

**Программный комплекс универсальная интеграционная платформа для построения систем информационного мониторинга и анализа больших объёмов данных (ПК УИП «ГЛОРИЯ»).**

**Инструкция по установке и запуску**

**2022**

# СОДЕРЖАНИЕ

1 Состав.....	3
2 Условия выполнения программы.....	3
3 Состав ПО.....	4
4 Установка ПО .....	4
4.1 Подготовка для Windows .....	4
4.2 Подготовка для AstraLinux (Orel).....	4
4.3 Подготовка СУБД PostgreSQL.....	6
4.4 Распаковка и конфигурирование ПО .....	9
5 Запуск ПО .....	9
5.1 Запуск ПО в Windows .....	9
5.2 Запуск ПО в AstraLinux.....	10

## 1 Состав

Программный комплекс универсальная интеграционная платформа для построения систем информационного мониторинга и анализа больших объёмов данных (ПК УИП «ГЛОРИЯ») (далее ПО) состоит из следующих основных модулей:

- сервер бизнес-логики (далее Сервер);
- сервер публикации Web-интерфейса (далее Web-сервер);

## 2 Условия выполнения программы

На рабочем месте должен быть установлен персональный компьютер (далее ПК), со следующими минимальными характеристиками:

- частота процессора 3,0 ГГц;
- объем оперативной памяти 2 ГБ;
- размер жесткого диска 250 ГБ;
- сетевая карта Ethernet 10/100 Мбит/с;
- ОС Windows 7SP1 и выше либо AstraLinux 2.12.40 и выше.

Компьютер должен быть оснащен монитором с диагональю не менее 21" и разрешением экрана не ниже 1920x1080.

### 3 Состав ПО

- UIP.Server.zip – архив с Сервером
- UIP.Web.zip – архив с Web-сервером
- uip.backup – резервная копия базы данных для PostgreSQL

### 4 Установка ПО

Поскольку ПО представляет собой кроссплатформенный комплекс, ниже будет изложен процесс установки для Windows и для защищённой системы AstraLinux. ПО реализована на платформе .Net Core 6.0 и использует СУБД PostgreSQL для хранения данных.

#### 4.1 Подготовка для Windows

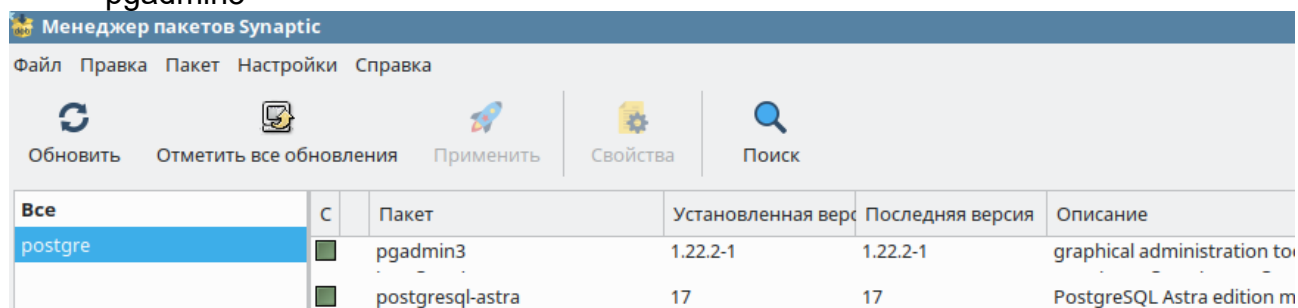
Windows не имеет централизованного менеджера пакетов, поэтому установка необходимых для ПО компонент осуществляется непосредственно скачиванием нужных инсталляторов:

- .Net 6.0 можно скачать на странице <https://dotnet.microsoft.com/en-us/download/dotnet/6.0> по ссылке для соответствующей платформы, установка в соответствии с инструкциями инсталлятора
- PostgreSQL можно скачать на странице <https://www.enterprisedb.com/downloads/postgres-postgresql-downloads> для соответствующей платформы, установка в соответствии с инструкциями инсталлятора (примечание: PostgreSQL может быть установлен на другом компьютере, если у вас есть к нему доступ то пропустите установку и перейдите к восстановлению БД на том сервере)

#### 4.2 Подготовка для AstraLinux (Orel)

Если при установке операционной системы не был установлен компонент СУБД, то его необходимо установить с помощью встроенного менеджера пакетов (Synaptic), где можно установить два мета-пакета, включающих в себя всё нужное для работы СУБД PostgreSQL:

- postgresql-astra
- pgadmin3



В зависимости от установки системы в менеджере пакетов может присутствовать возможность работы с репозиториями без публичного ключа или нет. Если поддержка есть, и то следует проверить что подключены (либо подключить) репозитории:

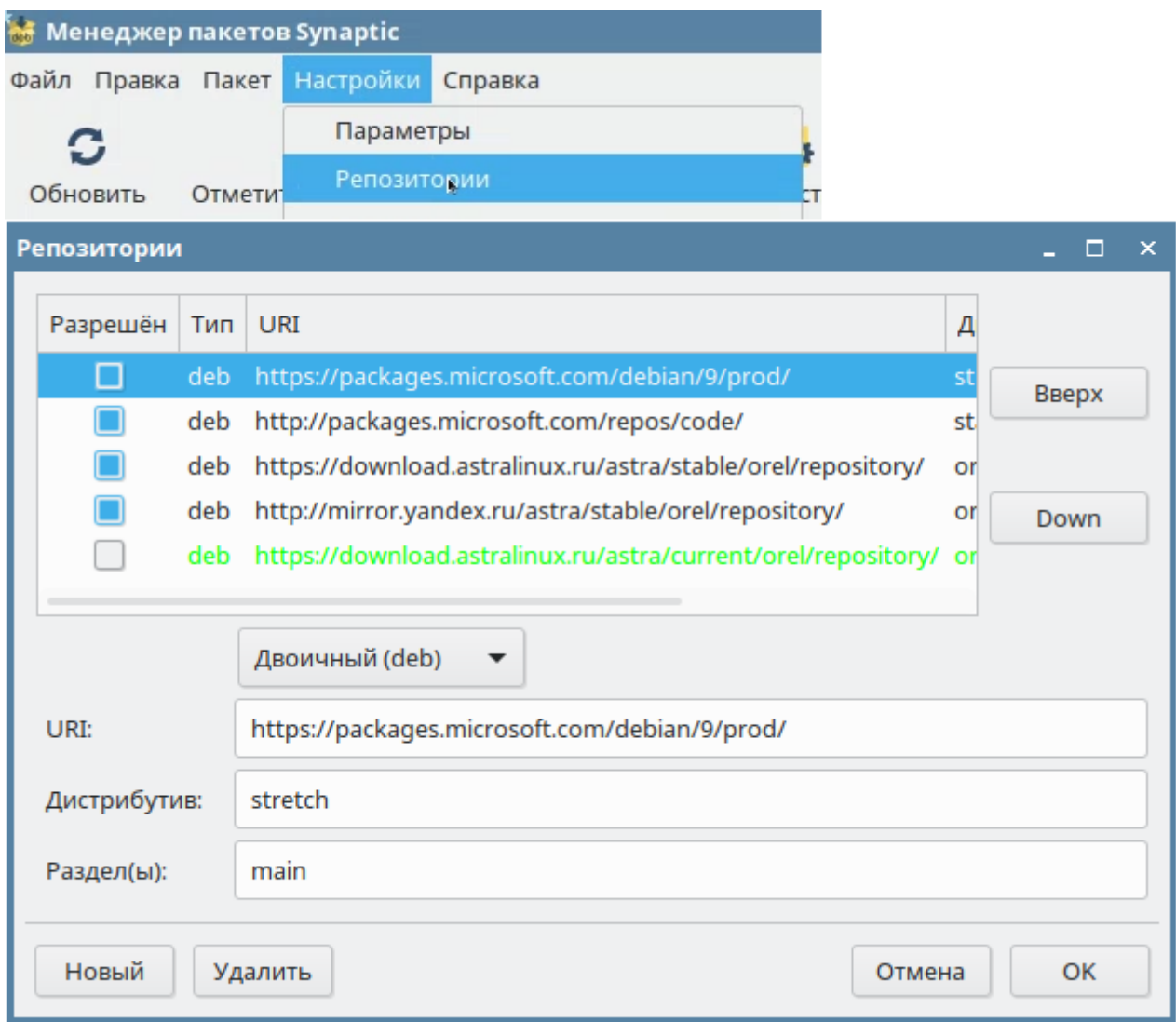
URI

Дистрибутив      Раздел(ы)

<http://packages.microsoft.com/repos/code/>  
<https://packages.microsoft.com/debian/9/prod/>

stable  
stretch

main  
main



После чего установить мета-пакет `aspnetcore-runtime-6.0` с которым установятся и остальные компоненты для работы среды .Net Core 6.0.

Если менеджер пакетов не позволяет обращаться к этим репозиториям, то последнюю версию .Net 6.0 можно скачать для соответствующей платформы по ссылке <https://dotnet.microsoft.com/en-us/download/dotnet/6.0> после чего в папке где скачан соответственный архив выполнить команды:

```
mkdir -p $HOME/dotnet && tar xzf dotnet-sdk-6.0.102-linux-x64.tar.gz -C $HOME/dotnet
```

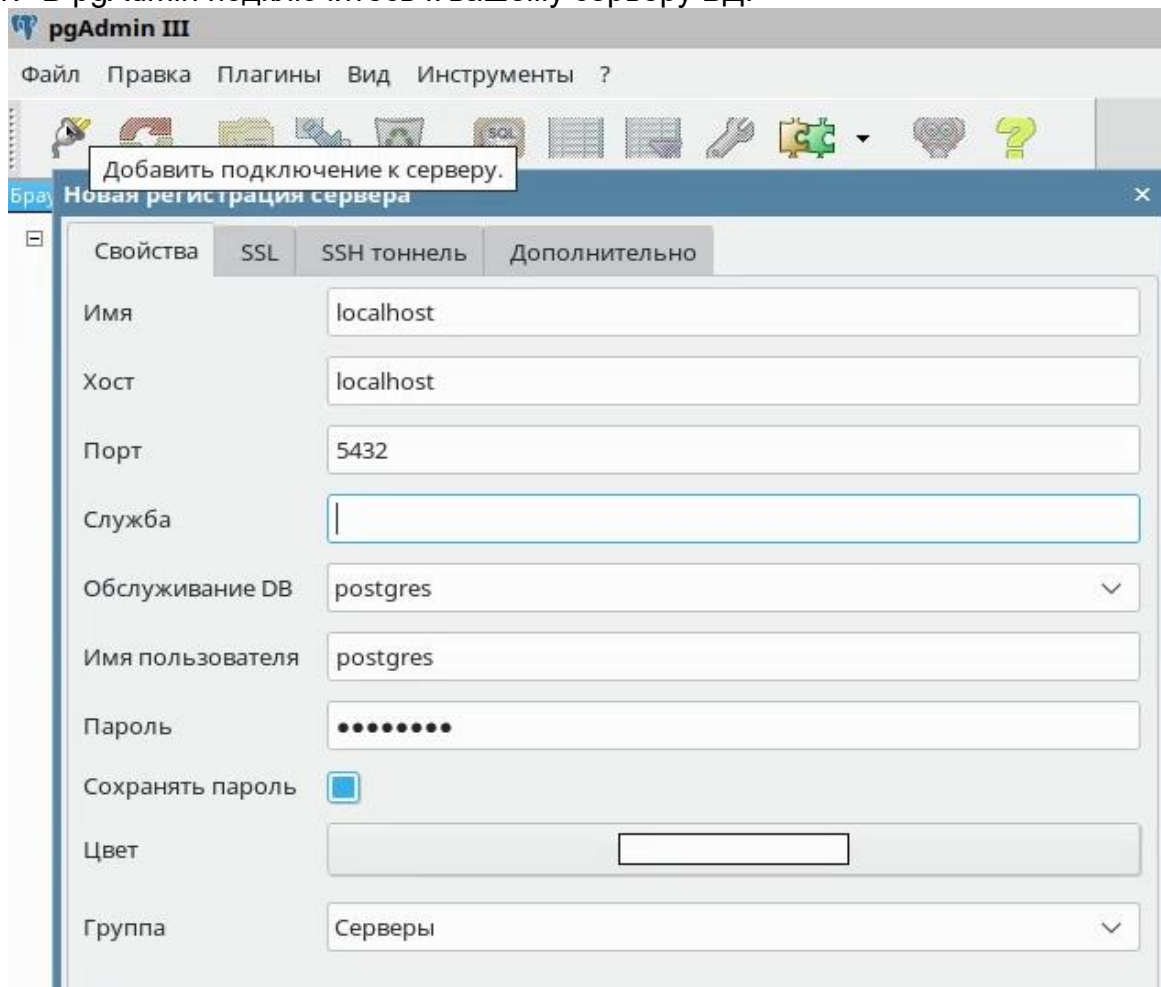
```
export DOTNET_ROOT=$HOME/dotnet  
export PATH=$PATH:$HOME/dotnet
```

где `dotnet-sdk-6.0.102-linux-x64.tar.gz` – скачанный архив, а вместо `$HOME` может быть путь где вам удобнее установить .Net 6.0;

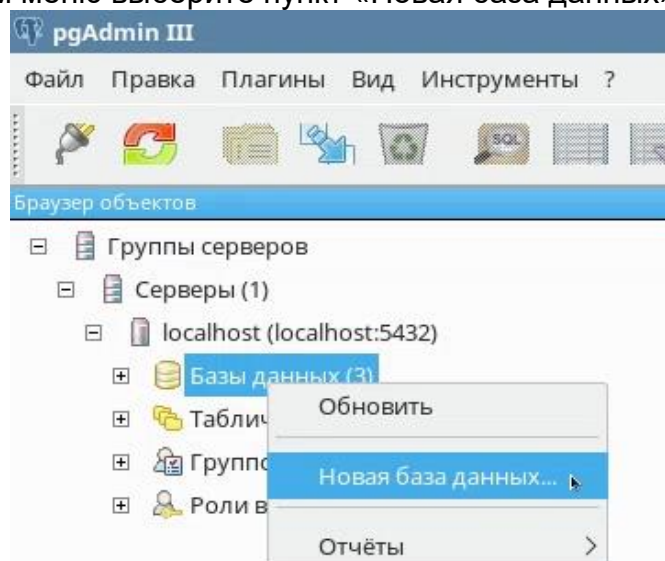
### 4.3 Подготовка СУБД PostgreSQL

После установки компонент, нужно создать и восстановить из резервной копии базу данных. Установщик PostgreSQL вместе с самой СУБД предоставляет инструмент pgAdmin, при помощи которой удобнее восстановить базу данных:

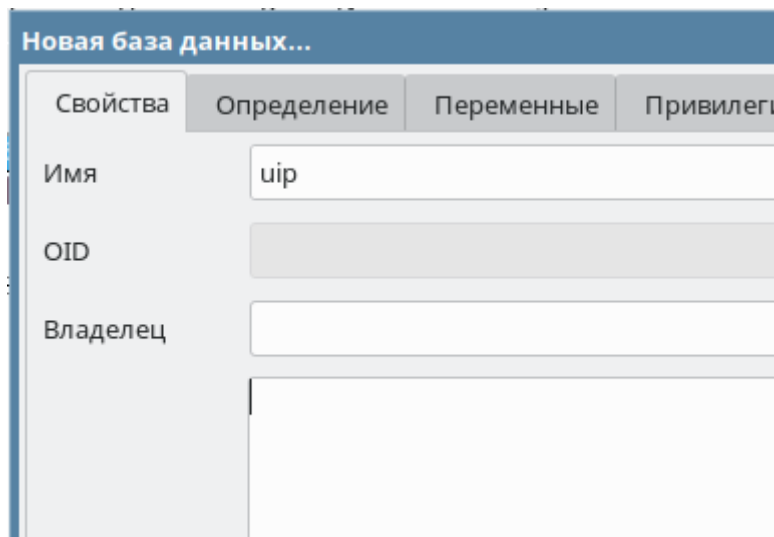
1. В pgAdmin подключитесь к вашему серверу БД.



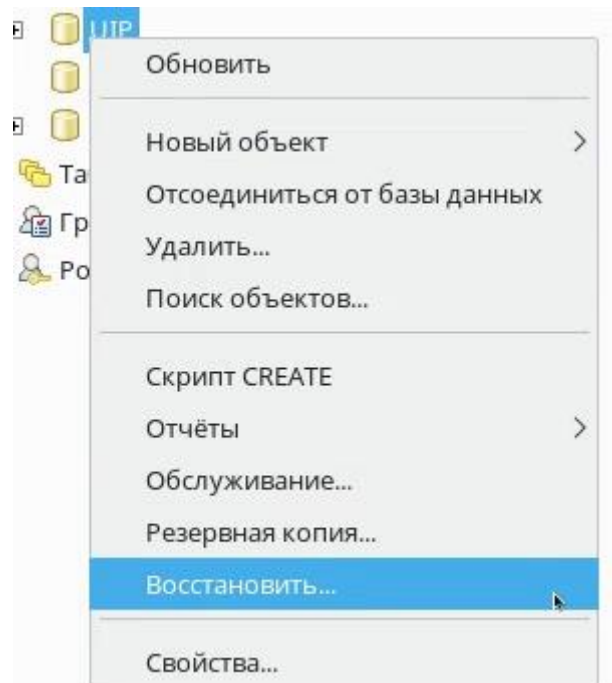
2. В контекстном меню выберите пункт «Новая база данных»



3. В появившемся окне введите название БД (напр. uip) и подтвердите создание



4. В списке баз данных, выберите созданную и в контекстном меню выберите пункт «Восстановить»



5. В появившемся окне укажите путь к файлу uip.backup и подтвердите восстановление.

Примечание:

По умолчанию, после установки PostgreSQL создаёт пользователя «postgres», либо с пустым паролем, который система предложит сменить при первом подключении, либо с паролем «postgres». Однако, установщик для AstraLinux может задать другой пароль, тогда его можно сменить выполнив следующие команды в терминале:

```
su – postgres  
psql  
\password
```

```
root@astra:/home# su - postgres
postgres@astra:~$ psql
psql (9.6.20)
Введите "help", чтобы получить справку.

postgres=# \password
Введите новый пароль:
Повторите его:
postgres=# █
```

Обратите внимание, при вводе пароля в терминале Linux не отображаются никакие символы, вводите пароль внимательно.



## 4.4 Распаковка и конфигурирование ПО

Архивы UIP.Server.zip и UIP.Web.zip следует распаковать в отдельные папки. В папке с файлами из архива UIP.Server.zip найдите файл appsettings.json и откройте его в любом текстовом редакторе. В файле содержится единственный параметр — строка подключения к БД.

```
{
  "ConnectionStrings": {
    "Default": "host=localhost;port=5432;database=ui;username=postgres;password=postgres;"
  }
}
```

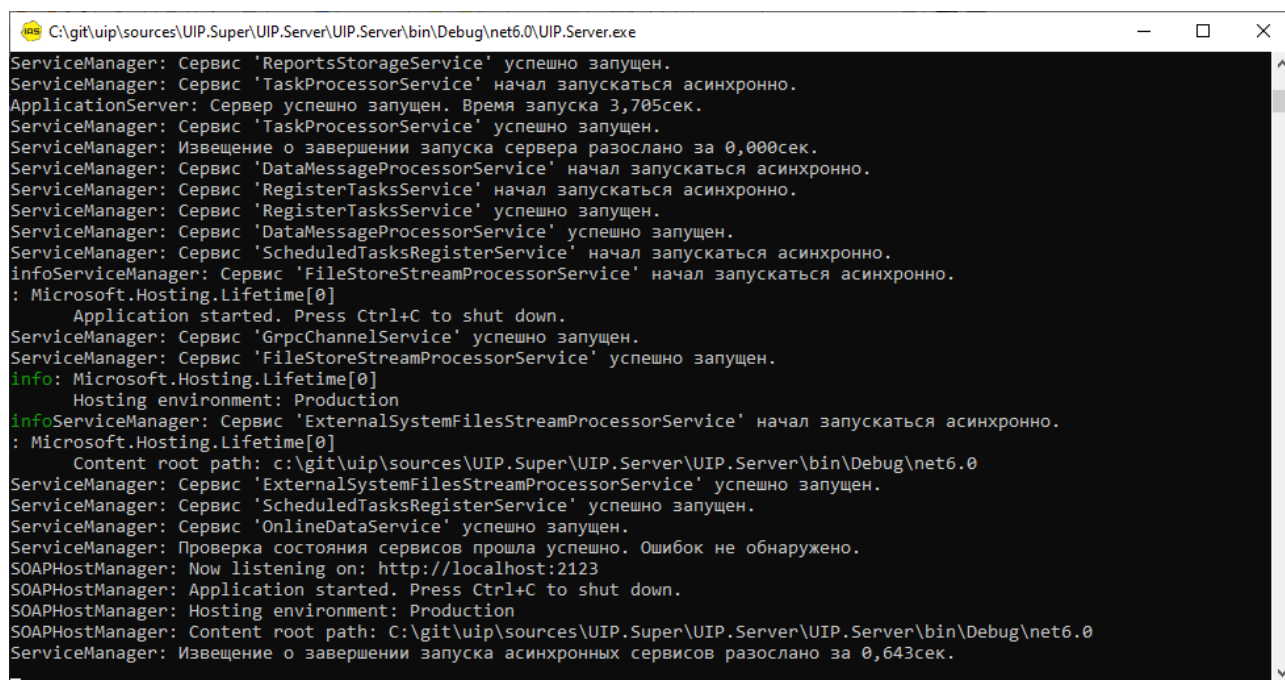
Если вы устанавливали PostgreSQL на этом же компьютере тогда в строке подключения нужно заменить параметры database={название созданной БД} и пароль пользователя password={заданный при установке пароль}.

Если вы решили использовать другой сервер PostgreSQL то укажите его параметры.

## 5 Запуск ПО

### 5.1 Запуск ПО в Windows

Для запуска Сервера в соответствующей папке нужно запустить файл UIP.Server.exe, откроется консольное окно, отображающее журнал запуска.



```
C:\git\ui\sources\UIP.Super\UIP.Server\UIP.Server\bin\Debug\net6.0\UIP.Server.exe
ServiceManager: Сервис 'ReportsStorageService' успешно запущен.
ServiceManager: Сервис 'TaskProcessorService' начал запускаться асинхронно.
ApplicationServer: Сервер успешно запущен. Время запуска 3,705сек.
ServiceManager: Сервис 'TaskProcessorService' успешно запущен.
ServiceManager: Извещение о завершении запуска сервера разослано за 0,000сек.
ServiceManager: Сервис 'DataMessageProcessorService' начал запускаться асинхронно.
ServiceManager: Сервис 'RegisterTasksService' начал запускаться асинхронно.
ServiceManager: Сервис 'RegisterTasksService' успешно запущен.
ServiceManager: Сервис 'DataMessageProcessorService' успешно запущен.
ServiceManager: Сервис 'ScheduledTasksRegisterService' начал запускаться асинхронно.
InfoServiceManager: Сервис 'FileStoreStreamProcessorService' начал запускаться асинхронно.
: Microsoft.Hosting.Lifetime[0]
  Application started. Press Ctrl+C to shut down.
ServiceManager: Сервис 'GrpcChannelService' успешно запущен.
ServiceManager: Сервис 'FileStoreStreamProcessorService' успешно запущен.
Info: Microsoft.Hosting.Lifetime[0]
  Hosting environment: Production
InfoServiceManager: Сервис 'ExternalSystemFilesStreamProcessorService' начал запускаться асинхронно.
: Microsoft.Hosting.Lifetime[0]
  Content root path: c:\git\ui\sources\UIP.Super\UIP.Server\UIP.Server\bin\Debug\net6.0
ServiceManager: Сервис 'ExternalSystemFilesStreamProcessorService' успешно запущен.
ServiceManager: Сервис 'ScheduledTasksRegisterService' успешно запущен.
ServiceManager: Сервис 'OnlineDataService' успешно запущен.
ServiceManager: Проверка состояния сервисов прошла успешно. Ошибок не обнаружено.
SOAPHostManager: Now listening on: http://localhost:2123
SOAPHostManager: Application started. Press Ctrl+C to shut down.
SOAPHostManager: Hosting environment: Production
SOAPHostManager: Content root path: C:\git\ui\sources\UIP.Super\UIP.Server\UIP.Server\bin\Debug\net6.0
ServiceManager: Извещение о завершении запуска асинхронных сервисов разослано за 0,643сек.
```

Если всё установлено и сконфигурировано корректно, то в последних строках журнала можно увидеть строку «ServiceManager: Проверка состояния сервисов прошла успешно. Ошибок не обнаружено.»

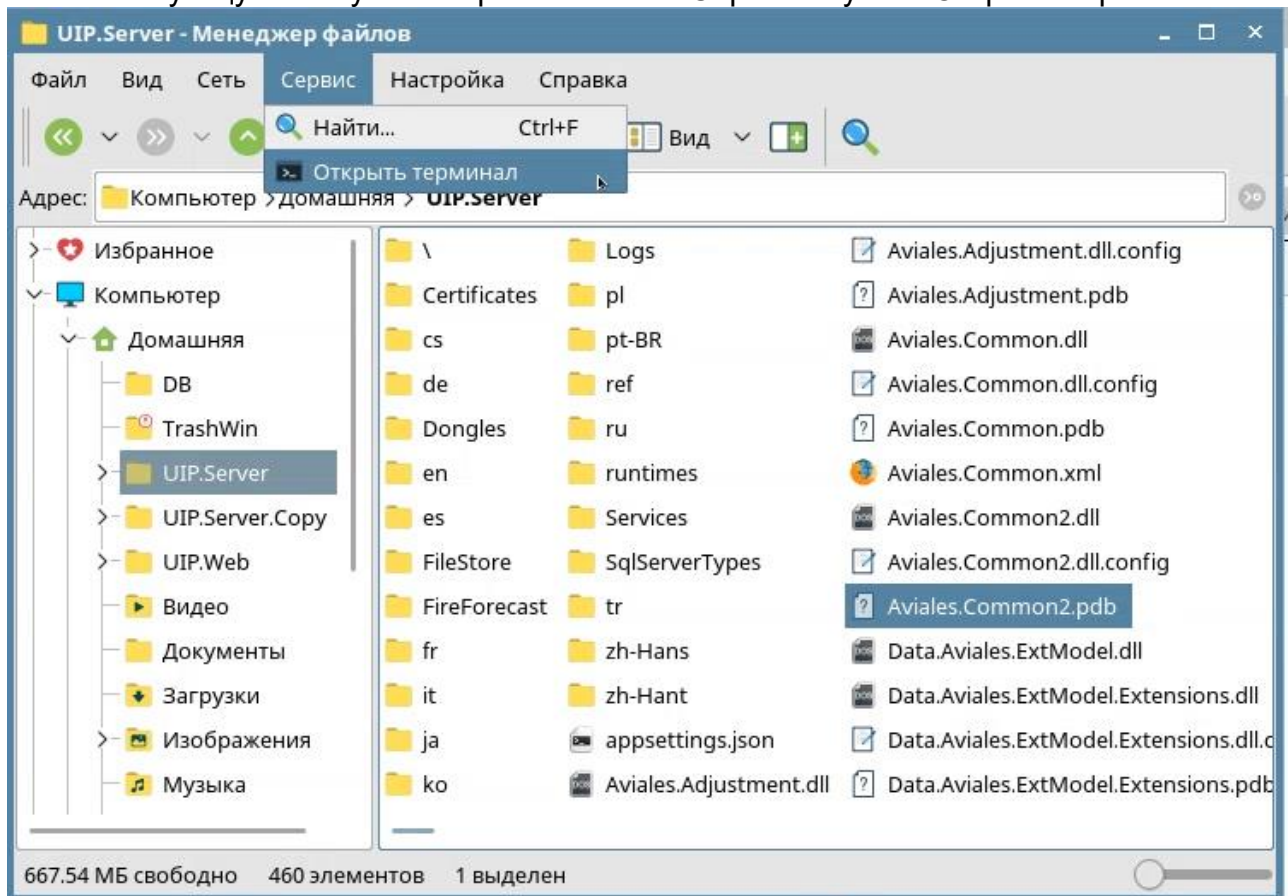
Для запуска Web-сервера в соответствующей папке нужно запустить файл UIP.Web.exe, откроется консольное окно отображающее журнал запуска.

```
D:\Work\UIP.Web.publish\UIP.Web.exe
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:5001
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: D:\Work\UIP.Web.publish
```

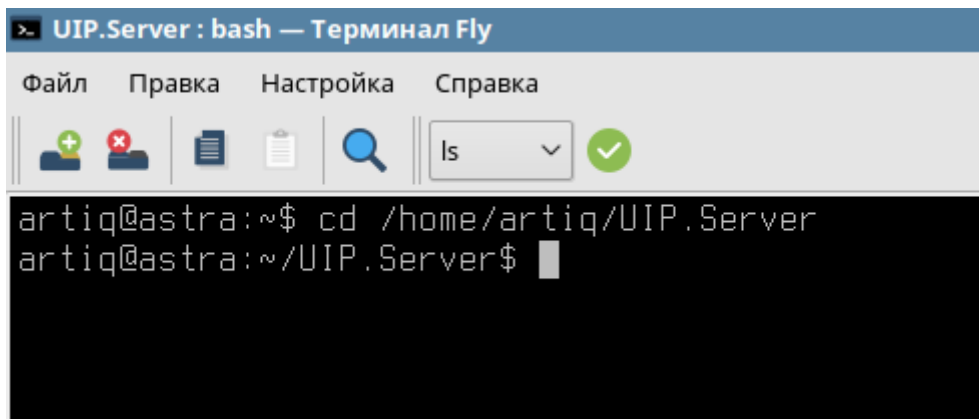
Если всё установлено и сконфигурировано корректно, то в открытом консольном будет присутствовать строка «Application started. Press Ctrl+C to shut down.»

## 5.2 Запуск ПО в AstraLinux

Для запуска ПО под AstraLinux нужно открыть по окну терминала для каждой папки куда разархивировано ПО. Для этого в менеджере файлов перейдите в соответствующую папку и выберите в меню «Сервис» пункт «Открыть терминал».

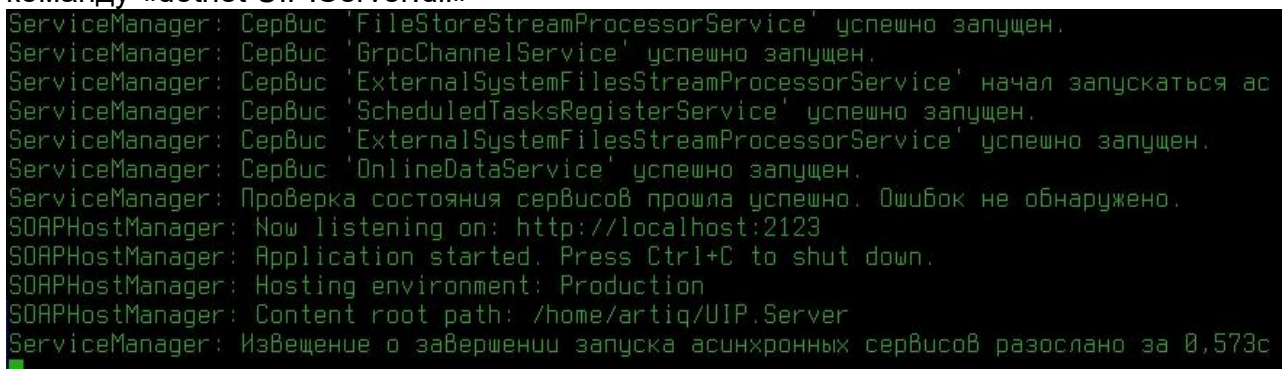


Также можно открыть терминал из меню приложений системы и перейти в нужную папку с помощью команды «cd [путь к папке ПО]»



```
UIP.Server : bash — Терминал Fly
Файл  Правка  Настройка  Справка
[Icons: +, -, document, search, ls, checkmark]
artiq@astra:~$ cd /home/artiq/UIP.Server
artiq@astra:~/UIP.Server$
```

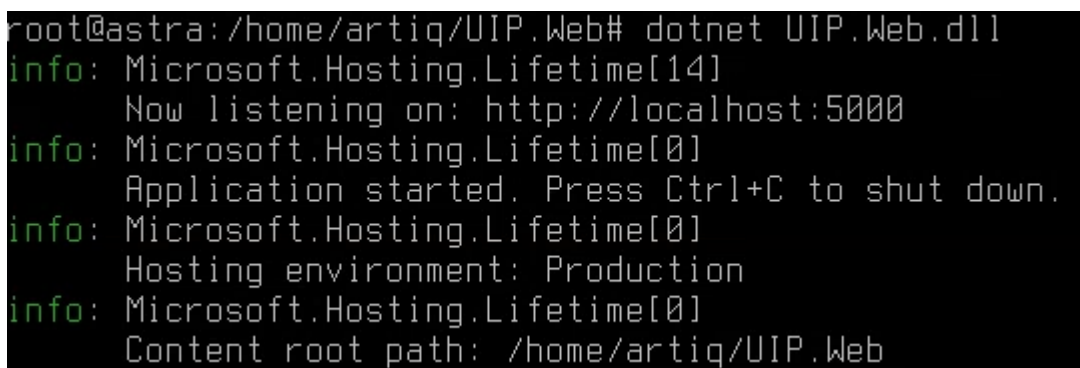
Для запуска Сервера в соответствующем терминале нужно выполнить команду «dotnet UIP.Server.dll»



```
ServiceManager: Сервис 'FileStoreStreamProcessorService' успешно запущен.
ServiceManager: Сервис 'GrpcChannelService' успешно запущен.
ServiceManager: Сервис 'ExternalSystemFilesStreamProcessorService' начал запускаться ас
ServiceManager: Сервис 'ScheduledTasksRegisterService' успешно запущен.
ServiceManager: Сервис 'ExternalSystemFilesStreamProcessorService' успешно запущен.
ServiceManager: Сервис 'OnlineDataService' успешно запущен.
ServiceManager: Проверка состояния сервисов прошла успешно. Ошибок не обнаружено.
SOAHostManager: Now listening on: http://localhost:2123
SOAHostManager: Application started. Press Ctrl+C to shut down.
SOAHostManager: Hosting environment: Production
SOAHostManager: Content root path: /home/artiq/UIP.Server
ServiceManager: Извещение о завершении запуска асинхронных сервисов разослано за 0,573с
```

Если всё установлено и сконфигурировано корректно, то в последних строках журнала можно увидеть строку «ServiceManager: Проверка состояния сервисов прошла успешно. Ошибок не обнаружено.»

Для запуска Сервера в соответствующем терминале нужно выполнить команду «dotnet UIP.Web.dll»



```
root@astra:/home/artiq/UIP.Web# dotnet UIP.Web.dll
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/artiq/UIP.Web
```

Если всё установлено и сконфигурировано корректно, то в открытом консольном будет присутствовать строка «Application started. Press Ctrl+C to shut down.»

После запуска приложений в браузере можно будет открыть сайт по адресу <http://localhost:5000> и увидеть ПО в работе.