

Troubleshoot the Dify connector

Use this runbook to identify the failed boundary and choose the narrowest safe action. Pause related mutations whenever the provider outcome is uncertain.

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

Preserve the original Action ID, idempotency key, route, capability, descriptor identity, provider status, task ID, and durable state before changing a deployment.

Identify the evidence surface

Surface	Evidence shape	First trust check
Process startup	Error text and non-zero exit	Configuration, files, storage, and admission
Pre-envelope host rejection	Unsigned compact JSON	Controlled TLS path; no valid envelope exists to sign
Connector-host response	Signed AIP envelope	Signature, sender, recipient, and correlation
Durable Action	State, result, and events	Read it even when the HTTP exchange failed
Dify provider	Status, response, request ID, or task ID	Fixed origin and scoped credential
Stream callback	Ordered chunks and terminal result	Action identity and callback ownership

Automate on exact code, category, retry flag, uncertainty flag, and durable state. Message text is diagnostic and is not a stable interface.

Choose the first check

Symptom	First check	Safe first action
Process exits	First startup error	Correct one owning input; do not suppress validation
Capability is absent	Authenticated manifest and descriptor mode	Correct configuration or admission; never construct the ID manually
<code>/health</code> works and <code>/ready</code> fails	Drain, lease, runtime, and connector fields	Follow the failing owner
One new credential makes readiness fail	App or Knowledge probe for that descriptor	Correct or remove it through replacement
Provider returns 401 or 403	Capability domain and selected key file	Rotate the scoped credential through replacement
Read fails temporarily	Published retry support and Action budget	Retry only the same eligible read
Mutation has no definitive result	<code>uncertain_outcome</code> and provider state	Reconcile before another mutation
Stream ends without terminal output	Last event, task mapping, and cleanup result	Preserve uncertainty and inspect remote task state
Cancellation is unconfirmed	Durable task ID and original user	Do not report or assume remote stop
Binary output looks corrupt	Base64, size, and media policy	Decode the declared envelope; do not use text fallback

Resolve startup failures

Startup fails closed for these product conditions:

- both descriptor domains are empty;
- a descriptor file is malformed, oversized, or above its count ceiling;
- an ID, name, mode, or key-file reference is invalid;
- app or Knowledge IDs are duplicated;
- an app combines global and per-app key sources;
- an app has no available key source;
- a key file is empty, invalid UTF-8, unreadable, or oversized;
- the provider URL is not a permitted origin;
- the response limit is outside its supported range;
- host admission, durable storage, identity, or task-store preparation fails.

Correct the first reported boundary and restart as a replacement replica. Product descriptors and secrets are not live-reloaded.

Require `/ready`, exact manifest discovery, and a read-only provider canary after repair. A direct Dify request does not prove AIP route or admission.

Follow readiness to its owner

Failing field	Owner	Decision
<code>draining</code>	Host lifecycle	Let pinned work settle or replace the replica
<code>lease_valid</code>	Registry heartbeat and fencing	Recover the admitted identity through control plane
<code>runtime.ready</code>	Durable PostgreSQL state	Restore storage before accepting work
<code>connector.ready</code>	Dify origin and all configured credentials	Isolate the failed descriptor or provider boundary
Ready host is not selected	Registry binding and route	Re-resolve the admitted route
<code>connector_host.capacity</code>	Replica-local concurrency	Apply bounded backoff under Action policy

Connector health probes every configured key. One inaccessible app or Knowledge workspace makes the connector not ready.

Look up Dify connector codes

Code	Meaning	Decision
<code>connector.dify.authentication</code>	Dify returned 401 or 403	Correct key, scope, descriptor domain, or provider access
<code>connector.dify.remote_temporary</code>	Dify returned 429 or 5xx	Retry a safe read only; reconcile a mutation
<code>connector.dify.remote_rejected</code>	Other non-success status	Correct input or provider state; inspect mutation outcome
<code>connector.dify.transport</code>	Provider transport failed	Retry a safe read only; mutation may be uncertain
<code>connector.dify.invalid_stream</code>	SSE framing, data, bound, or terminal rule failed	Preserve task evidence and cleanup result
<code>connector.dify.response_too_large</code>	Complete response crossed the configured bound	Reduce output or review the bounded limit
<code>connector.dify.cancelled</code>	Runtime cancellation interrupted work	Check remote-stop confirmation
<code>connector.dify.task_store</code>	Durable task correlation failed	Restore profile state; do not claim cancellation
<code>connector.dify.invalid_credential</code>	Key material is invalid UTF-8	Replace the owner-only key file and process
<code>connector.dify.configuration_or_input</code>	Configuration, capability, URL, or Action input is invalid	Correct the named boundary; do not retry unchanged

The legacy untyped connector mapping uses the generic `connector.dify` code with `retryable: false`. Prefer the signed typed result from the standalone host and preserve which execution path produced the evidence.

Diagnose discovery differences

Observation	Decision
App alias is absent	Verify app descriptor, key source, parsing, and admission
One suffix is absent	Compare its compiled mode projection
All Knowledge suffixes are absent	Verify a separate Knowledge descriptor and key
Count differs by one	Include the app alias in the calculation
Count differs by 46	Compare Knowledge credential count
Manifest changed after file edit	Replace the startup-owned replica
Provider rejects a projected suffix	Check the documented mode mismatch before changing discovery

Do not call the host directly or synthesize a missing capability ID to bypass tenant discovery.

Diagnose input and route failures

Symptom	First check
Placeholder remains in provider path	Required top-level path field and exact name
Query is rejected	<code>query</code> must be an object with scalar or scalar-array values
GET is rejected before dispatch	Remove the request <code>body</code>
JSON mutation is rejected	<code>body</code> must be an object and a stable key must exist
User-scoped operation is rejected	Supply the required top-level <code>user</code>
Upload is rejected	Verify base64, filename, size, and multipart data support
App invocation lacks content	Supply query or prompt for completion and chat families
Provider route is unexpected	Compare app mode and capability suffix, not display name

Correct the original Action input. Do not move path or query fields into `body`, because the connector owns their wire placement.

Diagnose Knowledge operations

Use the Knowledge credential selected by the capability prefix. App keys and the legacy global app key cannot authorize Knowledge routes.

Async document indexing returns provider document and batch identities. Poll or read those identities before treating creation as searchable completion.

Multi-document metadata changes can commit per document. A partial provider failure requires per-document reconciliation, not whole-request replay.

Segment update can return HTTP success while embedded data reports an indexing failure. Inspect the semantic body before declaring success.

Diagnose streams and cancellation

Typed streaming exists for every app alias and exactly two explicit operations: `workflow.run_by_id` and `workflow.event.stream`.

Pipeline run and node-run are buffered because their frozen response kind is JSON. Missing chunks for those capabilities is expected connector behavior.

An invalid stream can follow provider admission. Preserve the last valid event, task ID, response bound, and cleanup outcome.

Cancellation of `workflow.event.stream` closes only the local subscription. Other remote cancellation needs a captured task ID and the original `user`.

Diagnose binary output

Only `file.preview`, `audio.synthesize`, and `document.download_zip` return the binary envelope. Decode `content_base64`, verify `size_bytes`, and enforce the declared media policy.

Non-JSON output from a JSON-declared operation becomes a diagnostic text fallback. It is not binary-safe.

Decide whether to retry

Apply these checks in order:

1. verify the signed response or durable Action result;
2. identify the exact catalogue operation;
3. classify it as read or mutation;
4. establish provider outcome for any mutation;
5. require published retry support and a true retry flag;
6. preserve Action ID, input, and idempotency key;
7. respect attempt, elapsed-time, and backoff limits.

Only safe reads can become retryable for Dify temporary, transport, or invalid-stream failures. No connector-mapped mutation is automatically safe to retry.

Collect and route evidence

Retain a bounded incident bundle:

- UTC interval, source and deployment revisions, instance, replica, and tenant;

- descriptor identity, mode, credential revision, and manifest hash;
- Action, correlation, capability, route, and key-reference identities;
- exact code, category, retry flag, uncertainty flag, and remote status;
- readiness snapshot and eight host metrics;
- last valid stream event, task ID, cleanup result, and provider request ID;
- redacted provider state used to reconcile a mutation.

Route configuration defects to the connector owner, lease and capacity defects to the fleet operator, and Dify-access defects to the credential owner.

Escalate cross-tenant output, route misbinding, signature failure, credential exposure, lost task state, and unresolved mutation outcomes as integrity or security incidents.

Verify recovery

Close the incident only when the normal gateway route works, readiness and lease agree, exact manifest discovery is restored, and durable Action state is terminal.

Every uncertain provider mutation needs an explicit reconciliation decision. Record evidence and remove temporary containment deliberately.

Related documentation

- [Dify health, observability, and recovery](#)
- [Dify connector configuration](#)
- [Dify streaming, cancellation, and binary responses](#)
- [Dify modes and capability projection](#)
- [Errors and retry decisions](#)