



Zbix CRM Operations and Maintenance Guide

Product release 1.2.0

Zbix Technologies

Documentation edition 1.2.0-r1

28 October 2026

Monitoring, backup verification, recovery preparation, maintenance and practical troubleshooting.

Use the contents to find a task, then review its prerequisites and expected result. Screenshots show demonstration data and preserve the original interface. Your permissions, organization layouts and enabled capabilities can change the controls you see.

Private review edition. Source-reviewed procedures have not been executed against a CRM installation as part of this documentation project. Platform acceptance and provider configuration require their stated qualification.

Contents

Collect useful support evidence	3
Read System Health and verify service readiness	3
Operational logging and support evidence	7
Routine application maintenance checklist	8
Troubleshoot queued failed or uncertain email	9
Troubleshoot attachment upload and download	10
Troubleshoot import validation and execution	11
Troubleshoot report totals and missing records	12
Prepare a safe support case	13
Understand recovery points and backup services	13
Prepare backup access keys and storage	15
Create an encrypted recovery point	17
Verify a recovery point independently	18
Test a recovery point in isolation	18
Schedule recurring recovery points	20
Configure and validate backup storage	21
Manage recovery point retention	22
Prepare a CRM upgrade	23
Apply an approved application upgrade	24
Plan recovery and rollback decisions	25
Monitor services jobs and recovery readiness	26

Collect useful support evidence

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

A useful support request identifies the operation, expected behavior and safe evidence needed to reproduce the issue. It should separate user access problems from service or provider failures.

Help administrators and support engineers diagnose the issue without disclosing credentials or changing records unnecessarily.

Before you begin

- Permission to inspect the affected page.
- Administrator help for protected health, job and audit evidence.

Procedure

1. Record the date/time and timezone, module/page, operation and current record or job state. Describe the expected result and what actually happened.
2. Capture the exact displayed error text and a screenshot after removing confidential customer information and authentication material.
3. Record the application release/package version and, for administrators, the schema and runtime identity shown by System Health. Distinguish missing identity fields from a verified match.
4. Check the relevant troubleshooting article. For access problems, compare effective permissions and record visibility; for queued work, inspect worker heartbeat and execution history.
5. For email or Microsoft synchronization, include safe provider status, attempt outcome and whether the outcome is uncertain. Avoid automatic retries when a remote write may have succeeded.
6. Escalate through your organization's approved support channel with the evidence and steps already attempted. Preserve existing records, volumes and logs until the owner approves a recovery action.

Expected result

Support receives enough context to investigate while protected information and existing application state remain intact.

Important considerations

- Do not include passwords, TOTP codes, recovery codes, API tokens, cookie values, private keys, raw environment files or unredacted diagnostic exports.
- Do not treat a successful health probe as proof that every worker, provider and recovery path is functional.
- No support email address or service-level promise is inferred by this documentation. Use the contact route in your service agreement.

Troubleshooting

- Issue cannot be reproduced: provide exact filters, timezone, role scope and the sequence of actions.
- Only one user is affected: check that user's active status, session, module role, team and provider connection before altering system settings.

Related knowledge base articles: Read System Health and verify service readiness; Module Permissions; Create and send Email Activities.

Read System Health and verify service readiness

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

System Health provides administrator-visible evidence for database reachability, worker heartbeats, queue state, private storage, backup status and recent HTTP counters. Runtime identity reports the source commit, image identities and database migration when supplied by the installation. These checks describe different layers and must be interpreted together.

Use the health page for routine checks and incident triage before attempting corrective service or data changes.

Before you begin

- Administrator identity.

- The intended release and the time range of the incident.

Procedure

1. Open Administration > System Health and choose Refresh system health. Record the checked time and warnings in the private operations log.
2. Confirm database reachability, then examine worker last-seen and last-success times. The backend considers a running heartbeat healthy when it is no more than 90 seconds old.
3. Inspect queued and failed counts and the oldest due-work age for the affected operation. Open the corresponding job or activity history for details.
4. Review storage availability, writability and free capacity. Escalate a storage warning before uploads, imports or recovery work exhaust the volume.
5. Review runtime identity against the release plan. Treat unreported fields as unavailable evidence rather than proof of a match.
6. Inspect Backup & Recovery for artifact verification, isolated restore and destination evidence. A legacy receipt alone does not establish recoverability.

Expected result

The operator has a dated health assessment with specific follow-up actions for the affected component.

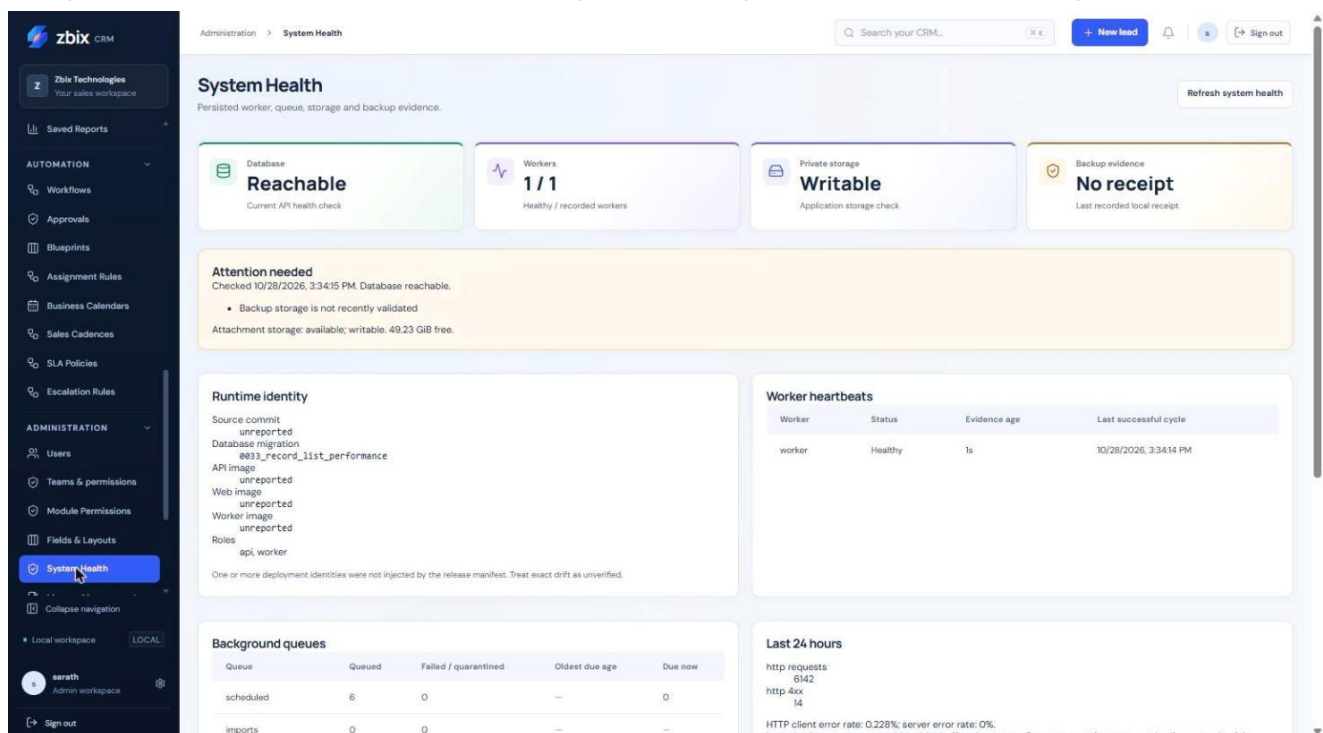
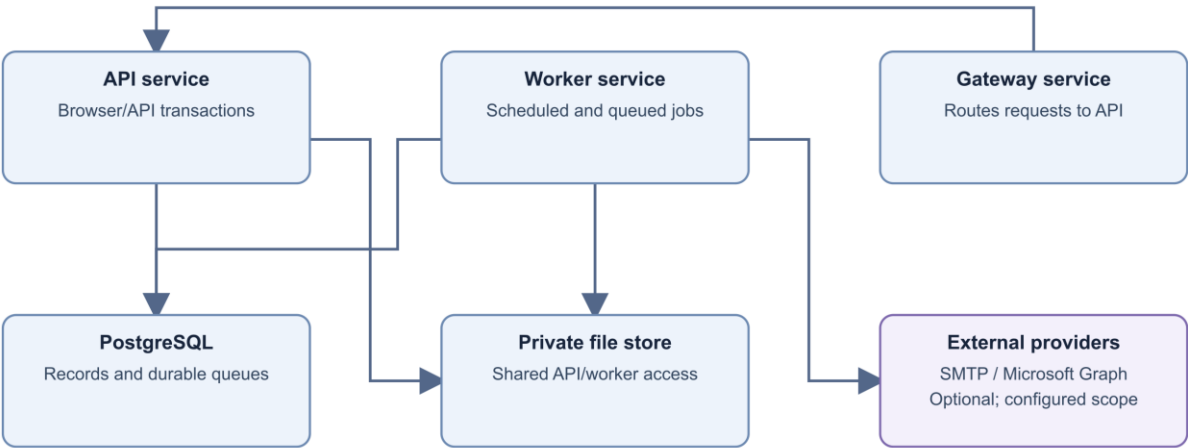


Figure 1 System health dashboard SS-043

Service dependencies

Dependencies to check when only part of the application is healthy

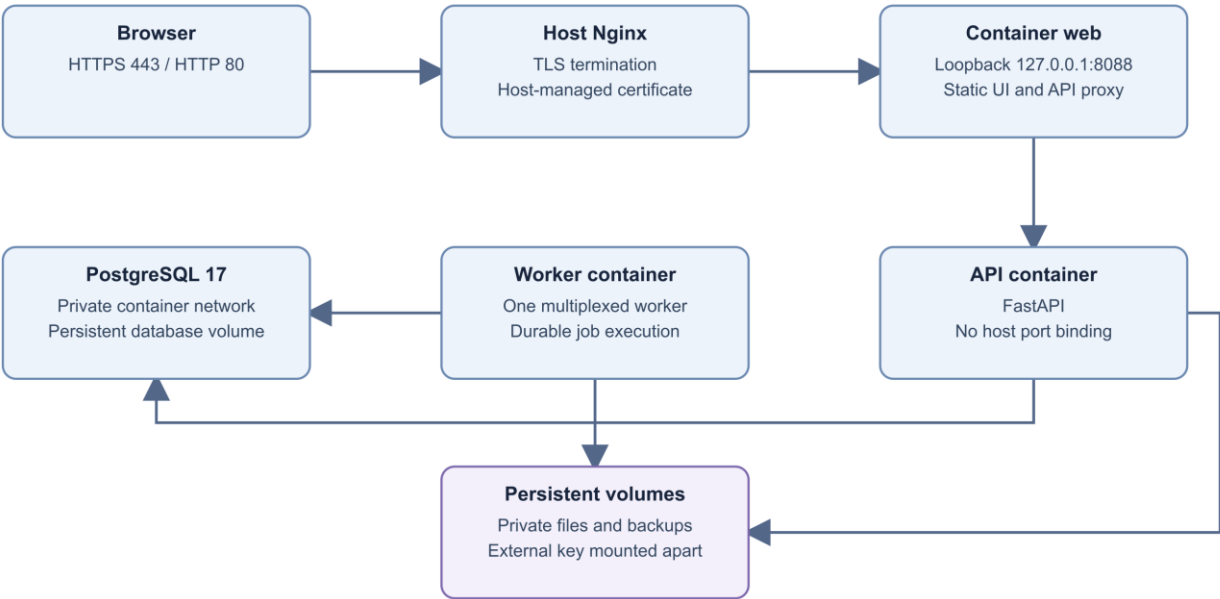


Database readiness does not prove worker progress or provider delivery.
Provider availability depends on configured operation and actor authorization.

Figure 2 Service dependencies

Linux deployment topology

Source-qualified Ubuntu amd64 package; one host

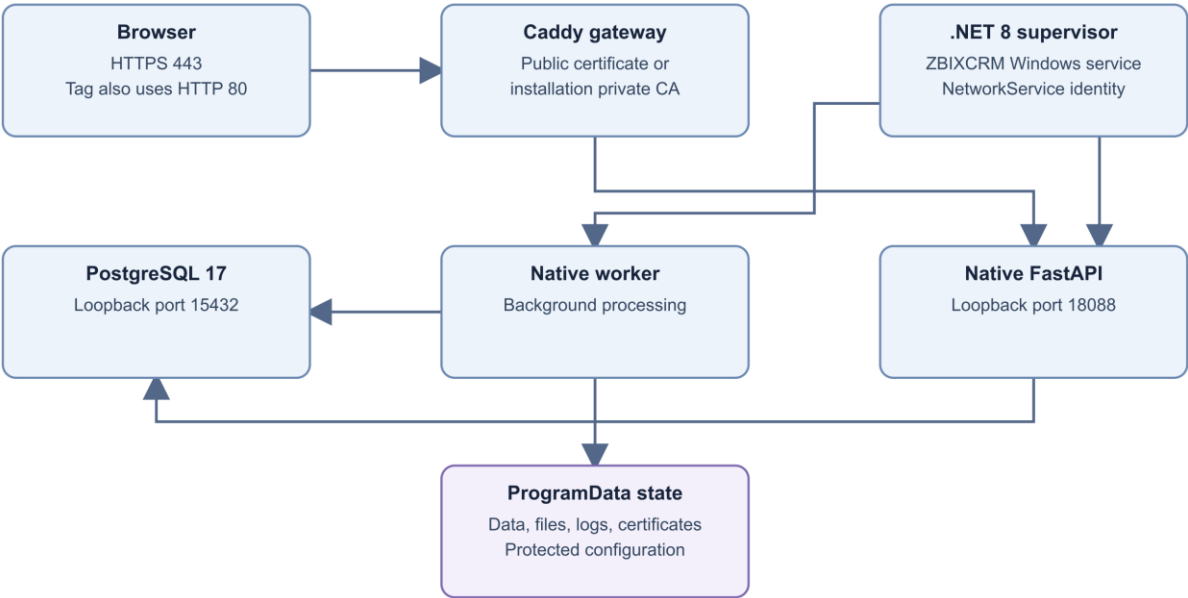


systemd wraps Docker Compose; database and API remain private.
Optional provider egress is configured separately. No HA or external broker is established.

Figure 3 Linux deployment topology

Native Windows topology

Tagged v1.2 runtime; supervised native processes

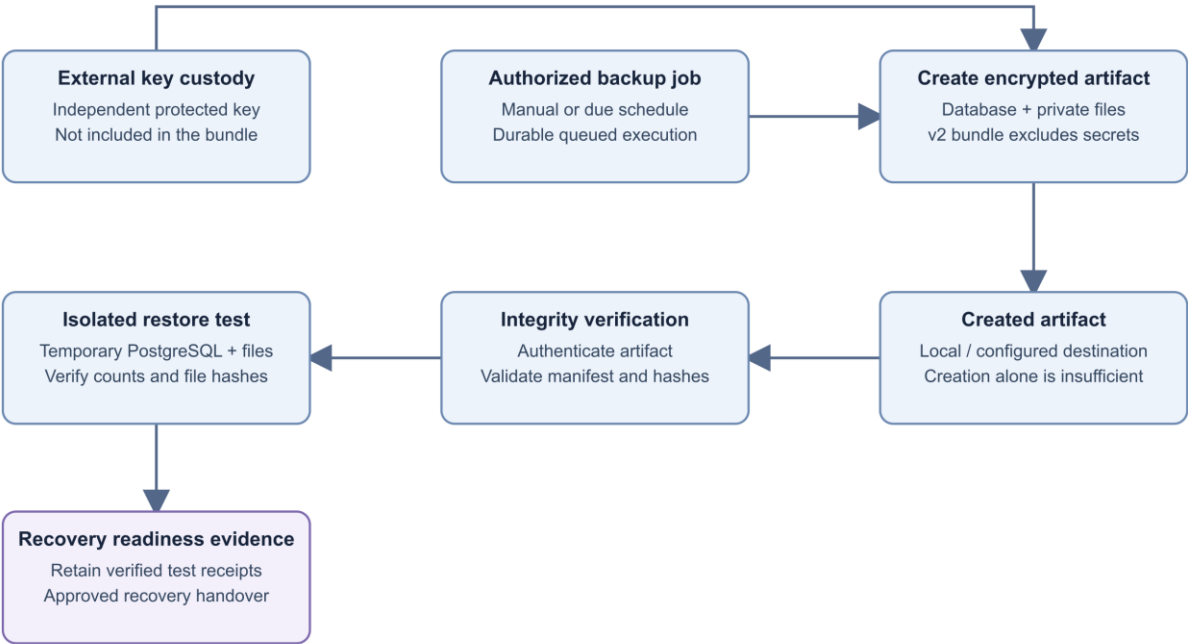


Supervisor also starts PostgreSQL and Caddy; its links illustrate process supervision.
Optional port 80 / IIS coexistence correction is post-tag, not qualified by v1.2 source.

Figure 4 Native Windows topology

Backup verification and recovery readiness

Separate creation, integrity verification and isolated restore evidence



Created, integrity verified and restore tested are distinct states; no live restore UI is provided.
Keep independent key custody and off-host copies; same-host storage is not disaster recovery.

Figure 5 Backup verification and recovery readiness

Important considerations

- GET /api/health is liveness; GET /api/ready checks database readiness. Neither alone proves background work or an external provider works.

- HTTP percentages are derived from bounded recent counters; absent request samples do not establish zero errors.
- Healthy backup status depends on several recorded prerequisites. Do not replace a degraded status with an assumption based on a file existing.

Troubleshooting

- The health page is denied: identity administration access is required.
- Database is reachable but jobs are delayed: investigate worker heartbeat, due work and operation-level history.
- Runtime identity is unreported: have the deployment owner reconcile the artifact and configuration privately.

Related knowledge base articles: Background processing and durable jobs; Operational logging and support evidence; Routine application maintenance checklist.

Operational logging and support evidence

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

The application records safe operational telemetry and durable business audit history for different purposes. HTTP telemetry tracks generated request identifiers, matched route templates, status and duration. Worker errors use bounded categories. Business histories record actor and operation context. These sources help diagnose incidents without routinely recording raw secrets or full request bodies.

Use this article to prepare a useful support case and preserve operational evidence without exposing authentication or customer data.

Before you begin

- Authorized access to the relevant application history or service logs.
- An incident time, timezone, affected operation and product release.

Procedure

1. Reproduce the issue once if safe, then record the visible safe error and the time. Avoid repeated mutations when the first outcome is unknown.
2. Inspect the record, document, import, communication or workflow history appropriate to the issue. Note the last successful state and failed attempt.
3. Use System Health to record worker and queue context around the same time.
4. Collect sanitized service logs through the installation's documented log command. Retain request identifiers and error categories; remove customer content, tokens, reset links and configuration values.
5. Escalate with a concise reproduction, expected and actual result, release identity, affected module and redacted evidence. Keep internal identifiers in the private support channel only when needed.

Expected result

Support receives enough context to correlate a failure while sensitive content remains protected.

Important considerations

- The application server uses app.server to install safe Uvicorn logging. Package service definitions should retain that entry point.
- Proxy access logs deliberately avoid raw request targets, and raw proxy error logs may be suppressed. A failure before the API can therefore require gateway-level diagnosis.
- Telemetry buffers are bounded during database outages; prolonged outages may lose older detailed counters.
- Audit history is application evidence, not a claim of an externally certified or database-administrator-proof audit system.

Troubleshooting

- No application request appears for a failed page load: check the gateway/network layer and correlate the time.
- A generic server error provides little detail: retain its request context and inspect safe service categories rather than enabling uncontrolled payload logging.

Related knowledge base articles: Read System Health and verify service readiness; Backend architecture and application services; Prepare a safe support case.

Routine application maintenance checklist

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

Routine maintenance combines service health, storage capacity, failed work, access review and recovery evidence. Zbix CRM exposes these controls through System Health, operation histories and administration pages. A running web page is only one part of operational readiness. Maintenance should preserve business records and use the package-specific upgrade and recovery runbooks for disruptive work.

Use this checklist during an agreed operations review and before handing the installation to another support team.

Before you begin

- Administrator access and an approved maintenance responsibility.
- The release manifest, recovery owner and provider ownership records.

Procedure

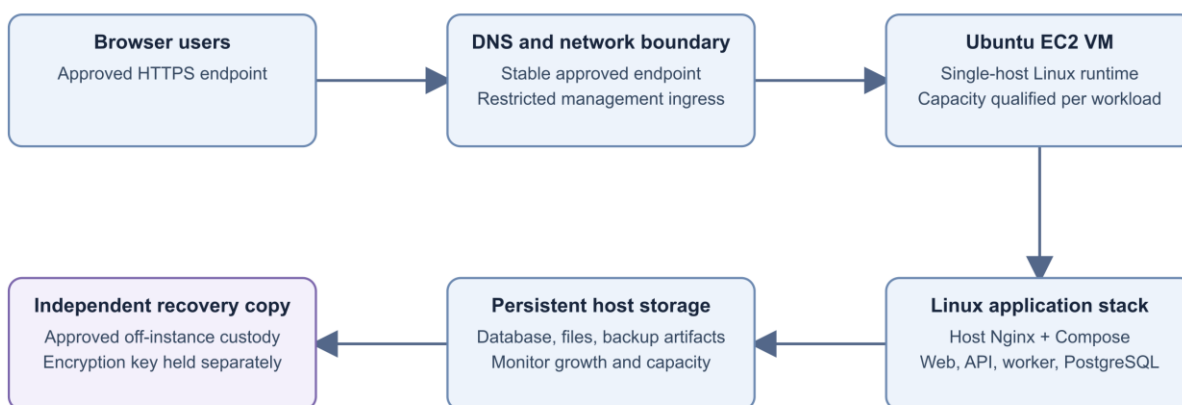
1. Refresh System Health and review database, worker heartbeat, due queues, storage and runtime identity. Record unresolved warnings with an owner.
2. Review failed imports, email attempts, workflow executions and billing runs. Classify definite failures separately from uncertain external outcomes before retrying.
3. Review Backup & Recovery for recent verified recovery points, destination validation and isolated restore-test evidence. Confirm separate recovery-key custody with the designated owner.
4. Review inactive users, role assignments and unexpected sessions. Confirm required service identities remain active and narrowly scoped.
5. Check SMTP/Microsoft connection status and expiry or reconnect notices where these integrations are used. Do not send test messages to real customers as a routine health probe.
6. Before an upgrade, follow the release-specific preparation and isolated rehearsal process. Afterward, verify the intended migration and representative read-only business views.

Expected result

The maintenance record identifies current health, outstanding work, recovery readiness and accountable owners.

AWS EC2 deployment context

Recorded Linux deployment uses the same self-hosted package

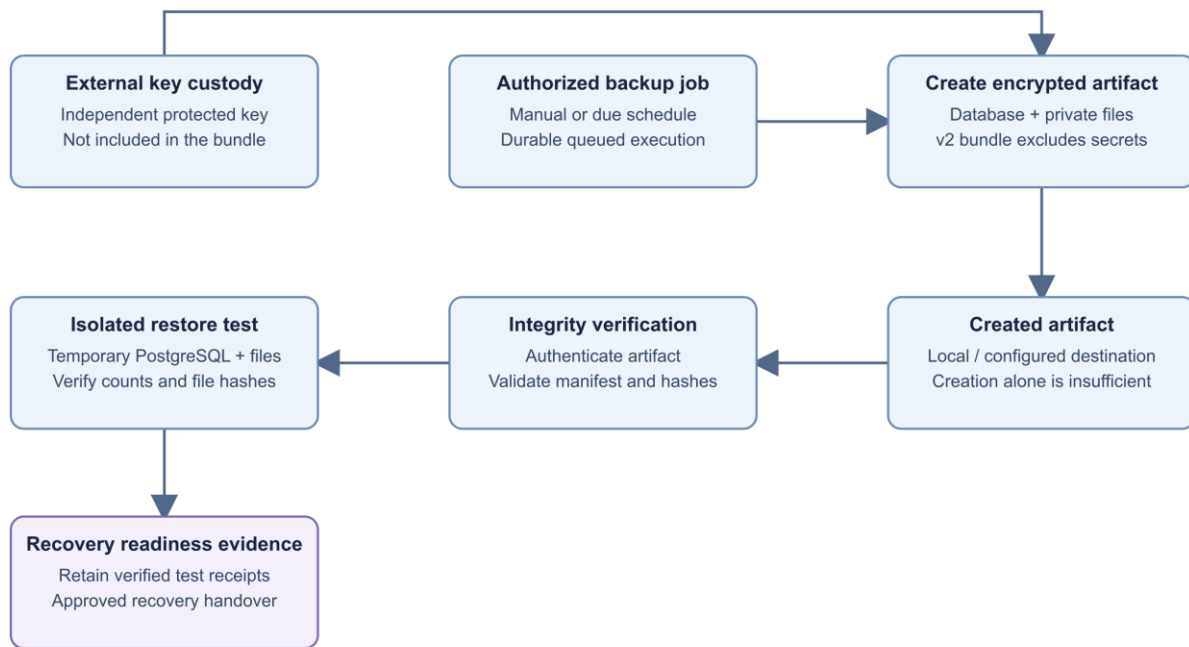


No managed load balancer, RDS, HA cluster or managed SaaS is established by this topology.
Off-instance copy and key custody require deployment-specific operational validation.

Figure 6 AWS EC2 deployment context

Backup verification and recovery readiness

Separate creation, integrity verification and isolated restore evidence



Created, integrity verified and restore tested are distinct states; no live restore UI is provided.
Keep independent key custody and off-host copies; same-host storage is not disaster recovery.

Figure 7 Backup verification and recovery readiness

Important considerations

- The product does not establish a universal maintenance interval, RPO or RTO. Set those through the customer's approved operating policy.
- Do not remove database or attachment volumes to resolve a routine incident.
- Import source retention and external backup retention require deliberate operational review.
- A healthy local status does not replace independent service monitoring and recovery exercises.

Troubleshooting

- Repeated failures remain after a restart: inspect operation history and configuration dependencies rather than restarting indefinitely.
- Storage is approaching capacity: identify the responsible data class and use an approved retention process.

Related knowledge base articles: Read System Health and verify service readiness; Background processing and durable jobs; Operational logging and support evidence; Prepare a safe support case.

Troubleshoot queued failed or uncertain email

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

An Email Activity can be a draft, queued work, a provider-accepted message, a definite failure, an uncertain external outcome or a simulation. These states require different responses. SMTP and Graph sends use the stored message and recheck permissions at execution. Local mock receipts never prove an external email was delivered.

Use this procedure when a user says a message did not arrive or a queued activity is not progressing.

Before you begin

- Access to the Email Activity and its related CRM record.
- Administrator/provider support for transport or mailbox evidence when required.

Procedure

1. Open Email Activities, find the message and inspect its provider, status, scheduled time and attempts. Confirm the recipient and related record are the intended ones.

2. If Draft, the message has not been queued. If scheduled for the future, wait for the due time unless an authorized user changes the operation.
3. If Queued and overdue, inspect System Health worker and communication queue evidence. Check the owner remains active and can access the related record.
4. If Failed, review the safe failure and whether the outcome is uncertain. Use Retry failed send only when offered for a definitely failed attempt.
5. If Accepted, inspect provider trace or the controlled recipient mailbox. Check spam/quarantine and recipient addressing before sending a replacement.
6. If Simulated / Mock, create and review a real-provider send only when external sending is intended and configured.

Expected result

The operator identifies whether the problem is queue processing, permissions, transport acceptance, recipient delivery or a simulated operation.

Important considerations

- A stale worker claim or network loss can leave an uncertain result. The provider may have accepted the message even when CRM did not receive confirmation.
- Graph attachments are bounded to 3 MB total in the adapter. Provider and application quotas can also prevent sending.
- Graph sending is capped at 25 queued messages per user and 100 per organization in 24 hours in this implementation.
- Marking a Quote Sent records a commercial state and does not itself transmit an email.

Troubleshooting

- Repeated definitely failed attempts: resolve the provider or permission cause before another reviewed retry.
- Unknown provider outcome: escalate with sanitized attempt time, provider and message identity through private support; do not include message bodies or tokens.

Related knowledge base articles: [Configure SMTP email delivery](#); [Connect Microsoft 365 mail and calendar](#); [Background processing and durable jobs](#); [Prepare a safe support case](#).

Troubleshoot attachment upload and download

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

Attachment access depends on both the attachment action and visibility of its parent CRM record. New files enter a pending scan state. A download can remain blocked even though its filename appears in the record. File format validation and malware scan evidence are separate checks; an unavailable scanner is not a clean result.

Use this procedure for rejected uploads, quarantined files and missing downloads while preserving original evidence.

Before you begin

- Access to the parent record and appropriate attachment permissions.
- The original file, its type and size, without exposing its confidential contents in a ticket.

Procedure

1. Open the parent record and its Attachments tab. Confirm that the record itself is accessible to the affected user.
2. For upload failures, compare the file type and size with the supported types and installed limit. Use a simple filename of at most 200 characters.
3. Review the displayed scan state. Pending requires worker progress; unavailable or error requires scanner/service diagnosis; infected files must remain quarantined.
4. Ask the administrator to inspect System Health and scan diagnostics when processing does not advance. Verify storage availability as well as worker activity.
5. After an authorized scan or storage repair, refresh the attachment and try the normal download action. Do not retrieve private file bytes through an exposed server directory.
6. Escalate persistent failures with the parent module, attachment name/type/size, scan state, time and safe error.

Expected result

The cause is identified as permission, content/size validation, scan evidence or storage integrity, with the original file preserved.

Important considerations

- PDF, DOCX, XLSX, PNG, JPG/JPEG, TXT and CSV are allowlisted, subject to content checks. Renaming an unsupported file does not make it valid.
- Production release requires matching clean evidence and actual content. A local compatibility allowance is not a general production bypass.
- Pending, infected and error states do not become downloadable merely because an administrator can view the record.
- Deletion is a recorded operation with later retention cleanup; use the supported control instead of removing storage objects manually.

Troubleshooting

- Download returns not found: check both record visibility and attachment permission before assuming the bytes are absent.
- Scanner unavailable: resolve the configured scanning service and reprocess through the supported operation.
- Checksum/evidence mismatch: preserve the artifact and escalate for integrity review.

Related knowledge base articles: Private files and generated document storage; Authorization and permission evaluation; Read System Health and verify service readiness; Prepare a safe support case.

Troubleshoot import validation and execution

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

Import Jobs stages CSV/XLSX source data, maps columns, validates a preview and requires explicit confirmation before writing CRM records. Validation creates no CRM records. Execution commits bounded batches, so cancellation or a later error does not undo rows already imported. Retry resumes interrupted work; it does not blindly rerun successful rows.

Use this guide to resolve invalid previews, stalled jobs and partial outcomes without duplicating records.

Before you begin

- The original uploader or another user with the required scoped job access and action permissions.
- A retained source file and the intended target module: Leads, Accounts, Contacts or Products.

Procedure

1. Open Administration > Import Jobs and inspect the job phase, processed counts and safe error. Determine whether it is parsing, mapping, validating, awaiting confirmation or importing.
2. If mapping is incomplete, map only supported target fields and select the intended duplicate handling. Validate again and review the current generation.
3. Review invalid and duplicate rows before confirming. Download error report CSV when available and correct the source rather than ignoring rejected relationships or required fields.
4. If confirmation is stale, refresh and validate the latest mapping. Changing mapping invalidates the previous preview token.
5. For a stalled or failed job, inspect worker health and current uploader permissions. Use Retry only for interrupted failed/cancelled work that the interface allows.
6. For completed row failures, prepare a corrected new file containing the intended unresolved rows. Reconcile created, updated, skipped and failed counts before another import.

Expected result

The import has a reviewed preview or a reconciled execution outcome, with committed records and failed rows clearly distinguished.

Important considerations

- Background jobs cap at 10,000 rows and 50 columns; the upload byte limit defaults to 10 MiB. Legacy synchronous imports retain a 1,000-row limit.
- XLSX formulas, error cells, macros, external links and unsafe archive content are rejected. Store dates as ISO date text.
- Mapping becomes immutable after execution starts. Cancellation waits for the active bounded batch and stops later batches.
- A repeated upload of the same source can return the existing job; this prevents accidental recreation.

Troubleshooting

- Ready preview cannot confirm: refresh after mapping changes and review the current validation result.
- Some rows imported before cancellation: reconcile the execution counts; cancellation is not rollback.
- Repeated data errors: fix the source and upload a corrected file rather than retrying unchanged failed rows.

Related knowledge base articles: Background processing and durable jobs; API validation and safe concurrency; Private files and generated document storage; Prepare a safe support case.

Troubleshoot report totals and missing records

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

Reports and saved reports run against the current user's authorized data. Sharing a report definition does not grant the owner's record access. Currency amounts remain separated, and dashboard date filters have metric-specific meanings. Differences between users or between a list and a dashboard can therefore be expected rather than evidence of lost data.

Use this procedure to reconcile a reported discrepancy before changing records or reporting permissions.

Before you begin

- The report name, current filters, selected currency and user context.
- Access to the underlying records needed for reconciliation.

Procedure

1. Record the report type, date range, owner/team filters, pipeline/stage and displayed currency. Apply the filters and refresh before comparing results.
2. Compare the same report as the affected user. Review effective module and source-record permissions if another user sees different totals.
3. Check the metric definition. Dashboard filters distinguish creation-based populations from closed opportunities using their close dates; do not assume all cards use one date field.
4. Compare amounts within one currency. Do not sum separate currencies or overlapping team groups as an organization total.
5. For exports, confirm export permission on both the saved report and its source module, and narrow the selection if it exceeds the 10,000-row bound.
6. Escalate a remaining discrepancy with a small, sanitized record sample and the exact filters and metric definition.

Expected result

The comparison uses equivalent scope, dates, currencies and permissions, exposing a reproducible discrepancy only when one remains.

Important considerations

- Team memberships can overlap, so team totals need not add to the organization total.
- Weighted pipeline is a management projection using configured probabilities, not recognized revenue or a promise of cash collection.
- A saved report stores a definition; its result changes with current data and current access.
- Record-list summary cards may describe all accessible records while table filters narrow the visible rows. Read the page's scope note.

Troubleshooting

- CSV is denied: review both source and report export grants.
- A report becomes empty after role changes: inspect source visibility and parent relationships before recreating it.
- Historical trend differs from current pipeline: compare closed-period metrics with current open-stage metrics, which measure different populations.

Related knowledge base articles: Authorization and permission evaluation; CRM and commercial data relationships; API validation and safe concurrency; Prepare a safe support case.

Prepare a safe support case

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

A useful Zbix CRM support case identifies the affected release, operation, user scope and time, then provides the smallest reproducible example. Existing records, operation histories and health evidence usually provide more useful context than a screenshot alone. Sensitive configuration and authentication material must remain outside the case.

Use this checklist when a documented resolution does not restore the expected behavior or when an uncertain external operation requires review.

Before you begin

- A reproducible symptom or a preserved failed operation.
- Permission to share the chosen demonstration or sanitized evidence with the support recipient.

Procedure

1. State the symptom, expected result and business impact. Include the module, visible action and whether the issue affects one user or several.
2. Record the product release and database migration from available approved evidence. If runtime identity is unreported, say that explicitly.
3. Provide the incident date, time and timezone. Include the safe error text and request identifier when available.
4. Describe the steps already taken, including refresh, permission review, job history and health checks. Identify any uncertain email or calendar outcome before further retries.
5. Attach a screenshot only after checking it for customer data, account recovery links, API tokens, authenticator secrets and private infrastructure information.
6. Keep the source record and operation history intact until the case is resolved. Agree on any test, retry or data correction before applying it to customer records.

Expected result

Support can reproduce or correlate the issue with a clear account of what is known, what remains uncertain and what actions have already occurred.

Important considerations

- Never include .env contents, passwords, bearer tokens, private keys, recovery codes or full reset URLs.
- A demonstration screenshot can show an empty state accurately; it should not be replaced with fabricated records.
- Use an isolated test environment for mutation-based reproduction where feasible.
- A public forum or knowledge-base article is not an appropriate destination for private operational identifiers.

Troubleshooting

- The issue cannot be reproduced: preserve the original time and operation identity, then compare logs/history instead of repeatedly making changes.
- Support requests a destructive action: verify a recovery point and obtain the required operational approval before proceeding.

Related knowledge base articles: Operational logging and support evidence; Read System Health and verify service readiness; Troubleshoot queued failed or uncertain email; Troubleshoot import validation and execution.

Understand recovery points and backup services

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

Backup & Recovery coordinates durable jobs for encrypted recovery-point creation, independent integrity verification and isolated restore testing. It distinguishes these states because a completed backup job alone does not prove that the artifact can be restored.

Understand which components must be available and which evidence is needed when assessing recoverability.

Before you begin

- Access to Backup & Recovery appropriate to your responsibilities.
- Knowledge of the installation's database, private-file storage, destination and external key custody.

Procedure

1. Identify the source PostgreSQL instance and private-file store protected by the self-hosted deployment. The backup runtime uses a consistent exported PostgreSQL snapshot and captures referenced private files.
2. Identify the destination and recovery encryption key. The key is supplied independently of the backup artifact and must remain recoverable if the application host is lost.
3. Follow a recovery point through its states: creation, independent integrity verification and isolated restore test. Review the associated durable job and receipt for each operation.
4. Understand the restore-test boundary. It creates a temporary database and file area, compares recovered data and hashes, and does not start a customer-facing application or external integrations.
5. Separate self-hosted recovery from provider-managed recovery. The ordinary worker does not execute a managed cloud database snapshot or point-in-time recovery simply because provider metadata is present.

Expected result

Operators can explain the protected data, supporting services, external dependencies and the meaning of each recovery state.

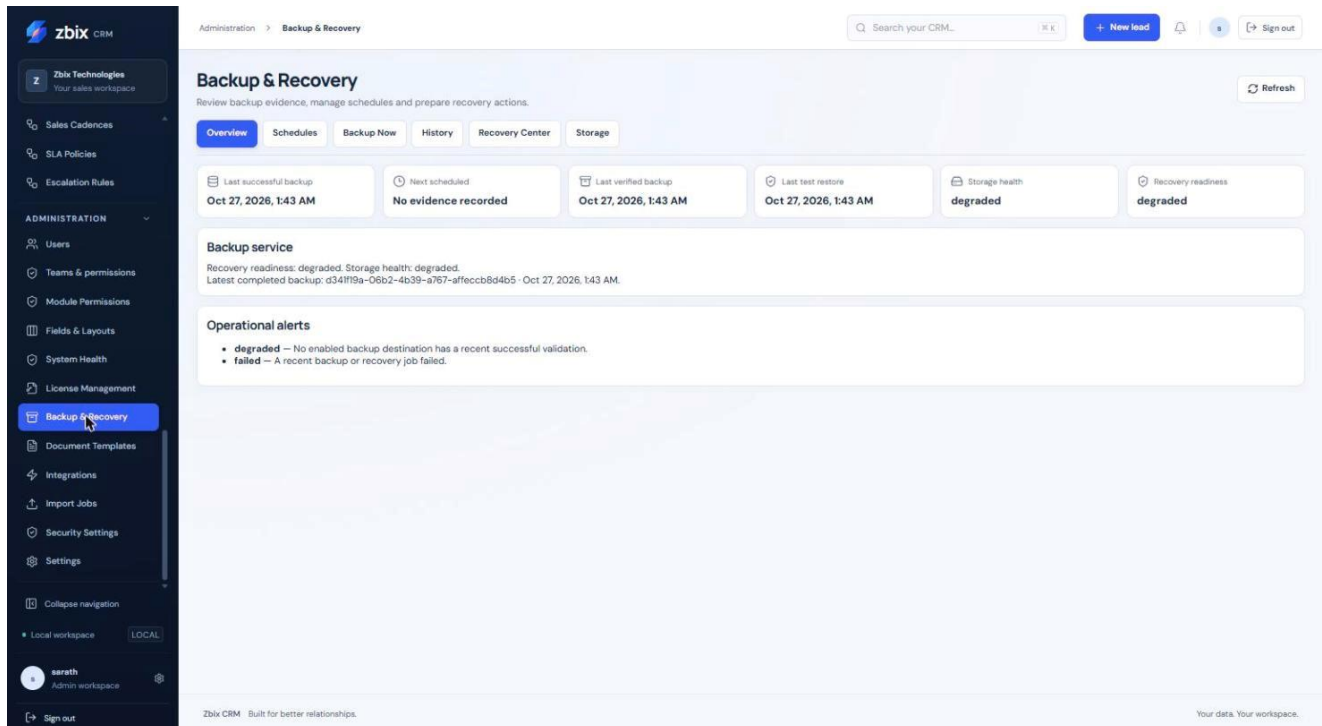
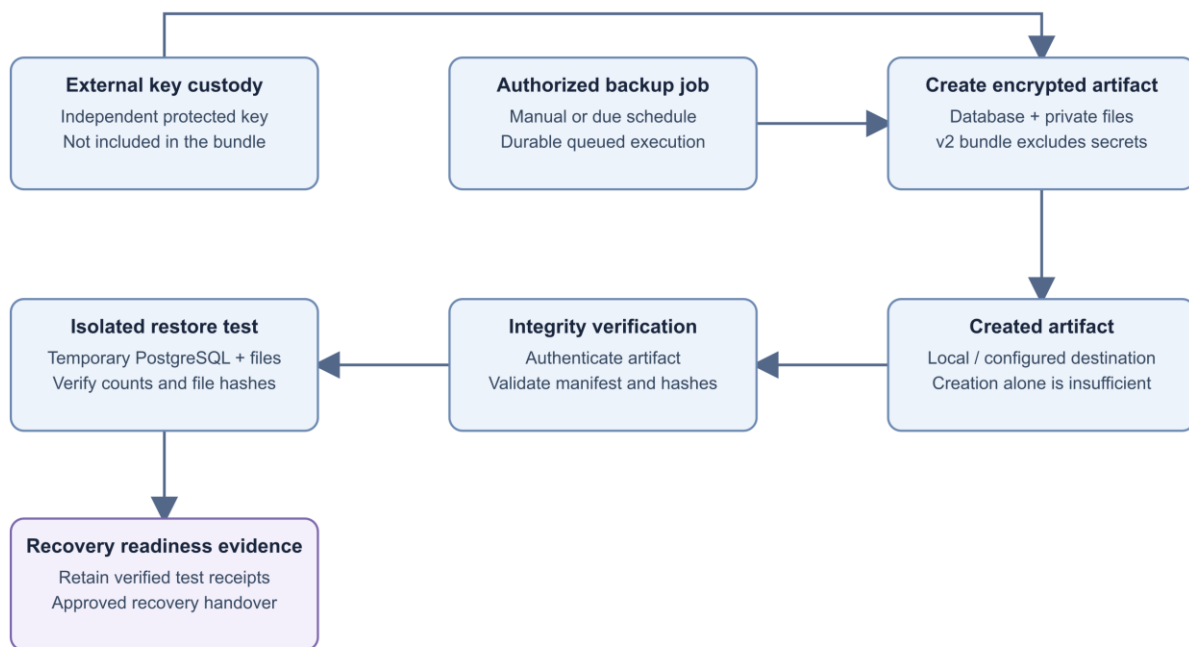


Figure 8 Backup & recovery overview SS-045

Backup verification and recovery readiness

Separate creation, integrity verification and isolated restore evidence



Created, integrity verified and restore tested are distinct states; no live restore UI is provided.
Keep independent key custody and off-host copies; same-host storage is not disaster recovery.

Figure 9 Backup verification and recovery readiness

Important considerations

- The v2 recovery configuration records required external secrets but does not include their values. Database credentials, MFA and communication keys, integration signing material and the backup recovery key need independent custody.
- Local and S3-compatible destinations are implemented. Managed database/PITR, Azure, GCS and SFTP entries do not establish connected provider recovery.
- No live restore action is available in CRM. Production recovery is a separately authorized operator process.
- The encrypted bundle is memory-bound. Capacity and recovery objectives require deployment-specific qualification.

Troubleshooting

- If Overview shows a successful backup but no verified or tested point, complete the missing operation and inspect its receipt.
- If an artifact is available but its required key is not, escalate to the recovery-key custodian. Do not replace keys and assume old artifacts will remain recoverable.
- If a provider operation remains a handoff request, confirm the deployment's actual connected recovery service rather than repeatedly queuing the request.

Related knowledge base articles: Prepare backup access keys and storage; Create an encrypted recovery point; Verify a recovery point independently; Test a recovery point in isolation; Assess managed SaaS capabilities.

Prepare backup access keys and storage

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

Recovery operations need more than access to the Backup & Recovery page. The installation needs a working worker, protected storage, an independently recoverable encryption key and the specific permissions for each action. Prepare recovery operations without assuming that administrator status grants every backup privilege.

Before you begin

- A named backup operator and an independent recovery/key custodian.
- An approved storage destination and enough capacity for staging, verification and isolated restore work.

- An authorized plan for protecting all external secrets required to recover the installation.

Procedure

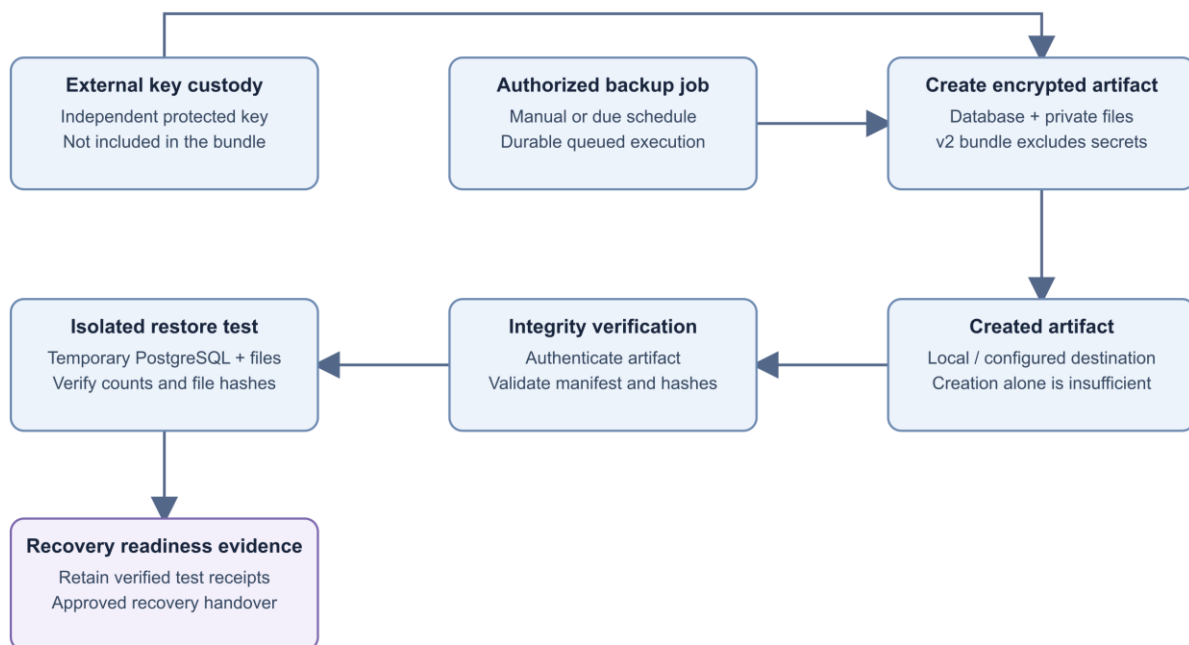
1. Review the user's effective backup permissions. Default administrators can view the dashboard and history, manage schedules, start manual backups and verify artifacts. Destination management, artifact download and test restore require the appropriate explicit grant or recovery-operator authorization.
2. Confirm PostgreSQL, the private-file store and the worker are healthy. A ready API does not prove that the worker can execute queued recovery jobs.
3. Have the deployment operator configure the backup encryption key securely outside the artifact. Record its custody and recovery procedure without copying the value into an article, ticket or screenshot.
4. Protect the additional external secrets required by the installation. The v2 artifact records that these are required but intentionally excludes secret values.
5. Create or select an approved destination and validate it in Storage. For remote storage, confirm the approved credential reference and TLS configuration through the deployment operator.
6. Check capacity and arrange an isolated restore target. Define who reviews verification and restore receipts and who responds to backup failure alerts.
7. Run a manual recovery-point cycle before enabling a recurring schedule. Keep the schedule paused until creation, integrity verification and isolated restore testing succeed.

Expected result

The authorized operator can create a recovery point and the recovery owner can independently verify and rehearse recovery with the required keys and storage.

Backup verification and recovery readiness

Separate creation, integrity verification and isolated restore evidence



Created, integrity verified and restore tested are distinct states; no live restore UI is provided.
Keep independent key custody and off-host copies; same-host storage is not disaster recovery.

Figure 10 Backup verification and recovery readiness

Important considerations

- Read, download, destination management and restore permissions are separate. A visible button does not override a server-side permission check.
- The recovery key is not a substitute for the application's other durable encryption keys.
- Use an approved off-host destination and separate key custody to protect against host loss. A same-host destination alone is insufficient.

Troubleshooting

- For permission denied, ask the authorized recovery administrator to review the specific requested operation; do not broaden unrelated CRM permissions.
- For missing-key or destination errors, correct server-side configuration through the deployment operator and then validate again.
- Escalate with the operation, job identifier, error code and time. Do not send key values or raw environment files.

Related knowledge base articles: Configure and validate backup storage; Create an encrypted recovery point; Test a recovery point in isolation; Schedule recurring recovery points.

Create an encrypted recovery point

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

Backup Now submits a durable job to the worker and shows its persisted progress. The resulting encrypted artifact is a recovery-point candidate until independent verification and an isolated restore test have established the corresponding evidence.

Capture the database and referenced private files before an important change or as part of the approved backup routine.

Before you begin

- The start manual backup permission and access to Administration > Backup & Recovery.
- Healthy database, private-file storage and worker.
- A configured recovery key, validated destination and adequate free space.

Procedure

1. Open Administration > Backup & Recovery and review Overview for existing recovery alerts. Resolve storage or key problems before starting a new job.
2. Open Backup Now. Select the approved Destination, or use the service default only when the deployment owner has confirmed what that default is.
3. Choose Start backup once. The screen records submission; it is not an immediate success confirmation. Follow the persisted job status in Backup Now or History.
4. If the job is queued, confirm the worker is running. When it is running, monitor the reported progress without assuming that an absent percentage means failure.
5. After completion, record the artifact identifier, size, storage location and completion time. Review the verification state separately; creation may be followed by an automatic verification job.
6. Complete independent verification and an isolated restore test according to the related procedures before using the point as evidence for an upgrade or recovery decision.

Linux package alternative; select the authorized administrator

```
sudo zbix-crm backup --email admin@example.com
```

Expected result

History contains a completed backup job linked to an encrypted artifact, with the next verification and restore-test steps clearly identified.

Important considerations

- The page disables another manual start while an active job is shown. Refresh and inspect the existing job before submitting duplicate work.
- Required external secrets are not included in the v2 recovery artifact. Preserve them through the separately approved custody process.
- A timeout in the packaged CLI may leave a job running; inspect its persisted status before retrying.

Troubleshooting

- If the job stays queued, check worker heartbeat and queue processing in System Health.
- If creation fails, inspect the job's error code and stage. Common operational areas to check are destination capacity, key availability, PostgreSQL access and referenced-file availability.
- Do not delete a previous verified artifact to conceal a new failure. Escalate with the job identifier and sanitized diagnostic details.

Related knowledge base articles: Prepare backup access keys and storage; Verify a recovery point independently; Test a recovery point in isolation; Monitor services jobs and recovery readiness.

Verify a recovery point independently

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

Verification checks the encrypted artifact and its manifest independently of the creation operation. A successful receipt changes the evidence available for that specific artifact; it does not prove that a complete application can be recovered within a particular time.

Establish artifact integrity before an isolated restore test or an upgrade recovery decision.

Before you begin

- The verify backup permission.
- A completed artifact that belongs to your organization.
- Access to its destination and the correct recovery key.

Procedure

1. Open Administration > Backup & Recovery > History and identify the artifact produced by the intended backup. Check its time and storage location so that you do not verify a different point accidentally.
2. Open Recovery Center and locate that artifact. Choose Verify. If automatic verification already completed, review that receipt and the current artifact state before requesting another run.
3. Follow the Verification operation in History. Review the job status and any error code; a submitted or running job is not a successful verification.
4. After completion, confirm the artifact is integrity verified and that the verification time refers to the intended point.
5. Retain the verification result with the change or recovery record and proceed to Test restore when the operation requires recoverability evidence.

Linux package alternative; replace the artifact placeholder with the approved ID

```
sudo zbix-crm verify-backup --email admin@example.com --artifact-id APPROVED_ARTIFACT_ID
```

Expected result

The selected artifact has a successful independent integrity-verification result and is eligible for an authorized isolated restore test.

Important considerations

- Verification and restore testing are different operations. Neither a checksum listed in a manifest nor an uploaded object alone is a completed restore rehearsal.
- Use the same artifact identity throughout creation, verification and restore-test records.
- Preserve failed evidence for investigation rather than relabeling or replacing it with a successful result from another artifact.

Troubleshooting

- For an unavailable artifact, confirm its destination, catalog identity and storage access. Do not substitute a similarly named file without verifying provenance.
- For an integrity or decryption failure, preserve the object and contact the recovery owner. Check that the correct independently held key is available.
- Escalate the artifact and job identifiers, stage, time and error code. Never send the recovery key.

Related knowledge base articles: Create an encrypted recovery point; Test a recovery point in isolation; Configure and validate backup storage; Plan recovery and rollback decisions.

Test a recovery point in isolation

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

Test restore rehearses database and private-file recovery in a separate temporary target. The runtime compares recovered table counts and file hashes and records a receipt. It does not replace the running CRM database or start external integrations.

Establish that an integrity-verified point can be recovered by the implemented isolated test before relying on it for an upgrade or recovery plan.

Before you begin

- Explicit test restore permission or authorized recovery-operator access; ordinary administrator defaults do not grant this action.
- An integrity-verified recovery point and access to its correct key and destination.
- Sufficient capacity and database privileges for a separate temporary restore target.

Procedure

1. Open Administration > Backup & Recovery > Recovery Center. Read the displayed recovery scope, impact and required credentials before choosing an artifact.
2. Select the point whose identity and verification receipt match the approved change or recovery record. Confirm integrity verification has succeeded.
3. Choose Test restore and monitor the Test restore operation in History. Do not interpret a queued request as a successful rehearsal.
4. Allow the worker to create its isolated database and file area, restore the point and compare recorded counts and hashes. The operation also checks supported migration and licensing-installation metadata.
5. Review the completed result and artifact verification state. Record the restore-test time, artifact identity and receipt with the recovery plan.
6. If the rehearsal fails, preserve the last known usable recovery point and resolve the failure before an upgrade or destructive maintenance decision.

Linux package alternative; requires a verified artifact and authorized restore access

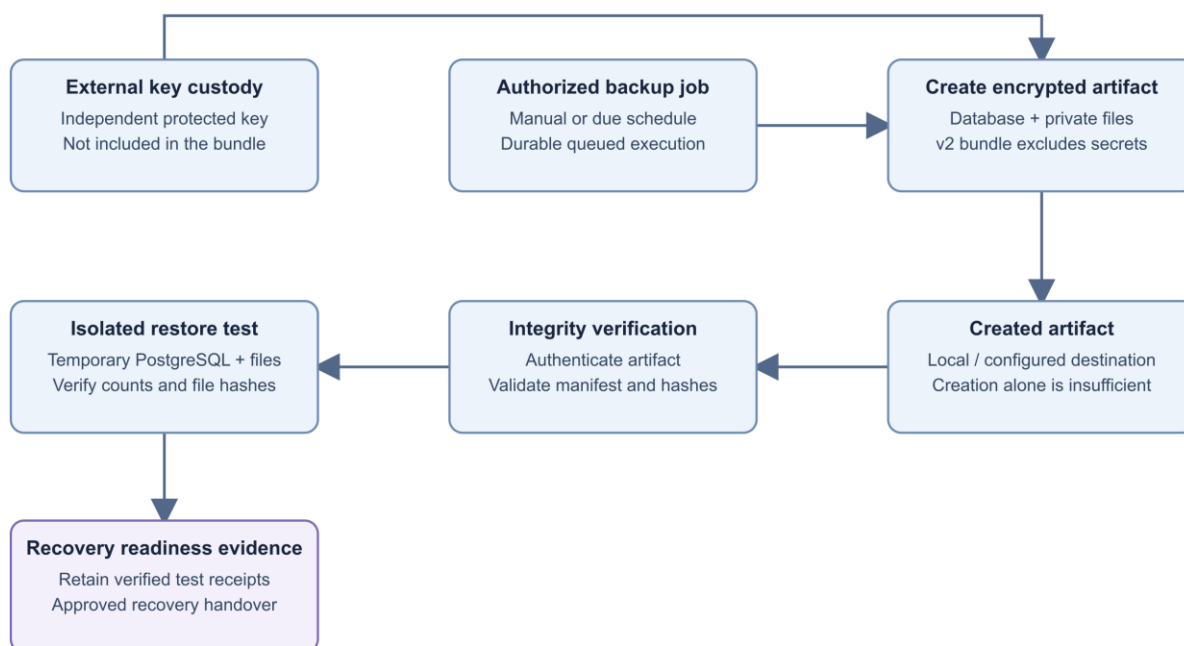
```
sudo zbix-crm test-restore --email admin@example.com --artifact-id APPROVED_ARTIFACT_ID
```

Expected result

The selected artifact is marked restore tested after a successful isolated rehearsal with a persisted result.

Backup verification and recovery readiness

Separate creation, integrity verification and isolated restore evidence



Created, integrity verified and restore tested are distinct states; no live restore UI is provided.
Keep independent key custody and off-host copies; same-host storage is not disaster recovery.

Figure 11 Backup verification and recovery readiness

Important considerations

- This test validates the implemented database/file restore checks. It is not a full user-acceptance exercise, an external integration test or a production RTO guarantee.

- No live restore button is provided. Production recovery requires an independently reviewed operator procedure and authorization.
- The runtime cleans up only its owned temporary target. Do not manually remove another database or storage area while investigating a failed test.

Troubleshooting

- If the operation is denied, review the dedicated restore permission with the recovery administrator.
- If it reports that verification is required, complete the independent verification for that exact artifact.
- For schema, file or database errors, preserve the receipt and involve the recovery owner. Record package/schema identities without including credentials.

Related knowledge base articles: Verify a recovery point independently; Prepare backup access keys and storage; Prepare a CRM upgrade; Plan recovery and rollback decisions.

Schedule recurring recovery points

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

The Schedules tab stores daily, weekly and monthly backup schedules with an IANA timezone and daily/weekly/monthly retention counts. New schedules created in the interface are saved paused, allowing the operator to review them before enabling execution.

Automate the approved backup routine after manual creation, verification and isolated restore testing have succeeded.

Before you begin

- The manage schedules permission.
- An enabled, validated destination and a working backup worker.
- Approved key custody, capacity, retention and failure-monitoring arrangements.

Procedure

1. Open Administration > Backup & Recovery > Schedules and review existing schedules to avoid duplicate coverage.
2. Enter a descriptive Schedule name and select the intended Destination. Choose Daily, Weekly or Monthly. For Weekly select the weekday; for Monthly enter the day of month.
3. Set Local time and a valid IANA Timezone, for example Africa/Kampala. Review the next occurrence in the schedule list rather than relying on your workstation timezone.
4. Set Daily recovery points, Weekly recovery points and Monthly recovery points according to the approved retention policy. Save schedule.
5. Confirm the saved schedule is Paused, with the expected destination, frequency and next-run information. Complete a manual backup cycle and resolve storage or recovery alerts before activation.
6. Choose Enable when the operational prerequisites are satisfied. After the first due occurrence, inspect History and confirm the scheduled job, its verification result and the expected destination.
7. Use Edit to change the schedule or Pause to stop future scheduling during an approved maintenance period. Review the next run after any timezone or time change.

Expected result

An enabled schedule produces durable backup jobs at the reviewed local-time cadence, and an operator monitors their actual outcomes.

Fields and controls

Field or control	Rule
Schedule name	Required descriptive name.
Frequency	Daily, weekly or monthly in the interface.
Timezone	Valid IANA name; not a numeric UTC offset.
Local time	24-hour HH:MM value.
Day of month	1-31; a shorter month uses its last valid day.

Field or control	Rule
Daily retention	1-3650 recovery points.
Weekly retention	0-520 recovery points.
Monthly retention	0-240 recovery points.

Important considerations

- The interface uses an interval of one for the chosen frequency and a single catch-up policy. It does not run every historical missed occurrence after downtime.
- Monthly days beyond the month's length are clamped to the last day. Daylight-saving transitions require deployment-specific validation; choose a suitable local time and review affected dates.
- A saved schedule is configuration evidence. Confirm a successful due run and separate verification rather than assuming scheduling proves recovery readiness.
- Failure-recipient metadata is not proof of external alert delivery. Test the actual monitoring route separately.

Troubleshooting

- If no run appears, check Enabled status, timezone, next run and worker health before creating another schedule.
- If the destination is absent from the selector, confirm it exists and is enabled.
- If a job runs but verification fails, follow the artifact failure rather than changing the schedule time to hide the alert. Escalate with schedule and job identifiers.

Related knowledge base articles: Prepare backup access keys and storage; Create an encrypted recovery point; Manage recovery point retention; Monitor services jobs and recovery readiness.

Configure and validate backup storage

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

The Storage tab supports managed local storage and S3-compatible storage. Destination configuration and validation are separate from successful backup, verification and restore operations. Remote storage credentials stay on the server and are referenced by name rather than entered as values in the page.

Choose an approved destination that can retain encrypted recovery points and support the required verification and recovery process.

Before you begin

- Permission to manage destinations or recovery-operator authorization.
- A storage owner, approved retention policy and sufficient capacity.
- For S3-compatible storage, an approved HTTPS endpoint and server-side credential environment reference.

Procedure

1. Open Administration > Backup & Recovery > Storage. Review existing destinations and their Validation and Capacity values.
2. To add managed local storage, enter Name, select Managed local storage and provide a Relative path beneath the configured backup storage root. Choose Save destination.
3. For S3-compatible storage, select that kind and enter HTTPS endpoint, Region, Bucket, optional Prefix and Credential environment reference. Have the deployment operator provision the referenced credential securely before validation.
4. Choose Validate for the destination and review the returned status. Correct connectivity, access, capacity or TLS issues before using it in a schedule.
5. Select the destination for a manual backup. Confirm artifact creation, then independently verify the uploaded/materialized artifact and complete an isolated restore test.
6. Document whether the destination survives loss of the application host. If it is local to the same host, arrange an independently approved off-host recovery strategy and separate key custody.

Expected result

The destination is validated and has a demonstrated recovery-point cycle, with a clear retention and custody owner.

Important considerations

- Managed local paths are relative to the configured backup root. Traversal, symlink escapes and unsuitable locations are rejected.
- S3 TLS verification is required for normal use. The application uses a server-side secret reference; do not paste access keys into article examples or database JSON.
- S3-compatible behavior was qualified against an isolated fixture. An actual provider account, permissions, lifecycle and recovery process need their own validation.
- Azure, GCS, SFTP and managed database/PITR schema kinds are not connected implementations in this release.

Troubleshooting

- If a local destination is rejected, check its relative path, write access and available capacity within the managed root.
- If S3 validation fails, have the operator check endpoint reachability, TLS, bucket access and the exact configured secret reference.
- If upload succeeds but later verification fails, preserve the artifact and review its identity/hash and retrieval permissions. Escalate without sending credential values.

Related knowledge base articles: Prepare backup access keys and storage; Create an encrypted recovery point; Verify a recovery point independently; Manage recovery point retention.

Manage recovery point retention

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

Retention uses the daily, weekly and monthly counts attached to a backup schedule. After successful integrity verification, the worker evaluates usable artifacts by day, week and month and records keep/delete decisions. Retention is not a substitute for off-host custody or restore testing.

Control local storage growth while preserving usable recovery history and respecting immutable destinations.

Before you begin

- An approved business retention policy and a storage owner.
- A schedule with reviewed daily/weekly/monthly retention counts.
- At least one independently verified and restore-tested recovery point.

Procedure

1. Determine how far back the business needs to recover and how many daily, weekly and monthly points it can store. Keep statutory or contractual retention requirements separate from unsupported compliance claims.
2. Enter the approved counts in Schedules and save the configuration. Review capacity before enabling a new cadence or increasing retention.
3. After a scheduled point is created and independently verified, review the resulting history and storage use. Retention operates on usable catalogued artifacts, not on arbitrary files in a directory.
4. Confirm that the newest usable point and artifacts involved in active verification or restore work remain protected.
5. Coordinate remote-object lifecycle and immutability with the storage provider. Do not assume the application will delete remote artifacts according to the same local policy.
6. Investigate cleanup failures before storage fills. Preserve the decision history and resolve the underlying permission or storage problem rather than deleting files outside the managed process.

Expected result

Storage growth follows the approved policy while protected usable points and active recovery work are retained.

Important considerations

- The retention grouping uses UTC day, ISO week and month buckets, even when the schedule is expressed in a local timezone.
- The newest usable recovery point, active verification/restore jobs and immutable destinations are protected.
- Application deletion is limited to catalogued local objects under the managed destination. Remote deletion remains provider controlled.
- A backup retained for a long period still depends on recoverable keys and a compatible tested recovery process.

Troubleshooting

- If disk use does not fall after a retention change, check immutability, active jobs, artifact usability and cleanup-failed status.
- If no recent verified point exists, restore the backup/verification process before reducing retention.
- Escalate storage pressure with current capacity, policy and failed operation details. Do not remove the newest usable point as an emergency shortcut.

Related knowledge base articles: Schedule recurring recovery points; Configure and validate backup storage; Verify a recovery point independently; Plan capacity for CRM workloads.

Prepare a CRM upgrade

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

An upgrade changes the application release and may apply forward-only database migrations. Preparation must establish the current identity, target identity, a usable recovery point and an approved response if migration or readiness fails.

Make a release change reviewable before interrupting service or changing the database.

Before you begin

- A release owner, database/recovery owner and approved maintenance window.
- The target package, trusted digest and release-specific upgrade guidance.
- Access to the current installation's version, schema, backup evidence and operational monitoring.

Procedure

1. Record the current package version, source identity where available and database migration. Record the target release and supported upgrade route. Resolve ambiguous version labels before proceeding.
2. Verify the target package and review its migration, configuration and operational changes. Keep the existing configuration, external keys and persistent storage protected.
3. Create a fresh recovery point from the current installation. Complete independent verification and an isolated restore test for that exact artifact.
4. Arrange an approved off-host copy or destination and independent key custody. Confirm the recovery team can retrieve what is needed if the host becomes unavailable.
5. Rehearse the target upgrade against an isolated restored environment when the approved change process requires it. Record preserved data, schema result, readiness, worker behavior and relevant business checks.
6. Agree the service interruption window, communications, validation steps and decision owner. Define when to remain stopped, recover, or perform a reviewed forward repair.
7. Retain the prior package and the recovery evidence. Begin the upgrade only after the prerequisite review is complete.

Expected result

The change record contains matching current/target identities, verified recovery evidence, validation steps and an approved failure response.

Important considerations

- A package download, database dump or recovery-point identifier alone does not prove recovery readiness.
- Do not replace encryption keys or remove volumes during an upgrade.
- Historical acceptance of one upgrade does not qualify every source/target version pair. The 27 October report documents a specific v1.2 upgrade.

Troubleshooting

- If a restore rehearsal fails, resolve it before cutover and preserve the last usable point.
- If release documentation conflicts, use the exact package/source identity and obtain release-owner clarification; do not combine instructions from old v0.x and v1.x packages.
- Escalate missing recovery keys, unsupported schema paths and unverified target packages before service interruption.

Related knowledge base articles: Apply an approved application upgrade; Plan recovery and rollback decisions; Create an encrypted recovery point; Test a recovery point in isolation; Complete the installation readiness review.

Apply an approved application upgrade

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

The Linux package provides an upgrade command that verifies package content and image identities, installs a versioned release, applies forward migrations and checks readiness. Native Windows updates use a different package lifecycle and must follow their release-specific qualification.

Execute the approved package upgrade and verify the resulting application and background services.

Before you begin

- Completed upgrade preparation with a fresh independently verified and isolated-restored recovery point.
- The approved target package and a release-specific supported upgrade path.
- An authorized maintenance window and operator access to the target.

Procedure

1. Confirm the current installation and intended target one final time. Check that the approved recovery artifact and required external keys are available.
2. For Linux, extract the approved target archive into a separate staging directory and verify its inner checksums and manifest. Use the target package's own bin/zbix-crm command so that its runtime files match the intended release.
3. Set the required recovery-point acknowledgement through the approved maintenance procedure, using the identifier whose verification and restore-test receipts were reviewed. The operator remains responsible for the validity of that recovery evidence.
4. Run the target package's upgrade command during the approved window. It stops API/worker writers, applies forward migrations, installs the updated service command and restarts the stack. Monitor the command outcome; do not assume completion from a new image being present.
5. Check status, version and readiness. Confirm the expected migration, worker heartbeat, storage access and representative authorized application operations. Review background queues for new failures.
6. For native Windows, use the publisher-approved update procedure for the exact package after a protected-state backup and rehearsal. The installer can stop/re-register the existing service, but that behavior alone does not qualify all Windows upgrade paths.
7. Record the resulting package/schema identities, validation results and any unresolved items. Reopen normal use only when the change owner accepts the result.

Post-upgrade Linux verification

```
sudo zbix-crm status
sudo zbix-crm version
curl -fsS http://127.0.0.1:8088/api/ready
```

Expected result

The approved target release is running with the expected schema and service behavior, or the installation remains controlled while the documented failure response is applied.

Important considerations

- The Linux installer requires an acknowledged recovery-point ID; it does not replace independent verification and restore testing.
- On Linux migration failure, the previous release setting is restored and services remain stopped. No automatic database downgrade occurs.
- Do not run the new-install procedure over an existing Linux installation or improvise a Windows upgrade based solely on retained ProgramData.

Troubleshooting

- If package or image identity checks fail, stop and obtain the correct package rather than editing integrity records.
- If migration fails, follow the approved recovery/forward-repair decision while services remain controlled.
- If readiness passes but worker or business checks fail, keep the change open and investigate the affected service. Escalate with sanitized command outcome and exact identities.

Related knowledge base articles: Prepare a CRM upgrade; Plan recovery and rollback decisions; Monitor services jobs and recovery readiness; Test a recovery point in isolation.

Plan recovery and rollback decisions

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

Zbix CRM separates isolated restore testing from live recovery. A failed upgrade may leave a database at a state that cannot safely be used by an older application image. Returning a release symlink or reinstalling binaries is therefore not a complete data rollback.

Give the recovery owner a controlled decision process for an upgrade failure or service loss without inventing a one-click live restore capability.

Before you begin

- An authorized incident/change owner and a recovery operator.
- A known recovery point with independent verification and restore-test evidence.
- Independent access to required external keys, approved packages and destination storage.
- A deployment-specific, reviewed live recovery procedure.

Procedure

1. Identify the incident scope: application-only failure, migration failure, database loss, private-file loss or host loss. Record the current package and known schema state before changing anything else.
2. Preserve the existing state and relevant sanitized logs where safe. Stop writes only through the approved operational procedure, and avoid deleting volumes or overwriting a failed target prematurely.
3. Choose between a reviewed forward repair and recovery to a known point. Confirm the data-loss window by comparing the recovery-point time with the last accepted business activity.
4. Validate artifact provenance, integrity/restore receipts, source and target identity, available keys and target isolation. Do not select an artifact solely because its filename looks recent.
5. Carry out live recovery only under the approved deployment-specific runbook. CRM exposes test restores and recovery guidance, not a live overwrite action. Provider-managed recovery requires the actual authorized provider process.
6. After recovery, verify package/schema compatibility, database and private-file consistency, identity/licensing state, readiness and worker behavior. Test representative user operations before enabling normal traffic.
7. Document the recovered point, any data-loss interval, reconciliation work and the observed restoration duration. Use those measured results to refine the disaster-recovery plan.

Expected result

The incident has a documented, authorized recovery decision and a validated service restoration, with any data loss and follow-up reconciliation made explicit.

Important considerations

- No production RPO or RTO is guaranteed by the application or by an isolated restore receipt.
- The Linux upgrade failure path does not downgrade the database. Services remain stopped until a reviewed recovery or repair is completed.
- Required external secrets are outside the v2 artifact. Recovery depends on their independent custody.
- Pooled-tenant and managed-cloud PITR recovery are not qualified as ordinary self-hosted worker operations.

Troubleshooting

- If the required key or artifact is unavailable, escalate to the designated custody owner and preserve all remaining evidence.
- If the target schema or files do not match the selected recovery point, do not open the service to users; resolve the mismatch with the recovery owner.
- If provider recovery is only a queued handoff, contact the authorized deployment provider rather than treating the queue state as restoration progress.

Related knowledge base articles: Understand recovery points and backup services; Test a recovery point in isolation; Prepare a CRM upgrade; Apply an approved application upgrade; Assess managed SaaS capabilities.

Monitor services jobs and recovery readiness

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

System Health and Backup & Recovery provide complementary operational evidence. Readiness checks database access; worker heartbeat and queues show whether background work is being processed; recovery evidence shows whether artifacts have been created, verified and tested.

Detect service degradation early and collect useful diagnostic information without exposing credentials or relying on a single green status.

Before you begin

- Authorized access to Administration > System Health and the relevant recovery information.
- A named operator and a tested external alert route where required.
- For host diagnostics, approved Linux or Windows administration access.

Procedure

1. Review System Health for database, worker and storage status, queue failures, oldest due work and recent request metrics. Compare observations over time when diagnosing intermittent behavior.
2. Check both `/api/health` and `/api/ready` through the intended operational path. A liveness response does not establish database readiness, and readiness alone does not establish worker progress.
3. Review the worker heartbeat and active/failed queues. Correlate delayed activities, email, imports, automation and backups with worker status and due-work age.
4. Open Backup & Recovery Overview and compare last successful backup, last verified backup, last test restore, storage health and next scheduled backup. Resolve operational alerts using the specific artifact/job history.
5. On packaged Linux, use status, version, bounded logs and diagnose to gather release and host evidence. On native Windows, use Server Manager, the Windows service state and protected service/gateway logs.
6. Review disk growth, capacity headroom, service restarts and certificate renewal. On cloud hosts also review the infrastructure metrics appropriate to the instance/storage model.
7. Record the symptom, time, request/job identifier, release/schema identity and affected operation when escalating. Remove credentials, tokens, personal data and raw configuration from support material.

Bounded Linux package diagnostics

```
sudo zbix-crm status
sudo zbix-crm version
sudo zbix-crm logs --tail 100 api worker
sudo zbix-crm diagnose
```

Expected result

The operator can distinguish an unavailable API, failed database readiness, stalled worker, storage pressure and incomplete recovery evidence, and has a repeatable escalation record.

Important considerations

- The Linux worker container health check uses a persisted running heartbeat with an age limit of 90 seconds. A container process alone does not prove useful work.
- System Health does not replace independent host/cloud monitoring or prove external email delivery.
- A configuration screen or healthy destination is not a successful backup restore test.
- Keep maintenance such as retention review, certificate renewal and recovery rehearsal on an owned operational schedule.

Troubleshooting

- For database readiness failure, inspect database/container service health and connectivity before restarting unrelated components.
- For a stale worker, check bounded worker logs, database availability and resource pressure. Review existing job state before resubmitting work.
- For repeated errors, preserve the relevant diagnostic window and escalate with exact status and IDs rather than screenshots containing secrets.

Related knowledge base articles: Plan capacity for CRM workloads; Schedule recurring recovery points; Verify a recovery point independently; Plan network access DNS and HTTPS; Plan recovery and rollback decisions.