ssh root@89.117.146.214

install portainer on the server or docker-compose.yml

1. **Install and Run Portainer**

Since you're using Docker, you can easily set up Portainer with the following steps:

**Step 1: Create a Persistent Volume for Portainer**

This ensures Portainer keeps data even after the container is restarted.

bash

CopyEdit

docker volume create portainer\_data

**Step 2: Run Portainer in a Docker Container**

Now, deploy Portainer with the following command:

bash

CopyEdit

docker run -d \

--name portainer \

--restart always \

-p 9000:9000 \

-p 8000:8000 \

-v /var/run/docker.sock:/var/run/docker.sock \

-v portainer\_data:/data \

portainer/portainer-ce

89.117.146.214:9000 to test

1. **Build Docker using the existing  project code ﻿**

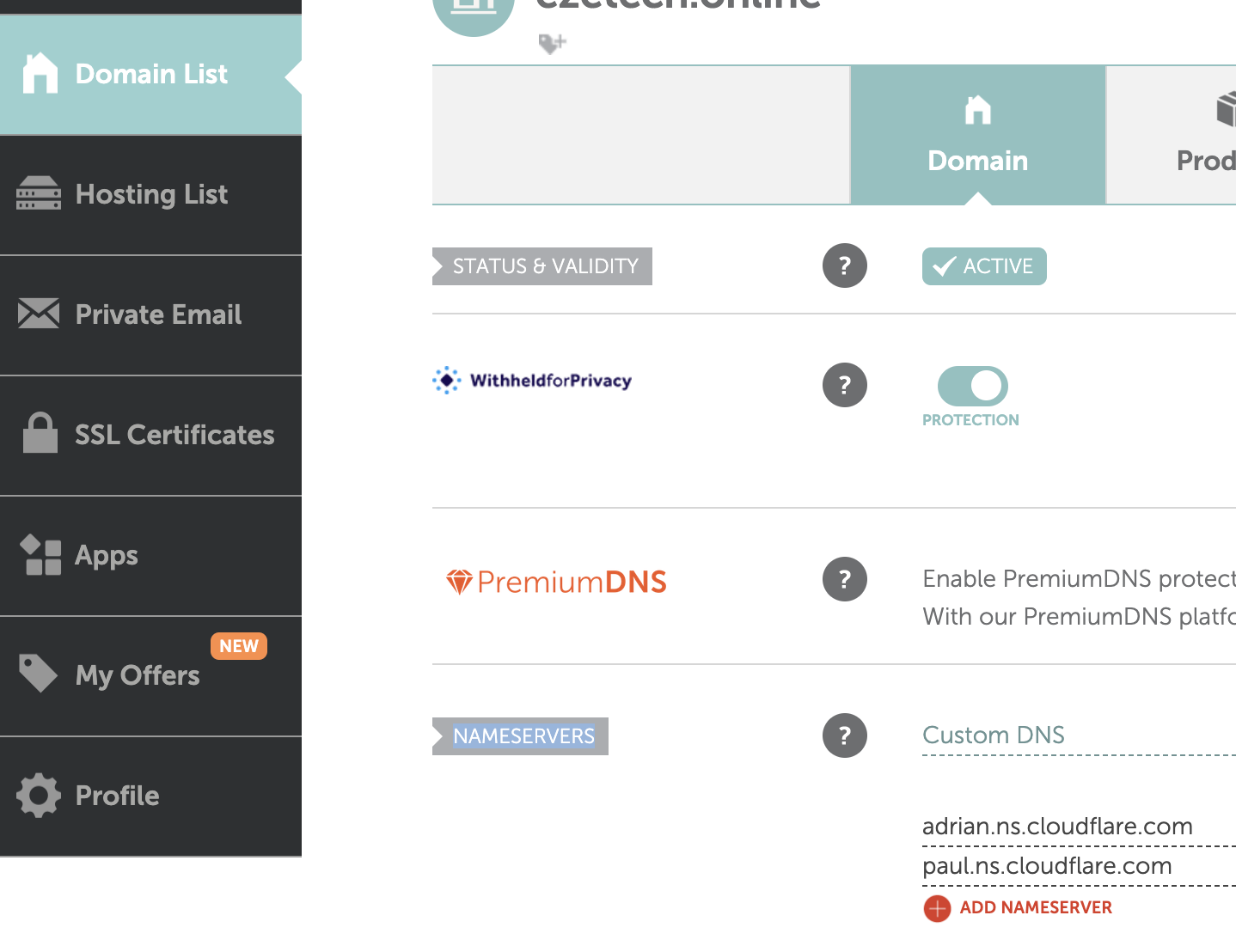
API\_HOST=localhost:3000 // localhost is server because project is stored on VPS

docker compose up -d —build

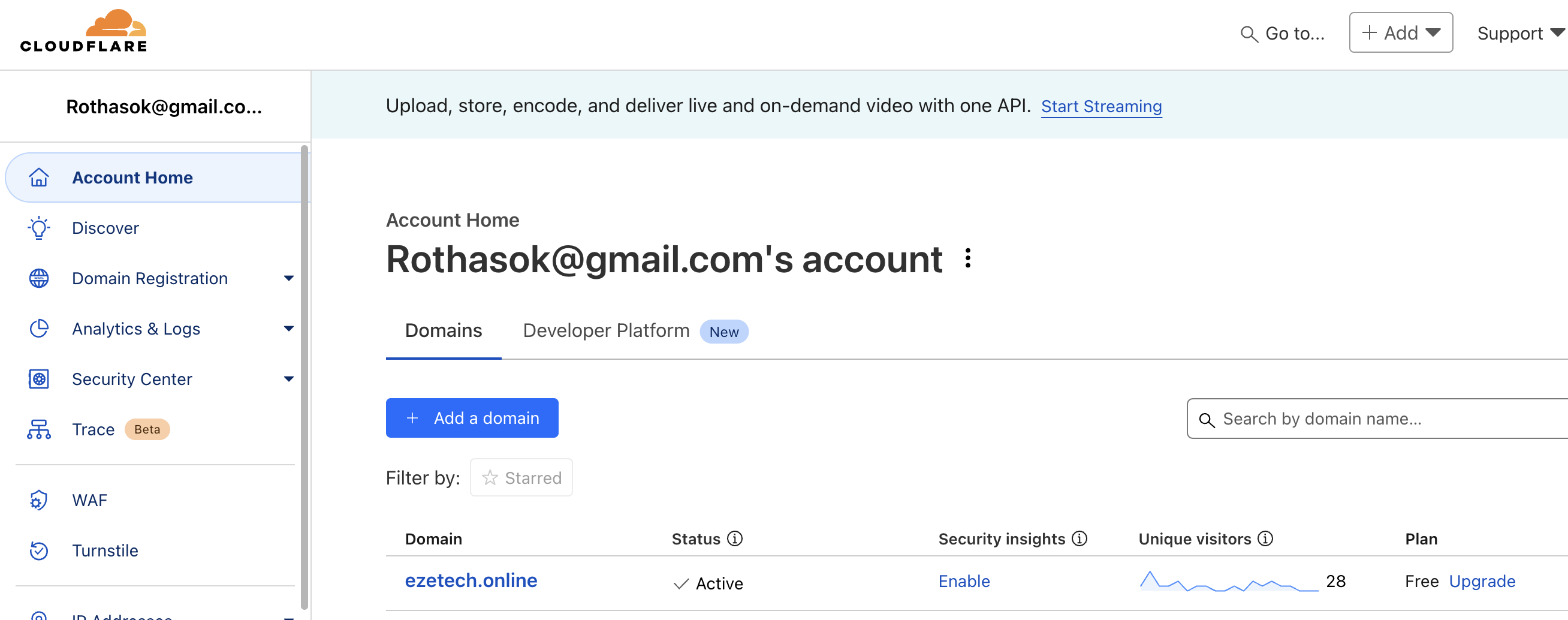
1. **Testing using IP server**

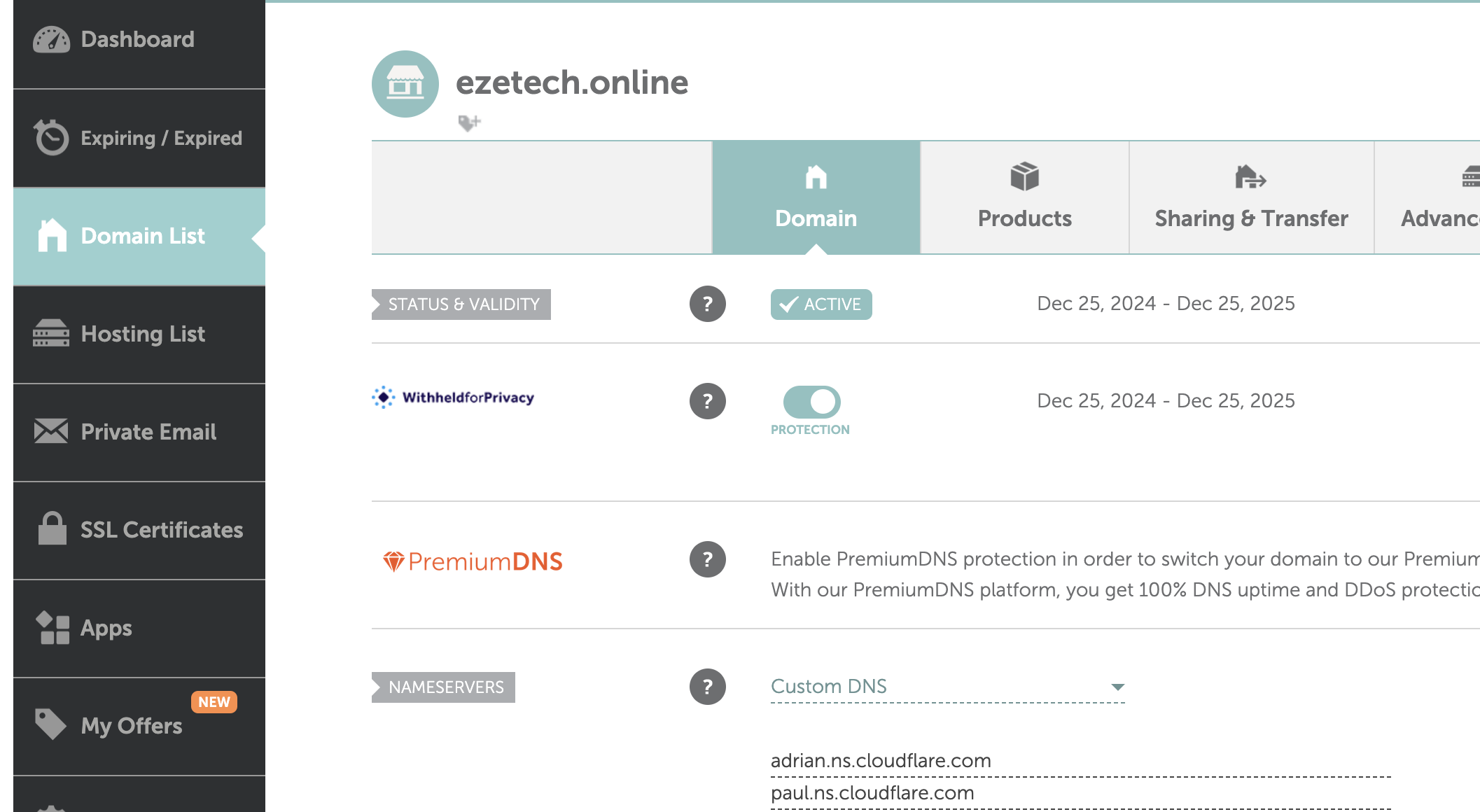
<https://api.ezetech.online/v1/owner?page=1&limit=5>

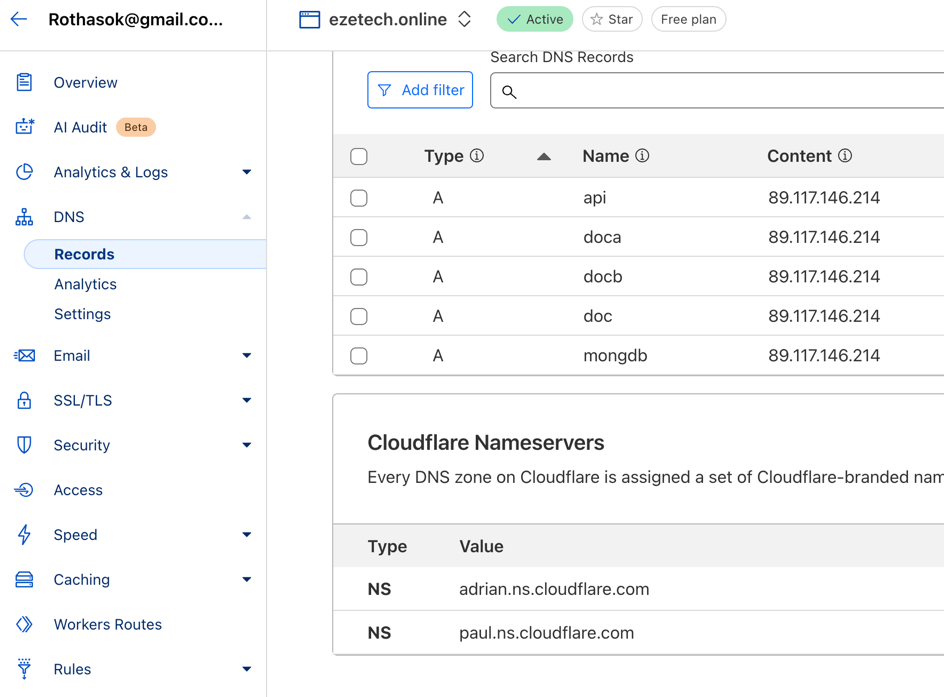
1. **Change IP to domain**  
   Namecheap domain



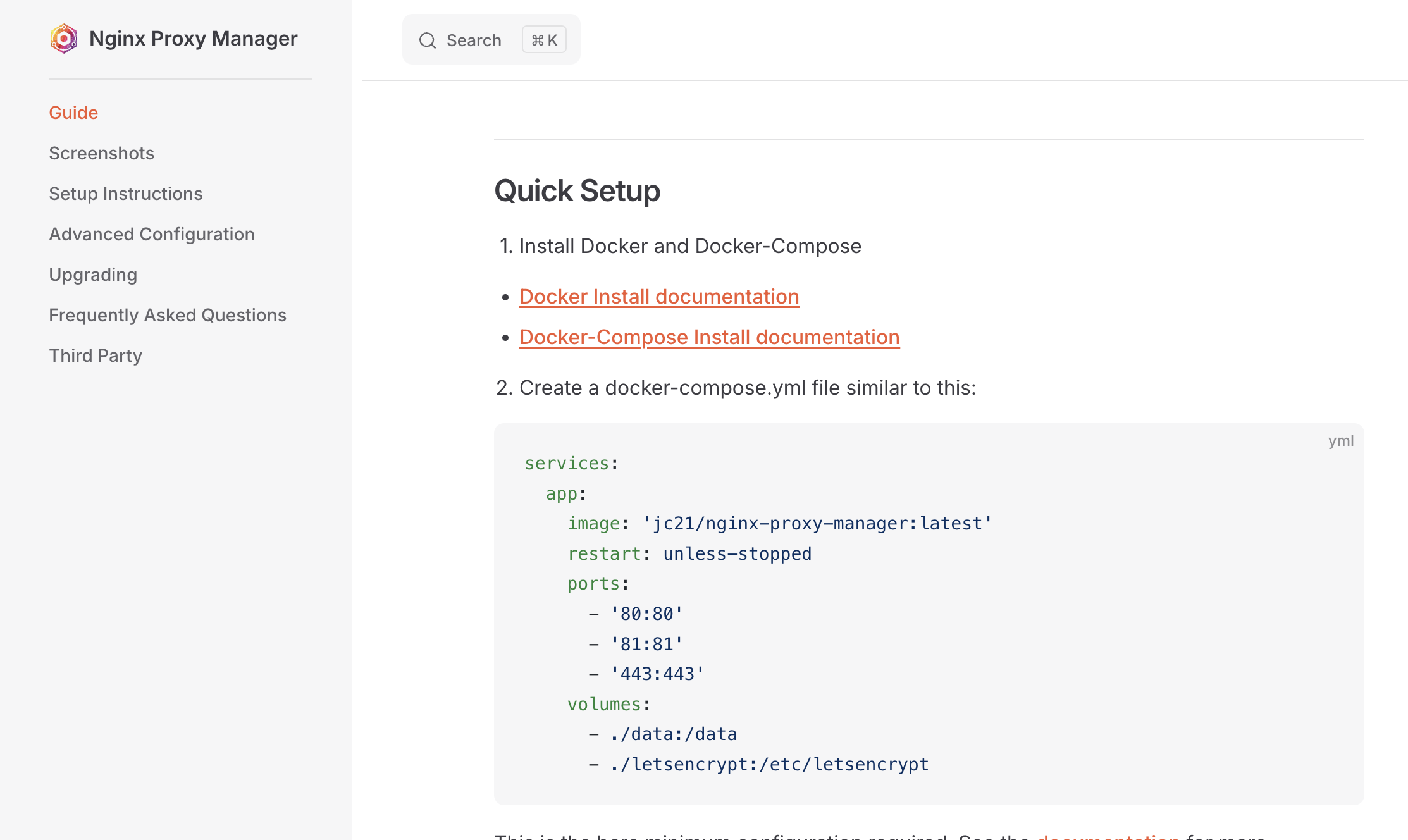
* Cloudflare: map domain to ip

Connect domain to Cloudflare🡪 click on add domain   


In Namecheap domain  




* NginX port 81 : manage resources  
    
    
  install Nginx from the the Nginx Webpage



services:

app:

image: 'jc21/nginx-proxy-manager:latest'

restart: unless-stopped

ports:

- '80:80'

- '81:81'

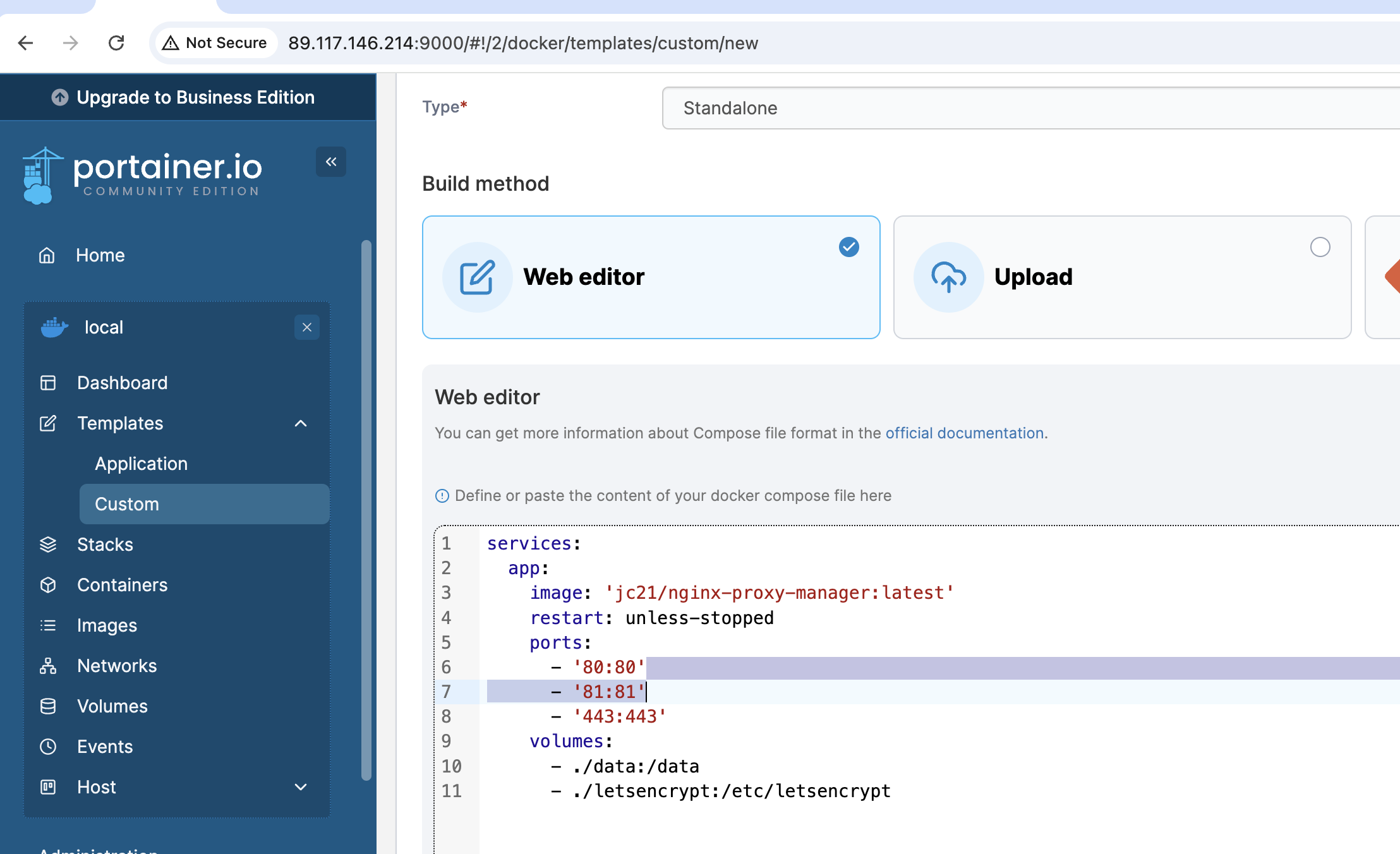
- '443:443'

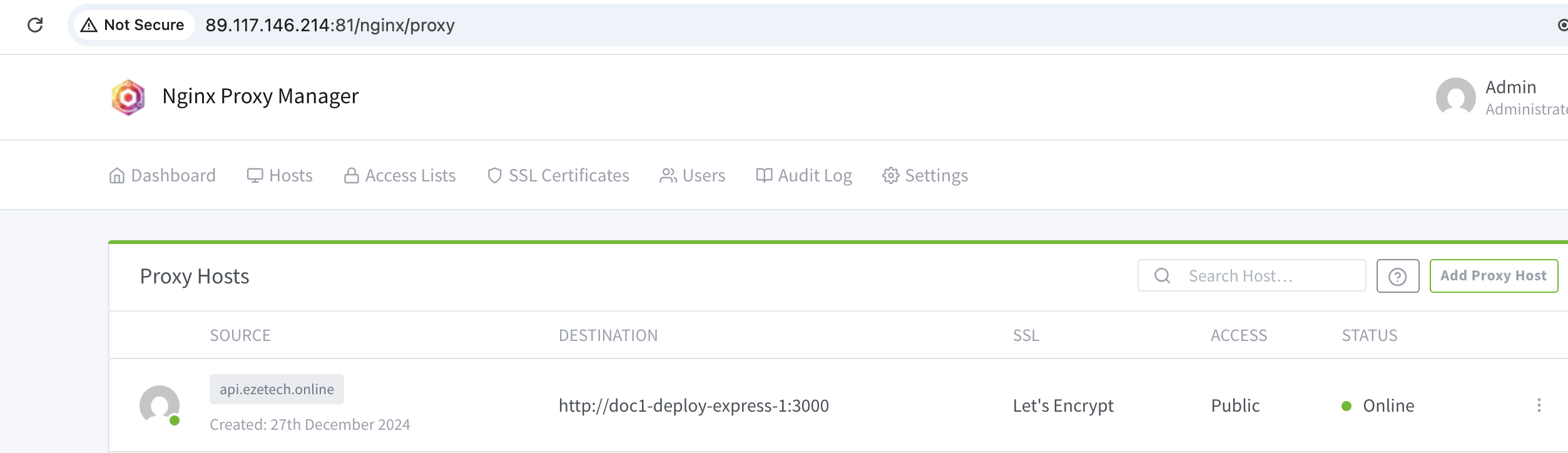
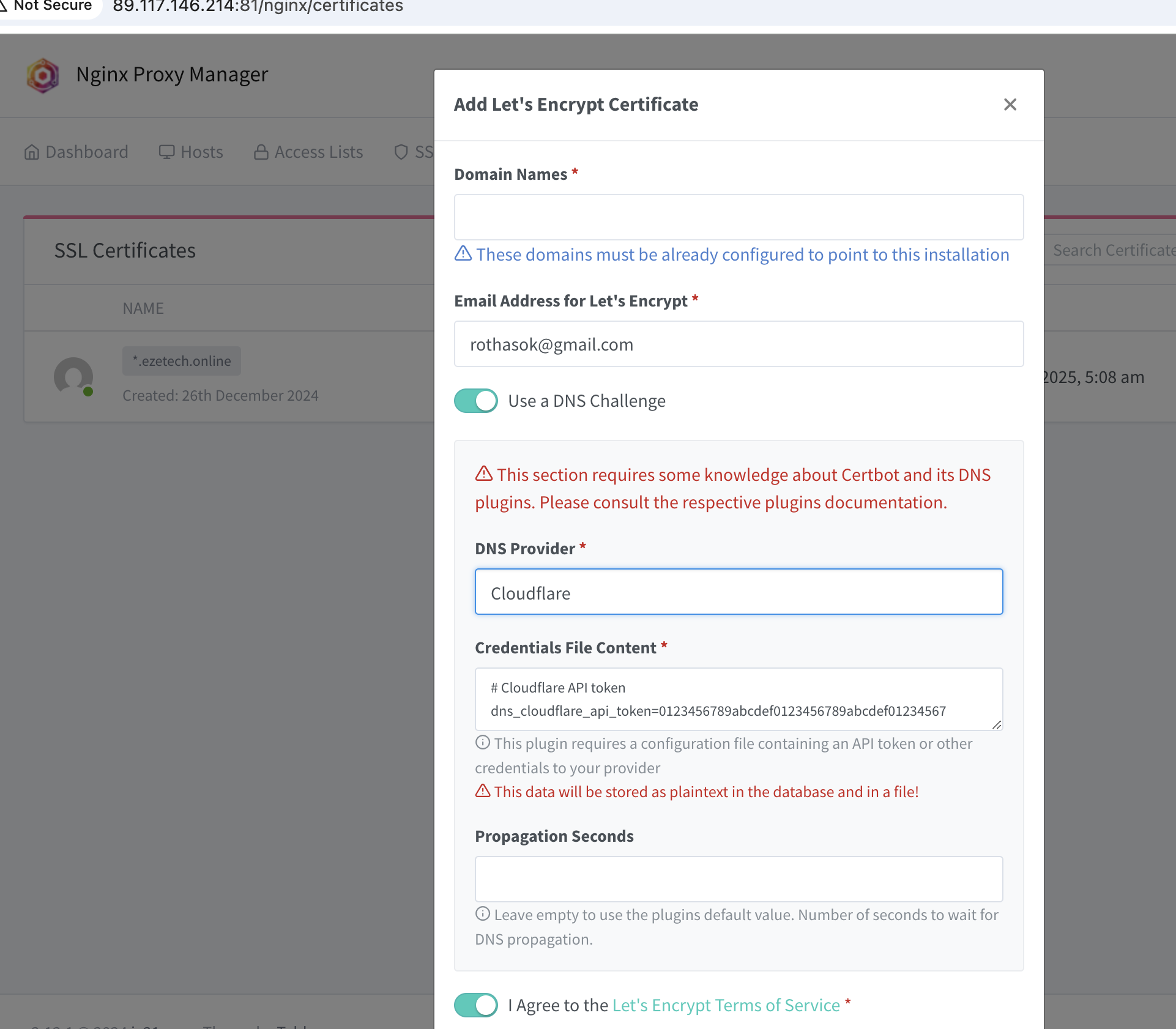
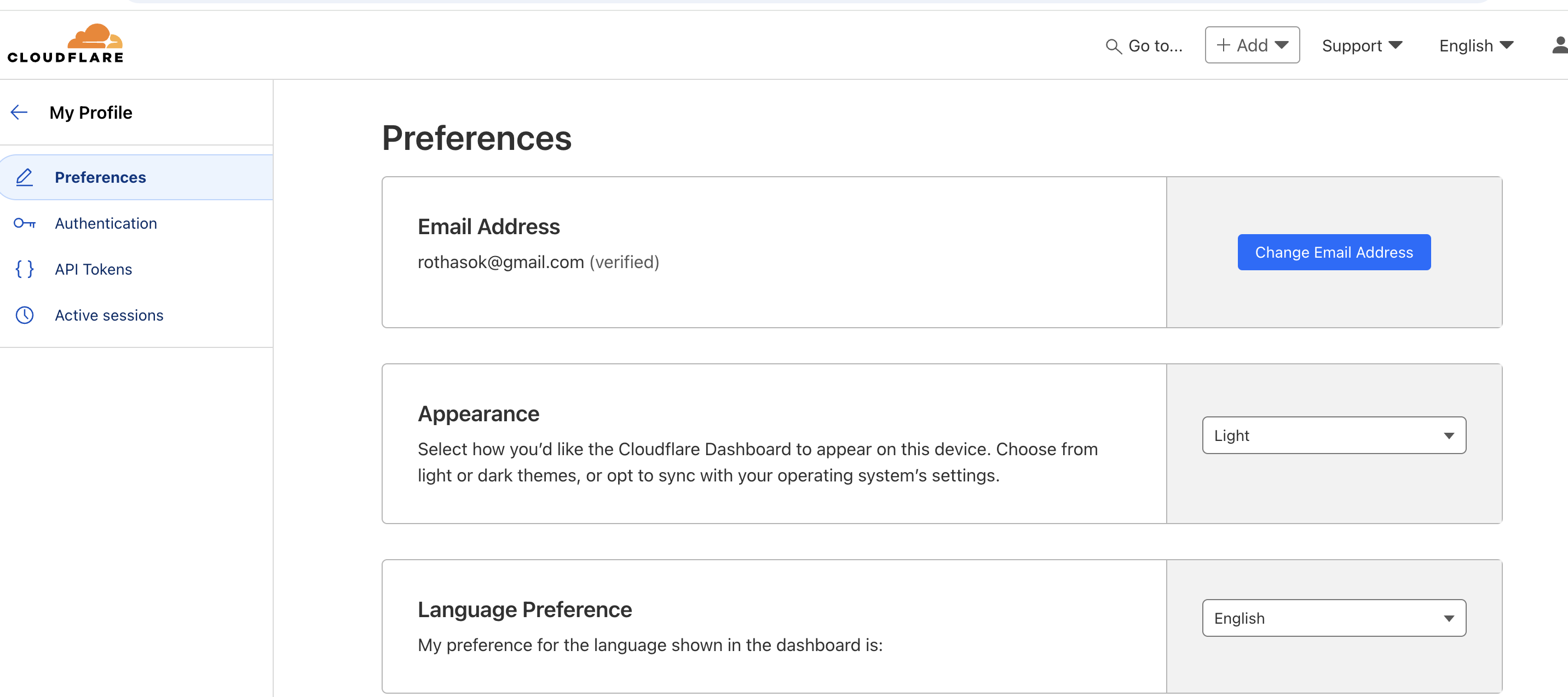
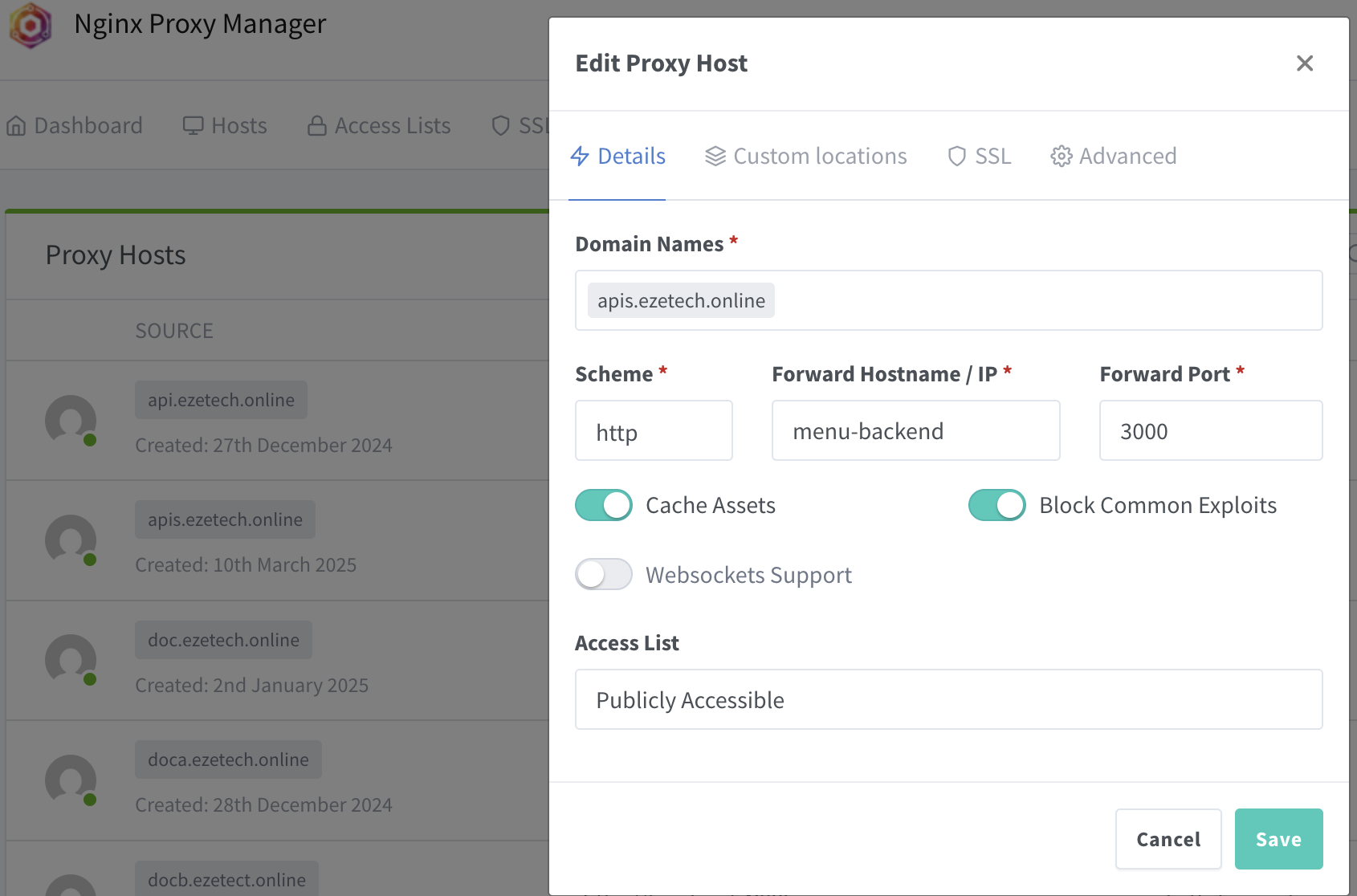
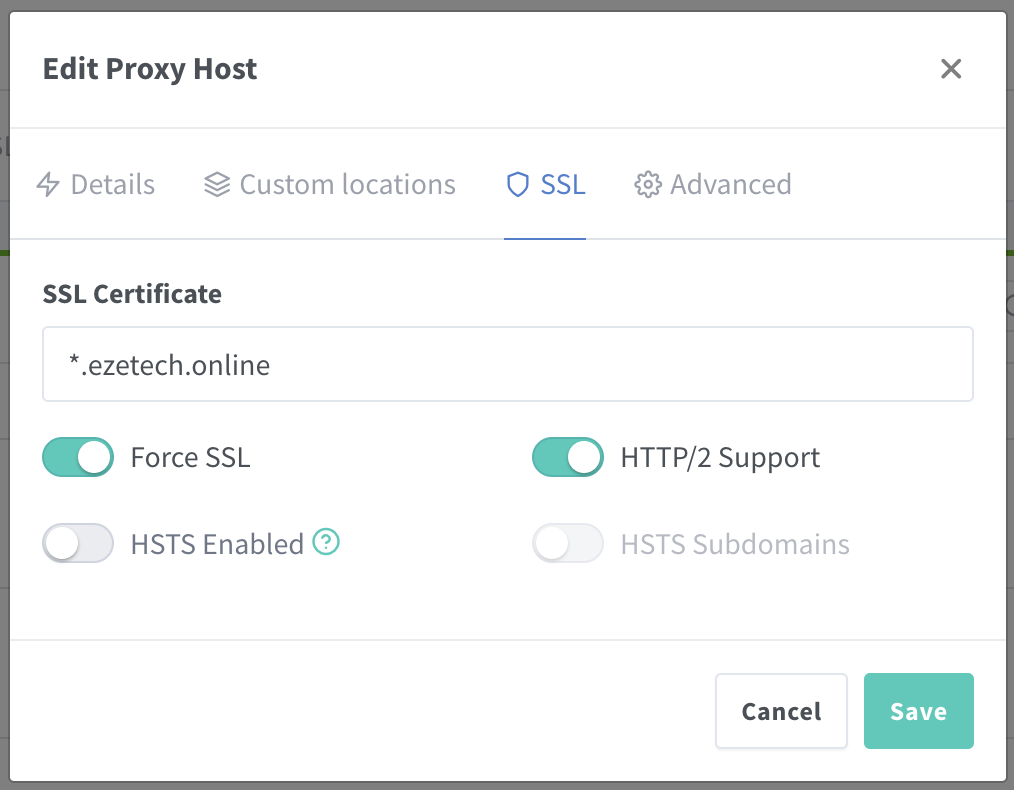
volumes:

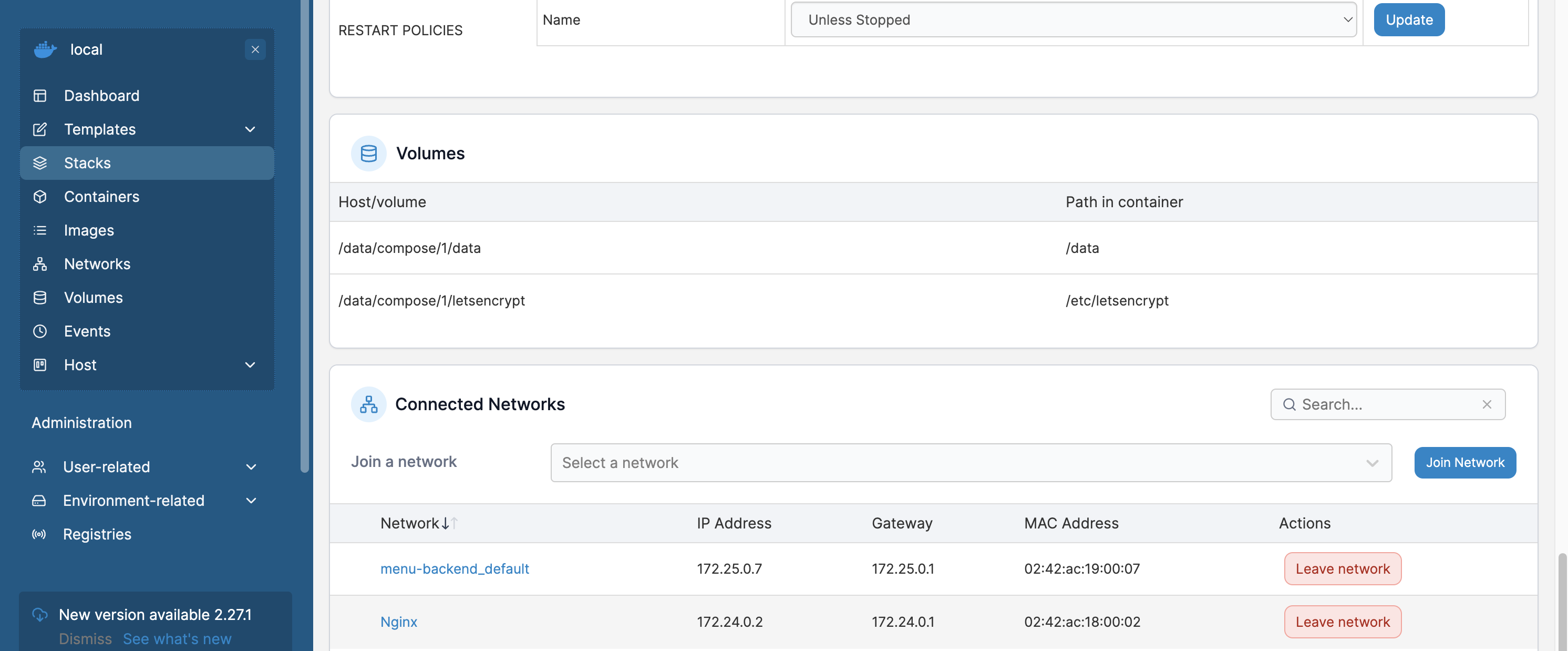
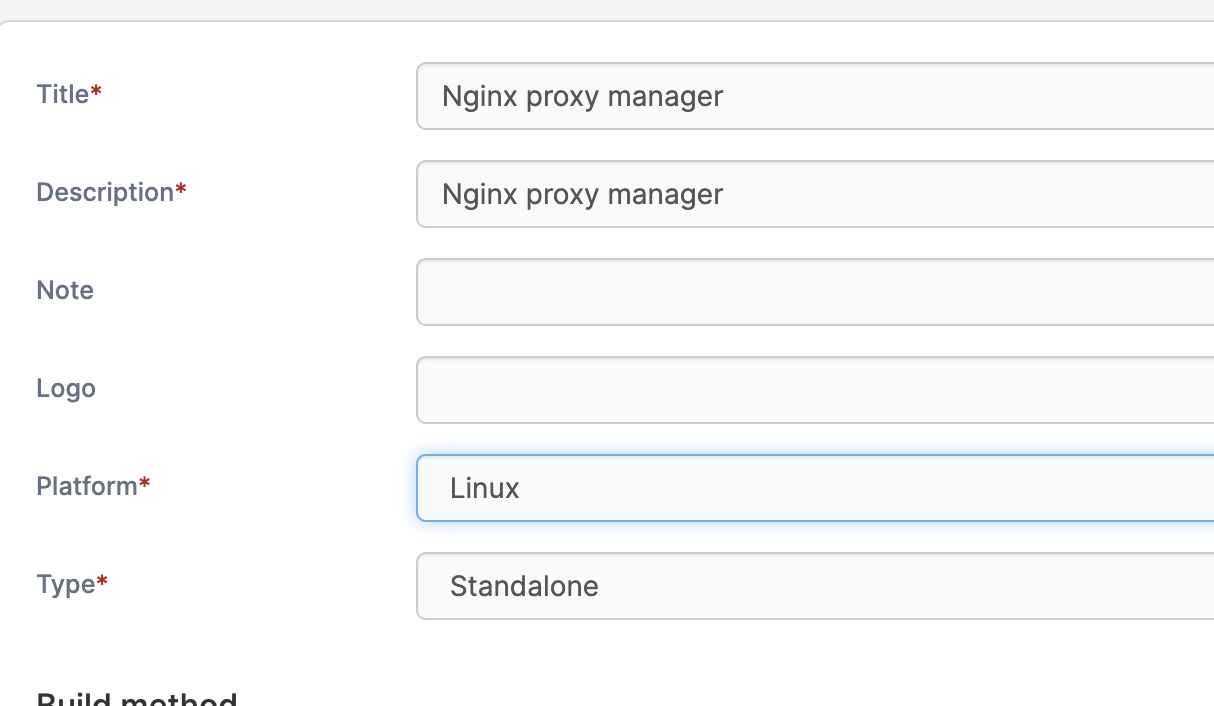
- ./data:/data

- ./letsencrypt:/etc/letsencrypt

And past to



* 
* In case new SSL in proxy  
    
    
  take token from cloudflare  
  
*   
  - forward Hostname/Ip\* : Container’s name of express (menu-backend)
* 

Join the same network for Inginx and Container (Portainer)  
  
  
  
Testing: https://apis.ezetech.online/v1/owner/search?page=1&limit=5&query='owner'  
  
  
  
\* CI/CD

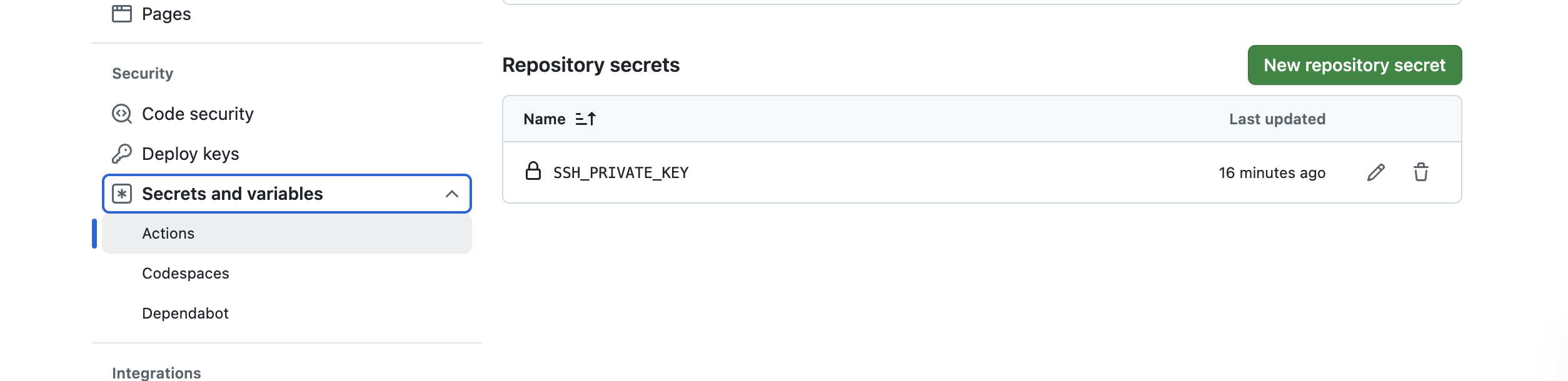
Generate Key and copy to server

ssh-keygen -t rsa -b 4096 -C "rothasok@gmail.com" -f ~/.ssh/contabo\_server\_key

ssh-copy-id -i ~/.ssh/ contabo\_server\_key.pub [root@89.117.146.214](mailto:root@89.117.146.214)

ssh -i ~/.ssh/contabo\_server\_key [root@89.117.146.214](mailto:root@89.117.146.214) (login)

Add Scret to Github Respository ( copy private key to git action secret)



Create folder: .github/workflows/backend-ci-cd.yml  
  
 name: Deploy to Contabo Server

on:

push:

branches:

- main # Adjust to your branch name

jobs:

deploy:

runs-on: ubuntu-latest

steps:

- name: Checkout code

uses: actions/checkout@v3

- name: Setup SSH Key

run: |

mkdir -p ~/.ssh

echo "${{ secrets.SSH\_PRIVATE\_KEY }}" > ~/.ssh/id\_rsa

chmod 600 ~/.ssh/id\_rsa

ssh-keyscan -H 89.117.146.214 >> ~/.ssh/known\_hosts

- name: Deploy to Server

run: |

ssh -i ~/.ssh/id\_rsa root@89.117.146.214 "cd /root/menu-backend && git pull && docker compose up -d --build"  
  
  
Run Comand

git add .github/workflows/deploy.yml

git commit -m "Fix: Added event trigger to GitHub Actions"

git push origin main