onecli / onecli Public

Code | Issues 4 | Pull requests 3 | Actions | Projects | Security | Insights

About

Open-source credential vault, give your AI agents access to services without exposing keys.

onecli.sh

#nodejs #rust #cli #security #postgres #vault #mcp #secret-management #security-tools #ai-agents #openclaw #nanoclaw

Readme
Apache-2.0 license
Code of conduct
Contributing
Activity
Custom properties
1.3k stars
5 watching
50 forks
Report repository

Releases 24

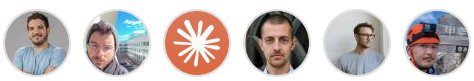
v1.6.0 Latest
5 hours ago

+ 23 releases

Packages 1

onecli

Contributors 6



Languages

TypeScript 60.6% **Rust** 36.3% **Other** 3.1%

guyb1 fix: rewrite denormalize migration to handle non-empty tables (#106)

a6e75b7 · 1 hour ago

.agents/skills	fix: initial	2 weeks ago
.claude/skills	clean unnecessary (#10)	2 weeks ago
.github	chore: add Prisma migration drif…	yesterday
.husky	fix: auth login flow, schema migr…	2 days ago
.vscode	chore: rust-analyzer for vscode …	2 weeks ago
apps	refactor: switch IDs to UUID and…	2 hours ago
assets	fix: crop excess whitespace fro…	2 weeks ago
docker	fix: install OpenSSL dev header…	5 days ago
docs	feat: Bitwarden vault support (#60)	5 days ago
packages	fix: rewrite denormalize migratio…	1 hour ago
.dockerignore	chore: align naming (#38)	2 weeks ago
.env.example	fix: add gateway auth extractor, …	last week
.gitignore	chore: align naming (#38)	2 weeks ago
.mise.toml	fix: add mise config, page head…	2 weeks ago
.prettierignore	fix: initial	2 weeks ago
CHANGELOG.md	chore(main): release 1.6.0 (#99)	5 hours ago
CLAUDE.md	fix: claude md (#2)	2 weeks ago
CODE_OF_CONDUCT.md	Adding official Apache 2.0 Licen…	2 weeks ago
CONTRIBUTING.md	Adding official Apache 2.0 Licen…	2 weeks ago
LICENSE	Adding official Apache 2.0 Licen…	2 weeks ago
README.md	Update Discord link in README…	yesterday
package.json	chore(main): release 1.6.0 (#99)	5 hours ago

pnpm-lock.yaml	fix: auth login flow, schema migr…	2 days ago
pnpm-workspace.yaml	fix: initial	2 weeks ago
skills-lock.json	fix(proxy): improve auth handlin…	2 weeks ago
turbo.json	feat: add account layer for multi-…	yesterday

README | Code of conduct | Contributing | Apache-2.0 license

Agent-first credential vault

The secret vault for AI agents.
Store once. Inject anywhere. Agents never see the keys.

Website · Docs · Discord

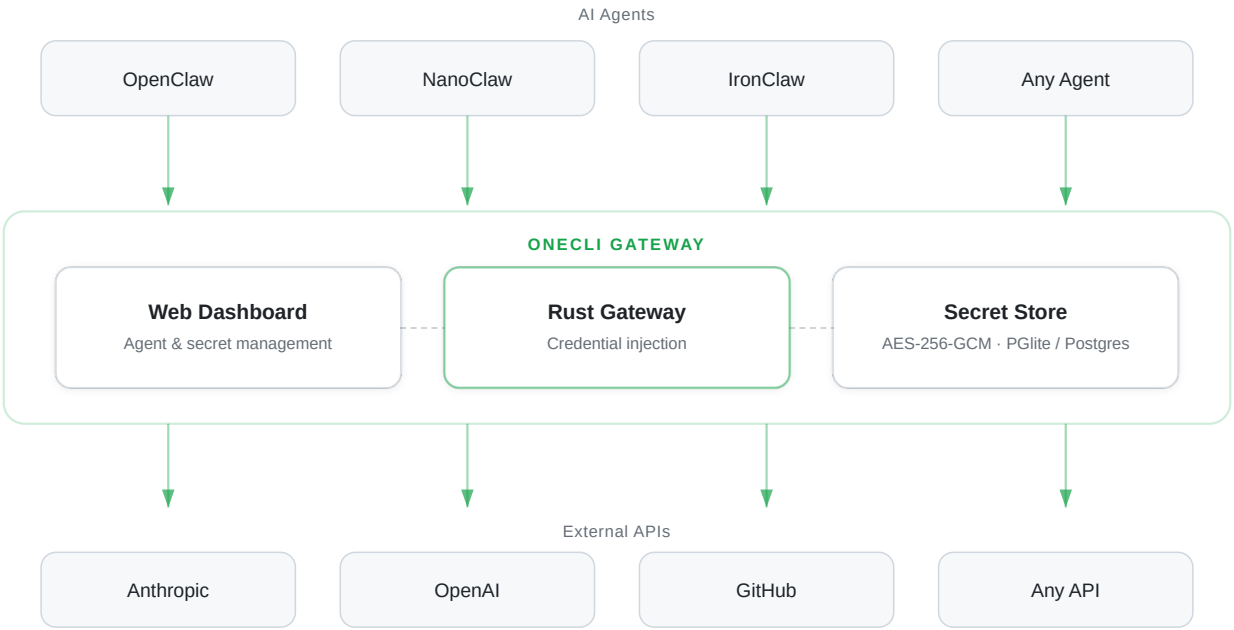
What is OneCLI?

OneCLI is an open-source gateway that sits between your AI agents and the services they call. Instead of baking API keys into every agent, you store credentials once in OneCLI and the gateway injects them transparently. Agents never see the secrets.

Why we built it: AI agents need to call dozens of APIs, but giving each agent raw credentials is a security risk. OneCLI solves this with a single gateway that handles auth, so you get one place to manage access, rotate keys, and see what every agent is doing.

How it works: You store your real API credentials in OneCLI and give your agents placeholder keys (e.g. `FAKE_KEY`). When an agent makes an HTTP call through the gateway, the OneCLI gateway matches the request to the right credentials, swaps the `FAKE_KEY` for the `REAL_KEY`, decrypts them, and injects them into the outbound request. The agent never touches the real secrets. It just makes normal HTTP calls and the gateway handles the swap.

Architecture



- **Rust Gateway**: fast HTTP gateway that intercepts outbound requests and injects credentials. Agents authenticate with access tokens via `Proxy-Authorization` headers.
- **Web Dashboard**: Next.js app for managing agents, secrets, and permissions. Provides the API the gateway uses to resolve which credentials to inject for each request.
- **Secret Store**: AES-256-GCM encrypted credential storage. Secrets are decrypted only at request time, matched by host and path patterns, and injected by the gateway as headers.

Quick Start

The fastest way to run OneCLI locally:

```
git clone https://github.com/onecli/onecli.git
cd onecli
docker compose -f docker/docker-compose.yml up
```

Open http://localhost:10254, create an agent, add your secrets, and point your agent's HTTP gateway to `localhost:10255`.

Features

- **Transparent credential injection**: agents make normal HTTP calls, the gateway handles auth
- **Encrypted secret storage**: AES-256-GCM encryption at rest, decrypted only at request time
- **Host & path matching**: route secrets to the right API endpoints with pattern matching
- **Multi-agent support**: each agent gets its own access token with scoped permissions
- **Easy setup**: `docker compose -f docker/docker-compose.yml up` starts everything (app + PostgreSQL)
- **Two auth modes**: single-user (no login) for local use, or Google OAuth for teams
- **Rust gateway**: fast, memory-safe HTTP gateway with MITM interception for HTTPS
- **Vault integration**: connect Bitwarden (or other password managers) for on-demand credential injection without storing secrets on the server

Project Structure

```
apps/
  web/            # Next.js app (dashboard + API, port 10254)
  gateway/        # Rust gateway (credential injection, port 10255)
packages/
  db/             # Prisma ORM + migrations
  ui/             # Shared UI components (shadcn/ui)
docker/
  Dockerfile      # App image (gateway + web)
  docker-compose.yml
```

Local Development

Prerequisites

- **mise** (installs Node.js, pnpm, and other tools)
- **Rust** (for the gateway)
- **Docker** (for PostgreSQL)

Setup

```
mise install
pnpm install
cp .env.example .env
pnpm db:generate
pnpm db:up          # Start PostgreSQL
pnpm db:migrate     # Apply migrations
pnpm dev
```

Dashboard at http://localhost:10254, gateway at http://localhost:10255.

Commands

Command	Description
`pnpm dev`	Start web + gateway in dev mode
`pnpm build`	Production build
`pnpm check`	Lint + types + format
`pnpm db:up`	Start PostgreSQL (Docker)
`pnpm db:down`	Stop PostgreSQL
`pnpm db:generate`	Generate Prisma client
`pnpm db:migrate`	Run database migrations
`pnpm db:studio`	Open Prisma Studio

Configuration

All environment variables are optional for local development:

Variable	Description	Default
`DATABASE_URL`	PostgreSQL connection string	See `.env.example`
`NEXTAUTH_SECRET`	Enables Google OAuth (multi-user)	Single-user mode
`GOOGLE_CLIENT_ID`	Google OAuth client ID	—
`GOOGLE_CLIENT_SECRET`	Google OAuth client secret	—
`SECRET_ENCRYPTION_KEY`	AES-256-GCM encryption key	Auto-generated

Contributing

We welcome contributions! Please read our Contributing Guide and Code of Conduct before getting started.