

Fleet scale SLO

Use this procedure to check that connector catalog and route resolution remain within the source-owned `aip-fleet-slo-v1` regression bounds. The gate uses a real PostgreSQL database and synthetic registry records. It measures registry behavior

SECTION	Qualification and Evidence
TYPE	Qualification evidence
PROTOCOL	AIP 1.0
SOURCE	docs/testing/fleet-scale-slo.md

This page describes the test at source revision `97be86e9efedf07ecf1783b03800f683f107fb04`. No scale run was performed while preparing the page. A result applies only to the recorded source, database, host, dataset, configuration, and time interval.

Prevent a silent skip

The test is opt-in. If `AIP_POSTGRES_SCALE_TEST_URL` is missing or empty, it prints a skip message and returns success. A green `cargo test` exit status is therefore not sufficient evidence.

Accept a run only when its combined output contains exactly one final line beginning with:

```
TEXT
AIP_SCALE_RESULT=
```

The JSON after that prefix must identify `aip-fleet-slo-v1` and contain `"passed": true`. CI and local wrappers must assert the result line rather than relying only on the test process exit code.

Use the default dataset for comparable results

Dataset dimension	Default	Definition
Connector types	1,000	One active version is initially created for each type
Global capabilities	1,000	Unique definitions reused across versions and tenants
Capability providers	100,000	Each version supplies 100 capabilities
Connector instances	10,000	Distributed across 100 tenants and the connector types
Ready replicas	10,000	One replica with capacity 100 per instance
Tenant bindings	100,000	Ten capability bindings per instance
Tenants	100	Fixed by the test
Parallel catalog reads	32	One first-page read for each of the first 32 tenants
Concurrent route resolutions	500	Timed route assignments for one tenant and capability
Lease churn	1,000 replicas	The smaller of 1,000 and the configured instance count

The configurable counts must be positive. Capabilities must be at least 100, and instances must be at least the number of connector types.

Override these values only for a named custom campaign:

Environment variable	Default
AIP_SCALE_CONNECTOR_TYPES	1000
AIP_SCALE_CAPABILITIES	1000
AIP_SCALE_INSTANCES	10000
AIP_SCALE_CONCURRENT_ROUTES	500

The thresholds do not change when the dataset is overridden. Record every override and do not compare a custom dataset directly with the default CI baseline.

Apply the fixed thresholds

Metric	Gate	Exact source value
Seed and lifecycle setup	At most 60 seconds	60000 ms
Parallel catalog p99	At most 2 seconds	2000000 us
Concurrent route p99	At most 3 seconds	3000000 us
Route throughput	At least 100 assignments per second	100.0/s
Rust test process high-water RSS	At most 512 MiB	524288 KiB
PostgreSQL database growth	At most 1 GiB	1073741824 bytes

The setup timer includes seeding plus the version rollout, noisy-tenant case, and retry-storm case. It is broader than raw insert time.

Catalog p99 is computed from 32 concurrent first-page reads at the default dataset. Route p99 and throughput use the 500 concurrent route tasks. The test uses its own integer percentile function over the collected samples; do not substitute a monitoring system's percentile method when comparing results.

Process RSS is the Rust test process high-water mark. It does not include PostgreSQL memory. Database growth is the change in `pg_database_size` for the current database and can be distorted by unrelated concurrent work. Use an isolated database.

If the platform-specific RSS probe fails, the source records zero instead of failing the test. Reject a memory-bound claim when `process_rss_kib_observed_max` is zero or otherwise implausible.

Verify the functional invariants

The numeric thresholds are only part of the gate. The same run must also prove:

Invariant	Required behavior
Version rollout	One instance moves to a new active version, its old replica drains, and a new ready replica serves the new version
Noisy-tenant isolation	Of 64 simultaneous requests for a tenant with budget four, four are admitted and 60 are rejected while four healthy-tenant requests succeed
Retry-storm isolation	Of 64 simultaneous retries with budget eight, eight are admitted and 56 are rejected while eight healthy requests succeed
Catalog pagination	Page size never exceeds 50, revision and total remain stable, and every returned capability is unique
Route distribution	Every eligible replica is used when the sample is large enough, and the largest and smallest assignment counts differ by at most three
Reservation release	Settling all timed routes leaves zero active assignments in the run's replicas
Retry fencing	Up to 100 repeated action IDs resolve to their original assignments

Invariant	Required behavior
Credential rotation	An existing route remains pinned to credential revision one while a new action uses revision two
Indexed routing	The plan does not sequentially scan tenant bindings or replicas and uses an accepted binding index
Lease churn	Up to 1,000 leases renew, expire, and transition through bounded batches

These assertions make the test a mixed functional and performance regression gate. A threshold pass with a missing lifecycle assertion is not a valid result.

Prepare a disposable PostgreSQL database

Use a database dedicated to this campaign. The test creates a unique UUIDv7 run identifier and prefixes its rows, then removes those rows after a successful result. If the test fails or is interrupted before the final cleanup call, its run-scoped rows may remain.

For that reason:

- do not point the test at a production or shared development database;
- take the database baseline before unrelated jobs start;
- ensure the database user can install and use the registry schema;
- record PostgreSQL version, `max_connections`, `shared_buffers`, storage

class, host resources, and connection pool settings;

- discard or restore the dedicated database after a failed run instead of

guessing which partial rows are safe to delete.

Set a descriptive environment label. It must contain 1 to 128 ASCII letters, digits, `-`, `_`, `.`, or `/`. If omitted, the result says `local-unspecified`, which is too weak for comparison evidence.

Run the gate and retain its output

From the exact source root, use a shell with pipeline failure propagation:

```
BASH
export AIP_POSTGRES_SCALE_TEST_URL='postgresql://user:password@host/database'
export AIP_SCALE_ENVIRONMENT_LABEL='dedicated-linux-amd64-postgres-17.6'

set -o pipefail
cargo test -p aip-connector-registry-postgres --test scale \
  indexed_catalog_and_routing_remain_bounded_at_fleet_scale \
  -- --nocapture 2>&1 | tee .aipd-scale-qualification.log
```

Use a secret-file or CI secret mechanism for the real database URL. Do not commit the command history, environment, or log if it contains credentials.

Extract and validate the final result:

```
BASH
test "$(grep -c '^AIP_SCALE_RESULT=' .aipd-scale-qualification.log)" -eq 1

result=$(sed -n 's/^AIP_SCALE_RESULT=//p' \
  .aipd-scale-qualification.log | tail -n 1)

test -n "$result"
printf '%s\n' "$result" | jq -e \
  '.slo.profile == "aip-fleet-slo-v1" and .slo.passed == true'
```

Reject the evidence if the result line is missing, duplicated, malformed, or does not match the expected source profile.

Require both a successful pipeline exit and the validated JSON. The source prints `AIP_SCALE_RESULT` before deleting its run-scoped rows, so cleanup can still fail after a JSON object says `passed: true`. The `pipefail` setting preserves that late test failure through `tee`.

Inspect the result fields

The final JSON records:

- environment label, operating system, architecture, PostgreSQL version and

selected settings;

- exact dataset counts;
- threshold values embedded by the source;
- catalog and routing p50, p95, and p99 samples;
- routing throughput, errors, eligible replicas, and action count;
- version rollout, tenant isolation, retry isolation, credential rotation, and

lease-churn outcomes;

- observed process RSS, database sizes and growth, available parallelism, and

connection pool snapshot.

Retain the full combined log, parsed JSON, source commit, dependency lockfile, database image or package identity, database configuration, host inventory, start and finish times, and cryptographic digests for the evidence files.

The log also contains the routing query plan. Review it even when the summary passes, because it explains which index path produced the measured result.

Compare runs honestly

Compare two results only when their source, dataset, PostgreSQL major version, database settings, host resources, architecture, storage class, and contention are equivalent. Otherwise describe the difference as an environment observation rather than a regression.

Use repeated runs and distribution-aware production telemetry for capacity planning. One 32-sample catalog percentile and one 500-route burst are useful CI guards, not a full latency study.

The reviewed GitHub and Gitea CI definitions use PostgreSQL 17.6, provide an explicit environment label, retain the combined log, and declare a 30-day artifact lifetime. Copy evidence required for a release decision before it expires.

State the result narrowly

A valid claim is:

The identified source passed `aip-fleet-slo-v1` with the recorded dataset and PostgreSQL environment at the stated time.

The result does not qualify connector processes, external products, network transports, multi-host behavior, application execution, arbitrary fleet sizes, or production capacity. Combine it with the [connector fleet failure matrix](#) and deployment-specific load, recovery, and observability campaigns.

Related documentation

- [Testing and evidence index](#)
- [Registry and routing architecture](#)
- [Metrics reference](#)
- [Connector fleet qualification](#)
- [Conformance and qualification](#)