

@ship.zone/szci

Documentation for @ship.zone/szci

- [readme.md for @ship.zone/szci](#)
- [changelog.md for @ship.zone/szci](#)

readme.md for @ship.zone/szci

A lightweight CI/CD orchestrator built with Deno that unifies Node.js, Docker, NPM, SSH, and Git workflows into a single CLI. Compiles to standalone binaries — no runtime required.

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.

📦 Architecture

szci is a **thin orchestrator** — it doesn't reinvent the wheel. Instead, it wires up best-in-class tools and handles the CI-specific glue:

Domain	What szci does	Delegates to
📦 Docker	Bridges <code>SZCI_LOGIN_DOCKER*</code> env vars, auto-detects GitLab CI tokens	<code>@git.zone/tsdocker</code>
📦 NPM	Generates <code>.npmrc</code> from <code>SZCI_TOKEN_NPM*</code> env vars, handles multi-registry publish	<code>pnpm</code> + <code>npm publish</code>
📦 Node.js	Manages Node versions via NVM with named aliases	<code>nvm</code>
📦 SSH	Deploys SSH keys from env vars to <code>~/.ssh</code>	<code>@push.rocks/smartssh</code>
📦 Git	Mirrors repos to GitHub	<code>git remote</code>

CLI Input (Deno.args)

↓

SmartCLI Router

↓

```
└─ szci docker *   → env var bridging → npx @git.zone/tsdocker
└─ szci npm *      → .npmrc generation → pnpm / npm publish
└─ szci node *     → version aliasing  → nvm install
└─ szci git *      → token injection   → git push --mirror
└─ szci ssh *      → key parsing        → write to ~/.ssh
```

📦 Installation

Automated Installer (Recommended)

The easiest way to install szci is using the automated installer script. It handles platform detection, binary download, and global setup:

```
curl -sSL https://code.foss.global/ship.zone/szci/raw/branch/main/install.sh | sudo bash
```

To install a specific version:

```
curl -sSL https://code.foss.global/ship.zone/szci/raw/branch/main/install.sh | sudo bash -s --
--version v7.0.0
```

Via NPM (downloads pre-compiled binary)

```
npm install -g @ship.zone/szci
```

From Source

```
git clone https://code.foss.global/ship.zone/szci.git
cd szci
deno task compile
```

The compiled binaries land in `dist/binaries/` for Linux (x64/arm64), macOS (x64/arm64), and Windows (x64).

☐ Quick Start

```
# Setup Node.js environment
szci node install stable

# Install project dependencies
szci npm install

# Build & push Docker images
szci docker build
szci docker push registry.example.com

# Run tests
szci npm test
```

☐ CLI Reference

szci docker — Docker Operations

All Docker commands delegate to `@git.zone/tsdocker` after bridging environment variables.

```
szci docker build           # Build all Dockerfiles in cwd
szci docker login          # Login to all configured registries
szci docker prepare        # Alias for login
szci docker push registry.example.com # Push images to a registry
szci docker pull registry.example.com # Pull images from a registry
szci docker test           # Test Dockerfiles
```

Env var bridging: Before delegating, szci converts its own env var format to tsdocker's expected format:

```
SZCI_LOGIN_DOCKER_1 → DOCKER_REGISTRY_1
SZCI_LOGIN_DOCKER_2 → DOCKER_REGISTRY_2
...
```

In GitLab CI, `CI_JOB_TOKEN` is automatically bridged as `DOCKER_REGISTRY_0` for `registry.gitlab.com`.

szci npm — NPM/pnpm Workflows

```
szci npm install      # Runs pnpm install
szci npm build        # Runs pnpm run build
szci npm test         # Runs pnpm test
szci npm prepare      # Generates ~/.npmrc from SZCI_TOKEN_NPM* env vars
szci npm publish      # Full workflow: prepare → install → build → clean → npm publish
```

The `publish` command supports multi-registry publishing. If `npmAccessLevel` is `public` and a Verdaccio registry is configured, it publishes to both npm and Verdaccio automatically.

szci node — Node.js Version Management

```
szci node install stable # Node.js 22
szci node install lts    # Node.js 20
szci node install legacy # Node.js 18
szci node install 21     # Any specific version
```

Uses NVM under the hood (auto-detected at `/usr/local/nvm/nvm.sh` or `~/.nvm/nvm.sh`). After installing, it:

1. Sets the installed version as `nvm alias default`
2. Upgrades npm to latest
3. Installs any global tools listed in `npmextra.json` → `npmGlobalTools`

szci ssh — SSH Key Deployment

```
szci ssh prepare # Deploy SSH keys from env vars to ~/.ssh
```

Reads all `SZCI_SSHKEY_*` env vars and writes the keys to disk.

szci git — Git Mirroring

```
szci git mirror # Mirror repository to GitHub
```

Pushes all branches and tags to a GitHub mirror. Requires `SZCI_GIT_GITHUBTOKEN`. Refuses to mirror packages marked as `private` in `package.json`.

⚙️ Configuration

npmextra.json

Place this in your project root to configure szci behavior:

```
{
  "@ship.zone/szci": {
    "npmGlobalTools": ["typescript", "pnpm"],
    "npmAccessLevel": "public",
    "npmRegistryUrl": "registry.npmjs.org",
    "urlCloudly": "https://cloudly.example.com"
  }
}
```

Option	Type	Default	Description
npmGlobalTools	string[]	[]	Global npm packages to install during <code>node install</code>
npmAccessLevel	"public" "private"	"private"	Access level for <code>npm publish</code>
npmRegistryUrl	string	"registry.npmjs.org"	Default npm registry
urlCloudly	string?	—	Cloudly endpoint URL (can also be set via <code>SZCI_URL_CLOUDLY</code>)

📁 Environment Variables

Docker Registry Authentication

Pipe-delimited format: `registry|username|password`

```
SZCI_LOGIN_DOCKER_1="registry.example.com|myuser|mypass"
SZCI_LOGIN_DOCKER_2="ghcr.io|token|ghp_xxxx"
```

In **GitLab CI**, the `CI_JOB_TOKEN` is automatically used for `registry.gitlab.com` — no manual config needed.

NPM Registry Authentication

Pipe-delimited format: `registry|token[|plain]`

```
# Base64-encoded token (default)
SZCI_TOKEN_NPM_1="registry.npmjs.org|dGhlLXRva2VuLWhlcmU="

# Plain text token
SZCI_TOKEN_NPM_2="verdaccio.example.com|the-token-here|plain"
```

SSH Keys

Pipe-delimited format: `host|privKeyBase64|pubKeyBase64`

```
SZCI_SSHKEY_1="github.com|BASE64_PRIVATE_KEY|BASE64_PUBLIC_KEY"
SZCI_SSHKEY_2="gitlab.com|BASE64_PRIVATE_KEY|##" # Use ## to skip a field
```

Git Mirroring

```
SZCI_GIT_GITHUBTOKEN="ghp_your_personal_access_token"
SZCI_GIT_GITHUBGROUP="your-org" # Defaults to repo owner
SZCI_GIT_GITHUB="your-repo" # Defaults to repo name
```

Debugging & Testing

```
DEBUG_SZCI="true" # Log all shell commands before execution
SZCI_TEST="true" # Test mode: mocks bash execution, skips SSH disk writes
```

Full Environment Variable Reference

Variable	Purpose
<code>SZCI_LOGIN_DOCKER_*</code>	Docker registry credentials (pipe-delimited)
<code>SZCI_TOKEN_NPM_*</code>	NPM registry auth tokens (pipe-delimited)
<code>SZCI_SSHKEY_*</code>	SSH key pairs (pipe-delimited)

Variable	Purpose
SZCI_GIT_GITHUBTOKEN	GitHub personal access token for mirroring
SZCI_GIT_GITHUBGROUP	GitHub org/user for mirror target
SZCI_GIT_GITHUB	GitHub repo name for mirror target
SZCI_URL_CLOUDLY	Cloudly endpoint URL
SZCI_COMPUTED_REPOURL	Override auto-detected repo URL
DEBUG_SZCI	Enable verbose shell command logging
SZCI_TEST	Enable test mode (mock execution)

☐ CI/CD Integration Examples

GitLab CI

```
image: node:22

stages:
  - prepare
  - build
  - test
  - deploy

before_script:
  - curl -sSL https://code.foss.global/ship.zone/szci/raw/branch/main/install.sh | bash

prepare:
  stage: prepare
  script:
    - szci node install stable
    - szci npm install

build:
  stage: build
  script:
    - szci docker build
```

```
test:
  stage: test
  script:
    - szci npm test

deploy:
  stage: deploy
  script:
    - szci docker push $CI_REGISTRY
only:
  - master
```

“ In GitLab CI, `CI_JOB_TOKEN` is automatically detected — `szci` will login to `registry.gitlab.com` without any extra config.

GitHub Actions

```
name: CI/CD

on:
  push:
    branches: [main]

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4

      - name: Install szci
        run: curl -sSL https://code.foss.global/ship.zone/szci/raw/branch/main/install.sh |
sudo bash

      - name: Setup Node.js
        run: szci node install stable

      - name: Install & Build
```

```

run: |
  szci npm install
  szci npm build

- name: Test
  run: szci npm test

- name: Push Docker
  if: github.ref == 'refs/heads/main'
  run: szci docker push ghcr.io
  env:
    SZCI_LOGIN_DOCKER_1: "ghcr.io|${{ github.actor }}|${{ secrets.GITHUB_TOKEN }}"

```

Gitea CI (Woodpecker)

```

steps:
  - name: ci
    image: node:22
    commands:
      - curl -sSL https://code.foss.global/ship.zone/szci/raw/branch/main/install.sh | bash
      - szci node install stable
      - szci npm install
      - szci docker build
      - szci npm test

```

📦 Migration from npmc

Upgrading from `@ship.zone/npmci`? Three steps:

- Rename the binary** — replace `npmci` with `szci` in all CI scripts
- Rename env vars** — `NPMCI_*` → `SZCI_*` (same formats, just the prefix changed)
- Docker ops** — now delegate to `@git.zone/tsdocker` (installed automatically via `npm`)

Old	New
<code>NPMCI_LOGIN_DOCKER*</code>	<code>SZCI_LOGIN_DOCKER*</code>
<code>NPMCI_TOKEN_NPM*</code>	<code>SZCI_TOKEN_NPM*</code>
<code>NPMCI_SSHKEY_*</code>	<code>SZCI_SSHKEY_*</code>

Old	New
NPMCI_GIT_GITHUBTOKEN	SZCI_GIT_GITHUBTOKEN
NPMCI_URL_CLOUDLY	SZCI_URL_CLOUDLY
DEBUG_NPMCI	DEBUG_SZCI
NPMTS_TEST	SZCI_TEST

Development

```
# Type check
deno task check

# Run tests
deno task test

# Watch tests
deno task test:watch

# Dev mode
deno task dev -- --help

# Compile binaries for all platforms
deno task compile

# Format code
deno task fmt

# Lint
deno task lint
```

License and Legal Information

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

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 or third parties, 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 or the guidelines of the respective third-party owners, and any usage must be approved in writing. Third-party trademarks used herein are the property of their respective owners and used only in a descriptive manner, e.g. for an implementation of an API or similar.

Company Information

Task Venture Capital GmbH

Registered at District Court Bremen HRB 35230 HB, Germany

For any legal inquiries or 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 @ship.zone/szci

2026-02-08 - 7.1.2 - fix(readme)

Correct license link and fix markdown line break for company name in README

- Updated README link to the license file from `./license` to `./LICENSE`
- Added trailing spaces after 'Task Venture Capital GmbH' to preserve a Markdown line break
- Documentation-only change; no code modified

2026-02-07 - 7.1.1 - fix(workflows)

normalize default branch to 'main' across CI/release workflows and docs; update package metadata; clean `.gitignore` and fix readme license link

- Updated workflow triggers and documentation from 'master' to 'main' (`.gitea/workflows/ci.yml`, `.gitea/workflows/release.yml`, `.gitea/workflows/README.md`) including install script URLs referenced in release instructions.
- Updated `npmextra.json` @`git.zone/cli` metadata: changed `projectType` from 'npm' to 'deno' and `ghosthost` from 'gitlab.com' to 'code.foss.global'.
- Cleaned `.gitignore` by removing outdated entries (`coverage/`, `public/`, `pages/`, `.yarn/`, `.rpt2_cache`) and minor whitespace fixes.
- Normalized license filename link in `readme.md` from 'LICENSE' to lowercase 'license'.

2026-02-07 - 7.1.0 - feat(installer)

switch installer/docs to use main branch; add automated installer instructions and update CI examples to use the installer; add release registries and access level in `npmextra.json`

- Replaced master branch URLs with main in `install.sh` and README examples

- Added "Automated Installer (Recommended)" section to README with curl install examples (including versioned install)
- Replaced npm install -g @ship.zone/szci in CI examples with the curl-based installer invocation (some uses sudo)
- Updated npmextra.json: removed npmAccessLevel for @ship.zone/szci and added release.registries and release.accessLevel entries

2026-02-06 - 7.0.0 - BREAKING CHANGE(szci)

delegate Docker operations to @git.zone/tsdocker, remove internal Docker managers and deprecated modules, simplify CLI and env var handling

- Delegate all Docker actions to @git.zone/tsdocker via npx; Szc/dockerManager now bridges SZCI_LOGIN_DOCKER* and CI_JOB_TOKEN to DOCKER_REGISTRY_N and invokes npx @git.zone/tsdocker for build/login/push/pull/test.
- Removed internal Docker implementation: Dockerfile, DockerRegistry, RegistryStorage, related helpers and plugin wrappers removed from ts/dockerManager.
- Removed Cloudrone manager and other deprecated modules and their plugin shims: manager.cloudrone, mod_clean, mod_command, mod_precheck, mod_trigger (and corresponding CLI commands: cloudrone, clean, command, precheck, trigger).
- CLI and exports simplified: Dockerfile export removed from mod.ts; Szc CLI now delegates docker command to the simplified Szc/dockerManager.
- Updated environment handling: bridges SZCI_LOGIN_DOCKER* → DOCKER_REGISTRY_N and auto-bridges GitLab CI CI_JOB_TOKEN to DOCKER_REGISTRY_0.
- Node.js default mappings updated: stable→22, lts→20, legacy→18.
- Dependencies and plugins cleaned-up: removed unused/obsolete deps (e.g. @push.rocks/lik, smartdelay, through2) from deno.json and szci/plugins.ts.
- Docs updated (readme.md, readme.hints.md) to reflect architecture, tsdocker delegation, env var bridging and migration notes from npmci.
- BREAKING: CI configs and any workflows relying on internal Docker classes or the removed CLI commands must be updated to use tsdocker and the new env var/command flows.

2025-10-26 - 6.0.1 - fix(tests)

Migrate tests to Deno native runner and update Deno config

- Convert test suites from tap/tapbundle to Deno.test and @std/assert

- Replace CommonJS-style runtime imports with .ts module imports for Deno (test files updated)
- Use Deno.env.set to configure test environment variables and restore working directory after tests
- Update test/test.cloudly.ts to import CloudlyConnector directly and disable TLS verification for tests
- Adjust deno.json version field (6.0.0 -> 5.0.0) as part of Deno configuration changes
- Add local project .claude/settings.local.json for tooling permissions

2025-10-26 - 6.0.0 - BREAKING CHANGE(szci)

Rename project from npmci to szci and migrate runtime to Deno; add compiled binaries, installer and wrapper; update imports, env handling and package metadata

- Major rename/refactor: Npmci -> Szci across the codebase (classes, filenames, modules and exports).
- Migrate runtime to Deno: add deno.json, mod.ts entry point, use Deno.std imports and .ts module imports throughout.
- Add compilation and distribution tooling: scripts/compile-all.sh, scripts/install-binary.js, bin/szci-wrapper.js and dist/binaries layout for prebuilt executables.
- Package metadata updated: package.json renamed/rewritten for @ship.zone/szci and bumped to 5.0.0, updated publishConfig, files and scripts.
- Environment API changes: replaced process.env usages with Deno.env accessors and updated path constants (Szci paths).
- Refactored helper modules: npmci.bash -> szci.bash, updated smartshell/bash wrappers and other manager modules to Deno patterns.
- Tests and imports updated to new module paths and package layout; .gitignore updated to ignore deno artifacts.
- Breaking changes to callers: CLI name, class names, programmatic API, binary installation and environment handling have changed and may require updates in integrations and CI configurations.

2024-11-17 - 4.1.37 - fix(docker)

Enhanced base image extraction logic from Dockerfile

- Improved dockerBaseImage to accurately extract base images considering ARG variables.
- Added support for parsing Dockerfile content without external libraries.

- Enhanced error handling for missing FROM instructions.

2024-11-17 - 4.1.36 - fix(docker)

Improve logging for Dockerfile build order with base image details.

- Enhance logging in Dockerfile sorting process to include base image information.

2024-11-17 - 4.1.35 - fix(docker)

Fix Dockerfile dependency sorting and enhance environment variable handling for GitHub repos

- Refined the algorithm for sorting Dockerfiles based on dependencies to ensure proper build order.
- Enhanced environment variable handling in the NpmciEnv class to support conditional assignments.
- Updated various dependencies in package.json for improved performance and compatibility.
- Added error handling to circular dependency detection in Dockerfile sorting.

2024-11-05 - 4.1.34 - fix(connector)

Remove unused typedrequest implementation in cloudlyconnector

- Removed commented out code that initialized typedrequest in CloudlyConnector.

2024-11-05 - 4.1.33 - fix(core)

Updated dependencies and improved npm preparation logic.

- Updated @git.zone/tsbuild from ^2.1.84 to ^2.2.0.
- Updated @git.zone/tsrun from ^1.2.49 to ^1.3.3.
- Updated @types/node from ^22.7.9 to ^22.8.7.
- Updated @serve.zone/api from ^1.2.1 to ^4.3.1.

- Improved npm preparation logic to handle empty tokens gracefully.

2024-10-23 - 4.1.32 - fix(dependencies)

Update project dependencies to latest versions

- Updated development dependencies, including @git.zone/tsbuild and @git.zone/tsrun.
- Updated production dependencies such as @api.global/typedrequest and @push.rocks/smartfile.

2022-10-24 - 4.0.11 - prerelease

now includes a precheck for more generic runner execution

- Implemented a precheck feature for runners.

2022-10-09 to 2022-10-11 - 4.0.0 to 4.0.10 - migration

internal migrations and fixes

- Major switch to ESM style module: **BREAKING CHANGE**.
- Multiple fixes in core functionalities and module updates.

2019-11-26 - 3.1.73 - fixes

correctly setting npm cache and other updates

- Ensured correct npm cache setting during preparation.
- Various core updates.

2018-12-23 - 3.1.19 - privacy updates

enhanced mirroring controls for private code

- Now refusing to mirror private code.

2018-11-24 - 3.1.2 - ci improvement

removed unnecessary build dependency

- Removed npmts build dependency in CI pipeline.

2018-09-22 - 3.0.59 - enhancement

integrated smartlog for improved logging

- Logs now utilize smartlog for better management.

2017-09-08 - 3.0.14 - analytics

added analytics features

- Enabled analytics throughout the system.

2017-08-29 - 3.0.9 - docker enhancements

docker improvements and build args implementation

- Implemented working `dockerBuildArgEnvMap`.

2017-07-27 - 2.4.0 - stability improvements

various updates to stabilize the environment

- Fixed npmci versioning issues.

2016-11-25 - 2.3.24 - global tools

improved handling for global tool installations

- Improved install handling for needed global tools.