

Руководство администратора

Программный комплекс "PRO.Отчеты"

Серверная инфраструктура

Версия: 3.1.2

Разработано при поддержке Фонда содействия инновациям



PRO.Отчеты

ООО "ИнРиС"

394006, г. Воронеж,
ул. 20-летия Октября, д. 103

8 (473) 239-96-93

support@inris.ru

Аннотация

Настоящий документ представляет собой руководство администратора Программного комплекса **PRO.Отчеты**.

В руководстве администратора описаны общие принципы настройки **PRO.Отчеты** на условия эксплуатации и порядок действий администратора при выполнении операций. Функционал **PRO.Отчеты** может быть ограничен в зависимости от редакции, лицензии или прав пользователя.

Руководство администратора основано на том предположении, что вся необходимая инфраструктура для функционирования **PRO.Отчеты** развернута и готова к использованию. Сетевой адрес устройства, на котором разворачивается **сервер базы данных** доступен с **сервера приложений**, а **сервер приложений** доступен по назначенному адресу **на устройствах пользователей**.

Приведенная в документе информация актуальна для **PRO.Отчеты** версии 3.1.2.

Содержание

1	Общие сведения.....	4
1.1	Назначение программы	4
1.2	Минимальный состав технических средств	4
1.3	Минимальный состав программных средств	5
2	Структура программы	6
2.1	Серверная часть	Ошибка! Закладка не определена.
2.2	Клиентская часть	Ошибка! Закладка не определена.
2.3	Архитектурная модель	Ошибка! Закладка не определена.
2.4	Структура PRO.Отчеты	Ошибка! Закладка не определена.
3	Установка и настройка PRO.Отчеты	8
3.1	Установка и настройка PRO.Отчеты на сертифицированную ОС	8
3.2	Установка и настройка PRO.Отчеты на прочие ОС Linux	9
3.2.1	Авторизация и подготовительные мероприятия	9
3.2.2	Обновление и установка компонентов и приложений	11
3.2.3	Настройка компонентов на условия эксплуатации	13
3.2.4	Завершающие операции	15
4	Проверка программы	17

1 Общие сведения

1.1 Назначение программы

Программный комплекс **PRO.Отчеты** (далее - **PRO.Отчеты**) обеспечивает автоматизацию процессов сбора данных, подготовки и анализа отчетов различной степени сложности в организациях любой сферы деятельности с простой или сложной (иерархической) структурой.

PRO.Отчеты представляет собой веб-приложение, взаимодействие с которым осуществляется при помощи браузера.

Для корректной работы **PRO.Отчеты** рекомендуется использовать последнюю версию современного браузера.

1.2 Минимальный состав технических средств

ПК **PRO.Отчеты** корректно функционирует на аппаратных платформах со следующими минимальными характеристиками:

1. Для серверной части:

- Процессор – не менее 1 процессора 4 ядра по 3,4 ГГц;
- ОЗУ – не менее 8 Гб;
- ЖМД – не менее 120 Гб;
-
- Сетевой адаптер – не менее 100 Мбит/с;
- Клавиатура USB или беспроводная;
- Манипулятор «мышь».

2. Для сетевой станции (клиентской части):

- Процессор – не менее 2.3 ГГц;
- ОЗУ – не менее 4 Гб;
- ЖМД – не менее 120 Гб;
- Сетевой адаптер – не менее 100 Мбит/с;
- монитор с минимальным разрешением 1280x768 пикселей;
- Клавиатура USB или беспроводная;
- Манипулятор «мышь» USB или беспроводной.

1.3 Минимальный состав программных средств

ПК **PRO.Отчеты**, функционирует под управлением следующих операционных систем (ОС) и программных средств:

1. Серверная часть:

а. сертифицированные ОС:

- Альт Сервер 10 и выше;
- Astra Linux 1.8 и выше;
- Ред ОС Сервер 7.3 и выше;
- Ред ОС Сервер 8 и выше.

б. прочие ОС:

- Debian 11 – 12 и выше;
- Ubuntu 20.04 – 24.04 и выше;
- Microsoft Windows Server 2012;
- Microsoft Windows Server 2016;
- Microsoft Windows Server 2019;

2. Клиентская часть: Современный браузер.

3. Дополнительные программные средства, устанавливаемые из репозитория в ходе развертывания, либо входящие в установочный комплект:

- libpq5;
- openssl;
- postgresql15-server;
- libgdipplus;
- nginx;
- dotnet-sdk-7.0.

4. Дополнительные права и полномочия пользователя назначаются на следующие файлы и каталоги:

```
install.sh
```

```
/dotnet
```

```
/var/www/html/front
```

```
/var/www/html/back
```

2 Архитектура программы

PRO.Отчеты представляет собой кроссплатформенное веб-приложение, состоящее из веб-клиента и сервера, взаимодействие между которыми осуществляется при помощи сетевых HTTP-протоколов посредством REST API.

Система **PRO.Отчеты** включает в себя:

- **Frontend** – веб-приложение, реализованное в виде Vue приложения для браузеров. Frontend взаимодействует с сервером Системы по протоколу HTTP/HTTPS с использованием тела запроса в формате JSON.
- **Балансировщик нагрузки (HTTP Proxy)** маршрутизирует и балансирует запросы Front-end, а также предоставляет статический контент для Front-end. Балансировщик нагрузки представляет из себя сервер Nginx.
- Веб-сервисы:
 - **Authenticate service** – обеспечивает идентификацию и аутентификацию пользователя в Системе;
 - **Manage service** – отвечает за настройку конфигурации отчетов, шаблоны печатных форм, импорт данных из внешних источников, а также управление учетными записями и правами доступа пользователей;
 - **Workspace service** – отвечает за работу пользователя с отчетами;
 - **Background service** – представляет собой набор компонентов и утилит, отвечающих за выполнения расчетов документов.
- СУБД с базами данных:
 - **глобальная база данных**, хранящая информацию о пользователях и обеспечивающая защищенность **PRO.Отчеты** от несанкционированного доступа к информации;
 - **клиентские базы данных**, обеспечивающие распределенный доступ к информации каждого клиента.

С точки зрения пользователей архитектура **PRO.Отчеты** реализована по трехзвенной клиент-серверной архитектуре, состоящей из клиента, сервера приложений и сервера баз данных.

Логическая структура **PRO.Отчеты** содержит три уровня:

- слой представления (клиент) обеспечивает представление **PRO.Отчеты** и результатов, понятных конечному пользователю. Клиент представляет данные пользователю, а также обеспечивает возможность ввода и изменения данных при наличии у пользователя соответствующих прав. Данный слой включает компоненты пользовательского интерфейса и не имеет прямого доступа к базе данным;
- слой бизнес-логики (сервер приложений) представлен сервисом исполнения бизнес-логики, который координирует **PRO.Отчеты**, обрабатывает запросы, выполняет логические решения и вычисления. Компоненты выполняются на сервере и разделяют общие ресурсы сервера;
- слой данных (сервер базы данных) обеспечивает долговременное эффективное хранение данных, хранит модели данных, используемых сущностей и рамках бизнес-логики приложения, отвечает за мониторинг транзакций и поддерживает консистентное состояние данных. Уровень хранения включает СУБД и компоненты для доступа к данным. Уровень хранения предоставляет программный интерфейс для объектов уровня сервера приложений.

Архитектура **PRO.Отчеты** позволяет распределять нагрузку за счет возможности расположения баз данных как на одном сервере, так и на разных.

3 Установка и настройка PRO.Отчеты

3.1 Установка и настройка PRO.Отчеты на сертифицированную ОС

Для установки **PRO.Отчеты** с настройками на условия применения **по умолчанию**:

Настройки **по умолчанию** предполагают, что компоненты базы данных и сервера приложений развертываются **на одном устройстве** и после чистой установки, **сертифицированной** для работы с **PRO.Отчеты** ОС.

1. Откройте консоль, либо авторизуйтесь под именем пользователя.
2. Скопируйте архив `ProReport.tar.gz` из состава поставки **PRO.Отчеты** в домашнюю директорию пользователя.
3. Распакуйте архив командой:

```
tar -xvf ProReport.tar.gz
```

 В домашнем каталоге пользователя появится каталог `ProReport` с установочным комплектом.

4. Выполните команду:

```
su-
```

 Появится запрос авторизации.

5. Авторизуйтесь. Перейдите в каталог `/home/user/ProReport` выполнив команду:

```
cd /home/user/ProReport
```

6. Выполните команду:


```
chmod +x install.sh
```

7. Выполните команду:

```
./install.sh
```

 Начнется выполнение скрипта. Ответьте утвердительно на вопросы, которые будут задавать в процессе работы скрипта.

8. Выполните команду:

```
reboot
```

 Установка успешно завершена.

3.2 Установка и настройка PRO.Отчеты на прочие ОС Linux

Действия по установке приведены на примере ОС Astra Linux 1.8.

3.2.1 Авторизация и подготовительные мероприятия

1. Откройте консоль, либо авторизуйтесь под именем пользователя.
2. Скопируйте архив `ProReport.tar.gz` из состава поставки **PRO.Отчеты** в домашнюю директорию пользователя.
3. Распакуйте архив командой:

```
tar -xvf ProReport.tar.gz
```

 В домашнем каталоге пользователя появится каталог `ProReport` с установочным комплектом.

4. Выполните команду:

```
sudo su -
```

 Появится запрос авторизации.

5. Авторизуйтесь. Перейдите в каталог `/home/%User_Name%/ProReport/sql`

выполнив команду:

```
cd /home/%User_Name%/ProReport/sql
```

6. Для назначения пароля пользователя базы данных **postgres** в файле **0_hangfire_0.sql** в первой строке:

```
alter role postgres password 'QWErty654#@!";
```

замените значение **QWErty654#@!"** на пароль, который необходимо установить.

7. Для настройки конфигурации сервисов для работы с базой данных в файлах **appsettings.json**, находящихся в каталогах

```
/home/%User_Name%/ProReport/back/background
```

```
/home/%User_Name%/ProReport/back/identity
```

```
/home/%User_Name%/ProReport/back/workspace
```

```
/home/%User_Name%/ProReport/back/manage
```

в секции **ConnectionStrings** в свойствах **GlobalConnection**, **HangfireConnection** и **Connection** укажите адрес, порт, имя базы данных и учетные данные.

Например:

```
Host=127.0.0.1;Port=5432;Database=pr_global;Username=postgres;Password=QWErty654#@!
```

где:

- **Host=127.0.0.1** – Адрес базы данных.

В случае, если сервер базы данных развернут по другому адресу укажите его в настройках всех свойств всех сервисов.

- **Port=5432** – порт для доступа к серверу базы данных.

- Database=pr_global – имя базы данных.
- Username=postgres – имя пользователя базы данных.
- Password=QWErty654#@!" – пароль пользователя базы данных.

В случае, если вы изменили пароль в пункте 6 укажите его для корректной работы сервисов в настройках всех свойств всех сервисов.

8. Перейдите в каталог /home/%User_Name%/ProReport выполнив команду:

```
cd /home/%User_Name%/ProReport
```

3.2.2 Обновление и установка компонентов и приложений

1. Для установки сторонних компонентов необходимо подключить репозиторий установочного диска, либо включить поддержку внешних репозиториях. Для включения внешнего репозитория раскомментируйте первую строку в файле:

```
/etc/apt/sources.list
```

2. Для обновления операционной системы выполните команды:

```
apt update
```

 Дождитесь выполнения обновления.

3. Для установки библиотеки подключения к серверу баз данных **PostgreSQL** выполните команду:

```
apt install libpq5
```

 Дождитесь выполнения установки. Ответе **утвердительно** на все запросы, появляющиеся в процессе установки.

4. Для установки криптографической библиотеки **OpenSSL** выполните команду:

```
apt install openssl
```

 Дождитесь выполнения установки. Ответе **утвердительно** на все запросы, появляющиеся в процессе установки.

5. Для установки системы управления базами данных **PostgreSQL 15** выполните команду:

```
apt install postgresql-15
```

 Дождитесь выполнения установки. Ответе **утвердительно** на все запросы, появляющиеся в процессе установки.

6. Для установки библиотеки выполните команду:

```
apt install libgdiplus
```

 Дождитесь выполнения установки. Ответе **утвердительно** на все запросы, появляющиеся в процессе установки.

7. Для установки WEB-сервера **nginx** выполните команду:

```
apt install nginx
```

 WEB-сервер **nginx** будет установлен.

8. Для остановки WEB-сервера **nginx** выполните команду:

```
systemctl stop nginx
```

 Служба WEB-сервера **nginx** будет остановлена.

9. Для остановки службы системы управления базами данных **PostgreSQL 15** выполните команду:

```
systemctl stop postgresql
```

 Служба системы управления базами данных **PostgreSQL 15** будет остановлена.

3.2.3 Настройка компонентов на условия эксплуатации

1. Отредактируйте файл:

```
/etc/postgresql/15/main/postgresql.conf
```

Включите приведенные в Таблица 1 параметры и установите им следующие значения:

Таблица 1 Параметры и значения настроек PostgreSQL

Параметр	Значение
listen_addresses	'*'
wal_level	hot_standby
wal_log_hints	on
max_wal_senders	2
max_replication_slots	1
hot_standby	on
track_counts	on
autovacuum	on
autovacuum_max_workers	3
autovacuum_naptime	30min
autovacuum_vacuum_threshold	50
autovacuum_analyze_threshold	50
autovacuum_vacuum_scale_factor	0.2
autovacuum_analyze_scale_factor	0.1
Timezone	Europe/Moscow

2. Отредактируйте файл:

```
/etc/postgresql/15/main/pg_hba.conf
```

добавьте в конец параметры:

```
host replication postgres all md5
```

```
host all postgres all md5
```

ПРИМЕЧАНИЕ: Отредактированные файлы `pg_hba.conf` и `postgresql.conf` доступны в каталоге:

```
/home/%User_Name%/ProReport/pf_conf
```

и **могут** быть скопированы в место назначения командой:

```
yes | cp -rf /home/%User_Name%/ProReport/pf_conf/*  
/etc/postgresql/15/main/
```

3. Перейдите в каталог `/home/%User_Name%/ProReport` выполнив команду:

```
cd /home/%User_Name%/ProReport
```

4. Скопируйте каталог `/home/%User_Name%/ProReport/dotnet` в корневой каталог. Для этого выполните команду:

```
cp -rf -t / /home/%User_Name%/ProReport/dotnet
```

5. Скопируйте каталог `/home/%User_Name%/ProReport/back/` в каталог `/var/www/html/`. Для этого выполните команду:

```
cp -rf -t /var/www/html/ /home/%User_Name%/ProReport/back/
```

6. Скопируйте каталог `/home/%User_Name%/ProReport/front/` в каталог `/var/www/html/`. Для этого выполните команду:

```
cp -rf -t /var/www/html/ /home/%User_Name%/ProReport/front/
```

7. Скопируйте содержимое каталога `/ProReport/services/` в каталог `/etc/systemd/system`. Для этого выполните команду:

```
yes | cp -rf /home/%User_Name%/ProReport/services/* /etc/systemd/system/
```

8. Скопируйте содержимое каталога `/home/%User_Name%/ProReport/sites-enabled.d` в каталог `/etc/nginx/sites-enabled`. Для этого выполните команду:

```
yes | cp -rf /home/%User_Name%/ProReport/sites-enabled/* /etc/nginx/sites-enabled/
```

9. Для предоставления прав доступа к директориям выполните команды:

```
chmod -R 777 /dotnet  
  
chmod -R 777 /var/www/html/front  
  
chmod -R 777 /var/www/html/back
```

10. Для запуска службы системы управления базами данных **PostgreSQL 15** выполните команду:

```
service postgresql start
```

11. Для создания и наполнения баз данных первичными сведениями выполните команды:

```
psql -U postgres -f sql/0_hangfire_0.sql  
  
psql --dbname=pr_global --username=postgres -f sql/1_pr_global_0.sql  
  
psql --dbname=pr_workspace_1 --username=postgres -f  
sql/2_pr_workspace_1_0.sql
```

3.2.4 Завершающие операции

1. Для включения сервисов выполните команды:

```
systemctl enable postgresql  
  
systemctl enable nginx  
  
systemctl enable background  
  
systemctl enable identity  
  
systemctl enable manage  
  
systemctl enable workspace
```

2. Для запуска сервисов выполните команды:

```
systemctl start nginx  
  
systemctl start workspace  
  
systemctl start background  
  
systemctl start identity  
  
systemctl start manage
```

3. Для перезагрузки выполните команду:

```
reboot
```



Установка успешно завершена.

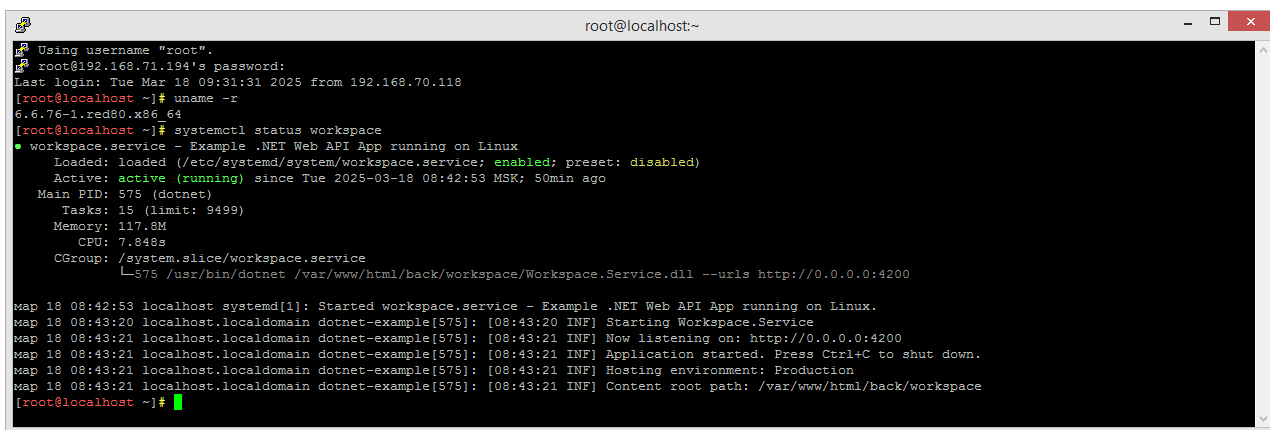
4 Проверка программы

Для проверки работоспособности **PRO.Отчеты** выполните следующие команды:

1. Откройте консоль, либо авторизуйтесь.
2. Выполните команду:

```
systemctl status workspace
```

 Отсутствие ошибок свидетельствует об успешном развертывании сервиса.



```
root@localhost:~  
Using username "root".  
root@192.168.71.194's password:  
Last login: Tue Mar 18 09:31:31 2025 from 192.168.70.118  
[root@localhost ~]# uname -r  
6.6.76-1.redhat8.x86_64  
[root@localhost ~]# systemctl status workspace  
● workspace.service - Example .NET Web API App running on Linux  
   Loaded: loaded (/etc/systemd/system/workspace.service; enabled; preset: disabled)  
   Active: active (running) since Tue 2025-03-18 08:42:53 MSK; 50min ago  
     Main PID: 575 (dotnet)  
       Tasks: 15 (limit: 9499)  
      Memory: 117.8M  
         CPU: 7.848s  
    CGroup: /system.slice/workspace.service  
            └─575 /usr/bin/dotnet /var/www/html/back/workspace/Workspace.Service.dll --urls http://0.0.0.0:4200  
  
map 18 08:42:53 localhost systemd[1]: Started workspace.service - Example .NET Web API App running on Linux.  
map 18 08:43:20 localhost.localdomain dotnet-example[575]: [08:43:20 INF] Starting Workspace.Service  
map 18 08:43:21 localhost.localdomain dotnet-example[575]: [08:43:21 INF] Now listening on: http://0.0.0.0:4200  
map 18 08:43:21 localhost.localdomain dotnet-example[575]: [08:43:21 INF] Application started. Press Ctrl+C to shut down.  
map 18 08:43:21 localhost.localdomain dotnet-example[575]: [08:43:21 INF] Hosting environment: Production  
map 18 08:43:21 localhost.localdomain dotnet-example[575]: [08:43:21 INF] Content root path: /var/www/html/back/workspace  
[root@localhost ~]#
```

Рисунок 1 Результат выполнения команды `systemctl status workspace`.

3. Выполните команду:

```
systemctl status background
```

 Отсутствие ошибок свидетельствует о успешном развертывании сервиса.

```
root@localhost:~  
[root@localhost ~]# systemctl status background  
● background.service - Example .NET Web API App running on Linux  
   Loaded: loaded (/etc/systemd/system/background.service; enabled; preset: disabled)  
   Active: active (running) since Tue 2025-03-18 08:42:52 MSK; 53min ago  
     Main PID: 550 (dotnet)  
        Tasks: 124 (limit: 9499)  
       Memory: 91.6M  
          CPU: 28.338s  
     CGroup: /system.slice/background.service  
             └─550 /usr/bin/dotnet /var/www/html/back/background/BackgroundJobs.Service.dll --urls http://0.0.0.0:5005  
  
map 18 08:43:14 localhost.localdomain dotnet-example[550]:      Shutdown timeout: 00:02:00  
map 18 08:43:14 localhost.localdomain dotnet-example[550]:      Schedule polling interval: 00:00:15  
map 18 08:43:14 localhost.localdomain dotnet-example[550]: [08:43:14 INF] Application started. Press Ctrl+C to shut down.  
map 18 08:43:14 localhost.localdomain dotnet-example[550]: [08:43:14 INF] Hosting environment: Production  
map 18 08:43:14 localhost.localdomain dotnet-example[550]: [08:43:14 INF] Content root path: /var/www/html/back/background  
map 18 08:43:15 localhost.localdomain dotnet-example[550]: [08:43:15 INF] Server localhost:550:a59000f9 successfully announced in 884,5814 ms  
map 18 08:43:15 localhost.localdomain dotnet-example[550]: [08:43:15 INF] Server localhost:550:a59000f9 is starting the registered dispatchers: ServerWatch  
map 18 08:43:15 localhost.localdomain dotnet-example[550]: [08:43:15 INF] 1 servers were removed due to timeout  
map 18 08:43:16 localhost.localdomain dotnet-example[550]: [08:43:16 INF] Server localhost:550:a59000f9 all the dispatchers started  
map 18 08:43:34 localhost.localdomain dotnet-example[550]: [08:43:34 INF] Execution DelayedJobScheduler recovered from the Faulted state after 00:00:17.00  
[root@localhost ~]#
```

Рисунок 2 Результат выполнения команды `systemctl status background`.

4. Выполните команду:

```
systemctl status identity
```

➡ Отсутствие ошибок свидетельствует о успешном разворачивании сервиса.

```
root@localhost:~  
[root@localhost ~]# systemctl status identity  
● identity.service - Example .NET Web API App running on Linux  
   Loaded: loaded (/etc/systemd/system/identity.service; enabled; preset: disabled)  
   Active: active (running) since Tue 2025-03-18 08:42:52 MSK; 53min ago  
     Main PID: 553 (dotnet)  
        Tasks: 13 (limit: 9499)  
       Memory: 110.1M  
          CPU: 6.948s  
     CGroup: /system.slice/identity.service  
             └─553 /usr/bin/dotnet /var/www/html/back/identity/Identity.Service.dll --urls http://0.0.0.0:4100  
  
map 18 08:42:52 localhost systemd[1]: Started identity.service - Example .NET Web API App running on Linux.  
map 18 08:43:06 localhost.localdomain dotnet-example[553]: [08:43:06 INF] Starting Identity.Service  
map 18 08:43:09 localhost.localdomain dotnet-example[553]: [08:43:09 INF] Now listening on: http://0.0.0.0:4100  
map 18 08:43:09 localhost.localdomain dotnet-example[553]: [08:43:09 INF] Application started. Press Ctrl+C to shut down.  
map 18 08:43:09 localhost.localdomain dotnet-example[553]: [08:43:09 INF] Hosting environment: Production  
map 18 08:43:09 localhost.localdomain dotnet-example[553]: [08:43:09 INF] Content root path: /var/www/html/back/identity  
[root@localhost ~]#
```

Рисунок 3 Результат выполнения команды `systemctl status identity`.

5. Выполните команду:

```
systemctl status manage
```

➡ Отсутствие ошибок свидетельствует о успешном разворачивании сервиса.

```
root@localhost:~  
[root@localhost ~]# systemctl status manage  
● manage.service - Example .NET Web API App running on Linux  
   Loaded: loaded (/etc/systemd/system/manage.service; enabled; preset: disabled)  
   Active: active (running) since Tue 2025-03-18 08:43:34 MSK; 53min ago  
     Main PID: 1750 (dotnet)  
        Tasks: 13 (limit: 9499)  
       Memory: 41.9M  
          CPU: 1.683s  
     CGroup: /system.slice/manage.service  
             └─1750 /usr/bin/dotnet /var/www/html/back/manage/Manage.Service.dll --urls http://0.0.0.0:4300  
  
map 18 08:43:34 localhost.localdomain systemd[1]: manage.service: Consumed 1.407s CPU time.  
map 18 08:43:34 localhost.localdomain systemd[1]: Started manage.service - Example .NET Web API App running on Linux.  
map 18 08:43:35 localhost.localdomain dotnet-example[1750]: [08:43:35 INF] Starting Manage.Service  
map 18 08:43:35 localhost.localdomain dotnet-example[1750]: [08:43:35 INF] Now listening on: http://0.0.0.0:4300  
map 18 08:43:35 localhost.localdomain dotnet-example[1750]: [08:43:35 INF] Application started. Press Ctrl+C to shut down.  
map 18 08:43:35 localhost.localdomain dotnet-example[1750]: [08:43:35 INF] Hosting environment: Production  
map 18 08:43:35 localhost.localdomain dotnet-example[1750]: [08:43:35 INF] Content root path: /var/www/html/back/manage  
[root@localhost ~]#
```

Рисунок 4 Результат выполнения команды `systemctl status manage`.

6. Успешное выполнение команд и отсутствие ошибок свидетельствует, что настройка на условия применения прошла успешно.