123CMS

ПРОГРАММА ДЛЯ ЭВМ

Инструкция по установке экземпляра программного обеспечения

Порядок предоставления экземпляра в облачном варианте и полная процедура установки в контуре заказчика (on-premises)

Наименование ПО	**Программа для ЭВМ «123CMS»**
Правообладатель	**Сарбашев Никита Борисович**
Версия документа	**1.2**
Год	**2026 г.**

1. Общие сведения

1.1. Назначение документа

Настоящий документ содержит указания по установке экземпляра программы для ЭВМ «123CMS» — системы управления контентом для сетей цифровых информационных экранов (digital signage).

Документ описывает два способа получения работающего экземпляра:

- **облачный вариант** — экземпляр разворачивается и эксплуатируется правообладателем на его собственной инфраструктуре, заказчик получает доступ к готовой консоли;
- **вариант установки в контуре заказчика (on-premises)** — весь программный комплекс устанавливается и работает на сервере заказчика, внутри его собственной сети.

Программное обеспечение в обоих вариантах одно и то же: один и тот же набор сервисов, одна и та же консоль оператора, один и тот же прикладной программный интерфейс. Различаются только сторона, эксплуатирующая инфраструктуру, способ подтверждения прав использования и типовой режим связи с медиаплеерами.

1.2. Краткие сведения о программе

123CMS — мультитенантная (многоарендная) серверная система, предоставляющая единую веб-консоль для управления парком медиаплееров: загрузка и подготовка медиаматериалов, формирование плейлистов, многослойное расписание показов, оперативные команды устройствам, мониторинг состояния и аналитика воспроизведения.

Поддерживаемое оборудование воспроизведения — медиаплееры SpinetiX HMP. Взаимодействие с ними выполняется исключительно по стандартным сетевым протоколам: HTTPS для доставки медиаматериалов по WebDAV, JSON-RPC поверх HTTPS для управляющих команд, HTTP для чтения состояния устройства. Каких-либо проприетарных драйверов, агентов или закрытых компонентов на стороне сервера для этого не требуется.

1.3. Состав программного комплекса

Экземпляр 123CMS представляет собой набор из семи взаимодействующих сервисов, запускаемых в контейнерах средствами Docker Compose:

Сервис	Базовый образ	Назначение
`caddy`	`caddy:2.8`	Обратный прокси-сервер и терминация TLS. Единственный сервис, публикующий сетевые порты (80 и 443)
`api`	`123cms-api:latest`	Прикладной сервер (FastAPI, Python 3.12): программный интерфейс консоли, интерфейс для медиаплееров и фоновые обработчики
`migrate`	`123cms-api:latest`	Одноразовая задача приведения схемы базы данных к актуальному состоянию (Alembic)
`postgres`	`postgres:16`	Реляционная база данных — основное хранилище состояния системы
`redis`	`redis:7`	Очереди задач, блокировки, кратковременный кеш и состояние подсистемы безопасности
`webdav`	`123cms-webdav:bookworm`	Источник доставки контента: nginx с модулями WebDAV, отдающий подготовленное дерево файлов только на чтение

Сервис	Базовый образ	Назначение
`db-backup`	`postgres:16`	Ежесуточное создание зашифрованной резервной копии базы данных

Все состояние экземпляра хранится в именованных томах Docker: `pgdata` (база данных), `redisdata`, `storage` (загруженные исходные файлы, перекодированные версии и подготовленные для каждого плеера деревья файлов), `backup-data` (резервные копии), `caddy-data` и `caddy-config` (сертификаты и рабочая конфигурация TLS). Файловые системы самих контейнеров одноразовые и не содержат данных, требующих сохранения.

2. Предоставление экземпляра в облачном варианте

2.1. Порядок предоставления

В облачном варианте установка на стороне заказчика не выполняется. Экземпляр создается правообладателем (или уполномоченным им лицом) на инфраструктуре, размещенной и эксплуатируемой правообладателем.

Последовательность действий:

1. Заказчик направляет заявку с указанием наименования организации, требуемого количества экранов, числа учетных записей операторов и объема хранилища медиаматериалов.
2. Правообладатель создает арендатора (tenant) — изолированную область данных заказчика в общей платформе, — задает параметры подписки (тарифный план, число лицензионных мест, лимиты по площадкам, пользователям и хранилищу, срок действия) и создает первую учетную запись администратора.
3. Заказчик получает адрес консоли, адрес электронной почты первого администратора и одноразовый пароль, а также данные для подключения второго фактора аутентификации.
4. При первом входе администратор меняет пароль и завершает подключение двухфакторной аутентификации по алгоритму TOTP.

2.2. Требования на стороне заказчика

Для работы в облачном варианте на рабочих местах операторов достаточно:

- современного веб-браузера с поддержкой актуальных версий стандартов HTML5, CSS и ECMAScript (Google Chrome, Microsoft Edge, Mozilla Firefox, Safari) — консоль представляет собой одностраничное веб-приложение и не требует установки клиентского программного обеспечения;
- доступа к сети Интернет по протоколу HTTPS к адресу консоли;
- мобильного или настольного приложения — генератора одноразовых кодов TOTP для второго фактора аутентификации.

Медиаплееры должны иметь возможность устанавливать исходящие соединения по HTTPS к адресу экземпляра. Входящий доступ к плеерам со стороны сети Интернет не требуется: в облачном варианте по умолчанию применяется режим RPC Concentrator, при котором плеер сам инициирует соединение с сервером.

2.3. Подтверждение прав использования

В облачном варианте подписка представлена записью в базе данных платформы и проверяется на стороне сервера при каждом запросе. Продление подписки, изменение числа лицензионных мест и квот выполняются правообладателем; каких-либо файлов лицензии на стороне заказчика устанавливать и обслуживать не требуется.

3. Установка в контуре заказчика (on-premises)

3.1. Требования к серверу

Установка выполняется на сервер под управлением операционной системы семейства Linux. Требования к программной среде:

Требование	Примечание
Операционная система	Linux с поддержкой Docker (проверено на дистрибутивах на базе Debian)
Docker Engine	Команда `docker version` должна выполняться успешно
Плагин Docker Compose	Команда `docker compose version` должна выполняться успешно
`openssl`	Используется установщиком однократно для генерации секретов
Свободное дисковое пространство	С запасом на объем медиатеки, перекодированных версий, деревья доставки для каждого плеера и резервные копии
Сетевые порты	На хосте должны быть свободны порты 80 и 443
Имя хоста консоли	Может быть внутренним, например `signage.corp.local`

Аппаратные требования (объем оперативной памяти, число вычислительных ядер, объем хранилища) определяются размером парка медиаплееров и интенсивностью перекодирования видео и согласовываются со службой технической поддержки при планировании внедрения.

Экземпляр может быть установлен в полностью изолированном от сети Интернет контуре: проверка прав использования выполняется локально, обращений к внешним серверам при работе не происходит, а сертификат TLS для внутреннего имени хоста выпускается встроенным удостоверяющим центром прокси-сервера Caddy.

3.2. Состав поставляемого комплекта

Для установки заказчику передается комплект поставки, содержащий:

1. **Дерево каталогов `infra/`** — файлы описания стека и установщик. Значимые для установки каталоги: `infra/onprem/` (файл `compose.onprem.yml`, конфигурация прокси `Caddyfile.onprem`, установщик `install.sh`, каталог `licence/` для файла лицензии, каталог `app/` со статическими файлами собранной консоли), а также каталоги `infra/postgres/` (сценарии первичной инициализации базы данных), `infra/deploy/` (конфигурация и файл сборки образа WebDAV) и `infra/backup/` (сценарии резервного копирования и учебного восстановления), на которые ссылаются по относительным путям файл описания стека и установщик.
2. **Архив образов контейнеров** `123cms-images.tar`, содержащий образы `123cms-api:latest` и `123cms-webdav:bookworm`. Для хостов, имеющих доступ к реестру образов, вместо архива может быть предоставлен доступ к реестру.

3. **Файл лицензии** `licence.json` — подписанный офлайн-токен, выпущенный для конкретного экземпляра.
4. **Открытый ключ платформы** — короткая строка в кодировке base64, передаваемая вместе с файлом лицензии в виде значения переменной окружения либо в виде файла `pubkey.b64`.

3.3. Подготовка сервера

1. Установить Docker Engine и плагин Docker Compose штатными средствами используемого дистрибутива и убедиться в работоспособности:

```
docker version
docker compose version
```

1. Разместить дерево `infra/` из комплекта поставки в постоянном каталоге на сервере, например `/opt/123cms/infra`.

1. Загрузить образы контейнеров. Для изолированного контура:

```
docker load -i 123cms-images.tar
```

Для контура с доступом к реестру — получить образы из реестра и присвоить им теги `123cms-api:latest` и `123cms-webdav:bookworm`.

1. Убедиться, что образы доступны локально:

```
docker image inspect 123cms-api:latest --format '{{.Id}}'
```

1. Определить имя хоста, по которому будет доступна консоль, и обеспечить его разрешение в адрес сервера средствами внутренней службы доменных имен либо записями в файле `hosts` на рабочих местах операторов.

3.4. Размещение файла лицензии

Файл лицензии и, при поставке отдельным файлом, открытый ключ платформы размещаются в каталоге `infra/onprem/licence/`:

```
cp /path/to/licence.json infra/onprem/licence/licence.json
cp /path/to/pubkey.b64   infra/onprem/licence/pubkey.b64
```

Второй файл не обязателен: открытый ключ может быть передан установщику через переменную окружения `LICENSE_PUBLIC_KEY_B64`.

Файл лицензии представляет собой компактный токен, содержащий набор утверждений о параметрах предоставленных прав: наименование организации, тарифный план, число лицензионных мест, лимиты по площадкам, пользователям и объему хранилища, перечень включенных языков интерфейса и дополнительных модулей, даты начала и окончания срока, длительность льготного периода и признак бессрочной лицензии. Токен подписан по алгоритму Ed25519 закрытым ключом, который никогда не передается заказчику; в состав поставляемого экземпляра входит только соответствующий открытый ключ, достаточный для проверки подписи. Проверка выполняется полностью локально, обращений к внешним серверам не производится.

3.5. Запуск установщика

Установка выполняется сценарием `install.sh`, находящимся в каталоге `infra/onprem/`. Сценарий управляется переменными окружения и рассчитан на неинтерактивное выполнение: значение запрашивается у оператора только в том случае, если оно не задано и сценарий запущен в терминале; в остальных случаях отсутствие обязательного значения приводит к остановке с сообщением об ошибке.

Сценарий идемпотентен — повторный запуск сохраняет ранее сгенерированные секреты и заново применяет все остальные шаги, поэтому его безопасно выполнять повторно после устранения выявленной проблемы.

Пример запуска:

```
cd infra/onprem
DOMAIN=signage.corp.local \
ADMIN_EMAIL=admin@corp.example \
LICENSE_PUBLIC_KEY_B64=<открытый-ключ-платформы> \
./install.sh
```

Переменные, принимаемые установщиком:

Переменная	Обязательность	Назначение
`DOMAIN`	Обязательна	Имя хоста, по которому публикуется консоль. Может быть внутренним
`ADMIN_EMAIL`	Обязательна	Адрес электронной почты первой учетной записи администратора
`ADMIN_NAME`	Необязательна	Отображаемое имя администратора; при отсутствии принимает значение `Administrator`
`LICENSE_PUBLIC_KEY_B64`	Обязательна, если не размещен файл `licence/pubkey.b64`	Открытый ключ платформы в кодировке base64url

3.6. Последовательность действий установщика

Установщик выполняет следующие шаги.

Шаг 1. Проверка предварительных условий. Проверяется наличие исполняемого файла `docker`, работоспособность плагина `docker compose` и наличие локально загруженного образа `123cms-api:latest`. При отсутствии любого из условий установка прекращается с указанием причины.

Шаг 2. Сбор исходных данных. Считываются `DOMAIN`, `ADMIN_EMAIL` и открытый ключ платформы (из переменной окружения либо из файла `licence/pubkey.b64`); проверяется наличие файла `licence/licence.json`.

Шаг 3. Однократная генерация секретов. При первом запуске командой `openssl rand -base64 32` формируются два ключа:

- `ENCRYPTION_KEY` — ключ симметричного шифрования данных, хранимых в базе в защищенном виде (пароли управления медиаплеерами, пароли WebDAV, секреты второго фактора аутентификации);
- `BACKUP_ENC_PASSPHRASE` — парольная фраза шифрования резервных копий.

Оба значения вместе с `DOMAIN`, `ADMIN_EMAIL` и открытым ключом записываются в файл `.env.onprem` с правами доступа `600`. При повторном запуске установщик сначала считывает существующий файл `.env.onprem` и сохраняет оба секрета без изменений.

> Повторная генерация `ENCRYPTION_KEY` делает нечитаемыми все ранее сохраненные в базе защищенные данные. Повторная генерация `BACKUP_ENC_PASSPHRASE` делает невозможной расшифровку всех ранее созданных резервных копий. Именно поэтому установщик никогда не перезаписывает эти значения самостоятельно.

Шаг 4. Проверка лицензии до запуска сервисов. Внутри образа `123cms-api` выполняется сценарий `scripts/verify_licence.py`, проверяющий подпись файла `licence/licence.json` по переданному открытому ключу. Сценарий выводит на экран содержащиеся в лицензии сведения (организация, тарифный план, число мест, срок действия) и завершает установку с ошибкой при неверной подписи. Некорректный файл лицензии не может пройти дальше этого шага.

Шаг 5. Запуск стека и приведение схемы базы данных. Выполняется команда:

```
docker compose -f compose.onprem.yml up -d --build
```

Сервис `migrate` (`alembic upgrade head`) объявлен зависимостью сервиса `api` в файле описания стека, поэтому схема базы данных приводится к актуальному состоянию автоматически, до запуска прикладного сервера. Миграции выполняются от имени отдельной роли базы данных `cms_migrate`, обладающей правами на изменение структуры; сам прикладной сервер работает от имени роли `cms_runtime` с ограниченным набором прав.

Шаг 6. Ожидание готовности прикладного сервера. Установщик опрашивает состояние сервиса `api` каждые 5 секунд в течение не более 5 минут. Если состояние `healthy` не достигнуто, установка завершается с указанием на просмотр журнала:

```
docker compose -f compose.onprem.yml logs api
```

Шаг 7. Создание арендатора и первого администратора. Из проверенной лицензии считывается наименование организации, из него формируется машиночитаемый идентификатор, после чего внутри работающего контейнера `api` выполняется сценарий `scripts/provision_live.py`. Создаются арендатор и первая учетная запись администратора; в терминал выводятся одноразовый пароль и данные для подключения второго фактора аутентификации.

Шаг 8. Вывод результата. Печатается адрес консоли `https://<DOMAIN>/app`, напоминание о смене пароля при первом входе и порядок продления лицензии.

3.7. Развернутая конфигурация

После успешной установки на сервере работают семь сервисов, перечисленных в п. 1.3. Сетевая модель развернутого экземпляра:

- порты в сеть хоста публикует только сервис `caddy` — 80 и 443;
- сервисы `postgres` и `redis` доступны исключительно внутри внутренней сети Docker Compose и не имеют опубликованных портов;
- сервис `webdav` также не публикует портов и доступен только через прокси-сервер по пути `/dav`;
- все входящие соединения — как от браузеров операторов, так и от медиаплееров — терминируются на прокси-сервере по протоколу TLS.

Правила маршрутизации прокси-сервера:

Путь запроса	Назначение
`/api/*`, `/player-api/*`, `/ws/*`	Прикладной сервер: интерфейс консоли, интерфейс для медиаплееров, соединения WebSocket
`/dav/*`	Источник доставки контента. Префикс `/dav` передается источнику без изменений — это необходимо для корректной работы обхода каталогов медиаплеерами
`/app/*`	Статические файлы консоли из каталога `./app`
`/platform`, `/platform/*`	Перенаправление в `/app{uri}`
`/`	Перенаправление на `/app/`

Для внутреннего имени хоста в файле `Caddyfile.onprem` задан режим `tls internal`: сертификат выпускается встроенным удостоверяющим центром прокси-сервера, обращение к внешним службам проверки не производится и экземпляр работает в изолированном контуре. При наличии собственного сертификата, выданного удостоверяющим центром организации, строка `tls internal` заменяется на `tls /path/cert.pem /path/key.pem`, а файлы сертификата и ключа монтируются в контейнер.

Прокси-сервер дополнительно устанавливает в каждом ответе заголовки `Strict-Transport-Security` и `X-Content-Type-Options: nosniff`, сжимает ответы и задает заголовок `X-Real-IP` из фактического сетевого адреса подключения — это исключает подмену клиентом адреса, на основании которого принимаются решения о блокировках и ограничении частоты запросов.

3.8. Проверка после установки

1. Убедиться, что все сервисы запущены, а сервисы, имеющие проверку работоспособности (`postgres`, `redis`, `api`, `webdav`), находятся в состоянии `healthy`:

```
docker compose -f compose.onprem.yml ps
```

Сервис `migrate` должен быть завершен с нулевым кодом возврата.

1. Проверить состояние прикладного сервера:

```
docker compose -f compose.onprem.yml exec -T api \
  python -c "import urllib.request;
print(urllib.request.urlopen('http://localhost:8000/healthz').read().decode())"
```

Ответ должен содержать `"status": "ok"` и состояние `"ok"` для компонентов `database`, `redis` и `workers`.

1. Проверить доступность консоли:

```
curl -o /dev/null -s -w '%{http_code}\n' https://<DOMAIN>/app/
```

Ожидаемый код ответа — `200`.

1. Открыть в браузере адрес `https://<DOMAIN>/app`, при использовании внутреннего удостоверяющего центра принять выпущенный сертификат, войти с указанным адресом электронной почты администратора и одноразовым паролем, полученным от установщика, и сменить пароль при первом входе.

3.9. Первичная настройка после установки

После первого входа рекомендуется выполнить в разделе «Система» консоли:

1. Создать площадки (объекты) с указанием города и часового пояса — расписание показов вычисляется по местному времени площадки.
2. Для площадок, где медиаплееры будут работать в прямом режиме, задать разрешенные диапазоны сетевых адресов, по которым сервер обращается к устройствам.
3. Пригласить учетные записи операторов, назначив каждой минимально достаточные роль и область видимости по площадкам.
4. При необходимости скорректировать пресеты перекодирования и политику воспроизведения (обязательность согласования загружаемых материалов, правила разделения материалов в ротации).
5. Проверить фирменное оформление консоли и перечень включенных модулей.

Порядок работы с перечисленными разделами изложен в документе «Руководство по эксплуатации ПО».

4. Обслуживание установленного экземпляра

4.1. Продление срока действия лицензии

Продление представляет собой замену файла лицензии на новый подписанный файл с более поздней датой окончания срока:

```
cp /path/to/new-licence.json infra/onprem/licence/licence.json
docker compose -f compose.onprem.yml restart api
```

Переустановка и изменение данных не требуются. Перед заменой файл может быть проверен:

```
docker run --rm -v "$(pwd)/licence:/run/licence:ro" \
  -e LICENSE_FILE=/run/licence/licence.json \
  -e LICENSE_PUBLIC_KEY_B64=<открытый-ключ-платформы> \
  123cms-api:latest python scripts/verify_licence.py
```

Ограничения применяются мягко. Истечение срока действия либо отсутствие, повреждение или непрохождение проверки подписи файла лицензии блокируют только действия, расширяющие использование системы: подключение новых медиаплееров, загрузку новых материалов и создание новых показов. Воспроизведение уже опубликованного контента, мониторинг, оповещения, запуск экстренных сценариев, работа с журналом аудита и выгрузка отчетов сохраняются в полном объеме при любом состоянии лицензии. Ни при каких обстоятельствах экземпляр не прекращает работу и не гасит работающие экраны.

Для защиты от обхода истечения срока путем перевода системных часов назад экземпляр хранит наибольшее наблюдавшееся значение времени. Если локальное время оказывается существенно раньше этого значения, состояние лицензии считается сомнительным и экземпляр переходит в льготное состояние с записью соответствующего события в журнал.

4.2. Обновление на новую версию

Обновление выполняется по процедуре, изложенной в документе «Описание процессов, обеспечивающих поддержание жизненного цикла ПО». Краткая последовательность:

1. Создать резервную копию базы данных.
2. Загрузить образы нового выпуска (`docker load -i 123cms-images.tar`) и обновить статические файлы консоли в каталоге `infra/onprem/app/` .
3. Выполнить миграции базы данных отдельным явным шагом и дождаться нулевого кода возврата:

```
docker compose -f compose.onprem.yml run --rm migrate
```

1. Пересобрать образ источника доставки контента и поднять стек:

```
docker compose -f compose.onprem.yml build webdav
docker compose -f compose.onprem.yml up -d
```

1. Проверить работоспособность по п. 3.8.

4.3. Резервное копирование

Сервис `db-backup` начинает работу с первого запуска стека и ежесуточно, в час, заданный параметром `BACKUP_HOUR_UTC` (по умолчанию 01:00 UTC), создает резервную копию базы данных в формате `pg_dump -Fc` , зашифрованную алгоритмом AES-256 с использованием парольной фразы `BACKUP_ENC_PASSPHRASE` . Копии старше `BACKUP_RETENTION_DAYS` суток (по умолчанию 14) удаляются автоматически.

Копии записываются в том Docker `backup-data` , то есть по умолчанию хранятся на том же сервере, что и сама база данных. Регулярный перенос копий на отдельное хранилище и хранение парольной фразы отдельно от копий обеспечиваются службой эксплуатации заказчика.

5. Типовые затруднения при установке

Признак	Вероятная причина	Действия
Установщик сообщает, что образ `123cms-api:latest` не найден	Образы не загружены из архива поставки либо загружены с другими тегами	Выполнить `docker load -i 123cms-images.tar` , проверить теги командой `docker image ls`
Установщик сообщает об ошибке проверки лицензии	Несоответствие открытого ключа файлу лицензии либо повреждение файла при передаче	Запросить у поставщика повторную выдачу файла лицензии и открытого ключа, повторить проверку сценарием `verify_licence.py`

Признак	Вероятная причина	Действия
Сервис `api` не переходит в состояние `healthy`	Ошибка миграции базы данных, недоступность базы данных или Redis, некорректные значения в `.env.onprem`	Просмотреть журнал `docker compose -f compose.onprem.yml logs api` и журнал сервиса `migrate`
Консоль недоступна по адресу `https://<DOMAIN>/app`	Имя хоста не разрешается в адрес сервера; порты 80 или 443 заняты другим приложением или закрыты межсетевым экраном	Проверить разрешение имени, занятость портов и правила межсетевого экрана
Браузер сообщает о недоверенном сертификате	Используется режим `tls internal` — сертификат выпущен встроенным удостоверяющим центром	Принять сертификат либо распространить корневой сертификат прокси-сервера на рабочие места, либо перейти на сертификат собственного удостоверяющего центра
Медиаплеер зарегистрирован, но не выходит на связь	Не заданы или заданы неверно адрес концентратора и параметры доставки контента на самом устройстве; исходящие соединения заблокированы	Сверить значения, выданные мастером подключения плеера в консоли, с настройками устройства; проверить прохождение исходящих соединений по HTTPS

При невозможности устранить затруднение самостоятельно следует обратиться в службу технической поддержки, указав тип установки, версию экземпляра, точное описание выполненных действий и относящиеся к событию фрагменты журналов. Контактные данные приведены в документе «Описание процессов, обеспечивающих поддержание жизненного цикла ПО».