ИННОТЕХ

Руководство администратора
Ассистент для команд разработки

г. Москва, 2025 г.

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ

Система «Ассистент для команд разработки» (далее - Система) автоматизирует генерацию программного кода, кода тестов ПО, написание сценариев тестирования и текста для технической и сопровождающей документации на ПО, используя ИИ для анализа введенной Пользователем информации (запросы на естественном языке, требования, блоки кода) и генерации качественных ответов от LLM (оптимизированного кода, тестов и текста на основе шаблонов). Система интегрируется в вашу среду разработки или может быть открыта в веб-браузере, что повышает скорость разработки программного кода, снижает количество ошибок при его написании и улучшает качество кода.

Основные функции и возможности:

1. Для аутентификации пользователей Система позволяет генерировать API-ключи.
2. Система поддерживает текстовый чат с возможностью отправки Пользователем запросов к LLM и получения ответов от LLM, поддержкой контекста и истории диалога.
3. Система обеспечивает возможность изменения и сохранения параметров генерации ответов LLM в текстовом чате (температура, длина ответа, глубина контекста).
4. Система позволяет генерировать программный код на основе текстового описания задачи. Использование данной функции возможно в веб-интерфейсе или интегрированной среде разработки (IDE):
   - Генерация кода по описанию задачи на естественном языке;
   - Оптимизация и рефакторинг существующего кода;
   - Генерация комментариев к коду.
5. Система автоматически генерирует unit-тесты для конкретных частей программного кода, что позволяет сократить время тестирования:
   - Генерация unit-тестов.
6. Система автоматизирует такие задачи, как документирование кода и генерация комментариев:
   - Генерация комментариев к коду;
   - Автоматическое код-ревью.
7. Система обеспечивает возможность загружать документацию в качестве дополнительных знаний для LLM в различных форматах (.txt, .java, .js, .ts, .html, .MD).

2. ТЕРМИНЫ И ОБОЗНАЧЕНИЯ

В тексте настоящего документа представлены следующие сокращения (см. **Таблица 1-1**).

***Таблица 1-1** — Перечень сокращений*

Сокращение/аббревиатура	Значение
ИИ	Искусственный интеллект
АКР	Ассистент для команд разработки
Система	Ассистент для команд разработки
IDE	Интегрированная среда разработки
LLM	Большая языковая модель

В тексте настоящего документа представлены следующие термины и определения (см. **Таблица 1-2**).

Таблица 1-2 — Перечень терминов и определений

Аутентификация	Проверка подлинности предъявленного Пользователем идентификатора
Администратор Системы	Пользователь Системы с ролью «Администратор»
Заказчик	Физическое или юридическое лицо, которое инициирует, заказывает и финансирует разработку, модификацию или поддержку программного обеспечения Системы. После приемки услуг по разработке Системы Заказчик является Владельцем Системы
Пользователь	Работник, наделенный правами доступа к информационным ресурсам организации
Подсказка	Системный промпт для LLM - представляет собой руководящие инструкции или исходные данные, которые задаются модели, чтобы она могла генерировать ответы на запросы пользователя
Роль	Набор полномочий, который необходим Пользователю для выполнения определённых рабочих задач. Каждый сотрудник может иметь одну или несколько ролей, а каждая роль может содержать от одного до множества полномочий, которые разрешены Пользователю в рамках этой роли. Роли Пользователей в Системе: – «Пользователь»; – «Администратор».
ИТ-инфраструктура	Совокупность всего программного обеспечения, оборудования, сетей и подключенных сервисов, образующих ИТ-среду организации
Файл	Цифровой носитель информации, содержащий текстовые данные, с любым из перечисленных расширений: .txt, .java, .js, .ts, .html, .MD.

3. ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ СИСТЕМЫ

3.1. Требования к инфраструктуре

Таблица 1. Требования к Kubernetes кластеру для Control Plane

	До 100 польз.	**До 500 польз.**	**До 1000 польз.**	**До 5000 польз.**
CPU	4 vCPU	8 vCPU	16 vCPU	32 vCPU
RAM	16 GB	32 GB	64 GB	128 GB
Storage	50 GB	100 GB	200 GB	1 TB

Таблица 2. Требования к Kubernetes кластеру для Worker Node

	До 100 польз.	**До 500 польз.**	**До 1000 польз.**	**До 5000 польз.**
CPU	4-8 vCPU	8-16 vCPU	16-32 vCPU	32-64 vCPU
RAM	32 GB	64 GB	128 GB	256 GB
Storage	50 GB	100 GB	200 GB	1 TB
Дополнительно*	1 GPU	1-2 GPU	2-4 GPU	4-8 GPU

3.2. Требования к установленному программному обеспечению

Таблица 3. Требования к внешнему программному обеспечению

PostgreSQL [Version]				
	CPU	**RAM**	**Storage**	**Прочие требования**
S (до 100 пользователей)	2 vCPU	4 GB	50 GB	– Стандартная конфигурация – Автоматическое резервное копирование данных
M (до 500 пользователей)	4 vCPU	8 GB	100 GB	– Стандартная конфигурация – Автоматическое резервное копирование данных
L (до 1000 пользователей)	6 vCPU	16 GB	200 GB	– Автоматическое резервное копирование данных – Настройка кэширования
XL (до 5000 пользователей)	8 vCPU	32 GB	500 GB	– Репликация – Настройка кэширования
2XL (до 10000 пользователей)	12 vCPU	64 GB	1 TB	– Репликация – Настройка кэширования – Настройка индексации
MinIO [Version]				
	CPU	**RAM**	**Storage**	**Прочие требования**

Без требования к пользователям	-	4 GB	10 GB	– S3-совместимое хранилище
Nginx [Version]				
	CPU	**RAM**	**Storage**	**Прочие требования**
S (до 100 пользователей)	1 vCPU	2 GB	20 GB	– Стандартная конфигурация
M (до 500 пользователей)	2 vCPU	4 GB	40 GB	– Оптимизация производительности
L (до 1000 пользователей)	4 vCPU	8 GB	80 GB	– Настройка балансировки нагрузки
XL (до 5000 пользователей)	6 vCPU	16 GB	160 GB	– Настройка кэширования – Оптимизация SSL
2XL (до 10000 пользователей)	8 vCPU	32 GB	320 GB	– Настройка многоуровневого кэширования – Конфигурирование для обеспечения отказоустойчивости
OpenSearch [Version]				
	CPU	**RAM**	**Storage**	**Прочие требования**
S (до 100 пользователей)	2 vCPU	4 GB	50 GB	– Стандартная конфигурация
M (до 500 пользователей)	4 vCPU	8 GB	100 GB	– Настройка индексации
L (до 1000 пользователей)	8 vCPU	16 GB	200 GB	– Настройка кластеризации
XL (до 5000 пользователей)	16 vCPU	32 GB	500 GB	– Репликация – Конфигурирование для обеспечения отказоустойчивости
2XL (до 10000 пользователей)	24 vCPU	64 GB	1 TB	– Гео-репликация – Настройка безопасности и производительности

4. ПЕРВОНАЧАЛЬНАЯ УСТАНОВКА СИСТЕМЫ

4.1. Состав компонентов системы

Состав, версии и описание сервисов системы представлены в таблице ниже.

Таблица 4. Перечень сервисов поставки системы

№	Название модуля	Наименование компонента	Имя сервиса	Версия	Описание
1	Модуль веб-интерфейса	Веб-интерфейс	akr-web-interface	2025.1-01	Модуль веб-интерфейса с аутентификацией с генерацией API-ключей и веб-чатом для ответов на вопросы по разработке ПО обеспечивает аутентификацию и управление доступом пользователей к Системе и предоставляет интерфейс для взаимодействия (текстовых диалогов) пользователей с LLM. Отвечает за обработку запросов, генерацию ответов, выполнение команд и поддержку диалогов
2	Модуль управления запросами к LLM	Модуль управления запросами к LLM	akr-backend-inference	2025.1-01	Модуль управления запросами к LLM предоставляет пользователям возможность изменять параметры генеративной модели (температуру, креативность, длину ответов), сохранять индивидуальные настройки и выбирать предустановленные конфигурации, создавать,

					редактировать и сохранять подсказки (промпты) для улучшения ответов модели
3	Плагин для IDE VS Code	Плагин для IDE VS Code	akr-ide-vs-code	2025.1-01	Плагин для IDE VS Code предоставляет интерфейс для взаимодействия (текстовых диалогов) пользователей с LLM в VS Code, автодополнение кода проекта, открытого в VS Code, генерацию комментариев к программному коду и рекомендаций по улучшению кода в текстовом диалоге в VS Code, а также возможность вставки отдельных блоков кода из текстового чата в проект, открытый в IDE.
4	Плагин для IDE IntelliJ IDEA	Плагин для IDE IntelliJ IDEA	akr-ide-jetbrains	2025.1-01	Плагин для IDE IntelliJ IDEA предоставляет интерфейс для взаимодействия (текстовых диалогов) пользователей с LLM в IDE IntelliJ IDEA, автодополнение кода проекта, открытого в IDE IntelliJ IDEA, генерацию комментариев к программному коду и рекомендаций по улучшению кода в

					текстовом диалоге в IDE IntelliJ IDEA, а также возможность вставки отдельных блоков кода из текстового чата в проект, открытый в IDE.

4.2. План ввода системы в эксплуатацию

Корректная последовательность запуска сервисов критически важна, поскольку первый сервис поддерживают работу всех компонентов, связанных с inference.

- akr-backend-inference – производит базовую инициализацию системы, подготавливает все модули к корректной и стабильной работе, обеспечивает исполнение всех фоновых задач системы,
- akr-web-interface – обеспечивает развертывание веб-интерфейса.
- akr-ide-vs-code – обеспечивает сборку плагина для VS Code.
- akr-ide-jetbrains – обеспечивает сборку плагина для Jetbrains.

5. ПЕРВОНАЧАЛЬНАЯ УСТАНОВКА СИСТЕМЫ

Все необходимые настройки описаны в Главе 4, дополнительная конфигурация не требуется.

5.1. Первоначальная настройка

После успешного запуска сервисов дополнительная настройка со стороны системного администратора не требуется.

5.2. Интеграция с внешними системами

После успешного запуска сервисов дополнительная настройка со стороны системного администратора не требуется.

5.3. Журналирование

Логирование сервисов осуществляется автоматически, и дополнительная настройка не требуется.

6. ПЕРВОНАЧАЛЬНАЯ УСТАНОВКА СИСТЕМЫ

Описание приведено в документе «Руководство пользователя».

7. РЕКОМЕНДАЦИИ ПО ОБЕСПЕЧЕНИЮ НАДЕЖНОСТИ И ВОССТАНОВЛЕНИЮ РАБОТЫ СИСТЕМЫ

7.1. Диагностика системы

Для сервисов с REST API предусмотрены следующие механизмы проверки работоспособности:

- Healtz-check: эндпоинт /healthz для проверки общего состояния сервиса;
- Readiness probe: эндпоинт /readyz для проверки готовности сервиса к обработке запросов;
- Эндпоинты доступны через Swagger или могут быть вызваны с помощью HTTP-запросов.

7.2. Резервное копирование

Резервное копирование системы рекомендуется производить по следующему расписанию:

- Полный бэкап: осуществляется в выходные дни 1 раз в неделю;
- Инкрементальный бэкап: выполняется ежедневно, один раз в сутки;
- Архивирование логов: производится несколько раз в течение дня.

Для повышения отказоустойчивости необходимо выполнять резервное копирование следующих компонентов:

- База данных Postgres: содержит конфигурации для обучения и инференса моделей, логи обучения и прочие данные, необходимые для работы всех сервисов;
- S3 хранилище: содержит артефакты обучения, а также обученные модели.

7.3. Восстановление из резервной копии

Особых указаний по процедуре восстановления из резервной копии не предусмотрено.