WHITEPAPER · CLOUDM MIGRATE 5.0 AND LATER

Data security in a self-hosted CloudM Migrate deployment

Where your data goes during a migration, what is kept, for how long, and who can reach it.

WRITTEN FOR

Security reviewers, compliance officers and infrastructure architects evaluating CloudM Migrate against a security questionnaire or a data-protection review, and the presales engineers who answer those questions on our behalf.

Applies to CloudM Migrate 5.0 and later, self-hosted. September 2026.

0 Executive summary

A migration is a temporary, high-volume movement of an organization's most sensitive data. The security question is not whether the tool is encrypted. It is whose infrastructure the data crosses, what is kept, and for how long.

A self-hosted CloudM Migrate deployment keeps migration content inside infrastructure the customer controls. Content is held only transiently, and encrypted while an item is in flight. What persists afterwards is the metadata needed to run delta passes and produce reports. CloudM itself receives licensing metadata and nothing else. That combination is what allows the product to fit regimes where processing location and personnel access are the deciding factors, rather than encryption alone.

THE SIX FACTS, AND WHERE TO READ THEM

Data path. Migration content moves between the customer's servers and the source and destination platforms. It does not traverse CloudM infrastructure. Section 2

In transit. HTTPS to platform APIs, TLS 1.2 or higher as negotiated. No inbound connections from the internet. Section 3

During processing. One item at a time; small items in memory, larger items in temporary files encrypted with AES-256 and deleted after transfer. Section 4

What persists. Project configuration, user lists, run results and reports, plus item identifiers and content hashes for delta passes. Not content. Section 5

What CloudM receives. Licensing metadata from the servers, and console usage analytics from the administrator's browser. No message bodies, file contents, file names or folder structures. Section 6

Deployment control. The servers, the network and their lifecycle belong to the customer, so deletion is theirs to perform and theirs to verify. Sections 5 and 9

1 Why migration security is a different problem

Steady-state security controls are designed around a stable picture: known systems, known volumes, standing access that changes slowly. A migration breaks all three at once. It moves an organization's entire mail, file and calendar estate in a compressed window, frequently

over a single cutover weekend, using credentials elevated enough to read every mailbox and every drive in the tenant. The controls that govern normal operation were not written for that shape of activity, which is why migration projects attract a separate review.

Reviewers who do this well converge on three questions.

1. **Whose infrastructure does the data cross?** Not only at rest, but every hop in between.
2. **What is retained once the migration is finished?** Including the records the tool needs to keep in order to work.
3. **Who can access any of it?** Which people, in which organization, under which controls.

CloudM Migrate is available in two deployment models, and they answer those questions differently. The hosted service is run and maintained by CloudM: it removes the infrastructure work, keeps the platform current without customer effort, and is the right choice for the large majority of migrations. This paper is not an argument against it. It describes the second model, self-hosted, for the cases where the answer to question one has to be “ours, entirely” — because a regulator, a contract or a data-protection team requires processing to happen on infrastructure the customer owns and staff the customer employs.

2 Deployment model and trust boundary

In a self-hosted deployment, CloudM Migrate is installed software. The customer provides Windows servers, either on premises or in their own cloud tenant, and runs the product on them. A deployment consists of a primary server, which hosts the web interface and coordinates the migration, and optionally one or more secondary servers that add parallel processing capacity. PostgreSQL and Redis support the application; both run on the primary server by default, and both can be placed on separate servers the customer provides.

CloudM runs nothing in the data path. There is no CloudM-operated relay, queue or staging tier between the source and destination platforms. Migration content is read by the customer's servers from the source platform's API and written by the customer's servers to the destination platform's API.

That makes the trust boundary straightforward to draw. Everything the product does — the interface, the processing, the databases, the reports — sits inside the customer's network. From the Migrate servers, two kinds of connection cross the boundary outbound: calls to the source and destination platform APIs, which carry the content being migrated, and calls to the CloudM licensing service, which carry licensing metadata. Separately, the

administrator's browser loads analytics, support and font resources when the web console is open; those connections originate from the browser, not from the servers, and are covered in sections 3 and 6. Nothing crosses the boundary inbound.

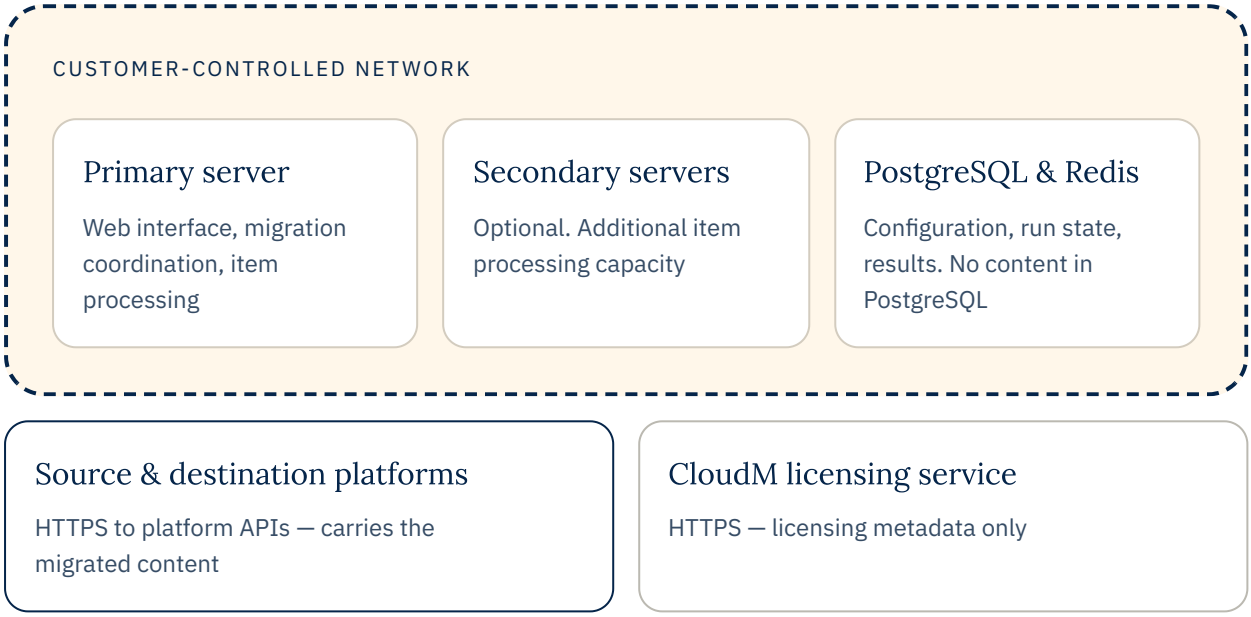

Figure 1. Trust boundary and data flow. The dashed boundary encloses everything the customer controls, and the two boxes below it are the only destinations the Migrate servers reach outbound. The administrator's browser reaches analytics, support and font endpoints separately; see sections 3 and 6.

3 Data in transit

Migration content moves over HTTPS between the customer's servers and the source and destination platforms, using TLS 1.2 or higher as negotiated between Windows and the platform endpoint. The product uses the operating system's TLS stack, so the negotiated version and cipher suite follow the customer's own Windows configuration and platform policy rather than a setting inside the application.

Administrators reach the web interface over HTTPS on port 443. The installation provides a self-signed certificate so the interface is available immediately; it is intended to be replaced with a certificate issued by the customer's own authority.

No inbound connections from the internet are required. The interface is reached from inside the customer's network, and connections between the primary and secondary servers are internal traffic on documented ports, scoped by firewall rule to the local subnet.

Outbound destinations

Two separate allow-lists apply, and it is worth reviewing them as two. A proxy log will show both, and the distinction between them is the distinction between the data path and the administrator's session.

FROM THE MIGRATE SERVERS

- The source and destination platform API endpoints, for the migration itself
- `tracker.cloudm.io` and `migratortracker-hrd.appspot.com`, for licensing
- `storage.googleapis.com`, for installer prerequisites and update checks
- `support.cloudm.io`, for support access

FROM THE ADMINISTRATOR'S BROWSER, WHEN THE WEB CONSOLE IS OPEN

- `content.pendo.cloudm.io` and Pendo's own hosts, for product analytics
- `static.zdassets.com` and `*.zendesk.com`, for the in-product support widget
- `fonts.googleapis.com` and `fonts.gstatic.com`, for fonts. The HTML metrics output also pulls the Google Charts loader from `www.gstatic.com` when it is opened

None of the browser-side endpoints sit in the migration data path. They are reached by the administrator's browser during a console session, and they see that session, not the content being migrated. Section 6 sets out what each one receives.

4 Data during processing

This is the section reviewers question most closely, because it is where content is briefly at its most exposed: read out of one platform, held somewhere, written into another.

CloudM Migrate processes one user at a time, and within a user, one item at a time. There is no bulk staging area: at no point is a copy of the estate, or of a mailbox, assembled anywhere. The unit of work is a single item, and the handling depends on its size.

Items under 10 MB are handled in memory and never written to disk. Larger items are written to a temporary file, encrypted with AES-256 using a key generated for that item, and the file is deleted once the item has been written to the destination.

Alongside the items, the product keeps working state. Each user being migrated has a working cache: a SQLite database, encrypted with AES-256, holding the bookkeeping needed to move that user's data and removed when the migration completes or the

environment is cleaned up. Redis, on the primary server and inside the customer's network, holds live migration state — run history, statistics, the retry queue and server heartbeats. Migration content is never written to PostgreSQL.

STORE	WHAT GOES IN	ENCRYPTED	WHEN IT IS REMOVED
Server memory	Items under 10 MB, in full	Not applicable	As soon as the item is written to the destination
Temporary files	Items of 10 MB and above, one file per item	AES-256, per-item key	Deleted once the item is written to the destination
Per-user working cache	SQLite database of per-user migration bookkeeping	AES-256	On completion, or when the environment is cleaned up
Redis	Migration history, live statistics, retry queue, heartbeats	On the primary server, firewall-scoped to the local subnet	History is exported to the project record; state clears with the run
PostgreSQL	Project configuration, connection settings, user lists, run results. No content	Connection credentials stored encrypted	When the project is archived and deleted, or the server is decommissioned

Table 1. Data lifecycle by store, during and after a migration run.

5 Data at rest after the migration

Reviewers are right to treat “nothing is kept” as a marketing line. Something is kept, deliberately, and it is worth being precise about what and why.

PostgreSQL holds the project record: configuration, connection settings with credentials stored encrypted, the user lists in scope, run results and generated reports. It also holds migration history — item identifiers and content hashes, not content. That history is the mechanism behind delta and incremental passes. When a project runs again, it compares what the source now holds against what it has already moved, and transfers only the difference. Without it, a cutover-weekend delta would mean re-reading and re-writing the entire estate. Migration reports and failed-item lists are retained as files on the server for the customer to download.

Removing all of it is a customer action. Migration history can be cleared from a project; the project itself can be archived and then deleted, which removes its record and results; and the servers, being the customer's own, can be decommissioned and wiped under the customer's normal asset process. No part of that sequence depends on CloudM performing a deletion on the customer's behalf, which also means the customer can evidence it directly in an audit.

On integrity, the product verifies transferred items using hashes where the destination platform exposes them, retries transient failures, and records anything that could not be moved in a failed-item list the customer can act on and re-run. The integrity approach is documented in full in the CloudM data-integrity statement, which is the right reference for a reviewer who needs that level of detail.

6 What leaves the customer's environment

Only one destination ever sees customer content: the platforms the customer is migrating between. The rest divide into calls made by the Migrate servers and calls made by the administrator's browser during a console session.

DESTINATION	WHAT IS SENT	WHAT IS NEVER SENT
Source and destination platforms	The migrated content itself, via their APIs	—
CloudM licensing service	Licensing metadata: the identifier of each migrated account, the source and destination platform, the product version and license consumption	Message bodies, file contents, file names, folder structures
`storage.googleapis.com`	Requests for installer prerequisites and update checks	Anything about the customer's data
Pendo — product analytics, from the administrator's browser	Administrator email address, display name, role, product version, Migrate user id, console page URLs and clicks	Migration content, platform credentials, end-user identities
Zendesk — support widget, from the administrator's browser	The widget load, and whatever the administrator types into a support conversation	Anything unless the administrator sends it
Google Fonts and Google Charts, from the administrator's browser	Font and script requests	Anything about the customer's data

Table 2. Outbound destinations, what is sent and what is not.

Pendo and Zendesk are usage analytics and support tooling on the administrator console. They are not part of the migration data path. There is currently no in-product switch to turn either of them off; customers who need to block them do so at the firewall or proxy, and the console continues to work when they are blocked. A reviewer will find these endpoints in a proxy log whether or not the paper mentions them, so they are listed here in full.

7 Access control and administration

Access to a self-hosted deployment has two layers, and both belong to the customer.

Inside the application, administrators sign in to the web interface with named accounts, and what they can do is governed by their assigned role. Sign-in and administrative activity are recorded in an audit log the customer can review.

Below the application, access to the servers is ordinary Windows administration under the customer's existing controls — domain policy, privileged access management, whatever they already run. CloudM has no access to those servers. Support engagements happen through the customer, on the customer's terms, with the customer in control of what is shared.

The credentials the product uses to reach the platforms are provisioned by the customer in their own tenants: on Google, a service account with domain-wide delegation restricted to the scopes the migration requires; on Microsoft, an app registration with the documented permissions. Because platform scopes and permissions change with the platforms, the current lists live in the CloudM knowledge base rather than in this paper, and should be read from there at review time.

For multi-server deployments, Service Manager lets an administrator control the Migrate services on every server from the primary. Day-to-day operation therefore does not require interactive logins to the secondaries, which keeps the number of interactive sessions on those machines low.

8 Fitting regulated environments

Residency and sovereignty. Processing happens wherever the customer places the servers. If the servers are in a particular country, region or cloud tenant, the processing is there too. No part of the migration is routed to a CloudM region, and there is no CloudM-side region to select, because there is no CloudM-side infrastructure in the data path.

ITAR-style regimes. Where export-control rules apply, encryption is not the qualifying control. Location and personnel are: where the data is processed, and who is able to access it. A self-hosted deployment lets the customer restrict both, because the infrastructure and the people who administer it are theirs. CloudM is not ITAR registered, and the compliance determination for any specific program belongs to the customer and their counsel.

Microsoft government clouds. CloudM Migrate supports the government cloud endpoints natively, including USGovernmentL4 (GCC High) and USGovernmentL5 (DoD), so a migration into those environments does not depend on custom endpoint configuration.

Where this comes up. Most commonly in government and the defense supply chain, in financial services under regulator-directed data handling, and in healthcare. The pattern is the same in each: the deciding factor is not the strength of the encryption but the location of the processing and the identity of the people with access.

This section describes fit, not certification. CloudM does not claim compliance with these regimes on a customer's behalf, and nothing here should be read as such a claim.

9 Operational responsibilities

Self-hosting moves real work to the customer. Reviewers should see that clearly before a deployment is agreed.

CLOUDM	CUSTOMER
Maintains the application, issues releases and fixes, documents requirements and configuration	Provides, patches and hardens the servers and operating systems
Provides product support and migration guidance	Owns network controls, firewall rules and the outbound allow-list
Documents the platform scopes and permissions the product requires	Provisions and rotates platform credentials in its own tenants
Records administrative activity in the product's audit log	Monitors, backs up and decommissions the servers, and reviews the audit log

Table 3. Shared responsibility in a self-hosted deployment.

10 Summary and next steps

A self-hosted CloudM Migrate deployment answers the three questions a migration review turns on. The data crosses the customer's infrastructure and the platform APIs, and nothing else. Content is held transiently and encrypted while an item is in flight. What persists is project configuration, results and the item identifiers and hashes that make delta passes possible, all of it on servers the customer can wipe. From the servers, CloudM receives licensing metadata; from the administrator's browser, the console reports usage analytics and loads support and font resources, none of which touch the data path.

For infrastructure detail beyond this paper — sizing, topology and performance — see the CloudM Migrate architecture material. For installation, ports, prerequisites and platform configuration, see the CloudM knowledge base. For a formal security review, a completed

questionnaire or a supplier assurance pack, contact your CloudM account team, who will bring in the product and engineering reviewers the questions need.

Appendix A Ports, endpoints and prerequisites

Summarized here so the paper stands alone. The knowledge base is authoritative and should be checked against the version being deployed.

CONNECTION	DIRECTION	NOTES
Web interface, HTTPS 443	Internal to primary	Self-signed certificate by default, replaceable with a customer certificate
Platform APIs, HTTPS 443	Outbound	Carries migration content. TLS 1.2 or higher as negotiated
Licensing, HTTPS 443	Outbound	`tracker.cloudm.io`, `migratortracker-hrd.appspot.com`
Prerequisites and updates, HTTPS 443	Outbound	`storage.googleapis.com`, `support.cloudm.io`
Console analytics, support widget and fonts, HTTPS 443	Outbound from the administrator's browser	`content.pendo.cloudm.io` and Pendo hosts, `static.zdassets.com`, `*.zendesk.com`, `fonts.googleapis.com`, `fonts.gstatic.com`, `www.gstatic.com`. Not in the data path; can be blocked at the proxy
Primary to secondary servers	Internal	Documented ports, firewall-scoped to the local subnet
PostgreSQL and Redis	Internal	On the primary by default, or on customer-provided servers

Table A1. Connections a deployment uses.

Prerequisites. Windows servers meeting the published system requirements, PostgreSQL and Redis (bundled with the installer or customer-provided), outbound HTTPS to the destinations above, and platform credentials provisioned in the source and destination tenants.

Appendix B Glossary

Primary server. The server hosting the web interface and coordinating the migration.

Secondary server. An optional additional server that processes items in parallel with the primary.

Batch. A group of users migrated together as one unit of work.

Delta pass. A re-run that transfers only what has changed since the previous run, using migration history.

Environment scan. A pre-migration read of the source tenant to size and validate the migration.

Service Manager. The tool for controlling Migrate services across all servers from the primary.