ПРОГРАММА ДЛЯ ЭВМ «СИНОД»

(вопросно-ответная система на основе базы знаний)

Инструкция по установке экземпляра программного обеспечения

предоставленного для проведения экспертной проверки

1. Назначение документа

Настоящий документ является инструкцией по установке экземпляра программы для электронных вычислительных машин «Синод» (далее — Программа), предоставленного для проведения экспертной проверки в целях включения Программы в реестр российского программного обеспечения.

Документ содержит описание требований к аппаратному и программному обеспечению, порядок получения дистрибутива, пошаговую инструкцию по установке, настройке и запуску Программы на сервере, а также порядок проверки работоспособности.

Документ предназначен для технических специалистов, осуществляющих развёртывание и настройку Программы.

2. Состав поставки (дистрибутив)

Экземпляр Программы для экспертной проверки поставляется в виде корневых каталогов с исходным кодом и сопутствующих конфигурационных файлов. Состав поставки включает следующие компоненты:

Таблица 2.1 — Состав поставляемого экземпляра Программы

Компонент	Тип	Описание
sinod-ai-service	Корневой каталог с исходным кодом	Сервис Программы, отвечающий за связь с RAG-моделью и принятие запросов к ней от пользователей (Spring Boot, Java 21)
sinod-api-gateway	Корневой каталог с исходным кодом	Сервис Программы, шлюз, принимающий все запросы от пользователей и маршрутизирующий их к соответствующим сервисам Программы (Spring Boot, Java 21)
sinod-auth-service	Корневой каталог с исходным кодом	Сервис авторизации Программы, обеспечивающий проверку прав доступа

Компонент	Тип	Описание
		и валидацию токенов в микросервисной архитектуре (Spring Boot, Java 21)
sinod-frontend	Корневой каталог с исходным кодом	Клиентская часть Программы (React, статические файлы)
sinod-knowledge-base	Корневой каталог с исходным кодом	Сервис эмбеддингов и взаимодействия с LLM Программы (Python, FastAPI, PyTorch)
sinod-llmbox	Корневой каталог с исходным кодом	Сервис доступа к языковым моделям Яндекс и ГосПромпт по API-токенам (Python, FastAPI)
postgres	Docker-образ (публичный)	СУБД с расширением векторного поиска pgvector
keycloak	Docker-образ (публичный)	Сервер аутентификации Keycloak.
redis	Docker-образ (публичный)	СУБД для хранения данных в оперативной памяти
consul	Docker-образ (публичный)	Инструмент для управления сетевой инфраструктурой и координации микросервисов
docker-compose.yaml	Конфигурационный файл	Файл оркестрации всех контейнеров Программы
.env.example	Конфигурационный файл	Шаблон переменных окружения
init-db.sql	Исполняемый SQL-скрипт	Скрипт для создания необходимых баз данных
deploy.sh	Исполняемый BASH-скрипт	Скрипт для автоматического запуска приложения

3. Требования к системе

3.1. Требования к аппаратному обеспечению

Таблица 3.1 — Минимальные требования к аппаратному обеспечению сервера

Параметр	Минимальное значение	Рекомендуемое значение
Архитектура процессора	x86-64 (AMD64)	x86-64 (AMD64)
Количество ядер процессора	4 ядра	8 ядер
Тактовая частота процессора	2,0 ГГц	3,0 ГГц
Оперативная память (RAM)	8 ГБ	16 ГБ
Дисковое пространство	50 ГБ свободно	100 ГБ свободно

Тип диска	SSD	SSD
Сетевой интерфейс	1 Гбит/с	1 Гбит/с
Графический ускоритель (GPU, опционально)	CUDA 11+, не менее 4 ГБ видеопамяти	CUDA 12+, не менее 8 ГБ видеопамяти

3.2. Требования к программному обеспечению

Таблица 3.2 — Требования к программному обеспечению сервера

Программное обеспечение	**Версия**	**Примечание**
Операционная система	Ubuntu 24.04 LTS и совместимые Linux-дистрибутивы	Debian 12+, CentOS Stream 9+, Astra Linux Special Edition, РЕД ОС также поддерживаются
Docker Engine	20.10 и выше	Обязательно
Docker Compose	v2.x (Compose V2)	Обязательно
Git	2.x	Для клонирования конфигурации
Браузер (для проверки)	Chrome 110+ / Firefox 110+	На рабочем месте проверяющего
Веб-сервер	nginx 1.28.0+	Для маршрутизации и проксирования запросов с внешней сети

Примечание. *Поддержка операционных систем Windows и macOS не гарантируется. Рекомендуется использовать Linux-сервер.*

3.3. Требования к сетевому окружению

Сервер должен иметь доступ к сети Интернет для загрузки Docker-образов (при первоначальной установке).

Доступ к серверу по SSH (порт 22) должен быть открыт для специалиста, выполняющего установку.

При эксплуатации с веб-сервером должны быть открыты порты 80 и 443 для подключений из внешней сети. Также необходимо иметь доменные имена и SSL-сертификаты для сервисов согласно таблице 3.3. Порты микросервисов будут открыты исключительно на локальный адрес (127.0.0.1), на который будет осуществляться проксирование с веб-сервера. Порты, используемые сервисами, представлены также в таблице 3.3

Таблица 3.3 — Доступ к сервисам при использовании веб-сервера

Сервис	Пример домена	Доступ с внешней сети	Используемые порты
sinod-frontend	domen.ru	Да	127.0.0.1:8888
sinod-api-gateway	gate.domen.ru	Да	127.0.0.1:8080
sinod-knowledge-base	know.domen.ru	Только с IP-адреса сервера	127.0.0.1:8012
sinod-llmbox	llmbox.domen.ru	Только с IP-адреса сервера	127.0.0.1:8001
sinod-auth-service	-	Нет	-
sinod-ai-service	-	Нет	-
keycloak	keycloak.domen.ru	Да	127.0.0.1:8889
postgres	-	Нет	127.0.0.1:5432
redis	-	Нет	-
consul	-	Нет	127.0.0.1:8500, 127.0.0.1:8600

Примечание. *Доступ к внешней сети регулируется в конфигурации вашего веб-сервера для конкретного сервиса*

3.4. Получение дистрибутива

3.4.1. Подготовка файла переменных окружения

Перед запуском Программы необходимо создать файл .env с переменными окружения. Состав переменных приведён в Таблице 3.4.

Таблица 3.4 — Переменные окружения

Переменная	Назначение	Пример значения
Общие		
PG_HOST	Адрес хоста СУБД PostgreSQL	postgres
PG_PORT	Порт СУБД PostgreSQL	5432
PG_USER	Имя пользователя БД	postgres
PG_PASSWORD	Пароль пользователя БД	<задаётся администратором>

Переменная	Назначение	Пример значения
CONSUL_HOST	Адрес хоста Consul	consul
CONSUL_PORT	Порт Consul	8050
sinod-knowledge-base		
KB_PG_DB	Имя базы данных sinod-knowledge base	know_db
RAG_TOP_K	Размер выдачи поиска по умолчанию	5
RAG_THRESHOLD	Минимальный порог релевантности	0.2
HYBRID_SEARCH	Включение гибридного поиска	true
FTS_HIT_MIN_COSINE	Минимальный cosine для статьи, найденной FTS	0.2
HYBRID_INCLUDE_FTS_ONLY	В hybrid-режиме подтягивает статьи, найденные только FTS	true
sinod-api-gateway		
REDIS_HOST	Адрес хоста Redis	redis
REDIS_PASSWORD	Пароль Redis	<задаётся администратором>
KEYCLOAK_CLIENT_ID	Идентификатор сервиса в Keycloak	api-gateway
KEYCLOAK_ISSUER_URI	Адрес реалма Sinod в Keycloak	https://keycloak.domen.ru/realms/sinod
sinod-ai-service		
AI_PG_DB	Имя базы данных sinod-ai-service	ai_service_db
LLM_RESPONSE_URL	Адрес sinod-llmbox	https://llmbox.domen.ru
LLM_TOKEN	Токен для доступа к sinod-llmbox	qwerty321
KB_URL	Адрес sinod-knowledge-base	https://know.domen.ru
sinod-frontend		
VITE_BASE_URL	Адрес sinod-api-gateway. Указывается в .env.production в корневом каталоге	https://gate.domen.ru

Переменная	Назначение	Пример значения
VITE_WS_URL	Адрес sinod-api-gateway для websockets. Указывается в .env.production в корневом каталоге	wss://gate.domen.ru
sinod-auth-service		
KEYCLOAK_URI	Базовый URL-адрес сервера Keycloak	https://keycloak.domen.ru
KEYCLOAK_APPLICATION_REALM	Название изолированной области (реалма) в Keycloak	sinod
AUTH_BFF_CLIENT_ID	Идентификатор бэкенд-прослойки (Backend-for-Frontend)	auth-bff
AUTH_BFF_CLIENT_SECRET	Секретный пароль бэкенд-прослойки	qwert4321
AUTH_EXCHANGER_CLIENT_ID	Идентификатор специального технического клиента, который обменивает одни токены на другие (механизм Token Exchange)	auth-exchanger
FRONTEND_REDIRECT_URI	URL-адрес для возврата на фронтенд	https://domen.ru/auth/callback
sinod-llmbox		
YANDEX_GPT_OPEN_AI_KEY	API-ключ для авторизации в сервисах YandexGPT через интерфейс, совместимый с OpenAI	qwertY2133
YANDEX_KEY_ID	Идентификатор открытого ключа доступа, используемый для подписи JWT-токенов при аутентификации	qwerty321
YANDEX_PRIVATE_KEY	Секретный приватный ключ для шифрования и подтверждения подлинности	"PLEASE DO NOT REMOVE THIS LINE! 11234123qrerqwe
YANDEX_SERVICE_ACCOUNT_ID	Уникальный идентификатор технического аккаунта в Yandex Cloud, от имени которого выполняются запросы	qwerqw123
postgres		
POSTGRES_USER	Пользователь-админ для инициализации PostgreSQL	<тот же, что и PG_USER>
POSTGRES_PASSWORD	Пароль пользователя-админа для инициализации PostgreSQL	<тот же, что и PG_PASSWORD>

Переменная	Назначение	Пример значения
POSTGRES_DB	База данных для инициализации PostgreSQL	postgres
keycloak		
KC_DB	Тип используемой базы данных	postgres
KC_DB_URL	Сетевой адрес, порт и имя базы данных для подключения Keycloak к серверу PostgreSQL.	jdbc:postgresql://postgres:5432/keycloak
KC_DB_USERNAME	Имя пользователя (логин) для авторизации в базе данных	<тот же, что и PG_USER>
KC_DB_PASSWORD	Пароль учетной записи базы данных	<тот же, что и PG_PASSWORD>
KEYCLOAK_ADMIN	Имя пользователя (логин) для главной учетной записи администратора самого Keycloak	kc_admin
KEYCLOAK_ADMIN_PASSWORD	Пароль учетной записи администратора Keycloak	<задаётся администратором>
KC_PROXY_HEADERS	Настраивает Keycloak на доверие нужным заголовкам, когда он работает за веб-сервером	xforwarded
KC_HOSTNAME	Основной домен Keycloak	keycloak.domen.ru
KC_FEATURES	Дополнительные функции Keycloak	admin-fine-grained-authz,token-exchange
KC_HTTP_ENABLED	Разрешает Keycloak принимать обычный незашифрованный HTTP-трафик	true

3.4.2. Запуск через docker-compose

В корневой директории выполнить команду

```
sudo chmod +x deploy.sh && ./deploy.sh
```

При первом запуске Программа выполнит следующие действия в автоматическом режиме:

1. Создание Docker-сети
2. Сборка образов всех микросервисов и скачивание образов сопутствующей инфраструктуры
3. Инициализация миграций sinod-ai-service
4. Инициализация sinod-knowledge-base – запуск миграций, скачивание модели FRIDA

5. Инициализация PostrgeSQL – создание базовой БД, выполнение скрипта на создание целевых БД, включение расширения pgvector
6. Запуск всех микросервисов и инфраструктуры

Проверка успешного запуска осуществляется посредством команды:

```
docker logs <наименование_контейнера>
```

В случае успешного запуска не должно быть никаких ошибок в консоли. Время первого запуска может составлять от 2 до 10 минут в зависимости от скорости загрузки весов модели.

После запуска микросервисов необходимо создать в Keycloak реалм и два клиента – для backend-for-frontend (bff) и exchanger и вписать их данные в соответствующие переменные окружения

3.5. Обновление Программы

Стандартная процедура обновления Программы:

1. Создать резервную копию базы данных.

2. Получить новую версию дистрибутива от правообладателя.

3. Заменить файлы исходного кода и/или обновить версию Docker-образа.

4. Запустить deploy.sh

5. Программа автоматически применит необходимые миграции схемы базы данных при запуске.

3.6. Решение типовых проблем

Таблица 3.2 — Типовые проблемы и способы их устранения

Проблема	Возможная причина	Способ устранения
Контейнер приложения не стартует, в логах ошибка подключения к БД	СУБД не успела инициализироваться	Подождать 30-60 секунд после старта compose; проверить логи СУБД
Ошибка загрузки модели FRIDA при первом старте	Нет сетевого доступа к HuggingFace	Использовать предварительно загруженный кэш модели или офлайн-зеркало
Поиск возвращает пустой результат при наличии данных	Не применены миграции эмбеддингов	Проверить статус миграций в таблице schema_migrations; перезапустить контейнер
Длительное время ответа на запросы поиска	Большой объём данных и отсутствие ускорения	Включить GPU при наличии; ограничить размер top_k; добавить HNSW-индекс на embedding

Проблема	Возможная причина	Способ устранения
Ошибка 'UNIQUE constraint violation' при сохранении статьи	Сработала защита от дубликатов фрагментов	Поведение штатное — повторные фрагменты пропускаются автоматически