



**ИНФОРМАЦИОННО-УПРАВЛЯЮЩАЯ СИСТЕМА
«ЭКСПОРТ/ИМПОРТ ЭЛЕКТРОЭНЕРГИИ В ЗАРУБЕЖНЫЕ
ЭНЕРГОСИСТЕМЫ - 24»**

**ИНСТРУКЦИЯ ПО УСТАНОВКЕ
И НАСТРОЙКЕ**

ВЕРСИЯ 1.2.6

РЕДАКЦИЯ 1.2.6 ОТ 06.02.2025

Москва
2025



Содержание

1. ОСНОВНЫЕ ПОНЯТИЯ, ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ	3
2. НАЗНАЧЕНИЕ РУКОВОДСТВА	5
3. ТРЕБОВАНИЯ К ПРОГРАММНЫМ/АППАРАТНЫМ РЕСУРСАМ	5
3.1. Требования к аппаратному обеспечению.	5
3.2. Требования к программному обеспечению	5
3.3. Предварительная настройка окружения	6
4. УСТАНОВКА КОМПОНЕНТОВ СИСТЕМЫ	10
4.1. Установка и настройка серверов SERVER24-app	10
4.1.1. Загрузка конфигурации	10
4.1.2. Настройка шаблона переменных	10
4.1.3. Загрузка корневых сертификатов СО ЕЭС	13
4.1.4. Установка Noproxy	13
4.1.5. Настройка NProxу	13
4.2. Установка keeplived	15
4.2.1. Настройка keeplived	15
4.2.2. Настройка Docker-engine	17
4.3. Запуск контейнера	18
4.4. Установка и настройка серверов SERVER24-Web	19
4.4.1. Установка nginx	19
4.4.2. Настройка nginx	19
4.4.3. Настройка Nginx	21
5. УСТАНОВКА И НАСТРОЙКА СЕРВЕРОВ SERVER24-DB	22
5.1. Список используемых переменных	22
5.2. Установка и настройка СУБД	24
5.2.1. Установка сервиса etcd	24
5.2.2. Настройка Etcd	24
5.2.3. Установка СУБД	26
5.2.4. Установка Patroni	26
5.2.5. Настройка Patroni	28
5.2.6. Настройка СУБД	31
5.2.7. Настройка резервного копирования СУБД	32
6. ПЕРЕДАЧА ДАННЫХ ГРУППЕ КТО	34
7. ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ	34



1. ОСНОВНЫЕ ПОНЯТИЯ, ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ

CPU	Центральный процессор
DNS	Система, которая переводит доменные имена в IP-адреса, позволяя пользователям использовать понятные имена для доступа к веб-сайтам.
Docker	Платформа для разработки, доставки и запуска приложений в контейнерах, что обеспечивает изоляцию и портативность приложений.
etcd	Распределенный, высоко доступный хранилище ключ-значение, часто используемое для хранения конфигурационных данных и метаданных.
Frontend	Часть приложения, которая взаимодействует с пользователем и отображает данные, полученные от backend.
HAProxy	Высокопроизводительный TCP/HTTP балансировщик нагрузки, который распределяет трафик между несколькими серверами для обеспечения высокой доступности и отказоустойчивости.
HDD	Устройство хранения информации, дисковый накопитель
HTTPS	Расширение протокола HTTP для поддержки шифрования в целях повышения безопасности.
IP	уникальный числовой идентификатор устройства в компьютерной сети.
Keepalived	Инструмент для обеспечения высокой доступности (HA) на уровне сети, который обычно используется в паре с HAProxy или Nginx для обеспечения отказоустойчивости.
Nexus	Репозиторий менеджер, используемый для хранения и управления артефактами, такими как библиотеки и пакеты, используемые в процессе разработки ПО.
Nginx	Веб-сервер и обратный прокси-сервер, часто используемый для балансировки нагрузки, кэширования и в качестве веб-сервера для обслуживания статического контента.
RAM	Оперативная память.
SMB	Протокол обмена сетевыми файлами.



SMTP	Сетевой протокол, предназначенный для передачи электронной почты в сетях TCP/IP.
SSH	Защищённый сетевой протокол для удалённого управления сервером через интернет.
SSL	Технология, обеспечивающая зашифрованное соединение между веб-сервером и браузером для защиты передаваемых данных.
TCP	Протокол сети интернет, который позволяет двум хостам создать соединение и обмениваться потоками данных.
UDP	Сетевой протокол транспортного уровня, используемый для установления соединений с низкой задержкой и устойчивостью к потерям между приложениями в режиме онлайн
VRRP	Сетевой протокол, предназначенный для увеличения доступности маршрутизаторов, выполняющих роль шлюза по умолчанию.
БД	Система для хранения и управления данными
ВМ	Программная эмуляция компьютера, которая выполняет операционные системы и приложения как на физическом компьютере.
ПАК ЕСМ	Комплекс программных и аппаратных средств для мониторинга и управления различными системами и сервисами.
ПО	Программное обеспечение
Сервер AD	Доменная служба Active Directory – службы каталогов корпорации Microsoft для операционных систем семейства Windows Server
СУБД	Система управления базами данных
СУБД	Программное обеспечение для создания, управления и манипулирования базами данных, обеспечивающее безопасность и целостность данных.
УЗ	Индивидуальная учетная запись пользователя в системе, включающая логин и пароль для аутентификации и авторизации.



2. НАЗНАЧЕНИЕ РУКОВОДСТВА

Инструкция описывает действия администратора по установке и настройке ИУС «ИСЭИ-24» (далее по тексту – Система).

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

3. ТРЕБОВАНИЯ К ПРОГРАММНЫМ/АППАРАТНЫМ РЕСУРСАМ

Для установки Системы необходимо подготовить сервера с операционной системой Astra Linux Special Edition в соответствии с данными, указанными в этой главе.

3.1. Требования к аппаратному обеспечению.

Рекомендованные характеристики серверов указаны в [Таблице 1](#).

Таблица 1 – Рекомендуемая конфигурация серверов Системы

Тип сервера	Кол-во	Характеристики сервера		
		vCPU core	RAM Gb	HDD Gb
server-web	2	4	8	72
server-app	2	6	8	65
server-db	3	8	8	320
ИТОГО	7	44	56	1234

Характеристики сервера соответствуют одному серверу. В строке «ИТОГО» указана итоговая сумма ресурсов для всех серверов.

3.2. Требования к программному обеспечению

1. На серверах **SERVER-web** должно быть установлено следующее ПО:
 - Операционная система – Astra Linux Special Edition 1.7 (Орёл);
 - Nginx версии 1.22.1+.



2. На серверах **SERVER-app** должно быть установлено следующее ПО:
 - Операционная система – Astra Linux Special Edition 1.7 (Орёл);
 - HAProxy версии 2.x.
 - ПО Keepalived (версия 2.2.7).
 - Docker 24.0.x.
3. На серверах **SERVER-db** должно быть установлено следующее ПО:
 - Операционная система – Astra Linux Special Edition 1.7 (Орёл);
 - СУБД – Postgres SQL Pro 16 STD;
 - ПО Patroni 2.1.12+;
 - Etcd 3.5.1+.

3.3. Предварительная настройка окружения

Для запуска Системы необходимо:

1. Запросить сертификат в формате pfx для обеспечения шифрованного соединения с пользовательским сайтом: server.so-ups.ru.

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 server.pfx:

```
sudo openssl pkcs12 -in ~/server.pfx -clcerts -nokeys -out  
/etc/nginx/ssl/server.crt  
sudo openssl pkcs12 -in ~/server.pfx -nocerts -out  
~/server.key  
sudo openssl rsa -in ~/server.key -out  
/etc/nginx/ssl/server.key
```

2. Запросить УЗ для доступа к ФПА.
3. Запросить УЗ для доступа к артефактам в Nexus.
4. В AD необходимо запросить создание системной учетной записи Server, например, domain\server, и соответствующий ей почтовый ящик, например server@comm.



5. Предоставить системной УЗ права на чтение параметров из ОИК СК-11 через REST API.
6. Для настройки подключения к пользовательскому веб-сайту из интернета по URL `https://full_domain_name`, где `full_domain_name` – полное доменное имя для пользовательского сайта, необходимо в DNS создать Aliases с `full_domain_name`, например `server.so-ups.ru`. После чего привязать к созданному Aliases полное доменное имя сервера WAF.
7. Для сервера SERVER-Web (в ДМЗ) создать в DNS внешние (белые) ip-адреса для обеспечения подключения от WAF к серверу SERVER-Web.
8. Также необходимо зарезервировать общий ip-адрес, по которому будет доступен кластер БД.
9. Для настройки системы необходимо на серверах Linux создать учетную запись пользователя “user” и добавить данного пользователя в группу sudo. Все дальнейшие настройки будут описаны для УЗ с именем “user”.

[Таблица 2](#) содержит список сетевых взаимодействий Системы.

Таблица 2 – Сетевое взаимодействие Системы

Источник	Приёмник	Протокол/Порт
Сервера приложений Системы (SERVER-app)		
Компьютер администратора Системы	Сервера приложений (Linux)	TCP-port(SSH) TCP- port
Сервер ПАК ЕСМ	Сервера приложений (Linux)	TCP- port UDP- port
Сервера приложений (Linux)	Сервер ПАК ЕСМ	UDP- port
Сервера приложений (Linux)	Сервера СУБД Системы	TCP- port TCP- port
Сервера приложений (Linux)	Сервер AD (контроллер домена)	TCP- port (LDAPS)
Сервера приложений (Linux)	Сервер ФПА – хранилище конфигурации (server)	TCP- port (HTTPS)



Сервера приложений (Linux)	Сервер ФПА – хранилище артефактов (server)	TCP- port (HTTPS) TCP- port
Сервера приложений (Linux)	Почтовый сервер	TCP- port (IMAPS), TCP- port (SMTP)
Сервера приложений (Linux)	Public API ОИК СК-11	TCP- port (HTTPS) TCP- port (HTTPS)
Сервера приложений (Linux)	Сервер CLUBR server	TCP- port TCP- port UDP- port UDP- port
Сервера приложений (Linux)	Сервер точного времени	UDP- port
Web-сервера Системы (SERVER-web)		
Компьютер администратора Системы	Web-сервера Системы	TCP- port (SSH) TCP- port (HTTPS)
Пользователи Системы	Web-сервера Системы	TCP- port (HTTPS)
Сервер ПАК ЕСМ	Web-сервера Системы - локальная инсталляция	TCP- port (HTTPS), UDP- port
Web-сервера Системы	Сервера приложений (Linux) - локальная инсталляция	TCP- port UDP- port
Web-сервера Системы	Web-сервера Системы	VRRP
Web-сервера Системы	Сервер ФПА – хранилище артефактов (server)	TCP- port (HTTPS) TCP- port
Web-сервера Системы	Сервер точного времени	UDP- port
Сервер СУБД Системы (SERVER-db)		
Компьютер администратора Системы	Сервера СУБД Системы	TCP- port (SSH) TCP- port TCP- port , TCP- port , TCP- port , TCP- port , TCP- port ,
Сервера приложений (Linux)	Сервера СУБД Системы	TCP- port , TCP- port , TCP- port , TCP- port , TCP- port , TCP- port ,
Сервера СУБД Системы	Сервера СУБД Системы	TCP- port , TCP- port , TCP- port ,



Информационно-управляющая система «ИСЭИ-24»

		TCP- port , TCP- port , TCP- port ,
--	--	---



4. УСТАНОВКА КОМПОНЕНТОВ СИСТЕМЫ

Предварительная настройка серверов Системы.

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

```
sudo apt update && sudo apt install snmpd
```

4.1. Установка и настройка серверов SERVER24-app

4.1.1. Загрузка конфигурации

Для удобства работы все необходимые для запуска Системы скрипты размещены в `git` репозитории ФПА.

Считывание данных из файлов конфигурации производится при запуске сервиса.

Для изменения конфигурации необходимо внести изменения в конфигурационные файлы и произвести перезапуск сервисов.

Для настройки сервисов необходимо подключиться к каждому серверу приложений по `SSH` и выполнять следующую последовательность действий:

```
git clone https://server.comm/server24/config.git -b main ~/config/  
cd ~/config  
git checkout main
```

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

Перейти в директорию с шаблоном запуска:

`cd ~/config/` и используя в качестве шаблона файл: `~/config/.env.example` создать новый файл: `~/config/.env` используя команду:

```
cp ~/config/.env.example ~/config/.env
```

Заполнить параметры используя информацию из следующего раздела

4.1.2. Настройка шаблона переменных

[Таблица 3](#) содержит описание переменных, используемых в шаблоне файла `.env`. В шаблоне файла некоторые переменные закомментированы, то есть содержат в начале строки символ `#`. Такие переменные являются необязательными и их изменение пользователем не подразумевается.

Таблица 3 – Список переменных файла `.env`



Переменные	Комментарий	Пример
DB_HOST	Общий IP адрес, выделенный для БД	ip
DB_PORT	Порт БД	port
DB_NAME	Имя БД	Server-db
DB_USER	Имя пользователя БД	Server_user
DB_PASSWORD	Пароль пользователя БД	password
HIKARI_MAX_LIFETIME	Таймаут максимального времени сессии с БД	39000
TEMP_DIR	Путь к временной папке системы	/temp
MAIL_HOST	Хост почтового сервера	server
MAIL_PORT	Порт почтового сервера	port
MAIL_USE_SSL	Включение\отключение SSL	True
MAIL_SSL_SOCKET_FACTORY_PORT	Порт, к которому можно подключиться при использовании фабрики сокетов.	port
MAIL_USER	Имя пользователя от почтового ящика	name@comm
MAIL_PASSWORD	Пароль пользователя	password
MAIL_FROM_TITLE	Почта для отправки уведомлений	name@comm
SERVER_URL	Адрес сервера системы в ссылках почтовых сообщений	https://ссылка
SK11_LOGIN	Логин пользователя СК-11	login
SK11_PASSWORD	Пароль пользователя СК-11	password
SK11_HOSTS	список хостов для работы с ОИК СК-11 через запятую без пробелов	https://server:port, https://server:port
SMB_DOMAIN	Домен учётной записи SMB для работы с Clubr	ldaps://ad.comm
SMB_USERNAME	Логин учётной записи SMB для работы с Clubr	login
SMB_PASSWORD	Пароль учётной записи SMB для работы с Clubr	password



ESM_ADDRESSES	Адреса серверов ECM, принимающих SNMP Trap,	ip:port, ip:port
ESM_COMMUNITY	SNMP Community	public
ESM_ENTERPRISES	SNMP префикс НТЦ ЕЭС ИК	ENTERPRISES
ESM_SYSTEM_OID	SNMP идентификатор системы ИСЭИ24 НТЦ ЕЭС ИК	oid
ESM_TRAP_MESSAGE_INTERVAL_IN_MIN	Переменная, определяющая минимальный интервал отправки повторных SNMP Trap. В минутах	5
ESM_SOURCE_IP	Адрес сервера, источника трапа, для ECM	ip
ESM_SOURCE_HOSTNAME	Имя сервера, источника трапа, для ECM	server-app-p.comm

Для удобства изменения переменные представлены в формате .env файла с объявленными переменными:

<Имя>=<Значение>

Пример работы с переменной SERVER_URL при условии того, что адрес основного сервера приложений (Linux) - https://server.so-ups.ru:

Переменная в шаблоне конфигурации:

```
SERVER_URL=http://${ SERVER_URL:- https://server.so-ups.ru
}
```

Переменная после присвоения значения:

```
SERVER_URL= https://server.comm
```

При использовании значений, содержащих технические символы (~!@#\$%^*()_-"[]{};:'/<>), значение переменной обрамляется одинарными кавычками.

Считывание данных из файлов конфигурации производится при запуске сервиса.

4.1.3. Загрузка корневых сертификатов СО ЕЭС

Для корректной работы с https в Систему необходимо загрузить корневые сертификаты СО ЕЭС.

Для этого необходимо загрузить корневые сертификат из удостоверяющего центра СО ЕЭС и скопировать их в папку ~/config/ssl.

При запуске Системы сертификаты будут помещены внутрь докер контейнера и доступны Системе.

4.1.4. Установка Noproxy

Для установки haproxy необходимо подключиться к каждой VM server-app по SSH и выполнить следующую последовательность действий:

```
sudo apt install haproxy -y  
sudo systemctl enable haproxy
```

4.1.5. Настройка HAProxy

1. Настроить конфигурационный файлы HAProxy, через команду:

```
nano /etc/haproxy/haproxy.cfg
```

Содержание конфигурационного файла должно быть следующим:

```
global  
    maxconn 100  
  
defaults  
    log global  
    mode tcp  
    retries 2  
    timeout client 30m  
    timeout connect 4s  
    timeout server 30m  
    timeout check 5s  
  
listen stats  
    mode http  
    bind *:port  
    stats enable  
    stats uri /  
  
listen postgres  
    bind *:port  
    option httpchk
```



```
http-check expect status 200
default-server inter 3s fall 3 rise 2 on-marked-down
shutdown-sessions
server node1 < IP NODE1>:port maxconn 100 check port
port
server node2 <IP NODE2>:port maxconn 100 check port port
server node3 <IP NODE3>:port maxconn 100 check port port
```

где:

port – порт статистики для haproxy;

port – порт для подключения к кластеру БД PostgreSQL ИУС ИСЭИ-24;

<IP NODE{1,2,3}> – ip-адреса всех трех узлов серверов БД;

port – порт подключения к PostgreSQL;

port – порт restart patroni.

Пример:

```
global
maxconn 100

defaults
log global
mode tcp
retries 2
timeout client 30m
timeout connect 4s
timeout server 30m
timeout check 5s

listen stats
mode http
bind *:port
stats enable
stats uri /

listen postgres
bind *:port
option httpchk
http-check expect status 200
default-server inter 3s fall 3 rise 2 on-marked-down
shutdown-sessions
server node1 ip:port maxconn 100 check port port
server node2 ip:port maxconn 100 check port port
server node3 ip:port maxconn 100 check port port
```

2. Перезагрузить HAProxy:

```
service haproxy restart
```

3. Проверить корректность работы сервиса HAProxy:

```
service haproxy status
```

Статус сервиса должен соответствовать **active (running)**.

4.2. Установка keepalived

ПО keepalived необходимо для организации отказоустойчивого кластера. Для установки keepalived необходимо подключиться к каждой ВМ server24-app по SSH и выполнить следующую последовательность действий:

```
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
```

Далее необходимо создать исполняемый файл /etc/keepalived/chk_haproxy.sh с содержимым:

```
#!/bin/bash
ha_code=$(curl --noproxy \* -m 3 -s -o /dev/null -w
%{http_code} http://localhost:port)
ha_status=$(curl --noproxy \* -m 3 -s
http://localhost:port/\;csv | grep "postgres,BACKEND" |
cut -d, -f18 )
if [[ $ha_code -eq 200 && $ha_status == UP ]]
then
    exit 0
else
    exit 1
fi
```

Так же необходимо добавить сервис в автозагрузку командой:

```
systemctl enable keepalived
```

4.2.1. Настройка keepalived

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 `<IP>` необходимо заменить на `ip` адрес, выделенный для работы сервиса, запрошенный в [п. 3.3](#).

```
global_defs {
    script_user root
    enable_script_security
}
vrrp_script chk_haproxy {
    script "ps -C chk_haproxy"
    interval 2
}
vrrp_instance SERVER_APP {
    state MASTER #BACKUP Для основного узла MASTER для
резервного BACKUP
    interface eth0 #Указываем интерфейс, к которому будет
привязан VRRP instance
    virtual_router_id id #Уникальное значение кластера
#Должен быть одинаков на всех хостах в instance
#допустимые значения от 1 до 255.
    priority id #Для основного узла указываем 110 для резервного
100.
    advert_int 4
    #Настройка аутентификации по паролю
    authentication {
        auth_type password
        auth_pass 0000
    }
    #Настройка виртуального сетевого интерфейса
    virtual_ipaddress {
        <IP> dev eth0 label eth0:vip
    }
    track_script {
        chk_haproxy
    }
}
```

После чего необходимо перезапустить сервис командой:

```
systemctl restart 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(SERVER_APP) Entering MASTER STATE
```

Для резервного сервера в выводе должно содержаться сообщение:

```
VRRP_Instance(SERVER_APP) Entering BACKUP STATE
```

4.2.2. Настройка Docker-engine

Для настройки Docker-engine на серверах приложений необходимо выполнить последовательно следующие команды:

1. Переходим в консоль root для повышения привилегий:

```
sudo su
```

2. Обновляем список доступных пакетов и устанавливаем необходимые:

```
apt-get update
```

```
apt-get install -y git curl unzip
```

3. Загружаем установочный пакет из ФПА:

```
curl -L https://asdu-fpa-nexus.com/repository/ASDU_Distributivs/docker/docker.zip -o  
~/docker.zip  
cd ~/  
unzip ~/docker.zip  
cd ~/docker
```

4. Устанавливаем Docker engine:

```
dpkg -i ./*.deb
```

5. Устанавливаем docker compose:

```
cp ./docker-compose /usr/local/bin/docker-compose  
sudo chmod +x /usr/local/bin/docker-compose  
exit
```

6. Далее добавляем в конфигурацию докера настройки сети, чтобы 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": "ip/00",
```



```
"default-address-pools": [{  
    "base": "ip/00",  
    "size": 24  
}]  
}
```

7. Производим запуск Docker-engine:

```
sudo systemctl start docker
```

- Включаем Docker-engine в автозагрузку:

```
sudo systemctl enable docker
```

- Включаем пользователя в группу docker для запуска контейнеров:

```
sudo usermod -aG docker user
```

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

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

9. Ожидаемый ответ:

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

4.3. Запуск контейнера

1. Загрузите папку с конфигурационными файлами

2. Авторизуйтесь в Nexus:

```
docker login asdu-fpa-nexus.comm
```

3. Загрузите Docker-образ с Nexus:

```
docker pull server:port/server/server-app:port
```

4. Запустите Docker-контейнер:

Для запуска Docker-контейнера достаточно запустить скрипт установки:

```
# Переходим в каталог проекта  
cd ./config  
# Запуск скрипта установки  
./start.sh
```

Данный скрипт проверит уже установленные сервисы, и установит свежие версии приложений.

Чтобы убедиться в отсутствии ошибок, необходимо через несколько минут после завершения установки выполнить команду:

```
docker ps
```

4.4. Установка и настройка серверов SERVER24-Web

4.4.1. Установка nginx

1. Для установки nginx необходимо подключиться к каждой VM server24-web по SSH и выполнить следующую последовательность действий:

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

2. Добавить сервис nginx в автозапуск и запустить сервис:

```
sudo systemctl start nginx  
sudo systemctl enable nginx
```

4.4.2. Настройка nginx

Для настройки nginx необходимо подключиться к каждой VM server24-web по SSH и выполнить следующую последовательность действий:

1. Удалить автоматически созданный файл конфигурации nginx:

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

2. Очистить директорию www командой:

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

3. Создать директорию веб сайта:

```
mkdir /var/www/server
```

4. Предоставить права УЗ user, в группу которого будут входить все DevOps-инженеры, на директорию с web-приложением server, используя команду:

```
sudo chown -R user:to-users /var/www
```

5. Заполнить настройки взаимодействия с сервисами по шаблону ниже, используя команду:

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

Шаблон:

```
upstream server-service {  
    server server-backend ip1:port;  
    server server-backend ip2:port;  
}
```

6. Заполнить конфигфайл веб-сайта по шаблону ниже, используя команду:

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



Шаблон:

```
server {
    listen port;
    server_name server.so-ups.ru;    if ($request_method !~
^ (GET|HEAD|POST|PUT|DELETE|OPTIONS|PATCH)$ ) {
        return port; }
    return port https://$host$request_uri;
}

# Добавление SSL конфигурации
server {
    listen port ssl;
    server_name server.comm;

    ssl_certificate /etc/nginx/ssl/http-server.pem;
    ssl_certificate_key /etc/nginx/ssl/http-server.key;

    ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3;
    ssl_prefer_server_ciphers on;

    client_max_body_size 50M;
    large_client_header_buffers 8 64k;

    root    /var/www/server ;

    location / {
        try_files $uri $uri/ /index.html =404;
    }

    location /server-service/ {
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_pass ссылка;
        proxy_connect_timeout 5s;
    }
}
```

7. Убедимся, что конфигурация nginx настроена правильно командой:

```
nginx -T
```

8. Перезапустим сервис nginx:

```
systemctl restart nginx
```



9. Установка и настройка web серверов закончена. Для проверки работоспособности Nginx необходимо выполнить команду:

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

Ожидаемый ответ:

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

4.4.3. Настройка Nginx

Для настройки Web серверов необходимо на каждой VM server24-web скачать артефакт frontend сервиса с ФПА. При помощи команды:

```
curl -u "username:password" -L "https://server/repository/server/server-front-1.2.6.tar.gz" -o ~/server-front.tar.gz
```

Где username:password это логин и пароль пользователя выполняющего установку в Active Directory Системного оператора.

1. Разархивировать артефакт

```
tar -xvf ~/server-front.tar.gz
```

2. Очистить директорию web сайта командой

```
rm -r /var/www/server/*
```

3. Переместить файлы сервиса в директорию веб сайта командой:

```
cp -r ./front/build/* /var/www/server/*
```

4. Удалить временные файлы сервиса:

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

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

5. После чего необходимо загрузить на каждую VM server24-web SSL сертификаты полученные в [п. 3.3](#) в каталог /etc/nginx/ssl/, расположенный на Web серверах Системы (рекомендуется использовать ПО WinSCP1).

Для проверки работоспособности веб сайта необходимо перейти по веб ссылке, соответствующей имени сайта, которое мы зарегистрировали в [п. 3.3](#). Ожидаемый результат – отображение страницы авторизации системы.



5. УСТАНОВКА И НАСТРОЙКА СЕРВЕРОВ SERVER24-DB

5.1. Список используемых переменных

В [таблице](#) представлены переменные, используемые для настройки ПО
в п. 5

Таблица 4 – Список переменных

Переменные	Пример	Комментарий
scope	pgsql_server	Название области (scope) для Patroni, должно быть одинаковым на всех узлах кластера для согласованности.
namespace	/cluster_server/	Пространство имен для Patroni, должно быть одинаковым на всех узлах кластера для согласованности.
name	postgres3	Имя узла в кластере, должно быть уникальным для каждого узла.
listen	ip:port	IP-адрес и порт, на которых REST API будет слушать запросы на данном узле.
connect_address	ip:8008	IP-адрес и порт, по которым другие узлы будут подключаться к REST API данного узла.
hosts	ip1:port,ip2:port,ip3:port	Список IP-адресов и портов всех узлов кластера etcd, используемых Patroni для координации.
username	patroni	Имя пользователя для подключения к etcd.
sqlnode	ip1/00	IP-адрес первой ноды, используемый в конфигурации pg_hba.conf для разрешения репликации.
sqlnode	ip2/00	IP-адрес второй ноды, используемый в конфигурации pg_hba.conf для разрешения репликации.
sqlnode	ip3/00	IP-адрес третьей ноды, используемый в конфигурации pg_hba.conf



		для разрешения репликации.
password	пароль	Пароль для пользователя admin в Patroni, необходимо придумать и задать уникальный пароль.
listen	ip:port	IP-адрес и порт, на которых PostgreSQL будет слушать запросы на данном узле.
connect_address	ip:port	IP-адрес и порт, по которым другие узлы будут подключаться к PostgreSQL данного узла.
data_dir	/data/patroni	Директория для хранения данных PostgreSQL, должна быть создана ранее и иметь нужные права доступа.
bin_dir	/opt/pgpro/std-16/bin	Путь к директории с исполняемыми файлами PostgreSQL.
pgpass	/tmp/pgpass	Путь к файлу pgpass для аутентификации PostgreSQL.
username (authentication.replication)	postgres	Имя пользователя для репликации PostgreSQL.
password (authentication.replication)	пароль	Пароль для пользователя репликации PostgreSQL, необходимо придумать и задать уникальный пароль.
username (authentication.superuser)	postgres	Имя суперпользователя PostgreSQL.
password (authentication.superuser)	пароль	Пароль для суперпользователя PostgreSQL, необходимо придумать и задать уникальный пароль.
bind (listen stats)	*:port	IP-адрес и порт для доступа к статистике HAProxy.
bind (listen postgres)	*:port	IP-адрес и порт для доступа к PostgreSQL через HAProxy.
node1 IP and port	ip:port	IP-адрес и порт для node1 в конфигурации HAProxy.
node2 IP and port	ip:port	IP-адрес и порт для node2 в конфигурации HAProxy.
node3 IP and port	ip:port	IP-адрес и порт для node3 в конфигурации HAProxy.



check port	8008	Порт для проверки состояния серверов в конфигурации HAProxy.
------------	------	--

5.2. Установка и настройка СУБД

5.2.1. Установка сервиса etcd

Данный сервис необходим для работы ПО Patroni и выполняет роль хранилища конфигурации и технической информации для PostgreSQL и Patroni. Для установки etcd необходимо подключиться по ssh на каждый узел кластера БД, выделенный для установки СУБД и выполнить следующие команды:

```
sudo apt-get install etcd
```

5.2.2. Настройка Etcd

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

- 1) Настроить конфигурационный файл согласно шаблону (см. ниже),
через команду:

```
nano /etc/default/etcd
```

Таблица 5 – Список переменных

Переменная	Пример	Комментарий
ETCD_NAME	name	hostname текущей машины
ETCD_LISTEN_PEER_URLS	http://ip:port	адрес текущей машины
ETCD_LISTEN_CLIENT_URLS	http://ip:port	адрес текущей машины
ETCD_INITIAL_ADVERTISE_PEER_URLS	http://ip:port	адрес текущей машины
ETCD_INITIAL_CLUSTER	name1=http://ip:port, name2=http://ip:port	адреса всех машин в кластере etcd
ETCD_INITIAL_CLUSTER_STATE	New	статус текущего кластера
ETCD_INITIAL_CLUSTER_TOKEN	etcd-cluster	токен кластера
ETCD_ADVERTISE_CLIENT_URLS	http://ip:port	адрес текущей машины



Пример:

```
[member]
ETCD_NAME=sqlnode1
ETCD_LISTEN_PEER_URLS="http://ip:port"
ETCD_LISTEN_CLIENT_URLS=" http://ip:port "
[cluster]
ETCD_INITIAL_ADVERTISE_PEER_URLS=" http://ip:port "
ETCD_INITIAL_CLUSTER="=sqlnode1= http://ip:port,sqlnode2=
http://ip:port ,sqlnode3= http://ip:port
ETCD_INITIAL_CLUSTER_STATE="new"
ETCD_INITIAL_CLUSTER_TOKEN="etcd-cluster"
ETCD_ADVERTISE_CLIENT_URLS=" http://ip:port "
```

Для проверки корректности установки etcd необходимо выполнить команду:

```
sudo systemctl status etcd
```

При успешной установке должно появиться сообщение (пример):

```
etcd.service - etcd key-value store Loaded:
loaded(/lib/systemd/system/etcd.service; enabled; vendor preset:
enabled) Active: active (running) since Tue 2023-07-19 14:21:45
UTC; 1h 23min ago Docs: https://github.com/coreos/etcd Main PID:
1234 (etcd) Tasks: 6 (limit: 4915) Memory: 52.0M CGroup:
/system.slice/etcd.service └─1234 /usr/local/bin/etcd
```

Проверить работоспособность кластера (на любом из узлов кластера):

```
sudo etcdctl member list
```

В результате будет показано состояния всех узлов кластера, и указано кто в данный момент является лидером.

Пример вывода команды:

```
90b34b35be64721: name=etcd1 peerURLs= http://ip:port
clientURLs= http://ip:port isLeader=false
19b668c907898b11: name=etcd3 peerURLs= http://ip:port
clientURLs= http://ip:port isLeader=true
bc5bb71b7803f7fe: name=etcd2 peerURLs= http://ip:port
clientURLs= http://ip:port isLeader=false
```

Создаем юнит файл сервиса etcd по шаблону (см. ниже)

```
sudo cat << EOF > /etc/systemd/system/etcd.service
```

Шаблон:

```
sudo cat << EOF > /etc/systemd/system/etcd.service
```



```
[Unit]
Description=etcd key-value store
After=network-online.target
Wants=network-online.target

[Service]
Type=notify
EnvironmentFile=-/etc/default/etcd
User=etcd
Group=etcd
ExecStart=/usr/local/bin/etcd
Restart=always
LimitNOFILE=40000

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

5.2.3. Установка СУБД

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

1. Обновить список пакетов с репозитория:

```
apt-get update
```

2. Установить пакет Postgres:

```
wget http://server/std-16/keys/pgpro-repo-add.sh
sudo sh pgpro-repo-add.sh
```

Установка

```
sudo apt-get install postgrespro-std-16
```

3. Присвоить УЗ postgres пароль командой:

```
sudo passwd postgres
```

На запрос системы необходимо дважды ввести пароль.

5.2.4. Установка Patroni

1. Необходимо подключиться по ssh на каждый узел кластера БД, выделенный для установки СУБД и остановить сервис и отключить postgres на всех узлах кластера баз данных и приложений:

```
sudo systemctl stop postgrespro-std-16
sudo systemctl disable postgrespro-std-16
```



2. Установить `patroni` на каждом из узлов кластера баз данных и приложений с помощью следующих команд:

```
#Устанавливаем Python

sudo apt-get update
sudo apt-get install python3-pip python3-dev python3-
requests postgrespro-std-16-dev postgrespro-std-16-libs -y

#Создаем файл конфигурации для python

sudo cat <<EOF > /etc/pip.conf
[global]
index =
https://<login_fpa>:<password_fpa>@server/repository/pypi-
group/pypi
index-url = https://
<login_fpa>:<password_fpa>@server/repository/pypi-group/simple
trusted-host = server
EOF
```

<login_fpa>:<password_fpa>- логин и пароль к ФПА, запрошенные в [п. 3.3.](#)

```
#Устанавливаем пакеты patroni

pip3 install --upgrade pip
export PATH="/opt/pgpro/std-16/bin/:$PATH"
pip3 install psycopg2
pip3 install patroni[etcd]==3.0.2
pip3 install psycopg2-binary

#Удаляем оригинальный инстанс СУБД

sudo rm -fr /var/lib/pgpro/std-16/data/*

#Добавляем английскую локаль

sed -i "s/# en_US.UTF-8/en_US.UTF-8/" /etc/locale.gen
locale-gen en_US.UTF-8
```

3. Создаем каталоги для хранения БД:

```
sudo mkdir -p /data/patroni
sudo chmod 700 /data/patroni
sudo chown -R postgres:postgres /data
```

4. Создаем юнит файл сервиса `patroni` по шаблону (см. ниже)

```
sudo cat << EOF > /etc/systemd/system/patroni.service
```

Шаблон:

[Unit]

```
Description=Runners to orchestrate a high-availability
PostgreSQL
After=syslog.target network.target

[Service]
Type=simple
User=postgres
Group=postgres
ExecStart=/usr/local/bin/patroni /etc/patroni.yaml
KillMode=process
TimeoutSec=30
Restart=no

[Install]
WantedBy=multi-user.target\

EOF
```

5.2.5. Настройка Patroni

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

1. Создаем настроенный файл сервиса patroni после чего корректируем переменные согласно комментариям:

```
sudo cat << EOF > /etc/patroni.yaml
scope: pgsql_server # должно быть одинаковым на всех нодах
namespace: /cluster_server/ # должно быть одинаковым на всех
нодах
name: postgres3 # должно быть разным на всех нодах

restapi:
  listen: ip3:port # адрес той ноды, в которой находится
этот файл
  connect_address: ip3:port # адрес той ноды, в которой
находится этот файл

etcd3:
  hosts: ip1:port, ip2:port, ip3:port # перечислите здесь
все ваши ноды, в случае если вы устанавливаете etcd на них же
  username: patroni
```



```
# this section (bootstrap) will be written into
Etcd:/<namespace>/<scope>/config after initializing new cluster
# and all other cluster members will use it as a `global
configuration`
bootstrap:
  dcs:
    ttl: 100
    loop_wait: 10
    retry_timeout: 10
    maximum_lag_on_failover: 1048576
    postgresql:
      use_pg_rewind: true
      use_slots: true
      parameters:
        wal_level: replica
        hot_standby: "on"
        wal_keep_segments: 512
        max_wal_senders: 5
        max_replication_slots: 5
        checkpoint_timeout: 30

  initdb:
    - encoding: UTF8
    - data-checksums
    - locale: en_US.UTF8
    # init pg_hba.conf должен содержать адреса ВСЕХ машин,
    # используемых в кластере
  pg_hba:
    - host replication postgres ::1/128 md5
    - host replication postgres 127.0.0.1/8 md5
    - host replication postgres ip1/00 md5
    - host replication postgres ip2/00 md5
    - host replication postgres ip3/00 md5
    - host all all 0.0.0.0/0 md5

  users:
    admin:
      options:
        - createrole
        - createdb

  postgresql:
    listen: ip1:port # адрес той ноды, в которой находится
    этот файл
```



```

connect_address: ip1:port # адрес той ноды, в которой
находится этот файл
data_dir: /data/patroni # эту директорию создаст скрипт,
описанный выше и установит нужные права
bin_dir: /opt/pgpro/std-16/bin # укажите путь до вашей
директории с postgresql
pgpass: /tmp/pgpass
authentication:
  replication:
    username: name
    password: *** #придумать пароль
  superuser:
    username: name
    password: *** #придумать пароль
create_replica_methods:
  basebackup:
    checkpoint: "fast"
parameters:
  unix_socket_directories: "."

tags:
  nofailover: false
  noloadbalance: false
  clonefrom: false
  nosync: false
EOF

```

2. Используя команду `nano /etc/patroni.yaml` отредактируем файл конфигурации согласно комментариям.

3. Запускаем сервис Patroni командой:

```
systemctl start patroni
```

4. Проверяем работу сервиса используя команду:

```
patronictl -c /etc/patroni.yaml list
```

5. Ожидаемый результат после запуска сервиса на всех узлах кластера:

```

+ Cluster: server (7099461315590300498) --+-----+-----+
| Member      | Host          | Role    | State  | TL | Lag in MB |
+-----+-----+-----+-----+-----+-----+
+
| postgres2   | ip            | Replica | running | 13 | 0 |
| postgres3   | ip            | Replica | running | 13 | 0 |
| postgres4   | ip            | Leader  | running | 13 |  |

```



+

5.2.6. Настройка СУБД

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

Выполнить команды в соответствии с шаблоном (см. ниже).

[Таблица 6](#) содержит описание параметров, указанных в шаблоне.

Таблица 6 – Параметры конфигурации БД

Переменные	Пример	Комментарии
\$PG_PSWD	password	Пароль привилегированной учетной записи PostgreSQL
\$DB_NAME	server-db	Имя БД
\$DB_USER	Server_user	УЗ для доступа к БД
\$DB_PASS	password	Пароль для УЗ \$DB_USER
\$MAIN_DB	ip	Общий IP адрес серверов СУБД

Шаблон:

```
#Переключиться в консоль привилегированного пользователя
СУБД
su postgres
#Войти в консоль СУБД
psql
#Изменить пароль входа в СУБД для пользователя postgres
ALTER USER postgres WITH PASSWORD '$PG_PSWD';
#Создать УЗ для БД
CREATE USER "$DB_USER" WITH PASSWORD '$DB_PASS' LOGIN;
#Создать основную БД
CREATE DATABASE "$DB_NAME";
#Предоставить права к БД для УЗ
GRANT ALL ON DATABASE "$DB_NAME" TO "$DB_USER" WITH GRANT
OPTION;
#Выйти из консоли СУБД
\q
#Выйти из консоли пользователя postgres
exit
```

Пример:

```
su postgres
psql
ALTER USER name WITH PASSWORD '*****';
CREATE USER "server_user" WITH PASSWORD '*****' LOGIN;
CREATE DATABASE "server-db";
```

```
GRANT ALL ON DATABASE "server_user" TO "server-db" WITH  
GRANT OPTION;  
\  
Exit
```

5.2.7. Настройка резервного копирования СУБД

Для создания резервных копий баз необходимо настроить сохранения резервных копий и логов транзакций в сетевой каталог. Хранение резервных копий рекомендуется на сетевом каталоге. Для облегченного доступа к резервным копиям рекомендуется создать сетевую папку на сервере под управлением любой версии Windows, а также создать учетную запись и предоставить ей права на запись как в файловой системе, так и на уровне сетевого доступа. Для настройки резервного копирования кластера СУБД Postgres на сетевой диск доступный по протоколу SMB необходимо подключиться к консоли узла 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:

```
username=<логин>  
password=<пароль>  
domain=<домен: например, domain>
```

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 «/»;



- /srv/backup – точка (каталог) монтирования на сервере Linux, созданный на шаге 3 текущего раздела;
- /root/.smbclient – полный путь файла с параметрами доступа к сетевому каталогу Windows, созданному на шаге 2 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 директорию для хранения резервных копий СУБД.

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

7. Настроить ежедневное создание полной копии СУБД. Для этого на сервере СУБД, используя команду `sudo -u postgres crontab -e` добавляем в cron строку:

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

MYIP заменить на IP сервера.

\$REPLICA_PSWD – пароль пользователя, от которого будет производиться бэкап(replication)

В результате каждый день в 22-00 будет создаваться, сжатая архиватором gzip, полная архивная копия СУБД.

6. ПЕРЕДАЧА ДАННЫХ ГРУППЕ КТО

После выполнения установки группе КТО необходимо передать:

1. IP адреса и имена ВМ Системы;
2. Пароли и УЗ для подключения к БД;
3. Пароли и УЗ для подключения к серверам Системы.

7. ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

№ п/п	Автор	Редакция	Дата	Описание изменения
1	АО «НТЦ ЕЭС Информационные комплексы»	1.0	05.09.2024	Первая версия инструкции по установке и настройке.
2	АО «НТЦ ЕЭС Информационные комплексы»	1.2.0	28.10.2024	Вторая версия инструкции по установке и настройке.
3	АО «НТЦ ЕЭС Информационные комплексы»	1.2.1	13.11.2024	Третья версия инструкции по установке и настройке. Изменено описание переменных .env файла
4	АО «НТЦ ЕЭС Информационные комплексы»	1.2.5	17.12.2024	Четвертая версия инструкции по установке и настройке. Редакторские правки .env файла
5	АО «НТЦ ЕЭС Информационные комплексы»	1.2.6	06.02.2025	Пятая версия инструкции по установке и настройке. Добавлены описания ЕСМ в .env файл
6	АО «НТЦ ЕЭС Информационные комплексы»	1.2.6.HF2	17.02.2025	Добавлены таймаут подключения nginx. Добавлено создание загрузочного unit файла для etcd
7	АО «НТЦ ЕЭС Информационные комплексы»	1.2.6.HF4	21.03.2025	Добавлен параметр максимальной продолжительности сессии с БД, внесены изменения в образец конфигурации patroni