

МКООО «ГАЗПРОМ ИНТЕРНЭШНЛ ЛИМИТЕД»

**СИСТЕМА МОНИТОРИНГА
ПРОИЗВОДСТВЕННЫХ ОБЪЕКТОВ,
МОДЕРНИЗИРОВАННАЯ
МКООО «ГАЗПРОМ ИНТЕРНЭШНЛ ЛИМИТЕД»
(СМПО М)**

Инструкция по установке

2023

СОДЕРЖАНИЕ

1 ОБЩИЕ ПОЛОЖЕНИЯ.....	4
2 ВВЕДЕНИЕ	5
2.1 Область применения	5
2.2 Функциональная архитектура системы	5
2.3 Перечень объектов автоматизации, на которых используется система	6
2.4 Требования к численности и квалификации обслуживающего персонала	6
3 ПОДГОТОВКА К РАБОТЕ	7
3.1 Состав системы.....	7
3.2 Программные и аппаратные требования к системе	7
3.3 Подготовка к развертыванию подсистем.....	7
3.4 Порядок развертывания подсистем	7
4 УСТАНОВКА И НАСТРОЙКА СИСТЕМЫ.....	9
4.1 Установка и настройка РЕД ОС.....	9
4.2 Установка и настройка СУБД PostgreSQL	9
4.3 Установка и настройка сервера доступа, контроля и обработки данных (микросервисов) 10	
4.3.1 Настройка файла Hosts	10
4.3.2 Установка и настройка сервера «NGINX».....	11
4.3.3 Установка и настройка Java JDK	13
4.3.4 Установка и настройка Redis	15
4.3.5 Установка и настройка Apache Kafka.....	18
4.3.6 Установка и настройка Apache Solr.....	25
4.3.7 Установка и настройка микросервисов.....	28
4.3.8 Установка и настройка дополнительных микросервисов (через Docker)	32
4.3.9 Проверка работоспособности сервера доступа, контроля и обработки данных (микросервисов)	32
4.3.10 Схема резервирования сервера доступа, контроля и обработки данных (микросервисов)	34
4.3.11 Схема восстановления сервера доступа, контроля и обработки данных (микросервисов)	34
4.4 Установка и настройка сервера бизнес-аналитики Apache Superset.....	36
4.4.1 Настройка файла Hosts	36
4.4.2 Установка и настройка сервера «NGINX».....	37
4.4.3 Установка Docker	40
4.4.4 Установка Docker-compose.....	42

4.4.5	Установка Apache Superset	44
4.4.6	Проверка работоспособности сервера бизнес-аналитики Apache Superset	45
4.4.7	Импорт (загрузка) на сервер бизнес-аналитики Apache Superset шаблонов дашбордов из состава дистрибутива системы.....	46
4.4.8	Настройка связи сервера бизнес-аналитики Apache Superset с сервером СУБД PostgreSQL	48
5	ПОРЯДОК ЗАПУСКА, МОНИТОРИНГА И ОСТАНОВКИ СИСТЕМНОГО И ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	51
5.1	Порядок запуска системного и прикладного программного обеспечения.....	51
5.2	Порядок мониторинга системного и прикладного программного обеспечения.....	52
5.3	Порядок остановки системного и прикладного программного обеспечения	52
	Приложение А1 Содержимое файлов конфигурации.....	53
A1.1	/var/lib/postgresql/12/main/postgresql.conf.....	53
A1.2	/var/lib/postgresql/12/main/pg_hba.conf.....	53
A1.3	/etc/corosync/corosync.conf.....	54
	ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ.....	55

1 ОБЩИЕ ПОЛОЖЕНИЯ

Данная инструкция содержит описание действий по развертыванию Системы мониторинга производственных объектов МКООО «ГАЗПРОМ ИНТЕРНЭШНЛ ЛИМИТЕД».

2 ВВЕДЕНИЕ

2.1 Область применения

«СМПО М» предназначена для автоматизации централизованного сбора, обработки, хранения, консолидации и анализа производственных данных и технико-экономических показателей производственных процессов нефтегазовых проектов по направлениям: поиск, разведка, разработка и добыча углеводородов, а также по направлению выполнения нефтесервисных контрактов.

2.2 Функциональная архитектура системы

Система «СМПО М» состоит из следующих системотехнических подсистем:

- подсистемы администрирования (ПА);
- подсистемы информационной безопасности (ПИБ);
- подсистемы нормативно-справочной информации (ПНСИ);
- подсистемы настройки системы (ПНС);
- подсистемы подготовки и ввода данных (ППВД);
- подсистемы сбора и интеграции данных (ПСИД);
- подсистемы хранения данных (ПХД);
- подсистемы регламентной отчетности (ПРО);
- аналитической подсистемы (АП).

Подсистема ПА предназначена для работы с учетными данными пользователей «СМПО М», управление ролями, правами доступа и настройками системы.

Подсистема ПИБ предназначена для обеспечения целостности и конфиденциальности данных «СМПО М» при осуществлении сбора, долговременного хранения и предоставлении доступа к этим данным подсистемам и пользователям «СМПО М».

Подсистема ПНСИ предназначена для обеспечения единой технологии синхронизации, актуализации и доступа к данным справочников и классификаторов «СМПО М».

Подсистема ПНС предназначена для гибкой настройки ряда функциональных возможностей «СМПО М» без привлечения программистов. Подсистема ПНС позволяет настраивать и изменять функциональные возможности «СМПО М» в части сбора и обработки данных.

Подсистема ППВД предназначена для ввода, консолидации, и первичной обработки производственных данных.

Подсистема ПСИД предназначена для автоматизированного формирования реестра входных форм данных, полученных с производственных объектов «СМПО М», автоматизированной обработки и сохранения этих данных, мониторинга данного процесса сбора и интеграции данных «СМПО М».

Подсистема ПХД предназначена для надежного оперативного и долговременного (не менее 10 лет) хранения данных «СМПО М» с использованием системы управления базами данных (СУБД).

Подсистема ПРО предназначена для настройки, формирования и рассылки регламентных отчётов по утверждённому списку.

Аналитическая АП предназначена для формирования и рассылки аналитических отчетов. Подсистема АП обеспечивает возможность по заранее определённым критериям (объект, диапазон дат, диапазон глубин и т.п.) выбрать список показателей и отобразить их в виде таблиц, диаграмм, графиков.

Функциональные возможности «СМПО М» позволяют на базе перечисленных системотехнических подсистем, за счет настройки Системы, за счет расширения структур данных и документов Системы, настраивать работу различного рода прикладных функциональных подсистем.

Система «СМПО М» на текущем этапе состоит из следующих прикладных функциональных подсистем:

- подсистемы учета производственных объектов (ПУО);
- подсистемы планирования (ПП);
- подсистемы производственного контроля (ППК).

2.3 Перечень объектов автоматизации, на которых используется система

Подсистемы «СМПО М» установлены и эксплуатируются на следующих объектах МКООО «ГАЗПРОМ ИНТЕРНЭШНЛ ЛИМИТЕД»:

- в подразделениях центрального уровня:
 - 1) в корпоративном сервисном центре (КСЦ);
 - 2) в профильных функциональных подразделениях;
- на уровне производственных объектов.

2.4 Требования к численности и квалификации обслуживающего персонала

Численность инженерного обслуживающего персонала (системных администраторов) должна обеспечивать круглосуточное поддержание системы в рабочем состоянии. Особых требований к численности персонала не предъявляется.

3 ПОДГОТОВКА К РАБОТЕ

3.1 Состав системы

Описание состава «СМПО М» приведено в документе «Система мониторинга производственных объектов МКООО «ГАЗПРОМ ИНТЕРНЭШНЛ ЛИМИТЕД» (СМПО М). Общее описание системы»

3.2 Программные и аппаратные требования к системе

Требования и спецификация комплекса технических средств и программного обеспечения «СМПО М» представлена в документе «Система мониторинга производственных объектов МКООО «ГАЗПРОМ ИНТЕРНЭШНЛ ЛИМИТЕД» (СМПО М). Спецификация».

Описание комплекса технических средств «СМПО М» представлено в документе «Система мониторинга производственных объектов МКООО «ГАЗПРОМ ИНТЕРНЭШНЛ ЛИМИТЕД» (СМПО М). Описание комплекса технических средств».

Описание программного обеспечения «СМПО М» представлено в документе «Система мониторинга производственных объектов МКООО «ГАЗПРОМ ИНТЕРНЭШНЛ ЛИМИТЕД» (СМПО М).

3.3 Подготовка к развертыванию подсистем

Перед развертыванием «СМПО М» необходимо выполнить ряд следующих действий:

- подготовить полный комплект дистрибутивов программного обеспечения необходимого для развертывания «СМПО М»;
- организовать файловый архив для хранения резервных копий дистрибутивов программных компонентов системы и конфигурационных файлов;
- подготовить списки пользователей «СМПО М» в подразделениях центрального уровня и на производственных объектах;
- изучить эксплуатационную документацию.

3.4 Порядок развертывания подсистем

Порядок развертывания подсистем «СМПО М» представлен в таблице 1.

Таблица 1 – Порядок развёртывания подсистем

№ п.п.	Наименование программного обеспечения
1	2
1. Сервер реляционной СУБД (основной и резервные) (S-03,S-04,S-05)	
1.1	Операционная система Ред ОС/Astra Linux
1.2	СУБД PostgreSQL/PostgresPro
2. Сервер доступа, контроля и обработки данных (микросервисов) (основной и резервный) (S-01,S-02)	
2.1	Операционная система Ред ОС/Astra Linux
2.2	Микросервисы Платформы «САКУРА PRO» (СМПО М)
3. Сервер бизнес-аналитики (S-06)	
3.1	Операционная система Ред ОС/Astra Linux
3.2	Сервер бизнес-аналитики

Общий порядок установки системного и прикладного программного обеспечения отдельных компонент системы приведен в разделе 4 «УСТАНОВКА И НАСТРОЙКА СИСТЕМЫ».

4 УСТАНОВКА И НАСТРОЙКА СИСТЕМЫ

4.1 Установка и настройка РЕД ОС

Установка РЕД ОС осуществляется на серверах доступа, контроля и обработки данных (микросервисов) (S-01,S-02), на серверах СУБД (S-03,S-04,S-05) и на сервере бизнес-аналитики (S-06).

4.2 Установка и настройка СУБД PostgreSQL

Установка СУБД PostgreSQL осуществляется на серверах СУБД (S-03,S-04,S-05). Если сервера СУБД PostgreSQL резервируются в «холодном» режиме, то СУБД PostgreSQL устанавливается на каждый сервер, то необходимо воспользоваться инструкцией данного раздела.

Для развертывания первичного дампа базы данных с базовыми настройками системы (для случая установки непосредственно на сервере БД) необходимо выполнить команду (пример выполнения команды см. Рисунок 111):

```
pg_restore -h localhost -U postgres -w -v -d {%DBNAME%} {%DBFILE%}
```

, где:

{%DBNAME%} – наименование базы данных;
{%DBFILE%} – полный путь к файлу дампа БД.

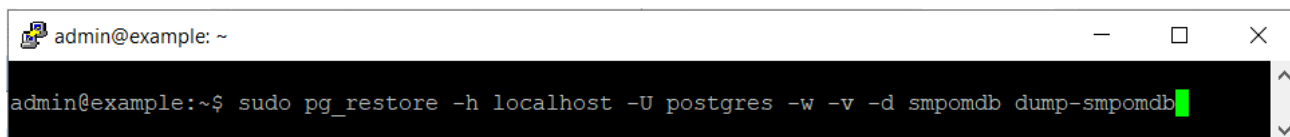


Рисунок 11 – Пример выполнения команды по развертыванию дампа БД СМПО М

Для развертывания первичного дампа базы данных с базовыми настройками системы (для случая установки удаленно с ПЭВМ администратора БД под управлением РЕД ОС) необходимо выполнить команду (используется программный комплекс «pgAdmin»):

```
C:\Program Files\pgAdmin\runtime\pg_restore.exe" --host "{%DBSERVERIP%}" --port  
"5432" --username "postgres" --no-password --dbname "{%DBNAME%}" --verbose  
"{%DBFILE%}"
```

, где:

{%DBSERVERIP%} – IP адрес сервера;
{%DBNAME%} – наименование базы данных;
{%DBFILE%} – полный путь к файлу дампа БД.

4.3 Установка и настройка сервера доступа, контроля и обработки данных (микросервисов)

Установка Сервера доступа, контроля и обработки данных осуществляется на серверах доступа, контроля и обработки данных (микросервисов) (S-01 – основной ,S-02 – резервный, в режиме «холодного резерва»).

4.3.1 Настройка файла Hosts

1. Найти и открыть для корректировки файл «hosts» (пример выполнения команды см. Рисунок):

```
sudo nano /etc/hosts
```



Рисунок 2 – Пример выполнения команды открытия для корректировки файла «hosts»

2. Дописать в файл «hosts» следующие строки (пример выполнения команды см. Рисунок 3):

```
127.0.0.1 auth-service
127.0.0.1 eureka
127.0.0.1 config
127.0.0.1 redisvc
127.0.0.1 kafkasrv
127.0.0.1 solrsrv

{%SERVERDBIP%} {%SERVERDBNAME%} # Сервер базы данных
{%SERVERMSIP%} {%SERVERMSNAME%} # Сервер доступа, контроля и обработки данных
{%SERVERBIIP%} {%SERVERBINAME%} # Сервер аналитической подсистемы
```

, где:

{%SERVERDBIP%} и {%SERVERDBNAME%} – IP адрес и имя Сервера базы данных,
{%SERVERMSIP%} и {%SERVERMSNAME%} – IP адрес и имя Сервера доступа, контроля и
обработки данных (микросервисов),
{%SERVERBIIP%} и {%SERVERBINAME%} – IP адрес и имя Сервера аналитической
подсистемы.

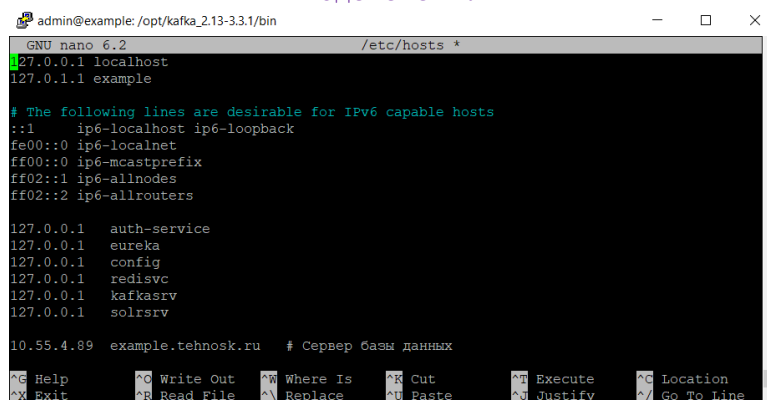


Рисунок 3 – Пример корректировки файла «hosts»

4.3.2 Установка и настройка сервера «NGINX»

1. Обновить списки пакетов операционной системы из репозиторий:

```
sudo apt update
```

Установить Nginx:

```
sudo apt install nginx
```

2. Добавить программу Nginx в автозагрузку (пример выполнения команды см. Рисунок 4):

```
sudo systemctl enable nginx
```

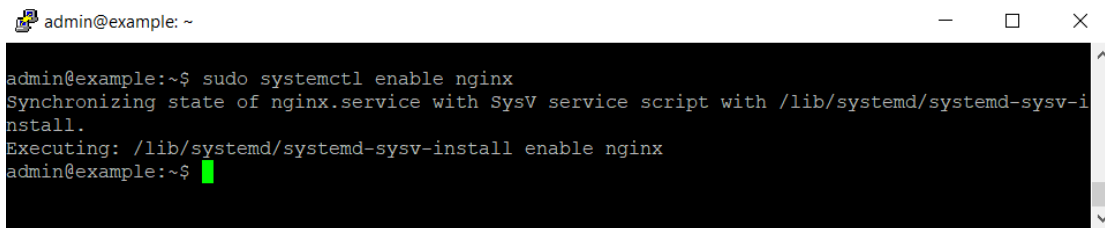


Рисунок 4 – Пример выполнения команды добавления программы Nginx в автозагрузку

3. Выложить (заранее подготовленные) ключи (full.crt и key-decrypted.key) для работы протокола «HTTPS» в один из каталогов сервера (пример размещения файлов см. Рисунок 5).

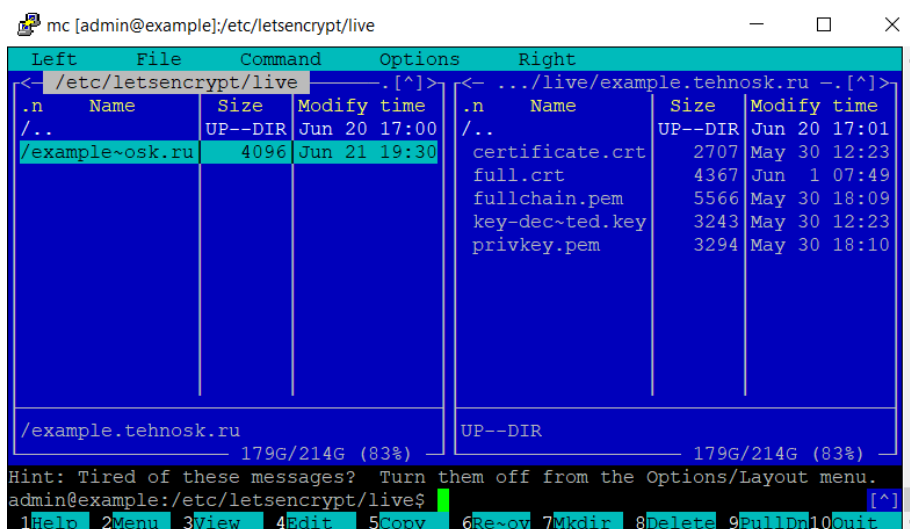


Рисунок 5 – Пример размещения файлов – ключей на сервере

4. Перейти в каталог «conf.d» по пути /etc/nginx/conf.d (пример выполнения команды см. Рисунок 6):

```
cd /etc/nginx/conf.d
```



5. Создать в каталоге «/etc/nginx/conf.d» файл конфигурации «nginx_smpom.conf» и в нем указать актуальное имя сервера и пути к ключам для работы протокола «HTTPS» в соответствии с представленным ниже примером (пример корректировки файла см. Рисунок 7 и Рисунок 8):

```
server {
    server_name {%SERVERNAME%}; # Имя сервера в формате example.tehnosk.ru;

    access_log /var/log/nginx/test_access.log;
    error_log /var/log/nginx/test_error.log;

    location / {
        proxy_pass http://127.0.0.1:7000/;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-for $remote_addr;
        port_in_redirect off;
        proxy_connect_timeout 6000;
        client_max_body_size 100M;
    }

    listen 443 ssl http2; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/{%SERVERNAME%}/fullchain.pem; # managed by
Certbot # Указать путь до сертификатов
    ssl_certificate_key /etc/letsencrypt/live/{%SERVERNAME%}/privkey.pem; # managed
by Certbot# Указать путь до ключа
    include /etc/letsencrypt/options-ssl-nginx.conf;
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

    gzip on;
    gzip_vary on;
    gzip_min_length 1024;
    gzip_proxied expired no-cache no-store private auth;
    gzip_types text/plain text/css text/xml text/javascript application/x-javascript
application/xml application/json;
    gzip_disable "MSIE [1-6]\.";
}
server {
    if ($host = {%SERVERNAME%}) { # Имя сервера в формате example.tehnosk.ru
        return 301 https://$host$request_uri;
    } # managed by Certbot

    server_name {%SERVERNAME%}; # Имя сервера в формате example.tehnosk.ru
    listen 80;
    return 404; # managed by Certbot
}
```

, где: {%SERVERNAME%} – Доменное имя сервера.

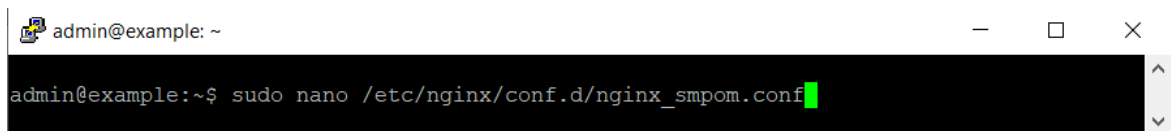


Рисунок 7 – Пример выполнения команды открытия файла «nginx_smpom.conf»

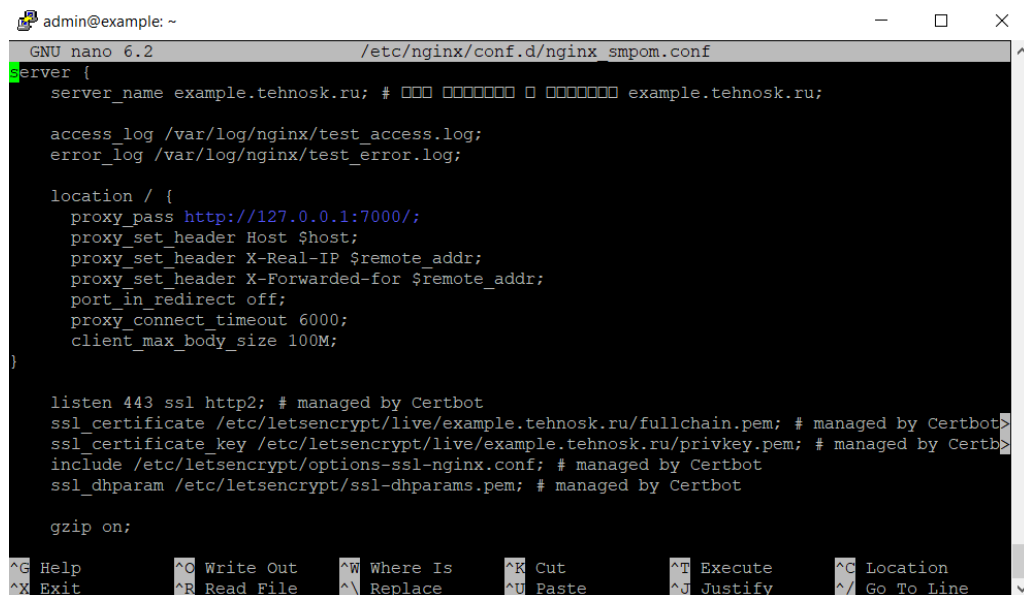


Рисунок 8 – Пример корректировки файла «nginx_smpom.conf»

6. Запустить Nginx (пример выполнения команды см. Рисунок 9):

```
sudo service nginx start
```



Рисунок 9 – Пример выполнения команды запуска программы Nginx

7. Проверить статус работы Nginx (пример выполнения команды см. Рисунок 112):

```
sudo service nginx status
```

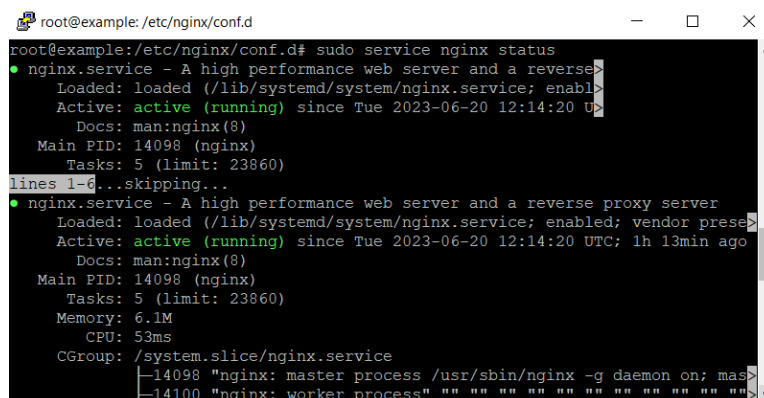


Рисунок 10 – Пример выполнения команды проверки статуса работы программы Nginx

4.3.3 Установка и настройка Java JDK

1. Установить утилиту «wget»:

```
apt install wget
```

2. Скачать дистрибутив с Java JDK 17 (пример выполнения команды см. Рисунок 11211):

```
wget https://download.oracle.com/java/17/archive/jdk-17.0.6_linux-x64_bin.tar.gz
```



```
admin@example: ~  
admin@example:~$ wget https://download.oracle.com/java/17/archive/jdk-17.0.6_linux-x64_bin.tar.gz  
--2023-06-21 06:12:46-- https://download.oracle.com/java/17/archive/jdk-17.0.6_linux-x64_bin.tar.gz  
Resolving download.oracle.com (download.oracle.com)... 2.18.32.49  
Connecting to download.oracle.com (download.oracle.com)|2.18.32.49|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 181577323 (173M) [application/x-gzip]  
Saving to: 'jdk-17.0.6_linux-x64_bin.tar.gz'  
  
jdk-17.0.6_linux-x64_bin. 100%[=====>] 173.17M  2.30MB/s   in 76s  
2023-06-21 06:14:02 (2.27 MB/s) - 'jdk-17.0.6_linux-x64_bin.tar.gz' saved [181577323/181577323]  
admin@example:~$
```

Рисунок 112 – Пример выполнения команды скачивания дистрибутива Java JDK

3. Перейти в каталог (каталога нет, то предварительно создать его), в который необходимо выполнить установку Java JDK (пример выполнения команды см. Рисунок 12):

```
cd /opt/java17/
```

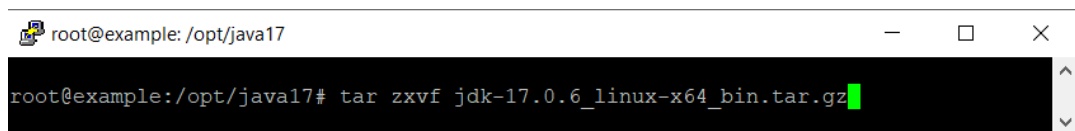


```
admin@example: /opt/java17  
admin@example:~$ cd /opt/java17/  
admin@example:/opt/java17$
```

Рисунок 12 – Пример выполнения команды перехода в каталог установки Java JDK

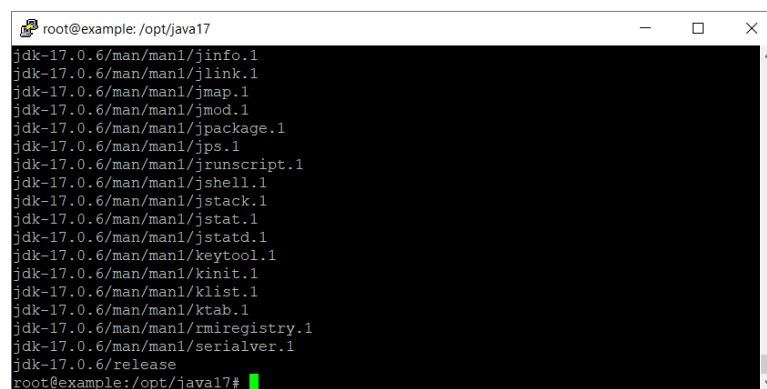
4. Переместить ранее дистрибутив Java JDK в текущий каталог и распаковать tar-архив (пример выполнения команды см. Рисунок 1313):

```
tar zxvf jdk-17.0.6_linux-x64_bin.tar.gz
```



```
root@example: /opt/java17  
root@example:/opt/java17# tar zxvf jdk-17.0.6_linux-x64_bin.tar.gz
```

Рисунок 13 – Пример выполнения команды распаковки архива Java JDK. Начало



```
root@example: /opt/java17  
jdk-17.0.6/man/man1/jinfo.1  
jdk-17.0.6/man/man1/jlink.1  
jdk-17.0.6/man/man1/jmap.1  
jdk-17.0.6/man/man1/jmod.1  
jdk-17.0.6/man/man1/jpackage.1  
jdk-17.0.6/man/man1/jps.1  
jdk-17.0.6/man/man1/jrunscript.1  
jdk-17.0.6/man/man1/jshell.1  
jdk-17.0.6/man/man1/jstack.1  
jdk-17.0.6/man/man1/jstat.1  
jdk-17.0.6/man/man1/jstatd.1  
jdk-17.0.6/man/man1/keytool.1  
jdk-17.0.6/man/man1/kinit.1  
jdk-17.0.6/man/man1/klist.1  
jdk-17.0.6/man/man1/ktab.1  
jdk-17.0.6/man/man1/rmiregistry.1  
jdk-17.0.6/man/man1/serialver.1  
jdk-17.0.6/release  
root@example:/opt/java17#
```

Рисунок 13 – Пример выполнения команды распаковки архива Java JDK. Окончание

Файлы Java Jdk устанавливаются в каталог jdk-17.0.6, созданный в текущем каталоге «/opt/java17/jdk-17.0.6».

5. Открыть и скорректировать файл с общесистемными переменными окружения (пример выполнения команды см. Рисунок):

```
sudo nano /etc/profile
```



Рисунок 15 – Пример выполнения команды открытия файла с переменными окружения

В файл «profile» необходимо добавить следующие строки (пример выполнения команды см. Рисунок 16):

```
JAVA_HOME=/opt/java17/jdk-17.0.6
PATH=$PATH:$HOME/bin:$JAVA_HOME/bin
export JAVA_HOME
export JRE_HOME
export PATH
```

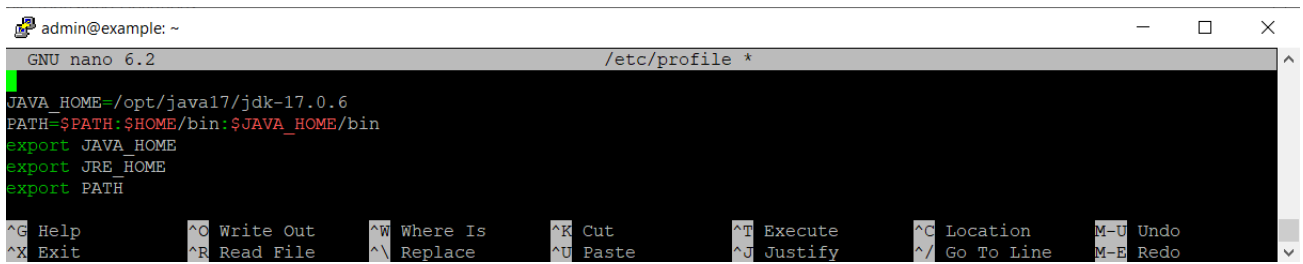


Рисунок 16 – Пример выполнения команды открытия файла с переменными окружения

6. Проверить работоспособность Java JDK на сервере (пример выполнения команды см. Рисунок 17):

```
java -version
```

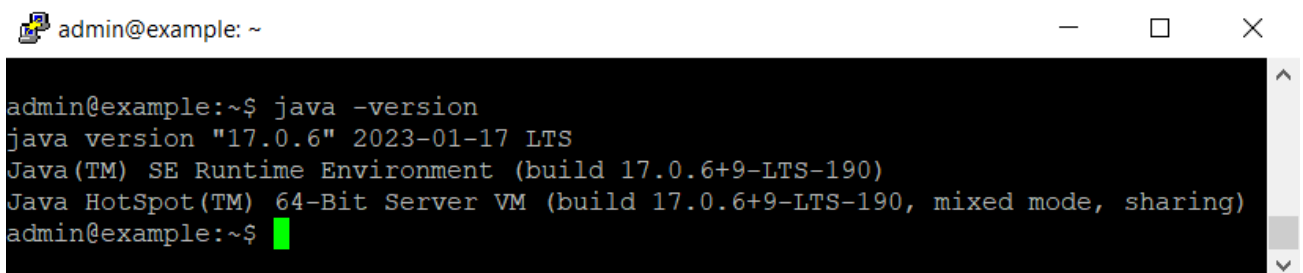


Рисунок 17 – Пример выполнения команды проверки работоспособности Java JDK

7. Удалить файл jdk-17.0.6_linux-x64_bin.tar.gz если необходимо сэкономить пространство на диске.

4.3.4 Установка и настройка Redis

1. Установить Redis:

```
sudo apt install redis-server
```

Открыть для редактирования файл «/etc/redis/redis.conf» (пример выполнения команды см. Рисунок):

```
sudo nano /etc/redis/redis.conf
```

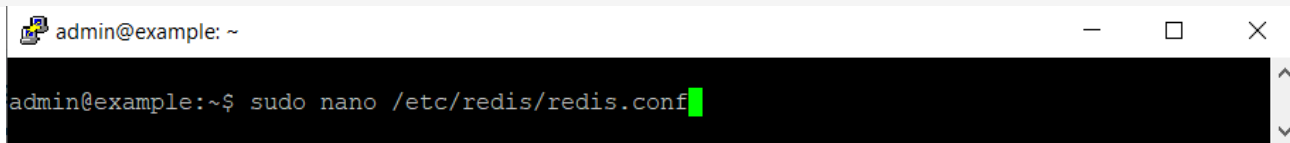


Рисунок 18 – Пример выполнения команды открытия для редактирования «redis.conf»

В файле «redis.conf» (в соответствии со структурой и правилами заполнения данного файла) необходимо (примеры файла см. Рисунок 194 - Рисунок 210):

```
# Найти параметр «bind» и указать в нем IP-адрес текущего сервера Redis
# для взаимодействия с Redis
Bind 127.0.0.1 {%SERVERREDISIP%}

# Найти параметр «port», проверить и при необходимости указать порт 6379
# для взаимодействия с Redis
Bind 127.0.0.1 {%SERVERREDISPORT%}

# Найти параметр «requirepass», раскомментировать его и указать в нем пароль
# для взаимодействия с Redis
requirepass {%SERVERREDISPASS%}
```

, где:

{%SERVERREDISIP%} – IP адрес Сервера, на котором установлен Redis,
{%SERVERREDISPORT%} – Порт для взаимодействия с Redis,
{%SERVERREDISPASS%} – Пароль для взаимодействия с Redis.

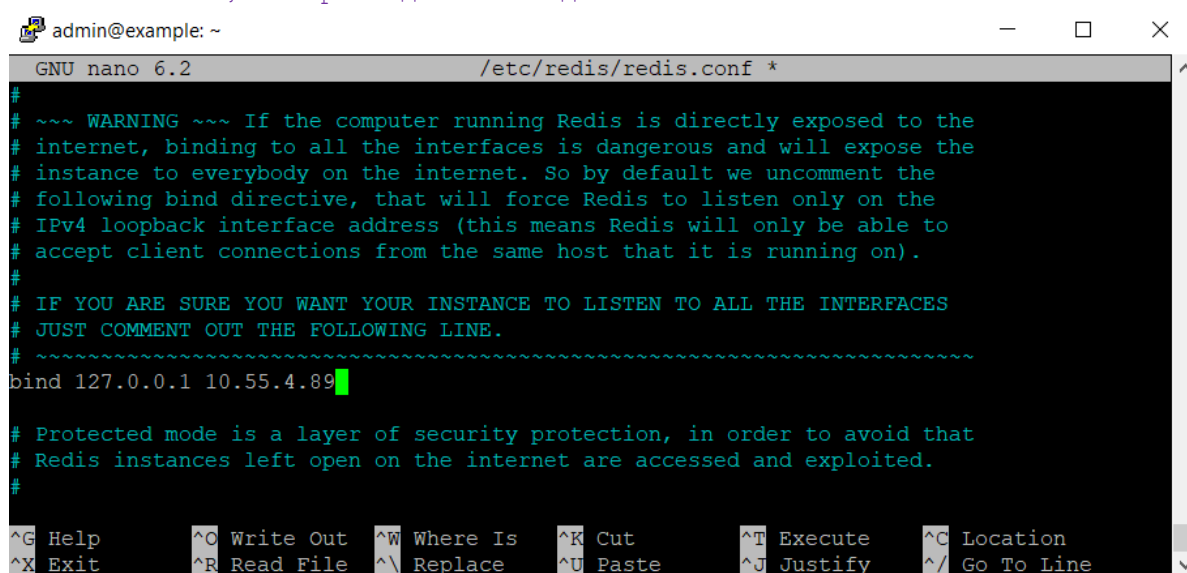
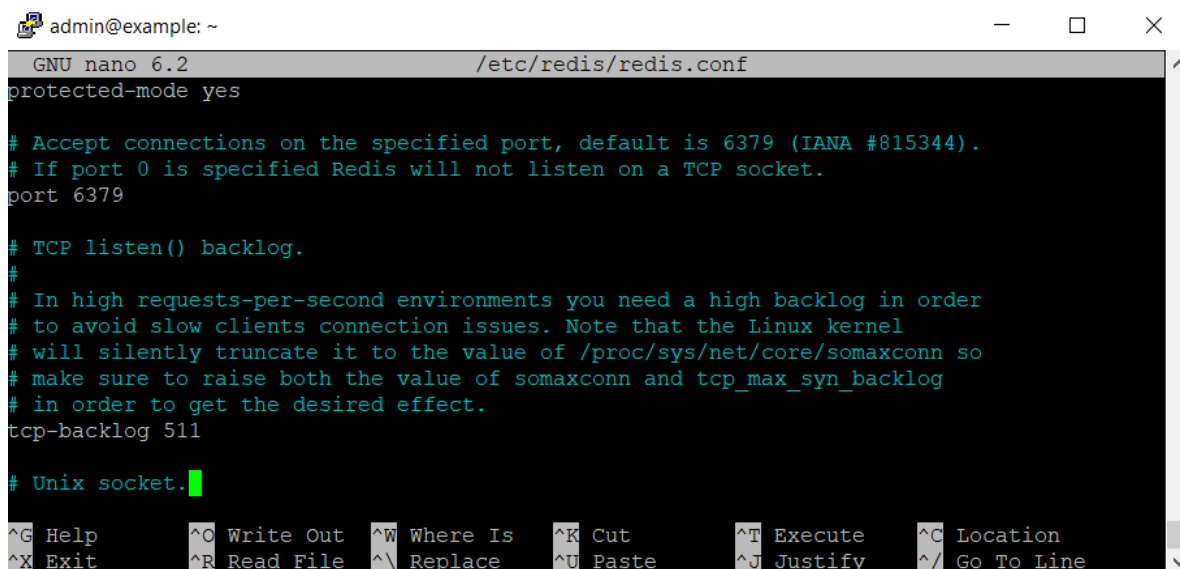


Рисунок 194 – Пример корректировки файла «redis.conf»



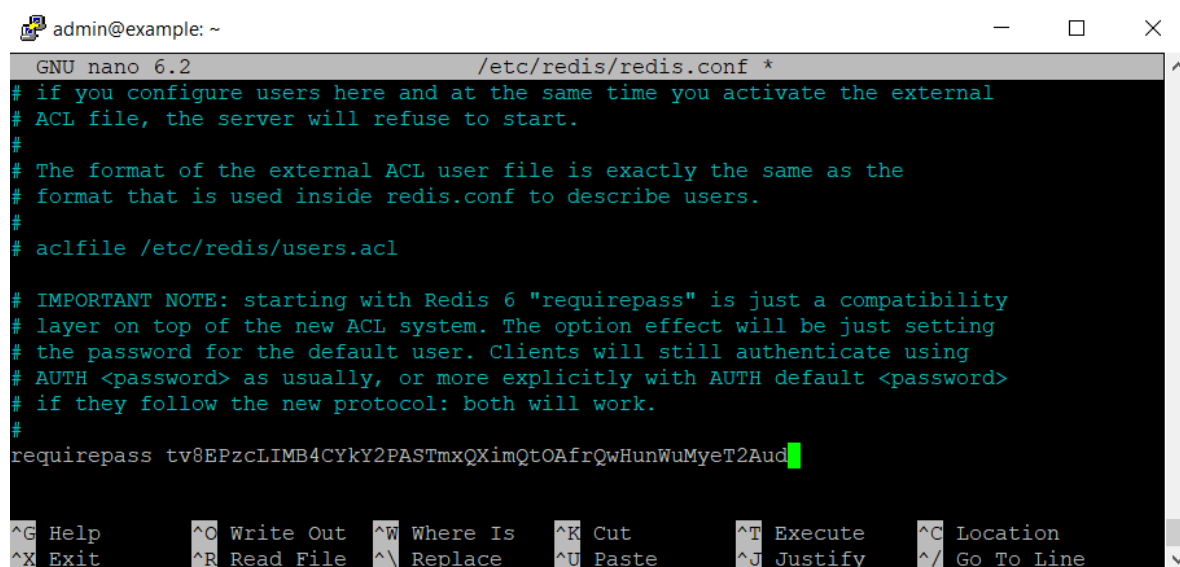
```
admin@example: ~
GNU nano 6.2 /etc/redis/redis.conf
protected-mode yes

# Accept connections on the specified port, default is 6379 (IANA #815344).
# If port 0 is specified Redis will not listen on a TCP socket.
port 6379

# TCP listen() backlog.
#
# In high requests-per-second environments you need a high backlog in order
# to avoid slow clients connection issues. Note that the Linux kernel
# will silently truncate it to the value of /proc/sys/net/core/somaxconn so
# make sure to raise both the value of somaxconn and tcp_max_syn_backlog
# in order to get the desired effect.
tcp-backlog 511

# Unix socket.
```

Рисунок 5 – Пример корректировки файла «redis.conf»



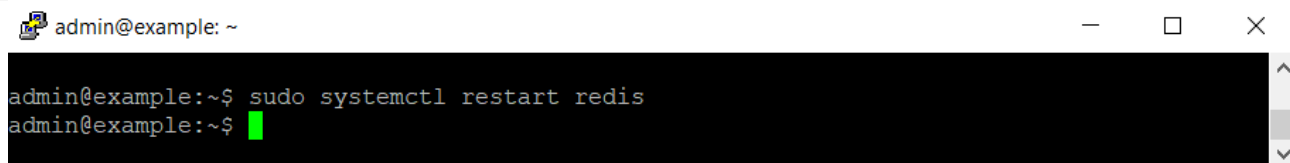
```
admin@example: ~
GNU nano 6.2 /etc/redis/redis.conf *
# if you configure users here and at the same time you activate the external
# ACL file, the server will refuse to start.
#
# The format of the external ACL user file is exactly the same as the
# format that is used inside redis.conf to describe users.
#
# aclfile /etc/redis/users.acl

# IMPORTANT NOTE: starting with Redis 6 "requirepass" is just a compatibility
# layer on top of the new ACL system. The option effect will be just setting
# the password for the default user. Clients will still authenticate using
# AUTH <password> as usually, or more explicitly with AUTH default <password>
# if they follow the new protocol: both will work.
#
requirepass tv8EPzcLIMB4CYkY2PASTmxQXimQtOAfrQwHunWuMyeT2Aud
```

Рисунок 21 – Пример корректировки файла «redis.conf»

2. Перезапустить Redis (пример выполнения команды см. Рисунок 22622):

```
sudo systemctl restart redis
```



```
admin@example: ~
admin@example:~$ sudo systemctl restart redis
admin@example:~$
```

Рисунок 226 – Пример выполнения команды перезапуска программы Redis

3. Проверить статус работы Redis (пример выполнения команды см. Рисунок 23723):

```
sudo systemctl status redis
```

```
admin@example: ~
admin@example:~$ sudo systemctl status redis
● redis-server.service - Advanced key-value store
   Loaded: loaded (/lib/systemd/system/redis-server.service; enabled; vendor preset: e>
lines 1-2...skipping...
● redis-server.service - Advanced key-value store
   Loaded: loaded (/lib/systemd/system/redis-server.service; enabled; vendor preset: e>
   Active: active (running) since Wed 2023-06-21 07:07:13 UTC; 2min 28s ago
     Docs: http://redis.io/documentation,
           man:redis-server(1)
  Main PID: 20332 (redis-server)
    Status: "Ready to accept connections"
     Tasks: 5 (limit: 23860)
  Memory: 2.6M
     CPU: 350ms
   CGroup: /system.slice/redis-server.service
           └─20332 "/usr/bin/redis-server 127.0.0.1:6379" "" "" "" "" "" "" "" "" "" "
```

Рисунок 237 – Пример выполнения команды проверки статуса работы программы Redis

4.3.5 Установка и настройка Apache Kafka

1. Скачать дистрибутив программы Kafka (пример выполнения команды см. Рисунок):

```
wget https://archive.apache.org/dist/kafka/3.3.1/kafka_2.13-3.3.1.tgz
```

```
admin@example: ~
admin@example:~$ wget https://archive.apache.org/dist/kafka/3.3.1/kafka_2.13-3.3.1.tgz
--2023-06-21 10:13:55-- https://archive.apache.org/dist/kafka/3.3.1/kafka_2.13-3.3.1.tgz
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 2a01:4f9:1a:a084::2
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 105053134 (100M) [application/x-gzip]
Saving to: 'kafka_2.13-3.3.1.tgz'

kafka_2.13-3.3.1.tgz  100%[=====>] 100.19M  82.5KB/s   in 19m 47s

2023-06-21 10:33:43 (86.4 KB/s) - 'kafka_2.13-3.3.1.tgz' saved [105053134/105053134]

admin@example:~$
```

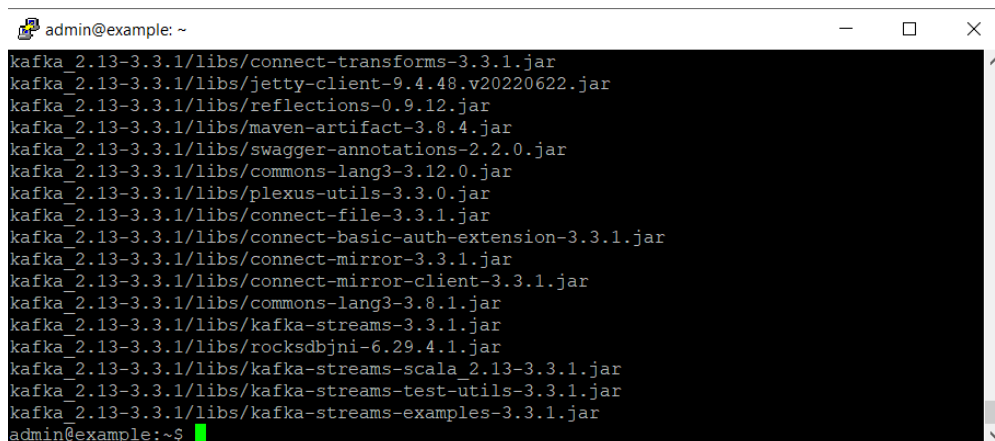
Рисунок 24 – Пример выполнения команды скачивания дистрибутива программы Kafka

2. Распаковать полученный массив (пример выполнения команды см. Рисунок 28 и Рисунок):

```
tar zxvf kafka_2.13-3.3.1.tgz
```

```
admin@example: ~
admin@example:~$ tar zxvf kafka_2.13-3.3.1.tgz
```

Рисунок 28 – Пример выполнения команды распаковки программы Kafka. Начало

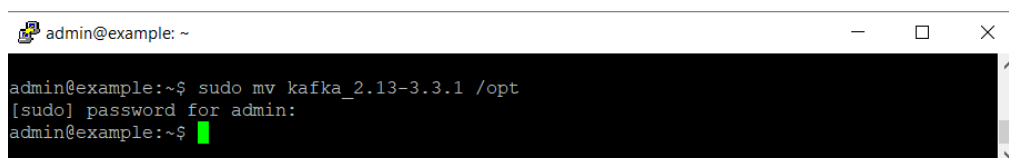


```
admin@example: ~  
kafka_2.13-3.3.1/libs/connect-transforms-3.3.1.jar  
kafka_2.13-3.3.1/libs/jetty-client-9.4.48.v20220622.jar  
kafka_2.13-3.3.1/libs/reflections-0.9.12.jar  
kafka_2.13-3.3.1/libs/maven-artifact-3.8.4.jar  
kafka_2.13-3.3.1/libs/swagger-annotations-2.2.0.jar  
kafka_2.13-3.3.1/libs/commons-lang3-3.12.0.jar  
kafka_2.13-3.3.1/libs/plexus-utils-3.3.0.jar  
kafka_2.13-3.3.1/libs/connect-file-3.3.1.jar  
kafka_2.13-3.3.1/libs/connect-basic-auth-extension-3.3.1.jar  
kafka_2.13-3.3.1/libs/connect-mirror-3.3.1.jar  
kafka_2.13-3.3.1/libs/connect-mirror-client-3.3.1.jar  
kafka_2.13-3.3.1/libs/commons-lang3-3.8.1.jar  
kafka_2.13-3.3.1/libs/kafka-streams-3.3.1.jar  
kafka_2.13-3.3.1/libs/rocksdbjni-6.29.4.1.jar  
kafka_2.13-3.3.1/libs/kafka-streams-scala_2.13-3.3.1.jar  
kafka_2.13-3.3.1/libs/kafka-streams-test-utils-3.3.1.jar  
kafka_2.13-3.3.1/libs/kafka-streams-examples-3.3.1.jar  
admin@example:~$
```

Рисунок 26 – Пример выполнения команды распаковки программы Kafka. Окончание

3. Переместить развернутую папку приложения Kafka в директорию «/opt» (пример выполнения команды см. Рисунок 27):

```
sudo mv kafka_2.13-3.3.1 /opt
```



```
admin@example: ~  
admin@example:~$ sudo mv kafka_2.13-3.3.1 /opt  
[sudo] password for admin:  
admin@example:~$
```

Рисунок 27 – Пример выполнения команды переноса программы Kafka в директорию «/opt»

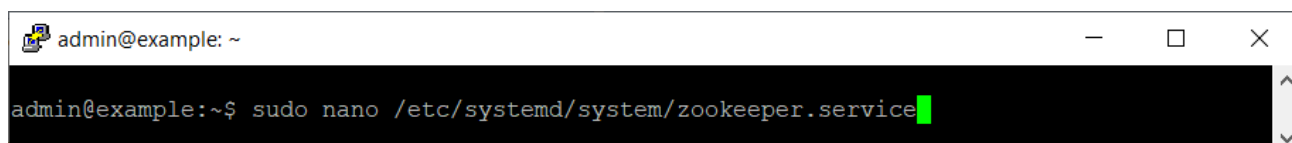
4. Создать сервисный файл systemd для сервиса «zookeeper.service» и поместить его в /etc/systemd/system (пример файла см. Рисунок 28 и Рисунок 29):

Файл «zookeeper.service»:

```
[Unit]  
Description=Apache Zookeeper server  
Documentation=http://zookeeper.apache.org  
Requires=network.target remote-fs.target  
After=network.target remote-fs.target  
  
[Service]  
Type=simple  
Environment="JAVA_HOME=/opt/java17/jdk-17.0.6"  
ExecStart=/usr/bin/bash /opt/kafka_2.13-3.3.1/bin/zookeeper-server-start.sh  
/opt/kafka_2.13-3.3.1/config/zookeeper.properties  
ExecStop=/usr/bin/bash /opt/kafka_2.13-3.3.1/bin/zookeeper-server-stop.sh  
Restart=on-abnormal  
  
[Install]  
WantedBy=multi-user.target
```

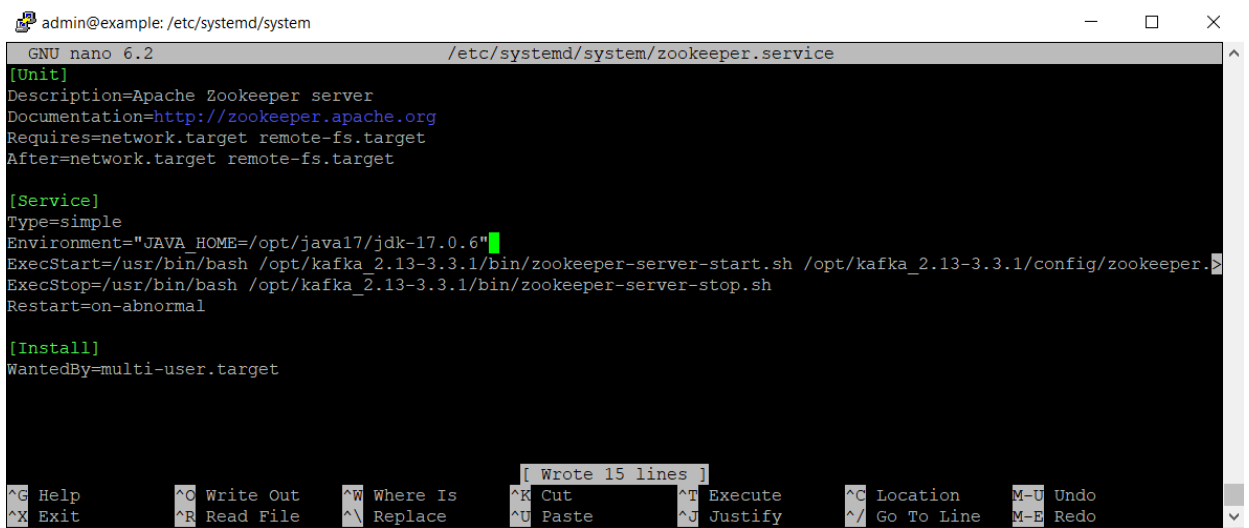
Команда для редактирования файла «zookeeper.service»:

```
sudo nano /etc/systemd/system/zookeeper.service
```



```
admin@example: ~  
admin@example:~$ sudo nano /etc/systemd/system/zookeeper.service
```

Рисунок 28 – Пример выполнения команды открытия для редактирования файла «zookeeper.service»



```
admin@example: /etc/systemd/system
GNU nano 6.2 /etc/systemd/system/zookeeper.service
[Unit]
Description=Apache Zookeeper server
Documentation=http://zookeeper.apache.org
Requires=network.target remote-fs.target
After=network.target remote-fs.target

[Service]
Type=simple
Environment="JAVA_HOME=/opt/java17/jdk-17.0.6"
ExecStart=/usr/bin/bash /opt/kafka_2.13-3.3.1/bin/zookeeper-server-start.sh /opt/kafka_2.13-3.3.1/config/zookeeper.
ExecStop=/usr/bin/bash /opt/kafka_2.13-3.3.1/bin/zookeeper-server-stop.sh
Restart=on-abnormal

[Install]
WantedBy=multi-user.target

[Wrote 15 lines]
```

Рисунок 29 – Пример файла «zookeeper.service»

5. Создать сервисный файл systemd для сервиса «kafka.service» и поместить его в /etc/systemd/system (пример файла см. Рисунок 930 и Рисунок 3131):

Файл «kafka.service»:

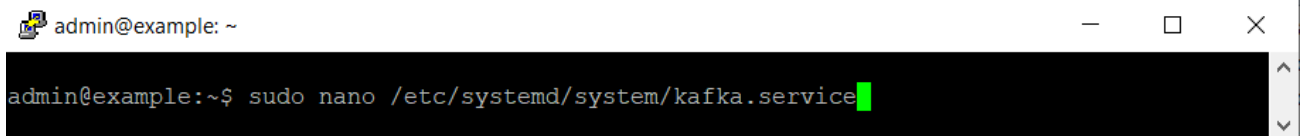
```
[Unit]
Description=Apache Kafka Server
Documentation=http://kafka.apache.org/documentation.html
Requires=zookeeper.service

[Service]
Type=simple
Environment="JAVA_HOME=/opt/java17/jdk-17.0.6"
ExecStart=/usr/bin/bash /opt/kafka_2.13-3.3.1/bin/kafka-server-start.sh
/opt/kafka_2.13-3.3.1/config/server.properties
ExecStop=/usr/bin/bash /opt/kafka_2.13-3.3.1/kafka-server-stop.sh

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

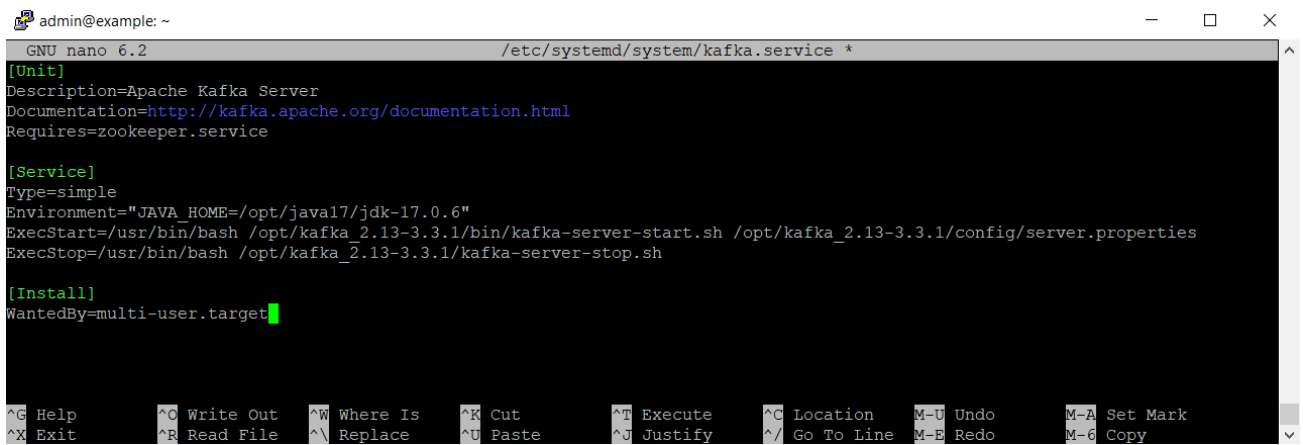
Команда для редактирования файла «kafka.service»:

```
sudo nano /etc/systemd/system/kafka.service
```



```
admin@example: ~
admin@example:~$ sudo nano /etc/systemd/system/kafka.service
```

Рисунок 9 – Пример выполнения команды открытия для редактирования файла «kafka.service»



```
admin@example: ~
GNU nano 6.2 /etc/systemd/system/kafka.service *
[Unit]
Description=Apache Kafka Server
Documentation=http://kafka.apache.org/documentation.html
Requires=zookeeper.service

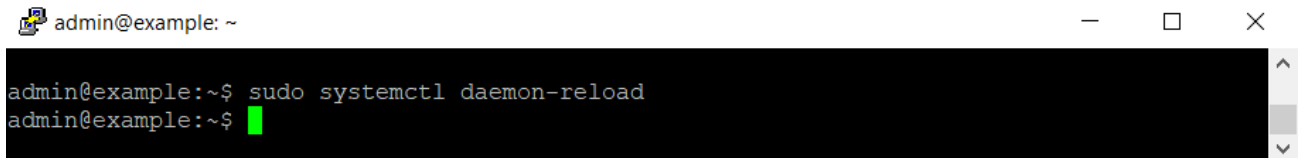
[Service]
Type=simple
Environment="JAVA_HOME=/opt/java17/jdk-17.0.6"
ExecStart=/usr/bin/bash /opt/kafka_2.13-3.3.1/bin/kafka-server-start.sh /opt/kafka_2.13-3.3.1/config/server.properties
ExecStop=/usr/bin/bash /opt/kafka_2.13-3.3.1/kafka-server-stop.sh

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

Рисунок 31 – Пример файла «kafka.service»

6. Обновить перечень доступных сервисов операционной системы (пример выполнения команды см. Рисунок 10):

```
sudo systemctl daemon-reload
```

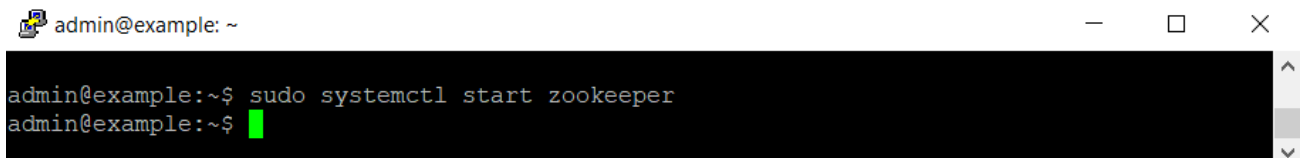


```
admin@example: ~
admin@example:~$ sudo systemctl daemon-reload
admin@example:~$
```

Рисунок 10 – Пример выполнения команды обновления сервисов операционной системы

7. Запустить сервис zookeeper (пример выполнения команды см. Рисунок 11):

```
sudo systemctl start zookeeper
```

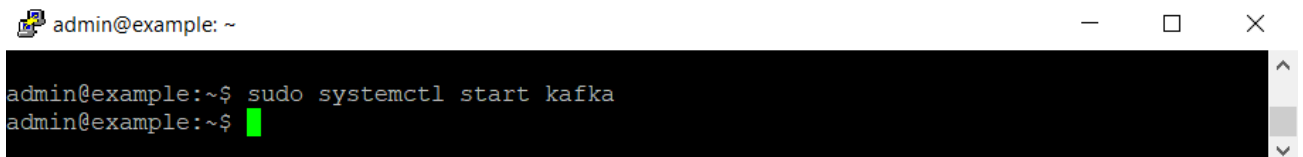


```
admin@example: ~
admin@example:~$ sudo systemctl start zookeeper
admin@example:~$
```

Рисунок 11 – Пример выполнения команды запуска сервиса «zookeeper»

8. Запустить сервис kafka (пример выполнения команды см. Рисунок 12):

```
sudo systemctl start kafka
```



```
admin@example: ~
admin@example:~$ sudo systemctl start kafka
admin@example:~$
```

Рисунок 12 – Пример выполнения команды запуска сервиса «kafka»

9. Убедиться, что сервис zookeeper запущен (пример выполнения команды см. Рисунок 13):

```
sudo systemctl status zookeeper
```

```
admin@example: /etc/systemd/system
admin@example:/etc/systemd/system$ sudo systemctl status zookeeper
● zookeeper.service - Apache Zookeeper server
   Loaded: loaded (/etc/systemd/system/zookeeper.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-06-22 12:25:11 UTC; 16s ago
     Docs: http://zookeeper.apache.org
    Main PID: 46445 (java)
      Tasks: 28 (limit: 23860)
     Memory: 50.4M
        CPU: 1.780s
    CGroup: /system.slice/zookeeper.service
            └─46445 /opt/java17/jdk-17.0.6/bin/java -Xmx512M -Xms512M -server -XX:+UseG1GC -XX:MaxGCPauseMillis=20>

Jun 22 12:25:13 example bash[46445]: [2023-06-22 12:25:13,008] INFO zookeeper.commitLogCount=500 (org.apache.zookee>
Jun 22 12:25:13 example bash[46445]: [2023-06-22 12:25:13,015] INFO zookeeper.snapshot.compression.method = CHECKED>
Jun 22 12:25:13 example bash[46445]: [2023-06-22 12:25:13,015] INFO Snapshotting: 0x0 to /tmp/zookeeper/version-2/s>
Jun 22 12:25:13 example bash[46445]: [2023-06-22 12:25:13,019] INFO Snapshot loaded in 10 ms, highest zxid is 0x0,>
Jun 22 12:25:13 example bash[46445]: [2023-06-22 12:25:13,019] INFO Snapshotting: 0x0 to /tmp/zookeeper/version-2/s>
Jun 22 12:25:13 example bash[46445]: [2023-06-22 12:25:13,020] INFO Snapshot taken in 0 ms (org.apache.zookeeper.se>
Jun 22 12:25:13 example bash[46445]: [2023-06-22 12:25:13,038] INFO PrepRequestProcessor (sid:0) started, reconfigE>
Jun 22 12:25:13 example bash[46445]: [2023-06-22 12:25:13,038] INFO zookeeper.request.throttler.shutdownTimeout = 1>
Jun 22 12:25:13 example bash[46445]: [2023-06-22 12:25:13,055] INFO Using checkIntervalMs=60000 maxPerMinute=10000>
Jun 22 12:25:13 example bash[46445]: [2023-06-22 12:25:13,056] INFO ZooKeeper audit is disabled. (org.apache.zookee>
lines 1-21/21 (END)
```

Рисунок 13 – Пример выполнения команды проверки статуса работы сервиса «zookeeper»

10. Убедиться, что сервис kafka запущен (пример выполнения команды см. Рисунок 14):

```
sudo systemctl status kafka
```

```
admin@example: /etc/systemd/system
admin@example:/etc/systemd/system$ sudo systemctl status kafka
● kafka.service - Apache Kafka Server
   Loaded: loaded (/etc/systemd/system/kafka.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-06-22 12:27:46 UTC; 19s ago
     Docs: http://kafka.apache.org/documentation.html
    Main PID: 46832 (java)
      Tasks: 74 (limit: 23860)
     Memory: 306.5M
        CPU: 5.324s
    CGroup: /system.slice/kafka.service
            └─46832 /opt/java17/jdk-17.0.6/bin/java -Xmx1G -Xms1G -server -XX:+UseG1GC -XX:MaxGCPauseMillis=20 -XX>

Jun 22 12:27:49 example bash[46832]: [2023-06-22 12:27:49,468] INFO [TransactionCoordinator id=0] Startup complete.>
Jun 22 12:27:49 example bash[46832]: [2023-06-22 12:27:49,524] INFO [ExpirationReaper-0-AlterAcls]: Starting (kafka>
Jun 22 12:27:49 example bash[46832]: [2023-06-22 12:27:49,556] INFO [/config/changes-event-process-thread]: Startin>
Jun 22 12:27:49 example bash[46832]: [2023-06-22 12:27:49,584] INFO [SocketServer listenerType=ZK_BROKER, nodeId=0]>
Jun 22 12:27:49 example bash[46832]: [2023-06-22 12:27:49,599] INFO Kafka version: 3.3.1 (org.apache.kafka.common.u>
Jun 22 12:27:49 example bash[46832]: [2023-06-22 12:27:49,600] INFO Kafka commitId: e23c59d00e687ff5 (org.apache.ka>
Jun 22 12:27:49 example bash[46832]: [2023-06-22 12:27:49,600] INFO Kafka startTimeMs: 1687436869596 (org.apache.ka>
Jun 22 12:27:49 example bash[46832]: [2023-06-22 12:27:49,601] INFO [KafkaServer id=0] started (kafka.server.KafkaS>
Jun 22 12:27:49 example bash[46832]: [2023-06-22 12:27:49,647] INFO [BrokerToControllerChannelManager broker=0 name>
Jun 22 12:27:49 example bash[46832]: [2023-06-22 12:27:49,651] INFO [BrokerToControllerChannelManager broker=0 name>
lines 1-21/21 (END)
```

Рисунок 14 – Пример выполнения команды проверки статуса работы сервиса «kafka»

11. Найти и открыть для корректировки файл «hosts» (пример выполнения команды см. Рисунок 15):

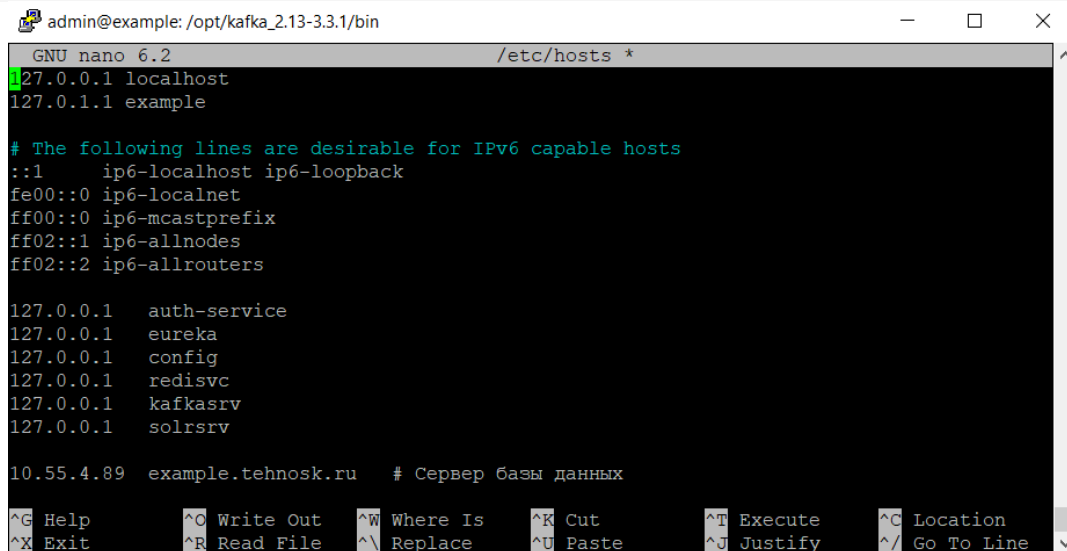
```
sudo nano /etc/hosts
```

```
admin@example: ~
admin@example:~$ sudo nano /etc/hosts
```

Рисунок 15 – Пример выполнения команды открытия файла для корректировки «hosts»

Проверить наличие, и если нет, дописать в файл «hosts» следующую строку (пример выполнения команды см. Рисунок 15):

```
127.0.0.1 kafkasrv
```



```
admin@example: /opt/kafka_2.13-3.3.1/bin
GNU nano 6.2 /etc/hosts *
127.0.0.1 localhost
127.0.1.1 example

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

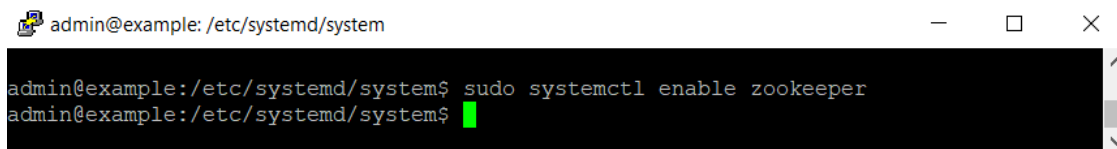
127.0.0.1 auth-service
127.0.0.1 eureka
127.0.0.1 config
127.0.0.1 redisvc
127.0.0.1 kafkasrv
127.0.0.1 solrsrv

10.55.4.89 example.tehnosk.ru # Сервер базы данных
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line
```

Рисунок 39 – Пример корректировки файла «hosts»

12. Для автоматического запуска сервиса «zookeeper» при каждом старте операционной системы необходимо выполнить команду (пример выполнения команды см. Рисунок 16):

```
sudo systemctl enable zookeeper
```

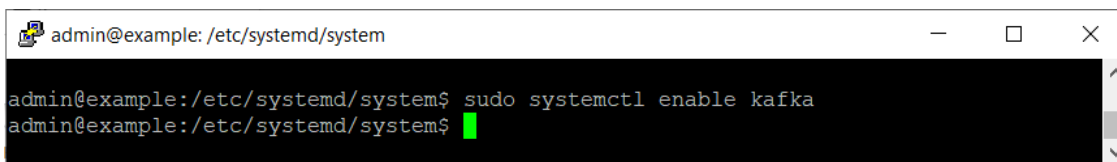


```
admin@example: /etc/systemd/system
admin@example:/etc/systemd/system$ sudo systemctl enable zookeeper
admin@example:/etc/systemd/system$
```

Рисунок 16 – Пример выполнения команды для службы «zookeeper»

13. Для автоматического запуска сервиса «kafka» при каждом старте операционной системы необходимо выполнить команду (пример выполнения команды см. Рисунок 17):

```
sudo systemctl enable kafka
```

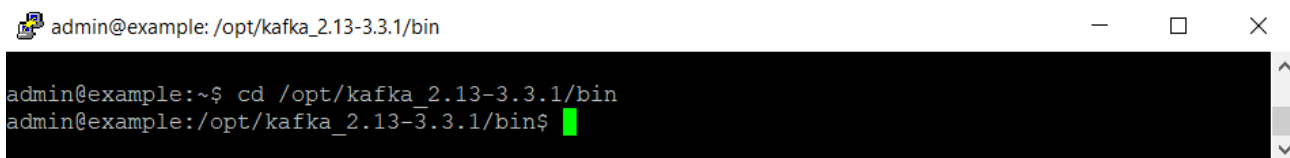


```
admin@example: /etc/systemd/system
admin@example:/etc/systemd/system$ sudo systemctl enable kafka
admin@example:/etc/systemd/system$
```

Рисунок 17 – Пример выполнения команды для службы «kafka»

14. Перейти в каталог /opt/kafka_2.13-3.3.1/bin (пример выполнения команды см. Рисунок 18):

```
cd /opt/kafka_2.13-3.3.1/bin
```

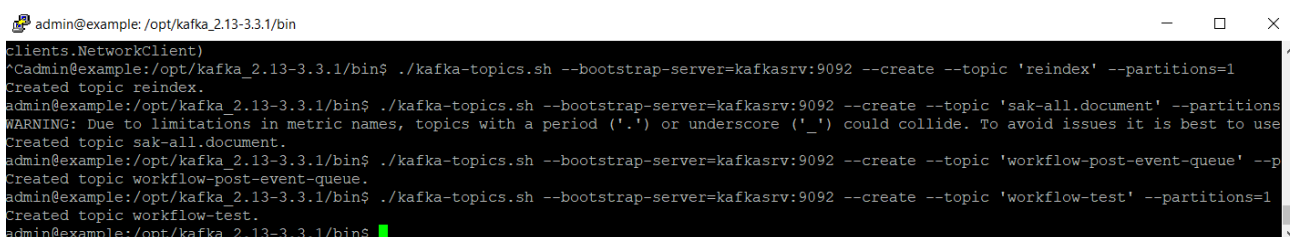


```
admin@example: /opt/kafka_2.13-3.3.1/bin  
admin@example:~$ cd /opt/kafka_2.13-3.3.1/bin  
admin@example:/opt/kafka_2.13-3.3.1/bin$
```

Рисунок 18 – Пример выполнения команды перехода в каталог «/opt/kafka_2.13-3.3.1/bin»

15. Выполнить команды создания очередей (пример выполнения команды см. Рисунок 19):

```
./kafka-topics.sh --bootstrap-server=kafkasrv:9092 --create --topic 'reindex' --  
partitions=1  
./kafka-topics.sh --bootstrap-server=kafkasrv:9092 --create --topic 'sak-  
all.document' --partitions=1  
./kafka-topics.sh --bootstrap-server=kafkasrv:9092 --create --topic 'workflow-post-  
event-queue' --partitions=1  
./kafka-topics.sh --bootstrap-server=kafkasrv:9092 --create --topic 'workflow-test'  
--partitions=1
```



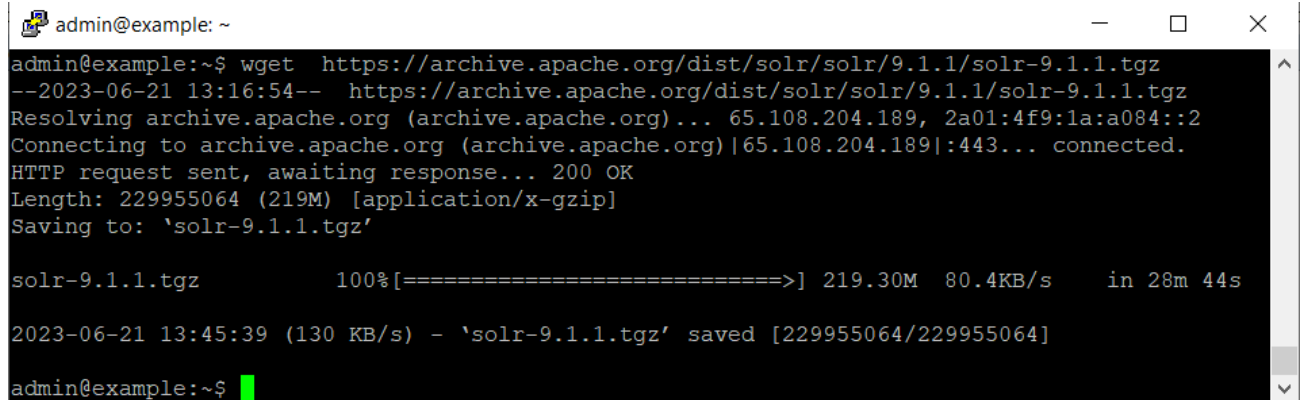
```
admin@example: /opt/kafka_2.13-3.3.1/bin  
clients.NetworkClient)  
admin@example:/opt/kafka_2.13-3.3.1/bin$ ./kafka-topics.sh --bootstrap-server=kafkasrv:9092 --create --topic 'reindex' --partitions=1  
Created topic reindex.  
admin@example:/opt/kafka_2.13-3.3.1/bin$ ./kafka-topics.sh --bootstrap-server=kafkasrv:9092 --create --topic 'sak-all.document' --partitions  
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could collide. To avoid issues it is best to use  
Created topic sak-all.document.  
admin@example:/opt/kafka_2.13-3.3.1/bin$ ./kafka-topics.sh --bootstrap-server=kafkasrv:9092 --create --topic 'workflow-post-event-queue' --p  
Created topic workflow-post-event-queue.  
admin@example:/opt/kafka_2.13-3.3.1/bin$ ./kafka-topics.sh --bootstrap-server=kafkasrv:9092 --create --topic 'workflow-test' --partitions=1  
Created topic workflow-test.  
admin@example:/opt/kafka_2.13-3.3.1/bin$
```

Рисунок 19 – Пример выполнения команд создания очередей в программе Kafka

4.3.6 Установка и настройка Apache Solr

1. Скачать дистрибутив программы Solr (пример выполнения команды см. Рисунок 20):

```
wget https://archive.apache.org/dist/solr/solr/9.1.1/solr-9.1.1.tgz
```



```
admin@example: ~  
admin@example:~$ wget https://archive.apache.org/dist/solr/solr/9.1.1/solr-9.1.1.tgz  
--2023-06-21 13:16:54-- https://archive.apache.org/dist/solr/solr/9.1.1/solr-9.1.1.tgz  
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 2a01:4f9:1a:a084::2  
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 229955064 (219M) [application/x-gzip]  
Saving to: 'solr-9.1.1.tgz'  
  
solr-9.1.1.tgz      100%[=====>] 219.30M  80.4KB/s   in 28m 44s  
2023-06-21 13:45:39 (130 KB/s) - 'solr-9.1.1.tgz' saved [229955064/229955064]  
admin@example:~$
```

Рисунок 20 – Пример выполнения команд скачивания дистрибутива программы Solr

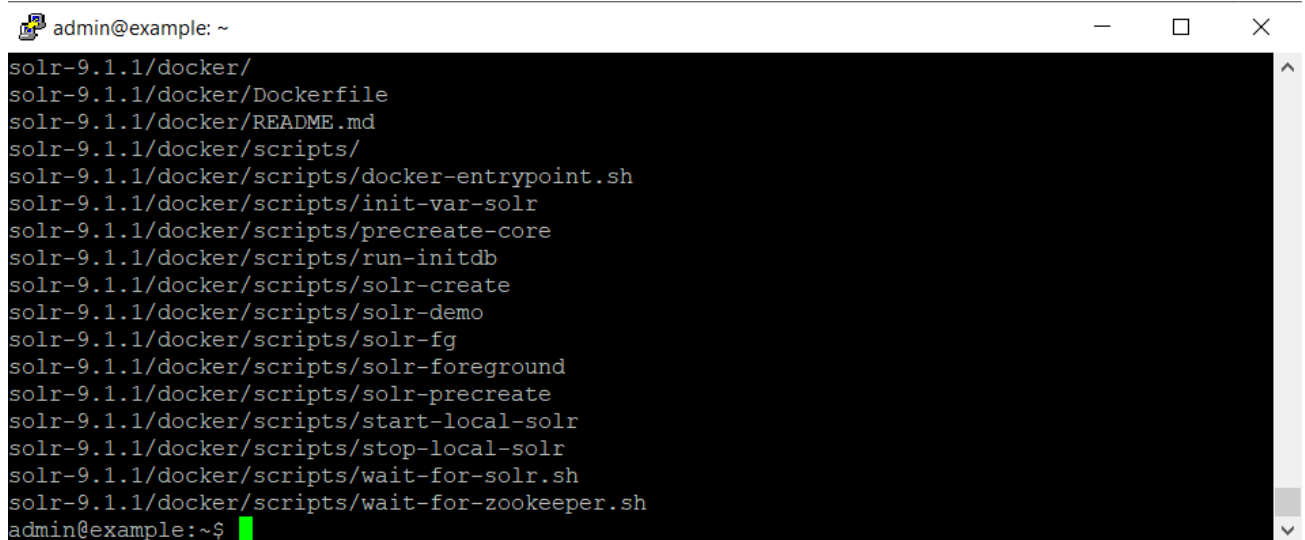
2. Распаковать полученный архив в каталог «/opt» (пример выполнения команды см. Рисунок 21 и Рисунок 22):

```
tar zxvf solr-9.1.1.tgz
```



```
admin@example: ~  
admin@example:~$ tar zxvf solr-9.1.1.tgz
```

Рисунок 21 – Пример выполнения команд разархивирования архива программы Solr. Начало



```
admin@example: ~  
solr-9.1.1/docker/  
solr-9.1.1/docker/Dockerfile  
solr-9.1.1/docker/README.md  
solr-9.1.1/docker/scripts/  
solr-9.1.1/docker/scripts/docker-entrypoint.sh  
solr-9.1.1/docker/scripts/init-var-solr  
solr-9.1.1/docker/scripts/precreate-core  
solr-9.1.1/docker/scripts/run-initdb  
solr-9.1.1/docker/scripts/solr-create  
solr-9.1.1/docker/scripts/solr-demo  
solr-9.1.1/docker/scripts/solr-fg  
solr-9.1.1/docker/scripts/solr-foreground  
solr-9.1.1/docker/scripts/solr-precreate  
solr-9.1.1/docker/scripts/start-local-solr  
solr-9.1.1/docker/scripts/stop-local-solr  
solr-9.1.1/docker/scripts/wait-for-solr.sh  
solr-9.1.1/docker/scripts/wait-for-zookeeper.sh  
admin@example:~$
```

Рисунок 22 – Пример выполнения команд разархивирования архива программы Solr. Окончание

- Переместить рабочий каталог программы Solr в каталог «/opt» (пример выполнения команды см. Рисунок 23):

```
sudo mv solr-9.1.1 /opt
```



```
admin@example: ~  
admin@example:~$ sudo mv solr-9.1.1 /opt  
[sudo] password for admin:  
admin@example:~$
```

Рисунок 23 – Пример выполнения команд перемещения рабочего каталога программы Solr

3. Перейти в каталог «/opt/solr-9.1.1/bin» (пример выполнения команды см. Рисунок):

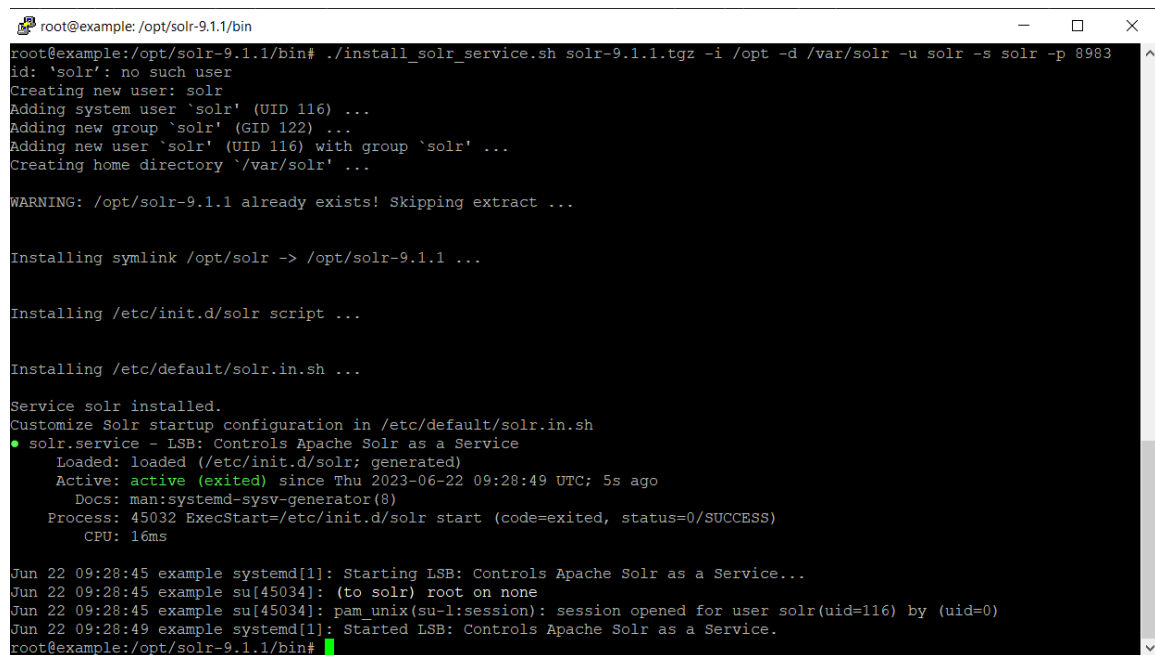


```
admin@example: /opt/solr-9.1.1/bin  
admin@example:~$ cd /opt/solr-9.1.1/bin/  
admin@example:/opt/solr-9.1.1/bin$
```

Рисунок 48 – Пример выполнения команд перехода в каталог «/opt/solr-9.1.1/bin»

4. Выполнить инициализацию программы Solr (пример выполнения команды см. Рисунок):

```
sudo ./install_solr_service.sh solr-9.1.1.tgz -i /opt -d /var/solr -u solr -s solr -p 8983
```

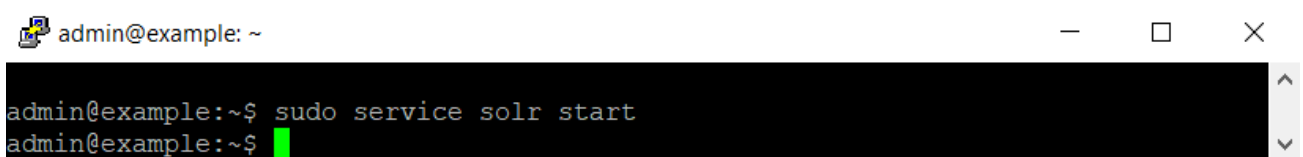


```
root@example: /opt/solr-9.1.1/bin  
root@example:/opt/solr-9.1.1/bin# ./install_solr_service.sh solr-9.1.1.tgz -i /opt -d /var/solr -u solr -s solr -p 8983  
id: 'solr': no such user  
Creating new user: solr  
Adding system user 'solr' (UID 116) ...  
Adding new group 'solr' (GID 122) ...  
Adding new user 'solr' (UID 116) with group 'solr' ...  
Creating home directory '/var/solr' ...  
  
WARNING: /opt/solr-9.1.1 already exists! Skipping extract ...  
  
Installing symlink /opt/solr -> /opt/solr-9.1.1 ...  
  
Installing /etc/init.d/solr script ...  
  
Installing /etc/default/solr.in.sh ...  
  
Service solr installed.  
Customize Solr startup configuration in /etc/default/solr.in.sh  
● solr.service - LSB: Controls Apache Solr as a Service  
   Loaded: loaded (/etc/init.d/solr; generated)  
   Active: active (exited) since Thu 2023-06-22 09:28:49 UTC; 5s ago  
   Docs: man:systemd-sysv-generator(8)  
   Process: 45032 ExecStart=/etc/init.d/solr start (code=exited, status=0/SUCCESS)  
   CPU: 16ms  
  
Jun 22 09:28:45 example systemd[1]: Starting LSB: Controls Apache Solr as a Service...  
Jun 22 09:28:45 example su[45034]: (to solr) root on none  
Jun 22 09:28:45 example su[45034]: pam_unix(su-l:session): session opened for user solr(uid=116) by (uid=0)  
Jun 22 09:28:49 example systemd[1]: Started LSB: Controls Apache Solr as a Service.  
root@example:/opt/solr-9.1.1/bin#
```

Рисунок 49 – Пример выполнения команд инициализации программы Solr

5. Выполнить запуск программы Solr (пример выполнения команды см. Рисунок 24):

```
sudo service solr start
```

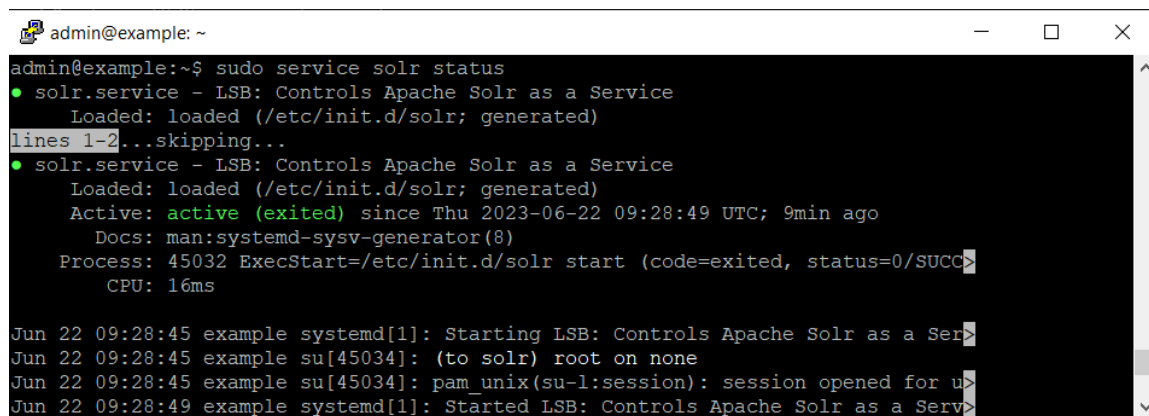


```
admin@example: ~  
admin@example:~$ sudo service solr start  
admin@example:~$
```

Рисунок 24 – Пример выполнения команды запуска программы Solr

6. Проверить статус программы Solr (пример выполнения команды см. Рисунок 25):

```
sudo service solr status
```

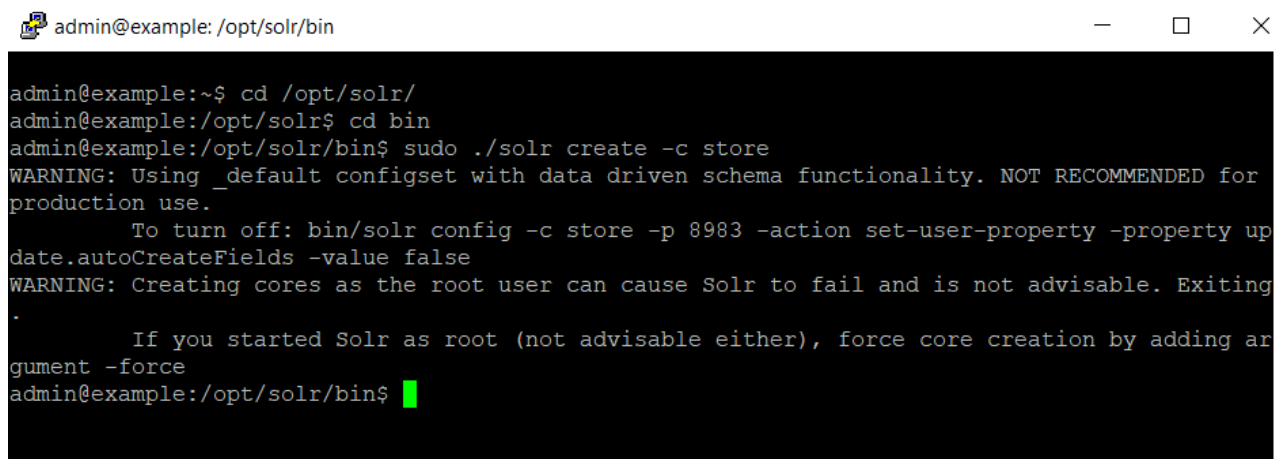


```
admin@example: ~  
admin@example:~$ sudo service solr status  
● solr.service - LSB: Controls Apache Solr as a Service  
   Loaded: loaded (/etc/init.d/solr; generated)  
   lines 1-2...skipping...  
● solr.service - LSB: Controls Apache Solr as a Service  
   Loaded: loaded (/etc/init.d/solr; generated)  
   Active: active (exited) since Thu 2023-06-22 09:28:49 UTC; 9min ago  
   Docs: man:systemd-sysv-generator(8)  
   Process: 45032 ExecStart=/etc/init.d/solr start (code=exited, status=0/SUCCESS)  
   CPU: 16ms  
  
Jun 22 09:28:45 example systemd[1]: Starting LSB: Controls Apache Solr as a Service:  
Jun 22 09:28:45 example su[45034]: (to solr) root on none  
Jun 22 09:28:45 example su[45034]: pam_unix(su-l:session): session opened for user solr  
Jun 22 09:28:49 example systemd[1]: Started LSB: Controls Apache Solr as a Service:
```

Рисунок 25 – Пример выполнения команды проверки статуса работы программы Solr

7. Создать хранилище программы Solr (пример выполнения команды см. Рисунок 26):

```
cd /opt/solr/bin/  
./solr create -c store
```



```
admin@example: /opt/solr/bin  
admin@example:~$ cd /opt/solr/  
admin@example:/opt/solr$ cd bin  
admin@example:/opt/solr/bin$ sudo ./solr create -c store  
WARNING: Using _default configset with data driven schema functionality. NOT RECOMMENDED for  
production use.  
To turn off: bin/solr config -c store -p 8983 -action set-user-property -property up  
date.autoCreateFields -value false  
WARNING: Creating cores as the root user can cause Solr to fail and is not advisable. Exiting  
.  
If you started Solr as root (not advisable either), force core creation by adding ar  
gument -force  
admin@example:/opt/solr/bin$
```

Рисунок 26 – Пример выполнения команды создания хранилища программы Solr

8. Найти и открыть для корректировки файл «hosts» (пример выполнения команды см. Рисунок 27):

```
sudo nano /etc/hosts
```



```
admin@example: ~  
admin@example:~$ sudo nano /etc/hosts
```

Рисунок 27 – Пример выполнения команды открытия файла для корректировки «hosts»

Проверить наличие, и если нет, дописать в файл «hosts» следующую строку (пример файла см. Рисунок 7342):

```
127.0.0.1 solrsrv
```

```

admin@example: /opt/kafka_2.13-3.3.1/bin
GNU nano 6.2 /etc/hosts *
27.0.0.1 localhost
127.0.1.1 example

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

127.0.0.1 auth-service
127.0.0.1 eureka
127.0.0.1 config
127.0.0.1 redisvc
127.0.0.1 kafkasrv
127.0.0.1 solrsrv

10.55.4.89 example.tehnosk.ru # Сервер базы данных
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^N Replace ^U Paste ^J Justify ^_ Go To Line

```

Рисунок 28 – Пример корректировки файла «hosts»

4.3.7 Установка и настройка микросервисов

1. Скопировать и разархивировать файл «jre1.8.0_201-amd64.rar» (из состава дистрибутива «СМПО М») в папку «/usr/java/» (см. Рисунок).

```

mc [admin@example]:/usr/java/jre1.8.0_201-amd64
Left File Command Options Right
<- /usr/java -.[^]> <- .../java/jre1.8.0_201-amd64 -.[^]>
.n Name Size Modify time .n Name Size Modify time
/.. UP--DIR Jun 21 18:02 /.. UP--DIR Jun 21 18:17
/jre1.8.~amd64 4096 Jun 21 18:01 /.java 4096 Jun 21 18:01
/bin 4096 Jun 21 18:01
/lib 4096 Jun 21 18:01
/man 4096 Jun 21 18:01
/plugin 4096 Jun 21 18:01
COPYRIGHT 3244 Dec 15 2018
LICENSE 40 Dec 15 2018
README 46 Dec 15 2018
THIRDP~AFX.txt 108109 Dec 12 2018
THIRDP~DME.txt 155002 Dec 15 2018
Welcome.html 955 Dec 15 2018
UP--DIR 179G/214G (83%) UP--DIR 179G/214G (83%)
Hint: The homepage of GNU Midnight Commander: http://www.midnight-commander.o
admin@example:/usr/java/jre1.8.0_201-amd64$
1Help 2Menu 3View 4Edit 5Copy 6Re~ov 7Mkdir 8Delete 9PullDn 10Quit

```

Рисунок 55 – Пример размещения каталога «jre1.8.0_201-amd64»

2. Скопировать микросервисы «СМПО М» (из состава дистрибутива «СМПО М») в папку «/opt/smpo-cloud/apps/» (см. Рисунок 29 и Рисунок).

```

mc [admin@example]:/opt/smpo-cloud
Left File Command Options Right
<- /opt/smpo-cloud -.[^]> <- /opt/smpo-cloud/apps -.[^]>
.n Name Size Modify time .n Name Size Modify time
/.. UP--DIR Jun 22 09:28 /.. UP--DIR Jun 22 07:59
/apps 4096 Jun 22 05:08 /activit~service 4096 Jun 22 05:38
/logs 4096 Jun 22 13:56 /activit~service 4096 Jun 22 05:38
/scripts 4096 Jun 22 06:16 /activit~service 4096 Jun 22 05:38
/ais-service 4096 Jun 22 05:38
/apps 4096 Jun 22 05:08
/auth-service 4096 Jun 22 05:38
/blank-l~service 4096 Jun 22 05:38
/blank-service 4096 Jun 22 05:38
/bugete~service 4096 Jun 22 05:38
/databas~service 4096 Jun 22 13:38
/documen~service 4096 Jun 22 05:38
/apps 179G/214G (83%) UP--DIR 179G/214G (83%)
Hint: Completion: use M-Tab (or Esc+Tab). Type it twice to get a list.
admin@example:/opt/smpo-cloud$
1Help 2Menu 3View 4Edit 5Copy 6Re~ov 7Mkdir 8Delete 9PullDn 10Quit

```

Рисунок 29 – Пример размещения каталога «apps»

activities-info-import-service	document-service	modeling	pdf-service	spde-service
activities-info-service	eureka	modeling-document-service	permission-service	store-service
activity-service	forms-desc-service	modelling-service	reference-service	user-groups-service
ais-service	gateway	nsi-service	sakura-activity-service	user-service
auth-service	groovy-exec-service	object-attr-service	sakura-service	witsml-service
blank-log-service	idx-service	object-create-service	schedule-service	workflow-service
blank-service	lang-service	objects-bookmark-service	sdpe-setup-service	application.yml
bugete-blanks-service	mail-service	object-service	sk-lang_services	logback.xml
database-service	model-fx	object-short-report-service	sk-modeller-api	output.txt

Рисунок 57 – Перечень каталогов с микросервисами системы

3. Скопировать скрипты «СМПО М» (из состава дистрибутива «СМПО М») в папку «/opt/smpo-cloud/scripts» (см. Рисунок и Рисунок).

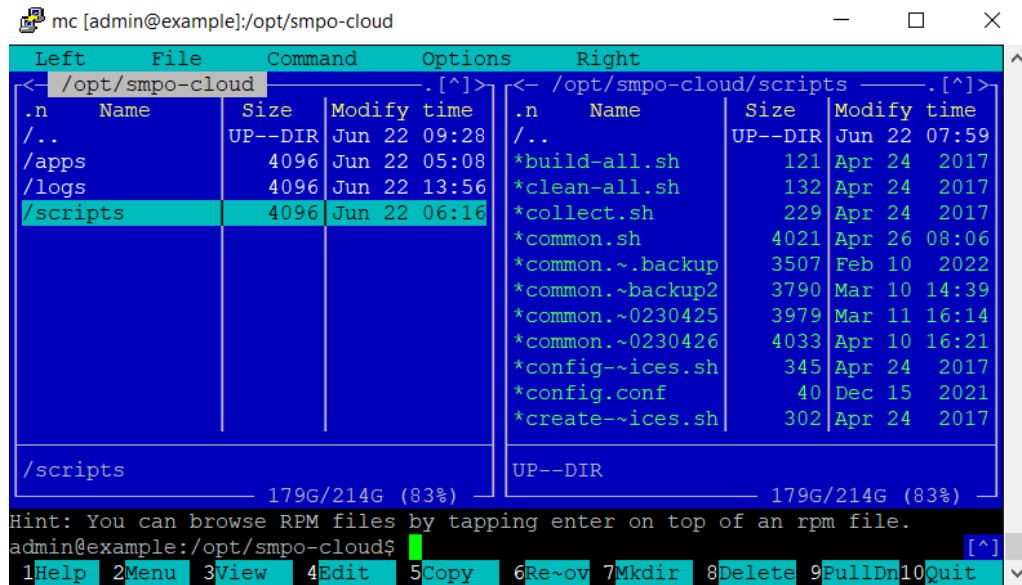


Рисунок 58 – Пример размещения каталога «scripts»

build-all.sh	config-services.sh	logback	start-smcl-local.sh
clean-all.sh	create-services.sh	restart-service.sh	start-svc-profiles.sh
collect.sh	deploy-uploaded-file.sh	services	stop-service.sh
common.sh	deploy-uploaded-front.sh	smpo-cloud.conf	stop-smcl.sh
common.sh.backup	deploy-uploaded-front.sh_orig	start-service.sh	tasks
config.conf	deploy-uploaded-service.sh	start-smcl.sh	

Рисунок 59 – Перечень скриптов системы

4. В скрипте «common.sh» изменить значение переменной «JAVA_PATH_TO_BIN», указав в ней путь к каталогу «jre1.8.0_201-amd64/bin/» (см. Рисунок 30):

```
sudo nano /opt/smpo-cloud/scripts/common.sh
```

```

admin@example: ~
GNU nano 6.2 /opt/smpo-cloud/scripts/common.sh
#!/usr/bin/env bash

#set -e

#JAVA_PATH_TO_BIN="${JAVA_PATH_TO_BIN:-}"
JAVA_PATH_TO_BIN="/usr/java/jre1.8.0_201-amd64/bin/"
APP_JAVA_PATH="${APP_JAVA_PATH:-../apps}"
LOG_JAVA_PATH="${LOG_JAVA_PATH:-../logs}"
MEM_ARGS="-Xmx512m -Xms256m"

WAIT_TIME="${WAIT_TIME:-5}"
RETRIES="${RETRIES:-20}"

# 000000 00000000, 0 0000000 00000 00000000 000000 (000000000, 000000000000)
SERVICES=`cat services.txt`
TASKS=`cat tasks.txt`

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
  
```

Рисунок 30 – Пример корректировки файла «common.sh»

5. Проверить и поправить все конфигурационные файлы с расширением «yml» в каталогах с микросервисами по пути /var/sakura-cloud/apps/. В этих файлах:

5.1. Необходимо настроить связи сервисов с базой данных системы (указать имя сервера, имя базы данных, пользователя и его пароль для подключения):

```

datasource:
  db-smpo:
    url: jdbc:postgresql://{%DBSERVERNAME%}:5432/{%DBNAME%}
    username: {%DBUSERNAME%}
    password: {%DBUSERPASSWORD%}
  
```

, где:

{%DBSERVERNAME%} – Имя или IP адрес сервера базы данных системы,
 {%DBNAME%} – Имя базы данных системы,
 {%DBUSERNAME%} – Имя пользователя базы данных,
 {%DBUSERPASSWORD%} – Пароль пользователя базы данных.

5.2. Необходимо настроить связи сервисов с контроллером домена системы (указать имя сервера, имя пользователя и его пароль для подключения):

```

ldap:
  contextSource:
    url: ldap://{%DOMAINIP%}:389
    base: OU={%DOMAINUSER%},DC={%DOMAIN%},DC=ru
    userDn: {%DOMAINUSER%}
    password: {%DOMAINUSERPASSWORD%}
    domain: {%DOMAIN%}
  
```

, где:

{%DOMAINIP%} – IP адрес контроллера домена системы,
 {%DOMAINUSER%} – Имя пользователя домена,
 {%DOMAINUSERPASSWORD%} – Пароль пользователя домена,
 {%DOMAIN%} – Имя домена.

5.3. Необходимо настроить связи сервисов с почтовым сервером системы (указать имя сервера, имя пользователя и его пароль для подключения) (файл /var/smpo-cloud/apps/mail-service/application.yml):

```

mail:
  host: smtp.mail.ru
  port: 465
  username: {%USERMAIL%}
  password: {%USERMAILPASSWORD%}
  
```

, где:

{%USERMAIL%} – Почтовый адрес пользователя системы для рассылки сообщений,
{%USERMAILPASSWORD%} – Почтовый пароль пользователя системы для рассылки сообщений.

5.4. Необходимо настроить ssl (файл /var/smpo-cloud/apps/gateway/application.yml):

```
ssl:  
enable: true  
key-store: /etc/letsencrypt/live/{%DOMAIN%}/keystore.p12  
key-alias: "{%DOMAIN%}"  
key-store-password:  
key-password:  
key-store-type: PKCS12
```

, где:

– Имя домена.

6. Создать подкаталог «logs» в каталоге «/opt/smpo-cloud» (в данном подкаталоге сервисы будут автоматически создавать log-файлы своей работы) (см. Рисунок 31).

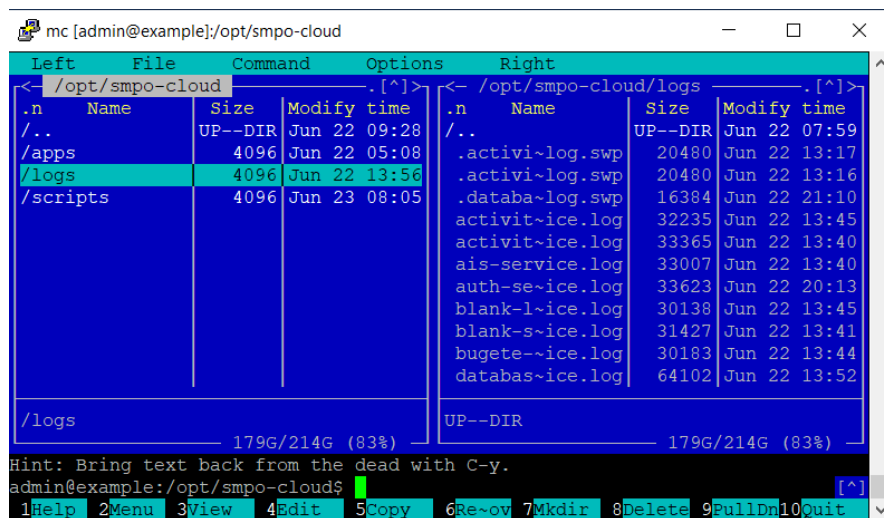


Рисунок 31 – Пример размещения каталога «logs»

7. Для запуска системы необходимо из папки «/opt/smpo-cloud/scripts» запустить следующий скрипт – начнут запускаться микросервисы и в папке «logs» начнут создаваться log-файлы (пример выполнения команды см. Рисунок 32):

```
sudo ./start-smcl.sh
```

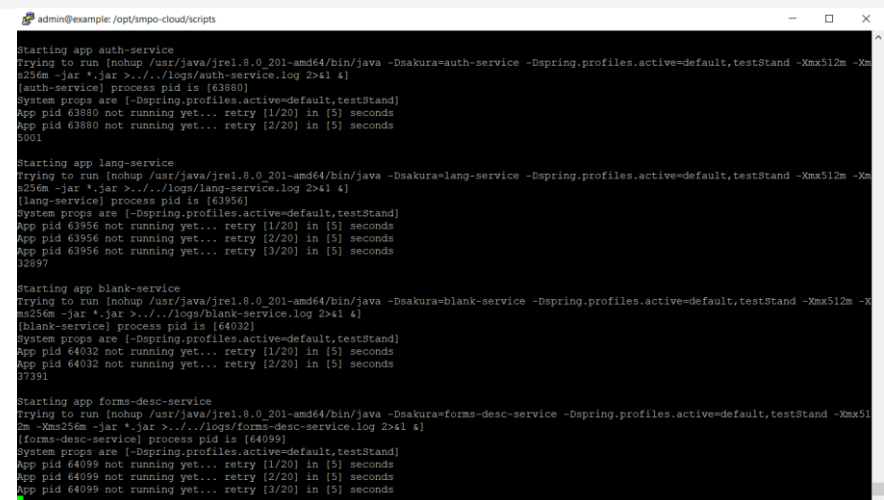


Рисунок 32 – Пример выполнения команды по запуску микросервисов «СМПО М»

4.3.8 Установка и настройка дополнительных микросервисов (через Docker)

1. Добавить в операционную систему репозиторий программного обеспечения Docker:

```
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/РЕД ОС bionic stable"
```

2. Установить Docker-ce:

```
sudo apt install docker-ce
```

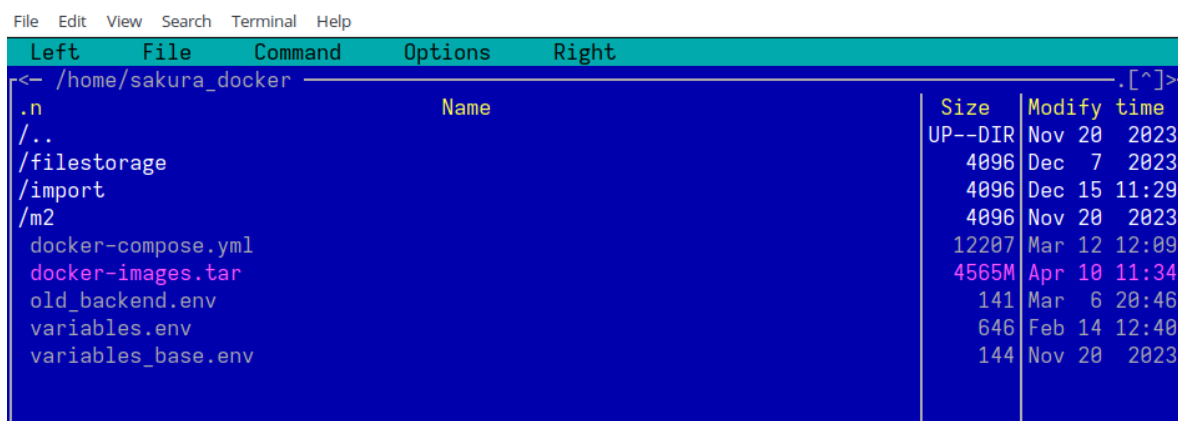
3. Добавить программу Docker в автозапуск:

```
sudo systemctl enable --now docker
```

4. Установить docker-compose:

```
sudo curl -L "https://github.com/docker/compose/releases/download/v2.12.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose; sudo chmod +x /usr/local/bin/docker-compose
```

5. Скопировать docker-compose.yml и дополнительные файлы и папки (из состава дистрибутива «СМПО М») в каталог /home/sakura_docker (см. Рисунок 33)



Left	File	Command	Options	Right
File Edit View Search Terminal Help				
Left File Command Options Right				
[<- /home/sakura_docker .[^]>				
	.n		Name	
	/..			Size Modify time
	/filestorage			UP--DIR Nov 20 2023
	/import			4096 Dec 7 2023
	/m2			4096 Dec 15 11:29
	docker-compose.yml			4096 Nov 20 2023
	docker-images.tar			12207 Mar 12 12:09
	old_backend.env			4565M Apr 10 11:34
	variables.env			141 Mar 6 20:46
	variables_base.env			646 Feb 14 12:40
				144 Nov 20 2023

Рисунок 33 – Перечень папок и файлов каталога «/home/sakura_docker»

6. Запустить команду по разворачиванию и запуску дополнительных микросервисов «СМПО М» через Docker:

```
docker-compose -f docker-compose.yml up -d
```

4.3.9 Проверка работоспособности сервера доступа, контроля и обработки данных (микросервисов)

Для проверки работоспособности сервера доступа, контроля и обработки данных (микросервисов) необходимо осуществить вход Администратором или Пользователем в автоматизированное рабочее место (АРМ) «СМПО М» (запустить Web-клиента «СМПО М»). Для

запуска АРМ «СМПО М» необходимо открыть Web-браузер и в адресной строке ввести запрос к серверу вида

```
https://{%DOMAINSERVERNAME%}
```

, где:

{%DOMAINSERVERNAME%} – Доменное имя сервера доступа, контроля и обработки данных (сервер микросервисов)

При успешном подключении к серверу доступа, контроля и обработки данных (серверу микросервисов) откроется окно авторизации для входа в АРМ «СМПО М» (см. Рисунок 34).

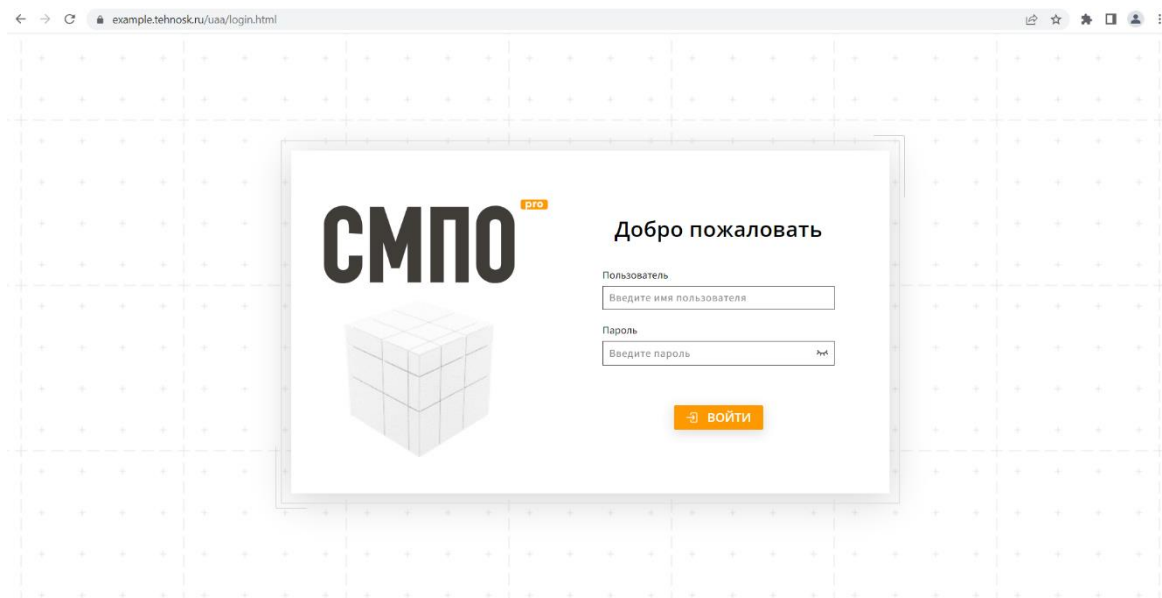


Рисунок 34 – Пример проверки работоспособности сервера доступа, контроля и обработки данных (микросервисов). Запуск Web-клиента «СМПО М». Окно авторизации

При успешной авторизации Администратор или Пользователь «СМПО М» успешно входит в АРМ «СМПО М». В зависимости от назначенных прав Администратору или Пользователю «СМПО М» будет доступны определенные режимы и функции системы (см. Рисунок 35).

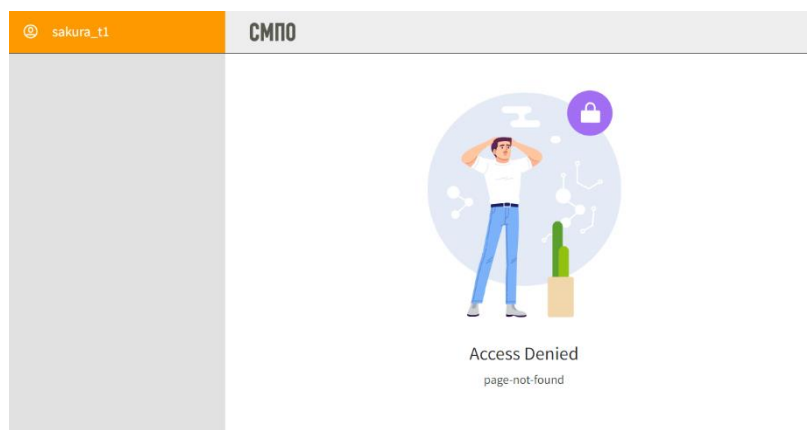


Рисунок 35 – Пример проверки работоспособности сервера доступа, контроля и обработки данных (микросервисов). Запуск Web-клиента «СМПО М». АРМ

4.3.10 Схема резервирования сервера доступа, контроля и обработки данных (микросервисов)

Резервирование сервера доступа, контроля и обработки данных (микросервисов) может осуществляться по одному из следующих сценариев, приведенных в таблице 2.

Таблица 2 – Схема резервирования сервера доступа, контроля и обработки данных (микросервисов)

Тип	Резервная единица оборудования	Периодичность создания (настройки) резерва
1	2	3
Физические серверы	Резервные серверы	Модификация (обновление) сервера доступа, контроля и обработки данных (микросервисов) на резервном оборудовании должна осуществляться при каждой модификации (обновлении) сервера доступа, контроля и обработки данных (микросервисов) на основных серверах.
Виртуальные серверы	Резервируется системой виртуальных машин, под управлением которой работают данные виртуальные серверы	Резервирование сервера доступа, контроля и обработки данных (микросервисов) с использованием копии виртуальной машины должно осуществляться после каждой модификации сервера доступа, контроля и обработки данных (микросервисов). На случай, если модификация оказалась неудачной, рекомендуется хранить две резервные копии виртуальных машин (с предыдущей и настоящей модификациями сервера доступа, контроля и обработки данных (микросервисов)).

4.3.11 Схема восстановления сервера доступа, контроля и обработки данных (микросервисов)

Восстановление сервера доступа, контроля и обработки данных (микросервисов) может осуществляться по одному из следующих сценариев, приведенных в таблице 2.

Таблица 2 – Схема восстановления сервера доступа, контроля и обработки данных (микросервисов)

Тип	Резервная единица оборудования	Действия по восстановлению
1	2	3
Физические серверы	Резервный сервер	1) исключить из рабочего процесса вышедшее из строя оборудование; 2) ввести в работу резервное оборудование; 3) осуществить настройку ПО и систем, взаимодействующих с вышедшей из строя единицей КТС и ПО, установленным на ней, по переключению на взаимодействие с резервной единицей.
Виртуальные серверы	Резервируется системой виртуальных машин, под управлением которой работают данные виртуальные серверы	1) исключить из рабочего процесса вышедшую из строя виртуальную машину; 2) ввести в работу резервную копию вышедшей из строя виртуальной машины; 3) осуществить настройку ПО и систем, взаимодействующих с вышедшей из строя виртуальной машиной и ПО, установленным на ней, по переключению на взаимодействие с резервной единицей.

4.4 Установка и настройка сервера бизнес-аналитики Apache Superset

Установка Apache Superset осуществляется на сервере бизнес-аналитики (S-06).

4.4.1 Настройка файла Hosts

1. Найти и открыть для корректировки файл «hosts» (пример выполнения команды см. Рисунок 36):

```
sudo nano /etc/hosts
```

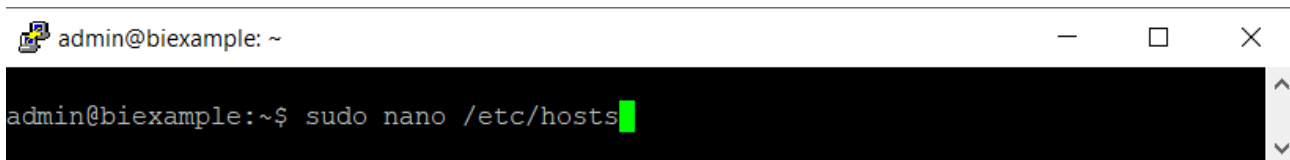


Рисунок 36 – Пример выполнения команды открытия для корректировки файла «hosts»

2. Дописать в файл «hosts» следующие строки (пример выполнения команды см. Рисунок 37):

```
127.0.0.1    localhost
127.0.0.1    biexample

{%SERVERDBIP%} {%SERVERDBNAME%} # Сервер базы данных
{%SERVERMSIP%} {%SERVERMSNAME%} # Сервер доступа, контроля и обработки данных
{%SERVERBIIP%} {%SERVERBINAME%} # Сервер аналитической подсистемы
```

, где:

{%SERVERDBIP%} и {%SERVERDBNAME%} – IP адрес и имя Сервера базы данных,
{%SERVERMSIP%} и {%SERVERMSNAME%} – IP адрес и имя Сервера доступа, контроля и
обработки данных (микросервисов),
{%SERVERBIIP%} и {%SERVERBINAME%} – IP адрес и имя Сервера аналитической
подсистемы.

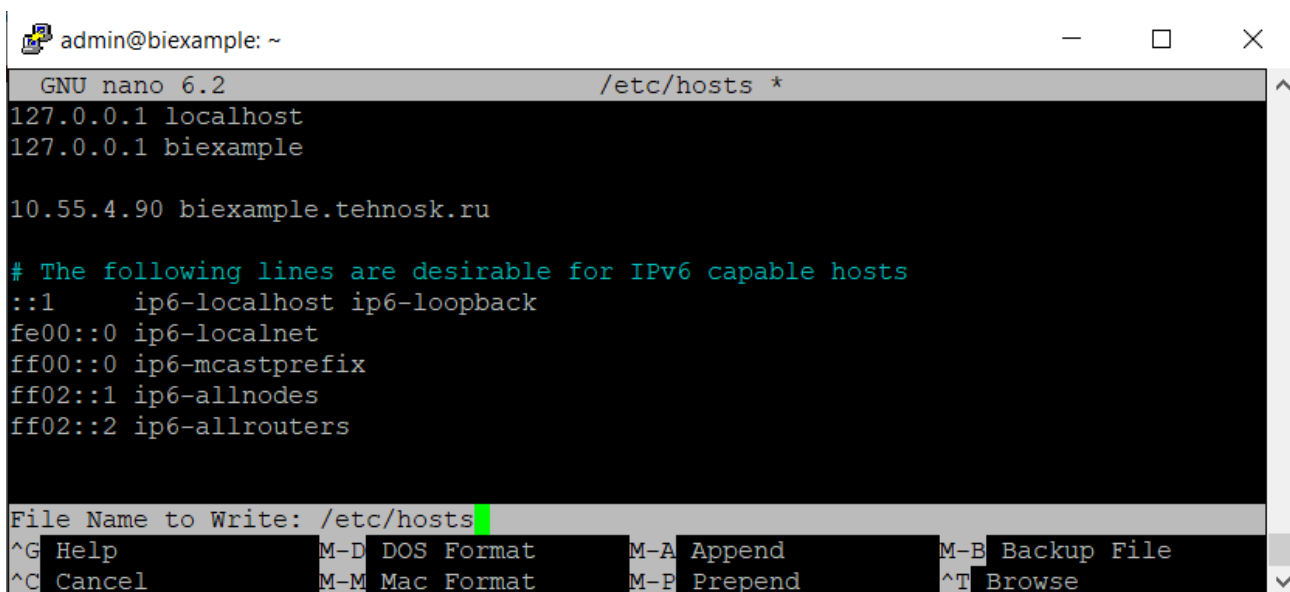


Рисунок 37 – Пример корректировки файла «hosts»

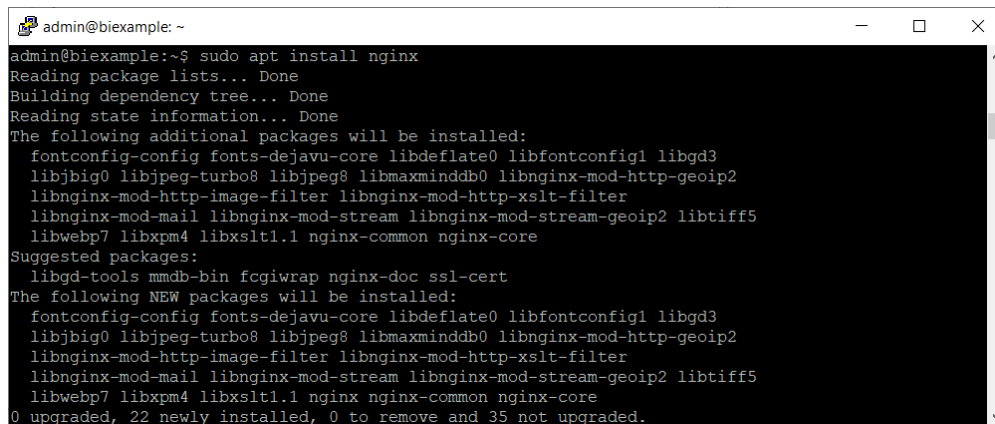
4.4.2 Установка и настройка сервера «NGINX»

1. Обновить списки пакетов операционной системы из депозитариев:

```
sudo apt update
```

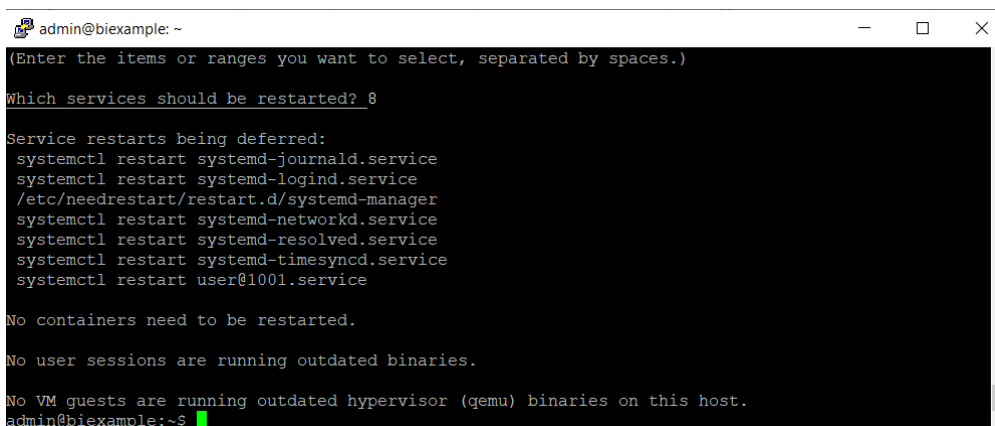
2. Установить Nginx (пример выполнения команды см. Рисунок и Рисунок 693869):

```
sudo apt install nginx
```



```
admin@biexample: ~  
admin@biexample:~$ sudo apt install nginx  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  fontconfig-config fonts-dejavu-core libdeflate0 libfontconfig1 libgd3  
  libjbig0 libjpeg-turbo8 libjpeg8 libmaxminddb0 libnginx-mod-http-geoip2  
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter  
  libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geoip2 libtiff5  
  libwebp7 libxpm4 libxslt1.1 nginx-common nginx-core  
Suggested packages:  
  libgd-tools mmdns-bin fcgiwrap nginx-doc ssl-cert  
The following NEW packages will be installed:  
  fontconfig-config fonts-dejavu-core libdeflate0 libfontconfig1 libgd3  
  libjbig0 libjpeg-turbo8 libjpeg8 libmaxminddb0 libnginx-mod-http-geoip2  
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter  
  libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geoip2 libtiff5  
  libwebp7 libxpm4 libxslt1.1 nginx nginx-common nginx-core  
0 upgraded, 22 newly installed, 0 to remove and 35 not upgraded.
```

Рисунок 68 – Пример выполнения команды установки Nginx. Начало

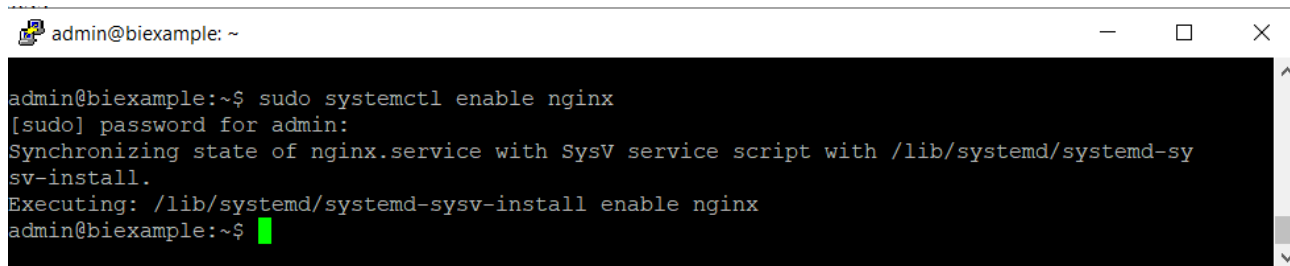


```
admin@biexample: ~  
(Enter the items or ranges you want to select, separated by spaces.)  
Which services should be restarted? 8  
Service restarts being deferred:  
systemctl restart systemd-journald.service  
systemctl restart systemd-logind.service  
/etc/needrestart/restart.d/systemd-manager  
systemctl restart systemd-networkd.service  
systemctl restart systemd-resolved.service  
systemctl restart systemd-timesyncd.service  
systemctl restart user@1001.service  
No containers need to be restarted.  
No user sessions are running outdated binaries.  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
admin@biexample:~$
```

Рисунок 6938 – Пример выполнения команды установки Nginx. Окончание

3. Добавить программу Nginx в автозагрузку (пример выполнения команды см. Рисунок 39):

```
sudo systemctl enable nginx
```



```
admin@biexample: ~  
admin@biexample:~$ sudo systemctl enable nginx  
[sudo] password for admin:  
Synchronizing state of nginx.service with SysV service script with /lib/systemd/systemd-sv-install.  
Executing: /lib/systemd/systemd-sv-install enable nginx  
admin@biexample:~$
```

Рисунок 39 – Пример выполнения команды добавления программы Nginx в автозагрузку

4. Выложить (заранее подготовленные) ключи (full.crt и key-decrypted.key) для работы протокола «HTTPS» в один из каталогов сервера (пример размещения файлов см. Рисунок 40).

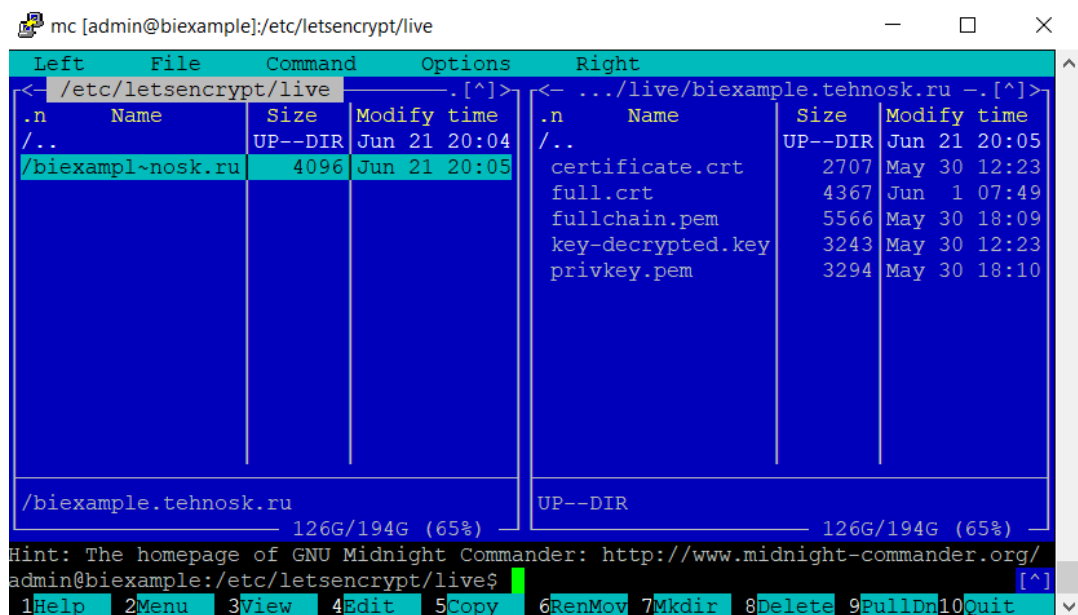


Рисунок 40 – Пример размещения файлов – ключей на сервере

5. Перейти в каталог «conf.d» по пути `/etc/nginx/conf.d` (пример выполнения команды см. Рисунок 724172):

```
cd /etc/nginx/conf.d
```



Рисунок 7241 – Пример выполнения команды по переходу в каталог «/etc/nginx/conf.d»

6. Создать в каталоге «/etc/nginx/conf.d» файл конфигурации «nginx_smprom.conf» и в нем указать актуальное имя сервера и пути к ключам для работы протокола «HTTPS» в соответствии с представленным ниже примером (пример корректировки файла см. Рисунок 734273 и Рисунок 4374):

```
server {
    if ($host = {%SERVERNAME%}) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    listen 80;
    listen [::]:80;
    server_name {%SERVERNAME%};
    return 301 https://$server_name$request_uri;
}

server {
    server_name {%SERVERNAME%};
    location / {
        proxy_set_header    Host                $host;
        proxy_set_header    X-Real-IP            $remote_addr;
        proxy_set_header    X-Forwarded-Host     $host;
        proxy_set_header    X-Forwarded-Server   $host;
```

```

        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto "https";
        proxy_pass http://{%SERVERIP%}:8088/;
        client_max_body_size 10M;
        proxy_read_timeout 300;
        proxy_connect_timeout 300;
        proxy_send_timeout 300;
    }

listen 443 ssl; # managed by Certbot
ssl_certificate /etc/letsencrypt/live/{%SERVERNAME%}/full.crt;
ssl_certificate_key /etc/letsencrypt/live/{%SERVERNAME%}/key-decrypted.key;
}

```

, где:

{%SERVERNAME%} – Доменное имя сервера.
 {%SERVERIP%} – IP-адрес сервера.



Рисунок 7342 – Пример выполнения команды открытия файла «nginx_bismpom.conf»

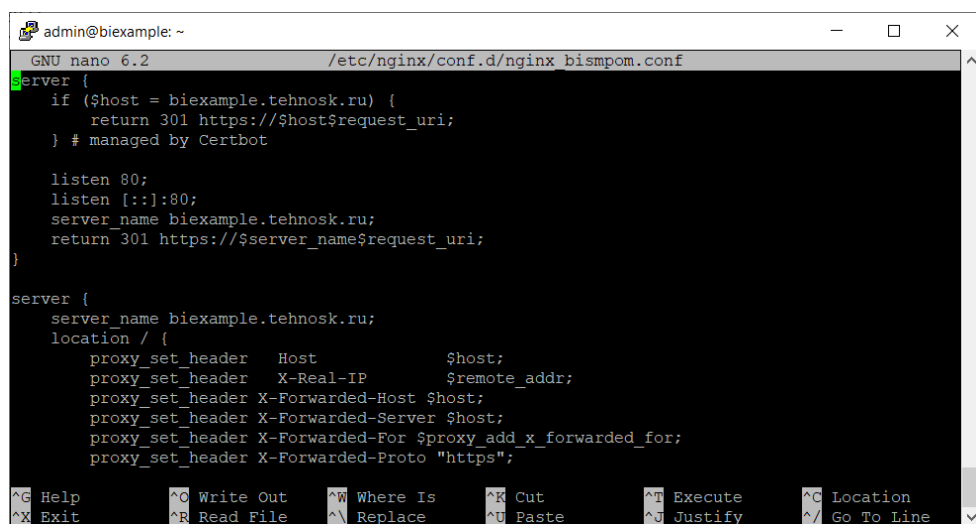


Рисунок 43 – Пример корректировки файла «nginx_bismpom.conf»

7. Запустить Nginx (пример выполнения команды см. Рисунок 754475):

```
sudo service nginx start
```



Рисунок 7544 – Пример выполнения команды запуска программы Nginx

8. Проверить статус работы Nginx (пример выполнения команды см. Рисунок 7645):

```
sudo service nginx status
```


6. Установить Docker.

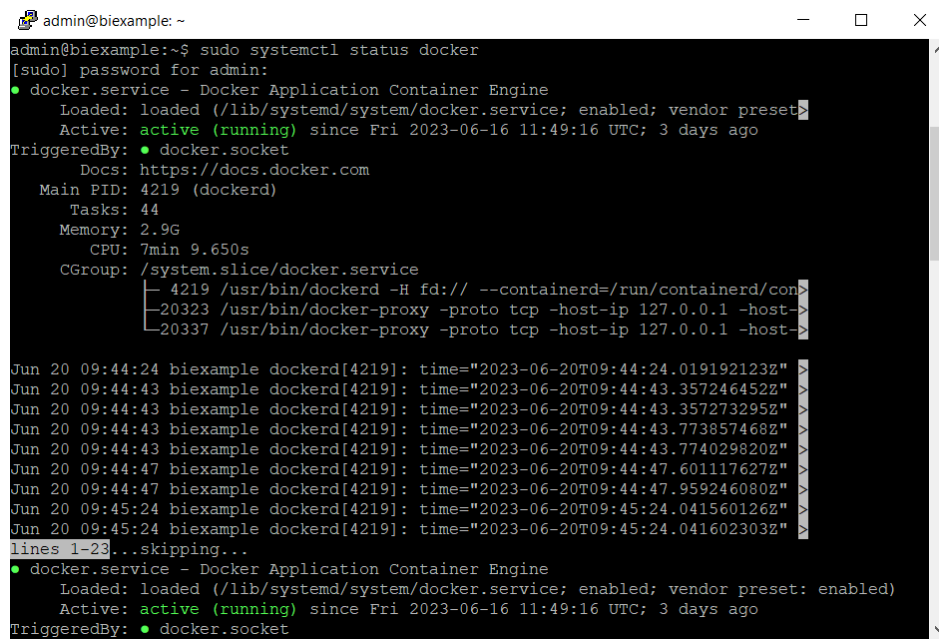
```
sudo apt install docker-ce
```

7. Проверить результат установки Docker (пример выполнения команды см. Рисунок 774677):

```
sudo systemctl status docker
```

Должны увидеть следующий результат выполнения команды:

```
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset:
   Active: active (running) since Fri 2022-04-01 21:30:25 UTC; 22s ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 7854 (dockerd)
      Tasks: 7
     Memory: 38.3M
        CPU: 340ms
    CGroup: /system.slice/docker.service
            └─7854 /usr/bin/dockerd -H fd:// --
containerd=/run/containerd/containerd.sock
. . . . .
```



```
admin@biexample: ~
admin@biexample:~$ sudo systemctl status docker
[sudo] password for admin:
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset:
   Active: active (running) since Fri 2023-06-16 11:49:16 UTC; 3 days ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 4219 (dockerd)
      Tasks: 44
     Memory: 2.9G
        CPU: 7min 9.650s
    CGroup: /system.slice/docker.service
            └─ 4219 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/com
            └─20323 /usr/bin/docker-proxy -proto tcp -host-ip 127.0.0.1 -host-
            └─20337 /usr/bin/docker-proxy -proto tcp -host-ip 127.0.0.1 -host-
Jun 20 09:44:24 biexample dockerd[4219]: time="2023-06-20T09:44:24.019192123Z"
Jun 20 09:44:43 biexample dockerd[4219]: time="2023-06-20T09:44:43.357246452Z"
Jun 20 09:44:43 biexample dockerd[4219]: time="2023-06-20T09:44:43.357273295Z"
Jun 20 09:44:43 biexample dockerd[4219]: time="2023-06-20T09:44:43.773857468Z"
Jun 20 09:44:43 biexample dockerd[4219]: time="2023-06-20T09:44:43.774029820Z"
Jun 20 09:44:47 biexample dockerd[4219]: time="2023-06-20T09:44:47.601117627Z"
Jun 20 09:44:47 biexample dockerd[4219]: time="2023-06-20T09:44:47.959246080Z"
Jun 20 09:45:24 biexample dockerd[4219]: time="2023-06-20T09:45:24.041560126Z"
Jun 20 09:45:24 biexample dockerd[4219]: time="2023-06-20T09:45:24.041602303Z"
lines 1-23...skipping...
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset:
   Active: active (running) since Fri 2023-06-16 11:49:16 UTC; 3 days ago
   TriggeredBy: ● docker.socket
```

Рисунок 7746 – Пример выполнения команды проверки статуса Docker

8. Запустить контейнер:

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

Должны увидеть следующий результат выполнения команды:

```
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:bfe6a278a0a267fad2634554f4f0c6f31981eea41c553fdf5a83e95a41d40c38
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
```

This message shows that your installation appears to be working correctly.

.

9. Добавить текущего пользователя в группу docker (пример выполнения команды см. Рисунок 78):

```
sudo usermod -aG docker {%USER%}
```

, где:

{%USER%} – Пользователь операционной системы текущей сессии.

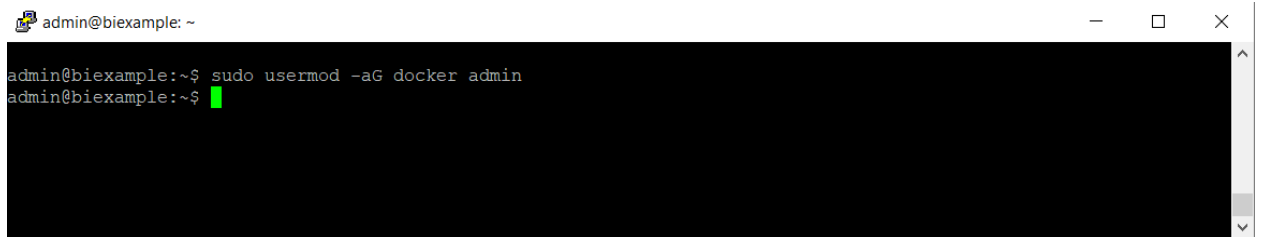


Рисунок 78 – Пример выполнения команды добавления текущего пользователя в группу docker

10. Перелогинится и повторно запустить контейнер:

```
docker run hello-world
```

4.4.4 Установка Docker-compose

1. Создать каталог (пример выполнения команды см. Рисунок 8):

```
sudo mkdir -p ~/.docker/cli-plugins/
```

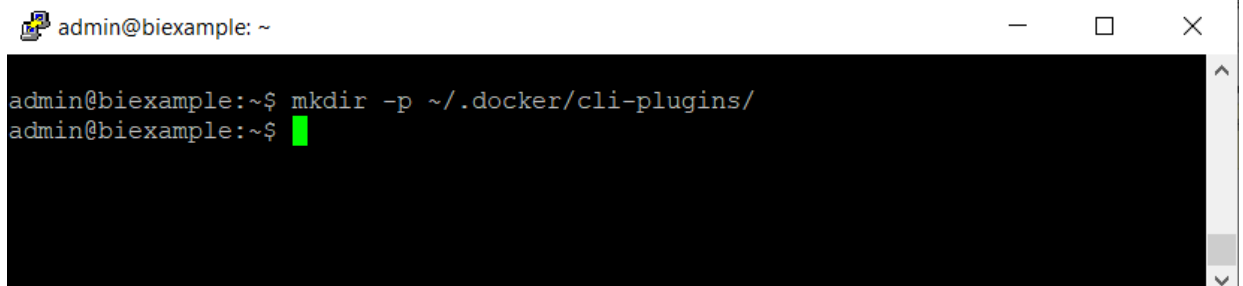


Рисунок 78 – Пример выполнения команды создания каталога

2. Устанавливаем Docker-compose (пример выполнения команды см. Рисунок79):

```
sudo curl -SL https://github.com/docker/compose/releases/download/v2.14.2/docker-  
compose-linux-x86_64 -o ~/.docker/cli-plugins/docker-compose
```

```
admin@biexample: ~  
admin@biexample:~$ sudo curl -SL https://github.com/docker/compose/releases/download/v2.14.2/docker-compose-linux-x86_64 -o ~/.docker/cli-plugins/docker-compose  
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current  
           Dload  Upload  Total   Spent    Left     Speed  
0         0     0     0      0      0      0      0      0  --:--:-- --:--:-- --:--:--    0  
100 42.8M  100 42.8M    0     0  284k      0  0:02:34  0:02:34 --:--:-- 309k  
admin@biexample:~$
```

Рисунок 79 – Пример выполнения команды установки Docker-compose

3. Сделать файл запускаемым (пример выполнения команды см. Рисунок 8047):

```
sudo chmod +x ~/.docker/cli-plugins/docker-compose
```

```
admin@biexample: ~  
admin@biexample:~$ sudo chmod +x ~/.docker/cli-plugins/docker-compose  
admin@biexample:~$
```

Рисунок 8047 – Пример выполнения команды по изменению атрибутов файла

4. Проверить работоспособность Docker-compose (пример выполнения команды см. Рисунок 814881):

```
docker compose version
```

Должны увидеть следующий результат выполнения команды:

```
Docker Compose version v2.14.2
```

```
admin@biexample: ~  
admin@biexample:~$ docker compose version  
Docker Compose version v2.14.2  
admin@biexample:~$
```

Рисунок 8148 – Пример выполнения команды определения версии Docker-compose

4.4.5 Установка Apache Superset

1. Обновить существующий список пакетов.

```
sudo apt update
```

2. Распаковать архив «superset.tar.gz» (из состава дистрибутива «СМПО М») по пути home/ (пример выполнения команды см. Рисунок 49):

```
tar xvzf superset.tar.gz
```

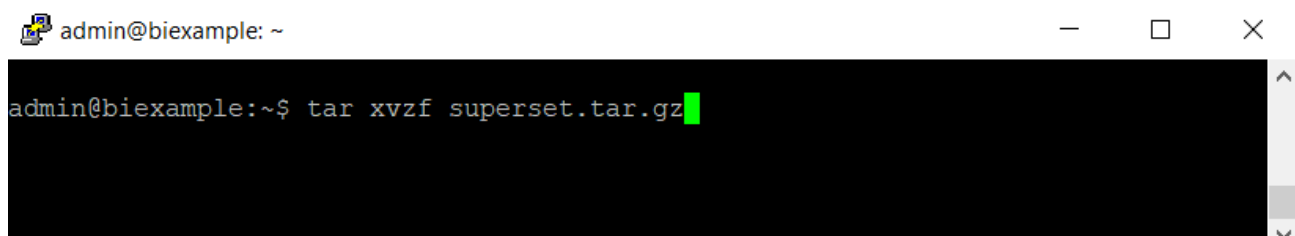


Рисунок 49 – Пример выполнения команды распаковки архива

3. Перейти в каталог установки программного обеспечения (пример выполнения команды см. Рисунок 5083):

```
cd /home/superset/superset/
```

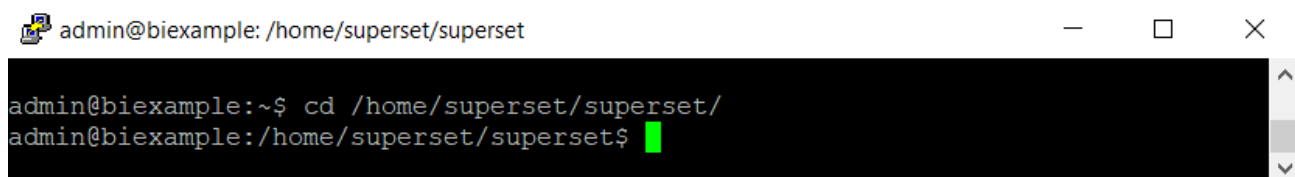


Рисунок 50 – Пример выполнения команды перехода в каталог

4. Запустить программное обеспечение «Apache Superset» из контейнера (пример выполнения команды см. Рисунок 845184):

```
sudo docker compose -f docker-compose.yml up
```

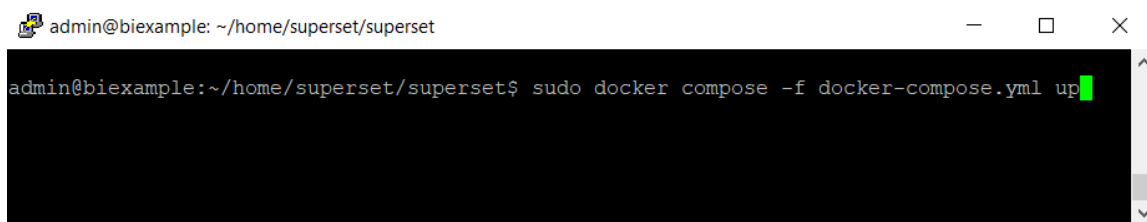


Рисунок 8451 – Пример выполнения команды запуска программного обеспечения «Apache Superset» из контейнера

```
admin@biexample: ~/home/superset/superset
[+] Running 7/9ling 14.1s
.: redis 6 layers [ ] 0B/0B Pulling 22.6s
.: superset-tests-worker Pulling 22.6s
.: db 12 layers [ ] 0B/0B Pulling 22.6s
.: superset Pulling 22.6s
! superset-websocket Warning 3.5s
.: superset-worker-beat Pulling 22.5s
.: superset-init 20 layers [.,#####.##] 30.85MB/124.8MB Pulling 22.5s
.: superset-worker Pulling 22.5s
.: superset-node 8 layers [ ] 0B/0B Pulling 22.5s
```

Рисунок 8552 – Пример выполнения команды запуска программного обеспечения «Apache Superset» из контейнера. Продолжение

4.4.6 Проверка работоспособности сервера бизнес-аналитики Apache Superset

Для проверки работоспособности сервера бизнес-аналитики Apache Superset необходимо запустить Web-клиент Apache Superset.

Для запуска Web-клиента Apache Superset необходимо открыть Web-браузер и в адресной строке ввести запрос к серверу вида:

```
https://{DOMAINBISERVERNAME%}
```

, где:

{DOMAINBISERVERNAME%} – Доменное имя сервера бизнес-аналитики

При успешном подключении к серверу бизнес-аналитики Apache Superset откроется окно авторизации для входа в систему бизнес-аналитики (см. Рисунок 865386).

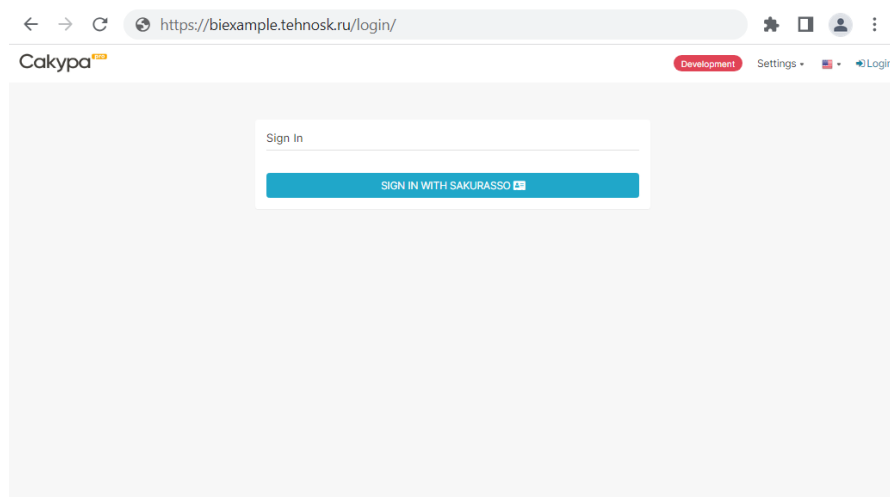


Рисунок 8653 – Пример внешнего подключения к серверу бизнес-аналитики Apache Superset

При успешной авторизации осуществляется вход в систему бизнес-аналитики (см. Рисунок 875487).

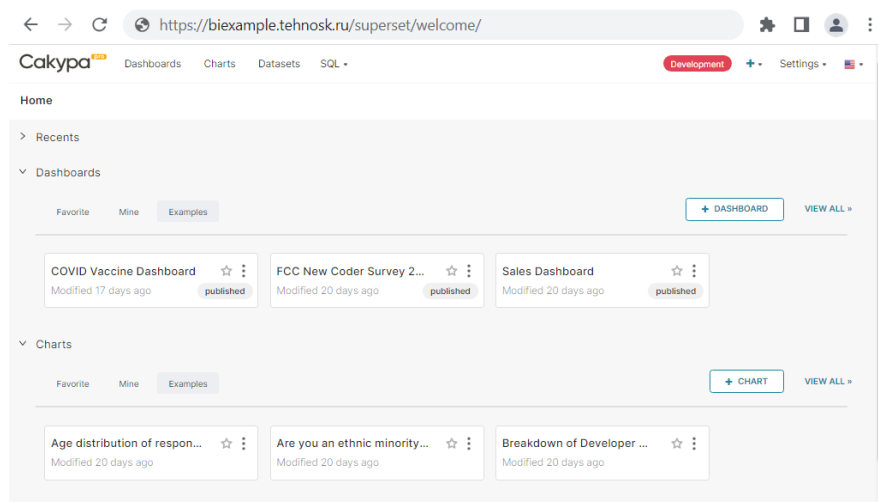


Рисунок 8754 – Пример внешнего подключения к серверу бизнес-аналитики Apache Superset

4.4.7 Импорт (загрузка) на сервер бизнес-аналитики Apache Superset шаблонов дашбордов из состава дистрибутива системы

После установки и проверки работоспособности сервера бизнес-аналитики Apache Superset необходимо с использованием Web-клиент Apache Superset открыть реестр дашбордов сервера бизнес-аналитики Apache Superset и выполнить импорт (загрузку) на сервера бизнес-аналитики Apache Superset базовый набор дашбордов из состава дистрибутива системы.

Для этого необходимо в интерфейсе сервера бизнес-аналитики Apache Superset выбрать раздел «Dashboards» и далее в правом верхнем углу нажать на кнопку «Import dashboard» (см. 88) - откроется окно импорта (см. Рисунок 89559).

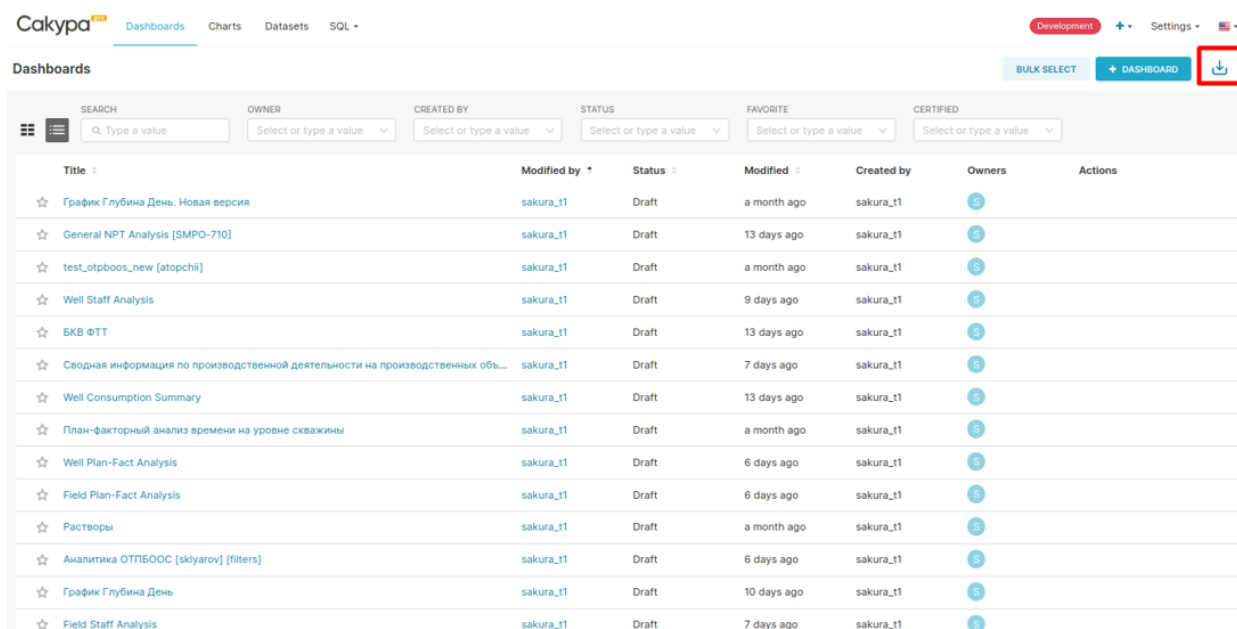


Рисунок 88 – Реестр дашбордов сервера бизнес-аналитики Apache Superset

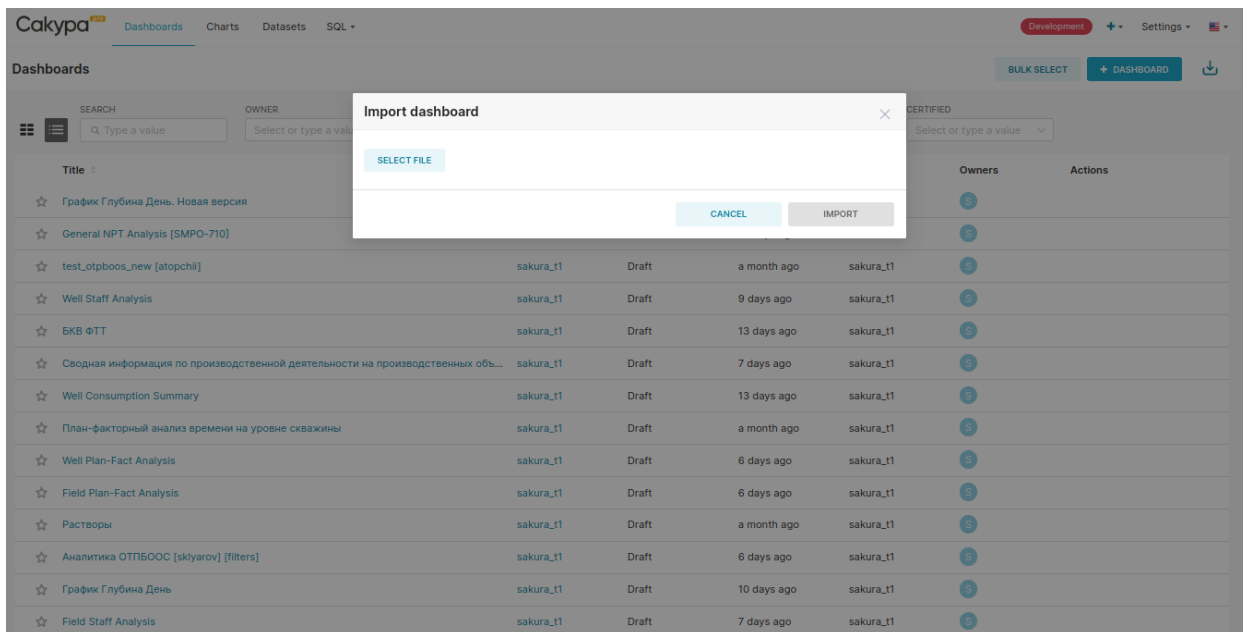


Рисунок 8955 – Реестр дашбордов сервера бизнес-аналитики Apache Superset. Импорт дашборда

В окне импорта необходимо нажать на кнопку «Select File» (см. Рисунок 895589) – откроется окно выбора файла для импорта (см. Рисунок 560). Необходимо выбрать один файл – шаблон дашборда и нажать на кнопку «Select». Далее в окне импорта необходимо нажать на кнопку «Import» (см. Рисунок 895589) и дождаться импорта (загрузки) нового дашборда в реестре дашбордов (см. Рисунок 88).

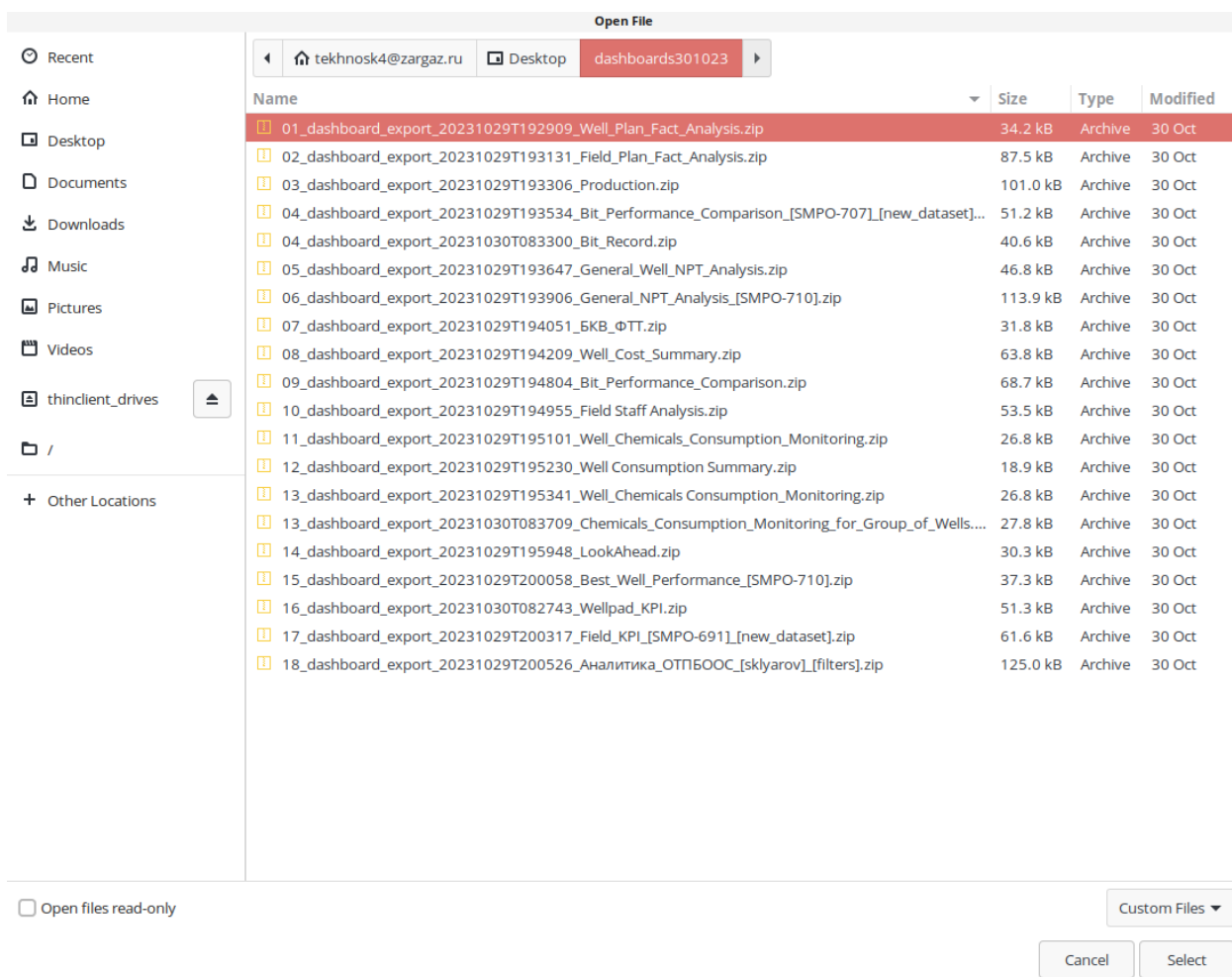


Рисунок 560 – Реестр дашбордов сервера бизнес-аналитики Apache Superset.
Окно выбора файла шаблона дашборда

Необходимо последовательно повторить данную операцию для всех шаблонов дашбордов из состава дистрибутива системы.

4.4.8 Настройка связи сервера бизнес-аналитики Apache Superset с сервером СУБД PostgreSQL

Для настройки связи сервера бизнес-аналитики Apache Superset с сервером СУБД PostgreSQL необходимо с использованием Web-клиент Apache Superset открыть реестр соединений с базами данных.

Для этого необходимо в интерфейсе сервера бизнес-аналитики Apache Superset выбрать раздел «Dashboards», далее в правом верхнем углу открыть выпадающее меню «Settings» и в нем найти и нажать на пункт «Database Connections» (см. Рисунок 957) - откроется реестр соединений с базами данных (см. Рисунок 58).

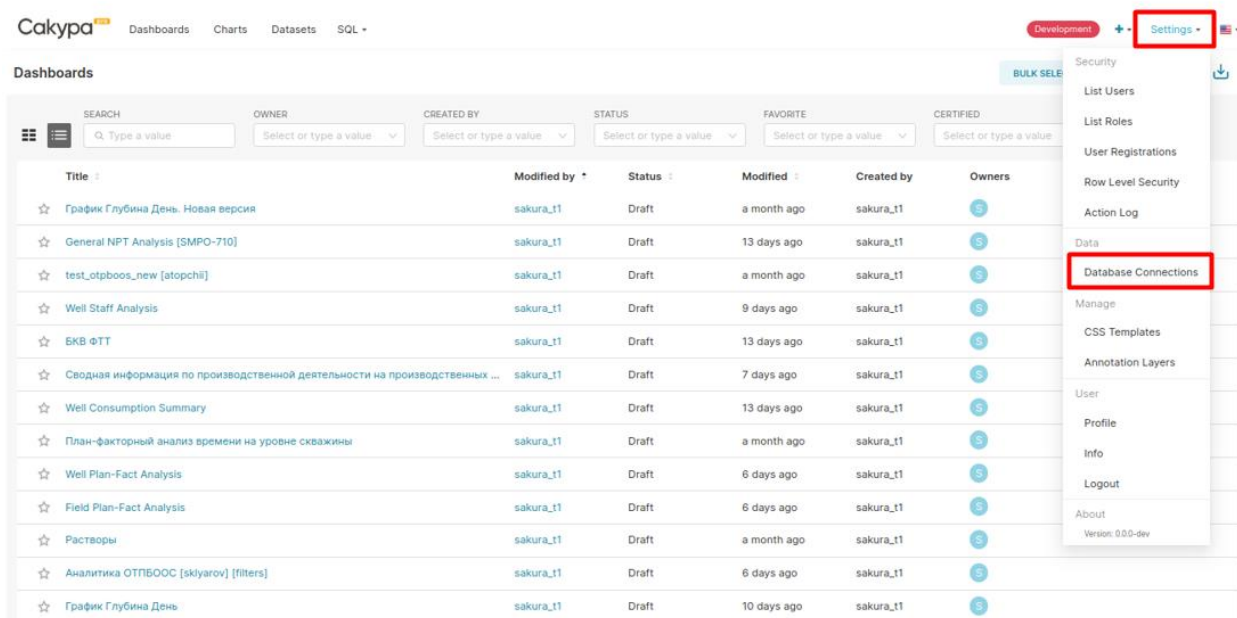


Рисунок 957 – Реестр дашбордов сервера бизнес-аналитики Apache Superset.
Переход к реестру соединений с базами данных

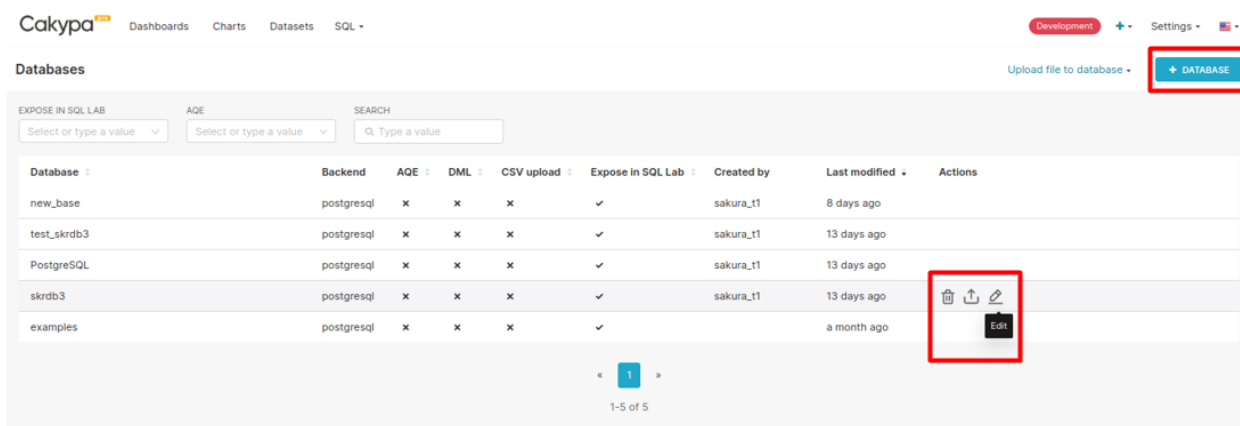


Рисунок 58 – Реестр соединений с базами данных сервера бизнес-аналитики Apache Superset

В реестре соединений с базами данных есть возможность создавать новые соединения и редактировать существующие соединения. При развертывании системы создавать новые соединения не требуются, они будут созданы автоматически при импорте (загрузке) шаблонов дашбордов (см. подраздел 4.4.7 Импорт (загрузка) на сервер бизнес-аналитики Apache Superset шаблонов дашбордов из состава дистрибутива системы). Однако после импорта (загрузки) шаблонов будут созданы соединения с базами данных, которые были использованы разработчиками при разработке дашбордов и для таких соединений необходимо проверить и настроить параметры этих соединений на работу с актуальной базой данных, которая используется для развертывания данного экземпляра системы.

Для этого необходимо в интерфейсе сервера бизнес-аналитики Apache Superset в реестре соединений с базами данных по каждому существующему соединению выбрать учетную запись соединения и нажать на кнопку «Edit» (см. Рисунок 58) – откроется окно ввода и редактирования параметров соединения с базой данных (см. Рисунок 59).

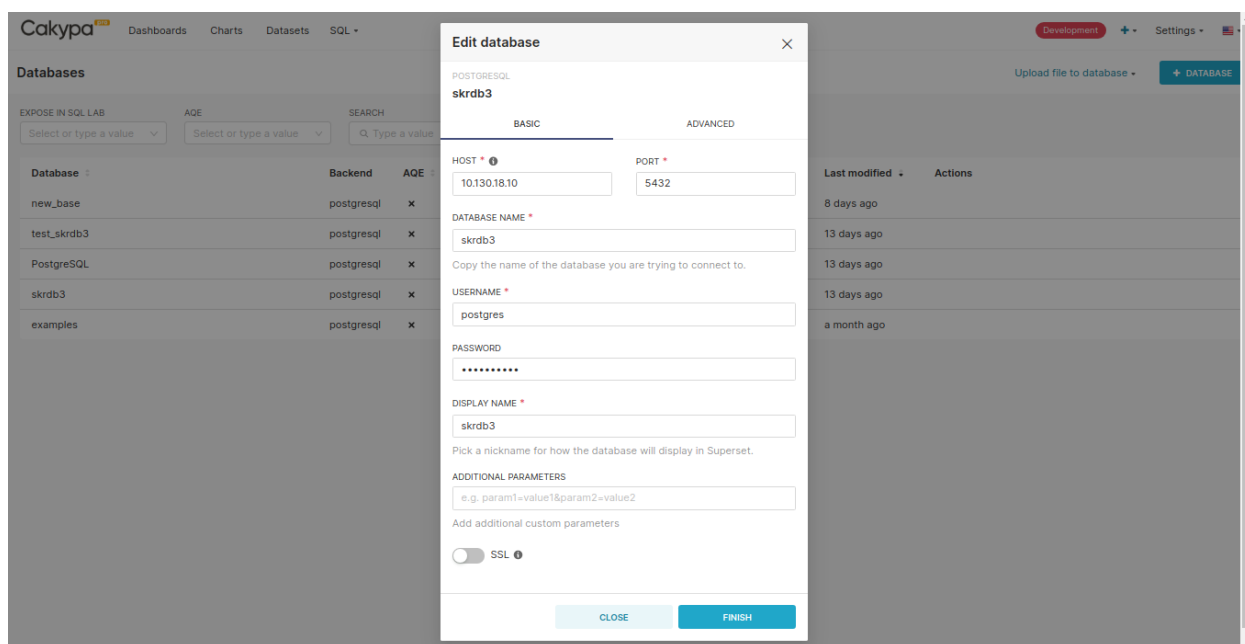


Рисунок 59 – Окно настройки параметров связи сервера бизнес-аналитики Apache Superset с сервером СУБД PostgreSQL. Вариант 1

В окне ввода и редактирования параметров соединения с базой данных необходимо проверить и при необходимости ввести (скорректировать) следующие параметры:

- HOST – ip адрес сервера СУБД PostgreSQL;
- PORT – порт сервера СУБД PostgreSQL;
- DATABASE NAME – имя базы данных системы;
- USERNAME – имя пользователя базы данных системы;
- PASSWORD – пароль пользователя базы данных системы.

Для сохранения данных необходимо нажать на кнопку «Finish» (см. Рисунок 59).

Для ряда соединений параметры задаются в виде единой строки в формате «postgresql+psycopg2://postgres:XXXXXXXXXX@10.130.18.10:5432/skrdb3» с указанием ip адрес сервера СУБД PostgreSQL и порт сервера СУБД PostgreSQL (см. Рисунок 94).

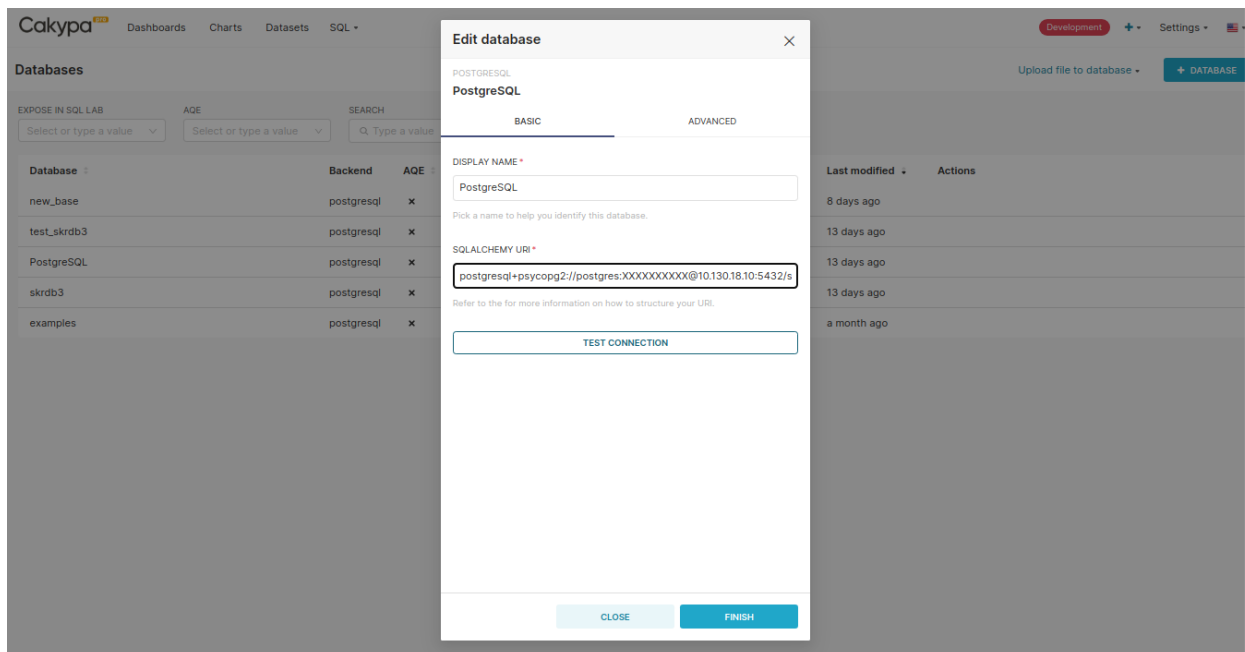


Рисунок 94– Окно настройки параметров связи сервера бизнес-аналитики Apache Superset с сервером СУБД PostgreSQL. Вариант 2

5 ПОРЯДОК ЗАПУСКА, МОНИТОРИНГА И ОСТАНОВКИ СИСТЕМНОГО И ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

5.1 Порядок запуска системного и прикладного программного обеспечения

Для корректной работы «СМПО М» необходимо придерживаться строгой последовательности запуска системного и прикладного программного обеспечения. Так в первую очередь необходимо осуществить запуск программного обеспечения, входящего в состав Подсистемы хранения данных (ПХД) «СМПО М», вторым должен быть запущен сервер доступа, контроля и обработки данных (микросервисов), а уже потом возможно запускать автоматизированные рабочие места пользователей «СМПО М». В таблице 3 приведен список программного обеспечения и способы его запуска.

Таблица 3 – Порядок запуска системного и прикладного программного обеспечения «СМПО М»

№ п/п	Наименование программного обеспечения	Описание процесса запуска программного обеспечения
1	2	3
1	Подсистема хранения данных (ПХД) «СМПО М»	
1.1	СУБД PostgreSQL	Запуск осуществляется выполнением команды: «service postgresql start»
2.	Сервер доступа, контроля и обработки данных (микросервисов) «СМПО М»	
2.1	Сервер микросервисов	Запуск осуществляется выполнением команды: «docker-compose up --build -d»
3.	Рабочее место пользователя «СМПО М»	
3.1	Рабочее место пользователя «СМПО М» уровня центрального подразделения	Для запуска рабочего места пользователя «СМПО М» необходимо на ПЭВМ пользователя открыть Web-браузер и на Портале МКООО «ГАЗПРОМ ИНТЕРНЭШНЛ ЛИМИТЕД» перейти по ссылке к рабочему месту пользователя «СМПО М». Загрузится окно авторизации. В окно авторизации необходимо внести учетные данные пользователя и нажать клавишу «ВОЙТИ». Если учетные данные в окне авторизации были введены верно, то откроется главная страница рабочего места пользователя «СМПО М».

5.2 Порядок мониторинга системного и прикладного программного обеспечения

В таблице 4 приведен перечень программного обеспечения и способы контроля его работоспособности.

Таблица 4 – Способы контроля работоспособности системного и прикладного программного обеспечения «СМПО М»

№ п/п	Наименование ПО	Метод контроля	Средства контроля
1	2	3	4
1	Подсистема хранения данных (ПХД) «СМПО М»		
1.1	СУБД PostgreSQL	Проверка состояния СУБД PostgreSQL осуществляется посредством контроля log-файлов.	log-файл «pg_log» расположен в директории «\$PGDATA/pg_log» или «var/log/postgresql»
2.	Сервер доступа, контроля и обработки данных (микросервисов) «СМПО М»		
2.1	Сервер микросервисов	Проверка состояния микросервисов осуществляется посредством контроля log-файлов.	log-файлы «ms_log» расположены в директории «/var/sakura-cloud/log»

5.3 Порядок остановки системного и прикладного программного обеспечения

В таблице 5 приведен список программного обеспечения и способы его остановки.

Таблица 52 – Порядок остановки системного и прикладного программного обеспечения «СМПО М»

№ п/п	Наименование ПО	Автоматическая остановка Ручная остановка
1	2	3
1.	Сервер доступа, контроля и обработки данных (микросервисов) «СМПО М»	
1.1.	Сервер микросервисов	Остановка осуществляется выполнением команды «docker-compose --stop»
2.	Подсистема хранения данных (ПХД) «СМПО М»	
2.1.	СУБД PostgreSQL	Остановка осуществляется выполнением команды «service postgresql stop»

№ п/п	Наименование ПО	Автоматическая остановка
		Ручная остановка
1	2	3
3.	Рабочее место пользователя «СМПО М»	
3.1.	Рабочее место пользователя «СМПО М»	Для завершения работы на рабочем месте пользователя «СМПО М» необходимо завершить все начатые операции и нажать функциональную клавишу «ВЫХОД». После этого можно закрыть

Приложение А1 Содержимое файлов конфигурации

А1.1 /var/lib/postgresql/12/main/postgresql.conf

```

max_connections = 100
shared_buffers = 128MB
dynamic_shared_memory_type = posix
max_wal_size = 1GB
min_wal_size = 80MB
log_timezone = 'Europe/Moscow'
datestyle = 'iso, mdy'
timezone = 'Europe/Moscow'
lc_messages = 'C.UTF-8'
lc_monetary = 'C.UTF-8'
lc_numeric = 'C.UTF-8'
lc_time = 'C.UTF-8'
default_text_search_config = 'pg_catalog.english'
listen_addresses = '*'
wal_level = replica
logging_collector = on
hot_standby = on
lc_messages = 'C'
wal_keep_segments = 10

```

А1.2 /var/lib/postgresql/12/main/pg_hba.conf

```

# TYPE          DATABASE         USER        ADDRESS            METHOD
# "local" is for Unix domain socket connections only
Local           all             all         trust
# IPv4 local connections:
host            all             all         127.0.0.1/32      trust
# IPv6 local connections:
host            all             all         ::1/128           trust
#Allow replication connections from localhost, by a user with the
#replication privilege
Local           replication     all         trust
host            replication     all         127.0.0.1/32      trust
host            replication     all         ::1/128           trust
host            replication     all         192.168.110.0/24   trust
host            all             all         192.168.110.0/24   trust

```

A1.3 /etc/corosync/corosync.conf

```
totem {
version: 2
cluster_name: pgcluster
transport: knet
crypto_cipher: aes256
crypto_hash: sha256
}
nodelist {
node {
ring0_addr: 192.168.110.200
name: node1
nodeid: 1
}
node {
ring0_addr: 192.168.110.201
name: node2
nodeid: 2
}
node {
ring0_addr: 192.168.110.203
name: node3
nodeid: 3
}
}
quorum {
provider: corosync_votequorum
}
logging { to_logfile: yes
logfile: /var/log/corosync/corosync.logto_syslog: yes
timestamp: on
}
```

ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ

НСИ	–	Нормативно-справочная информация
СМПО М	–	Система мониторинга производственных объектов модернизированная
СУБД	–	Система управления базами данных