

СОМБУХ. Руководство по эксплуатации

Данный документ содержит сведения, необходимые для установки и администрирования Программного продукта Comбух (далее - система).

ООО «Базис Технологии» (далее - разработчик системы) оставляет за собой право вносить изменения в настоящий документ без предварительного уведомления.

Данный документ и его отдельные части не подлежат воспроизведению, публикации и передаче третьим лицам без письменного разрешения разработчика системы.

Редакция от 08.06.2023.

© ООО «Базис Технологии», 2022 – 2023. Все права защищены.

- [1. Список используемых аббревиатур и сокращений](#)
- [2. Системные требования](#)
 - [2.1. Минимальные системные требования](#)
 - [2.2. Рекомендуемые системные требования](#)
- [3. Необходимое программное обеспечение и состав дистрибутива системы](#)
- [4. Подготовка инфраструктуры](#)
 - [4.1. Создание папок](#)
 - [4.2. Установление SSH соединение серверов](#)
 - [4.3. Миграция образов](#)
 - [4.4. Миграция Compose и конфигурационных файлов](#)
 - [4.5. Редактирование Compose и конфигурационного файлов](#)
 - [4.6. Примеры Compose и конфигурационных файлов для сервисов системы](#)
 - [4.6.1. Comбух](#)
 - [4.6.2. Redis](#)
 - [4.6.3. Audit Logger](#)
 - [4.6.4. Apache Airflow](#)
 - [4.6.5. Graylog](#)
 - [4.6.6. Email domain validation](#)
 - [4.7. Создание ролей и пользователей \(DB PostgreSQL\)](#)
 - [4.8. Создание баз данных](#)
- [5. Установка сервисов](#)
 - [5.1. Установка Comбух](#)
 - [5.2. Установка Apache Airflow](#)
 - [5.3. Установка и настройка Graylog](#)
- [6. Обновление или перезапуск сервисов](#)
 - [6.1. Описание базовых команд](#)
- [7. Smoke тестирование функционала сервисов](#)
 - [7.1. Проверка сервисов Comбух](#)
- [8. Действия при возможных неисправностях](#)
 - [8.1. Действия в случае несоблюдения условий выполнения технологического процесса, в том числе при длительных отказах технических средств](#)
 - [8.2. Действия по восстановлению программ и/или данных при отказе магнитных носителей или обнаружении ошибок в данных](#)

- [8.3. Действия в случаях обнаружении несанкционированного вмешательства в данные](#)
 - [8.4. Действия в других аварийных ситуациях](#)
 - [8.4.1. Возможные аварийные ситуации](#)
 - [8.4.2. Необходимые действия](#)
 - [9. Администрирование системы](#)
 - [9.1. Вкладка "Состояние системы"](#)
 - [9.1.1. Особенности экрана "Состояние системы"](#)
 - [9.1.2. Отображаемые модули системы \(виджеты\)](#)
 - [9.2. Вкладка "Журнал системных сообщений"](#)
 - [9.2.1. Особенности экрана "Журнал системных сообщений"](#)
 - [9.3. Вкладка "Управление пользователями"](#)
 - [9.3.1. Особенности экрана "Управление пользователями"](#)
 - [10. Начало работы с системой](#)
-

Аннотация

В первой части документа описан порядок действий по установке и настройке компонентов системы, а также указаны рекомендации по устранению возможных ошибок, неисправностей и подготовке рабочего окружения.

Во второй части документа описан порядок работы с Панелью администратора в системе (раздел системы "Администрирование").

В третьей части документа описаны первые шаги по работе с системой.

Документ предназначен для пользователей, имеющих права администратора системы и доступ к системным настройкам аппаратного обеспечения системы (системных администраторов).

Данное руководство разработано для Combox версии 2.0

Данное руководство не является универсальным, но может являться основой для установки и обновления сервисов.

1. Список используемых аббревиатур и сокращений

APP	Application / Приложение
DB	Database / База данных
SSH	Secure shell / Безопасная оболочка
ЛКМ	Клик левой кнопкой мыши

2. Системные требования

Количество необходимых серверов для корректного функционирования системы зависит от конкретной задачи.

В данном документе для простоты при описании действий администратора предполагается, что сервер один.

Рекомендуемое минимальное количество серверов: 3.

При таком подходе предлагается разделить их назначение следующим образом:

- Сервер 1: APP сервер для установки приложений
- Сервер 2: DB сервер для установки PostgreSQL
- Сервер 3: FTP сервер

В качестве инструмента для администрирования баз данных рекомендуется использовать pgAdmin (<https://www.pgadmin.org/download/>), но допускается также использование других аналогичных инструментов.

2.1. Минимальные системные требования

К аппаратному и программному обеспечению для функционирования системы предъявляются следующие требования:

- Сервера приложений:
 - Сервер 1:
Аппаратное обеспечение: 8 CPU, 16 GB RAM, 50 GB HDD;
Программное обеспечение: Docker;
Назначение: ключевые модули системы ("Фабрика данных", "Apache Airflow", "Graylog" и другие).
- Сервера СУБД:
 - Сервер 2:
Аппаратное обеспечение: 4 CPU, 8 GB RAM, 50 GB HDD;
Программное обеспечение: Postgres;
Назначение: СУБД системы ("Фабрика данных", справочники, словари и тд).
- Сервера FTP:
 - Сервер 3:
Аппаратное обеспечение: 2 CPU, 2 GB RAM, 50 GB HDD;
Программное обеспечение: SFTPGo;
Назначение: SFTP системы ("Фабрика данных").

2.2. Рекомендуемые системные требования

К аппаратному и программному обеспечению для полноценного функционирования системы предъявляются следующие требования:

- Сервера приложений:

- Сервер 1:
Аппаратное обеспечение: 4 CPU, 12 GB RAM, 50 GB HDD;
Программное обеспечение: Docker;
Назначение: ключевые модули системы ("Фабрика данных" и другие).
- Сервер 2:
Аппаратное обеспечение: 4 CPU, 8 GB RAM, 50 GB HDD;
Программное обеспечение: Docker;
Назначение: ключевые модули системы ("Apache Airflow" и другие).
- Сервер 3:
Аппаратное обеспечение: 8 CPU, 8 GB RAM, 50 GB HDD;
Программное обеспечение: Graylog;
Назначение: нативно развернутые модули системы "Graylog".
- Сервера СУБД:
 - Сервер 4:
Аппаратное обеспечение: 4 CPU, 8 GB RAM, 50 GB HDD;
Программное обеспечение: Postgres;
Назначение: СУБД системы ("Фабрика данных", справочники, словари и тд).
- Сервера FTP:
 - Сервер 5:
Аппаратное обеспечение: 2 CPU, 2 GB RAM, 50 GB HDD;
Программное обеспечение: SFTPGo;
Назначение: SFTP системы ("Фабрика данных").

3. Необходимое программное обеспечение и состав дистрибутива системы

Для корректной работы платформа Comбуx требует наличия следующего предустановленного ПО:

Программное обеспечение	Рекомендуемая версия
Docker	20.10.21
PostgreSQL	14.8
SFTPGo	2.4.0

Состав дистрибутива:

Название сервиса системы	Используемая версия
Comбуx	2.4.4
Redis	7.0.5-alpine

Audit Logger	3.3.0
Apache Airflow	2.5.1
Graylog	4.3

Используемый APP сервер должен иметь доступ к сети Интернет.

4. Подготовка инфраструктуры

4.1. Создание папок

- Создание папки для использования контейнерами производится командой:

```
mkdir <название_папки>
```

Пример: `mkdir test`

- Перечень папок для создания:

`/certificate` используется для хранения сертификата, используемого в контейнерах Combox и Audit Logger

- Удаление папки производится командой:

```
rmdir <название_папки> если папка пустая
```

```
rm -Rf <название_папки> если папка содержит файлы
```

Пример: `rmdir test` ИЛИ `rm -Rf test`

4.2. Установление SSH соединение серверов

Для установки системы необходимо разместить на APP сервере Docker-образы. Перед началом миграции образов и файлов между локальным и удаленным сервером необходимо установить SSH соединение.

- На локальном сервере генерируется ssh ключ алгоритмом ed25519 с указанием пароля:

```
ssh-keygen -t ed25519
```

- На локальном сервере осуществляется переход в папку `.ssh` из корневой директории:

```
cd .ssh/
```

- Необходимо скопировать с локального сервера строку открытого ключа из `id_ed25519.pub` на удаленный сервер в `.ssh/authorized_keys`

Локальный сервер:

```
nano id_ed25519.pub -> COPY TEXT -> CLOSE (Ctrl + X)
```

Удаленный сервер:

```
cd .ssh/
```

```
nano authorized_keys -> PASTE TEXT -> SAVE (Ctrl + O)
```

4.3. Миграция образов

Одним из вариантов передачи образов на удаленный сервер является использование команд на локальном сервере.

- Необходимо скачать образ из Harbor на локальный сервер

```
docker pull <исходный образ>:<тег>
```

Пример: `docker pull registry.dmbasis.ru/combyx/userinterfaceui:2.4.1`

- Переименовывается и создается новый тег для образа (при необходимости)

```
docker image tag <исходный образ>:<тег> <целевой образ>:<тег>
```

Пример: `docker image tag registry.dmbasis.ru/combyx/userinterfaceui:2.4.1 combyx/userinterfaceui:2.4.1`

- Необходимо передать образ на удаленный сервер, используя ранее сгенерированный SSH ключ

```
docker save <исходный образ>:<тег> | ssh -i ~/.ssh/id_ed25519 <логин>@<сервер> -p <номер порта> 'docker load'
```

Пример: `docker save combyx/userinterfaceui:2.4.1 | ssh -i ~/.ssh/id_ed25519 combyx@185.35.145.84 -p 22022 'docker load'`

4.4. Миграция Compose и конфигурационных файлов

- Одним из вариантов передачи файлов на удаленный сервер является использование команды на локальном сервере:

```
scp -r <путь к директории>/* <логин>@<сервер>:/<куда копировать>
```

Пример: `scp -r test/* combyx@combyx-qa-app.dmbasis.ru:/folder`

- Также возможно ручное создание файлов, используя текстовый редактор с последующим копированием/вставкой содержимого:

```
nano <название файла>.yaml Создание YAML файла
```

```
nano <название файла>.conf Создание файла конфигурации
```

Пример: `nano combyx.yaml`

4.5. Редактирование Compose и конфигурационных файлов

- Перед началом развертывания системы в Compose и конфигурационных файлах необходимо обновить переменные значения:

Пример:

```
combyx.yaml >> image: ${MIGRATION_IMAGE}
```

```
conf.env >> MIGRATION_IMAGE=combyx/migration:2.4.1
```

- Одним из вариантов редактирования Compose и конфигурационного файла является использование текстового редактора `nano`:

`nano <название файла>` открываем файл в редакторе `nano`, вносим правки в текст

`Ctrl + O` сохранить изменения

`Ctrl + X` выйти из редактора

`Ctrl + X, N` выйти из редактора без сохранения, если были внесены изменения

Более расширенные возможности можно прочитать в официальной документации <https://www.nano-editor.org/dist/latest/nano.html>

4.6. Примеры Compose и конфигурационных файлов для сервисов системы

- В представленных примерах синим цветом выделены необязательные параметры, красным цветом выделены параметры для замены на требуемые.

4.6.1. Combyx

4.6.1.1. Compose файл

COMBYX

version: "3.9"

services:

migration:

image: \${MIGRATION_IMAGE}

deploy:

restart_policy:

condition: none

environment:

-

ConnectionStrings__DefaultConnection=\${CONNECTION_STRING_MAIN}\${WITH_POOLING}

ui-backend:

image: \${UI_BACKEND_IMAGE}

deploy:

restart_policy:

condition: any

environment:

- TZ=Europe/Moscow

- ASPNETCORE_ENVIRONMENT=Development

- ASPNETCORE_URLS=https://+:443;http://+:80

- ASPNETCORE_Kestrel__Certificates__Default__Path=\${CERT_PATH}

-

ASPNETCORE_Kestrel__Certificates__Default__Password=\${CERT_PASSWORD}

-

ConnectionStrings__DefaultConnection=\${CONNECTION_STRING_MAIN}\${WITH_POOLING}

- LockMaster__ConnectionString=\${CONNECTION_STRING_MAIN}

- Redis__Host=\${REDIS_HOST}\${ABORT_CONNECT}

- Redis__RefreshInterval=00:10:00

- Serilog__Enrich__0=FromLogContext

- Serilog__Enrich__1=WithMachineName

- Serilog__Enrich__2=WithThreadId

- Serilog__MinimumLevel__Default=Information

-

Serilog__MinimumLevel__Override__Microsoft.AspNetCore.Mvc.Infrastructure=Warning

-

Serilog__MinimumLevel__Override__Microsoft.EntityFrameworkCore.Database.Command=Warning

- Serilog__MinimumLevel__Override__Microsoft.EntityFrameworkCore.Query=Error

- Serilog__MinimumLevel__Override__System=Warning

-

Serilog__MinimumLevel__Override__Microsoft.AspNetCore.Hosting.Diagnostics=Warning

- Serilog__Using__0=Serilog.Sinks.Async
- Serilog__WriteTo__0__Args__configure__0__Name=Console
- Serilog__WriteTo__0__Name=Async

Serilog__WriteTo__1__Args__configure__0__Args__facility=\${UI_BACKEND_FACILITY}

Serilog__WriteTo__1__Args__configure__0__Args__hostnameOrAddress=\${HTTP_PROTOCOL}\${GELF_ENDPOINT}\${GELF_DIRECTORY}

- Serilog__WriteTo__1__Args__configure__0__Args__port=\${GELF_PORT}
- Serilog__WriteTo__1__Args__configure__0__Args__transportType=Http
- Serilog__WriteTo__1__Args__configure__0__Name=Graylog
- Serilog__WriteTo__1__Name=Async
- SystemSecurityKey=\${SECURITY_KEY}
- RequestLogging__IgnorePaths__0=/source-sample/.*

Logging__LogLevel__Microsoft.EntityFrameworkCore.Database.Command=Warning

- RequestLogging__Ignore__Paths__1=/healthcheck
- AuditGrpcService__Url=\${AUDIT_LOGGER_URL}
- Airflow__Host=\${AIRFLOW_HOST}
- Airflow__UserName=\${AIRFLOW_USER}
- Airflow__Password=\${AIRFLOW_PASSWORD}

volumes:

- "\${CERTIFICATE_VOLUME}"

ports:

- "\${UI_BACKEND_HTTPS_PORT}:443"

ui-frontend:

image: \${UI_FRONTEND_IMAGE}

deploy:

restart_policy:
condition: any

environment:

- TZ=Europe/Moscow
- APP_SERVER_URL=\${UI_BACKEND_URL}
- APP_PROCESS_MANAGER_URL=\${PYPROCESS_MANAGER_URL}

ports:

- "\${UI_FRONTEND_HTTPS_PORT}:443"

pyprocess-manager:

image: \${PYPROCESS_MANAGER_IMAGE}

deploy:

replicas: 1

resources:

limits:

cpus: "1"

memory: 512M

restart_policy:

condition: any

environment:

- FLASK_HOST=0.0.0.0
- FLASK_PORT=4050

- API_ENDPOINT=\${AIRFLOW_HOST}
- API_TOKEN=\${AIRFLOW_TOKEN}
- DB_USER=\${POSTGRES_USER}
- DB_PASSWORD=\${POSTGRES_PASSWORD}
- DB_HOST=\${POSTGRES_SERVER}
- DB_PORT=\${POSTGRES_PORT}
- DB_DATABASE=\${POSTGRES_DATABASE}
- DB_SCHEMA=\${POSTGRES_SCHEMA}
- DB_CHUNKSIZE=100000
- DB_MINCONN=5
- DB_MAXCONN=20
- REDIS_HOST=\${REDIS_HOST}
- REDIS_PORT=\${REDIS_PORT}
- FTP_HOST=\${FTP_HOST}
- FTP_PORT=\${FTP_PORT}
- FTP_USERNAME=\${FTP_USER}
- FTP_PASSWORD=\${FTP_PASSWORD}
- FTP_FOLDER_PATH=\${FTP_FOLDER}

directory-manager:

image: \${DIRECTORY_MANAGER_IMAGE}

deploy:

replicas: 1

resources:

limits:

cpus: "1"

memory: 512M

restart_policy:

condition: any

environment:

- AllowedHosts=*

-

ConnectionString__DefaultConnection=\${CONNECTION_STRING_MAIN}\${WITH_POOLING}

- DirectoryManagementOptions__MaxInputValuesSize=500
- KestrelHost__Http1Port=5206
- KestrelHost__Http2Port=5006
- KestrelHost__Http2PortHttps=5106
- Logging__LogLevel__Default=Information
- Logging__LogLevel__Microsoft.AspNetCore=Warning
- Logging__LogLevel__Microsoft.EntityFrameworkCore=Warning

ports:

- "\${DIRECTORY_MANAGER_HTTPS_PORT}:5106"

4.6.1.2. Файл конфигурации

CONF

```
-----
# IMAGES
-----
```

REDIS_IMAGE=redis:7.0.5-alpine
MIGRATION_IMAGE=[combyx/migration:2.4.3](#)
UI_BACKEND_IMAGE=[combyx/userinterfaceapi:2.4.3](#)
UI_FRONTEND_IMAGE=[combyx/userinterfaceui:2.4.3](#)
DIRECTORY_MANAGER_IMAGE=[combyx/directorymanager:2.4.3](#)
PYPROCESS_MANAGER_IMAGE=[combyx/pyprocess.manager:2.4.2-rc1](#)
AUDIT_LOGGER_IMAGE=[audit_logger/audit.logger:3.3.0](#)
DOMAIN_VALIDATION_IMAGE=[combyx/domainvalidation:2.4.3](#)

POSTGRES CONNECTION

CONNECTION_STRING_MAIN=Server=[pg_host](#);Port=[pg_port](#);Database=[pg_database](#)
;UserId=[pg_username](#);Password=[pg_password](#);
WITH_POOLING=Pooling=False
POSTGRES_SERVER=[pg_host](#)
POSTGRES_PORT=[pg_port](#)
POSTGRES_SCHEMA=[pg_scheme](#)
POSTGRES_DATABASE=[pg_database](#)
POSTGRES_USER=[pg_username](#)
POSTGRES_PASSWORD=[pg_password](#)

REDIS CONNECTION

REDIS_HOST=redis
ABORT_CONNECT=,abortConnect=false
REDIS_PORT=6379

AIRFLOW CONNECTION

AIRFLOW_HOST=[airflow_host](#)
AIRFLOW_USER=[airflow_username](#)
AIRFLOW_PASSWORD=[airflow_password](#)
AIRFLOW_TOKEN=[airflow_token](#)

GRAYLOG

HTTP_PROTOCOL=http://
HTTPS_PROTOCOL=https://
GELF_ENDPOINT=[graylog_host](#)
GELF_DIRECTORY=/gelf
GELF_PORT=[gelf_port](#)
AUDIT_LOGGER_GELF=[audit_logger_gelf_port](#)

GRAYLOG_USER=**graylog_username**
GRAYLOG_PASSWORD=**graylog_password**

UI_BACKEND_FACILITY=UI-API_COMBYX

OTHER SERVICES CONFIG

UI_BACKEND_URL=<https://ui-backend>
PYPROCESS_MANAGER_URL=<http://pyprocess-manager:4050/>
AUDIT_LOGGER_URL=<https://audit-logger-tcp:5107>

UI_BACKEND_HTTPS_PORT=15082
UI_FRONTEND_HTTPS_PORT=15083
DIRECTORY_MANAGER_HTTPS_PORT=5106
AUDIT_LOGGER_HTTPS_PORT=5107

CERT_PATH=/app/https/**certificate.pfx**
CERT_PASSWORD=**certificate_password**
SECURITY_KEY=**key**

FTP_HOST=**ftp_host**
FTP_PORT=**ftp_port**
FTP_USER=**ftp_username**
FTP_PASSWORD=**ftp_password**
FTP_FOLDER=./test

VOLUMES

CERTIFICATE_VOLUME=**/certificate_path**:/app/https

4.6.2. Redis

4.6.2.1. Compose файл

REDIS

version: "3.9"

services:

redis:

image: \${REDIS_IMAGE}

deploy:

replicas: 1

restart_policy:

condition: any

environment:

- TZ=Europe/Moscow

4.6.2.2. Файл конфигурации

Аналогично как для Comбух.

4.6.3. Audit Logger

4.6.3.1. Compose файл

AUDIT_LOGGER

version: "3.9"

services:

audit-logger-tcp:

image: \${AUDIT_LOGGER_IMAGE}

deploy:

replicas: 1

resources:

limits:

cpus: "1"

memory: 512M

restart_policy:

condition: any

environment:

- ASPNETCORE_Kestrel__Certificates__Default__Path=\${CERT_PATH}

-

ASPNETCORE_Kestrel__Certificates__Default__Password=\${CERT_PASSWORD}

- AllowedHosts=*

- Logging__LogLevel__Default=Information

- Logging__LogLevel__Microsoft.AspNetCore=Warning

- GrayLogRestOptions__Host=\${HTTPS_PROTOCOL}\${GELF_ENDPOINT}

- GrayLogRestOptions__Username=\${GRAYLOG_USER}

- GrayLogRestOptions__Password=\${GRAYLOG_PASSWORD}

- KestrelHost__Http1Port=5207

- KestrelHost__Http2Port=5007

- KestrelHost__Http2PortHttps=5107

- AuditLogger__ServiceType=Redis

- AuditLogger__ServiceSettings__Host=\${REDIS_HOST}

- AuditLogger__ServiceSettings__GroupName=redis-auditlogger-tcp

- AuditLogger__ServiceSettings__StreamName=redis-auditlogger-tcp

- AuditLogger__ServiceSettings__MessageCount=1000

- AuditLogger__ClientType=TCP

- AuditLogger__ClientSettings__Host=\${GELF_ENDPOINT}

- AuditLogger__ClientSettings__Port=\${AUDIT_LOGGER_GELF}

volumes:

- "\${CERTIFICATE_VOLUME}"

ports:

- "\${AUDIT_LOGGER_HTTPS_PORT}:5107"

4.6.3.2. Файл конфигурации

Аналогично как для Comбух.

4.6.4. Apache Airflow

4.6.4.1. Compose файл

APACHE AIRFLOW

```
# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements. See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership. The ASF licenses this file
# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
# with the License. You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing,
# software distributed under the License is distributed on an
# "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
# KIND, either express or implied. See the License for the
# specific language governing permissions and limitations
# under the License.
#

# Basic Airflow cluster configuration for CeleryExecutor with Redis and PostgreSQL.
#
# WARNING: This configuration is for local development. Do not use it in a production
# deployment.
#
# This configuration supports basic configuration using environment variables or an .env
# file
# The following variables are supported:
#
# AIRFLOW_IMAGE_NAME          - Docker image name used to run Airflow.
#                               Default: apache/airflow:2.5.1
# AIRFLOW_UID                 - User ID in Airflow containers
#                               Default: 50000
# AIRFLOW_PROJ_DIR            - Base path to which all the files will be volumed.
#                               Default: .
# Those configurations are useful mostly in case of standalone testing/running Airflow in
# test/try-out mode
#
# _AIRFLOW_WWW_USER_USERNAME  - Username for the administrator account
# (if requested).
#                               Default: airflow
# _AIRFLOW_WWW_USER_PASSWORD  - Password for the administrator account (if
# requested).
#                               Default: airflow
# _PIP_ADDITIONAL_REQUIREMENTS - Additional PIP requirements to add when
# starting all containers.
#                               Default: "
#
# Feel free to modify this file to suit your needs.
---
version: '3'
```

```

x-airflow-common:
  &airflow-common
  # In order to add custom dependencies or upgrade provider packages you can use
  your extended image.
  # Comment the image line, place your Dockerfile in the directory where you placed the
  docker-compose.yaml
  # and uncomment the "build" line below, Then run `docker-compose build` to build the
  images.
  image: ${AIRFLOW_IMAGE_NAME:-apache/airflow:2.5.1}
  # build: .
  environment:
    &airflow-common-env
    AIRFLOW__CORE__EXECUTOR: CeleryExecutor
    AIRFLOW__DATABASE__SQL_ALCHEMY_CONN:
  postgresql+psycopg2://airflow:airflow@postgres/airflow
    # For backward compatibility, with Airflow <2.3
    AIRFLOW__CORE__SQL_ALCHEMY_CONN:
  postgresql+psycopg2://airflow:airflow@postgres/airflow
    AIRFLOW__CELERY__RESULT_BACKEND: db+postgresql://airflow:airflow@postgres/airflow
    AIRFLOW__CELERY__BROKER_URL: redis://:@redis:6379/0
    AIRFLOW__CORE__FERNET_KEY: "
    AIRFLOW__CORE__DAGS_ARE_PAUSED_AT_CREATION: 'true'
    AIRFLOW__CORE__LOAD_EXAMPLES: 'false'
    AIRFLOW__API__AUTH_BACKENDS:
  'airflow.api.auth.backend.basic_auth,airflow.api.auth.backend.session'
    _PIP_ADDITIONAL_REQUIREMENTS: ${_PIP_ADDITIONAL_REQUIREMENTS:-}
    AIRFLOW__WEBSERVER__EXPOSE_CONFIG: 'true'
    AIRFLOW__CORE__DEFAULT_TIMEZONE: Europe/Moscow
    AIRFLOW__CORE__PARALLELISM: '16'
  volumes:
    - ${AIRFLOW_PROJ_DIR:-.}/dags:/opt/airflow/dags
    - ${AIRFLOW_PROJ_DIR:-.}/logs:/opt/airflow/logs
    - ${AIRFLOW_PROJ_DIR:-.}/plugins:/opt/airflow/plugins
  user: "${AIRFLOW_UID:-50000}:0"
  depends_on:
    &airflow-common-depends-on
    redis:
      condition: service_healthy
    postgres:
      condition: service_healthy

services:
  postgres:
    image: postgres:13
    environment:
      POSTGRES_USER: airflow
      POSTGRES_PASSWORD: airflow
      POSTGRES_DB: airflow
    volumes:
      - postgres-db-volume:/var/lib/postgresql/data
    healthcheck:
      test: ["CMD", "pg_isready", "-U", "airflow"]

```

interval: 5s
retries: 5
restart: always

redis:

image: redis:latest
expose:
- 6379
healthcheck:
test: ["CMD", "redis-cli", "ping"]
interval: 5s
timeout: 30s
retries: 50
restart: always

airflow-webserver:

<<: *airflow-common
command: webserver
ports:
- 8080:8080
healthcheck:
test: ["CMD", "curl", "--fail", "<http://localhost:8080/health>"]
interval: 10s
timeout: 10s
retries: 5
restart: always
depends_on:
<<: *airflow-common-depends-on
airflow-init:
condition: service_completed_successfully

airflow-scheduler:

<<: *airflow-common
command: scheduler
healthcheck:
test: ["CMD-SHELL", 'airflow jobs check --job-type SchedulerJob --hostname
"\${HOSTNAME}"]
interval: 10s
timeout: 10s
retries: 5
restart: always
depends_on:
<<: *airflow-common-depends-on
airflow-init:
condition: service_completed_successfully

airflow-worker:

<<: *airflow-common
command: celery worker
healthcheck:
test:
- "CMD-SHELL"

```

- 'celery --app airflow.executors.celery_executor.app inspect ping -d
"celery@${HOSTNAME}"
  interval: 10s
  timeout: 10s
  retries: 5
environment:
  <<: *airflow-common-env
  # Required to handle warm shutdown of the celery workers properly
  # See https://airflow.apache.org/docs/docker-stack/entrypoint.html#signal-propagation
  DUMB_INIT_SETSID: "0"
restart: always
depends_on:
  <<: *airflow-common-depends-on
airflow-init:
  condition: service_completed_successfully

airflow-triggerer:
  <<: *airflow-common
  command: triggerer
  healthcheck:
    test: ["CMD-SHELL", 'airflow jobs check --job-type TriggererJob --hostname
"${HOSTNAME}"]
    interval: 10s
    timeout: 10s
    retries: 5
  restart: always
  depends_on:
    <<: *airflow-common-depends-on
  airflow-init:
    condition: service_completed_successfully

airflow-init:
  <<: *airflow-common
  entrypoint: /bin/bash
  # yamllint disable rule:line-length
  command:
    - -C
    - |
      function ver() {
        printf "%04d%04d%04d%04d" ${1//./ }
      }
      airflow_version=${$(AIRFLOW__LOGGING__LOGGING_LEVEL=INFO && gosu
airflow airflow version)}
      airflow_version_comparable=${$(ver ${airflow_version})}
      min_airflow_version=2.2.0
      min_airflow_version_comparable=${$(ver ${min_airflow_version})}
      if (( airflow_version_comparable < min_airflow_version_comparable )); then
        echo
        echo -e "\033[1;31mERROR!!!: Too old Airflow version ${airflow_version}!\e[0m"
        echo "The minimum Airflow version supported: ${min_airflow_version}. Only use
this or higher!"
        echo

```

```

    exit 1
fi
if [[ -z "${AIRFLOW_UID}" ]]; then
    echo
    echo -e "\033[1;33mWARNING!!!: AIRFLOW_UID not set!\e[0m"
    echo "If you are on Linux, you SHOULD follow the instructions below to set "
    echo "AIRFLOW_UID environment variable, otherwise files will be owned by
root."
    echo "For other operating systems you can get rid of the warning with manually
created .env file:"
    echo " See: https://airflow.apache.org/docs/apache-airflow/stable/howto/docker-
compose/index.html#setting-the-right-airflow-user"
    echo
fi
one_meg=1048576
mem_available=$((($(getconf _PHYS_PAGES) * $(getconf PAGE_SIZE) /
one_meg))
cpus_available=$(grep -cE 'cpu[0-9]+' /proc/stat)
disk_available=$(df / | tail -1 | awk '{print $4}')
warning_resources="false"
if (( mem_available < 4000 )) ; then
    echo
    echo -e "\033[1;33mWARNING!!!: Not enough memory available for
Docker.\e[0m"
    echo "At least 4GB of memory required. You have $(numfmt --to iec
$(mem_available * one_meg))"
    echo
    warning_resources="true"
fi
if (( cpus_available < 2 )) ; then
    echo
    echo -e "\033[1;33mWARNING!!!: Not enough CPUS available for Docker.\e[0m"
    echo "At least 2 CPUs recommended. You have ${cpus_available}"
    echo
    warning_resources="true"
fi
if (( disk_available < one_meg * 10 )) ; then
    echo
    echo -e "\033[1;33mWARNING!!!: Not enough Disk space available for
Docker.\e[0m"
    echo "At least 10 GBs recommended. You have $(numfmt --to iec
$(disk_available * 1024 ))"
    echo
    warning_resources="true"
fi
if [[ ${warning_resources} == "true" ]]; then
    echo
    echo -e "\033[1;33mWARNING!!!: You have not enough resources to run Airflow
(see above)!\e[0m"
    echo "Please follow the instructions to increase amount of resources available:"
    echo " https://airflow.apache.org/docs/apache-airflow/stable/howto/docker-compose/index.html#before-
you-begin"

```

```
    echo
  fi
  mkdir -p /sources/logs /sources/dags /sources/plugins
  chown -R "${AIRFLOW_UID}:0" /sources/{logs,dags,plugins}
  exec /entrypoint airflow version
# yamllint enable rule:line-length
environment:
  <<: *airflow-common-env
  _AIRFLOW_DB_UPGRADE: 'true'
  _AIRFLOW_WWW_USER_CREATE: 'true'
  _AIRFLOW_WWW_USER_USERNAME:
${_AIRFLOW_WWW_USER_USERNAME:-airflow}
  _AIRFLOW_WWW_USER_PASSWORD:
${_AIRFLOW_WWW_USER_PASSWORD:-airflow}
  user: "0:0"
  volumes:
  - ${AIRFLOW_PROJ_DIR:-.}/sources
```

airflow-cli:

```
<<: *airflow-common
profiles:
  - debug
environment:
  <<: *airflow-common-env
  CONNECTION_CHECK_MAX_COUNT: "0"
# Workaround for entrypoint issue. See: https://github.com/apache/airflow/issues/16252
command:
  - bash
  - -c
  - airflow
```

You can enable flower by adding "--profile flower" option e.g. docker-compose --profile flower up

or by explicitly targeted on the command line e.g. docker-compose up flower.

See: <https://docs.docker.com/compose/profiles/>

flower:

```
<<: *airflow-common
command: celery flower
profiles:
  - flower
ports:
  - 5555:5555
healthcheck:
  test: ["CMD", "curl", "--fail", "http://localhost:5555/"]
  interval: 10s
  timeout: 10s
  retries: 5
restart: always
depends_on:
  <<: *airflow-common-depends-on
  airflow-init:
  condition: service_completed_successfully
```

volumes:
 postgres-db-volume:

4.6.4.2. Файл конфигурации

не требуется

4.6.5. Graylog

4.6.5.1. Compose файл

GRAYLOG

version: '3.9'

services:

 # MongoDB: https://hub.docker.com/_/mongo/

 mongo:

 image: mongo:4.2

 deploy:

 replicas: 1

 restart_policy:

 condition: any

 volumes:

 - ./data/[mongo/data/db](#)

 environment:

 - TZ=Europe/Moscow

 # Elasticsearch: <https://www.elastic.co/guide/en/elasticsearch/reference/7.10/docker.html>

 elasticsearch:

 image: elasticsearch:7.17.5

 environment:

 - TZ=Europe/Moscow

 - discovery.type=single-node

 - http.host=0.0.0.0

 - transport.host=localhost

 - network.host=0.0.0.0

 - "ES_JAVA_OPTS=-Dlog4j2.formatMsgNoLookups=true -Xms1024m -

Xmx1024m"

 - xpack.security.enabled=false

 volumes:

 - ./data/[els/usr/share/elasticsearch/data](#)

 ulimits:

 memlock:

 soft: -1

 hard: -1

 deploy:

 resources:

 limits:

 memory: 1g

 # Graylog: <https://hub.docker.com/r/graylog/graylog/>

 graylog:

```
image: graylog/graylog:4.2
environment:
  - TZ=Europe/Moscow
  - GRAYLOG_TIMEZONE=Europe/Moscow
  - GRAYLOG_ROOT_TIMEZONE=Europe/Moscow
  # CHANGE ME (must be at least 16 characters)!
  - GRAYLOG_PASSWORD_SECRET=password
  # Password: admin
  -
GRAYLOG_ROOT_PASSWORD_SHA2=8c6976e5b5410415bde908bd4dee15dfb167a
9c873fc4bb8a81f6f2ab448a918
  - GRAYLOG_HTTP_EXTERNAL_URI=http://127.0.0.1:9000/
entrypoint: /usr/bin/tini -- wait-for-it elasticsearch:9200 -- /docker-entrypoint.sh
deploy:
  replicas: 1
  restart_policy:
    condition: any
depends_on:
  - mongo
  - elasticsearch
ports:
  # Graylog web interface and REST API
  - 9000:9000
  # Syslog TCP
  - 1514:1514
  # Syslog UDP
  - 1514:1514/udp
  # GELF TCP
  - gelf_port:gelf_port
  - audit_logger_gelf_port:audit_logger_gelf_port
  # GELF UDP
  - gelf_port:gelf_port/udp
  - audit_logger_gelf_port:audit_logger_gelf_port/udp
volumes:
  data:
```

4.6.5.2. Файл конфигурации

не требуется

4.6.6. Email domain validation

4.6.6.1. Compose файл

DOMAIN VALIDATION

version: "3.9"

services:

domain-validation:

image: \${DOMAIN_VALIDATION_IMAGE}

deploy:

replicas: 1

```
resources:
  limits:
    cpus: "1"
    memory: 500M
  restart_policy:
    condition: any
environment:
  - TZ=Europe/Moscow
```

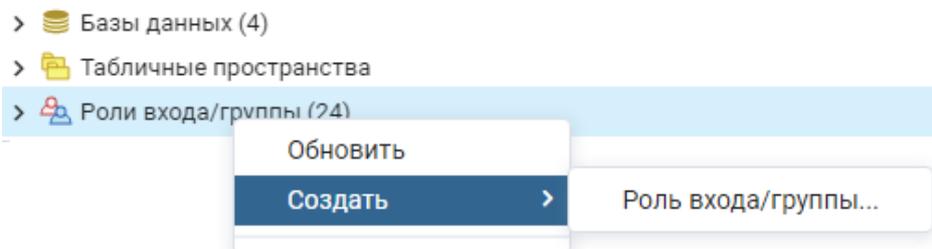
4.6.6.2. Файл конфигурации

Аналогично как для Comбух.

4.7. Создание ролей и пользователей (DB PostgreSQL)

Описание ниже приведено для случая использования pgAdmin (скачать можно по ссылке: <https://www.pgadmin.org/download/>)

- Необходимо подключиться к серверу, используя учетные данные администратора.
- Далее ПКМ на вкладку Роли входа/группы, выбирать пункт Создать - Роль входа/группы...



- Создается группа для владельцев базы данных: на вкладке General необходимо указать Имя группы и далее нажать на кнопку Сохранить.

Создание Роль входа/группы

General **Определение** Права Членство Параметры Безопасность SQL

Имя

Комментарии

- Необходимо создать пользователя, который войдет в группу владельцев, для этого повторяются два предыдущих пункта:
 - на вкладке **General** необходимо указать **Имя** пользователя;

Создание Роль входа/группы

General **Определение** Права Членство Параметры Безопасность SQL

Имя

Комментарии

- - на вкладке **Определение** необходимо указать **Пароль** для входа;

Создание Роль входа/группы

General **Определение** Права Членство Параметры Безопасность SQL

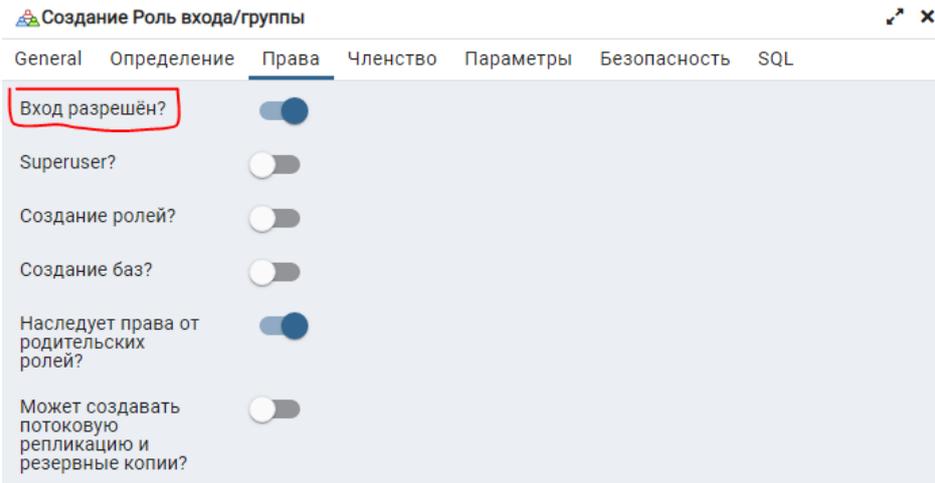
Пароль

Роль активна до

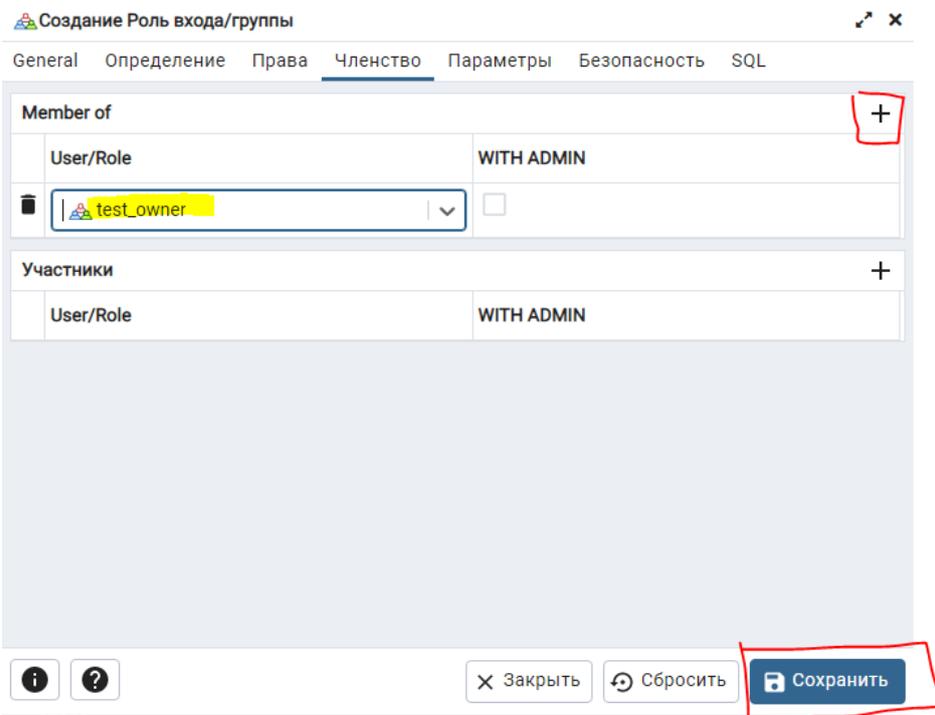
Please note that if you leave this field blank, then password will never expire.

Макс. число подключений

- - на вкладке **Права** необходимо выбрать значение "Да" для пункта **Вход разрешен?**;

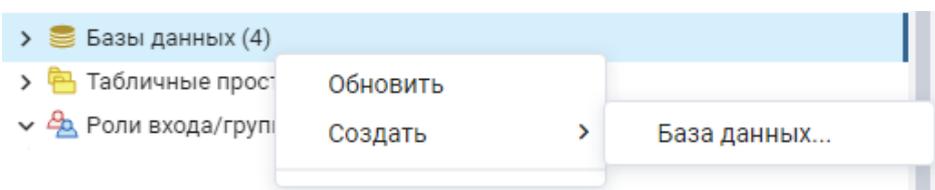


- на вкладке **Членство/Member of** необходимо нажать на кнопку + (Add row), в **User/Role** указывается ранее созданная группа, и далее необходимо нажать на кнопку **Сохранить**.



4.8. Создание баз данных

- Необходимо подключиться к серверу, используя учетные данные администратора.
- Далее ПКМ на вкладку **Базы данных**, выбрать **Создать - База данных...**



- Необходимо создать базу данных: на вкладке **General** необходимо указать название в пункте **База данных**, затем выбрать в пункте **Владелец** ранее созданную группу владельцев, после чего нажать на кнопку **Сохранить**.

Создание База данных

General | Определение | Безопасность | Параметры | Дополнительно | SQL

База данных: test

Владелец: test_owner

Комментарий

Закреть | Сбросить | Сохранить

- Отключаемся от сервера и повторно подключаемся от имени созданного ранее пользователя предварительно внося в параметры подключения имя пользователя.

combyx-rc-db-2.dmbasis.ru

combyx-rc-db-2.dmbasis.ru

General | Соединение | SSL | SSH Tunnel | Дополнительно

Имя/адрес сервера: combyx-rc-db-2.dmbasis.ru

Порт: 5432

Служебная база данных: postgres

Имя пользователя: atkachev

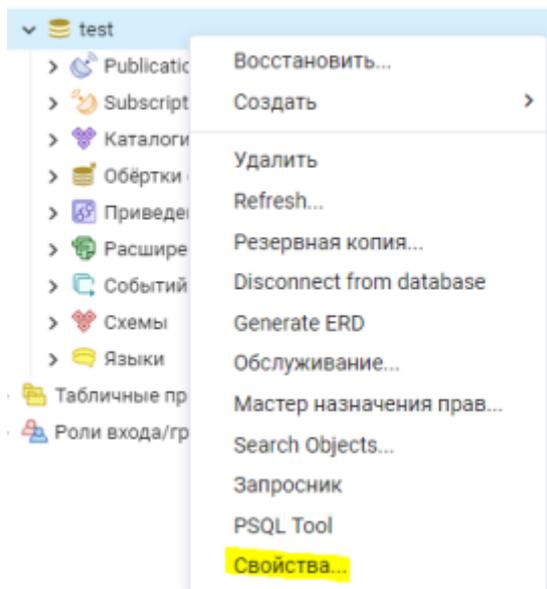
Керберос аутентификация?

Роль: test_user

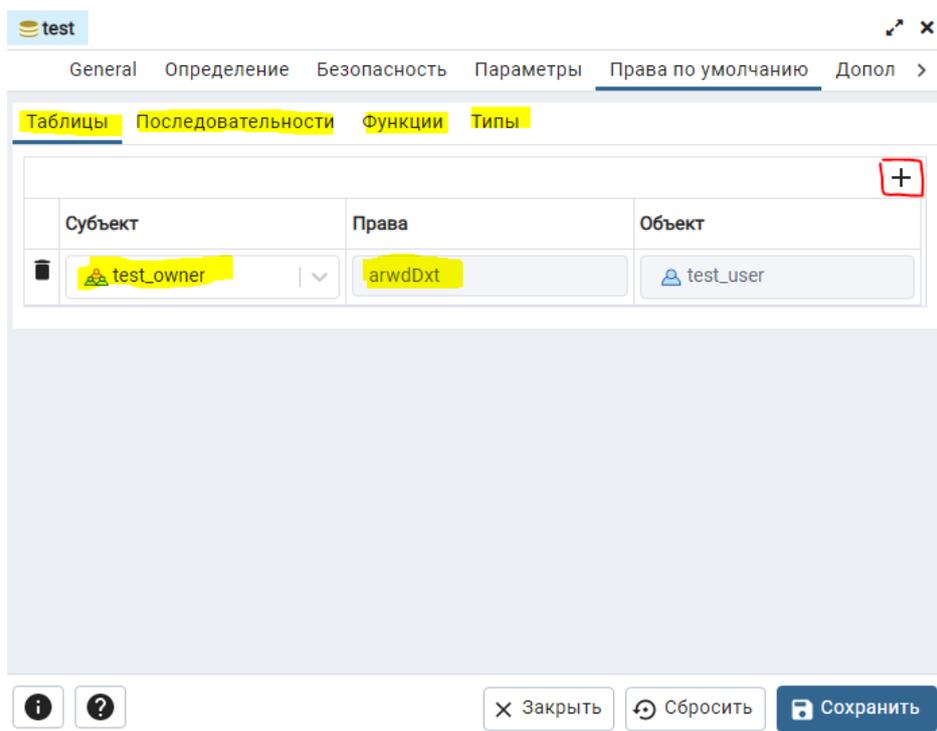
Service

Закреть | Сбросить | Сохранить

- Необходимо выполнить ПКМ на созданную базу данных и выбрать пункт **Свойства**.



- Необходимо перейти во вкладку **Права по умолчанию**, и во вкладках **Таблицы**, **Последовательности**, **Функции** И **Типы** **НАЖИМАТЬ + (Add row)**.
- В качестве **Субъект** выбирается ранее созданная группа владельцев, в качестве **Права** выбираются все возможные права.



5. Установка сервисов

5.1. Установка Combox

- Одним из вариантов установки сервисов Combox на сервере происходит при помощи команды:

```
env $(cat <файл конфигураций> | grep ^[A-Z] | xargs) docker stack deploy -c <Compose файл> <название стека>
```

Пример: `env $(cat conf.env | grep ^[A-Z] | xargs) docker stack deploy -c combyx.yml combyx`

5.2. Установка Apache Airflow

- При необходимости сервис можно разместить в стеке с Combyx, либо разместить на другом сервере.
- Рекомендуемое название Compose файла: `docker-compose.yml`.
- Установить `docker-compose`:

```
sudo apt install docker-compose
```

- Создать папки. Перечень папок для создания:

```
/dags           используется для DAG
```

```
/logs           используется для логов
```

```
/plugins        используется для плагинов
```

- Установка Airflow производится из директории с его Compose файлом при помощи команды:

```
docker-compose up -d
```

- Дополнительно необходимо в контейнер сервиса установить библиотеку:

```
docker exec -it airflow-docker_airflow-worker_1 /bin/bash
```

 Вход в контейнер

```
pip install openpyxl
```

 Установка библиотеки openpyxl

```
Ctrl + P + Q
```

 Выход из контейнера

5.3. Установка и настройка Graylog

- При необходимости сервис можно разместить в стеке с Combyx, либо разместить на другом сервере. Есть возможность использовать нативно развернутую версию - установка производится согласно официальной документации.
- Установка Graylog производится из директории с его Compose файлом при помощи команды:

```
docker stack deploy -c <Compose файл> <название стека>
```

Пример: `docker stack deploy -c graylog.yml combyx`

- После установки необходимо перейти в интерфейс, затем авторизоваться, используя учетные данные, и выбрать в меню пункт **System / Inputs - Inputs**.
- На открывшейся вкладке нажать на раскрывающееся меню **Select input**, выбрать **GELF TCP**.
- В открывшемся окне указать **Port**, и (при необходимости) выбрать текущую ноду в **Node**.
- Получившийся Input необходимо использовать для связки с параметром **GELF_PORT=gelf_port** в конфигурационном файле.
- При необходимости повторить данную процедуру, задав новый порт для параметра **AUDIT_LOGGER_GELF=audit_logger_gelf_port**, если требуется использование нового порта.

6. Обновление или перезапуск сервисов

6.1. Описание базовых команд

- В случае, если необходимо обновить сервисы, то изменения в Compose и конфигурационный файл вносятся при помощи текстового редактора, например, nano (см. пункт 4.5).
- Для обновления сервисов необходимо передать на удаленный сервер новые образы (см. пункт 4.3).
- Одним из вариантов обновления или перезапуска сервисов Combyx и Graylog на сервере происходит при помощи команд:

```
docker service rm <название сервиса>
```

```
env $(cat <файл конфигураций> | grep ^[A-Z] | xargs) docker stack deploy -c <Compose файл> <название стека>
```

Пример:

```
docker service rm combyx_migration
```

```
env $(cat conf.env | grep ^[A-Z] | xargs) docker stack deploy -c combyx.yml combyx
```

- Для перезапуска сервисов Airflow используется команда:

```
docker-compose down && docker-compose up -d
```

7. Smoke тестирование функционала сервисов

7.1. Проверка сервисов Combyx

- Migration:

- Проверка миграции схем и таблиц: после запуска сервиса должны появиться схемы `core`, `fl`, `structure`, `us` и `voc` с заполненными таблицами.
- Проверка миграции обновлений таблиц: после перезапуска сервиса должны обновиться схемы/таблицы/столбцы в зависимости от обновления.
- UI Interface и Backend:
 - Проверка работы сайта: перейти на страницу интерфейса и посетить различные страницы, при этом не должно выпадать никаких ошибок взаимодействия с другими микросервисами.
 - Проверка создания подключения к FTP: используя функционал `Администрирование - Соединение` создать новое или войти в уже имеющееся подключение и нажать `Проверить соединение`.

8. Действия при возможных неисправностях

8.1. Действия в случае несоблюдения условий выполнения технологического процесса, в том числе при длительных отказах технических средств

Если те или иные действия в системе Combox были выполнены с несоблюдением условий выполнения технологического процесса и привели к аварийной ситуации, рекомендуется произвести попытку сохранения редактируемых данных с целью избежать потери данных.

При возникновении любых неполадок при работе с системой, которые не могут быть самостоятельно устранены пользователем, необходимо обращаться в службу поддержки разработчика системы. При этом необходимо описать признаки аварийной ситуации и действия, которые были выполнены пользователем непосредственно перед возникновением аварийной ситуации.

При неисправности технических средств, входящих в состав системы Combox, необходимо задействовать исправное техническое средство взамен неисправного.

8.2. Действия по восстановлению программ и/или данных при отказе магнитных носителей или обнаружении ошибок в данных

При отказе жесткого диска компьютера или сервера необходимо задействовать другой компьютер/сервер или жесткий диск и переустановить на него систему согласно описанию в данном документе.

Данные, утерянные вследствие отказа магнитного носителя сервера системы, восстанавливаются из копии этих данных на сервере. Для восстановления данных

при отказе магнитного носителя сервера системы необходимо после переустановки и настройки приложения перезапустить приложение Combox.

При обнаружении ошибок в настройках системы Combox или необходимого стороннего ПО следует исправить ошибки в настройках в соответствии с данным документом.

8.3. Действия в случаях обнаружении несанкционированного вмешательства в данные

В случае обнаружения несанкционированного вмешательства в данные системы необходимо обратиться в службу технической поддержки разработчика системы. При этом необходимо описать признаки и предполагаемый характер вмешательства, указать перечень затронутых данных и быть готовым по требованию службы технической поддержки описать признаки аварийной ситуации и действия, которые были выполнены пользователем непосредственно перед возникновением аварийной ситуации.

8.4. Действия в других аварийных ситуациях

8.4.1. Возможные аварийные ситуации

1. Отказ базы данных;
2. Отказ GrayLog;
3. Отказ фабрики данных Apache Airflow;
4. Отказ Docker.

8.4.2. Необходимые действия

1. Перезапуск контейнеров;
2. При возникновении любых неполадок при работе с системой, которые не могут быть самостоятельно устранены администратором, необходимо обращаться в службу поддержки разработчика системы. При этом необходимо описать признаки аварийной ситуации и действия, которые были выполнены пользователем непосредственно перед возникновением аварийной ситуации.

9. Администрирование системы

Для целей администрирования системы в интерфейсе системы предназначен специализированный раздел "Администрирование".

Раздел системы "Администрирование" доступен для входа только для тех пользователей системы, которым назначена роль "Администратор". В данный раздел пользователь может перейти из Главного меню.

В разделе "Администрирование" пользователям доступен следующий функционал:

- Просмотр базовой (агрегированной) информации о работоспособности системы (вкладка "Состояние системы");
- Детальный просмотр состояния работоспособности системы и ее отдельных модулей (вкладка "Журнал системных сообщений");
- Создание и редактирование данных пользователей системы, а также назначение им соответствующих ролей (вкладка "Управление пользователями").

9.1. Вкладка "Состояние системы"

Первая вкладка, открывается автоматически при входе в раздел "Администрирование". Пользователь видит на экране список ключевых параметров системы (совокупность этих параметров обычно называют "здоровьем системы", а процесс мониторинга этих параметров - "проверкой здоровья системы").

9.1.1. Особенности экрана "Состояние системы"

Экран "Состояние системы" располагается на отдельной вкладке "Состояние системы" раздела "Администрирование". Данный экран предназначен для мониторинга работоспособности Системы администраторами. Он представляет собой единый отчет ("дашборд"), в котором на одном экране отображается ключевая информация о состоянии системы и ее отдельных модулей (сервисов).

Администраторы имеют возможность быстро оценить общую ситуацию ("все работает в штатном режиме" / "есть проблемы"), а также возможность детально ознакомиться с текущими процессами и их статусом. При этом базовая агрегированная информация представлена на экране "Состояние системы", а с деталями пользователь может ознакомиться на экране "Журнал системных сообщений" (посмотреть "логи" системы). При этом данные экраны связаны - при клике на интересующих показателях на экране "Состояние системы" пользователь переключается на соответствующий отчет с детальной информацией по определенному процессу обработки данных.

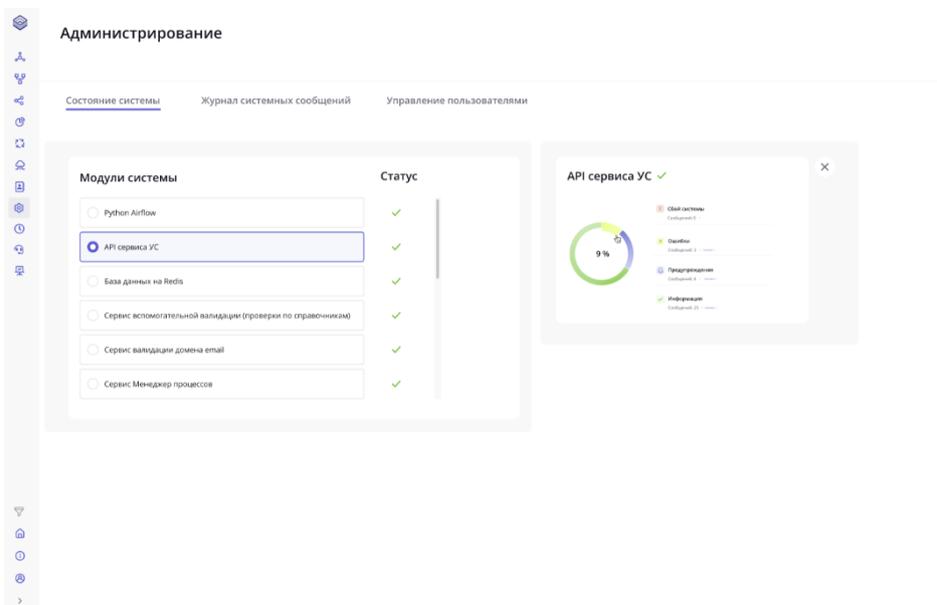
Информация на данном экране относится к программной инфраструктуре (работоспособность сервисов, работает ли очередь сообщений и тп). Информирование об общем состоянии системы представлено в виде графического отчета (дашборда), который включает в себя все ключевые показатели работы системы. Каждый ключевой показатель представляет собой "виджет", демонстрирующий некоторый отдельный показатель или совокупность показателей в рамках одной темы, с пояснением - все в порядке (ОК на зеленом фоне) или нет (соответствующее сообщение).

Некоторые ключевые показатели являются ссылкой, кликнув на которую Администратор попадает на экран с более детальной информацией относительно выбранного показателя или проблемы.

9.1.2. Отображаемые модули системы (виджеты)

- Python Airflow;
- API сервиса УС;

- База данных на PostgreSQL (основное хранилище);
- База данных на Redis;
- Сервис вспомогательной валидации (проверки по справочникам);
- Сервис валидации домена email;
- Сервис Менеджер процессов;
- Сервис Процессор витрин;
- Очередь сообщений (RabbitMQ);
- Сервис логирования (Graylog).



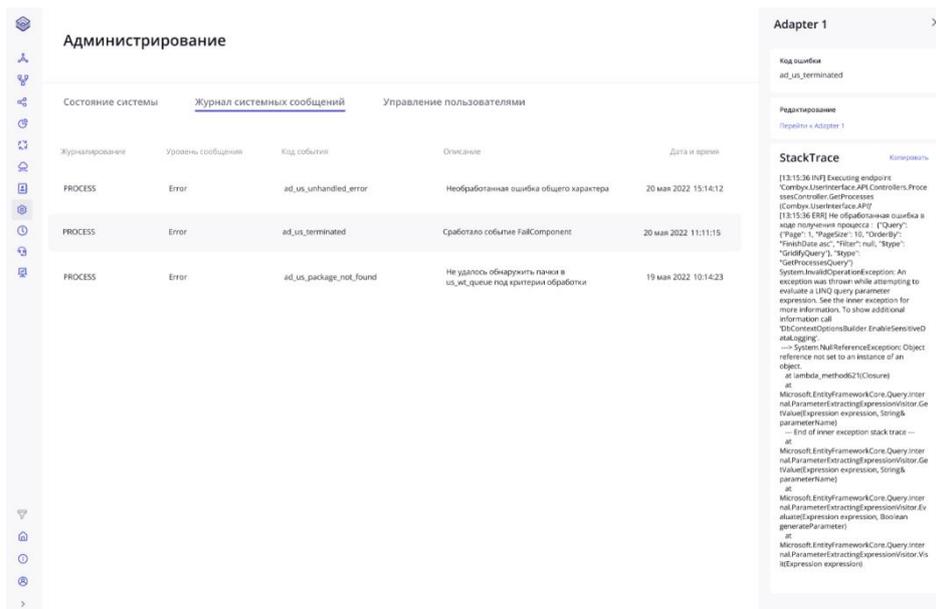
9.2. Вкладка "Журнал системных сообщений"

9.2.1. Особенности экрана "Журнал системных сообщений"

Экран "Журнал системных сообщений" располагается на отдельной вкладке "Журнал системных сообщений" раздела "Администрирование". Данный экран предназначен для детального мониторинга работоспособности системы администраторами.

У администраторов есть возможность детально ознакомиться с текущими процессами и их статусом. Т.е. если проблема возникла в каком-то модуле системы, информирование об этом предполагается как на уровне целого модуля, так и на уровне самого нижнего доступного уровня процессов. Другими словами, у администраторов системы появляется возможность узнать о проблеме, и максимально быстро локализовать ее.

Экран "Журнал системных сообщений" является интерфейсом пользователя (администратора) для просмотра системных сообщений, зафиксированных в Graylog. Структура отображения информации на данном экране соответствует записанным данным в Graylog.



9.3. Вкладка "Управление пользователями"

9.3.1. Особенности экрана "Управление пользователями"

Экран "Управление пользователями" располагается на отдельной вкладке "Управление пользователями" раздела "Администрирование". Данный экран предоставляет функционал для создания, редактирования и удаления пользователей системы, а также для назначений ролей пользователям.

Пользователь с правами администратора системы может выполнить следующие действия:

- - просмотреть список всех пользователей системы;
 - просмотреть все имеющиеся данные о любом пользователе системы (кроме пароля);
 - создать нового пользователя, включая назначение для него аутентификационных данных и роли(ей);
 - изменить данные существующего пользователя, включая его аутентификационные данные и роль(и);
 - удалить любого пользователя системы (кроме себя и Главного администратора);

Система позволяет хранить следующий набор данных о пользователе:

- - ID пользователя;
 - Фамилия пользователя;
 - Имя пользователя;
 - Отчество пользователя;
 - Подразделение/служба/департамент/отдел пользователя;
 - Должность пользователя;
 - Email пользователя;
 - Мобильный телефон пользователя;

- Флаг подписки на оповещения системы;
- Роли пользователя (список всех ролей);
- Логин пользователя;
- Пароль пользователя (зашифрованный);
- Дата регистрации пользователя;
- Дата последней смены пароля.

В системе можно назначить пользователям следующие роли (один пользователь может иметь несколько ролей, при этом описанные ниже роли заранее созданы):

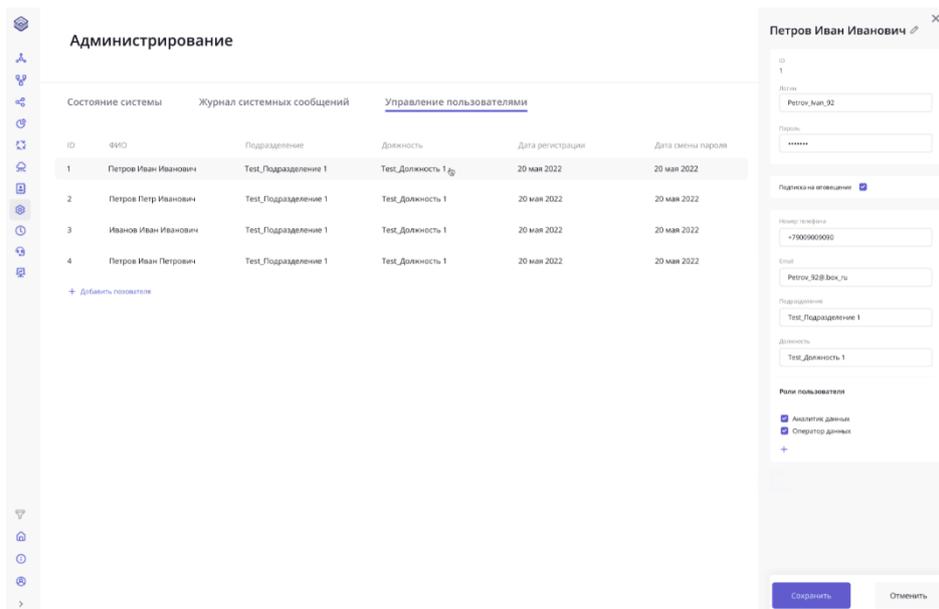
- Администратор системы (уровень доступа 1):
Администратор выполняет настройку системы, осуществляет управление правами доступа пользователей, контролирует действия пользователей системы, проводит обновление конфигурации системы.
Администратор обладает максимально полным списком возможностей по контролю системы в целом, а также всех ее пользователей.
- Аналитик данных (уровень доступа 2):
Аналитику данных не доступны функции управления системой, его роль предполагает доступ к данным только в режиме чтения с целью их анализа.
- Оператор данных (уровень доступа 3):
Оператору данных доступны функции по прямому управлению данными - фильтрация, добавление, редактирование и удаление. Большинство модулей системы ему недоступны, а функционал ограничен только работой с данными.

При выборе вкладки отобразится реестр пользователей системы. Для отображения подробной информации о пользователе необходимо лкм на строке с его данными:

The image shows two parts of a web application interface. On the left is a table titled 'Администрирование' with a sub-tab 'Управление пользователями'. The table has columns: ID, ФИО, Подразделение, Должность, Дата регистрации, and Дата смены пароля. It contains four rows of user data. On the right is a modal window for editing the user 'Петров Иван Иванович'. It shows fields for ID (1), Login (Петров_Иван_ИИ), Password (masked), a checked 'Подписка на оповещения' checkbox, phone number (+79000000000), email (Петров_ИИИ@Box.ru), and roles 'Аналитик данных' and 'Оператор данных' (both checked). There is a 'Управление ролями >' link and 'Редактировать' / 'Отменить' buttons at the bottom.

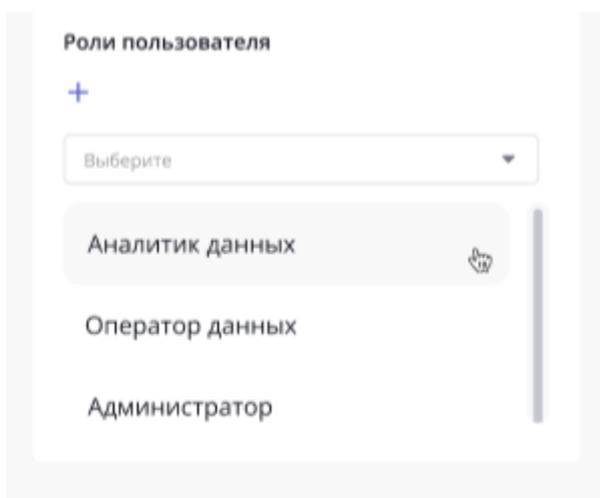
ID	ФИО	Подразделение	Должность	Дата регистрации	Дата смены пароля
1	Петров Иван Иванович	Тест_Подразделение 1	Тест_Должность 1	20 мая 2022	20 мая 2022
2	Петров Петр Иванович	Тест_Подразделение 1	Тест_Должность 2	20 мая 2022	20 мая 2022
3	Иванов Иван Иванович	Тест_Подразделение 1	Тест_Должность 3	20 мая 2022	20 мая 2022
4	Петров Иван Петрович	Тест_Подразделение 1	Тест_Должность 3	20 мая 2022	20 мая 2022

Для редактирования информации о пользователе необходимо лкм на иконке "карандаш" рядом с его ФИО:



В режиме редактирования пользователя можно изменить его данные, а также список назначенных ему ролей.

Для добавления роли необходимо лкм на иконку плюс "+" в разделе Роли пользователя, выбрать роль, которую требуется добавить:



Для удаления роли необходимо лкм на флаге (галочке) рядом с ранее добавленной ролью.

После внесения необходимых изменений для обновления информации о пользователе необходимо нажать на кнопку **Сохранить**.

Для создания нового пользователя системы необходимо лкм на кнопку "+ Добавить пользователя" под реестром пользователей.

Отобразится панель для ввода информации о новом пользователе:

Имя пользователя



Логин

Пароль

Подписка на оповещение

Номер телефона

Email

Подразделение

Должность

Роли пользователя



Выберите 

Аналитик данных 

Оператор данных

Администратор

Сохранить

Отменить

После ввода всей необходимой информации о пользователе необходимо нажать на кнопку "Сохранить".

При нажатии на кнопке "Отмена" произойдет отмена процесса создания нового пользователя.

10. Начало работы с системой

Сразу после установки и настройки системы по умолчанию создан только один пользователь - Главный администратор.

Данный пользователь имеет максимальные права при работе в системе, и при этом он не может быть удален.

Заданные первоначальные аутентификационные данные Главного администратора:

- логин: admin
- пароль: admin

При открытии главной страницы системы необходимо ввести указанные аутентификационные данные.

При первом входе в систему Главный администратор обязан сменить свой пароль в разделе системы "Администрирование".

Остальные пользователи системы создаются Главным администратором, либо другим пользователем с правами Администратора (такого пользователя также должен создать Главный администратор).

Каждый новый пользователь получает свои аутентификационные данные от администратора системы в том режиме и через те каналы коммуникаций, которые используются на данном предприятии для передачи конфиденциальных данных.