

**Платформа разработки аналитических
приложений
Insight**

INSIGHT

Подготовка сервера и установка приложений
на сервере

ОГЛАВЛЕНИЕ

1.	ОБЩАЯ ИНФОРМАЦИЯ.....	3
2.	ПОДГОТОВКА.....	4
3.	УСТАНОВКА И ЗАПУСК ANSIBLE.....	4
3.1.	НАСТРОЙКА ХОСТОВ	5
4.	ОНЛАЙН УСТАНОВКА КОМПОНЕНТОВ INSIGHT.....	7
4.1.	УСТАНОВКА АДМИН-АККАУНТА ПРИ ПЕРВОМ ЗАПУСКЕ.....	7
4.2.	УСТАНОВКА NGINX.....	7
4.3.	УСТАНОВКА MYSQL.....	7
4.4.	УСТАНОВКА POSTGRESQL	7
4.5.	POSTGRESQL - ДОБАВЛЕНИЕ ПОЛЬЗОВАТЕЛЕЙ И БД.....	7
4.6.	УСТАНОВКА JENKINS.....	7
4.7.	УСТАНОВКА KEYCLOAK	7
4.8.	УСТАНОВКА APACHE NIFI.....	7
4.9.	УСТАНОВКА DOCKER(DOCKER-COMPOSE)	7
5.	ОФФЛАЙН УСТАНОВКА КОМПОНЕНТОВ INSIGHT	8
5.1.	УСТАНОВКА NGINX.....	8
5.2.	УСТАНОВКА MYSQL.....	8
5.3.	УСТАНОВКА POSTGRESQL	8
5.4.	УСТАНОВКА JENKINS.....	8
5.5.	УСТАНОВКА KEYCLOAK	8
5.6.	УСТАНОВКА APACHE NIFI.....	8
5.7.	УСТАНОВКА DOCKER(DOCKER-COMPOSE)	8
6.	УСТАНОВКА NGINX	8
7.	УСТАНОВКА MYSQL.....	9
8.	УСТАНОВКА POSTGRESQL.....	10
9.	УСТАНОВКА JENKINS	10
10.	УСТАНОВКА KEYCLOAK	11
11.	УСТАНОВКА APACHE NIFI.....	12
12.	УСТАНОВКА DOCKER(DOCKER-COMPOSE)	13
13.	УСТАНОВКА INSIGHT	13

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.

В программный стек необходимый для работы, входит:

- NGINX;
- MYSQL;
- PostgreSQL;
- Jenkins;
- Keycloak;
- Apache NIFI;
- Docker.

Установка сводится к следующим шагам:

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

2. ПОДГОТОВКА

Плейбуки используемые для развертки хранятся в *git.goodt.me*

Файлы необходимые для установки компонентов хранятся в *art.goodt.me*

Важно!

При размещении плейбука в каталоге отличном от */ansible/playbooks/* внутри команды необходимо прописывать адрес директории где размещены плейбука

Пример команды с расположением плейбука */ansible/playbooks/*

```
ansible-playbook dremio_standalone.yml --extra-vars "mysecret=***** , ram=16"
```

Пример команды с расположением файла в */abc/playbooks/*

```
ansible-playbook -i /abc/playbooks/dremio_standalone.yml --extra-vars "mysecret=***** , ram=16"
```

Перед разверткой представленные плейбуки необходимо отредактировать, вписав в поле *hosts* хост или группу хостов, где будут произведены действия из плейбука
Значения доступных хостов для ansible, указываются в файле */etc/ansible/hosts*

При размещении файла *hosts* в каталоге отличном от */etc/ansible/hosts* внутри команд необходимо использовать опцию *-i hosts* где *hosts* адрес каталога в котором размещается файл

Пример команды с расположением файла в */etc/ansible/hosts*:

```
ansible-playbook dremio_standalone.yml --extra-vars "mysecret=***** , ram=16"
```

Пример команды с расположением файла в */abc/ansible/hosts*:

```
ansible-playbook -i /abc/ansible/hosts dremio_standalone.yml --extra-vars "mysecret=***** , ram=16"
```

3. УСТАНОВКА И ЗАПУСК ANSIBLE

Первым шагом необходимо подключить репозиторий

```
yum install epel-release
```

Установить ansible

yum install ansible

После базовой установки необходимо выполнить установку дополнительных компонентов и импорт файла *hosts* с помощью плейбука *ansible.yml*

Примеры типовых форматирований и именовании:

- *mysecret* - пароль от хранилища артефактов *art.goodt.me*

Пример команды для онлайн установки:

```
ansible-playbook ansible.yml --extra-vars "mysecret=*****"
```

Пример команды для оффлайн установки:

```
ansible-playbook ansible.yml --extra-vars "offline=true"
```

Для запуска оффлайн установки, необходимо чтобы в папке */abc/distr* находились следующие файлы:

- *python-crypto-2.6.1-1.el7.rf.x86_64.rpm*
- *python-httplib2-0.9.2-1.el7.noarch.rpm*
- *python-jinja2-2.7.2-4.el7.noarch.rpm*
- *python-keyczar-0.71c-2.el7.noarch.rpm*
- *sshpas-1.05-1.el7.rf.x86_64.rpm*
- *ansible-2.9.27-1.el7.noarch.rpm*
- *ansible-playbook.tar*

3.1. НАСТРОЙКА ХОСТОВ

Перед началом работы необходимо указать хосты в файле *hosts*

Для того чтобы просмотреть/отредактировать содержание файла */etc/ansible/hosts* используйте команду

```
sudo nano /etc/ansible/hosts
```

Пример содержания импортируемого файла *hosts*, где

[*group-name*] - это наименование группы хоста

ip-адрес - это внешний или внутренний *ip* адрес хоста на которой установлен или будет установлен тот или иной компонент системы (группа может содержать 1 и более *ip* адресов).

hosts

[*dremio-master*]

ip-адрес

[dremio-executor]
ip-адрес VM

[dremio-fs]
ip-адрес

[pg]
ip-адрес

[keycloak]
ip-адрес

[zabbix]
ip-адрес

[nifi]
ip-адрес

[airflow]
ip-адрес

[editor]
ip-адрес

[jenkins]
ip-адрес

[dremio-connector]
ip-адрес

[player]
ip-адрес

[mysql]
ip-адрес

4. ОНЛАЙН УСТАНОВКА КОМПОНЕНТОВ INSIGHT

4.1. УСТАНОВКА АДМИН-АККАУНТА ПРИ ПЕРВОМ ЗАПУСКЕ

```
ansible-playbook dremio_admin.yml --extra-vars "user=admin , password=*****"
```

4.2. УСТАНОВКА NGINX

```
ansible-playbook balanser_nginx.yml
```

4.3. УСТАНОВКА MYSQL

```
ansible-playbook mysql.yml --extra-vars "mysecret=***** , root_password=*****"
```

4.4. УСТАНОВКА POSTGRESQL

```
ansible-playbook -i hosts pgsql.yml --extra-vars "mysecret=***** , pg_pass=*****"
```

4.5. POSTGRESQL - ДОБАВЛЕНИЕ ПОЛЬЗОВАТЕЛЕЙ И БД

```
ansible-playbook pgsql_rtl_db.yml --extra-vars "pg_user_pass=*****"
```

4.6. УСТАНОВКА JENKINS

```
ansible-playbook jenkins.yml --extra-vars "mysecret=*****"
```

4.7. УСТАНОВКА KEYCLOAK

Параметры запуска плейбука находятся в `vars/main.yml`

Для ssl необходимо положить в директорию `file/` актуальный сертификат (`fullchain.crt`) в pem формате, а также приватный ключ (`private.key`) к сертификату

```
ansible-playbook keycloak.yml
```

После запуска keycloak будет доступен по следующему пути:
<https://hostname:9443/auth/>

Создание realm через импорт.

Параметры запуска плейбука находятся в `vars/main.yml`

```
ansible-playbook keycloak-add-realm.yml
```

4.8. УСТАНОВКА APACHE NIFI

```
ansible-playbook nifi.yml
```

4.9. УСТАНОВКА DOCKER(DOCKER-COMPOSE)

```
ansible-playbook docker.yml --extra-vars "mysecret=*****"
```

5. ОФФЛАЙН УСТАНОВКА КОМПОНЕНТОВ INSIGHT

5.1. УСТАНОВКА NGINX

```
ansible-playbook balancer_nginx.yml --extra-vars "offline=true"
```

5.2. УСТАНОВКА MYSQL

```
ansible-playbook mysql.yml --extra-vars " offline=true , root_password=*****"
```

5.3. УСТАНОВКА POSTGRESQL

```
ansible-playbook pgsql.yml --extra-vars "pg_pass=***** , offline=true"
```

PostgreSQL - добавление пользователей и БД

```
ansible-playbook pgsql_rtl_db.yml --extra-vars "pg_user_pass=*****"
```

5.4. УСТАНОВКА JENKINS

```
ansible-playbook jenkins.yml --extra-vars "offline=true"
```

5.5. УСТАНОВКА KEYCLOAK

Параметры запуска плейбука находятся в vars/main.yml

Для ssl необходимо положить в директорию file/ актуальный сертификат (fullchain.crt) в pem формате, а также приватный ключ (private.key) к сертификату

```
ansible-playbook keycloak.yml --extra-vars "offline=true, mysecret=*****"
```

После запуска keycloak будет доступен по следующему пути:
<https://hostname:9443/auth/>

Создание realm через импорт.

Параметры запуска плейбука находятся в vars/main.yml

```
ansible-playbook keycloak-add-realm.yml
```

5.6. УСТАНОВКА APACHE NIFI

```
ansible-playbook nifi.yml --extra-vars "offline=true"
```

5.7. УСТАНОВКА DOCKER(DOCKER-COMPOSE)

```
ansible-playbook docker.yaml --extra-vars "mysecret=*****, offline=true"
```

6. УСТАНОВКА NGINX

Примеры типовых форматирований и именований:

- `offline` - ключ указывающий на тип установки (`true` - оффлайн установка)
- `DREMIO_CONNECTOR_URL` - `proxy_pass` в конфиге `dremio_connector.vonf nginx`
- `FILEUPLOAD_URL` - `proxy_pass` в конфиге `fileupload.conf nginx`
- `KEYCLOAK_URL` - `proxy_pass` в конфиге `keycloak.conf nginx`
- `LEARNING_URL` - `proxy_pass` в конфиге `learning.conf nginx`
- `MONITOR_URL` - `proxy_pass` в конфиге `monitor.conf nginx`
- `ORGSTRUCTURE_URL` - `proxy_pass` в конфиге `orgstructure.conf nginx`
- `QUIZ_URL` - `proxy_pass` в конфиге `quiz.conf nginx`
- `ROSTALENT_URL` - `proxy_pass` в конфиге `rostalent.conf nginx`
- `TASKSETTING_URL` - `proxy_pass` в конфиге `tasksetting.conf nginx`
- `SERVER_NAME`: `server_name` в конфиге `ssl.conf nginx`

Пример команды для онлайн установки:

```
ansible-playbook balanser_nginx.yml
```

Для запуска оффлайн установки, необходимо чтобы в папке `/abc/distr` находились следующие файлы:

- `nginx-1.20.1-1.el7.ngx.x86_64.rpm`

Пример команды для оффлайн установки:

```
ansible-playbook balancer_nginx.yml --extra-vars "offline=true"
```

7. УСТАНОВКА MYSQL

Примеры типовых форматирований и именований:

- `user` - логин от хранилища артефактов `art.goodt.me`
- `mysecret` - пароль от хранилища артефактов `art.goodt.me`
- `root_password` - задать пароль `root` пользователя `Mysql` (8 символов, минимум 1 цифра, 1 заглавная буква, 1 спец.символ)

Пример команды для онлайн установки:

```
ansible-playbook mysql.yml --extra-vars " mysecret=***** , root_password=*****"
```

Для запуска оффлайн установки, необходимо чтобы в папке `/abc/distr` находились следующие файлы:

- `mysql-5.7.38-1.el7.x86_64.rpm-bundle.tar`
- `MySQL-python-1.2.5-1.el7.x86_64.rpm`

Пример команды для оффлайн установки:

```
ansible-playbook mysql.yml --extra-vars " offline=true , root_password=*****"
```

8. УСТАНОВКА POSTGRESQL

Примеры типовых форматирований и именований:

- *user* - логин от хранилища артефактов art.goodt.me
- *mysecret* - пароль от хранилища артефактов art.goodt.me
- *pg_pass* - пароль для пользователя postgres
- *offline* - ключ указывающий на тип установки (true - оффлайн установка)

Пример команды для онлайн установки:

```
ansible-playbook -i hosts pgsql.yml --extra-vars "mysecret=***** , pg_pass=*****"
```

Для запуска оффлайн установки, необходимо чтобы в папке /abc/distr находились следующие файлы:

- *postgresql11-11.15-1PGDG.rhel7.x86_64.tar.gz*

Пример команды для оффлайн установки:

```
ansible-playbook pgsql.yml --extra-vars "pg_pass=***** , offline=true"
```

PostgreSQL - добавление пользователей и БД

Примеры типовых форматирований и именований:

- *pg_user_pass* - переменная часть пароля пользователя

Пример запуска заполнения PostgreSQL:

```
ansible-playbook pgsql_rtl_db.yml --extra-vars "pg_user_pass=*****"
```

9. УСТАНОВКА JENKINS

Примеры типовых форматирований и именований:

- *user* - логин от хранилища артефактов art.goodt.me
- *mysecret* - пароль от хранилища артефактов art.goodt.me

Пример команды для онлайн установки:

```
ansible-playbook jenkins.yml --extra-vars "mysecret=*****"
```

Для запуска оффлайн установки, необходимо чтобы в папке `/abc/distr` находились следующие файлы:

- `jenkins.tar`

Пример команды для оффлайн установки:

```
ansible-playbook jenkins.yml --extra-vars "offline=true"
```

10. УСТАНОВКА KEYCLOAK

Параметры запуска плейбука находятся в `vars/main.yml`

Примеры типовых форматирований и именований:

- `offline` - ключ указывающий на тип установки (`true` - оффлайн установка)
- `artifactory` - путь для скачивания дистрибутивов
- `mysecret` - пароль от `art.goodt.me`
- `keycloak_host` - имя сервера, где будет установлен Keycloak
- `db_host` - имя сервера баз данных
- `db_name` - имя базы для Keycloak
- `db_user` - имя пользователя, под которым будет выполнено подключение к БД
- `db_pass` - пароль пользователя, под которым будет выполнено подключение к БД
- `keycloak_user` - имя пользователя, под которым будет выполнено подключение к Keycloak
- `keycloak_pass` - пароль пользователя, под которым будет выполнено подключение к Keycloak

Для `ssl` необходимо положить в директорию `file/` актуальный сертификат (`fullchain.crt`) в `pem` формате, а также приватный ключ (`private.key`) к сертификату

Пример команды для онлайн установки:

```
ansible-playbook keycloak.yml
```

Для запуска оффлайн установки, необходимо чтобы в папке `/abc/distr` находились следующие файлы:

- `keycloak-18.0.2.tar.gz`
- `jdk-11.0.9_linux-x64_bin.tar`

Пример команды для оффлайн установки:

```
ansible-playbook keycloak.yml --extra-vars "offline=true, mysecret=*****"
```

После запуска keycloak будет доступен по следующему пути:
<https://hostname:9443/auth/>

Создание realm через импорт.

Параметры запуска плейбука находятся в `vars/main.yml`

- `realm_name` - название realm

Пример команды для импорта:

```
ansible-playbook keycloak-add-realm.yml
```

11. УСТАНОВКА APACHE NIFI

Параметры запуска плейбука находятся в `vars/main.yml`:

- `offline` - ключ указывающий на тип установки (`true` - оффлайн установка)
- `artifactory` - путь для скачивания дистрибутивов
- `user` - логин от хранилища артефактов `art.goodt.me`
- `mysecret` - пароль от `art.goodt.me`
- `fqdn_nifi` - адрес или полное доменное имя куда разворачиваем Apache NIFI
- `nifi_port` - порт на котором будет разворачиваться Apache NIFI
- `JVM_memory_max` - Объем максимальной памяти выделяемый для java
- `nifi_user` - имя создаваемого пользователя в Apache NIFI
- `nifi_password` - пароль создаваемого пользователя. Должен содержать не менее 12 символов
- `keystore_password` - пароль от хранилища `keystore.jks`
- `key_passwd` - пароль сертификата в хранилище ключей. Если не установлено, будет использоваться значение `keystore_password`.
- `truststore_password` - пароль от хранилища `truststore.jks`

Пример команды для онлайн установки:

```
ansible-playbook nifi.yml
```

Для запуска оффлайн установки, необходимо чтобы в папке `/abc/distr/` находились следующие файлы:

- `jdk-8u301-linux-x64.tar.gz`
- `nifi-1.15.3-bin.tar.gz`

Пример команды для оффлайн установки:

```
ansible-playbook nifi.yml --extra-vars "offline=true"
```

Для ssl необходимо положить в директорию file/ актуальные хранилища сертификатов (keystore.jks, truststore.jks)

12. УСТАНОВКА DOCKER(DOCKER-COMPOSE)

Параметры запуска плейбука находятся в vars/main.yml:

- *offline* - ключ указывающий на тип установки (true - оффлайн установка)
- *mysecret* - ключ указывает пароль для доступа на скачивание архива с пакетами
- *DREMIO_CONNECTOR_VER* - версия дремио коннектора
- *FILEUPLOAD_VER* - версия апп fileupload
- *INS_EDITOR_VER* - версия эдитора

Пример команды для онлайн установки:

```
ansible-playbook docker.yaml --extra-vars "mysecret=*****"
```

Для запуска оффлайн установки, необходимо чтобы в папке */abc/distr/docker* находился архив с пакетами:

- *docker-pack.tar*

Пример команды для оффлайн установки:

```
ansible-playbook docker.yaml --extra-vars "mysecret=*****, offline=true"
```

При оффлайн установки приложений через Docker необходимо разместить образы для запуска приложений в каталог */abc/distr/docker/images*

13. УСТАНОВКА INSIGHT

Перед началом установки компонентов Insight просмотрите доступные версии релизов размещенных в *art.goodt.me* и убедитесь в наличии необходимой версии для установки.

Для запуска чарта необходимо выполнить bash скрипт со следующим содержанием

```
helm install insight goodt/insight \  
--namespace=test-insight \  
--set version=release_ins_1.0b2 \  
--set host.name=$app_host \  
--set imageCredentials.username= \  
--set imageCredentials.password= \  
--set mysql.enabled=false \  

```

```

--set env.RTL_DREMIO_HTTP_URL=https://$dremio_host:9047/ \
--set env.RTL_DREMIOCONNECTOR_URL=https://$app_host/rtl-api/dremio-connector \
--set dremioconnector.values.RTL_DREMIO_USER=dremio_connector \
--set dremioconnector.values.RTL_DREMIO_PASS= \
--set dremioconnector.values.RTL_DREMIO_JDBC_URL= \
--set rtlgoodteditorplayer.values.UEP_API_URL=https://$app_host/api/ \
--set rtlgoodteditorplayer.values.UEP_PLAYER_URL=https://$app_host/player/ \
--set rtlgoodteditorplayer.values.UEP_SUP_DB_HOST=$ip_mysql_db \
--set rtlgoodteditorplayer.values.UEP_SUP_DB_PASS= \
--set rtlgoodteditorplayer.values.UEP_SUP_DB_NAME=sup_editor

```

Не забывайте поставить обратный слэш на предыдущей строке

Задать версии компонентов можно с помощью дополнения команды выше: `-f components_version.yaml` Где `components_version.yaml` подготовленный файл вида:

```

dremioconnector:
  version: release_ins_1.0b4
rtlfileupload:
  version: release_ins_1.0b4
rtlgoodteditorplayer:
  version: release_ins_1.1b0

```

Так же файл можно сформировать скриптом `set_components_vers_from_art_repo_file.sh` расположенным в репозитории `ansible-scripts`

Таблица 1: Описание параметров скрипта запуска чарта Insight:

Name	Description	Value
env.RTL_DREMIOCONNECTOR_URL	URL для подключения к Dremio Connector компоненту	https://\$app_host/rtl-api/dremio-connector
dremioconnector.values.RTL_DREMIO_JDBC_URL	URL для подключения к Dremio	dremio_host:port
dremioconnector.values.RTL_DREMIO_PASS	Номер порта компонента Dremio Connector	4000
dremioconnector.values.RTL_DREMIO_JDBC_URL	RSA ключ для подключения Dremio Connector	rostalent.rsa

Name	Description	Value
dremioconnector.values.RTL_APP_PORT	Enable Disable параметр для авторизации по ключу	false
dremioconnector.values.RTL_KEY_RS	URL для подключения к WFM	http://localhost
dremioconnector.values.RTL_DREMIO_AUTH	Enable Disable параметр для авторизации по ключу	http://localhost
mysql.enabled	Включить развертывание mysql БД в виде контейнера(бета)	false
mysql.auth.password	Задать пароль для пользователя sup_editor	-
rtlgoodteditorplayer.values.UEP_SUP_DB_HOST	IP адрес для подключения к БД Editor компонента	mysql_db_host
rtlgoodteditorplayer.values.UEP_SUP_DB_NAME	Имя БД Editor компонента	sup_editor
rtlgoodteditorplayer.values.UEP_SUP_DB_PASS	Пароль пользователя sup_editor для БД Editor компонента	-

Внимание!

UEP_SUP_DB_HOST на данный момент, ввиду тех. особенностей миграции, работает только в виде IP адреса.

Для установки параметров Keycloak необходимо дополнительно указать в bash скрипт следующие строки:

```
--set env.RTL_PROFILE=prod \  
--set env.RTL_KCLK_URL= \  
--set env.RTL_KCLK_REALM= \  
--set env.RTL_KCLK_USER= \  
--set env.RTL_KCLK_PASS= \  
--set env.RTL_KCLK_CLID= \  
--set env.RTL_KCLK_CLSECRET= \
```

```
--set env.RTL_USE_MINIO=false \
--set env.RTL_USE_FILESYSTEM=true \
--set env.RTL_USE_WEBDAV=false
```

Не забывайте поставить обратный слэш на предыдущей строке

Таблица 2: Описание параметров keycloak:

Name	Description	Value
env.RTL_PROFILE	Профиль конфигурации	prod
env.RTL_KCLK_URL	URL ссылка для использования Keycloak	https://keycloak_url
env.RTL_KCLK_REALM	-	-
env.RTL_KCLK_USER	Имя пользователя, под которым будет выполнено подключение к Keycloak	-
env.RTL_KCLK_PASS	Пароль пользователя, под которым будет выполнено подключение к Keycloak	-
env.RTL_KCLK_CLID	-	-
env.RTL_KCLK_CLSECRET	-	-
env.RTL_USE_MINIO	-	-
env.RTL_USE_FILESYSTEM	-	-
env.RTL_USE_WEBDAV	-	-

Текущая версия helm chart предназначена для для использования DNS имен вида *.goodt.me и использует SSL сертификат для данных имен.
 Текущая версия helm chart по умолчанию использует SSO авторизацию через сервер <https://goodt-auth.goodt.me:9443>

По умолчанию, для Insight, включены компоненты *editor-player*, *dremio-connector*, *fileupload*. Для отключения данных компонентов можно использовать модификаторы

```
--set dremioconnector.enabled=false
--set rtlfileupload.enabled=false
```

При необходимости удаления Insight выполните:

```
helm uninstall insight --namespace=test-insight
```