Installation and Configuration Guide

Step-by-step instructions for installing, configuring and setting up the PUQVPNCP WHMCS module — panel preparation, WHMCS integration, server configuration, location routing and product configuration.

- Setup guide — PUQVPNCP panel
- Add server (PUQVPNCP panel)
- Product configuration

Setup guide — PUQVPNCP panel

PUQVPNCP module **WHMCS**

Order now | Download | COMMUNITY | PUQVPNCP

Before you can connect WHMCS, you need a running PUQVPNCP panel, an **API token** that WHMCS will use for every operation, and at least one **VPN network** with the protocols you want to expose enabled. This page walks through both.

---

1. Panel reachability

- The panel must be reachable from the WHMCS server over the network.
- HTTPS is strongly recommended. If you use a self-signed certificate, remember that SSL verification is enabled when the **Secure** checkbox is ticked on the WHMCS server record — use a publicly trusted certificate, or place the panel behind a reverse-proxy with one.

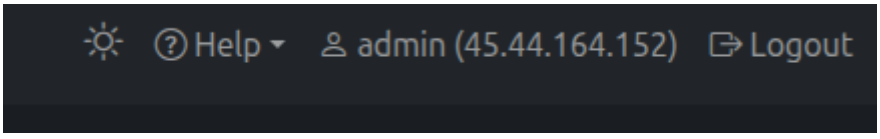
---

2. Issue an API token

The module authenticates to the panel with a **Bearer token** issued from the admin's profile page.

Step 1 — Open Profile

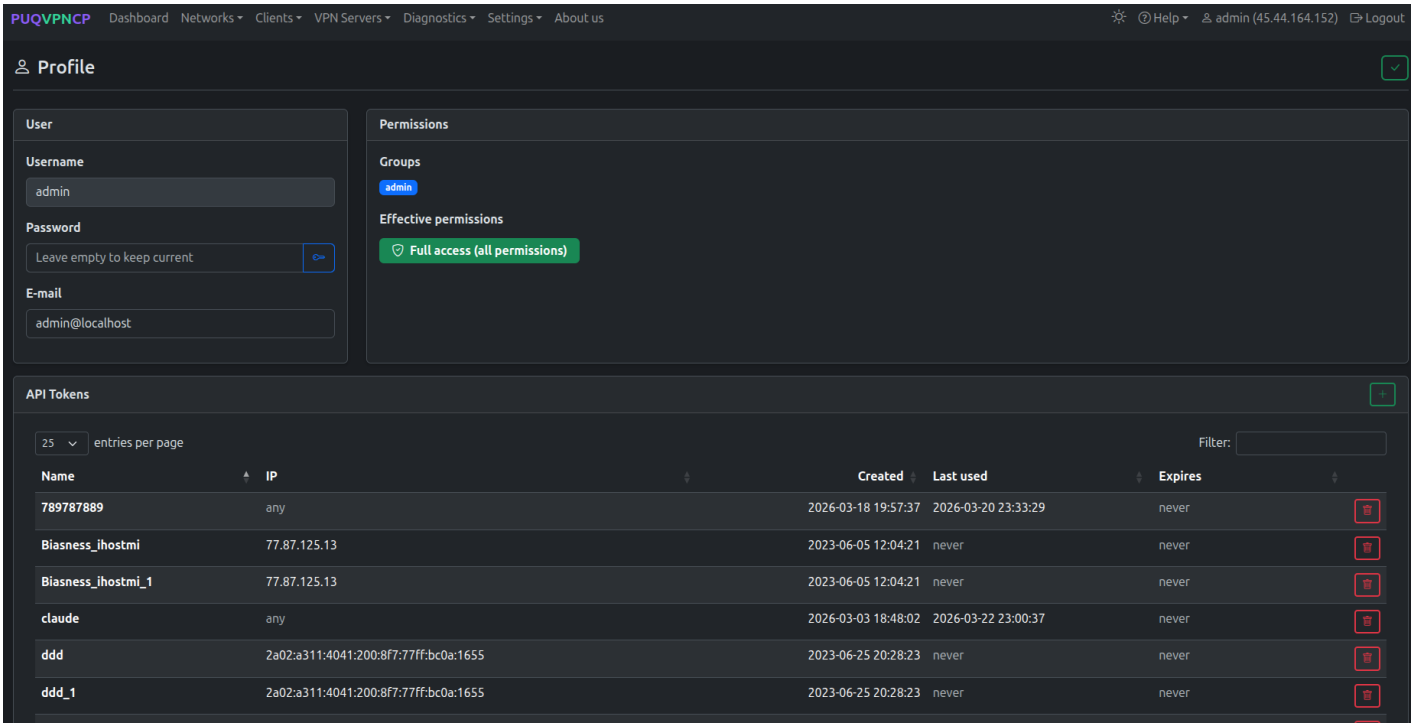
Click your username in the top-right corner of the panel and select **Profile**.



26-puqvpncp-profile-menu.png

Step 2 — Open the API Tokens section

Scroll down to the **API Tokens** card. Click the green **+** button on the right to create a new token.



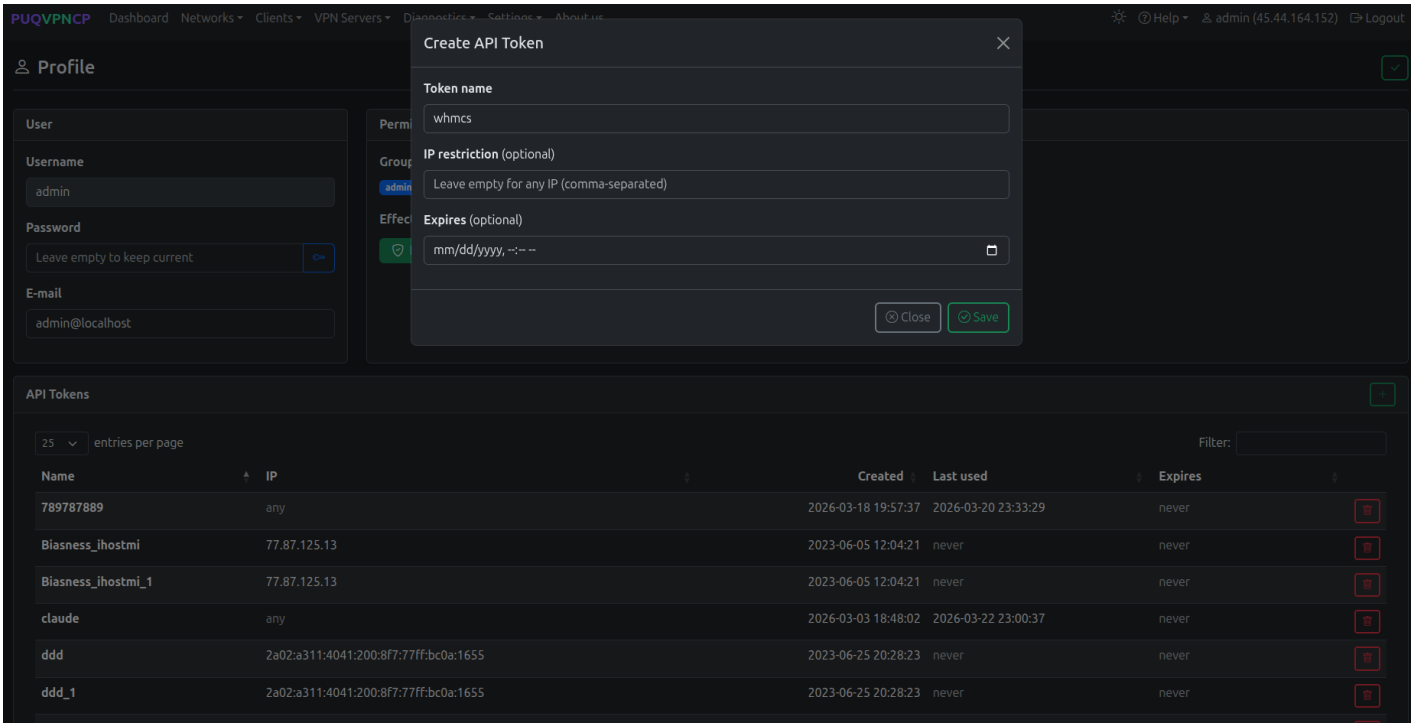
27-puqvpncp-profile-tokens.png

Step 3 — Create the token

Fill in the modal:

- **Token name** — a label that identifies the consumer, e.g. `whmcs`.
- **IP restriction** *(optional)* — a comma-separated list of IPs allowed to use this token. Leave empty to accept the token from any IP, or set it to your WHMCS server's IP for tighter security.
- **Expires** *(optional)* — an expiration date. Leave empty for a non-expiring token.

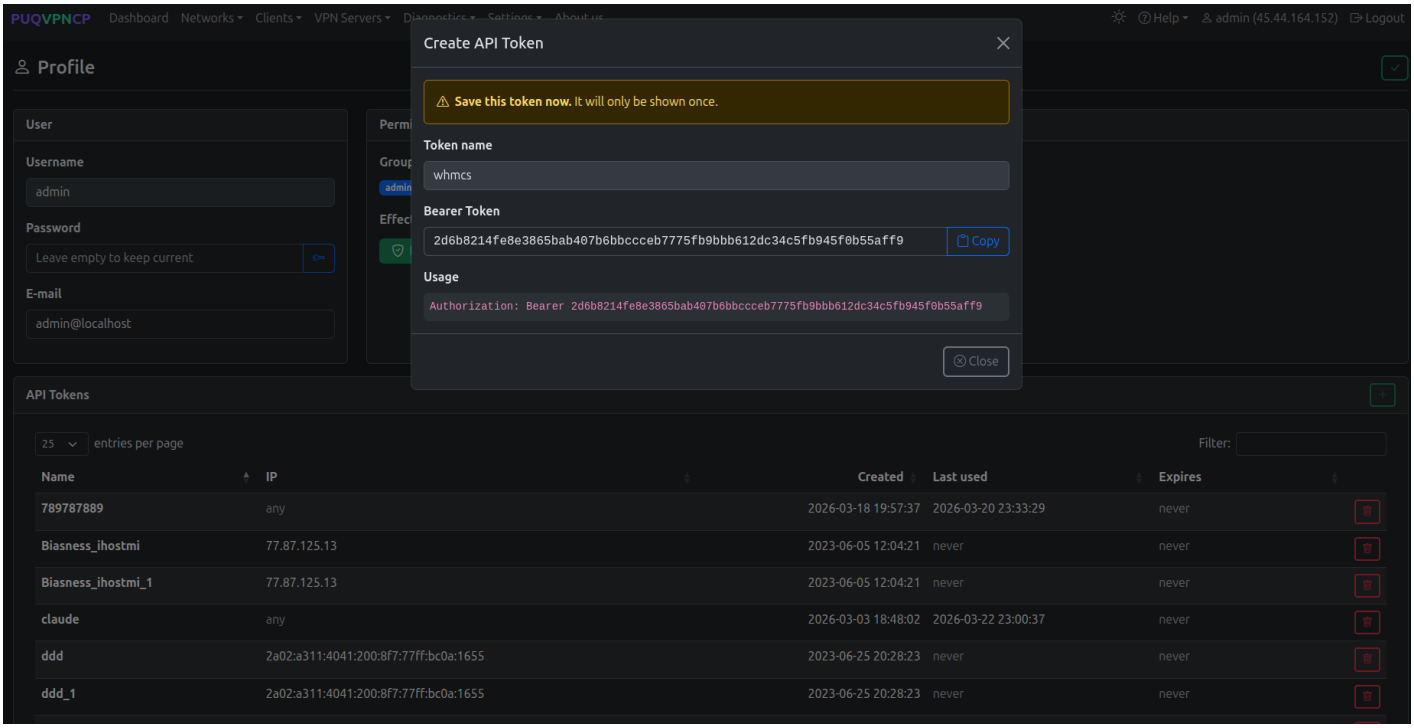
Click **Save**.



28-puqvpncp-token-create.png

Step 4 — Copy the Bearer token

The next dialog shows the **Bearer Token**. Click **Copy** and store it somewhere safe — **the token is shown only once and cannot be retrieved later**. If you lose it, delete the token and generate a new one.



29-puqvpncp-token-bearer.png

> The token grants the user's **effective permissions** — the `admin` group used in the screenshot has full access. For tighter control, create a dedicated user/permission group on the panel and issue a token for that user instead.

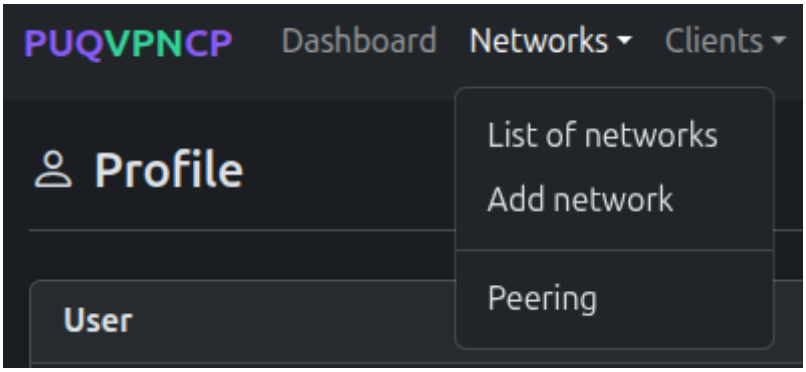
You will paste this token into the **Password** field of the WHMCS server record — see Add server.

---

3. Create a VPN network

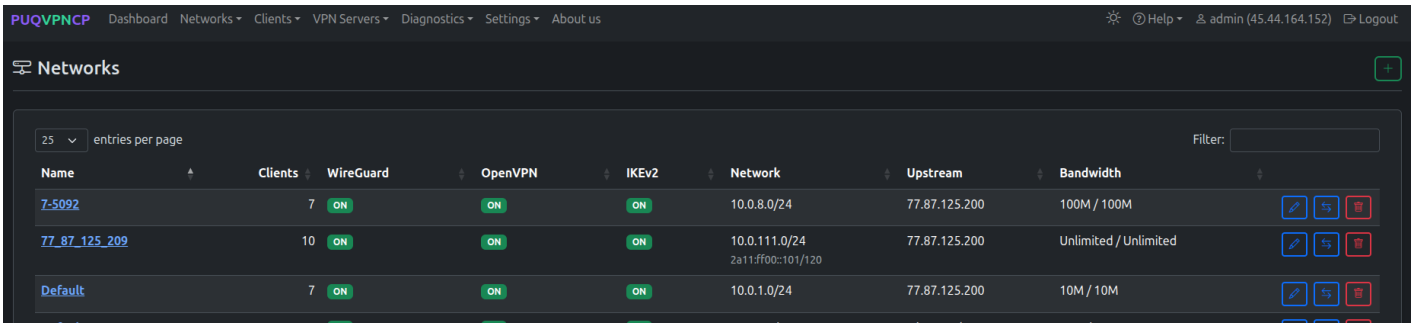
The module needs at least one VPN network on the panel. On the WHMCS product configuration page, every available network is listed as a tickable `server → network` pair.

Step 1 — Open the Networks list



30-puqvpncp-networks-menu.png

The list shows every existing network with the status of WireGuard / OpenVPN / IKEv2, the IPv4 subnet, the upstream interface and the bandwidth caps.



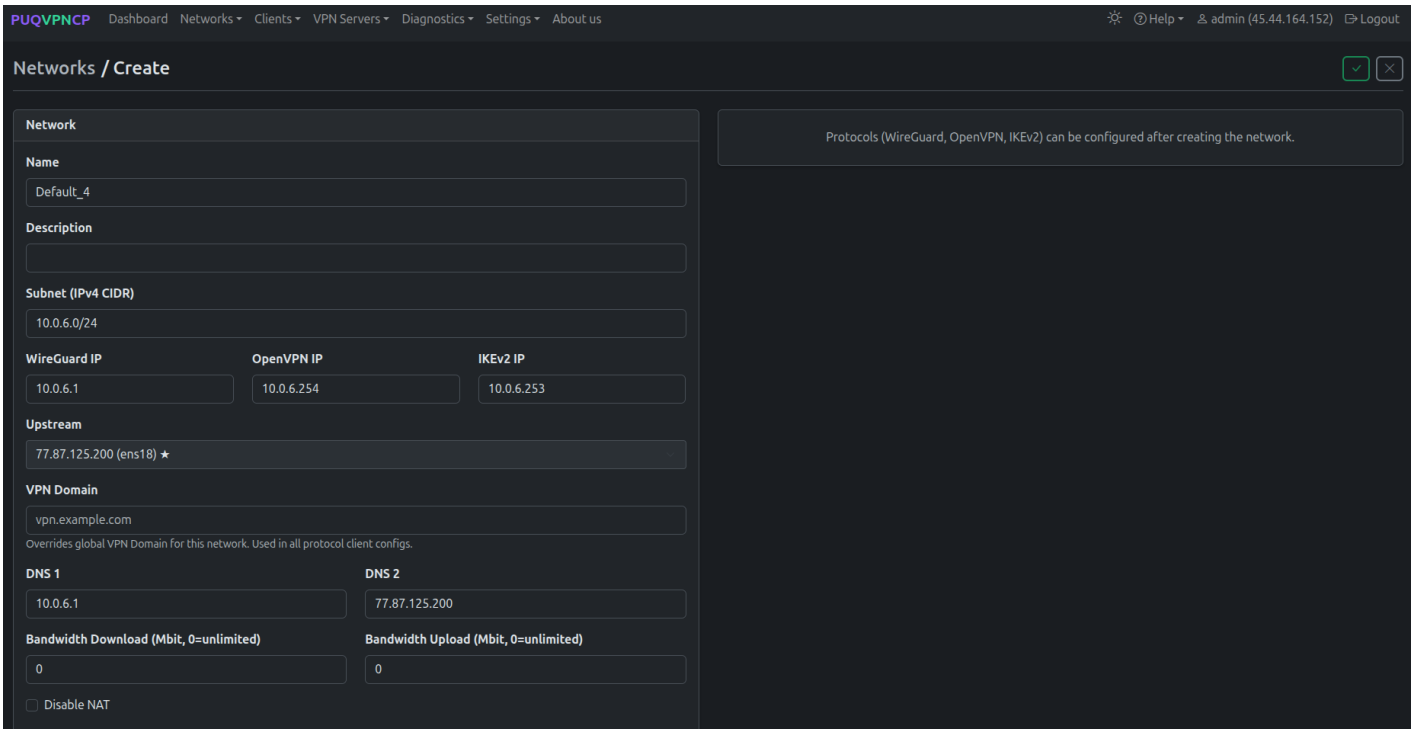
31-puqvpncp-networks-list.png

Step 2 — Add a network

Click the green **+** button in the top-right corner of the Networks page. Fill the **Create** form:

- **Name** — internal identifier of the network (used in the WHMCS product configuration).
- **Description** *(optional)* — human-readable note.
- **Subnet (IPv4 CIDR)** — VPN subnet that will be assigned to clients (e.g. `10.0.6.0/24`).
- **WireGuard IP / OpenVPN IP / IKEv2 IP** — gateway addresses for each protocol inside the subnet.
- **Upstream** — the host network interface used as the egress for this VPN network.
- **VPN Domain** *(optional)* — overrides the global VPN Domain in all client configs for this network.
- **DNS 1 / DNS 2** — DNS servers pushed to clients.
- **Bandwidth Download / Upload** — network-wide caps in Mbit/s (`0` = unlimited).
- **Disable NAT** — leave unchecked unless you route the VPN subnet upstream yourself.

Click the green ✓ in the top-right to save. Protocols (WireGuard / OpenVPN / IKEv2) are configured **after** the network is created.

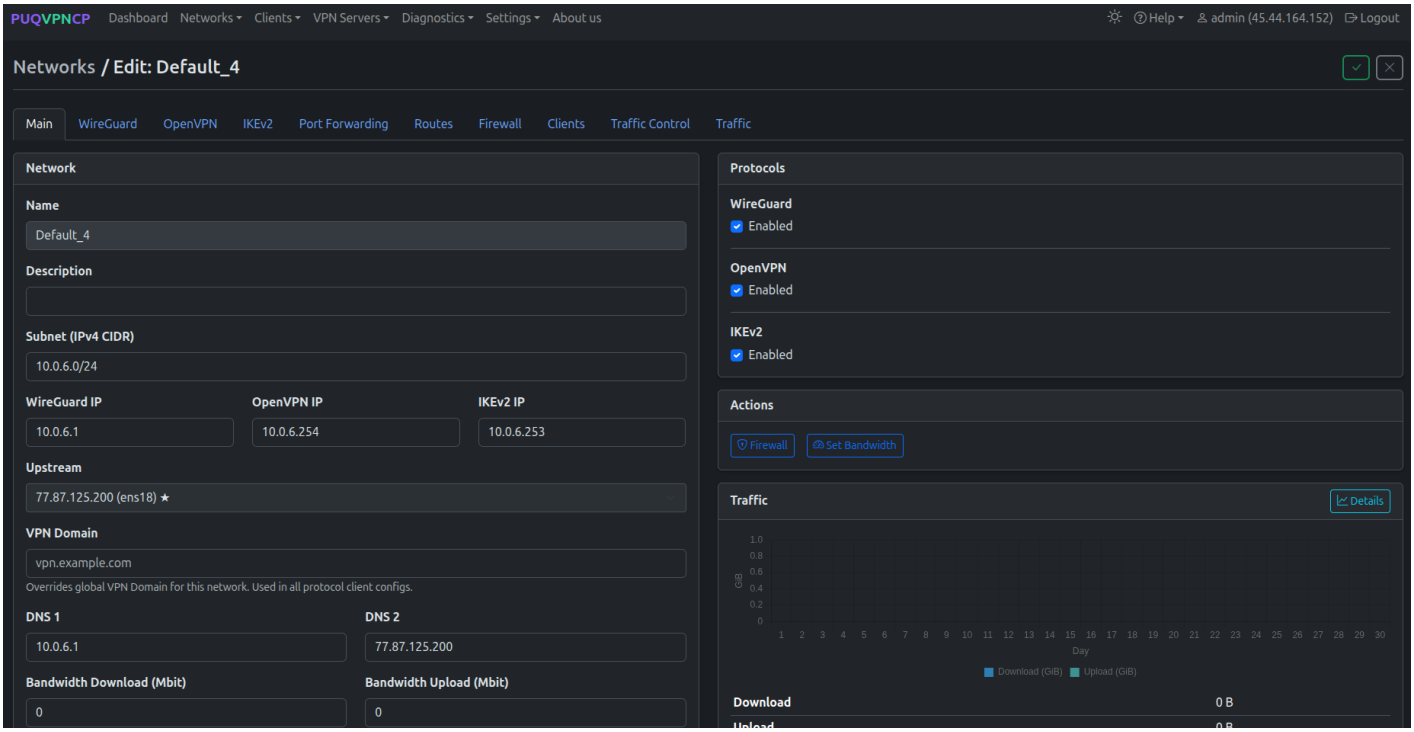


32-puqvpncp-network-create.png

Step 3 — Enable protocols on the network

After saving, you land on the network's **Edit** page with a row of tabs (Main / WireGuard / OpenVPN/ IKEv2 / Port Forwarding / Routes / Firewall / Clients / Traffic Control / Traffic) and a **Protocols**card on the right.

Tick **Enabled** for every protocol you want to offer to customers via WHMCS. The WHMCS module reads this state from `GET /api/v1/network/{name}` — disabled protocols are hidden in the client area and shown greyed-out (with a tooltip) in the admin service tab.



33-puqvpncp-network-edit-protocols.png

Open each protocol-specific tab (**WireGuard**, **OpenVPN**, **IKEv2**) to fine-tune ports, ciphers, MTU and other parameters as needed. Defaults are sensible for most deployments.

What's next

- Add the panel to WHMCS — see Add server.
- Configure a WHMCS product backed by this panel — see Product configuration.

Add server (PUQVPNCP panel)

PUQVPNCP module **WHMCS**

Order now | Download | COMMUNITY | PUQVPNCP

Add a PUQVPNCP panel to WHMCS

Navigate to **System Settings → Servers → Add New Server**.

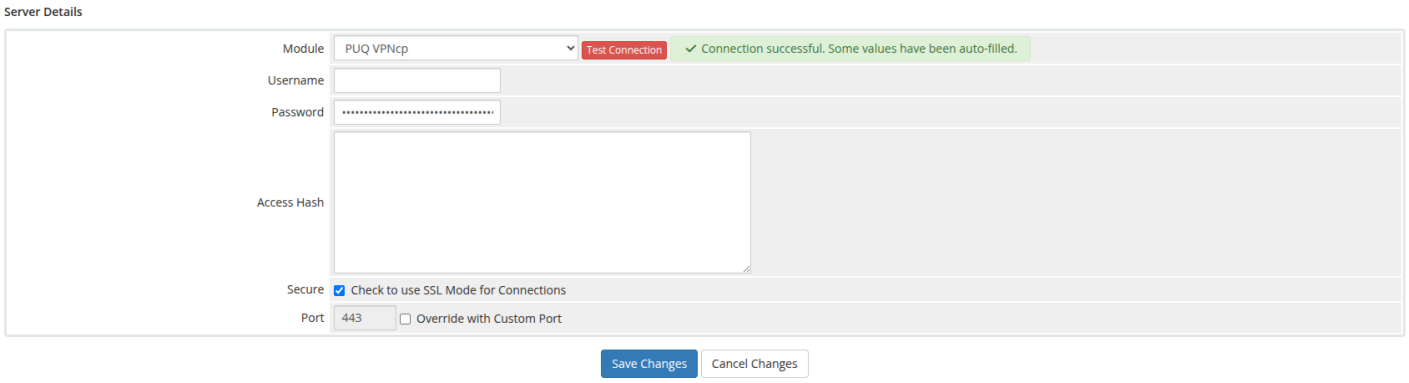
Step 1 — General settings

- **Name** — a descriptive label (e.g. `VPN Frankfurt`).
- **Hostname** — the panel's fully-qualified hostname (e.g. `vpn-fra.example.com`).
- **IP** — optional; used as fallback if hostname is empty.
- **Port** — leave empty for the default `80`/`443`, or enter a custom port.
- **Secure** — check when the panel is served over HTTPS (strongly recommended). SSL verification is enabled when this is checked.

04-add-server-1.png

Step 2 — Module settings

1. Server Details section, **Type** dropdown: select **puqVPNcp**.
2. Leave **Username** empty (not used).
3. Paste the panel's **API token** into the **Password** field — this is what the module sends as `Authorization: Bearer <token>` for every API call.
4. Click **Test connection** — it calls `/api/v1/system/status`, `/api/v1/license` and `/api/v1/network` and returns OK on success.



05-add-server-2.png

> **Important:** The API token must have permissions to manage clients, query networks and read system status.

Step 3 — Assign to a server group

For multi-server deployments, add the server to a WHMCS **server group**. Products bound to that

group will list networks from every reachable server in it on the **VPN Networks** tree of the product configuration page — pick which `server → network` pairs are allowed for that product.

Product configuration

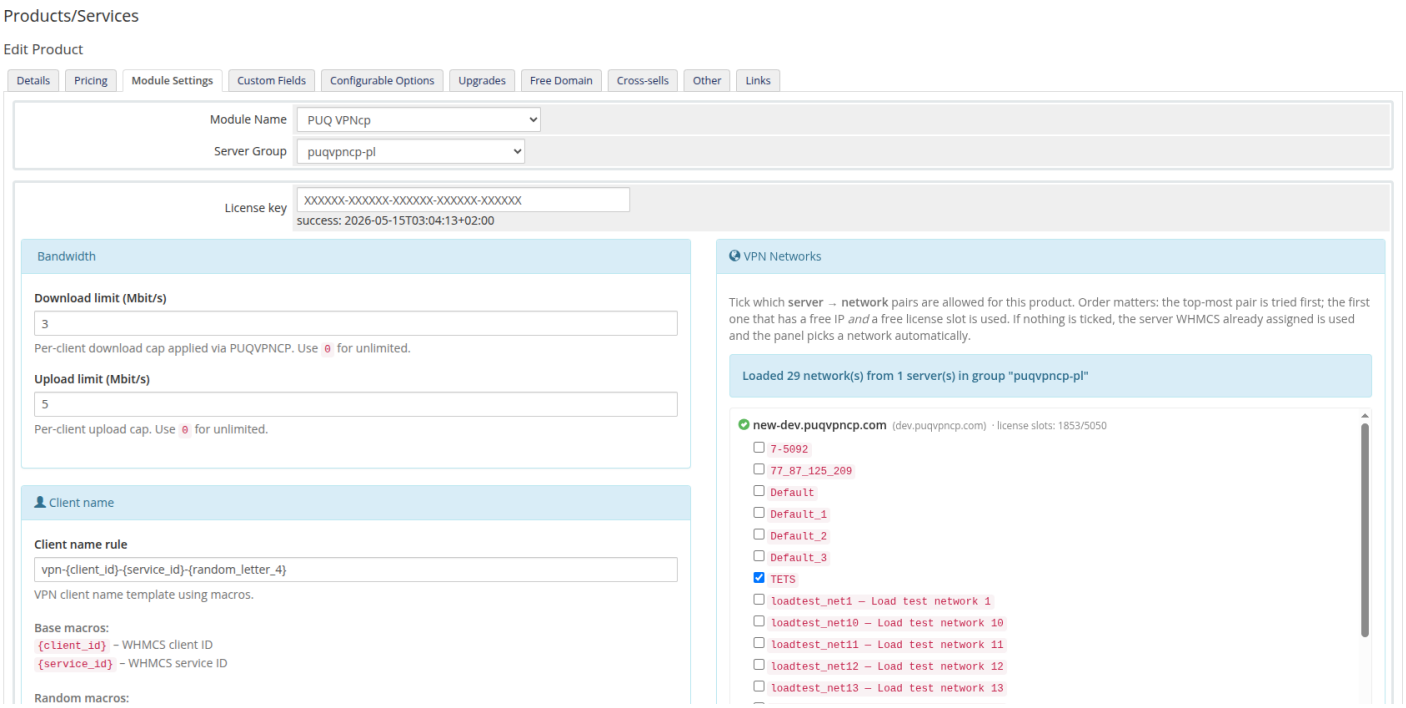
PUQVPNCP module **WHMCS**

Order now | Download | COMMUNITY | PUQVPNCP

Create the product

1. Navigate to **System Settings → Products/Services → Products/Services → Create a New Product**.
2. Product Type: **Other**. Give it a name (e.g. `VPN – 100 Mbit/s`).
3. On the **Module Settings** tab, pick **PUQ VPNcp** as the module and assign a server (or a server group).

After saving, the module injects a **PUQ VPNcp settings** panel below the standard module fields. All settings are persisted as JSON in `configoption24` of the product record.

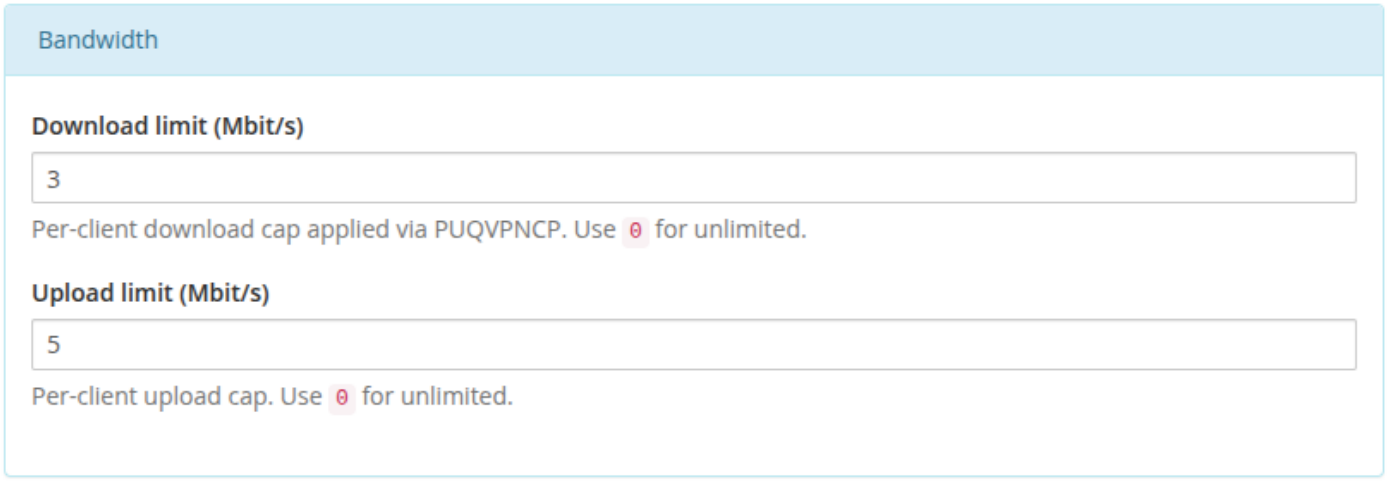


06-product-configuration.png

License key

Paste your licence key into the **License key** field. The status row underneath shows the result of the most recent verification (`success: <timestamp>` or an error). The module re-checks the licence on every product page render and caches the result for the validity period encoded in the licence response.

Bandwidth



07-product-config-bandwidth.png

- **Download limit (Mbit/s)** — per-client download cap. `0` = unlimited.
- **Upload limit (Mbit/s)** — per-client upload cap. `0` = unlimited.

The module applies these via `PUT /api/v1/client/{name}` right after creating the client. The `POST /api/v1/client` create call does not accept bandwidth fields — the limits are always pushed through the follow-up update.

If the bandwidth update fails, the freshly created client is **rolled back** with `DELETE /api/v1/client/{name}` so billing never starts charging for an uncapped VPN.

Client name

08-product-config-client-name.png

- **Client name rule** — template used to generate the VPN client identifier. Default: `vpn-{client_id}-{service_id}-{random_letter_4}`.

Available macros:

Base macros

- `{client_id}` — WHMCS client ID
- `{service_id}` — WHMCS service ID

Random macros

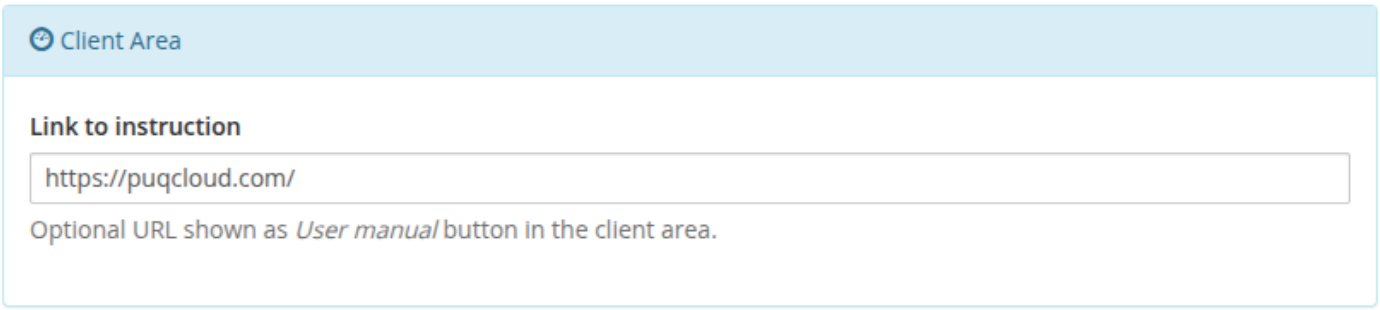
- `{random_digit_N}` — N random digits (e.g. `{random_digit_5}`)
- `{random_letter_N}` — N random lowercase letters

Date & time macros

- `{unixtime}`, `{year}`, `{month}`, `{day}`, `{hour}`, `{minute}`, `{second}`

If the generated name already exists in `tblhosting.username`, the module appends `-1`, `-2`, … until it finds a free one.

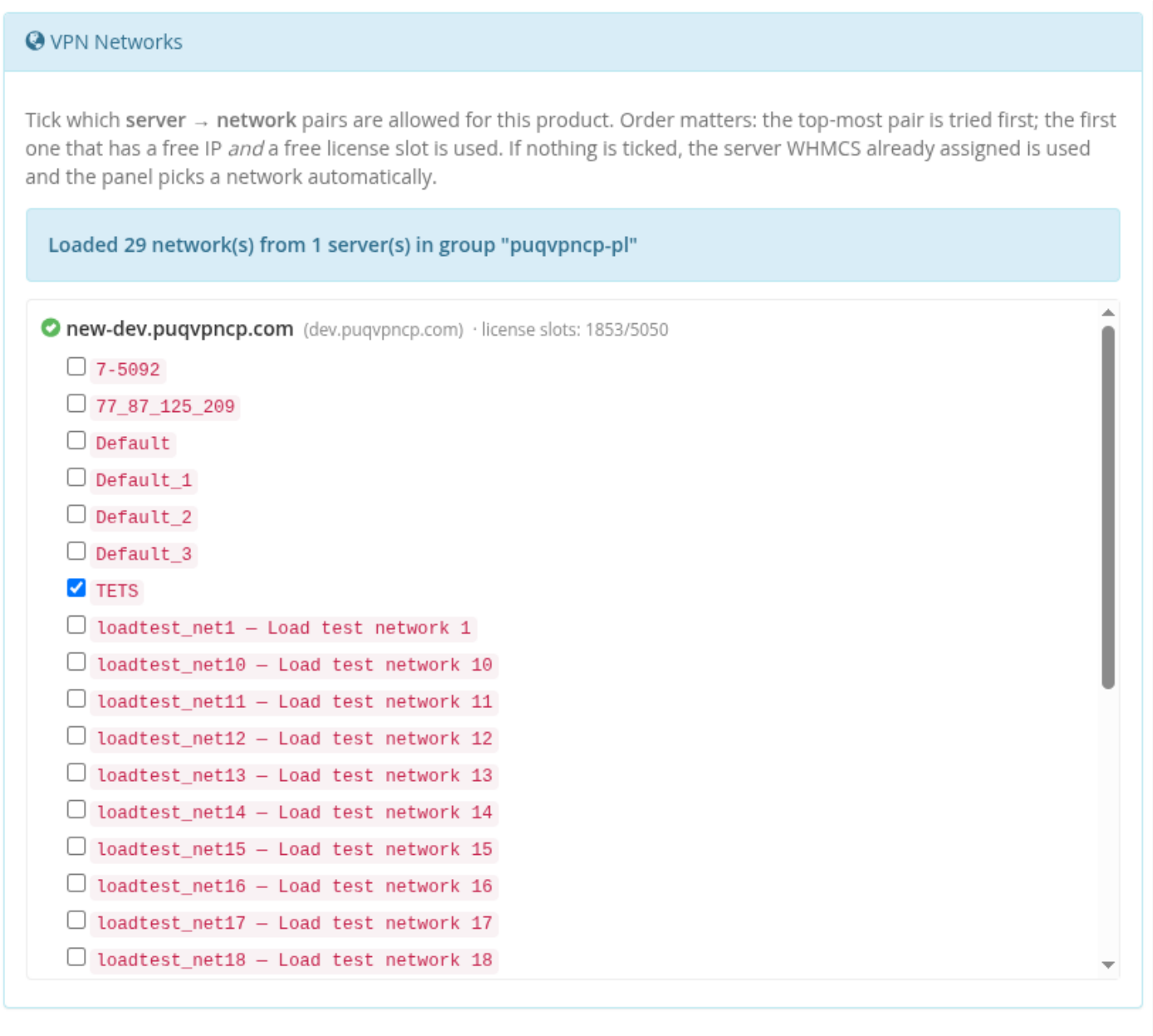
Client Area



09-product-config-client-area.png

- **Link to instruction** — optional URL shown as a **User manual** button at the top of the client-area home screen. Leave empty to hide the button.

VPN Networks



10-product-config-vpn-networks.png

On opening the product, the module contacts **every enabled `puqVPNcp` server in the product's server group** and calls `GET /api/v1/network` on each. The UI then shows a per-server tree — unreachable servers remain visible with their error so you can see exactly what went wrong. License-slot capacity (`used / total`) is displayed next to each reachable server.

Each checkbox is a **`server → network`** pair. Ticking the same network name on two different servers creates two independent pairs.

How a server and network are picked at deploy time

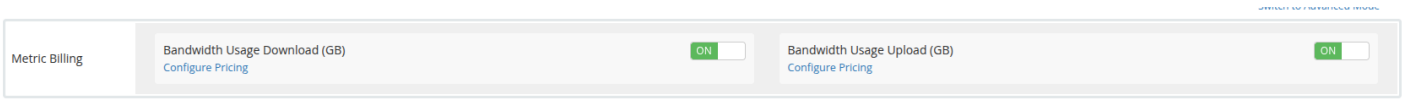
1. The module reads all ticked pairs in the order they appear in the list (top to bottom).

2. For each pair it checks (a) free licence slots on that server (`count_accounts < count_accounts_available` from `GET /api/v1/system/status`) **and** (b) at least one free IP on the network (`GET /api/v1/network/{name}/available_ip`).
3. **The first pair that passes both checks wins.** The service is **reassigned** (`tblhosting.server` is updated) to the selected server, and the client is created on the selected network via `POST /api/v1/client`.
4. If nothing is ticked, `network_name` is omitted from the create call and the panel of the server WHMCS already picked decides automatically.
5. If ticks exist but none is deployable (every chosen server is out of slots or every chosen network is full), provisioning fails — nothing is silently created on a wrong network.

Because order matters, put your **preferred** pairs at the top.

If the whole group is unreachable a red banner appears at the top of the section; previously saved ticks are preserved through hidden inputs so your configuration is not lost when you re-save the product.

Metric Billing (optional)



11-product-config-metric-billing.png

The module ships a WHMCS **Usage Billing** provider with two metrics:

- **Bandwidth Usage Download (GB)**
- **Bandwidth Usage Upload (GB)**

Enable the metrics on the product's **Pricing** tab to charge customers per GB of traffic. The provider pulls daily totals directly from the panel via `GET /api/v1/client/{name}/traffic/{Y}/{m}` and reports them in gigabytes for the current calendar month. No local accumulation table is used — values come live from the panel each time WHMCS runs the usage-billing cron.