АКЦИОНЕРНОЕ ОБЩЕСТВО «РТСофт»

**Информационная система СРЗА (ИС СРЗА)**

# ИНСТРУКЦИЯ ПО УСТАНОВКЕ И НАСТРОЙКЕ

**Москва, 2024**

Взам. инв. №

Подп. и дата

Инв. № подл.

| Изм. | Кол.у | Лист | № | Подп. | Дата |
|---|---|---|---|---|---|
| | | | | | |
| | | | | | |

**2019РТС_Д0328**

| | | | | | |
|---|---|---|---|---|---|
| Разраб. | | Иванов | | | |
| Пров. | | Иванов | | | |
| | | | | | |
| | | | | | |
| ГИП | | Иванов | | | |

**Информационная система СРЗА (ИС СРЗА)**
Инструкция по установке и настройке на испытательном стенде

| Стадия | Лист | Листов |
|---|---|---|
| Э | 1 | 49 |

АО «РТСофт»

**ОГЛАВЛЕНИЕ**

## ТЕРМИНЫ, ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ

| Сокращение | Расшифровка |
|---|---|
| AD | Active Directory |
| API | Программный интерфейс приложения (англ. Application Programming Interface) |
| HDD | Накопитель на жёстких магнитных дисках (англ. Hard Disk Drive) |
| JSON | Текстовый формат обмена данными, основанный на JavaScript (англ. JavaScript Object Notation) |
| SNMP | Простой протокол сетевого управления (англ. Simple Network Management Protocol) |
| URL | Единый указатель ресурса (англ. Uniform Resource Locator) |
| АРМ | Автоматизированное рабочее место |
| АС | Автоматизированная система |
| БД | База данных |
| ИС СРЗА | Информационная Система Службы РЗА (Система) |
| ОЗУ | Оперативное запоминающее устройство |
| ОС | Операционная система |
| ПО | Программное обеспечение |
| ЦПУ | Центральное процессорное устройство |

# 1 ВВЕДЕНИЕ

## 1.1 Состав комплекта поставки

В комплект поставки ПО Системы входят инсталляционные пакеты для всех основных компонентов Системы. Перечень этих пакетов представлен в Таблице 1.

Таблица 1. Перечень инсталляционных пакетов

| Компонент | Название пакета | Описание |
|---|---|---|
| Сервер баз данных | dbrza_distrib.zip | Архив с ПО |
| Сервер приложений | | |
| Веб-сервер | | |
| Модуль распределенного хранения файлов | ИС СРЗА. Модуль распределенного файлового хранилища_<№версии>_installer_(x64) | Дистрибутив для установки региональных модулей хранилища файлов |

Также в комплект поставки ПО Системы включены следующие дополнительные пакеты:

Таблица 2. Перечень серверов с запущенными на них сервисами

| № | Назначение сервера | Сервисы |
|---|---|---|
| 1 | Сервер БД1 | postgresql. Версия от 10<br>haproxy1. Версия 2.2.9<br>redis1 и sentinel1. Версия от 5.0.0<br>keepalived. Версия от 5.4.0<br>etcd. Версия 3.3.25 |
| 2 | Сервер БД2 | postgresql. Версия от 10<br>haproxy1. Версия 2.2.9<br>redis1 и sentinel1. Версия от 5.0.0<br>keepalived. Версия от 5.4.0 |
| 3 | Сервер БД3 | postgresql. Версия от 10<br>haproxy1. Версия 2.2.9<br>redis1 и sentinel1. Версия от 5.0.0<br>keepalived. Версия от 5.4.0 |
| 4 | Сервер приложений | docker последней версии<br>dbrza_backend_1<br>dbrza_backend_2<br>backend-celery<br>backend-ws<br>storage |

| № | Назначение сервера | Сервисы |
|---|---|---|
| | | blanks<br>logger<br>rabbitmq. Версия 3.7.5<br>nginx. Версия от 1.20<br>mongodb. Версия 4.4.* |
| 5 | Веб-сервер | nginx. Версия от 1.20<br>spnego. Версия 1.1 |
| 6 | Интеграционный linux-сервер | docker последней версии<br>aipintegration<br>scorplogservice<br>cimservice<br>accountservice |
| 7 | Интеграционный windows-сервер | dbrza_landocs |

### 1.2 Последовательность установки

Установка и настройка ПО Системы на боевом стенде производится в следующей последовательности:

1) Проверка серверов, предварительно подготовленных в соответствии с требованиями, представленными в Таблице 2.
2) Установка ПО Системы в следующей последовательности:

- Серверы баз данных;
- Серверы приложений;
- Веб-сервер.

## 2 ПОДГОТОВКА К УСТАНОВКЕ

ПО Системы устанавливается на виртуальные или физические сервера, подготовленные в соответствии с требованиями, указанными в Таблице 3.

Таблица 3. Требования к серверам

| Компонент | Аппаратное обеспечение | Программное обеспечение |
|---|---|---|
| Сервер баз данных 1 (Основное хранилище) | ЦПУ - 8 ядер;<br>ОЗУ - не менее 16 GB;<br>HDD - не менее 1 ТВ свободного дискового пространства | ОС Astra Linux SE 1.7;<br>СУБД PostgreSQL;<br>СУБД Redis; |

| Компонент | Аппаратное обеспечение | Программное обеспечение |
|---|---|---|
| | RAID5 | |
| Сервер баз данных 2 (Основное хранилище) | ЦПУ - 8 ядер;<br>ОЗУ - не менее 16 GB;<br>HDD - не менее 1 TB свободного дискового пространства<br>RAID5 | ОС Astra Linux SE 1.7;<br>СУБД PostgreSQL;<br>СУБД Redis; |
| Сервер баз данных 3 (Основное хранилище) | ЦПУ - 8 ядер;<br>ОЗУ - не менее 16 GB;<br>HDD - не менее 1 TB свободного дискового пространства<br>RAID5 | ОС Astra Linux SE 1.7;<br>СУБД PostgreSQL;<br>СУБД Redis+; |
| Сервер приложений (совмещает также в себе роль файлового хранилища) | ЦПУ - 8 ядер;<br>ОЗУ - не менее 16 GB;<br>HDD - не менее 50 GB свободного дискового пространства | ОС Astra Linux SE 1.7;<br>ПО Nginx;<br>ПО Docker |
| Сервер взаимодействия с внешними системами | ЦПУ - 2 ядра;<br>ОЗУ - не менее 16 GB;<br>HDD - не менее 60 GB свободного дискового пространства | ОС Microsoft Windows Server 2012 R2 Standard или выше;<br>ПО Microsoft .Net Framework 4.7 и выше; |
| Веб-сервер | ЦПУ – 6 ядер;<br>ОЗУ – не менее 6 GB;<br>HDD – не менее 20 GB свободного дискового пространства. | ОС Astra Linux SE 1.7;<br>ПО Nginx; |
| Сервер интеграции Linux | ЦПУ - 4 ядра;<br>ОЗУ - не менее 12 GB;<br>HDD - не менее 120 GB свободного дискового пространства | ОС Astra Linux SE 1.7;<br>ПО Docker. |
| Сервер интеграции Windows | ЦПУ - 4 ядра;<br>ОЗУ - не менее 4 GB;<br>HDD - не менее 60 GB свободного дискового пространства | ОС Microsoft Windows Server 2012 R2 Standard или выше;<br>ПО Microsoft .Net Framework 4.7 и выше; |

## 2.1 Действия персонала перед началом установки ПО

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

Перед началом установки соответствующий инсталляционный пакет должен быть скопирован на диск целевого сервера.

В случае использования виртуальных машин предпочтительнее использование сетевого файлового хранилища, доступ к которому может быть получен с каждой из ОС целевых серверов. Инсталляционный пакет, расположенный на сетевом файловом хранилище, должен быть скопирован на ОС целевого сервера.

При использовании не виртуальных, а физических серверов предпочтительнее может оказаться вариант использования USB-носителей, на который необходимо предварительно скопировать требуемые инсталляционные пакеты.

Таким образом, в пользовательской папке каждого сервера (например, в папке «Рабочий стол» для ОС Windows Server или /home/<user>/ для ОС Linux) должен быть размещен соответствующий инсталляционный пакет (см. таблицу 1) и все дополнительные пакеты.

Для установки программного обеспечения поддержки запуска Докер-контейнеров необходимо наличие доступа к репозиторию в сети интернет, либо к локальной копии репозитория.

# 3 УСТАНОВКА И НАСТРОЙКА ПО

## 3.1 Установка и настройка программного обеспечения Докер-контейнеров

### 3.1.1 Установка и настройка docker

Установку необходимо произвести на сервере-приложений и интеграционном сервере. Для установки и настройки docker необходимо на компьютере с установленным Linux и доступом в интернет произвести следующие действия:

1. Обновление установщика пакетов

```
# sudo apt-get update
```

2. Установка Docker

```
# sudo apt install docker.io
```

3. После установки Docker рекомендуется предоставить администратору право работать с контейнерами не используя sudo. Для этого пользователя нужно включить в группу docker:

```
# sudo exec su - $USER
```

4. Проверка работы Docker

```
# docker ps
```

5. Создать конфиг daemon.json файл в папке /etc/docker/ с текстом:

```
{
    "insecure-registries" : [" … "]
}
```

6. Перезапустить докер сервис:

```
# sudo systemctl restart docker
```

### 3.1.2 Установка и настройка docker-compose

1. Положить файл docker-compose на сервер
2. Переместить docker-compose в рабочую директорию

```
# sudo cp docker-compose /usr/local/bin/docker-compose
```

3. Установить права доступа

```
# sudo chmod +x /usr/local/bin/docker-compose
```

4. Проверить версию

```
# docker-compose --version
```

## 3.2 Копирование установочных файлов

Скопировать на сервера следующие файлы, поставляющиеся в архиве dbrza_distrib.zip. Файлы для сервера приложений – backend. Файлы для сервера интеграции Linux – linux_int. Файлы для сервера интеграции Windows – windows_int:

| | | |
|---|---|---|
| Сервер приложений | /opt/dbrza/backend/ | docker-compose.yml<br>docker-compose.override.yml<br>.env |
| | /opt/dbrza/backend/keys/ | certificate.crt<br>private.key |
| | /etc/nginx/ | default.conf |
| | /opt/dbrza/backend/ | templates |
| Интеграционный сервер Linux | /opt/scorpion/deploy | docker-compose.override.yml<br>docker-compose.yml<br>.env<br>appsettings.json<br>appsettings_scorp.json |
| | /opt/scorpion/int_deploy | appsettings.json<br>docker-compose.yml<br>docker-compose.override.yml<br>.env<br>krb5.conf<br>SQLExecConfig.json |

| | | |
|---|---|---|
| | | SQLExecConfig2.json<br>SQLExecConfig3.json |
| Интеграционный сервер Windows | C:\inetpub\wwwroot | App_Data<br>aspnet_client<br>bin |

### 3.3 Установка и настройка серверов баз данных

Для кластера БД на каждой ноде (Сервер БД1, Сервер БД2, Сервер БД3) нужно провести дополнительную конфигурацию:

1. Подключить и настроить репозитории, содержащие Postgres.
2. Выполнить обновление списка пакетов и системы:
   # sudo apt update && sudo apt upgrade
3. Установить пакеты (необходимые пакеты)
   # sudo apt install postgresql-14 etcd patroni haproxy keepalived redis redis-sentinel openssl ca-certificates
4. Запретить запуск и остановить службу postgresql
   # sudo systemctl stoppostgresql
   # sudo systemctl disable postgresql
5. Разрешить пересылку транзитных пакетов и использование процессами виртуального IP
   # sudo sh -c "cat << EOF >> /etc/sysctl.conf
   net.ipv4.ip_forward = 1
   net.ipv4.ip_nonlocal_bind = 1
   EOF"
6. Проверить результат командой
   # sysctl -p
7. Настройка приложения keepalived:

   а. Создать конфигурационный файл по адресу:
   # /etc/keepalived/keepalived.conf

   б. Добавить в конфигурационный файл параметры согласно листингу ниже, заменив данные на свои.

   Замены потребуют параметры:

```
vrrp_instance
interface
auth_pass
virtual_ipaddress
```

   б.1 Для мастер ноды

```
vrrp_instance MyCluster {
```

```
interface eth0
state MASTER
virtual_router_id 200
priority 100
advert_int 1
authentication {
auth_type PASS
auth_pass [AUTH_PASSWORD]
}
virtual_ipaddress  {
[VIRTUAL_IP]
}
}
```

б.2 Для резервной ноды

```
vrrp_instance dbrzadb {
interface eth0
state BACKUP
virtual_router_id 200
priority 99
advert_int 1
authentication {
auth_type PASS
auth_pass [AUTH_PASSWORD]
}
virtual_ipaddress  {
[VIRTUAL_IP] }
}
```

8. Выполнить команду для перезапуска сервиса с новыми настройками (на каждом сервере)

   # sudo systemctl restart keepalived

9. Настройка etcd задается в файле /etc/default/etcd Необходимые параметры для замены:

   ETCD_NAME – Название ноды, должно быть уникальное

   ETCD_INITIAL_CLUSTER_TOKEN – Токен для авторизации новой ноды кластера, один для всех

   ETCD_INITIAL_CLUSTER – перечислить ноды, которые будем использовать, формат:

   Node1=http://ip,Node2=ip,http://Node3=ip

   ETCD_INITIAL_ADVERTISE_PEER_URLS – список равноправных URL-адресов, по которым его могут найти остальные узлы кластера. Эти адреса используются для передачи данных по кластеру. По крайней мере, один из этих адресов должен быть маршрутизируемым для всех членов кластера. Могут содержать доменные имена. Используется только при первом запуске нового узла кластера.

   ETCD_ADVERTISE_CLIENT_URLS – Список равноправных URL-адресов, по которым его могут найти остальные узлы кластера. Эти адреса

используются для передачи данных по кластеру. По крайней мере, один из этих адресов должен быть маршрутизируемым для всех членов кластера. Могут содержать доменные имена.

ETCD_LISTEN_PEER_URLS – задаёт схему и точку подключения для остальных узлов кластера, по шаблону scheme://IP:port. Схема может быть http, https. Альтернатива, unix:// или unixs:// для юникс сокетов. Если в качестве IP адреса указано 0.0.0.0, то указанный порт будет прослушиваться на всех интерфейсах

ETCD_LISTEN_CLIENT_URLS – задаёт схему и точку подключения для клиентов кластера. В остальном совпадает с ETCD_LISTEN_PEER_URLS.

a.1 Пример настройки MASTER ноды:

```
ETCD_NAME="dbrzadb1"
ETCD_DATA_DIR="/var/lib/etcd"
ETCD_INITIAL_CLUSTER_STATE="new"
ETCD_INITIAL_CLUSTER_TOKEN="etcd-cluster"
ETCD_INITIAL_CLUSTER="dbrzadb1=http://[NODE_1_IP]:2380,dbrzadb2=http://
[NODE_2_IP]:2380,dbrzadb3=http:// [NODE_3_IP]:2380"
ETCD_INITIAL_ADVERTISE_PEER_URLS="http://[NODE_1_IP]:2380"
ETCD_ADVERTISE_CLIENT_URLS="http://[NODE_1_IP]:2379,http://ip"
ETCD_LISTEN_PEER_URLS="http://0.0.0.0:2380"
ETCD_LISTEN_CLIENT_URLS=http://0.0.0.0:2379
```

a.2 Пример настройки BACKUP нод:

```
ETCD_NAME="dbrzadb1
ETCD_DATA_DIR="/var/lib/etcd"
ETCD_INITIAL_CLUSTER_STATE="new"
ETCD_INITIAL_CLUSTER_TOKEN="etcd-cluster"
ETCD_INITIAL_CLUSTER="dbrzadb1=http://[NODE_1_IP]:2380,dbrzadb2=http://
[NODE_2_IP]:2380,dbrzadb3=http:// [NODE_3_IP]:2380"
ETCD_INITIAL_ADVERTISE_PEER_URLS="http://[NODE_1_IP]:2380"
ETCD_ADVERTISE_CLIENT_URLS="http://[NODE_1_IP]:2379,http://ip"
ETCD_LISTEN_PEER_URLS="http://0.0.0.0:2380"
ETCD_LISTEN_CLIENT_URLS=http://0.0.0.0:2379
```

10. Перезапускаем службу etcd на всех нодах

```
# sudo systemctl restart etcd
```

11. Проверяем работу:

```
# etcdctl --endpoints=http://[IP_ADDRESS]:2379 member list
```

пример работы выглядит так:

```
9a20d64f814efc90: name=dbrzadb3 peerURLs=http://[NODE_3_IP]:2380
clientURLs=http://ip:2379,http://[NODE_3_IP]:2379 isLeader=true
b3ee076680ce52e0: name=dbrzadb2 peerURLs=http://[NODE_2_IP]:2380
clientURLs=http://ip,http://[NODE_2_IP]:2379 isLeader=false
d04622c318d779a4: name=dbrzadb1 peerURLs=http:// [NODE_1_IP]:2380
clientURLs=http://ip:2379,http:// [NODE_1_IP]:2379 isLeader=false
```

12. Настройка patroni

Пример конфигурации задается в файле /etc/patroni/config.yml

```
restapi: # Настройки для мониторинга HAproxy
listen: [NODE_1_IP]:8008
connect_address: [NODE_1_IP]:8008
etcd:
  hosts: [NODE_1_IP]:2379, [NODE_2_IP]:2379, [NODE_3_IP]:2379 # перечисляем адреса и
порты нод ETCD
bootstrap:
  dcs:
    ttl: 30
    loop_wait: 10
    retry_timeout : 10
    maximum_lag_on_failover: 1048576
    postgresql:
      use_pg_rewind: true
      use_slots: true
      parameters:
        wal_keep_segments: 100
  initdb:
  - encoding: UTF8
  - data-checksums
  pg_hba:
  - host replication replicator 0.0.0.0/0 md5
  - host all all 0.0.0.0/0 md5

  users: # создаем пользователей с нужными правами
    postgres:
      password: [REDIS_PASSWORD]
      options:
        - createrole
        - createdb
    replicator:
      password: [REDIS_PASSWORD]
      options:
        - replication
postgresql:
  listen: 0.0.0.0:5432
  connect_address: [NODE_1_IP]:5432
  data_dir: /var/lib/postgresql/14/main/
  bin_dir: /usr/lib/postgresql/14/bin/
  config_dir: /etc/postgresql/14/main/

  authentication:
    replication:
      username: replicator
      password: [REDIS_PASSWORD]
    superuser:
      username: postgres
      password: [REDIS_PASSWORD]
parameters:
  unix_socket_directories: '/var/run/postgresql/'
  stats_temp_directory: /var/lib/pgsql_stats_tmp
```

13. Разрешить подключение пользователям postgres по сети, необходимо выполнить на всех нодах:

```
# sudo sh -c "cat << EOF >> /etc/postgresql/14/main/pg_hba.conf
host replication replicator 0.0.0.0/0 md5
host all all 0.0.0.0/0 md5
```

EOF"

14. Перезапустить сервис patroni на каждой ноде

# sudo systemctl restart patroni

15. Настройка redis sentinel

A. Создать файл /etc/redis/sentinel.conf

A.1 Для мастер ноды:

```
supervised systemd
protected-mode no
port 26379
bind [NODE_1_IP] ip ::1 # какой адрес слушает демон, меняем первый IP на адрес
мастера
daemonize no
sentinel monitor dbrzadb1 [NODE_1_IP] 6379 1 # отслеживаем состоянии ноды, меняем на
IP мастера, dbrzadb1 – на название ноды мастера
sentinel down-after-milliseconds dbrzadb1 3000 # меняем dbrzadb1  на название ноды
мастера
acllog-max-len 128

sentinel failover-timeout dbrzadb1 5000 # меняем dbrzadb1 на название ноды мастера
sentinel deny-scripts-reconfig yes
sentinel resolve-hostnames yes
sentinel announce-hostnames yes
```

A.2

```
protected-mode no
supervised systemd
port 26379
bind [NODE_2_IP] ip ::1 # какой адрес слушает демон, меняем первый IP на адрес
мастера

daemonize no
sentinel monitor dbrzadb1 [NODE_3_IP] 6379 1
sentinel down-after-milliseconds dbrzadb1 3000 # отслеживаем состоянии ноды, меняем
на IP мастера, dbrzadb1 – на название ноды мастера
acllog-max-len 128
sentinel auth-pass dbrzadb1 qwerty1234 # меняем dbrzadb1 на название ноды мастера и
на свой пароль

sentinel failover-timeout dbrzadb1 5000 # dbrzadb1 меняем на название ноды мастера
sentinel deny-scripts-reconfig yes
sentinel resolve-hostnames yes
sentinel announce-hostnames yes
```

16.Создать Unit файл для автозапуска демона sentinel

```
sudo sh -c "cat << EOF >> /etc/systemd/system/sentinel.service
[Unit]
Description=Redis Sentinel
After=network.target

[Service]
User=redis
Group=redis
ExecStart=/usr/bin/redis-server /etc/redis/sentinel.conf --sentinel
ExecStop=/usr/bin/redis-cli shutdown
Restart=always
```

```
[Install]
WantedBy=multi-user.target
EOF"
```

17. Необходимо задать параметры для работы redis в файле конфигурации по пути /etc/redis/redis.conf

A.1 Для master ноды

```
bind NODE_IP ip ::1
requirepass "[REDIS_PASSWORD]" # заменить на свой пароль
```

A.2 на slave нодах:

```
bind NODE_IP ip ::1
replicaof [NODE_3_IP] 6379 # заменить [NODE_3_IP] на IP мастер ноды
masterauth "[REDIS_PASSWORD]" # пароль указанный для мастер ноды
requirepass "[REDIS_PASSWORD]" # пароль для доступа к экземпляру redis
```

18. На каждой ноде запустить sentinel

```
# sudo systemctl daemon-reload && sudo systemctl start sentinel.service
```

19. Настройка haproxy

Добавить в конфигурационный файл по пути /etc/haproxy/haproxy.cfg

Необходимо будет заменить строку вида:

```
dbrzadb1 [NODE_1_IP]:5432
dbrzadb1 [NODE_1_IP]:6379
```

на:

```
Название_ноды IP_Ноды:Порт_Ноды
global
    maxconn 100
    log 127.0.0.1 local0
defaults
   log global
    mode tcp
    retries 2
    timeout client 30m
    timeout connect 4s
    timeout server 30m
    timeout check 5s
listen stats
    mode http
    bind *:7000
    stats enable
    stats uri /
listen postgres
    bind *:5000
    option httpchk
    http-check expect status 200
    default-server inter 3s fall 3 rise 2 on-marked-down shutdown-sessions
    server dbrzadb1 [NODE_1_IP]:5432 maxconn 100 check port 8008 # указываем первую
ноду
    server dbrzadb2 [NODE_2_IP]:5432 maxconn 100 check port 8008 # указываем вторую
ноду
    server dbrzadb3 [NODE_3_IP]:5432 maxconn 100 check port 8008 # указываем 3 ноду
listen redis
    bind *:6000
```

```
    mode tcp
    option tcplog
    option tcp-check
    #uncomment these lines if you have basic auth
    tcp-check send AUTH \[REDIS_PASSWORD]\r\n
    tcp-check expect string +OK
    tcp-check send PING\r\n
    tcp-check expect string +PONG
    tcp-check send info\ replication\r\n
    tcp-check expect string role:master
    tcp-check send QUIT\r\n
    tcp-check expect string +OK
    server dbrzadb1 [NODE_1_IP]:6379 maxconn 1024 check inter 1s # указываем первую
ноду
    server dbrzadb2 [NODE_2_IP]:6379 maxconn 1024 check inter 1s # указываем первую
ноду
    server dbrzadb3 [NODE_3_IP]:6379 maxconn 1024 check inter 1s # указываем первую
ноду
```

20. Далее необходимо перечитать конфигурационные файлы демонов и перезапустить службы на нодах

    # sudo systemctl daemon-reload && sudo systemctl restart haproxy redis sentinel patroni

21. Для проверки работоспособности можно перейти по адресу:

    # http://виртуальный_ip:7000

    *или*

    # http://Нода_IP:7000

    с соблюдением очередности:

    1. Сервер БД1
    2. Сервер БД2
    3. Сервер БД3

### 3.3.1 Первичная настройка базы данных

Выполняется только на Сервере БД1 (мастер-нода). После запуска контейнеров на Сервере БД1 выполнить следующие команды:

1. Зайти под пользователем postgres в psql:

# psql -U postgres -h [ip_адрес_мастер-ноды] -p 5000 -W

2. Создать пользователя dbrza:

# create user dbrza;

3. Создать базу данных dbrza с владельцем dbrza:

# create database dbrza with owner dbrza;

4. Передать пользователю dbrza права postgres:

# grant postgres to dbrza;

5. Добавить права на подключение пользователя dbrza к базе данных dbrza:

# grant connect on database dbrza to dbrza;

6. Назначить пользователю dbrza пароль dbrza:

# alter user dbrza with password 'dbrza'; Заполнение базы данных из дампа

Для запуска системы с предзагруженными данными можно воспользоваться заполнением БД из дампа на сервере БД1:

1. Поместить дамп на сервер
2. Выполнить команду:

# psql -U postgres -h [ip_адрес_мастер-ноды] -p 5000 -W -d dbrza < имя дампа

3. Дождаться завершения и выйти из контейнера

### 3.4 Установка и настройка серверов распределенных хранилищ

Подключение к серверам хранилищ обеспечивается за счет установки в корпоративной сети на выделенном хранилище приложения, а также установки связи с помощью интерфейса веб-приложения Систем подключения к серверу по URL-ссылке с форматом:

Protocol_Domain Name_Port, например, https://....

**Примечание**: требуется обязательное использование Port.

### 3.5 Установка и настройка серверов приложений

1. Обновление установщика пакетов:
   # sudo apt update
2. Установка пакета rabbitmq:
   # sudo apt install rabbitmq-server
3. Запуск плагина для управления rabbitmq:
   # sudo rabbitmq-plugins enable rabbitmq_management
4. Создание пользователя и задание пароля для rabbitmq:
   # sudo rabbitmqctl add_user user password && \
   sudo rabbitmqctl set_user_tags user administrator && \
   sudo rabbitmqctl set_permissions -p / user ".*" ".*" ".*"
5. Проверить, что служба запустилась:
   # systemctl status rabbitmq-server
6. Если не запустилась, запустить вручную:
   # systemctl start rabbitmq-server
7. Установка пакета gnupg и curl:
   # sudo apt-get install gnupg curl
8. Создание директории:
   # sudo mkdir /usr/share/keyrings
9. Подключить и настроить репозиторий, содержащий Mongodb.

10. Обновление установщика пакетов:

```
# sudo apt-get update
```

11. Установка mongo:

```
# sudo apt-get install -y mongodb-org
```

12. Задание значений:

```
# echo "mongodb-org hold" | sudo dpkg --set-selections \
echo "mongodb-org-server hold" | sudo dpkg --set-selections \
echo "mongodb-org-shell hold" | sudo dpkg --set-selections \
echo "mongodb-org-mongos hold" | sudo dpkg --set-selections \
echo "mongodb-org-tools hold" | sudo dpkg --set-selections
```

13. Запуск службы mongodb:

```
# sudo systemctl start mongod
```

14. Проверка статуса mongo:

```
# sudo systemctl status mongod
```

15. Выполнить логин к registry с образами:

```
# sudo docker login -u <username> -p <password> registry.local:
```

16. Установка nginx:

```
# sudo apt install nginx
```

17. Проверка работы:

```
# sudo systemctl status nginx
```

18. Скопировать файл конфигурации nginx:

```
# sudo cp /home/rza/default.conf /etc/nginx/conf.d/default.conf
```

19. Перезапуск службы nginx:

```
# sudo systemctl restart nginx
```

20. Проверка работы:

```
# sudo systemctl status nginx
```

21. Создать директорию проекта:

```
# sudo mkdir -p /opt/dbrza/backend
```

22. Перенести дистрибутив в директорию проекта:

```
# cp /home/<username>/docker-compose.yml /opt/dbrza/backend
# cp /home/<username>/docker-compose.override.yml /opt/dbrza/backend
# cp /home/<username>/.env /opt/dbrza/backend
```

23. Перейти в директорию проекта:

```
# cd /opt/dbrza/backend
```

24. Заполнить учетные записи и необходимые адреса в файле .env:

```
# nano .env
```

25. На сервере из директории, где располагается docker-compose.yml выполнить команды:

```
# sudo docker-compose pull
# sudo docker-compose up -d
```

26. Выдача прав на каталог:

```
# sudo chmod -R 777 /opt/dbrza/backend/upload
```

За состоянием сервисов можно наблюдать с помощью команды:

```
# sudo docker logs -f <имя контейнера>
```

### 3.6 Распаковка SSL ключей

На сервере приложений:

1. Перенести pfx сертификат на сервер.
2. Распаковать сертификат:
   ```
   # openssl pkcs12 -in certificate.pfx -clcerts -nokeys -out certificate.crt
   ```
3. Распаковать ключ:
   ```
   # openssl pkcs12 -in certificate.pfx -nodes -nocerts -out key-encrypted.key
   ```

### 3.7 Установка и настройка веб-сервера

1. Перейти во временный каталог:
   ```
   cd /tmp
   ```
2. Обновление установщика пакетов
   ```
   # sudo apt-get update
   ```
3. Установка дополнительных пакетов:
   ```
   # sudo apt install -y libkrb5support0=1.18.3-6+deb11u2astra1 \
   libkrb5-3=1.18.3-6+deb11u2astra1 \
   build-essential \
    wget \
    tar \
    gcc \
    make \
    libssl-dev \
    libpcre3-dev \
    krb5-user \
    libpam-krb5 \
    krb5-multidev=1.17-3+deb10u5 \
    libkrb5-dev \
    zlib1g-dev \
    libcap2-bin \
    libgd-dev \
    libpcre++-dev \
    libxslt-dev \
    git
   ```
4. Скачивание nginx и модуль spnego. Переместить их в директорию /tmp
5. Переход в каталог nginx:
   ```
   # cd /tmp/nginx
   ```

6. Установка пакетов для сборки nginx:

```
# sudo apt install -y libssl1.1=1.1.1n-0+deb10u3-astra4+ci5 libssl-dev libxml2-dev install
libxslt-dev libxpm4=1:3.5.12-1 libxpm-dev libtiff5=4.1.0+git191117-2~deb10u4
libexpat1=2.2.6-2+deb10u5 libexpat1-dev libfontconfig-dev libgd-dev
libkrb5support0=1.18.3-6+deb11u2astra1 libkrb5-3=1.18.3-6+deb11u2astra1 libkrb5-
3=1.18.3-6+deb11u2astra1 libk5crypto3=1.18.3-6+deb11u2astra1 libgssapi-krb5-2=1.18.3-
6+deb11u2astra1 libgssrpc4=1.18.3-6+deb11u2astra1 libkadm5srv-mit12=1.18.3-
6+deb11u2astra1 libkadm5clnt-mit12=1.18.3-6+deb11u2astra1 libcom-err2=1.44.5-
1+deb10u3 comerr-dev krb5-multidev=1.18.3-6+deb11u2astra1 libkrb5-dev
```

7. Сборка nginx с модулем spnego:

```
# sudo ./configure \
        --user=nginx \
        --prefix=/usr/share/nginx \
        --sbin-path=/usr/sbin/nginx \
        --conf-path=/etc/nginx/nginx.conf \
        --http-log-path=/var/log/nginx/access.log \
        --pid-path=/var/run/nginx.pid \
        --with-mail=dynamic \
        --with-mail_ssl_module \
        --with-stream=dynamic \
        --with-stream_ssl_module \
        --with-stream_realip_module \
        --with-stream_ssl_preread_module \
        --with-http_auth_request_module \
        --with-http_secure_link_module \
        --with-http_gzip_static_module \
        --with-http_stub_status_module \
        --with-http_ssl_module \
        --with-compat \
        --with-pcre \
        --with-pcre-jit \
        --with-http_gunzip_module \
        --with-http_gzip_static_module \
        --with-http_image_filter_module=dynamic \
        --with-http_sub_module \
        --with-http_xslt_module=dynamic \
        --with-stream=dynamic \
        --with-stream_ssl_module \
        --with-mail=dynamic \
        --with-mail_ssl_module\
        --add-module=spnego-http-auth-nginx-module && \
      sudo make && \
      sudo make install
```

8. Создание дополнительных папок:

```
#  sudo mkdir -p /usr/lib/nginx/modules && \
```

```
sudo mkdir -p /var/lib/nginx/body && \
sudo mkdir -p /var/lib/nginx/fastcgi && \
sudo mkdir -p /var/lib/nginx/proxy && \
sudo mkdir -p /var/lib/nginx/scgi && \
sudo mkdir -p /var/lib/nginx/uwsgi && \
sudo mkdir -p /etc/nginx/conf.d
```

9. Настройка пользователя nginx:

```
# sudo adduser --system --no-create-home --group --disabled-login --disabled-password
nginx && \
  sudo setcap cap_net_bind_service=ep /usr/sbin/nginx
```

10. Создание службы nginx:

```
# sudo nano /lib/systemd/system/nginx.service
```

11. В файл необходимо вписать:

```
[Unit]
Description=The NGINX HTTP and reverse proxy server
After=syslog.target network-online.target remote-fs.target nss-lookup.target
Wants=network-online.target

[Service]
Type=forking
PIDFile=/var/run/nginx.pid
ExecStartPre=/usr/sbin/nginx -t
ExecStart=/usr/sbin/nginx
ExecReload=/usr/sbin/nginx -s reload
ExecStop=/bin/kill -s QUIT $MAINPID
PrivateTmp=true

[Install]
WantedBy=multi-user.target
```

12. Запуск службы nginx:

```
# systemctl start nginx
```

13. Проверка службы:

```
# systemctl status nginx
```

14. Создание рабочих каталогов:

```
# mkdir -p /etc/ssl/<название сервера>/ \
mkdir -p /etc/nginx/ldap/
```

15. Положить сертификаты веб-сервера:

```
# sudo cp certificate.crt /etc/ssl/<название сервера>/certificate.crt
# sudo cp key-encrypted.key /etc/ssl/<название сервера>/key-encrypted.key
```

16. Перенести на сервер содержимое архива frontend.zip

17. Переложить фронтенд-приложение в рабочую директорию:

# sudo cp /home/infraadmin/html/* /usr/share/nginx/html/.

18. Переложить конфигурацию nginx (предварительно изменить в файле путь к сертификатам из п.15):
    #sudo cp default.conf /etc/nginx/conf.d/default.conf
19. Отредактировать файл default.conf указав правильный путь к сертификатам из п.15:
    # sudo nano /etc/nginx/conf.d/default.conf
20. Переложить новый spnego.keytab файл:
    # sudo cp spnego.keytab /etc/nginx/ldap/spnego.keytab
21. Переложить krb5.conf:
    # sudo cp krb5.conf /etc/krb5.conf
22. Перезагрузить nginx:
    # systemctl restart nginx
23. Удалить все временные файлы для сборки:
    # rm -rf /tmp/*

## 3.8 Установка и настройка сервера взаимодействия с внешними системами на Linux

### 3.8.1 Предварительные работы

Скопировать конфигурационные файлы из linux_int в директорию /opt/scorpion

### 3.8.2 Установка и настройка интеграционного окружения Системы с ИУС «АИП»

Запустить интеграционные сервисы:

1. Перейти в директорию /opt/scorpion/deploy:
   cd /opt/scorpion/deploy
2. Выполнить команду
   docker-compose up -d

### 3.8.3 Установка и настройка интеграционного окружения Системы с сервисом AD

#### 3.8.3.1 Настройка параметров соединения с сервисом AD и требования к ведению групп и учетных записей AD

Основные параметры взаимодействия с контроллером доменов описаны в разделе 3.4 пункт 3 настоящей инструкции по установке и настройке.

Доступ к контроллеру доменов осуществляется **по протоколу LDAP** через глобальный порт **3268**. Это обусловлено необходимостью поиска учетных записей во всем лесу доменов. Подключение осуществляется с помощью сервисной учетной записи.

Для корректной работы протокола LDAP с контроллером доменов необходимо, чтобы группы пользователей были определенного типа:

- Universal Security Group;
- Universal DistributionGroup.

**Внимание!** Доменные группы должны быть обязательно либо Universal Security, либо Universal Distribution. В ином случае возможны проблемы с подключением учетных записей этих групп AD с соответствующими Ролями.

Доступ к группам узлов возможен только для групп указанных типов по причинам:

- домен имеет иерархическую структуру;
- доступ осуществляется по протоколу LDAP через глобальный порт центрального узла.

Записи без указанных типов, расположенные на центральном узле, могут не вызывать данной проблемы. Типы групп в этом случае не важны.

### 3.9 Установка модуля распределённого хранения документов

Модули распределенного хранения документов (РХД) устанавливаются **только** на серверах региональных ОДУ и РДУ с целью обеспечения доступности документов и снижения трафика на центральный узел.

#### 3.9.1 Установка модуля локального хранилища

1. Запустить исполняемый файл **ИС СРЗА. Модуль распределенного файлового хранилища*.exe**.
2. Выбрать язык установки:

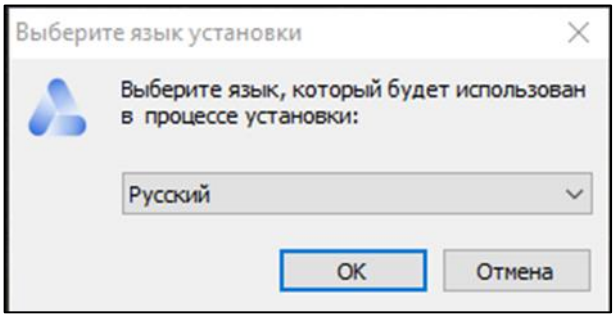


Рисунок 2. Выбор языка установки

3. Указать путь к папке, где будут храниться файлы хранилища.

**Внимание!** Рекомендуем предварительно определиться с местом хранения документов и создать папку для хранения, желательно на другом логическом диске, например **d:\dist_fs_files**

**Внимание!** Для сохранности документов хранилища необходимо настроить резервное копирование указанной папки.

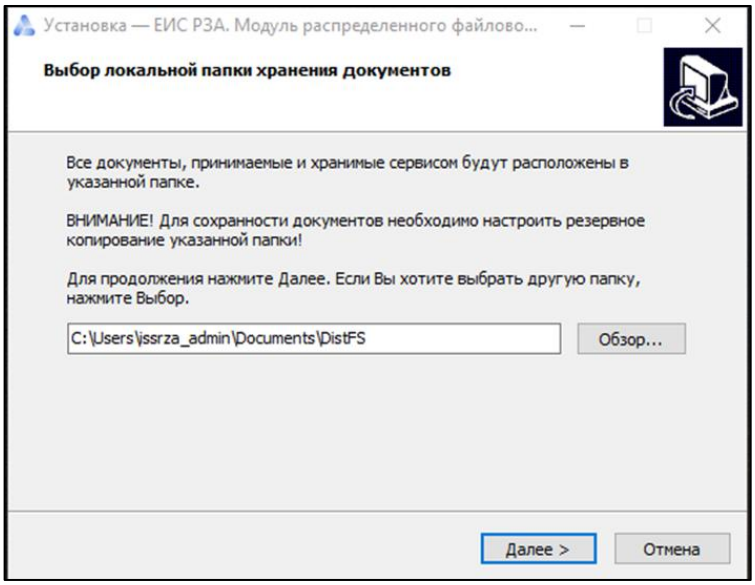


Рисунок 3. Выбор пути к папке для хранения файлов

4. Указать адрес и порт центрального сервера ИС СРЗА (адрес стенда с протоколом https и портом 443) и Сервер для устанавливаемого (текущего) модуля в виде Доменного наименование сервера или Адрес и порт в поле «Адрес и порт текущего модуля хранилища» (сервер, на который производится установка).

**Внимание!** Для корректной работы службы по протоколу https адреса необходимо задавать в виде **dns-записи.**

**Внимание!** Необходимо наличие двунаправленного сетевого доступа между модулем локального хранения и центральным сервером в составе Веб-сервера и Сервера приложения Системы:

| | | | |
|---|---|---|---|
| Веб-сервер ИС СРЗА | → | Модуль локального хранения | Порт, указанный при установке (по умолчанию 8002) |
| | ← | | 443 |
| Сервер приложения ИС СРЗА | → | | Порт, указанный при установке (по умолчанию 8002) |
| | ← | | 443 |

Центральный узел периодически проверяет доступность распределенных модулей хранения.

Данная проверка указывает на наличие сетевого доступа от центрального узла к хранилищу, а также то, что модуль распределенного хранения запущен.

**Внимание!** Для работы пользователя с хранилищем необходимо наличие сетевого доступа непосредственно от рабочего места пользователя к хранилищу. В случае отсутствия прямого сетевого доступа от рабочего места пользователя до модуля процедура загрузки файла через интерфейс Системы невозможна.

Проверка сетевой доступности хранилища для конечного пользователя осуществляется непосредственно перед началом загрузки/скачивания файла через интерфейс Системы.

В случае отсутствия прямого доступа к хранилищу при скачивании, запрос перенаправляется через центральный узел, и обеспечивает возможность получения файла через временное хранилище Системы.

1. Выбрать папку установки модуля:

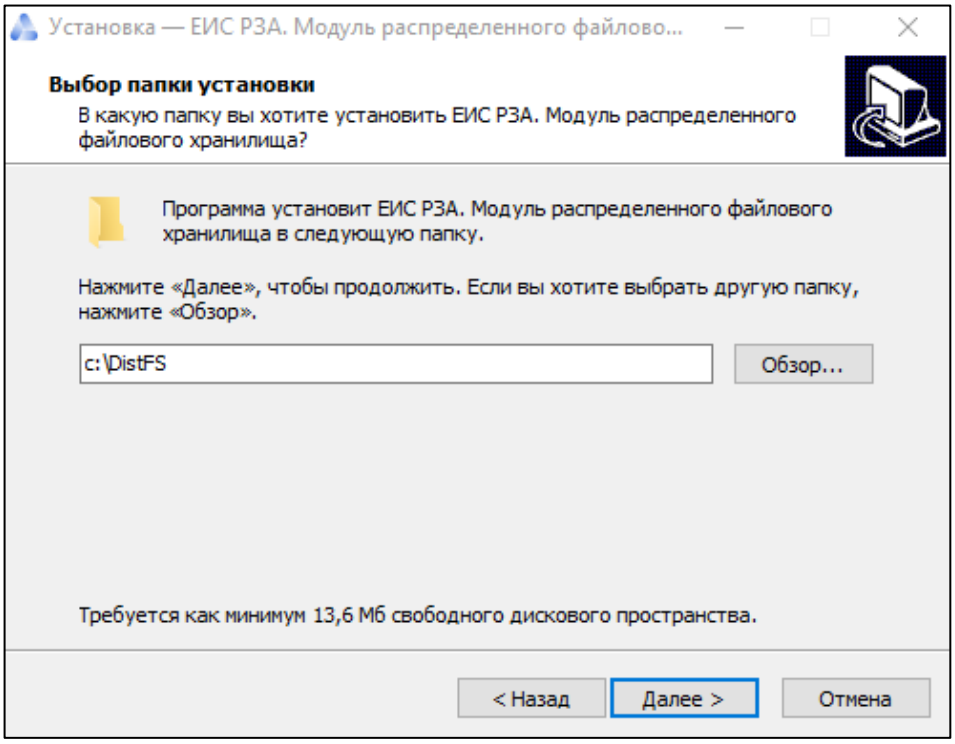


Рисунок 5. Выбор пути к папке, в которой будет установлен модуль

2. Выбрать папку в меню «Пуск»:

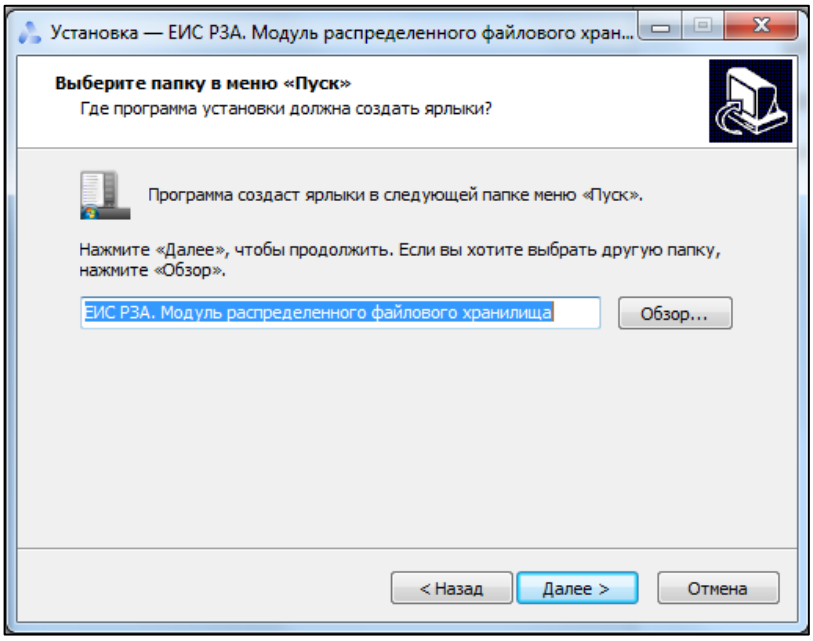


Рисунок 6. Создание ярлыка модуля в папке меню «Пуск»

3. Подтвердить начало установки:

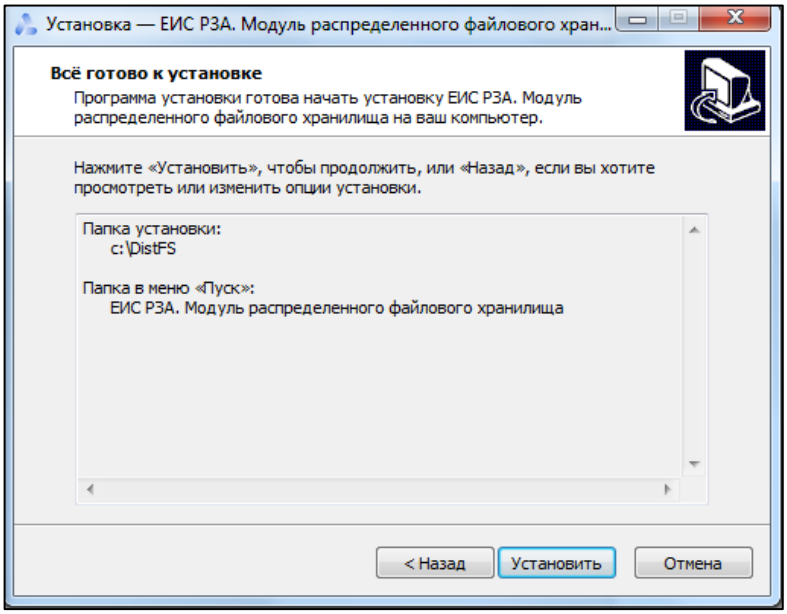


Рисунок 7. Подтверждение для начала установки модуля

4. После завершения установки в инструменте “**Управление компьютером**” - Службы должна появиться служба DistFS Service в статусе «Работает»:

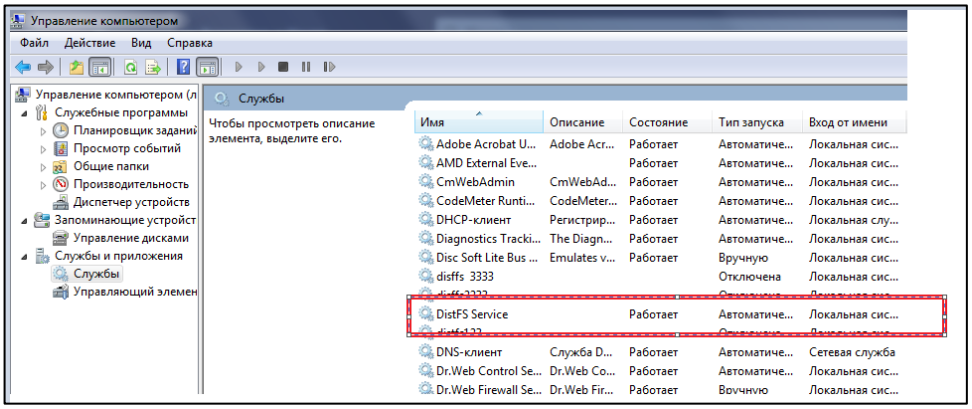


Рисунок 8. Просмотр состояния службы DistFS Service на Сервере модуля РХД.

### 3.9.2 Настройка доступа по протоколу HTTPS

Для адреса модуля, указанного при установке, необходимо сгенерировать ssl-ключи.

Приложение поддерживает ключи формата **crt/key** или **pem**

Сертификаты должны иметь следующие названия: ***certificate.crt***, ***private.key*** или ***certificate.pem***, ***private.pem***. Если полученные ключи имеют другое наименования, то требуется переименовать сертификаты.

Если сертификат предоставляется файлом с расширением **pfx**, то его необходимо конвертировать в формат **crt/key** по инструкции из раздела 3.7 пункт 2. Выполнять конвертацию желательно в ОС Linux.

Сертификат должен быть создан для DNS имени, заданного при установке (рисунок 9). Если для сертификата требуется задать несколько DNS имен, необходимо задавать их корректно, согласно правилам того ПО, в котором создается сертификат.

В случае, если сертификат был некорректно сконвертирован, служба модуля РХД может не запуститься или зависнуть. В этом случае необходимо проводить конвертацию сертификата из **pfx** в **crt/key** в ОС Linux или изначально создавать и использовать сертификаты в **pem**-формате.

1. Полученные сертификаты необходимо скопировать в папку keys, расположенную в папке установки модуля.

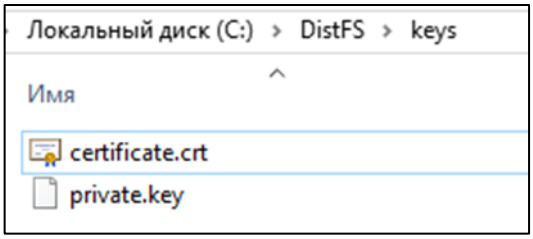


Рисунок 10. Файлы ssl-ключей в папке keys модуля.

2. Перезапустить службу **DistFS Service** в инструменте **“Управление компьютером” - Службы**

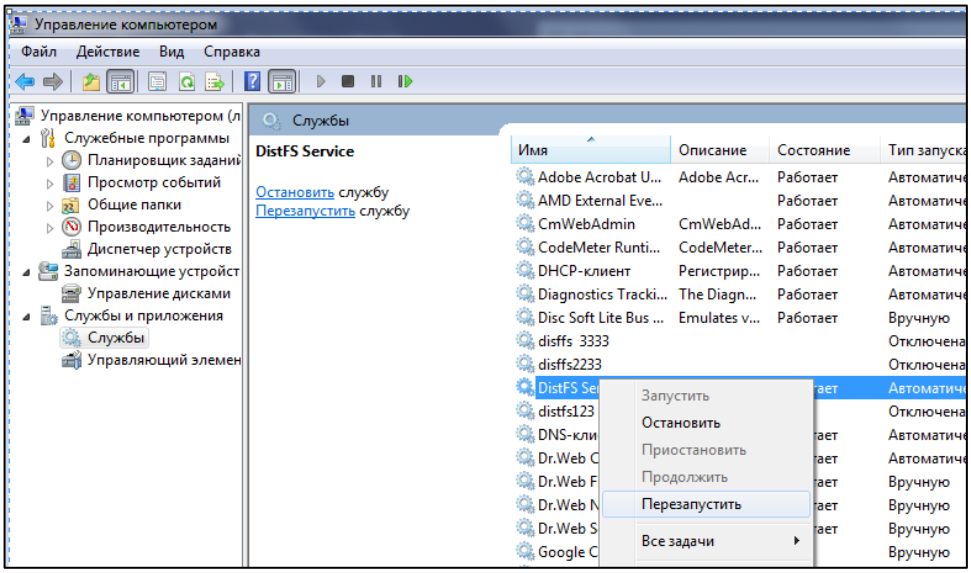


Рисунок 11. Перезапуск службы DistFS Service для настраиваемого модуля

### 3.9.3 Запуск модуля локального хранилища

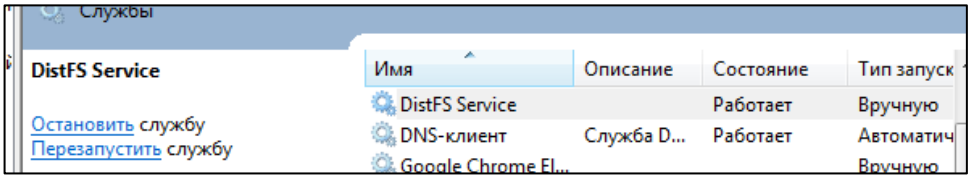


Рисунок 12. Запуск приложения

Запуск, останов и перезапуск службы осуществляется с помощью инструмента **«Управление компьютером» - Службы**

Информация о работе модуля («Логи») располагаются в приложении **«Просмотр событий»:**

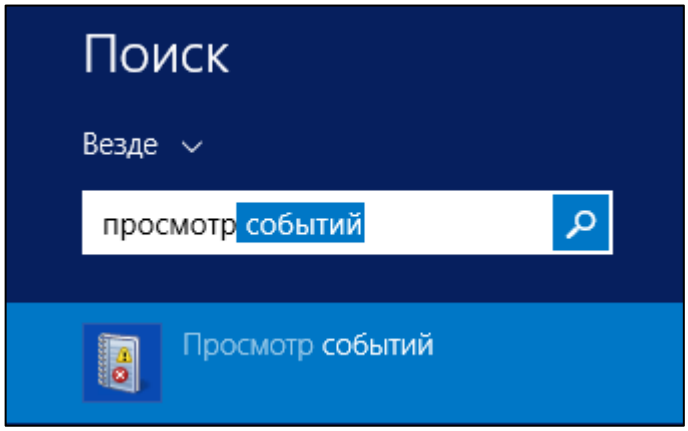


Рисунок 13. Просмотр событий (логи)

Журнал событий (Event Viewer) -> Журналы Windows (Windows Logs) -> Приложение (Application).

Загруженные в хранилище файлы хранятся по пути, указанному в инструкции по установке:

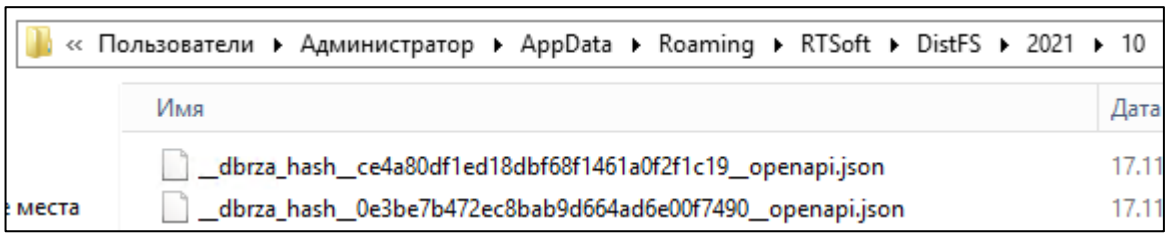


Рисунок 14. Просмотр файлов, хранящихся в установленном модуле РХД

### 3.9.4 Подключение модуля локального хранилища к ИС СРЗА

Добавить хранилище в графическом интерфейсе системы ИС СРЗА.

При правильной настройке в течение 1 минуты Статус переключится на зеленый.

Для проверки доступности модуля для конечного пользователя достаточно ввести в строке браузера. Можно задать IP-адресом или DNS-именем:

<ссылка_на_хранилище>/ping

При наличии связи появится следующее сообщение:

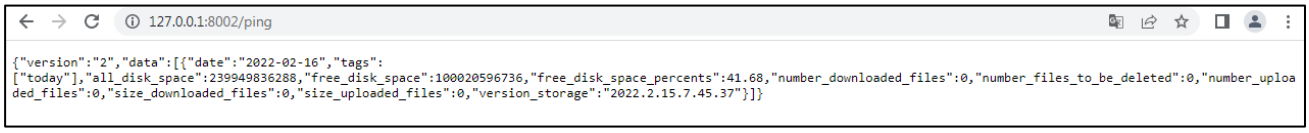

Рисунок 16. Положительный ответ модуля
при проверке доступности конечному пользователю

### 3.9.5 Настройка очистки хранилища от удаленных файлов

Начиная с версии 0.3 модуля РХД добавлена возможность очистки хранилища от файлов, которые были помечены как удаленные (добавлено расширение .deleted)

Запуск очистки может быть выполнен в ручном режиме путем выполнения команды:

<путь_к_модулю_рхд>\distfs.exe clean

, где:

<путь_к_модулю_рхд> - путь, к модулю, заданному при установке


Рисунок 17. Ввод команды в консоли

Также запуск очистки может быть добавлен в планировщик задач Windows:

1. Открыть **Планировщик задач Windows** и нажать «**Создать простую задачу**».

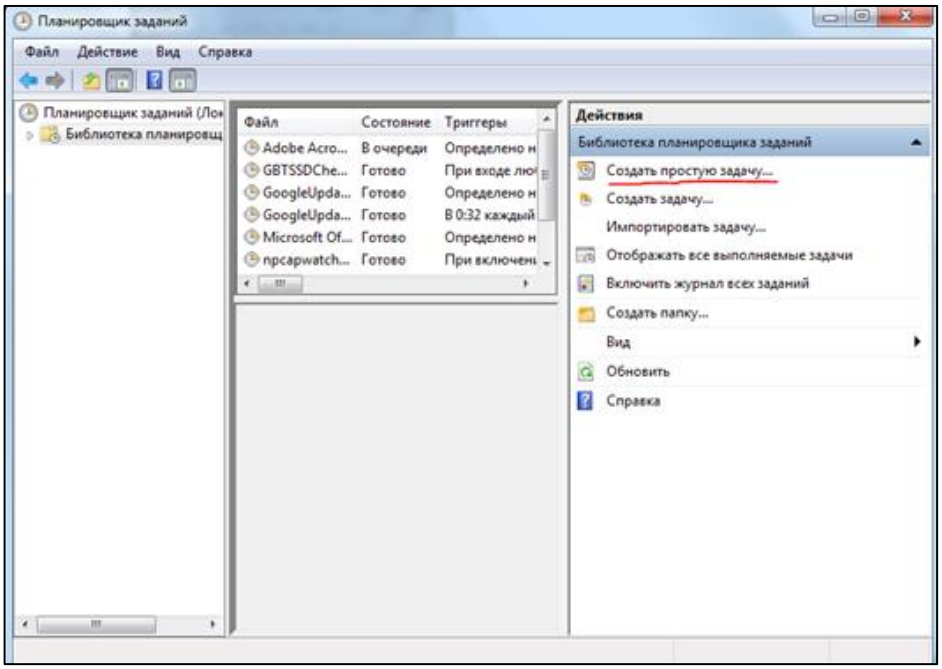


Рисунок 18. Планировщик задач в Windows

2. Задать название задачи:

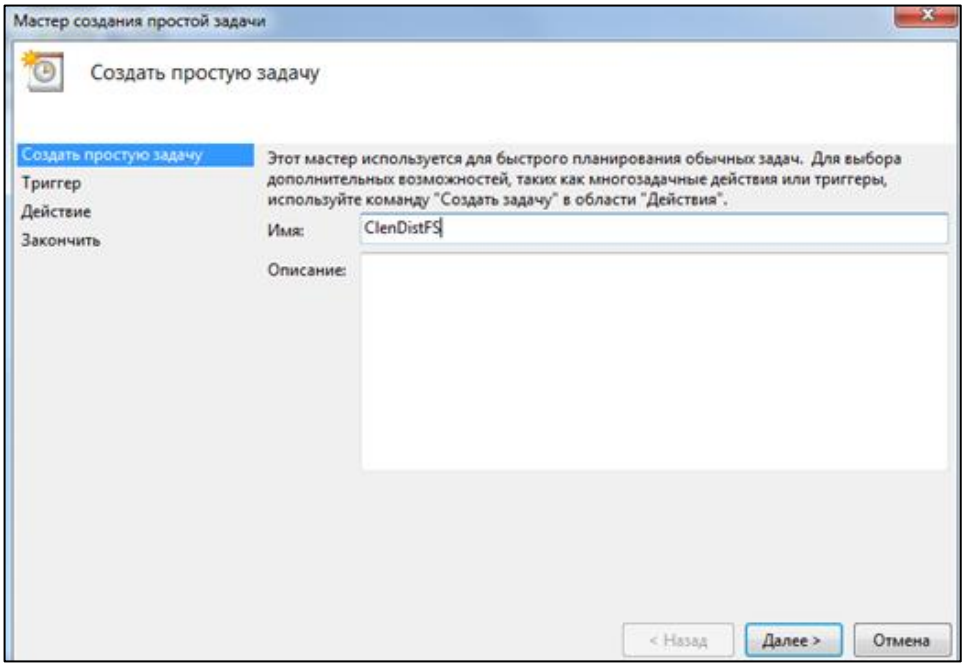


Рисунок 19. Создание задачи

3. Задать интервал запуска:

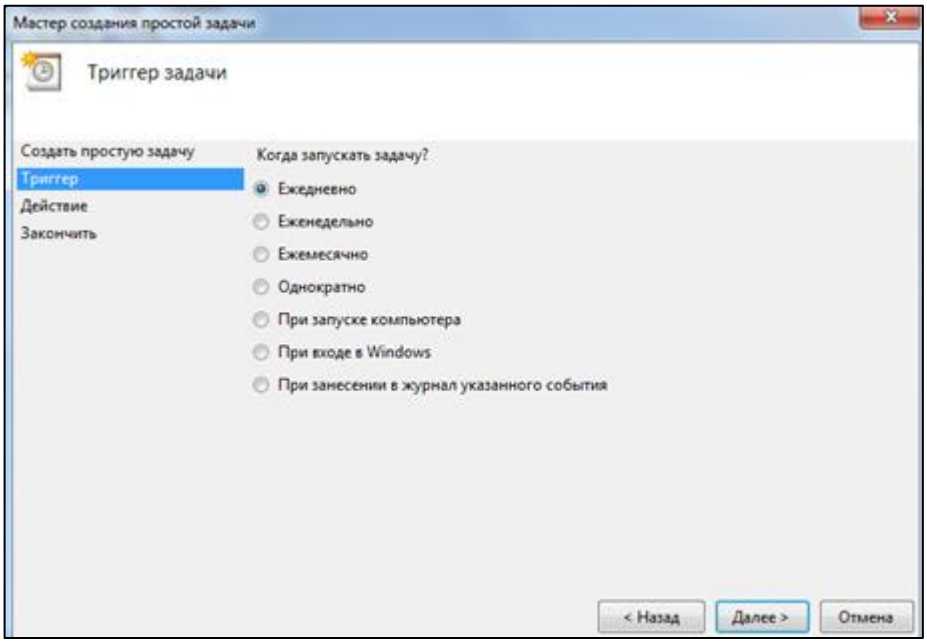


Рисунок 20. Добавление интервала запуска

4. Задать время запуска:

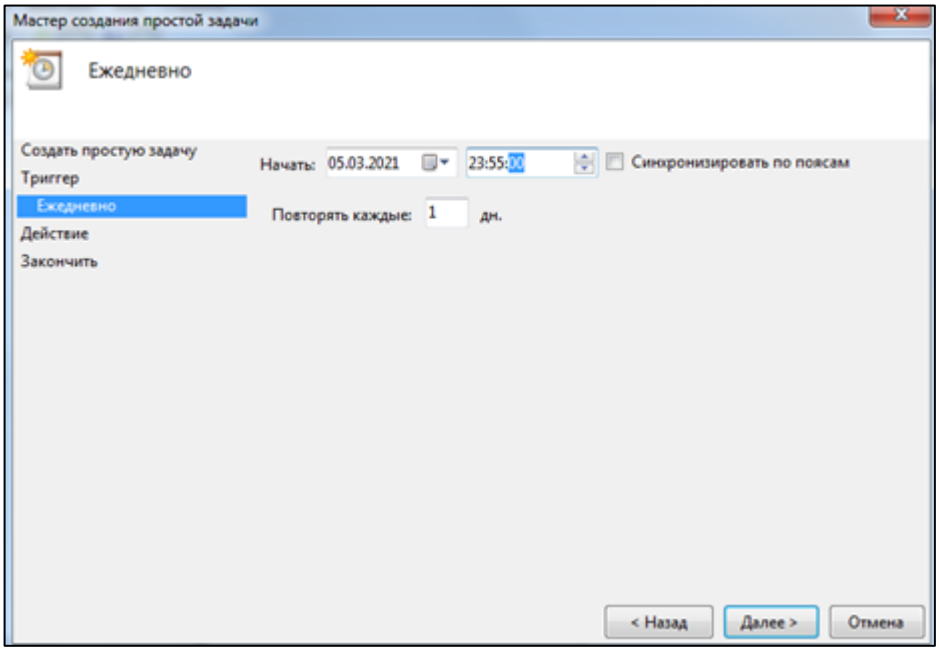


Рисунок 21. Добавление времени запуска

5. Выбрать «Запустить программу»:

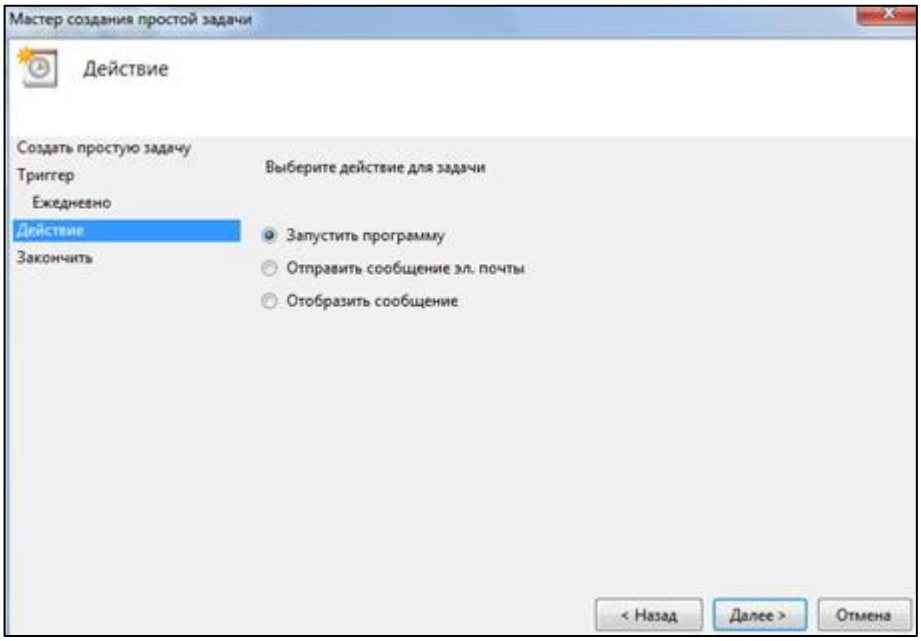


Рисунок 22. Добавление действия в виде запуска программы

6. Выбрать в качестве программы исполняемый файл модуля РХД:

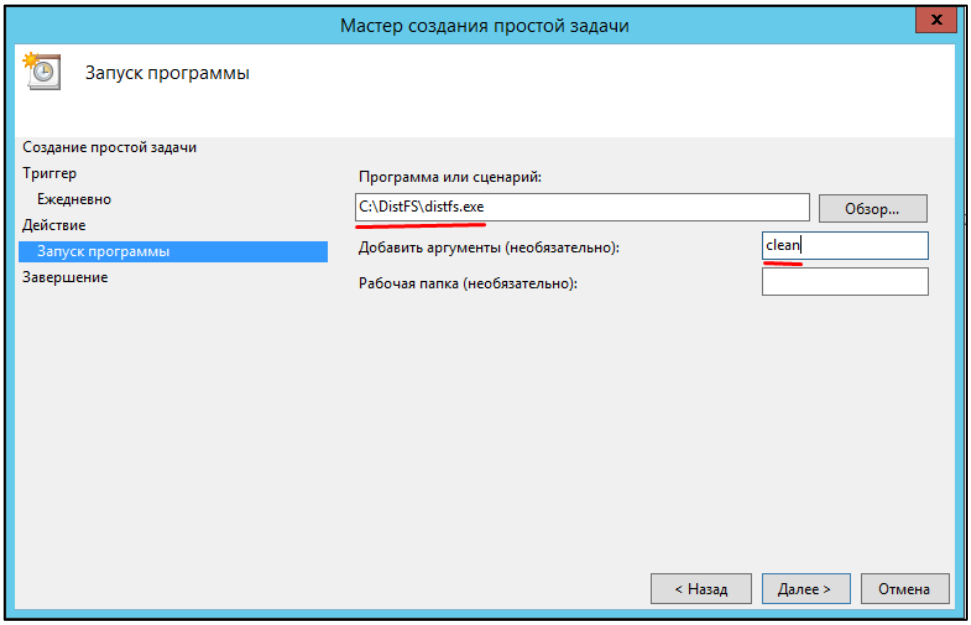


Рисунок 23. Выбор запуска сценария «Программа модуля РХД»

7. Поставить галочку «Открыть окно «Свойства»…»:

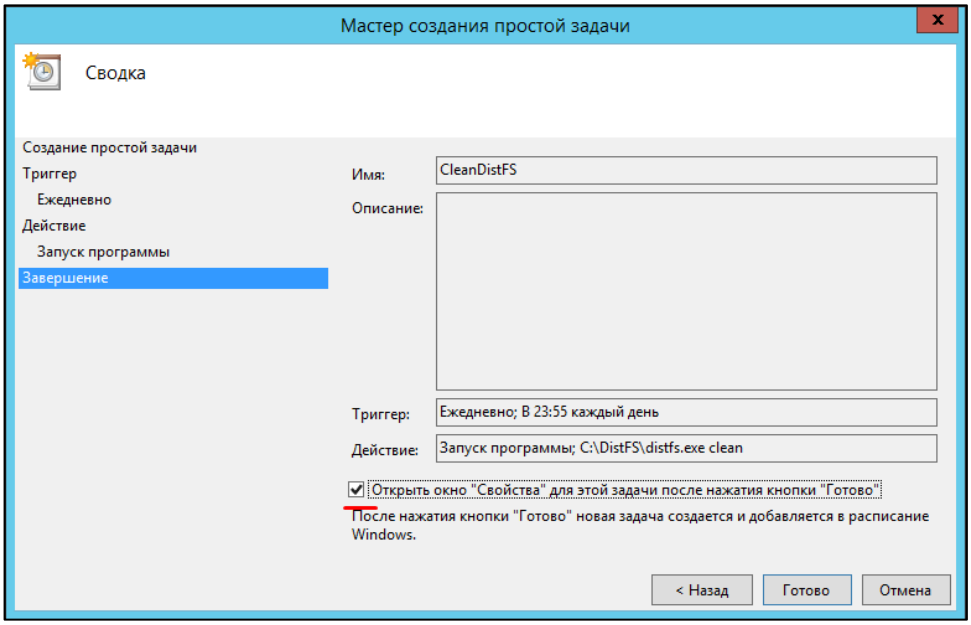


Рисунок 24. По окончании создания задачи выбрать «Открыть окно Свойства»

8. В окне свойств задать учетную запись, от которой будет запускаться процедура очистки:

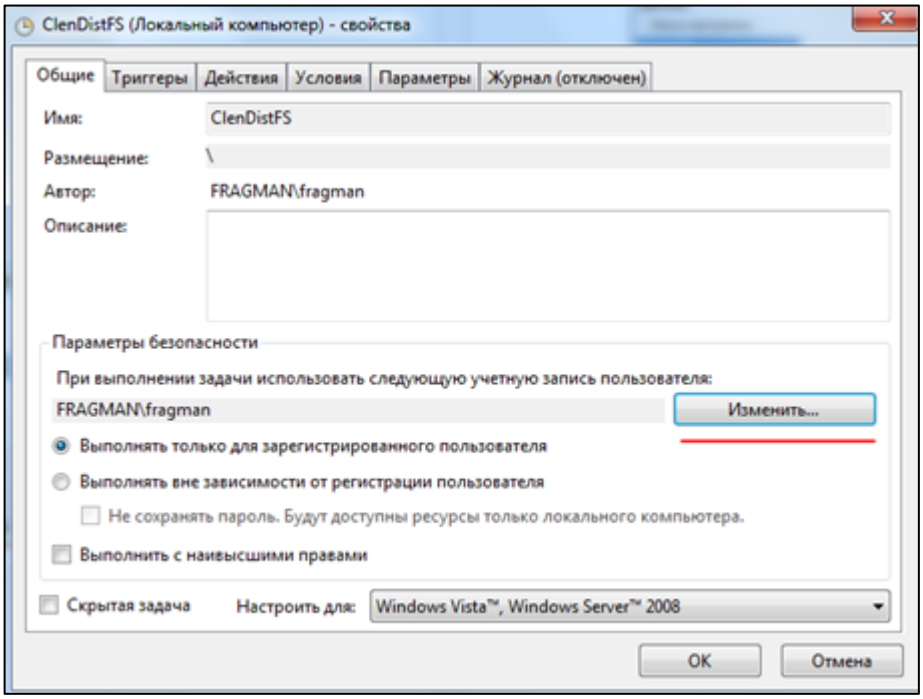


Рисунок 25. Задание учетной записи, от имени которой будет производиться процедура очистки

9. Перейти на <ip-адрес приложения>/admin
10. Найти вкладку «Настройки хранилища».
11. По умолчанию стоит настройка default.storage.setting.name, ее значение установлено на 60 дней.
12. Чтобы задать новые параметры, нужно нажать на кнопку «Добавить настройки хранилища».
13. Адрес необходимо задавать точно такой же при установке хранилища, но без порта. Также нужно задать количество дней, по прошествии которых будут очищаться удаленные файлы.

## 3.10 Установка и настройка сервера мониторинга (опционально)

На сервере с установленной ОС Ubuntu Server необходимо произвести следующие действия в описываемой последовательности:

1. Установить **ПО InfluxDB** из пакета **influxdb**, входящего в состав комплекта поставки ПО Системы, с помощью запуска следующих команд в окне терминала сервера от имени пользователя, обладающего правами администратора:

```
sudo apt-get install influxdb
sudo systemctl unmask influxdb.service
sudo systemctl start influxdb
```

2. Установить **ПО Grafana** из пакета **grafana**, входящего в состав комплекта поставки ПО Системы с помощью запуска следующих команд в окне терминала сервера от имени пользователя, обладающего правами администратора:

```
sudo apt install grafana -y
sudo systemctl start grafana-server
sudo systemctl enable grafana-server
```

3. Настроить **ПО InfluxDB** и **Grafana** с помощью соответствующих скриптов, входящих в состав комплекта поставки ПО Систем, с помощью запуска следующих команд в окне терминала сервера от имени пользователя, обладающего правами администратора:

```
influxd -config /etc/influxdb/influxdb.conf
echo $INFLUXDB_CONFIG_PATH
/etc/influxdb/influxdb.conf
influxd
```

# 4 ДЕЙСТВИЯ, ВЫПОЛНЯЕМЫЕ ПЕРЕД ПЕРВЫМ ЗАПУСКОМ ПО

При успешном выполнении всех действий, описанных в разделе 3 настоящей инструкции ПО Системы будет установлено и настроено необходимым для работы Системы образом. После завершения работы инсталляционных пакетов на каждом из серверов происходит автоматический запуск необходимых сервисов, Система полностью готова к работе и не требуется выполнять какие-либо специальные действия для запуска ПО.

Для проверки правильности функционирования ПО Системы необходимо выполнить пункты, описанные в разделе 6 настоящей инструкции.

После завершения работ по инсталляции ПО Системы на серверах и АРМ пользователей можно начинать работу с Системой с помощью клиентских приложений – «тонкого» клиента, функционирующего на основе пользовательского веб-браузера.

# 5 ПРИМЕРЫ ФАЙЛОВ КОНФИГУРАЦИИ ИНТЕГРАЦИОННЫХ СЕРВИСОВ

## 5.1 AccountService


```
{
 "Logging": {
  "LogLevel": {
   "Default": "Warning"
  }
 },
 "ConnectionStrings": {
  "DevServer": "Server=ip; Port=5000; Database=dbrza; User Id=postgres; Password=<pass> Trust Server Certificate=true; SslMode=Prefer",
 },
 "AccountService.Model": {
  "CurrentConnectionStringName": "DevServer",
  "ProviderType": "PostgreSQL"
 },
 "Divisionz.Scaffold.WebApi.AccountService.Model": {
  "CurrentConnectionStringName": "DevServer",
  "ProviderType": "PostgreSQL"
 },
 "Divisionz.Scaffold.WebApi.System.Model": {
  "CurrentConnectionStringName": "DevServer",
  "ProviderType": "PostgreSQL"
 },
 "AllowedHosts": "*",
 "Divisionz.Scaffold.WebApi.Authentication": {
  "CurrentAuthenticationSchema": "Bearer",
  "Windows": {
   "Sysadmin": "<domain>\...",
   "DEPath": "GC://<domain>",
   "Username": "username",
   "Password": "<pass>"

  },
  "Bearer": {
   "AuthOptions": {
```

```
      "ValidateIssuer": true,
      "XMLPublic":
"<RSAKeyValue><Modulus>….Modulus><Exponent>AQAB</Exponent></RSAKey
Value>",
      "PublicKey": "….",
      "Issuer": "Divisionz",
      "ValidateAudience": true,
      "Audience": "http://localhost:59392",
      "ValidateLifetime": true,
      "LifeTime": 1500
    }
   }
  }
}
```

### 5.2 ScorpLogService

```
{
 "ConnectionStrings": {
  "DevServer": "Server=<IP>; Port=5000; Database=dbrza; User Id=<uname>;
Password=<pass>; Trust Server Certificate=true; SslMode=Prefer",
 },
 "CimServiceClientManager": {
  "endpoint": "http://ip:8080/",
  "login": "system",
  "password": "<pass>"
 },
 "ScorpLogService.Model": {
  "CurrentConnectionStringName": "DevServer",
  "ProviderType": "PostgreSQL"
 },
 "Divisionz.Scaffold.ScaffoldSettings": {
  "SystemUserName": "system",
  "ApiPrefix": "api/…",
  "ServiceId": "…."
 },
 "Divisionz.Scaffold.WebApi.System.Model": {
  "CurrentConnectionStringName": "DevServer",
  "ProviderType": "PostgreSQL"
```

```
  },
  "Divisionz.Scaffold.WebApi.Authentication.Model": {
    "CurrentConnectionStringName": "DevServer",
    "ProviderType": "PostgreSQL"
  },
  "Service": {
    "ApiPrefix": "api/…",
    "Id": "…..",
    "Name": "ScorpLogService"
  },
  "ClientManagerBase": {
    "endpoint": "http://ip:6036/",
    "login": "system",
    "password": "<pass>"
  },
  "Divisionz.Scaffold.WebApi.Authentication": {
    "CurrentAuthenticationSchema": "Bearer",
    "Bearer": {
      "AuthOptions": {
        "ValidateIssuer": true,
        "XMLPublic": "<RSAKeyValue><Modulus…",
        "PublicKey": "….",
        “Issuer”: “Divisionz”,
        “ValidateAudience”: true,
        “Audience”: “http://localhost:59392”,
        «ValidateLifetime»: true,
        «LifeTime»: 15000
      }
    }
  }
}
```

### 5.3 CimService

```
{
  “ConnectionStrings”: {
    “DevServer”: “Server=ip; Port=5000; Database=dbrza; User Id=<name>; Password
Trust Server Certificate=true; SslMode=Prefer"
  },
```

```
  "CimService.Model": {
    "CurrentConnectionStringName": "DevServer",
    "ProviderType": "PostgreSQL"
  },
  "Divisionz.Scaffold.WebApi.System.Model": {
    "CurrentConnectionStringName": "DevServer",
    "ProviderType": "PostgreSQL"
  },
  "Divisionz.Scaffold.ScaffoldSettings": {
    "ApiPrefix": "api/…",
    "ServiceId": "……",
    "ServiceName": "CimService"
  },
    "Service": {
    "ApiPrefix": "api/…",
    "Id": "…..",
    "Name": "CimService"
  },
  "ClientManagerBase": {
    "endpoint": "http://ip:6036/",
    "login": "system",
    "password": "<pass>"
  },
  "Divisionz.Scaffold.WebApi.Authentication": {
    "CurrentAuthenticationSchema": "Bearer",
    "Bearer": {
      "AuthOptions": {
        "ValidateIssuer": true,
        "XMLPublic":
"<RSAKeyValue><Modulus>….</Modulus><Exponent>AQAB</Exponent></RSAKe
yValue>",
        "PublicKey": "….",
        "Issuer": "Divisionz",
        "ValidateAudience": true,
        "Audience": "http://localhost:59392",
        "ValidateLifetime": true,
        "LifeTime": 15000
      }
    }
```

```
  }
}
```

### 5.4 AipIntegration

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "CimServiceClientManager": {
    "account": "http://ip:6036",
    "endpoint": "http://ip:8080",
    "login": "system",
    "password": "<pass>"
  },
  "ScorpLogServiceClientManager": {
    "account": "http://ip:6036",
    "endpoint": "http://ip:6025",
    "login": "system",
    "password": "<pass>"
  },
  "AllowedHosts": "*",
  "DebugMode": false,
  "DefaultModelVersionOptions": {
    "Model": "…",
    "Version": 2043,
        "GIDModel": "…"
  },
  "DimServiceClientOptions": {
    "Login": "name",
    "Password": "<pass>",
    "Domain": "<domain>",
    "endpoint": "http://ip",
    "timeOut": 15
  },
```

```
"BeforeMergeScriptPath": [
  "/app/cimservice.importer/dbrza/dbrza_startIntegration"
],
"AfterMergeScriptPath": [
  "/app/cimservice.importer/dbrza/dbrza"
],
"GetDimObjectsServiceOptions": {
  "DebugDataDirectory": "Services//Fake//Data//Data"
},
"ScheduleTime": {
    "Hour": 2,
    "Minute": 0,
    "Interval": 24
},
"SupportedProtectionDeviceFunctionClasses": {
  "Map": [
    {
      "Id": " …
      "Name": "RelayProtection"
    },
    {
      "Id": "… ",
      "Name": "ControlAutomation"
    },
    {
      "Id": "… ",
      "Name": "EmergencyControlAutomation"
    },
    {
      "Id": "… ",
      "Name": "RegulatingAutomation"
    },
    {
      "Id": "… ",
      "Name": "DisturbanceRecorder"
    },
    {
      "Id": "…",
      "Name": "OperationControlAutomation"
```

```
    },
    {
     "Id": "…",
     "Name": "ProtectionEquipment"
    },
    {
     "Id": "…",
     "Name": "AutomaticLoadTransfer"
    },
    {
     "Id": "…",
     "Name": "AutomaticLoadTransferBD"
    },
    {
     "Id": "…",
     "Name": "AutomaticLoadTransferUD"
    },
    {
     "Id": "…",
     "Name": "UnderfrequencyLoadShedding"
    }
  }
}
```

### 5.5 LanDocs

```
{
   "Исполнительный аппарат": {
   "SSOKey": "…",
   "Endpoint": "https://...",
   "JournalConfigs":
    [
     {
      "LandocsJournalType": "In",
      "LandocsJournalId": 340638
     },
     {
      "LandocsJournalType": "Out",
      "LandocsJournalId": 340639
```

```
      },
      {
        "LandocsJournalType": "Service",
        "LandocsJournalId": 340645
      }
    ]
  },
"": {
  "SSOKey": "…",
  "Endpoint": "https://...",
  "JournalConfigs":
    [
      {
        "LandocsJournalType": "In",
        "LandocsJournalId": 340638
      },
      {
        "LandocsJournalType": "Out",
        "LandocsJournalId": 340639
      },
      {
        "LandocsJournalType": "Service",
        "LandocsJournalId": 340645
      }
    ]
  },
```

# 6 ПРОВЕРКА ПРАВИЛЬНОСТИ ФУНКЦИОНИРОВАНИЯ

Проверка правильности функционирования Системы производится Администратором в ИА.

Для проверки правильности функционирования Системы следует:

1) С помощью веб-браузера, установленного на АРМ пользователя выполнить авторизацию на веб-сайте Системы.
2) На убедиться в отсутствии актуальных SNMP сообщений о критических системных ошибках.
3) Убедиться в отсутствии сообщений о критических системных ошибках в лог-файлах Системы.
4) На веб-сайте Системы в журнале системных событий применить фильтр «Системные ошибки» и убедиться в отсутствии свежих записей о системных ошибках.

# 7 ОБНОВЛЕНИЕ ПО

Обновление ПО необходимо выполнять в следующем порядке:

1. Сервер приложений
2. Веб-сервер

## 7.1 Сервер приложений

1. Перенести все файлы из папки backend на сервер приложений (в каталог /home/rza).
2. Сделать бэкап прошлой версии:

   # sudo cp -r /opt/dbrza/backend /opt/dbrza/backup/

3. Скопировать новые файлы:

   # sudo cp docker-compose.yml /opt/dbrza/backend

4. Перейти в директорию приложения:

   # cd /opt/dbrza/backend

5. Залогиниться в хранилище:

   # sudo docker login -u "ivanov-ma" -p "панель_от_хранилища" "registry.local"

6. Скачать/обновить образы докера:

   # sudo docker-compose pull

7. Выключить приложение:

   # sudo docker-compose down

8. Запустить обновленную версию:

   # sudo docker-compose up -d

9. Скопировать статику из контейнера на сервер:

   # sudo docker cp dbrza_backend_1:/code/static .

10.Перенести ее на веб-сервер:

# sudo scp -r static rza@[адрес_веб-сервера]:/home/rza

### 7.2 Веб-сервер

1. Перенести все файлы из папки frontend на веб-сервер (в каталог /home/rza).
2. Скопировать новые файлы:

   # sudo cp /home/rza/html/* /usr/share/nginx/.

3. Скопировать статику в каталог фронта:

   # sudo cp -r static /usr/share/nginx/html/.

4. Перезагрузить nginx:

   # sudo systemctl restart nginx

# 8 УДАЛЕНИЕ ПО

Удаление ПО Системы должно производиться пользователем, имеющим права администратора сервера.

Для удаления ПО Системы следует:

1) На Сервере приложений остановить докер-контейнеры:

```
# docker-compose down
```

2) Удалить содержимое директории, в которой расположен файл **docker-compose.yml**.
3) На Сервере баз данных остановить докер-контейнеры:

```
# docker-compose down
```

4) Удалить содержимое директории, в которой расположен файл **docker-compose.yml**.