Troubleshoot the Twenty connector

Use this runbook to identify the first safe check for a Twenty connector failure. Preserve original instance, replica, Action, key, provider, replay, event, and outbox identities before changing configuration or state.

SECTION	Connectors
TYPE	Connector
PROTOCOL	AIP 1.0
SOURCE	docs/connectors/twenty/troubleshooting/README.md

Begin with the narrowest symptom. Escalate with bounded evidence when the listed check cannot establish a safe recovery.

Triage before changing state

Capture:

- `/health`, `/ready`, and `/metrics` from the affected replica;
- connector type, version, instance, replica, artifact, and configuration revision;
- tenant, external account, workspace, and credential revision;
- operation allowlist digest and discovered manifest digest;
- Action ID, capability, canonical input hash, approval, and key digest;
- provider HTTP status, request ID, error code, retryable, and uncertain flags;
- webhook response, bounded header validation, event ID, replay scope, and outbox state;
- PostgreSQL, registry lease, lifecycle, and deployment events.

Do not capture provider token, webhook secret, signing seed, database password, full idempotency key, raw webhook body, or unredacted personal data.

Remove a not-ready replica from new assignments. Do not delete its database or start a duplicate replica identity during triage.

Diagnose startup failures

Symptom	First check	Do not
Base URL rejected	Require a host, allowed scheme, and no credentials, path, query, or fragment	Enable broad HTTP or redirects
Workspace rejected	Supply a canonical RFC 4122 UUID version 1 through 5	Replace it with an AIP tenant string
API token file rejected	Check regular-file type, owner-only mode, UTF-8, non-empty content, and 16 KiB bound	Print or move the token into environment text
Webhook secret file rejected	Apply the same secret-file checks to a distinct secret	Reuse the provider API token
Response bound rejected	Choose `1` through `268435456` bytes	Raise unrelated ingress limits
Operation file rejected	Check 128 KiB, JSON string array, known suffixes, and non-empty set	Add arbitrary paths as suffixes
Webhook secret causes startup failure	Configure an authorized central event endpoint and trust path	Mount the route without durable publication
External-account fields rejected	Configure ID and system together or omit both	Assume the host compares them with workspace automatically
Credential handle rejected	Configure ID, issuer, non-empty scopes, and revision as one unit	Drop revision fencing to pass startup
DID or signing seed rejected	Verify exact file/value exclusivity and key encoding	Generate replacement identity without admission review

Startup validation is fail-closed. Correct the deployment record or secret projection and replace the replica through the lifecycle process.

Diagnose liveness and readiness

Observation	Meaning	Next check
`/health` fails	Process, listener, ingress, or container boundary	Process exit and bind ownership
`/health` passes, `/ready` returns `503`	At least one route-eligibility condition failed	Structured readiness fields
`draining: true`	Lifecycle intentionally removed readiness	Drain owner and in-flight count
`lease_valid: false`	Registry lease expired or recovery has not committed	Control-plane identity and lease metrics
`runtime.ready: false`	Durable runtime store probe failed	PostgreSQL and recovery report
`connector.ready: false`	Authenticated core OpenAPI probe failed	Origin, TLS, token, workspace, provider status

The connector health probe exercises only `GET /rest/open-api/core`. A passing probe does not prove mutation, metadata, webhook, or custom-object permission.

Do not force route weight onto a `503` replica. Restore all readiness conditions and observe the exact lease before reassignment.

Diagnose missing or rejected capabilities

Symptom	First check	Decision
Capability absent from discovery	Compare operation file, manifest, admission, tenant, route, and lifecycle	Admit the intended fixed suffix through change control
Capability present but invocation says operation not allowed	Compare running process allowlist with admitted manifest	Remove route and correct configuration drift
Another tenant cannot discover the capability	Verify tenant and external-account route binding	Treat as expected isolation unless policy says otherwise
A syntactically valid new suffix is rejected	Compare with the 22-operation source catalogue	Add a typed source operation; do not tunnel a path
Manifest count differs	Compare allowlist set, artifact digest, and config revision	Re-admit the exact replacement artifact and manifest

Discovery is an authorization and routing projection. Editing an operation file does not update a running host.

Diagnose invalid input

`connector.twenty.invalid_request` is a permanent precondition failure for these common causes:

Message or context	Correction
Object grammar	Use 1 through 128 ASCII alphanumeric or underscore characters, starting with a letter
Record or metadata UUID	Use an RFC 4122 UUID version 1 through 5
Unknown metadata resource	Choose one of the twelve fixed resource names
Missing or non-object body	Supply the operation's required JSON body shape
Create-many count	Supply 1 through 200 object bodies
Duplicate body alternatives	Supply exactly one of `data` or `ids`, with 1 through 200 members
Merge IDs	Supply 2 through 9 unique UUIDs
Merge priority	Use an index smaller than the actual ID array length
Unknown or null query	Use only the operation's closed query set and non-null values
Bulk mutation filter	Supply a non-empty value other than whitespace, `{}`, or `null`
Missing mutation key	Supply 1 through 1,024 ASCII graphic bytes
Request body too large	Keep serialized provider JSON within 16 MiB

Corrected input is a new intent when semantics change. Do not reuse an idempotency key that already owns a different canonical input hash.

Diagnose provider failures

Connector code	Category	First decision
`connector.twenty.authentication`	Auth	Verify fixed token, origin, workspace, credential revision, and provider permission
`connector.twenty.remote_temporary`	Temporary	Preserve Action and apply its operation-specific retry contract
`connector.twenty.remote_rejected`	Permanent	Compare current provider schema, body, filter, ID, and permission
`connector.twenty.transport`	Temporary	Determine connect failure versus post-connection ambiguity
`connector.twenty.invalid_response`	Connector	Check response limit, JSON validity, proxy transformation, and provider drift
`connector.twenty.storage`	Temporary	Restore replay profile state without replacing scope

Provider `401` and `403` map to authentication and are not retryable or uncertain. Provider `429` and `5xx` map to temporary failure.

Other provider non-success statuses map to permanent rejection. Retained error details contain only bounded safe fields, so use provider-side evidence under appropriate access when more context is required.

Decide whether a retry is safe

Operation class	Connector retry support	Required decision
Eight reads	Yes	Repeat the same read intent according to policy
`record.create`	Yes	Original Action, key, workspace, object, body, and array index only
`record.create_many`	Yes	Original Action, key, unchanged array order and bodies only
`record.update`	Yes	Original Action, key, target, and body only
Other eleven mutations	No	Reconcile provider state before another write

`retryable: true` does not authorize a new Action, new key, changed input, or different external account. `uncertain_outcome: true` means another provider effect can be unsafe even when retryable is false.

The connector exposes no reconciliation capability. Use bounded authorized reads, provider audit evidence, and the original AIP chain.

Diagnose deterministic create behavior

Symptom	First check
Create returns an existing record	Verify forced `upsert=true`, workspace, key, and derived or explicit ID
Batch member targets an unexpected ID	Compare original array order and zero-based index
Explicit ID is rejected	Validate RFC 4122 version and variant
Repeated create collides	Compare canonical input hash and key ownership
Provider result lacks a usable ID	Stop and perform a bounded read; do not guess or repeat create

The connector derives missing UUID v5 values from workspace, key, and index. Changing any input changes deterministic ownership.

Upsert and deterministic IDs limit duplicate risk. They do not prove global exactly-once provider behavior.

Diagnose metadata and webhook-secret input

Symptom	First check
Metadata resource rejected	Use one of the twelve fixed names and an operation allowed for it
Page-layout type ignored or rejected	Supply valid type with `objectMetadataId`
Tabs or widgets list rejected	Supply required `pageLayoutId` or `pageLayoutTabId` UUID
Webhook create rejects `secret`	Remove it; connector injects the host-owned value
Webhook create rejects rotation control	Remove `rotateConfiguredSecret`; it is update-only
Webhook update rejects rotation	Use exactly Boolean `true`, not `false` or a string
Webhook mutation says not configured	Provision the host webhook secret and event endpoint through deployment
Read-back omits a secret	Treat as expected recursive redaction

All metadata mutations are critical and unsafe to retry. Refresh the provider metadata schema and dependency graph before any replacement intent.

Diagnose restore and merge

Symptom	First check
Single restore uses a collection request	Expected mapping at pinned revision; connector adds `filter=id[eq]:<id>`
Restore finds no record	Verify prior soft-delete state, exact workspace, object, and ID
Restore after permanent destroy fails	Expected boundary; no connector rollback exists
Merge body rejected	Check exact keys, 2-to-9 unique IDs, in-range priority, and Boolean dry-run
Merge result is uncertain	Preserve all IDs and read current survivor state before another mutation

The single-restore workaround exists because the pinned provider path parser rejects its declared three-segment route. Requalify this mapping when provider revision changes.

Diagnose webhook HTTP responses

Response	First checks	Unsafe workaround
`400 webhook.missing_header`	Exact three header names, encoding, and non-empty values	Synthesizing headers at an untrusted proxy
`401 webhook.authentication_failed`	Raw bytes, time, signature, secret revision, nonce, JSON, workspace, payload fields	Logging secret or body to reveal subreason
`503 webhook.storage_unavailable`	Replay store, event log, PostgreSQL, profile state, durable outbox	Clearing replay or outbox state
`200 centrally_acknowledged`	Search central event by deterministic ID	Treating it as proof of downstream consumer completion
`200 durably_queued`	Inspect local outbox and central route	Asking provider to resend the same nonce

Nonce admission happens before JSON validation and event persistence. A late failure can leave the nonce fenced without a published event.

Recover from the stage that owns the delivery. Do not change instance identity, event ID, or nonce merely to bypass the fence.

Diagnose storage, lease, and drain state

Symptom	First check	Recovery boundary
Runtime storage not ready	Database URL, TLS, role, capacity, locks, and recovery report	Preserve Action, replay, event, and outbox state
Admission blocked by recovery errors	Exact queued-Action and delegation records	Repair records; do not force registration
Lease expired	Provider and storage readiness plus control-plane path	Stable recovery request identity
Repeated heartbeat failure	Connector probe, storage probe, TLS, DID, registry identity	Keep replica unrouted
Drain does not complete	In-flight gauge and original Action statuses	Wait within deadline, then preserve uncertainty
Offline transition fails	Control-plane state and old lease identity	Do not start duplicate replica ID

An empty replacement database is not recovery. It can remove durable fences while provider effects and central events remain.

Prepare an escalation bundle

Include:

- exact source revision and immutable host image digest;

- connector version, instance, replica, tenant, workspace, and config revision;
- redacted effective configuration and operation allowlist digest;
- `/ready` body and bounded metric snapshot;
- Action, approval, key digest, checkpoint, result, error, and receipt identities;
- provider status and bounded request ID;
- for webhooks, response class, header-shape results, replay scope, event ID, and outbox outcome;
- PostgreSQL recovery, lease, drain, deployment, and incident timestamps;
- commands and decisions already applied;
- explicit missing evidence and unresolved uncertainty.

Exclude secret bytes and unredacted customer data. State whether evidence comes from source review, deployed artifact inspection, controlled execution, or live provider observation.

Related documentation

- Twenty connector configuration
- Twenty capability index
- Twenty health, observability, and recovery
- Twenty webhook security, replay, and events
- Errors and retry decisions