Инструкция по установки Системы мониторинга производственных объектов

2024

СОДЕРЖАНИЕ

1 ПОДГОТОВКА К РАБОТЕ

1.1 Подготовка к развертыванию подсистем

Перед развертыванием «САКУРА PRO» необходимо выполнить ряд следующих действий:

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

1.2 Порядок развертывания подсистем

Порядок развертывания подсистем «САКУРА PRO» представлен в таблице 1.

Таблица 1 – Порядок развёртывания подсистем

№ п.п.	Наименование программного обеспечения	Пункт данного руководства
1	**2**	**3**
1. Сервер реляционной СУБД (основной и резервные) (S-03,S-04,S-05)		
1.1	Операционная система Ubuntu	4.1
1.2	СУБД PostgreSQL	4.2
2. Сервер доступа, контроля и обработки данных (микросервисов) (основной и резервный) (S-01,S-02)		
2.1	Операционная система Ubuntu	4.1
2.2	Микросервисы Платформы «САКУРА PRO» (САКУРА PRO)	4.3
3. Сервер бизнес-аналитики (S-06)		
3.1	Операционная система Ubuntu	4.1
3.2	Сервер бизнес-аналитики	4.4
4. Серверы контроллера домена (S-07, S-08)		
4.1	Операционная система MS Windows Server 2012	4.4
4.2	Контроллер домена	4.6
5. Сервер электронной почты (основной и резервный) (S-09, S-10)		

№ п.п.	Наименование программного обеспечения	Пункт данного руководства
1	**2**	**3**
5.1	Операционная система MS Windows Server 2012	4.4
5.2	Почтовый сервер Microsoft Exchange	4.7
	6. ПЭВМ	
6.1	ОС семейства MS Windows или семейства Linux	4.8.1
6.2	Веб-браузер Google Chrome или Mozilla Firefox.	4.8.2

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

2 УСТАНОВКА И НАСТРОЙКА СИСТЕМЫ

2.1 Установка и настройка ОС Ubuntu

Установка ОС Ubuntu 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).

2.1.1 Установка и настройка ОС Ubuntu

1. Подключить образ *.iso (Ubuntu-server 14.04 или выше) из передаваемого инсталляционного пакета в качестве установочного диска к серверу, на котором планируется развернуть сервер приложений. **Внимание! Необходимо установить англоязычную версию операционной системы с поддержкой региональных настроек (locale) «en_US.UTF-8».**

2. Загрузить сервер с установочного диска. Отобразится диалоговое окно языковых параметров процесса установки (см. Рисунок 1).

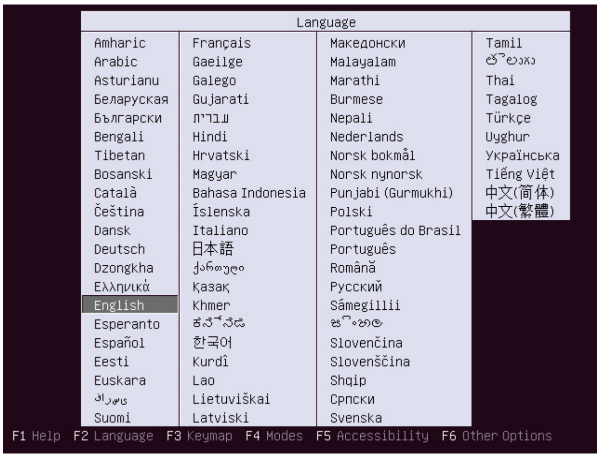


Рисунок 1 – Вид диалогового окна «Выбор языковых параметров процесса установки»

3. Выбрать необходимый язык и нажать клавишу Enter. После этого на экране отобразится диалоговое окно выбора вариантов работы установщика (см. Рисунок 2).

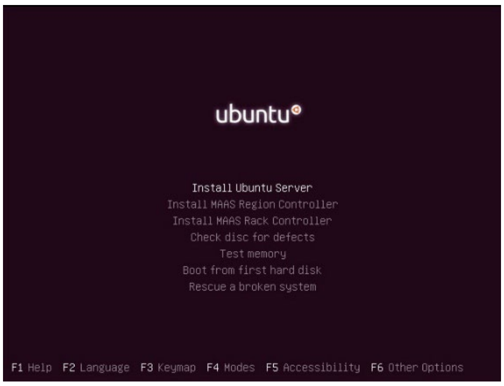


Рисунок 2 – Вид диалогового окна «Выбор вариантов работы установщика»

В диалоговом окне выбрать 1-й пункт (Install Ubuntu Server) и нажать клавишу «Enter» (см. Рисунок 2).

4. Последовательно отобразится ряд диалоговых окон выбора языка интерфейса устанавливаемой системы, местоположения и часового пояса, языковых настроек и раскладки клавиатуры. В указанных окнах необходимо установить соответствующие значения и нажать «Enter» для перехода к следующему этапу установки.

5. Отобразится диалоговое окно ввода сетевого имени сервера. В нем следует ввести сетевое имя сервера и выбрать «<Continue>» (см. Рисунок 3).

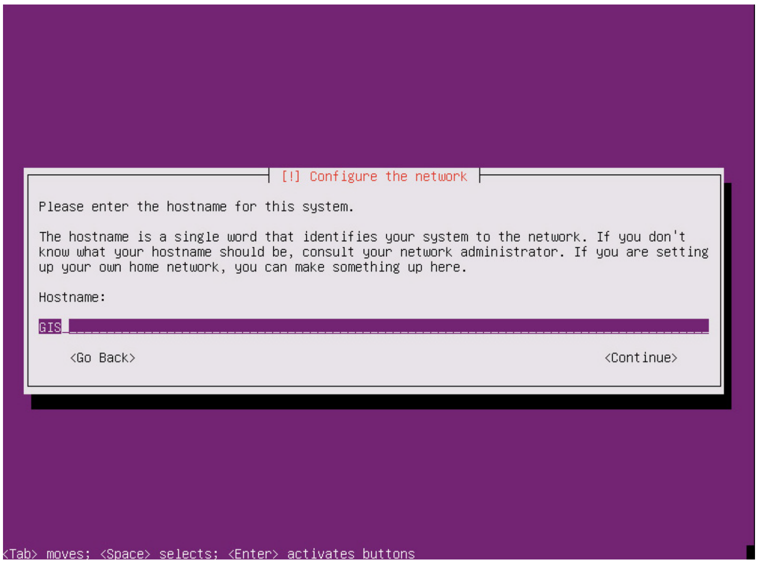


Рисунок 3 – Вид диалогового окна ввода сетевого имени сервера

6. На данном шаге отобразиться ряд последовательных диалоговых окон ввода имени пользователя и пароля и режима шифрования домашнего каталога пользователя. В указанных окнах необходимо внести соответствующие данные и выполнить настройки. Для перехода к следующему этапу нажать клавишу «Enter».

7. Отобразится диалоговое окно подтверждения выбора часового пояса. На данном этапе необходимо подтвердить корректность выбора часового пояса и нажать клавишу «Enter» (см. Рисунок 4).

Рисунок 4 – Вид диалогового окна подтверждения выбора часового пояса

8. Отобразится ряд последовательных диалоговых окон выбора раздела жесткого диска для установки системы. На данном этапе необходимо выбрать соответствующий раздел (разделы) и перейти к следующему шагу.

9. Отобразится диалоговое окно ввода данных о подключении к сети интернет через прокси-сервер. Если прокси-сервер не используется, оставить поле ввода пустым и выбрать «<Continue>» (см. Рисунок 5).

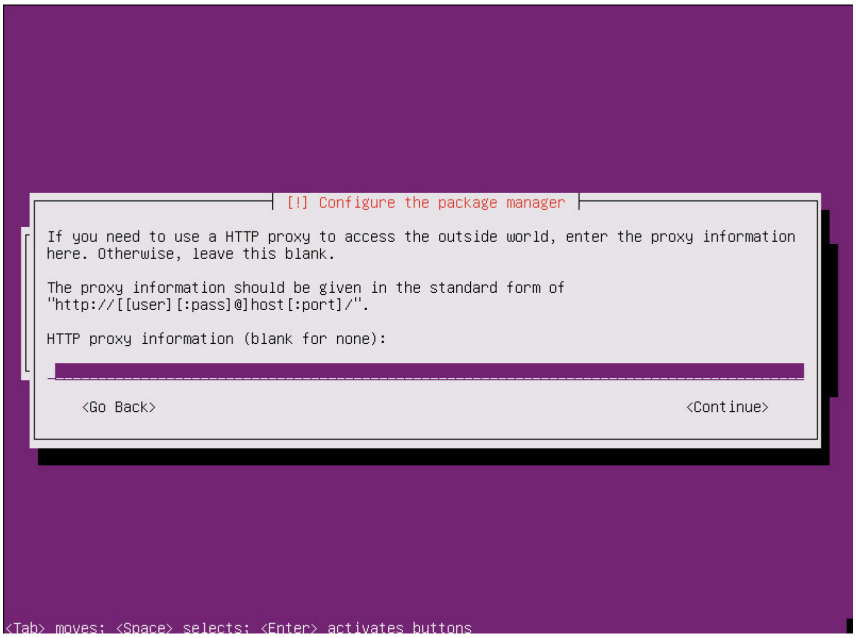


Рисунок 5 – Вид диалогового окна ввода информации о подключении к интернет через прокси-сервер

10. Отобразится диалоговое окно выбора режима обновления системы. Выбрать вариант «No automatic updates» («без автоматических обновлений») (см. Рисунок 6) и перейти к следующему шагу, нажав клавишу «Enter».

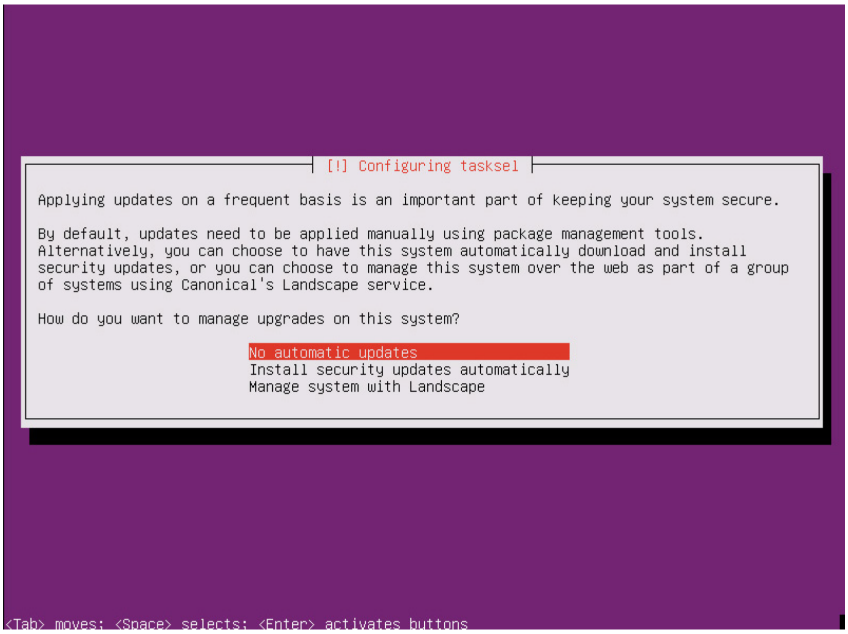


Рисунок 6 – Вид диалогового окна выбора режима обновления системы

11. В окне выбора устанавливаемого дополнительного программного обеспечения необходимо выбрать минимум следующие 2 пункта (см. Рисунок 7):

– PostgreSQL;

– Standard system utilites.

И нажать клавишу «<Continue>» для продолжения процесса установки.

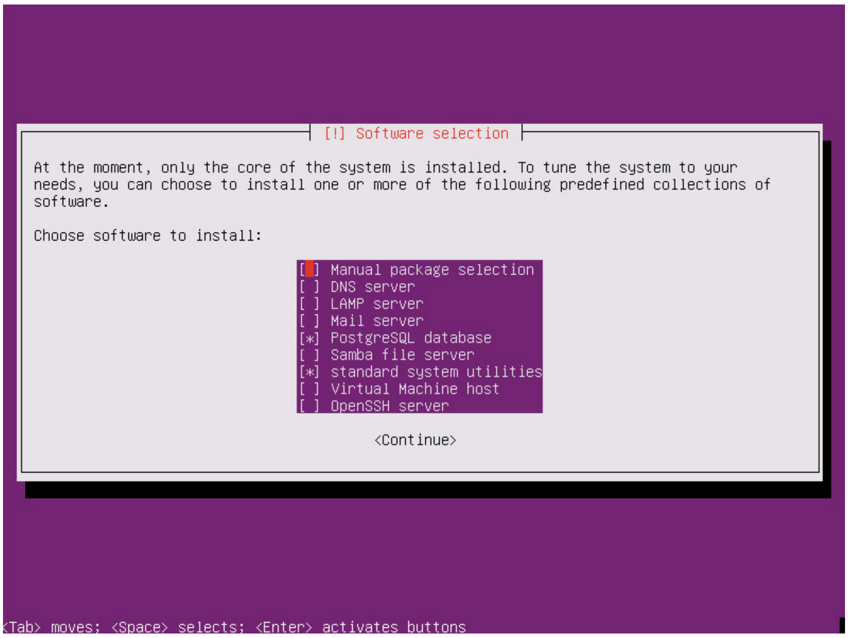


Рисунок 7 – Вид диалогового окна выбора дополнительно устанавливаемого программного обеспечения

12. Отобразится диалоговое окно выбора места установки загрузчика. Выбрать «<Yes>» и нажать клавишу «Enter» для перехода к следующему этапу (см. Рисунок 8).

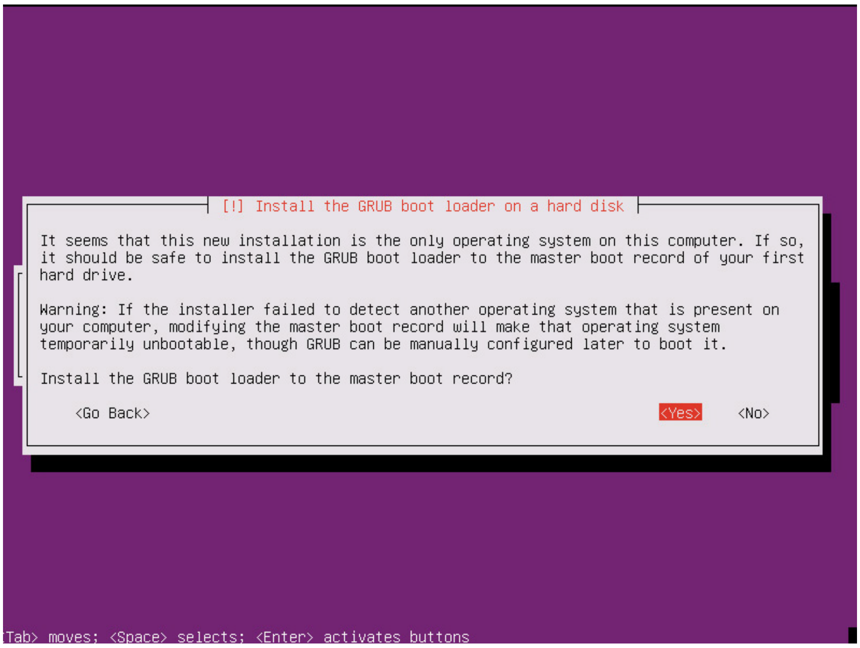


Рисунок 8 – Вид диалогового окна выбора места установки загрузчика

13. Отобразится окно с сообщением об окончании процесса установки системы (см. Рисунок 9). Выбрать «<Continue>», после чего сервер перезагрузится автоматически.

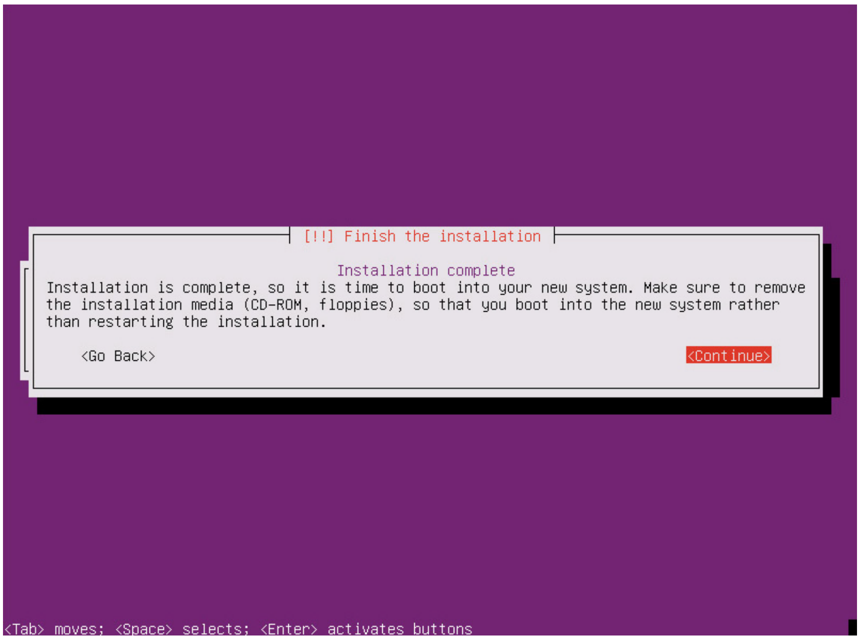


Рисунок 9 – Вид диалогового окна с сообщением об окончании процесса установки

14. После перезагрузки на экране отобразится окно загрузчика с выбором вариантов загрузки системы (см. Рисунок 10). Необходимо выбрать обычный режим. После загрузки операционной системы и ввода логина и пароля пользователя в терминале отобразится сообщение об условиях использования установленного программного обеспечения (см. Рисунок 11).

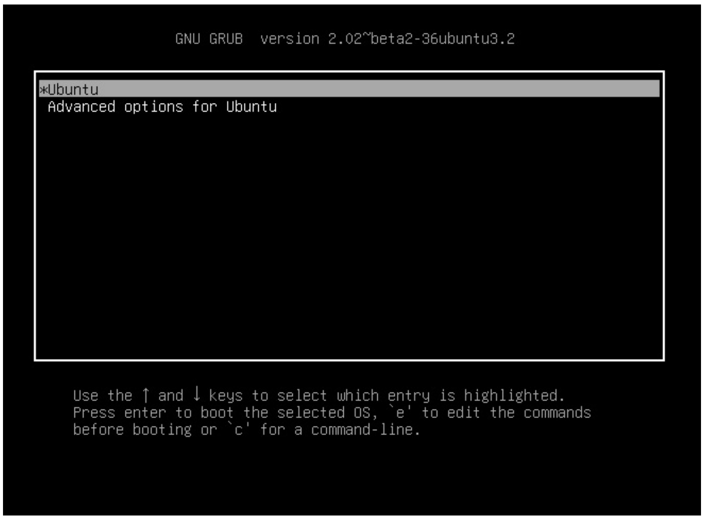


Рисунок 10 – Вид окна загрузчика с отображением вариантов загрузки ОС

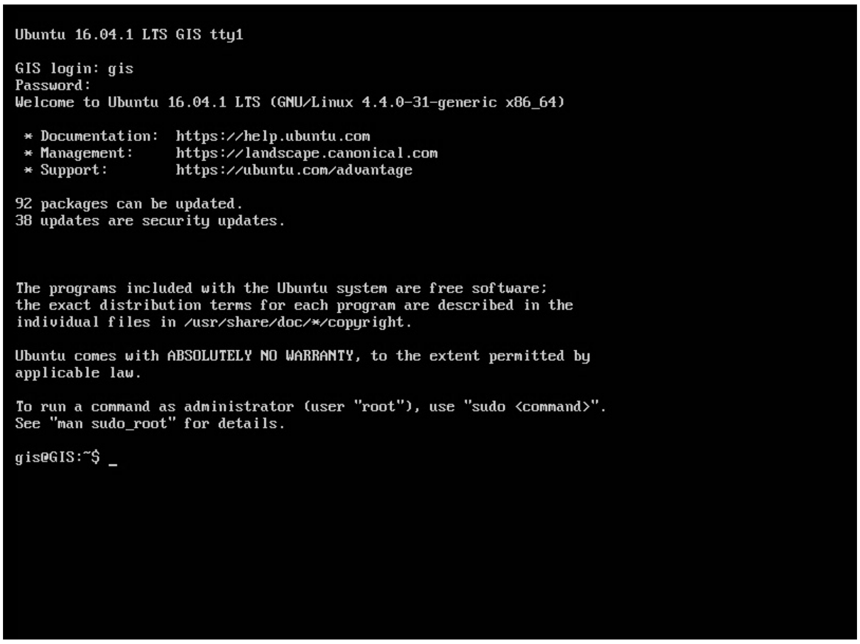


Рисунок 11 – Вид окна терминала после первой загрузки ОС с сообщением об условиях использования установленного программного обеспечения

2.1.2 Схема резервирования ОС Ubuntu

Резервирование операционной системы Ubuntu может осуществляться по одному из следующих сценариев, приведенных в таблице 2.

Таблица 2 – Схема резервирования ОС Ubuntu

Тип	**Резервная единица оборудования**	**Периодичность создания (настройки) резерва**
1	**2**	**3**
Виртуальные серверы	Резервируется системой виртуальных машин, под управлением которой работают данные виртуальные серверы	Резервные ОС с использованием копии виртуальной машины должно осуществляться после каждой модификации ОС.

Тип	Резервная единица оборудования	Периодичность создания (настройки) резерва
1	2	3
		На случай, если модификация оказалась неудачной, рекомендуется хранить две резервные копии виртуальных машин (с предыдущей и настоящей модификациями ОС). Помимо резервных копий виртуальных машин с предустановленным системным и прикладным ПО, возможно создать, хранить и обновлять резервную копию виртуальной машины только с предустановленной ОС.

2.1.3 Схема восстановления ОС Ubuntu

Восстановление операционной системы Ubuntu может осуществляться по одному из следующих сценариев, приведенных в таблице 3.

Таблица 3 – Схема восстановления ОС Ubuntu

Тип	Резервная единица оборудования	Действия по восстановлению
1	2	3
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.

2.2 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. Если на серверах организуется кластер, то необходимо воспользоваться инструкцией из Приложения (см. ПРИЛОЖЕНИЕ А. ИНСТРУКЦИЯ ПО РАЗВЕРТЫВАНИЮ ОТКАЗОУСТОЙЧИВОГО КЛАСТЕРА POSTGRESQL).

2.2.1 Установка СУБД PostgreSQL

СУБД PostgreSQL входит в состав ОС Ubuntu и может быть установлена при установке операционной системы (описание процесса установки ОС Ubuntu приведено в настоящем документе в предыдущем пункте). В случае необходимости установки СУБД PostgreSQL на уже установленной операционной системе используются следующая последовательность команд.

1. Обновить списки пакетов операционной системы из репозиториев (пример выполнения команды см. Рисунок 12):

```
sudo apt update
```

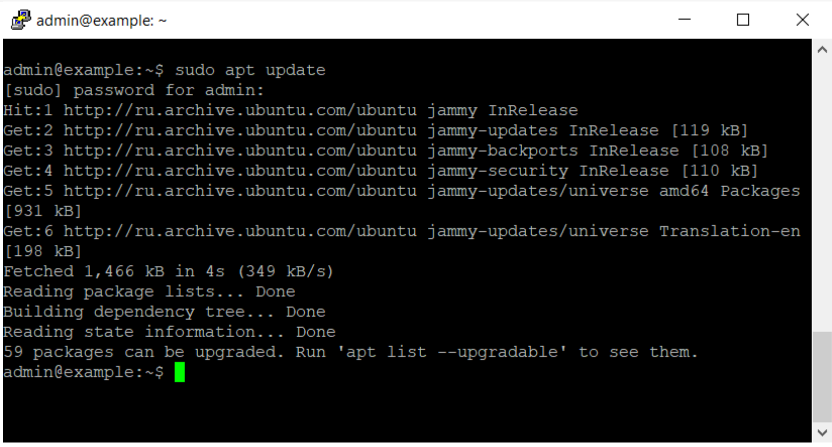


Рисунок 12 – Пример выполнения команды обновления пакетов операционной системы из репозитариев

2. Импортировать ключ GPG в операционную систему (пример выполнения команды см. Рисунок 13):

```
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -
```

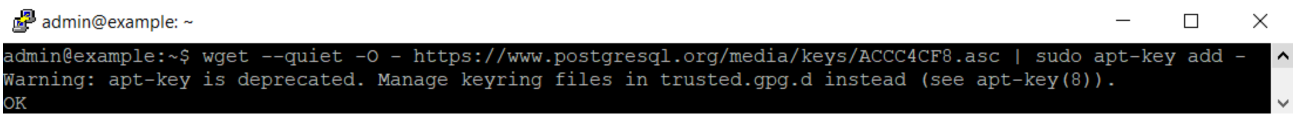


Рисунок 13 – Пример выполнения команды импорта ключа GPG в операционную систему

3. Добавить репозиторий PostgreSQL 12 в операционную систему (пример выполнения команды см. Рисунок 14):

```
echo "deb http://apt.postgresql.org/pub/repos/apt/ `lsb_release -cs`-pgdg main" |sudo tee  /etc/apt/sources.list.d/pgdg.list
```

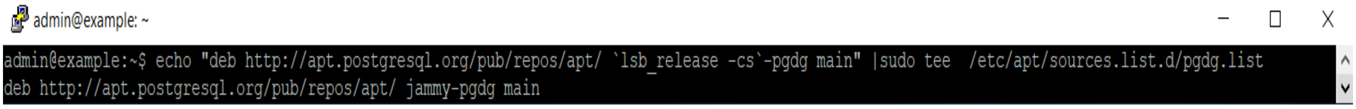


Рисунок 14 – Пример выполнения команды добавления репозитория PostgreSQL 12 в операционную систему

4. Повторно обновить списки пакетов операционной системы из репозиториев (пример выполнения команды см. Рисунок 15):

```
sudo apt update
```

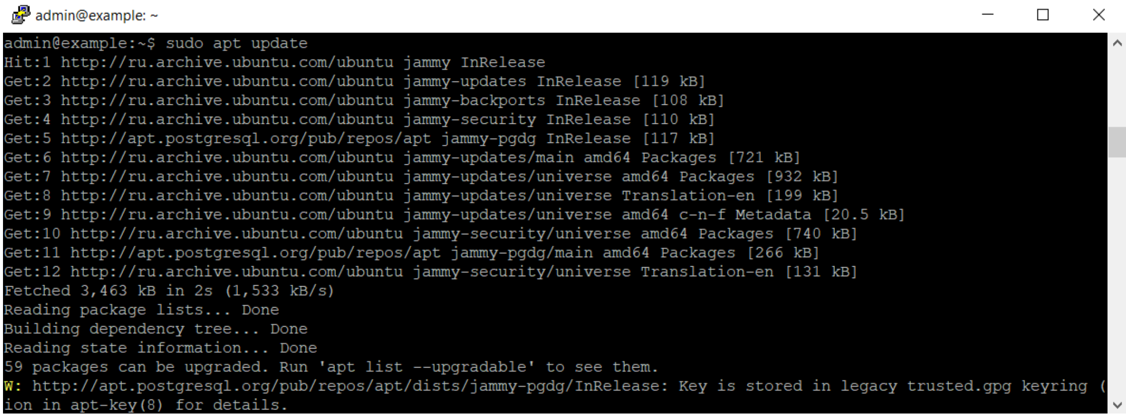


Рисунок 15 – Пример выполнения команды обновления пакетов операционной системы из репозитариев

5. Установить СУБД PostgreSQL (пример выполнения команды см. Рисунок 16 и Рисунок 17):

```
sudo apt -y install postgresql-12 postgresql-client-12
```

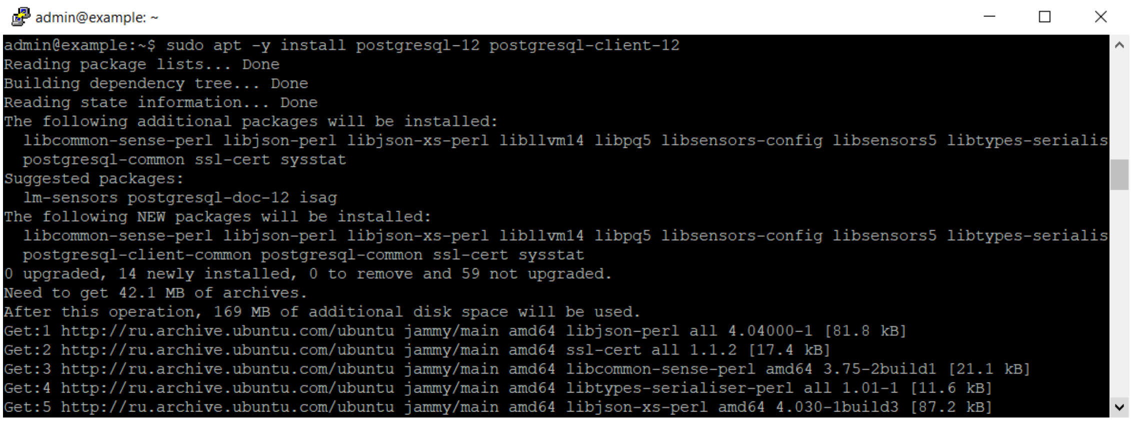


Рисунок 16 – Пример выполнения команды установки СУБД PostgreSQL. Начало

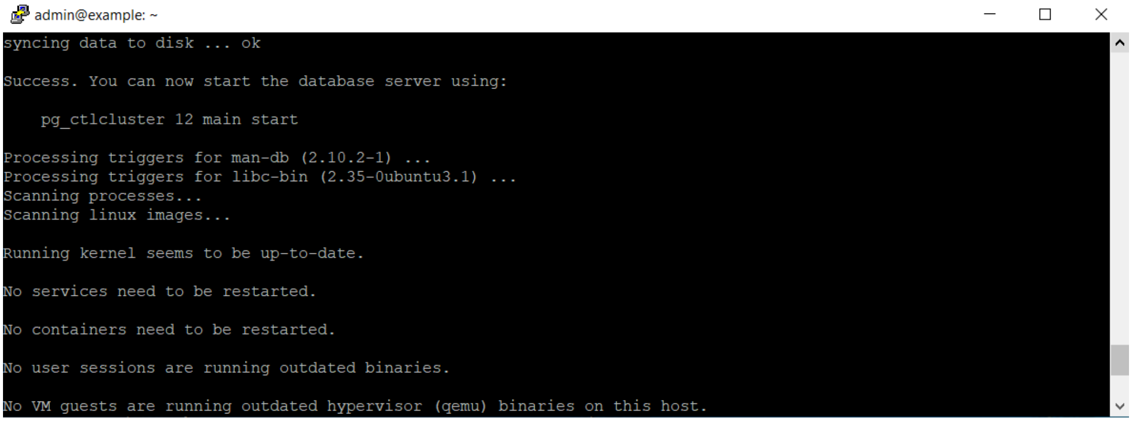


Рисунок 17 – Пример выполнения команды установки СУБД PostgreSQL. Окончание

6. Проверить версию, установленной СУБД PostgreSQL (пример выполнения команды см. Рисунок 18):

```
pg_config --version
```

Рисунок 18 – Пример выполнения команды проверки версии СУБД PostgreSQL

7. Предоставить возможность подключатся к СУБД PostgreSQL локально и удаленно с определенного набора серверов определенной подсети или определенного набора ip-адресов. Выполнить корректировку файла «pg_hba.conf» (пример выполнения команды см. Рисунок 19):

```
sudo nano /etc/postgresql/12/main/pg_hba.conf
```

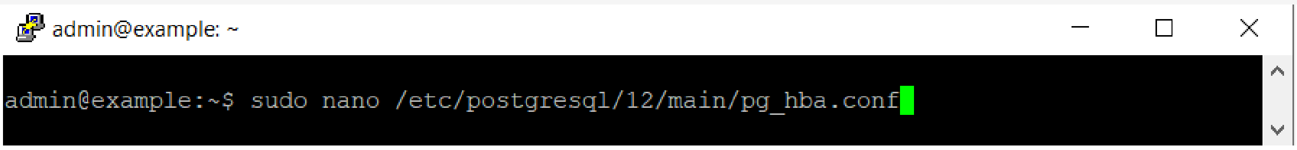


Рисунок 19 – Пример выполнения команды открытия файла «pg_hba.conf»

В файле «pg_hba.conf» (в соответствии со структурой и правилами заполнения данного файла) необходимо (пример файла см. Рисунок 20):

```
# В существующей строке указать «trust», тем самым разрешив локально обращатся
# к СУБД PostgresSQL по IP-адрессу 127.0.0.1 или через localhost
host    all    all    127.0.0.1/32    trust

# В существующей строке указать «trust», тем самым разрешив локально обращаться
# к СУБД PostgresSQL по IP-адрессу 127.0.0.1 или через localhost
host    all    all    ::1/128         trust

# В новой строке указать IP-адрес Сервера доступа, контроля и обработки данных
# (микросервисов) и «trust», тем самым разрешив обращатся к СУБД PostgresSQL
# данному Серверу доступа, контроля и обработки данных САКУРА PRO.
host    all    all    10.55.4.89/16  trust

# По аналогии указать другие IP-адреса Серверов и ПЭВМ, с которых планируется
# обращаться к СУБД PostgreSQL.
```

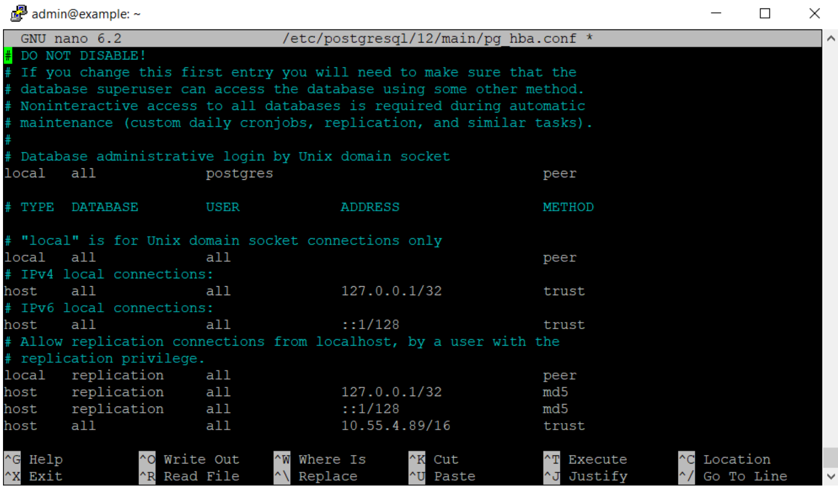


Рисунок 20 – Пример корректировки файла «pg_hba.conf»

8. Указать IP-адрес Сервера СУБД PostgreSQL. Выполнить корректировку файла «postgresql.conf» (пример выполнения команды см. Рисунок 21):

```
sudo nano /etc/postgresql/12/main/postgresql.conf
```

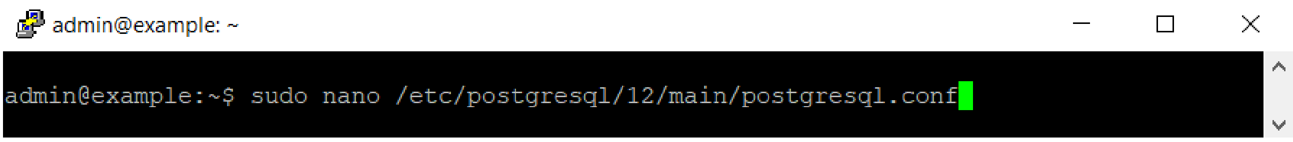


Рисунок 21 – Пример выполнения команды открытия файла «postgresql.conf»

В файле «postgresql.conf» (в соответствии со структурой и правилами заполнения данного файла) необходимо (пример файла см. Рисунок 22):

```
# Раскомментировать строку «listen_addresses» и указать в ней IP-адрес
# сервера СУБД PostgreSQL

listen_addresses = ’10.55.4.89’
```

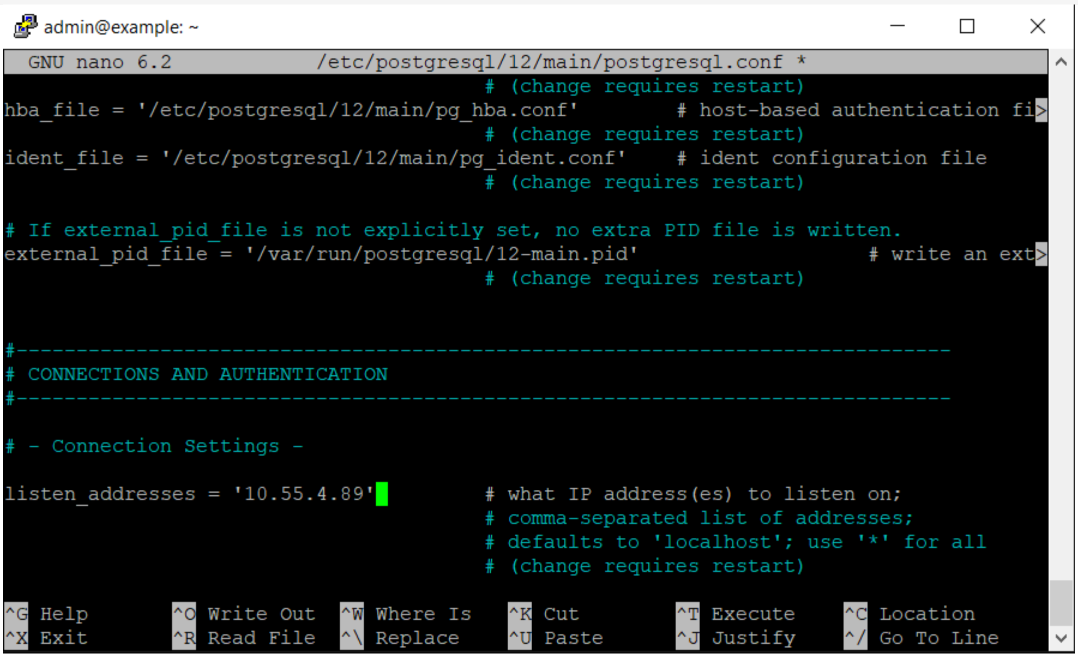


Рисунок 22 – Пример корректировки файла «postgresql.conf»

9. Перезагрузить СУБД PostgreSQL (пример выполнения команды см. Рисунок 23):

```
sudo systemctl restart postgresql
```

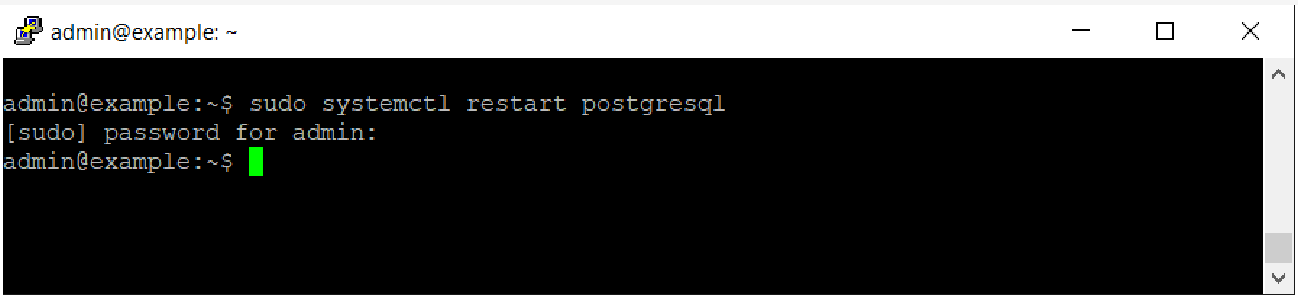


Рисунок 23 – Пример выполнения команды перезагрузки СУБД PostgreSQL

10. Проверить статус СУБД PostgreSQL (пример выполнения команды см. Рисунок 24):

```
sudo systemctl status postgresql
```

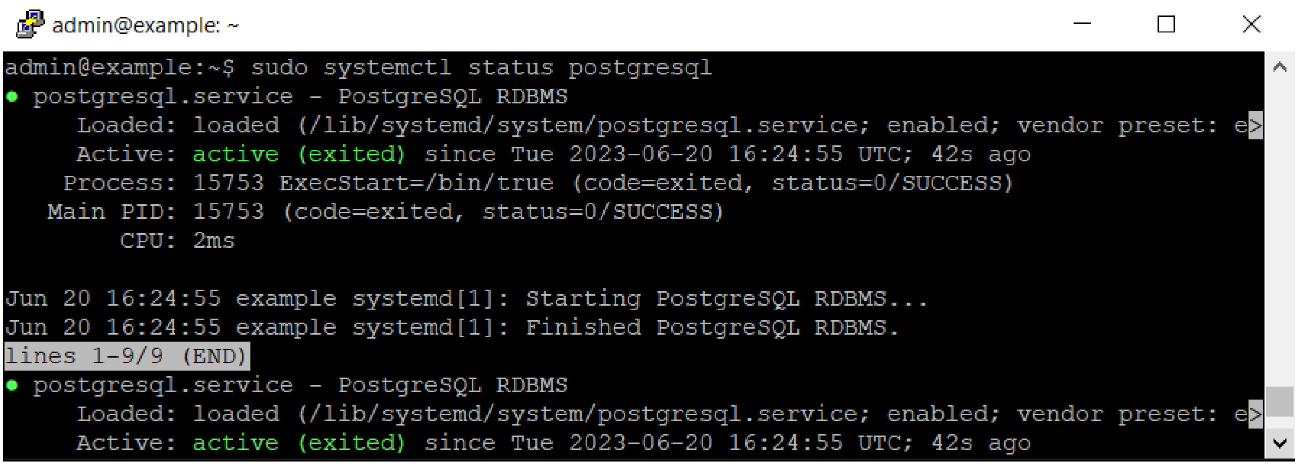


Рисунок 24 – Пример выполнения команды проверки стутуса СУБД PostgreSQL

11. Для дальнейшей работы с СУБД PostgreSQL (для создания БД и пользователя САКУРА PRO) необходимо запустить утилиту - терминал «psql» (пример выполнения команды см. Рисунок 25):

```
sudo -u postgres psql
```

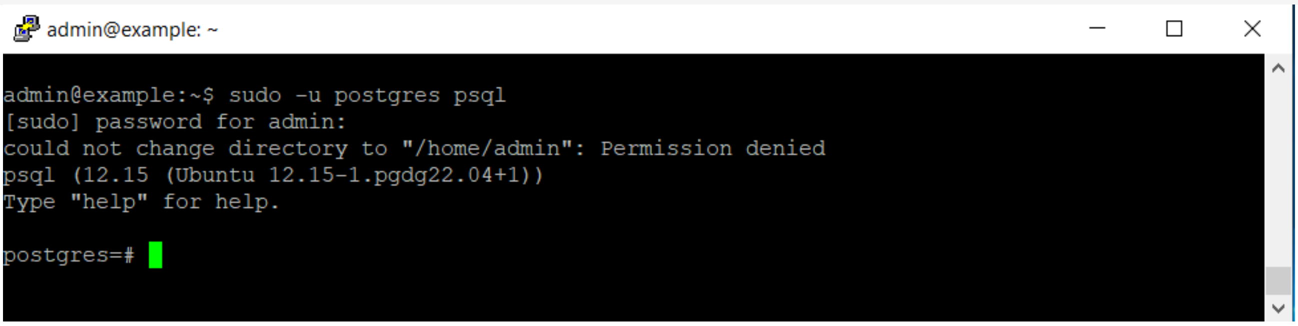


Рисунок 25 – Пример выполнения команды запуска утилиты «PSQL»

2.2.2 Создание базы данных

Для создания базы данных необходимо средствами СУБД PostgreSQL (в контексте утилиты «psql») выполнить следующую команду (пример выполнения команды см. Рисунок 26):

```
CREATE DATABASE {%DBNAME%}
WITH
OWNER = postgres
ENCODING = 'UTF8'
LC_COLLATE = 'en_US.UTF-8'
LC_CTYPE = 'en_US.UTF-8'
TABLESPACE = pg_default
CONNECTION LIMIT = -1
IS_TEMPLATE = False
```

, где:

```
{%DBNAME%} - имя создаваемой базы данных.
```

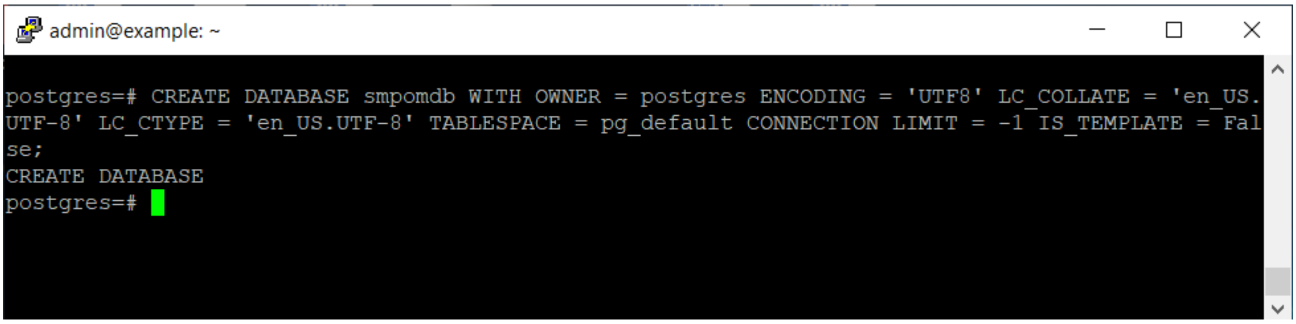


Рисунок 26 – Пример выполнения команды PSQL по созданию БД САКУРА PRO

2.2.3 Создание ролей

Для создания ролей необходимо средствами СУБД PostgreSQL (в контексте утилиты «psql») выполнить следующую команду (пример выполнения команды см. Рисунок 27):

```
CREATE ROLE {%ROLENAME%} NOSUPERUSER NOCREATEDB NOCREATEROLE INHERIT LOGIN
NOREPLICATION PASSWORD {%ROLEPASSWORD%};
```

, где:

```
{%ROLENAME%} – наименование роли (рекомендуется “Sakura Pro”);
{%ROLEPASSWORD%} – пароль роли.
```

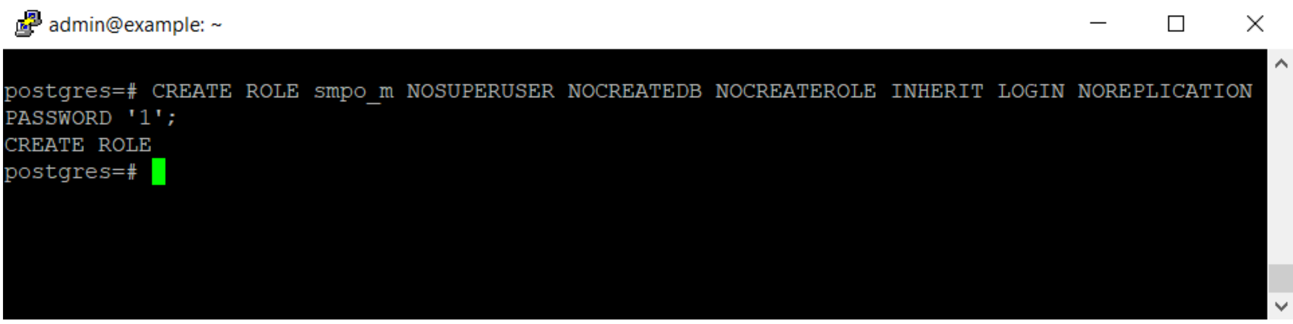


Рисунок 27 – Пример выполнения команды PSQL по созданию Пользовтаеля\Роли БД САКУРА PRO

2.2.4 Развертывание дампа базы данных

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 28):

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

, где:

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

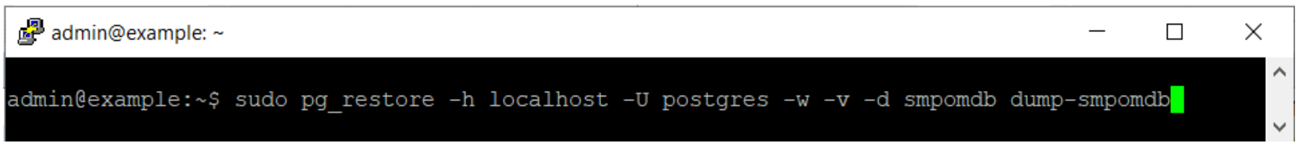


Рисунок 28 – Пример выполнения команды по развертыванию дампа БД САКУРА PRO

Для развертывания первичного дампа базы данных с базовыми настройками системы (для случая установки удаленно с ПЭВМ администратора БД под управлением ОС MS Windows) 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 дамба БД.
```

2.2.5 Проверка работоспособности СУБД PostgreSQL

Для проверки работоспособности СУБД PostgreSQL (кроме встроенной утилиты «psql»), для проверки возможного внешнего подключения к СУБД PostgreSQL, можно воспользоваться одним из существующих клиентов (вьюеров) баз данных (например, программой DBeaver). Для подключения к базе данных СУБД PostgreSQL необходимо указать: IP-адрес сервера, порт, имя базы данных, имя пользователя (роли) и ее пароль (см. Рисунок 29). При успешном подключении к БД PostgreSQL программа визуально подтвердит факт успешного подключения и покажет перечень объектов БД.

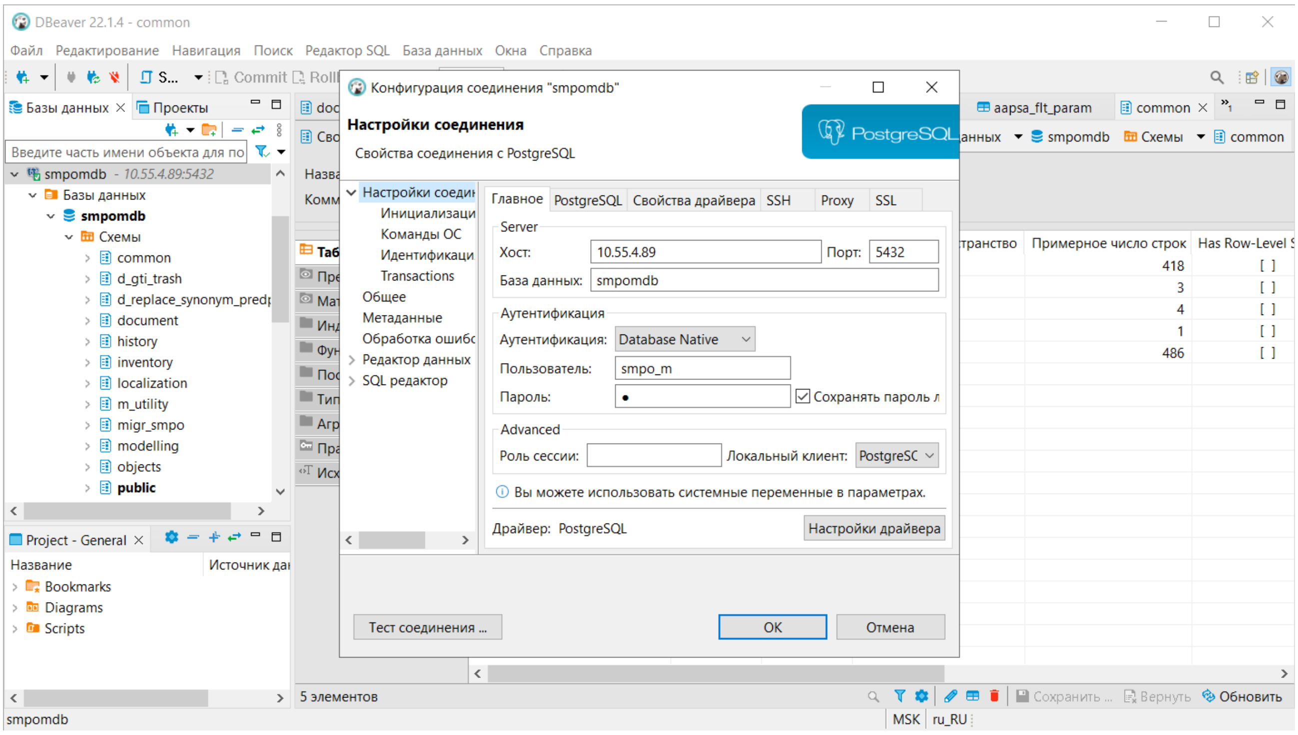


Рисунок 29 – Пример внешнего подключения к БД СУБД POstgreSQL

2.2.6 Схема резервирования СУБД PostgreSQL

Резервирование СУБД PostgreSQL может осуществляться по одному из следующих сценариев, приведенных в таблице 4.

Таблица 4 – Схема резервирования СУБД PostgreSQL

Тип	**Резервная единица оборудования**	**Периодичность создания (настройки) резерва**
1	**2**	**3**
Виртуальные серверы	Резервируется системой виртуальных машин, под управлением которой работают данные виртуальные серверы	Резервирование СУБД с использованием копии виртуальной машины должно осуществляться после каждой модификации СУБД. На случай, если модификация оказалась неудачной, рекомендуется хранить две резервные копии виртуальных машин (с предыдущей и настоящей модификациями СУБД).
Создание резервной копии БД СУБД PostgreSQL	Резервный сервер СУБД PostgreSQL или Сервер микросервисов, на которые должны копироваться резервные копии БД	Резервирование СУБД с использованием создания копии БД должно осуществляться ежедневно, до и после после каждой модификации СУБД. На случай, если модификация оказалась неудачной, рекомендуется хранить несколько резервных копии БД

2.2.7 Схема восстановления СУБД PostgreSQL

Восстановление СУБД PostgreSQL может осуществляться по одному из следующих сценариев, приведенных в таблице 5.

Таблица 5 – Схема восстановления СУБД PostgreSQL

Тип	**Резервная единица оборудования**	**Действия по восстановлению**
1	**2**	**3**
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.
Создание резервной копии БД СУБД PostgreSQL	Резервный сервер СУБД PostgreSQL или Сервер микросервисов, на которые должны копироваться резервные копии БД	1) восстановить работу БД СУБД PostgreSQL путем подъёма резервной копии БД

Скрипты по созданию и восстановлению резервных копий БД СУБД PostgreSQL представлены в Приложении (см. ПРИЛОЖЕНИЕ Б. СКРИПТЫ ПО СОЗДАНИЮ И ВОССТАНОВЛЕНИЮ РЕЗЕРВНОЙ КОПИИ БД СУБД POSTGRESQL).

После восстановления работоспособности СУБД в любом из указанных случаев необходимо проверить и при необходимости актуализировать базы данных (поднять резервн копию («бэкап») и «накатить» журналы транзакций).

2.3 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»).

2.3.1 Настройка файла Hosts

1. Найти и открыть для корректировки файл «hosts» (пример выполнения команды см. Рисунок 30):

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

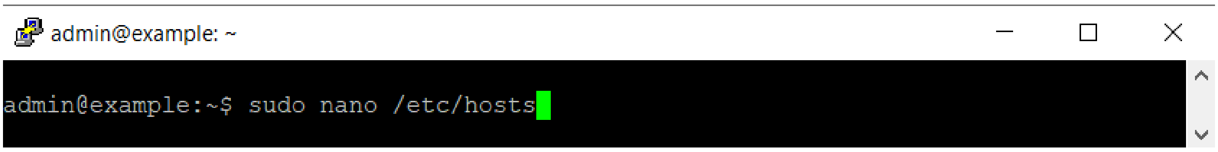


Рисунок 30 – Пример выполнения команды открытия для корректировки файла «hosts»

2. Дописать в файл «hosts» следующие строки (пример выполнения команды см. Рисунок 31):

```
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.
```

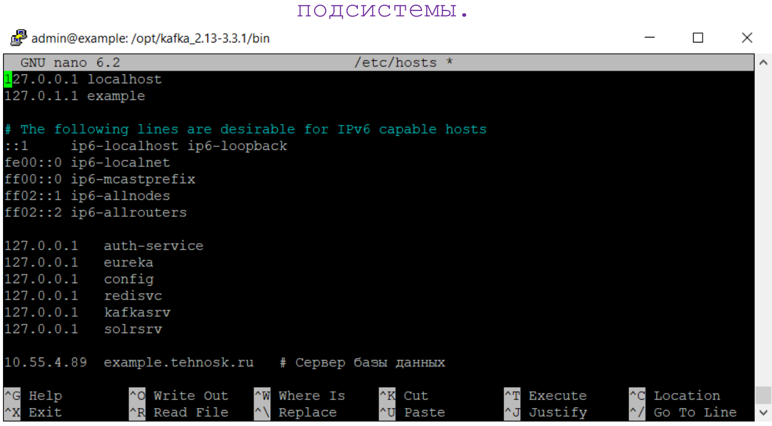
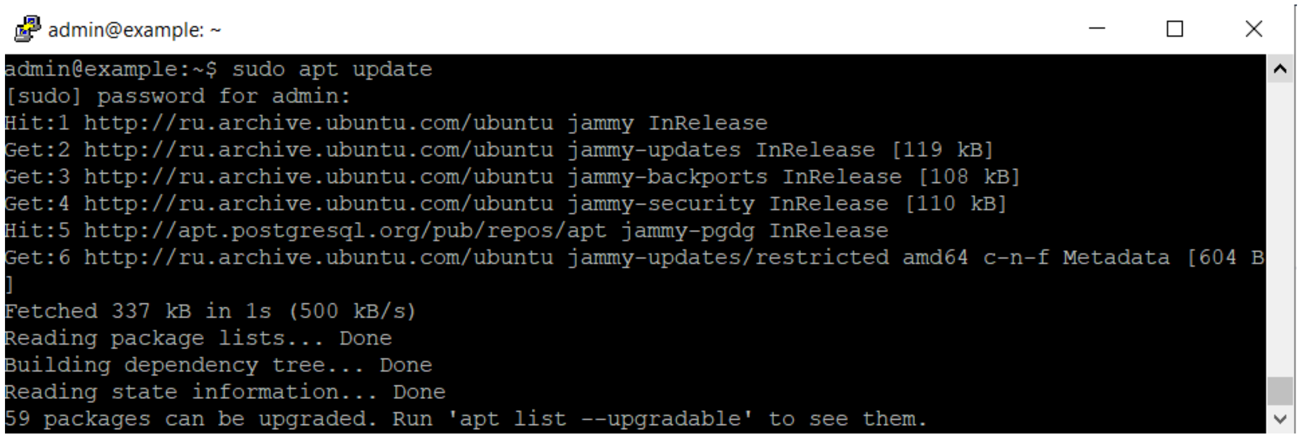


Рисунок 31 – Пример корректировки файла «hosts»

2.3.2 Установка и настройка сервера «NGINX»

1. Обновить списки пакетов операционной системы из репозиториев (пример выполнения команды см. Рисунок 32):

```
sudo apt update
```

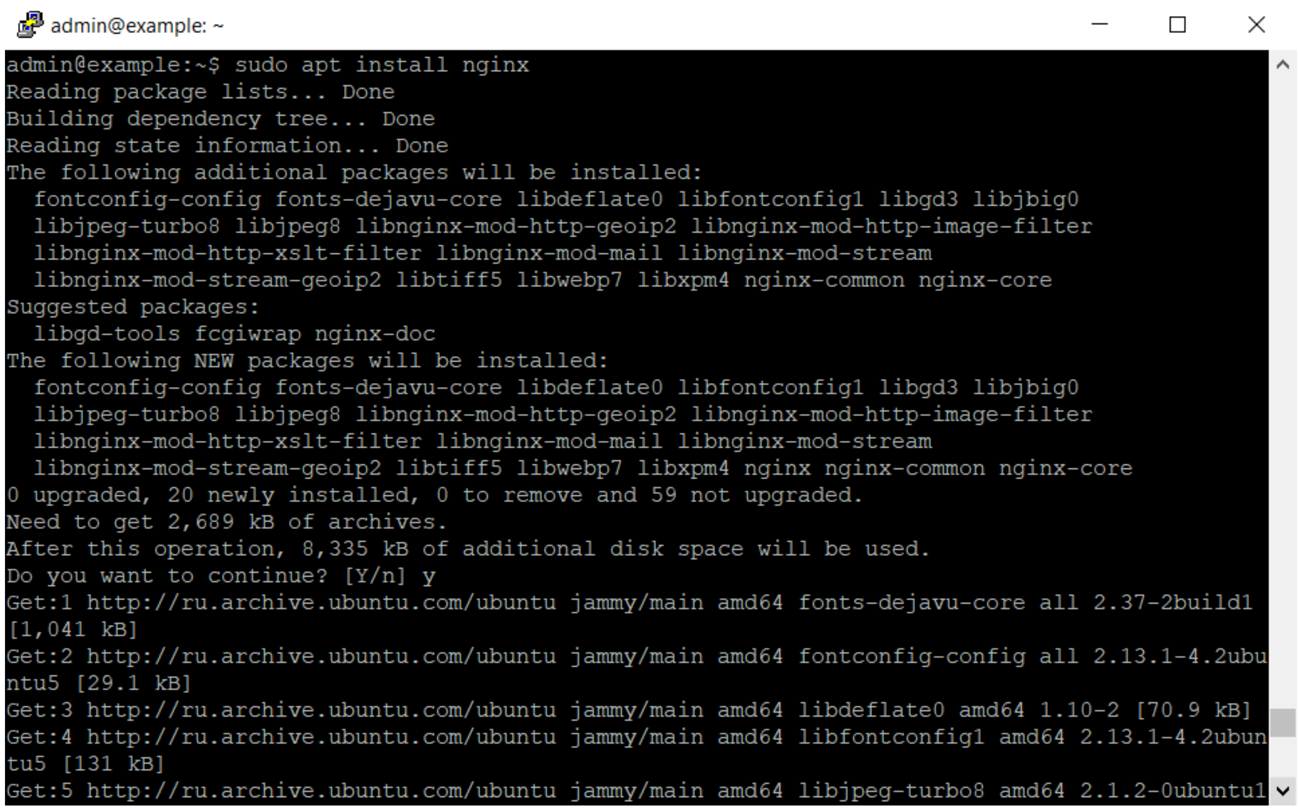


```
admin@example: ~
admin@example:~$ sudo apt update
[sudo] password for admin:
Hit:1 http://ru.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ru.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://ru.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:4 http://ru.archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:5 http://apt.postgresql.org/pub/repos/apt jammy-pgdg InRelease
Get:6 http://ru.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [604 B
]
Fetched 337 kB in 1s (500 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
59 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Рисунок 32 – Пример выполнения команды обновления пакетов операционной системы

2. Установить Nginx (пример выполнения команды см. Рисунок 33 и Рисунок 34):

```
sudo apt install nginx
```



```
admin@example: ~
admin@example:~$ 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 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 nginx-common nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core libdeflate0 libfontconfig1 libgd3 libjbig0
  libjpeg-turbo8 libjpeg8 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 nginx nginx-common nginx-core
0 upgraded, 20 newly installed, 0 to remove and 59 not upgraded.
Need to get 2,689 kB of archives.
After this operation, 8,335 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ru.archive.ubuntu.com/ubuntu jammy/main amd64 fonts-dejavu-core all 2.37-2build1
[1,041 kB]
Get:2 http://ru.archive.ubuntu.com/ubuntu jammy/main amd64 fontconfig-config all 2.13.1-4.2ubu
ntu5 [29.1 kB]
Get:3 http://ru.archive.ubuntu.com/ubuntu jammy/main amd64 libdeflate0 amd64 1.10-2 [70.9 kB]
Get:4 http://ru.archive.ubuntu.com/ubuntu jammy/main amd64 libfontconfig1 amd64 2.13.1-4.2ubun
tu5 [131 kB]
Get:5 http://ru.archive.ubuntu.com/ubuntu jammy/main amd64 libjpeg-turbo8 amd64 2.1.2-0ubuntu1
```

Рисунок 33 – Пример выполнения команды установки Nginx. Начало

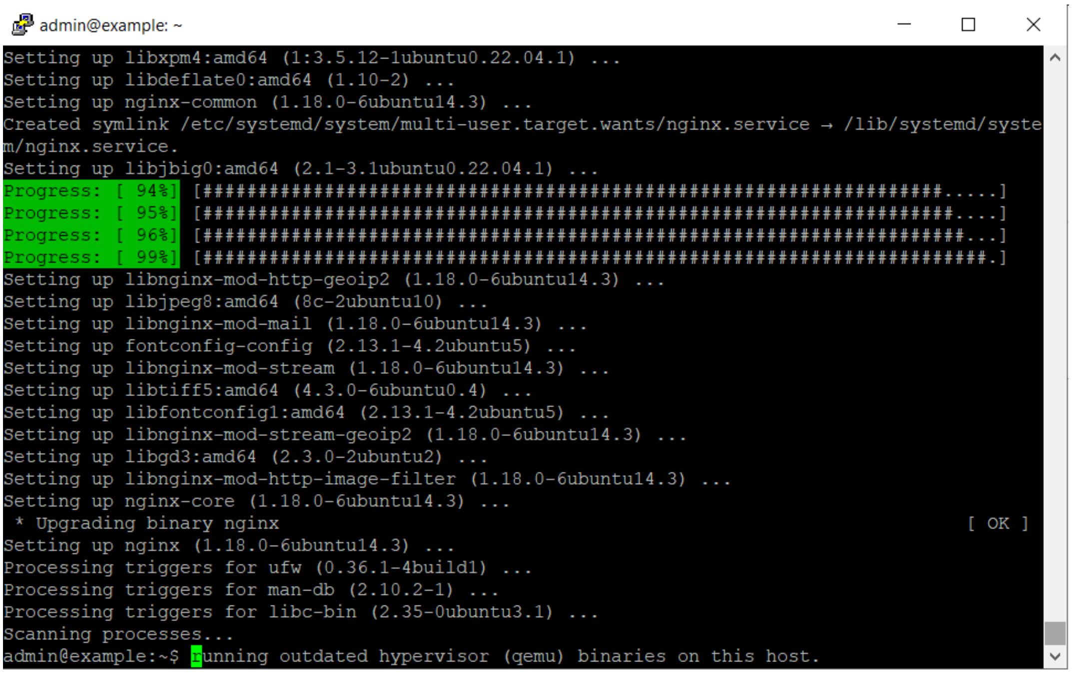


Рисунок 34 – Пример выполнения команды установки Nginx. Окончание

3. Добывить программу Nginx в автозагрузку (пример выполнения команды см. Рисунок 35):

```
sudo systemctl enable nginx
```

Рисунок 35 – 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 41).

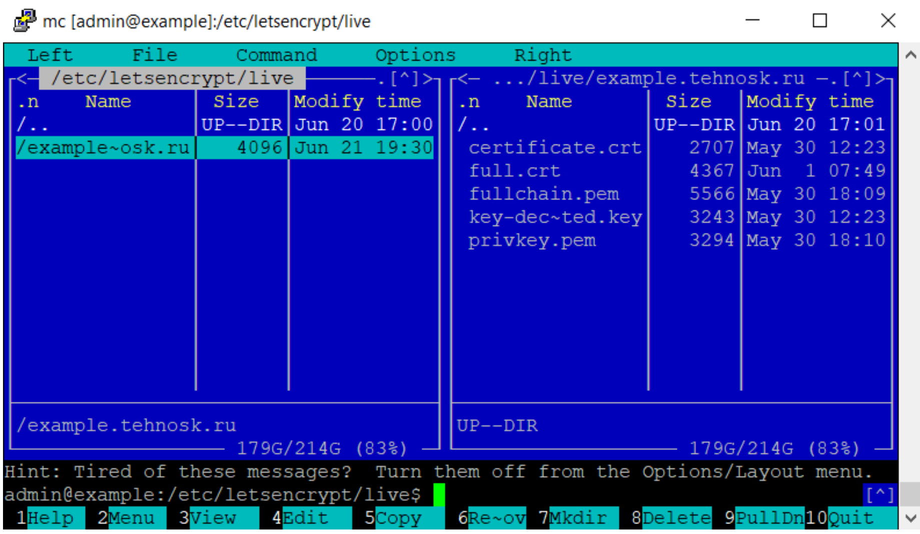


Рисунок 36 – Пример размещения файлов – ключей на сервере

5. Перейти в каталог «conf.d» по пути /etc/nginx/conf.d (пример выполнения команды см. Рисунок 37):

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

Рисунок 37 – Пример выполнения команды по переходу в каталог «/etc/nginx/conf.d»

6. Создать в каталоге «/etc/nginx/conf.d» файл конфигурации «nginx_Sakura Pro.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 38 и Рисунок 39):

```
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 38 – Пример выполнения команды открытия файла «nginx_Sakura Pro.conf»

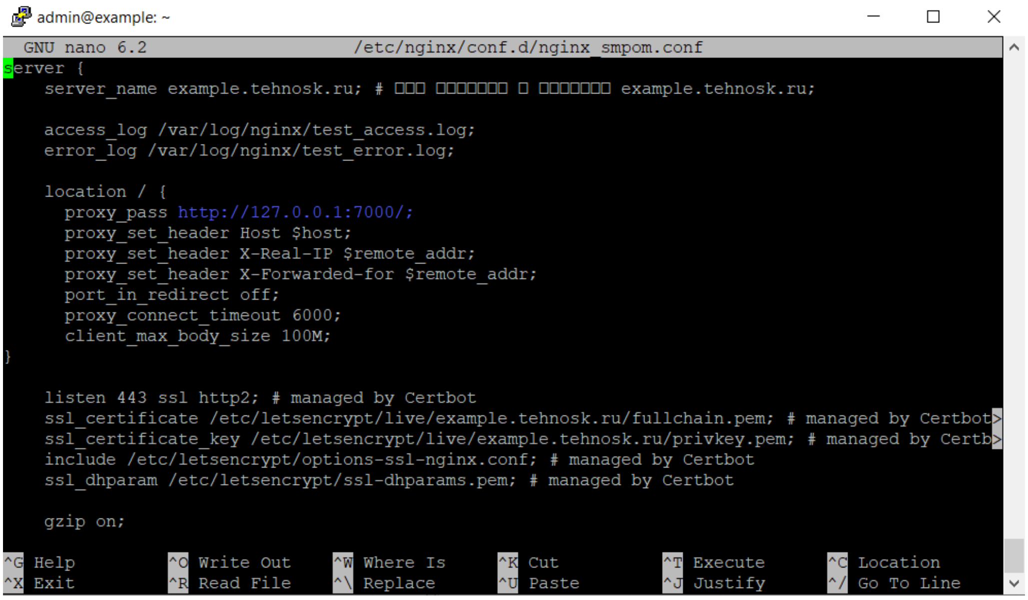


Рисунок 39 – Пример корректировки файла «nginx_Sakura Pro.conf»

7. Запустить Nginx (пример выполнения команды см. Рисунок 40):

```
sudo service nginx start
```

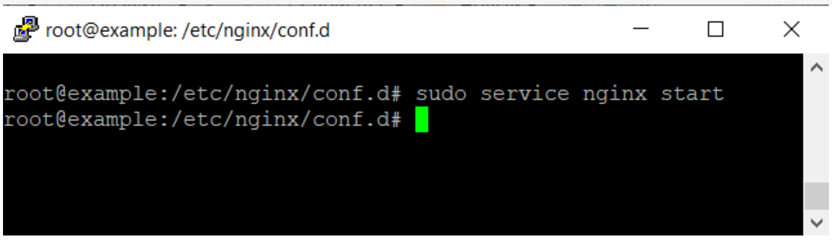


Рисунок 40 – Пример выполнения команды запуска программы Nginx

8. Проверить статус работы Nginx (пример выполнения команды см. Рисунок 42):

```
sudo service nginx status
```

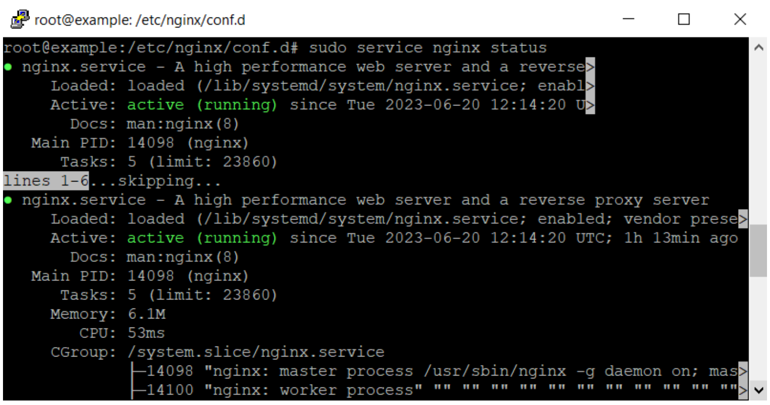


Рисунок 41 – Пример выполнения команды проверки статуса работы программы Nginx

2.3.3 Установка и настройка Java JDK

1. Установить утилиту «wget»:

```
apt install wget
```

2. Скачать дистрибутив с Java JDK 17 (пример выполнения команды см. Рисунок 42 Рисунок 42)):

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

Рисунок 42 – 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 43):

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

Рисунок 43 – 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 44):

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

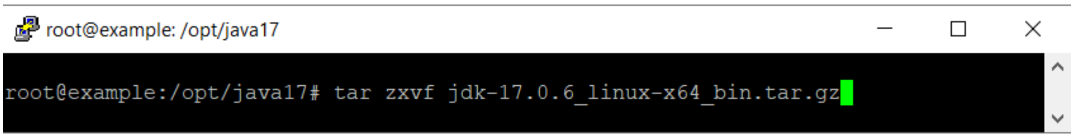


Рисунок 44 – Пример выполнения команды распаковки архива Java JDK. Начало

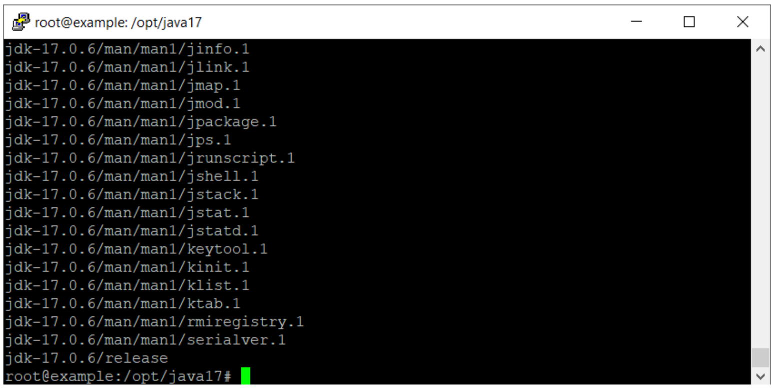


Рисунок 45 – Пример выполнения команды распаковки архива Java JDK. Окончание

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 46):

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

Рисунок 46 – Пример выполнения команды открытия файла с переменными окружения

В файл «profile» необходимо добавить следующие строки (пример выполнения команды см. Рисунок 47):

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

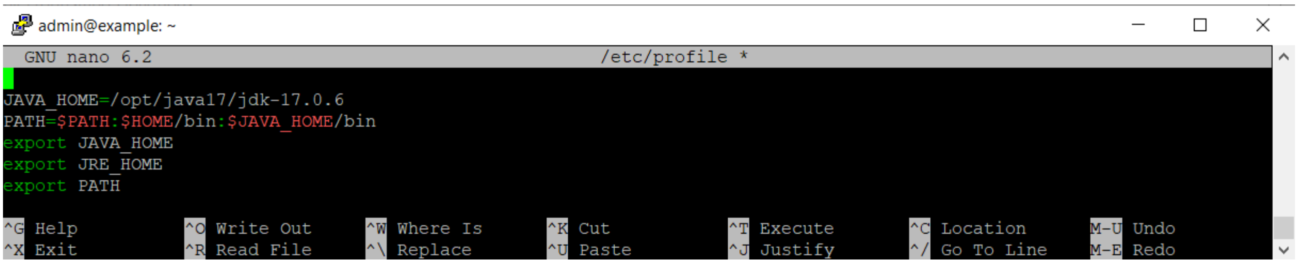


Рисунок 47 – Пример выполнения команды открытия файла с переменными окружения

6. Проверить работоспособность Java JDK на сервере (пример выполнения команды см. Рисунок 48):

```
java -version
```

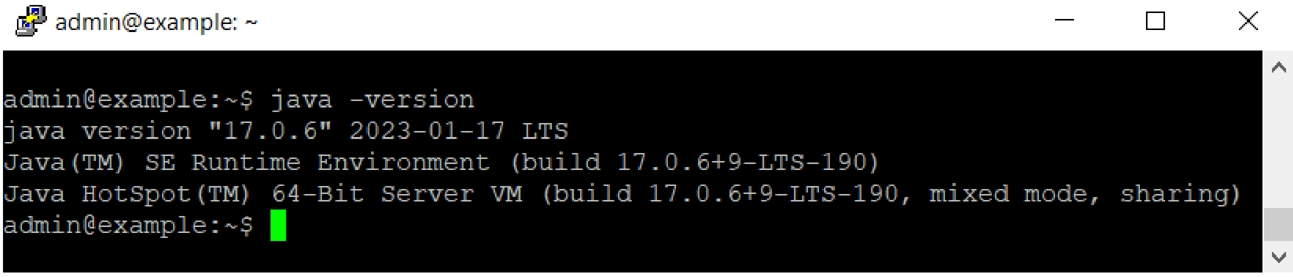


Рисунок 48 – Пример выполнения команды проверки работоспособности Java JDK

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

2.3.4 Установка и настройка Redis

1. Установить Redis (пример выполнения команды см. Рисунок 49 и Рисунок 50):

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

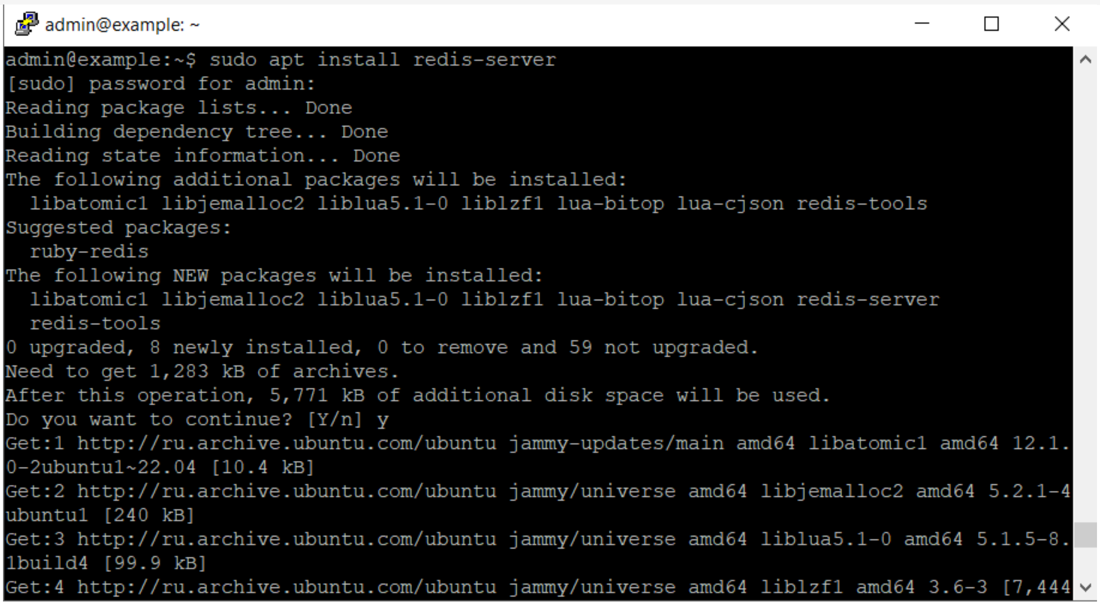


Рисунок 49 – Пример выполнения команды установки программы Redis. Начало

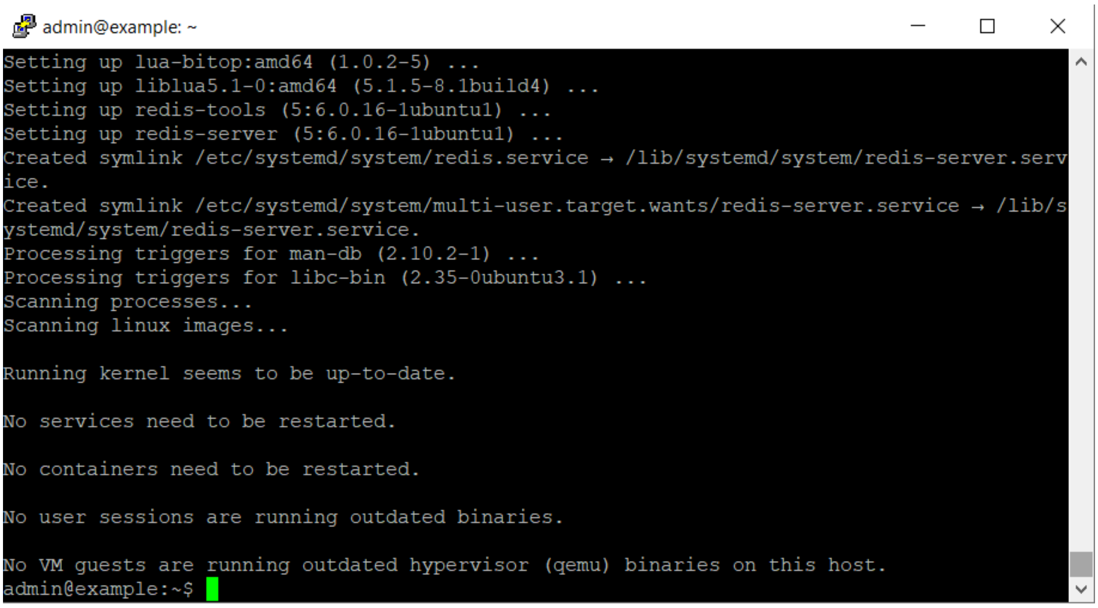


Рисунок 50 – Пример выполнения команды установки программы Redis. Окончание

2. Открыть для редактирования файл «/etc/redis/redis.conf» (пример выполнения команды см. Рисунок 51):

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

Рисунок 51 – 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 52 - Рисунок 54):

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

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

# Найти парамтер «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.
```

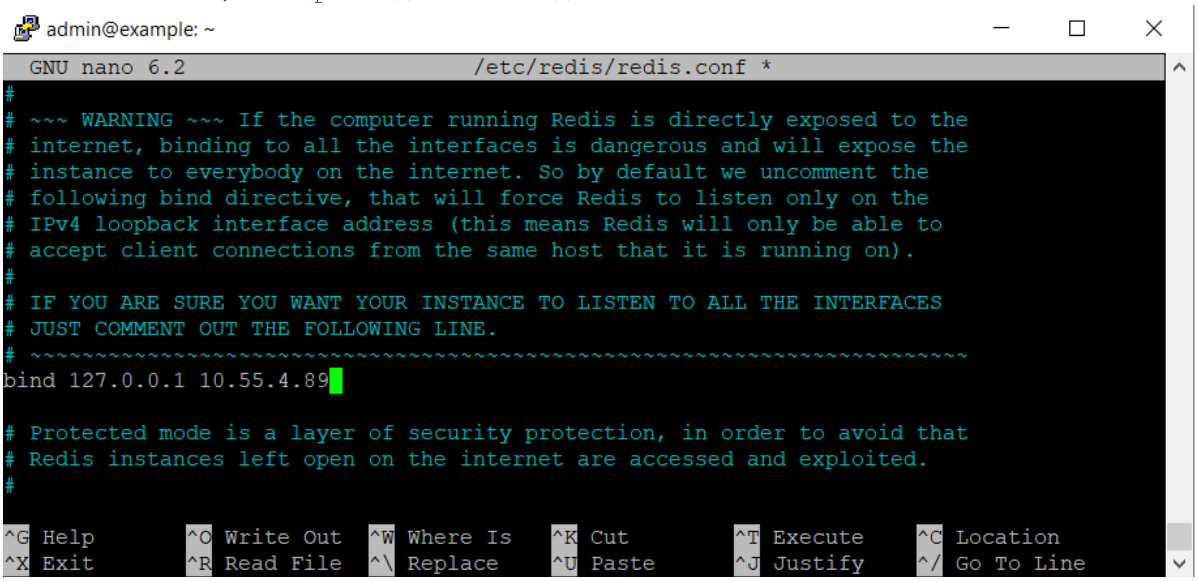


Рисунок 52 – Пример корректировки файла «redis.conf»

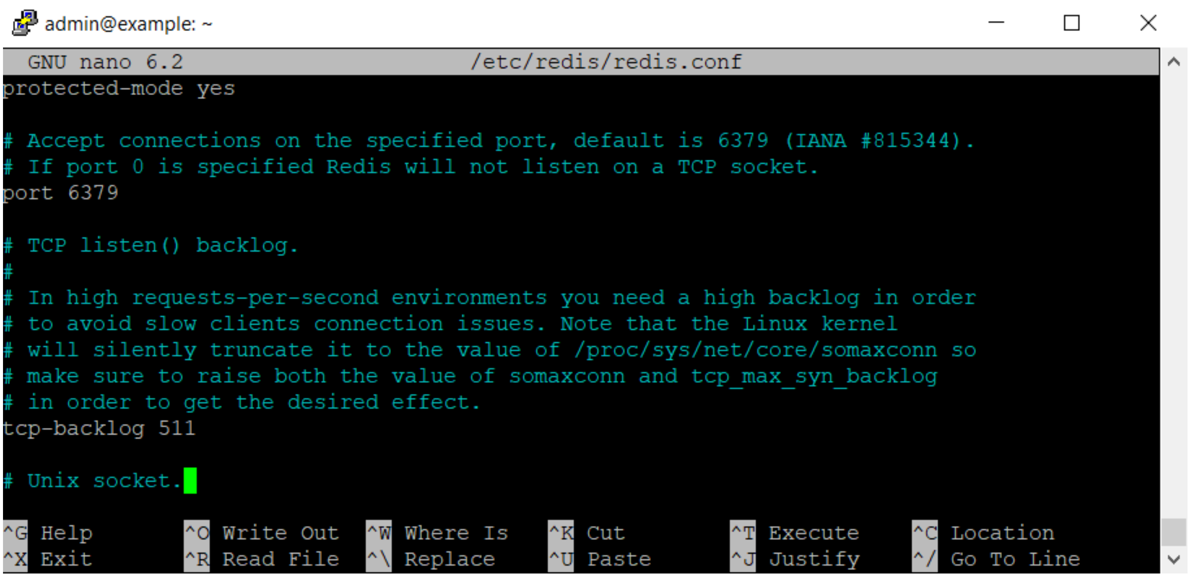


Рисунок 53 – Пример корректировки файла «redis.conf»

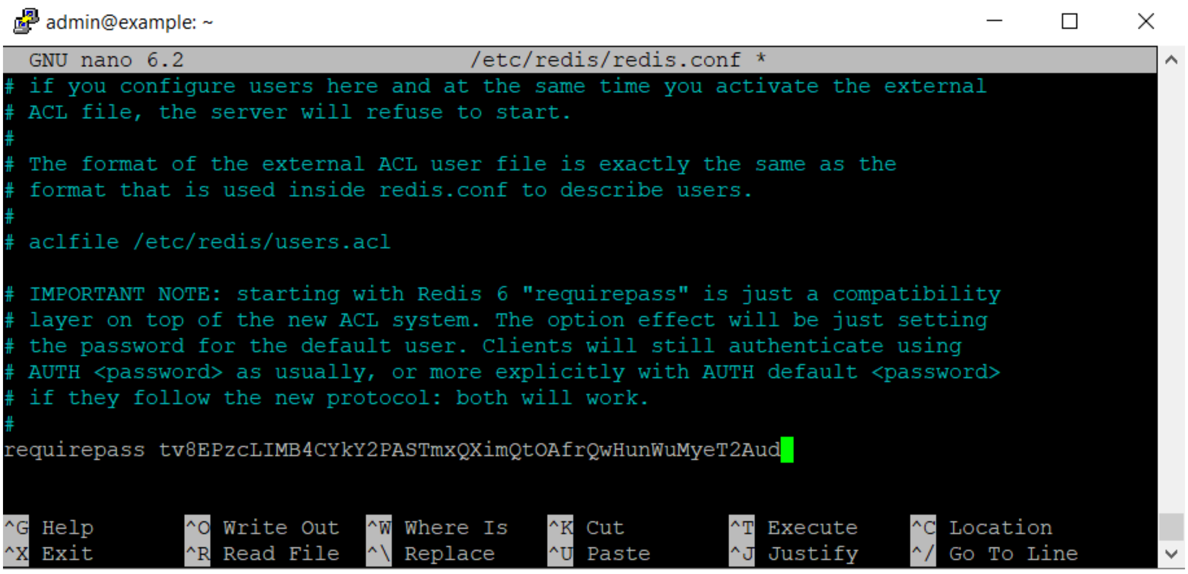


Рисунок 54 – Пример корректировки файла «redis.conf»

3. Перезапустить Redis (пример выполнения команды см. Рисунок 55):

```
sudo systemctl restart redis
```

Рисунок 55 – Пример выполнения команды перезапуска программы Redis

4. Проверить статус работы Redis (пример выполнения команды см. Рисунок 56):

```
sudo systemctl status redis
```

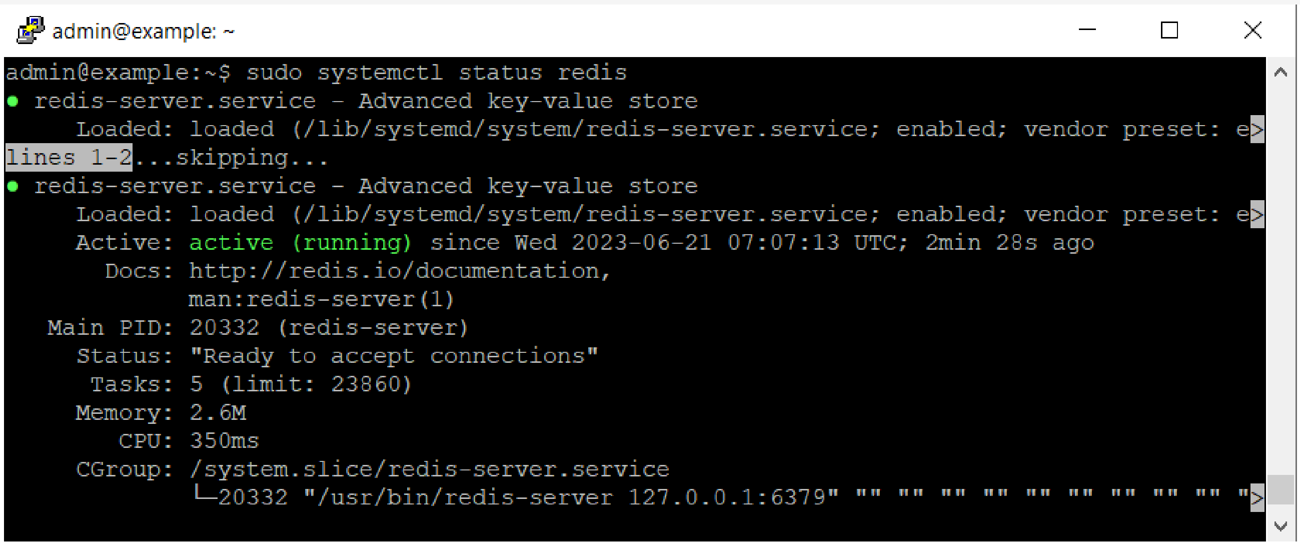


Рисунок 56 – Пример выполнения команды проверки статуса работы программы Redis

2.3.5 Установка и настройка Apache Kafka

1. Скачать дистрибутив программы Kafka (пример выполнения команды см. Рисунок 57):

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

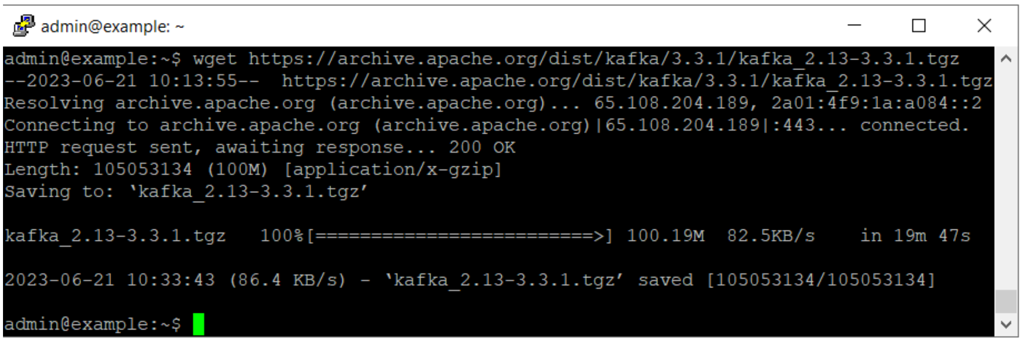


Рисунок 57 – Пример выполнения команды скачивания дистрибутива программы Kafka

2. Распаковать полученный массив (пример выполнения команды см. Рисунок 58 и Рисунок 59):

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

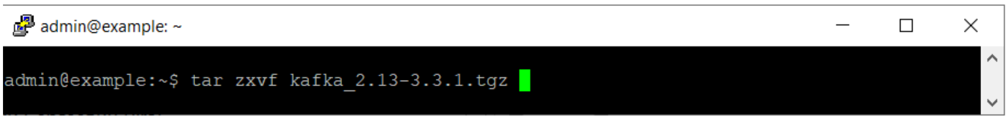


Рисунок 58 – Пример выполнения команды распаковки программы Kafka. Начало

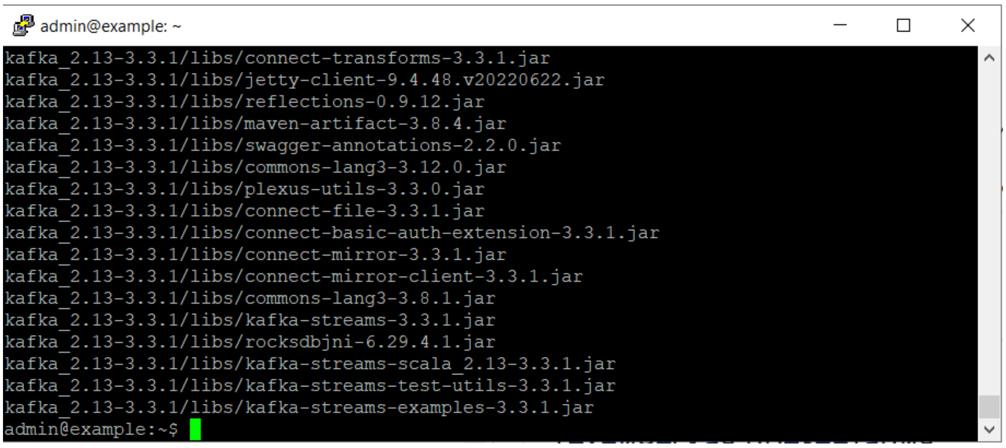


Рисунок 59 – Пример выполнения команды распаковки программы Kafka. Окончание

3. Переместить развернутую папку приложения Kafka в директорию «/opt» (пример выполнения команды см. Рисунок 60):

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

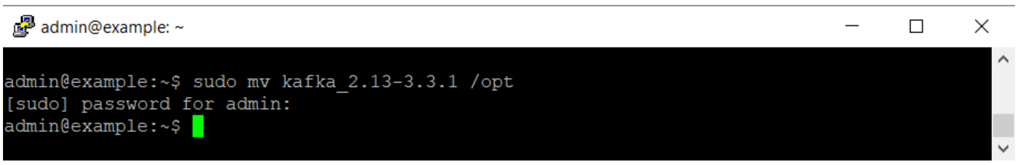


Рисунок 60 – Пример выполнения команды переноса программы Kafka в директорию «/opt»

4. Создать сервисный файл systemd для сервиса «zookeeper.service» и поместить его в /etc/systemd/system (пример файла см. Рисунок 61 и Рисунок 62):

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
```

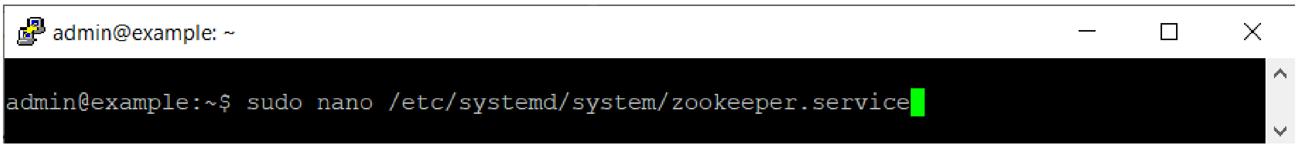


Рисунок 61 – Пример выполнения команды открытия для редактирования файла «zookeeper.service»

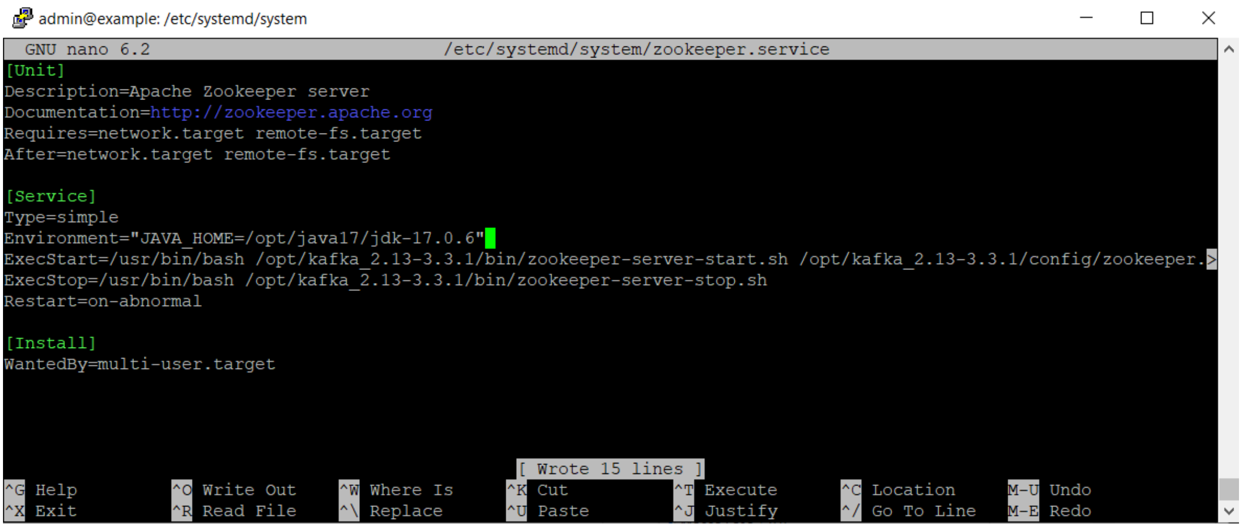


Рисунок 62 – Пример файла «zookeeper.service»

5. Создать сервисный файл systemd для сервиса «kafka.service» и поместить его в /etc/systemd/system (пример файла см. Рисунок 63 и Рисунок 64):

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
```

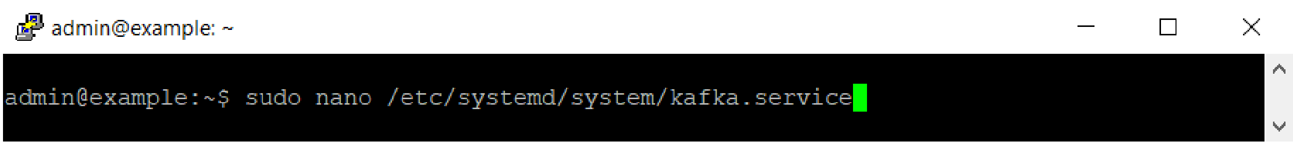


Рисунок 63 – Пример выполнения команды открытия для редактирования файла «kafka.service»

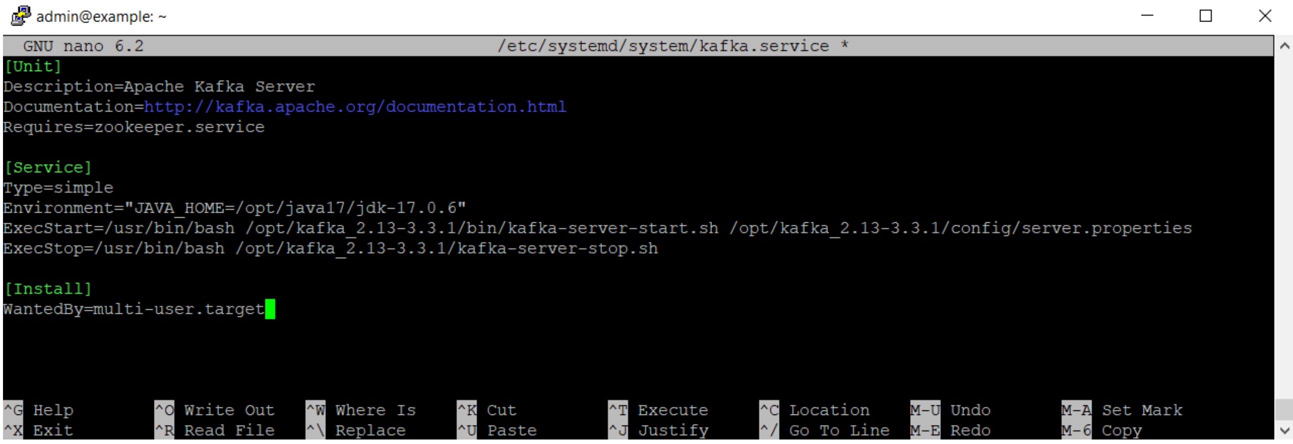


Рисунок 64 – Пример файла «kafka.service»

6. Обновить перечень доступных сервисов операционной системы (пример выполнения команды см. Рисунок 65):

```
sudo systemctl daemon-reload
```

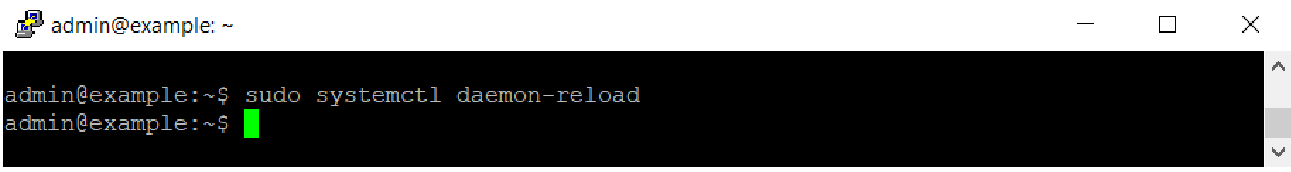


Рисунок 65 – Пример выполнения команды обновления сервисов операционной системы

7. Запустить сервис zookeeper (пример выполнения команды см. Рисунок 66):

```
sudo systemctl start zookeeper
```

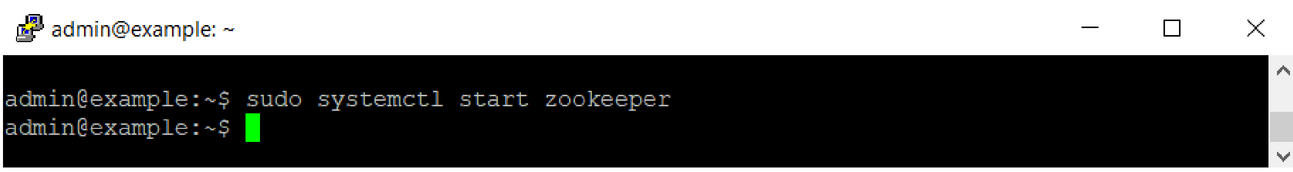


Рисунок 66 – Пример выполнения команды запуска сервиса «zookeeper»

8. Запустить сервис kafka (пример выполнения команды см. Рисунок 67):

```
sudo systemctl start kafka
```

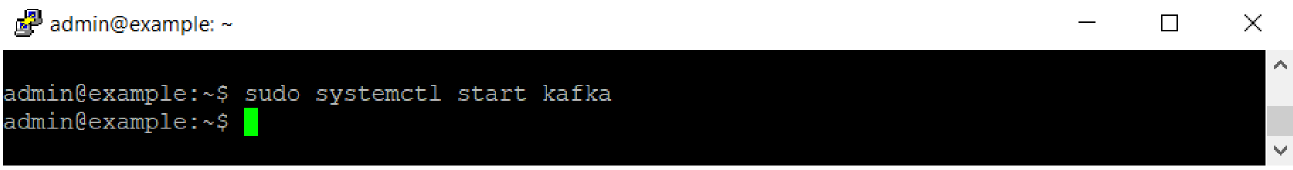


Рисунок 67 – Пример выполнения команды запуска сервиса «kafka»

9. Убедиться что сервис zookeeper запущен (пример выполнения команды см. Рисунок 68):

```
sudo systemctl status zookeeper
```

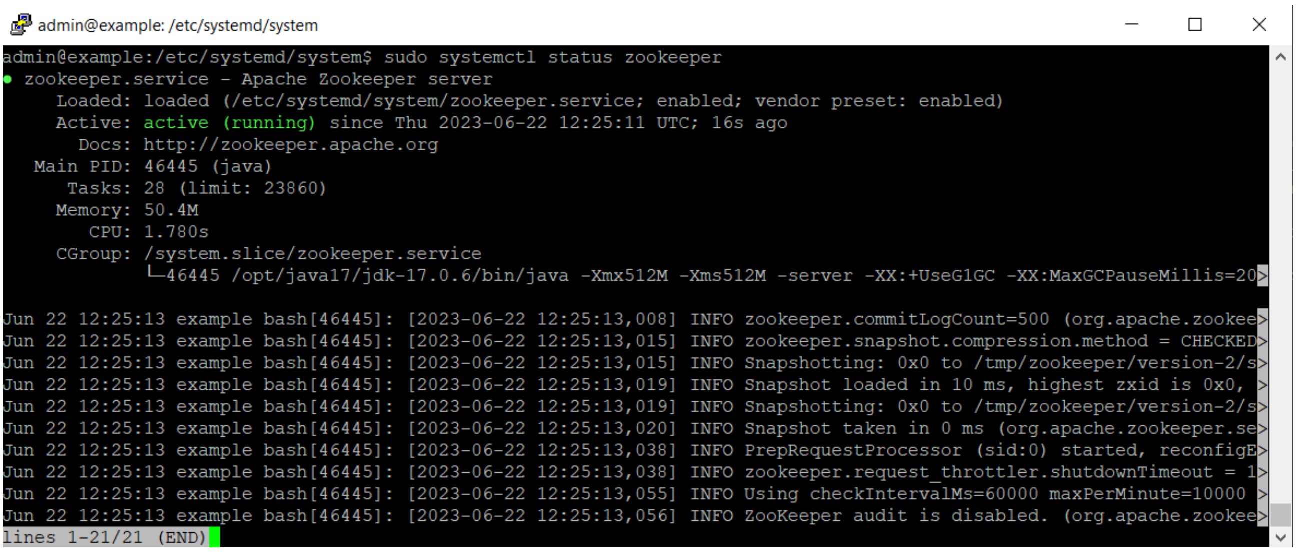


Рисунок 68 – Пример выполнения команды проверки статуса работы сервиса «zookeeper»

10. Убедиться что сервис kafka запущен (пример выполнения команды см. Рисунок 69):

```
sudo systemctl status kafka
```

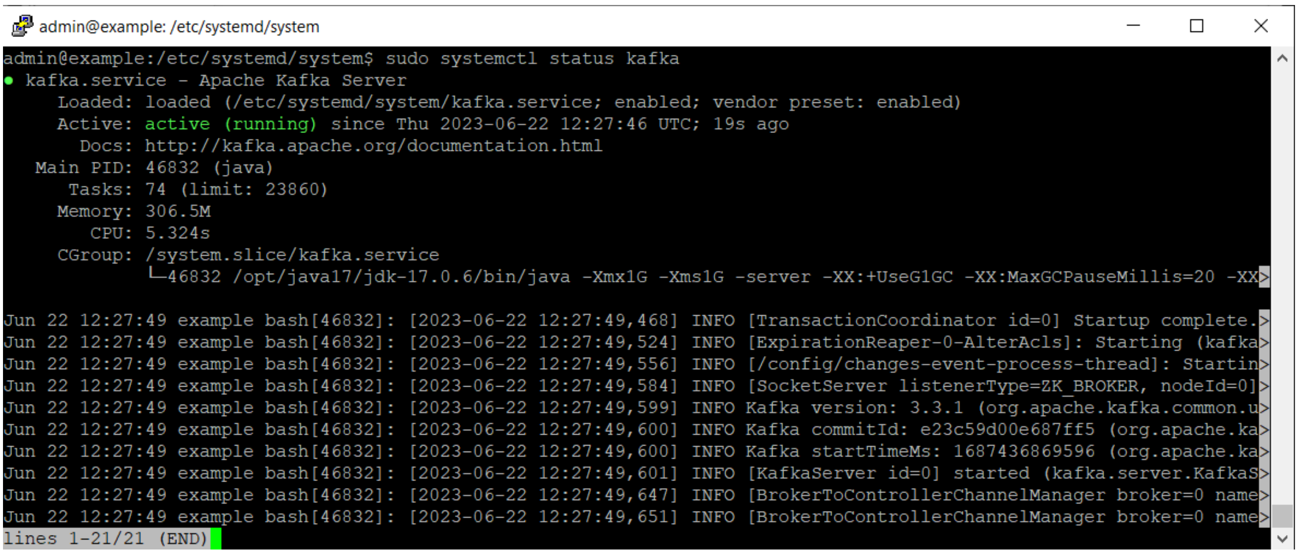


Рисунок 69 – Пример выполнения команды проверки статуса работы сервисв «kafka»

11. Найти и открыть для корректировки файл «hosts» (пример выполнения команды см.Рисунок 70):

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

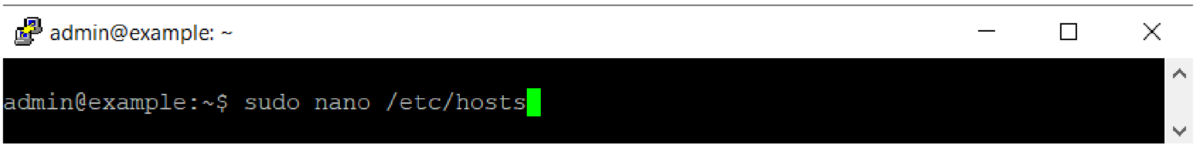


Рисунок 70 – Пример выполнения команды открытия файла для корректировки «hosts»

Проверить наличие, и если нет, дописать в файл «hosts» следующую строку (пример выполнения команды см. Рисунок 71):

```
127.0.0.1   kafkasrv
```

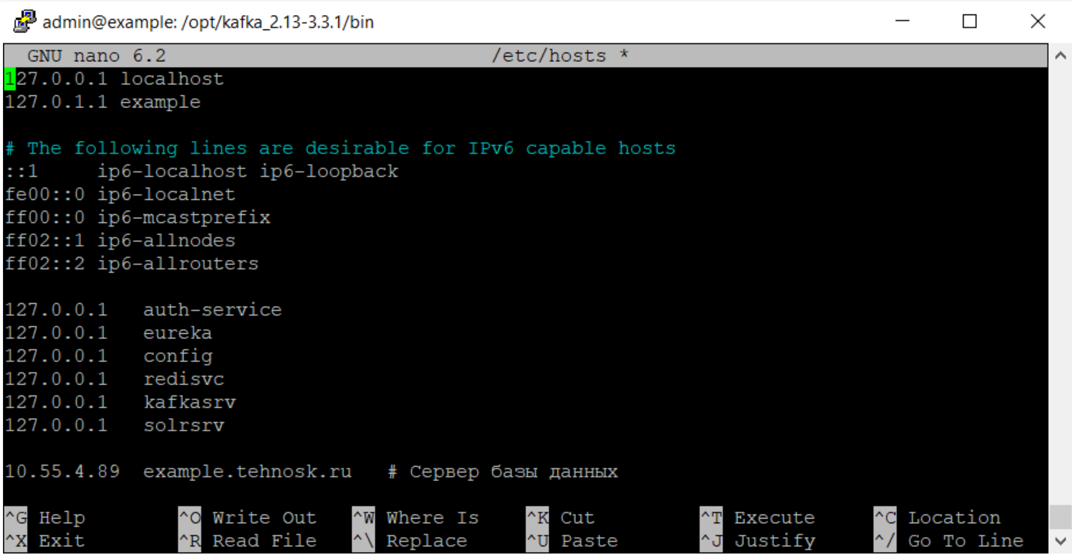


Рисунок 71 – Пример корректировки файла «hosts»

12. Для автоматического запуска сервиса «zookeeper» при каждом старте операционной системы необходимо выполнить команду (пример выполнения команды см. Рисунок 72):

```
sudo systemctl enable zookeeper
```

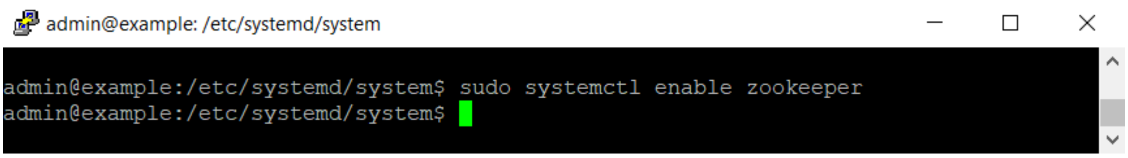


Рисунок 72 – Пример выполнения команды для службы «zookeeper»

13. Для автоматического запуска сервиса «kafka» при каждом старте операционной системы необходимо выполнить команду (пример выполнения команды см. Рисунок 73):

```
sudo systemctl enable kafka
```

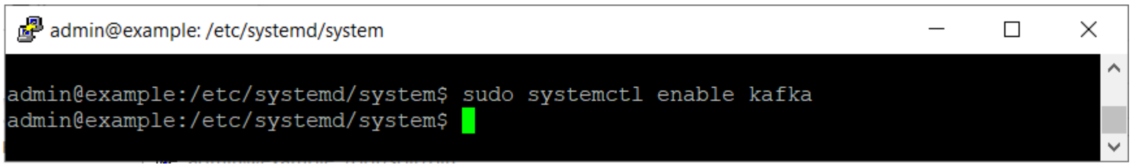


Рисунок 73 – Пример выполнения команды для службы «kafka»

14. Перейти в каталог /opt/kafka_2.13-3.3.1/bin (пример выполнения команды см. Рисунок 74):

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

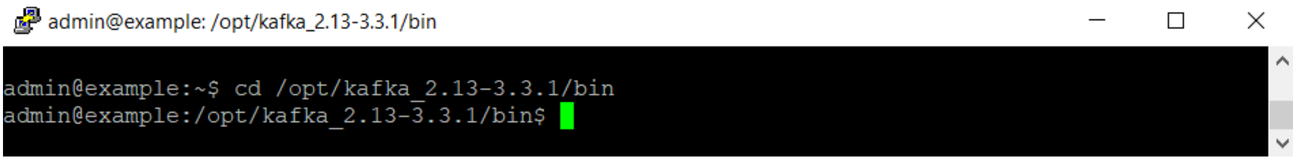


Рисунок 74 – Пример выполнения команды перехода в каталог «/opt/kafka_2.13-3.3.1/bin»

15. Выполнить команды создания очередей (пример выполнения команды см. Рисунок 75):

```
./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
```

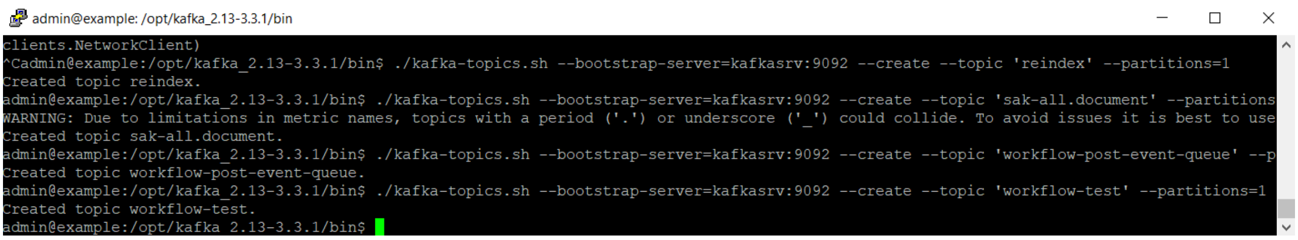


Рисунок 75 – Пример выполнения команд создания очередей в программе Kafka

2.3.6 Установка и настройка Apache Solr

1. Скачать дистрибутив программы Solr (пример выполнения команды см. Рисунок 76):

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

Рисунок 76 – Пример выполнения команд скачивания дистрибутива программы Solr

2. Распаковать полученный архив в каталог «/opt» (пример выполнения команды см. Рисунок 77 и Рисунок 78):

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

Рисунок 77 – Пример выполнения команд разархивирования архива программы Solr. Начало

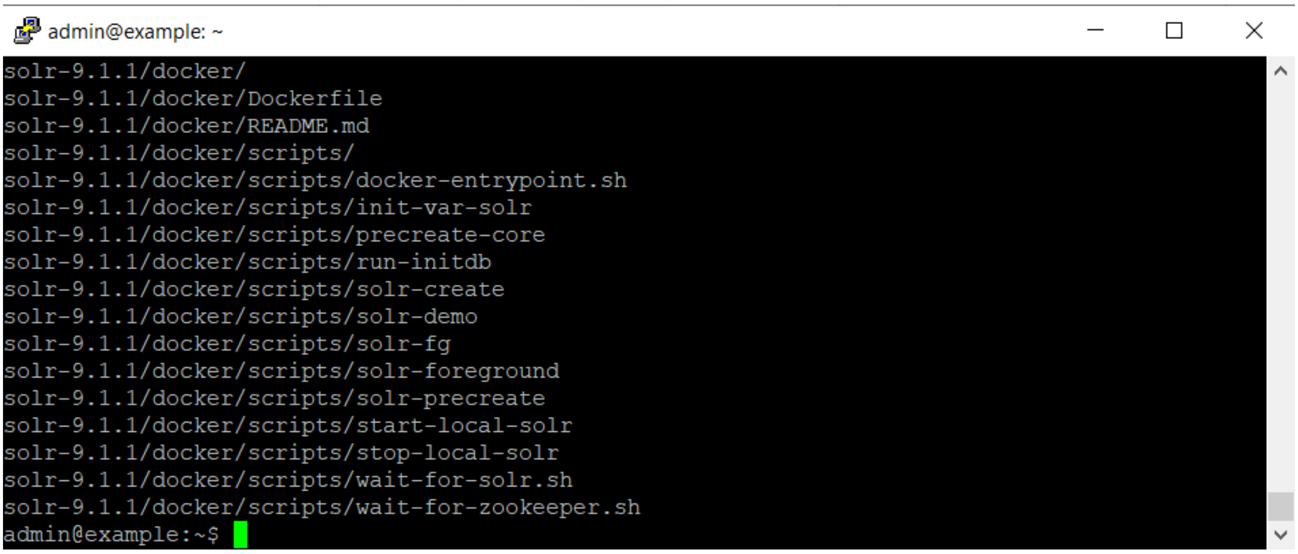


Рисунок 78 – Пример выполнения команд разархивирования архива программы Solr. Окончание

3. Переместить рабочий каталог программы Solr в каталог «/opt» (пример выполнения команды см. Рисунок 79):

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

Рисунок 79 – Пример выполнения команд перемещения рабочего каталога программы Solr

4. Перейти в каталог «/opt/solr-9.1.1/bin» (пример выполнения команды см. Рисунок 80):

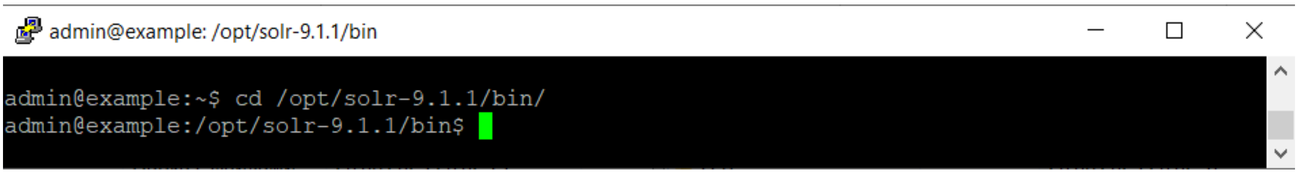


Рисунок 80 – Пример выполнения команд перехода в каталог «/opt/solr-9.1.1/bin»

5. Выполнить инициализацию программы Solr (пример выполнения команды см. Рисунок 81):

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

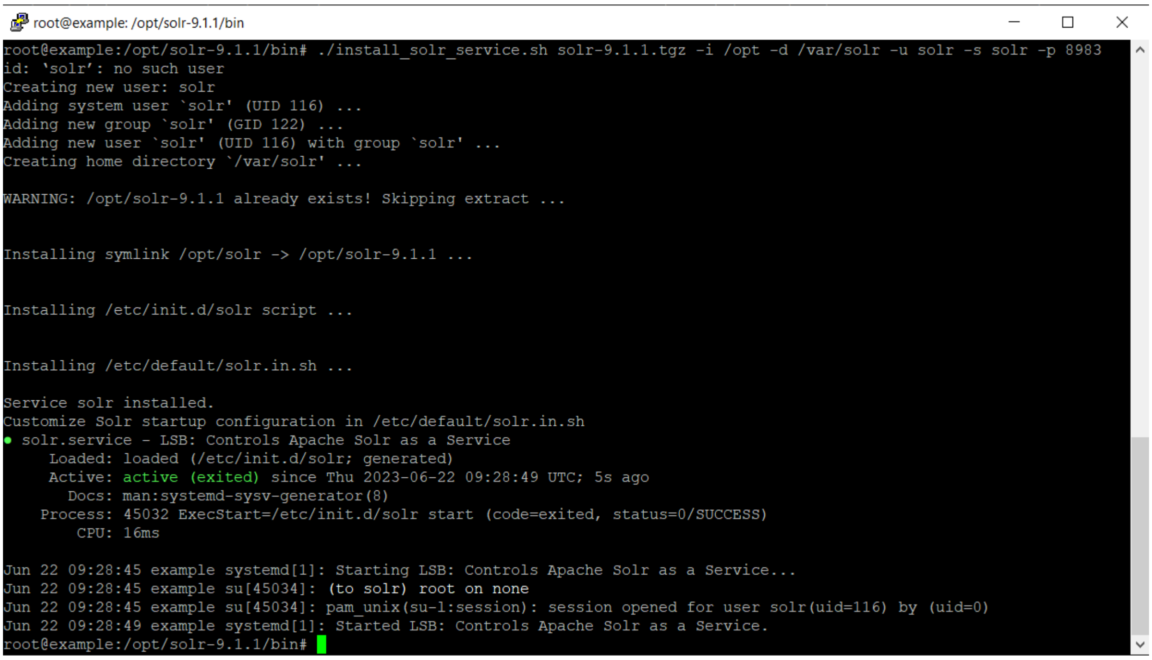


Рисунок 81 – Пример выполнения команд инициализации программы Solr

6. Выполнить запуск программы Solr (пример выполнения команды см. Рисунок 82):

```
sudo service solr start
```

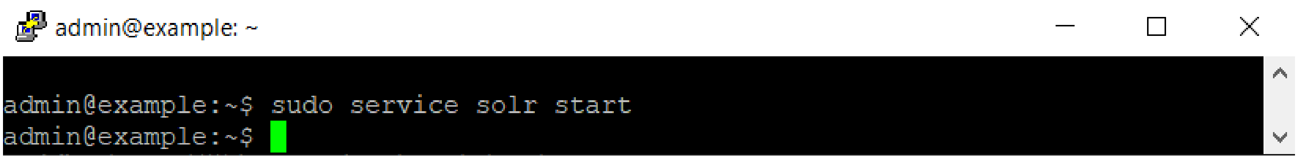
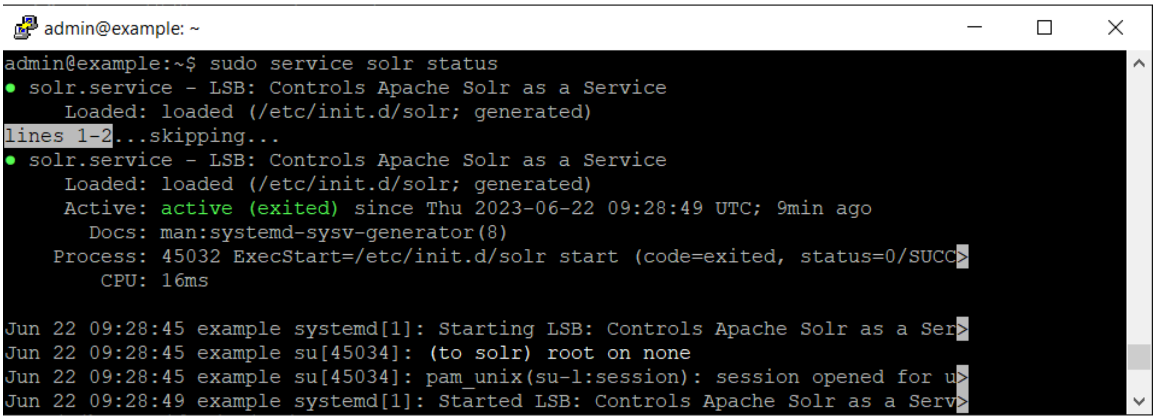


Рисунок 82 – Пример выполнения команды запуска программы Solr

7. Проверить статус прогрммы Solr (пример выполнения команды см. Рисунок 83):

```
sudo service solr status
```



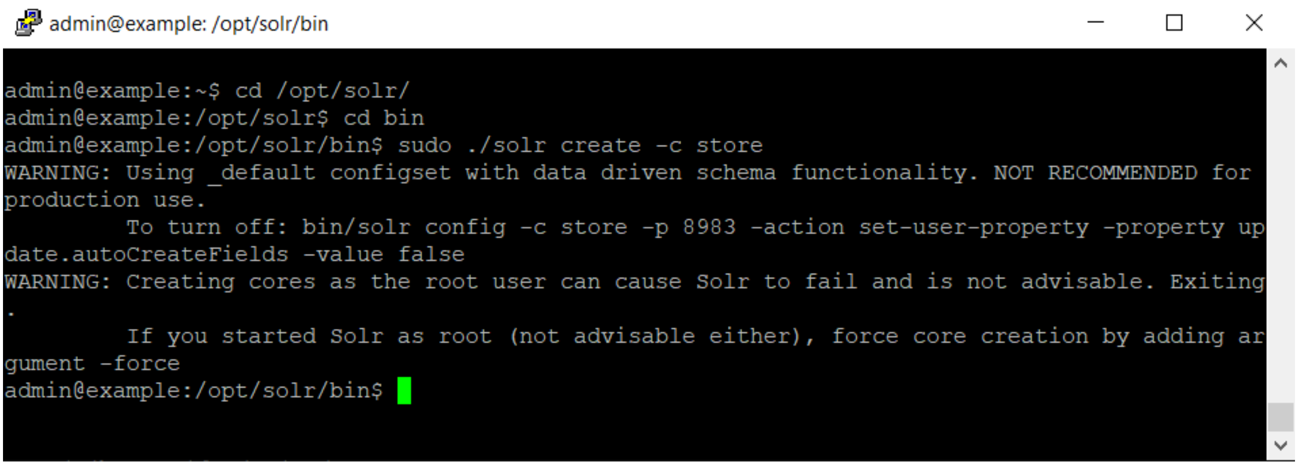
```
admin@example:~$ sudo service solr status
● solr.service - LSB: Controls Apache Solr as a Service
     Loaded: loaded (/etc/init.d/solr; generated)
lines 1-2...skipping...
● solr.service - LSB: Controls Apache Solr as a Service
     Loaded: loaded (/etc/init.d/solr; generated)
     Active: active (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/SUCC>
        CPU: 16ms

Jun 22 09:28:45 example systemd[1]: Starting LSB: Controls Apache Solr as a Ser>
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 u>
Jun 22 09:28:49 example systemd[1]: Started LSB: Controls Apache Solr as a Serv>
```

Рисунок 83 – Пример выполнения команды проверки статуса работы программы Solr

8. Создать хранилище программы Solr (пример выполнения команды см. Рисунок 84):

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

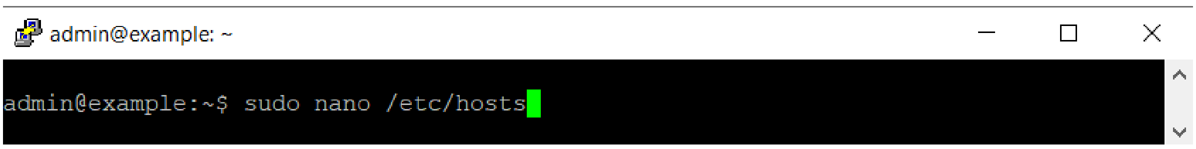


```
admin@example:~$ cd /opt/solr/
admin@example:/opt/solr$ cd bin
admin@example:/opt/solr/bin$ sudo ./solr create -c store
WARNING: Using _default configset with data driven schema functionality. NOT RECOMMENDED for production use.
         To turn off: bin/solr config -c store -p 8983 -action set-user-property -property update.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 argument -force
admin@example:/opt/solr/bin$
```

Рисунок 84 – Пример выполнения команды создания хранилища программы Solr

9. Найти и открыть для коректировки файл «hosts» (пример выполнения команды см. Рисунок 85):

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



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

Рисунок 85 – Пример выполнения команды открытия файла для корректировки «hosts»

Проверить наличие, и если нет, дописать в файл «hosts» следующую строку (пример файла см. Рисунок 106):

```
127.0.0.1 solrsrv
```

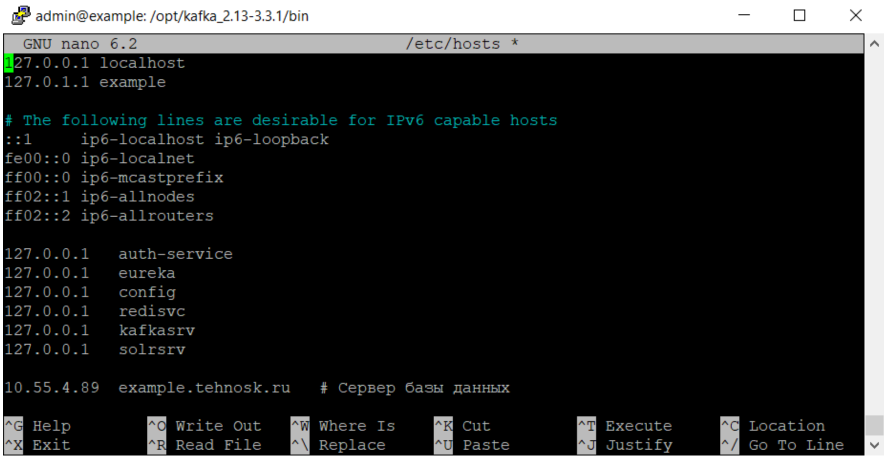


Рисунок 86 – Пример корректировки файла «hosts»

2.3.7 Установка и настройка микросервисов

1. Скопировать и разархивировать файл «jre1.8.0_201-amd64.rar» (из состава дистрибутива «САКУРА PRO») в папку «/usr/java/» (см. Рисунок 87).

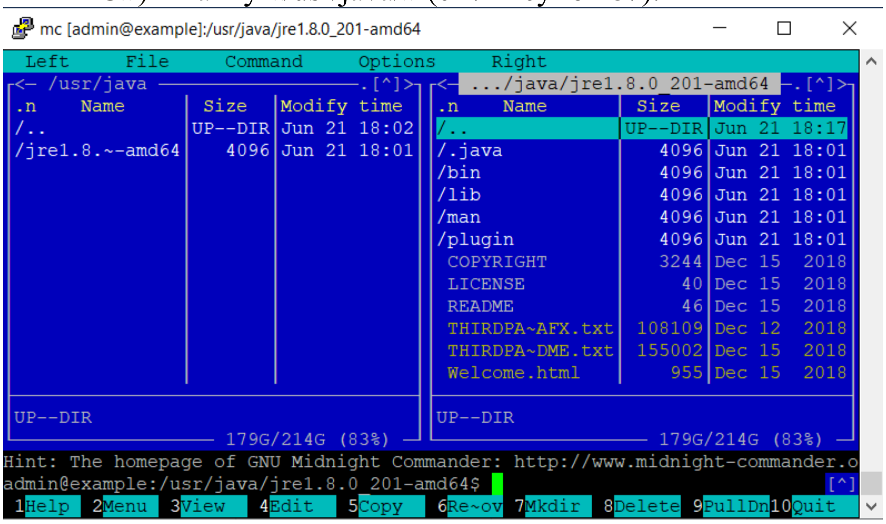


Рисунок 87 – Пример размещения каталога «jre1.8.0_201-amd64»

2. Скопировать микросервисы «САКУРА PRO» (из состава дистрибутива «САКУРА PRO») в папку «/opt/Sakura Pro-cloud/apps/» (см. Рисунок 88 и Рисунок 89).

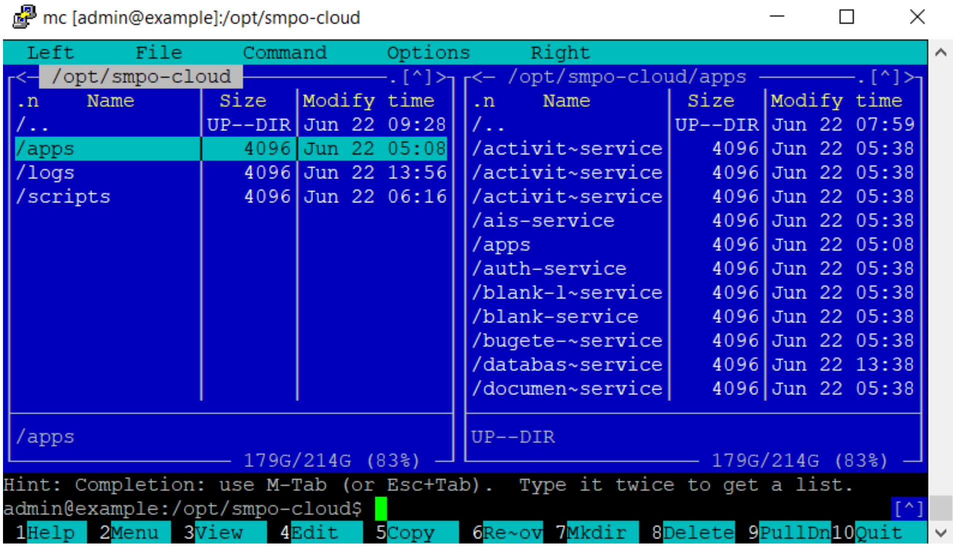


Рисунок 88 – Пример размещения каталога «apps»

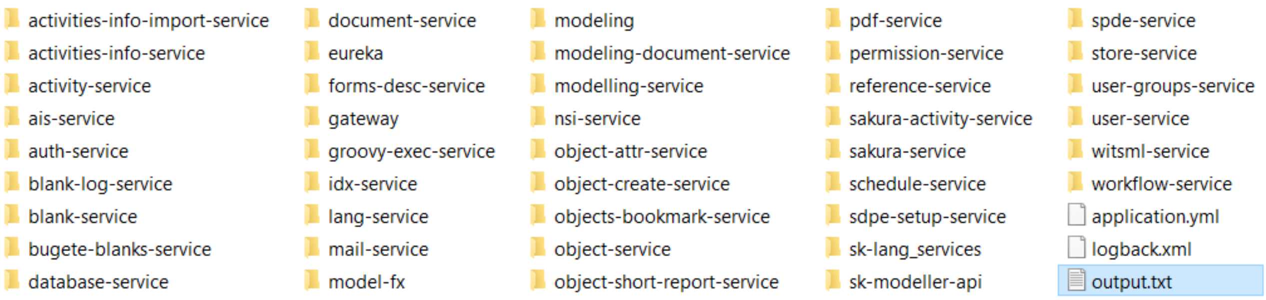


Рисунок 89 – Перечень каталогов с микросервисами системы

3. Скопировать скрипты «САКУРА PRO» (из состава дистрибутива «САКУРА PRO») в папку «/opt/Sakura Pro-cloud/scripts» (см. Рисунок 90 и Рисунок 91).

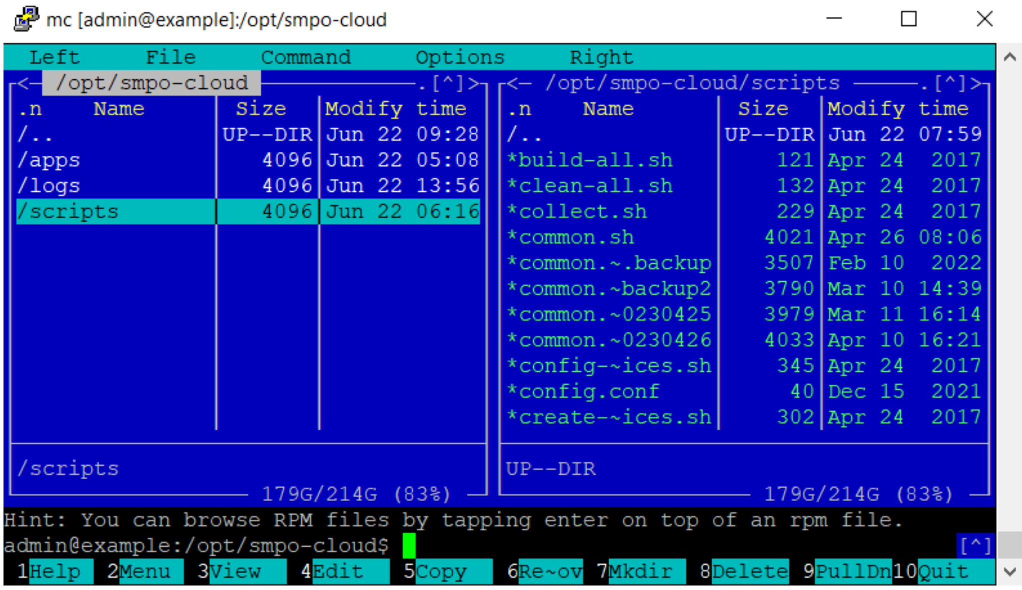


Рисунок 90 – Пример размещения каталога «scripts»

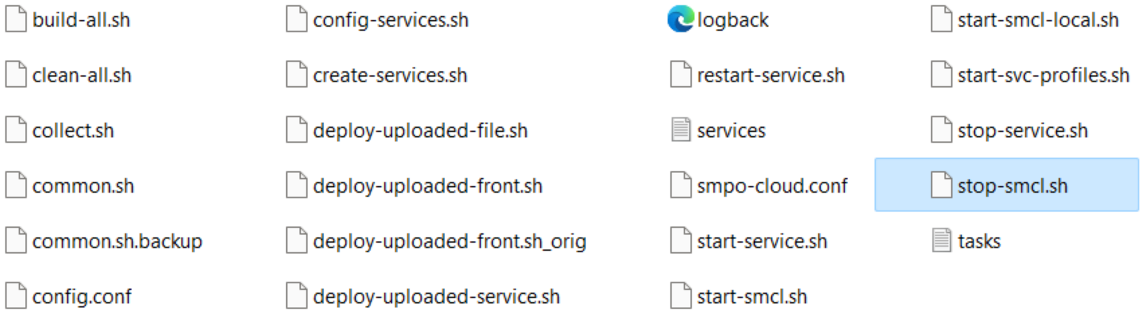


Рисунок 91 – Перечень скриптов системы

4. В скрипте «common.sh» именить значение переменной «JAVA_PATH_TO_BIN», указав в ней путь к каталогу «jre1.8.0_201-amd64/bin/» (см. Рисунок 92):

```
sudo nano /opt/Sakura Pro-cloud/scripts/common.sh
```

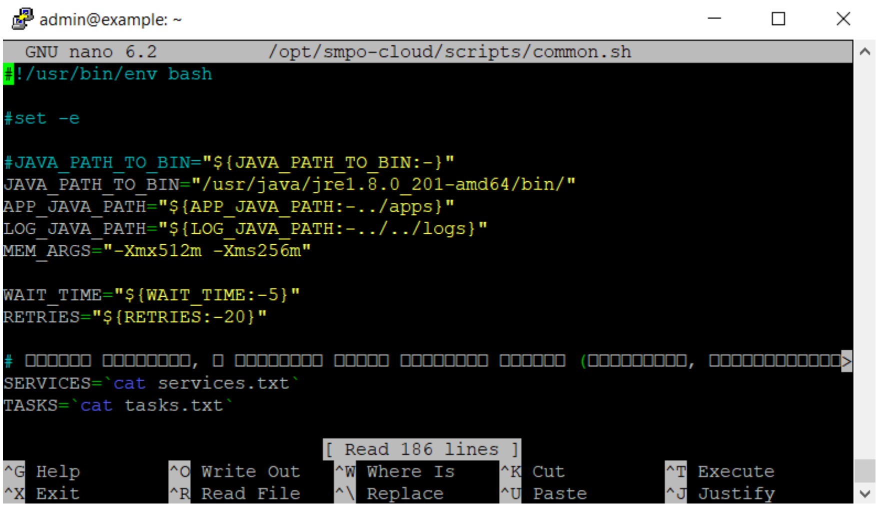


Рисунок 92 – Пример корректировки файла «common.sh»

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-Sakura Pro:
    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={%DOMAIUSER%},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/Sakura Pro-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/Sakura Pro-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/Sakura Pro-cloud» (в данном подкаталоге сервисы будут автоматически создавать log-файлы своей работы) (см. Рисунок 93).

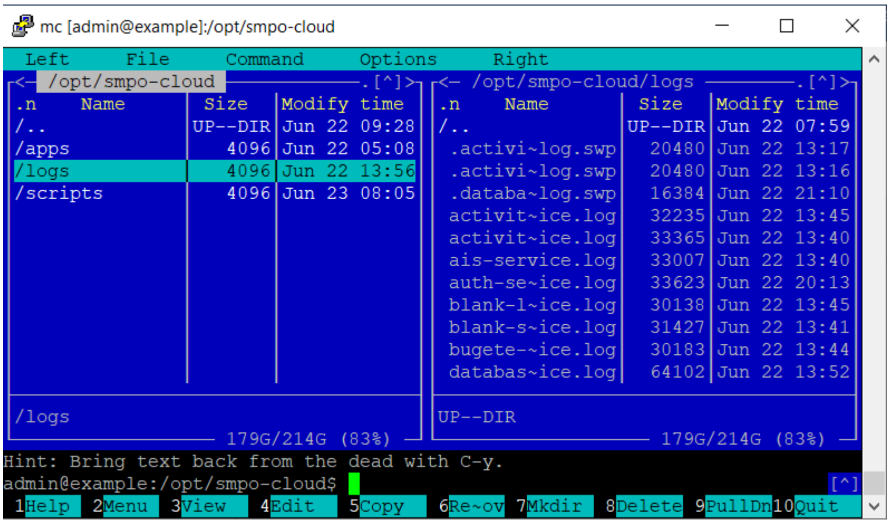


Рисунок 93 – Пример размещения каталога «logs»

7. Для запуска системы необходимо из папки «/opt/Sakura Pro-cloud/scripts» запустить следующий скрипт – начнут запускаться микросервисы и в папке «logs» начнут создаваться log-файлы (пример выполнения команды см. Рисунок 94):

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

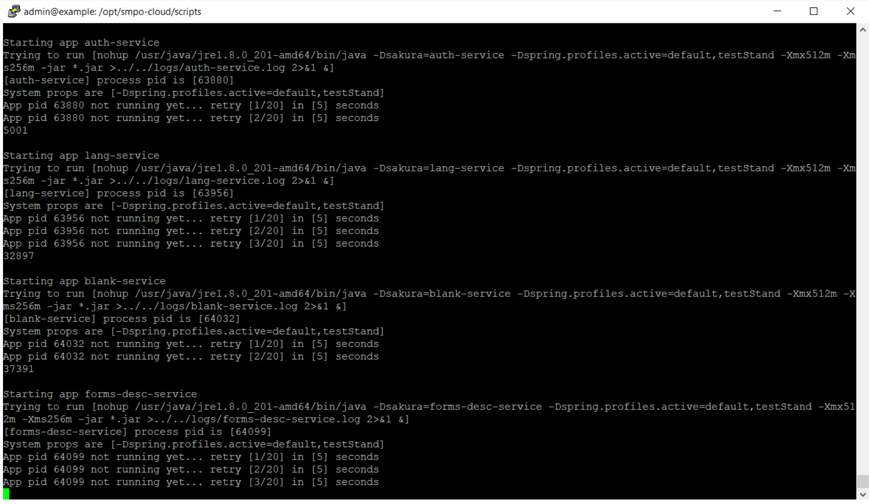


Рисунок 94 – Пример выполнения команды по запуску микросервисов «САКУРА PRO»

2.3.8 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/ubuntu 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 «САКУРА PRO») в каталог /home/sakura_docker (см. Рисунок 95)

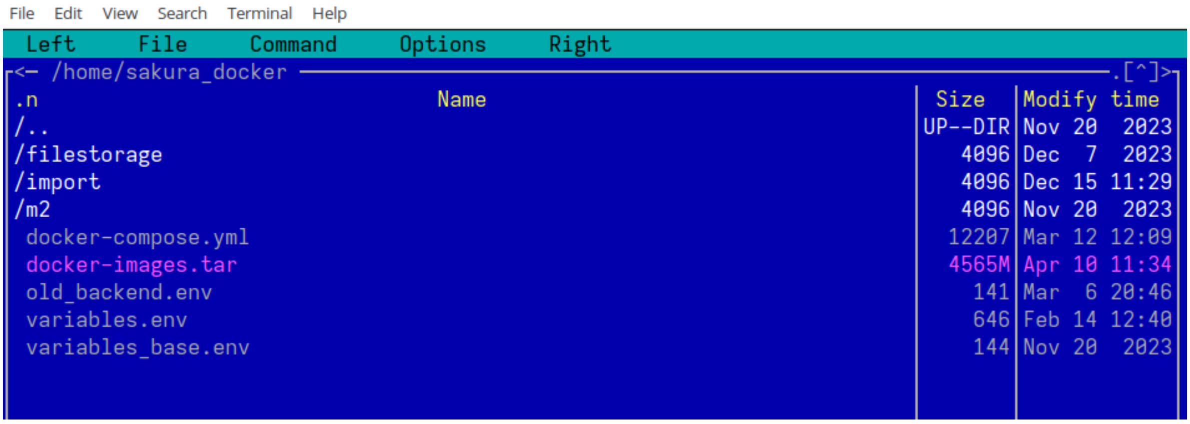


Рисунок 95 – Перечень папок и файлов каталога «/home/sakura_docker»

6. Запустить команду по развертыванию и запуску дополнительных микросервисов «САКУРА PRO» через Docker:

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

2.3.9 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) «САКУРА PRO» (запустить Web-клиента «САКУРА PRO»). Для запуска АРМ «САКУРА PRO» необходимо открыть Web-браузер и в адресной строке ввести запрос к серверу вида (пример файла см. Рисунок 96):

```
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 «САКУРА PRO» (см. Рисунок 96).

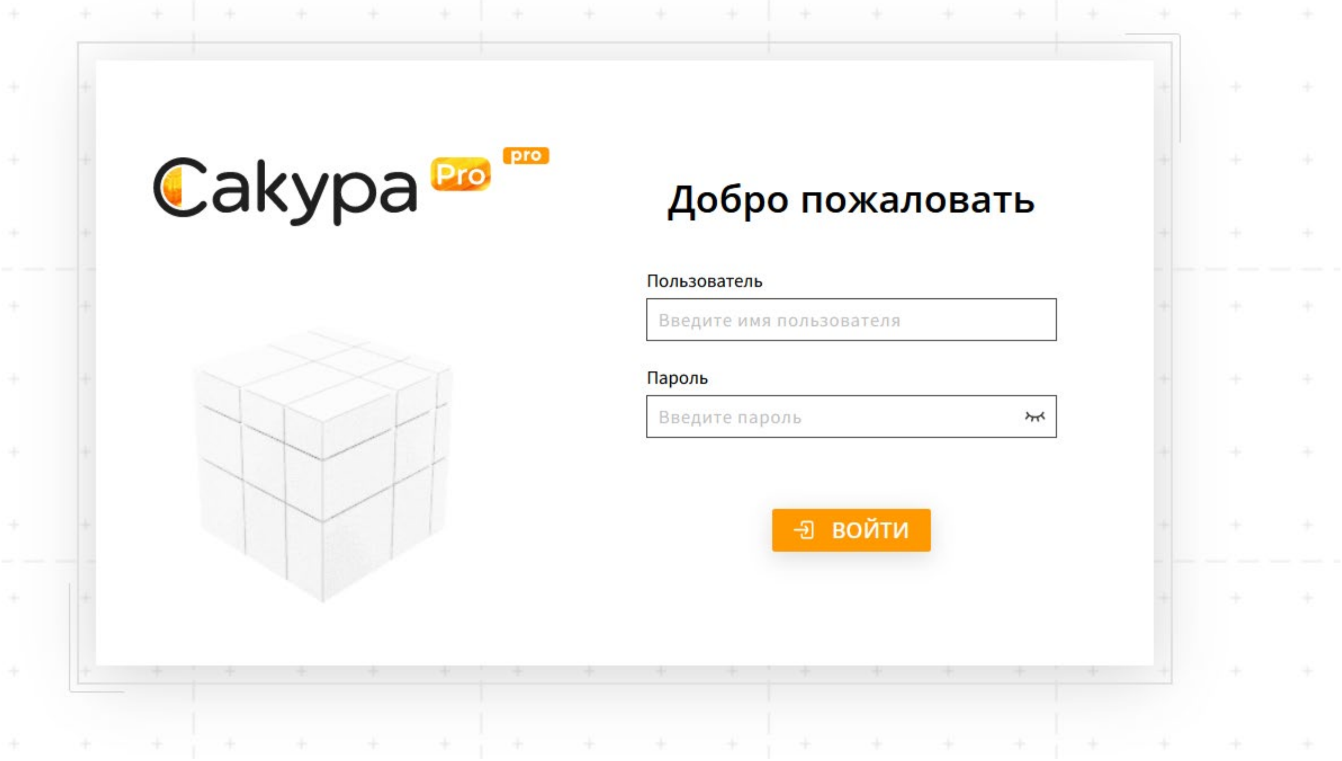


Рисунок 96 – Пример проверки работоспособности сервера доступа, контроля и обработки данных (микросервисов). Запуск Web-клиента «САКУРА PRO». Окно авторизации

При успешной авторизации Администратор или Пользователь «САКУРА PRO» успешно входит в АРМ «САКУРА PRO». В зависимости от назначенных прав Администратору или Пользователю «САКУРА PRO» будет доступны определенные режимы и функции системы (см. Рисунок 97).

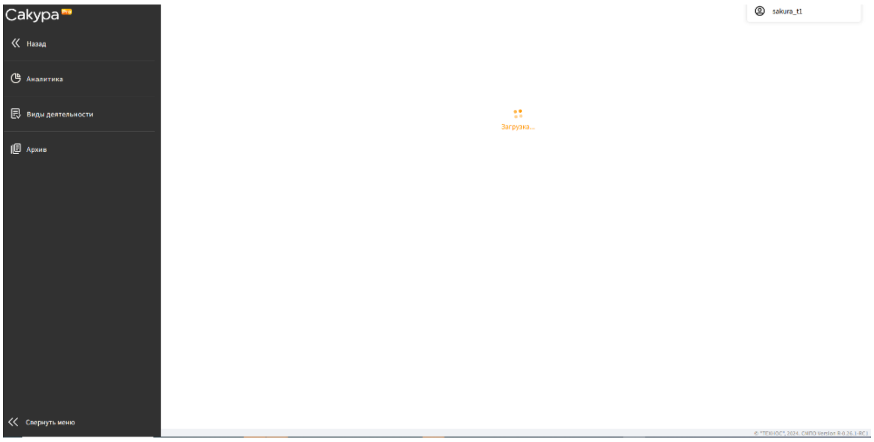


Рисунок 97 – Пример проверки работоспособности сервера доступа, контроля и обработки данных (микросервисов). Запуск Web-клиента «САКУРА PRO». АРМ

2.3.10 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 6.

Таблица 6 – 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 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.
Виртуальныесерверы	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 осуществлятся 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)).

2.3.11 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 9.

Таблица 7 – 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.

Тип	**Резервная единица оборудования**	**Действия по восстановлению**
1	**2**	**3**
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.

2.4 Установка и настройка сервера бизнес-аналитики Apache Superset

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

2.4.1 Настройка файла Hosts

1. Найти и открыть для корректировки файл «hosts» (пример выполнения команды см. Рисунок 98):

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

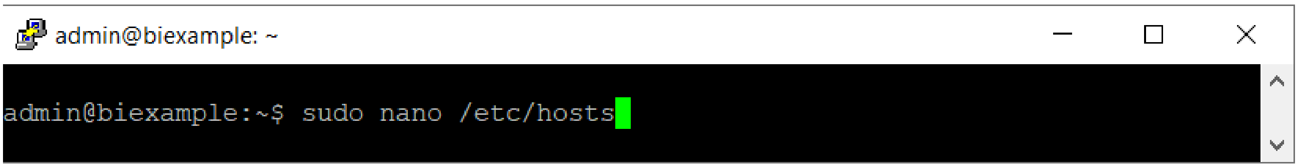


Рисунок 98 – Пример выполнения команды открытия для корректировки файла «hosts»

2. Дописать в файл «hosts» следующие строки (пример выполнения команды см. Рисунок 99):

```
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.
```

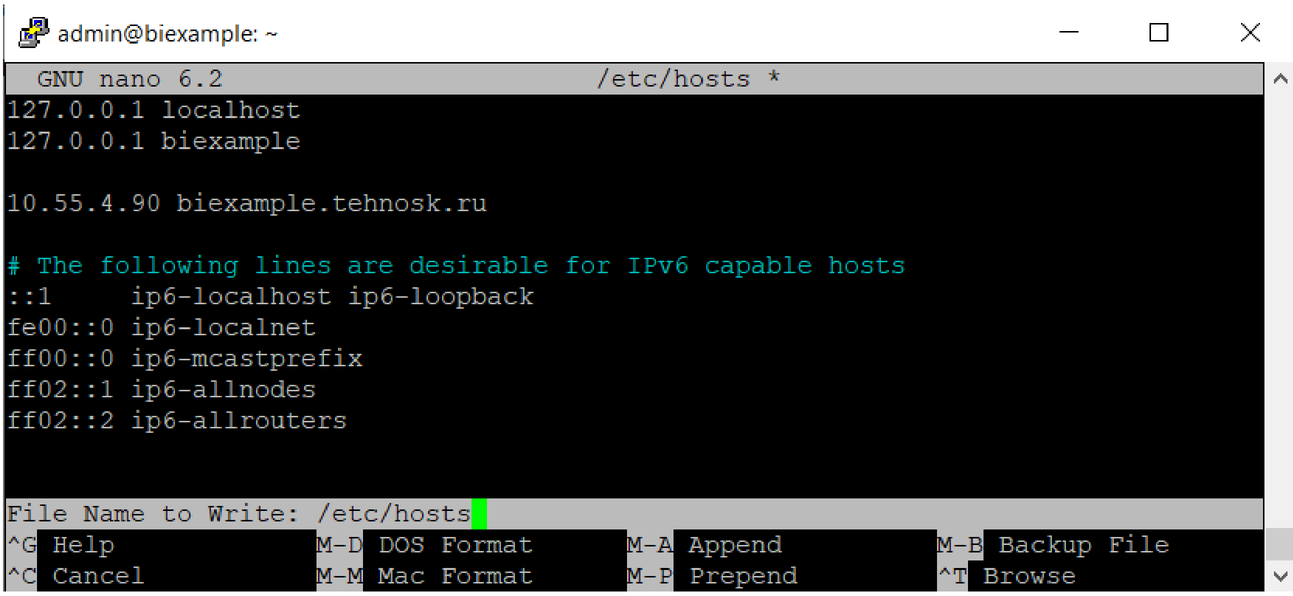


Рисунок 99 – Пример корректировки файла «hosts»

2.4.2 Установка и настройка сервера «NGINX»

1. Обновить списки пакетов операционной системы из депозитариев (пример выполнения команды см. Рисунок 100):

```
sudo apt update
```

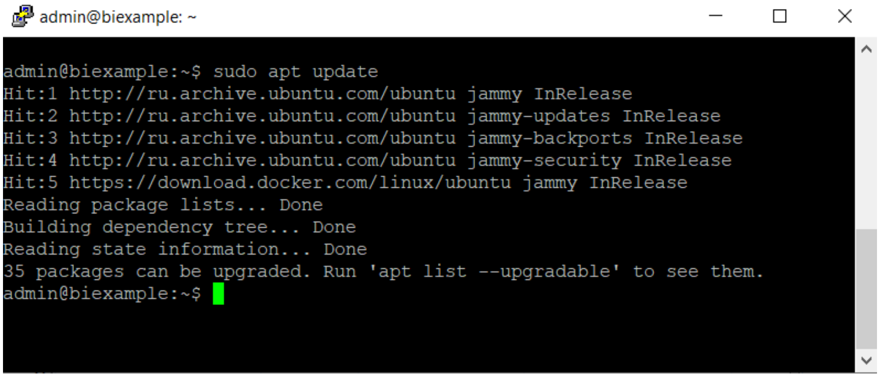


Рисунок 100 – Пример выполнения команды обновления пакетов операционной системы

2. Установить Nginx (пример выполнения команды см. Рисунок 101 и Рисунок 102):

```
sudo apt install nginx
```

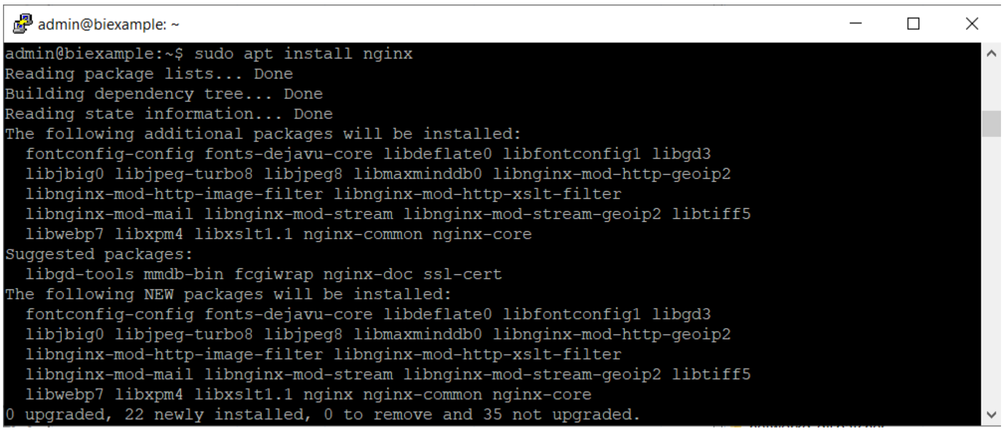


Рисунок 101 – Пример выполнения команды установки Nginx. Начало

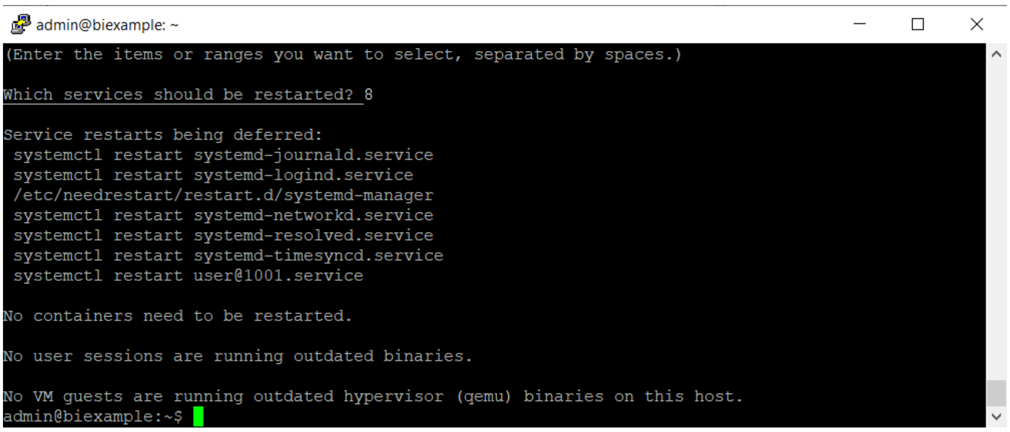


Рисунок 102 – Пример выполнения команды установки Nginx. Окончание

3. Добавить программу Nginx в автозагрузку (пример выполнения команды см. Рисунок 103):

```
sudo systemctl enable nginx
```

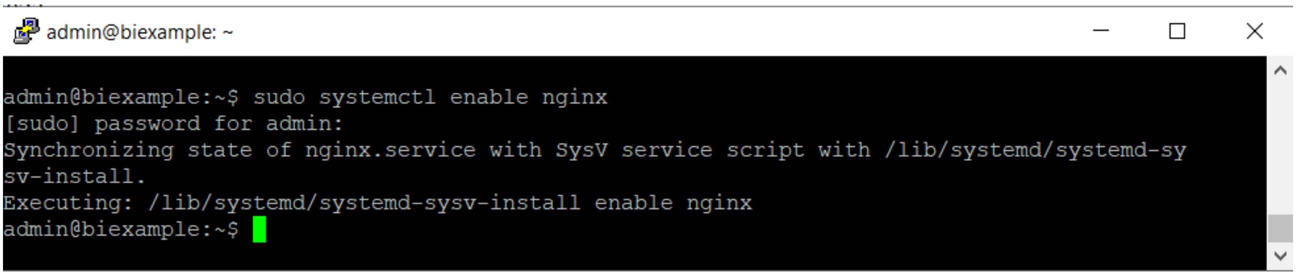


Рисунок 103 – Пример выполнения команды добавления программы Nginx в автозагрузку

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 104).

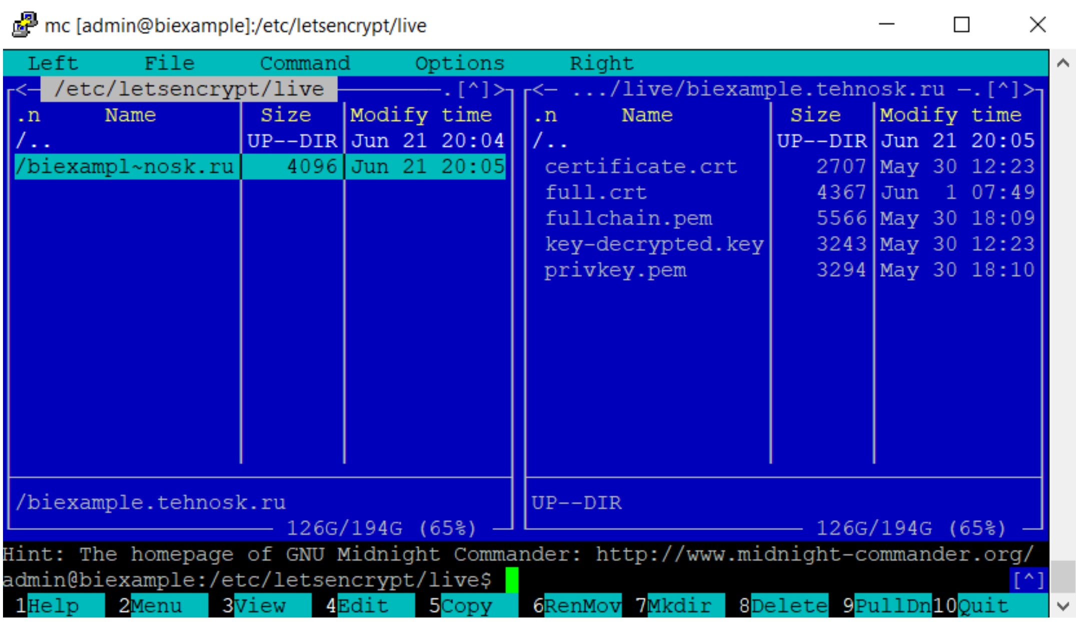


Рисунок 104 – Пример размещения файлов – ключей на сервере

5. Перейти в каталог «conf.d» по пути /etc/nginx/conf.d (пример выполнения команды см. Рисунок 105):

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

Рисунок 105 – Пример выполнения команды по переходу в каталог «/etc/nginx/conf.d»

6. Создать в каталоге «/etc/nginx/conf.d» файл конфигурации «nginx_Sakura Pro.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 106 и Рисунок 107):

```
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 106 – Пример выполнения команды открытия файла «nginx_biSakura Pro.conf»

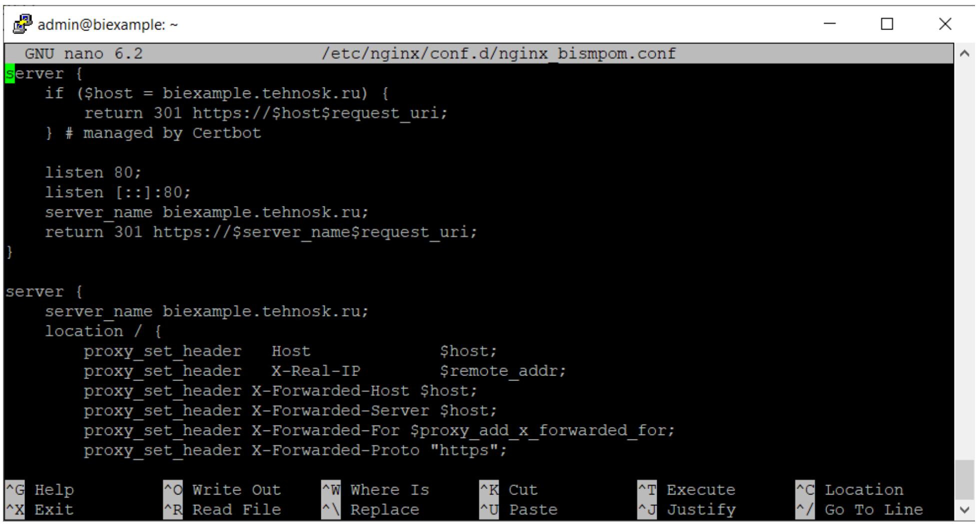


Рисунок 107 – Пример корректировки файла «nginx_biSakura Pro.conf»

7. Запустить Nginx (пример выполнения команды см. Рисунок 108):

```
sudo service nginx start
```

Рисунок 108 – Пример выполнения команды запуска программы Nginx

8. Проверить статус работы Nginx (пример выполнения команды см. Рисунок 109):

```
sudo service nginx status
```

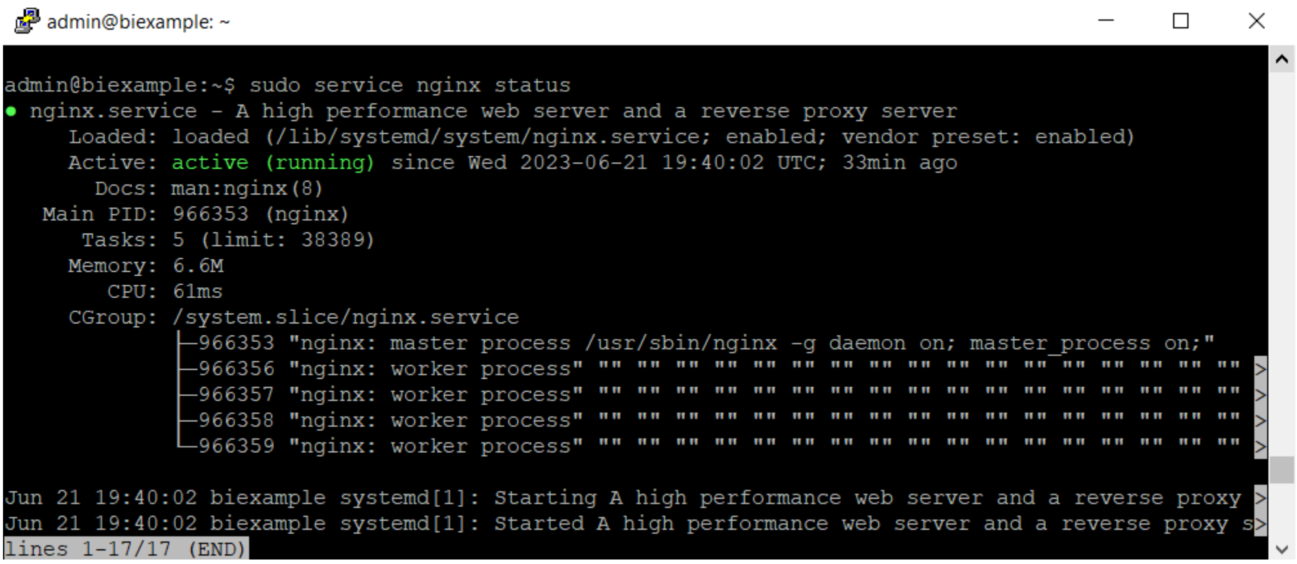


Рисунок 109 – Пример выполнения команды проверки статуса работы программы Nginx

2.4.3 Установка Docker

1. Обновить существующий список пакетов операционной системы (пример выполнения команды см. Рисунок 110):

```
sudo apt update
```

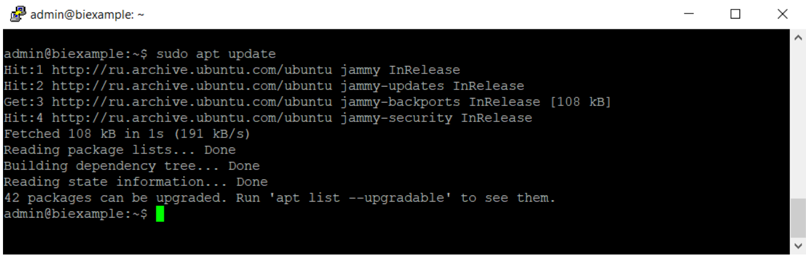


Рисунок 110 – Пример выполнения команды обновления списка пакетов операционной системы

2. Установить несколько обязательных пакетов, которые позволяют apt использовать пакеты по HTTPS (пример выполнения команды см. Рисунок 111 и Рисунок 112):

```
sudo apt install apt-transport-https ca-certificates curl software-properties-
common
```

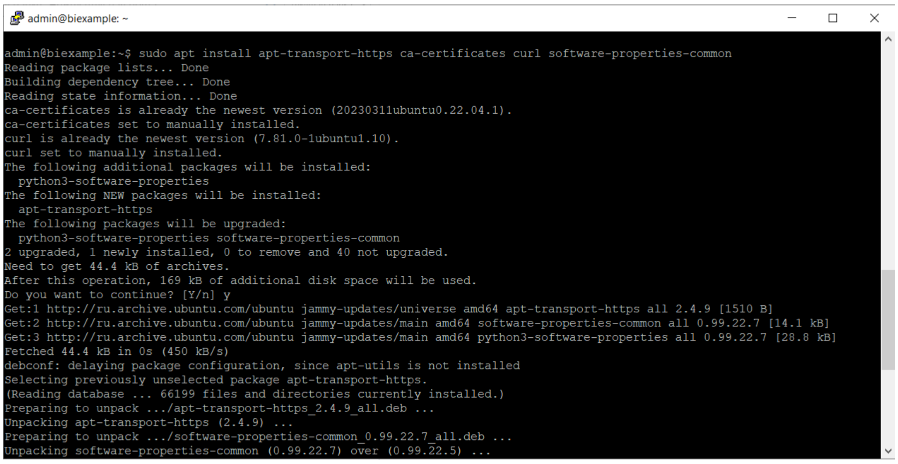


Рисунок 111 – Пример выполнения команды установки обязательных пакетов. Начало

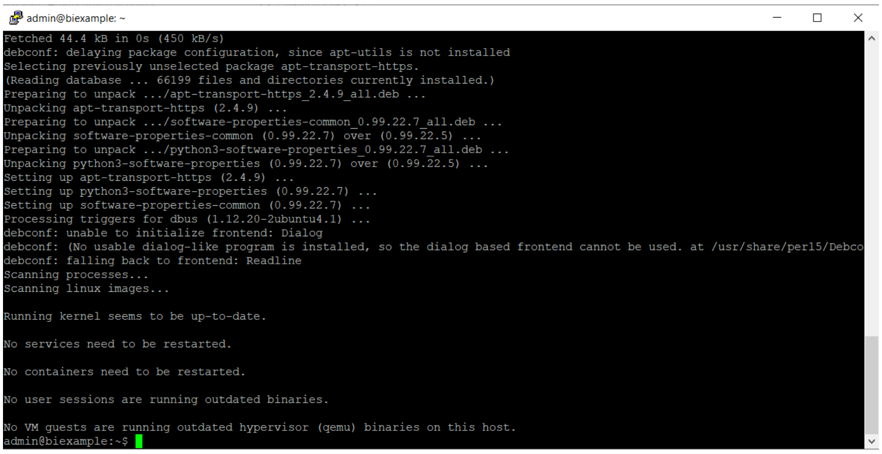


Рисунок 112 – Пример выполнения команды установки обязательных пакетов. Окончание

3. Добавить ключ GPG официального репозитория Docker в операционную систему (пример выполнения команды см. Рисунок 113):

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/docker-archive-keyring.gpg
```

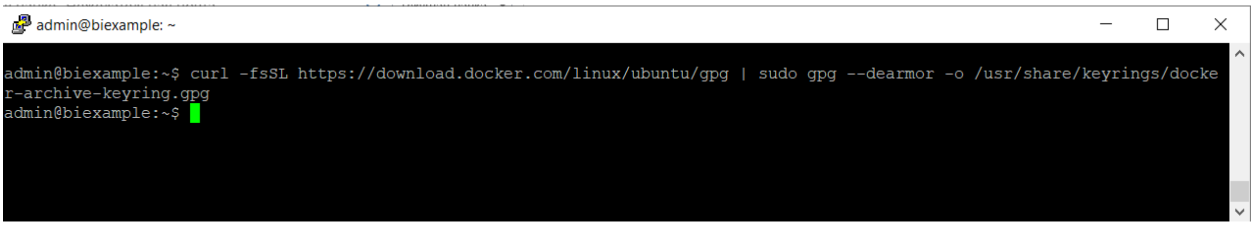


Рисунок 113 – Пример выполнения команды добавления ключа GPG официального репозитория Docker в операционную систему

4. Добавить репозиторий Docker в операционную систему (пример выполнения команды см. Рисунок 114):

```
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-
archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs)
stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

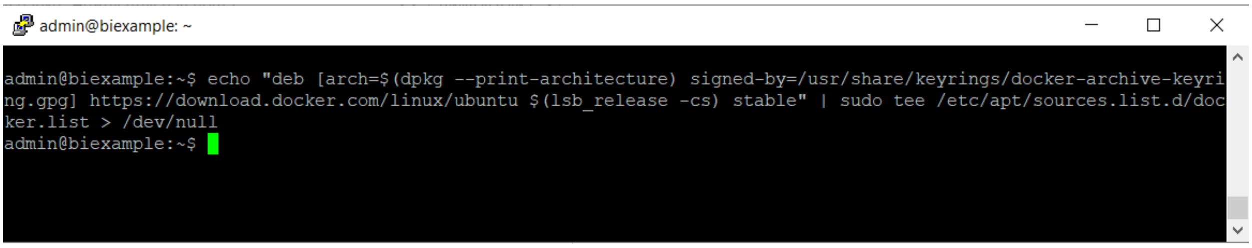


Рисунок 114 – Пример выполнения команды добавления репозитория Docker в операционную систему

5. Обновить список пакетов операционной системы (пример выполнения команды см. Рисунок 115):

```
sudo apt update
```

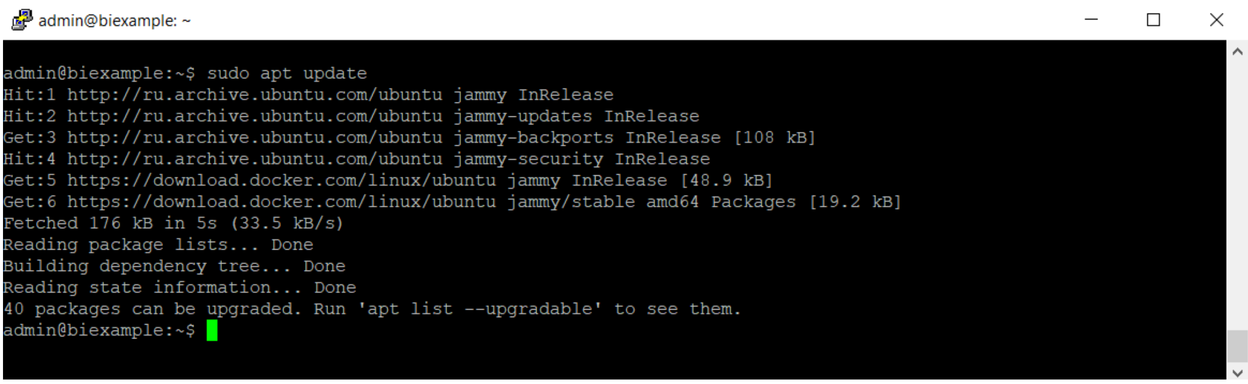


Рисунок 115 – Пример выполнения команды обновления списка пакетов операционной системы

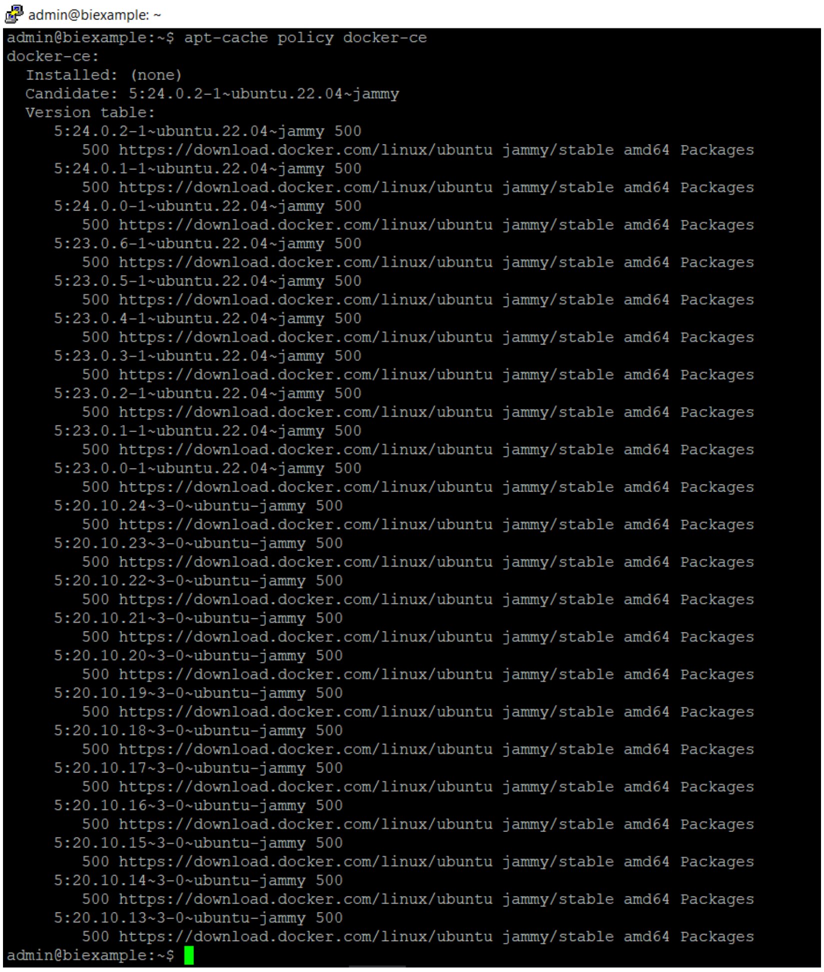
6. Убедиться, что установка будет из репозитория Docker, а не Ubuntu (пример выполнения команды см. Рисунок 116):

```
apt-cache policy docker-ce
```

Должны увидеть следующий результат выполнения команды:

```
 docker-ce:
 Installed: (none)
 Candidate: 5:20.10.14~3-0~ubuntu-jammy
```

```
Version table:
   5:20.10.14~3-0~ubuntu-jammy 500
      500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
   5:20.10.13~3-0~ubuntu-jammy 500
      500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     . . . . . . . . . . . . . .
```

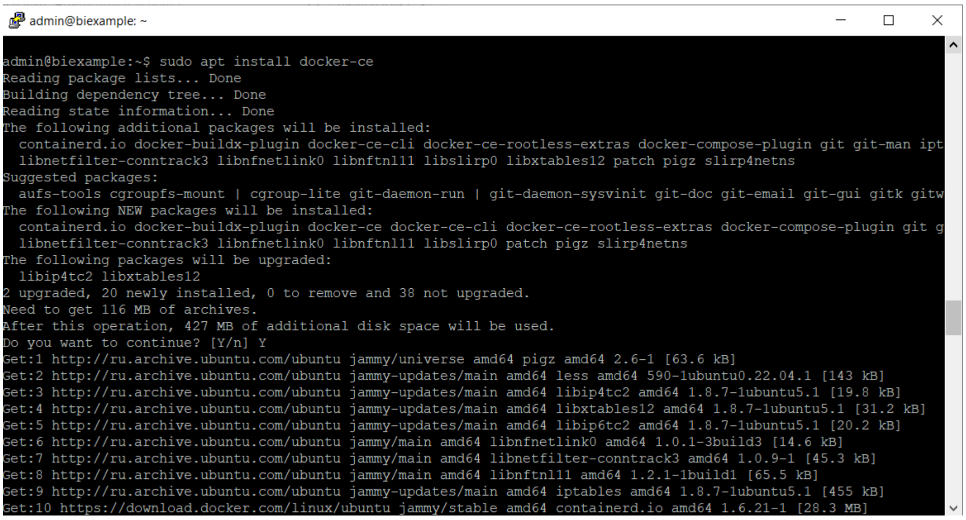


```
admin@biexample: ~
admin@biexample:~$ apt-cache policy docker-ce
docker-ce:
  Installed: (none)
  Candidate: 5:24.0.2-1~ubuntu.22.04~jammy
  Version table:
     5:24.0.2-1~ubuntu.22.04~jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:24.0.1-1~ubuntu.22.04~jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:24.0.0-1~ubuntu.22.04~jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:23.0.6-1~ubuntu.22.04~jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:23.0.5-1~ubuntu.22.04~jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:23.0.4-1~ubuntu.22.04~jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:23.0.3-1~ubuntu.22.04~jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:23.0.2-1~ubuntu.22.04~jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:23.0.1-1~ubuntu.22.04~jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:23.0.0-1~ubuntu.22.04~jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:20.10.24~3-0~ubuntu-jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:20.10.23~3-0~ubuntu-jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:20.10.22~3-0~ubuntu-jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:20.10.21~3-0~ubuntu-jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:20.10.20~3-0~ubuntu-jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:20.10.19~3-0~ubuntu-jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:20.10.18~3-0~ubuntu-jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:20.10.17~3-0~ubuntu-jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:20.10.16~3-0~ubuntu-jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:20.10.15~3-0~ubuntu-jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:20.10.14~3-0~ubuntu-jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:20.10.13~3-0~ubuntu-jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
admin@biexample:~$
```

Рисунок 116 – Пример проверки того, что установка Docker будет вестись из репозитория Docker

7. Установить Docker (пример выполнения команды см. Рисунок 117):

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



```
admin@biexample: ~
admin@biexample:~$ sudo apt install docker-ce
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin git git-man ipt
  libnetfilter-conntrack3 libnfnetlink0 libnftnl11 libslirp0 libxtables12 patch pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitw
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin git g
  libnetfilter-conntrack3 libnfnetlink0 libnftnl11 libslirp0 patch pigz slirp4netns
The following packages will be upgraded:
  libip4tc2 libxtables12
2 upgraded, 20 newly installed, 0 to remove and 38 not upgraded.
Need to get 116 MB of archives.
After this operation, 427 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://ru.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:2 http://ru.archive.ubuntu.com/ubuntu jammy-updates/main amd64 less amd64 590-1ubuntu0.22.04.1 [143 kB]
Get:3 http://ru.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libip4tc2 amd64 1.8.7-1ubuntu5.1 [19.8 kB]
Get:4 http://ru.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libxtables12 amd64 1.8.7-1ubuntu5.1 [31.2 kB]
Get:5 http://ru.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libip6tc2 amd64 1.8.7-1ubuntu5.1 [20.2 kB]
Get:6 http://ru.archive.ubuntu.com/ubuntu jammy/main amd64 libnfnetlink0 amd64 1.0.1-3build3 [14.6 kB]
Get:7 http://ru.archive.ubuntu.com/ubuntu jammy/main amd64 libnetfilter-conntrack3 amd64 1.0.9-1 [45.3 kB]
Get:8 http://ru.archive.ubuntu.com/ubuntu jammy/main amd64 libnftnl11 amd64 1.2.1-1build1 [65.5 kB]
Get:9 http://ru.archive.ubuntu.com/ubuntu jammy-updates/main amd64 iptables amd64 1.8.7-1ubuntu5.1 [455 kB]
Get:10 https://download.docker.com/linux/ubuntu jammy/stable amd64 containerd.io amd64 1.6.21-1 [28.3 MB]
```

Рисунок 117 – Пример выполнения команды утановки Docker

8. Проверить результат установки Docker (пример выполнения команды см. Рисунок 118):

```
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:
enabled)
     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
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

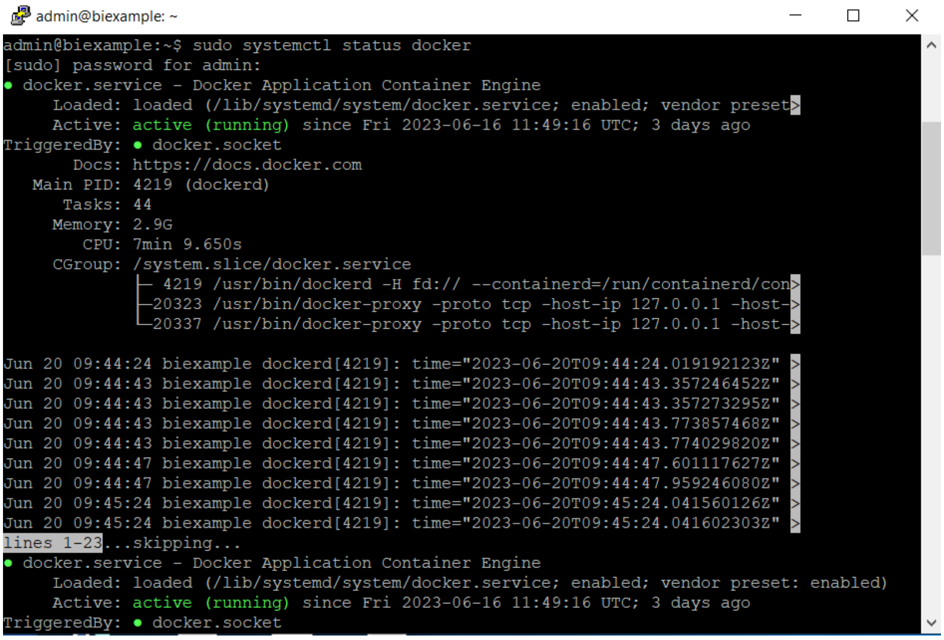


Рисунок 118 – Пример выполнения команды проверки статуса Docker

9. Запустить контейнер (пример выполнения команды см. Рисунок 119):

```
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:bfea6278a0a267fad2634554f4f0c6f31981eea41c553fdf5a83e95a41d40c38
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

.  .  .  .  .  .  .  .  .  .  .
```

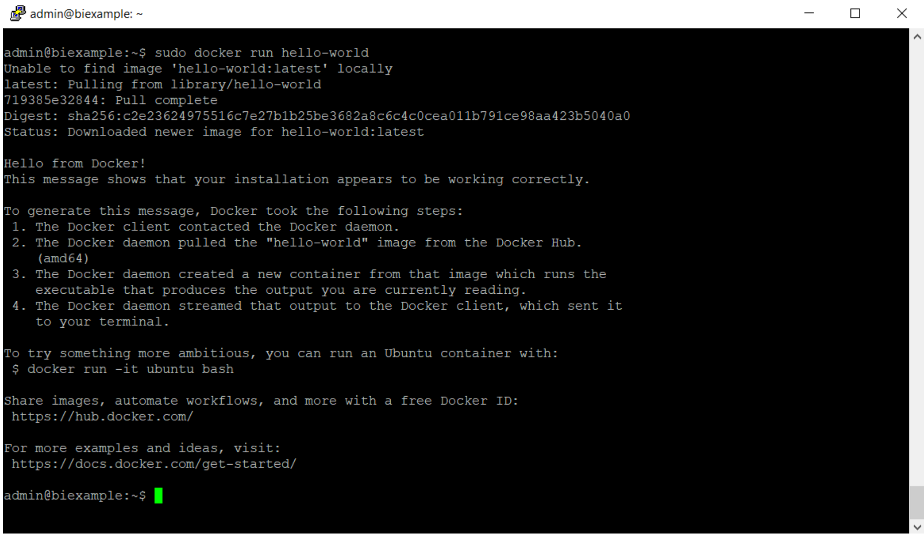


Рисунок 119 – Пример выполнения команды на запуск контейнер

10. Добавить текущего пользователя в группу docker (пример выполнения команды см. Рисунок 120):

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

, где:

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

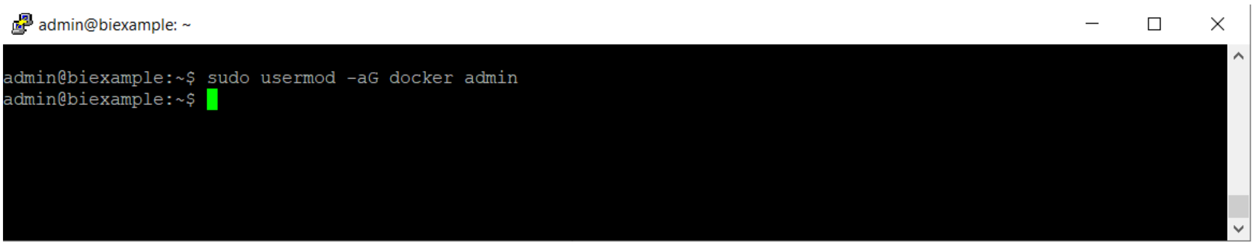


Рисунок 120 – Пример выполнения команды добавления текущего пользователя в группу docker

11. Перелогинится и повторно запустить контейнер (пример выполнения команды см. Рисунок 121):

```
docker run hello-world
```

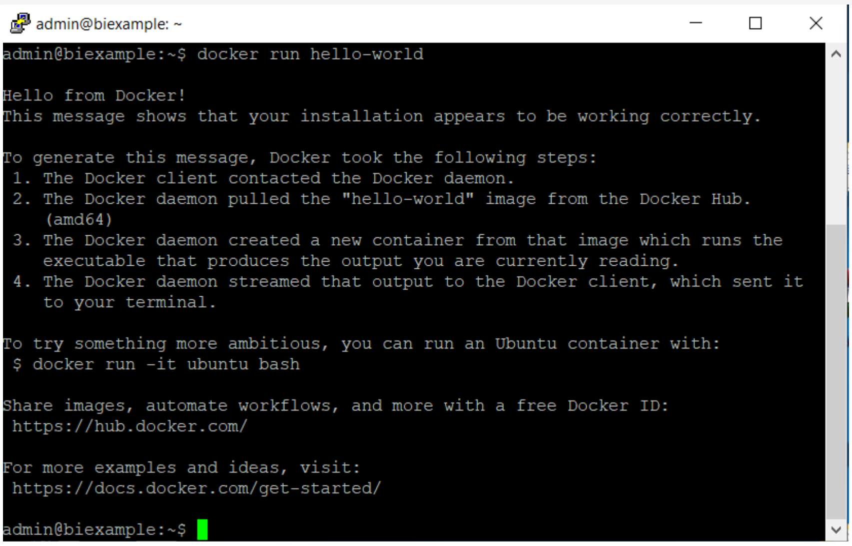


Рисунок 121 – Пример выполнения команды на запуск контейнер

2.4.4 Установка Docker-compose

1. Создать каталог (пример выполнения команды см. Рисунок 122):

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

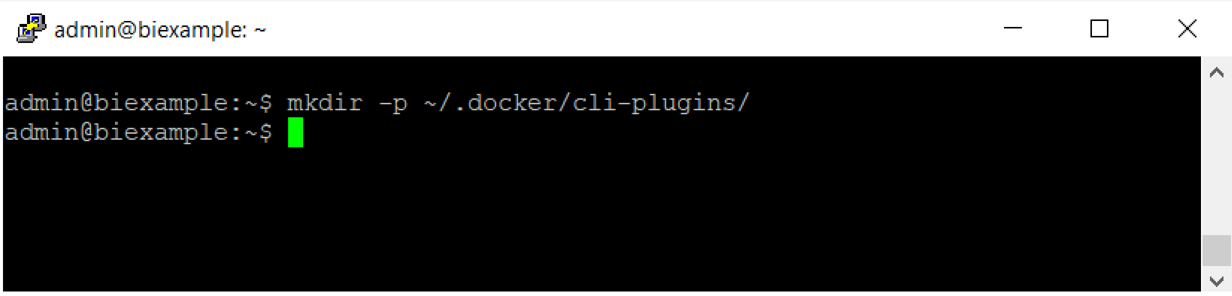


Рисунок 122 – Пример выполнения команды создания каталога

2. Устанавливаем Docker-compose (пример выполнения команды см. Рисунок 123):

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

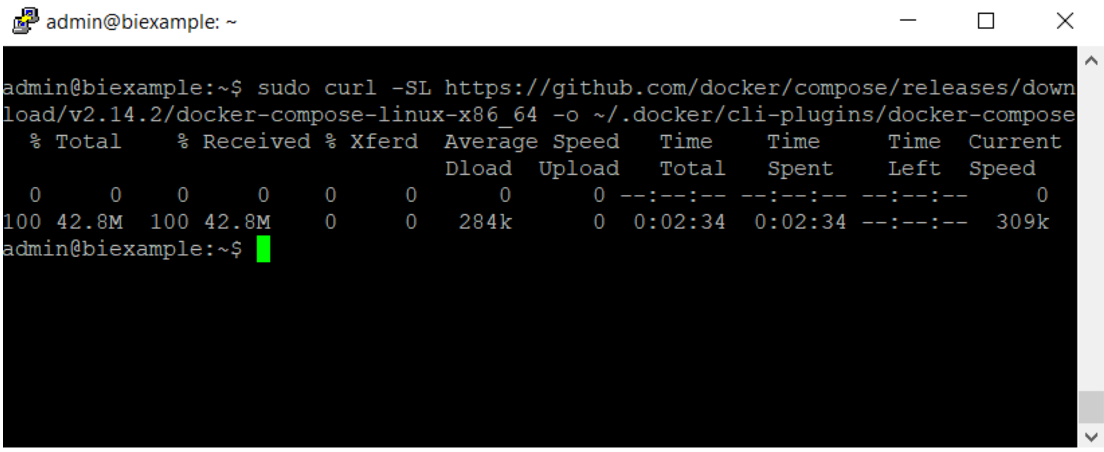


Рисунок 123 – Пример выполнения команды установки Docker-compose

3. Сделать файл запускаемым (пример выполнения команды см. Рисунок 124):

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

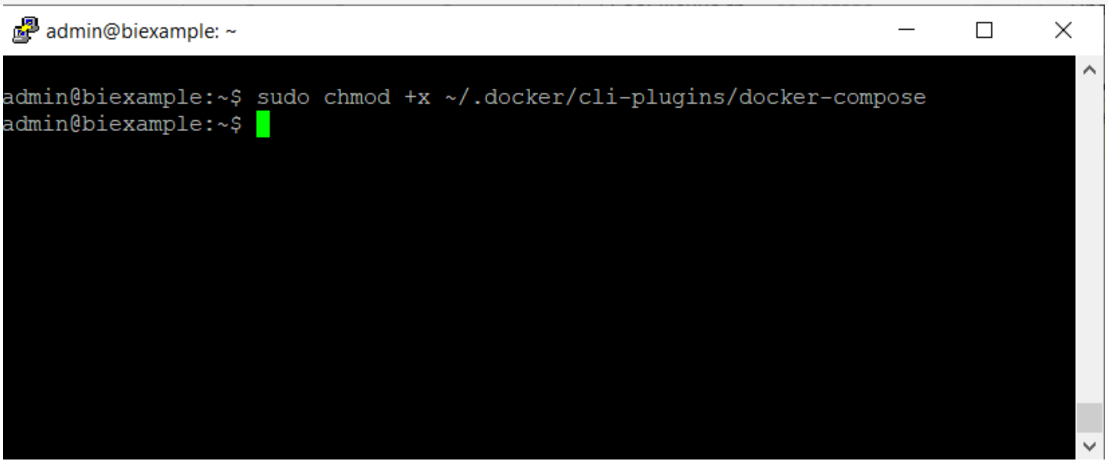


Рисунок 124 – Пример выполнения команды по изменению атрибутов файла

4. Проверить работоспособность Docker-compose (пример выполнения команды см. Рисунок 125):

```
docker compose version
```

Должны увидеть следующий результат выполнения команды:

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

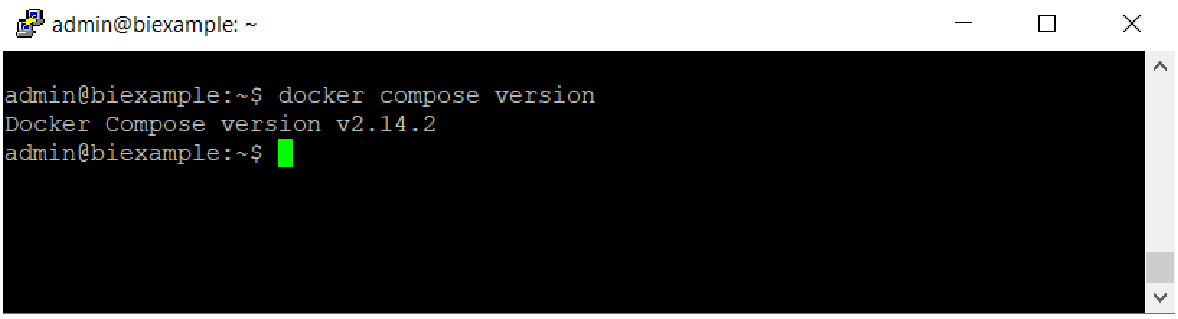


Рисунок 125 – Пример выполнения команды определения версии Docker-compose

2.4.5 Установка Apatch Superset

1. Обновить существующий список пакетов (пример выполнения команды см. Рисунок 126):

```
sudo apt update
```

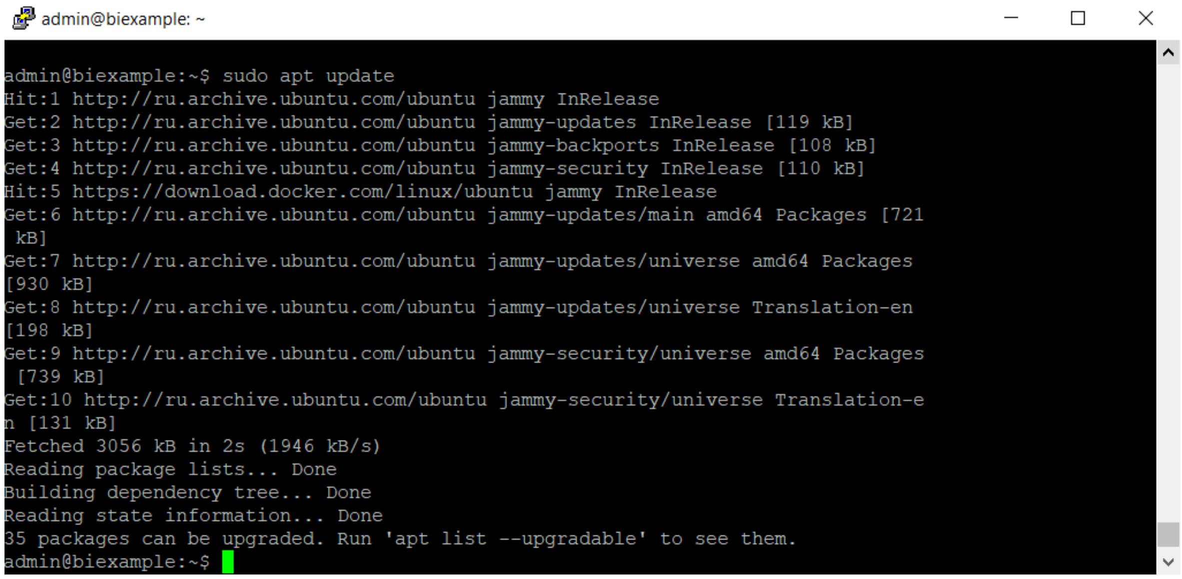


Рисунок 126 – Пример выполения команды обновления пакето операционной системы

2. Распоковать архив «superset.tar.gz» (из состава дистрибутива «САКУРА PRO») по пути home/ (пример выполнения команды см. Рисунок 127):

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

Рисунок 127 – Пример выполнения команды распаковки архива

3. Перейти в каталог установки программного обеспечения (пример выполнения команды см. Рисунок 128):

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

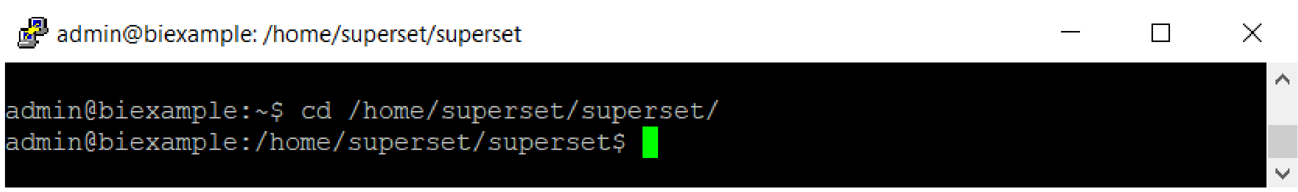


Рисунок 128 – Пример выполнения команды перехода в каталог

4. Запустить программное обеспечение «Apatch Superset» из контейнера (пример выполнения команды см. Рисунок 129):

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

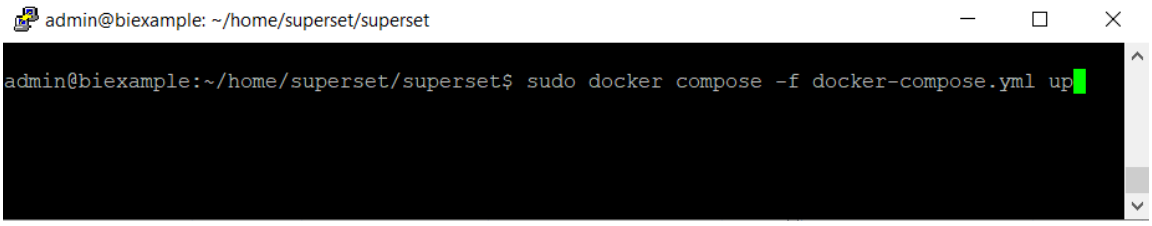


Рисунок 129 – Пример выполнения команды запуска программного обеспечения «Apatch Superset» из контейнера

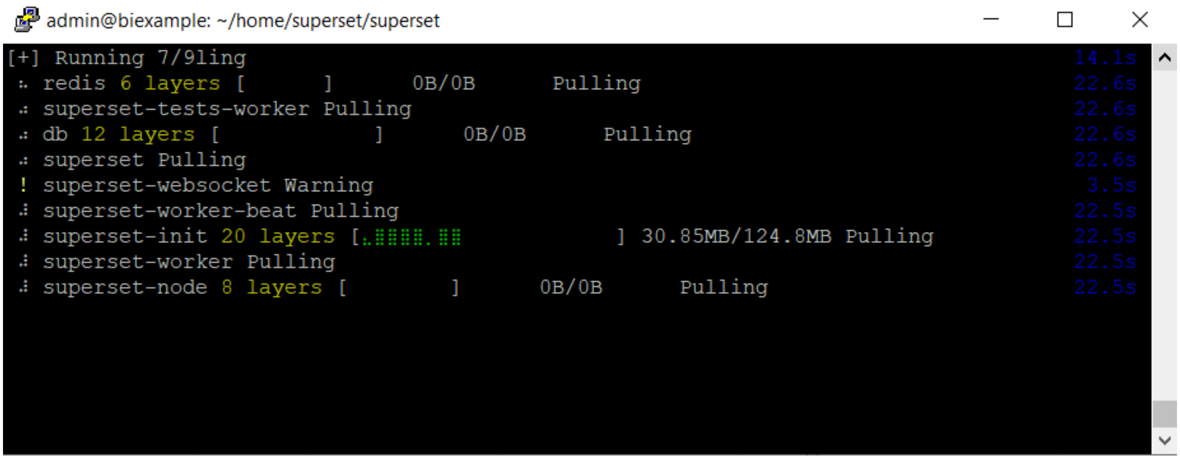


Рисунок 130 – Пример выполнения команды запуска программного обеспечения «Apatch Superset» из контейнера. Продолжение

2.4.6 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 (пример файла см. Рисунок 131):

```
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 131).

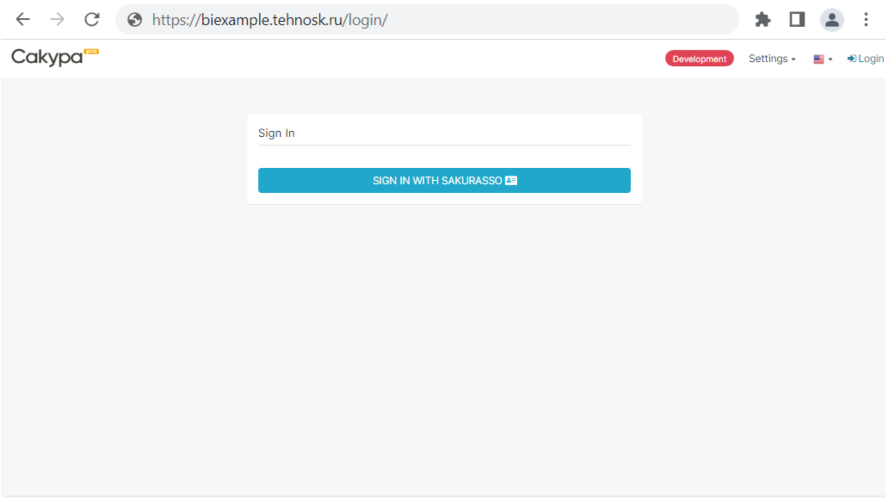


Рисунок 131 – Пример внешнего подключения к серверу бизнес-аналитики Apache Superset

При успешной авторизации осуществдяется вход в систему бизнес-аналитики (см. Рисунок 132).

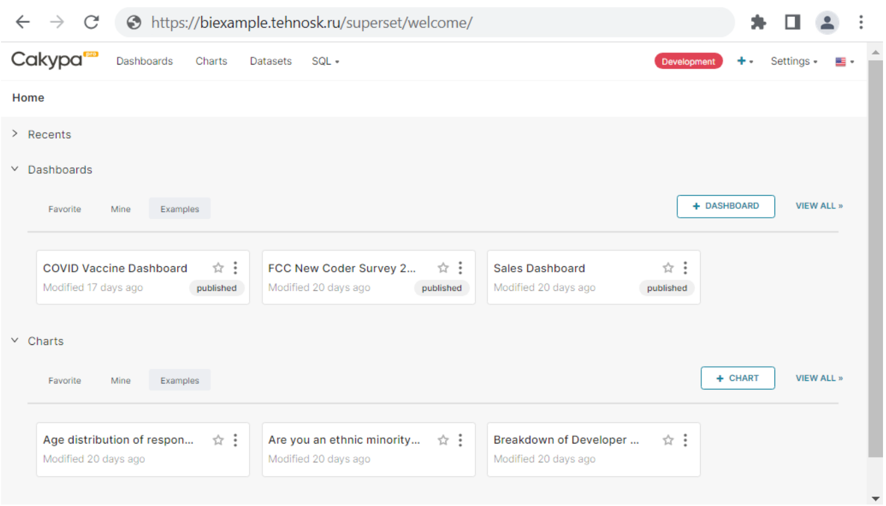


Рисунок 132 – Пример внешнего подключения к серверу бизнес-аналитики Apache Superset

2.4.7 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. Рисунок 133) - откроется окно импорта (см. Рисунок 134).

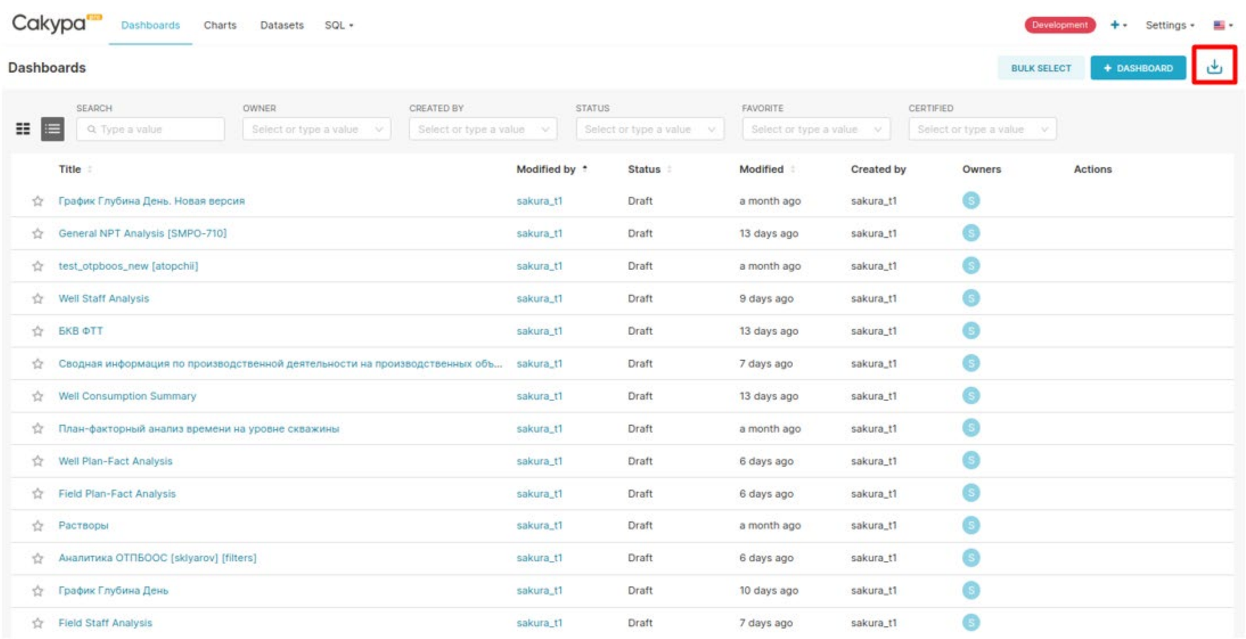


Рисунок 133 – Реестр дашбордов сервера бизнес-аналитики Apache Superset

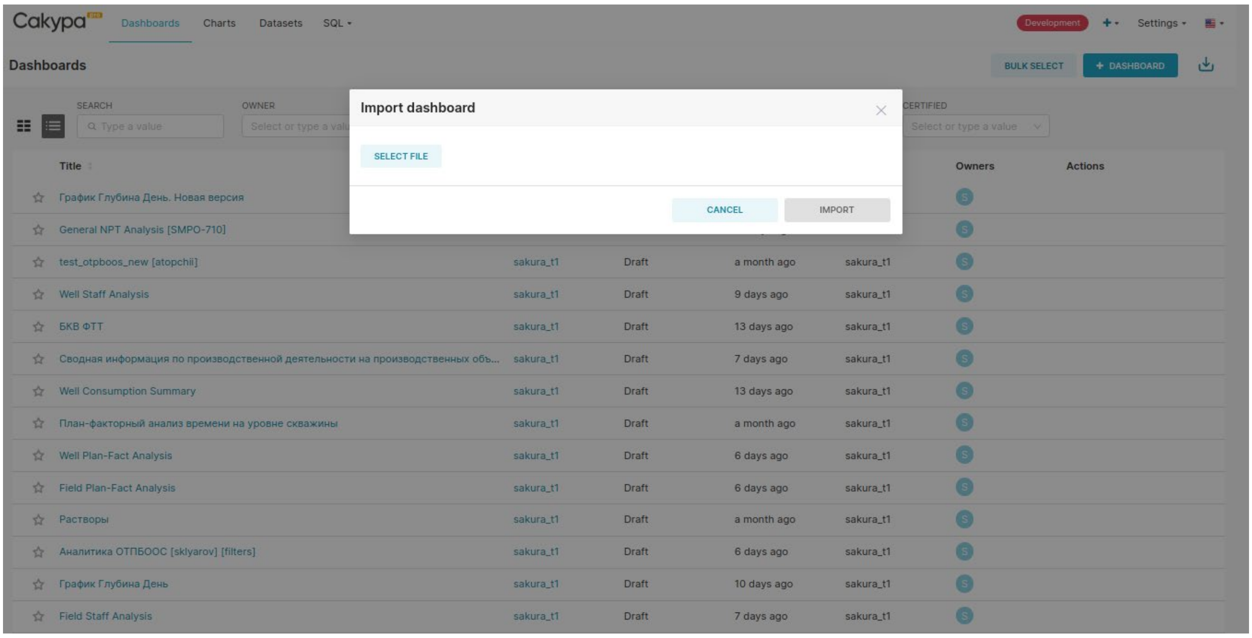


Рисунок 134 – Реестр дашбордов сервера бизнес-аналитики Apache Superset. Импорт дашборда

В окне импорта необходимо нажать на кнопку «Select File» (см. Рисунок 134) – откроется окно выбора файла для импорта (см. Рисунок 135). Необходимо выбрать одни файл – шаблон дашборда и нажать на кнопку «Select». Далее в окне импорта необходимо нажать на кнопку «Import» (см. Рисунок 134) и дождаться импорта (загрузки) нового дашборда в реестре дашбордов (см. Рисунок 133).

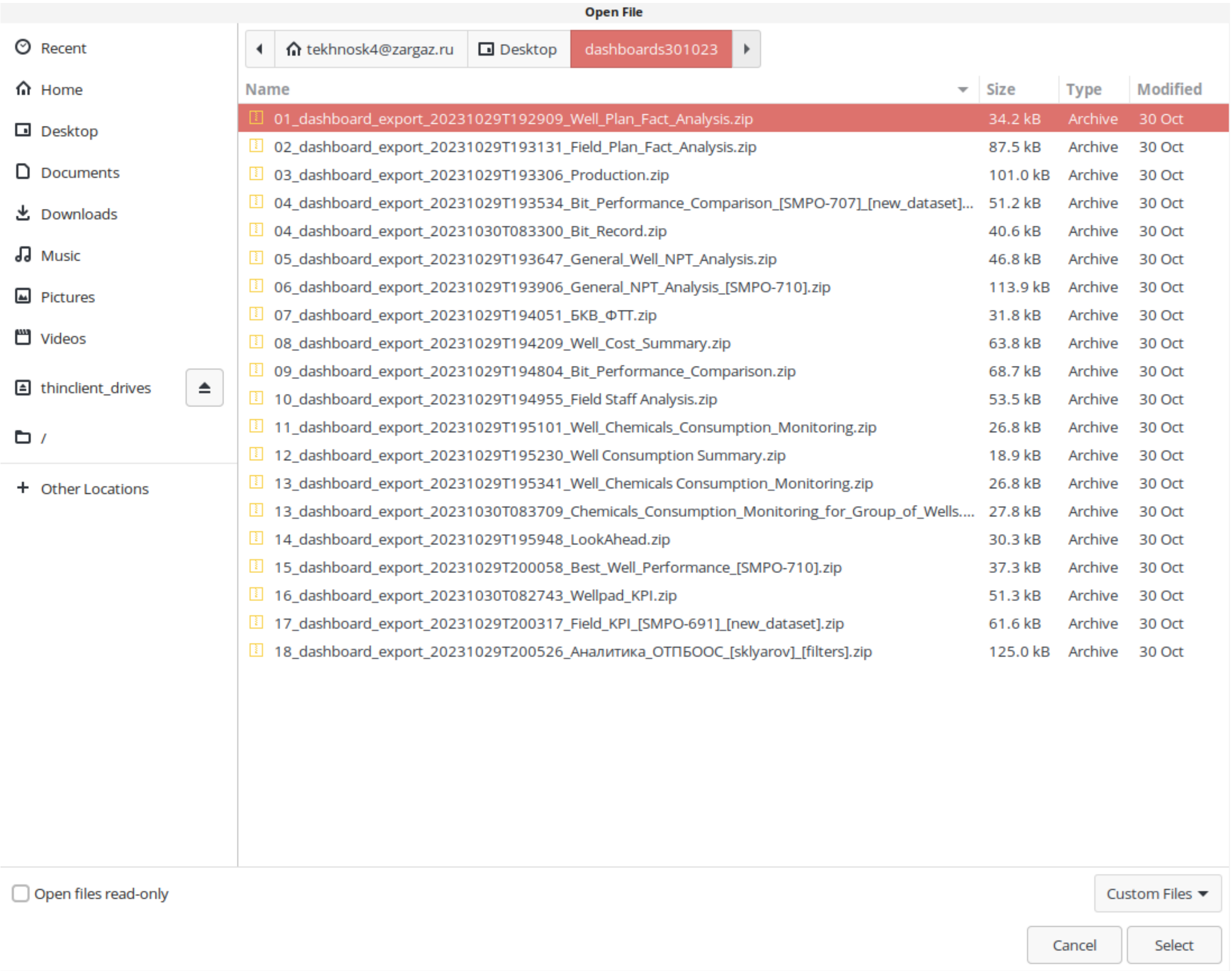


Рисунок 135 – Реестр дашбордов сервера бизнес-аналитики Apache Superset. Окно выбора файла шаблона дашборда

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

2.4.8 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 136) - откроется реестр соединений с базами данных (см. Рисунок 137).

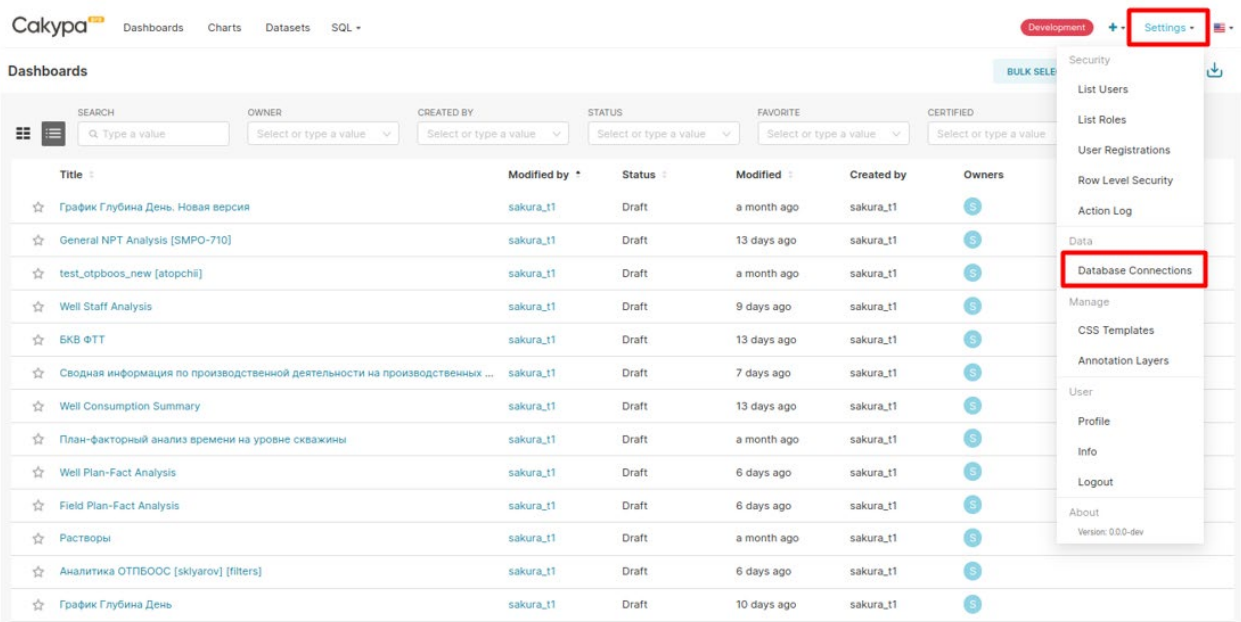


Рисунок 136 – Реестр дашбордов сервера бизнес-аналитики Apache Superset. Переход к реестру соединений с базами данных

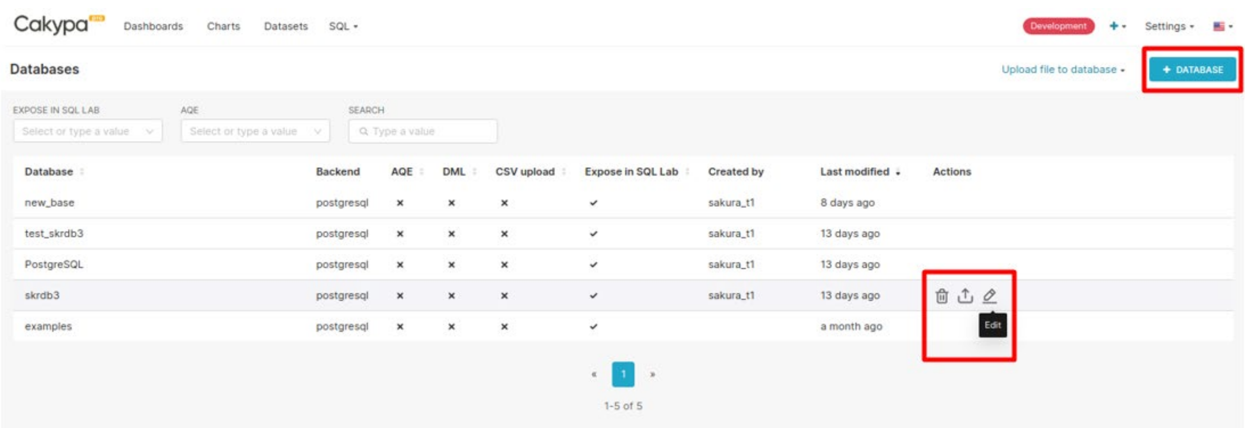


Рисунок 137 – Реестр соединений с базами данных сервера бизнес-аналитики Apache Superset

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 137) – откроется окно ввода и редактирования параметров соединения с базой данных (см. Рисунок 138).

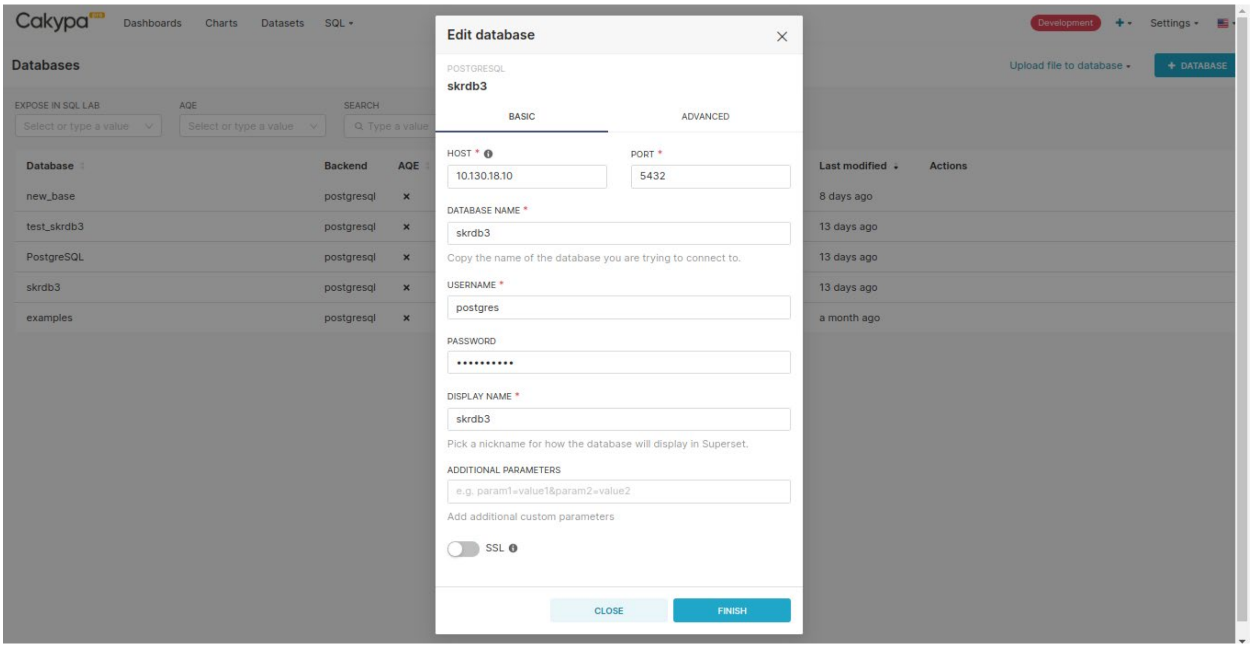


Рисунок 138 – Окно настройки параметров связи сервера бизнес-аналитики Apache Superset с сервером СУБД PostgreSQL. Вариант 1

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 138).

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 139).

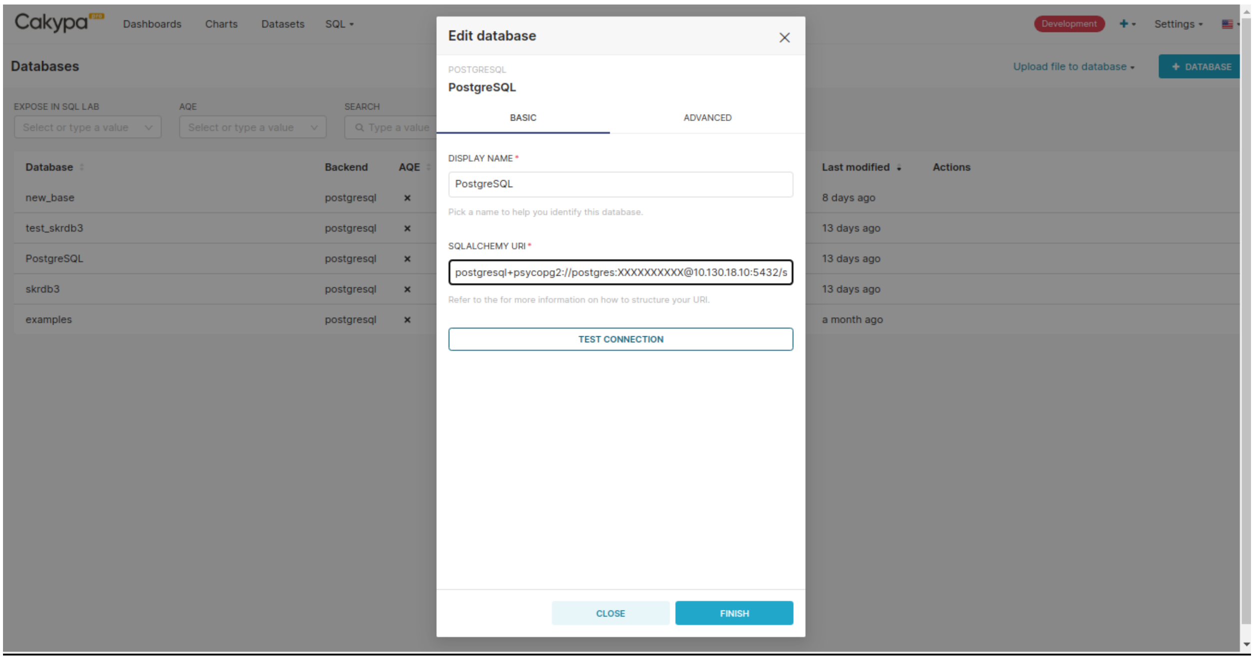


Рисунок 139 – Окно настройки параметров связи сервера бизнес-аналитики Apache Superset с сервером СУБД PostgreSQL. Вариант 2

2.5 Установка операционной системы Windows Server 2012

Установка Windows Server 2012 осуществляется на серверах контролеров домена (S-07,S-08) и на серверах электронной почты (S-09,S-10).

Установка ОС Windows Server 2012 производится согласно официальной документации разработчика данного программного обеспечения.

2.6 Развертывание доменной структуры

Развертывание домена производится в ОС Windows Server 2012 штатными средствами согласно официальной документации разработчика данного программного обеспечения.

2.7 Установка почтового сервера

Почтовый сервер разворачивается на сервере под управлением ОС Windows Server 2012 на базе программного продукта Microsoft Exchange согласно официальной документации разработчика.

2.8 Настройка рабочего места

2.8.1 Установка операционной системы Windows 7 | Windows 10 | Windows 11

Операционная система Windows 7 | Windows 10 | Windows 11 устанавливается на ПЭВМ пользователей согласно официальной документации разработчика данной ОС.

2.8.2 Установка WEB-браузеров

Веб-браузер Google Chrome или Mozilla Firefox устанавливается на ПЭВМ пользователей согласно официальной документации разработчиков данного программного обеспечения.

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

3.1 Порядок запуска системного и прикладного программного обеспечения

Для корректной работы «САКУРА PRO» 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) «САКУРА PRO», вторым должен быть запущен сервер доступа, контроля и обработки данных (микросервисов), а уже потом возможно запускать автоматизированные рабочие места пользователей «САКУРА PRO». В таблице 8 приведен список программного обеспечения и способы его запуска.

Таблица 8 – Порядок запуска системного и прикладного программного обеспечения «САКУРА PRO»

№ п/п	Наименование программного обеспечения	Описание процесса запуска программного обеспечения
1	**2**	**3**
1	Подсистема хранения данных «САКУРА PRO»	
1.1	СУБД PostgreSQL	Запуск осуществляется выполнением команды: «service postgresql start»
2.	Сервер доступа, контроля и обработки данных (микросервисов) «САКУРА PRO»	
2.1	Сервер микросервисов	Запуск осуществляется выполнением команды: «docker-compose up --build –d»
3.	Рабочее место пользователя «САКУРА PRO»	
3.1	Рабочее место пользователя «САКУРА PRO» уровня центрального подразделения	Для запуска рабочего места пользователя «САКУРА PRO» необходимо на ПЭВМ пользователя открыть Web-браузер перейти по ссылке к рабочему месту пользователя «САКУРА PRO». 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 «САКУРА PRO».

3.2 Порядок мониторинга системного и прикладного программного обеспечения

В таблице 9 приведен перечень программного обеспечения и способы контроля его работоспособности.

Таблица 9 – Способы контроля работоспособности системного и прикладного программного обеспечения «САКУРА PRO»

№ п/п	Наименование ПО	Метод контроля	Средства контроля
1	**2**	**3**	**4**
1	Подсистема хранения данных «САКУРА PRO»		
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) «САКУРА PRO»		
2.1	Сервер микросервисов	Проверка состояния микросервисов осуществляется посредством контроля log-файлов.	log-файлs «ms_log» расположены в директории «/var/sakura-cloud/log»

3.3 Порядок остановки системного и прикладного программного обеспечения

В таблице 10 приведен список программного обеспечения и способы его остановки.

Таблица 10 – Порядок остановки системного и прикладного программного обеспечения «САКУРА PRO»

№ п/п	Наименование ПО	Автоматическая остановка Ручная остановка
1	**2**	**3**
1.	Сервер доступа, контроля и обработки данных (микросервисов) «САКУРА PRO»	
1.1.	Сервер микросервисов	Остановка осуществляется выполнением команды «docker-compose --stop»
2.	Подсистема хранения данных (ПХД) «САКУРА PRO»	
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 «САКУРА PRO»	
3.1.	Рабочее место пользователя «САКУРА PRO»	Для завершения работы на рабочем месте пользователя «САКУРА PRO» необходимо завершить все начатые операции и нажать функциональную клавишу «ВЫХОД». После этого можно закрыть

4 ОБНОВЛЕНИЕ СИСТЕМЫ

1.1. Обновление системного программного обеспечения

Обновление системного программного обеспечения (ОС, СУБД и т.д.) должно осуществляться только с использованием официальных депозитариев распространения новых версий данного системного программного обеспечения, которые принадлежат разработчикам (владельцам) данного программного обеспечения.

Обновление системного программного обеспечения в промышленных условиях должно осуществляться только после выполнения следующих условий:

- После согласования с разработчиками или организацией (подразделением), занимающейся техническим сопровождением данной Системы
- После обновления документов ТРП на Систему;
- После изучения документации на обновляемую версию системного программного обеспечения;
- После тестирования функционирования Системы на обновленном системном программном обеспечении на Стенде.

1.2. Обновление прикладного программного обеспечения

Обновление прикладного программного обеспечения (Сервисов, БД и т.д.) должно осуществляться только с использованием официальных дистрибутивов от разработчика системного программного обеспечения.

Для обновления прикладного программного обеспечения необходимо выполнить запуск инсталляционных скриптов – отдельно для сервисов и отдельно для баз данных хранилища данных. Порядок запуска скриптов описывается в инструкции, прилагаемой к конкретному пакету обновлений.

Обновление прикладного программного обеспечения в промышленных условиях должно осуществляться только после выполнения следующих условий:

- После официального получения пакета обновлений прикладного обеспечения от службы поддержки разработчика Системы
- После изучения документации на обновляемую версию прикладного программного обеспечения;
- После тестирования новой версии прикладного программного обеспечения Системы на Стенде.

5 АВАРИЙНЫЕ СИТУАЦИИ И ВОССТАНОВЛЕНИЕ СИСТЕМЫ

При ошибках в работе аппаратных средств (кроме носителей данных и программ) восстановление функции системы возлагается на ОС.

При ошибках, связанных с программным обеспечением (ОС и драйверы устройств), восстановление работоспособности возлагается на ОС.

При неверных действиях пользователей, неверных форматах или недопустимых значениях входных данных, система выдает пользователю соответствующие сообщения, после чего возвращается в рабочее состояние, предшествовавшее неверной (недопустимой) команде или некорректному вводу данных.

ПРИЛОЖЕНИЕ А. ИНСТРУКЦИЯ ПО РАЗВЕРТЫВАНИЮ ОТКАЗОУСТОЙЧИВОГО КЛАСТЕРА POSTGRESQL

Для развёртывания отказоустойчивого кластера PostgreSQL используется следующее ПО:

- операционная система Ubuntu 20.04 LTS;
- гипервизор KVM;
- СУБД PostgreSQL 12.6;
- Pacemaker 2.0.3;
- Corosync 3.0.3;
- Fence-agents 4.5.2;
- Resource-agents 4.7.0.

Допускается использование ПО версий выше указанных.

Приложение А1.1 Подготовка к установке

Для развёртывания кластера необходимо подготовить инфраструктуру, состоящую из минимум трёх узлов. Возможно использование как физических, так и виртуальных машин (как в примере ниже).

Узлы могут иметь произвольные названия. Здесь и ниже три узла называются S-03:node1, S-04:node2 и S-05:node3.

Выполните следующие действия:

1. В качестве узлов подготовьте три виртуальные машины одинаковой конфигурации и установите на них операционную систему Ubuntu 20.04 LTS.
2. Настройте синхронизацию времени между всеми узлами.
3. На всех узлах добавьте IP-адреса всех узлов в файл /etc/hosts.

Приложение А1.2 Установка ПО

На все подготовленные узлы кластера установите необходимое ПО при помощи команды:

$ sudo apt install corosync pcs pacemaker fence-agents postgresql

Поскольку пакет pacemaker содержит версию пакета resource-agents, которая не поддерживает работу с PostgreSQL версии 12, следует заменить его пакетом версии 4.7.0. Для этого:

1. Удалите resource-agents со всеми зависимостями:

$ sudo dpkg --remove --force-depends resource-agents

2. Скачайте и установите пакет resource-agents версии 4.7.0:

$ wget http://ftp.de.debian.org/debian/pool/main/r/resource- agents/resource-agents_4.7.0-1_amd64.deb

$ sudo dpkg -i resource-agents_4.7.0-1_amd64.deb

Приложение А1.3 Настройка кластера

Для первоначальной настройки кластера выполните следующие действия.

1. Измените парольпользователя hacluster, который был создан автоматически на всех узлах кластера при установке пакетов :

$ sudo passwd hacluster

2. Произведите аутентификацию на одном из узлов кластера:

$ sudo pcs host auth node1 node2 node3

3. Удалите все существующие файлы конфигурации кластера с помощью команды:

$ sudo pcs cluster destroy

4. Создайте кластер под названием pgcluster:

$ sudo pcs cluster setup --force pgcluster node1 addr=ipaddr node2 addr=ipaddr node3 addr=ipaddr

Параметры команды: paddr − IP-адрес добавляемого узла; node1, node2, node3 − сетевые имена узлов кластера.

Внимание! IP-адреса всех узлов кластера должны быть явно указаны в файле /etc/hosts на всех узлах кластера.

5. Запустите кластер и настройте автоматический запуск кластера при запуске системы:

$ sudo systemctl enable corosync.service

$ sudo systemctl enable pacemaker.service

$ sudo pcs cluster enable --all

Эти действия завершают настройку кластера. Чтобы узнать статус кластера используйте команду:

$ sudo pcs cluster status

Чтобы проверить синхронизацию узлов кластера используйте команду:

$ sudo corosync-cmapctl | grep members

Для отслеживания состояния кластера в реальном времени можно использовать команду:

$ sudo crm_mon -Afr

Приложение A1.4 Настройка PostgreSQL

После первоначальной настройки кластера следует настроить СУБД PostgreSQL следующим образом.

1. Отключите запуск postgresql.service при загрузке системы на каждом узле кластера. Включать и отключать сервис при необходимости теперь будет Pacemaker.

$ sudo systemctl disable postgresql.service

2. На узле (в нашем случае это node1), который первоначально будет являться мастером, инициализируйте новую базу данных:

$ sudo -u postgres /usr/lib/postgresql/12/bin/initdb –D /var/lib/postgresql/12/main

Если база уже запущена, то остановите процесс postgresql, очистите директорию /var/lib/postgresql/12/main и выполните команду ещё раз.

3. Запустите базу:

$ sudo -u postgres /usr/lib/postgresql/12/bin/pg_ctl –D /var/lib/postgresql/12/main start

4. Создайте пользователя для репликации:

$ sudo -u postgres /usr/lib/postgresql/12/bin/createuser -- replication -P repl

5. Измените файл /var/lib/postgresql/12/main/pg_hba.conf и добавьте в него необходимые разрешения следующим образом:

```
# 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
```

Внимание! В двух нижних строках в столбце ADDRESS следует указать вашу подсеть.

6. Измените файл /var/lib/postgresql/12/main/postgresql.conf и добавьте в него следующие строки:

```
listen_addresses = '*'
wal_level = replica
logging_collector = on
hot_standby = on
lc_messages = 'C'
wal_keep_segments = 10
```

7. Перезапустите СУБД PostgreSQL:

$ sudo -u postgres /usr/lib/postgresql/12/bin/pg_ctl –D /var/lib/postgresql/12/main stop

$ sudo -u postgres /usr/lib/postgresql/12/bin/pg_ctl –D /var/lib/postgresql/12/main start

На других двух узлах (node2 и node3) выполните следующие действия:

1. Остановите процесс postgresql, если он запущен.
2. Очистите директорию /var/lib/postgresql/12/main.
3. Скопируйте базу данных с мастера (node1) при помощи команды:

$ sudo -u postgres pg_basebackup -D /var/lib/postgresql/12/main -h 192.168.110.200 -X stream

Параметр 192.168.110.200 в данном случае − это IP-адрес узла node1, на котором мы произвели первоначальную настройку PostgreSQL и инициализировали базу данных.

Приложение А1.5 Настройка ресурсов кластера

А1.5.1 Создание ресурсов кластера

После настройки СУБД PostgreSQL на всех узлах кластера следует создать ресурсы кластера. Для этого выполните следующие действия:

1. Создайте ресурс с именем virtual_ip типа IPaddr2, который будет использоваться для подключение в базе данных PostgreSQL:

$ sudo pcs resource create virtual_ip IPaddr2 ip="192.168.110.204" cidr_netmask="24" meta migration-threshold="0" op monitor timeout="60s" interval="10s" on-fail="restart" op stop timeout="60s" interval="0s" on-fail="ignore" op start timeout="60s" interval="0s" on-fail="stop"

В качестве значения параметра ip укажите любой адрес из подсети, в которой находятся узлы кластера. Этот IP-адрес будет автоматически добавляться на интерфейс того узла, который будет являться мастером в тот момент времени.

2. Создайте ресурс с именем my-pgsql типа pgsql для управления конфигурацией PostgreSQL:

$ sudo pcs resource create my-pgsql pgsql pgctl="/usr/lib/postgresql/12/bin/pg_ctl" psql="/usr/lib/postgresql/12/bin/psql" pgdata="/var/lib/postgresql/12/main" rep_mode="sync" node_list="node1 node2 node3" master_ip="192.168.110.204" restart_on_promote="false" check_wal_receiver="true" pgport="5432" primary_conninfo_opt="password=12345" repuser="repl"

3. Для созданного выше ресурса my-pgsql укажите, что он может иметь одно из нескольких состояний и менять их в зависимости от типа узла (master и slave):

$ sudo pcs resource promotable my-pgsql promoted-max=1 promoted- node-max=1 clone-max=3 clone-node-max=1 notify=true

Свяжите два созданных выше ресурса, чтобы они запускались вместе на одном узле, и установите очерёдность запуска таким образом, чтобы ресурс virtual_ip запускался только после успешного запуска ресурса my-pgsql. Для этого создайте группу ресурсов master-group и добавьте в неё ресурсы:

$ sudo pcs resource group add master-group virtual_ip

$ sudo pcs constraint colocation add master-group with Master my- pgsql-clone

$ sudo pcs constraint order promote my-pgsql-clone then start master-group symmetrical=false score=INFINITY

$ sudo pcs constraint order demote my-pgsql-clone then stop master-group symmetrical=false score=0

A1.5.2 Создание ресурсов фенсинга

Для защиты разделяемых ресурсов и изоляции узла кластера при его неисправности существует механизм фенсинга (изоляции).

Чтобы вывести список доступных fence-agents используйте команду:

$ sudo pcs stonith list

При использовании виртуальных машин в качестве узлов кластера можно использовать агент fence_virsh.

Чтобы вывести необходимые настройки для выбранного агента используйте команду:

$ sudo pcs stonith describe fence_virsh

A1.5.3 Настройка доступа по ssh

Чтобы настроить доступ поssh к серверу с гипервизором под пользователем root по ключу выполните следующие действия.

1. На сервере в файле /etc/ssh/sshd_config установите значение параметра PermitRootLogin равное yes.
2. Перезагрузите службу sshd:

$ sudo systemctl restart sshd.service

3. На всех узлах сгенерируйте ключи при помощи команды:

$ ssh-keygen

4. На всех узлах отправьте публичный ключ на сервер с гипервизором:

$ ssh-copy-id root@192.168.110.1

5. На сервере в файле /etc/ssh/sshd_config закомментируйте параметр PermitRootLogin (чтобы он не применялся в конфигурации).
6. Перезагрузите службу sshd для применения настроек:

$ sudo systemctl restart sshd.service

Для проверки работы fence_virsh перед настройкой можно использовать команду:

$ sudo fence_virsh -a 192.168.110.1 -l root -n node1 -x –k /home/user/.ssh/id_rsa -o list

Параметры команды:

– a 192.168.110.1 - IP-адрес сервера, на котором запущен гипервизор KVM;
– l root − логин пользователя для подключения к серверу с гипервизором по ssh;
– n node1 − название виртуальной машины в гипервизоре;
– k /home/user/.ssh/id_rsa − путь к ключу, созданному при помощи команды ssh-keygen.

В результате выполнения команда выведет список всех виртуальных машин в гипервизоре.

Теперь следует создать и настроить ресурсы фенсинга для всех узлов кластера. Выполните следующие действия.

1. Создайте ресурс фенсинга fence_node1 для первого узла (node1) при помощи команды:

$ sudo pcs stonith create fence_node1 fence_virsh pcmk_host_list="node1" ipaddr="192.168.110.1" login="root" identity_file="/home/u/.ssh/id_rsa" pcmk_reboot_action="reboot" pcmk_monitor_timeout=60s plug=node1

Параметры команды:

pcmk_host_list – какими узлами кластера может управлять данный ресурс;

plug – название виртуальной машины в гипервизоре.

2. Аналогично создайте ресурсы fence_node2 и fence_node3 для узлов node2 и node3.

После создания ресурсов фенсинга для каждого узла, необходимо настроить их таким образом, чтобы они не запускались на тех узлах, для перезагрузки которых они созданы.

1. Для ресурса fence_node1 выполните команду:

$ sudo pcs constraint location fence_node1 avoids node1=INFINITY

2. Выполните аналогичную команду для других узлов:

$ **sudo pcs constraint location fence_node2 avoids node2=INFINITY**

$ sudo pcs constraint location fence_node3 avoids node3=INFINITY

3. Перезагрузите все виртуальные машины в кластере.

На этом базовая настройка отказоустойчивого кластера PostgreSQL окончена.

Приложение А1.6 Содержимое файлов конфигурации

A1.6.1 /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
```

A1.6.2 /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
```

A1.6.3 /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
}
```

ПРИЛОЖЕНИЕ Б. СКРИПТЫ ПО СОЗДАНИЮ И ВОССТАНОВЛЕНИЮ РЕЗЕРВНОЙ КОПИИ БД СУБД POSTGRESQL

Приложение Б1. Создание резервной копии БД СУБД POSTGRESQL

```
#!/bin/bash

# Set the database name and the backup file name
DB_NAME="database"
BACKUP_FILE="database_backup.sql"
date=date=$(date '+%Y-%m-%d')

# Set the username and password for the database
DB_USERNAME="username"
DB_PASSWORD="password"

# Dump the database to a file
pg_dump -U $DB_USERNAME -W $DB_NAME > $BACKUP_FILE_${date}
echo "Database has been done on ${date}" >> /tmp/backup_log_${date}.log
```

Приложение Б2. Восстановление БД СУБД POSTGRESQL из резервной копии

```
#!/bin/bash

# Set the database name and the backup file name
DB_NAME="database"
BACKUP_FILE="database_backup.sql"
date=date=$(date '+%Y-%m-%d')

# Set the username and password for the database
DB_USERNAME="username"
DB_PASSWORD="password"

pg_restore -U $DB_USERNAME –w -v -d $DB_NAME $BACKUP_FILE_${date}
```

ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ

НСИ	–	Нормативно-справочная информация
САКУРА PRO	–	Система автоматизированного контроля и управления ресурсами и активами
СУБД	–	Система управления базами данных