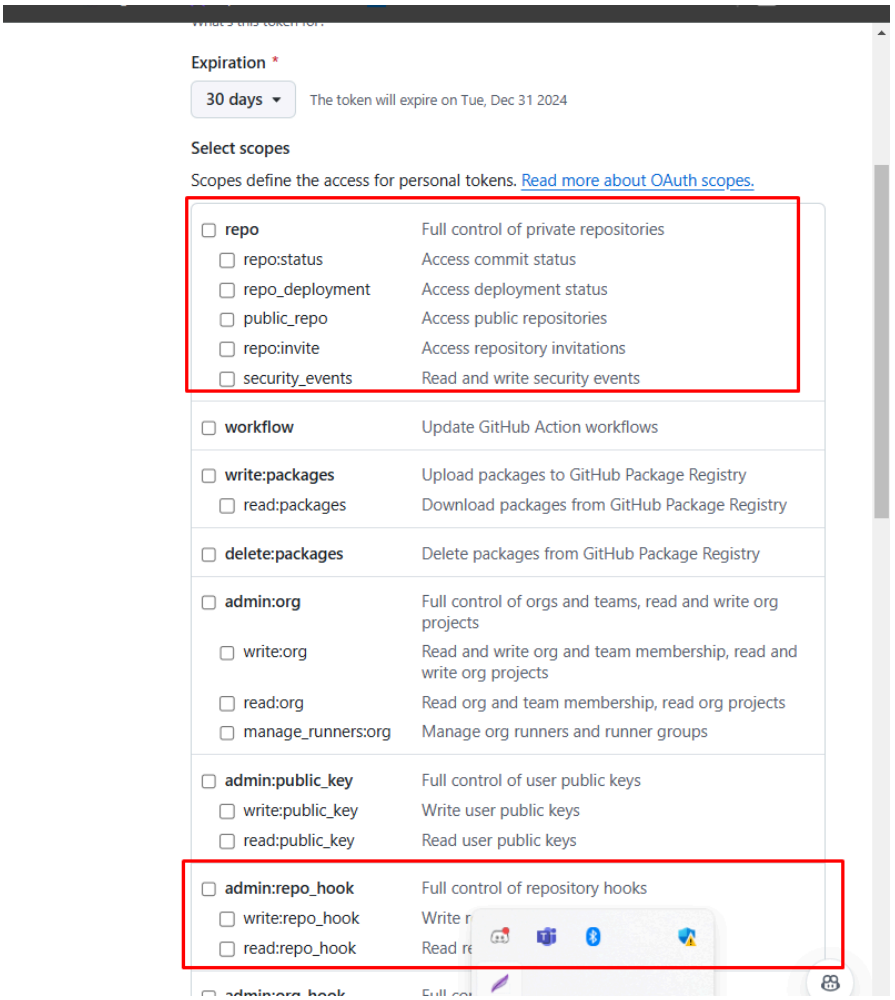
Implementing the Jenkins-GitHub Integration

This lab will show the integration of the github with the jenkins using the personal access and setting up Continous integration pipeline project using webhooks trigger.

Step 1: Generate a GitHub Personal Access Token

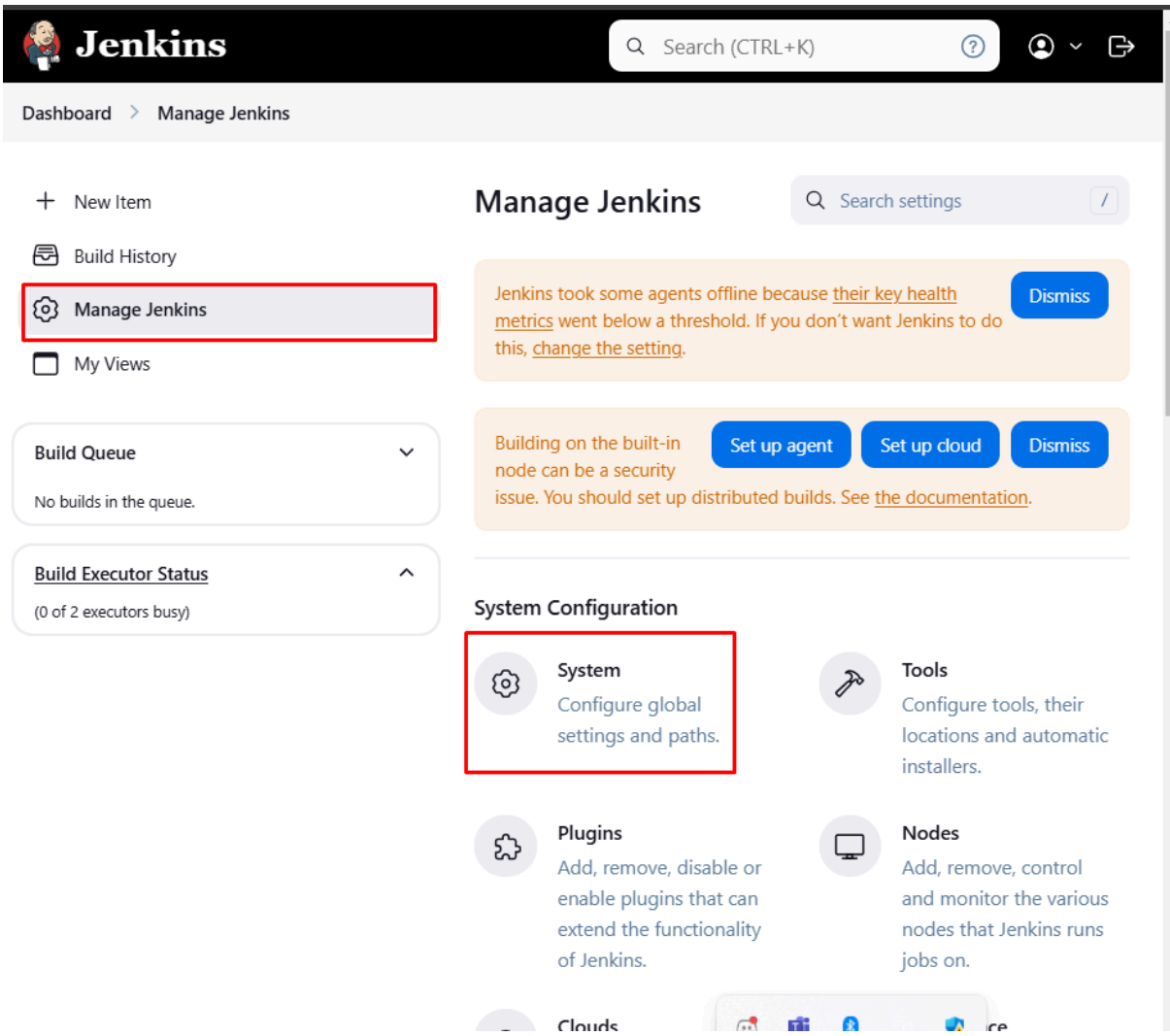
1. Go to your **GitHub Account > Settings > Developer Settings > Personal Access Tokens > Tokens (classic)**.
2. Click on **Generate new token** (classic), select scopes:
    - `repo` for accessing private repositories.
    - `admin:repo_hook` for managing webhooks.



3. Copy thce token and paste it somewhere.

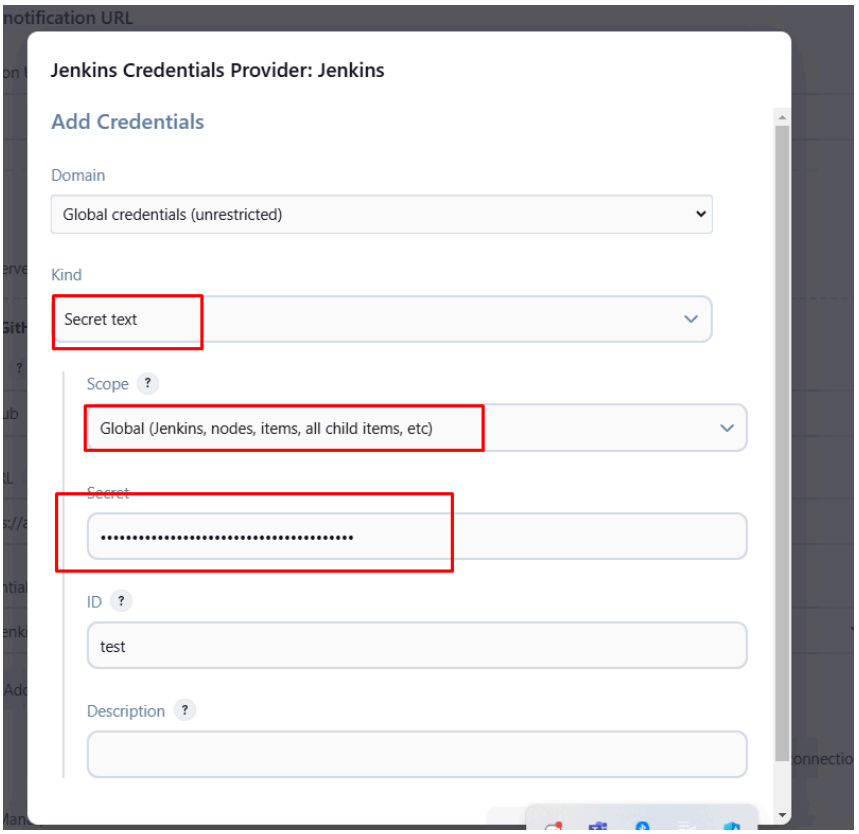
Step 2: Set Up Jenkins(install the Jenkins on AWS EC2)

1. **Install Necessary Plugins**
    - Go to **Manage Jenkins > Plugins > Available plugins**.
    - You can first even go to the installed plugins and search if they are already installed for you
    - Install the following plugins:
        - **Git Plugin**
        - **GitHub Integration Plugin**
        - **GitHub Branch Source Plugin**
2. **Configure Global Settings**
    - Navigate to **Manage Jenkins > System**.



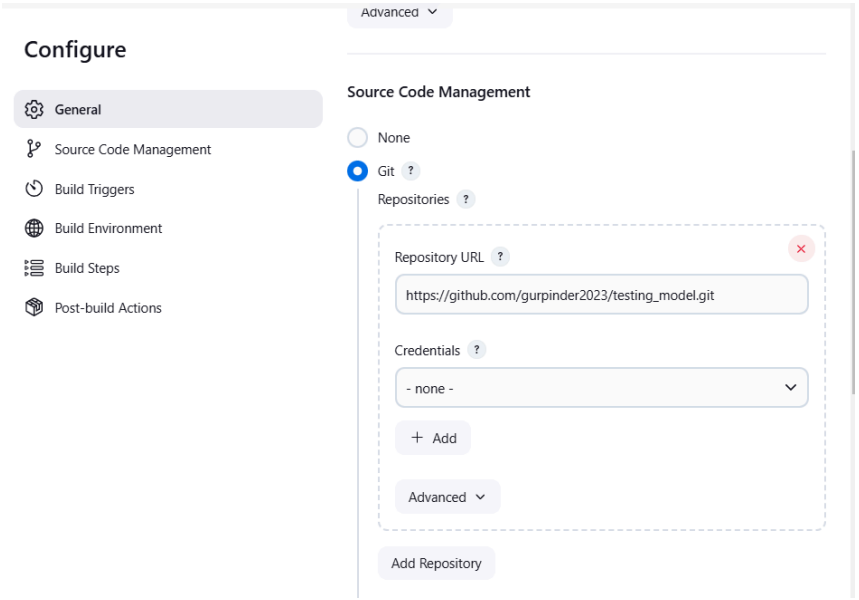
    - Under **GitHub**, click **Add GitHub Server**.
        - Provide a name which can be any name , keep the api url as it is.
        - Under the credentials click on add.

- Make sure to select global for the domain and for kind select secret text. And scope it global and then paste the secret that you copied in step1 and name your credential in id.

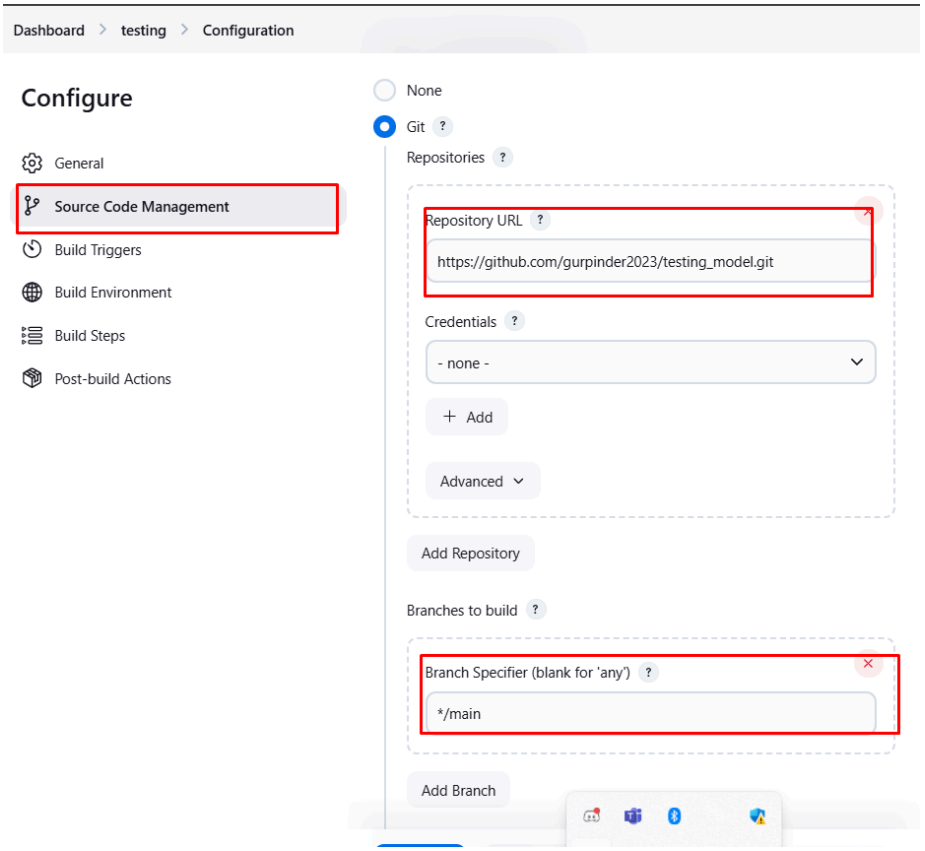


Step 3: Create a Jenkins Job for the Repository

1. Go to Jenkins Dashboard and click **New Item**.
2. Choose **Freestyle project** or **Pipeline**, then enter a name and click **OK**.
3. Under **Source Code Management**:
    - Select **Git**.
    - Provide your repository URL.



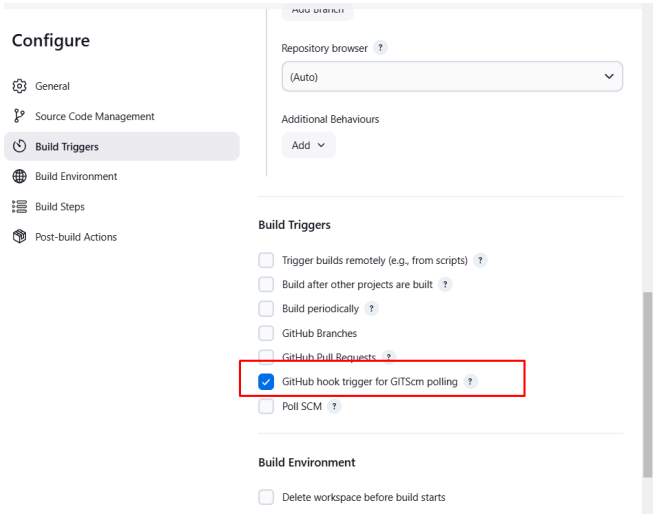
- Then put the branch you want to build from which is in your github, mine is main so i select main



- Use **Credentials** if required (username, password, or PAT). but as we already configured the credential in the system no need to do that.

4. Under Build Triggers:

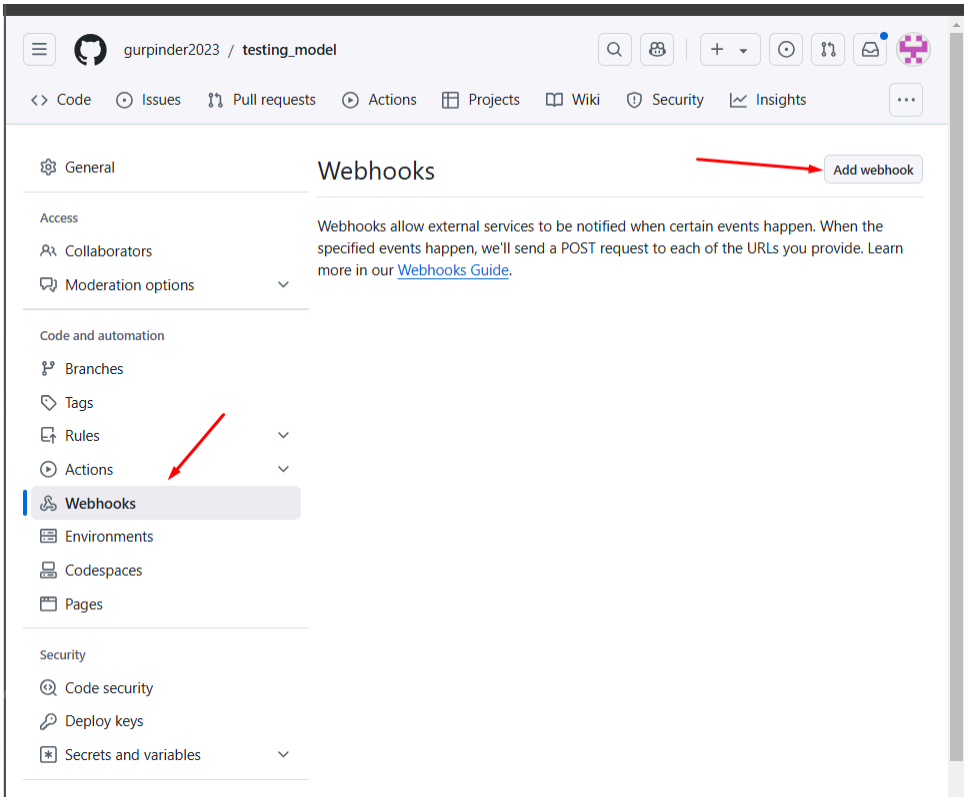
- Click on the Github hook trigger for GITScm polling.



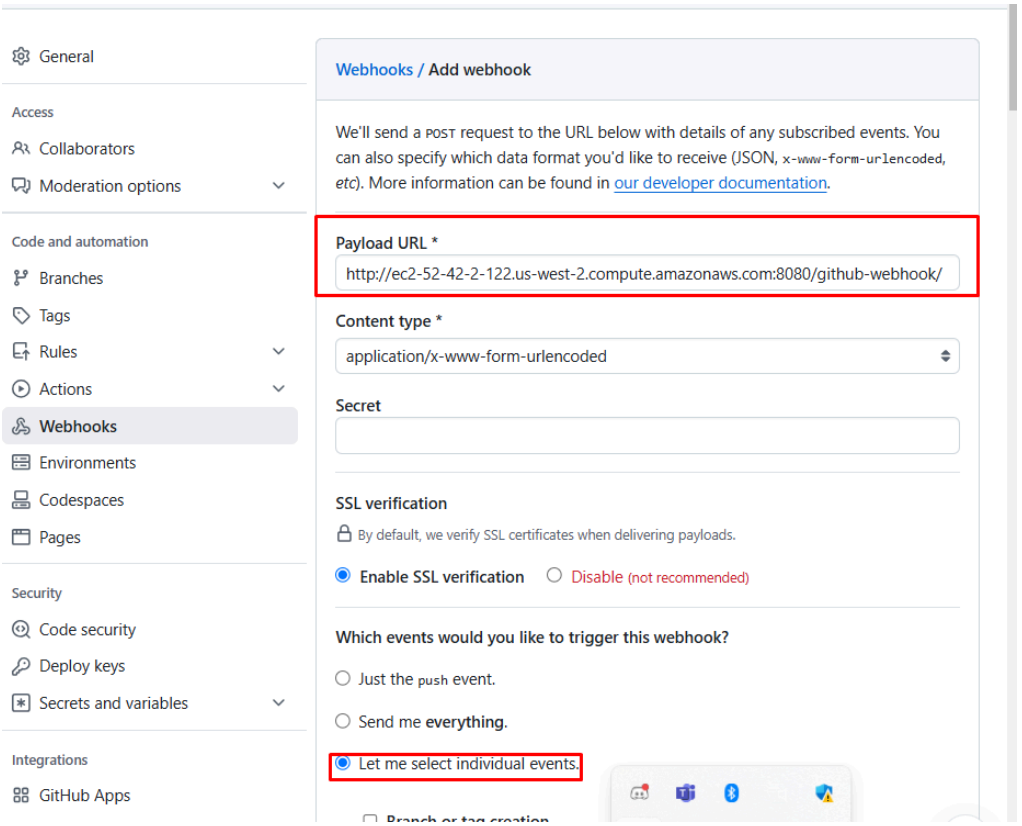
5. Save the project

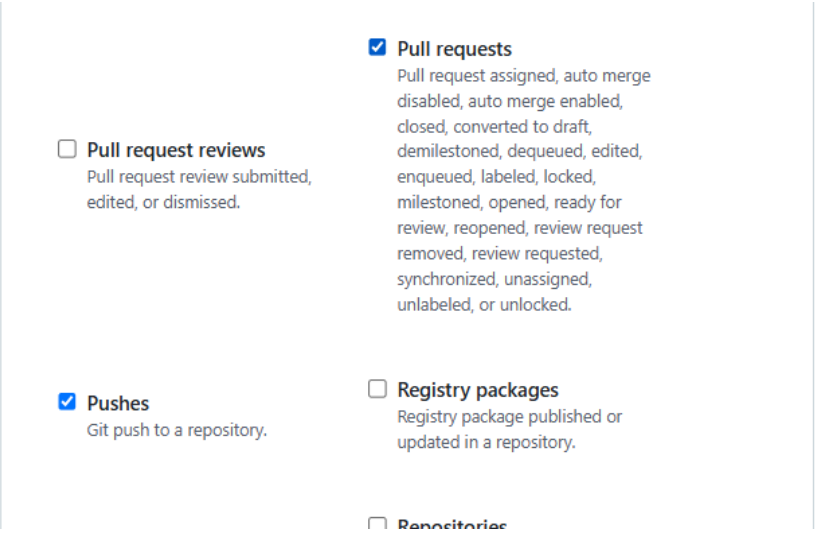
Step 4: Configure GitHub Webhook

1. Go to your GitHub repository.
2. Navigate to **Settings > Webhooks > Add Webhook**.



3. In the **Payload URL**, enter your Jenkins URL followed by `/github-webhook/`, e.g., `http://<jenkins-server-ip>:8080/github-webhook/`.
4. Click on let me select individual events. And select pull and pushes.

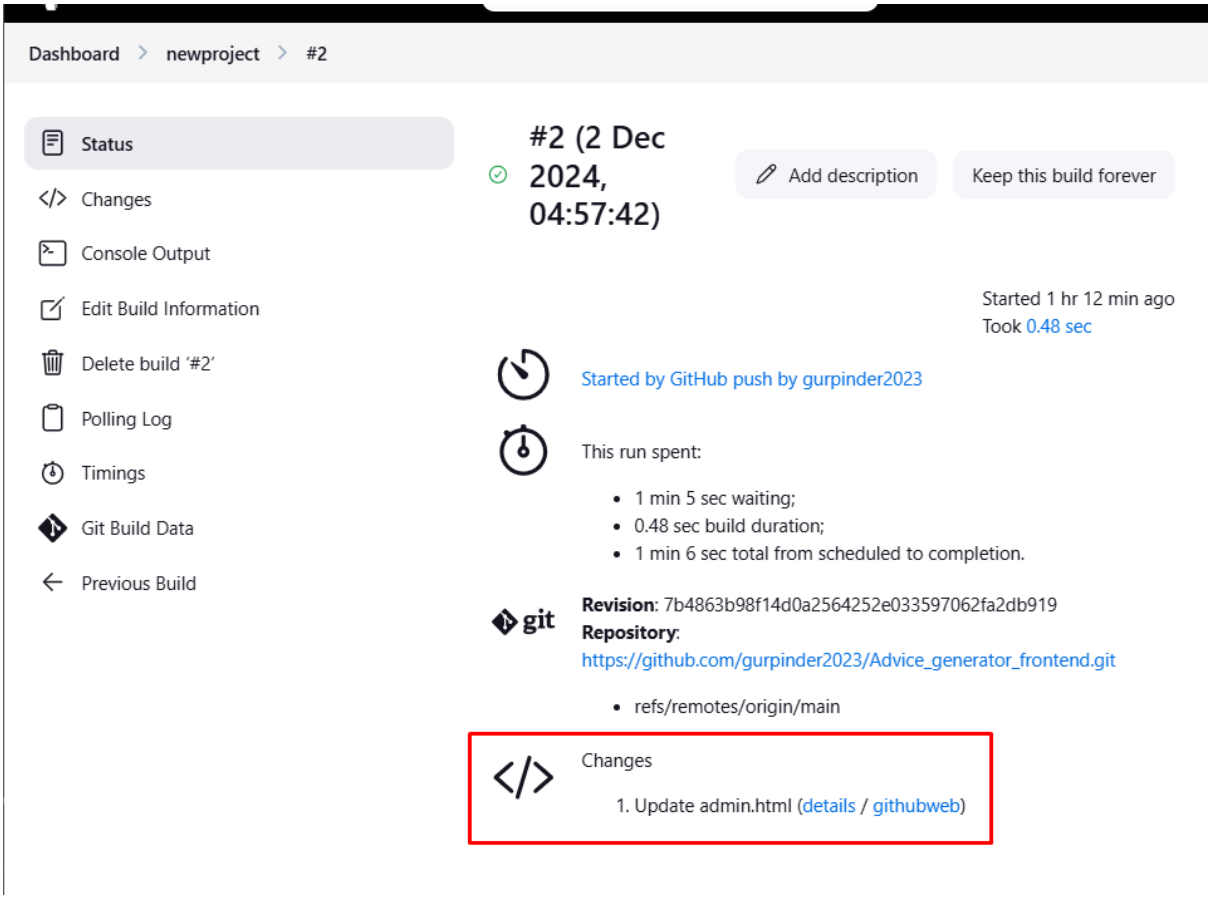




5. Save the webhook.

Step 5: Test the Integration

1. Push a change to your GitHub repository
2. Go to Jenkins and see if a build was triggered automatically.
3. The after the build success ful you will see the changes in the logs



Step 6 : (optional if want to deploy) Create a Jenkins Job for the Repository

1. If you want to deploy the simple nginx html files to test.
2. Install the nginx server on the instance using following commands:
    - sudo yum update -y
    - sudo yum install nginx -y
    - sudo systemctl start nginx
    - sudo systemctl enable nginx

3. Choose **Freestyle project** or **Pipeline**, and follow all the above steps from 1 to 5
4. Under the **Build Steps:** Select execute shell and type the following.

```
# Navigate to the Jenkins workspace

cd $WORKSPACE



# Copy files to the Nginx web directory

sudo cp -r * /usr/share/nginx/html/



# Restart Nginx to apply changes

sudo systemctl restart nginx
```

5. Build might fails because jenkins can't run sudo commands because it does not have access to sudo so go to the git bash and type
    - sudo visudo
    - Add the following line at the end of the file:

      bash

      Copy code

      ```
      jenkins ALL=(ALL) NOPASSWD: /bin/cp, /bin/systemctl
      ```

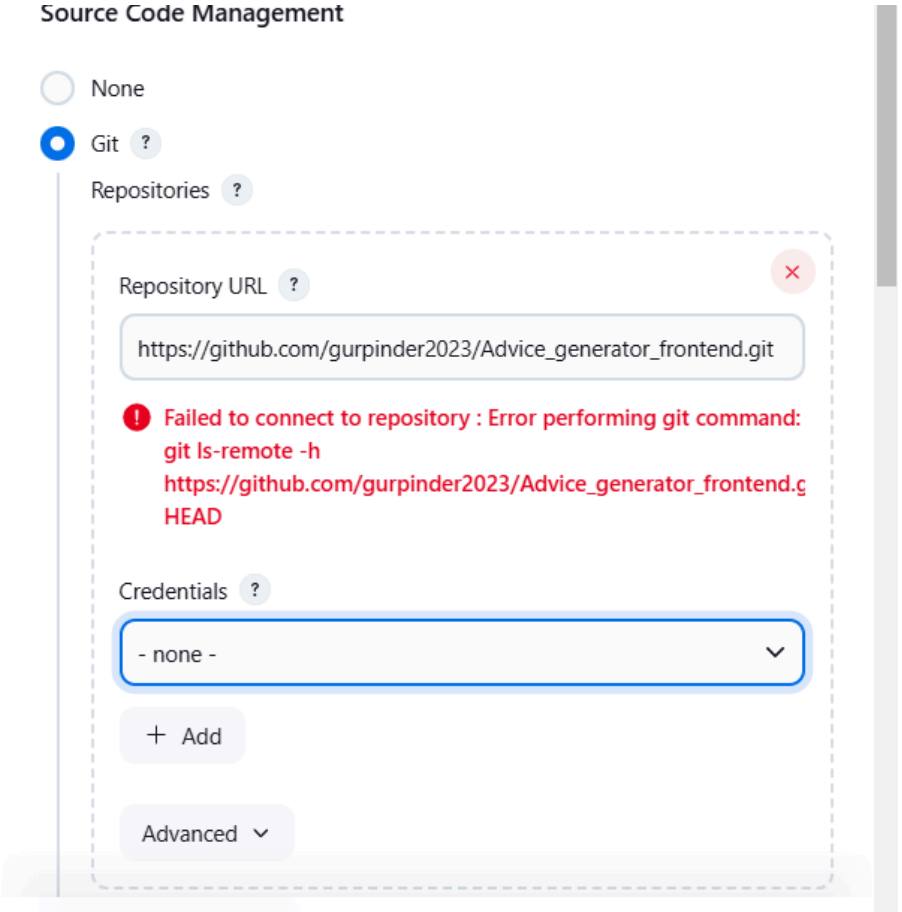
      This allows the `jenkins` user to run `cp` and `systemctl` commands without being prompted for a password.

6. Deploy the project again and then access the page using http:// public-ip-address:80 at port 80.
7. Now if you make any changes it will be visible in deployed.

TROUBLESHOOTING GUIDE:

1. Github integration error:

Make sure you install the git in the instance by using the commands like
sudo yum update -y
sudo yum install git -y
git –version
This will avoid the errors like the one shown below



2. Unable to build:

Sometimes when you build the project it continuously shows you in process then check the node
Go to the **Manage Jenkins > nodes** and see if the node is online and have **more than 1 executor** and if it is offline make it online then you will see that your project will be started building