

One Minute Manager

Operations Manual — oneminutemanager.ai

This manual covers day-to-day operation, monitoring, and maintenance of the One Minute Manager marketing site, including lead capture, transactional email, custom domain, and security.

1. System Overview

Stack

Layer	Technology
Framework	TanStack Start (React 19, Vite 7)
Hosting	Lovable (Cloudflare Worker runtime)
Database / Auth	Lovable Cloud (Supabase)
Email queue	pgmq + scheduled processor route
Custom domain	oneminutemanager.ai (+ www)

Key URLs

- Production: <https://oneminutemanager.ai> (and www.)
- Preview: project--18788426-1fa0-4f84-b720-53bc13e91fbe-dev.lovable.app
- Backend admin: Project Settings → View Backend

2. Site Map & Routes

Path	File	Purpose
/	src/routes/index.tsx	Marketing landing page (hero, modules, pricing, CTAs)
/privacy	src/routes/privacy.tsx	Privacy policy
/terms	src/routes/terms.tsx	Terms of service
/email/unsubscribe	src/routes/email/unsubscribe.ts	One-click unsubscribe handler
/lovable/email/transactional/send	.../send.ts	Internal send endpoint (Zod-validated)
/lovable/email/transactional/preview	.../preview.ts	Template preview
/lovable/email/queue/process	.../process.ts	Worker that drains pgmq queue
/lovable/email/suppression	.../suppression.ts	Suppression list management

3. Lead Capture Flow

How a submission travels end-to-end.

- User opens any **LeadDialog** (nav, hero, pricing, footer CTAs) and submits the form.
- Frontend calls the **submitLead** server function (*src/lib/leads.functions.ts*).
- Zod validates inputs; row inserted into **public.leads** via the service-role admin client.
- An email is enqueued to **pgmq.transactional_emails** using the *lead-notification* template.
- The queue processor route picks it up and delivers via the configured email provider.
- An unsubscribe token is upserted in **email_unsubscribe_tokens** for the recipient.
- Send is recorded in **email_send_log** (pending → sent / failed).

Verifying a submission

- Submit a test lead from the preview site.
- Backend → **leads**: confirm new row.
- Backend → **email_send_log**: confirm *sent* status for *lead-notification*.
- Check the inbox configured by **LEAD_NOTIFICATION_EMAIL** (currently **Leads@oneminutemanager.ai**).

4. Secrets & Configuration

Secret	Purpose
LEAD_NOTIFICATION_EMAIL	Inbox that receives every new lead alert
SUPABASE_SERVICE_ROLE_KEY	Server-only admin client (never expose)
SUPABASE_URL / PUBLISHABLE_KEY	Public client config (safe in bundle)

Update a secret: *Project Settings* → *Secrets* → *edit* → *Save*. Server functions read it on the next call (`process.env.NAME`).

5. Transactional Email

Templates

- Templates live in `src/lib/email-templates/` and are registered in `registry.ts`.
- Each template must expose a Zod **schema**; the `/send` route rejects payloads that fail validation.
- Current template: **lead-notification** (fields: name, email, company, role, message, source, submittedAt).

Queue & delivery

- Mail is enqueued to `pgmq.transactional_emails` via `enqueue_email` (SECURITY DEFINER, service-role only).
- The processor route reads batches, sends, and logs the outcome in `email_send_log`.
- Failures move to a DLQ via `move_to_dlq`. Inspect failed rows in the backend to retry or debug.
- Sender domain: **notify.oneminutemanager.ai**. DNS records must remain verified for delivery.

Adding a new template

- Create `src/lib/email-templates/<name>.tsx` exporting **component**, **subject**, and **schema**.
- Register it in `registry.ts`.
- Preview at `/lovable/email/transactional/preview?template=<name>`.
- Send via the `/send` route with a JSON body matching the schema.

6. Custom Domain

- Apex: **oneminutemanager.ai**; subdomain: **www.oneminutemanager.ai**.
- DNS records required at the registrar:

```
A @ → 185.158.133.1
A www → 185.158.133.1
TXT _lovable → (verification value from Project Settings)
```

- Manage in Project Settings → Domains. Re-verify if records change.
- Sender domain DNS (SPF/DKIM/DMARC) is separate and managed under the email provider.

7. Security Posture

- All public tables have RLS enabled with explicit policies; **leads** is service-role only.
- Email queue wrapper functions are **SECURITY DEFINER** with **EXECUTE** restricted to *service_role*.
- The */send* route enforces per-template Zod validation to prevent injection.
- Service-role key is never imported from client code (file: *client.server.ts*).
- Run a fresh scan: Project Settings → Security → Run scan. Fix findings before publishing.

8. Operational Runbooks

Lead alerts not arriving

- Confirm **LEAD_NOTIFICATION_EMAIL** is set and spelled correctly.
- Check **email_send_log** for the most recent *lead-notification* rows.
- If status is *failed*, inspect **error_message** and the pgmq DLQ.
- Verify sender DNS (SPF/DKIM/DMARC) for **notify.oneminutemanager.ai**.
- Check the recipient's spam folder; whitelist the sender address.

Form submits but no row in leads

- Open browser devtools → Network → inspect the *submitLead* request and response.
- Check Lovable Cloud function logs for *submitLead insert error*.
- Validate RLS policies on **public.leads** still allow service-role inserts.

Site won't load on custom domain

- Confirm A and TXT records match the values in Project Settings → Domains.

- Wait for DNS propagation (up to a few hours).
- Re-publish the project after any router or root-route change.

Rotating a secret

- Update the value in Project Settings → Secrets.
- No redeploy needed; the next server-function invocation picks it up.
- For SUPABASE_SERVICE_ROLE_KEY rotation, use the backend admin UI and update the secret in the same window.

9. Editing Site Content

- Headlines, modules, pricing tiers, and CTAs all live in **src/routes/index.tsx**.
- Hero image: **src/assets/principal.jpg** (replace file or re-generate to update).
- Global metadata / title / OG tags: **src/routes/__root.tsx** (root-level) plus per-route head().
- Quick visual tweaks (text, colors, fonts) — use **Visual Edits** in the Lovable composer to avoid prompt cost.

Publishing

- Use the Publish action in Lovable to push the latest preview to **oneminutemanager.ai**.
- After publishing, smoke-test: load home page, open a LeadDialog, submit a test lead, confirm the alert email.

10. Quick Reference

Task	Where
Change notification inbox	Secrets → LEAD_NOTIFICATION_EMAIL
View leads	Backend → Tables → leads
Inspect email status	Backend → Tables → email_send_log
Retry failed mail	Backend → pgmq DLQ for transactional_emails
Update domain DNS	Registrar + Project Settings → Domains
Run security scan	Project Settings → Security
Publish updates	Lovable composer → Publish