

Ontaric GmbH

Resourcecloud

Audit log

Who changed what, when and from where - and how you can rely on it. A guide for administrators, auditors and data protection officers.

As of 11 October 2026

Ontaric GmbH · Jürgen-Töpfer-Straße 113 · 22763 Hamburg, Germany
datenschutz@resourcecloud.io

Contents

Summary

1. What is recorded
2. Who can see it and how you set it up
3. Using the page
4. Detecting tampering
5. Operations: key, storage, backup
6. Retention and privacy
7. Interfaces
8. Checking it independently
9. Limits
10. Frequently asked questions

Contact

Audit log

Who changed what, when and from where – across tasks, projects, people, permissions, settings, invoices and time bookings. Entries cannot be edited or deleted, and a verification chain makes any later change detectable.

Summary

A page of its own

Admin > Audit Log. Only the RC Admin and the *Auditor* role can open it; it appears on no task or project page.

Read-only

Nobody can edit or delete entries, not even an administrator. A verification chain makes changes to the database detectable.

No effort

The log runs by itself after the update. Resourcecloud keeps the key and the anchor copies in the S3 storage it already uses for files. Only one copy of the key is yours to keep.

Your decisions

Who is an Auditor, how long entries are kept (default 10 years) and when a person is pseudonymised.

What you should do

1. Make sure the **scheduler container is running** and the **S3 file storage** is set up (section 5). The log needs both to seal entries and to store anchor copies.
2. Give the **Auditor** role to the people who should read the log (section 2).
3. Set the **retention period** if 10 years does not suit you (section 6).
4. **Download the key** and keep it with your other important secrets; confirm that on the page (section 5). As RC Admin you are responsible for that copy.
5. Back up the **database and the S3 bucket together** (section 5). The verification key is in the bucket.
6. Tell your works council and your data protection officer that the log exists (section 6).

1. What is recorded

Recorded	Never recorded
Sign-in: sign-in (password, LDAP, passkey, SSO), sign-out, failed attempts People and access: users created, changed, activated, deactivated, deleted; password changes; security roles; team membership Settings: global settings, teams, API tokens, webhooks Tasks: created, changed, finished, reopened, deleted, restored - with the old and new value Projects and milestones: status, settings, dates, costs Money and time: invoices (status, amounts), changed or deleted time bookings, vacation requests and decisions, worktime corrections Master data: customers, companies, custom fields, project templates, baselines Data leaving the system: invoice downloads, the one-time passwords of a bulk import, exports of the log itself Imports: one summary entry per import run	Passwords, tokens and other secrets (shown as ***) The text of comments and descriptions - only that a description changed, and its length A person's name, e-mail address and phone number inside a change - only their length What someone only <i>looked at</i> Date moves that the autoplaner makes by itself Making a time booking (every comment books minutes); changing or deleting a booking is recorded

Every entry states **who** (a person, an API token, the MCP access, the scheduler or the system), **through what** (web, REST API, MCP, mail, import, scheduler), from **which IP address**, **what** happened, to **which object** (its name at the time), which **team and project** it belongs to, and what changed as *old* → *new*.

Private tasks leave only the fact that something happened - no title, no content. The entry is visible only to the people on the task, not even to an RC Admin who is not one of them.

The log starts with the update that introduces it. Older history stays in the existing per-area logs (project log, story log, invoice events).

2. Who can see it and how you set it up

Two roles may read the log: the **RC Admin** and the **Auditor**. Team leads and project managers do not see it - it is not a management report. The Auditor can read, filter, export and check the integrity; they cannot change anything, including the retention. Retention and pseudonymising are reserved for the RC Admin.

Giving someone the Auditor role

1. Open **Admin > Org Management** and the user who should read the log.
2. Add the role **Auditor** to the secondary roles (category *Administration*) and save.
3. The user now sees **Audit Log** in the Admin menu.

Granting the role is itself in the log (*Roles changed*).

3. Using the page

- **Filter** by period, who, what happened, search text and entries per page (25 to 200). In *What happened* the start of an action is enough: `user.` shows everything about users, `task.done` only finished tasks, `auth.login_failed` failed sign-ins.
- **Page through** with *Older entries* and *Newer entries*. The page stays fast however large the log gets.
- **Export** what the current filters show as CSV or JSON (all of it, not only the visible page). The export is itself an entry. Cells that a spreadsheet would run as a formula are made harmless.
- The **Details** column shows changes as *field: old → new*.

Typical questions

Question	How to find it
Who gave this user the Finance role?	What happened: <code>user.roles_changed</code> , search: the user name
Who changed this task's due date?	Search: the task title (or object number); the details show old and new
Who downloaded the invoice?	What happened: <code>export.</code>
When was this setting changed?	What happened: <code>settings.</code> ; the details name the field
Were there sign-in attempts that failed?	What happened: <code>auth.login_failed</code> ; a user name appears only if the account exists
What did this API token do?	Who: the token owner's name; under it the column <i>Who</i> shows the access (REST API or MCP)

4. Detecting tampering

Every entry is written in the same database transaction as the change itself: if the change is rolled back, so is its entry. About half a minute later the scheduler **seals** new entries: it gives each a position and computes a keyed hash over the entry and the hash of the one before it. All entries thus form a **chain**. If an entry is later changed, removed or inserted, the chain no longer fits from that point on.

In addition the database refuses UPDATE and DELETE on sealed entries where it allows installing that protection (the page shows it), and the scheduler stores a **signed anchor copy once a day** in the S3 storage: "at position N the chain stood at hash H." Someone who has the database *and* the key and rewrites the whole chain cannot reach those copies – the comparison gives it away.

Checking the integrity

The **Verify integrity** button recomputes the whole chain and compares the anchor copies. It reads the entire log; use it when it matters, not every minute. Possible results:

Message	Meaning	What you do
The chain is intact	All entries fit together, and the anchor copies match the chain.	Nothing.
... was changed	An entry or the one before it was altered after sealing.	Preserve the state, contact your security officer and Ontaric. The message names the position of the first affected entry.
... an entry is missing	An entry is missing between two others.	As above.
... link does not fit / entries at the end are missing	The order was altered, or the newest entries were removed.	As above.
... sealed with a different key	The key the entries were sealed with is missing (the message gives its ID). Usually the bucket or the configuration was changed.	Add the key you downloaded (card <i>Retention and privacy, Add an older key</i>). If it is lost, choose <i>Accept as lost</i> (section 5).
Anchor copy ... changed / missing / wrong signature	The log was rewritten after the copy was made, or the copy itself was altered.	As for "was changed".

The page header shows how many entries are **sealed** and how many are **waiting**. If entries wait for more than a few minutes, the scheduler is not running.

5. Operations: key, storage, backup

What Resourcecloud needs

Component	Purpose
Scheduler container	Seals entries (every minute), stores the daily anchor copy and removes entries after the retention period. If it does not run, the log stays complete but unsealed, and there are no anchor copies.
S3 file storage	Holds the verification key (<prefix>/audit/audit.key), older keys you add (<prefix>/audit/keys/) and the anchor copies (<prefix>/audit/anchors/). It survives restarts, updates and several replicas - unlike a container's disk.
Database	Holds the log itself.

You do not have to configure anything: on first use Resourcecloud creates the key in the bucket. All containers (web, scheduler, replicas) read the same key.

Without S3 file storage Resourcecloud falls back to a file on the server's disk. If that disk does not persist (containers without a volume, several replicas), every restart starts a new key and entries sealed before can no longer be verified. The page then shows a warning. Set up S3 storage, or set the key yourself (next paragraph).

Downloading and keeping the key

As soon as Resourcecloud has created the key, a notice appears at the top of the page: "**Download the audit key and keep it safe**". The RC Admin and the Auditor can download the key there as a small text file (resourcecloud-audit-key-<ID>.txt). A link *Download key* stays available in the *Integrity* area.

- **Keep the file where you keep important secrets** (a password manager, a vault) - not only on the server and not in the database backup.
- Then, as RC Admin, click "**I have stored it**". The notice disappears, and the page shows who confirmed the copy. As RC Admin you are responsible for that copy existing.
- Every download and every confirmation is in the log (without the key itself).
- Whoever has the key can verify the chain - and also forge one. Give the file only to people who may have it.

Setting the key yourself (optional)

If the key should not live in the same bucket as the files, set it in your deployment secrets: the environment variable RC_AUDIT_KEY (at least 32 random characters, identical for all containers). It takes

precedence over everything else. The key then lives where you already manage the database credentials, and no longer in the bucket.

Backup and restore

- **Back up the database and the bucket together.** The database holds the log, the bucket holds the key and the anchor copies. Versioning or replication of the bucket, as you use for files anyway, is enough.
- **Restoring database and bucket:** verification keeps working, the key comes from the bucket.
- **Restoring only the database into a new, empty bucket:** a new key is created, and new entries are sealed with it. *Verify integrity* then reports that the key of the older entries is missing (with its ID). As RC Admin, add it under *Retention and privacy > Add an older key* (paste the whole file you downloaded, or just the key from its "Key" line). The whole chain can then be verified again, across both keys.
- **Key lost** (no copy left): no entry is lost; they stay readable and exportable. After the message, choose **"Accept as lost"** and confirm. Those entries are then no longer verified by content, but still for order and links: if one is removed or inserted, it shows. The check says how many entries were only checked that way. Who accepted it is in the log. If you add the key later after all, the entries are fully verified again.

Protecting the anchor copies better

The harder the anchor copies are to alter, the more they achieve. Use your S3 provider's **object lock** (or at least versioning) for the prefix `audit/anchors/`, and do not give the bucket credentials to the database administration.

Storage and load

An entry takes about 1 KB; a million entries take roughly 1 GB of database. Writing an entry costs less than a millisecond, and sealing runs in the scheduler without burdening any request.

6. Retention and privacy

RC Admins see the card **Retention and privacy** on the page.

- **Keep entries for (years):** the default is 10, the usual horizon for business records; **0 keeps them forever**. Once a day the scheduler removes older entries from the oldest end; the rest stays verifiable. The removal and every change of the setting are entries themselves.
- **Pseudonymise a person** (for an erasure request): enter the user name and confirm. Every entry by or about that person then shows a pseudonym such as pseudonym-ab12cd34. The same person gets the same pseudonym (as long as the key stays the same), so their entries still belong together, but the name cannot be read back. It **cannot be undone**, it works for deleted accounts too, and the verification chain stays valid. First name, surname and e-mail address are not in the entries in plain text in the first place.

What is personal data

The log contains user names and IP addresses of the people acting. It records **changes**, not what someone looked at, and not working time. Mention it in your record of processing activities, with its purpose (traceability and security of changes) and the retention period.

Works council: tell your works council that the log exists, who can open it and how long it is kept. How Resourcecloud treats the analysis of individual employees is described in the *Works Council Package*. This guide is not legal advice.

7. Interfaces

The log can only be **read**; no interface can write to it. Access follows the same rule as the page: the user who owns the API token must be an RC Admin or Auditor. A read-scope token is enough.

Call	Purpose
GET /api/v1/audit-log	Entries, newest first, with the same filters as the page; page with next and prev.
GET /api/v1/audit-log/verify	Counts, retention, anchor status and the result of the check. Reads the whole log.

The same queries exist as MCP tools (`list_audit_log`, `verify_audit_log`). The full description is in the API documentation. Forwarding the log to a SIEM system does not exist yet; until then you can fetch the JSON export or the query on a schedule.

8. Checking it independently

An auditor does not have to trust the button. Every sealed entry carries, in the JSON export and the API answer, a chain block with the raw values the hash was computed over. For the entry at position n and the hash of the previous entry p (64 zeros for the first):

`hash = HMAC-SHA256(key, p + "|" + JSON array)`, the array holding these fields in this order: `id`, `seq`, `at`, `actor_id`, `actor_type`, `via`, `ip`, `request_id`, `action`, `object_type`, `object_id`, `team_id`, `project_id`, `changes`, `meta`, `restricted_to` (each as text, null stays null; JSON without escaping of slashes or non-ASCII characters). The actor's name and the object's name are **not** part of the hash - that is why a person can be pseudonymised without breaking the chain. You give the auditor the key. A more detailed technical description is in the guide *audit-log-guide*.

9. Limits

- The chain makes changes **detectable**, it does not prevent them: *tamper-evident*, not *immutable*. With the anchor copies in a protected bucket, even a complete rewrite by someone with the database and the key is detectable.
- Entries from the last half minute are not sealed yet.
- What a browser exports by itself (the table export buttons on report pages) never reaches the server and is not in the log. Downloads the server makes (invoices, bulk-import passwords, the log) are.
- Changes made directly in the database, past Resourcecloud, are not recorded; a missing or altered entry in the chain and the anchor copies do show that the log was touched.
- The log does not record what someone only looked at.
- If someone tampers with the database directly before the scheduler has sealed the entries, that window of at most a minute cannot be protected.

10. Frequently asked questions

Can an RC Admin delete or change entries?

No. There is no function for it. Anyone who tries directly in the database breaks the chain; the check names the spot.

Why don't I see my own sign-in from this morning?

Sign-ins are recorded from the update that introduces the log. Someone who signed in before and stayed signed in shows up after their next sign-in.

Why do some entries show a lock?

They concern private tasks; only the people involved see them.

Why do some entries "wait" for the seal?

The scheduler seals entries older than about half a minute every minute. If some stay longer, the scheduler is not running.

Do we have to back up the key?

Yes: download it when Resourcecloud asks you to, and keep it with your secrets (section 5). It is also in the S3 bucket and is backed up with it. If you set RC_AUDIT_KEY yourself, it belongs with your secrets anyway.

We lost the key. Now what?

You need no support: *Verify integrity* names the ID of the missing key. Add it if you find a copy, or choose *Accept as lost*. New entries are always sealed and verified with the current key.

What happens if we change the bucket?

Copy the prefix `audit/` along, or set the key as RC_AUDIT_KEY. Otherwise a new key is created (see "Backup and restore").

Can we switch the log off?

No. It is part of the platform. You decide how long entries are kept.

Contact

Ontaric GmbH

Jürgen-Töpfer-Straße 113

22763 Hamburg, Germany

Enquiries: datenschutz@resourcecloud.io

As of 11 October 2026. The content reflects the state of the product on that date and is backed by automated tests where technically possible. Not legal advice.