|  |
| --- |
|  |
| Инструкция по установке  информационной системы  RT.Data Governance |
|  |
| Москва 2019 |

Содержание

[Термины и определения 3](#_Toc28187946)

[1 Необходимые технические ресурсы 4](#_Toc28187947)

[2 Подготовка к установке 5](#_Toc28187948)

[3 Установка Системы 6](#_Toc28187949)

Термины и определения

Термины, использующиеся в данном документе, и их определения представлены ниже (Таблица 1).

Таблица — Термины и определения

| Термин/сокращение | Определение |
| --- | --- |
| Data Governance, Система | Решение, которое осуществляет следующие функции:  ведение реестров унифицированной методологии и алгоритмов формирования расчётных показателей;  поиск терминов для повторного использования;  выстраивание потоков данных от источника данных до средств визуализации;  обеспечение формирования отчётов по реестру и глоссарию. |

Необходимые технические ресурсы

Для того чтобы установить решение Data Governance (далее – Система) необходимы следующие технические ресурсы:

1. Сервер VM в количестве 1 (одной) штуки со следующими минимальными параметрами: vCPU — 8; RAM — 16 GB; HDD — 200 GB; OS — Ubuntu

Подготовка к установке

Для подготовки к установке Системы необходимо выполнить следующие действия:

1. Установить Docker на сервер.
2. Установить Docker-compose на сервер.
3. Распаковать архив с дистрибутивом в необходимой директории.

Установка Системы

ВАЖНО: Во время настройки учитывайте все отступы и различные символы, такие как кавычки ('). Удаление отступов/символов может привести к ошибкам.

Для того чтобы установить Систему необходимо выполнить следующие действия:

1. Опционально, если необходимо использовать прокси сервер.

Если для доступа к сети Интернет необходимо использовать прокси, то необходимо в файле конфигурации демона Docker указать прокси-сервер в настройках окружения (например, для Ubuntu в файле /usr/lib/systemd/system/docker.service в разделе [Service] указать переменную Environment="http\_proxy=PROXY\_USER:PROXY\_PASS@PROXY\_SERVER:PROXY\_PORT”).

В файле datagovernance/Dockerfile ниже строки ADD ./requirements.txt . добавить строку ENV HTTP\_PROXY "http://proxy.example.com:80"

где http://proxy.example.com:80 — адрес вашего прокси сервера

Далее в файле datagovernance/Dockerfile следующие 2 команды:

RUN pip3 install -r requirements.txt

RUN pip3 install gunicorn meinheld

После команды install необходимо добавить флаг прокси со значением вашего прокси сервера, пример того, что должно получиться, ниже:

RUN pip3 install --proxy="http://proxy.example.com:80" -r requirements.txt;

RUN pip3 install --proxy="http://proxy.example.com:80" gunicorn meinheld,

где http://proxy.example.com:80 — адрес вашего прокси сервера

1. Укажите домен, на котором планируется использовать решение, в файле nginx/datagovernance в строке server\_name test.ru; (уже имеется в файле), где test.ru — ваш домен для решения.
2. Опционально, если необходимо включить ldap авторизацию.

В файле djangoproject/settings.py в списке AUTHENTICATION\_BACKENDS закомментировать (поставить символ # перед строкой) 'django.contrib.auth.backends.ModelBackend', и раскомментировать (убрать символ # в начале строки) 'core.backends.LDAPAuthBackend'.

Далее в строке AUTH\_LDAP\_SERVER\_URI = 'ldap://example.com' указать адрес ldap сервера, где ldap://example.com — адрес вашего ldap сервера.

Далее необходимо указать учётную запись для доступа решения в LDAP. Для этого в файле djangoproject/settings.py:

в строке AUTH\_LDAP\_BIND\_DN = 'exampleUser' указать имя пользователя, где exampleUser — имя пользователя LDAP;

в строке AUTH\_LDAP\_BIND\_PASSWORD = 'passwd' указать пароль пользователя LDAP, где passwd — пароль.

Далее необходимо указать Domain Component вашего ldap сервера. Для этого в файле djangoproject/settings.py:

в настройке AUTH\_LDAP\_USER\_SEARCH найдите строку 'dc=rt,dc=ru' и укажите туда свой Domain Component.

Пример для домена rt.ru: 'dc=rt,dc=ru'.

1. docker-compose up -d --build
2. docker-compose run --rm datagovernance python3 manage.py migrate
3. docker-compose run --rm datagovernance python3 manage.py loaddata glossary/fixtures/dictionary.json
4. docker-compose run --rm datagovernance python3 manage.py loaddata glossary/fixtures/dict\_values.json
5. docker-compose run --rm datagovernance python3 manage.py migrate
6. docker-compose run --rm datagovernance python3 manage.py init\_groups\_and\_permissions
7. docker-compose run --rm datagovernance python3 manage.py collectstatic (если дистрибутив спросит о замене файлов, ответить yes)
8. Для того, чтобы создать пользователя с правами администратора, используйте следующую команду:

docker-compose run --rm datagovernance python3 manage.py createsuperuser2 --username username --password password --noinput --email 'adminuser@dg.ru'

где username — имя пользователя (если была настроена ldap авторизация, необходимо указать имя пользователя ldap), password — пароль пользователя (если была настроена ldap авторизация, то он ни на что не повлияет, укажите что угодно).

1. Опционально, если не была настроена ldap авторизация.

Для того, чтобы создать пользователя в системе, используйте команду docker-compose run --rm datagovernance python3 manage.py create\_user username password

где username — имя пользователя, password — пароль пользователя.

Далее созданному пользователю необходимо указать ФИО и прочие данные с помощью панели администратора.