



Инструкция по установке и запуску программного обеспечения „Маяк“

ПРОГРАММА ДЛЯ ЭВМ

Система управления производством «Маяк»

Версия v1.0.0

Настоящая инструкция устанавливает порядок установки, настройки и запуска экземпляра программного обеспечения «Система управления производством «Маяк» (далее - Программное обеспечение, ПО).

Программное обеспечение представляет собой веб-приложение, доступ к которому осуществляется посредством браузера без необходимости установки дополнительного клиентского приложения на рабочее место пользователя.

Развертывание серверной части Программного обеспечения осуществляется в контейнеризированной среде одним из следующих способов:

на отдельном сервере с использованием Docker Compose;

в кластере Kubernetes с использованием Helm.

Облачная версия Программного обеспечения, эксплуатируемая правообладателем, развертывается с использованием Kubernetes и Helm.

В настоящей инструкции приведены требования к программному окружению, последовательность установки и запуска Программного обеспечения, а также описание необходимых параметров конфигурации.

01 Требования к программному и аппаратному обеспечению

Компонент	Требование
ОС сервера	64-разрядная операционная система Linux с поддержкой Docker, в том числе Astra Linux Special Edition 1.8.6 и РЕД ОС 8.0.3.
Программное окружение	Docker Engine и Docker Compose версии 2.24 или новее (нужна поддержка !reset); для варианта Б — кластер Kubernetes и Helm 3
Оперативная память	не менее 8 ГБ (лимит PHP — до 6 ГБ на процесс при разборе больших файлов)
СУБД	PostgreSQL 16 (входит в поставку Compose) или внешний сервер PostgreSQL
Файловое хранилище	S3-совместимое хранилище (в облачной версии — Yandex Object Storage)
Почта	SMTP-сервер для отправки кодов входа
Рабочее место пользователя	современный браузер (Яндекс Браузер, Chromium, Firefox) в любой ОС

02 Установка на отдельный сервер с использованием Docker Compose

- 01 Установите Docker Engine и плагин Docker Compose из репозитория ОС и проверьте версию командой `docker compose version` (Рисунок 1).

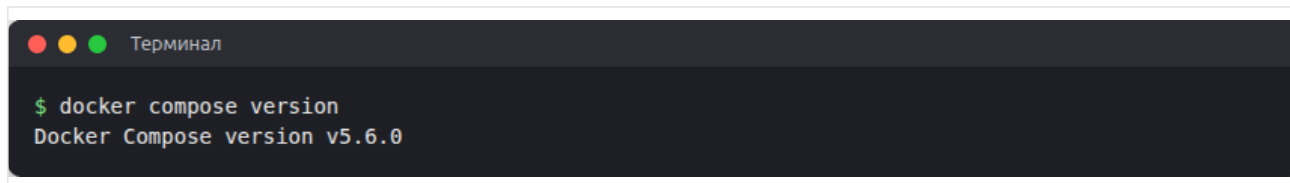


Рисунок 1 — Проверка версии Docker Compose

- 02 Получите дистрибутив у правообладателя, распакуйте его в рабочий каталог, например `/opt/майак`, и перейдите в него (Рисунок 2). Все дальнейшие команды выполняются из этого каталога.

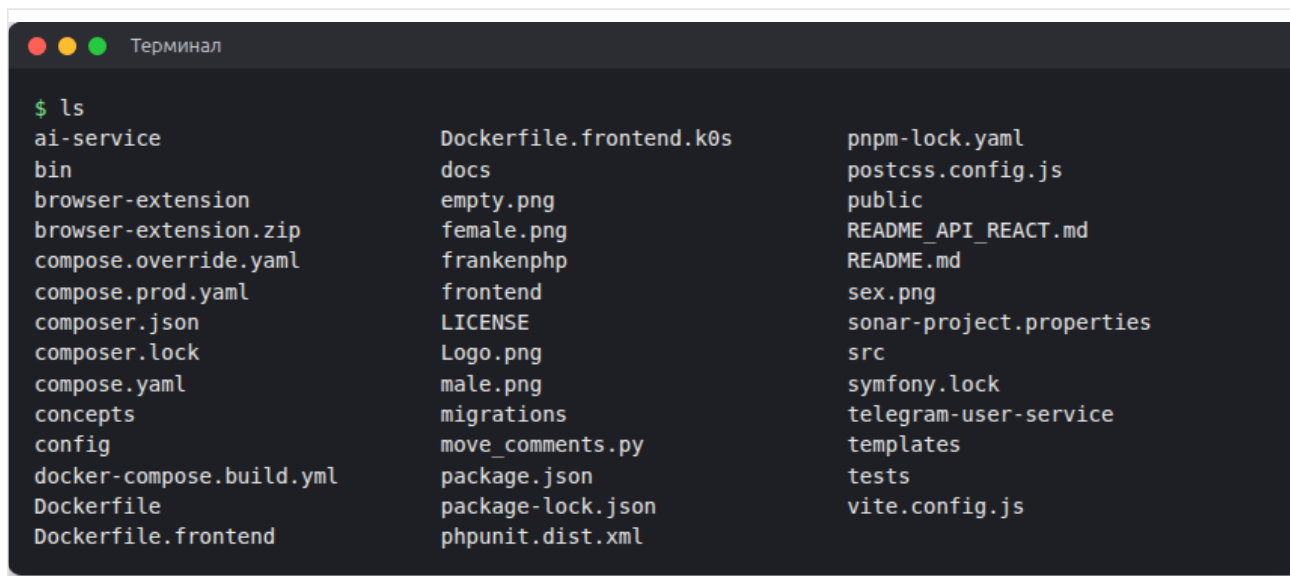
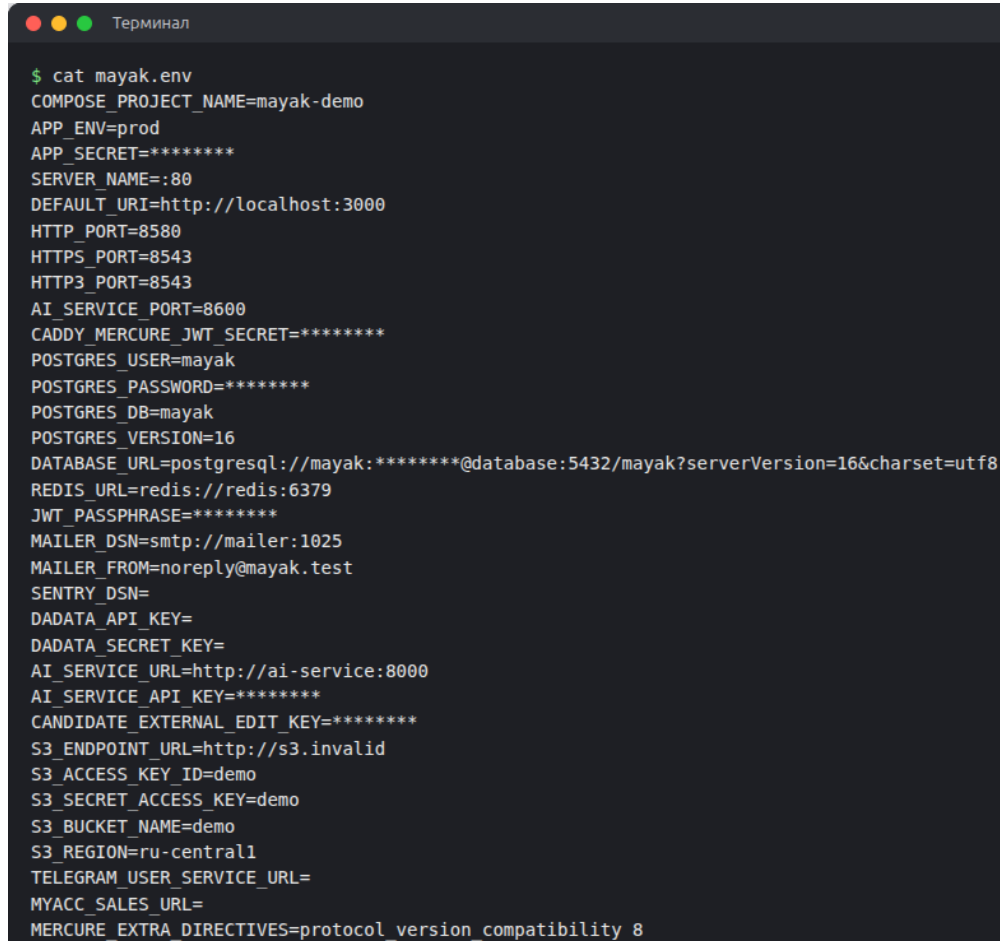


Рисунок 2 — Содержимое каталога дистрибутива

- 03 Создайте файл `майак.env` с настройками экземпляра (Рисунок 3, секреты скрыты). Этот файл читают и Docker Compose, и приложение; переменные описаны в таблице после шагов.



```
$ cat mayak.env
COMPOSE_PROJECT_NAME=mayak-demo
APP_ENV=prod
APP_SECRET=*****
SERVER_NAME=:80
DEFAULT_URI=http://localhost:3000
HTTP_PORT=8580
HTTPS_PORT=8543
HTTP3_PORT=8543
AI_SERVICE_PORT=8600
CADDY_MERCURE_JWT_SECRET=*****
POSTGRES_USER=mayak
POSTGRES_PASSWORD=*****
POSTGRES_DB=mayak
POSTGRES_VERSION=16
DATABASE_URL=postgresql://mayak:*****@database:5432/mayak?serverVersion=16&charset=utf8
REDIS_URL=redis://redis:6379
JWT_PASSPHRASE=*****
MAILER_DSN=smtp://mailer:1025
MAILER_FROM=noreply@mayak.test
SENTRY_DSN=
DADATA_API_KEY=
DADATA_SECRET_KEY=
AI_SERVICE_URL=http://ai-service:8000
AI_SERVICE_API_KEY=*****
CANDIDATE_EXTERNAL_EDIT_KEY=*****
S3_ENDPOINT_URL=http://s3.invalid
S3_ACCESS_KEY_ID=demo
S3_SECRET_ACCESS_KEY=demo
S3_BUCKET_NAME=demo
S3_REGION=ru-central1
TELEGRAM_USER_SERVICE_URL=
MYACC_SALES_URL=
MERCURE_EXTRA_DIRECTIVES=protocol_version_compatibility 8
```

Рисунок 3 — Файл настроек mayak.env (тестовый стенд)

04 Создайте файл ai-service/prod.env с настройками AI-сервиса: API_KEY (совпадает с AI_SERVICE_API_KEY), YANDEX_API_KEY, YANDEX_FOLDER_ID, YANDEX_BASE_URL, MODEL_PATH. Без ключей Yandex Cloud система работает, но AI-функции недоступны.

05 Создайте файл compose.site.yaml. Он подключает настройки к контейнерам, собирает итоговый образ веб-сервера с интерфейсом и закрывает доступ к БД и Redis извне сервера:

```
services:
  php:
    env_file: ./mayak.env
  ai-service:
    env_file: ./ai-service/prod.env
  frontend:
    build:
      target: !reset null
  database:
    ports: !reset []
  redis:
    ports: !reset []
```

06 Соберите и запустите приложение; первая сборка занимает несколько минут (Рисунок 4):

```
docker compose --env-file mayak.env -f compose.yaml -f compose.prod.yaml -f
compose.site.yaml up -d --build --wait
```

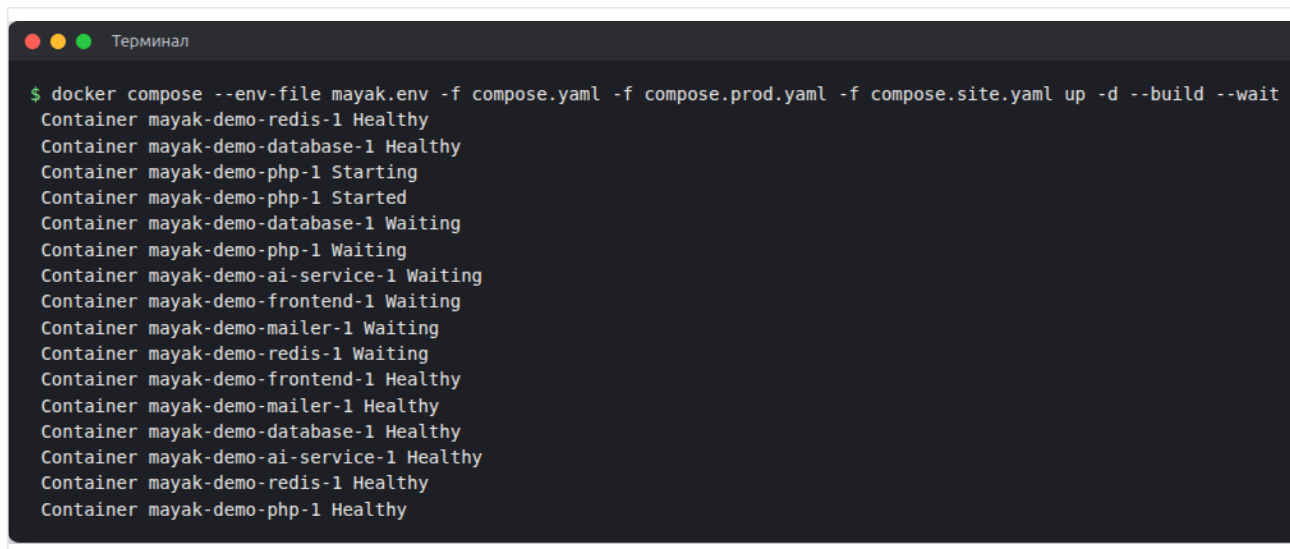


Рисунок 4 — Сборка и запуск контейнеров

07 Проверьте, что контейнеры php, frontend, ai-service, database и redis запущены (Рисунок 5). При первом запуске приложение само создаёт структуру БД (180 миграций) и ключи подписи токенов.

```
docker compose --env-file mayak.env -f compose.yaml -f compose.prod.yaml -f
compose.site.yaml ps --format 'table {{.Service}}\t{{.Status}}'
```

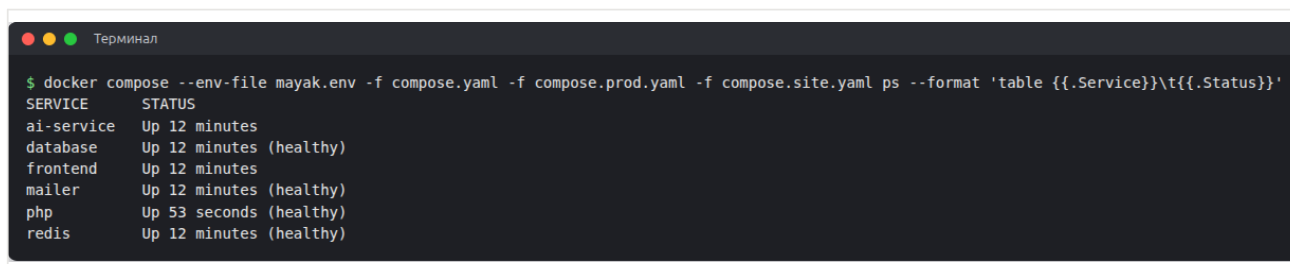
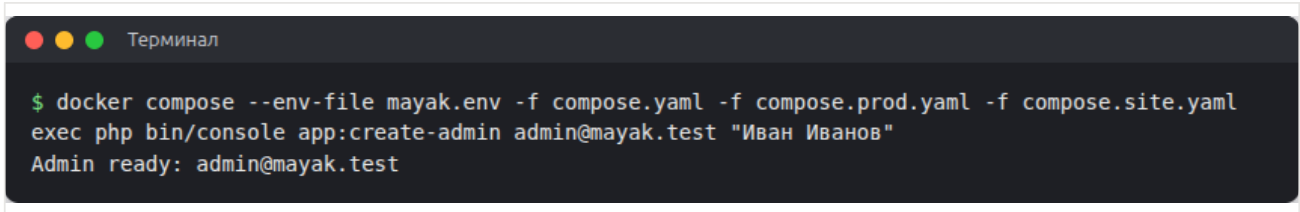


Рисунок 5 — Состояние контейнеров (mailer — почтовый сервер тестового стенда)

08 Создайте первого администратора, указав его e-mail и имя (Рисунок 6):

```
docker compose --env-file mayak.env -f compose.yaml -f compose.prod.yaml -f
compose.site.yaml exec php bin/console app:create-admin admin@company.ru "Иван
Иванов"
```



```
Терминал
$ docker compose --env-file mayak.env -f compose.yaml -f compose.prod.yaml -f compose.site.yaml
exec php bin/console app:create-admin admin@mayak.test "Иван Иванов"
Admin ready: admin@mayak.test
```

Рисунок 6 — Создание администратора

- 09** Откройте в браузере адрес <http://<адрес сервера>:3000> — появится страница входа (Рисунок 7). Для работы по HTTPS поставьте перед приложением веб-сервер с TLS-сертификатом.

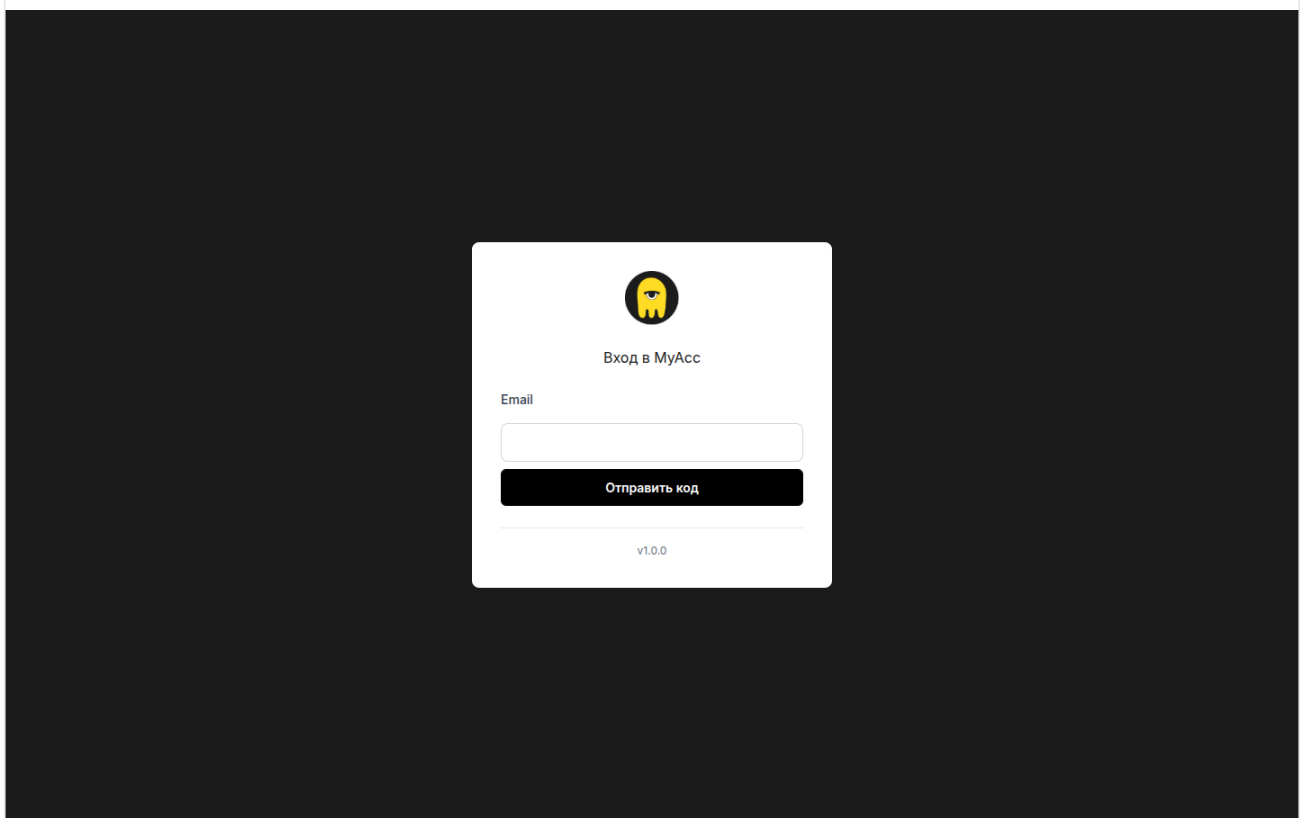


Рисунок 7 — Страница входа

- 10** Введите e-mail администратора и нажмите «Отправить код». Шестизначный код придёт на почту и действует 10 минут (Рисунок 8).

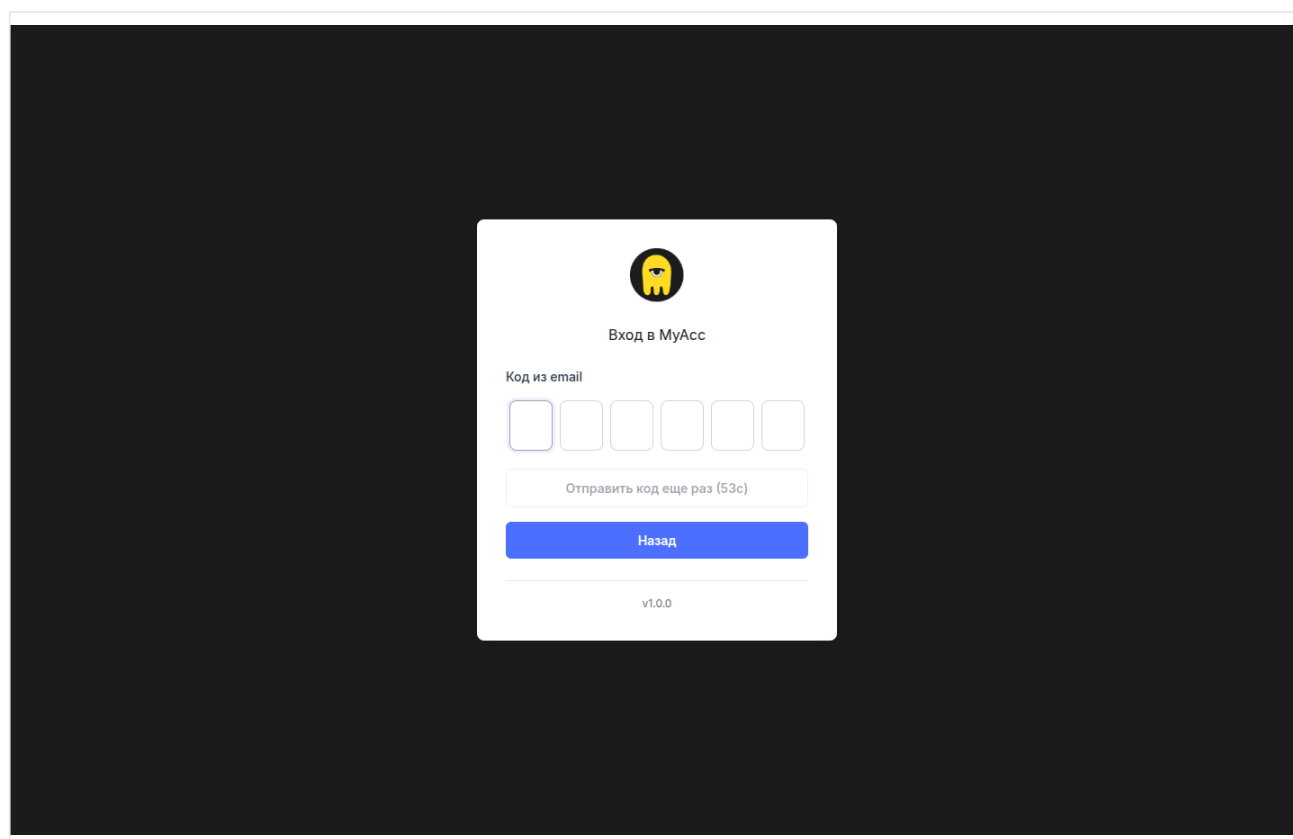


Рисунок 8 — Ввод кода из письма

- 11** Введите полученный код подтверждения. После успешной авторизации откроется главная страница Программного обеспечения (Рисунок 9). Установка и первоначальный запуск Программного обеспечения завершены.

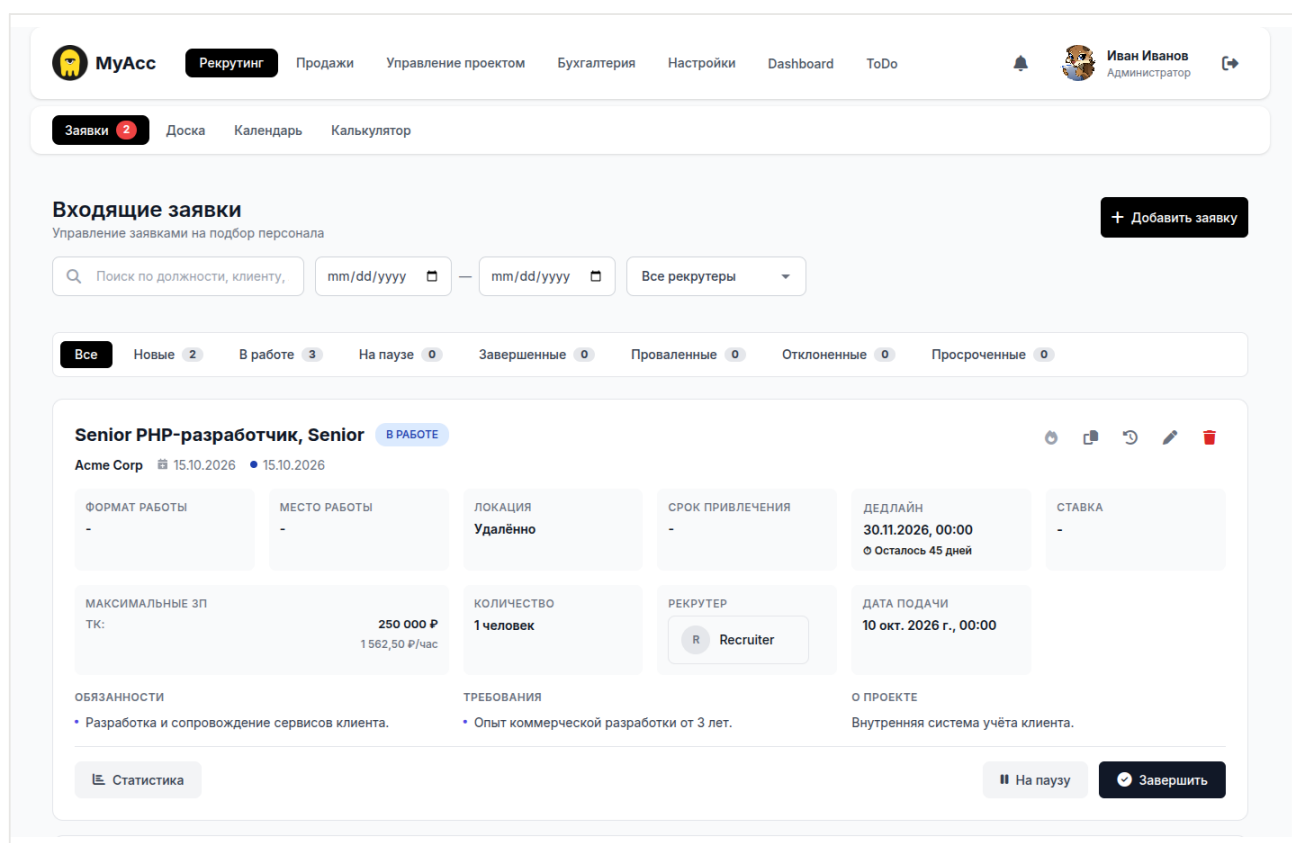


Рисунок 9 — Главная страница после входа

- 12 При необходимости включите интеграции в разделе «Настройки» → «Интеграции» и добавьте пользователей (Рисунок 10).

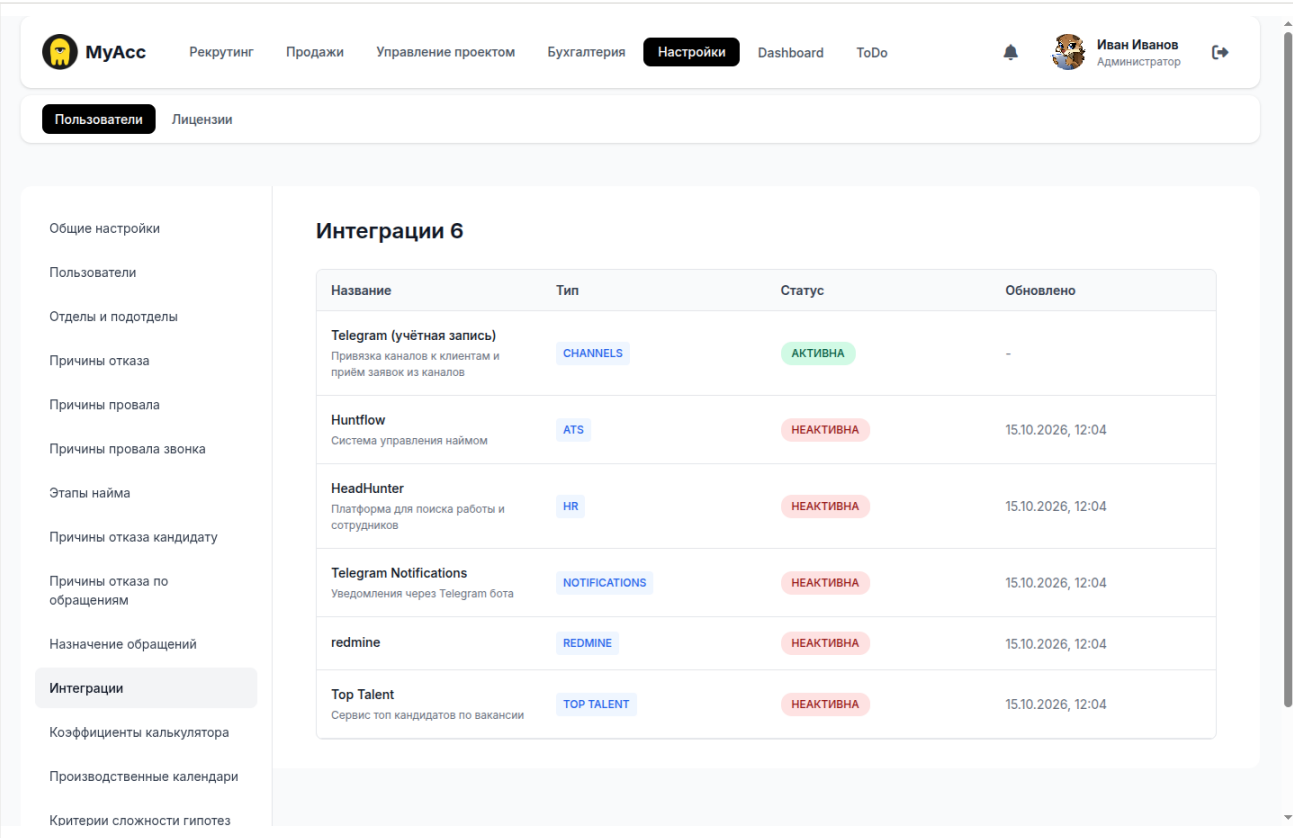


Рисунок 10 — Раздел «Интеграции»

2.1. Параметры конфигурации программного обеспечения

Для настройки Программного обеспечения используется файл `mayak.env`, содержащий переменные окружения. В таблице ниже приведены наименования переменных, их назначение и сведения об обязательности заполнения.

Переменная	Назначение	Обязательная
COMPOSE_PROJECT_NAME	имя проекта Docker Compose, например <code>mayak</code>	да
APP_ENV	режим работы, всегда <code>prod</code>	да
APP_SECRET, JWT_PASSPHRASE, CADDY_MERCURE_JWT_SECRET	случайные строки: секрет приложения, пароль ключей токенов, ключ Mercure	да
SERVER_NAME	всегда <code>:80</code> : по этому адресу веб-сервер интерфейса обращается к серверу приложения	да
MERCURE_EXTRA_DIRECTIVES	<code>protocol_version_compatibility 8</code> — режим совместимости встроенного сервера Mercure	да

Переменная	Назначение	Обязательная
POSTGRES_USER, POSTGRES_PASSWORD, POSTGRES_DB, DATABASE_URL	встроенная БД; DATABASE_URL указывает на сервис database или внешний сервер	да
REDIS_URL	redis://redis:6379	да
MAILER_DSN, MAILER_FROM	SMTP-сервер и адрес отправителя писем	да
S3_ENDPOINT_URL, S3_REGION, S3_BUCKET_NAME, S3_ACCESS_KEY_ID, S3_SECRET_ACCESS_KEY	файловое хранилище	да
AI_SERVICE_URL, AI_SERVICE_API_KEY	адрес (http://ai-service:8000) и ключ AI-сервиса	да
CANDIDATE_EXTERNAL_EDIT_KEY	ключ шифрования ссылок для кандидатов, не короче 32 символов	да
HTTP_PORT, HTTPS_PORT, HTTP3_PORT, AI_SERVICE_PORT	порты на сервере, если стандартные заняты	нет
DADATA_API_KEY, DADATA_SECRET_KEY, SENTRY_DSN, MYACC_SALES_URL, MYACC_SALES_SSO_SECRET	интеграции; пустое значение отключает интеграцию	нет

03 Установка в Kubernetes с использованием Helm

01 Загрузите Docker-образы recruiting-crm, recruiting-nginx и recruiting-ai-service в реестр образов, доступный кластеру.

02 Подготовьте внешний сервер PostgreSQL и файл настроек values с параметрами подключения к БД (env.pg_*), почты, Redis, домена и ресурсов подов.

03 Выполните установку чарта:

```
helm upgrade --install mayak ../helm --namespace mayak --create-namespace -f values.yaml
--set image.tag=<версия> --wait
```

04 Чарт сам создаёт ключи подписи токенов, применяет миграции БД, запускает под приложения, AI-сервис, Redis и плановые задачи, а также выпускает TLS-сертификат для домена через cert-manager.

05 Создайте администратора:

```
kubectl -n mayak exec deploy/mayak-recruting-crm -c php-crm -- bin/console  
app:create-admin admin@company.ru "Иван Иванов"
```

06 Откройте в веб-браузере доменное имя развернутого экземпляра Программного обеспечения и выполните вход в систему в порядке, описанном в шагах 10–11 раздела 2 настоящей инструкции.

Имя mayak для релиза и пространства имён — пример; deployment получает имя <имя релиза>-recruting-crm.