Cloudflare

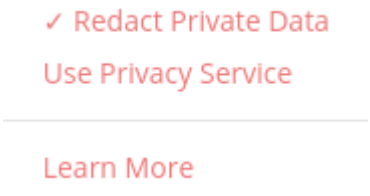
Cloudflare, Inc. is an American web-infrastructure and website-security company, providing content-delivery-network services, DDoS mitigation, Internet security, and distributed domain-name-server services.

- Setup your Domain with Cloudflare
- Create Domain Records to Point to your Home Server on Cloudflare Using Nginx Progy Manager

Setup your Domain with Cloudflare

Image not found or type unknown

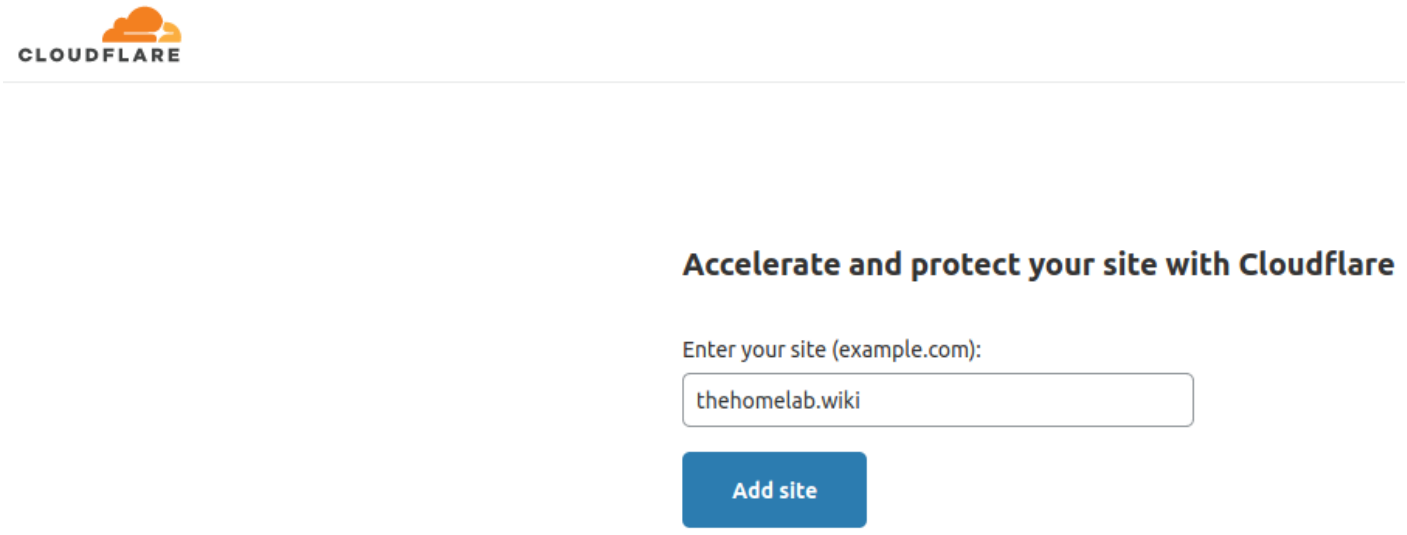
To use Cloudflare as your domain controller, you need to have a domain name already purchased. I use Porkbun because they have a very user friendly dashboard and each domain comes with free domain privacy or redaction.



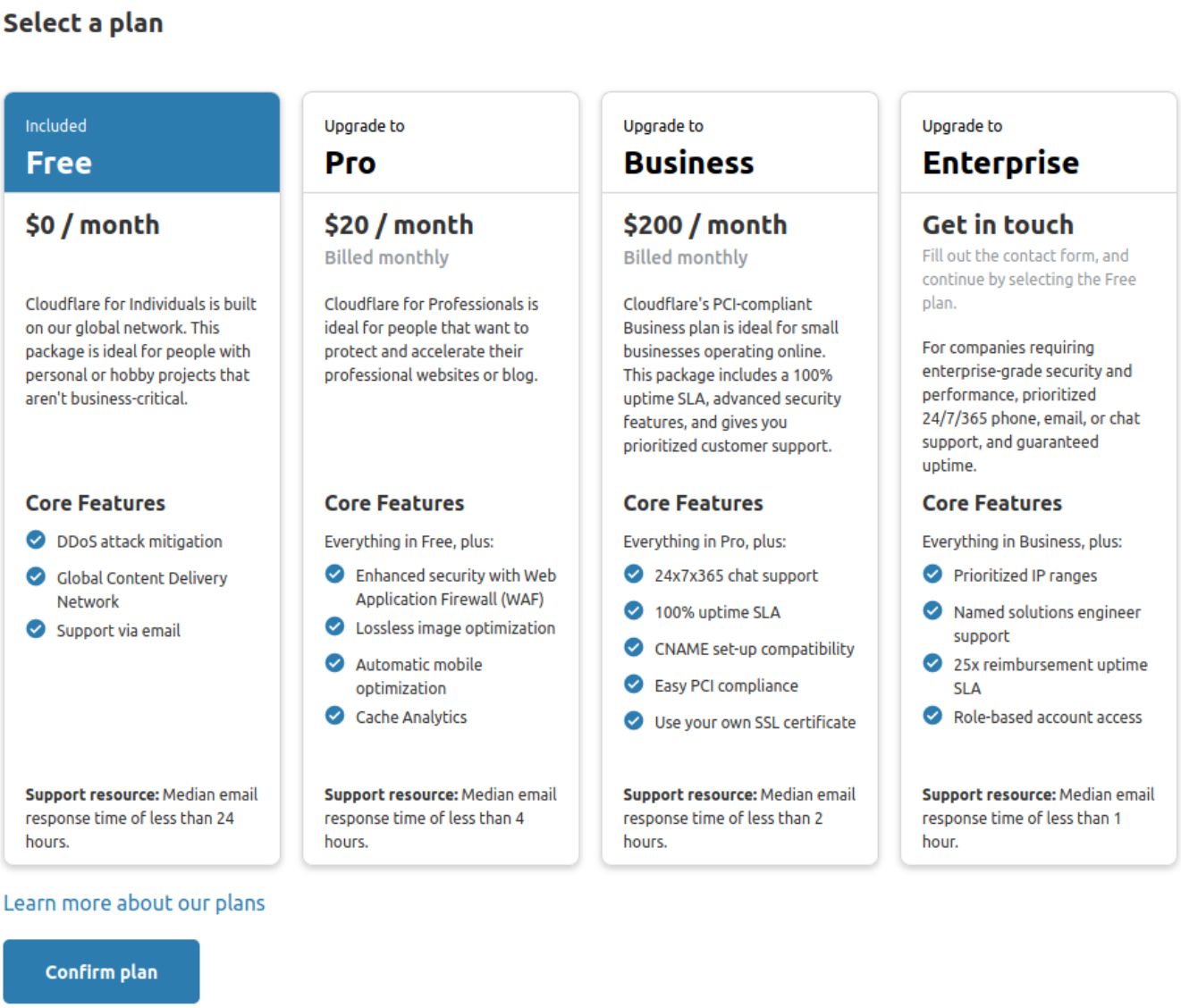
Once you have your domain purchased, you need to create a free Cloudflare account.

Adding your domain to Cloudflare

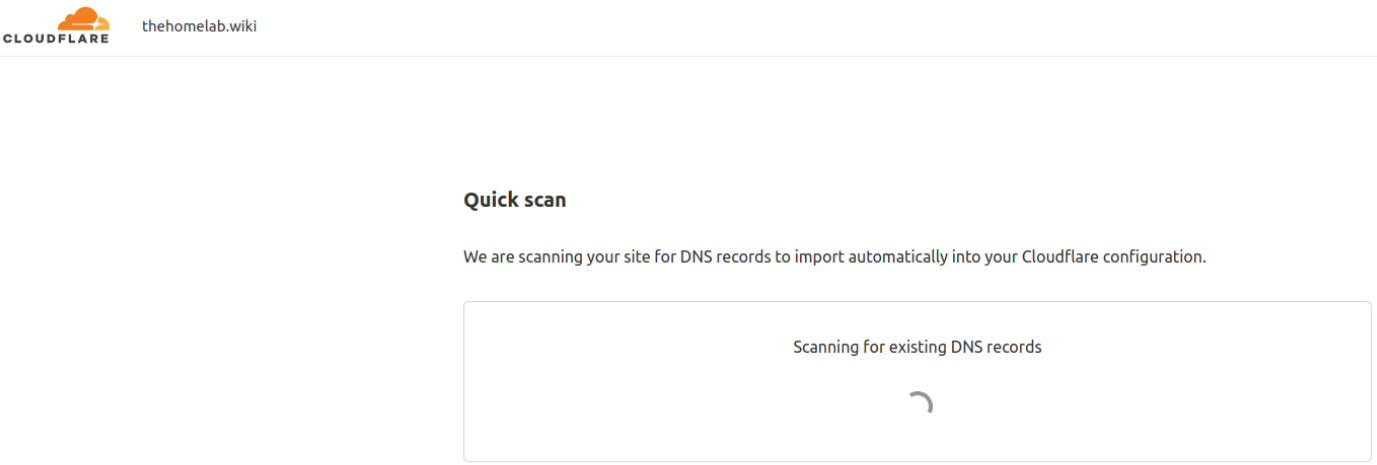
The first time you log in to Cloudflare you'll see place to add your domain name.



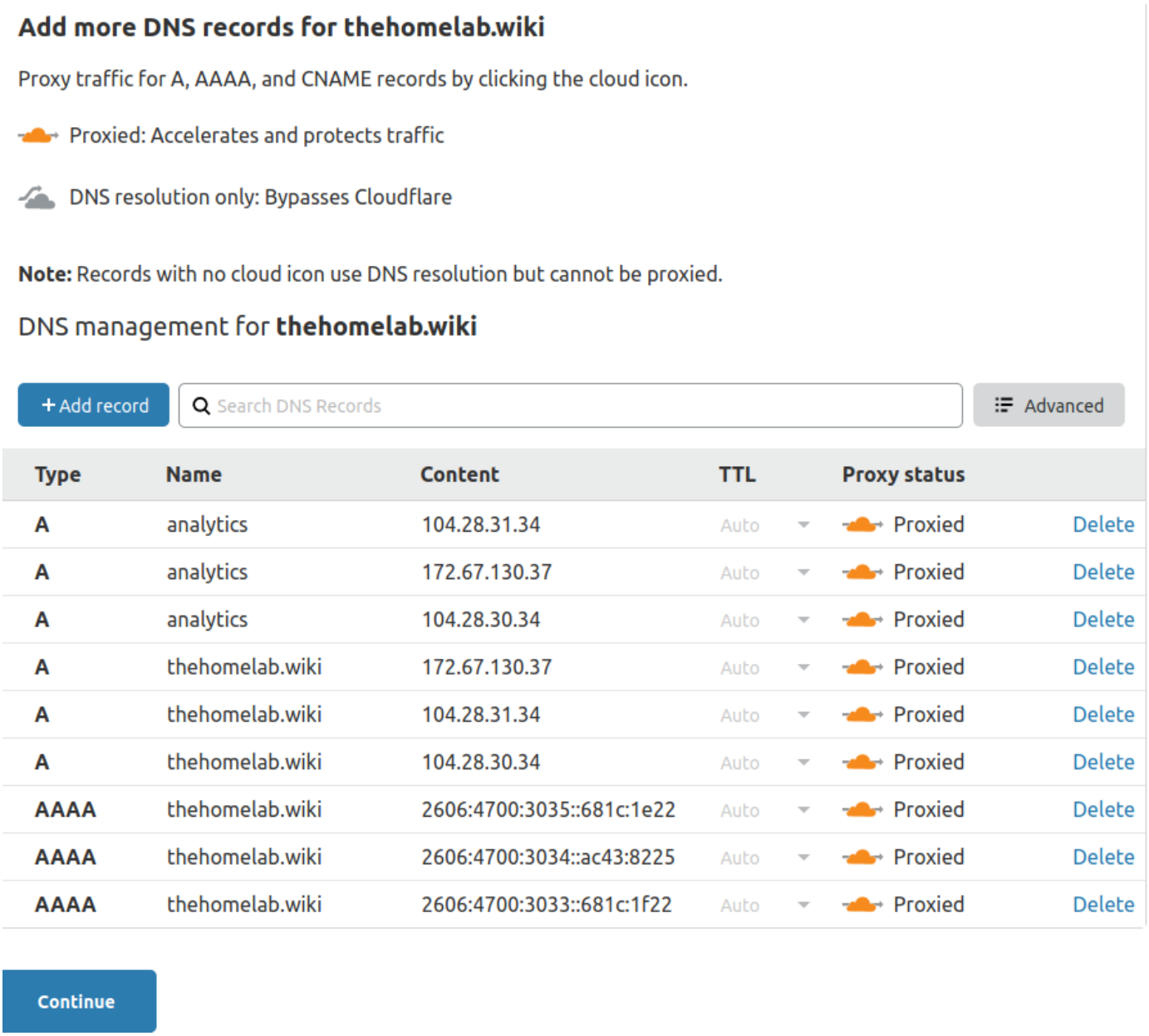
Now click "Add site" then choose the free plan and click "Confirm plan".



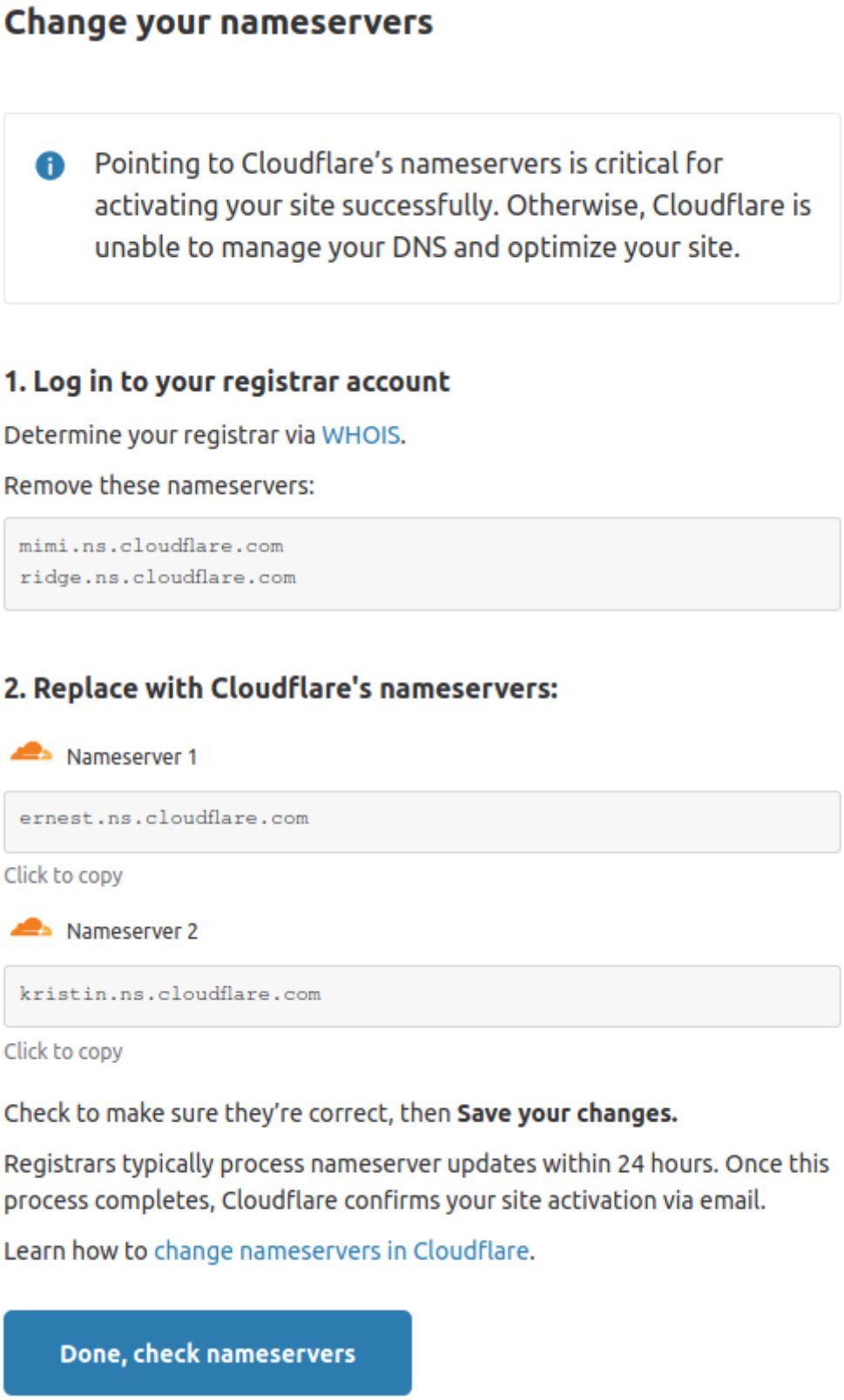
Now Cloudflare will scan your current dns records. These records will most likely be using the DNS records of your domain reseller. In my case it would be Porkbun DNS.



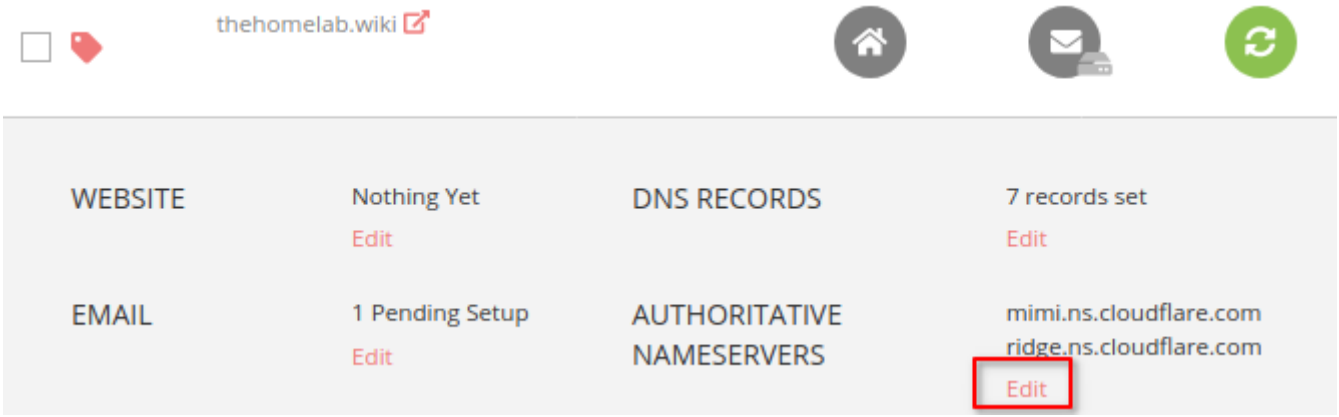
Since this domain is already using Cloudflare, it shows the cloudflare dns IPs



Now click "Continue". Next we have to replace the domain reseller name servers with the provided Cloudflare name servers. again, I already did this.

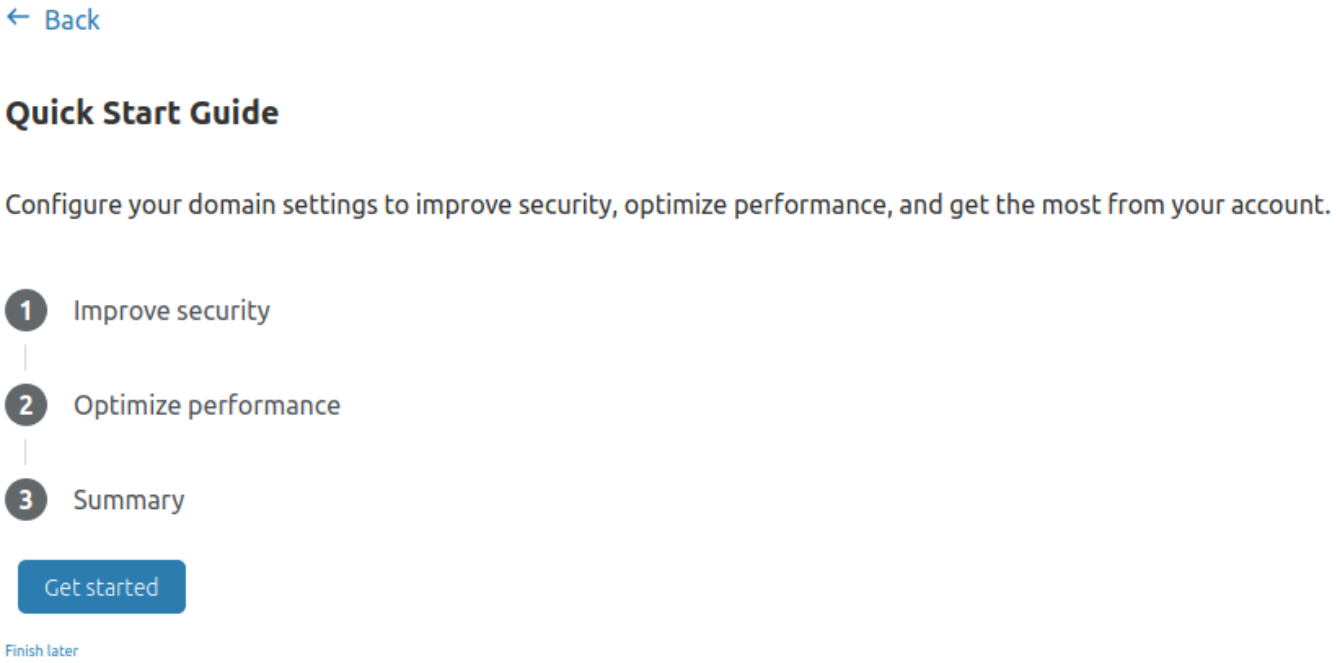


Don't click "Done" until you go to your domain provider and change the name servers. This is where you do it in Porkbun.

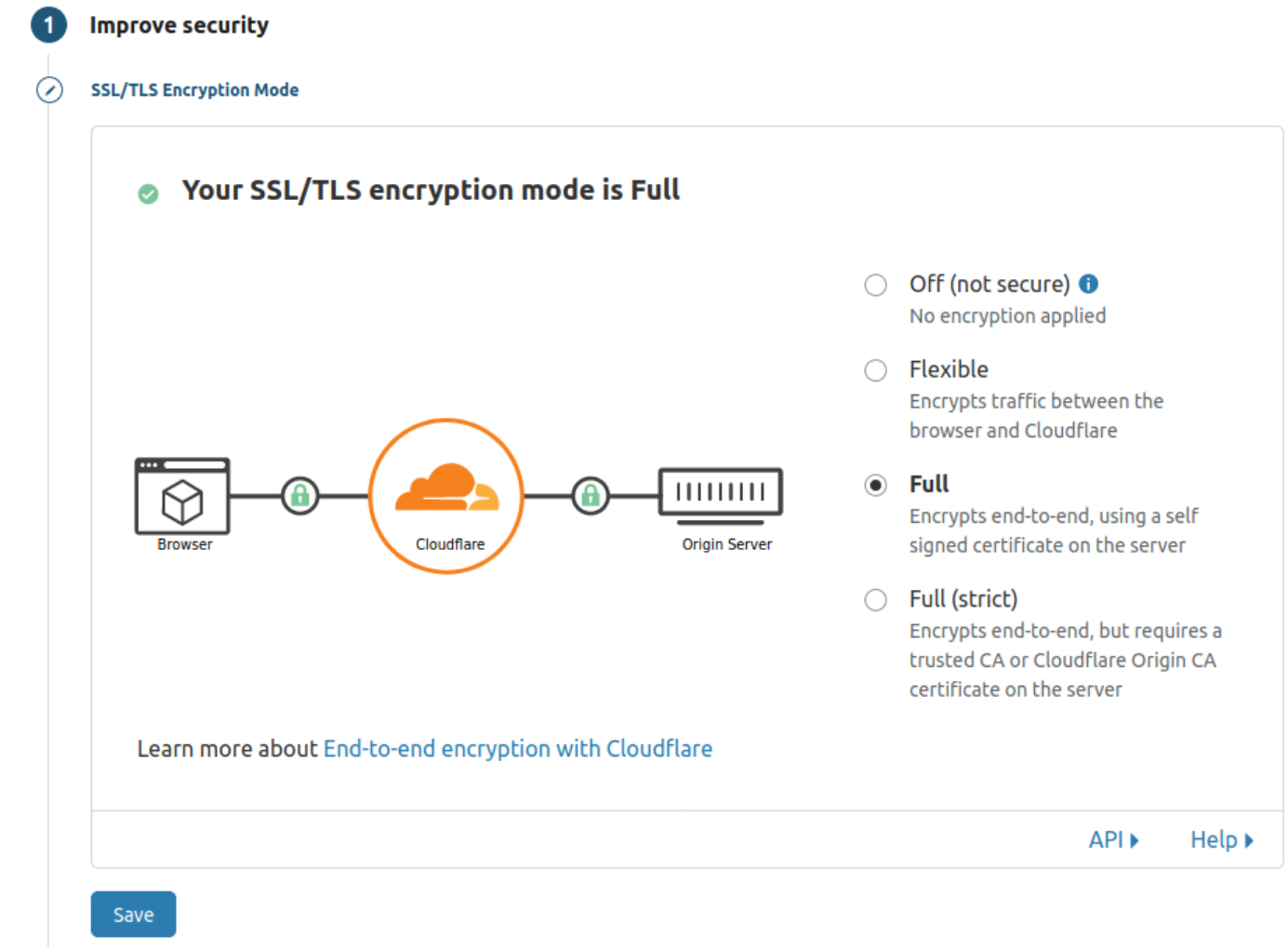


Once the domain name name servers have been changed, click "Done" in Cloudflare.

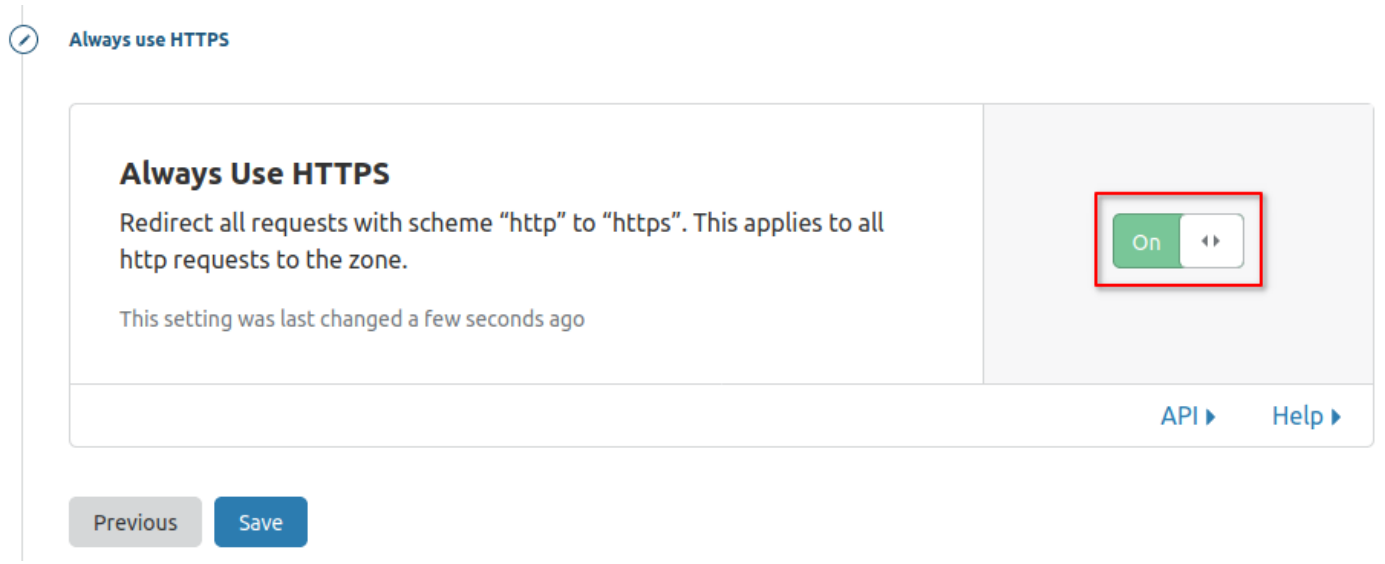
Now it will take you through a "Quick start guide" where you can make a few adjustments to your settings.



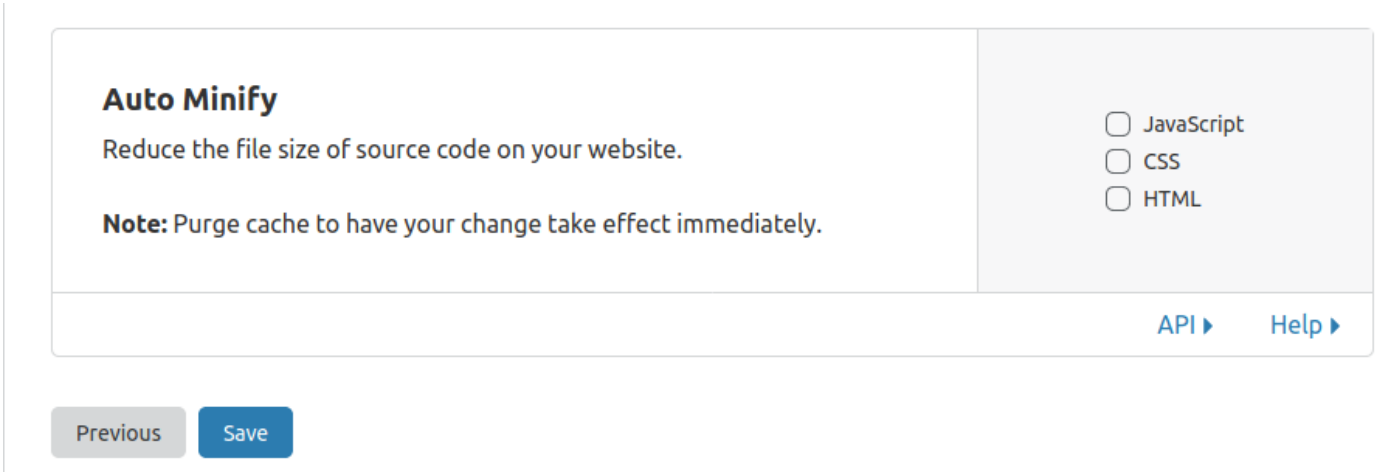
I like to set the Encryption method to FULL because this seems to be the best suited option when using a reverse proxy.



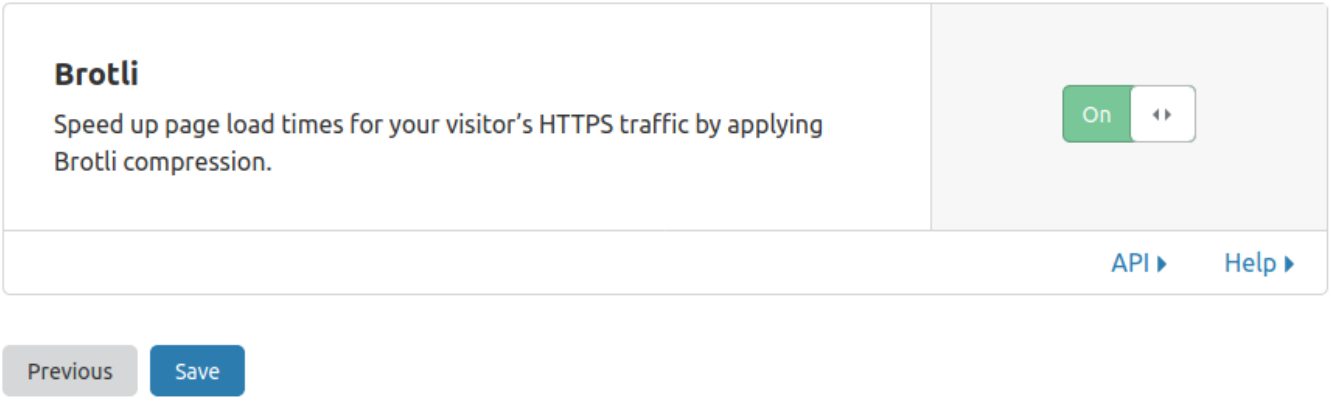
I turn on "always use HTTPS" because this will automatically send traffic through your SSL.



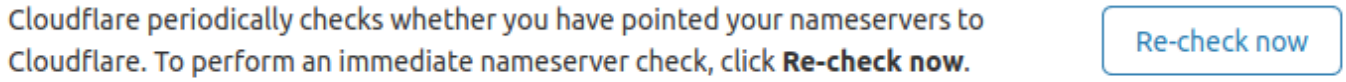
Auto Minify. I never check or change anything here. Just click "Save" and move on.



By default, Brotli is on. Leave this as is. It's always good to have more speed!



Finish the guide and wait for your domain servers to change to Cloudflare.



You can click "Re-check now" once to get a status update. You will get an email when your domain is ready to be managed through Cloudflare.

Now you can ping your domain to see that it is indeed using the Cloudflare DNS.

Image not found or type unknown

Create Domain Records to Point to your Home Server on Cloudflare Using Nginx Progy Manager

Image not found or type unknown

Please refer to the "Setup your Domain with Cloudflare" page before getting started here.

This works best for those who have either a static IP address or a long lease. We have Verizon FioS and have never seen our WAN IP change and we do not pay for a static IP. Maybe we are just lucky. But either way, I will set this up so if our IP does change, all we have to do is change one record for all the rest to follow suite.

We will be using Nginx Proxy Manager for keeping track of our hosts and SSL certificates. I found it is the most user friendly application for this purpose. More specifically, I use the jlesage/docker-nginx-proxy-manager docker image.

Before We Begin With Nginx Proxy Manager

Part 1

I want to preface this by expressing that it is best to run Nginx Proxy Manager on a dedicated machine, VM, container or the likes. This way we have access to port 80 and 443 on the machine so we won't have any conflicts with ports. Port 80 and 443 are the ONLY ports we have to expose on our router to get this setup and working. This allows us to create more services on our network

and expose them to the internet WITHOUT opening more ports for those services to be accessed remotely. Pretty rad isn't it?

Another thing to note is if this machine goes down, all of the hosts will go down with it. That's why it is a good reason to separate this from your other shenanigans.

I run Nginx Proxy Manager on a LXC container in Proxmox but I won't be discussing that aspect of the setup.

Part 2

Nginx Proxy Manager works in conjunction with Docker. Docker allows us to install services and applications and assign ports during installation. These ports are what we use to route our traffic when exposing them using Nginx Proxy Manager. This guide assumes you already know how to use Docker. It's important that you understand the fundamentals and basics of Docker before moving forward.

Creating the Docker container

Begin by executing the following compose command. I use Portianer for easier management, but you can just as easily toss the compose file on your machine and run a docker-compose up -d.

```
docker run -d \
    --name=nginx-proxy-manager \
    -p 8181:8181 \
    -p 80:8080 \
    -p 443:4443 \
    -v /docker/appdata/nginx-proxy-manager:/config:rw \
    jlesage/nginx-proxy-manager
```

Once this is finished installing, you can access the web UI on port 8181 where you will be asked to login.

Default Administrator Account

After a fresh install, use the following credentials to login:

- Email address: `admin@example.com`
- Password: `changeme`

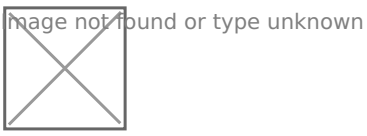
You should immediately change the login credentials before moving forward.

Router Configuration

Now that this is done, we have to configure our router to point the ports to this machine. So take note of the local IP where Nginx Proxy Manager is installed. You can find the IP by typing this command in the terminal.

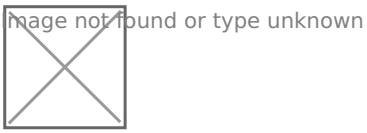
```
ip addr
```

Then you will see it there, usually number 2 in the list or next to your NIC ID.



So in my case I have to forward ports 80 and 443 to 192.168.1.165 in my router. I assume you already knew this but just in case...

There are tons of different routers out there however, most of them are quite similar when it comes to port forwarding. I can't assume you all have a FioS gateway like I do but this is what it looks like. To access your router you will have to know the IP address of the router to get to the administration. Then login. Again, this guide assumes you already know how to do that.



Here you can see I added 2 rules for port 80 and 443 to the IP of the machine Nginx Proxy Manager is hosted on. Now, from here on out, Nginx Proxy Manager will act as our "pseudo router" where we only need to route the traffic to the domain rather than opening more ports.

Again, Please refer to the "Setup your Domain with Cloudflare" page before getting started on this next section. If you did not do this first, the following will not work.

Creating the A Record

1. Log into your Cloudflare CP and go to the DNS page and click Add record
2. Select Type A
3. Put your domain name in
4. Add your WAN IP go here to see it
5. Make sure Proxy status is DNS only (for now)
6. Save the record.

The reason we have to leave the proxy disabled (for now) is so Let's Encrypt can assign the certificate. Once the Host is setup. we can return here and cloak the IP by enabling the proxy status. If you will not be using the root domain, you can go ahead and proxy it now, otherwise create the Host in Nginx Proxy Manager first.

Add the Host

For this part, we have to have a service in mind that we want to expose. It's best to only expose services that have authentication. Such as a Wordpress blog. For applications like Wordpress, it's best to setup the domain BEFORE running the install because wordpress assigns URL's in the databse. So setup Wordpress in Docker but don't run the install until you get the domain setup in the Host in Nginx Proxy Manager.

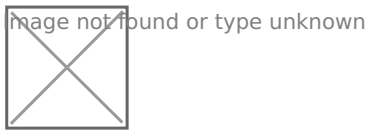
Take note of the docker host IP and the port Wordpress is running on.

In Nginx Proxy Manager go to Hosts

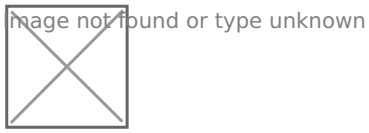
Click on Add Proxy Host button (upper right)

NOTICE: The Domain Name is the domain we setup in Cloudflare. We are Forwarding the domain to the IP of our Wordpress/Docker host on port 8977.

Click SSL at the top to request a SSL certificate then click where it says "None" to drop down and select "Request a new SSL Certificate".

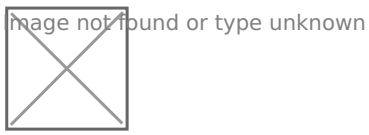


Now tick and agree to Let's Encrypt ToS. Then click Save.

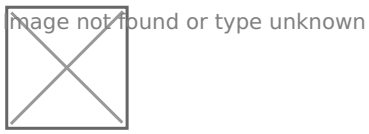


When it's finished, it will close and take you to the dashboard. You will have to go back into this menu under SSL and enable "Force SSL" for this Host. This will ensure that your domain is only accessed on https no matter how it is typed into the address bar.

> Don't forget to go back into the Cloudflare CP and enable the Proxy on the record to cloak your IP. If you're having issues accessing your domain, go into SSL/TLS and set the encryption to Full and try again.

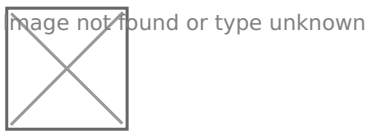


Now you can navigate to the domain in your address bar and begin your Wordpress installation on your domain name that's hosted on your own server!



Adding more Records in Cloudflare

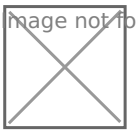
Now that we have established that our root domain is pointing to our WAN IP, we can add more records using subdomains and CNAMES. Each record from here forward will be added as a CNAME derived from our A record. This way if our WAN IP does ever change, all we have to do is change the IP on the A record we made and all of our CNAMES will inherit the new IP.



This example will create bookstack.l33t.host then we will setup the Host on Nginx Proxy Manager the same way we did for Wordpress. Don't forget to leave the records unproxied until you create the Host in Nginx Proxy Manager.

Wrapping Up

You will discover when using Nginx Proxy Manager, some apps like Bookstack require Websockets enabled to work properly. As you tinker with it, you will learn things like this and it will make more sense how things work. Websockets can be enabled in the Nginx Proxy Manager Host



You can support the Nginx Proxy Manager developer on the original Github page. You can subscribe to my Youtube channel for more video guides too. The video below explains this guide. However I recommend using the CNAME method above for adding records rather than using all A records like I did in the video. Just incase your IP does change.

https://www.youtube.com/embed/cI17WMKtntA