AI-03016 Cloudflare Tailscale Wireguard pfSense

Owner	Ⓡ Rigel Arcayan
Tags	Project
Created time	@September 20, 2025 8:59 AM

What the history of cloudfare and tailscale? Does cloudflare have a free tier account? Regarding Tailscale, I now have an account, do I have now a poor man's public domain? Do I need to purchase a domain name for my public website which is behind cgnat? Is cloudflare a good option or there is another vendor? Create a step procedure to expose my Ubuntu servers in my home behind a cgnat pldt ip address. I would like my wordpress, OpenEMR, openproject, osTicket be exposed to public internet.

Cloudflare and Tailscale are two companies that have become critical components of modern internet infrastructure, but they have distinct histories and purposes.

Cloudflare 🌐

Founded in 2009 by Matthew Prince, Lee Holloway, and Michelle Zatlyn, Cloudflare started as a service to fight email spam and protect websites from cyber threats, particularly DDoS (Distributed Denial-of-Service) attacks.1 Its foundational product, Project Honey Pot, was an early effort to track spammers.2 The company's name was a suggestion from a friend, describing their service as a "firewall in the cloud."3 Cloudflare officially launched in 2010 at TechCrunch Disrupt and quickly gained traction by offering a free tier for website security and performance.4 It has since grown into a major infrastructure provider, offering a wide range of services from content delivery network (CDN) and DNS to security solutions for businesses and individuals.5

Tailscale 🦎

Tailscale was founded in 2019 by former Google engineers Avery Pennarun, David Crawshaw, David Carney, and Brad Fitzpatrick.6 The company's name is inspired by the Google research paper "The Tail at Scale," which explores the performance challenges of large-scale distributed systems.7 Tailscale's core product is a software-defined mesh VPN that simplifies network access.8 It is built on the open-source WireGuard protocol and is designed to create secure, private networks among devices, regardless of their location.9 Tailscale's approach is to provide a "zero config" VPN as a service, making it easy for users to connect their devices and access resources from anywhere.10

Cloudflare Free Tier

Yes, Cloudflare offers a very generous free tier that includes its core services for websites.11 This free plan provides features like unmetered DDoS protection, a CDN (Content Delivery Network), and a free SSL certificate. It's a popular choice for personal websites, blogs, and small projects. The free tier also includes a limited number of Cloudflare Tunnels, which is the service you'd use to expose your local server.12

Tailscale and Public Domains

Your Tailscale account does not give you a "poor man's public domain." While Tailscale assigns a unique, static IP address (from the `100.64.0.0/10` range) to each of your connected devices, these addresses are **private and not publicly routable** on the internet.13 They only work within your private Tailscale network, which consists of the devices you have authenticated and connected. Think of it as your own private local network that extends across the internet. Therefore, other people on the internet cannot use your Tailscale IP to access your server unless they are also on your Tailscale network.

To make your website public, you will need a public domain name.

Exposing a Server Behind CGNAT

Exposing a server from your home behind a Carrier-Grade NAT (CGNAT) requires a method that bypasses the limitations of not having a unique public IP address.14 Cloudflare Tunnel is an excellent solution for this.15 It works by creating a secure, outbound connection from your server to Cloudflare's network, which then

proxies public internet requests to your server. This bypasses CGNAT entirely as no inbound connections are required. Another option is a self-hosted solution using a low-cost VPS (Virtual Private Server) as a jump point, but Cloudflare is generally the simplest and most reliable for most users, especially since its free tier includes the necessary features. Other vendors like Pinggy.io also offer similar services.16

Step-by-Step Procedure to Expose Your Ubuntu Servers

1. **Purchase a Domain Name:** Buy a custom domain name from a registrar (e.g., GoDaddy, Namecheap). You need a public domain to use with Cloudflare Tunnel.
2. **Add Your Domain to Cloudflare:**
   - Sign up for a free Cloudflare account.17
   - Add your domain to your Cloudflare dashboard and follow the instructions to change your nameservers at your domain registrar to point to Cloudflare. This is a crucial step that delegates DNS control to Cloudflare.
3. **Install `cloudflared` on Your Ubuntu Server:**
   - Cloudflare Tunnel requires a small daemon, `cloudflared`, to run on your server.
   - On your Ubuntu server, open a terminal and run the following commands to install `cloudflared`:Bash
     `sudo apt-get update && sudo apt-get install cloudflared`
   - Authenticate `cloudflared` with your Cloudflare account:Bash
     `cloudflared tunnel login`

     This command will provide a URL for you to log in to your Cloudflare account and authorize the tunnel.
4. **Create and Configure the Tunnel:**
   - Create a named tunnel:Bash
     `cloudflared tunnel create <TUNNEL_NAME>`

     For example, `cloudflared tunnel create my-home-server-tunnel`. This will create the tunnel and give you a unique ID.

- Create a configuration file for the tunnel.18 This file will tell Cloudflare which local services to expose. The file is typically located at `~/.cloudflared/config.yml`.
- Here is an example `config.yml` that exposes your services:YAML

```
tunnel: <TUNNEL_ID>
credentials-file: /root/.cloudflared/<TUNNEL_ID>.json

ingress:
- hostname: wordpress.yourdomain.com
  service: http://localhost:80
- hostname: openemr.yourdomain.com
  service: http://localhost:81
- hostname: openproject.yourdomain.com
  service: http://localhost:82
- hostname: osticket.yourdomain.com
  service: http://localhost:83
- service: http_status:404
```

  - **Note:** Replace `<TUNNEL_ID>` with the ID from the previous step and `yourdomain.com` with your actual domain. Also, ensure your local applications (WordPress, etc.) are running on the specified ports (`80`, `81`, `82`, `83`) on your server.

5. **Create DNS Records and Run the Tunnel:**
   - Tell Cloudflare to route traffic from your hostnames to your tunnel.Bash

     ```
     cloudflared tunnel route dns <TUNNEL_NAME> <HOSTNAME>
     ```

     Repeat this command for each of your hostnames (e.g., `wordpress.yourdomain.com`, `openemr.yourdomain.com`, etc.).
   - Run the tunnel as a system service so it starts automatically:Bash

     ```
     sudo cloudflared tunnel run <TUNNEL_NAME>
     ```
   - The `cloudflared` service will now start, establish a secure connection to Cloudflare, and make your services publicly accessible at the hostnames you configured.

How To Self Host Behind CGNAT With Wireguard and pfsense

This video provides a more technical, self-hosted approach to bypassing CGNAT, offering an alternative to Cloudflare Tunnel.

WireGuard is a modern, open-source VPN protocol known for its simplicity, speed, and strong cryptography. Developed by **Jason A. Donenfeld**, its design philosophy focuses on a small codebase to reduce the attack surface, making it easier to audit for security vulnerabilities. The protocol passes traffic over UDP and uses a simplified public/private key exchange, similar to SSH, for secure connections. Initial development began around 2015, and it was first released as a Linux kernel module. It has since expanded to be cross-platform, with implementations for Windows, macOS, BSD, iOS, and Android.

---

pfSense is a free, open-source firewall and router software distribution based on FreeBSD. It can be installed on a physical computer or a virtual machine to create a dedicated network appliance. The project began in **2004** as a fork of the m0n0wall project, created by **Chris Buechler and Scott Ullrich**. Its name is derived from the **packet filtering tool PF** (Packet Filter) used in the operating system. The first release was in 2006. The company behind pfSense is **Netgate**, which provides commercial support and sells hardware appliances pre-loaded with the software.

Self-Hosting Behind CGNAT with WireGuard and pfSense

CGNAT (Carrier-Grade Network Address Translation) is a technique used by internet service providers (ISPs) to conserve IPv4 addresses. It places multiple customers behind a single public IPv4 address, which prevents you from directly port forwarding to your local network. This makes it impossible to host a server that can be accessed from the public internet using your IPv4 address alone.

To get around CGNAT for self-hosting with WireGuard and pfSense, you need an intermediary server with a public IP address. This is typically a Virtual Private Server (VPS) hosted in a data center. The general idea is to set up a WireGuard tunnel between your pfSense firewall (the client) and the VPS (the server). All traffic from the outside world is then directed to the VPS, which forwards it to your pfSense instance over the encrypted WireGuard tunnel.

Step-by-Step Procedure

1. Set up a VPS:

- Purchase a low-cost VPS from a cloud provider. A basic plan is usually sufficient.
- Install a clean, lightweight Linux distribution like Debian or Ubuntu on the VPS.

2. Configure WireGuard on the VPS (Server):

- Install the WireGuard package on the VPS.
- Generate a public/private key pair for the server. You can use the `wg genkey | tee privatekey | wg pubkey > publickey` command.
- Create a WireGuard configuration file (e.g., `/etc/wireguard/wg0.conf`).
- In the configuration file, set the **server's private key**, a **listen port** (e.g., 51820), and a **tunnel address** (e.g., `10.0.0.1/24`).
- Enable IP forwarding and configure firewall rules (using `iptables` or `ufw`) on the VPS to allow traffic to pass through the WireGuard tunnel and to your network.

3. Configure WireGuard on pfSense (Client):

- In the pfSense web interface, go to **System > Package Manager > Available Packages** and install the **WireGuard** package.
- Go to **VPN > WireGuard > Tunnels** and add a new tunnel.
- Give it a description, a **listen port** (it can be the same as the VPS), and generate a new key pair for the pfSense side.
- Add a **peer**. This is where you will add the details for your VPS.
- Enter the **VPS's public key**, the **VPS's public IP address** as the endpoint, and the **listen port** you chose for the VPS.
- In the **Allowed IPs** field, enter the WireGuard tunnel network (e.g., `10.0.0.0/24`) and any public IP addresses you want to route through the tunnel.

4. Assign the WireGuard Interface and Configure Firewall Rules:

- Go to **Interfaces > Assignments** and add the new WireGuard tunnel as an interface.
- Enable the new interface and give it a name (e.g., `WG0`).

- Go to **Firewall > Rules > WG0** and add a rule to pass traffic from your WireGuard network (`10.0.0.0/24`) to your local network.

5. Forward Ports on the VPS:

- Configure port forwarding on your VPS to direct traffic from a public port to the specific internal IP and port of the service you want to expose on your home network. This is done using `iptables` rules on the VPS. For example, to expose a web server on port 8080 on your home network, you would forward traffic from the VPS's public port 80 to the WireGuard tunnel address of your pfSense (`10.0.0.2:8080`).

Setting up this configuration allows you to access services on your home network remotely by connecting to the public IP address of your VPS. The VPS then securely routes the traffic to your pfSense firewall over the WireGuard tunnel, effectively bypassing the CGNAT.

How To Self Host Behind CGNAT With Wireguard and pfsense

This video is a detailed guide on setting up a self-hosted solution using WireGuard and pfSense to bypass CGNAT.