

АНО «ГОРЬКИЙ ТЕХ»

**ЭКСПЛУАТАЦИОННАЯ ДОКУМЕНТАЦИЯ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

«Корпоративный электронный помощник (КЭП)»

Версия 1.0

ГОСТ 19.505-79
ГОСТ Р 59795-2021

Нижний Новгород
2025

Оглавление

ЧАСТЬ 1. ИНСТРУКЦИЯ ПО ЭКСПЛУАТАЦИИ.....	4
1.1. Назначение программы.....	4
1.2. Условия выполнения программы.....	4
1.3. Выполнение программы.....	5
1.4. Команды управления и ответы программы.....	7
1.5. Сообщения оператору.....	9
ЧАСТЬ 2. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ.....	11
2.1. Введение.....	11
2.2. Подготовка к работе.....	11
2.3. Работа с Telegram-ботом.....	12
2.4. Работа с ботом МАХ.....	13
2.5. Работа с веб-виджетом.....	13
2.6. Типовые сценарии использования.....	13
2.7. Рекомендации по формулировке запросов.....	14
2.8. Возможные сообщения и реакция пользователя.....	14
ЧАСТЬ 3. РУКОВОДСТВО АДМИНИСТРАТОРА.....	16
3.1. Общие сведения.....	16
3.2. Развёртывание системы.....	17
3.3. Управление системой.....	18
3.4. Управление базой знаний.....	19
3.5. Мониторинг и наблюдаемость.....	19
3.6. Резервное копирование и восстановление.....	20
3.7. Безопасность.....	21
3.8. Диагностика и устранение неисправностей.....	22

ЧАСТЬ 1. ИНСТРУКЦИЯ ПО ЭКСПЛУАТАЦИИ

Настоящая часть подготовлена в соответствии с ГОСТ 19.505-79 «Руководство оператора. Требования к содержанию и оформлению» и описывает последовательность действий, обеспечивающих загрузку, запуск, выполнение и завершение программы, состав функций, формат и возможные варианты команд, а также ответы программы на эти команды.

1.1. Назначение программы

Программа «Корпоративный электронный помощник (КЭП)» предназначена для автоматизированной обработки текстовых запросов пользователей и предоставления информационно-справочных ответов на основе произвольного набора данных (датасета). Программа реализована на архитектуре RAG, объединяющей семантический поиск по векторной базе данных и генерацию ответов с помощью большой языковой модели.

Эксплуатационные режимы:

- круглосуточный режим работы (24/7) с регламентными технологическими паузами;
- многопользовательский режим — до 1000 одновременных сессий;
- сервисный (административный) режим — выполнение операций администрирования.

1.2. Условия выполнения программы

Минимальные аппаратные требования сервера:

- процессор — 4 ядра x86-64;
- оперативная память — 8 ГБ;
- дисковое пространство — 100 ГБ SSD;
- сетевой интерфейс — 100 Мбит/с.

Рекомендуемая конфигурация сервера:

- процессор — 8 ядер x86-64;
- оперативная память — 16 ГБ;
- дисковое пространство — 500 ГБ NVMe SSD;
- сетевой интерфейс — 1 Гбит/с.

Программное обеспечение сервера:

- операционная система: Ubuntu Server 22.04 LTS (рекомендуется), Debian 11+, CentOS Stream 9 или Windows 11;
- Docker Engine 24.0 и выше, Docker Compose 2.20 и выше;
- текстовый редактор для правки конфигурационных файлов.

Сетевые требования:

- пропускная способность канала Интернет — не менее 10 Мбит/с;
- входящие порты на сервере: 80/TCP, 443/TCP, 22/TCP;
- сервисные порты: 5000/TCP (веб), 8080/TCP (API), 3000/TCP (Grafana), 9443/TCP (Portainer);
- разрешённый исходящий HTTPS до api.telegram.org и api.max.ru.

1.3. Выполнение программы

Загрузка программы.

- 1) Получить дистрибутив (по ссылке у разработчика или клонированием Git):

```
git clone -b CPU_Build https://github.com/Skailin24/Gorky KEP.git  
/opt/kep
```

- 2) Перейти в каталог:

```
cd /opt/kep
```

- 3) Подготовить файл конфигурации:

```
cp _env.example .env  
nano .env
```

Заполнить обязательные параметры:

```
TELEGRAM_BOT_TOKEN=<токен>  
MAX_BOT_TOKEN=<токен>
```

- 4) Разместить датасет в каталоге data/:

```
mkdir -p data
cp government_support_measures.xlsx data/
```

5) Выполнить сборку Docker-образов:

```
docker compose build
```

Запуск программы.

Команда штатного запуска:

```
docker compose up -d
```

Время запуска — 30–160 секунд (первый запуск с загрузкой модели Ollama занимает 5–10 минут). Альтернативные варианты — `make up`, `make dev`, `make monitoring`.

Проверка успешного запуска:

```
docker compose ps
```

Доступные сервисные адреса после запуска:

Сервис	Адрес
Веб-виджет (демо)	<code>http://<server_ip>:5000/demo</code>
Backend REST API	<code>http://<server_ip>:8080</code>
Swagger-документация API	<code>http://<server_ip>:8080/docs</code>
Grafana — мониторинг	<code>http://<server_ip>:3000</code>
Portainer — управление	<code>https://<server_ip>:9443</code>

Выполнение программы.

В режиме штатного выполнения программа автоматически принимает запросы пользователей через сконфигурированные каналы и формирует ответы.

Алгоритм обработки одного запроса:

- 1) Пользователь отправляет сообщение в Telegram, MAX или веб-виджет.
- 2) Бот-интерфейс направляет запрос на Backend через `POST /query`.
- 3) Backend выполняет семантический поиск в ChromaDB.
- 4) Найденные документы и запрос передаются в LLM (Ollama, Gemma3:12b) для генерации ответа.
- 5) Ответ возвращается в исходный канал и отображается пользователю.
- 6) Запрос и ответ фиксируются в журнале обращений.

Завершение работы программы.

Штатное завершение:

```
docker compose down
```

Контейнеры останавливаются, ресурсы освобождаются, данные сохраняются в томах. Время — до 60 секунд.

Принудительное завершение (только в нештатной ситуации):

```
docker compose kill
```

1.4. Команды управления и ответы программы

Команды Telegram-бота и бота МАХ.

Команда / кнопка	Назначение	Формат / условия	Ответ программы
/start	Инициация сессии, отображение главного меню	Текстовая команда; ввод в начале работы или для возврата в меню	Приветственное сообщение и главное меню (3 кнопки)
 Подбор мер поддержки	Переход в режим свободного запроса	Кнопка главного меню	Приглашение сформулировать запрос
 Частые вопросы	Просмотр часто задаваемых вопросов	Кнопка главного меню	Список часто задаваемых вопросов
 Справка	Краткое описание возможностей бота	Кнопка главного меню	Информационное сообщение
Свободный текстовый ввод	Запрос к базе знаний	Произвольная строка до 4000 символов	Сгенерированный ответ (5–10 секунд)
Кнопка очистки диалога	Сброс контекста сессии	Экранная кнопка под сообщением	Подтверждение очистки контекста

Элементы управления веб-виджета.

Элемент интерфейса	Назначение	Ответ программы
--------------------	------------	-----------------

Поле ввода сообщения	Ввод текстового запроса	Запрос отображается в области переписки
Кнопка «Отправить»	Отправка запроса на сервер	Ответ выводится в области переписки за 5–10 секунд
Кнопки главного меню	Подбор мер поддержки / Частые вопросы / Справка	Аналогично соответствующим кнопкам Telegram-бота
Кнопка очистки диалога	Сброс контекста сессии	Контекст сбрасывается; область переписки очищается

Команды управления серверной частью.

Команда	Назначение	Ответ программы
docker compose up -d	Запуск всех компонентов	Контейнеры запущены, exit code 0
docker compose down	Штатная остановка	Контейнеры остановлены и удалены
docker compose ps	Статус контейнеров	Таблица состояний контейнеров
docker compose logs -f	Журналы в реальном времени	Поток сообщений всех сервисов
docker compose restart <service>	Перезапуск сервиса	Указанный сервис перезапущен
docker stats	Потребление ресурсов	Таблица CPU/MEM/I-O по контейнерам
make build	Пересборка образов	Образы пересобраны (5–15 мин)
make health	Проверка работоспособности API	JSON со статусом компонентов; HTTP 200/503
make logs	Журналы всех сервисов	Полный журнал
make logs-backend / logs-bots	Журналы конкретных сервисов	Журнал backend / ботов

make clean	Удаление контейнеров	Контейнеры удалены, тома сохранены
------------	----------------------	------------------------------------

Эндпоинты REST API.

Полная документация — Swagger UI по адресу http://<server_ip>:8080/docs.

Метод и путь	Назначение	Ответ программы
POST /query	Обработка запроса пользователя	JSON: answer, sources, session_id; HTTP 200
GET /history	История обращений сессии	JSON-массив записей переписки; HTTP 200
GET /measures	Поиск записей по датасету	JSON-массив записей; HTTP 200
POST /health	Состояние компонентов	JSON со статусами; HTTP 200 (норма) / 503 (сбой)

1.5. Сообщения оператору

Сообщения пользовательского интерфейса.

Тип сообщения	Условие появления	Действие пользователя
Приветствие и главное меню	После /start или возврата в меню	Выбрать раздел кнопкой или ввести запрос
Сгенерированный ответ	На запрос пользователя	Изучить ответ; задать уточняющий вопрос
Уточняющий вопрос системы	Запрос недостаточно полон	Уточнить запрос
«Ничего не найдено»	Нет релевантных записей	Переформулировать запрос
«Сообщение слишком длинное»	Длина запроса > 4000 символов	Сократить и повторить
«Произошла техническая ошибка»	Временный сбой компонента	Повторить через 1–2 минуты

Системные коды возврата.

Код / сообщение	Значение	Действие администратора
Exit code 0	Команда выполнена успешно	Не требуются
Exit code != 0	Ошибка выполнения	docker compose logs
HTTP 200	Запрос к API успешен	Не требуются
HTTP 400	Неверный формат запроса	Сверить со схемой Swagger
HTTP 404	Эндпоинт не существует	Проверить путь по Swagger
HTTP 500	Внутренняя ошибка backend	make logs-backend
HTTP 503	Компонент недоступен	docker compose ps; перезапустить

ЧАСТЬ 2. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Настоящая часть адресована конечным пользователям системы — гражданам и сотрудникам организаций, обращающимся за справочной информацией. Подготовлена по ГОСТ Р 59795-2021.

2.1. Введение

«КЭП» — это интеллектуальный справочный сервис, позволяющий получать актуальную информацию из специализированной базы знаний в диалоговом режиме на естественном русском языке.

Возможности для пользователя:

- получение информации о мерах поддержки на основе загруженного датасета;
- пошаговое сопровождение по получению меры поддержки;
- просмотр часто задаваемых вопросов и готовых ответов на них;
- уточнение неполных или неоднозначных запросов в диалоге;
- получение ответа в любое время суток (24/7).

Работа с системой не требует специальной подготовки — достаточно навыков использования мессенджеров или веб-браузера.

2.2. Подготовка к работе

Система предоставляет три равнозначных канала обращения; пользователь выбирает удобный самостоятельно:

Канал	Способ доступа
Telegram-бот	Поиск бота по имени в Telegram
Бот мессенджера МАХ	Поиск бота по имени в МАХ
Веб-виджет	Переход по ссылке на сайте организации

Минимальные требования к рабочему месту:

- клиент Telegram (мобильный/десктоп) актуальной версии;




- клиент МАХ актуальной версии;
- веб-браузер с поддержкой HTML5 и JavaScript;
- доступ в сеть Интернет.

2.3. Работа с Telegram-ботом

Поиск и запуск бота:


- 1) Откройте Telegram.
- 2) В строке поиска введите имя бота, предоставленное организацией.
- 3) Откройте чат и нажмите «Начать» (или введите /start вручную).
- 4) В ответ бот пришлёт приветствие и главное меню.

Главное меню содержит три кнопки:


-  Подбор мер поддержки — режим свободного запроса;
-  Частые вопросы — список наиболее распространённых вопросов;
-  Справка — описание возможностей бота.

Для возврата в меню в любой момент введите /start. Для сброса контекста сессии (смена темы) нажмите кнопку очистки диалога.


Подбор мер поддержки.

1) Нажмите « Подбор мер поддержки». 2) Введите вопрос обычным текстом, например: «Какие меры поддержки доступны самозанятым?». 3) В течение 5–10 секунд бот пришлёт ответ. 4) При необходимости задайте уточняющий вопрос — система учитывает контекст диалога.

Частые вопросы.

Нажмите « Частые вопросы», выберите интересующий пункт нажатием соответствующей кнопки — будет выведен подготовленный ответ.

Справка.

Нажмите « Справка» в главном меню — бот выведет краткое описание назначения и порядка работы.

2.4. Работа с ботом МАХ

Бот в МАХ функционально полностью идентичен Telegram-боту. Откройте приложение МАХ, найдите бота по имени, нажмите «Начать» или введите /start. Используйте главное меню и свободный ввод так же, как в Telegram.

2.5. Работа с веб-виджетом


Веб-виджет открывается переходом по ссылке, размещённой на ресурсе организации, либо по адресу `http://<server_ip>:5000/demo`. Элементы интерфейса:

- область переписки — отображение истории диалога;
- поле ввода сообщения — ввод текстового запроса;
- кнопка «Отправить» — отправка запроса на сервер;
- кнопки главного меню — те же три раздела, что и в боте;
- кнопка очистки диалога — сброс контекста сессии.


Порядок работы: введите вопрос в поле ввода и нажмите «Отправить» (или клавишу Enter). Ответ появится в области переписки за 5–10 секунд.

2.6. Типовые сценарии использования

Сценарий 1. Получение справки по конкретной мере поддержки.

Запустите бота /start → « Подбор мер поддержки» → введите конкретный вопрос («Как оформить субсидию на жильё молодой семье?») → получите ответ → при необходимости уточните.

Сценарий 2. Просмотр частых вопросов.

/start → « Частые вопросы» → выберите вопрос из списка → получите готовый ответ.

Сценарий 3. Уточнение запроса.

Введите общий вопрос («Какие льготы есть для пенсионеров?») → получите обзорный ответ → задайте уточняющий вопрос («А по транспортному налогу?») — система использует контекст и сформирует точный ответ.

Сценарий 4. Смена темы.

Нажмите кнопку очистки диалога → контекст сбросится → введите новый вопрос — он будет обработан независимо от предыдущей переписки.

2.7. Рекомендации по формулировке запросов

— формулируйте вопрос полным предложением, указывая объект и интересующее действие;

— избегайте односложных запросов («субсидия», «налог») — они слишком общие;

— указывайте контекст, важный для подбора ответа: категорию заявителя, регион, цель;

— при общем ответе задайте уточняющий вопрос — система учтёт контекст;

— если по теме нет ответа, попробуйте переформулировать с использованием синонимов.

2.8. Возможные сообщения и реакция пользователя

Сообщение системы	Условие	Действия пользователя
Приветствие и главное меню	После /start или возврата в меню	Выбрать раздел или ввести запрос
Сгенерированный ответ	После отправки запроса	Изучить ответ; задать уточняющий вопрос
«Ничего не найдено»	В базе нет релевантных записей	Переформулировать запрос
«Уточните, пожалуйста, что именно вас интересует»	Запрос слишком общий	Дополнить запрос конкретикой

«Сообщение слишком длинное»	Запрос > 4000 символов	Сократить и повторить
«Произошла техническая ошибка»	Временный сбой	Подождать 1–2 минуты и повторить; при повторении — обратиться в поддержку

ЧАСТЬ 3. РУКОВОДСТВО АДМИНИСТРАТОРА

Настоящая часть адресована администраторам, выполняющим установку, настройку, сопровождение и техническое обслуживание системы. Подготовлена по ГОСТ Р 59795-2021.

3.1. Общие сведения

Функциональные обязанности администратора:

- первоначальное развёртывание ПО на серверном оборудовании;
- конфигурирование переменных окружения и параметров запуска;
- управление содержимым базы знаний;
- ежедневный контроль работоспособности компонентов;
- выполнение резервного копирования и восстановления;
- обновление системы до новых версий;
- первичная диагностика и устранение неисправностей.

Требования к квалификации:

- опыт администрирования Linux-систем;
- уверенное владение командной оболочкой bash;
- знание Docker и Docker Compose;
- понимание основ TCP/IP и firewall;
- базовые навыки работы с Git.

Состав компонентов системы:

Компонент	Технология	Назначение
backend	Python 3.12 + FastAPI	REST API, бизнес-логика
bot-telegram	python-telegram-bot	Канал Telegram
bot-max	Python + REST	Канал MAX
web-frontend	React + Nginx	Веб-виджет
vectordb	ChromaDB	Векторная база (семантический поиск)
llm	Ollama + Gemma3:12b	Локальная языковая модель

redis	Redis 7	Кеширование
prometheus / grafana	Prometheus / Grafana	Сбор метрик и визуализация
portainer	Portainer	Веб-интерфейс управления контейнерами

3.2. Развёртывание системы

1) Подготовка окружения сервера:

```
sudo apt update && sudo apt upgrade -y
sudo apt install -y git curl make ca-certificates
sudo ufw allow 22/tcp; sudo ufw allow 80/tcp; sudo ufw allow 443/tcp
sudo ufw enable
```

2) Установка Docker и Docker Compose (по официальной инструкции):

```
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg \
-o /etc/apt/keyrings/docker.asc
echo "deb [arch=$(dpkg --print-architecture) \
signed-by=/etc/apt/keyrings/docker.asc] \
https://download.docker.com/linux/ubuntu \
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list
sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io \
docker-compose-plugin
sudo usermod -aG docker $USER && newgrp docker
```

3) Получение дистрибутива:

```
git clone -b CPU_Build \
https://github.com/Skailin24/Gorky KEP.git /opt/kep
```

4) Настройка переменных окружения:

```
cd /opt/kep && cp _env.example .env && nano .env
```

Обязательные переменные:

Переменная	Назначение
TELEGRAM_BOT_TOKEN	Токен Telegram-бота
MAX_BOT_TOKEN	Токен бота MAX
BACKEND_PORT	Порт API (по умолчанию 8080)
WEB_PORT	Порт веб-виджета (по умолчанию 5000)
GRAFANA_PORT	Порт Grafana (по умолчанию 3000)
PORTAINER_PORT	Порт Portainer (по умолчанию 9443)

LLM_MODEL	Модель Ollama (по умолчанию gemma3:12b)
VECTOR_DB_PATH	Каталог ChromaDB (по умолчанию ./data/chroma)

5) Размещение датасета и запуск:

```
mkdir -p data && cp government_support_measures.xlsx data/
make build && make up
```

Первый запуск требует загрузки модели Ollama (~8 ГБ), время ожидания — 5–10 минут.

6) Настройка автозапуска (systemd):

```
sudo nano /etc/systemd/system/kep.service
```

Содержимое:

```
[Unit]
Description=KEP Docker Compose stack
Requires=docker.service
After=docker.service

[Service]
Type=oneshot
RemainAfterExit=yes
WorkingDirectory=/opt/kep
ExecStart=/usr/bin/docker compose up -d
ExecStop=/usr/bin/docker compose down
User=<имя_пользователя>

[Install]
WantedBy=multi-user.target
sudo systemctl daemon-reload
sudo systemctl enable kep && sudo systemctl start kep
```

3.3. Управление системой

Базовые команды управления:

Команда	Назначение
docker compose up -d	Запуск всех контейнеров
docker compose down	Штатная остановка
docker compose ps	Статус контейнеров
docker compose logs -f [<service>]	Журналы
docker compose restart <service>	Перезапуск сервиса
docker stats	Потребление ресурсов
make build / up / dev / monitoring	Целевые операции через make

make health	Проверка работоспособности API
make logs / logs-backend / logs-bots	Журналы
make clean	Удаление контейнеров (тома сохраняются)

Обновление до новой версии:

1) Создать резервную копию (см. III.6). 2) make down. 3) git pull origin CPU_Build. 4) make build && make up. 5) make ps && make health.

3.4. Управление базой знаний

Структура датасета (XLSX, минимально):

- уникальный идентификатор записи;
- текстовое содержимое для индексации;
- категория (опционально);
- метаданные (опционально);
- ссылка на источник (опционально).

Актуализация:

```
cp /path/to/new_dataset.xlsx
/opt/kep/data/government_support_measures.xlsx
docker compose restart backend
```

Время переиндексации зависит от размера датасета (1–10 минут).

Прогресс виден в журнале backend. Принудительная пересборка индекса с очисткой (при смене предметной области):

```
docker compose down
docker volume rm kep_chroma_data
docker compose up -d
```

3.5. Мониторинг и наблюдаемость

Grafana (http://<server_ip>:3000) — преднастроенные дашборды:

- общее состояние контейнеров (uptime, рестарты);
- загрузка CPU и оперативной памяти;
- количество запросов в секунду к API;

- распределение времён отклика (p50/p95/p99);
- счётчики ошибок HTTP по эндпоинтам.

Учётные данные Grafana — переменные `GRAFANA_USER` / `GRAFANA_PASSWORD` (по умолчанию `admin/admin`, рекомендуется сменить).

Portainer (https://<server_ip>:9443) — управление контейнерами через веб-интерфейс: запуск/остановка/перезапуск, журналы, метрики, редактирование переменных окружения.

Журналы сервисов:

```
docker compose logs           # все сервисы
docker compose logs -f       # поток в реальном времени
docker compose logs --tail 100 # последние 100 строк
docker compose logs backend  # один сервис
```

Уровень детализации задаётся `LOG_LEVEL`.

Health-check:

```
make health
curl -X POST http://localhost:8080/health
```

Корректный ответ — HTTP 200 и JSON `status: ok` по каждому компоненту.

При HTTP 503 — см. раздел III.8.

3.6. Резервное копирование и восстановление

Подлежит копированию:

- файл `.env`;
- каталог `data/`;
- Docker-том `postgres_data` (если используется);
- Docker-том `chroma_data`;
- журналы за период по политике организации.

Типовой скрипт резервного копирования:

```
#!/bin/bash
BACKUP_DIR=/var/backups/kep/$(date +%Y%m%d_%H%M%S)
mkdir -p $BACKUP_DIR
cd /opt/kep && docker compose down
cp .env $BACKUP_DIR/
tar czf $BACKUP_DIR/data.tar.gz data/
docker run --rm -v kep_chroma_data:/src -v $BACKUP_DIR:/dst \
```

```
alpine tar czf /dst/chroma_data.tar.gz -C /src .
docker compose up -d
```

Рекомендуется планировать ежедневное копирование через cron, хранить не менее 30 дней, размещать на отдельном носителе и шифровать (AES-256).

Восстановление:

```
cd /opt/kep && docker compose down
cp /var/backups/kep/<метка>/.env .
tar xzf /var/backups/kep/<метка>/data.tar.gz -C .
docker volume rm kep_chroma_data && docker volume create
kep_chroma_data
docker run --rm -v kep_chroma_data:/dst \
    -v /var/backups/kep/<метка>:/src \
    alpine tar xzf /src/chroma_data.tar.gz -C /dst
docker compose up -d && make health
```

3.7. Безопасность

— не размещайте `.env` под системой контроля версий (добавьте в `.gitignore`);

— ограничьте права доступа: `chmod 600 .env`;

— при компрометации токена — отзовите в @BotFather / личном кабинете MAX, выпустите новый и обновите `.env`;

— не передавайте токены в открытых каналах связи;

— ограничьте доступ к административным портам (3000, 9443, 8080) на firewall;

— вынесите внешние сервисы за reverse proxy (Nginx) с принудительным TLS;

— используйте сертификаты от доверенного УЦ;

— регулярно обновляйте Docker, ОС и базовые образы.

Все действия администратора фиксируются штатными средствами ОС (`journalctl`, `/var/log/auth.log`). Действия с базой знаний и обращения пользователей журналируются в `stdout` соответствующих сервисов.

3.8. Диагностика и устранение неисправностей

Симптом	Возможная причина	Действия по устранению
Контейнер Exited	Конфигурация / ресурсы / порты	<code>docker compose logs <service></code> ; проверить <code>.env</code> ; освободить порты
Backend HTTP 503	Недоступен Ollama / ChromaDB	<code>docker compose ps</code> ; перезапустить; проверить диск
Долгий первый запуск (>15 мин)	Загрузка модели Ollama (8 ГБ)	Дождаться; контролировать журнал ollama
Высокий CPU/RAM	Большой поток / нехватка ресурсов	Метрики Grafana; увеличить ресурсы; масштабирование
Бот не отвечает	Неверный токен / сетевая проблема	Проверить токен; исходящий доступ к <code>api.telegram.org</code> / <code>api.max.ru</code> ; журналы ботов
Веб-виджет недоступен	web-frontend не запущен / порт закрыт	<code>docker compose ps</code> ; <code>sudo ufw status</code>
Нерелевантные ответы	Устаревший датасет / повреждён индекс	Актуализация датасета; принудительная переиндексация
Заполнен диск	Накопление журналов / образов	<code>docker system prune -a</code> ; ротация журналов в <code>daemon.json</code>

ТЕХНИЧЕСКАЯ ПОДДЕРЖКА

Контактная информация службы технической поддержки разработчика:

— электронная почта: tickets@gorky.tech;

— режим работы: пн–пт, 09:00–18:00 МСК;

— адрес: 603000, г. Нижний Новгород, ул. Алексеевская, д. 6/16, офис 214
(21).

При обращении укажите канал и обстоятельства возникновения проблемы; приложите выгрузку журналов и результат `make health`. Регламентные сроки реагирования и устранения определены в документе «Описание процессов, обеспечивающих поддержание жизненного цикла программного обеспечения».