

# Services

Setting up new services in my Homelab

- [Setup OpenProject](#)
- [Home Assistant Monitoring of Furnace Filter](#)
- [Setup NetBooting with Unraid Docker and Firewalla](#)

# Setup OpenProject

## Setting up OpenProject in my Homelab

I am starting this on a fresh install of Ubuntu 22.04 Jammy in my ProxMox Server. I am going to use the DEB/RPM installation as the docker installation is not recommended for production usage. First, lets do an update and upgrade, make sure you are logged in as a user that has root privileges.

```
sudo apt-get update

sudo apt-get upgrade -y
```

Next, install package certificates, the OpenProject keys, and the OpenProject List

```
sudo apt-get install apt-transport-https ca-certificates wget

wget -q0- https://dl.packager.io/srv/opf/openproject/key | sudo apt-key add -

sudo wget -O /etc/apt/sources.list.d/openproject.list \
https://dl.packager.io/srv/opf/openproject/stable/12/installer/ubuntu/22.04.repo
```

Now to actually install OpenProject

```
sudo apt-get update

sudo apt-get install openproject
```

Now, proceed with initial installation.

```
sudo openproject reconfigure
```

Now, to proceed with the installation. I choose not to install the BIM as I won't work in construction project management. I chose to let OpenProject install a SQL server of its choosing as I like it all in one container if I ever want to delete it, or re-create it. Next, it will ask you to install a web-server, or use one that you already have, I chose to let it install the default Apache 2 server. After this, it will ask you to enter your fully qualified domain name, I have this as a subdomain under my TLD ( Top Level Domain ), that I will be setting up my reverse proxy for. After that, it will ask you if you wish to install this as a path prefix, since I didn't want to do that, I just left it empty.

Next it will ask you for SSL configurations, as I am going to be using my Nginx reverse proxy to provide the SSL certificates and pass the connection on to the actual OpenProject Server. So, I chose no. Next it will ask you for SVN/Git integration, since I won't be working with code, I also chose not to add this. Next, you administrator email, and if you would like to install a memcached server. After this, it will start to install OpenProject. If you would like to change your answers, simply do the following.

```
sudo openproject configure
```

At this point, you should be able to login to the local instance of your OpenProject server.

<http://server-local-ip>

Now, to setup the internet side of things. First, I created a new CNAME to be a subdomain of my TLD using Cloudflare. I then pointed that CNAME to my home IP address, and turned off Cloudflare Proxy. My router already has ports 80 and 443 set to be open to my SWAG container, I just have to add a new Proxy-Conf to the container. This is done by going to where you mapped the config volume of the container to, and going to /nginx/proxy-confs folder. There should be quite a few confs already in there, either add one more, or rename one of the example ones.

Below I have added what my openproject.subdomain.conf is, you will have to make a few changes for your local machine.

**CNAME** - change this to whatever you put as your CNAME in Cloudflare

**SERVERIP** - change this to whatever the local IP is for your OpenProject server

```
# Open Project Server

server {

    listen 443 ssl;
    listen [::]:443 ssl;
    server_name CNAME.*;

    include /config/nginx/ssl.conf;

    client_max_body_size 0;

    location / {
        include /config/nginx/proxy.conf;
        include /config/nginx/resolver.conf;
        set $upstream_app SERVERIP;
        set $upstream_port 80;
```

```
    set $upstream_proto http;
    proxy_pass $upstream_proto://$upstream_app:$upstream_port;
    proxy_set_header X-Forwarded-Proto https;
}
}
```

The first lines tell it to listen on port 443 ( SSL encrypted port ), line 7 tells it what subdomain to listen for. Next, lines 9 & 11 include the SSL configurations, and allow any body size. Lines 14 & 15 tell it where the proxy configuration is, and where the resolvers are. Lines 16, 17 and 18 tell the SWAG proxy about the server that it is proxying, with line 19 actually telling it to do the proxying. I had to add Line 20 to allow SWAG to pass along the SSL header to the OpenProject server, otherwise the server will think that it is not getting a proper HTTPS response, when the proxy server is doing all the work. That should be it, you should now be able to access your new OpenProject server from <https://CNAME.TopLevelDomain.com>

# Home Assistant Monitoring of Furnace Filter

## General Description of Problem

I have a typical high efficiency natural gas furnace in my house, controlled by an Ecobee 3. My issue is that overtime the furnace filter gets clogged. When this happens, the flame inside the furnace heats up to the point where it trips itself out, forces the flame off, and keeps the air running on low to make sure the furnace is fine. On the surface, this might seem like something to just set a reminders every 3 months on my phone to change the filter, but, since we don't have AC, the furnace doesn't run during the summer months. This means that I would actually have to just inspect the filter ( which would be time consuming ), or push it off until the next week ( and then completely forget about it ).

## How to Solve

1. Install a sensor with a light that will turn on when the pressure differential across the filter is too great. This will work in theory, however, I will still have to check the light, as well as finding and installing the differential sensor. This will be a backup plan.
2. Install a sensor to monitor either the fan or the flame sensor on the furnace. Both of these methods have the same problem, they would both mean that I would need to design and build a one time use circuit.

## My Solution

Since I don't want to be installing any custom sensors, I thought about just using what I already have. My Ecobee is already controlling the furnace, it knows when it is requesting heat, and it knows when the fan is on. Enter Home Assistant. It will allow me to monitor the Ecobee, and will send me an alert if the fan is on for X amount of time. If the fan is on for over 20 minutes, it probably means that the furnace has tripped, and it is time to replace the filter to allow the furnace to breath.

## Home Assistance Setup

I did this the easy way, using a script from <https://tteck.github.io/Proxmox/> Since you should never just run scripts that you find on the internet on your machines, I first looked at the code. The code seemed to do what it promised, namely to download and extract the current KVM ( qcow2 ) image from Home Assistant, and then ask how and where to set it up on the machine. As soon as the script finished, I was able to find the local IP of the Home Assistant server under the summary page of the node it had just created.

After that was setup, I went to that IP on port 8123. Created my local username and password, and then I was greeted by a bunch of integrations I could setup. I saw that Ecobee was in the list, so I clicked and entered in the correct details to link it to my cloud Ecobee account. I then setup another integration of my Slack server so I can be notified when the fan has been running. At this point, I thought just setup an automation that will notify me when the fan has been running for over 5 minutes. This will allow me to test that the Slack integration is working, and that Home Assistance knows that the fan has been running for 5 minutes. This is when I ran into 2 problems.

1. I couldn't select the fan as a trigger. The Ecobee knows that that fan is on from the screen, however, it is not exposing that to Home Assistant through the cloud integration. I am assuming this is to save on the data it sends to the cloud.
2. Even when I did select something that I could trigger on, the heat coming on, the update time is too slow to be of any practical usage other than logging. This is because Home Assistant is talking to the Ecobee cloud server which is getting updates from my home Ecobee. Less than ideal.

So, how to solve these issues. Well, as it turns out, with a bit of Googling, they are both able to be solved, and even improved upon when you connect the Ecobee to Home Assistant via the Apple HomeKit integration. This will allow the Ecobee to directly talk to the Home Assistant server, so even if my home internet goes out, I will still be connected. I might not be able to get the notifications, but, I will still be able to log in Home Assistant. I accomplished this by going into the menu on the Ecobee, and starting the Homekit setup. This displayed a QR code, and a string of numbers at the top. When you go back into Home Assistant, you input those numbers, and it will connect.

At this point, I thought I could just go to the automation tab, and pull in the automation, since using the Homekit connection will allow me to see the current hvac\_action parameter. Unfortunately, this is not the case as it only has a limited selection of things that it will automatically pull in. However, I can now see the state that I want to monitor, so we can make an automation, just not with the nice GUI that it provides. Below is the code that I used.

```
alias: Slack Notify When Filter Clogged
description: "provides notification on a slack channel when hvac_action - heating has been on
for over 20 minutes "
trigger:
  - platform: state
    entity_id: climate.my_ecobee_2
    attribute: hvac_action
```

```
from: idle
to: heating
for:
  minutes: 20
action:
  - service: notify.SlackChannel
  data:
    message: The filter needs changing
    title: Furnace
    target: home-automation
mode: single
```

This code will monitor the hvac\_action state. If it has been in the state of "heating" for 20 minutes, it will send out a notification on the slack channel provided. To test this, first I put the time to 5 minutes to see if it would trigger, as well as putting in my old clogged filter to make sure that it will trigger.

I know that most people will simply just set a reminder to change the filter every 3 months, but, this is a simple automation that will notify me when the filter actually needs changing because it is too clogged to allow air through it anymore.

# Setup NetBooting with Unraid Docker and Firewalla

First we are going to setup NetBoot.xyz in a docker container. This is done for me using the wonderful Unraid container made by LSIO, however, this can also be done with the following docker compose file

```
---
version: "2.1"
services:
  netbootxyz:
    image: lscr.io/linuxserver/netbootxyz:latest
    container_name: netbootxyz
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Etc/UTC
      - MENU_VERSION=1.9.9 #optional
      - PORT_RANGE=30000:30010 #optional
      - SUBFOLDER=/ #optional
    volumes:
      - /path/to/config:/config
      - /path/to/assets:/assets #optional
    ports:
      - 3000:3000
      - 69:69/udp
      - 8080:80 #optional
    restart: unless-stopped
```

I needed to adjust some of the ports so they weren't interacting with my previously installed items, but that is all. After that is started, netboot.xyz is up and running. If you want to see the configuration, go to the IP:3000. The only thing that I changed at that point was to upgrade the menu's to the latest.

Now it is time to setup the DHCP server to give out the TFTP server where it can be booted from. This is where you can find a lot of router setups, however, Firewalla is not included in this listing.

<https://docs.linuxserver.io/images/docker-netbootxyz/>

To setup Firewalla to allow the computers to netboot, we have to do 2 things. The first thing to do is to get SSH access to your device to add a file.

- Tap on the top right gears button.
- Tap on "**Advanced**".
- Tap on "**Configurations**".
- Tap on "**SSH Console**".
- Tap on the "\*\*\*\*\*".
- If it's the first time you using SSH, please tap on "**Reset Password**" before you start.
- Tap on "**Reveal Password**".
- A list of networks will be shown on the UI. (Gold/Purple only)

Now to SSH into the device itself use the following command.

```
ssh pi@x.x.x.x
```

where x.x.x.x is your Firewalla IP address. This will prompt you for a password, which the above procedure will give you. Next we will want to create the follow file in  
~/.router/config/dhcp/conf/pxe.conf

```
# next-server X.X.X.X
# filename "netboot.xyz.kpxe"
dhcp-option=66,X.X.X.X
dhcp-option=tag:br0,67,netboot.xyz.kpxe
# dhcp-option-force=tag:br0,66,"X.X.X.X"
dhcp-boot=tag:br0,netboot.xyz.kpxe,tftp,X.X.X.X
```

Please add in your X.X.X.X where that is the IP address of your netboot.xyz installation.

Please note that firewalla only has "vi" installed by default, so this is a small reminder on how to use it.

```
cd ~/.router/config/dhcp/conf/
vi pxe.conf
i ( press for insert mode )
paste in the code from above
press the escape key ( to exit insert mode )
:wq ( type this in to "write" and "quit" )
```

At this point, it seems like this will work, however there is more settings in the firewalla app to make.

- Open the App, get to the home network, and tap Network
- Press the Lan that has your DHCP server on it
- Tap Edit in the top right
- Scroll down and click on the Add DHCP Option button
- Option code 66, option value is the IP of your netboot.xyz installation
- Option code 67, option value is netboot.xyz.kpxe

The above will allow computer that don't use a UEFI bios to netboot. However, if we want to get UEFI computers booting using netboot, we have to make a couple changes. I haven't found a change that allows both UEFI and non-UEFI computers to boot at the same time. To make the change, edit the file on the firewalla device to the file below.

```
# next-server X.X.X.X
# filename "netboot.xyz.kpxe"
dhcp-option=66,X.X.X.X
dhcp-option=tag:br0,67,netboot.xyz.efi
# dhcp-option-force=tag:br0,66,"X.X.X.X"
dhcp-boot=tag:br0,netboot.xyz.efi,tftp,X.X.X.X
```

and also in your option 67 on the firewalla app.

This should allow your computers to net boot and not have to worry about those pesky install disks ever again.