

changelog.md for @serve.zone/onebox

2026-03-24 - 1.24.2 - fix(deps)

bump runtime and build tool dependencies

- update @design.estate/dees-catalog to ^3.49.0
- update development tooling packages @git.zone/tsbundle, @git.zone/tsdeno, and @git.zone/tswatch

2026-03-24 - 1.24.1 - fix(repo)

migrate smart build config to .smartconfig.json and tidy repository metadata

- Rename npmpextra.json to .smartconfig.json and extend it with CLI project metadata for the repository.
- Mark the package as private and add an empty pnpm overrides block in package.json.
- Expand .gitignore to cover common build artifacts, caches, install directories, and local tooling folders.
- Reformat changelog and README files for cleaner spacing and Markdown table alignment without changing documented behavior.

2026-03-24 - 1.24.0 - feat(backup)

add containerarchive-backed backup storage, restore, download, and pruning support

- add database support for archive snapshot IDs and stored size tracking for backups
- initialize and close the backup archive during onebox lifecycle startup and shutdown
- allow backup download and restore flows to work with archive snapshots as well as legacy file-based backups
- schedule daily archive pruning based on the most generous configured retention policy
- replace smart3 with smartstorage for registry-backed S3-compatible storage

2026-03-21 - 1.23.0 - feat(appstore)

add remote app store templates with service upgrades and Redis/MariaDB platform support

- introduces an App Store manager, API handlers, shared request types, and web UI flow for browsing remote templates and deploying services from template metadata
- tracks app template id and version on services, adds upgrade discovery and migration-based service upgrades, and includes a database migration for template version columns
- adds Redis and MariaDB platform service providers with provisioning plus backup and restore support, and exposes their requirements through service creation and app template config

2026-03-18 - 1.22.2 - fix(web-ui)

stabilize app store service creation flow and add Ghost sqlite defaults

- Defers App Store navigation to the services view to avoid destroying the current view during the deploy event handler.
- Processes pending app templates after services view updates so the create flow opens reliably.
- Adds default Ghost environment variables for sqlite3 and the database file path in the App Store template.
- Removes obsolete Gitea CI and npm publish workflow definitions.

2026-03-18 - 1.22.1 - fix(repo)

no changes to commit

2026-03-18 - 1.22.0 - feat(web-appstore)

add an App Store view for quick service deployment from curated templates

- adds a new App Store tab to the web UI with curated Docker app templates
- passes selected app templates through UI state into the services view for quick deployment
- supports quick deploy creation with prefilled image, port, environment variables, and optional platform service flags

- updates @serve.zone/catalog to ^2.8.0 to support the new app store view

2026-03-18 - 1.21.0 - feat(opsserver)

add container workspace API and backend execution environment for services

- introduces typed workspace handlers for reading, writing, listing, creating, removing, and executing commands inside service containers
- adds frontend backend-execution environment integration so the service view can open a workspace against a selected service
- extends Docker exec lookup to resolve Swarm service container IDs when a direct container ID is unavailable

2026-03-17 - 1.20.0 - feat(ops-dashboard)

stream user service logs to the ops dashboard and resolve service containers for Docker log streaming

- add typed socket support for pushing live user service log entries to the web app
- extend platform log streaming to include running user services with separate dashboard handlers
- fall back from direct container lookup to service-to-container resolution when streaming Docker logs
- update log parsing to preserve timestamps and infer log levels for service log entries
- bump @serve.zone/catalog to ^2.7.0

2026-03-17 - 1.19.12 - fix(repo)

no changes to commit

2026-03-17 - 1.19.11 - fix(repo)

no changes to commit

2026-03-17 - 1.19.10 - fix(repo)

no changes to commit

2026-03-17 - 1.19.9 - fix(repo)

no changes to commit

2026-03-17 - 1.19.8 - fix(repo)

no changes to commit

2026-03-17 - 1.19.7 - fix(repo)

no changes to commit

2026-03-17 - 1.19.6 - fix(repository)

no changes to commit

2026-03-17 - 1.19.5 - fix(repo)

no changes to commit

2026-03-17 - 1.19.4 - fix(repository)

no changes to commit

2026-03-16 - 1.19.3 - fix(repo)

no changes to commit

2026-03-16 - 1.19.2 - fix(docs)

remove outdated UI screenshot assets from project documentation

- Deletes multiple PNG screenshots that documented previous dashboard, service form, and hello-world states.
- Reduces repository clutter by removing obsolete image assets no longer needed in docs.

2026-03-16 - 1.19.1 - fix(dashboard)

add updated dashboard screenshots for refresh and resource usage states

- Adds new dashboard screenshots covering post-refresh, resource usage, and populated data views.
- Updates visual assets to document current dashboard behavior and UI states.

2026-03-16 - 1.19.1 - fix(dashboard)

add aggregated resource usage stats to the dashboard

- Aggregate CPU, memory, and network stats across all running user and platform service containers in `getSystemStatus`
- Extend `ISystemStatus.docker` interface with `cpuUsage`, `memoryUsage`, `memoryTotal`, `networkIn`, `networkOut` fields
- Fix `getContainerStats` to properly handle Swarm service IDs by catching exceptions and falling back to label-based container lookup
- Wire dashboard resource usage card to display real aggregated data from the backend

2026-03-16 - 1.19.0 - feat(opsserver,web)

add real-time platform service log streaming to the dashboard

- stream running platform service container logs from the ops server to connected dashboard clients via `TypedSocket`
- parse Docker log timestamps and levels for both pushed and fetched platform service log entries
- enhance the platform service detail view with mapped statuses and predefined host, port, version, and config metadata
- add the `typedsocket` dependency and update the catalog package for dashboard support

2026-03-16 - 1.18.5 - fix(platform-services)

fix platform service detail view navigation and log display

- Add back button to platform service detail view for returning to services list
- Fix DOM lifecycle when switching between platform services (destroy and recreate dees-chart-log)
- Fix timestamp format for log entries to use ISO 8601 for dees-chart-log compatibility
- Clear previous stats/logs state before fetching new platform service data

2026-03-16 - 1.18.4 - fix(repo)

no changes to commit

2026-03-16 - 1.18.3 - fix(deps)

bump @serve.zone/catalog to ^2.6.1

- Updates the @serve.zone/catalog runtime dependency from ^2.6.0 to ^2.6.1.

2026-03-16 - 1.18.2 - fix(repo)

no changes to commit

2026-03-16 - 1.18.1 - fix(repo)

no changes to commit

2026-03-16 - 1.18.0 - feat(platform-services)

add platform service log retrieval and display in the services UI

- add typed request support in the ops server to fetch Docker logs for platform service containers
- store fetched platform service logs in web app state and load them when opening platform service details
- render platform service logs in the services detail view and add sidebar icons for main navigation tabs

2026-03-16 - 1.17.4 - fix(docs)

add hello world running screenshot for documentation

- Adds a new PNG asset showing the application in a running hello world state.
- Supports project documentation or README usage without changing runtime behavior.

2026-03-16 - 1.17.3 - fix(mongodb)

downgrade the MongoDB service image to 4.4 and use the legacy mongo shell for container operations

- changes the default MongoDB container image from mongo:7 to mongo:4.4
- replaces mongosh with mongo for health checks, provisioning, and deprovisioning inside the container

2026-03-16 - 1.17.2 - fix(platform-services)

provision ClickHouse, MinIO, and MongoDB resources via docker exec instead of host port access

- switch ClickHouse provisioning and teardown to in-container client commands to avoid host port mapping issues
- replace MinIO host-side S3 API calls with in-container mc commands for bucket creation and removal
- run MongoDB provisioning and deprovisioning through mongosh inside the container and improve docker exec failure reporting

2026-03-16 - 1.17.1 - fix(repo)

no changes to commit

2026-03-16 - 1.17.0 - feat(web/services)

add deploy service action to the services view

- Adds a prominent "Deploy Service" button to the services page header.
- Routes users into the create service view directly from the services listing.
- Includes a new service creation form screenshot asset for the updated interface.

2026-03-16 - 1.16.0 - feat(services)

add platform service navigation and stats in the services UI

- add platform service stats state and fetch action
- show platform services in the services list and open a platform detail view
- enable dashboard clicks to jump directly to the selected platform service
- refresh platform service stats after start and restart actions
- bump @serve.zone/catalog to ^2.6.0 for the new platform service UI components

2026-03-16 - 1.15.3 - fix(install)

refresh systemd service configuration before restarting previously running installations

- Re-enable the systemd service during updates so unit file changes are applied before restart
- Add a log message indicating the service configuration is being refreshed

2026-03-16 - 1.15.2 - fix(systemd)

set HOME and DENO_DIR for the systemd service environment

- Adds HOME=/root to the generated onebox systemd unit
- Adds DENO_DIR=/root/.cache/deno so Deno cache paths are available when running as a service

2026-03-16 - 1.15.1 - fix(systemd)

move Docker installation and swarm initialization to systemd enable flow

- Ensures Docker is installed before writing and enabling the systemd unit that depends on docker.service.
- Removes Docker auto-installation from Onebox initialization so setup happens in the service management path.

2026-03-16 - 1.15.0 - feat(systemd)

replace smartdaemon-based service management with native systemd commands

- adds a dedicated OneboxSystemd manager for enabling, disabling, starting, stopping, checking status, and following logs
- introduces a new `onebox systemd` CLI command set and updates install and help output to use it
- removes the smartdaemon dependency and related service management code

2026-03-16 - 1.14.10 - fix(services)

stop auto-update monitoring during shutdown

- Track the auto-update polling interval in the services manager
- Clear the auto-update interval when Onebox shuts down to prevent background checks after shutdown

2026-03-16 - 1.14.9 - fix(repo)

no changes to commit

2026-03-16 - 1.14.8 - fix(repo)

no changes to commit

2026-03-16 - 1.14.7 - fix(repo)

no changes to commit

2026-03-16 - 1.14.6 - fix(project)

no changes to commit

2026-03-16 - 1.14.5 - fix(onebox)

move Docker auto-install and swarm initialization into Onebox startup flow

- removes Docker setup from daemon service installation
- ensures Docker is installed before Docker initialization during Onebox startup
- preserves automatic Docker Swarm initialization on fresh servers

2026-03-16 - 1.14.4 - fix(repo)

no changes to commit

2026-03-16 - 1.14.3 - fix(repo)

no changes to commit

2026-03-16 - 1.14.2 - fix(repo)

no changes to commit

2026-03-16 - 1.14.1 - fix(repo)

no changes to commit

2026-03-16 - 1.14.0 - feat(daemon)

auto-install Docker and initialize Swarm during daemon service setup

- Adds a Docker availability check before installing the Onebox daemon service
- Installs Docker automatically when it is missing using the standard installation script

- Attempts to initialize Docker Swarm after installation and handles already-initialized environments gracefully

2026-03-16 - 1.13.17 - fix(ci)

remove forced container image pulling from Gitea workflow jobs

- Drops the `--pull always` container option from CI, npm publish, and release workflows.
- Keeps workflow container images unchanged while avoiding forced pulls on every job run.

2026-03-16 - 1.13.16 - fix(ci)

refresh workflow container images on every run and bump @apiclient.xyz/docker to ^5.1.1

- add `--pull always` to CI, release, and npm publish workflow containers to avoid stale images
- update @apiclient.xyz/docker from ^5.1.0 to ^5.1.1 in deno.json

2026-03-15 - 1.13.15 - fix(repo)

no changes to commit

2026-03-15 - 1.13.14 - fix(repo)

no changes to commit

2026-03-15 - 1.13.13 - fix(repo)

no changes to commit

2026-03-15 - 1.13.12 - fix(ci)

run npm install with `--ignore-scripts` in CI and release workflows

- Update CI workflow dependency installation steps to skip lifecycle scripts during builds.
- Apply the same install change to the release workflow for consistent automation behavior.

2026-03-15 - 1.13.11 - fix(project)

no changes to commit

2026-03-15 - 1.13.10 - fix(deps)

bump @git.zone/tsdeno to ^1.2.0

- Updates the tsdeno development dependency from ^1.1.1 to ^1.2.0.

2026-03-15 - 1.13.9 - fix(repo)

no changes to commit

2026-03-15 - 1.13.8 - fix(repo)

no changes to commit

2026-03-15 - 1.13.7 - fix(repo)

no changes to commit

2026-03-15 - 1.13.6 - fix(ci)

correct workflow container image registry path

- Update Gitea CI, release, and npm publish workflows to use the corrected ht-docker-node image path
- Align all workflow container references from hosttoday to host.today to prevent pipeline image resolution issues

2026-03-15 - 1.13.5 - fix(workflows)

switch Gitea workflow containers from ht-docker-dbase to ht-docker-node

- Updates the CI, release, and npm publish workflows to use the Node-focused container image consistently.
- Aligns workflow runtime images with the project's Node and Deno build and publish steps.

2026-03-15 - 1.13.4 - fix(ci)

run workflows in the shared build container and enable corepack for pnpm installs

- adds the ht-docker-dbase container image to CI, release, and npm publish workflows
- enables corepack before pnpm install in build and release jobs to ensure package manager availability

2026-03-15 - 1.13.3 - fix(build)

replace custom Deno compile scripts with tsdeno-based binary builds in CI and release workflows

- adds @git.zone/tsdeno as a dev dependency and configures compile targets in npmextra.json
- updates CI and release workflows to install Node.js dependencies before running tsdeno compile
- removes the legacy scripts/compile-all.sh script and points the compile task to tsdeno compile

2026-03-15 - 1.13.2 - fix(scripts)

install production dependencies before compiling binaries and exclude local node_modules from builds

- Adds a dependency installation step using the application entrypoint before cross-platform compilation
- Updates all deno compile targets to use --node-modules-dir=none to avoid bundling local node_modules

2026-03-15 - 1.13.1 - fix(deno)

remove nodeModulesDir from Deno configuration

- Drops the explicit nodeModulesDir setting from deno.json.
- Keeps the package version unchanged at 1.13.0 while simplifying runtime configuration.

2026-03-15 - 1.13.0 - feat(install)

improve installer with version selection, service restart handling, and upgrade documentation

- Adds installer command-line options for help, specific version selection, and custom install directory.
- Fetches the latest release from the Gitea API when no version is provided and installs the matching platform binary.
- Preserves Onebox data directories, stops and restarts the systemd service during updates, and refreshes installation instructions in the README including upgrade usage.

2026-03-15 - 1.12.1 - fix(package.json)

update package metadata

- Single metadata-only file changed (+1, -1)
- No source code or runtime behavior modified; safe patch release

2026-03-15 - 1.12.0 - feat(cli,release)

add self-upgrade command and automate CI, release, and npm publishing workflows

- adds a new `onebox upgrade` CLI command that checks the latest release and reinstalls the current binary via the installer script
- introduces Gitea CI workflows for type checks, build verification, multi-platform binary compilation, release creation, and npm publishing
- adds a reusable release template describing installation options, supported platforms, and checksum availability

2026-03-03 - 1.11.0 - feat(services)

map backend service data to UI components, add stats & logs parsing, fetch service stats, and fix logs request param

- Fix: rename service logs request property from 'lines' to 'tail' when calling `typedRequest`
- Add data transformation helpers: `formatBytes`, `parseImageString`, `mapStatus`, `toServiceDetail`, `toServiceStats`, `parseLogs`
- Transform service list and detail props to match `@serve.zone/catalog` component interfaces (`map status`, `image`, `repo/tag`, `timestamps`, `registry`)

- Dispatch `fetchServiceStatsAction` on service click and surface transformed stats with default values to avoid nulls
- Parse and normalize logs into timestamp/message pairs for the detail view

2026-03-02 - 1.10.3 - fix(bin)

make `bin/onebox-wrapper.js` executable

- Metadata-only change: file mode updated for `bin/onebox-wrapper.js` to include the executable bit
- No source or behavior changes to the code

2026-03-02 - 1.10.2 - fix(build)

update `build/watch` configuration, switch to `esbuild` bundler and `tswatch`, and bump catalog and tooling dependencies

- Switch watch script to `'tswatch'` (replaced previous `concurrently` command invoking `deno + tswatch`).
- `npmextra.json`: set bundler to `'esbuild'`, enable production mode, include `html/index.html` in the bundle, and extend `watchPatterns` to include `./html/*`.
- Backend watcher: expanded watch globs and changed command to include `--unstable-ffi` and runtime flags (`--ephemeral --monitor`); restart and debounce kept.
- Bump runtime deps: `@design.estate/dees-catalog -> ^3.43.3`, `@serve.zone/catalog -> ^2.5.0`.
- Bump devDependencies: `@git.zone/tsbundle -> ^2.9.0`, `@git.zone/tswatch -> ^3.2.0`.

2026-02-24 - 1.10.1 - fix(package.json)

update package metadata

- Single metadata-only file changed (+1 -1)
- No source code or runtime behavior modified; safe patch release
- Current package version is 1.10.0; recommend patch bump to 1.10.1

2026-02-24 - 1.10.0 - feat(opsserver)

introduce `OpsServer` (`TypedRequest` API) and new lightweight web UI; replace legacy Angular UI and add typed interfaces

- Add OpsServer (ts/opsserver) with TypedRequest handlers for admin, services, platform, dns, domains, registry, network, backups, schedules, settings and logs.
- Integrate typedrequest/typedserver and smartjwt/smartguard plugins (ts/plugins.ts) and add comprehensive ts_interfaces for requests and data shapes.
- Replace legacy HTTP server usage with OpsServer throughout daemon, Onebox class and CLI (ts/classes/daemon.ts, ts/classes/onebox.ts, ts/cli.ts).
- Implement log streaming via VirtualStream and support for downloading/restoring backups and registry token management within handlers.
- Introduce new web UI built with dees-element web components under ts_web (ob-app-shell and views) and bundle/watch tooling (npmextra.json, tsbundle/tswatch integration).
- Update package.json: add build/watch scripts, tsbundle/tswatch dev deps and new runtime dependencies for typedrequest and catalog components.
- Remove large Angular-based ui application and related services/components in ui/ (major cleanup of Angular code and assets).
- Note: This adds many new endpoints and internal API changes (TypedRequest-based); consumers of the old UI/HTTP endpoints should migrate to the new OpsServer TypedRequest API and web components.

2025-12-03 - 1.9.2 - fix(ui)

Add VS Code configs for the UI workspace and normalize dark theme CSS variables

- Add VS Code workspace files under ui/.vscode:
 - ◦ extensions.json: recommend the Angular language support extension
 - ◦ launch.json: Chrome launch configurations for 'ng serve' and 'ng test' (preLaunchTask hooks)
 - ◦ tasks.json: npm 'start' and 'test' tasks with a background TypeScript problem matcher to improve dev workflow
- Update ui/src/styles.css dark theme variables to use neutral black/gray HSL values for background, foreground, cards, popovers, accents, borders, inputs and ring to improve contrast and consistency

2025-11-27 - 1.9.1 - fix(ui)

Correct import success toast and add VS Code launch/tasks recommendations for the UI

- Fix backup import success toast in backups-tab.component to reference response.data.service.name (previously response.data.serviceName), preventing incorrect service name display.
- Add VS Code workspace settings for the UI: extensions recommendation, launch configurations for 'ng serve' and 'ng test', and npm tasks for start/test to simplify local development and debugging.

2025-11-27 - 1.9.0 - feat(backups)

Add backup import API and improve backup download/import flow in UI

- Backend: add `/api/backups/import` endpoint to accept multipart file uploads or JSON with a URL and import backups (saves temp file, validates `.tar.enc`, calls `backupManager.restoreBackup` in import mode).
- Backend: server-side import handler downloads remote backup URLs, stores temporary file, invokes restore/import logic and cleans up temp files.
- Frontend: add `downloadBackup`, `importBackupFromFile` and `importBackupFromUrl` methods to `ApiService`; trigger browser download using Blob and object URL with Authorization header.
- Frontend: replace raw download link in service detail UI with a Download button that calls `downloadBackup` and shows success/error toasts.
- Dev: add VS Code launch, tasks and recommended extensions for the ui workspace to simplify local development.

2025-11-27 - 1.8.0 - feat(backup)

Add backup scheduling system with GFS retention, API and UI integration

- Introduce backup scheduling subsystem (`BackupScheduler`) and integrate it into Onebox lifecycle (init & shutdown)
- Extend `BackupManager.createBackup` to accept schedule metadata (`scheduleId`) so scheduled runs are tracked
- Add GFS-style retention policy support (`IRetentionPolicy` + `RETENTION_PRESETS`) and expose per-tier retention in types
- Database migrations and repository changes: create `backups` and `backup_schedules` tables, add `schedule_id`, per-tier retention columns, and scope (`all/pattern/service`) support (migrations up to version 12)
- HTTP API: add backup schedule endpoints (`GET/POST/PUT/DELETE /api/backup-schedules`), trigger endpoint (`/api/backup-schedules/:id/trigger`), and service-scoped schedule endpoints
- UI: add API client methods for backup schedules and register a Backups tab in Services UI to surface schedules/backups
- Add task scheduling dependency (`@push.rocks/taskbuffer`) and export it via `plugins.ts`; update `deno.json` accordingly
- Type and repository updates across codebase to support schedule-aware backups, schedule CRUD, and retention enforcement

2025-11-27 - 1.7.0 - feat(backup)

Add backup system: BackupManager, DB schema, API endpoints and UI support

Introduce a complete service backup/restore subsystem with encrypted archives, database records and REST endpoints. Implements BackupManager with export/import for service config, platform resources (MongoDB, MinIO, ClickHouse), and Docker images; adds BackupRepository and migrations for backups table and include_image_in_backup; integrates backup flows into the HTTP API and the UI client; exposes backup password management and restore modes (restore/import/clone). Wire BackupManager into Onebox initialization.

- Add BackupManager implementing create/restore/export/import/encrypt/decrypt workflows (service config, platform resource dumps, Docker image export/import) and support for restore modes: restore, import, clone.
- Add BackupRepository and database migrations: create backups table and add include_image_in_backup column to services; database API methods for create/get/list/delete backups.
- Add HTTP API endpoints for backup management: list/create/get/download/delete backups, restore backups (/api/backups/restore) and backup password endpoints (/api/settings/backup-password).
- Update UI ApiService and types: add IBackup, IRestoreOptions, IRestoreResult, IBackupPasswordStatus and corresponding ApiService methods (getBackups, createBackup, getBackup, deleteBackup, getBackupDownloadUrl, restoreBackup, setBackupPassword, checkBackupPassword).
- Expose includeImageInBackup flag on service model and persist it in ServiceRepository (defaults to true for existing rows); service update flow supports toggling this option.
- Integrate BackupManager into Onebox core (initialized in Onebox constructor) and wire HTTP handlers to use the new manager; add DB repository export/import glue so backups are stored and referenced by ID.

2025-11-27 - 1.6.0 - feat(ui.dashboard)

Add Resource Usage card to dashboard and make dashboard cards full-height; add VSCode launch/tasks/config

- Introduce ResourceUsageCardComponent and include it as a full-width row in the dashboard layout.
- Make several dashboard card components (Certificates, Traffic, Platform Services) full-height by adding host classes and applying h-full to ui-card elements for consistent card sizing.
- Reflow dashboard rows (insert Resource Usage as a dedicated row and update row numbering) to improve visual layout.
- Add VSCode workspace configuration: recommended Angular extension, launch configurations for ng serve/ng test, and npm tasks to run/start the UI in development.

2025-11-27 - 1.5.0 - feat(network)

Add traffic stats endpoint and dashboard UI; enhance platform services and certificate health reporting

- Add `/api/network/traffic-stats` GET endpoint to the HTTP API with an optional `minutes` query parameter (validated, 1-60).
- Implement traffic statistics aggregation in `CaddyLogReceiver` using rolling per-minute buckets (`requestCount`, `errorCount`, `avgResponseTime`, `totalBytes`, `statusCounts`, `requestsPerMinute`, `errorRate`).
- Expose `getTrafficStats(minutes?)` in the Angular `ApiService` and add `ITrafficStats` type to the client API types.
- Add dashboard UI components: `TrafficCard`, `PlatformServicesCard`, `CertificatesCard` and integrate them into the main Dashboard (including links to Platform Services).
- Enhance system status data: `platformServices` entries now include `displayName` and `resourceCount`; add `certificateHealth` summary (`valid`, `expiringSoon`, `expired`, `expiringDomains`) returned by `Onebox` status.
- Platform services manager and `Onebox` code updated to surface provider information and resource counts for the UI.
- Add VSCode workspace `launch/tasks` recommendations for the UI development environment.

2025-11-26 - 1.4.0 - feat(platform-services)

Add ClickHouse platform service support and improve related healthchecks and tooling

- Add ClickHouse as a first-class platform service: register provider, provision/cleanup support and env var injection
- Expose ClickHouse endpoints in the HTTP API routing (`list/get/start/stop/stats`) and map default port (8123)
- Enable services to request ClickHouse as a platform requirement (`enableClickHouse / platformRequirements`) during deploy/provision flows
- Fix ClickHouse container health check to use absolute `wget` path (`/usr/bin/wget`) for more reliable in-container checks
- Add VS Code workspace `launch/tasks/extensions` configs for the UI (`ui.vscode/*`) to improve local dev experience

2025-11-26 - 1.3.0 - feat(platform-services)

Add ClickHouse platform service support (provider, types, provisioning, UI and port mappings)

- Introduce ClickHouse as a first-class platform service: added ClickHouseProvider and registered it in PlatformServicesManager
- Support provisioning ClickHouse resources for user services and storing encrypted credentials in platform_resources
- Add ClickHouse to core types (TPlatformServiceType, IPlatformRequirements, IServiceDeployOptions) and service DB handling so services can request ClickHouse
- Inject ClickHouse-related environment variables into deployed services (CLICKHOUSE_* mappings) when provisioning resources
- Expose ClickHouse default port (8123) in platform port mappings / network targets
- UI: add checkbox and description for enabling ClickHouse during service creation; form now submits enableClickHouse
- Add VS Code recommendations and launch/tasks for the UI development workflow

2025-11-26 - 1.2.1 - fix(platform-services/minio)

Improve MinIO provider: reuse existing data and credentials, use host-bound port for provisioning, and safer provisioning/deprovisioning

- MinIO provider now detects existing data directory and will reuse stored admin credentials when available instead of regenerating them.
- If data exists but no credentials are stored, MinIO deployment will wipe the data directory to avoid credential mismatch and fail early with a clear error if wiping fails.
- Provisioning and deprovisioning now connect to MinIO via the container's host-mapped port (127.0.0.1:<hostPort>) instead of relying on overlay network addresses; an error is thrown when the host port mapping cannot be determined.
- Bucket provisioning creates policies and returns environment variables using container network hostnames for in-network access; a warning notes that per-service MinIO accounts are TODO and root credentials are used for now.
- Added logging improvements around MinIO deploy/provision/deprovision steps for easier debugging.
- Added VSCode workspace files (extensions, launch, tasks) for the ui project to improve developer experience.

2025-11-26 - 1.2.0 - feat(ui)

Sync UI tab state with URL and update routes/links

- Add VSCode workspace recommendations, launch and tasks configs for the UI (ui/.vscode/*)
- Update Angular routes to support tab URL segments and default redirects for services, network and registries
- Change service detail route to use explicit 'detail/:name' path and update links accordingly
- Make ServicesList, Registries and Network components read tab from route params and navigate on tab changes; add ngOnDestroy to unsubscribe
- Update Domain detail template link to point to the new services detail route

2025-11-26 - 1.1.0 - feat(platform-services)

Add platform service log streaming, improve health checks and provisioning robustness

- Add WebSocket log streaming support for platform services (backend + UI) to stream MinIO/MongoDB/Caddy logs in real time
- Improve platform service lifecycle: detect unhealthy 'running' containers, mark for redeploy and wait/retry health checks with detailed logging
- MinIO health check now uses container IP (via Docker) instead of hostname to reliably probe the service
- MongoDB and MinIO providers updated to use host-mapped ports for host-side provisioning and connect via 127.0.0.1:<hostPort>
- Docker manager: pullImage now actively pulls images and createContainer binds service ports to localhost so host-based provisioning works
- UI: platform service detail page can start/stop/clear platform log streams; log stream service state cleared on disconnect to avoid stale logs
- Caddy / reverse-proxy improvements to manage certificates and routes via the Caddy manager (Caddy runs as Docker service)
- Add VSCode workspace helpers (extensions, launch, tasks) to improve developer experience

All notable changes to this project will be documented in this file.

The format is based on [Keep a Changelog](#), and this project adheres to [Semantic Versioning](#).

[Unreleased]

Added

- Initial project structure
- Core architecture classes
- Docker container management
- Nginx reverse proxy integration
- Cloudflare DNS management
- Let's Encrypt SSL automation
- SQLite database layer
- Angular web UI
- Multi-user authentication
- Systemd daemon integration
- CLI commands for all operations
- Metrics collection and historical data
- Log aggregation
- Registry authentication support

[1.0.0] - TBD

Added

- First stable release

Revision #2

Created 2026-03-28 13:11:52 UTC by foss.global Team

Updated 2026-03-29 16:54:22 UTC by foss.global Team