



Zbix CRM Installation Guide

Product release 1.2.0

Zbix Technologies

Documentation edition 1.2.0-r1

28 October 2026

Platform requirements and release-specific Linux, native Windows Server, AWS and local development deployment procedures.

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

Choose a deployment model	3
Prepare an Ubuntu server.....	4
Prepare a Windows server.....	5
Plan network access DNS and HTTPS.....	7
Plan capacity for CRM workloads.....	8
Complete the installation readiness review.....	9
Install Zbix CRM on Ubuntu	9
Install the native Windows Server edition	11
Deploy a self hosted CRM server on AWS EC2	13
Run the local Docker Compose environment	15

Choose a deployment model

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

Zbix CRM 1.2.0 is distributed as a self-hosted Linux server package and a native Windows Server installer. Both provide shared browser access to a CRM installation. A server running in AWS remains a self-hosted deployment when the customer or implementation team operates its database, application and recovery processes.

Choose an installation method before ordering infrastructure or preparing an implementation plan. The deployment model determines service management, HTTPS configuration, storage paths and recovery responsibilities.

Before you begin

- An approved release package and a named implementation owner.
- A decision about who operates the host, database, certificates and recovery keys.

Procedure

1. Choose the Ubuntu amd64 package when Docker Engine, Compose v2 and systemd are the intended host platform. Use the Linux requirements and installation guides together.
2. Choose the native Windows Server package when the team requires a Windows service and Server Manager. The package includes its application runtime, PostgreSQL and HTTPS gateway; it does not require Docker or WSL.
3. For AWS, choose the operating system first. Follow the corresponding Linux or Windows installation method, then add cloud networking, storage and operational controls. Do not combine commands from the two packages.
4. Use the repository Docker Compose environment for local development and isolated testing. Keep its volumes and configuration separate from an installed customer system.
5. If a managed SaaS service is required, assess the separate SaaS readiness article. Establish the control plane, provisioning, subscription authority, provider recovery and operational responsibilities before describing a managed service as available.

Expected result

The implementation plan names one release, one operating model, its supported installation procedure and the owners of ongoing operations.

Fields and controls

Field or control	Rule
Ubuntu amd64	Offline application images with Docker Compose and systemd; host HTTPS prerequisites are prepared separately.
Native Windows x64	Bundled runtime and database supervised by a Windows service; browser-based users.
AWS EC2	An infrastructure location for either self-hosted package, with separate cloud operational responsibilities.
Local Compose	Development and isolated testing; not a customer production qualification.
Managed SaaS	Partial supporting capabilities; full managed-service readiness is not established.

Important considerations

- The release baseline is tag v1.2.0-20261027. Its published database migration head is 0033_record_list_performance.
- Source inspection verifies package behavior. Historical acceptance records describe particular test environments and do not establish every customer platform or workload.
- An existing cloud-hosted installation does not prove unrestricted commercial readiness, high availability or a managed multi-tenant SaaS service.

Troubleshooting

- If package names, reported versions and deployment records disagree, stop the installation handover and obtain the matching release manifest. Preserve existing data while the release identity is resolved.

- Escalate unsupported operating systems, ARM hosts, high-availability designs and pooled-tenant recovery requirements to the implementation owner before installation.

Related knowledge base articles: [Install Zbix CRM on Ubuntu](#); [Install the native Windows Server edition](#); [Deploy a self-hosted CRM server on AWS EC2](#); [Assess managed SaaS capabilities](#).

Prepare an Ubuntu server

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

The Linux distribution targets Ubuntu on amd64 hardware. Its installer verifies the operating-system family, architecture, available memory, free space, Docker availability, Compose v2 and the application port before creating an installation. The recorded EC2 acceptance environment used Ubuntu 26.04 amd64.

Prepare a host that can load the offline application images and retain enough capacity for migrations, imports, automation and encrypted recovery work.

Before you begin

- An Ubuntu amd64 host with local root or sudo administration.
- A release-specific manifest and approved host package sources.
- Storage and network decisions agreed with the implementation owner.

Procedure

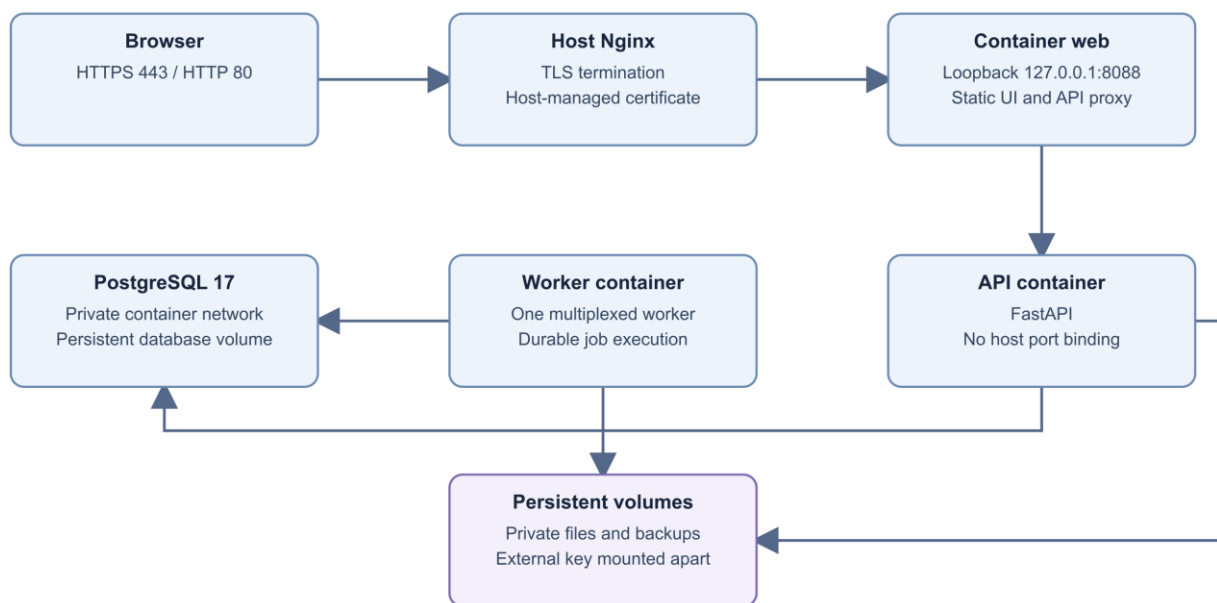
1. Confirm the host architecture is amd64 and verify that the selected Ubuntu release is qualified for the package. The installer accepts the Ubuntu family, but that check is not a support statement for every Ubuntu version.
2. Provide the documented low-volume floor of two vCPU, 8 GiB advertised RAM and at least 20 GiB free disk before the package is loaded. The memory gate allows the normal difference between advertised and usable RAM.
3. Install and validate Docker Engine and the Docker Compose v2 plugin. The application installer deliberately leaves Docker installation to the host administrator.
4. Check that TCP 8088 is unused. The packaged web service binds that port to loopback; a host HTTPS gateway provides external browser access.
5. Prepare systemd and the normal Ubuntu utilities used by the installer, including OpenSSL, curl, Python 3 and checksum tools. Prepare Nginx and Certbot separately if this host will terminate public HTTPS.
6. Reserve space beyond the installation floor for PostgreSQL growth, private files, imports, backup staging, retained recovery points, container images and logs. Review capacity before onboarding production data.

Expected result

The host passes the package preflight and has a documented growth allowance rather than only enough room for the initial installation.

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 1 Linux deployment topology

Important considerations

- The shell preflight requires at least 7,864,320 KiB usable MemTotal and 20,971,520 KiB free space on /var/lib. The documented CPU floor is planning guidance; the shell does not count CPUs.
- Application images can be carried offline. Docker Engine, host Nginx and Certbot are not included in that application archive.
- Four vCPU and 8-16 GiB RAM is a documented planning target for heavier workloads, not a measured performance guarantee.

Troubleshooting

- If preflight rejects memory or disk, increase actual host capacity or free approved space before retrying. Do not bypass the gate to make an undersized host appear qualified.
- If Docker is installed but unavailable, check the daemon and administrative access. If Compose is absent, install the supported v2 plugin before running the CRM installer.
- Escalate a non-Ubuntu or non-amd64 requirement; no equivalent package qualification is established here.

Related knowledge base articles: Plan capacity for CRM workloads; Plan network access DNS and HTTPS; Complete the installation readiness review; Install Zbix CRM on Ubuntu.

Prepare a Windows server

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

The native Windows package creates a shared Zbix CRM server accessed from browsers. It includes native application processes, PostgreSQL 17, the compiled web interface, a Windows service host and Caddy for HTTPS. There is no Docker, WSL, Hyper-V, Linux VM or WebView2 requirement for this installation method.

Validate the Windows host and access design without borrowing Linux resource figures or assuming that an installer operating-system check is a complete support certification.

Before you begin

- Administrator access to an always-on x64-compatible Windows host.
- The approved native Windows installer and its release checksum.
- An agreed public-domain, private-network or combined access design.

Procedure

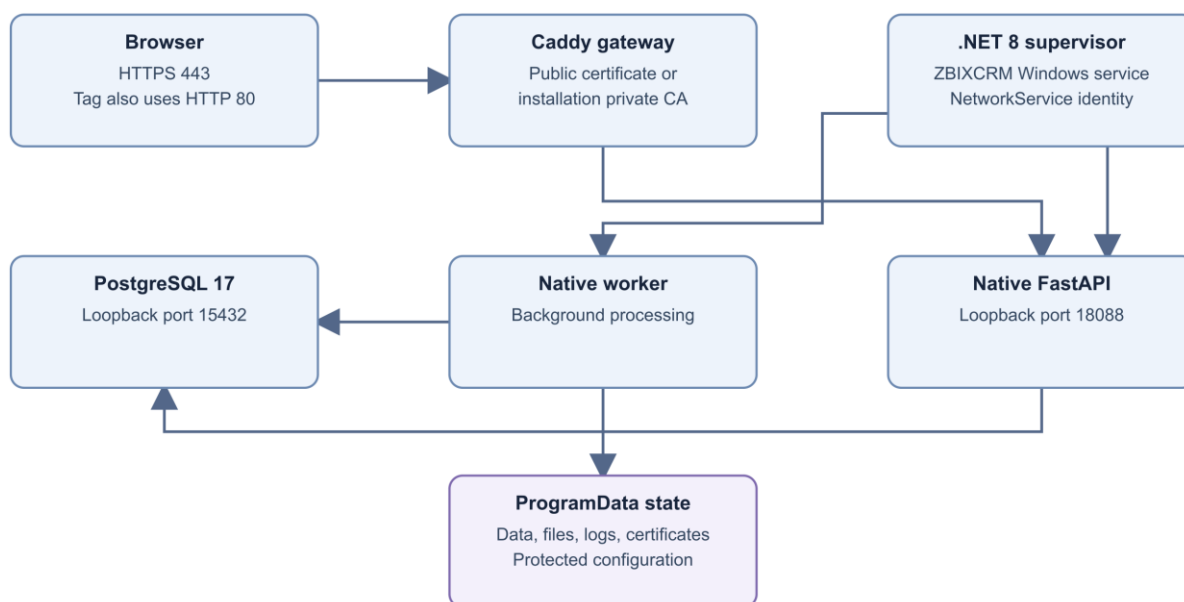
1. Confirm the target Windows build with the implementation owner. The installer has a minimum version gate of 10.0.17763 and x64-compatible architecture; the available evidence does not provide a complete certified Windows Server edition/build matrix.
2. Size CPU, memory and storage through a workload-specific assessment and an isolated installation test. No verified Windows hardware minimum was found in the reviewed release evidence.
3. Provide space for application binaries, PostgreSQL, private files, logs, temporary recovery work and retained backups. Keep recovery-key custody independent of the data drive.
4. Reserve the gateway ports required by the exact installer release. Keep the internal PostgreSQL and API listeners private. Resolve existing web-server port ownership before installation.
5. For public access, prepare a stable DNS name and the required inbound HTTPS and certificate-validation connectivity. For private hostnames or addresses, plan managed distribution of the exported office CA certificate.
6. Confirm that the organization accepts the distribution package under its software approval process. The published 1.2.0 installer is not Authenticode-signed; browser TLS certificates do not replace publisher signing.

Expected result

The host, Windows build, capacity, software provenance and client trust plan are approved for an isolated installation and subsequent operational handover.

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 2 Native Windows topology

Important considerations

- The service host is packaged as a self-contained .NET Windows application. End users only need an appropriate browser and network access.
- The 25 October native acceptance report covers isolated runtime startup and HTTPS on the earlier 1.0.0 package. The published 1.2.0 package must retain its own qualification status.
- The runtime uses ProgramData for persistent state. Reinstalling binaries is not a substitute for an independently verified recovery point.

Troubleshooting

- If the installer rejects the operating system or architecture, obtain a compatible approved package rather than bypassing the check.

- If another service occupies a gateway port, have the implementation owner review the exact release and coexistence design before starting CRM.
- Escalate missing hardware qualification or unsigned-software policy questions before customer onboarding.

Related knowledge base articles: Install the native Windows Server edition; Plan network access DNS and HTTPS; Plan capacity for CRM workloads; Complete the installation readiness review.

Plan network access DNS and HTTPS

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

Zbix CRM exposes browser access through an HTTPS gateway while keeping the database and application listeners private. Linux and native Windows packages use different gateway arrangements, so firewall and certificate instructions must follow the selected package.

Provide trusted browser access and a maintainable certificate-renewal path without exposing database or internal API ports.

Before you begin

- A selected deployment model and an approved DNS name such as `crm.example.com`.
- Control of the relevant DNS, gateway and firewall configuration.
- A plan for certificate issuance, renewal and administrative access.

Procedure

1. Map the intended browser name to a stable service endpoint. If both public and office users use the same public name, consider split DNS where the organization supports it.
2. For Linux, route host Nginx HTTPS to the packaged web service on `127.0.0.1:8088`. Keep PostgreSQL and API container ports unpublished. The reviewed public Linux deployment also uses HTTP 80 for certificate validation and HTTPS redirection.
3. For native Windows, use the bundled Caddy gateway. Keep the default internal database port 15432 and API port 18088 on loopback. Verify HTTP/HTTPS requirements against the exact installer version, particularly when another web service is present.
4. Prepare approved hostnames consistently across DNS, gateway configuration, trusted hosts and application origin. A customer hostname requires release-specific configuration adaptation; changing only DNS is insufficient.
5. Issue the appropriate certificate through the selected gateway procedure. For Windows private names or IP addresses, export the office CA public certificate and distribute it using managed client trust. Keep the CA private key on the server.
6. Test browser access with normal certificate validation, verify the expected hostname and expiration, and test renewal using the gateway procedure. Confirm external users cannot connect to the database or internal API ports.
7. Record who owns renewal failures and changes to DNS or network exposure. Restrict SSH or RDP to the organization's approved administration path.

Expected result

Users reach the correct application through trusted HTTPS, renewal has an owner, and the database and internal application listeners remain private.

Important considerations

- Do not disable certificate validation to turn a failed HTTPS check into a pass.
- A private CA is appropriate only when its trust is deliberately deployed to the intended clients. Exporting its public certificate does not authorize sharing private keys.
- Proxy and application origin settings affect login and provider callbacks. Preserve the approved origin consistently across the deployment.

Troubleshooting

- For a certificate warning, compare the URL hostname with the certificate and check client trust, DNS and expiry. Resolve the cause before onboarding users.
- For a gateway error with a healthy database, check the application readiness endpoint and gateway upstream configuration.
- For issuance or renewal failure, review challenge reachability and gateway logs. Escalate with sanitized status, time and hostname; never send certificate private keys.

Related knowledge base articles: Install Zbix CRM on Ubuntu; Install the native Windows Server edition; Deploy a self-hosted CRM server on AWS EC2; Monitor services jobs and recovery readiness.

Plan capacity for CRM workloads

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

Initial installation capacity and sustained operating capacity are different. Imports, recurring billing, document generation, reporting, automation and encrypted backups can overlap and increase memory, CPU and disk demand after an otherwise healthy installation.

Set a measured resource plan and growth thresholds for the expected users, records, private files and background work.

Before you begin

- Expected user concurrency, record volume, import sizes and file-retention requirements.
- A selected platform and an isolated representative test environment.
- Monitoring for host capacity, service health and background queues.

Procedure

1. Start from the package-specific installation requirements. For Linux the documented low-volume floor is two vCPU, 8 GiB advertised RAM and 20 GiB free disk before loading images. Do not apply those figures as a verified Windows minimum.
2. Estimate database and private-file growth separately. Include import sources, generated PDFs, logs, temporary backup staging and retained encrypted recovery points.
3. For heavier Linux work, evaluate the documented planning target of four vCPU and 8-16 GiB RAM. Maintain at least 25 percent steady-state memory headroom and test the periods when background activities overlap.
4. Rehearse representative imports, reports, automation and backup/restore tests in isolation. Record duration, queue delay, peak memory, disk growth, restarts and out-of-memory events.
5. On burstable cloud instances, monitor CPU-credit behavior as well as CPU percentage. Review storage performance and available space before increasing concurrent workloads.
6. Set operational thresholds and review them after significant data growth, new integrations or changed backup retention. Approve scaling based on measured results rather than an idle dashboard.

Expected result

The team has a workload-specific resource baseline and capacity triggers tied to observed behavior.

Important considerations

- The four Linux service memory ceilings total approximately 4.75 GiB; host services and short-lived migration or recovery work also consume resources.
- The v2 encrypted backup format is memory-bound and defaults to a 1 GiB bundle limit. Large installations require explicit backup capacity qualification.
- No universal maximum-user, throughput, latency, RPO or RTO guarantee is established by the reviewed evidence.

Troubleshooting

- If jobs accumulate while the API remains ready, examine worker heartbeat, due-work age, queue failures and resource saturation.
- If a backup exceeds its configured bound, preserve the last verified recovery point and have the recovery owner review capacity and the supported backup approach.
- Escalate recurring out-of-memory events or repeated service restarts with a sanitized time series and workload description.

Related knowledge base articles: Prepare an Ubuntu server; Prepare a Windows server; Prepare backup access keys and storage; Monitor services jobs and recovery readiness.

Complete the installation readiness review

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

A successful installer is one part of a usable CRM deployment. Release identity, administrator onboarding, licensing, HTTPS, recovery and monitoring must be prepared before the first operational handover.

Make installation prerequisites explicit and prevent an unconfigured or unqualified environment from being treated as customer ready.

Before you begin

- An installation owner and an operational handover owner.
- The approved release package, manifest and matching platform guide.

Procedure

1. Record the product version, source or package identity, expected database migration and package checksum. Confirm that all instructions refer to that release.
2. Confirm the target operating system, architecture, capacity, storage paths and port availability. Document any environment-specific qualification still required.
3. Prepare the intended domain and HTTPS design, approved application origin, client trust and restricted administrative network access.
4. Arrange the first administrator identity and a recovery path. Plan enrollment of administrators in MFA before enforcing a policy that could lock out the support team.
5. Prepare the installation-bound license request process and approved verification trust. Do not assume a demo entitlement grants customer production rights.
6. Assign custody for the backup recovery key and other required external secrets. Prepare an approved recovery destination, verification and isolated restore test.
7. Define monitoring ownership for readiness, worker health, backup failures, queue age, storage and certificate renewal. Decide how alerts will reach an operator and test the selected route.
8. For an upgrade, add a current verified recovery point, isolated restore evidence, a maintenance window and an approved rollback/forward-repair decision.

Expected result

Every prerequisite has an owner and evidence, or is recorded as an open qualification item before installation or upgrade begins.

Important considerations

- Do not replace an operational qualification with a screenshot of a configuration screen.
- The absence of a current recovery point, approved license or working alert route remains an open handover item even when the login page loads.
- This checklist does not authorize a deployment, database change or public publication.

Troubleshooting

- If a prerequisite cannot be verified, retain the existing service and resolve that item before cutover.
- Escalate mismatched package versions, unsupported targets and recovery gaps to the release and operations owners with the exact evidence that is missing.

Related knowledge base articles: Choose a deployment model; Install Zbix CRM on Ubuntu; Install the native Windows Server edition; Prepare a CRM upgrade; Activate and review a self-hosted license.

Install Zbix CRM on Ubuntu

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

The Linux server package installs a self-hosted CRM stack from bundled, versioned container images. The package creates protected configuration and persistent storage, registers a systemd service, applies the release migrations and checks readiness. Host Docker and the public HTTPS gateway are prepared separately.

Install the approved Linux release on a new qualified Ubuntu amd64 host and complete the operational checks required before users begin work.

Before you begin

- A new approved Ubuntu amd64 target that meets the Linux requirements, with sudo access.
- Docker Engine and Compose v2 available; free loopback port 8088.
- The approved Linux archive, independently obtained SHA256 and matching release manifest.
- An approved hostname, HTTPS plan, initial administrator and license/recovery arrangements.

Procedure

1. Verify the archive name and outer SHA256 against the publisher's approved release information before extraction. For release 1.2.0 the archive is ZBIX-CRM-Server-1.2.0-20261027-linux-x64.tar.gz. A matching filename alone does not establish integrity.
2. Extract the archive to a staging directory and open its manifest.json and image-lock.tsv. Confirm the expected release and source identity. Run the inner checksum verification from the extracted package root and resolve every mismatch before continuing.
3. Complete the host preflight: Ubuntu amd64, Docker Engine, Compose v2, available memory and disk, and an unused TCP 8088. Review the release-specific hostname configuration with the implementation owner; customer-domain adaptation must cover the gateway, application origin and trusted host settings.
4. On this new authorized target, run `sudo ./bin/zbix-crm install` from the extracted package. The installer rejects an existing installation. It loads or verifies the exact bundled image identities, creates the installation directories, generates protected local configuration and a separate recovery key, writes the versioned release and registers systemd.
5. Wait for the installer's readiness result. Run `sudo zbix-crm status` and `sudo zbix-crm version`. Compare the version and migration with the approved manifest; the v1.2 release specifies 0033_record_list_performance. Check the local readiness endpoint before exposing browser access.
6. Run `sudo zbix-crm bootstrap-admin --email admin@example.com`, substituting the approved administrator address. Enter the password at the hidden prompt. Bootstrap authorization is local/root-only, lasts one hour after installation and is consumed after success. If it expires, an authorized local operator can use `bootstrap-renew`.
7. Complete the separately reviewed host Nginx and certificate procedure for the approved domain. Verify trusted HTTPS, the expected browser name, HTTP-to-HTTPS behavior, certificate renewal and private database/API exposure. The application package does not automatically qualify a new customer DNS or TLS configuration.
8. Sign in through HTTPS, review License Management, enroll the appropriate administrators in MFA and complete the security handover. Create a recovery point, verify it independently and run an isolated restore test. Record monitoring ownership and outstanding qualifications before onboarding business records.

Run from the verified extracted package on a new authorized host

```
sha256sum -c checksums.sha256
sudo ./bin/zbix-crm install
sudo zbix-crm status
sudo zbix-crm version
```

Local readiness and administrator bootstrap; replace the example email

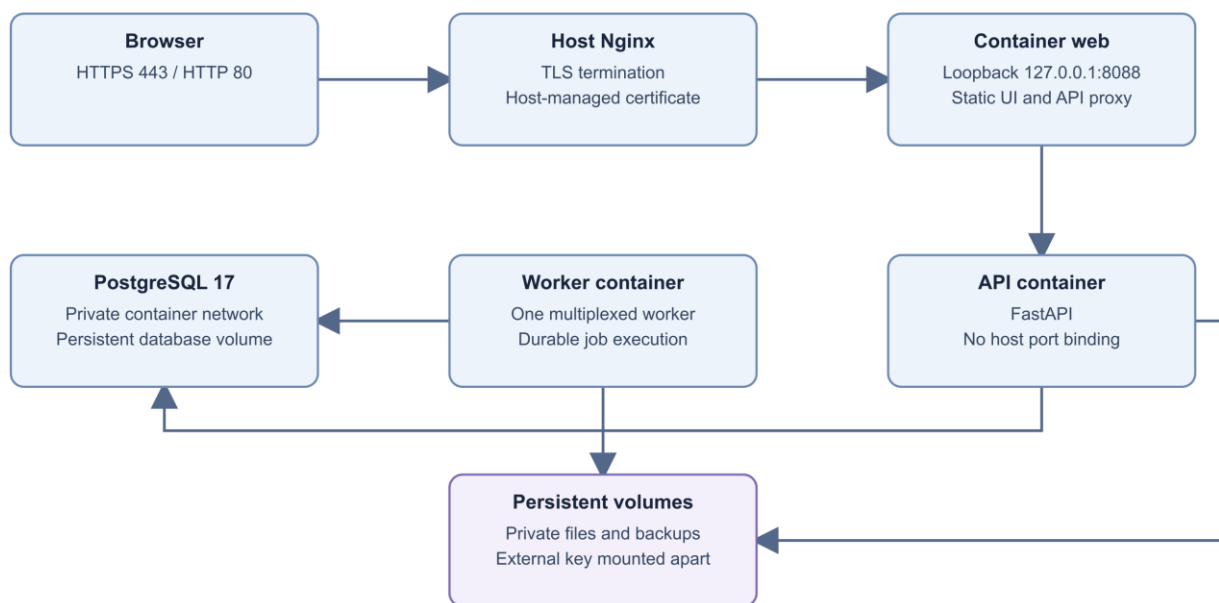
```
curl -fsS http://127.0.0.1:8088/api/ready
sudo zbix-crm bootstrap-admin --email admin@example.com
```

Expected result

The installed services are healthy, release and schema identity match the approved package, browser access uses trusted HTTPS, and administrator, license and recovery handover are documented.

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

Important considerations

- Installation performs database initialization and forward migrations. Use this procedure only on the approved target; use the upgrade procedure for an existing installation.
- Persistent directories are /opt/zbix-crm, /etc/zbix-crm and /var/lib/zbix-crm. Preserve configuration, private files, database storage and separately held recovery secrets.
- The offline archive contains application images; it does not bundle Docker, host Nginx or Certbot.
- These steps were reviewed against source. No fresh installer or migration was executed for this documentation.

Troubleshooting

- For a failed checksum or image identity check, obtain a clean approved package. Do not edit the manifest to accept unexpected bytes.
- If readiness fails, inspect status and bounded API/worker logs, then check database readiness and storage. Preserve the diagnostic evidence and avoid deleting volumes as a repair step.
- If bootstrap reports expiration or prior consumption, resolve the local authorization state with the installation owner. Do not create a public bootstrap endpoint.
- Escalate with package version, expected schema, failed step and sanitized diagnostic output. Exclude configuration values and keys.

Related knowledge base articles: Prepare an Ubuntu server; Plan network access DNS and HTTPS; Complete the installation readiness review; Create an encrypted recovery point; Activate and review a self-hosted license.

Install the native Windows Server edition

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

The native installer turns a qualified Windows host into a shared CRM server. It installs the application, PostgreSQL 17, a compiled browser interface and the HTTPS gateway, supervised by the ZBIX CRM Windows service. Users connect from browsers; they do not install a separate CRM desktop client.

Configure a native Windows deployment for public-domain access, private-office access or both, then verify its readiness for operational handover.

Before you begin

- An approved compatible x64 Windows host and administrator access.
- A publisher-approved installer with its trusted release checksum; the published 1.2.0 installer is unsigned.
- Available gateway ports and an approved DNS/client-trust plan.
- Initial administrator details, license activation arrangements and recovery-key custody.

Procedure

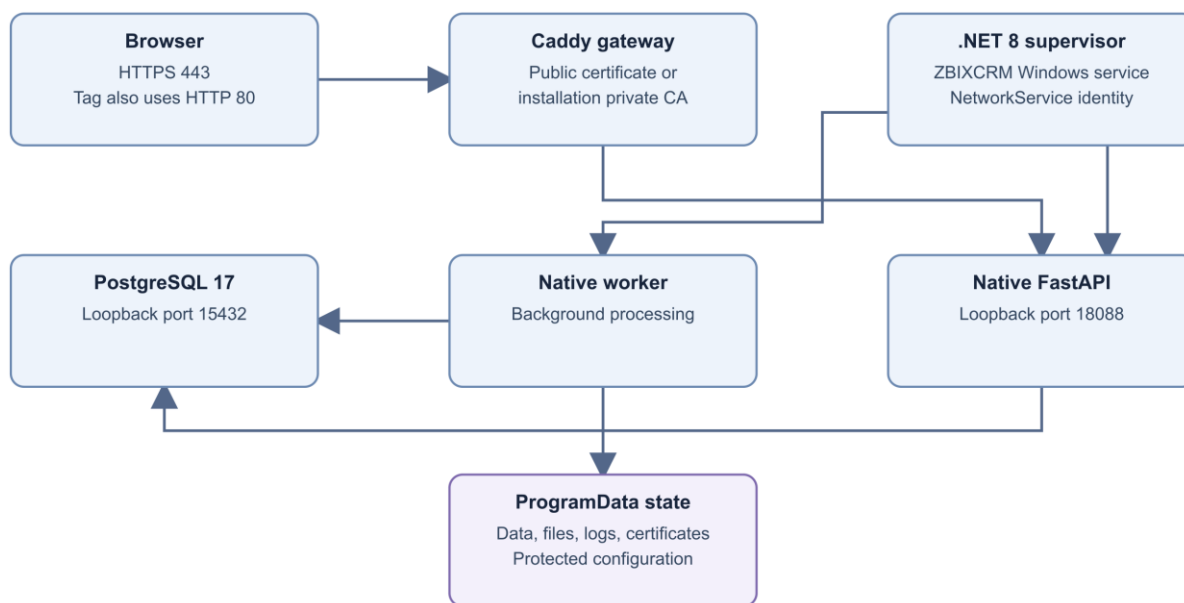
1. Check the installer identity and SHA256 against the approved release information. For v1.2.0 the package is ZBIX-CRM-Windows-Server-x64-Setup-1.2.0.exe. Resolve publisher-signing requirements through the organization's software approval process before installation.
2. Confirm the Windows build, workload capacity and gateway port ownership. The installer enforces x64 compatibility and an operating-system version floor, but the target environment still needs qualification. Keep the internal database and API ports unavailable from the LAN or Internet.
3. Run the approved installer as administrator. It installs the runtime and registers the automatic ZBIX CRM Server service and gateway firewall rules. Open ZBIX CRM Server Manager from the installed shortcut.
4. For public access, enter Public domain (optional) using a DNS name such as crm.example.com, and enter Certificate email. The name must resolve to the intended gateway and certificate validation must reach the host according to the exact package's network requirements.
5. For private access, enter Private hostnames or IPv4 addresses, one per line. At least one public or private name is required. Where available, use split DNS with the public name so office clients can use the same trusted public certificate.
6. Enter the first administrator email, a password of at least 12 characters and its confirmation. Choose Save configuration and start. The runtime initializes PostgreSQL when needed, prepares configuration, applies migrations, and starts the API, worker and HTTPS gateway.
7. Check Server Manager status and the Windows service. Open the configured HTTPS address and verify the application readiness endpoint and sign-in page. Do not treat an internal API response as proof that the public gateway is accessible from an intended user device.
8. For private CA mode, choose Export office CA certificate after the server has started. Distribute this public root through the organization's managed client-trust process, then verify a browser connection without warnings. Keep the CA private key on the server.
9. Complete License Management, administrator MFA, recovery storage, an independently verified recovery point and an isolated restore test. Record the installed release, operational owners and any uncompleted qualification before handing over to users.

Expected result

The Windows service is running, intended clients reach trusted HTTPS, the initial administrator can sign in, and license, monitoring and recovery responsibilities are documented.

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

Important considerations

- Persistent data, configuration, certificates, logs and backups are stored under ProgramData\ZBIXCRM. Uninstall retains this state; retain an independent verified recovery point as well.
- No Docker, WSL, Hyper-V, Linux VM or WebView2 dependency is required by this package.
- Follow the exact installer release's gateway-port instructions. Later source changes are not automatically part of an earlier installer.
- The reviewed native acceptance report describes an isolated earlier package. These v1.2 steps are source-reviewed and were not independently executed for this documentation.

Troubleshooting

- If configuration is rejected, check that a public name or private host is present and that the chosen ports are valid and distinct.
- If startup fails, review Server Manager status, the service state and protected service/gateway logs. Check port conflicts, database startup and readiness before repeating configuration.
- If private HTTPS is untrusted, confirm the correct exported office CA reached the intended client trust store and that the URL matches a configured name.
- Escalate with installer version, checksum, target Windows build, failed step and sanitized logs. Never attach runtime configuration, passwords or private keys.

Related knowledge base articles: Prepare a Windows server; Plan network access DNS and HTTPS; Activate and review a self-hosted license; Prepare backup access keys and storage; Monitor services jobs and recovery readiness.

Deploy a self hosted CRM server on AWS EC2

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

AWS EC2 can host either the Ubuntu distribution or the native Windows Server package. The application remains self-hosted: the operating team is responsible for the instance, network exposure, certificates, updates, storage and recoverability. The available Linux deployment report records a v1.2.0 installation on Ubuntu 26.04 amd64.

Combine the appropriate package installation with an explicit cloud networking and operations plan. This guide does not create AWS resources or imply that a complete managed SaaS control plane is included.

Before you begin

- An approved AWS account and operator permissions for the intended infrastructure.
- A selected Linux or Windows package, release manifest and qualified operating-system image.
- Approved network, DNS, storage, recovery and monitoring designs.
- A controlled administration path and a maintenance/change owner.

Procedure

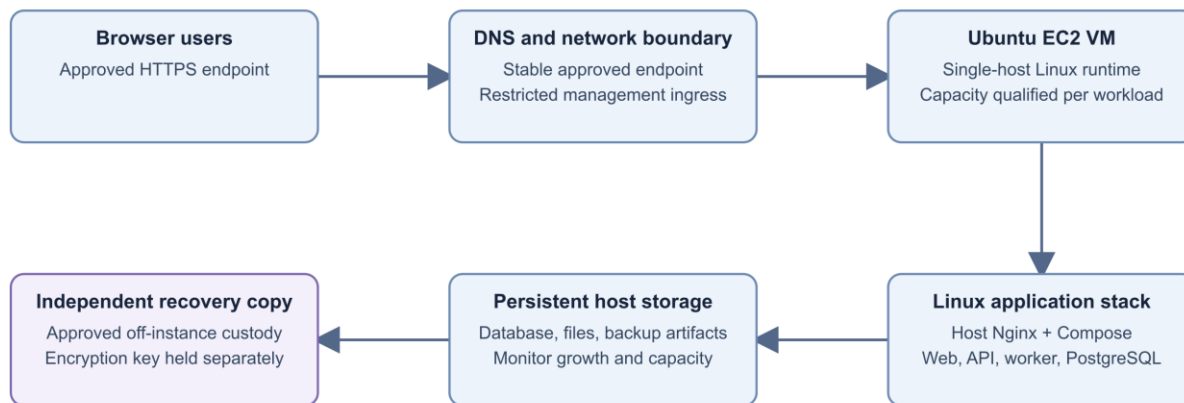
1. Select the operating system and architecture before selecting the package. For the Linux path, the recorded acceptance configuration used a t3a.large with two vCPU and 8 GiB advertised RAM. Treat it as a documented test/deployment configuration, not a universal capacity guarantee or mandatory instance family.
2. Plan persistent storage for the database, private files, image/runtime footprint, temporary recovery work and retention. Confirm the cloud storage protection and performance settings through the cloud team rather than inferring them from application health.
3. Provide a stable service endpoint and prepare a DNS name such as crm.example.com. Restrict SSH or RDP to the approved administrator path. Expose only the selected gateway's public ports; do not expose PostgreSQL, the internal API or the Linux loopback web port.
4. Transfer the approved release package through the organization's controlled process and verify its outer SHA256 and inner manifest where applicable. The Linux distribution loads bundled application images; it does not require compiling the CRM source on the instance.
5. Follow the complete Linux installation guide for Ubuntu, or the native Windows guide for Windows. Keep their directory layouts, service controls and gateway requirements distinct. Do not mix a Windows Server Manager procedure with Linux Compose commands.
6. Complete the approved DNS and certificate configuration. On the Linux topology, host Nginx terminates HTTPS and forwards to the loopback web container. On native Windows, the bundled Caddy gateway serves the configured public/private names. Validate trusted HTTPS from the intended user network.
7. Check release identity, migration, readiness and service health. Verify that the worker is processing and that unauthenticated access is denied where authentication is required. Complete administrator bootstrap and license activation through the relevant package workflow.
8. Create a recovery point, independently verify it and perform an isolated restore test. Protect required keys separately and arrange approved off-instance retention. A backup on the same EC2 instance is not protection against loss of that instance.
9. Configure independent monitoring for readiness, worker heartbeat/queues, storage, certificate renewal and recovery failures. For burstable instances include CPU-credit monitoring. Record alert ownership and conduct an operational handover before increasing workload.

Expected result

The selected self-hosted package operates behind trusted HTTPS on the approved cloud network, with identified owners for security, capacity, backup and maintenance.

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 5 AWS EC2 deployment context

Important considerations

- The dated v1.2 deployment record is evidence of one running installation and one qualified upgrade/recovery preparation. It is not a fresh cloud audit performed for this guide.
- This guide makes no assertion about a particular AWS account, region, instance identifier, storage encryption setting or current security group.
- Load balancers and alternative TLS termination require a separately reviewed origin, proxy and certificate design. They are not interchangeable with the direct gateway procedure.
- Do not describe EC2 hosting alone as high availability, managed multi-tenancy or a service-level guarantee.

Troubleshooting

- If HTTPS fails while local readiness succeeds, check DNS, gateway configuration, certificate validation and the relevant network rules.
- If the instance is healthy but background work is delayed, inspect worker heartbeat and resource pressure, including burstable CPU credits.
- If off-instance recovery has not been tested, keep that qualification open even when an artifact upload succeeds.
- Escalate with package version, selected OS, failed layer, time and sanitized diagnostics. Do not share cloud credentials or private infrastructure identifiers in public support posts.

Related knowledge base articles: Choose a deployment model; Install Zbix CRM on Ubuntu; Install the native Windows Server edition; Plan network access DNS and HTTPS; Plan capacity for CRM workloads; Configure and validate backup storage.

Run the local Docker Compose environment

Zbix CRM 1.2.0 | Source reviewed | Reviewed 28 October 2026

The repository includes a Docker Compose environment for local development and testing. Its web service is exposed on loopback and its database and application services stay on the Compose network. This environment is distinct from the versioned Linux server distribution and the native Windows installer.

Give engineers an isolated place to inspect or test a matching source release without changing an existing CRM installation.

Before you begin

- An authorized development checkout of the intended source release.
- Docker Engine and Compose available, with sufficient host capacity.
- A separate development configuration and storage; no production credentials or restored customer data unless separately authorized.

Procedure

1. Read the checkout's README, Compose definition and release notes before starting services. Confirm that the source revision matches the feature or issue being tested.
2. Choose a separate development directory and Compose project/storage plan. Check that its host port and volume names do not collide with an installed CRM environment.
3. For a new development environment, prepare configuration from the release's example using protected local values. Preserve any existing configuration and volumes when resuming an established environment.
4. Use the documented local startup method. The PowerShell helper is scripts/start.ps1; the underlying Compose startup builds and starts the stack. Startup can apply migrations, so run it only against the authorized isolated database.
5. Open the local web address and check /api/health and /api/ready. Create the initial development administrator through the CLI's hidden password prompt.
6. Use fictional test data only when explicitly needed. The seed-demo command is a data-writing operation and is not part of a read-only review.
7. Record the exact source revision, configuration profile and test dataset when reporting results. Do not treat local success as production deployment or capacity qualification.

Development startup on an authorized isolated checkout

```
docker compose up -d --build
docker compose ps
```

Create a development administrator through the hidden password prompt

```
docker compose exec api python -m app.cli create-admin --email admin@example.com
```

Expected result

A source-matched, isolated development stack is available without sharing the installed customer database or private-file volumes.

Important considerations

- The documented local address is http://localhost:8088. Verify that it is unused before startup; do not displace another installation.
- A local HTTP profile does not establish production cookie, TLS, network or recovery readiness.
- Never use volume-removal or reset commands as routine troubleshooting for a persistent CRM database.

Troubleshooting

- If the host port is in use, identify the existing service and allocate an isolated test configuration instead of stopping it without authorization.
- If startup fails, inspect bounded Compose logs and readiness. Preserve configuration and data while diagnosing the failure.
- Escalate source/schema mismatches with the commit, expected migration and sanitized logs.

Related knowledge base articles: Choose a deployment model; Complete the installation readiness review; Monitor services jobs and recovery readiness.