Cal.diy connector configuration

This reference describes the configuration accepted by aip-host-cal-diy at the reviewed source revision. A complete process combines eight Cal.diy-specific settings with the shared connector-host bootstrap contract.

SECTION	Connectors
TYPE	Connector
PROTOCOL	AIP 1.0
SOURCE	docs/connectors/cal-diy/reference/configuration.md

All settings are read at startup except the optional credential-revision policy file, which is re-read before provider side effects. Provider credentials, webhook secrets, destination prefixes, and other startup settings are not documented as hot-reloadable.

Configuration identity

Property	Value
Binary	`aip-host-cal-diy`
AIP workspace version	`1.0.0`
Reviewed source	`97be86e9efedf07ecf1783b03800f683f107fb04`
Connector id	`cal-diy`
Connector profile	`aip.connector.cal_diy.v1`
Pinned upstream source	`f00434927386c9ecdcbd7e6c5f82d22044a245bc`
Provider API family	Cal.diy API v2

Each field is available through the listed environment variable. Cal.diy fields also expose the listed command-line flag. Shared fields use the equivalent kebab-case connector-host flag. Use one source of truth per setting so an orchestrator and a manual override cannot silently disagree.

Secret values belong in files. Environment variables and command-line flags name those files or carry non-secret identity and policy values.

Cal.diy-specific settings

Environment variable	Command-line flag	Required	Default	Meaning and validation
`AIPD_CAL_DIY_BASE_URL`	`--cal-diy-base-url`	Yes	None	Fixed Cal.diy origin; HTTPS required except loopback development HTTP; credentials in the URL are rejected
`AIPD_CAL_DIY_ACCOUNT_ID`	`--cal-diy-account-id`	Yes	None	Stable state and provider-account boundary; 1–128 safe ASCII characters
`AIPD_CAL_DIY_BEARER_TOKEN_FILE`	`--cal-diy-bearer-token-file`	Conditional	None	Owner-only file for Bearer, API-key, managed-user, or OAuth access-token material
`AIPD_CAL_DIY_OAUTH_CLIENT_ID`	`--cal-diy-oauth-client-id`	Conditional	None	Non-empty public Cal platform client id; requires the client-secret file
`AIPD_CAL_DIY_OAUTH_CLIENT_SECRET_FILE`	`--cal-diy-oauth-client-secret-file`	Conditional	None	Owner-only Cal platform client-secret file; requires the client id
`AIPD_CAL_DIY_MAX_RESPONSE_BYTES`	`--cal-diy-max-response-bytes`	No	`8388608`	Maximum provider body read into memory; integer from 1 through `67108864`
`AIPD_CAL_DIY_WEBHOOK_SECRET_FILES`	`--cal-diy-webhook-secret-file`	No	Empty	Comma-delimited or repeatable `SUBSCRIPTION_ID=SECRET_FILE` mappings
`AIPD_CAL_DIY_WEBHOOK_SUBSCRIBER_PREFIXES`	`--cal-diy-webhook-subscriber-prefix`	No	Empty	Comma-delimited or repeatable deployment-owned HTTPS destination prefixes

Authentication groups

Exactly one provider authentication group must be complete:

```
TEXT
Bearer = AIPD_CAL_DIY_BEARER_TOKEN_FILE
```

or:

```
TEXT
Cal platform client = AIPD_CAL_DIY_OAUTH_CLIENT_ID
                    + AIPD_CAL_DIY_OAUTH_CLIENT_SECRET_FILE
```

The Bearer setting conflicts with both client settings. A partial client pair also fails startup. The provider client sends either `Authorization: Bearer …` or the `x-cal-client-id` and `x-cal-secret-key` headers. It performs no browser OAuth flow, token exchange, or refresh.

Base URL

The base URL parser requires:

- an `https` URL with an authority; or
- `http` only when the host is a loopback name or address;
- no username or password;
- no action-selected origin.

The parser removes query and fragment components, normalizes the trailing slash, and refuses redirects. The provider request timeout is fixed at 30 seconds.

Account id

The account id must contain 1 through 128 ASCII letters, digits, dots, underscores, or hyphens. It scopes the durable Cal.diy mutation ledger together with the normalized base URL.

For a standalone deployment, make it equal to the shared `AIP_CONNECTOR_HOST_EXTERNAL_ACCOUNT_ID` by deployment policy. The reviewed code validates both fields but does not promise to compare them automatically.

Webhook settings

Webhook ingestion is disabled when no secret mappings are configured. When one or more mappings are present, the host adds this route:

```
TEXT
POST /webhooks/cal-diy/{subscription_id}
```

It also requires `AIP_CONNECTOR_HOST_EVENT_ENDPOINT`; startup fails if the host cannot construct its central event publisher.

Secret mappings

Each mapping has the form:

```
TEXT
bookings-primary=/run/secrets/cal-diy-webhook-bookings
```

The reference before `=` must:

- contain 1 through 128 ASCII bytes;
- start with an ASCII letter or digit;
- contain only letters, digits, `.`, `_`, `~`, or `-`;
- be unique in the host configuration.

The path after `=` must name a non-empty owner-controlled secret file. The host reads at most 16 KiB from each file. Do not use a comma in the reference or path when the environment variable's comma-delimited form is used.

Subscriber prefixes

Each prefix must be an absolute credential-free HTTPS URL with a host and no query or fragment. A missing trailing slash is normalized. Duplicates are removed.

For a provider webhook create or update, the candidate URL must have the same scheme, host, effective port, and an exact path made from the prefix plus the operation's trusted subscription selector. An empty prefix list denies every provider webhook destination.

Secret mappings control inbound verification. Subscriber prefixes control provider webhook creation. Configuring one does not automatically configure the other.

Fixed webhook bounds

Bound	Fixed value	Operator setting
HTTP request body	1 MiB	None
Signed event maximum age	86,400 seconds	None
Signed event future skew	300 seconds	None
Replay retention	86,400 seconds	None
Replay entries per scope	20,000	None
Supported webhook version	`2021-10-20`	None

These values are source constants, not environment settings.

Shared process and endpoint settings

Environment variable	Required	Default	Meaning and constraint
`AIP_CONNECTOR_HOST_BIND`	No	`0.0.0.0:8081`	Local listener behind deployment ingress
`AIP_CONNECTOR_HOST_PUBLIC_ENDPOINT`	Yes	None	Advertised URL ending exactly in `/aip/v1/messages`
`AIP_CONNECTOR_HOST_CONTROL_ENDPOINT`	Yes	None	Lifecycle endpoint with the exact connector-control path
`AIP_CONNECTOR_HOST_CONTROL_DID`	Alternative	None	Inline control-plane `did:key`
`AIP_CONNECTOR_HOST_CONTROL_DID_FILE`	Alternative	None	File containing the control-plane `did:key`
`AIP_CONNECTOR_HOST_GATEWAY_PRINCIPAL_ID`	Yes	None	Only central gateway principal accepted by the host
`AIP_CONNECTOR_HOST_GATEWAY_DID`	Alternative	None	Inline gateway `did:key`
`AIP_CONNECTOR_HOST_GATEWAY_DID_FILE`	Alternative	None	File containing the gateway `did:key`
`AIP_CONNECTOR_HOST_TLS_CA_FILE`	No	None	Optional PEM root used for private-PKI host, control, callback, and event TLS
`AIP_CONNECTOR_HOST_ALLOW_LOOPBACK_HTTP`	No	`false`	Permits public and control HTTP only with a loopback bind and loopback URLs

For each peer DID, configure the inline value or file, not both. Both peers are required. A DID must parse to a usable verification key.

Public and control endpoints require HTTPS, have no embedded credentials, query, or fragment, and use their exact protocol paths. The loopback exception is a development boundary; enabling it on a non-loopback bind fails startup.

Durable storage and replica identity

Environment variable	Required	Default	Meaning and constraint
`AIP_CONNECTOR_HOST_DATABASE_URL_FILE`	Yes	None	Owner-only PostgreSQL URL file for all host runtime stores
`AIP_CONNECTOR_HOST_SIGNING_SEED_FILE`	Yes	None	Owner-only file containing one 32-byte Ed25519 seed as 64 hexadecimal characters
`AIP_CONNECTOR_HOST_TYPE_ID`	Yes	None	Admitted Cal.diy connector type id
`AIP_CONNECTOR_HOST_VERSION_ID`	Yes	None	Immutable admitted connector version id
`AIP_CONNECTOR_HOST_INSTANCE_ID`	Yes	None	Logical tenant-owned provider instance
`AIP_CONNECTOR_HOST_REPLICA_ID`	Yes	None	Pre-provisioned concrete process identity
`AIP_CONNECTOR_HOST_ARTIFACT_DIGEST`	Yes	None	Exact immutable OCI digest for the running host
`AIP_CONNECTOR_HOST_CONFIG_REVISION`	No	`1`	Positive monotonic non-secret configuration revision

The database schema and durable state must belong to this connector boundary. Changing an artifact, contract, account ownership, or replica identity without updating its admission and configuration identity is not a dynamic

setting.

Tenant and external account settings

Environment variable	Required	Default	Meaning and constraint
`AIP_CONNECTOR_HOST_TENANT_ID`	Yes	None	Verified tenant owned by this instance; non-empty, at most 512 bytes
`AIP_CONNECTOR_HOST_TENANT_SYSTEM`	No	None	Source system for the tenant reference
`AIP_CONNECTOR_HOST_MEMBERSHIP_ID`	Yes	None	Stable verified membership assertion; non-empty, at most 512 bytes
`AIP_CONNECTOR_HOST_EXTERNAL_ACCOUNT_ID`	Pair	None	Stable Cal.diy account reference; 1–512 bytes
`AIP_CONNECTOR_HOST_EXTERNAL_ACCOUNT_SYSTEM`	Pair	None	Product name for the account reference; 1–128 bytes

External account id and system must be configured together or omitted together. A normal Cal.diy instance sets the system to `cal_diy` and binds one tenant to one account-specific host.

Trust, topology, and secret-provider identity

Environment variable	Required	Default	Meaning and constraint
`AIP_CONNECTOR_HOST_TRUST_DOMAIN`	Yes	None	Shared host, gateway, and lifecycle trust domain
`AIP_CONNECTOR_HOST_REGION`	No	`global`	Topology-aware routing region
`AIP_CONNECTOR_HOST_ZONE`	No	`default`	Independent failure zone
`AIP_CONNECTOR_HOST_CAPACITY_CLASS`	No	`standard`	Operator-defined replica capacity class
`AIP_CONNECTOR_HOST_SECRET_PROVIDER_REF`	Yes	None	Non-secret reference naming the deployment secret provider

Region, zone, and capacity class affect declared topology. They do not move a process, configure a network, or increase the local concurrency limit.

The secret-provider reference is not the Cal.diy credential. The standalone binary reads the actual provider secret from its Cal.diy-specific file.

Credential identity and revocation settings

Environment variable	Required	Default	Meaning and constraint
`AIP_CONNECTOR_HOST_CREDENTIAL_ID`	Group	None	Opaque non-secret credential id
`AIP_CONNECTOR_HOST_CREDENTIAL_ISSUER`	Group	None	Trusted handle issuer
`AIP_CONNECTOR_HOST_CREDENTIAL_SCOPES`	Group	Empty	Comma-delimited or repeatable non-empty verified scopes
`AIP_CONNECTOR_HOST_CREDENTIAL_REVISION`	Group	None	Pinned credential revision for assignments and side-effect checks
`AIP_CONNECTOR_HOST_CREDENTIAL_POLICY_FILE`	No	Static current revision	Reloadable non-secret JSON authorization policy

Id, issuer, at least one scope, and revision must be configured together or all omitted. For Cal.diy, the stable logical issuer is `cal_diy_deployment` and operation scopes use `cal_diy:`.

If a policy file is set, it must authorize the configured revision at startup and whenever it is re-read before a provider side effect. Without a file, the host constructs a static policy authorizing the configured current revision.

Callback and event delivery settings

Environment variable	Required	Default	Meaning and constraint
`AIP_CONNECTOR_HOST_CALLBACK_ENDPOINT`	No	None	Fixed central result and stream callback endpoint
`AIP_CONNECTOR_HOST_CALLBACK_ALLOW_HTTP`	No	`false`	Controlled-development exception; requires a callback endpoint
`AIP_CONNECTOR_HOST_CALLBACK_ALLOW_PRIVATE`	No	`false`	Private-address exception; requires a callback endpoint
`AIP_CONNECTOR_HOST_EVENT_ENDPOINT`	Conditional	None	Fixed central connector-event ingress; required when Cal.diy webhooks are enabled
`AIP_CONNECTOR_HOST_EVENT_ALLOW_HTTP`	No	`false`	Controlled-development exception; requires an event endpoint
`AIP_CONNECTOR_HOST_EVENT_ALLOW_PRIVATE`	No	`false`	Private-address exception; requires an event endpoint

Both dispatchers use the host signing identity and the configured private TLS root when present. Network exceptions are explicit; setting an exception without its endpoint fails startup.

Lifecycle and resource limits

Environment variable	Default	Valid range or relation	Effect
`AIP_CONNECTOR_HOST_MAX_IN_FLIGHT`	`32`	Greater than zero	Local concurrent action permits for the replica
`AIP_CONNECTOR_HOST_LEASE_TTL_MS`	`30000`	`1000–300000`	Registry lease lifetime
`AIP_CONNECTOR_HOST_HEARTBEAT_MS`	`10000`	Greater than zero and no more than half the lease TTL	Lease renewal interval
`AIP_CONNECTOR_HOST_DRAIN_TIMEOUT_MS`	`30000`	`1–300000`	Graceful shutdown deadline
`AIP_CONNECTOR_HOST_HEALTH_TIMEOUT_MS`	`2000`	`1–30000`	Provider and storage readiness probe deadline
`AIP_CONNECTOR_HOST_CONTROL_TIMEOUT_MS`	`5000`	`1–30000`	Lifecycle control request timeout

Manifest size, capability count, and native request-body limits also exist in the common host but are fixed by this binary rather than exposed as Cal.diy settings.

File requirements

File	Maximum	Content and permission rule
Provider Bearer or client secret	16 KiB	Non-empty UTF-8 after trimming; regular non-symlink; no Unix group or world permission bits
Webhook HMAC secret	16 KiB	Same secret-file checks; one file per configured reference
PostgreSQL URL	16 KiB	Same secret-file checks
Host signing seed	256 bytes read bound	Exactly 64 hexadecimal characters encoding 32 bytes; same secret-file checks
Gateway or control-plane DID file	512 bytes	Non-empty DID text; same owner-only file checks
TLS CA	1 MiB	Regular non-symlink; no Unix group or world write bits
Credential revision policy	64 KiB	Valid bounded JSON; regular non-symlink; no Unix group or world write bits

The reader checks metadata before and after opening a file and, on Unix, rejects an inode or device change during that operation. It rejects empty or oversized files before parsing them.

Fixed provider versions

The connector selects versions per operation family. They are not deployment settings.

Family	Header or payload version
General and booking routes	`2024-08-13`
Slot and reservation routes	`2024-09-04`
Event type and event-type webhook routes	`2024-06-14`
Private-link routes	`2024-04-15`
Schedule routes	`2024-06-11`
Inbound webhook payload	`2021-10-20`

Changing a Cal.diy upstream revision may require different route versions. Do not override these values outside a source, schema, test, and qualification change.

Configuration supplements

These examples show only Cal.diy additions. They are not complete host or production configurations and intentionally contain no secret value.

Static Bearer account

```
TEXT
AIPD_CAL_DIY_BASE_URL=https://cal.example.test
AIPD_CAL_DIY_ACCOUNT_ID=calendar_primary
AIPD_CAL_DIY_BEARER_TOKEN_FILE=/run/secrets/cal-diy-provider
AIPD_CAL_DIY_MAX_RESPONSE_BYTES=8388608
```

Cal platform client headers

```
TEXT
AIPD_CAL_DIY_BASE_URL=https://cal.example.test
AIPD_CAL_DIY_ACCOUNT_ID=calendar_primary
AIPD_CAL_DIY_OAUTH_CLIENT_ID=client_public_id
AIPD_CAL_DIY_OAUTH_CLIENT_SECRET_FILE=/run/secrets/cal-diy-provider
```

Signed webhook ingress and creation

```
TEXT
AIPD_CAL_DIY_WEBHOOK_SECRET_FILES=bookings-primary=/run/secrets/cal-webhook-primary
AIPD_CAL_DIY_WEBHOOK_SUBSCRIBER_PREFIXES=https://aip.example.test/cal/webhooks/
AIP_CONNECTOR_HOST_EVENT_ENDPOINT=https://aip.example.test/aip/v1/connector-events
```

The provider destination for selector `bookings-primary` must then be exactly `https://aip.example.test/cal/webhooks/bookings-primary`. Deployment ingress must forward that path to the host's `/webhooks/cal-diy/bookings-primary` route; the connector does not configure the external reverse proxy.

Startup and reload behavior

Startup proceeds in this order:

1. parse Cal.diy and common host settings;
2. read and validate one provider authentication group;
3. construct the fixed Cal.diy client and account boundary;
4. apply the provider response bound and webhook destination policy;
5. validate host identity, peer trust, files, endpoints, and credential policy;
6. discover the source-derived manifest without provider I/O;
7. open the PostgreSQL runtime stores and bind Cal.diy durable state to them;
8. configure webhook secrets, replay state, route, and event publisher when

enabled;

9. start the host lifecycle, registration, readiness, and request server.

Setting class	Reload behavior at the reviewed source
Credential revision policy JSON	Re-read before provider side effects
Provider token or client secret	Read once at startup
Webhook HMAC files	Read once at startup
Subscriber prefixes	Parsed once at startup
Identity, topology, endpoints, limits, and API versions	Startup-only

Readiness includes durable storage and connector-local provider health. A syntactically valid configuration can still remain unready because admission, registration, storage, TLS, DNS, or the provider profile request fails.

Configuration failures

Failure	Corrective decision
Both auth groups or a partial client pair	Select exactly one complete group
Unsafe or oversized secret file	Fix the owner-controlled mount; do not weaken file checks or print its content
Base URL fails validation	Use the fixed HTTPS origin with no credentials, query, or fragment
Public or control endpoint fails validation	Restore its exact path and HTTPS boundary; use the loopback exception only on loopback
Peer DID alternatives conflict	Keep the inline value or file, not both, and verify the admitted peer
External account pair is incomplete	Set both id and system or omit both
Credential group is incomplete	Set id, issuer, scopes, and revision together
Credential policy denies the revision	Correct the reviewed policy or roll forward the declared revision; do not bypass the check
Webhook mapping is rejected	Fix `REF=FILE`, identifier grammar, duplicate refs, or file ownership
Webhook publisher is missing	Configure the central event endpoint before enabling webhook secrets
Heartbeat relation is rejected	Keep a nonzero heartbeat at or below half the lease TTL
Host remains unready	Separate configuration, admission, storage, control-plane, and provider-health evidence before rollback

Related documentation

- [Configure authentication and tenant

routing](../getting-started/authentication-and-tenant-routing.md)

- Cal.diy connector overview
- Shared connector-host configuration
- Environment variables
- Deploy the connector fleet