

Settings

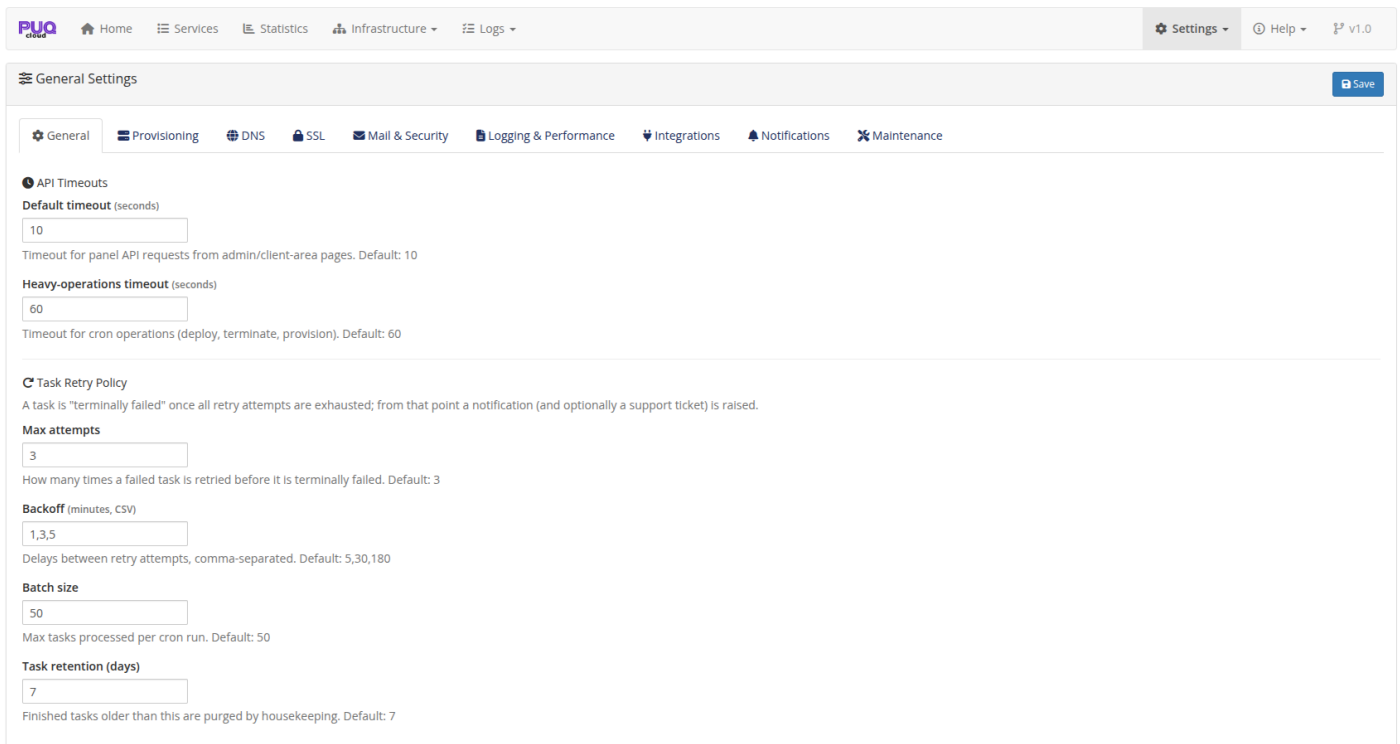
PUQ Web Hosting module **WHMCS**

[Order now](#) | [Download](#) | [Community](#)

Settings has three pages — **General**, **Cron** and **Vanity widget**. The General page is organised into tabs. The defaults are sensible; tune them as you scale.

General

The **General** tab itself holds API timeouts and the **task retry policy** (max attempts, back-off minutes, batch size per cron run, finished-task retention):

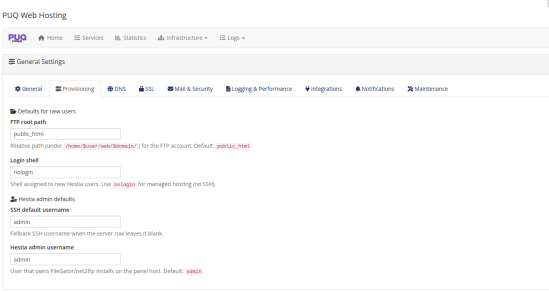
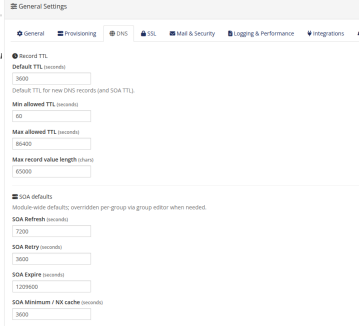
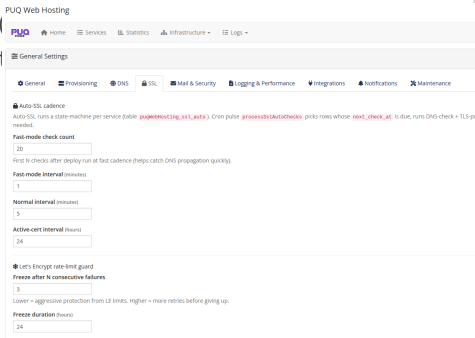
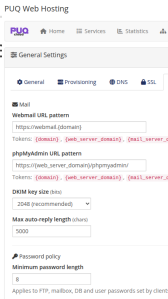


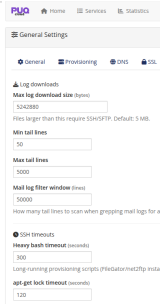
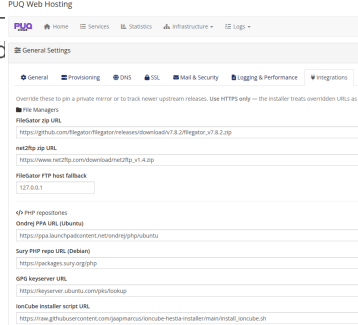
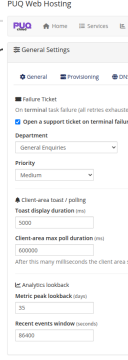
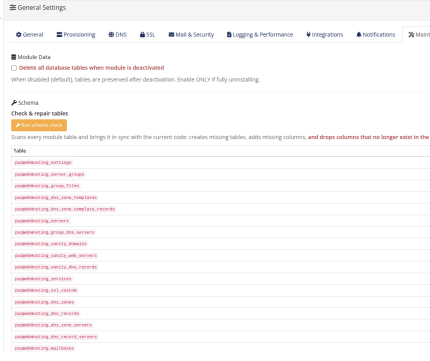
The screenshot shows the WHMCS General Settings page. The top navigation bar includes Home, Services, Statistics, Infrastructure, Logs, Settings, Help, and v1.0. The main content area is titled "General Settings" and has a "Save" button. The settings are organized into tabs: General, Provisioning, DNS, SSL, Mail & Security, Logging & Performance, Integrations, Notifications, and Maintenance. The "General" tab is active and shows the following settings:

- API Timeouts**
 - Default timeout (seconds)**: 10 (Default: 10). Timeout for panel API requests from admin/client-area pages.
 - Heavy-operations timeout (seconds)**: 60 (Default: 60). Timeout for cron operations (deploy, terminate, provision).
- Task Retry Policy**
 - A task is "terminally failed" once all retry attempts are exhausted; from that point a notification (and optionally a support ticket) is raised.
 - Max attempts**: 3 (Default: 3). How many times a failed task is retried before it is terminally failed.
 - Backoff (minutes, CSV)**: 1,3,5 (Default: 5,30,180). Delays between retry attempts, comma-separated.
 - Batch size**: 50 (Default: 50). Max tasks processed per cron run.
 - Task retention (days)**: 7 (Default: 7). Finished tasks older than this are purged by housekeeping.

The remaining tabs:

Tab	Controls
-----	----------

<p>Provisioning</p>	<p>Defaults for <code>nologin</code> for usernames.</p> 
<p>DNS</p>	<p>Record TTL (default/min/max), module-wide SOA defaults.</p> 
<p>SSL</p>	<p>Auto-SSL cadence (interval, active-cert), rate-limit guard.</p> 
<p>Mail & Security</p>	<p>Webmail / phpMyAdmin URL patterns, DKIM keys, auto-reply length, minimum password length.</p> 

<p>Logging & Performance</p>	<p>Log download size cap, tail line limits, mail-log window, SSH heavy-bash & apt-lock timeouts.</p> 
<p>Integrations</p>	<p>Overridable upstream URLs – host fallback, PHP repos (OnDemand), keyserver, IonCube installer.</p> 
<p>Notifications</p>	<p>Failure-ticket toggle (department + priority), client toast/poll durations, analytics lookback windows.</p> 
<p>Maintenance</p>	<p>The schema Check & Repair toggle, data-retention toggle.</p> 

Cron

The **Cron** page picks the cron mode (WHMCS vs Standalone), shows the crontab line to install, and

lists every scheduled task with its enable/interval/last-run/status, plus concurrency limits.

Cron Settings

Cron Mode

WHMCS Hook — Cron tasks run automatically via WHMCS AfterCronJob hook (~every 5 min)

Standalone Cron File — Add cron.php to system crontab (runs every minute, intervals managed internally)

➤ Crontab Setup

Add the following line to your system crontab (`crontab -e`):

```
* * * * * php /home/whmcs-dev-puq-pL/web/whmcs-dev-puq-pL/public_html/modules/addons/puq_web_hosting/cron.php > /dev/null 2>&1
```

CLI usage: `php cron.php --help` for all options. Show status: `php cron.php --list`. Each tick the runner evaluates the interval table below (auto-queues) and then drains the task queue end-to-end.

Task Intervals

Auto-queue jobs that periodically push work into the task queue. Set the interval (minutes) for each, or 0 to disable. Independently, the queue itself is drained end-to-end on every cron tick — there is no separate setting for that.

Task	Interval (min)	Last Run	Status
Per active server, enqueue one light server.collectStatus task (probes + read-only usage list + capacity — single SSH session, no recompute) <code>syncServerStatus</code> — default: 5m	60	2026-06-03 21:11:02	Active
Per active HestiaCP server, enqueue server.processSysQueue (v-update-sys-queue disk + traffic) — disabled by default; opt in when Hestia's own scheduler lag is a problem <code>processSysQueue</code> — default: 0m	0	never	Disabled
Probe awaiting_remote tasks (recompute, backup, restore, SSL, custom): one batched SSH per server, flips finished jobs to success/error <code>processRemoteProbes</code> — default: 1m	1	2026-06-03 21:55:02	Active
Retry tasks stuck in error with attempts < max_attempts (exponential back-off) <code>retryFailedTasks</code> — default: 5m	5	2026-06-03 21:51:02	Active
Send queued client notifications via WHMCS Email Templates (configuration13 mapping) <code>processNotifications</code> — default: 5m	5	2026-06-03 21:51:02	Active
Consolidated housekeeping: sweep stuck DNS deletes (>30 min) + drop rows for terminated services + prune finished tasks/operation_log older than task_retention_days <code>housekeeping</code> — default: 30m	30	2026-06-03 21:47:02	Active
Auto-SSL: DNS-check + TLS probe; conditional Let's Encrypt issue (per-row scheduling — 1 min fast / ssl_normal_check_minutes / 1440 min for active) <code>processSslAutoChecks</code> — default: 1m	1	2026-06-03 21:54:03	Active
Nightly NS / DNS-records verification for active services (stamps verified_at, emits verify_failed event) <code>verifyServices</code> — default: 1440m	1	2026-06-03 21:55:27	Active
Daily expiry scan over customer-uploaded SSL certs; queues ssl_custom_expiring_14 / _3 / _expired <code>sslCustomcheck</code> — default: 1440m	1440	2026-06-02 23:56:22	Active

Concurrency Protection

Lock Timeout (seconds)

600

If a cron process runs longer than this, the lock is considered stale and auto-removed. Default: 600 (10 min).

Vanity widget

Generates the standalone "claim your name" shop widget — see **Vanity Mode** → **The vanity shop widget**.

Revision #8

Created 4 June 2026 15:10:35 by Ruslan

Updated 7 June 2026 20:54:09 by Ruslan