

Ship & operate

Environments

🕒 4 min read · Updated 13 October 2026 · Verified against [a661527](#) on 2026-09-26

Local, staging and production side by side: hosts, API platforms, data, protection and who deploys what.

<https://docs.gurbanisoul.com/process/environments/>

	Local	Staging	Production
Branch	working tree	integration	main
Web	<code>serve.py</code> on:7777	<code>sggs-staging.vercel.app</code> (SSO-protected; open logged in)	gurbanisoul.com (canonical)
API	same process	Vercel functions in the web project, one per context + <code>all</code> (ADR-0011)	Vercel functions in the web project: every <code>/api</code> path is answered by <code>all</code> (v1.3.10, and 1.4.0 keeps it so); each context moves to its own function, as on staging, in a later release after the app is live (Render <code>sggs-knowledge-base.onrender.com</code> kept as rollback)
iOS	simulator	TestFlight Internal	TestFlight External / App Store
DB profile	full	full	full (public profile until the English licence is recorded)
Who deploys	you	<code>deploy-staging.yml</code> on push to integration	<code>deploy-production.yml</code> on push to main

Production domain. The canonical public host is **gurbanisoul.com** (apex, 200, no redirect). `www.gurbanisoul.com` and the legacy Vercel alias `sggs-knowledge-base.vercel.app` both **308** → apex, so there is one indexable host. DNS is Cloudflare DNS-only → Vercel; full topology and the email routing are in [docs/website/README.md](#).

Same-origin API (no CORS)

The browser always calls `/api/*` on the site's own host, so no frontend configuration or CORS is needed. On staging the rewrites are internal and **host-conditioned**: each context's prefixes go to its function, everything else to `all`. Production's rules come after them, internal too and host-less, so they catch every other host, including an unaliased deployment URL. `production.services` is empty: production (v1.3.10, and 1.4.0 after it) sends every `/api` path to `all`. Listing the five contexts there, in a later release after the app is live, sends each context's prefixes to its own function in production too, and everything else (and `/readyz`, `/healthz`) to `all`. Until then every path goes to `all`, the single API's role on the same database and code. The rules are generated (`tools/gen_gateway.py` from `gateway/routes.json`), for example:

```
{ "source": "/api/timing(/.*)?",
  "has": [{ "type": "host", "value": "sggs-staging.vercel.app" }],
  "destination": "/api/svc/knowledge" },
{ "source": "/api/timing(/.*)?",
  "destination": "/api/svc/knowledge" },
{ "source": "/api/(.*)",
  "destination": "/api/svc/all" }
```

Splitting production's contexts out — all five in one release, after the app is live — and retiring Render afterwards are [runbook: services-production](#).

As-code

- **Deploys are CI-gated** — see [runbook: deploy](#). The platforms' own git auto-deploys are switched off after cutover.
- Render: sggs-knowledge-base (from webapp/Dockerfile) is a manually managed service; no Blueprint is committed. The site's /api no longer reaches it, but deploy-production.yml keeps deploying it every release as the API rollback target until it is retired. Staging no longer uses Render (ADR-0011).
- Vercel: frontend/vercel.json (root dir = frontend/) — production branch main, staging a custom environment tracking integration.
- deploy-verify CI curls /api/health and asserts /api/meta.version matches the branch's APP_VERSION after each deploy (catches a stale API).

Environment variables (Vercel)

Var	Scope	Purpose
PUBLIC_NEWSLETTER_FORM_URL	Production + Preview (unset in Development/CI/local)	Buttondown embed form endpoint for the landing's launch-notice sign-up. Unset ⇒ the section, form and any Buttondown reference are absent from the built HTML. Public POST endpoint, not a secret, but must never be committed (NewsletterPrivacy gate). See <u>runbook: newsletter</u> .

The App Store Connect API-key secrets used by TestFlight live in **GitHub**

Environments (production/staging), not Vercel — never echo a value; a pasted secret is a leaked secret.

Rollback

Web: Vercel Instant Rollback (automatic when the public smoke fails after promote).

API: the functions ship inside the web deployment, so a web rollback restores the previous deployment's functions and rewrites with it. To move the whole API back to Render, set production.api_platform back to render with its api origin in gateway/routes.json, regenerate and release; Render keeps deploying every release

until it is retired, so this stays available until then (runbook: services-production, step 4). Data: revert the dataset .lock.json bump — every pinned object stays available in sggs-data. See runbook: rollback.

Human-gated setup (one time)

Authorize the Vercel and Render GitHub Apps on the org, create the Vercel staging custom environment + alias, add the deploy secrets to the GitHub Environments (runbook: deploy). The App Store Connect API-key secrets live in the app repository's environments.

Sri Guru Granth Sahib Ji — KnowledgeBase wiki. Generated from commit c3d9360. Updated 13 October 2026. Verified against a661527 on 2026-09-26.
This copy may be out of date; the page is at <https://docs.gurbanisoul.com/process/environments/>