Authenticate and isolate a Twenty workspace

Use this guide to bind one standalone Twenty host to one verified AIP tenant, one fixed provider origin, and one Twenty workspace. The API token stays in an owner-controlled file and is added to provider requests inside the connector.

SECTION	Connectors
TYPE	Connector
PROTOCOL	AIP 1.0
SOURCE	docs/connectors/twenty/getting-started/authentication-and-workspace-isolation.md

The workspace is deployment identity, not Action input. A caller can select an admitted record object or metadata resource, but it cannot select another origin, workspace, API token, webhook secret, or AIP tenant.

Keep the credential layers separate

Six values protect different boundaries. Do not reuse one value for another layer.

Layer	Protects	Secret
Caller credential	Application to `aipd`	Yes
Gateway signature	`aipd` to the connector host	The gateway signing key is secret
Host credential revision	Route authorization before provider work	No provider secret bytes
Twenty API token	Connector host to the fixed workspace	Yes
Twenty webhook HMAC secret	Provider ingress to the host	Yes, and separate from the API token
Host signing seed	Connector host responses and callbacks	Yes

An AIP native bearer-token file authenticates a caller to `aipd`. It must not point to the Twenty API-token file. Credential revision metadata can fence an admitted route, but it does not contain or replace the provider token.

Prerequisites

Before changing the binding, require:

- one admitted Twenty connector version and instance;
- one verified AIP tenant and membership;
- one stable Twenty origin and RFC 4122 workspace UUID;
- one provider token limited to the admitted operation set;
- owner-controlled mounts for every secret file;
- exact host, gateway, control-plane, artifact, and registry identities;
- a rollback window in which the old replica and credential remain valid.

Record source revision, image digest, connector version, instance, replica, tenant, external account, workspace, and credential revision. Do not record secret values.

1. Provision the provider token file

Create the token through the deployment's secret manager and mount it as a regular owner-controlled file. On Unix, any group or world permission bit causes the common secret reader to reject the file.

A representative permission shape is:

```
TEXT
-rw------- 1 <host-uid> <host-gid> ... /run/secrets/twenty-api-token
```

The file must contain one non-empty UTF-8 value after surrounding ASCII whitespace is removed. The Twenty host reads at most 16 KiB. The common reader rejects a symlink, non-regular file, replacement race, empty file, or oversized file and zeroizes its temporary string.

Do not place the token in:

- an Action, manifest, registry row, capability input, or admission package;
- a command-line argument or ordinary environment variable;
- a base URL, image layer, Compose file, log, or evidence bundle;
- the non-secret credential ID, issuer, revision, or provider reference;
- the webhook-secret file.

2. Configure the fixed provider boundary

Configure one origin, workspace, and token-file path:

```
TEXT
AIP_TWENTY_BASE_URL=https://crm.example.test
AIP_TWENTY_WORKSPACE_ID=0190c42f-2d5a-7000-8000-000000000001
AIP_TWENTY_API_TOKEN_FILE=/run/secrets/twenty-api-token
```

The origin must use HTTPS unless it is loopback HTTP or an operator explicitly permits HTTP for a trusted private network. It cannot contain a username, password, path prefix, query, or fragment. Provider redirects remain disabled.

The workspace ID must be a canonical 36-character RFC 4122 UUID with version 1 through 5 and an RFC variant nibble. The connector uses this UUID for record identity derivation and signed webhook workspace comparison.

The connector applies the provider token as an HTTP `Bearer` credential. The origin, workspace, and token are fixed when the host process starts.

3. Bind the host to the tenant and external account

Configure fleet identity separately from the provider credential:

```
TEXT
AIP_CONNECTOR_HOST_TENANT_ID=tenant-crm-production
AIP_CONNECTOR_HOST_MEMBERSHIP_ID=membership-twenty-primary
AIP_CONNECTOR_HOST_EXTERNAL_ACCOUNT_ID=0190c42f-2d5a-7000-8000-000000000001
AIP_CONNECTOR_HOST_EXTERNAL_ACCOUNT_SYSTEM=twenty
AIP_CONNECTOR_HOST_SECRET_PROVIDER_REF=secret://production/twenty/primary
```

External-account ID and system must be present together or both absent. For a tenant-owned workspace, set both and require the external-account ID to equal `AIP_TWENTY_WORKSPACE_ID` through deployment policy.

The reviewed host validates the two settings independently. It does not compare external-account ID with the Twenty workspace automatically. Admission review and deployment validation must detect a mismatch before routing is enabled.

Discovery uses the verified AIP tenant as the manifest namespace when one is present. The provider workspace remains the connector's fixed account boundary. These identifiers serve different roles and need not use the same

syntax.

The registry binding must associate only the intended tenant and admitted Twenty capabilities with this instance. A signed Action cannot override the verified tenant or redirect the host to another workspace.

4. Configure credential identity and revision fencing

The common host can attach a non-secret credential handle to its verified execution context. Configure all handle fields together:

```
TEXT
AIP_CONNECTOR_HOST_CREDENTIAL_ID=twenty-primary-provider
AIP_CONNECTOR_HOST_CREDENTIAL_ISSUER=twenty-deployment
AIP_CONNECTOR_HOST_CREDENTIAL_SCOPES=twenty:record.list,twenty:record.create
AIP_CONNECTOR_HOST_CREDENTIAL_REVISION=twenty-primary-2026-07-27-01
AIP_CONNECTOR_HOST_CREDENTIAL_POLICY_FILE=/run/config/twenty-credential-policy.json
```

ID, issuer, revision, and at least one non-empty scope form one handle. Partial configuration fails startup. Use scopes derived from the admitted operation set rather than a wildcard copied from a controlled fixture.

The reloadable policy can deny the configured or pinned revision before a provider side effect. An unreadable, invalid, or non-authorizing policy fails closed. This mechanism does not hot-reload the provider token file.

5. Keep webhook credentials independent

Configure webhook ingress only when the deployment needs it:

```
TEXT
AIP_TWENTY_WEBHOOK_SECRET_FILE=/run/secrets/twenty-webhook-hmac
AIP_CONNECTOR_HOST_EVENT_ENDPOINT=https://aipd.example.test/aip/v1/events
```

The webhook secret uses the same bounded secret-file reader as the API token, but it has a different purpose and value. Omission disables the Twenty webhook route. Supplying the secret without a central event endpoint fails startup.

The connector accepts a signed webhook only when the payload `workspaceId` equals `AIP_TWENTY_WORKSPACE_ID`. HMAC validity cannot authorize an event from another configured workspace.

Metadata Actions cannot provide a webhook `secret`. Create injects the host-owned configured secret, update injects it only for an explicit rotation request, and connector results recursively remove secret fields.

6. Verify isolation before enabling mutations

Keep new mutation bindings disabled while admitting and checking the host. Verify in this order:

1. the host registers with the expected connector type, version, instance,

replica, artifact digest, tenant, and external account;

2. authenticated health reaches the fixed workspace without printing the token;
3. discovery for the intended tenant exposes only the admitted operations;
4. a bounded `record.list` completes through the signed gateway path;
5. the result belongs to the intended workspace and object;
6. an unauthorized tenant has no eligible Twenty route;
7. an Action cannot provide origin, workspace, or credential material;
8. a revoked credential revision is denied before provider work;
9. a webhook from a different workspace fails even with valid field shapes.

Use a read before any provider mutation. If returned data belongs to another workspace, disable the binding and reconcile workspace, external-account, credential, and registry identities.

7. Rotate the provider token

The Twenty token is read when the host starts. Replacing the mounted file under a running process is not a documented token reload.

Rotate with a bounded replica rollout:

1. create a least-privilege replacement token for the same workspace;
2. mount it at a new owner-controlled path without changing the old replica;
3. allocate a new credential revision and host configuration revision;
4. update revision policy and admission material through controlled paths;
5. start a replacement replica with the new token file;
6. verify authenticated health, a bounded read, and unauthorized-tenant isolation;
7. route new assignments to the replacement and drain the old replica;
8. revoke the old token after in-flight work and rollback needs settle;
9. retain identities, checks, drain state, and revocation evidence without token bytes.

Use a separate controlled procedure for webhook-secret rotation. Provider webhook configuration and host secret rollout must overlap without exposing the secret in Action input or evidence.

Resolve authentication and isolation failures

Symptom	Decision
Token file is rejected	Check file type, ownership, mode, UTF-8 content, and 16 KiB bound without printing it
Provider origin is rejected	Use an allowed origin with no credentials, path, query, or fragment
Workspace ID is rejected	Supply an RFC 4122 UUID version 1 through 5 in canonical form
External-account configuration is rejected	Set ID and system together, then compare the ID with the workspace UUID
Credential handle is rejected	Configure ID, issuer, revision, and non-empty scopes together
Revision policy denies the route	Keep mutations disabled and repair policy or the pinned revision
Host registers but discovery is empty	Compare tenant, membership, operation allowlist, admission, and route policy
Provider returns `401` or `403`	Check token ownership, expiry, permissions, origin, and workspace; do not retry a mutation
Provider returns a redirect	Correct the fixed origin or provider routing; credentials are not forwarded
Webhook workspace differs	Reject the event and reconcile provider webhook ownership with the host binding

Related documentation

- Twenty connector overview
- Twenty connector quickstart
- Rotate connector credentials
- Connector registry and routing
- Security model