Change Package

Proxmox KVM module **WHMCS**

Order now | Download | Community

Overview

A package change (upgrade or downgrade) reconfigures a live VM to match a different product plan — new CPU/RAM values, larger disks, different network settings, changed firewall options. Just like Deploy, this runs asynchronously as a resumable state machine driven by the cron. The client does not wait for all the Proxmox API calls to finish — WHMCS accepts the upgrade order immediately and the module walks the VM through the change over the next one or two cron ticks.

Change Package Pipeline

```
change_package → cp_update_ip → cp_stop → cp_cpu_ram →
cp_system_disk_size → cp_system_disk_bandwidth →
cp_additional_disk → cp_additional_disk_size → cp_additional_disk_bandwidth →
cp_network → cp_firewall → cp_start → ready
```

Step descriptions

State (done)	Next action	What happens
`change_package`	Update IP + DNS + firewall	Reload config, reload remote VM data, update IP allocation if the new package changes IP requirements, refresh anti-spoofing IPSet, and refresh forward + reverse DNS records to reflect any IP changes.

State (done)	Next action	What happens
`cp_update_ip`	Stop VM	Stop the VM. Required because some downstream steps (disk resize, CPU/RAM limits) cannot be applied to a running VM. Polls until the VM reports `stopped`.
`cp_stop`	Set CPU & RAM	Apply new CPU cores / sockets / memory from the new package. Skipped if the values are unchanged.
`cp_cpu_ram`	Resize system disk	Grow the system disk to the new size (Proxmox cannot shrink disks — a smaller target is silently skipped).
`cp_system_disk_size`	System disk I/O	Apply new bandwidth limits to the system disk.
`cp_system_disk_bandwidth`	Additional disk	Create an additional disk if the new package includes one and the VM does not have it yet.
`cp_additional_disk`	Additional disk size	Grow the additional disk.
`cp_additional_disk_size`	Additional disk I/O	Apply new bandwidth limits.
`cp_additional_disk_bandwidth`	Network	Update bridge, VLAN and NIC rate limit.
`cp_network`	Firewall	Re-apply per-product firewall options and refresh the anti-spoofing IPSet with the current IP list.
`cp_firewall`	Start VM	Power the VM back on.
`cp_start`	Verify running	Wait up to 5 ticks for the guest to report `running`.
`ready`	—	Done. VM is live with the new package.

Skip-if-unchanged optimization

Every `cp_*` step first compares the **current** VM configuration with the **target** package configuration. If they match, the step logs `skip (no change)` and advances immediately. A downgrade that only reduces RAM, for example, doesn't touch the disks, network, or firewall — it stops the VM, applies RAM, restarts. In practice most package changes complete in 20-40 seconds real time.

In the log this shows up as lines like `cp_system_disk_size skip → cp_system_disk_bandwidth`. Useful for auditing what actually changed during a given upgrade.

DNS refresh during change package

The first step (`change_package → cp_update_ip`) triggers a full DNS sync — old records for removed IPs are deleted, new records for added IPs are created. All matching DNS zones are updated. With v3.2 each operation is logged live so admins can watch the refresh happen:

```
2026-04-18 18:46:28 PUQ Proxmox KVM Cron Start
2026-04-18 18:46:28 [processVirtualMachines] Running (interval: 1m, last: 18:43:56)

--- ChangePackage: service=5551 vm=2002 status=change_package ---
2026-04-18 18:46:33       · fwd-del OK [hestiacp] puqcloud.com ← 5551-1776530141.puqcloud.com
2026-04-18 18:46:34       · fwd-del OK [powerdns] puqcloud.com ← 5551-1776530141.puqcloud.com
2026-04-18 18:46:35       · rev-del OK [powerdns] 130.168.192.in-addr.arpa ← 192.168.130.2
2026-04-18 18:46:36       · rev-del OK [powerdns] 130.168.192.in-addr.arpa ← 192.168.130.7
2026-04-18 18:46:36       · rev-del OK [powerdns] 130.168.192.in-addr.arpa ← 192.168.130.8
2026-04-18 18:46:37       · rev-del OK [powerdns] 130.168.192.in-addr.arpa ← 192.168.130.9
2026-04-18 18:46:41       · rev-del OK [hestiacp] 130.168.192.in-addr.arpa ← 192.168.130.2
2026-04-18 18:46:42       · rev-del OK [hestiacp] 130.168.192.in-addr.arpa ← 192.168.130.7
2026-04-18 18:46:43       · rev-del OK [hestiacp] 130.168.192.in-addr.arpa ← 192.168.130.8
2026-04-18 18:46:44       · rev-del OK [hestiacp] 130.168.192.in-addr.arpa ← 192.168.130.9
2026-04-18 18:46:46       · rev-del OK [hestiacp] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:000b
2026-04-18 18:46:47       · rev-del OK [hestiacp] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:000c
2026-04-18 18:46:49       · rev-del OK [hestiacp] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:000d
2026-04-18 18:46:50       · rev-del OK [hestiacp] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:000e
2026-04-18 18:46:52       · rev-del OK [hestiacp] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:000f
2026-04-18 18:46:53       · rev-del OK [hestiacp] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:0010
2026-04-18 18:46:55       · rev-del OK [hestiacp] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:0011
2026-04-18 18:46:56       · rev-del OK [hestiacp] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:0012
2026-04-18 18:46:57       · rev-del OK [hestiacp] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:0013
2026-04-18 18:46:58       · rev-del OK [powerdns] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:000b
2026-04-18 18:46:59       · rev-del OK [powerdns] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:000c
2026-04-18 18:46:59       · rev-del OK [powerdns] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:000d
2026-04-18 18:47:00       · rev-del OK [powerdns] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:000e
2026-04-18 18:47:00       · rev-del OK [powerdns] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:000f
2026-04-18 18:47:01       · rev-del OK [powerdns] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:0010
2026-04-18 18:47:01       · rev-del OK [powerdns] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:0011
2026-04-18 18:47:02       · rev-del OK [powerdns] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:0012
2026-04-18 18:47:03       · rev-del OK [powerdns] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:0013
2026-04-18 18:47:08       · fwd OK [hestiacp] puqcloud.com ← 5551-1776530141.puqcloud.com (2 IPs)
2026-04-18 18:47:10       · fwd OK [powerdns] puqcloud.com ← 5551-1776530141.puqcloud.com (2 IPs)
2026-04-18 18:47:11       · rev OK [powerdns] 130.168.192.in-addr.arpa ← 192.168.130.2 → 5551-1776530141.puqcloud.com
2026-04-18 18:47:12       · rev OK [powerdns] 130.168.192.in-addr.arpa ← 192.168.130.7 → 5551-1776530141.puqcloud.com
2026-04-18 18:47:13       · rev OK [powerdns] 130.168.192.in-addr.arpa ← 192.168.130.8 → 5551-1776530141.puqcloud.com
2026-04-18 18:47:15       · rev OK [powerdns] 130.168.192.in-addr.arpa ← 192.168.130.9 → 5551-1776530141.puqcloud.com
2026-04-18 18:47:17       · rev OK [hestiacp] 130.168.192.in-addr.arpa ← 192.168.130.2 → 5551-1776530141.puqcloud.com
2026-04-18 18:47:19       · rev OK [hestiacp] 130.168.192.in-addr.arpa ← 192.168.130.7 → 5551-1776530141.puqcloud.com
```

cron-change-package-dns-refresh.png

`fwd-del OK` / `rev-del OK` lines are the old records being removed; `fwd OK` / `rev OK` lines below are the new records being created. Failures in one provider do not block the rest — DNS errors are non-blocking exactly like they are during deploy.

Full change package walkthrough

A complete upgrade in the cron output, including a step that failed to start the VM on the first attempt and was retried on the next tick:

```
2026-04-18 18:47:10       · fwd OK [powerdns] puqcloud.com ← 5551-1776530141.puqcloud.com (2 IPs)
2026-04-18 18:47:11       · rev OK [powerdns] 130.168.192.in-addr.arpa ← 192.168.130.2 → 5551-1776530141.puqcloud.com
2026-04-18 18:47:12       · rev OK [powerdns] 130.168.192.in-addr.arpa ← 192.168.130.7 → 5551-1776530141.puqcloud.com
2026-04-18 18:47:13       · rev OK [powerdns] 130.168.192.in-addr.arpa ← 192.168.130.8 → 5551-1776530141.puqcloud.com
2026-04-18 18:47:15       · rev OK [powerdns] 130.168.192.in-addr.arpa ← 192.168.130.9 → 5551-1776530141.puqcloud.com
2026-04-18 18:47:17       · rev OK [hestiacp] 130.168.192.in-addr.arpa ← 192.168.130.2 → 5551-1776530141.puqcloud.com
2026-04-18 18:47:19       · rev OK [hestiacp] 130.168.192.in-addr.arpa ← 192.168.130.7 → 5551-1776530141.puqcloud.com
2026-04-18 18:47:22       · rev OK [hestiacp] 130.168.192.in-addr.arpa ← 192.168.130.8 → 5551-1776530141.puqcloud.com
2026-04-18 18:47:25       · rev OK [hestiacp] 130.168.192.in-addr.arpa ← 192.168.130.9 → 5551-1776530141.puqcloud.com
2026-04-18 18:47:27       · rev OK [hestiacp] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:000b → 5551-1776530141.puqcloud.com
2026-04-18 18:47:30       · rev OK [hestiacp] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:000c → 5551-1776530141.puqcloud.com
2026-04-18 18:47:33       · rev OK [hestiacp] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:000d → 5551-1776530141.puqcloud.com
2026-04-18 18:47:36       · rev OK [hestiacp] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:000e → 5551-1776530141.puqcloud.com
2026-04-18 18:47:39       · rev OK [hestiacp] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:000f → 5551-1776530141.puqcloud.com
2026-04-18 18:47:42       · rev OK [hestiacp] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:0010 → 5551-1776530141.puqcloud.com
2026-04-18 18:47:45       · rev OK [hestiacp] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:0011 → 5551-1776530141.puqcloud.com
2026-04-18 18:47:48       · rev OK [hestiacp] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:0012 → 5551-1776530141.puqcloud.com
2026-04-18 18:47:51       · rev OK [hestiacp] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:0013 → 5551-1776530141.puqcloud.com
2026-04-18 18:47:52       · rev OK [powerdns] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:000b → 5551-1776530141.puqcloud.com
2026-04-18 18:47:54       · rev OK [powerdns] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:000c → 5551-1776530141.puqcloud.com
2026-04-18 18:47:55       · rev OK [powerdns] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:000d → 5551-1776530141.puqcloud.com
2026-04-18 18:47:56       · rev OK [powerdns] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:000e → 5551-1776530141.puqcloud.com
2026-04-18 18:47:57       · rev OK [powerdns] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:000f → 5551-1776530141.puqcloud.com
2026-04-18 18:47:58       · rev OK [powerdns] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:0010 → 5551-1776530141.puqcloud.com
2026-04-18 18:48:00       · rev OK [powerdns] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:0011 → 5551-1776530141.puqcloud.com
2026-04-18 18:48:01       · rev OK [powerdns] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:0012 → 5551-1776530141.puqcloud.com
2026-04-18 18:48:02       · rev OK [powerdns] 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa ← 2001:0db8:0000:0000:0000:0000:0000:0013 → 5551-1776530141.puqcloud.com
  [+]  1. Update IP + DNS + Firewall          success (change_package -> cp_update_ip) [93.96s]
  [+]  2. Stop VM                             success (cp_update_ip -> cp_stop) [10.18s]
  [+]  3. Set CPU & RAM                       success (cp_stop -> cp_cpu_ram) [0.14s]
  [+]  4. Resize system disk (skip — no change) success (cp_cpu_ram -> cp_system_disk_size) [0.05s]
  [+]  5. Set system disk I/O (skip — no change) success (cp_system_disk_size -> cp_system_disk_bandwidth) [0.06s]
  [+]  6. Create additional disk (skip — exists) success (cp_system_disk_bandwidth -> cp_additional_disk) [0.05s]
  [+]  7. Resize additional disk (skip — no change) success (cp_additional_disk -> cp_additional_disk_size) [0.05s]
  [+]  8. Set additional disk I/O (skip — no change) success (cp_additional_disk_size -> cp_additional_disk_bandwidth) [0.05s]
  [+]  9. Configure network (skip — no change) success (cp_additional_disk_bandwidth -> cp_network) [0.05s]
  [+] 10. Configure firewall                  success (cp_network -> cp_firewall) [0.06s]
  [!] 11. Start VM                            VM failed to start (status: stopped). Will retry. (cp_firewall -> cp_firewall) [5.11s]
--- ChangePackage paused: waiting at [cp_firewall] ---
2026-04-18 18:48:28 [processVirtualMachines] Done (119.8s) — processed: 1, errors: 1
2026-04-18 18:48:28   sid:5551 action:change_package [VM failed to start (status: stopped). Will retry.]
2026-04-18 18:48:28 [removeOldSnapshots] Running (interval: 1m, last: 18:43:56)
2026-04-18 18:48:28 [removeOldSnapshots] Done (0s) — processed: 0, errors: 0
```

cron-change-package-steps-with-retry.png

Successful steps show `success`, skipped steps show `skip (no change)`, and the failed `cp_start` step gets retried until it succeeds. At no point is the state machine forced to restart from the beginning — retry only re-runs the step that didn't complete.

Log viewing

Every change package run writes a structured entry to `vm_last_action_log` on the VM record. In the addon's VM Management the **Log** modal shows each step with its duration, result, and any skip markers:

```
2026-04-10 03:54:38 PUQ Proxmox KVM Cron Start
2026-04-10 03:54:38 [processVirtualMachines] Running (interval: 1m, last: 03:53:51)

--- ChangePackage: service=5546 vm=2002 status=cp_firewall ---
  [+]  1. Start VM                            success (cp_firewall -> cp_start) [0.05s]
  [+]  2. Verify running                      success (cp_start -> ready) [0.05s]
--- ChangePackage complete: service=5546 status=ready ---
2026-04-10 03:54:40 [processVirtualMachines] Done (2.1s) — processed: 1, errors: 0
2026-04-10 03:54:40   sid:5546 action:change_package [success]
2026-04-10 03:54:40 [removeOldSnapshots] Running (interval: 1m, last: 03:53:51)
2026-04-10 03:54:40 [removeOldSnapshots] Done (0s) — processed: 0, errors: 0
2026-04-10 03:54:40 [restoreBackupStatus] Running (interval: 1m, last: 03:53:51)
2026-04-10 03:54:40 [restoreBackupStatus] Done (0s) — processed: 0, errors: 0
2026-04-10 03:54:40 [nowBackupStatus] Running (interval: 1m, last: 03:53:51)
2026-04-10 03:54:40 [nowBackupStatus] Done (0s) — processed: 0, errors: 0
2026-04-10 03:54:40 [scheduleBackup] Running (interval: 1m, last: 03:53:51)
2026-04-10 03:54:40 [scheduleBackup] Done (0s) — processed: 0, errors: 0
2026-04-10 03:54:40   sid:5546 [skipped] already_done_today, last:2026-04-10
2026-04-10 03:54:40 [collectingStatistics] Running (interval: 60m, last: 03:53:51)
2026-04-10 03:54:40 [collectingStatistics] Done (0s) — processed: 1, errors: 0
2026-04-10 03:54:40   sid:5546 [success] last:never, collected
2026-04-10 03:54:40 PUQ Proxmox KVM Cron End (2.2s total)
```

cron-change-package-log-success.png

If the last run had any failures the modal shows a red error banner at the top with the failure reason.

Retry semantics

Change package uses the same "no retry limit, no time bomb" design as deploy:

- A failed step keeps the VM in its current `cp_*` state.
- The next cron tick retries **only that step**, not the whole pipeline.
- Earlier successful steps are never repeated — disk resizes, for example, are not redone on retry.
- A persistent failure is visible in every cron log entry until an admin addresses the root cause.

During the `cp_stop` → `cp_start` window the VM is offline for as long as the hardware changes take. For most upgrades this is under a minute. For large disk resizes it can be longer — Proxmox needs to finish the storage operation before `cp_start` can proceed.

What triggers a change package

A package change starts when `vm_status` is set to `change_package`. That happens when:

- A client completes an upgrade/downgrade order in the client area.
- An admin clicks **Change Package** in the service module commands.
- `ChangePackage` is called through the WHMCS API.

The module verifies the current `vm_status` is either `ready` (normal path) or already `change_package` (idempotent) before setting state. An in-progress deploy or terminate will be respected — the change package request waits until the VM returns to `ready`.

Caveats

- **Disks cannot be shrunk.** Proxmox does not support shrinking virtual disks safely. A downgrade to a smaller disk size logs "skip (new size is smaller)" and keeps the existing larger disk. Billing is unaffected — WHMCS tracks the package, not the disk size on disk.
- **VM must be healthy to stop gracefully.** If the guest OS is unresponsive, `cp_stop` may take longer and eventually force-stop.
- **Cross-node migration during upgrade is not performed.** The VM stays on its current node. If you need to move a VM to a different node during an upgrade, do the

migration separately in Proxmox first.

Related reading

- Deploy Process — first-time provisioning using the same state-machine pattern.
- Terminate Process — async service teardown.
- DNS Zones & Integration — how the DNS refresh during `change_package → cp_update_ip` works.

Revision #13
Created 10 April 2026 12:08:17 by Ruslan
Updated 8 October 2026 19:21:12 by Ruslan