**Jenkins and Pipelines.**

- Jenkins is a CI/CD & Automation tool, open source, written in Java. It can be used to automate tasks related to building, testing and delivering or deploying software. Required JRE to run.

- Before proceeding, we have to setup and install Jenkins. First, we need to install Java Development Kit (JDK) 11 or 17 from their official website:

<https://www.oracle.com/java/technologies/javase/jdk11-archive-downloads.html>  
<https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html>

- Then download a LTS version for Windows from the Jenkins official website:

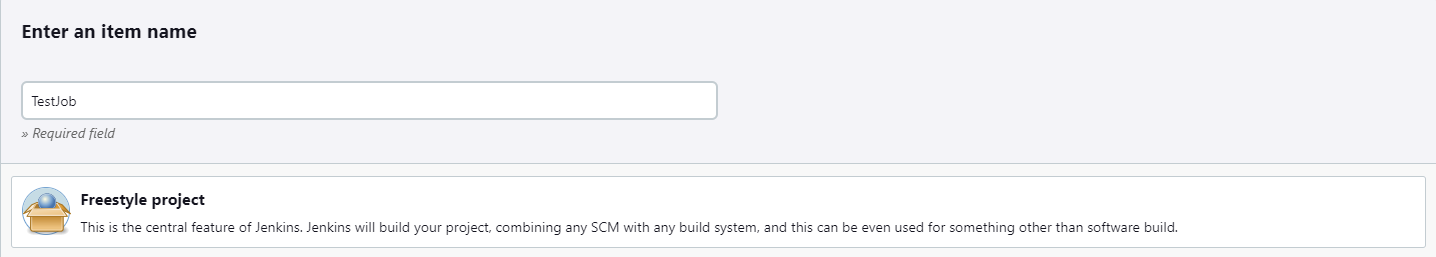
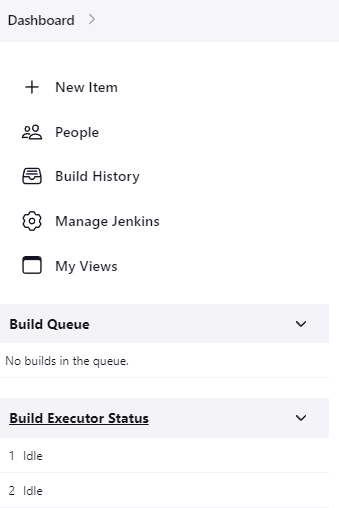
<https://www.jenkins.io/download/#downloading-jenkins>

- We can then follow the installation steps through here - <https://www.jenkins.io/doc/book/installing/windows/>

**1. Creating our first job.**

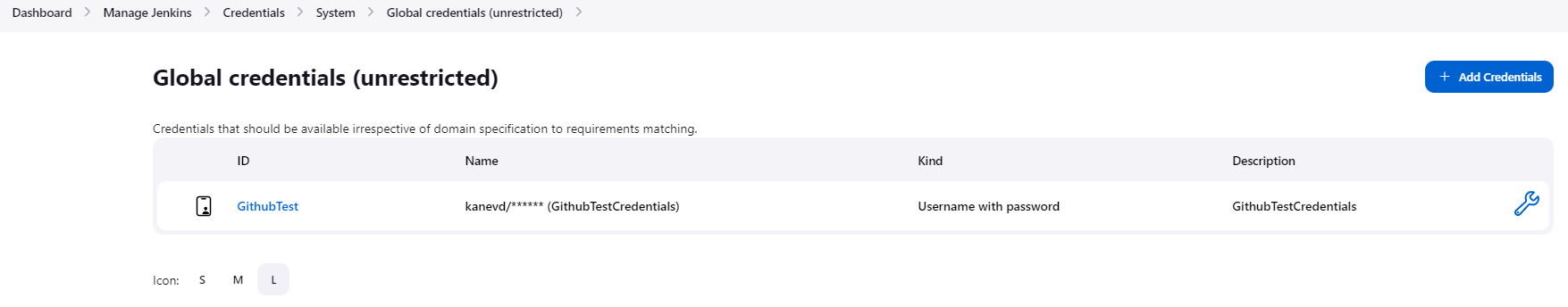
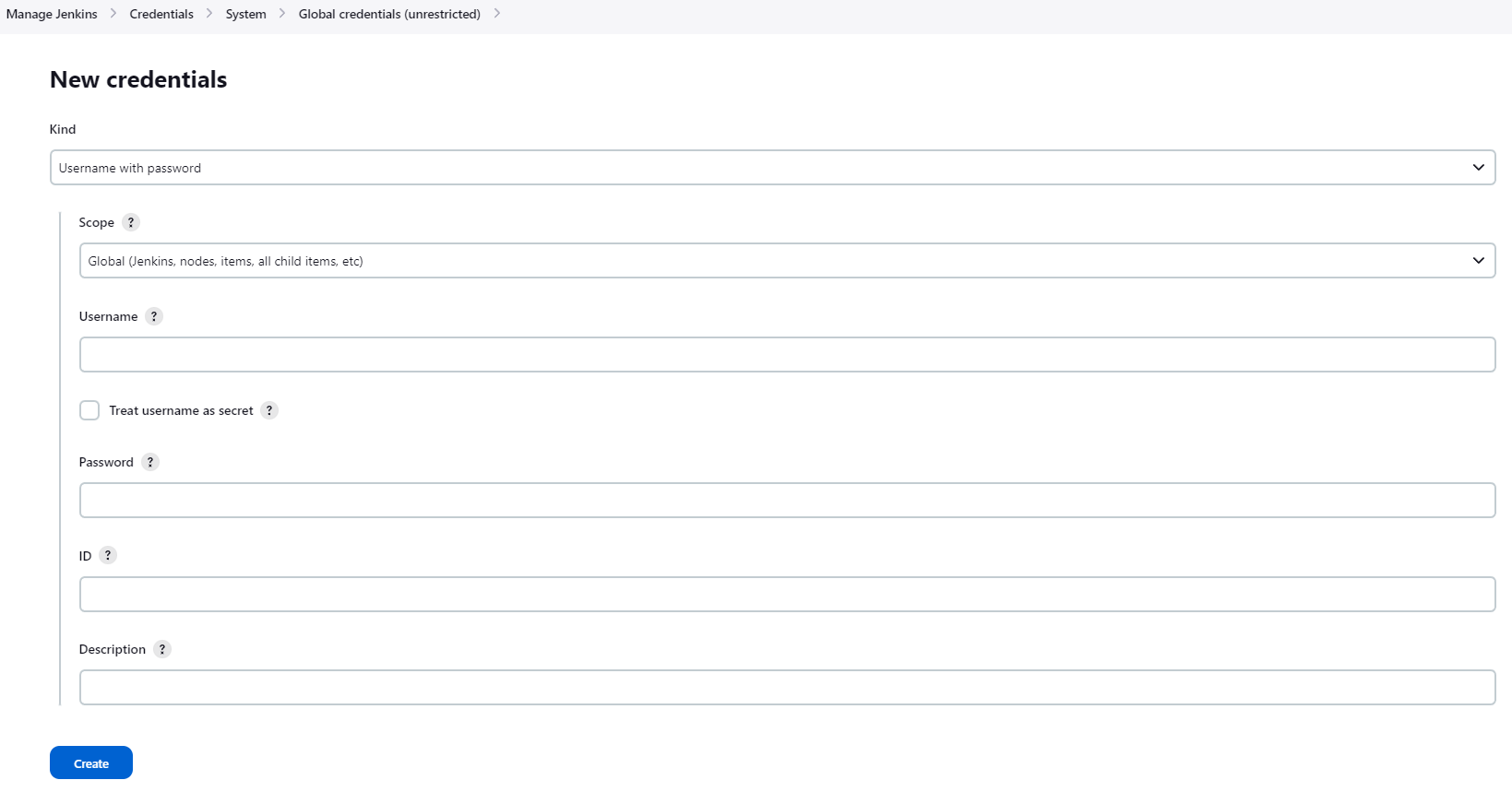
- navigate to <http://localhost:8080/> and login to the Jenkins account.

- press on “New Item” on the left side of the screen, enter your desired name and select Freestyle Project:

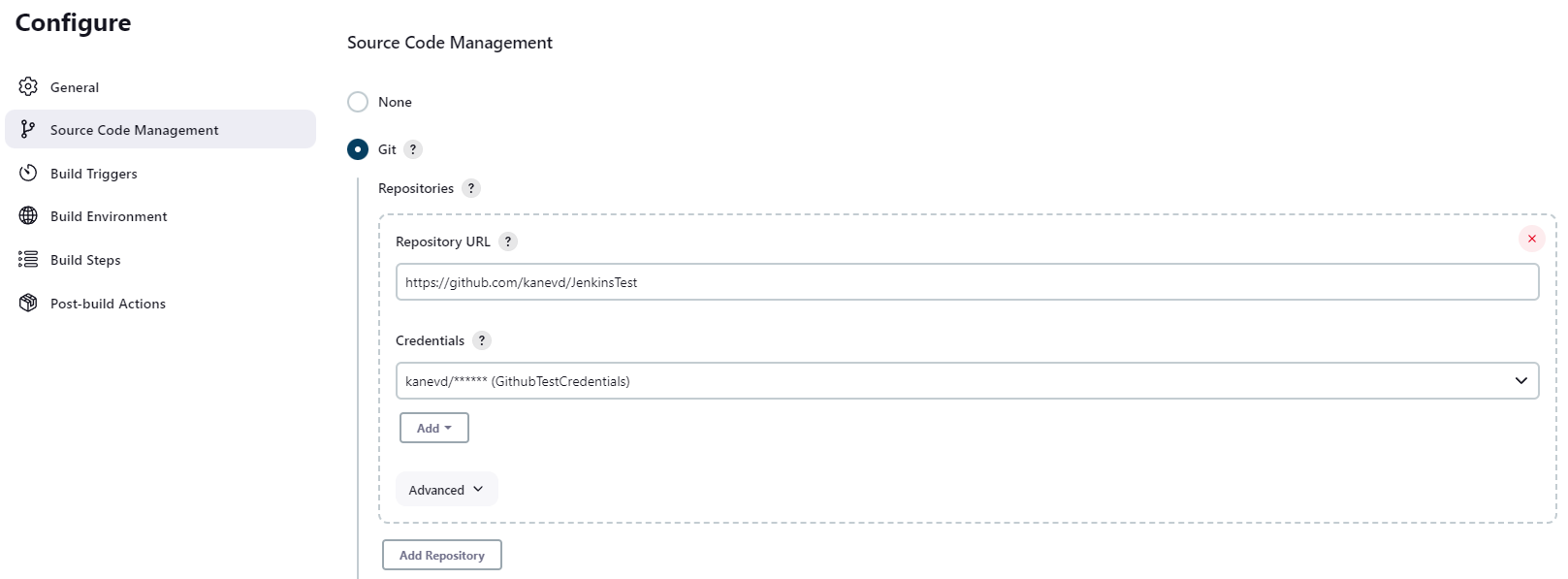


**2. Connecting to our Git Repository and adding our credentials.**

- first, we have to add our credentials that we will be using for the repository. Press on the Dashboard -> Manage Jenkins -> Manage Credentials -> System -> Global credentials -> Add Credentials. (I have already added mine):

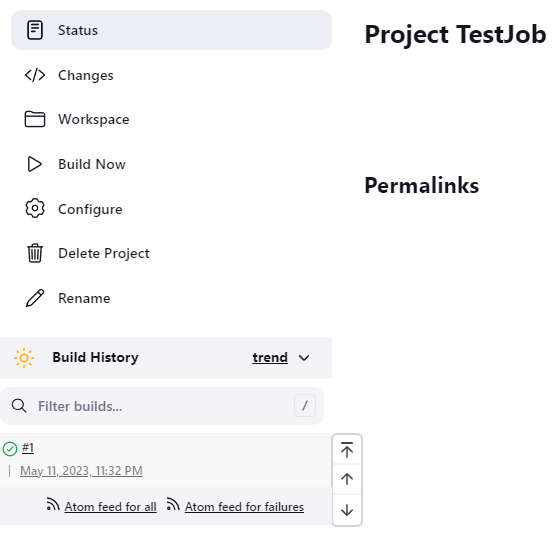


- then, we select the job that we have created (TestJob) -> Configure -> Source Code Management -> Git and add your repository & credentials, then Apply and Save at the bottom of the screen:

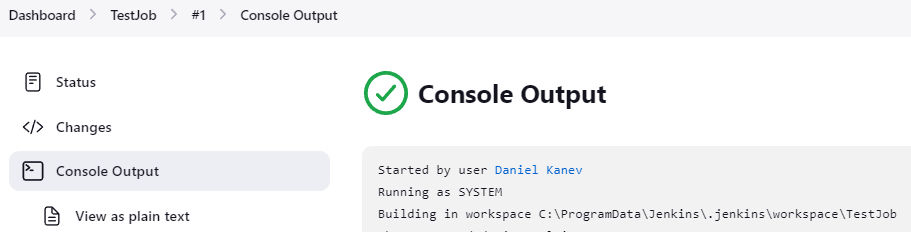


- in the steps above, the branch names should match in the Source Code Management and your Github Repository.

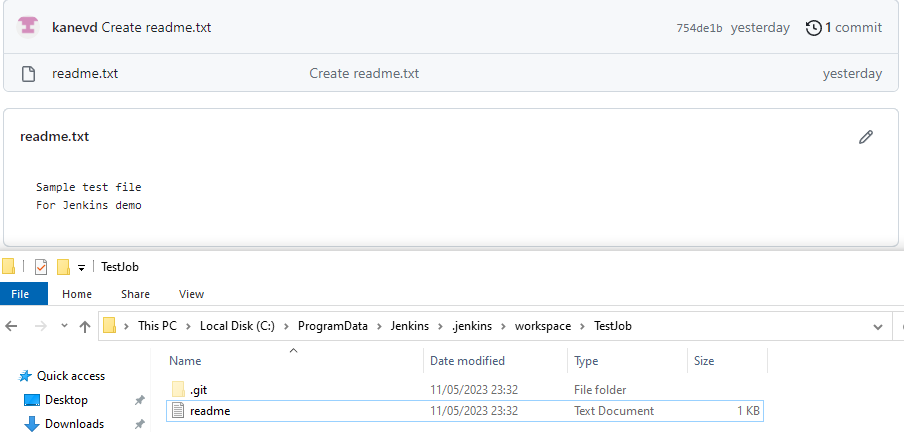
- we can now build it by pressing on Build Now (you will see 1 successful build in the Build History):



- if we press on the first build above, we can enter it’s Console Output and see where it was copied:



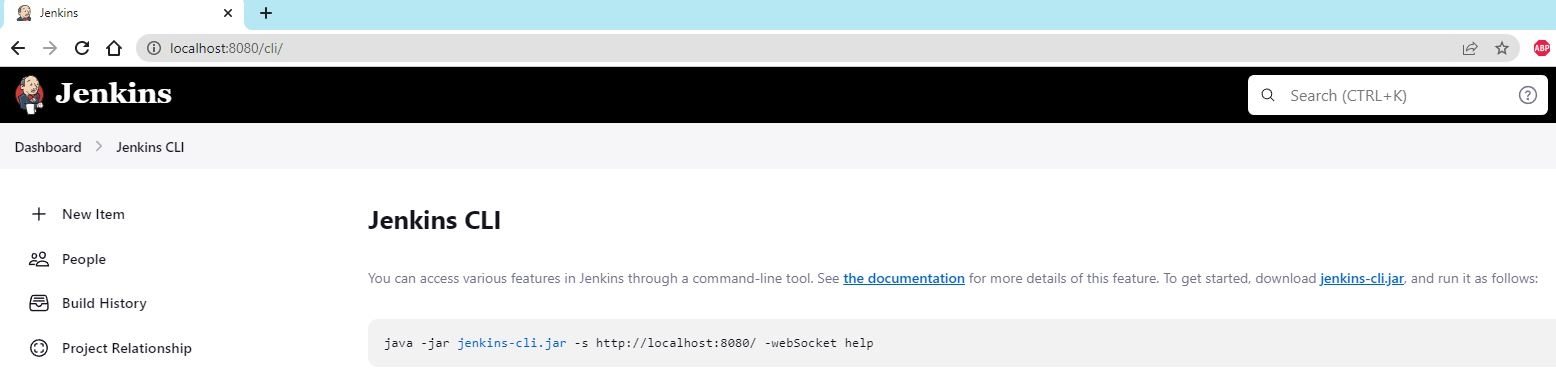
- now we can check our local computer’s folder and see that it was successfully copied:



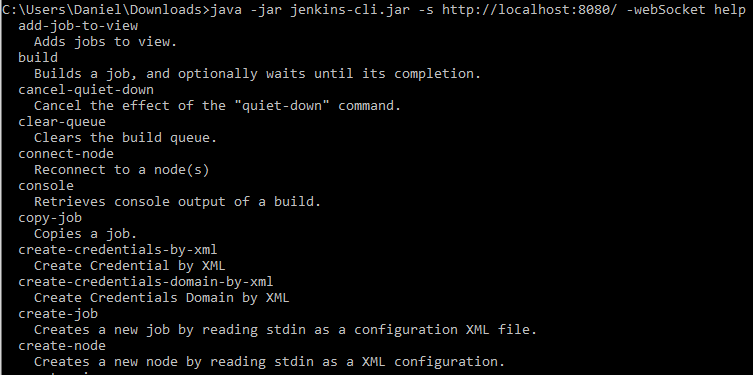
**3. Jenkins CLI.**

- sometimes, we may need to run jobs or do any setup/updates through CLI as it’s faster, easier and it can be integrated with different processes.

- to open the CLI, you simply need to add /cli to your url - http://localhost:8080/cli



- we have to download the .jar file you see above, run it (you have to be in the folder where you placed the file, or just type cmd in the search field of your folder where the file is, so it would open the exact needed folder). Run the command:



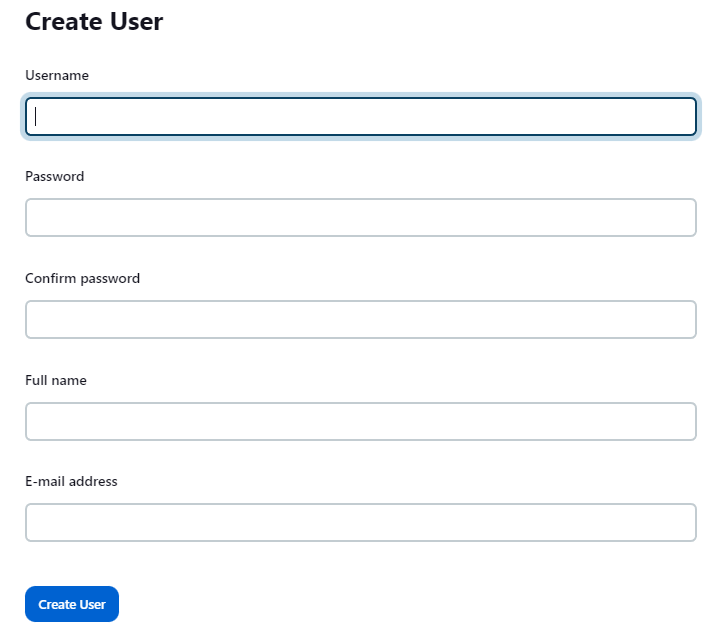
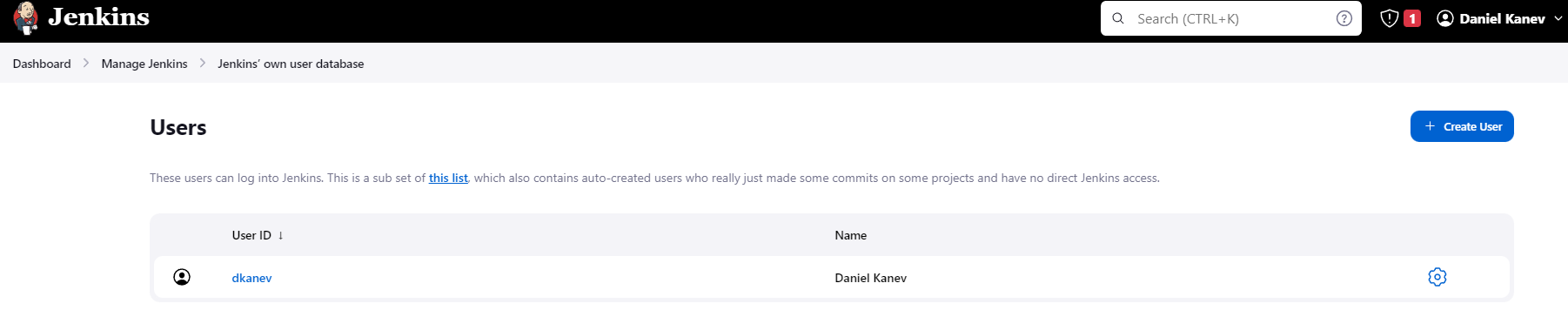
- all the available commands can also be found at the http://localhost:8080/cli link. We will test the build one.



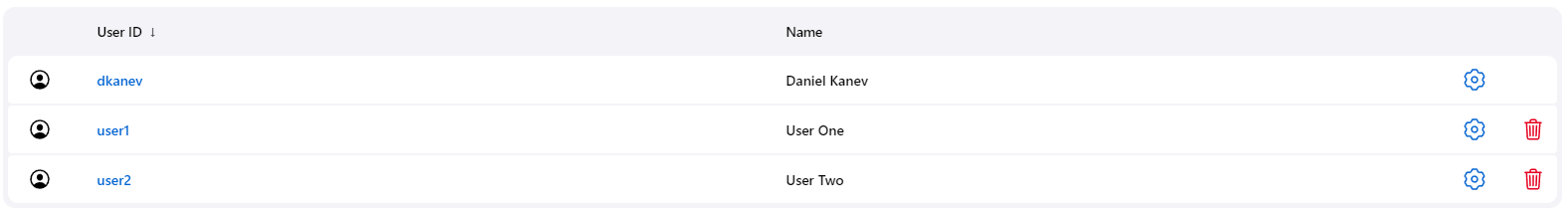
**4. Create and configure users.**

- we can create and configure new users, assign different roles and change permissions.

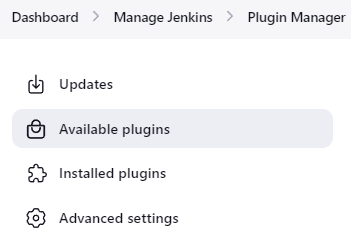
- to start, navigate to the Dashboard -> Manage Jenkins -> Manage Users and click Create User:



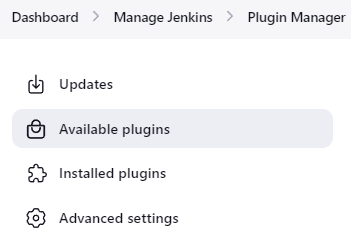
- I created 2 new users:



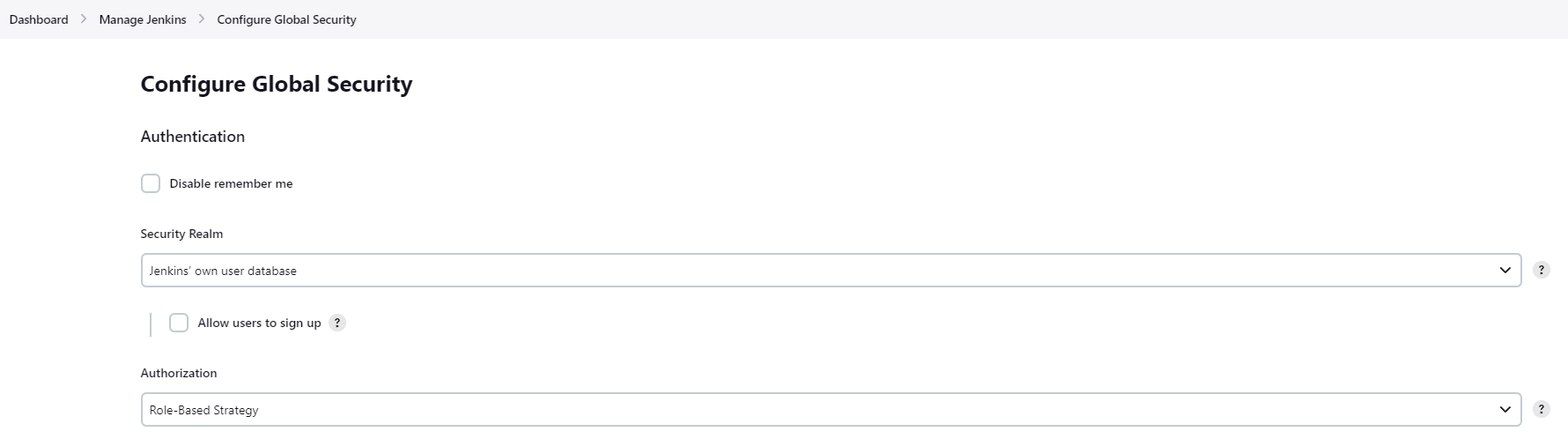
- for roles, we have to install the required plugin. From the Jenkins Dashboard -> Manage Jenkins -> Manage Plugins -> Available Plugins and search for role:



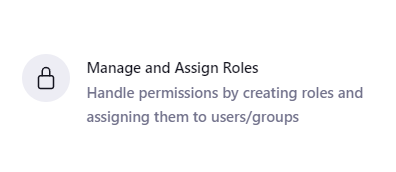
- I have already installed it, so mine is in Installed plugins:



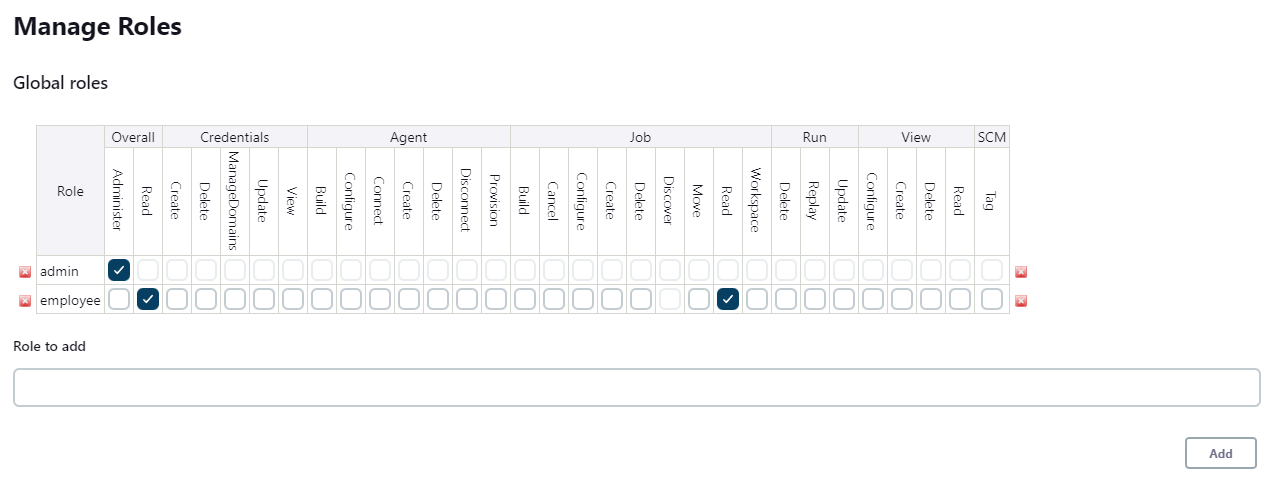
- to enable it, go to the Jenkins Dashboard -> Manage Jenkins -> Configure Global Security and select Role-based Strategy under Authorization:

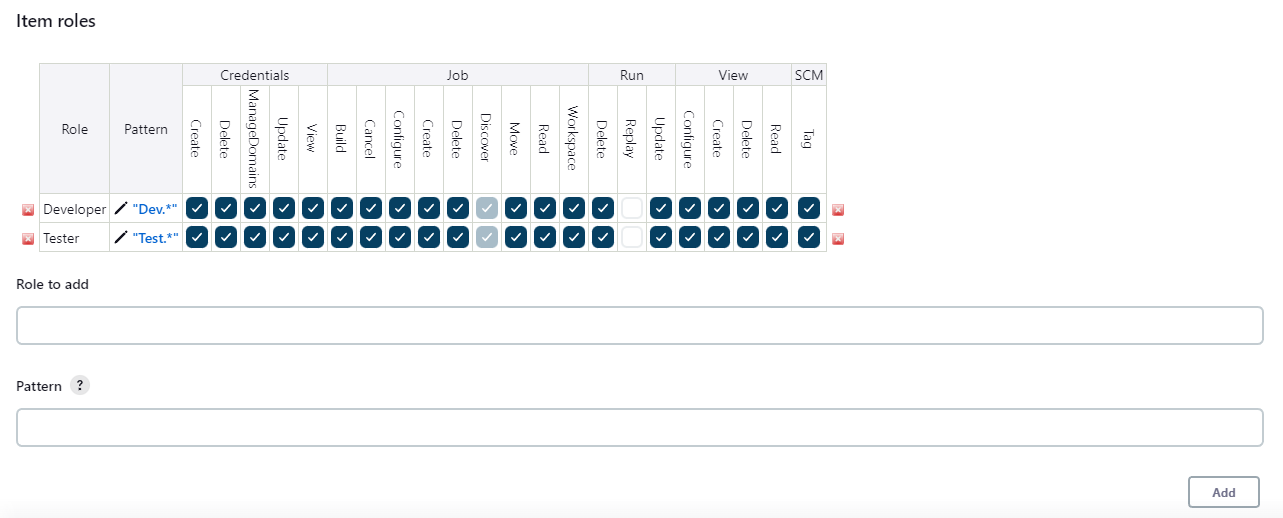


- to manage and assign roles, go back to the Dashboard -> Manage Jenkins -> and select Manage and Assign Roles:



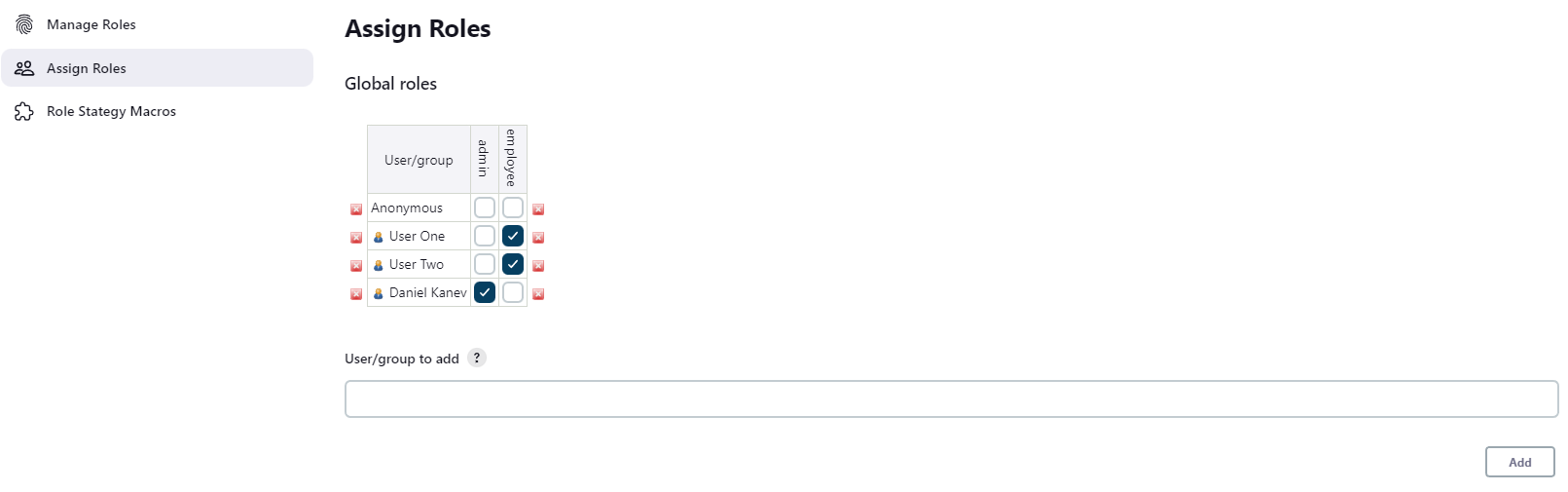
- I have created 1 role and 2 item roles, with the following permissions:



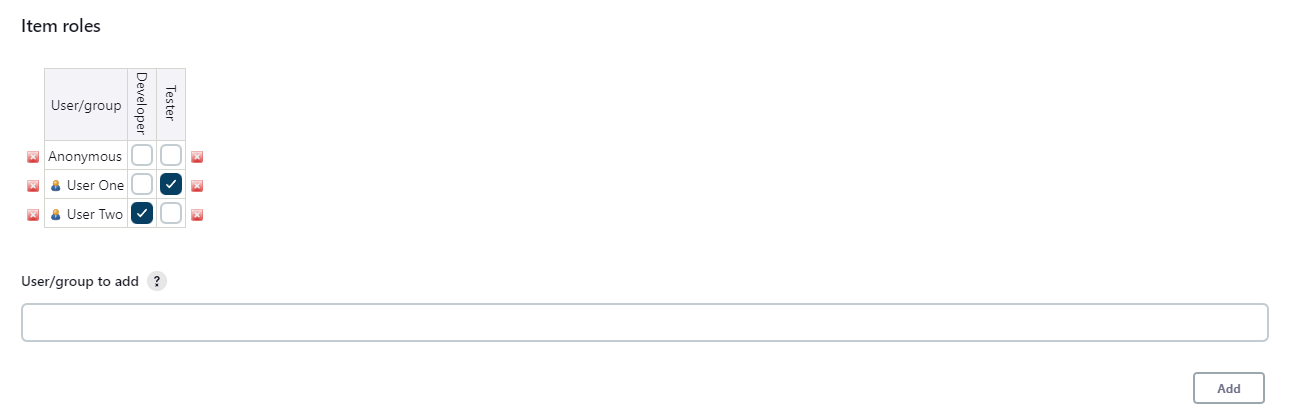


- the patterns Test.\* and Dev.\* would have full access and permissions to each Developer or Tester jobs.

- to add them as employees, go to the Assign Roles on the left:

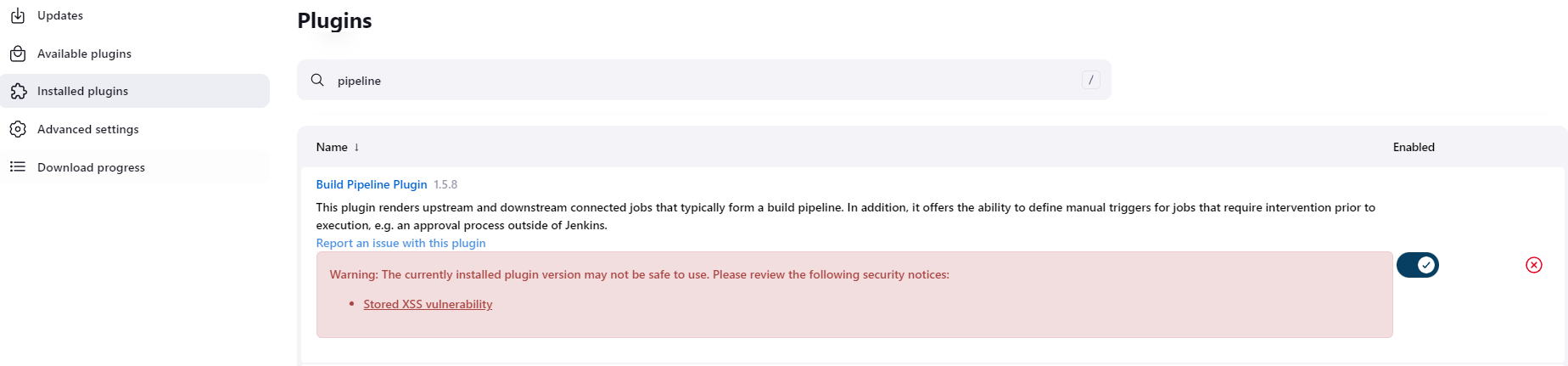


- and scroll down for the item specific roles:



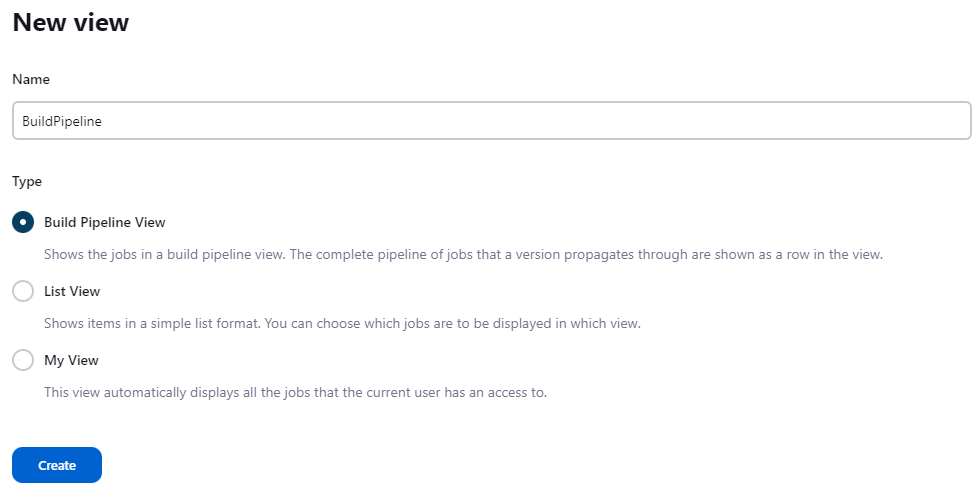
**5. Build a pipeline view.**

- we have to install the required plugin first. Dashboard -> Manage Jenkins -> Manage Plugins and search for build pipeline. I have already installed the required one:



- it’s not recommended installing plugins with similar warnings, but we will use it for the example.

- back on the Dashboard where we can see all our jobs, press on the + sign near All -> enter a name and select Build Pipeline View:



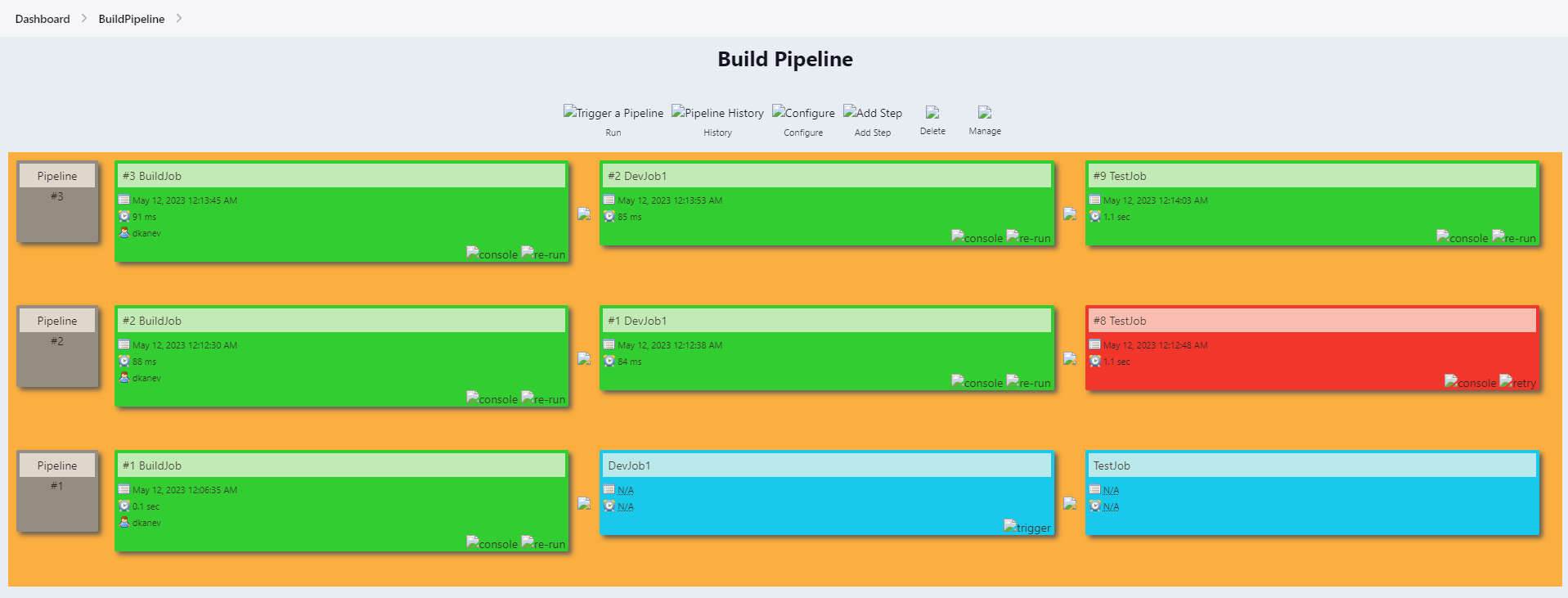
- and select our initial job:



- for the testing, we will also display 5 builds:



- we will be presented with the pipeline view:

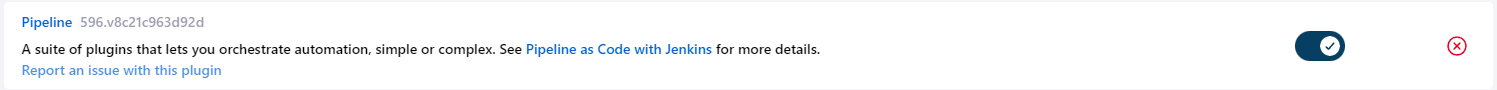


**6. Pipeline and Jenkinsfile.**

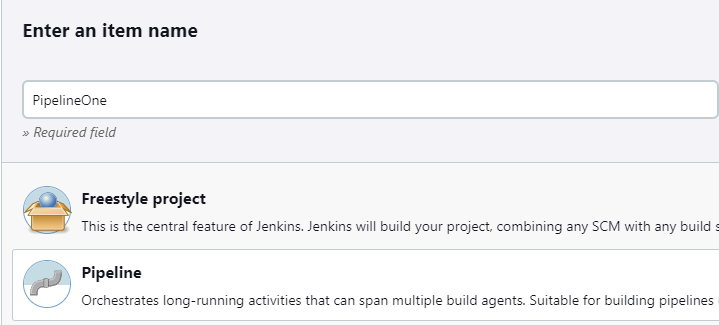
- pipeline – process to implement CI/CD, all the jobs/stage of the projects can be integrated.

- jenkinsfile – pipeline as a code, as the code/scripts are written in it.

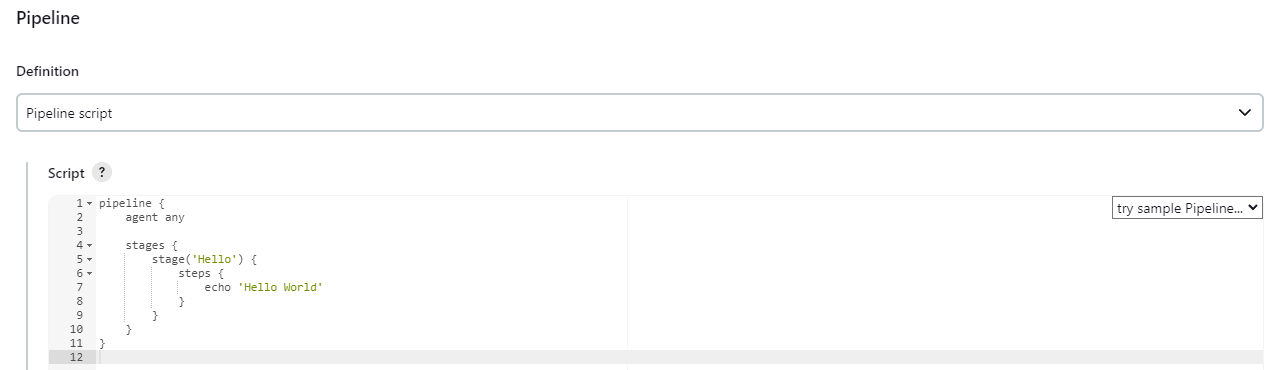
- first, we need to check if the required plugin is installed (it’s usually installed by default when starting Jenkins for the first time):



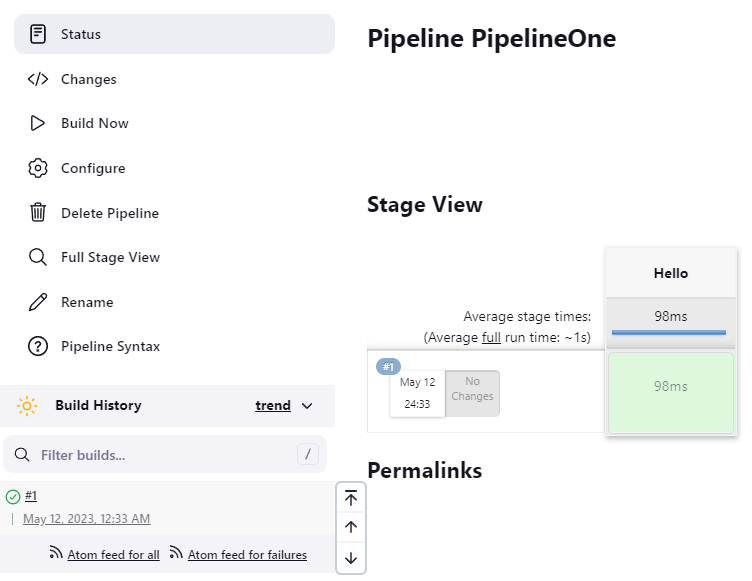
- to create a pipeline, go to the Jenkins Dashboard -> +New Item -> enter a name and select Pipeline:



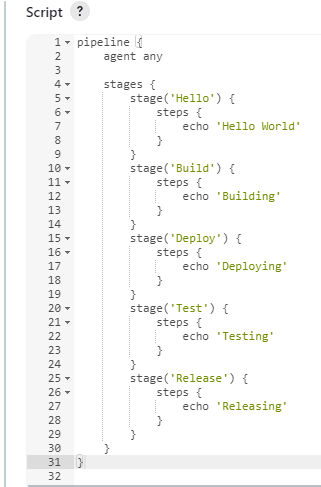
- the first test will be with a simple Pipeline script. Go to the Pipeline -> Configure -> scroll down to Pipeline and from the drop-down menu, select the Hello World script:



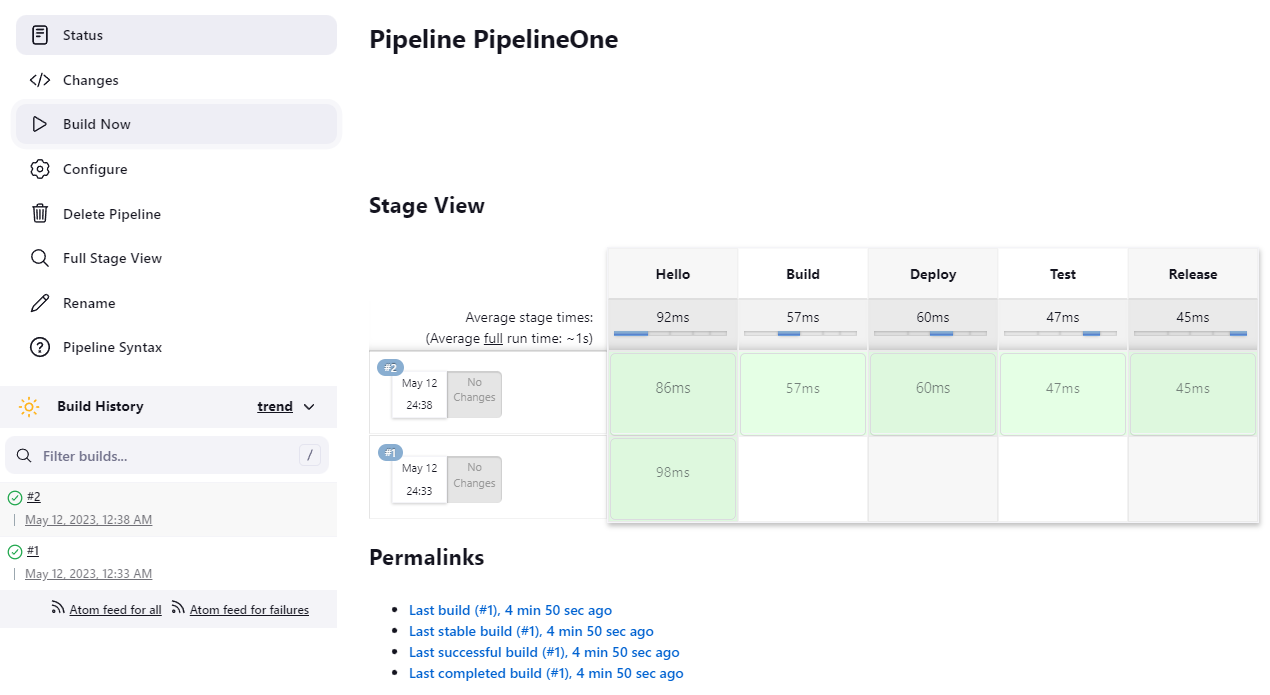
- when press Build Now, we would get the following result:



- we can expand the previous script by adding additional stages:

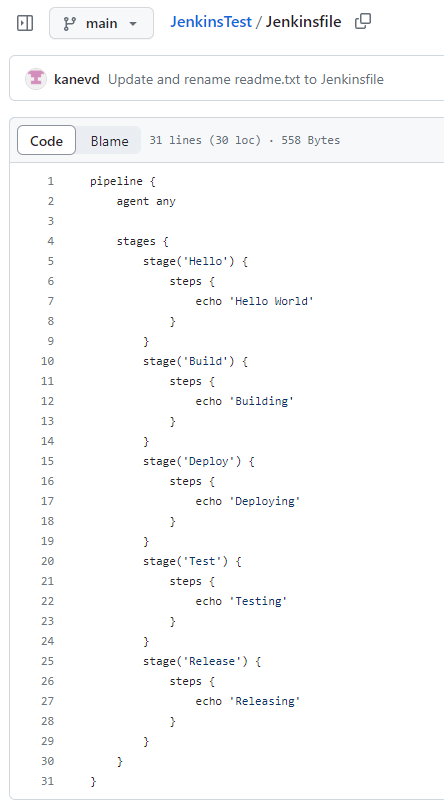


- and build it again:

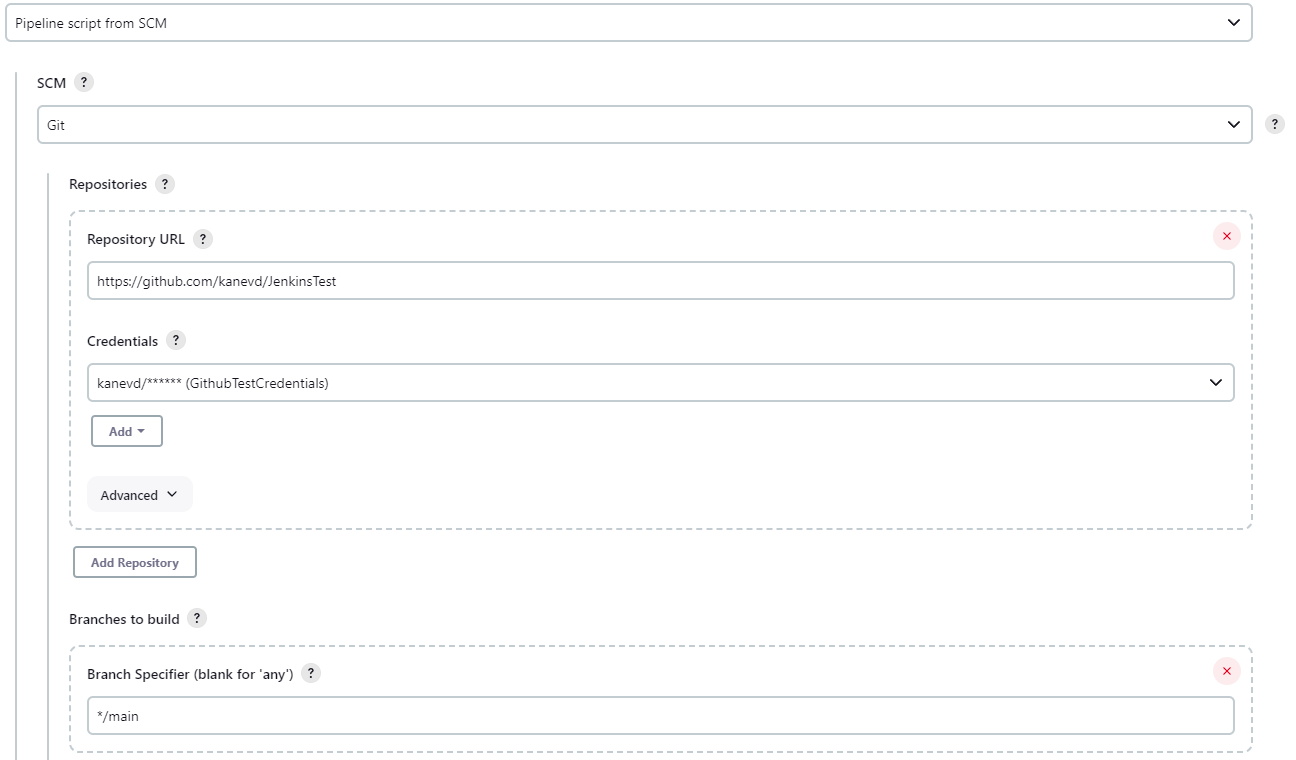


**7. Jenkinsfile from Git SCM.**

- here, I created a JenkinsTest public repository and a jenkinsfile into it, adding the previous script from above:



- back to our Dashobard -> PipelineOne -> Configure -> scroll down to Pipeline and select Pipeline script from SCM and add the following settings:



- then build it again:

