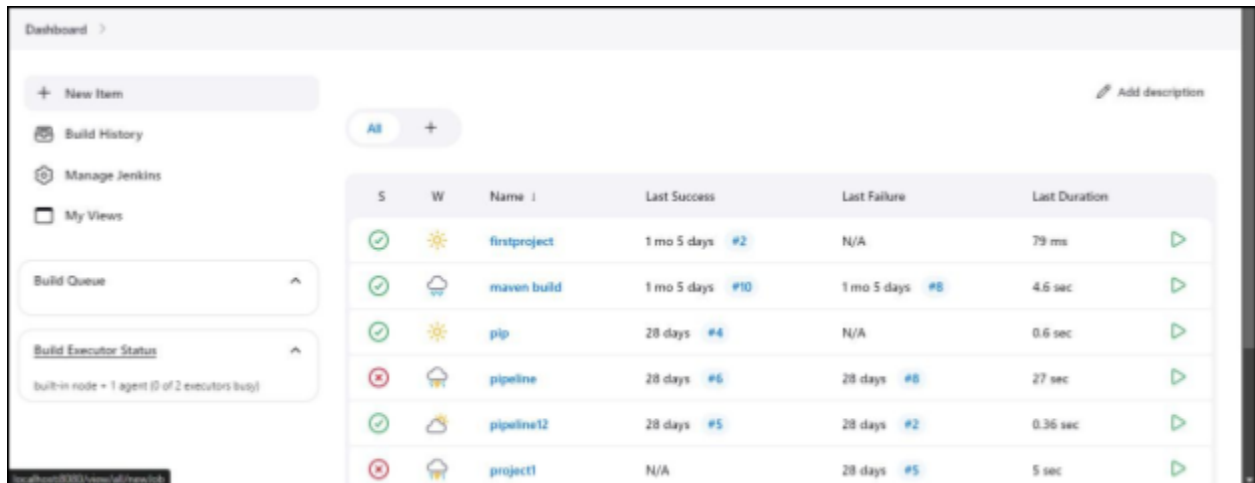


Experiment No. 7

Procedure :

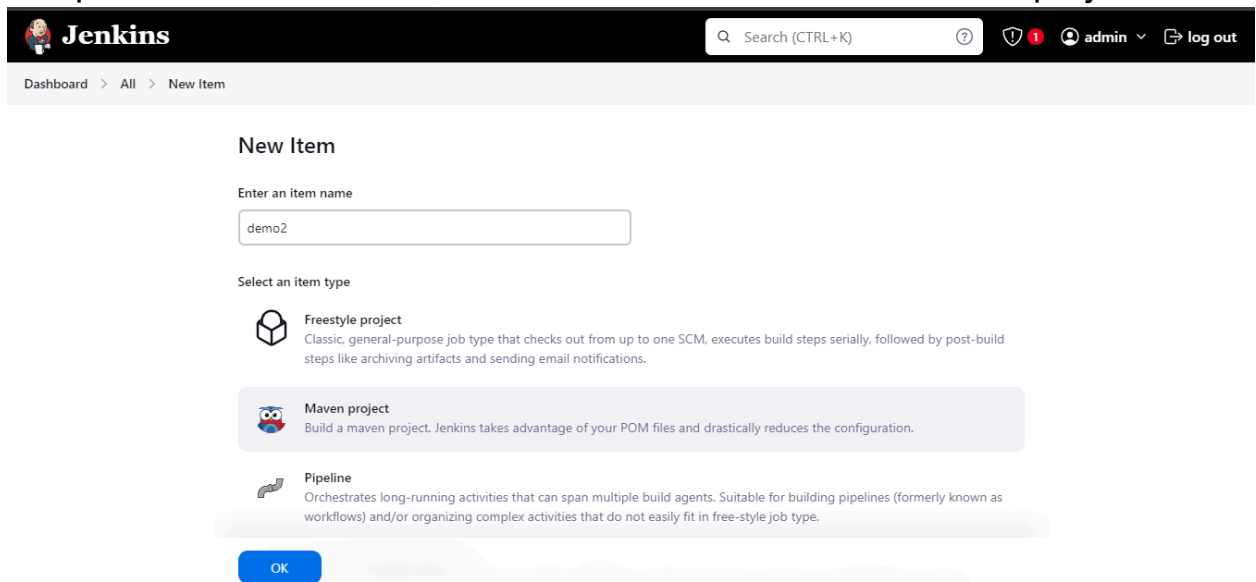
Step 1 : Visit Jenkins dashboard.



The screenshot shows the Jenkins Dashboard with a table of build items. The table has columns for Status (S), Weather icon (W), Name, Last Success, Last Failure, and Last Duration. The items listed are firstproject, maven build, pip, pipeline, pipeline12, and project1.

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	firstproject	1 mo 5 days #2	N/A	79 ms
✓	☁	maven build	1 mo 5 days #10	1 mo 5 days #8	4.6 sec
✓	☀	pip	28 days #4	N/A	0.6 sec
✗	☁	pipeline	28 days #6	28 days #8	27 sec
✓	☁	pipeline12	28 days #5	28 days #2	0.36 sec
✗	☁	project1	N/A	28 days #5	5 sec

Step 2 : Create new item, enter item name and select maven project.



The screenshot shows the Jenkins 'New Item' form. The 'Enter an item name' field contains 'demo2'. Under 'Select an item type', the 'Maven project' option is selected, which is highlighted in a light blue box. The 'Freestyle project' and 'Pipeline' options are also visible. At the bottom, there is an 'OK' button.

Jenkins Search (CTRL+K) admin log out

Dashboard > All > New Item

New Item

Enter an item name

demo2

Select an item type

- Freestyle project**
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Maven project**
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
- Pipeline**
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

OK

Step 3 : In project configuration, enter GitHub repository.

The screenshot shows the Jenkins Configuration page for a project named 'demo1'. The 'Source Code Management' tab is selected. Under the 'Git' section, the 'Repository URL' is set to 'https://github.com/jglick/simple-maven-project-with-tests.git'. The 'Credentials' dropdown is set to '- none -'. There are buttons for '+ Add' and 'Advanced'. At the bottom, there are 'Save' and 'Apply' buttons.

Dashboard > demo1 > Configuration

Source Code Management

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Pre Steps
- Build
- Post Steps
- Build Settings
- Post-build Actions

None

Git ?

Repositories ?

Repository URL ?

https://github.com/jglick/simple-maven-project-with-tests.git

Credentials ?

- none -

+ Add

Advanced

Save Apply

Step 4 : enter pom.xml and goals as test.

The screenshot shows the Jenkins Configuration page for a project named 'Selenium_Build'. The 'Build' section is selected. The 'Root POM' is set to 'pom.xml'. The 'Goals and options' field is set to 'test'. There is an 'Advanced' button at the bottom.

Dashboard > Selenium_Build > Configuration

Build

Root POM ?

pom.xml

Goals and options ?


test

Advanced

Step 5 : JUnit version and scope can be seen in pom.xml

```
4.0.0 test simple-maven-project-with-tests 1.0-SNAPSHOT jar MIT License http://opensource.org/licenses/MIT org.apache.maven.plugins maven-surefire-plugin 2.18.1 junit junit 4.13.1 test UTF-8 1.8 1.8
This XML file does not appear to have any style information associated with it. The document tree is shown below:
<?xml version="1.0" encoding="UTF-8" ?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>test</groupId>
  <artifactId>simple-maven-project-with-tests</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <license>
    <license>
      <name>MIT License</name>
      <url>http://opensource.org/licenses/MIT</url>
    </license>
  </license>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-surefire-plugin</artifactId>
        <version>2.18.1</version>
      </plugin>
    </plugins>
  </build>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.13.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

Step 6 : Different outputs

**Jenkins**

Search (CTRL+K)

admin log out

Dashboard > demo >

Status

</> Changes

Build Now

Configure

Delete Pipeline

Stages

Rename

Pipeline Syntax

demo

Last Successful Artifacts

simple-maven-project-with-tests-1.0-SNAPSHOT.jar

2.04 KIB

view

Latest Test Result (no failures)

Permalinks

- Last build (#4), 2 min 15 sec ago
- Last stable build (#4), 2 min 15 sec ago
- Last successful build (#4), 2 min 15 sec ago
- Last failed build (#3), 9 min 24 sec ago
- Last unsuccessful build (#3), 9 min 24 sec ago
- Last completed build (#4), 2 min 15 sec ago

Build History

trend

Filter...

#4

Sep 18, 2024, 9:16 AM

Test R

#4

Passed 6

Skipped 1

Failed 0

Running test.OtherTest

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 sec - in test.OtherTest

Running test.SomeTest

Tests run: 6, Failures: 0, Errors: 0, Skipped: 1, Time elapsed: 0 sec - in test.SomeTest

Results :

- Status
- </> Changes
- Workspace
- ▶ Build Now
- ⚙️ Configure
- 🗑️ Delete Maven project
- 📄 Modules
- ✎️ Rename

Build History trend

Filter...

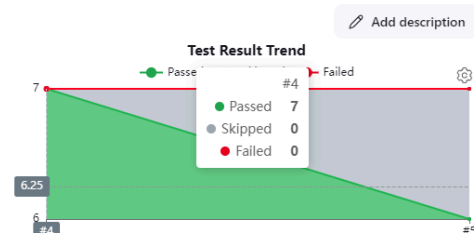
#5
Sep 18, 2024, 9:18 AM

Maven project demo1

Latest Test Result (no failures)

Permalinks

- Last build (#5), 9.8 sec ago
- Last stable build (#5), 9.8 sec ago
- Last successful build (#5), 9.8 sec ago
- Last failed build (#3), 5 min 23 sec ago
- Last unsuccessful build (#3), 5 min 23 sec ago
- Last completed build (#5), 9.8 sec ago



- Status
- </> Changes
- 📄 Console Output
- ✎️ Edit Build Information
- 🗑️ Delete build '#21'
- ⌚ Timings
- 📁 Git Build Data
- 📄 Test Result
- 🔗 Pipeline Overview
- 📄 Pipeline Console
- 🔄 Restart from Stage
- 🔄 Replay

Build #21 (Sep 18, 2024, 9:25:29 AM)

Keep this build forever

Add description

Started 1 min 13 sec ago
Took 5.9 sec

Build Artifacts

simple-maven-project-with-tests-1.0-SNAPSHOT.jar 2.04 KiB view

Started by user admin

This run spent:

- 2 ms waiting;
- 5.9 sec build duration;
- 5.9 sec total from scheduled to completion.

Revision: 9a5e9832814aa17fa7e514e5890fc6cf913d9a77

Repository: <https://github.com/jglick/simple-maven-project-with-tests.git>

- refs/remotes/origin/master

Test Result (2 failures / ±0)

test.SomeTest.test4

- Status
- </> Changes
- ▶ Build Now
- ⚙️ Configure
- 🗑️ Delete Pipeline
- 📁 Stages
- ✎️ Rename
- ? Pipeline Syntax

Build History trend

Filter...

#21
Sep 18, 2024, 9:25 AM

demo_selenium

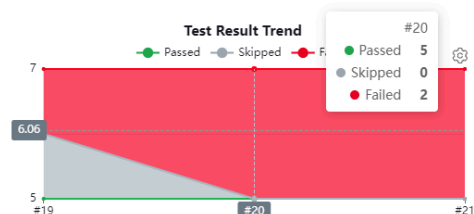
Last Successful Artifacts

simple-maven-project-with-tests-1.0-SNAPSHOT.jar 2.04 KiB view

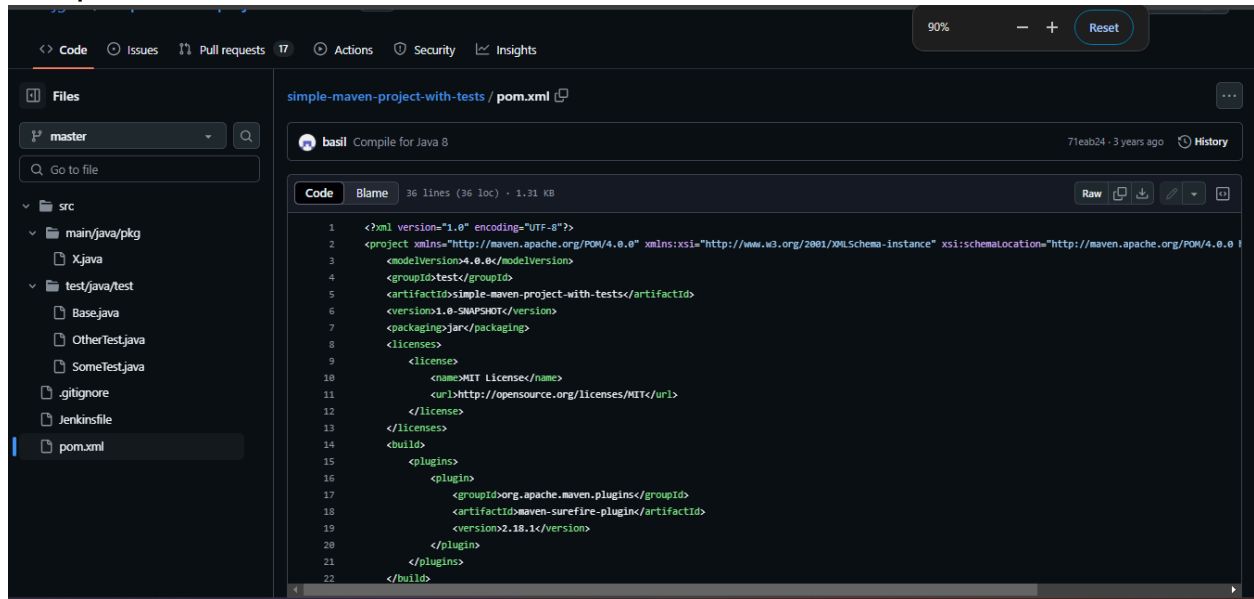
Latest Test Result (2 failures / ±0)

Permalinks

- Last build (#21), 39 min ago
- Last successful build (#21), 39 min ago
- Last failed build (#18), 1 hr 5 min ago
- Last unstable build (#21), 39 min ago
- Last unsuccessful build (#21), 39 min ago
- Last completed build (#21), 39 min ago



Step 7 : Codes



The screenshot shows the GitHub interface for the repository 'simple-maven-project-with-tests'. The 'Code' tab is selected, displaying the 'pom.xml' file. The file is 36 lines long, 36 loc, and 1.31 KB. The content of the file is as follows:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>test</groupId>
5   <artifactId>simple-maven-project-with-tests</artifactId>
6   <version>1.0-SNAPSHOT</version>
7   <packaging>jar</packaging>
8   <licenses>
9     <license>
10       <name>MIT License</name>
11       <url>http://opensource.org/licenses/MIT</url>
12     </license>
13   </licenses>
14   <build>
15     <plugins>
16       <plugin>
17         <groupId>org.apache.maven.plugins</groupId>
18         <artifactId>maven-surefire-plugin</artifactId>
19         <version>2.18.1</version>
20       </plugin>
21     </plugins>
22   </build>
```



The screenshot shows the GitHub interface for the repository 'simple-maven-project-with-tests', specifically the 'src / test / java / test / Base.java' file. The file is 41 lines long, 36 loc, and 1.45 KB. The content of the file is as follows:

```
17 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
18 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
19 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
20 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
21 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
22 * THE SOFTWARE.
23 */
24
25 package test;
26
27 import static org.junit.Assert.*;
28 import org.junit.internal.AssumptionViolatedException;
29
30 class Base {
31
32     protected void run() {
33         double r = Math.random();
34         if (r < 0.1) {
35             fail("oops");
36         } else if (r < 0.2) {
37             throw new AssumptionViolatedException("skipping");
38         }
39     }
40
41 }
```