Платформа документации провайдеров Terraform

Автоматизированный сбор, обработка и векторизация документации провайдеров Terraform для управления инфраструктурой на основе ИИ

ОБЗОР ПРОЕКТА

Цель проекта

Создание аналитической платформы для автоматизированного сбора, очистки, предварительной обработки, маркировки и хранения документации Terraform provider для обеспечения AI агента актуальными, структурированными данными для генерации корректных Terraform configurations.

Платформа решает критические проблемы в DevOps automation, где AI агенты, управляющие инфраструктурой, требуют надежной, версионированной документации для множества Terraform providers.

Основные возможности

- *Автоматизированный web scraping of provider docs*
- *Конвертация HTML to structured text*
- *Организация данных с учетом Version-aware*
- *Генерация Vector embedding*
- *Quality assurance pipelines*

Проблема бизнеса

Область: *DevOps / Infrastructure as Code (Terraform)*

Фрагментированные источники

Документация, разбросанная по многочисленным реестрам, официальным сайтам и репозиториям провайдеров, затрудняет унифицированный доступ для автоматизированных систем.

Сложность HTML

Исходный формат HTML с навигационными элементами, меню и несущественным контентом требует сложного синтаксического анализа для извлечения значимой документации.

Несоответствия версий

Различия между версиями провайдеров создают риск генерации неверных конфигураций, когда контекст версии теряется или не соответствует.

Без структурированной, чувствительной к версиям документации, AI agents рискуют создавать некорректные конфигурации Terraform, что может привести к сбоям инфраструктуры и уязвимостям безопасности.

Стратегия сбора данных

01

Идентификация источников

Официальные сайты документации провайдеров Terraform, реестры провайдеров с информацией о версиях, страницы ресурсов, источники данных и руководства по использованию.

02

Обнаружение провайдеров

Автоматический поиск и компиляция списков провайдеров с их доступными версиями из Terraform Registry API.

03

Web Scraping

Извлечение страниц документации для каждой пары "провайдер-версия", захватывающее полный HTML с метаданными.

04

Сохранение метаданных

Хранение необработанных HTML-документов с URL, версией провайдера и аннотациями типа страницы для отслеживаемости.

ОБРАБОТКА ДАННЫХ

Конвейер очистки и предварительной обработки

Очистка данных

- *Удаление навигационных и вспомогательных элементов*
- *Извлечение основного содержимого страницы*
- *Сохранение структуры заголовков и таблиц*
- *Сохранение вложенных блоков и примеров кода HCL*
- *Преобразование HTML в чистый текстовый формат*

Нормализация

- *Стандартизация форматирования текста*
- *Унификация названий и структуры разделов*
- *Восстановление отсутствующих заголовков разделов*
- *Проверка целостности описаний аргументов*

Разметка данных

Извлечение логических разделов и обогащение метаданных:

- Arguments

  Входные параметры и опции конфигурации

- Attributes

  Выходные значения и вычисляемые свойства

- Import

  Инструкции по импорту ресурсов

- Examples

  Примеры кода HCL и шаблоны использования

Архитектура хранения

Файловое хранилище

Необработанные HTML и очищенные TXT файлы, организованные по провайдеру и версии

MongoDB

Хранение метаданных с настройками провайдера и отслеживанием версий

Qdrant

Векторная база данных для индексированных текстовых фрагментов и семантического поиска

Многоуровневый подход к хранению данных поддерживает взаимосвязи между необработанными документами, обработанным контентом и векторными встраиваниями, обеспечивая эффективный поиск для запросов AI-агентов.

Архитектура платформы

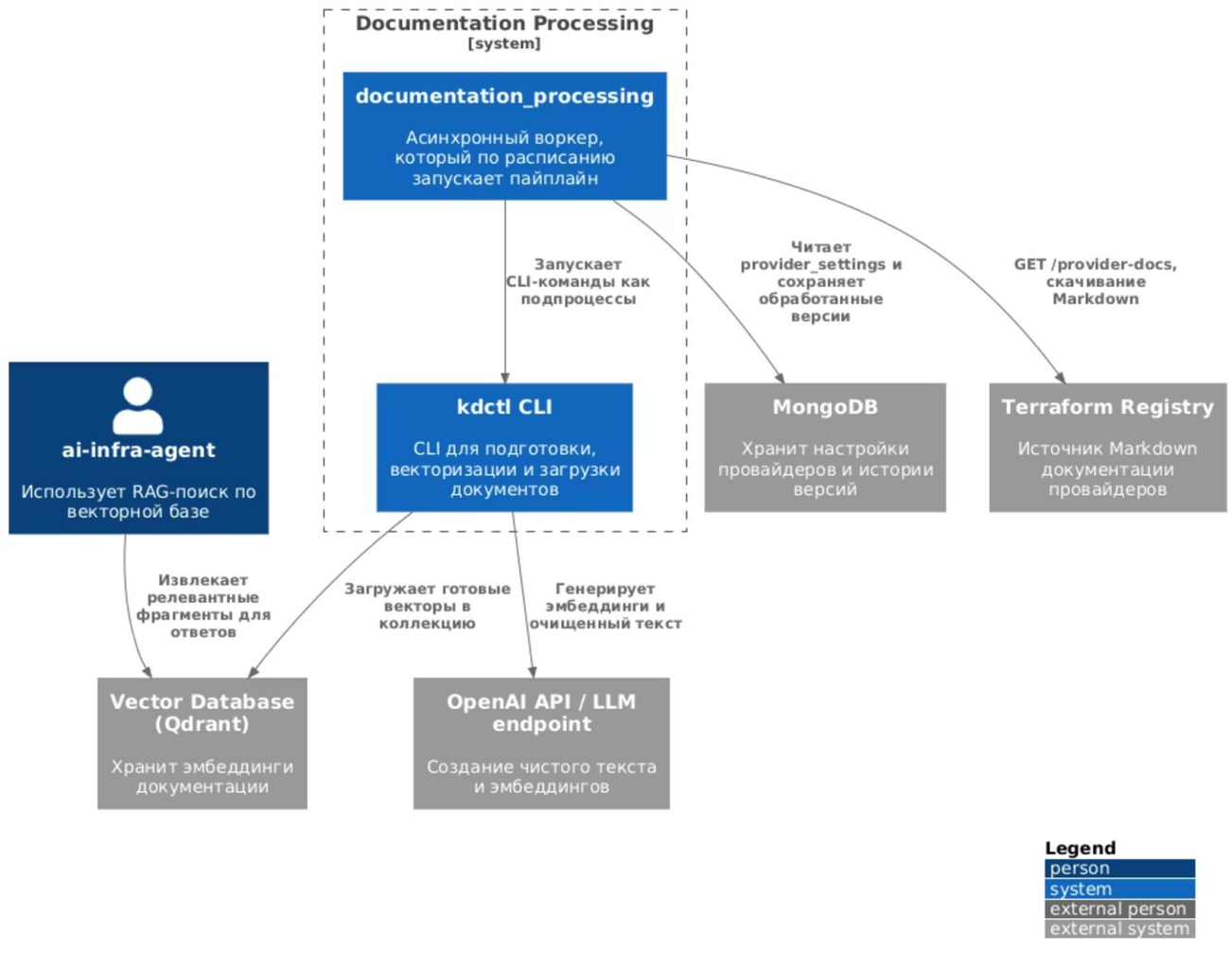
Точка входа

`src/documentation_processing/main.py` *создает контейнер зависимостей и запускает* `Application.run()`

Выполнение воркера

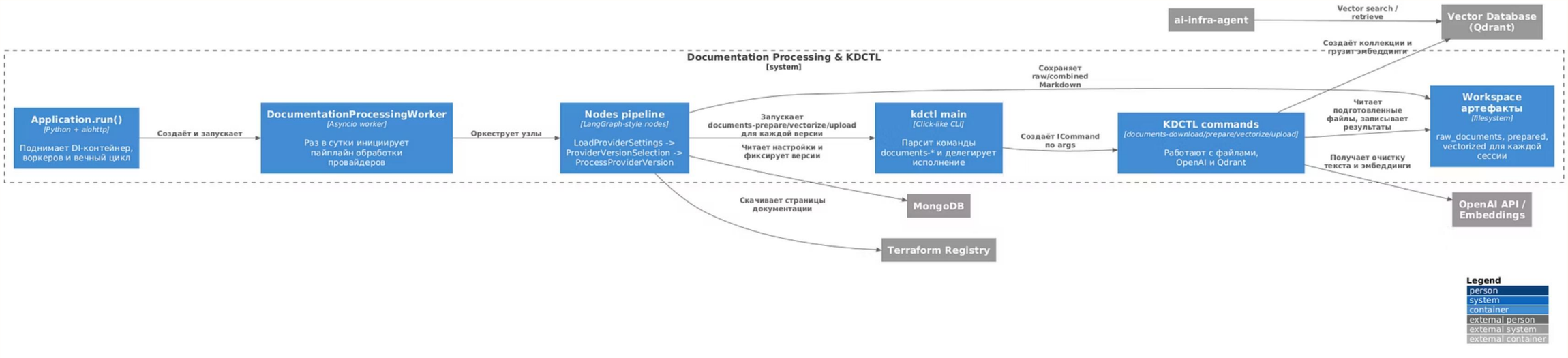
`DocumentationProcessingWorker` *выполняет конвейер, затем спит 24 часа перед следующим циклом*

Контекст системы (C4 диаграмма)



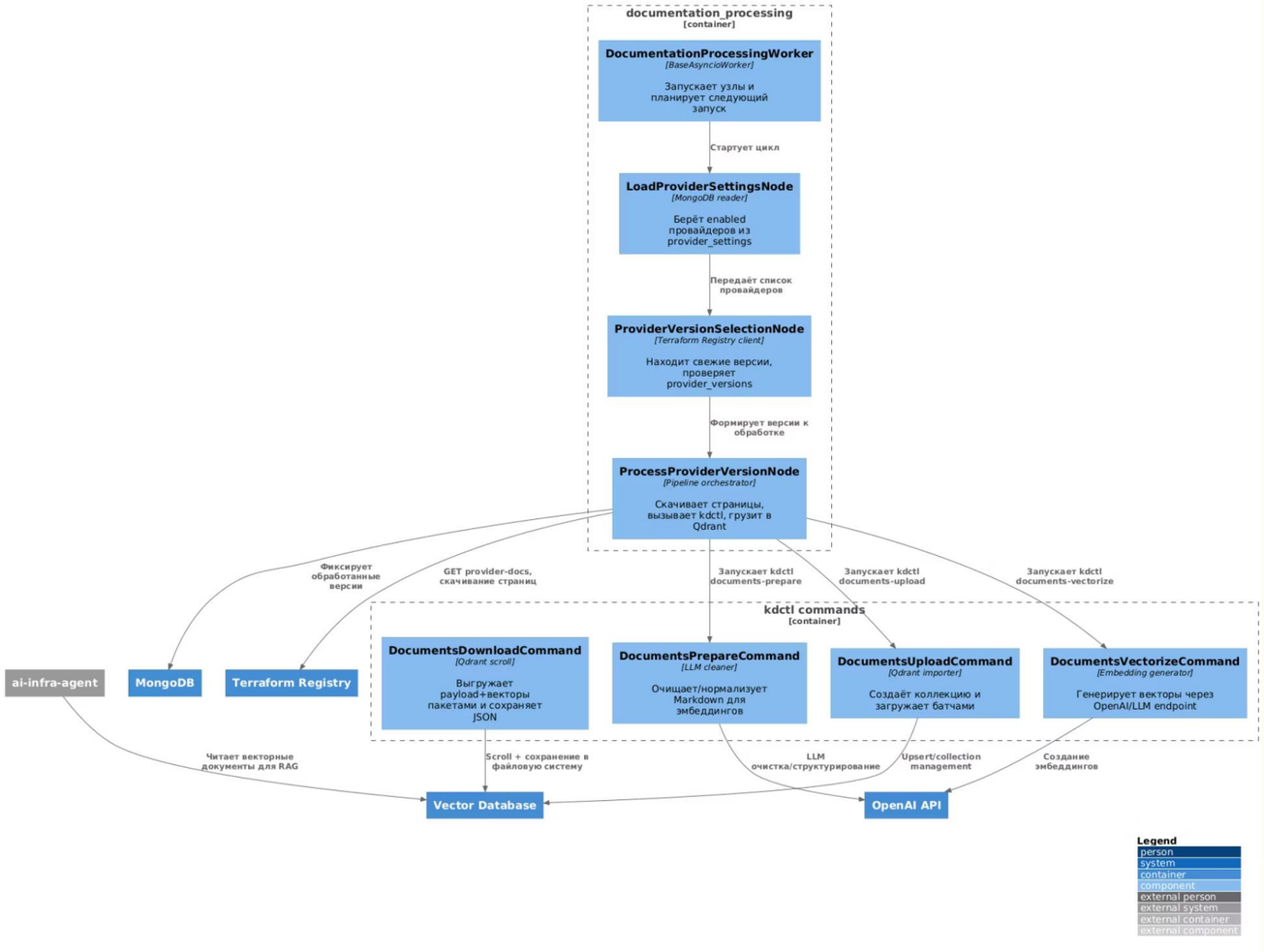
На этой диаграмме представлен общий обзор "Платформы документации провайдеров Terraform", ее основных пользователей и внешних систем, с которыми она взаимодействует.

Диаграмма контейнеров



На этой диаграмме представлено высокоуровневое взаимодействие между ключевыми компонентами платформы, включая внутренние службы и внешние системы, обеспечивающие полный цикл обработки документации.

Диаграмма компонентов



Эта диаграмма детализирует внутренние компоненты платформы и их взаимодействия, демонстрируя, как обрабатывается документация от сбора до векторизации и загрузки.

Этапы конвейера обработки

1

Загрузка настроек провайдера

Чтение коллекции `provider_settings` из MongoDB, фильтрация активных провайдеров, формирование списка обработки

2

Выбор версии

Запрос Terraform Registry API для получения последних версий, пропуск уже обработанных версий в `provider_versions`

3

Загрузка документов

Получение `page IDs` и контента из Registry, сохранение в `raw_documents`, объединение в унифицированный Markdown

4

Подготовка и векторизация

Выполнение `kdctl documents-prepare` для очистки, затем `kdctl documents-vectorize` для создания `embeddings`

5

Загрузка и запись

Отправка `vectors` в Qdrant через `kdctl documents-upload`, запись успешного выполнения в MongoDB

Конфигурация и зависимости

Переменные окружения (DPB_*)

MongoDB	*DPB_DB_MONGO__* для настроек и истории версий*
Qdrant	*DPB_DB_QDRANT__* для подключения к векторному хранилищу*
OpenAI	*Ключ API, название модели, необязательный базовый URL*
Общие	*Имя коллекции, настройки уровня логирования*

Основные зависимости

- Beanie/MongoDB

  Настройки провайдера и отслеживание обработанных версий

- aiohttp

  Асинхронный HTTP-клиент для вызовов Registry API

- kdctl CLI

  Подготовка документов, векторизация и загрузка