TERMIT

Termit SE Apps

Руководство по установке

Version 1, 2026-10-19

WIKI.ORIONSOFT.RU

Содержание

1. Подготовка окружения . . . 4
2. Межсетевой экран . . . 5
   - РЕД ОС . . . 5
   - Astra Linux . . . 6
   - ALT Linux . . . 6
3. База данных . . . 7
   - РЕД ОС . . . 7
   - Astra Linux . . . 10
   - ALT Linux . . . 14
   - Debian . . . 17
   - OpenSUSE . . . 19
4. Балансировка . . . 25
   - 4.1. Установка и настройка HAProxy . . . 26
     - Проброс пользовательского IP-адреса . . . 26
     - Установка и настройка HAProxy . . . 27
     - Передача сертификата клиента для аутентификации по смарт-карте . . . 42
   - 4.2. Установка и настройка Keepalived . . . 42
     - 4.2.1. Установка Keepalived на серверах . . . 42
     - 4.2.2. Настройка Keepalived на основном сервере . . . 43
     - 4.2.3. Настройка Keepalived на резервном сервере . . . 44
     - 4.2.4. Проверка работоспособности . . . 45
5. Брокеры . . . 47
   - РЕД ОС . . . 47
   - Astra Linux . . . 49
   - ALT Linux . . . 52
   - Debian . . . 54
   - OpenSUSE . . . 56
6. Дополнительный брокер . . . 59
   - РЕД ОС . . . 59
   - Astra Linux . . . 60
   - ALT Linux . . . 61
   - Debian . . . 63
   - OpenSUSE . . . 64
7. Дополнительное ПО . . . 66
   - РЕД ОС . . . 66
   - Astra Linux . . . 68

ALT Linux . . . 73
Debian . . . 75
OpenSUSE . . . 77
8. Аутентификация по сертификату . . . 81
При выключенном Kerberos на десктоп-клиенте . . . 81
Linux . . . 81
При включенном Kerberos на десктоп-клиенте . . . 84
Linux . . . 84
9. Шлюзы . . . 89
РЕД ОС . . . 89
Astra Linux . . . 89
ALT Linux . . . 90
Debian . . . 90
OpenSUSE . . . 91
10. ГОСТ-шифрование . . . 92
10.1. Настройка серверов . . . 92
10.2. Настройка клиентских компьютеров . . . 96
10.3. Обновление Termit агента шлюза с ГОСТ-stunnel . . . 97
11. Установка десктоп-клиента . . . 99
12. Ключи запуска десктоп-клиента . . . 102
Параметры запуска десктоп-клиента . . . 102
13. Сертификаты . . . 103
РЕД ОС . . . 103
Astra Linux . . . 104
ALT Linux . . . 105
OpenSUSE . . . 106
14. Выпуск самоподписанного сертификата . . . 108
14.1. Создание корневого сертификата (CA) . . . 108
14.2. Создание сертификата сервера . . . 108
15. Обновление пакетов десктоп-клиента на РЕД ОС при установке в закрытом контуре . . . 110
16. Смарт-карты . . . 112
Требования к архитектуре . . . 112
Требования к сертификату . . . 112
Требования к смарт-картам . . . 112
Настройка системы . . . 113
Настройка балансировщиков нагрузки . . . 113
Аутентификация в десктоп-клиенте и на портале администрирования на ОС Linux . . . 115
Аутентификация в приложениях на ТС под управлением ОС Linux . . . 115

17. Настройка печати . . . 117
Терминальный сервер на ОС Linux . . . 117
18. Деинсталляция компонентов . . . 119
Удаление брокера . . . 119
Удаление агента . . . 119
Удаление десктоп-клиента . . . 120
РЕД ОС . . . 120
Astra Linux . . . 120
ALT Linux . . . 120

Глава 1. Подготовка окружения

Перед развертыванием Termit подготовьте окружение:

1. Проверьте, что подсети Docker Compose не пересекаются с сетями для серверов баз данных, LDAP-каталогов, терминалов и брокеров. Подробнее в статье Ошибки при пересечении IP-адресов между Docker и хост-системой.
2. Разверните отдельные серверы:
   - для брокеров.

     Количество брокеров зависит от типа развертывания:

     - для «Standalone» необходим один брокер;
     - для «High Availability» - не меньше трех брокеров.

     Подробнее о расчете ресурсов в разделе.
   - для ТС под управлением Linux в зависимости от требований;
   - (Опционально) для сервера баз данных. Если в вашем окружении уже развернут сервер с базой данных Postgres, вы можете использовать его.
   - (Опционально) для MFA.
   - (Опционально) для балансировщиков нагрузки Haproxy.
3. Проверьте доступ к службе каталогов и PostgreSQL.
4. Добавьте терминальные серверы в домен службы каталогов.
5. В каталоге пользователей создайте сервисную учетную запись для синхронизации объектов из службы каталогов. Обязательно отключите опцию «User must change password at next logon» и включите «Password never expires».
6. В каталоге пользователей создайте группы безопасности для назначения ролей: администратора, технической поддержки, аудитора ИБ и пользователя, в соответствии с требованиями вашей компании.
7. В каталоге пользователей добавьте в состав групп, созданных на предыдущем шаге, учетные записи пользователей для взаимодействия с Termit.

Глава 2. Межсетевой экран

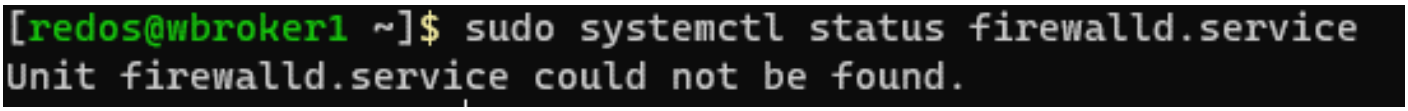
В этом разделе описано, как настроить межсетевой экран (МСЭ).

РЕД ОС

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

```
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:

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

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

   ```
   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:

     ```
     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:

```
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:

```
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:

```
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.

Глава 3. База данных

1. Повторное использование базы данных для новой инсталляции невозможно.
2. Не поддерживается развертывание базы данных 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 16. 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:

```
sudo dnf install postgresql16-server
```

2. Инициализируйте базу данных с помощью команды:

```
sudo postgresql-16-setup initdb
```

3. Запустите сервис PostgreSQL с помощью команды:

```
sudo systemctl enable postgresql-16.service --now
```

4. Чтобы разрешить удаленное подключение к СУБД, в файле конфигурации `/var/lib/pgsql/16/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. Чтобы принимать подключения к СУБД только через нужный интерфейс, 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)
```

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/16/data/postgresql.conf` для параметра `max_connections` укажите значение `500`.
6. Чтобы разрешить удаленное подключение к СУБД с паролем, добавьте в файл конфигурации `/var/lib/pgsql/16/data/pg_hba.conf` строку:

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

В инструкции предлагается добавить в файл конфигурации `/var/lib/pgsql/16/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.

```
# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     peer
# IPv4 local connections:
#host    all             all             127.0.0.1/32            md5
host    all             all             0.0.0.0/0           md5
host    all             all             0.0.0.0/0               password
```

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

7. Запустите сессию служебного пользователя postgres с помощью команды:

```
sudo su - postgres
```

8. Запустите командную оболочку PostgreSQL от имени пользователя `postgres` с помощью команды:

```
psql
```

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

```
CREATE DATABASE %Имя_Базы%;
```

     b. Создайте нового пользователя базы данных с паролем с помощью команды:

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

     c. Назначьте права владельца базы данных созданному пользователю с помощью команды:

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

     d. Завершите работу с оболочкой psql и сессией пользователя postgres с помощью команды:

```
exit
```

e. Примените изменения, перезапустив службу PostgreSQL с помощью команды:

```
sudo systemctl restart postgresql-16.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:

```
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 сервис PostgreSQL с помощью команды:

```
sudo systemctl restart postgresql-16.service
```

Astra Linux

СУБД PostgreSQL версии 15 доступен для Astra Linux версии 1.8. Для версии 1.7 и ниже - недоступен.

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:

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

2. Проверьте статус сервиса PostgreSQL с помощью команды:

```
sudo systemctl status postgresql
```

3. При необходимости запустите сервис с помощью команды:

```
sudo systemctl start postgresql
```

4. Чтобы разрешить удаленное подключение к СУБД, в файле конфигурации `/etc/postgresql/15/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.

```
#------------------------------------------------------------------------------
# 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)
```

> 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/15/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/15/main/pg_hba.conf` строку:

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

В инструкции предлагается добавить в файл конфигурации `/etc/postgresql/15/main/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.

```
# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     peer
# IPv4 local connections:
#host    all             all             127.0.0.1/32            md5
host    all             all             0.0.0.0/0            md5
host    all             all             0.0.0.0/0               password
```

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`:

```
# Return zero data instead of ENOENT/ENODATA in the absence of record
zero_if_notfound: yes
```

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:
    - выбрана опция «подключиться к 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
```

ALT Linux

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

1. Установите компоненты PostgreSQL с помощью команды:

```
sudo apt-get install postgresql15-server
```

2. Инициализируйте сервер БД с помощью команды:

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

3. Запустите и добавьте автозапуск с помощью команды:

```
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:

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

В инструкции предлагается добавить в файл конфигурации `/var/lib/pgsql/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.

```
# 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:

```
sudo systemctl restart postgresql
```

8. Перейдите в консоль управления БД с помощью команды:

```
psql -U postgres
```

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

```
CREATE DATABASE %Имя_Базы%;
```

     b. Создайте нового пользователя базы данных с паролем с помощью команды:

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

     c. Назначьте права владельца базы данных созданному пользователю с помощью команды:

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

     d. Завершите работу с оболочкой psql и сессией пользователя postgres с помощью команды:

```
exit
```

     e. Примените изменения, перезапустив службу 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 сервис PostgreSQL с помощью команды:

```
sudo systemctl restart postgresql
```

Debian

1. Установите сервис PostgreSQL с помощью команды:

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

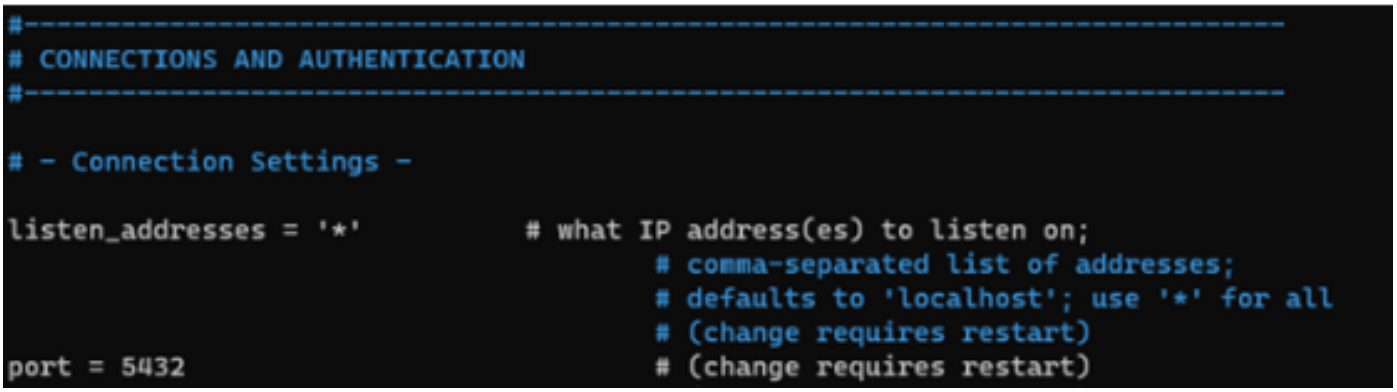
2. Проверьте статус сервиса с помощью команды:

```
sudo systemctl status postgresql
```

3. При необходимости запустите сервис с помощью команды:

```
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:

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

7. Запустите сессию служебного пользователя postgres с помощью команды:

```
sudo su - postgres
```

8. Запустите командную оболочку postgres с помощью команды:

```
psql
```

9. Если при установке брокера:
   - выбрана опция «подключиться к уже существующей базе данных», то выполните шаги:
     a. Создайте новую 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 базы данных созданному пользователю с помощью команды:

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

d. Завершите работу с оболочкой psql и сессией пользователя postgres с помощью команды:

```
exit
```

e. Примените изменения, перезапустив 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
```

10. Перезапустите сервис PostgreSQL с помощью команды:

```
sudo systemctl restart postgresql
```

OpenSUSE

1. Установите компоненты PostgreSQL с помощью команды:

При необходимости измените версию PostgreSQL.

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

2. Запустите и добавьте автозапуск с помощью команды:

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

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.

```
#------------------------------------------------------------------------------
# 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 = '/run/postgresql, /tmp'      # comma-separated list of directories
                                # (change requires restart)
#unix_socket_group = ''                 # (change requires restart)
#unix_socket_permissions = 0777         # begin with 0 to use octal notation
                                # (change requires restart)
#bonjour = off                          # advertise server via Bonjour
                                # (change requires restart)
#bonjour_name = ''                      # defaults to the computer name
                                # (change requires restart)
```

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:

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

В инструкции предлагается добавить в файл конфигурации

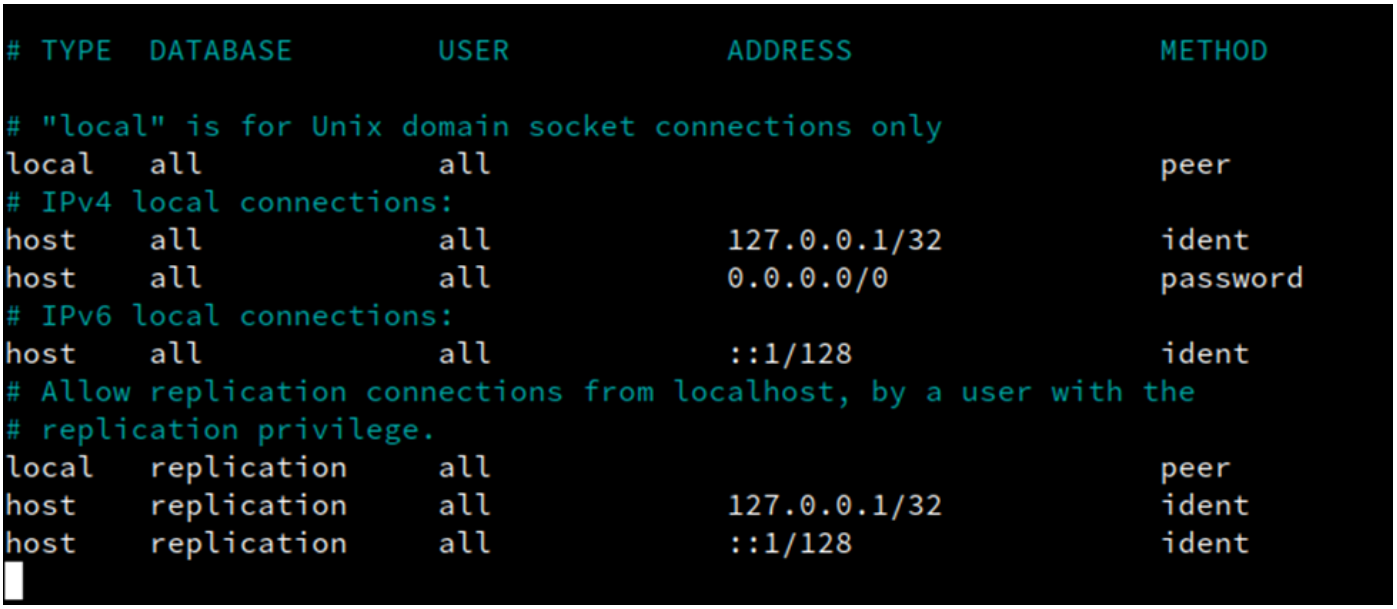
`/var/lib/pgsql/15/data/pg_hba.conf` строку:

```
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.



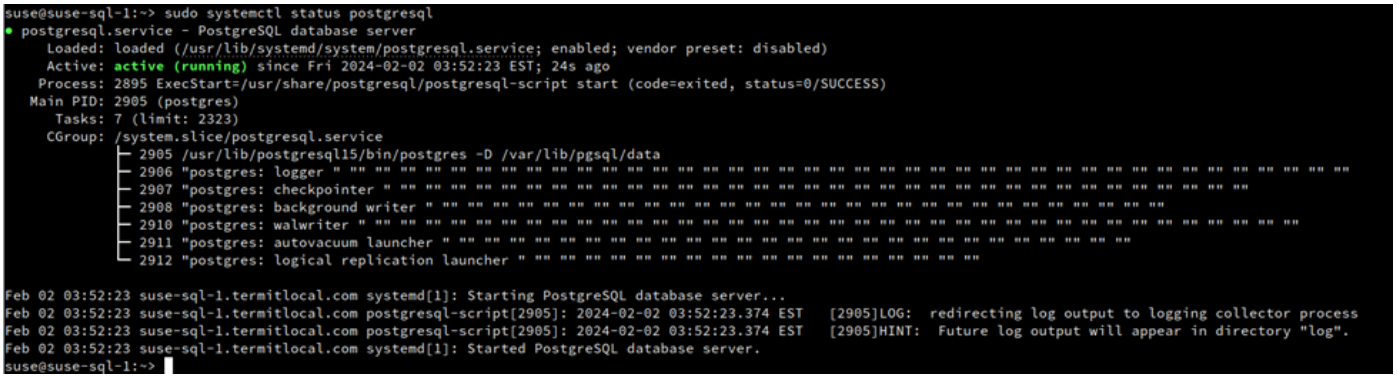
6. Перезапустите сервис с помощью команды:

```
sudo systemctl restart postgresql
```

7. Проверьте статус сервиса с помощью команды:

```
sudo systemctl status postgresql
```

Статус должен быть «Active».



8. Перейдите в консоль управления БД с помощью команды:

```
psql -U postgres
```

9. Если при установке брокера:
   - выбрана опция «подключиться к уже существующей базе данных», то выполните шаги:
     a. Создайте новую 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 базы данных созданному пользователю с помощью команды:

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

     d. Завершите работу с оболочкой psql и сессией пользователя postgres с помощью команды:

```
exit
```

     e. Примените изменения, перезапустив службу 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 сервис PostgreSQL с помощью команды:

```
sudo systemctl restart postgresql
```

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

1. Создайте новую базу данных с помощью команды:

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

2. Назначьте права владельца базы данных созданному пользователю 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. Назначьте пользователю необходимые привилегии на схему `public`:
   - `GRANT CREATE ON SCHEMA public TO %Имя_Пользователя%;`;

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

6. Установите дефолтные привилегии для будущих объектов в схеме `public` с помощью команды:

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

Глава 4. Балансировка

В этом разделе приведены инструкции по созданию балансировщика нагрузки с высокой доступностью, используя HAProxy и Keepalived. Эти инструменты помогут вам разработать отказоустойчивую архитектуру: она обеспечит минимальное время простоя и повысит производительность системы.

HAProxy и Keepalived обычно используются совместно для построения отказоустойчивой архитектуры, при этом выполняют разные функции:

- HAProxy отвечает за балансировку нагрузки и маршрутизацию трафика между брокерами. Он обеспечивает распределение подключений и поддерживает работу в режимах TCP/HTTP, а также SSL Passthrough или SSL Bridging.
- Keepalived отвечает за высокую доступность самого балансировщика. Он следит за состоянием узлов и с помощью механизма виртуального IP (VIP) автоматически переключает трафик на доступный узел в случае сбоя.

На практике оба сервиса устанавливают на одних и тех же серверах, но для полноценной отказоустойчивости требуется минимум два узла, между которыми Keepalived будет переключать VIP в случае сбоя.

Если вы планируете конфигурацию с одним брокером, то пропустите установку балансировщика нагрузки и перейдите к установке брокера,.

В текущей реализации поддерживаются:

- Балансировка на уровне TCP (для внутреннего трафика TC – брокер).
- Балансировка на уровне HTTP (для внешнего трафика десктоп-клиент – брокер).
- Режим работы SSL: Passthrough или Bridging.

Ниже приведены инструкции по установке и настройке HAProxy и Keepalived. Вы можете использовать другой балансировщик нагрузки, однако убедитесь, что он настроен для работы в режиме SSL Passthrough или SSL Bridging.

Перейти к:

- Установке и настройке HAProxy

- Установке и настройке Keepalived

4.1. Установка и настройка HAProxy

HAProxy (High Availability Proxy) — универсальный балансировщик нагрузки с открытым исходным кодом и программное обеспечение прокси-сервера. Его задача — эффективно распределять входящий сетевой трафик между несколькими брокерами для обеспечения высокой доступности, надежности и оптимальной производительности системы.

Для корректной работы системы выберите:

- алгоритм балансировки с привязкой пользователя к определенному серверу: balance source и hash-type consistent. Подробнее о типах балансировки;
- уровень балансировки (TCP/HTTP) и режим SSL (SSL Passthrough, SSL Bridging) в зависимости от направления трафика:
  - Балансировка на уровне TCP — используется для внутреннего трафика TC — брокер. Передаёт зашифрованный трафик напрямую, без расшифровки (SSL Passthrough).
  - Балансировка на уровне HTTP — применяется для внешнего трафика десктоп-клиент — брокер. Позволяет проверять сертификаты клиентов и добавлять заголовки (SSL Bridging).

Проброс пользовательского IP-адреса

Для корректного отображения реального IP-адреса клиента в журнале событий портала администратора необходимо включить проброс IP-адреса на балансировщике нагрузки.

Балансировщик должен передавать исходный IP-адрес клиента на серверы брокеров. В HAProxy это реализуется с помощью директив:

```
http-request set-header X-Real-IP %[src]
```

Где:

- `http-request set-header X-Real-IP %[src]` — передаёт реальный IP-адрес в заголовок X-Real-IP, который используется системой Termit для отображения в журналах.

Если используется другой балансировщик, необходимо настроить аналогичную функцию проброса исходного IP-адреса, чтобы брокеры получали реальный IP-адрес клиента. Эти настройки должны применяться в

> секции, которая обслуживает клиентский трафик (HTTP/HTTPS).

Установка и настройка HAProxy

Далее в разделе описано, как установить и настроить балансировщик нагрузки HAProxy для операционных систем: РЕД ОС, Astra Linux, Debian, Alt Linux и OpenSUSE.

РЕД ОС

Для установки используется последняя доступная в репозитории РЕД ОС версия HAProxy. Подробнее о конфигурации — на официальном сайте РЕД ОС.

1. Установите ПО HAProxy с помощью команды:

```
sudo dnf install haproxy
```

2. Включите службу haproxy и добавьте автозагрузку с помощью команды:

```
sudo systemctl enable haproxy --now
```

3. Добавьте в конец файла конфигурации `/etc/haproxy/haproxy.cfg` настройку серверов для балансировки.

Пример конфигурации для внутреннего балансировщика в режиме SSL Bridging

В примере приведен файл конфигурации с типом балансировки Source IP Hash.

```
frontend termit_https
   mode http
   bind :80
   bind :443 ssl crt /etc/haproxy/cert.pem
   http-request redirect scheme https unless { ssl_fc }
   http-request set-header X-Real-IP %[src]
   default_backend termit_servers_https

backend termit_servers_https
   option httpchk
   http-check connect port 9090 ssl
   http-check send meth GET  uri /_/broker-health
   mode http
   balance source
   hash-type consistent
   server %Имя_Термит_Брокера_1% %IP_Адрес_Термит_Брокера_1%:443 ssl check ca-file
/etc/haproxy/ca-int-broker-merged.pem verify required
   server %Имя_Термит_Брокера_2% %IP_Адрес_Термит_Брокера_2%:443 ssl check ca-file
```

```
/etc/haproxy/ca-int-broker-merged.pem verify required

frontend termit_8443_https
   mode tcp
   bind :8443
   default_backend termit_servers_8443_https

backend termit_servers_8443_https
   option httpchk
   http-check connect port 9090 ssl
   http-check send meth GET  uri /_/broker-health
   mode tcp
   balance leastconn
   server %Имя_Термит_Брокера_1% %IP_Адрес_Термит_Брокера_1%:8443 check check-ssl verify
none
   server %Имя_Термит_Брокера_2% %IP_Адрес_Термит_Брокера_2%:8443 check check-ssl verify
none
```

Если адрес внутреннего балансировщика «internal.termit.local», а адрес портала внутреннего доступа «internal.termit.local», тогда:

- `/etc/haproxy/cert.pem` - объединённый PEM-файл TLS-сертификата балансировщика HAProxy, содержащий:
  - сертификат на имя портала внутреннего доступа «internal.termit.local».
  - закрытый ключ от сертификата портала внутреннего доступа «internal.termit.local».
  - корневой сертификат, которым подписан сертификат портала внутреннего доступа «internal.termit.local».
- `/etc/haproxy/ca-int-broker-merged.pem` - файл, состоящий из:
  - корневого сертификата, которым подписан сертификат на имя портала внутр. доступа «internal.termit.local», **если** он был загружен по инструкции;
  - (если имеется) промежуточного сертификата, которым подписан сертификат на имя портала внутр. доступа «internal.termit.local», **если** он был загружен по инструкции;
  - служебного самоподписанного сертификата, скачиваемый по адресу https://FQDN_брокера:9090/ca.

Пример конфигурации для внешнего балансировщика в режиме SSL Bridging

В примере приведен файл конфигурации с типом балансировки Source IP Hash.

```
frontend termit_https
   mode http
   bind :443 ssl crt /etc/haproxy/external-lb-cert.pem
   http-request set-header X-Real-IP %[src]
```

```
    default_backend termit_servers_https

backend termit_servers_https
    option httpchk
    http-check connect port 9090 ssl
    http-check send meth GET  uri /_/broker-health
    mode http
    balance source
    hash-type consistent
    server %Имя_Термит_Брокера_1% %IP_Адрес_Термит_Брокера_1%:10443 ssl check ca-file
/etc/haproxy/ca-ext-broker-merged.pem verify required
    server %Имя_Термит_Брокера_2% %IP_Адрес_Термит_Брокера_2%:10443 ssl check ca-file
/etc/haproxy/ca-ext-broker-merged.pem verify required
```

Если адрес внутреннего балансировщика «ext-lb.termit.local», а адрес портала внутреннего доступа «external.termit.local», тогда:

- `/etc/haproxy/external-lb-cert.pem` - объединённый PEM-файл TLS-сертификата балансировщика HAProxy, содержащий:
  - сертификат на имя портала внутреннего доступа «external.termit.local».
  - закрытый ключ от сертификата портала внутреннего доступа «external.termit.local».
  - корневой сертификат, которым подписан сертификат портала внутреннего доступа «external.termit.local».
- `/etc/haproxy/ca-ext-broker-merged.pem` - файл, состоящий из:
  - корневого сертификата, которым подписан сертификат на имя портала внутреннего доступа «external.termit.local», **если** он был загружен по инструкции;
  - (если имеется) промежуточного сертификата, которым подписан сертификат на имя портала внутреннего доступа «external.termit.local», **если** он был загружен по инструкции;
  - служебные самоподписанные сертификаты, скачиваемые по адресу https://FQDN_брокера:9090/ca.

4. Проверьте созданный файл конфигурации с помощью команды ниже. Вывод команды не должен содержать ошибок. Предупреждения (warnings) не являются ошибками.

```
sudo haproxy -f /etc/haproxy/haproxy.cfg -c
```

5. Перезапустите службу с помощью команды:

```
sudo systemctl restart haproxy
```

6. Разрешите работу в SELinux с помощью команд:

   для внутреннего балансировщика

```
sudo semanage port -a -t http_port_t -p tcp 9090 || sudo semanage port -m -t http_port_t
-p tcp 9090
sudo semanage port -a -t http_port_t -p tcp 8443 || sudo semanage port -m -t http_port_t
-p tcp 8443
sudo chcon -t cert_t /etc/haproxy/cert.pem
sudo chcon -t cert_t /etc/haproxy/ca-int-broker-merged.pem
```

   для внешнего балансировщика

```
sudo semanage port -a -t http_port_t -p tcp 9090 || sudo semanage port -m -t http_port_t
-p tcp 9090
sudo semanage port -a -t http_port_t -p tcp 10443 || sudo semanage port -m -t http_port_t
-p tcp 10443
sudo chcon -t cert_t /etc/haproxy/external-lb-cert.pem
sudo chcon -t cert_t /etc/haproxy/ca-ext-broker-merged.pe
```

7. Проверьте доступность Termit по URL адресу, указанному при установке брокер серверов.

Astra Linux

Для установки используется последняя доступная в репозитории Astra Linux версия HAProxy. Подробнее о конфигурации — на официальном сайте Astra Linux.

Для корректной работы HAProxy на Astra Linux в режимах высокой или максимальной защищенности может понадобится отключение или дополнительная настройка ненулевых классификационных меток. Подробнее о настройке можно ознакомиться на официальном сайте Astra Linux.

1. Установите ПО HAProxy с помощью команды:

```
sudo apt install haproxy
```

2. Включите службу haproxy и добавьте автозагрузку с помощью команды:

```
sudo systemctl enable haproxy --now
```

3. Добавьте в конец файла конфигурации `/etc/haproxy/haproxy.cfg` настройку серверов для балансировки.

Пример конфигурации для внутреннего балансировщика в режиме SSL Bridging

В примере приведен файл конфигурации с типом балансировки Source IP Hash.

```
frontend termit_https
   mode http
   bind :80
   bind :443 ssl crt /etc/haproxy/cert.pem
   http-request redirect scheme https unless { ssl_fc }
   http-request set-header X-Real-IP %[src]
   default_backend termit_servers_https

backend termit_servers_https
   option httpchk
   http-check connect port 9090 ssl
   http-check send meth GET  uri /_/broker-health
   mode http
   balance source
   hash-type consistent
   server %Имя_Термит_Брокера_1% %IP_Адрес_Термит_Брокера_1%:443 ssl check ca-file
/etc/haproxy/ca-int-broker-merged.pem verify required
   server %Имя_Термит_Брокера_2% %IP_Адрес_Термит_Брокера_2%:443 ssl check ca-file
/etc/haproxy/ca-int-broker-merged.pem verify required

frontend termit_8443_https
   mode tcp
   bind :8443
   default_backend termit_servers_8443_https

backend termit_servers_8443_https
   option httpchk
   http-check connect port 9090 ssl
   http-check send meth GET  uri /_/broker-health
   mode tcp
   balance leastconn
   server %Имя_Термит_Брокера_1% %IP_Адрес_Термит_Брокера_1%:8443 check check-ssl verify
none
   server %Имя_Термит_Брокера_2% %IP_Адрес_Термит_Брокера_2%:8443 check check-ssl verify
none
```

Если адрес внутреннего балансировщика «internal.termit.local», а адрес портала внутреннего доступа «internal.termit.local», тогда:

- `/etc/haproxy/cert.pem` - объединённый PEM-файл TLS-сертификата балансировщика HAProxy, содержащий:

- сертификат на имя портала внутреннего доступа «internal.termit.local».
- закрытый ключ от сертификата портала внутреннего доступа «internal.termit.local».
- корневой сертификат, которым подписан сертификат портала внутреннего доступа «internal.termit.local».

◦ `/etc/haproxy/ca-int-broker-merged.pem` - файл, состоящий из:

- корневого сертификата, которым подписан сертификат на имя портала внутр. доступа «internal.termit.local», **если** он был загружен по инструкции;
- (если имеется) промежуточного сертификата, которым подписан сертификат на имя портала внутр. доступа «internal.termit.local», **если** он был загружен по инструкции;
- служебного самоподписанного сертификата, скачиваемый по адресу https://FQDN_брокера:9090/ca.

Пример конфигурации для внешнего балансировщика в режиме SSL Bridging

В примере приведен файл конфигурации с типом балансировки Source IP Hash.

```
frontend termit_https
   mode http
   bind :443 ssl crt /etc/haproxy/external-lb-cert.pem
   http-request set-header X-Real-IP %[src]
   default_backend termit_servers_https

backend termit_servers_https
   option httpchk
   http-check connect port 9090 ssl
   http-check send meth GET  uri /_/broker-health
   mode http
   balance source
   hash-type consistent
   server %Имя_Термит_Брокера_1% %IP_Адрес_Термит_Брокера_1%:10443 ssl check ca-file
/etc/haproxy/ca-ext-broker-merged.pem verify required
   server %Имя_Термит_Брокера_2% %IP_Адрес_Термит_Брокера_2%:10443 ssl check ca-file
/etc/haproxy/ca-ext-broker-merged.pem verify required
```

Если адрес внутреннего балансировщика «ext-lb.termit.local», а адрес портала внутреннего доступа «external.termit.local», тогда:

◦ `/etc/haproxy/external-lb-cert.pem` - объединённый PEM-файл TLS-сертификата балансировщика HAProxy, содержащий:

- сертификат на имя портала внутреннего доступа «external.termit.local».
- закрытый ключ от сертификата портала внутреннего доступа

«external.termit.local».
    - корневой сертификат, которым подписан сертификат портала внутреннего доступа «external.termit.local».
  - `/etc/haproxy/ca-ext-broker-merged.pem` - файл, состоящий из:
    - корневого сертификата, которым подписан сертификат на имя портала внутреннего доступа «external.termit.local», **если** он был загружен по инструкции;
    - (если имеется) промежуточного сертификата, которым подписан сертификат на имя портала внутреннего доступа «external.termit.local», **если** он был загружен по инструкции;
    - служебные самоподписанные сертификаты, скачиваемые по адресу https://FQDN_брокера:9090/ca.

4. Проверьте созданный файл конфигурации с помощью команды ниже. Вывод команды не должен содержать ошибок. Предупреждения (warnings) не являются ошибками.

```
sudo haproxy -f /etc/haproxy/haproxy.cfg -c
```

5. Перезапустите службу с помощью команды:

```
sudo systemctl restart haproxy
```

6. Проверьте доступность Termit по URL адресу, указанному при установке брокера.

Debian

1. Установите ПО HAProxy с помощью команды:

```
sudo systemctl install haproxy
```

2. Включите службу haproxy и добавьте автозагрузку с помощью команды:

```
sudo systemctl enable haproxy --now
```

3. Добавьте в конец файла конфигурации `/etc/haproxy/haproxy.cfg` настройку серверов для балансировки.

Пример конфигурации для внутреннего балансировщика в режиме SSL Bridging

В примере приведен файл конфигурации с типом балансировки Source IP Hash.

```
frontend termit_https
   mode http
   bind :80
   bind :443 ssl crt /etc/haproxy/cert.pem
   http-request redirect scheme https unless { ssl_fc }
   http-request set-header X-Real-IP %[src]
   default_backend termit_servers_https

backend termit_servers_https
   option httpchk
   http-check connect port 9090 ssl
   http-check send meth GET  uri /_/broker-health
   mode http
   balance source
   hash-type consistent
   server %Имя_Термит_Брокера_1% %IP_Адрес_Термит_Брокера_1%:443 ssl check ca-file
/etc/haproxy/ca-int-broker-merged.pem verify required
   server %Имя_Термит_Брокера_2% %IP_Адрес_Термит_Брокера_2%:443 ssl check ca-file
/etc/haproxy/ca-int-broker-merged.pem verify required

frontend termit_8443_https
   mode tcp
   bind :8443
   default_backend termit_servers_8443_https

backend termit_servers_8443_https
   option httpchk
   http-check connect port 9090 ssl
   http-check send meth GET  uri /_/broker-health
   mode tcp
   balance leastconn
   server %Имя_Термит_Брокера_1% %IP_Адрес_Термит_Брокера_1%:8443 check check-ssl verify
none
   server %Имя_Термит_Брокера_2% %IP_Адрес_Термит_Брокера_2%:8443 check check-ssl verify
none
```

Если адрес внутреннего балансировщика «internal.termit.local», а адрес портала внутреннего доступа «internal.termit.local», тогда:

- `/etc/haproxy/cert.pem` - объединённый PEM-файл TLS-сертификата балансировщика HAProxy, содержащий:
  - сертификат на имя портала внутреннего доступа «internal.termit.local».
  - закрытый ключ от сертификата портала внутреннего доступа «internal.termit.local».
  - корневой сертификат, которым подписан сертификат портала внутреннего доступа «internal.termit.local».
- `/etc/haproxy/ca-int-broker-merged.pem` - файл, состоящий из:

- корневого сертификата, которым подписан сертификат на имя портала внутр. доступа «internal.termit.local», **если** он был загружен по инструкции;
- (если имеется) промежуточного сертификата, которым подписан сертификат на имя портала внутр. доступа «internal.termit.local», **если** он был загружен по инструкции;
- служебного самоподписанного сертификата, скачиваемый по адресу https://FQDN_брокера:9090/ca.

Пример конфигурации для внешнего балансировщика в режиме SSL Bridging

В примере приведен файл конфигурации с типом балансировки Source IP Hash.

```
frontend termit_https
   mode http
   bind :443 ssl crt /etc/haproxy/external-lb-cert.pem
   http-request set-header X-Real-IP %[src]
   default_backend termit_servers_https

backend termit_servers_https
   option httpchk
   http-check connect port 9090 ssl
   http-check send meth GET  uri /_/broker-health
   mode http
   balance source
   hash-type consistent
   server %Имя_Термит_Брокера_1% %IP_Адрес_Термит_Брокера_1%:10443 ssl check ca-file
/etc/haproxy/ca-ext-broker-merged.pem verify required
   server %Имя_Термит_Брокера_2% %IP_Адрес_Термит_Брокера_2%:10443 ssl check ca-file
/etc/haproxy/ca-ext-broker-merged.pem verify required
```

Если адрес внутреннего балансировщика «ext-lb.termit.local», а адрес портала внутреннего доступа «external.termit.local», тогда:

- `/etc/haproxy/external-lb-cert.pem` - объединённый PEM-файл TLS-сертификата балансировщика HAProxy, содержащий:
  - сертификат на имя портала внутреннего доступа «external.termit.local».
  - закрытый ключ от сертификата портала внутреннего доступа «external.termit.local».
  - корневой сертификат, которым подписан сертификат портала внутреннего доступа «external.termit.local».
- `/etc/haproxy/ca-ext-broker-merged.pem` - файл, состоящий из:
  - корневого сертификата, которым подписан сертификат на имя портала

внутреннего доступа «external.termit.local», **если** он был загружен по инструкции;

- (если имеется) промежуточного сертификата, которым подписан сертификат на имя портала внутреннего доступа «external.termit.local», **если** он был загружен по инструкции;
- служебные самоподписанные сертификаты, скачиваемые по адресу https://FQDN_брокера:9090/ca.

4. Проверьте созданный файл конфигурации с помощью команды ниже. Вывод команды не должен содержать ошибок. Предупреждения (warnings) не являются ошибками.

```
sudo haproxy -f /etc/haproxy/haproxy.cfg -c
```

5. Перезапустите службу с помощью команды:

```
sudo systemctl restart haproxy
```

6. Проверьте доступность Termit по URL адресу, указанному при установке брокера.

ALT Linux

1. Установите ПО HAProxy с помощью команды:

```
sudo apt-get install haproxy
```

2. Включите службу haproxy и добавьте автозагрузку с помощью команды:

```
sudo systemctl enable haproxy --now
```

3. Добавьте в конец файла конфигурации `/etc/haproxy/haproxy.cfg` настройку серверов для балансировки.

Пример конфигурации для внутреннего балансировщика в режиме SSL Bridging

В примере приведен файл конфигурации с типом балансировки Source IP Hash.

```
frontend termit_https
   mode http
   bind :80
   bind :443 ssl crt /etc/haproxy/cert.pem
   http-request redirect scheme https unless { ssl_fc }
   http-request set-header X-Real-IP %[src]
```

```
   default_backend termit_servers_https

backend termit_servers_https
   option httpchk
   http-check connect port 9090 ssl
   http-check send meth GET  uri /_/broker-health
   mode http
   balance source
   hash-type consistent
   server %Имя_Термит_Брокера_1% %IP_Адрес_Термит_Брокера_1%:443 ssl check ca-file
/etc/haproxy/ca-int-broker-merged.pem verify required
   server %Имя_Термит_Брокера_2% %IP_Адрес_Термит_Брокера_2%:443 ssl check ca-file
/etc/haproxy/ca-int-broker-merged.pem verify required

frontend termit_8443_https
   mode tcp
   bind :8443
   default_backend termit_servers_8443_https

backend termit_servers_8443_https
   option httpchk
   http-check connect port 9090 ssl
   http-check send meth GET  uri /_/broker-health
   mode tcp
   balance leastconn
   server %Имя_Термит_Брокера_1% %IP_Адрес_Термит_Брокера_1%:8443 check check-ssl verify
none
   server %Имя_Термит_Брокера_2% %IP_Адрес_Термит_Брокера_2%:8443 check check-ssl verify
none
```

Если адрес внутреннего балансировщика «internal.termit.local», а адрес портала внутреннего доступа «internal.termit.local», тогда:

- `/etc/haproxy/cert.pem` - объединённый PEM-файл TLS-сертификата балансировщика HAProxy, содержащий:
  - сертификат на имя портала внутреннего доступа «internal.termit.local».
  - закрытый ключ от сертификата портала внутреннего доступа «internal.termit.local».
  - корневой сертификат, которым подписан сертификат портала внутреннего доступа «internal.termit.local».
- `/etc/haproxy/ca-int-broker-merged.pem` - файл, состоящий из:
  - корневого сертификата, которым подписан сертификат на имя портала внутр. доступа «internal.termit.local», **если** он был загружен по инструкции;
  - (если имеется) промежуточного сертификата, которым подписан сертификат на имя портала внутр. доступа «internal.termit.local», **если** он был загружен по инструкции;

- служебного самоподписанного сертификата, скачиваемый по адресу https://FQDN_брокера:9090/ca.

Пример конфигурации для внешнего балансировщика в режиме SSL Bridging

В примере приведен файл конфигурации с типом балансировки Source IP Hash.

```
frontend termit_https
   mode http
   bind :443 ssl crt /etc/haproxy/external-lb-cert.pem
   http-request set-header X-Real-IP %[src]
   default_backend termit_servers_https

backend termit_servers_https
   option httpchk
   http-check connect port 9090 ssl
   http-check send meth GET  uri /_/broker-health
   mode http
   balance source
   hash-type consistent
   server %Имя_Термит_Брокера_1% %IP_Адрес_Термит_Брокера_1%:10443 ssl check ca-file
/etc/haproxy/ca-ext-broker-merged.pem verify required
   server %Имя_Термит_Брокера_2% %IP_Адрес_Термит_Брокера_2%:10443 ssl check ca-file
/etc/haproxy/ca-ext-broker-merged.pem verify required
```

Если адрес внутреннего балансировщика «ext-lb.termit.local», а адрес портала внутреннего доступа «external.termit.local», тогда:

- `/etc/haproxy/external-lb-cert.pem` - объединённый PEM-файл TLS-сертификата балансировщика HAProxy, содержащий:
  - сертификат на имя портала внутреннего доступа «external.termit.local».
  - закрытый ключ от сертификата портала внутреннего доступа «external.termit.local».
  - корневой сертификат, которым подписан сертификат портала внутреннего доступа «external.termit.local».
- `/etc/haproxy/ca-ext-broker-merged.pem` - файл, состоящий из:
  - корневого сертификата, которым подписан сертификат на имя портала внутреннего доступа «external.termit.local», **если** он был загружен по инструкции;
  - (если имеется) промежуточного сертификата, которым подписан сертификат на имя портала внутреннего доступа «external.termit.local», **если** он был загружен по инструкции;
  - служебные самоподписанные сертификаты, скачиваемые по адресу

https://FQDN_брокера:9090/ca.

4. Проверьте созданный файл конфигурации с помощью команды ниже. Вывод команды не должен содержать ошибок. Предупреждения (warnings) не являются ошибками.

```
sudo haproxy -f /etc/haproxy/haproxy.cfg -c
```

5. Перезапустите службу с помощью команды:

```
sudo systemctl restart haproxy
```

6. Проверьте доступность Termit по URL адресу, указанному при установке брокера.

OpenSUSE

1. Установите ПО HAProxy с помощью команды:

```
sudo zypper install haproxy
```

2. Включите службу haproxy и добавьте автозагрузку с помощью команды:

```
sudo systemctl enable haproxy --now
```

3. Добавьте в конец файла конфигурации `/etc/haproxy/haproxy.cfg` настройку серверов для балансировки.

Пример конфигурации для внутреннего балансировщика в режиме SSL Bridging

В примере приведен файл конфигурации с типом балансировки Source IP Hash.

```
frontend termit_https
   mode http
   bind :80
   bind :443 ssl crt /etc/haproxy/cert.pem
   http-request redirect scheme https unless { ssl_fc }
   http-request set-header X-Real-IP %[src]
   default_backend termit_servers_https

backend termit_servers_https
   option httpchk
   http-check connect port 9090 ssl
   http-check send meth GET  uri /_/broker-health
   mode http
```

```
   balance source
   hash-type consistent
   server %Имя_Термит_Брокера_1% %IP_Адрес_Термит_Брокера_1%:443 ssl check ca-file
/etc/haproxy/ca-int-broker-merged.pem verify required
   server %Имя_Термит_Брокера_2% %IP_Адрес_Термит_Брокера_2%:443 ssl check ca-file
/etc/haproxy/ca-int-broker-merged.pem verify required

frontend termit_8443_https
   mode tcp
   bind :8443
   default_backend termit_servers_8443_https

backend termit_servers_8443_https
   option httpchk
   http-check connect port 9090 ssl
   http-check send meth GET  uri /_/broker-health
   mode tcp
   balance leastconn
   server %Имя_Термит_Брокера_1% %IP_Адрес_Термит_Брокера_1%:8443 check check-ssl verify
none
   server %Имя_Термит_Брокера_2% %IP_Адрес_Термит_Брокера_2%:8443 check check-ssl verify
none
```

Если адрес внутреннего балансировщика «internal.termit.local», а адрес портала внутреннего доступа «internal.termit.local», тогда:

- `/etc/haproxy/cert.pem` - объединённый PEM-файл TLS-сертификата балансировщика HAProxy, содержащий:
  - сертификат на имя портала внутреннего доступа «internal.termit.local».
  - закрытый ключ от сертификата портала внутреннего доступа «internal.termit.local».
  - корневой сертификат, которым подписан сертификат портала внутреннего доступа «internal.termit.local».
- `/etc/haproxy/ca-int-broker-merged.pem` - файл, состоящий из:
  - корневого сертификата, которым подписан сертификат на имя портала внутр. доступа «internal.termit.local», **если** он был загружен по инструкции;
  - (если имеется) промежуточного сертификата, которым подписан сертификат на имя портала внутр. доступа «internal.termit.local», **если** он был загружен по инструкции;
  - служебного самоподписанного сертификата, скачиваемый по адресу https://FQDN_брокера:9090/ca.

Пример конфигурации для внешнего балансировщика в режиме SSL Bridging

В примере приведен файл конфигурации с типом балансировки Source IP Hash.

```
frontend termit_https
   mode http
   bind :443 ssl crt /etc/haproxy/external-lb-cert.pem
   http-request set-header X-Real-IP %[src]
   default_backend termit_servers_https

backend termit_servers_https
   option httpchk
   http-check connect port 9090 ssl
   http-check send meth GET  uri /_/broker-health
   mode http
   balance source
   hash-type consistent
   server %Имя_Термит_Брокера_1% %IP_Адрес_Термит_Брокера_1%:10443 ssl check ca-file
/etc/haproxy/ca-ext-broker-merged.pem verify required
   server %Имя_Термит_Брокера_2% %IP_Адрес_Термит_Брокера_2%:10443 ssl check ca-file
/etc/haproxy/ca-ext-broker-merged.pem verify required
```

Если адрес внутреннего балансировщика «ext-lb.termit.local», а адрес портала внутреннего доступа «external.termit.local», тогда:

- `/etc/haproxy/external-lb-cert.pem` - объединённый PEM-файл TLS-сертификата балансировщика HAProxy, содержащий:
  - сертификат на имя портала внутреннего доступа «external.termit.local».
  - закрытый ключ от сертификата портала внутреннего доступа «external.termit.local».
  - корневой сертификат, которым подписан сертификат портала внутреннего доступа «external.termit.local».
- `/etc/haproxy/ca-ext-broker-merged.pem` - файл, состоящий из:
  - корневого сертификата, которым подписан сертификат на имя портала внутреннего доступа «external.termit.local», **если** он был загружен по инструкции;
  - (если имеется) промежуточного сертификата, которым подписан сертификат на имя портала внутреннего доступа «external.termit.local», **если** он был загружен по инструкции;
  - служебные самоподписанные сертификаты, скачиваемые по адресу https://FQDN_брокера:9090/ca.

4. Проверьте созданный файл конфигурации с помощью команды ниже. Вывод команды не должен содержать ошибок. Предупреждения (warnings) не являются ошибками.

```
sudo haproxy -f /etc/haproxy/haproxy.cfg -c
```

5. Перезапустите службу с помощью команды:

```
sudo systemctl restart haproxy
```

6. Проверьте доступность Termit по URL адресу, указанному при установке брокера.

Передача сертификата клиента для аутентификации по смарт-карте

Приведенные настройки передачи клиентского сертификата необходимы только при использовании аутентификации по смарт-карте на брокере или ТС. Если смарт-карты не применяются, изменять конфигурацию балансировщика нагрузки не требуется.

Для корректной работы аутентификации пользователей с помощью смарт-карт балансировщик нагрузки должен передавать клиентский сертификат брокеру. Это необходимо, чтобы брокер мог проверить подлинность пользователя и сопоставить его с учётной записью.

Настройки балансировщиков нагрузки для поддержки аутентификации по смарт-карте описаны в разделе.

4.2. Установка и настройка Keepalived

Keepalived — программный пакет с открытым исходным кодом, который обеспечивает высокую доступность (HA) и отказоустойчивость для систем на базе Linux. Он используется для:

- управления виртуальными IP-адресами (VIP);
- обеспечения бесперебойной доступности сервисов в сценариях, где могут произойти сбои системы или приложения.

В этом разделе описано, как установить и настроить Keepalived для операционных систем: РЕД ОС, Astra Linux, Debian, Alt Linux и OpenSUSE.

4.2.1. Установка Keepalived на серверах

Перед установкой проверьте, что Keepalived установлен на серверах:

- основном;

- резервном.

Его можно установить с помощью менеджера пакетов системы для:

РЕД ОС

```
sudo dnf install keepalived
```

Astra Linux

```
sudo apt install keepalived
```

Debian

```
sudo apt install keepalived
```

Alt Linux

```
sudo apt-get install keepalived
```

OpenSUSE

```
sudo zypper install keepalived
```

4.2.2. Настройка Keepalived на основном сервере

1. Отредактируйте файл конфигурации Keepalived, который по умолчанию находится по адресу `/etc/keepalived/keepalived.conf`, на основном сервере с помощью команды ниже:

   ```
   sudo nano /etc/keepalived/keepalived.conf
   ```

2. Добавьте следующую конфигурацию в файл:

   ```
   vrrp_instance VI_1 {

       state MASTER
       interface %Имя_сетевого_интерфейса%
       virtual_router_id %Уникальный_идентификатор_роутера_VRRP%
       priority %Приоритет_основного_сервера%
       advert_int %Интервал_обновления_состояния_сервера%
   ```

```
    authentication {
        auth_type PASS
        auth_pass %Пароль%
    }
    virtual_ipaddress {
        %Виртуальный_IP-адрес%
    }
}
```

Где:

- **%Имя_сетевого_интерфейса%** — имя сетевого интерфейса, например «eth0».
- **%Уникальный_идентификатор_роутера_VRRP%** — уникальный идентификатор роутера VRRP, например «51». Он должен совпадать на серверах.
- **%Приоритет_основного_сервера%** — приоритет основного сервера, например «100». Он должен быть выше, чем на резервном.
- **%Интервал_обновления_состояния_сервера%** — интервал обновления состояния сервера, например «1».
- **%Пароль%** — пароль, например «password123». Он должен быть одинаковым на серверах.
- **%Виртуальный_IP-адрес%** — VIP для службы, например «192.168.1.100».

3. Сохраните и закройте файл.
4. Чтобы применить изменения, перезапустите Keepalived с помощью команд:

```
sudo systemctl restart keepalived

sudo systemctl enable keepalived
```

4.2.3. Настройка Keepalived на резервном сервере

1. Отредактируйте файл конфигурации Keepalived, который по умолчанию находится по адресу `/etc/keepalived/keepalived.conf`, на резервном сервере с помощью команды ниже:

```
sudo nano /etc/keepalived/keepalived.conf
```

2. Добавьте следующую конфигурацию в файл:

```
vrrp_instance VI_1 {

    state BACKUP
```

```
    interface %Имя_сетевого_интерфейса%
    virtual_router_id %Уникальный_идентификатор_роутера_VRRP%
    priority %Приоритет_резервного_сервера%
    advert_int %Интервал_обновления_состояния_сервера%
    authentication {
        auth_type PASS
        auth_pass %Пароль%
    }
    virtual_ipaddress {
        %Виртуальный_IP-адрес%
    }
}
```

Где:

- **%Имя_сетевого_интерфейса%** — имя сетевого интерфейса, например «eth0».
- **%Уникальный_идентификатор_роутера_VRRP%** — уникальный идентификатор роутера VRRP, например «51». Он должен совпадать на серверах.
- **%Приоритет_резервного_сервера%** — приоритет резервного сервера, например «90». Он должен быть ниже, чем на основном.
- **%Интервал_обновления_состояния_сервера%** — интервал обновления состояния сервера, например «1».
- **%Пароль%** — пароль, например «password123». Он должен быть одинаковым на серверах.
- **%Виртуальный_IP-адрес%** — VIP для службы, например «192.168.1.100».

3. Сохраните и закройте файл.
4. Чтобы применить изменения, перезапустите Keepalived с помощью команд:

```
sudo systemctl restart keepalived

sudo systemctl enable keepalived
```

4.2.4. Проверка работоспособности

1. Проверьте, что Keepalived работает на серверах с помощью команды:

```
sudo systemctl status keepalived
```

2. Проверьте, какой из серверов обрабатывает VIP.

```
ip a | grep %Виртуальный_IP-адрес%
```

На основном сервере этот IP-адрес должен быть привязан к сетевому интерфейсу.

3. Для тестирования отказоустойчивости отключите сетевой интерфейс на основном сервере с помощью команды:

```
sudo ifdown %Имя_сетевого_интерфейса%
```

4. После отключения основного сервера VIP должен автоматически перейти на резервный сервер. Это можно проверить с помощью команды:

```
ip a | grep %Виртуальный_IP-адрес%
```

5. Восстановите работу интерфейса на основном сервере:

```
sudo ifup %Имя_сетевого_интерфейса%
```

Глава 5. Брокеры

В этом разделе описаны шаги по установке и первичной настройке сервера с ролью брокер, входящего в состав Termit.

1. Не поддерживается развертывание БД и брокера на одном сервере.
2. Подсеть инфраструктуры не должна совпадать с сетью Docker (по умолчанию - 172.17.0.0/16). При совпадении — измените ее, подробнее в статье Ошибки при пересечении IP-адресов между Docker и хост-системой.

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. Скопируйте дистрибутив на сервер.
4. Распакуйте архив с помощью команды:

```
unzip termit-2.*.*-*.zip
```

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

```
sudo chmod +x ./broker-installer
```

6. Запустите скрипт:

```
sudo ./broker-installer install
```

7. Введите пароль администратора системы.
8. Укажите имя узла брокера. Оно должно соответствовать FQDN-адресу брокера.
9. Появится сообщение:

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

   Укажите «1».

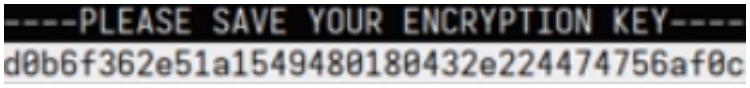
10. Введите адрес портала внутреннего доступа (например, «internal.termit.local»), по нему будут доступны сервисы Termit. DNS-запись на имя портала внутреннего доступа должна: в инсталляции «Standalone» указывать на IP-адрес брокера; в инсталляции «High Availability» указывать на IP-адрес внутреннего балансировщика.
11. Появится сообщение:

```
Do you want to connect to an existing database or create a new one? (1/2)

1. Connect to an existing database
2. Create new database

Enter 1 or 2?
```

    a. Если вы хотите подключиться к существующей базе данных (БД), то:
        i. Укажите «1».
        ii. Укажите имя хоста БД, например «db.example.com».
        iii. Укажите порт «5432».
        iv. Укажите имя БД, например «example».
        v. Укажите имя пользователя БД, например «termit».
        vi. Введите пароль для БД.
    b. Если вы хотите создать новую базу данных (БД), то:
        i. Укажите «2».
        ii. Укажите имя хоста БД, например «db.example.com».
        iii. Укажите порт «5432».
        iv. Укажите имя новой БД.
        v. Укажите имя пользователя БД, например «termit».

vi. Введите пароль для БД.

Установка брокера занимает около двух минут. Во время запуска отображается ключ шифрования БД, который необходимо сохранить. Этот ключ нужен для установки дополнительных брокеров.



Для подтверждения успешности инсталляции, перейдите на страницу портала внутреннего доступа (например, «internal.termit.local») и проверьте, что он открывается, и отображается страница аутентификации. В появившемся окне аутентификации укажите логин «tadm» и пароль «admin» от учетной записи по умолчанию.

- Если брокер установился с ошибкой, то проверьте логи в контейнерах: «config-service» и «broker-client» с помощью команды:

```
sudo docker logs %ID_Контейнера%
```

- В случае переустановки брокера выполните шаги по деинсталляции.

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):

1. Для установки компонентов необходимы настроенные репозитории.
2. Выбор команды для установки Docker Compose зависит от версии операционной системы.

1. Установите Docker и Docker Compose:
   - Для Astra Linux 1.7 выполните команду:

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

Пакет `docker-compose` в Astra Linux 1.7 является устаревшим. В связи с этим могут возникать различия в разделителях в названиях контейнеров (комбинация символов _ и -).

- Для Astra Linux 1.8:

  a. выполните команду:

  ```
  sudo apt install docker.io docker-compose-v2
  ```

  Подробнее об установке Docker Compose v2 в Astra Linux 1.8 можно прочесть на официальном сайте Astra Linux,.

  b. добавьте в конфигурационный файл `/etc/default/grub` в параметр `GRUB_CMDLINE_LINUX_DEFAULT` запись:

  ```
  parsec.execstack=1
  ```

  c. обновите конфигурацию GRUB и выполните перезагрузку системы:

  ```
  sudo update-grub
  sudo reboot
  ```

2. Скопируйте дистрибутив на сервер.
3. Распакуйте архив с помощью команды:

```
unzip termit-2.*.*-*.zip
```

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

```
sudo chmod +x ./broker-installer
```

5. Запустите скрипт:

```
sudo ./broker-installer install
```

6. Введите пароль администратора системы.
7. Укажите имя узла брокера. Оно должно соответствовать FQDN-адресу брокера.
8. Появится сообщение:

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

```
Enter 1 or 2»
```

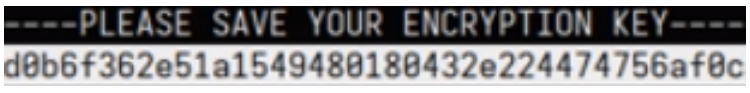
Укажите «1».

9. Введите адрес портала внутреннего доступа (например, «internal.termit.local»), по нему будут доступны сервисы Termit. DNS-запись на имя портала внутреннего доступа должна: в инсталляции «Standalone» указывать на IP-адрес брокера; в инсталляции «High Availability» указывать на IP-адрес внутреннего балансировщика.
10. Появится сообщение:

```
Do you want to connect to an existing database or create a new one? (1/2)

1. Connect to an existing database
2. Create new database

Enter 1 or 2?
```

   a. Если вы хотите подключиться к существующей базе данных (БД), то:
      i. Укажите «1».
      ii. Укажите имя хоста БД, например «db.example.com».
      iii. Укажите порт «5432».
      iv. Укажите имя БД, например «example».
      v. Укажите имя пользователя БД, например «termit».
      vi. Введите пароль для БД.
   b. Если вы хотите создать новую базу данных (БД), то:
      i. Укажите «2».
      ii. Укажите имя хоста БД, например «db.example.com».
      iii. Укажите порт «5432».
      iv. Укажите имя новой БД.
      v. Укажите имя пользователя БД, например «termit».
      vi. Введите пароль для БД.

Установка брокера занимает около двух минут. Во время запуска отображается ключ шифрования БД, который необходимо сохранить. Этот ключ нужен для установки дополнительных брокеров.



Для подтверждения успешности инсталляции, перейдите на страницу портала внутреннего доступа (например, «internal.termit.local») и проверьте, что он открывается, и отображается страница аутентификации. В появившемся окне аутентификации укажите логин «tadm» и пароль «admin» от учетной записи по умолчанию.

- Если брокер установился с ошибкой, то проверьте логи в контейнерах: «config-service» и «broker-client» с помощью команды:

```
sudo docker logs %ID_Контейнера%
```

- В случае переустановки брокера выполните шаги по деинсталляции.
- Подробнее о настройках Docker в операционной системе Astra Linux можно прочесть на официальном сайте Astra Linux.

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
```

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

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

3. Распакуйте архив с помощью команды:

```
unzip ./termit-2.1*.*-*.zip
```

4. Запустите установку с помощью команды:

```
sudo ./broker-installer install
```

5. Введите пароль администратора системы.
6. Укажите имя узла брокера. Оно должно соответствовать FQDN-адресу брокера.
7. Появится сообщение:

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

   Укажите «1».

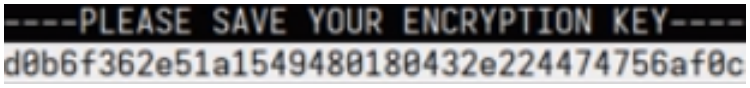
8. Введите адрес портала внутреннего доступа (например, «internal.termit.local»), по нему будут доступны сервисы Termit. DNS-запись на имя портала внутреннего доступа должна: в инсталляции «Standalone» указывать на IP-адрес брокера; в инсталляции «High Availability» указывать на IP-адрес внутреннего балансировщика.
9. Появится сообщение:

```
Do you want to connect to an existing database or create a new one? (1/2)

1. Connect to an existing database
2. Create new database

Enter 1 or 2?
```

   a. Если вы хотите подключиться к существующей базе данных (БД), то:
      i. Укажите «1».
      ii. Укажите имя хоста БД, например «db.example.com».
      iii. Укажите порт «5432».
      iv. Укажите имя БД, например «example».
      v. Укажите имя пользователя БД, например «termit».
      vi. Введите пароль для БД.
   b. Если вы хотите создать новую базу данных (БД), то:
      i. Укажите «2».
      ii. Укажите имя хоста БД, например «db.example.com».
      iii. Укажите порт «5432».
      iv. Укажите имя новой БД.
      v. Укажите имя пользователя БД, например «termit».

vi. Введите пароль для БД.

Установка брокера занимает около двух минут. Во время запуска отображается ключ шифрования БД, который необходимо сохранить. Этот ключ нужен для установки дополнительных брокеров.



Для подтверждения успешности инсталляции, перейдите на страницу портала внутреннего доступа (например, «internal.termit.local») и проверьте, что он открывается, и отображается страница аутентификации. В появившемся окне аутентификации укажите логин «tadm» и пароль «admin» от учетной записи по умолчанию.

- Если брокер установился с ошибкой, то проверьте логи в контейнерах: «config-service» и «broker-client» с помощью команды:

```
sudo docker logs %ID_Контейнера%
```

- В случае переустановки брокера выполните шаги по деинсталляции.

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. Распакуйте архив с помощью команды:

```
unzip termit-2.*.*-*.zip
```

4. Запустите установку с помощью команды:

```
sudo ./broker-installer install
```

5. Введите пароль администратора системы.
6. Укажите имя узла брокера. Оно должно соответствовать FQDN-адресу брокера.
7. Появится сообщение:

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

   Укажите «1».

8. Введите адрес портала внутреннего доступа (например, «internal.termit.local»), по нему будут доступны сервисы Termit. DNS-запись на имя портала внутреннего доступа должна: в инсталляции «Standalone» указывать на IP-адрес брокера; в инсталляции «High Availability» указывать на IP-адрес внутреннего балансировщика.
9. Появится сообщение:

```
Do you want to connect to an existing database or create a new one? (1/2)

1. Connect to an existing database
2. Create new database

Enter 1 or 2?
```

   a. Если вы хотите подключиться к существующей базе данных (БД), то:
      i. Укажите «1».
      ii. Укажите имя хоста БД, например «db.example.com».
      iii. Укажите порт «5432».
      iv. Укажите имя БД, например «example».
      v. Укажите имя пользователя БД, например «termit».
      vi. Введите пароль для БД.
   b. Если вы хотите создать новую базу данных (БД), то:
      i. Укажите «2».
      ii. Укажите имя хоста БД, например «db.example.com».

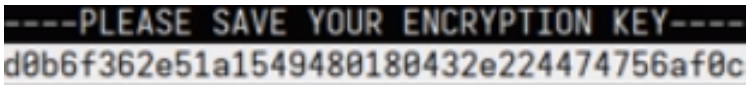
iii. Укажите порт «5432».

iv. Укажите имя новой БД.

v. Укажите имя пользователя БД, например «termit».

vi. Введите пароль для БД.

Установка брокера занимает около двух минут. Во время запуска отображается ключ шифрования БД, который необходимо сохранить. Этот ключ нужен для установки дополнительных брокеров.



Для подтверждения успешности инсталляции, перейдите на страницу портала внутреннего доступа (например, «internal.termit.local») и проверьте, что он открывается, и отображается страница аутентификации. В появившемся окне аутентификации укажите логин «tadm» и пароль «admin» от учетной записи по умолчанию.

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:

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

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

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

3. Скопируйте дистрибутив на сервер.
4. Распакуйте архив с помощью команды:

```
unzip ./termit-2.1*.*-*.zip
```

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

```
sudo chmod +x ./broker-installer
```

6. Запустите скрипт:

```
sudo ./broker-installer install
```

7. Введите пароль администратора системы.
8. Укажите имя узла брокера. Оно должно соответствовать FQDN-адресу брокера.
9. Появится сообщение:

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

   Укажите «1».

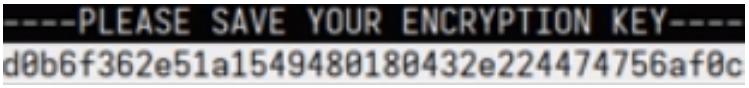
10. Введите адрес портала внутреннего доступа (например, «internal.termit.local»), по нему будут доступны сервисы Termit. DNS-запись на имя портала внутреннего доступа должна: в инсталляции «Standalone» указывать на IP-адрес брокера; в инсталляции «High Availability» указывать на IP-адрес внутреннего балансировщика.
11. Появится сообщение:

```
Do you want to connect to an existing database or create a new one? (1/2)

1. Connect to an existing database
2. Create new database

Enter 1 or 2?
```

    a. Если вы хотите подключиться к существующей базе данных (БД), то:
        i. Укажите «1».
        ii. Укажите имя хоста БД, например «db.example.com».
        iii. Укажите порт «5432».
        iv. Укажите имя БД, например «example».
        v. Укажите имя пользователя БД, например «termit».
        vi. Введите пароль для БД.
    b. Если вы хотите создать новую базу данных (БД), то:

i. Укажите «2».
ii. Укажите имя хоста БД, например «db.example.com».
iii. Укажите порт «5432».
iv. Укажите имя новой БД.
v. Укажите имя пользователя БД, например «termit».
vi. Введите пароль для БД.

Установка брокера занимает около двух минут. Во время запуска отображается ключ шифрования БД, который необходимо сохранить. Этот ключ нужен для установки дополнительных брокеров.



Для подтверждения успешности инсталляции, перейдите на страницу портала внутреннего доступа (например, «internal.termit.local») и проверьте, что он открывается, и отображается страница аутентификации. В появившемся окне аутентификации укажите логин «tadm» и пароль «admin» от учетной записи по умолчанию.

- Если брокер установился с ошибкой, то проверьте логи в контейнерах: «config-service» и «broker-client» с помощью команды:

```
sudo docker logs %ID_Контейнера%
```

- В случае переустановки брокера выполните шаги по деинсталляции.

Глава 6. Дополнительный брокер

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 ./broker-installer
```

4. Запустите скрипт:

```
sudo ./broker-installer install
```

5. Введите пароль администратора системы.

6. Укажите имя узла брокера. Оно должно соответствовать FQDN-адресу брокера.
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:

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

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

```
sudo systemctl start docker
```

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

```
sudo chmod +x ./broker-installer
```

4. Запустите скрипт:

```
sudo ./broker-installer install
```

5. Введите пароль администратора системы.
6. Укажите имя узла брокера. Оно должно соответствовать FQDN-адресу брокера.
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 ./broker-installer
```

4. Запустите скрипт:

```
sudo ./broker-installer install
```

5. Введите пароль администратора системы.
6. Укажите имя узла брокера. Оно должно соответствовать FQDN-адресу брокера.
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:

```
sudo chmod +x ./broker-installer
```

4. Запустите скрипт:

```
sudo ./broker-installer install
```

5. Введите пароль администратора системы.
6. Укажите имя узла брокера. Оно должно соответствовать FQDN-адресу брокера.

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:

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

2. Запустите службу с помощью команды:

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

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

```
sudo chmod +x ./broker-installer
```

4. Запустите скрипт:

```
sudo ./broker-installer install
```

5. Введите пароль администратора системы.
6. Укажите имя узла брокера. Оно должно соответствовать FQDN-адресу брокера.
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».

Глава 7. Дополнительное ПО

Установите дополнительное ПО перед установкой агента на ТС.

РЕД ОС

Проверьте, что все ТС, созданные базовым способом, введены в домен. Подробнее в статье Ввод РЕД ОС в домен.

1. В файле конфигурации `sshd (/etc/ssh/sshd_config)`, в параметре `X11Forwarding`, укажите значение «yes».
2. Чтобы установить X2Go-сервер и сопутствующие компоненты на ТС, выполните команду:

```
sudo dnf install x2goserver-xsession x2goserver-fmbindings x2goserver-common x2goserver
x2goagent x2goserver-printing cups-x2go xrdp openbox
```

3. Чтобы избежать ошибки при монтировании сетевых дисков и папок, установите пакеты `cifs-utils` и `autofs` с помощью команды:

```
sudo dnf install cifs-utils autofs
```

4. Для корректного перенаправления смарт-карт, установите утилиту `socat` с помощью команды:

```
sudo dnf install socat
```

5. Для работы со смарт-картами:
   a. установите утилиту `pcsc-tools`:

```
sudo dnf install pcsc-tools
```

   b. запустите службу с помощью команды:

```
sudo systemctl enable pcscd.service --now
```

6. Установите пакет `patch`, необходимый для переноса изменений между разными версиями текстовых файлов с помощью команды:

```
sudo dnf install patch
```

7. Настройте сетевой доступ к локальным хранилищам, установив пакет NFS-хранилища:

```
sudo dnf install nfs-utils nfs4-acl-tools
```

8. Для поддержки теневых подключений Установите пакеты, необходимые для работы теневых подключений, с помощью команды:

```
sudo dnf install x2goserver-desktopsharing qt5-qtsvg qt5-qtbase qt5-qtbase-gui
```

   Для автозапуска сервиса теневых сессий также выполните следующие команды:

```
sudo cp /usr/share/applications/x2godesktopsharing.desktop /etc/xdg/autostart/

sudo sed -i 's|^Exec=/usr/bin/x2godesktopsharing$|& --activate-desktop-sharing|'
/etc/xdg/autostart/x2godesktopsharing.desktop
```

9. Для подключения через веб-клиент Настройте `xrdp`:
   a. Убедитесь, что в файле `/etc/xrdp/xrdp.ini` в секции `Session types` блок настроек `[Xorg]` расположен первым. Если это не так — переместите его в начало секции.
   b. Отредактируйте файл `/etc/xrdp/sesman.ini`:
      i. Убедитесь, что параметр `Policy` имеет значение `Default`.
      ii. Убедитесь, что параметры `TerminalServerUsers` и `TerminalServerAdmins` имеют значение `all`. В противном случае придется добавлять каждого пользователя в группы `tsusers` или `tsadmins`.
      iii. Убедитесь, что параметр `AllowAlternateShell` имеет значение `yes`.
   c. Запустите службу `xrdp` с помощью команды:

```
sudo systemctl enable xrdp --now
```

   d. Выполните перезагрузку системы.

Перезагрузка рекомендуется для корректного применения настроек. В некоторых случаях без перезагрузки могут возникать проблемы с подключением.

10. Установите агент, используя скрипт, полученный при создании серверов в каталогах ресурсов формата **ТС базовым способом**.

Astra Linux

Проверьте, что все ТС, созданные базовым способом, введены в домен. Подробнее в статье Ввод Astra Linux в домен.

Далее проверьте конфигурацию файла `sssd.conf`, находящегося по пути `/etc/sssd/sssd.conf`, и установите значение `False` в следующей строке:

```
use_fully_qualified_names = False
```

1. В файле конфигурации `sshd (/etc/ssh/sshd_config)`, в параметре `X11Forwarding`, укажите значение «yes».
2. В файле конфигурации `/etc/systemd/logind.conf`, в параметре `KillUserProcesses` задайте значение «no» для корректной работы переподключения сессии и перезагрузите ТС.
3. Чтобы установить X2Go-сервер и сопутствующие компоненты на ТС, выполните команду:

```
sudo apt install x2goserver-xsession x2goserver-fmbindings x2goserver-common x2goserver-x2goagent x2goserver-printing cups-x2go xrdp openbox
```

4. Чтобы избежать ошибки при монтировании сетевых дисков и папок, установите пакеты `cifs-utils` и `autofs` с помощью команды:

```
sudo apt install cifs-utils autofs
```

5. Для корректного перенаправления смарт-карт, установите утилиту `socat` с помощью команды:

```
sudo apt install socat
```

6. Для работы со смарт-картами:
   a. установите утилиты `pcscd` и `pcsc-tools` с помощью команды:

```
sudo apt install pcscd pcsc-tools
```

b. запустите службу с помощью команды:

```
sudo systemctl enable pcscd.service --now
```

7. Установите пакет `patch`, необходимый для переноса изменений между разными версиями текстовых файлов с помощью команды:

```
sudo apt install patch
```

8. Для поддержки теневых подключений Установите пакеты, необходимые для работы теневых подключений, с помощью команды:

```
sudo apt install x2goserver-desktopsharing libqt5svg5 libqt5core5a libqt5gui5
libqt5widgets5 libqt5network5
```

Во время установки система предложит настроить подключение к теневым сессиям. Рекомендуемые настройки:

- `Create x2godesktopsharing group? [да/нет]` — нет,
- `Use existing group for X2Go Desktop Sharing? [да/нет]` — нет,
- `Auto-start X2Go Desktop Sharing applet? [да/нет]` — да,
- `Auto-activate X2Go Desktop Sharing? [да/нет]` — да.

9. Для подключения через веб-клиент Настройте `xrdp`:
   a. Убедитесь, что в файле `/etc/xrdp/xrdp.ini` в секции `Session types` блок настроек `[Xorg]` расположен первым. Если это не так — переместите его в начало секции.
   b. Отредактируйте файл `/etc/xrdp/sesman.ini`:
      i. Убедитесь, что параметр `Policy` имеет значение `Default`.
      ii. Убедитесь, что параметры `TerminalServerUsers` и `TerminalServerAdmins` имеют значение `all`. В противном случае придется добавлять каждого пользователя в группы `tsusers` или `tsadmins`.
      iii. Убедитесь, что параметр `AllowAlternateShell` имеет значение `yes`.
   c. Запустите службу `xrdp` с помощью команды:

```
sudo systemctl enable xrdp --now
```

d. Выполните перезагрузку системы.

Перезагрузка рекомендуется для корректного применения настроек. В некоторых случаях без перезагрузки могут возникать проблемы с подключением.

10. Для протокола Pulsar Для версии 1.8 Создайте на целевой машине скрипт pulsar-astra18-desktop-mofidy.sh, который дописывает изменения в целевой файл (проверку на сессию pulsar), со следующим содержимым:

Содержимое скрипта

```
#!/usr/bin/env bash

set -euo pipefail

usage() {
    cat <<EOF
Usage: $0 <file>

Finds an if condition containing a comparison with "$XDG_SESSION_ID"
inside [ ... ] or [[ ... ]], then inserts the specified code after
the corresponding "fi".

Options:
    -h, --help    Show this help

Example:
    $0 /path/to/script.sh
EOF
}

# ------------------------------------------------------------
# Проверка аргументов
# ------------------------------------------------------------

if [[ $# -eq 0 ]]; then
    echo "Error: file argument is required." >&2
    echo "Use '$0 --help' for help." >&2
    exit 1
fi

if [[ "$1" == "-h" || "$1" == "--help" ]]; then
    usage
    exit 0
fi

if [[ $# -ne 1 ]]; then
    echo "Error: expected exactly one argument." >&2
```

```
    echo "Use '$0 --help' for help." >&2
    exit 1
fi

FILE="$1"

if [[ ! -f "$FILE" ]]; then
    echo "Error: file does not exist or is not a regular file: $FILE" >&2
    exit 1
fi

read -r -d '' INSERT_CODE <<'EOF' || true

if [[ "$(loginctl show-session "$sess" -p Service --value 2>/dev/null)" == "pulsar" ]];
then
  continue
fi
EOF

# ------------------------------------------------------------

BACKUP="${FILE}.bak"
TMP="${FILE}.tmp.$$"

trap 'rm -f -- "$TMP"' EXIT

# Создаём backup
cp -- "$FILE" "$BACKUP"

awk -v code="$INSERT_CODE" '
{
    # Убираем ведущие пробелы для проверки комментариев
    line = $0
    sub(/^[[:space:]]*/, "", line)

    # Полностью игнорируем строки-комментарии
    if (line ~ /^#/) {
        print
        next
    }

    # Ищем if с [ ... ] или [[ ... ]]
    # и сравнением с "$XDG_SESSION_ID".
    #
    # Подойдут:
    #   if [ "$sess" == "$XDG_SESSION_ID" ]; then
    #   if [[ "$sess" == "$XDG_SESSION_ID" ]]; then
    #
    if (line ~ /^if[[:space:]]+\[/ &&
        line ~ /==[[:space:]]*"\$XDG_SESSION_ID"/ &&
        line ~ /\]/ &&
        line ~ /then/) {
```

```
        # Запоминаем отступ найденного if
        match($0, /^[[:space:]]*/)
        if_indent = substr($0, RSTART, RLENGTH)

        found = 1
        print
        next
    }

    # Ищем закрывающий fi
    if (found == 1 && line ~ /^fi[[:space:]]*;/) {
        print

        n = split(code, lines, "\n")
        for (i = 1; i <= n; i++)
            print if_indent lines[i]

        found = 2
        next
    }

    if (found == 1 && line == "fi") {
        print

        n = split(code, lines, "\n")
        for (i = 1; i <= n; i++)
            print if_indent lines[i]

        found = 2
        next
    }

    print
}

END {
    if (found == 0) {
        print "Error: condition containing == \"\$XDG_SESSION_ID\" was not found." >
"/dev/stderr"
        exit 1
    }

    if (found == 1) {
        print "Error: matching fi was not found." > "/dev/stderr"
        exit 1
    }
}
' "$FILE" > "$TMP"

# Сохраняем исходные права, включая executable bit
chmod --reference="$FILE" "$TMP"

# Заменяем оригинал
```

```
mv -- "$TMP" "$FILE"

trap - EXIT

echo "Done: $FILE"
echo "Backup: $BACKUP"
```

Затем выполните команду:

```
sudo ./pulsar-astra18-desktop-mofidy.sh /usr/bin/fly-check-switch-sess.sh
```

Скрипт создает резервную копию целевого файла и требует для выполнения права суперпользователя.

11. Установите агент, используя скрипт, полученный при создании серверов в каталогах ресурсов формата **ТС базовым способом**.

ALT Linux

Проверьте, что все ТС, созданные базовым способом, введены в домен. Подробнее в статье Ввод ALT Linux в домен.

1. В файле конфигурации `sshd (/etc/openssh/sshd_config)`, в параметре `X11Forwarding`, укажите значение «yes».
2. В файле конфигурации `/etc/systemd/logind.conf`, в параметре `KillUserProcesses` задайте значение «no» для корректной работы переподключения сессии и перезагрузите ТС.
3. Чтобы установить X2Go-сервер и сопутствующие компоненты на ТС, выполните команду:

```
sudo apt-get install x2goserver-xsession x2goserver-fmbindings x2goserver-x2goagent
x2goserver-printing cups-x2go xrdp openbox
```

4. Добавьте X2Go в автозапуск с помощью команды:

```
sudo systemctl enable x2goserver --now
```

5. Чтобы избежать ошибки при монтировании сетевых дисков и папок, установите пакеты `cifs-utils` и `autofs` с помощью команды:

```
sudo apt-get install cifs-utils autofs
```

6. Для корректного перенаправления смарт-карт, установите утилиту `socat` с помощью команды:

```
sudo apt-get install socat
```

7. Для работы со смарт-картами:
   a. установите утилиты `pcsc-tools` и `pcsc-lite-ccid`:

```
sudo apt-get install pcsc-tools pcsc-lite-ccid
```

   b. запустите службу с помощью команды:

```
sudo systemctl enable pcscd.service --now
```

8. Установите пакет `patch`, необходимый для переноса изменений между разными версиями текстовых файлов с помощью команды:

```
sudo apt-get install patch
```

9. Установите пакет `libqt5widgets5`, необходимый для запуска программ, которые были написаны с использованием графических элементов Qt 5 с помощью команды:

```
sudo apt-get install libqt5widgets5
```

10. Для поддержки теневых подключений Установите пакеты, необходимые для работы теневых подключений, с помощью команды:

```
sudo apt install x2goserver-desktopsharing libqt5svg5 libqt5core5a libqt5gui5
libqt5widgets5 libqt5network5
```

11. Для подключения через веб-клиент Настройте `xrdp`:
    a. Убедитесь, что в файле `/etc/xrdp/xrdp.ini` в секции `Session types` блок настроек `[Xorg]` расположен первым. Если это не так — переместите его в начало секции.
    b. Отредактируйте файл `/etc/xrdp/sesman.ini`:

i. Убедитесь, что параметр `Policy` имеет значение `Default`.

ii. Убедитесь, что параметры `TerminalServerUsers` и `TerminalServerAdmins` имеют значение `all`. В противном случае придется добавлять каждого пользователя в группы `tsusers` или `tsadmins`.

iii. Убедитесь, что параметр `AllowAlternateShell` имеет значение `yes`.

c. Запустите службу `xrdp` с помощью команды:

```
sudo systemctl enable xrdp --now
```

d. Выполните перезагрузку системы.

Перезагрузка рекомендуется для корректного применения настроек. В некоторых случаях без перезагрузки могут возникать проблемы с подключением.

12. Установите агент, используя скрипт, полученный при создании серверов в каталогах ресурсов формата **ТС базовым способом**.

Debian

Проверьте, что все ТС, созданные базовым способом, введены в домен. Подробнее в статье Ввод Debian в домен.

1. В файле конфигурации `sshd (/etc/ssh/sshd_config)`, в параметре `X11Forwarding`, укажите значение «yes».
2. Чтобы установить X2Go-сервер и сопутствующие компоненты на ТС, выполните команду:

```
sudo apt install x2goserver-xsession x2goserver-fmbindings x2goserver-common x2goserver-x2goagent x2goserver-printing cups-x2go xrdp openbox
```

3. Чтобы избежать ошибки при монтировании сетевых дисков, установите пакет `cifs-utils` и `autofs` с помощью команды:

```
sudo apt install cifs-utils autofs
```

4. Для корректного перенаправления смарт-карт, установите утилиту `socat` с помощью команды:

```
sudo apt install socat
```

5. Для работы со смарт-картами:
   a. установите утилиты `pcsc-tools` и `pcsc-lite-ccid`:

      ```
      sudo apt install pcsc-tools pcsc-lite-ccid
      ```

   b. запустите службу с помощью команды:

      ```
      sudo systemctl enable pcscd.service --now
      ```

6. Установите пакет `patch`, необходимый для переноса изменений между разными версиями текстовых файлов с помощью команды:

   ```
   sudo apt install patch
   ```

7. Установите пакет `libqt5widgets5`, необходимый для запуска программ, которые были написаны с использованием графических элементов Qt 5 с помощью команды:

   ```
   sudo apt install libqt5widgets5
   ```

8. Для поддержки теневых подключений Установите пакеты, необходимые для работы теневых подключений, с помощью команды:

   ```
   sudo apt-get install x2goserver-desktopsharing libqt5-svg libqt5-core libqt5-gui libqt5-
   widgets libqt5-network libqt5-xcbqpa
   ```

   Во время установки система предложит настроить теневые сессии. Рекомендуемые настройки:

   - `Create x2godesktopsharing group? [да/нет]` — n,
   - `Use existing group for X2Go Desktop Sharing? [да/нет]` — n,
   - `Auto-start X2Go Desktop Sharing applet? [да/нет]` — y,
   - `Auto-activate X2Go Desktop Sharing? [да/нет]` — y.

   Если в дальнейшем необходимо изменить данные настройки, запустите `sudo dpkg-reconfigure x2goserver-desktopsharing`.

9. Для подключения через веб-клиент Настройте `xrdp`:
   a. Убедитесь, что в файле `/etc/xrdp/xrdp.ini` в секции `Session types` блок настроек `[Xorg]` расположен первым. Если это не так — переместите его в начало секции.
   b. Отредактируйте файл `/etc/xrdp/sesman.ini`:
      i. Убедитесь, что параметр `Policy` имеет значение `Default`.
      ii. Убедитесь, что параметры `TerminalServerUsers` и `TerminalServerAdmins` имеют значение `all`. В противном случае придется добавлять каждого пользователя в группы `tsusers` или `tsadmins`.
      iii. Убедитесь, что параметр `AllowAlternateShell` имеет значение `yes`.
   c. Запустите службу `xrdp` с помощью команды:

      ```
      sudo systemctl enable xrdp --now
      ```

   d. Выполните перезагрузку системы.

Перезагрузка рекомендуется для корректного применения настроек. В некоторых случаях без перезагрузки могут возникать проблемы с подключением.

10. Установите агент, используя скрипт, полученный при создании серверов в каталогах ресурсов формата **ТС базовым способом**.

OpenSUSE

В этой инструкции приведен пример установки пакета для X2Go, скачанного по этой ссылке.

- В репозиториях OpenSUSE нет официального пакета для X2Go. Есть только экспериментальная версия, доступная по ссылке.
- Проверьте, что все ТС, созданные базовым способом, введены в домен. Подробнее в статье Ввод OpenSUSE в домен. Далее проверьте конфигурацию файла `sssd.conf`, находящийся по пути `sudo vim /etc/sssd/sssd.conf`, и установите значение `False` в следующей строке:

  ```
  use_fully_qualified_names = False
  ```

- Проверьте, что на всех ТС доступны требуемые порты. Подробнее в статье

Требования к развертыванию.

1. Добавьте репозиторий X2Go с помощью команды:

```
sudo zypper addrepo -f http://packages.x2go.org/opensuse/15.4/main x2go_repos
```

2. Обновите репозиторий с помощью команды:

```
sudo zypper refresh
```

3. Чтобы установить X2Go-сервер и сопутствующие компоненты на ТС, выполните команду:

```
sudo zypper install x2goserver x2goserver-printing cups-x2go xrdp openbox
```

4. Добавьте X2Go в автозапуск с помощью команды:

```
sudo systemctl enable x2goserver.service --now
```

5. Чтобы избежать ошибки при монтировании сетевых дисков, установите пакеты `cifs-utils` и `autofs` с помощью команды:

```
sudo zypper install cifs-utils autofs
```

6. Для работы со смарт-картами:
   a. установите утилиту `pcsc-lite` и `pcsc-tools` с помощью команды:

   ```
   sudo zypper install pcsc-lite pcsc-tools
   ```

   b. запустите службу с помощью команды:

   ```
   sudo systemctl enable pcscd.service --now
   ```

7. Установите пакет `patch`, необходимый для переноса изменений между разными версиями текстовых файлов с помощью команды:

```
sudo zypper install patch
```

8. Для корректного перенаправления смарт-карт, установите утилиту `socat` с помощью команды:

```
sudo zypper install socat
```

9. Для поддержки теневых подключений Установите пакеты, необходимые для работы теневых подключений, с помощью команды:

```
sudo zypper install x2goserver-desktopsharing libQt5Svg5 libQt5Core5 libQt5Gui5
libQt5Widgets5 libQt5Network5
```

10. Для подключения через веб-клиент Настройте `xrdp`:
    a. Убедитесь, что в файле `/etc/xrdp/xrdp.ini` в секции `Session types` блок настроек `[Xorg]` расположен первым. Если это не так — переместите его в начало секции.
    b. Отредактируйте файл `/etc/xrdp/sesman.ini`:
        i. Убедитесь, что параметр `Policy` имеет значение `Default`.
        ii. Убедитесь, что параметры `TerminalServerUsers` и `TerminalServerAdmins` имеют значение `all`. В противном случае придется добавлять каждого пользователя в группы `tsusers` или `tsadmins`.
        iii. Убедитесь, что параметр `AllowAlternateShell` имеет значение `yes`.
    c. Запустите службу `xrdp` с помощью команды:

```
sudo systemctl enable xrdp --now
```

    d. Выполните перезагрузку системы.

Перезагрузка рекомендуется для корректного применения настроек. В некоторых случаях без перезагрузки могут возникать проблемы с подключением.

11. Установите агент, используя скрипт, полученный при создании серверов в каталогах ресурсов формата **ТС**, добавив в конце параметр `-d /etc/systemd/system`.

Пример скрипта:

```
rm -f ./agent-installer && wget --no-check-certificate https://<Портал внутреннего
доступа>/dist/agent-installer -O agent-installer && sudo chmod +x agent-installer && sudo
./agent-installer install -p x2go,pulsar -b https://<Портал внутреннего доступа> -d
```

```
/etc/systemd/system && sudo ./agent-installer register -b https://<Портал внутреннего доступа> -t <секрет>
```

Глава 8. Аутентификация по сертификату

При выключенном Kerberos на десктоп-клиенте

В этом разделе описано, как настроить аутентификацию пользователя на терминальном сервере по сертификату при выключенном Kerberos на десктоп-клиенте.

Linux

Для повышения безопасности подключения пользователей к терминальным серверам настройте аутентификацию по сертификату. Это можно сделать тремя способами:

Способ 1. Через условный блок, используя один порт для подключения

1. Для этого после установки терминального агента в конце файла `/etc/ssh/sshd_config`(РЕД ОС, AstraLinux, Debian, OpenSUSE)/ `/etc/openssh/sshd_config`(ALT Linux) добавьте строки:

   ```
   TrustedUserCAKeys /etc/termit-agent/termit-ssh-ca.pub
   AuthenticationMethods publickey,password
   Match User ИМЯ_АДМИНИСТРАТОРА_СИСТЕМЫ
         AuthenticationMethods password publickey
   ```

   Таким образом обычным пользователям разрешается заходить только с использованием временных сертификатов, подписанных доверенным ключом. При этом администратор системы может подключаться с использованием пароля или ключа, как и до внесения изменений.

2. Перезагрузите sshd с помощью команды:

   ```
   sudo systemctl restart sshd
   ```

Способ 2. Через условный блок, используя два порта (для пользователей и администратора)

1. Для этого после установки терминального агента в файле `/etc/ssh/sshd_config`(РЕД ОС, AstraLinux, Debian, OpenSUSE)/ `/etc/openssh/sshd_config`(ALT Linux) раскомментируйте строку с указанием порта, если она ранее не была изменена (Смотрите Ручное изменение портов в файле конфигурации шлюза и агента,).

2. Ниже добавьте похожую строку с указанием произвольного порта, по которому сможет подключаться только администратор системы:

```
Port 22
Port 2222
```

Где:

- **Port 22** — порт для пользователей;
- **Port 2222** — порт для администратора.

3. В том же файле в конце добавьте строки:

```
Match LocalPort 22
      TrustedUserCAKeys /etc/termit-agent/termit-ssh-ca.pub
      AuthenticationMethods publickey,password
Match LocalPort 2222, User ИМЯ_АДМИНИСТРАТОРА_СИСТЕМЫ
      AuthenticationMethods password publickey
```

Таким образом обычным пользователям разрешается заходить только с использованием временных сертификатов, подписанных доверенным ключом, но только через порт 22. При этом администратор системы может подключаться с использованием пароля или ключа, как и до внесения изменений, но только через порт 2222.

4. Перезагрузите sshd с помощью команды:

```
sudo systemctl restart sshd
```

На системах с SELinux (например РЕД ОС) могут возникнуть проблемы при использовании нового порта для SSH. Чтобы разрешить сервису SSH использовать новый порт, выполните команду:

```
sudo semanage port -a -t ssh_port_t -p tcp 2222
```

Способ 3. Через дублирование ssh-сервиса

1. Создайте копии конфигурационного файла с помощью команды:
   i. Для РЕД ОС, AstraLinux, Debian, OpenSUSE:

```
sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config_crt_pwd
```

ii. Для ALT Linux:

```
sudo cp /etc/openssh/sshd_config /etc/openssh/sshd_config_crt_pwd
```

2. В новом конфигурационном файле измените строку с портом на любой другой:

```
Port 2222
```

3. Создайте новый сервис sshd через копирование конфигурации существующего сервиса с помощью команды:

```
sudo cp /usr/lib/systemd/system/sshd.service /usr/lib/systemd/system/sshd_crt_pwd.service
```

4. Измените строку ExecStart (укажите путь до нового конфигурационного файла `sshd_config_crt_pwd`) в скопированном файле:
   i. Для РЕД ОС, AstraLinux, Debian, OpenSUSE:

```
ExecStart=/usr/sbin/sshd -D -f /etc/ssh/sshd_config_crt_pwd
```

   ii. Для ALT Linux:

```
ExecStart=/usr/sbin/sshd -D -f /etc/openssh/sshd_config_crt_pwd
```

5. Запустите и добавьте в автозагрузку новый сервис с помощью команды:

```
sudo systemctl enable --now sshd_crt_pwd.service
```

На системах с SELinux (например РЕД ОС) могут возникнуть проблемы при использовании нового порта для SSH. Чтобы разрешить сервису SSH использовать новый порт, выполните команду:

```
sudo semanage port -a -t ssh_port_t -p tcp 2222
```

6. Добавьте в конце конфигурационного файла первого экземпляра ssh-сервиса `/etc/ssh/sshd_config`/ `/etc/openssh/sshd_config`(ALT Linux):
   i. Для РЕД ОС, AstraLinux, Debian, OpenSUSE:

```
AllowUsers ИМЯ_АДМИНИСТРАТОРА_СИСТЕМЫ
```

ii. Для ALT Linux:

```
AllowUsers ИМЯ_АДМИНИСТРАТОРА_СИСТЕМЫ
```

Таким образом по порту 22 сможет подключаться только администратор системы.

7. Добавьте в конце конфигурационного файла первого экземпляра ssh-сервиса `/etc/ssh/sshd_config_crt_pwd`:

i. Для РЕД ОС, AstraLinux, Debian, OpenSUSE:

```
TrustedUserCAKeys /etc/termit-agent/termit-ssh-ca.pub
AuthenticationMethods publickey,password
```

ii. Для ALT Linux:

```
TrustedUserCAKeys /etc/termit-agent/termit-ssh-ca.pub
AuthenticationMethods publickey,password
```

Таким образом по 2222 порту смогут подключаться остальные пользователи, которые предоставят сертификат, подписанный доверенным ключом.

8. Перезапустите сервисы с помощью команды:

```
sudo systemctl restart sshd && sudo systemctl restart sshd_crt_pwd
```

При включенном Kerberos на десктоп-клиенте

В этом разделе описано, как настроить аутентификацию пользователя на терминальном сервере по сертификату при включенном Kerberos на десктоп-клиенте.

Linux

Для повышения безопасности подключения пользователей к терминальным серверам настройте аутентификацию по сертификату. Это можно сделать тремя способами:

Способ 1. Через условный блок, используя один порт для подключения

1. Для этого после установки терминального агента в конце файла `/etc/ssh/sshd_config`(РЕД ОС, AstraLinux, Debian, OpenSUSE)/ `/etc/openssh/sshd_config`(ALT Linux) добавьте строки:

```
TrustedUserCAKeys /etc/termit-agent/termit-ssh-ca.pub
AuthenticationMethods gssapi-with-mic publickey,password
Match User ИМЯ_АДМИНИСТРАТОРА_СИСТЕМЫ
      AuthenticationMethods password publickey
```

   Таким образом обычным пользователям разрешается заходить только с использованием временных сертификатов, подписанных доверенным ключом. При этом администратор системы может подключаться с использованием пароля или ключа, как и до внесения изменений.

2. Перезагрузите sshd с помощью команды:

```
sudo systemctl restart sshd
```

Способ 2. Через условный блок, используя два порта (для пользователей и администратора)

1. Для этого после установки терминального агента в файле `/etc/ssh/sshd_config`(РЕД ОС, AstraLinux, Debian, OpenSUSE)/ `/etc/openssh/sshd_config`(ALT Linux) раскомментируйте строку с указанием порта, если она ранее не была изменена (Смотрите Ручное изменение портов в файле конфигурации шлюза и агента,).
2. Ниже добавьте похожую строку с указанием произвольного порта, по которому сможет подключаться только администратор системы:

```
Port 22
Port 2222
```

   Где:

   - **Port 22** — порт для пользователей;
   - **Port 2222** — порт для администратора.

3. В том же файле в конце добавьте строки:

```
Match LocalPort 22
      TrustedUserCAKeys /etc/termit-agent/termit-ssh-ca.pub
      AuthenticationMethods gssapi-with-mic publickey,password
Match LocalPort 2222, User ИМЯ_АДМИНИСТРАТОРА_СИСТЕМЫ
```

```
    AuthenticationMethods password publickey
```

Таким образом обычным пользователям разрешается заходить только с использованием временных сертификатов, подписанных доверенным ключом, но только через порт 22. При этом администратор системы может подключаться с использованием пароля или ключа, как и до внесения изменений, но только через порт 2222.

4. Перезагрузите sshd с помощью команды:

```
sudo systemctl restart sshd
```

На системах с SELinux (например РЕД ОС) могут возникнуть проблемы при использовании нового порта для SSH. Чтобы разрешить сервису SSH использовать новый порт, выполните команду:

```
sudo semanage port -a -t ssh_port_t -p tcp 2222
```

Способ 3. Через дублирование ssh-сервиса

1. Создайте копии конфигурационного файла с помощью команды:
   i. Для РЕД ОС, AstraLinux, Debian, OpenSUSE:

   ```
   sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config_crt_pwd
   ```

   ii. Для ALT Linux:

   ```
   sudo cp /etc/openssh/sshd_config /etc/openssh/sshd_config_crt_pwd
   ```

2. В новом конфигурационном файле измените строку с портом на любой другой:

```
Port 2222
```

3. Создайте новый сервис sshd через копирование конфигурации существующего сервиса с помощью команды:

```
sudo cp /usr/lib/systemd/system/sshd.service /usr/lib/systemd/system/sshd_crt_pwd.service
```

4. Измените строку ExecStart (укажите путь до нового конфигурационного файла `sshd_config_crt_pwd`) в скопированном файле:
   i. Для РЕД ОС, AstraLinux, Debian, OpenSUSE:

```
ExecStart=/usr/sbin/sshd -D -f /etc/ssh/sshd_config_crt_pwd
```

   ii. Для ALT Linux:

```
ExecStart=/usr/sbin/sshd -D -f /etc/openssh/sshd_config_crt_pwd
```

5. Запустите и добавьте в автозагрузку новый сервис с помощью команды:

```
sudo systemctl enable --now sshd_crt_pwd.service
```

На системах с SELinux (например РЕД ОС) могут возникнуть проблемы при использовании нового порта для SSH. Чтобы разрешить сервису SSH использовать новый порт, выполните команду:

```
sudo semanage port -a -t ssh_port_t -p tcp 2222
```

6. Добавьте в конце конфигурационного файла первого экземпляра ssh-сервиса `/etc/ssh/sshd_config`/ `/etc/openssh/sshd_config`(ALT Linux):
   i. Для РЕД ОС, AstraLinux, Debian, OpenSUSE:

```
AllowUsers ИМЯ_АДМИНИСТРАТОРА_СИСТЕМЫ
```

   ii. Для ALT Linux:

```
AllowUsers ИМЯ_АДМИНИСТРАТОРА_СИСТЕМЫ
```

   Таким образом по порту 22 сможет подключаться только администратор системы.

7. Добавьте в конце конфигурационного файла первого экземпляра ssh-сервиса `/etc/ssh/sshd_config_crt_pwd`:
   i. Для РЕД ОС, AstraLinux, Debian, OpenSUSE:

```
TrustedUserCAKeys /etc/termit-agent/termit-ssh-ca.pub
```

```
AuthenticationMethods gssapi-with-mic publickey,password
```

ii. Для ALT Linux:

```
TrustedUserCAKeys /etc/termit-agent/termit-ssh-ca.pub
AuthenticationMethods gssapi-with-mic publickey,password
```

Таким образом по 2222 порту смогут подключаться остальные пользователи, которые предоставят сертификат, подписанный доверенным ключом.

8. Перезапустите сервисы с помощью команды:

```
sudo systemctl restart sshd && sudo systemctl restart sshd_crt_pwd
```

Глава 9. Шлюзы

В этом разделе описано, как установить шлюз удаленного доступа и настроить аутентификацию.

Для передачи данных между десктоп-клиентом и шлюзом применяется протокол TLS версии не ниже 1.2.

РЕД ОС

1. Установите дополнительное ПО на TLS-шлюз:

```
sudo dnf install wget
```

2. Установите агент, используя скрипт, полученный при создании TLS-шлюза по следующему пути:
   a. На портале администрирования в левом меню выберите раздел **Настройки**.
   b. Перейдите на вкладку **Внешний доступ**.
   c. Наведите курсор на шлюз и нажмите ⋮.
   d. Нажмите **[Установить агент]**.
   e. Скопируйте команду для установки агента и выполните ее на шлюзе удаленного доступа.
3. Разрешите работу в SELinux с помощью команды:

```
sudo semanage port -a -t http_port_t -p tcp 4443 || sudo semanage port -m -t http_port_t
-p tcp 4443
```

Astra Linux

1. Установите дополнительное ПО на TLS-шлюз:

```
sudo apt install wget
```

2. Установите агент, используя скрипт, полученный при создании TLS-шлюза по следующему пути:

a. На портале администрирования в левом меню выберите раздел **Настройки**.
b. Перейдите на вкладку **Внешний доступ**.
c. Наведите курсор на шлюз и нажмите ⋮.
d. Нажмите **[Установить агент]**.
e. Скопируйте команду для установки агента и выполните ее на шлюзе удаленного доступа.

ALT Linux

1. Установите дополнительное ПО на TLS-шлюз:

```
sudo apt-get install wget
```

2. Установите агент, используя скрипт, полученный при создании TLS-шлюза по следующему пути:
   a. На портале администрирования в левом меню выберите раздел **Настройки**.
   b. Перейдите на вкладку **Внешний доступ**.
   c. Наведите курсор на шлюз и нажмите ⋮.
   d. Нажмите **[Установить агент]**.
   e. Скопируйте команду для установки агента и выполните ее на шлюзе удаленного доступа.

Debian

1. Установите дополнительное ПО на TLS-шлюз:

```
sudo apt install wget
```

2. Установите агент, используя скрипт, полученный при создании TLS-шлюза по следующему пути:
   a. На портале администрирования в левом меню выберите раздел **Настройки**.
   b. Перейдите на вкладку **Внешний доступ**.
   c. Наведите курсор на шлюз и нажмите ⋮.
   d. Нажмите **[Установить агент]**.

e. Скопируйте команду для установки агента и выполните ее на шлюзе удаленного доступа.

OpenSUSE

1. Установите дополнительное ПО на TLS-шлюз:

```
sudo zypper install wget
```

2. Установите агент, используя скрипт, полученный при создании TLS-шлюза по следующему пути:
   a. На портале администрирования в левом меню выберите раздел **Настройки**.
   b. Перейдите на вкладку **Внешний доступ**.
   c. Наведите курсор на шлюз и нажмите ⋮.
   d. Нажмите **[Установить агент]**.
   e. Скопируйте команду для установки агента и выполните ее на шлюзе удаленного доступа.

Глава 10. ГОСТ-шифрование

Термит поддерживает подключение внешних клиентов с использованием туннелей, шифрованных ГОСТ-алгоритмами. Для этого в DMZ зоне можно разместить сервер с ГОСТ-stunnel, который будет принимать входящие подключения, и пересылать их на внешние балансировщики нагрузки и шлюзы. Либо, можно настроить ГОСТ-stunnel на шлюзе и\или внешнем балансировщике. Данная инструкция описывает конфигурацию ГОСТ-stunnel на шлюзе.

10.1. Настройка серверов

Для поддержки подключений пользователей, необходимо установить и настроить stunnel (версия от КриптоПро) на шлюзе внешнего доступа.

Настройка TLS-шлюза

1. Установите КриптоПро CSP. Версия `stunnel` от КриптоПро поставляется в комплекте с этим продуктом.

   Начиная с CSP 5.0 R3 в демонстрационную лицензию больше не входит «TLS-сервер», поэтому в целях тестирования можно установить версию CSP 5.0 R2.

   a. Скачайте архив для нужной ОС по ссылке.
   b. Распакуйте полученный архив с помощью команды:

   ```
   tar -xvzf linux-amd64*.tgz
   ```

   c. При необходимости установите пакет `lsb` для корректной инсталляции КриптоПро CSP.
   d. Запустите установку КриптоПро CSP с помощью команды:

   ```
   sudo ./install_gui.sh
   ```

   При установке необходимо выбрать компонент `stunnel`.

2. Сгенерируйте ключи на шлюзе.
   a. Создайте контейнер на носителе `HDIMAGE`, задав пустой пароль:

```
sudo /opt/cprocsp/bin/amd64/csptest -keyset -newkeyset -cont '\\.\HDIMAGE\termit'
```

b. Создайте запрос на получение сертификата, заменяя `%e@mail%` (необязательный параметр), `%gw_FQDN%` на используемые:

```
sudo /opt/cprocsp/bin/amd64/cryptcp -creatrqst -dn "E=%e@mail%, C=RU, CN=%gw_FQDN%,
SN=%gw_FQDN%" -hashalg 1.2.643.7.1.1.2.2 -nokeygen -both -ku -cont
'\\.\HDIMAGE\termit' gw.req
```

c. Выведите содержимое файла `gw.req` с помощью команды:

```
cat gw.req
```

3. Получите сертификат от центра сертификации (ЦС).

Для работы потребуются следующие сертификаты:

- сертификат шлюза;
- корневой сертификат ЦС;
- промежуточные сертификаты ЦС (при наличии).

Если в вашей организации используется корпоративный центр сертификации, порядок получения сертификатов определяется внутренними регламентами.

Ниже в качестве примера приведён процесс получения сертификата от центра сертификации КриптоПро:

a. Откройте портал по ссылке.

b. Вставьте содержимое файла `gw.req` в поле `Base-64-шифрованный запрос сертификата`.

c. Нажмите **[Выдать]**.

d. Нажмите "Загрузить сертификат", полученный файл `certnew.cer` переименуйте в `gw.cer`.

e. Нажмите **[Загрузить цепочку сертификатов]**. Полученный файл `certnew.p7b` переименуйте в `gw.p7b`.

f. Откройте портал по ссылке.

g. Нажмите **[Загрузка сертификата ЦС]**. Переименуйте полученный файл на

`ca.cer`.

4. Установите сертификаты на шлюзе.
   a. Загрузите полученные сертификаты на шлюз:
      - `gw.cer` - сертификат шлюза,
      - `ca.cer` - сертификат корневого Центра Сертификации,
      - `gw.p7b` - сертификат шлюза и промежуточных центров сертификации.
   b. Установите сертификат ЦС с помощью команды:

      ```
      sudo /opt/cprocsp/bin/amd64/certmgr -inst -store uRoot -file ca.cer
      ```

      После выполнения команды необходимо нажать `o` и затем `Enter` для подтверждения добавления корневого сертификата.

   c. Установите промежуточный сертификат из цепочки сертификатов:

      ```
      sudo /opt/cprocsp/bin/amd64/certmgr -inst -store uRoot -file gw.p7b
      ```

      i. После выполнения команды введите номер промежуточного сертификата (например, `1`) и нажмите `Enter`.
      ii. Для подтверждения добавления корневого сертификата нажмите `o` и затем `Enter`.
      iii. Если файл `gw.p7b` содержит несколько промежуточных сертификатов (или они находятся в другом файле), добавьте их аналогичным способом.
   d. Установите сертификат шлюза:

      ```
      sudo /opt/cprocsp/bin/amd64/certmgr -inst -file gw.cer -cont '\\.\HDIMAGE\termit'
      ```

   e. Проверьте статус сертификата шлюза с помощью команды:

      ```
      sudo /opt/cprocsp/bin/amd64/certmgr -list -store uMy
      ```

      Значение поля `PrivateKey Link` должно быть `Yes`.

5. Настройте и запустите ГОСТ-stunnel на шлюзе:
   a. Остановите и отключите сервис `termit-stunnel` на шлюзе:

      ```
      sudo systemctl disable --now termit-stunnel
      ```

Данную процедуру необходимо повторять каждый раз после установки\переустановки\обновления агента шлюза.

b. Создайте папку `/etc/stunnel-gost/` и скопируйте в неё файл из сертификата `gw.cer`.

c. В папке `/etc/stunnel-gost/` создайте файл конфигурации `stunnel-gost.conf` со следующим содержимым:

```
pid=/var/opt/cprocsp/tmp/stunnel_cli.pid
output=/var/log/stunnel_gost.log
socket = l:TCP_NODELAY=1
socket = r:TCP_NODELAY=1
debug = 5

[https-gw]
accept = 0.0.0.0:4444
connect = 127.0.0.1:8080
cert=/etc/stunnel-gost/gw.cer
verify=1

[https-broker]
accept = 0.0.0.0:4445
connect = %fqdn_или_ip_внеш_балансировщика%:443
cert=/etc/stunnel-gost/gw.cer
verify=1
```

d. В созданном файле `stunnel-gost.conf` замените следующие параметры:

`%fqdn_или_ip_внеш_балансировщика%` — замените на FQDN или IP-адрес внешнего балансировщика.

e. Запустите ГОСТ-stunnel:

```
sudo /opt/cprocsp/sbin/amd64/stunnel_thread /etc/stunnel-gost/stunnel-gost.conf
```

f. Проверьте логи `stunnel` на наличие ошибок:

```
sudo cat /var/log/stunnel_gost.log
```

ГОСТ-stunnel на сервере необходимо запускать каждый раз после перезагрузки сервера. Можно добавить команду его запуска в автозагрузку или настроить запуск через сервис.

10.2. Настройка клиентских компьютеров

Для подключения к ГОСТ-stunnel, на клиентском компьютере необходимо настроить ГОСТ-stunnel, а также стартовать его перед запуском Термит десктоп-клиента, а сам Термит десктоп-клиент необходимо запускать с соответствующим параметром.

Настройка клиентских компьютеров с ОС Linux

Начиная с CSP 5.0 R3 в демонстрационную лицензию больше не входит «TLS-сервер», поэтому в целях тестирования можно установить версию CSP 5.0 R2.

1. Скачайте и установите КриптоПро CSP по ссылке.
2. Распакуйте полученный архив с помощью команды:

```
tar -xvzf linux-amd64.tgz
```

3. Установите пакет `lsb`, необходимый для корректной инсталляции КриптоПро CSP.
4. Запустите установку КриптоПро CSP с помощью команды:

```
sudo ./install_gui.sh
```

   При установке необходимо выбрать компонент `stunnel`.

5. Создайте конфигурационный файл `~/stunnel-gost/stunnel-gost.conf` с содержимым ниже:

```
pid=/tmp/stunnel_cli.pid
output=/home/%YourUserName%/stunnel-gost/stunnel.log
socket=l:TCP_NODELAY=1
socket=r:TCP_NODELAY=1
debug=5

[GW]
client = yes
accept  = 127.0.0.1:4444
connect = %fqdn_или_ip_шлюза%:4444
verify = 2

[BROKER]
client = yes
accept  = 127.0.0.1:4445
connect = %fqdn_или_ip_шлюза%:4445
verify = 2
```

6. В созданном файле замените следующие параметры:
   - `%YourUserName%` — замените на реальное имя пользователя, так как переменные среды и символ `~` в данном файле не поддерживаются;
   - `%fqdn_или_ip_шлюза%` — замените на FQDN или IP-адрес шлюза.
7. Создайте файл `~/stunnel-gost/proxyconf.json` со следующим содержимым:

```
{
  "gateway": {
    "host": "127.0.0.1",
    "port": 4444
  },
  "broker": {
    "host": "127.0.0.1",
    "port": 4445
  }
}
```

8. Добавьте сертификаты в доверенные корневые центры сертификации с помощью `certmgr`:
   - корневой сертификат из файла `ca.cer`:

```
/opt/cprocsp/bin/amd64/certmgr -inst -store uRoot -file ca.cer
```

   - промежуточные сертификаты из файла `gw.p7b`:

```
/opt/cprocsp/bin/amd64/certmgr -inst -store uRoot -file gw.p7b
```

9. Запустите ГОСТ-stunnel с помощью команды:

```
/opt/cprocsp/sbin/amd64/stunnel_thread ~/stunnel-gost/stunnel-gost.conf
```

10. Запустите десктоп-клиент Termit с параметром `--proxy-conf`:

```
/opt/termit-desktop/termit-desktop --proxy-conf ~/stunnel-gost/proxyconf.json
```

10.3. Обновление Termit агента шлюза с ГОСТ-stunnel

Для обновления агента Termit на шлюзе:

1. Остановите ГОСТ-stunnel с помощью команды:

```
sudo pkill stunnel_thread
```

2. Обновите агент Termit на шлюзе.
3. Остановите и отключите сервис `termit-stunnel` на шлюзе:

```
sudo systemctl disable --now termit-stunnel
```

4. Запустите ГОСТ-stunnel с помощью команды:

```
sudo /opt/cprocsp/sbin/amd64/stunnel_thread etc/stunnel-gost/stunnel-gost.conf
```

Глава 11. Установка десктоп-клиента

Перед началом работы скачайте и установите десктоп-клиент. Сделать это можно двумя способами: скачать файл для установки по ссылке или скачать файл для установки из портала администрирования/веб-клиента Termit. Для того, чтобы скачать и установить дескоп-клиент:

1. Войдите на портал администрирования/веб-клиент Termit. В адресной строке браузера введите адрес портала внутреннего доступа, например https://internal.termit.local/admin.

Использование веб-клиента Termit доступно, только если на портале администрирования активирована опция **Доступ к пользовательскому порталу**. Подробнее о настройке.

2. Выберите домен:
   - **Авто**. Система автоматически выберет LDAP-каталог из списка доступных каталогов или внутренних пользователей.
   - **Внутренние пользователи**. Вход будет выполнен под учетной записью локального администратора «tadm».
   - **Имя LDAP-каталога**, отображаемое полное имя которого было задано при добавлении в систему, например «ldap123».
3. Для входа под учетной записью администратора системы укажите в поле ввода:
   - **Логин** — введите логин.
   - **Пароль** — введите пароль.
   - **Одноразовый код** — введите полученный из приложения-аутентификатора одноразовый пароль.

1. Для входа под учетной записью локального администратора используйте по умолчанию логин `tadm` и пароль `admin`.
2. Локальная учетная запись предназначена только для настройки брокера. Ее нельзя использовать для входа в десктоп-клиент, запуска приложений и рабочих столов.

4. Нажмите **[Войти]**.

После успешного входа в систему откроется интерфейс портала администрирования.

5. В левом меню выберите раздел **Скачать клиент**.
6. Выберите операционную систему и установите десктоп-клиент.

Linux

РЕД ОС

1. Перейдите на вкладку **Linux**.
2. Скачайте **Десктоп-клиент для Linux › RPM** и выполните шаги по установке.
3. Обновите репозиторий с помощью команды:

```
sudo dnf update
```

4. Установите десктоп-клиент:

```
sudo dnf install ./termit-desktop-2.*.rpm
```

Где:

`./termit-desktop-2.*.rpm` — название файла.

Astra Linux

1. Перейдите на вкладку **Linux**.
2. Скачайте **Десктоп-клиент для Linux › DEB**.
3. Установите клиент Termit:

```
sudo apt install ./termit-desktop-2.*.deb
```

Где:

`./termit-desktop-2.*.deb` — название файла.

Установка в закрытой программной среде

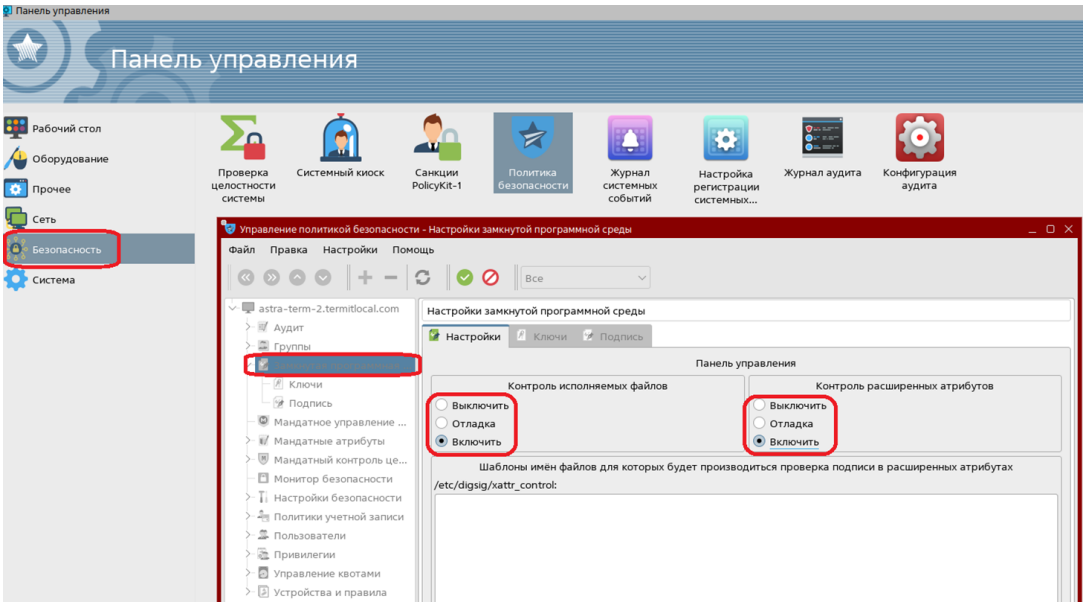
1. Перед установкой поместите публичный (открытый) ключ, скачанный с брокера по пути https://broker.example.com/dist/termit_astra.key, в директорию `/etc/digsig/keys` с помощью команды:

```
sudo cp %Путь_к_Открытому_Ключу% /etc/digsig/keys
```

2. Обновите ключи с помощью команды:

```
sudo update-initramfs -u -k all
```

3. Перезагрузите ПК.
4. Включите режим замкнутой программой среды (ЗПС), если не был включен ранее:



5. Установите десктоп-клиент в обычном режиме.

   При отображении «Ошибка сегментирования» ключ не может быть считан.

ALT Linux

1. Перейдите на вкладку **Linux**.
2. Скачайте **Десктоп-клиент для Linux › RPM**.
3. Обновите репозиторий с помощью команды:

```
sudo apt-get update
```

4. Установите десктоп-клиент:

```
sudo apt-get install ./termit-desktop-2.*.rpm
```

Десктоп-клиент будет установлен и готов к запуску.

Глава 12. Ключи запуска десктоп-клиента

Параметры запуска десктоп-клиента

- `-v` — включить debug-уровень логирования десктоп-клиента и клиентов подключений к сессиям.
- `--ignore-certs` — игнорировать ошибку проверки сертификата при подключении к балансировщику/брокеру.
- `--proxy-conf "<путь_до_JSON>"` — настройка, позволяющая использовать сторонние средства шифрования для связи с балансировщиками/брокерами и ТС (через шлюз). Пример настройки см. в разделе **ГОСТ-шифрование**,.

Глава 13. Сертификаты

Для успешного подключения к Termit необходимо, чтобы используемый корневой сертификат был добавлен в список доверенных на десктоп-клиенте.

РЕД ОС

1. Скопируйте файл корневого сертификата в каталог `/etc/pki/ca-trust/source/anchors/` с помощью команды:

   ```
   sudo cp %Путь_к_сертификату% /etc/pki/ca-trust/source/anchors/
   ```

   Где: **%Путь_к_сертификату%** — полный путь к файлу сертификата в формате PEM.

2. Чтобы применить изменения, выполните команды:

   ```
   sudo update-ca-trust force-enable
   sudo update-ca-trust extract
   ```

3. Перезапустите браузер и проверьте доверие сертификату, открыв адрес брокера.

Для систем Linux иногда необходима дополнительная ручная конфигурация хранилищ браузеров Firefox и Chrome\Chromium:

Chrome\Chromium

1. Перейдите в меню **Настройки › Конфиденциальность и безопасность › Безопасность › Настроить сертификаты** и перейдите на вкладку **Центры сертификации**.
2. Добавьте необходимый сертификат или цепочку сертификатов с помощью кнопки **Импорт**.
3. Укажите необходимый уровень доверия.

Firefox

1. Перейдите в **Настройки › Приватность и защита › Защита › Сертификаты › Просмотр сертификатов**.
2. Добавьте необходимый сертификат или цепочку сертификатов с помощью кнопки **Импорт**.

3. Укажите необходимый уровень доверия.

Astra Linux

Сертификат должен быть в формате `CRT`.

Если формат сертификата отличается, необходимо выполнить его конвертацию используя утилиту `openssl`, например:

- Если сертификат имеет кодировку `DER`:

```
openssl x509 -inform DER -in %Исходный_Сертификат% -out %Конечный_Сертификат%
```

- Если исходный сертификат имеет кодировку `PEM`:

```
openssl x509 -inform PEM -in %Исходный_Сертификат% -out %Конечный_Сертификат%
```

Где:

- **%Исходный_Сертификат%** — путь к исходному сертификату «certificate.cer».
- **%Конечный_Сертификат%** — путь к конечному сертификату, например «certificate.crt».

1. Скопируйте полученные выше сертификаты в каталог `/usr/local/share/ca-certificates` с помощью команды:

   ```
   sudo cp %Путь_к_сертификату% /usr/local/share/ca-certificates
   ```

   Где: **%Путь_к_сертификату%** — полный путь к файлу сертификата.

2. Чтобы применить изменения, выполните команду:

   ```
   sudo update-ca-certificates
   ```

3. Перезапустите браузер и проверьте доверие сертификату, открыв адрес брокера.

Для систем Linux иногда необходима дополнительная ручная конфигурация хранилищ браузеров Firefox и Chrome\Chromium:

Chrome\Chromium

1. Перейдите в меню **Настройки** › **Конфиденциальность и безопасность** › **Безопасность** › **Настроить сертификаты** и перейдите на вкладку **Центры сертификации**.
2. Добавьте необходимый сертификат или цепочку сертификатов с помощью кнопки **Импорт**.
3. Укажите необходимый уровень доверия.

Firefox

1. Перейдите в **Настройки** › **Приватность и защита** › **Защита** › **Сертификаты** › **Просмотр сертификатов**.
2. Добавьте необходимый сертификат или цепочку сертификатов с помощью кнопки **Импорт**.
3. Укажите необходимый уровень доверия.

ALT Linux

Подробнее об установке сертификата можно прочесть на официальном сайте.

1. Скопируйте корневой сертификат в хранилище сертификатов с помощью команды ниже. Сертификат должен быть в формате `.CRT`.

```
sudo cp %путь_к_сертификату% /etc/pki/ca-trust/source/anchors/
```

2. Обновите хранилище сертификатов с помощью команды:

```
sudo update-ca-trust
```

Для систем Linux иногда необходима дополнительная ручная конфигурация хранилищ браузеров Firefox и Chrome\Chromium:

Chrome\Chromium

1. Перейдите в меню **Настройки** › **Конфиденциальность и безопасность** › **Безопасность** › **Настроить сертификаты** и перейдите на вкладку **Центры сертификации**.
2. Добавьте необходимый сертификат или цепочку сертификатов с помощью кнопки **Импорт**.

3. Укажите необходимый уровень доверия.

Firefox

1. Перейдите в **Настройки** › **Приватность и защита** › **Защита** › **Сертификаты** › **Просмотр сертификатов**.
2. Добавьте необходимый сертификат или цепочку сертификатов с помощью кнопки **Импорт**.
3. Укажите необходимый уровень доверия.

OpenSUSE

1. Скопируйте файл корневого сертификата в каталог `/etc/pki/trust/anchors` с помощью команды:

   ```
   sudo cp %Путь_к_сертификату% /etc/pki/trust/anchors
   ```

   Где: **%Путь_к_сертификату%** — полный путь к файлу сертификата в формате `PEM`.

2. Обновите хранилище сертификатов с помощью команды:

   ```
   sudo update-ca-certificates
   ```

3. Перезапустите браузер и проверьте доверие сертификату, открыв адрес брокера.

Для систем Linux иногда необходима дополнительная ручная конфигурация хранилищ браузеров Firefox и Chrome\Chromium:

Chrome\Chromium

1. Перейдите в меню **Настройки** › **Конфиденциальность и безопасность** › **Безопасность** › **Настроить сертификаты** и перейдите на вкладку **Центры сертификации**.
2. Добавьте необходимый сертификат или цепочку сертификатов с помощью кнопки **Импорт**.
3. Укажите необходимый уровень доверия.

Firefox

1. Перейдите в **Настройки** › **Приватность и защита** › **Защита** › **Сертификаты** › **Просмотр сертификатов**.

2. Добавьте необходимый сертификат или цепочку сертификатов с помощью кнопки **Импорт**.
3. Укажите необходимый уровень доверия.

Глава 14. Выпуск самоподписанного сертификата

14.1. 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.

14.2. 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 команды:

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

3. Выпустите сертификат на основе сформированного запроса:

```
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.

4. Проверьте созданный сертификат и наличие в нем `subjectAltName` с помощью команды:

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

Результат:

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

5. Импортируйте ключ и сертификат в настройках HTTPS.

Глава 15. Обновление пакетов десктоп-клиента на РЕД ОС при установке в закрытом контуре

Когда нет доступа к внешней сети или есть доступ только к определенным ресурсам, то можно обновить пакеты десктоп-клиента на РЕД ОС в закрытом контуре.

1. Создайте каталог на ПК с помощью команды:

```
$ mkdir /home/user/repo
```

2. Скачайте пакет `termit-desktop` в каталог `repo`.
3. Скачайте пакеты из репозитория в каталог `repo` с помощью команды:

```
$ dnf repoquery --requires --resolve --recursive ./termit-desktop-2.***.rpm | grep -v
"i686" | sort -u > packages.txt
$ dnf download $(cat packages.txt) --downloaddir .
```

4. Перед созданием репозитория установите пакет с помощью команды:

```
sudo dnf install createrepo_c
```

5. Создайте репозиторий с помощью команды:

```
$ createrepo_c .
```

6. Перенесите пакеты на другой ПК, на котором нет доступа к внешней сети.
7. Отключите репозитории на ПК с помощью команды:

```
$ sudo dnf config-manager --set-disabled kernels updates base
```

8. Добавьте локальный репозиторий с помощью команды:

```
$ sudo dnf config-manager --add-repo /home/user/repo
```

9. Отключите проверку ключей с помощью команды:

```
$ sudo dnf config-manager --save --setopt=home_user_repo.gpgcheck=0
```

10. Установите пакет с помощью команды:

```
$ sudo dnf install termit-desktop-2.***.rpm
```

Глава 16. Смарт-карты

Аутентификация пользователей с использованием смарт-карт возможна с рабочих станций на ОС Linux. Смарт-карты могут использоваться в качестве метода многофакторной аутентификации на портале администрирования, брокере и в корпоративных приложениях. Одно из требований использования смарт-карт для аутентификации — это наличие балансировщиков нагрузки перед брокерами.

Требования к архитектуре

В инфраструктуре Termit должен использоваться балансировщик нагрузки перед брокерами, настроенный по инструкции из раздела Установка и настройка HAProxy.

Требования к сертификату

- В «Точках распространения списков отзыва (CRL)» необходимо указать путь до CRL в формате URL=http:// , и соответственно нужно, чтобы CRL-файл был опубликован по этому пути.
- SAN (дополнительное имя субъекта) — Имя субъекта=%UPN%, где %UPN% — User Principal Name, например user1@termit.local.
- Опубликованный по http CRL-файл не старше 7 дней.
- Использования ключа — Цифровая подпись, Шифрование ключей (a0).
- Улучшенный ключ:
  - Проверка подлинности клиента (1.3.6.1.5.5.7.3.2);
  - Вход со смарт-картой (1.3.6.1.4.1.311.20.2.2).
- При работе с Центром сертификации Active Directory рекомендуется использовать шаблон «Вход по смарт-карте» или его копию.

Требования к смарт-картам

- К клиентской машине может быть подключена одна смарт-карта.
- Смарт-карты Рутокен Lite и JaCarta LT не подходят для аутентификации на ТС под управлением ОС Linux.

Аутентифицироваться по смарт-карте можно в десктоп-клиенте и на портале администрирования, а также на ТС при подключении по

протоколам RDP и X2Go. Смарт-карты не поддерживаются при использовании настройки Запрашивать ОТП до пароля при использовании МФА,.

Настройка системы

Перед настройкой аутентификации по смарт-картам, убедитесь, что можете открыть опубликованные на ТС приложения через Termit с использованием логина/пароля или SSO.

Настройка балансировщиков нагрузки

Для корректной работы аутентификации пользователей с помощью смарт-карт, балансировщик нагрузки должен передавать клиентский сертификат брокеру. Это необходимо, чтобы брокер мог проверить подлинность пользователя и сопоставить его с учётной записью. В HAProxy это реализуется с помощью настроек в секции frontend, обслуживающей HTTPS-трафик - `frontend termit_https`. Остальные настройки HAProxy до изменения должны соответствовать рекомендуемым в инструкции,.

Если используется другой балансировщик нагрузки, необходимо настроить аналогичную передачу клиентского сертификата, чтобы брокер мог выполнять аутентификацию пользователей по смарт-картам.

Для настройки балансировщика нагрузки HAProxy при использовании смарт-карты выполните следующие шаги:

1. Измените файл конфигурации `/etc/haproxy/haproxy.cfg` на внутреннем и (при наличии) внешних балансировщиках:

   Пример конфигурации для внутреннего балансировщика в режиме SSL Bridging

   Измените строку с:

   ```
   bind :443 ssl crt /etc/haproxy/cert.pem
   ```

   на следующее значение:

   ```
   bind :443 ssl crt /etc/haproxy/cert.pem force-tlsv12 ca-file /etc/haproxy/ca-clients.pem
   verify optional
   ```

```
http-request set-header X-SSL-Client-Cert "-----BEGIN CERTIFICATE-----
%{+Q}[ssl_c_der,base64]-----END CERTIFICATE-----"
```

где `/etc/haproxy/ca-clients.pem` - корневой сертификат (CA), которым подписаны клиентские сертификаты десктоп-клиентов.

Пример конфигурации для внешнего балансировщика в режиме SSL Bridging

Измените строку с:

```
bind :443 ssl crt /etc/haproxy/external-lb-cert.pem
```

на следующее значение:

```
bind :443 ssl crt /etc/haproxy/external-lb-cert.pem force-tlsv12 ca-file /etc/haproxy/ca-
clients.pem verify optional
http-request set-header X-SSL-Client-Cert "-----BEGIN CERTIFICATE-----
%{+Q}[ssl_c_der,base64]-----END CERTIFICATE-----"
```

где `/etc/haproxy/ca-clients.pem` - корневой сертификат (CA), которым подписаны клиентские сертификаты десктоп-клиентов.

2. Проверьте отредактированный файл конфигурации с помощью команды:

```
sudo haproxy -f /etc/haproxy/haproxy.cfg -c
```

Вывод команды не должен содержать ошибок. Предупреждения (warnings) не являются ошибками.

3. Перезапустите службу с помощью команды:

```
sudo systemctl restart haproxy
```

4. Проверьте доступность Termit по URL адресу, указанному при установке брокера.

Если используется другой балансировщик нагрузки, необходимо настроить аналогичную передачу клиентского сертификата, чтобы брокер мог выполнять аутентификацию пользователей по смарт-картам.

Аутентификация в десктоп-клиенте и на портале администрирования на ОС Linux

Для аутентификации в десктоп-клиенте и на портале администрирования по смарт-карте необходимо выполнить следующие шаги:

1. Добавьте PKCS#11 библиотеку используемого устройства в базу данных NSS, чтобы браузер и приложения могли получать доступ к сертификатам, хранящимся на смарт-карте. Пример для Рутокен:

   ```
   modutil -dbdir sql:$HOME/.pki/nssdb/ -add "rutoken test" -libfile
   /opt/aktivco/rcc/librtpkcs11ecp.so -mechanisms FRIENDLY
   ```

   Для других типов смарт-карт необходимо указать соответствующую PKCS#11 библиотеку, предоставляемую производителем устройства.

2. Проверьте, что токен подключен в один из слотов с помощью команды:

   ```
   modutil -dbdir sql:$HOME/.pki/nssdb/ -list
   ```

3. Сконфигурируйте балансировщик нагрузки по инструкции.
4. Добавьте сертификат центра сертификации, которым подписан сертификат смарт-карты, в доверенные в UI брокера.
5. Установите на клиентскую машину драйвер смарт-карты.
6. Убедитесь, что сертификат со смарт-карты соответствует требованиям.

Аутентификация в приложениях на ТС под управлением ОС Linux

Для аутентификации в приложениях на ТС под управлением ОС Linux необходимо выполнить следующие шаги:

1. Установите драйвер смарт-карты на ТС. Обычно присутствует в ОС Linux.
2. Установите библиотеку PKCS#11 на ТС. Например, соответствующую библиотеку для rutoken можно скачать по ссылке https://www.rutoken.ru/support/download/pkcs/, выбрав нужный файл.
3. На Astra Linux и Debian ТС установите пакет krb5-pkinit на ТС. Данный шаг пропускается для других ОС.

4. Внесите следующие изменения в файл **/etc/krb5.conf** в секции [libdefaults] на ТС:
   a. Добавьте параметр **pkinit_anchors** (можно задать несколько пар параметр-значение), и в качестве значения укажите путь до корневого сертификата в формате PEM центра сертификации, который сгенерировал сертификат, записанный на смарт-карту, с префиксом «FILE:». Например, `pkinit_anchors = FILE:/certs/ca.pem`.
   b. Добавьте параметр **pkinit_identities** (можно задать несколько пар параметр-значение), и в качестве значения укажите путь до библиотеки PKCS#11, соответствующей используемым смарт-картам, с префиксом «PKCS11:». Например, `pkinit_identities = PKCS11:/opt/aktivco/rutokenecp/x86_64/librtpkcs11ecp.so`.
5. В файл **/etc/krb5.conf** в секцию [realms] добавьте нужный realm домена:

```
pkinit_kdc_hostname=ad.termit.local
pkinit_eku_checking=kpServerAuth
```

6. Убедитесь, что настроена аутентификация по сертификатам в ` sshd_config`. Учтите, что параметр `AuthenticationMethods` будет только со значением `publickey`, например:

```
TrustedUserCAKeys /etc/termit-agent/termit-ssh-ca.pub

AuthenticationMethods publickey

Match User ИМЯ_АДМИНИСТРАТОРА_СИСТЕМЫ
    AuthenticationMethods password publickey
Match all
```

   Подробнее в разделе.

7. Убедитесь, что в политиках Termit разрешёно перенаправление смарт-карт.
8. Убедитесь, что в настройках десктоп-клиента включено перенаправление смарт-карт.
9. Аутентифицируйтесь в десктоп-клиенте по смарт-карте после соответствующих настроек.

Глава 17. Настройка печати

Терминальный сервер на ОС Linux

Ниже описано, как настроить печать на терминальных серверах на ОС Linux.

На РЕД ОС в конфигурационном файле `/etc/security/limits.conf` замените «-» на «soft»:

- Проверьте, что на терминальном сервере установлены x2goserver-printing, cups-x2go из репозитория.
- Проверьте, что на терминальном сервере в утилите добавлен принтер, у которого в названии содержится «CUPS-X2GO», без дополнительных настроек.
- Проверьте версию Ghostscript с помощью команды:

```
ghostscript -v
```

Если версия 9.54 и выше, то необходимо настроить печать. Для этого:

1. Откройте в редакторе файл `/etc/cups/cups-x2go.conf` с помощью команды:

   ```
   nano /etc/cups/cups-x2go.conf
   ```

2. Добавьте в конец файла строку:

   Кроме РЕД ОС 8.

   ```
   ps2pdf=/usr/bin/gs -q -dCompatibilityLevel=1.4 -dNOPAUSE -dBATCH -dSAFER -sDEVICE=pdfwrite
   -sOutputFile=- -dAutoRotatePages=/PageByPage -dAutoFilterColorImages=false
   -dColorImageFilter=/FlateEncode -dPDFSETTINGS=/prepress -dDoNumCopies -c 3000000
   setvmthreshold -f %s
   ```

3. Перезапустите CUPS:

   ```
   systemctl restart cups
   ```

Печать через десктоп-клиент можно настроить через инструмент для просмотра PDF-файлов. Для этого проверьте, что установлен инструмент для просмотра PDF-файлов (работает с Foxit PDF Reader, Adobe Acrobat Reader), который выбран программой по умолчанию для открытия файла в формате `*.pdf`.

Глава 18. Деинсталляция компонентов

Деинсталляция компонентов Termit включает в себя удаление:

- брокера;
- агента;
- клиента.

Удаление брокера

Чтобы удалить брокер:

1. Перейдите в каталог, куда был распакован дистрибутив.
2. Запустите скрипт с параметром `uninstall` с помощью команды:

```
sudo ./broker-installer uninstall
```

При деинсталляции брокера база данных удаляется вручную. Повторное использование базы данных для новой инсталляции невозможно.

Удаление агента

Чтобы удалить агент:

1. На портале администрирования, в левом меню выберите раздел **Серверы**.
2. Наведите курсор на нужный сервер и нажмите на FQDN адрес сервера.
3. В правом верхнем углу нажмите **[Дополнительно]**.
4. Выберите **Удалить агент**.
5. Скопируйте актуальную команду На портале администрирования или предложенную ниже:

```
sudo ./agent-installer uninstall
```

6. Перейдите в каталог, где был запущен скрипт установки агента.
7. Выполните скопированную команду в терминале.

Агент удален на ТС.

Удаление десктоп-клиента

РЕД ОС

Чтобы удалить десктоп-клиент, введите команду:

```
sudo dnf remove termit-desktop
```

Astra Linux

Чтобы удалить десктоп-клиент, введите команду:

```
sudo apt remove termit-desktop
```

ALT Linux

Чтобы удалить десктоп-клиент, введите команду:

```
sudo apt-get remove termit-desktop
```