



Инструкция по установке и обновлению сервисов Smart Services

Содержание

Развертывание сервисов.....	3
Подготовка к развертыванию.....	3
Установка Docker Engine.....	3
Установка Docker Compose.....	6
Развертывание.....	7
Подключение дополнительных серверов.....	10
Завершение развертывания.....	11
Запуск и настройка инструмента checker.html.....	13
Установка инструментов Elasticsearch.....	14
Установка поисковой системы Elasticsearch.....	14
Установка сервиса Filebeat.....	18
Обновление сервисов.....	20
Сервисы установлены на одном сервере.....	20
Сервисы установлены на разных серверах.....	20
Удаление сервисов.....	21

Развертывание сервисов

Перед установкой ознакомьтесь с типовыми требованиями к аппаратному и программному обеспечению. Подробнее см. в документе «Smart Services. Типовые требования к программному и аппаратному обеспечению».

1. [Выполните подготовительные действия](#) для развертывания сервисов.
2. [Разверните сервисы](#) с помощью скриптов.
3. [Выполните завершающие действия](#).
4. [Проверьте состояние сервисов](#) после запуска с помощью инструмента checker.html.
5. [Установите инструменты Elasticsearch](#) для работы с лог-файлами.

Подготовка к развертыванию

1. Скачайте необходимые для работы сервисов программные компоненты, указанные в типовых требованиях. Подробнее см. в документе «Smart Services. Типовые требования к программному и аппаратному обеспечению».
2. Установите на сервер, на котором планируется развертывание сервисов Smart Services, компоненты в следующем порядке:
 - [Docker Engine](#);
 - [Docker Compose](#);
 - Python, если он не был установлен ранее.

ПРИМЕЧАНИЕ. Если сервисы будут развертываться [на разных серверах](#), то на каждом из них должны быть установлены перечисленные компоненты.

Установка Docker Engine

Порядок установки компонентов Docker Engine зависит от используемого дистрибутива Linux.

Подробнее о порядке установки Docker на Ubuntu 20.04 LTS см. в документации Docker статью [Install Docker Engine](#). Порядок установки на [Альт Сервер 9.1](#) и [Astra Linux](#) Common Edition 2.12.40 (Орел) см. в соответствующих разделах справки.

Альт Сервер 9.1

ПРИМЕЧАНИЕ. Из-за специфических настроек безопасности в системе Альт Сервер 9.1 у пользователей по умолчанию отсутствуют права на выполнение команды sudo. Чтобы получить права, выполните соответствующие [настройки](#).

1. Получите список обновлений. Для этого выполните команды:

```
sudo apt-get update
sudo apt-get install ca-certificates curl
```

2. Подключите репозиторий **Sisyphus**:

```
sudo apt-repo add sisyphus
```

- Обновите информацию из репозитория:

```
sudo apt-get update
```

- Убедитесь, что установка Docker будет производиться из репозитория **Sisyphus**, а не из репозитория по умолчанию. Для этого выполните команду:

```
sudo apt-cache policy docker-engine
```

Пример ответа:

```
установлен: (нет)
```

```
кандидат: 20.10.6-alt1:sisyphus+271055.100.1.1@1619787705
```

ПРИМЕЧАНИЕ. Если вместо ответа возвращается ошибка или версия предлагаемого к установке пакета ниже 20, обратитесь в службу поддержки.

- Установите Docker:

```
sudo apt-get install docker-engine
```

ПРИМЕЧАНИЕ. Вместе с Docker обязательно должны установиться пакеты **containerd.io** и **docker-ce-cli**.

- Проверьте установленную версию Docker. Она должна быть не ниже 20. Для этого выполните команду:

```
sudo docker -v
```

- Отключите репозиторий **Sisyphus**:

```
sudo apt-repo rm sisyphus
```

- Проверьте, что сервис запустился:

```
sudo systemctl status docker
```

Команда должна вернуть статус «active», выделенный зеленым цветом.

- Включите сервис вручную, если он не запустился автоматически. Для этого выдайте разрешение на автоматический запуск docker вместе с системой, последовательно выполнив команды:

```
sudo systemctl enable --now docker
```

```
sudo systemctl start docker
```

ПРИМЕЧАНИЕ. Рекомендуется всегда запускать сервис docker с помощью команды sudo. При необходимости можно настроить запуск сервиса docker без использования sudo. Для этого нужно добавить пользователя текущего сеанса в группу docker. При этом следует учитывать, что добавление пользователей в группу docker повышает риск несанкционированного доступа к системным файлам, поэтому выполнять это действие не рекомендуется.

Настройка прав для выполнения команды sudo

- Переключитесь с текущего пользователя на пользователя root. Для этого выполните команду:

```
su -
```

- Введите пароль пользователя **root**, заданный при установке операционной системы.

- Откройте файл /etc/sudoers. Для этого выполните команду:

```
visudo
```

ПРИМЕЧАНИЕ. В результате указанный файл откроется в редакторе vi. Подробнее об использовании редактора см. в статье [Работа с редактором vi](#).

4. Раскомментируйте строку `WHEEL_USERS ALL=(ALL) ALL`, удалив символ `#`, и сохраните файл.
5. Проверьте, что пользователь добавлен в группу **wheel**. Пример:


```
cat /etc/group | grep wheel
```

 При необходимости включите пользователя в группу:


```
usermod -aG wheel <имя пользователя>
```
6. Перезайдите в ОС под текущим пользователем.

Astra Linux Common Edition 2.12.40

ВАЖНО. В инструкции приводятся готовые примеры кода и командных строк. При копировании текста убедитесь в отсутствии лишних пробелов, переносов строк и спецсимволов.

1. Получите список обновлений. Для этого выполните команды:


```
sudo apt update
sudo apt install apt-transport-https ca-certificates curl software-properties-common gnupg2
```
2. Добавьте ключ официального репозитория **Docker**:


```
sudo curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -
```
3. Добавьте репозиторий **Docker** в систему в зависимости от типа прав пользователя:
 - при работе под пользователем с правами **root** выполните команду:


```
echo "deb [arch=amd64] https://download.docker.com/linux/debian buster stable" >> /etc/apt/sources.list
```
 - при работе под другим пользователем выполните команду:


```
sudo bash -c "echo 'deb [arch=amd64] https://download.docker.com/linux/debian buster stable' >> /etc/apt/sources.list"
```
4. Обновите информацию из репозитория:


```
sudo apt update
```
5. Убедитесь, что установка Docker будет производиться из репозитория **Docker**, а не из репозитория по умолчанию. Для этого выполните команду:


```
sudo apt-cache policy docker-ce
```

 Пример ответа:


```
docker-ce:
  установлен: (нет)
  кандидат: 5:20.10.6~3-0~debian-buster
```

ПРИМЕЧАНИЕ. Если вместо ответа возвращается ошибка или версия предлагаемого к установке пакета ниже 20, обратитесь в службу поддержки.
6. Установите Docker:


```
sudo apt install docker-ce
```

ПРИМЕЧАНИЕ. Вместе с Docker обязательно должны установиться пакеты **containerd.io** и **docker-ce-cli**.

7. Проверьте, что сервис запустился:

```
sudo systemctl status docker
```

Команда должна вернуть статус «active», выделенный зеленым цветом.

8. Включите сервис вручную, если он не запустился автоматически. Для этого выдайте разрешение на автоматический запуск `docker` вместе с системой, последовательно выполнив команды:

```
sudo systemctl enable --now docker
```

```
sudo systemctl start docker
```

ПРИМЕЧАНИЕ. Рекомендуется всегда запускать сервис `docker` с помощью команды `sudo`. При необходимости можно настроить запуск сервиса `docker` без использования `sudo`. Для этого нужно добавить пользователя текущего сеанса в группу `docker`. При этом следует учитывать, что добавление пользователей в группу `docker` повышает риск несанкционированного доступа к системным файлам, поэтому выполнять это действие не рекомендуется.

Установка Docker Compose

Для работы сервисов Smart Services установите Docker Compose версии 1.29.2.

ВАЖНО. В инструкции приводятся готовые примеры кода и командных строк. При копировании текста убедитесь в отсутствии лишних пробелов, переносов строк и спецсимволов.

1. Получите права на запуск скриптов. Для этого выполните команду:

```
source change-mod.sh
```

2. С привилегиями суперпользователя выполните команду:

```
./docker-compose-install.sh
```

ВАЖНО. Если на сервере уже использовалась более ранняя версия Docker Compose, то она будет заменена на версию 1.29.2. Существующий исполняемый файл `/usr/local/bin/docker-compose` будет переименован в `/usr/local/bin/docker-compose_backup`.

3. Проверьте, что установлена нужная версия Docker Compose, выполнив команду:

```
docker-compose -v
```

Если установка выполнена успешно, команда выведет версию 1.29.2.

Пример ответа:

```
docker-compose version 1.29.2, build 8a1c60f6
```

Развертывание

Для развертывания сервисов Smart Services используются docker-контейнеры. Каждому сервису соответствует отдельный контейнер.

ВАЖНО. В инструкции приводятся готовые примеры кода и командных строк. При копировании текста убедитесь в отсутствии лишних пробелов, переносов строк и спецсимволов.

Порядок развертывания

1. Скопируйте на сервер архив с Docker-образами сервисов и скриптами их развертывания и распакуйте с помощью команды:

```
tar -xf <Имя архива>
```

ПРИМЕЧАНИЕ. Если планируется использование нескольких серверов, скопируйте архив на каждый из них.

2. После распаковки архива перейдите в корневой каталог сервисов. Для этого выполните команду:

```
cd <путь до корневого каталога сервисов>
```

ВАЖНО. Далее все команды выполняются из корневого каталога сервисов, который обозначается как ./

3. Получите права на запуск скриптов. Для этого выполните команду:

```
source change-mod.sh
```

Если сервисы развертываются в операционной системе Альт Сервер, выдайте другим пользователям права на запись в папки, в которых хранятся трейсы, с помощью команды:

```
sudo chmod -R o+w ./etc/elastic-apm
```

4. Загрузите Docker-образы сервисов. Для этого выполните команду:

```
./load-images.sh
```

5. Убедитесь, что все образы загружены. Для этого выполните команду:

```
docker images
```

Количество образов должно быть равно количеству архивов в каталоге ./images.o

6. Инициализируйте инструмент управления взаимодействием контейнеров Docker Swarm, если он не был инициализирован ранее. Для этого выполните команду:

```
./swarm-init.sh
```

В результате текущий сервер станет управляющим по отношению к другим серверам.

Если на сервере используется несколько IP-адресов, то явно задайте адрес, который будет использоваться для обращения к серверу. Пример:

```
./swarm-init.sh 172.16.64.104
```

7. Задайте метку сервера, на котором будут запускаться docker контейнеры с сервисами. Для этого выполните команду:

```
./swarm-add-label-to-node.sh <метка> <имя узла>
```

где:

- **<метка>** – значение параметра **manager_node_label** из конфигурационного файла `.env`. По умолчанию все сервисы запускаются на управляющем сервере, где значение метки – **main**. При необходимости значение метки можно изменить. Если сервисы запускаются на одном сервере, то метка для них должна быть одинаковой;
- **<имя узла>** – имя текущего сервера.

Чтобы узнать имя текущего сервера, выполните команду:

```
hostname
```

Чтобы узнать значение текущей метки, выполните команду:

```
docker node inspect <имя узла>
```

В соответствующем параметре ответа будет указано текущее значение. Пример:

```
"Spec": {
  "Labels": {
    "service": "main"
  },
  "Role": "manager",
  "Availability": "active"
}
```

8. Добавьте рассчитанные лимиты объема оперативной памяти для Text Extractor Service и Fact Extractor Learn Service в конфигурационный файл `deploy.yml`. Для этого измените значение параметра **memory** в соответствующих секциях файла, указав рассчитанный объем в мегабайтах. Сервису Text Extractor Service соответствует секция **dtes**, сервису Fact Extractor Learn Service – **dfesl**:

deploy:

```
mode: replicated
replicas: ${dtes_replicas}
placement:
  constraints:
    - "node.labels.service == ${dtes_node_label}"
  max_replicas_per_node: 1
restart_policy:
  condition: any
  delay: 15s
  window: 180s
resources:
  limits:
    memory: 16384m
```

ПРИМЕЧАНИЕ. Если для распознавания и обучения используется два экземпляра сервиса Fact Extractor Learn Service, то в первом случае лимит оперативной памяти имеет постоянное значение – 2 ГБ. Лимит оперативной памяти для обучения определяется с учетом количества обрабатываемых документов.

Если для распознавания и обучения используется один экземпляр Fact Extractor Learn Service, лимит оперативной памяти для распознавания также будет равен 2 ГБ, а итоговый лимит памяти равен объему памяти для обучения согласно таблице, которая используется для расчета объема памяти для обучения моделей классификации и извлечения фактов. Например, при 2 ГБ для распознавания и 16 ГБ для обучения на 400 документах итоговый лимит 16 ГБ. Таким образом, итоговый лимит равен наибольшему значению лимита для экземпляра сервиса.

9. Убедитесь, что в конфигурационном файле `.env`:
- в параметре **timezone** указан текущий часовой пояс;
 - в параметре **dss_allow_local_path** указано значение **true**;
 - в параметре **dtcs_allow_local_path** указано значение **True**.
10. Запросите ключ активации сервисов в службе поддержки и скопируйте полученный файл `ActivationKey.aak` в каталог `./etc/smart-service/license`. После установки и запуска сервисов активация выполнится автоматически.
11. Запустите контейнеры сервисов Smart Services. Для этого выполните команду:


```
./start-services.sh
```

ПРИМЕЧАНИЕ. На запуск уходит несколько минут.

12. Проверьте, что контейнеры запустились, одним из способов:
- с помощью команды:


```
docker service ls
```

 В ответе в колонке «Replicas» проверьте состояние запущенных сервисов: **0/1** – сервис еще не запущен, **1/1** – сервис запущен;
 - с помощью Portainer. Для этого:
 - а) Войдите в Portainer на сетевой адрес сервера на порт, указанный в конфигурационном файле `.env` в секции **Portainer**. По умолчанию **8999**.
 - б) Задайте логин и пароль для входа.
 - в) На домашней странице выберите текущий Endpoint – **primary**.
 - г) Перейдите в раздел «Stacks» и нажмите на стек **smart-services**.
 - д) В выпадающем пункте **Items per page** внизу страницы установите значение **100**.
 - е) В колонке «Scheduling Mode» отслеживайте состояние запущенных контейнеров:
 - **0/1** – сервис еще не запущен;
 - **1/1** – сервис запущен.

ПРИМЕЧАНИЕ. Чтобы увидеть лог-файлы, разверните строку нужного сервиса по кнопке  и в столбце **Actions** нажмите на кнопку **Logs**.

13. Если необходимо изменить настройки, например, изменить порт подключения к Smart Service, внесите изменения в конфигурационный файл `.env`.
14. После изменения параметров конфигурации перезапустите сервисы. Для этого последовательно:
- остановите сервисы:


```
./stop-services.sh
```
 - проверьте, что все сервисы остановлены, с помощью команды:


```
docker ps -a
```
 - повторно запустите сервисы:


```
./start-services.sh
```

В результате сервисы запустятся с учетом изменений, внесенных в конфигурационный файл.

15. При необходимости разверните отдельные сервисы на [других серверах](#).

16. Выполните [завершающие работы](#).

В результате развертывания:

- из архива загружаются docker-образы;
- запускаются контейнеры с сервисами Smart Services.

Подключение дополнительных серверов

Если сервисы развертываются на разных серверах, например Text Extractor Service устанавливается на отдельный, то их нужно связать между собой. Связь представляет собой Swarm-кластер, где есть [управляющий](#) сервер и подключаемые к нему дополнительные сервера.

Если подключаются дополнительные сервера, то на управляющем должны развертываться следующие сервисы:

- сервисы Smart Services: Smart Service, Identity Service, Publishing Service;
- Elastic APM;
- Influxdb;
- Grafana;
- Telegraf.

Другие сервисы и компоненты можно разворачивать на дополнительных серверах.

Подключение еще одного сервера выполняется в следующем порядке:

1. Остановите на управляющем сервере сервисы Smart Services. Для этого последовательно выполните команду:

```
./stop-services.sh
```

2. Проверьте, что все сервисы остановлены, с помощью команды:

```
docker ps -a
```

3. Сгенерируйте команду с ключом для подключения дополнительного сервера. Для этого выполните команду:

```
./swarm-get-token.sh
```

4. Выполните полученную команду на подключаемом сервере, чтобы создать связь с управляющим сервером.

5. Проверьте, что сервер подключен успешно. Для этого на управляющем сервере выполните команду:

```
docker node ls
```

6. Измените значение метки дополнительного сервера. Для этого в секции **ServicePlacement** конфигурационного файла .env задайте значения параметра **node_label** для сервиса, развернутого отдельно. Например:

```
dtes_node_label = dtes
```

ПРИМЕЧАНИЕ. Значение параметра **node_label** должно отличаться от [параметра manager_node_label](#) для управляющего сервера.

7. Задайте метку для вновь подключенного сервера. Для этого выполните команду на управляющем сервере:

```
./swarm-add-label-to-node.sh <метка> <имя узла>
```

где:

- **<метка>** – значение параметра **node_label** из конфигурационного файла .env. По умолчанию **main**;
- **<имя узла>** – имя подключенного сервера.

Пример:

```
./swarm-add-label-to-node.sh dtes SMART-SERVER-2
```

8. Повторите на подключаемом сервере [шаги 1-5](#) инструкции по развертыванию сервисов.

ВАЖНО. Путь до корневого каталога сервисов должен быть одинаковым на всех серверах.

9. Запустите [контейнеры](#).

Завершение развертывания

1. Импортируйте *модели извлечения фактов*, если факты должны извлекаться на основе обучаемых моделей. Для этого запустите скрипт `import_models.py` из каталога `./etc/factextractor-learn-service/trained_models` с параметрами:

- **-url.** Путь до сервиса Smart Service. Параметр обязателен для заполнения;
- **-login.** Логин пользователя для аутентификации. Если параметр не указан, используется анонимная аутентификация (при условии, что она включена);
- **-password.** Пароль пользователя для аутентификации. Указывается, только когда указан логин.

Пример команды:

```
python import_models.py -url http://smart:61100
```

Пример команды с дополнительными параметрами:

```
python import_models.py -url http://smart:61100 -login user -password 123
```

ПРИМЕЧАНИЕ. В зависимости от настроек ОС может потребоваться указание конкретной версии Python, например `python3.7`, а вместо имени сервера – его IP-адрес.

В результате работы скрипта в Smart Services инициализируются базовые модели извлечения фактов.

2. Импортируйте базовые *модели классификации документов*.

Для **первичной загрузки моделей в новый классификатор** запустите скрипт `import_models.py` из каталога `./etc/textclassifier-service/trained_models` с параметрами:

- **-url.** Путь до сервиса Smart Service. Параметр обязателен для заполнения;
- **-login.** Логин пользователя для аутентификации. Если параметр не указан, используется анонимная аутентификация (при условии, что она включена);
- **-password.** Пароль пользователя для аутентификации. Указывается, когда указан логин.

Пример команды:

```
python import_models.py -url http://smart:61100
```

Пример команды с дополнительными параметрами:

```
python import_models.py -url http://smart:61100 -login user -password 123
```

ПРИМЕЧАНИЕ. В зависимости от настроек ОС может потребоваться указание конкретной версии Python, например python3.7, а вместо имени сервера – его IP-адрес.

В результате создаются классификаторы по виду документа и по первым страницам, в которые импортируются соответствующие базовые модели классификации.

ПРИМЕЧАНИЕ. Если уже существуют классификаторы с именами **Базовый классификатор по виду документа** и **Базовый классификатор первая/не первая страница**, то в результате работы скрипта обновятся их модели.

Для **загрузки моделей классификации в существующий классификатор** запустите тот же скрипт с теми же параметрами, указав дополнительно:

- **-id_kind.** Идентификатор классификатора документов по виду. Параметр не обязателен для заполнения;
- **-id_first.** Идентификатор классификатора по первым страницам. Параметр не обязателен для заполнения.

Пример команды:

```
python import_models.py -url http://smart:61100 -id_kind 1 -id_first 2
```

Пример команды с дополнительными параметрами:

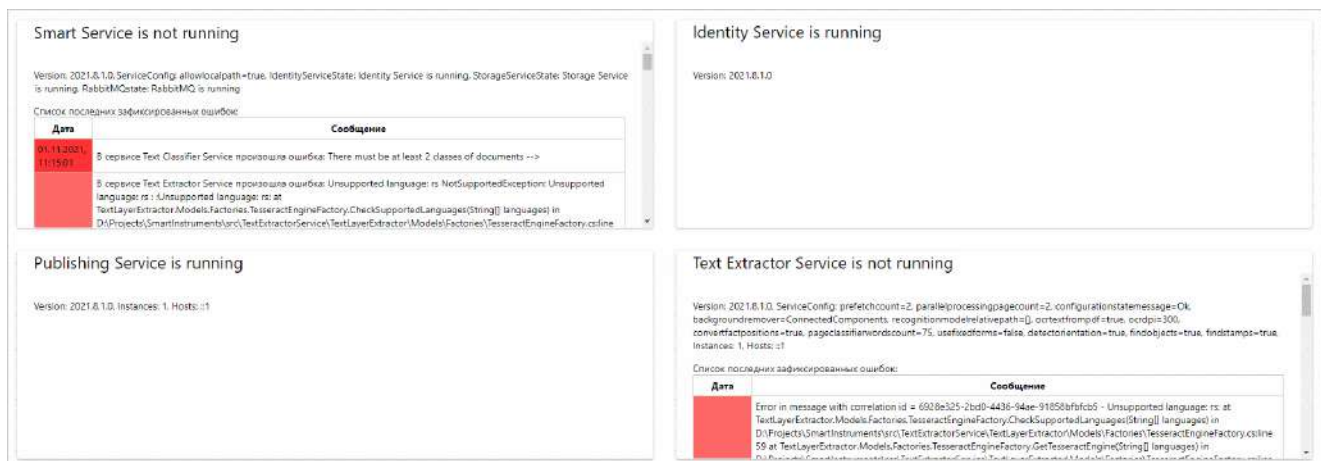
```
python import_models.py -url http://smart:61100 -id_kind 1 -id_first 2 -  
login user -password 123
```

В результате в классификатор по виду документа с идентификатором **1** и в классификатор по страницам с идентификатором **2** импортируются базовые модели классификации.

ПРИМЕЧАНИЕ. Если классификаторов с такими идентификаторами нет, возвратится соответствующая ошибка.

Запуск и настройка инструмента checker.html

Инструмент checker.html позволяет быстро проверить состояние сервисов после запуска, а также проанализировать последние ошибки в работе. Если они есть, то отображается таблица со списком последних 10 зафиксированных ошибок. В таблице фиксируется дата и время ошибки, а также ее полное описание:



Состояние **is running** означает, что сервис запущен и работает.

Запуск

- если все сервисы Smart Services установлены на один сервер, в дистрибутиве из корневого каталога откройте файл checker.html;
- если сервисы Smart Services установлены на разные серверы, то настройте инструмент для распределенной работы и запустите файл checker.html на сервере со Smart Service.

Особенности:

- если используется Linux без графической оболочки, то настройте инструмент для распределенной работы и запускайте checker.html на сервере с графической оболочкой.

Настройка инструмента для распределенной работы

Чтобы настроить инструмент для распределенной работы, на сервере со Smart Service в файле checker.html укажите имя или IP-адрес дополнительного сервера. Например, укажите имя или IP-адрес сервера с Text Extractor Service, если он развернут отдельно.

Особенности:

- в операционной системе на базе Linux рекомендуется указывать IP-адреса сервисов;
- если используется Linux без графической оболочки, скопируйте файл checker.html на сервер с графической оболочкой и внесите в него изменения с помощью любого текстового редактора. При этом измените IP-адреса всех серверов.

Пример настройки утилиты с помощью Notepad++:

```

<body>
  <div id="container">
    </div>
  <script>
    const service_urls = {"Smart Service": "http://smart:61100/info", "Identity Service": "http://smart:9030/info",
      "Publishing Service": "http://smart:9057/info", "Text Extractor Service": "http://intelligent:9027/api/info/state",
      "Text Classifier Service": "http://smart:9007/", "Image Classifier Service": "http://smart:9008/",
      "Fact Extractor Service": "http://smart:9017/", "Fact Extractor Rules Service": "http://smart:9018/",
      "Fact Extractor Learn Service": "http://smart:9019/",
      "Object Detection Service": "http://smart:9047/"}
  </script>

```

Установка инструментов Elasticsearch

Для загрузки лог-файлов в Elasticsearch и их последующего анализа:

- [установите поисковую систему Elasticsearch и панель управления Kibana](#);
- [настройте шаблоны индекса в Kibana](#);
- [установите и настройте сервис Filebeat](#).

Подробнее о работе инструментов см. в документации [Elasticsearch](#), [Kibana](#), [Filebeat](#).

Установка поисковой системы Elasticsearch

В службе поддержки запросите дистрибутивы Elasticsearch и Kibana. Дистрибутивы предоставляются бесплатно, дополнительную лицензию приобретать не нужно.

Чтобы установить Elasticsearch:

1. Распакуйте ZIP-архив с дистрибутивом Elasticsearch в отдельную локальную папку.
2. В командной строке перейдите в папку <Локальная папка с Elasticsearch>\bin и выполните команду:

```
elasticsearch-service.bat install
```

В результате устанавливается служба Elasticsearch.

3. В оснастке **Службы** запустите службу **Elasticsearch (elasticsearch-service-x64)**.
Настройте автозапуск службы, чтобы она сама стартовала после перезагрузки операционной системы. Для этого откройте свойства службы, в раскрывающемся списке **Тип запуска** выберите значение **Автоматически** и нажмите на кнопку **ОК**.
4. В браузере перейдите по адресу <http://localhost:9200>, откроется стартовая страница Elasticsearch.

Пример:

```
{
  "name" : "elastic",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "tdrMX2SyQOm-bT724ViLoA",
  "version" : {
    "number" : "7.16.3",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "4e6e4eab2297e949ec994e688dad46290d018022",
    "build_date" : "2022-01-06T23:43:02.825887787Z",
    "build_snapshot" : false,
    "lucene_version" : "8.10.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

5. [Настройте](#) обратный прокси-сервер (Reverse Proxy) и ограничение по IP-адресам для безопасного подключения к Elasticsearch.
6. [Настройте](#) фильтрацию запросов в IIS для корректного индексирования текстов документов.

Чтобы установить панель управления Kibana:

1. Распакуйте ZIP-архив с дистрибутивом Kibana в отдельную локальную папку.
2. Запустите файл <Папка с Kibana>\bin\kibana.bat. В результате откроется окно консольного приложения Kibana. Чтобы продолжить работу с Kibana, не закрывайте окно. Работу Kibana можно возобновить в любой момент, запустив указанный bat-файл.
3. В браузере перейдите по адресу <http://localhost:5601>, откроется стартовая страница Kibana.

Обратный прокси-сервер

Установка ролей и компонентов

1. На сервере с установленной поисковой системой Elasticsearch запустите оснастку **Диспетчер сервера**.
2. В меню **Управление** выберите действие **Добавить роли и компоненты**. Запустится мастер добавления ролей и компонентов.
3. На странице «Перед началом работы» нажмите на кнопку **Далее**.
4. На странице «Выбор типа установки» выберите пункт **Установка ролей или компонентов** и нажмите на кнопку **Далее**.
5. На странице «Выбор целевого сервера» выберите пункт **Выберите сервер из пула серверов** и укажите сервер, на котором необходимо добавить роли и компоненты. Нажмите на кнопку **Далее**.
6. На странице «Выбор ролей сервера» последовательно раскройте узлы **Веб-сервер (IIS)**, **Веб-сервер**, **Безопасность** и установите флажок **IP-адрес и ограничения домена**. Нажмите на кнопку **Далее**.
7. На странице «Выбор компонентов» нажмите на кнопку **Далее**.

8. На странице «Подтверждение установки компонентов» нажмите на кнопку **Установить**.
9. На странице «Ход установки» убедитесь, что установка роли успешно завершена, и нажмите на кнопку **Заккрыть**.
10. Установите расширение [URL Rewrite](#).

Настройка обратного прокси-сервера

1. В приложении **Диспетчер служб IIS** добавьте веб-сайт. Для этого на панели «Подключения» перейдите к узлу сайты и в контекстном меню выберите пункт **Добавить веб-сайт**.
2. В окне «Добавить веб-сайт» заполните поля:
 - **Имя сайта** – например, ElasticsearchSite;
 - **Физический путь** – расположение сайта Elasticsearch, например, **C:\inetpub\wwwroot\ElasticsearchSite**;
 - **Порт** – **9201**;
 - **Имя узла** – *.

Значения остальных полей оставьте без изменений. Нажмите на кнопку **ОК**.

3. На начальной странице созданного сайта выберите действие **Переопределение URL-адресов**.
4. На странице «Переопределение URL-адресов» на панели выберите действие **Добавить правила**.
5. В открывшемся окне «Добавить правила» выберите шаблон **Обратный прокси-сервер** и нажмите на кнопку **ОК**.
6. В открывшемся окне «Добавить правила обратного прокси-сервера» настройте правило для входящего трафика. В качестве имени сервера, на который будут перенаправляться HTTP-запросы, укажите **localhost:9200**. Затем установите флажок **Включить загрузку SSL** и нажмите на кнопку **ОК**.
7. На начальной странице сайта выберите действие **Ограничения IP-адресов и доменов**.
8. На странице «Ограничения IP-адресов и доменов» выберите действие **Добавить разрешающий элемент**.
9. В окне «Добавить разрешающее правило ограничения» в поле **Определенный IP-адрес** укажите IP-адрес, к которому должен быть разрешен доступ. Нажмите на кнопку **ОК**.
10. На панели выберите действие **Изменить параметры**. В открывшемся окне в раскрывающемся списке «Доступ для неуказанных клиентов» выберите пункт **Запретить** и нажмите на кнопку **ОК**.
11. На начальной странице сайта перейдите в группу «Управление» и выберите действие **Редактор конфигураций**.
12. На открывшейся странице редактора конфигураций в раскрывающемся списке **Раздел** выберите **system.web/httpRuntime**.
13. В табличной части для атрибута **relaxedUrlToFileSystemMapping** укажите значение **true**.

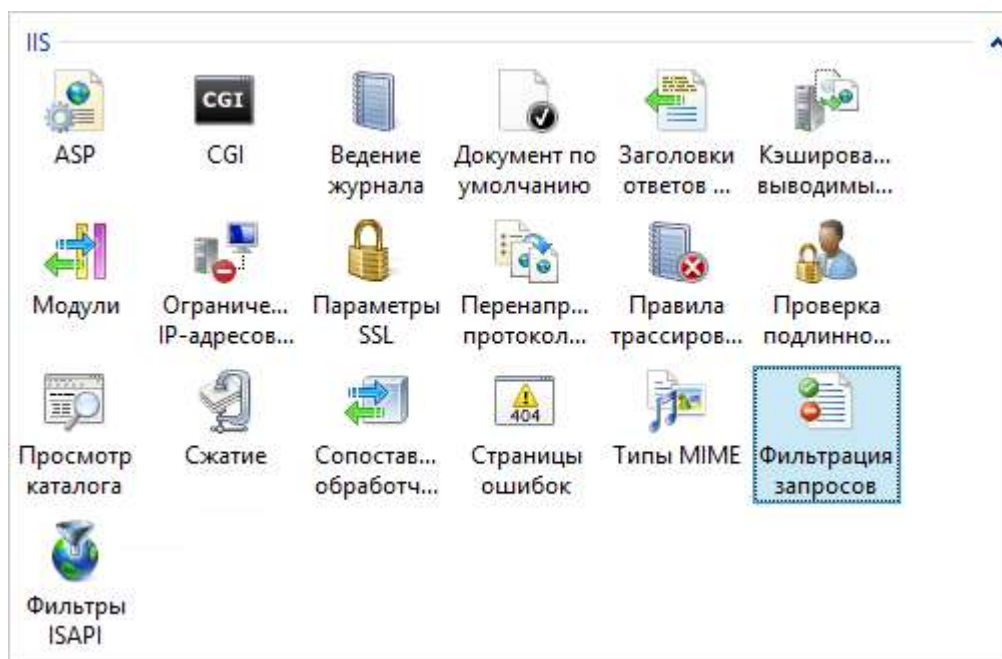
- В атрибуте **requestPathInvalidCharacters** укажите произвольное значение, переместите курсор на другую строку, затем вернитесь и удалите введенные символы. В результате значение атрибута должно полностью очиститься. Нажмите на кнопку **Применить**.
- Проверьте заданные настройки. Для этого в браузере на другом компьютере откройте страницу <http://<Имя компьютера с Elasticsearch>:9201>. В результате откроется тестовая страница Elasticsearch.

ПРИМЕЧАНИЕ. Если для доступа в Интернет используется прокси-сервер, то в его исключения добавьте адрес компьютера с установленными компонентами Elasticsearch.

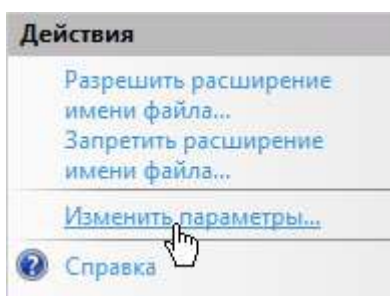
Фильтрация запросов в IIS

По умолчанию IIS обрабатывает запросы, содержимое которых не превышает 30 МБ. Для корректной работы необходимо с запасом увеличить максимально допустимый размер запроса.

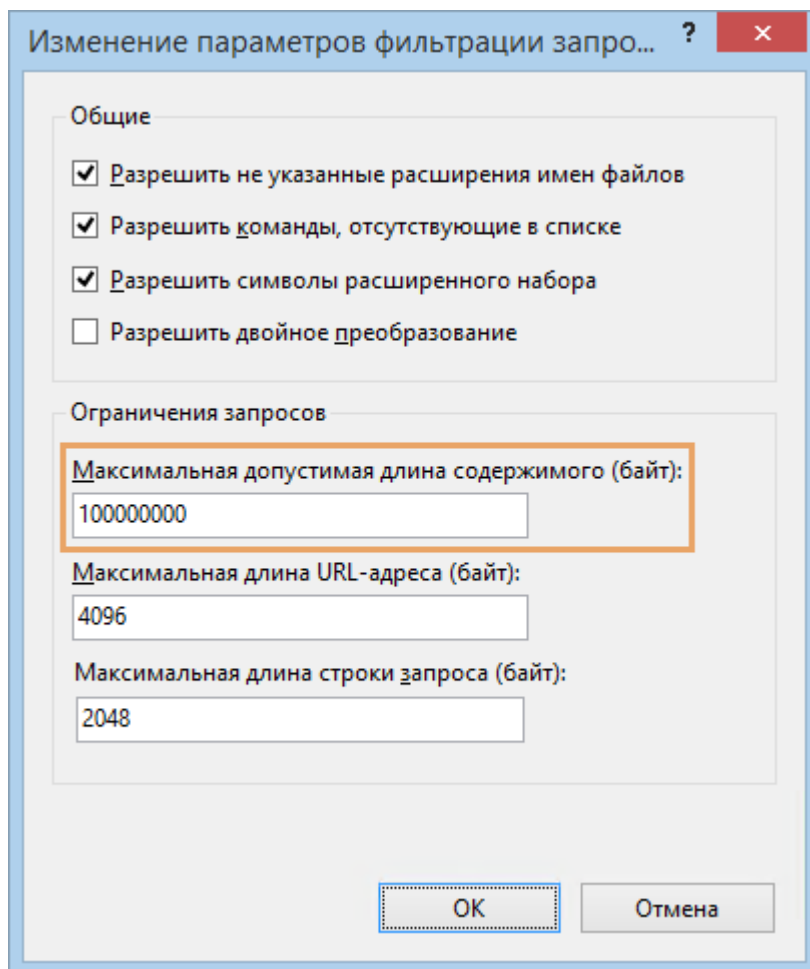
- В приложении **Диспетчер служб IIS** перейдите к сайту Elasticsearch и на его начальной странице выберите действие **Фильтрация запросов**:



- На странице «Фильтрация запросов» на панели действий выберите действие **Изменить параметры**:



3. В открывшемся окне в поле **Максимальная допустимая длина содержимого (байт)** укажите значение **100 000 000** (100 МБ):



4. Нажмите на кнопку **OK**.

Установка сервиса Filebeat

ВАЖНО. В инструкции приводятся готовые примеры кода и командных строк. При копировании текста убедитесь в отсутствии лишних пробелов и спецсимволов.

Автоматическая загрузка лог-файлов в Elasticsearch выполняется с помощью сервиса Filebeat. Сервис входит в линейку продуктов Elasticsearch и распространяется бесплатно.

Чтобы настроить загрузку лог-файлов с помощью Filebeat:

1. Скачайте и установите Filebeat 7.14.0. Для этого выполните команды:

```
curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.14.0-linux-x86_64.tar.gz
tar xzvf filebeat-7.14.0-linux-x86_64.tar.gz
```

2. Настройте сервис с помощью конфигурационного файла **filebeat.yml**. Путь до конфигурационного файла по умолчанию: `/etc/filebeat/filebeat.yml`.

В секции **filebeat.inputs** укажите параметры получения и декодировки файлов:

- **paths** – пути до лог-файлов сервисов в формате <Папка с лог-файлами сервиса>/*.log>. Пути до папок с лог-файлами указываются в конфигурационных файлах каждого сервиса;
- **json.keys_under_root – true;**
- **json.overwrite_keys – true;**
- **json.add_error_key – true;**
- **json.message_key – message.**

В секции **output.elasticsearch** укажите параметры загрузки данных в Elasticsearch:

- **hosts** – адрес Elasticsearch.

В остальных секциях оставьте значения по умолчанию.

Пример настройки:

```
filebeat.inputs:
- type: log
  enabled: true
  paths:
    - C:\Smart_Services\SmartService\SmartService\Logs\*.log
    - C:\Smart_Services\TextExtractorService\logs\*.log
    - C:\Smart_Services\PublishingService\Logs\*.log
    - C:\Smart_Services\TextClassifierService\logs\*.log
    - C:\Smart_Services\ImageClassifierService\logs\*.log
    - C:\Smart_Services\FactExtractorServices\Base\logs\*.log
    - C:\Smart_Services\FactExtractorServices\Learn\logs\*.log
    - C:\Smart_Services\FactExtractorServices\Rules\logs\*.log
    - C:\Smart_Services\ObjectDetectionService\logs\*.log
  json.keys_under_root: true
  json.overwrite_keys: true
  json.add_error_key: true
  json.message_key: message

output.elasticsearch:
  hosts: ["elasticsearchhost:9200"]
```

3. Запустите Filebeat с помощью команды:

```
sudo ./filebeat
```

Обновление сервисов

Перед обновлением ознакомьтесь с типовыми требованиями и убедитесь, что ваше аппаратное и программное обеспечение актуально. Подробнее см. в документе «Smart Services. Типовые требования к программному и аппаратному обеспечению».

Алгоритм обновления сервисов Smart Services зависит от конфигурации установки:

- сервисы установлены на [одном сервере](#);
- сервисы установлены на [разных серверах](#).

Сервисы установлены на одном сервере

1. [Удалите предыдущую версию](#) сервисов Smart Services.
2. [Разверните новую версию](#) сервисов Smart Services.

ПРИМЕЧАНИЕ. При необходимости при развертывании на [шаге 1](#) замените конфигурационный файл .env на аналогичный конфигурационный файл, сохраненный от предыдущей установленной версии.

Сервисы установлены на разных серверах

1. На основном сервере [удалите предыдущую версию](#) сервисов Smart Services.
2. На дополнительных серверах удалите Docker-образы сервисов средствами Docker. Подробнее см. в документации Docker статью [docker rmi](#).
3. [Разверните новую версию](#) сервисов Smart Services.
ПРИМЕЧАНИЕ. При необходимости при развертывании на [шаге 1](#) замените конфигурационный файл .env на аналогичный конфигурационный файл, сохраненный от предыдущей установленной версии.
4. [Подключите дополнительные серверы](#).

Удаление сервисов

Чтобы удалить сервисы Smart Services с сервера:

1. Сохраните текущий конфигурационный файл `.env`.
2. Остановите сервисы. Для этого выполните команду:

```
./stop-services.sh
```
3. Проверьте, что все сервисы остановлены, с помощью команды:

```
docker ps -a
```
4. Удалите Docker-образы сервисов средствами Docker. Подробнее см. в документации Docker статью [docker rmi](#).
5. Удалите корневой каталог, в котором находятся скрипты управления сервисами.
6. При необходимости удалите хранилища данных (volumes) Portainer, Grafana, InfluxDB, Postgres, RabbitMQ и Minio средствами Docker. Например, если не планируете использовать текущие хранилища данных с [новой версией сервисов](#). Подробнее см. в документации Docker статью [docker volume rm](#).

ВАЖНО. Данные из хранилищ удаляются без возможности восстановления.