Operate the connector host lifecycle

Use this guide to move one standalone connector-host replica through startup, readiness, normal service, graceful drain, offline retention, and recovery. It applies to the common host used by Cal.diy, Hermes Agent, Chatwoot, Dify, CrewAI, a

SECTION	Deploy and Operate
TYPE	Guide
PROTOCOL	AIP 1.0
SOURCE	docs/guides/connector-host-lifecycle.md

The common bootstrap owns the lifecycle at source revision `97be86e9efedf07ecf1783b03800f683f107fb04`. Product hosts supply connector configuration and authenticated provider ingress, but they all use the same durable recovery, signed control-plane, lease, drain, and shutdown sequence.

The desired result is not merely a running process. It is a replica whose runtime state recovered cleanly, admitted identity matches the registry, health lease is current, product and storage probes are ready, and shutdown can retain the evidence needed for pinned work.

Lifecycle model

The registry exposes three replica states: `ready`, `draining`, and `offline`. Preparation and recovery happen before registration and are not registry states.

```
MERMAID
flowchart LR
    P["Pre-provisioned offline replica"] --> B["Prepare, build, and recover"]
    B -->|"signed register"| R["Ready with bounded lease"]
    R -->|"healthy heartbeat"| R
    R -->|"SIGTERM or Ctrl-C"| D["Draining"]
    D -->|"server stops"| O["Offline"]
    R -->|"health failure or lease expiry"| O
    O -->|"healthy recovery and signed register"| R
```

In text: a pre-provisioned offline replica prepares durable state and registers as ready. Healthy heartbeats renew its lease. A planned signal moves it to draining and then offline. Failed health or an expired lease removes it from new routing. A clean offline or lease-expired replica can register again after durable recovery.

`draining` preserves pinned work while preventing new route assignments. `offline` makes the replica unavailable. Neither state deletes durable action, receipt, approval, transaction, callback, or assignment records.

Keep lifecycle authority narrow

Run the standalone `aip-connector-control-plane` with its restricted lifecycle database role. The service exposes only signed `register`, `heartbeat`, `drain`, and `offline` operations at `/aip/v1/connector-control`. It must not retain connector catalog administrator credentials.

Before admitting hosts, verify the lifecycle service:

- reports process health at `/health`;
- reports the expected registry schema, catalog revision, and data-pool state

at `/ready`;

- exposes only approved metrics through the observability boundary;
- signs responses with the DID pinned by every host;
- uses TLS outside explicitly enabled loopback development;
- has no control/admin database pool in its process snapshot.

Each host signs lifecycle requests with its own pre-provisioned replica key. The service compares that DID with the registry record before applying a command. Requests and responses have a 30-second lifetime, allow at most five seconds of future clock skew, and are limited to 64 KiB. Request IDs, payload digests, durable replay claims, and monotonic health revisions make retries idempotent and sequence-fenced.

Do not script unsigned writes to replica status columns. That bypasses peer identity, replay, lease-sequence, and immutable configuration checks.

Prerequisites for one replica

Require all of the following before starting the process:

- an active admitted connector version with valid artifact evidence;
- an enabled tenant-owned connector instance;
- a pre-provisioned replica with the intended endpoint, peer principal, host

DID, trust domain, native transport profile, topology, and capacity;

- the exact manifest and artifact digests represented by the version;
- instance configuration revision and secret-provider reference;
- owner-controlled host signing, database, provider, and optional DID files;
- a durable connector runtime PostgreSQL backend;
- network trust to the fixed lifecycle endpoint and the pinned lifecycle DID;
- a process supervisor that sends SIGTERM and allows a sufficient termination

grace period;

- metrics and registry visibility before traffic is enabled.

The version, instance, and replica are different identities. Do not reuse a replica ID for another endpoint, key, instance, version, topology, or principal. Create and admit a new replica instead.

Start the host

Use the product host binary and common fields documented in Connector host configuration. The common lifecycle performs these steps in order:

1. validate common and product configuration and bounded file metadata;
2. open the durable runtime database and install its immutable schema;
3. perform tenant-scoped connector discovery;
4. bind the manifest to the deployment trust domain and signing principal;
5. build the native gateway and authenticated product-owned routes;
6. recover durable runtime work before registration;
7. probe durable storage and product connector health;
8. send a signed registration to the lifecycle service;

9. apply the returned lease and start heartbeat renewal;
10. begin serving health, readiness, metrics, manifest, native AIP, and any

product-owned ingress routes.

Recovery reports pending approvals, recoverable transactions, completed queued actions, completed delegations, and resumed callback deliveries. If queued or delegation recovery contains errors, registration is blocked. Correct or reconcile the durable records; do not mark the replica ready manually.

Registration succeeds only when the running process matches the registry:

Running claim	Registry owner
Connector version and type	Admitted version and instance
Manifest and artifact digests	Immutable version and attestation
Tenant, configuration revision, and secret reference	Enabled instance
Endpoint, peer principal, DID, trust domain, topology, and capacity	Pre-provisioned replica
Capability count	Admitted manifest

The initial proposal must report `ready`, zero active assignments, health revision zero, and no prior lifecycle request. The control plane assigns the committed health revision and lease expiry.

Verify readiness before traffic

After registration, check three different views:

1. host `/health` confirms that the process and AIP 1.0 surface respond;
2. host `/ready` confirms a valid lease, no drain, readable and writable

durable storage, and a ready product connector;

3. registry and tenant-scoped discovery confirm a ready eligible replica under

the intended enabled binding.

Do not substitute one view for another. `/health` does not inspect the lease, storage, or connector. Core `aipd` readiness does not require one particular connector replica or binding. Registry readiness does not prove every provider scope or mutation.

Enable traffic only after the three views agree and a reviewed non-mutating check reaches the intended provider boundary.

Observe heartbeats and leases

The default host lease lasts 30 seconds and the default heartbeat interval is 10 seconds. Configuration enforces a heartbeat interval no greater than half the lease lifetime. The first heartbeat is jittered around the configured interval, and missed timer ticks are skipped rather than replayed in a burst.

Configure the host and lifecycle service with compatible lease expectations; the expiry returned by the trusted lifecycle service is authoritative.

Each heartbeat carries:

- replica ID;
- previous lease sequence;
- current local in-flight action count;
- combined connector and storage readiness.

The lifecycle service rejects an in-flight count above admitted replica capacity and rejects a stale previous sequence. A healthy heartbeat moves or keeps the replica in `ready` and advances its health revision. An unhealthy heartbeat moves it to `offline` and expires its lease, preventing new routing.

Monitor the common host metrics together:

Metric	Interpretation
`aip_connector_host_ready`	Combined local lease, drain, storage, and connector state
`aip_connector_host_storage_ready`	Durable runtime probe result
`aip_connector_host_in_flight`	Actions currently executing in this process
`aip_connector_host_requests_total`	Native requests observed by the host
`aip_connector_host_rejected_total`	Requests refused before successful handling
`aip_connector_host_heartbeat_failures_total`	Failed renewal or recovery attempts
`aip_connector_host_lease_recovery_attempts_total`	Expired-lease registration attempts
`aip_connector_host_lease_recoveries_total`	Successful expired-lease recoveries

A single heartbeat failure is not lease loss. Alert on the relationship between failure growth, last valid lease expiry, `/ready`, fleet-ready replicas, and available capacity.

Drain a replica cleanly

For planned maintenance, ensure another ready replica can take new assignments or pause the affected tenant capability. Then send SIGTERM through the process supervisor. Ctrl-C follows the same common path for an interactive process.

The host performs shutdown in this order:

1. set its local draining flag before contacting the lifecycle service;
2. request a signed `draining` transition with the current lease sequence;
3. stop heartbeat renewals while draining;
4. reject new `Action` messages locally and stop provider-event publication;
5. keep the listener available while the local in-flight action count falls;
6. after zero in-flight actions or the configured drain interval, request Axum

graceful server shutdown;

7. stop the heartbeat worker;
8. send the signed `offline` transition;
9. return success only when server, heartbeat, drain, and offline results are

clean.

The registry `draining` state prevents new assignments but retains existing routes. A local drain remains active even if the remote drain request fails. New actions receive a retryable `connector_host.not_ready` response while the process remains reachable.

`drain-timeout-ms` bounds the initial wait for the host's in-flight counter. It does not promise to cancel provider work, terminate every open HTTP connection, or make an external mutation reversible. Configure the workload supervisor's termination grace to cover the drain interval, lifecycle calls, server graceful shutdown, and an evidence-based margin. A forced kill changes this to the crash-recovery path.

Verify all of the following before removing the old process or volume:

- registry state is `offline` and its lease is expired;
- the host process exited without a lifecycle aggregate error;
- no assignment remains unexplained;
- durable recovery and provider audit records are retained;
- replacement fleet capacity and tenant discovery remain correct.

Replace replicas without identity reuse

For a rolling deployment, pre-provision and start a new replica with a new ID, endpoint, and signing key. Require its readiness and registry identity before signalling the old replica. Drain one failure domain at a time and keep enough capacity for both active work and retry pressure.

Do not change the endpoint or key behind an existing live replica record. A registration using an existing replica ID is accepted only after that record is cleanly offline with zero active assignments or after its lease expires. Immutable identity and endpoint mismatches remain conflicts after either event.

Credential changes additionally require the binding and secret-material procedure in Rotate connector credentials.

Recover from a control-plane outage

When heartbeat delivery fails, the host retains its last committed lease until the recorded expiry. It increments the heartbeat-failure counter. After expiry, `/ready` returns unavailable and new actions are rejected even if the process and provider health remain good.

If local storage and connector health are ready, the heartbeat worker switches from renewal to signed re-registration. For an ambiguous transport or control-plane response, it reuses the exact request ID and registration bytes; the durable replay guard turns a lost response into an idempotent retry. A successful response applies the new sequence and restores readiness.

An admission or configuration rejection clears the pending request identity so a later attempt is not confused with a consumed replay. It does not repair the underlying mismatch. Compare version, instance, replica, digest, tenant, configuration revision, secret reference, capability count, DID, and clock before restarting anything.

Do not extend a lease in the database, change a replica to `ready`, or weaken the pinned lifecycle DID during an outage.

Recover after a hard crash

If the process cannot send drain and offline transitions:

1. stop traffic to the failed endpoint without deleting action records;
2. keep a distinct healthy replica serving new work where available;
3. wait for the failed replica lease to expire before reusing its identity;
4. preserve the same durable runtime database and exact immutable replica

identity for recovery;

5. start the process and let durable recovery run before registration;
6. require the recovered capacity to be at least the registry's retained active

assignment count;

7. verify the new lease sequence, readiness, and recovery summary;
8. resume existing actions with their original action IDs and pinned routes;

9. reconcile any provider effect whose completion is not proved.

Registration before clean offline or lease expiry is rejected as a competing takeover. Once eligible, the control plane advances the health revision and preserves registry-owned active assignments for the same replica. The product-neutral daemon also expires stale replicas in bounded maintenance batches and releases their active admission reservations; it does not authorize creating replacement actions for ambiguous work.

Never convert an uncertain old action into a fresh action ID merely to obtain a new replica. The original ID, route assignment, idempotency key, receipts, and provider audit record are the evidence boundary.

Respond to health regression

When connector or durable storage health fails:

- confirm host `/ready` is unavailable;
- confirm the replica becomes ineligible for new routing;
- preserve another ready replica or pause the binding;
- identify whether the failure is product, credential, network, storage, or

lifecycle related;

- repair the owning boundary without changing admitted identity;
- allow automatic re-registration after both local probes recover;
- verify a non-mutating action before restoring full traffic.

Do not repeatedly restart a host whose durable recovery fails. Recovery occurs before registration specifically to prevent a damaged queue or delegation state from being advertised as ready.

Escalation matrix

Observation	Likely boundary	Safe next action
Process exits before routes open	Bootstrap, storage, discovery, recovery, or registration	Read the typed startup error and reconcile the named boundary.
`/health` succeeds and `/ready` fails	Lease, drain, storage, or connector health	Inspect the structured readiness fields and corresponding metrics.
Heartbeat failures rise but lease remains valid	Lifecycle network, TLS, clock, or service	Restore the signed path before expiry; do not edit registry state.
Lease recovery attempts rise without successes	Persistent admission mismatch or control outage	Compare exact identity and control-plane response category.
Drain begins but process does not exit	In-flight or open connection remains	Preserve evidence, extend supervisor grace if safe, and avoid a blind kill.
Same replica cannot register	Live lease, active clean-offline condition, capacity, or immutable mismatch	Wait for eligibility or create a distinct pre-provisioned replica.
Replica is offline after provider health failure	Fail-closed heartbeat transition	Repair provider health and let the same process re-register.

For the complete initial deployment, use Deploy the connector fleet. Keep lifecycle choices inside the production boundaries in Deploy AIP in production, and use Profiles and connectors to distinguish the host process from its connector semantics and transport profiles.