

@serve.zone/onebo

X

a single server quick hosting tool

- [readme.md for @serve.zone/onebox](#)
- [changelog.md for @serve.zone/onebox](#)

readme.md for @serve.zone/onebox

“ ☐ Self-hosted Docker Swarm platform with Caddy reverse proxy, automatic SSL, and real-time WebSocket updates

Onebox transforms any Linux server into a powerful container hosting platform. Deploy Docker Swarm services with automatic HTTPS, DNS configuration, and Caddy reverse proxy running as a Docker service - all managed through a beautiful Angular web interface with real-time updates.

Issue Reporting and Security

For reporting bugs, issues, or security vulnerabilities, please visit community.foss.global/. This is the central community hub for all issue reporting. Developers who sign and comply with our contribution agreement and go through identification can also get a code.foss.global/ account to submit Pull Requests directly.







What Makes Onebox Different? ☐

- **Caddy Reverse Proxy in Docker** - Production-grade HTTP/HTTPS proxy running as a Swarm service with native service discovery, HTTP/2, HTTP/3, and bidirectional WebSocket proxying
- **Docker Swarm First** - All workloads (including the reverse proxy!) run as Swarm services on the overlay network for seamless service-to-service communication
- **Real-time Everything** - WebSocket-powered live updates for service status, logs, and metrics across all connected clients
- **Single Executable** - Compiles to a standalone binary - just run it, no dependencies
- **Private Registry Included** - Built-in Docker registry with token-based auth and auto-deploy on push
- **Zero Config SSL** - Automatic Let's Encrypt certificates with inline `load_pem` (no volume mounts needed)
- **Cloudflare Integration** - Automatic DNS record management and zone synchronization






- **Modern Stack** - Deno runtime + SQLite database + Angular 19 UI

Features





Core Platform

-  **Docker Swarm Management** - Deploy, scale, and orchestrate services with Swarm mode
-  **Caddy Reverse Proxy** - Production-grade proxy running as Docker service with SNI, HTTP/2, HTTP/3
-  **Automatic SSL Certificates** - Let's Encrypt integration with hot-reload and renewal monitoring
-  **Cloudflare DNS Integration** - Automatic DNS record creation and zone synchronization
-  **Built-in Registry** - Private Docker registry with per-service tokens and auto-update
-  **Real-time WebSocket Updates** - Live service status, logs, and system events

Monitoring & Management

-  **Metrics Collection** - Historical CPU, memory, and network stats (every 60s)
-  **Centralized Logging** - Container logs with streaming and retention policies
-  **Angular Web UI** - Modern, responsive interface with real-time updates
-  **Multi-user Support** - Role-based access control (admin/user)
-  **SQLite Database** - Embedded, zero-configuration storage

Developer Experience

-  **Auto-update on Push** - Push to registry and services update automatically
-  **Private Registry Support** - Use Docker Hub, Gitea, or custom registries
-  **Systemd Integration** - Run as a daemon with auto-restart
-  **Full CLI & API** - Manage everything from terminal or HTTP API

Quick Start

Installation

```
# One-line install (recommended)
curl -sSL https://code.foss.global/serve.zone/onebox/raw/branch/main/install.sh | sudo bash

# Install a specific version
curl -sSL https://code.foss.global/serve.zone/onebox/raw/branch/main/install.sh | sudo bash -s
-- --version v1.11.0

# Or install from npm
pnpm install -g @serve.zone/onebox
```

First Run

```
# Start the server in development mode
onebox server --ephemeral

# In another terminal, deploy your first service
onebox service add myapp \
  --image nginx:latest \
  --domain app.example.com \
  --port 80
```

Access the Web UI

Open `http://localhost:3000` in your browser.

Default credentials:

- Username: `admin`
- Password: `admin`

⚠ **Change the default password immediately after first login!**

Production Setup

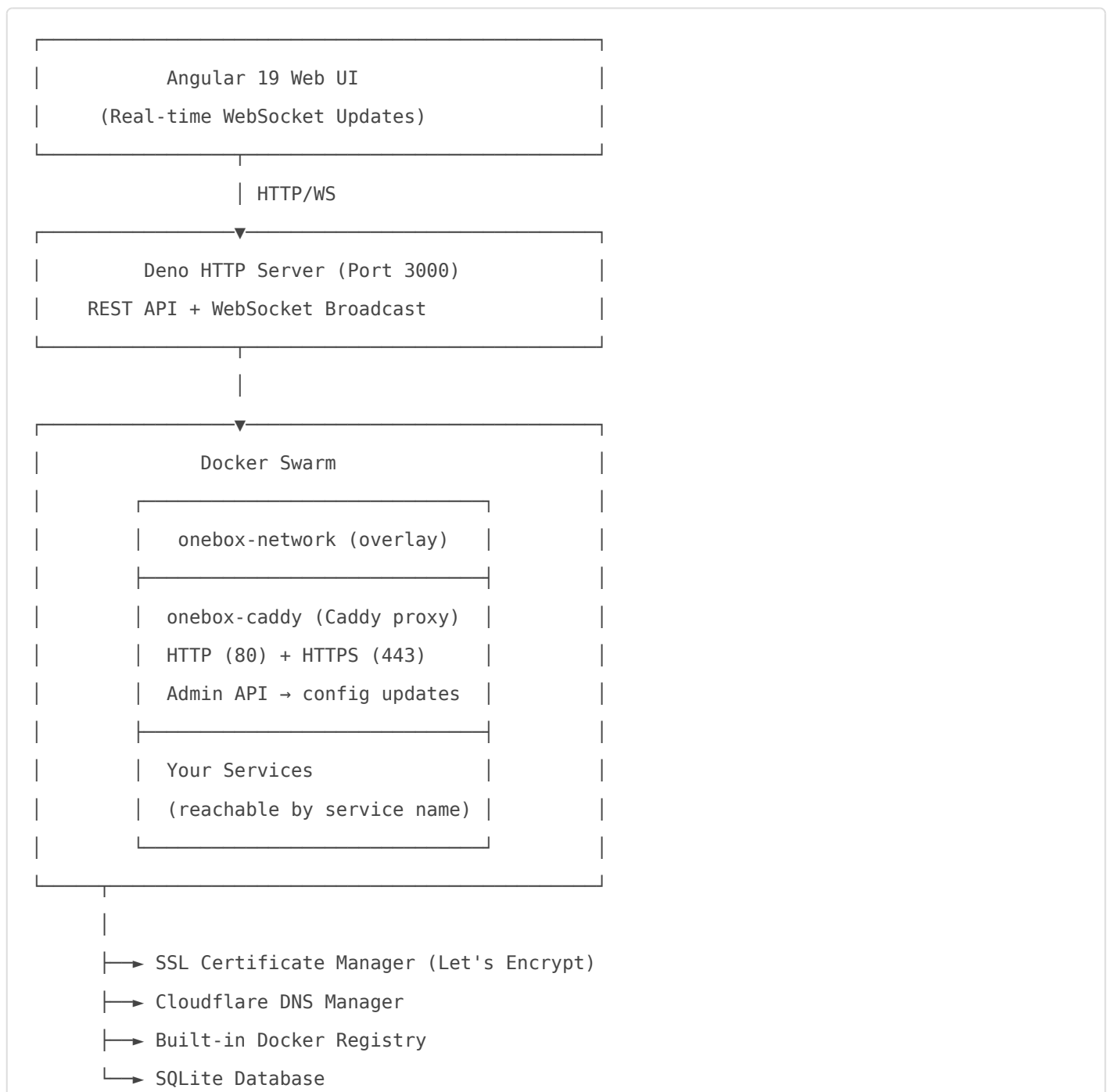
```
# Install as systemd service
sudo onebox daemon install
```

```
# Start the daemon
sudo onebox daemon start

# View logs
sudo onebox daemon logs
```

Architecture

Onebox is built with modern technologies for performance and developer experience:



Core Components

Component	Description
Deno Runtime	Modern TypeScript with built-in security
Caddy Reverse Proxy	Docker Swarm service with HTTP/2, HTTP/3, SNI, and WebSocket support
Docker Swarm	Container orchestration (all workloads run as services)
SQLite Database	Configuration, metrics, and user data
WebSocket Server	Real-time bidirectional communication
Let's Encrypt	Automatic SSL certificate management
Cloudflare API	DNS record automation

CLI Reference

Service Management

```
# Deploy a service
onebox service add <name> --image <image> --domain <domain> [--port <port>] [--env KEY=VALUE]

# Deploy with Onebox Registry (auto-update on push)
onebox service add myapp --use-onebox-registry --domain myapp.example.com

# List services
onebox service list

# Control services
onebox service start <name>
onebox service stop <name>
onebox service restart <name>

# Remove service
onebox service remove <name>

# View logs
onebox service logs <name>
```

Server Management

```
# Start server (development)
onebox server --ephemeral          # Runs in foreground with monitoring

# Start server (production)
onebox daemon install             # Install systemd service
onebox daemon start               # Start daemon
onebox daemon stop                # Stop daemon
onebox daemon logs                # View logs
```

Registry Management

```
# Add external registry credentials
onebox registry add --url registry.example.com --username user --password pass

# List registries
onebox registry list

# Remove registry
onebox registry remove <url>
```

DNS Management

```
# Add DNS record (requires Cloudflare config)
onebox dns add <domain>

# List DNS records
onebox dns list

# Sync from Cloudflare
onebox dns sync

# Remove DNS record
onebox dns remove <domain>
```

SSL Management

```
# Renew expiring certificates
onebox ssl renew

# Force renew specific domain
onebox ssl force-renew <domain>

# List certificates
onebox ssl list
```

Configuration

```
# Show all settings
onebox config show

# Set configuration value
onebox config set <key> <value>

# Example: Configure Cloudflare
onebox config set cloudflareAPIKey your-api-key
onebox config set cloudflareEmail your@email.com
onebox config set cloudflareZoneID your-zone-id
```

System Status

```
# Get full system status
onebox status
```

Upgrade

```
# Upgrade to the latest version (requires root)
sudo onebox upgrade
```

Configuration

System Requirements

- **Linux** (x64 or ARM64)
- **Docker** installed and running
- **Docker Swarm** initialized (`docker swarm init`)
- **Root/sudo access** for ports 80/443
- **(Optional) Cloudflare account** for DNS automation

Data Locations

Data	Location
Database	<code>./onebox.db</code> (or custom path)
SSL Certificates	Managed by CertManager
Registry Data	<code>./.nokit/registry-data</code>

Environment Variables

```
# Database location
ONEBOX_DB_PATH=/path/to/onebox.db

# HTTP server port (default: 3000)
ONEBOX_HTTP_PORT=3000

# Enable debug logging
ONEBOX_DEBUG=true
```

Development

Setup

```
# Clone repository
git clone https://code.foss.global/serve.zone/onebox
cd onebox

# Start development server (auto-restart on changes)
pnpm run watch
```

Tasks

```
# Development server (auto-restart on changes)
deno task dev

# Run tests
deno task test

# Watch mode for tests
deno task test:watch

# Compile binaries for all platforms
deno task compile
```

Project Structure

```
onebox/
├─ ts/
│  └─ classes/                # Core implementations
│     └─ onebox.ts           # Main coordinator
│     └─ reverseproxy.ts    # Reverse proxy orchestration
│     └─ caddy.ts           # Caddy Docker service management
│     └─ docker.ts          # Docker Swarm API
│     └─ httpserver.ts      # REST API + WebSocket
│     └─ services.ts        # Service orchestration
│     └─ certmanager.ts     # SSL certificate management
│     └─ cert-requirement-manager.ts # Certificate requirements
│     └─ ssl.ts             # SSL utilities
│     └─ registry.ts        # Built-in Docker registry
│     └─ registries.ts      # External registry management
```

```

| | └─ dns.ts                # DNS record management
| | └─ cloudflare-sync.ts   # Cloudflare zone sync
| | └─ daemon.ts           # Systemd daemon management
| | └─ apiclient.ts        # API client utilities
| └─ database/             # Database layer (repository pattern)
| | └─ index.ts            # Main OneboxDatabase class
| | └─ base.repository.ts  # Base repository class
| | └─ repositories/      # Domain-specific repositories
| |   └─ service.repository.ts
| |   └─ certificate.repository.ts
| |   └─ auth.repository.ts
| |   └─ metrics.repository.ts
| |   └─ ...
| └─ cli.ts                # CLI router
| └─ types.ts              # TypeScript interfaces
| └─ logging.ts            # Logging utilities
| └─ plugins.ts            # Dependency imports
└─ ui/                     # Angular 19 web interface
└─ test/                   # Test files
└─ mod.ts                  # Main entry point
└─ deno.json               # Deno configuration

```

API Endpoints

The HTTP server exposes a comprehensive REST API:

Authentication

Method	Endpoint	Description
POST	/api/auth/login	User authentication (returns token)

Services

Method	Endpoint	Description
GET	/api/services	List all services
POST	/api/services	Create/deploy service
GET	/api/services/:name	Get service details
PUT	/api/services/:name	Update service

Method	Endpoint	Description
DELETE	/api/services/:name	Delete service
POST	/api/services/:name/start	Start service
POST	/api/services/:name/stop	Stop service
POST	/api/services/:name/restart	Restart service
GET	/api/services/:name/logs	Get service logs
WS	/api/services/:name/logs/stream	Stream logs via WebSocket

SSL Certificates

Method	Endpoint	Description
GET	/api/ssl/list	List all certificates
GET	/api/ssl/:domain	Get certificate details
POST	/api/ssl/obtain	Request new certificate
POST	/api/ssl/:domain/renew	Force renew certificate

Domains

Method	Endpoint	Description
GET	/api/domains	List all domains
GET	/api/domains/:domain	Get domain details
POST	/api/domains/sync	Sync domains from Cloudflare

DNS Records

Method	Endpoint	Description
GET	/api/dns	List DNS records
POST	/api/dns	Create DNS record
DELETE	/api/dns/:domain	Delete DNS record
POST	/api/dns/sync	Sync DNS from Cloudflare

Registry

Method	Endpoint	Description
GET	/api/registry/tags/:service	Get registry tags for service
GET	/api/registry/tokens	List registry tokens

Method	Endpoint	Description
POST	/api/registry/tokens	Create registry token
DELETE	/api/registry/tokens/:id	Delete registry token

System

Method	Endpoint	Description
GET	/api/status	System status
GET	/api/settings	Get settings
PUT	/api/settings	Update settings
WS	/api/ws	WebSocket for real-time updates

WebSocket Messages

Real-time updates are broadcast via WebSocket:

```
// Service lifecycle updates
{
  type: 'service_update',
  action: 'created' | 'updated' | 'deleted' | 'started' | 'stopped',
  service: { id, name, status, ... }
}

// Service status changes
{
  type: 'service_status',
  service: { id, name, status, ... }
}

// System status updates
{
  type: 'system_status',
  status: { docker, reverseProxy, services, ... }
}
```

Advanced Usage

Using the Built-in Registry

```
# Deploy a service with Onebox Registry
onebox service add myapp \
  --use-onebox-registry \
  --domain myapp.example.com \
  --auto-update-on-push

# Get the registry token for pushing images
# (Token is automatically created and stored in database)

# Push your image
docker tag myimage:latest localhost:4000/myapp:latest
docker push localhost:4000/myapp:latest

# Service automatically updates! ☐☐
```

Registry Token Management

```
# Create a CI/CD token via API
curl -X POST http://localhost:3000/api/registry/tokens \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d '{"name": "github-actions", "type": "ci", "scope": ["myapp"], "expiresIn": "90d"}'

# Use token for docker login
docker login localhost:4000 -u ci -p <token>
```

Cloudflare DNS Integration

```
# Configure Cloudflare (one-time setup)
onebox config set cloudflareAPIKey your-api-key
onebox config set cloudflareEmail your@email.com
onebox config set cloudflareZoneID your-zone-id

# Deploy with automatic DNS
```

```
onebox service add myapp \  
  --image nginx:latest \  
  --domain myapp.example.com  
  
# DNS record is automatically created!  
  
# Sync all domains from Cloudflare  
onebox dns sync
```

SSL Certificate Management

SSL certificates are automatically obtained and renewed:

- Certificates are requested when a service with a domain is deployed
- Renewal happens automatically 30 days before expiry
- Certificates are hot-reloaded without downtime
- Force renewal: `onebox ssl force-renew <domain>`

Monitoring and Metrics

Metrics are collected every 60 seconds (configurable):

```
# Set metrics interval (milliseconds)  
onebox config set metricsInterval 30000  
  
# View in web UI or query database directly  
sqlite3 onebox.db "SELECT * FROM metrics WHERE service_id = 1 ORDER BY timestamp DESC LIMIT  
10"
```

Troubleshooting

Docker Swarm Not Initialized

```
# Initialize Docker Swarm  
docker swarm init
```

```
# Verify swarm mode
docker info | grep "Swarm: active"
```

Port Already in Use

```
# Check what's using port 80/443
sudo lsof -i :80
sudo lsof -i :443

# Kill the process or change Onebox ports
onebox config set httpPort 8080
```

SSL Certificate Issues

```
# Check certificate status
onebox ssl list

# Verify DNS is pointing to your server
dig +short yourdomain.com

# Force certificate renewal
onebox ssl force-renew yourdomain.com
```

WebSocket Connection Issues

- Ensure firewall allows WebSocket connections
- Check browser console for connection errors
- Verify `/api/ws` endpoint is accessible

Service Not Starting

```
# Check Docker logs
docker service logs <service-name>

# Check Onebox logs
onebox daemon logs
```

```
# Verify image exists
docker images | grep <image-name>
```

License and Legal Information

This repository contains open-source code that is licensed under the MIT License. A copy of the MIT License can be found in the [license](#) file within this repository.

Please note: The MIT License does not grant permission to use the trade names, trademarks, service marks, or product names of the project, except as required for reasonable and customary use in describing the origin of the work and reproducing the content of the NOTICE file.

Trademarks

This project is owned and maintained by Task Venture Capital GmbH. The names and logos associated with Task Venture Capital GmbH and any related products or services are trademarks of Task Venture Capital GmbH and are not included within the scope of the MIT license granted herein. Use of these trademarks must comply with Task Venture Capital GmbH's Trademark Guidelines, and any usage must be approved in writing by Task Venture Capital GmbH.

Issue Reporting and Security

For reporting bugs, issues, or security vulnerabilities, please visit community.foss.global/. This is the central community hub for all issue reporting. Developers who sign and comply with our contribution agreement and go through identification can also get a code.foss.global/ account to submit Pull Requests directly.

Company Information

Task Venture Capital GmbH Registered at District court Bremen HRB 35230 HB, Germany

For any legal inquiries or if you require further information, please contact us via email at hello@task.vc.

By using this repository, you acknowledge that you have read this section, agree to comply with its terms, and understand that the licensing of the code does not imply endorsement by Task Venture Capital GmbH of any derivative works.

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 npmextra.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 smartS3 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 pnpm 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