

Cloud9 Project Overview

Secure, Scalable, AI-Powered File Management

Introduction



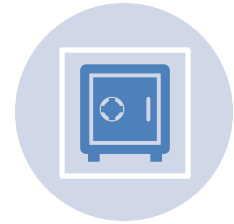
SECURE, SCALABLE,
INTELLIGENT FILE
MANAGEMENT



UPLOAD/STORE ANY
FILE TYPE



BUILT-IN VIEWERS
(IMAGE, VIDEO)



SECURE FILE SHARING,
PUBLIC HOSTING



AI-POWERED
SEMANTIC SEARCH

Key Features

User authentication with JWT

Folder-based UI, intuitive management

HTML5 video/image viewers

Signed links for sharing

AI semantic search

Email document distribution

Cloud-native storage options

AI Semantic Search: How It Works

- Content extraction: text, image, video, audio
- LLM analysis: keywords, topics, sentiment, summaries
- Embeddings with HuggingFace models
- Indexed in MongoDB for fast/private search

```
const resetState = () => {  
  setSearchText("");  
  setDebouncedSearchText("");  
};  
  
const onSemanticSearch = async () => {  
  if (!searchText) return;  
  setDebouncedSearchText(searchText); // Ensure dropdown height 100px  
  setIsSemantic(true);  
  setShowSuggestions(true);  
  setAiLoading(true);  
  console.log("AI search triggered with:", searchText);  
  try {  
    const results = await semanticSearchAPI(searchText);  
    setSemanticResults({  
      fileList: results.files || [],  
      folderList: results.folders || [],  
    });  
  } catch (e) {  
    setSemanticResults({ fileList: [], folderList: [] });  
  }  
  setAiLoading(false);  
};  
  
const outOfContainerClick = useCallback(() => {  
  closeDrawer();  
  setShowSuggestions(false);  
}, []);  
  
const { wrapperRef } = useClickOutOfBounds(outOfContainerClick);  
  
const onSearch = (e: any) => {  
  e.preventDefault();  
  setShowSuggestions(false);  
  if (isMedia) {  
    if (searchText.length) {  
      navigate(`/search-media/${searchText}`);  
    } else {  

```

Semantic Search Process & Accuracy



Query embedded
and matched via
cosine similarity



Search by meaning,
context, sentiment



Robust to typos and
vague queries



Uses private
MongoDB, not Azure
AI Search

Tech Stack

Frontend: React, Tailwind CSS, Vite

Backend: Node.js, Express, TypeScript

Database: MongoDB/DocumentDB

Auth: JWT | Storage: Local, AWS S3, Azure Blob

AI: OpenAI, HuggingFace, AssemblyAI

Media: PDF.js, Tesseract.js, ffmpeg

Deployment & Architecture



Cloud-native,
Dockerized



CI/CD ready,
environment-based
config



Horizontally scalable



Supports Azure, AWS,
others

Getting Started



Get API keys: NScale,
HuggingFace, AssemblyAI

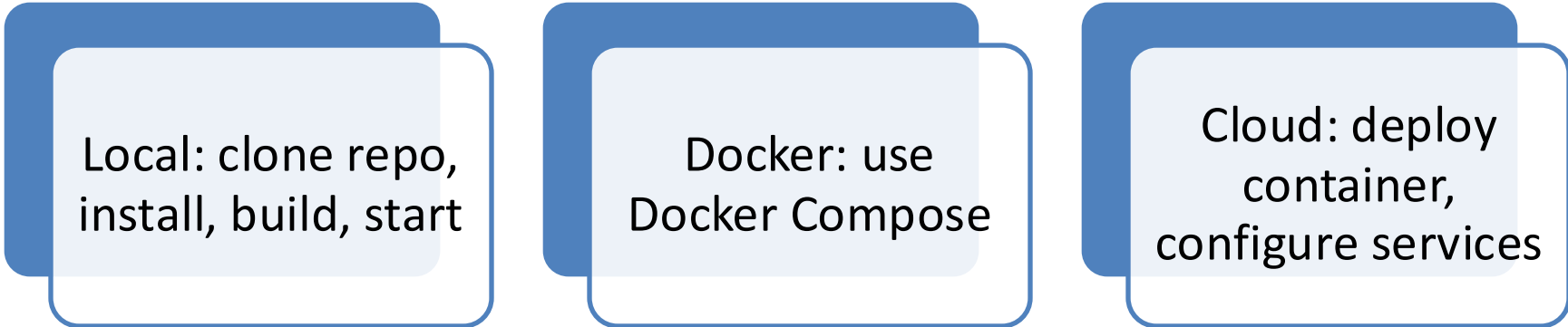


Set environment
variables (.env)



Requirements: Node.js >=
18, MongoDB, Git

Development & Deployment



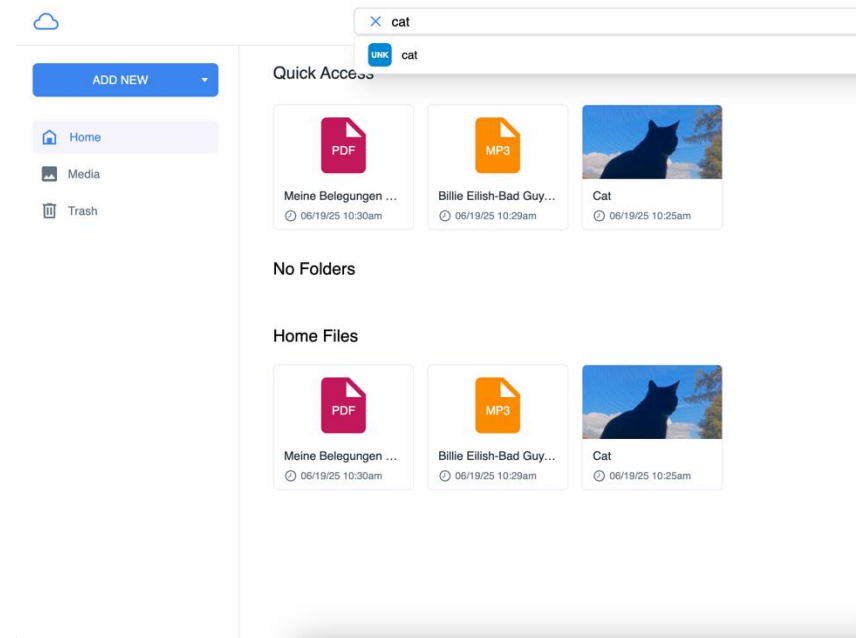
Local: clone repo,
install, build, start

Docker: use
Docker Compose

Cloud: deploy
container,
configure services

User Guide

- Sign Up & Login: Create account, JWT-secured login
- Upload Files: Drag & drop or select files, organize into folders
- View & Manage Files: In-browser viewing, rename/move/delete
- Share Securely: Signed links with permissions and expiry
- Use AI Semantic Search: Natural language search by meaning
- Email Distribution: Send documents directly via email
- Host Public Files: Public links for certificates, etc.



Pipeline Overview

- CI/CD Pipeline:
 - Code Commit: Push to Git
- Build: Vite + TypeScript build process
- Test: Run automated tests
- Dockerize: Create Docker images
- Deploy: Push to cloud containers (Azure, AWS)
- AI Processing Pipeline:
 - File Upload: Triggers AI pipeline
 - Content Extraction: Text, image, video, audio
 - LLM Analysis: Summaries, keywords, sentiment
 - Embedding: Generate vectors with HuggingFace
 - Indexing: Store semantic data in MongoDB

