



Программный комплекс «Ресурсная карта Web»

Руководство по установке

Версия от 27.07.2026

Оглавление

1. Установка операционной системы	3
1.1. РЕД ОС.....	3
2. Установка .NET	3
3. Установка СУБД.....	3
3.1. Postgres Pro.....	3
4. Установка программного комплекса	6
4.1. Базы данных.....	6
4.2. Приложение	7
Запуск .NET приложения.....	7
Аутентификация.....	8

1. Установка операционной системы

Программный комплекс может работать на разных операционных системах: Windows, РЕД ОС, Astra Linux, Alt Linux.

1.1. РЕД ОС

Образ системы (РЕД ОС 8 | Ядро Linux 6.6) для частных пользователей можно скачать с [сайта](#) компании РЕД СОФТ. При этом Iso включает конфигурации как для рабочей станции, так и для сервера. После установки рекомендуется установить все требуемые системой обновления.

2. Установка .NET

.NET (ранее известна как .NET Core) — бесплатная модульная платформа с открытым исходным кодом для разработки программного обеспечения. В рамках текущей статьи будет рассмотрена установка .NET в составе пакета dotnet-sdk, т.к. он включает в себя сразу всю платформу. На данный момент используется версия 10.0. Описание установки может затрагивать предыдущие версии платформы (в данном случае версию 8.0).

Перед началом установки перейдите в сеанс пользователя **root**:

```
[admin@localhost ~]$ su -
```

После ввода пароля мы перейдем в пользователя **root**:

```
[root@localhost ~]#
```

Далее выполните команду проверки обновлений:

```
[root@localhost ~]# dnf update
```

После этого можно выполнить команду установки платформы:

```
[root@localhost ~]# dnf install dotnet-sdk-8.0
```

Для проверки установленных SDK воспользуйтесь следующей командой:

```
[root@localhost ~]# dotnet --list-sdks  
8.0.110 [/usr/lib64/dotnet/sdk]
```

Для проверки установленных сред выполнения воспользуйтесь командой:

```
[root@localhost ~]# dotnet --list-runtimes  
Microsoft.AspNetCore.App 8.0.10 [/usr/lib64/dotnet/shared/Microsoft.AspNetCore.App]  
Microsoft.NETCore.App 8.0.10 [/usr/lib64/dotnet/shared/Microsoft.NETCore.App]
```

3. Установка СУБД

3.1. Postgres Pro

Подробную инструкцию по установке Postgres Pro Standard 17 под разные операционные системы вы можете найти на [сайте](#) производителя.

Для начала нужно добавить репозиторий в систему:

```
[root@localhost ~]# wget https://repo.postgrespro.ru/std/std-17/keys/pgpro-repo-add.sh
```

```
[root@localhost ~]# sh pgpro-repo-add.sh
```

Если продукт единственный Postgres на вашей машине, и вы хотите сразу получить готовую к употреблению базу:

```
[root@localhost ~]# yum install postgrespro-std-17
```

Проверяем успешность установки PostgreSQL:

```
[root@localhost ~]# systemctl status postgrespro-std-17.service
● postgrespro-std-17.service - Postgres Pro std 17 database server
   Loaded: loaded (/usr/lib/systemd/system/postgrespro-std-17.service; enable>
   Active: active (running) since Wed 2025-01-15 16:27:10 MSK; 9min ago
     Main PID: 2959916 (postgres)
       Tasks: 7 (limit: 4624)
      Memory: 49.7M
         CPU: 301ms
    CGroup: /system.slice/postgrespro-std-17.service
            └─2959916 /opt/pgpro/std-17/bin/postgres -D /var/lib/pgpro/std-17/>
              └─2959917 "postgres: logger "
                └─2959918 "postgres: checkpointer "
                  └─2959919 "postgres: background writer "
                    └─2959921 "postgres: walwriter "
                      └─2959922 "postgres: autovacuum launcher "
                        └─2959923 "postgres: logical replication launcher "
```

Для просмотра списка баз данных и задания пароля к автоматически созданной учетной записи postgres необходимо войти в Postgres Pro Standard 17 под учетной записью postgres (из учетной записи root):

```
[root@localhost ~]# su - postgres
[postgres@localhost ~]$
```

Далее запускаем терминальный клиент для работы с Postgres:

```
[postgres@localhost ~]$ psql
psql (17.2)
```

Посмотреть список баз данных можно так:

```
postgres=# \l
          Список баз данных
CU | Имя | Владелец | Кодировка | Провайдер локали | LC_COLLATE | LC_CTYPE | Локаль | Правила I
-----+-----+-----+-----+-----+-----+-----+-----+-----
postgres | postgres | UTF8 | icu | ru_RU.UTF-8 | ru_RU.UTF-8 | ru-RU |
template0 | postgres | UTF8 | icu | ru_RU.UTF-8 | ru_RU.UTF-8 | ru-RU |
  =c/postgres |  +
  |  |
  | postgres=CtC/postgres
template1 | postgres | UTF8 | icu | ru_RU.UTF-8 | ru_RU.UTF-8 | ru-RU |
  =c/postgres |  +
  |  |
  | postgres=CtC/postgres
(3 строки)
```

Для задания или смены пароля выполните следующую команду:

```
postgres=# \password
```

Выйти из терминального клиента можно:

```
postgres=# \q
```

На этом установка PostgreSQL Pro окончена и необходима его настройка. По умолчанию Postgres Pro работают только с локальными подключениями, поэтому после установки необходимо настроить удаленный доступ. Файлы конфигурации расположены в папке, которая отличается для разных операционных систем и версий. Для внесения изменений в настройки откройте в текстовом редакторе файл конфигурации (не забывайте перейти обратно в сеанс пользователя **root**):

```
[postgres@localhost ~]$ su -
```

```
[root@localhost ~]# nano /var/lib/pgpro/std-17/data/pg_hba.conf
```

Откроется файл вида:

```
# 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
# IPv6 local connections:
host       all          all          ::1/128      md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local      replication  all              peer
host       replication  all          127.0.0.1/32  md5
host       replication  all          ::1/128      md5
```

Отредактируйте строку, указав вместо 127.0.0.1/32 адресное пространство локальной сети организации, например, 192.168.1.1/24. Если нужно открыть для всей сети, укажите 0.0.0.0/0. Получится так:

```

GNU nano 7.2 /var/lib/pgpro/std-17/data/pg_hba.conf
# listen on a non-local interface via the listen_addresses
# configuration parameter, or via the -i or -h command line switches.

# TYPE      DATABASE      USER      ADDRESS      METHOD

# "local" is for Unix domain socket connections only
local      all             all                                     peer
# IPv4 local connections:
host       all             all        0.0.0.0/0     md5
# IPv6 local connections:
host       all             all        ::1/128      md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local      replication     all                                     peer
host       replication     all        127.0.0.1/32  md5
host       replication     all        ::1/128      md5

```

Сохраняем файл.

Далее откройте в текстовом редакторе файл конфигурации postgresql.conf (Он лежит по тому же пути).

```
[root@localhost ~]# nano /var/lib/pgpro/std-17/data/postgresql.conf
```

Найдите параметр `listen_addresses` и убедитесь, что он имеет следующий вид: **`listen_addresses = '*' # what IP address(es) to listen on;`**

Если вид отличается, то приведите к указанной форме, установив значение равное `'*'`.

```

# - 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)

```

Сохраняем файл и перезагружаем сервис:

```
[root@localhost ~]# systemctl restart postgrespro-std-17.service
```

4. Установка программного комплекса

4.1. Базы данных

Для администрирования и разработки баз данных PostgreSQL применяется программный продукт pgAdmin4. Он позволяет выполнять задачи мониторинга, обслуживания, конфигурирования сервера PostgreSQL, а также создавать и выполнять SQL-запросы.

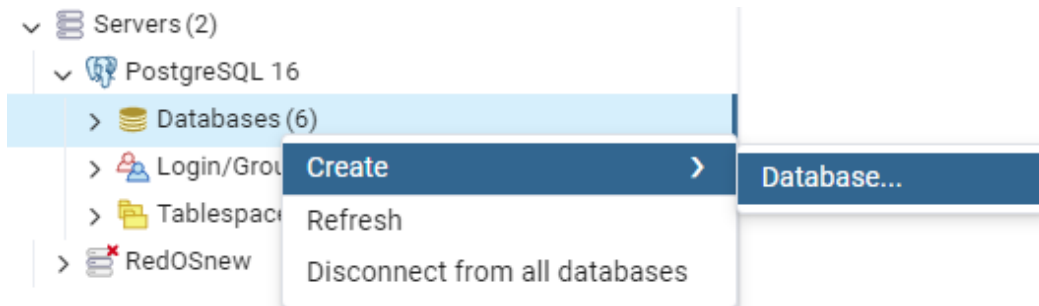
Для установки pgAdmin4 перейдите в сеанс пользователя root (командой `su -`) и выполните команду:

```
[root@localhost ~]# dnf install pgadmin4 pgadmin4-qt
```

Программный комплекс «Ресурсная карта Web» использует несколько баз данных:

- Customs – основная база данных с ключевыми справочниками
- CustomsFiles – база данных с прикрепленными файлами
- CustomsMain – база данных с общими справочниками

Нам необходимо эти базы данных создать через pgAdmin4:



Далее нужно запросить у технической поддержки [дистрибьютора](#) набор скриптов для наполнения баз данных таблицами и первичными данными. Для каждой базы данных будет свой индивидуальный скрипт. Не забудьте перед выполнением скрипта проверить к какой базе вы подключены.

4.2. Приложение

Чтобы развернуть программный комплекс «Ресурсная карта Web» вам необходимо запросить у технической поддержки [дистрибьютора](#) дистрибутив приложения в виде архива с файлами. Скачиваем и извлекаем данные из архива в заранее созданный каталог. Например, каталог (в своих документах) customs. Все что вам остается, это скорректировать файл appsettings.json. В нем необходимо изменить параметры подключения к основной базе данных (как правило это база Customs).

Запуск .NET приложения

Если нет необходимости использования веб серверов, например, Nginx или Apache, то можно запустить приложение непосредственно через платформу .NET.

Для это вам необходимо перейти в каталог с приложением и запустить исполняемый файл dll:

- cd /home/admin/customs/
- dotnet CustomsWeb.dll

```
[admin@localhost ~]$ cd /home/admin/customs/
[admin@localhost customs]$ dotnet CustomsWeb.dll
warn: Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager[35]
      No XML encryptor configured. Key {5e684d9f-21a9-46b7-8c14-a466e057593b} may be persisted to storage in unencrypted form.
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: /home/admin/customs
```

Далее, как указано в терминале, нужно в браузере перейти по адресу <http://localhost:5000>

Аутентификация

Для аутентификации приложения используется OpenID Connect (OIDC) протокол. используется для аутентификации веб-приложений. OIDC позволяет приложению доверять другой системе — поставщику удостоверений (OIDC IdP) — для обработки входов и подтверждения того, кто является пользователем. В нашем случае таким приложением является [KEYCLOAK](#). Он должен быть установлен и настроен у вас.

В самом приложении за взаимодействие с поставщиком отвечает раздел файла конфигурации (appsettings.json) OpenIDConnectSettings.

```
"OpenIDConnectSettings": {  
  "Realm": "myrealm",  
  "ClientId": "rescard",  
  "AuthServerUrl": "http://localhost:8080/",  
  "HttpsRequired": false,  
  "Secret": "5G9IVdRQjL5k9A0LBkb13vFgAYqZTeEp"  
},
```

Где (все эти параметры настроены соответствующим образом в KEYCLOAK):

- Realm - область аутентификации
- ClientId - уникальный идентификатор, присвоенный приложению сервером авторизации при регистрации
- AuthServerUrl - базовый URL сервера авторизации
- HtpsRequired - требует использования протокола HTTPS
- Secret - секрет, который используется в контексте аутентификации

В общих чертах при запуске приложения происходит переход на страницу сервера авторизации (через настройку AuthServerUrl), где пользователь должен авторизоваться, например, через введение логина и пароля (или с помощью учетной записи домена). При успешной авторизации происходит переход на страницу приложения.