АКЦИОНЕРНОЕ ОБЩЕСТВО
«СИСТЕМНЫЙ ОПЕРАТОР ЕДИНОЙ ЭНЕРГЕТИЧЕСКОЙ СИСТЕМЫ»

ИНСТРУКЦИЯ ПО УСТАНОВКЕ И НАСТРОЙКЕ

программного обеспечения мониторинга синхронных качаний активной мощности по данным СМПР в режиме реального времени

Москва, 2024

СОДЕРЖАНИЕ

Перечень сокращений

Таблица 1. Перечень сокращений

Сокращение	Описание или расшифровка
Адаптер	Выделенная программная служба, выполняющая конкретную функцию (обычно подключения к внешним системам)
АССИ СМПР	Автоматизированная система сбора данных с регистраторов системы мониторинга переходных режимов
БД	База данных
МЭК-104	ГОСТ Р МЭК 60870-5-104-2004 – стукандарт с набором протоколов передачи данных телемеханики
ПО СКАМ	Программное обеспечение мониторинга синхронных качаний активной мощности в контролируемых сечениях
СК	Синхронные качания
СУБД	Система управления базами данных
ТИ	Телеизмерение
API	Application Programming Interface - Интерфейс программирования приложений

1 Состав ПО СКАМ для установки

Программное обеспечение СКАМ состоит из следующих элементов:

- системное ПО;
- Docker-образы компонентов;
- конфигурации.

1.1 Системное ПО

Системное ПО обеспечивает среду запуска приложения и состоит из следующих элементов:

- ОС Astra Linux;
- среда выполнения приложения Docker с возможностью запуска контейнеров приложений в виртуальной среде;
- СУБД Postgres Pro Ent.

1.2 Docker-образы компонентов

Docker-образы компонентов представляют собой основу для запуска виртуальной среды конкретного компонента в виде контейнера. Состав образов в рамках ПО СКАМ следующий:

- skam-service – серверный компонент обработки, хранения СВИ и расчета параметров СК;
- skam-ui – пользовательский интерфейс приложения.

1.3 Конфигурации

Конфигурации описывают связи между компонентами, необходимые им ресурсы и связь с внешними ресурсами. Поставляются в виде отдельного архива и при необходимости подлежат редактированию. Подробнее состав и назначение каждого редактируемого параметра в конфигурационных файлах описан в разделе установки ПО СКАМ.

Состав файлов следующий:

- skam-service.config - конфигурационный файл настроек серверной части приложения;
- env-specific.json - конфигурационный файл веб-приложения;

– nginx.conf - конфигурационный файл nginx веб-приложения для настройки интеграции серверной части и веб-приложения, а также ssl-сертификатов.

2 Системные требования

Для функционирования ПО СКАМ требуется один сервер (виртуальный или физический), соответствующий следующим характеристикам:

- 16х ядерный CPU с архитектурой x86-64;
- 32 Гб RAM;
- 1 Тб дискового пространства.

3 Установка системного ПО

Необходимо установить следующее системное ПО, необходимое для функционирования системы:

– Astra Linux "Орел" версии 2.12 и выше;

– Docker версии не ниже 19.03.

3.1 Установка Astra Linux

Установка описана в официальной документации, расположенной на сайте производителя – https://astralinux.ru/assets/docs/AstraLinuxCE_install_2-12.pdf.

3.2 Установка Docker

Установка docker для операционной системы Astra Linux ничем не отличается от установки на ОС Debian, описанной по адресу https://docs.docker.com/engine/install/debian/ .

Для этого надо установить (если не установлено) ПО для интеграции с https-репозиториями для пакетного менеджера apt (все последующие команды - команды командной строки bash):

```
sudo apt-get update
```

```
sudo apt-get install \
apt-transport-https \
ca-certificates \
curl \
gnupg-agent \
software-properties-common
```

Далее нужно установить официальный PGP ключ репозитория производителя docker:

```
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add
```

Далее добавьте репозиторий docker:

```
sudo add-apt-repository \
 "deb [arch=amd64] https://download.docker.com/linux/debian \
 (lsb_release -cs) \
 stable"
```

После чего произведите обновление доступных пакетов и осуществите их установку:

```
 sudo apt-get update
 sudo apt-get install docker-ce docker-ce-cli containerd.io
```

Проверка корректной установки docker осуществляется запуском тестового контейнера:

```
sudo docker run hello-world
```

После его запуска будет отображено сообщение «hello world» в командной строке и контейнер завершит свою работу. Это значит, что docker установлен успешно.

4 Установка образов приложения (docker)

Далее необходимо загрузить образы приложения в репозиторий Docker:

```
sudo docker load -i skam-service.tar

sudo docker load -i skam-ui.tar
```

5 Установка и настройка СУБД

Установите СУБД PostgresPro в соответствии с инструкцией по установке, предоставленной производителем данного ПО: https://postgrespro.ru/docs/enterprise/13/binary-installation-on-linux.

Минимальными требованиями для подключения СКАМ к данной СУБД являются:

1. в postgresql.conf (как правило, расположение следующеее: /etc/postgresql/<version>/main/postgresql.conf) выполните настройку переменных:
    a. data_directory установить на ‘mnt/datastorage/skam-share-storage/postgres’
    b. listen_addresses = '0.0.0.0'
2. в pg_hba.conf указать с каких адресов возможно подключение, а именно для всех ip-адресов СКАМ.
3. Ограничить память сервиса до 2 Гб.
4. В файле skam-service.config в параметре data-source.url обновите ip-адрес до СУБД PostgresPro. В случае действующего СКАМ дополнительно требуется обновить контейнер skam-service для применения изменений в конфигурационном файле.
5. Требуется наличие созданной базы данных postgres.

6 Настройка приложения

Для запуска загруженных образов приложений предварительно необходимо создать сеть внутри docker, через которую будут общаться компоненты приложения между собой и с внешней по отношению к docker сетью. Сеть с наименованием skam-network (далее будет использоваться в привязке к запускаемым контейнерам) создается следующей командой:

```
docker network create skam-network
```

Далее создаем контейнер для сервиса расчетов СКАМ с запуском в режиме daemon (-d) и опцией автозапуска (--restart unless-stopped), привязкой к skam-network, dns-именем skam-service внутри skam-network, а также монтируем конфигурацию приложения внутрь контейнера из файловой системы сервера по пути /mnt/datastorage/skam-share-storage/skam-service.config.

Перед созданием необходимо установить нужную timezone: изменить путь до файла нужной timezone на хосте (/usr/share/zoneinfo/**Europe/Moscow**) в аргументе --mount type=bind, source=/usr/share/zoneinfo/**Europe/Moscow**, target=/etc/localtime.

```
sudo mkdir -p /mnt/datastorage/skam-share-storage/logs/skam-service


docker run -d --restart unless-stopped  \
--net skam-network \
--name skam-service  \
--env JAVA_OPTS="-Xmx2048m -XX:+UseG1GC -XX:+HeapDumpOnOutOfMemoryError -
XX:HeapDumpPath=/mnt" \
--mount type=bind,source=/mnt/datastorage/skam-share-storage/skam-
service.config,target=/etc/opt/ch/application.properties,readonly \
--mount type=bind,source=/mnt/datastorage/skam-share-
storage/kerberos.keytab,target=/etc/opt/kerberos.keytab,readonly \
--mount type=bind,source=/mnt/datastorage/skam-share-storage/logs/skam-
service,target=/usr/local/tomcat/logs \
--mount type=bind,source=/usr/share/zoneinfo/Europe/Moscow,target=/etc/localtime
\
--publish 'port':8080 \
ap/skam-service:0.1
```

Создаем контейнер веб-сервера с пользовательским интерфейсом:

```
docker run -d --restart unless-stopped  \
--net skam-network \
--name skam-ui \
--mount type=bind,source=/mnt/datastorage/skam-share-storage/env-
specific.json,target=/usr/share/nginx/html/assets/env-specific.json,readonly \
--mount type=bind,source=/mnt/datastorage/skam-share-
storage/nginx.conf,target=/etc/nginx/nginx.conf,readonly \
--mount type=bind,source=/mnt/datastorage/skam-share-
storage/tls.key,target=/usr/share/nginx/ssl/cert.key,readonly \
--mount type=bind,source=/mnt/datastorage/skam-share-
```

```
storage/tls.crt,target=/usr/share/nginx/ssl/cert.pem,readonly \
--publish 'port':80 \
--publish 443:443 \
ap/skam-ui:0.1
```

где привязываем также путь до ssl-сертификата с приватным ключом (/mnt/datastorage/skam-share-storage/tls.crt и mnt/datastorage/skam-share-storage/tls.key) и конфигурацию nginx, сконфигурированную под использование этих ключей, а также создающую reverse proxy на сервис через единый https-порт.

В файле nginx.conf необходимо задать dns-имя для веб-сервера - присвоить ключу server_name значение dns-имени.

Все файлы конфигураций (а также файлы с ключами для ssl и kerberos) самого приложения должны храниться на сервере в директории /mnt/datastorage/skam-share-storage.

В СКАМ в части настроек приложения присутствуют следующие файлы:

- skam-service.config - конфигурационный файл настроек серверной части приложения;
- env-specific.json - конфигурационный файл веб-приложения;
- kerberos.keytab - перманентный kerberos-токен для сервиса с SPN для веб-сервера;
- tls.key/tls.crt - закрытый и открытый ключи ssl-сертификата;
- nginx.conf - конфигурационный файл nginx.

6.1 env-specific.json

В файле env-specific.json требуется указать следующие настройки:

(конструкции в виде "{{ SKAM_SERVER_DNS_NAME }}" являются переменными и подлежат замене (вместе со скобками {{}}))

```
    {
 "backendhost": "https://{{ SKAM_SERVER_DNS_NAME }}/backend",
 "mapurl": "https://{{ SKAM_SERVER_DNS_NAME }}/map/map-
server/images/osm_tiles/{z}/{x}/{y}.png",
 "oauth2": {
    "endpoint": "https://{{ SKAM_SERVER_DNS_NAME }}/backend/oauth",
    "clientId": "***",
    "clientSecret": "***",
    "redirectUri": "https://{{ SKAM_SERVER_DNS_NAME }}/monitoring",
    "authType": "ldap"
 }
}
```

Где:

– «backendhost» - внешний адрес к backend REST API. Необходимо явное указание внешнего доступа, т.к. он в последующем отправляется пользователю как ресурс клиентского приложения.
– «oauth2:endpoint» - адрес к API выдачи токенов для аутентификации.
– «oauth2:redirectUri» - адрес куда будет перенаправлен пользователь после аутентификации.
– «oauth2:authType» - тип аутентификации.

6.2 skam-service.config

В файле skam-service.config задаются настройки доступа к АС СИ СМПР и ОИК СК-11, почте, а также авторизации:

– rest-topology-provider.auth.client-id – имя сервисной УЗ;
– rest-topology-provider.auth.secret-id – пароль сервисной УЗ;
– thrift-telemetry-value-provider.host – адрес thrift api АС СИ СМПР;
– thrift-telemetry-value-provider.port – порт thrift api АС СИ СМПР;
– thrift-telemetry-value-provider.username - имя пользователя для доступа к АС СИ СМПР;
– thrift-telemetry-value-provider.password – пароль;
– data-quality-configuration.ldap.userDn=*** УЗ преднастроенного администратора в системе для проверки присутствия пользователей в службе каталогов;
– data-quality-configuration.ldap.password=123123 пароль преднастроенного администратора;
– data-quality-configuration.mail.sender – почтовый ящик отправителя рассылаемых отчетов;
– spring.mail.host – хост почтового сервера;
– spring.mail.port – порт почтового сервера;
– spring.mail.username – пользователь, под которым происходит авторизация и отправка почты;
– spring.mail.password – пароль пользователя.

Пароль для {{ SKAM_SERVICE_USER_PASSWORD }} не должен содержать спец. символы "**!*'();:@&=+$,/?#[]**".

Файл конфигурации чувствителен к пробелам в конце строки.

6.3 kerberos.keytab

Для аутентификации клиентов посредством схемы Negotiate для HTTP с поддержкой Kerberos необходимо выпустить ketyab файл.

Конструкции в виде «{{ SERVER_DNS_NAME }}» являются переменными и подлежат замене (вместе со скобками {{}}):

- если имя переменной указано как CamelCase - значение регистрозависимое;
- если имя переменной указано как UPPERCASE - значение должно быть в верхнем регистре;
- если имя переменной указано как lower_case - значение должно быть в нижнем регистре.

Открыть cmd от имени администратора службы каталогов. Сгенерировать kerberos.keytab файл для заданного spn:

```
ktpass -princ HTTP/{{ skam_server_dns_name_with_domain }}@{{ DOMAIN }} -pass {{
password }} -mapuser {{ domain }}\{{ Username }} -crypto ALL -ptype
KRB5_NT_PRINCIPAL -out kerberos.keytab
```

Полученный keytab-файл нужно перенести на сервер приложений по следующему пути:

```
/mnt/datastorage/skam-share-storage/kerberos.keytab
```

Изменить следующие настройки в конфигурационном файле skam-service.config:

```
data-quality-configuration.kerberos.service-principal-name=HTTP/{{
skam_server_dns_name_with_domain            }}@{{             DOMAIN            }}
data-quality-configuration.kerberos.keytab-location=/etc/opt/kerberos.keytab
data-quality-configuration.jwt.redirect-uri=https://{{
skam_server_dns_name_with_domain }}/monitoring
```

Где:

- data-quality-configuration.kerberos.service-principal-name - назначенный SPN. Обычно имеет вид HTTP/{{ skam_server_dns_name_with_domain }}@{{ DOMAIN }};
- data-quality-configuration.kerberos.keytab-location - полный путь до .keytab файла внутри контейнера (/etc/opt/kerberos.keytab);
- data-quality-configuration.jwt.redirect-uri - адрес на который вернёт backend после успешной аутентификации.

Изменить следующие настройки (не стирая предыдущие для oauth2) в конфигурационном файле env-specific.json:

```
"oauth2": {
   ...
   "redirectUri": "https://{{skam_server_dns_name_with_domain}}/monitoring",
   "authType": "kerberos"}
```

Где:

- «oauth2:authType» - тип аутентификации;
- «oauth2:redirectUri» - адрес, на который вернёт backend после успешной аутентификации.

Далее необходимо перезагрузить контейнеры skam-service и skam-ui.

6.4 nginx.conf

В файле nginx.conf конструкции в виде «{{ SERVER_DNS_NAME }}» являются переменными и подлежат замене.

6.5 tls.key и tls.crt

Для работы протокола https веб-сервер должен быть настроен на использование приватного и публичного ssl-ключей.

Для этого предварительно необходимо сгенерировать ssl-сертификат.

Если сертификат сгенерирован в формате .pfx необходимо получить из сертификата файлы tls.key (приватный ключ) и tls.crt (открытый сертификат) через openssl, для этого необходимо выполнить следующие команды:

```
openssl pkcs12 -in SERVER_DNS_NAME.pfx -nocerts -nodes -out tls.key
openssl pkcs12 -in tls.pfx -clcerts -nokeys -out tls.crt
```

Данные файлы следует поместить в директорию mnt/datastorage/skam-share-storage сервера.

7 Проверка корректности установки приложения

Проверка состоит из двух этапов – статуса каждого из контейнеров приложения, а также доступности веб-интерфейса.

Для проверки статусов каждого из контейнеров приложения необходимо выполнить команду:

```
sudo docker ps
```

и проверить, что все контейнеры в пространстве skam запущены.

Для проверки веб-интерфейса необходимо зайти по адресу https://<dns-имя СКАМ> и проверить доступность веб-интерфейса.

Также при выборе конкретного контейнера можно открыть логи и увидеть факт успешности или неуспешности запуска приложения (в случае отсутствия записей с началом в строке ERROR запуск можно считать удачным).