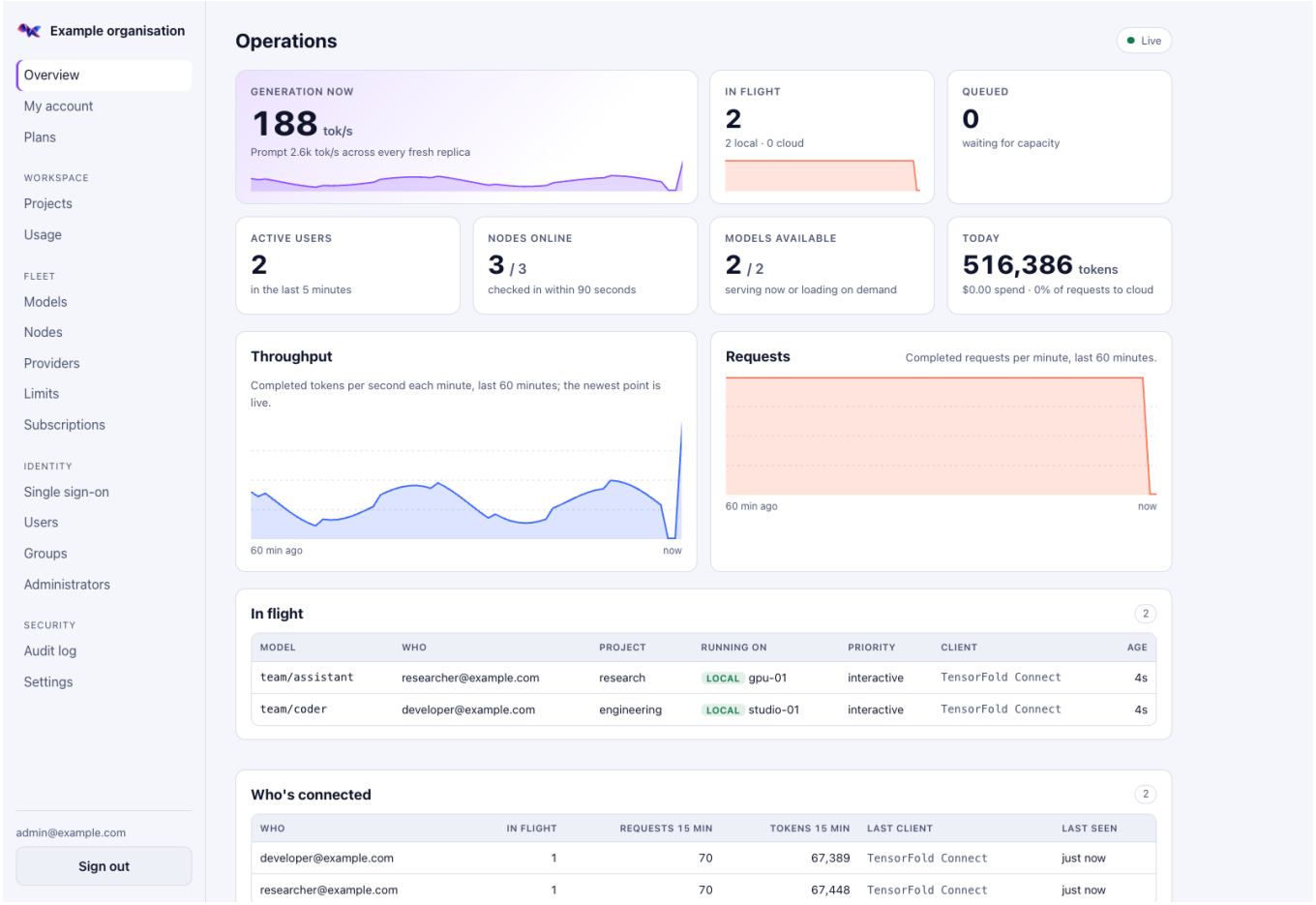
FREE TO DEPLOY. SUPPORT THROUGH BACKING.

StarLight Enterprise evaluation

A practical starting point for platform, security and operations teams evaluating private AI infrastructure.

Choose one workload. Review the controls. Measure the deployment you intend to operate.



Actual StarLight console captured on 2 November 2026. Organisation, hardware and usage data are illustrative; figures are not benchmark evidence.

The whole project is free and remains open source under Apache-2.0. Every improvement we make, including work with enterprise users, is published on GitHub. Self-managed deployments do not include direct support. Agreed donations, sponsorship or sustained backing fund deployment assistance and ongoing support.

Reviewed 5 November 2026 against source 885b4967063d. Online evaluation pack | Pilot guide and cost planner

01 / DEPLOYMENT

Run a focused pilot.

Start with a named owner, one application and representative, approved test data. Record your model, request sizes, expected peak concurrency and data classification. Select a checkpoint TensorFold qualifies on your hardware.

Requirement	Prepare before starting
Control plane	A host with Git, Docker and Docker Compose; Postgres 15 or newer is deployed by the evaluation configuration.
Inference	At least one serving node, the engine pinned by engine.json and your chosen qualified model.
Identity and network	An identity provider or the documented development IdP trial; public/node origins, TLS and network paths appropriate to the environment.

Build the reviewed source snapshot

```
git clone https://github.com/ashhart/tf-starlight.git
cd tf-starlight
git checkout 885b4967063ddabe692126f5ba3c9bdf2f6134d5
cd deploy/compose
./bootstrap.sh
docker compose build
docker compose run --rm starlight init --org "Example Corp"
docker compose up -d
```

Open the one-time setup link, configure sign-in and complete first-owner setup. Provision the compatible engine on a node, create an enrolment token, run the printed command on that node, then add a model and deployment in the console.

For remote nodes and clients, configure the correct origins and TLS first. Treat setup links and enrolment tokens as credentials. Use Connect or an API key with a compatible application.

Quick start | Installation and listeners | Node setup | Engine compatibility pin

This is a pinned source-build evaluation path. Check the repository releases for packaged builds. The setup is a starting point for a pilot, not a statement that your production environment has been qualified.

02 / SECURITY REVIEW

Follow identities and data.

People and applications reach StarLight using company identity, Connect or scoped API credentials. StarLight checks the project, model grant, data classification and budget before routing a request to an eligible local replica or approved provider.

Boundary	Documented controls	Review in your environment
People and clients	OIDC/SAML, SCIM provisioning, roles and scoped credentials.	Real tenant, group mappings, logout/deactivation and managed client rollout.
Requests and routes	Project model grants, classification, budgets, credential guard, local/provider routing and offline mode.	Allowed data routes, sensitive test cases, cache isolation policy and offline provisioning.
Gateway and nodes	Mutual TLS, pinned internal CA and request-bound signed permits.	Network paths, certificate custody, enrolment, revocation and service accounts.
Secrets and records	Sealed credentials, master key and a verifiable audit-event chain with export.	Database access, key custody, restore, retention and SIEM ingestion.

Optional external routes

Provider access is policy-controlled. Offline mode disables provider routes; the organisation must provision dependencies and model artifacts for that environment. Review the exact data permitted to leave each boundary.

> These are documented implementation controls, not a compliance certification. Automated vulnerability and code scans do not replace an independent security review. Complete the applicable production qualification and security checks.

Security model and residual risks | Vulnerability reporting | Identity-provider guides

Review the software supply chain

Review the StarLight licence, third-party notices, pinned engine and a dependency inventory for the exact version. Model and external-provider licences are separate. Include available release provenance in your approval process.

Apache-2.0 licence | Third-party notices | Release verification

03 / OPERATIONS AND BACKING

Agree the operating responsibilities.

Postgres stores state and is part of service availability. StarLight instances may sit behind a load balancer; serving nodes and Connect have their own versions, credentials and lifecycle.

Area	Organisation operating plan
Recovery	Back up Postgres and the master key. Restore into a separate environment and verify the doctor and audit checks. Record recovery time and data loss.
Availability	Exercise controller, gateway and node loss and database failover under the intended workload. Define the error budget and alert owners.
Changes	Pin versions and review migrations and release notes. Test upgrades and rollback; database rollback can require restoring a backup.
Monitoring	Collect metrics, set alerts, restrict metrics/admin access, export audit records and define retention.
Ownership	Name owners for identity, hosting, model artifacts, data policy, keys, backups, upgrades and incidents.

Free to deploy. Support through backing.

Self-managed deployments are free and do not include direct support. Organisations fund deployment assistance and ongoing support through agreed donations, sponsorship or sustained project backing.

Before relying on support, agree the scope, supported versions, duration, named contacts and escalation arrangements. All improvements remain open source and are published on GitHub.

Operations runbook | Metrics reference | Discuss backing and support

04 / RECORDED EVIDENCE

Read the scope with the result.

The following are project-reported checks recorded in public documentation at the reviewed source revision. No new hardware or inference benchmark was performed for this pack.

8,000-user control-plane test

Synthetic engine

8,000 distinct API users, one request each, with 512 concurrent workers and a simulated engine answering in 15 ms. The recorded one-gateway run achieved 3,180-3,335 requests/s.

Exercises identity, admission, routing and accounting. This is not model throughput or 8,000 simultaneous inference requests.

A real TensorFold engine

One recorded configuration

Apple M3 Ultra, TensorFold 1.2.0 and Qwen3.5-2B through an enrolled node. The console reported 230-408 generated tokens/s with four requests running.

A real-engine smoke check for that setup. It does not qualify a different model, GPU, cluster or workload.

Compose quick start

Documented CI check

The README commands were exercised on Linux with Docker 29.4.2, Compose 5.1.3, Postgres 17.11 and TensorFold 1.2.1. Checks include health, setup, authentication, doctor and audit verification.

Single-host evaluation path. Production identity, GPU and recovery checks remain deployment-specific.

Release and security checks

Documented automated checks

A 5 November release dry run built 15 binary archives, deployment assets and a chart. The qualification record also documents Go vulnerability scanning and CodeQL findings and fixes.

The dry run signed and published nothing. Automated checks do not replace an independent security review.

> The synthetic 8,000-user profile exercises control-plane overhead with a simulated engine. It is not a measurement of model serving capacity or 8,000 simultaneous inference requests. The real-engine check is one recorded model/hardware configuration.

Qualification record at 885b4967063d | Online evidence and current-source links

05 / VALIDATION AND ACCEPTANCE

Close the gates for your rollout.

Use your real identity tenant, serving hardware and workload. Record the date, software versions, configuration, test method, raw results and limitations when an applicable gate passes.

Open gate	Evidence to record
Your serving hardware	Validate the actual GPU/backend, node service, model, rank group and restart behaviour you plan to use.
Company identity and rollout	Exercise your real identity tenant, group mapping, deactivation and managed desktop configuration.
Real workload and scale	Measure queue wait, time to first token, completed requests, errors and resource use on real inference nodes at the planned peak.
Availability and recovery	Exercise controller, gateway and node loss; database failover; and a restore with the master key. Record recovery times and data loss.
Independent security review	Complete a security review and penetration test of the gateway, identity, node and console before production use.

Pilot acceptance record

Measure	Agreed target	Measured result
Task quality and correct tool use	_____	_____
Peak concurrency and completed requests	_____	_____
p95 queue wait / first token / total time	_____	_____
Errors and policy enforcement	_____	_____
Recovery time and lost work	_____	_____

Technical owner: _____ Security reviewer: _____

Decision / conditions for expansion: _____

Full open-gate checklist | Pilot measurement guide

06 / WORKLOAD ECONOMICS AND DESIGN PARTNERS

Plan costs from your own inputs.

Use equivalent workloads, quality requirements and accounting boundaries when comparing options. The calculator does not predict model speed, capacity or a saving from unmeasured workloads.

Input	Your value in one currency
Hardware cost / allocation life in months	_____ / _____
Measured average fleet kW / monthly hours	_____ / _____
Electricity price per kWh	_____
Monthly hosting, database and storage	_____
Monthly operations and staff time	_____
Monthly agreed backing and external services	_____ / _____
Successful requests per month	_____
Equivalent current monthly cost, optional	_____

Monthly total = hardware cost / life in months + fleet kW x running hours x cost per kWh + hosting + operations + project backing + external services. Cost per 1,000 successful requests = monthly total / successful requests x 1,000.

Illustrative arithmetic only: 12,000 of hardware over 36 months; 0.8 kW for 720 hours at 0.25/kWh; hosting 80, operations 600, backing 250 and external services 100 gives 1,507.33 per month. At 50,000 successful requests, that is 30.15 per 1,000. These are example inputs, not StarLight measurements or support prices.

Help validate a real enterprise workload

Design partners bring a named technical owner, a bounded workload and an environment they can test. Share feedback on deployment, identity, quality and operation. Improvements developed together are published on GitHub. Public names, results and case studies require agreement.

Self-managed pilots are free. Direct assistance is arranged through project backing.

Interactive cost planner | Discuss a design partnership

enquiries@tensorfold.dev