GitHub, AWS, Database Ownership Checklist

BY JETTHOUGHTS TEAM - MAY 17, 2026 - 10 MIN READ

Template companion to Lesson 4.2 of the From Idea to First Paying Customer course. Where this fits: run after Lesson 4.1 routes you to a path that involves anyone else touching your infrastructure.

> ***Routing by M4.1 decision (read this first):***
>
> - ***Path 1 (Validate without code)***: skip this entirely - no infrastructure to audit yet.
> - ***Path 2 (Self-serve - Lovable + Supabase + Stripe)***: 5-minute confirmation, not a 45-minute audit. Your accounts will be in your own email automatically because you sign up yourself. Skim the 12 items below to confirm, save the checklist, and return in full the day you hire your first contractor or Fractional CTO.
> - ***Path 3 (Fractional CTO bridge)***: run the full 45-minute audit BEFORE you give the FCTO any credential.
> - ***Path 4 (Hire a team)***: run the full audit BEFORE the contractor's first commit. This is the safety net that prevents the credential trap in the next section.

A 45-minute audit that tells you whether you own your company's code, cloud, and domain - or someone else does. For Path 2 self-serve founders, this is a 5-minute confirmation; for Path 3 and Path 4 founders, the full 45-minute version.

By the end of one Friday (or the 5 minutes if you're self-serve) you will know whether you can fire your dev team on Monday and still ship a hotfix on Tuesday. You will not have read a line of code. You will have logged into 12 accounts and answered one question for each: is the email on this account mine, or is it theirs?

Copy-pasteable audit spreadsheet - copy these 12 rows into Notion or Google Sheets before you start:

#	Check	Pass?	Actual email/name on file	Recovery needed?
1	GitHub org owner is you	☐		
2	You can remove every contractor right now	☐		
3	Branch protection: you have admin override on `main`	☐		
4	AWS root account email is on your domain	☐		
5	Your card pays the AWS bill; you can see 12 months of invoices	☐		
6	IAM admin user is yours, MFA on, password in your vault	☐		
7	Production DB password: you can read it tonight without paging anyone	☐		
8	Secrets store admin is you; you can rotate every secret tomorrow	☐		
9	DB backups run nightly; you can restore one to staging by yourself	☐		
10	Domain registrar: WHOIS in your name, your email, your card	☐		
11	DNS provider: you can log in and add an A record right now	☐		
12	Third-party API keys (Stripe, SendGrid, OpenAI, etc.): accounts in your name	☐		

Scoring: All 12 ✓ = you own your company. Any ✕ = fix that row before contractor kickoff.

Why this exists

The account that locks you out is the one set up under someone else's email on Day 1 and never moved - and you find it a year later, once the person who opened it is gone. That is how it played out for a founder we saw last year: she handed a contractor her company credit card on Day 1, and he opened the accounts fast and shipped the MVP inside six weeks. Nobody moved the accounts to her email afterwards. A year later the contractor left the industry, and she discovered the GitHub org, the AWS root, and the domain were all registered to an inbox nobody checked. Recovery is slow and expensive when the accounts are not in your name. The audit below catches all of this on Day 1 in 45 minutes.

Most contractors are not trying to lock you out. They set the accounts up under their own email on Day 1 because it was the fastest way to start, and nobody ever moved them. The damage is the same either way.

How to use it

Run the audit on a **Friday afternoon, alone**. 45 minutes if your accounts are tidy, 90 if they are not. Bring your company credit card, your password manager, and a fresh Notion doc.

Do not tell the team you are running it. If anything is wrong, you want to fix the access first and have the conversation second. A founder who asks “can you transfer the GitHub org to me?” on Monday gets a different answer than a founder who already owns the org and is asking why she did not on Day 1.

For each item, write the answer in your doc - pass or fail. If fail, copy the exact email or account name. You will need it for the recovery step.

The 12-item checklist

Code ownership

#	Check	What PASS looks like
1	**GitHub org owner.** Open the org. Settings - People. Who is listed as Owner?	Your name and email. Not the agency’s, not a shared `dev@` mailbox.
2	**Repo settings.** Settings - Collaborators. Can you remove every contractor right now without asking permission?	You can. Org-owner permission means you can remove any user.
3	**Branch protection on `main`.** Settings - Branches. Is `main` protected, and can you override the protection in an emergency?	`main` is protected, and your account has admin override rights.

Cloud ownership

#	Check	What PASS looks like
4	**AWS root account email.** AWS console - top-right - Account. What is the root user email?	Email on a domain you control: `you@yourcompany.com`.
5	**Billing access.** AWS Billing dashboard. Whose card pays the bill? Can you see invoices for the last 12 months?	Your card or your company AmEx, and you can download every invoice.
6	**IAM admin user.** IAM - Users. Is there an admin user that is yours, separate from root?	Yes, with MFA on, and the password in your password manager.

Database and secrets

#	Check	What PASS looks like
7	**Production DB credentials.** Where is the prod DB password stored? Can you read it tonight without paging the lead engineer?	You open AWS Secrets Manager (or 1Password / Vault) and see it yourself.
8	**Secrets store ownership.** Secrets Manager, Vault, Doppler, or `.env` files in a private repo - who is the admin?	You are. If a developer rage-quits tonight, you can rotate every secret tomorrow.
9	**Database backups.** When was the last successful backup? Can you restore one to a staging DB by yourself?	Backups run nightly, the last 7 days are listed in RDS, and you have a one-page restore runbook.

Domain and external services

#	Check	What PASS looks like
10	**Domain registrar.** Log in to Namecheap, GoDaddy, or Cloudflare Registrar. Whose name is on the WHOIS? Whose email gets the renewal notice?	Yours, on your company email. The account is paid by your card.
11	**DNS provider.** If DNS lives elsewhere (Cloudflare, Route 53), can you log in and add an A record right now?	Yes, on an account in your name, with MFA.
12	**Third-party API keys.** Stripe, SendGrid, OpenAI, Twilio, Plaid - whoever you pay every month. For each, is the account in your name and the billing on your card?	No agency owns an account that touches your customers' money or data.

If you do not know what an item means, that is part of the result. "I have never heard of Secrets Manager" is a failed answer for #7 - it tells you nobody has briefed you on where production passwords live.

What good looks like vs. what bad looks like

#1 - GitHub org owner

> Bad: Owner is `contractor@external-email.com`. You are listed as a Member. Good: Owner is `founder@mycompany.com` (you). Contractor engineers are added as Outside Collaborators.

If the contractor's email is on the Owner row, they can delete the org tomorrow morning and GitHub support will not help you.

#4 - AWS root account email

Signal	Root email	Recovery if you lose access
Fail	`aws@external-email.com`, you have IAM only, never logged into root	None - the contractor controls it
Pass	`aws@mycompany.com`, password in your 1Password, MFA on your phone with a backup code in your safe	You reset it yourself in five minutes

The root account owns everything underneath it. If the contractor controls the root email, they can lock you out of every AWS service in 10 minutes.

#7 - Production database password rotation

The fail looks like one sentence: *"Marcos knows it. I would have to ask him."* The pass looks like one action: you open AWS Secrets Manager right now, read the password, and remember the last time you rotated it (e.g. March, when the previous DBA left).

If only one person can rotate the prod DB password, you do not have a database. You have a single point of failure.

#10 - Domain registrar

> Bad: Renewals come to a contractor's email. You have never logged into the registrar. Good: Logged into Namecheap with your account. WHOIS shows your name. Auto-renew is on, charged to your card.

Plan a 14-day buffer for a domain transfer: ICANN's release window is five days, and many registrars add a 60-day lock after registration or a recent transfer. If the contractor will not release the auth code, your customers cannot reach your site for two weeks.

What to do if the audit fails

Do not panic. Do not tell the team yet.

Most failures are sloppy Day-1 setup, not malice. Frame the ask as "can you help me move this over, doing some housekeeping" - you will get a faster transfer than "why is my company under your name?"

Recovery order

> 📧 ***Copy-pasteable recovery email - send to the team if the audit fails (do NOT panic, frame as housekeeping):***
>
> Subject: `Quick housekeeping — moving a few accounts to my email`
>
> Hi [lead engineer / contractor contact],
>
> Doing some account housekeeping this week. Can you help me move these over?
>
> - [List only the failed items, one per line. Example: “GitHub org owner → founder@mycompany.com”]
>
> Should take 5-10 minutes each. Happy to jump on a call if easier.
>
> Thanks, [Your name]

- **Code (#1-3):** GitHub org transfer takes five minutes. Slack the lead engineer with your target account email.
- **Cloud (#4-6):** AWS root email change is self-service if you have the root password. If you do not, AWS support recovers it with your incorporation documents (3-5 business days).
- **Database and secrets (#7-9):** Set up your own AWS Secrets Manager (or 1Password vault) tonight. Migrate secrets next sprint. Schedule a backup-restore test.
- **Domain (#10-11):** Initiate the registrar transfer to an account in your name. Get the auth code. Budget 14 days. Do not change DNS during the window.
- **Third-party services (#12):** Most SaaS tools let you change account email and billing card from the settings page. One at a time, so receipts stay readable.

When to escalate

If the contractor does not transfer the GitHub org within 7 days, the AWS root within 14 days, or the domain auth code at all, retain a lawyer. Negotiate for a month yourself and you usually lose. The legal fee is $2K-$5K. The cost of a stalled checkout is much higher.

If you are in this spot, the contractor red flags checklist is the next read - a failed audit usually correlates with three or four other red flags - and the step-by-step exit guide covers the 30-day transition.

What to do after

- **Run this audit every quarter.** Twenty minutes once you have done it the first time. Recurring calendar block, last Friday of every quarter.
- **Add it to your due diligence checklist for the next hire.** Before you sign a contract, get it in writing that all GitHub orgs, AWS accounts, domain registrar accounts, secrets stores, and third-party API accounts will be

created under your company email from Day 1. Put it in the SOW. The founder's guide to hiring a contractor walks through the clauses worth requesting.

- **Forward the audit answers to your investor or board the same day.** Three quarterly audits in a row in their inbox is the cheapest investor-trust signal you will ever ship.

Built by JetThoughts as part of the From Idea to First Paying Customer curriculum.