

# How Deployment Works

PUQ Web Hosting module **WHMCS**


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

## Orders return instantly — work happens in the background

When a service is created (a WHMCS order goes active, an admin clicks **Create**, or the API is called) the module does **not** try to build everything inside the HTTP request. It writes a service record, returns `success` to WHMCS immediately, and lets the **cron-driven task queue** do the real work.

This is what the customer sees right after ordering — a live deployment splash that polls for progress and refreshes itself:

- ★ Overview ^
- i Information
- 🔧 Actions ^
- 🔄 Request Cancellation



Vanity  
HOSTING

ACTIVE

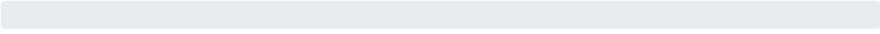
Request Cancellation

Registration Date  
Wednesday, June 3rd, 2026  
Recurring Amount  
\$6.33  
Billing Cycle  
Annually  
Next Due Date  
Thursday, June 3rd, 2027  
Payment Method  
PayPal

- 🌐 Manage
- ⚙️ Configurable Options
- i Additional Information

🚧 Service is being deployed  
Domain: ruslan2.benchwords.com

What's happening now: Configuring web plan



Progress: 0 of 4 steps completed · 4 pending

🕒 This page refreshes automatically. Full deployment typically takes 5-15 minutes.

▶ Recent activity

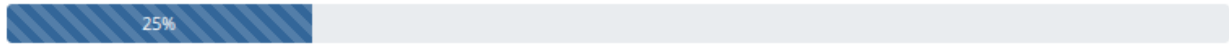
Powered by WHMCompleteSolution

Each step is a queued **task** processed by the cron runner. As steps complete, the bar advances and the recent-activity feed updates:

## Service is being deployed

Domain: ruslan2.benchwords.com

What's happening now: Setting up web account



Progress: 1 of 4 steps completed · 1 in progress · 2 pending

*i* This page refreshes automatically. Full deployment typically takes 5-15 minutes.

Recent activity

Stage	State	Updated
Configuring web plan	done	2026-06-03 23:41:32
Setting up web account	in progress	2026-06-03 23:41:32
Configuring web hosting	queued	2026-06-03 23:41:10
Creating mailbox	queued	2026-06-03 23:41:10

When the last step finishes the page flips to the live control panel:

Manage Configurable Options Additional Information

Service is being deployed  
Domain: ruslan2.benchwords.com

What's happening now: Deployment finished — refreshing...

100%

Progress: 5 of 5 steps completed

This page refreshes automatically. Full deployment typically takes 5-15 minutes.

Recent activity

Stage	State	Updated
Installing SSL certificate	done	2026-06-03 23:42:20
Creating mailbox	done	2026-06-03 23:41:51
Configuring web hosting	done	2026-06-03 23:41:46
Setting up web account	done	2026-06-03 23:41:37
Configuring web plan	done	2026-06-03 23:41:32

This design makes provisioning resilient to slow panels, SSH timeouts, apt-get locks and long operations — none of them block the order, and any failed step is simply retried.

## The pipeline (Split example)

A Split service is built by a chain of idempotent tasks, roughly in this order:

1. **Create the web package + web user** on a Web node.
2. **Add the web domain** (and the web account's quota/PHP settings).
3. **Create the mail user + mail domain** on a Mail node (with DKIM).
4. **Create the DNS user + zone** and **deploy the zone** to every attached DNS node.
5. **Seed Auto-SSL** so Let's Encrypt issues as soon as DNS resolves.

Unified runs the same idea with one combined account; Vanity runs a trimmed chain (one web account, one mailbox on the shared mail user, one DNS record). Every task is **idempotent** — re-running it is safe — so the chain can resume from wherever it stopped.

## Watching and driving it from the

# admin panel

The admin service panel's **Deploy** tab shows the live deploy status and a full **timeline** of every task with its duration and result, plus a **Redeploy (hard reset)** button:

PUQ Web Hosting — admin panel

The screenshot shows the 'Deploy' tab in the admin panel. At the top, there's a navigation bar with various service icons: Overview, Web settings, SSL, FTP (1), Databases (1), DNS (25), Mailboxes (2), Cron (1), Backups (7), Deploy (0), and Tasks (11). Below this, the 'Deploy status' is shown as 'ACTIVE' with a 100% progress bar. There are 'Refresh' and 'Redeploy (hard reset)' buttons. A 'success: 11' indicator is present. The main section is a 'Timeline' showing a list of tasks with their timestamps, names, and durations. The tasks include 'server.recomputeUser', 'service.resyncRole', and 'mail.backupDelete'. At the bottom, there's a table header with columns: Reg Date, Name, Pricing, Status, Next Due Date.

The **Tasks** tab lists this service's tasks (action, server, attempts, timestamps); each row opens a detail modal with the streamed SSH log:

PUQ Web Hosting — admin panel

The screenshot shows the 'Tasks' tab in the admin panel. It features a 'Reload' button and '2 task(s)' count. A 'Show 25 entries' dropdown is visible. A search bar is present. The main content is a table with columns: ID, Action, Target, Server, Status, Attempts, Created, Started, Finished, Error, and Actions. Two task entries are shown, both with a 'SUCCESS' status. The first entry is for 'web.addCron' on server #3, and the second is for 'web.addFTP' on server #3. At the bottom, there's a 'Showing 1 to 2 of 2 entries' indicator and 'Previous' and 'Next' navigation buttons.

ID	Action	Target	Server	Status	Attempts	Created	Started	Finished	Error	Actions
#2568	web.addCron	cron #16	#3	SUCCESS	1/3	2026-06-03 23:06:13	2026-06-03 23:07:24	2026-06-03 23:07:26		[icon]
#2567	web.addFTP	ftp #20	#3	SUCCESS	1/3	2026-06-03 23:05:35	2026-06-03 23:06:03	2026-06-03 23:06:06		[icon]

## Retries, failures and tickets

A failed task is retried with exponential back-off. Once all attempts are exhausted it is **terminally failed**, the service shows a deploy error, and — if you enabled it — the module can open a **support ticket** for your staff. The retry policy and the failure-ticket behaviour are configured in addon **Settings** (see *Cron & Automation* → *Task queue, retries & tickets*).

“ **Key takeaway:** deployment is an asynchronous, idempotent, resumable pipeline. The order is accepted instantly, the queue does the work, failures self-heal on retry, and both you and the customer get live progress.

---

Revision #4

Created 4 June 2026 15:11:41 by Ruslan

Updated 4 June 2026 16:43:20 by Ruslan