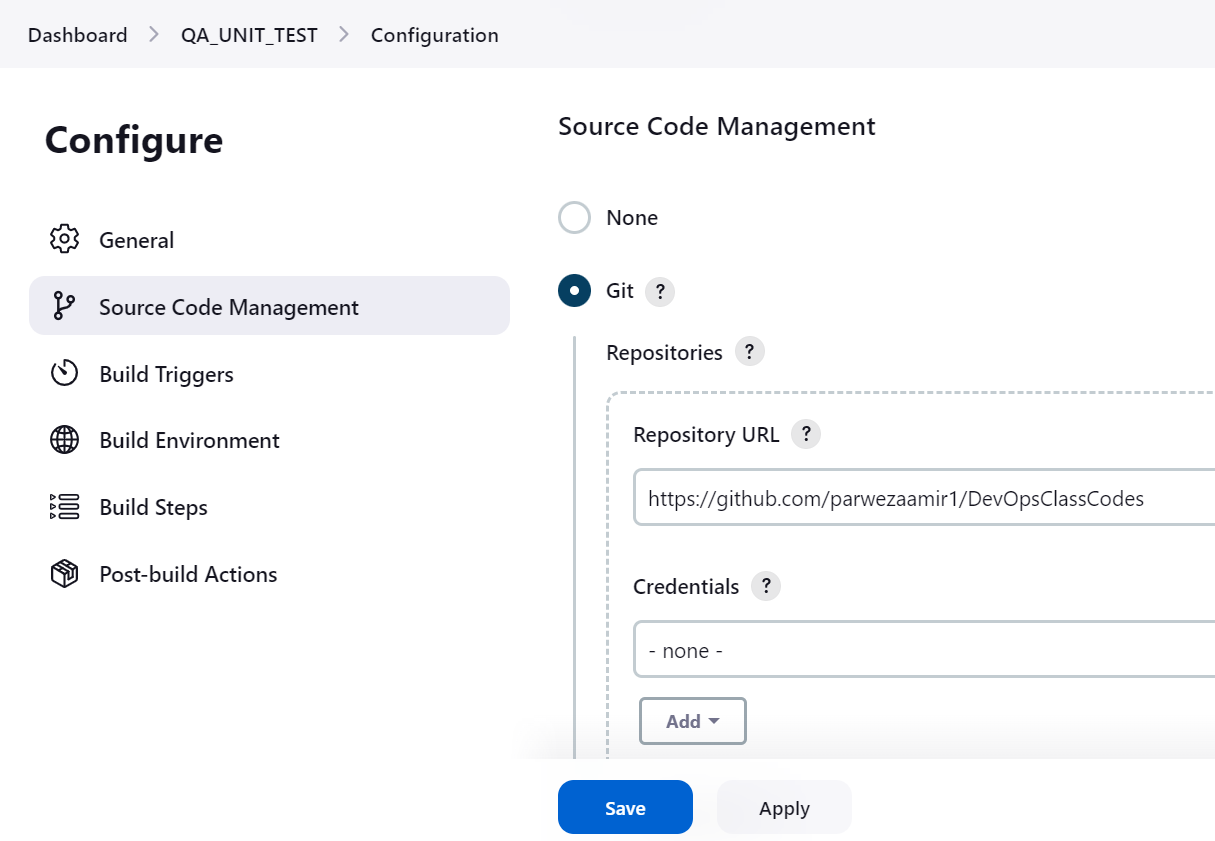
****

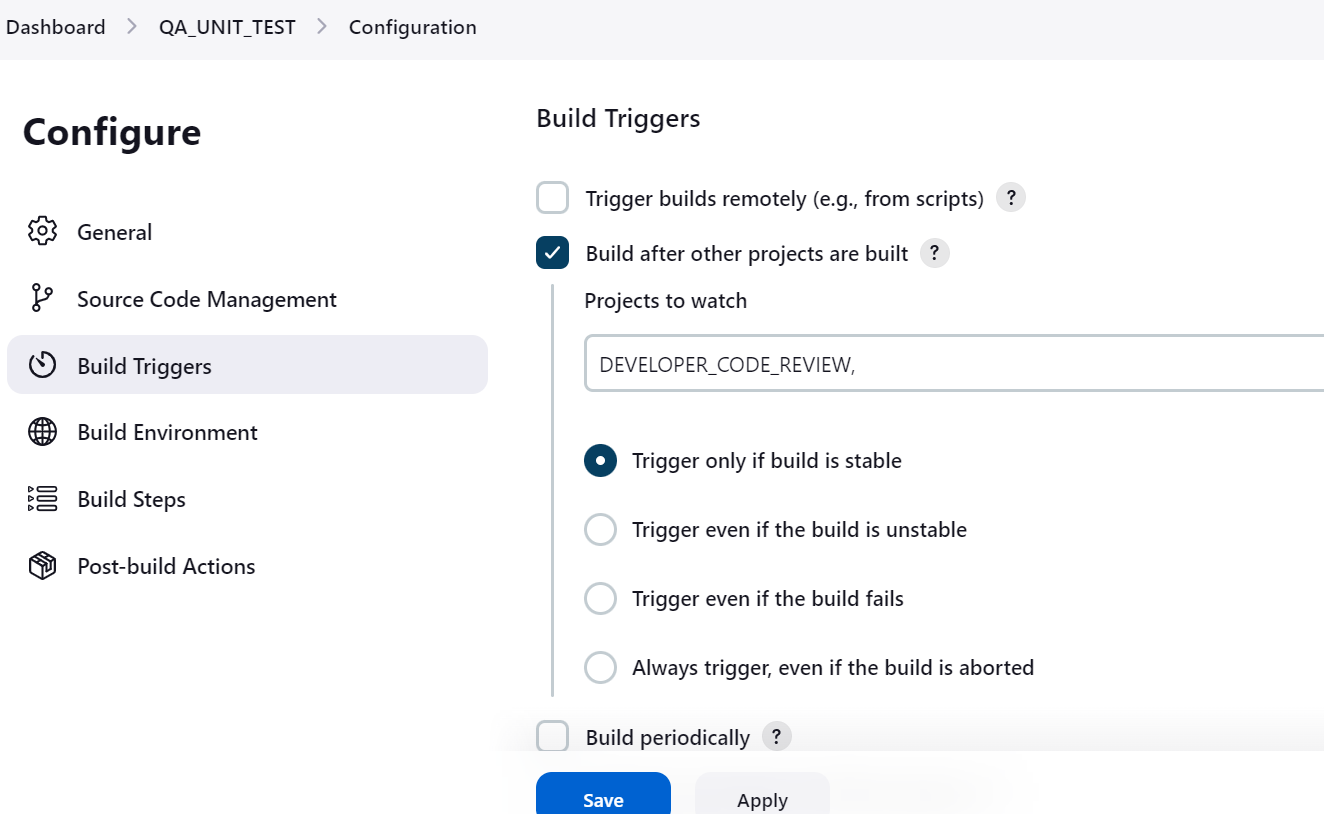
****

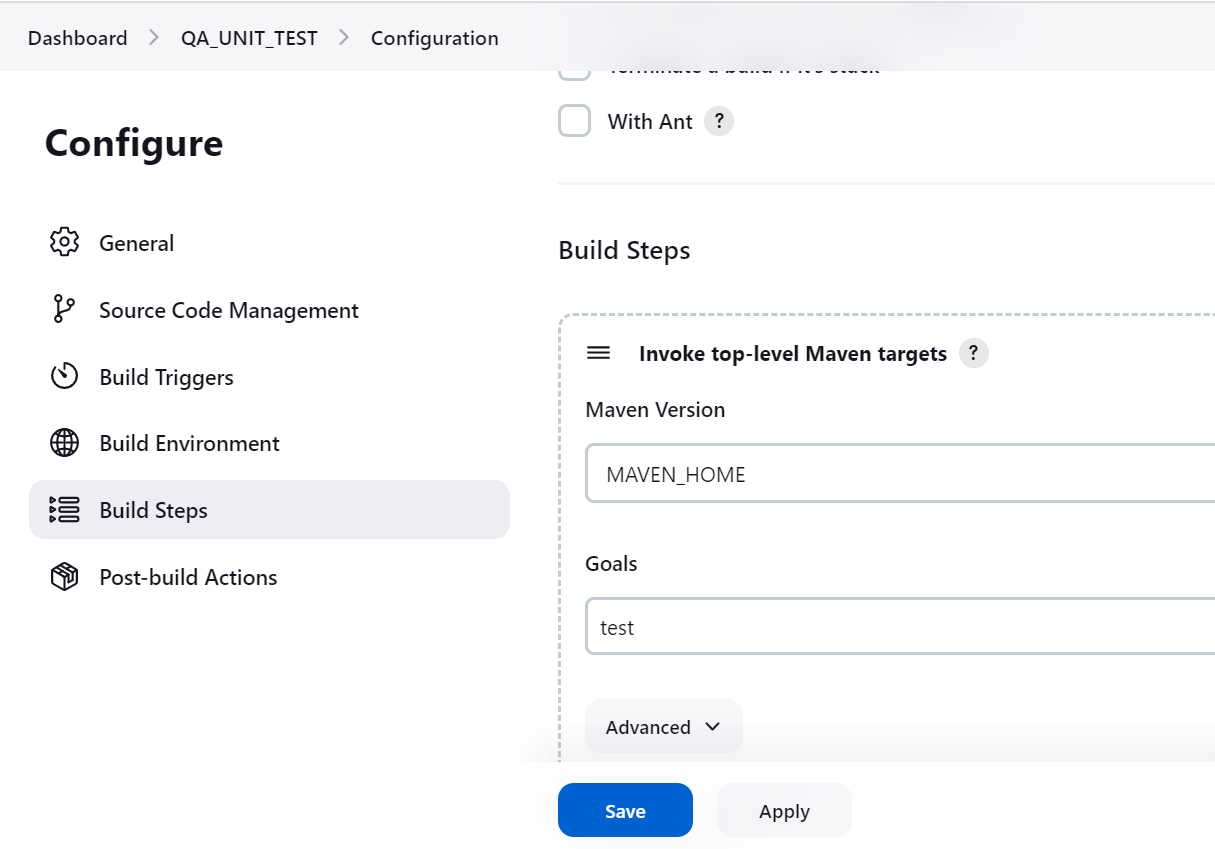
****

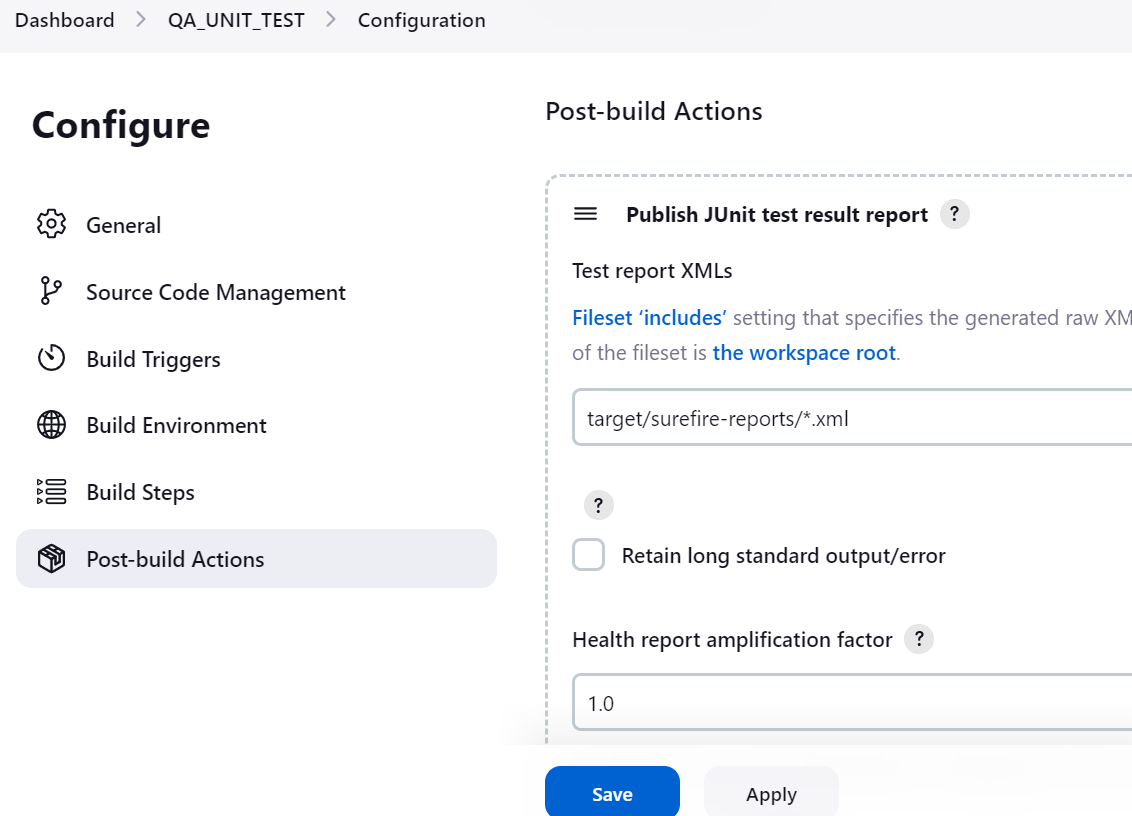
1. **QA\_UNIT\_TEST**

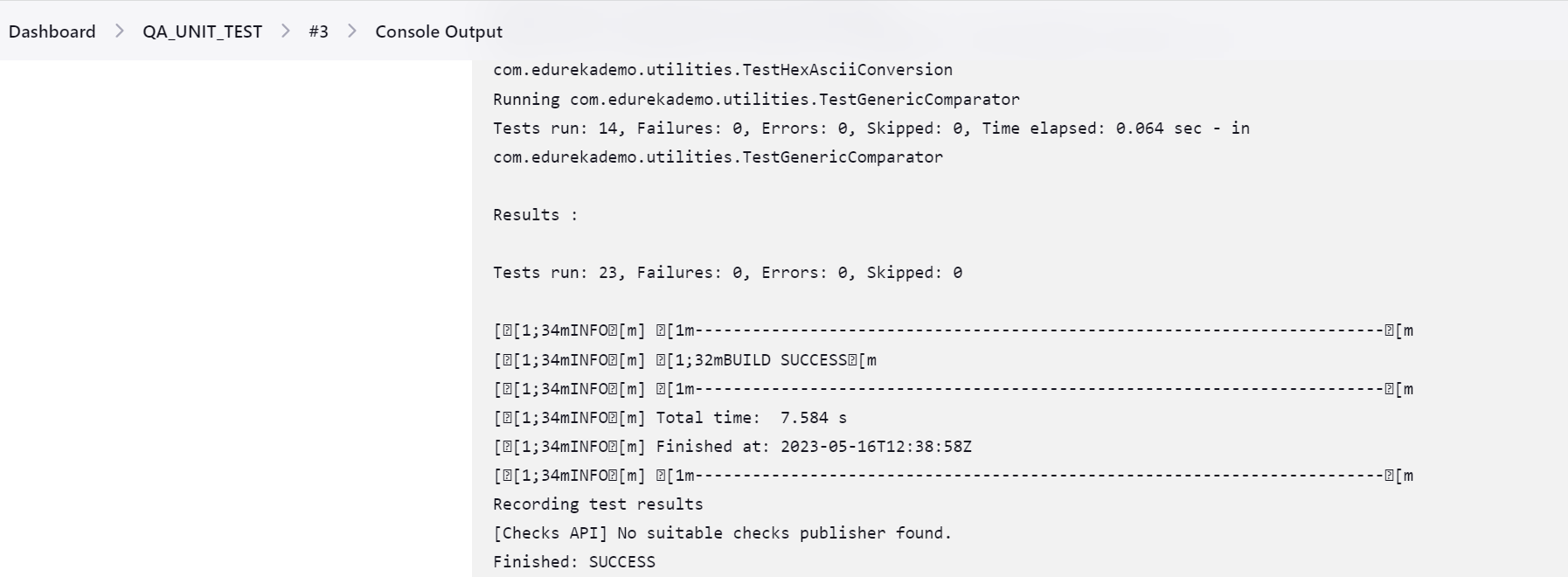
****

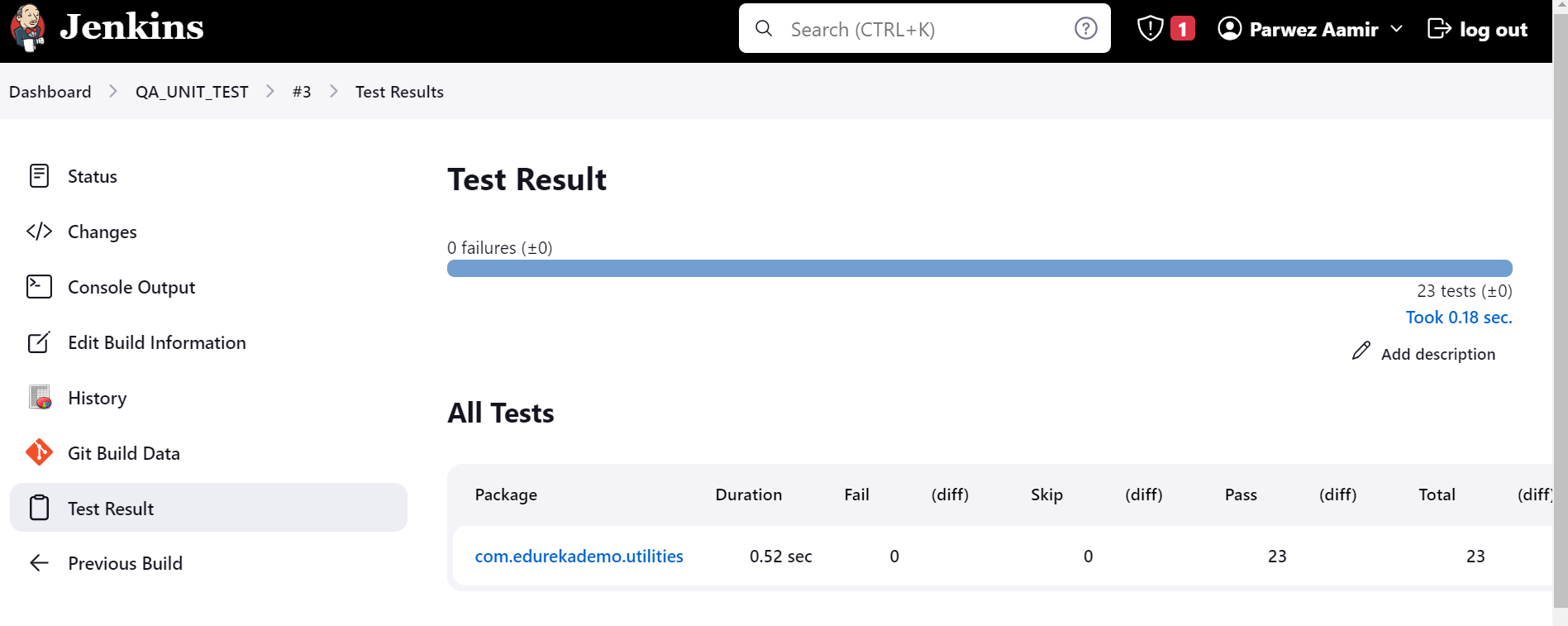
****

****

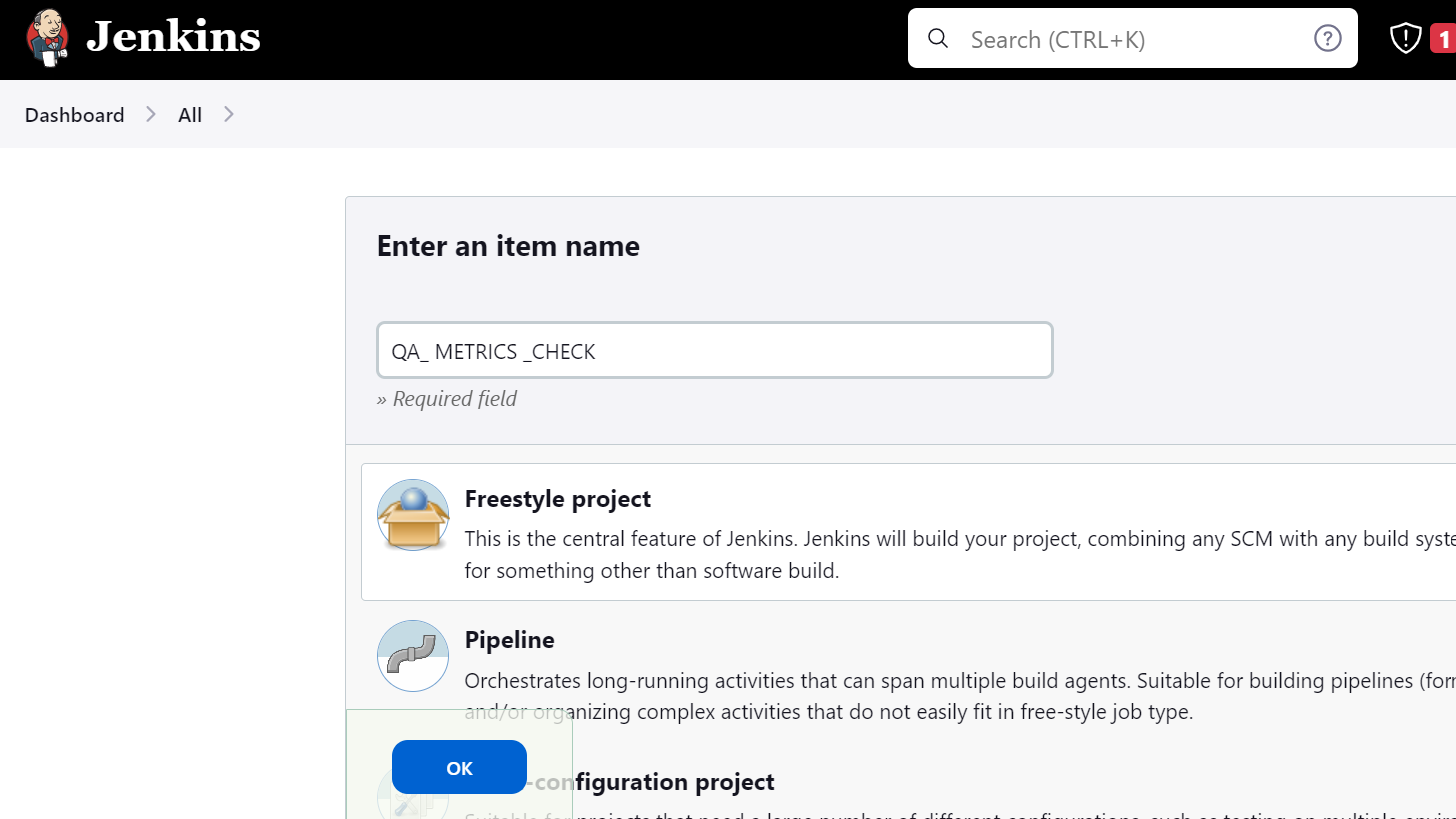
****

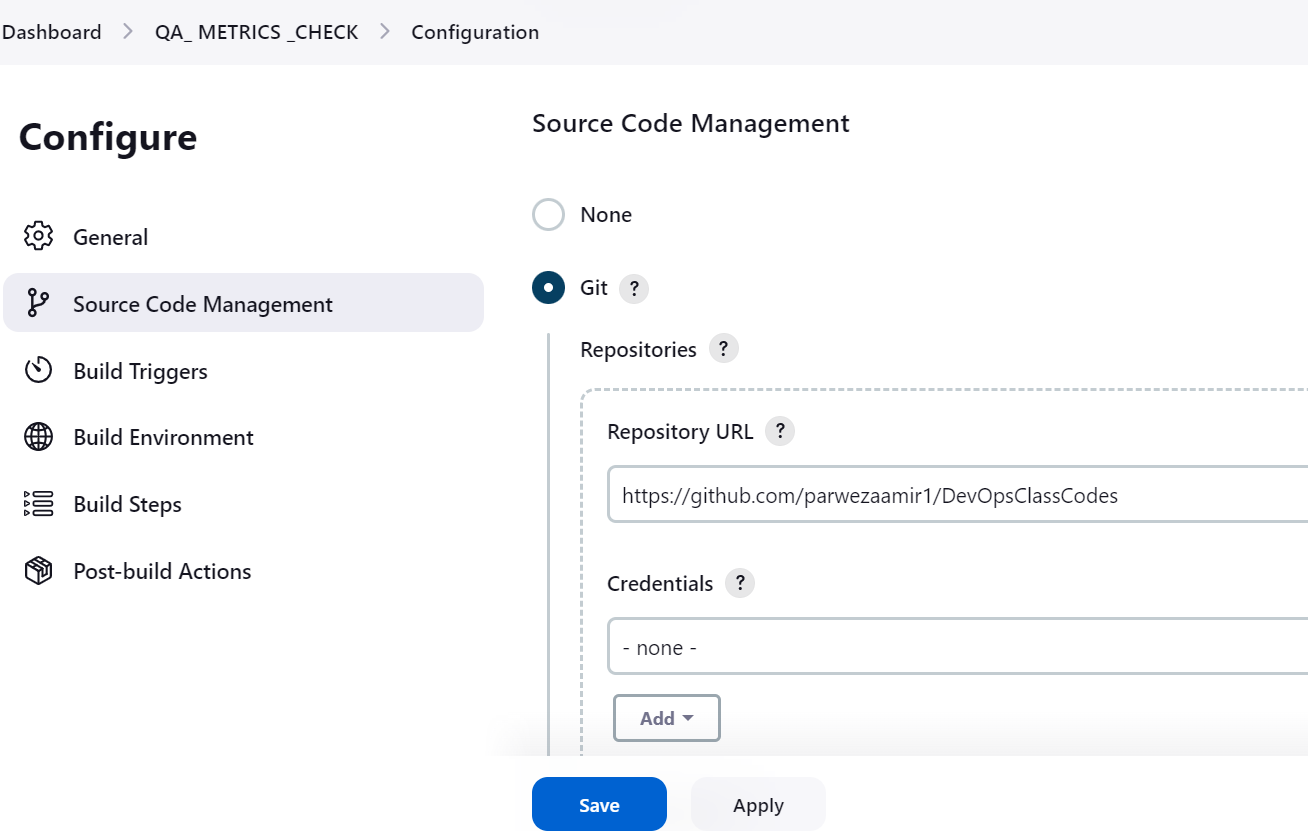
****

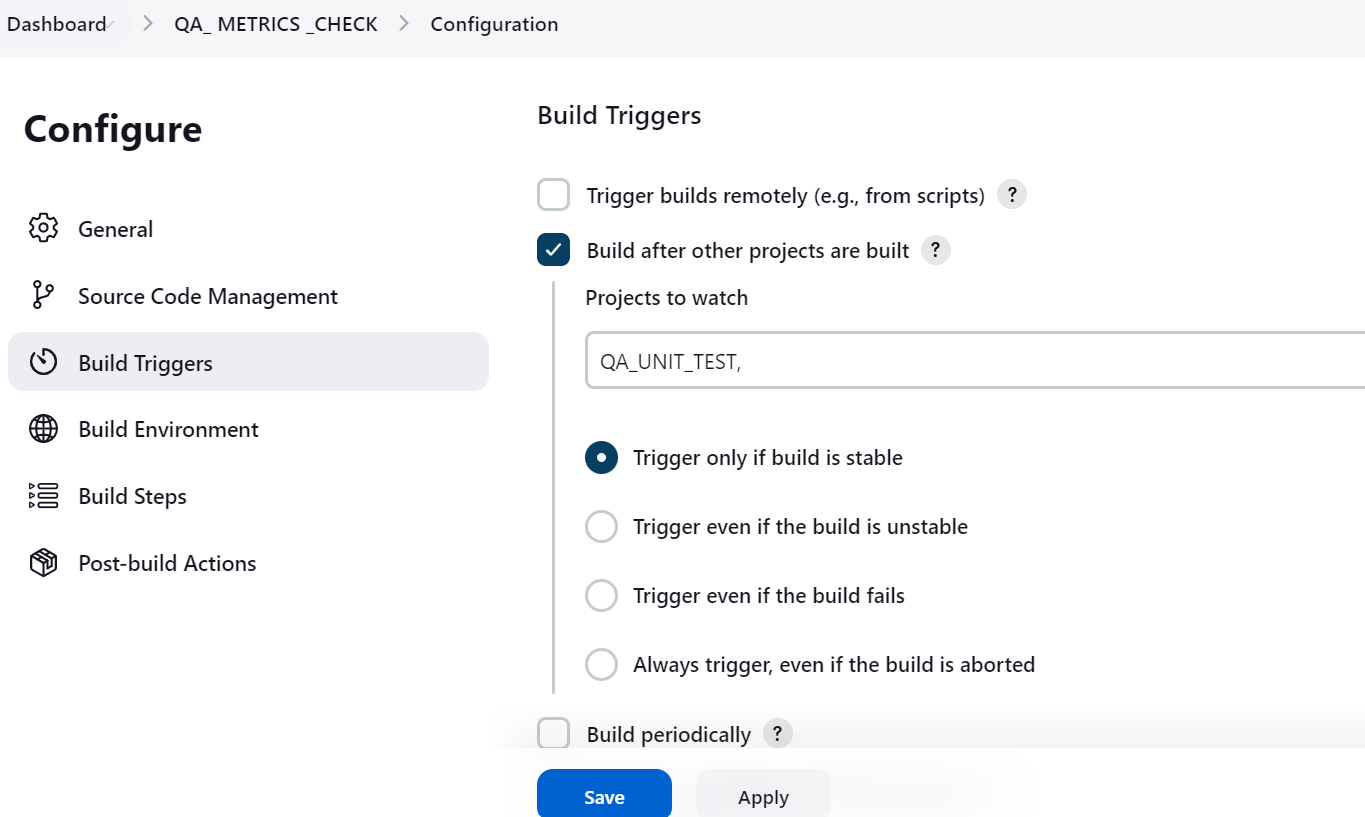
****

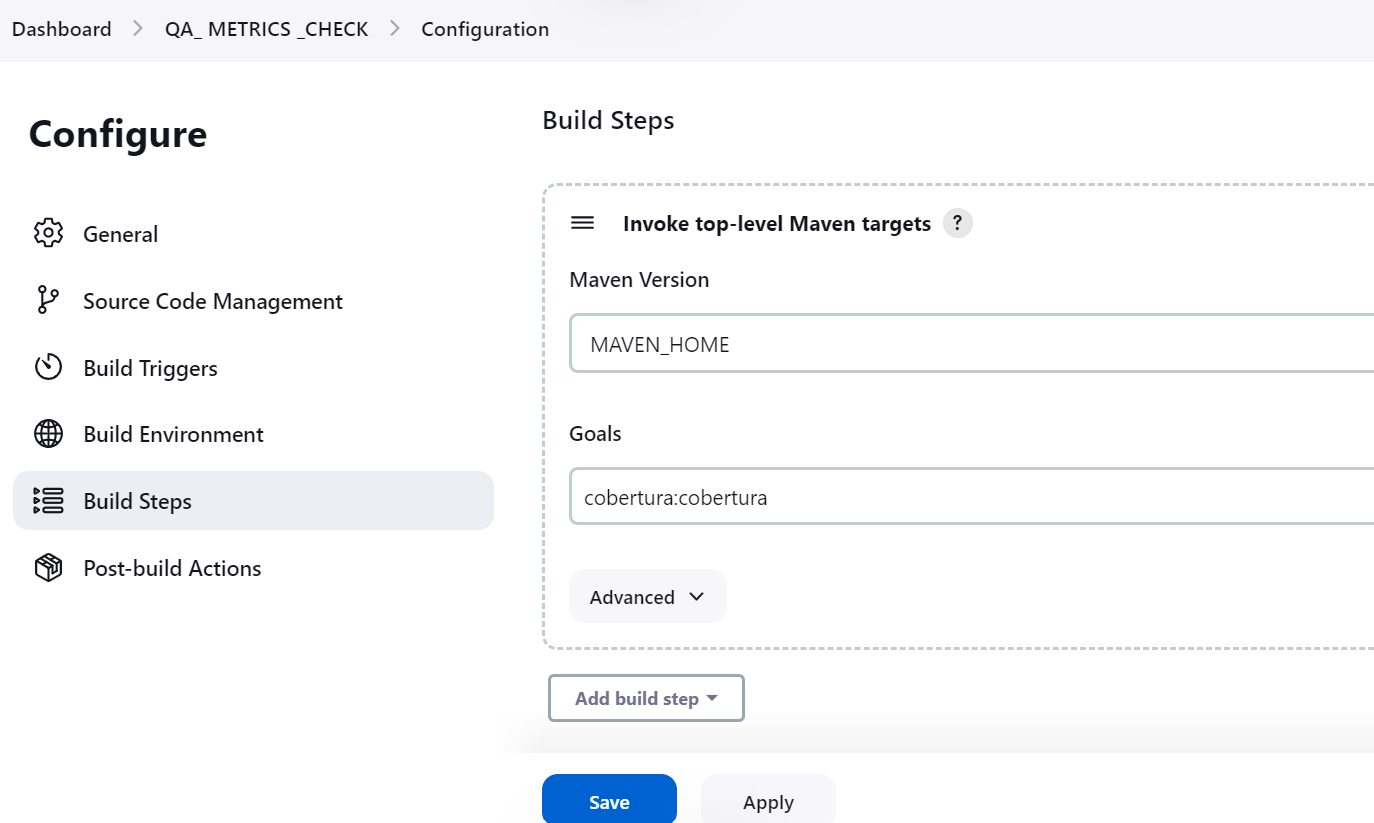
****

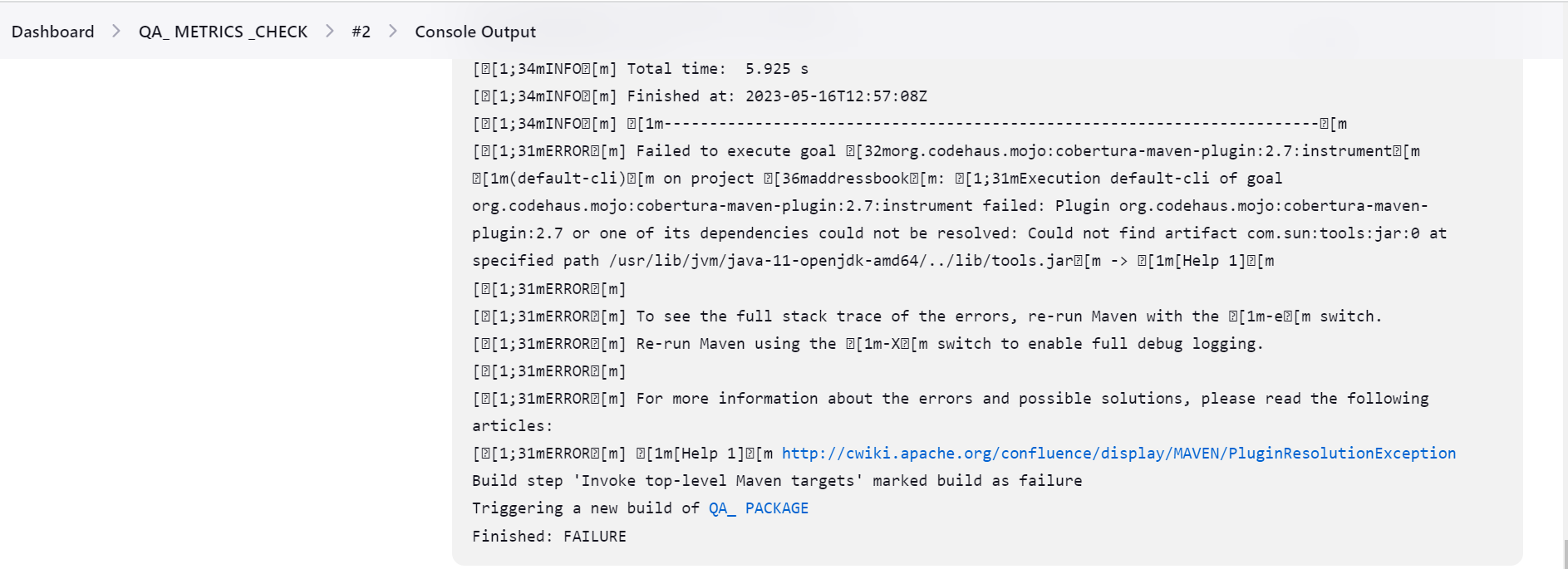
1. **QA\_METRICS\_CHECK**

****

****

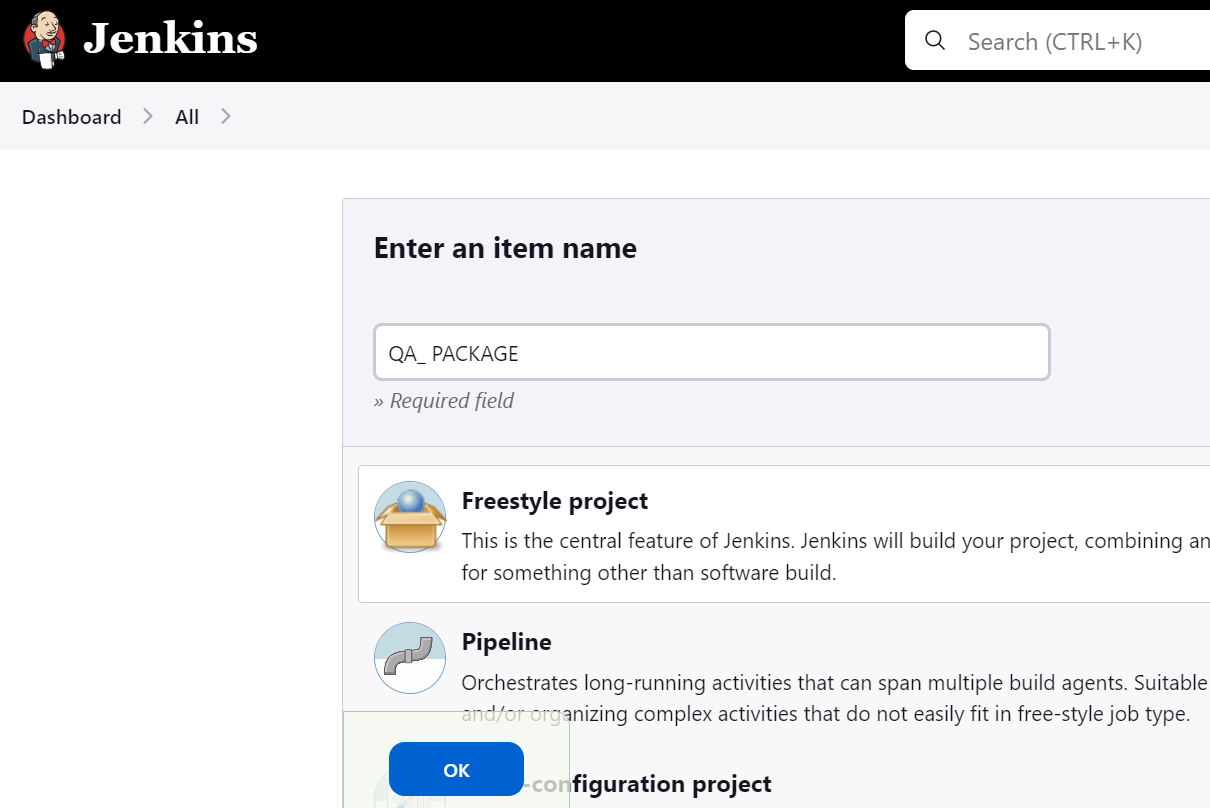
****

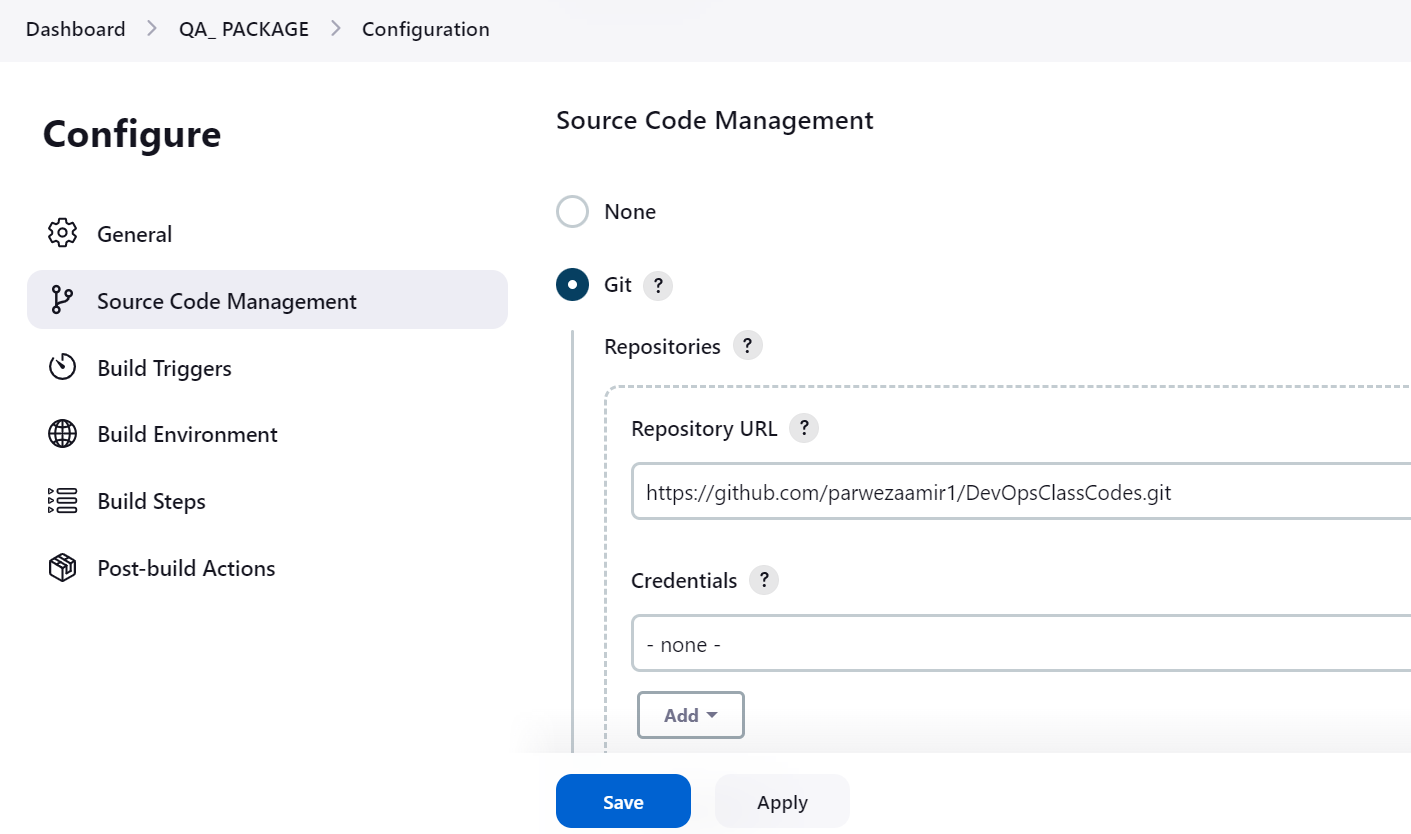
****

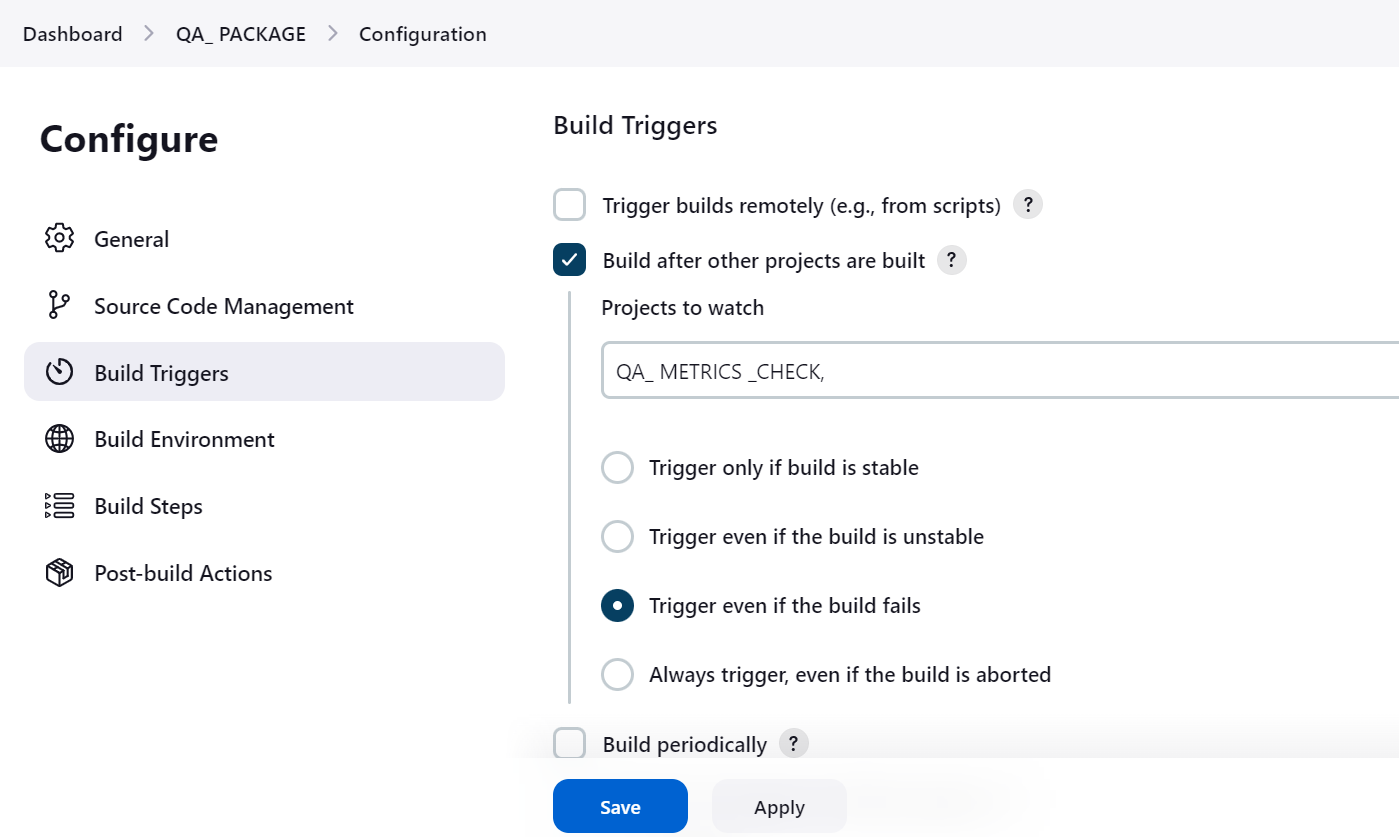
****

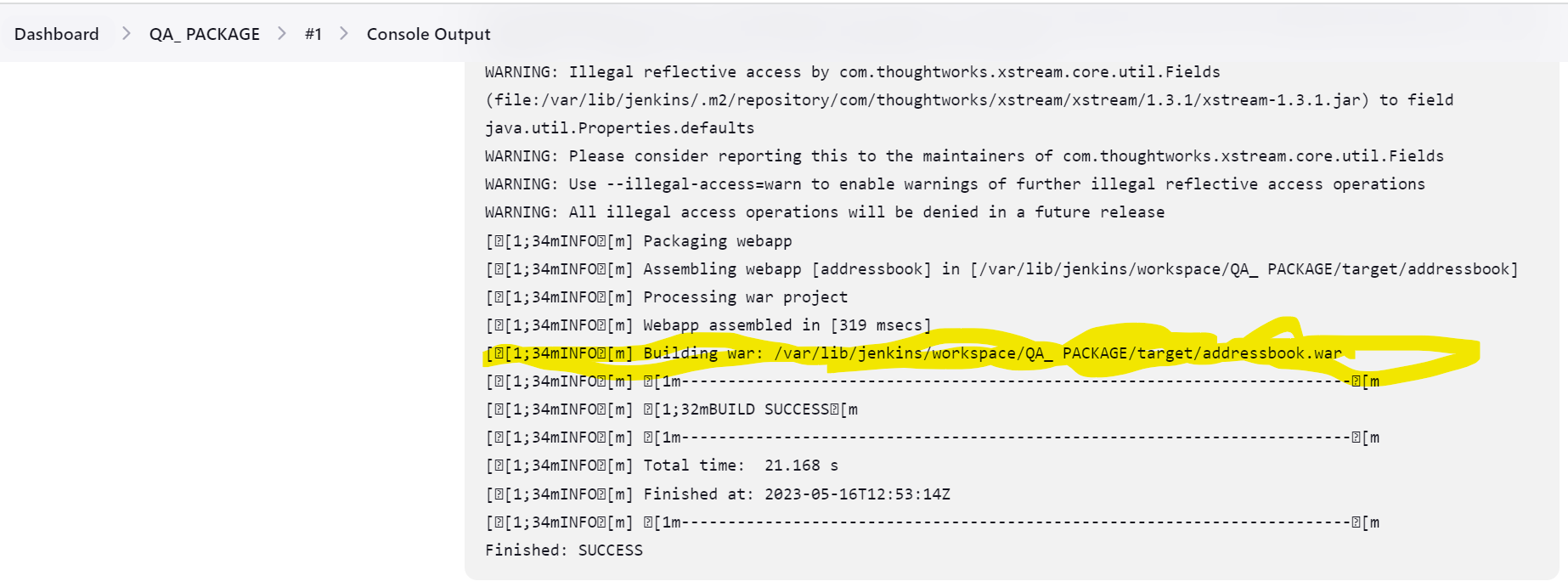
**Cobertura plugin has dependency on java which is not present in JAVA 11, but it works fine on JAVA 8. I have checked on my other machine.**

1. **QA\_PACKAGE**

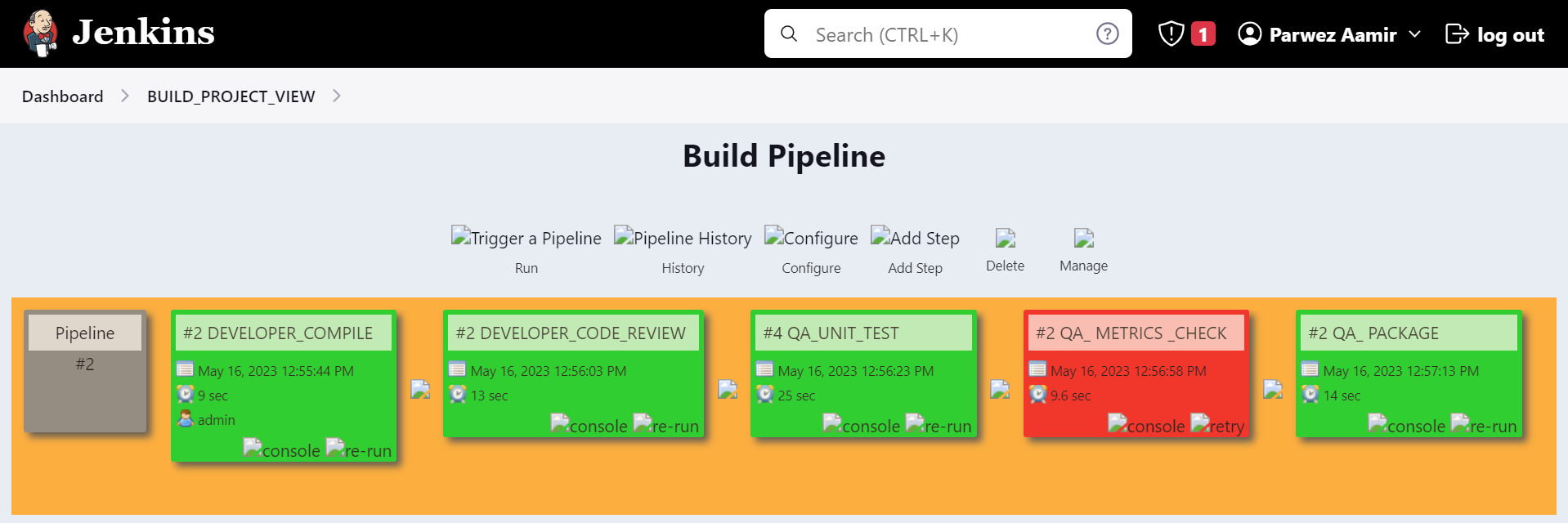
****

****

****

****

1. **BUILD\_PIPELINE\_VIEW**

****

**Here, QA\_METRICS\_CHECK got failed because cobertura plugin has dependency. It works fine on JAVA 8 but not on JAVA 11. I have JAVA 11 in my setup.**

1. **For Web test and load test**

To run web tests and load tests using Jenkins, you can leverage various plugins and tools available within the Jenkins ecosystem.

1. Install necessary plugins:
   * Jenkins Performance Plugin: to analyze and visualize load test results.
   * Jenkins Selenium Plugin: This plugin integrates Selenium WebDriver for web testing.
2. Create a Jenkins pipeline:
   * Define a Jenkins pipeline script (Jenkinsfile) in your project's version control repository or directly within the Jenkins pipeline editor.
   * Ensure that your Jenkins agent has the necessary tools and libraries installed for executing web tests and load tests.
3. Define stages for web tests:
   * Create a stage in your pipeline for web tests. Within this stage, you can execute functional tests using Selenium WebDriver.
   * Use appropriate programming languages and frameworks to write your tests and integrate them into the pipeline script.
4. Define stages for load tests:
   * Create a separate stage in your pipeline for load tests. There are multiple options to choose from for load testing tools:
     + Apache JMeter: It's a popular open-source load testing tool that can be integrated into your Jenkins pipeline.
     + Gatling: Another open-source load testing tool that provides excellent scalability and can be executed through Jenkins.
     + Other commercial load testing tools: If you have licensed tools like LoadRunner or BlazeMeter, they often provide Jenkins integrations.
   * Configure the load testing tool of your choice within the load testing stage and provide the necessary test scenarios, data, and configuration.
5. Integrate test reporting and analysis:
   * Configure the Jenkins Performance Plugin to analyze and visualize the load test results.
   * Configure test result notifications and notifications for failed tests or performance issues.
6. Run the pipeline:
   * Trigger the pipeline manually or schedule it to run automatically based on your requirements.
   * Jenkins will execute the web tests and load tests defined in the pipeline, generating test reports and performance analysis.