# Sync GitHub GitLab

To set up a mirror from GitLab to GitHub, you need to follow these steps:

1. Create a [GitHub personal access token](https://help.github.com/en/articles/creating-a-personal-access-token-for-the-command-line) with the public\_repo box checked.

**Creating GitHub token**

Profile > Settings > Developer settings > Personal access tokens > Generate New > Select the scopes, or permissions(public\_repo  cheked) >Generate

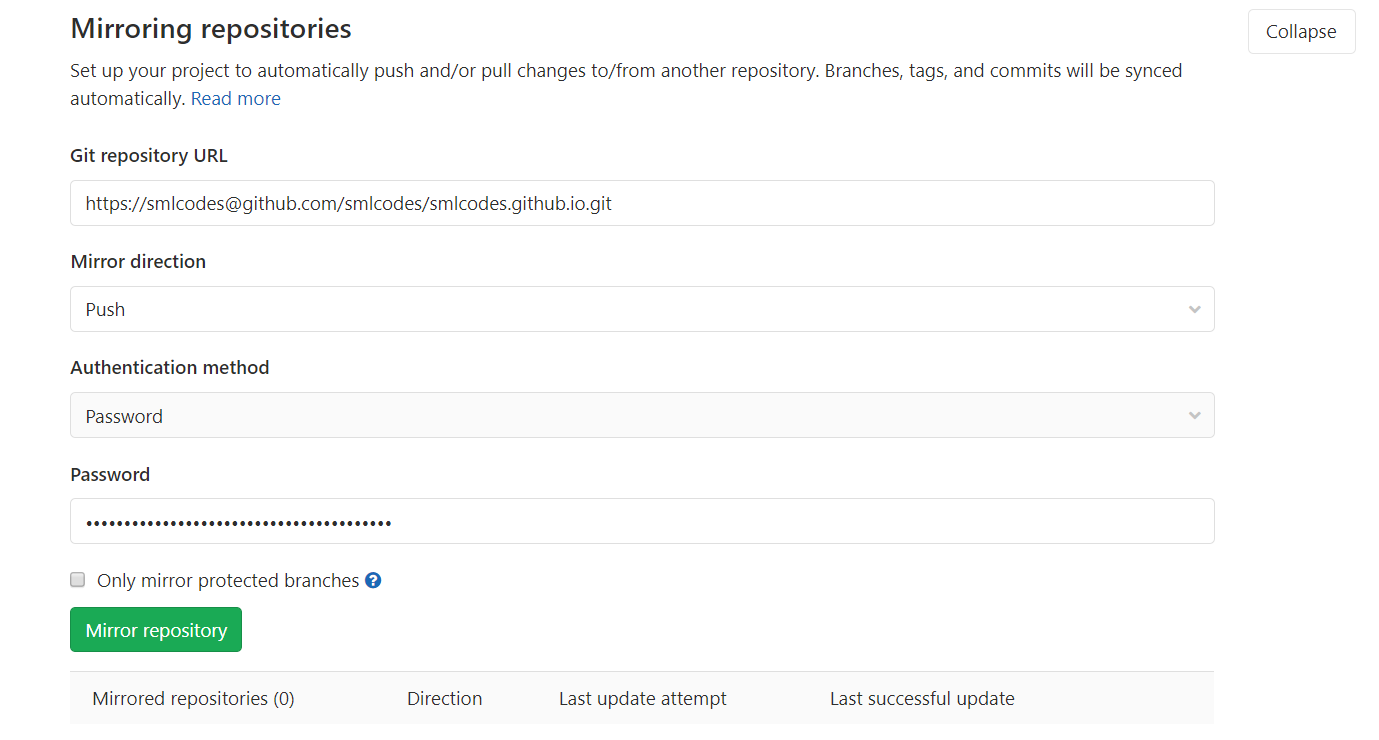
copy the token to your clipboard : \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Fill in the **Git repository URL** field using this format: [https://<your\_github\_username>@github.com/<your\_github\_group or username>/<your\_github\_project>.git](https://%3cyour_github_username%3e@github.com/%3cyour_github_group%20or%20username%3e/%3cyour_github_project%3e.git).

https://smlcodes@github.com/smlcodes/springmvc.git

Cross Check By hitting above URL in browser

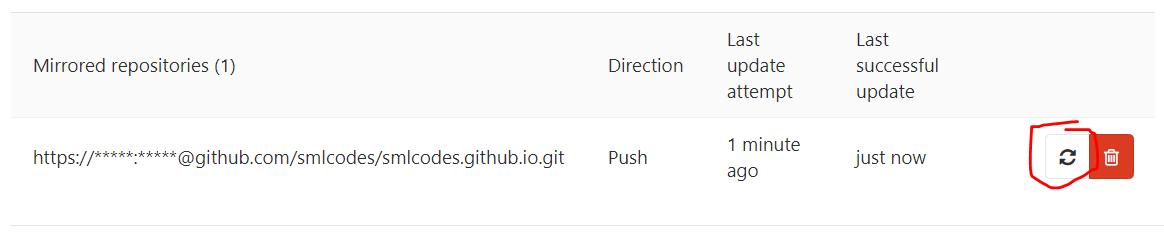
1. Fill in **Password** field with your GitHub personal access token.



1. Click the **Mirror repository** button.

The mirrored repository will be listed. For example, https://\*\*\*\*\*:\*\*\*\*\*@github.com/<your\_github\_group>/<your\_github\_project>.git.

The repository will push soon. To force a push, click the appropriate button



<https://docs.gitlab.com/ee/user/project/repository/repository_mirroring.html>

<https://docs.gitlab.com/ee/user/project/repository/repository_mirroring.html>

# GitHub - Custom Domain Configuration

## GitHub Pages Configuration

**Create Your Website**

Once you’ve [signed in](https://github.com/login), you’ll create a new repository to get started.

**Provide Repository Name**,

it must be a proper GitHub domain:

Syntax

<username>.github.io

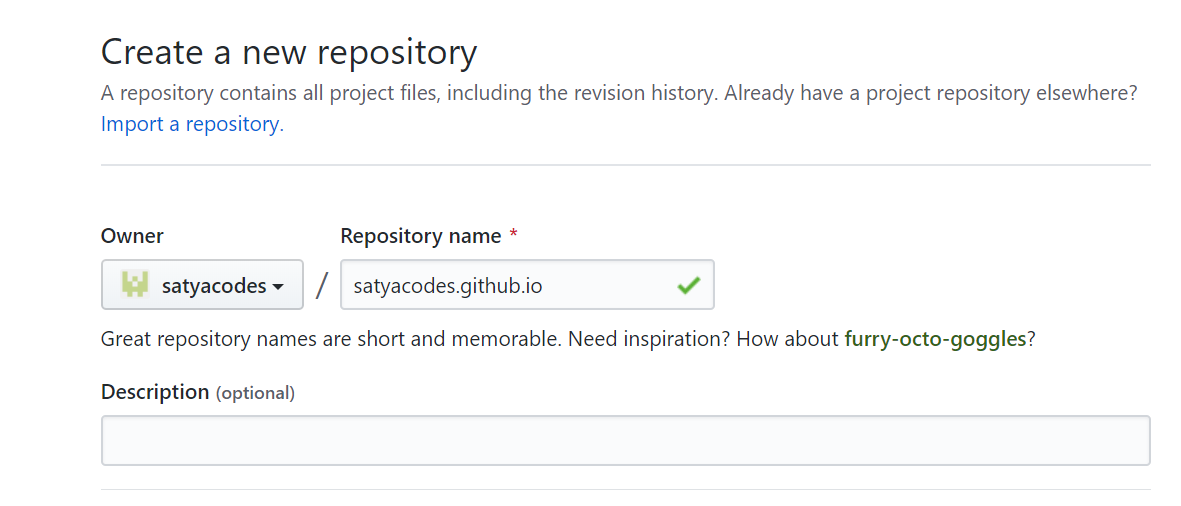
Example

satyacodes.github.io

#must be Public Repository

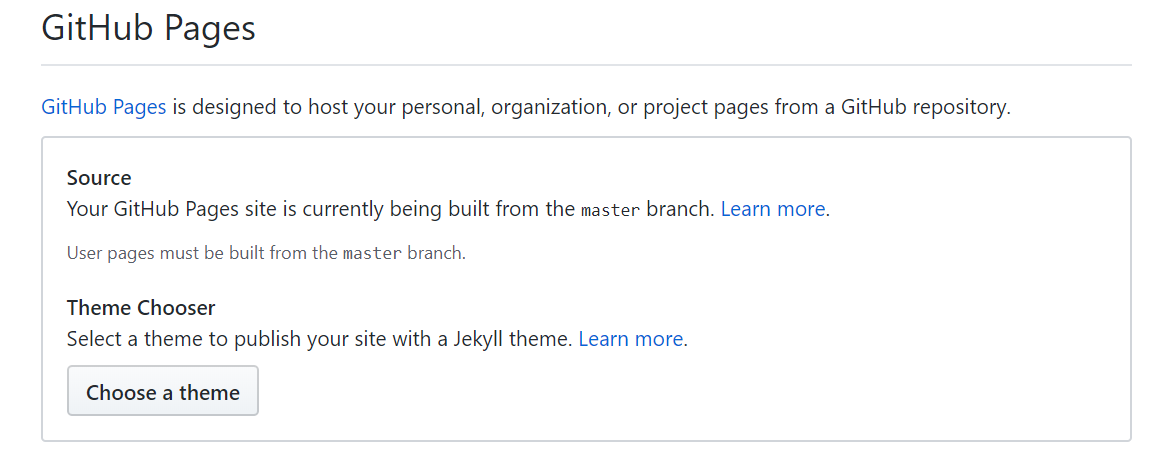
**To Change Username**

* In the upper-right corner > **click your profile photo** > then click **Settings**.
* In the left sidebar, click **Account settings**.
* In the "**Change username**" section, click Change username.



**Configure Theme**

Repository > Settings > GitHub Pages Section > Choose a theme

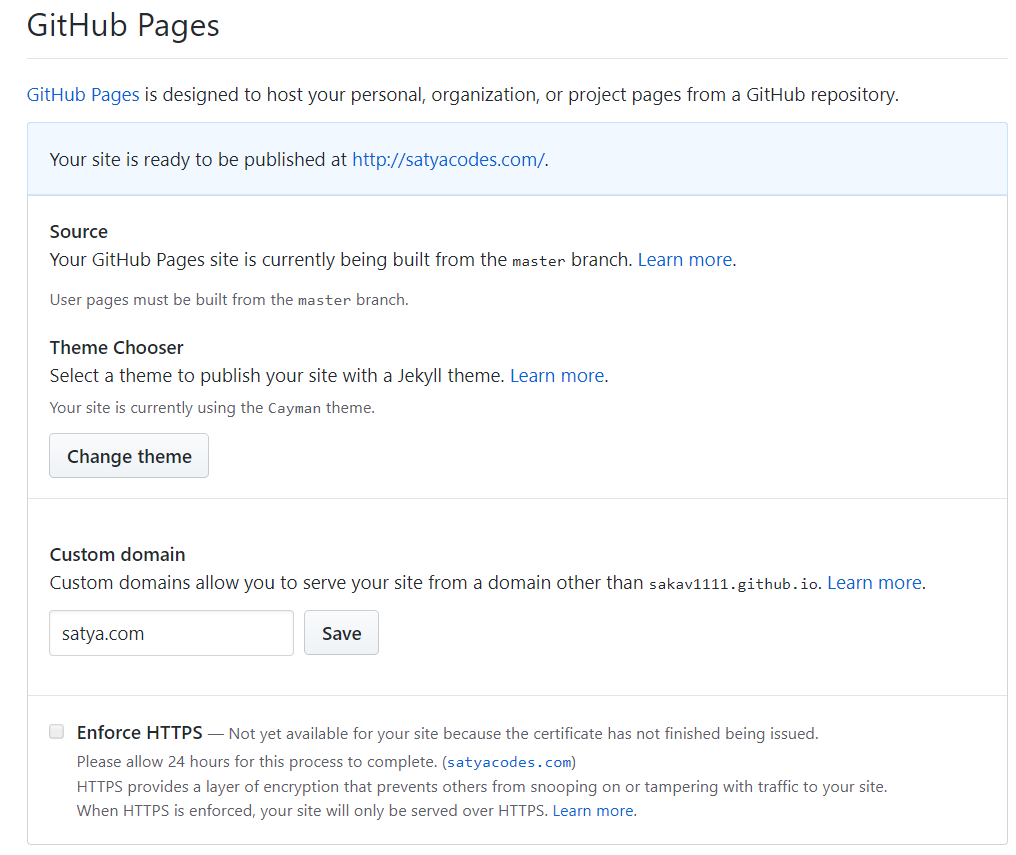


Commit Changes.

## GitHub Custom Domain Configuration

**Add custom domain**

Go to Repository > Setting > GitHub pages Section > Custom Domain : Enter Domain (Tick : Enforce HTTPS)

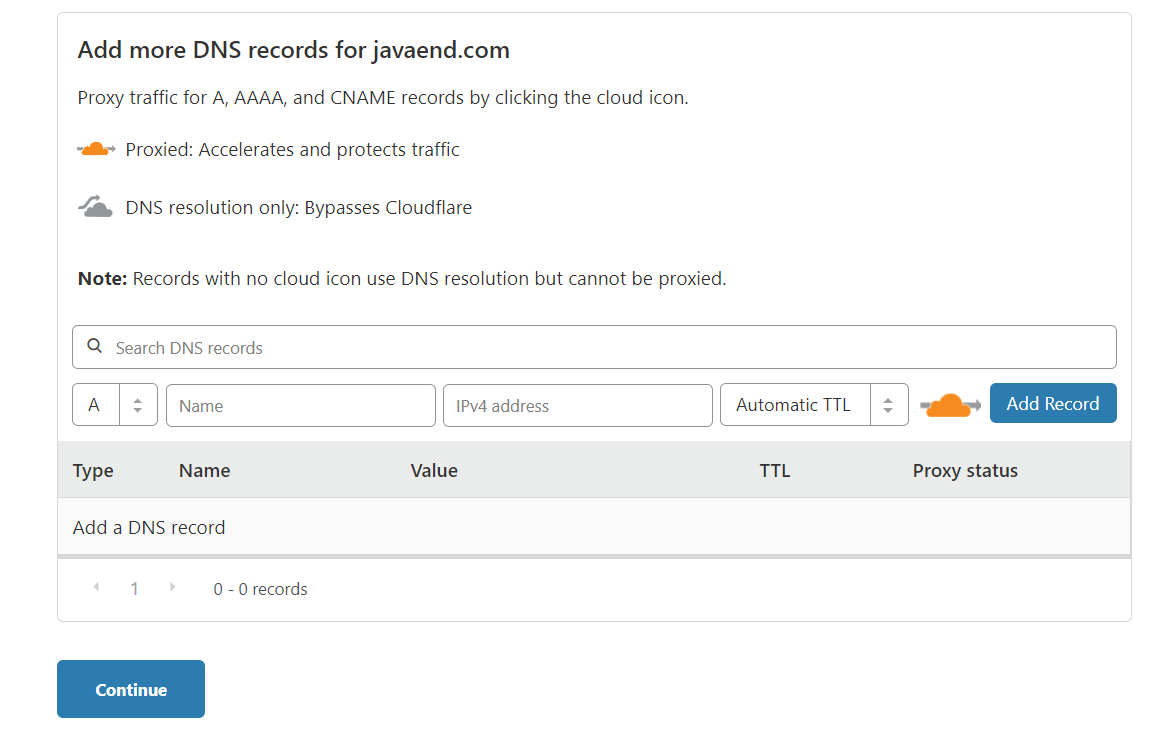


## GitHub Cloudflare Configuration

GitHub 🡪 Cloudflare 🡪 GoDaddy

### GitHub to Cloudflare

Login to Cloudflare > Add Site : satya.com > Select Plan : It will scan the DNS records.

It will Navigate to DNS Entries Page 

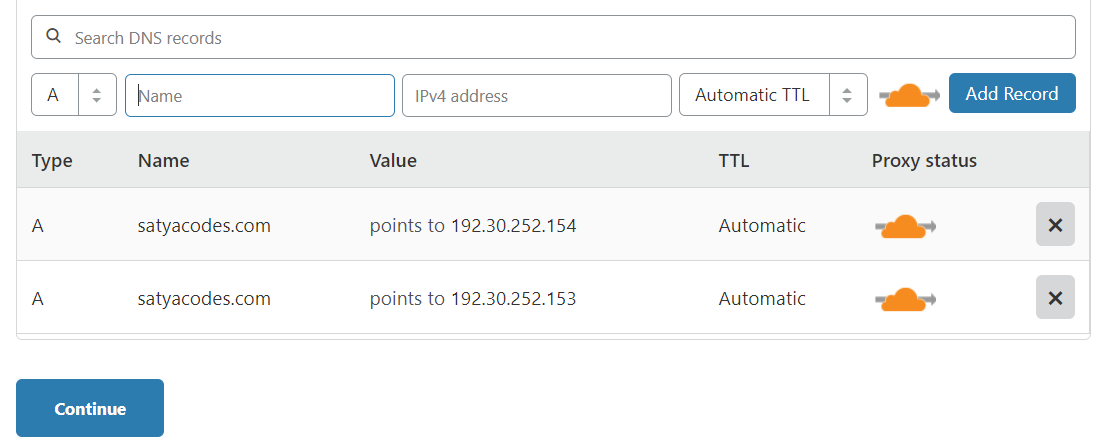
**DNS – A Records Entry**

In this step, we inform Cloudflare to point "our domain to the Github Pages server" using two A Record DNS entry:

**A** name Records : GitHub DNS Servers

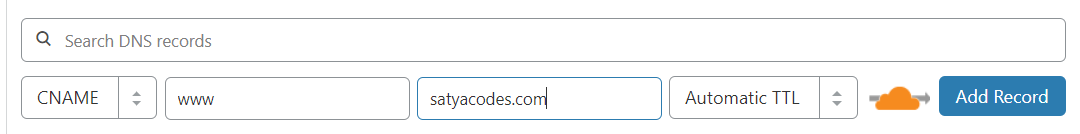
192.30.252.153

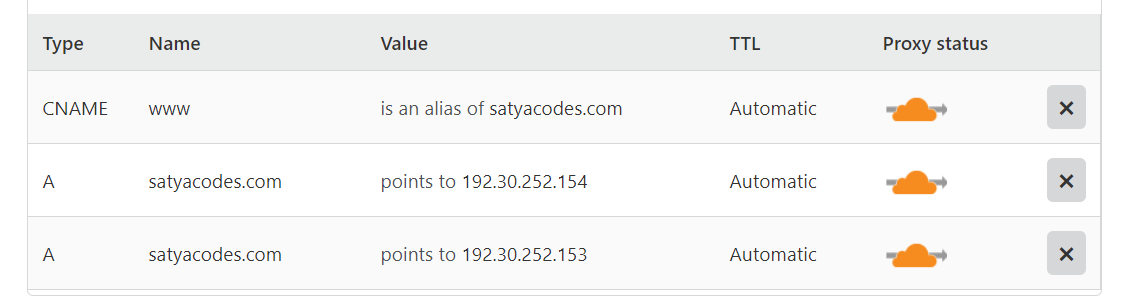
192.30.252.154



**CNAME record DNS entry:**

which will point your subdomain(www) to your apex domain(@).



Finally, DNS records for Your domain looks like

### Cloudflare to GoDaddy

**Get Cloudflare Name Servers**

To use Cloudflare, ensure your authoritative DNS servers, or nameservers have been changed. These are your assigned Cloudflare nameservers.

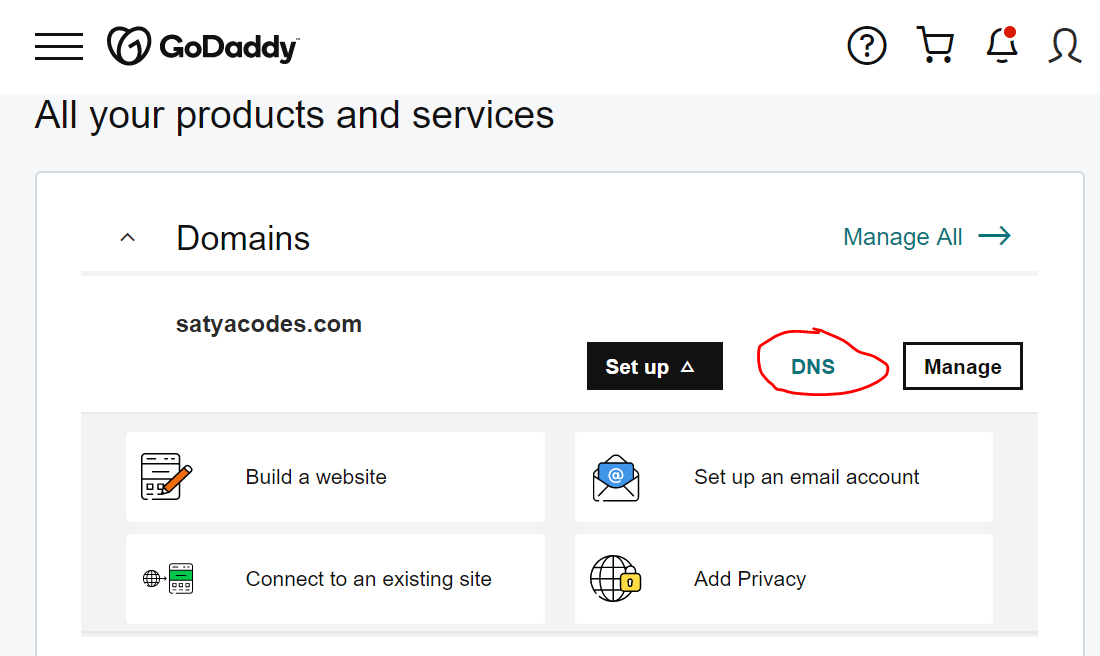
Type Value

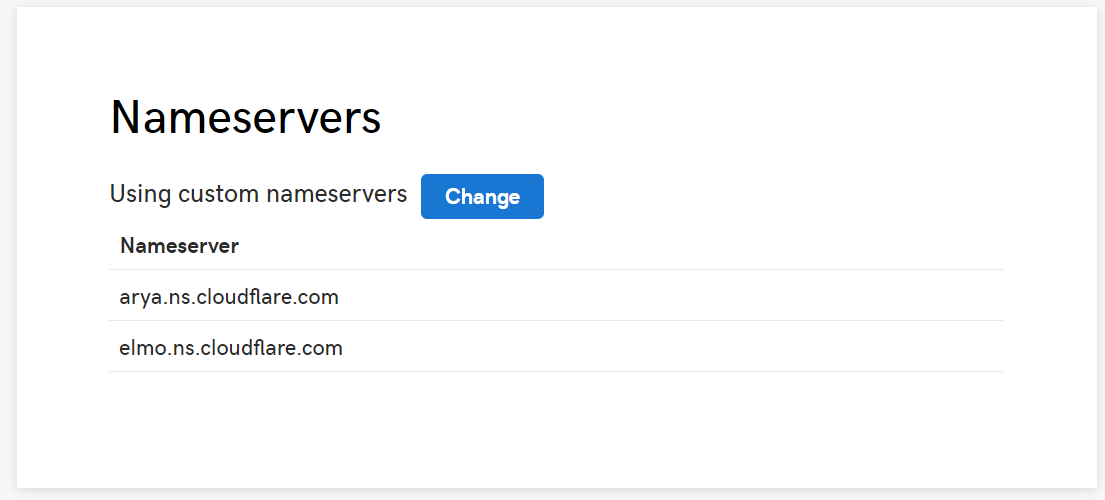
NS arya.ns.cloudflare.com

NS elmo.ns.cloudflare.com

**Update in GoDaddy Name Servers**

Login to GoDaddy > Product > Domain Tab : DNS



Update Cloudflare nameservers 

That’s it. Access Website Now

## Ref.

<https://guides.github.com/features/pages/>

<https://www.freecodecamp.org/news/an-illustrated-guide-for-setting-up-your-website-using-github-cloudflare-5a7a11ca9465/>

# GitLab – Custom Domain Configuration

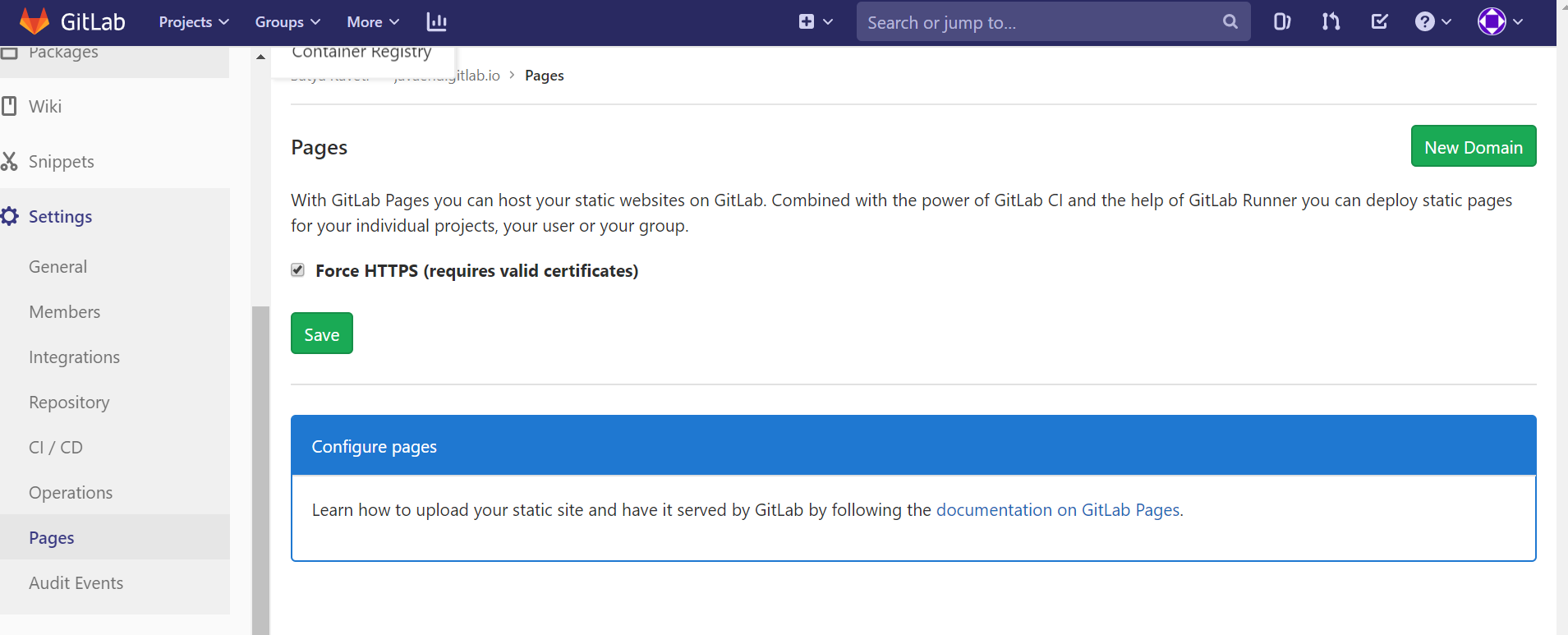
## GitLab Pages Configuration

**Create Project**

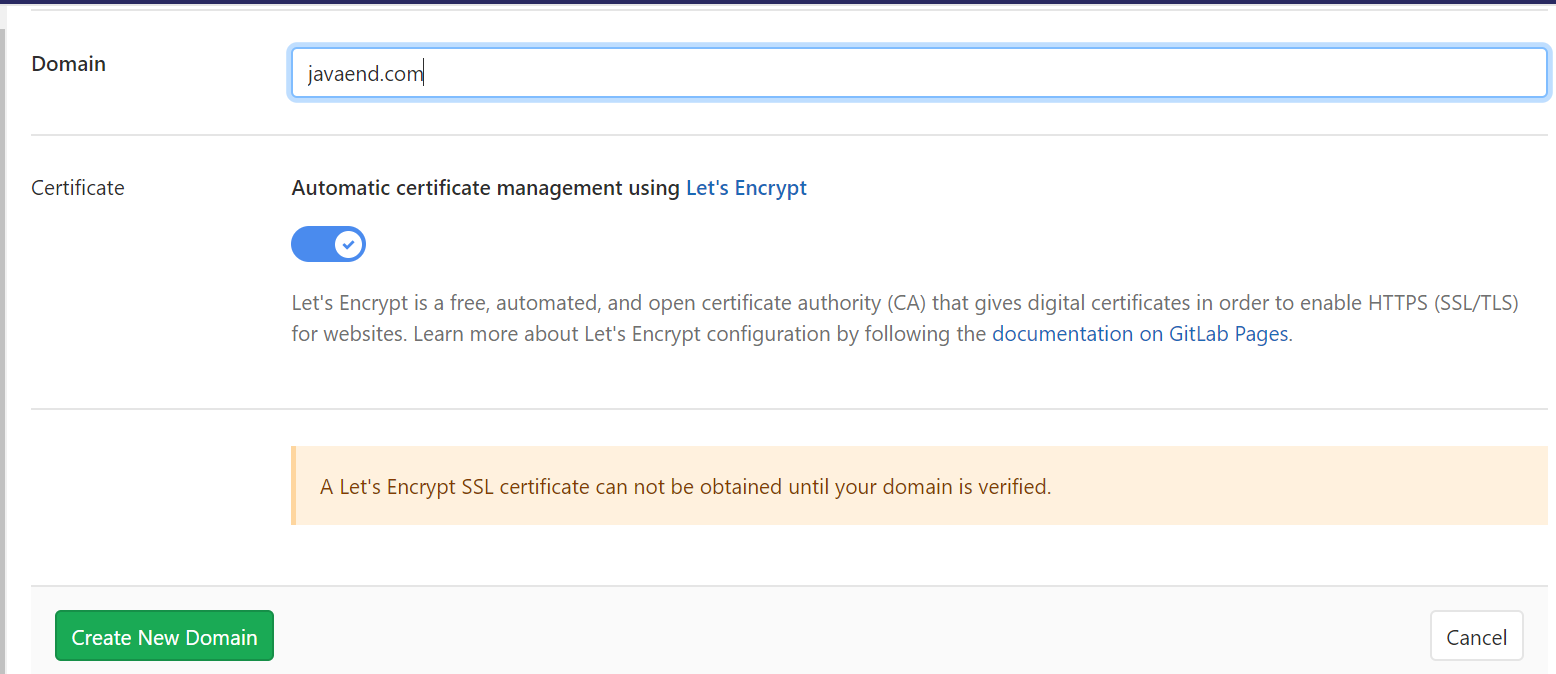
Project name must be : <username>.gitlab.io

**Create Pages & domain**

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



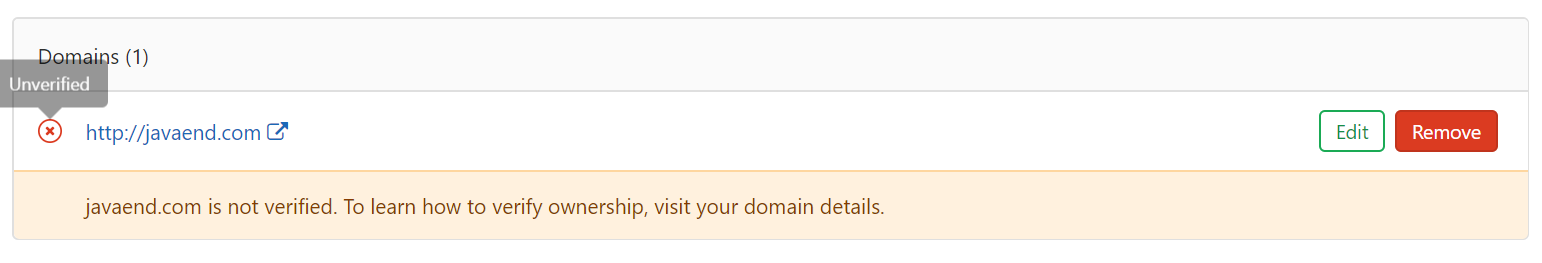


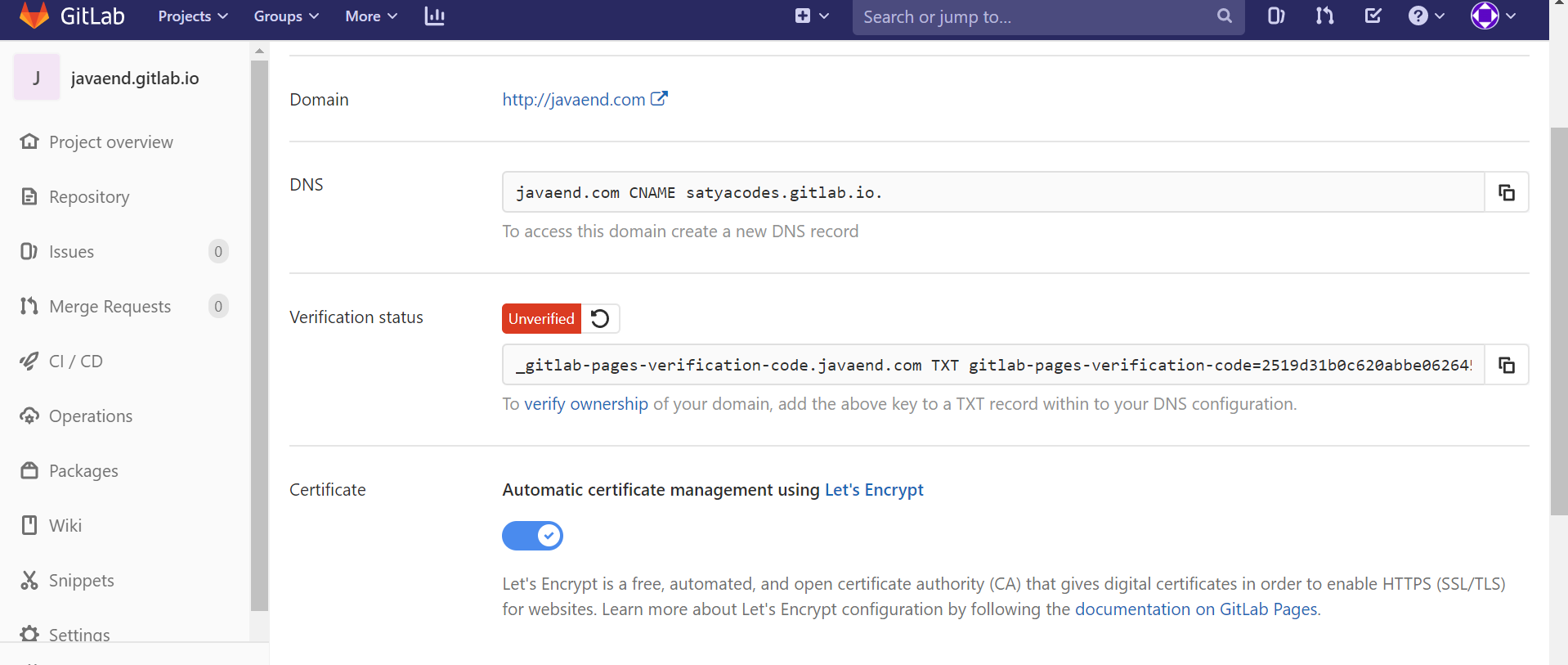


## GitLab Cloudflare Configuration

Login to Cloudflare > Add Site : satya.com > Select Plan : It will scan the DNS records. It will Navigate to DNS Entries Page

Please configure DNS Records as mentioned in GitLab. You can get those details by:

**Javaend.gitlab.io > Settings > Pages > Domains (1): EDIT** 



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



To Build Jekyll we bust configure CICD, use below code for example

#.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

