



# Система Регистрации Диспетчерских Команд АО «СО ЕЭС»

# ИНСТРУКЦИЯ ПО УСТАНОВКЕ И НАСТРОЙКЕ ЦЕНТРАЛИЗОВАНОЙ ЧАСТИ СРДК (ПРИКЛАДНОЕ ПО)

Редакция 1.2 от 01.08.2024

Москва, 2024





# Содержание

1.	Осн	овные понятия, определения и сокращения	3
2.	Назі	начение руководства	3
3	. По	одготовка конфигураций приложения	4
	4.	Установка приложений	7
	5.	Настройка отказоустойчивости приложения	7
	6.	Настройка НАргоху	10
7	. 06	бновление приложения	11





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

Для запуска сервиса необходимо произвести предварительные настройки.

#обновить список доступных пакетов sudo apt-get update

Далее необходимо установить пакет конфигураций.

Далее необходимо перейти в директорию с конфигурацией (cd ./config/) и произвести настройку.

Для настройки конфигурации необходимо скопировать файл с шаблоном конфигурации (ср ./.env-center-example ./.env) и заполнить в нем переменные, указанные в таблице 4.

Таблица 1 Список переменных, используемых в env-example

Переменные	Пример	Комментарии
\$CENTER_DB_HOST	ip-address	IP адрес, используется локальный
		адрес текущего сервера
		приложения
\$CENTER_DB_PORT	port	Порт
\$CENTER_DB_NAME	bd_name	Название оперативной базы данных
\$CENTER_DB_USER	bd_user	Пользователь оперативной базы данных
\$CENTER_DB_PASS	password	Пароль пользователя оперативной базы данных
\$ARCHIVE_DB_HOST	ip-address	IP адрес, используется локальный
		адрес текущего сервера приложения
\$ARCHIVE_DB_PORT	port	Порт
\$ARCHIVE_DB_NAME	bd_name	Название архивной базы данных
\$ARCHIVE_DB_USER	bd_user	Пользователь архивной базы данных
\$ARCHIVE_DB_PASS	password	Пароль пользователя архивной базы данных
\$BACK_END_PORT	port	Порт, на котором работает бэкэнд
\$FRONT_END_PORT	port	Порт, на котором работает фронтэнд
\$FRONT_END_SSL_PORT	port	Порт, на котором работает фронтэнд на защищенном протоколе
\$LDAP_CDU_URL	domen.local	Адрес расположения ldap сервера
\$AUTH_CDU_USER	domen\login	УЗ используемая для
		сканирования ldap каталогов, и авторизации пользователей
\$AUTH_CDU_PASS	password	Пароль от УЗ AUTH_CDU_USER
\$LDAP_ODUXX_URL	domen.local	Адрес расположения ldap сервера
\$AUTH_ODUXX_USER	domen\login	УЗ используемая для





		сканирования ldap каталогов, и
		авторизации пользователей
\$AUTH_ODUXX_PASS	password	Пароль от УЗ AUTH_ODUXX_USER
\$SEARCH_GROUP_ODU_XX	OU=SRDK	LDAP Путь по каталога с группой
		доступа ADMIN_SRDK_GROUP
\$ESS_CONF_URL	http://server-	Адрес для подключения к АРІ ЕСС
	name:port/api/xxxx/xxxx	
\$ESS_CONF_NAME	domen\login	УЗ используемая для взаимодействия с ЕСС
\$ESS_CONF_PASS	password	Пароль от УЗ ESS_CONF_NAME
\$SERVER_IP_ADDRESS	ip-address	IP адрес виртуального интерфейса
\$SK11_USER	domen\login	УЗ используемая для взаимодействия с СК-11
\$SK11_PASS	password	Пароль от УЗ SK11_USER
\$SK11_MODEL	xxxxxxx-0000-0000-	Идентификатор объектной модели
	xxxx-000000xxxxxx	СК-11 к объектам которой будут
		осуществляться доступ.
\$ADMIN_SRDK_GROUP	group	Группа доступа администраторов к системе
\$BRANCH_URL	https://server-	URL распределенной части ИА
	name.so/api/v1	
\$ELK_USER	login	Учетная запись системы сбора логов
\$ELK_PASSWORD	password	Пароль УЗ ELK_USER
\$ESM_ADDRESS	ip-address	Адрес подключения к системе ЕСМ
\$ESM_PORT	port	Порт подключения к системе ЕСМ
\$ESM_COMMUNITY	public	Сообщество чтения ЕСМ
\$ESM_APPOID	5560000	OID системы в ECM
\$PASSWORD_ENCODE_PASSWO RD	1111	Ключ шифрования паролей в БД
\$PASSWORD_ENCODE_SALT	2222	Дополнительный ключ шифрования паролей в БД
\$APP_MODES_BASEURL	http://server-	Адрес подключения к системе
	name.local:port	модес терминал.
\$MAIL_DEFAULT_HOST	server-name.local	Адрес почтового сервера
\$MAIL_DEFAULT_PORT	port	Порт почтового сервера
\$MAIL_DEFAULT_AUTH	true	Использование авторизации
\$MAIL_DEFAULT_USERNAME	login	УЗ для подключения к почтовому серверу
\$MAIL_DEFAULT_PASSWORD	password	Пароль для подключения к почтовому серверу
\$MAIL_DEFAULT FROM	login@domen.local	Имя почтового ящика, с которого
		производиться отправка почтовых уведомлений
\$MAIL DEFAULT STARTTLS	true	Использование TLS-шифрования
	1	





\$OPAM_PATH	/home/user/opam	Путь к месту монтирования сетевой папки ОпАМ
\$APP_MEM_USAGE	XG	Использование максимального количества оперативной памяти приложением. Необходимо <sup>1</sup> / <sub>2</sub> от всей.
\$APP_MODES_ID_LAYER	id	Идентификатор слоя ПБР-24
\$GRAPH_TIMEZONE	3	Часовой пояс для отображения графиков
\$APP_MODES_MAX_IN_MEMOR Y_KBYTES	8192	объем данных полученного из modes
\$APP_HTTP_REQUEST_LOGGING _ENABLED	true	Включает логирование входящих запросов.
\$APP_HTTP_REQUEST_LOGGING _STORAGE	/var/log/SRDK/xxxx	Директория для хранения входящих запросов
\$APP_HTTP_REQUEST_LOGGING _ENDPOINTS	/api/v1/xxxx/saveDataP G	Список API URI, по которым необходимо для логирования
\$APP_HTTP_REQUEST_LOGGING _AGE	P7D	Максимальный срок хранения файлов логов
\$APP_TIME_ZONE	Europe/Moscow	Часовой пояс
\$APP_SK11_MONITORING_ENAB LED	false	Включен ли мониторинг
\$APP_SK11_MONITORING_STOR AGE	/var/log/SRDK/sk11	Директория хранения логов в СК- 11
\$APP_SK11_MONITORING_AGE	P1D	Максимальный срок хранения файлов логов
\$APP_SK11_MONITORING_ENDP OINT_PATTERNS	'/**/measurement- values/**'	Список шаблонов АРІ запросов для мониторинга
\$APP_DISTRIBUTED_PART_REA D_TIMEOUT	120000	Время ожидания входящего трафика от распределенной части в миллисекундах.
\$APP_SK11_CERT		Зашифрованный контейнер с корневыми сертификатом
\$APP_DISTRIBUTED_PART_URL_ SUFFIX		Эту переменную необходимо добавить, но значение присвоить пустое.
\$APP_DISTRIBUTED_PART_URL_ SUFFIX		Зашифрованный контейнер с корневыми сертификатом УЦ
\$APP_DISTRIBUTED_PART_SSL_ KEY_STORE_PASSWORD		Пароль от контейнера с корневыми сертификатом УЦ

Система Регистрации Диспетчерских Команд АО «СО ЕЭС»

Необходимо заполнить все поля LDA\_XX\_URL, AUTH\_XX\_USER, AUTH\_XX\_PASS и SEARCH\_GROUP\_ODU\_XX т.к. это необходимо для корректного запуска системы. Значение можно использовать те же, что и для вашего домена.





Далее необходимо внести правки в конфигурационный файл балансировщика:

```
Переходим в каталог nginx
cd ./nginx/
Копируем пример конфигурации
cp ./default.conf.example ./default.conf
```

Затем исправляем в полученном файле переменные:

\${SERVER\_IP\_ADDRESS} - адрес виртуального интерфейса \${URL} - доменное имя системы

Затем необходимо добавить сертификат и приватный ключ для корректной работы https. Сертификат и ключ необходимо положить в каталог ./config/nginx/.

# 4. Установка приложений

Для установки всех приложений достаточно запустить скрипт установки:

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

В выводе должно быть 2 контейнера с имена соответствующими названиям сервисов, а именно: back-end и front-end, и в колонке STATUS у каждого из них должно быть значение – UP N minutes (где N количество минут после установки).

# 5. Настройка отказоустойчивости приложения

Данные действия аналогично выполняются на всех серверах приложений с незначительными отличиями в конфигурации. Для реализации отказоустойчивости, необходимо выделить дополнительный виртуальный IP адрес, через который и будет производиться подключение пользователей к системе. В дальнейшем для настройки будут использоваться следующие переменные, которые необходимо будет заменить в конфигурациях:

\$IP\_MASTER – IP адрес основного сервера;





## \$IP\_SLAVE – IP адрес резервного сервера; \$VIRT\_IP – основной IP адрес системы.

Для установки системы необходимо установить keepalived:

```
apt-get update
apt-get install keepalived
```

Далее создаем скрипт для проверки состояния приложения /etc/keepalived/script.sh следующего содержания:

#!/bin/bash lockfile=/tmp/keepalived\_lock

```
if ( set -o noclobber; echo "$$" > "$lockfile") 2> /dev/null;
then
```

```
front_code=$(curl --noproxy \* -k -s -o /dev/null -w %{http_code} -m 1 https://localhost)
back_code=`curl
                          --noproxy
                                             \*
                                                         -.5
                                                                   -m
                                                                               1
                                                                                         /dev/null
http://localhost:port/api/v1/actuator/health/keepalived/\python3
                                                                    -C
                                                                           "import
                                                                                      sys,
                                                                                             json;
print(json.load(sys.stdin)['status'])"`
if [[ $front_code -eq 200 && $back_code == UP ]]
then
   echo 0 > "/tmp/chk_keep_status"
  rm -f "$lockfile"
  trap - INT TERM EXIT
  exit 0
else
   echo 1 > "/tmp/chk_keep_status"
  rm -f "$lockfile"
  trap - INT TERM EXIT
  exit 1
fi
else
  echo "Failed to acquire lockfile: $lockfile."
  kill -9 $(cat $lockfile) 2>/dev/null
  rm -f "$lockfile"
  exit $(cat /tmp/chk_keep_status)
fi
```

После создания файла, нужно разрешить для данного файла выполнение, командой:

```
chmod 777 /etc/keepalived/script.sh
```

Далее создаем непосредственно конфигурационный файл /etc/keepalived/config.conf:

```
vrrp_script chk_app {
    script "/etc/keepalived/script.sh"
```





```
Система Регистрации Диспетчерских Команд АО «СО ЕЭС»
```

```
user nobody
    interval 2
    fall 3
    rise 3
}
global defs {
    router id uMASTER
    enable script security
}
vrrp instance VI 1 {
    state MASTER
    interface eth0
    virtual router id 230
    priority 101
    advert int 1
    unicast src ip $IP MASTER
    unicast peer {
      $IP_SLAVE
    }
   virtual ipaddress {
       $VIRT IP/32 dev eth0 label eth0:1
    }
    track script {
      script
    }
}
```

```
Конфигурация резервного сервера аналогична, но с незначительными отличиями:
```

```
vrrp_script chk_app {
    script "/etc/keepalived/script.sh"
    user nobody
    interval 2
    fall 3
    rise 3
}
global_defs {
    router_id uBACKUP
    enable_script_security
}
vrrp_instance VI_1 {
    state BACKUP
    interface eth0
    virtual router id 230
```





```
Система Регистрации Диспетчерских Команд АО «СО ЕЭС»
```

```
priority 100
advert_int 1
unicast_src_ip $IP_SLAVE
unicast_peer {
    $IP_MASTER
}
virtual_ipaddress {
    $VIRT_IP /32 dev eth0 label eth0:1
}
track_script {
    chk_app
}
```

После сохранения конфигурационных файлов необходимо запустить службу keepalived, а так же настроить автоматический запуск при старте системы:

```
systemctl start keepalived systemctl enable keepalived
```

# 6. Настройка НАргоху

}

Для настройки сервиса необходимо установить haproxy:

apt-get install haproxy -y

и внести конфигурацию в файл /etc/haproxy/config.cfg:

```
qlobal
    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 *:5000
    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
server node2 <ip node2>:port maxconn 100 check port
server node3 <ip node3>:port maxconn 100 check port
```

В указанной конфигурации необходимо заменить параметры <ip node1,2,3> - на ip-адреса узлов кластера СУБД

После этого перезапускаем сервис:

systemctl restart haproxy systemctl enable haproxy

Для проверки установки необходимо выполнить команду:

systemctl status haproxy | grep active

Ожидаемый ответ:

Active: active (running)

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

В выводе должно быть 2 контейнера с имена соответствующими названиям сервисов, а именно: back-end и front-end, и в колонке STATUS у каждого из них должно быть значение – UP N minutes (где N количество минут после установки).

#### 7. Обновление приложения

Для обновления приложения достаточно зайти в каталог config, и запустить скрипт установки:

cd ~/config

Сравнить файлы .env и .env-branch-example, если появились новые параметры, необходимо добавить их в .env, и заполнить.

./script.sh

Если необходимо обновить конкретный компонент, то можно запустить cкрипты script1.sh или script2.sh

В случае изменения каких либо настроек, например учетных данных, необходимо внести соответствующие изменения в .env, и после этого так же запустить скрипты script.sh, script1.sh или script2.sh





# 8. Настройка интерфейса для взаимодействия с распределенными частями СРДК.

В интерфейсе распределенной части Системы необходимо настроить взаимодействие с распределенными частями Системы:

IP = ip-address:port

В текущей версии для каждой части Системы необходимо указать IP или доменное имя без http или https вначале. А также порт бэкенда (port).

№п/п	Автор	Редакция	Дата	Описание изменения

#### Лист регистрации изменений