**Experiment 1**

**Aim:** Installation of Jenkins and execution of a simple job in Jenkins.

**Procedure**:

**Step 1**: Install java on the system. The command to install java on the linux system is:

yum install java-11-openjdk.x86\_64 –y.

**Step 2:** Install Jenkins using the following command:

wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -

sudo sh -c 'echo deb https://pkg.jenkins.io/debian binary/ > \

/etc/apt/sources.list.d/jenkins.list'

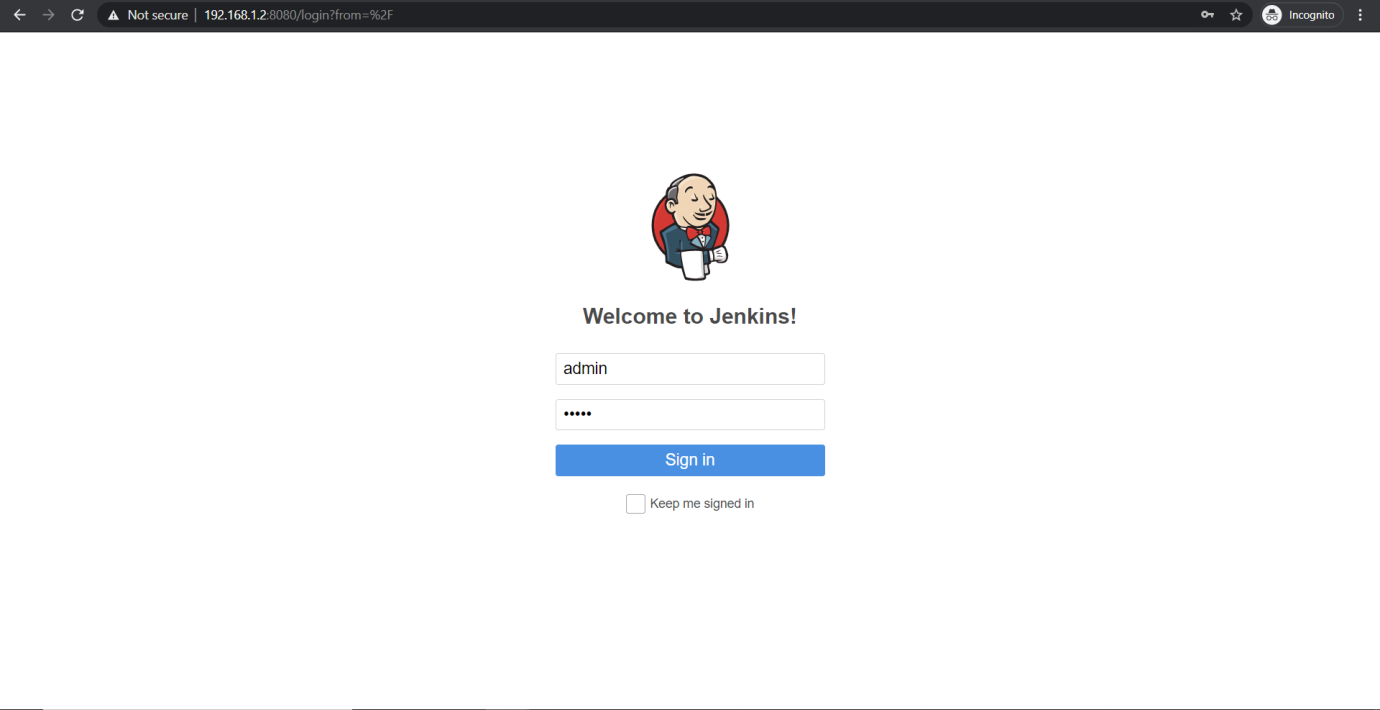
sudo apt-get update

sudo apt-get install jenkins

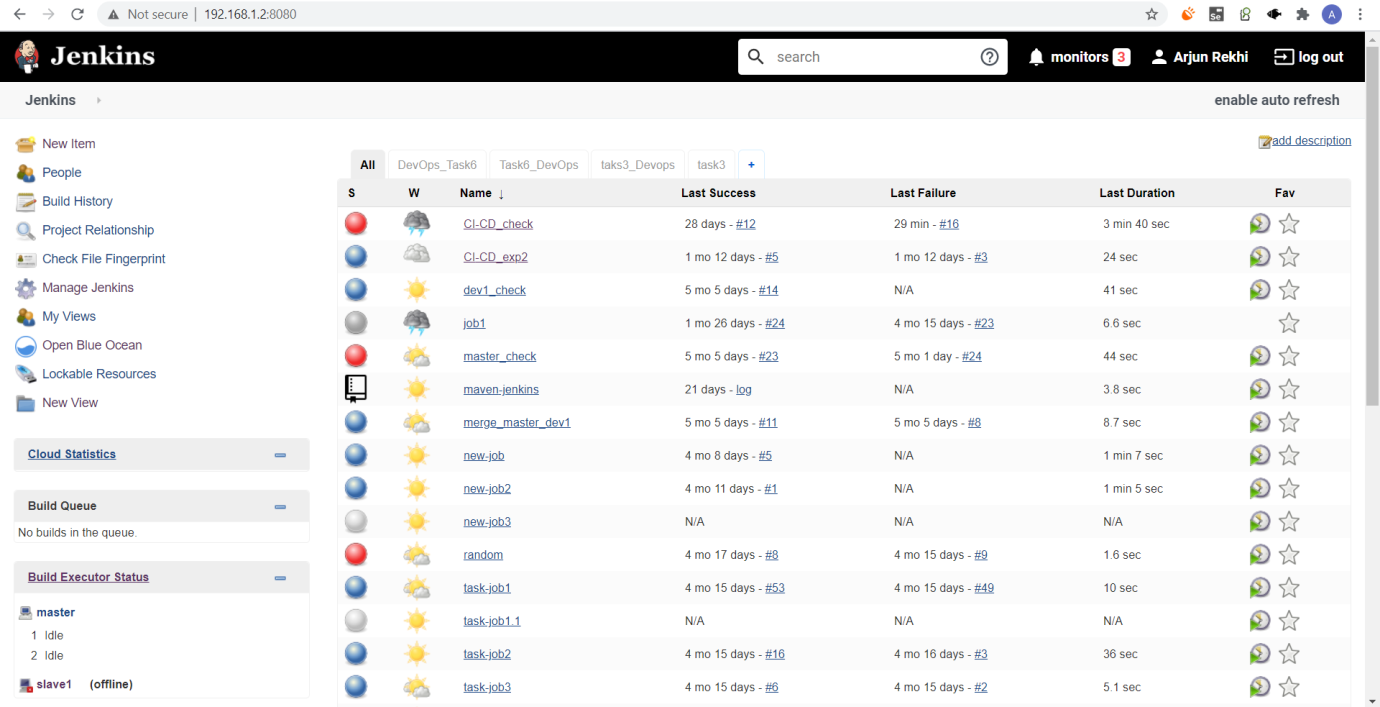
**Step 3:** Start Jenkins using the following command:

systemctl start Jenkins

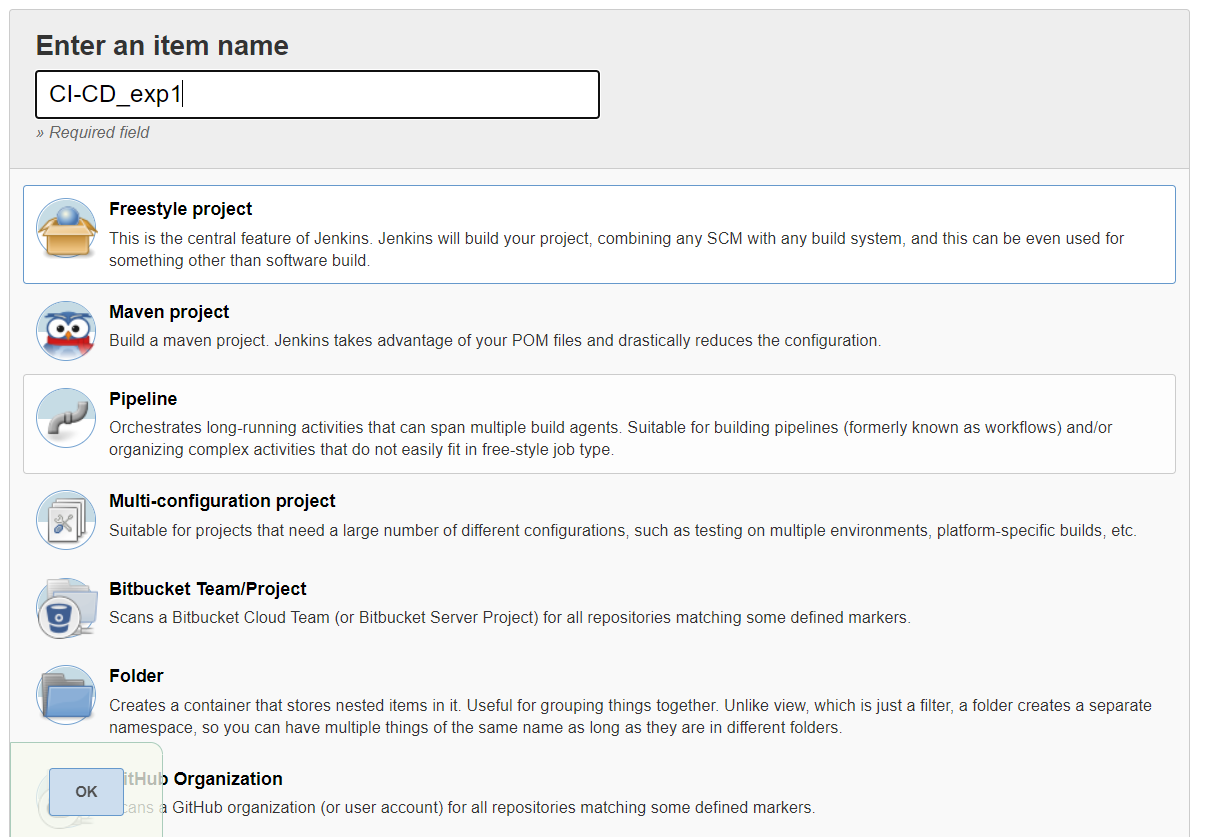
**Step 4:** Access the Jenkins dashboard on port number 8080.Give the initial password from the location given and install the suggested plugins.



**Step 5:** Click on new item and create a new job.



**Step 6**: Configure the job as follows:





**Step 7:** The output as seen in console output:

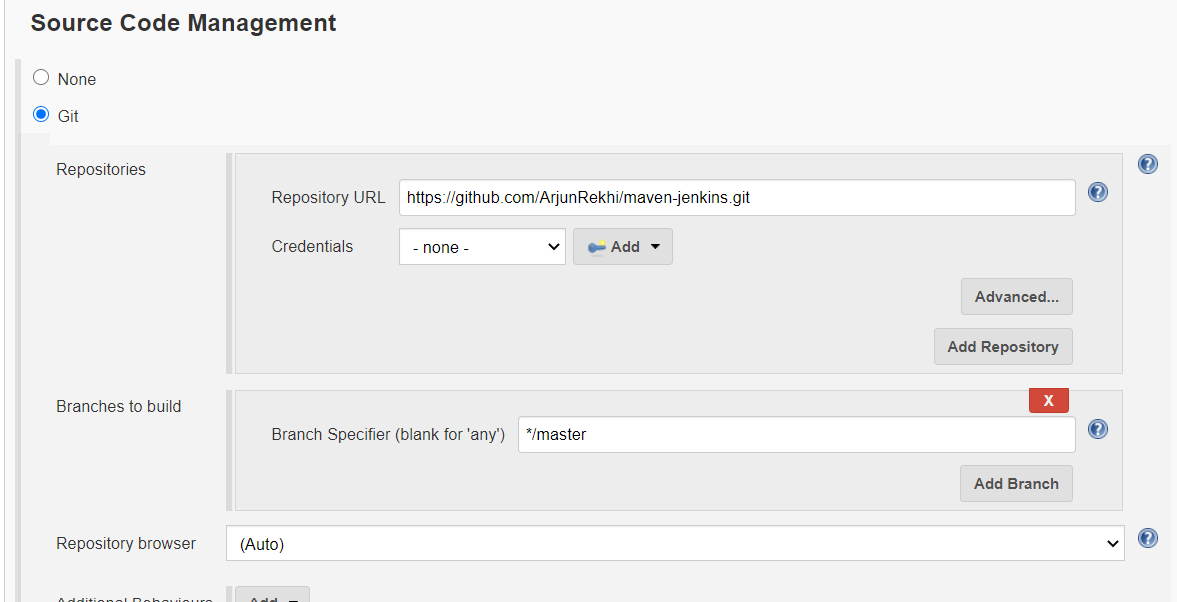


**Experiment 2**

**Aim:** Jenkins integration with github

**Procedure:**

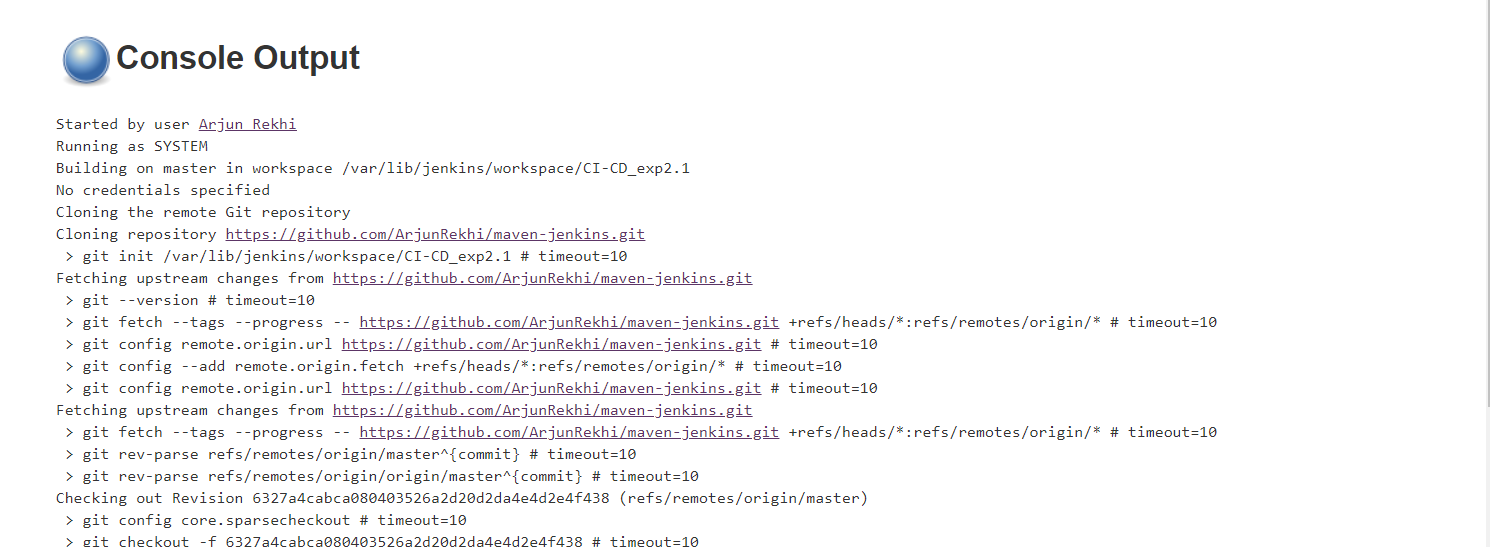
**Step 1:** Create a new job and configure it as follows:



****

**Step 2:** Click on build now and start the job. Jenkins will fetch the code from github and copy its contents in its workspace and will then execute the mvn clean command on the project.

**Step 3:** The output as seen in the console output after the job is completed



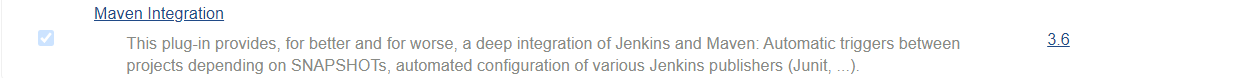
****

**Experiment 3**

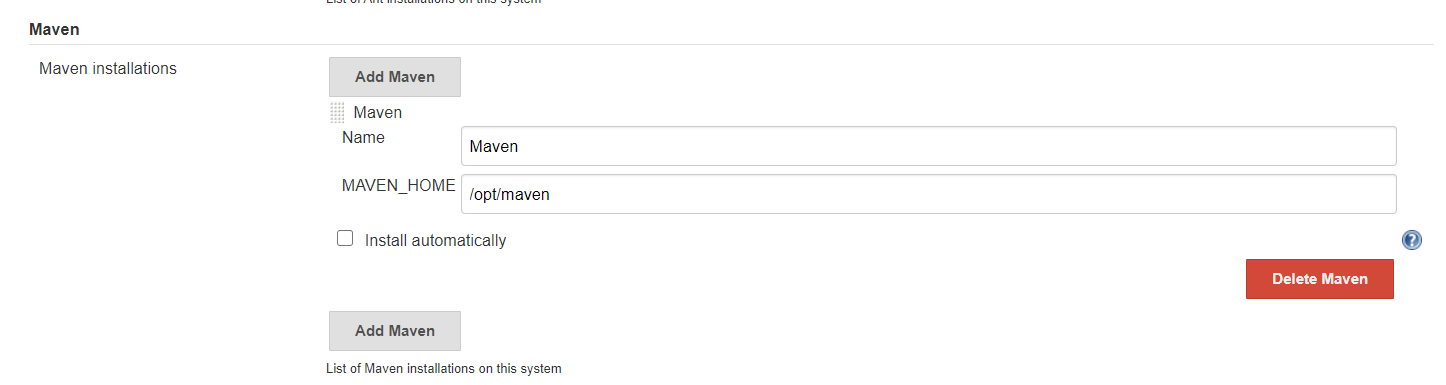
**Aim:** Jenkins integration with maven and github.

**Procedure:**

**Step 1:** Install the following plugin

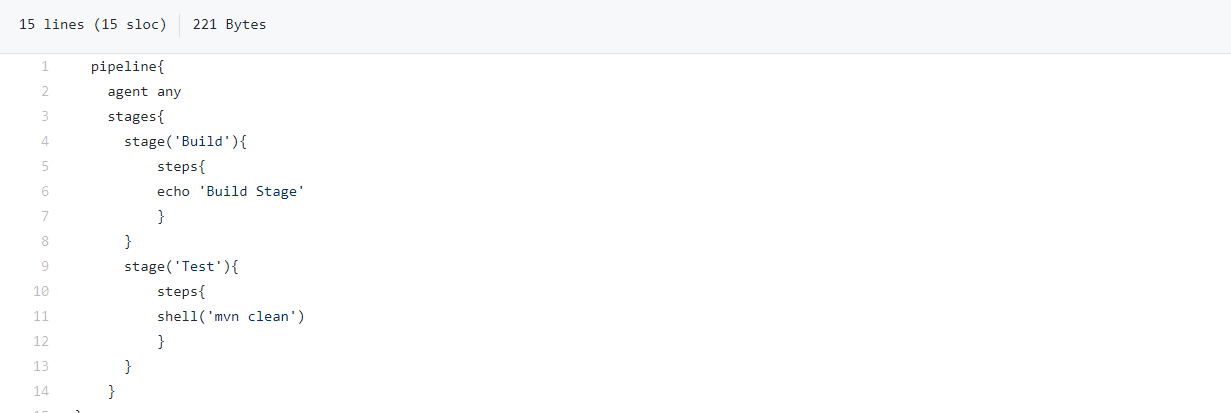
****

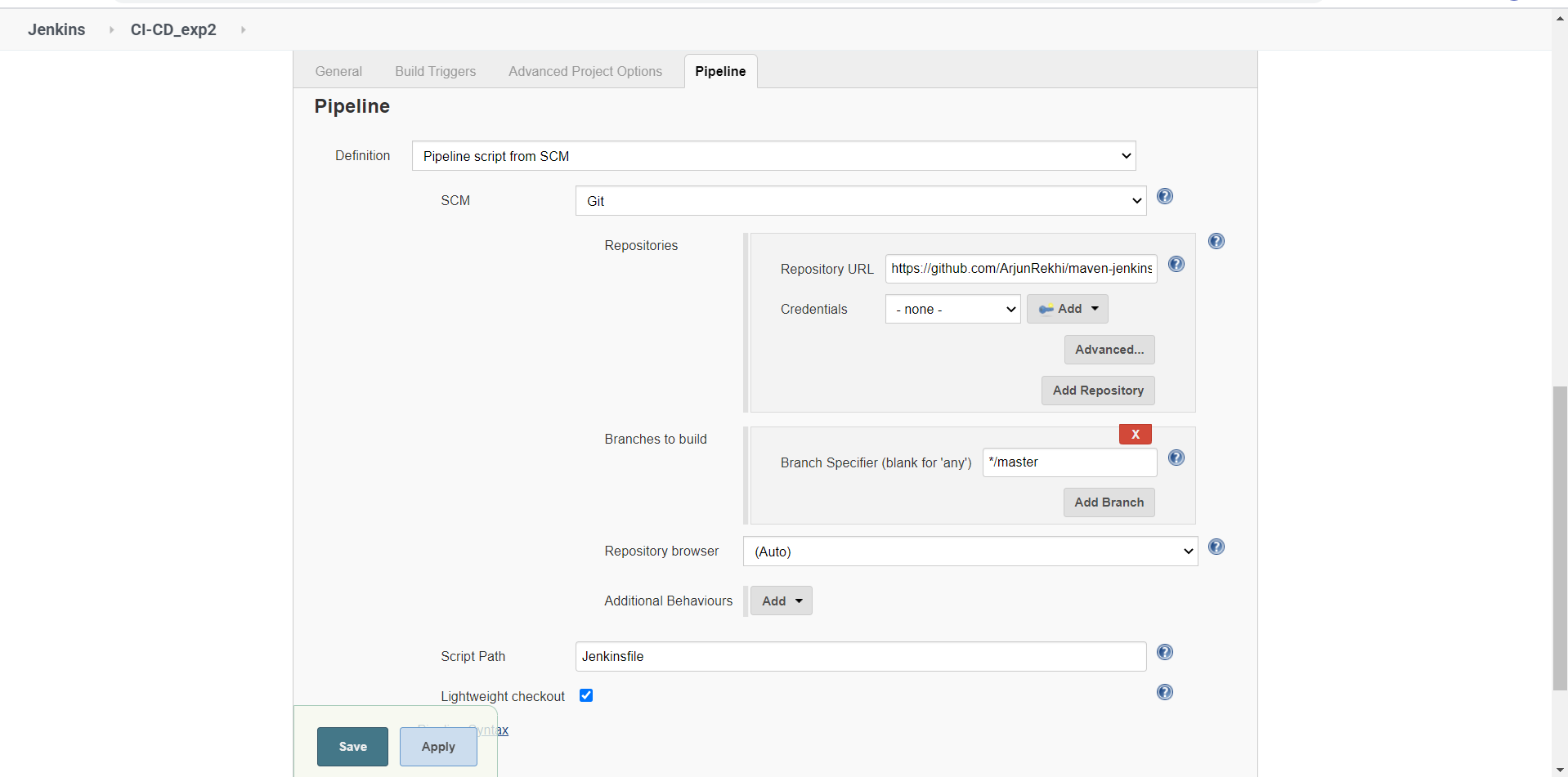
**Step2:** Give the path where the maven is installed in global tool configuration.

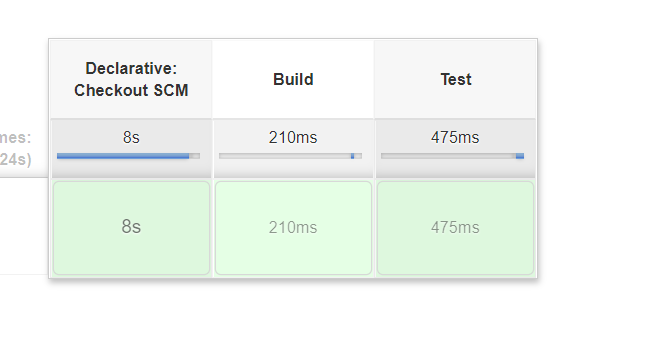


**Step 3:** Configure the job to get the code from github and the jenkinsfile from github then run the job. Here the pipeline code will configure the pipeline such that first is the build phase and will echo ‘build stage’ as output and will then configure the second part of the pipeline as Test and run the ‘mvn clean’ command on the project.

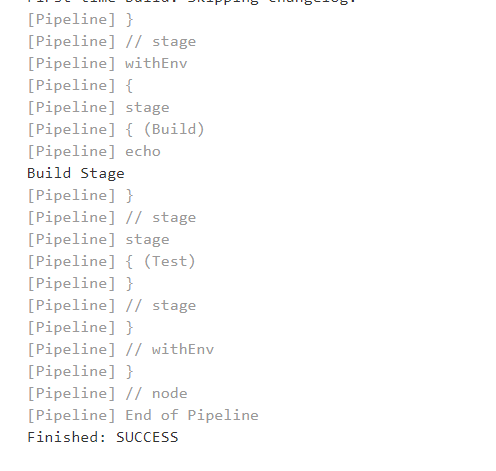
The pipeline code:



****

****

**Step 4:** Output as seen in the console output

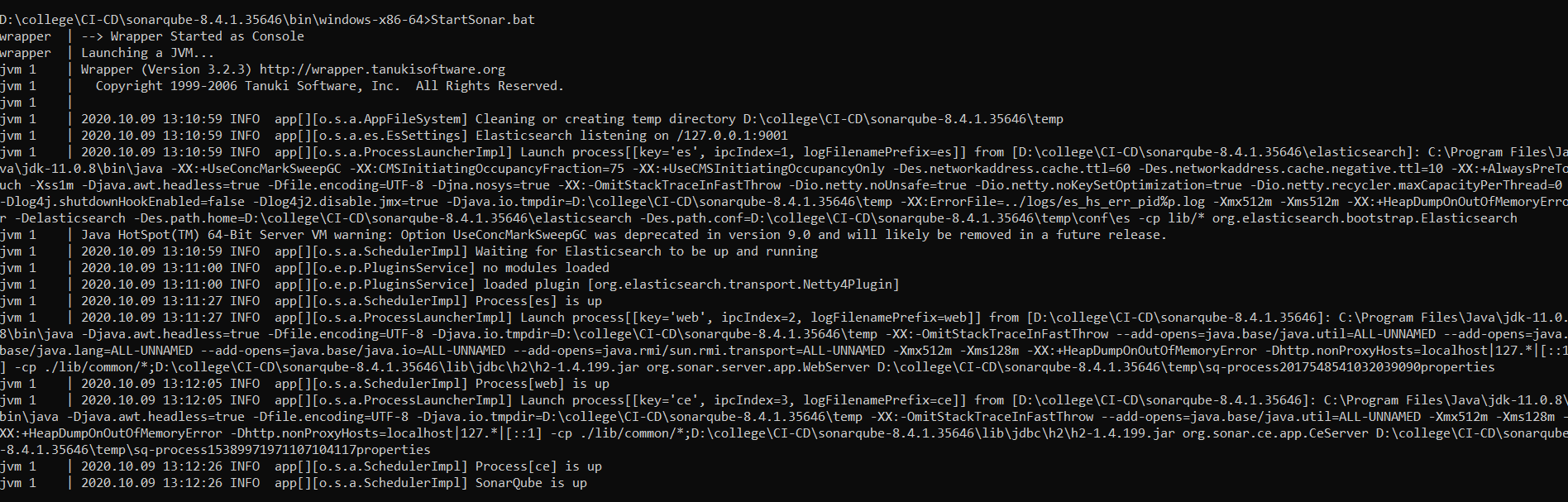
****

**Experiment 4**

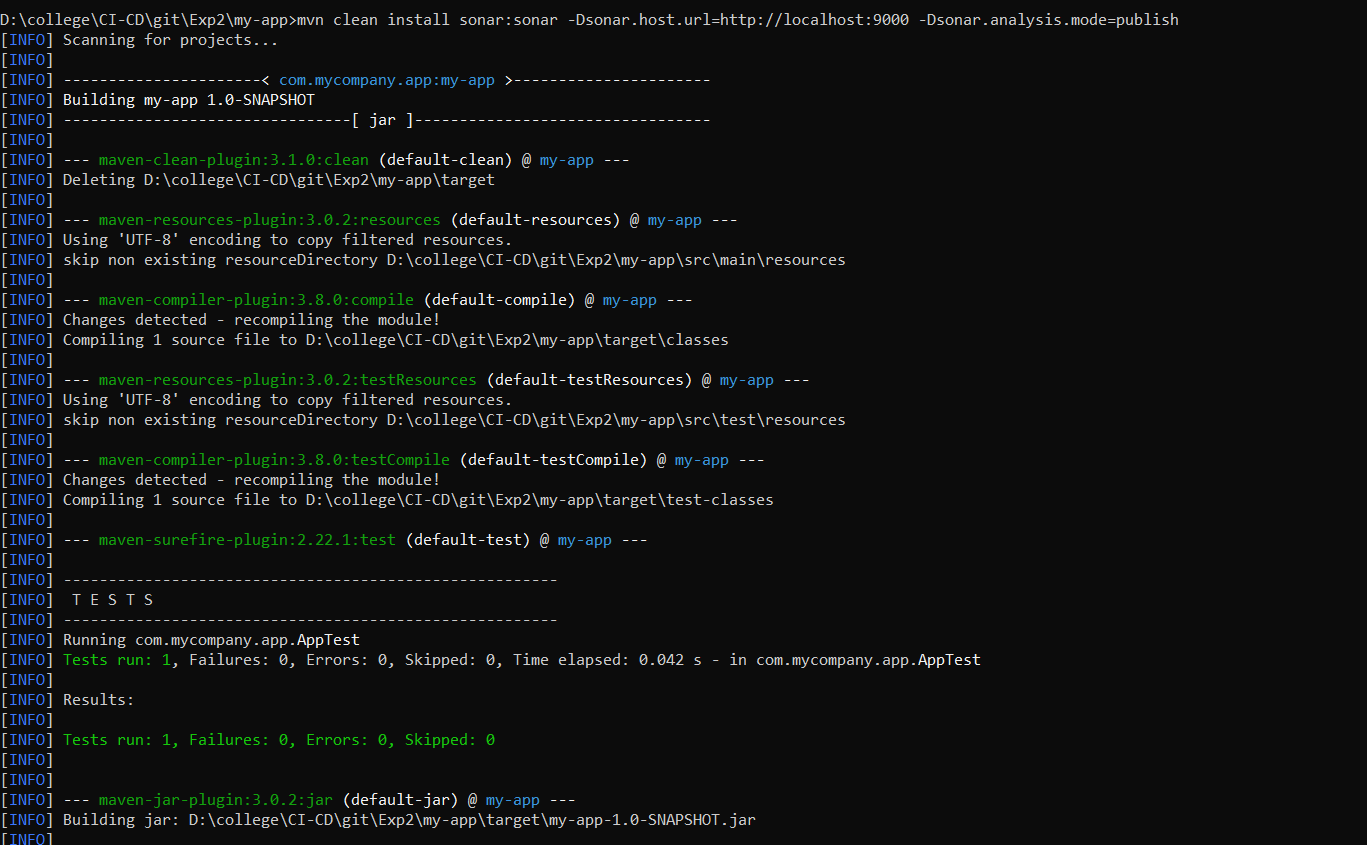
**Aim:** Static code analysis using Sonarqube.

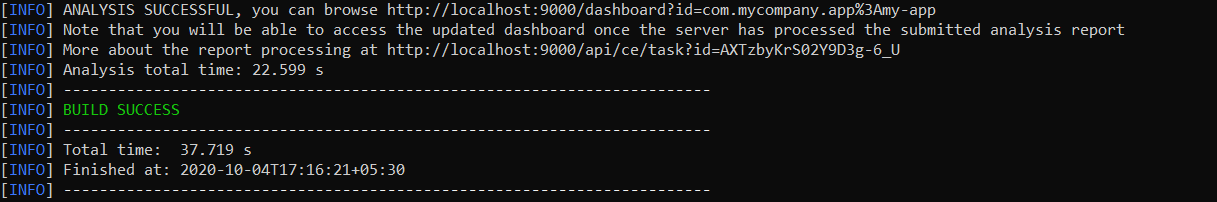
**Procedure**:

**Step 1:** Start sonarqube using the StartSonar.bat command.

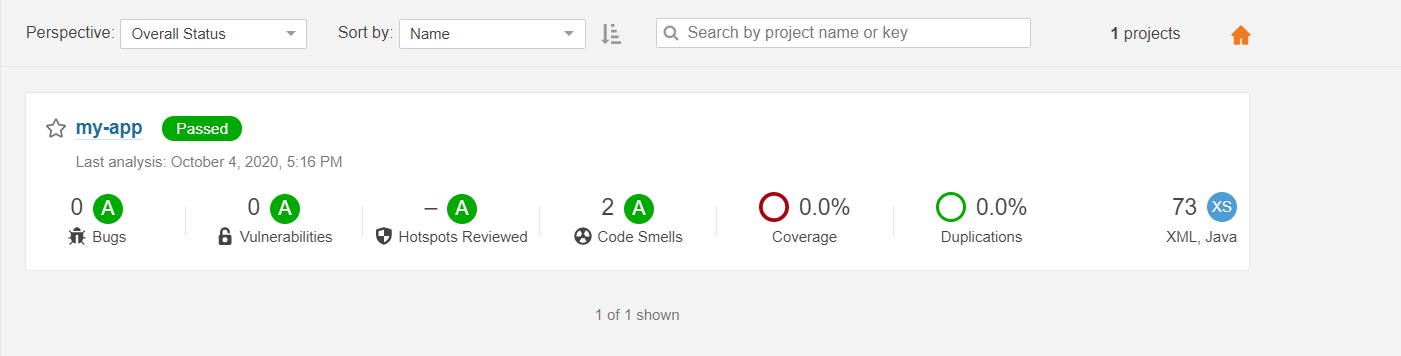


**Step 2:** Run the following command to upload the code on sonarqube and execute the static code analysis on it. The output on running the command





**Step 3:** Sonarqube runs on port number 9000. To open the dashboard write localhost:9000. The output as seen on the dashboard.



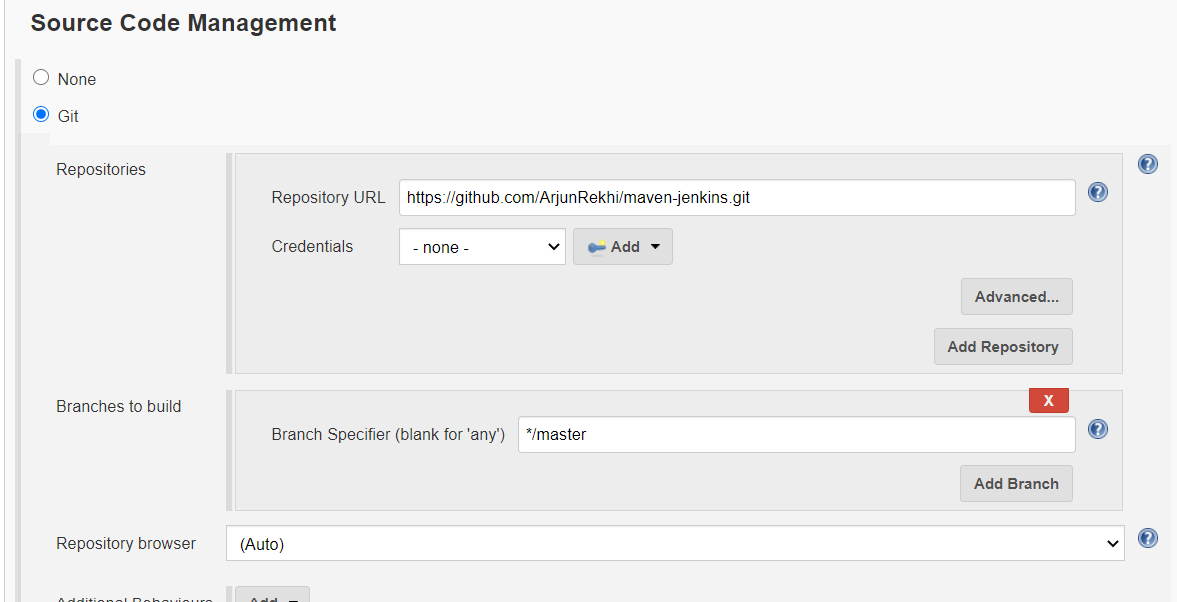
**Experiment 5**

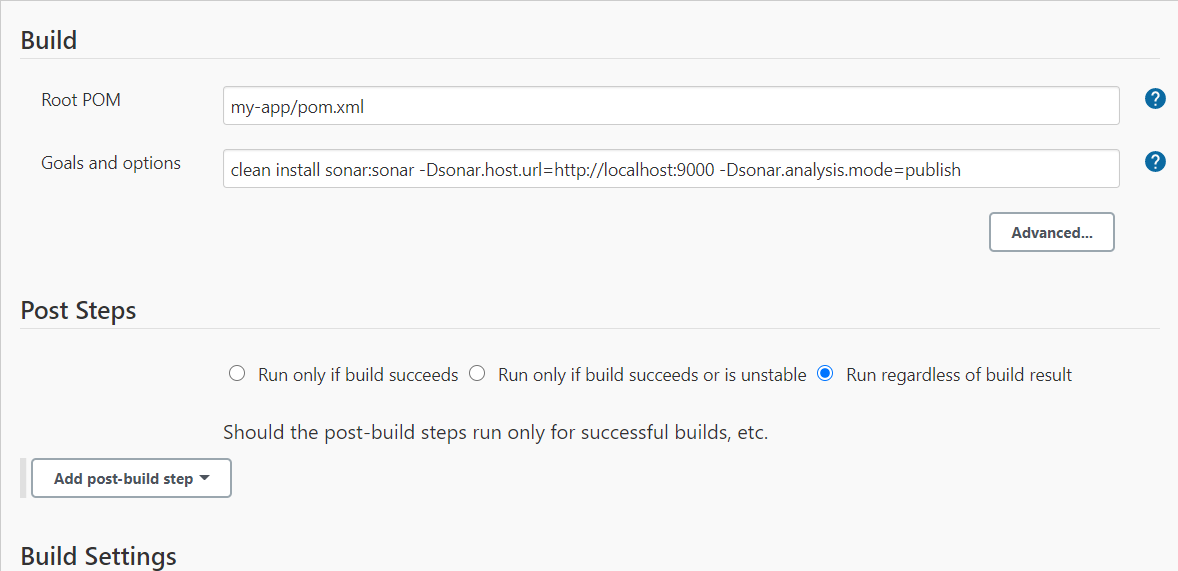
**Aim:** Jenkins integration with Sonarqube and github.

**Procedure**:

**Step 1**: Start Jenkins and Sonarqube

**Step 2**: Configure the Jenkins job to deploy the code on sonarqube so that it can perform static code analysis on it. Configure the job as follows:

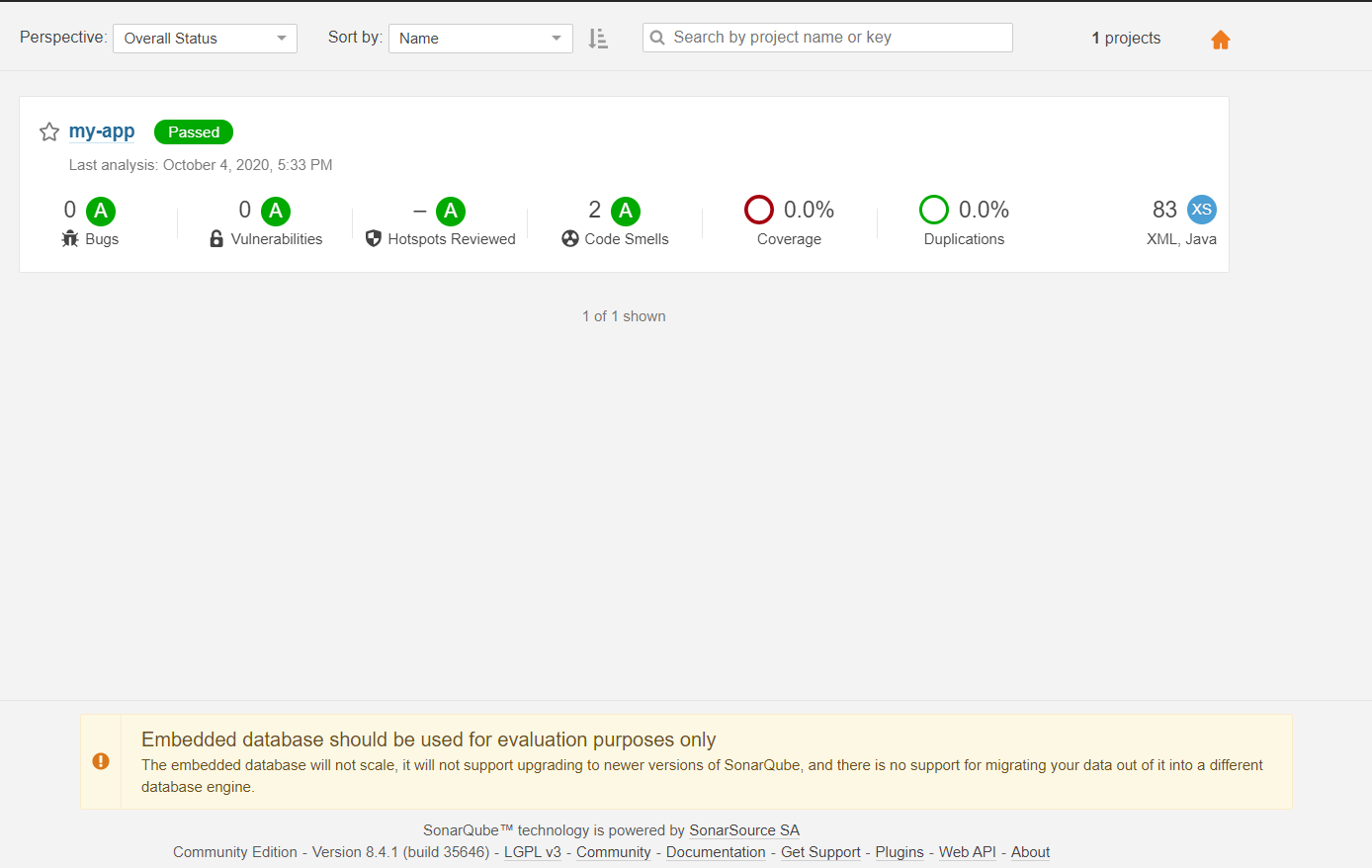




**Step 3:** After this job the code has been deployed on sonarqube and it will perform static code analysis on it. The output as seen in the console output of jenkins.



**Step 4:** The output on the sonarqube dashboard:



Submitted by:

Arjun Rekhi

500068109

R171218028