Система Регистрации Диспетчерских Команд АО «СО ЕЭС»

ИНСТРУКЦИЯ ПО УСТАНОВКЕ И НАСТРОЙКЕ РАСПРЕДЕЛЕННОЙ ЧАСТИ СРДК (УСТАНОВКА СИСТЕМНОГО ПО)

Редакция 1.2 от 01.08.2024

Москва, 2024

Содержание

Оглавление

1.	Основные понятия, определения и сокращения		
2.	2. Назначение руководства		
3. Требования к программным/аппаратным ресурсам		ования к программным/аппаратным ресурсам	4
3	.1	Технические требования к аппаратному обеспечению	4
3	.2	Требования к программному обеспечению	4
3	.4	Сетевой доступ Ошибка! Закладка не определе	на.
4.	Уста	новка и настройка компонентов системы	5
4	.1	Комплект поставки	5
4	.2	Предварительная настройка серверов Системы	5
4	.3	Предварительное получение УЗ в смежных системах Ошиби	ka!
3	аклад	цка не определена.	
4	.4	Установка и настройка СУБД	6
	4.4.1	Установка Postgres Pro(все узлы кластера БД)	6
	4.4.2	Организация кластера etcd(все узлы кластера БД)	6
4.4.3 Запуск сервиса Patroni следящим за кластером Pos кластера БД)		Запуск сервиса Patroni следящим за кластером Postgres(все узлы гера БД)	л 9
	4.4.4	Создание баз данных и пользователей(основной сервер СУБД)	14
4	.5	Настройка резервирования базы данных (основной сервер СУБД)	15
4	.6	Восстановление кластера БД	17
	4.6.1	Восстановление кластера БД при рассинхронизации нод	17
	4.6.2	Восстановление кластера БД если все ноды не работоспособны	17
4	.7	Установка системы Docker engine(все сервера приложений)	19
5	Лист	регистрации изменений	21

БД	База данных.			
ИС	Информационная система.			
ПО	Программное обеспечение.			
СУБД	Система управления базами данных.			
У3	Учтённая запись.			
Docker	Программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации.			
WAL	Журнал перезаписи, в который записываются все транзакции, выполняемые на сервер СУБД.			
SSH	Проток для удаленного администрирования linux-серверов			
SMB Проток удаленного доступа к файлам и папкам, явля стандартом для сетей Windows				

1. Основные понятия, определения и сокращения

2. Назначение руководства

Инструкция описывает действия администраторов Системы по установке и настройке СРДК.

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

3. Требования к программным/аппаратным ресурсам

3.1 Технические требования к аппаратному обеспечению

Перечень и характеристики серверов централизованной части Системы представлены в таблице 1.

Таблица 1 – Рекомендуемое количество аппаратных ресурсов для распределенного узла

N⁰	Серверы	Кол-во	Рекомендованные характеристики		
		серверов	серверов		В
			CPU	RAM	HDD
			core	Gb	Gb
1	Центральный сервер приложений ИА (SERVER- APPLICATION-C)	2	6	12	100
2	Сервер СУБД (SERVER- DB-C)	3	2	4	125
	Итого	5	18	36	575

Для развертывания системы необходимо выделить 6 IP адресов:

- 1. З ІР адреса для серверов кластера СУБД;
- 2. 2 ІР адреса для серверов приложения;
- 3. 1 ІР адрес для виртуального интерфейса отказоустойчивости, который

будет основным адресом для доступа к системе.

3.2 Требования к программному обеспечению

Серверы приложений Системы:

- Операционная система Astra Linux 1.7;
- ПО Docker Engine версии 20.10+;
- ПО Kaspersky Endpoint Security;
- ПО НАРгоху версии 1.8+;
- ПО ПО Keepalived версии 2.0.10 +,

Серверы баз данных:

- Операционная система Astra Linux 1.7;
- ПО Kaspersky Endpoint Security;
- ПО PostgresPro Enterpise 13;
- ПО Patroni версии 2.1.4 +;
- ПО etcd версии 3.2.26+;

4. УСТАНОВКА И НАСТРОЙКА КОМПОНЕНТОВ СИСТЕМЫ

Процесс установки делится на несколько этапов, разделённых на несколько исполнителей:

П. 4.2 – 4.6 – системный администратор

П. 4.7.1,4.7.2 – администратор системы

П. 4.7.3,4.7.4 – системный администратор

4.1 Комплект поставки

Система поставляется в виде исходного кода в репозитории <u>https://server-name.local/srdk/</u>. Данный исходный код компилируется, и далее в виде dockerобразов доставляется на сервер.

4.2 Предварительная настройка серверов Системы

Для установки приложения необходимо подготовить серверы с операционной системой Astra Linux® в соответствии с данными, указанными в разделе 3 "Требования к программным/аппаратным ресурсам".

Для настройки системы необходимо создать учетную запись (УЗ) пользователя.

Все дальнейшие настройки будут описаны для УЗ с именем "user", в качестве примера.

На серверах приложений для корректного взаимодействия с системами фонда программ и алгоритмов необходимо установить на сервере корневые сертификаты для сайтов <u>https:// server-name.local/</u> и <u>https://server-name.local</u>. Для этого необходимо

- 1) открыть сайт <u>https:// server-name.local/</u>в Yandex Browser;
- 2) нажать на иконку 🔎 слева от наименования сайта в адресной строке;
- 3) нажать на «Подробнее»;
- 4) нажать на «Данные сертификата»;
- 5) в открывшемся окне «Сертификат» выбираем вкладку «Путь сертификации», далее выбираем в цепочке сертификат «корневой сертифкат»;
- 6) нажать на кнопку «Просмотр сертификата»;
- 7) в открывшемся окне «Сертификат» выбрать вкладку «Состав»;
- 8) нажать кнопку «Копировать в файл»;
- 9) в открывшемся окне нажать «Далее»;
- 10) в открывшемся окне нажать «Далее»;
- 11) задать путь и имя файла в который будет экспортирован корневой сертификат и нажать «Далее»;
- 12) нажать «Готово»;

13) должно появиться окно «Экспорт успешно выполнен», где нажать «Ок».

После этого необходимо скопировать полученный корневой сертификат на сервер Системы в формате .crt в каталог:

```
sudo cp CEPTUΦUKAT.crt /usr/local/share/ca-
certificates/
```

И выполнить обновление корневых сертификатов:

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

4.3 Установка и настройка СУБД

4.3.1 Установка Postgres Pro(все узлы кластера БД)

Для установки Postgres PRO Enterprise необходимо подключиться по ssh на каждый узел кластера БД, выделенный для установки СУБД и выполнить следующие:

```
#Повышаем привилегии пользователя
sudo su
#Добавляем репозиторий Postgres PRO Enterprise
echo "deb [trusted=yes arch=amd64]
http://repositoriiserver-name.local/ postgrespro13 main
contrib non-free">> /etc/apt/sources.list.d/pgpro.list
#Добавляем исключение для прокси
echo 'Acquire::http::PROXY::.local repositoriiserver-
name.local"DIRECT";' >> /etc/apt/apt.conf.d/noproxy
#Обновить список пакетов с репозитория
apt-get update
#Установить пакет Postgres и rsync
apt install -y postgrespro-ent-13
```

4.3.2 Организация кластера etcd (все узлы кластера БД)

Не рекомендую устанавливать etcd на тех же машинах, где будет находится patroni и postgresql, так как для etcd очень важна нагрузка на диски но в целях уменьшения кол-ва BM etcd будет установлен на BM кластера СУБД.

Для установки etcd необходимо подключиться по ssh на каждый узел кластера БД, выделенный для установки СУБД.

```
#Загружаем необходимые пакеты
apt-get update
apt-get install -y git curl unzip
#Загружаем установочный пакет
curl -L https:// server-name.local/repository/
Distributivs/etcd/etcd.zip -o ~/etcd.zip
cd ~/
unzip ~/etcd.zip
cd ~/etcd
```

```
#Устанавливаем ETCD
dpkg -i ./*deb
```

После установки необходимо остановить сервисы etcd на всех узлах кластера баз данных и приложений:

systemctl stop etcd

Удалить содержимое папки /var/lib/etcd/ на всех узлах кластера баз данных и приложений, для создания нового кластера:

```
rm -rf /var/lib/etcd/*
```

Изменить конфигурационный файл /etc/default/etcd на всех узлах кластера баз данных и приложений в соответствии с шаблоном:

ETCD_NAME=<HOSTNAME> ETCD_LISTEN_PEER_URLS="http://<MYIP>:PORT1,http://127.0.0. 0: PORT1" ETCD_LISTEN_CLIENT_URLS="http://<MYIP>: PORT2,http://127.0.0.1: PORT2" ETCD_INITIAL_ADVERTISE_PEER_URLS="http://<MYIP>:PORT1" ETCD_INITIAL_CLUSTER="<HOSTNAME NODE N>=http://<IP NODE N>:PORT1,<HOSTNAME NODE N+1>=http://<IP NODE N+1>:PORT1,<HOSTNAME NODE N+2>=http://<IP NODE N+2>:PORT1" ETCD_INITIAL_CLUSTER_STATE="new" ETCD_INITIAL_CLUSTER_TOKEN="etcd-cluster" ETCD_ADVERTISE_CLIENT_URLS="http://<MYIP>:PORT2"

, где

<HOSTNAME> - полное имя сервера, которое прописано на DNS сервере (пример – appserversoi-name.local);

<МҮІР> - адрес текущего сервера;

<HOSTNAME NODE N>, <HOSTNAME NODE N+1>... - полные имена узлов, участвующих в кластере;

<IP NODE N>, <IP NODE N+1> - ір-адреса узлов, участвующих в кластере;

Количество узлов должно быть нечетным.

При использовании прокси узлы etcd должны быть добавлены в строку исключения прокси сервера (no_proxy)

Запустить сервис etcd на всех узлах кластера баз данных и приложений:

systemctl start etcd

Проверить работоспособность кластера (на любом из узлов кластера):

etcdctl member list

В результате мы должны увидеть состояния всех узлов кластера, и кто в данный момент является лидером.

Пример вывода команды:

90b34b35be64721: name=etc1 peerURLs=http://ip-adress:port clientURLs=http://ip-adress:port isLeader=false 19b668c907898b11: name=etc3 peerURLs=http://ip-adress:port clientURLs=http://ip-adress:port isLeader=true bc5bb71b7803f7fe: name=etc2 peerURLs=http://ip-adress:port clientURLs=http://ip-adress:port isLeader=false

Сгенерировать согласно политике СО пароли для служебного пользователя etcd (patroni) и для администратора etcd (root). В пароле рекомендуется использование только английских больших и маленьких букв и цифр.

На лидере кластера для сервиса etcd создать учетную запись администратора etcd (root) и учетную запись служебного пользователя etcd (patroni) следующими командами:

etcdctl user add root etcdctl role add srdk_patroni_clust etcdctl role grant srdk_patroni_clust --readwrite --path "/cluster_srdk/*" etcdctl user add patroni etcdctl user grant patroni --roles srdk patroni clust

в процессе создания учётных записей в консоли появится запрос ввода нового пароля для каждого нового пользователя, в ответ на запрос необходимо вводить соответствующие пароли сформированные ранее. Пароль необходимо вводить сразу, если после ввода появляется сообщение context deadline exceeded, то это означает что превышено время ожидания и процедуру создания пользователя не выполнена и её необходимо произвести повторно. Убедиться в успешном создании групп и пользователей:

sudo etcdctl --username patroni role get srdk_patroni_clust sudo etcdctl --username patroni user get patroni sudo etcdctl --username patroni user get root

Включить авторизацию

sudo etcdctl auth enable Отключить у гостевой роли права на чтение и запись sudo etcdctl --username root role revoke guest -path '/*' --readwrite

4.3.3 Запуск сервиса Patroni следящим за кластером Postgres (все узлы кластера БД)

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 systemct1 stop postgrespro-ent-13
sudo systemct1 disable postgrespro-ent-13

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-ent-13-dev postgrespro-ent-13-libs -y
#Создаем файл конфигурации для python
sudo cat <<EOF > /etc/pip.conf
[global]
index = https://login:password@server-
name.local/repository/pypi-group/pypi
index-url = https://login:password@server-
name.local/repository/pypi-group/simple
trusted-host = server-name.local
```

EOF #Устанавливаем пакеты patroni pip3 install --upgrade pip export PATH="/opt/pgpro/ent-13/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/ent-13/data/* #Добавляем английскую локаль sed -i "s/# en_US.UTF-8/en_US.UTF-8/" /etc/locale.gen locale-gen en_US.UTF-8

Создаем каталоги для хранения БД:

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

Создаем юнит файл сервиса patroni

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

Создаем настроечный файл сервиса patroni после чего корректируем переменные согласно комментариям

sudo cat << EOF > /etc/patroni.yaml

scope: pgsql_sepg # должно быть одинаковым на всех нодах namespace: /cluster_srdk/ # должно быть одинаковым на всех нодах

name: postgres3 # должно быть разным на всех нодах

restapi:

listen: ip-adress:port # адрес той ноды, в которой находится этот файл

connect_address: ip-adress:port # адрес той ноды, в которой находится этот файл

etcd:

hosts: ip-adress:port, ip-adress:port, ip-adress:port # перечислите здесь все ваши ноды, в случае если вы устанавливаете etcd на них же

username: patroni

password: пароль_пользователя{}patroni{}coзданный при настройке etcd

```
# 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
```

Для включения логирования необходимо раскомментировать параметры ниже:

#log_directory: pg_log
#log_duration: true
#log_min_duration_statement: 1
#logging_collector: true

initdb:

- encoding: UTF8
- data-checksums
- locale: en US.UTF8

init pg_hba.conf должен содержать адреса BCEX машин, используемых в кластере

pg_hba:

- host replication postgres ::1/128 md5
- host replication postgres 127.0.0.1/8 md5
- host replication postgres ip-address/24 md5
- host replication postgres ip-address /24 md5
- host replication postgres ip-address /24 md5
- host all all 0.0.0/0 md5

users:

admin:

password: *** #придумать пароль options:

- createrole

```
postgresql:
```

listen: ip-address:port # адрес той ноды, в которой находится этот файл

```
connect_address: ip-address: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/ent-13/bin # укажите путь до вашей директории с postgresql

pgpass: /tmp/pgpass

authentication:

replication:

username: postgres

password: *** #придумать пароль

superuser:

username: postgres

password: *** #придумать пароль

create replica methods:

basebackup:

checkpoint: "fast"

parameters:

unix socket directories: "."

tags:

nofailover: false noloadbalance: false clonefrom: false nosync: false

EOF

Используя команду

nano /etc/patroni.yaml

редактирум файл конфигурации согласно комментариям.

Запускаем сервис Patroni командой:

systemctl start patroni

Проверяем работу сервиса используя команду:

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

Ожидаемый результат после запуска сервиса на всех узлах кластера:

4.3.4 Создание баз данных и пользователей (основной сервер СУБД)

Для настройки основного сервера СУБД необходимо создать учетные записи и базы данных для сервисов Системы. Для этого необходимо определить лидера кластера Patroni используя команду:

patronictl -c /etc/patroni.yaml list

После чего подключиться через SSH к лидеру кластера и выполнить команды в соответствии с шаблоном (см. ниже).

Таблица 3 содержит описание параметров, указанных в шаблоне.

Таблица 3 – Параметры конфигурации Б	оаметры конфигурации БД
--------------------------------------	-------------------------

Переменные	Пример	Комментарий	
\$PG_RSWD	password	Пароль привилегированной учетной записи postgres	
\$BRANCH_DB_HOST	ip-address	IP адрес лидера кластера patroni	
\$BRANCH_DB_PORT	port	Порт адрес оперативной базы данных	
\$BRANCH_DB_NAME	Srdkbranch	Название адрес оперативной базы данных	
\$BRANCH_DB_USER	branchuser	Пользователь адрес оперативной базы данных	
\$BRANCH_DB_PASS	password	Пароль пользователя адрес оперативной базы данных	

Шаблон:

#Boйти в консоль СУБД под привилегированным пользователем su postgres -c "psql -h \$BRANCH_DB_HOST" #Coздать УЗ для оперативной БД CREATE USER "\$BRANCH_DB_USER" WITH PASSWORD '\$BRANCH_DB_PASS' LOGIN; #Coздать оперативную БД CREATE DATABASE "\$BRANCH_DB_NAME"; #Предоставить права к БД для УЗ

```
GRANT ALL ON DATABASE "$BRANCH_DB_NAME" TO
"$BRANCH_DB_USER" WITH GRANT OPTION;
#Выйти из консоли СУЕД
\q
Пример:
su postgres -c "psql -h $BRANCH_DB_HOST"
CREATE USER "branchuser" WITH PASSWORD '*****' LOGIN;
CREATE DATABASE "srdkbranch";
GRANT ALL ON DATABASE "srdkbranch" TO "branchuser" WITH
GRANT OPTION;
\q
exit
```

4.4 Настройка резервирования базы данных (основной сервер 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.local>
```

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
```

• //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 директории 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 –е 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 5000
```

, где {MYIP} заменить на общий IP адрес сервера приложений.

\$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.

8. Для очистки каталога с резервными копиями СУБД на сервере СУБД используется команда sudo –u postgres crontab –e и добавляется в cron строка. Например, такое задание:

```
40 23 * * * /usr/bin/find /srv/backup/postgres/ -maxdepth
1 -type d -mtime +14 -exec rm -rf {} \;
```

В результате ежедневно будет производится очистка резервных копий СУБД, будут удалены архивы старше 14 дней.

ВАЖНО: глубину хранения резервных копий определяет администратор, процесс удаления старых резервных копий также определяет администратор. Для обеспечения возможности восстановления данных в случае сбоя необходимо наличие всех резервных копий в соответствии с инструкциями.

9. В случае если sudo -u postgres crontab -e завершается с ошибкой, для выполнения пунктов 7 и 8 необходимо добавить в файл /etc/crontab строки

```
00 22 * * * postgres PGPASSWORD="$REPLICA_PSWD"
pg_basebackup -h MYIP -U replication -F t -D
/srv/backup/postgres/$(date +\%Y\%m\%d) -X stream -z -p
5000
```

```
40 23 * * * postgres /usr/bin/find /srv/backup/postgres/ - maxdepth 1 -type d -mtime +14 -exec rm -rf {} \;
```

С аналогичной заменой МУІР на адрес сервера.

4.5 Восстановление кластера БД

4.5.1 Восстановление кластера БД при рассинхронизации нод

Для проверки корректной работы репликаций, достаточно обратится к статусу кластера:

patronictl -c /etc/patroni.yaml list

Если в выводе напротив сервера реплики в столбце «Lag in MB» значение более 10Мб:

, тогда необходимо подключится к соответствующе ноде по ssh, и выполнить там команду повторной инициализации ноды:

patronictl -c /etc/patroni.yaml reinit {имя кластера} {имя ноды}

Бывает что при инициализации в логе патрони имеется запись :

```
ERROR: could not open file "pg_wal/0000002E.history": No such file or directory
```

Имя файла может отличаться. В данной ситуации необходимо создать данный файл на лидере кластера в директории /data/patroni/pg_wal/, в данном примере:

touch /data/patroni/pg wal/000002E.history

После этого передача данных восстановится, и кластер перейдет в консистентное состояние

4.6.2 Восстановление кластера БД если все ноды не работоспособны

Отключить авторизацию в кластере etcd

etcdctl --username root auth disable

Вернуть права гостевой роли

etcdctl role grant root --readwrite --path "/*"

Для восстановления кластера СУБД необходимо подключиться к каждому из

узлов кластера по SSH:

Сделать копию конфигурационного файла patroni

cp /etc/patroni.yaml /etc/patroni.yaml.reserv

Остановить службы СУБД и Patroni на всех нодах кластера:

systemctl stop patroni

, после чего

удалить каталог баз данных на всех нодах кластера. А так же удалить данные о предыдущем состоянии кластера из etcd:

etcdctl rm /cluster_srdk/pgsql_sepg/initialize

На лидере создать папку /srv/backup/tmp/patroni и распаковать в неё резервную копию.

Создать файл /etc/patroni/pg_base_restore.sh с содержимым:

```
#!/bin/bash
mv /srv/backup/tmp/patroni/* /data/patroni/
```

, и сделать его исполняемым:

```
chmod +x /etc/patroni/pg_base_restore.sh
```

Исправить файл конфигурации patroni /etc/patroni.yml для восстановления:

```
# initdb:
# - encoding: UTF8
# - data-checksums
# - locale: en_US.UTF8
method: pgbase
pgbase:
command: /etc/patroni/pg_base_restore.sh
keep existing recovery conf: true
```

Запустить службу Patroni на лидере:

systemctl start patroni

, и удостовериться в том, что СУБД работает и принимает подключения:

```
psql -h {MYIP} -U postgres -p 5432 -c '\conninfo'
```

```
, где {MYIP} – адрес текущего сервера БД.
Восстановить файл конфигурации
patroni в исходное состояние, путем
```

исправления файла конфигурации patroni /etc/patroni.yml :

```
initdb:
- encoding: UTF8
```

```
- data-checksums
- locale: en_US.UTF8
# method: pgbase
# pgbase:
# command: /etc/patroni/pg_base_restore.sh
# keep_existing_recovery_conf: true
```

Запустить службу Patroni на репликах и произвести переинициализацию реплик:

```
patronictl -c /etc/patroni.yaml reinit {имя_кластера} {имя ноды}
```

Включить авторизацию в кластере etcd:

sudo etcdctl auth enable

Отключить у гостевой роли права на чтение и запись:

```
sudo etcdctl --username root role revoke guest -path '/*'
--readwrite
```

4.6 Установка системы Docker engine (все сервера приложений)

Для установки Docker Engine требуется подключиться к серверу приложений, на котором будут использоваться сервисы, по ssh и выполнить последовательно следующие команды:

```
#Перейти в root для повышения привилегий
sudo su
#Обновить список доступных пакетов
apt-get update
#Установить пакеты необходимые для добавления репозитория
Docker
apt-get install -y apt-transport-https ca-certificates
curl gnupg-agent software-properties-common
#Загрузить ключ репозитория Docker
curl -fsSL https://server-name.local/repository/apt-proxy-
docker/linux/debian/gpg | apt-key add -
#Добавить репозиторий Docker
echo "deb https:// server-name.local/repository/apt-proxy-
docker/linux/debian buster stable" >
/etc/apt/sources.list.d/docker.list
#Добавляем исключение для прокси
echo 'Acquire::http::PROXY:: server-name.local "DIRECT";'
>> /etc/apt/apt.conf.d/noproxy
#Обновить список доступных пакетов
apt-get update
#Установить Docker-engine
apt-get install -y docker-ce docker-ce-cli containerd.io
```

Далее добавляем в конфигурацию докера настройки сети, дабы исключить использование подсетей, занятых во внутренних сетях СО.

Создаем, или открываем файл /etc/docker/daemon.json и добавляем туда, следующую конфигурацию:

```
{
    "live-restore": true,
    "bip": "ip-address/24",
    "default-address-pools": [{
        "base": " ip-address/16",
        "size": 24
    }]
}
```

После этого:

```
#Запустить демон Docker-engine
systemctl start docker
#Включить демон Docker-engine в автозагрузку
systemctl enable docker
#Включить пользователя в группу docker для запуска
контейнеров.
usermod -aG docker user
#Выходим из привелегированного режима
exit
```

Для проверки установки необходимо выполнить команду:

systemctl status docker |grep active

Ожидаемый ответ:

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

5 ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

№п/п	Автор	Редакция	Дата	Описание изменения