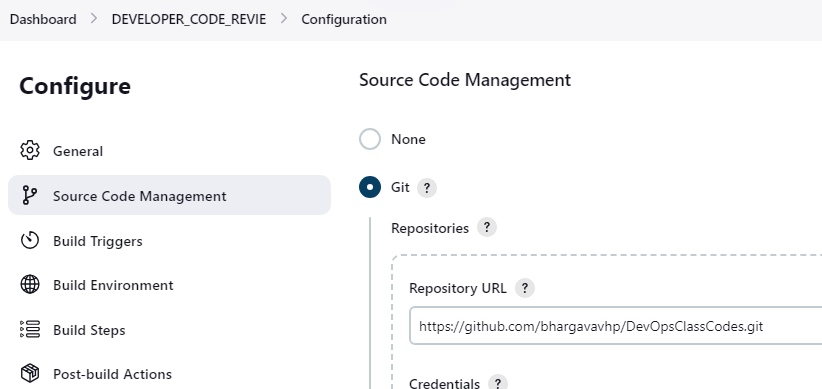
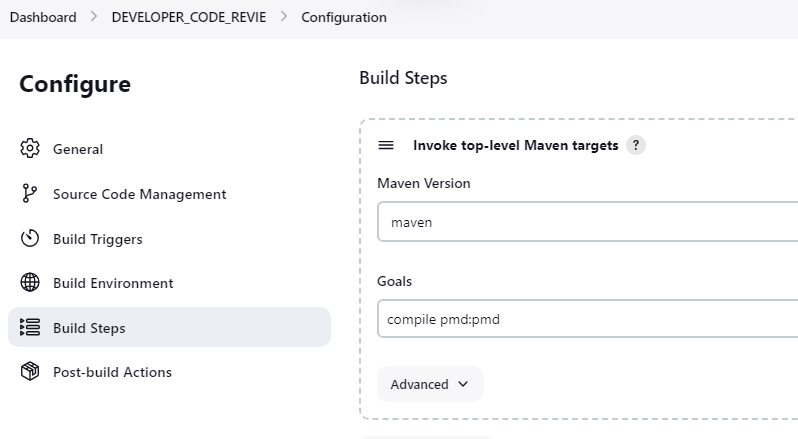
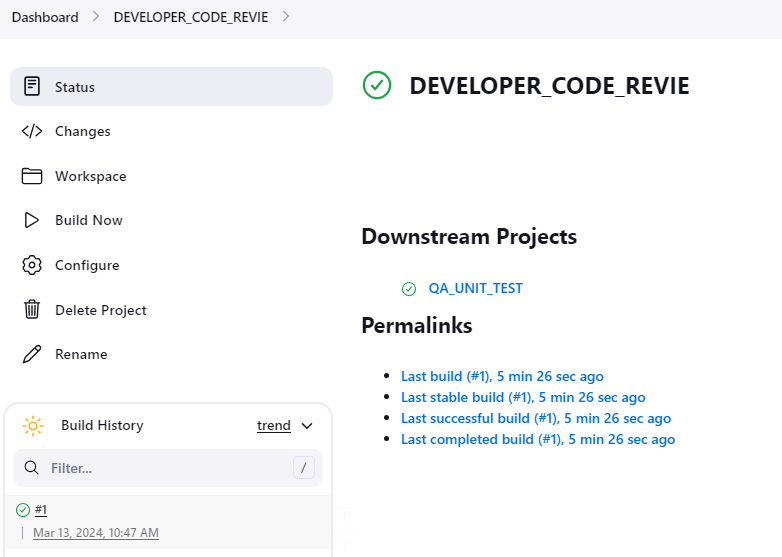
Name: Bhargava Varanasi

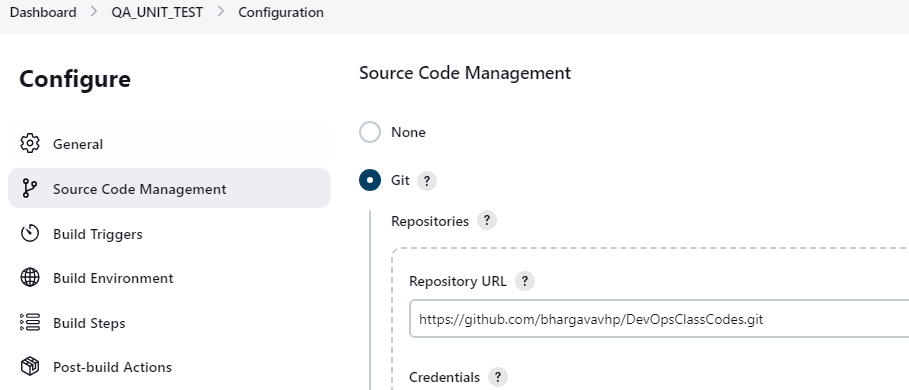
Module 4: Assignement

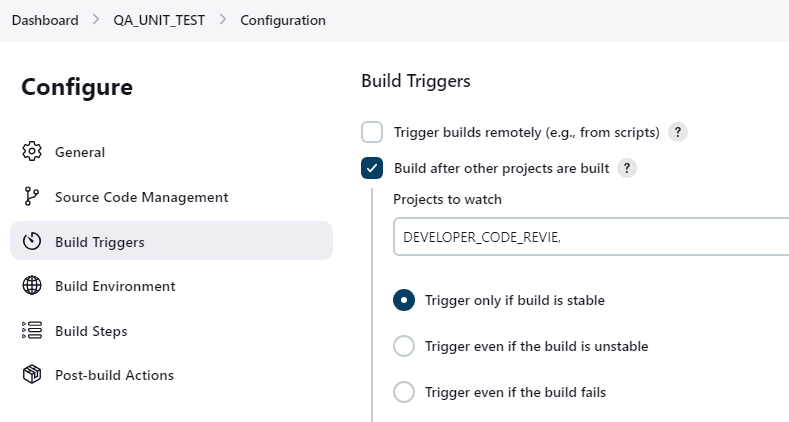
* Create a freestyle project with the name QA\_UNIT\_TEST in Jenkins that is driven from job DEVELOPER\_CODE\_REVIEW and performs unit testing Take ascreenshot of the console output showingasuccessful build of unit testing

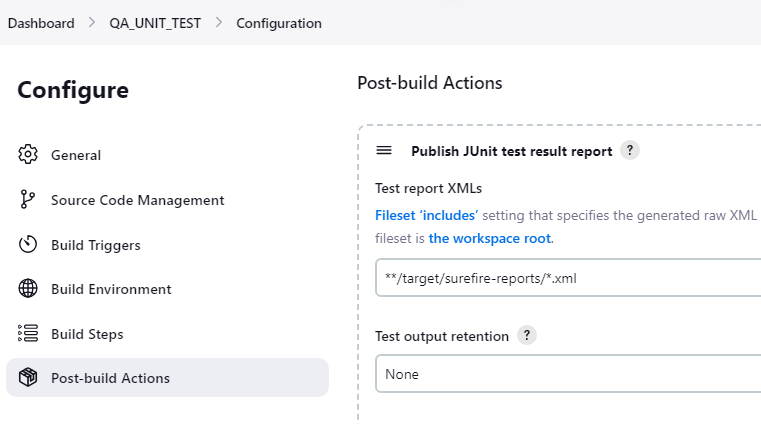
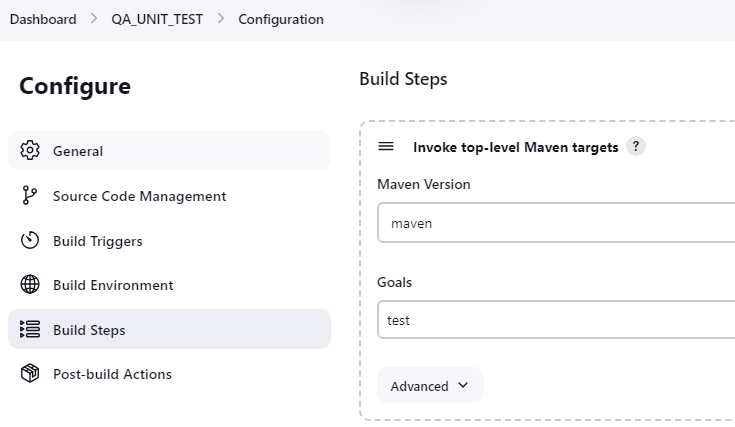


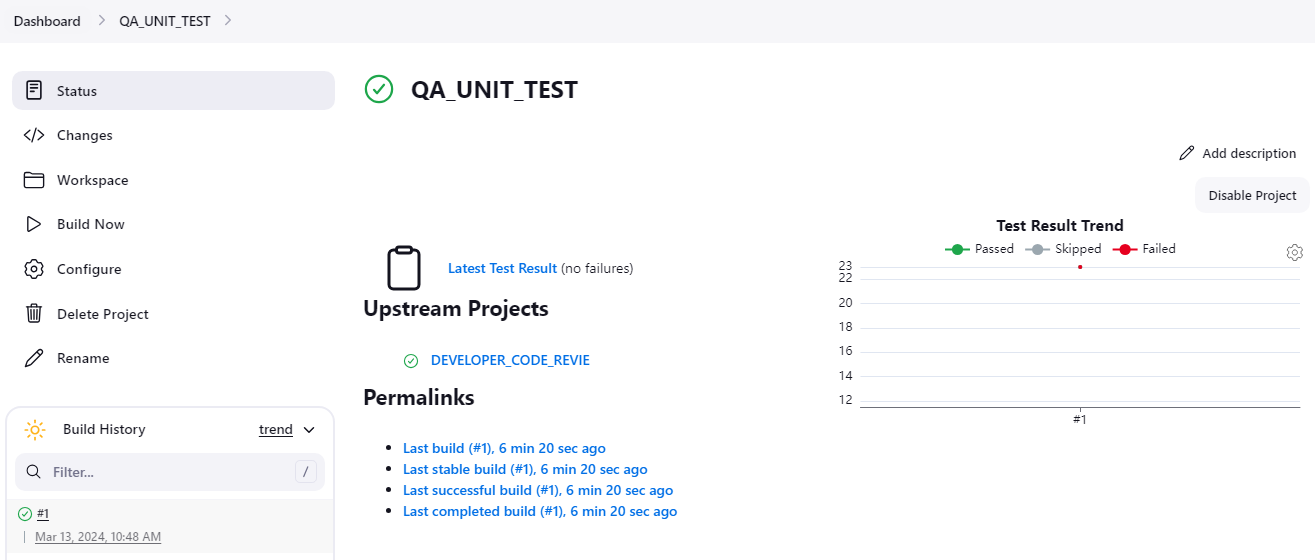






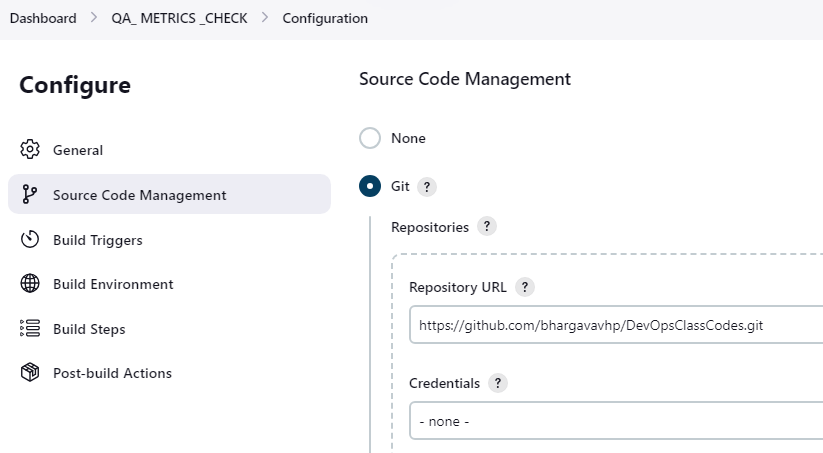


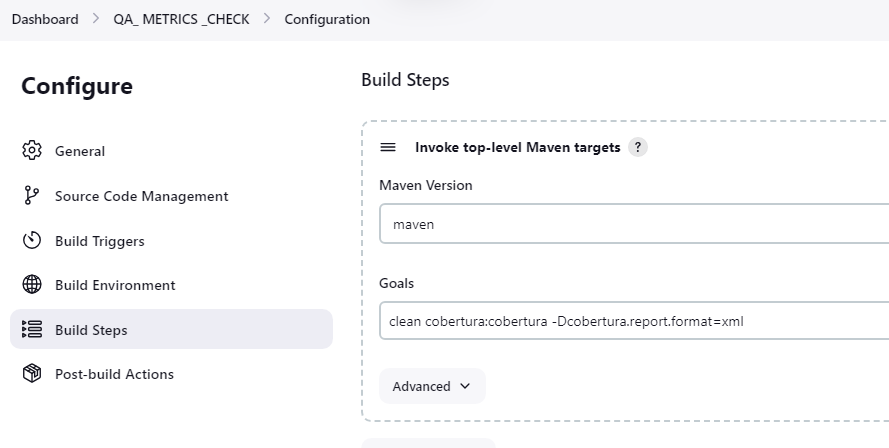


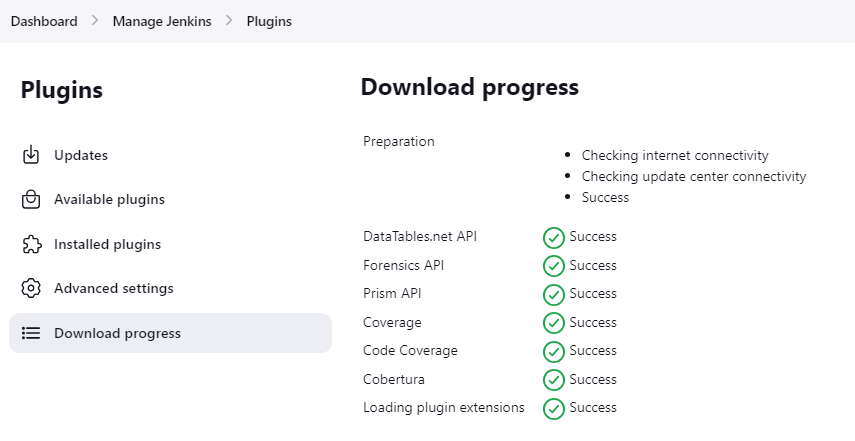


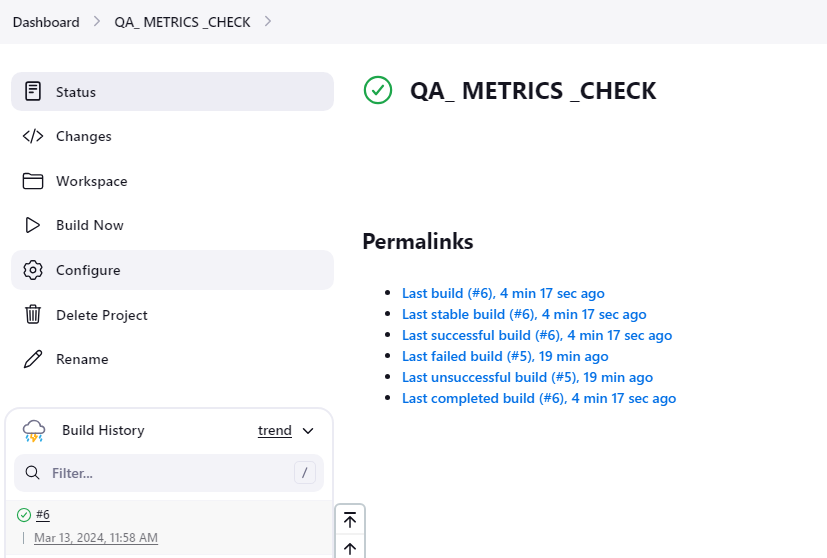
* Create a freestyle project with thename QA\_ METRICS \_CHECK in Jenkins to check the test cases.Make surethe Coberturaplugin is installed in JenkinsTake ascreenshot of the metrics from the dashboard of the project.

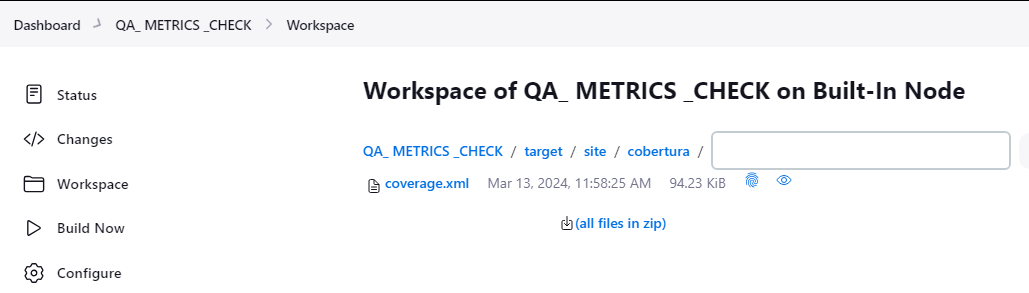
Note: configure java version 1.8 for cobertura compatibility



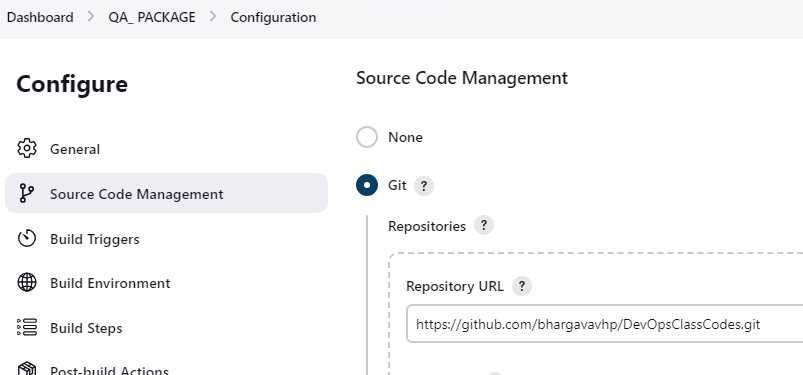


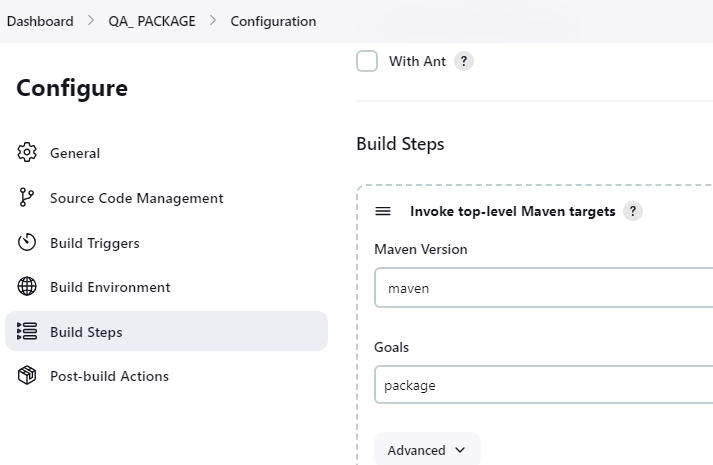


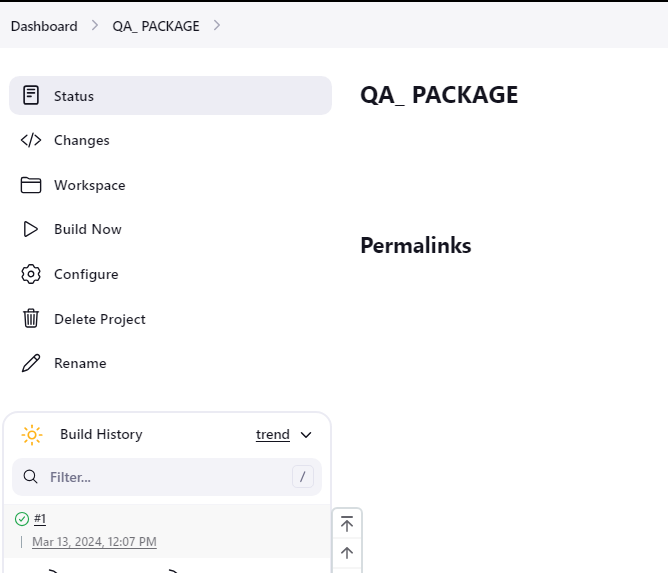


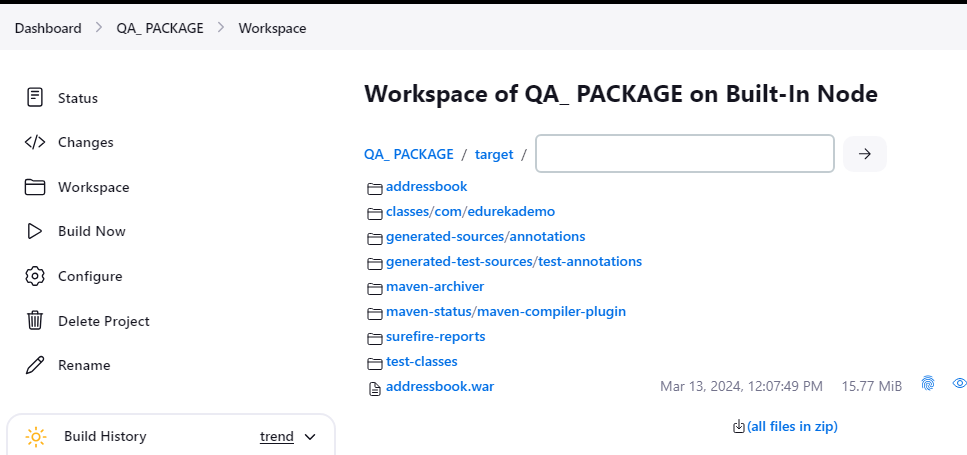


* Create a freestyle project with the name QA\_ PACKAGE in Jenkins to create an executable jar/war file.Take ascreenshot of the target folder created in the workspace.

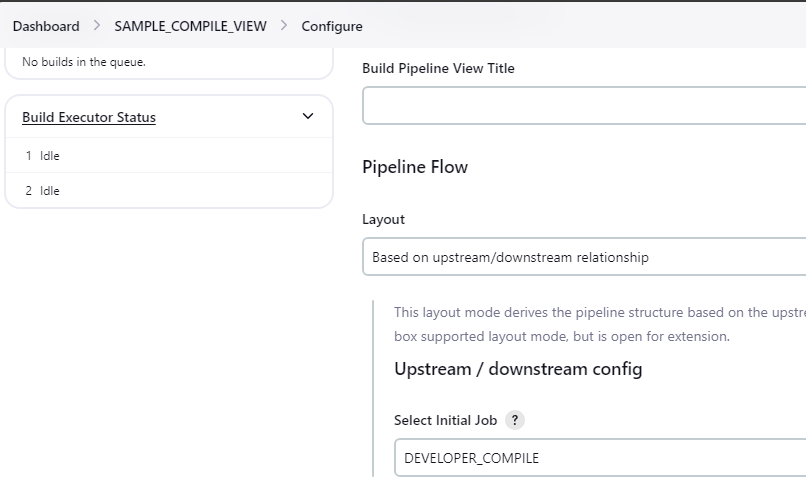


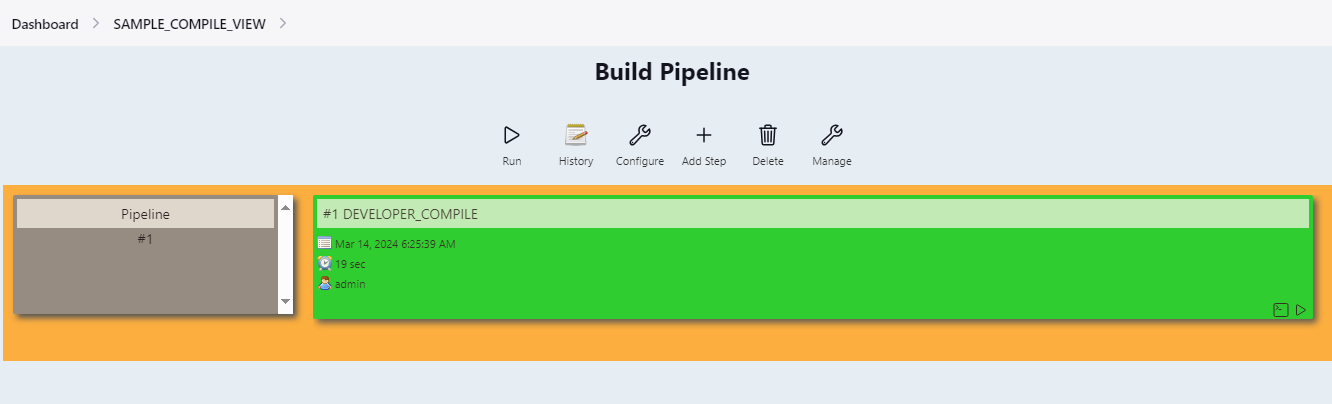


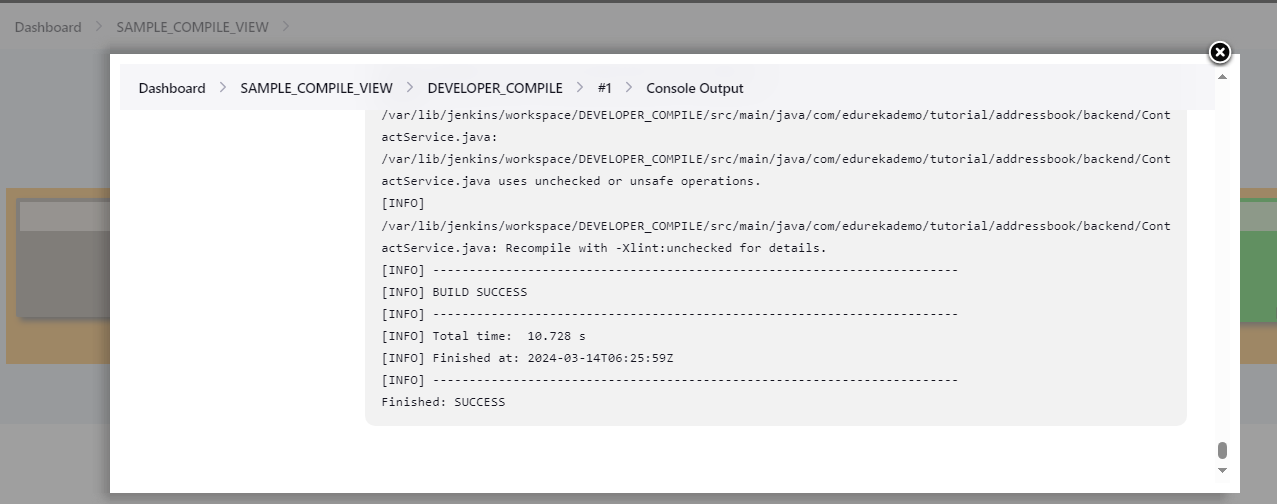




* Create a pipeline named SAMPLE\_COMPILE\_VIEW with Build Pipeline View option, selectDEVELOPER\_COMPILE project under layout section,and run the pipeline to check the console outputTake ascreenshot of the pipeline dashboard showing the status of the projects







* The pipelines can also be extended to running web tests and load tests. Explain how you would do the same using Jenkins?

Extending Jenkins pipelines to include web tests and load tests can be accomplished by integrating Jenkins with appropriate testing tools and plugins. Here's a general outline of how we can set up web tests and load tests in Jenkins pipelines:

1. Install Required Plugins: First, install Jenkins plugins that provide integration with web testing and load testing tools. Some commonly used plugins:

- Selenium Plugin: For running Selenium-based web tests.

- JUnit Plugin: For parsing JUnit XML test reports.

- Performance Plugin: For analyzing performance test results.

- Gatling Plugin: For integrating with Gatling, a popular load testing tool.

- JMeter Plugin: For integrating with Apache JMeter, another widely used load testing tool.

2. Configure Test Execution Steps: Define stages in Jenkins pipeline for running web tests and load tests. Each stage will include the necessary commands or scripts to execute the tests.

3. Configure Test Environments: Set up Jenkins agent nodes with the necessary software and dependencies for executing web tests and load tests. Ensure that agents have browsers, WebDriver, and any other tools required for web testing. For load testing, install Gatling or JMeter on the agent nodes.

4. Triggering Test Execution: Configure triggers to start the pipeline when code changes are detected.We can also schedule periodic executions for running tests at specific intervals.

5. Monitoring and Reporting: Use Jenkins plugins to monitor test execution and analyze test results. Plugins like the Performance Plugin, JUnit Plugin, and Gatling Plugin provide detailed reports and metrics for web tests and load tests. Configure the pipeline to publish test results and performance metrics as artifacts or to external systems for further analysis.