

**Программный комплекс для решения задач экспресс-анализа работы релейной защиты и автоматики на базе программного обеспечения «PF.Protection» с функциями интеграции с системой сбора неоперативной технологической информации для диспетчерских центров (АСА РЗА) для нужд АО «СО ЕЭС»**

**ИНСТРУКЦИЯ ПО УСТАНОВКЕ И НАСТРОЙКЕ**

Москва, 2025

Инв. № подл.	Подп. и дата	Взам. инв. №	2019РТС_Д0328								
			Изм.	Кол.у	Лист	№	Подп.	Дата	Стадия	Лист	Листов
Разраб.	Иванов								Э	1	25
Пров.	Иванов								АО «РТСофт»		
ГИП	Иванов								АО «РТСофт»		

АСА РЗА  
Инструкция по установке и  
настройке на испытательном стенде

# ОГЛАВЛЕНИЕ

<b>ТЕРМИНЫ, ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ .....</b>	<b>4</b>
<b>Условные обозначения.....</b>	<b>5</b>
<b>1 ВВЕДЕНИЕ.....</b>	<b>6</b>
1.1 Комплект поставки .....	6
1.2 Последовательность установки.....	7
<b>2 ПОДГОТОВКА К УСТАНОВКЕ .....</b>	<b>7</b>
2.1 Действия персонала перед началом установки ПО .....	9
<b>3 УСТАНОВКА И НАСТРОЙКА.....</b>	<b>9</b>
3.1 Системное ПО .....	9
3.1.1 Database сервер .....	9
3.1.1.1 Redis.....	9
3.1.1.2 PostgreSQL .....	10
3.1.2 Backend сервер.....	11
3.1.2.1 Подготовка.....	11
3.1.2.2 Docker & Docker Compose .....	11
3.1.3 Integration сервер .....	12
3.1.3.1 Подготовка.....	12
3.1.3.2 Docker & Docker Compose.....	13
3.1.4 Frontend сервер .....	13
3.1.4.1 Подготовка.....	13
3.1.4.2 SSL сертификат .....	13
3.1.4.3 NGINX.....	14
3.2 Сервисное ПО .....	15
3.2.1 Backend сервер.....	15
3.2.2 Integration сервер .....	17
3.2.3 Frontend сервер .....	18
<b>4 ОБНОВЛЕНИЕ.....</b>	<b>20</b>
4.1 Database сервер .....	20
4.2 Backend сервер .....	20
4.3 Integration сервер.....	21

Инв.	№	Подп. и дата	Взам. инв. №

4.4	Frontend сервер.....	22
<b>5</b>	<b>ПЕРЕНОС ДАННЫХ .....</b>	<b>22</b>
5.1	Подготовка .....	22
5.2	Базы данных .....	22
5.3	Тома.....	24

Инв. №	Подп. и дата	Взам. инв. № _____

Изм.	Кол.у	Лист	№	Подп.	Дата

## ТЕРМИНЫ, ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ

Сокращение	Расшифровка
ACA РЗА	Программный комплекс для решения задач экспресс-анализа работы релейной защиты и автоматики на базе программного обеспечения «PF.Protection» с функциями интеграции с системой сбора неоперативной технологической информации для диспетчерских центров (ACA РЗА) для нужд АО «СО ЕЭС»
БД	База данных
ОЗУ	Оперативное запоминающее устройство
ОС	Операционная система
ПО	Программное обеспечение
ЦПУ	Центральное процессорное устройство

Инв.	№	Подп. и дата	Взам. инв. №

Изм.	Кол.	у	Лист	№	Подп.	Дата

ACA РЗА  
Инструкция по установке и настройке  
на испытательном стенде

Лист 4

## УСЛОВНЫЕ ОБОЗНАЧЕНИЯ

**Синим** цветом выделены команды, могут быть целиком скопированы и вставлены в эмулятор терминала Linux.

**Красным** цветом выделены фрагменты, нуждающиеся в замене на правильные, например, имена файлов, адреса серверов и т.д.

Инв. №	Подп. и дата	Взам. инв. №

Изм.	Кол.у	Лист	№	Подп.	Дата

АСА РЗА  
Инструкция по установке и настройке  
на испытательном стенде

Лист  
5

# 1 ВВЕДЕНИЕ

## 1.1 Комплект поставки

Установочные пакеты для каждого сервера перечислены в таблице 1.

Таблица 1. Перечень пакетов установки

Сервер	Пакет	Описание	Состав	
			Директория	Файл
Database	-	-	-	-
Backend	back.tar.gz	Сжатый архив формата tar.gz.	back	.env - переменные среды docker-compose.yml - перечень сервисов для развертывания
			-	daemon.json - настройки Docker
Integration	int.tar.gz		cim	.env - переменные среды docker-compose.yml - перечень сервисов для развертывания healthcheckservice_settings.json - настройки конкретного сервиса
			int	.env - переменные среды aipintegration_appsettings.json - настройки конкретного сервиса aipintegration_editablesettings.json - настройки конкретного сервиса docker-compose.yml - перечень сервисов для развертывания khisubscriptionservice_appsettings.Development.json - настройки конкретного сервиса khisubscriptionservice_appsettings.json - настройки конкретного сервиса khisubscriptionservice_editablesettings.json - настройки конкретного сервиса krb5.conf - конфигурация Kerberos для аутентификации

№	Подп. и дата	Взам. инв. №

Извм.	Кол.у	Лист	№	Подп.	Дата

				signalsrestintegrationservice_ediatablessettings.json - настройки конкретного сервиса
			-	SQLExecConfig.json - перечень SQL-скриптов для миграций БД
Frontend	front.tar.gz		-	daemon.json - настройки Docker
			-	default.conf - специфическая конфигурация проекта для NGINX
				nginx.conf - базовая конфигурация NGINX
		html		* - файлы интерфейса

## 1.2 Последовательность установки

Установка производится в следующем порядке:

- Database сервер;
- Backend сервер;
- Integration сервер;
- Frontend сервер.

## 2 ПОДГОТОВКА К УСТАНОВКЕ

Все виртуальные или физические сервера должны иметь подключение к сети и доступ к репозиторио с необходимым в ходе установки и настройки ПО.

ПО устанавливается на сервера, подготовленные в соответствии с требованиями, указанными в таблице 2.

Таблица 2. Конфигурация оборудования

Инв. №	№ Подп. и дата	Взам. №	Сервер						Требования					
			1	2	3	4	5	6	7	8	9	10	11	12
			Database сервер											
			Backend сервер											
			Integration сервер											

Изм.	Кол.у	Лист	№	Подп.	Дата	АСА РЗА Инструкция по установке и настройке на испытательном стенде	Лист
							7

	Диск - 200 GB
Frontend сервер	ЦПУ - 2 ядра ОЗУ - 4 GB Диск - 100 GB

Перечень требуемого системного ПО представлен в таблице 3.

Табл. 3 Программная конфигурация - системное ПО

Сервер	ПО
Database сервер	ОС Astra Linux СУБД Redis СУБД PostgreSQL
Backend сервер	ОС Astra Linux ПО Docker ПО Docker Compose
Integration сервер	ОС Astra Linux  ПО Docker ПО Docker Compose
Frontend сервер	ОС Astra Linux  ПО NGINX

В ходе настройки и установки системы на сервера будут установлены сервисы, перечисленные в таблице 4.

Таблица 4. Программная конфигурация – сервисное ПО

Сервер	ПО
Database сервер	-
Backend сервер	asm-backend asm-backend-ws asm-backend-celery asm-backend-celery-exchange asm-backend-celery-scheduler asm-backend-celery-email-worker asm-backend-celery-readiness-index asm-backend-celery-analysis-worker asm-backend-celery-pf-modeling-worker asm-backend-celery-integration-worker asm-backend-celery-oscillogram-gc-worker asm-backend-celery-scorpion-parser-worker

Инв. №	Подп. и дата	Взам. инв. №

Изм.	Кол.у	Лист	№	Подп.	Дата

Integration сервер	cimservice asmpublisher stateservice repairsservice accountservice scorplogservice  mopop-linux aipintegrationservice khisubscriptionservice signalsrestintegrationservice
Frontend сервер	-

## 2.1 Действия персонала перед началом установки ПО

Все действия по установке проводятся под системной учетной записью с правами администратора.

Архивы перед установкой и настройкой стоит размещать в директории `/home/имя_пользователя`.

В системе должен быть установлен любой редактор текста. По умолчанию в семействе операционных систем GNU/Linux, как правило, присутствует редактор Vi и nano (последний рекомендуется). От пользователя требуются базовые навыки работы с одним из них или любым другим, установленным в системе.

Для установки программного обеспечения, поставляемого в виде инструкций по запуску при помощи Docker и Docker Compose, необходимо наличие доступа в репозиторий в сети Интернет, либо к локальной копии репозитория.

# 3 УСТАНОВКА И НАСТРОЙКА

## 3.1 Системное ПО

### 3.1.1 Database сервер

#### 3.1.1.1 Redis

- Обновить список доступных пакетов программного обеспечения:

```
sudo apt update
```

- Установить Redis:

Инв. №	Подп. и дата	Взам. инв. №
Изм.	Кол.у	Лист

Изм.	Кол.у	Лист	№	Подп.	Дата

**`sudo apt install -y redis-server`**

3. Изменить файл `/etc/redis/redis.conf` любым редактором (например, `nano /etc/redis/redis.conf`):

- Стока 87:

bind 0.0.0.0 -::1

- Стока 111:

protected-mode no

- Стока 327 (необходимо раскомментировать - убрать # в начале):

supervised systemd

4. Перезапустить службу Redis:

**`sudo systemctl restart redis`**

5. Проверить статус службы Redis:

**`sudo systemctl status redis`**

Вывод статуса службы Redis, если активен – Redis готов принимать входящие соединения.

### **3.1.1.2 PostgreSQL**

1. Подключить и настроить репозитории Postgres.

2. Обновить список доступных пакетов программного обеспечения:

**`sudo apt update`**

3. Установить пакеты для работы OpenSSL и корневые сертификаты:

**`sudo apt install openssl ca-certificates`**

4. Установить PostgreSQL и Python пакет для него:

**`sudo apt -y install postgresql postgresql-plpython`**

5. Изменить файл `/etc/postgresql/main/postgresql.conf` любым редактором (например, `nano /etc/postgresql/main/postgresql.conf`):

- Стока 60 (необходимо раскомментировать - убрать # в начале):

listen\_addresses = '0.0.0.0'

6. Изменить файл `/etc/postgresql/main/pg_hba.conf` любым редактором (например, `nano /etc/postgresql/main/pg_hba.conf`):

№	Подп. и дата	Взам. №
Инв.		

Изм.	Кол.у	Лист	№	Подп.	Дата

- Стока 97:

```
host all all 0.0.0.0/0 scram-sha-256
```

7. Перезапустить службу PostgreSQL:

```
sudo systemctl restart postgresql
```

8. Проверить статус службы PostgreSQL:

```
sudo systemctl status postgresql
```

Вывод статуса службы PostgreSQL, если активен – PostgreSQL готов принимать входящие соединения.

9. Изменить пароль для пользователя postgres:

```
psql
```

```
ALTER USER postgres WITH PASSWORD '...';
```

10. Создать базу данных:

```
CREATE DATABASE asmdb;
```

Подсказка для выхода из psql и пользователя postgres:

```
\q
```

```
exit
```

Настроены данные для входа, создана нужная база данных. Системное ПО на database сервере настроено. Database сервер настроен.

### 3.1.2 Backend сервер

#### 3.1.2.1 Подготовка

1. Разместить в домашней директории пользователя (/home/**имя\_пользователя**) архив back.tar.gz.
2. Создать директорию установки:

```
sudo mkdir -p /opt/rtsoft/asa
```

3. Распаковать архив:

```
sudo tar -xzf back.tar.gz -C /opt/rtsoft/asa
```

#### 3.1.2.2 Docker & Docker Compose

1. Создать каталог для конфигурации Docker реестра:

№	Подп. и дата	Взам. №
Изм.	Кол.у	Лист

`sudo mkdir /etc/docker`

2. Переместить файл daemon.json (из комплекта поставки) в /etc/docker:

`sudo mv daemon.json /etc/docker`

3. Изменить файл /etc/docker/daemon.json любым редактором (например, `nano /etc/docker/daemon.json`):

- Стока 2:

"insecure-registries": ["**адрес\_docker\_реестра**"]

4. Обновить список доступных пакетов программного обеспечения:

`sudo apt update`

5. Установить Docker и Docker Compose:

`sudo apt install -y docker.io docker-compose`

6. Проверить версию:

`docker --version`

`docker-compose --version`

8. Создать Docker сеть для контейнеров:

`docker network create asm_net`

9. Создать Docker тома для контейнеров:

`docker volume create osc_fs`

`docker volume create static_fs`

`docker volume create media_fs`

9. Пройти аутентификацию в Docker реестре:

`docker login -u имя_пользователя -p пароль адрес_docker_реестра`

Docker настроен и готов к работе. Системное ПО на backend сервере настроено.

### 3.1.3 Integration сервер

#### 3.1.3.1 Подготовка

1. Разместить в домашней директории пользователя (`/home/имя_пользователя`) архив int.tar.gz.

Инв. №	Подп. и дата	Взам. инв. №
Изм.	Кол.у	Лист №

Изм.	Кол.у	Лист №	Подп.	Дата

2. Создать директорию установки:  

```
sudo mkdir -p /opt/rtsoft/asa
```
3. Распаковать архив:  

```
sudo tar -xzf int.tar.gz -C /opt/rtsoft/asa
```

### **3.1.3.2 Docker & Docker Compose**

1. Провести установку и настройку аналогично backend сервера.

Docker настроен и готов к работе. Системное ПО на integration сервере настроено.

### **3.1.4 Frontend сервер**

#### **3.1.4.1 Подготовка**

1. Разместить в домашней директории пользователя (/home/**имя\_пользователя**) архив front.tar.gz.
2. Создать директорию установки:  

```
sudo mkdir -p /opt/rtsoft/asa
```
3. Распаковать архив:  

```
sudo tar -xzf front.tar.gz -C /opt/rtsoft/asa
```

#### **3.1.4.2 SSL сертификат**

1. Разместить в домашней директории пользователя (/home/**имя\_пользователя**) файл .pfx (не входит в комплект поставки).
2. Обновить список доступных пакетов программного обеспечения:

```
sudo apt update
```

3. Установить пакеты для работы OpenSSL и корневые сертификаты:

```
sudo apt install -y openssl ca-certificates
```

4. Распаковать сертификат:  

```
openssl pkcs12 -in имя_файла_сертификата.pfx -clcerts -nokeys -out public.crt  
openssl pkcs12 -in имя_файла_сертификата.pfx -nodes -nocerts -out private.key
```
5. Разместить публичную часть сертификата и приватный ключ в /etc/ssl:  

```
sudo mv public.crt /etc/ssl  
sudo mv private.key /etc/ssl
```

Инв. №	Подп. и дата	Взам. инв. №
Изм.	Кол.у	Лист №

Изм.	Кол.у	Лист №	Подп.	Дата	

```
sudo mv private.key /etc/ssl
```

### 3.1.4.3 NGINX

1. Обновить список доступных пакетов программного обеспечения:

```
sudo apt update
```

2. Установить NGINX:

```
sudo apt install -y nginx
```

3. Проверить статус службы NGINX:

```
sudo systemctl status nginx
```

Вывод статуса службы NGINX, если активен – NGINX может обрабатывать запросы.

Кроме того, можно подключиться к серверу через браузер, вписав в поисковую строку его адрес. Браузер отобразит приветственную страницу NGINX.

## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

4. Разместить файл базовой конфигурации nginx.conf (из комплекта поставки) в /etc/nginx:

```
sudo mv nginx.conf /etc/nginx
```

5. Разместить файл расширенной конфигурации default.conf (из комплекта поставки) в /etc/nginx/conf.d:

```
sudo mv default.conf /etc/nginx/conf.d
```

6. Изменить файл /etc/nginx/nginx.conf:

- Стока 1:

user имя\_пользователя\_системы;

7. Изменить файл /etc/nginx/conf.d/default.conf:

- Стока 3:

Инв. №	№ Подп. и дата	Взам. инв. №
Изм.	Кол.у	Лист

Изм.	Кол.у	Лист	№	Подп.	Дата

server\_name адрес\_frontend\_сервера;

- Стока 14:

resolver адрес\_dns\_сервера valid=30s;

- Стока 16:

set \$int адрес\_integration\_сервера;

- Стока 17:

set \$db адрес\_database\_сервера;

- Стока 18:

set \$back адрес\_backend\_сервера;

Следующие строки менять, только если распаковка сертификата происходила в файлы, отличные от public.crt и private.key:

- Стока 11:

ssl\_certificate /etc/ssl/public.crt;

- Стока 12:

ssl\_certificate\_key /etc/ssl/private.key;

## 8. Перезапустить NGINX:

`sudo nginx -s reload`

Также перезапуск можно осуществить командой (вариант выше предпочтительнее):

`sudo systemctl restart nginx`

NGINX настроен и готов к работе. Системное ПО на frontend сервере настроено. Системное ПО настроено.

## 3.2 Сервисное ПО

### 3.2.1 Backend сервер

#### 1. Перейти в директорию back:

`cd /opt/rtsoft/asa/back`

#### 2. Изменить файл .env любым редактором (например, `nano .env`):

- Стока 1:

Инв. №	Подп. и дата	Взам. №
Изм.	Кол.у	Лист

Изм.	Кол.у	Лист	№	Подп.	Дата

REGISTRY\_NAME=адрес\_docker\_реестра

- Стока 8:

DB\_HOST=адрес\_хоста\_баз\_данных

- Стока 11 - установить пароль, ранее использовавшийся при настройке PostgreSQL на database сервере в разделе настоящего руководства:

DB\_PASS=пароль\_базы\_данных

- Стока 13:

RD\_HOST=адрес\_хоста\_баз\_данных

- Стока 30:

PFCLLOUD\_API\_URL=https://адрес\_хоста\_PF.P:порт

- Стока 31:

DELIVERY\_API\_URL=https://адрес\_хоста\_PF.P:порт

- Стока 40:

FILE\_UPLOAD\_URL="https://адрес\_frontend\_сервера/api/relays/junkyardgate/collector"

- Стока 41:

CONFIG\_UPLOAD\_URL="https://адрес\_frontend\_сервера/api/relay\_configuration/"

- Запустить контейнеры:

**docker-compose up -d**

- Проверить запуск контейнеров через 10 минут с ввода последней команды:

**docker ps -a**

Контейнеры имеют статус Up (healthy). Если часть контейнеров имеет статус Created – вновь выполнить **docker-compose up -d** и повторять до тех пор, пока статус всех контейнеров не будет Up (healthy). Если какой-либо контейнер в статусе Exited – произошла ошибка. Рекомендуется проверить настройки подключений к БД.

Сервисное ПО на backend сервер настроено. Настройка backend сервера завершена.

Инв. №	№ Подп. и дата	Взам. инв. №

Изм.	Кол.у	Лист	№	Подп.	Дата

### 3.2.2 Integration сервер

- Перейти в директорию cim:

```
cd /opt/rtsoft/asa/cim
```

- Изменить файл .env любым редактором (например, nano .env):

- Строка 1:

```
REGISTRY_NAME=адрес_docker_реестра
```

- Строки 10-15 включительно - часть строки, обозначенную многоточием, не менять:

```
...Server=адрес_хоста_баз_данных...Password=пароль_базы_дан  
ных...
```

- Строка 84:

```
asmpublisher_asmendpoint=https://адрес_frontend_сервера
```

- Изменить файл healthcheckservice\_settings.json любым редактором (например, nano healthcheckservice\_settings.json):

- Строки 3, 16, 29, 42, 55, 68, 80 - часть строки, обозначенную многоточием, не менять:

```
"signalRHubUrl": "http://адрес_хоста_интеграции..."
```

- Запустить контейнеры:

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

- Проверить запуск контейнеров через минуту с ввода последней команды:

```
docker ps -a
```

Контейнеры asmpublisher, repairsservice, stateservice, scorplogservice, cimservice, accountservice имеют статус Up (healthy).

- Перейти в директорию int:

```
cd /opt/rtsoft/asa/int
```

- Изменить файл .env любым редактором (например, nano .env):

- Строка 1:

```
REGISTRY_NAME=адрес_docker_реестра
```

- Строка 68 - часть строки, обозначенную многоточием, не менять:

№	Подп. и дата	Взам. №
Извм.	Кол.у	Лист

Извм.	Кол.у	Лист	№	Подп.	Дата

...Server=адрес\_хоста\_баз\_данных...Password=пароль\_базы\_дан  
ных...

8. Изменить файл aipintegration\_appsettings.json любым редактором (например, [nano aipintegration\\_appsettings.json](#)):

- Стока 11 - часть строки, обозначенную многоточием, не менять:

...Server=адрес\_хоста\_баз\_данных...Password=пароль\_базы\_дан  
ных...

9. Изменить файл khisubscriptionservice\_appsettings.json любым редактором (например, [nano khisubscriptionservice\\_appsettings.json](#)):

- Стока 4 - часть строки, обозначенную многоточием, не менять:

...Server=адрес\_хоста\_баз\_данных...Password=пароль\_базы\_дан  
ных...

10. Изменить файл SQLExecConfig.json любым редактором (например, [nano SQLExecConfig.json](#)):

- Стока 2 - часть строки, обозначенную многоточием, не менять:

...Server=адрес\_хоста\_баз\_данных...Password=пароль\_базы\_дан  
ных...

11. Запустить контейнеры:

[docker-compose up -d](#)

12. Проверить запуск контейнеров через минуту с ввода последней команды:

[docker ps -a](#)

Контейнеры signalsrestintegrationservice, aipintegrationservice, morop-linux, khisubscriptionservice имеют статус Up.

Сервисное ПО на integration сервере настроено. Настройка integration сервера завершена.

### 3.2.3 Frontend сервер

1. Перейти в директорию front:

[cd /opt/rtsoft/asa/front](#)

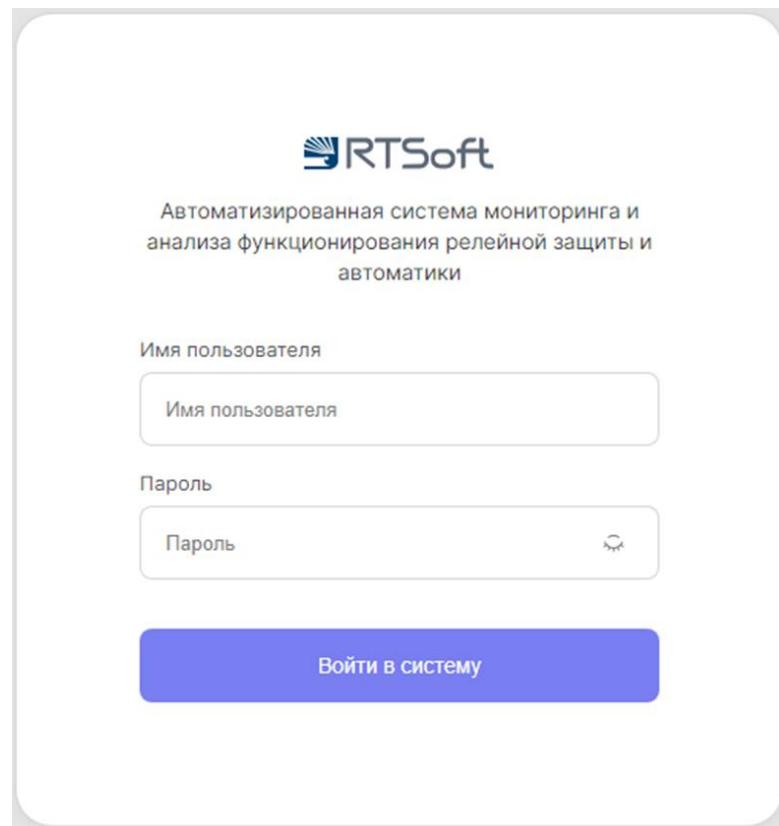
2. Разместить директорию html (из комплекта поставки) в /usr/share/nginx:

[sudo mv html /usr/share/nginx](#)

Инв. №	№Подп. и дата	Взам. №

Изм.	Кол.у	Лист	№	Подп.	Дата

3. Следует перейти на адрес сервера в браузере (или обновить страницу через Ctrl + F5, если ранее был произведен вход на приветственную страницу NGINX), должна отобразиться страница входа.



4. Убедиться, что запросы проходят между серверами, можно, введя заведомо неверные данные входа (любое сочетание букв / цифр / специальных символов, которое пропустит форма). В итоге должна отобразиться ошибка о несуществующем пользователе.

Инв. №	Подп. и дата	Взам. инв. №

A screenshot of the RTSoft login page, identical to the one above but with an error message. The user name 'awsdefsafd' and password 'awdgedew4e5tsghw4e5hdgs4w5' were entered. Below the login button, a red error message reads 'Пользователь с таким именем или паролем не найден' (User with such name or password not found). The rest of the page is identical to the successful login screenshot.

NGINX обрабатывает запросы клиента. Сервисное ПО на frontend сервере настроено. Настройка frontend сервера завершена. Настройка АСА РЗА завершена.

Изм.	Кол.у	Лист	№	Подп.	Дата

## 4 ОБНОВЛЕНИЕ

### 4.1 Database сервер

Сервисное ПО не устанавливается на данный сервер, обновление не требуется.

### 4.2 Backend сервер

1. Разместить в домашней директории пользователя (`/home/имя_пользователя`) архив `back.tar.gz` из комплекта обновления.
2. Сделать резервную копию предыдущей конфигурации:

```
cp -r /opt/asa/back /opt/asa/back-<текущая_дата>
```

3. Обновить текущую конфигурацию:

```
rm -r /opt/asa/back && tar xzf back.tar.gz -C /opt/asa
```

4. Перейти в директорию сервисов:

```
cd /opt/asa/back
```

5. Обновить образы:

```
docker-compose pull
```

6. Перезапустить сервисы:

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

7. Через 20 минут проверить состояние сервисов:

```
docker ps -a
```

Контейнеры имеют статус Up (healthy). Здесь и далее иные статусы рассматриваются как ошибки. В таком случае рекомендуется приостановить обновление, сбрасывая журнал («лог») из контейнера и передать его разработчикам.

Для сбора журнала можно выполнить команду `docker logs --since 30m имя_контейнера > файл.log 2>&1`.

В дальнейшем рекомендация по сбору логов будет опущена.

Обновление сервисного ПО на backend сервере произведено.

Инв. №	Подп. и дата	Взам. инв. №

Изм.	Кол.у	Лист	№	Подп.	Дата

### 4.3 Integration сервер

1. Разместить в домашней директории пользователя (/home/**имя\_пользователя**) архив cim.tar.gz из комплекта обновления.
2. Сделать резервную копию предыдущей конфигурации:

```
cp -r /opt/asa/cim /opt/asa/cim-<текущая_дата>
```

3. Обновить текущую конфигурацию:

```
rm -r /opt/asa/cim && tar xzf cim.tar.gz -C /opt/asa
```

4. Перейти в директорию сервисов:

```
cd /opt/asa/cim
```

5. Обновить образы:

```
docker-compose pull
```

6. Перезапустить сервисы:

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

7. Разместить в домашней директории пользователя (/home/**имя\_пользователя**) архив int.tar.gz из комплекта обновления.

8. Сделать резервную копию предыдущей конфигурации:

```
cp -r /opt/asa/int /opt/asa/int-<текущая_дата>
```

9. Обновить текущую конфигурацию:

```
rm -r /opt/asa/int && tar xzf int.tar.gz -C /opt/asa
```

10. Перейти в директорию сервисов:

```
cd /opt/asa/int
```

11. Обновить образы:

```
docker-compose pull
```

12. Перезапустить сервисы:

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

13. Через 20 минут проверить состояние сервисов:

```
docker ps -a
```

Инв. №	Подп. и дата	Взам. инв. №
Изм.	Кол.у	Лист №

Изм.	Кол.у	Лист №	Подп.	Дата	

Контейнеры asmpublisher, repairsservice, stateservice, scorplogservice, cimservice, accountservice имеют статус Up (healthy).

Контейнеры signalsrestintegrationservice, aipintegrationservice, mopop-linux, khisubscriptionservice имеют статус Up.

Обновление сервисного ПО на integration сервере произведено.

## 4.4 Frontend сервер

Сервисное ПО не устанавливается на данный сервер, однако, может потребоваться обновление статических файлов интерфейса, управляемых через NGINX.

1. Разместить в домашней директории пользователя (/home/**имя\_пользователя**) архив html.tar.gz из комплекта обновления.
2. Обновить текущую конфигурацию:

```
rm -r /usr/share/nginx/html && tar xzf html.tar.gz -C /usr/share/nginx
```

## 5 ПЕРЕНОС ДАННЫХ

Будем различать «старый» сервер – сервер, с которого требуется забрать данные и перенести их на «новый» сервер – сервер, на который требуется перенести данные.

### 5.1 Подготовка

1. Остановить сервисы на новом backend сервере, если происходил их запуск:

```
cd /opt/rtsoft/asa/back && docker-compose stop
```

2. Остановить сервисы на новом integration сервере, если происходил их запуск:

```
cd /opt/rtsoft/asa/cim && docker-compose stop
```

```
cd /opt/rtsoft/asa/int && docker-compose stop
```

### 5.2 Базы данных

1. Передать данные из старых баз в новые. Подразумевается, что есть прямой доступ между серверами, и новые базы пусты, поскольку новые сервера не эксплуатировались, следовательно, создание резервных копий не требуется:

```
pg_dump -Fc -E UTF8 AccountService | pg_restore -h адрес_нового_хоста_баз_данных -U postgres -Fc -c -d AccountService
```

Инв. №	Подп. и дата	Взам. инв. №

Изм.	Кол.у	Лист	№	Подп.	Дата

```
pg_dump -Fc -E UTF8 aipint_service | pg_restore -h  
адрес_нового_хоста_баз_данных -U postgres -Fc -c -d aipint_service -h
```

```
pg_dump -Fc -E UTF8 asmdb | pg_restore -h адрес_нового_хоста_баз_данных -  
U postgres -Fc -c -d asmdb
```

```
pg_dump -Fc -E UTF8 HealthCheck | pg_restore -h  
адрес_нового_хоста_баз_данных -U postgres -Fc -c -d HealthCheck
```

```
pg_dump -Fc -E UTF8 KHISubscription | pg_restore -h  
адрес_нового_хоста_баз_данных -U postgres -Fc -c -d KHISubscription
```

```
pg_dump -Fc -E UTF8 RepairService | pg_restore -h  
адрес_нового_хоста_баз_данных -U postgres -Fc -c -d RepairService
```

```
pg_dump -Fc -E UTF8 ScorpLogService | pg_restore -h  
адрес_нового_хоста_баз_данных -U postgres -Fc -c -d ScorpLogService
```

```
pg_dump -Fc -E UTF8 StateService | pg_restore -h  
адрес_нового_хоста_баз_данных -U postgres -Fc -c -d StateService
```

2. Запустить сервисы на новом backend сервере, если происходило их отключение:

```
cd /opt/rtsoft/asa/back && docker-compose up -d
```

3. Запустить сервисы на новом integration сервере, если происходило их отключение:

```
cd /opt/rtsoft/asa/cim && docker-compose up -d
```

```
cd /opt/rtsoft/asa/int && docker-compose up -d
```

4. Если прямого доступа нет, требуется создать резервные копии старых баз данных, вручную или через pgAdmin:

```
pg_dump -Fc -E UTF8 AccountService > AccountService.dump
```

```
pg_dump -Fc -E UTF8 aipint_service > aipint_service.dump
```

```
pg_dump -Fc -E UTF8 asmdb > asmdb.dump
```

```
pg_dump -Fc -E UTF8 HealthCheck > HealthCheck.dump
```

```
pg_dump -Fc -E UTF8 KHISubscription > KHISubscription.dump
```

```
pg_dump -Fc -E UTF8 RepairService > RepairService.dump
```

```
pg_dump -Fc -E UTF8 ScorpLogService > ScorpLogService.dump
```

Инв.	№	Подп. и дата	Взам. №

[pg\\_dump -Fc -E UTF8 StateService > StateService.dump](#)

5. Перенести резервные копии любым доступным способом на новый database сервер.

6. Восстановить базы данных из резервных копий, вручную или через pgAdmin:

[pg\\_restore -Fc -c -d AccountService AccountService.dump](#)

[pg\\_restore -Fc -c -d aipint\\_service aipint\\_service.dump](#)

[pg\\_restore -Fc -c -d asmdb asmdb.dump](#)

[pg\\_restore -Fc -c -d HealthCheck HealthCheck.dump](#)

[pg\\_restore -Fc -c -d KHISubscription KHISubscription.dump](#)

[pg\\_restore -Fc -c -d RepairService RepairService.dump](#)

[pg\\_restore -Fc -c -d ScorpLogService ScorpLogService.dump](#)

[pg\\_restore -Fc -c -d StateService StateService.dump](#)

### 5.3 Тома

1. Скопировать данные на старом backend сервере:

[docker cp asm-backend:/var/osc\\_fs osc\\_fs](#)

[docker cp asm-backend:/var/media media\\_fs](#)

[docker cp asm-backend:/opt/asm/static static\\_fs](#)

2. Сжать в архив:

[tar czf volumes.tar.gz osc\\_fs media\\_fs static\\_fs](#)

3. Перенести архив любым доступным способом на новый backend сервер, например, в директорию [/home/имя\\_пользователя](#).

4. Распаковать архив:

[tar xzf volumes.tar.gz](#)

5. Разместить данные в директориях контейнера:

[docker cp osc\\_fs/. asm-backend:/var/osc\\_fs](#)

[docker cp media\\_fs/. asm-backend:/var/media](#)

Инв.	№	Подп. и дата	Взам. №
Изм.	Кол.у	Лист	№

Изм.	Кол.у	Лист	№	Подп.	Дата

`docker cp static_fs/. asm-backend:/opt/asm/static`

Перенос данных со старых серверов на новые завершен.

Инв. №	Подп. и дата	Взам. инв. №

Изм.	Кол.у	Лист	№	Подп.	Дата

**АСА РЗА**  
Инструкция по установке и настройке  
на испытательном стенде

Лист  
25