

Deploy Jekyll Website to Github and Godaddy (Custom domain)

wxguy

2022-05-02 20:00:00 +0530

Contents

Understand Theme	1
Push Changes to GitHub	1
Changes at GitHub repo	2
Enable Custom Domain (Optional)	2

After I wrote my first blog post, I decided to go for pushing it to Github. It took some time to figure out how to do it. So, I decided to document it here.

Understand Theme

Most of the tutorials available online about deploying the Jekyll site to GitHub says that you simply push your files to your remote repo and GitHub would automatically build your site. But that is not true for all the themes. Each theme has its workflow as decided by its developers. They are generally documented in their wiki or documentation site. The Jekyll theme I use `Chirpy` is very well documented and instructions are clear on how to deploy. Therefore, spend some to reading and understanding the workflow mentioned in int documentation <https://chirpy.cotes.page>.

Publishing the local Jekyll site to the remote repo and publishing the same in <https://wxguy.github.io> involves modification to be done at the local and remote repo. Let's discuss them one by one.

Push Changes to GitHub

This `Chirpy` theme uses GitHub Actions to generate your site on a remote server. You need to have enough permission to do it in the remote repo. Go to [GitHub/username/username.io/settings](#) --> Actions --> General --> Scroll down the page to select "Workflow Permissions" and "Read and write permissions".

There is a shell script `deploy.sh` under the `tools` directory that will run on the remote server. After writing your article, you can push local files to remote through

`git` . Firstly, go to the root of your local repo and do the following to initialise your repo and add all files.

```
git init
git add .
```

You can check the status using the `git status` command. Though you can use `git commit -m 'Fresh deployment'` for staging your commit and `git push -u origin master` for pushing it to the remote repo, I could not succeed as it did not allow me to do so due to authentication failure. Therefore, I used <https://www.gitkraken.com/> Git GUI for pushing the files. I found it convenient and easy to use. It automatically used my ssh key to push the commit which also solved my authentication issue.

Changes at GitHub repo

Once the commit is made to GitHub, it will trigger the GitHub Actions workflow. This would run the `tools/deploy.sh` file on the remote server to generate the site. In my case, the workflow did not completed successfully. Then I went to the remote repo on GitHub and status of run under the `Actions` tab. I had to rerun the process to complete it successfully. This will create another branch `gh-pages` in the remote repo. The last step to be forwarded is to set the branch and directory for publishing your site. Go to GitHub repo --> settings --> Pages --> Select `gh-pages` as a branch and `/root` as a directory. Ensure to save the changes. That's it. Wait for a few min. The site `username.github.io` would be up and running after a few min.

It is highly recommended that you clear your cache and cookies before testing your site under `https://username.github.io` .

Enable Custom Domain (Optional)

This step is an optional one. If you wish to have your domain display your bog, then it is possible to do by following a few more additional steps. **Remember that it cost you money.** The first step is to buy a domain. There are many sites where you can buy your custom domain. I have chosen <https://www.godaddy.com> for the sole reason that it was cheap. You can use any other domain sale related site to purchase a domain name of your interest.

I bought `http://wxguy.in` . I had to modify at two locations to make the site live. Firstly, log in to <https://www.godaddy.com> and select `Manage Domain` which you will find under the newly purchased domain id. Then click `Domain Settings` and then click `Manage DNS` . Your domain settings will look like the below.

Type	Name	Data	TLL
A	@	custom	600 seconds
NS	@	ns51.domaincontrol.com.	1 Hour
NS	@	ns52.domaincontrol.com.	1 Hour

Type	Name	Data	TLL
CNAME	www	your-domain-name.	1 Hour
CNAME	_domainconnect	_domainconnect.gd.domaincontrol.com.	1 Hour
SOA	@	Primary nameserver: ns51.domaincontrol.com.	1 Hour

We need to insert four DNS addresses `185.199.108.153` , `185.199.109.153` , `185.199.110.153` , and `185.199.111.153` to point to GitHub. We can edit the first row (Type `A`) to insert `185.199.108.153` . Then add three more additional Type `A` rows to insert the remaining three IP addresses `185.199.109.153` , `185.199.110.153` , and `185.199.111.153` .

Then edit the `CNAME` row where `your-domain-name.` is indicated to insert your GitHub pages site name (`username.github.io`). Do not edit any other fields. The final table would now look like this.

Type	Name	Data	TLL
A	@	<code>185.199.108.153</code>	600 seconds
A	@	<code>185.199.109.153</code>	600 seconds
A	@	<code>185.199.110.153</code>	600 seconds
A	@	<code>185.199.111.153</code>	600 seconds
NS	@	ns51.domaincontrol.com.	1 Hour
NS	@	ns52.domaincontrol.com.	1 Hour
CNAME	www	<code>wxguy.github.io</code>	1 Hour
CNAME	_domainconnect	_domainconnect.gd.domaincontrol.com.	1 Hour
SOA	@	Primary nameserver: ns51.domaincontrol.com.	1 Hour

Now head over to your local repo and create a file called `CNAME` and write your domain name you have purchased in it ie. `wxguy.in` . Push the files to GitHub. Wait for a few min to take effect.

To check if all the options are loaded properly, you can go to `https://who.is` and enter your domain name `www.wxguy.in` in the search field. Once the site is listed click on `DNC Records` and it should show all the records you have entered on your domain like below.

DNS Records for wxguy.in

Hostname	Type	TTL	Priority	Content
wxguy.in	SOA	3600		ns51.domaincontrol.com dns@jomax.net 2022050104 28800 7200 604800 600
wxguy.in	NS	3600		ns51.domaincontrol.com
wxguy.in	NS	3600		ns52.domaincontrol.com
wxguy.in	A	600		185.199.109.153
wxguy.in	A	600		185.199.110.153
wxguy.in	A	600		185.199.111.153
wxguy.in	A	600		185.199.108.153
www.wxguy.in	A	3600		185.199.108.153
www.wxguy.in	A	3600		185.199.109.153
www.wxguy.in	A	3600		185.199.110.153
www.wxguy.in	A	3600		185.199.111.153
www.wxguy.in	AAAA	3600		2606:50c0:8000::153
www.wxguy.in	AAAA	3600		2606:50c0:8001::153
www.wxguy.in	AAAA	3600		2606:50c0:8002::153
www.wxguy.in	AAAA	3600		2606:50c0:8003::153
www.wxguy.in	CNAME	3600		wxguy.github.io

DNS Records for `wxguy.in` from *whois.com*

Alternatively, you can also check the status using the `dig` command.

```
$ dig wxguy.in

; <<>> DiG 9.18.2 <<>> wxguy.in
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 16649
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;wxguy.in.                IN      A

;; ANSWER SECTION:
wxguy.in.                 600     IN      A      185.199.108.153
wxguy.in.                 600     IN      A      185.199.111.153
wxguy.in.                 600     IN      A      185.199.110.153
wxguy.in.                 600     IN      A      185.199.109.153

;; Query time: 80 msec
```

```
;; SERVER: 192.168.1.1#53(192.168.1.1) (UDP)
;; WHEN: Mon May 02 23:09:15 IST 2022
;; MSG SIZE rcvd: 101
```

Now onwards, you can access your site on your own domain `wxguy.in` directly instead of `username.github.io` . Even if you try to access `username.github.io` , it will redirect to `wxguy.in` .