

DataForge

Инструкция по установке

Версия 1.2

Оглавление

1. Требования к системе	4
2. Необходимые материалы.....	4
3. Установка и обновление продукта	5
3.1 Создание сервера	5
3.2 Установка необходимых служб	5
3.3 Установка Docker.....	6
3.4 Подключение к серверу	6
3.4.1 Получение данных от IT-отдела	6
3.4.2 Подготовка ключей SSH.....	7
3.4.3 Запуск терминала	7
3.4.4 Команда подключения.....	7
3.4.5 Подтверждение подключения	8
3.4.6 Проверка Docker	8
3.5 Загрузка архива DataForge на сервер.....	8
3.6 Создание рабочей папки и распаковка	8
3.7 Структура проекта.....	10
3.8 Переменные окружения	10
3.9 Настройка окружения.....	13
3.10 Настройка SSL-сертификатов для HTTPS	13
3.11 Настройка IP-адреса сервера.....	14
3.12 Инициализация базы данных	15
3.13 Получение токена для доступа к образам	15
3.14 Альтернатива: ключ сервисного аккаунта (без срока действия).....	15
3.15 Запуск приложения.....	16
3.16 Как остановить и удалить.....	16
3.17 Готовые URL после запуска	16
3.18 Учетные данные для первого запуска	16
3.19 Создание и применение бэкапов.....	16
3.20 Обновление приложения.....	17
4. Настройки.....	17
4.1 Раздел «Моя компания»	17
4.1.1 Лицензии	17

4.1.2	Пользователи компании	18
4.1.3	Завершение настройки.....	19

1. Требования к системе

В данном разделе приведены технические условия, необходимые для корректной установки и эксплуатации платформы: требования к клиентскому оборудованию и браузерам, среде развертывания, поддерживаемым операционным системам, минимальным серверным ресурсам, а также перечень СУБД, с которыми система может взаимодействовать.

Клиентская часть

Требование	Описание
Поддерживаемые браузеры	Последние стабильные версии Google Chrome и Yandex Browser
Доступ в интернет	Требуется только при удалённом расположении сервера. При локальном размещении серверной части доступ в интернет не нужен

Среда развертывания (контейнеризация)

Продукт поставляется в виде Docker-контейнеров. Требования к среде:

Компонент	Требование
Docker Desktop	Версия 3.0.0 и выше (для сред с графическим интерфейсом)
Docker Engine + Compose	Docker Engine 19.03.0+ и Docker Compose 1.28.0+ (для серверных сред)

Поддерживаемые операционные системы (любые ОС, совместимые с указанными версиями Docker):

- Российские ОС: Astra Linux, РЕД ОС, ALT Linux
- Зарубежные ОС: Ubuntu, Debian, другие дистрибутивы Linux

Минимальные серверные ресурсы (на одну инсталляцию/компанию)

Ресурс	Требование
Процессор	4 vCPU (2.0 GHz)
Оперативная память	8 ГБ
Дисковое пространство	100 ГБ (SSD) для базы данных метаданных и хранения файлов

Поддерживаемые СУБД для подключений

- PostgreSQL и российские аналоги: PostgreSQL, Postgres Pro, Jatoba
- ClickHouse

Поддерживаются стандартные драйверы и протоколы для подключения к указанным СУБД.

2. Необходимые материалы

Перед началом установки убедитесь, что в вашем распоряжении имеются следующие материалы, предоставляемые вендором:

Материал	Описание
----------	----------

Архив с дистрибутивом DC_DF.zip	Содержит файлы проекта: docker-compose.yaml, .env, вспомогательные скрипты (init.sh, set-server-host.sh).
Токен доступа к AWS ECR	Команда для авторизации Docker в приватном реестре образов (действителен 12 часов).
Лицензионный файл .lic	Файл лицензии, активирующий функционал продукта.

Данные артефакты необходимы для успешного развертывания и последующей работы системы.

3. Установка и обновление продукта

3.1 Создание сервера

Создайте сервер, соответствующий указанным выше требованиям.

- В качестве примера, в данной инструкции используется работа с операционной системой Linux «Debian 12 Bookworm».
- Администратору необходимо создать на этом сервере пользователя с необходимыми правами и настроенным доступом через SSH, под которым будет выполняться установка.

```
# User privilege specification
root ALL=(ALL:ALL) ALL

# Defaults requiretty

# Allow dev user to run specific commands as root

<USERNAME> ALL=(root) /bin/sh,/bin/mkdir,/bin/rm,/bin/cp, /bin/chmod, /bin/system-
ctl,/bin/kill,/usr/sbin/sys-
ctl,/usr/bin/gpasswd,/usr/sbin/groupadd,/usr/bin/chown,/usr/sbin/mod-
probe,/usr/bin/echo,/usr/bin/sed,/usr/sbin/swapoff ,/usr/sbin/ip ,/bin/sudo
```

3.2 Установка необходимых служб

Для обеспечения комфортной работы с системой и выполнения операций с архивами установите на сервер следующий набор программного обеспечения:

```
sudo apt update
sudo apt install wget curl unzip ca-certificates net-tools build-essential openssl
```

```

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
wget is already the newest version (1.21.3-1+deb12u1).
wget set to manually installed.
curl is already the newest version (7.88.1-10+deb12u14).
unzip is already the newest version (6.0-28).
ca-certificates is already the newest version (20230311+deb12u1).
ca-certificates set to manually installed.
openssl is already the newest version (3.0.19-1~deb12u2).
openssl set to manually installed.
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu cpp cpp-12 dirmngr dpkg-dev fakeroot fontconfig-config
  fonts-dejavu-core g++ g++-12 gcc gcc-12 gnupg gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client gpg-wks-server
  gpgconf gpgsm libabsl20220623 libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libaom3
  libasan8 libassuan0 libatomic1 libavif15 libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev
  libctf-nobfd0 libctf0 libdavid6 libde265-0 libdeflate0 libdpkg-perl libfakeroot libfile-fcntllock-perl
  libfontconfig1 libgav1-1 libgcc-12-dev libgd3 libgomp1 libgprofng0 libheif1 libisl23 libitm1 libjbig0
  libjpeg62-turbo libksba8 liblerc4 liblsan0 libmpc3 libmpfr6 libnph0 libnsl-dev libquadmath0 librav1e0
  libstdc++-12-dev libsvtav1enc1 libtiff6 libtirpc-dev libtsan2 libubsan1 libwebp7 libx265-199 libxpm4 libyuv0
  linux-libc-dev make manpages manpages-dev patch pinentry-curses rpcsvc-proto
Suggested packages:
  binutils-doc cpp-doc gcc-12-locales cpp-12-doc pinentry-gnome3 tor debian-keyring g++-multilib g++-12-multilib
  gcc-12-doc gcc-multilib autoconf automake libtool flex bison gdb gcc-doc gcc-12-multilib parcimonie xloadimage
  sddaemon glibc-doc git bzr libgd-tools libstdc++-12-doc make-doc man-browser ed diffutils-doc pinentry-doc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential cpp cpp-12 dirmngr dpkg-dev fakeroot
  fontconfig-config fonts-dejavu-core g++ g++-12 gcc gcc-12 gnupg gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client
  gpg-wks-server gpgconf gpgsm libabsl20220623 libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libaom3 libasan8 libassuan0 libatomic1 libavif15 libbinutils libc-dev-bin libc-devtools
  libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdavid6 libde265-0 libdeflate0 libdpkg-perl libfakeroot
  libfile-fcntllock-perl libfontconfig1 libgav1-1 libgcc-12-dev libgd3 libgomp1 libgprofng0 libheif1 libisl23 libitm1
  libjbig0 libjpeg62-turbo libksba8 liblerc4 liblsan0 libmpc3 libmpfr6 libnph0 libnsl-dev libquadmath0 librav1e0
  libstdc++-12-dev libsvtav1enc1 libtiff6 libtirpc-dev libtsan2 libubsan1 libwebp7 libx265-199 libxpm4 libyuv0
  linux-libc-dev make manpages manpages-dev net-tools patch pinentry-curses rpcsvc-proto
0 upgraded, 85 newly installed, 0 to remove and 0 not upgraded.
Need to get 88.3 MB of archives.
After this operation, 333 MB of additional disk space will be used.
Do you want to continue? [Y/n] |

```

3.3 Установка Docker

Официальная инструкция по установке Docker: [Get Docker](#)

```

ubuntu@dataforge-local:~$ sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plu
gin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras git git-man iptables libcurl3-gnutls liberror-perl libip6tc2 libnetfilter-conntrack3
  libnfnetwork0 libslirp0 pigz slirp4netns
Suggested packages:
  cgroupfs-mount | cgroup-lite docker-model-plugin git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk
  gitweb git-cvs git-mediawiki git-svn firewallld
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin git
  git-man iptables libcurl3-gnutls liberror-perl libip6tc2 libnetfilter-conntrack3 libnfnetwork0 libslirp0 pigz
  slirp4netns
0 upgraded, 17 newly installed, 0 to remove and 0 not upgraded.
Need to get 105 MB of archives.
After this operation, 417 MB of additional disk space will be used.
Do you want to continue? [Y/n]

```

Добавляем текущего пользователя в группу докер для работы без `sudo`

```
sudo usermod -aG docker $USER && newgrp docker
```

3.4 Подключение к серверу

В случае, когда создание и настройка сервера выполняются сторонней ИТ-службой (аутсорсинг), а работа с продуктом DataForge остаётся за вами, следуйте приведённой ниже последовательности действий. Если вы самостоятельно создаёте сервер и устанавливаете продукт, данный раздел можно пропустить.

3.4.1 Получение данных от ИТ-отдела

Перед началом работы запросите у ИТ-специалистов:

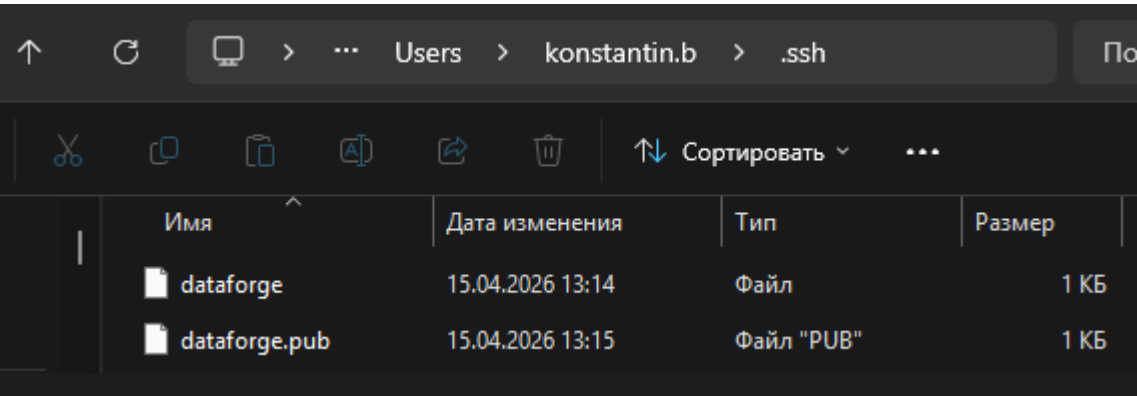
- Файл с токеном (приватный ключ) для подключения к серверу.
- Адрес сервера для SSH-подключения.

3.4.2 Подготовка ключей SSH

Разместите полученные файлы ключей в директории:

Пример по умолчанию

C:\Users\<имя_пользователя>\.ssh



3.4.3 Запуск терминала

Все операции выполняются в командной строке (терминале). Откройте одно из следующих распространённых приложений (или аналоги):

- Windows PowerShell
- Командная строка (CMD)
- Git Bash

3.4.4 Команда подключения

Введите команду:

Пример, в команде должен быть указан путь к вашему ключу, а также ваш логин и ip адрес

ssh -i .ssh\dataforge ubuntu@111.88.151.149

Элемент	Описание
ssh	Программа для безопасного подключения к удалённому серверу
-i .ssh\dataforge	Путь к приватному ключу (в Windows используются обратные слэши \)
ubuntu@111.88.151.149	Логин <Логин для подключения> и публичный IP-адрес сервера

```
PS C:\Users\konstantin.b> ssh -i .ssh\dataforge ubuntu@111.88.159.147
Linux dataforge-local 6.1.0-44-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.164-1 (2026-03-09) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr 15 09:19:51 2026 from 213.33.244.140
```

3.4.5 Подтверждение подключения

При первом подключении система запросит подтверждение доверия к хосту (fingerprint). Введите `yes` и нажмите `Enter`.

```
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

3.4.6 Проверка Docker

Перед тем как загружать архив, убедимся, что Docker установлен и вы можете им пользоваться без `sudo` (это уже сделано на этапе подготовки). Выполните:

```
docker ps
```

Если вы видите таблицу с заголовками (скорее всего пустую) — Docker работает.

```
ubuntu@dataforge-local:~/dataforge_folder/DC_DF$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED
ED	STATUS	PORTS	

Если появляется ошибка `permission denied`, значит пользователь еще не добавлен в группу `docker`. Попросите администратора выполнить команду, которую он уже запланировал:

```
sudo usermod -aG docker $USER && newgrp docker
```

После этого выйдите из сессии (`exit`) и зайдите заново.

3.5 Загрузка архива DataForge на сервер

Теперь нужно перенести архив с вашего компьютера на сервер. Для этого используется команда `scp` (secure copy).

На вашем локальном компьютере (в новом окне терминала, не в сессии на сервере!) выполните:

```
scp -i .ssh\dataforge /путь/к/архиву/DC_DF.zip ubuntu@111.88.151.149:~
```

- `-i .ssh\dataforge` — снова указываем ключ.
- `/путь/к/архиву/DC_DF.zip` — замените на реальный путь к файлу архива, который вам дали.
- `ubuntu@111.88.151.149:~` — копируем в домашнюю папку пользователя `ubuntu` на сервере (символ `~` означает домашнюю директорию).

```
C:\Users\konstantin.b>scp -i .ssh\dataforge C:/Users/konstantin.b/Downloads/DF/DC_DF.zip ubuntu@111.88.159.147:~
DC_DF.zip 100% 59KB 1.1MB/s 00:00
```

3.6 Создание рабочей папки и распаковка

Теперь снова перейдите в SSH-сессию на сервере.

- **Создайте папку** для проекта, например, `dataforge_folder`:

```
mkdir ~/dataforge_folder
```

- **Переместите архив** в эту папку и перейдите в нее:

```
mv ~/DC_DF.zip ~/dataforge_folder/ #Перемещает архив в папку
```

```
cd ~/dataforge_folder #Перейти в папку
```

```
ubuntu@dataforge-local:~$ mkdir ~/dataforge_folder
ubuntu@dataforge-local:~$ mv ~/DC_DF.zip ~/dataforge_folder
ubuntu@dataforge-local:~$ cd ~/dataforge_folder
ubuntu@dataforge-local:~/dataforge_folder$
```

- **Распакуйте архив.** Обычно используется `unzip`. Если команда не найдена, установите:

```
sudo apt install unzip -y
```

Затем распакуйте:

```
unzip DC_DF.zip
```

```
ubuntu@dataforge-local:~/dataforge_folder$ unzip DC_DF.zip
Archive:  DC_DF.zip
  inflating: DC_DF/docker-compose.yaml
  inflating: DC_DF/ssl/key.pem
  inflating: DC_DF/ssl/cert.pem
  inflating: DC_DF/update-app.sh
  inflating: DC_DF/init.sh
  inflating: DC_DF/nginx/Dockerfile
  inflating: DC_DF/nginx/nginx.conf
  inflating: DC_DF/nginx/entrypoint.sh
  inflating: DC_DF/set-server-host.sh
  inflating: DC_DF/.env
  inflating: DC_DF/.git/REBASE_HEAD
  inflating: DC_DF/.git/ORIG_HEAD
  inflating: DC_DF/.git/config
  inflating: DC_DF/.git/objects/0c/1357043fa93638c17af7d516a1f52b308b76e5
```

После распаковки вы увидите несколько файлов и папок. Структура проекта более подробно описана в следующей главе.

Для просмотра полного списка файлов и папок в текущей директории используйте команду:

```
ls -la
```

```
ubuntu@dataforge-local:~/dataforge_folder/DC_DF$ ls -la
total 60
drwxr-xr-x 6 ubuntu ubuntu 4096 Apr 17 07:36 .
drwxr-xr-x 3 ubuntu ubuntu 4096 Apr 15 15:14 ..
-rw-r--r-- 1 ubuntu ubuntu 3609 Apr 16 13:49 docker-compose.yaml
-rw-r--r-- 1 ubuntu ubuntu 2806 Apr 16 13:04 .env
-rw-r--r-- 1 ubuntu ubuntu 2804 Apr 16 12:44 .env.backup.20260416_124443
-rw-r--r-- 1 ubuntu ubuntu 2801 Apr 16 12:45 .env.backup.20260416_124529
drwxr-xr-x 8 ubuntu ubuntu 4096 Apr 15 14:53 .git
drwxr-xr-x 2 ubuntu ubuntu 4096 Apr 15 14:53 .idea
-rw-r--r-- 1 ubuntu ubuntu 162 Jan 26 17:05 init.sh
drwxr-xr-x 2 ubuntu ubuntu 4096 Apr 15 14:53 nginx
-rwxr-xr-x 1 ubuntu ubuntu 4825 Jan 26 17:05 set-server-host.sh
drwxr-xr-x 2 ubuntu ubuntu 4096 Apr 16 14:03 ssl
-rw-r--r-- 1 ubuntu ubuntu 7114 Jan 26 17:05 update-app.sh
```

Все дальнейшие запросы необходимо осуществлять в корневой папке, с распакованным архивом. Используйте команду чтобы перейти в созданную при распаковке папку:

```
Cd DC_DF
```

- **(Опционально)** Удалите архив, чтобы не занимал место:

```
rm DC_DF.zip
```

3.7 Структура проекта

```
DC_DF.zip
```

```
├─ .env                # готовые переменные окружения
├─ docker-compose.yml  # описание всех сервисов
├─ set-server-host.sh  # скрипт для настройки имени сервера
├─ update-app.sh       # скрипт для обновления приложения
├─ backups/            # директория для хранения резервных копий
├─ nginx/              # конфигурационные файлы Nginx
└─ ssl/                # сертификаты SSL/TLS для HTTPS
```

3.8 Переменные окружения

Переменная	Значение по умолчанию	Назначение
SERVER_HOST	localhost	Адрес сервера
SERVER_PROTOCOL	http	Протокол сервера (http или https)
WS_PROTOCOL	ws	Протокол WebSocket (ws или wss)
DB_ENGINE	postgres	Драйвер БД, который активирует приложение

DB_HOST	postgres	Хост БД
DB_PORT	5432	Порт БД
DB_NAME	dataforge	Название БД
DB_USER	postgres	Имя пользователя БД
DB_PASSWORD	pwd	Пароль БД
ACCESS_TOKEN_SECRET	nQjF...	Секретный ключ для подписи JWT access-токена
REFRESH_TOKEN_SECRET	w+54...	Секретный ключ для подписи JWT refresh-токена
ApiHashBodyKey	\$2a\$10Cw...	Ключ для шифрования/дешифрования тела API (например, пароля пользователя)
VITE_API_HASH_BODY_KEY	\$2a\$10Cw...	Ключ для фронтенда, аналогично ApiHashBodyKey
LICENSE_PUBLIC_KEY	-----BEGIN PUBLIC KEY----- -- ...	Публичный ключ лицензии для проверки цифровой подписи
ACCESS_TOKEN_EXPIRATION	60s	Время жизни access-токена
REFRESH_TOKEN_EXPIRATION	7d	Время жизни refresh-токена
APP_PORT	3001	Порт основного API
WS_PORT	3002	Порт WebSocket-сервиса
FRONTEND_PORT	8080	Порт фронтенда
OBJECT_STORAGE_HOST	seaweedfs	Хост объектного хранилища
OBJECT_STORAGE_PORT	8333	Порт SeaweedFS S3 API

OBJECT_STORAGE_ACCESS_KEY	admin	Access Key для SeaweedFS S3
OBJECT_STORAGE_SECRET_KEY	adminadmin	Secret Key для SeaweedFS S3
SMTP_USERNAME	-	Пользователь SMTP
SMTP_PASSWORD	-	Пароль пользователя SMTP
SMTP_HOST	-	Хост SMTP-сервера
SMTP_PORT	-	Порт SMTP
SMTP_SECURE	-	true — TLS «сразу» (обычно порт 465). false — без TLS на старте (для 587 обычно используется STARTTLS)
SMTP_TLS_REJECT_UNAUTHORIZED	-	Проверка TLS-сертификата сервера
SMTP_TLS_SERVERNAME	-	SNI/имя хоста для проверки сертификата
SMTP_TLS_CIPHERS	-	Явный список cipher suites для TLS
SMTP_REQUIRE_TLS	-	Требовать TLS/STARTTLS
SMTP_IGNORE_TLS	-	Игнорировать STARTTLS
StorageBucket	my-bucket	Имя корзины для файлов
StorageCertificatesBucket	my-certificates-bucket	Бакет для хранения сертификатов/SSL файлов
SUPER_ADMIN_EMAIL	admin@example.com	Почта суперадминистратора
SUPER_ADMIN_PASSWORD	adminadmin	Пароль суперадминистратора
COMPANY_NAME	Main Company	Название компании

Важно (1): Чтобы можно было подключаться к приложению по локальной сети, в параметре `SERVER_HOST` необходимо указать доменное имя или IP-адрес сервера, на котором установлено приложение, и по которому пользователи будут к нему подключаться.

Важно (2): После изменений переменных, обязательно запустить приложение с `--build` флагом, иначе изменения не применятся.

3.9 Настройка окружения

DataForge использует переменные окружения для конфигурации. Теперь нужно открыть файл `.env` для редактирования. Вы можете использовать `nano` — простой редактор командной строки:

```
nano .env
```

```
GNU nano 7.2 .env
#####
## НАСТРОЙКИ СЕРВЕРА ##
#####

# Базовый адрес сервера (замените на IP вашего сервера)
SERVER_HOST=localhost
# Протокол (http или https)
SERVER_PROTOCOL=http
# WebSocket протокол (ws или wss)
WS_PROTOCOL=ws

#####
## БАЗА ДАННЫХ (MySQL) ##
#####

# Хост БД (название из docker-compose)
DB_HOST=mysql

# Порт БД (стандартный MySQL порт)
DB_PORT=3306

# Имя базы данных
DB_NAME=dataforge

# Учетные данные
DB_USER=root
DB_PASSWORD=pwd

#####
## БЕЗОПАСНОСТЬ И АУТЕНТИФИКАЦИЯ ##
#####
```

Навигация в `nano`: стрелочки для перемещения, `Ctrl+O` для сохранения (подтвердите имя файла `Enter`), `Ctrl+X` для выхода.

Если вы не уверены какие значения хотите изменить, пока оставьте как есть или обратитесь к разделу переменные окружения для уточнения. Многие поля уже предзаполнены значениями по умолчанию, и их можно оставить.

3.10 Настройка SSL-сертификатов для HTTPS

Для работы приложения через HTTPS необходимо получить SSL-сертификат для домена, указанного в переменной `SERVER_HOST`.

Сертификаты должны быть размещены в папке `./ssl`, которая находится рядом с файлами проекта. Стандартно набор файлов может включать:

- `cert.pem` или `имя_домена.crt` — основной сертификат.
- `key.pem` или `имя_домена.key` — приватный ключ.

- `ca.crt` или `имя_домена_ca.crt` — промежуточные сертификаты, если они предоставлены центром сертификации.
- `имя_домена.p7b` — альтернативный формат сертификата (опционально, если используется).

Для создания самоподписанного сертификата необходимо внести следующие команды:

```
#Создание папки ssl
mkdir -p ssl

#Создание необходимых файлов в папке
openssl req -x509 -newkey rsa:4096 -nodes -out ssl/cert.pem -keyout ssl/key.pem -days 365 -subj
```

Если сертификат состоит из нескольких файлов (например, основной и промежуточные), их необходимо объединить в один файл, чтобы приложение могло корректно настроить HTTPS. Для этого можно использовать следующие команды, которые создадут объединённый сертификат и скопируют ключ в папку `./ssl`:

Linux / MacOS

```
# Объединить основной сертификат и промежуточные
cat имя_домена.crt имя_домена_ca.crt > ./cert.pem

# Скопировать приватный ключ
cp имя_домена.key ./key.pem
```

Windows (PowerShell или Git Bash)

```
# Объединить основной сертификат и промежуточные
cat имя_домена.crt имя_домена_ca.crt > .\cert.pem

# Скопировать приватный ключ
cp имя_домена.key .\key.pem
```

После выполнения этих команд в папке будут созданы файлы `cert.pem` и `key.pem`, которые необходимо оставить защищёнными от несанкционированного доступа. Эти файлы нужно разместить в папке `./ssl` так, чтобы полный путь к ним был `./ssl/cert.pem` и `./ssl/key.pem`. Приложение будет использовать именно эти файлы для настройки HTTPS-соединений, обеспечивая безопасный доступ к сервисам.

3.11 Настройка IP-адреса сервера

При первом запуске или при изменении сетевых настроек выполните автоматическую настройку IP адреса сервера с помощью скрипта `set-server-host.sh`:

```
# Сделайте скрипт исполняемым (при первом запуске)
chmod +x set-server-host.sh
```

```
# Запустите скрипт настройки
./set-server-host.sh
```

Скрипт предложит выбрать тип IP адреса:

- Локальный IP - для доступа в локальной сети (192.168.x.x, 10.x.x.x).
- Внешний IP - для доступа из интернета.
- Ручной ввод - для ввода конкретного IP или домена.

```
ubuntu@dataforge-local:~/dataforge_folder/DC_DF$ ./set-server-host.sh
=== Настройка SERVER_HOST ===

Выберите тип IP адреса:
1) Локальный IP (для доступа в локальной сети: 192.168.x.x, 10.x.x.x)
2) Внешний IP (для доступа из интернета)
3) Ввести IP вручную

Ваш выбор (1-3): _
```

После выполнения скрипт автоматически обновит переменную `SERVER_HOST` в файле `.env` и создаст резервную копию для возможности отката при необходимости.

3.12 Инициализация базы данных

База данных создаётся автоматически при первом запуске контейнера PostgreSQL. Контейнер использует переменные `DB_NAME`, `DB_USER` и `DB_PASSWORD` из `.env` для инициализации (через стандартные `POSTGRES_DB`, `POSTGRES_USER`, `POSTGRES_PASSWORD` PostgreSQL-образа), поэтому никаких отдельных скриптов запускать не требуется.

При повторных запусках данные сохраняются в Docker volume и не пересоздаются.

3.13 Получение токена для доступа к образам

Образы хранятся в приватном репозитории Yandex Container Registry. Чтобы Docker мог их скачать, выполните команду, которую вы получите от нас:

```
# Пример, в команде будет реальный токен

docker login -u iam -p "Token" cr.yandex
```

Токен действует **ровно 12 часов**. По истечении срока запросите новый.

3.14 Альтернатива: ключ сервисного аккаунта (без срока действия)

Если ручное получение токена раз в 12 часов неудобно, мы можем выдать вам **авторизованный ключ сервисного аккаунта** (`key.json`) с правом только на pull нужного репозитория. Этот ключ выпускается один раз и не имеет срока действия — вы сможете логиниться без нашего участия:

```
# Файл key.json мы передадим вам отдельно

cat key.json | docker login --username json_key --password-stdin cr.yandex
```

3.15 Запуск приложения

В папке с файлами откройте терминал и выполните команду:

```
docker compose up -d
```

Docker скачает и поднимет образы. При последующих запусках этой команды будет происходить только поднятие локальных образов.

```
ubuntu@dataforge-local:~/dataforge_folder/DC_DF$ docker compose up -d
WARN[0000] /home/ubuntu/dataforge_folder/DC_DF/docker-compose.yaml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] up 41/46
 ✔ Image 502502776148.dkr.ecr.us-east-1.amazonaws.com/dataforge/sockets:latest [#####] Pulling 28.6s
 ✔ Image minio/mc [#####] Pulled 15.1s
 ✔ Image 502502776148.dkr.ecr.us-east-1.amazonaws.com/dataforge/app:latest [#####] Pulling 28.6s
 ✔ Image mysql:8.0 [#####] 247.3MB / 247.3MB Pulling 28.6s
 ✔ Image quay.io/minio/minio:RELEASE.2025-04-22T22-12-26Z [#####] Pulled 16.6s
```

3.16 Как остановить и удалить

```
docker compose down # остановить
```

```
docker compose down -v # остановить и удалить тома (очистить данные)
```

3.17 Готовые URL после запуска

После запуска приложения, оно будет располагаться по адресу [https://\\${SERVER_HOST}](https://${SERVER_HOST}) (по умолчанию <https://localhost>)

3.18 Учетные данные для первого запуска

При первом запуске приложения автоматически создается суперадминистратор, если его учетные данные не изменялись в `.env`, то можно зайти под следующими данными:

- Почта: `admin@example.com`
- Пароль: `adminadmin`

Важно! В целях безопасности измените логин и пароль суперадминистратора, установленные по умолчанию, на уникальные значения сразу после первого входа в систему.

3.19 Создание и применение бэкапов

Создание бэкапов зависит от операционной системы сервера, для их выполнения необходимо выполнить одну из следующих команд:

```
# # Linux/MacOS/WSL
```

```
docker compose exec -e PGPASSWORD=[password] postgres pg_dump -U [user] -d [database_name] > backups/backup_$(date +%Y%m%d_%H%M%S).sql
```

```
# Если переменные среды не изменялись
```

```
docker compose exec -e PGPASSWORD=pwd postgres pg_dump -U postgres -d dataforge > backups/backup_$(date +%Y%m%d_%H%M%S).sql
```

```
# Windows PowerShell
```

```
docker compose exec -e PGPASSWORD=[password] postgres pg_dump -U [user] -d [database_name] > backups/backup_$(Get-Date -Format "yyyyMMdd_HH:mm:ss").sql
```

```
# Если переменные среды не изменялись
```

```
docker compose exec -e PGPASSWORD=pwd postgres pg_dump -U postgres -d dataforge > backups/backup_$(Get-Date -Format "yyyyMMdd_HH:mm:ss").sql
```

Если изменялся пароль или название БД, то необходимо заменить его на верные значения.

Для применения бэкапов используются следующие команды:

```
# Пересоздаём БД (DROP + CREATE)
```

```
docker compose exec -e PGPASSWORD=pwd postgres psql -U postgres -d postgres -c "DROP DATABASE dataforge;"
```

```
docker compose exec -e PGPASSWORD=pwd postgres psql -U postgres -d postgres -c "CREATE DATABASE dataforge;"
```

```
# Linux/MacOS/WSL
```

```
docker compose exec -T -e PGPASSWORD=[password] postgres psql -U [user] -d [database_name] < backups/backup_file.sql
```

```
# Если переменные среды не изменялись
```

```
docker compose exec -T -e PGPASSWORD=pwd postgres psql -U postgres -d dataforge < backups/backup_file.sql
```

```
# Windows PowerShell
```

```
Get-Content backups/backup_file.sql -Encoding UTF8 | docker compose exec -T -e PGPASSWORD=[password] postgres psql -U [user] -d [database_name]
```

```
# Если переменные среды не изменялись
```

```
Get-Content backups/backup_file.sql -Encoding UTF8 | docker compose exec -T -e PGPASSWORD=pwd postgres psql -U postgres -d dataforge
```

3.20 Обновление приложения

После создания бэкапа, необходимо выполнить следующие команды для полного обновления приложения:

```
docker compose down
```

```
docker compose up -d --build --pull always
```

4. Настройки

4.1 Раздел «Моя компания»

4.1.1 Лицензии

В верхней части раздела **«Информация о компании»** отображаются сведения о вашей организации: название компании и активная лицензия продукта. Здесь же находятся кнопки для управления лицензией:

Информация о компании

Название компании

Main Company

Лицензия

✓ Тип "Профессиональный", срок действия с 20.01.2026 по 31.05.2026

- Загрузить лицензию — открывает окно выбора файла. Выберите локальный файл с расширением .lic, и система применит новую лицензию.
- Посмотреть лицензию — показывает модальное окно с подробными параметрами: тип лицензии, даты начала и окончания, а также лимиты по количеству пользователей, проектов и версий проектов. Параметры лицензии:
 - Тип лицензии
 - Дата начала лицензии
 - Дата окончания лицензии
 - Количество пользователей
 - Количество проектов
 - Количество версий проектов

Информация о лицензии

Тип лицензии

Профессиональный

Дата начала

20.01.2026

Дата окончания

31.05.2026

Пользователи

1 из 5

Проекты

1 из 10

Версии проектов

1 из 30

Заккрыть

- Удалить лицензию — после подтверждения удаляет текущую лицензию. После этого вы сможете загрузить новый лицензионный файл.

4.1.2 Пользователи компании

Ниже расположен список всех пользователей вашей компании. В таблице представлены следующие данные:

- ID, Имя, Фамилия, Адрес электронной почты
- Роль пользователя (администратор, разработчик, аналитик, наблюдатель)
- Последняя дата входа, Статус

- Действия — кнопки «Изменить» и «Удалить» для каждой строки.

Пользователи компании

[+ Добавить пользователей](#)

#ID	Имя	Фамилия	Адрес электронной почты	Роль пользователя	Последняя дата входа	Статус	Действия
1	superadmin		df-admin@businessqlik.com	Суперадминистратор	25.03.2026, 12:53	Включен	

Чтобы добавить нового пользователя, нажмите кнопку «Добавить пользователя» над списком. В открывшемся окне укажите его электронную почту и выберите роль. После сохранения пользователь получит приглашение на указанный адрес.

Добавить пользователей

Электронная почта

Роль пользователя

Разработчик

Администратор

Разработчик

Аналитик

Наблюдатель

Редактирование пользователя доступно через кнопку «Изменить» в строке списка. Вы можете изменить имя, фамилию и электронную почту. Изменение роли и статуса доступно только администратору системы.

Редактировать пользователя

Имя

Пользователь

Фамилия

Администратор

Электронная почта

konstantin.b@businessqlik.com

Компания

BI Consult 2

Роль пользователя

Администратор

Статус

Включен

API-ключ

[Создать API-ключ](#)

Отмена

Сохранить

Кнопка «Удалить» требует подтверждения — после этого доступ пользователя к системе будет закрыт.

4.1.3 Завершение настройки

После того как вы добавили пользователей, настройка учётных записей в вашей компании считается завершённой.

Система готова к полноценной работе: пользователи могут входить в систему под своими ролями, администратор — управлять доступом, а аналитики и разработчики — создавать проекты и витрины данных.

При необходимости вы всегда можете вернуться в этот раздел, чтобы добавить новых сотрудников, скорректировать роли или отозвать доступ. Все изменения применяются мгновенно и не требуют перезапуска системы.

Теперь вы можете переходить к полноценной работе с продуктом.