

Cal.diy isolated-live qualification on 13 July 2026

Historical result: PASS. The retained artifact records qualification at 2026-07-13T12:05:20.671410Z and restart verification at 2026-07-13T12:05:26.883165Z.

SECTION	Qualification and Evidence
TYPE	Qualification evidence
PROTOCOL	AIP 1.0
SOURCE	docs/testing/cal-diy-isolated-live.md

This result applies only to the AIP source digest, harness, images, Cal.diy revision, topology, and assertions identified below. It does not qualify the current documentation review revision `97be86e9efedf07ecf1783b03800f683f107fb04`, another Cal.diy deployment, or the complete AIP platform.

Understand the evidence level

The result is historical live-product evidence. Its strongest retained record is a redacted JSON artifact with exact identities and observations. The original narrative report adds procedure and control detail, while retained source confirms the structure of the campaign. Neither source code nor a successful historical status proves that the campaign passed again.

Evidence	What it supports	Limitation
Retained JSON result	Exact artifact identities, UTC times, status, correlation values, product observations, webhook outcomes, and restart result	It does not contain raw logs, command exits, cleanup confirmation, or a checksum manifest of the full run
Historical narrative	Declared scope, topology, security controls, procedure, assertions, and supplemental deterministic gates	These statements are not all independently represented in the JSON result
Retained source	The Cal.diy commit and tree, historical AIP base commit, and current form of the qualification campaign	The run included uncommitted AIP material identified only by digest

The retained JSON bytes have SHA-256 digest `507c2136aa695a5e34efddd6fe5dc6b50e46de7d898c5ad39c3d5f7ebbbb82b3`.

Identify the exact artifacts

Artifact	Recorded identity
AIP source base commit	d1d1a79d030d16e12d2ec8b46201e29cdfbdce5e
AIP source digest used by the build	07b500c9ac0ff8ac5becc6b77c5661587f8e08de97522e21fed0e98edd94cee0
Qualification harness digest	605d580cb913ccf6673e6ca6465261b5593e71df806ca93f4b6d9def79950691
AIP qualification image	sha256:60ace4f074ad7f790801c6171e3f1dc9b66e7bf0dea4ed7291233a5b107131fd
AIP builder image	sha256:77237dd363a0b127bb5ef532c2d64c0deb380b738e43a9c4bdac73398d6d0a08
AIP runtime base image	sha256:240c36104698159b16dd76db37a24d97d04487d7635a6bb1a6f60064141fc969
Cal.diy upstream commit	f00434927386c9ecdcbd7e6c5f82d22044a245bc
Cal.diy source tree	ae0db7000d0479c20af5c82de60863a34165e2e2
Cal.diy image	sha256:52be5e3d24f65c4c42793702d93e68fa19ffa2c2fd45269e3ae60e6e81f0cc4a
Qualification runner image	sha256:50ae7f4dda0ef887ff4259bcfb610a2d6d4ea78bf6440a49c0ae583050a3a01a

The Cal.diy commit is present in the retained official upstream checkout. Its Git tree is exactly the source-tree value in the artifact. The AIP base commit is retained in a historical source checkout, but the build's source digest is not that commit's Git tree because the campaign included uncommitted material.

The digest makes that historical input distinguishable. It does not preserve the changed files or make the exact source tree reconstructable. A new release claim therefore requires a clean build and a new campaign.

Review the isolated topology

The historical report describes:

- one AIP daemon and one pinned Cal.diy API deployment;
- separate PostgreSQL databases for AIP and Cal.diy;
- Redis and an OAuth 2.0 token-introspection service;
- one non-agent qualification runner;
- two internal container networks with no published qualification port.

The machine artifact records the dedicated project name `aip-cal-diy-qualification`, `external_network_access: false`, and an empty `agent_services` list. These values establish the recorded isolation boundary; they do not describe a production network.

Review the reported security boundary

The historical procedure used separate OAuth bearer tokens and distinct requester and approver principals. The retained artifact identifies the two principals and states `credential_material_in_protocol: false`.

The narrative further reports these controls:

- trusted identity mapped the requester to one tenant, one external Cal.diy

account, and one opaque credential handle;

- the host resolved protected credential material only at invocation time;

- envelopes, MCP results, manifests, lifecycle state, evidence, and errors did

not contain credential material;

- the native AIP endpoint used a separate owner-only bearer-token file;
- webhook verification used the exact raw body, HMAC-SHA256, a signed-time

window, constant-time comparison, and durable duplicate suppression.

The JSON result retains the credential-boundary boolean and webhook outcomes, not the secret files or raw signed requests. This is appropriate for redaction, but it prevents an independent byte-for-byte signature review from this artifact alone.

Reconstruct the reported procedure

The historical campaign reported this sequence:

1. Create owner-only runtime secrets and reset only the dedicated campaign project and its volumes.
2. Build the identified AIP image without registry access and inspect selected source files in the identified Cal.diy image.
3. Start the isolated product, databases, Redis, identity service, AIP daemon, and qualification runner.
4. Verify native AIP and MCP Streamable HTTP discovery.
5. Read real availability, plan a booking without mutation, and submit a commit that waits for independent approval.
6. Approve the immutable request through a separately authenticated principal and verify one correlated booking.
7. Replay the same action and idempotency identity and verify no second booking.
8. Exercise valid, duplicate, invalid-signature, and stale webhook deliveries.
9. Restart the AIP namespace owner and bound processes, then verify durable action, transaction, event, idempotency, and provider facts.

The historical report says the stack remained available for inspection after success. It does not retain an independent cleanup observation.

Inspect the retained results

Observation	Retained value
Result schema	<code>aip.cal-diy.qualification.v1</code>
Status	<code>passed</code>
Native message type	<code>aip.core.v1.action_result</code>
MCP transport	<code>streamable_http</code>
Stable MCP tools verified	<code>true</code>
Admitted Cal.diy capabilities	<code>81</code>
Available slot capacity	<code>8</code>
Booking count before	<code>0</code>
Pending approval observed	<code>true</code>
Authenticated decision observed	<code>true</code>
Completed action state	<code>completed</code>
Booking count after approval and replay	<code>1</code>
Idempotent replay observed	<code>true</code>
Valid webhook delivery	<code>HTTP 204</code>
Duplicate webhook delivery	<code>HTTP 204; retained event count 1</code>
Invalid webhook HMAC	<code>HTTP 401</code>
Stale webhook timestamp	<code>HTTP 400</code>
Restart verified	<code>true</code>

The artifact also retains action, plan, commit, approval, transaction, run, booking, booking-seat, webhook-marker, and subscription correlation values. Those identifiers make the observations internally traceable without exposing credentials.

Separate reported assertions from retained facts

The historical narrative reports eleven successful boundaries: native invocation, independent MCP sessions, the 81-capability admission set, live slot discovery, non-mutating plan, approval-blocked commit, authenticated approval, one booking, idempotent replay, four webhook cases, and restart recovery.

The JSON artifact retains a related observation for each group, but it does not prove every intermediate request, provider response, database query, or process log independently. Treat the page as a result record, not as a replay of the evidence-generation process.

Account for skipped and missing evidence

The result excludes:

- every connector other than Cal.diy;
- capacity, latency, backup, disaster recovery, and rollout qualification;
- production credentials, traffic, data, topology, and failure modes;
- operations outside the campaign's booking, approval, replay, webhook, and restart scope;

- independent evidence that the historically reported workspace format, lint, unit, integration, and conformance commands passed;

- raw logs, command exit records, a full-run checksum manifest, and cleanup

confirmation.

The current qualification standard makes verified cleanup part of `PASS`. Because this historical artifact has no cleanup field and the narrative says the stack remained available for inspection, it must not be accepted as a new campaign result under the current standard. Its recorded historical status remains `PASS` under the scope used on 13 July 2026.

Reproduce only a new campaign

The current source retains a campaign entry point:

```
SH
examples/cal-diy-qualification/qualify.sh
```

Running it would create a new result for the current source, harness, images, configuration, and time. It would not reproduce the historical run unless all recorded inputs, including the unavailable uncommitted source material, were restored exactly. Follow the current [live-product procedure](#) and record the new evidence independently.

Retained evidence

- [Machine-readable Cal.diy result](#)
- [Testing and qualification index](#)
- [Live-product qualification procedure](#)
- [Connector upstream baselines](#)
- [Implementation status](#)
- [Cal.diy connector](#)