

UNIT III CONTINUOUS INTEGRATION USING JENKINS**6**

Install & Configure Jenkins, Jenkins Architecture Overview, Creating a Jenkins Job, Configuring a Jenkins job, Introduction to Plugins, Adding Plugins to Jenkins, Commonly used plugins (Git Plugin, Parameter Plugin, HTML Publisher, Copy Artifact and Extended choice parameters). Configuring Jenkins to work with java, Git and Maven, Creating a Jenkins Build and Jenkins workspace.

CREATING AND CONFIGURING A JENKINS JOB

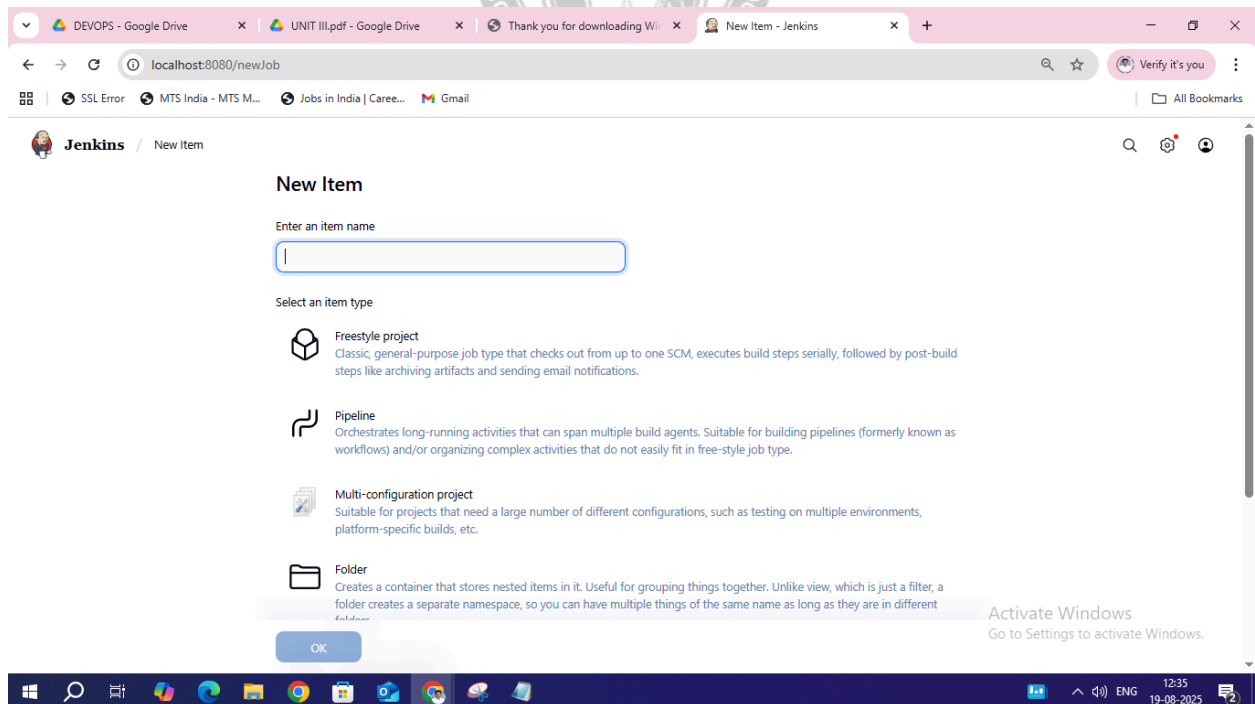
In Jenkins, Jobs (also called Projects) represent tasks or automated processes. These tasks often involve building software, running tests, deploying applications, or performing any other repetitive task that can be automated.

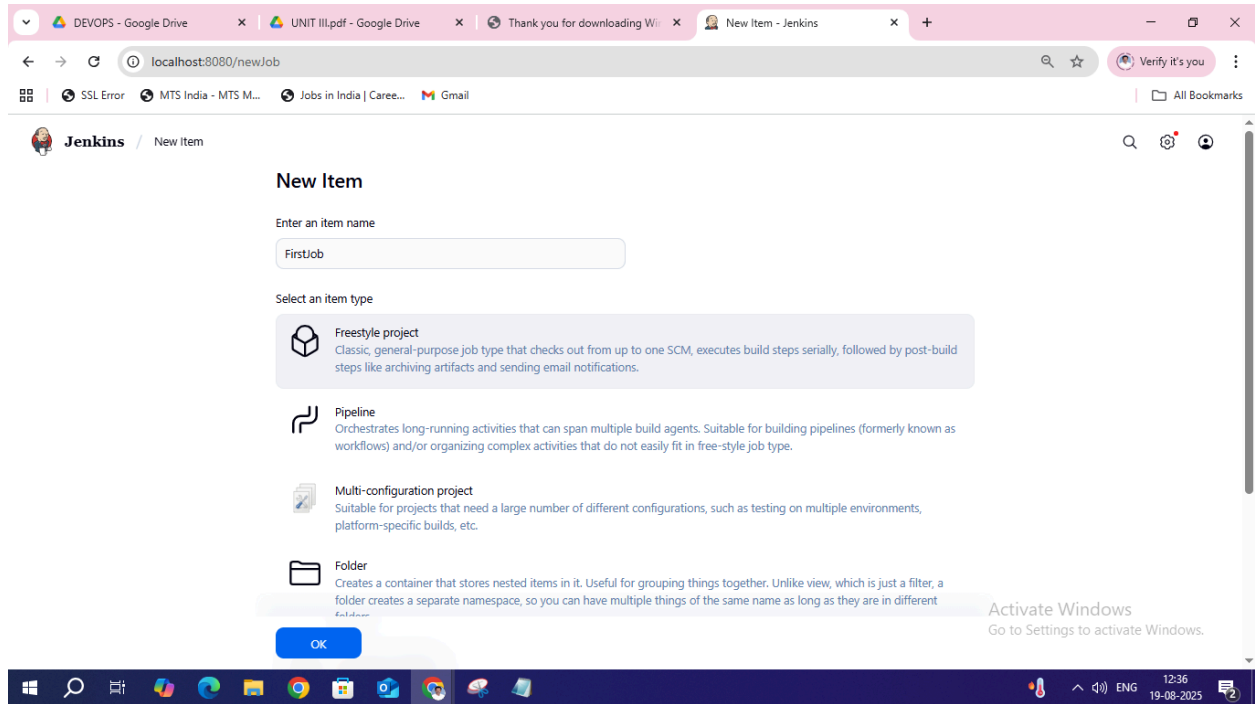
Jobs are at the core of Jenkins' functionality, enabling continuous integration and continuous delivery (CI/CD) pipelines.

Jobs include triggers, build steps, and post-build actions, enabling streamlined automation in development workflows. Jenkins job is created by using the following steps.

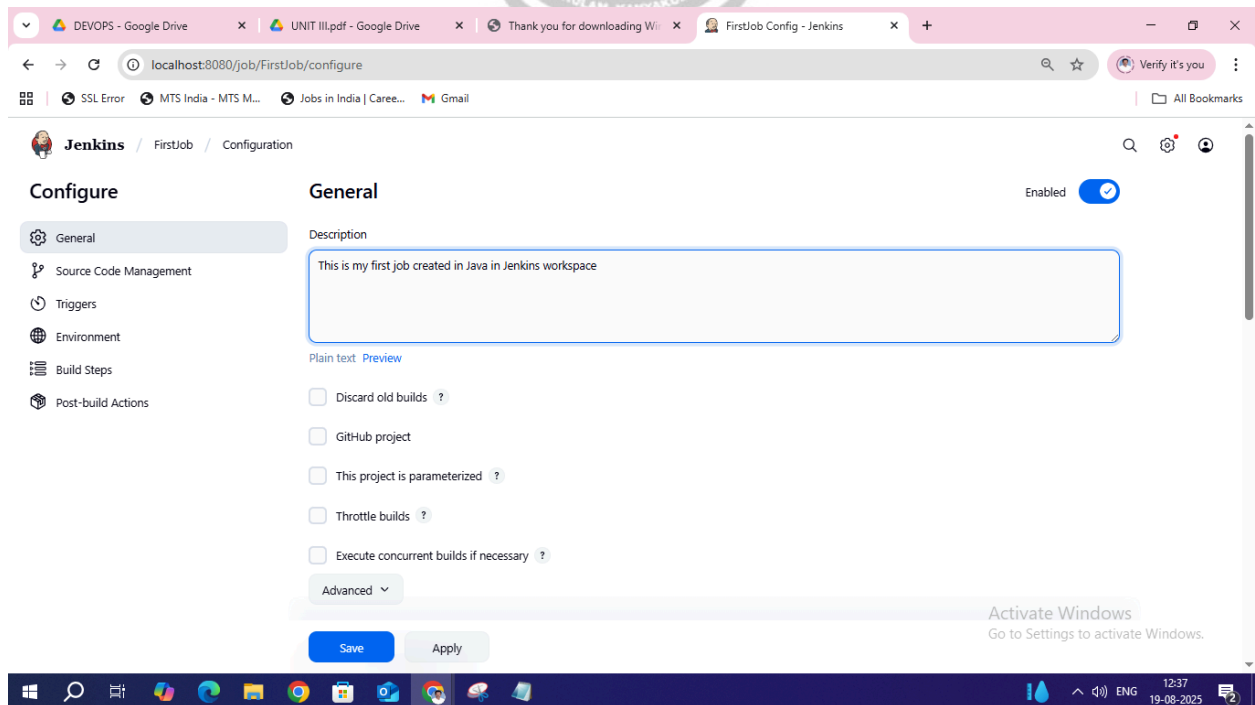
Step 1: Open Jenkins dashboard in browser by typing localhost:8080

Step 2: Create a free style project by clicking New Item and provide the name and click OK

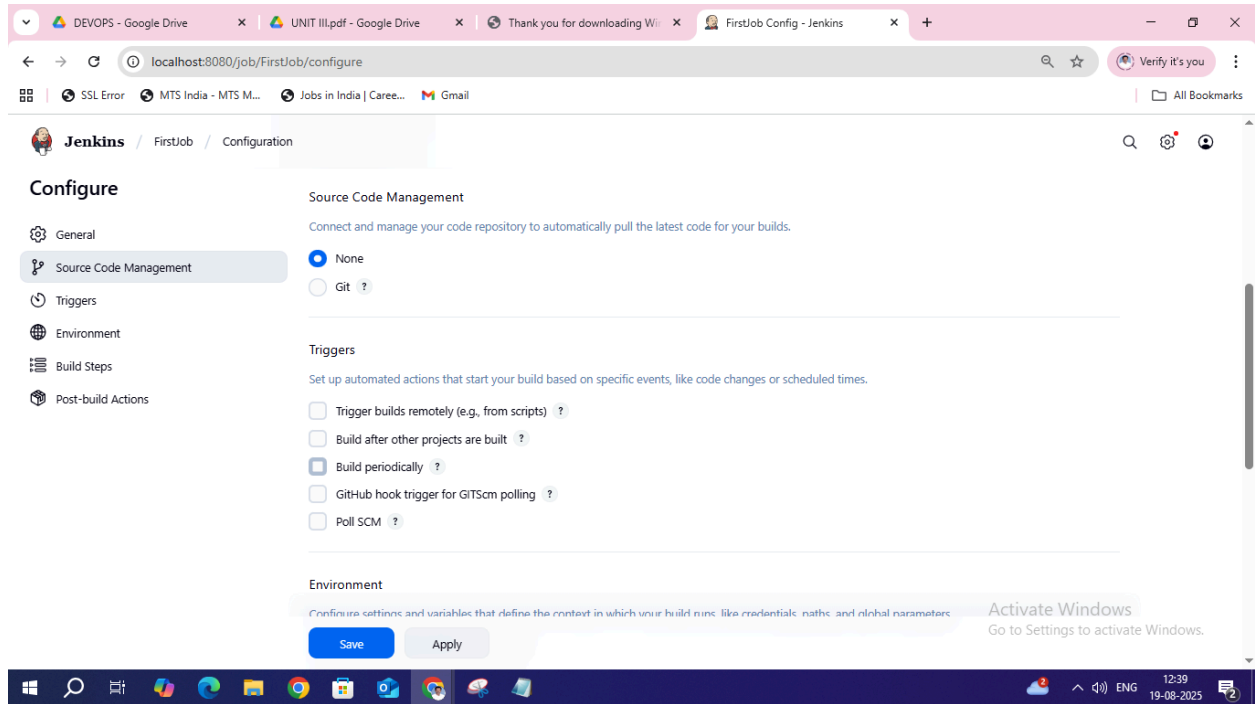




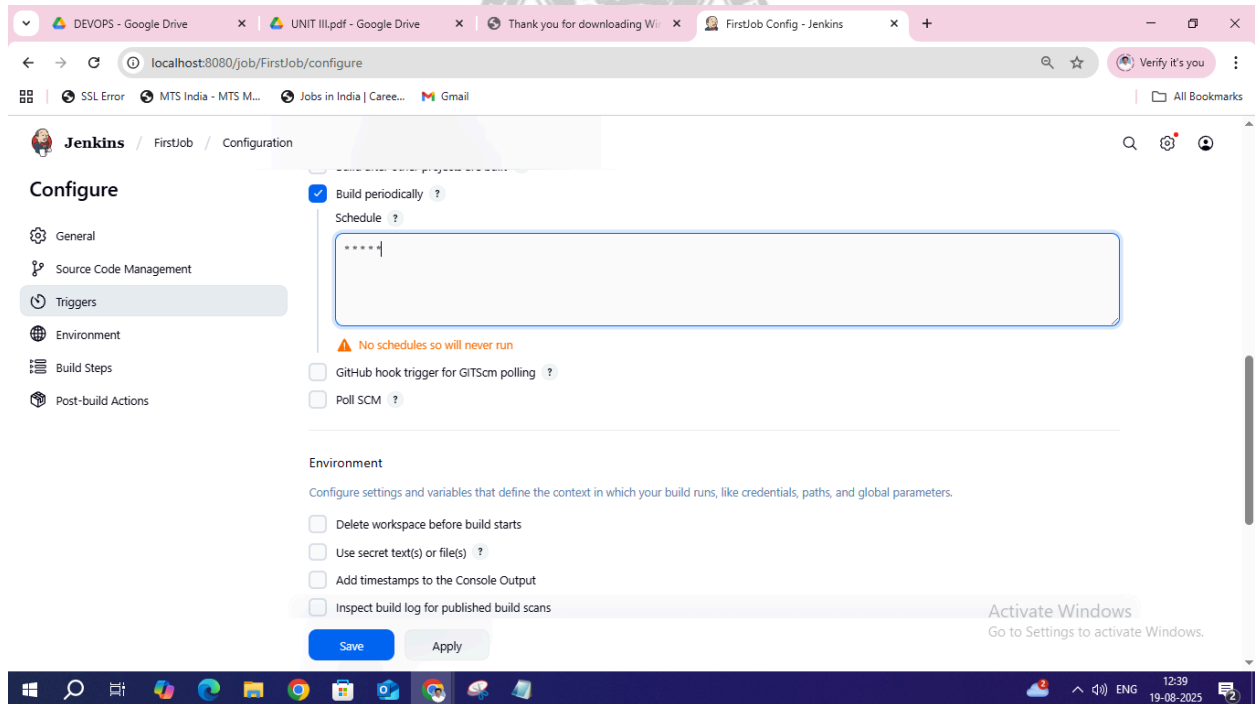
Step 3: Configure the project as follows:
Provide the description



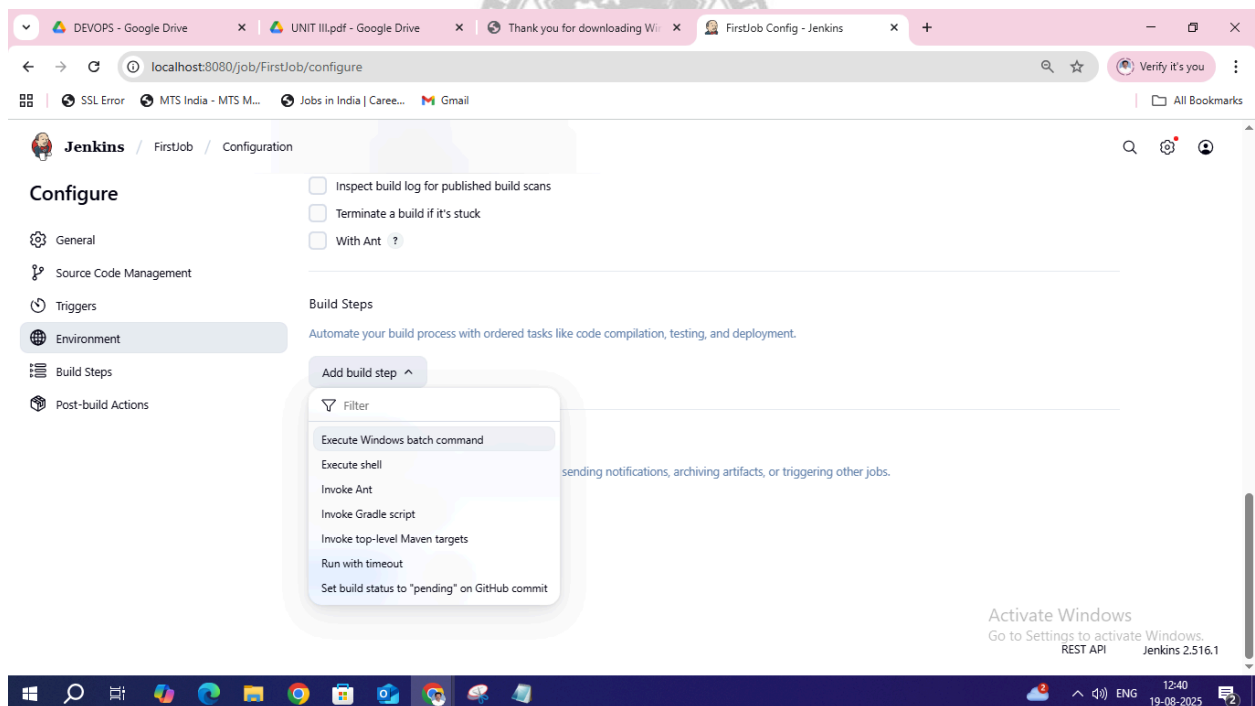
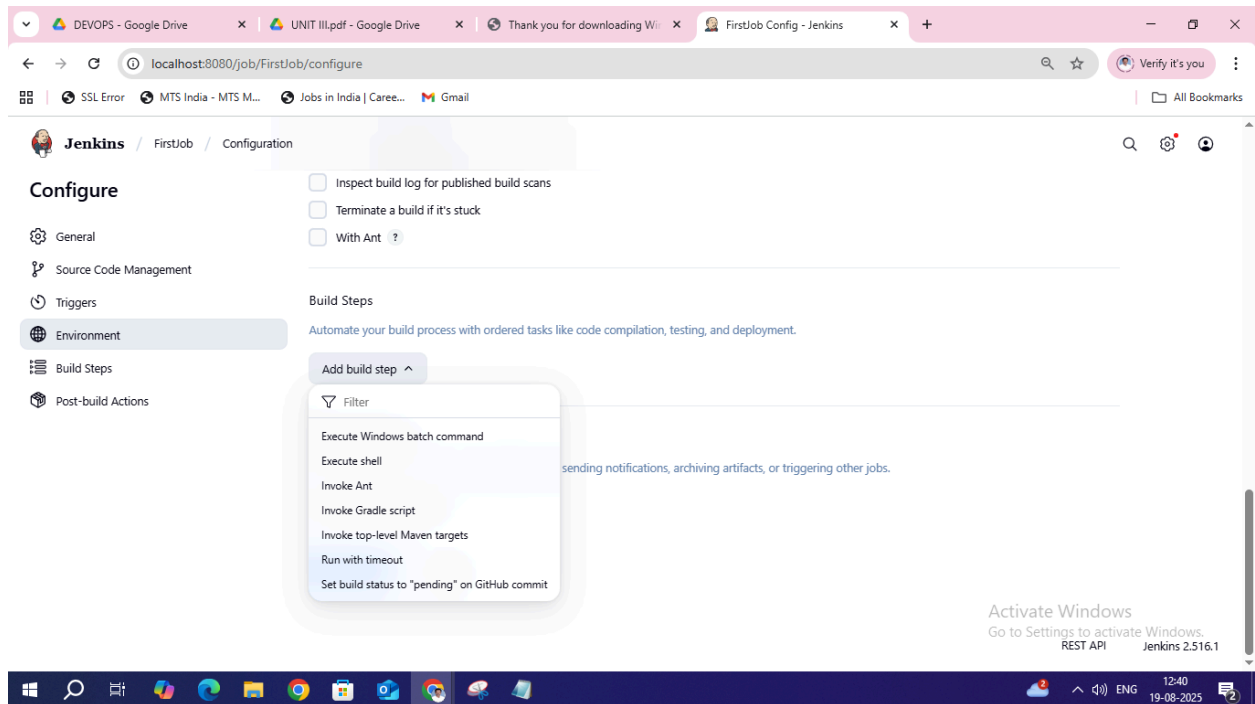
Step 4: In the source code management, specify the location of the source code.



Step 5: In the triggers sections, specify the expression to do periodic build action.



Step 6: In the build sections, specify the steps that you want Jenkins to perform when the job is run. Type the java commands to execute the java program



The screenshot shows the Jenkins web interface in a browser. The address bar indicates the URL is `localhost:8080/job/FirstJob/configure`. The page title is "Jenkins / FirstJob / Configuration". On the left sidebar, the "Configure" tab is selected, showing a list of configuration options: General, Source Code Management, Triggers, Environment, Build Steps, and Post-build Actions. The main content area is titled "Configure" and contains a text box for build steps with the following code:

```
javac FirstJob.java
java FirstJob
```

Below the build steps section, there is a "Post-build Actions" section with a description: "Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs." At the bottom of the configuration page, there are two buttons: "Save" and "Apply". A Windows watermark is visible in the bottom right corner.

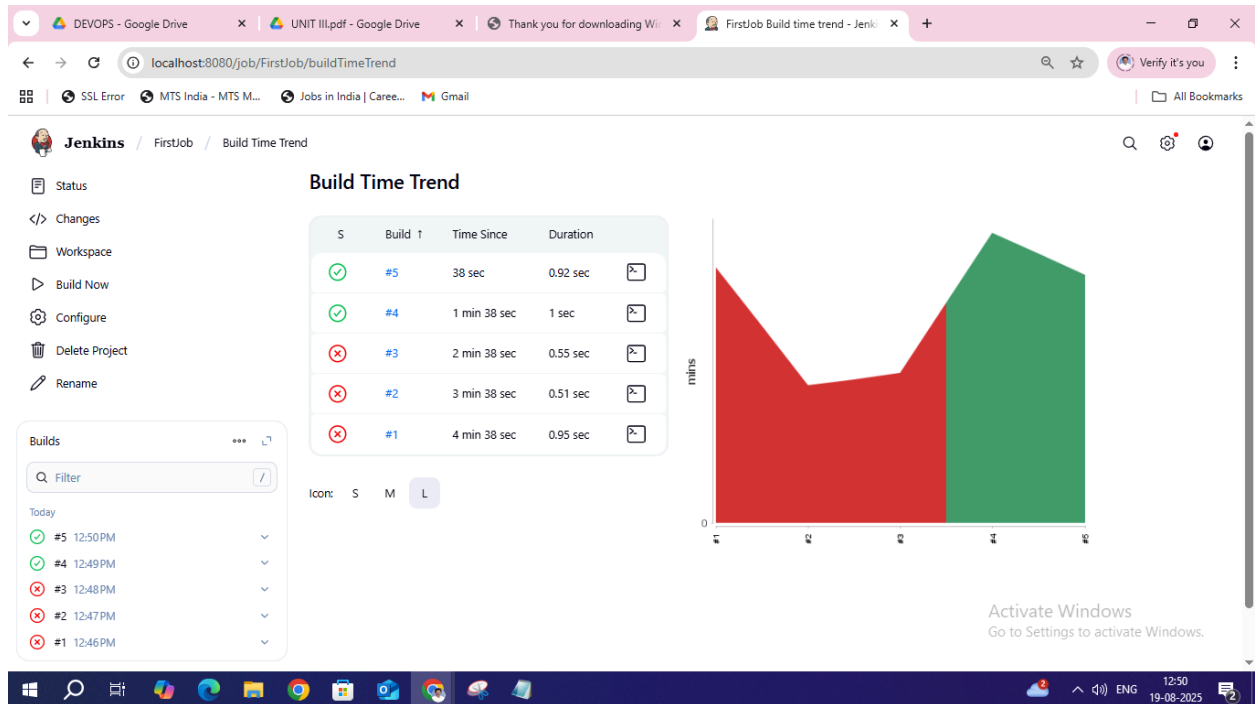
Step 7: Click Apply and Save

This screenshot is identical to the one above, showing the Jenkins configuration page for "FirstJob". However, a green confirmation message "Saved" is now visible in the bottom left corner of the page, indicating that the configuration has been successfully saved. The "Save" button is highlighted in blue, and the "Apply" button is in light blue.

The image shows two screenshots from a Windows desktop. The top screenshot is of the Jenkins web interface at `localhost:8080/job/FirstJob/`. It displays the 'FirstJob' configuration page with a status of 'Success' and a description: 'This is my first job created in Java in Jenkins workspace'. The 'Builds' section shows 'No builds'. The bottom screenshot is of a Windows File Explorer window showing the directory `This PC > Local Disk (C:) > ProgramData > Jenkins > .jenkins > workspace > FirstJob`. It contains two files: `FirstJob.class` (1 KB, CLASS File) and `FirstJob` (1 KB, Java Source File).

```
public class FirstJob
{
    public static void main(String args[])
    {
        System.out.println("First Java Program in Jenkins");
    }
}
```

Step 8: Check the status of the build by clicking on the build number

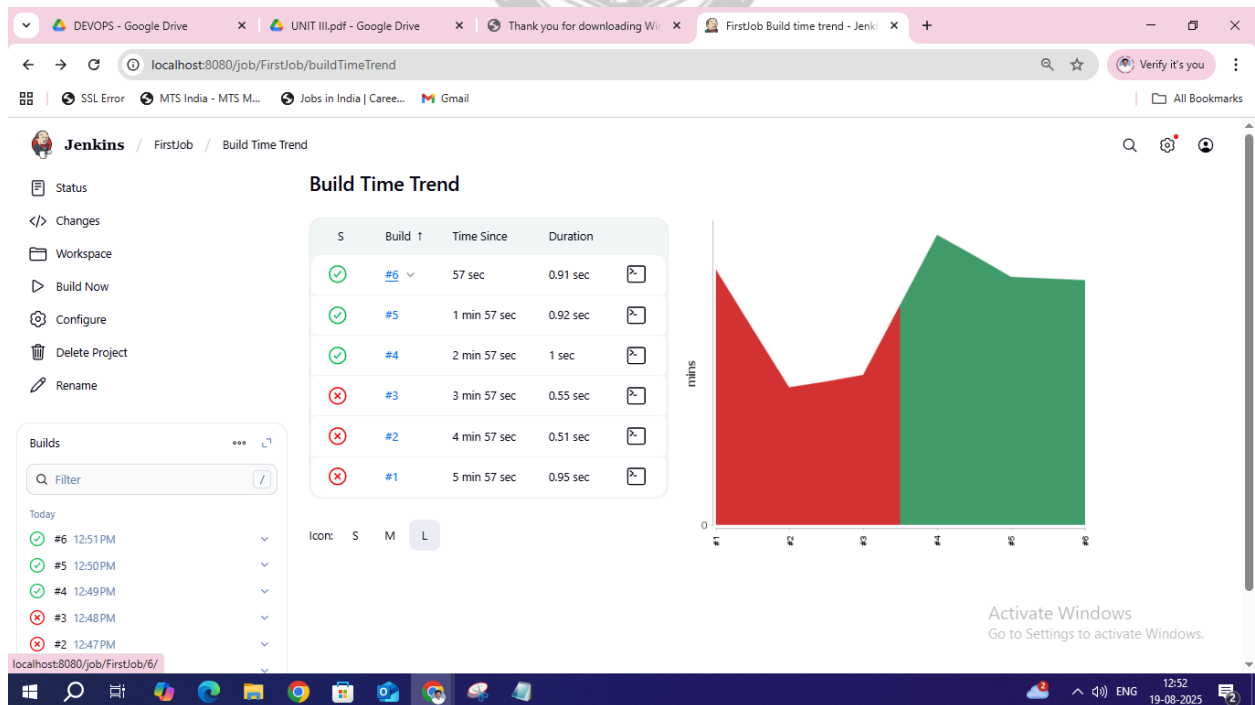


The screenshot shows the Jenkins interface for a job named 'FirstJob'. The 'Build Time Trend' page is active, displaying a table of builds and a corresponding area chart. The table lists builds #1 through #5, with their status (Success or Failure), duration, and time since the previous build. The chart visualizes the build time trend, with red bars for failed builds and green bars for successful builds. The y-axis represents time in minutes.

S	Build	Time Since	Duration
✓	#5	38 sec	0.92 sec
✓	#4	1 min 38 sec	1 sec
✗	#3	2 min 38 sec	0.55 sec
✗	#2	3 min 38 sec	0.51 sec
✗	#1	4 min 38 sec	0.95 sec

Builds list on the left:

- #5 12:50 PM
- #4 12:49 PM
- #3 12:48 PM
- #2 12:47 PM
- #1 12:46 PM

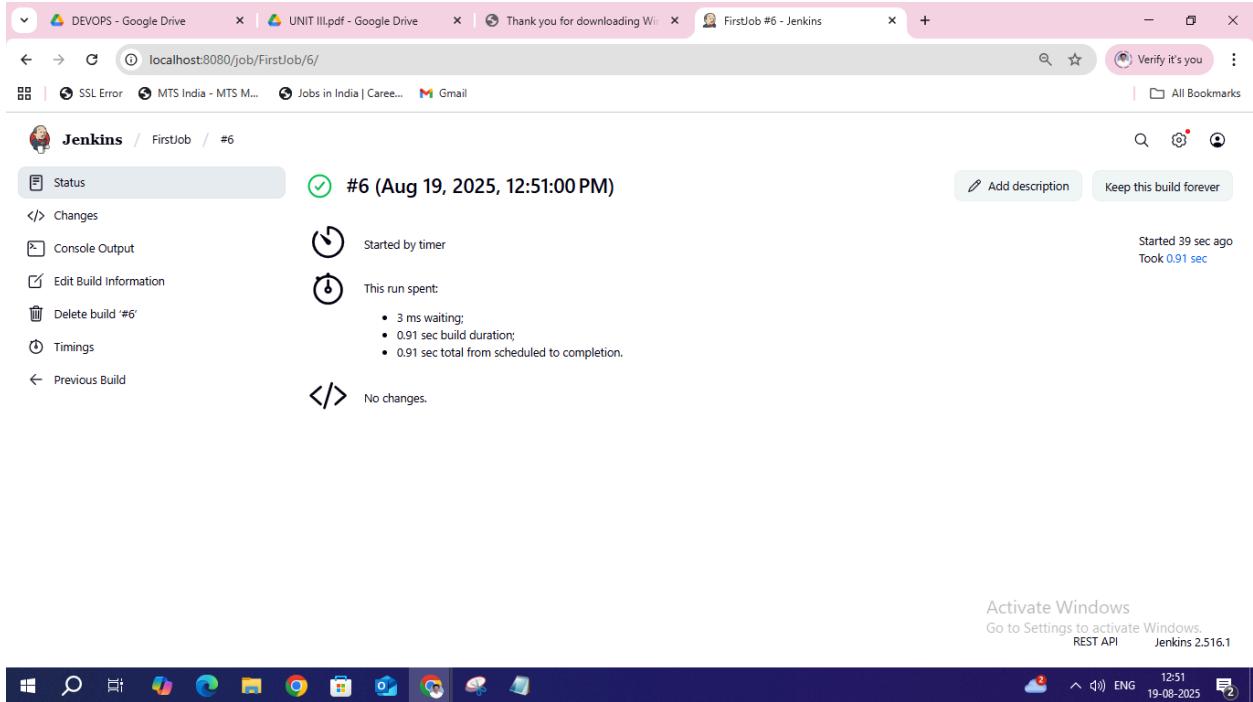


The screenshot shows the Jenkins interface for a job named 'FirstJob'. The 'Build Time Trend' page is active, displaying a table of builds and a corresponding area chart. The table lists builds #1 through #6, with their status (Success or Failure), duration, and time since the previous build. The chart visualizes the build time trend, with red bars for failed builds and green bars for successful builds. The y-axis represents time in minutes.

S	Build	Time Since	Duration
✓	#6	57 sec	0.91 sec
✓	#5	1 min 57 sec	0.92 sec
✓	#4	2 min 57 sec	1 sec
✗	#3	3 min 57 sec	0.55 sec
✗	#2	4 min 57 sec	0.51 sec
✗	#1	5 min 57 sec	0.95 sec

Builds list on the left:

- #6 12:51 PM
- #5 12:50 PM
- #4 12:49 PM
- #3 12:48 PM
- #2 12:47 PM



The screenshot shows the Jenkins web interface for a job named 'FirstJob'. The build #6, dated August 19, 2025, at 12:51:00 PM, is in a successful state. The console output shows the build was started by a timer, running as SYSTEM. It executed a command to call a batch file, then compiled a Java file (FirstJob.java) and ran it. The build finished successfully.

Jenkins / FirstJob / #6

Status: **#6 (Aug 19, 2025, 12:51:00 PM)**

Started by timer

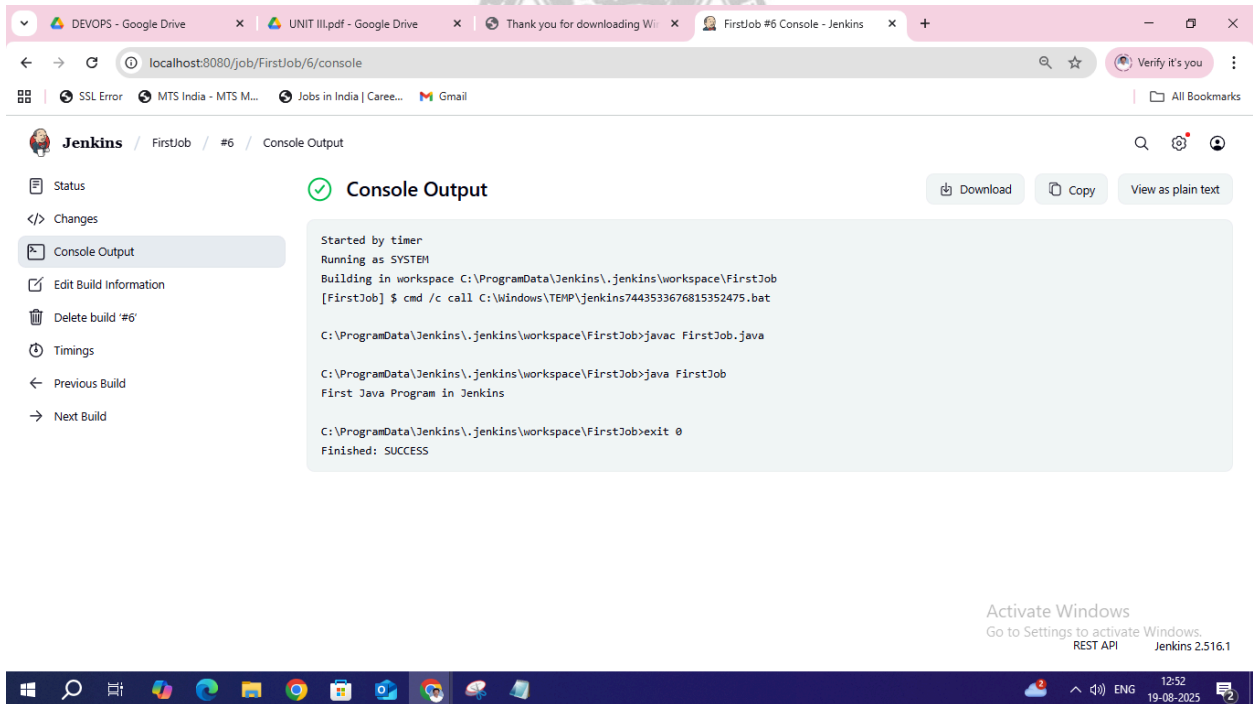
This run spent:

- 3 ms waiting;
- 0.91 sec build duration;
- 0.91 sec total from scheduled to completion.

No changes.

Activate Windows
Go to Settings to activate Windows.
REST API Jenkins 2.516.1

Step 9: Check the output in Console Output



The screenshot shows the Jenkins web interface for the same job 'FirstJob', but now viewing the 'Console Output' tab. The build #6 is still successful. The console output shows the build was started by a timer, running as SYSTEM. It executed a command to call a batch file, then compiled a Java file (FirstJob.java) and ran it. The build finished successfully.

Jenkins / FirstJob / #6 / Console Output

Status: **Console Output**

Started by timer

Running as SYSTEM

Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\FirstJob

[FirstJob] \$ cmd /c call C:\Windows\TEMP\jenkins7443533676815352475.bat

C:\ProgramData\Jenkins\jenkins\workspace\FirstJob>javac FirstJob.java

C:\ProgramData\Jenkins\jenkins\workspace\FirstJob>java FirstJob

First Java Program in Jenkins

C:\ProgramData\Jenkins\jenkins\workspace\FirstJob>exit 0

Finished: SUCCESS

Activate Windows
Go to Settings to activate Windows.
REST API Jenkins 2.516.1