HOW TO MAKE A NEXTCLOUD PI BOX WORK AS REVERSE PROXY TO YUNOHOST

Posted on June 21,2019 by admin

Background

The situation was that I wanted to test out the very cool project Yunohost but I already had Nextcloudpi (another awesome project!) running on my local network. I already had a DDNS service (No-ip) running which was pointing to my Nextcloudpi ("NCP" moving forward) box, and a second DNS service that I set up which pointed to my router for the purpose of Yunohost ("YH" moving forward). You can read about that cool DNS solution in my other blog post, by the way, as it works really well and gives a bit more power.. and it's free.

The problem was that ports 443 and 80 were being used by NCP but YH needed them as well. The only options appeared to be:

a) change the ports of one of the machines (complicated because clients outside of the LAN in the world webs won't know those ports) or
b) figure out what a 'reverse proxy' is and then make it work

The challenge was that NCP was using Apache whilst YH uses NGINX - both of which are capable of reverse proxy. So, in order to do this I ended up doing some learning of both although it turns out it wasn't really needed after all. C'est la vie…at least I learned some things!

At the end of the journey of trying about 10,000 different settings in the Apache default configuration file that comes with NCP (and other Apache installs) called "000-default.conf" it started working after adding just two lines to my configuration which seemed not to be in any other tutorial online for some reason. The key two lines that were needed were:

SSLEngine On
SSLProxyEngine On

Without those two lines it would just never work even though the rest of my settings were right.

Ok, enough of my hard journey story, let's log the actual configuration and steps so that anyone who wants to do the same setup can save the pain!

Assumptions

Before we begin, I will assume that you already have the following set up:

1. Server A (in my case NCP) running Apache which is already successfully reachable and working from the outside world. Through this machine Server B will be reached.
2. Server B (in my case YH) running whatever (I think) but in my case it's running NGINX and this box is the one we are trying to make visible to the outside world through ports 80 and 443
3. You have a domain (nameofyourdomain.com in this tutorial) which you own and which is already successfully hitting your router (You can test by pinging the domain and seeing the IP address of your router show up). You can do this with my other tutorial mentioned above as well. You can also get a free 'domain' from services like No-ip if you don't care what the domain looks like.
4. You have full access to SSH into both machines, but in this case Server A is the critical one.
5. You are using an Ubuntu environment and have know how to open a Terminal and use it (roughly)
6. You are willing to learn and try things if this doesn't perfectly work as per this specific example. I'll

give you a few resource links as well to help you in case your set up needs tweaking.

Let's Begin - Setting up Apache Default Config on Server A

1. ssh into Server A (format ssh username@your.IP.Address)
2. Change directory (cd) to your Apache2 sites-available directory. In my case it looks like this but if you aren't using NCP it might be different
   cd /etc/apache2/sites-available
3. Type this command to back up your Server A apache settings. If you mess anything up you can restore this one and delete the default and rename it back to original name.

```
sudo cp 000-default.conf 000-default.backup
```

1. Check to make sure the new file with .backup is showing up by typing 'ls'. If it's there then proceed.
2. Copy the sample configuration below into your clipboard
3. Open the default Apache config file with this command (if you haven't used nano before probably good to do a quick online overview) for editting:
   sudo nano 000-default-conf
4. you may have some settings already in this file (you should) at the top. Scroll down to the bottom of whatever is there and then paste in the sample you have copied from below with the control + shift + v (If you don't hold shift it won't paste)
5. Go through the newly-pasted configs and adjust to your settings changing domain names and ip addresses to yours.
6. Control x to save and exit, 'y' to save modified buffer and 'enter' key to write your changes
7. Restart apache with this command to see if it works (this will shut down whatever stuff is running on Server A so probably good idea to do this wisely if the server is currently being used by others…:

```
sudo systemctl restart apache2
```

If you get nice silence from your terminal, and no 'journalctl' messages, then things are going the right direction.

Run Let's Encrypt Manually for SSL certs on Server A

For this step, to be honest, I'm not sure if you need to do it because certs are already on both boxes for NCP and YH. But you might not have that so I'll provide the steps since after I did them nothing was worse and everything was working... I would love to get some feedback on this step.

1. Install Let's Encrypt tools:
   sudo apt-get install python-certbot-apache
2. Run it
   sudo certbot --apache -d example.com -d www.example.com

Let's Finish - Test Server B

Go to your domain from outside your LAN (just to make sure you are getting a real test) and try to hit Server B. I find mobile phone data plans are good for this kind of testing, otherwise, call your grandma and ask her what happens when she goes to nameofyourdomain.com…

If it works, you're done.

If it doesn't you might need to tweak your settings.

Sample Configuration - copy this and adjust to your set up

Your IP address will obviously be changed to the correct one where your Server B is. Copy everything in the code block below.

```
 <VirtualHost *:80>
    ServerAdmin name@nameofyourdomain.com
    ServerName nameofyourdomain.com
    ServerAlias www.nameofyourdomain.com
   ProxyPreserveHost on
   ProxyPass / http://192.168.1.37:80/
   ProxyPassReverse / http://192.168.1.37:80/
</VirtualHost>
#Listen 443
<VirtualHost *:443>
    SSLEngine On
    SSLProxyEngine On
     ServerAdmin name@nameofyourdomain.com
     ServerName nameofyourdomain.com
     ServerAlias www.nameofyourdomain.com
     ProxyPreserveHost on
     ProxyPass / https://192.168.1.37:443/
     ProxyPassReverse / https://192.168.37:443/</VirtualHost>
```

FULL Sample Configuration Reference (DO NOT COPY THIS ONE)

This is what my config looked like when everything was done and working.

The 'Rewrite engine' stuff here was added by Lets Encrypt when it was run so it 'should' appear in your config after you run it after initial settings have been added. Same with the 'Include' stuff and the SSL certificate stuff at the bottom of the second entry.

```
<VirtualHost *:80>
    ServerAdmin name@nameofyourdomain.com
    ServerName nameofyourdomain.com
    ServerAlias www.nameofyourdomain.com
   ProxyPreserveHost on
   ProxyPass / http://192.168.1.37:80/
   ProxyPassReverse / http://192.168.1.37:80/
RewriteEngine on
RewriteCond %{SERVER_NAME} =nameofyourdomain.com [OR]
RewriteCond %{SERVER_NAME} =www.nameofyourdomain.com
RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,NE,R=permanent]
</VirtualHost>
#Listen 443
<VirtualHost *:443>
    SSLEngine On
    SSLProxyEngine On
     ServerAdmin name@nameofyourdomain.com
     ServerName nameofyourdomain.com
     ServerAlias www.nameofyourdomain.com
     ProxyPreserveHost on
     ProxyPass / https://192.168.1.37:443/
     ProxyPassReverse / https://192.168.37:443/
Include /etc/letsencrypt/options-ssl-apache.conf
SSLCertificateFile /etc/letsencrypt/live/nameofyourdomain.com/fullchain.pem
```

SSLCertificateKeyFile /etc/letsencrypt/live/nameofyourdomain.com/privkey.pem</VirtualHost>

Random Keywords and messy spam from the Journey

This next section is merely a copy/paste of all the steps I was trying to try to get this working. The purpose is not to follow any of these instructions but merely to leave as keywords in hopes that other people trying the same things will end up finding this blog and save themself the pain! :) So, don't use the next section for any form of tutorial but feel free to read and learn.

1. set up individual virtual host conf files on box 1 else:

We were unable to find a vhost with a ServerName or Address of mydomain.ca.
Which virtual host would you like to choose?

1: nextcloud.conf | mydomain.hopto.org | HTTPS | Enabled
2: ncp.conf | | HTTPS | Enabled
3: 000-default.conf | | | Enabled

Select the appropriate number [1-3] then [enter] (press 'c' to cancel):

Select the appropriate number [1-3] then [enter] (press 'c' to cancel): c
No vhost exists with servername or alias of mydomain.ca. No vhost was selected. Please specify ServerName or ServerAlias in the Apache config.
No vhost selected

hmm.

finding apache config…

seems like one shouldn't mess with this… and that lets encxrypt probably does it for you

1. sudo apt-get install python-certbot-apache (apparently not installed on ncp somehow..)
2. created basic conf file in /sites-available
3. restarted apache - worked
4. added symlink to sites-enabled, restarted apache, breaks
5. run certbot without enabled…with usual
   sudo certbot --apache -d example.com -d www.example.com

pi@nextcloudpi:/etc/apache2 $ sudo certbot --apache -d mydomain.ca -d www.mydomain.ca
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator apache, Installer apache
Cert not yet due for renewal

You have an existing certificate that has exactly the same domains or certificate name you requested and isn't close to expiry.
(ref: /etc/letsencrypt/renewal/mydomain.ca.conf)

What would you like to do?

1: Attempt to reinstall this existing certificate
2: Renew & replace the cert (limit ~5 per 7 days)

---

choosing option 2

fail. same error above

now trying to go back to simply 443 config in 000-default but wtihout ssl engine stuff.

now running:
sudo certbot --apache -d mydomain.ca -d www.mydomain.ca

this is something… progress….

the bad part:

Failed redirect for mydomain.ca
Unable to set enhancement redirect for mydomain.ca
Unable to find corresponding HTTP vhost; Unable to create one as intended addresses conflict; Current configuration does not support automated redirection

the good part

IMPORTANT NOTES:

- We were unable to set up enhancement redirect for your server, however, we successfully installed your certificate.
- Congratulations! Your certificate and chain have been saved at: /etc/letsencrypt/live/mydomain.ca/fullchain.pem
  Your key file has been saved at: /etc/letsencrypt/live/mydomain.ca/privkey.pem
  Your cert will expire on 2019-09-14. To obtain a new or tweaked version of this certificate in the future, simply run certbot again with the "certonly" option. To non-interactively renew all of your certificates, run "certbot renew"

Posted in:Freedom And Privacy,Life Skills,Nextcloud,Technology,Tutorial,Ubuntu | Tagged:Nextcloud,Nextcloudpi,Nginx,Ubuntu,Yunohost | With 0 comments