OPNSense

- HAProxy Simple Configuration for local webserver
- Adguard Home communications error to 127.0.0.1#53: connection refused
- Disable IPv6 in OPNSense
- How to enable the HAProxy statistics page in OPNsense
- HowTo Restore a Google Drive backup file in OPNsense
- Run frequent Speedtest in OPNsense
- Voorgestelde waarden voor Advanced tab in Unbound
- Setup os-ddclient for when external IP address changes

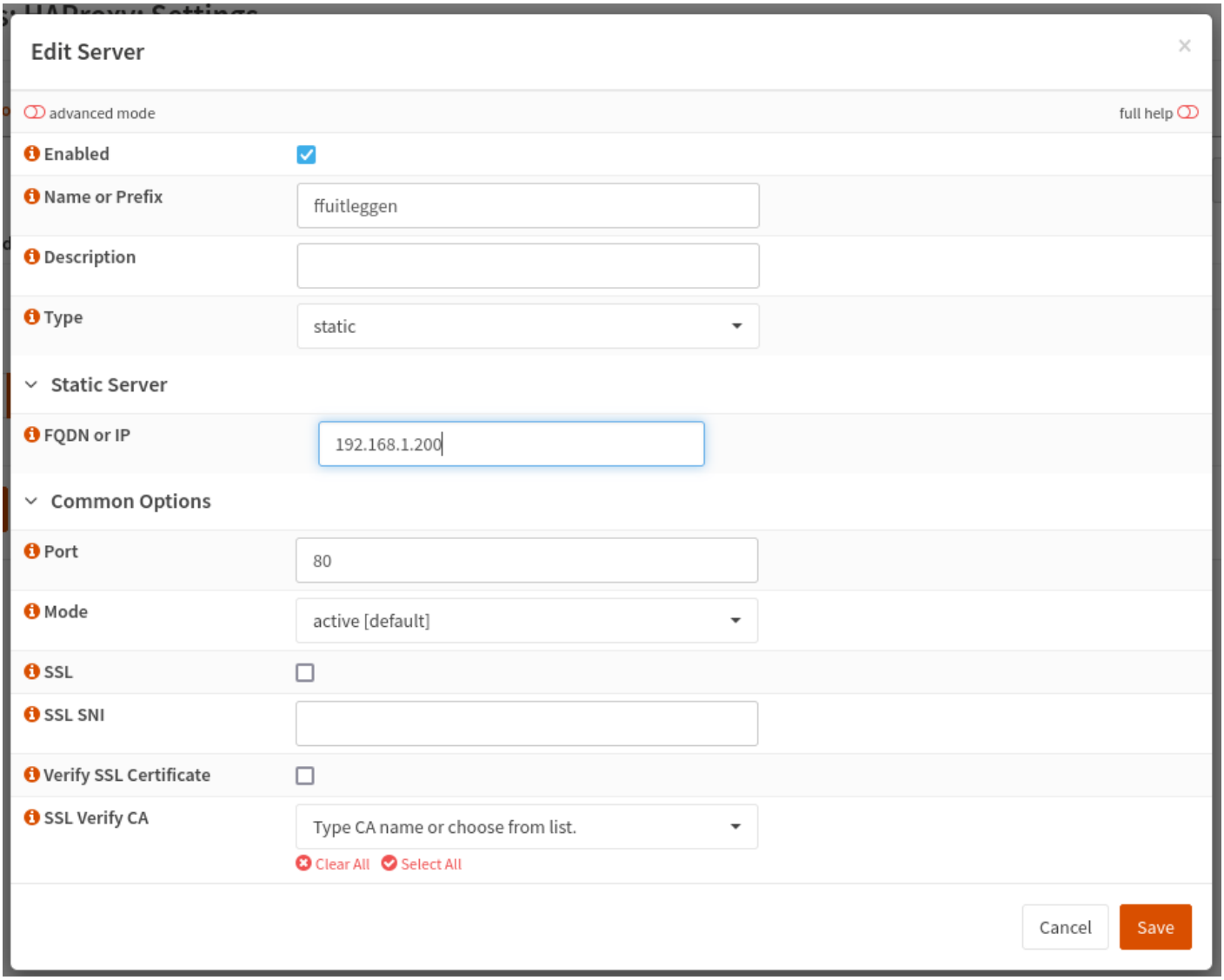
HAProxy Simple Configuration for local webserver

Parameters for this setup

Local webserver is on ip address 192.168.1.200 and uses port 80

Step 1: Define a Real server

- Name: **anything** you like to recognize the webserver
- IP address: The IP address of the internal webserver e.g. **192.168.1.200**
- Port: **80**
- SSL: **disable**



1. Define a Virtual service->Backend Pool
   - Name: **anything** you like
   - Servers: **The server you defined in the step 1** (remember to press TAB after entering server name)

Edit Backend Pool

advanced mode | full help

Enabled	☑
Name	ffuitlleggen
Description	
Mode	HTTP (Layer 7) [default]
Balancing Algorithm	Source-IP Hash [default]
Servers	ffuitleggen × Clear All Copy Paste Text
FastCGI Application	None
Resolver Options	 Clear All Copy Paste Text
Enable Health Checking	☑

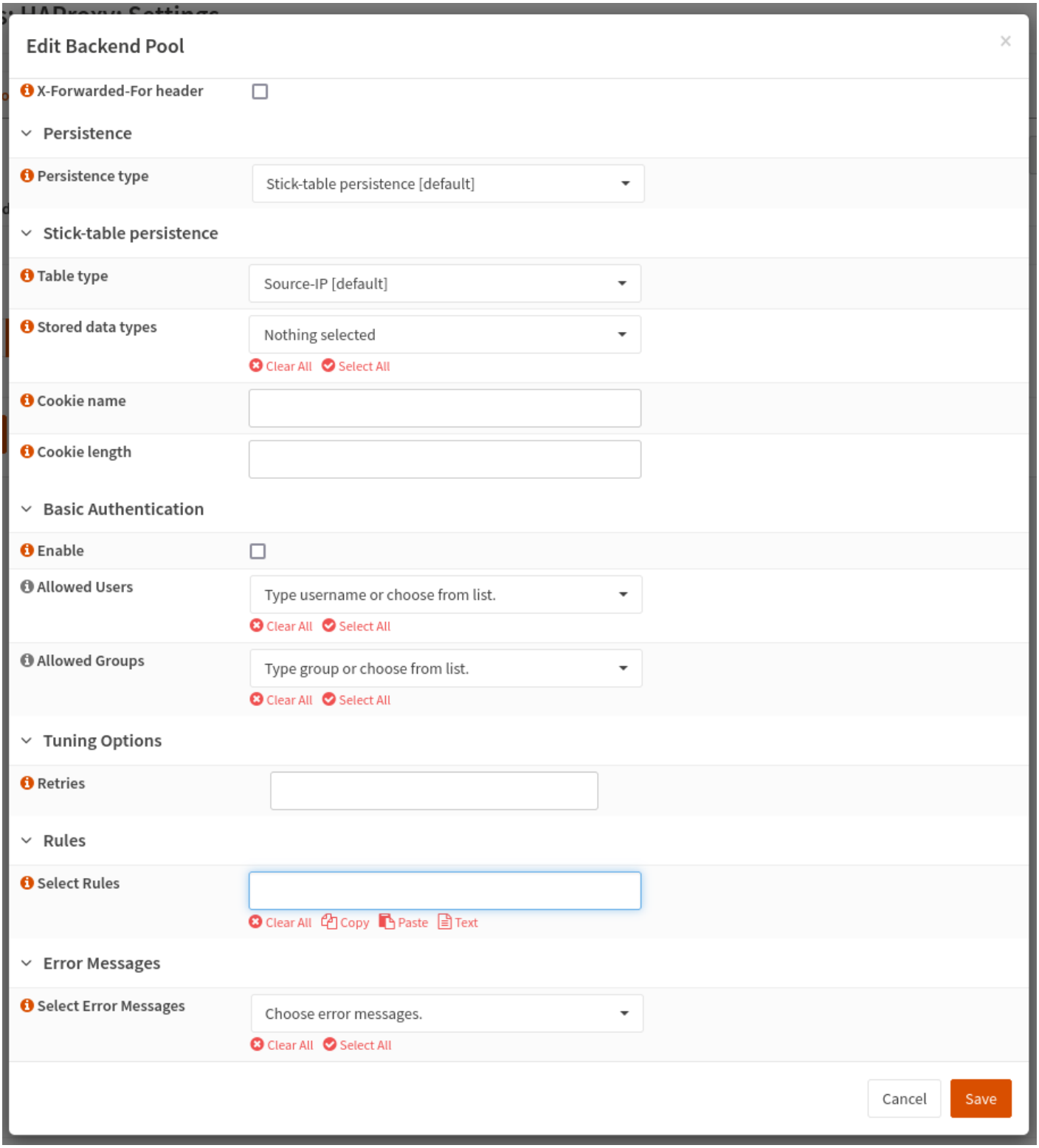
Health Checking

Health Monitor	None
Log Status Changes	☐
E-Mail Alert	None

HTTP(S) settings

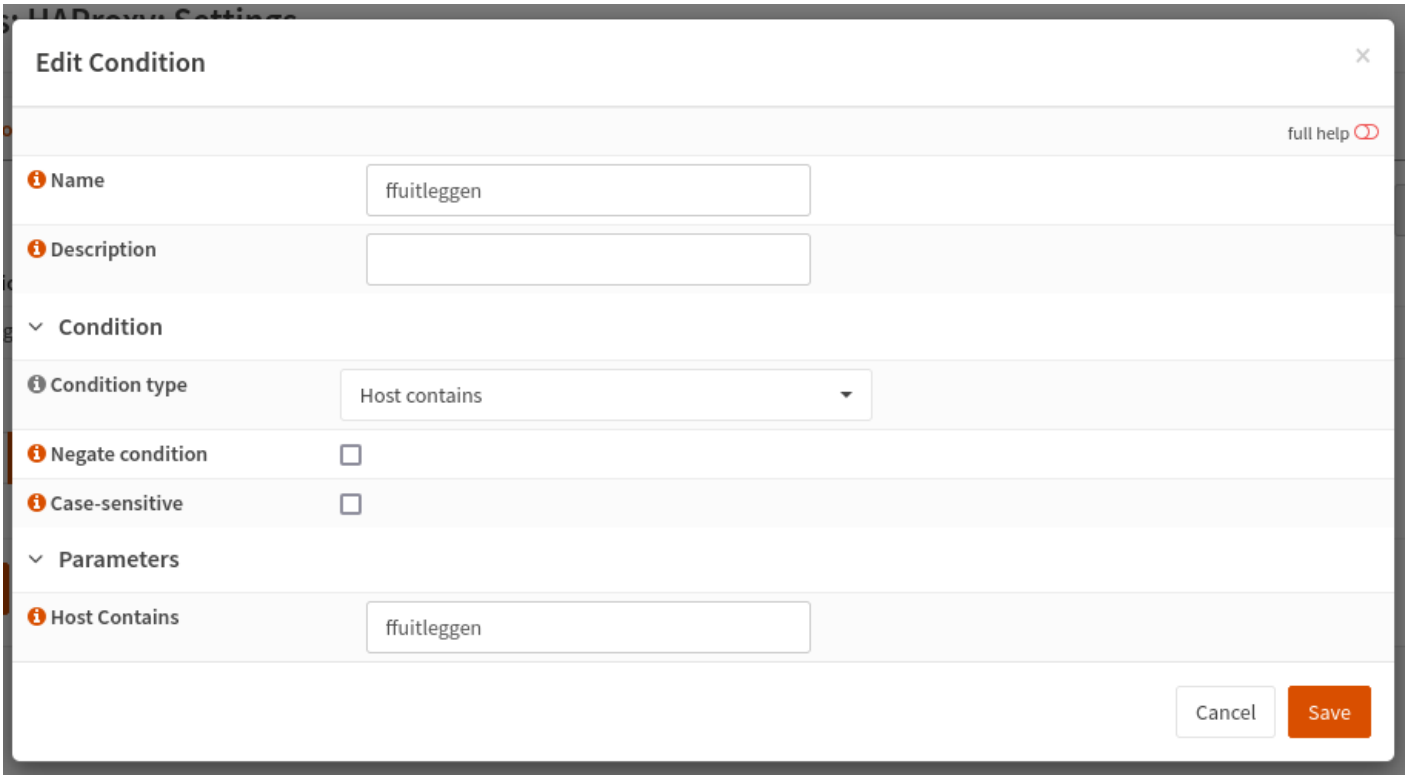
Enable HTTP/2	☑
HTTP/2 without TLS	☐
Advertise Protocols (ALPN)	HTTP/2 × HTTP/1.1 × Clear All Copy Paste Text
Forwarded header (RFC7239)	☐
Forwarded header parameters	

Cancel | Save



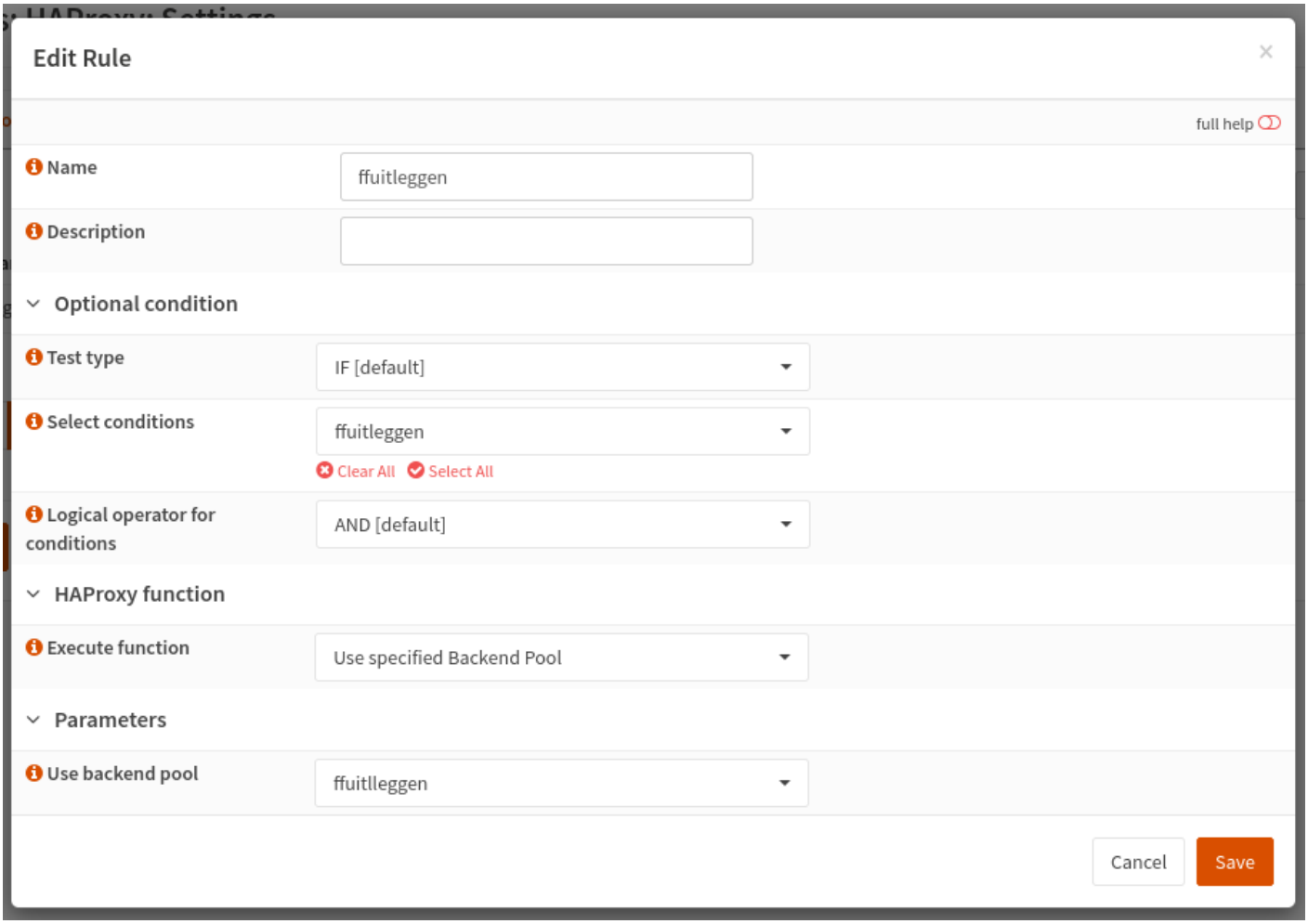
Step 2: Define a condition:

- Name: **anything** you like
- Condition: **Host contains** or you can use any other condition to match like **Host matches** and use the full url.
- Host string = **Anything to recognize the URL** or the full url in case of host matches.



Step 3: Define a rule

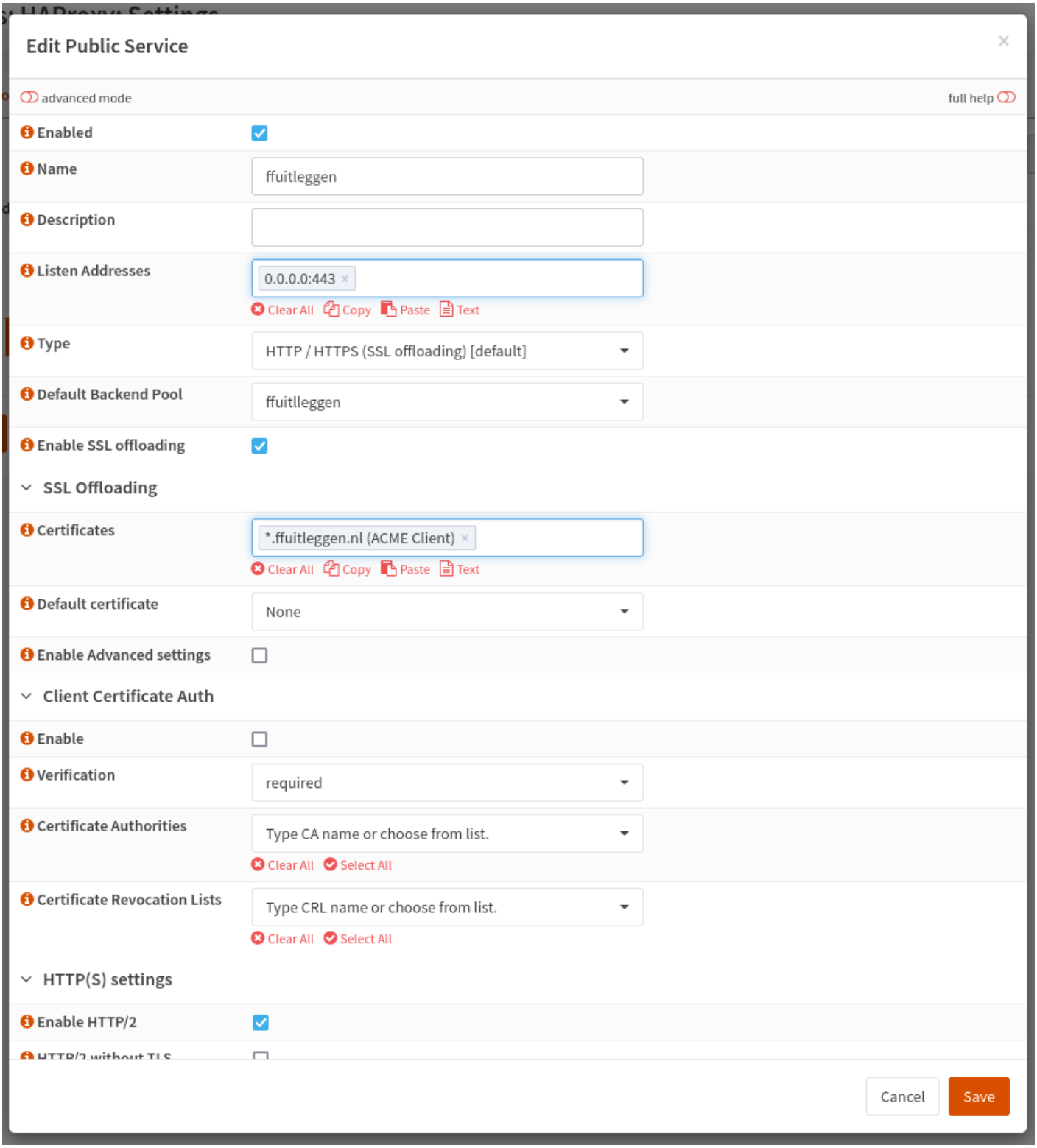
- Name: **anything** you like
- Select Conditions: **Select the webserver** from the dropdown menu
- Under HAProxy function > Execute function: **Use specified Backend Pool**
- Use backend pool: **Select the backend Pool** from the drop down menu

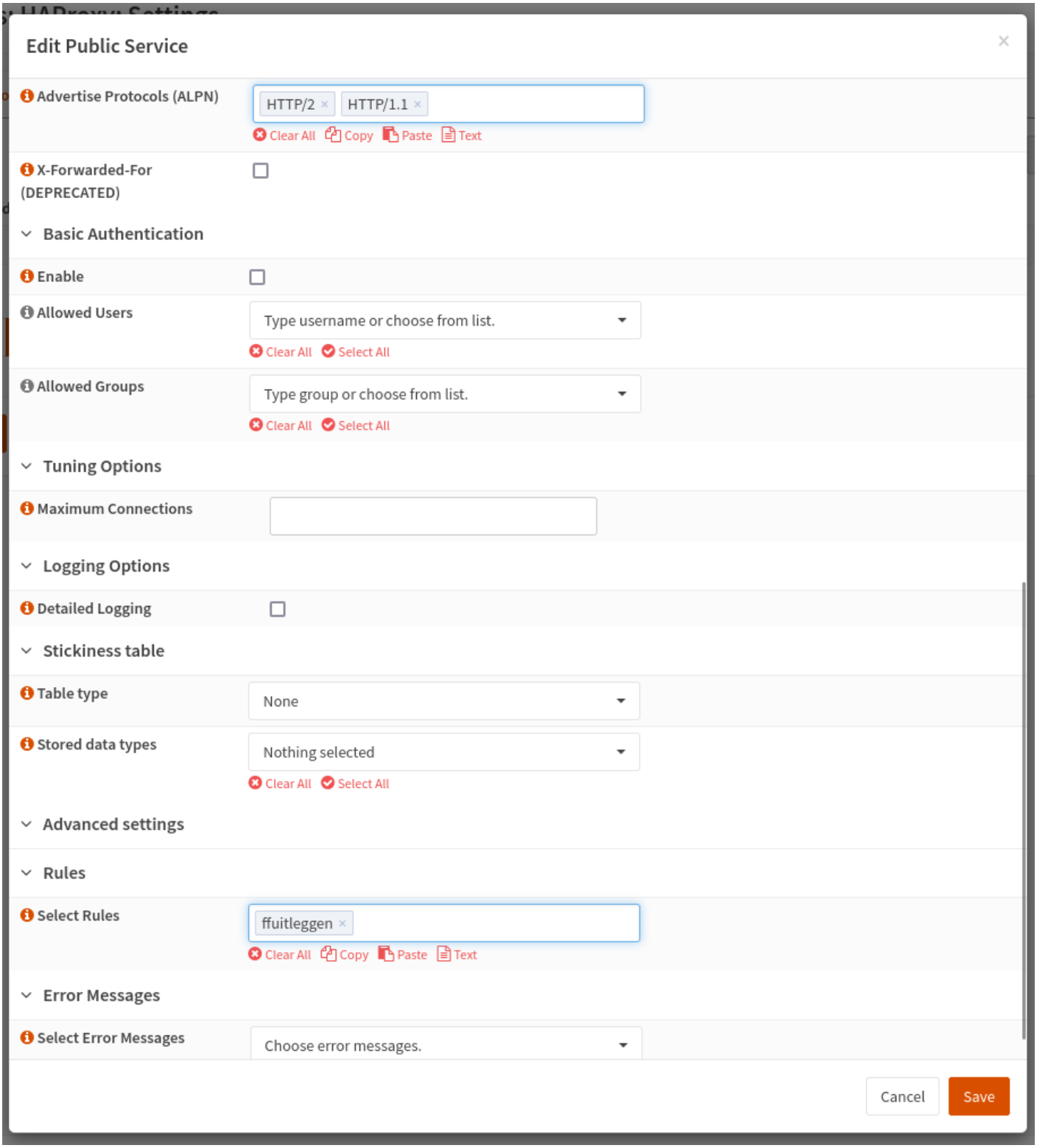


Step 4: Define a Virtual Service

Under Public Service:

- Name: **anything** you like
- Listen addresses: **0.0.0.0:443** (TAB)
- Enable SSL offloading: **Checked**
- Default backend pool: **Select from dropdown menu** (TAB)
- Certificate: **your Let's Encrypt certificate**
- Under Advanced settings:
  - Select rules: **The rule you made earlier**


Edit Public Service
advanced mode
full help
Enabled
Name
ffuitleggen
Description
Listen Addresses
0.0.0.0:443
Clear All
Copy
Paste
Text
Type
HTTP / HTTPS (SSL offloading) [default]
Default Backend Pool
ffuitlleggen
Enable SSL offloading
SSL Offloading
Certificates
*.ffuitleggen.nl (ACME Client)
Clear All
Copy
Paste
Text
Default certificate
None
Enable Advanced settings
Client Certificate Auth
Enable
Verification
required
Certificate Authorities
Type CA name or choose from list.
Clear All
Select All
Certificate Revocation Lists
Type CRL name or choose from list.
Clear All
Select All
HTTP(S) settings
Enable HTTP/2
Cancel
Save


Edit Public Service
Advertise Protocols (ALPN)
HTTP/2
HTTP/1.1
Clear All
Copy
Paste
Text
X-Forwarded-For (DEPRECATED)
Basic Authentication
Enable
Allowed Users
Type username or choose from list.
Clear All
Select All
Allowed Groups
Type group or choose from list.
Clear All
Select All
Tuning Options
Maximum Connections
Logging Options
Detailed Logging
Stickiness table
Table type
None
Stored data types
Nothing selected
Clear All
Select All
Advanced settings
Rules
Select Rules
ffuitleggen
Clear All
Copy
Paste
Text
Error Messages
Select Error Messages
Choose error messages.
Cancel
Save

Adguard Home communications error to 127.0.0.1#53: connection refused

When you **cannot update OPNsense** and you see an error in a SSH session when you try to run:

`root@OPNsense:~ # dig @127.0.0.1 -p 53 google.com`

```
dig @127.0.0.1 -p 53 google.com
```

You probably have a wrong binding in the Adguard config file.

To solve this issue:

```
nano AdGuardhome.yaml
```

```
cd /usr/local/AdGuardHome
```

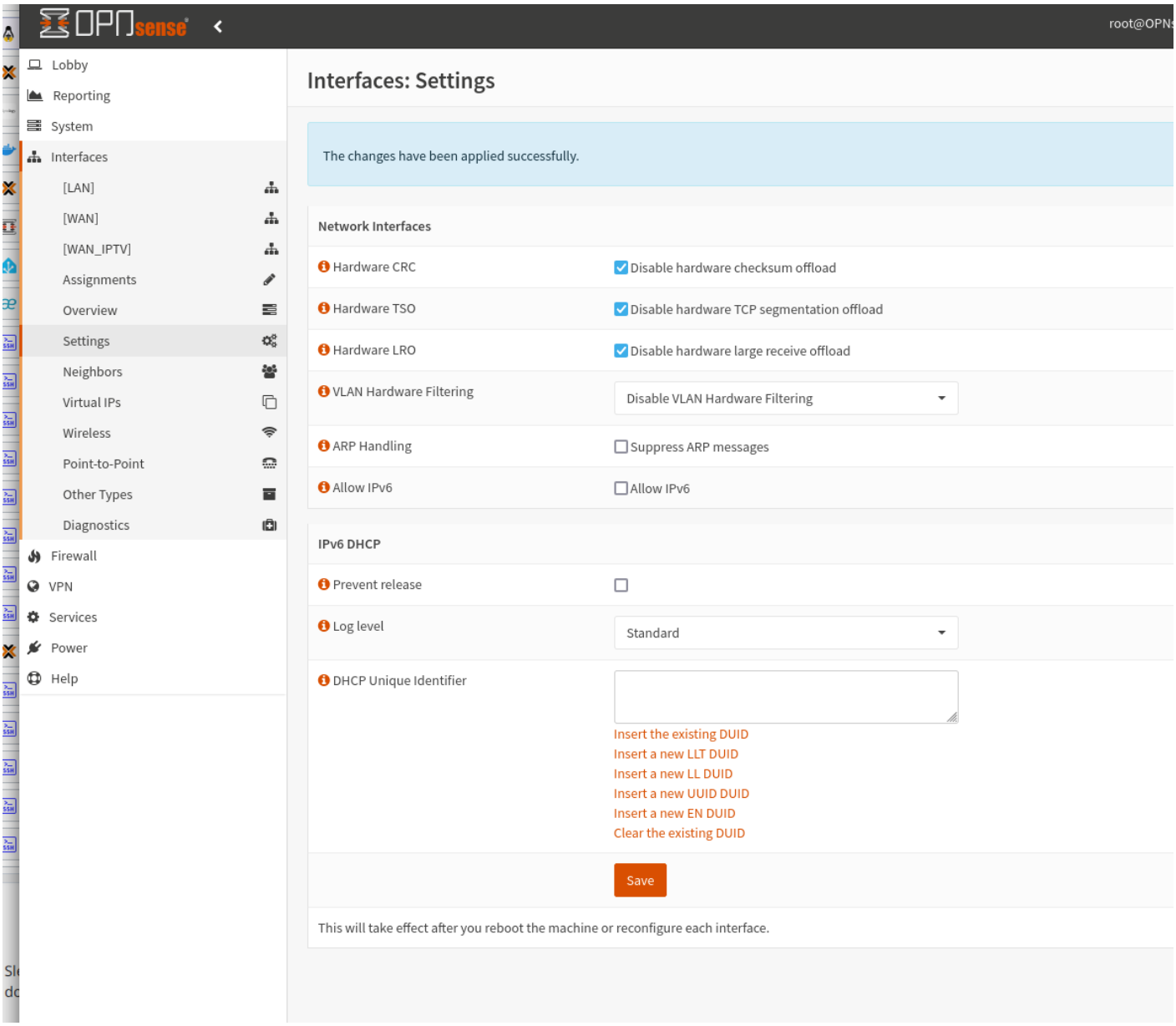
Change the bind (from a local ip address) to:

dns:
** bind_hosts:**
**   - 0.0.0.0**

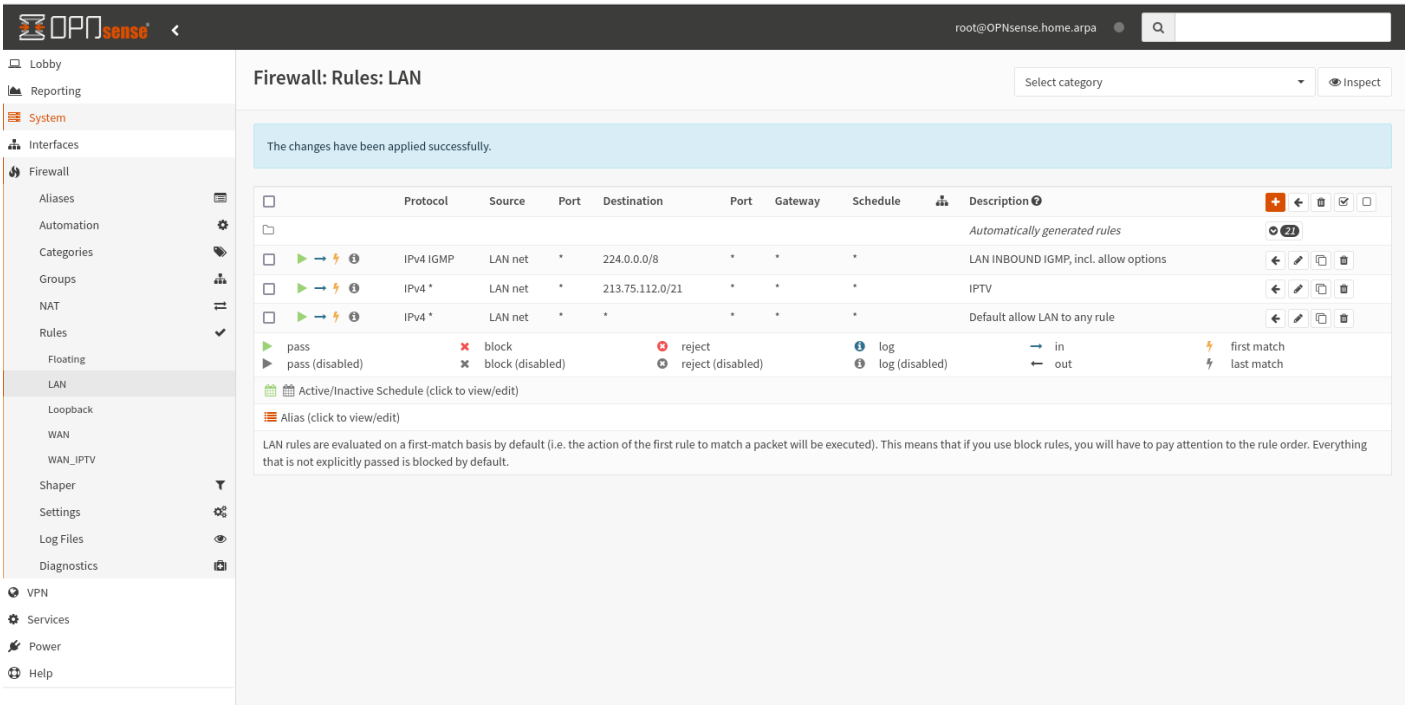
Then restart Adguard Home

Disable IPv6 in OPNSense

Set IPv6 on all interfaces on ' None' and also remove the ' Allow IPv6' vinkje.



Remove also the ' Allow IPv6' rule in de firewall rules:


root@OPNsense.home.arpa
Lobby
Reporting
System
Interfaces
Firewall
Aliases
Automation
Categories
Groups
NAT
Rules
Floating
LAN
Loopback
WAN
WAN_IPTV
Shaper
Settings
Log Files
Diagnostics
VPN
Services
Power
Help
Firewall: Rules: LAN
Select category
Inspect
The changes have been applied successfully.
Protocol
Source
Port
Destination
Port
Gateway
Schedule
Description
Automatically generated rules
21
IPv4 IGMP
LAN net
*
224.0.0.0/8
*
*
*
LAN INBOUND IGMP, incl. allow options
IPv4 *
LAN net
*
213.75.112.0/21
*
*
*
IPTV
IPv4 *
LAN net
*
*
*
*
*
Default allow LAN to any rule
pass
block
reject
log
in
first match
pass (disabled)
block (disabled)
reject (disabled)
log (disabled)
out
last match
Active/Inactive Schedule (click to view/edit)
Alias (click to view/edit)
LAN rules are evaluated on a first-match basis by default (i.e. the action of the first rule to match a packet will be executed). This means that if you use block rules, you will have to pay attention to the rule order. Everything that is not explicitly passed is blocked by default.

How to enable the HAProxy statistics page in OPNsense

Step 1: Edit Global Settings

In the left-hand menu, go to **Services** , select **HAPproxy** and then and then **Settings**.

- Under the **Settings** tab, locate the **Global Parameters**
- Enable ' **Advanced Mode**' on the top left of the page
- Add or modify the following configuration line in the “**Custom Options**” field (on the bottom of the picture):

```
stats socket /var/run/haproxy.socket group proxy mode 775 level admin
```

This enables a UNIX socket for administrative purposes.

Global Parameters

Run as root

☐

Enable or disable HAProxy running as user root. Enabling this option is strongly discouraged.
NOTE: Running as user root could be a security issue but it may be required by some features.

HAProxy threads

4

Number of threads to create for each HAProxy process.

Maximum connections

5000

Sets the maximum number of concurrent connections per HAProxy process.
NOTE: Consider raising the settings for kern.maxfiles and kern.maxfilesperproc in System: Settings: Tunables, otherwise HAProxy will fail to open the specified number of connections.

DNS prefer IP family

IPv4

This option allows to choose which IP family is preferred when resolving DNS names. This is useful when IPv6 or IPv4 is not available. It solves a common issue with OCSP updates.

Verify SSL Server Certificates

no preference [default]

This enforces a certain behavior for SSL verify on servers, ignoring per-server settings. If set to 'enforce verify', server certificates are verified. If set to 'disable verify', server certificates are not verified. The default is 'no preference' to only use per-server configurations and not enforce a global default for all servers.

Maximum SSL DH Size

4096

Sets the maximum size of the Diffie-Hellman parameters used for generating the ephemeral/temporary Diffie-Hellman key in case of DHE key exchange (default is 1024).
NOTE: Higher values will increase the CPU load. For more information about the "tune.ssl.default-dh-param" option please see the HAProxy Documentation.

Buffer size

16384

Change the buffer size (in bytes). Lower values allow more sessions to coexist in the same amount of RAM, and higher values allow some applications with very large cookies to work. The default value is 16384.
NOTE: It is strongly recommended not to change this from the default value, as very low values will break some services such as statistics, and values larger than default size will increase memory usage, possibly causing the system to run out of memory.

Maximum RAM per LUA process

0

Sets the maximum amount of RAM in megabytes per process usable by Lua. By default it is zero which means unlimited. It is important to set a limit to ensure that a bug in a script will not result in the system running out of memory.

Spread checks

2

Add some randomness in the check interval between 0 and +/- 50%. A value between 2 and 5 seems to show good results. The default value is 0 (disabled).

Enable old bogus PROXY v2 implementation

☐

A bug in the PROXY protocol v2 implementation was present in HAProxy up to version 2.1. Enabling this option reverts this old buggy behaviour.

Custom options

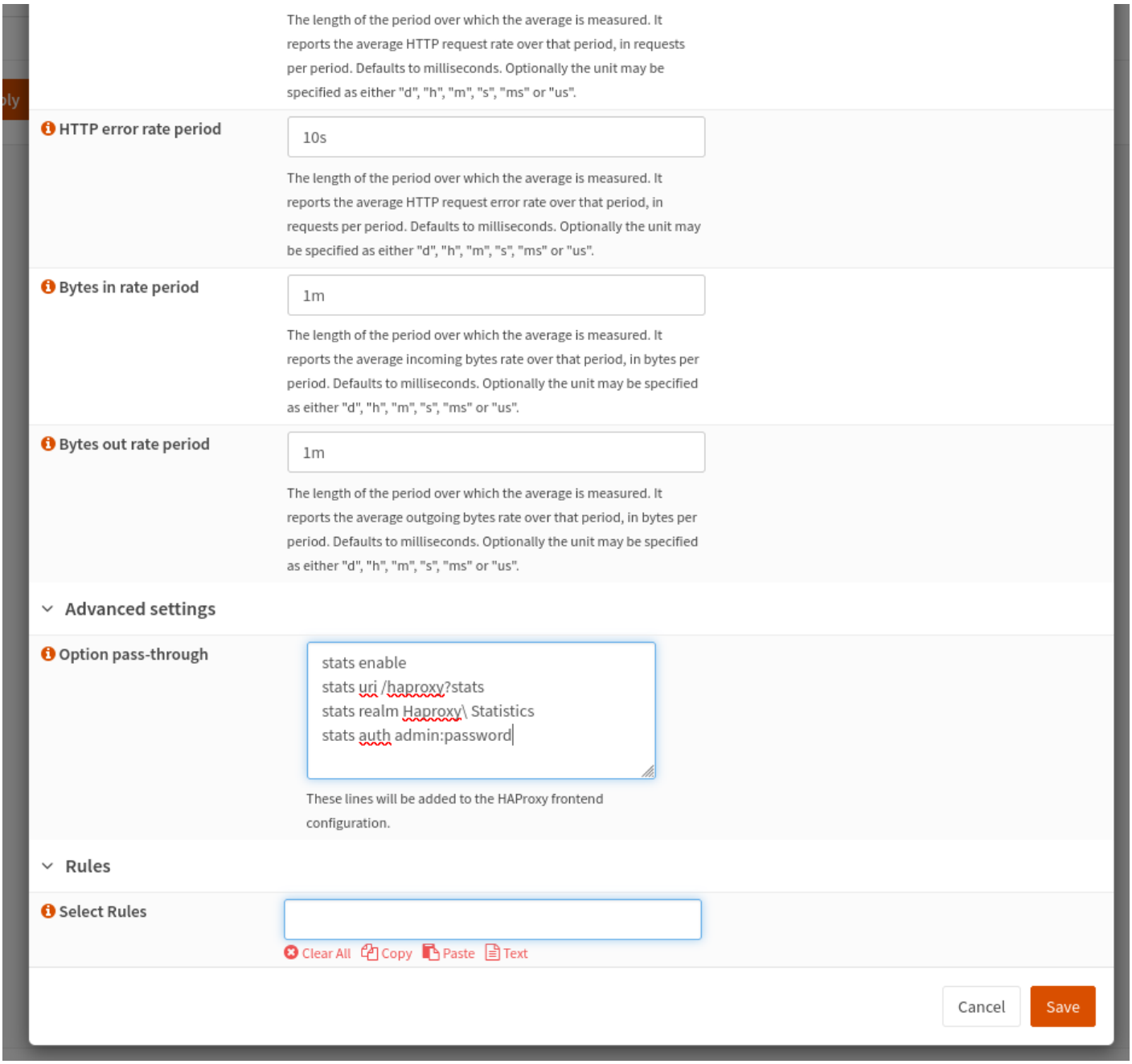
stats socket /var/run/haproxy.socket group proxy mo …

Step 2: Configure Statistics in Frontend Settings

- Go to **Virtual Servers** in the Top menu
- Click the + sign and add a new Public Service: '**StatsFrontend**'
- In this frontend, configure it as follows:
  - Set Name: **StatsFrontend**
  - Set Listen Adressess: set to local IP address of OPNsense (e.g. 192.168.2.1) with the default port 8822
  - Set Type to default **HTTP/HTTPS (SSL offloading) [default]**
  - Scroll all the way down to “**Advanced Settings**”, add these lines in the “**Option Pass-through**” field:

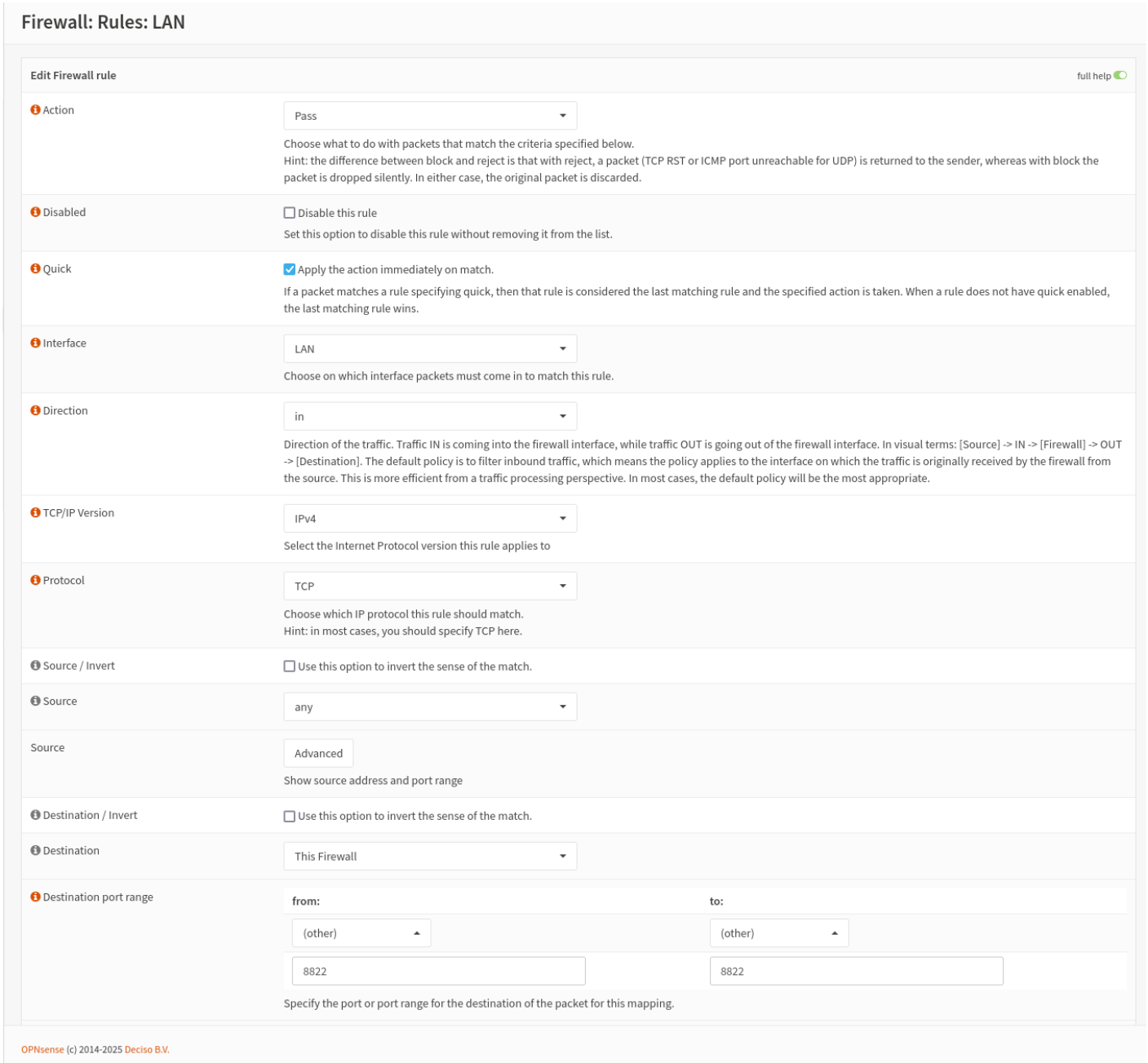
  ```
  stats enable
  stats uri /haproxy?stats
  stats realm Haproxy\ Statistics
  stats auth admin:password123
  ```

  Replace **admin** with your desired username and **password** with a strong password.
  - Click on “Save” and then apply changes by clicking on “Apply”.



Step 3: Configure Firewall Rules

1. **Allow Access to the Statistics Port**:
   - Navigate to **Firewall > Rules > LAN**
   - Create a new rule with these parameters:
     - Action: **Pass**
     - Protocol: **TCP**
     - Destination: **This Firewall**
     - Destination Port Range: **Other and the 8822**
     - Description: **Access the Statistics page**
     - Leave everything else to the default values
     - Save the rule and click on “Apply Changes”.



Step 4: Test Access to the Statistics Page

1. Open a web browser from a device allowed by your firewall rules.
2. Enter the URL for accessing statistics, such as:

```
http://192.168.2.1:8822/haproxy?stats
```

Enter the username and password you configured earlier when prompted.

If everything is configured correctly, you should see HAProxy’s statistics page displaying real-time data about connections, backends, frontends, etc.

HAProxy version 2.8.13-8eb4f8c, released 2024/12/12

Statistics Report for pid 64479

> General process information

pid = 64479 (process #1, nbproc = 1, nbthread = 4)
uptime = 0d 0h00m24s; warnings = 0
system limits: memmax = unlimited; ulimit-n = 10037
maxsock = 10037; **maxconn** = 5000; **reached** = 0; **maxpipes** = 0
current conns = 1; current pipes = 0/0; conn rate = 0/sec; bit rate = 0.000 kbps
Running tasks: 0/23; idle = 100 %

- active UP
- active UP, going down
- active DOWN, going up
- active or backup DOWN
- active or backup DOWN for maintenance (MAINT)
- active or backup SOFT STOPPED for maintenance
- backup UP
- backup UP, going down
- backup DOWN, going up
- not checked

Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

Display option:

- Scope :
- Hide 'DOWN' servers
- Refresh now
- CSV export
- JSON export (schema)

External resources:

- Primary site
- Updates (v2.8)
- Online manual

ffuitleggen

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				0	0	-	0	0	5 000	0			0	0	0	0	0					OPEN								

StatsFrontend

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				0	1	-	1	1	5 000	1			420	469	0	0	0					OPEN								

ffuitleggen

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
ffuitleggen	0	0	-	0	0		0	0	2000	0	0	?	0	0		0		0	0	0	0	*no check*		1/1	Y	-				-
Backend	0	0		0	0		0	0	500	0	0	?	0	0	0	0		0	0	0	0	25s UP		1/1	1	0		0	0s	

local_statistics

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				0	0	-	0	0	5 000	0			0	0	0	0	0					OPEN								
Backend	0	0		0	0		0	0	500	0	0	?	0	0	0	0		0	0	0	0	25s UP		0/0	0	0		0		

HowTo Restore a Google Drive backup file in OPNsense

A description on how to use Google Drive backup feature in OPNsense can be found here:

https://docs.opnsense.org/manual/how-tos/cloud_backup.html

> **But this is a rather complicated process, so read carefully!**

You probably already know that you need a P12 key to store the backup files on Google Drive. Why is that?

The **P12 key** you created for use with Google Drive backups in OPNsense plays an important role in the **authentication process** between OPNsense and Google's API. Here's exactly what the P12 key does during the backup and restore process:

---

1. What does the P12 key do?

The **P12 key** (a so-called **PKCS#12 file**) contains a **private key** that OPNsense uses to **cryptographically authenticate** itself as a **service account** to Google. It is linked to a **Google Cloud service account** that has access to your Google Drive.

In short:

1. **Authentication**
   When creating a backup, OPNsense connects to the Google Drive API.
2. **Signing a JWT (JSON Web Token)**
   OPNsense generates a specially formatted token (JWT) and signs it with the private key from the P12 file.
3. **Token exchange with Google**
   The signed JWT is sent to Google's OAuth 2.0 token endpoint, and in return, OPNsense receives an **access token**.
4. **Access to Google Drive**
   With that access token, OPNsense can upload or download files from Google Drive on

behalf of the service account.

Why is this necessary?

Unlike user-based authentication (which uses a browser and interactive OAuth consent), this is a **server-to-server authentication method**. That’s ideal because OPNsense needs to perform **automated backups** without requiring manual login or user consent each time.

What happens during a restore?

During a restore:

1. OPNsense follows the same authentication process to access Google Drive.
2. It retrieves the list of XML backup files.
3. The user selects a file, and OPNsense downloads it via the API (authorized through the same token process).

Important security note

- **Treat the P12 file as a secret.**
  Anyone with access to this key — and who knows the associated service account — can access your backup files.
- Make sure only OPNsense (and you as the administrator) have access to this file.

2. How to restore a backup from Google Drive when using Linux?

With kio-Gdrive installed in Dolphin is not working because Google blocks this

That’s a known limitation of **kio-gdrive** in combination with Google’s stricter security policies. Google now blocks applications that are **not verified or have not gone through OAuth validation**, which applies to many open-source or locally installed apps like kio-gdrive.

Why does Google block kio-gdrive?

Google sees kio-gdrive as an **"unverified app"** requesting access to **sensitive scopes** (such as full access to your Drive). Because of that, Google refuses to complete the OAuth flow.

Solutions

1. Use `rclone` instead of kio-gdrive (recommended)

As mentioned earlier, `rclone` is the most robust and Google-compliant way to access Google Drive locally without running into OAuth issues. Rclone is recognized by Google, uses approved scopes, and just works.

➡ **Advantage**: Stable and compatible with service accounts.

Installeren en instellen

Install rclone:

```
sudo apt install rclone
```

Configure `rclone` for Google Drive:

```
rclone config
```

Choose:

- `n` to create a **new remote configuration**.
- Enter a name, for example: `gdrive`.
- Choose option `13` for **Google Drive**.
- Leave **Client ID** and **Client Secret** blank (or enter the values from your Google Cloud Console if you have them).
- Select `"service_account_file"` when prompted and provide the path to your `.p12` or `.json` file.

> ⚠ **Note:** `rclone` works **best with JSON service account keys** rather than P12.
> If you only have a P12 key, you may first need to manually generate a JSON key via the **Google Cloud Console**.
> Because we use OPNsense, we have no choice but using a P12 key.

Verify your setup:

```
rclone ls gdrive:
```

Download a backup file:

```
rclone copy gdrive:opnsense-backups/config-2025-05-28.xml .
```

Upload the file via the OPNsense web interface:

Go to:
`System` → `Configuration` → `Backups` → `Restore` → Upload the `.xml` file.

---

2. Create your own Google API project (advanced)

If you really want to keep using kio-gdrive (not recommended), then you'll need to:

1. Go to the Google Cloud Console.
2. Create a new project.
3. Enable the **Google Drive API**.
4. Create OAuth 2.0 client credentials.
5. Manually configure these credentials in kio-gdrive.

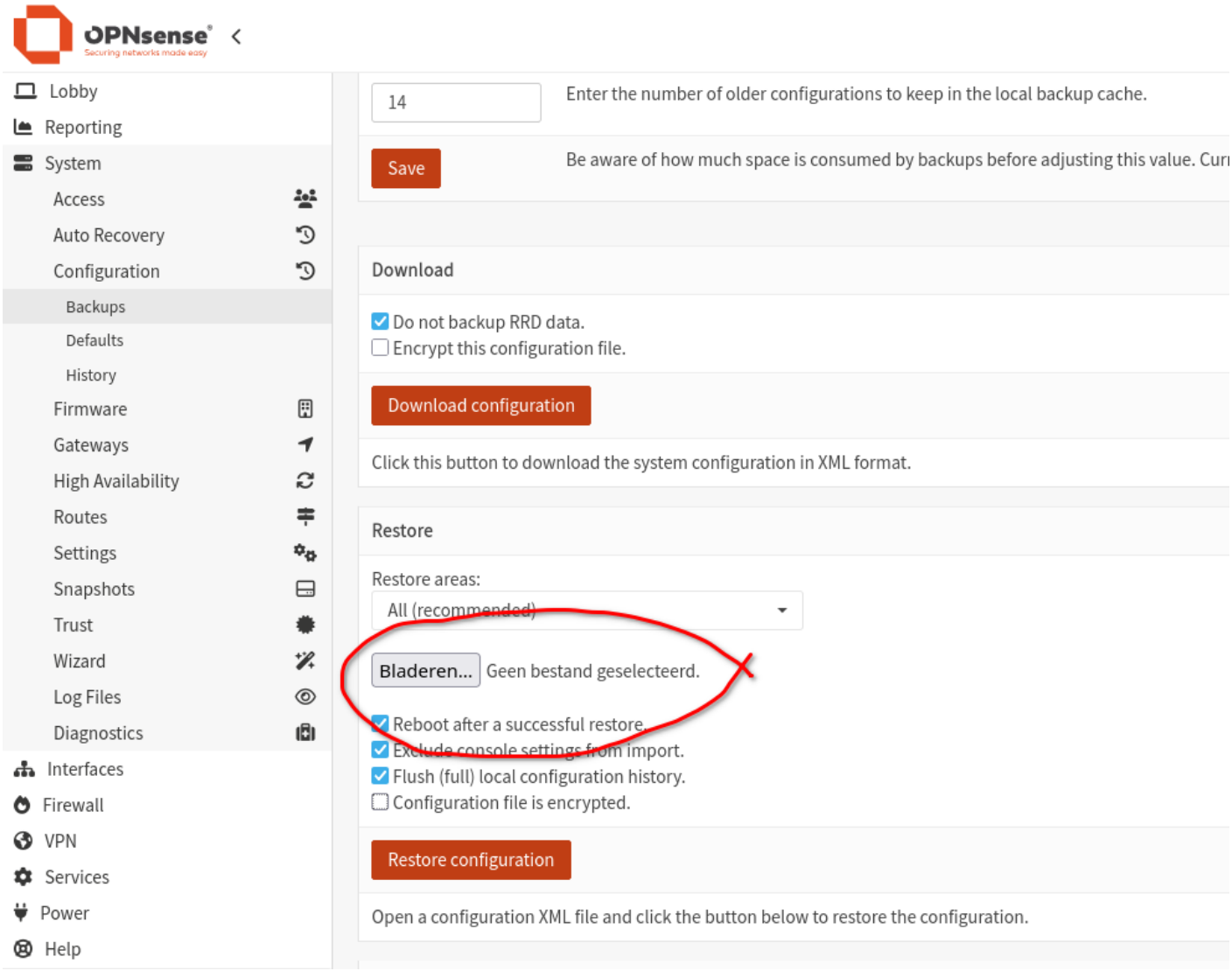
➡ Google will still show a warning ("unverified app") and you'll need to bypass it by clicking "Advanced" → "Proceed to...".

⚠ This is unstable and error-prone, especially if you use 2FA.

---

3. Download backups manually via your browser

This is the easiest method for occasional use: just log in to Google Drive via your browser and download the XML backup file manually. Because I have no need for frequent accessing the backup files I use this method and select the downloaded file under **Restore**:



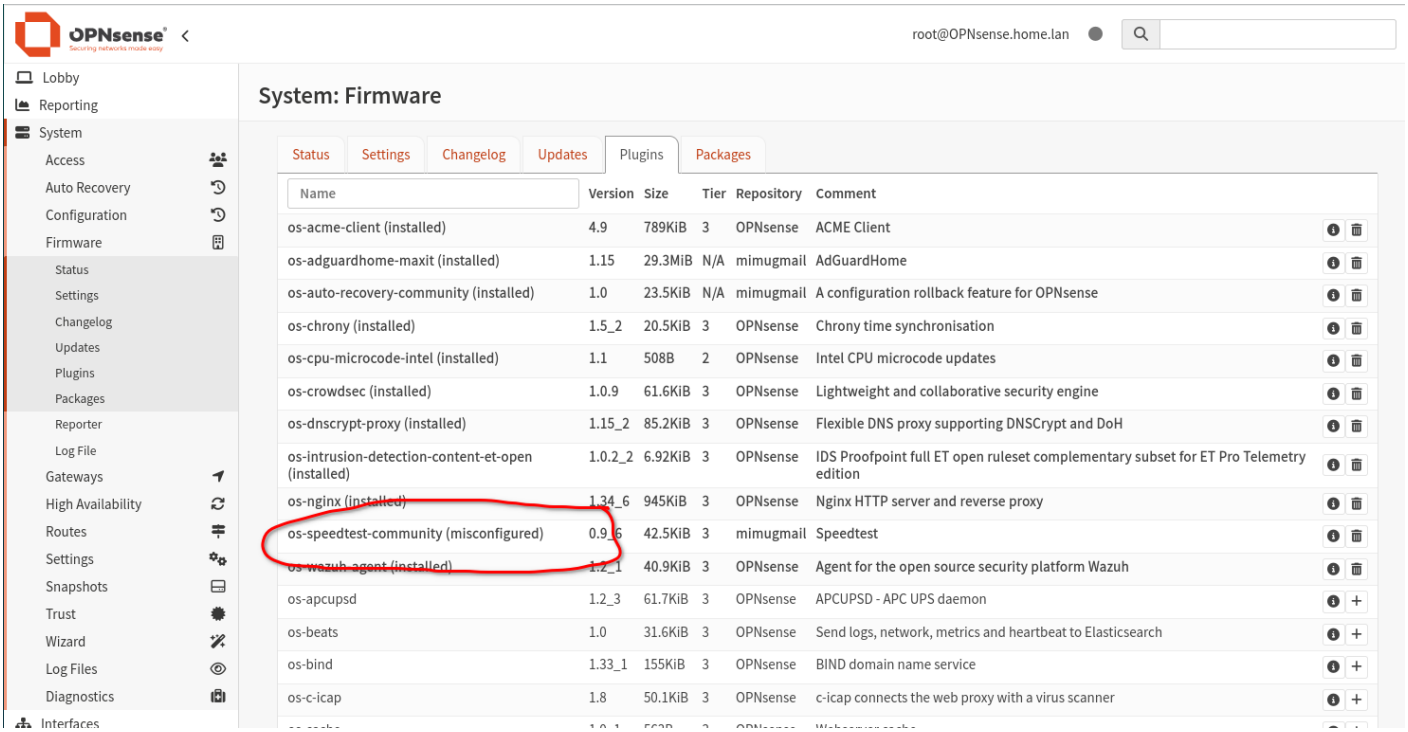
You don't need the P12 key to restore, because it is already known in OPNsense.

Run frequent Speedtest in OPNsense

To be able to use the Speedtest plugin, you need to install the Mimugmail repository:

https://github.com/mimugmail/opn-repo

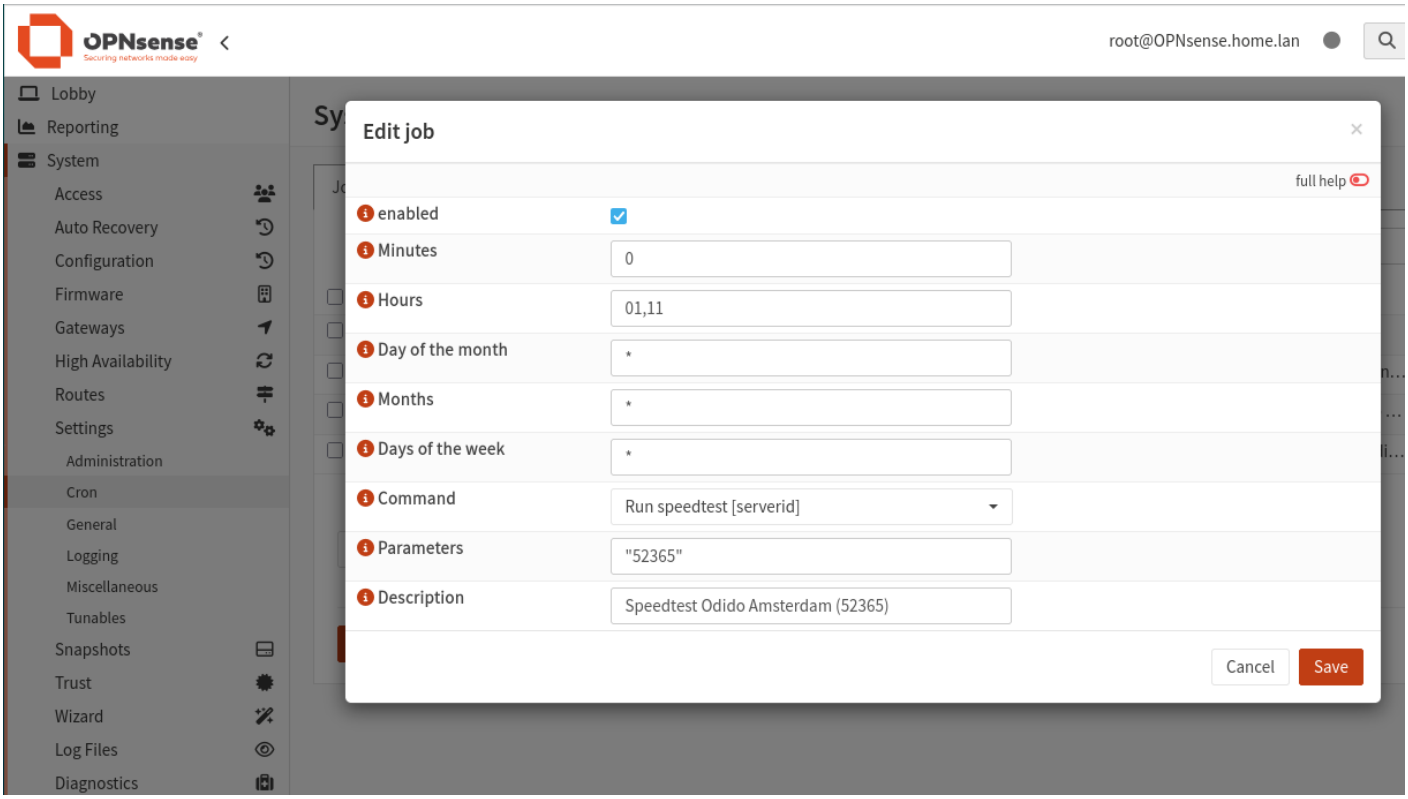
Then go to System>Firmware>Plugins and install the '**os-speedtest-community**' plugin in OPNsense.



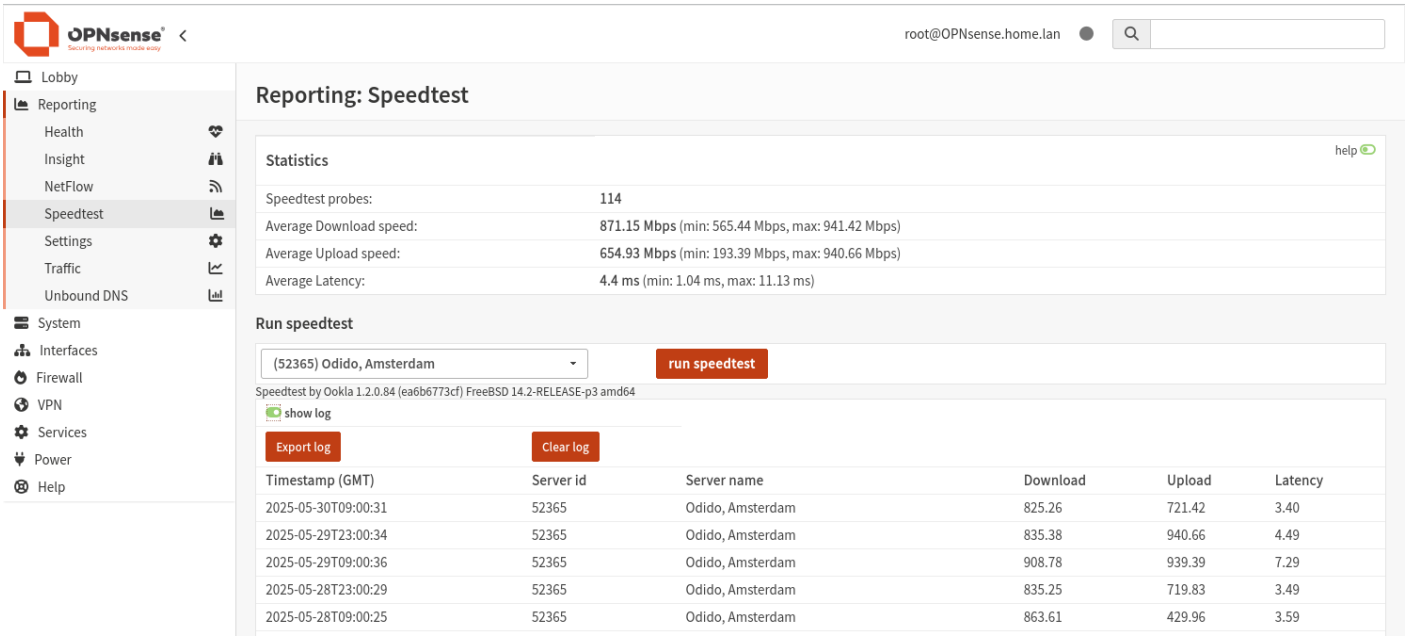
> Yes, there is a "misconfigured" notice, but everything will work fine :-)

In OPNsense "Reporting > Speedtest", locate desired server and make note of server ID, (the numbers between the brackets in "(#####) Server Name")

go to "System > Settings > Cron" create/edit the speedtest entry.
Set "Command" field to "Run speedtest [serverid]"
Set "Parameters" field to "#####" you dont need to append with "--server"... literally... just the numbers.



Under Reporting>Speedtest you can find the results. Just click on the little "show log" switch.



Voorgestelde waarden voor Advanced tab in Unbound

> **Screenshots staan onderaan deze post.**

Basisinstellingen voor privacy en veiligheid:

1. **Hide Identity**:
   - **Aan**: Verbergt de identiteit van de Unbound-server (zoals het feit dat je Unbound gebruikt) in DNS-responses. Dit is handig voor privacy.
2. **Hide Version**:
   - **Aan**: Verbergt de versie van Unbound die wordt gebruikt in de DNS-responses. Dit voorkomt dat een aanvaller de versie van je server kan achterhalen.
3. **Prefetch DNS Key Support**:
   - **Aan**: Verbetert de prestaties van DNSSEC-queries door vooraf de sleutels van de domeinen in te laden, wat de latentie kan verminderen.
4. **Harden DNSSEC Data**:
   - **Aan**: Zorgt ervoor dat DNSSEC-resolutie strikt wordt gehandhaafd. Dit betekent dat alle DNSSEC-handtekeningen strikt worden gecontroleerd, wat de veiligheid verhoogt.
5. **Aggressive NSEC**:
   - **Aan**: Verhoogt de veiligheid door gebruik te maken van de agressieve NSEC-beveiliging, wat voorkomt dat er onterechte 'NXDOMAIN' antwoorden worden gegeven voor niet-bestaande domeinen.
6. **Strict QNAME Minimisation**:
   - **Aan**: Minimaliseert de hoeveelheid informatie die Unbound naar upstream DNS-servers stuurt, wat kan helpen om je privacy te beschermen en gegevenslekken te verminderen.
7. **Rebind protection networks**:
   - **Aan**: Activeer dit om rebind-aanvallen te voorkomen, waarbij een aanvaller je DNS-resolver probeert te misleiden door interne IP-adressen te maskeren.

Prestatiesettings:

8. **Outgoing TCP Buffers**:
    - Standaardinstellingen zijn vaak prima, maar je kunt het verhogen als je een snelle verbinding hebt met veel gelijktijdige queries. Dit kan helpen bij het omgaan met grotere verzoeken via TCP.
9. **Incoming TCP Buffers**:
    - Eveneens kun je de standaardinstellingen gebruiken, maar bij hoge verkeersvolumes kan het verhogen van de bufferwaarde helpen om betere prestaties te bereiken.
10. **Number of queries per thread**:
    - **2 tot 4** is een goede instelling voor de meeste omgevingen, afhankelijk van de belasting. Verhoog dit als je veel queries verwerkt.
11. **Outgoing Range**:
    - Standaard is 10-20 prima. Verhoog dit om grotere hoeveelheden gelijktijdige uitgaande verzoeken te verwerken.
12. **Jostle Timeout** en **Discard Timeout**:
    - Deze zijn goed ingesteld op **default**, maar als je last hebt van time-outs bij het resolven van verzoeken, kun je experimenteren met langere time-outs.

Cachinginstellingen:

13. **Message Cache Size**:
    - Dit kan verhoogd worden voor grotere netwerken (bijv. 50-100 MB), afhankelijk van de hoeveelheid DNS-verkeer die je genereert.
14. **RRset Cache Size**:
    - Vergroot deze als je veel verzoeken hebt naar dezelfde domeinen. Standaardwaarden zijn vaak voldoende, maar grotere netwerken kunnen baat hebben bij een grotere cache.
15. **Maximum TTL for RRsets and messages**:
    - **Maximaal 86400 (1 dag)** is een gangbare instelling voor de TTL. Dit bepaalt hoe lang records worden bewaard. Dit kan de prestaties verbeteren, maar verhoogt het risico van verouderde informatie.
16. **Minimum TTL for RRsets and messages**:
    - **Standaardinstelling** van 0 is prima, tenzij je wilt forceren dat bepaalde records voor een minimumtijd worden bewaard.
17. **TTL for Host Cache entries**:
    - Dit moet niet te hoog ingesteld worden. **900 seconden (15 minuten)** is een gangbare waarde, wat een goede balans biedt tussen prestaties en actualiteit van gegevens.
18. **Keep probing down hosts**:
    - **Uit** is vaak goed, tenzij je wilt dat Unbound blijft proberen om onbereikbare hosts te bereiken, wat de prestaties kan beïnvloeden.
19. **Number of Hosts to cache**:

- Een hogere waarde, zoals **1000**, kan nuttig zijn voor grotere netwerken. Dit bepaalt hoeveel hostnamen in het cachegeheugen worden bewaard.

Logginginstellingen:

20. **Log Queries**:
    - **Uit** voor minder logverkeer en meer privacy, tenzij je specifieke diagnostiek nodig hebt.
21. **Log Replies**:
    - **Uit** is meestal voldoende om de belasting op je logs laag te houden.
22. **Tag Queries and Replies**:
    - **Uit** voor betere prestaties, tenzij je een gedetailleerde audit nodig hebt.
23. **Log local actions**:
    - **Aan** kan handig zijn voor probleemoplossing, maar dit kan leiden tot veel logverkeer.
24. **Log SERVFAIL**:
    - **Aan** kan nuttig zijn voor diagnostiek van problemen, maar zet het uit voor meer geoptimaliseerde prestaties.

Diverse instellingen:

25. **Serve Expired Settings**:
    - **Aan**: Verbetert de beschikbaarheid van gegevens door verlopen gegevens toch te serveren als tijdelijke oplossing (bijvoorbeeld als de upstream DNS niet reageert).
26. **Serve Expired Responses**:
    - **Aan** zorgt ervoor dat verouderde gegevens toch kunnen worden bediend, maar dit kan veiligheidsrisico's inhouden als je vertrouwt op de validiteit van de gegevens.
27. **Extended Statistics**:
    - **Aan** voor gedetailleerdere statistieken, maar dit verhoogt de belasting op de server.
28. **Log Level Verbosity**:
    - **2 of 3** voor gedetailleerde logs. Je kunt dit verhogen naar **4** als je gedetailleerde foutopsporingsinformatie nodig hebt.
29. **Log validation level**:
    - **Minimaal** of **Fouten** voor minder logverkeer, tenzij je actief DNSSEC-validatieproblemen onderzoekt.

Aanbevolen instellingen (samenvatting):

- **Privacy**: Hide Identity, Hide Version, Harden DNSSEC Data, Strict QNAME Minimisation.
- **Veiligheid**: Rebind protection networks, Aggressive NSEC.
- **Prestaties**: Prefetch DNS Key Support, Outgoing Range, Number of queries per thread.
- **Cache**: Message Cache Size, RRset Cache Size, Maximum TTL for RRsets.
- **Logging**: Zet logging zoveel mogelijk uit voor betere prestaties, maar log SERVFAIL als je problemen wilt onderzoeken.

Door deze instellingen zorgvuldig af te stemmen, kun je de prestaties, privacy en veiligheid van je DNS-resolutie optimaliseren.

- Lobby
- Reporting
- System
- Interfaces
- Firewall
- VPN
- Services
  - ACME Client
  - Adguardhome
  - Captive Portal
  - CrowdSec
  - DHCRelay
  - DNSCrypt-Proxy
  - Dnsmasq DNS & DHCP
  - Intrusion Detection
  - ISC DHCPv4
  - ISC DHCPv6
  - Kea DHCP
  - Monit
  - Network Time
  - Nginx
  - OpenDNS
  - Unbound DNS
    - General
    - Overrides
    - Advanced
    - Access Lists
    - Blocklist
    - Query Forwarding
    - DNS over TLS
    - Statistics
    - Log File
  - Wazuh Agent
- Power
- Help

Services: Unbound DNS: Advanced

General Settings

Hide Identity	☑
Hide Version	☑
Prefetch DNS Key Support	☑
Harden DNSSEC Data	☑
Aggressive NSEC	☑
Strict QNAME Minimisation	☑
Outgoing TCP Buffers	
Incoming TCP Buffers	
Number of queries per thread	4
Outgoing Range	
Jostle Timeout	
Discard Timeout	
Private Domains	Clear All Copy Paste Text
Rebind protection networks	0.0.0.0/8 × 10.0.0.0/8 × 100.64.0.0/10 × 169.254.0.0/16 × 172.16.0.0/12 × 192.0.2.0/24 × 192.168.0.0/16 × 198.18.0.0/15 × 198.51.100.0/24 × 203.0.113.0/24 × 233.252.0.0/24 × ::1/128 × 2001:db8::/32 × fc00::/8 × fd00::/8 × fe80::/10 × Clear All Copy Paste Text
Insecure Domains	Clear All Copy Paste Text

Serve Expired Settings

Serve Expired Responses	☐

Logging Settings

Extended Statistics	☐
Log Queries	☐
Log Replies	☐
Tag Queries and Replies	☐

- Lobby
- Reporting
- System
- Interfaces
- Firewall
- VPN
- Services
  - ACME Client
  - Adguardhome
  - Captive Portal
  - CrowdSec
  - DHCRelay
  - DNSCrypt-Proxy
  - Dnsmasq DNS & DHCP
  - Intrusion Detection
  - ISC DHCPv4
  - ISC DHCPv6
  - Kea DHCP
  - Monit
  - Network Time
  - Nginx
  - OpenDNS
  - Unbound DNS
    - General
    - Overrides
    - Advanced
    - Access Lists
    - Blocklist
    - Query Forwarding
    - DNS over TLS
    - Statistics
    - Log File
  - Wazuh Agent
- Power
- Help

2001:db8::/32 × fc00::/8 × fd00::/8 × fe80::/10 ×

Clear All Copy Paste Text

Insecure Domains

Clear All Copy Paste Text

Serve Expired Settings

Serve Expired Responses ☐

Logging Settings

Extended Statistics ☐

Log Queries ☐

Log Replies ☐

Tag Queries and Replies ☐

Log local actions ☐

Log SERVFAIL ☐

Log Level Verbosity Level 2

Log validation level Level 0 (Default)

Cache Settings

Prefetch Support ☑

Unwanted Reply Threshold

Message Cache Size 256M

RRset Cache Size

Maximum TTL for RRsets and messages 86400

Maximum Negative TTL for RRsets and messages

Minimum TTL for RRsets and messages 300

TTL for Host Cache entries 900

Keep probing down hosts ☐

Number of Hosts to cache 1000

Apply

Setup os-ddclient for when external IP address changes

Here's a step-by-step guide in **English** on how to configure **os-ddclient** in **OPNsense** to automatically update your **Cloudflare DNS records** when your external IP address changes (e.g., from your ISP).

> If you get the concept of this then you should be able to do this for DuckDNS or Dynu DNS or one of the other options in a similar way.

Prerequisites

Before you begin:

- You need a **Cloudflare account**.
- You need a domain managed by Cloudflare.
- You should have your **Cloudflare API token** or **API key** (Token is preferred for security).
- `os-ddclient` must be **installed and enabled** in OPNsense (`System > Firmware > Plugins` > install `os-ddclient`).

Step-by-step Configuration

1. **Enable os-ddclient**

Go to:

- `Services > Dynamic DNS > Settings`
- Check **Enable**
- Click **Save**

2. **Add a Dynamic DNS Account**

Go to:

- `Services > Dynamic DNS > Accounts`
- Click **Add**

Now fill in the required fields:

General Settings

- **Enabled**: ☐
- **Description**: e.g., `Cloudflare mydomain.com`
- **Service**: `cloudflare`

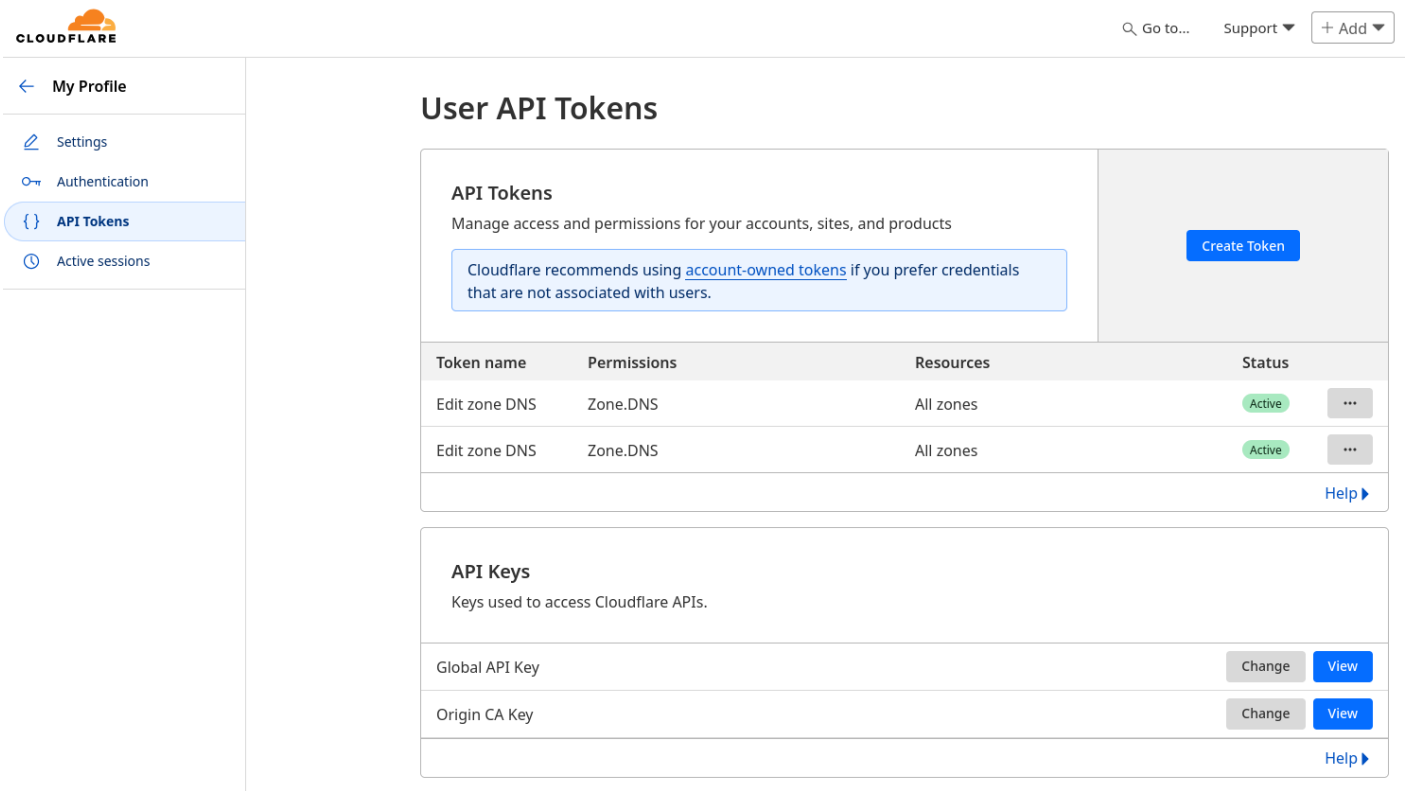
Cloudflare API Credentials

You have two options: API Token (preferred) or Global API Key.

If you're using an **API Token**:

- **Username**: leave blank *>>there are problem reported when this is filled in and you are using the* API key method.
- **Password**: paste your **API Token**
- Make sure your API token has permission to:
  - Zone: DNS → Edit
  - Zone: Zone → Read

I have made a new API token. When logged in on Cloudflare, go to the top right and click on **Profile** . Then go to **{} API Tokens** and add a new token for "**Edit zone DNS**". Save the token on a safe place.



If you're using a **Global API Key**:

- **Username / Email**: your Cloudflare account email
- **Password**: your **Global API Key**

> **The use of the Global API key for this is not recommended!**

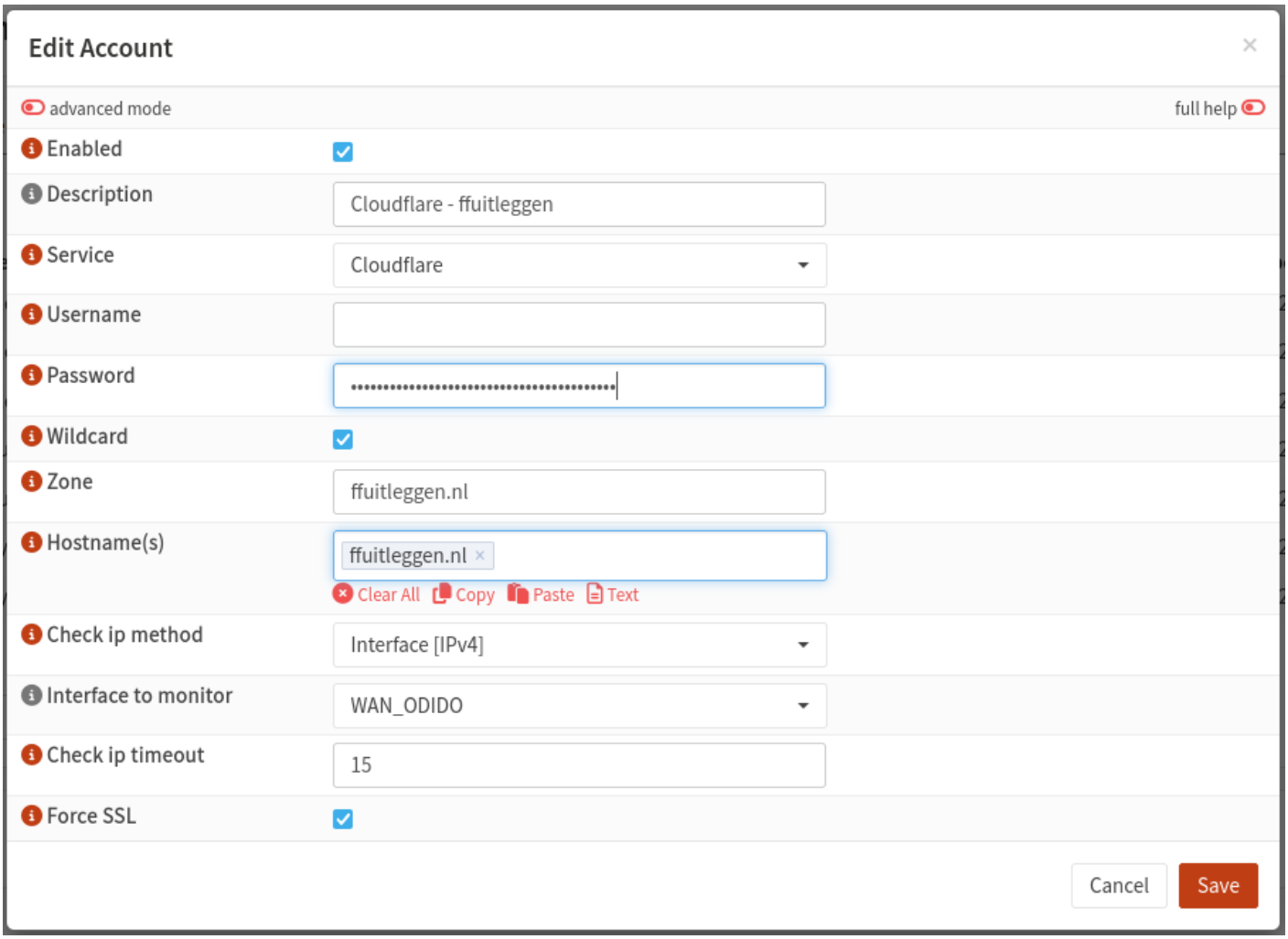
Hostname Details

- **Wildcard**: usually unchecked unless you want `*.home.example.com` updated too
- **Zone**: your domain name (e.g., `example.com`)
- **Hostname(s)**: the DNS record(s) you want to update (e.g., `home.example.com`) >>*when the same API key is used then you can add multiple domain names here.*

IP Settings

- **Check IP Method**: `Interface` (or use `Web` for online IP detection)
- **Interface to monitor**: `WAN`
- **Check ip timeout**: `10` >> default value in minutes. Leave it as it is.
- **Force SSL**: ☑ (recommended)

Click **Save**, then **Apply**.



Test the Setup

1. After saving, go to the `Services > Dynamic DNS > Log File`.
2. Click **Run now** next to your entry to test it.
3. Check the log to see if the IP update succeeded.

Done!

Your OPNsense box will now monitor your WAN IP and automatically update your **Cloudflare DNS A or AAAA record** whenever your public IP changes.