Name: Kaushal Kotkar Department: Cyber Security

Div: BE-15 Roll No: 54

Subject: DSO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Experiment No. – 3** | | | | |
| **Date of Performance:** | 29/7/2024 | | | |
| **Date of Submission:** | 05/8/2024 | | | |
| Program Execution/ formation/ correction/  ethical practices (06) | Timely Submission  (01) | Viva (03) | Experiment Total (10) | Sign with Date |
|  |  |  |  |  |

**Experiment No. 3**

**Aim:** To implement Jenkins pipeline using scripted/declarative pipeline.

**Lab Outcome:** CSL701.2 Apply Jenkins to Build, Deploy and Test the Software Applications

**Theory:**

**Jenkins:** Jenkins is an open-source automation server that aids in automating various aspects of the software development lifecycle. It facilitates building, testing, and deploying software projects.

**Why Use Jenkins:** Jenkins streamlines development processes, enhances collaboration, and automates repetitive tasks. It offers extensibility through plugins and supports continuous integration and delivery.

**Features:** Jenkins provides an intuitive web interface, supports a wide range of plugins, and offers robust integration with version control systems, testing frameworks, and deployment tools.

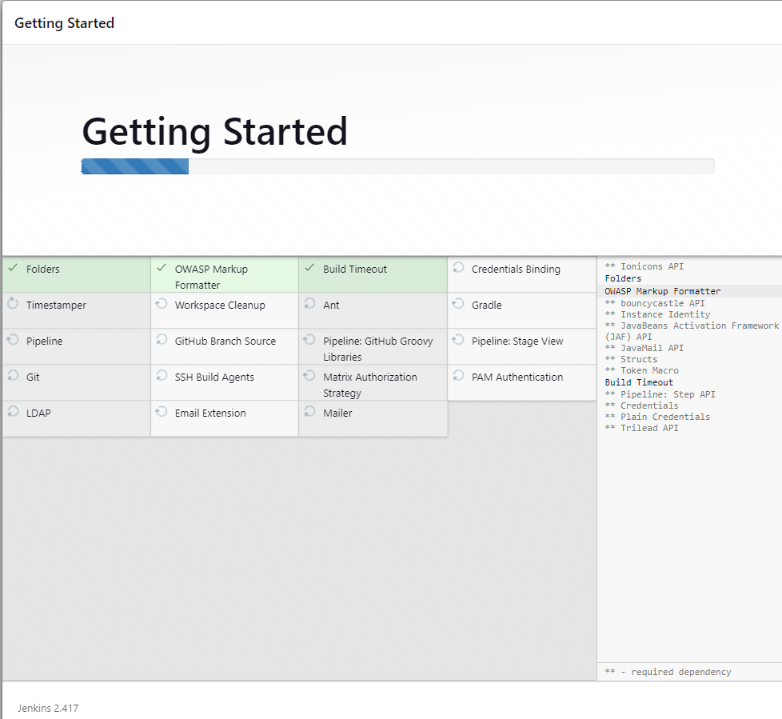
**Pipeline:** A pipeline in Jenkins is a set of steps that define how software is built, tested, and deployed. It provides a structured approach to automating the entire delivery process.

**Steps to Create a Scripted Pipeline:**

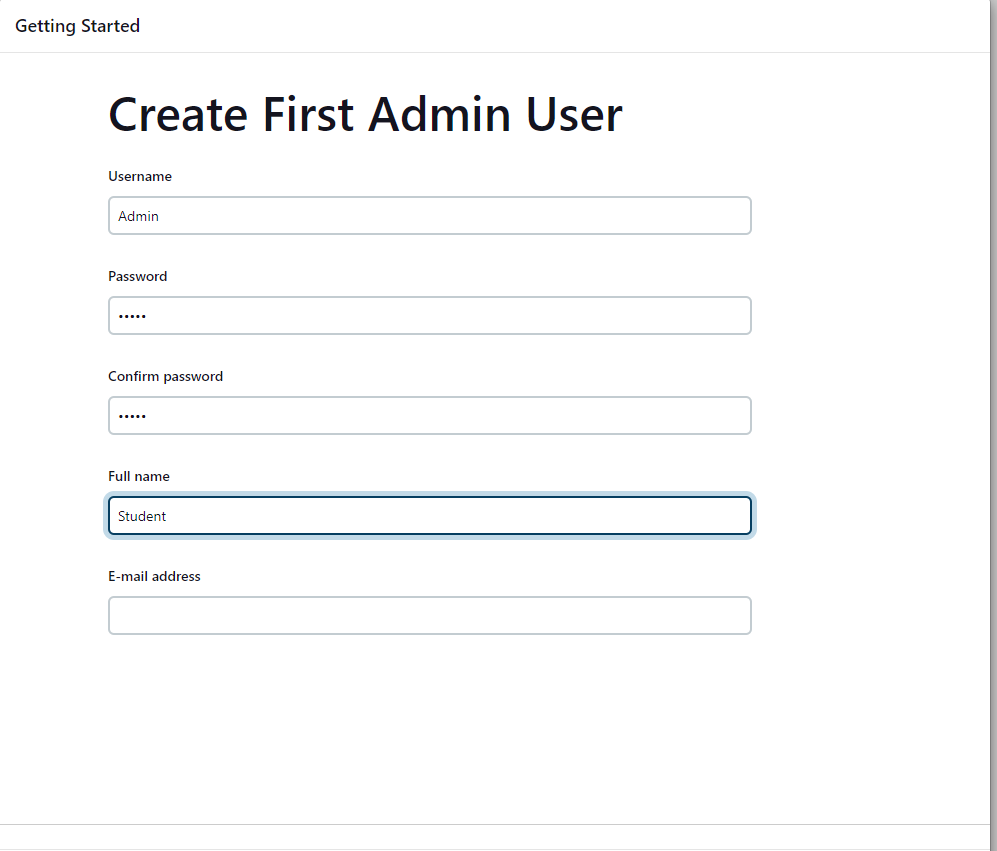
1. **Install Jenkins:** Set up Jenkins on your server.
2. **Create a New Item:** In Jenkins, create a new "Pipeline" project.
3. **Define Scripted Pipeline:** In the project configuration, select "Pipeline script" and write your scripted pipeline code.
4. **Define Stages:** Define stages for building, testing, and deployment using the stage directive.
5. **Add Steps:** Within each stage, use steps like sh for shell commands, git for version control operations, etc.
6. **Configure Post-Build Actions:** Set up post-build actions such as notifications, reports, or deployment triggers.
7. **Save and Run:** Save your pipeline configuration and run it to observe the flow and outcome.

**Output:**

Launch the Jekins

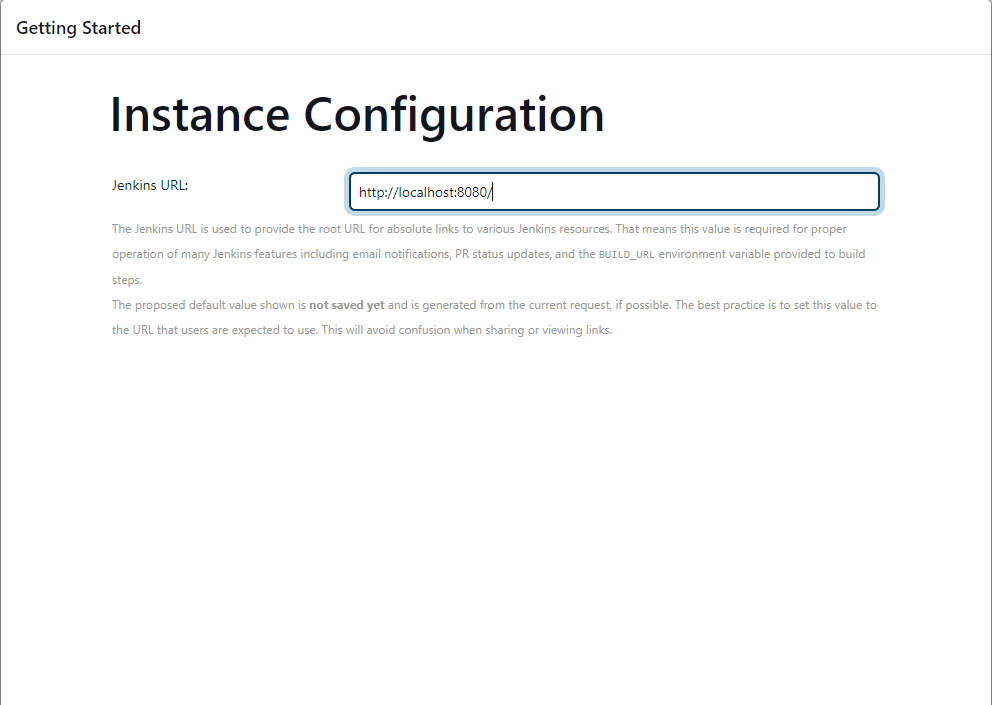


Create an Account

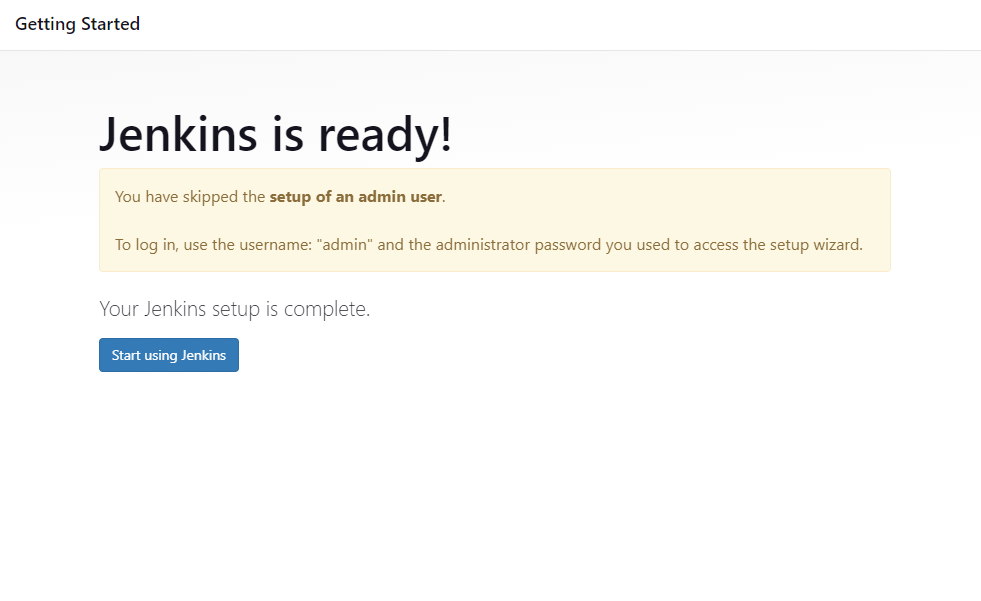


Configure it

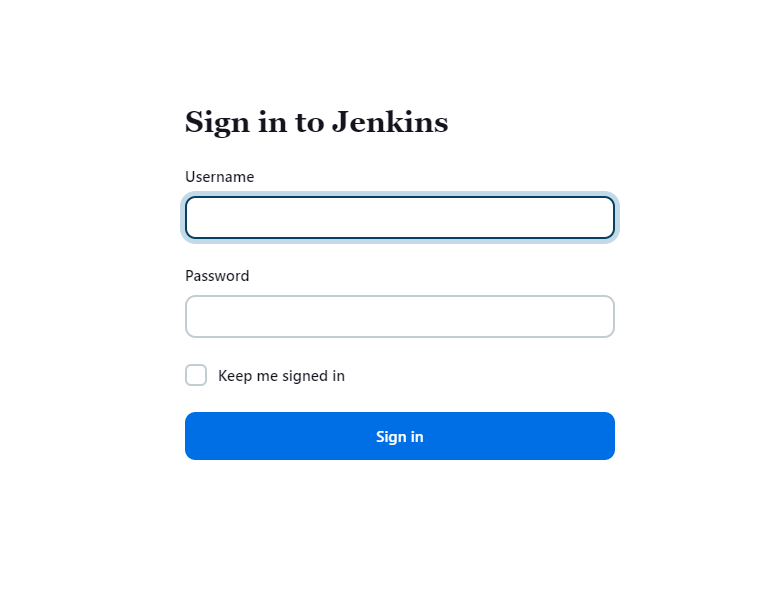
Keep default localhost:8000



Completed the installation

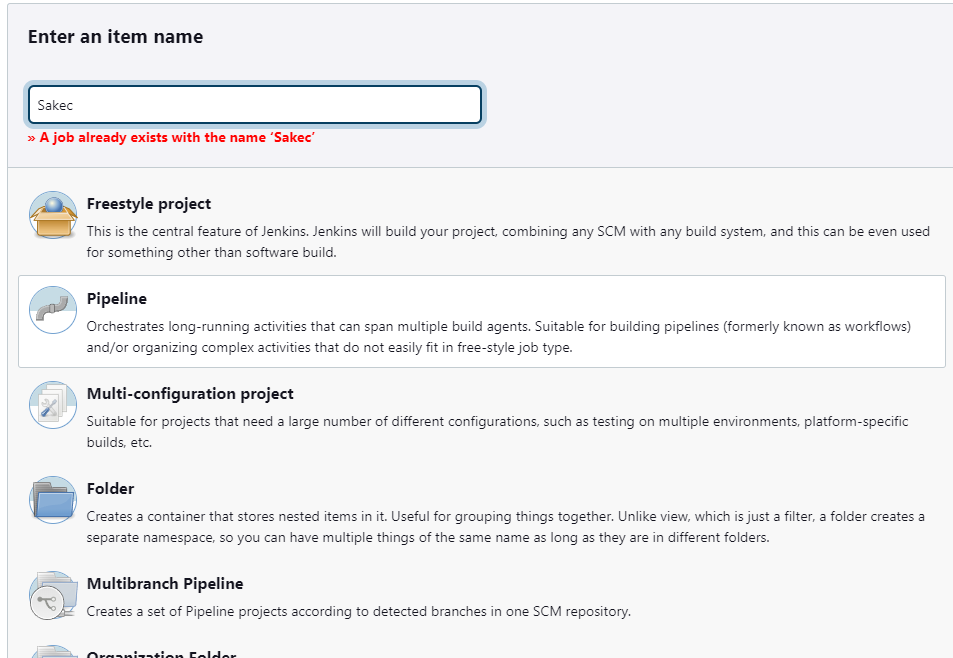


Sign into the account

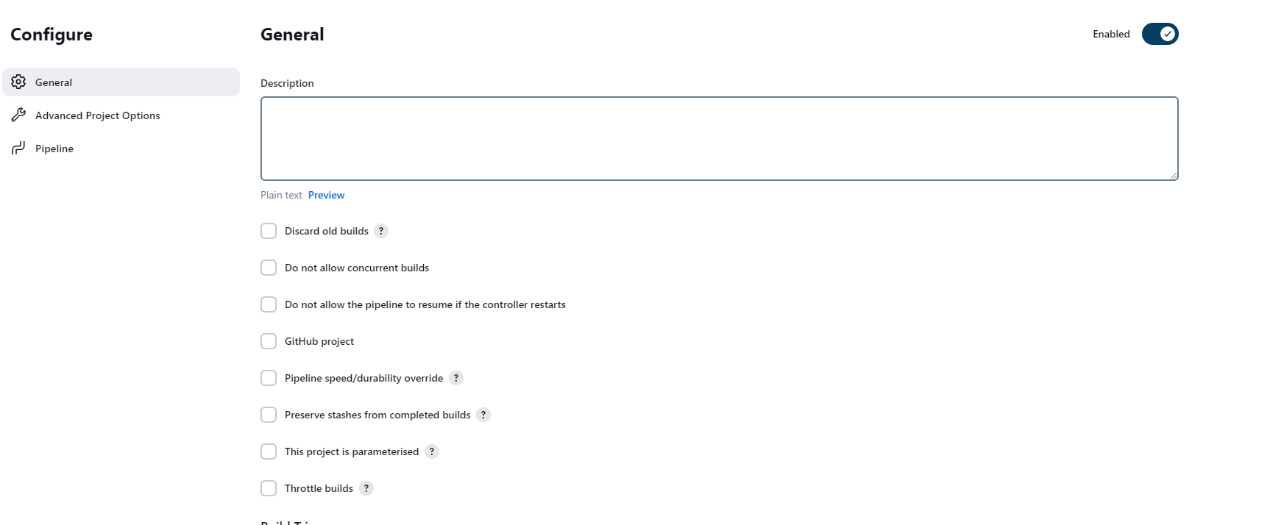


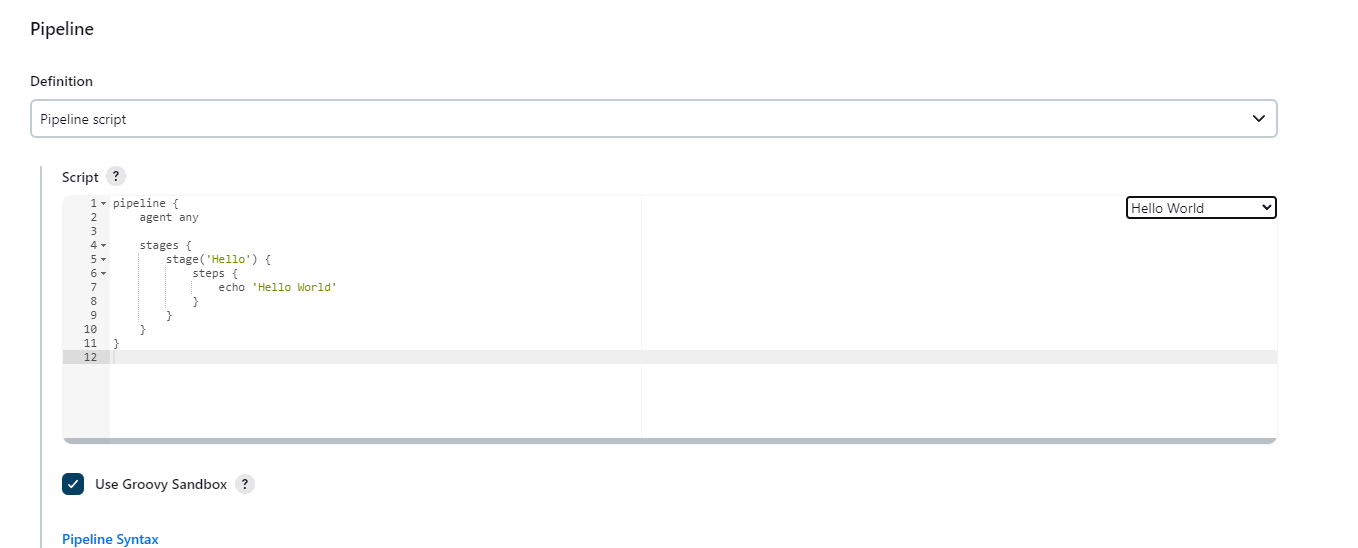
Creating a pipeline

Enter the name of pipeline and select pipeline from below option



Configure the pipeline & add the script

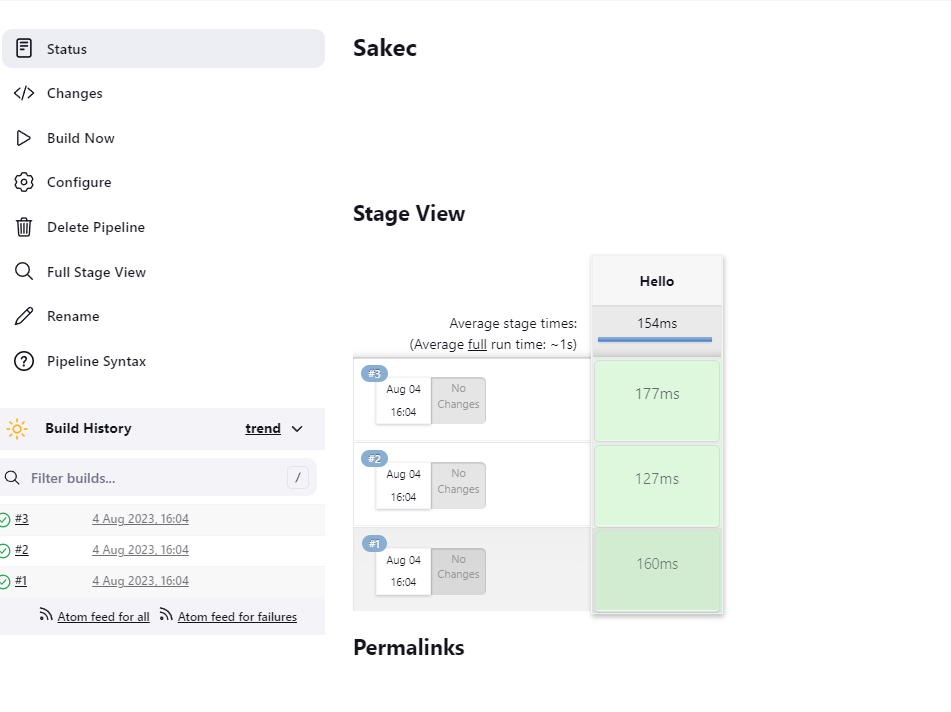




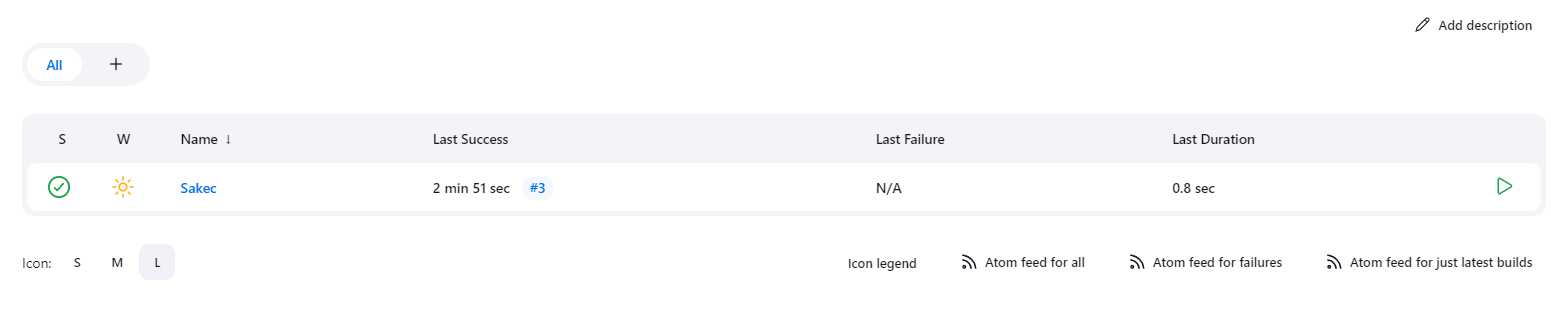
Apply the script and save it

After saving go to build now option which will build the stage

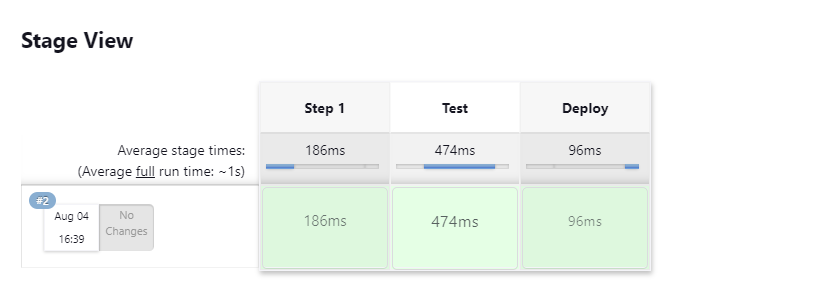
If again click the build now, it will create another stage on it



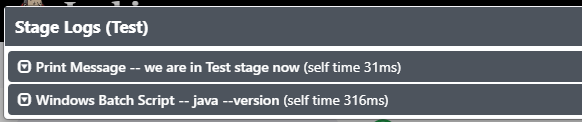
Dashboard

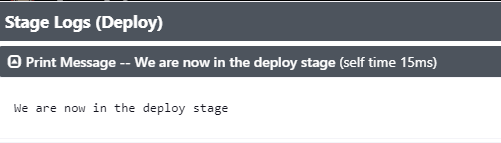


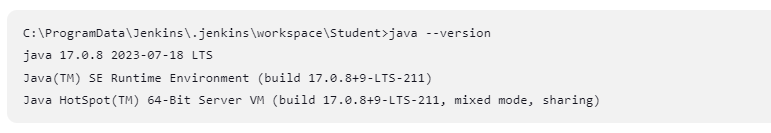
Now use another script to create pipeline



See the logs of Deploy







**Conclusion:**

Implementing a Jenkins scripted pipeline empowers software teams to automate development workflows efficiently. It enhances collaboration, accelerates delivery, and ensures consistent software quality.