

Общество с ограниченной ответственностью
«Центр Инновационных Технологий и Разработок»
(ООО «ЦИНТЕРА»)

ИНСТРУКЦИЯ ПО УСТАНОВКЕ

«Платформа хранения и распространения медиаконтента Рябина.Медиа»

Версия: 0.4

Дата: 29.07.2025

Москва-2025

ОГЛАВЛЕНИЕ

1	ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ.....	3
2	АННОТАЦИЯ	4
3	ОБЩИЕ СВЕДЕНИЯ.....	5
3.1	Наименование программы	5
3.2	Назначение программы	5
3.3	Область применения	5
3.4	Особенности распространения и применения.....	5
4	ТРЕБОВАНИЯ К ТЕХНИЧЕСКИМ СРЕДСТВАМ	6
5	ТРЕБОВАНИЯ К ОБЩЕМУ ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ	7
6	УРОВЕНЬ ПОДГОТОВКИ ПОЛЬЗОВАТЕЛЕЙ	8
6.1	Требования к квалификации персонала при проверке на стенде.....	8
6.2	Требования к квалификации персонала при штатной установке и эксплуатации	8
7	УСТАНОВКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМЫ	9
7.1	Установка серверного ПО, необходимого для работы системы	9
7.2	Предварительная настройка проекта	10
7.3	Подключение к сайту.....	18

1 Термины и определения

Термин	Определение
ПО (Система)	Программное обеспечение «Платформа хранения и распространения медиаконтента Рябина.Медиа», разработанное компанией ООО «ЦИНТЕРА», предназначенное для обеспечения защищенной и масштабируемой среды для хранения, трансляции, распространения и архивирования медиаконтента корпоративных и публичных мероприятий с возможностью интеграции с корпоративными порталами, системами видеоконференцсвязи, образовательными платформами и другими цифровыми сервисами.
Администратор ПО	Специалист, ответственный за установку, настройку и поддержание работоспособности ПО «Платформа хранения и распространения медиаконтента Рябина.Медиа».

2 Аннотация

Настоящий документ содержит сведения о порядке установки, настройки и проверки функционирования программного обеспечения «Платформа хранения и распространения медиаконтента Рябина.Медиа» (далее – Система).

Документ содержит пошаговую инструкцию по развертыванию экземпляра Системы и предназначен для использования экспертом при развертывании проверочного экземпляра Системы на тестовом стенде.

3 Общие сведения

3.1 Наименование программы

Полное наименование программы: «Платформа хранения и распространения медиаконтента Рябина.Медиа»

Сокращенное наименование программы: Рябина.Медиа. В рамках настоящего документа употребляется также термин «Система».

Программное обеспечение «Рябина.Медиа» – это российское программное обеспечение, организация-разработчик: Общество с ограниченной ответственностью "Центр Инновационных Технологий и Разработок" (ООО «ЦИНТЕРА»).

Сайт организации-разработчика: <https://cintera.ru/>.

Организация-правообладатель: Общество с ограниченной ответственностью "Центр Инновационных Технологий и Разработок" (ООО «ЦИНТЕРА»).

3.2 Назначение программы

ПО «Рябина.Медиа» предназначено для организации защищенной и масштабируемой среды для хранения, трансляции, распространения и архивирования медиаконтента корпоративных и публичных мероприятий с возможностью интеграции с корпоративными порталами, системами видеоконференцсвязи, образовательными платформами и другими цифровыми сервисами.

3.3 Область применения

Область применения Системы: загрузка, хранение, управление и распространение видео с настройкой параметров безопасности, аналитикой и отчетностью

Система реализуется и применяется в виде SaaS веб-приложения.

3.4 Особенности распространения и применения

Система реализована в виде SaaS веб-приложения и обеспечивает взаимодействие с пользователем по технологии «тонкий клиент» с применением распространенных веб-браузеров (Chrome, Safari, Mozilla, Edge, Яндекс.Браузер и др.).

Серверная часть также может быть установлена в контуре клиента (on-prem) и функционирует под управлением операционной системы Ubuntu.

4 Требования к техническим средствам

Для проверки функционирования Системы на стенде применяется технологическая ПЭВМ с характеристиками не хуже:

- процессор: Intel или AMD;
- объем ОЗУ: 4 ГБ;
- сетевой интерфейс: Ethernet, 100 Мб/с.

Технологическая ПЭВМ должна быть подключена к сети Интернет.

5 Требования к общему программному обеспечению

Для установки и проверки функционирования Системы на сервере должно быть установлено следующее общее программное обеспечение:

- OS Ubuntu 22.04 - операционная система на базе Linux.
- система контейнеризации Docker (<https://www.docker.com>), включая инструментальное средство Docker Compose.
- программа управления доступом к репозиторию исходного кода Git
- редактор Nano

6 Уровень подготовки пользователей

6.1 Требования к квалификации персонала при проверке на стенде

Для выполнения установки и проверки функционирования Системы на стенде пользователь должен обладать следующими знаниями и навыками:

- навыки уверенного пользователя персонального компьютера под управлением ОС Linux Ubuntu (версии 22.04 или выше);
- навыки использования веб-браузера типа Google Chrome;
- знание общих принципов функционирования веб-приложений;
- общих принципов функционирования компьютерных сетей;
- навыки использования программного средства Docker.

6.2 Требования к квалификации персонала при штатной установке и эксплуатации

Установка Системы в процессе основного жизненного цикла выполняется сотрудниками организации-разработчика. Конечными пользователями установка не выполняется.

Эксплуатация выполняется конечными пользователями, которые должны обладать следующими знаниями и навыками:

- навыки работы на персональном компьютере;
- навыки работы с веб-приложениями с использованием веб-браузера.

7 Установка программного обеспечения системы

7.1 Установка серверного ПО, необходимого для работы системы

Для работы проекта требуется **Docker** окружение и **Docker Compose plugin**, так как проект работает на контейнерах, для клонирования исходного кода проекта, потребуется программа **git**, а для настройки конфигурации нужен текстовый редактор **nano**.

7.1.1 Установка docker

Перед установкой, выполните команду **docker -v**

Если эта команда выполнена успешно, и вы увидели строку с версией Docker - например *Docker version 27.4.0, build bde2b89*, то docker установлен и дальнейшие шаги по установке Docker не требуются.

Если Docker не установлен, для установки необходимо выполнить установку в соответствии с инструкцией <https://docs.docker.com/engine/install/ubuntu/>.

Для проверки установки Docker необходимо выполнить команду «**docker info**». Если эта команда выполнена успешно, значит установка завершена успешно. В выводе команды удостовериться в установке плагина Docker Compose.

7.1.2 Установка git

Git – это программа контроля версий программного обеспечения.

```
#Обновляем информацию о доступных пакетах
sudo apt-get update
#Устанавливаем Git
sudo apt-get install -y git
#Проверка установки Git
git --version
```

7.1.3 Установка nano

```
#Обновляем информацию о доступных пакетах
sudo apt-get update
#Устанавливаем Nano
sudo apt-get install -y nano
```

7.2 Предварительная настройка проекта

В процессе настройки проекта потребуется:

- Клонировать несколько Git репозиториях;
- Настроить переменные окружения;
- Запустить окружение docker контейнеров;
- Произвести первичную настройку базы данных и проекта в целом.

7.2.1 Предварительная настройка проекта

Для клонирования репозиториях проектов, у вас должна быть учетная запись на сервере репозиториях компании разработчика, клонирование возможно по https протоколу с вводом логина и пароля.

Первый репозиторий – это docker окружение с letsencrypt и reverse-proxy контейнерами.

Контейнером letsencrypt будут автоматически выдаваться и продляться SSL сертификаты доменов.

Если необходимость в бесплатных сертификатах и дополнительных окружения отсутствует, то этот шаг можно пропустить, далее будет описан процесс настройки своих сертификатов на обычном nginx контейнере.

Создайте на сервере директорию, где будут храниться все файлы проекта, например директорию /var/docker.

Перейдите в директорию /var/docker и клонируйте репозиторий с letsencrypt и reverse-proxy контейнерами.

```
# Переход в директорию
cd /var/docker
# Клонирование репозитория
git clone https://gitlab.cintera.ru/docker/general.git
```

После ввода команды клонирования потребуется ввести логин и пароль учетной записи предоставленные компанией разработчиком.

После клонирования репозитория, автоматически будет создана директория `general` в которой будут развернуты файлы проекта.

Категорически не рекомендуется менять название этой директории, иначе возникнут проблемы с развертыванием проекта, так как при развертывании текущих контейнеров, `docker` создаст внутреннюю сеть, в которой будет фигурировать название директории, если поменять название, то придется менять название сети во всех других репозиториях контейнеров.

Далее нужно перейти в директорию с проектом `general`, создать файл переменных окружения и запустить создание контейнеров.

```
# Переход в директорию с проектов general
cd /var/docker/general

# Создание файла переменных окружения
cp .env.example .env

# Запуск проекта
docker compose up -d
```

В результате должны будут подняться контейнеры `reverse-proxy` и `letsencrypt`.

Убедитесь в этом выполнив команду:

```
docker ps
```

Теперь можно приступить к развертыванию основных контейнеров проекта.

Второй репозиторий – это окружение контейнеров проекта, таких как база данных, интерпретатор языка PHP и другие.

Это основные контейнеры, без которых приложение не сможет работать.

Перейдите в директорию `/var/docker` и клонируйте репозиторий.

```
# Переход в директорию
cd /var/docker

# Клонирование репозитория
git clone https://gitlab.cintera.ru/docker/production.git
```

Далее:

- Перейти в директорию с проектом `general`;
- Создать файл переменных окружения;
- Указать значения переменных;
- Создать файл `nginx` конфигурации из шаблона;
- Указать домен в конфигурации `nginx`.

Файла конфигурации nginx: example.conf – для развертывания проекта с окружением letsencrypt и с reverse-proxy;

При редактировании файла .env - переменных окружения, потребуется задать название проекта, которое будет использоваться в именах контейнеров и указать домены или список доменов, которые должны ссылаться на это окружение.

Вы также можете изменить значения переменных, например ВАЖНО указать сложный пароль к базе данных.

```
#----- General Configuration -----#
PROJECT_NAME=MyProject

# Список доменов через запятую без пробелов, например www.domain.com,domain.com
HOST_LIST=domain.com
HOST_EMAIL=support@domain.com
TZ=Europe/Moscow

#----- MySQL Configuration -----#
MYSQL_ROOT_PASSWORD=123
MYSQL_USER=oko
MYSQL_DATABASE=db_oko
MYSQL_PASSWORD=123
MYSQL_PORT=33061

#----- Redis Configuration -----#
REDIS_PASSWORD=123
```

При редактировании файла nginx – файла конфигурации, потребуется задать название домена в секции server_name

```
server {
    listen 80;
    listen [::]:80;
    # Здесь в server_name нужно указать адрес домена, например domain.com
    server_name domain.com;
```

Пример списка команд

```
# Переход в директорию с проектом production окружения
cd /var/docker/production
# Создание файла переменных окружения
cp .env.example .env
# Редактирование файла
nano .env
# Создание конфигурационного файла nginx из шаблона
cp sources/production/nginx/conf.d/example.conf sources/production/nginx/conf.d/domain.c
onf
# Редактирование файла конфигурации nginx
```

```
nano sources/production/nginx/conf.d/domain.conf
```

Далее необходимо запустить проект.

Выполните команду:

```
docker compose up -d
```

Начнется сборка контейнеров, которая займет порядка 5 минут.

После чего должны быть созданы контейнеры проекта которые будут называться по названию проекта, указанному в .env файле (см рис. 1):

- nginx;
- php;
- db;
- redis;
- queue_worker.

ВАЖНО! Контейнер queue_worker в данный момент завершит выполнение с ошибкой, так как пока что отсутствуют библиотеки и файлы проекта, он будет запущен на следующем шаге.

```

root@server-e2qe7c: /var/doc
=> => extracting sha256:38e992d287c563d332ed8c9bdefec32a84f50f94cc39bb4729a3e73c 0.0s
=> => extracting sha256:9e9aab598f58e86706b73c23e0ff33fc4eb60a5e69424147f9cb7c02 0.0s
=> => extracting sha256:4de87b37f4ad0b499be3db4cc6831db3d8158867f12dabc0bf42046b 0.0s
=> [nginx] exporting to image 0.1s
=> => exporting layers 0.0s
=> => writing image sha256:2eac809490caa0ef35bc29694276503e68b71a6a0b0a73998ed9f 0.0s
=> => naming to docker.io/library/tube-nginx:v2.0 0.0s
[+] Running 6/6
✔Network project_default Created 0.2s
✔Volume "project_db_data" Created 0.0s
✔Container prj-tube_redis_1 Started 0.3s
✔Container prj-tube_db_1 Started 0.3s
✔Container prj-tube_php_1 Started 0.1s
✔Container prj-tube_nginx_1 Started 0.1s
root@server-e2qe7c: /var/docker/project#

```

Рисунок 1 – Визуализация завершения загрузки образов docker контейнеров

Третий репозиторий – это непосредственно сам код приложения.

Требуется выполнить следующие команды:

```

# Перейдите в директорию с проектом production в директорию app
cd /var/docker/production/app
# Клонирование репозитория
https://gitlab.cintera.ru/newtube/cintera.git okocrm

```

Инициализация проекта состоит из выполнения нескольких консольных команд и настройки файла конфигурации.

Для инициализации SSH подключение, которое потребуется для загрузки библиотек проекта, выполните команду:

```
docker exec -it prj-tube_php_1 ssh -T git@gitlab.cintera.ru
```

Далее, будет выведен запрос, действительно ли вы хотите подключиться к git репозиторию, введите «yes» и нажмите «Enter».

После этого необходимо выполнить команду установки библиотек проекта

```
docker exec -it prj-tube_php_1 composer install --prefer-dist -vvv
```

Будет выведен запрос, действительно ли вы хотите подключиться к git репозиторию, введите «yes» и нажмите «Enter».

Процесс загрузки библиотек не слишком быстрый, может занять до 10 минут, дождитесь завершения загрузки.

7.2.2 Настройка конфигурации проекта

Первичная настройки конфигурации проекта, это создание аккаунта администратора, установка прав доступа на директории проекта и установка плагинов.

```
#Установка прав доступа на директорию с временными файлами, вроде логов и  
кэша  
sudo chmod 777 -R app/okocrm/runtime/
```

Следующая команда, создает конфигурационные файлы проекта.

```
docker exec -it prj-tube_php_1 php init
```

После ввода команды будет задан вопрос о том, какое окружение необходимо инициализировать.

На этом шаге, введите цифру «0» и нажмите Enter, а затем введите «yes» для подтверждения и нажмите «Enter».

В результате выполнения команды, будут созданы конфигурационные файлы проекта, как это показано на рисунке.

Далее система запросит url адрес проекта, адрес панели администратора и путь до статических файлов проекта.

Введите адрес вашего сайта вместе с https, например: <https://edu.ryabina.media/>,

затем для панели администратора: <https://edu.ryabina.media/backend>

и для директории статических файлов: <https://edu.ryabina.media/static/>

Следующие шаги, это указание типа базы данных, просто нажмите «Enter» чтобы по умолчанию установилось значение «mysql».

Следующий шаг, это указание хоста базы данных, нажмите «Enter» чтобы по умолчанию установилось значение «db».

Следующий шаг, это указание порта доступа для базы данных, нажмите «Enter» чтобы по умолчанию установилось пустое значение «db».

Следующие шаги, где требуется ввести:

- Название базы данных (database name);
- Имя пользователя базы данных (username);
- Пароль пользователя (password);

Здесь вводятся те же значения, которые указывали при редактировании файла «db.env» в разделе 7.2.1.

Далее вводим команду инициализации плагинов проекта.

```
docker exec -it prj-tube_php_1 php yii plugin/install-all
```

После ввода команды начнется установка плагинов

```

root@server-e2qe7c: /var/docker/project/app/okocrm#
Done.
Updating bttree/yupe2-video
> create table {{bttree_video_category}} ... done (time: 0.013s)
> create table {{bttree_video}} ... done (time: 0.012s)
> create table {{bttree_video_categories}} ... done (time: 0.011s)
> create table {{bttree_video_tag}} ... done (time: 0.016s)
> create table {{bttree_video_tags}} ... done (time: 0.014s)
> add primary key tag_video_pk on bttree_video_tags (tag_id,video_id) ... done (time: 0.019s)
> add primary key category_video_pk on bttree_video_categories (category_id,video_id) ... done (time: 0.021s)
> add foreign key fk_bttree_video_category_parent_id: {{bttree_video_category}} (parent_id) references {{bttree_video_category}} (id) ... done (time: 0.022s)
> add foreign key fk_bttree_video_category_id: {{bttree_video}} (category_id) references {{bttree_video_category}} (id) ... done (time: 0.033s)
> add foreign key fk_bttree_video_author: {{bttree_video}} (author_id) references {{bttree_user_user}} (id) ... done (time: 0.029s)
> add foreign key fk_bttree_video_created_by: {{bttree_video}} (created_by) references {{bttree_user_user}} (id) ... done (time: 0.034s)
> add foreign key fk_bttree_video_updated_by: {{bttree_video}} (updated_by) references {{bttree_user_user}} (id) ... done (time: 0.031s)
> new role 'bttree_video.manage-all' added.
> new role 'bttree_video.manage-category' added.
> new role 'bttree_video.manage-tag' added.
> create table {{bttree_video_image}} ... done (time: 0.012s)
> add foreign key fk_bttree_video_image_video: {{bttree_video_image}} (video_id) references {{bttree_video}} (id) ... done (time: 0.027s)
> create table {{bttree_video_comment}} ... done (time: 0.013s)
> add foreign key fk_bttree_comment_video: {{bttree_video_comment}} (video_id) references {{bttree_video}} (id) ... done (time: 0.031s)
> add foreign key fk_bttree_comment_video_created_by: {{bttree_video_comment}} (created_by) references {{bttree_user_user}} (id) ... done (time: 0.045s)
> add foreign key fk_bttree_comment_video_updated_by: {{bttree_video_comment}} (updated_by) references {{bttree_user_user}} (id) ... done (time: 0.040s)
> add column recommended boolean DEFAULT FALSE to table {{bttree_video}} ... done (time: 0.039s)
> create table {{bttree_video_like}} ... done (time: 0.033s)
> add column cnt integer DEFAULT 0 to table {{bttree_video}} ... done (time: 0.030s)
> create table {{bttree_video_history}} ... done (time: 0.009s)
> create table {{bttree_video_favourite}} ... done (time: 0.009s)
> add column id pk to table {{bttree_video_history}} ... done (time: 0.020s)
> add column created_at timestamp DEFAULT CURRENT_TIMESTAMP to table {{bttree_video_history}} ... done (time: 0.003s)
> add column ip string(255) NULL DEFAULT NULL to table {{bttree_video_history}} ... done (time: 0.027s)
> add column browser string(255) NULL DEFAULT NULL to table {{bttree_video_history}} ... done (time: 0.021s)
> add column referral string(255) NULL DEFAULT NULL to table {{bttree_video_history}} ... done (time: 0.016s)
> create table {{bttree_video_like}} ... done (time: 0.017s)
> add column like integer DEFAULT 0 to table {{bttree_video}} ... done (time: 0.023s)
> add column dislike integer DEFAULT 0 to table {{bttree_video}} ... done (time: 0.023s)
> add column duration integer DEFAULT 0 to table {{bttree_video}} ... done (time: 0.025s)
Done.
root@server-e2qe7c: /var/docker/project/app/okocrm#

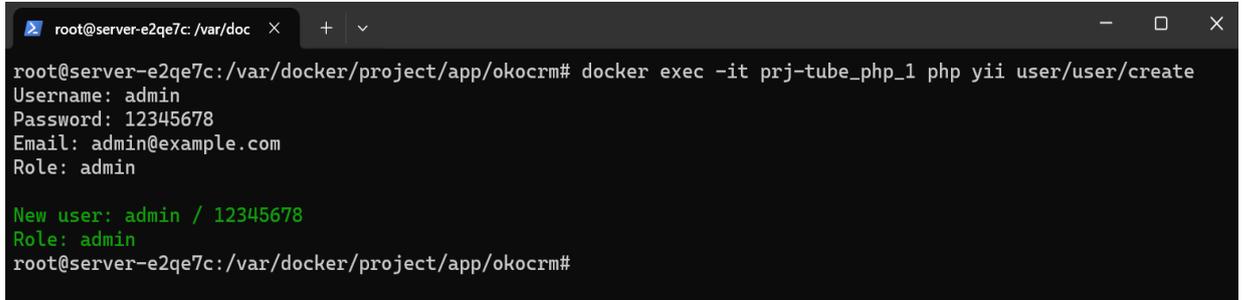
```

Рисунок 2 – Установка плагинов

Следующим шагом создадим администратора, для этого выполните команду

```
docker exec -it prj-tube_php_1 php yii user/user/create
```

Это команда создаст учетную запись главного администратора проекта, с полными правами доступа.



```
root@server-e2qe7c: /var/doc x + v
root@server-e2qe7c:/var/docker/project/app/okocrm# docker exec -it prj-tube_php_1 php yii user/user/create
Username: admin
Password: 12345678
Email: admin@example.com
Role: admin

New user: admin / 12345678
Role: admin
root@server-e2qe7c:/var/docker/project/app/okocrm#
```

Рисунок 3 – Создание учетной записи главного администратора

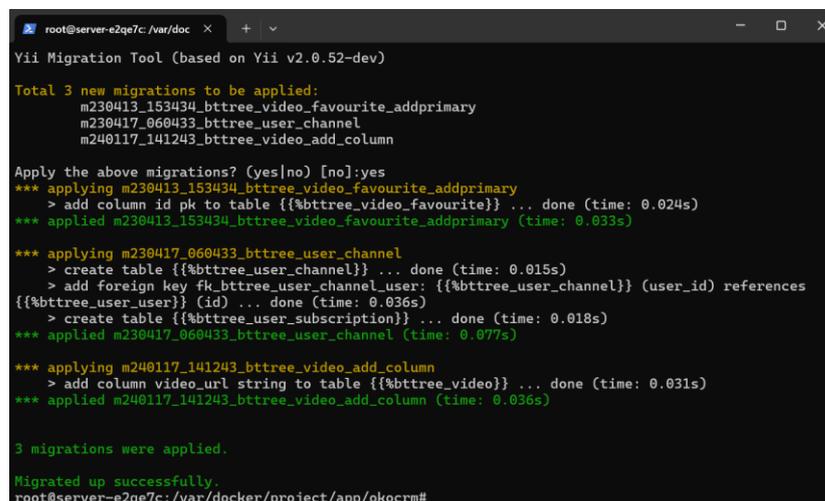
При создании учетной записи администратора, указывайте сложный пароль длиной не менее 8 символов.

Роль обязательно указывайте «admin».

Теперь остается применить права доступа к файлам конфигурации проекта и применить миграции, для этого выполните следующие команды:

```
#Применение прав доступа на файлы и директории
sudo chmod 777 -R /var/docker/project/app/okocrm/configs/plugins-local.php
sudo chmod 777 -R /var/docker/project/app/okocrm/web/static/assets
#Применение миграций проекта
docker exec -it prj-tube_php_1 php yii migrate
```

После ввода команды применения миграций, будет задан вопрос, действительно ли применить миграции, введите «yes» и нажмите «Enter».



```
root@server-e2qe7c: /var/doc x + v
Yii Migration Tool (based on Yii v2.0.52-dev)

Total 3 new migrations to be applied:
  m230413_153434_bttree_video_favourite_addprimary
  m230417_060433_bttree_user_channel
  m240117_141243_bttree_video_add_column

Apply the above migrations? (yes|no) [no]:yes
*** applying m230413_153434_bttree_video_favourite_addprimary
  > add column id pk to table {{%bttree_video_favourite}} ... done (time: 0.024s)
*** applied m230413_153434_bttree_video_favourite_addprimary (time: 0.033s)

*** applying m230417_060433_bttree_user_channel
  > create table {{%bttree_user_channel}} ... done (time: 0.015s)
  > add foreign key fk_bttree_user_channel_user: {{%bttree_user_channel}} (user_id) references
  {{%bttree_user_user}} (id) ... done (time: 0.036s)
  > create table {{%bttree_user_subscription}} ... done (time: 0.018s)
*** applied m230417_060433_bttree_user_channel (time: 0.077s)

*** applying m240117_141243_bttree_video_add_column
  > add column video_url string to table {{%bttree_video}} ... done (time: 0.031s)
*** applied m240117_141243_bttree_video_add_column (time: 0.036s)

3 migrations were applied.

Migrated up successfully.
root@server-e2qe7c:/var/docker/project/app/okocrm#
```

Рисунок 4 – Применение миграций

Если все пройдет корректно и миграции будут применены, то должна появиться надпись «Migrated up successfully».

На этом установку и запуск проекта можно считать завершенным.

7.3 Подключение к сайту

Подключиться к проверочному экземпляру Системы можно, используя следующие точки доступа:

- – для веб-сайта: пример - <https://stage.ryabina.media>
- - для панели администратора: пример - <https://stage.ryabina.media/backend>

Шаг 1. Откройте веб-браузер и укажите в адресной строке следующий URL-адрес:

<https://stage.ryabina.media/backend>

При выполнении этого действия происходит запуск страницы логина в соответствии с рисунком 5.

Рисунок. 5. Страница логина

Шаг 2. Введите имя и пароль ниже и нажмите кнопку «ВОЙТИ».

Логин: **[имя пользователя]**

Пароль: **[пароль пользователя]**

После выполнения входа в Систему откроется страница со сводными сведениями в соответствии с рисунком 6. Система готова к функционированию в соответствии с инструкцией пользователя.

ID	Изображение	Название видео	Категория	Главная	Статус	Создана
176		Итоги: II Форум негосударственного образования	Для Прессы	Да	Опубликовано	24 апр. 2025 г., 18:30:14
177		Лемана Про_Запись диспетчера Schneider Electric	О проекте	Нет	Черновик	9 мая 2025 г., 18:05:14

Рисунок. 6. Главная страница административной части портала пользователя