«Информационная система «Система автоматизированного планирования электроэнергетических режимов» (САПЭР 2022)

ИНСТРУКЦИЯ ПО УСТАНОВКЕ И НАСТРОЙКЕ

Версия 1.4.1

Москва 2023

СОДЕРЖАНИЕ

1. Основные термины, определения и сокращения

AD	Служба каталогов, являющаяся единым хранилищем данных организации и контролирующая доступ для пользователей на основе политики безопасности каталога.
API	Описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой.
DNS	Компьютерная распределённая система для получения информации о доменах.
Docker	Программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации.
HTTP	HyperText Transfer Protocol – протокол прикладного уровня передачи данных.
HTTPS	HyperText Transfer Protocol Secure – расширение протокола HTTP для поддержки шифрования в целях повышения безопасности.
Java	Строго типизированный объектно-ориентированный язык программирования общего назначения.
JavaScript	Прототипно-ориентированный сценарный язык программирования.
JSON	Текстовый формат обмена данными, основанный на JavaScript.
LDAP	Протокол взаимодействия со службой каталогов (AD).
LDAPS	LDAP с поддержкой SSL.
Nexus	Менеджер репозиториев предназначенный для проксирования репозиториев и хранения ПО.
REST	Архитектурный стиль взаимодействия компонентов распределённого приложения в сети. REST представляет собой согласованный набор ограничений, учитываемых при проектировании распределённой гипермедиа-системы.
SNMP	Протокол, используемый для управления сетевыми устройствами.
SSL	Криптографический протокол, обеспечивающий защищённую передачу данных между узлами в сети.
SSH	Протокол удаленного управления компьютером с операционной системой Linux.
CPU	Центральный процессор.
RAM	Оперативная память.
HDD	Жесткий диск.
БД	База данных.
ИА	Исполнительный аппарат АО «СО ЕЭС».
ИК	Исходный код.
ИУС	Информационно-управляющая системы.
ПАК	Программно-аппаратный комплекс.

ПАК ЕСМ	ПАК «Единая система мониторинга».
ПО	Программное обеспечение.
СУБД	Система управления базами данных.
УЗ	Учётная запись.

2. Назначение руководства

Инструкция описывает действия системного администратора и администратора системы комплексного технического обслуживания (далее - КТО) по установке и настройке программы для электронных вычислительный машин «Информационная система «Система автоматизированного планирования электроэнергетических режимов» (далее – САПЭР 2022, Система).

Перечисленные в инструкции команды выполняются с использованием SSH-клиента, например – PuTTY.

Специалист, выполняющий действия по установке и настройке Системы, должен владеть базовыми знаниями по:

- Администрированию ОС Astra Linux и Windows Server 2019;
- Работе с Git репозиториями;
- Администрированию СУБД Postgres;
- Администрированию IIS.

Установка и настройка Системы выполняется:

- Системным администратором – главы 3-6;
- Администратором системы КТО – глава 7.

3. Требования к программным / аппаратным ресурсам

3.1. Требования к аппаратному обеспечению

Рекомендованные характеристики серверов указаны в таблице 1.

Таблица 1 – Рекомендуемая конфигурация серверов Системы

№	Серверы	Кол-во серверов	Рекомендованные характеристики серверов			
			CPU, core	RAM, Gb	HDD, Gb	avx
1	Вэб-сервер	2	2	4	30	
2	Сервер приложений	2	8	16	50	
3	Сервер расчетов	2	8	26	50	X
4	Брокер сообщений	3	2	6	40	
5	Сервер оптимизации	2	4	10	60	
6	Сервер СУБД	2	4	32	12520	
	Итого	**13**	**58**	**194**	**25540**	

Рекомендованные характеристики сервера соответствуют одному серверу. В строке «Итого» указана итоговая сумма ресурсов для всех серверов.

Таблица 2 отражает минимальные аппаратные характеристики серверов Системы.

Таблица 2 – Минимальная конфигурация серверов Системы

№	Серверы	Кол-во серверов	Минимальные характеристики серверов			
			CPU, core	RAM, Gb	HDD, Gb	avx
1	Вэб-сервер	1	2	4	30	
2	Сервер приложений	2	8	16	50	
3	Сервер расчетов	1	8	26	50	X
4	Брокер сообщений	3	2	6	40	
5	Сервер оптимизации	1	4	10	60	
6	Сервер СУБД	1	4	32	12520	
	Итого	**9**	**40**	**122**	**13020**	

Минимальные характеристики сервера соответствуют одному серверу. В строке «Итого» указана итоговая сумма ресурсов для всех серверов.

3.2. Требования к программному обеспечению

На веб-серверах должно быть установлено следующее ПО:
- Операционная система – Astra Linux Special Edition 1.7.4+;
- Kaspersky Endpoint Security;
- ПО Keepalived;
- ПО Nginx версии 1.14.1+.
- NFS server;
- NFS client;

На серверах расчетов должно быть установлено следующее ПО:
- Операционная система – Astra Linux Special Edition 1.7.4+;
- Kaspersky Endpoint Security;
- ПО Docker Engine версии 19.03+;

На серверах приложений должно быть установлено следующее ПО:
- Операционная система – Astra Linux Special Edition 1.7.4+;
- Kaspersky Endpoint Security;
- ПО Docker Engine версии 19.03+;
- NFS client;

На серверах СУБД должно быть установлено следующее ПО:
- Операционная система – Astra Linux Special Edition 1.7.4+;
- Kaspersky Endpoint Security;
- ПО Keepalived;
- СУБД – Postgres Pro Standard версии 14+.

На серверах выполнявших роль брокеров сообщений:
- Операционная система – Astra Linux Special Edition 1.7.4+;
- Kaspersky Endpoint Security;
- ПО Apache Kafka.

На серверах оптимизации:
- Операционная система – Windows Server 2019+ RUS;
- Kaspersky Endpoint Security;
- Microsoft .Net Framework 4.7.2 +;
- Веб-сервер IIS с поддержкой ASP, ASP.NET 4.7;
- CMake;
- Microsoft Build Tools;
- .NET 5.0 Runtime.

3.3. Предварительная настройка окружения

Для запуска Системы необходимо:

1. Для каждого кластера keepalived необходимо запросить общий IP адрес.
2. Зарегистрировать DNS запись для вэб интерфейса Системы на общий IP адрес, присвоенный кластеру keepalived серверов балансировки нагрузки.
3. Выпустить SSL сертификаты в PEM[1] формате для сайта Системы.

Если сертификаты предоставлены в формате PFX необходимо произвести конвертацию сертификата в PEM формат. Для конвертации рекомендуется использовать библиотеку *openssl*, документация для ПО доступна по ссылке: https://www.openssl.org/docs/manmaster/man1/openssl.html

Пример конвертации сертификата с именем my.pfx:

```
openssl pkcs12 -in ~/my.pfx -clcerts -nokeys -out
/var/www/html/web.crt
openssl pkcs12 -in ~/my.pfx -nocerts -out ~/my.key
openssl rsa -in ~/my.key -out /var/www/html/web.key
```

4. Создать в AD сервисную УЗ для Системы (необходимо, чтобы сервисная УЗ имела одинаковые CommonName и SamAccountName).
5. Запросить УЗ для доступа к ФПА с исходным кодом Системы, а также для получения конфигурационных файлов и артефактов сборки.
6. Запросить УЗ для доступа к артефактам Системы расположенным на Nexus сервере ФПА.
7. Запросить для сервисных УЗ Системы доступ в ИУС СОДП ДП.
8. Запросить для сервисных УЗ Системы доступ в ИУС Энергия.
9. Запросить для сервисных УЗ Системы доступ в ИУС Балансы.
10. Запросить для сервисных УЗ Системы доступ в ПАК ЕСС.
11. Запросить для сервисных УЗ Системы доступ в ОИК СК-11.
12. Запросить у администратора МОПОП включение серверов расчетов в рассылку данных МОПОП.
13. Запросить для сервисных УЗ Системы доступ в ИУС СИМ ЗРП
14. Создать в AD для каждой роли используемой в Системе группы пользователей.

[1] Необходима пара ключей (открытый и закрытый ключ), расширения по умолчанию данной пары - .ctr и .key

Роль	Пример
Администратор системы	<домен уровня ИА>\<группа AD администратора системы>
Администратор НСИ ИА	<домен уровня ИА>\<группа AD администратора НСИ>
Технолог ИА	<домен уровня ИА>\<группа AD технолога>
Просмотр ИА	<домен уровня ИА>\<группа AD технолога>
Администратор НСИ ОДУ	<домен уровня ОДУ>\<группа AD администратора НСИ>
Технолог ОДУ	<домен уровня ОДУ>\<группа AD технолога>
Просмотр ОДУ	<домен уровня ОДУ>\<группа AD технолога>
Администратор НСИ РДУ	<домен уровня РДУ>\<группа AD администратора НСИ>
Технолог РДУ	<домен уровня РДУ>\<группа AD технолога>
Просмотр РДУ	<домен уровня РДУ>\<группа AD технолога>

15. Добавить УЗ пользователей Системы в созданные группы AD. (Поле Company должно советовать наименованию ДЦ)
16. Запросить УЗ для подключения к директории, используемой для резервного копирования Системы.

3.4. Сетевой доступ

Таблица 3 содержит список сетевых взаимодействий Системы.

Таблица 3 – Сетевое взаимодействий Системы

Источник	Приёмник	Протокол/Порт
Сервера приложений Системы		
Компьютер администратора Системы	Сервер приложений	TCP-22 (SSH), TCP-1111 (HTTP), TCP-10001 (HTTP), TCP-10002 (HTTP), TCP-10004 (HTTP), TCP-10005 (HTTP), TCP-10006 (HTTP),
Сервер ПАК ЕСМ	Сервер приложений	TCP-1111 (HTTP), TCP-10001 (HTTP), TCP-10002 (HTTP), TCP-10004 (HTTP), TCP-10005 (HTTP), TCP-10006 (HTTP), UPD-161
Сервер приложений	Сервер ПАК ЕСМ	UPD-162
Сервер приложений	Сервер приложений	TCP-1111 (HTTP), TCP-10005 (HTTP),
Сервер приложений	Веб-сервер	TCP-443 (HTTPS), TCP-111 (NFS), UDP-111 (NFS), TCP-2049 (NFS), UDP-2049 (NFS), TCP-32765-32768 (NFS), UDP-32765-32768 (NFS),
Веб-сервер	Веб-сервер	TCP-111 (NFS), UDP-111 (NFS), TCP-2049 (NFS), UDP-2049 (NFS), TCP-32765-32768 (NFS), UDP-32765-32768 (NFS),
Сервер приложений	Сервера СУБД	TCP-5432

Источник	Приёмник	Протокол/Порт
Сервер приложений	Брокер сообщений	TCP-2181,9092
Сервер приложений	Сервера AD (контроллеры домена)	TCP-636 (LDAPS)
Сервер приложений	Сервера AD (глобальный каталог)	TCP- 3268
Сервер приложений	Сервер ФПА – хранилище конфигурации	TCP-443 (HTTPS)
Сервер приложений	Сервер ФПА – хранилище артефактов	TCP-18181
Сервер приложений	Сервер точного времени	UDP-123
Сервера расчетов Системы		
Компьютер администратора Системы	Сервер расчетов	TCP-22 (SSH), TCP-10003 (HTTP), TCP-10007 (HTTP)
Сервер ПАК ЕСМ	Сервер расчетов	TCP-10003 (HTTP), TCP-10007 (HTTP), UPD-161
Сервер расчетов	Сервер ПАК ЕСМ	UPD-162
Сервер расчетов	Сервер приложений	TCP-1111 (HTTP), TCP-10005 (HTTP),
Сервер расчетов	Веб-сервер	TCP-443 (HTTPS)
Сервер расчетов	Сервера СУБД	TCP-5432
Сервер приложений	Брокер сообщений	TCP-2181, TCP-9092
Сервер расчетов	Сервер ФПА – хранилище конфигурации	TCP-443 (HTTPS)
Сервер расчетов	Сервер ФПА – хранилище артефактов	TCP-18181
Сервер расчетов	ИУС СОДП ДП	TCP-443 (HTTPS)
Сервер расчетов	ИУС Энергия	TCP-443 (HTTPS), TCP-80 (HTTP)
Сервер расчетов	ИУС Балансы	TCP-443 (HTTPS)
Сервер расчетов	ИУС ОИК-11	TCP-443 (HTTPS), TCP-9443 (HTTPS)

Источник	Приёмник	Протокол/Порт
Сервер расчетов	ИУС СИМ ЗРП	TCP-9876 (HTTP)
Сервер расчетов	ИУС ЕСС	TCP-5000 (HTTP)
Сервер расчетов	Сервер точного времени	UDP-123
Веб-сервера Системы		
Компьютер администратора Системы	Веб-сервер	TCP-22 (SSH), TCP-443 (HTTPS)
Пользователи Системы	Веб-сервер	TCP-443 (HTTPS)
Сервер ПАК ЕСМ	Веб-сервер	TCP-443 (HTTPS), UPD-161
Веб-сервер	Сервер ПАК ЕСМ	UPD-162
Веб-сервер	Сервер приложений	TCP-10001 (HTTP), TCP-10002 (HTTP), TCP-10004 (HTTP), TCP-10005 (HTTP), TCP-10006 (HTTP) TCP-111 (NFS), UDP-111 (NFS), TCP-2049 (NFS), UDP-2049 (NFS), TCP-32765-32768 (NFS), UDP-32765-32768 (NFS),
Веб-сервер	Веб-сервер	TCP-111 (NFS), UDP-111 (NFS), TCP-2049 (NFS), UDP-2049 (NFS), TCP-32765-32768 (NFS), UDP-32765-32768 (NFS),
Веб-сервер	Сервер приложений	TCP-10003 (HTTP), TCP-10007 (HTTP)
Веб-сервер	Сервер оптимизации	TCP-5000 (HTTP), TCP-5001 (HTTP)
Веб-сервер	Веб-сервер	VRRP
Веб-сервер	Сервера точного времени	UDP-123
Сервера оптимизации расчетов Системы		

Источник	Приёмник	Протокол/Порт
Компьютер администратора Системы	Сервер оптимизации	TCP-3389 (SSH), TCP-5000 (HTTPS), TCP-5001 (HTTPS),
Сервер ПАК ЕСМ	Сервер оптимизации	TCP-5000 (HTTPS), UPD-161
Сервер оптимизации	Сервер ПАК ЕСМ	UPD-162
Сервер оптимизации	Сервера точного времени	UDP-123
Брокер сообщений		
Компьютер администратора Системы	Брокер сообщений	TCP-22 (SSH), TCP-2181, TCP-9092
Сервер ПАК ЕСМ	Брокер сообщений	TCP-2181, TCP-9092, UPD-161
Брокер сообщений	Сервер ПАК ЕСМ	UPD-162
Брокер сообщений	Брокер сообщений	TCP-2181 (HTTP), TCP-2888 (HTTP), TCP-3888 (HTTP)
Брокер сообщений	Сервер точного времени	UDP-123
Сервера СУБД Системы		
Компьютер администратора Системы	Сервера СУБД	TCP-22 (SSH), TCP-5432
Сервер ПАК ЕСМ	Сервера СУБД	TCP-5432, UPD-161
Сервера СУБД	Сервер ПАК ЕСМ	UPD-162
Сервера СУБД	Сервера СУБД	TCP-22 (SSH), TCP-5432, VRRP
Сервера СУБД	Сервер точного времени	UDP-123

4. Установка компонентов системы

4.1. Предварительная настройка серверов Системы

Для установки Системы необходимо подготовить сервера с в соответствии с данными, указанными в разделе 3 настоящей инструкции.

Рекомендовано использование lvm для организации дисковой системы, так как необходимый объём HDD может быть изменен в процессе эксплуатации.

Так же необходимо настроить репозиторий ОС и Postgres согласно инструкции по настройке репозиториев, предоставляемой СПАК.

Для настройки Системы необходимо создать учетную запись администратора системы (**user)** на сервере и добавить данного пользователя в группу администраторов (sudo).

Для поддержки Системы необходимо создать учетную запись группы технического обслуживания (kto**)**.

Все дальнейшие настройки выполняемые от администратора Системы будут описаны для УЗ с именем **user**.

Для корректного взаимодействия с системами СО, поддерживавшими шифрование SSL необходимо добавить корневые сертификаты доверенного центра сертификации в хранилище сертификатов ОС. Для этого необходимо загрузить все коревые сертификаты в формате crt с сайта https://www.so-ups.ru/about/udc/ и разместить их в директории /usr/local/share/ca-certificates/, после чего выполнить команду:

```
sudo update-ca-certificates
```

Необходимо синхронизировать время ВМ с сервером точного времени используя команду:

```
sudo ntpdate <IP Сервера точного времени>
```

Так же необходимо настроить автоматическую синхронизацию с сервером точного времени добавив указанную команду в crontab используя команды:

#добавляем задачу синхронизации времени раз в 3 часа

```
sudo su
crontab -l > foocron
echo "00 */3 * * * /usr/sbin/ntpdate <IP Сервера точного времени>" >> foocron
crontab foocron
rm foocron
exit
```

Для интеграции с ПАК ЕСМ необходимо установить пакет snmpd, используя команду:

```
sudo apt update && sudo apt install snmpd curl unzip
netcat atop
```

Для выполнения скритов длящих более 1 минуты необходимо установить приложение screen:

```
sudo apt update && sudo apt install screen
```

4.2. Установка СУБД

4.2.1. Установка СУБД

Для установки PostgreSQL необходимо подключиться по ssh на каждый узел кластера БД, выделенный для установки СУБД и выполнить следующие команды:

1. Добавить локаль en_US.UTF-8:

```
sudo  dpkg-reconfigure locales
В открывшемся меню необходимо выбрать локали: en_US.UTF-8
и ru_RU.UTF-8 после чего согласиться с изменениями
```

2. Установить СУБД PostgreSQL:

```
#Обновить список пакетов с репозитория
sudo apt update
#Установить пакет postgresql и rsync
sudo apt install -y postgrespro-std-14 rsync
```

3. Разрешить подключение к PostgreSQL с внешних узлов:

```
#Повысить привилегии пользователя
sudo su
#Разрешить авторизацию пользователей в PostgreSQL с любого
# ip адреса
echo "host all all 0.0.0.0/0 md5" >> /var/lib/pgpro/std-
14/data/pg_hba.conf
#Разрешить подключения со всех сетевых интерфейсов
echo "listen_addresses = '*'" >> /var/lib/pgpro/std-
14/data/postgresql.conf
```

4. Настроить PostgreSQL на локаль en_US.UTF-8, для этого в файле /var/lib/pgpro/std-14/data/postgresql.conf необходимо заменить параметр lc_messages на:

```
lc_messages 'en_US.UTF-8';
```

5. Запустить СУБД PostgreSQL:

```
#добавить в автозапуск сервис PostgreSQL
sudo systemctl enable postgrespro-std-14.service
```

```
#перезапустить сервис PostgreSQL
sudo systemctl restart postgrespro-std-14.service
#Произвести проверку сервиса PostgreSQL
systemctl status postgrespro-std-14.service
```

В строке, которая начинается с «Active:» должен быть указан статус «active (running)»

6. Присвоить УЗ postgres пароль командой:

```
sudo passwd postgres
```

на запрос системы необходимо дважды ввести пароль.

4.2.2. Настройка основного сервера СУБД

Для настройки основного сервера СУБД необходимо создать учетные записи и базы данных для сервисов Системы. Для этого необходимо:

1. Выполнить команды в соответствии с шаблоном (см. ниже).
   Таблица 4 содержит описание параметров, указанных в шаблоне.

Таблица 4 – Параметры конфигурации БД

Переменные	**Пример**	**Комментарии**
$PG_PSWD	PassWord	Пароль привилегированной учетной записи PostgreSQL
$ACL_DB	saper-acl	Имя БД для сервиса acl
$ACL_DB_USER	saper-acl	УЗ для доступа к БД сервиса acl
$ACL_DB_PASS	YR6m5B1	Пароль для УЗ $ACL_DB_USER
$CALCULATION_DB	saper-calculation	Имя основной БД для сервиса calculation-service
$CALCULATION_DB_USER	saper-calculation	УЗ для доступа к БД сервиса calculation-service
$CALCULATION_DB_PASS	YR6m5B2	Пароль для УЗ $CALCULATION_DB_USER
$MSK_CALC_DB	saper-msk-calculation	Имя БД для сервиса msk-calculation-service
$MSK_CALC_DB_USER	saper-msk-calculation	УЗ для доступа к БД сервиса msk-calculation-service
$MSK_CALC_DB_PASS	YR6m5B1	Пароль для УЗ $MSK_CALC_DB_USER
$DATA_DB	saper-data-manager	Имя БД для сервиса data-manager

Переменные	Пример	Комментарии
$DATA_DB_USER	saper-data-manager	УЗ для доступа к БД сервиса data-manager
$DATA_DB_PASS	YR6m5B1	Пароль для УЗ $DATA_DB_USER
$TASK_DB	saper-task-manager	Имя БД для сервиса task-manager
$TASK_DB_USER	saper-task-manager	УЗ для доступа к БД сервиса task-manager
$TASK_DB_PASS	YR6m5B1	Пароль для УЗ $TASK_DB_USER
$USER_DB	saper-user-manager	Имя БД для сервиса user-manager
$USER_DB_USER	saper-user-manager	УЗ для доступа к БД сервиса user-manager
$USER_DB_PASS	YR6m5B1	Пароль для УЗ $ USER_DB_USER
$REPLICA_LOGIN	replication	УЗ для репликации кластера СУБД
$REPLICA_PSWD	YR6m5B3	Пароль для УЗ $REPLICA_LOGIN
$OPT_DB	Saper-opt	Имя БД сервиса оптимизации расчетов
$OPT_DB_USER	Saper-opt	УЗ для доступа к БД сервиса оптимизации расчётов
$OPT_DB_PASS	YR6m5B3	Пароль для УЗ $OPT_DB_USER
$MAIN_SDB	192.168.0.0/24	IP адрес основного сервера СУБД

Шаблон:

```
#Переключиться в консоль привилегированного пользователя СУБД
sudo su postgres
#Войти в консоль СУБД
psql
#Изменить пароль входа в СУБД для пользователя postgres
ALTER USER postgres WITH PASSWORD '$PG_PSWD';
#Создать УЗ для БД сервиса user-manager
CREATE USER "$USER_DB_USER" WITH PASSWORD '$USER_DB_PASS' LOGIN;
#Создать БД для сервиса captcha-service
CREATE DATABASE "$USER_DB";
#Предоставить права к БД для УЗ сервиса captcha-service
GRANT ALL ON DATABASE "$USER_DB" TO "$USER_DB_USER" WITH GRANT OPTION;
#Создать УЗ для репликации кластера СУБД
```

```
CREATE ROLE "$REPLICA_LOGIN" WITH REPLICATION PASSWORD
"$REPLICA_PSWD" LOGIN;
#Выйти из консоли СУБД
\q
#Выйти из консоли пользователя postgres
Exit
```

Пример:

```
su postgres
psql
ALTER USER postgres WITH PASSWORD '*****';
CREATE USER "saper-acl" WITH PASSWORD '*****' LOGIN;
CREATE DATABASE "saper-acl";
GRANT ALL ON DATABASE "saper-acl" TO "saper-acl" WITH
GRANT OPTION;
CREATE USER "saper-calculation" WITH PASSWORD '*****'
LOGIN;
CREATE DATABASE "saper-calculation";
GRANT ALL ON DATABASE "saper-calculation" TO "saper-
calculation" WITH GRANT OPTION;
CREATE USER "saper-msk-calculation" WITH PASSWORD '*****'
LOGIN;
CREATE DATABASE "saper-msk-calculation";
GRANT ALL ON DATABASE "saper-msk-calculation" TO "saper-
msk-calculation" WITH GRANT OPTION;
CREATE USER "saper-data-manager" WITH PASSWORD '*****'
LOGIN;
CREATE DATABASE "saper-data-manager";
GRANT ALL ON DATABASE "saper-data-manager" TO "saper-data-
manager" WITH GRANT OPTION;
CREATE USER "saper-task-manager" WITH PASSWORD '*****'
LOGIN;
CREATE DATABASE "saper-task-manager";
GRANT ALL ON DATABASE "saper-task-manager" TO "saper-task-
manager" WITH GRANT OPTION;
CREATE USER "saper-user-manager" WITH PASSWORD '*****'
LOGIN;
CREATE DATABASE "saper-user-manager";
GRANT ALL ON DATABASE "saper-user-manager" TO "saper-user-
manager" WITH GRANT OPTION;
CREATE USER "saper-opt" WITH PASSWORD '*****' LOGIN;
CREATE DATABASE "saper-opt";
GRANT ALL ON DATABASE "saper-opt" TO "saper-opt" WITH
GRANT OPTION;
CREATE ROLE "replication" WITH REPLICATION PASSWORD
'*****' LOGIN;
```

```
\q
exit
```

2. Подключиться к основному узлу СУБД и выполнить последовательно команды для предварительной настройки репликации:

```
#Повысить привилегии пользователя
sudo su
#Создать директорию для хранения архива журнала транзакций
#(WAL)
mkdir /var/lib/pgpro/std-14/archive/
#Предоставить права на данную директорию УЗ postgres
chown postgres:postgres /var/lib/pgpro/std-14/archive/
chmod 700 /var/lib/pgpro/std-14/archive/
#Разрешить подключение к СУБД УЗ для репликации кластера
echo "host replication replication 0.0.0.0/0   md5" >>
/var/lib/pgpro/std-14/data/pg_hba.conf
```

Для очистки архива журнала транзакций СУБД необходимо добавить следующую задачу в cron (`sudo crontab -e`). Команда для очистки:

```
10 6 * * * /usr/bin/find /var/lib/pgpro/std-14/archive/ -
type f -mtime +2 -exec rm {} \;
```

Для настройки потоковой репликации необходимо добавить следующие строки в файл /var/lib/pgpro/std-14/data/postgresql.conf:

```
#Разрешить потоковую репликацию
wal_level = replica
#Задать число одновременных подключений с ведомых серверов
max_wal_senders = 3
#Задать количество сегментов WAL журнала, хранящихся в
СУБД
wal_keep_segments = 75
#Разрешить хранение архива WAL
archive_mode = on
#Команда для выполнения архивирования WAL в ранее
созданный каталог
archive_command = 'cp %p /var/lib/pgpro/std-14/archive/%f'
```

и перезагрузить сервис PostgreSQL командой:

```
sudo systemctl restart postgrespro-std-14.service
#Произвести проверку сервиса PostgreSQL
systemctl status postgrespro-std-14.service
```

В строке начинавшейся с «Active:» должен быть указан статус «active (running)»

4.2.3. Настройка резервного сервера СУБД

Для настройки резаервного сервера СУБД необходимо подключиться к резервному узлу СУБД и выполнить последовательно команды настройки резервного узла кластера PostgreSQL:

```
#Повысить привилегии пользователя
sudo su
#Остановить службу СУБД
systemctl stop postgresql
#Удалить каталог с СУБД
rm -rf /var/lib/pgpro/std-14/data/*
#Переключиться на привилегированного пользователя СУБД
su postgres
#Создать копию состояния основного узла PostgreSQL в
качестве отправной точки для репликации (поменять
$MAIN_SDB IP адрес основного сервера СУБД)
pg_basebackup -h $MAIN_SDB -U replication -D
/var/lib/pgpro/std-14/data/ -P -p 5432
#Настроить СУБД на прием подключений от клиентов
echo "hot_standby = on" >> /var/lib/pgpro/std-
14/data/postgresql.conf
```

Приложение запросит пароль от УЗ replication, созданной на предыдущем шаге настройки.

Для запуска резервного сервера необходимо синхронизировать архив wal логов, для этого, заменив `$DB_IP_SLAVE` на IP адрес резервного сервера СУБД, можно использовать команду:

```
rsync -rahzP /var/lib/pgpro/std-14/archive/ $DB_IP_SLAVE:
/var/lib/pgpro/std-14/archive/
```

После завершения синхронизации необходимо создать файл настройки репликации СУБД командой:

```
nano /var/lib/pgpro/std-14/data/recovery.conf
```

И заполнить его содержимым согласно шаблону (Переменные необходимо заменить значениями. Таблица 4 содержит описание переменных):

```
standby_mode = 'on'
primary_conninfo = 'host=$MAIN_SDB port=5432
user=$REPLICA_LOGIN password=$REPLICA_PSWD'
trigger_file = '/var/lib/pgpro/std-
14/data/promote_to_master'
restore_command = 'cp /var/lib/pgpro/std-14/archive/%f
"%p"'
```

Следующим шагом необходимо включить сервис PostgreSQL командой:

```
sudo systemctl start postgrespro-std-14.service
```

Для проверки работоспособности на основном сервере необходимо выполнить команду:

```
sudo -u postgres psql -c 'SELECT
client_addr,pg_xlog_location_diff(s.sent_location,s.replay
_location) byte_lag FROM pg_stat_replication;'
```

Ожидаемый результат – в выводе команды присутствует IP адрес резервного сервера.

4.2.1. Установка keepalived для кластера СУБД

ПО keepalived необходимо только в случае установки сервиса в отказоустойчивой конфигурации.

Для установки и настройки необходимо последовательно выполнить следующие команды на каждом сервере СУБД:

```
sudo su
apt-get update
apt-get install keepalived -y
echo net.ipv4.ip_forward=1 >> /etc/sysctl.conf
sysctl -p
touch /etc/keepalived/keepalived.conf
```

Для завершения конфигурации keepalived необходимо отредактировать конфигурационный файл командой `sudo nano /etc/keepalived/keepalived.conf`, добавив в него нижеприведенную конфигурацию и изменить значение `priority` в зависимости от роли сервера (основной/резервный).

Переменную <DB_IP> необходимо заменить на ip-адрес, выделенный для работы с кластером СУБД.

```
vrrp_instance saper_db {
state MASTER #BACKUP
interface eth0 #Указываем интерфейс, к которому будет
привязан VRRP instance
virtual_router_id 10 #Должен быть одинаков на всех хостах
в instance.
priority 110 #Для основного узла указываем 110 для
резервного 100.
advert_int 4
#Настройка аутентификации по паролю
authentication {
auth_type PASS
auth_pass 1111
}
#Настройка виртуального сетевого интерфейса
```

```
virtual_ipaddress {
        <DB_IP> dev eth0 label eth0:vip
}
}
```

После чего необходимо перезапустить сервис командой:

```
systemctl restart keepalived
```

Так же необходимо добавить сервис в автозагрузку командой:

```
systemctl enable keepalived
```

Установка и настройка keepalived закончена для проверки установки необходимо выполнить команду:

```
systemctl status keepalived
```

Статус сервиса должен соответствовать active (running).

Для основного сервера в выводе должно содержаться сообщение:

```
VRRP_Instance(solar_http) Entering MASTER STATE
```

Для резервного сервера в выводе должно содержаться сообщение:

```
VRRP_Instance(solar_http) Entering BACKUP STATE
```

4.2.2. Настройка резервного копирования СУБД

Для настройки резервного копирования кластера СУБД Postgres на сетевой диск доступный по протоколу SMB необходимо подключиться к консоли каждого узла кластера через ssh и выполнить следующие действия:

1. Произвести установку cifs-utils:

```
sudo apt update
sudo apt install -y cifs-utils
```

2. Создать файл /root/.smbclient с параметрами доступа к сетевому каталогу Windows:

```
sudo nano /root/.smbclient
```

Заполнить файл, указав логин, пароль, домен от УЗ имеющей доступ к хранилищу резервных копий:

```
username=<логин>
password=<пароль>
domain=<домен> например уровня ИА>
```

3. Создать каталог на сервере Linux, в который будет монтироваться сетевой каталог Windows:

```
sudo mkdir /srv/backup
```

4. Настроить автоматическое монтирование сетевого диска. Для этого необходимо отредактировать файл `/etc/fstab`, командой `sudo nano /etc/fstab` и добавить в данный файл строку:

```
//winserver/Share/ /srv/backup cifs
uid=postgres,gid=postgres,rw,credentials=/root/.smbclient,
file_mode=0600,dir_mode=0777 0 0
```

где:

- `//winserver/Share/` – путь к сетевому каталогу Windows, заменить на нужный путь, при этом меняем «\» на «/»);
- `/mnt/share` – точка (каталог) монтирования на сервере Linux, заменить на путь, созданный на шаге 4) текущего раздела;
- `/root/.smbclient` – полный путь файла с параметрами доступа к сетевому каталогу Windows, заменить на путь к файлу, созданному на шаге 3) текущего раздела.

Внимание! Если в пути каталога встречается «пробел» необходимо указывать его через запись «\040».

5. Запустить процесс монтирования каталогов в соответствии с настройками, указанными в файле /etc/fstab:

```
sudo mount -a
```

6. Создать директории для хранения резервных копий СУБД:

```
sudo mkdir /srv/backup/postgres-update
sudo mkdir /srv/backup/postgres
sudo mkdir /srv/backup/wal
```

7. Предоставить пользователям группы КТО:

- Доступ на чтение и запись к директории /srv/backup/postgres-update;
- Доступ на чтение к директории /srv/backup/postgres;
- Доступ на чтение к директории /srv/backup/wal;

8. Создать задачу переноса wal на сервер резервных копий:

```
00 * * * * rsync -zvr : /var/lib/pgpro/std-14/archive/
/srv/backup/wal/
```

9. Добавить настройки очитки WAL архива. Для этого, используя команду `sudo -u postgres crontab -e`, необходимо добавить строку:

```
10 6 * * * /usr/bin/find /srv/backup/wal/ -type f -mtime
+7 -exec rm {} \;
```

10. Настраиваем ежедневное создание полной копии СУБД. Для этого на резервном сервере СУБД, используя команду `sudo -u postgres crontab -e` добавляем в cron строку, заменив переменную `$REPLICA_PSWD` на пароль от УЗ replication, который создаётся в следующем разделе:

```
00 22 * * * PGPASSWORD="$REPLICA_PSWD"  pg_basebackup -h
localhost -U replication -F t -D
/srv/backup/postgres/$(date +\%Y\%m\%d) -x -z -p 5432
```

В результате каждый день в 21-30 будет создаваться, сжатая архиватором gzip, полная архивная копия СУБД.

11. Настраиваем очистку каталога с резервными копиями СУБД, для этого на резервном сервере СУБД, используя команду `sudo -u postgres crontab –e` добавляем в cron строку:

```
40 23 * * * /usr/bin/find /srv/backup/postgres/ -maxdepth
1  -type d -mtime +5 -exec rm -rf {} \;
```

В результате ежедневно будет производится очистка резервных копий СУБД. Будут удалены архивы старше 5 дней.

4.3. Установка кластера Kafka

Необходимо подключиться по ssh на каждый из серверов, выделенных для установки Kafka и выполнить следующие команды для установки сервиса, работающего из-под УЗ kafka:

1. Необходимо создать локального пользователя kafka и присвоить ему пароль.
2. Создать необходимые директории:

```
#Повысить привилегии пользователя
sudo su
#Обновить список пакетов
apt-get updated
#Установить Java:
apt update && apt install -y openjdk-8-jre
#Создать директорию для ПО kafka
mkdir /opt/kafka
#Предоставить права на созданную директорию УЗ user
chown kafka:kafka /opt/kafka
#создаем директорию для журналов сервиса
mkdir /var/log/kafka/
#меняем собственика дериктории
chown kafka:kafka /var/log/kafka/
#Настроить ротацию журналов сервиса (управление журналами
системы: копирование, очищение, архивация, удаление
устаревших архивов и перезапуск процесса, пишущего журнал)
cat <<EOF >> /etc/logrotate.d/kafka
/var/log/kafka/*.log {
        weekly
        rotate 10
        copytruncate
```

```
        delaycompress
        compress
        notifempty
}
EOF
#Перезапустить сервис ротации журналов
systemctl restart rsyslog
#Перенаправить журнал сервиса zookeeper в отдельный файл
cat <<EOF >> /etc/rsyslog.d/zookeeper.conf
if $programname == 'zookeeper' then
/var/log/kafka/zookeeper.log
& stop
EOF
#Перезагрузить сервис ротации журналов
systemctl restart rsyslog

#Создать юнит файл для запуска сервиса zookeeper
cat <<EOF >>/etc/systemd/system/zookeeper.service
[Unit]
Requires=network.target remote-fs.target
After=network.target remote-fs.target

[Service]
Type=simple
User=kafka
ExecStart=/opt/kafka/bin/zookeeper-server-start.sh
/opt/kafka/config/zookeeper.properties
ExecStop=/opt/kafka/bin/zookeeper-server-stop.sh
Restart=on-failure
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=zookeeper

[Install]
WantedBy=multi-user.target
EOF

#Создать юнит файл для запуска сервиса kafka
sudo cat <<EOF >> /etc/systemd/system/kafka.service
[Unit]
Requires=zookeeper.service
After=zookeeper.service

[Service]
Type=simple
```

```
User=kafka
Environment="KAFKA_JMX_OPTS=-
Dcom.sun.management.jmxremote=true -
Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.ssl=false -
Djava.net.preferIPv4Stack=true -
Dcom.sun.management.jmxremote.local.only=false"
ExecStart=/bin/sh -c '/opt/kafka/bin/kafka-server-start.sh
/opt/kafka/config/server.properties >>
/var/log/kafka/kafka.log 2>&1'
ExecStop=/opt/kafka/bin/kafka-server-stop.sh
Restart=on-failure

[Install]
WantedBy=multi-user.target
EOF

#Переключиться в УЗ kafka
su kafka
#Скачать сервис с официального репозитория
wget -O ~/kafka.tgz "<адрес в формате https://>"
#Перейти в рабочую директорию сервиса
cd /opt/kafka
#Разархивировать сервис в его рабочую директорию:
tar -xvzf ~/kafka.tgz --strip 1
#Создать директорию для хранения данных zookeeper
mkdir -p /opt/kafka/zookeeper/data
#Создать директорию для хранения данных kafka
mkdir -p /opt/kafka/kafka-logs
```

В конфигурацию сервиса необходимо добавить параметры в соответствии с шаблоном ниже. Для изменения конфигурации используйте команду:

```
nano /opt/kafka/config/zookeeper.properties
```

В шаблоне необходимо заменить переменные $KAFKA_1, $KAFKA_2, $KAFKA_3 на DNS адреса серверов, предназначенных для развертывания кластера Kafka.

Шаблон:

```
dataDir=/opt/kafka/zookeeper/data
clientPort=2181
maxClientCnxns=100
admin.enableServer=false
quorumListenOnAllIPs=true
server.1=$KAFKA_1:2888:3888
```

```
server.2=$KAFKA_2:2888:3888
server.3=$KAFKA_3:2888:3888
initLimit=5
syncLimit=2
```

Пример:

```
dataDir=/opt/kafka/zookeeper/data
clientPort=2181
maxClientCnxns=100
admin.enableServer=false
quorumListenOnAllIPs=true
server.1=kafka1.mydep.myorg:2888:3888
server.2=kafka2.mydep.myorg:2888:3888
server.3=kafka3.mydep.myorg:2888:3888
initLimit=5
syncLimit=2
```

Также необходимо создать каталог:

```
mkdir -p /opt/kafka/zookeeper/data/
```

И файл /opt/kafka/zookeeper/data/myid с уникальным значением для каждого сервера в кластере (для server.1 - 1, server.2 - 2 server.3 - 3), для этого можно использовать команду:

```
echo 1 > /opt/kafka/zookeeper/data/myid
echo 2 > /opt/kafka/zookeeper/data/myid
echo 3 > /opt/kafka/zookeeper/data/myid
```

Внести изменения в файл конфигурации `server.properties`, используя команду:

```
nano /opt/kafka/config/server.properties
```

необходимо изменить в соответствии с примером. Параметр `broker.id` должен соответствовать значению в файле `/opt/kafka/zookeeper/data/myid`, параметр `zookeeper.connect` должен содержать перечисление всех серверов кластера, а параметр `advertised.listeners` должен содержать DNS имя сервера.

Пример:

```
broker.id=1 #2 #3 #должен быть уникальным для каждого
сервера в кластере
listeners=PLAINTEXT://:9092
advertised.listeners=PLAINTEXT://kafka3.mydep.myorg:9092
num.network.threads=3
num.io.threads=8
socket.send.buffer.bytes=102400
socket.receive.buffer.bytes=102400
socket.request.max.bytes=104857600
log.dirs=/opt/kafka/kafka-logs
```

```
num.partitions=1
num.recovery.threads.per.data.dir=1
offsets.topic.replication.factor=2
transaction.state.log.replication.factor=2
transaction.state.log.min.isr=2
log.flush.interval.ms=1000
log.retention.hours=48
log.segment.bytes=367001600
log.retention.check.interval.ms=300000
zookeeper.connect= kafka1.mydep.myorg:2181,
kafka2.mydep.myorg:2181, kafka3.mydep.myorg:2181
zookeeper.connection.timeout.ms=18000
group.initial.rebalance.delay.ms=0
delete.topic.enable=true
```

После чего необходимо запустить сервисы на всех серверах кластера следующими командами:

```
sudo systemctl enable zookeeper
sudo systemctl start zookeeper
sudo systemctl enable kafka
sudo systemctl start kafka
```

Для проверки работоспособности на каждом из серверов кластера необходим выполнить команду:

```
/opt/kafka/bin/zookeeper-shell.sh localhost:2181 ls
/brokers/ids
```

Ожидаемый вывод:

```
WatchedEvent state:SyncConnected type:None path:null
[1, 2, 3]
```

4.4. Настройка сервера балансировки нагрузки

4.4.1. Установка Nginx

Для работы front необходимо установить Nginx. Для этого необходимо подключиться к каждому серверу балансировки нагрузки по SSH и выполнить следующую последовательность действий:

1. Установить Nginx при помощи команды:

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

2. Удалить автоматически созданный файл конфигурации nginx:

   ```
   sudo rm /etc/nginx/sites-available/default
   ```

3. Запустить файловый менеджер командой `sudo mc` и перенести SSL сертификат в директорию /var/www/.
4. Создать директорию сервиса **frontend:**

```
sudo mkdir /var/www/html/saper
```

5. Предоставить права УЗ kto на директорию с веб-приложением frontend-web, используя команду:

```
sudo chown -R kto:kto /var/www
```

6. Заполнить настройки взаимодействия с сервисами Системы, используя команду:

```
sudo nano /etc/nginx/conf.d/saper.conf
```

Необходимо заполнить файл конфигурации сервиса согласно нижеприведенному шаблону. Таблица 5 содержит описание переменных, используемых в шаблоне.

Таблица 5 - Список переменных в конфигурационных файлах Nginx

$SITE-NAME	DNS имя сайта Системы
$KEY_PATH	Путь до файла, содержавшего закрытый ключ
$CRT_PATH	Путь до файла, содержавшего открытый ключ
$BACKEND-IP1	IP адрес основного сервера приложений
$BACKEND-IP2	IP адрес резервного сервера приложений
$CALC-IP1	IP адрес основного сервера расчетов
$CALC-IP2	IP адрес резервного сервера расчетов
$OPT-IP1	IP адрес основного сервера оптимизации расчетов
$ OPT-IP2	IP адрес резервного сервера оптимизации расчетов

Для удобства изменения переменные представлены в формате:

```
${Имя:-<Значение по умолчанию>}
```

После редактирования переменные должны принять вид:

```
<Значение>
```

Шаблон:

```
upstream data-manager {
    server ${BACKEND-IP1:-XXX.XXX.XXX.XXX}:10002;
    server ${BACKEND-IP2:-XX.XXX.XXX.XXX}:10002;
    }
upstream user-manager {
    server ${BACKEND-IP1:-XXX.XXX.XXX.XXX}:10001;
    server ${BACKEND-IP2:-XXX.XXX.XXX.XXX}:10001;
    }
upstream calculation {
    server ${CALC-IP1:-XXX.XXX.XXX.XXX}:10003;
    server ${CALC-IP2:-XXX.XXX.XXX.XXX}:10003;
```

```
    }
upstream task-manager {
    server ${BACKEND-IP1:-XXX.XXX.XXX.XXX}:10004;
    server ${BACKEND-IP2:-XXX.XXX.XXX.XXX}:10004;
    }
upstream acl {
    server ${BACKEND-IP1:-XXX.XXX.XXX.XXX}:10006;
    server ${BACKEND-IP2:-XXX.XXX.XXX.XXX}:10006;
    }
upstream msk-calculation {
    server ${CALC-IP1:-XXX.XXX.XXX.XXX}:10007;
    server ${CALC-IP2:-XXX.XXX.XXX.XXX}:10007;
}
upstream saper-msk-opt {
    server ${OPT-IP1:-XXX.XXX.XXX.XXX}:5000 max_conns=4;
    server ${OPT-IP2:-XXX.XXX.XXX.XXX}:5000 max_conns=4;
}

server {
    if (-f /etc/nginx/maintenance) {
        return 503;
    }
    error_page 503 @maintenance;
    location @maintenance {
        rewrite ^(.*)$ /server-error.html break;
    }
    listen       443 ssl;
    ssl_certificate ${CRT_PATH:-/var/www/html/web.crt};
    ssl_certificate_key  ${KEY_PATH:-/var/www/html/web.key
};
    server_name  ${SITE-NAME:- saper.mydep.myorg};
    root   /var/www/html/saper/;
    gzip on;
    gzip_types text/css application/javascript
application/json image/svg+xml;
    gzip_comp_level 9;
    etag on;

    client_max_body_size 600m;
    proxy_connect_timeout       60;
    proxy_send_timeout          60;
    proxy_read_timeout          2400;
    send_timeout                2400;

    #charset koi8-r;
```

```
        access_log /var/log/nginx/nginx_access.log  main;
        error_log   /var/log/nginx/nginx_error.log error;
        location / {
            try_files $uri $uri/ /index.html =404;
    }

        location /index.html {
            add_header Cache-Control 'no-store, no-cache,
    must-revalidate, proxy-revalidate, max-age=0';
            if_modified_since off;
            expires off;
            etag off;
        }
        location /config.json {
            add_header Cache-Control 'no-store, no-cache,
    must-revalidate, proxy-revalidate, max-age=0';
            if_modified_since off;
            expires off;
            etag off;
        }
            location /user-manager/api/ {
            proxy_connect_timeout 1s;
            proxy_pass http://user-manager;
        }
            location /data-manager/api/ {
            proxy_connect_timeout 1s;
            #client_max_body_size 0;
            proxy_http_version 1.1;
            proxy_request_buffering off;
            proxy_pass http://data-manager;
        }
            location /calculation/api/ {
            proxy_connect_timeout 1s;
             proxy_pass http://calculation;
        }
            location /task-manager/api/ {
            proxy_connect_timeout 1s;
            proxy_pass http://task-manager;
        }
            #client_max_body_size 100m;
            location /acl/api/ {
            proxy_connect_timeout 1s;
            proxy_pass http://acl;
        }
            location /msk-calculation/api/ {
```

```
        proxy_connect_timeout 1s;
        proxy_pass http://msk-calculation;
    }
   location /saper-msk-opt/ {
        proxy_pass http://saper-msk-opt/;
        client_max_body_size 600m;
#        proxy_pass https://saper-msk-opt/saper-msk-opt/;
    }
}

server {
    listen       80;
    server_name  ${SITE-NAME:-saper.mydep.myorg};
    gzip on;
    gzip_types text/css application/javascript
application/json image/svg+xml;
    gzip_comp_level 9;
    etag on;

    client_max_body_size 600m;

    proxy_connect_timeout      60;
    proxy_send_timeout         60;
    proxy_read_timeout         2400;
    send_timeout               2400;

    #charset koi8-r;
    access_log  /var/log/nginx/nginx_access_80.log  main;
    error_log   /var/log/nginx/nginx_error_80.log error;

        location /user-manager/api/ {
        proxy_connect_timeout 1s;
        proxy_pass http://user-manager;
    }
        location /data-manager/api/ {
        proxy_connect_timeout 1s;
        client_max_body_size 0;
        proxy_http_version 1.1;
        proxy_request_buffering off;
        proxy_pass http://data-manager;
    }
        location /calculation/api/ {
         proxy_connect_timeout 1s;
         proxy_pass http://calculation;
    }
```

```
        location /task-manager/api/ {
        proxy_connect_timeout 1s;
        proxy_pass http://task-manager;
    }
        #client_max_body_size 100m;
        location /acl/api/ {
        proxy_connect_timeout 1s;
        proxy_pass http://acl;
    }
        location /msk-calculation/api/ {
        proxy_connect_timeout 1s;
        proxy_pass http://msk-calculation;
    }
        location /saper-msk-opt/ {
        proxy_pass http://saper-msk-opt/;
        client_max_body_size 600m;

    }
}
```

```
Для применения настроек необходимо перезагрузить Nginx
командой:
sudo systemctl restart nginx
sudo systemctl enable nginx
```

Установка и настройка сервиса **front** закончена. Для проверки работоспособности Nginx необходимо выполнить команду:

```
systemctl status nginx |grep active
```

Ожидаемый ответ:

```
Active: active (running)
```

4.4.2. Настройка NFS

Для настройки NFS необходимо подключиться по SSH к основному серверу приложений, настроить сервис.

Для создания сетевой директории необходимо выполнить следующие команды:

1. Установить NFS сервер:

```
sudo apt update
sudo apt install nfs-kernel-server rsync -y
```

2. Создать локальную директорию для сетевого диска.

```
sudo mkdir /server/upload-storage/
```

3. Добавить в файл /etc/exports следующую строку, заменив переменные на ip адреса серверов:

```
/server/upload-storage ${BACKEND-IP1:-
XXX.XXX.XXX.XXX}(rw,sync,no_subtree_check,no_root_squash)
${BACKEND-IP2:-
XXX.XXX.XXX.XXX}(rw,sync,no_subtree_check,no_root_squash)
${WEB-IP1:-
XXX.XXX.XXX.XXX}(rw,sync,no_subtree_check,no_root_squash)
${WEB-IP2:-
XXX.XXX.XXX.XXX}(rw,sync,no_subtree_check,no_root_squash)
```

4. Обновляем таблицу экспорта NFS:

```
sudo exportfs -a
```

5. Добавляем NFS сервер в автозагрузку:

```
sudo systemctl enable nfs-server
```

6. Перезагружаем NFS сервер:

```
sudo systemctl restart nfs-server
```

7. Установить NFS клиент:

```
sudo apt update
sudo apt install nfs-common -y
```

Следующие действия выполняются для кластерной конфигурации:

8. Создать локальную директорию для сетевого диска

```
sudo mkdir /server/upload-storage1/
```

9. Добавить в /etc/fstab строку стоку
   `${MAIN_SRV}:/server/upload-storage  /server/upload-storage nfs auto,nofail,noatime,nolock,intr,tcp,actimeo=1800 0 0`
   При этом заменив `${MAIN_SRV}`:
   – для основного сервера балансировки IP адресом резервного
   – для резервного сервера балансировки IP адресом основного
10. Выполнить монтирование командой:

```
sudo mount –a
```

11. Настроить синхронизацию между основным и резервным сервером балансировки добавив в cron строку:

```
05 * * * * rsync -a /server/upload-storage/
/server/upload-storage1/
```

4.4.3. Настройка keepalived для кластера балансировки нагрузки

ПО keepalived необходимо только в случае установки сервиса в кластерной конфигурации.

Для установки и настройки необходимо последовательно выполнить следующие команды на каждом сервере балансировки нагрузки:

```
sudo su
apt-get update
apt-get install keepalived -y
echo net.ipv4.ip_forward=1 >> /etc/sysctl.conf
sysctl -p
touch /etc/keepalived/keepalived.conf
```

Для завершения конфигурации keepalived необходимо отредактировать конфигурационный файл командой

```
sudo nano /etc/keepalived/keepalived.conf,
```

добавив в него нижеприведенную конфигурацию и изменить значение priority в зависимости от роли сервера (основной/резервный).

Переменную <Nginx_IP> необходимо заменить на ip адрес, выделенный для работы с кластером балансировки нагрузки.

```
lobal_defs {
script_user root
enable_script_security
}
vrrp_script chk_nginx {
    script "ps -C nginx"
    interval 2
}
vrrp_instance saper_nginx {
state MASTER #BACKUP
interface eth0 #Указываем интерфейс, к которому будет
привязан VRRP instance
virtual_router_id 10 #Должен быть одинаков на всех хостах
в instance.
priority 110 #Для основного узла указываем 110 для
резервного 100.
advert_int 4
#Настройка аутентификации по паролю
authentication {
auth_type PASS
auth_pass 1111
}
#Настройка виртуального сетевого интерфейса
virtual_ipaddress {
        <Nginx_IP> dev eth0 label eth0:vip
}
}
```

После чего необходимо перезапустить сервис командой:

```
systemctl restart keepalived
```

Так же необходимо добавить сервис в автозагрузку командой:

```
systemctl enable keepalived
```

Установка и настройка keepalived закончена для проверки установки необходимо выполнить команду:

```
systemctl status keepalived
```

Статус сервиса должен соответствовать active (running).

Для основного сервера в выводе должно содержаться сообщение:

```
VRRP_Instance(solar_http) Entering MASTER STATE
```

Для резервного сервера в выводе должно содержаться сообщение:

```
VRRP_Instance(solar_http) Entering BACKUP STATE
```

4.5. Установка Docker-engine

Для настройки Docker-engine необходимо подключиться к каждому серверу приложений и серверу расчетов по SSH и выполнить последовательно следующие команды.

```
#Переходим в консоль root для повышения привилегий
sudo su
#Обновляем список доступных пакетов и устанавливаем
необходимые
apt-get update
apt-get install -y git curl unzip
#Загружаем установочный пакет из репозитория
curl -L <адрес до дистрибутива в формате https://>  -o
~/docker.zip
cd ~/
unzip ~/docker.zip
cd ~/docker
#Устанавливаем Docker engine
dpkg -i ./*deb
#Устанавливаем docker compouse
cp ./docker-compose /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
exit
```

Далее добавляем в конфигурацию докера настройки сети, дабы исключить использование подсетей, занятых во внутренних сетях СО. Создаем файл конфигурации используя команду:

```
sudo nano /etc/docker/daemon.json
```

и добавляем туда, следующую конфигурацию:

```
{
        "live-restore": true,
        "bip": "192.168.10.1/24",
```

```
        "default-address-pools": [{
                "base": "192.168.0.0/16",
                "size": 24
        }]
}
```

Производим запуск Docker-engine:

```
#Запускаем демон Docker-engine
systemctl start docker
#Включаем демон Docker-engine в автозагрузку
systemctl enable docker
```

Включаем пользователей в группу docker для запуска контейнеров

```
usermod -aG docker user
usermod -aG docker kto
```

Установка docker-engine закончена. Для проверки установки необходимо выполнить команду:

```
systemctl status docker |grep active
```

Ожидаемый ответ:

```
Active: active (running)
```

4.6. Настройка NFS на серверах приложений

Необходимо настроить NFS клиент. Для этого необходимо подключиться по SSH к каждому серверу приложений и настроить подключения к сетевой папке.

1. Установить NFS клиент:

```
sudo apt update
sudo apt install nfs-common -y
```

2. Создать локальную директорию для сетевого диска

```
sudo mkdir /server/upload-storage/
```

3. Добавить в файл /etc/fstab строку. ${MAIN_SRV} – необходимо заменить на IP адрес выделенный для кластера балансировки нагрузки:

```
${MAIN_SRV}:/server/upload-storage /server/upload-storage
nfs auto,nofail,noatime,nolock,intr,tcp,actimeo=1800 0 0
```

4. Выполнить монтирование командой:

```
sudo mount -a
```

5. Настройка сервера оптимизации расчетов

Необходимо подготовить сервер согласно инструкции: «Инструкция по сборке, установке и настройке сервиса оптимизации».

Для предоставления доступа администраторам Системы необходимо добавить группу `ia-ntcic-admin` в группу локальных администраторов.

6. Передача данных группе КТО

После выполнения установки группе КТО необходимо передать:

1. IP адреса и имена ВМ Системы;
2. Данные, полученные при выполнении работ, указанных в главе 3.3;
3. Пароль от УЗ kto;
4. Пароли и УЗ для подключения к БД;
5. Пароль от УЗ kafka;

7. Настройка компонентов системы

7.1. Подготовка кластера Kafka

После установки необходимо настроить очереди сообщений, для этого необходимо подключиться к одному из серверов кластера kafka по протоколу ssh от УЗ kto и выполнить следующие команды:

1. Переключиться на пользователя kafka:

```
su kafka
```

2. Создать топики:

```
/opt/kafka/bin/kafka-topics.sh --create --bootstrap-server
localhost:2181 –replication-factor 2 --partitions 20 --
topic task-manager.publishing
/opt/kafka/bin/kafka-topics.sh --create --bootstrap-server
localhost:2181 –replication-factor 2 --partitions 20 --
topic task-manager.completion
/opt/kafka/bin/kafka-topics.sh --create --bootstrap-server
localhost:2181 –replication-factor 2 --partitions 20 --
topic task-manager.result
/opt/kafka/bin/kafka-topics.sh --create --bootstrap-server
localhost:2181 –replication-factor 2 --partitions 20 --
topic calculation.request
/opt/kafka/bin/kafka-topics.sh --create --bootstrap-server
localhost:2181 –replication-factor 2 --partitions 20 --
topic acl.policies.changed
```

7.2. Предварительная настройка серверов приложений и расчетов

7.2.1. Настройка Docker-engine

Для загрузки контейнеров с сервисами необходимо авторизоваться в хранилище артефактов. Для этого необходимо подключится к каждому серверу приложений и каждому серверу расчетов и воспользоваться командой:

```
docker login myrepository.mydep.myorg:18181
```

На запрос авторизации необходимо ввести данные УЗ, который предоставлен доступ к проекту в ФПА.

Ожидаемый ответ:

```
Login Succeeded
```

7.2.2. Загрузка конфигурационных файлов сервисов

Конфигурационные файлы расположены в репозитории <адрес в формате https://>. Конфигурация стенда храниться в отдельной ветке репозитория. Конфигурация продуктивного стенда храниться в ветке master, полигона в ветке test.

Для редактирования конфигурации необходимо скорректировать значения параметров в файлах. env и config-service/.env. Таблица 6 описывает переменные, которые необходимо скорректировать.

Таблица 6 Список параметров, используемых в env-example

Переменная	Пример	Описание
DB_IP	XXX.XXX.XXX.XXX:5432	IP адрес и порт подключения к кластеру СУБД.
ACL_DB	saper-acl	Имя БД для сервиса acl
ACL_DB_USER	saper-acl	УЗ для доступа к БД сервиса acl
CALCULATION_DB	saper-calculation	Имя основной БД для сервиса calculation-service
CALCULATION_DB_USER	saper-calculation	УЗ для доступа к БД сервиса calculation-service

Переменная	Пример	Описание
MSK_CALC_DB	saper-msk-calculation	Имя БД для сервиса msk-calculation-service
MSK_CALC_DB_USER	saper-msk-calculation	УЗ для доступа к БД сервиса msk-calculation-service
DATA_DB	saper-data-manager	Имя БД для сервиса data-manager
DATA_DB_USER	saper-data-manager	УЗ для доступа к БД сервиса data-manager
TASK_DB	saper-task-manager	Имя БД для сервиса task-manager
TASK_DB_USER	saper-task-manager	УЗ для доступа к БД сервиса task-manager
USER_DB	saper-user-manager	Имя БД для сервиса user-manager
USER_DB_USER	saper-user-manager	УЗ для доступа к БД сервиса user-manager
SERVICE_PROXY	https://sapermydep.myorg:443	Адрес кластера серверов балансировки.
KAFKA_SERVER	s-kafka1:9092,s-kafka2:9092, s-kafka3:9092,	Перечисление DNS имен серверов, входящих в кластер брокера сообщений.
SOAP_CA_CERTS	'/u3+7QAAAA'	Открытый ключ корневого сертификата.
SPRING_BOOT_ADMIN_URL	'http://10.0.0.8:1111,http:// 10.0.0.9:1111'	Перечисление серверов на которых расположен сервис spring-boot-admin
SPRING_BOOT_ADMIN_USER_NAME	saper-admin	Имя пользователя для авторизации на сервисе spring-boot-admin

Переменная	Пример	Описание
LDAP_URL	ldap://myldap.mydep.myorg:3268	Адрес подключения к глобальному каталогу AD Допускается указание нескольких контроллеров через пробел.
SAPER_TECH_USER	<домен уровня ИА>\<сервисная УЗ>	УЗ технического пользователя Системы
CONFIGURATION_URI	http://10.0.0.8:10005/configuration/api/v1/properties	Адрес сервера, на котором расположен сервис configuration

После изменения значений параметров необходимо распространить конфигурацию на сервера системы.

Для загрузки конфигурационных файлов сервисов необходимо подключиться к каждому серверу приложений и серверу расчетов по SSH и выполнять следующую последовательность действий:

1. Загрузить репозиторий с шаблоном конфигурации запуска, используя команду:

```
git clone https://myrepository.mydep.myorg/myconfig
```

На запрос авторизации необходимо ввести данные УЗ, имеющей доступ к репозиторию проекта в ФПА.

2. Перейти в директорию с конфигурацией и переключиться на ветку, соответствующую стенду (Конфигурация продуктивного стенда храниться в ветке master, полигона в ветке test) используя команды:

```
cd ~/config/
#$branch необходимо заменить на имя ветки
# master -конфигурация продуктива/ test конфигурация полигона
git checkout $branch
```

3. Загрузить изменения конфигурации:

```
git pull
```

На запрос авторизации необходимо ввести данные УЗ, имеющей доступ к репозиторию проекта в ФПА.

7.2.3. Настройка и запуск сервиса configuration

Для настройки и запуска сервиса **configuration** необходимо выполнить действия, указанные в разделе 7.2. Подключиться по SSH к серверу приложений и выполнить следующую последовательность действий:

1. Перейти в директорию с шаблоном запуска cd ~/config/ и используя в качестве шаблона файл: ~/config/config-service/.pasw-example создать новый файл: ~/config/config-service/.pasw.
2. Заполнить параметры в файле ~/config/config-service/.pasw.

Таблица 7 описывает переменные, которые необходимо заменить в файле .pasw.

Таблица 7 - Список параметров, используемых в .pasw

Переменная	Пример	Описание
ACL_DB_PASS	R6m5B1	Пароль для УЗ $ACL_DB_USER
CALCULATION_DB_PASS	R6m5B2	Пароль для УЗ $CALCULATION_DB_USER
MSK_CALC_DB_PASS	R6m5B3	Пароль для УЗ $MSK_CALC_DB_USER
DATA_DB_PASS	R6m5B4	Пароль для УЗ $DATA_DB_USER
SPRING_BOOT_ADMIN_USER_PASSWORD	Fehtyd4hdyK	Пароль для пользователя указанного в **SPRING_BOOT_ADMIN_USER_NAME**
SAPER_TECH_PASSWD	R6m5B5	Пароль для технической учетной записи **LDAP_MANAGER_LOGIN**
$TASK_DB_PASS	YR6m5B1	Пароль для УЗ $TASK_DB_USER
$USER_DB_PASS	YR6m5B1	Пароль для УЗ $ USER_DB_USER

3. Запустить сервис. Для запуска сервиса необходимо использовать SH скрипт, выполнив команду:

```
./configuration.sh
```

(SH скрипт расположен в директории с шаблоном конфигурации).

Ожидаемый результат выполнения команды – запущен docker-контейнер configuration**.**

7.2.4. Запуск сервиса spring-boot-admin

Для запуска сервиса необходимо выполнить действия, указанные в разделе 7.2–7.2.3. Подключиться по SSH к каждому серверу приложений и выполнить следующую последовательность действий:

Перейти в директорию с шаблоном запуска cd ~/config/ и запустить скрипт запуска сервиса:

```
./spring-boot-admin.sh
```

(SH скрипт расположен в директории с шаблоном конфигурации)

Ожидаемый результат выполнения команды – запущен docker-контейнер spring-boot-admin**.**

7.2.5. Запуск сервиса acl

Для запуска сервиса необходимо выполнить действия, указанные в разделе 7.2-7.2.4. Подключиться по SSH к каждому серверу приложений и выполнить следующую последовательность действий:

Перейти в директорию с шаблоном запуска cd ~/config/ и запустить скрипт запуска сервиса:

```
./acl.sh
```

(SH скрипт расположен в директории с шаблоном конфигурации).

Ожидаемый результат выполнения команды – запущен docker-контейнер acl**.**

7.2.6. Настройка и запуск сервиса data-manager

Для запуска сервиса необходимо выполнить действия, указанные в разделе 7.2-7.2.4. Подключиться по SSH к каждому серверу приложений.

Для запуска сервиса необходимо выполнить следующие команды:

4. Перейти в директорию с шаблоном запуска cd ~/config/ и запустить скрипт запуска сервиса:

```
./data-manager.sh
```

(SH скрипт расположен в директории с шаблоном конфигурации).

Ожидаемый результат выполнения команды – запущен docker-контейнер data-manager.

7.2.7. Запуск сервиса task-manager

Для запуска сервиса необходимо выполнить действия, указанные в разделе 7.2-7.2.4. Подключиться по SSH к каждому серверу приложений и выполнить следующую последовательность действий:

Перейти в директорию с шаблоном запуска cd ~/config/ и запустить скрипт запуска сервиса:

```
./task-manager.sh
```

(SH скрипт расположен в директории с шаблоном конфигурации).

Ожидаемый результат выполнения команды – запущен docker-контейнер task-manager.

7.2.8. Запуск сервиса user-manager

Для запуска сервиса необходимо выполнить действия, указанные в разделе 7.2-7.2.4. Подключиться по SSH к каждому серверу приложений и выполнить следующую последовательность действий:

Перейти в директорию с шаблоном запуска cd ~/config/ и запустить скрипт запуска сервиса:

```
./user-manager.sh
```

(SH скрипт расположен в директории с шаблоном конфигурации).

Ожидаемый результат выполнения команды – запущен docker-контейнер user-manager.

7.2.9. Запуск сервиса msk-calculation-service

Для запуска сервиса необходимо выполнить действия указанные в разделе 7.2-7.2.4. Подключиться по SSH к каждому серверу расчетов и выполнить следующую последовательность действий:

Перейти в директорию с шаблоном запуска cd ~/config/ и запустить скрипт запуска сервиса:

```
./msk-calculation-service.sh
```

(SH скрипт расположен в директории с шаблоном конфигурации).

Ожидаемый результат выполнения команды – запущен docker-контейнер msk-calculation-service.

7.2.10. Запуск сервиса calculation-service

Для запуска сервиса необходимо выполнить действия, указанные в разделе 7.2-7.2.4. Подключиться по SSH к каждому серверу расчетов и выполнить следующую последовательность действий:

Перейти в директорию с шаблоном запуска cd ~/config/ и запустить скрипт запуска сервиса:

```
./calculation-service.sh
```

(SH скрипт расположен в директории с шаблоном конфигурации)

Ожидаемый результат выполнения команды – запущен docker-контейнер calculation-service**.**

7.3. Установка и запуск сервиса оптимизации расчетов

Для запуска сервиса оптимизации расчётов необходимо подключиться к каждому серверу оптимизации расчётов при помощи RDP и выполнить действия, описанные в документе: Инструкция по сборке, установке и настройке сервиса оптимизации.

7.4. Настройка Nginx

Для настройки сервиса необходимо загрузить артефакт сервиса **front** с ФПА. Артефакт доступен по ссылке: https://myrepository.mydep.myorg/myartifact

После чего необходимо загрузить на каждый сервер балансировки нагрузки SSL сертификат и артефакт в домашнюю папку (~/), расположенную на сервере Системы (рекомендуется использовать ПО WinSCP).

1. Разархивировать артефакт сервиса командой:

```
tar -xvf front-poligon.tar.gz
```

(необходимо заменить «./artifacts.zip» на путь к артефакту сервиса)

2. Очистить директорию веб сайта командой:

```
rm -r /var/www/html/saper/*
```

3. Переместить файлы сервиса в директорию веб сайта командой:

```
cp -r ./front/build/* /var/www/html/saper/
```

(необходимо заменить «./» на путь к разархивированному артефакту)

4. Удалить временные файлы сервиса:

```
rm -rf ./front/build/
```

(необходимо заменить «**./**» на путь к разархивированному артефакту)

Для проверки работоспособности сервиса необходимо перейти по веб-ссылке, соответствующей имени сайта **Системы**. Ожидаемый результат – отображение стартовой страницы сервиса **front**.

В случае если стартовая страница приложения не загружается, рекомендуется обратиться к лог файлам Nginx и устранить зафиксированную в них проблему. Для просмотра лог файлов Nginx (если не менялись пути в конфигурационном файле выше) необходимо воспользоваться следующими командами:

```
#Error лог
sudo cat /var/log/nginx/error.log
#Access лог
sudo cat /var/log/nginx/access.log
```

8. Лист регистрации изменений

№п/п	Автор	Редакция	Дата	Описание изменения
1	АО «НТЦ ЕЭС Информационные комплексы»	1.0	24.06.2021	Первая версия
2	АО «НТЦ ЕЭС Информационные комплексы»	1.0	18.10.2021	Вторая версия
3	АО «НТЦ ЕЭС Информационные комплексы»	1.3.2	08.02.2023	Внесение исправлений и актуализация
4	АО «НТЦ ЕЭС Информационные комплексы»	1.3.3	20.03.2023	Изменение порядка разделов, разделение «установки» и «настройки»
5	АО «НТЦ ЕЭС Информационные комплексы»	1.3.6	30.03.2023	Внесение исправлений и актуализация
6	АО «НТЦ ЕЭС Информационные комплексы»	1.3.13	04.07.2023	Внесение исправлений и актуализация команд настройки Kafka
7	АО «НТЦ ЕЭС Информационные комплексы»	1.4.1	26.07.2023	Изменение версии ПО