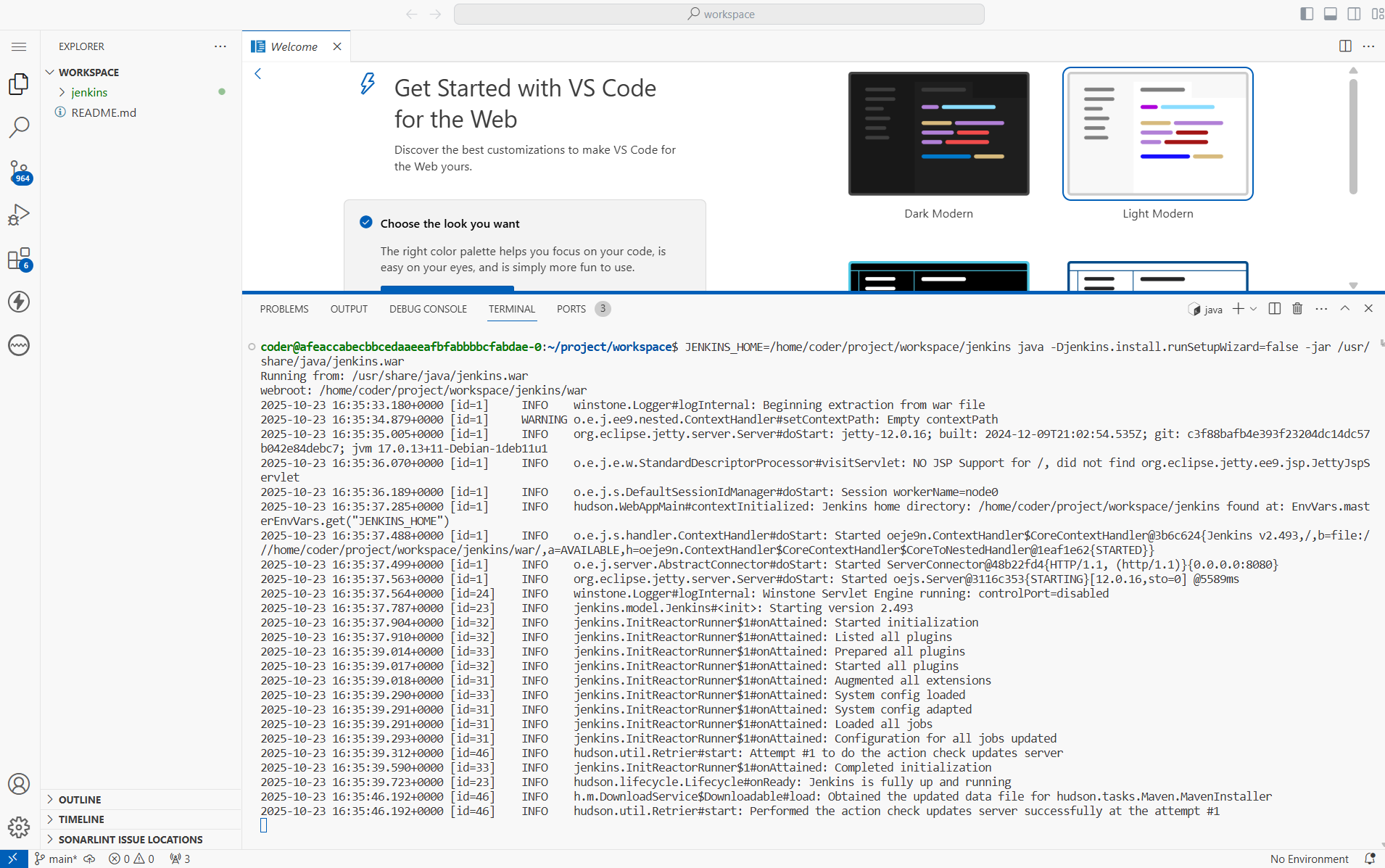
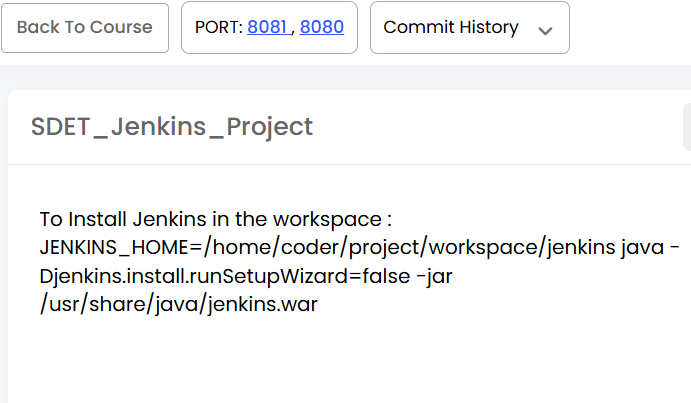
Jenkins Sdet Python

Step 1: Open the workspace of Jenkins Project.  
Step 2: copy the path of Jenkins\_home “JENKINS\_HOME=/home/coder/project/workspace/jenkins java -Djenkins.install.runSetupWizard=false -jar /usr/share/java/jenkins.war”

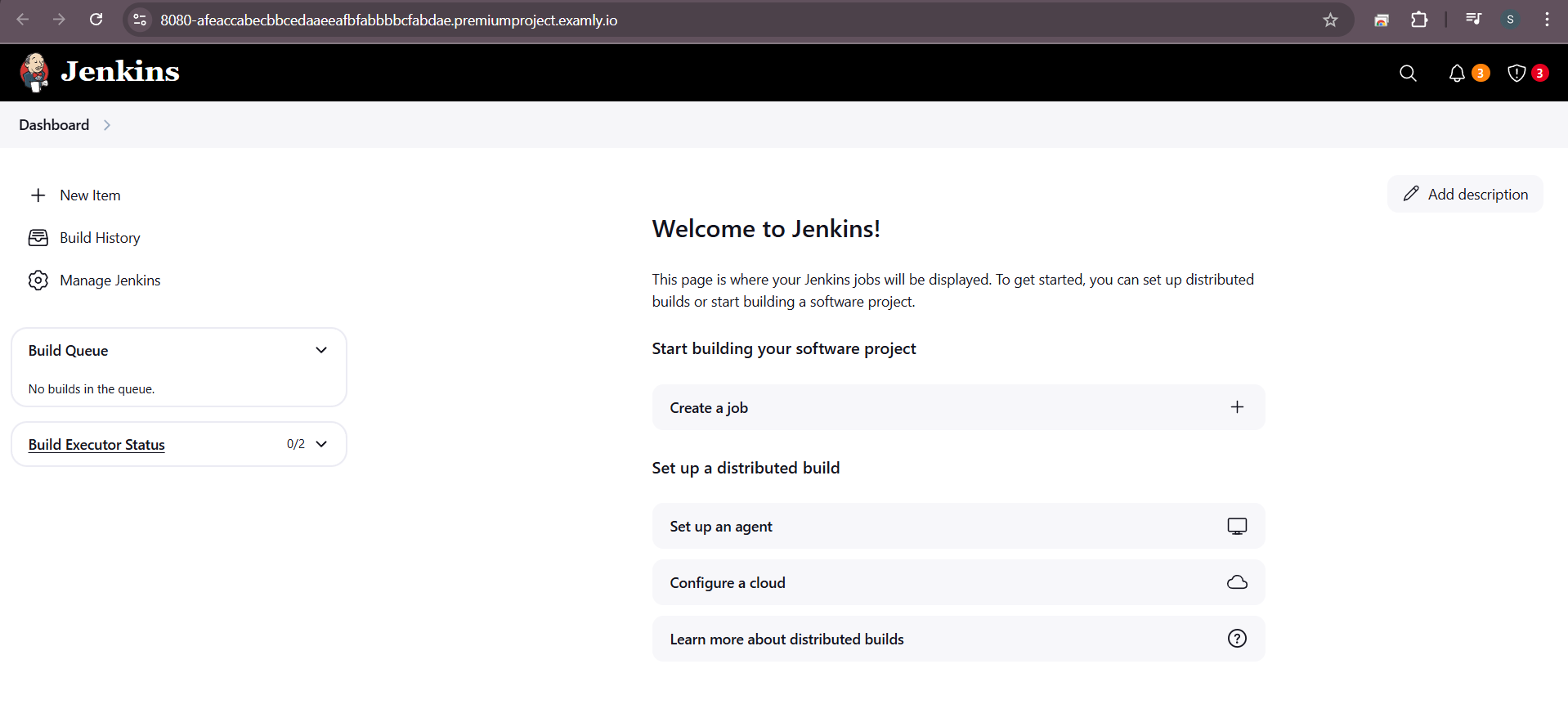
Step 3: open the terminal and paste the jenkins url and past and press enter



Step 4: once the Jenkins is running successful. Kindly click on 8080 port.



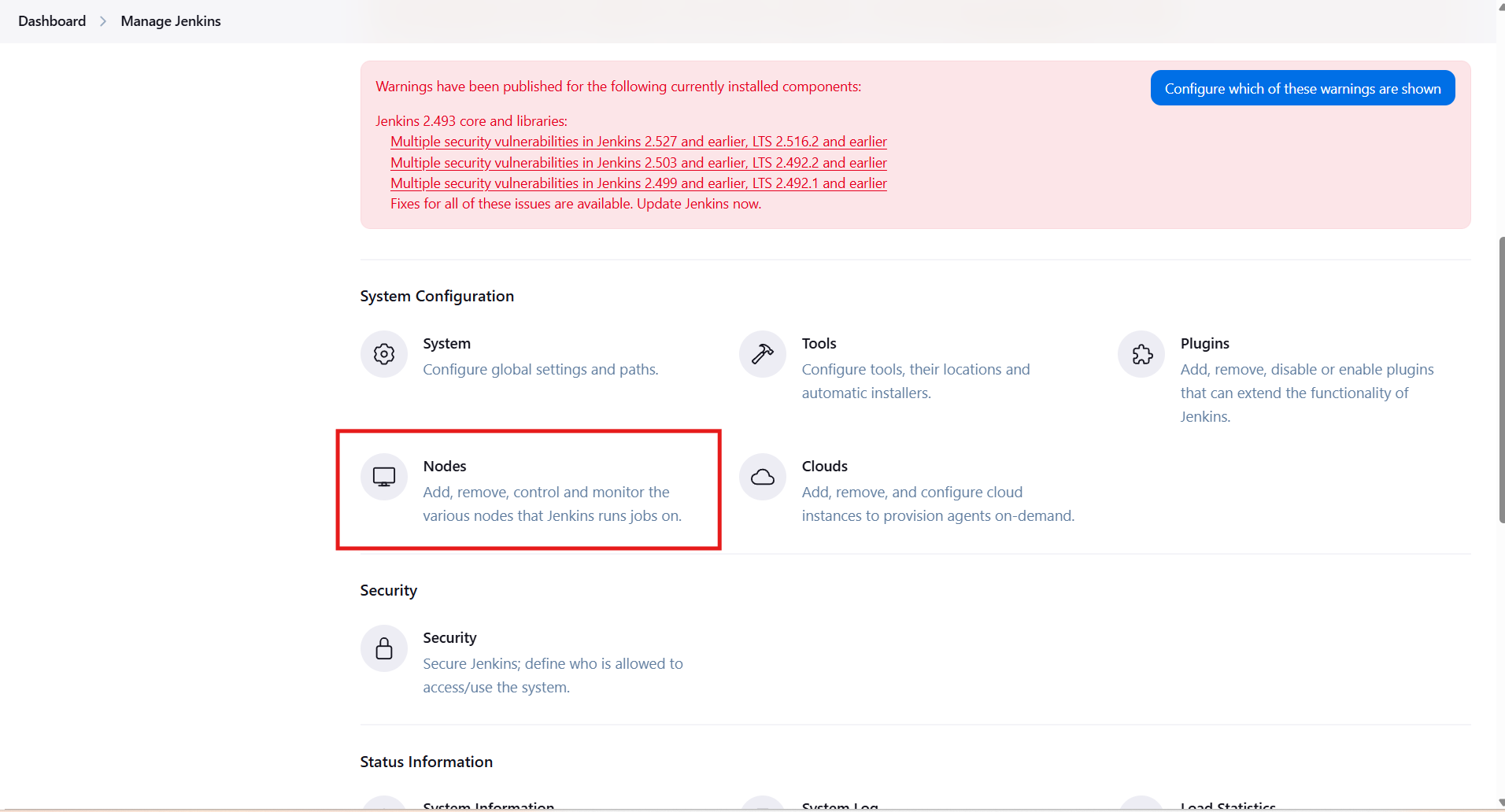
Step 5: once click on 8080 it will open the dashboard of Jenkins.



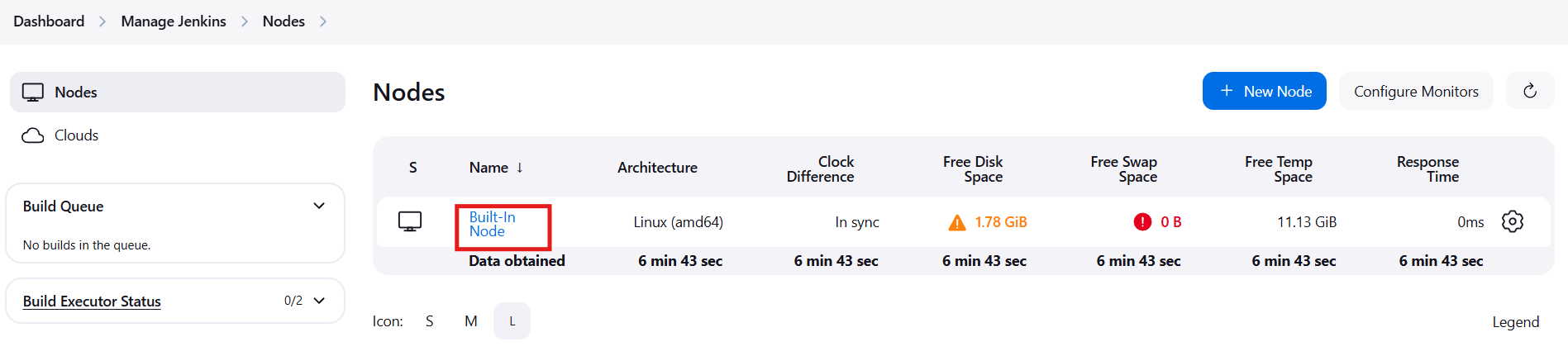
Step 6: check the node is online or not

Click on manage Jenkins

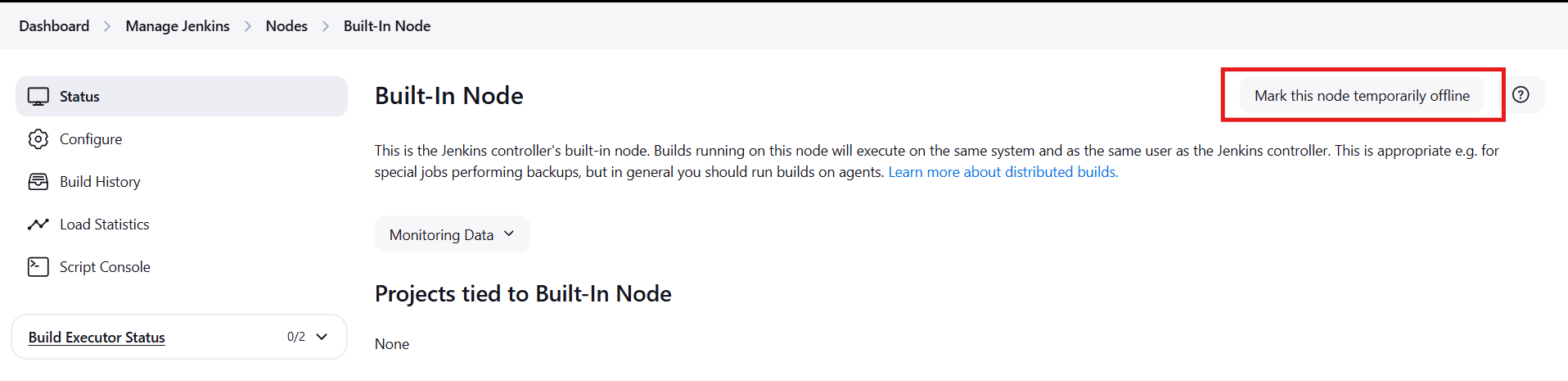
Scroll down and click on Nodes



In that click on built in node



And Verify that this option should be visible



If this is visible that means your node is online

Step 7:

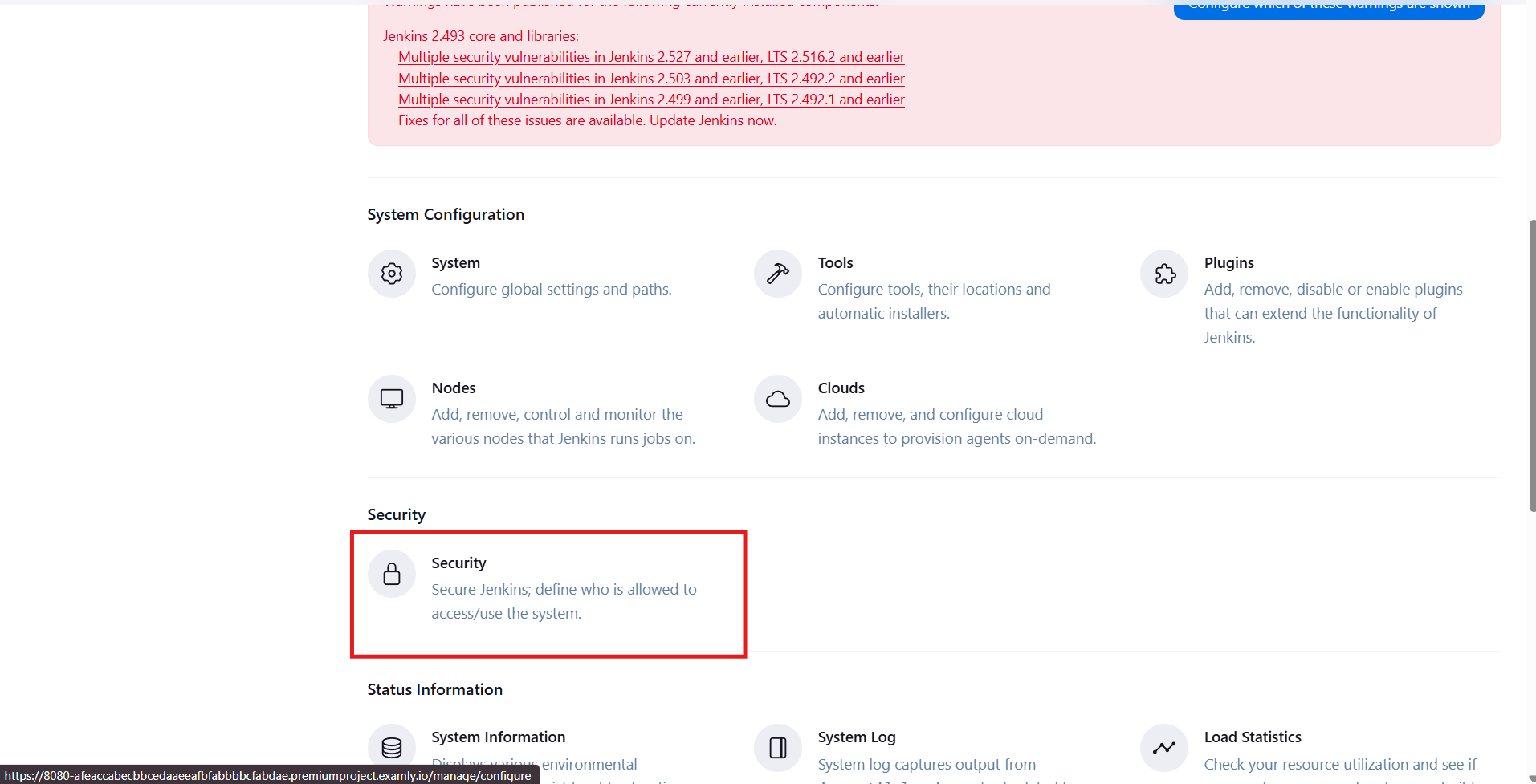
We need to enable the Enable proxy compatibility server.

To this first come to home page of Jenkins by clicking on the Logo.

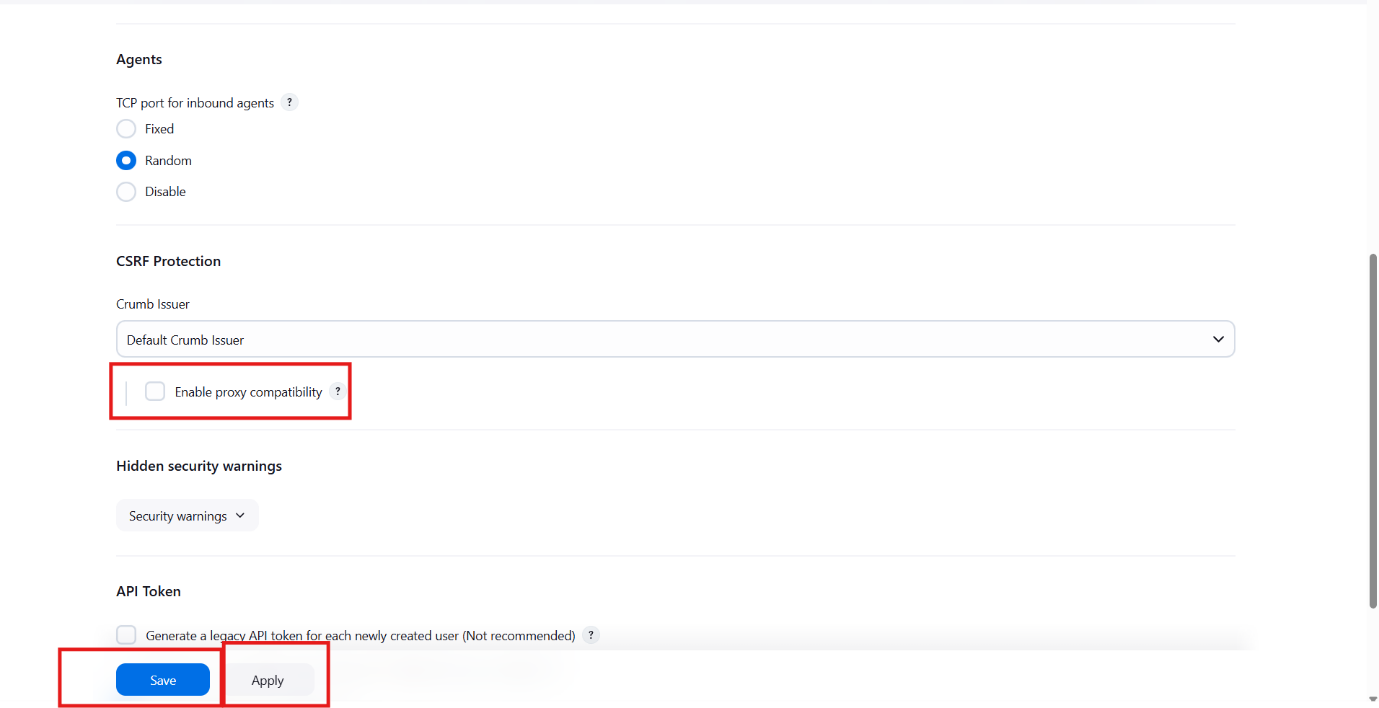


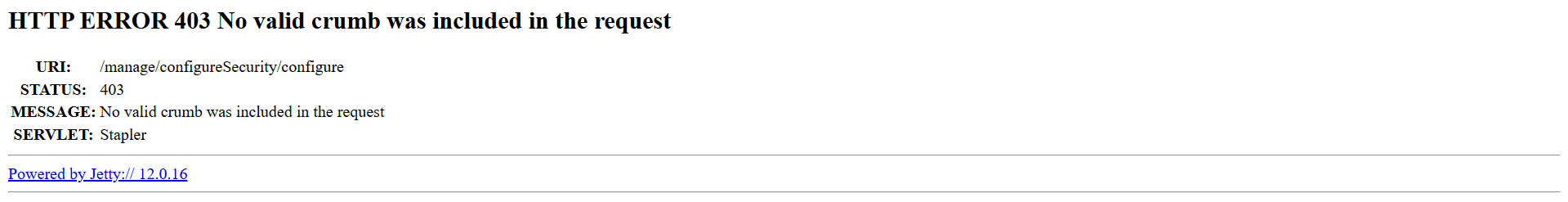
Then click on manage Jenkins

And scrroll down and click on security



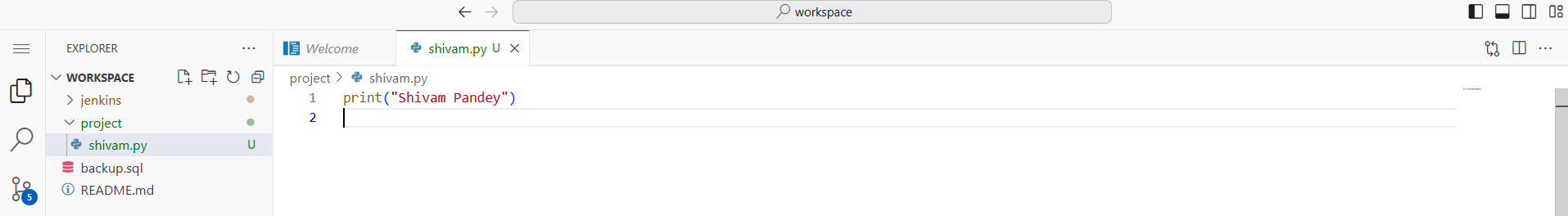
Scroll down till CSRF Protection  
and click on Enable proxy compatibility



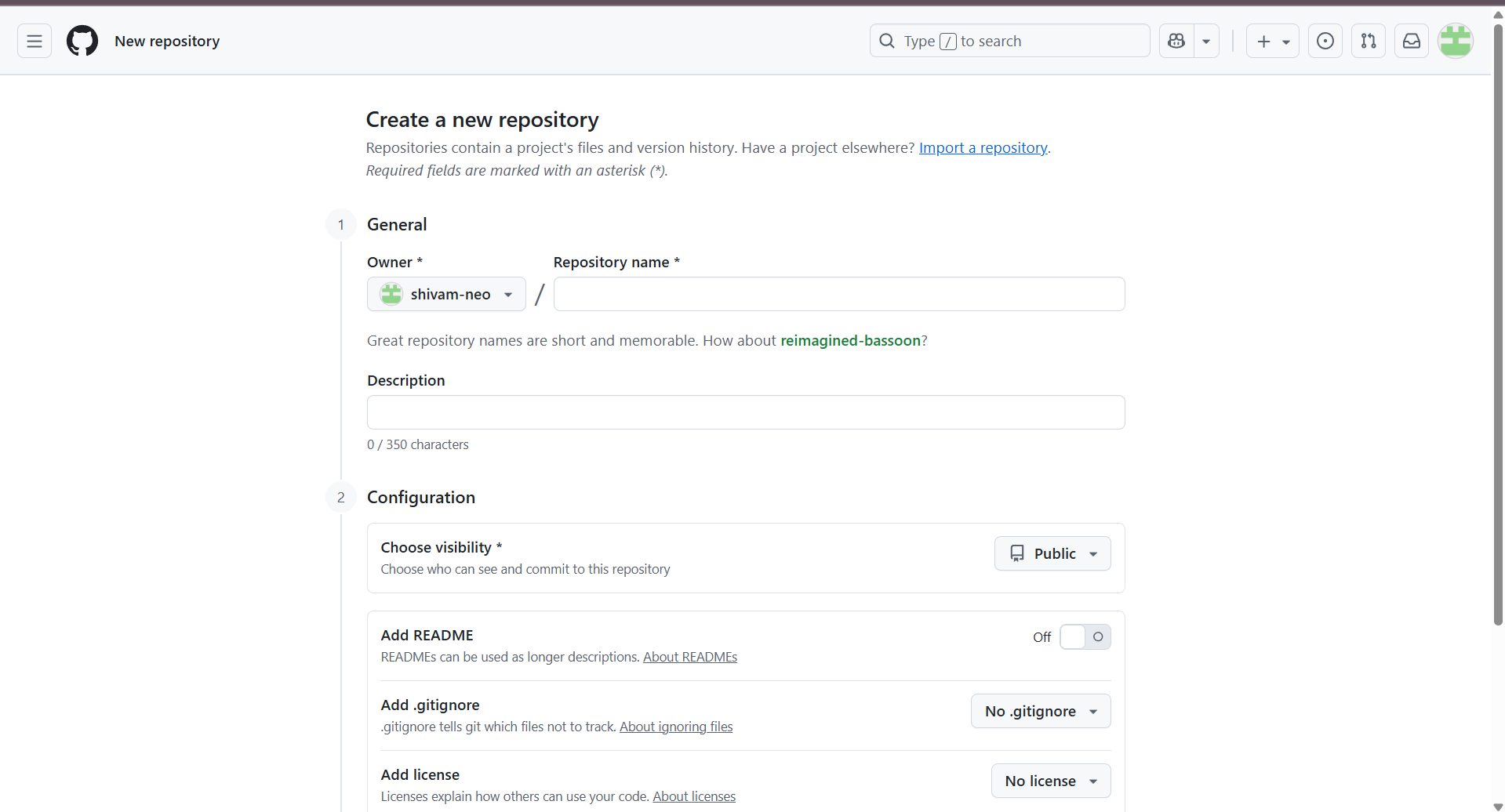
And click on apply and then save   
if error occur   


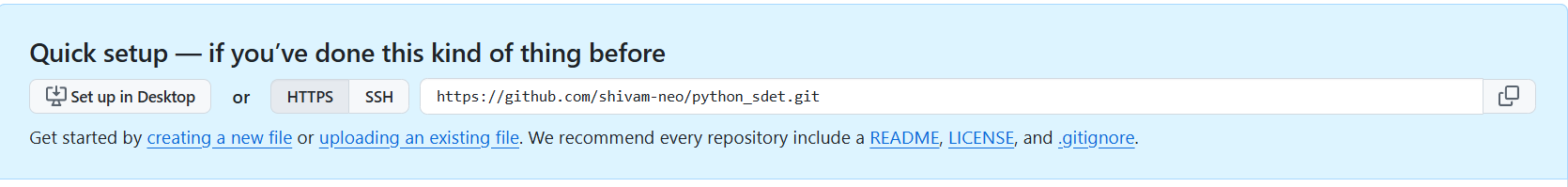
Go back and do the steps again

After save come back to Project workspace create a project and add the python code



After this kindly login the github and create a new repo



After creating the repo kindly copy the Https url.  


After copy kindly and back to workspace and now do the git operations in the new terminal

1. git init
2. git remote add origin url
3. git add .
4. git commit –m"message”
5. git push origin main

Then after push one pop up will come kindly allow that

Then one code will be generated kindly copy the code and click on copy and contiune

Then one more pop up will come copy the url “https://github.com/login/device” and paste in new tab and give the codein that and click on autorized the vs code

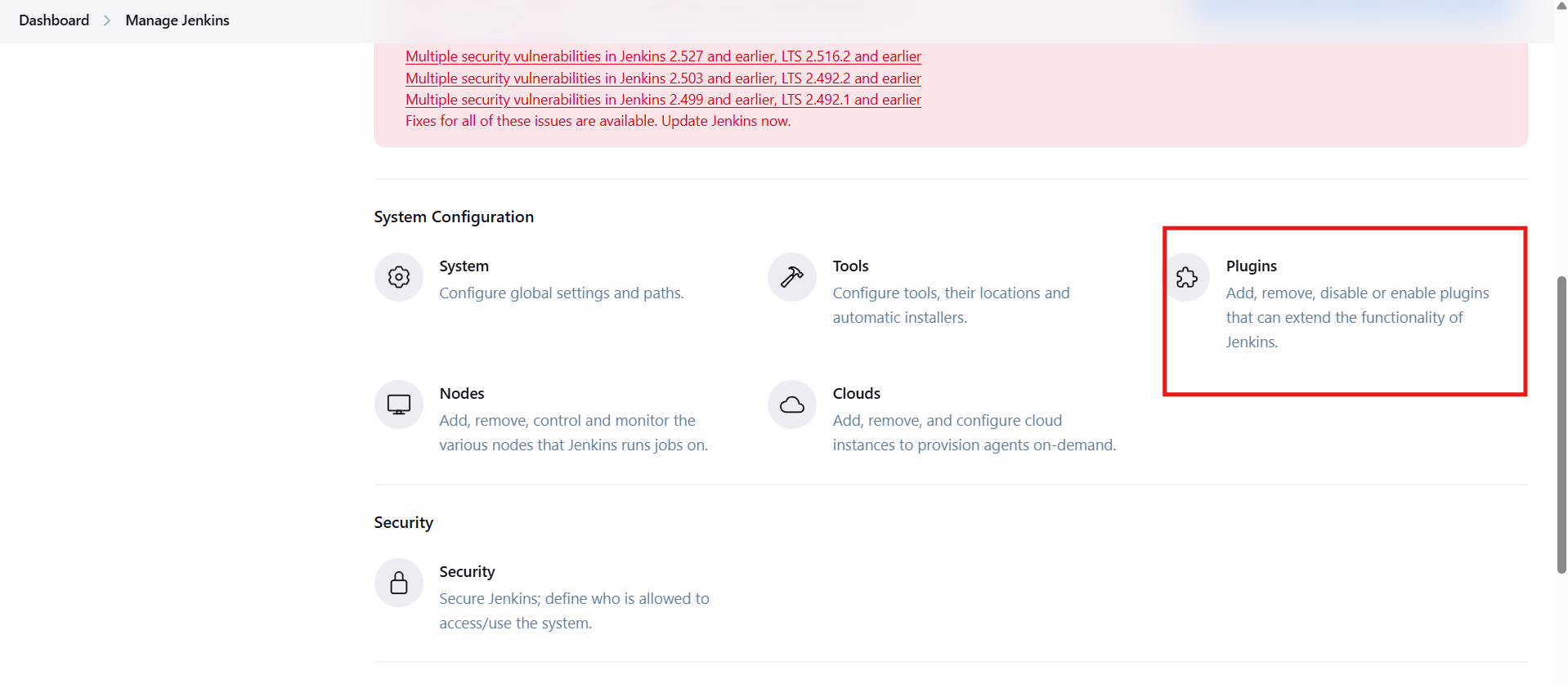
Once authorized the code will be push to the GitHub

Step 8:

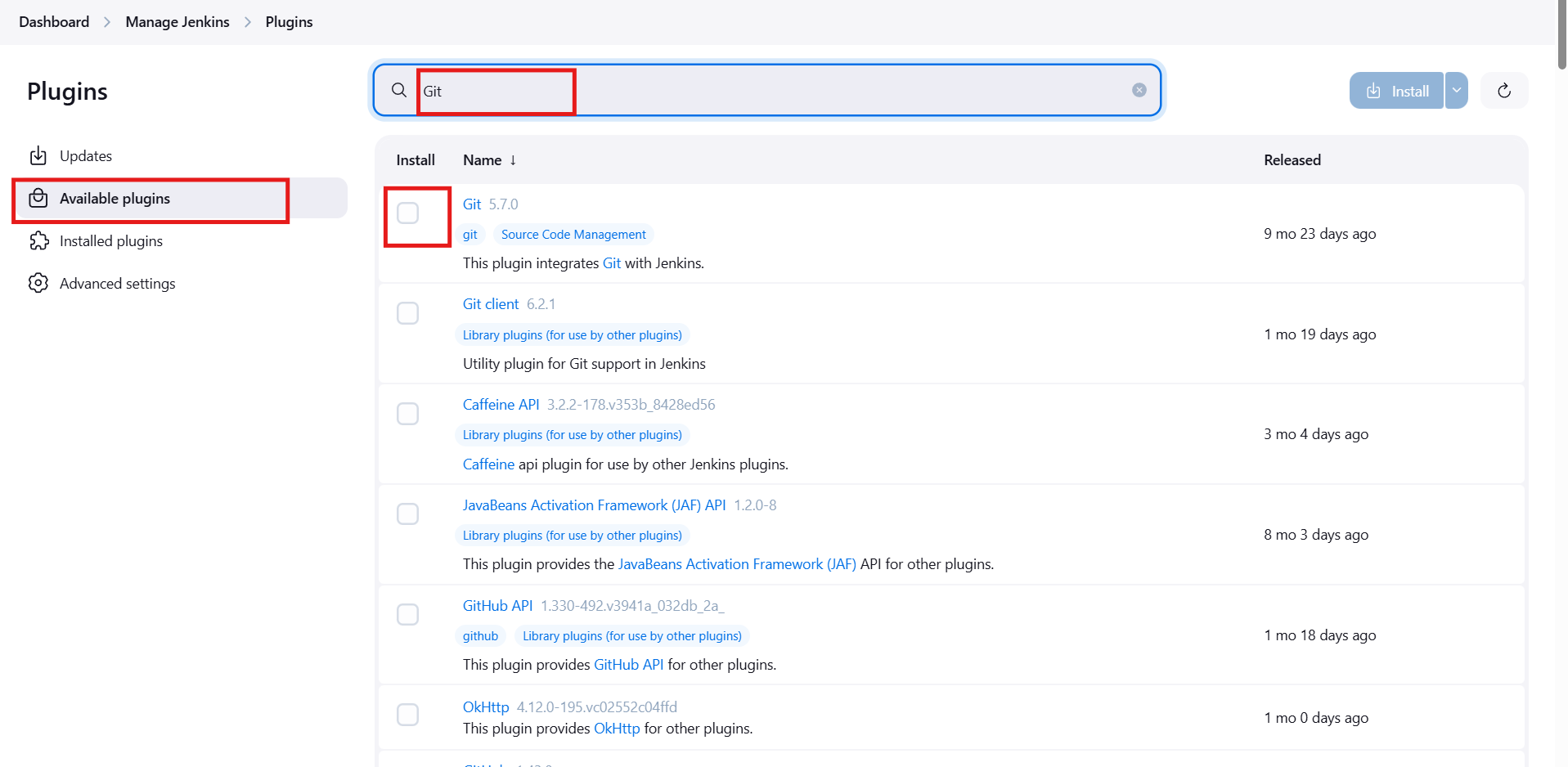
Install the plugin of git in Jenkins

Open the Jenkins port 8080 and go to the dashboard and click on manage Jenkins

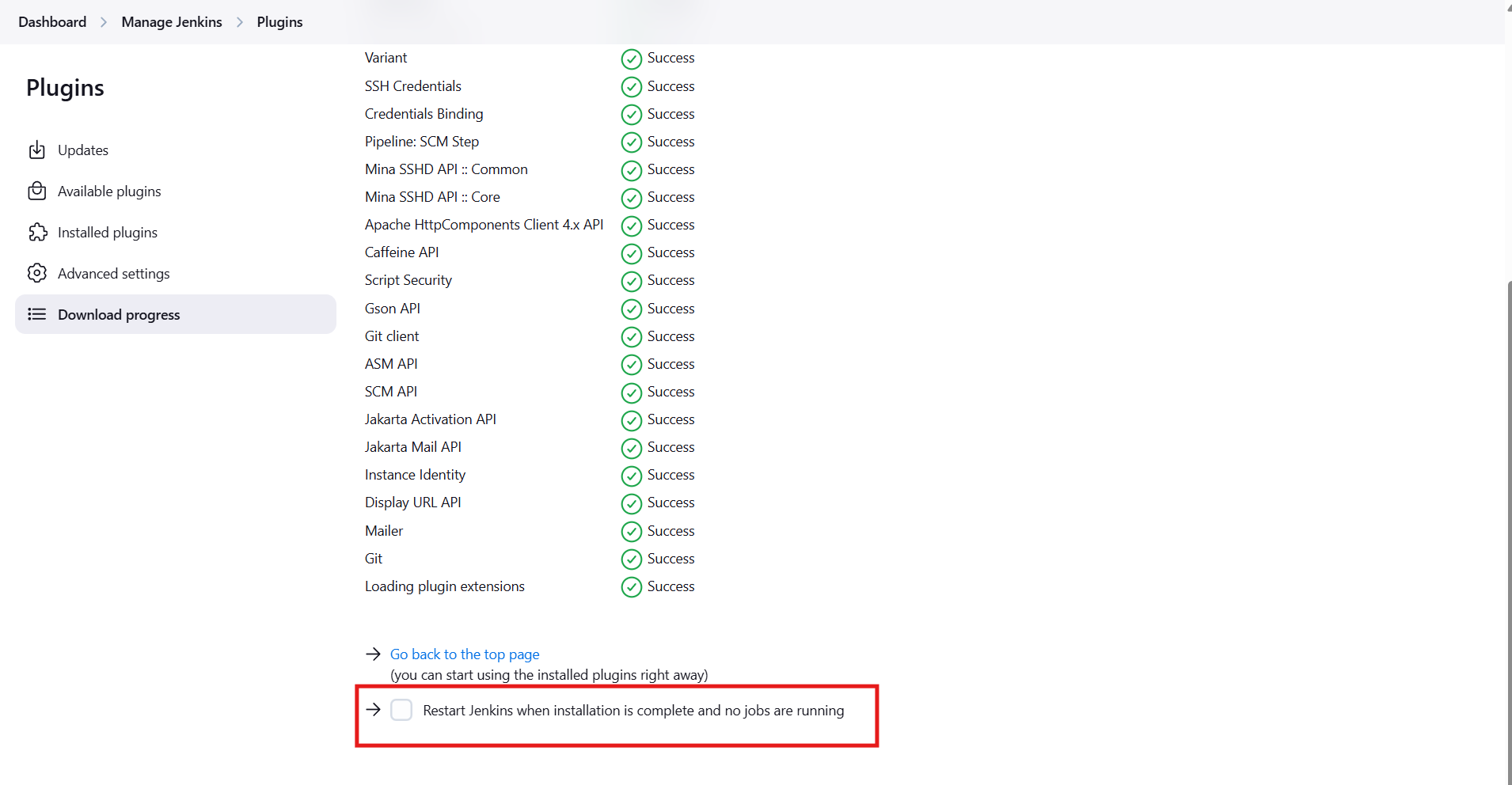
Scroll down and click on Plugins



And then click on available plugins and then click on search bar and enter git and select the first one and click on install



Once install kindly click the checkbox of restart one



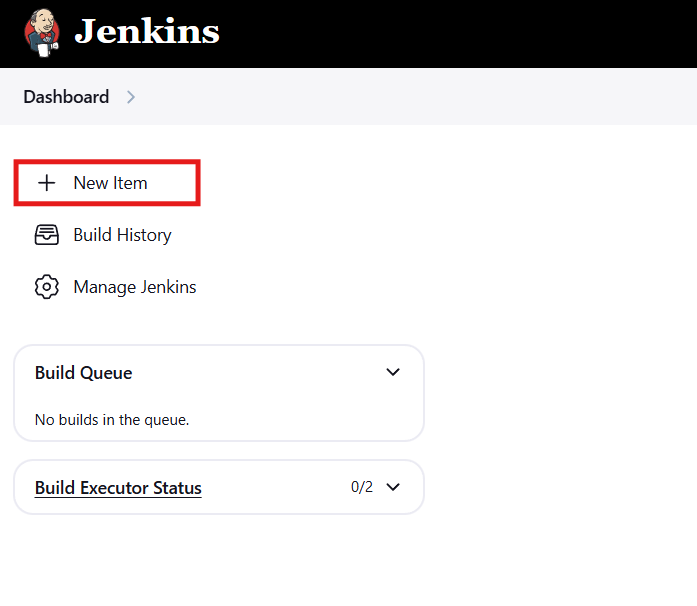
Once done then

Step 9:

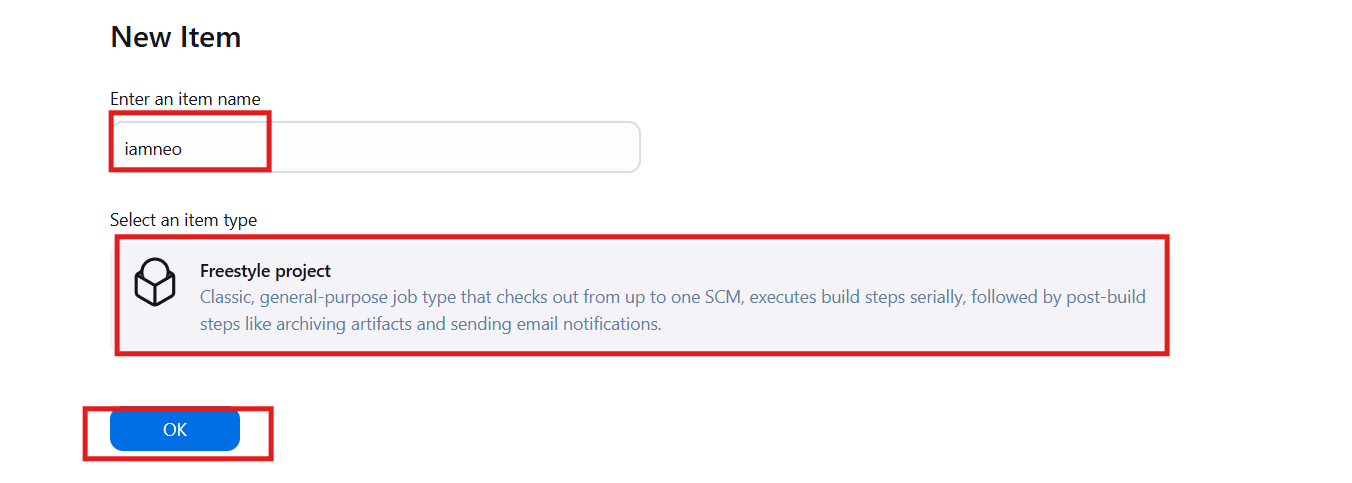
Now need to create a job

Click on Jenkins logo to go back to dashboard

Click on new Item



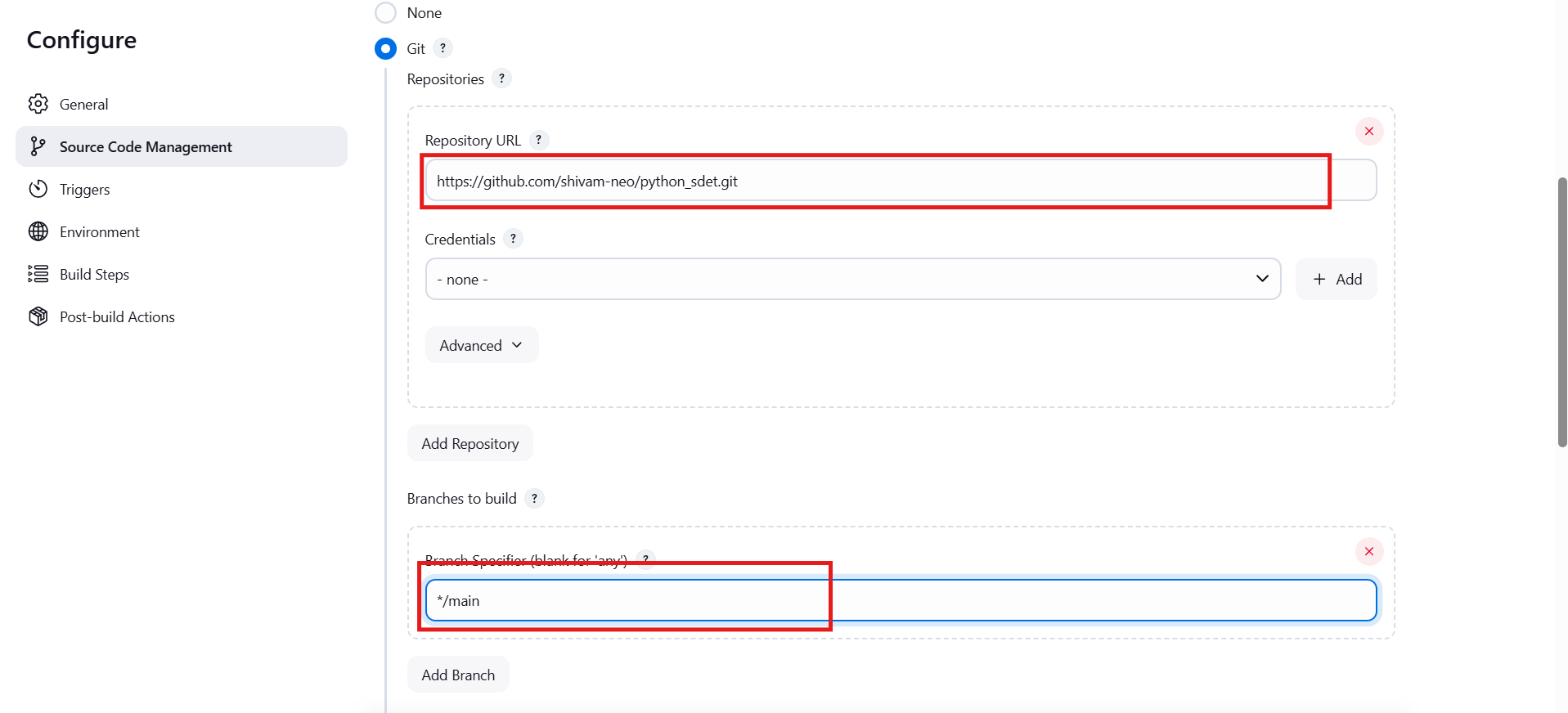
Give the name for the job and click on free style and then click on okay

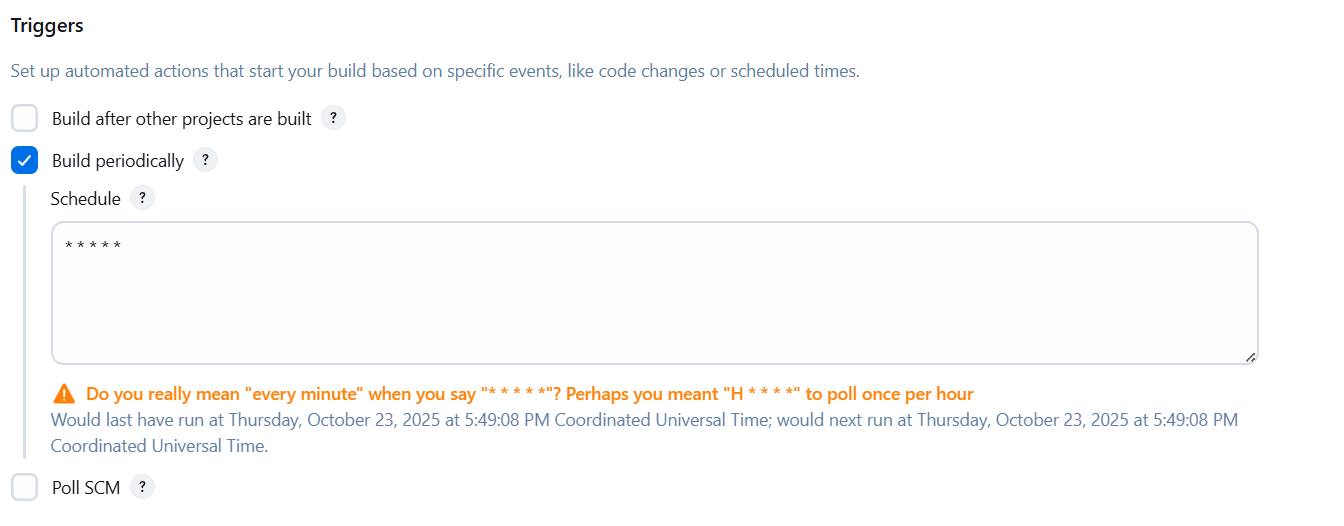


Then in source Code Management  
click on git

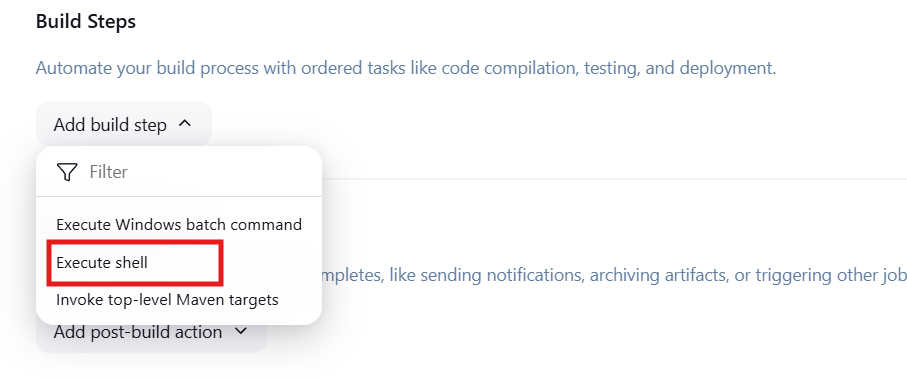
And give the repo url

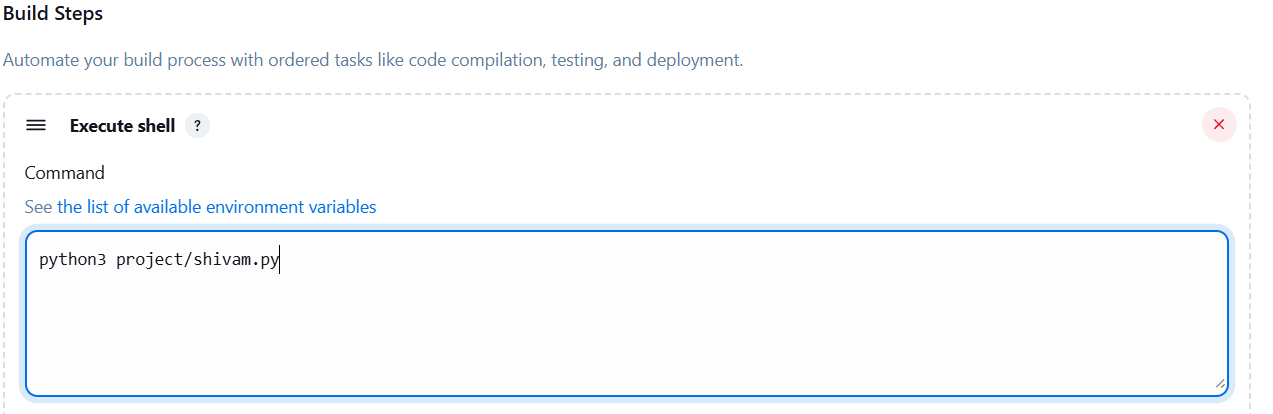
And change the branch to main from master



In Trigger i want after ever mi n it should run so i have configure like that   


And last Build Steps in that select the Execute shell(linux)

And the code which need to be given in the terminal



After this apply and save

Output

