

readme.md for @serve.zone/cloudly

Multi-cloud orchestration made simple. Manage containerized applications across cloud providers with Docker Swarmkit, featuring web dashboards, CLI tools, and powerful APIs.

☐ What is Cloudly?

Cloudly is your command center for multi-cloud infrastructure. It abstracts away the complexity of managing resources across different cloud providers while giving you the power and flexibility you need for modern DevOps workflows.

☐ Key Features

- ☐ **Multi-Cloud Management** - Seamlessly orchestrate resources across Cloudflare, Hetzner, DigitalOcean and more
- ☐ **Docker Swarmkit Integration** - Native container orchestration with production-grade reliability
- ☐ **Secret Management** - Secure handling of credentials, API keys, and sensitive configuration
- ☐ **Web Dashboard** - Beautiful, responsive UI built with modern web components
- ☐ **CLI & API** - Full programmatic control through TypeScript/JavaScript APIs and command-line tools
- ☐ **SSL/TLS Automation** - Automatic certificate provisioning via Let's Encrypt
- ☐ **Comprehensive Logging** - Built-in log aggregation and monitoring capabilities
- ☐ **Task Scheduling** - Automated workflows and background job management

☐ Quick Start

Installation

```
# Install the main package
pnpm add @serve.zone/cloudly

# Or install the CLI globally
pnpm add -g @serve.zone/cli

# Or just the API client
pnpm add @serve.zone/api
```

Basic Setup

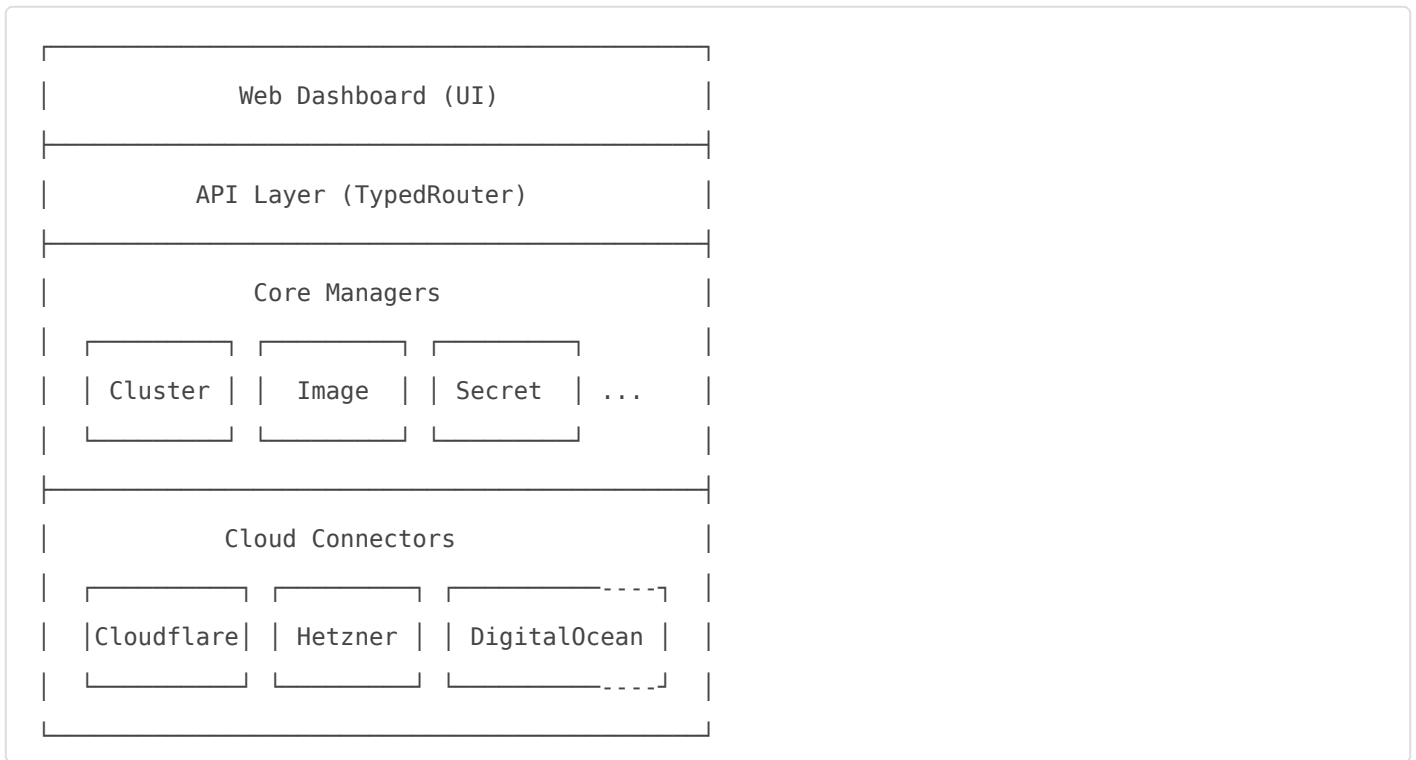
```
import { Cloudly } from '@serve.zone/cloudly';

// Initialize Cloudly with your configuration
const cloudly = new Cloudly({
  cfToken: process.env.CLOUDFLARE_TOKEN,
  hetznerToken: process.env.HETZNER_TOKEN,
  environment: 'production',
  letsEncryptEmail: 'certs@example.com',
  publicUrl: 'cloudly.example.com',
  publicPort: 443,
  mongoDescriptor: {
    mongoDbUrl: process.env.MONGODB_URL,
    mongoDbName: 'cloudly',
    mongoDbUser: process.env.MONGODB_USER,
    mongoDbPass: process.env.MONGODB_PASS,
  }
});

// Start the platform
await cloudly.start();
console.log('☑️ Cloudly is running!');
```

☑️ Architecture

Cloudly follows a modular architecture with clear separation of concerns:



☐☐ Core Components

☐☐ Managers

- **AuthManager** - Identity and access management
- **ClusterManager** - Docker Swarm cluster orchestration
- **ImageManager** - Container image lifecycle management
- **SecretManager** - Secure credential storage and distribution
- **ServerManager** - Cloud server provisioning and management
- **TaskManager** - Background job scheduling and execution

☐☐ Connectors

- **CloudflareConnector** - DNS, CDN, and edge services
- **LetsencryptConnector** - Automatic SSL certificate provisioning
- **MongodbConnector** - Database persistence layer
- **HetznerConnector** - German cloud infrastructure
- **DigitalOceanConnector** - Developer-friendly cloud resources

☐☐ Usage Examples

Managing Clusters

```
// Create a new cluster
const cluster = await cloudly.clusterManager.createCluster({
  name: 'production-cluster',
  region: 'eu-central',
  nodeCount: 3
});

// Deploy a service
await cluster.deployService({
  name: 'api-service',
  image: 'myapp:latest',
  replicas: 3,
  ports: [{ published: 80, target: 3000 }]
});
```

Secret Management

```
// Create a secret group
const secretGroup = await cloudly.secretManager.createSecretGroup({
  name: 'api-credentials',
  secrets: [
    { key: 'API_KEY', value: process.env.API_KEY },
    { key: 'DB_PASSWORD', value: process.env.DB_PASSWORD }
  ]
});

// Create a bundle for deployment
const bundle = await cloudly.secretManager.createSecretBundle({
  name: 'production-secrets',
  secretGroups: [secretGroup]
});
```

DNS Management

```
// Create DNS records via Cloudflare
const record = await cloudly.cloudflareConnector.createDNSRecord(
  'example.com',
  'api.example.com',
  'A',
  '192.168.1.1'
);
```

Web Dashboard

```
import { html } from '@design.estate/dees-element';

// Create a custom dashboard view
const dashboard = html`
  <cloudly-dashboard>
    <cloudly-view-clusters></cloudly-view-clusters>
    <cloudly-view-dns></cloudly-view-dns>
    <cloudly-view-images></cloudly-view-images>
  </cloudly-dashboard>
`;

document.body.appendChild(dashboard);
```

CLI Usage

The CLI provides quick access to all Cloudly features:

```
# Login to your Cloudly instance
servezone login --url https://cloudly.example.com

# List clusters
servezone clusters list

# Deploy a service
servezone deploy --cluster prod --image myapp:latest
```

```
# Manage secrets
servezone secrets create --name api-key --value "secret123"

# View logs
servezone logs --service api-service --follow
```

☐☐ Package Exports

This monorepo publishes multiple packages:

- **@serve.zone/cloudly** - Main orchestration platform
- **@serve.zone/api** - TypeScript/JavaScript API client
- **@serve.zone/cli** - Command-line interface
- **@serve.zone/interfaces** - Shared TypeScript interfaces

☐☐ Security Features

- **End-to-end encryption** for secrets
- **Role-based access control** (RBAC)
- **Automatic SSL/TLS** certificate management
- **Secure token-based authentication**
- **Audit logging** for compliance

☐☐ Production Ready

Cloudly is battle-tested in production environments managing:

- High-traffic web applications
- Microservice architectures
- CI/CD pipelines
- Data processing workloads
- Real-time communication systems

☐☐ Development

```
# Clone the repository
git clone https://gitlab.com/servezone/private/cloudly.git

# Install dependencies
pnpm install

# Run tests
pnpm test

# Build the project
pnpm build

# Start development mode
pnpm watch
```

Documentation

For detailed documentation, API references, and guides, visit our [documentation site](#).

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.

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.

Revision #3

Created 2026-03-28 11:14:29 UTC by foss.global Team

Updated 2026-03-28 12:21:15 UTC by foss.global Team