Self-Hosted Email Infrastructure

Latest engineering update · 12 November 2026

Prepared by Zahidul Islam for clients and developers.

Latest verified outcome

The service now has twelve mailboxes on two domains and a private web hub that reaches all twelve behind one login; the second domain and its six mailboxes were added on 12 November. The sixth mailbox went through the same drift-checked, hash-verified release and passed IMAP and SMTP login checks. On 2 November a test from each of the six arrived in the Inbox of an owner-controlled Gmail account under one sender name; that is one owner test, not a prediction for other recipients. A second hub release the same day added Gmail-style folders, sending to several people and file attachments.

A private mail hub on the same server

The hub runs in its own container next to the mail server. The owner signs in with a password and an authenticator code; mailbox passwords stay on the server. It lists every inbox together, shows messages as plain text without scripts or remote images, and replies from the mailbox that received the message, in the same thread.

Messages go out at once or at a set time in the recipient's time zone, with clock-change gaps refused. A durable queue applies per-mailbox hourly and daily caps below the server limits, a do-not-contact list, a duplicate check and one follow-up per first message, skipped if the recipient wrote back. Refusals are retried a limited number of times; a send interrupted after the message was handed over is marked uncertain and never resent automatically.

The second release adds a folder list like Gmail: All mail, Inbox, Sent, Drafts, Junk, Trash and any custom folders, with unread and total counts, older mail 50 at a time and search in every folder. One message can go to up to 25 people, either as a separate email to each, with caps, the do-not-contact list and the duplicate check applied per person and nothing queued if one is refused, or as one email with everyone copied. Up to 10 files can be attached; received attachments always download and never open inside the page.

How it was verified

56 automated tests in the first release and 98 in the second cover login, form protection, scheduling, the queue, folders, attachments and send outcomes, including a scripted mail server that refuses, drops and accepts messages. Lint, code-security and secret scans are clean, and pinned dependencies show no known vulnerabilities. Before release the hub ran against the live mail server: send now, a scheduled send, a refused duplicate, a threaded reply and a follow-up skipped after a reply.

In production: HTTPS and security headers checked; cross-site and token-less posts and a wrong login refused; the application port unreachable from the internet; the container non-root, read-only and without extra privileges. 22 release files and 18 application files in the running container match the reviewed source by hash. A scheduled test left 12 seconds after its due time and reached the Gmail Inbox, and an off-server backup copy, restored into a separate hub with no network, passed 14 of 14 checks: sign-in, queue and history intact, healthy 3.4 seconds after the stop.

Before the second release the live hub was compared with the recorded first release and showed no drift; a database snapshot was taken before the upgrade, and afterwards 26 release files and 18

application files matched the source. A real email to two people with a copy and two files arrived in an owner Inbox with both files byte-identical, the sent copy was filed in Sent, uploads worked over HTTP/1.1 and HTTP/2, and the isolated restore passed 14 of 14 checks again on the upgraded database.

On 12 November a second domain was added without disturbing the first. The mail server was compared with its recorded release before the change, and the existing accounts, signing keys and hub jobs were kept. The new signing key turned out to be generated but not used by the running server; adding it to the signing map fixed that, and all six new mailboxes then passed IMAP and SMTP logins. One scheduled test from each arrived in an owner Gmail Inbox with SPF, DKIM and DMARC passing for the new domain over TLS 1.3. An encrypted copy of all twelve accounts was taken off the server and restored with no network: all twelve logins and the expected inbox counts passed.

Problems found before release

An independent review rejected the first version: a strict referrer setting makes browsers send a blank origin on form posts, so every login would have been blocked. It also found a login lock attackers could keep extended, a health check blind to a stopped scheduler and a follow-up failure after a manually resolved send. New tests showed a rejection after the message body was counted as sent. All were fixed and retested; a second review accepted the release with minor follow-ups, applied before it went live.

The second release's review found four serious problems: the public login form read a request body before checking its size, a deploy would have replaced the only copy of the release it rolls back to, large text attachments could exceed the server's message limit once encoded, and people copied on group emails were missed by the duplicate check. All were fixed with tests and re-reviewed. The first real send also showed text files encoded in a way the server's own spam filter penalises, putting a copy in Junk; a smaller readable encoding moved it to the Inbox.

Scope and limits

One owner only. Gmail accounts, HTML display, bulk campaigns and bounce handling are outside these releases; attachments are limited to 10 files and 10 MB per message and removed from the hub 7 days after sending; caps count hub sends, not webmail sends; the restore was rehearsed on a small database, not a full server rebuild. No paid service was added.

Latest before / after · 2 and 12 November

Comparison of the service as recorded on 21 October with the work verified on 2 and 12 November. Older comparisons remain below.

Side-by-side changes

Area	Before · 21 October	After · 2–12 November
Mailboxes and access	Five mailboxes, each with its own webmail login.	Six, then twelve on 12 November. One hub login with password and authenticator code; one unified inbox.
Domains	One domain with six mailboxes.	Two domains and twelve mailboxes in the same hub; each domain signs and aligns its own mail.

Replies	Sent from whichever webmail session was open.	Sent from the receiving mailbox, in the same thread.
Scheduling	Immediate sending only; no scheduler.	Durable queue at the recipient's local time; clock-change gaps refused.
Sending rules	Server limits only; contact history by hand.	Per-mailbox caps, do-not-contact list, duplicate check, one follow-up skipped on reply.
Unclear sends	No queue to classify outcomes.	Limited retries for refusals; interrupted sends marked uncertain, never resent.
Folders and search	Folders only inside each separate webmail login.	All mail, Inbox, Sent, Drafts, Junk, Trash and custom folders for every mailbox, with counts, paging and search.
Several recipients	Possible in webmail, with no caps or duplicate check per person.	Up to 25: a separate email to each with the rules per person, or one email with everyone copied.
Attachments	Webmail only, outside the queue and its rules.	Up to 10 files and 10 MB, sent with scheduled mail; received files download, never open in the page.
Inbox test	21 October comparison: Spam at one Gmail destination.	Six of six reached an owner Gmail Inbox in one test on 2 November.
Recovery	Mail server backup with an isolated restore (19 October); no hub yet.	Daily hub database backup; an off-server copy restored in isolation passed 14 of 14 checks.
Release proof and cost	Mail server release hash-matched; existing server.	Hub and running container hash-match source; 56 then 98 tests; each release reviewed; same server, no paid service.

The table lists only work that was built and checked. Inbox placement depends on the receiving provider and changes over time.

Self-Hosted Email Infrastructure

Latest engineering update · 21 October 2026

Prepared by Zahidul Islam for clients and developers.

Latest verified outcome

The five-account service now has a dated operating review in addition to its original installation evidence. A fresh external check passed all 22 selected protocol, authentication and denial tests. Sixteen owner-only diagnostic messages passed SPF, DKIM and DMARC across three Gmail destinations; the ten latest comparison originals also verify TLS 1.3. These results establish the checked identities and transport, without promising how every recipient will handle future mail.

Operational reconciliation

A reconciliation matched all 25 initial messages submitted on 20 October to their exact Message-IDs in the live Sent folders, five per account. Remote-server records show 23 accepted messages and two delivery failures. A Sent copy proves a recorded attempt; a recipient server's acceptance does not prove reading, a qualified enquiry or a paid engagement. Failed routes remain excluded from further contact under the maintained workflow.

Google diagnostics and DNS preservation

Google Postmaster Tools now confirms domain ownership. The verification added one TXT record while preserving all 23 existing DNS records unchanged, producing 24 records in the captured inventory. Independent Google and Cloudflare resolver checks confirmed the added record. Google currently provides insufficient traffic data for a reputation assessment; ownership verification is diagnostic access, not delivery approval.

Scope and continuity

The update adds verification and monitoring evidence rather than a new mailserver release. No new paid service or mailserver configuration change was introduced by these checks. Existing hosting charges and operator responsibilities continue. Campaign sending remains paused pending operator review; no background campaign scheduler has been installed.

The original engineering sections follow the new comparison page. Their architecture, native tables, deployment preservation checks, recovery drill and operating limits are retained. Original measurements remain dated to the installation; the later checks do not retroactively repeat those tests.

Latest before / after

Comparison of the original 19 October installation with the additional evidence recorded on 21 October. The original project-level before/after table remains in section 01.

Side-by-side engineering changes

Area	Before · 19 October	After · 21 October
Authentication evidence	Five initial owner Gmail receipts passed SPF, DKIM and DMARC.	Sixteen owner diagnostics passed all three checks across three Gmail destinations; latest ten originals verify TLS 1.3.
External protocol review	Installation tests covered account access, submission and rejection cases.	Fresh review: 22 of 22 selected account, TLS, sender-identity, relay and size checks passed.
Operational send audit	Setup correspondence only; no prospect campaign within installation scope.	All 25 later initial attempts matched live Sent IDs: five per account; 23 remote accepted and two failed.

Google diagnostics	Postmaster ownership was not part of the original setup evidence.	Domain ownership verified. Traffic/reputation data remain insufficient for a delivery assessment.
DNS preservation	Mail records were added while preserving the original zone.	One verification TXT added: 23 existing records unchanged; 24 total, with two public resolver readbacks.
Evidence handling	Setup receipts and a private operator runbook.	Current reconciliation distinguishes attempted, accepted and failed messages; diagnostics excluded from campaign counts.
Scheduling and cost	Immediate SMTP/webmail; existing server reused.	Unchanged: no background campaign scheduler or new paid mail service; existing hosting costs continue.

The comparison separates implemented functionality, executed checks and unproven outcomes. The private operating record retains exact receipts and identifiers; this public case study excludes access material and correspondence metadata.

Self-Hosted Email Infrastructure

Engineering case study · 19 October 2026

Prepared by Zahidul Islam for clients and developers.

Original setup result · 19 October 2026

Five independent domain mailboxes now support authenticated sending, receiving and browser access on an existing virtual server. The implementation combines Docker Mailserver, Roundcube and Caddy, with Cloudflare managing DNS. Each mailbox has its own credentials and inbox. The work included external delivery tests, denial tests, dependency outages, an encrypted off-server backup and an isolated restore.

Owner-controlled Gmail tests verified end-to-end receipt and SPF, DKIM and DMARC authentication. A later reply was also sent through the actual webmail interface within an existing conversation. These observations establish the tested transport and authentication paths; they do not establish guaranteed delivery for new recipients.

What this shows

This was an integration and operations project using established upstream software. The engineering contribution covered discovery, service boundaries, configuration, account isolation, sender restrictions, failure handling, deployment checks, recovery tooling and handover. It did not involve writing a new SMTP server or claiming ownership of the upstream products.

The final deployment comparison matched twenty locally owned files, preserved seventeen existing reverse-proxy files and retained sixty-four unrelated running container identities. Those checks show a controlled change on a shared host at the measured time. They do not establish uninterrupted availability over a longer period.

Scope at publication

The original case-study sections describe the installation and verification completed on 19 October 2026. The latest update at the front records the additional checks completed on 21 October. Test correspondence used the owner's accounts. No prospect campaign was sent or scheduled as part of setup. The system has no campaign scheduler. Broader operating gaps, including external alerts, automatic off-server backup collection and image vulnerability scanning, remain explicit later in this report.

01 · Before and after

The starting point was a registered domain without a connected receiving mail provider. A domain purchase had been mistaken for a mailbox service because an earlier hosting package had bundled both. The result makes those responsibilities explicit.

Area	Before this project	Verified result
Receiving route	No domain MX records	MX points to the dedicated mail host
Mailbox accounts	No connected domain inboxes	Five separate authenticated inboxes
Outbound access	No configured mailbox submission	TLS SMTP submission on two supported ports
Browser access	No domain webmail	HTTPS Roundcube login, reading and sending
Sender identity	Authentication not established	SPF, DKIM and DMARC passed on five Gmail receipts
Reverse DNS	Generic provider hostname	IPv4 PTR matches the mail hostname
Sender restrictions	No installed mail controls	Relay and cross-account sender attempts rejected
Recovery evidence	No mailbox backup or restore	Encrypted PC copy and isolated five-account restore
Shared-host preservation	Existing workloads required protection	Prior proxy files and running container IDs preserved

A useful outcome with a clear boundary

The owner can use five real accounts through the same webmail application or configure them together in a mail client. Receiving aliases offer additional contact names without creating extra logins. Outgoing identity remains tied to the authenticated primary account.

There is no claimed sales, conversion or revenue improvement. Setup tests are not business traffic. Authentication checks and recipient-side delivery checks are recorded separately. The before-and-after result is functional infrastructure and a documented operating process, with deliverability still requiring observation.

02 · Discovery, requirements and alternatives

Separate the domain from the mailbox service

The first investigation checked the actual DNS inventory rather than assuming that registration included mailboxes. The zone contained eighteen records and no MX record. An Email Routing settings request was denied with the available API credential, so the report did not claim to have inspected that internal product configuration. The DNS result established the missing receiving route.

Cloudflare Email Routing handles incoming routing or forwarding; it is not the five-user IMAP mailbox service required here. Cloudflare also documents a separate outbound transactional email offering. These capabilities should not be confused with automatically receiving five conventional mailbox logins after buying a domain. The selected implementation hosts mailbox storage and authenticated submission independently of Cloudflare's routing product.

Requirements that drove the design

The owner needed both directions of mail, five separate accounts, browser access, standard mail-client support, recoverable credentials and instructions a future operator could follow. Existing applications on the server had to remain intact. The installation also needed practical checks for authentication, unauthorized relay, sender spoofing, delivery, data persistence and recovery.

The existing server already had substantial memory and disk headroom at preflight. Direct outbound SMTP connectivity was tested against real receiving servers before deployment. That avoided building a service around an unverified assumption that the hosting provider would permit outbound delivery.

Why this implementation

Docker Mailserver with Roundcube provided the required mail protocols and a familiar browser client without adding a hosted-mail subscription. A larger integrated suite was considered less suitable for this small installation. Hosted mail remained a reasonable alternative with lower operational responsibility, but free-plan eligibility and feature restrictions would have required separate account-specific checks.

Reusing the existing server reduced incremental service commitments. It also transferred patching, backups, reputation monitoring and outage recovery to the owner. This trade-off is central to the case: lower subscription overhead does not remove operational work or the existing server bill.

03 · Architecture and mail flow

Components and trust boundaries

Cloudflare provides the authoritative DNS records. The virtual server runs a dedicated Docker mail stack with persistent storage. Docker Mailserver supplies Postfix, Dovecot and Rspamd; Roundcube supplies the browser interface. Caddy terminates HTTPS for webmail and supplies the issued certificate material used by the mail services.

Internet sender → MX lookup → Postfix → filtering → recipient mailbox.

Mail client → TLS authentication → Dovecot → that account's inbox.

Browser → HTTPS Caddy → Roundcube → authenticated IMAP and SMTP.

Authenticated sender → submission checks → filtering and limits → outbound queue → recipient server.

These are distinct paths. A successful HTTPS request proves that the web front end responds; it does not prove that mail can leave the server, authenticate at Gmail or arrive in the correct inbox. Verification therefore followed the complete paths rather than stopping at container health.

Network exposure

Only the selected IPv4 mail listeners are published: server-to-server SMTP, authenticated submission and encrypted IMAP. Roundcube's application listener is bound to loopback behind the existing reverse proxy. Database, Redis and Rspamd administration are not public services. The stack uses a dedicated bridge network and does not mount the Docker socket or use host networking.

Persistent state is separate from the replaceable release configuration. Mailbox data, account hashes, queue state, signing keys, certificates and the webmail database must survive container recreation. A release-directory comparison and persistent-state checks support that separation.

Failure domain and ownership

The two containers share one virtual server. This is a small single-host installation, not high availability. A host outage can affect both webmail and mail handling. DNS hosting outside the server does not remove that failure domain.

The owner's mailbox service is the source of truth for messages and delivery queues. Campaign records are a separate source of truth for prospect selection, approvals, suppression and follow-up timing. Installing mail infrastructure does not merge those responsibilities or authorize a campaign.

04 · DNS, authentication and TLS

Publish a coherent mail identity

The deployment added a dedicated address record, an MX record, SPF, a DKIM public key and a DMARC monitoring policy while preserving the existing zone records. The mail address record is DNS-only. Ordinary Cloudflare web proxying was not applied to the mail protocols.

Reverse DNS needed a separate change at the server provider. The original provider hostname was replaced with the mail hostname for the IPv4 address used to send. The resulting forward and reverse identity was checked externally. IPv6 was not published or presented as tested mail transport.

SPF authorizes the selected sending infrastructure. DKIM signs messages with a domain-controlled key. DMARC checks alignment with the visible sender domain. The initial policy monitors rather than claiming to enforce rejection of all unauthenticated mail. Publishing these records and observing their actual evaluation are separate steps.

Verify at the receiving side

All five initial Gmail receipts reported SPF, DKIM and DMARC pass. That is stronger evidence than merely finding TXT records in DNS because it reflects evaluation of messages that traversed the real

outbound path. These receipts verify authentication for the tested messages, without guaranteeing the handling of future correspondence.

Protect client connections

The service offers implicit TLS SMTP on port 465, STARTTLS submission on 587 and encrypted IMAP on 993. External clients verified the certificate hostname and trust chain. Submission did not advertise authentication before STARTTLS on the tested clear connection. An invalid password was rejected.

Certificate maintenance checks the hostname, remaining validity and certificate/private-key match before copying renewed material. It compares both existing files, then reloads the active mail services when a change is required. A regression test covers an interrupted state where the certificate is current but the key still needs repair.

Successful certificate import is not proof of a future ACME renewal. The renewal source, timer and validation path are installed; the evidence records the maintenance checks actually executed during setup.

05 · Accounts, aliases and everyday use

Five credentials, one webmail application

Each mailbox uses its full email address as the username and has a different random password. The five accounts have separate inboxes. They share the webmail URL and server settings, not a single login. A desktop mail client can hold all five account configurations and present them in one interface.

Passwords, private keys, recovery material and live service details are maintained in restricted private records. They are intentionally absent from this public case study. Operators should retrieve them from the approved recovery store rather than placing them in chat messages, scripts, process arguments or campaign documents.

Receiving aliases direct selected role addresses to the primary mailbox. An alias is a delivery rule, not an additional independent inbox. The installed sending policy allows the authenticated primary identity; receiving an alias does not automatically grant permission to send as that alias or another account.

Sending is immediate

Roundcube and SMTP submit mail immediately. No campaign scheduler was installed. Maintenance timers import Caddy-renewed certificates, take backups and run health checks; they do not schedule prospect messages. This distinction prevents a future operator from labelling an unscheduled MIME file or ordinary SMTP submission as a scheduled campaign.

The optional local command helper defaults to read-only status and inbox operations. Its explicit send command uses a reviewed message, a stable Message-ID and a local journal. It distinguishes local server acceptance from remote delivery and saves a Sent copy. A duplicate or uncertain submission is not blindly repeated.

Follow-up continuity

A reply needs a new Message-ID while referring to the original message through In-Reply-To and References. Reusing an old Message-ID to represent a new follow-up would confuse deduplication and delivery reconciliation. The runbook explains this along with checking replies, hard bounces and removal requests before another contact.

These workflow instructions help the next operator use the service accurately. They do not replace recipient verification, campaign-specific approval, applicable marketing rules or a durable scheduler where timed background execution is required.

Update · 2 November 2026: scheduled sending at the recipient's local time, caps, a do-not-contact list and single follow-ups are now provided by the private mail hub described in the latest update. Webmail remains available.

06 · Abuse controls and dependency failures

Restrict the identity that can send

The mail service checks both the SMTP envelope sender and the visible From header against the authenticated primary mailbox. External tests rejected attempts to submit under another account's identity. Unauthenticated relaying and delivery to an unknown local mailbox were also rejected. These are observed controls for the tested cases, not a claim that every mail abuse technique has been evaluated.

Message size, attachment size and mailbox quotas bound ordinary use. Resource limits constrain the two containers on the shared host. Rspamd's rate rules limit bursts and refill over time, with separate global and authenticated-user controls. A sixth message in the tested global burst was rejected.

Token buckets are not exact daily campaign counters. They should not be described as enforcing the owner's new-contact and follow-up allowance. The campaign ledger must separately track actual sends, due follow-ups and suppression. Five accounts also share domain and IP reputation; additional usernames do not create independent reputations.

Test what happens when controls are unavailable

An induced Redis outage exposed a material issue: the installed rate-limit module logged its dependency error but initially allowed the authenticated message path to continue. The fix was a small reproducible patch tied to the exact upstream source hash, mounted into the pinned image. It changes that authenticated error path to a temporary rejection.

The final Redis failure test returned SMTP 451. A separate filter outage returned a temporary 454 response. The dependencies were restored after each test. Temporary rejection matters because the client can retry later while the service avoids treating an unavailable control as successful validation.

Maintain the exception carefully

The patch is an integration responsibility, not an upstream security certification. A future image upgrade must review whether the patch is still required, whether its source match is valid and whether both

failure tests still behave correctly. Failure drills must run in an appropriate maintenance window because they intentionally interrupt the new stack's filtering dependencies.

07 · Deployment on a shared server

Establish the baseline before changing the host

Preflight captured existing containers, network allocations, proxy configuration, available resources and required port availability. At the recorded point the server had about 17 GiB of available memory and 215 GiB of free disk. These are dated observations, not a capacity benchmark or a promise of future headroom.

An initial network allocation failed because Docker's default address pools were exhausted. The response was to inventory existing subnets and routes and explicitly assign an unused private subnet to the new stack. Existing networks were not pruned or recreated to make room.

Make the release reproducible

Docker Mailserver 16.0.1 and the Roundcube 1.7.4 nonroot image were pinned by digest. Environment-specific values are owned by central settings and environment configuration. Deployment tooling checks the existing live-file hash before replacing a locally owned file and stops if the state differs from the expected baseline.

The mail container has a 2 GiB memory limit and 1.5 CPU limit; webmail has 384 MiB and 0.5 CPU. Process limits and no-new-privileges are configured. Roundcube runs as an unprivileged service user with capabilities dropped. The mail image retains the privileges required by its normal service entry point; the report does not claim both containers run as nonroot.

Verify the resulting change

Final parity checks matched all twenty deployed files included in the release manifest. All seventeen pre-existing Caddy files retained their hashes, and all sixty-four unrelated container identities remained running and unchanged. This provides concrete preservation evidence beyond a statement that the other applications appeared to work.

These checks have limits. Matching files do not measure every application journey, and preserving container identities does not establish a long-term availability SLO. No broad load test, million-user claim or unrelated application release was part of the mail installation.

08 · Verification results and delivery limits

The tests combined external protocol clients, owner-controlled Gmail correspondence, real browser use and an isolated restored service. Setup traffic was the owner's own test correspondence, not customer or prospect traffic.

Check	Observed result
Independent IMAP authentication	Five of five accounts passed

Invalid credentials	Rejected
Unauthorized relay and sender identity	Rejected in tested cases
External outbound authentication	SPF, DKIM and DMARC pass on all five receipts
Owner-controlled Gmail receipt	Five authenticated test messages received
Replies to the five accounts	Each arrived in its intended separate inbox
Real webmail conversation reply	Received in Gmail Inbox
Dependency failure behavior	Redis 451; filter 454 temporary rejection
Isolated restored accounts	Five of five authenticated with inbox data
Maintenance regression checks	Four tests passed

Read the evidence at the right level

SMTP acceptance, remote delivery, authentication and recipient-side handling are different outcomes. A reply in an existing conversation can behave differently from a new first-contact message. This small test set cannot establish a reliable delivery rate across recipients or providers.

The fresh-context source review passed the installed five-account functionality within the tested scope. Of seven original acceptance groups, five were fully verified and two remained partial. The partial groups concern broader interface coverage and operational checks described later; they were not silently converted into passes.

Python compilation, Compose validation and direct security scans supplemented the real flows. Bandit reported no medium or high findings in the selected helper scope, and Gitleaks found no real secrets in the scanned source. These tools do not replace access-control tests, reputation monitoring or a complete security assessment.

09 · Backup design and isolated recovery

Capture a consistent, useful snapshot

The backup routine stops only the new mail stack, archives persistent state and release configuration, then restarts the services through a finally path. This trades a short planned interruption for a consistent small installation snapshot. It avoids pretending that copying live queue and database files necessarily provides a usable recovery point.

The snapshot includes mailbox contents, queue/state data, account hashes, DKIM material, TLS material and the Roundcube SQLite database. Runtime FIFOs and device entries are excluded. The archive is encrypted with AES-256-CBC using PBKDF2, with a separate HMAC-SHA256 checked before decryption. This describes the implemented scheme; it is not a general recommendation to design a custom backup cryptosystem.

Cleanup regressions cover archive, encryption and restart failures so plaintext archives are removed on those error paths. One encrypted snapshot was copied to the owner's PC and checked using its authenticator and hash. Private recovery keys are kept outside the public documentation.

Prove the restore with actual services

The restore drill used a separate directory and a temporary container without public ports or external networking. It authenticated all five restored accounts, found their real inbox messages and validated SQLite integrity. A real Roundcube PHP process running as its normal unprivileged user also completed a database write and rollback.

The tiny snapshot drill took 9.1 seconds. That measurement applies to the tested data and restore procedure. It is not a full host-replacement recovery time, and it does not include acquiring a replacement server, restoring DNS or rebuilding every operating dependency.

Recovery targets remain targets

The operating targets are a 24-hour recovery point and a two-hour recovery time. Automatic collection of off-server copies was not installed. One verified PC copy does not establish continuous off-server protection, and daily local snapshots do not survive every host-loss scenario. Sustained RPO and full-disaster RTO remain unproven until the owner completes and operates those additional processes.

10 · Problems encountered and lessons

Network allocation failed on a busy host

The first Docker network creation failed because the default allocation pools had no room. It was a shared-host inventory issue, not a reason to delete other applications' networks. Existing routes and allocations were checked before selecting a dedicated unused subnet. The final preservation comparison confirmed the unrelated workloads remained in place.

Lesson: inspect the shared environment before adding infrastructure, and make resource ownership explicit rather than depending on default allocation.

A limiter's dependency error allowed sending

The initial Redis outage showed that a working rate limit during normal operation did not prove safe behavior when its storage dependency failed. The upstream module's error path needed a temporary rejection for authenticated sending. A narrow, source-hash-checked patch and a real SMTP failure test established the corrected behavior.

Lesson: test dependency loss at the user-visible boundary; an error log alone is not evidence that a request was rejected.

An archive was insufficient without service-user checks

Early restore attempts encountered transient mail runtime files, offline startup assumptions and webmail database ownership. Safe extraction and service-specific ownership mattered as much as successfully decrypting the archive. The final drill ran the actual mail and webmail runtimes, including the unprivileged database write check.

Lesson: a restore is only useful when the real services and their actual users can use the recovered state.

Authentication and delivery need separate evidence

Owner-controlled Gmail tests verified authentication and receipt. The later webmail reply was recorded as a separate test condition. Original receipt metadata was retained so observations could be compared without treating different message types as interchangeable.

Lesson: distinguish transport, authentication and reputation outcomes. Report an inconvenient delivery result directly and improve the operating process from that evidence.

11 · Operating model, costs and next steps

Daily ownership and handover

The owner has an account registry, private access record, maintenance and recovery runbook, configuration source and verification receipts. Current workflow files link to those records so a future operator can discover them without reconstructing the conversation. Existing Gmail accounts and their queues were preserved.

Before any authorized outreach, the operator should check inboxes and Junk, delivery failures, refusal/removal requests, suppressions and prior contact history. The intended first real outreach date was 20 October; setup tests do not start the introductory period. The recorded plan allows five new contacts per account per day initially and ten later, with only genuinely due single follow-ups. These are planning limits, not a promise of safe volume or inbox placement.

Installed checks and remaining work

TLS and health checks run on five-minute timers; a daily backup timer is installed. Health checks cover service state, disk headroom, backup age and queued mail older than fifteen minutes. Actual maintenance invocations succeeded during setup. Future scheduled execution and a real certificate-renewal event were not observed in this test window.

External alert delivery, automatic off-server backup collection, full-disaster recovery measurement and image vulnerability/SBOM checks remain outstanding. Desktop webmail login, reading, composing, sending and logout were exercised; full mobile, keyboard and accessibility coverage remains incomplete. No claim of comprehensive security, compliance, high availability or unrestricted production readiness is made.

Economics and review triggers

The project added no paid mail provider, new hosting subscription or paid relay. Existing server costs continue, along with operator time, storage growth and maintenance responsibility. Reusing spare capacity should not be described as free hosting.

Consider a managed provider when operational effort, recovery needs or delivery requirements exceed what the owner can maintain. Reassess capacity and rate controls using actual traffic and queue behavior before increasing volume. If background scheduling is required, build and review a durable scheduler separately; immediate SMTP submission is not an adequate substitute.

Update · 2 November 2026: a durable scheduler was built and reviewed separately as the private mail hub. It runs as a second container on the same server with no new paid service; see the latest update for its checks and limits.

12 · Evidence, responsible use and references

Evidence boundary

The original installation metrics come from the 19 October 2026 receipts: external protocol tests, Gmail authentication, dependency-failure responses, backup authentication, isolated runtime restore and local/live comparisons. The dated 21 October update adds later verification and message-reconciliation evidence. Private receipts retain the exact commands, state and recovery details. They are not included here because they contain operational identifiers or private correspondence metadata.

This is a public account of the engineering work, not an invitation to access the service. Passwords, keys, owner test addresses, IP addresses, private network allocations, internal release paths and raw provider/account identifiers are excluded. The account count, component choices, failure modes and scoped results remain useful without publishing access material.

Responsible operation

Working mailboxes do not grant permission to contact every recipient. The operator must assess the applicable rules for the actual campaign and destination. The FTC guidance addresses truthful identity and subjects, commercial-message requirements and opt-outs. The ICO explains that UK business marketing rules depend on the recipient and data involved. This installation does not certify any campaign's legality or regulatory compliance.

All five accounts represent the same owner. Their shared domain/IP reputation must not be treated as five independent reputations or as a way to evade sending restrictions. Refusals and removal requests belong in a maintained suppression process, and an uncertain delivery outcome should be reconciled before retrying.

Upstream and regulator references

Cloudflare Email Service: routing and sending are separate capabilities

Docker Mailserver documentation

Official Roundcube container project

Google email sender guidelines

FTC commercial email guidance

ICO business-to-business marketing guidance

References checked 19 October 2026. Software and provider behavior can change; the dated installed versions and executed evidence define this report's scope.