

# Сервер приложений .NET Core на Linux

Версия 8.0



Эта документация предоставляется с ограничениями на использование и защищена законами об интеллектуальной собственности. За исключением случаев, прямо разрешенных в вашем лицензионном соглашении или разрешенных законом, вы не можете использовать, копировать, воспроизводить, переводить, транслировать, изменять, лицензировать, передавать, распространять, демонстрировать, выполнять, публиковать или отображать любую часть в любой форме или посредством любые значения. Обратный инжиниринг, дизассемблирование или декомпиляция этой документации, если это не требуется по закону для взаимодействия, запрещены.

Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления и не может гарантировать отсутствие ошибок. Если вы обнаружите какие-либо ошибки, сообщите нам о них в письменной форме.

# Содержание

<b>Подготовка к установке Creatio .NET Core на Linux</b>	<b>4</b>
Подготовить установочные файлы	4
Настроить ConnectionStrings.config	4
<b>Установить Creatio .NET Core на Linux</b>	<b>6</b>
Способ 1. Установить Creatio .NET Core непосредственно на Linux	6
Способ 2. Установить Creatio .NET Core на Linux с использованием Docker	8
Настроить работу Creatio .NetCore на https	11

# Подготовка к установке Creatio .NET Core на Linux

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

## Подготовить установочные файлы

Файлы, необходимые для установки Creatio, находятся в архиве. Чтобы распаковать архив, воспользуйтесь командой:

```
unzip -d /path/to/application/directory/ CREATIO_ARCHIVE_NAME.zip
```

**/path/to/application/directory/** — путь к папке, в которую необходимо распаковать архив. Рекомендуем распаковывать архив с установочными файлами в пустую папку:

```
unzip creatio_archive_name.zip
```

**Creatio\_archive\_name.zip** — название архива с установочными файлами Creatio.

**На заметку.** Если при выполнении команды `unzip` возникает ошибка "command not found", то вам необходимо установить пакет распаковки архивов:

```
sudo apt-get install unzip
```

## Настроить ConnectionStrings.config

В файле `ConnectionStrings.config`, который находится в корневой папке сайта Creatio, содержатся параметры подключения вашего приложения к базе данных и внешним сервисам.

## Отредактировать файл ConnectionStrings.config

1. Перейдите в корневой каталог сайта Creatio **~\WebAppRoot\Creatio**.
2. Откройте в текстовом редакторе файл `ConnectionStrings.config`.
3. Укажите параметры подключения **connectionStrings** вашего сайта.

**Пример настроенного файла ConnectionStrings.config**

```
<?xml version="1.0" encoding="utf-8"?>

<connectionStrings>

<add name="db" connectionString="Server=[ Имя сервера базы данных ];Port=31436;Database=[ Имя бд ];password=[ Пароль администратора ];user=[ Имя пользователя ];" />

<add name="redis" connectionString="host=[ Имя компьютера ];db=[ Номер БД Redis ];port=6379" />

<add name="tempDirectoryPath" connectionString="%TEMP%/%USER%/%APPLICATION%" />

<add name="influx" connectionString="url=[ Адрес сервиса сбора метрик сайта ]; user=; password=; " />

<add name="clientPerformanceLoggerServiceUri" connectionString="[ Адрес сервиса логгирования ]" />

<add name="messageBroker" connectionString="amqp://[ Имя пользователя MessageBroker ]:[ Пароль администратора ]@" />

</connectionStrings>
```

## Обязательные настройки ConnectionStrings.config

Параметры подключения к базе данных и серверу кэширования необходимы для работы приложения Creatio.

- **db** — отвечает за соединение с базой данных. В нем настраивается путь к базе данных, с которой необходимо установить связь, и способ авторизации на сервере базы данных.

```
<add name="db" connectionString="Server=[ Имя сервера базы данных ];Port=31436;Database=[ Имя бд ];password=[ Пароль администратора ];user=[ Имя пользователя ];" />
```

- **redis** — отвечает за взаимодействие с сервером Redis:

```
<add name="redis" connectionString="host=[ Имя компьютера ];db=[ Номер БД Redis ];port=6379;" />
```

**Важно.** Номер базы данных Redis должен быть уникальным для каждого сайта приложения.

## Необязательные настройки ConnectionStrings.config

Параметры подключения к внешним сервисам опциональны, они заполняются только в том случае, когда необходимо настроить интеграцию, например, для сбора метрик сайта.

- **tempDirectoryPath** — путь к временной директории, используемой механизмом установки пакетов:

```
<add name="tempDirectoryPath" connectionString=[ Путь к временной директории, используемой механизмом установки пакетов ] />
```

- **influx** — отвечает за взаимодействие с сервисом сбора метрик сайта. Данный параметр заполняется только в том случае, когда необходимо собрать статистику использования функциональности для последующей отладки.

```
<add name="influx" connectionString="url=[ Адрес сервиса сбора метрик сайта ]; user=[ Пользователь ];" />
```

- **clientPerformanceLoggerServiceUri** — отвечает за взаимодействие с сервисом логгирования. Данный параметр заполняется только в том случае, когда необходимо собрать информацию о загрузке страниц приложения.

```
<add name="clientPerformanceLoggerServiceUri" connectionString="[ Адрес сервиса логгирования ];" />
```

- **messageBroker** — отвечает за взаимодействие с сервисом RabbitMQ. Данный параметр заполняется только в том случае, когда необходимо настроить горизонтальное масштабирование нагрузки с помощью RabbitMQ.

```
<add name="messageBroker" connectionString="amqp://[ Имя пользователя MessageBroker ]:[ Пароль ]@" />
```

## Установить Creatio .NET Core на Linux

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Перед тем, как развернуть сервер, необходимо выполнить следующие действия:

- Подготовить установочные файлы Creatio. [Подробнее >>>](#)
- Развернуть сервер базы данных. [Подробнее >>>](#)
- Развернуть сервер кэширования данных Creatio (Redis). [Подробнее >>>](#)
- Внести изменения в файл конфигурации ConnectionStrings.config. [Подробнее >>>](#)

**На заметку.** Запуск сервера PostgreSQL в Docker описан в [документации Docker](#).

## Способ 1. Установить Creatio .NET Core непосредственно на Linux

Чтобы развернуть сервер приложения:

- Установите .NET Core, GDI+ совместимый API для семейства UNIX-подобных операционных систем, библиотеки разработки и файлы заголовков для GNU C. [Подробнее >>>](#)
- Запустите сервер приложения Creatio. [Подробнее >>>](#)

## Установить .NET Core и другие зависимости Creatio

1. Скачайте пакеты microsoft-prod package:

```
wget -q https://packages.microsoft.com/config/ubuntu/18.04/packages-microsoft-prod.deb -O pac
```

2. Войдите в систему как администратор (root):

```
sudo su
```

3. Установите загруженные пакеты:

```
dpkg -i packages-microsoft-prod.deb
```

4. Обновите список пакетов:

```
apt-get update
```

5. Установите транспорт АРТ для загрузки по протоколу HTTP Secure:

```
apt-get install apt-transport-https
```

6. Обновите список пакетов:

```
apt-get update
```

7. Установите .NET Core:

```
apt-get install dotnet-sdk-3.1
```

8. Установите GDI+ совместимый API для альтернативных Windows операционных систем:

```
apt-get install -y libgdiplus
```

9. Установите библиотеки разработки и файлы заголовков для GNU C:

```
apt-get install -y libc6-dev
```

10. Выйдите из root-сессии:

```
exit
```

## Запуск сервера приложения Creatio

**На заметку.** Если вы разворачиваете среду разработки .Net Core с доступом по протоколу http, то перед запуском приложения необходимо внести изменения в конфигурационный файл Terrasoft.WebHost.dll.config, который находится в корневой папке приложения. Для параметра "add key" укажите значение

```
<add key="CookiesSameSiteMode" value="Lax" />
```

Это обеспечит корректную работу как по протоколу https, так и по http. В тоже время при такой настройке не будет работать мобильное приложение.

Чтобы запустить приложение:

1. Перейдите в папку с установочными файлами Creatio:

```
cd /path/to/application/directory/
```

2. Запустите сервер .Net Core:

```
COMPlus_ThreadPool_ForceMinWorkerThreads=100 dotnet Terrasoft.WebHost.dll
```

Версия Creatio для HTTP будет доступна через порт 5000.

Версия Creatio для HTTPS будет доступна через порт 5002.

**На заметку.** Первый вход в систему после настройки приложения осуществляется под учетной записью Supervisor (логин — Supervisor; пароль — Supervisor).

## Способ 2. Установить Creatio .NET Core на Linux с использованием Docker

Используйте этот метод разворачивания для запуска приложения Creatio изолированно от операционной системы. Установка сервера Redis, восстановление БД и настройка файла ConnectionStrings.config выполняются аналогично, по описанной выше инструкции.

Чтобы развернуть сервер приложения с использованием Docker:



- Настройте доступ к серверу кэширования данных (Redis) из контейнера Docker. [Подробнее >>>](#)
- Установите Docker. [Подробнее >>>](#)
- Создайте Dockerfile. [Подробнее >>>](#)
- Создайте и запустите образ Docker. [Подробнее >>>](#)

**Важно.** Развертывание приложения в Docker рекомендуется для сред разработки и тестирования. Развертывание промышленной среды в Docker станет доступно в будущих релизах, после реализации возможности обновления приложений в Docker.

## Настроить сервер кэширования данных Creatio (Redis)

1. Откройте **redis.conf** в текстовом редакторе от имени пользователя root. Например, для этого можно использовать текстовый редактор Nano:

```
sudo nano /etc/redis/redis.conf
```

2. Найдите запись "**bind 127.0.0.1 ::1**". Замените ее на запись "**bind 0.0.0.0**", чтобы использовать все доступные интерфейсы IPV4.
3. Сохраните изменения и закройте текстовый редактор.
4. Перезагрузите сервер Redis:

```
sudo systemctl restart redis-server
```

## Установить Docker

Чтобы установить Docker, выполните команду:

```
sudo apt-get install docker
```

## Создать Dockerfile

**/path/to/application/directory/** — путь к папке, в которую распакован архив с установочными файлами Creatio.

1. Перейдите в папку приложения:

```
cd /path/to/application/directory/
```

2. Создайте Dockerfile в текстовом редакторе. Например, для этого можно использовать текстовый редактор Nano:

```
nano Dockerfile
```

### 3. Добавьте следующий код:

```
FROM mcr.microsoft.com/dotnet/core/sdk:3.1 AS base
EXPOSE 5000 5002
RUN apt-get update && \
apt-get -y --no-install-recommends install \
libgdipplus \
libc6-dev && \
apt-get clean all && \
rm -rf /var/lib/apt/lists/* /var/cache/apt/*
WORKDIR /app
COPY . ./
FROM base AS final
WORKDIR /app
ENV ASPNETCORE_ENVIRONMENT Development
ENV TZ US/Eastern
ENV COMPlus_ThreadPool_ForceMinWorkerThreads 100
ENTRYPOINT ["dotnet", "Terrasoft.WebHost.dll"]
```

### 4. Нажмите Ctrl+O, чтобы сохранить изменения.

### 5. Закройте редактор нажатием Ctrl+X.

## Создать и запустить образ Docker

**На заметку..** Если вы разворачиваете среду разработки .Net Core с доступом по протоколу http, то перед запуском образа Docker необходимо внести изменения в конфигурационный файл Terrasoft.WebHost.dll.config, который находится в корневой папке приложения. Для параметра "add key" укажите значение

```
<add key="CookiesSameSiteMode" value="Lax" />
```

Это обеспечит корректную работу как по протоколу https, так и по http. В тоже время при такой настройке не будет работать мобильное приложение.

### Создайте образ Docker:

```
docker build -f Dockerfile -t creatioimg .
```

Запустите созданный образ:

```
docker run -p http_port_number:5000 -p https_port_number:5002 -d --dns=DNS_server_ip --dns-sear
```

**http\_port\_number** — номер порта. Docker будет предоставлять версию HTTP через этот порт.

**https\_port\_number** — номер порта. Docker будет предоставлять версию HTTPS через этот порт.

**DNS\_server\_ip** — IP-адрес сервера DNS, который позволяет серверу определять адреса интернет-доменов. Вы можете использовать несколько значений **--dns** для разных DNS-серверов.

**DNS\_address\_suffix** — поисковый домен DNS, который позволяет контейнеру искать не полностью определенные имена хостов. Вы можете использовать несколько значений **--dns-search** для разных поисковых доменов DNS.

**На заметку.** Добавьте **--restart**, чтобы по команде всегда использовался определенный контейнер.

Версия Creatio для HTTP будет доступна через порт **http\_port\_number**.

Версия Creatio для HTTPS будет доступна через порт **https\_port\_number**.

**На заметку.** Первый вход в систему после настройки приложения осуществляется под учетной записью Supervisor (логин — Supervisor; пароль — Supervisor).

## Настроить работу Creatio .NetCore на https

Прежде чем приступить к работе в приложении по протоколу HTTPS, вам необходимо:

1. Получить цифровой сертификат от центра сертификации в формате \*.pfx.

**На заметку.** При использовании самоподписанного сертификата подключение мобильного приложения к сайту Creatio будет недоступно из-за ограничений политики безопасности мобильных приложений.

2. Открыть файл appsettings.json, который находится в корневом каталоге приложения.
3. В блоке "Https" указать адрес вашего сайта, путь к полученному сертификату и пароль.

### Пример блока "Https" файла appsettings.json

```
"Https": {
  "Url": "https://::5002",
  "Certificate": {
    "Path": "C:\\Projects\\site\\20210215_103239\\localhost.pfx",
    "Password": "Password"
  }
}
```

**На заметку.** Вы можете указать как относительный путь к сертификату, так и абсолютный. Обратите внимание, что абсолютный путь должен быть указан в json-совместимом формате.