Wayne Out There (.com)

wayne-out-there

Stuff that matters to Wayne

HOW TO SET UP YUNOHOST WITH YOUR OWN DOMAIN (USING CLOUDFLARE)

Posted on June 03,2019 by admin

EDIT 19/06/12 - made some tweaks to this after realizing a few small errors. Sorry if you followed before June 12 :(

I really wanted to self-host a kind of 'family box' which would allow me to have self-hosted email, Nextcloud, websites, and a few other basic things and not have it running on someone else's server. During the process of searching I came across Yunohost (Pronounced "Why You No Host?"). I installed it on an old test box (super old) with their own documentation and it was really quite simple, especially if you have done any kind of operating system installation before.

My specific goal was to make it all work with a domain that I own (nameofyourdomain.com for this tutorial). I feel that having a strange email address (the default Yunohost email setup looks weird and is awkward) is of little value for most people so this step must be overcome to become a viable solution for myself and other people I know.

OPTION 1 - DIRECT WITH REGISTRAR (EASIEST)

If your registrar gives you full control of your DNS, CNAME, MX etc settings you might not even need this Cloudflare-Yunohost tutorial. My registrar didn't allow me to do what I needed so I went to the next step. I don't have enough experience to speak about the different registrars and their settings so research that yourself if you want. Otherwise, move on to this exciting Cloudflare-Yunohost setup…

OPTION 2 - WITH A CLOUDFLARE SCRIPT

As mentioned above, my registrar didn't make it clear how to do CNAME stuff and mess with MX records, etc, so I ended up searching high and low for an open, free and reliable solution. Thanks to the free software community, I was pointed towards Cloudflare. People I trust and like consider Cloudflare to be 'good guys' and that was enough for me to trust and try.

Forgive my limited understanding and description, but I'll do my best here: Cloudflare is a super robust 'web traffic controller' which gives the website admin person (since you are installing Yunohost that is you now!) really powerful control over how data moves to/from the domain/servers. They have cool controls and a nice interface too. Anyway, they have a free account you can start which allows you to do everything in this tutorial and through the process you'll get a chance to see how nice Cloudflare ("CF" moving forward) is too.

Note: this tutorial assumes you are using Ubuntu or at least have the same terminal commands.

PRE-FLIGHT BULLET POINTS
We're going to do this:

1. Tell your domain registrar to point traffic to CF
2. Tell your Yunohost ("YH" moving forward) box to point to CF
3. Stick a free software script on your YH box that automatically tells CF where your YH box is every 30 minutes (in case your IP address changes)(replaces dynamic dns service need…)

THE FLIGHT

1. Do the YH setup as per yunohost. Make sure your router's ports are open! Check this page about ports and note that some ISPs will (unethically?) block you from using port 25 (email) and in this case you might be completely out of luck or have to change your ISP so you should check that first on this page. If port 25 is blocked you should be able to use everything except email (nextcloud,

wordpress, etc should work) so it's not completely without hope… TIP! If you get your domain setup first in YH sometimes Cloudflare will be able to magically import all your stuff automatically helping you avoid the manual inputs over at CF

2. Get a Cloudflare account
3. Log into your domain registrar and change nameservers to the ones shown in your CF account. This guy's video is pretty good if you haven't done it before.
4. Take note of this project, which is the script which will automate the DNS updates stuff. Special thanks to the programmer!
5. ssh into your yunohost box by typing (where 123 stuff is the local IP address of your YH box):
   ssh admin@123.123.123.12
   This will get you into your YH box where you can stick the script files into your home directory.
6. Clone the cloudflare-ddns project files above into your YH box by typing this into your terminal (TIP! do NOT use 'sudo' here!):
   git clone https://github.com/adrienbrignon/cloudflare-ddns.git
7. Then change to your new directory:
   cd cloudflare-ddns
8. Then change to the zones directory within:
   cd zones
9. Then copy the example yml file so that it duplicates and is named to your own domain:
   cp example.com.yml nameofyourdomain.com.yml
10. Now open the file so you can edit the contents:
    sudo nano nameofyourdomain.com.yml
11. Now edit the 'admin@example.com' line and change to the email you registered your CF account with
12. Change whatever it says to the right of cp_api_key: to your cloudflare api key. There is a link of how to find that right in the terminal window but in case it stresses you out and you miss it here is the link
13. Change zone name (cf_zone:) to: nameofyourdomain.com
14. Set all the DNS stuff so that the file looks like this where the # signs are 'comments' telling the script to forget about this part:

Only write the subdomain ('ddns' for 'ddns.example.com')

```
cf_records:
- '@':
type: A
# proxied: true
log: ERROR
```

- 'www':
  type: A
  # - 'ddns':
  # type: AAAA
  # ttl: 300
  # proxied: false
  # log: INFO

If you compare to the example file you can see the changes.

I just commented-out with hashtags the AAAA stuff since apparently I don't need it (a great contributer told me) as well as the smallest but most painful one, the 'Proxied: true" line! This one, if you don't put a hashtag in front, will, every time your cron job runs, tell CF to make CF the controller of the DNS and then basically shut down your websites and your email will also stop working. Then you have to go in and turn the orange cloud back to grey again.

1. Save and exit the nano editor with control x and 'yes'

2. Then move back up one level in the directory so you can run next command:
   cd ..
3. Run a one-off test to see if it's working as per the usage docs with this command
   python cloudflare-ddns.py -z nameofyourdomain.com

if it's 'working' you should:

a) see a success report back from your terminal that looks something like this:

```
2019-05-31 05:16:15,165 | INFO | The record 'www.nameofyoudomain.ca' (A) is already up to date
```

and

b) should be able to now go to your CF account and see the IP address of where your YUNO-box is / public-facing router listed in the DNS area. You can check this part by going into your router (or one of those 'what's my IP address?" websites) and compare your router's IP address with the IP address in Cloudflare - they should be the same now. If not, assure that the orange cloud in CF is turned off by clicking. It will change to a grey colour when off and this is what you want.

If everything is looking good, let's move on to making this update process happen automated in the backgroun since it would not be fun to have to run this test script every day or a few times a day!

Now that your config file for this script is all good, let's go and do the cron job thing

1. Type:
   crontab -e

Probably if you haven't done this already it will say it's blank and give you two options. choose 'nano' because it's easier (option 1) and not VIM because it's brutal and hurts

1. The script provided in the usage page for doing this next step assumes you know what you are doing, that you understand Linux file structures and paths and even cron jobs. I didn't. So, I'm going to spare you the pain here (you can read the pain below in the bonus section(s) if you are bored or like learning) and tweak this script so that you have a higher chance of this working. First, this is what was provided from the usage page:

Every 30 minutes, update my Cloudflare records.

```
*/30 * * * * python /path/to/cloudflare-ddns.py -z example.com
```

If you just simply ran this tutorial that means your Cloudflare directory that you git-cloned in step 6 above is in your home directory. However, you need to add in the user into the path for this to work properly. Also, until you know this thing is working, I would advise you add in the MAILTO option above the script so that you can get a few emails for a few hours confirming it is or isn't working. You can go back in and remove or comment out with a # the MAILTO line (or comment it out with a hashtag) after you are sure everything is working.

So, here is what I did that finally made it work

Every 30 minutes, update my Cloudflare records.

```
MAILTO=myemail@myreliablemail.ca
*/30 * * * * python /home/admin/cloudflare-ddns/cloudflare-ddns.py -z nameofyourdomain.com
```

If everything is working, you'll keep getting 'success' emails that look like this everytime the cron job runs:

```
2019-05-31 05:30:05,942 | INFO | The record 'www.mylataylor.ca' (A) is already up to date
```

At this point I went back in (see step 17 above) in and hashtagged out the MAILTO= line so the emails stopped coming every thirty minutes.

ADJUST YOUR CLOUDFLARE SETTINGS!

1. CRYPTO/SSL SETTING
   This one took me an addition day to figure out. I was getting continual TOO_MANY_REDIRECTS errors when trying to access my Yunobox. The problem was solved by the click of one box in my Cloudflare settings as follows:

crypto / SSL : change to 'FULL' in the dropdown.

1. TURN YOUR ORANGE CLOUDS TO GREY
   If you don't do this step, your email and a few other things won't work. Just click the orange clounds in your DNS section so they turn grey. That's it.

DO YOUR LETS ENCRYPT SSL CERTIFICATE ON YOUR YUNHOST BOX

If you try to do your letsencrypt SSL cert before these steps are done, it won't let you (from my experience) But at this point it should all work. In your YH admin interface, just go to domains, nameofyourdomain.com, SSL certificate, and then 'install lets encrypt'

CONCLUSIONS

Now your Yunobox should be automatically reporting back your router's IP address to Cloudflare and Cloudflare is routing your website traffic through its nameservers, etc. As long as your ports and certificates are working, you should now be able to start using it with your own domain.

Now that this is done and you go to your new domain and nothing is there that's because… there is nothing there. Go figure. So you have to install an 'app' (ie wordpress, nextcloud) through the Yunohost app area of admin. I'll do a separate blog on that probably, but it's pretty easy.

Posted in:Freedom And Privacy,Life Skills,Nextcloud,Technology,Tutorial,Ubuntu | Tagged:Howto,Yunohost | With 2 comments