

**Платформа быстрой разработки аналитических
приложений
Insight™**

INSIGHT

**Подготовка сервера и установка приложений
на сервере**

ОГЛАВЛЕНИЕ

1. ОБЩАЯ ИНФОРМАЦИЯ	3
2. ПОДГОТОВКА СТРУКТУРЫ ПАПОК.....	4
3. УСТАНОВКА И ЗАПУСК DOCKER	4
4. УСТАНОВКА DREMIO	5
5. УСТАНОВКА ПЛАГИНА GOODT-DREMIO-PLUGIN	6
6. УСТАНОВКА SDK RTL-DREMIO-CONNECTOR	6
7. УСТАНОВКА RTL-GOODTEDITOR-PLAYER	7
8. ЗАПУСК ПРИЛОЖЕНИЙ.....	7
9. АЛЬТЕРНАТИВНАЯ УСТАНОВКА БЕЗ ИСПОЛЬЗОВАНИЯ DOCKER	8
9.1. ПОДГОТОВКА СТРУКТУРЫ ПАПОК.....	8
9.2. УСТАНОВКА ВНЕШНИХ ЗАВИСИМОСТЕЙ	8
9.3. УСТАНОВКА DREMIO	9
9.4. УСТАНОВКА ПРИЛОЖЕНИЙ	10
10. ПРОВЕРКА РАБОТОСПОСОБНОСТИ УСТАНОВЛЕННОГО ПРИЛОЖЕНИЯ	12

1. ОБЩАЯ ИНФОРМАЦИЯ

Программный комплекс Insight является сложно организованной системой, для нормального функционирования которой требуется стопроцентное удовлетворение внешних зависимостей, настроек операционной системы и корректное указание значений переменных в конфигурационных файлах.

Ввиду специфики экосистемы Linux и разнообразия сборок даже одноименных дистрибутивов, возможны нюансы развертывания, которые не удалось осветить в данной инструкции.

В случае возникновения сложностей при развертывании, ошибок при запуске и прочих проблемах, необходимо обратиться по адресу электронной почты: devops@goodt.me

Insight это продукт, объединяющий в себе **методологию и комплекс программных продуктов** как собственной разработки, так и Opensource решения.

Установка сводится к следующим шагам:

1. Подготовка структуры папок;
2. Установка и запуск Docker;
3. Установка Dremio;
4. Установка плагина goodt-dremio-plugin;
5. Установка SDK rtl-dremio-connector;
6. Установка rtl-goodteditor-player;
7. Запуск приложений;
8. Альтернативная установка без использования Docker:
 1. Подготовка структуры папок;
 2. Установка внешних зависимостей;
 3. Установка Dremio;
 4. Установка приложений.
9. Проверка работоспособности установленных приложений.

2. ПОДГОТОВКА СТРУКТУРЫ ПАПОК

Приложение должно быть размещено в строгой структуре папок. Следование этому требованию обеспечит совместимость со скриптами, предназначенными для дальнейшего обновления приложения.

- abc
 - distr
 - scripts
 - data
 - dremio
 - app
 - goodt
 - goodt-dremio-plugin

Для создания структуры выполните команду

```
sudo mkdir -p /abc/distr/scripts /abc/data/dremio /abc/app/goodt/goodt-dremio-plugin
```

3. УСТАНОВКА И ЗАПУСК DOCKER

Добавьте репозиторий:

```
sudo yum install -y yum-utils
sudo yum-config-manager --add-repo
https://download.docker.com/linux/centos/docker-ce.repo
```

Установите Docker:

```
sudo yum install docker-ce docker-ce-cli containerd.io
```

Включите автозапуск службы Docker:

```
sudo systemctl enable docker
```

Запустите службу Docker:

```
sudo systemctl start docker
```

4. УСТАНОВКА DREMIO

На момент написания данного руководства в полном объеме поддерживается версия Dremio 4.7.2. Выполните загрузку docker-образа данной версии в локальный репозиторий:

```
sudo docker pull dremio/dremio-oss:4.7.2
```

Выберите IP-адрес интерфейса, к которому будет привязан контейнер с Dremio. Для получения списка адресов выполните:

```
hostname -I
```

Вывод будет примерно такой:

```
172.20.55.241 172.17.0.1
```

Далее в инструкции будет использоваться адрес **172.20.55.241**.

Создайте файл **/abc/distr/scripts/dremio.sh**:

```
sudo touch /abc/distr/scripts/dremio.sh
```

Вставьте в него следующее содержимое:

```
docker run -d --name dremio \  
    --mount type=bind,source=/abc/data/dremio,target=/opt/dremio/data \  
    -v /abc/app/goodt/goodt-dremio-plugin/goodt-dremio- \  
plugins.jar:/opt/dremio/jars/goodt-dremio-plugins.jar \  
    -p 172.20.55.241:9047:9047 \  
    -p 172.20.55.241:31010:31010 \  
    -p 172.20.55.241:45678:45678 \  
    -p 172.20.55.241:2181:2181 \  
    dremio/dremio-oss:4.7.2 \  
    --bind=0.0.0.0/0
```

Назначьте файлу атрибут "исполняемый":

```
sudo chmod +x /abc/distr/scripts/dremio.sh
```

Придумайте имя пользователя и пароль, который будет использоваться для доступа в Dremio. Далее в инструкции будет использоваться следующая учетная запись:

Пользователь: dremio_admin

Пароль: \$7roNgP@s\$w0r9

5. УСТАНОВКА ПЛАГИНА GOODT-DREMIO-PLUGIN

Загрузите файл

https://storage.googleapis.com/artifacts.good-big-data.appspot.com/apps/goodt-dremio-plugins-4_master.jar в папку ***/abc/app/goodt/goodt-dremio-plugin***:

```
cd /abc/app/goodt/goodt-dremio-plugin
wget https://storage.googleapis.com/artifacts.good-big-data.appspot.com/apps/goodt-dremio-plugins-4_master.jar
```

Создайте символическую ссылку ***/abc/app/goodt/goodt-dremio-plugin/goodt-dremio-plugins.jar*** на только что загруженный файл:

```
ln -s /abc/app/goodt/goodt-dremio-plugin/goodt-dremio-plugins-4_master.jar
/abc/app/goodt/goodt-dremio-plugin/goodt-dremio-plugins.jar
```

6. УСТАНОВКА SDK RTL-DREMIO-CONNECTOR

Загрузите файл образа https://storage.googleapis.com/artifacts.good-big-data.appspot.com/apps/rtl-dremio-connector-38_master.tar в папку ***/abc/distr***:

```
cd /abc/distr
wget https://storage.googleapis.com/artifacts.good-big-data.appspot.com/apps/rtl-dremio-connector-38_master.tar
```

Выполните загрузку образа в локальный репозиторий:

```
sudo docker load -i rtl-dremio-connector-38_master.tar
```

Создайте файл ***/abc/distr/scripts/dremio-connector.sh***:

```
touch /abc/distr/scripts/dremio-connector.sh
```

Вставьте в него следующее содержимое:

```
docker run -d --name=dremio-connector -p 4400:4400 \
  -e "RTL_DREMIO_USER=dremio_admin" \
  -e "RTL_DREMIO_PASS=$7roNgP@s$w0r9" \
  -e "RTL_DREMIO_HOST=172.20.55.241" \
  -e "RTL_DREMIO_PORT=31010" \
  -e "RTL_APP_PORT=4400" \
  containers.goodt.me/rtl-dremio-connector/rtl-dremio-connector-
38_master:latest \
  --bind=0.0.0.0/0
```

Назначьте файлу атрибут "исполняемый":

```
sudo chmod +x /abc/distr/scripts/dremio-connector.sh
```

7. УСТАНОВКА RTL-GOODTEDITOR-PLAYER

Загрузите файл образа https://storage.googleapis.com/artifacts.good-big-data.appspot.com/apps/rtl-goodteditor-player-406_master.tar в папку `/abc/distr`:

```
cd /abc/distr
wget https://storage.googleapis.com/artifacts.good-big-data.appspot.com/apps/rtl-goodteditor-player-406_master.tar
```

Выполните загрузку образа в локальный репозиторий:

```
sudo docker load -i rtl-goodteditor-player-406_master.tar
```

Создайте файл `/abc/distr/scripts/player.sh`:

```
touch /abc/distr/scripts/player.sh
```

Вставьте в него следующее содержимое:

```
docker run -d --name player \
  -p 172.20.55.241:80:80 \
  -e "UEP_WFM_URL=http://localhost" \
  -e "UEP_DREMIO_URL=http://localhost:4000" \
  -e "UEP_APP_JSON=app_test_cnp" \
  containers.goodt.me/rtl-goodteditor-player/rtl-goodteditor-player-
380_master \
  --bind=0.0.0.0/0
```

Назначьте файлу атрибут "исполняемый":

```
sudo chmod +x /abc/distr/scripts/player.sh
```

8. ЗАПУСК ПРИЛОЖЕНИЙ

Запустите приложения в следующем порядке:

1. Dremio
2. SDK rtl-dremio-connector
3. rtl-goodteditor-player

Для это выполните команды:

```
cd /abc/distr/scripts
./dremio.sh
./dremio-connector.sh
./player.sh
```

9. АЛЬТЕРНАТИВНАЯ УСТАНОВКА БЕЗ ИСПОЛЬЗОВАНИЯ DOCKER

Внимание! Данный способ приводится для справки и не рекомендован к использованию!

9.1. ПОДГОТОВКА СТРУКТУРЫ ПАПЕК

Приложение должно быть размещено в строгой структуре папок. Следование этому требованию обеспечит совместимость со скриптами, предназначенными для дальнейшего обновления приложения.

- abc
 - bin
 - dremio
 - conf
 - player
 - distr
 - scripts
 - data
 - dremio
 - app
 - goodt
 - dremio-connector
 - goodt-dremio-plugin

Для создания структуры выполните команду

```
sudo mkdir -p /abc/distr/scripts /abc/data/dremio /abc/app/goodt/goodt-dremio-plugin /abc/bin/dremio /abc/app/goodt/dremio-connector /abc/conf/player
```

9.2. УСТАНОВКА ВНЕШНИХ ЗАВИСИМОСТЕЙ

Скачайте архив с Java 8 и распакуйте его в папке **/abc/bin**:

```
cd /abc/bin
wget https://storage.googleapis.com/wfm-dep/linserver/jre-8u131-linux-x64.tar.gz
tar -xf /abc/distr/jre-8u131-linux-x64.tar.gz
mv jre1.8.0_131 java
```

Пропишите путь в переменную **JAVA_HOME** в файле **~/.bash_profile**:

```
Для добавления можно использовать скрипт
echo 'JAVA_HOME=/abc/bin/java ' >> ~/.bash_profile \
&& echo 'export JAVA_HOME ' >> ~/.bash_profile \
&& echo 'PATH=$JAVA_HOME/bin:$PATH ' >> ~/.bash_profile \
```



```
&& echo 'export PATH' >> ~/.bash_profile
```

Примените изменения:

```
. ~/.bash_profile
```

Установите NVM и PM2:

```
curl --silent --location https://rpm.nodesource.com/setup_12.x | sudo bash  
-  
sudo yum -y install nodejs  
sudo npm install -g pm2 node-gyp-install gulp jspm
```

Убедитесь, что pm2 успешно установился:

```
pm2 -version
```

Вывод должен быть примерно такой:

```
[PM2] Spawning PM2 daemon with pm2_home=/root/.pm2  
[PM2] PM2 Successfully daemonized  
4.4.0
```

Установите Nginx:

```
sudo yum install -y epel-release  
sudo yum install -y nginx
```

Отредактируйте файл конфигурации Nginx, выполнив команду:

```
sudo sed -i 's,/usr/share/nginx/html,/abc/app/goodt/player, '  
/etc/nginx/conf.d/default.conf
```

Включите автозапуск службы Nginx и запустите службу:

```
sudo systemctl enable nginx  
sudo systemctl start nginx
```

9.3. УСТАНОВКА DREMIO

Установите Dremio в папку **/abc/bin/dremio** в соответствии с официальной документацией: <https://docs.dremio.com/quickstart/standalone-quickstart.html>

Придумайте имя пользователя и пароль, который будет использоваться для доступа в Dremio. Далее в инструкции будет использоваться следующая учетная запись:

Пользователь: dremio_admin
Пароль: \$7roNgP@s\$w0r9

Если вы используете брандмауер, то откройте в нем порты, необходимые для работы Dremio:

- 9047
- 31010
- 45678
- 2181

9.4. УСТАНОВКА ПРИЛОЖЕНИЙ

SDK rti-dremio-connector

Загрузите архив https://storage.googleapis.com/artifacts.good-big-data.appspot.com/apps/goodt-dremio-connector-38_master.zip в папку */abc/distr*:

```
cd /abc/distr
wget https://storage.googleapis.com/artifacts.good-big-data.appspot.com/apps/goodt-dremio-connector-38_master.zip
```

Распакуйте содержимое архива в папку */abc/app/goodt/dremio-connector*:

```
unzip goodt-dremio-connector-38_master.zip -d /abc/app/goodt/dremio-connector
```

Создайте файл */abc/app/goodt/dremio-connector/dremio-connector.json*:

```
/abc/app/goodt/dremio-connector/dremio-connector.json
```

Добавьте в него следующее содержимое:

```
{
  "apps": [
    {
      "name": "goodt-dremio-connector-38_master",
      "cwd": ".",
      "script": "/abc/bin/java/bin/java",
      "args": ["-jar", "/abc/app/goodt/dremio-connector/goodt-dremio-connector-38_master.jar", "--goodt.dremio.host=localhost", "--goodt.dremio.port=31010", "--goodt.dremio.user=dremio_admin", "--goodt.dremio.pass=$7roNgP@s$w0r9", "--server.port=4000"],
      "watch": ["/abc/app/goodt/dremio-connector/goodt-dremio-connector-38_master.jar"],
      "node_args": [],
      "log_date_format": "YYYY-MM-DD HH:mm Z",
      "exec_interpreter": "",
      "exec_mode": "fork"
    }
  ]
}
```

Установка плагина goodt-dremio-plugin

Загрузите файл https://storage.googleapis.com/artifacts.good-big-data.appspot.com/apps/goodt-dremio-plugins-4_master.jar в папку `/abc/app/goodt/goodt-dremio-plugin`:

```
cd /abc/app/goodt/goodt-dremio-plugin
wget https://storage.googleapis.com/artifacts.good-big-data.appspot.com/apps/goodt-dremio-plugins-4_master.jar
```

Создайте символическую ссылку `/abc/bin/dremio/jars/goodt-dremio-plugins.jar` на только что загруженный файл:

```
ln -s /abc/app/goodt/goodt-dremio-plugin/goodt-dremio-plugins-4_master.jar
/abc/bin/dremio/jars/goodt-dremio-plugins.jar
```

Установка rtl-goodteditor-player

Загрузите архив https://storage.googleapis.com/artifacts.good-big-data.appspot.com/apps/goodtplayer-406_master.zip в папку `/abc/distr`:

```
cd /abc/distr
wget https://storage.googleapis.com/artifacts.good-big-data.appspot.com/apps/goodtplayer-406_master.zip
```

Распакуйте скачанный файл и поместите его содержимое в папку приложения:

```
unzip goodtplayer-406_master.zip
mv ./dist /abc/app/goodt/player
```

Создайте файл `/abc/conf/player/production.json`:

```
touch /abc/conf/player/production.json
```

Добавьте в него следующее содержимое:

```
{
  "api": {
    "wfm": {
      "baseUrl": "http://localhost"
    },
    "dremio": {
      "baseUrl": "http://localhost:4000"
    }
  }
}
```

Создайте символическую ссылку на файл конфигурации в папке приложения:

```
ln -s /abc/conf/player/production.json /abc/app/goodt/player/config
```

Переместите папку с файлами **app.json** в папку **/abc/conf/player**:

```
mv /abc/app/goodt/player/appjson/* /abc/conf/player
```

Создайте символическую ссылку для **app.json**, содержащего данные для функционального отображения и для файла **style.css**:

```
ln -s /abc/conf/player/app_test_cnp/app.json /abc/app/goodt/player
ln -s /abc/conf/player/app_test_cnp/style.css /abc/app/goodt/player/css
```

Запуск приложений

1. Запустите Dremio в соответствии с официальной документацией
2. Запустите приложение SDK rti-dremio-connector :

```
pm2 start /abc/app/goodt/dremio-connector/dremio-connector.json
```

10. ПРОВЕРКА РАБОТОСПОСОБНОСТИ УСТАНОВЛЕННОГО ПРИЛОЖЕНИЯ

Наиболее простым способом проверки работоспособности является отправка **HTTP-GET** запроса командой **curl**.

Проверку следует проводить с того узла, на котором установлено приложение. Такая проверка исключит влияние настроек брандмауэра узла, а так же настроек промежуточного сетевого оборудования. Все запросы будут отправляться на **localhost**.

rtl-goodteditor-player

```
curl http://localhost/
```

Ответ работоспособного приложения (приведен в сокращенной форме, только начало и конец тела ответа, значения HTML-тегов могут отличаться в каждом конкретном случае):

```
<!DOCTYPE html><html lang="en"><head><meta charset="utf-8"><meta http-
equiv="X-UA-Compatible" content="IE=edge"><meta name="viewport"
content="width=device-width,initial-scale=1"><link rel="icon"
href="favicon.ico"><title>goodteditor-player</title><link rel="stylesheet"
href="css/style.css"><link href="css/chunk-00bb3896.faec886d.css"
rel="prefetch">
.
.
</script></body></html>
```

goodt-dremio-plugin

```
curl http://localhost:4000/api
```

Ответ работоспособного приложения (значение поля ***timestamp*** будет отличаться от приведенного):

```
{"timestamp":"2020-10-23T06:48:54.710+0000","status":404,"error":"Not Found","message":"No message available","path":"/"}
```