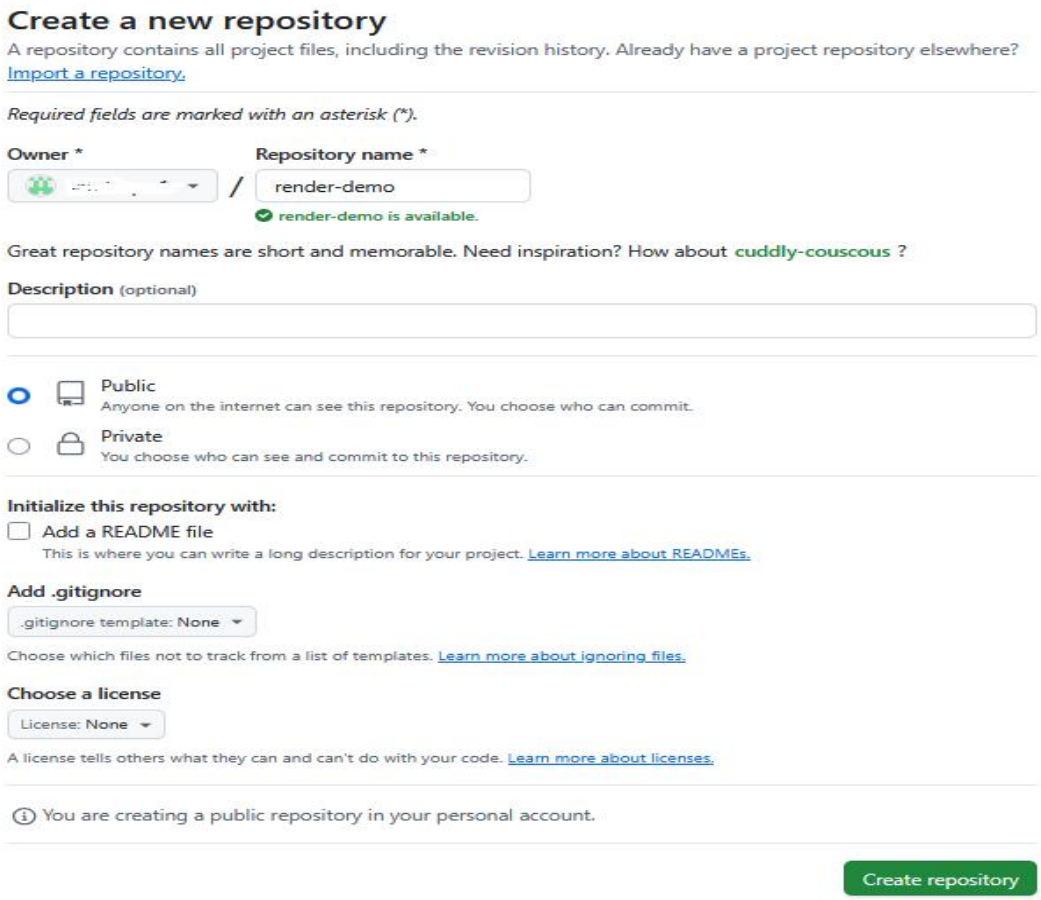
Deploy a Static Website on the Render Cloud

In this tutorial we will deploy a static website on the render cloud. A GitHub repository will be connected to the Render cloud website. A webhook will be configured for this GitHub repository so that when we push the code from our local repository to this GitHub repository, the webhook will activate the build in Render and it will update the website automatically.

1. Create a blank/uninitialized repository on GitHub.
Login to GitHub using your account. Click New button to create a new repository.

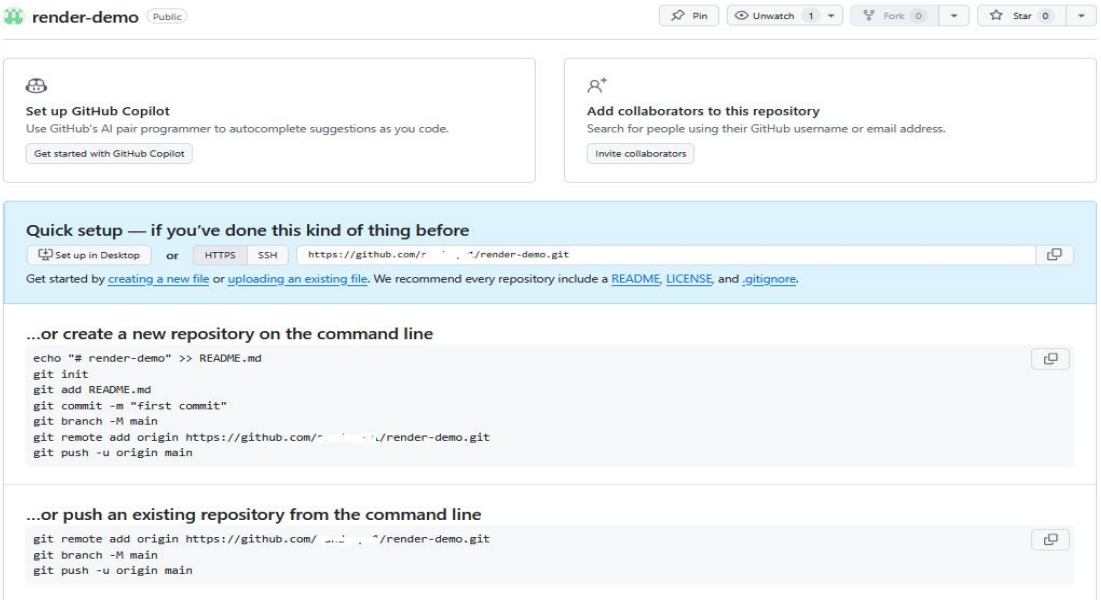
Provide a name for the repository. Here the namegiven is render-demo. Keep all other settings as it is. Do not select the check box of Add a README file.



Click Create repository button.

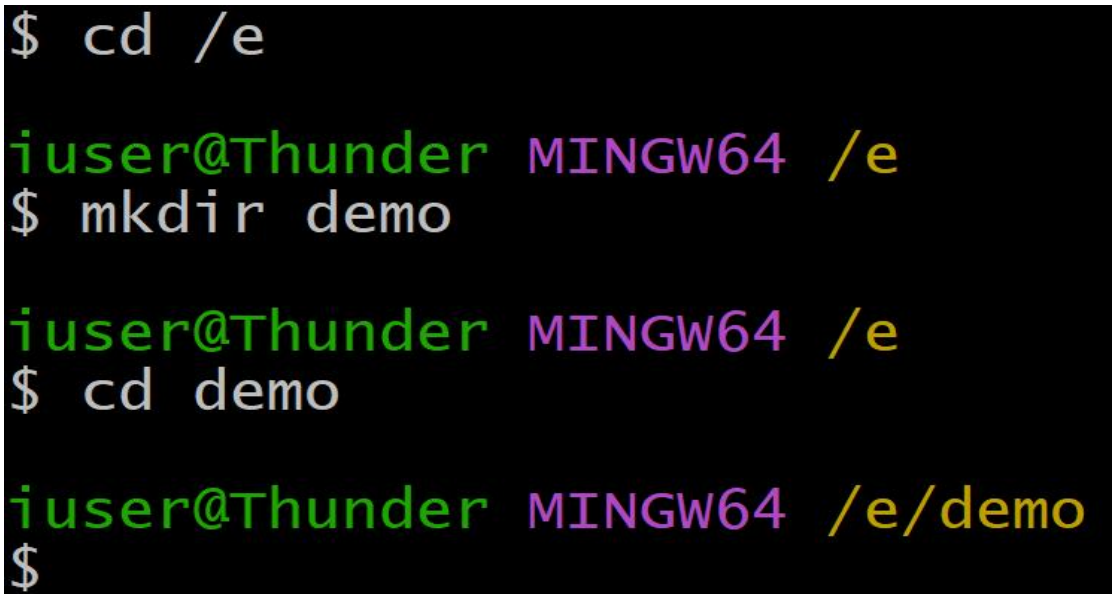
Once the repository is created, you should be ale to see the following page.

Do not close the browser.



2. Next we will create a local repository on our machine and link it to the above repository. We will create the files for the website in our local repository and then push these files to the above GitHub repository.

Go to your computer/Laptop. Open Git Bash for Windows or shell for Linux/Mac.



Here I went to my E drive. Then I created a directory by name demo. Then I went into the demo directory.
If you do not have a E drive on your machine then you can create a directory on C or D drive also.
Make sure you enter the demo directory.

Now we will create an index.html file inside it. Then initialize this directory as a git repository. Make sure the name of the file is index.html.

```
iuser@Thunder MINGW64 /e/demo
$ vim index.html
```

Press I button to go into the insert mode. Type following in the file.

```
This site is under construction!!!
~
~
```

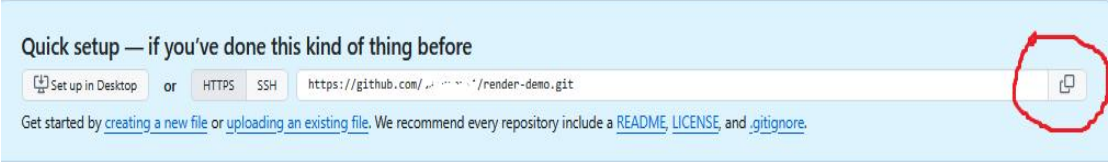
To save the file press Esc button. Then type **:wq**.
Now we will initialize this directory as git repository and commit the file. Give the following commands.

```
iuser@Thunder MINGW64 /e/demo
$ git init
Initialized empty Git repository in E:/demo/.git/
```

```
iuser@Thunder MINGW64 /e/demo (master)
$ git add -A
```

```
iuser@Thunder MINGW64 /e/demo (master)
$ git commit -m "index.html added"
```

Now we will link the local repository to the remote GitHub repository.
For this go to the GitHub web page in the browser. Then click the copy button to copy the URL of your repository as shown below.



Then go back to your git bash window. Type following command to link the local repository to the remote repository.
git remote add origin <paste your GitHub URL>

```
iuser@Thunder MINGW64 /e/demo (master)
$ git remote add origin https://github.com/ /render-demo.git
```

Now we will rename our local branch master to main. This is because the GitHub by default uses the default branch name as main.
Give the following command to rename the master branch to main.

```
iuser@Thunder MINGW64 /e/demo (master)
$ git branch -M master main

iuser@Thunder MINGW64 /e/demo (main)
$
```

Verify in the prompt that the branch name is now main.

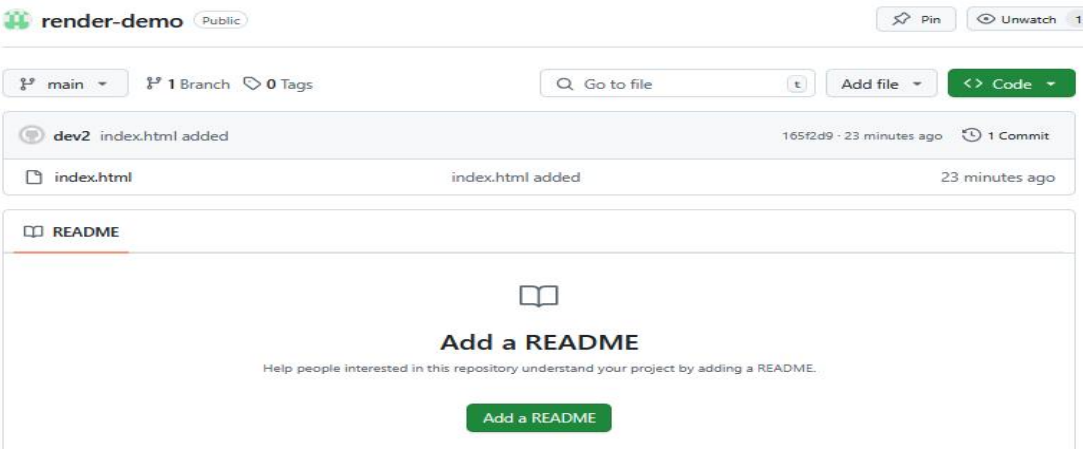
Now we will push the code to the remote GitHub repository.

```
iuser@Thunder MINGW64 /e/demo (main)
$ git push origin main
```

The command output should be as shown below.

```
$ git push origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 240 bytes | 240.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/       ./render-demo.git
 * [new branch]      main -> main
```

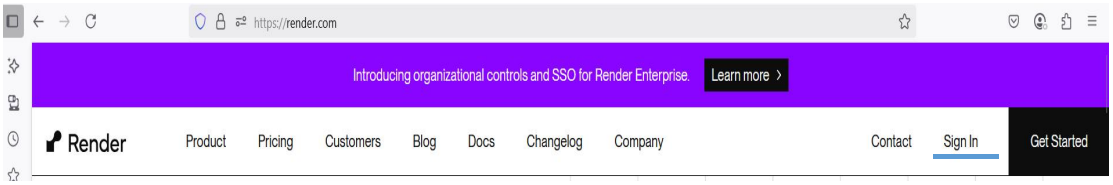
Now go to the GitHub repository and refresh the page. You should be able to see the index.html file as shown below.



3. Now we will create a static website on render using this GitHub repository.

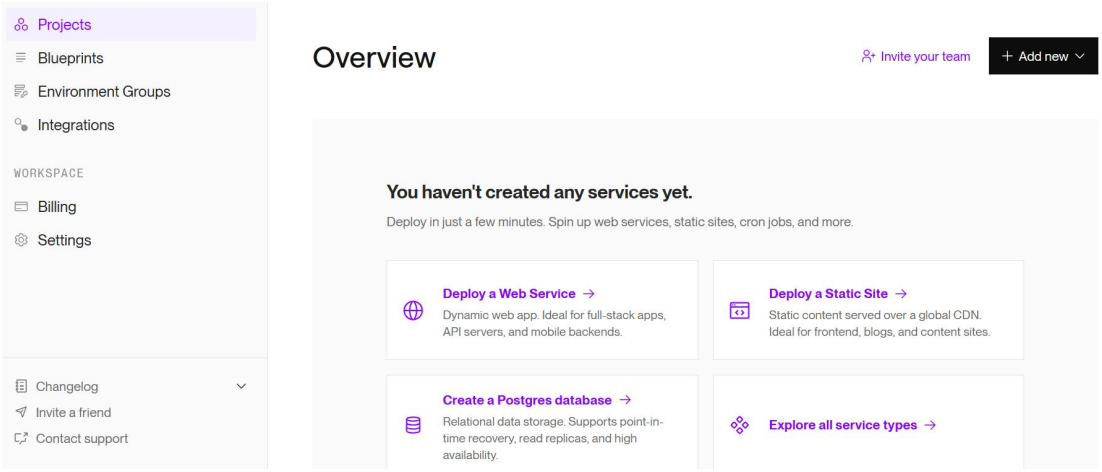
Go to the render.com website.
https://render.com/

Once the website opens, click Sign In option on right top corner.



Use your Google ID or GitHub ID to login.

Once you login, you will get following web page.



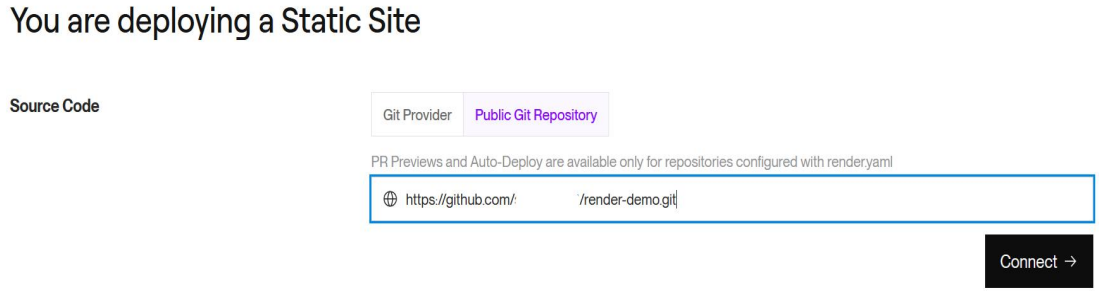
If you get a different page, still you will get the Add New button marked in black colour.

Click Add New button and select Static Site option as shown below.

OR

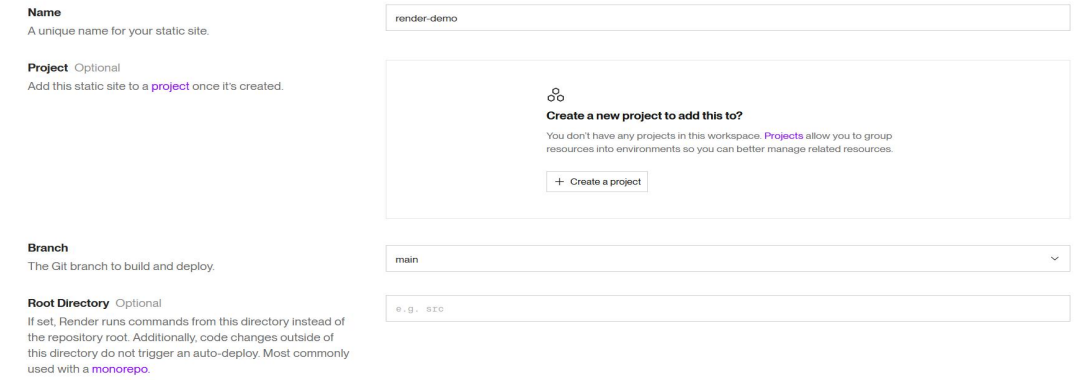
You can click the Deploy a static Site option as shown below.

Once you select the above option then following web page will open.

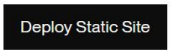


On this page click Public Git Repository option. In the box below paste the URL of your GitHub repository. Click Connect button.

Following options will be displayed.

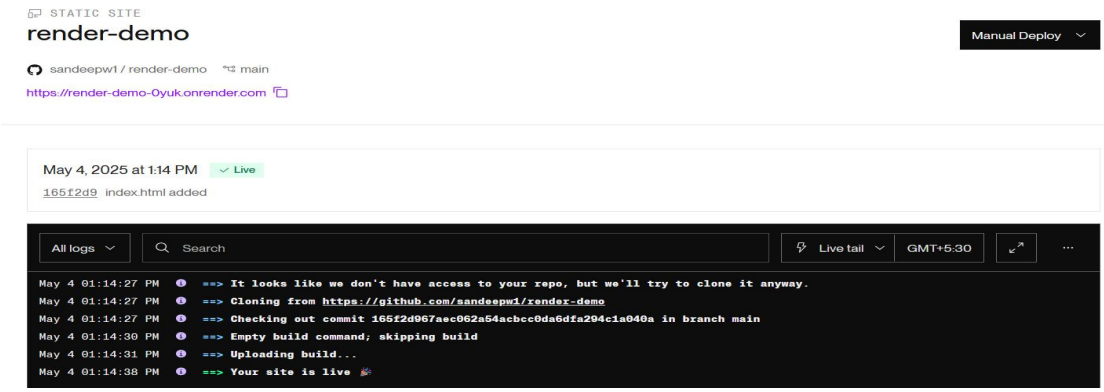


Do not change anything. Scroll down and click

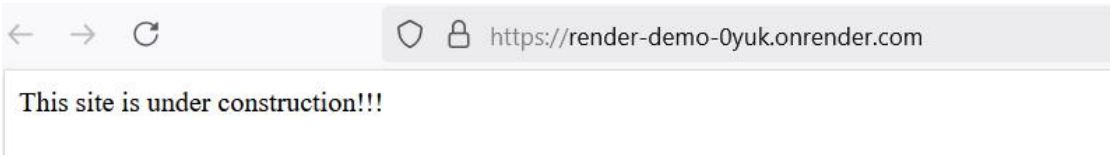


In case if it gives error for Publish Directory option, type . (dot) in the filed as shown below.

Once you click deploy static site button, following web page will open.

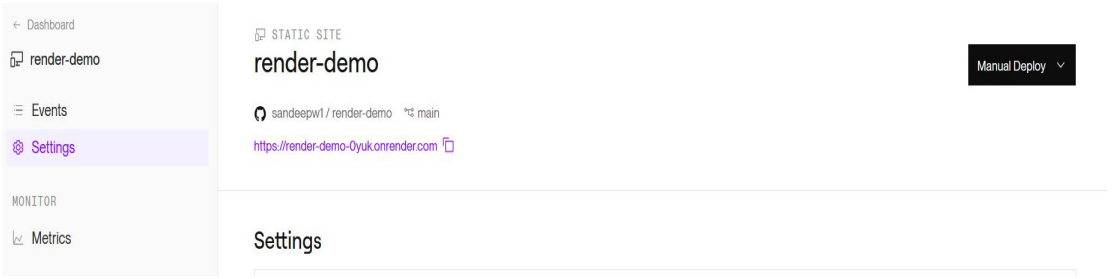


Click the URL shown in the above web page. It is shown in the purple colour. It is the URL where your web site will be available on the Internet. When you click the URL, a new tab will open in the browser and it will display the content of your index.html page.



This is how you have deployed your website on the Render.

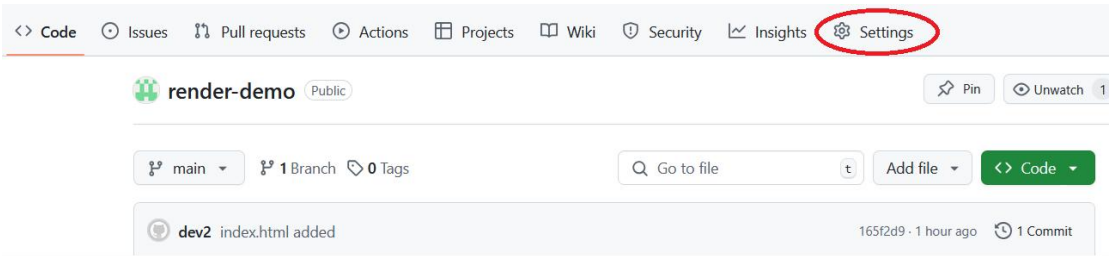
4. Next we want our website to update automatically when we modify the index.html file in our local repository and push it to the GitHub repository. For this we need to provide our website webhook URL to our GitHub repository. For this on the Render website page click the settings option on the left side as shown below.



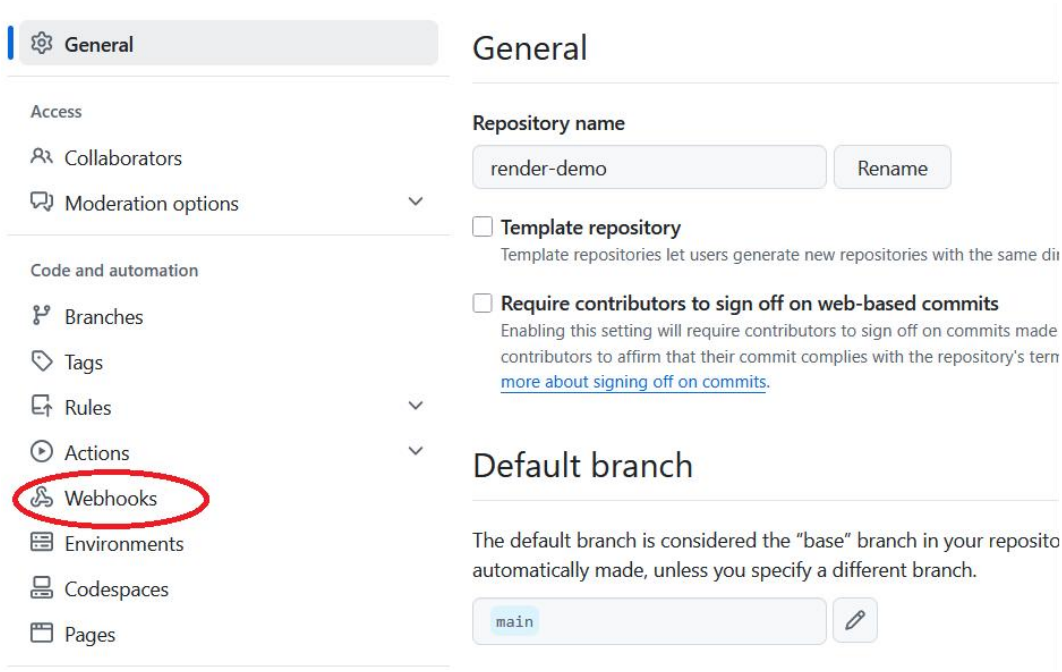
Scroll down the page to the Auto deploy section.

Click the copy button to copy the Deploy Hook. This is the URL where GitHub will inform Render about the changes in the repository code.

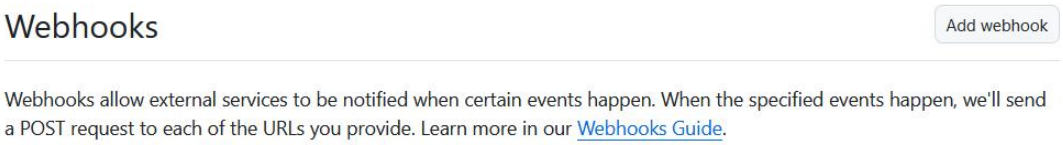
Now go to your GitHub repository tab. Click the Settings option as shown below.



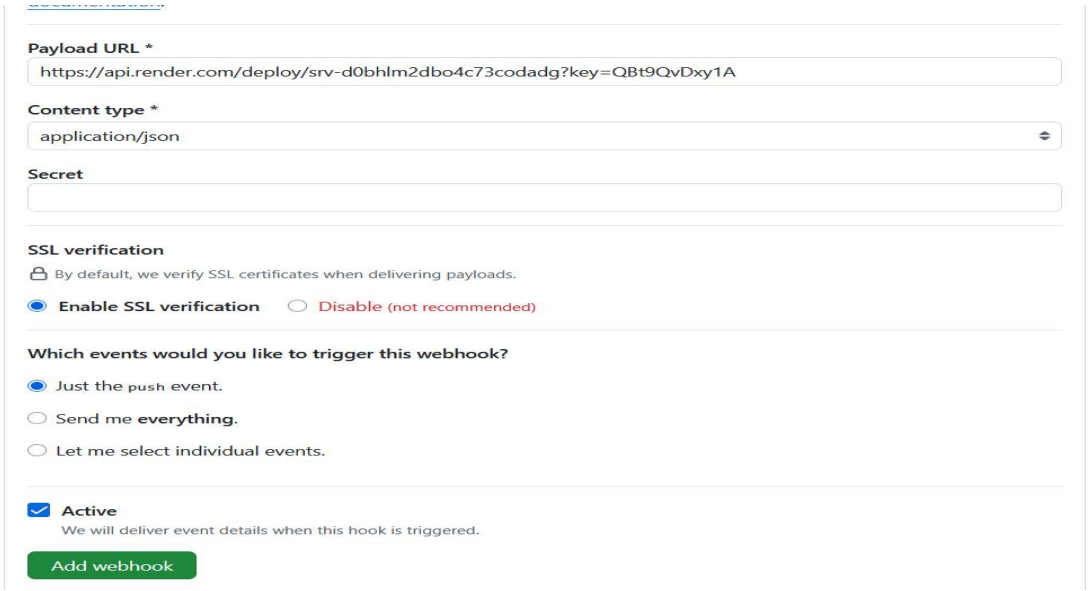
On the settings page, Click the webhooks option as shown below.



Following page will open.

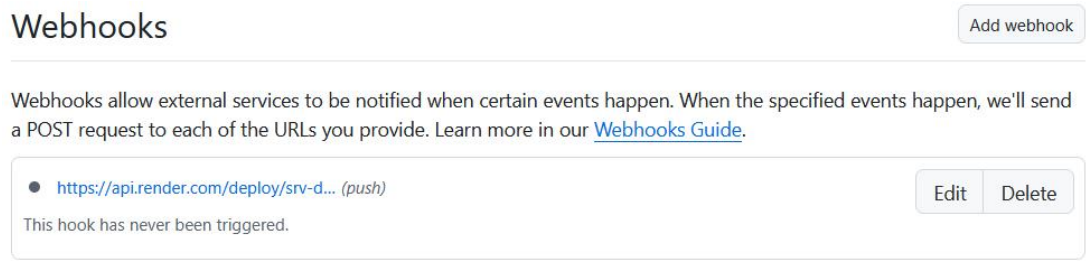


Click Add webhook button. Git may ask you to provide password for security reason. The following page will open.



In the Payload URL box paste the Deploy Hook URL copied from the Render. In the content type select application/json option as shown above. Then click Add webhook button.

The following page should be displayed.



Now go to the Git Bash on your local computer. Edit the index.html file.

```
iuser@Thunder MINGW64 /e/demo
$ vim index.html
```

```
This site is under construction!!!
```

Change the above current content of the file to following.

```
If you see this message, it means the render auto deploy is working!!!
```

Save the file (:wq).
Now add the file to staging area and then commit.

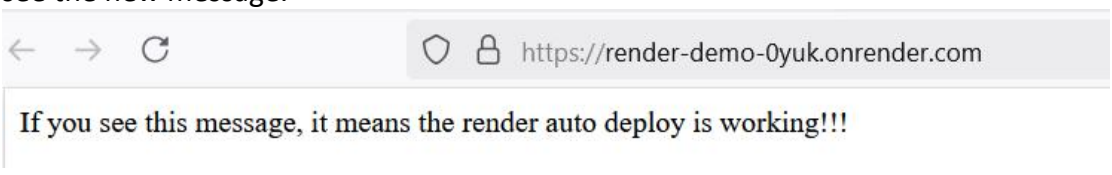
```
iuser@Thunder MINGW64 /e/demo (main)
$ git add -A
warning: in the working copy of 'index.html', LF
he next time Git touches it

iuser@Thunder MINGW64 /e/demo (main)
$ git commit -m "index.html modified 1"
[main c29a4f0] index.html modified 1
 1 file changed, 2 insertions(+), 1 deletion(-)
```

Now push the code to the GitHub repository.

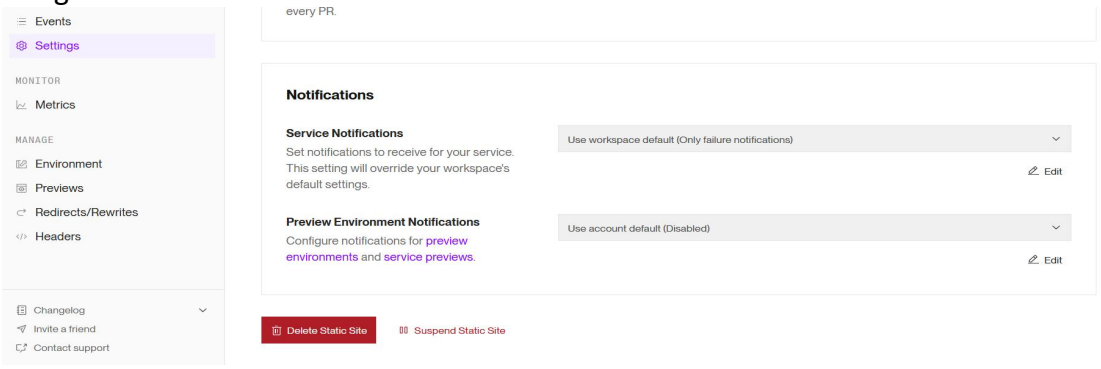
```
iuser@Thunder MINGW64 /e/demo (main)
$ git push origin main
```

Wait for 2 to 3 minutes. Go to your website URL and refresh. You should be able to see the new message.



Clean Up

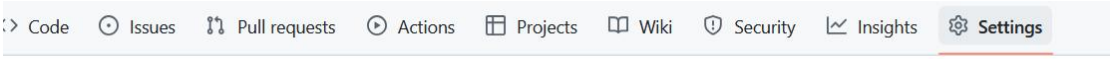
It is better to delete everything after the tutorial.
To delete the static website created on Render, go to the Render tab.
Click Settings option on the left side. Scroll down. At the end of the settings page you will get the Delete Static Site button. This is shown below.



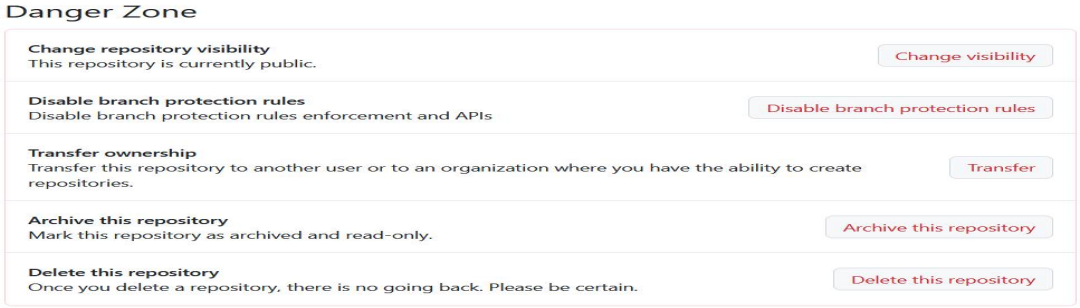
When you click the Delete Static Site button, following Window will open.

Either type sudo delete static site render-demo or copy paste in the box. Then the Delete Static Site button will be enabled. Click that button to delete the static site.

Delete the GitHub Repository.
Go to the GitHub in the browser.



Click the settings.
Scroll down to the Danger Zone.



Click Delete the Repository button. Follow the procedure to delete the repository.