Удаление "зависших" (долго исполняющихся) задач

1. Проблема

Запущенная задача выполняется слишком долго (происходит процесс "зависания" / бесконечного выполнения задачи).

2. Решение

2.1. Вариант 1. Рекомендуемый метод

1. Проверить на хосте, который в данный момент обладает ролью `Storage Pool Manager (SPM)`, список запущенных задач командой `vdsm-client Host getAllTasksInfo`, например:

```
vdsm-client Host getAllTasksInfo
{
    "9022a6e0-06cf-4066-b9f7-cbe23ffe851e": {
        "verb": "copyImage",
        "id": "9022a6e0-06cf-4066-b9f7-cbe23ffe851e"
    },
    "954d8ea4-6ab6-4ec5-9d1a-3374ec106a8c": {
        "verb": "prepareMerge",
        "id": "954d8ea4-6ab6-4ec5-9d1a-3374ec106a8c"
    },
    "c3fb509c-0554-4f10-9f73-32d346ccf012": {
        "verb": "copyImage",
        "id": "c3fb509c-0554-4f10-9f73-32d346ccf012"
    }
}
```

2. После получения идентификатора задачи необходимо узнать её статус с помощь команды `vdsm-client Task getStatus taskID=<TASKID>`, где `<TASKID>` получен из предыдущей команды. Например:

```
vdsm-client Task getStatus taskID=c3fb509c-0554-4f10-9f73-32d346ccf012
{
    "message": "running job 1 of 1",
    "code": 0,
```

```
    "taskID": "c3fb509c-0554-4f10-9f73-32d346ccf012",
    "taskResult": "",
    "taskState": "running"
}
```

3. Для остановки задачи и очистки задачи используются следующие команды:

```
vdsm-client Task stop taskID=<TaskID>
vdsm-client Task clear taskID=<TaskID>
```

4. После остановки (очистки) необходимо проверить, список всех задач, например:

```
vdsm-client Host getAllTasksInfo
```

5. Если задача не была остановлена, то необходимо обратиться к альтернативному решению проблемы.

2.2. Вариант 2. Альтернативный метод

Если при попытке получить список задач на хосте в выводе пустое значение или задача не останавливается по методу, описанному в **Варианте 1**, то необходимо выполнить следующие действия:

1. Перевести кластер в режим обслуживания (на хосте).

```
hosted-engine --set-maintenance --mode=global
```

2. Подключиться к менеджеру управления по SSH / через консоль и перейти в каталог `/usr/share/ovirt-engine/setup/dbutils`
3. Выполнить команды для получения пароля от БД.

```
source /etc/ovirt-engine/engine.conf.d/10-setup-database.conf
export PGPASSWORD=$ENGINE_DB_PASSWORD
```

   Пример успешного выполнения:

```
echo $ENGINE_DB_PASSWORD
baaaAAAAAbbbbbbbbbbbbB
```

4. Остановить службу ovirt-engine.

```
systemctl stop ovirt-engine
```

5. Выполнить команду для удаления задач, которые выполняются слишком долго ("зависли" / бесконечно выполняются).

```
cd /usr/share/ovirt-engine/setup/dbutils
./taskcleaner.sh -RAz
```

6. В случае необходимости можно удалить **все задачи**:

```
cd /usr/share/ovirt-engine/setup/dbutils
./taskcleaner.sh -RA
```

7. Вывести кластер из режима обслуживания (на хосте).

```
# hosted-engine --set-maintenance --mode=none
```

8. Запустить службу ovirt-engine (на менеджере управления).

```
systemctl start ovirt-engine
```

Базовая конфигурация базы данных

1. Повторное использование базы данных для новой инсталляции невозможно.
2. Не поддерживается развертывание БД и брокера на одном сервере.

В этом разделе представлена базовая информация по подготовке сервера баз данных PostgreSQL.

РЕД ОС

В этой инструкции используется последняя доступная версия СУБД PostgreSQL 15. В случае если используется другая версия, то выполняемые команды необходимо изменить. Более подробно можно ознакомиться на сайте документации РЕД ОС.

1. Установите PostgreSQL с помощью команды:

```bash
sudo dnf install postgresql15-server
```

2. Инициализируйте базу данных с помощью команды:

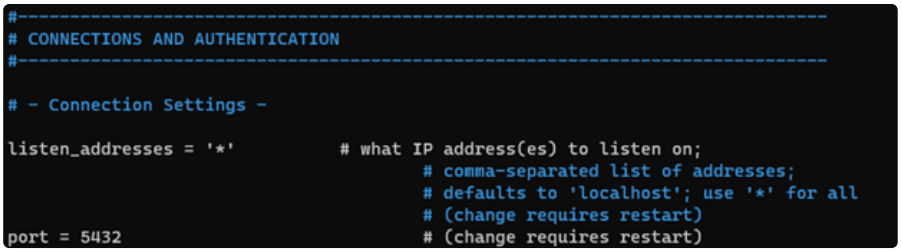
```bash
sudo postgresql-15-setup initdb
```

3. Запустите сервис PostgreSQL с помощью команды:

```bash
sudo systemctl enable postgresql-15.service --now
```

4. Чтобы разрешить удаленное подключение к СУБД, в файле конфигурации `/var/lib/pgsql/15/data/postgresql.conf` параметр `listen_addresses` должен соответствовать значению `'*'` :

В инструкции предлагается установить для параметра `listen_addresses` значение * . Это позволит принимать подключения по всем доступным сетевым интерфейсам. Если требуется ограничить доступ только к определенному интерфейсу, то укажите конкретный IP-адрес или имя интерфейса вместо * . Например: `listen_addresses` ='192.168.1.1' или `listen_addresses` ='eth0'. Это поможет ограничить подключение к СУБД только с нужного интерфейса, обеспечивая дополнительный уровень безопасности.



Отображение параметров на скриншоте может отличаться. Зависит от версии PostgreSQL.

5. Чтобы обеспечить достаточное количество слотов для подключений в высоконагруженной системе, в файле конфигурации `/var/lib/pgsql/15/data/postgresql.conf` параметр `max_connections` должен соответствовать значению `500`.

6. Чтобы разрешить удаленное подключение к СУБД с паролем, добавьте в файл конфигурации `/var/lib/pgsql/15/data/pg_hba.conf` строку:

```
host    all             all             0.0.0.0/0               password
```
BASH

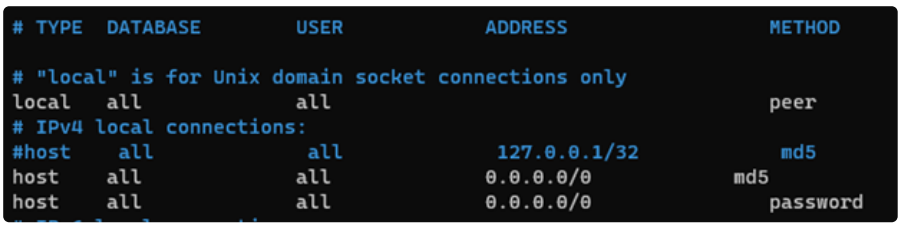
В инструкции предлагается добавить в файл конфигурации `/var/lib/pgsql/15/data/pg_hba.conf` строку:

```
host    all             all             0.0.0.0/0               password
```
BASH

Эта строка разрешает подключение к СУБД с использованием пароля со всех IP-адресов. Для повышения безопасности рекомендуется указать конкретные IP-адреса или диапазоны адресов, с которых будут разрешены подключения. Например:

```
host    all             all             192.168.1.0/24          password
```
BASH

Это ограничит доступ к СУБД только для адресов в указанном диапазоне, снижая риск несанкционированных подключений.



Отображение параметров на скриншоте может отличаться. Зависит от версии PostgreSQL.

7. Запустите сессию служебного пользователя postgres с помощью команды:

```
sudo su - postgres
```
BASH

8. Запустите командную оболочку postgres с помощью команды:

```bash
psql
```

9. Если при установке брокера:
   - будет выбрана опция «подключиться к уже существующей базе данных», то выполните шаги:
     a. Создайте пустую базу данных с помощью команды:

     ```bash
     CREATE DATABASE %Имя_Базы%;
     ```

     b. Создайте нового пользователя с паролем с помощью команды:

     ```bash
     CREATE USER %Имя_Пользователя% WITH PASSWORD '%Пароль_Пользователя%';
     ```

     c. Назначьте права владельца для созданного пользователя на созданную базу с помощью команды:

     ```bash
     ALTER DATABASE %Имя_Базы% OWNER TO %Имя_Пользователя%;
     ```

     d. Выйдите из командной оболочки psql и сессии пользователя postgres с помощью команды:

     ```bash
     exit
     ```

     e. Перезапустите сервис PostgreSQL с помощью команды:

     ```bash
     sudo systemctl restart postgresql-15.service
     ```

   - будет выбрана опция «создать новую базу данных», то выполните шаги:
     a. Создайте нового пользователя с паролем и правами на создание новых баз данных с помощью команды:

     ```bash
     CREATE USER %Имя_Пользователя% WITH PASSWORD '%Пароль_Пользователя%'
     CREATEDB;
     ```

     b. Выйдите из командной оболочки psql и сессии пользователя postgres с помощью команды:

     ```bash
     exit
     ```

     c. Перезапустите сервис PostgreSQL с помощью команды:

     ```bash
     sudo systemctl restart postgresql-15.service
     ```

Astra Linux

В случае инсталляций в режиме контроля мандатного доступа («Воронеж» и «Смоленск») может потребоваться дополнительная конфигурация операционной системы. Подробнее о настройках PostgreSQL и мандатном управлении доступа.

1. Установите PostgreSQL из репозитория с помощью команды:

```bash
sudo apt install postgresql-11
```

2. Проверьте статус сервиса PostgreSQL с помощью команды:

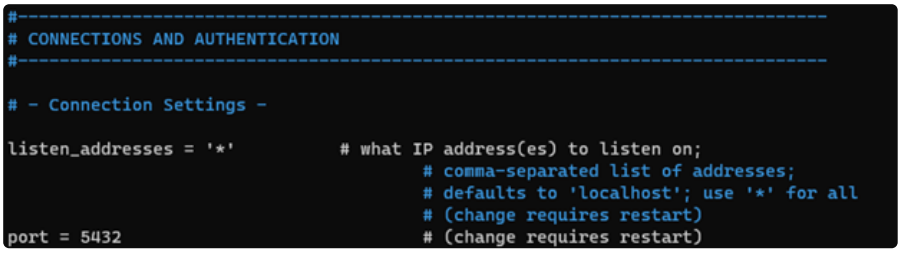
```bash
sudo systemctl status postgresql
```

3. При необходимости запустите сервис с помощью команды:

```bash
sudo systemctl start postgresql
```

4. Чтобы разрешить удаленное подключение к СУБД, в файле конфигурации `/etc/postgresql/11/main/postgresql.conf` параметр `listen_addresses` должен соответствовать значению `'*'`:

> В инструкции предлагается установить для параметра `listen_addresses` значение * . Это позволит принимать подключения по всем доступным сетевым интерфейсам. Если требуется ограничить доступ только к определенному интерфейсу, то укажите конкретный IP-адрес или имя интерфейса вместо * . Например: `listen_addresses`='192.168.1.1' или `listen_addresses`='eth0'. Это поможет ограничить подключение к СУБД только с нужного интерфейса, обеспечивая дополнительный уровень безопасности.



> Отображение параметров на скриншоте может отличаться. Зависит от версии PostgreSQL.

5. Чтобы обеспечить достаточное количество слотов для подключений в высоконагруженной системе, в файле конфигурации `/etc/postgresql/11/main/postgresql.conf` параметр `max_connections` должен соответствовать значению `500`.

6. Чтобы разрешить удаленное подключение к СУБД с паролем, добавьте в файл конфигурации `/etc/postgresql/11/main/pg_hba.conf` строку:

```
host    all             all             0.0.0.0/0               password
```

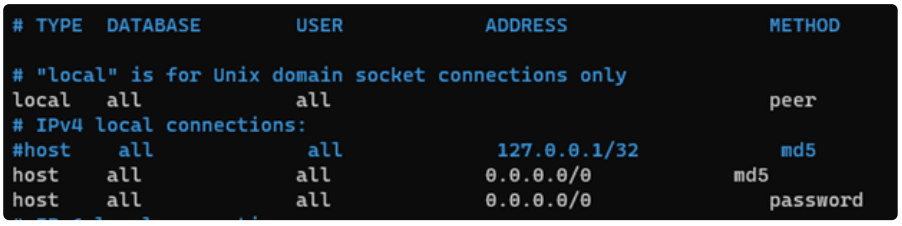
В инструкции предлагается добавить в файл конфигурации `/var/lib/pgsql/15/data/pg_hba.conf` строку:

```
host    all             all             0.0.0.0/0               password
```

Эта строка разрешает подключение к СУБД с использованием пароля со всех IP-адресов. Для повышения безопасности рекомендуется указать конкретные IP-адреса или диапазоны адресов, с которых будут разрешены подключения. Например:

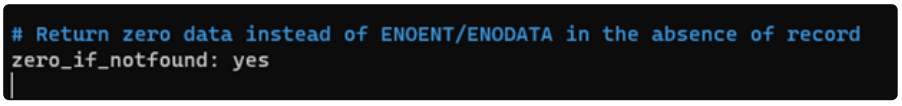
```
host    all             all             192.168.1.0/24          password
```

Это ограничит доступ к СУБД только для адресов в указанном диапазоне, снижая риск несанкционированных подключений.



Отображение параметров на скриншоте может отличаться. Зависит от версии PostgreSQL.

7. Чтобы разрешить удаленное подключение к СУБД пользователю в режиме контроля мандатного доступа, в параметре `zero_if_notfound` задайте значение `yes` в файле `/etc/parsec/mswitch.conf`:



8. Запустите сессию служебного пользователя postgres с помощью команды:

```
sudo su - postgres
```

9. Запустите командную оболочку postgres с помощью команды:

```
psql
```

10. Если при установке брокера:
    - будет выбрана опция «подключиться к уже существующей базе данных», то выполните шаги:
        a. Создайте пустую базу данных с помощью команды:

```
CREATE DATABASE %Имя_Базы%;
```

b. Создайте нового пользователя с паролем с помощью команды:

```
CREATE USER %Имя_Пользователя% WITH PASSWORD '%Пароль_Пользователя%';
```

c. Назначьте права владельца для созданного пользователя на созданную базу с помощью команды:

```
ALTER DATABASE %Имя_Базы% OWNER TO %Имя_Пользователя%;
```

d. Выйдите из командной оболочки psql и сессии пользователя postgres с помощью команды:

```
exit
```

e. Перезапустите сервис PostgreSQL с помощью команды:

```
sudo systemctl restart postgresql
```

- будет выбрана опция «создать новую базу данных», то выполните шаги:

a. Создайте нового пользователя с паролем и правами на создание новых баз данных с помощью команды:

```
CREATE USER %Имя_Пользователя% WITH PASSWORD '%Пароль_Пользователя%'
CREATEDB;
```

b. Выйдите из командной оболочки psql и сессии пользователя postgres с помощью команды:

```
exit
```

c. Перезапустите PostgreSQL с помощью команды:

```
sudo systemctl restart postgresql
```

Debian

1. Установите сервис PostgreSQL с помощью команды:

```
sudo apt install postgresql-15
```

2. Проверьте статус сервиса с помощью команды:

```bash
sudo systemctl status postgresql
```

3. При необходимости запустите сервис с помощью команды:

```bash
sudo systemctl enable postgresql --now
```

4. Чтобы разрешить удаленное подключение к СУБД, в файле конфигурации `/etc/postgresql/15/main/postgresql.conf`, в параметре `listen_addresses` должно быть значение '*':

5. Чтобы обеспечить достаточное количество слотов для подключений в высоконагруженной системе, в файле конфигурации `/etc/postgresql/15/main/postgresql.conf` параметр `max_connections` должен соответствовать значению `500`.

6. Чтобы разрешить удаленное подключение к СУБД, добавьте в файл конфигурации `/etc/postgresql/15/main/pg_hba.conf` строку:

```bash
host    all             all             0.0.0.0/0               password
```

7. Запустите сессию служебного пользователя postgres с помощью команды:

```bash
sudo su - postgres
```

8. Запустите командную оболочку postgres с помощью команды:

```bash
psql
```

9. Если при установке брокера:
   - будет выбрана опция «подключиться к уже существующей базе данных», то выполните шаги:
     a. Создайте пустую базу данных с помощью команды:

```bash
CREATE DATABASE %Имя_Базы%;
```

     b. Создайте нового пользователя с паролем с помощью команды:

```bash
CREATE USER %Имя_Пользователя% WITH PASSWORD '%Пароль_Пользователя%';
```

c. Назначьте права владельца для созданного пользователя на созданную базу с помощью команды:

```
ALTER DATABASE %Имя_Базы% OWNER TO %Имя_Пользователя%;
```
BASH

d. Выйдите из командной оболочки psql и сессии пользователя postgres с помощью команды:

```
exit
```
BASH

e. Перезапустите сервис PostgreSQL с помощью команды:

```
sudo systemctl restart postgresql
```
BASH

- будет выбрана опция «создать новую базу данных», то выполните шаги:

a. Создайте нового пользователя с паролем и правами на создание новых баз данных с помощью команды:

```
CREATE USER %Имя_Пользователя% WITH PASSWORD '%Пароль_Пользователя%'
CREATEDB;
```
BASH

b. Выйдите из командной оболочки psql и сессии пользователя postgres с помощью команды:

```
exit
```
BASH

10. Перезапустите сервис PostgreSQL с помощью команды:

```
sudo systemctl restart postgresql
```
BASH

ALT Linux

Подробнее об установке PostgreSQL можно прочесть на официальном сайте ALT Linux.

1. Установите компоненты PostgreSQL с помощью команды:

```
sudo apt-get install postgresql11-server
```
BASH

2. Инициализируйте сервер БД с помощью команды:

```
sudo /etc/init.d/postgresql initdb
```
BASH

3. Запустите и добавьте автозапуск с помощью команды:

```bash
sudo systemctl enable postgresql --now
```

4. Чтобы разрешить удаленное подключение к СУБД, в файле конфигурации `/var/lib/pgsql/data/postgresql.conf` параметр `listen_addresses` должен соответствовать значению '*'.

В инструкции предлагается установить для параметра `listen_addresses` значение * . Это позволит принимать подключения по всем доступным сетевым интерфейсам. Если требуется ограничить доступ только к определенному интерфейсу, то укажите конкретный IP-адрес или имя интерфейса вместо * . Например: `listen_addresses` ='192.168.1.1' или `listen_addresses` ='eth0'. Это поможет ограничить подключение к СУБД только с нужного интерфейса, обеспечивая дополнительный уровень безопасности.

```
#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------

# - Connection Settings -

listen_addresses = '*'          # what IP address(es) to listen on;
                                        # comma-separated list of addresses;
                                        # defaults to 'localhost'; use '*' for all
                                        # (change requires restart)
#port = 5432                            # (change requires restart)
max_connections = 100                   # (change requires restart)
#superuser_reserved_connections = 3     # (change requires restart)
#unix_socket_directories = '/tmp'       # comma-separated list of directories
                                        # (change requires restart)
```

5. Чтобы обеспечить достаточное количество слотов для подключений в высоконагруженной системе, в файле конфигурации `/var/lib/pgsql/data/postgresql.conf` параметр `max_connections` должен соответствовать значению `500` .

6. Чтобы разрешить удаленное подключение к СУБД с паролем, добавьте в файл конфигурации `/var/lib/pgsql/data/pg_hba.conf` строку:

```bash
host    all             all             0.0.0.0/0               password
```

В инструкции предлагается добавить в файл конфигурации `/var/lib/pgsql/15/data/pg_hba.conf` строку:

```bash
host    all             all             0.0.0.0/0               password
```

Эта строка разрешает подключение к СУБД с использованием пароля со всех IP-адресов. Для повышения безопасности рекомендуется указать конкретные IP-адреса или диапазоны адресов, с которых будут разрешены подключения. Например:

```bash
host    all             all             192.168.1.0/24          password
```

Это ограничит доступ к СУБД только для адресов в указанном диапазоне, снижая риск несанкционированных подключений.

```
# 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
host    all             all             0.0.0.0/0               password
# IPv6 local connections:
host    all             all             ::1/128                 trust
```

7. Перезапустите сервис с помощью команды:

```bash
sudo systemctl restart postgresql
```

8. Перейдите в консоль управления БД с помощью команды:

```bash
psql –U postgres
```

9. Если при установке брокера:
   - будет выбрана опция «подключиться к уже существующей базе данных», то выполните шаги:
     a. Создайте пустую базу данных с помощью команды:

```bash
CREATE DATABASE %Имя_Базы%;
```

     b. Создайте нового пользователя с паролем с помощью команды:

```bash
CREATE USER %Имя_Пользователя% WITH PASSWORD '%Пароль_Пользователя%';
```

     c. Назначьте права владельца для созданного пользователя на созданную базу с помощью команды:

```bash
ALTER DATABASE %Имя_Базы% OWNER TO %Имя_Пользователя%;
```

     d. Выйдите из командной оболочки psql и сессии пользователя postgres с помощью команды:

```bash
exit
```

     e. Перезапустите сервис PostgreSQL с помощью команды:

```bash
sudo systemctl restart postgresql
```

   - будет выбрана опция «создать новую базу данных», то выполните шаги:
     a. Создайте нового пользователя с паролем и правами на создание новых баз данных с помощью команды:

```
CREATE USER %Имя_Пользователя% WITH PASSWORD '%Пароль_Пользователя%'
CREATEDB;
```
BASH

b. Выйдите из командной оболочки psql и сессии пользователя postgres с помощью команды:

```
exit
```
BASH

c. Перезапустите сервис PostgreSQL с помощью команды:

```
sudo systemctl restart postgresql
```
BASH

OpenSUSE

1. Установите компоненты PostgreSQL с помощью команды:

> При необходимости измените версию PostgreSQL.

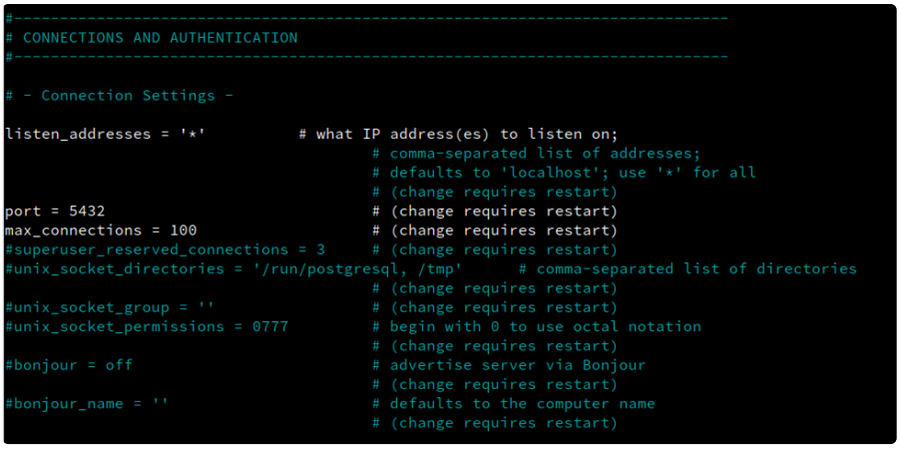
```
sudo zypper install postgresql postgresql-server postgresql-contrib
```
BASH

2. Запустите и добавьте автозапуск с помощью команды:

```
sudo systemctl enable postgresql --now
```
BASH

3. Чтобы разрешить удаленное подключение к СУБД, в файле конфигурации `/var/lib/pgsql/data/postgresql.conf`, параметр `listen_addresses` должен соответствовать значению '*'.

> В инструкции предлагается установить для параметра `listen_addresses` значение * . Это позволит принимать подключения по всем доступным сетевым интерфейсам. Если требуется ограничить доступ только к определенному интерфейсу, то укажите конкретный IP-адрес или имя интерфейса вместо * . Например: `listen_addresses`='192.168.1.1' или `listen_addresses`='eth0'. Это поможет ограничить подключение к СУБД только с нужного интерфейса, обеспечивая дополнительный уровень безопасности.



4. Чтобы обеспечить достаточное количество слотов для подключений в высоконагруженной системе, в файле конфигурации `/var/lib/pgsql/data/postgresql.conf` параметр `max_connections` должен соответствовать значению `500`.

5. Чтобы разрешить удаленное подключение к СУБД с паролем, добавьте в файл конфигурации `/var/lib/pgsql/data/pg_hba.conf` строку:

```
BASH
host    all             all             0.0.0.0/0               password
```

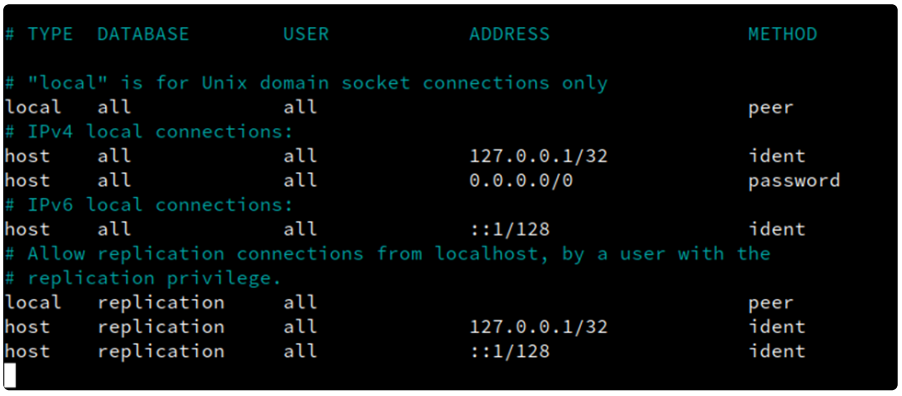
В инструкции предлагается добавить в файл конфигурации `/var/lib/pgsql/15/data/pg_hba.conf` строку:

```
BASH
host    all             all             0.0.0.0/0               password
```

Эта строка разрешает подключение к СУБД с использованием пароля со всех IP-адресов. Для повышения безопасности рекомендуется указать конкретные IP-адреса или диапазоны адресов, с которых будут разрешены подключения. Например:

```
BASH
host    all             all             192.168.1.0/24          password
```

Это ограничит доступ к СУБД только для адресов в указанном диапазоне, снижая риск несанкционированных подключений.



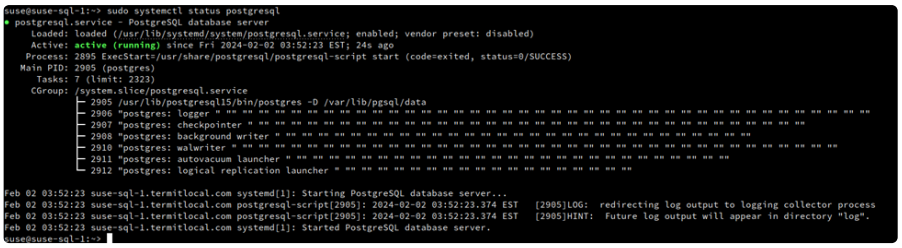
6. Перезапустите сервис с помощью команды:

```bash
sudo systemctl restart postgresql
```

7. Проверьте статус сервиса с помощью команды:

```bash
sudo systemctl status postgresql
```

Статус должен быть «Active».



8. Перейдите в консоль управления БД с помощью команды:

```bash
psql –U postgres
```

9. Если при установке брокера:
   - будет выбрана опция «подключиться к уже существующей базе данных», то выполните шаги:
     a. Создайте пустую базу данных с помощью команды:

```bash
CREATE DATABASE %Имя_Базы%;
```

     b. Создайте нового пользователя с паролем с помощью команды:

```bash
CREATE USER %Имя_Пользователя% WITH PASSWORD '%Пароль_Пользователя%';
```

     c. Назначьте права владельца для созданного пользователя на созданную базу с помощью команды:

```bash
ALTER DATABASE %Имя_Базы% OWNER TO %Имя_Пользователя%;
```

     d. Выйдите из командной оболочки psql и сессии пользователя postgres с помощью команды:

```bash
exit
```

     e. Перезапустите сервис PostgreSQL с помощью команды:

```bash
sudo systemctl restart postgresql
```

   - будет выбрана опция «создать новую базу данных», то выполните шаги:

a. Создайте нового пользователя с паролем и правами на создание новых баз данных с помощью команды:

```bash
CREATE USER %Имя_Пользователя% WITH PASSWORD '%Пароль_Пользователя%'
CREATEDB;
```

b. Выйдите из командной оболочки psql и сессии пользователя postgres с помощью команды:

```bash
exit
```

c. Перезапустите сервис PostgreSQL с помощью команды:

```bash
sudo systemctl restart postgresql
```

Если при установке выбирается опция «подключиться к уже существующей базе данных», вместо назначения пользователя владельцем можно использовать минимально необходимый набор прав. Для этого выполните следующие шаги:

1. Создайте пустую базу данных с помощью команды:

```
CREATE DATABASE %Имя_Базы%;
```

2. Создайте нового пользователя с паролем с помощью команды:

```
CREATE USER %Имя_Пользователя% WITH PASSWORD '%Пароль_Пользователя%';
```

3. Предоставьте пользователю доступ к базе данных с помощью команды:

```
GRANT CONNECT ON DATABASE %Имя_Базы% TO %Имя_Пользователя%;
```

4. Подключитесь к базе данных с помощью команды:

```
\c %Имя_Базы%
```

5. Предоставьте пользователю права на создание объектов и использование схемы `public`:

```
GRANT CREATE, USAGE ON SCHEMA public TO %Имя_Пользователя%;
```

6. Предоставьте пользователю права на выполнение операций с уже существующими таблицами в схеме `public`:

```
GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA public TO
%Имя_Пользователя%;
```

7. Установите дефолтные привилегии для будущих объектов в схеме `public`:

```
ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT, INSERT, UPDATE, DELETE
ON TABLES TO %Имя_Пользователя%
```

Выпуск самоподписанного сертификата

1. Создание корневого сертификата (CA)

Ниже описано, как создать корневой сертификат с помощью OpenSSL.

1. Создайте закрытый ключ корневого сертификата с помощью команды:

   ```
   openssl genrsa -out ca.key 4096
   ```

2. Создайте корневой сертификат на основе закрытого ключа с помощью команды:

   ```
   openssl req -new -x509 -days 365 -key ca.key -out ca.cert.pem
   ```

   При появлении запроса введите пароль для закрытого ключа и сведения об организации, такие как страна или регион, штат, организация, подразделение и FQDN.

   Результат:

   - ca.key — закрытый ключ корневого сертификата;
   - ca.cert.pem — корневой сертификат.

2. Создание сертификата сервера

Ниже описано, как создать сертификат сервера с помощью OpenSSL.

1. Создайте закрытый ключ сертификата сервера с помощью команды:

   ```
   openssl genrsa -out server.key 4096
   ```

2. Создайте запрос на подпись сертификата на основе закрытого ключа с помощью команд:

   ```
   openssl req -new -key server.key -out server.csr -addext "subjectAltName = DNS:broker.example.com"
   ```

   ```
   openssl x509 -req -days 365 -in server.csr -CA ca.cert.pem -CAkey ca.key -CAcreateserial -out server.crt -extfile <(printf "subjectAltName=DNS:broker.example.com")
   ```

При появлении запроса введите пароль для корневого ключа и сведения об организации, такие как страна или регион, штат, организация, подразделение и FQDN. `CN` должно соответствовать DNS-имени сервера, например broker.example.com.

3. Выпустите сертификат на основе ранее сформированного запроса с помощью команды:

   ```
   openssl x509 -in server.crt -text -noout -ext subjectAltName
   ```

   Результат:

   - server.key — закрытый ключ для сертификата сервера;
   - server.crt — сертификат сервера.

4. Импортируйте ключ и сертификат в настройках HTTPS.

Настройка межсетевого экрана

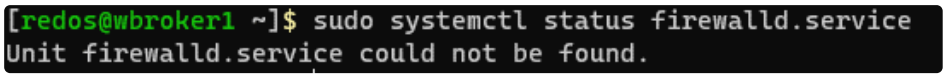
В этом разделе описано, как настроить межсетевой экран (МСЭ).

РЕД ОС

Чтобы проверить настройки МСЭ на каждом из серверов, выполните команду:

```bash
sudo systemctl status firewalld.service
```

Если в выводе команды отображается информация, как на изображении ниже, то дальнейшие шаги по конфигурации МСЭ можно пропустить, так как он выключен.



В случае если вывод команды отличается, то необходима дополнительная конфигурация МСЭ. Для этого:

1. Определите активную зону МСЭ с помощью команды:

   ```bash
   firewall-cmd --get-default-zone
   ```

2. Добавьте правила по списку портов в необходимую активную зону с помощью команды:

   ```bash
   sudo firewall-cmd --zone=%Активная_Зона% --permanent --add-port=%Порт%/tcp
   ```

   Где:

   - **%Активная_Зона%** — текущая активная зона из шага 1.
   - **%Порт%** — необходимый порт. Список портов смотрите в разделе Порты.

     Например:

     ```bash
     sudo firewall-cmd --zone=public --permanent --add-port=5432/tcp
     ```

     Подробнее о настройке МСЭ можно прочесть на официальном сайте РЕД ОС.

Astra Linux

Чтобы проверить настройки МСЭ на каждом из серверов, выполните команду:

```bash
sudo ufw status
```

Если выполнение команды возвращает ответ «Status: inactive», то дальнейшие шаги по конфигурации МСЭ можно пропустить, так как он выключен.

В случае если вывод команды отличается, то необходима дополнительная конфигурация МСЭ. Для этого добавьте разрешения командой:

```bash
sudo ufw allow %Порт%/tcp
```

Где: **%Порт%** — необходимый порт. Список портов смотрите в разделе Порты.

- Подробнее о настройке МСЭ можно прочесть на официальном сайте Astra Linux.
- При инсталляции Astra Linux в режиме контроля мандатного доступа может потребоваться дополнительная конфигурация.

ALT Linux

Чтобы проверить настройки МСЭ на каждом из серверов, выполните команду:

```bash
sudo efw
```

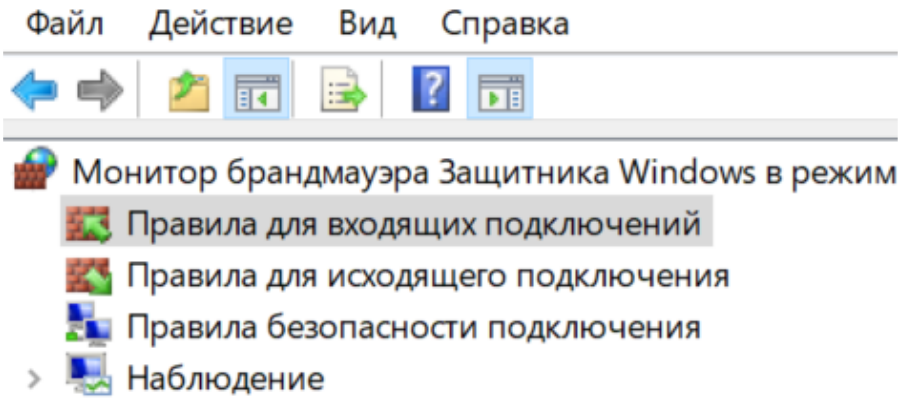
Если выполнение команды возвращает ответ «Firewall is disabled», то дальнейшие шаги по конфигурации МСЭ можно пропустить, так как он выключен.

В случае если вывод команды отличается, то необходима дополнительная конфигурация МСЭ. Для этого добавьте правила. В зависимости от используемого МСЭ команды могут отличаться. Подробнее о создании правил МСЭ описано в статьях Etcnet Firewall и UFW.

Windows

Чтобы настроить межсетевой экран на сервере:

1. Откройте любым способом **Монитор брандмауэра Защитника Windows в режиме повышенной безопасности**.
2. Выберите **Правила для входящих подключений**.



3. Нажмите **[Создать правило]**.
4. На вкладке **Тип правила** включите опцию **Для порта**.
5. Нажмите **[Далее]**.
6. На вкладке **Протокол и порты** оставьте **Протокол TCP** по умолчанию. В параметре **Определенные локальные порты:** укажите «8443». Полный список портов смотрите в разделе Порты.
7. Нажмите **Далее > Далее > Далее**.
8. На вкладке **Имя** задайте имя, например «Termit Agent».

Можно создать правило в межсетевом экране с помощью PowerShell:

```bash
New-NetFirewallRule -DisplayName "termit-agent TCP 8443" -Direction inbound -
Profile Any -Action Allow -LocalPort 8443 -Protocol TCP
```

Установка дополнительного брокера

В этом разделе описано, как установить второй и следующие брокеры.

Перед добавлением брокеров скопируйте дистрибутив на сервер, распакуйте архив и подготовьте ключ шифрования. Рекомендуется создать три брокера для повышения отказоустойчивости, так как:

- если один из брокеров выйдет из строя, то другие продолжат работать, обеспечивая бесперебойную работу системы;
- вероятность одновременного сбоя сразу трех брокеров гораздо ниже, что делает систему более устойчивой к неполадкам.

РЕД ОС

Перед установкой брокера выполните шаги 1-2 по установке Docker, Docker Compose или перейдите к шагу 3 (установочный скрипт выполнит шаги 1-2, запрашивая подтверждение):

Для установки компонентов необходимы настроенные репозитории.

1. Чтобы установить Docker, Docker Compose, выполните команду:

```
sudo dnf install docker-ce docker-ce-cli docker-compose
```

2. Чтобы установленные компоненты добавить в автозагрузку, выполните команду:

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

3. При необходимости назначьте права на запуск скрипта с помощью команды:

```
sudo chmod +x ./install.sh
```

4. Запустите скрипт:

```
sudo ./install.sh install
```

5. Введите пароль администратора системы.
6. Укажите имя узла брокера. Имя может быть любым.
7. Появится сообщение:

```
«What do you want to do? (1/2)
1. Install first node of new cluster
2. Add new node to existing cluster
Enter 1 or 2»
```

Укажите «2».

8. Укажите имя хоста базы данных (БД), например «db.example.com».
9. Укажите порт «5432».
10. Укажите имя БД, например «example».
11. Укажите имя пользователя БД, например «termit».
12. Введите пароль для БД.
13. Вставьте скопированный ключ шифрования.
14. Введите адрес, указанный на шаге, и проверьте статус брокера в меню после аутентификации.

Установка брокера занимает около двух минут. Для подтверждения успешной операции перейдите на портал администрирования и проверьте, что в разделе **Брокеры** брокер появился в списке со статусом «Работает».

Astra Linux

Перед установкой брокера выполните шаги 1-2 по установке Docker, Docker Compose или перейдите к шагу 3 (установочный скрипт выполнит шаги 1-2, запрашивая подтверждение):

> Для установки компонентов необходимы настроенные репозитории.

1. Чтобы установить Docker, Docker Compose, выполните команду:

```bash
sudo apt install docker.io docker-compose
```

2. Чтобы установленные компоненты добавить в автозагрузку, выполните команду:

```bash
sudo systemctl start docker
```

3. При необходимости назначьте права на запуск скрипта с помощью команды:

```bash
sudo chmod +x ./install.sh
```

4. Запустите скрипт:

```
sudo ./install.sh install
```

5. Введите пароль администратора системы.
6. Укажите имя узла брокера. Имя может быть любым.
7. Появится сообщение:

```
«What do you want to do? (1/2)
1. Install first node of new cluster
2. Add new node to existing cluster
Enter 1 or 2»
```

   Укажите «2».

8. Укажите имя хоста базы данных (БД), например «db.example.com».
9. Укажите порт «5432».
10. Укажите имя БД, например «example».
11. Укажите имя пользователя БД, например «termit».
12. Введите пароль для БД.
13. Вставьте скопированный ключ шифрования.
14. Введите адрес, указанный на шаге, и проверьте статус брокера в меню после аутентификации.

Установка брокера занимает около двух минут. Для подтверждения успешной операции перейдите на портал администрирования и проверьте, что в разделе **Брокеры** брокер появился в списке со статусом «Работает».

Debian

Перед установкой брокера выполните шаги 1-2 по установке Docker, Docker Compose или перейдите к шагу 3 (установочный скрипт выполнит шаги 1-2, запрашивая подтверждение):

> Для установки компонентов необходимы настроенные репозитории.

1. Чтобы установить Docker, Docker Compose, выполните команду:

```
sudo apt install docker.io docker-compose
```

2. Чтобы установленные компоненты добавить в автозагрузку, выполните команду:

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

3. При необходимости назначьте права на запуск скрипта с помощью команды:

```bash
sudo chmod +x ./install.sh
```

4. Запустите скрипт:

```bash
sudo ./install.sh install
```

5. Введите пароль администратора системы.
6. Укажите имя узла брокера. Имя может быть любым.
7. Появится сообщение:

```
«What do you want to do? (1/2)
1. Install first node of new cluster
2. Add new node to existing cluster
Enter 1 or 2»
```

   Укажите «2».

8. Укажите имя хоста базы данных (БД), например «db.example.com».
9. Укажите порт «5432».
10. Укажите имя БД, например «example».
11. Укажите имя пользователя БД, например «termit».
12. Введите пароль для БД.
13. Вставьте скопированный ключ шифрования.
14. Введите адрес, указанный на шаге, и проверьте статус брокера в меню после аутентификации.

Установка брокера занимает около двух минут. Для подтверждения успешной операции перейдите на портал администрирования и проверьте, что в разделе **Брокеры** брокер появился в списке со статусом «Работает».

ALT Linux

Перед установкой брокера выполните шаги 1-2 по установке Docker, Docker Compose или перейдите к шагу 3 (установочный скрипт выполнит шаги 1-2, запрашивая подтверждение):

> Для установки компонентов необходимы настроенные репозитории.

1. Чтобы установить Docker, Docker Compose, выполните команду:

```
sudo apt-get install docker-ce docker-compose
```

Для Docker Compose v2, устанавливаемого в качестве плагина (проверьте, что у вас установлен именно плагин с помощью команды `docker compose version` (не путать с `docker-compose --version`). Если вывод команды содержит установленную версию Docker Compose — у вас установлен плагин) необходимо создать символьную ссылку для корректной работы установочного скрипта. Для этого выполните команду:

```
sudo ln -s /usr/lib/docker/cli-plugins/docker-compose /usr/local/bin/docker-compose
```

2. Чтобы установленные компоненты добавить в автозагрузку, выполните команду:

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

3. При необходимости назначьте права на запуск скрипта с помощью команды:

```
sudo chmod +x ./install.sh
```

4. Запустите скрипт:

```
sudo ./install.sh install
```

5. Введите пароль администратора системы.
6. Укажите имя узла брокера. Имя может быть любым.
7. Появится сообщение:

```
«What do you want to do? (1/2)
1. Install first node of new cluster
2. Add new node to existing cluster
Enter 1 or 2»
```

Укажите «2».

8. Укажите имя хоста базы данных (БД), например «db.example.com».
9. Укажите порт «5432».
10. Укажите имя БД, например «example».
11. Укажите имя пользователя БД, например «termit».
12. Введите пароль для БД.
13. Вставьте скопированный ключ шифрования.
14. Введите адрес, указанный на шаге, и проверьте статус брокера в меню после аутентификации.

Установка брокера занимает около двух минут. Для подтверждения успешной операции перейдите на портал администрирования и проверьте, что в разделе **Брокеры** брокер появился в списке со статусом «Работает».

OpenSUSE

Перед установкой брокера выполните шаги 1-2 по установке Docker, Docker Compose или перейдите к шагу 3 (установочный скрипт выполнит шаги 1-2, запрашивая подтверждение):

Для установки компонентов необходимы настроенные репозитории.

1. Чтобы установить Docker, Docker Compose, выполните команду:

```
BASH
sudo zypper install docker docker-compose docker-compose-switch
```

2. Запустите службу с помощью команды:

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

3. При необходимости назначьте права на запуск скрипта с помощью команды:

```
BASH
sudo chmod +x ./install.sh
```

4. Запустите скрипт:

```
BASH
sudo ./install.sh install
```

5. Введите пароль администратора системы.
6. Укажите имя узла брокера. Имя может быть любым.
7. Появится сообщение:

```
«What do you want to do? (1/2)
1. Install first node of new cluster
2. Add new node to existing cluster
Enter 1 or 2»
```

   Укажите «2».

8. Укажите имя хоста базы данных (БД), например «db.example.com».
9. Укажите порт «5432».
10. Укажите имя БД, например «example».
11. Укажите имя пользователя БД, например «termit».

12. Введите пароль для БД.
13. Вставьте скопированный ключ шифрования.
14. Введите адрес, указанный на шаге, и проверьте статус брокера в меню после аутентификации.

Установка брокера занимает около двух минут. Для подтверждения успешной операции перейдите на портал администрирования и проверьте, что в разделе **Брокеры** брокер появился в списке со статусом «Работает».
