

# CrewAI training and testing capabilities

The train and test operations run bounded CrewAI evaluation workflows through the durable sidecar job boundary. Both can invoke models, tools, and crew tasks repeatedly, so iteration count is part of approval and cost review.

|          |                                                             |
|----------|-------------------------------------------------------------|
| SECTION  | Connectors                                                  |
| TYPE     | Connector                                                   |
| PROTOCOL | AIP 1.0                                                     |
| SOURCE   | docs/connectors/crewai/capabilities/training-and-testing.md |

Training writes an operator-owned artifact. Testing selects an evaluator model and returns completion metadata, not a connector-normalized score report.

## Compare train and test

| Property                | train                      | test                      |
|-------------------------|----------------------------|---------------------------|
| Capability              | cap:crewai:<crew_id>:train | cap:crewai:<crew_id>:test |
| Kind                    | Workflow                   | Workflow                  |
| Risk                    | High                       | Medium                    |
| Provider mutation       | Yes                        | Yes                       |
| Approval                | Required                   | Required                  |
| Idempotency key         | Required                   | Required                  |
| Streaming contract      | Yes                        | Yes                       |
| Advertised cancellation | Yes                        | Yes                       |
| Connector retry         | No                         | No                        |
| Required identity       | Safe artifact name         | Evaluator model string    |

Both operations create one durable sidecar job under their AIP Action ID. They execute the pinned synchronous CrewAI method in a worker thread.

## Train a crew

The exact input has two required fields and one optional object:

```
JSON
{
  "n_iterations": 3,
  "artifact_name": "support-training-2026-07.json",
  "inputs": {
    "scenario_set": "reviewed-support-cases"
  }
}
```

| Field         | Constraint                                        | Meaning                                           |
|---------------|---------------------------------------------------|---------------------------------------------------|
| n_iterations  | Integer from 1 through 100                        | Number of CrewAI training iterations              |
| artifact_name | ^[A-Za-z0-9][A-Za-z0-9._-]{0,127}\$               | Basename inside the configured training directory |
| inputs        | Optional object, at most 512 top-level properties | Inputs supplied to each training kickoff          |

Unknown fields are rejected. The complete encoded input also shares the 1 MiB sidecar limit.

The sidecar resolves the artifact as:

```
TEXT
<AIP_CREWAI_TRAINING_DIR>/<artifact_name>
```

The basename pattern excludes path separators and traversal segments. It does not create artifact versions, checksums, retention policy, or write-once semantics.

## Understand the reviewed training behavior

The sidecar calls:

```
TEXT
crew.train(n_iterations, artifact_path, inputs)
```

The operation catalogue records CrewAI source `bfa652a7`, whose package version is `1.15.2`, as its method-verification baseline. The default product image instead installs the frozen `1.15.5` package set.

The reviewed source method copies the crew, marks its tasks for human input, disables agent delegation in the training copy, and runs every iteration. It evaluates collected data per agent and writes to the selected artifact.

Those internal steps describe the source-verification baseline. The sidecar call signature is connector code, but a built `1.15.5` artifact needs its own qualification evidence before those internals are attributed to it.

A completed sidecar result has this shape:

```
JSON
{
  "status": "completed",
  "artifact_name": "support-training-2026-07.json",
  "n_iterations": 3
}
```

The result confirms that the pinned method returned. It does not include artifact bytes, a digest, evaluation details, or proof that another process cannot later change the file.

## Protect the training directory

`AIP_CREWAI_TRAINING_DIR` is required when the sidecar enables `train`. The path is created when absent and must resolve to a non-symlink directory.

Any group or world write bit causes startup to fail. The controlled deployment places the directory on the connector-owned durable sidecar volume.

Apply these controls outside the capability input:

- dedicate the directory to one active sidecar instance;
- assign owner-only write access to the sidecar user;
- inventory an artifact name before approving a replacement;
- copy accepted artifacts into versioned, integrity-protected storage;
- retain source, crew, model, tool, input-set, iteration, and approval identity;
- scan artifacts for confidential data before broader distribution.

An AIP caller selects only the safe basename. It cannot change the training root.

## Test a crew

The exact input is:

```
JSON
{
  "n_iterations": 5,
  "eval_llm": "openrouter/reviewed-evaluator",
  "inputs": {
    "scenario_set": "release-candidate-17"
  }
}
```

| Field        | Constraint                                                 | Meaning                                       |
|--------------|------------------------------------------------------------|-----------------------------------------------|
| n_iterations | Integer from 1 through 100                                 | Number of evaluated crew executions           |
| eval_llm     | Non-empty string, at most 256 bytes, no control characters | Evaluator passed to CrewAI model construction |
| inputs       | Optional object, at most 512 top-level properties          | Inputs supplied to evaluated kickoffs         |

Unknown fields are rejected. The connector does not map `eval_llm` to a credential or allowlist. The deployment's CrewAI configuration and provider environment determine whether that string resolves and which secret it uses.

At the pinned revision, the sidecar calls:

```
TEXT
crew.test(n_iterations, eval_llm, inputs)
```

CrewAI constructs the evaluator LLM, copies the crew, runs each iteration, and prints its evaluation result through CrewAI's own output path.

The connector returns only bounded completion metadata:

```
JSON
{
  "status": "completed",
  "n_iterations": 5,
  "eval_llm": "openrouter/reviewed-evaluator"
}
```

Do not treat this object as a scorecard. Retain the approved evaluator output through a deployment-owned evidence pipeline when scores are release evidence.

## Review approval and cost

Approval uses the complete input snapshot. Review at least:

| Decision                                | rail           | test           |
|-----------------------------------------|----------------|----------------|
| Crew identity and admitted artifact     | Required       | Required       |
| Iteration count                         | Required       | Required       |
| Input dataset authority and sensitivity | Required       | Required       |
| Models and tools invoked by crew tasks  | Required       | Required       |
| Provider budgets and rate limits        | Required       | Required       |
| Artifact target and overwrite policy    | Required       | Not applicable |
| Evaluator identity and credential scope | Not applicable | Required       |
| Telemetry and evidence destinations     | Required       | Required       |

The AIP risk label does not estimate token consumption. One approved Action can run the crew up to 100 times and can trigger additional evaluator calls.

Use separate Action IDs and keys for training and testing. A successful test does not authorize a later training mutation.

## Interpret streaming

Both capabilities advertise streaming because their durable jobs publish ordered `started` and terminal events. The connector can return those events through the streaming AIP path.

The pinned sidecar sets `crew.stream = false` for both methods. It does not project per-token CrewAI chunks from their synchronous worker threads.

Streaming transport therefore improves lifecycle observation. It does not turn training or testing into an interruptible provider call.

## Handle cancellation and timeout honestly

Both operations advertise cancellation. The sidecar accepts the request, cancels the async task, persists `cancelled`, and the Rust connector can return `remote_stop_confirmed: true`.

The synchronous provider method still runs in a worker thread. The reported confirmation acknowledges the sidecar route; it does not establish that training, evaluation, tools, or provider calls stopped.

The configured operation timeout bounds how long the coordinator waits. It does not terminate the underlying thread or undo model, tool, or file effects.

On a provider exception or timeout, the sidecar records `failed` with an exception class and generic message. That ordinary terminal event has no `uncertain_outcome` field, so operators must classify effects from operation semantics and external evidence.

The pinned sidecar logs the failure with `LOGGER.exception`. Provider exception text and traceback can therefore reach restricted logs even though the durable event omits them.

After a sidecar restart, a restored `running` record becomes failed with an uncertain outcome. The sidecar does not silently re-run it.

## Recover without duplicating work

1. preserve the original Action ID, key, input hash, crew, and artifact or evaluator identity;
2. read durable status and events for that Action;
3. inspect the artifact path for training without executing or overwriting it;
4. inspect provider, tool, and evaluator records correlated to the Action;
5. classify completion, failure, cancellation, or continuing uncertainty;
6. accept, quarantine, or remove an artifact under operator policy;
7. create a new Action only after reconciling every possible prior effect.

Do not repeat an Action with a different input under the same ID. The durable request hash rejects that collision.

## Diagnose capability failures

| Symptom                                     | Decision                                                               |
|---------------------------------------------|------------------------------------------------------------------------|
| <code>train</code> is absent from discovery | Review descriptor and sidecar allowlists before changing either        |
| Sidecar requires a training directory       | Configure a protected durable path before enabling <code>train</code>  |
| Artifact name is rejected                   | Use a safe basename of at most 128 characters                          |
| Artifact already exists                     | Apply deployment overwrite and version policy before approval          |
| Evaluator is rejected                       | Use a non-empty bounded model string without control characters        |
| Evaluator cannot authenticate               | Repair deployment-owned provider credentials, not Action input         |
| Action times out                            | Treat model, tool, and artifact effects as uncertain                   |
| Job reports <code>cancelled</code>          | Treat the worker thread and external effects as potentially continuing |
| Completion lacks scores                     | Capture CrewAI evaluator output through the approved evidence path     |
| Sidecar restarted during work               | Use the restored failed record and reconcile before any new Action     |

## Related documentation

- [CrewAI capability index](#)
- [CrewAI connector configuration](#)
- [Authentication and admitted crews](#)
- [Approvals and policy](#)
- [Errors and retry decisions](#)