Systems Integration and Testing Runbook

Nimbus Payment Gateway → Nexa Core Platform

Pilot deployment for Meridian Retail Group, executed jointly by Nexa Platform Engineering and Synapse Quality Engineering, on infrastructure hosted by Vertex Cloud and observed by Apex Monitoring.

Document ID: RB-2214-v3
Classification: Internal
Version: 3.0
Effective: 2026-07-14
Owner: D. Alcaraz

1 Introduction

1.1 Purpose

This runbook specifies the exact, repeatable sequence of steps required to integrate the Nimbus Payment Gateway (adapter release `v3.4.0`) into the Nexa Core Platform, and to validate that the integration behaves correctly before, during, and after the production cutover. It is written for the engineer holding the pager during the change window, not as a design reference: every command below is copy-pasteable and every validation step has an explicit pass/fail condition. Architectural rationale lives in the companion *Microservices Integration Interface Design Document*; this document assumes that design is final and focuses on execution.

1.2 Scope

- **In scope:** deployment of the Nimbus adapter service into the Vertex Cloud production cluster, schema changes required by the adapter, smoke and load validation, monitoring hookup in Apex, and rollback procedures.
- **Out of scope:** Nimbus's internal settlement reconciliation jobs (covered by Nimbus's own operations runbook), and the Meridian storefront checkout UI changes (tracked separately by the Meridian integration team).

1.3 Audience

IT operations and platform engineers executing or approving the change window; Synapse QA engineers running validation; on-call responders who may need to execute rollback without having read the design document first.

1.4 Revision History

Version	Date	Author	Summary of Changes
1.0	2026-04-02	D. Alcaraz (Nexa)	Initial draft, staging-only procedure.
2.0	2026-05-19	R. Okafor (Synapse QA)	Added load-test validation matrix and rollback triggers.
2.1	2026-06-11	D. Alcaraz (Nexa)	Corrected synctex-unrelated adapter port from 8443 to 8843.
3.0	2026-07-14	M. Sundberg (Vertex Cloud)	Production cutover sequence, blue/green promotion steps, Apex dashboard links.

Table 1: Document control log.

1.5 Related Documents

- *Microservices Integration Interface Design Document*, RB-2201, Nexa Platform Engineering.
- *Vertex Cloud Production Access and Change Management Policy*, VC-POL-009.
- *Apex Monitoring Alert Runbook: Payment Path*, APX-RB-114.

2 System Architecture Overview

2.1 Component Inventory

Table 2 lists every component touched by this change, its owning team, and the environment it runs in. All version numbers are the ones expected to be live immediately after cutover.

Component	Owner	Environment	Version
Nexa API Gateway	Nexa Platform Engineering	Vertex Cloud, `ap-south-1`	5.12.0
Nimbus Payment Adapter	Nimbus Integrations Team	Vertex Cloud, `ap-south-1`	3.4.0
Ledger Postgres Cluster	Vertex Cloud SRE	Vertex Cloud, `ap-south-1`	15.6
Synapse QA Harness	Synapse Quality Engineering	CI runner, ephemeral	2.9.1
Apex Monitoring Agent	Apex Observability	sidecar, all pods	1.7.3

Table 2: Components in scope for this integration.

2.2 Data and Control Flow

Figure 1 shows the request path exercised during both normal operation and the validation phase described in Section 4. Meridian's storefront calls the Nexa API Gateway, which routes payment-intent requests to the Nimbus adapter over an internal mTLS link on port `8843`. The adapter writes settlement records to the Vertex Cloud ledger cluster and returns a confirmation token to the gateway. Apex's monitoring agent samples latency and error rate at both the gateway and the adapter; Synapse's QA harness exercises the same path from outside the cluster during validation windows.

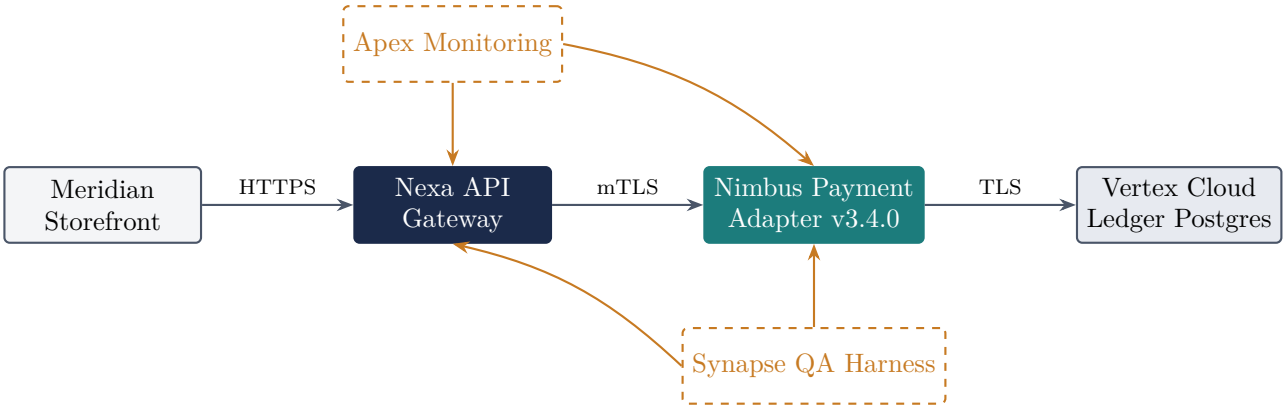

Figure 1: Request path and observation points during integration testing. Solid arrows show production traffic; dashed nodes and amber arrows show the monitoring and validation planes.

3 Deployment Sequence

3.1 Pre-Deployment Checklist

Confirm every item below before opening the change window. Do not proceed if any item is unchecked; escalate to the contacts in Section 6.

- ✔ Change window approved in Vertex Cloud change calendar (minimum 2 hours).
- ✔ Ledger Postgres cluster snapshot completed within the last 30 minutes.
- ✔ Nimbus adapter image `v3.4.0` present in the Vertex Cloud registry and signature-verified.
- ✔ Synapse QA harness green on the staging replica within the last 24 hours.
- ✔ Apex on-call engineer notified and dashboards open.

3.2 Step 1 – Freeze and Snapshot

Announce the freeze in the `#nexa-change-window` channel, then take a fresh, named snapshot of the ledger cluster so the rollback target in Section 5 is unambiguous.

Vertex Cloud CLI

```
$ vertex db snapshot create \
   --cluster ledger-prod-aps1 \
   --label pre-nimbus-3.4.0-cutover
```

3.3 Step 2 – Deploy Adapter to Staging Slot

Deploy the new adapter into the idle (green) slot of the blue/green pair. Traffic remains on the blue slot throughout this step.

Vertex Cloud CLI

```
$ vertex deploy apply \
   --service nimbus-payment-adapter \
   --slot green --image nimbus-adapter:3.4.0 \
   --env-from nimbus-adapter-prod.env
```

3.4 Step 3 – Run Smoke Tests Against the Green Slot

Point the Synapse harness at the green slot directly (bypassing the gateway) to confirm the adapter starts cleanly and can reach the ledger cluster.

Synapse QA Harness

```
$ synapse-qa run smoke-suite \
   --target https://nimbus-adapter-green.internal:8843 \
   --suite payment-intent-basic
```

The suite must report `0 failures`. If any test fails, do not proceed to Step 4; tear down the green slot and investigate.

3.5 Step 4 – Promote Green to Live

Once smoke tests pass, shift the Nexa API Gateway's upstream pool from blue to green. This is the only step that affects live Meridian traffic.

Vertex Cloud CLI

```
$ vertex gateway promote \
   --service nimbus-payment-adapter \
   --to green --drain-timeout 30s
```

3.6 Step 5 – Verify Apex Dashboards

Watch the *Payment Path* dashboard in Apex for five minutes. Error rate must remain below 0.5% and p95 latency must stay under 400 ms (see Section 4 for the full acceptance matrix).

3.7 Step 6 – Retire the Blue Slot

Once the gateway has run cleanly against green for at least fifteen minutes, decommission the old blue slot to free capacity.

Vertex Cloud CLI

```
$ vertex deploy retire \
   --service nimbus-payment-adapter --slot blue
```

Note

Keep the blue slot's image tag recorded even after retirement – Section 5 redeploys from it rather than rebuilding, so the rollback path does not depend on the registry still holding old build artifacts beyond the standard 30-day retention.

4 Validation and Testing Procedures

4.1 Test Case Matrix

Table 3 is the acceptance matrix Synapse QA runs at each stage of the sequence in Section 3. All four rows must show **Pass** before the change is considered complete; a **Fail** in any row triggers the rollback procedure in Section 5.

Test ID	Description	Method	Result
INT-041	Payment intent created and confirmed end-to-end through the gateway.	Synapse smoke suite	Pass
INT-042	Adapter correctly rejects a malformed currency code (expects HTTP 422).	Synapse smoke suite	Pass
INT-043	Ledger row written exactly once per confirmed intent (no duplicate writes under retry).	SQL validation query	Pass
INT-044	p95 latency stays under 400 ms at 250 requests/sec sustained for 10 minutes.	Load test, k6 via Synapse harness	Pass

Table 3: Acceptance test matrix for the Nimbus adapter integration.

4.2 Validation Query: Ledger Write Consistency

INT-043 is checked directly against the ledger cluster rather than through the API, so that a bug in the adapter's own reporting cannot mask a duplicate write. Run this after the smoke suite in Step 3 and again after promotion in Step 4.

Ledger Postgres

```
$ psql -h ledger-prod-aps1.vertex.internal -U ro_validator -d ledger -c \
   "SELECT intent_id, COUNT(*)
   FROM settlement_records
   WHERE created_at > now() - interval '1 hour'
   GROUP BY intent_id HAVING COUNT(*) > 1;"
```

The query must return zero rows. A nonzero result means the adapter is double-writing on retry and is a hard rollback trigger regardless of what the latency dashboard shows.

4.3 Latency Across Test Phases

Figure 2 plots p50 and p95 latency recorded by the Synapse harness at each test phase during the 2026-07-14 cutover rehearsal. Latency rises under load, as expected, but stays within the 400 ms p95 budget through the soak phase.

5 Rollback Procedures

Roll back immediately if any of the following occur

- Test INT-043 (ledger write consistency) returns any duplicate rows.
- p95 latency exceeds 400 ms for more than two consecutive minutes after promotion.
- Gateway error rate exceeds 1% for more than one minute after promotion.

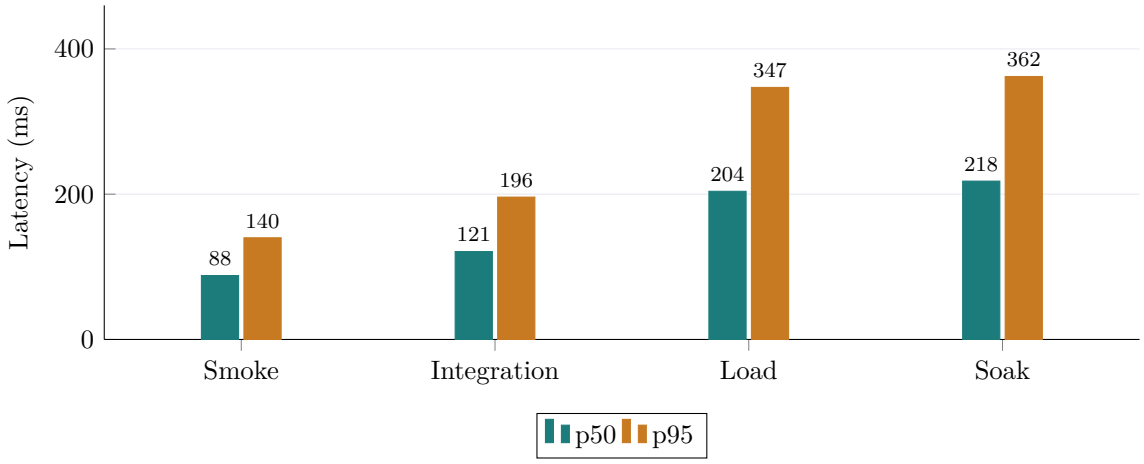

Figure 2: p50/p95 latency by test phase, 2026-07-14 cutover rehearsal. The 400 ms p95 budget from test INT-044 held through the soak phase.

- Apex reports the adapter's mTLS handshake failing against the ledger cluster.

5.1 Step 1 – Revert Gateway Upstream

If the blue slot has not yet been retired (Step 6 of Section 3), this alone restores service: point the gateway back at blue.

Vertex Cloud CLI

```
$ vertex gateway promote \
   --service nimbus-payment-adapter \
   --to blue --drain-timeout 10s
```

Confirm recovery on the Apex *Payment Path* dashboard within 60 seconds. If error rate and latency return to baseline, the rollback is complete; skip to Step 4.

5.2 Step 2 – Redeploy Blue if Already Retired

If blue was already retired, redeploy it from the recorded pre-cutover image tag rather than rebuilding.

Vertex Cloud CLI

```
$ vertex deploy apply \
   --service nimbus-payment-adapter \
   --slot blue --image nimbus-adapter:3.3.6 \
   --env-from nimbus-adapter-prod.env
```

Then repeat Step 1 above to shift traffic back to blue.

5.3 Step 3 – Restore the Ledger Snapshot (Duplicate-Write Cases Only)

Only required if INT-043 flagged duplicate settlement rows. Restore from the snapshot taken in Step 1 of Section 3; this is destructive to any rows written since the snapshot, so confirm with the on-call lead before running it.

Vertex Cloud CLI

```
$ vertex db snapshot restore \
   --cluster ledger-prod-aps1 \
   --label pre-nimbus-3.4.0-cutover --confirm
```

5.4 Step 4 – Post-Rollback Validation

Re-run the smoke suite (INT-041, INT-042) against the restored blue slot and confirm the ledger consistency query from Section 4 returns zero rows. Post the outcome in `#nexa-change-window` and

file an incident record per *Apex Monitoring Alert Runbook: Payment Path* (APX-RB-114) before closing the change window.

6 Appendix

6.1 Escalation Contacts

Organization	Name	Role	Contact
Nexa Platform Engineering	D. Alcaraz	Change owner, on-call primary	+1-555-0142
Synapse Quality Engineering	R. Okafor	QA lead, validation sign-off	r.okafor@synapse-qe.example
Vertex Cloud SRE	M. Sundberg	Infrastructure escalation	+1-555-0198
Apex Observability	T. Nkemelu	Monitoring and alerting on-call	t.nkemelu@apexobs.example
Meridian Retail Group	S. Whitfield	Client integration contact	+1-555-0177

Table 4: Escalation contacts for the Nimbus Gateway integration change window.

6.2 Acceptance Sign-Off

This runbook is considered successfully executed when all rows in Table 3 read **Pass**, the Apex *Payment Path* dashboard has shown a stable baseline for at least fifteen minutes post-promotion, and the change owner has posted a closure note in `#nexa-change-window` referencing this document's ID (RB-2214-v3).

6.3 References

Background standards informing the validation methodology used in this runbook: [3], [1], [2].

References

[1] AXELOS Global Best Practice. *ITIL Foundation: ITIL 4 Edition*. TSO (The Stationery Office), London, UK, 2019.

[2] IEEE Computer Society. IEEE standard for software and system test documentation. Technical Report 829-2008, Institute of Electrical and Electronics Engineers, 2008.

[3] International Software Testing Qualifications Board (ISTQB). Certified tester foundation level syllabus. Technical Report v2018, International Software Testing Qualifications Board (ISTQB), 2018. Defines the integration testing and test-level terminology used throughout this runbook.