



DESIGN-PARTNER EVALUATION · PROPOSED

Install the evaluation path

Installation guide · Disposable first, customer-controlled next

Use the exact source revision and `docs/runtime/QUICKSTART.md` supplied with your evaluation checkout. That file is the canonical command sequence. The kit does not contain credentials or a separately published Runtime SDK. Runtime code remains private; evaluation access and permitted use are established by the written agreement.

Local prerequisites

Node 22+, npm, running Docker, kind and kubectl; network access to obtain dependencies and pinned sample images. No production cluster, cloud account or LLM key is needed. The helper uses an isolated kubeconfig and named disposable kind cluster, not your current production context.

Run from the repository root. Start with the clean-checkout commands:

```
npm ci
npm run build
```

Continue with the quickstart’s **design-partner evaluation section**: isolated Docker database, kind cluster, local migration URL, generated fixture identities and protected environment file. Docker Compose is not required. Supply the migration credential only to setup, then unset it; the Runtime uses its generated restricted login. Setup changes disposable resources; never run it against production.

After setup, run the exact acceptance command:

```
export NIKXIUS_RUNTIME_DEMO_DIR=.tmp/design-partner-eval
node --env-file=.tmp/design-partner-eval/runtime.env \
  scripts/design-partner-acceptance.mjs \
  .tmp/design-partner-eval/runtime-config.json
```

Complete the customer path

Step	Observable result
Configure/start	Strict configuration accepted; <code>/health</code> responds and <code>/ready</code> reaches the database. Neither checks all external dependencies.
Enroll target	Configuration plus restart, not an enrollment API. The runner generates the exact rollback relationship after the earlier commit.

Step	Observable result
Establish earlier commit	A separately authorized image operation supplies the committed earlier-operation record required for rollback.
Grant/propose/review	Sponsor issues exact rollback grant; workload submits stable key; reviewer inspects current command hashes/version.
Execute/inspect	Operator reads lifecycle, commit and observation separately.
Fault/reconcile	Response loss becomes UNKNOWN; read-only reconciliation uses native evidence or retains uncertainty.
Export	Operation JSON plus optional signature; verify against the separately supplied public key.
Stop/uninstall	Stop the Node and disposable database; retain evidence before explicitly deleting named fixtures.

For operator inspection, serve `runtime-rollback-config.json` as shown in the quickstart and open <http://127.0.0.1:8789/operations>. The fixture's historical payments and `production-us-east` labels are synthetic; no production resources are involved. Native conformance suites reset shared fixtures, so run them before final acceptance.

The older public demo uses `set_image`; its fixture reset is not a Runtime rollback. Keep the new run record with its source revision. The sanitized internal verification summary is `artifacts/design-partner-2026-09-15/VALIDATION.json`; generated credentials remain private.

Customer nonproduction installation

Follow `docs/runtime/DEPLOYMENT.md` for image build, `migration/init-tenants`, registered IdP subjects/audiences, HTTPS origin, mounted rotating credential, reviewed RBAC and Node configuration. The example manifests do not provision IdP, PostgreSQL, ingress, backups or a registry. Customer identity token acquisition and native controller/admission ownership need explicit review.

The **15 September clean-source internal run** reached acceptance in **20 minutes 25 seconds**, including native tests and interleaved author work. Tools were already installed and download caches were warm. The acceptance runner itself took **2 minutes 16 seconds**. These are elapsed internal observations, not hands-on customer installation time. Record prerequisites, downloads and assistance in your run. **Under 60 minutes remains an unvalidated customer target**. See the included evaluation-validation summary for exact timings and limits.

If any step fails, preserve the exact command/version and redacted error. Do not bypass identity, TLS, least privilege or persistence to finish a demonstration. For UNKNOWN, use `docs/runtime/RECOVERY.md`; never mint a new key simply to retry a possibly executed mutation.