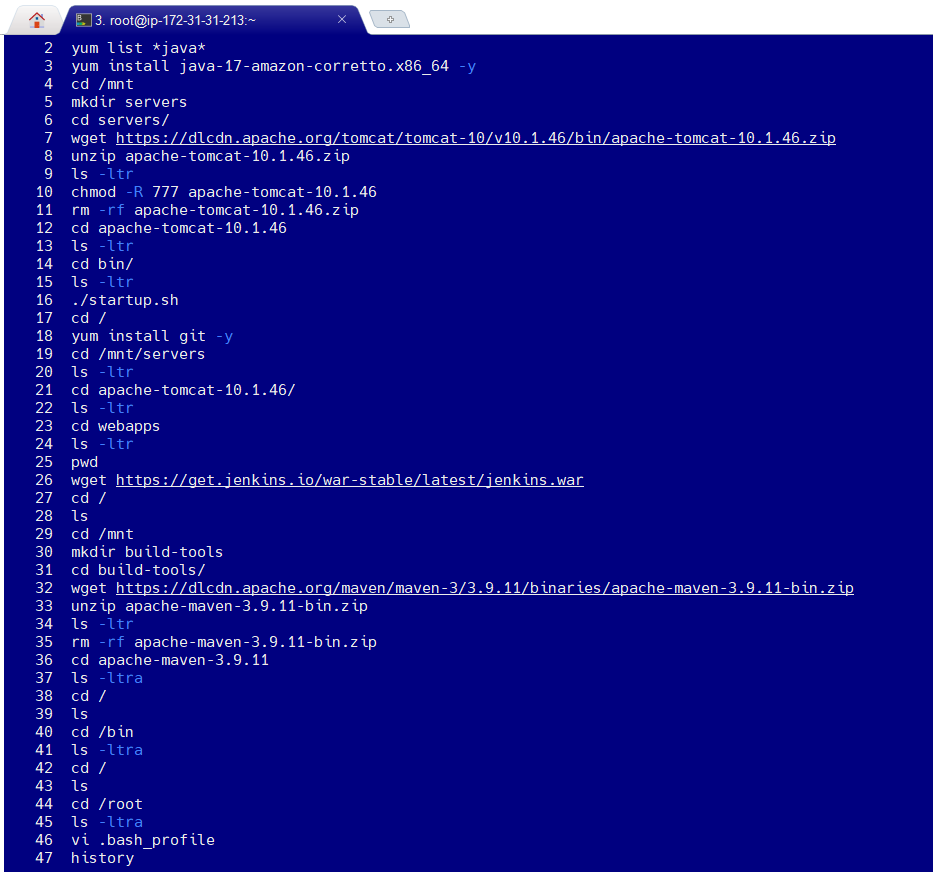
Assignment:

Step 1: Launch an EC2 instance. Setup Jenkins master on this EC2 instance i.e. install java, apache tomcat, Jenkins, maven, git.



Step 2: Launch another EC2 instance. Install and start httpd.

Step 3: In git make 3 repositories. Repository1, Repository2 and Repository3. The repositories are public.

If the required repositories are already present, clone them in Jenkins Master EC2 instance.

git clone Repository\_URL

Step 4: Login into Jenkins.

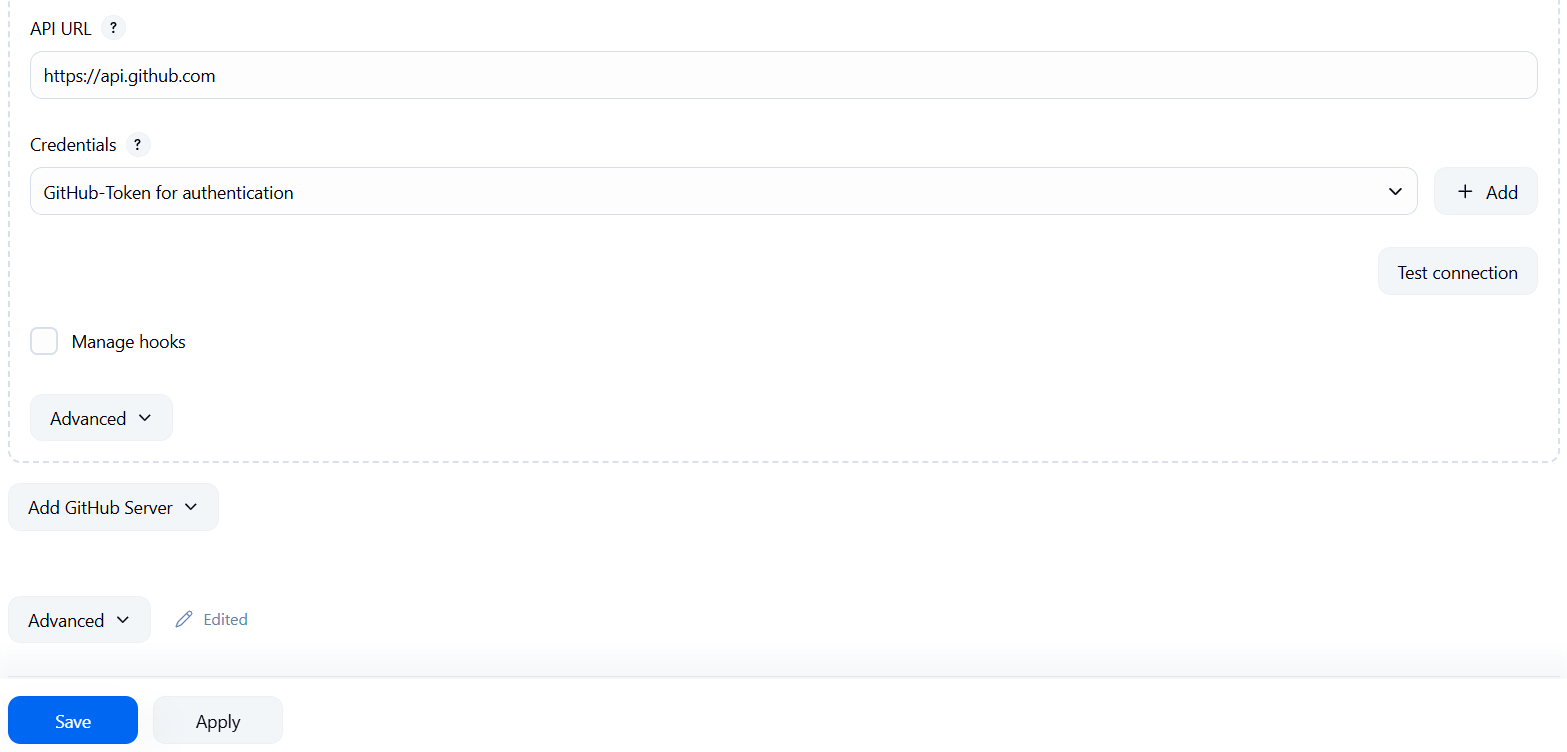
Go to Manage Jenkins → GitHub → Add GitHub Server → give name → Credentials, click on Add



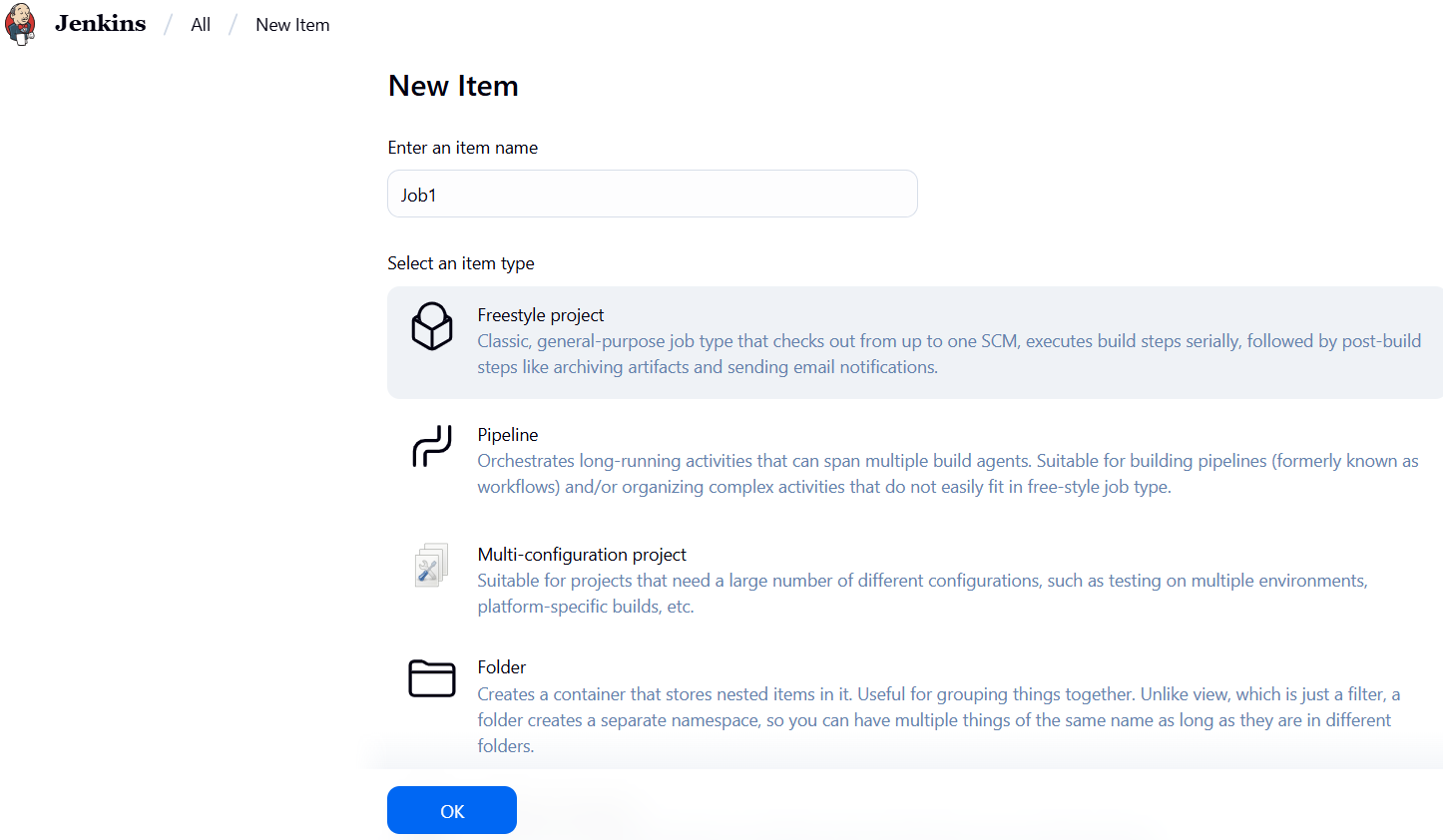
In Kind, select secret key → in Secret, paste the GitHub token → give ID → give Description → Add



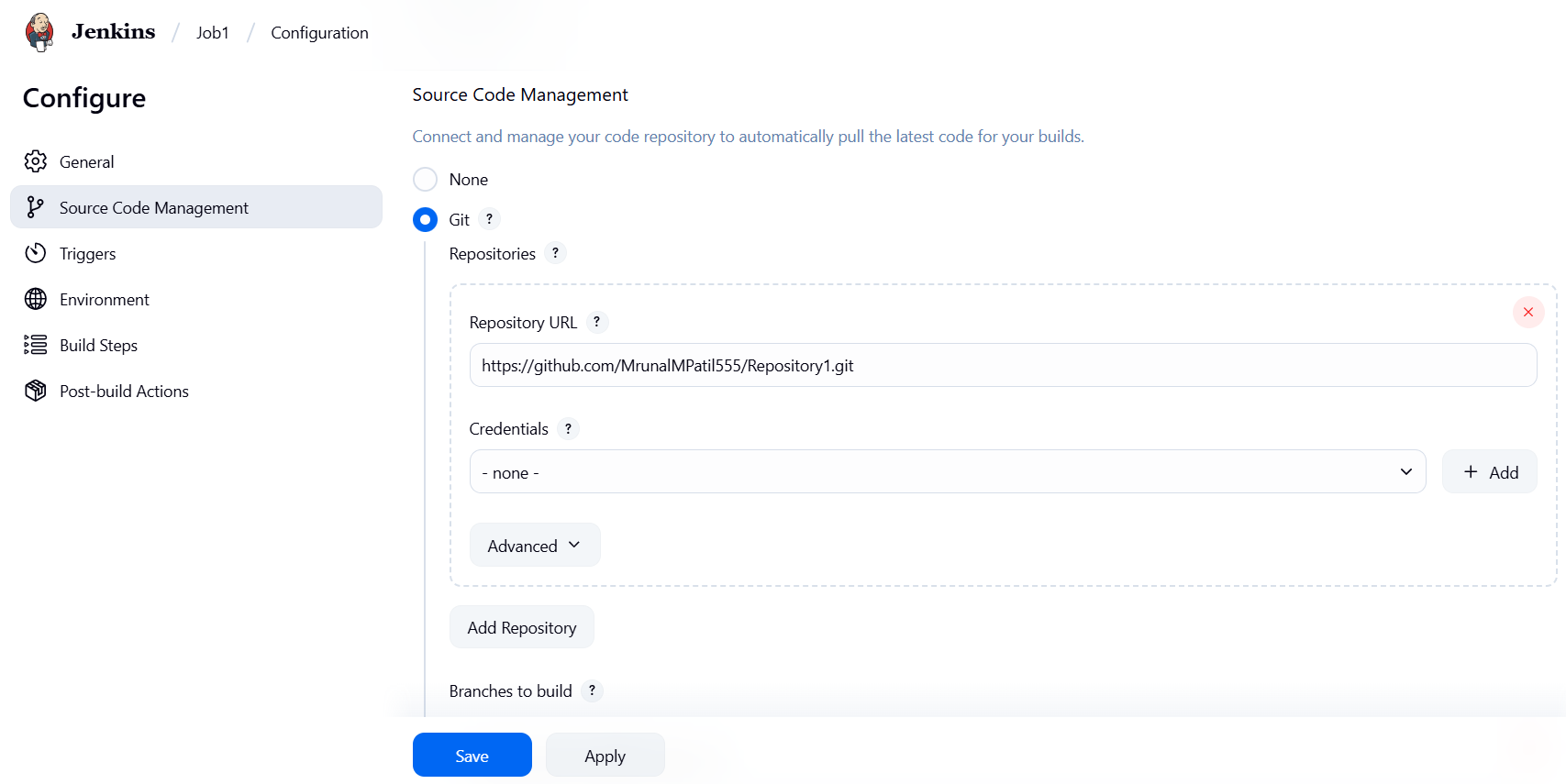
Now, select the credential you just created → Apply → Save



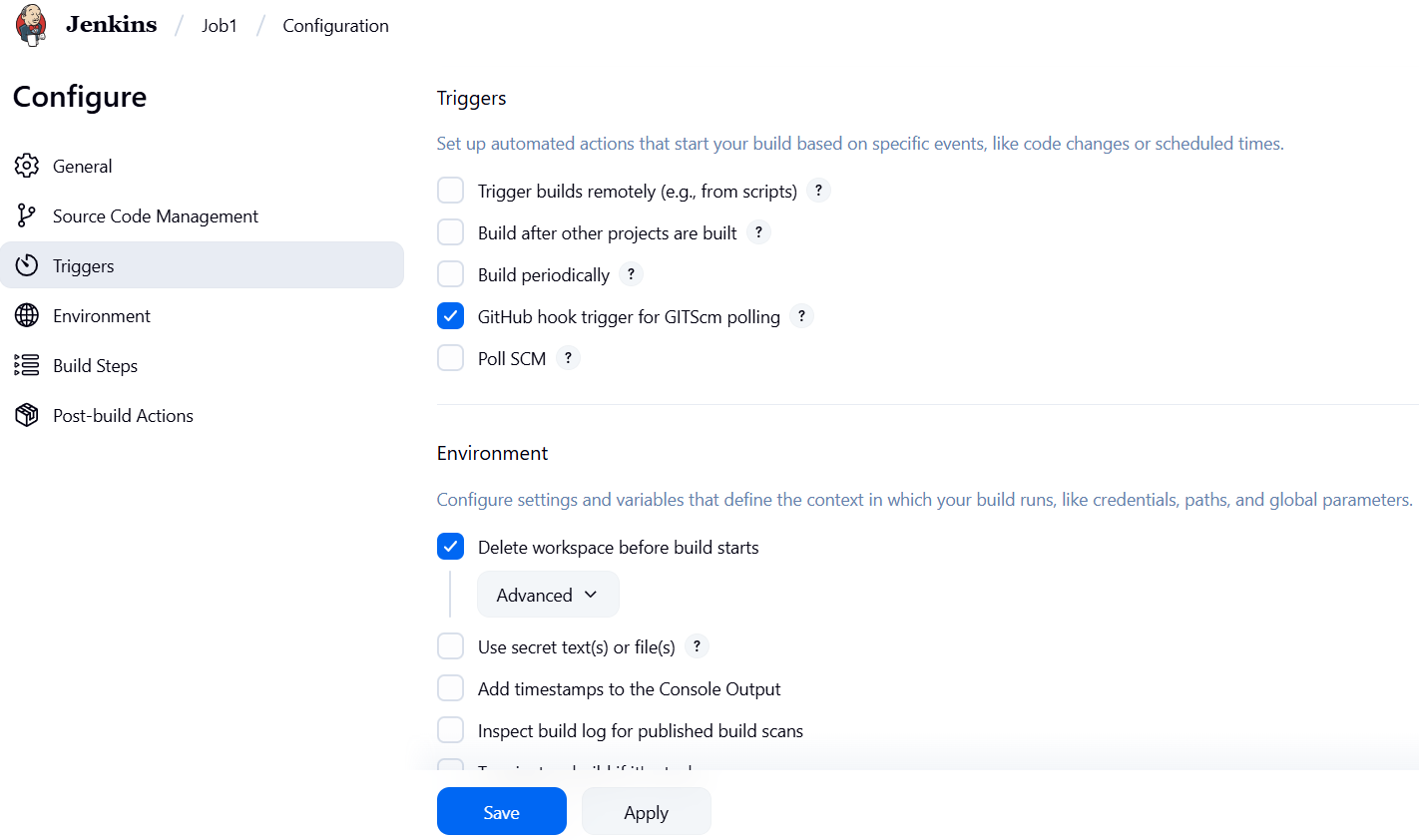
Step 5: Create a job.



In Source Code Management, select Git → paste the git repository URL



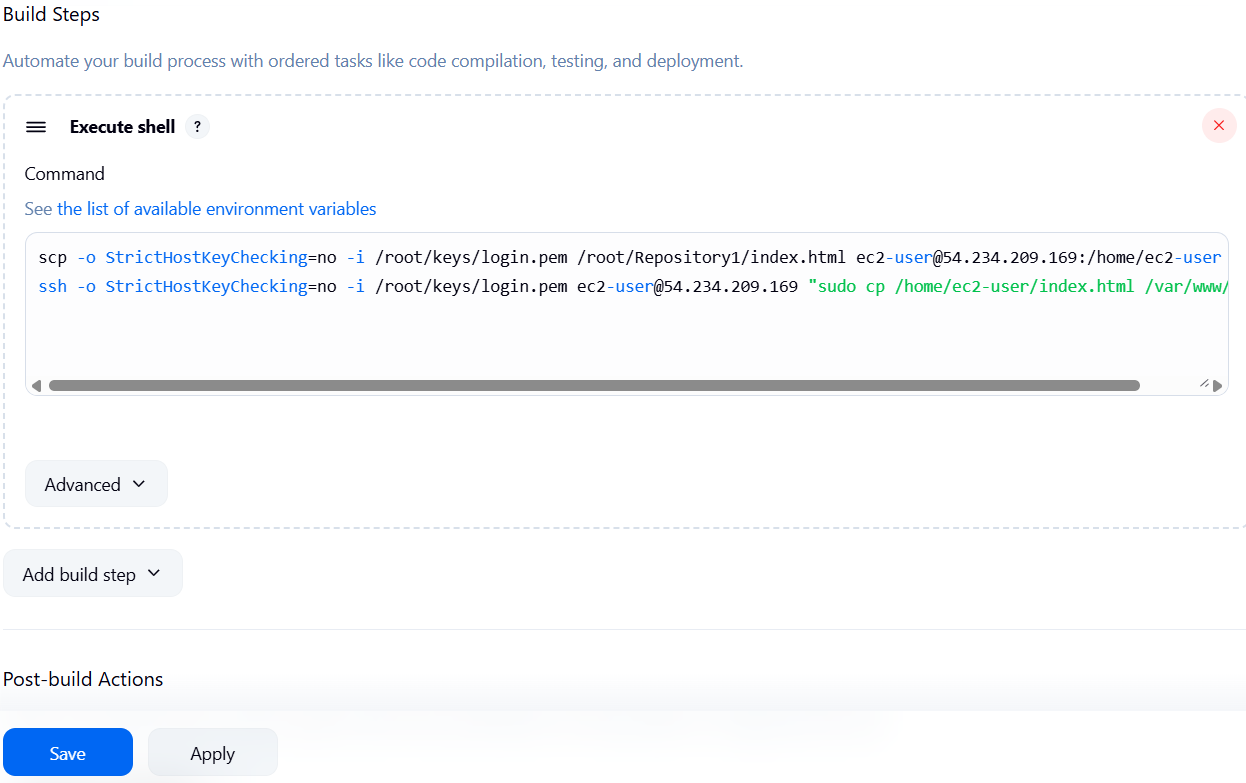
In Triggers, select GitHub hook trigger for GITScm polling → Apply → Save



In Build Steps, select Execute shell → write commands to copy file from Jenkins-Master to Deployment-Server

scp -o StrictHostKeyChecking=no -i /root/keys/login.pem /root/Repository1/index.html ec2-user@54.234.209.169:/home/ec2-user

ssh -o StrictHostKeyChecking=no -i /root/keys/login.pem ec2-user@54.234.209.169 "sudo cp /home/ec2-user/index.html /var/www/html/"



Step 6: On Jenkins-Master create .pem file.

cd /root

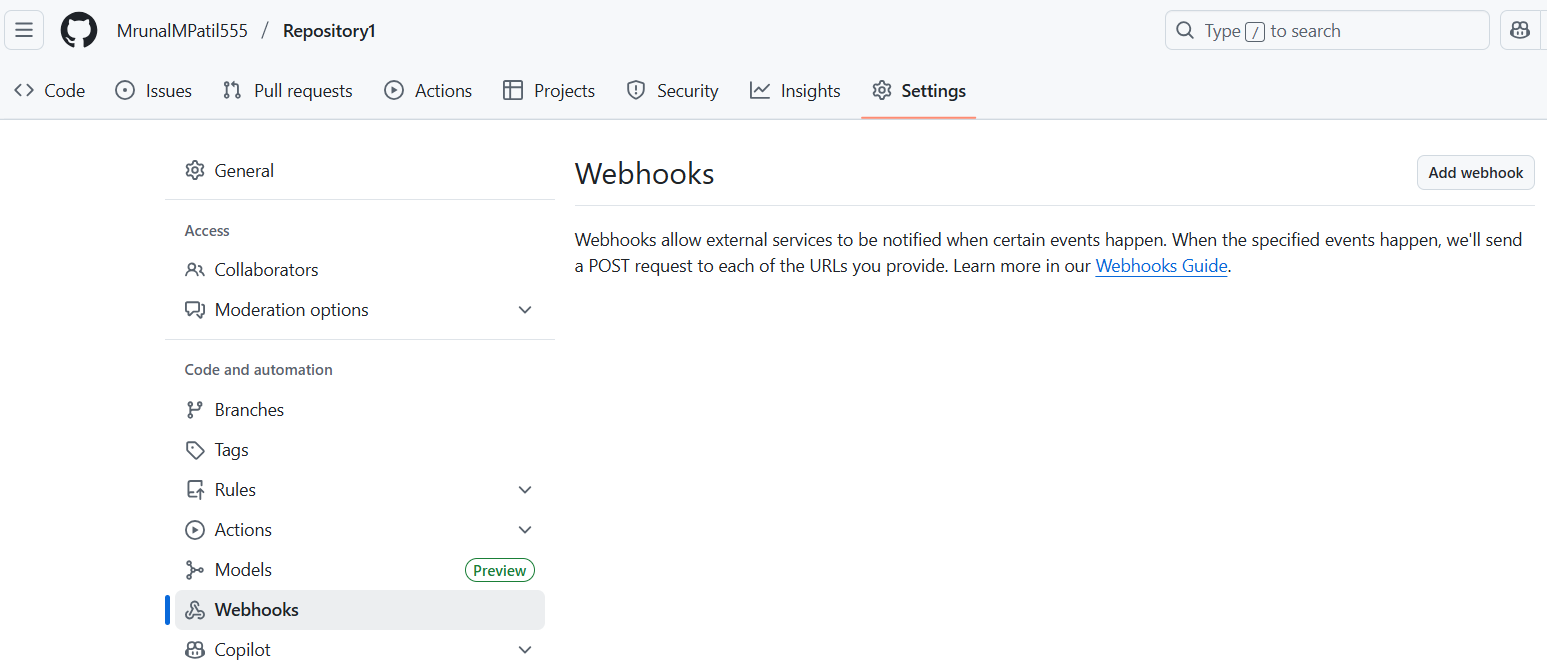
mkdir keys

cd keys

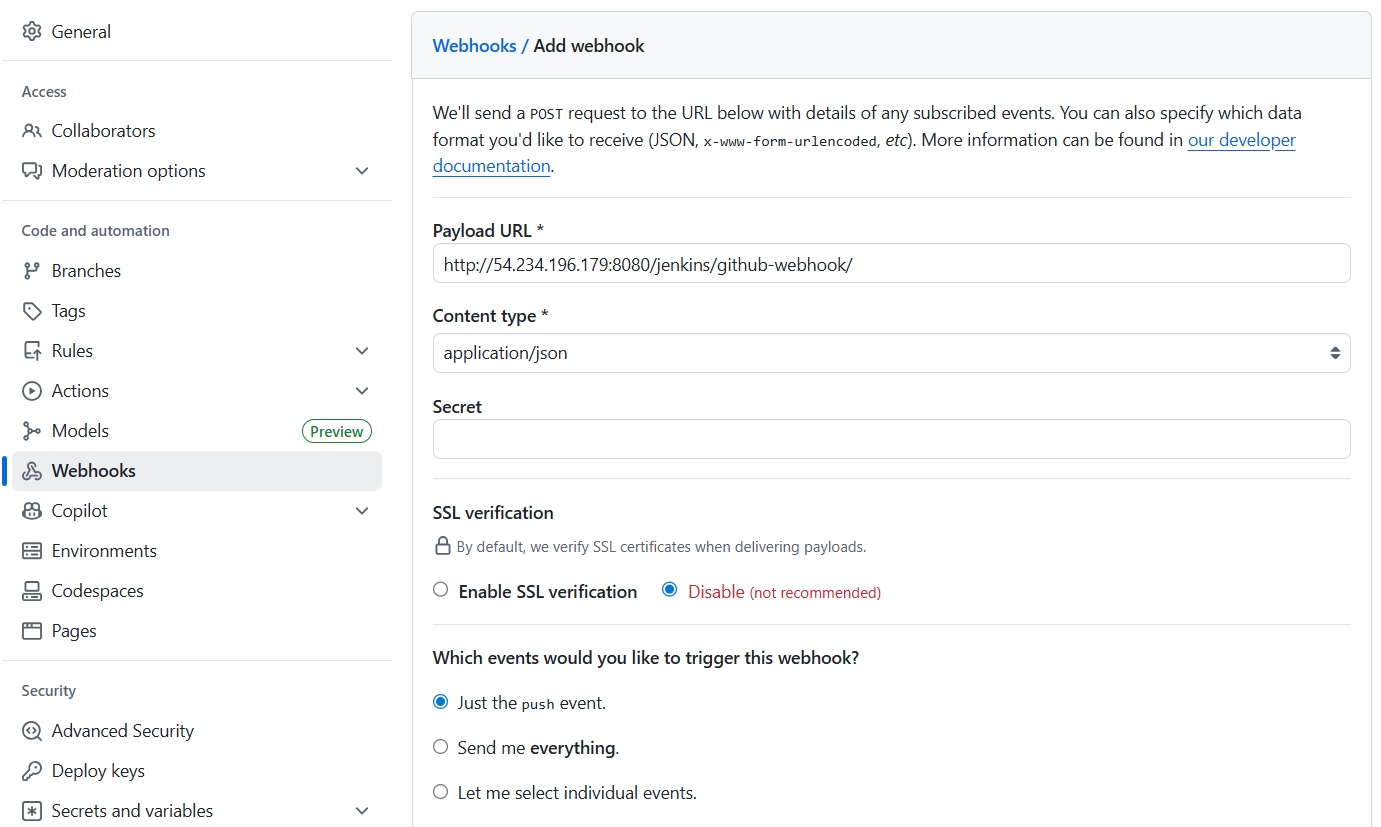
vi login.pem >> copy the .pem file

chmod 400 login.pem

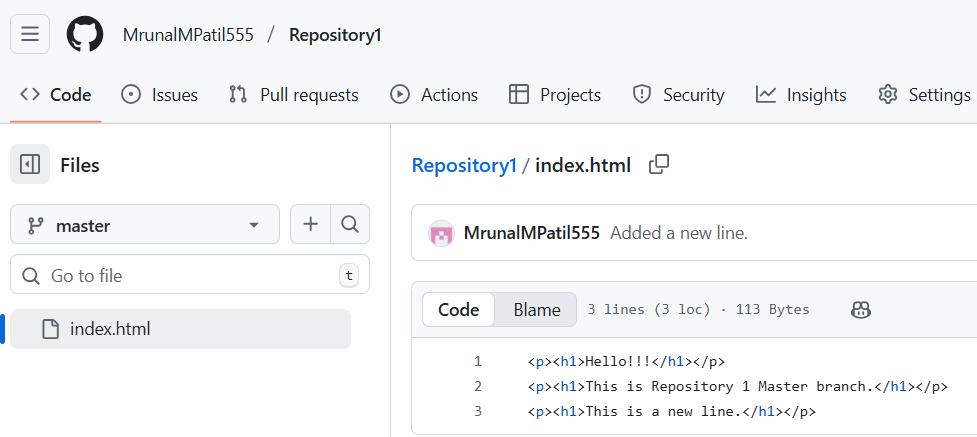
Step 7: go to your github repository → settings → webhooks → add webhook

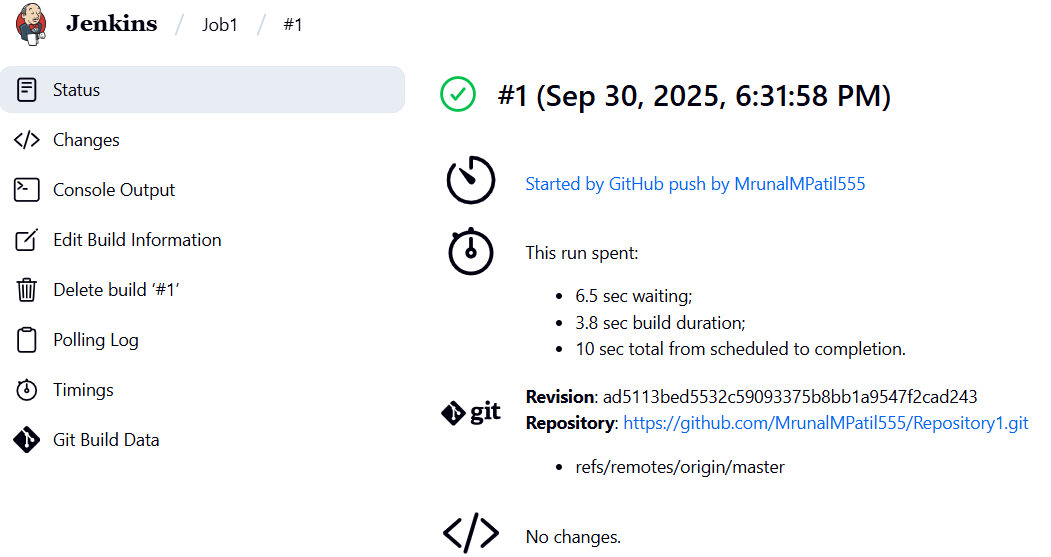


In Payload URL, enter Jenkins URL → In Content type, select application/json → Disable SSL verification → In which events to trigger, select just the push event → Add webhook

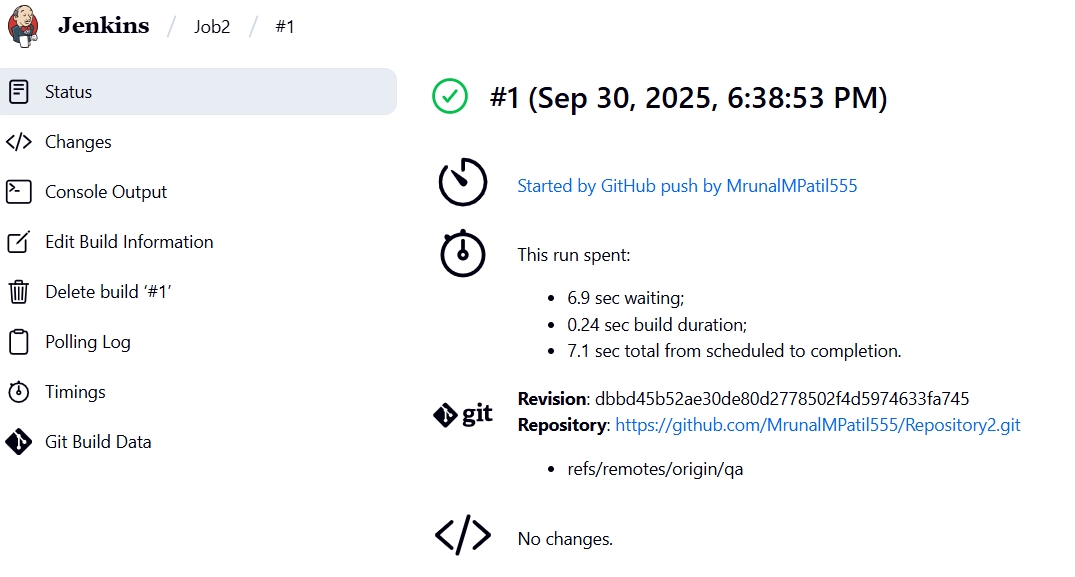


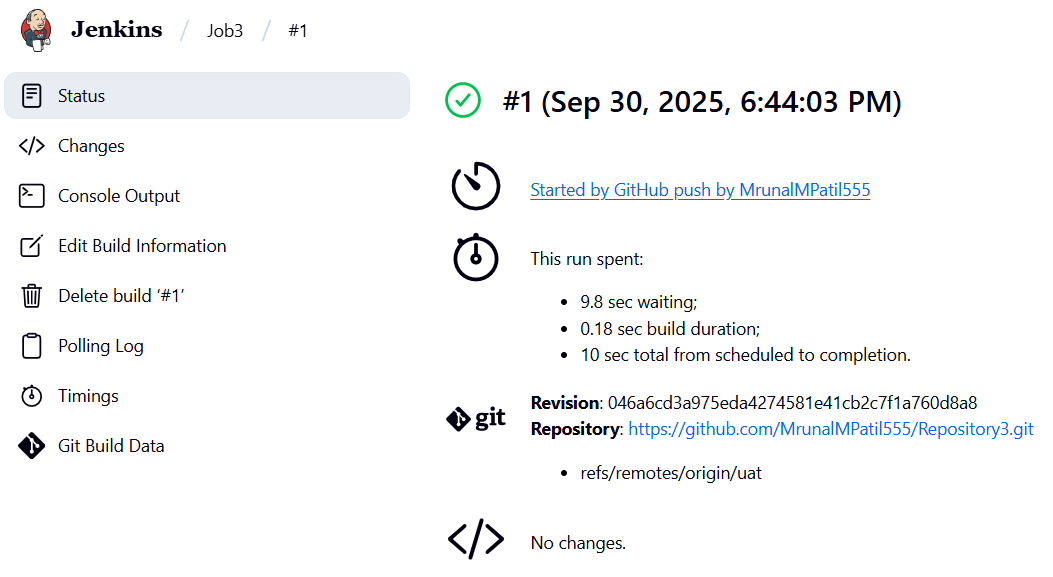
Step 8: make some changes to your code in git. Go to Jenkins, you will see the build has started.





Step 9: Similarly, create 2 more jobs for Repository 2 and Repository 3





Step 10: make changes to index.html file and push it to repository.

cd Repository2

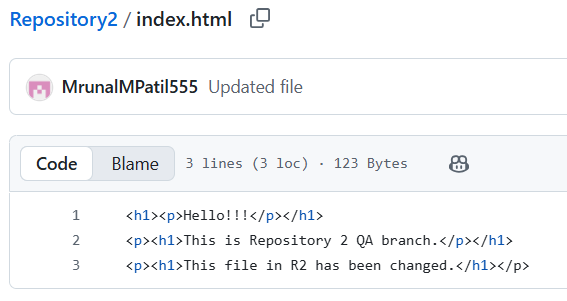
vi index.html >> This file in R2 has been changed.

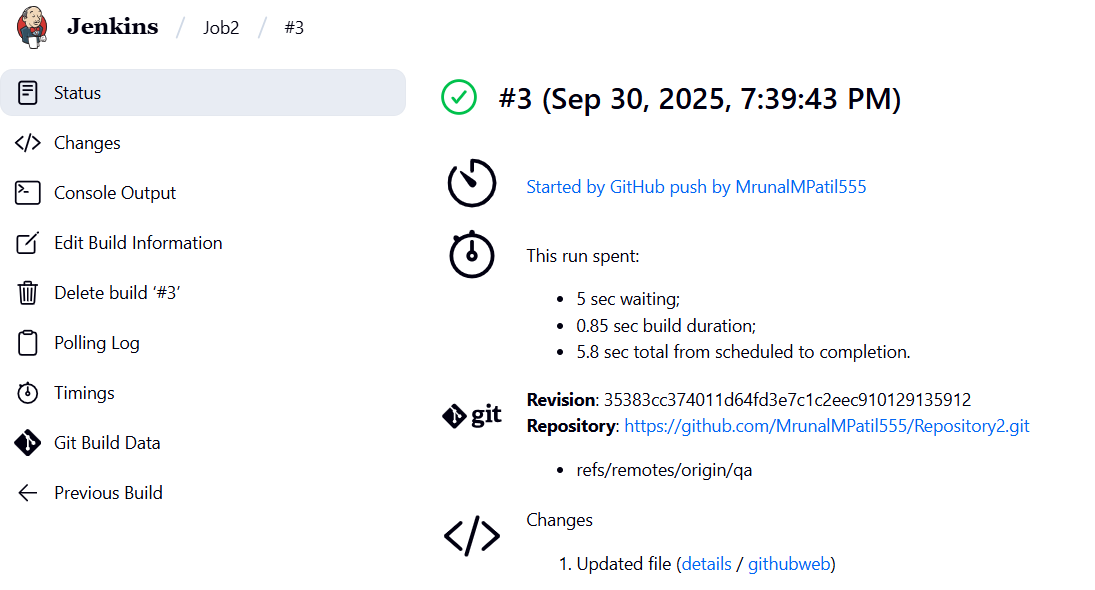
git add index.html

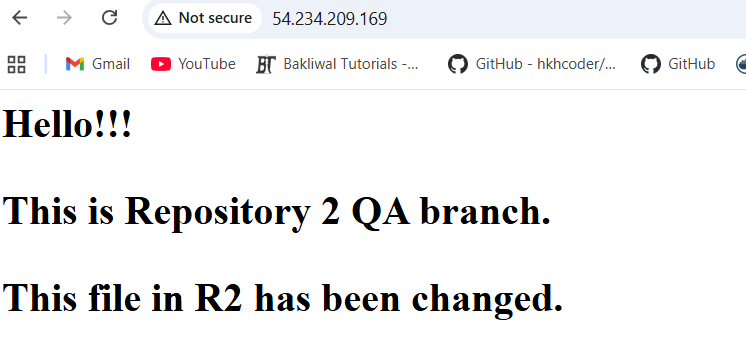
git commit -m “Updated file”

git push origin qa

You will see the index.html file in Repository2 has changed and in Jenkins the job has been built. On the server the index.html file has been deployed.







Step 11: Now if you change file in R3, you will see that file gets deployed.

