TECHNICAL BRIEF

How Hydra processes client requests

A system-level view of policy, DNS, flow assignment, isolated VPN routing, lifecycle management, and fail-closed recovery.

ARCHITECTURE | REQUEST PROCESSING | OPERATIONS

ANTIDESIGN SOFTWARE

HYDRA | PRODUCT DOCUMENTATION

System at a glance

Hydra is a session-aware Linux gateway that keeps a bounded pool of isolated IPVanish OpenVPN tunnels and selects a healthy path for each new client assignment.

2-20	1:1	256	30/30/60
supported active tunnels	active to replacement cohort	new-flow distribution buckets	default active / drain / hard stop

Core behavior

Every IPv4 source inside the connected LAN prefix receives the balanced VPN-only route once the desired pool is ready. A saved per-device policy is an optional override, not an enrollment requirement.

Two request planes

Plane	What arrives	What Hydra decides	Durable effect
Management	HTTPS status reads and self-service policy writes	Source-scoped access, validation, policy persistence, runtime reconciliation	Policy-only state in /var/lib/hydra/state.json
Data	DNS queries and new IPv4 flows	Eligible tunnel, affinity reuse, geography, health, capacity, and fail-closed outcome	Root-owned ownership journal plus nftables/conntrack runtime state

A client never supplies privileged network instructions. The root worker derives interfaces, pool limits, profiles, timing, and paths from the validated root-owned configuration.

Request path: from client to egress

Hydra separates intent, selection, and packet forwarding so a failed decision cannot silently fall through to the ordinary WAN route.

01	02	03	04	05
Observe	**Authorize**	**Select**	**Mark**	**Route**
Identify the request source and connected LAN prefix.	Apply subnet default or exact device override.	Choose only a healthy, non-draining eligible tunnel.	Record the path in DNS maps or the new flow's conntrack mark.	Policy-route into the matching isolated namespace.

1. Source and admission

- The connected IPv4 prefix on network.lanInterface is the admission boundary.
- Off-prefix sources and forwarded IPv6 are rejected by policy.
- An exact saved override is evaluated before the subnet default.

2. Eligibility

- Candidates must be healthy, accepting new work, inside capacity rules, and compatible with the client's country, city, or exact-server constraint.
- Below the configured minimum floor, Hydra withdraws new fallback assignments and remains fail closed while reconciliation continues.

3. Flow ownership

- The first packet of a new flow receives a Hydra mark. Conntrack restores that mark on later packets, keeping the established transport on its original path.
- Each active mark maps to one policy-routing table and one tunnel namespace. Source NAT happens inside that namespace.

No direct-WAN fallback

A required location with no healthy match receives a reserved non-routable sentinel mark. It cannot enter the active mark set, so that device's new traffic stops instead of escaping through a generic or direct route.

DNS processing and destination affinity

The optional Hydra DNS forwarder improves path consistency for applications that use several destinations while preserving the kill-switch boundary.

A	B	C	D	E
Receive	**Choose**	**Resolve**	**Publish**	**Reply**
Accept an authorized IPv4 client's DNS request.	Reuse valid affinity or select an eligible tunnel.	Send the upstream query with that tunnel's socket mark.	Write ownership and timed client-destination mappings.	Return the answer only after route publication succeeds.

Local management name

An on-prefix A query for the configured single-label name (default: hydra) is answered locally with Hydra's current LAN address. This answer bypasses upstream DNS and VPN selection, publishes no flow mapping, and remains available even when the client's Internet policy is waiting for a location. AAAA returns no data while IPv6 is blocked.

Affinity layers

Layer	Scope	Behavior
Flow	One established TCP/UDP flow	Mandatory. Conntrack keeps the original mark for the flow lifetime.
Destination	Related destinations for one client	Default sliding window, bounded by the absolute endpoint-assignment deadline.
Client sticky	All new work from one client	Strongest compatibility; selects one eligible tunnel for the lease.
Maximum rotation	Each new DNS assignment	Avoids the preceding tunnel when another eligible endpoint is available.

Timing rule

Activity may refresh short-term affinity only while the tunnel is healthy and accepting assignments, and never beyond the client's absolute maximum endpoint-assignment lifetime. Existing flows can continue during drain until they close or reach the system hard stop.

Policy and management request handling

The browser interface is source-scoped for end users and PAM-authenticated for the administrator.

End-user policy write

- Hydra derives the client identity from the observed request source. The unauthenticated form does not accept another client's address.
- Allowed controls are affinity mode, a bounded endpoint-assignment lifetime, and optional country, city, or exact catalog server.
- An exact server must exist in the imported profile catalog and match any selected country/city filters.
- The policy is atomically stored, then the privileged worker validates and reconciles it on the running pool without restarting tunnels.

Administrator settings write

- HTTPS and PAM authentication for the dedicated hydra account are mandatory.
- Only an approved subset of global pool, affinity, lifecycle, retention, and health settings is editable.
- The worker preserves unmanaged fields, validates the complete candidate, writes config.json.previous, atomically replaces config.json with ownership intact, and schedules a controlled restart.
- The dashboard tracks restart and health until pool and DNS readiness return. Invalid capacity or lifecycle relationships cause no write and no restart.

Authority	Owner / mode	Contents
/etc/hydra/config.json	root:hydra 0640	Global configuration and runtime mode
/var/lib/hydra/state.json	hydra:hydra 0640	Schema 2 per-device policies only
/var/lib/hydra-worker/ownership.json	root-owned private state	Privileged resources for deterministic recovery

Geographic demand without an unbounded pool

The selectable profile catalog is larger than the live tunnel cohort. Hydra satisfies a missing location by reusing existing replacement capacity.

01	02	03	04	05
Queue	**Resolve**	**Connect**	**Prove**	**Rebalance**
Keep only the requesting client fail closed.	Choose a matching validated catalog profile.	Bind it to a stopped replacement slot.	Require readiness and health before eligibility.	Drain a generic member only after the match is safe.

Safety properties

- Concurrent namespaces remain bounded to the active cohort plus the equal replacement cohort.
- A tunnel that is the sole healthy match for another constrained client is protected.
- Clients with the same requirement share one in-flight demand instead of starting duplicate work.
- A failed endpoint is briefly excluded so a retry can advance to another matching profile.
- Unrelated clients and established conntrack sessions continue while the requested location is prepared.

Visible status

The source-scoped runtime endpoint reports queued, resolving, connecting, ready, and retry states, including the selected profile, replacement slot, and number of eligible tunnels. The client dashboard polls this endpoint and never labels an unavailable route as active.

> **Catalog versus live pool**
>
> An empty seed list enables deterministic physical U.S.-first seeding across different cities. An explicit seed list remains an exact operator override and must contain exactly twice the desired active count.

Lifecycle, failure, and recovery

A tunnel is removed from new selection before it is retired. Planned events preserve established flows; hard failures preserve the fail-closed boundary.

State	New assignments	Existing flows	Transition rule
Starting	No	None	OpenVPN, routes, namespace, and health must become ready.
Healthy	Yes	Yes	Eligible subject to client policy and pool floor.
Draining	No	Yes	Replacement is healthy first; affinity stops refreshing.
Stopped	No	No	Resources are removed in dependency order.
Failed	No	May break	Withdraw immediately, clean ownership, and retry capacity.

Default lifecycle

0-30	30-60	60
minutes active	minutes draining	minute hard stop

Crash and reboot recovery

- The privileged worker journals ownership before network mutation.
- On startup, it validates the fail-closed firewall, removes only deterministic recorded resources, restores the original IPv4 forwarding value, clears the journal durably, and then opens its Unix socket.
- The unprivileged service remains unhealthy and retries until worker ownership, pool capacity, and DNS readiness are restored.
- Provider endpoint lookup uses Hydra's configured upstream DNS directly; provider-pushed DNS settings cannot rewrite the host resolver.

Expected interruption

Planned drain can preserve established sessions. A hard tunnel failure can break sessions using that tunnel, and a worker crash or host reboot can interrupt existing sessions. In every case, new traffic remains fail closed until Hydra reconstructs a healthy route.

Security and privilege boundaries

Hydra's control plane runs unprivileged. The narrow root worker owns only the network operations and files required to enforce routing and recovery.

Boundary	Protection
Worker socket	Root-owned peer-credential-authenticated Unix socket; only the hydra service identity is authorized.
Systemd confinement	NoNewPrivileges and a bounded capability set for network namespace and packet-policy work.
Firewall	Atomic nftables replacement after syntax check, signed policy verification, final fail-closed drop, and explicit IPv6 blocking.
Namespace isolation	Each OpenVPN process, TUN interface, routes, and source NAT live in a dedicated network namespace.
Secrets	VPN credentials are root-owned and are not placed in process arguments or logs.
State input	Regular-file, stable-identity, size, permission, schema, and value validation before worker use.
Management access	LAN-prefix and destination-scoped UFW rules; authenticated global settings; source-scoped self-service.

Current product boundaries

- IPv4 forwarding is supported. Forwarded IPv6 is blocked rather than tunneled.
- Hydra is not a DHCP server. Existing DHCP or manual client configuration must advertise Hydra as gateway and DNS.
- The initial provider integration is IPVanish OpenVPN.
- Direct non-VPN routing remains an administrator-only exception.
- Policies are keyed by IPv4 address; DHCP reservations are recommended for durable per-device overrides.

Operating profile

Hydra combines bounded capacity, explicit lifecycle controls, source-scoped management, and a fail-closed routing contract.

Designed behavior

- Isolated tunnel establishment, concurrent operation, group failure containment, and deterministic cleanup.
- Replacement-before-drain rotation, individual member recovery, privileged-worker reconstruction, and reboot recovery.
- Independent client DNS, flow, fallback, and fail-closed policy state with catalog-backed geography and absolute assignment limits.
- Bounded operation from 2 through 20 active tunnels with an equal replacement cohort.

> **Product contract**
>
> Hydra keeps the control plane unprivileged, delegates narrow network mutations to its root worker, and withdraws new assignments whenever the requested route cannot be safely satisfied.

Configuration defaults

Area	Default / contract
Pool	desired 12, minimum 10 in config.example.json; supported desired range 2-20
Timing	30 min affinity, 30 min active, 30 min drain, 60 min hard stop, 15 sec health cadence
Policy retention	24 hours of inactivity by default
Management	HTTPS on port 8443; canonical LAN name hydra
Runtime	Fresh and upgraded configurations remain control-plane until root explicitly enables gateway mode