

0-03

Access Request Checklist

The least access each system needs, from GitHub and CI to a database replica and LLM keys, and what I never need.

Axdale. Software engineering by Michael Lip, Warsaw, Poland (CET).

Version 1.0. 23 October 2026. Website <https://www.axdale.com>. Code record <https://github.com/theluckystrike>.

This checklist lists the least access I need, system by system. I send it pre-filled from the written plan. Grant only the rows the plan needs. Every grant goes to a named account for me, with multi-factor authentication where the system supports it. You can remove any access at any time, with no notice.

My account names are in the table below. Use these and nothing else.

System	My account	Status
GitHub	theluckystrike (https://github.com/theluckystrike)	Confirmed
Email for invitations	michael@axdale.com	Confirmed
Other systems	Invitation to michael@axdale.com	Created on invitation

1. GitHub or GitLab

Item	What to grant	Notes	Granted on	Removed on
Repository access	Outside collaborator on the named repositories only, role Write (GitHub) or Developer (GitLab)	A private fork that you own is also fine		

Item	What to grant	Notes	Granted on	Removed on
Main branch	Keep branch protection on, with required review and required checks	I never push to main and I do not merge my own pull requests unless you allow it in writing		
Organization role	None	I do not need membership of the organization, a team or admin rights	Not applicable	Not applicable
Issue tracker	Write on the project or label for this work	GitHub Issues, Linear or Jira		

2. CI

Item	What to grant	Notes	Granted on	Removed on
Pipeline runs	Runs that start from my pull requests	Comes with repository Write on GitHub Actions and GitLab CI		
Logs	Read the logs of runs on my pull requests	No access to other projects		
CI secrets	Test-scoped secrets only, in an environment limited to non-production	I never need organization secrets or deploy keys for production		

3. Cloud (AWS, Google Cloud or Azure)

Item	What to grant	Notes	Granted on	Removed on
Account	A named user in a sandbox or staging account, through your single sign-on or identity center	Never the root user, never a shared login		

Item	What to grant	Notes	Granted on	Removed on
Role	Read-only on staging by default. Write only on named sandbox resources the plan lists	For example AWS ReadOnlyAccess on the staging account	_____	_____
Session length	Short sessions with MFA, 8 hours or less	Access keys only if the plan needs them, with an end date	_____	_____
Production	None	A production action needs a written approval for that one action	Not applicable	Not applicable

4. Database

Item	What to grant	Notes	Granted on	Removed on
Source	A read replica or a staging copy with anonymized or test data	Never the primary production database	_____	_____
User	A read-only database user for me alone, with a statement timeout	SELECT on the named schemas or on masked views	_____	_____
Network	Through your bastion, VPN or a private tunnel you control	No public database endpoint	_____	_____
Personal data	Columns with personal data masked or excluded	If personal data is needed, sign the DPA (A-04 Data Processing Agreement) first	_____	_____

5. Error tracking and monitoring (for example Sentry or Datadog)

Item	What to grant	Notes	Granted on	Removed on
Role	Member with read access to the named projects only	No admin, no billing, no team management		
Data scrubbing	Keep server-side scrubbing of personal data on	I add tests so new logs carry no email addresses or phone numbers		

6. LLM provider keys

Pick one option. Option A is the strongest for procurement, because your code is processed under your own contract with the provider.

Option	What to grant	Notes	Granted on	Removed on
A. Your key	A key in a separate project on your provider account, with a monthly spend cap and rate limits	Shared through your secret manager or a one-time encrypted link. I store it only in a local file that git ignores		
B. My account	Nothing to grant. You tick my provider in the Order Form	Only the providers you approve see your code	Not applicable	Not applicable
Production keys	None	I never need a production key	Not applicable	Not applicable

7. What I never need

1. Owner or admin rights on your GitHub or GitLab organization, your cloud account or your identity provider.
2. The cloud root user, a shared login or a break-glass account.
3. Write access to production systems, production databases or production configuration.
4. Production secrets, production API keys or customer credentials.
5. Customer personal data, unless a DPA is signed and the plan needs it.
6. Billing, payroll, HR or finance systems.

7. Your chat workspace. Email and your tracker are enough.

8. How to share secrets

Share secrets only through your secret manager or a one-time encrypted link that expires. Secrets never go in email, chat, issues, pull requests, code or AI prompts. If a secret is exposed, I tell you within 24 hours and ask you to rotate it.

9. When access ends

I remove or hand back every row above within 5 Working Days after the engagement ends, and I ask you to rotate any key I held. O-05 Handover and Offboarding shows the confirmation you receive, with a date for each row.

Axdale, Michael Lip. Warsaw, Poland (CET). michael@axdale.com.
www.axdale.com. Code record github.com/theluckystrike. Version
1.0. 23 October 2026.