Installation and Configuration Guide

Step-by-step instructions for installing and configuring the PowerDNS WHMCS module: preparing the PowerDNS server and its HTTP API, installing or updating the module in WHMCS, adding the PowerDNS server, and configuring the product (zone quota, SOA defaults, name servers, blacklist and zone template).

- WHMCS Installation and Update
- PowerDNS setup (HTTP API)
- Add Server in WHMCS
- Product Configuration

WHMCS Installation and Update

PowerDNS module **WHMCS**

Order now | Download | Community

System requirements

Requirement	Minimum
WHMCS	8.x+, 9.x+
PHP	7.4, 8.1, 8.2, 8.3, 8.4
ionCube Loader	v15+
PowerDNS Authoritative	4.x+ with HTTP API enabled

> **Note:** The module uses ionCube encoding. Make sure ionCube Loader v15 or newer is installed and active on your server.

Installation

> **Note:** The module uses **ionCube 15**, providing universal support across PHP 7.4 through PHP 8.4.

> All versions can be found at this link:
> https://download.puqcloud.com/WHMCS/servers/PUQ_WHMCS-PowerDNS/
>
> Older module versions for WHMCS 8 are available in the archive directory:
> https://download.puqcloud.com/WHMCS/servers/PUQ_WHMCS-PowerDNS/archive/

1. Download the latest release from the PUQ download repository:

```
wget https://download.puqcloud.com/WHMCS/servers/PUQ_WHMCS-PowerDNS/PUQ_WHMCS-PowerDNS-latest.zip
```

2. Unpack the downloaded archive:

```
unzip PUQ_WHMCS-PowerDNS-latest.zip
```

3. Upload the `puqPowerDNS` folder into the `modules/servers/` directory of your WHMCS installation:

```
cp -r puqPowerDNS /path/to/whmcs/modules/servers/
```

4. Ensure proper ownership and permissions for your web server user (`www-data`, `nginx`, or `apache`):

```
chown -R www-data:www-data /path/to/whmcs/modules/servers/puqPowerDNS/
chmod -R 755 /path/to/whmcs/modules/servers/puqPowerDNS/
```

5. The module files are now in place. On first load in WHMCS, the automated migration engine will construct all required tables (`puqPowerDNS_zones`, `puq_license`, `puq_module_versions`). No manual SQL execution is needed.

Update

The update procedure consists of replacing existing module files with the newly downloaded release:

1. Back up your WHMCS database and `modules/servers/puqPowerDNS/` directory.
2. Download and unzip the latest release package.
3. Overwrite the files in `modules/servers/puqPowerDNS/` with the new version files.
4. Access any page in the WHMCS Admin Area. The module migration runner will automatically apply pending schema upgrades and log the new version in `puq_module_versions`.

5. Upgrading from legacy v2.x/v3.x releases is completely safe: existing product configurations are automatically migrated into the unified settings structure.

PowerDNS setup (HTTP API)

PowerDNS module **WHMCS**

Order now | Download | Community

The module communicates with your PowerDNS authoritative server via its built-in **HTTP API**. Before WHMCS can provision and manage DNS zones, you must enable the HTTP API and define an API key on your PowerDNS server.

---

1. Enable the HTTP API in PowerDNS

Open your PowerDNS main configuration file (typically `/etc/powerdns/pdns.conf` or a dedicated snippet file under `/etc/powerdns/pdns.d/`):

```
# Enable the built-in webserver and HTTP API
webserver=yes
webserver-address=0.0.0.0
webserver-port=8081
webserver-allow-from=0.0.0.0/0
api=yes
api-key=YOUR_SECRET_API_KEY_HERE
```

Configuration Parameters:

- `api-key`: A cryptographically random secret string. This value will be entered into WHMCS as the **Server Password**.
- `webserver-port`: Listening port for HTTP API requests (default is `8081`). If using HTTPS reverse proxy (such as Nginx with SSL), use port `443` and enable the **Secure (SSL)** checkbox in WHMCS.
- `webserver-allow-from`: Comma-separated list or CIDR masks of allowed source IP

addresses. Ensure the outbound IP address of your WHMCS server is included here.

Restart the PowerDNS service to apply the configuration:

```
systemctl restart pdns
```

---

2. Verify the HTTP API

From your WHMCS host (or any permitted workstation), test connectivity and authorization using `curl`:

```
curl -s -H 'X-API-Key: YOUR_SECRET_API_KEY_HERE' \
  http://YOUR_PDNS_HOST:8081/api/v1/servers/localhost | jq .
```

A successful response will return a JSON object with server attributes, including the `version` string:

```
{
  "type": "Server",
  "id": "localhost",
  "url": "/api/v1/servers/localhost",
  "daemon_type": "authoritative",
  "version": "4.8.4",
  "zones_url": "/api/v1/servers/localhost/zones{/zone}"
}
```

This is the exact API endpoint WHMCS verifies when you click the **Test Connection** button.

---

3. Master and Zone Behavior Notes

- **Master Zone Architecture**: Zones provisioned by the module are registered as **Master** (`type: "Master"`) with `SOA-EDIT-API = INCREASE`. This ensures that PowerDNS automatically increments the SOA serial number on every record mutation.
- **SOA & NS Origin**: Authoritative name servers and default SOA parameters written to each zone originate strictly from your WHMCS **Product Configuration** (Module Settings), rather than global values in `pdns.conf`.

Add Server in WHMCS

PowerDNS module **WHMCS**

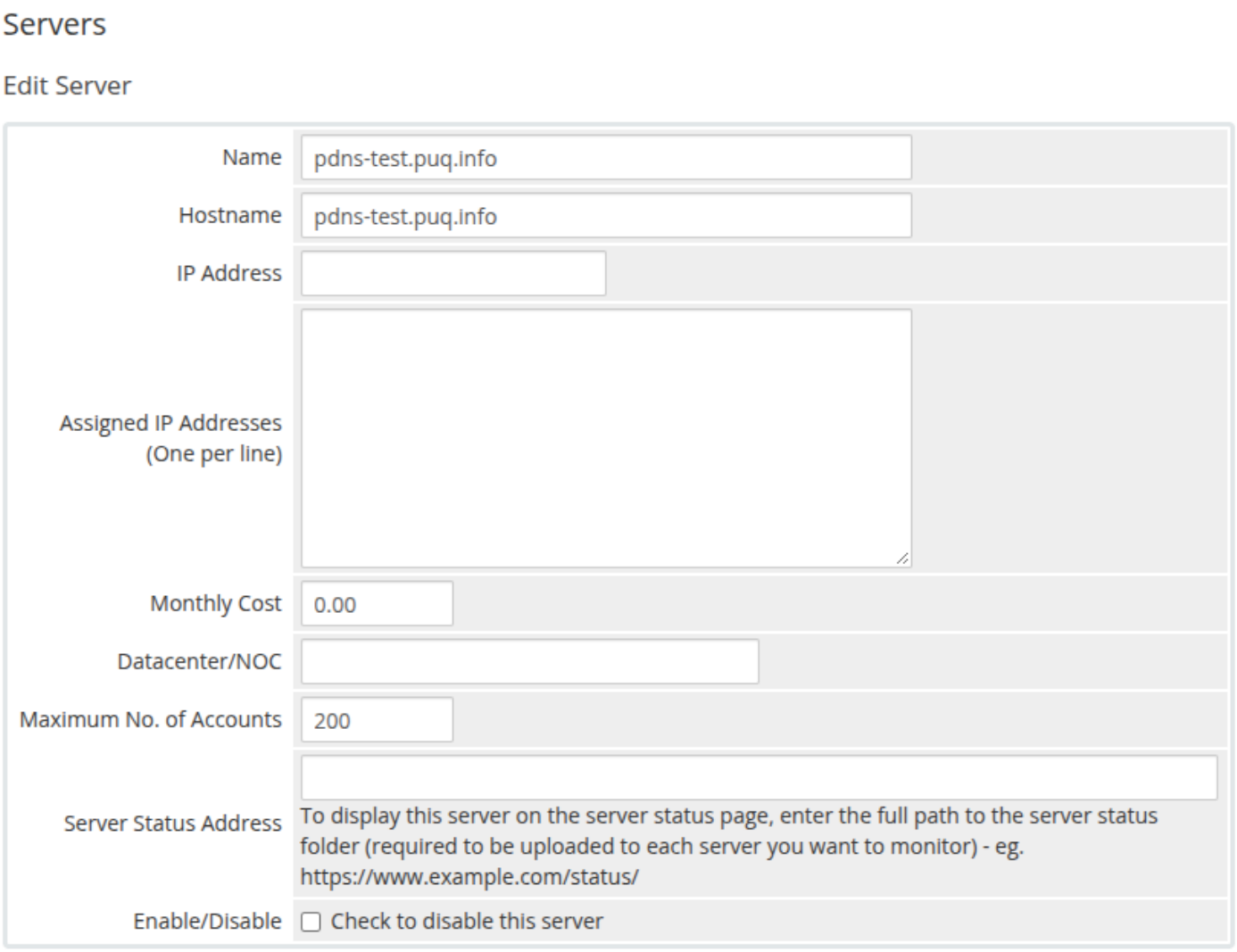
Order now | Download | Community

Register your PowerDNS authoritative server within WHMCS so products can provision against its HTTP API.

---

1. Add New Server

1. In the WHMCS Admin Area, navigate to **System Settings → Servers** (or **Setup → Products/Services → Servers**).
2. Click **Add New Server**.
3. Fill in the primary server details:
   - **Name**: A descriptive label for the server (e.g., `PowerDNS Cluster 1`).
   - **Hostname**: The fully qualified domain name or IP address of the PowerDNS server (e.g., `pdns-test.puq.info`).
   - **IP Address**: The IP address of the PowerDNS API endpoint (optional if hostname resolves).
   - **Max No. of Accounts**: Maximum accounts assigned to this server (e.g., `200`).



admin-server-setup.png

---

2. Server Details and API Credentials

Scroll down to the **Server Details** section:

Field	Required	Description
Type (Module)	Yes	Select **PUQ PowerDNS** from the dropdown list.
Username	No	Leave blank. PowerDNS HTTP API authentication does not use HTTP Basic usernames.

Field	Required	Description
Password	Yes	Enter your PowerDNS `api-key` (from `pdns.conf`). The module transmits this via the `X-API-Key` HTTP header.
Secure	Optional	Check **Tick to use SSL Mode for Connections** if PowerDNS API is secured with HTTPS.
Access Hash	No	Leave blank.

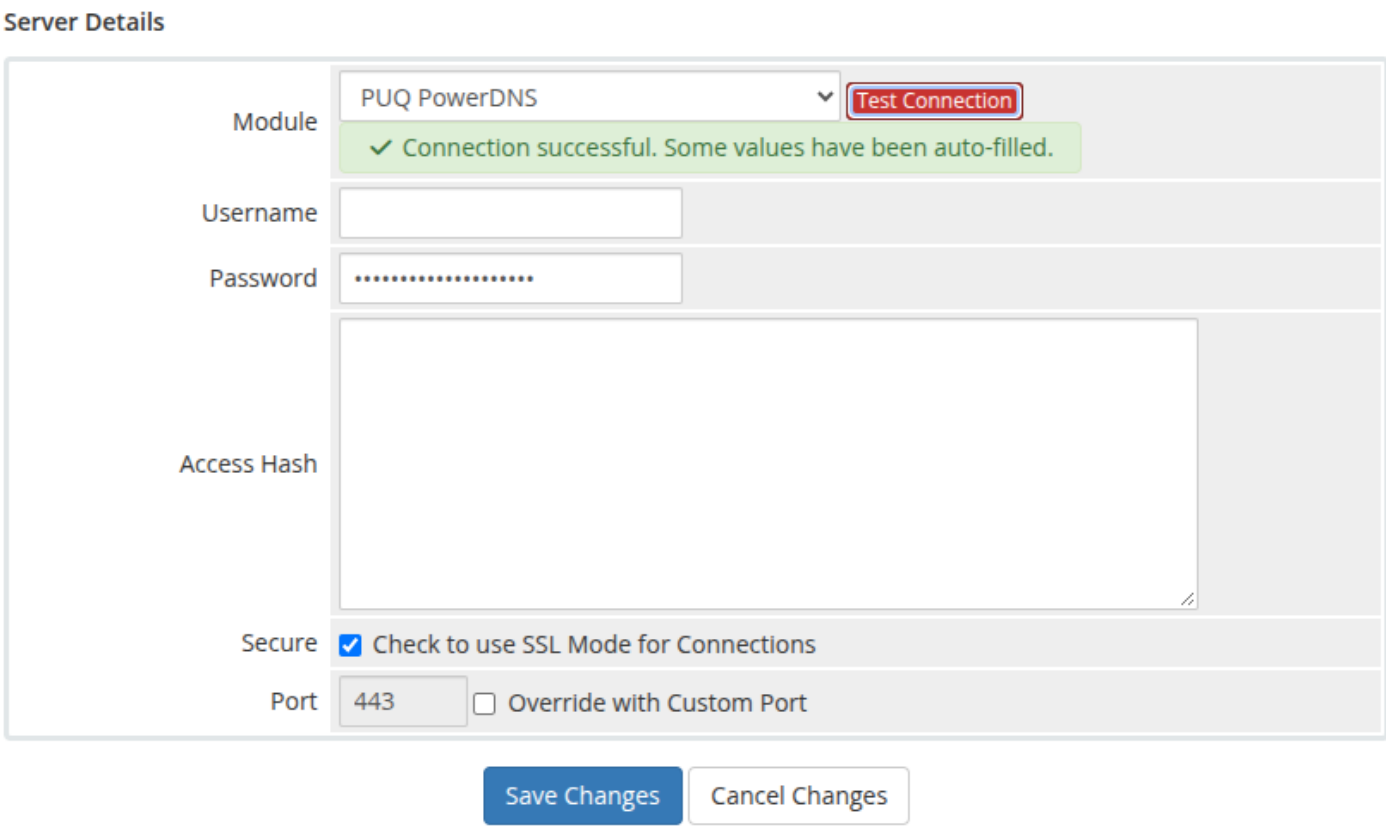
Port Configuration:

- When **Secure (SSL)** is **enabled**, the module connects via HTTPS on port `443` by default.
- When **Secure (SSL)** is **disabled**, the module connects via HTTP on port `8081` by default.
- You can override the port by specifying a custom port number in the server settings or hostname (e.g. `pdns.example.com:8443`).

---

3. Verify Connection

1. Click the **Test Connection** button.
2. The module calls `GET /api/v1/servers/localhost` on your PowerDNS server.
3. Upon success, WHMCS displays a green notification:
   "Success! The test connection was successful."
4. Click **Save Changes** to commit the server configuration.



admin-server-details-test-connection.png

---

4. Server Groups (Optional)

If you operate multiple PowerDNS nodes or clusters:

1. Navigate to **System Settings → Servers → Create New Group**.
2. Create a Server Group (e.g., `PowerDNS US Cluster`), choose your fill method (Add to least full server, or Fill active server until full), and add your PowerDNS servers.
3. Assign this server group to your products under **Module Settings**.

Product Configuration

PowerDNS module **WHMCS**

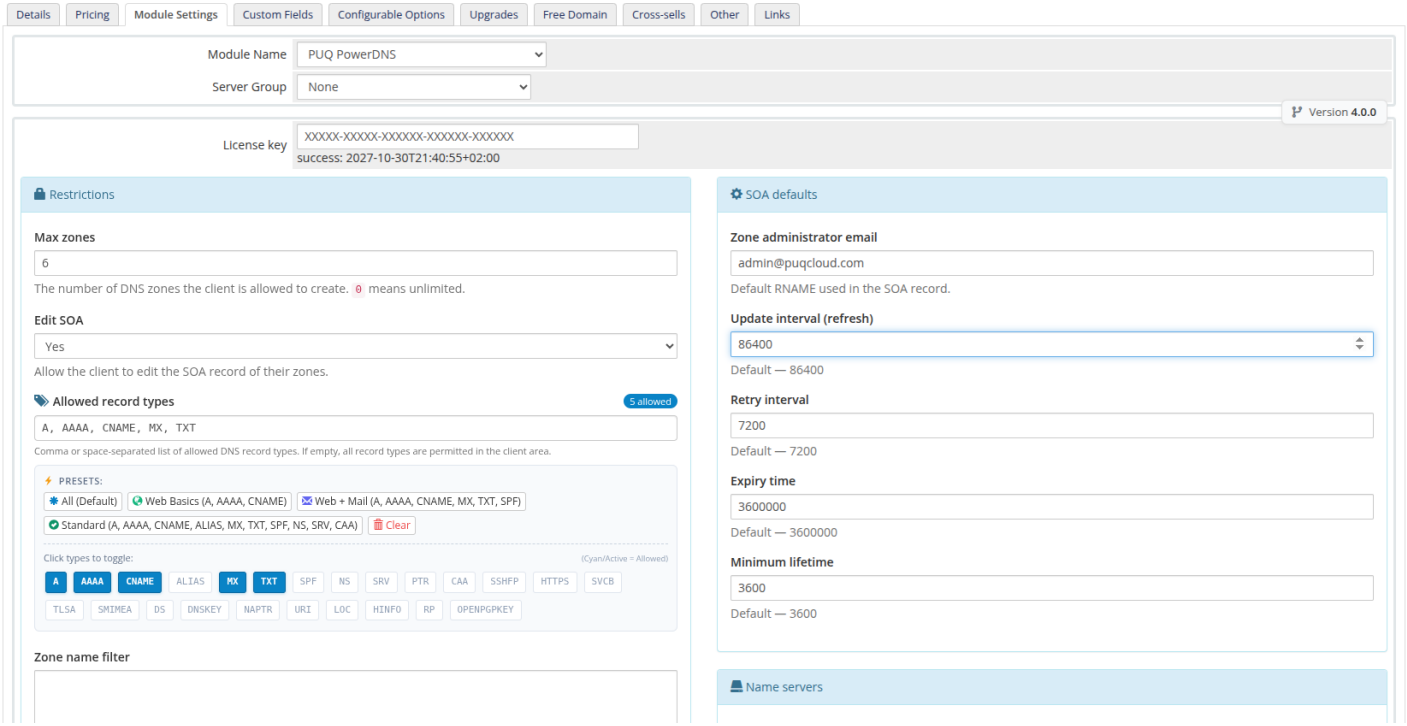
Order now | Download | Community

Create and configure DNS hosting products using the **PUQ PowerDNS** provisioning module. All product-level policies, quotas, record filters, and templates are managed under the product's **Module Settings** tab.

---

1. Create Product in WHMCS

1. In the WHMCS Admin Area, navigate to **System Settings** → **Products/Services** → **Create a New Product**.
2. Select **Product Type**: `Other` (or `Shared Hosting` / `Server/VPS`).
3. Enter the **Product Name** (e.g., `DNS Hosting Premium`), assign a **Product Group**, and configure pricing.
4. Switch to the **Module Settings** tab.
5. In **Module Name**, select **PUQ PowerDNS**.
6. Under **Server Group** (or individual server), choose your PowerDNS server.
7. Enter your **License key** in the designated field.
8. The unified module configuration form loads automatically beneath the standard fields.



admin-product-module-settings.png

---

2. Restrictions Configuration

The **Restrictions** panel defines client boundaries, zone quotas, and permissible DNS record types:

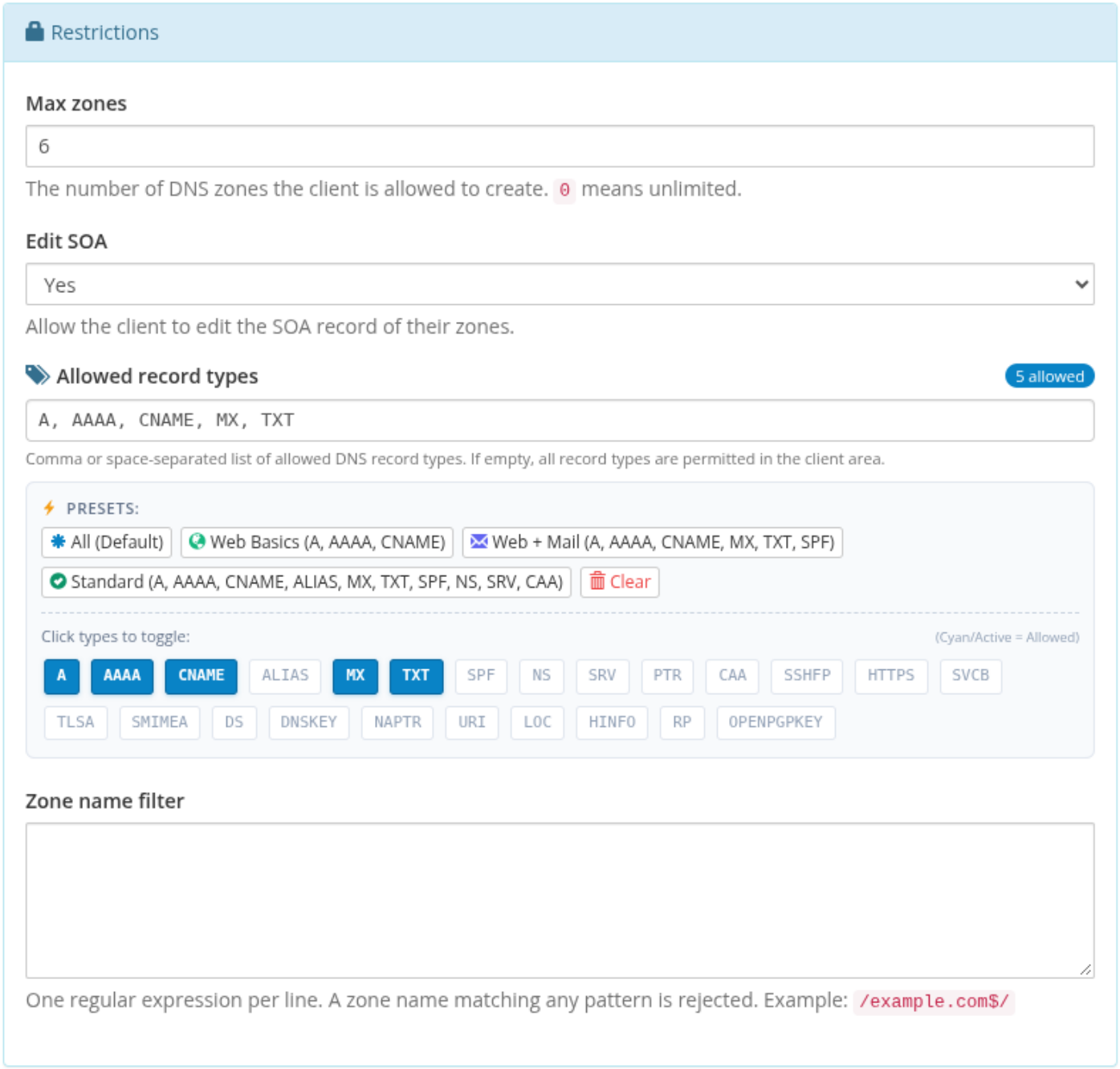
Setting	Type	Description
Max zones	Integer	Maximum number of DNS zones the client can create for this service. Enter `0` for unlimited zones.
Edit SOA	Dropdown	`Yes` allows clients to edit the SOA email and TTL from the client area. `No` locks the SOA record.
Allowed record types	Text / Badges	Comma-separated list of permitted record types (e.g., `A, AAAA, CNAME, MX, TXT`). If left completely empty, **all supported record types** (plus `SOA` if Edit SOA is enabled) are permitted.
Zone name filter	Textarea	Regular expressions (one per line) matched against prospective zone names. If a submitted domain matches any regex, creation is blocked. Example: `/(^|\.)internal\.corp$/i`.

Allowed Record Types & Quick Presets:

To streamline configuration, the interface provides quick preset buttons and interactive clickable badges:

- **All (Default)**: Clears the restriction filter so all supported DNS record types (`A, AAAA, AFSDB, ALIAS, CAA, CERT, CDNSKEY, CDS, CNAME, DNSKEY, DNAME, DS, HINFO, HTTPS, KEY, LOC, LUA, MX, NAPTR, NS, NSEC, NSEC3, NSEC3PARAM, OPENPGPKEY, PTR, RP, RRSIG, SMIMEA, SPF, SRV, SSHFP, SVCB, TKEY, TLSA, TSIG, TXT, URI`) are permitted.
- **Web Basics**: Quickly applies core web types (`A, AAAA, CNAME`).
- **Web + Mail**: Standard web and email types (`A, AAAA, CNAME, MX, TXT, SPF`).
- **Standard**: Expanded conventional hosting types (`A, AAAA, CNAME, ALIAS, MX, TXT, SPF, NS, SRV, CAA`).
- **Clear**: Clears the input field (which automatically enables all record types by default).
- **Interactive Badges**: Click any individual record badge to toggle it in or out of the allowed list instantly.

When restricted, the client area dynamically limits both the **Add Record** dropdown menu and client-side table filters, and enforces strict backend authorization before persisting records.



admin-product-restrictions.png

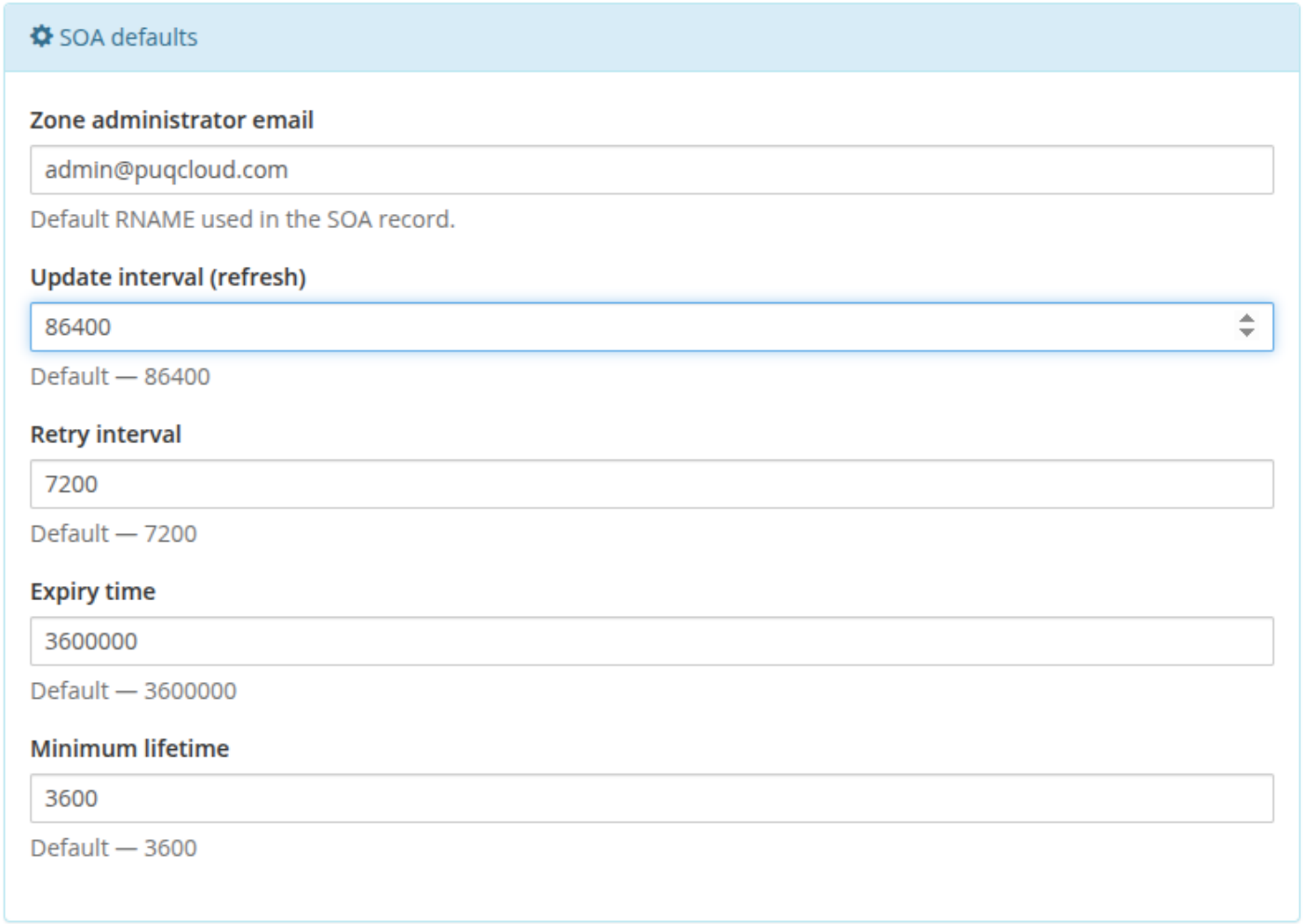
---

3. SOA Defaults

The **SOA defaults** panel defines the default parameters populated into the Start of Authority (SOA) record for every newly generated zone:

Field	Default Value	Purpose
Zone administrator email	`admin@example.com`	RNAME field of the SOA record (dots replaced automatically).

Field	Default Value	Purpose
Update interval (refresh)	`86400` (24h)	Frequency in seconds that secondary name servers query the master for changes.
Retry interval	`7200` (2h)	Frequency in seconds secondary name servers wait before retrying after a failed refresh.
Expiry time	`3600000` (~41 days)	Expiry window in seconds after which secondary servers stop answering authoritative queries if the master is unreachable.
Minimum lifetime	`3600` (1h)	Negative caching TTL for NXDOMAIN responses.



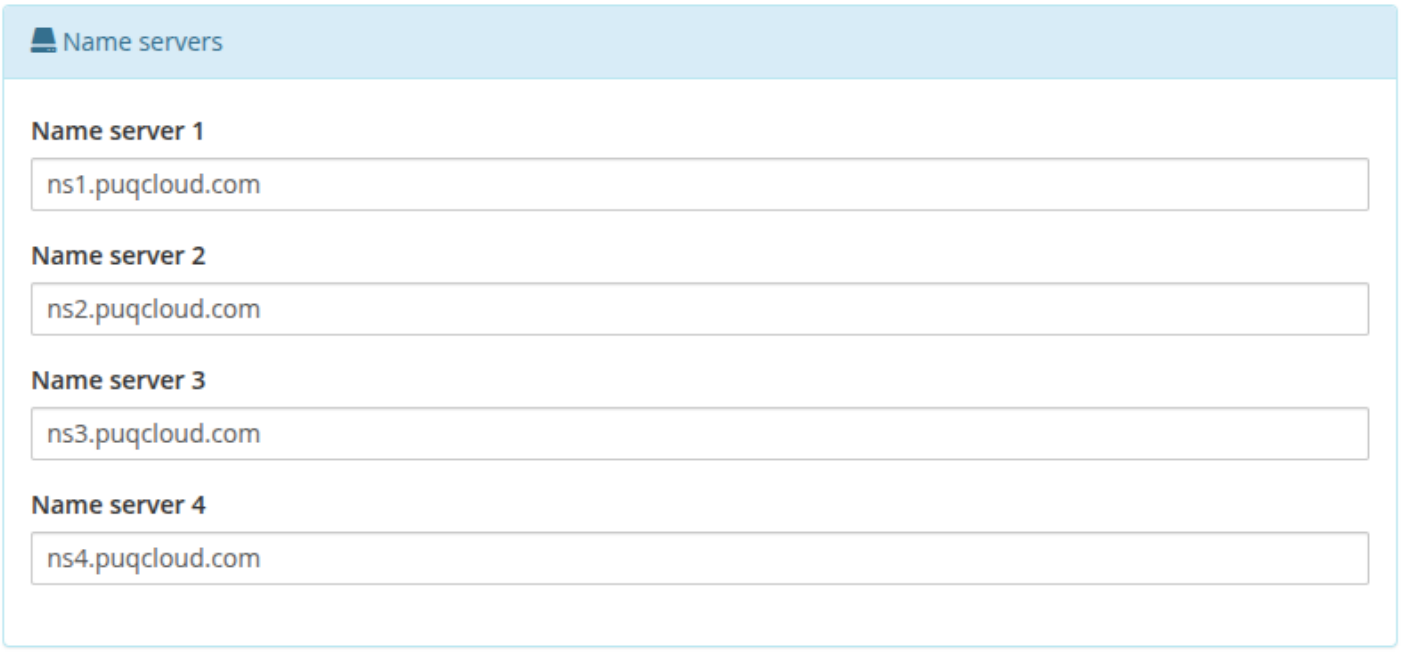
admin-product-soa-defaults.png

---

4. Name Servers

Specify up to four authoritative name servers (**Name server 1** through **Name server 4**):

- **Name server 1** is automatically used as the primary master (MNAME) in the SOA record.
- All declared name servers are automatically generated as authoritative `NS` resource records on every zone creation.
- These name servers are presented to clients on their service dashboard to register at their domain registrars.



admin-product-nameservers.png

---

5. Zone Template

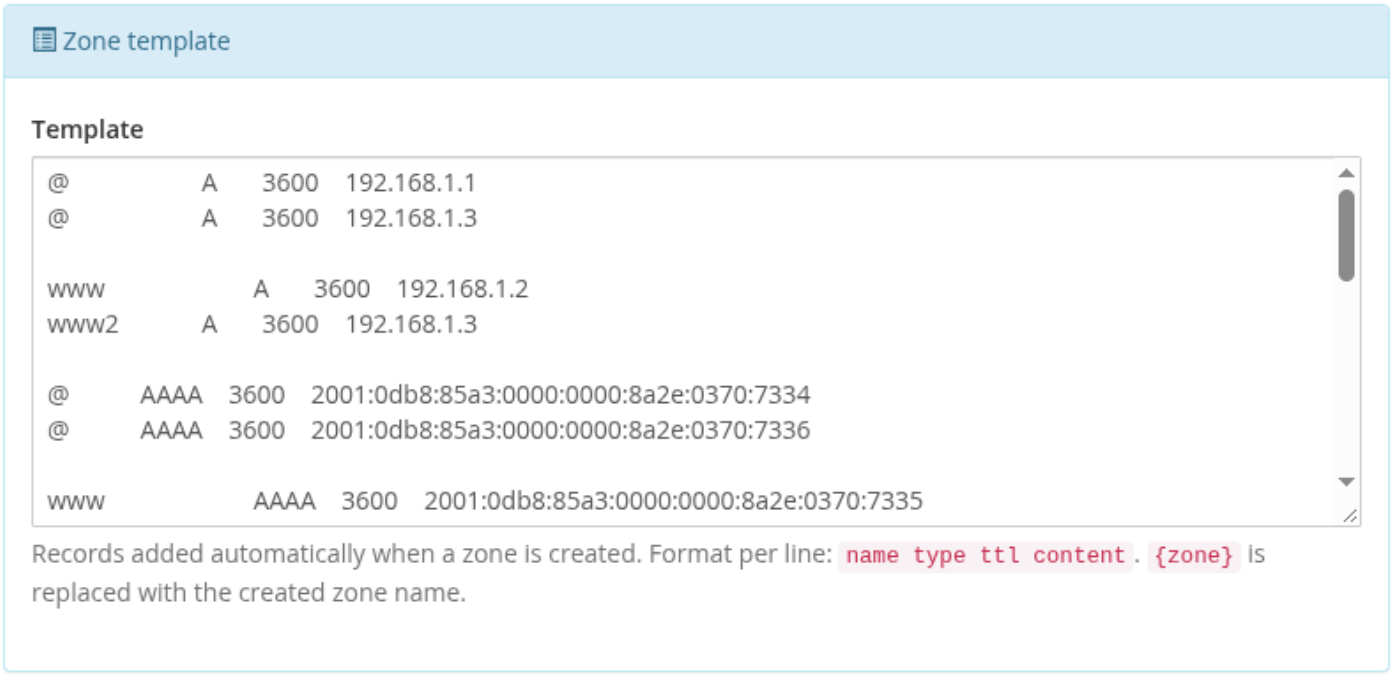
The **Zone template** allows providers to automatically populate standard DNS boilerplate records whenever a client creates a new zone.

- Enter one record per line in the format: `name type ttl content`
- Use the `{zone}` placeholder where the dynamically created zone name should be substituted.
- Use `@` for the zone apex.

Example Template:

```
@ A 3600 198.51.100.10
www CNAME 3600 {zone}
@ MX 3600 10 mail.{zone}
```

```
mail A 3600 198.51.100.25
@ TXT 3600 "v=spf1 mx a ~all"
```



admin-product-zone-template.png

---

6. Backward Compatibility & Automatic Migration

The module features a seamless auto-migration subsystem:

- Legacy products configured with module versions v2.x or v3.x (which utilized raw WHMCS `configoption3`, `configoption4`, `configoption5`, and `configoption7` slots) continue to function without interruption.
- Upon viewing or saving the product, the module automatically reads legacy parameters and migrates them into the unified JSON payload in `configoption24`.