

EXP NO: 6	Run regression tests using Maven Build pipeline in Azure
DATE:	

AIM: -

To Run regression tests using Maven Build pipeline in Azure.

ALGORITHM / PROCEDURE: -

1. Create or Modify Azure Pipeline: If you don't have an existing pipeline, create a new one in Azure DevOps as described earlier. If you already have a pipeline, modify it to include the test steps.
2. Choose Repository: Select the repository that contains your Maven project with the tests.
3. Define Pipeline Template: Choose Maven as your pipeline template or use the Starter Pipeline and manually define the pipeline.
4. Update azure-pipelines.yml: Add Maven commands to run the regression tests after building the project. The regression tests are usually executed by Maven through mvn test or mvn verify.
5. Add Maven Test Goals: In the azure-pipelines.yml, configure the goals to include test or verify to run the tests.
6. Configure Test Results: Optionally, you can publish test results to Azure DevOps for reporting purposes.
7. Save and Run the Pipeline: Once the pipeline is configured, save it and run it to execute the regression tests along with the build.

CODING: -

trigger:

- main # Adjust this to your default branch, like 'master' or 'develop'

pool:

vmImage: 'ubuntu-latest' # You can change to 'windows-latest' if required

steps:

- task: MavenAuthenticate@0 # Authenticate with your Maven feed (if necessary)

inputs:

mavenFeed: 'your-feed-name'

- task: Maven@3 # Maven task to run build and tests

inputs:

mavenPomFile: 'pom.xml' # Path to your Maven project file

options: '-X' # Optional: Enable verbose output

goals: 'clean verify' # Run clean and verify to compile, test, and package

javaHomeOption: 'JDKVersion' # Choose JDK version (e.g., '1.8', '11')

jdkVersionOption: '1.8'

mavenVersionOption: 'Default'

- task: PublishTestResults@2 # Publish test results to Azure DevOps

inputs:

testResultsFiles: '**/target/test-*.xml' # Path to test result files

testRunTitle: 'Maven Regression Tests' # Title for test run

- task: PublishBuildArtifacts@1 # Optionally publish build artifacts

inputs:

PathtoPublish: '\$(Build.ArtifactStagingDirectory)'

ArtifactName: 'drop'

publishLocation: 'Container'

OUTPUT: -

The screenshot displays the Azure DevOps interface. On the left, a sidebar shows the pipeline run 'Jobs in run #20230...' with a list of jobs: 'Load Test' (6m 15s), 'Initialize job' (1s), 'Checkout ALT - ...' (1s), 'AzureLoadT...' (6m 10s), 'PublishPipelineA...' (2s), 'Post-job: Check...' (<1s), and 'Finalize Job' (<1s). The 'AzureLoadT...' job is selected, showing its details on the right. The details pane is titled 'AzureLoadTest' and contains a log of the test run. The log shows the test plan upload, test run creation, and completion. A red box highlights the test results section, which includes a summary of the test run and a table of test criteria.

```
10 Uploaded test plan for the test
11 Creating and running a testRun for the test
12 View the load test run in progress at: https://portal.azure.com/#blade/Microsoft\_Azure...
13 TestRun completed
14
15 -----Summary -----
16 TestRun start time: Wed Aug 09 2023 07:37:37 GMT+0000 (Coordinated Universal Time)
17 TestRun end time: Wed Aug 09 2023 07:43:22 GMT+0000 (Coordinated Universal Time)
18 Virtual Users: 10
19 TestStatus: DONE
20
21 -----Test Criteria -----
22 Results :1 Pass 0 Fail
23
24 Criteria :Actual Value Result
25 percentage(error) > 20 6.91 PASSED
26
27 -----Client-side metrics-----
28
29
30 Homepage
31 response time : avg=2497ms min=35ms med=150ms max=62947ms p(90)=604ms p(95)
32 requests per sec : avg=5
33 total requests : 564
```

RESULT: -

Thus we successfully Run regression tests using Maven Build pipeline in Azure.