# sDig

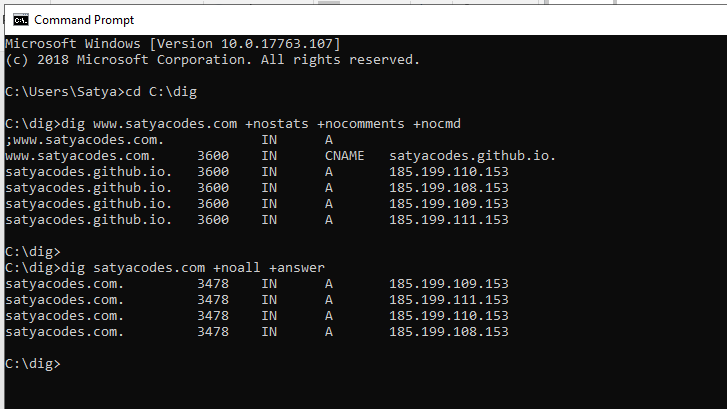
Download

<https://www.isc.org/download/>

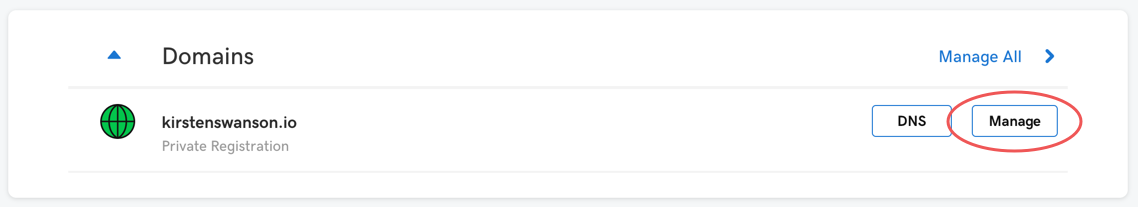
cd C:\dig

dig www.satyacodes.com +nostats +nocomments +nocmd

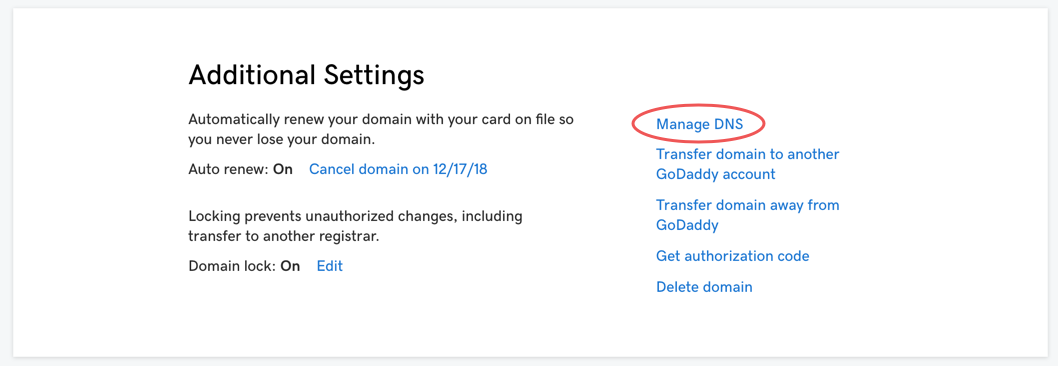
dig satyacodes.com +nostats +nocomments +nocmd

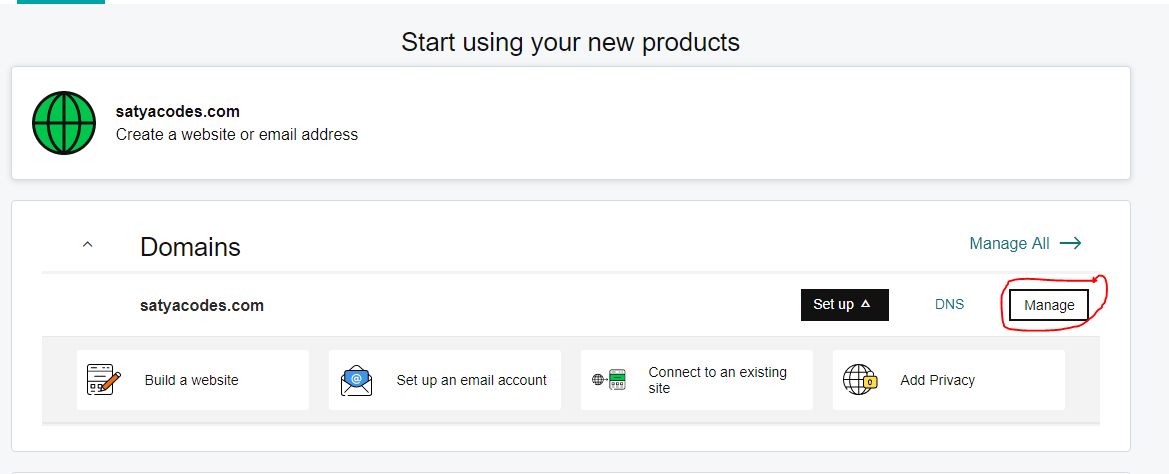


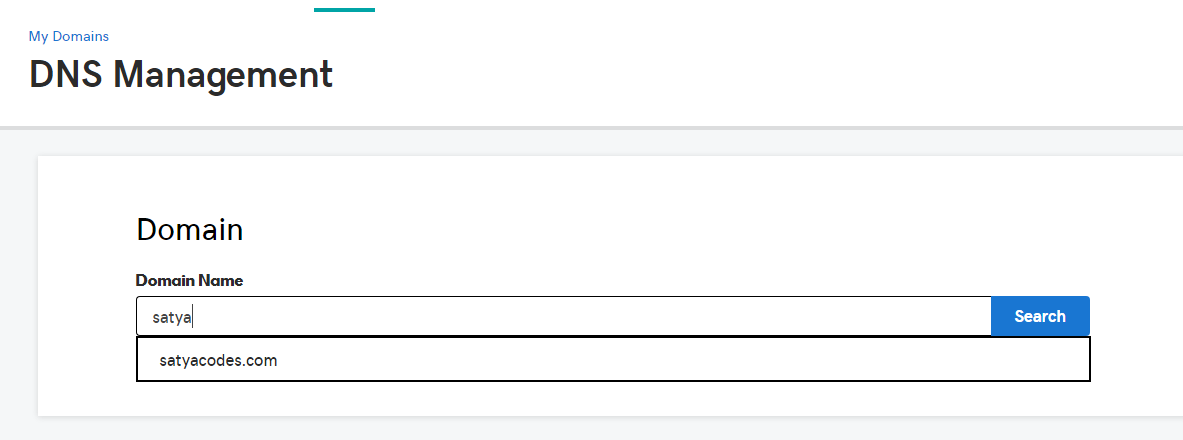
# GoDaddy – Git Pages



 Scroll down to the bottom of the page and click the “Manage DNS” button.

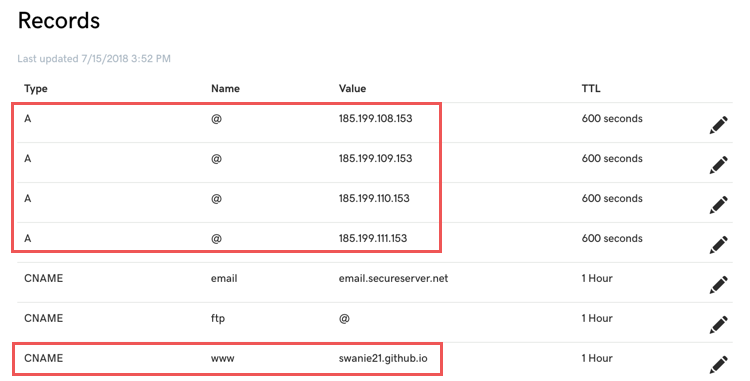


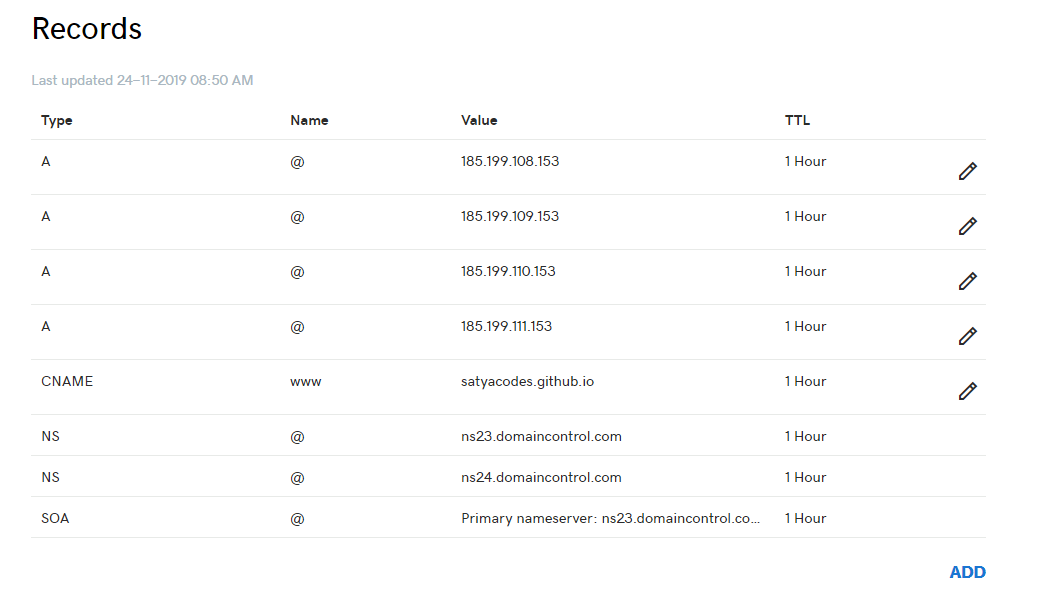




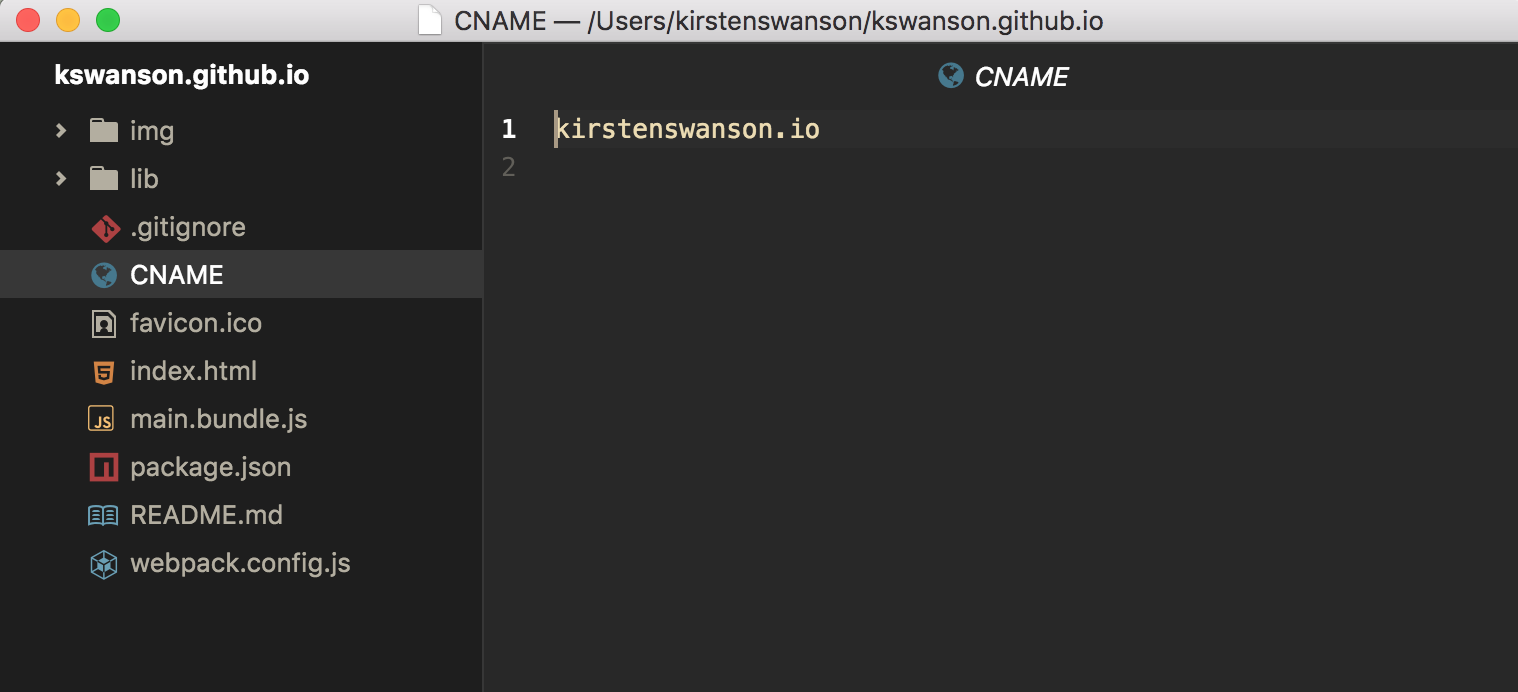
4. Within the DNS management page you will need to make five changes:

* In the Type “A” row update the IP address to: 185.199.108.153  
  (*this will point your custom domain to GitHub’s server over HTTPS*)
* In the CNAME row with Name “www” input your gh-pages website (*username.github.io*)
* At the bottom click the “ADD” button and make 3 more Type “A” rows with the IP addresses of: 185.199.109.153, 185.199.110.153, 185.199.111.153  
  (*don’t worry when you leave the page it will alphabetize the types*)





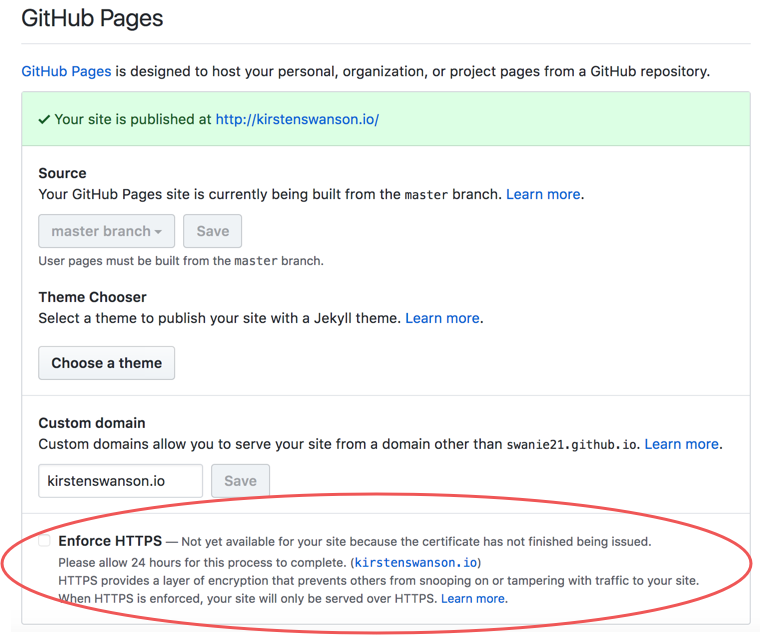
5. Go to your editor and in the repository of your website create a new file named “CNAME” in the root of your directory.

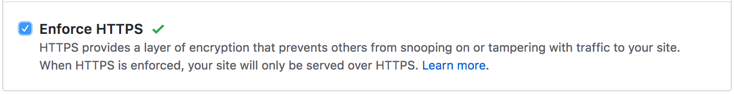


6. In the “CNAME” file add your domain name purchased from GoDaddy.

7. Add, commit, and push your changes to Github.

8. Within your repository you will need to allow your website to be served through HTTPS. Check “Enforce HTTPS” in the Settings tab of your repo. This permission might take awhile because the certificate needs to be issued, but make sure you check this enforcement option when allowed.





You can confirm that your DNS is set-up correctly by using the dig command in your terminal with your custom domain. You should see that your “A” Types point to the IP addresses that you had specified in GoDaddy’s DNS management page, in other words you’re pointing your DNS to GitHub’s server. Below is an example of the dig command:

C:\Users\Satya>cd C:\dig

C:\dig>dig www.satyacodes.com +nostats +nocomments +nocmd

;www.satyacodes.com. IN A

www.satyacodes.com. 3600 IN CNAME satyacodes.github.io.

satyacodes.github.io. 3600 IN A 185.199.110.153

satyacodes.github.io. 3600 IN A 185.199.108.153

satyacodes.github.io. 3600 IN A 185.199.109.153

satyacodes.github.io. 3600 IN A 185.199.111.153

cd C:\dig

dig satyacodes.com +noall +answer

satyacodes.com. 3478 IN A 185.199.109.153

satyacodes.com. 3478 IN A 185.199.111.153

satyacodes.com. 3478 IN A 185.199.110.153

satyacodes.com. 3478 IN A 185.199.108.153

C:\dig>

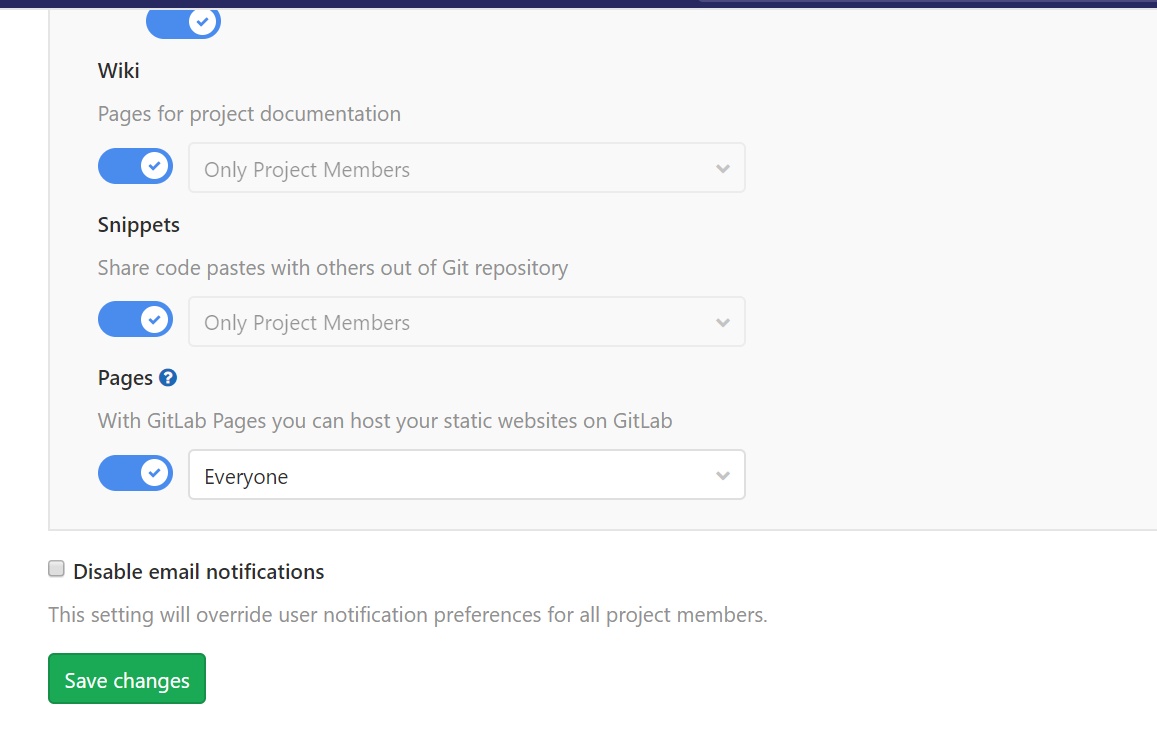
If you’re encountering any problems, this [GitHub Troubleshooting Custom Domains](https://help.github.com/articles/troubleshooting-custom-domains/) documentation is a good reference.

And just like that, you now have your very own custom DNS! As a side note, it may take some time for your DNS configuration to update as well as receiving your HTTPS certificate (allow 24 hours), but after that celebrate and share your website! 🎉

# Gitlab – Domain

Project name must be : satyacodes.gitlab.io

Navigate to your project’s **Setting > Pages** and click **+ New domain** to add your custom domain to GitLab





Root domains (example.com) require:

* A [DNS A record](https://docs.gitlab.com/ee/user/project/pages/custom_domains_ssl_tls_certification/dns_concepts.html#a-record) pointing your domain to the Pages server.

satyacodes.com CNAME satyacodes.gitlab.io.

* A [TXT record](https://docs.gitlab.com/ee/user/project/pages/custom_domains_ssl_tls_certification/dns_concepts.html#txt-record) to verify your domain’s ownership.

\_gitlab-pages-verification-code.satyacodes.com TXT gitlab-pages-verification-code=18226802e19b611120e034aca755362c



.gitlab-ci.yml

GitLab CICD

#.gitlab-ci.yml

image: ruby:2.6.5

# add bundle cache to 'vendor' for speeding up builds

cache:

paths:

- vendor/

before\_script:

- bundle install --path vendor

# the 'pages' job will deploy and build your site to the 'public' path

pages:

stage: deploy

script:

- bundle exec jekyll build -d public/

artifacts:

paths:

- public

only:

- master # this job will affect only the 'master' branch