



Infisical Project Structure

The open-source secret management platform. A full-stack application featuring a Node.js/Express backend, a React (Vite) frontend, and specialized infrastructure components.

Updated 2025-12-30

#infisical #react #vite #nodejs #express #security #open-source

 PNG

PDF PDF

 Copy

`</>` Prompt

Project Directory



[infisical/](#)

```

  ✓  📁 frontend/  Dashboard (React...)
    📁 src/  React applicati...
    📄 vite.config.ts  Vite configurat...
    📄 Dockerfile  Frontend contain...
  ✓  📁 backend/  API Server (Node.js)
    📁 src/  Server logic
    📄 package.json  Backend dependencies
    📄 Dockerfile  Backend container...
  📁 cli/  Go-based CLI tool
  📁 helm-charts/  Kubernetes deployment...
  📄 docker-compose.yml  Local development...
  📄 README.md  Project documentation

```

Repository Info

Repository - Infisical/infisical

Stars - 12k+

License - MIT

Last Analyzed - December 2025

 Tech Stack

Frontend – React with Vite

Backend – Node.js (Express/Fastify)

Database – MongoDB / PostgreSQL

Infrastructure – Docker & Kubernetes

Architecture Notes

Infisical uses a classic 'Split Stack' architecture rather than a tight monorepo. The `frontend` and `backend` are distinct top-level directories with their own build processes (Vite for frontend, standard Node build for backend). This is a very common pattern for teams that want clear separation of concerns.

Key Directories

frontend/ - The React-based dashboard for managing secrets

backend/ - The core API server handling encryption and storage

helm-charts/ - Production deployment configurations for Kubernetes

💡 Why This Structure?

A great reference for a standard, production-grade Full Stack JavaScript application. It shows how to organize a non-monorepo full-stack project where the frontend and backend are effectively separate applications living in the same repo.