Connector host configuration

This reference lists the common process configuration shared by every standalone connector host. Use it to configure the host identity, admitted artifact, tenant boundary, durable storage, central services, delivery routes, and lifecycle li

SECTION	Deploy and Operate
TYPE	Reference
PROTOCOL	AIP 1.0
SOURCE	docs/reference/connector-host-configuration.md

The contract is implemented by `ConnectorHostBootstrapArgs` at source revision `97be86e9efedf07ecf1783b03800f683f107fb04`. The Cal.diy, Hermes Agent, Chatwoot, Dify, CrewAI, and Twenty host binaries all flatten this structure into their command-line interface. Each binary adds its own provider settings after these common options.

All listed behaviors are implementation contracts for the reviewed revision. They are not new AIP protocol requirements, and the defaults are not capacity recommendations for an arbitrary production deployment.

Value sources

Every common option can be supplied as a command-line flag or through its listed environment variable. Do not configure conflicting values in both places. Values shown as **Required** have no implementation default; the host cannot prepare without them or a required alternative.

The host does not accept database credentials or signing material as literal command-line values. It reads those secrets from bounded, owner-controlled files. Provider credentials use opaque handles and product-owned secret files; they are never embedded in this common configuration.

Listener and central endpoints

CLI option	Environment variable	Default	Meaning and constraint
`--bind`	`AIP_CONNECTOR_HOST_BIND`	`0.0.0.0:8081`	Socket used by the host listener behind deployment ingress.
`--public-endpoint`	`AIP_CONNECTOR_HOST_PUBLIC_ENDPOINT`	**Required**	Advertised native endpoint. It must be absolute and end exactly in `/aip/v1/messages`.
`--control-plane-endpoint`	`AIP_CONNECTOR_HOST_CONTROL_ENDPOINT`	**Required**	Lifecycle service endpoint. It must be absolute and end exactly in `/aip/v1/connector-control`.
`--callback-endpoint`	`AIP_CONNECTOR_HOST_CALLBACK_ENDPOINT`	Unset	Central stream callback endpoint. If set, its path must be exactly `/aip/v1/connector-callbacks`.
`--event-endpoint`	`AIP_CONNECTOR_HOST_EVENT_ENDPOINT`	Unset	Central connector-event ingress. If set, its path must be exactly `/aip/v1/connector-events`.

Each URL requires a host and rejects embedded credentials, a query, and a fragment. The public and control endpoints require HTTPS. Plain HTTP is accepted only when `--allow-insecure-loopback-http` is enabled and both the listener and the corresponding URL are loopback-scoped.

Callback and event delivery have separate network exceptions. Their destination host is pinned into the outbound allowlist during preparation. The common host uses its native signing identity for both delivery paths; a different callback or event signer is rejected.

Admitted connector identity

These values identify the exact registry objects that the process claims when it registers. They must match the admitted records; they do not create those records.

CLI option	Environment variable	Default	Meaning and constraint
`--connector-type-id`	`AIP_CONNECTOR_HOST_TYPE_ID`	**Required**	Stable connector implementation family.
`--version-id`	`AIP_CONNECTOR_HOST_VERSION_ID`	**Required**	Immutable admitted connector version.
`--instance-id`	`AIP_CONNECTOR_HOST_INSTANCE_ID`	**Required**	Tenant-owned logical provider boundary.
`--replica-id`	`AIP_CONNECTOR_HOST_REPLICA_ID`	**Required**	Pre-provisioned concrete process replica.
`--artifact-digest`	`AIP_CONNECTOR_HOST_ARTIFACT_DIGEST`	**Required**	Exact `sha256:` digest of the running host artifact.
`--config-revision`	`AIP_CONNECTOR_HOST_CONFIG_REVISION`	`1`	Monotonic non-secret instance configuration revision; it must be greater than zero.
`--secret-provider-ref`	`AIP_CONNECTOR_HOST_SECRET_PROVIDER_REF`	**Required**	Non-secret opaque secret-provider reference containing 1–2,048 bytes.

`--artifact-digest` accepts the canonical SHA-256 form: the `sha256:` prefix followed by 64 hexadecimal characters. A changed artifact, admitted version, instance identity, or configuration revision must be reconciled through the registry workflow rather than silently changed at process restart.

Tenant and external account boundary

CLI option	Environment variable	Default	Meaning and constraint
`--tenant-id`	`AIP_CONNECTOR_HOST_TENANT_ID`	**Required**	Verified tenant owned by this instance; 1–512 bytes.
`--tenant-system`	`AIP_CONNECTOR_HOST_TENANT_SYSTEM`	Unset	Optional source system carried by the tenant reference.
`--membership-id`	`AIP_CONNECTOR_HOST_MEMBERSHIP_ID`	**Required**	Stable membership assertion for the instance; 1–512 bytes.
`--external-account-id`	`AIP_CONNECTOR_HOST_EXTERNAL_ACCOUNT_ID`	Unset	Stable provider account, workspace, application, or endpoint identifier; 1–512 bytes.
`--external-account-system`	`AIP_CONNECTOR_HOST_EXTERNAL_ACCOUNT_SYSTEM`	Unset	Provider system paired with the external account; 1–128 bytes.

The external account ID and system are one optional pair: configure both or neither. The host performs connector discovery with the configured tenant before it opens the listener. The resulting manifest is then bound to the deployment trust domain and host signing identity.

Gateway and lifecycle trust

CLI option	Environment variable	Default	Meaning and constraint
`--gateway-principal-id`	`AIP_CONNECTOR_HOST_GATEWAY_PRINCIPAL_ID`	**Required**	Service principal trusted to submit signed native requests from the central gateway.
`--gateway-did`	`AIP_CONNECTOR_HOST_GATEWAY_DID`	Alternative required	Inline `did:key` used to verify gateway requests.
`--gateway-did-file`	`AIP_CONNECTOR_HOST_GATEWAY_DID_FILE`	Alternative required	Owner-controlled file containing the gateway `did:key`.
`--control-plane-did`	`AIP_CONNECTOR_HOST_CONTROL_DID`	Alternative required	Inline `did:key` used to verify lifecycle responses.
`--control-plane-did-file`	`AIP_CONNECTOR_HOST_CONTROL_DID_FILE`	Alternative required	Owner-controlled file containing the lifecycle `did:key`.
`--signing-seed-file`	`AIP_CONNECTOR_HOST_SIGNING_SEED_FILE`	**Required**	Owner-controlled file containing the host Ed25519 seed as 64 hexadecimal characters.
`--trust-domain`	`AIP_CONNECTOR_HOST_TRUST_DOMAIN`	**Required**	Shared gateway and host trust domain; 1–253 bytes.
`--tls-ca-certificate-file`	`AIP_CONNECTOR_HOST_TLS_CA_FILE`	Unset	Additional PEM root for private-PKI outbound TLS; maximum 1 MiB.

Configure exactly one gateway DID source and exactly one control-plane DID source. Every DID value must parse as a `did:key` and contain no more than 512 bytes. The host derives its own `did:key` from the configured signing seed; it does not accept the host DID as an independent setting.

The optional CA augments the public root set. Certificate and hostname verification remain enabled. The host supplies the root to lifecycle, callback, and event delivery clients; it is not a switch for disabling TLS authentication.

Durable storage

CLI option	Environment variable	Default	Meaning and constraint
`--database-url-file`	`AIP_CONNECTOR_HOST_DATABASE_URL_FILE`	**Required**	Owner-controlled UTF-8 file containing the connector runtime PostgreSQL URL; maximum 16 KiB.

Preparation reads the database URL, connects to PostgreSQL, and installs the durable runtime schema before the process can serve. The common host constructs the action, event, idempotency, profile, session, approval, transaction, and execution stores from that backend. It does not fall back to ephemeral runtime state.

Product hosts may bind their own durable projections to the same backend. The schema and database privileges required by a product remain part of its connector-local configuration and deployment documentation.

Topology labels

CLI option	Environment variable	Default	Meaning and constraint
`--region`	`AIP_CONNECTOR_HOST_REGION`	`global`	Stable deployment or geographic region used for new route selection.
`--zone`	`AIP_CONNECTOR_HOST_ZONE`	`default`	Independent failure zone within the region.
`--capacity-class`	`AIP_CONNECTOR_HOST_CAPACITY_CLASS`	`standard`	Operator-defined class such as `standard` or `gpu`.

The registry validates each label as at most 128 ASCII letters, digits, dots, dashes, or underscores. Labels are indexed routing coordinates, not high-cardinality metric dimensions. They influence creation of new route assignments; an existing durable assignment remains pinned to its exact replica.

Opaque credential handle

The common host can bind a provider credential without receiving its secret value. The connector resolves the opaque handle inside its product boundary.

CLI option	Environment variable	Default	Meaning and constraint
`--credential-id`	`AIP_CONNECTOR_HOST_CREDENTIAL_ID`	Unset	Opaque provider credential handle.
`--credential-issuer`	`AIP_CONNECTOR_HOST_CREDENTIAL_ISSUER`	Unset	Issuer for the handle.
`--credential-scope`	`AIP_CONNECTOR_HOST_CREDENTIAL_SCOPES`	Unset	Verified scope; repeat the flag or use a comma-delimited environment value.
`--credential-revision-ref`	`AIP_CONNECTOR_HOST_CREDENTIAL_REVISION`	Unset	Opaque revision pinned by route assignments; 1–256 bytes.
`--credential-revision-policy-file`	`AIP_CONNECTOR_HOST_CREDENTIAL_POLICY_FILE`	Unset	Reloadable non-secret JSON authorization policy; maximum 64 KiB.

Credential ID, issuer, revision, and at least one non-empty scope form one tuple. Configure the complete tuple or omit all four. The common host fixes the credential tenant to `--tenant-id`; a mismatched tenant is rejected.

Without a policy file, the process creates a static policy that authorizes only the configured current revision. With a policy file, the host validates it at startup and reloads it immediately before each provider side effect. An unreadable, invalid, or non-authorizing file denies the action.

The file has this closed schema:

```
JSON
{
  "current_revision_ref": "credential-revision-current",
  "accepted_previous_revisions": ["credential-revision-previous"],
  "revoked_revisions": []
}
```

Unknown fields are rejected. Each revision reference contains 1–256 bytes, and the current, previous, and revoked sets cannot overlap. The complete replacement and emergency-revocation sequence belongs to Rotate connector credentials.

Callback and event network policy

CLI option	Environment variable	Default	Meaning and constraint
`--callback-allow-http`	`AIP_CONNECTOR_HOST_CALLBACK_ALLOW_HTTP`	`false`	Permit HTTP for the configured callback destination.
`--callback-allow-private-networks`	`AIP_CONNECTOR_HOST_CALLBACK_ALLOW_PRIVATE`	`false`	Permit a private callback destination.
`--event-allow-http`	`AIP_CONNECTOR_HOST_EVENT_ALLOW_HTTP`	`false`	Permit HTTP for the configured event-ingress destination.
`--event-allow-private-networks`	`AIP_CONNECTOR_HOST_EVENT_ALLOW_PRIVATE`	`false`	Permit a private event-ingress destination.

A callback exception is invalid without `--callback-endpoint`; an event exception is invalid without `--event-endpoint`. These switches weaken only the selected fixed outbound destination policy. They do not relax native ingress, lifecycle endpoint, DID, signature, or path validation.

Provider event publication also depends on host readiness. The built-in publisher accepts work only while the replica lease is valid, the host is not draining, durable storage is ready, and the product connector reports ready.

Lifecycle and resource limits

CLI option	Environment variable	Default	Accepted value
`--allow-insecure-loopback-http`	`AIP_CONNECTOR_HOST_ALLOW_LOOPBACK_HTTP`	`false`	Boolean; only valid with a loopback listener.
`--max-in-flight`	`AIP_CONNECTOR_HOST_MAX_IN_FLIGHT`	`32`	Integer greater than zero.
`--lease-ttl-ms`	`AIP_CONNECTOR_HOST_LEASE_TTL_MS`	`30000`	1,000–300,000 ms.
`--heartbeat-interval-ms`	`AIP_CONNECTOR_HOST_HEARTBEAT_MS`	`10000`	Greater than zero and no more than half the lease TTL.
`--drain-timeout-ms`	`AIP_CONNECTOR_HOST_DRAIN_TIMEOUT_MS`	`30000`	1–300,000 ms.
`--health-probe-timeout-ms`	`AIP_CONNECTOR_HOST_HEALTH_TIMEOUT_MS`	`2000`	1–30,000 ms.
`--control-timeout-ms`	`AIP_CONNECTOR_HOST_CONTROL_TIMEOUT_MS`	`5000`	1–30,000 ms.

`max-in-flight` is the replica-local action bound. Fleet admission policies can impose additional global, tenant, connector-type, instance, and binding limits. Increasing the local value cannot bypass those controls.

The reviewed implementation also fixes three non-configurable host limits:

Limit	Value
Maximum canonical manifest size	4 MiB
Maximum capabilities in one immutable connector version	512
Maximum native AIP request body	4 MiB

File requirements

Every common file input must be a non-empty regular file, not a symbolic link, and must remain the same file while it is opened and read. The implementation enforces the configured byte limit before and during the read.

On Unix systems, secret inputs must have no group or world permission bits. A mode such as `0600` satisfies that check. This owner-only rule applies to the database URL, signing seed, and current DID-file inputs.

Secret text must be valid UTF-8 after surrounding ASCII whitespace is removed, except for the seed, which is decoded as hexadecimal. Secret buffers are zeroized where the bootstrap API retains them temporarily.

The TLS CA and credential revision policy are non-secret configuration files. They may be group/world readable, but they must not be group/world writable; a mode such as `0644` satisfies that check. Replace a reloadable policy atomically so a reader never observes a partial JSON document.

Coupling rules

The following relationships are validated before or during host construction:

- loopback HTTP requires a loopback listener and loopback public and control

URL hosts;

- gateway DID and gateway DID file are mutually exclusive alternatives;
- control-plane DID and control-plane DID file are mutually exclusive

alternatives;

- external account ID and external account system are present together;

- credential ID, issuer, revision, and non-empty scopes are present together;
- the credential tenant equals the instance tenant;
- callback network exceptions require a callback endpoint;
- event network exceptions require an event endpoint;
- the signing principal and central gateway share the configured trust domain;
- the heartbeat interval is no more than half the lease TTL;
- artifact digest and configuration revision match admitted registry state.

A coupling error is a configuration failure. Do not work around it by dropping identity or transport validation; reconcile the deployment inputs with the admitted records.

Preparation, serving, and readiness

The common bootstrap has two phases. `prepare` validates configuration and file metadata, loads identities, performs tenant-scoped connector discovery, opens durable PostgreSQL storage, and constructs the signed lifecycle client. It does not register the replica.

The serving phase builds the host, recovers durable work, registers the exact replica, renews its lease, exposes the common routes, drains on SIGTERM or Ctrl-C, and marks the replica offline. Product binaries may add authenticated provider ingress routes without changing the common native surface.

Route	Readiness meaning
`/health`	The process and protocol surface are responsive.
`/ready`	Lease, drain state, durable storage, and product connector health allow new work.
`/metrics`	Common host metrics are available.
`/aip/v1/manifest`	The deployment-bound immutable manifest is available.
`/aip/v1/messages`	Signed native AIP message ingress is available.

A successful process health response does not prove route eligibility. Use `/ready`, registry status, and tenant-scoped discovery together before enabling a capability binding.

Error categories

Startup and serving errors identify the boundary that rejected the host:

Category	Typical cause	Operator response
Bootstrap configuration	Missing coupled fields, unsafe file metadata, invalid URL, DID, digest, or limit	Correct the input; do not start traffic.
Durable storage	PostgreSQL connection or schema installation failure	Restore database reachability and least-privilege schema access.
Product discovery	Connector cannot publish its tenant-scoped manifest	Correct only the product-owned configuration, then repeat preparation.
Host admission or control plane	Registry identity, trust, lease, or lifecycle response fails	Compare the process identity with admitted records and service trust.
Listener	Socket cannot bind	Correct the address, port ownership, or deployment ingress mapping.

The host fails closed on these errors. A restart with unchanged invalid inputs does not repair an admission, trust, credential policy, or file-permission failure.

Product-specific configuration

This page intentionally ends at the common host boundary. Provider base URLs, API keys, OAuth material, webhook secrets, account identifiers, feature flags, and product-specific health behavior belong to the six connector-local configuration pages. Do not infer one connector's provider settings from another connector or add an unreviewed connector to the public matrix.

For the complete deployment sequence, use Deploy the connector fleet. Review the system boundary in Deploy AIP in production, the signing model in Identity and trust, the host implementation contract in Build a connector, and the integration model in Profiles and connectors.