Draft Copy

Author: Gordon Stevens

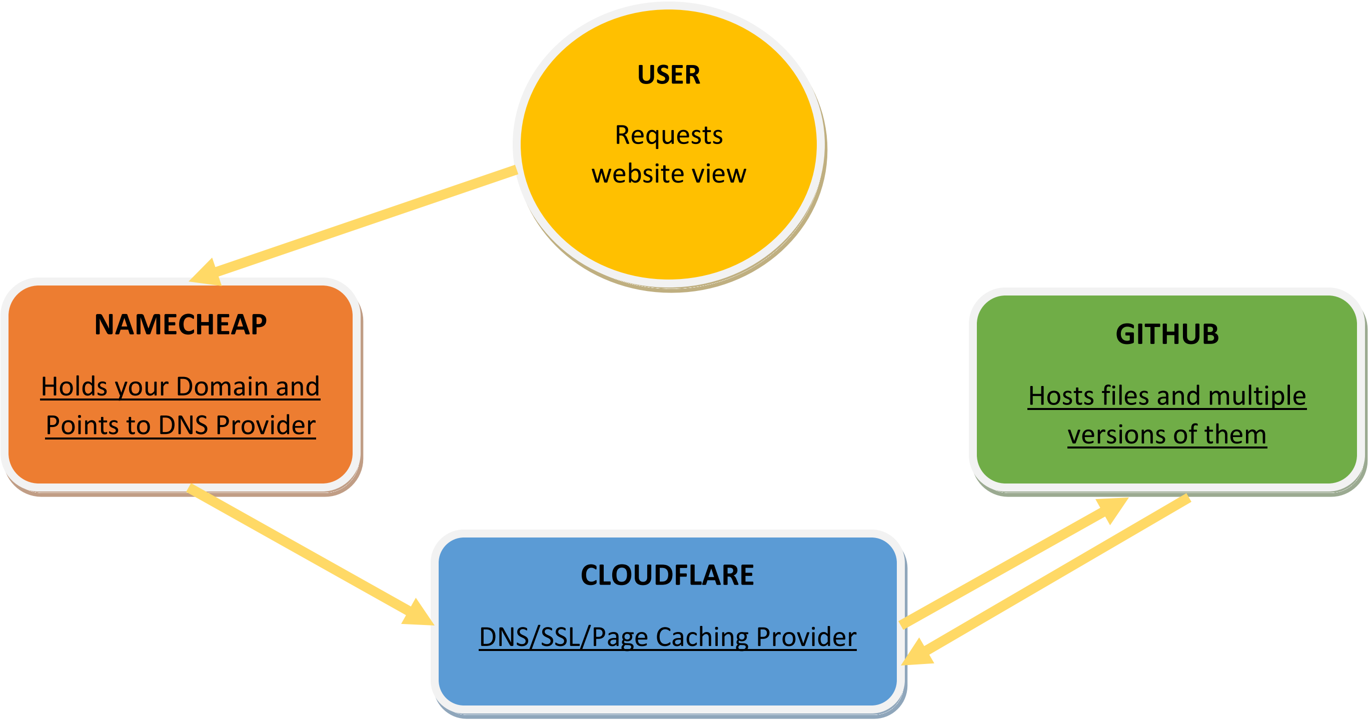
White Paper Title: Setup Namecheap Domain Registrar Service with Cloudflare DNS and Secure Socket Layer (SSL) Certifcate, hosted files on Github File Versioning and Repository Service.

Executive summary

To enable an ultra-low cost, reliable hosting solution for a website which does not require any server processing, the purchase and annual renewal of a domain name is the only cost incurred by the procedure discussed here within. The top-level overview of the system is that a domain name of choice is registered with domain name registrar namecheap.com, pointed to domain name service (DNS) provider Cloudflare, who also offer extra benefits including CNAME flattening to enable domain resolution to another host without interrupting the user experience, and Github Pages offering which provides static, non-dynamic file hosting and file versioning for free. All three services used here are enterprise grade, and if dynamic hosting is required, another host may be used, although perhaps at additional cost.

Technical Overview

Before the steps are outlined, it is important to understand some of the basics of what is going to be set up. The final product will look like this:


USER
Requests website view
NAMECHEAP
Holds your Domain and Points to DNS Provider
GITHUB
Hosts files and multiple versions of them
CLOUDFLARE
DNS/SSL/Page Caching Provider

Domain Name Service (DNS) can be a complex topic, but one of the main uses of DNS is to effectively provide a roadmap for data between a user and a server. It turns www.google.ca into an Internet Protocol (IP) address such as 66.87.44.21 and other routing and protocol techniques.

Secure Socket Layer (SSL) Certificate is issued by a Certificate Authority (CA) which is a piece of the code in the cryptographic framework of a website.

Cloudflare is a company which specializes in DNS security and routing. They offer a lot for their free user tier, including SSL. Using SSL gives your website a professional image and could be used for e-commerce if we used a webhost other than Github. In addition, Cloudflare also provides web page origin hiding, so when a user requests your website, the DNS that the user sees is Cloudflare's DNS instead of the webhost DNS. Another important feature is Cloudflare's ability to cache (keep a copy) of the website in memory so that if Github Pages becomes temporarily unavailable, Cloudflare automatically fetch the website from the cached content. This is also excellent for static websites because there is no server-side processing required. However this mechanism will break on dynamic websites such as WordPress which require user interaction at many stages.

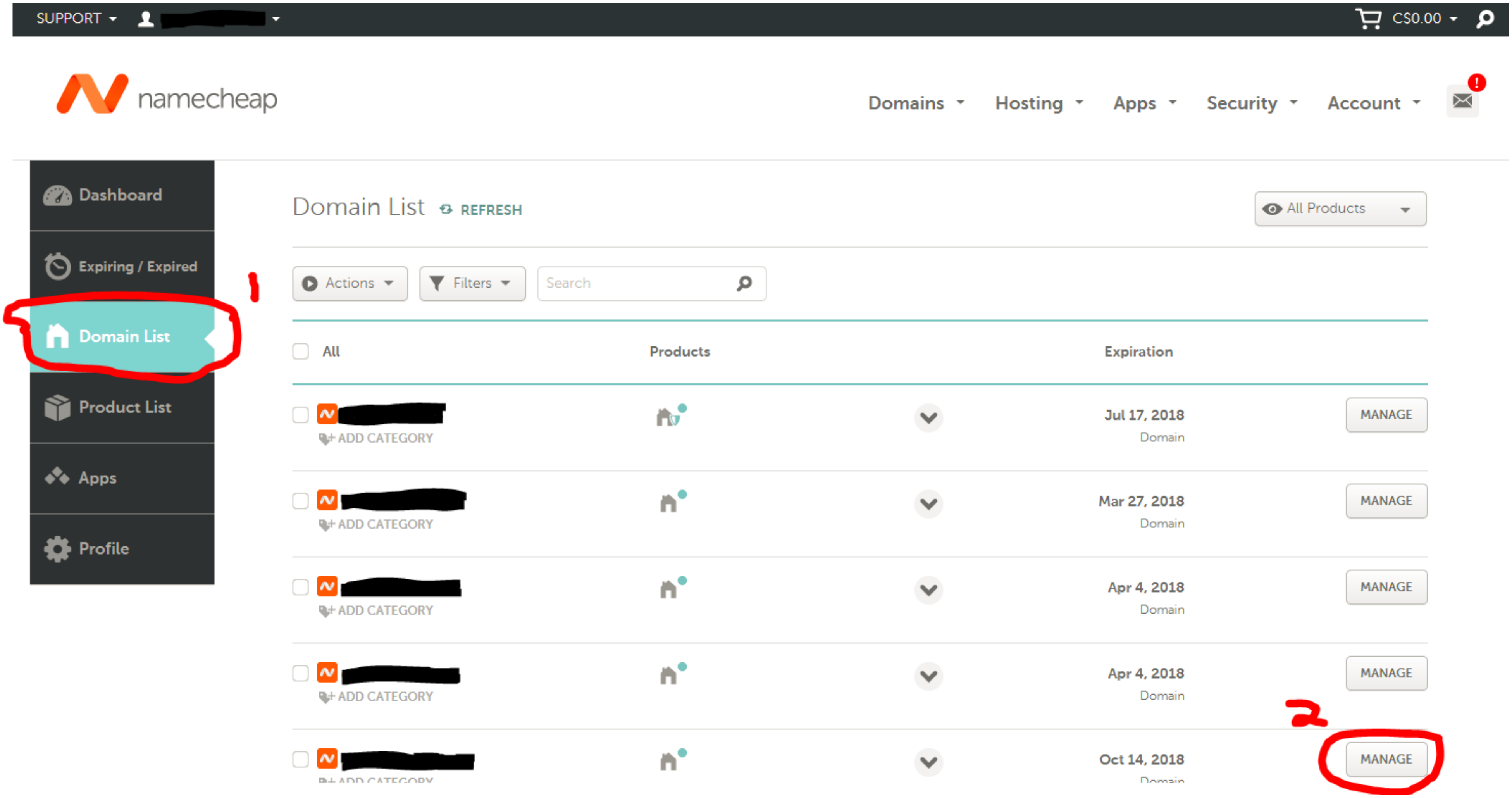
A user types your website address, such as www.traveltochina.ca, into their browser. The DNS provided by the users Internet Service Provider (ISP) then finds that Namecheap is the registrar to check for DNS. DNS is delegated to Cloudflare. Cloudflare is always hiding your DNS and fetches any pages from Github if Cloudflare's cache has expired. If the cache is disabled, Cloudflare still hides your DNS and can interpret subdomains as if they were root domains. Github is a file repository system which holds your files, and keeps all revisions of the files that are made. In this scenario, a public repository (free) is created with the username chosen to signup with Github, for example for the username shenzhen, create a public repository with the name of shenzhen.github.io and it will appear at that address.

In the earlier days of the Internet when there were fewer domain names registered, it was simple to assign a domain to an IP address and that was it, firstworldmemes.com would point to 66.78.43.213. This method uses DNS records called A records. As the Internet gained popularity and IP addresses were not as plentiful, services started to use CNAMEs. Canonical name DNS records are an alias, and are common-place in modern networking. CNAME's allow one IP address to host many domains, for example: firstworldmemes.com, migrationology.com, eslcafe.com, interpals.net and chengdu-foodie.com could all contain CNAME records which point to 66.78.43.213. Later, a DNS standard was created to allow CNAME flattening, which means that an alias can point to an alias, and while out of the scope of this whitepaper has several excellent use-cases, such as quickly delegating customers' websites. CNAME flattening allows our Cloudflare alias to point to Github Pages alias to finally find an appropriate IP address.

In summary, this setup uses Namecheap as a registrar which delegates DNS management to Cloudflare, which provides an SSL certificate (to enable https://), DNS hiding, page caching and filtering by country or origin. The CNAME flattening technique allows the system to transparently alias the DNS to Github Pages service where the website files originate from. Added functionality from Github includes file version system and reliable hosting. All three services have redundant enterprise-grade systems, financially backed by major companies with invested interests in each of these companies, so daisy-chaining the system holds little problems once setup has taken place.

Step-by-step guide

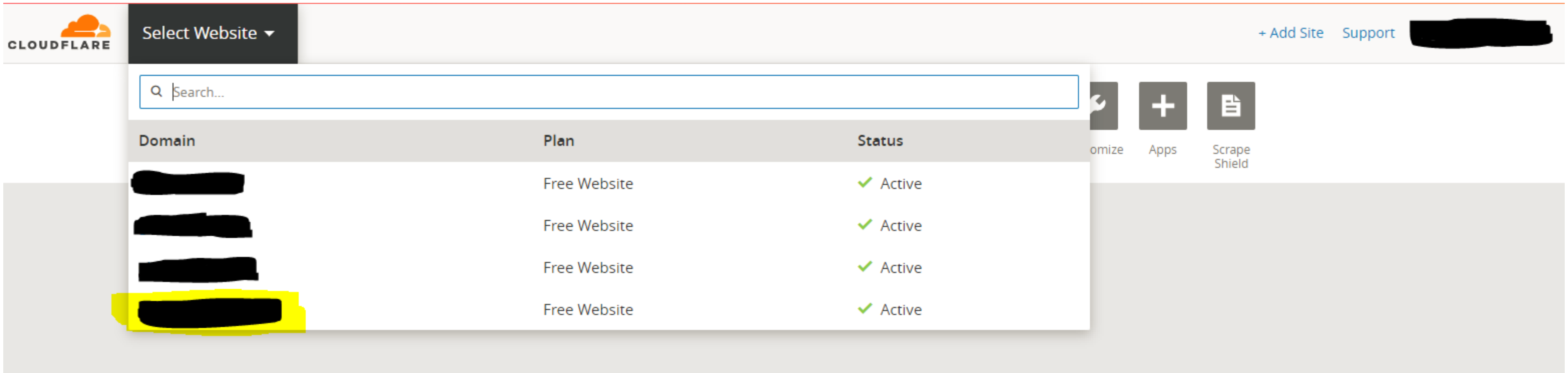
First you need to register a domain name of your choosing at namecheap.com, or log-in to your existing Namecheap account and buy a new domain or change the DNS on an existing domain in your account.



□ All	Products		Expiration	
□ [redacted] ADD CATEGORY			Jul 17, 2018 Domain	MANAGE
□ [redacted] ADD CATEGORY			Mar 27, 2018 Domain	MANAGE
□ [redacted] ADD CATEGORY			Apr 4, 2018 Domain	MANAGE
□ [redacted] ADD CATEGORY			Apr 4, 2018 Domain	MANAGE
□ [redacted] ADD CATEGORY			Oct 14, 2018 Domain	MANAGE

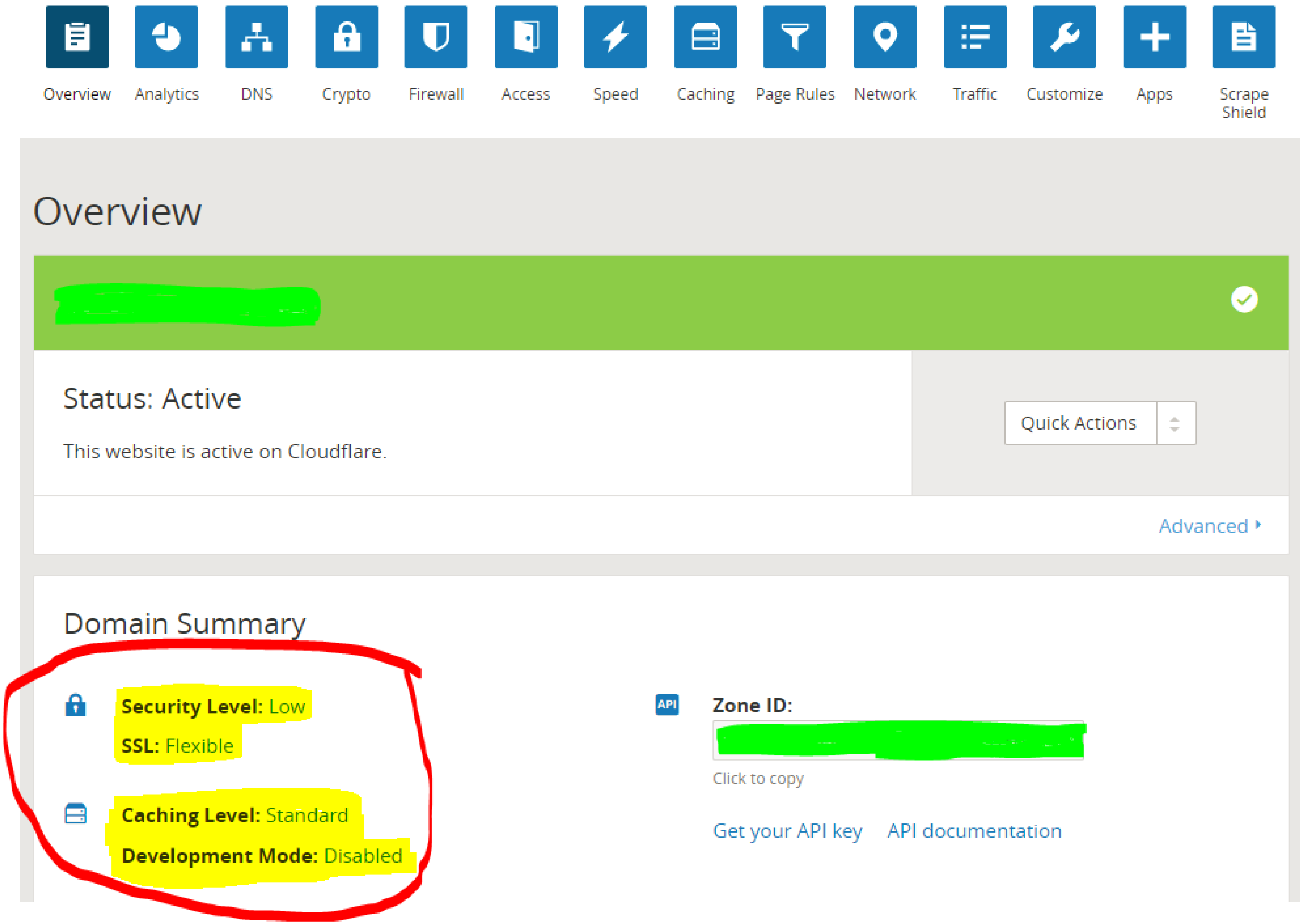
Leave this window open and to the side, because you will have to come back to this

Next, open a new browser tab and go to cloudflare.com. Signup for their free account, and when asked for domain, enter the domain name you have just purchased. At this point Cloudflare may advise you to update your domain name servers at namecheap to point to Cloudflare's DNS servers. This is what we want, but if it is not apparent, this is how you find them. Select your domain as shown below:



Domain	Plan	Status
[redacted]	Free Website	✓ Active
[redacted]	Free Website	✓ Active
[redacted]	Free Website	✓ Active
[redacted]	Free Website	✓ Active

Once you click your domain please note the row of blue navigation icons along the navigation.



Overview

[redacted]

Status: Active

This website is active on Cloudflare.

Domain Summary

Security Level: Low

SSL: Flexible

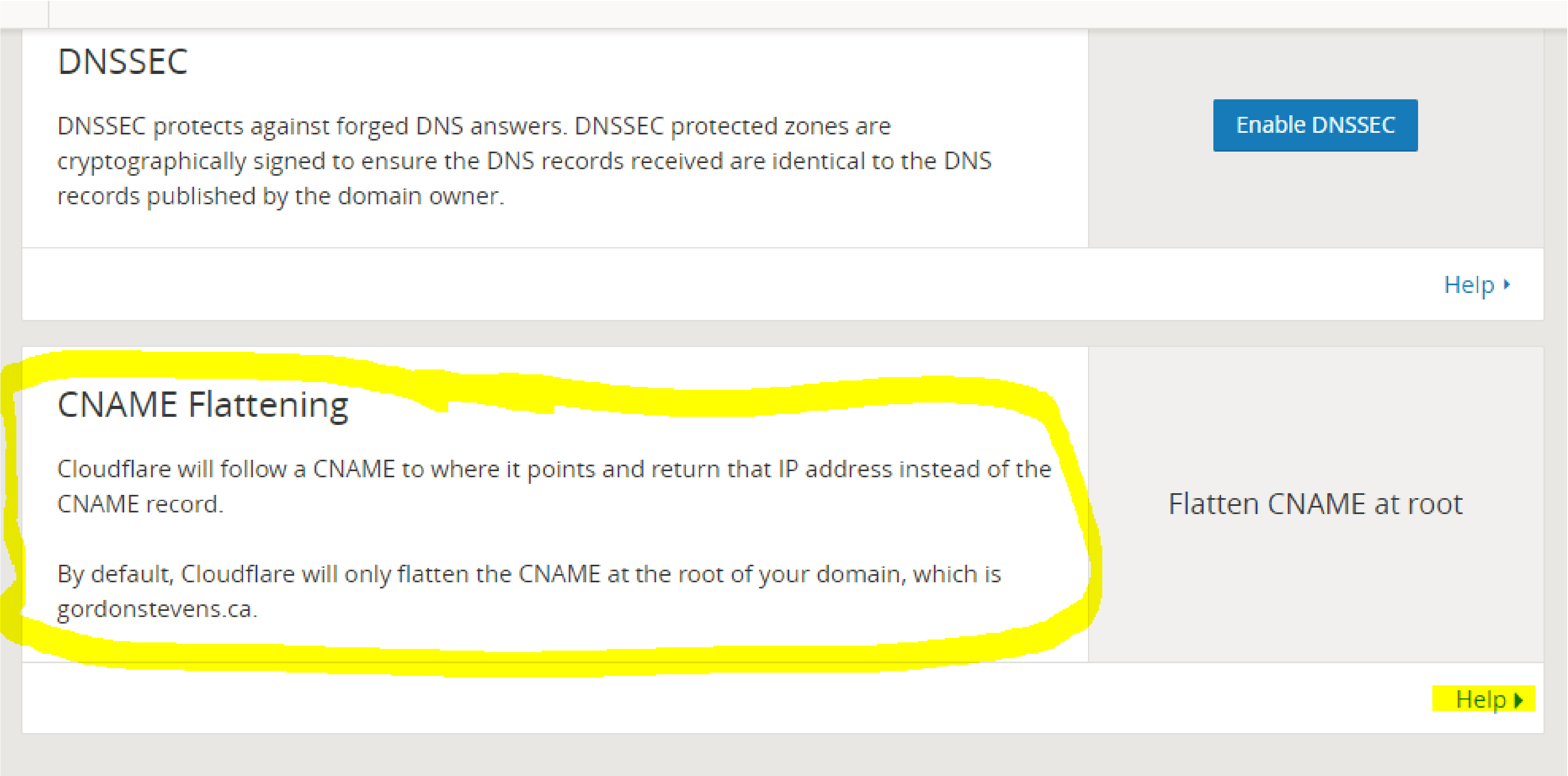
Caching Level: Standard

Development Mode: Disabled

Zone ID: [redacted]

Click to copy

As the blue navigation tabs are clicked from left to right, you will see the following information:



DNSSEC

DNSSEC protects against forged DNS answers. DNSSEC protected zones are cryptographically signed to ensure the DNS records received are identical to the DNS records published by the domain owner.

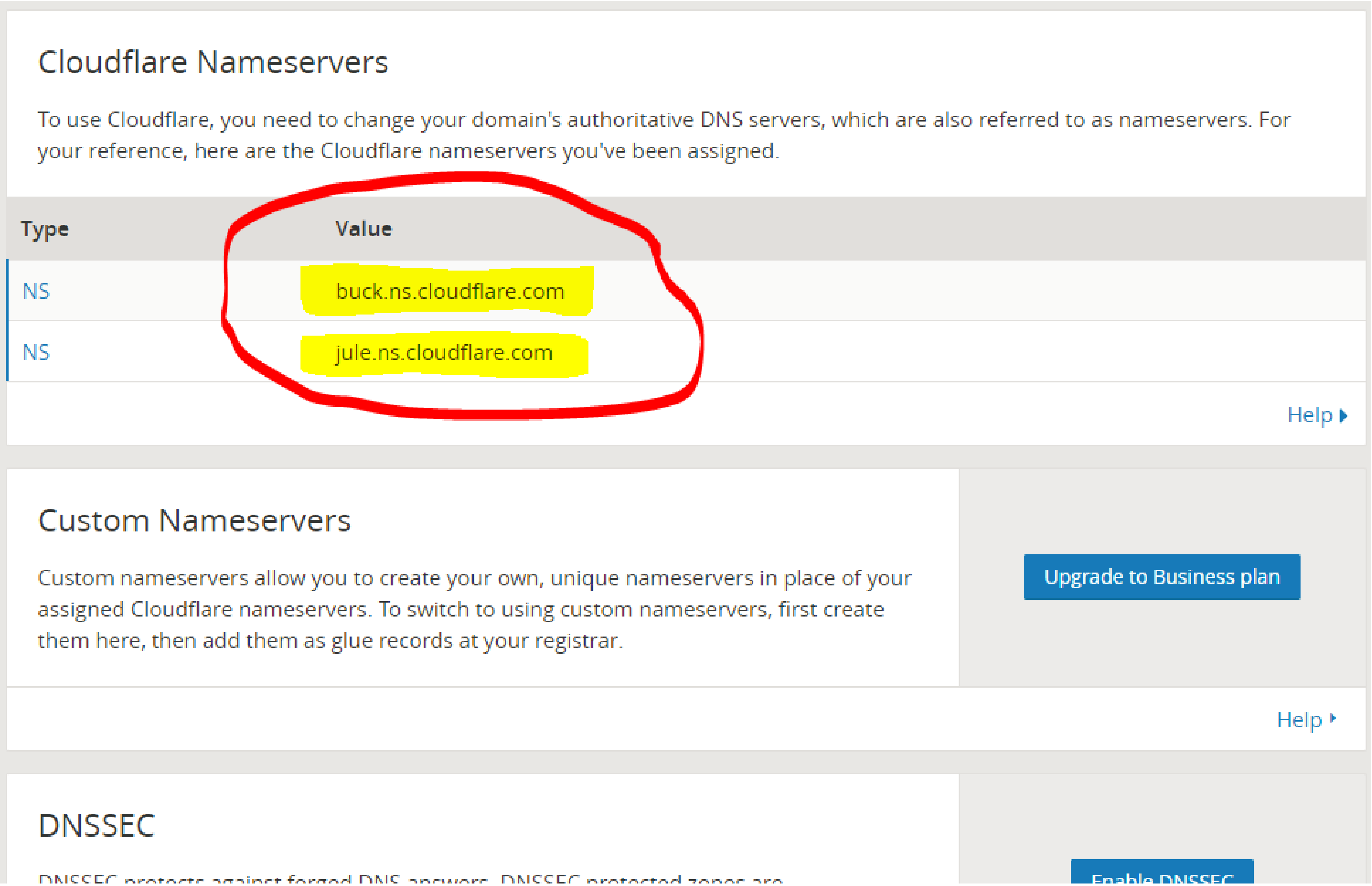
CNAME Flattening

Cloudflare will follow a CNAME to where it points and return that IP address instead of the CNAME record.

By default, Cloudflare will only flatten the CNAME at the root of your domain, which is gordonstevens.ca.

Flatten CNAME at root

Now we find the Cloudflare nameservers listed here:



Cloudflare Nameservers

To use Cloudflare, you need to change your domain's authoritative DNS servers, which are also referred to as nameservers. For your reference, here are the Cloudflare nameservers you've been assigned.

Type	Value
NS	buck.ns.cloudflare.com
NS	jule.ns.cloudflare.com

Custom Nameservers

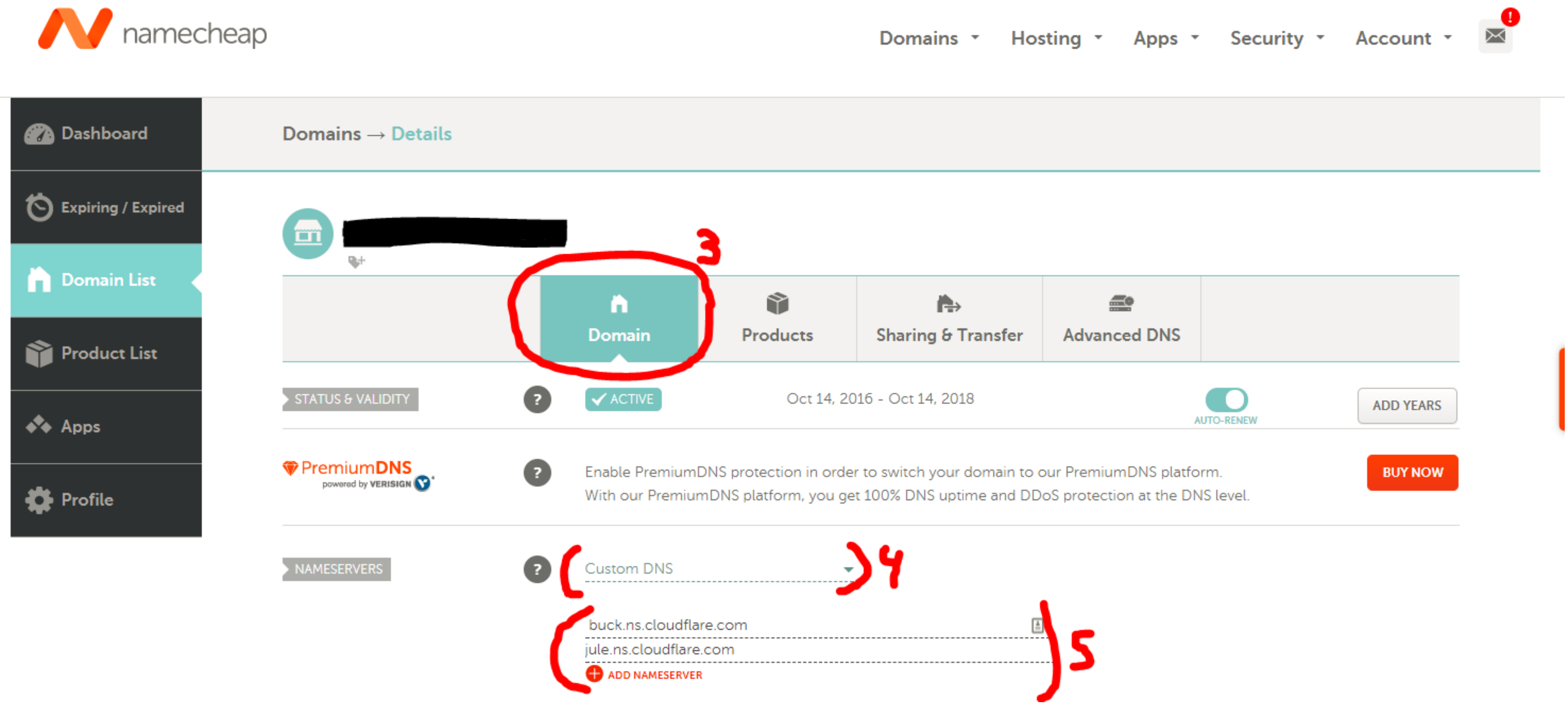
Custom nameservers allow you to create your own, unique nameservers in place of your assigned Cloudflare nameservers. To switch to using custom nameservers, first create them here, then add them as glue records at your registrar.

DNSSEC

DNSSEC protects against forged DNS answers. DNSSEC protected zones are

Copy whatever is displayed here in your account into notepad (Note: they may differ from buck and jule, always use what is supplied!)

Back to Namecheap, open up your DNS settings on your domain. Select left *Domain List* tab, select *Domains* tab (labelled 3 in the diagram), under *Name Servers* area, select *Custom DNS*, and add the Cloudflare nameservers from notepad. These absolutely, MUST be correct, or this setup will fail with non-obvious troubleshooting required. This change may be anywhere from an instant change to several days, which is unavoidable. After you enter the settings, please continue on by opening up Cloudflare and selecting your domain again.



Domains → Details

[redacted]

STATUS & VALIDITY: ✓ ACTIVE — Oct 14, 2016 - Oct 14, 2018 — AUTO-RENEW

PremiumDNS powered by VERISIGN: Enable PremiumDNS protection in order to switch your domain to our PremiumDNS platform. With our PremiumDNS platform, you get 100% DNS uptime and DDoS protection at the DNS level.

NAMESERVERS: Custom DNS

buck.ns.cloudflare.com

jule.ns.cloudflare.com

As before, click through the blue tabs to familiarize yourself a bit with Cloudflare and find the main DNS settings for your domain. This is where all of your domain's DNS settings are going to reside, if you use Google Hosted Apps for your domain name or Microsoft's Office 365 offerings, you can now add them here, as directed by Google or Microsoft's websites. This may involve adding Mail eXchanger (MX) records which direct your E-mail properly, and SRV records for Google Talk support. Also CNAME's for Google Calendar, Google Drive, and so forth. Those are all safe to add here, and now is an excellent time to do so.

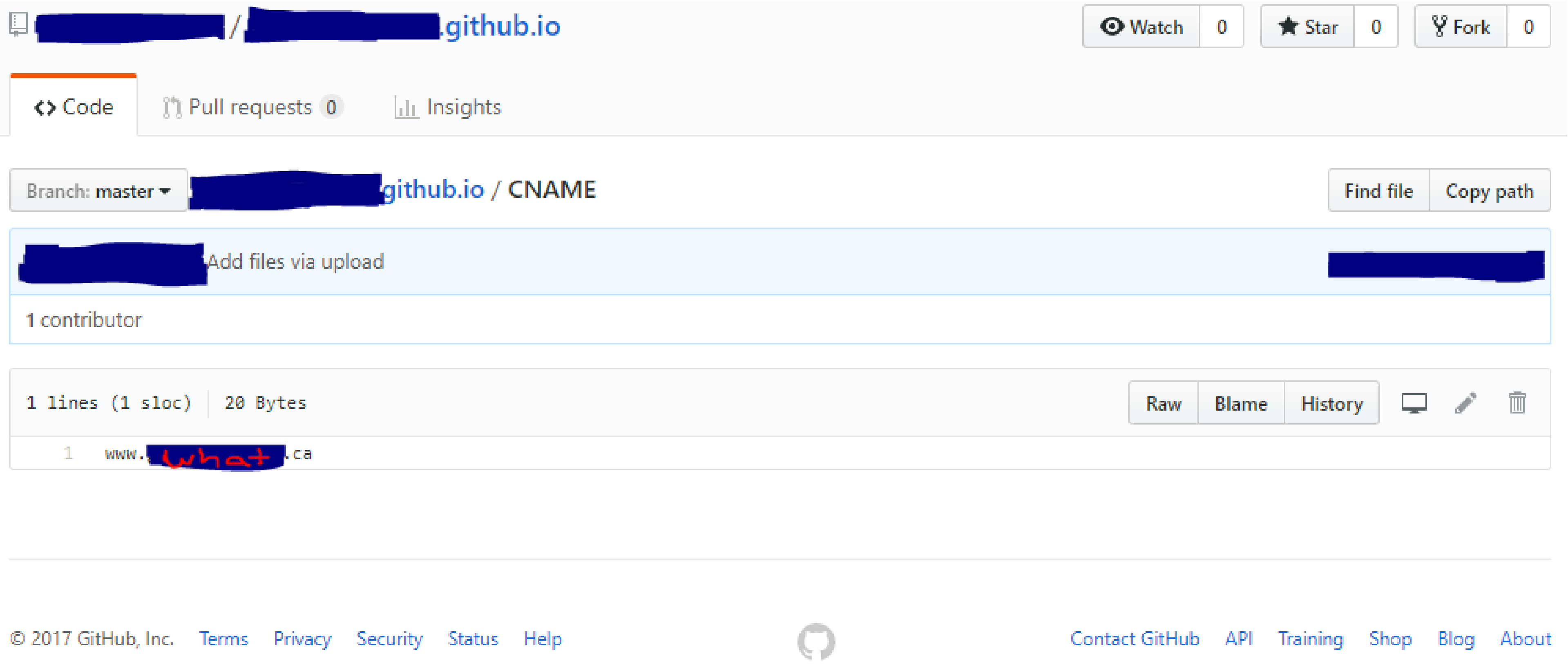
After adding any third party requirements mentioned above, its time to signup for a Github account, and carefully select a username that suits your domain name. Next create a public repository called *yourusername.github.io* where *yourusername* is the Github username. Git can be complicated, but for our usage, just some quick tutorials on the Github website or YouTube videos such as those produced by *Derek Banas* should be sufficient. Once Github and the appropriately named public repository has been created, post two files to the master branch, index.html and CNAME

For index.html, just create a new html with:

```
<html><body>This will work! Test test test! This should appear on the domain name!</body></html>
```

For CNAME, just add the following on Line 1 ONLY! Change www.what.ca to the domain name purchased earlier:

www.what.ca



[redacted] / [redacted].github.io

[redacted]github.io / CNAME

[redacted] Add files via upload [redacted]

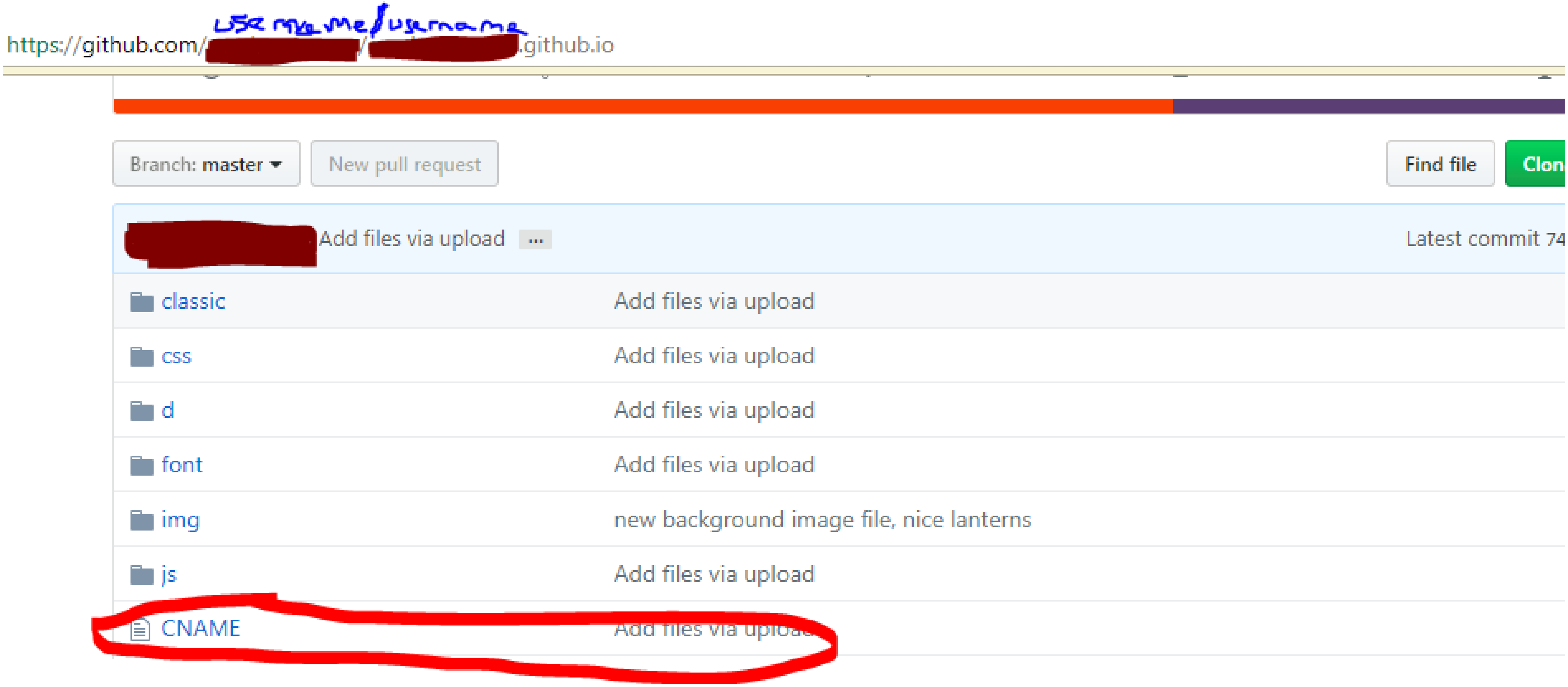
1 contributor

```
1 lines (1 sloc)  20 Bytes
```

```
www.[redacted]what.ca
```

Now COMMIT and PUSH both files to the public repository named *yourusername.github.io* in the MASTER branch.

After a few moments, when you view your repository ("repo") at github.com/yourusername/yourusername.github.io/ the following should be observed:



username/username

https://github.com/[redacted]/[redacted].github.io

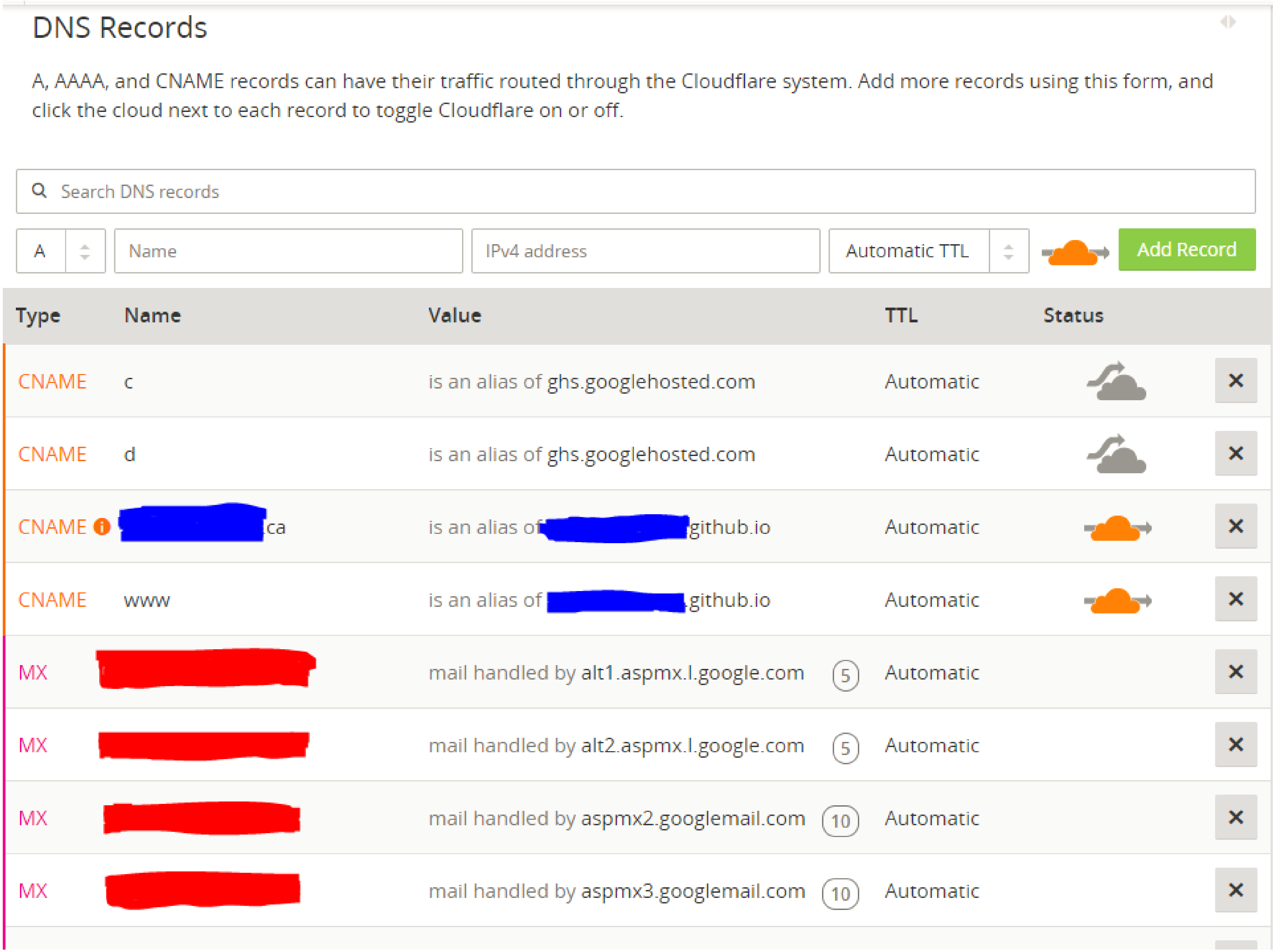
Branch: master ▾ | New pull request | Find file | Clon

[redacted] Add files via upload … Latest commit 74

classic	Add files via upload
css	Add files via upload
d	Add files via upload
font	Add files via upload
img	new background image file, nice lanterns
js	Add files via upload
CNAME	Add files via upload

CNAME will always exist in the root directory of the project with only the CNAME. Github Pages is specifically configured to see this file. As the site is developed more files can be added, but this file must remain intact.

At this stage, flip back to Cloudflare to finish configuration. Add the Github DNS as shown, with yourusername.github.com, where yourusername as from before:

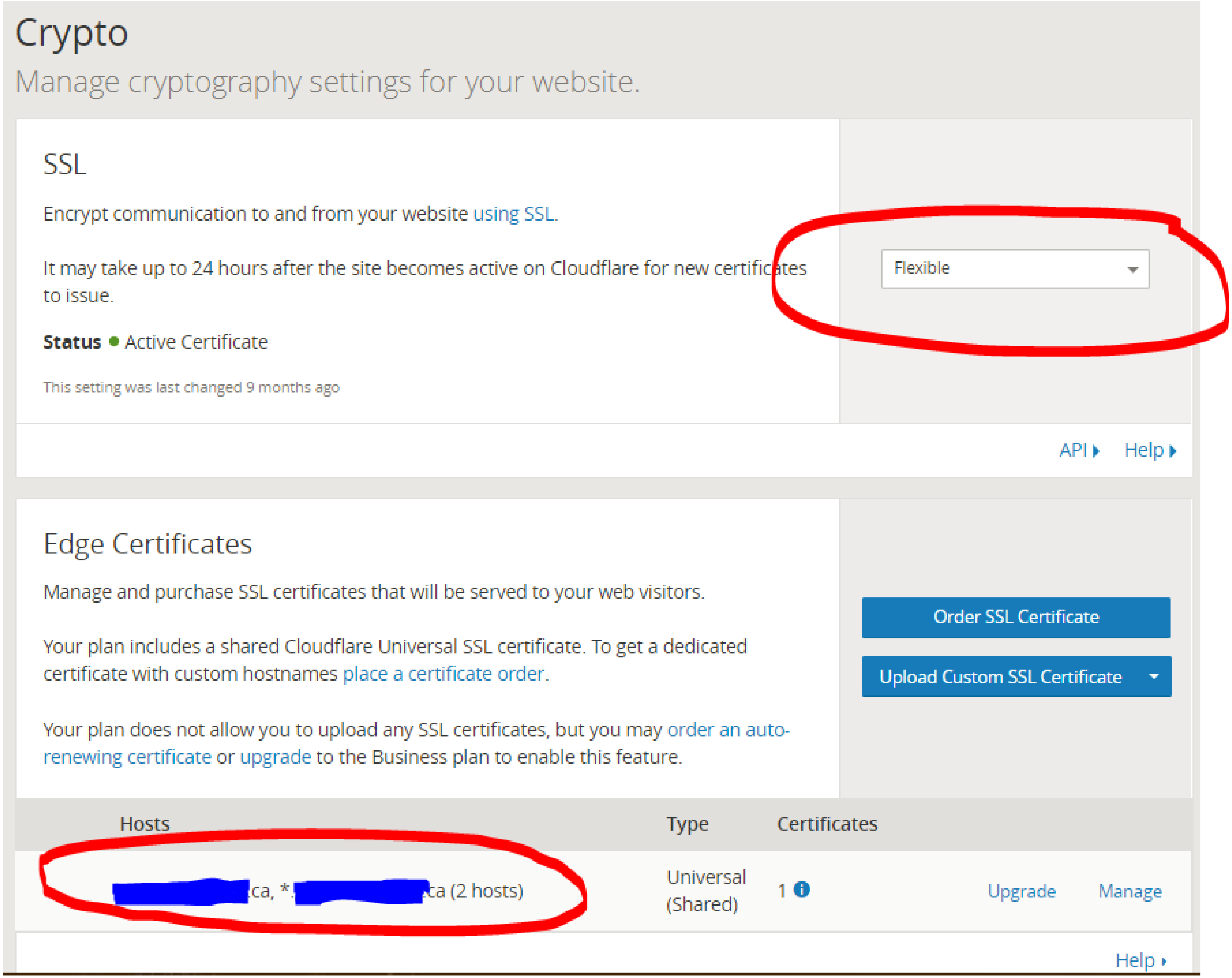


DNS Records

A, AAAA, and CNAME records can have their traffic routed through the Cloudflare system. Add more records using this form, and click the cloud next to each record to toggle Cloudflare on or off.

Type	Name	Value		TTL	Status	
CNAME	c	is an alias of ghs.googlehosted.com		Automatic	☁	✕
CNAME	d	is an alias of ghs.googlehosted.com		Automatic	☁	✕
CNAME ⓘ	[redacted]ca	is an alias of [redacted]github.io		Automatic	☁	✕
CNAME	www	is an alias of [redacted]github.io		Automatic	☁	✕
MX	[redacted]	mail handled by alt1.aspmx.l.google.com	5	Automatic		✕
MX	[redacted]	mail handled by alt2.aspmx.l.google.com	5	Automatic		✕
MX	[redacted]	mail handled by aspmx2.googlemail.com	10	Automatic		✕
MX	[redacted]	mail handled by aspmx3.googlemail.com	10	Automatic		✕

Continuing through the blue navigation tabs, find Crypto, and SSL section. Select *Flexible* setting. Read more clicking Help> at the lower right of the box



Crypto

Manage cryptography settings for your website.

SSL

Encrypt communication to and from your website using SSL.

It may take up to 24 hours after the site becomes active on Cloudflare for new certificates to issue.

Status ● Active Certificate

This setting was last changed 9 months ago

Flexible

Edge Certificates

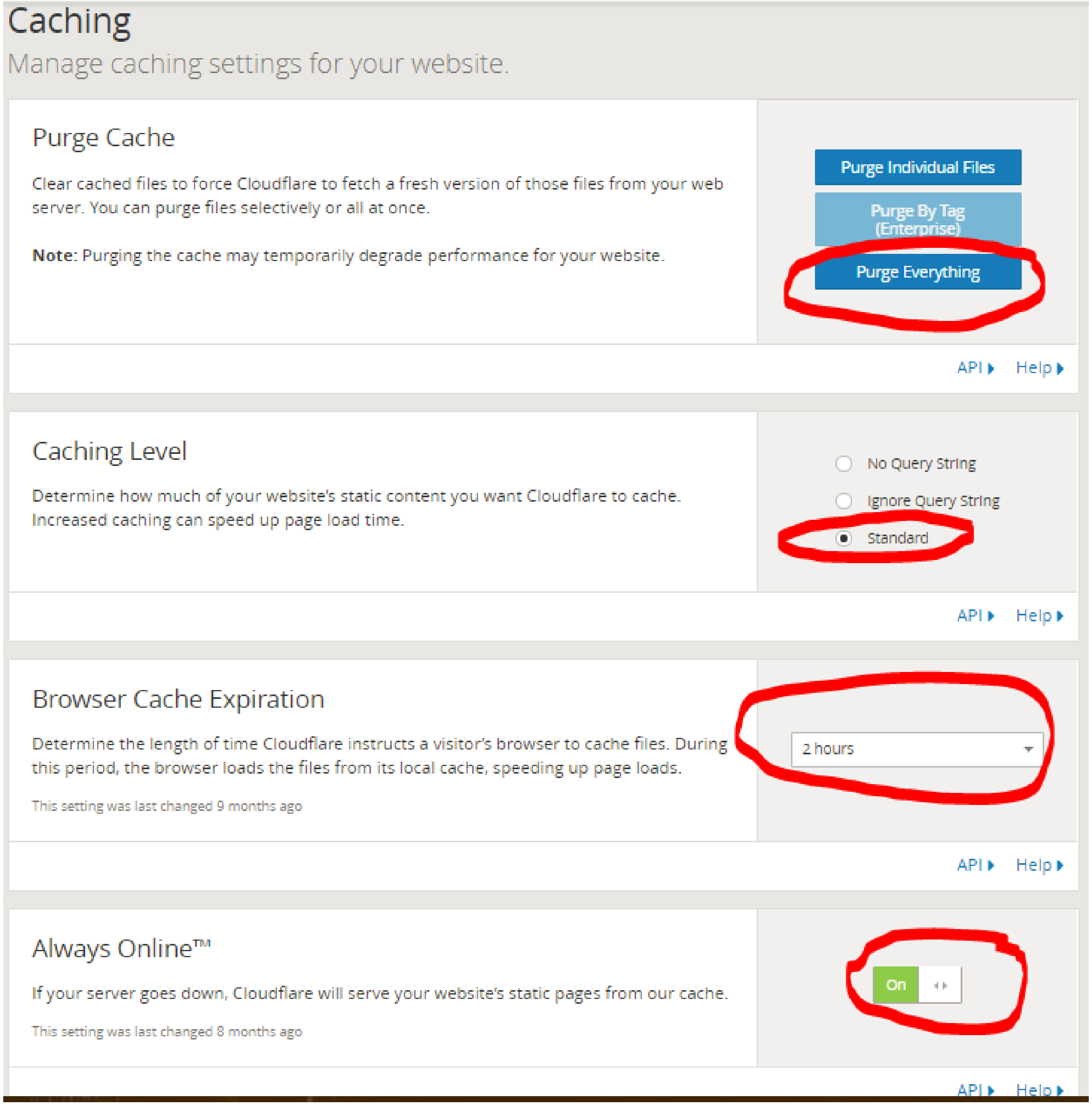
Manage and purchase SSL certificates that will be served to your web visitors.

Your plan includes a shared Cloudflare Universal SSL certificate. To get a dedicated certificate with custom hostnames place a certificate order.

Your plan does not allow you to upload any SSL certificates, but you may order an auto-renewing certificate or upgrade to the Business plan to enable this feature.

Hosts	Type	Certificates		
[redacted]ca, *[redacted]ca (2 hosts)	Universal (Shared)	1 ⓘ	Upgrade	Manage

The following are usual, but using the Help> functionality will explain in detail.



Caching

Manage caching settings for your website.

Purge Cache

Clear cached files to force Cloudflare to fetch a fresh version of those files from your web server. You can purge files selectively or all at once.

Note: Purging the cache may temporarily degrade performance for your website.

Caching Level

Determine how much of your website's static content you want Cloudflare to cache. Increased caching can speed up page load time.

○ No Query String

○ Ignore Query String

● Standard

Browser Cache Expiration

Determine the length of time Cloudflare instructs a visitor's browser to cache files. During this period, the browser loads the files from its local cache, speeding up page loads.

This setting was last changed 9 months ago

2 hours ▾

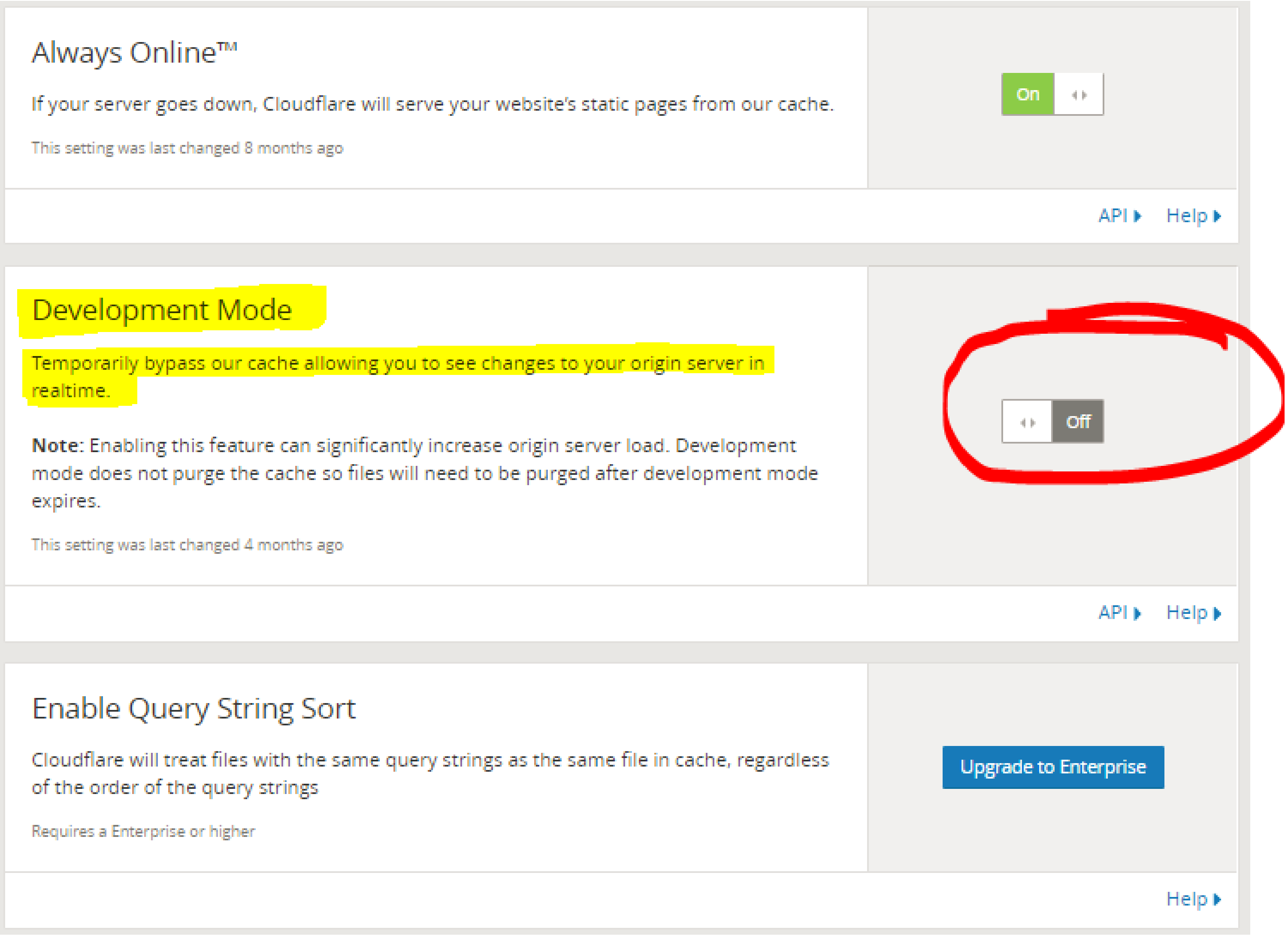
Always Online™

If your server goes down, Cloudflare will serve your website's static pages from our cache.

This setting was last changed 8 months ago

On

This setting is very important, when waiting for changes to be cached, *Development Mode* can be turned on, and select *Purge Everything* option above



Always Online™

If your server goes down, Cloudflare will serve your website's static pages from our cache.

This setting was last changed 8 months ago

Development Mode

Temporarily bypass our cache allowing you to see changes to your origin server in realtime.

Note: Enabling this feature can significantly increase origin server load. Development mode does not purge the cache so files will need to be purged after development mode expires.

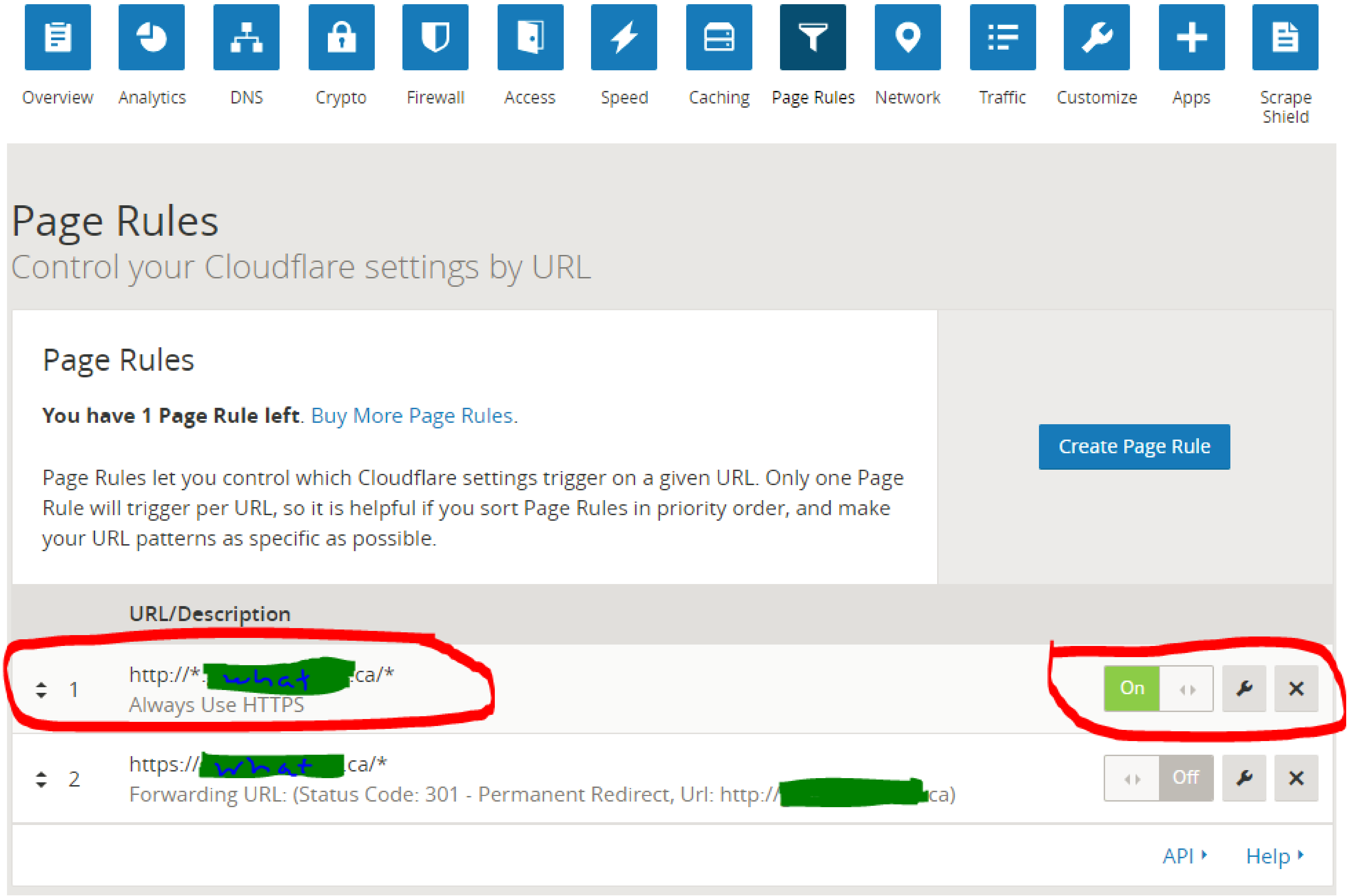
This setting was last changed 4 months ago

Enable Query String Sort

Cloudflare will treat files with the same query strings as the same file in cache, regardless of the order of the query strings

Requires a Enterprise or higher

The last mandatory step is to create a *Page Rule*. Ensure this points to http://*.whateveryourdomainis.ca/* and *Always Use HTTPS* is selected. Ignore the inactive second page rule in the diagram.



Page Rules

Control your Cloudflare settings by URL

Page Rules

You have 1 Page Rule left. Buy More Page Rules.

Page Rules let you control which Cloudflare settings trigger on a given URL. Only one Page Rule will trigger per URL, so it is helpful if you sort Page Rules in priority order, and make your URL patterns as specific as possible.

	URL/Description	
1	http://*.[redacted].ca/* Always Use HTTPS	On
2	https://[redacted].ca/* Forwarding URL: (Status Code: 301 - Permanent Redirect, Url: http://[redacted]ca)	Off

Now close all of your browser windows, and wait for the DNS to propagate throughout the Internet. This could take more than a day sometimes.

If everything worked, your index.html should show up when you go to your domain name. If it isn’t working, start from the beginning and do quick checks to make sure everything is exactly correct on DNS settings, that’s the biggest issue.

This has been a very stable and fast setup,