ИНСТРУКЦИЯ ПО УСТАНОВКЕ

программы для ЭВМ «Версиум AI»

1. Общие сведения

Наименование	Версиум AI (Versium AI)
Правообладатель	ООО «ВЕРСИУМ», ИНН 6670527467, ОГРН 1246600061014
Тип программы	Самостоятельно функционирующая программа для ЭВМ, серверное приложение с веб-интерфейсом
Назначение документа	Порядок установки, первичной настройки и проверки работоспособности программы
Способ поставки	Дистрибутив для установки в Docker-контейнерах: онлайн-установка с сервера дистрибутивов правообладателя или офлайн-комплект для изолированного контура
Варианты установки	All-in-one (все компоненты на одном сервере); разнесённая установка (приложение и СУБД на отдельных серверах)
Операционные системы	Windows, Linux
СУБД	PostgreSQL, Postgres Pro

Для установки правообладатель предоставляет учётные данные доступа к реестру образов и серверу дистрибутивов, а также лицензионный ключ, выпущенный на доменное имя экземпляра. Исходный код при установке не передаётся: все компоненты поставляются в виде готовых контейнерных образов.

2. Состав комплекта поставки

Элемент	Назначение
Установщик `install-versium-ai.sh`	Проверка окружения, установка Docker при его отсутствии, загрузка и проверка дистрибутива, формирование конфигурации, запуск, обновление и диагностика установки
Комплект конфигурации развёртывания	Файлы `docker-compose.yml` и `.env.example` для вариантов all-in-one и с внешней СУБД, каталог `ssl/` для сертификатов
Контейнерные образы программы	`versium-ai-api` (серверная часть), `versium-ai-frontend` (веб-интерфейс), `versium-ai-executor` (встроенный исполнитель), `versium-ai-nginx` (входной веб-сервер с поддержкой TLS)
Базовые образы	СУБД PostgreSQL 17 (только для варианта all-in-one), кэш в оперативной памяти, совместимый с протоколом Redis

Элемент	Назначение
Офлайн-комплект (по запросу)	Архив `versium-ai-offline-<версия>.tar.gz`: образы, конфигурация, манифест версии с контрольной суммой, установщик и при необходимости пакеты Docker Engine для целевой ОС
Лицензионный ключ	Файл с расширением `.lic`, подписанный электронной подписью ООО «ВЕРСИУМ»

После установки на сервере работают следующие контейнеры:

Контейнер	Назначение
nginx	Единственная внешняя точка входа: порты 80 (перенаправление на HTTPS) и 443
frontend	Веб-интерфейс; проксирует обращения к API, MCP и каналам обмена событиями на серверную часть
api	Серверная часть: шлюз, политики, маскирование, аудит, лицензирование
executor	Встроенный исполнитель AI-агентов; подключается к серверной части автоматически
cache	Кэш в оперативной памяти
postgres	СУБД (только в варианте all-in-one)

3. Требования к аппаратному и программному обеспечению

3.1. Аппаратные требования

Конфигурация	ОЗУ	Процессор	Диск
Минимальная	4 ГБ	2 vCPU	40 ГБ NVMe/SSD
Рекомендуемая	8 ГБ	2–4 vCPU	100 ГБ NVMe/SSD
С учётом роста	16 ГБ	4 vCPU	200 ГБ NVMe/SSD

Для ознакомления с программой и пилотной эксплуатации достаточно минимальной конфигурации. Объём загружаемых при первой установке образов составляет около 1,2 ГБ.

3.2. Программные требования

- **операционная система Linux** – Ubuntu 22.04 и 24.04, Debian 12 и 13, дистрибутивы семейства RHEL; российские дистрибутивы (Astra Linux, ALT, РЕД ОС) – при предварительно установленном Docker Engine;
- **операционная система Windows** – Windows с компонентом WSL 2 и Docker Desktop (Linux-контейнеры), см. п. 6.4;
- **Docker Engine** версии 24 и выше с подключаемым модулем Docker Compose v2 (на Linux устанавливается установщиком автоматически при его отсутствии);
- **служебные утилиты** – bash, curl, tar, openssl, python3 (входят в стандартную поставку перечисленных дистрибутивов);
- **внешняя СУБД** (для разнесённой установки) – PostgreSQL версии 15 и выше (рекомендуется 17) или Postgres Pro;

- **рабочее место пользователя** – современный веб-браузер.

3.3. Сетевые требования

- доменное имя экземпляра, разрешаемое в адрес сервера (запись DNS или файл hosts на рабочем месте эксперта); на это имя выпускается лицензионный ключ;
- свободные на сервере порты TCP 80 и 443 (допускается задать другие порты);
- для онлайн-установки – исходящий доступ по HTTPS к серверу дистрибутивов `releases.versium.ru`, реестру образов `registry.versium.ru` и реестрам базовых образов; для изолированного контура используется офлайн-комплект;
- для разнесённой установки – сетевой доступ с сервера приложения к порту СУБД (по умолчанию 5432).

4. Подготовка сервера

1. Установить операционную систему и обновить пакеты штатными средствами.
2. Создать учётную запись администратора с правами sudo (установщик запускается от обычного пользователя и сам запрашивает повышение прав).
3. Убедиться, что доменное имя экземпляра разрешается в адрес сервера, а порты 80 и 443 свободны:

```
getent hosts <доменное имя>
sudo ss -ltnp | grep -E ':(80|443) '
```

5. Получение дистрибутива

Онлайн-установка выполняется установщиком с сервера дистрибутивов правообладателя. Установщик загружает комплект конфигурации выбранной версии (по умолчанию – последней стабильной), сверяет его контрольную сумму SHA-256 с манифестом версии и выполняет вход в реестр образов по учётным данным, предоставленным правообладателем:

```
curl -fsSL -o install-versium-ai.sh \
    https://releases.versium.ru/install-versium-ai.sh
```

Для сервера без доступа в интернет правообладатель предоставляет офлайн-комплект `versium-ai-offline-<версия>.tar.gz`, который переносится на сервер любым доступным способом. Установка из офлайн-комплекта не обращается к внешним ресурсам.

6. Установка в варианте all-in-one

6.1. Онлайн-установка

Запустить установщик от имени пользователя с правами sudo:

```
bash install-versium-ai.sh
```

Установщик последовательно выполняет:

- определение дистрибутива ОС, установку Docker Engine и Docker Compose при их отсутствии, настройку ротации журналов контейнеров, добавление пользователя в группу docker;
- запрос логина и пароля реестра образов `registry.versium.ru` и вход в реестр;
- загрузку комплекта конфигурации и проверку контрольной суммы;

- запрос размещения СУБД: ответ `1` – СУБД в составе установки (all-in-one), ответ `2` – внешний сервер СУБД (см. раздел 7);
- запрос параметров: доменное имя экземпляра (`ApplicationDomain`), пароль встроенной учётной записи администратора `admin`, путь к TLS-сертификату (Enter – самоподписанный), пароль СУБД (сертификат – в формате PEM, ключ без пароля; самоподписанный сертификат выпускается сроком на один год); внутренние секреты подписи сеансов и подключения исполнителя формируются автоматически; если порт 80 или 443 занят, установщик предложит другой;
- запись параметров в файл `.env` с правами доступа 600;
- загрузку образов, запуск контейнеров и ожидание готовности веб-интерфейса (до 5 минут).

По завершении выводится адрес веб-интерфейса вида `https://<доменное имя>`. Каталог установки по умолчанию – `~/versium-ai`, журнал установки – `~/.versium-install/install-<время>.log`. Данные хранятся в именованных томах Docker (база данных, рабочие каталоги исполнителя), которые сохраняются при перезапуске и обновлении.

6.2. Установка из офлайн-комплекта

```
mkdir ~/versium-ai-offline
tar -xzf versium-ai-offline-<версия>.tar.gz -C ~/versium-ai-offline
cd ~/versium-ai-offline
bash install-versium-ai.sh
```

Установщик обнаруживает рядом с собой каталоги образов и конфигурации и переходит в офлайн-режим: загружает образы из архива, при отсутствии Docker устанавливает его из пакетов комплекта и далее выполняет те же шаги, что и при онлайн-установке, без обращения к реестру и серверу дистрибутивов.

6.3. Параметры установщика

Параметр	Назначение
`--version <тег>`	Установка конкретной версии вместо последней стабильной
`--install-dir <путь>`	Каталог установки (по умолчанию `~/versium-ai`)
`--http-port`, `--https-port`	Внешние порты HTTP и HTTPS вместо 80 и 443
`--ssl-cert`, `--ssl-key`	Собственный TLS-сертификат и закрытый ключ в формате PEM
`--external-db`	Установка с внешней СУБД (параметры подключения – см. раздел 7)
`--start-pack`	Применение стартового пакета конфигурации после установки
`upgrade`	Обновление существующей установки (раздел 10)
`update-check`	Диагностика установки без внесения изменений

6.4. Установка на сервер под управлением Windows

На Windows программа устанавливается в Linux-контейнерах Docker Desktop с подсистемой WSL 2 с использованием того же комплекта конфигурации:

1. Включить компонент WSL 2, установить Docker Desktop в режиме Linux-контейнеров и убедиться, что команда `docker compose version` выполняется.
2. Распаковать комплект конфигурации (вариант all-in-one) в каталог установки, например `C:\versium-ai`, скопировать `.env.example` в `.env` и заполнить параметры `VERSION`, `ApplicationDomain`, `POSTGRES_PASSWORD`, `ADMIN_PASSWORD`, `JWT_SECRET`, `EXECUTOR_TOKEN` (два последних – случайные строки не короче 32 байт).
3. Загрузить образы и запустить контейнеры в PowerShell:

```
cd C:\versium-ai
docker login registry.versium.ru
docker compose pull
docker compose up -d
```

В изолированном контуре вместо `docker login` и `docker compose pull` образы загружаются из офлайн-комплекта командой `docker load -i <файл образа>.tar`. Первичная настройка и проверка выполняются так же, как на Linux (разделы 8, 9).

7. Установка с внешней СУБД

В разнесённом варианте СУБД PostgreSQL или Postgres Pro размещается на отдельном сервере; на сервере приложения запускаются все контейнеры, кроме СУБД. На сервере СУБД заранее создаются база данных и пользователь приложения (программа не создаёт базу данных самостоятельно):

```
CREATE USER versiumai WITH PASSWORD '<пароль пользователя>';
CREATE DATABASE versiumaidb OWNER versiumai;
```

Пользователю требуются права CREATE, SELECT, INSERT, UPDATE, DELETE, REFERENCES на схему public и на схемы модулей, создаваемые при первом запуске. Доступ с сервера приложения к СУБД разрешается в настройках аутентификации СУБД (`pg_hba.conf`).

Установка на сервере приложения выполняется так же, как в разделе 6, с параметром `--external-db`; установщик запросит адрес и порт СУБД, имя базы, пользователя и пароль. Параметры можно передать сразу:

```
bash install-versium-ai.sh --external-db --db-server <адрес СУБД> \
    --db-port 5432 --db-name versiumaidb --db-user versiumai
```

Вместо отдельных параметров допускается полная строка подключения `--db-connection-string`. Структура базы данных создаётся и обновляется автоматически при запуске серверной части.

8. Первичная настройка

8.1. Вход администратора

Открыть в браузере `https://<доменное имя>` (при нестандартном порте – с указанием порта). При самоподписанном сертификате браузер выводит предупреждение, которое следует принять. Войти под учётной записью `admin` с паролем, заданным при установке.

8.2. Активация лицензионного ключа

До загрузки лицензионного ключа программа работает в ограниченном режиме: доступны только вход и управление лицензиями, остальные функции заблокированы (в интерфейсе

отображается сообщение «Нет активной лицензии»). Для активации:

1. Открыть раздел «Администрирование» → «Лицензии».
2. В блоке «Загрузка лицензии» выбрать или перетащить файл `.lic`, полученный от правообладателя, и нажать «Загрузить».
3. Убедиться, что в карточке «Статус лицензии» отображается состояние «Лицензировано», а также заказчик, URL экземпляра и срок действия.

Электронная подпись ключа проверяется встроенным модулем лицензирования локально по открытому ключу правообладателя, без обращения к внешним серверам. Ключ действует только для доменного имени, на которое он выпущен: оно должно совпадать с параметром `ApplicationDomain`, иначе ключ получает статус «Не совпадает URL».

8.3. Подключение провайдера языковой модели

1. Открыть раздел «Настройка ИИ» → «Провайдеры» и нажать «Добавить провайдера».
2. Указать имя и тип провайдера, базовый URL программного интерфейса и ключ доступа. Провайдером может быть внешний сервис, внутренняя модель организации или локальная модель в контуре заказчика; ключ доступа хранится в шлюзе в зашифрованном виде.
3. Нажать «Модели» у созданного провайдера и добавить одну или несколько моделей.
4. При необходимости в разделе «Безопасность» → «Маскирование» настроить механизмы маскирования и привязать их к провайдеру.

9. Проверка работоспособности

Проверить состояние контейнеров и доступность сервисов на сервере:

```
cd ~/versium-ai
docker compose ps
curl -k https://<доменное имя>/healthz
curl -k https://<доменное имя>/api/v1/version
```

Ожидаемый результат: все контейнеры в состоянии «Up», контейнеры api, frontend и postgres – «healthy»; проверка `/healthz` возвращает `ok`; запрос версии возвращает номер установленной версии.

Проверить работу в веб-интерфейсе:

- состояние лицензии – «Лицензировано» (раздел «Лицензии»);
- встроенный исполнитель отображается как подключённый (раздел «Исполнители»);
- в разделе «Чат» отправить запрос к подключённой модели и получить ответ;
- запрос и ответ отражены в разделе «Аудит» → «Аудит диалогов».

Диагностика установки без внесения изменений выполняется командой `bash install-versium-ai.sh update-check` из каталога установки; журналы серверной части – командой `docker compose logs api --tail 100`.

10. Обновление и удаление

10.1. Обновление

Перед обновлением рекомендуется создать резервную копию базы данных (копия включает и ключи защиты данных, хранящиеся в базе); в варианте all-in-one:

```
cd ~/versium-ai
docker compose exec -T postgres pg_dump -U versiumai versiumaidb | gzip >
backup.sql.gz
bash install-versium-ai.sh upgrade
```

Установщик загружает комплект конфигурации новой версии, проверяет контрольную сумму, заменяет `docker-compose.yml`, загружает новые образы и перезапускает контейнеры. Файл `.env`, сертификаты и тома с данными не изменяются; структура базы данных обновляется автоматически при запуске. Конкретная версия задаётся параметром `--version`. Для офлайн-обновления новый офлайн-комплект распаковывается в отдельный каталог, а установщик из него запускается из каталога установки: `bash ~/versium-ai-offline/install-versium-ai.sh upgrade`.

10.2. Удаление

```
cd ~/versium-ai
docker compose down -v
cd ~ && rm -rf ~/versium-ai
docker image rm $(docker images -q 'registry.versium.ru/versium-ai-*')
docker logout registry.versium.ru
```

Команда `docker compose down -v` безвозвратно удаляет тома с данными; для остановки с сохранением данных используется `docker compose down`. При разнесённой установке база данных на сервере СУБД удаляется отдельно средствами СУБД.

11. Типовые проблемы при установке

Признак	Действие
Порт 80 или 443 занят, ошибка `address already in use`	Освободить порт либо задать другие порты параметрами `--http-port`, `--https-port`; на работающей установке – `upgrade --https-port <порт>`
Не выполнен вход в `registry.versium.ru`	Проверить учётные данные реестра, предоставленные правообладателем, и доступ к реестру по HTTPS
Не удалась автоматическая установка Docker	Установить Docker Engine и Docker Compose из репозитория ОС вручную и повторить запуск установщика
Сертификат не принимается установщиком	Сертификат и ключ должны быть в формате PEM и составлять пару; пароль с ключа снимается командой `openssl rsa -in key.pem -out key-plain.pem`
Контейнер api перезапускается	Просмотреть `docker compose logs api --tail 100`; чаще всего причина – ошибка подключения к СУБД, проверить параметры в `.env`
Контейнер api в состоянии «starting» дольше 60 секунд	При первом запуске создание структуры базы данных занимает до 90 секунд; при большей задержке – просмотреть журнал api

Признак	Действие
Сообщение «Нет активной лицензии» или статус «Не совпадает URL»	Загрузить лицензионный ключ; доменное имя ключа и параметр `ApplicationDomain` в `.env` должны совпадать; после изменения `.env` выполнить `docker compose up -d`
Исполнитель не подключается, ответ чата приходит целиком	При работе через дополнительный обратный прокси-сервер включить на нём поддержку WebSocket и отключить буферизацию ответов
Изменённый в `.env` пароль администратора не действует	Пароль из `.env` применяется только при первичной инициализации базы данных; сменить пароль в интерфейсе («Сменить пароль»)