

УТВЕРЖДЕН
643.72410666.00067-07 98 01-ЛУ

ЗАЩИЩЕННАЯ СИСТЕМА УПРАВЛЕНИЯ
БАЗАМИ ДАННЫХ «ЈАТОВА»

Руководство по настройке. Часть 29.
Поддержка мониторинга СУБД
в части анализа запросов

643.72410666.00067-07 98 01-29

Листов 51

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

АННОТАЦИЯ

В документе приведены сведения, необходимые для установки и эксплуатации компонентов, предназначенных для мониторинга СУБД в части анализа запросов:

- Компонент «pg-explain». Версия компонента – 1.6.2;
- Компонент «pg-explain-db». Версия компонента – 1.6;
- Компонент «pg-monitor». Версия компонента – 1.6.5;
- Компонент «pg-monitor-collector». Версия компонента – 1.6.5;
- Компонент «pg-monitor-dispatcher». Версия компонента – 1.6.5.

Настоящее руководство предназначено для администраторов СУБД.



Все примеры в данном документе приведены для СУБД «Jatoba» версии ядра 5.x, для других версий все шаги выполняются аналогично, разница состоит в именах директорий.

Например, СУБД «Jatoba» версии 6.x по умолчанию устанавливается в директорию:

- ОС Windows – «C:\Program Files\GIS\Jatoba\6\bin»;
- ОС Linux – «/usr/jatoba-6/bin».

Примеры команд приведены для операционной системы Ubuntu 20.04. При развертывании в ОС, использующих систему управления пакетами RPM, необходимо заменить команды «apt install» на соответствующие команды (dnf/yum).



Важная информация

Для сертифицированной версии СУБД «Jatoba» поддерживается работа только на ОС, указанных в формуляре на поставку!

Степени важности примечаний, применяемые в документе:



Важная информация – указания, требующие особого внимания



Дополнительная информация – указания, позволяющие упростить работу с изделием

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

СОДЕРЖАНИЕ

1. Назначение компонентов.....	5
1.1. Условия применения.....	5
1.2. Ограничения по эксплуатации.....	6
2. Архитектура мониторинга в части анализа запросов.....	7
3. Установка и настройка целевой СУБД «Jatoba».....	8
3.1. Установка расширения auto_explain.....	8
3.1.1. Переменные расширения auto_explain.....	8
3.1.2. Настройка конфигурационного файла postgresql.conf целевой СУБД.....	11
3.2. Настройка файла pg_hba.conf целевой СУБД.....	12
3.3. Настройка SSH-сервера на целевой СУБД.....	13
3.3.1. Установка необходимых пакетов.....	13
3.3.2. Проверка статуса сервера.....	14
3.3.3. Разрешение SSH соединения через брандмауэр.....	15
3.3.4. Настройка сервера SSH.....	15
3.4. Установка программной платформы «Node.js».....	17
4. Установка и настройка pg-explain.....	18
4.1. Предварительные требования к установке.....	18
4.2. Установка pg_repack.....	19
4.3. Установка explain db.....	19
4.4. Установка pg-monitor.....	23
4.5. Установка pg-monitor-collector.....	26
4.6. Настройка SSH-доступа к узлам.....	27
4.6.1. Генерация ключей SSH.....	27
4.6.2. Загрузка ключа на сервер.....	28
4.6.3. Проверка созданного подключения.....	29
4.6.4. Копирование ключа SSH в каталог pg-monitor.....	30
4.7. Установка pg-explain.....	31
5. Настройка JDS для взаимодействия с сервисами.....	34
5.1. Настройка pg-explain на узле отдельном от узла JDS.....	34
5.1.1. Установка веб-сервера nginx на сервере служебной СУБД pg-explain.....	35
5.1.2. Создание сертификата и ключа.....	36
5.1.3. Создание конфигурации сайта.....	37
5.1.4. Конфигурирование компонента JDS на отдельном узле.....	39
5.2. Настройка pg-explain на одном узле с JDS.....	40
5.2.1. Установка компонента JDS.....	41
5.2.2. Веб-сервер nginx.....	41
5.2.3. Создание сертификата и ключа для pg-explain.....	41

5.2.4. Создание конфигурации сайта	41
5.2.5. Редактирование параметров компонента JDS	43
6. Обновление pg-explain	45
6.1. Предварительные требования	45
6.2. Процесс обновления	45
7. Ошибки	47
7.1. Ошибка FATAL: password authentication failed for user "postgres"	47
7.2. Ошибка ERROR: invalid locale name: "ru_RU.UTF-8"	47
Термины и определения	49
Перечень сокращений	50

1. НАЗНАЧЕНИЕ КОМПОНЕНТОВ

Компонент `pg-explain` — это инструмент для анализа планов запросов в СУБД. Он позволяет просматривать и анализировать планы запросов, созданные оптимизатором СУБД, и помогает разработчикам и администраторам баз данных понять, как СУБД выполняет запросы.

Компонент `pg-explain-db` — это инструмент для анализа производительности базы данных СУБД. Он предоставляет информацию о планах выполнения запросов, статистике и других показателях, которые помогают оптимизировать работу с данными.

Компонент `pg-monitor` — это библиотека для мониторинга событий в базе данных с использованием гибкой системы событий, предоставляемой пакетом `pg-promise`. Библиотека позволяет отслеживать и регистрировать события, такие как запросы, ошибки и транзакции, а также упрощает логирование событий для вашего приложения.

Компонент `pg-monitor-collector` — это часть программного обеспечения `pgwatch2`, которое собирает метрики из настроенных баз данных и сохраняет их в другой базе данных.

Компонент `pg-monitor-dispatcher` — это прослушиватель для СУБД, который слушает один канал базы данных и выполняет заданную команду при получении уведомления.

1.1. Условия применения

Компоненты могут использоваться:

- с СУБД «Jatoba» версий 5.x и выше;
- с установленным компонентом в ОС «nodejs» версии 20 и выше;
- под управлением операционных систем GNU/Linux приведенных в таблице 1.1.

Таблица 1.1 – Поддерживаемые ОС

№	Наименование ОС	Поддержка ОС	Сертификат ФСТЭК	
			№ серт.	Дата выдачи
1	Windows 10	X	—	—
2	Windows 11	X	—	—
3	Windows Server 2016	X	—	—
4	Windows Server 2019	X	—	—
5	Windows Server 2022	X	—	—
6	Astra Linux 1.7 Special Edition Смоленск (x86-64)	—	2557	30.01.2012
7	Astra Linux 1.8 (x86-64)	X	—	—
№ изменения: _____		Подпись отв. лица: _____	Дата внесения изм: _____	

№	Наименование ОС	Поддержка ОС	Сертификат ФСТЭК	
			№ серт.	Дата выдачи
8	Astra Linux 2.12 Common Edition Орел (x86-64)	X	—	—
9	Debian 10	X	—	—
10	Debian 11	X	—	—
11	Debian 12	X	—	—
12	АЛЪТ 8 СП	—	3866	10.08.2018
13	АЛЪТ 10 СП	X	3866	10.08.2018
14	АЛЪТ 9.1 Server	X	—	—
15	АЛЪТ 10 Server	X	—	—
16	Ubuntu 20.04	X	—	—
17	Ubuntu 22.04	X	—	—
18	Ubuntu 24.04	X	—	—
19	ОСНОВА2	X	—	—
20	РЕД ОС 7.3 Муром	—	4060	12.01.2019
21	РЕД ОС 8	X	—	—
22	РОСА 7.9	X	—	—
23	РОСА 12.4	X	—	—
24	RedHat Enterprise Linux 8	X	—	—
25	Oracle Linux 8.4	X	—	—

1.2. Ограничения по эксплуатации

Ограничений по совместимости с другими компонентами нет.

Не поддерживается работа компонента pg-monitor в ОС:

- ОС АЛЪТ 10;
- Astra Linux 2.12 Common Edition Орел (x86-64);
- Ubuntu 18 и старше.

2. АРХИТЕКТУРА МОНИТОРИНГА В ЧАСТИ АНАЛИЗА ЗАПРОСОВ

В архитектуре мониторинга в части анализа запросов, допустимы две основные конфигурации:

- с установкой pg-explain на одном сервере СУБД «Jatoba» с установленным компонентом JDS (см. п. 5.1);

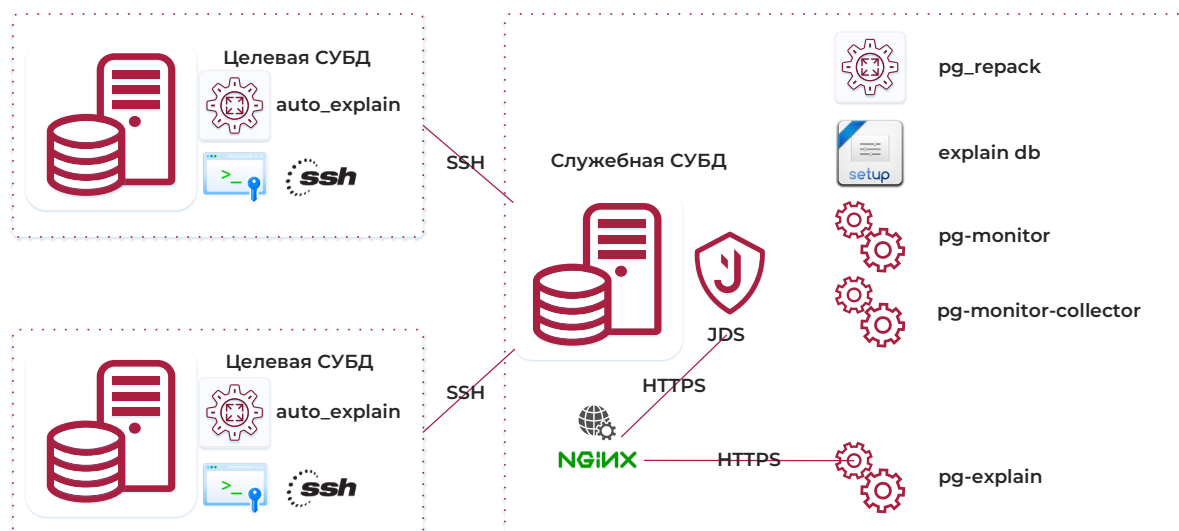


Рисунок 2.1 – Установка pg-explain и JDS на одном узле

- с установкой pg-explain выделенном сервере СУБД и отдельном сервере СУБД с установленным компонентом JDS (см. п. 5.1).

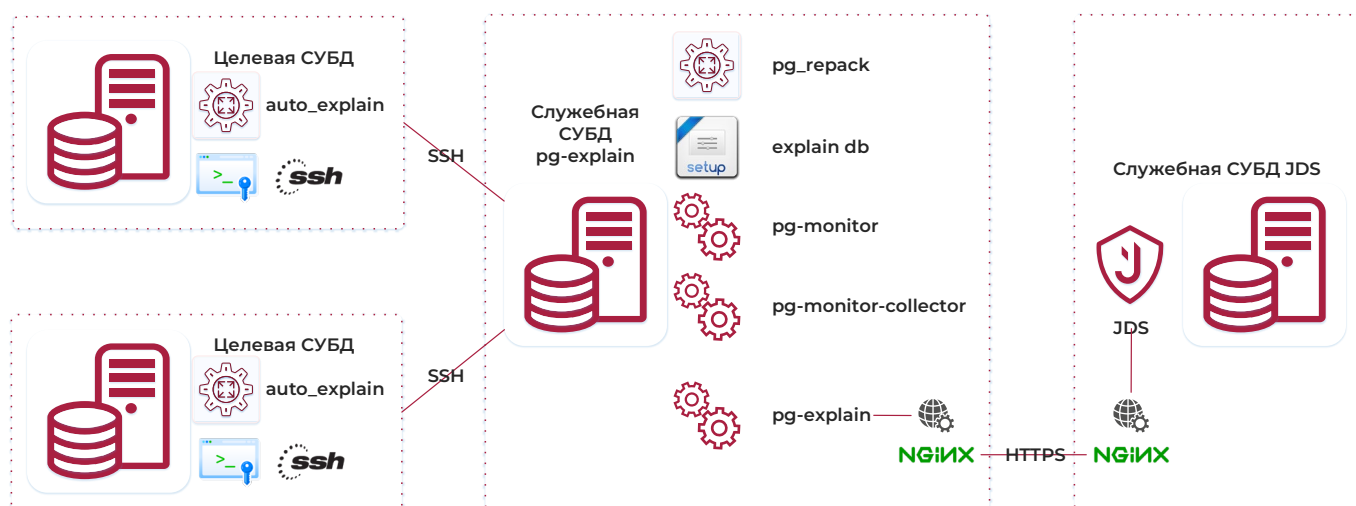


Рисунок 2.2 – Установка pg-explain и JDS на разных узлах

Данные с целевых СУБД собираются в БД pg-monitor.

3. УСТАНОВКА И НАСТРОЙКА ЦЕЛЕВОЙ СУБД «JATOBA»

СУБД «Jatoba» устанавливается в соответствии с документом «Руководство по установке».

Корректная работа pg-explain обеспечивается установкой системной локали «ru_RU.UTF-8» до развертывания СУБД «Jatoba».

Регистрация событий в СУБД осуществляется компонентами:

- «pgAudit»;
- «auto_explain».

Компонент «pgAudit» выполняет расширенную регистрацию событий безопасности. Настройка компонента приведена в документе «Защищенная система управления базами данных «Jatoba». Руководство администратора».

Компонент «auto_explain» выполняет протоколирование планов выполнения медленных запросов.

3.1. Установка расширения auto_explain

Чтобы не загружать компонент в процесс сервера, загрузка выполняется через переменную «shared_preload_libraries» в конфигурационном файле «postgresql.conf», как описано в п. 3.1.2 «Настройка конфигурационного файла postgresql.conf целевой СУБД» настоящего документа.

Установка пакета не требуется.

3.1.1. Переменные расширения auto_explain

Компонент имеет нижеперечисленные параметры, которые могут устанавливаться в конфигурационном файле «postgresql.conf» или изменяться суперпользователями СУБД «на лету» в рамках своих сеансов.

3.1.1.1 auto_explain.log_min_duration

```
auto_explain.log_min_duration (integer)
```

Переменная auto_explain.log_min_duration задаёт время выполнения оператора в миллисекундах, при превышении которого план оператора будет протоколироваться. При

значении равном 0 протоколируются все планы, а при -1 (по умолчанию) протоколирование планов отключается.

3.1.1.2 auto_explain.log_analyze

```
auto_explain.log_analyze (boolean)
```

При включении параметра `auto_explain.log_analyze` в протокол будет записываться вывод команды `EXPLAIN ANALYZE`, а не `EXPLAIN`. По умолчанию этот параметр отключен.

3.1.1.3 auto_explain.log_buffers

```
auto_explain.log_buffers (boolean)
```

Параметр `auto_explain.log_buffers` определяет, будет ли при протоколировании плана выполнения выводиться статистика об использовании буферов; он равносителен указанию `BUFFERS` команды `EXPLAIN`. Этот параметр действует, только если включён параметр [auto_explain.log_analyze](#). По умолчанию этот параметр отключен.

3.1.1.4 auto_explain.log_wal

```
auto_explain.log_wal (boolean)
```

Параметр `auto_explain.log_wal` определяет, будет ли при протоколировании плана выполнения выводиться статистика об использовании WAL; он равносителен указанию `WAL` команды `EXPLAIN`. Этот параметр действует, только если включён параметр [auto_explain.log_analyze](#). По умолчанию этот параметр отключён.

3.1.1.5 auto_explain.log_timing

```
auto_explain.log_timing (boolean)
```

Параметр `auto_explain.log_timing` определяет, будет ли при протоколировании плана выполнения выводиться длительность на уровне узлов: он равнозначен указанию `TIMING` команды `EXPLAIN`. Установка данного параметра может замедлить запросы в некоторых системах, так что возможно его следует отключать этот параметр, когда нужно знать только количество строк, но не точную длительность каждого узла. Этот параметр действует, только если включён [auto_explain.log_analyze](#). По умолчанию этот параметр отключён.

3.1.1.6 auto_explain.log_triggers

```
auto_explain.log_triggers (boolean)
```

При включении параметра `auto_explain.log_triggers` в протокол будет записываться статистика выполнения триггеров. Этот параметр действует, только если включён параметр [auto_explain.log_analyze](#). По умолчанию этот параметр отключён.

3.1.1.7 auto_explain.log_verbose

```
auto_explain.log_verbose (boolean)
```

Параметр `auto_explain.log_verbose` определяет, будут ли при протоколировании плана выполнения выводиться подробные сведения; он равнозначен указанию `VERBOSE` команды `EXPLAIN`. По умолчанию этот параметр отключён.

3.1.1.8 auto_explain.log_settings

```
auto_explain.log_settings (boolean)
```

Параметр `auto_explain.log_settings` определяет, будут ли вместе с планами выполнения выводиться изменённые параметры конфигурации. При его включении выводятся только те параметры, которые влияют на планирование запросов и имеют значения, отличающиеся от встроенных. По умолчанию этот параметр отключён. Изменить его могут только суперпользователи.

3.1.1.9 auto_explain.log_format

```
auto_explain.log_format (enum)
```

Параметр `auto_explain.log_format` выбирает формат вывода для `EXPLAIN`. Он может принимать значение `text`, `xml`, `json` и `yaml`. Значение по умолчанию — `text`. Изменить этот параметр могут только суперпользователи.

3.1.1.10 auto_explain.log_level

```
auto_explain.log_level (enum)
```

Параметр `auto_explain.log_level` выбирает уровень, с которым `auto_explain` будет выводить в протокол планы запросов. Допустимые значения: `DEBUG5`, `DEBUG4`, `DEBUG3`,

DEBUG2, DEBUG1, INFO, NOTICE, WARNING и LOG. По умолчанию подразумевается LOG. Изменить этот параметр могут только суперпользователи.

3.1.1.11 auto_explain.log_nested_statements

```
auto_explain.log_nested_statements (boolean)
```

При включении параметра `auto_explain.log_nested_statements` протоколированию могут подлежать и вложенные операторы (операторы, выполняемые внутри функции). Когда он отключён, протоколируются планы запросов только верхнего уровня. Изменить этот параметр могут только суперпользователи.

3.1.1.12 auto_explain.sample_rate

```
auto_explain.sample_rate (real)
```

Параметр `auto_explain.sample_rate` задаёт для `auto_explain` процент операторов, которые будут отслеживаться в каждом сеансе. Значение по умолчанию — 1, то есть отслеживаются все запросы. Вложенные операторы отслеживаются совместно — либо все, либо никакой из них. Изменить этот параметр могут только суперпользователи.

3.1.2. Настройка конфигурационного файла `postgresql.conf` целевой СУБД

В разделе «Shared Library Preloading», конфигурационного файла `/var/lib/jatoba/<ver>/data/postgresql.conf`, для последующей загрузки расширений `pgaudit` и `auto_explain`, установить параметры:

```
shared_preload_libraries = 'pgaudit,auto_explain'
```

Проверить и установить параметры регистрации событий в СУБД:

```
#-----  
# JATOBA LOGGING PARAMETERS  
#-----  
  
logging_collector = on  
log_directory = 'log'  
log_filename = 'jatoba-%Y-%m-%d_%H%M%S.log'  
log_rotation_age = 1d  
log_rotation_size = 0
```

```
log_truncate_on_rotation = on  
log_line_prefix = '%m [%p] app=%a host=%h user=%u db=%d '  
log_destination = 'stderr,csvlog'
```

Дополнительно установить параметры:

```
log_min_duration_statement = 10  
auto_explain.log_min_duration = 10  
auto_explain.log_nested_statements = true  
auto_explain.log_analyze = true  
auto_explain.log_buffers = true  
auto_explain.log_triggers = on  
track_io_timing = 'on'
```

Для работы компонента «auto_explain» загрузки библиотеки будет достаточно.

После применения установленных параметров, для установки компонента «pgaudit» потребуется войти в СУБД от имени и с правами пользователя «SUPERUSER», выполнить SQL-команду:

```
CREATE EXTENSION pgaudit;
```

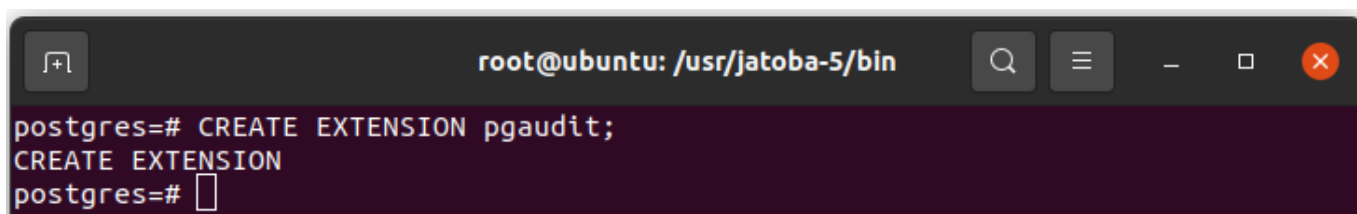


Рисунок 3.1 – Установка расширения «pgaudit»

3.2. Настройка файла pg_hba.conf целевой СУБД

На целевой СУБД должно быть разрешено подключение типа «local» роли «postgres» в режиме аутентификации «peer».

В конфигурационный файл /var/lib/jatoba/<ver>/data/pg_hba.conf добавить строку:

```
local all postgres peer
```

```

GNU nano 6.2 pg_hba.conf *
# TYPE      DATABASE      USER      ADDRESS      METHOD
# "local" is for Unix domain socket connections only
local       all             all              md5
# IPv4 local connections:
local       all             postgres        peer
host        all             all             127.0.0.1/32   md5
# IPv6 local connections:
host        all             all             ::1/128        md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local       replication     all              md5
host        replication     all             127.0.0.1/32   md5
host        replication     all             ::1/128        md5

```

Рисунок 3.2 – Строка подключения в конфигурационном файле pg_hba.conf

3.3. Настройка SSH-сервера на целевой СУБД

Подключение к целевой СУБД службой pg-monitor-collector осуществляется по протоколу SSH. Для этого на целевой СУБД должен быть установлен SSH-сервер.

3.3.1. Установка необходимых пакетов

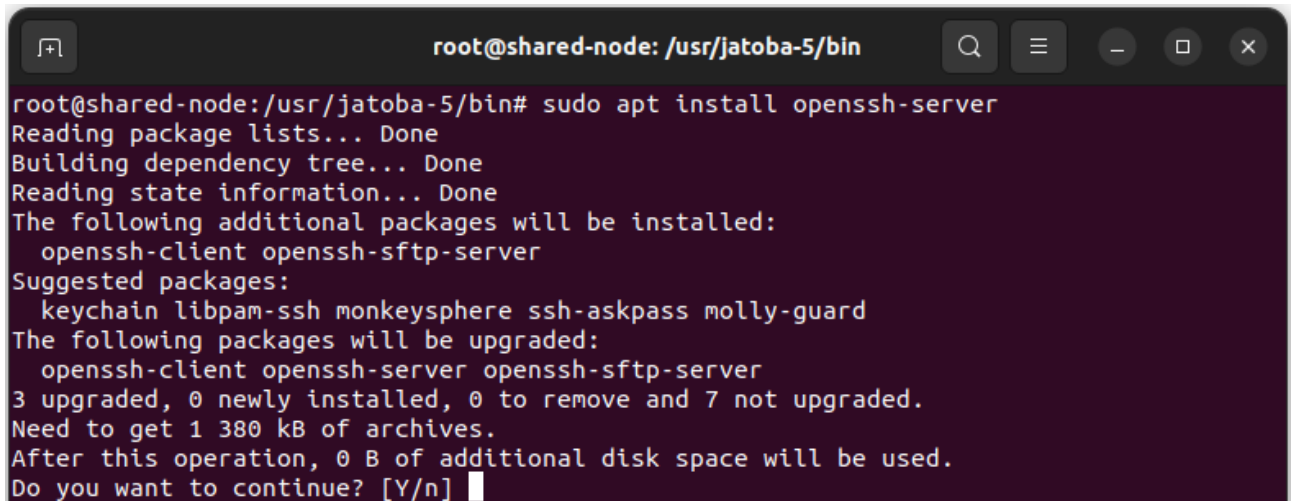
Установка необходимых пакетов выполняется от имени и с правами привилегированного пользователя в терминале ОС.

Первоначально обновляется ОС командой:

```
sudo apt update && sudo apt upgrade
```

Пакет, необходимый для запуска SSH-сервера, предоставляется компонентом openssh-server из OpenSSH и устанавливается командой:

```
sudo apt install openssh-server
```



```
root@shared-node: /usr/jatoba-5/bin# sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  openssh-client openssh-sftp-server
Suggested packages:
  keychain libpam-ssh monkeysphere ssh-askpass molly-guard
The following packages will be upgraded:
  openssh-client openssh-server openssh-sftp-server
3 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
Need to get 1 380 kB of archives.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n]
```

Рисунок 3.3 – Установка openssh-server

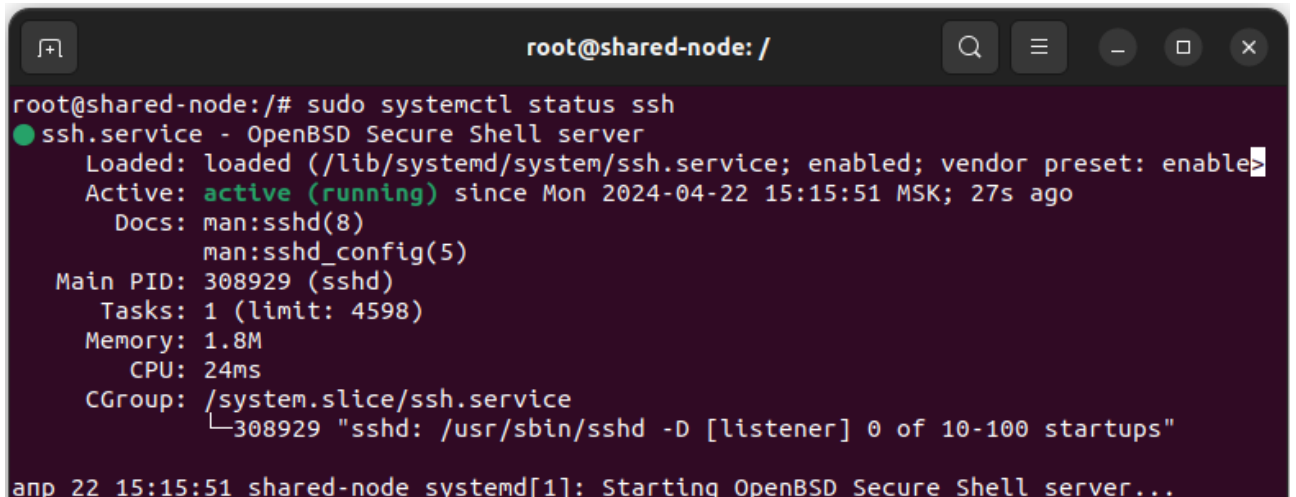
3.3.2. Проверка статуса сервера

После завершения загрузки и установки пакета служба SSH должна быть уже запущена. Статус службы проверяется командой:

```
service ssh status
```

Также можно использовать команды systemd:

```
sudo systemctl status ssh
```



```
root@shared-node: /# sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enable
   Active: active (running) since Mon 2024-04-22 15:15:51 MSK; 27s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 308929 (sshd)
     Tasks: 1 (limit: 4598)
    Memory: 1.8M
       CPU: 24ms
    CGroup: /system.slice/ssh.service
            └─308929 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

apr 22 15:15:51 shared-node systemd[1]: Starting OpenBSD Secure Shell server...
```

Рисунок 3.4 – Вывод статуса службы SSH

В выводе команды статус службы SSH должен быть в состоянии «Active».

Если служба не работает, она активируется командой:

```
sudo systemctl enable --now ssh
```

3.3.3. Разрешение SSH соединения через брандмауэр

В операционных системах Linux поставляется с утилита межсетевого экрана UFW (UncomplicatedFirewall), которая представляет собой интерфейс для утилиты командной строки iptables, который, в свою очередь, управляет сетевыми правилами.

Если брандмауэр активен, он может помешать подключению к SSH-серверу.

Чтобы настроить брандмауэр для разрешения требуемого доступа, необходимо выполнить следующую команду:

```
sudo ufw allow ssh
```

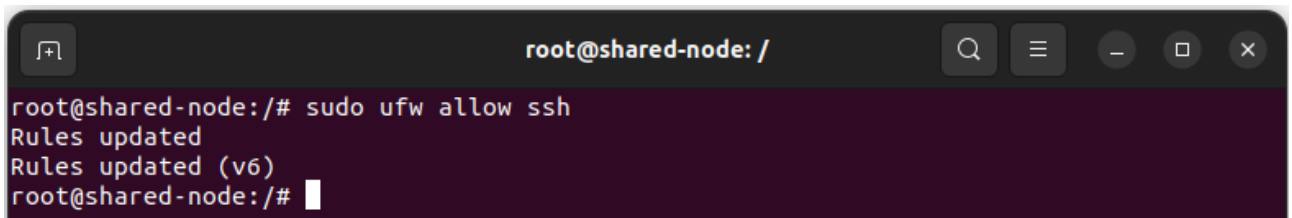


Рисунок 3.5 – Команда разрешения SSH-соединений

Статус UFW можно проверить командой:

```
sudo ufw status
```

На данном этапе SSH-сервер запущен и ожидает соединения от клиента.

3.3.4. Настройка сервера SSH

Настройки сервера SSH находятся в файле /etc/ssh/sshd_config, в котором требуется установить параметры, приведенные ниже.

3.3.4.1 Порт SSH

По умолчанию SSH работает на порту 22, но такое поведение является небезопасным, поскольку злоумышленник может попробовать выполнить «Bruteforce» атаку для перебора пароля. Порт задается строчкой:

```
Port 22
```

Необходимо изменить значение порта на требуемое.

3.3.4.2 Протокол SSH

По умолчанию сервер SSH может работать по двум версиям протокола для совместимости. Чтобы использовать только протокол версии два, необходимо раскомментировать строку и привести ее к такому виду:

```
Protocol 2
```

3.3.4.3 ROOT доступ

По умолчанию Root доступ по SSH разрешен, но такое поведение небезопасно, поэтому следует раскомментировать строку:

```
PermitRootLogin no
```

3.3.4.4 Доступ только определенного пользователя к SSH

Требуется разрешить доступ к SSH только для определенного пользователя или группы. Для этого необходимо добавить следующие строки:

```
AllowUsers User1, User2, User3  
AllowGroups Group1, Group2, Group3
```

Здесь User1 и Group1 – пользователь и группа, которым нужно разрешить доступ.

В рассматриваемом примере в ОС сервера целевых СУБД, используются пользователи admin и admin1. Им следует разрешить доступ по SSH, добавив строку:

```
AllowUsers admin, admin1
```

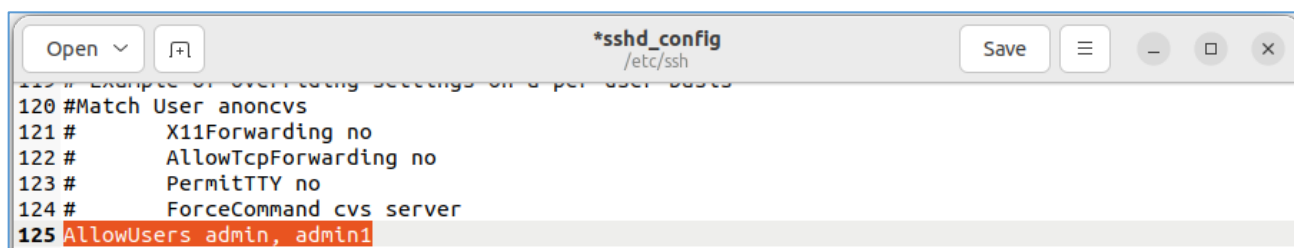
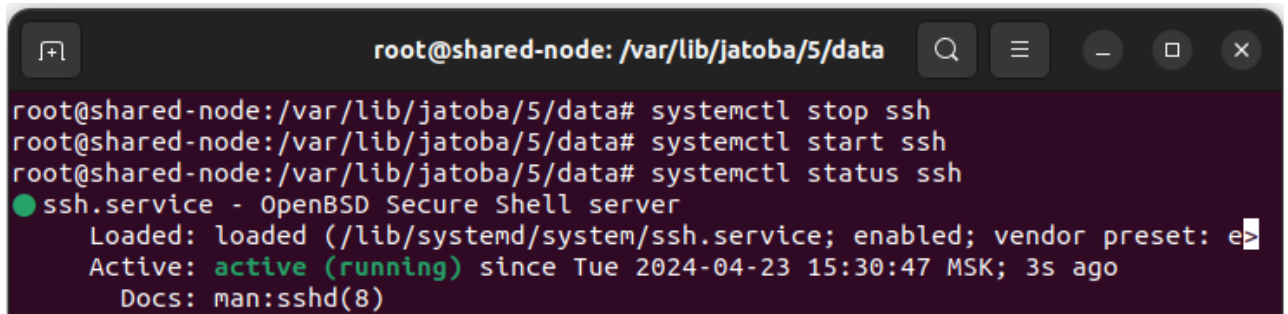


Рисунок 3.6 – Строка с именами пользователей которым разрешен доступ по SSH

Выполнив конфигурирование SSH сервера, потребуется перезагрузить службу командами:


```
# systemctl stop ssh
# systemctl start ssh
# systemctl status ssh
```



```
root@shared-node: /var/lib/jatoba/5/data
root@shared-node:/var/lib/jatoba/5/data# systemctl stop ssh
root@shared-node:/var/lib/jatoba/5/data# systemctl start ssh
root@shared-node:/var/lib/jatoba/5/data# systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: e
   Active: active (running) since Tue 2024-04-23 15:30:47 MSK; 3s ago
     Docs: man:sshd(8)
```

Рисунок 3.7 – Перезапуск службы SSH

3.4. Установка программной платформы «Node.js»

Требуется установка программной платформы «Node.js», которая позволяет использовать JavaScript версии 20 и выше.

Установка выполняется нижеописанными командами:

```
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo bash -
apt-get install -y nodejs
nodejs --version
```

4. УСТАНОВКА И НАСТРОЙКА PG-EXPLAIN

4.1. Предварительные требования к установке

На узлах системы должна быть установлена СУБД «Jatoba» в соответствии с документом «Защищенная система управления базами данных «Jatoba». Руководство по установке».

После установки СУБД обязательно устанавливается пароль для системного пользователя ОС «postgres»:

```
sudo passwd postgres
```

А также для пользователя СУБД:

```
# su -l postgres
# psql
# \password
```

Компонент пользовательского веб-интерфейса для администраторов «Jatoba data safe» (JDS) устанавливается в соответствии с документом «Защищенная система управления базами данных «Jatoba». Руководство по настройке. Часть 7. Пользовательский веб-интерфейс для администраторов. Компонент «Jatoba data safe», в зависимости от требуемой архитектуры, описанной в разделе 2 документа.

Установка пакетов, входящих в pg-explain, выполняется из архива JDS.

Для рассматриваемого примера, необходимо создать каталог /usr/share/jds командой:

```
sudo mkdir /usr/share/jds
```

С дистрибутивного диска «Disk1» из каталога «Jatoba Data Safe» скопировать файлы и каталог пакета установки в созданный каталог:

- каталог – packages, содержащий пакеты установки;
- каталог – utils, содержащий конфигурационные файлы;
- скрипт – jds.sh.

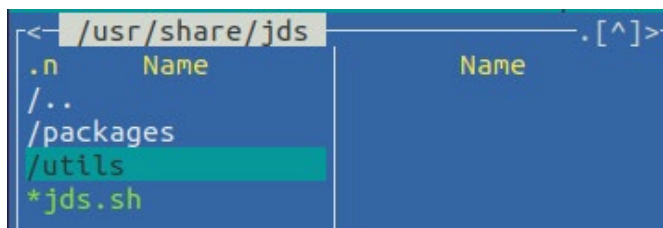


Рисунок 4.1 – Структура каталогов

В каталоге packages находятся пакеты pg-explain помимо пакета компонента JDS.

4.2. Установка pg_repack

Компонент pg_repack, требуемый для работы системы, включен в состав СУБД «Jatoba» в качестве внешней утилиты и расширения, начиная с версии 5.6.1-54937.

Установка компонента осуществляется командой:

```
apt install jatoba5-pg-repack
```

На данном шаге расширение в СУБД устанавливать не требуется, т.к. устанавливается последовательно по шагам описанным ниже в п. 4.3.

4.3. Установка explain db

Перейти в каталог с разархивированными пакетами JDS:

```
cd /usr/share/jds/packages
```

Установить пакет pg-explain-db_<version>-<buildnumber>_amd64.deb:

```
apt install ./pg-explain-db_1.5.15-20240216_amd64.deb
```

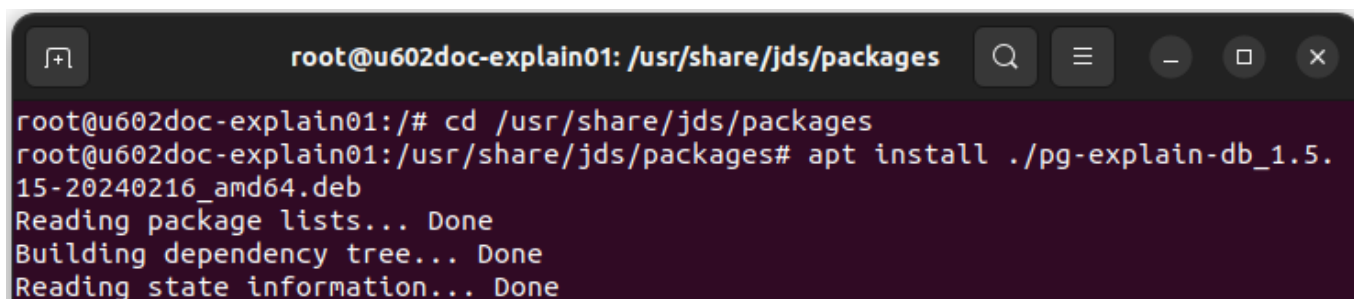


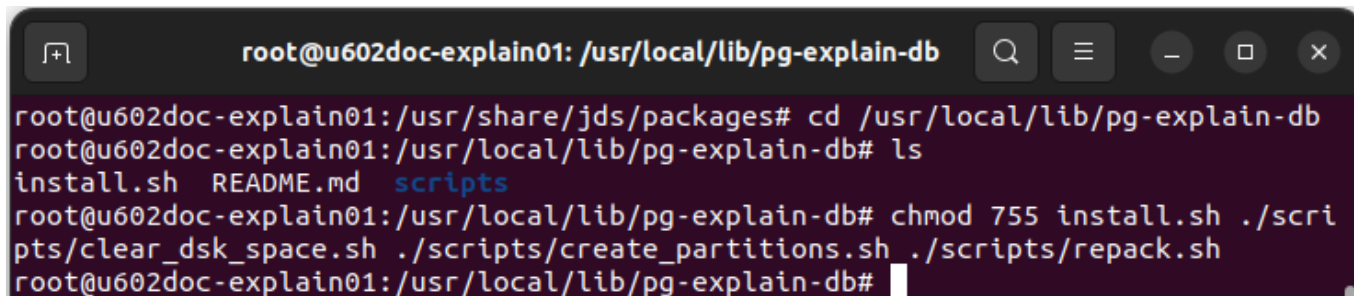
Рисунок 4.2 – Установка пакета pg-explain-db

Перейти в каталог /usr/local/lib/pg-explain-db:

```
cd /usr/local/lib/pg-explain-db
```

Установить права на выполнение:

```
chmod 755 install.sh ./scripts/clear_dsk_space.sh  
./scripts/create_partitions.sh ./scripts/repack.sh
```



```
root@u602doc-explain01: /usr/local/lib/pg-explain-db  
root@u602doc-explain01:/usr/share/jds/packages# cd /usr/local/lib/pg-explain-db  
root@u602doc-explain01:/usr/local/lib/pg-explain-db# ls  
install.sh  README.md  scripts  
root@u602doc-explain01:/usr/local/lib/pg-explain-db# chmod 755 install.sh ./scri  
pts/clear_dsk_space.sh ./scripts/create_partitions.sh ./scripts/repack.sh  
root@u602doc-explain01:/usr/local/lib/pg-explain-db#
```

Рисунок 4.3 – Команда установки прав на выполнение

С помощью текстового редактора открыть файл ./scripts/repack.sh:

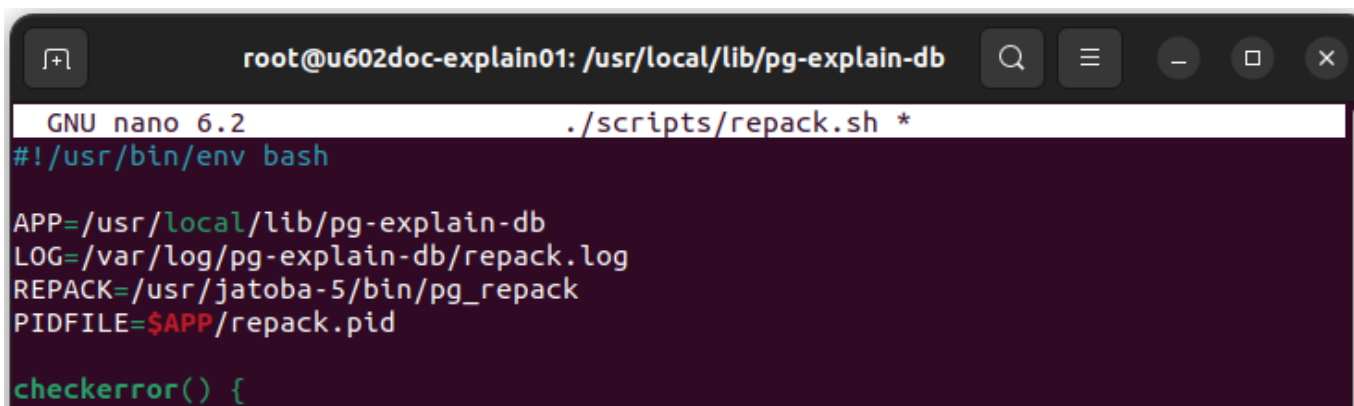
```
nano ./scripts/repack.sh
```

Исправить строку с указанием установленной версии СУБД «Jatoba»:

```
REPACK=/usr/pgsql-14/bin/pg_repack
```

на строку со следующим содержанием:

```
REPACK=/usr/jatoba-5/bin/pg_repack
```



```
GNU nano 6.2 ./scripts/repack.sh *  
#!/usr/bin/env bash  
  
APP=/usr/local/lib/pg-explain-db  
LOG=/var/log/pg-explain-db/repack.log  
REPACK=/usr/jatoba-5/bin/pg_repack  
PIDFILE=$APP/repack.pid  
  
checkerror() {
```

Рисунок 4.4 – Содержание файла ./scripts/repack.sh

Запустить основной скрипт установки:

```
./install.sh
```

В скрипте потребуется выполнить следующие шаги:

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

— Указать хост БД:

```
Enter database hostname (localhost):
```

— Указать порт БД:

```
Enter database port (5432):
```

— Указать имя создаваемой БД:

```
Enter database name (pg-monitor):
```

— Ввести имя пользователя, от имени которого будет создана БД:

```
Enter username to create database (postgres):
```

— Ввести пароль пользователя, от имени которого будет создана БД:

```
Enter password for user "postgres":
```

— Ввести имя пользователя БД:

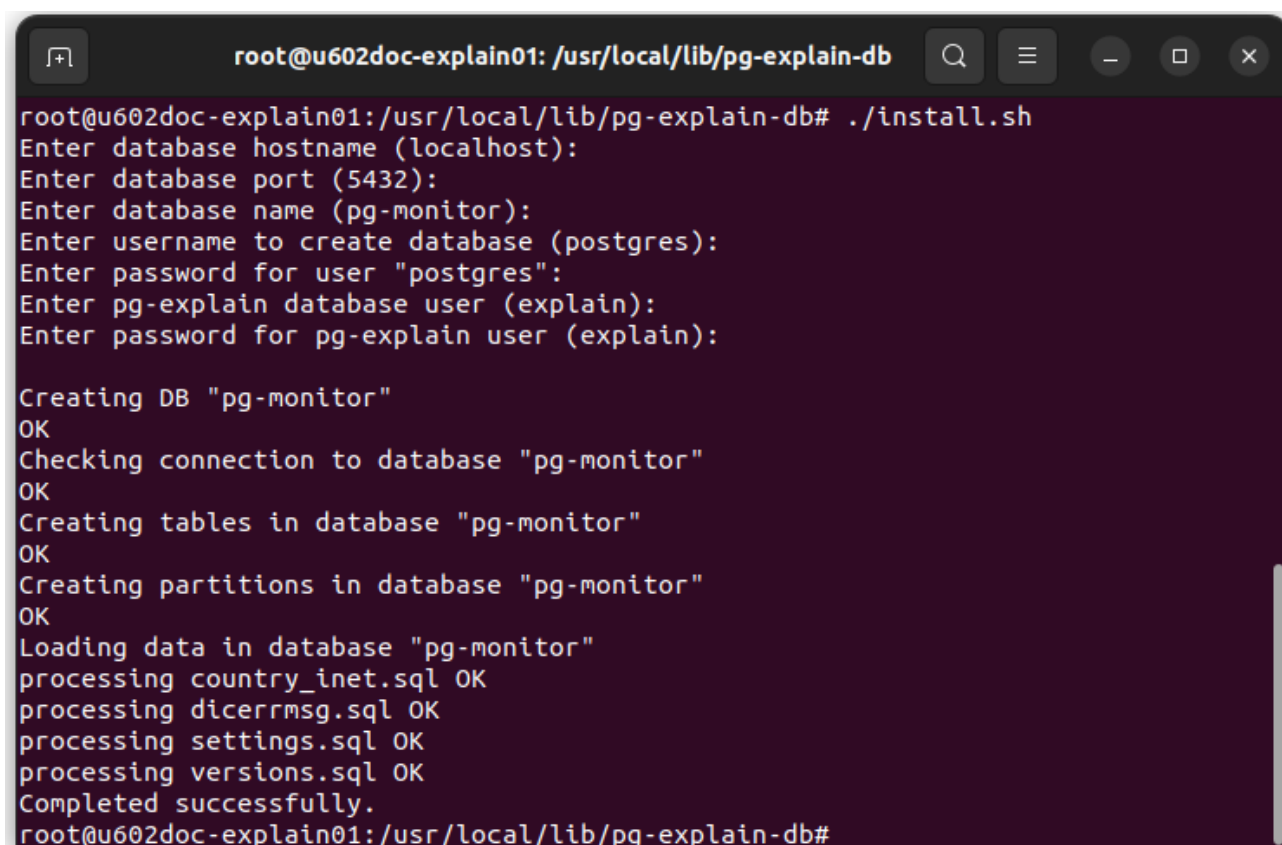
```
Enter pg-explain database user (explain):
```

— Ввести пароль для создаваемого пользователя:

```
Enter password for pg-explain user (explain):
```



При активированной парольной политике компонентом «securityprofile» устанавливаемый пароль должен соответствовать требованиям безопасности



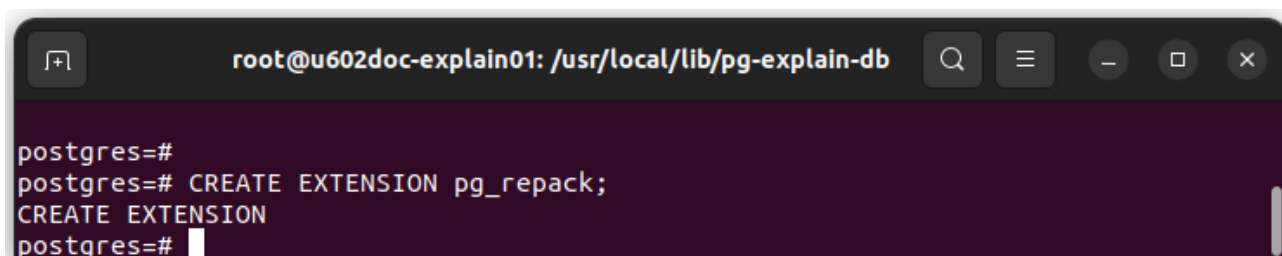
```
root@u602doc-explain01: /usr/local/lib/pg-explain-db
root@u602doc-explain01:/usr/local/lib/pg-explain-db# ./install.sh
Enter database hostname (localhost):
Enter database port (5432):
Enter database name (pg-monitor):
Enter username to create database (postgres):
Enter password for user "postgres":
Enter pg-explain database user (explain):
Enter password for pg-explain user (explain):

Creating DB "pg-monitor"
OK
Checking connection to database "pg-monitor"
OK
Creating tables in database "pg-monitor"
OK
Creating partitions in database "pg-monitor"
OK
Loading data in database "pg-monitor"
processing country_inet.sql OK
processing dicerrmsg.sql OK
processing settings.sql OK
processing versions.sql OK
Completed successfully.
root@u602doc-explain01:/usr/local/lib/pg-explain-db#
```

Рисунок 4.5 – Выполнение скрипта установки

Подключиться к созданной в процессе установки базе и создать в ней расширение:

```
CREATE EXTENSION pg_repack;
```



```
postgres=#
postgres=# CREATE EXTENSION pg_repack;
CREATE EXTENSION
postgres=#
```

Рисунок 4.6 – SQL-команда создания расширения pg_repack

Если установка пройдет с ошибками, удалить БД pg-monitor (название по умолчанию), исправить ошибки и запустить install.sh снова.

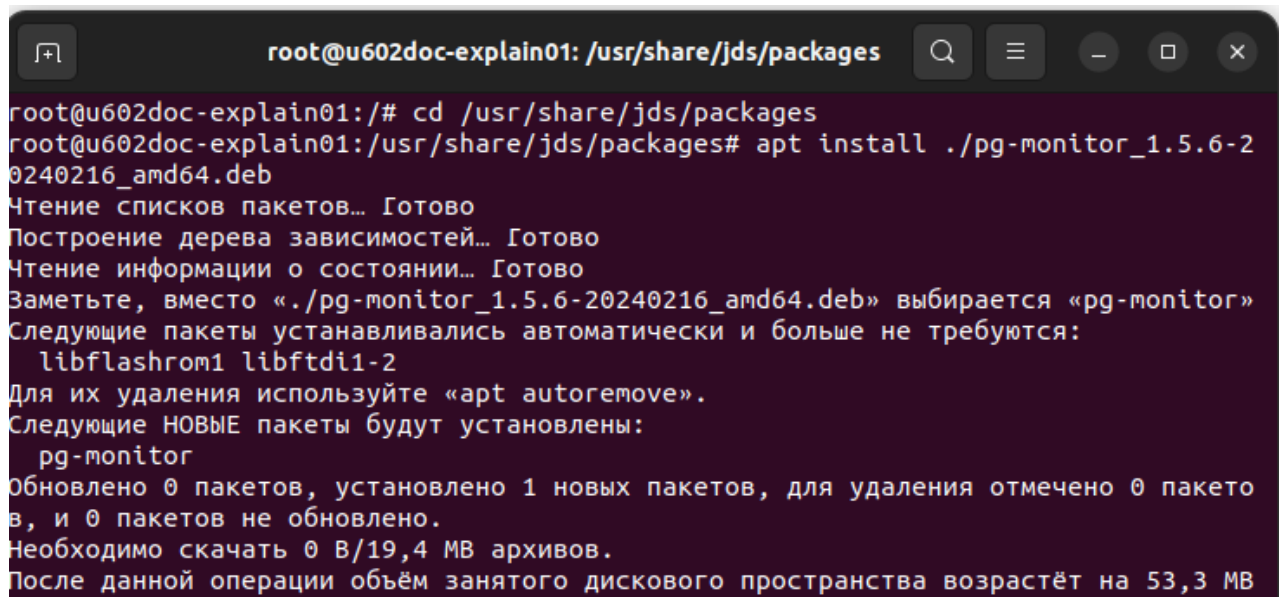
4.4. Установка pg-monitor

Пакет pg-monitor располагается в каталоге с разархивированными пакетами JDS:

```
cd /usr/share/jds/packages
```

Установка пакета pg-monitor_<version>-<buildnumber>_amd64.deb выполняется командой:

```
apt install ./pg-monitor_1.5.6-20240216_amd64.deb
```



```
root@u602doc-explain01: /usr/share/jds/packages
root@u602doc-explain01:/# cd /usr/share/jds/packages
root@u602doc-explain01:/usr/share/jds/packages# apt install ./pg-monitor_1.5.6-20240216_amd64.deb
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Заметьте, вместо «./pg-monitor_1.5.6-20240216_amd64.deb» выбирается «pg-monitor»
Следующие пакеты устанавливались автоматически и больше не требуются:
  libflashrom1 libftdi1-2
Для их удаления используйте «apt autoremove».
Следующие НОВЫЕ пакеты будут установлены:
  pg-monitor
Обновлено 0 пакетов, установлено 1 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не обновлено.
Необходимо скачать 0 B/19,4 MB архивов.
После данной операции объем занятого дискового пространства возрастёт на 53,3 MB
```

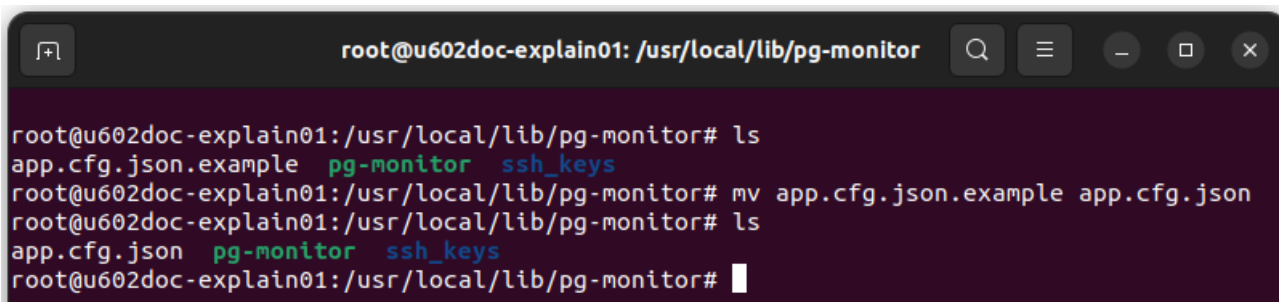
Рисунок 4.7 – Установка пакета pg-monitor

Перейти в каталог /usr/local/lib/pg-monitor:

```
cd /usr/local/lib/pg-monitor
```

Переименовать файл app.cfg.json.example в файл app.cfg.json с помощью команды:

```
mv app.cfg.json.example app.cfg.json
```



```
root@u602doc-explain01: /usr/local/lib/pg-monitor
root@u602doc-explain01:/usr/local/lib/pg-monitor# ls
app.cfg.json.example  pg-monitor  ssh_keys
root@u602doc-explain01:/usr/local/lib/pg-monitor# mv app.cfg.json.example app.cfg.json
root@u602doc-explain01:/usr/local/lib/pg-monitor# ls
app.cfg.json  pg-monitor  ssh_keys
root@u602doc-explain01:/usr/local/lib/pg-monitor#
```

Рисунок 4.8 – Переименование файла

Создать каталог для журнала аудита компонента:

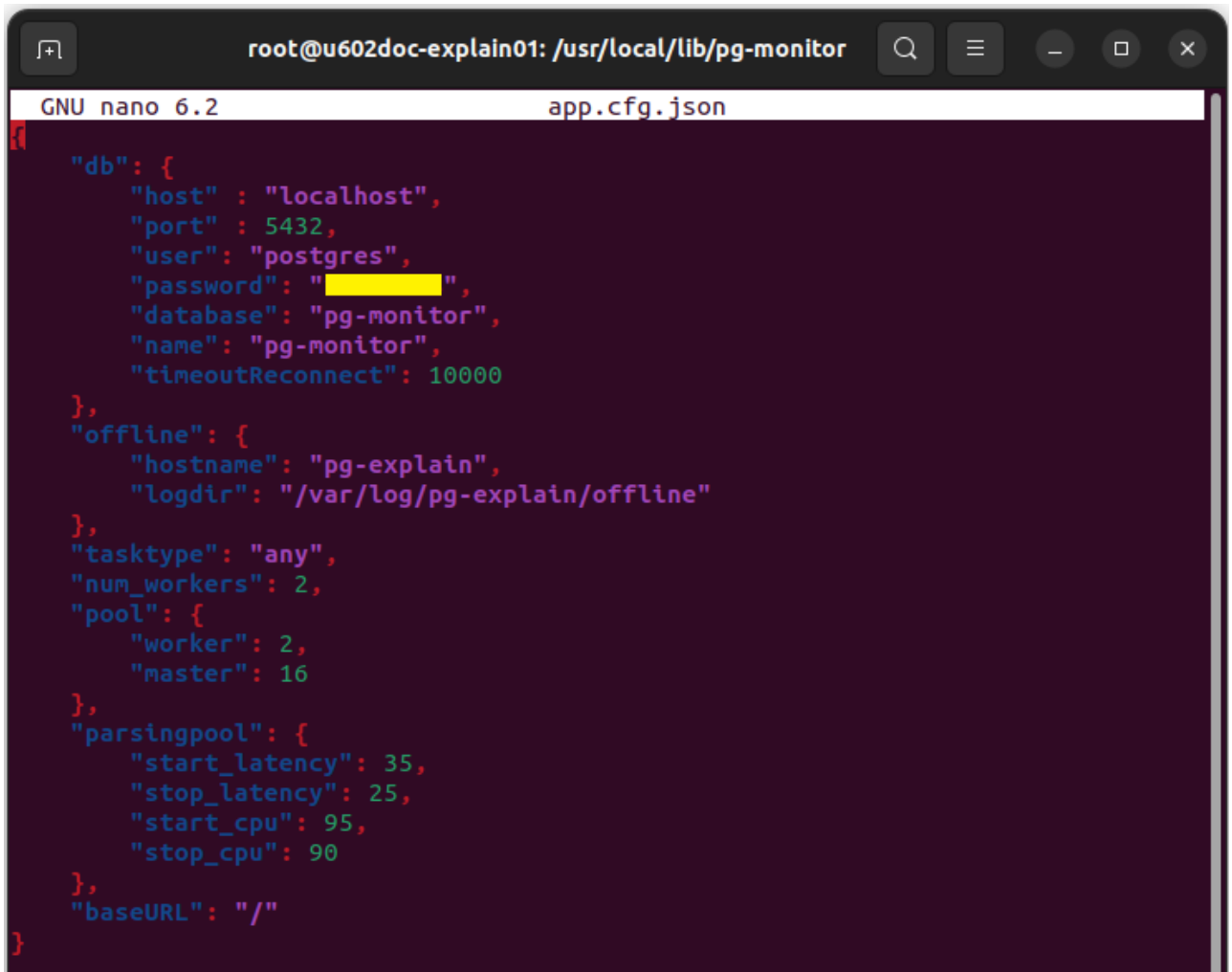
```
mkdir -p /var/log/pg-explain/offline
```

Отредактировать файл app.cfg.json командой:

```
nano app.cfg.json
```

Установить требуемые параметры, такие как пароль, база данных, num_workers по числу процессорных ядер и т. д.:

```
{
  "db": {
    "host" : "localhost",
    "port" : 5432,
    "user": "postgres",
    "password": "password",
    "database": "pg-monitor",
    "name": "pg-monitor",
    "timeoutReconnect": 10000
  },
  "offline": {
    "hostname": "pg-explain",
    "logdir": "/var/log/pg-explain/offline"
  },
  "tasktype": "any",
  "num_workers": 8,
  . . .
}
```


A screenshot of a terminal window. The title bar shows the user is root@u602doc-explain01 and the current directory is /usr/local/lib/pg-monitor. The terminal is running GNU nano 6.2, editing the file app.cfg.json. The content of the file is a JSON configuration for pg-monitor, including database connection details (localhost, port 5432, user postgres, password masked), offline logging settings (pg-explain, /var/log/pg-explain/offline), task type (any), number of workers (2), pool settings (2 workers, 16 masters), parsing pool settings (latency 35, stop latency 25, start CPU 95, stop CPU 90), and a base URL of /.

```
root@u602doc-explain01: /usr/local/lib/pg-monitor
GNU nano 6.2 app.cfg.json
{
  "db": {
    "host" : "localhost",
    "port" : 5432,
    "user": "postgres",
    "password": " ",
    "database": "pg-monitor",
    "name": "pg-monitor",
    "timeoutReconnect": 10000
  },
  "offline": {
    "hostname": "pg-explain",
    "logdir": "/var/log/pg-explain/offline"
  },
  "tasktype": "any",
  "num_workers": 2,
  "pool": {
    "worker": 2,
    "master": 16
  },
  "parsingpool": {
    "start_latency": 35,
    "stop_latency": 25,
    "start_cpu": 95,
    "stop_cpu": 90
  },
  "baseUrl": "/"
}
```

Рисунок 4.9 – Содержание конфигурационного файла app.cfg.json

Запустить службу pg-monitor командами в терминале ОС:

```
# systemctl start pg-monitor
# systemctl enable pg-monitor
# systemctl status pg-monitor
```

```

root@u602doc-explain01: /usr/local/lib/pg-monitor
root@u602doc-explain01:/usr/local/lib/pg-monitor# systemctl start pg-monitor
root@u602doc-explain01:/usr/local/lib/pg-monitor# systemctl enable pg-monitor
root@u602doc-explain01:/usr/local/lib/pg-monitor# systemctl status pg-monitor
● pg-monitor.service - pg-monitor-dispatcher
   Loaded: loaded (/lib/systemd/system/pg-monitor.service; enabled; vendor pr>
   Active: active (running) since Fri 2024-05-17 16:11:36 MSK; 18s ago
   Main PID: 15087 (pg-monitor)
     Tasks: 10 (limit: 4551)
    Memory: 49.4M
       CPU: 568ms
    CGroup: /system.slice/pg-monitor.service
            └─15087 /usr/local/lib/pg-monitor/pg-monitor --role=dispatcher --m>

```

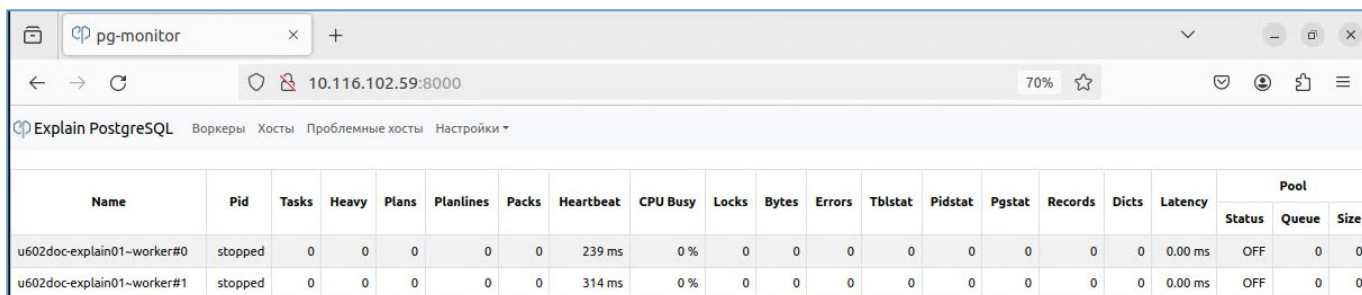
Рисунок 4.10 – Запуск и статус службы «pg-monitor»

В веб-браузере проверить состояние службы:

`http://<ip-адрес-сервиса-pg-monitor>:8000`

В рассматриваемом примере адрес будет следующим:

`http://10.116.102.59:8000`



The screenshot shows a web browser window with the address bar displaying '10.116.102.59:8000'. The page title is 'pg-monitor' and the content area shows a table of worker processes. The table has columns for Name, Pid, Tasks, Heavy, Plans, Planlines, Packs, Heartbeat, CPU Busy, Locks, Bytes, Errors, Tblstat, Pidstat, Pgstat, Records, Dicts, Latency, and a Pool section with Status, Queue, and Size. Two workers are listed, both in a 'stopped' state.

Name	Pid	Tasks	Heavy	Plans	Planlines	Packs	Heartbeat	CPU Busy	Locks	Bytes	Errors	Tblstat	Pidstat	Pgstat	Records	Dicts	Latency	Pool		
																		Status	Queue	Size
u602doc-explain01-worker#0	stopped	0	0	0	0	0	239 ms	0 %	0	0	0	0	0	0	0	0	0.00 ms	OFF	0	0
u602doc-explain01-worker#1	stopped	0	0	0	0	0	314 ms	0 %	0	0	0	0	0	0	0	0	0.00 ms	OFF	0	0

Рисунок 4.11 – Веб-интерфейс «pg-monitor»

4.5. Установка pg-monitor-collector

Служба «pg-monitor-collector» установится автоматически после установки «pg-monitor-collector».

Для полноценного функционирования достаточно добавить ее в автозагрузку ОС и проверить статус командами:

```

# systemctl enable pg-monitor-collector
# systemctl status pg-monitor-collector

```

```

root@u602doc-explain01: /usr/local/lib/pg-monitor
root@u602doc-explain01:/usr/local/lib/pg-monitor# systemctl enable pg-monitor-collector
Created symlink /etc/systemd/system/multi-user.target.wants/pg-monitor-collector.service → /lib/systemd/system/pg-monitor-collector.service.
root@u602doc-explain01:/usr/local/lib/pg-monitor# systemctl status pg-monitor-collector
● pg-monitor-collector.service - pg-monitor-collector
   Loaded: loaded (/lib/systemd/system/pg-monitor-collector.service; enabled; ▢)
   Active: active (running) since Fri 2024-05-17 16:11:36 MSK; 24min ago
     Main PID: 15085 (pg-monitor)
        Tasks: 10 (limit: 4551)
       Memory: 121.0M
          CPU: 1min 58.441s
      CGroup: /system.slice/pg-monitor-collector.service
              └─15085 /usr/local/lib/pg-monitor/pg-monitor --role=collector

мая 17 16:11:36 u602doc-explain01 systemd[1]: Starting pg-monitor-collector...
  
```

Рисунок 4.12 – Статус службы «pg-monitor-collector»

4.6. Настройка SSH-доступа к узлам

Мониторинг удаленных узлов по протоколу SSH требует предварительная настройка беспарольного доступа (по сертификату). Для чего необходимо создать ключи SSH для аутентификации на локальном сервере, при помощи утилиты `ssh-keygen`, которая входит в набор утилит OpenSSH. По умолчанию она создает пару 2048 битных RSA ключей.

4.6.1. Генерация ключей SSH

Генерация ключей SSH выполняется командой:

```
ssh-keygen
```

```

root@u602doc-explain01: /
root@u602doc-explain01:/# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
  
```

Рисунок 4.13 – Команда генерации ключей

Утилита предложит выбрать расположение ключей. По умолчанию ключи располагаются в папке `~/.ssh/`.

Секретный ключ будет называться `id_rsa`, а публичный `id_rsa.pub`.

Затем утилита предложит ввести пароль для дополнительного шифрования ключа на диске. Его можно не указывать, нажав «Enter».

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

```

root@u602doc-explain01: /
root@u602doc-explain01:/# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):

```

Рисунок 4.14 – Шаг ввода пароля для генерируемого ключа

Далее утилита сгенерирует ключи SSH.

```

root@u602doc-explain01:/# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:d7EUiUC4ivPULP1GToE/g/6gJC2To4qRs72ii8Qq0JY root@u602doc-explain01
The key's randomart image is:
+---[RSA 3072]-----+
|      ooo ...      |
|      . . o ..      |
|      + o .o        |
|      + o *. o       |
| . o * .S=.oo       |
|o.E O + o.+         |
|=+ = * . +         |
|B=. . . .          |
|@+o.               |
+----[SHA256]-----+
root@u602doc-explain01:/#

```

Рисунок 4.15 – Генерирование ключей SSH

4.6.2. Загрузка ключа на сервер

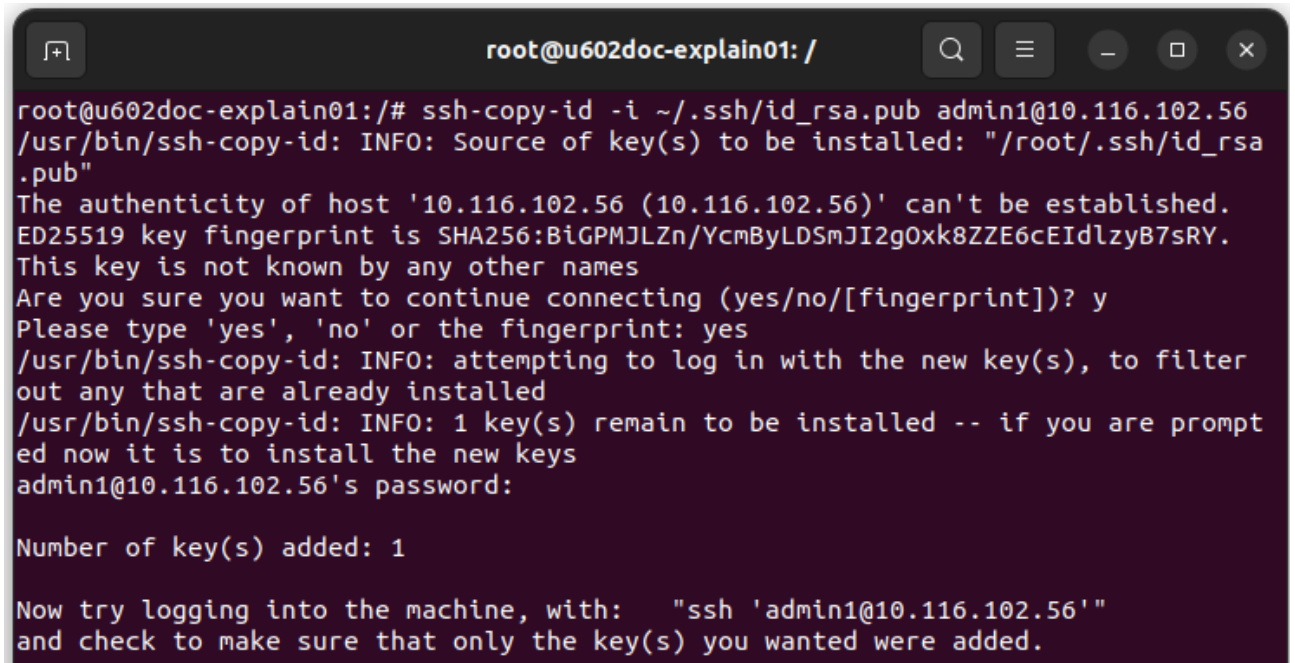
Когда генерация ключей завершена, следует загрузить ключ на сервер целевой СУБД «Jatoba». Загрузка выполняется утилитой `ssh-copy-id`. Она входит в пакет программ OpenSSH. Для загрузки ключа необходим пароль доступа к серверу по SSH.

Синтаксис команды:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub имя_пользователя@ip-сервера
```

В рассматриваемом примере ключ SSH требуется скопировать на целевую СУБД с IP-10.116.102.56 от пользователя ОС сервера СУБД (см. п. 3.3.4.4). Команда копирования будет следующей:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub admin1@10.116.102.56
```



```
root@u602doc-explain01: /
root@u602doc-explain01:/# ssh-copy-id -i ~/.ssh/id_rsa.pub admin1@10.116.102.56
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa
.pub"
The authenticity of host '10.116.102.56 (10.116.102.56)' can't be established.
ED25519 key fingerprint is SHA256:BiGPMJLZn/YcmByLDsmJI2gOxk8ZZE6cEIdlzyB7sRY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
admin1@10.116.102.56's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'admin1@10.116.102.56'"
and check to make sure that only the key(s) you wanted were added.
```

Рисунок 4.16 – Копирование ключа SSH на сервер SSH целевой СУБД

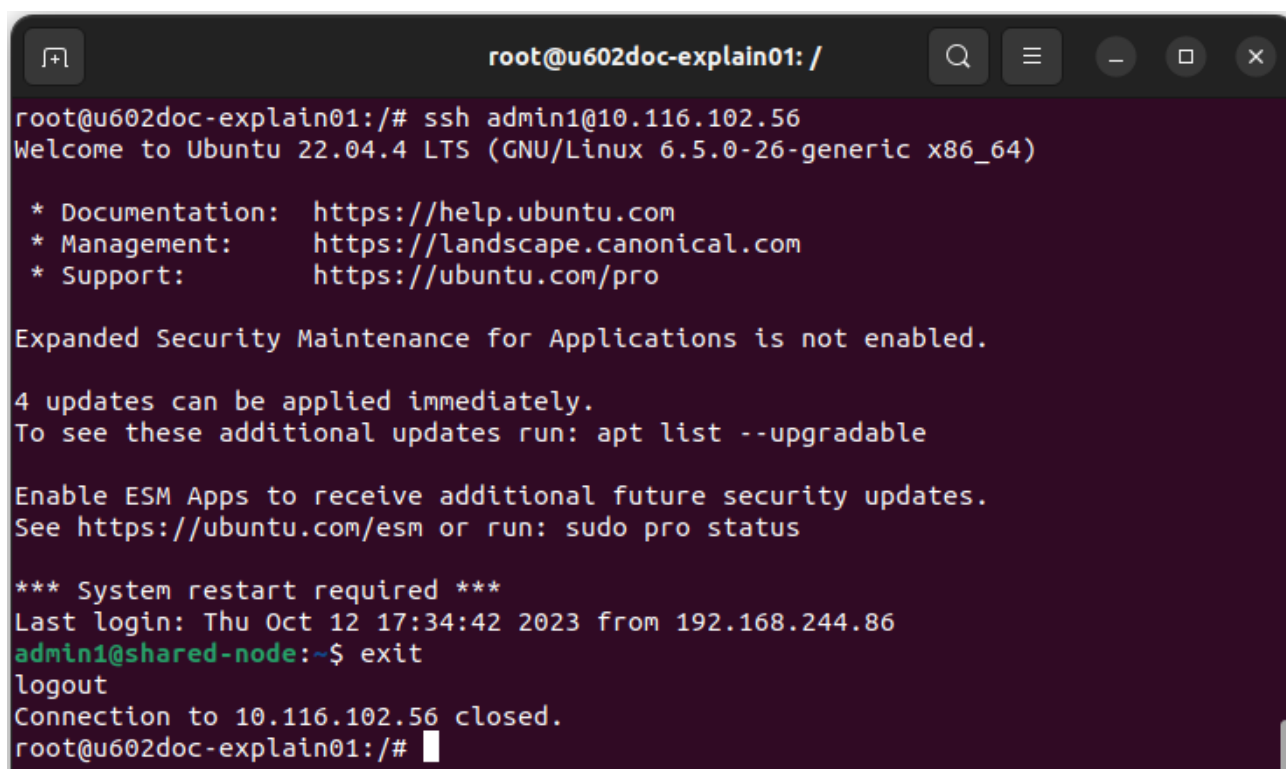
4.6.3. Проверка созданного подключения

Проверка созданного подключения выполняется командой с сервера с установленным pg_explain:

```
# ssh postgres@ip-сервера
# exit
```

В рассматриваемом примере команда подключения будет следующей:

```
ssh admin1@10.116.102.56
```



```
root@u602doc-explain01: /
root@u602doc-explain01:/# ssh admin1@10.116.102.56
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-26-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

4 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

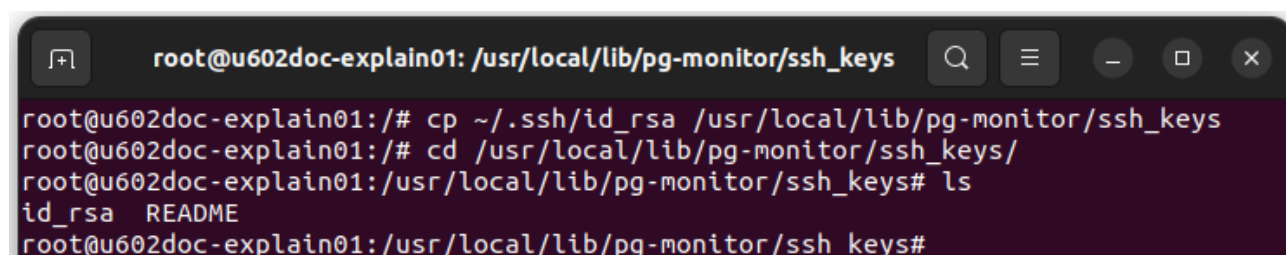
*** System restart required ***
Last login: Thu Oct 12 17:34:42 2023 from 192.168.244.86
admin1@shared-node:~$ exit
logout
Connection to 10.116.102.56 closed.
root@u602doc-explain01:/#
```

Рисунок 4.17 – Проверка подключения по SSH к целевой СУБД

4.6.4. Копирование ключа SSH в каталог pg-monitor

Скопировать закрытый ключ в каталог pg-monitor/ssh_keys. Файл должен обязательно присутствовать для запуска службы:

```
# cp ~/.ssh/id_rsa /usr/local/lib/pg-monitor/ssh_keys
```



```
root@u602doc-explain01: /usr/local/lib/pg-monitor/ssh_keys
root@u602doc-explain01:/# cp ~/.ssh/id_rsa /usr/local/lib/pg-monitor/ssh_keys
root@u602doc-explain01:/# cd /usr/local/lib/pg-monitor/ssh_keys/
root@u602doc-explain01:/usr/local/lib/pg-monitor/ssh_keys# ls
id_rsa  README
root@u602doc-explain01:/usr/local/lib/pg-monitor/ssh_keys#
```

Рисунок 4.18 – Копирование ключа SSH в каталог pg-monitor

Дать пользователю «explain» права на файл командой:

```
# chown explain:explain /usr/local/lib/pg-monitor/ssh_keys/id_rsa
```



```
root@u602doc-explain01: /usr/local/lib/pg-monitor/ssh_keys
root@u602doc-explain01:/usr/local/lib/pg-monitor/ssh_keys# chown explain:explain /usr/local/lib/pg-monitor/ssh_keys/id_rsa
root@u602doc-explain01:/usr/local/lib/pg-monitor/ssh_keys# ls -l
total 8
-rw----- 1 explain explain 2610 мая 21 14:31 id_rsa
-rw-r--r-- 1 root root 22 апр 26 10:44 README
root@u602doc-explain01:/usr/local/lib/pg-monitor/ssh_keys#
```

Рисунок 4.19 – Установка прав

Перезапустить службу коллектора:

```
systemctl restart pg-monitor-collector.service
```

4.7. Установка pg-explain

Установить пакет pg-explain_<version>-<buildnumber>_amd64.deb командами:

```
# cd /usr/share/jds/packages
# apt install ./pg-explain_1.5.9-20240216_amd64.deb
```

```
root@u602doc-explain01: /usr/share/jds/packages
root@u602doc-explain01:/usr/share/jds/packages# apt install ./pg-explain_1.5.9-20240216_amd64.deb
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Заметьте, вместо «./pg-explain_1.5.9-20240216_amd64.deb» выбирается «pg-explain»
Следующие пакеты устанавливались автоматически и больше не требуются:
  libflashrom1 libftdi1-2
Для их удаления используйте «sudo apt autoremove».
Следующие НОВЫЕ пакеты будут установлены:
  pg-explain
Обновлено 0 пакетов, установлено 1 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не обновлено.
Необходимо скачать 0 B/28,6 MB архивов.
После данной операции объем занятого дискового пространства возрастёт на 62,6 MB
.
Пол:1 /usr/share/jds/packages/pg-explain_1.5.9-20240216_amd64.deb pg-explain amd64 1.5.9-20240216 [28,6 MB]
Выбор ранее не выбранного пакета pg-explain.
(Чтение базы данных ... 85%
```

Рисунок 4.20 – Установка пакета pg-explain

Перейти в каталог /usr/local/lib/pg-explain:

```
cd /usr/local/lib/pg-explain
```

Переименовать файл app.conf.example:

```
mv app.conf.example app.conf
```

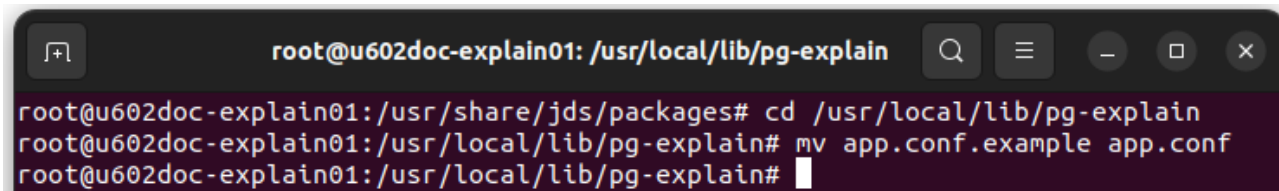


Рисунок 4.21 – Команда переименования файла

Отредактировать app.conf под требуемые настройки СУБД (БД по умолчанию pg-monitor):

```
nano app.conf
```

Параметры используются такие же, как и при установке компонента explain db, как описано в п. 4.3 настоящего документа и показано на рисунке 4.5.

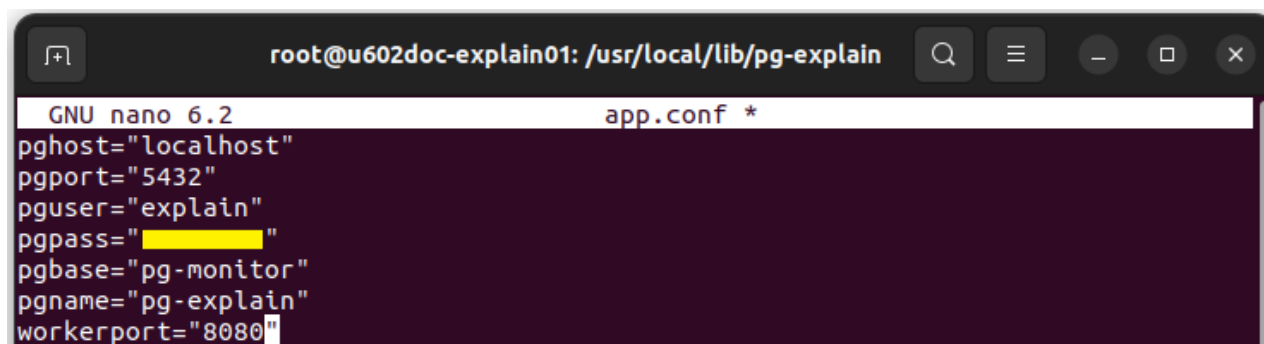


Рисунок 4.22 – Содержание файла app.conf

Запустить службу pg-explain:

```
# systemctl start pg-explain.service
# systemctl enable pg-explain.service
# systemctl status pg-explain.service
```



```
root@u602doc-explain01: /usr/local/lib/pg-explain
root@u602doc-explain01: /usr/local/lib/pg-explain# systemctl start pg-explain.service
root@u602doc-explain01: /usr/local/lib/pg-explain# systemctl enable pg-explain.service
root@u602doc-explain01: /usr/local/lib/pg-explain# systemctl status pg-explain.service
● pg-explain.service - pg-explain
   Loaded: loaded (/lib/systemd/system/pg-explain.service; enabled; vendor pre
   Active: active (running) since Tue 2024-05-21 16:28:19 MSK; 27s ago
   Main PID: 1096077 (pg-explain)
```

Рисунок 4.23 – Запуск и вывод статуса службы pg-explain.service

В веб-браузере проверить состояние службы:

`http://<ip-адрес-сервиса-pg-explain>:8080`

В рассматриваемом примере адрес сервиса pg-explain будет следующим:

`http://10.116.102.59:8080/`

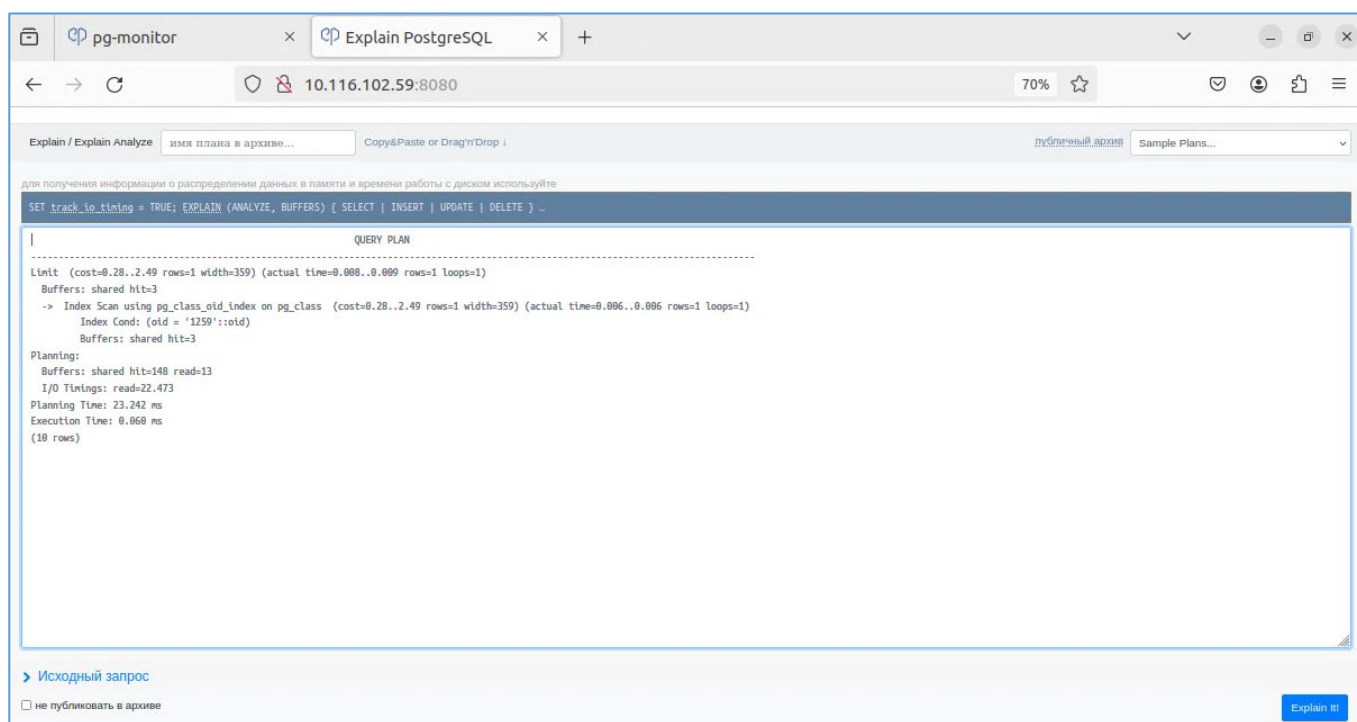


Рисунок 4.24 – Веб-страница сервиса pg-explain

При ошибке «Слишком много клиентов» в статусе службы увеличить параметр «max_connections» в файле конфигурационном файле «postgresql.conf» сервера с установленным компонентом «explain».

5. НАСТРОЙКА JDS ДЛЯ ВЗАИМОДЕЙСТВИЯ С СЕРВИСАМИ

Работа компонентов обеспечивается корректно настроенными протоколами, описанными в таблице 5.1

Таблица 5.1 – Протоколы взаимодействия компонентов

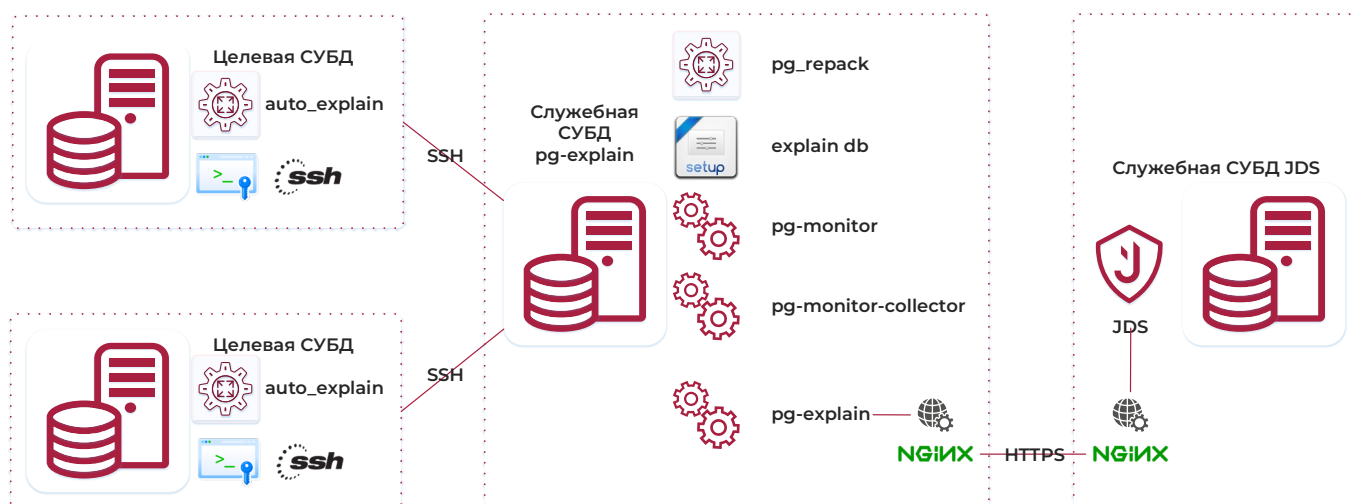
Компонент	Протокол	Компонент	Протокол	Описание
JDS	http	pg-explain	http	корректная работа
JDS	http/https	pg-explain	https	корректная работа при наличии у pg-explain валидного сертификата или, если сертификат не валидный (self-signed и т.п.), то после отдельного подтверждения pg-explain в браузере
JDS	https	pg-explain	http	данная связка работать не будет из-за ограничений безопасности браузеров

Поэтому требуется настроить реверс-прокси для перенаправления запросов к pg-explain.

5.1. Настройка pg-explain на узле отдельном от узла JDS

Компонент JDS имеет функциональную возможность работы без веб-сервера nginx. Поэтому веб-сервера nginx может быть установлен после.

Установка компонент JDS выполняется с помощью инсталлятора, а веб-сервер nginx с помощью скрипта установки, как описано в документе «Защищенная система управления базами данных «Jatoba». Руководство по настройке. Часть 7. Пользовательский веб-интерфейс для администраторов. Компонент «Jatoba data safe».



До конфигурирования компонентов должны быть выполнены действия, описанные в разделах:

- 3 «Установка и настройка целевой СУБД «Jatoba»;
- 4 «Установка и настройка pg-explain».

5.1.1. Установка веб-сервера nginx на сервере служебной СУБД pg-explain

Установка пакета nginx выполняется из репозитория ОС. Использование скрипта установки nginx.sh, расположенного в каталоге /usr/share/jds/utils, нецелесообразно, т.к. он выполнит конфигурирование веб-сервера для компонента JDS.

Установить nginx:

```
apt install -y nginx
```

```

root@u602doc-explain01: /
root@u602doc-explain01:/# apt install -y nginx
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Следующие пакеты устанавливались автоматически и больше не требуются:
  libflashrom1 libftdi1-2
Для их удаления используйте «sudo apt autoremove».
Будут установлены следующие дополнительные пакеты:
  libnginx-mod-http-geoip2 libnginx-mod-http-image-filter
  libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  libnginx-mod-stream-geoip2 nginx-common nginx-core
Предлагаемые пакеты:
  fcgiwrap nginx-doc
  
```

Рисунок 5.1 – Установка nginx

Проверить статус веб-сервера:

```
systemctl status nginx
```

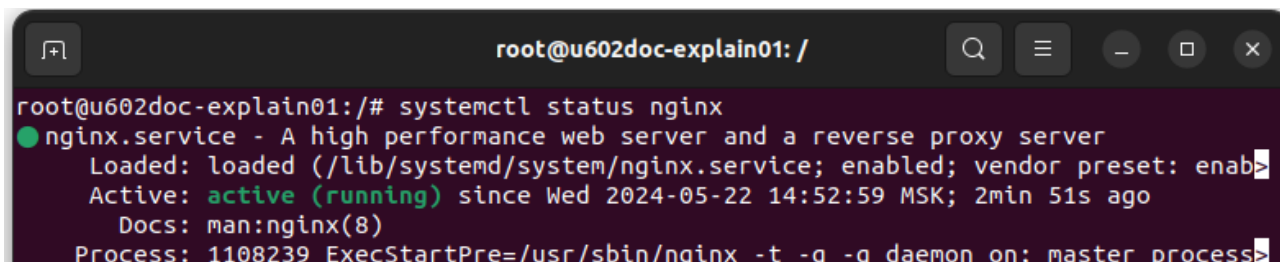


Рисунок 5.2 – Статус службы nginx

5.1.2. Создание сертификата и ключа

Создать папку для сертификата и ключа командой:

```
mkdir /etc/nginx/ssl
```

Создать сертификат и ключ: 4096

```
openssl req -x509 -nodes -days 3650 -newkey rsa:4096 -keyout
/etc/nginx/ssl/nginx-explain.key -out /etc/nginx/ssl/nginx-
explain.crt
```

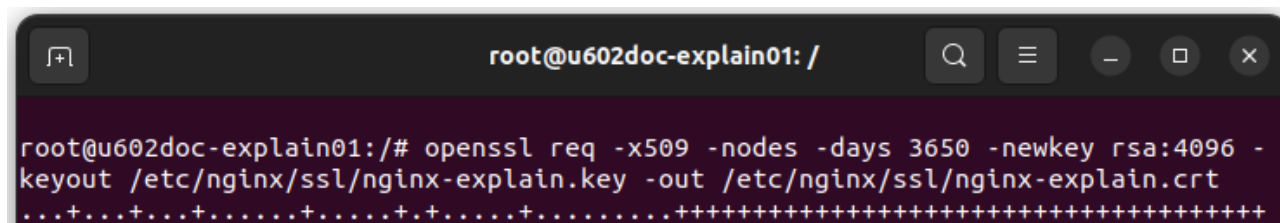


Рисунок 5.3 – Команда создания ключа и сертификата

Процесс формирования сертификата потребует ввода значений по параметрам, приведенным в таблице 5.2. В этой же таблице приведены значения, которые использовались для формирования примера.

Таблица 5.2 – Параметры формирования сертификата

Параметры	Примерные значения
Country Name (2 letter code) [AU]	RU
State or Province Name (full name) [Some-State]	Leningrad obl
Locality Name (eg, city) []	Saint Petersburg
Organization Name (eg, company) [Internet Widgits Pty Ltd]	Datagile
Organizational Unit Name (eg, section) []	Private
Common Name (e.g. server FQDN or YOUR name) []	NAME
Email Address []	test@datagile.ru

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм.: _____
--------------------	--------------------------	---------------------------

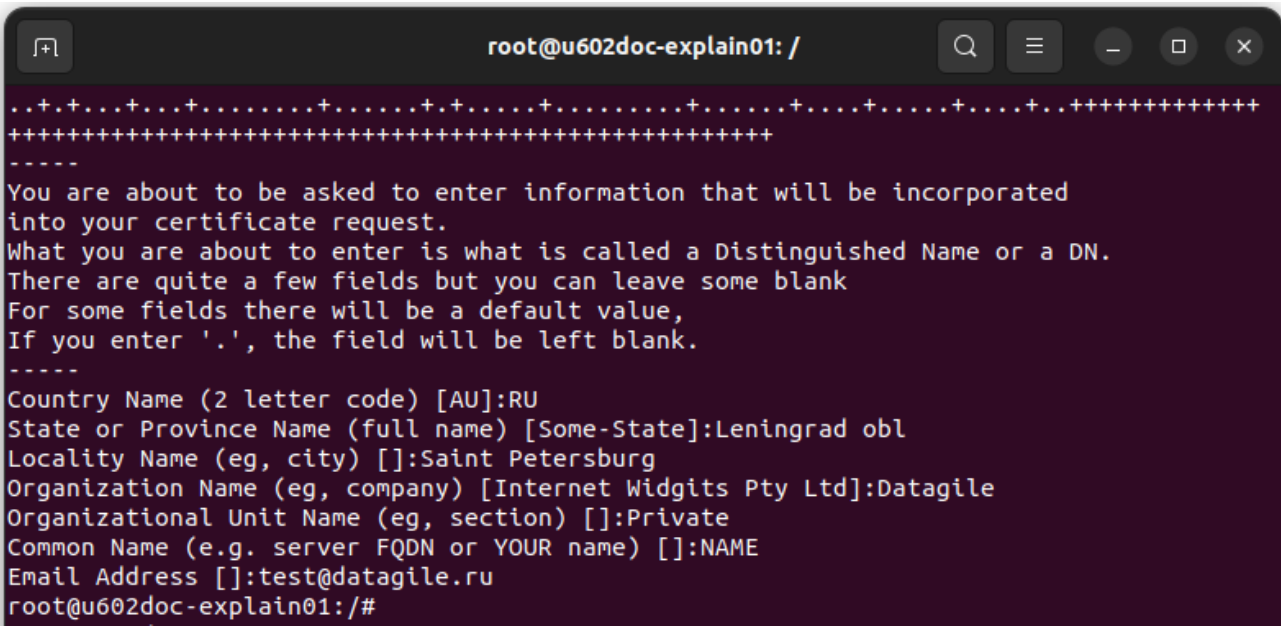


Рисунок 5.4 – Вводимые параметры для формирования сертификата

5.1.3. Создание конфигурации сайта

Создать файл конфигурации сайта командой:

```
nano /etc/nginx/conf.d/explain.https.conf
```

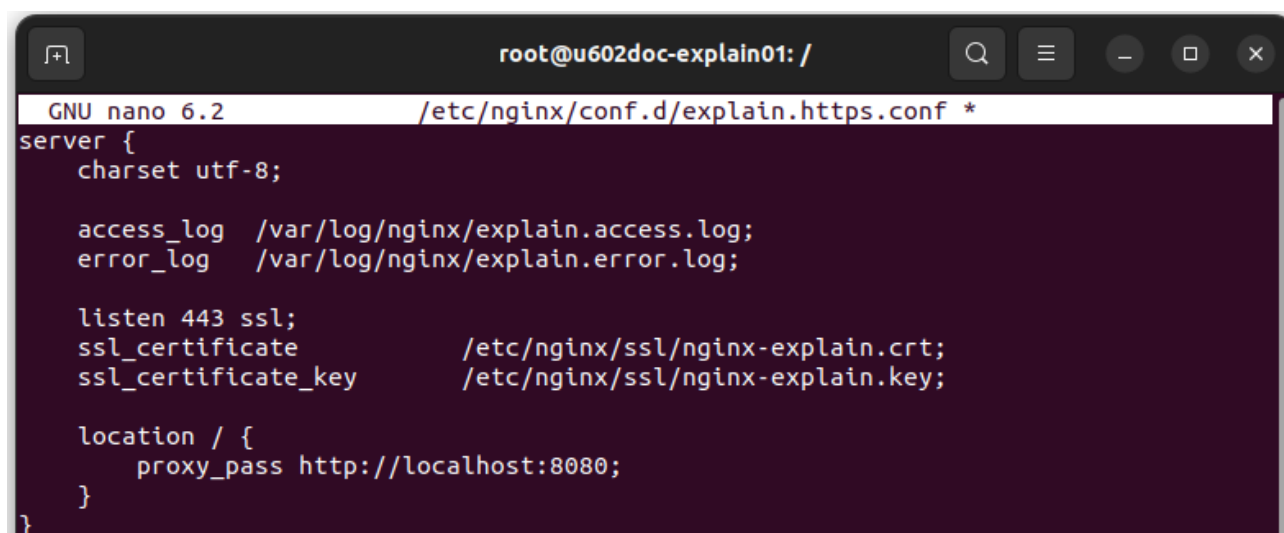
Вставить текст и сохранить:

```
server {
    charset utf-8;

    access_log    /var/log/nginx/explain.access.log;
    error_log     /var/log/nginx/explain.error.log;

    listen 443 ssl;
    ssl_certificate      /etc/nginx/ssl/nginx-explain.crt;
    ssl_certificate_key  /etc/nginx/ssl/nginx-explain.key;

    location / {
        proxy_pass http://localhost:8080;
    }
}
```



```
root@u602doc-explain01: /
GNU nano 6.2 /etc/nginx/conf.d/explain.https.conf *
server {
    charset utf-8;

    access_log /var/log/nginx/explain.access.log;
    error_log /var/log/nginx/explain.error.log;

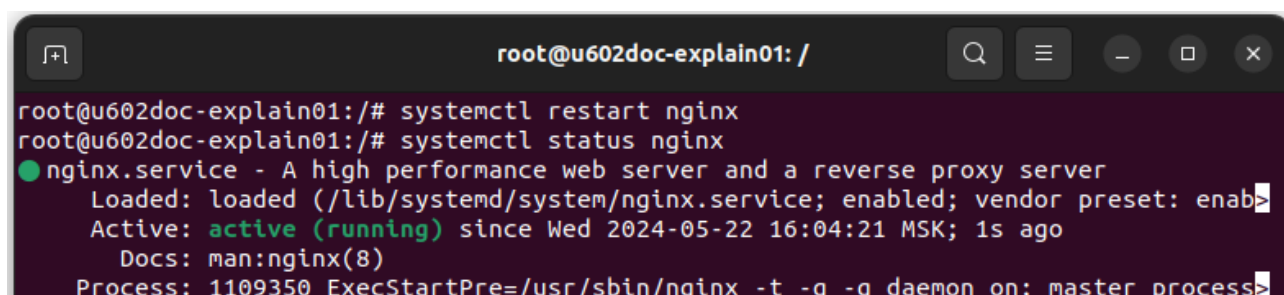
    listen 443 ssl;
    ssl_certificate /etc/nginx/ssl/nginx-explain.crt;
    ssl_certificate_key /etc/nginx/ssl/nginx-explain.key;

    location / {
        proxy_pass http://localhost:8080;
    }
}
```

Рисунок 5.5 – Содержание файл конфигурации сайта

Перезапустить службу nginx:

```
systemctl restart nginx
```



```
root@u602doc-explain01: /
root@u602doc-explain01:/# systemctl restart nginx
root@u602doc-explain01:/# systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enable>
   Active: active (running) since Wed 2024-05-22 16:04:21 MSK; 1s ago
     Docs: man:nginx(8)
   Process: 1109350 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master process>
```

Рисунок 5.6 – Перезапуск и вывод статуса службы nginx

Проверить в браузере работу explain по https, дать подтверждение системе безопасности при запросе про недействительный сертификат:

```
https://<адрес сервера explain>
```

В рассматриваемом примере используется адрес:

```
https://localhost:8080/
```



Рисунок 5.7 – Проверка работы сайта

5.1.4. Конфигурирование компонента JDS на отдельном узле

Взаимодействие JDS с сервисом pg-explain настраивается в конфигурационном файле компонента JDS appsettings.json. В свойстве PgExplainConfig.BaseAddress указать URL, по которому доступен https-сервис pg-explain.

Например

Выполнить команду редактирования файла appsettings.json:

```
nano /opt/jds/appsettings.json
```

Установить строки с синтаксисом:

```
"PgExplainConfig": {
  "BaseAddress": "https://<адрес сервера explain>"
```



Адрес должен быть указан без знака дробной черты (solidus) «/»

В рассматриваемом примере один из узлов имеет IP-адрес 10.116.102.59 и строка конфигурационного файла appsettings.json компонента JDS будет иметь следующий вид:

```
"PgExplainConfig": {
  "BaseAddress": "https://10.116.102.59"
```




Рисунок 5.8 – Содержание конфигурационного файла appsettings.json компонента JDS

Войти в веб-интерфейс компонента JDS. Перейти в подраздел «Анализ запросов». На вкладке «Настройки» нажать кнопку «Добавить». Ввести IP-адрес узла с наблюдаемой СУБД, порт (если он отличается от стандартного 5432) и отметить флагами пункты собираемой статистики и сохранить.

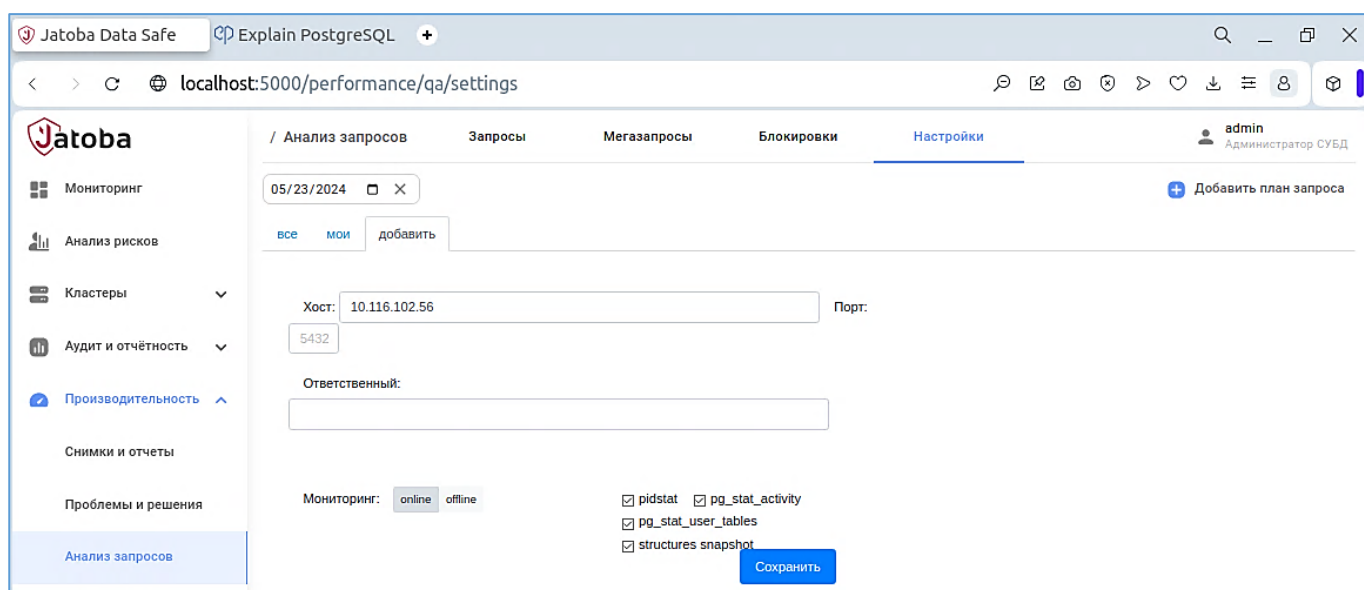


Рисунок 5.9 – Вкладка «Настройки» раздела «Анализ запросов»

5.2. Настройка pg-explain на одном узле с JDS

Компонент JDS может быть установлен до установки pg-explain. Порядок установки компонент не принципиален. Связь компонентом обеспечивается веб-сервером nginx с конфигурациями под каждый из компонентов.

В силу особенностей конфигурации компонентов потребуется редактировать параметры портов по протоколу HTTPS.

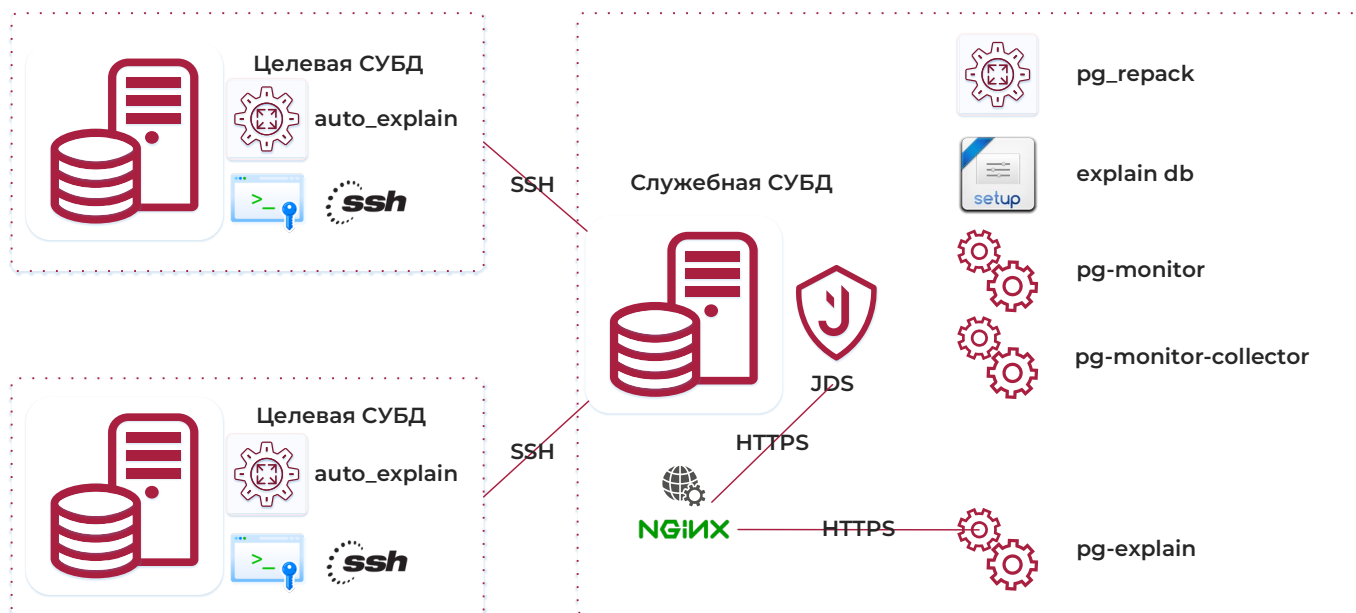


Рисунок 5.10 – Схема взаимодействия компонентов

5.2.1. Установка компонента JDS

Компонент пользовательского веб-интерфейса для администраторов «Jatoba data safe» (JDS) устанавливается в соответствии с документом «Защищенная система управления базами данных «Jatoba». Руководство по настройке. Часть 7. Пользовательский веб-интерфейс для администраторов. Компонент «Jatoba data safe», в зависимости от требуемой архитектуры, описанной в разделе 2 документа.

5.2.2. Веб-сервер nginx

Веб-сервер nginx устанавливается в соответствии с документом «Защищенная система управления базами данных «Jatoba». Руководство по настройке. Часть 7. Пользовательский веб-интерфейс для администраторов. Компонент «Jatoba data safe».

Выполнять действия, описанные в п. 5.1.1 документа, необязательно.

5.2.3. Создание сертификата и ключа для pg-explain

Создание сертификата и ключа для соединения по протоколу SSL описано в п. 5.1.2 документа.

5.2.4. Создание конфигурации сайта

Отличие выполняемых шагов, описанных в п. 5.1.3 документа состоит в том, что вместо порта 443 будет использоваться порт 444 SSL.

Для чего потребуется создать или отредактировать файл конфигурации сайта командой:

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

```
nano /etc/nginx/conf.d/explain.https.conf
```

Вставить текст и сохранить строки:

```
server {
    charset utf-8;

    access_log /var/log/nginx/explain.access.log;
    error_log /var/log/nginx/explain.error.log;

    listen 444 ssl;
    ssl_certificate /etc/nginx/ssl/nginx-explain.crt;
    ssl_certificate_key /etc/nginx/ssl/nginx-explain.key;

    location / {
        proxy_pass http://localhost:8080;
    }
}
```



Рисунок 5.11 – Содержание файла explain.https.conf

Применение параметров обеспечивается перезагрузкой службы nginx:

```
systemctl restart nginx
```

Проверить в веб-браузере работу explain по https, дать подтверждение системе безопасности, если спросит про недействительный сертификат.

5.2.5. Редактирование параметров компонента JDS

Взаимодействие JDS с сервисом pg-explain настраивается в конфигурационном файле компонента JDS appsettings.json. В свойстве PgExplainConfig.BaseAddress необходимо указать URL, по которому доступен https-сервис pg-explain.

Выполнить команду редактирования конфигурационного файла компонента JDS appsettings.json, командой:

```
nano /opt/jds/appsettings.json
```

Вставить параметры:

```
"PgExplainConfig": {  
  "BaseAddress": "https://<адрес сервера explain>:444"  
},
```



Адрес должен быть указан без знака дробной черты (solidus) «/»



Сервис explain работает на том же хосте, что и JDS, но в свойстве BaseAddress нужно указывать внешний IP-адрес (не localhost), т.к. обращение к pg-explain идет не от JDS, а от веб-браузера пользователя.

В рассматриваемом примере один из узлов имеет IP-адрес 10.116.102.59 и строка конфигурационного файла appsettings.json компонента JDS будет иметь следующий вид:

```
"PgExplainConfig": {  
  "BaseAddress": "https://10.116.102.59:444"
```

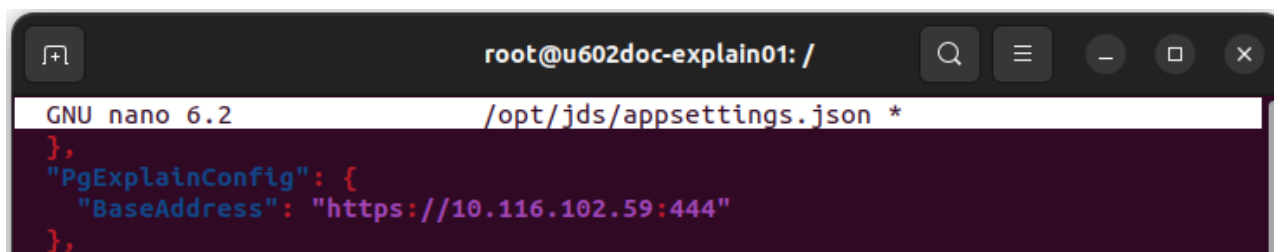


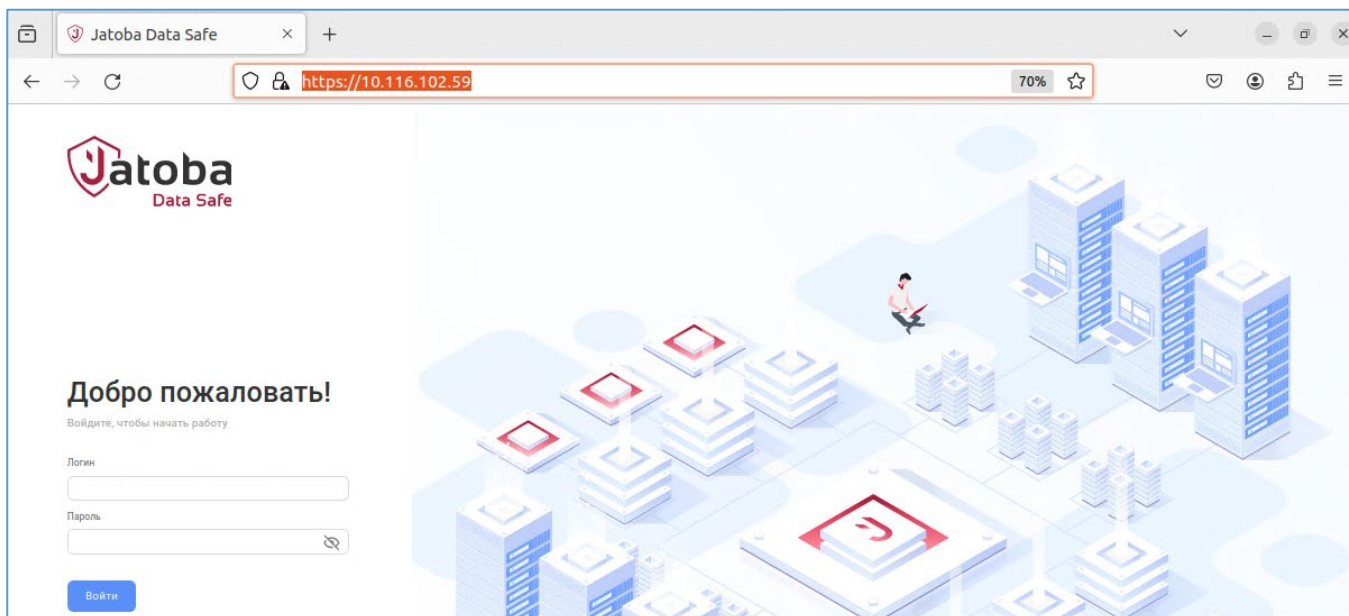
Рисунок 5.12 – Содержание конфигурационного файла приложения JDS appsettings.json
Сохранить файл и перезапустить службу «jds»

```
# systemctl restart jds
```

```
# systemctl status jds
```

Проверить доступность компонента JDS по адресу:

<https://10.116.102.59/>



Войти в веб-интерфейс компонента JDS. Перейти в подраздел «Анализ запросов». На вкладке «Настройки» нажать кнопку «Добавить». Ввести IP-адрес узла с наблюдаемой СУБД, порт (если он отличается от стандартного 5432) и отметить флагами пункты собираемой статистики и сохранить.

После чего отобразится добавленный узел.

6. ОБНОВЛЕНИЕ PG-EXPLAIN

6.1. Предварительные требования

Для выполнения обновления должны быть выполнены следующие требования:

- установлены СУБД Jatoba 6, JDS, explain версии 1.5.6;
- обновление происходит до версии explain 1.6.5;
- команды приведены для ОС Астра 1.7.

6.2. Процесс обновления

Остановить службы monitor и explain:

```
systemctl stop pg-monitor && systemctl stop pg-explain
```

Распаковать новый дистрибутив:

```
tar xvf /tmp/ jds-2.7.0-linux-x64-release-deb.tar.gz -C /usr/share/
```

Распаковать файлы новой версии pg-explain-db:

```
# cd /usr/share/jds/packages  
# apt install ./pg-explain-db*
```

Перейти в каталог со скриптами установки, задать права:

```
# cd /usr/local/lib/pg-explain-db  
# chmod 755 install.sh ./scripts/clear_dsk_space.sh  
./scripts/create_partitions.sh ./scripts/repack.sh
```

Изменить путь к бинарному файлу гераск в конфигурации:

```
sed -i 's/usr\pgsql-14\bin\/\/usr\jatoba-6\bin\/\/g'  
./scripts/repack.sh
```

Запустить процедуру обновления служебной базы данных:

```
./install.sh updatedb
```

Запустить установку новой версии pg-monitor:

```
# cd /usr/share/jds/packages  
# apt install ./pg-monitor_1.6.5-20240427_amd64.deb
```

Запустить службу и проверить ее статус:

```
systemctl start pg-monitor && systemctl status pg-monitor
```

Запустить установку новой версии pg-explain:

```
apt install ./pg-explain_1.6.2-20240427_amd64.deb
```

Запустить службу и проверить ее статус

```
systemctl start pg-explain.service
```

7. ОШИБКИ

7.1. Ошибка FATAL: password authentication failed for user "postgres"

Ошибка появляется при вводе некорректного пароля привилегированного пользователя СУБД.



```
root@u602doc-explain01: /usr/local/lib/pg-explain-db
root@u602doc-explain01:/usr/local/lib/pg-explain-db# ./install.sh
Enter database hostname (localhost):
Enter database port (5432):
Enter database name (pg-monitor):
Enter username to create database (postgres):
Enter password for user "postgres":
Enter pg-explain database user (explain):
Enter password for pg-explain user (explain):

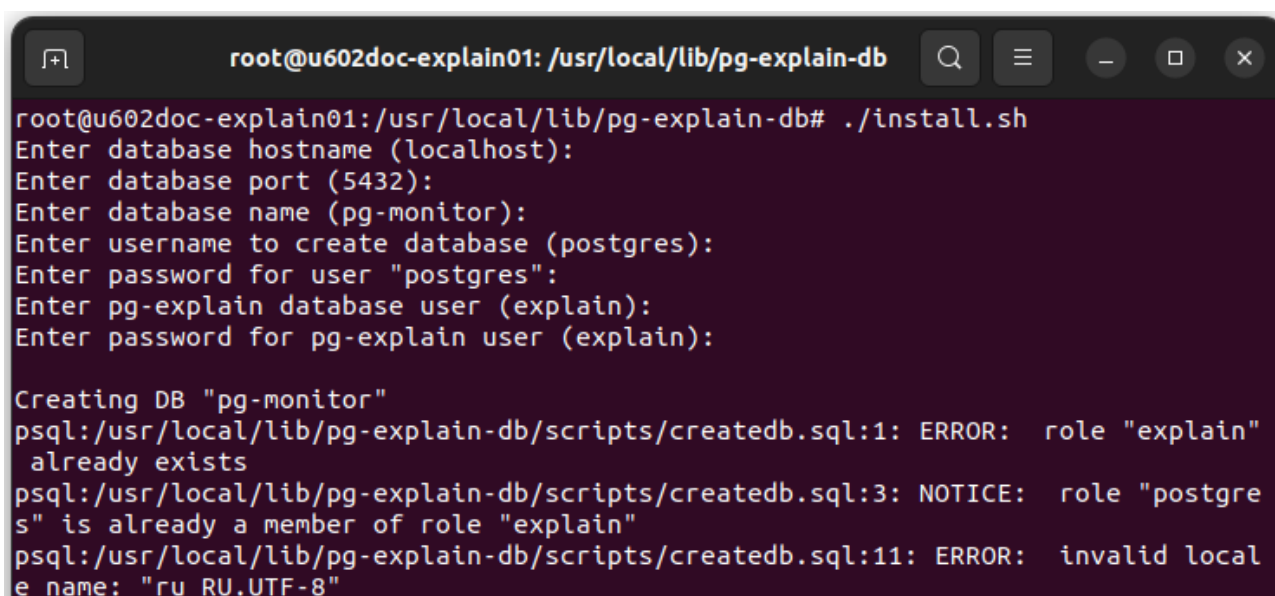
Creating DB "pg-monitor"
psql: error: connection to server at "localhost" (127.0.0.1), port 5432 failed:
FATAL:  password authentication failed for user "postgres"
password retrieved from file "/usr/local/lib/pg-explain-db/.pgpass"
Create DB failed.
root@u602doc-explain01:/usr/local/lib/pg-explain-db#
```

Рисунок 7.1 – Ошибка ввода пароля

Необходимо повторно запустить инсталлятор и указать корректный пароль.

7.2. Ошибка ERROR: invalid locale name: "ru_RU.UTF-8"

Ошибка появляется при отсутствии установленной в ОС локали «ru_RU.UTF-8».



```
root@u602doc-explain01: /usr/local/lib/pg-explain-db
root@u602doc-explain01:/usr/local/lib/pg-explain-db# ./install.sh
Enter database hostname (localhost):
Enter database port (5432):
Enter database name (pg-monitor):
Enter username to create database (postgres):
Enter password for user "postgres":
Enter pg-explain database user (explain):
Enter password for pg-explain user (explain):

Creating DB "pg-monitor"
psql:/usr/local/lib/pg-explain-db/scripts/createdb.sql:1: ERROR:  role "explain"
already exists
psql:/usr/local/lib/pg-explain-db/scripts/createdb.sql:3: NOTICE:  role "postgres"
is already a member of role "explain"
psql:/usr/local/lib/pg-explain-db/scripts/createdb.sql:11: ERROR:  invalid locale
name: "ru_RU.UTF-8"
```

Рисунок 7.2 – Ошибка при неустановленной локали

Необходимо установить системную локаль «ru_RU.UTF-8» командой:

```
dpkg-reconfigure locales
```

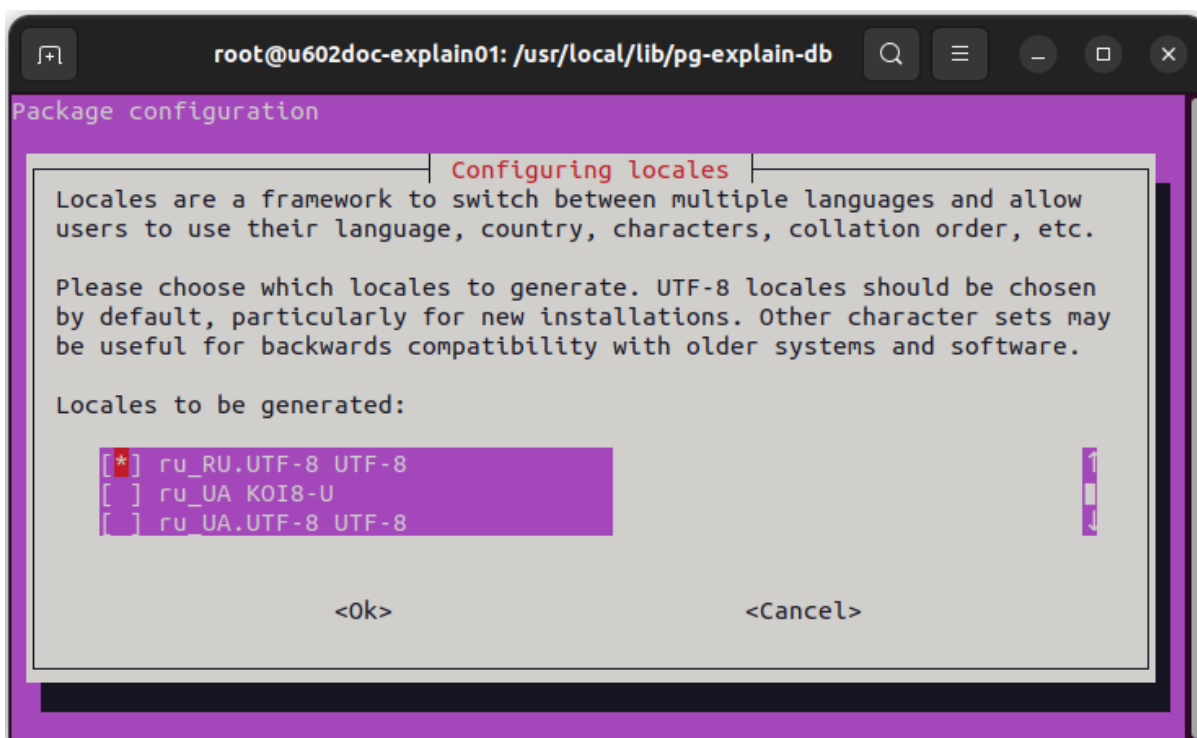


Рисунок 7.3 – Выбор локали в скрипте установки

В случае, если СУБД была установлена без локали «ru_RU.UTF-8» в ОС, то потребуется ее переустановка.

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

«/» – ГОСТ 34.302.2-91 (ИСО 8859/2-87) «Наборы 8 битных однобайтовых кодированных графических символов. латинский алфавит № 2» определяет символ как, «дробная черта» (англ. «solidus»).

SSH (Secure Shell) – сетевой протокол прикладного уровня, который позволяет осуществлять удаленное управление ОС и туннелирование TCP-соединений, например, для передачи файлов. Он шифрует весь трафик, включая передаваемые пароли, и допускает выбор различных алгоритмов шифрования. SSH-клиенты и серверы доступны для большинства сетевых ОС.

Аутентификация «peer» – режим аутентификации, при использовании которого пользователи автоматически аутентифицируются, если существует соответствующий пользователь СУБД с именем, совпадающим с именем ОС пользователя.

UFW (Uncomplicated Firewall) – утилита для конфигурирования межсетевого экрана Netfilter в ОС Linux. Она использует интерфейс командной строки и состоит из небольшого количества простых команд для лёгкого управления межсетевым экраном.

Iptables – утилита командной строки, стандартный интерфейс управления межсетевым экраном netfilter для ядер Linux, начиная с версии 2.4. Iptables используется для настройки правил фильтрации пакетов, маршрутизации и преобразования сетевых адресов.

Bruteforce (брутфорс) – метод взлома, при котором злоумышленник или тестировщик подбирает данные для входа в систему, используя различные комбинации паролей. Этот метод эффективен для взлома аккаунтов с простыми паролями, но сложен для сложных комбинаций, шифровок и фраз.

OpenSSH – набор программ, предоставляющих шифрование сеансов связи по компьютерным сетям с использованием протокола SSH. OpenSSH включает программы для клиента и сервера, а также инструменты для генерации ключей и аутентификации.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

SQL	–	Structured Query Language
БД	–	База данных
ОС	–	Операционная система
СУБД	–	Система управления базами данных

[illegible]

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------