Usando Nginx Proxy Manager

Problemática. Múltiples servicios sobre un mismo puerto

Imaginemos que tú tienes un servicio o página web la cual quieres exponer a internet, los navegadores al intentar acceder a una página por el puerto 80 (puerto inseguro) o 443 (puerto para páginas con certificado SSL) y esta es la razón de porque no vemos url con el siguiente formato **https://www.google.com:*443,*** entonces montaremos nuestra página sobre el puerto 80 o 443

Imaginemos que tienes 2 aplicaciones y ambas tiene certificado SSL, no podrás montar ambas paginas en el puerto 443 por lo que tendrás que abrir otro puerto del modem y solo podrás llamarlo con una url como la siguiente **https://www.google.com:*444*** ya que el modem solo puede asignar un puerto de un dispositivo a un puerto del modem.

Solución 3. Proxy Inverso

Para montar múltiples aplicaciones sobre un puerto podemos usar un servicio de proxy inverso como el de Nginx Proxy Manager, este servicio se basa en tener un dominio con sub dominós, por ejemplo tu dominio puede ser **duckdns.org** y para agregar un subdominio agregarías un nombre seguido de un punto y tu dominio, **myserver.duckdns.org.**

Todos los subdominio apuntan realmente a la misma ip publica pero Ng Proxy Manager puede redirigir a diferentes aplicaciones mediante los diferentes nombres de subdomino. esto te pemite tener montado multiples servicios sobre un mimsmo puerto.

Instalación

Para el proceso de instalación de Nginx Proxy Manager te recomiendo que uses el siguiente Docker compose docker compose y si no tienes idea de como ejecutar el stack puedes usar la Introducción a los comandos de uso común y consideraciones.

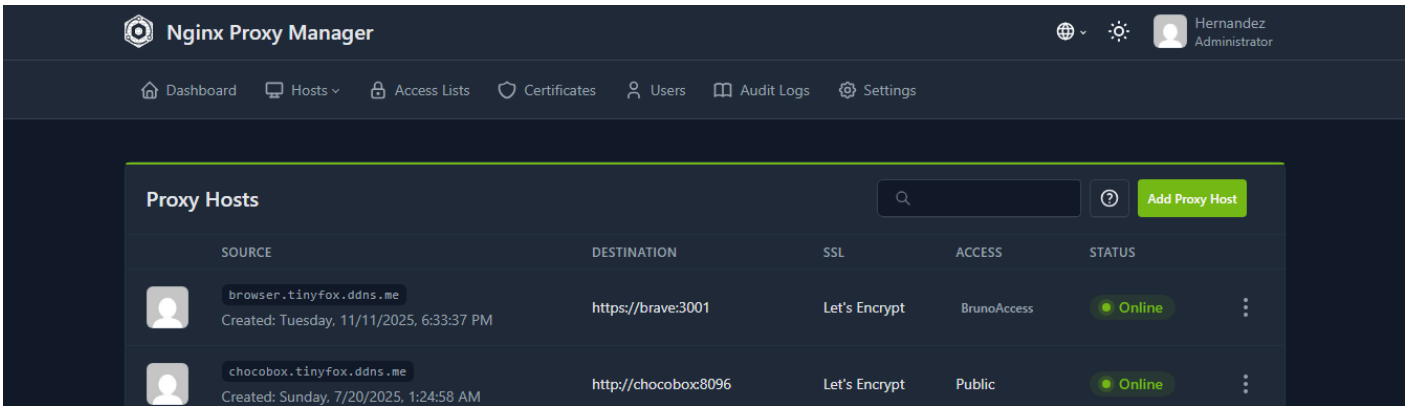
Antes de continuar necesitaremos un servicio el cual expondremos a internet, en este caso usaremos FileBrowser como ejemplo, te invito a que generes una instancia de este servicio o cualquier otro listado en Servicios Con Docker Compose.

Además tendrás que ingresar al panel de control de Nginx Proxy Manager, para esto, estando en la misma red que el servidor ingresaremos a **ip-del-servidor:81**

Posteriormente para el primer ingreso usaremos las siguientes credenciales

Usuario: **admin@example.com**
Password: **changeme**

Al ingresar te pedirá cambiar la contraseña y el correo, lo haces (no tienen porque ser reales pero si debe seguir formato de correo electrónico) y luego desde la pestaña de **Dashboard** haces click sobre el botón de **Proxy Host** y posteriormente en **Add Proxy Host**



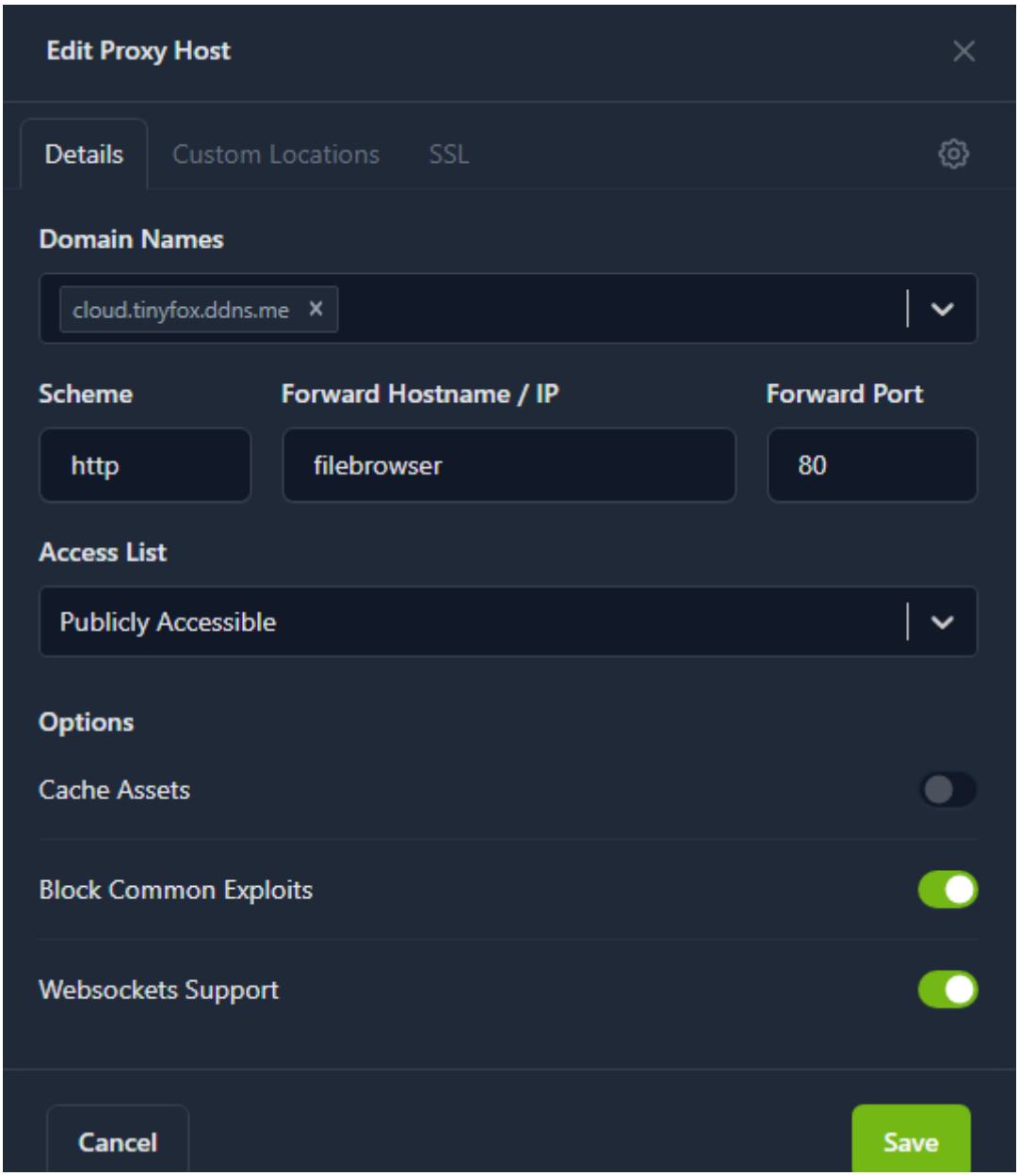
Ejemplos de uso del Proxy Inverso

Aquí vemos como exponer una instancia del servicio filebrowser, su puerto interno lo encontraras en la documentacion oficial aunque en muchas ocasiones podrás encontrar un ejemplo en los repositorios de Docker Hub Container Image Library, Ahí buscaras los Docker compose y la sección de puertos.

Veamos el siguiente fragmente de un docker-compose

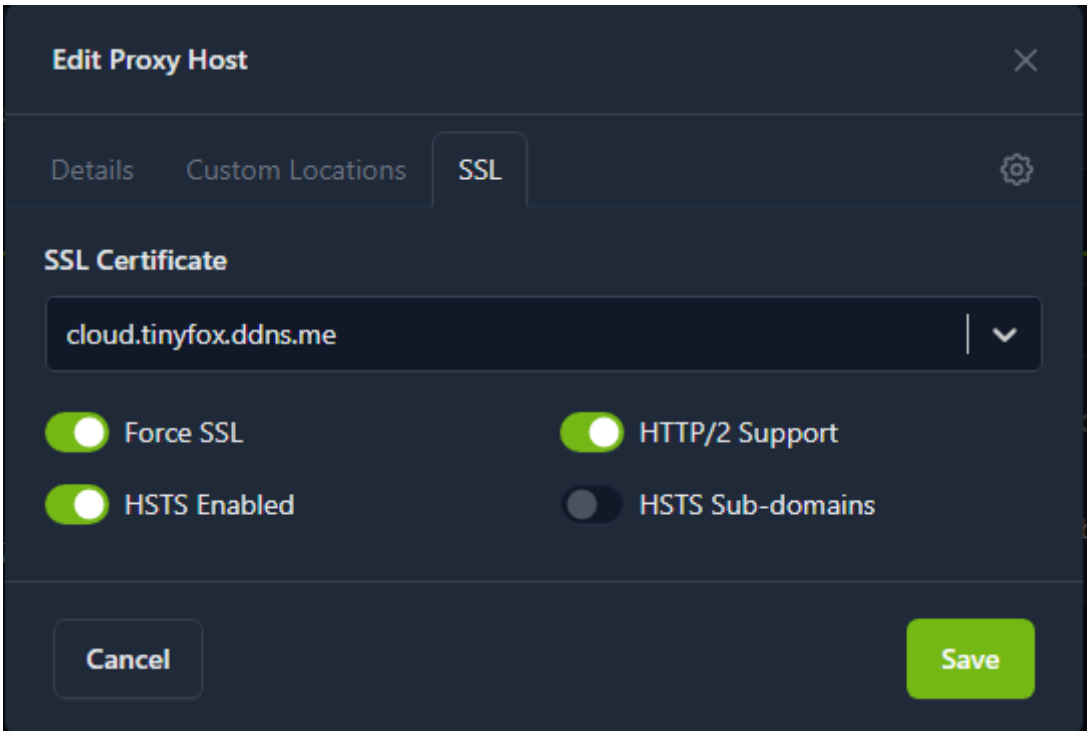
```
services:
  filebrowser:
    image: filebrowser/filebrowser:latest
    container_name: filebrowser
    ports:
      - 8009:80
```

En nuestro caso no necesitamos exponer este puerto y comentaremos las líneas de port y la redirección del puerto. Nos quedaremos con el dato de que el puerto interno (señalado siempre en el lado derecho) es el puerto 80 y tomaremos el nombre del contenedor y los ingresaremos de la siguiente manera.

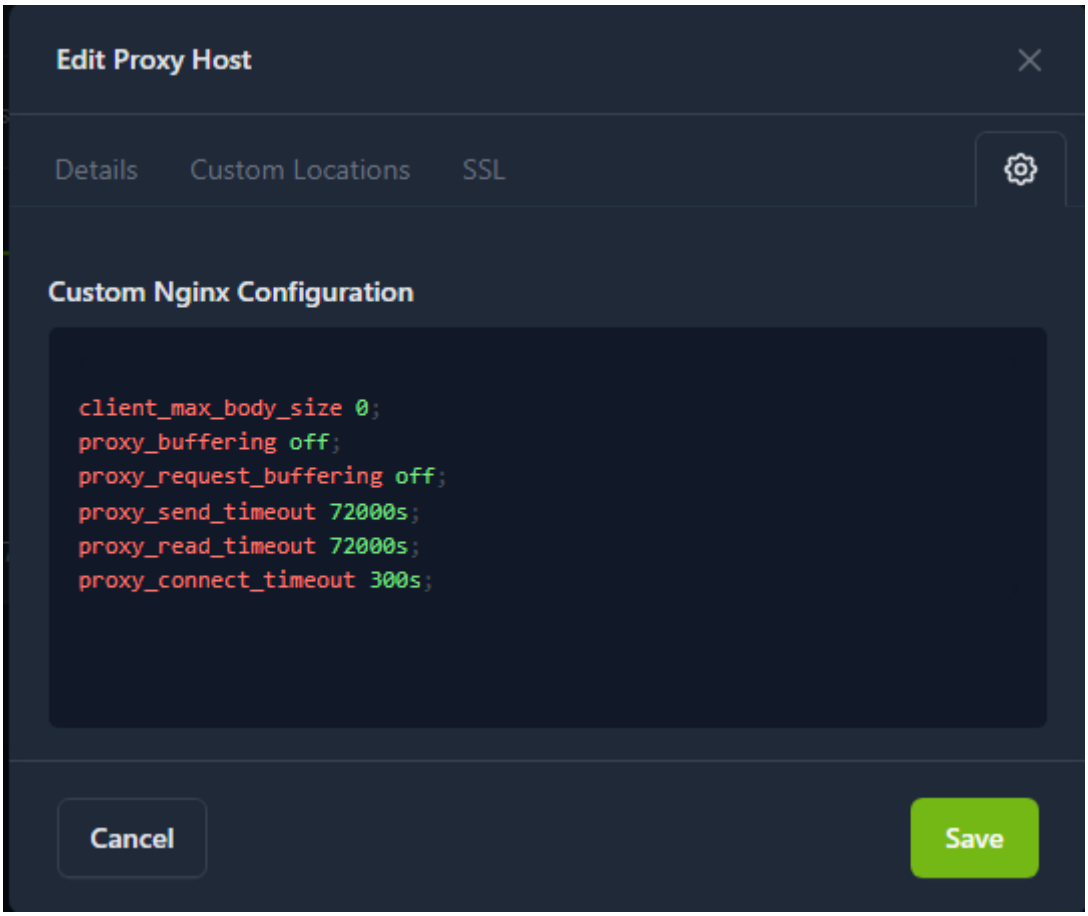


En la sección de Domain Names usaras el dominio generado por tu proveedor DDNS visto anteriormente y un subdominio a elegir pero que no se repita, en mi caso fue cloud y el dominio es tinyfox.ddns.me.

Posteriormente en la pestaña SSL generamos un certificado y habilitamos las siguientes casillas para forzar la redirección a el puerto 443



Por ultimo, al ser una aplicación que usa WevDav para transmisión de archivos seleccionamos el engranaje para ir a la pestaña de Custom Nginx Configuration y agregamos el contenido mencionado previamente en la documentación de [Nginx Proxy Manager]



Pruebas

Para probar que todo quedo bien ingresa a la página que escribiste en la sección de Domain Names y veras el servicio que creaste o no.

Extras

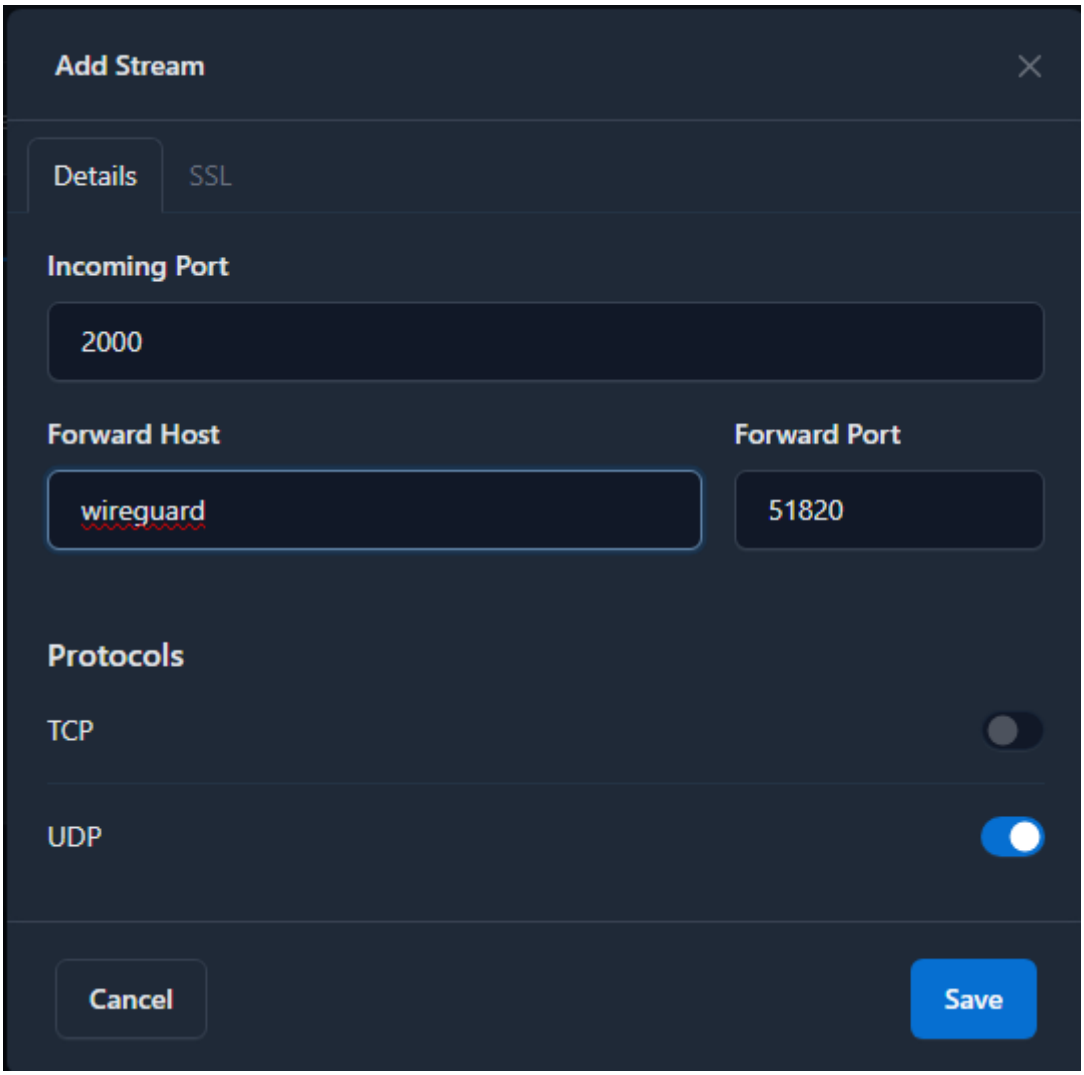
Recuerda que Nginx Proxy Manager es para servicios HTTP, si quieres exponer un servicio que no sea de este estilo tendrás que abrir otros puertos del modem que apunten al puerto del servidor o puedes usar Nginx Proxy Manager para redirigir este flujo a otros puertos y abrir estos de igual forma en tu modem; Esto solo por si quisieras cambiar el puerto por defecto de una aplicación y esta no te lo permitiera.

Si decides redirigir puertos con Nginx Proxy Manager tendrás que alterar el Docker compose de Nginx Proxy Manager y agregar los puertos internos y externos correspondientes en su docker compose, además de que si el servicio usa UDP tendrás que señalarlo en el docker compose de forma explícita ya que docker compose por defecto infiere que todos los puertos son TCP.

A continuación, veremos un ejemplo para redireccionar el puerto de wireguard mediante Ngix Proxy Manager, Si bien puedes redirigir puerto de la misma red que el servidor, ya que es otro servicio de docker recomiendo no exponer el puerto de wireguard a la red del servidor, es mejor usar la red creada en docker para montar servicios en Ngix Proxy Manager, en mi ejemplo de repositorio de docker se llamada sky_net.

Como puedes ver en este ejemplo estoy redirigiendo el puerto por defecto de wireguard (51820) al puerto 2000 y abro este puerto en Nginx Proxy Manager a con el mismo puerto interno y externo del contenedor para evitar confusiones al abrir el puerto del modem.

```
services:
  nginx-proxy:
    image: 'jc21/nginx-proxy-manager:latest'
    container_name: nginx-proxy-manager
    ports:
      - '80:80' # Public HTTP Port
      - '443:443' # Public HTTPS Port
      - '81:81' # Admin Web
      - '2000:2000/UDP' # Admin Web
```



Revisión #16
Creado 2025-11-25 18:37:01 UTC por Bruno
Actualizado 2025-12-03 01:34:38 UTC por Bruno