How do I set up GitHub Pages to redirect DNS requests from a subdomain (e.g. www) to the top-level domain (TLD, Apex record)?

Asked 5 years, 11 months ago Active 3 months ago Viewed 13k times

48

How do I configure a DNS service provider in such a way that requests to both `www.example.com` and `example.com` would show a website hosted on GitHub Pages? My browser's address bar should contain `example.com` when the website is opened.

My DNS service provider is `gandi.net`. It doesn't support the `ALIAS` DNS record type.

★ 39

dns github-pages

edited Jul 24 '19 at 14:57
Boann
41.9k 13 97 129

asked Apr 29 '14 at 21:15
Jay Zelenkov
3,095 2 20 29

1 Answer

Active Oldest Votes

108

Short answer

Step 1: Add a new file `CNAME` to your GitHub Pages repository containing only one line: your top-level domain name.

E.g.:

```
example.com
```

Step 2: [Optional] but highly recommended

2.1: Remove all other top-level records (prefixed with @) of type `A` from your DNS configuration.

2.2: Remove a `CNAME` record for the second-level domain `www` if it is present.

Step 3: Add these 5 entries to the very top of your DNS configuration:

```
@        A        185.199.108.153
@        A        185.199.109.153
@        A        185.199.110.153
@        A        185.199.111.153
www      CNAME    your_github_username.github.io.
```

Replace `your_github_username` with your actual GitHub username.

Step 4: Wait for your DNS changes to propagate.

DNS changes aren't effective immediately. They can take up to a full day to propagate.

Long answer

This issue has two sides. One is the DNS configuration itself. Another one is the way GitHub Pages forwards HTTP requests.

We need to know a few things to understand what GitHub is trying to say in their documentation.

DNS Entry Types

There are two types of DNS records which interest us: `CNAME` and `A`.

`A` is also known as `Apex` or sometimes as `root entry`. It forwards requests to a specified **fixed** IP address. `CNAME` entry forwards requests to a specified URL (actual valid plain text URL, not an IP address).

DNS Load balancing

GitHub has one central URL address which accepts all DNS requests for GitHub Pages: `http://username.github.io`. That URL is resolved to different IP addresses based on your geographical location. Website hosted on GitHub Pages is a simple collection of `HTML`, `CSS` and `JS` files. GitHub distributes these files to different servers across the globe. So that when your browser sends a request from Europe it receives data from a server in Europe. The same is valid for the requests from Asia and the USA.

What GitHub is trying to say

Since `A` records in DNS must contain IP addresses, and they must be either `185.199.108.153` or `185.199.109.153` or `185.199.110.153` or `185.199.111.153`, there is no way to forward requests to a server located somewhere in Europe or Asia. Your website hosted at GitHub Pages will be downloaded from a central GitHub Pages server. There is a minor risk that if GitHub Pages DNS servers (`x.x.x.153`) are down for some reason, all custom domains which use fixed GitHub Pages IP addresses will not be accessible (their DNS requests will not be resolvable).

That is why GitHub strongly suggests to either use a second-level domain for your GitHub Pages (e.g. `blog.example.com`) or use a DNS service provider that supports a record type `ALIAS` that acts as `A` record but forwards request to a URL address (e.g. `username.github.io`) instead of a fixed IP address.

How GitHub Pages treats HTTP requests

After a DNS request for `your_github_username.github.io.` is resolved into an IP address, e.g. `185.199.108.153` your browser sends an HTTP request to that server with an HTTP header `Host`. Below are `curl` examples that load the same website *(these examples might not work if you are behind a proxy server)*:

```
$> curl --header "Host: your_github_username.github.io" http://185.199.108.153/
$> curl --header "Host: www.example.com" http://185.199.108.153/
$> curl --header "Host: example.com" http://185.199.108.153/
```

This way GitHub Pages servers know which user website to serve.

> GitHub Pages server will automatically redirect HTTP request to the top-level domain if your `CNAME` file contains `example.com` but `www.example.com` is requested.
>
> The same is valid if your `CNAME` file contains `www.example.com` but the header `Host` in the HTTP request contains `example.com` .

Why can't I add a `CNAME` record entry that accepts a top-level request (`@`) to my DNS configuration?

Quote from the GitHub Pages documentation:

> **Warning**: Do not create a CNAME record for your custom apex domain! Doing so may cause issues with other services, such as email, on that domain.

References:

[Setting up a custom domain with GitHub Pages](#)

[My custom domain isn't working](#)

[Cannot access my GitHub Pages website by IP Address](#)

edited Dec 11 '19 at 0:04
MarcG
17.6k 13 65 73

answered Apr 29 '14 at 21:15
Jay Zelenkov
3,095 2 20 29

7 Best answer I've seen on SO in a long time. Great reference point for GH Pages DNS stuff. I might even go update our help article with some bits from this. :-) – Joel Glovier Apr 30 '14 at 18:33

@JoelGlovier Thanks Joel! I think updating GH Pages documentation is a great idea! Most people read official docs first. – Jay Zelenkov May 2 '14 at 17:55

1 A quick note (happened to me): a CNAME file in your repo is necessary in all cases (CNAME record for subdomain or A record for apex, ALIAS record). I had to set an A record for an apex domain so I skipped the CNAME file part. I knew I was missing something :) – Marco Ferrari Jul 27 '15 at 8:59

1 @JayZelenkov, Thanks and this is exactly what I was looking for! Out of curiosity, do you know how we can setup the site to use HTTPS instead of HTTP? – ayjay May 10 '16 at 17:00

1 **Caution!** The IP addresses in this post do not support HTTPS as they aren't pointing to GitHubs CDN. The correct IP addresses as of now are 185.199.108.153, 185.199.109.153, 185.199.110.153, 185.199.111.153 (see [GitHub Help on troubleshooting Custom Domains](#) for more information). – biolauri Mar 27 '19 at 13:06