

# 知识库管理界面需求

# 核心痛点与挑战

02



## 硬件兼容性

昇腾与英伟达生态差异大，部署复杂。



## 组件碎片化

模型、向量库、解析服务缺乏统一入口。



## 运维不可靠

缺乏可视化状态监控与便捷备份手段。

# 产品核心模块架构



## 状态监管

健康检查  
资源占用  
控制启停



## 部署配置

部署向导  
环境选择  
脚本导出



## 服务管理

模型配置  
参数调优  
对接服务



## 备份恢复

全量/增量  
数据重刷  
完整性校验



## 系统管理

角色权限  
JWT鉴权  
审计日志

底层支持: Docker / K8s / 异构芯片 (Ascend & NVIDIA)

# 核心特性：国产化适配与部署向导

04

1

## 硬件生态自适应

智能识别英伟达或昇腾硬件，自动匹配 vLLM 或 MindIE 推理引擎。

2

## 图形化配置向导

分步引导完成管理端、模型端、向量库及解析服务的配置，支持配置包导出。

3

## 无缝对接现有服务

支持已有服务的地址、端口配置与连接测试，灵活扩展。

[部署流程可视化插图]

模式选择

>

配置服务

>

生成/部署

# 服务管理：跨组件的统一调度中心

## 推理服务管理

管理 vLLM/MindIE 实例状态，监控 Token 吞吐量及任务队列。

## 向量库管理

监控集合(Collection)状态、连接池及磁盘存储水位。

## 文档解析管理

解析引擎健康度检测，支持 PDF/Word 等多格式异步处理监控。

service\_status\_dashboard.sh

[推理引擎] llama-3-8b-instruct

RUNNING

[向量库] milvus-standalone

HEALTHY

[解析服务] doc-parser-api

BUSY (Queue: 12)

NPU/GPU负载

68%

内存占用

14.2 GB

重启所有服务

导出日志

# 模型配置管理：精细化参数控制

05

## 推理参数矩阵

### 并发限制

Max Concurrency

控制单一模型最大请求数

### Token限制

Input/Output Limit

上下文窗口与生成长度

### Batch策略

Dynamic Batching

动态批处理大小与延迟

### 资源分配

Compute Strategy

显存分配与计算卡亲和性

**提示：** 修改核心计算资源或推理参数时，管理界面将明确提示“需要重启服务”并提供平滑迁移建议。

## 生命周期管理

### 模型加载

支持从本地路径或远程仓库拉取权重。

### 配置热更新

部分推理参数支持在线热更新（不重启）。

### 版本切换

支持 A/B 测试模式下的模型版本平滑切换。

### 配置导出

一键导出配置模板，支持快速克隆环境。

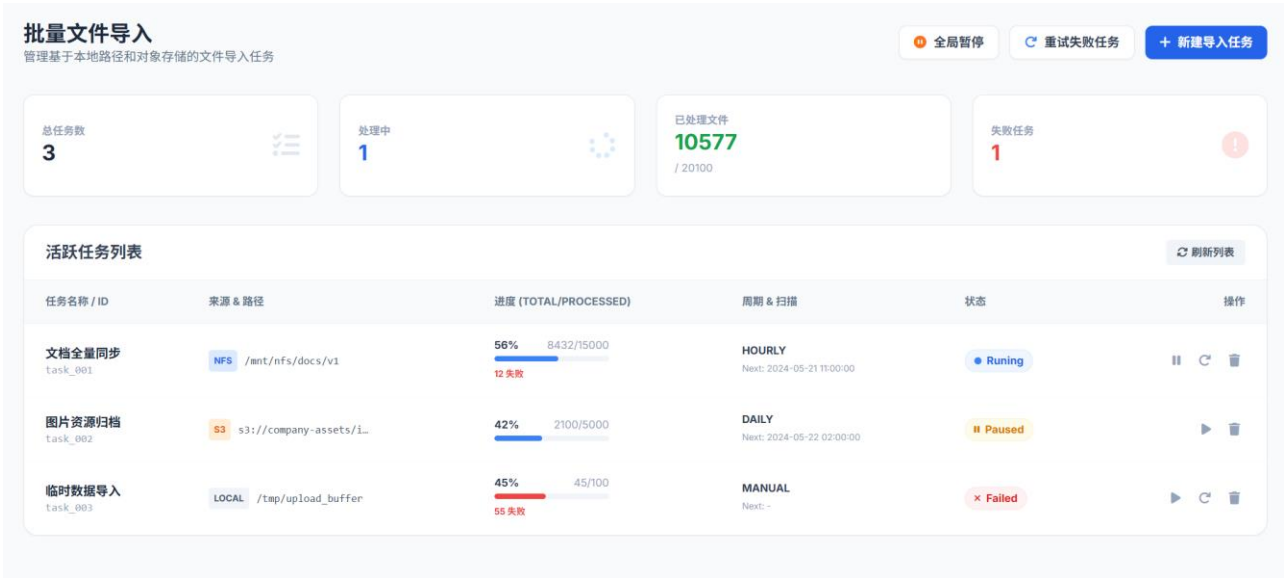
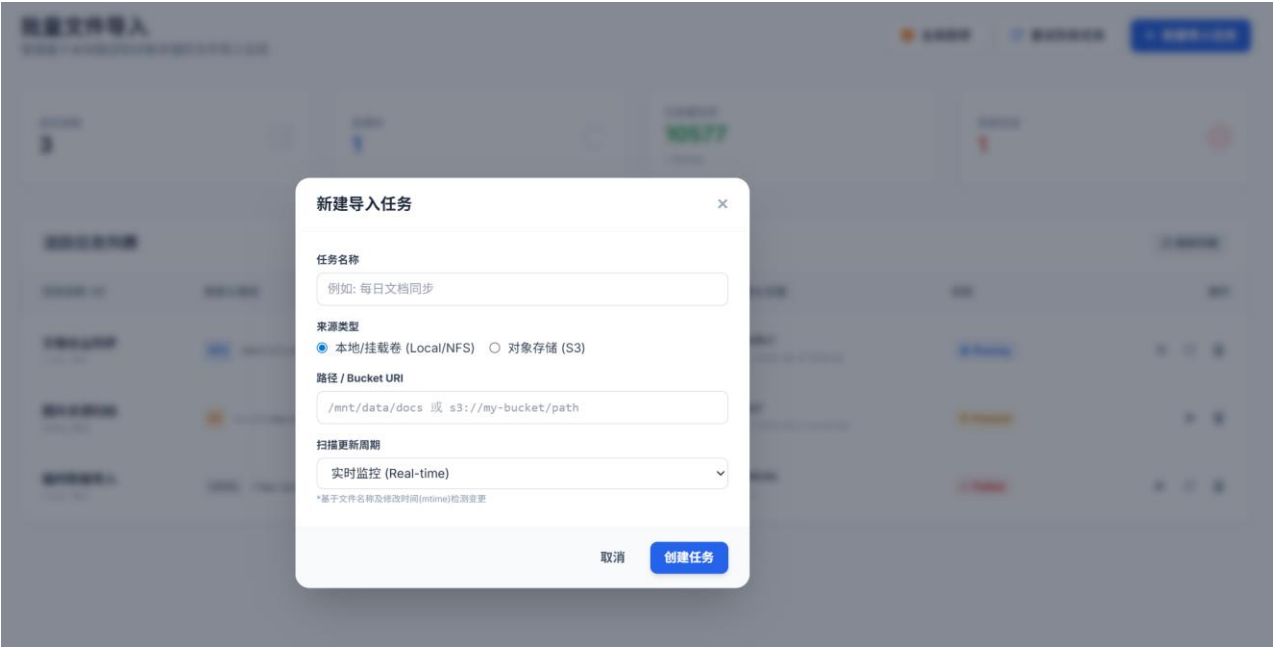
MVP SOLUTION OUTLINE

数据批量导入

- 从文件系统路径（本地、NFS）及对象存储路径批量导入数据。
- 提供专用的批量导入面板，支持状态显示、可断点续传的传输机制，以及基于文件名与修改日期（校验和）的变化检测功能。

具体要求：

- 状态显示** – 提供任务级别总览，显示文件总数与已处理数量。支持以下操作：
  - 全部任务的重新开始/暂停。
  - 单个任务的开始/暂停。
  - 失败任务的重新开始/清除。
- 变化检测** – 支持启用/禁用扫描功能，并允许配置扫描间隔：
  - 可选模式：实时、每小时、每日、每周、每月。
  - 提供用于管理备份周期的控制面板。



# 数据安全：闭环备份恢复体系

备份维度	实现方案	业务场景
全量备份	手动触发，整合数据库与文件系统快照	重大版本更新、环境迁移
增量备份	基于时间点触发，极简差异备份	日常数据变更记录（5min内完成）
应用重刷	镜像重新部署 + 备份点快速恢复	系统故障紧急修复、环境重置

**安全性：** 敏感信息加密存储 (SHA256)，支持 HTTPS 数据导出。

**可靠性：** 强制备份验证，确保恢复点真实可用。



# 实施路径与开发优先级

## Phase 1: 核心可用 (P0)

- 国产化硬件 (昇腾/英伟达) 基础适配
- 基础模型参数配置与服务对接
- 全量备份与基础恢复功能
- 关键服务状态实时监控与生命周期管理
- 用户登录鉴权与基础操作审计
- 数据批量加载 (本地盘、NFS、S3)



## Phase 2: 体验优化 (P1)

- 统一用户管理及单一登录对接
- 兼容Terraform基 配置文件 Infrastructure as code 部署方式
- 多维度资源看板 (硬件资源及知识库服务使用的深度分析)
- 高级诊断日志与故障恢复指引

- 参考样例: <https://github.com/kt-chan/anyadmin>

KB-Manager

● 系统正常运行

核心监控

状态监管仪表盘

组件管理

部署配置向导

服务与模型管理

批量文件导入

运维安全

备份与恢复

系统管理 (用户)

退出登录

服务监控仪表盘

实时查看集群健康度与资源利用率

刷新频率: 15s

系统状态良好

100% Online

运行中服务 8 / 8

NPU (Ascend)

算力负载 64.2%

14.2 / 32 GB

显存/内存占用 44.3%

Normal Queue

解析任务队列 12

服务运行清单

管理所有 >

服务名称	角色/类型	健康状况	操作
推理引擎 (llama-3-8b)	vLLM / MindIE	运行中	🔄 ⏻
向量库 (Milvus)	Vector DB	健康	🔄 ⏻
文档解析引擎	Doc Parser	任务处理中	🔄 ⏻
bge-large-zh-v1.5	Embedding	运行中	🔄 ⏻

数据备份与灾备恢复

上次备份 2024-05-20 02:00:00 (增量备份)

立即执行全量备份

快速恢复 可用备份点: 12 个

从最近备份点恢复

模型推理配置

MAX CONCURRENCY 并发限制 1 64 (Current) 256

TOKEN LIMIT 上下文长度 8,192 Tokens

动态批处理 (Dynamic Batching) ON

硬件加速类型 昇腾 MindIE

保存配置并重启服务

注: 更改推理参数需重启服务, 预计中断 15s

最近操作审计

Admin 修改了推理并发数 10分钟前 · 终端: 192.168.1.102

系统自检 全量备份完成 今天 02:00 · 自动化任务

File Edit Selection View Go Run Terminal Help

Run Full Run Frontend (anyadmin) Run Backend (anyadmin) Debug Frontend Tests (test) (anyadmin) Debug Backend Tests (Pytest) (anyadmin) Run Full Stack (anyadmin)

Node.js... Python Debugger... Add Config (anyadmin)... Add Config (workspace)...

知识库管理平台 MVP - 已成功启动

地址: http://localhost:3000/dashboard  
服务管理: http://localhost:3000/services  
部署配置: http://localhost:3000/deployment  
备份恢复: http://localhost:3000/backup  
系统管理: http://localhost:3000/system

登录凭据:  
- 管理员: admin / password  
- 操作员: operator-01 / password

[DEBUG] 2026-01-22T02:28:57.049Z: Attempting login for user: admin {}  
[INFO] 2026-01-22T02:28:57.096Z: User logged in successfully: admin {}  
[DEBUG] 2026-01-22T02:28:57.099Z: Fetching dashboard overview data {}  
[DEBUG] 2026-01-22T02:22:13.625Z: Fetching services list {}  
[DEBUG] 2026-01-22T02:22:21.183Z: Fetching dashboard overview data {}  
[DEBUG] 2026-01-22T02:24:06.182Z: Fetching dashboard overview data {}  
[DEBUG] 2026-01-22T02:24:06.283Z: Fetching dashboard overview data {}  
[DEBUG] 2026-01-22T02:24:06.383Z: Fetching dashboard overview data {}  
[DEBUG] 2026-01-22T02:24:30.020Z: Fetching dashboard overview data {}