**Jenkins Assignment**

1. How to change default port of Jenkins (8080) to some other port.

Default port for Jenkins [8080]

New Port for Jenkins [8081]

* vi /etc/default/Jenkins (Open Jenkins configuration file)
* Scroll down until you find the following lines:

# port for HTTP connector (default 8080; disable with -1)

HTTP\_PORT=8080

* HTTP\_PORT=8081 (Edit the second line to include the port number you want to specify)
* Esc, then: wq! (Save and exit the file)
* vi usr/lib/systemd/system/Jenkins.service (Open Jenkins Service)

Scroll down until you find the below line:

Environment="JENKINS\_PORT=8080"

Environment="JENKINS\_PORT=8081" (change the port with the new port number)

Esc, then: wq! (Save and exit the file)

* systemctl daemon-reload (Reload the Daemon)
* systemctl restart Jenkins (Restart Jenkins Service)
* Verify on the Webpage with Public IP: New port Number
* Now to make sure all the functionalities of Jenkins are working properly. Need make change to Jenkins URL to reflect the new port Number

Manage Jenkins > System > Jenkins Location > Jenkins URL

change from URL with old port Number to URL with new port number and hit save

systemctl restart Jenkins (Restart Jenkins Service)

1. How to change Home Directory of Jenkins.

Default home directory path [/var/lib/jenkins]

New Path we want to give is [/home/jenkins\_home/jenkins]

--------------------------------------------------------------------

* systemctl stop Jenkins (Stop Jenkins Service)
* mkdir /home/jenkins\_home (Create a new Jenkins Home directory using the mkdir command)
* chown jenkins:jenkins /home/jenkins\_home (Change permissions for the new Home directory)
* cp -prv /var/lib/jenkins /home/jenkins\_home (Copy the contents from the old Jenkins Home directory to the new one)
* usermod -d /home/jenkins\_home Jenkins (Assign Jenkins as the user for the new home directory)
* vi /etc/default/Jenkins (Open Jenkins configuration file)
* Scroll down until you reach the JENKINS\_HOME entry

Edit the line to include the path to the new home directory

In our Case

NAME=jenkins

JENKINS\_HOME=/var/lib/$NAME (old home directory path)

JENKINS\_HOME=/home/jenkins\_home/$NAME (new home directory path)

Esc, then: wq! (Save and exit the file)

* vi usr/lib/systemd/system/Jenkins.service (Open Jenkins Service)

Scroll down until you reach the Environment and Working Directory

Edit the line to include the path to the new home directory

In our Case

Environment="JENKINS\_HOME=/var/lib/jenkins" [OLD Entry]

WorkingDirectory=/var/lib/jenkins

Environment="JENKINS\_HOME=/home/jenkins\_home/jenkins" [NEW Entry]

WorkingDirectory=/home/jenkins\_home/jenkins

Esc, then: wq! (Save and exit the file)

* systemctl daemon-reload (Reload the Daemon)
* Rename the old Jenkins home directory

mv /var/lib/Jenkins /var/lib/Jenkins.old

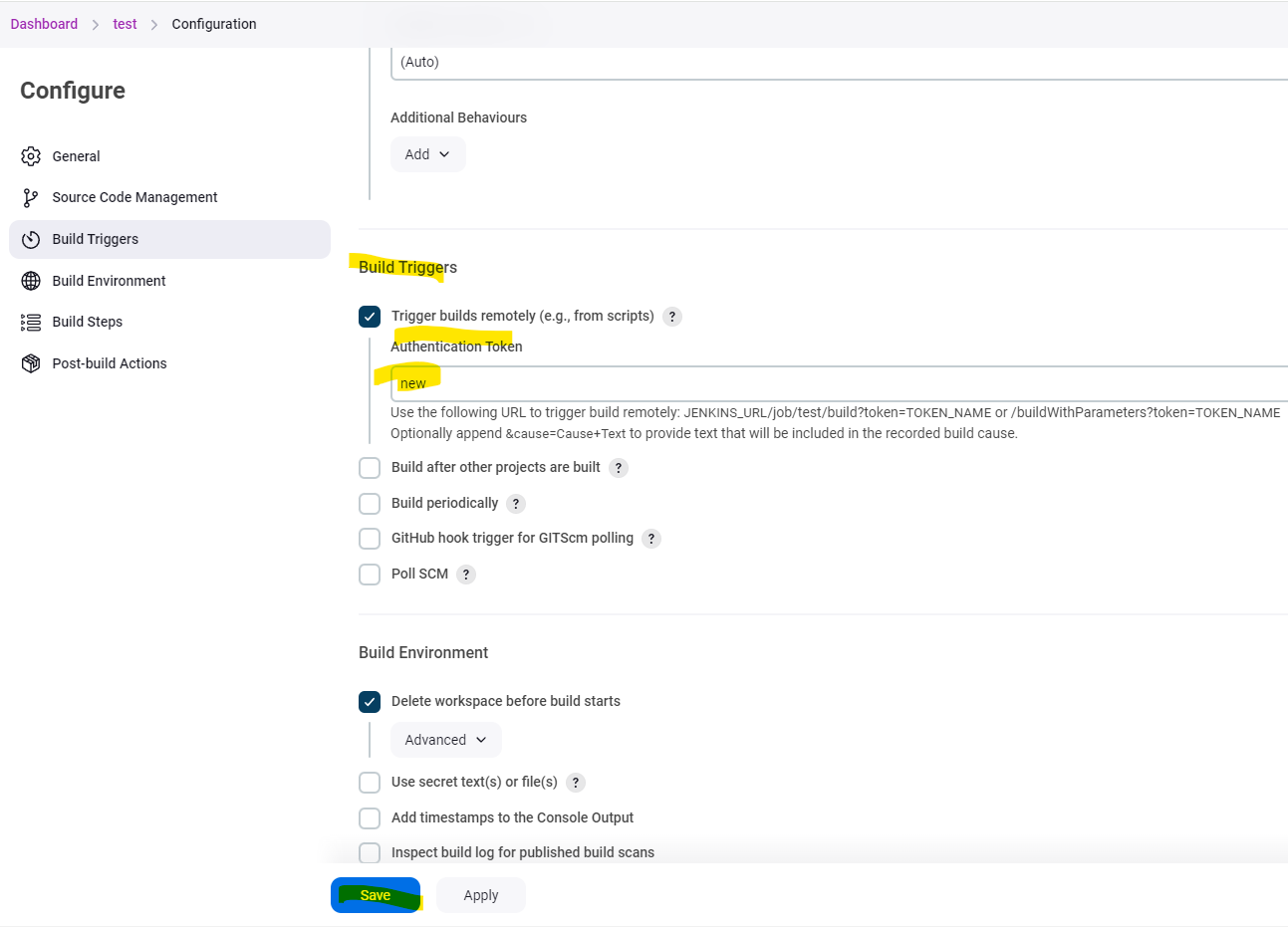
* systemctl start Jenkins (Start Jenkins Service)
* Verify the New Home directory under > Manage Jenkins > System > Home directory

1. How to run Remote Job in Jenkins.

We can trigger a job remotely through a Web URL in Jenkins also.

We will follow the below steps to achieve the same.

While configuring the job, we need to select Trigger builds remotely under Build Triggers, then we can give any authentication token eg new and Click Save.



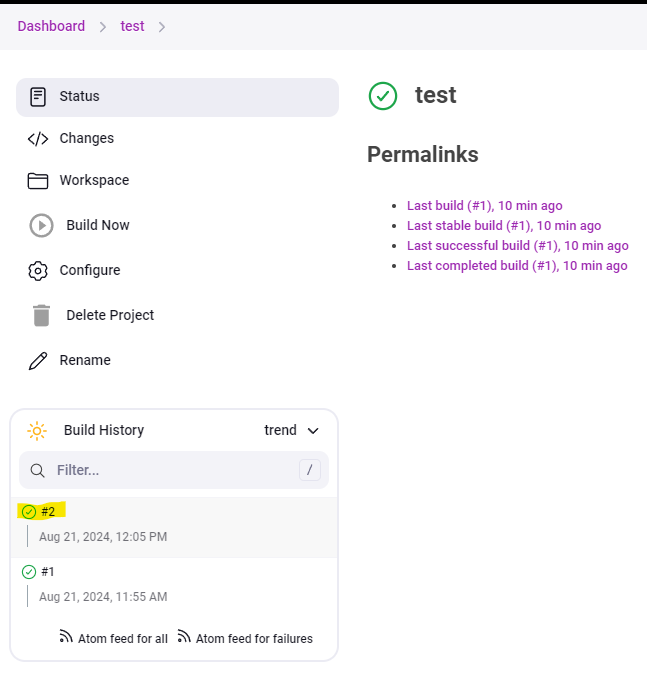
To trigger the build remotely we need use the below URL

JENKINS\_URL/job/test/build?token=TOKEN\_NAME or /buildWithParameters?token=TOKEN\_NAME

Below screenshot is in our case

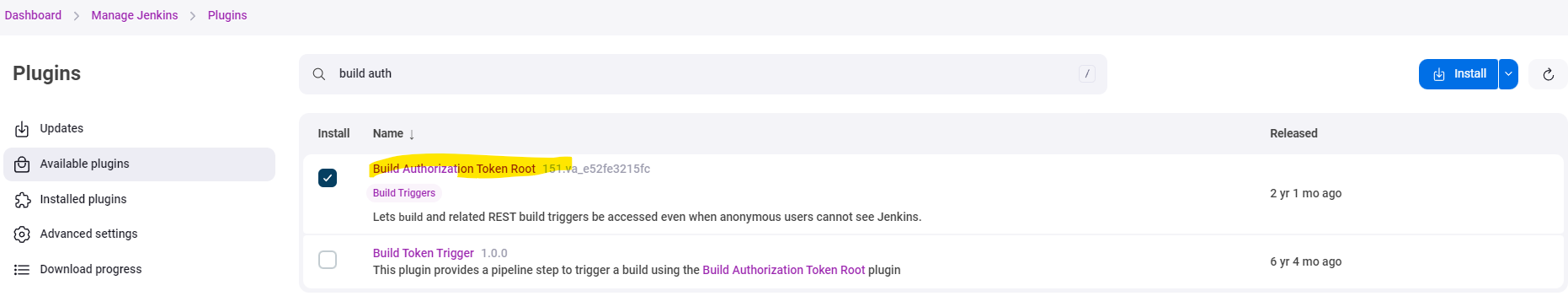


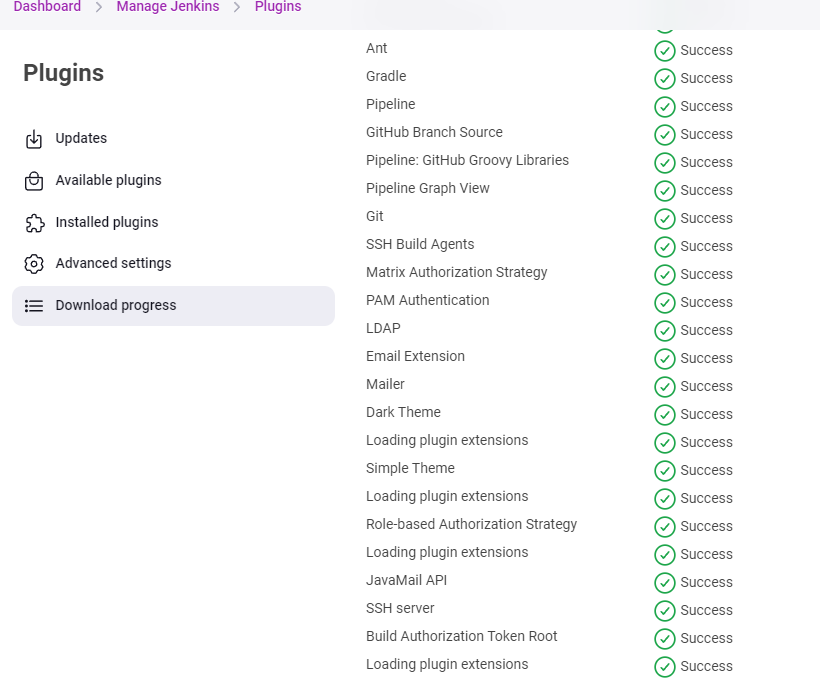
We see the 2nd Build is also executed successfully.



Alternate Way to run the job through remotely is by using a plugin

We need to install the plugin Build Authorization Token Root



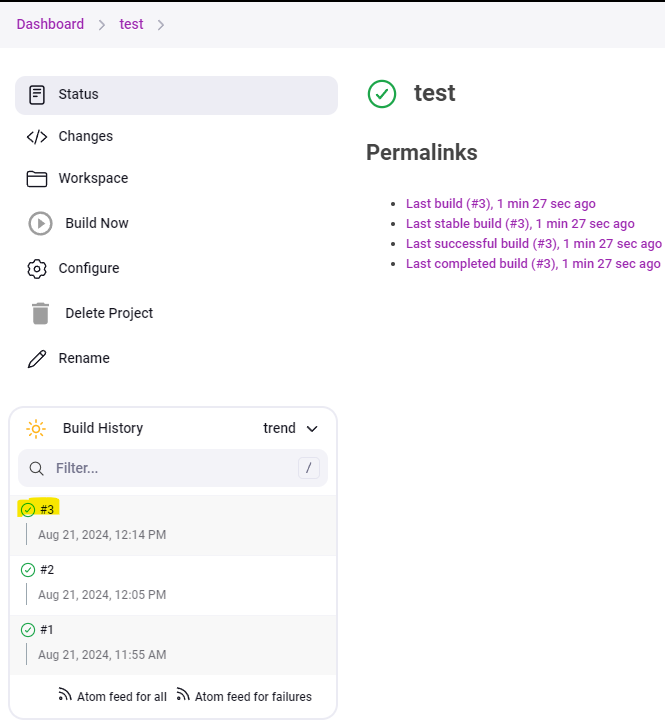


Now we will run the below URL to see if build runs.

<Jenkins\_URL>buildByToken/build?job=<Job\_Name>&token=<token Name>



See the Build 3 is also completed successfully.



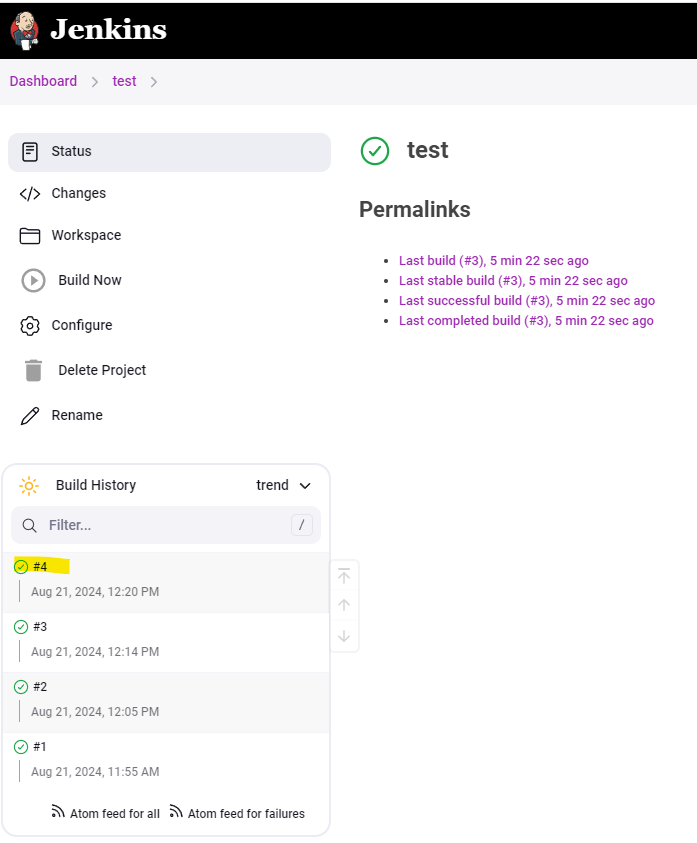
We can also run it on the Ubuntu terminal using the below command.

Just need to add “\” before the & and then run.

<Jenkins\_URL>buildByToken/build?job=<Job\_Name>\&token=<token Name>



Build 4 is completed successfully

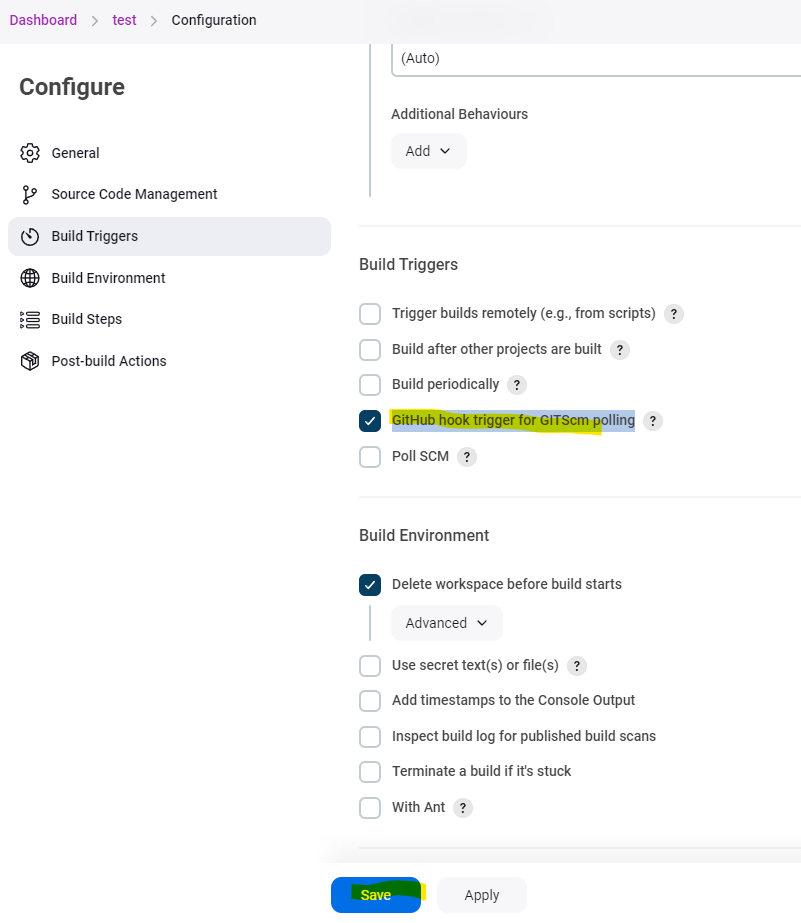


1. How to create GitHub hook trigger for GITScm polling through Jenkins.

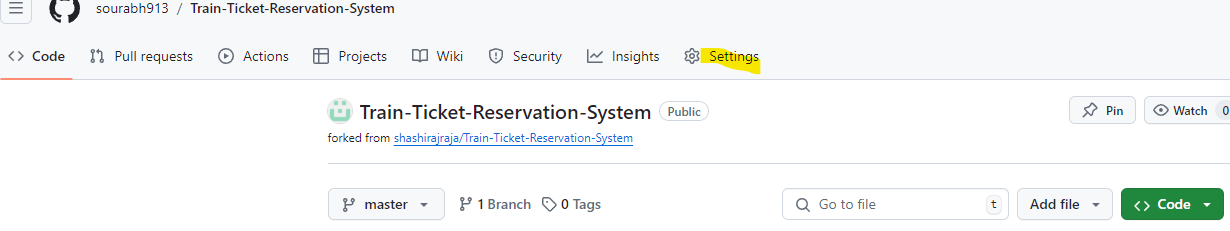
This is used when a job needs to be triggered when there is any change in the GitHub repo.

Steps to followed are below:

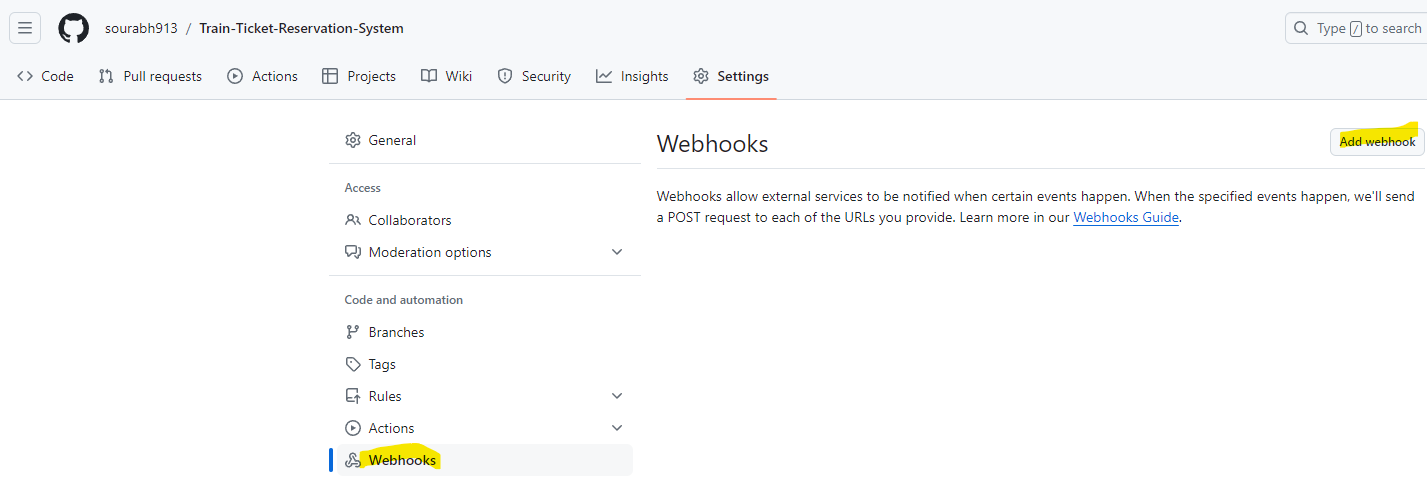
While configuring the job, Check the option of GitHub hook trigger for GITScm polling under Build Triggers and then click Save.



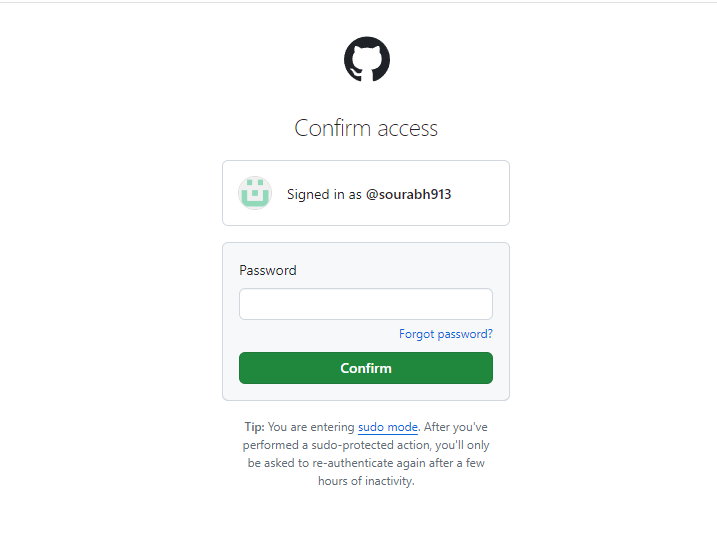
Then on the GitHub side, Go to Settings



Select Webhooks > Add webhook



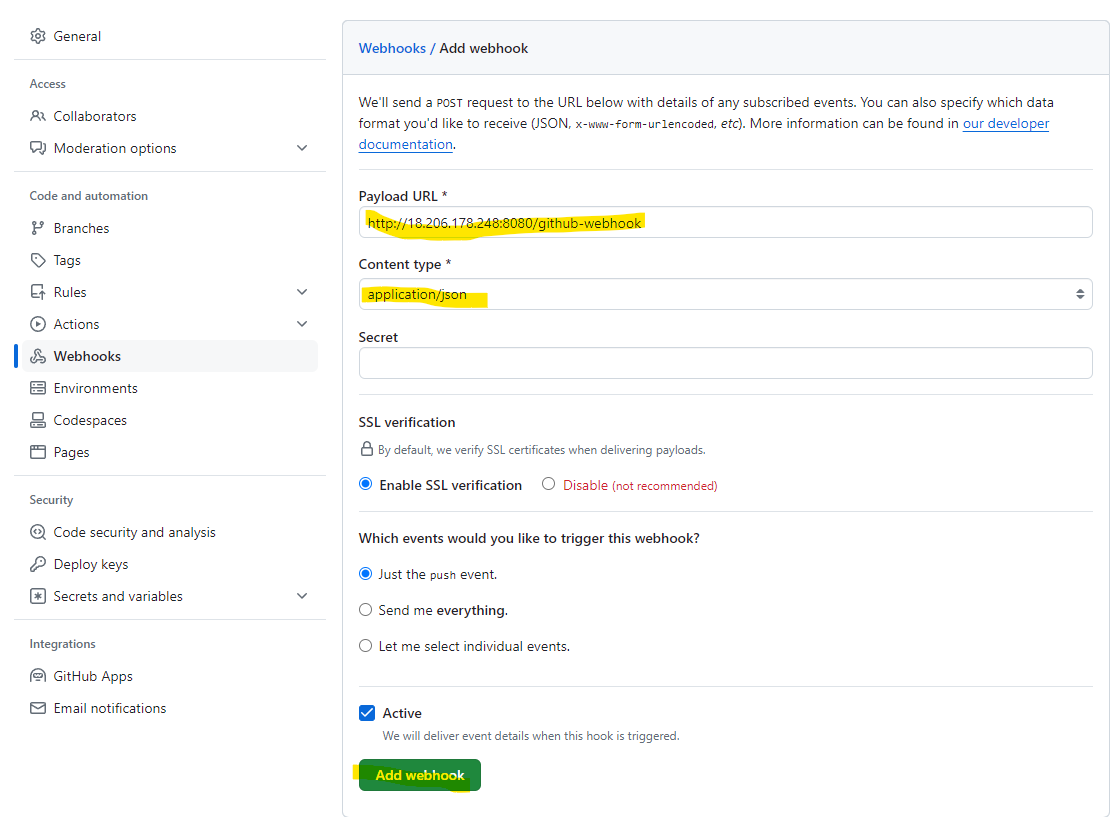
Need to put in password for your GitHub account.

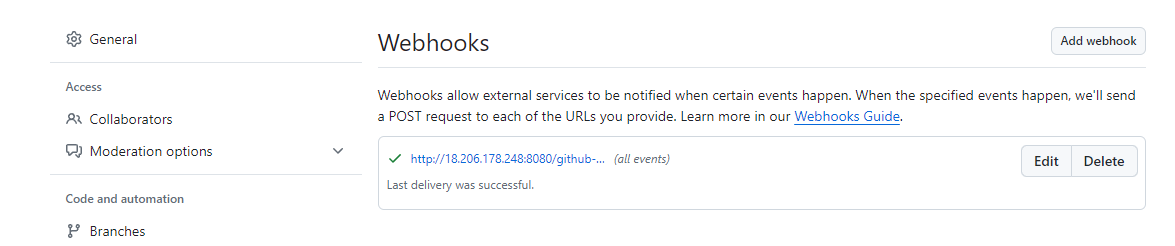


Then under the Payload URL, give the Jenkins URL /github webhook

And under Content type > Select application/json.

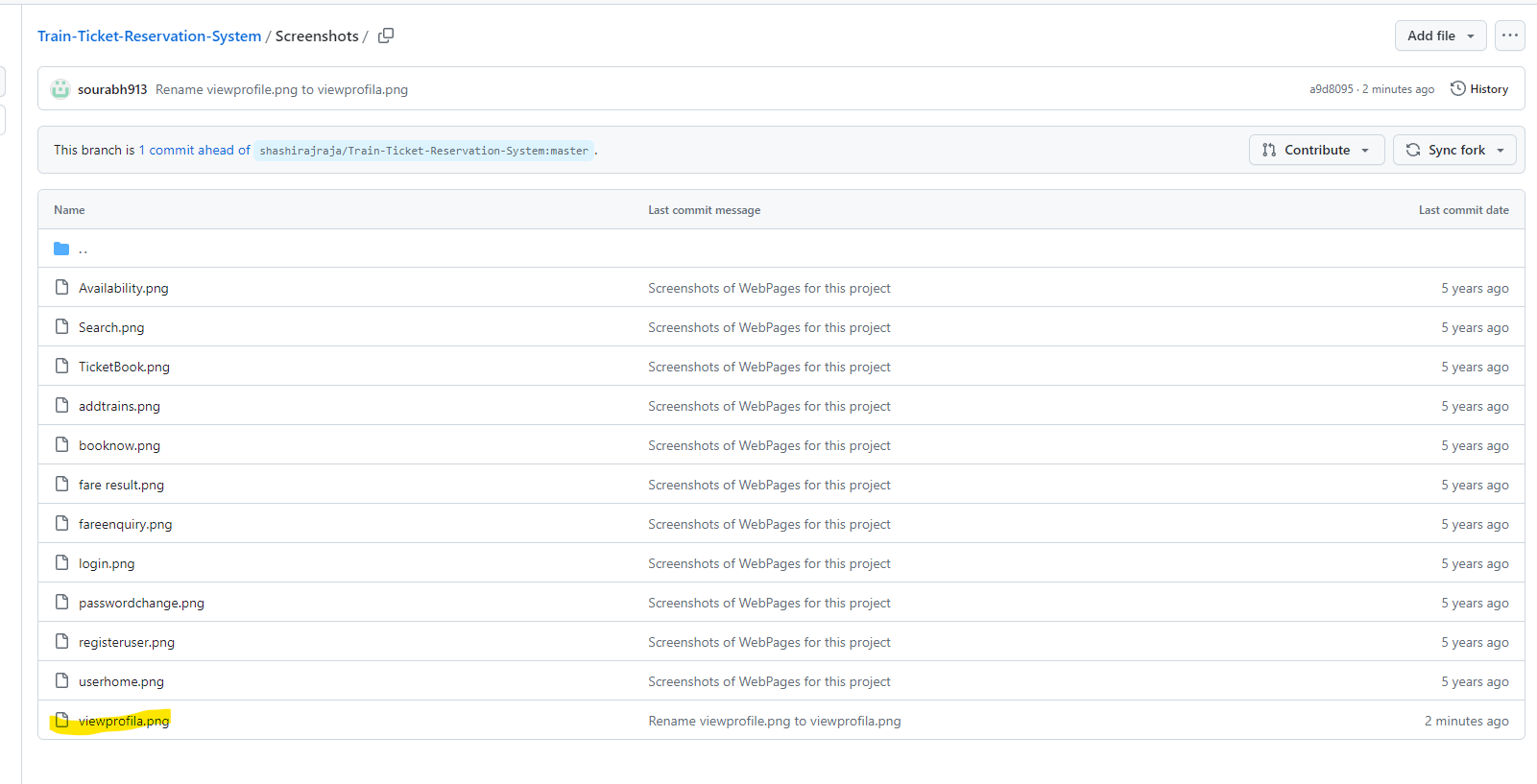
Click Add Webhook

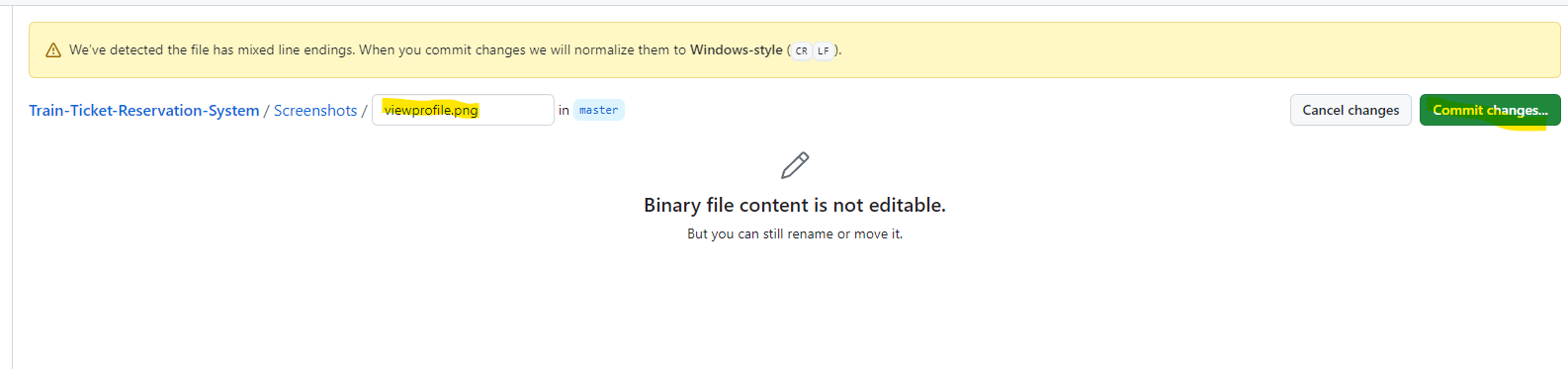


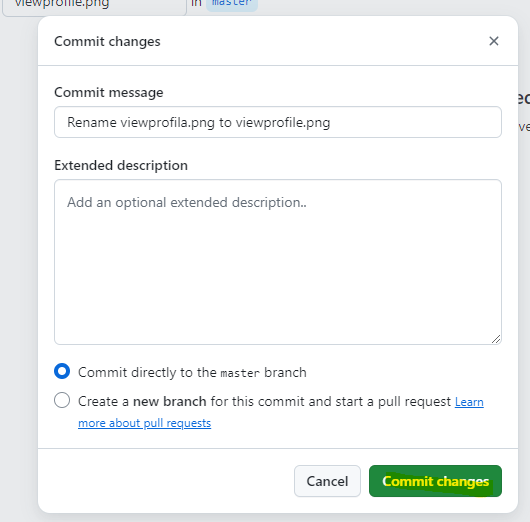


For Example, in my case I made a change under my repo > screenshots > rename viewprofila.png to viewprofile.png

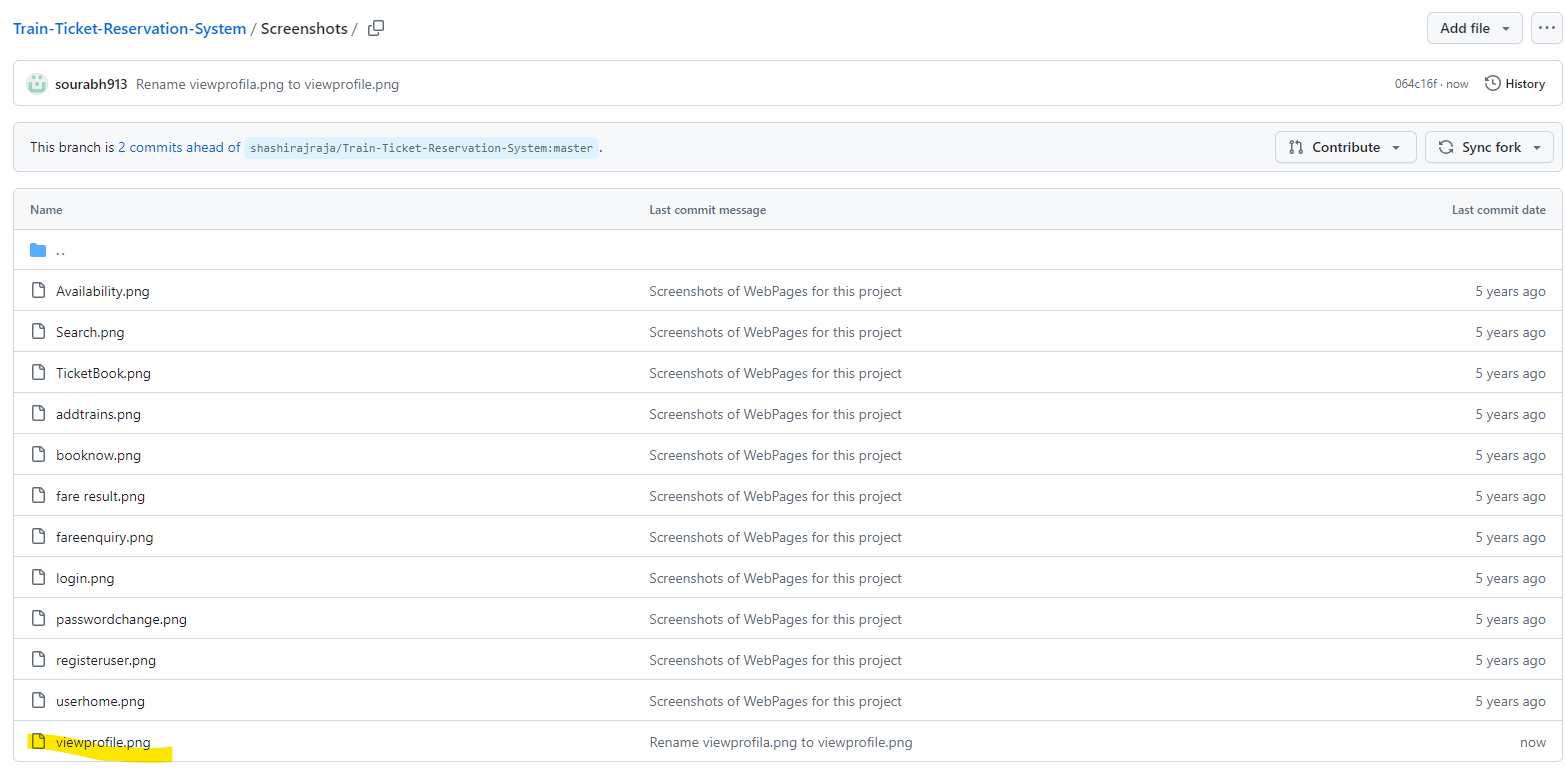
Before Change



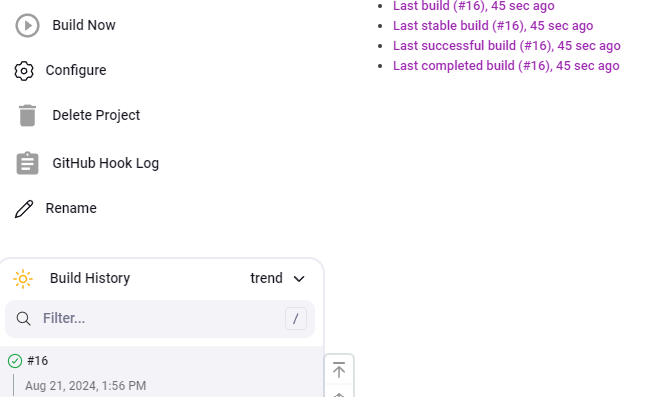




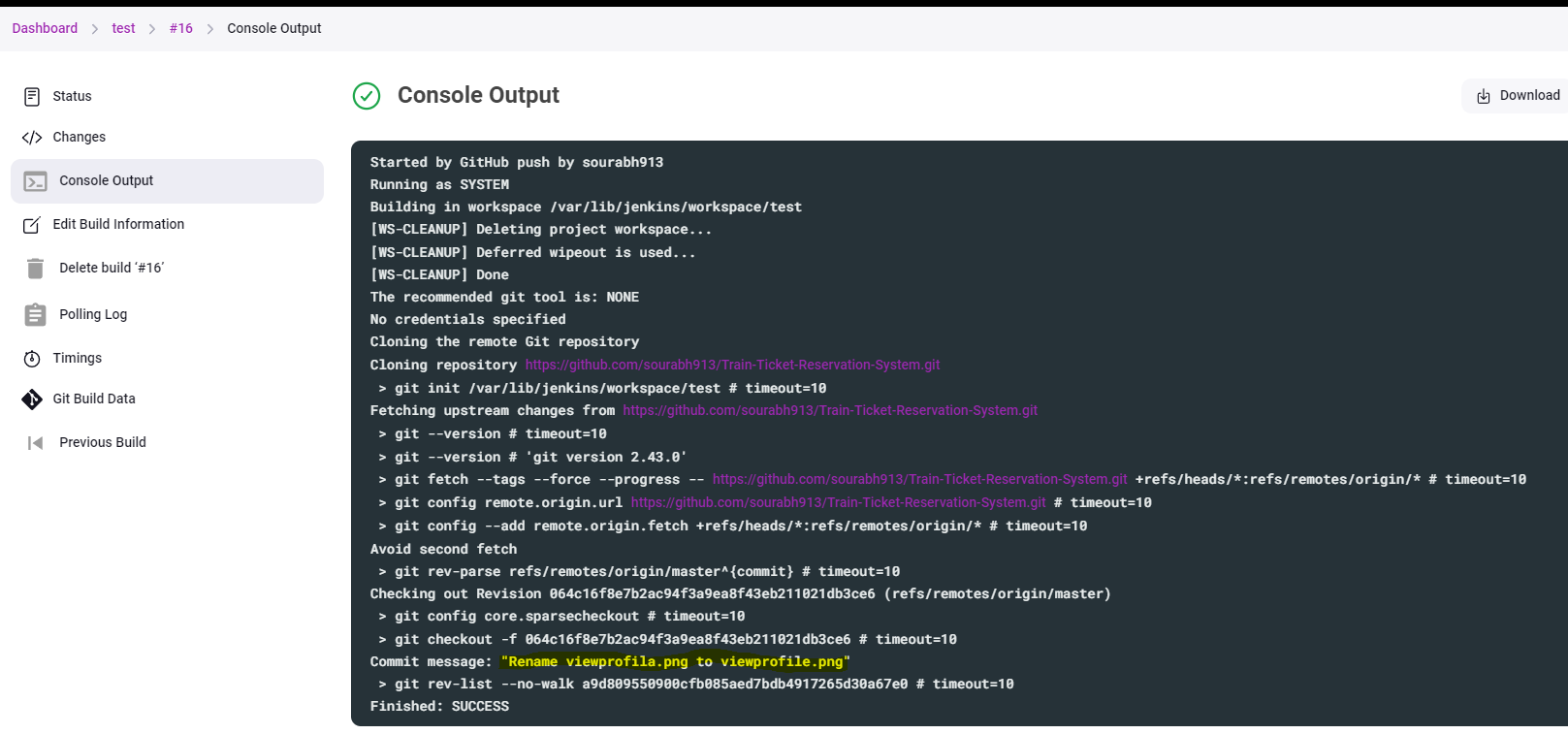
After Change



Immediately after the change, we see that build ran in Jenkins.



We can see the changes in the console output reflected successfully



1. How to take backup in Jenkins and Restore.