Программа для ЭВМ «Обучай»

Руководство по установке

Публичная редакция без инфраструктурных реквизитов

1. Назначение

Документ описывает получение и проверку преднастроенного удалённого экземпляра программы для ЭВМ «Обучай» (далее — Программа). Экземпляр уже развёрнут на сервере в Российской Федерации; эксперту не требуется собирать контейнеры, скачивать исходный код или настраивать домен.

Документ является публичной редакцией и не содержит IP-адресов, паролей и ключей. Конкретные реквизиты веб-шлюза, сервера, отдельный SSH-ключ и пароли тестовых учётных записей передаются в приватном комплекте доступа. На https://obuchai-public.ru размещены лендинг, документация и полноценное приложение; страницы приложения защищены веб-шлюзом, API использует Bearer-аутентификацию.

2. Требования к компьютеру эксперта

- компьютер с актуальным браузером;
- OpenSSH и Python 3;
- доступ к SSH-порту сервера, указанного в приватной инструкции;
- свободный локальный порт 8080 для просмотра сайта и порт 18080 для автоматической проверки.

Установка Программы на компьютер эксперта не требуется. Основной способ работы — веб-интерфейс по HTTPS на домене стенда; дополнительно доступен зашифрованный SSH-туннель.

3. Состав комплекта доступа

- START_HERE.md — краткая инструкция;
- docs/EXPERT.md — полная приватная инструкция с реквизитами сервера;
- credentials.txt — пароли тестовых аккаунтов, секретный файл;
- expert_ssh_ed25519 — отдельный приватный SSH-ключ, секретный файл;
- expert_ssh_ed25519.pub и ssh_known_hosts — публичный ключ и закреплённый ключ сервера;
- scripts/remote_check.py — автоматическая проверка доступа и основных функций;

- DELIVERY_SHA256SUMS — контрольные суммы файлов.

4. Проверка целостности

В каталоге комплекта на Linux выполнить sha256sum -c DELIVERY_SHA256SUMS, на macOS — shasum -a 256 -c DELIVERY_SHA256SUMS. Все файлы должны получить отметку OK.

Приватный SSH-ключ, credentials.txt, конфигурацию .env, дампы базы и резервные копии нельзя публиковать или передавать открытым каналом.

5. Открытие сайта

Открыть https://obuchai-public.ru/generator, войти в веб-шлюз с логином expert и затем в тестовую учётную запись; пароли указаны в приватном credentials.txt. Для дополнительного SSH-доступа установить безопасные права ключа chmod 600 expert_ssh_ed25519 и выполнить команду из docs/EXPERT.md либо подставить выданный адрес сервера в переменную RF_HOST:

```
ssh -i expert_ssh_ed25519 -o IdentitiesOnly=yes -o
StrictHostKeyChecking=yes -o UserKnownHostsFile=ssh_known_hosts -o
ExitOnForwardFailure=yes -N -L 8080:127.0.0.1:18080 expert@$RF_HOST
```

Оставить команду запущенной и открыть http://localhost:8080. Это дополнительный способ доступа: основной экземпляр уже доступен по HTTPS на https://obuchai-public.ru с веб-шлюзом и Bearer-аутентификацией API. После завершения работы туннель останавливается клавишами Ctrl+C.

6. Учётные записи и демонстрация

Выдаются три аккаунта: teacher@obuchai-private.com с бесплатным тарифом, pro@obuchai-private.com и demo@obuchai-private.com с платным тестовым доступом. Общий пароль находится в credentials.txt. Это изолированные записи, а не учётные записи реальных пользователей.

В аккаунте pro@ находится редакционная демонстрационная презентация на 20 слайдов. Её адрес указан в START_HERE.md и docs/EXPERT.md. Демонстрация позволяет проверить редактор, локальные шрифты, PPTX и PDF без производственных данных.

В проверочном профиле отключены платежи реальными деньгами, выплаты, исходящая почта и OAuth. Платный доступ выдан локально. Проверка входа выполняется на заранее созданных аккаунтах.

7. Автоматическая проверка без ИИ

Из каталога комплекта выполнить:

```
python3 scripts/remote_check.py
```

Сценарий проверяет HTTPS-лендинг, веб-шлюз, Bearer API, запрет самостоятельной регистрации и ответы 404 для служебных путей. Затем он входит в три тестовых аккаунта через домен и SSH-туннель, проверяет /auth/me, наличие не менее 10 тем презентаций, демонстрацию на 20 слайдов, а также скачивает и проверяет сигнатуры и размер PPTX и PDF. Туннель создаётся и закрывается автоматически. Итоговое сообщение: «HTTPS и SSH: все проверки без ИИ пройдены».

Дополнительно в браузере проверяются «Мои материалы», открытие демонстрации в редакторе, отображение русского интерфейса и локальных шрифтов, отсутствие сторонних браузерных запросов и выгрузка файлов.

8. Проверка функций с ИИ

В проверочном экземпляре настроен действующий ключ RouterAI. Для повторной проверки функций генерации на сервере выполняются scripts/smoke_routerai.py и scripts/smoke_routerai.py --images. Ключ хранится в секретной конфигурации и не включается в публичную документацию.

Проверяются быстрые и сложные текстовые сценарии, проверка фактов, OCR и анализ изображений, веб-поиск, генерация и доработка изображения. Затем в интерфейсе выполняются пользовательские сценарии создания материала, презентации и проверки работы по фотографии. В поставленном экземпляре живые API-вызовы всех перечисленных ролей и браузерный сценарий создания рабочего листа успешно пройдены. Повторная проверка выполняется после смены ключа, модели или серверного образа.

9. Администрирование и проверка на сервере

Для администрирования используется отдельная SSH-сессия по инструкции docs/EXPERT.md. Корень поставки: /opt/obuchai-rf. Основные команды:

```
cd /opt/obuchai-rf/deploy && docker compose ps
/opt/obuchai-rf/scripts/smoke.sh
```

Smoke-тест проверяет frontend, API с реальным запросом к PostgreSQL, состояние PostgreSQL и наличие не менее 10 тем. Наружу на проверочном VPS открыты SSH 22/tcp и порты 80/443 публичного Caddy с лендингом и приложением. Внутренний Caddy слушает 127.0.0.1:18080 и дополнительно взаимодействует с публичным Caddy через сеть public_app.

10. Резервное копирование и восстановление

Копия создаётся командой sudo /opt/obuchai-rf/scripts/backup.sh. Проверка выполняется командой sudo /opt/obuchai-rf/scripts/verify_backup.sh с указанием каталога копии. Проверяются SHA-256, восстановление PostgreSQL во временную БД, тестовые аккаунты, темы, демонстрация и файлы объектного хранилища. Временная БД затем удаляется.

11. Инвентаризация

Фактический состав и идентификаторы образов фиксируются командами:

```
docker ps --format '{{.Names}} {{.Image}} {{.Status}}'
docker inspect --format '{{.Name}} {{.Image}}' $(docker ps -q)
```

На VPS должны работать шесть контейнеров: Caddy приложения, отдельный публичный Caddy, backend, presentations, PostgreSQL и MinIO-совместимое хранилище. Публичный Caddy использует тот же образ caddy:2, но отдельную конфигурацию и изолированную Docker-сеть. В VM без публичного профиля работают пять контейнеров приложения. Сведения о версиях и идентификаторах образов также находятся в SOURCE_MANIFEST.json, результаты фактической приёмки — в docs/VERIFICATION.md.

12. Типовые ошибки

- SSH-туннель не запускается — проверить ключ, права 600, ssh_known_hosts и свободный локальный порт.
- Сайт не открывается — при запущенном туннеле проверить docker compose ps, Caddy и smoke.sh на сервере.
- Вход не выполняется — сверить аккаунт и пароль из актуального credentials.txt.
- Не формируются PPTX/PDF — проверить backend, presentations и повторить remote_check.py.
- Не работают ИИ-функции — проверить RouterAI-ключ, баланс и результаты живого smoke-теста; базовый HTTP healthcheck не подтверждает работу ИИ.