Railway incident recovery guide

Review status

This is a draft section for the Astra Security Railway incident recovery guide. It is not approved for publication.

- **Reviewer:** Ephrim / Astra Security security team
- **Next owner after approval:** Sanskriti, for inclusion in the published guide
- **Source currency:** Reflects Railway guidance available on October 22, 2026. The incident is ongoing, so recheck Railway's primary sources right before approval.

Scope note: Railway's October 21, 2026 disclosure concerns possible unauthorized access to customer environment variables and secrets. It is separate from the PHP2Shell vulnerabilities. Nothing in this guide should imply that PHP2Shell caused the Railway incident.

The customer-facing guide copy runs from "What happened" through "Sources". The final section, "Internal: security review checklist", is for reviewers only and must be removed before publishing.

What happened

Railway recommends that all users rotate their Railway-stored environment variables and private registry credentials as a precaution.

On October 21, 2026, Railway disclosed that an unauthorized third party may have accessed encrypted customer environment variables and obtained the decrypted contents of some of them. These variables may contain API keys, database credentials, and private registry credentials.

Railway initially indicated that the impact might be limited to a subset of users. It has said it will contact confirmed affected customers directly by email as its investigation progresses.

Step 1: Check whether your environment may be affected

Who should act: Railway workspace administrators, application owners, DevOps engineers, and security teams responsible for services deployed on Railway.

1. Read Railway's official security disclosure and follow incident updates on Railway Status.
2. Inventory the environment variables stored in Railway, including API keys, database credentials, private registry credentials, and other application secrets. [Security review]
3. Check whether your workspace uses Railway-managed MySQL, Redis, MongoDB, storage buckets, or GitHub connections. Railway has published a separate procedure for each: MySQL, Redis, MongoDB, and storage buckets and GitHub.
4. Watch for a direct email from Railway confirming whether your account was affected (source).

Don't wait for an individual notification. Railway's precautionary rotation guidance applies to all users, not only confirmed affected accounts (source).

Step 2: Rotate affected secrets and credentials

Railway recommends immediately rotating all environment variables stored in Railway, including API keys, and all private registry credentials. It also recommends reviewing service logs for unusual activity (source).

Apply the instructions that match the resources you use:

Resource	What to do	Avoid	Source
General environment variables and API keys	Rotate the credentials stored in Railway and update the applications or services that use them. Use each credential provider's supported revocation and replacement process. [Security review: the provider-process wording is Astra Security's]	—	Disclosure
MySQL (Railway template)	Open the MySQL service, go to Database → Config → Connection, and select Regenerate next to the password. Railway updates the password and related template variables together.	Manually editing MYSQLPASSWORD or MYSQL_ROOT_PASSWORD, or running ALTER USER.	MySQL guidance
Redis (Railway template)	Go to Database → Config → Connection and select Regenerate next to the password.	Manually editing REDISPASSWORD or using CONFIG SET requirepass.	Redis guidance
MongoDB (Railway template)	Follow Railway's published credential rotation instructions.	—	MongoDB guidance
Storage buckets	Railway says it invalidated and regenerated ACCESS_KEY_ID and SECRET_ACCESS_KEY for every bucket in every	Assuming a manual reset is still required.	Buckets and GitHub guidance

	environment. Verify the new credentials in each environment.		
GitHub connections and registry tokens	Railway invalidated the GitHub connection tokens it held and removed GitHub access tokens stored as registry credentials. Reconnect GitHub to Railway, then revoke the affected registry tokens separately.	Assuming removal from Railway invalidated the tokens on GitHub. It does not.	Buckets and GitHub guidance

Step 3: Redeploy dependent services and update external consumers

Rotating a credential does not guarantee that every application, worker, or integration is using the replacement.

1. Identify every service that consumes each rotated credential. [Security review]
2. Redeploy dependent Railway services so they read the updated variables at startup. [VERIFY: source]
3. Update consumers that use manually pasted connection strings or externally stored credentials. For MySQL and Redis, Railway distinguishes these from services that use reference variables (MySQL, Redis).
4. For storage buckets, redeploy the services that consume the new credentials (source).
5. Reconnect your GitHub account in Railway, and replace or revoke affected GitHub registry tokens as Railway directs (source).

Availability caveats: Railway reported that deployments were paused during its initial investigation. If redeployment is unavailable, follow Railway's incident updates rather than assuming a deployment succeeded. Regenerating a database password may also disconnect existing sessions, and dependent services may not reconnect until they are redeployed. [VERIFY: source for both caveats]

Step 4: Verify recovery and review for suspicious activity

A successful deployment does not prove that no unauthorized access occurred. After rotating credentials and restoring services:

- Confirm that dependent applications connect successfully with the replacement credentials. [VERIFY: source]
- Check deployment logs for authentication failures or other unexpected errors. [VERIFY: source]
- Confirm that external consumers are updated, and that old credentials no longer work where revocation can be independently verified. [VERIFY: source]
- Review Railway service logs for unusual activity, as Railway recommends (source).

- For Railway-managed MySQL, follow Railway's verification and misuse-review procedure.
- For Railway-managed Redis, follow Railway's verification and misuse-review procedure.
- For storage buckets and GitHub, review the workspace audit log to tell Railway's reset operations apart from changes your team made, and check each provider's activity records for unexpected access (source).

Record the rotation and recovery status, preserve relevant logs, and escalate suspicious activity to your security team. [Security review]

Additional recommendations from Astra Security

These go beyond Railway's published instructions. [Security review: the whole section needs approval against Astra Security's incident-response standards before publishing.]

- **Preserve evidence before making destructive changes.** Where feasible, keep audit logs, deployment logs, authentication events, and timestamps before they expire or are overwritten. Railway notes that audit-log retention depends on your plan, and recommends preserving evidence before restoring altered database data. [VERIFY: source]
- **Review credential scope.** Map which applications, cloud accounts, databases, and third-party integrations each potentially exposed credential could reach. Prioritize credentials with broad privileges.
- **Look for persistence.** Check for unfamiliar account activity, new credentials, unexpected deployments, or changed access permissions that could survive a password rotation.
- **Document recovery.** Record each affected resource, its rotation status, the services redeployed, external consumers updated, verification results, and open investigation items.

Need a hand?

If you'd like help working out which credentials or services may be affected, or how to prioritize recovery, reach out to your Astra Security CSM. We're happy to be a second set of eyes.

Sources

- Railway's October 21 security disclosure
- Railway Status: incident updates
- MySQL credential rotation and verification
- Redis credential rotation and verification
- MongoDB credential rotation
- Storage bucket credential reset and GitHub reconnection