

УТВЕРЖДЕН

RU.1107746574308.069-01 99 01-ЛУ

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
«Платформа разработки программного обеспечения БЗ»**

Инструкция по установке и проверке функционирования

RU.1107746574308.069-01 99 01

Листов 17

Инов. № подл.	Подп. и дата	Взам. инв. №	Инов. № дубл.	Подп. и дата

2023

АННОТАЦИЯ

Документ содержит сведения о порядке скачивания, установки, настройки и проверки функционирования программного обеспечения «Платформа разработки программного обеспечения БЗ» RU.1107746574308.069-01 (далее – Платформа).

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

СОДЕРЖАНИЕ

1. Общие сведения	4
1.1. Наименование программы.....	4
1.2. Назначение программы.....	4
1.3. Особенности распространения и применения.....	4
2. Требования к техническим средствам	5
3. Требования к общему программному обеспечению	6
4. Уровень подготовки пользователей	7
4.1. Требования к квалификации персонала при проверке Платформы.....	7
4.2. Требования к квалификации персонала при штатной установке и эксплуатации	7
5. Развертывание и установка	8
5.1. Развертывание серверной части Платформы на Linux.....	8
5.1.1. Подготовительные работы	8
5.1.2. Установка Docker и настройка контейнеров	8
5.1.3. Установка локальной версии Django.....	9
5.1.4. Выполнение миграций и проверка работоспособности	10
5.2. Развертывание фронтенд-части Платформы	10
5.3. Развертывание на продуктивном стенде.....	11
5.3.1. Плейбуки проектов.....	11
5.3.2. Развертывание.....	14
6. Запуск	15
7. Проверка функционирования	16

1. ОБЩИЕ СВЕДЕНИЯ

1.1. Наименование программы

Полное наименование программы: Платформа разработки программного обеспечения БЗ RU.1107746574308.069-01.

Сокращенное наименование программы: Платформа БЗ. В рамках настоящего документа употребляется также термин «Платформа».

Обозначение программы: RU.1107746574308.069-01.

Платформа разработки программного обеспечения БЗ» RU.1107746574308.069-01 – это российское программное обеспечение, организация-разработчик: Общество с ограниченной ответственностью «Большая Тройка» (ООО «Большая Тройка»).

Сайт организации-разработчика: <https://big3.ru>.

Организация-правообладатель: Общество с ограниченной ответственностью «Большая Тройка» (ООО «Большая Тройка»).

1.2. Назначение программы

Платформа представляет собой облачную платформу для создания и развития интернет-проектов, которая позволяет конструировать информационные системы и их мобильные версии.

1.3. Особенности распространения и применения

Платформа включает в себя:

- серверную часть;
- веб-приложение.

Серверная часть функционирует под управлением ОС Debian GNU/Linux, в том числе Astra Linux, и может разворачиваться на стороне организации-пользователя.

Веб-приложение обеспечивает взаимодействие с пользователем по технологии «тонкий клиент» с применением распространенных веб-браузеров (Mozilla Firefox, Microsoft Edge, Google Chrome, Safari, «Яндекс.Браузер»).

2. ТРЕБОВАНИЯ К ТЕХНИЧЕСКИМ СРЕДСТВАМ

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

- центральный процессор (CPU): Intel Core i3, 2 ГГц (или эквивалент);
- объем оперативной памяти (RAM): 8 ГБ;
- жесткий диск (HDD): 1x HDD – объем свободного пространства 10 ГБ;
- видеоадаптер: встроен в системную плату;
- сетевая плата: Ethernet 10 Мбит/с;
- дополнительное оборудование: монитор SVGA 1024x768, мышь, клавиатура

ПЭВМ должна быть подключена к сети Интернет.

3. ТРЕБОВАНИЯ К ОБЩЕМУ ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

Для установки и проверки функционирования Платформы на стенде на ПЭВМ должно быть установлено следующее общее программное обеспечение:

- операционная система (далее – ОС): Debian GNU/Linux;
- СУБД: PostgreSQL, ClickHouse, Redis;
- сервис брокера сообщений: RabbitMQ;
- сервис фоновых процедур: Celery;
- сервис регламентных заданий: Cron;
- сервис WebSocket;
- сервис полнотекстового поиска: Manticore Search
- программное средство Docker (<https://www.docker.com>), включая инструментальное средство Docker Swarm;
- веб-браузер Google Chrome актуальной версии.

4. УРОВЕНЬ ПОДГОТОВКИ ПОЛЬЗОВАТЕЛЕЙ

4.1. Требования к квалификации персонала при проверке Платформы

Для выполнения установки и проверки функционирования Платформы пользователь должен обладать следующими знаниями и навыками:

- навыки администрирования Linux-подобных систем;
- навыки использования программного средства Docker и Docker Swarm;
- навыки использования веб-браузера типа Google Chrome;
- знание общих принципов функционирования веб-приложений;
- общих принципов функционирования компьютерных сетей.

4.2. Требования к квалификации персонала при штатной установке и эксплуатации

Установка Платформы в процессе основного жизненного цикла выполняется организацией-разработчиком Платформы. Конечными пользователями установка не выполняется.

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

- знаниями соответствующей предметной области;
- знаниями функциональности Платформы согласно эксплуатационной документации;
- навыками работы на персональном компьютере под управлением ОС Microsoft Windows и Linux;
- навыками использования распространенных веб-браузеров актуальных общедоступных версий.

5. РАЗВЕРТЫВАНИЕ И УСТАНОВКА

5.1. Развертывание серверной части Платформы на Linux

Для развертывания серверной части Платформы на ОС Linux необходимо выполнить следующие действия:

1. выполнить подготовительные работы;
2. установить Docker и настроить контейнеры;
3. установить локальную версию Django;
4. выполнить миграции и проверить работоспособность.

5.1.1. Подготовительные работы

Для развертывания серверной части Платформы необходимо установить Git. Для этого в терминале необходимо выполнить следующие команды:

```
apt install git  
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Далее создать папку, из которой в дальнейшем будут выполняться действия. Название папки – это название нового проекта, который будет разрабатываться на базе Платформы.

```
export project_name=b3_custom  
bash <(wget -qO- https://storage.yandexcloud.net/oc-big3/deploy/init.sh) $project_name && \  
cd $project_name
```

5.1.2. Установка Docker и настройка контейнеров

Для установки Docker и настройки контейнеров необходимо выполнить следующие действия:

1. Установить Docker. При этом автоматически установится и Docker Compose.
2. Добавить своего пользователя в группу Docker.
3. Скопировать локальные настройки по умолчанию, выполнив команду:

```
cp ./${project_name}/settings_local.tpl.py «./${project_name}/settings_local.py»
```

ВНИМАНИЕ!!! Файл tpl копируется в обычные локальные настройки, которые игнорируются в Git. При необходимости можно исправить настройки после запуска проекта.

4. Сгенерировать сертификаты:

```
echo "Генерируем необходимые ключи для работы"
```



```
mkdir -p ./b3platform/access_management_django/jwt_certs/own/  
yes 'y' | ssh-keygen -q -t rsa -b 4096 -m PEM -N "" -f  
./b3platform/access_management_django/jwt_certs/own/jwt.key  
touch build-date.txt
```

5. Установить Docker, если еще не установлен, командой со следующими параметрами:

```
curl -fsSL https://get.docker.com -o get-docker.sh  
sudo sh ./get-docker.sh
```

6. Скопировать ENV-файл для локального запуска:

```
cp ./b3platform/devops/envs/env.local .env
```

7. Собрать основной образ локально командой со следующими параметрами:

```
echo "Собираем образы"  
bash ./b3platform/devops/build_local.sh test  
cp ./b3platform/devops/docker-compose-local.yml docker-compose-local.yml
```

где:

- ./b3platform/devops/build_local.sh – путь до файла скрипта. Скрипт находится в папке приложения. Также можно скачать файл из registry, выполнив команду `docker pull`;
- test – ветка разработки, с кодом которой развернется проект. Если ветку не указывать, то автоматически применится текущая ветка, которая открыта на ПК.

ВНИМАНИЕ!!! Все изменения производятся только в файле `settings_local.py`.

8. Создать и настроить контейнеры. После выполнения приведенных ниже команд корневой путь локального репозитория транслируется в Docker-контейнер, который запущен. Т.е. если локально переключается ветка, то и в Docker происходят изменения, Django при этом определяет изменения и перезагружается с новым кодом соответственно.

Необходимо выполнить следующие команды:

```
docker compose -f docker-compose-local.yml up -d
```

5.1.3. Установка локальной версии Django

Для Ubuntu 20/22 LTS (X64/X86/ARM) необходимо установить Python и зависимости, которые будут использоваться (PostgreSQL, ClickHouse, Redis, RabbitMQ, WebSocket):

```
add-apt-repository ppa:ubuntugis/ppa  
apt-get update  
apt-get install gdal-bin
```

```
apt-get install -y lsb-release gnupg jq ca-certificates curl
apt-get install --yes --no-install-recommends libgdal-dev python3-dev gdal-bin musl python3-gdal
cron less vim pylint netcat default-jre postgresql-client python3-venv make texlive-luatex texlive-
latex-base texlive-lang-cyrillic
apt-get clean autoclean
apt-get autoremove --yes
export CPLUS_INCLUDE_PATH=/usr/include/gdal
export C_INCLUDE_PATH=/usr/include/gdal
```

После завершения установки Python и зависимостей, которые будут в дальнейшем использоваться и активировав venv, необходимо выполнить установку библиотек:

```
python3 -m venv ./venv
source venv/bin/activate
pip install GDAL==$(gdal-config --version)
pip install -r requirements.txt
```

5.1.4. Выполнение миграций и проверка работоспособности

1. Рекомендуется локально устанавливать Django, а остальное – в Docker (БД, Redis, RabbitMQ, Celery и т.д.).

Если установка произведена в Docker и Django не был развернут локально (т.е. пропущен п. 5.1.2 настоящего документа), то сначала необходимо выполнить команду:

```
docker exec -it app-back_main-app-local_1 migrate
```

где app-back_main-app-local_1 – имя контейнера приложения

2. Запустить миграции. Если проект локально не устанавливался, то миграции запускаются в контейнере:

```
python manage.py migrate
```

3. Зайти в админ-панель по адресу localhost:8000/admin под пользователем admin с паролем: 12345678 (либо иным паролем, заданным в файле settings_local.py в параметре ADMIN_USER_PASSWORD).

5.2. Развертывание фронтенд-части Платформы

Для развертывания фронтенд-части Платформы необходимо выполнить следующие действия:

1. Создать директорию для файлов проекта и перейти в эту директорию.

2. Скачать (клонировать) основной репозиторий проекта:

```
git clone git@git.big3.ru:analytics/personal-area.git
```

3. Рекурсивно инициализировать все git-субмодули в локальном основном репозитории:

```
git submodule update --init --recursive
```

4. Инициализировать файлы локальной конфигурации окружения в приложении:

```
docker compose -f docker-compose-local.yml run --rm angular-lk yarn run init
```

5. Установить все NPM-зависимости и запустить локальный сервер разработки приложения с предварительной установкой всех NPM-зависимостей.

```
docker compose -f docker-compose-local.yml up --no-log-prefix
```

5.3. Развертывание на продуктивном стенде

5.3.1. Плейбуки проектов

5.3.1.1. Структура репозитория

Репозиторий имеет следующую структуру:

– inventories/

- group_vars/

- all.yml - файл с дефолтными переменными

- dev.yml - переменные для dev-окружения, описанного в hosts.yml

- ...

- secrets.yml - секреты (пароли)

- hosts.yml - inventory-файл со всеми хостами проекта

– collections/ - папка с зависимостями (задачи/роли/плагины и т.п., появится после установки)

- ansible_collections/

- big3/

- project_collection/ - коллекция ролей и плейбук для развертывания конкретного проекта

- helpers/ - коллекция общих ролей для всех проектов

- ...

– Makefile - точка входа возможных команд проекта

- ansible.cfg - конфигурация Ansible
- requirements.yml - Ansible galaxy файл с зависимостями-ролями

5.3.1.2. Установка зависимостей

Для работы с проектом достаточно выполнить команду:

```
make init
```

5.3.1.3. Работа с проектом

5.3.1.3.1. Inventory-файл

Для первоначальной настройки окружения с нуля необходимо добавить его в ip-адреса в inventory-файл. Наличие групп `swarm_manager` и `postgresql` является обязательным!

```
inventories/project_name/hosts.yml:
```

```
b3platform:                # название окружения
  children:
    swarm_manager:
      hosts:
        "1.2.3.4":         # ip-адрес окружения swarm-менеджера окружения
    postgresql:
      hosts:
        "1.2.3.5":         # ip-адрес PostgreSQL окружения
  secrets:
    children:
      b3platform:
```

Также можно дополнительно изучить документацию по организации inventory-файла:
https://docs.ansible.com/ansible/latest/inventory_guide/intro_inventory.html.

5.3.1.3.2. Секреты

Для работы с секретами необходимо:

1. Завести учётные записи в ресурсах:

- www.cryptopro.ru
- <https://storage.yandexcloud.net>

2. Добавить минимальные секреты в файл `inventories/group_vars/secrets.yml`:

```
# crypto pro license
```

```
сprosp_pin: ""
```

```
csp_license_key: ""

# dadata.ru

dadata_token: ""

edo_account_crypto_key: ""

# storage.yandexcloud.net

aws_access_key_id: ""

aws_secret_access_key: ""

aws_storage_bucket_name: ""
```

Переменная	Описание	Комментарий
dadata_token	Токен доступа в dadata.ru	
aws_access_key_id	access_key от S3 хранилища (https://storage.yandexcloud.net)	
aws_secret_access_key	access_key от S3 хранилища (https://storage.yandexcloud.net)	
aws_storage_bucket_name	Название бакета от S3 хранилища (https://storage.yandexcloud.net)	
pg_host	IP-адрес PostgreSQL (https://www.postgresql.org/)	По умолчанию используется IP-адрес swarm-менеджера
ch_password	Пароль от Clickhouse	Дефолтное значение в Ansible-коллекции big3.platform_collection
rabbitmq_password	Пароль от RabbitMQ	Дефолтное значение в Ansible-коллекции big3.platform_collection
redis_password	Пароль от Redis	Дефолтное значение в Ansible-коллекции big3.platform_collection

3. Для просмотра переменных по умолчанию необходимо выполнить команду:

```
cat collections/ansible_collections/big3/platform_collection/playbooks/group_vars/all.yml
```

4. Сгенерировать ключ и зашифровать файл с секретами *inventories/group_vars/secrets.yml*:

```
date +%s | sha256sum | base64 | head -c 32 > .ansible_password_file
ansible-vault encrypt inventories/group_vars/secrets.yml
```

5.3.1.3.3. Конфигурация Ansible

Для развертывания системы на удалённые хосты из inventory-файла (inventories/hosts.yml) на них должен быть прямой ssh-доступ по приватному ключу и пользователю, указанному в файле *ansible.cfg*:

```
private_key_file = ~/.ssh/ansible_rsa  
remote_user = ansible
```

При необходимости значения можно поменять на другие. Также можно дополнительно изучить документацию Ansible: https://docs.ansible.com/ansible/latest/inventory_guide/connection_details.html.

5.3.2. Развертывание

Для развертывания системы на продуктивном стенде необходимо:

1. Запустить плейбук для установки Swarm и всех необходимых зависимостей:

```
ansible-playbook big3.apro_collection.docker
```

2. Установить PostgreSQL (при необходимости и отсутствии уже существующего экземпляра):

```
ansible-playbook big3.apro_collection.postgresql
```

3. Запустить основное развертывание всей системы:

```
ansible-playbook big3.apro_collection.main -t full
```

4. Обновить только контейнеры с django-приложением через команду:

```
ansible-playbook big3.apro_collection.main -t back
```

5. Обновить только контейнеры с Angular-приложением через команду:

```
ansible-playbook big3.apro_collection.main -t front
```

6. ЗАПУСК

Клиентская часть Платформы функционирует в рамках веб-браузера и запускается автоматически в контексте веб-браузера. Открыв веб-браузер и указав в адресной строке URL-адрес, происходит запуск клиентской части Платформы (в контексте веб-браузера) и отображается страница входа в Платформы.

7. ПРОВЕРКА ФУНКЦИОНИРОВАНИЯ

В таблице 1 приведен порядок выполнения действий для проверки функционирования Платформы.

Таблица 1 – Порядок проверки функционирования Платформы

№ шага	Цель проверки	Действие	Ожидаемый результат
1	Развертывание Платформы	Выполнить действия, описанные в п. 5.2 настоящего документа	Стартовая страница Платформы открывается
2	Выполнить вход в Платформу (процедура аутентификации и авторизации)	Выполнить действия, описанные в п. 4.2 документа «Руководство оператора» (RU.1107746574308.069-01 34 01)	Стартовая страница Платформы с дополнительными функциональными элементами открывается
3	Проверка функционирования Платформы при работе с простыми формами		
3.1	Проверка операции по созданию и первичной настройке типа представления	Выполнить действия, описанные в п. 4.3.2 документа «Руководство оператора» (RU.1107746574308.069-01 34 01)	Операция выполняется корректно
3.2	Проверка операции по созданию и настройке формы списка	Выполнить действия, описанные в п. 4.3.3 документа «Руководство оператора» (RU.1107746574308.069-01 34 01)	Операция выполняется корректно
3.3	Проверка операции по созданию и настройке детальной страницы	Выполнить действия, описанные в п. 4.3.4 документа «Руководство оператора» (RU.1107746574308.069-01 34 01)	Операция выполняется корректно
4	Проверка функционирования Платформы при работе с формлами в конструкторе		
4.1	Проверка работоспособности формул	Выполнить действия, описанные в п. 4.4.4 документа «Руководство оператора» (RU.1107746574308.069-01 34 01)	Операция выполняется корректно
5	Проверка функционирования Платформы при работе с бизнес-процессами со статусами		
5.1	Проверка операции по созданию бизнес-процесса со статусами	Выполнить действия, описанные в п. 4.5.1 документа «Руководство оператора» (RU.1107746574308.069-01 34 01)	Операция выполняется корректно
5.2	Проверка операции по созданию прав доступа на документы	Выполнить действия, описанные в п. 4.5.2.1 документа «Руководство оператора» (RU.1107746574308.069-01 34 01)	Операция выполняется корректно
5.3	Проверка операции по созданию прав доступа на операции	Выполнить действия, описанные в п. 4.5.2.2 документа «Руководство оператора» (RU.1107746574308.069-01 34 01)	Операция выполняется корректно

6	Проверка функционирования Платформы при работе с печатными формами		
6.1	Проверка операции по созданию печатной формы	Выполнить действия, описанные в п. 4.6.1 документа «Руководство оператора» (RU.1107746574308.069-01 34 01)	Операция выполняется корректно
6.2	Проверка операции по созданию шаблонов печатных форм	Выполнить действия, описанные в п. 4.6.2 документа «Руководство оператора» (RU.1107746574308.069-01 34 01)	Операция выполняется корректно
6.3	Проверка операции по настройке подписания печатных форм	Выполнить действия, описанные в п. 4.6.3 документа «Руководство оператора» (RU.1107746574308.068-01 34 01)	Операция выполняется корректно
7	Проверка функционирования Платформы при работе с аналитическими панелями (дашбордами)		
7.1	Проверка операции по созданию аналитической панели (дашборда)	Выполнить действия, описанные в п. 4.7.1 документа «Руководство оператора» (RU.1107746574308.069-01 34 01)	Операция выполняется корректно
7.2	Проверка функционирования Платформы при работе с витриной данных		
7.3	Проверка операции по созданию витрины данных	Выполнить действия, описанные в п. 4.8.1 документа «Руководство оператора» (RU.1107746574308.069-01 34 01)	Операция выполняется корректно
8	Проверка функционирования Платформы при работе с аналитическими функциями		
8.1	Проверка операции по добавлению агрегации на форму/списка	Выполнить действия, описанные в п. 4.9.1 документа «Руководство оператора» (RU.1107746574308.069-01 34 01)	Операция выполняется корректно
8.2	Проверка операции по добавлению состояния на форму элемента/списка	Выполнить действия, описанные в п. 4.9.2 документа «Руководство оператора» (RU.1107746574308.069-01 34 01)	Операция выполняется корректно
9	Проверка настройки операций импорта/экспорта в Excel	Выполнить действия, описанные в п. 4.10 документа «Руководство оператора» (RU.1107746574308.069-01 34 01)	Операция выполняется корректно
10	Проверка функционирования Платформы при работе с публичными сайтами		
10.1	Проверка операции по созданию публичного сайта	Выполнить действия, описанные в п. 4.11.1 документа «Руководство оператора» (RU.1107746574308.069-01 34 01)	Операция выполняется корректно
11	Проверка функционирования модуля «Онбординг»	Выполнить действия, описанные в п. 4.12 документа «Руководство оператора» (RU.1107746574308.069-01 34 01)	Операция выполняется корректно