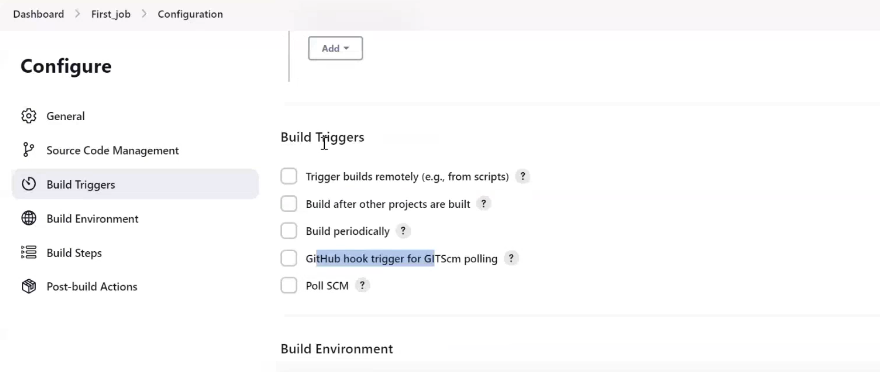
1. Configure 2 slave machines in Jenkins master.
2. Configure webhooks to Jenkins job.
3. Configure poll scm and build periodical options in Jenkins job.
4. Take backup of Jenkins server by using bash script.
5. Take backup of Jenkins using rethin backup plugin.
6. Setup a new Jenkins server and dump the backup taken in task4.

1Q: Configure 2 slave machines in Jenkins master.



Q: What is the use of build trigger?

A: With the help of BT we can decide when our job should trigger. Means this job automatically trigger without any manual intervention.

Webhooks: whenever there is changea in the source code, means whenever a new commit happends on our source code then automatically this job get trigger.

Webhooks are configured for automatic deployement, automatic CI/CD for development environment.

We can’t configure webhooks for production. Because we don’t want any change without our idea. Means every change we need to trigger, in production also, but should not be without our idea.

Whereas for development bcz there will be a lot of development activity happening for every few seconds or few minutes there will be some new code which will be push to our repository and they want to deploy and they want to test application. In this case

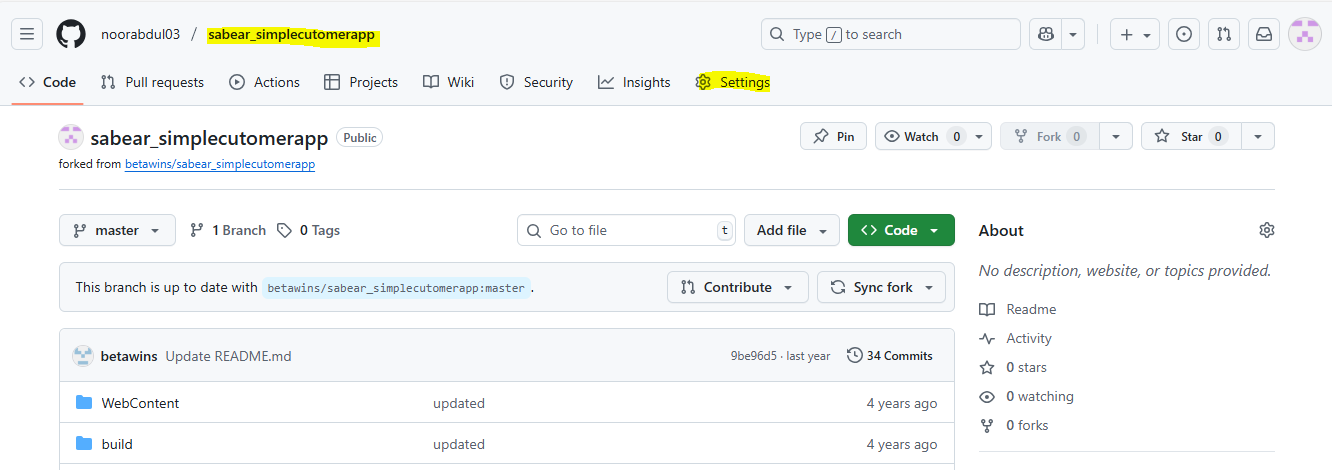
Ex: Develop has been new that he has completed the code, so we can’t login or raised the ticket as build trigger and we can’t build trigger bc he will be not having the access.

So we configure webhooks.

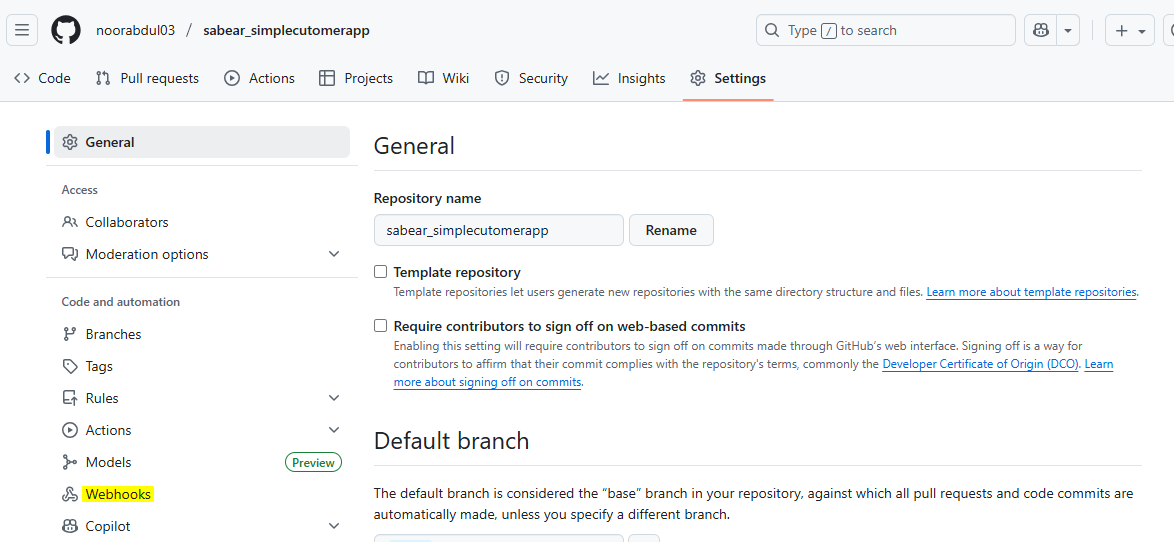
Q: How to configure webhooks?

Step1: Goto your github repository as below for sabear\_simplecutomerapp code I need to configure web hooks

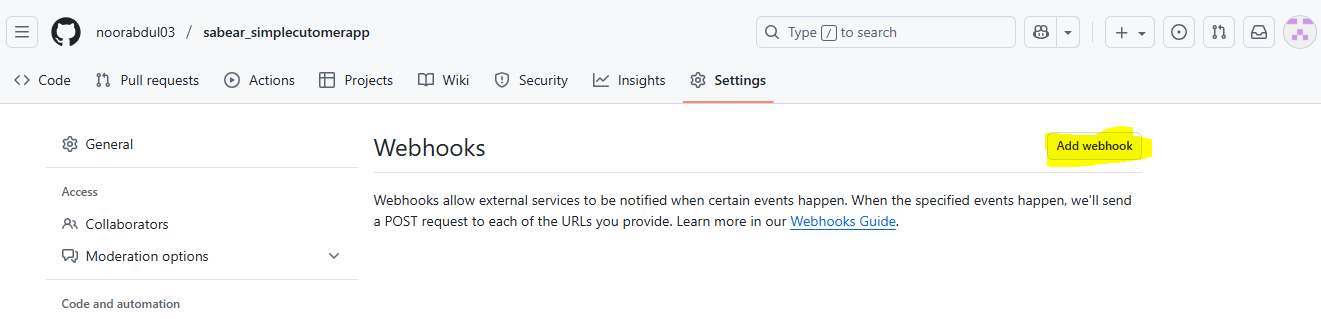
Go to settings



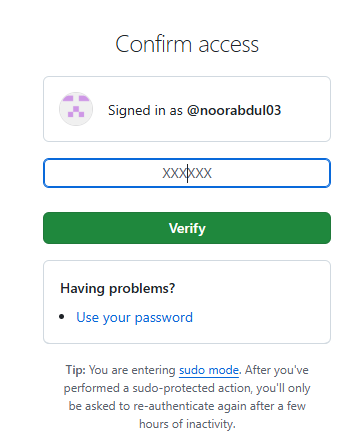
Click on webhooks



Click on Add webhook



Enter MFA



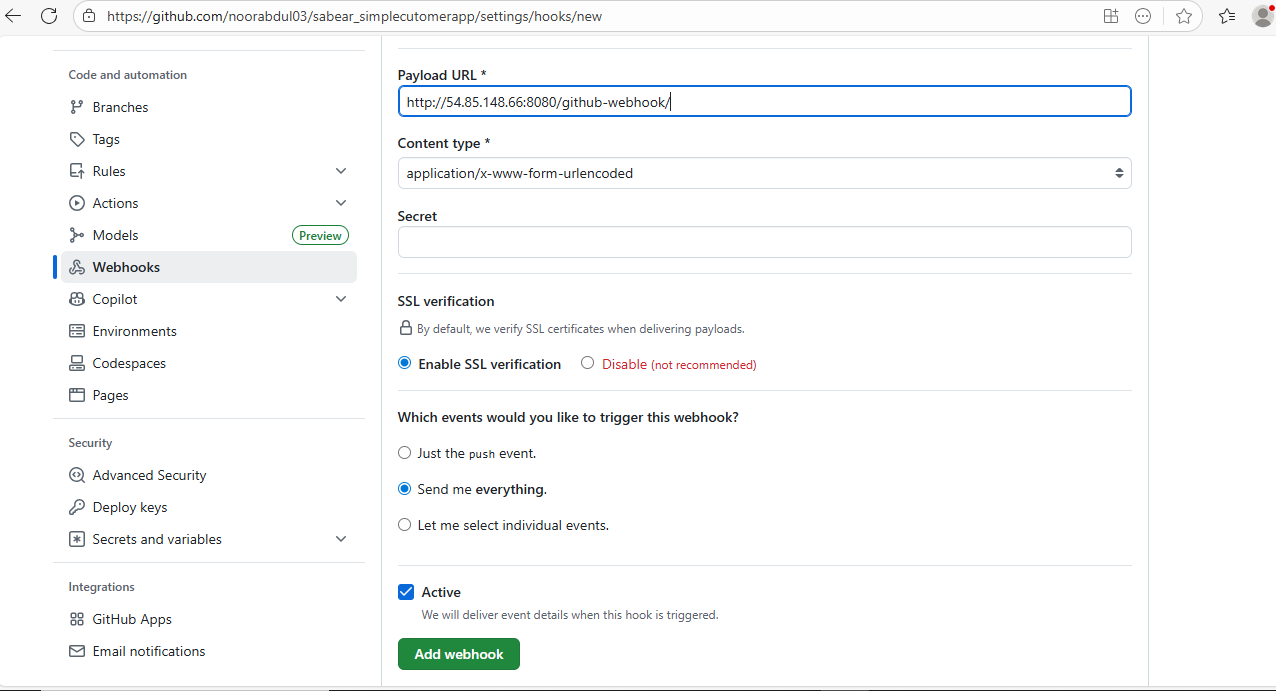
Step 2: Enter the payload URL

Q: What is payload URL

A: Means on which particular URL this needs to be configure.

So we have Jenkins URL 

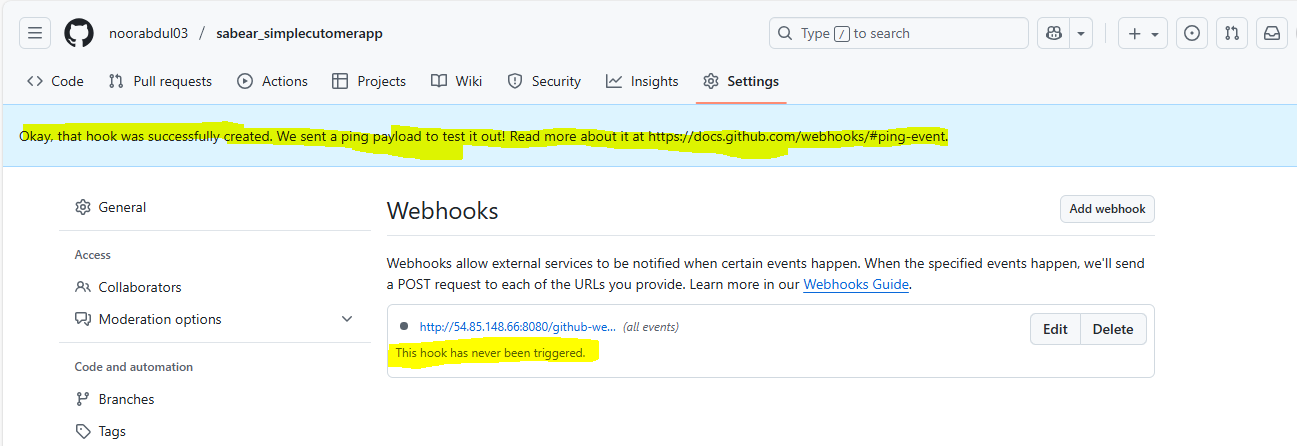
Configure this: copy URL and paste in payload URL



Here we can select which event needs to be trigger. Means anything which is been send to particular repository or any changes, so we can decide what need to be trigger.

Select send me everything: means whatever changes will implemented weather it is a PR weather it is a any commit then this payload URL should trigger our Jenkins jobs.

Click on Add webhook

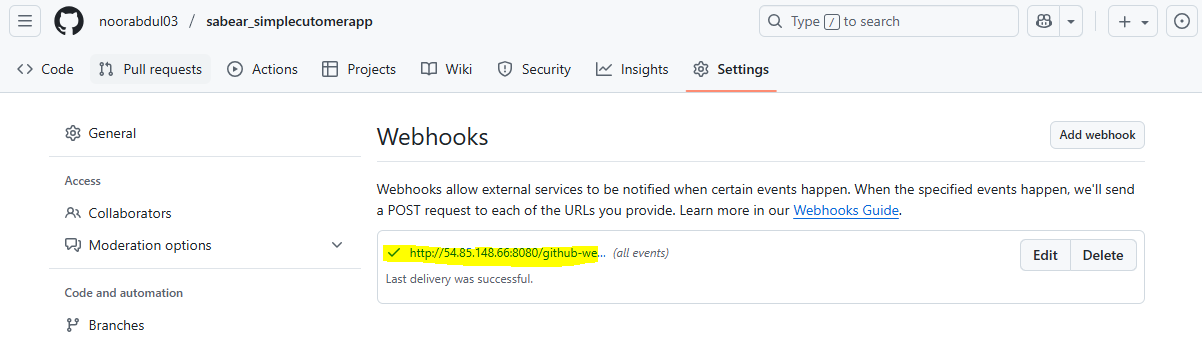


It is in indication means, not yet successful. So what it will do, it will sendout some text message to my JENKINS server. So once the message has been successfully delivered then we can see above as green one.

So once the test successfully validated then only in green colour.

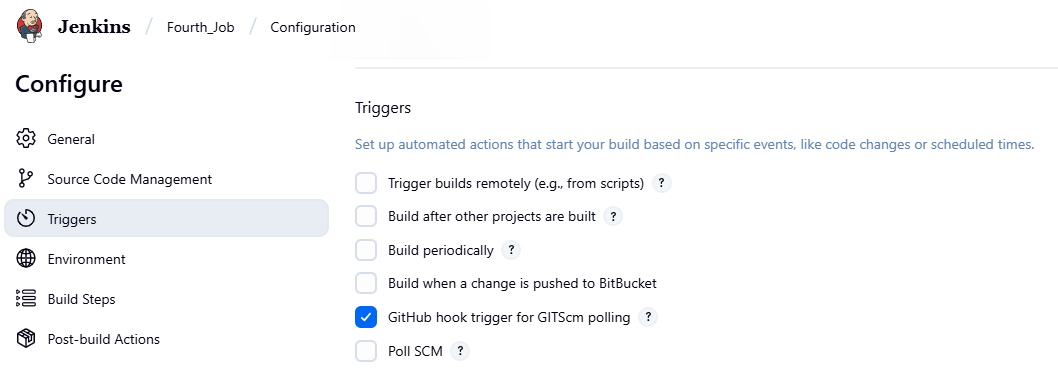
Refresh it only.

Now it was able to test our and successfully delivered the message



Step 3 : Once done, Go to Jenkins job (Fourth\_Job)

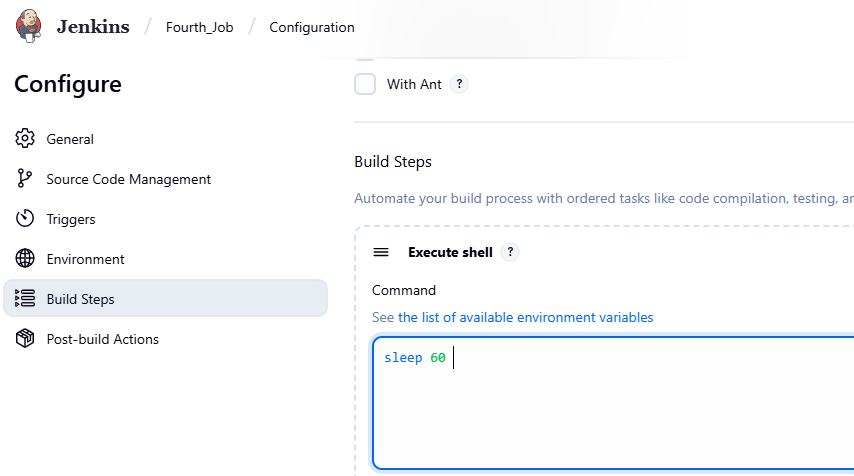
Just Enable Github hook trigger or GITScm polling



GITscm : scm: source code management

Once this job has been enable what will happen to our payload URL to our particular job we have synchronized, we have a connections Click on Save to this job.

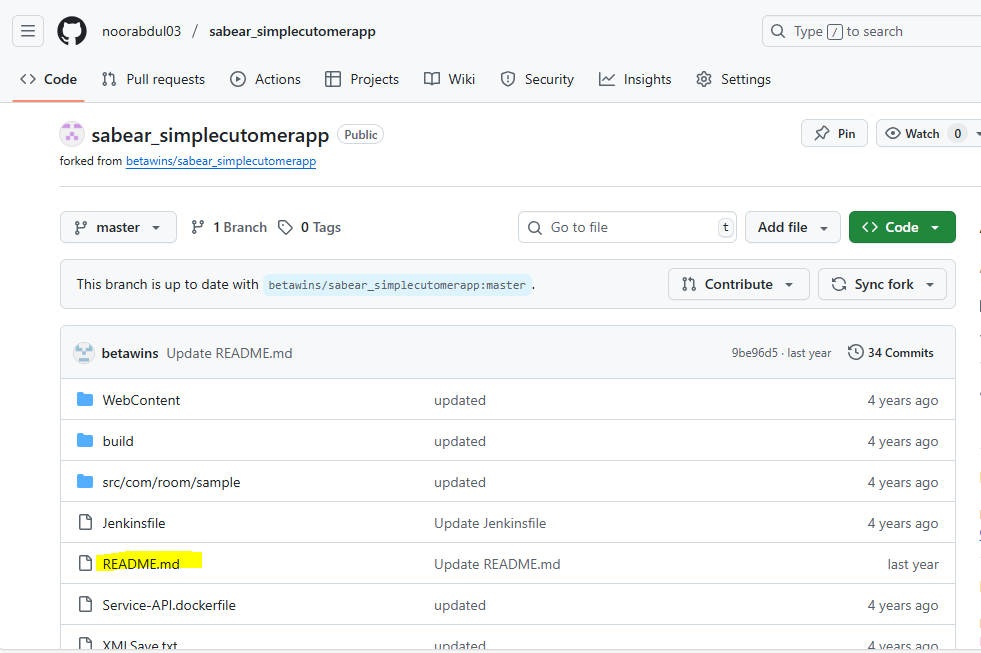
In this job, only for hold 2 seconds, I will add one basic sleep command. Sleep command hold my cursor for 60 seconds.



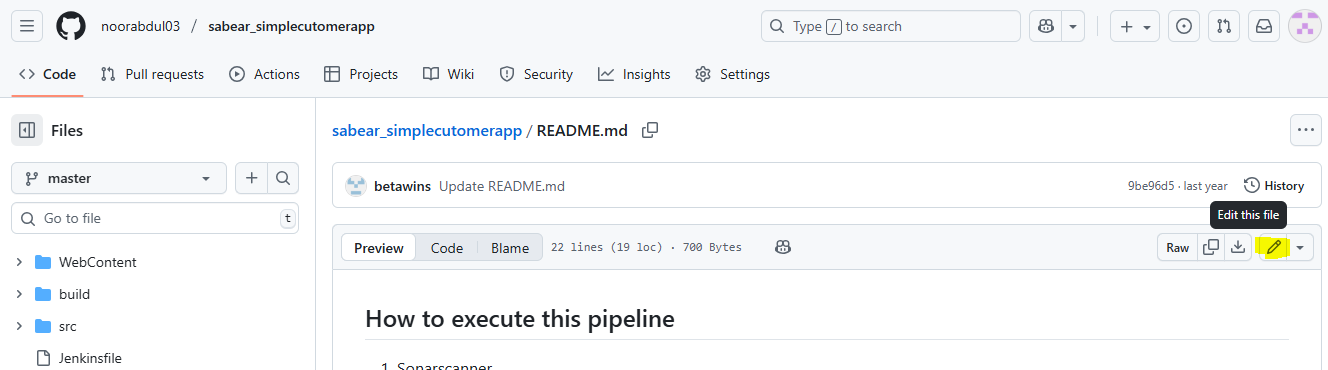
To view the output only

Click save

Step 4: Before deploying or executing job Go to GITHub to my code and I am trying to update my code after doing some changes.



Click on README.md



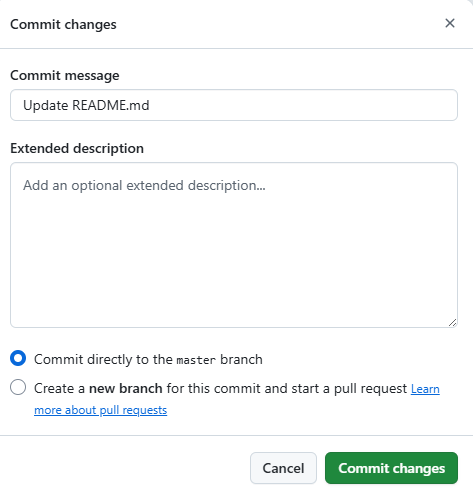
Click on Edit

And just or example (Add one . dot in this particular ffile and will update my commit)

Click on commit changes

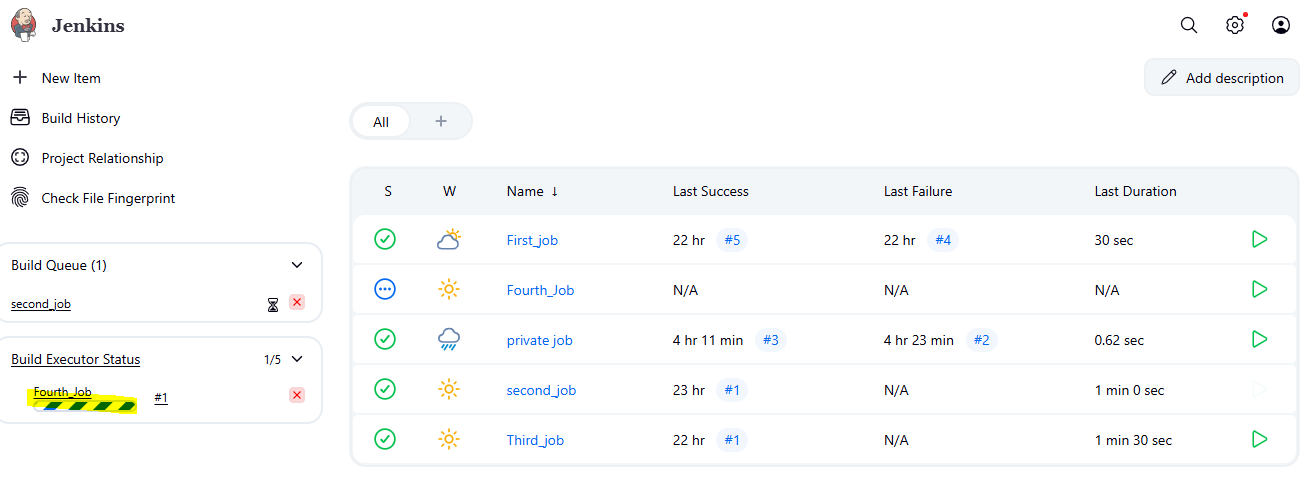


Click on commit changes again



So what happened, we have updated our commit. Means there is a new change in source code.

So what should happened or Jenkins should automatically trigger (see Fouth Job)



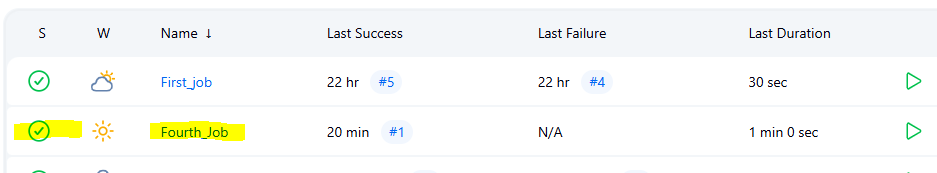
See this is automatically getting triggered without any manual intervention.

Webhooks Very important:

This is use for development environment.

So whenever developers are making any changes for that development environment we are going configure webhook. because everytime I will not be available to come to this job and doing click on BUILD NOW .

See the job has successfully completed.

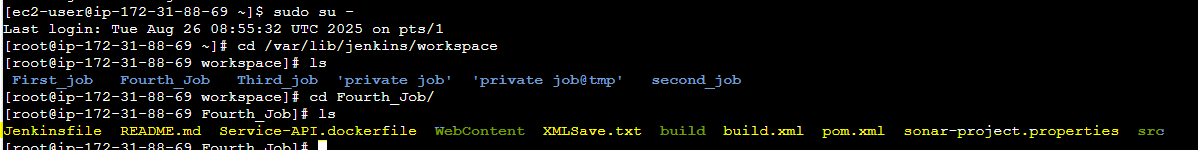


Interview question: What is webhooks?

Q: Where is my source code downloaded?

In my Jenkins server Ec2

Cd /var/lib/Jenkins/workspace/

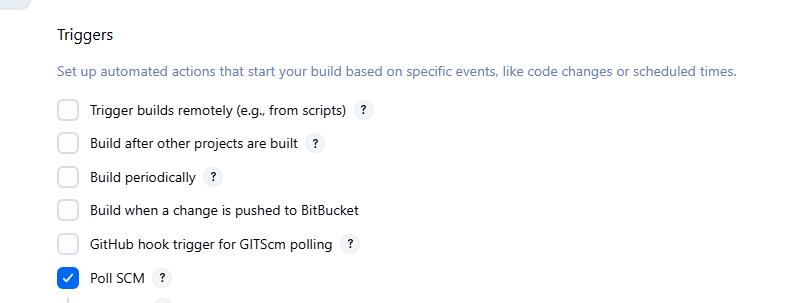


**All our souce code downloaded here. Means whatever we are trying to run on the Jenkins server we can find all the details in our workspace only.**

**Now**

**Poll SCM**

3) Configure poll scm and build periodical options in Jenkins job.



If I click on poll SCM

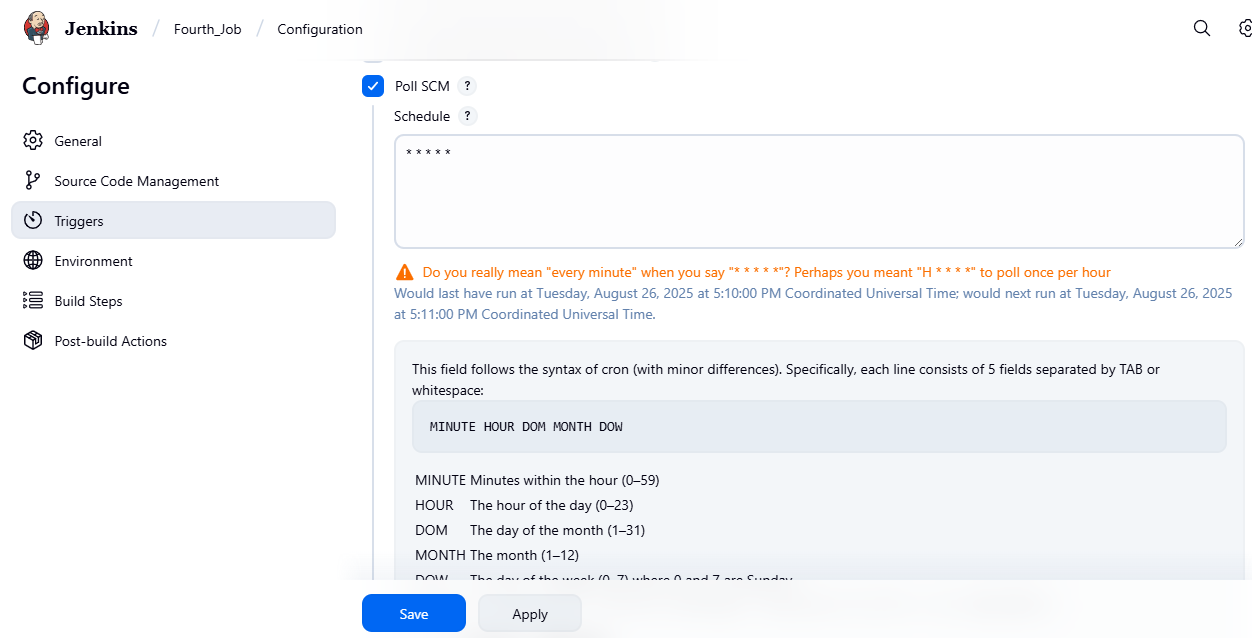
And I click on schchedule ? option Below I can see some information

This like a cron jobs, we can schedule at this particular time this job should get trigger

5 stars \*\*\*\*\* means every minute every hour every day every month every week

This job should get triggered

Click on save



Click on save

But I you see it wont show any changes in our job.

Now remove poll SCM and

select Build periodically and And I click on schchedule ? option Below I can see some information

This like a cron jobs, we can schedule at this particular time this job should get trigger

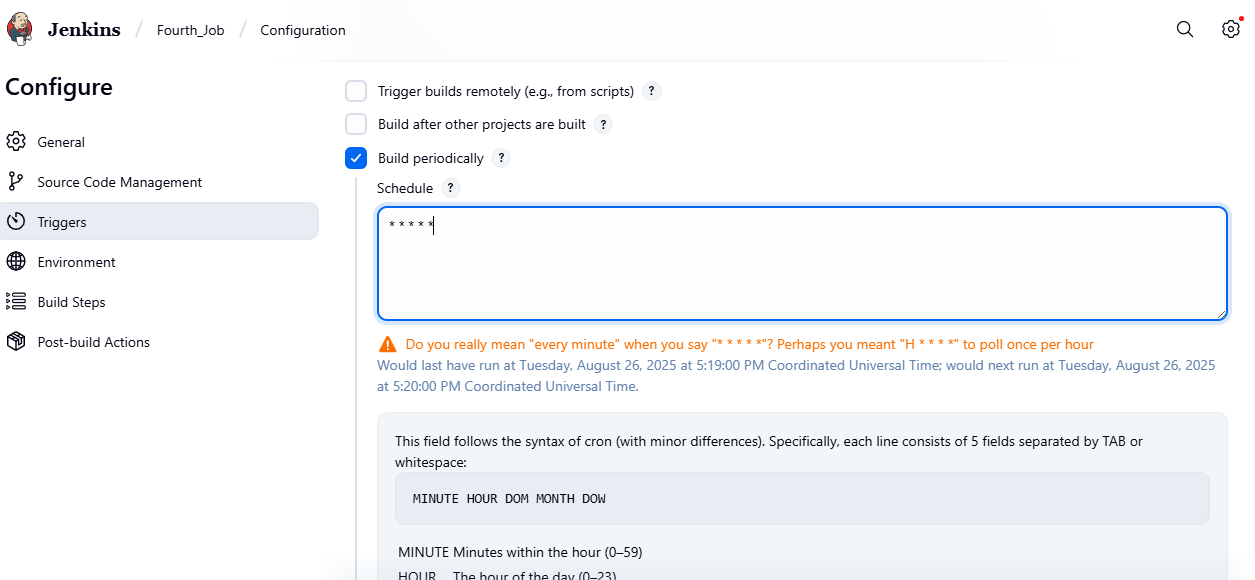
5 stars \*\*\*\*\* means every minute every hour every day every month every week

This job should get triggered

Click on save

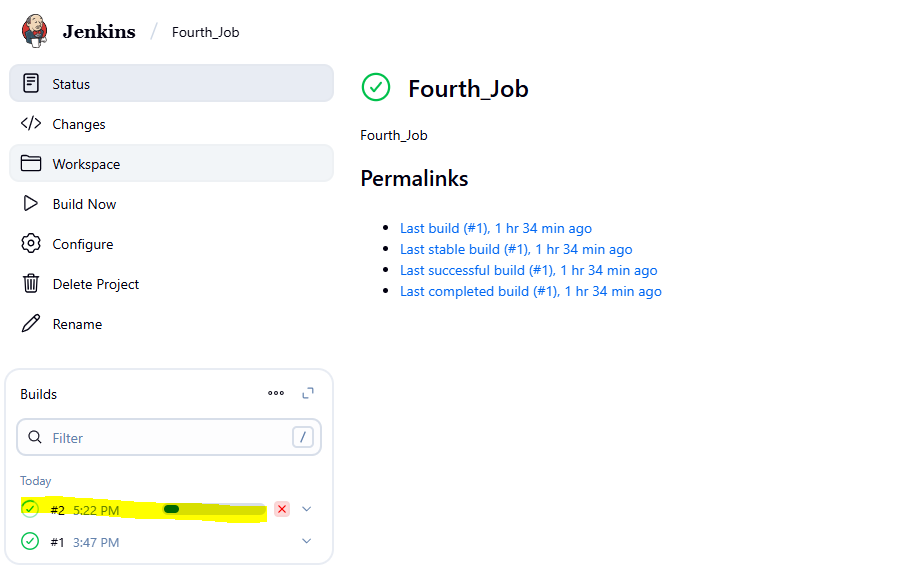
Check what happen? (Bcz we are having same fetaures in Build periodically and poll scm)

Now let us wait or 1 minute



Click save

See whats happen



See this job has triggered automatically If we keep Build periodicall and cron job.

**Q: Why the job didn’t triggered when Keep Poll scm**

**And what is the difference between build perical and poll scm**

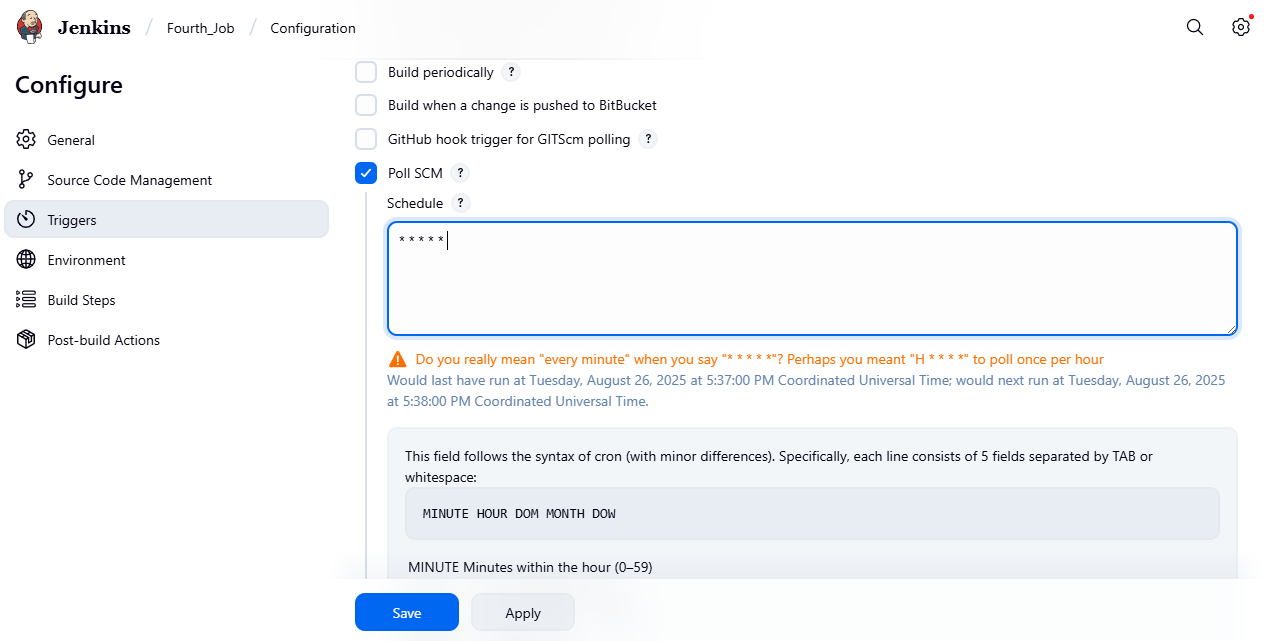
When I am keeping poll scm why it is not triggering bca it is going github repository and checking If there is any change or not.

If there is any change then only poll scm will trigger my job. If there is no change then only poll scm will never get trigger

\*\*\*where as build periodically what is doing?

Whether there is change or no change in code, it should trigger based upon the timing

**Task 3 : Now go to job and select poll scm**

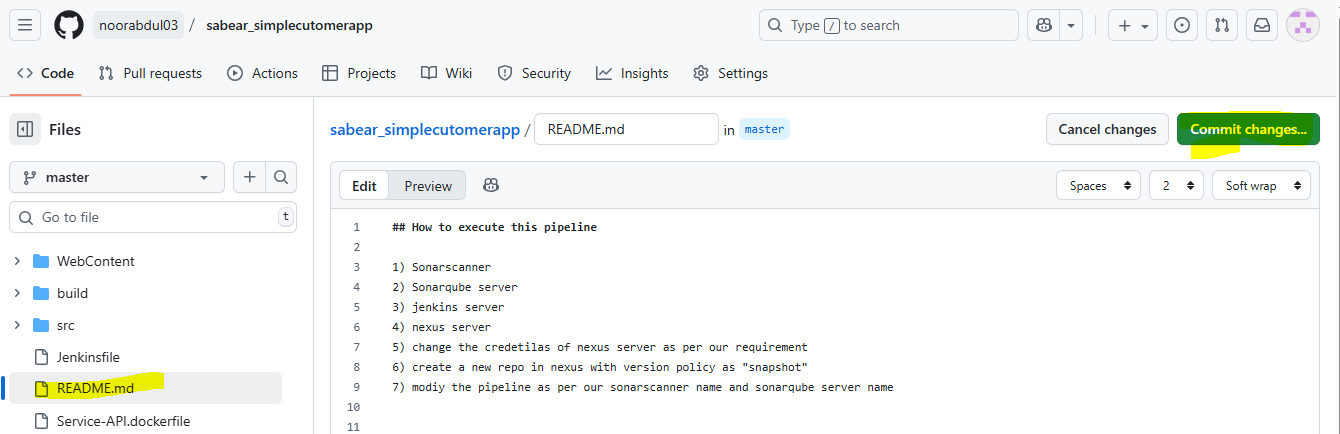


Click save

Step 2 : Just make some changes in the code in github repository

Go to red.md file and remove the dot at last or example

And do commit changes



See the below changes, the job is executing I any changes done in the code. If there is no change in the code then never triggered the job in poll scm.

