

«ПРАЙМ ГИС: АНАЛИТИКА»

ИНСТРУКЦИЯ ПО УСТАНОВКЕ И НАСТРОЙКЕ

Москва, 2019

Установка с использованием Docker

Установка Docker

Инструкции по установке доступны на сайте Docker по ссылке <https://docs.docker.com/install/>.

Требуется также установка модуля docker-compose: <https://docs.docker.com/compose/install/>.

Для debian, ubuntu:

Установка docker

```
sudo apt-get update

sudo apt-get install \
  apt-transport-https \
  ca-certificates \
  curl \
  gnupg2 \
  software-properties-common

curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add
-

sudo add-apt-repository \
  "deb [arch=amd64] https://download.docker.com/linux/debian \
  $(lsb_release -cs) \
  stable"
```

После установки

Создание группы «docker»:

```
sudo groupadd docker
```

Добавление текущего пользователя в группу:

```
sudo usermod -aG docker $USER
```

Выйдите и войдите в систему, чтобы информация о членстве в группе обновилась.

Установка docker-compose

```
sudo curl -L
  "https://github.com/docker/compose/releases/download/1.24.0/docker-
  compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose
```

Структура папок приложения

Config — Настройки

Docker — Файлы приложения для использования с Docker

Log — Журналы приложения

postgres-data — Файлы БД (папка будет создана при первом запуске)

Published — Данные опубликованных отчётов

PublishedExt – Данные опубликованных отчётов для внешнего доступа
Statdata – Данные для автоматически обновляемых источников данных

Управление приложением

Все указанные в этом разделе команды должны выполняться из папки Docker внутри папки приложения.

```
cd Docker
```

Запуск приложения

```
docker-compose up -d
```

При первом запуске будут загружены все необходимые для работы приложения компоненты и созданы образы Docker.

Для проверки работоспособности приложения выполните команду:

```
curl http://localhost:5000/Manager/ver.json
```

Результат:

```
"0.5.3.0"
```

После первого запуска, можно также использовать команду:

```
docker-compose start
```

Остановка приложения

```
docker-compose stop
```

Удаление приложения

```
docker-compose down
```

Настройки приложения

Ссылка на приложение

Ссылка на приложение: `http://<сервер>:<порт>/Manager/`

<сервер> – IP или имя сервера, где было установлено приложение.

<порт> – по умолчанию 5000

Основные настройки

Файл `Config/appsettings.json`

Для того чтобы настройки вступили в силу, перезапустите приложение командой:

```
docker-compose restart
```

Настройки:

```
{
  "appSettings": {
    "_": "Корневая папка для публикации в файловой системе. Если не указана - используется FTP.",
    "ExtDataFolder": "../PublishedExt",
    "_": "Настройки подключения к FTP-серверу опубликованных отчётов",
```

```

"ExtFtpServerUri": "#ftp-uri#",
"ExtFtpUsername": "#ftp-user#",
"ExtFtpPassword": "#ftp-password#",
"ExtFtpDataFolder": "#ftp-folder#",

"_": "Время, через которое полномочия пользователя перепроверяются (чч:мм:сс)",
"SessionValidityTimeout": "00:01:00",
"_": "Время неактивности пользователя, после которого сессия завершается (чч:мм:сс)",
"SessionTimeout": "01:00:00"
},

"clientSettings": {
  "rootExtUrl": "#root-ext-url#"
},

"auth": {
  "Mode": "Demo",
  "DemoUsers": [
    {
      "Username": "statgis-user",
      "Password": "test123",
      "Groups": "statgis-users"
    },
    {
      "Username": "statgis-admin",
      "Password": "test123",
      "Groups": "statgis-admins"
    }
  ],
  "LdapUri": "#ldap-uri#",
  "LdapUsername": "#ldap-username#",
  "LdapPassword": "#ldap-password#"
}
}

```

Номер порта

По-умолчанию приложение настроено на использование порта 5000.

Для того чтобы изменить порт, укажите его в файле Docker/docker-compose.yml. Например, для порта 80:

```

...
statgis-app:
  build: ./app
  image: statgis-app
  ports:
    - "80:5000"
...

```

Для того чтобы настройки вступили в силу, используйте команду:

```
docker-compose up -d
```

Настройки журнала

Файл Config/nlog.config

```

<?xml version="1.0" encoding="utf-8" ?>
<!--
  Настройки файлов журналов.
  Подробно эти настройки описаны в документе
https://github.com/nlog/nlog/wiki/Configuration-file
-->

```

```

<nlog xmlns="http://www.nlog-project.org/schemas/NLog.xsd"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      autoReload="true">

  <extensions>
    <add assembly="NLog.Web.AspNetCore"/>
  </extensions>

  <targets>
    <!--
      Атрибут encoding - кодировка файла журнала.
      Атрибут fileName - формат имени файла журнала.
      Атрибут layout - формат строки файла журнала.
      Атрибут archiveAboveSize - архивировать файл журнала при превышении данного размера.
      Атрибут archiveFileName - формат имени архивных файлов.
      Атрибут maxArchiveFiles - максимальное количество архивных файлов.
    -->
    <target name="file" xsi:type="File" keepFileOpen="false"
            encoding="utf-8"
            fileName="${basedir}/../Log/${logger}.log"
            layout="${longdate} ${event-properties:item=EventId_Id} ${uppercase:${level}}
${message}${onexception: ${exception}}${when:when=length('${aspnet-request-url}') > 0:inner=
(${aspnet-request-url})"
            archiveAboveSize="524288"
            archiveFileName="${basedir}/../Log/${logger}-{#####}.log.zip"
            archiveNumbering="Sequence"
            maxArchiveFiles="1024"
            enableArchiveFileCompression="true" />
    <target name="error-file" xsi:type="File" keepFileOpen="false"
            encoding="utf-8"
            fileName="${basedir}/../Log/${logger}-error.log"
            layout="${longdate} ${event-properties:item=EventId_Id} ${uppercase:${level}}
${message}${when:when=length('${aspnet-request-url}') > 0:inner= (${aspnet-request-
url})}${newline}${exception:format=ToString}"
            archiveAboveSize="524288"
            archiveFileName="${basedir}/../Log/${logger}-error-{#####}.log.zip"
            archiveNumbering="Sequence"
            maxArchiveFiles="1024"
            enableArchiveFileCompression="true" />
    <target name="ms-file" xsi:type="File" keepFileOpen="false"
            encoding="utf-8"
            fileName="${basedir}/../Log/Framework.log"
            layout="${longdate} ${logger} ${event-properties:item=EventId_Id}
${uppercase:${level}} ${message}${onexception: ${exception}}${when:when=length('${aspnet-
request-url}') > 0:inner= (${aspnet-request-url})"
            archiveAboveSize="524288"
            archiveFileName="${basedir}/../Log/Framework-{#####}.log.zip"
            archiveNumbering="Sequence"
            maxArchiveFiles="1024"
            enableArchiveFileCompression="true" />
    <target name="ms-error-file" xsi:type="File" keepFileOpen="false"
            encoding="utf-8"
            fileName="${basedir}/../Log/Framework-error.log"
            layout="${longdate} ${logger} ${event-properties:item=EventId_Id}
${uppercase:${level}} ${message}${when:when=length('${aspnet-request-url}') > 0:inner=
(${aspnet-request-url})}${newline}${exception:format=ToString}"
            archiveAboveSize="524288"
            archiveFileName="${basedir}/../Log/Framework-error-{#####}.log.zip"
            archiveNumbering="Sequence"
            maxArchiveFiles="1024"
            enableArchiveFileCompression="true" />
    <target name="console" xsi:type="ColoredConsole"
            layout="${longdate} ${logger} ${event-properties:item=EventId_Id}
${uppercase:${level}} ${message}${when:when=length('${aspnet-request-url}') > 0:inner=
(${aspnet-request-url})${onexception:${newline}${exception:format=ToString}}"

```

```

    />
</targets>

<rules>
  <!--
    Атрибут minlevel - минимальный уровень сообщений, сохраняемых в журнал.

    Уровни сообщений:
      Trace - подробная информация
      Debug - отладочная информация
      Info - информационные сообщения
      Warn - предупреждения
      Error - ошибки
      Fatal - фатальные ошибки
      Off - отключить
  -->
  <logger name="Microsoft.*" minlevel="Warn" writeTo="ms-error-file,console"/>
  <logger name="Microsoft.*" minlevel="Info" writeTo="ms-file" final="true"/>
  <logger name="*" minlevel="Trace" writeTo="file"/>
  <logger name="*" minlevel="Info" writeTo="console"/>
  <logger name="*" minlevel="Warn" writeTo="error-file"/>
</rules>
</nlog>

```

Настройка публикации отчётов для внешнего доступа на веб-сервере

По-умолчанию, внешние отчёты доступны по ссылкам вида:

```
http://<host>:5000/PublishedExt/<path>
```

Для того чтобы сделать отчёты доступными извне вы можете настроить свой реверсный прокси для обслуживания этих ссылок.

Альтернативно, приложение может размещать данные опубликованных для внешнего доступа отчётов на вашем веб-сервере. Для этого:

1. Скопируйте папку PublishedExt на веб-сервер, указав удобное для вас название (например StatGisExt);
2. Скопируйте папку Docker/app/Viewer на веб-сервер внутрь папки StatGisExt;
3. Организуйте доступ по ftp к папке StatGisExt с сервера приложения;
4. Укажите данные для доступа к папке по ftp в конфигурационном файле приложения:


```

"ExtFtpServerUri": "#ftp-uri#",
"ExtFtpUsername": "#ftp-user#",
"ExtFtpPassword": "#ftp-password#",
"ExtFtpDataFolder": "#ftp-folder#",

```
5. Укажите ссылку, по которой папка StatGisExt на веб-сервера доступна извне в конфигурационном файле приложения (например <https://my-web-server.com/StatGisExt>):

```

"clientSettings": {
  "rootExtUrl": "https://my-web-server.com/StatGisExt"
},

```

6. Перезапустите приложение:

```
cd Docker  
docker-compose restart
```

Обновление приложения

Для обновления предыдущих версий приложения до данной версии:

1. Замените папку `Docker/app` соответствующей папкой из комплекта поставки;
2. В папке `Docker` внутри папки приложения выполните команду:
`docker-compose up -d --build`

Принудительное обновление компонентов, входящих в состав «ПраймГИС: Аналитика», невозможно без участия владельцев Системы.