

Access Your Self Hosted Services WITHOUT Port Forwarding

In this video we're going to take a look at setting up remote access to your self-hosted applications with CloudFlare Tunnels. Using CloudFlare Tunnels allows us to create an encrypted connection between our server(s) and CloudFlare without the need to forward any ports.

- [Access Your Self Hosted Services WITHOUT Port Forwarding](#)

Access Your Self Hosted Services WITHOUT Port Forwarding

For more than a year we've been talking about using a reverse proxy to access your self-hosted apps via a domain name.

In this video we're going to take a look at setting up remote access to your self-hosted applications with CloudFlare Tunnels. Using CloudFlare Tunnels allows us to create an encrypted connection between our server(s) and CloudFlare without the need to forward any ports.

Prerequisites

We only need 2 things to get things set up (aside from our Docker server and self-hosted apps):

1. A domain name from your favorite registrar
2. A CloudFlare account

Resource links:

- ☐ <https://dash.teams.cloudflare.com/>
- ☐ <https://hub.docker.com/r/cloudflare/cloudflared>
- ☐ <https://dbt3ch.com/books/access-your-self-hosted-services-without-port-forwarding/page/cloudflare-tunnels>

Get early, ad-free access to new content by becoming a Patron or signing up for the members' only website!

- ☐ <https://www.patreon.com/dbtech>
- ☐ <https://dbtech.fans/>

Here is the docker-compose you can use to deploy this in Portainer:

```
version: "3"
services:
  cloudflared:
    command:
      - tunnel
      - --no-autoupdate
      - run
      - --token
```

```
- <your_token_here>  
container_name: cloudflared  
image: cloudflare/cloudflared:latest  
restart: always
```

There is probably a cleaner way to do this, but this works.