

Развертывание on-site

Версия 8.0



Эта документация предоставляется с ограничениями на использование и защищена законами об интеллектуальной собственности. За исключением случаев, прямо разрешенных в вашем лицензионном соглашении или разрешенных законом, вы не можете использовать, копировать, воспроизводить, переводить, транслировать, изменять, лицензировать, передавать, распространять, демонстрировать, выполнять, публиковать или отображать любую часть в любой форме или посредством любые значения. Обратный инжиниринг, дизассемблирование или декомпиляция этой документации, если это не требуется по закону для взаимодействия, запрещены.

Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления и не может гарантировать отсутствие ошибок. Если вы обнаружите какие-либо ошибки, сообщите нам о них в письменной форме.

Содержание

Настроить глобальный поиск	7
Настроить сервис глобального поиска (версия 3.0)	7
Системные требования к клиентскому компьютеру	21
Развернуть базу данных MS SQL	22
Общий порядок настройки сервера кэширования данных (Redis)	25
Проверить обязательные компоненты Windows	26
Проверить обязательные компоненты Windows Server 2016	28
Проверить обязательные компоненты Windows 10	38
Подготовка к установке Creatio .NET Core на Linux	39
Подготовить установочные файлы	39
Настроить ConnectionStrings.config	40
Настроить интеграцию с Google	42
Общий порядок установки Creatio	42
Установить Creatio NET Framework на Windows	42
Установить Creatio .NET Core на Linux	43
Настроить массовый поиск дублей	43
Компоненты сервиса массового поиска дублей	44
Настроить переменные окружения	45
Запустить контейнеры	45
Проверить успешность запуска контейнеров	45
Проверить логирование	45
Подключить функциональность массового поиска дублей в Creatio	45
Обновить версию массового поиска дублей	50
Рекомендуемые работы по обслуживанию сервиса	51
Системные требования к серверам	51
Общие рекомендации	54
Каналы связи	54
Требования к серверам для сервиса синхронизации Exchange Listener	55
Требования к программному обеспечению	55
Развернуть базу данных Oracle	56
Настроить ConnectionStrings.config для MS SQL Server	58
Настроить ConnectionStrings.config	58
Обязательные настройки ConnectionStrings.config	59
Необязательные настройки ConnectionStrings.config	60
Установить Creatio .NET Core на Linux	61
Способ 1. Установить Creatio .NET Core непосредственно на Linux	61

Способ 2. Установить Creatio .NET Core на Linux с использованием Docker	64
Настроить работу Creatio .NetCore на https	66
Настроить интеграцию с лендингами	67
Настроить Redis Sentinel	68
Особенности Sentinel	68
Минимальная отказоустойчивая конфигурация Redis Sentinel	69
Проблема разделения сети	70
Системные требования	71
Установить и настроить Redis Sentinel	71
Настроить Creatio для работы с Redis Sentinel	71
Часто задаваемые вопросы по глобальному поиску и дедупликации	73
Как узнать, какая версия глобального поиска используется?	73
Как запустить переиндексацию сайта?	73
Как включить логирование работы сервиса глобального поиска?	74
Какие метрики или системы слежения можно использовать для мониторинга работы глобального поиска?	74
Как настроить доступ к Elasticsearch по паролю?	74
Как добавить в индекс новый объект или изменить настройки индексируемых полей существующих объектов?	75
Как развернуть Elasticsearch на нескольких серверах с указанием одного URL? Как настроить кластеризацию?	76
Что делать, если не отображается настройка “Правила поиска дублей”?	76
Можно ли использовать сервисы глобального поиска и массового поиска дублей одновременно с двумя приложениями Creatio?	76
Как сортируются результаты поиска?	77
Поддерживается ли аутентификация Windows в микросервисах?	77
Развернуть базу данных PostgreSQL (Linux)	77
I. Установить PostgreSQL	78
II. Создать пользователя PostgreSQL	78
III. Восстановить из бэкапа базу данных PostgreSQL	80
IV. Изменить владельца базы данных (опционально)	83
Настроить ConnectionStrings.config для Oracle Database	83
Настроить ConnectionStrings.config	84
Обязательные настройки ConnectionStrings.config	84
Необязательные настройки ConnectionStrings.config	85
Настроить уведомления о разрешении обращений	85
Настроить Redis Cluster	86
Отказоустойчивость Redis Cluster	86
Системные требования Redis Cluster	86
Установка и настройка Redis Cluster	87
Настройка Creatio для работы с Redis Cluster	88

Развернуть базу данных PostgreSQL (Windows)	89
I. Установить PostgreSQL	89
II. Создать пользователя PostgreSQL	90
III. Восстановить из бэкапа базу данных PostgreSQL	92
Настроить ConnectionStrings.config для PostgreSQL	94
Настроить ConnectionStrings.config	94
Обязательные настройки ConnectionStrings.config	95
Необязательные настройки ConnectionStrings.config	95
Сервис машинного обучения	96
Подготовка к установке сервиса машинного обучения	96
Установить сервис машинного обучения	98
Обновить компоненты сервиса машинного обучения	100
Настроить сервис обогащения данных	101
Настроить Web.config	101
Дополнительная настройка Web.config (только для Oracle)	102
Настроить сервис синхронизации Exchange Listener	103
Определить конфигурацию сервиса Exchange Listener	104
Способы развертывания сервиса Exchange Listener	105
Настроить сервис Exchange Listener на стороне Creatio	110
Диагностика настроек Exchange Listener	111
Настроить запуск массовых рассылок	113
Настроить интеграцию с облачным сервисом отправки рассылок Cloud Email Service (для пользователей on-site)	113
Настроить список доменов отправителя	115
Выполнить дополнительные настройки интеграции с сервисом отправки рассылок	115
Разрешить мониторинг email-рассылок в Creatio on-site	116
Настроить сервер приложения на IIS	116
Настроить сайт приложения в IIS	117
Добавить пул приложения	119
Настроить горизонтальное масштабирование	125
Общий порядок развертывания	126
Установить балансировщик HAProxy	129
Настроить балансировщик HAProxy	130
Настроить вебсокет	134
Проверить настройки вебсокетов для Windows Server 2012 или Windows server 2016	134
Часто задаваемые вопросы по настройке вебсокетов	136
Настроить безопасный доступ к portalu	136
Перевести Creatio с HTTP на HTTPS	138
Настроить IIS	138

Настроить Web.config	143
Настроить систему управления версиями для среды разработки	144
Установить SVN и создать хранилище для Creatio	145
Подключить созданное хранилище к Creatio	146
Часто задаваемые вопросы по установке Creatio	149
Какие компоненты IIS (Internet Information Services) нужны для работы приложения Creatio?	149
Как перевести приложение с протокола HTTP на HTTPS?	150
Под каким пользователем выполняется первый вход в систему?	150
Влияет ли число активных пользователей Creatio на число пользователей Microsoft SQL Server?	151
Настроить доступ к сервису чатов	151
Продукты Creatio на платформе .NET Core	151
Реализация поддержки .NET Core в продуктах Creatio	151
Жизненный цикл продуктов на платформе .NET Framework и .NET Core	152
Разработка функциональности с одновременной поддержкой .NET Framework и .NET Core	153
Миграция с приложения .NET Framework на приложение .NET Core	153
Развертывание приложения на платформе .NET Core	153
Настроить авторизацию интегрированных приложений по протоколу OAuth 2.0	153
Установить и настроить Identity Service	154
Настроить интеграцию с Identity Service на стороне Creatio	157
Настроить авторизацию приложений по протоколу OAuth 2.0	158
Настроить интеграцию с файловым хранилищем S3	159
Шаг 1. Настроить хранилище S3	160
Шаг 2. Настроить хранение файлов	160

Настроить глобальный поиск

ПРОДУКТЫ: **ВСЕ ПРОДУКТЫ**

Сервис глобального поиска (Global Search Service) создан для интеграции ElasticSearch с Creatio и выполняет следующие функции:

- **Регистрирующую:**
 - Подписывает клиента, создавая индекс в ElasticSearch и сохраняет связь индекс-приложение.
 - Отключает клиента, по требованию удаляя индекс в ElasticSearch.
- **Транспортную:**
 - Участвует в процессе индексации — забирает данные из базы данных.

Для настройки глобального поиска необходимы два отдельных физических или виртуальных сервера (“сервер 1” и “сервер 2”) с установленной ОС Linux. Для расчета требований к серверам воспользуйтесь [калькулятором системных требований](#).

Развертывание компонентов глобального поиска выполняется на ПО Docker. С перечнем поддерживаемых ОС вы можете ознакомиться в [документации Docker](#).

На заметку. В зависимости от потребностей вашей компании можно использовать Docker Community Edition (CE) или Enterprise Edition (EE). Подробную информацию вы найдете в [документации Docker](#).

Важно. Для настройки глобального поиска необходимы базовые знания администрирования ПО Docker и ОС Linux.

Последовательность действий при настройке глобального поиска зависит от того, какую версию сервиса вы будете использовать. Для последней версии Creatio рекомендуется всегда использовать новейшую версию сервиса глобального поиска.

Для поддержания работоспособности сервиса и возможности быстрого восстановления данных после отказа, например, в случае перебоев с электроэнергией, рекомендуется раз в сутки выполнять резервное копирование Elasticsearch.

Если у вас возникнут вопросы в ходе настройки, то рекомендуем ознакомиться со статьей [Часто задаваемые вопросы по глобальному поиску и дедупликации](#).

Настроить сервис глобального поиска (версия 3.0)

На заметку. Настройки ниже актуальны для сервиса глобального поиска версии 3.0. Если вам необходимо настроить сервис глобального поиска более ранней версии, то ознакомьтесь с

описанием настроек в [документации Creatio версии 7.16](#).

Чтобы **обновить** глобальный поиск с версии 2.0 на версию 3.0, необходимо удалить все docker volume версии 2.0 на серверах 1 и 2 при помощи команды [docker-compose down -v](#) и повторно выполнить установку и настройку всех сервисов.

Компоненты глобального поиска

Развертывается на сервере 1:

- [elasticsearch](#) — поисковый движок.

Развертываются на сервере 2:

- [postgres](#) — база данных конфигурирования компонентов глобального поиска.
- [rabbitmq](#) — брокер сообщений.
- [redis](#) — хранилище данных, используемое для кеширования и быстрого действия.
- [gs-web-api](#) — web-сервис конфигурирования компонентов глобального поиска.
- [gs-web-indexing-service](#) — web-сервис для обработки запросов точечного индексирования данных из системы.
- [gs-search-service](#) — web-сервис поиска данных, проху для elasticsearch.
- [gs-scheduler](#) — планировщик задач индексации данных из Creatio в ElasticSearch.
- [gs-worker](#) — компонент индексирования данных из Creatio в ElasticSearch по задачам планировщика.
- [gs-worker-replay](#) — компонент, обрабатывающий результаты индексации (результаты работы gs-worker-a).
- [gs-worker-single](#) — компонент точечной индексации данных бизнес-процессов в ElasticSearch по запросу из бизнес-процесса.
- [gs-worker-single-replay](#) — компонент, обрабатывающий исключения в процессе точечной индексации (результаты работы gs-worker-single).
- [gs-worker-single-task](#) — компонент для постановки задач компоненту gs-worker-single.
- [gs-worker-querried-single-task](#) — компонент для формирования задач компоненту gs-worker-single.

Для настройки компонентов скачайте исходные файлы. [Скачать файлы](#).

Список портов, используемых компонентами глобального поиска:

Важно. Если вы используете FireWall, убедитесь, что все перечисленные порты доступны и открыты.

Название компонента	Исходящий порт	Входящий порт	Примечание
gs-web-api		81	Входящий порт настраивается переменной WEB_API_PORT
gs-web-indexing-service		82	Входящий порт настраивается переменной WEB_INDEXING_SERVICE_PORT
gs-search-service	9200	83	Входящий порт настраивается переменной SEARCH_SERVICE_PORT
gs-worker	9200		Необходимо соединение с сервером, на котором расположен elasticsearch
gs-worker-single	9200		Необходимо соединение с сервером, на котором расположен elasticsearch
elasticsearch		9200	

Последовательность действий для настройки глобального поиска

1. Установить Docker на физическую или виртуальную машину с операционной системой Linux.
2. Установить Docker-Compose.
3. Установить ElasticSearch.
4. Настроить переменные контейнеров.
5. Установить и запустить компоненты Global Search Service.
6. Подключить функциональность глобального поиска в Creatio.

Установить Docker

Для развертывания компонентов глобального поиска необходимо установить Docker на операционную систему Linux. Для установки воспользуйтесь инструкцией в [документации Docker](#).

Для проверки установленной версии Docker запустите команду **docker --version** на linux-машине.

Установить Docker-Compose

Для установки Docker-Compose воспользуйтесь инструкцией в [документации Docker](#).

Установить ElasticSearch

На заметку. В данной инструкции описано развертывание ElasticSearch в Docker-Compose. Вы

также можете развернуть его как daemon OS, что позволит обойтись без предварительной установки Docker и Docker-Compose. Для этого воспользуйтесь инструкцией в [документации ElasticSearch](#).

Для установки ElasticSearch:

1. На сервере, выделенном для установки ElasticSearch (сервер 1), зайдите в папку /opt.
2. Распакуйте в открытую папку архив с установочными файлами. [Скачать архив](#)
3. Перейдите в папку с компонентами /opt/compose/elasticsearch и выполните команду:

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

Команда может выполняться несколько минут.

4. После выполнения команды проверьте, что в логах нет ошибок. Для этого выполните команду

```
docker logs es-01
```

Настроить переменные контейнеров

Все контейнеры компонентов глобального поиска конфигурируются из файла с переменными среды (environment variables). Переменные содержатся в файле /opt/compose/services/.env. Установите значения переменных, отредактировав этот файл.

Название переменной	Описание	Значение по умолчанию
GS_ES_URL	Внешний хост ElasticSearch, который используется для доступа из Creatio. Необходимо указать ip-адрес сервера, на котором развернут ElasticSearch.	http://elasticsearch-publicip:9200
CURRENT_SERVER_IP	Внешний ip-адрес сервера, на котором развернуты сервисы глобального поиска (сервер 2).	10.0.0.1

На заметку. Чтобы проверить внешний ip-адрес сервера, выполните команду

```
hostname -I | awk '{ print $1 }'
```

Дополнительные переменные, которые управляют параметрами индексации данных в ElasticSearch

Название переменной	Описание	Значение по умолчанию
GS_DB_INCREMENT_DAYS	Количество дней, которое необходимо проиндексировать за одну итерацию планировщика. Для сравнения используются данные колонки ModifiedOn записей системы.	500 дней
GS_DB_FILL_QUEUE_INTERVAL	Интервал сбора данных из БД Creatio регулярным планировщиком. Чем меньше этот параметр, тем выше нагрузка на БД Creatio, но быстрее происходит первичная индексация.	30000 (указывается в миллисекундах)

Запустить контейнеры с компонентами Global Search Service

Важно. Для корректной работы контейнеров необходимо, чтобы UTC-время на linux-машине, на которой установлен Docker, соответствовало UTC-времени на сервере БД Creatio. Допустимое отклонение — до пяти минут. Иначе глобальный поиск может индексировать не все записи.

1. На сервере, выделенном для установки компонентов глобального поиска (сервер 2), зайдите в папку `opt`.
2. Распакуйте в открытую папку архив с установочными файлами. [Скачать архив](#)
3. Перейдите в папку с компонентами `/opt/compose/services` и выполните команду:

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

Проверить успешность запуска контейнеров

Для просмотра всех запущенных контейнеров глобального поиска введите в консоли команду:

```
docker ps --filter "label=service=gs" -a --format "table {{.Names}}\t{{.Ports}}\t{{.Status}}\t{{
```

Все запущенные контейнеры должны быть со статусом **Up**.

Логирование

По умолчанию в контейнере логирование происходит в `stdout` и `stderr`.

На заметку. `docker logs --tail 100 gs-worker-01` выводит 100 последних строк логов из контейнера `gs-worker`.

На заметку. При старте возможны ситуации, когда контейнер `mysql` или `rabbitmq` временно недоступны, так как они запускаются позже остальных компонентов. В этом случае дождитесь, пока в логах не отобразится сообщение об удачном подключении и запуске контейнера, например: `"Now listening on: http://[::]:80 Application started. Press Ctrl+C to shut down"`.

Подключить сервис глобального поиска в Creatio

Действия на сервере

Для подключения глобального поиска к Creatio выполните следующие действия на сервере 2, где находятся компоненты глобального поиска:

1. Для http-запросов установите утилиту `api-get` `install curl` или `yum install curl`.

```
apt-get install curl
```

2. Выполните http-запрос на регистрацию сайта в глобальном поиске, указав:

- a. `[DATABASE_TYPE]` — тип базы данных Creatio (`mssql`, `postgresql` или `oracle`).
- b. `[DATABASE_CONNECTION_STRING]` — строка подключения к БД Creatio.
- c. `[SITE_NAME]` — название сайта Creatio, например, `my-test-site`.
- d. `[SERVER2_IP_ADDRESS]` — IP-адрес Linux сервера, на котором развернуты компоненты глобального поиска.

```
curl -v -X POST -d '{"databaseType": "[DATABASE_TYPE]", "databaseConnectionString": "[DATAI
```

Пример для MS SQL. `curl -v -X POST -d '{"databaseType": "mssql", "databaseConnectionString": "Server=myserver\\mssql2016; Database=my-test-site; User Id=my-login; Password='my-password'; Connection Timeout=10"}' -H "Content-Type: application/json" http://[SERVER2_IP_ADDRESS]:81/sites/my-test-site`

Пример для PostgreSQL. `server=[SERVER_IP];port=5432;database=[DB_NAME];user id=[USER_NAME];password=[PASSWORD];timeout=10;commandtimeout=400;maxpoolsize=1024`

3. Выполните http-запрос на подключение поиска к ранее добавленному сайту, указав:

- a. `[SITE_NAME]` — название сайта Creatio, например, `my-test-site`.

- b. [*TEMPLATE_NAME*] — название шаблона поиска, который используется в Elasticsearch. Перечень доступных к использованию шаблонов приведен в таблице ниже.
- c. [*SERVER2_IP_ADDRESS*] — IP-адрес Linux сервера, на котором развернуты компоненты глобального поиска.

```
curl -v -X POST -d '{"templateName": "[TEMPLATE_NAME]"}' -H "Content-Type: application/json"
```

Пример. `curl -v -X POST -d '{"templateName": "default.json"}' -H "Content-Type: application/json" http://[SERVER2_IP_ADDRESS]:81/sites/my-test-site/search`

На заметку. Приведенный запрос вернет URL к созданному индексу в Elasticsearch. URL нужно сохранить и использовать в приведенном ниже sql-скрипте установки системных настроек.

Важно. Чтобы изменить шаблон поиска, выполните DELETE запрос на `/sites/{siteName}/search` и http-запрос на подключение поиска, описанный выше, после чего будет выполнена полная переиндексация сайта.

Все доступные шаблоны поиска и их характеристики приведены в таблице ниже:

	Старый шаблон (версия 1.6)	default.json	ngram_2.json	ngram_3.json	without_ng
Поиск по части слова	+	-	+	+	-
Поиск по слову с опечатками	+	-	+	+	-
Поиск по части номера телефона в средствах связи	+	+	+	+	-
Нечеткий поиск по средствам связи	+	+	+	+	-
Поиск по...

Поиск с перестановкой слов	+	+	+	+	+
Поиск по точному совпадению	+	+	+	+	+
Поиск по двум символам	-	-	+	-	-
Примерная скорость поиска (чем меньше, тем лучше)		1x	13x	7x	<1x
Размер индекса на elasticsearch (чем меньше, тем лучше)		1x	4x	2,5x	<1x
Время полной первичной индексации (чем меньше, тем лучше)		1x	1,8x	1,4x	<1x

Действия на стороне Creatio для СУБД MS SQL

1. Включите в Creatio функциональность (Feature Toggle) глобального поиска (GlobalSearch, GlobalSearch_V2 GlobalSearchRelatedEntityIndexing), выполнив sql-скрипт:

```

DECLARE @GS_REIndexingFeature NVARCHAR(50) = 'GlobalSearchRelatedEntityIndexing';
DECLARE @GS_REIndexingFeatureId UNIQUEIDENTIFIER = (SELECT TOP 1 Id FROM Feature WHERE
Code = @GS_REIndexingFeature);

DECLARE @GlobalSearchFeature NVARCHAR(50) = 'GlobalSearch';
DECLARE @GlobalSearchFeatureId UNIQUEIDENTIFIER = (SELECT TOP 1 Id FROM Feature WHERE Code =

DECLARE @GlobalSearchV2Feature NVARCHAR(50) = 'GlobalSearch_V2';
DECLARE @GlobalSearchV2FeatureId UNIQUEIDENTIFIER = (SELECT TOP 1 Id FROM Feature WHERE Code
DECLARE @allEmployeesId UNIQUEIDENTIFIER = 'A29A3BA5-4B0D-DE11-9A51-005056C00008';

IF (@GlobalSearchFeatureId IS NOT NULL)

```

```

BEGIN
    IF EXISTS (SELECT * FROM AdminUnitFeatureState WHERE FeatureId = @GlobalSearchFeatureId)
        UPDATE AdminUnitFeatureState SET FeatureState = 1 WHERE FeatureId = @GlobalSearchFeature
    ELSE
        INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@all
    END;
ELSE
    BEGIN
        SET @GlobalSearchFeatureId = NEWID()
        INSERT INTO Feature (Id, Name, Code) VALUES (@GlobalSearchFeatureId, @GlobalSearchFeature,
        INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@allEm
    END;

IF (@GlobalSearchV2FeatureId IS NOT NULL)
    BEGIN
        IF EXISTS (SELECT * FROM AdminUnitFeatureState WHERE FeatureId = @GlobalSearchV2FeatureId)
            UPDATE AdminUnitFeatureState SET FeatureState = 1 WHERE FeatureId = @GlobalSearchV2Featu
        ELSE
            INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@all
    END;
ELSE
    BEGIN
        SET @GlobalSearchV2FeatureId = NEWID()
        INSERT INTO Feature (Id, Name, Code) VALUES (@GlobalSearchV2FeatureId, @GlobalSearchV2Feat
        INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@allEm
    END;

IF (@GS_REIndexingFeatureId IS NOT NULL)
    BEGIN
        IF EXISTS (SELECT * FROM AdminUnitFeatureState WHERE FeatureId = @GS_REIndexingFeatureId)
            UPDATE AdminUnitFeatureState SET FeatureState = 1 WHERE FeatureId = @GS_REIndexingFeatureId
        ELSE
            INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@allEmp
    END;
ELSE
    BEGIN
        SET @GS_REIndexingFeatureId = NEWID()
        INSERT INTO Feature (Id, Name, Code) VALUES (@GS_REIndexingFeatureId, @GS_REIndexingFeature,
        INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@allEmpl
    END;

```

2. Установите значения системных настроек:

- a. "GlobalSearchUrl" — полный путь в elasticsearch с учетом индекса. Значение настройки возвращается при выполнении запроса на web-аpi для добавления поиска для сайта. Пример строки: `http://[SERVER2_IP_ADDRESS]:83/indexname`.
- b. "GlobalSearchConfigServiceURL" — ссылка на API глобального поиска. Значение по умолчанию: `http://[SERVER2_IP_ADDRESS]:81`.

- c. "GlobalSearchIndexingApiUrl" — ссылка на сервис моментальной индексации. Значение по умолчанию: `http://[SERVER2_IP_ADDRESS]:82`.

```
UPDATE SysSettingsValue
SET TextValue = [укажите URL к индексу ElasticSearch, строка типа: http://[SERVER2_IP_ADDRESS]:82]
WHERE SysSettingsId = (SELECT TOP 1 Id FROM SysSettings WHERE Code = 'GlobalSearchUrl')

UPDATE SysSettingsValue
SET TextValue = [укажите URL к Global Search Service, строка типа: http://[SERVER2_IP_ADDRESS]:82]
WHERE SysSettingsId = (SELECT TOP 1 Id FROM SysSettings WHERE Code = 'GlobalSearchConfigServiceUrl')

UPDATE SysSettingsValue
SET TextValue = [укажите URL к Global Search Indexing Service, строка типа: http://[SERVER2_IP_ADDRESS]:82]
WHERE SysSettingsId = (SELECT TOP 1 Id FROM SysSettings WHERE Code = 'GlobalSearchIndexingApiUrl')
```

3. Перезапустите приложение Creatio, очистите redis и выполните вход в приложение.

Действия на стороне Creatio для СУБД Oracle

1. Включите в Creatio функциональность (Feature Toggle) глобального поиска (GlobalSearch, GlobalSearch_V2 GlobalSearchRelatedEntityIndexing), выполнив sql-скрипт:

```
CREATE OR REPLACE FUNCTION
generate_uuid return varchar2 is
    v_uuid varchar2(38);
    v_guid varchar2(32);
BEGIN
    v_guid := sys_guid();
    v_uuid := lower(
        '{' ||
        substr(v_guid, 1,8) || '-' ||
        substr(v_guid, 9,4) || '-' ||
        substr(v_guid, 13,4) || '-' ||
        substr(v_guid, 17,4) || '-' ||
        substr(v_guid, 21) ||
        '}'
    );
    RETURN v_uuid;
END;
/

DECLARE
    GS_REIndexingFeature VARCHAR(50) := 'GlobalSearchRelatedEntityIndexing';
    GS_REIndexingFeatureId VARCHAR(38) := NULL;
    GS_REIndexingFeatureId_GUID VARCHAR(38) := generate_uuid();
```



```

GlobalSearchFeature VARCHAR(50) := 'GlobalSearch';
GlobalSearchFeatureId VARCHAR(38) := NULL;
GlobalSearchFeatureId_GUID VARCHAR(38) := generate_uuid();
GlobalSearchV2Feature VARCHAR(50) := 'GlobalSearch_V2';
GlobalSearchV2FeatureId VARCHAR(38) := NULL;
GlobalSearchV2FeatureId_GUID VARCHAR(38) := generate_uuid();
allEmployeesId VARCHAR(38) := '{7F3B869F-34F3-4F20-AB4D-7480A5FDF647}';
State_GlobalSearch VARCHAR(1) := NULL;
State_GlobalSearchV2 VARCHAR(1) := NULL;
State_GS_REI VARCHAR(1) := NULL;

BEGIN
    SELECT MAX("Id") INTO GlobalSearchFeatureId FROM "Feature" WHERE "Code" = GlobalSearchFeatu
    SELECT MAX("Id") INTO GlobalSearchV2FeatureId FROM "Feature" WHERE "Code" = GlobalSearchV2F
    SELECT MAX("Id") INTO GS_REIndexingFeatureId FROM "Feature" WHERE "Code" = GS_REIndexingFeatu

    SELECT MAX("FeatureState") INTO State_GlobalSearch FROM "AdminUnitFeatureState" WHERE "Feat
    SELECT MAX("FeatureState") INTO State_GlobalSearchV2 FROM "AdminUnitFeatureState" WHERE "Fe
    SELECT MAX("FeatureState") INTO State_GS_REI FROM "AdminUnitFeatureState" WHERE FeatureId"

    IF (GlobalSearchFeatureId IS NULL) THEN
        INSERT INTO "Feature" ("Id", "Name", "Code") VALUES (GlobalSearchFeatureId_GUID, Global
        INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VAL
    ELSE
        IF (State_GlobalSearch IS NOT NULL) THEN
            UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = GlobalSea
        ELSE
            INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId")
        END IF;
    END IF;

    IF (GlobalSearchV2FeatureId IS NULL) THEN
        INSERT INTO "Feature" ("Id", "Name", "Code") VALUES (GlobalSearchV2FeatureId_GUID, Glob
        INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VAL
    ELSE
        IF (State_GlobalSearchV2 IS NOT NULL) THEN
            UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = GlobalSea
        ELSE
            INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId")
        END IF;
    END IF;

    IF (GS_REIndexingFeatureId IS NULL) THEN
        INSERT INTO "Feature" ("Id", "Name", "Code") VALUES (GS_REIndexingFeatureId_GUID, GS_REIndex
        INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (
    ELSE
        IF (State_GS_REI IS NOT NULL) THEN
            UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = GS_REIndexingFeatu
        ELSE

```

```

INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState","FeatureId") VALUES (
END IF;
END IF;

END;

```

2. Установите значения системных настроек:

- “GlobalSearchUrl” — полный путь в elasticsearch с учетом индекса. Значение настройки возвращается при выполнении запроса на web-api для добавления поиска для сайта. Пример строки: `http://[SERVER2_IP_ADDRESS]:83/indexname`.
 - GlobalSearchConfigServiceURL” — ссылка на API глобального поиска. Значение по умолчанию: `http://[SERVER2_IP_ADDRESS]:81`.
 - “GlobalSearchIndexingApiUrl” — ссылка на сервис моментальной индексации. Значение по умолчанию: `http://[SERVER2_IP_ADDRESS]:82`.
- Для этого выполните следующий скрипт:

```

DECLARE
URL_SETTING_ID VARCHAR(38) := NULL;
CONFIG_URL_SETTING_ID VARCHAR(38) := NULL;
IND_API_SETTING_ID VARCHAR(38) := NULL;

URL_VAL_ID VARCHAR(38) := NULL;
CONFIG_URL_VAL_ID VARCHAR(38) := NULL;
IND_API_VAL_ID VARCHAR(38) := NULL;

SYS_ADMIN_UID VARCHAR(38) := '{A29A3BA5-4B0D-DE11-9A51-005056C00008}';

ES_IND VARCHAR(500) := '[укажите URL к индексу ElasticSearch, строка, сл. типа - http://[SE
CONFIG_URL VARCHAR(500) := '[укажите URL к Global Search Service, строка следующего типа: h
IND_API_URL VARCHAR(500) := '[укажите URL к Global Search Indexing Service, строка следующе
BEGIN
SELECT "Id" INTO URL_SETTING_ID FROM "SysSettings" WHERE "Code" = 'GlobalSearchUrl';
SELECT "Id" INTO CONFIG_URL_SETTING_ID FROM "SysSettings" WHERE "Code" = 'GlobalSearchConfi
SELECT "Id" INTO IND_API_SETTING_ID FROM "SysSettings" WHERE "Code" = 'GlobalSearchIndexing

SELECT MAX("Id") INTO URL_VAL_ID FROM "SysSettingsValue" WHERE "SysSettingsId" = URL_SETTI
SELECT MAX("Id") INTO CONFIG_URL_VAL_ID FROM "SysSettingsValue" WHERE "SysSettingsId" = CO
SELECT MAX("Id") INTO IND_API_VAL_ID FROM "SysSettingsValue" WHERE "SysSettingsId" = IND_A

IF (URL_VAL_ID IS NULL)
THEN
INSERT INTO "SysSettingsValue"
("SysSettingsId", "SysAdminUnitId", "IsDef", "TextValue")
VALUES
(URL_SETTING_ID, SYS_ADMIN_UID, '1', ES_IND);
ELSE

```

```

        UPDATE "SysSettingsValue" SET "TextValue" = ES_IND WHERE "SysSettingsId" = URL_SETTING_
    END IF;

    IF (CONFIG_URL_VAL_ID IS NULL)
    THEN
        INSERT INTO "SysSettingsValue"
            ("SysSettingsId", "SysAdminUnitId", "IsDef", "TextValue")
        VALUES
            (CONFIG_URL_SETTING_ID, SYS_ADMIN_UID, '1', CONFIG_URL);
    ELSE
        UPDATE "SysSettingsValue" SET "TextValue" = CONFIG_URL WHERE "SysSettingsId" = CONFIG_U
    END IF;

    IF (IND_API_VAL_ID IS NULL)
    THEN
        INSERT INTO "SysSettingsValue"
            ("SysSettingsId", "SysAdminUnitId", "IsDef", "TextValue")
        VALUES
            (IND_API_SETTING_ID, SYS_ADMIN_UID, '1', IND_API_URL);
    ELSE
        UPDATE "SysSettingsValue" SET "TextValue" = IND_API_URL WHERE "SysSettingsId" = IND_API
    END IF;
END;

```

3. Перезапустите приложение Creatio, очистите redis и выполните вход в приложение.

Действия на стороне Creatio для СУБД PostgreSQL

1. Включите в Creatio функциональность (Feature Toggle) глобального поиска (GlobalSearch, GlobalSearch_V2 GlobalSearchRelatedEntityIndexing), выполнив sql-скрипт:

```

DO $$

DECLARE
    GlobalSearchFeature VARCHAR(50) := 'GlobalSearch';
    GlobalSearchFeatureId uuid;
    GlobalSearchV2Feature VARCHAR(50) := 'GlobalSearch_V2';
    GlobalSearchV2FeatureId uuid;
    GS_RelatedEntityIndexingFeature VARCHAR(50) := 'GlobalSearchRelatedEntityIndexing';
    GS_RelatedEntityIndexingFeatureId uuid;
    allEmployeesId uuid := 'A29A3BA5-4B0D-DE11-9A51-005056C00008';

BEGIN

    SELECT "Id" INTO GlobalSearchFeatureId FROM "Feature"
    WHERE "Code" = GlobalSearchFeature
    LIMIT 1;

```

```

IF (GlobalSearchFeatureId IS NOT NULL)
    THEN
        IF EXISTS (SELECT * FROM "AdminUnitFeatureState" WHERE "FeatureId" = GlobalSearchFe
            UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = Global
        ELSE
            INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "Feature
        END IF;
    ELSE
        GlobalSearchFeatureId := uuid_generate_v4();
        INSERT INTO "Feature" ("Id", "Name", "Code") VALUES (GlobalSearchFeatureId, GlobalSear
        INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VA
    END IF;

    SELECT "Id" INTO GlobalSearchV2FeatureId FROM "Feature"
    WHERE "Code" = GlobalSearchV2Feature
    LIMIT 1;
    IF (GlobalSearchV2FeatureId IS NOT NULL)
        THEN
            IF EXISTS (SELECT * FROM "AdminUnitFeatureState" WHERE "FeatureId" = GlobalSearchV2Fe
                UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = GlobalS
            ELSE
                INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId"
            END IF;
        ELSE
            GlobalSearchV2FeatureId := uuid_generate_v4();
            INSERT INTO "Feature" ("Id", "Name", "Code") VALUES (GlobalSearchV2FeatureId, GlobalSe
            INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VA
        END IF;

    SELECT "Id" INTO GS_RelatedEntityIndexingFeatureId FROM "Feature" WHERE "Code" =GS_RelatedE
    IF (GS_RelatedEntityIndexingFeatureId IS NOT NULL)
        THEN
            IF EXISTS (SELECT * FROM "AdminUnitFeatureState" WHERE "FeatureId" = Bulk_ES_DD_FeatureId)
            UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = GS_RelatedEntityInd
        ELSE
            INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (
        END IF;
        ELSE
            GS_RelatedEntityIndexingFeatureId := uuid_generate_v4();
            INSERT INTO "Feature" ("Id", "Name", "Code") VALUES (GS_RelatedEntityIndexingFeatureId, GS_
            INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (
        END IF;
    END $$;

```

2. Установите значения системных настроек:

- a. "GlobalSearchUrl" — полный путь в elasticsearch с учетом индекса. Значение настройки возвращается при выполнении запроса на web-api для добавления поиска для сайта. Пример строки: `http://[SERVER2_IP_ADDRESS]:83/indexname`.

- b. GlobalSearchConfigServiceURL” — ссылка на API глобального поиска. Значение по умолчанию: `http://[SERVER2_IP_ADDRESS]:81`.
 - c. “GlobalSearchIndexingApiUrl” — ссылка на сервис моментальной индексации. Значение по умолчанию: `http://[SERVER2_IP_ADDRESS]:82`.
- Для этого выполните следующий скрипт:

```
UPDATE "SysSettingsValue"
SET "TextValue" = [укажите URL к индексу Elasticsearch, строка следующего типа: http://[SERVE
WHERE "SysSettingsId" = (SELECT "Id" FROM "SysSettings" WHERE "Code" = 'GlobalSearchUrl' LIM

UPDATE "SysSettingsValue"
SET "TextValue" = [укажите URL к Global Search Service, строка следующего типа: http://SERVER
WHERE "SysSettingsId" = (SELECT "Id" FROM "SysSettings" WHERE "Code" = 'GlobalSearchConfigSer

UPDATE "SysSettingsValue"
SET "TextValue" = [укажите URL к Global Search Indexing Service, строка, сл. типа - http://SE
WHERE "SysSettingsId" = (SELECT "Id" FROM "SysSettings" WHERE "Code" = 'GlobalSearchIndexingA
```

3. Перезапустите приложение Creatio, очистите redis и выполните вход в приложение.

Системные требования к клиентскому компьютеру

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Программные продукты на платформе Creatio — это web-приложения, которые не требуют установки клиентской части. Поскольку работа с приложением осуществляется в интернет-браузере, рекомендуемые параметры клиентского компьютера должны соответствовать требованиям браузера. Ориентировочные характеристики приведены в таблице:

Минимальное разрешение монитора	1280x768.
ЦПУ (CPU)	Pentium серии G/J или AMD A6-7xxx 3.0 Ghz и выше.
Оперативная память	Рекомендуемый объем — 4 ГБ.
НЖМД (HDD)	Рекомендуется 500 МБ свободного пространства.

На заметку. С рекомендуемыми параметрами для мобильных телефонов и планшетов вы можете ознакомиться в статье [“Системные требования к мобильным устройствам”](#).

Рекомендуемые требования к каналу связи соединения клиент-сервер — 512 Кбит/сек на одного активного пользователя.

На пользовательском компьютере должен быть установлен один из следующих браузеров:

Операционная система Microsoft Windows: <ul style="list-style-type: none"> • Windows 7; • Windows 8; • Windows 10 	Google Chrome последней официальной версии на дату релиза Creatio
	Mozilla Firefox последней официальной версии на дату релиза Creatio
	Microsoft Edge последней официальной версии на дату релиза Creatio
Операционная система OS X последней официальной версии	Apple Safari последней официальной версии на дату релиза Creatio

На заметку. Ознакомиться с датами выпуска версий Creatio вы можете в [календаре релизов](#).

Важно. Использование сторонних плагинов или расширений в браузерах может увеличить время загрузки страниц или вызвать их некорректное отображение.

Развернуть базу данных MS SQL

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Для работы с базой данных на сервере БД требуется установить Microsoft SQL Server Management Studio. Ознакомиться с описанием программы и скачать установочные файлы можно в [документации Microsoft](#).

На заметку. Развертывание Creatio с отказоустойчивостью на MS SQL успешно тестировалось. Для развертывания системы с высокой доступностью рекомендуется использовать группы доступности MS SQL Always On. Подробнее о технологии MS SQL Always On читайте в [документации Microsoft](#).

После установки Microsoft SQL Server Management Studio вам необходимо создать пользователей базы данных.

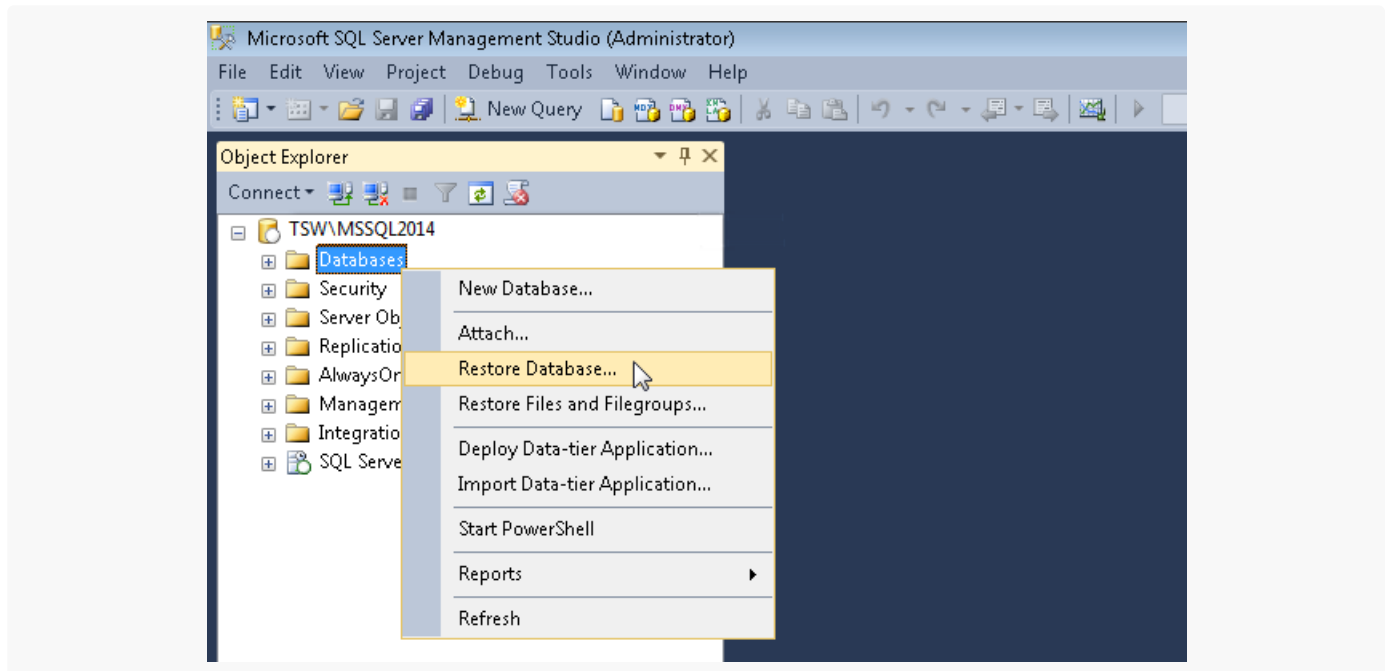
- Пользователь с ролью **"sysadmin"** и неограниченными полномочиями на уровне сервера базы данных — нужен для восстановления базы данных и настройки доступа к ней.
- Пользователь с ролью **"public"** и ограниченными полномочиями — используется для настройки безопасного подключения Creatio к базе данных через аутентификацию средствами MS SQL Server.

Подробнее о создании пользователей и настройке прав читайте в [документации Microsoft](#).

Для восстановления базы данных:

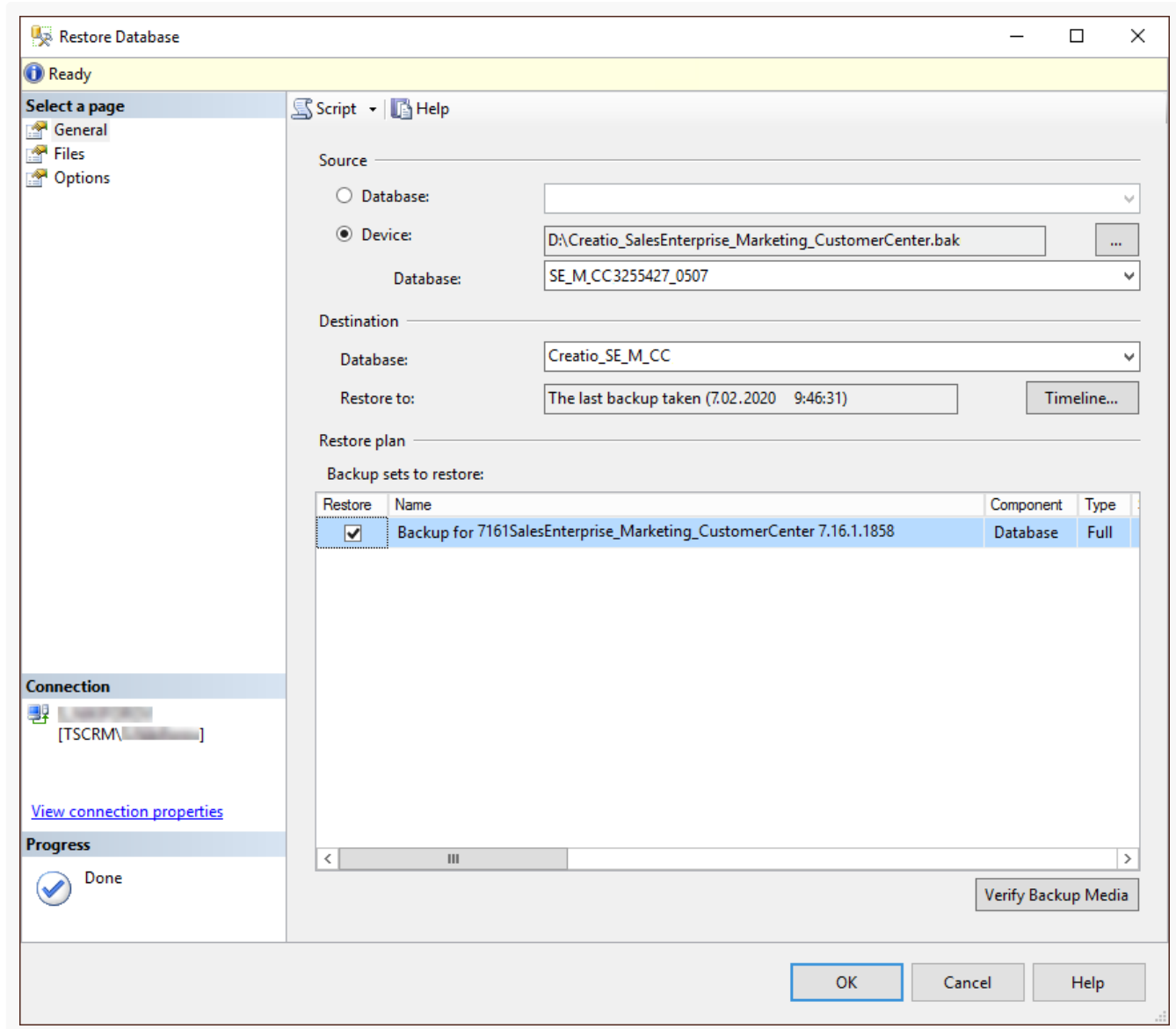
1. Авторизируйтесь в Microsoft SQL Server Management Studio как пользователь с ролью **"sysadmin"**.
2. Нажмите правой клавишей мыши по каталогу [*Databases*] и в контекстном меню выберите команду [*Restore Database*] ([Рис. 1](#)).

Рис. 1 — Выбор команды восстановления базы данных



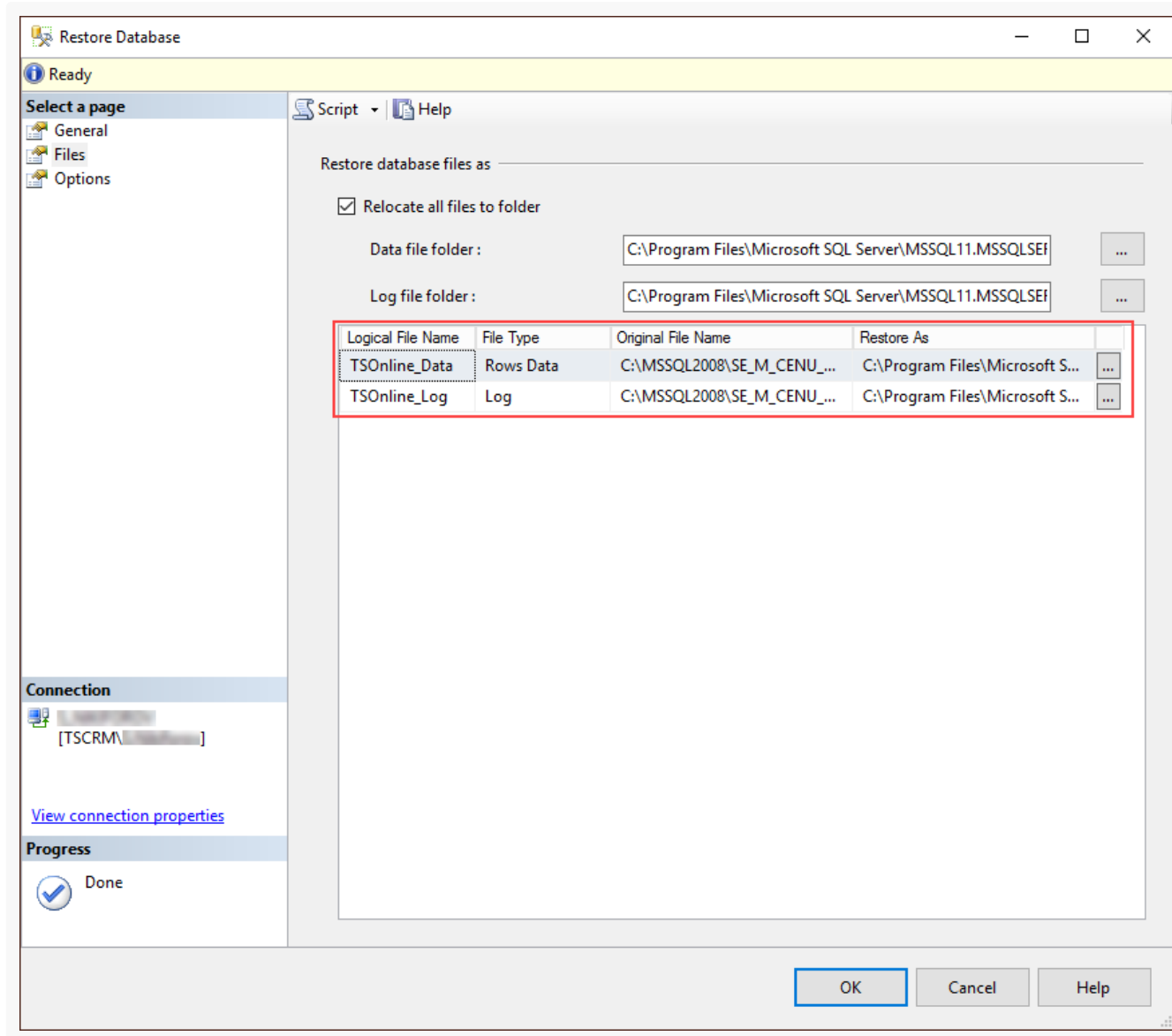
3. В окне [*Restore Database*]:
 - a. В поле [*Database*] введите название базы данных;
 - b. Выберите переключатель [*Device*] и укажите путь к файлу резервной копии базы данных. По умолчанию данный файл находится в директории ~\db с исполняемыми файлами Creatio ([Рис. 2](#)).

Рис. 2 — Выбор резервной копии базы данных



4. Укажите папку на сервере, в которой будет храниться развернутая база данных. Необходимо заранее создать папку, которая будет указываться для восстановления файлов базы данных, т.к. SQL сервер не имеет прав на создание директорий.
 - a. Перейдите на вкладку [Files].
 - b. В области [Restore the database files as] установите признак [Relocate all files and folders].
 - c. Укажите пути к папкам, в которые будут сохранены файлы базы данных **TS_Data.mdf** и **TS_Log.ldf** (Рис. 3).

Рис. 3 — Указание названий файлов и путей размещения файлов TS_Data.mdf и TS_Log.ldf.



5. Нажмите на кнопку [OK] и дождитесь завершения процесса восстановления базы данных.
6. Настройте для восстановленной базы возможность подключения пользователя MS SQL с ролью **"public"**, от имени которого приложение Creatio будет подключаться к базе данных:
 - a. В MS SQL Server Managment Studio найдите восстановленную базу данных Creatio.
 - b. Откройте вкладку [Security] выбранной базы данных.
 - c. В списке пользователей [Users] добавьте созданного ранее пользователя.
 - d. На вкладке [Membership] укажите роль **"db_owner"** — таким образом пользователю будет предоставлен неограниченный доступ к восстановленной базе.

Общий порядок настройки сервера кэширования данных (Redis)

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Использование сервера кэширования данных Redis позволит упростить выполнение трудоемких запросов к базе данных. Это ускоряет работу системы и снижает затраты ресурсов.

Пакет Redis доступен в стандартных репозиториях Debian. Ниже описана установка Redis на Debian и производных дистрибутивах, таких как Ubuntu и Linux Mint. Чтобы установить Redis:

1. Войдите в систему как администратор (root):

```
sudo su
```

2. Обновите список пакетов:

```
apt-get update
```

3. Установите Redis:

```
apt-get install redis-server
```

4. Настройте Redis таким образом, чтобы он запускался как системная служба **systemd**. Для этого:

- a. Откройте **redis.conf** в текстовом редакторе от имени пользователя root. Например, для этого можно использовать текстовый редактор Nano:

```
nano /etc/redis/redis.conf
```

- b. Найдите запись "**supervised no**". Замените запись на "**supervised systemd**".
- c. Сохраните изменения и закройте текстовый редактор.
- d. Перезагрузите сервер Redis:

```
systemctl restart redis-server
```

- e. Выйдите из root-сессии:

```
exit
```

Проверить обязательные компоненты Windows

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Перед созданием и настройкой web-сайта убедитесь, что на веб-сервере установлены:

- Компоненты Windows. Обратите внимание, что обязательно нужно установить компонент Microsoft Visual C++ 2010.
- Компоненты Web Server IIS.

Для обеспечения корректной компиляции приложения необходимо скачать и установить .NET Core SDK 3.1.301 и .NET Framework SDK v 4.7.2.

Пользователю, от имени которого в IIS запущен пул приложения, необходимо предоставить права на чтение, запись и удаление файлов и вложенных каталогов для каталога \Terrasoft.WebApp\Terrasoft.Configuration.

[Скачать 64-разрядный .NET Core SDK 3.1.301](#)

[Скачать 64-разрядный .NET Framework SDK v 4.7.2](#)

Важно. Для промышленной эксплуатации приложение Creatio должно быть развернуто на семействе ОС Windows Server. ОС Windows 10 может использоваться только для разработки конфигурации.

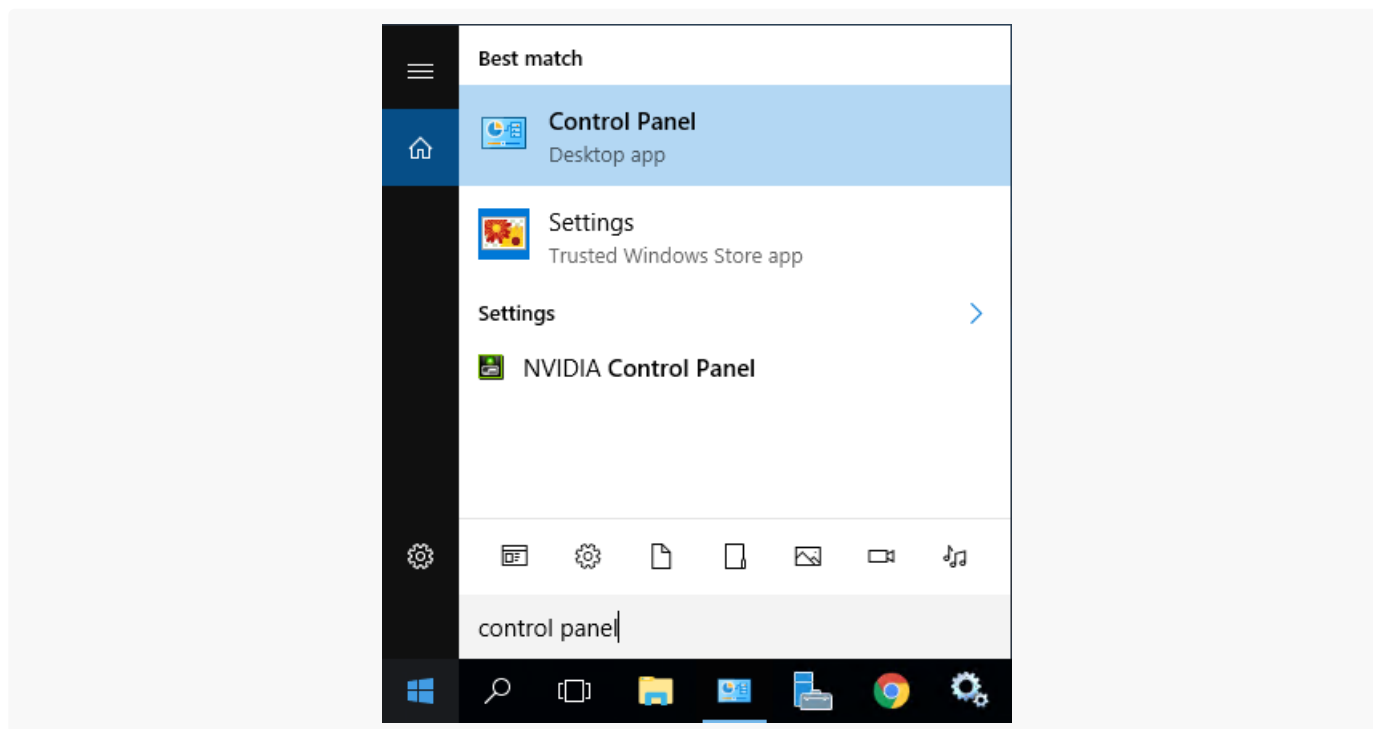
Название компонента	Элементы компонентов
Common HTTP Features	Static Content; Default Document; HTTP Errors; HTTP Redirection.
Application Development	ASP.Net; .Net Extensibility; ISAPI Extensions; ISAPI Filters. WebSocket Protocol.
Microsoft .Net Framework 3.5.1	Windows Communication Foundation HTTP Activation; Windows Communication Foundation Non-HTTP Activation.
Microsoft .Net Framework 4.7 Advanced Services и выше (Windows 8, Windows 10, Windows Server 2012, Windows Server 2016).	ASP.NET 4.6.2 или 4.7; WCF Services: HTTP Activation; Message Queuing (MSMQ) Activation; Named Pipe Activation; TCP Activation; TCP Port Sharing.
Health and Diagnostics	HTTP Logging; Logging Tools; Request Monitor; Custom Logging.
Security	Basic Authentication Request Filtering IP and Domain Restriction

Проверить обязательные компоненты Windows Server 2016

Чтобы проверить наличие необходимых компонентов:

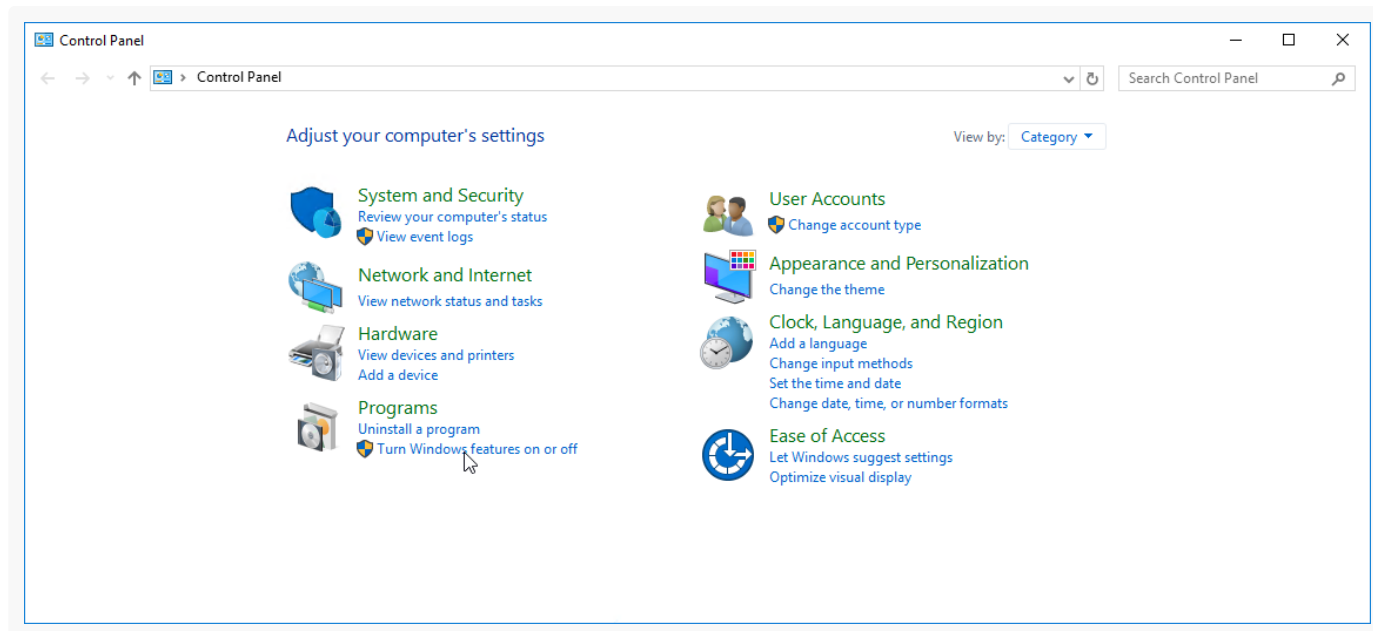
1. В поиске меню [*Start*] введите “control panel” и выберите пункт [*Control Panel*] ([Рис. 1](#)).

Рис. 1 — Раздел [*Control Panel*] меню [*Start*]



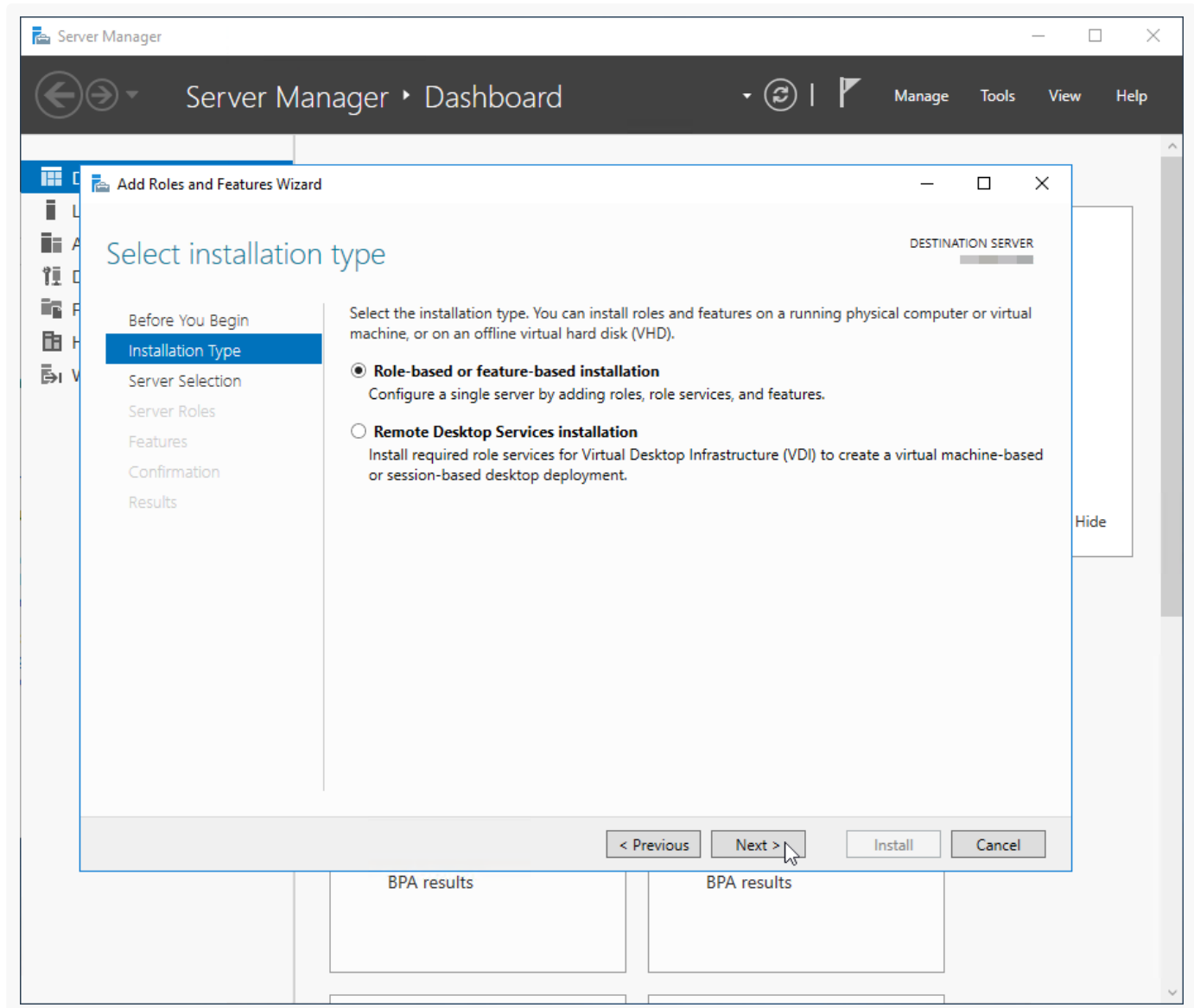
2. В открывшемся окне выберите команду [*Turn Windows features on or off*] ([Рис. 2](#)).

Рис. 2 — Команда [*Turn Windows features on or off*]



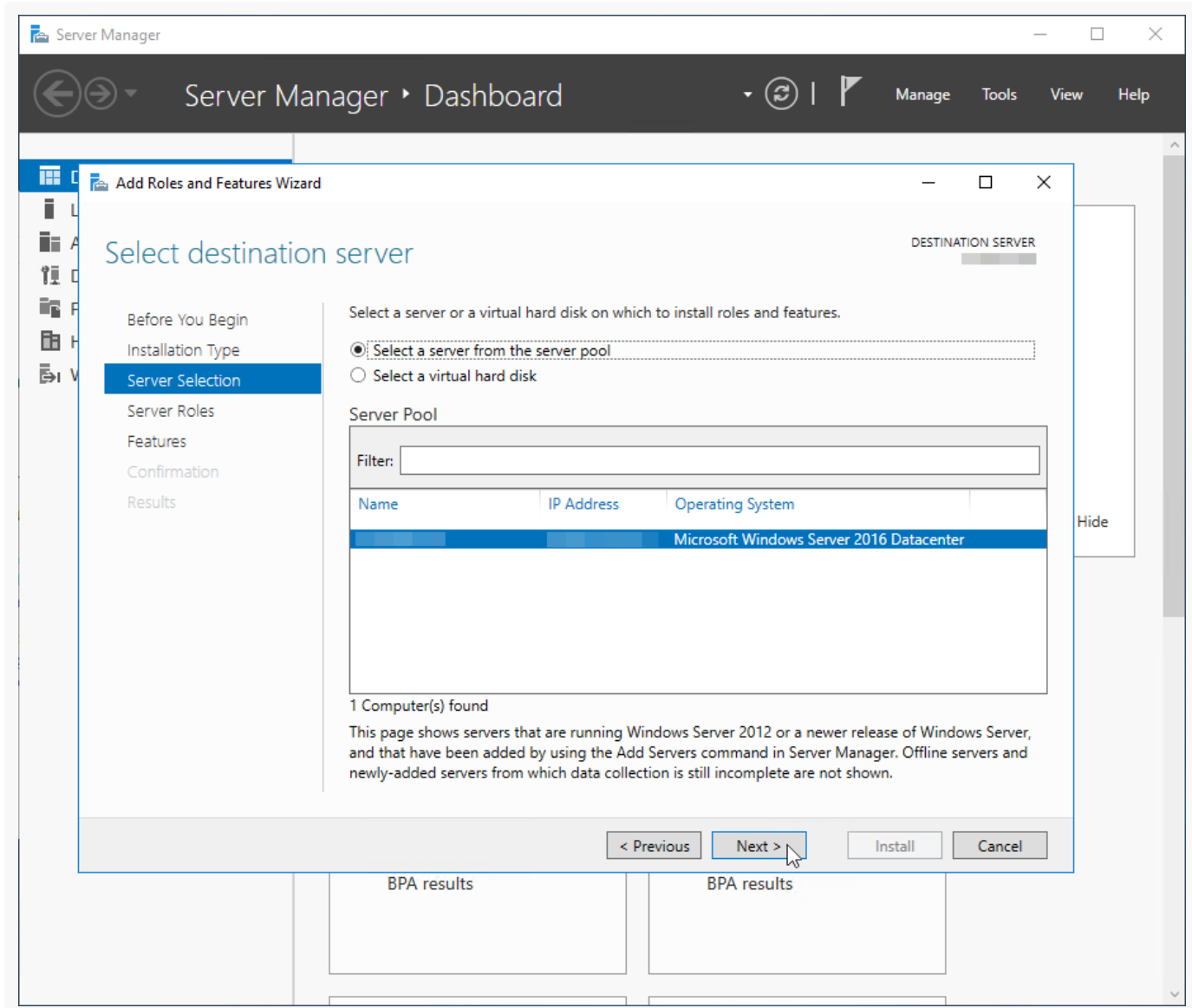
3. В окне [*Add Roles and Features Wizard*] выберите [*Role-based or feature-based installation*] и нажмите [*Next*] ([Рис. 3](#)).

Рис. 3 — Выбор установки на основе ролей



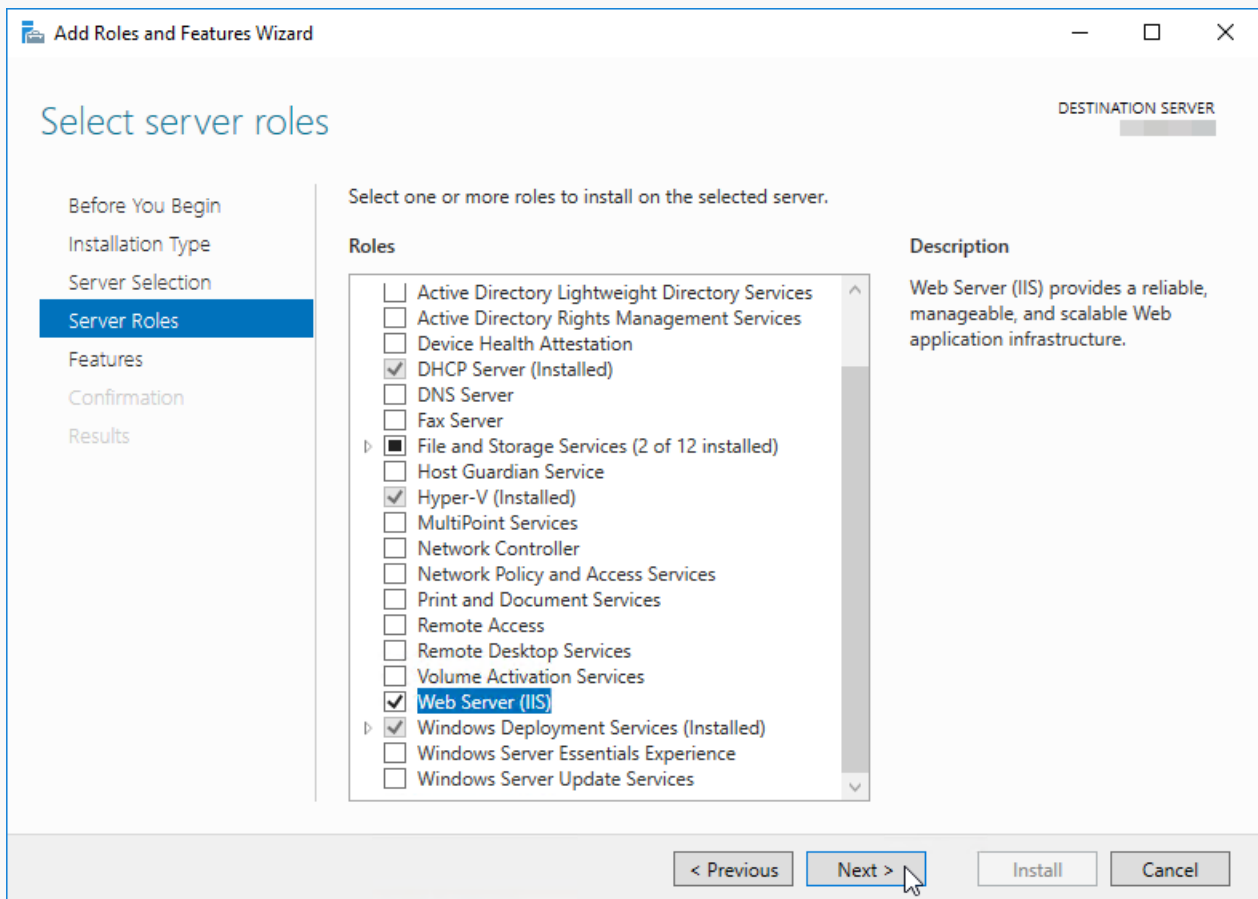
4. В области Server Pool выберите сервер, для которого необходимо установить компоненты, и нажмите [Next] ([Рис. 4](#)).

Рис. 4 — Выбор сервера



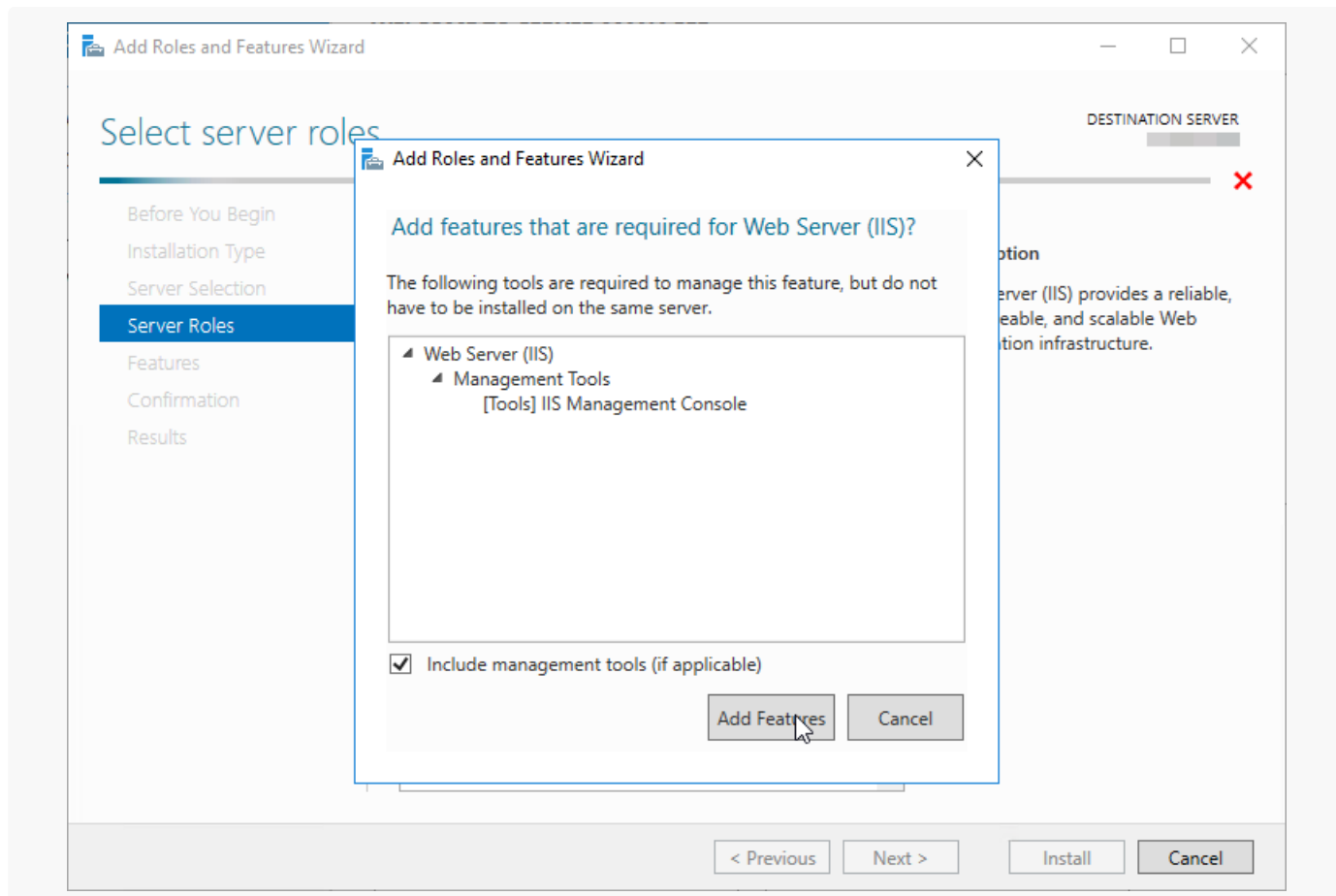
5. В списке ролей установите флажок возле значения Web Server (IIS). Нажмите [Next] ([Рис. 5](#)).

Рис. 5 — Выбор роли Web Server (IIS)



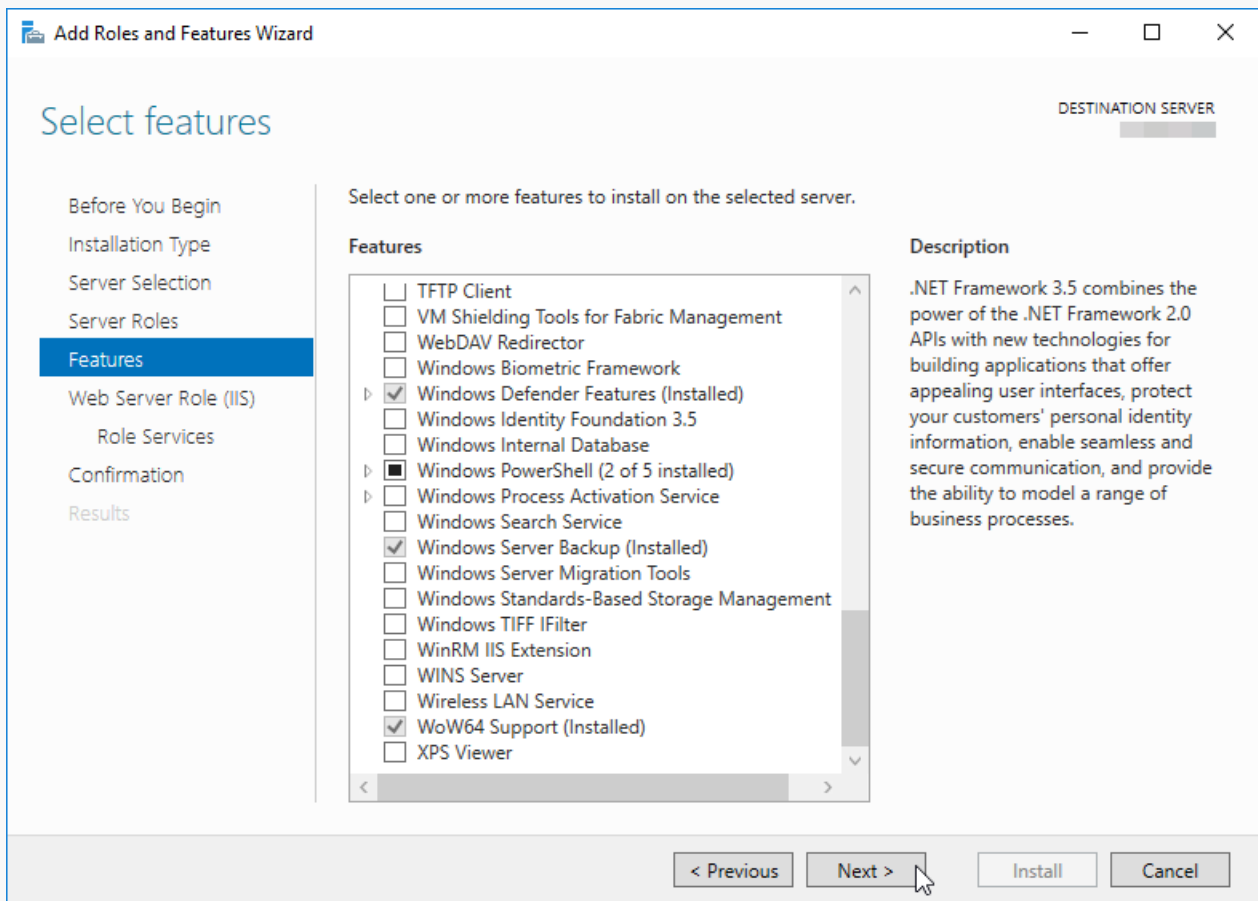
6. В открывшемся окне нажмите [*Add features*] ([Рис. 6](#)).

Рис. 6 — Подтверждение выбора функций



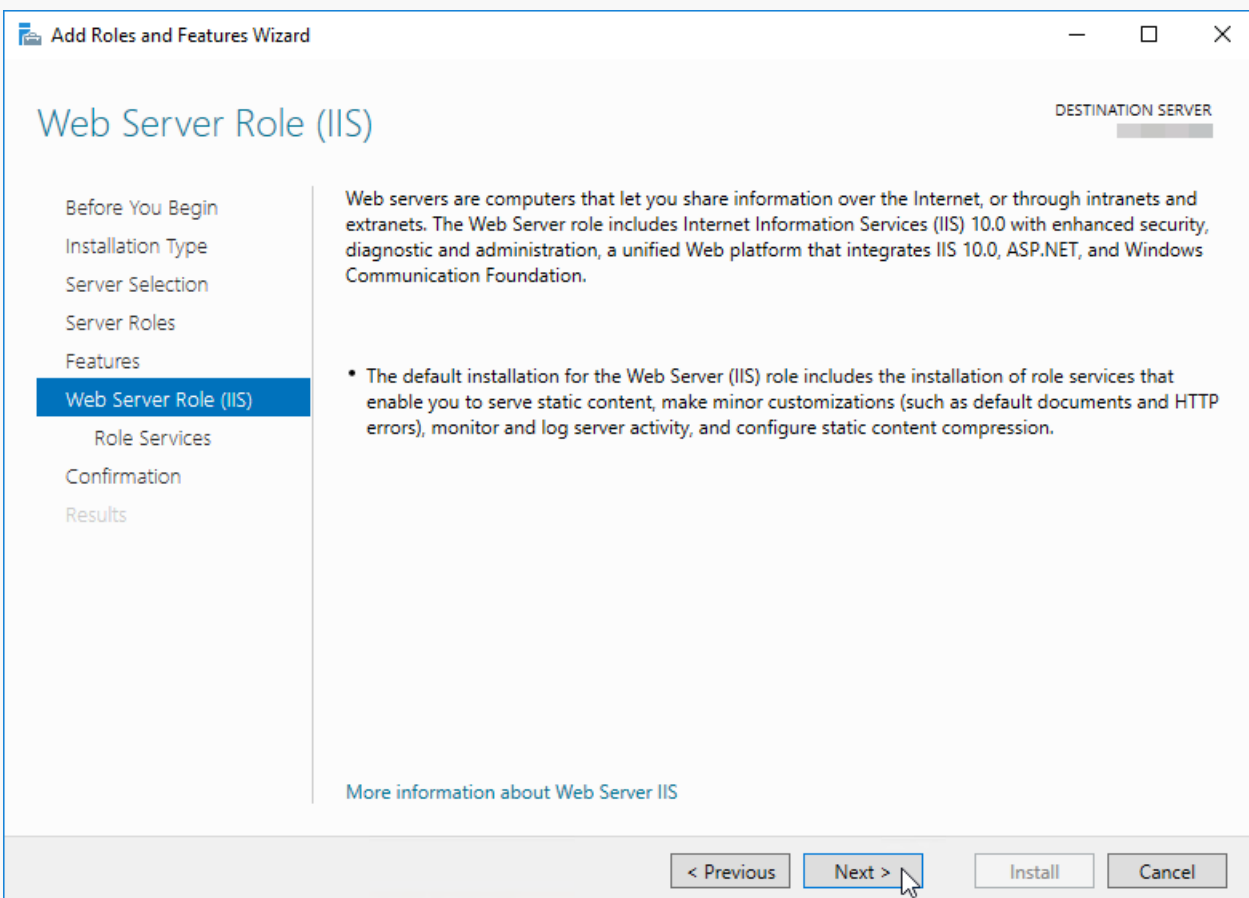
7. Выберите необходимые функции и нажмите [Next] ([Рис. 7](#)).

Рис. 7 — Выбор функций



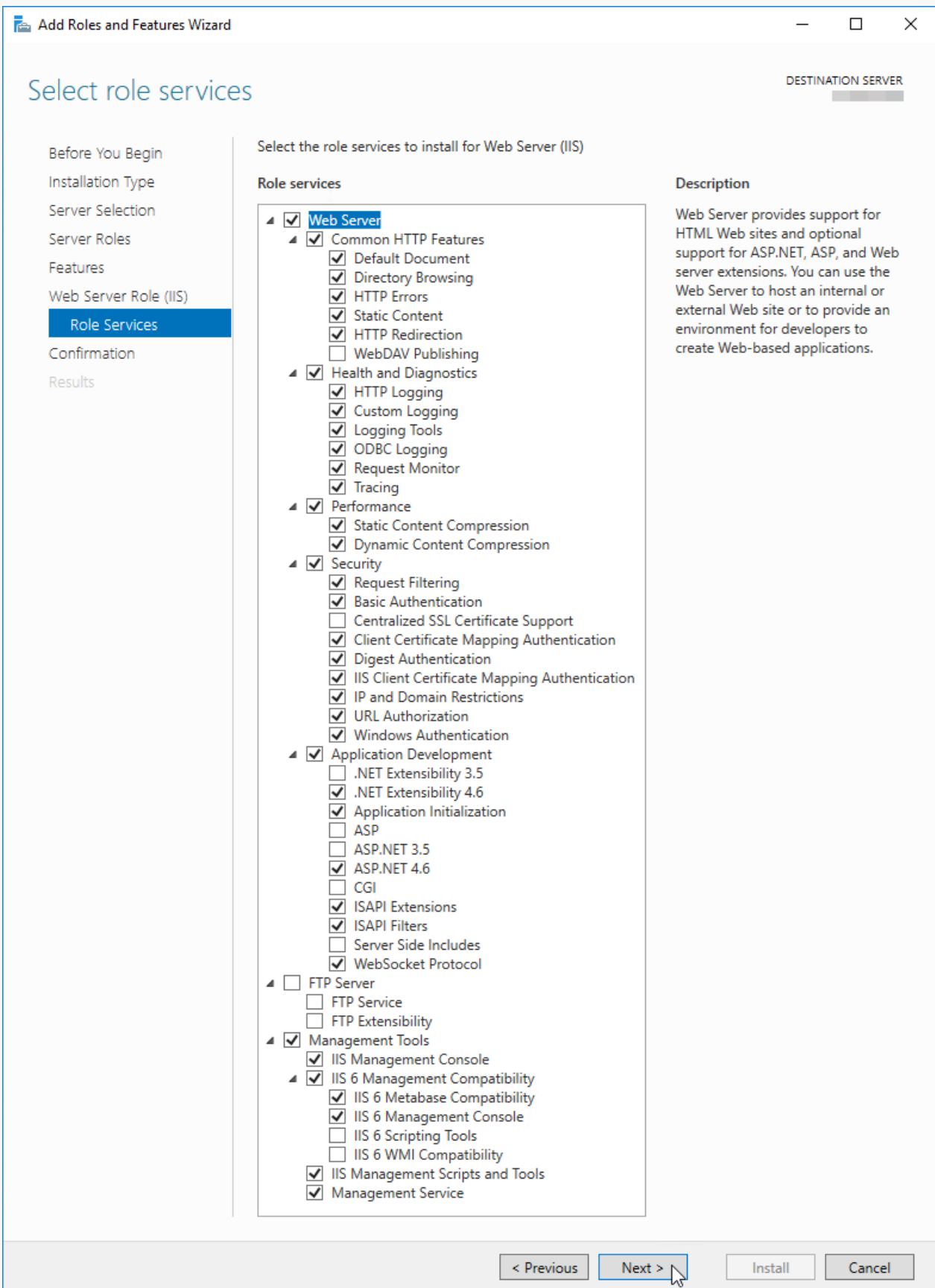
8. Нажмите [*Next*] для перехода к следующему шагу.

Рис. 8 — Подтверждение выбранной роли веб-сервера



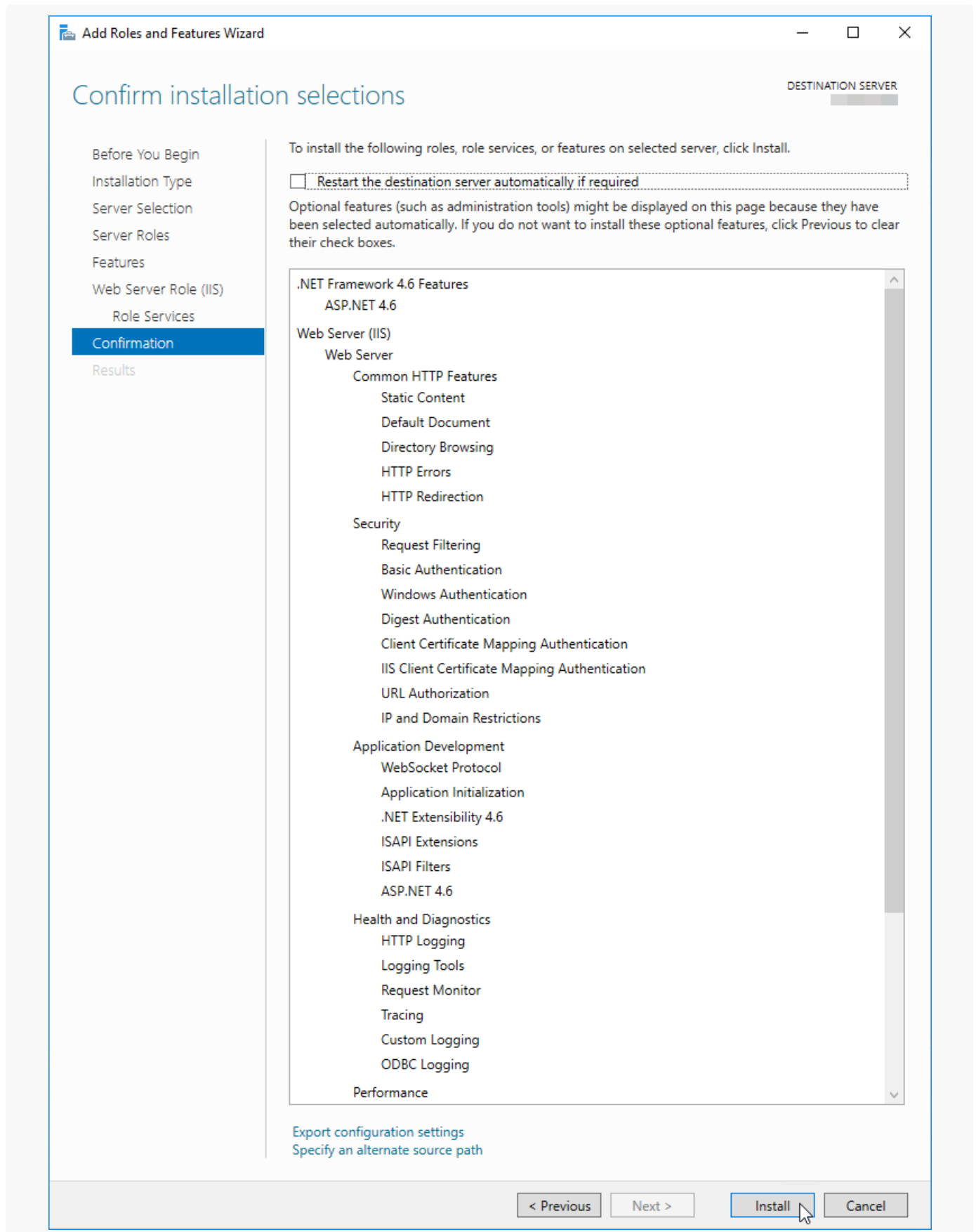
9. Убедитесь, что у вас выбраны все необходимые компоненты ([Рис. 9](#)).

Рис. 9 — Перечень необходимых компонентов Windows Server 2016



10. Нажмите [*Install*] (Рис. 10).

Рис. 10 — Подтверждение установки



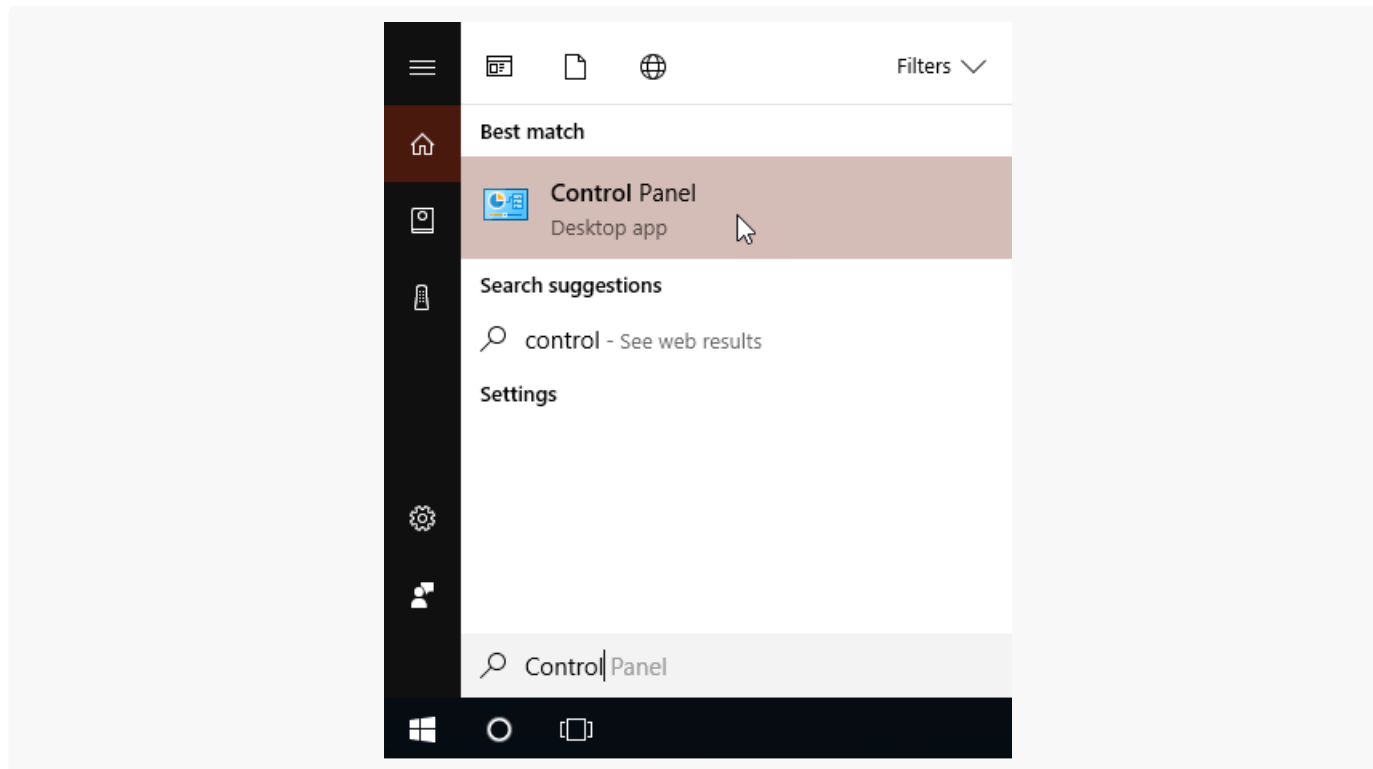
11.Перезагрузите сервер.

Проверить обязательные компоненты Windows 10

Чтобы проверить наличие необходимых компонентов:

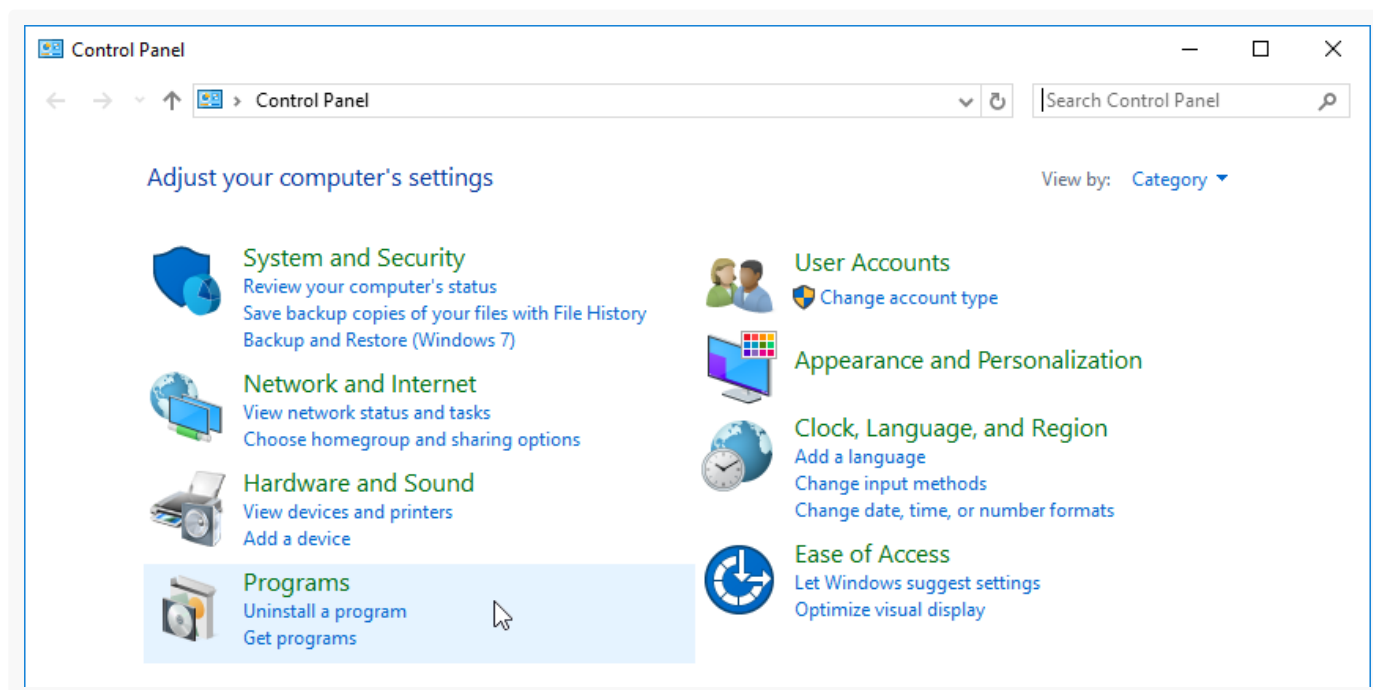
1. В поиске меню [*Start*] введите “control panel” и выберите пункт [*Control Panel*] ([Рис. 1](#)).

Рис. 1 — Раздел [*Control Panel*] меню [*Пуск*]



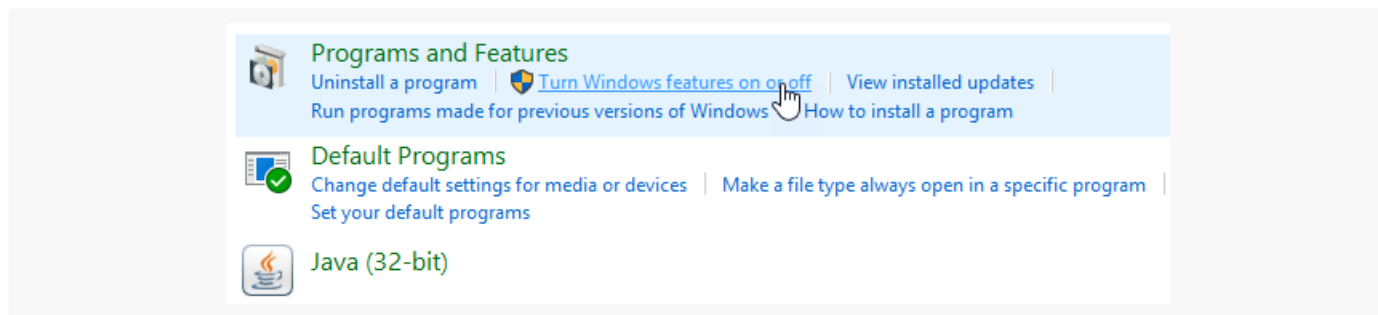
2. В открывшемся окне выберите пункт меню [*Programs*] ([Рис. 2](#)).

Рис. 2 — Меню [*Programs*] панели управления Windows



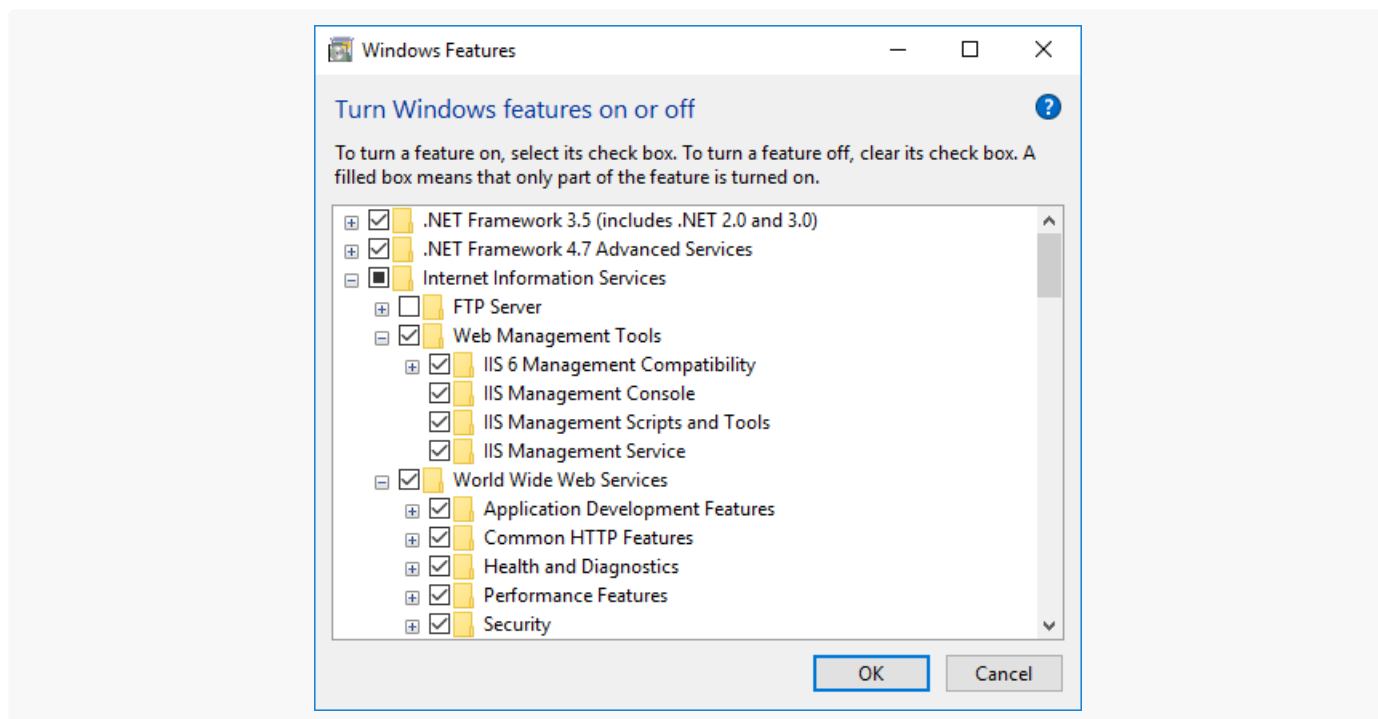
3. В разделе меню [*Programs and Features*] выберите команду [*Turn Windows features on or off*] ([Рис. 2](#)).

Рис. 3 — Команда [*Turn Windows features on or off*]



4. В окне [*Windows Features*] выберите все необходимые компоненты ([Рис. 4](#)).

Рис. 4 — Окно включения компонентов Web Server IIS и Windows



Подготовка к установке Creatio .NET Core на Linux

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Подготовить установочные файлы

Файлы, необходимые для установки Creatio, находятся в архиве. Чтобы распаковать архив, воспользуйтесь командой:

```
unzip -d /path/to/application/directory/ CREATIO_ARCHIVE_NAME.zip
```

/path/to/application/directory/ — путь к папке, в которую необходимо распаковать архив. Рекомендуем распаковывать архив с установочными файлами в пустую папку:

```
unzip creatio_archive_name.zip
```

Creatio_archive_name.zip — название архива с установочными файлами Creatio.

На заметку. Если при выполнении команды `unzip` возникает ошибка "command not found", то вам необходимо установить пакет распаковки архивов:

```
sudo apt-get install unzip
```

Настроить ConnectionStrings.config

В файле `ConnectionStrings.config`, который находится в корневой папке сайта Creatio, содержатся параметры подключения вашего приложения к базе данных и внешним сервисам.

Отредактировать файл ConnectionStrings.config

1. Перейдите в корневой каталог сайта Creatio **~\WebAppRoot\Creatio**.
2. Откройте в текстовом редакторе файл `ConnectionStrings.config`.
3. Укажите параметры подключения **connectionStrings** вашего сайта.

Пример настроенного файла ConnectionStrings.config

```
<?xml version="1.0" encoding="utf-8"?>

<connectionStrings>

<add name="db" connectionString="Server=[ Имя сервера базы данных ];Port=31436;Database=[ Имя бд ];provider=Microsoft.SqlServer.jdbc.4.0.1304.0" />

<add name="redis" connectionString="host=[ Имя компьютера ];db=[ Номер БД Redis ];port=6379" />

<add name="tempDirectoryPath" connectionString="%TEMP%/%USER%/%APPLICATION%" />

<add name="influx" connectionString="url=[ Адрес сервиса сбора метрик сайта ]; user=; password=; " />

<add name="clientPerformanceLoggerServiceUri" connectionString="[ Адрес сервиса логгирования ]" />
```



```
<add name="messageBroker" connectionString="amqp://[ Имя пользователя MessageBroker ]:[ Пароль ]" />
</connectionStrings>
```

Обязательные настройки ConnectionStrings.config

Параметры подключения к базе данных и серверу кэширования необходимы для работы приложения Creatio.

- **db** — отвечает за соединение с базой данных. В нем настраивается путь к базе данных, с которой необходимо установить связь, и способ авторизации на сервере базы данных.

```
<add name="db" connectionString="Server=[ Имя сервера базы данных ];Port=31436;Database=[ Имя
```

- **redis** — отвечает за взаимодействие с сервером Redis:

```
<add name="redis" connectionString="host=[ Имя компьютера ];db=[ Номер БД Redis ];port=6379;
```

Важно. Номер базы данных Redis должен быть уникальным для каждого сайта приложения.

Необязательные настройки ConnectionStrings.config

Параметры подключения к внешним сервисам опциональны, они заполняются только в том случае, когда необходимо настроить интеграцию, например, для сбора метрик сайта.

- **tempDirectoryPath** — путь к временной директории, используемой механизмом установки пакетов:

```
<add name="tempDirectoryPath" connectionString=[ Путь к временной директории, используемой ме
```

- **influx** — отвечает за взаимодействие с сервисом сбора метрик сайта. Данный параметр заполняется только в том случае, когда необходимо собрать статистику использования функциональности для последующей отладки.

```
<add name="influx" connectionString="url=[ Адрес сервиса сбора метрик сайта ]; user=[ Пользов
```

- **clientPerformanceLoggerServiceUri** — отвечает за взаимодействие с сервисом логгирования. Данный параметр заполняется только в том случае, когда необходимо собрать информацию о загрузке страниц приложения.

```
<add name="clientPerformanceLoggerServiceUri" connectionString="[ Адрес сервиса логгирования
```

- **messageBroker** — отвечает за взаимодействие с сервисом RabbitMQ. Данный параметр заполняется только в том случае, когда необходимо настроить горизонтальное масштабирование нагрузки с помощью RabbitMQ.

```
<add name="messageBroker" connectionString="amqp://[ Имя пользователя MessageBroker ]:[ Парол
```

Настроить интеграцию с Google

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Для того чтобы настроить интеграцию с сервисами Google, пользователям приложений Creatio on-site необходимо:

1. Зарегистрировать и настроить учетную запись Google.
2. Открыть доступ к Calendar API.
3. Сгенерировать ключи для интеграции ("Client ID" и "Client Secret").
4. Ввести полученные ключи в Creatio в качестве значений системных настроек.

Подробнее читайте в статье [Зарегистрировать приложение Creatio в Workspace](#).

Общий порядок установки Creatio

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Данное руководство содержит полный перечень этапов по развертыванию и настройке Creatio on-site на операционных системах Windows или Linux, а также ссылки на подробное описание каждого из этих этапов.

Установить Creatio NET Framework на Windows

В общем случае для развертывания приложения Creatio on-site вам необходимо:

1. Развернуть сервер кэширования данных Creatio (Redis). [Подробнее >>>](#)
2. Развернуть базу данных. Обратите внимание, что этот шаг отличается для различных СУБД. [Подробнее >>>](#)
3. Включить и установить обязательные компоненты Windows. [Подробнее >>>](#)
4. Установить последние обновления Windows.
5. Создать и настроить сайт приложения в IIS. [Подробнее >>>](#)
6. Внести изменения в файл конфигурации ConnectionStrings.config. Обратите внимание, что этот шаг отличается для различных СУБД. [Подробнее >>>](#)

7. Внести изменения в файл конфигурации Web.config. [Подробнее >>>](#)

После установки приложения необходимо выполнить **дополнительные настройки** для обеспечения корректной работы всех компонентов:

- Настроить вебсокеты. [Подробнее >>>](#)
- Перевести Creatio с HTTP на HTTPS. [Подробнее >>>](#)
- Развернуть и настроить глобальный поиск в Creatio. [Подробнее >>>](#)
- Настроить сервис машинного обучения. [Подробнее >>>](#)
- Настроить интеграции и доступ к интернету для дополнительной функциональности, например, для сервиса обогащения данных, интеграции с социальными сетями или синхронизации с Google. [Подробнее >>>](#)
- Настроить email-рассылки (только для продуктов, которые содержат Marketing Creatio). [Подробнее >>>](#)

Установить Creatio .NET Core на Linux

В общем случае для развертывания приложения Creatio on-site вам необходимо:

1. Подготовить установочные файлы Creatio. [Подробнее >>>](#)
2. Развернуть сервер базы данных. [Подробнее >>>](#)
3. Развернуть сервер кэширования данных Creatio (Redis). [Подробнее >>>](#)
4. Внести изменения в файл конфигурации ConnectionStrings.config. [Подробнее >>>](#)
5. Развернуть сервер приложения. [Подробнее >>>](#)

Если вы планируете запуск приложения непосредственно **с локального компьютера**:

1. Установите .NET Core, GDI+ совместимый API для семейства UNIX-подобных операционных систем, библиотеки разработки и файлы заголовков для GNU C. [Подробнее >>>](#)
2. Запустите сервер приложения Creatio. [Подробнее >>>](#)

Если вы планируете запуск приложения **в контейнере Docker**:

1. Настройте доступ к Redis из контейнера Docker. [Подробнее >>>](#)
2. Установите Docker. [Подробнее >>>](#)
3. Создайте Dockerfile. [Подробнее >>>](#)
4. Создайте и запустите образ Docker. [Подробнее >>>](#)

На заметку. Запуск сервера PostgreSQL в Docker описан в [документации Docker](#).

Настроить массовый поиск дублей

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Сервис массового поиска дублей используется для дедупликации в разделах Creatio.

Важно. Для работы сервиса массового поиска дублей необходимо предварительно настроить сервис глобального поиска в Elasticsearch. Информацию о настройке читайте в статье [Настроить глобальный поиск](#).

Для настройки сервиса массового поиска дублей необходимы базовые знания администрирования docker-compose и ОС Linux.

Сервис глобального поиска дублей имеет собственную версию. Данная инструкция универсальна для всех версий сервиса. Версии 1.0–1.5 совместимы с любой версией Creatio, в которой есть функциональность глобального поиска дублей. Версия сервиса 2.0 совместима с Creatio версий 7.15.4 — 7.17.0. Версия 3.0 совместима с Creatio версии 7.17.0 и выше и глобальным поиском версии 3.0.

Для расчета требований к серверам воспользуйтесь [калькулятором системных требований](#).

Компоненты сервиса массового поиска дублей

Для настройки сервиса необходимы:

1. Компоненты глобального поиска. Список приведен в статье [Настроить глобальный поиск](#).
2. Компоненты сервиса массового поиска дублей. Перечень компонентов приведен ниже.

[Mongodb](#) — документоориентированная система управления базами данных.

[dd-web-api](#) — web-сервис для общения в Creatio.

[dd-data-service](#) — внутренний сервис коммуникаций с mongodb.

[dd-duplicates-search-worker](#) — компонент поиска дублей.

[dd-duplicates-deletion-worker](#) — компонент, выполняющий точечное удаление дублей.

[dd-duplicates-confirmation-worker](#) — компонент, выполняющий группировку и фильтрацию найденных дублей с учетом уникальности.

[dd-duplicates-cleaner](#) — компонент очистки дублей.

[dd-deduplication-task-worker](#) — компонент постановки задачи дедупликации.

[dd-deduplication-preparation-worker](#) — компонент подготовки процесса дедупликации, формирует запросы для поиска дублей согласно правилам.

[dd-deduplication-task-diagnostic-worker](#) — компонент, контролирующий выполнение задачи поиска дублей.

Для настройки компонентов скачайте исходные файлы. [Скачать файлы](#).

1. Развернуть и настроить глобальный поиск в Creatio.
2. Скачать, разархивировать и скопировать необходимые для настройки исходные файлы на машину с установленным ПО docker, docker-compose. [Скачать файлы](#).
3. Настроить переменные окружения.
4. Запустить контейнеры.
5. Проверить успешность запуска контейнеров.
6. Проверить логирование.

7. Подключить функциональность массового поиска дублей в Creatio.

Настроить переменные окружения

Переменные окружения содержатся в файле `compose/.env`. Установите значения переменных, отредактировав этот файл.

Название переменной	Описание	Значение по умолчанию
ELASTICSEARCH_URI	Ip-адрес сервера, на котором был развернут ElasticSearch на шаге настройки глобального поиска в Creatio.	<code>http://user:password@external.elasticsearch-ip:9200/</code>

Запустить контейнеры

Для запуска контейнеров необходимо выполнить команду:

```
cd compose # перейти в папку compose
docker-compose up -d
```

Проверить успешность запуска контейнеров

Для просмотра списка всех запущенных контейнеров необходимо в консоли выполнить команду:

```
docker ps --filter "label=service=dd" -a --format "table {{.Names}}\t{{.Ports}}\t{{.Status}}\t{{
```

У всех запущенных контейнеров отобразится статус “Up”.

Проверить логирование

По умолчанию логирование происходит во время выполнения команды “`stdout`” контейнеров. Чтобы просмотреть последние 100 записей логов из контейнера `dd-data-service`, выполните команду:

```
docker logs --tail 100 dd-data-service
```

Подключить функциональность массового поиска дублей в Creatio

На стороне Creatio необходимо выполнить следующие шаги.

1. Установите значение системной настройки “Адрес сервиса дедупликации”.
2. Настройте права на системную операцию “Поиск дублей”.
3. Включите в Creatio функциональность массового поиска дублей. Обратите внимание, что данная настройка отличается для разных СУБД.
4. Перезапустите приложение Creatio.

Установить значение системной настройки “Адрес сервиса дедупликации”

В разделе [*Системные настройки*] найдите настройку “Адрес сервиса дедупликации” (DeduplicationWebApiUrl) и в ней укажите URL к dd-web-api, строка следующего типа:
http://external.deduplication-web-api:8086.

Настроить права на системную операцию “Поиск дублей”

В разделе [*Доступ к операциям*] откройте системную операцию “Поиск дублей” (CanSearchDuplicates) и на детали [*Доступ к операции*] раздайте права необходимым пользователям/ролям, которые смогут выполнять поиск дублей.

Включить в Creatio функциональность массового поиска дублей

Для включения функциональности (Feature Toggle) массового поиска дублей (Deduplication, ESDeduplication, BulkESDeduplication) необходимо выполнить SQL-скрипт. Этот скрипт различается в зависимости от используемой СУБД — MS SQL, Oracle или Postgre SQL.

Для СУБД MS SQL

```
DECLARE @DeduplicationFeature NVARCHAR(50) = 'Deduplication';
DECLARE @DeduplicationFeatureId UNIQUEIDENTIFIER = (SELECT TOP 1 Id FROM Feature WHERE Code = @DeduplicationFeature);

DECLARE @ESDeduplicationFeature NVARCHAR(50) = 'ESDeduplication';
DECLARE @ESDeduplicationFeatureId UNIQUEIDENTIFIER = (SELECT TOP 1 Id FROM Feature WHERE Code = @ESDeduplicationFeature);

DECLARE @Bulk_ES_DD_Feature NVARCHAR(50) = 'BulkESDeduplication';
DECLARE @Bulk_ES_DD_FeatureId UNIQUEIDENTIFIER = (SELECT TOP 1 Id FROM Feature WHERE Code = @Bulk_ES_DD_Feature);

DECLARE @allEmployeesId UNIQUEIDENTIFIER = 'A29A3BA5-4B0D-DE11-9A51-005056C00008';
IF (@DeduplicationFeatureId IS NOT NULL)
BEGIN
    IF EXISTS (SELECT * FROM AdminUnitFeatureState WHERE FeatureId = @DeduplicationFeatureId)
        UPDATE AdminUnitFeatureState SET FeatureState = 1 WHERE FeatureId = @DeduplicationFeatureId
    ELSE
        INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@allEmployeesId, 1, @DeduplicationFeatureId);
END
```

```

@DeduplicationFeatureId)
END;
ELSE
BEGIN
    SET @DeduplicationFeatureId = NEWID()
    INSERT INTO Feature (Id, Name, Code) VALUES
(@DeduplicationFeatureId, @DeduplicationFeature, @DeduplicationFeature)
    INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@allE
END;

IF (@ESDeduplicationFeatureId IS NOT NULL)
BEGIN
    IF EXISTS (SELECT * FROM AdminUnitFeatureState WHERE FeatureId = @ESDeduplicationFeatureId)
    UPDATE AdminUnitFeatureState SET FeatureState = 1 WHERE FeatureId = @ESDeduplicationFeature
    ELSE
    INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@allE
END;
ELSE
BEGIN
    SET @ESDeduplicationFeatureId = NEWID()
    INSERT INTO Feature (Id, Name, Code) VALUES (@ESDeduplicationFeatureId, @ESDeduplicationFe
    INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@allEn
END;

IF (@Bulk_ES_DD_FeatureId IS NOT NULL)
BEGIN
    IF EXISTS (SELECT * FROM AdminUnitFeatureState WHERE FeatureId = @Bulk_ES_DD_FeatureId)
    UPDATE AdminUnitFeatureState SET FeatureState = 1 WHERE FeatureId = @Bulk_ES_DD_FeatureId
    ELSE
    INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@allEn
END;
ELSE
BEGIN
    SET @Bulk_ES_DD_FeatureId = NEWID()
    INSERT INTO Feature (Id, Name, Code) VALUES (@Bulk_ES_DD_FeatureId, @Bulk_ES_DD_Feature, @
    INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@allEn
END;

```

Для СУБД Oracle

```

CREATE OR REPLACE FUNCTION
generate_uuid return varchar2 is
    v_uuid varchar2(38);
    v_guid varchar2(32);
BEGIN
    v_guid := sys_guid();
    v_uuid := lower(

```

```

'{' ||
    substr(v_guid, 1,8) || '-' ||
    substr(v_guid, 9,4) || '-' ||
    substr(v_guid, 13,4) || '-' ||
    substr(v_guid, 17,4) || '-' ||
    substr(v_guid, 21) ||
    '}'
);
RETURN v_uuid;
END;
/
DECLARE
    DeduplicationFeature VARCHAR(50) := 'Deduplication';
    DeduplicationFeatureId VARCHAR(38) := NULL;
    DeduplicationFeatureId_GUID VARCHAR(38) := generate_uuid();
    ESDeduplicationFeature VARCHAR(50) := 'ESDeduplication';
    ESDeduplicationFeatureId VARCHAR(38) := NULL;
    ESDeduplicationFeatureId_GUID VARCHAR(38) := generate_uuid();
    BulkESDeduplicationFeature VARCHAR(50) := 'BulkESDeduplication';
    BulkESDeduplicationFeatureId VARCHAR(38) := NULL;
    Bulk_ES_DD_GUID VARCHAR(38) := generate_uuid();
    allEmployeesId VARCHAR(38) := '{7F3B869F-34F3-4F20-AB4D-7480A5FDF647}';
    State_Deduplication VARCHAR(1) := NULL;
    State_ESDeduplication VARCHAR(1) := NULL;
    State_BulkESDeduplication VARCHAR(1) := NULL;
BEGIN
    SELECT MAX("Id") INTO DeduplicationFeatureId FROM "Feature" WHERE "Code" = DeduplicationFeature;
    SELECT MAX("Id") INTO ESDeduplicationFeatureId FROM "Feature" WHERE "Code" = ESDeduplicationFeature;
    SELECT MAX("Id") INTO BulkESDeduplicationFeatureId FROM "Feature" WHERE "Code" = BulkESDeduplicationFeature;
    SELECT MAX("FeatureState") INTO State_Deduplication FROM "AdminUnitFeatureState" WHERE "FeatureId" = DeduplicationFeatureId;
    SELECT MAX("FeatureState") INTO State_ESDeduplication FROM "AdminUnitFeatureState" WHERE "FeatureId" = ESDeduplicationFeatureId;
    SELECT MAX("FeatureState") INTO State_BulkESDeduplication FROM "AdminUnitFeatureState" WHERE "FeatureId" = BulkESDeduplicationFeatureId;
    IF (DeduplicationFeatureId IS NULL) THEN
        INSERT INTO "Feature" ("Id", "Name", "Code") VALUES (DeduplicationFeatureId_GUID, DeduplicationFeature, DeduplicationFeatureId);
        INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (allEmployeesId, State_Deduplication, DeduplicationFeatureId);
    ELSE
        IF (State_Deduplication IS NOT NULL) THEN
            UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = DeduplicationFeatureId;
        ELSE
            INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (allEmployeesId, State_Deduplication, DeduplicationFeatureId);
        END IF;
    END IF;
    IF (ESDeduplicationFeatureId IS NULL) THEN
        INSERT INTO "Feature" ("Id", "Name", "Code") VALUES (ESDeduplicationFeatureId_GUID, ESDeduplicationFeature, ESDeduplicationFeatureId);
        INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (allEmployeesId, State_ESDeduplication, ESDeduplicationFeatureId);
    ELSE
        IF (State_ESDeduplication IS NOT NULL) THEN
            UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = ESDeduplicationFeatureId;
        ELSE
            INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (allEmployeesId, State_ESDeduplication, ESDeduplicationFeatureId);
        END IF;
    END IF;
    IF (BulkESDeduplicationFeatureId IS NULL) THEN
        INSERT INTO "Feature" ("Id", "Name", "Code") VALUES (BulkESDeduplicationFeatureId_GUID, BulkESDeduplicationFeature, BulkESDeduplicationFeatureId);
        INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (allEmployeesId, State_BulkESDeduplication, BulkESDeduplicationFeatureId);
    ELSE
        IF (State_BulkESDeduplication IS NOT NULL) THEN
            UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = BulkESDeduplicationFeatureId;
        ELSE
            INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (allEmployeesId, State_BulkESDeduplication, BulkESDeduplicationFeatureId);
        END IF;
    END IF;
END;

```



```

        ELSE
            INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (
            END IF;
        END IF;
    IF (BulkESDeduplicationFeatureId IS NULL) THEN
        INSERT INTO "Feature" ("Id", "Name", "Code") VALUES (Bulk_ES_DD_GUID, BulkESDeduplicationF
        INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (
        ELSE
            IF (State_BulkESDeduplication IS NOT NULL) THEN
                UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = BulkESDeduplica
            ELSE
                INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (
            END IF;
        END IF;
    END;

```

Для СУБД PostgreSQL

```

DO $$

DECLARE
    DeduplicationFeature VARCHAR(50) := 'Deduplication';
    DeduplicationFeatureId uuid;

    ESDeduplicationFeature VARCHAR(50) := 'ESDeduplication';
    ESDeduplicationFeatureId uuid;

    Bulk_ES_DD_Feature VARCHAR(50) := 'BulkESDeduplication';
    Bulk_ES_DD_FeatureId uuid;

    allEmployeesId uuid := 'A29A3BA5-4B0D-DE11-9A51-005056C00008';

BEGIN

    SELECT "Id" INTO DeduplicationFeatureId FROM "Feature"
    WHERE "Code" = DeduplicationFeature
    LIMIT 1;
    IF (DeduplicationFeatureId IS NOT NULL)
    THEN
        IF EXISTS (SELECT * FROM "AdminUnitFeatureState" WHERE "FeatureId" = DeduplicationFeatureId)
            UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = DeduplicationFeatu
        ELSE
            INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (
        END IF;
    ELSE
        DeduplicationFeatureId := uuid_generate_v4();

```

```

INSERT INTO "Feature" ("Id", "Name", "Code") VALUES (DeduplicationFeatureId, DeduplicationFeat
INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (al
END IF;

SELECT "Id" INTO ESDeduplicationFeatureId FROM "Feature"
WHERE "Code" = ESDeduplicationFeature
LIMIT 1;
IF (ESDeduplicationFeatureId IS NOT NULL)
THEN
  IF EXISTS (SELECT * FROM "AdminUnitFeatureState" WHERE "FeatureId" = ESDeduplicationFeatureId
    UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = ESDeduplicationFeat
  ELSE
    INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (
  END IF;
ELSE
  ESDeduplicationFeatureId := uuid_generate_v4();
  INSERT INTO "Feature" ("Id", "Name", "Code") VALUES (ESDeduplicationFeatureId, ESDeduplication
  INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (al
END IF;

SELECT "Id" INTO Bulk_ES_DD_FeatureId FROM "Feature"
WHERE "Code" = Bulk_ES_DD_Feature
LIMIT 1;
IF (Bulk_ES_DD_FeatureId IS NOT NULL)
THEN
  IF EXISTS (SELECT * FROM "AdminUnitFeatureState" WHERE "FeatureId" = Bulk_ES_DD_FeatureId) TH
    UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = Bulk_ES_DD_Feature
  ELSE
    INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (
  END IF;
ELSE
  Bulk_ES_DD_FeatureId := uuid_generate_v4();
  INSERT INTO "Feature" ("Id", "Name", "Code") VALUES (Bulk_ES_DD_FeatureId, Bulk_ES_DD_Feature,
  INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (al
END IF;
END $$;

```

Перезапустить приложение Creatio

Очистите redis, перезапустите приложение Creatio и выполните вход в приложение.

Обновить версию массового поиска дублей

Для обновления сервиса массового поиска дублей с версии 2.0 на версию 3.0 выполните следующие шаги.

1. Сделайте резервную копию данных о дублях в приложении. Для этого через swagger web-api `http://[ip-адрес сервера]:8086/api` выполните команду `/api/snapshot/backup/gzip/{indexName}`, где {indexName} — название индекса (последние 64 символа) из системной настройки “Адрес сервиса поиска” (код “GlobalSearchUrl”).
2. Удалите все docker volume версии 2.0. Для этого перейдите в папку docker-compose файлов версии 2.0 и выполните команду [docker-compose down -v](#).
3. Установите сервис массового поиска дублей версии 3.0.
4. Загрузите в обновленный сервис данные о дублях, полученные на шаге 1. Для этого на новой версии сервиса выполните через swagger команду `/api/snapshot/restore/gzip`.

Рекомендуемые работы по обслуживанию сервиса

Для поддержания работоспособности сервиса и возможности быстрого восстановления данных, например, в случае перебоев с электроэнергией, рекомендуется раз в сутки выполнять резервное копирование mongodb.

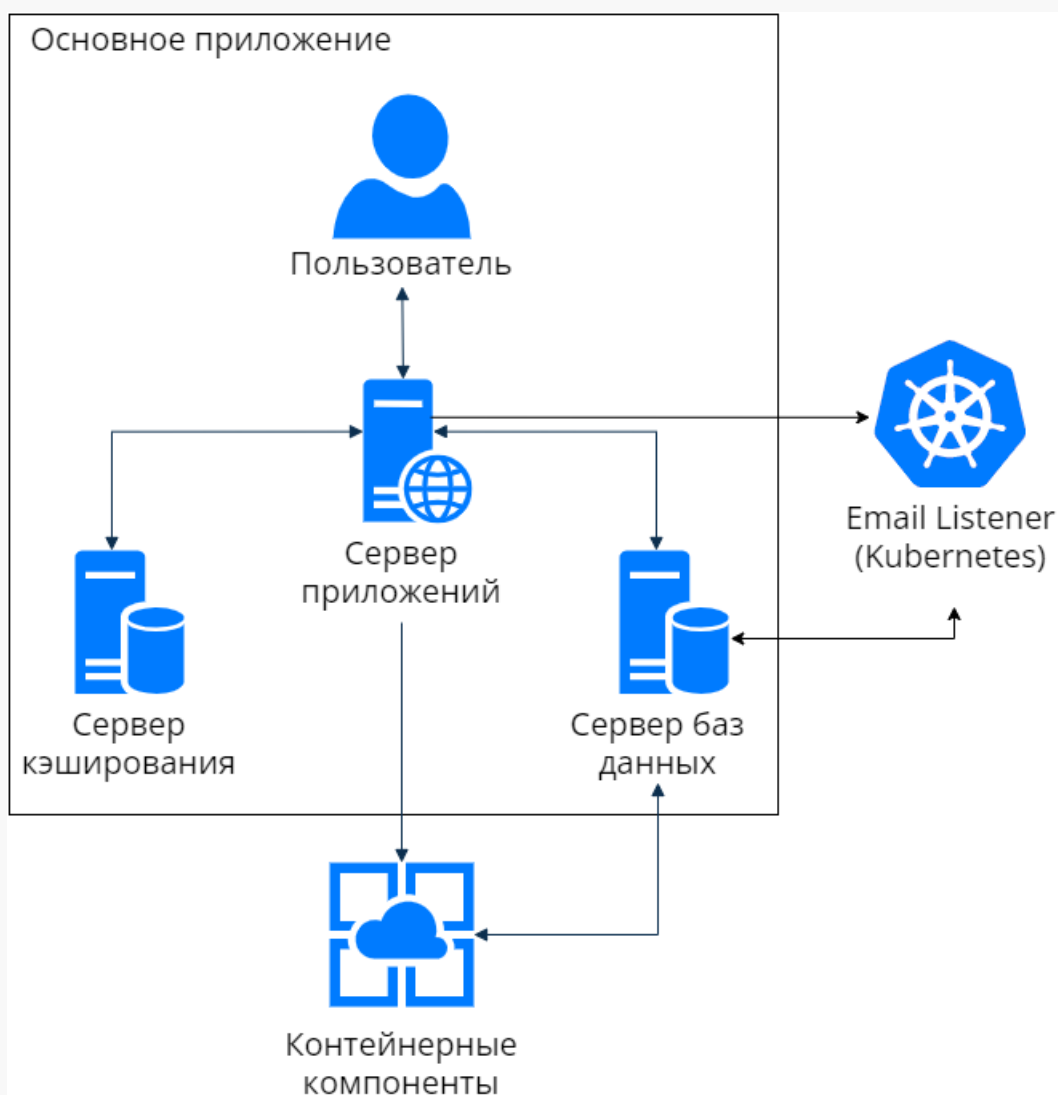
Системные требования к серверам

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Для работы Creatio необходимо развернуть следующие **обязательные компоненты** (Рис. 1):

- **Веб-сервер**, который также является **сервером приложений** Creatio.
- **Сервер баз данных** с выбранной системой управления базами данных (СУБД).
- **Сервер хранения сессий** (Redis), который также называют “**сервером кэширования**”.

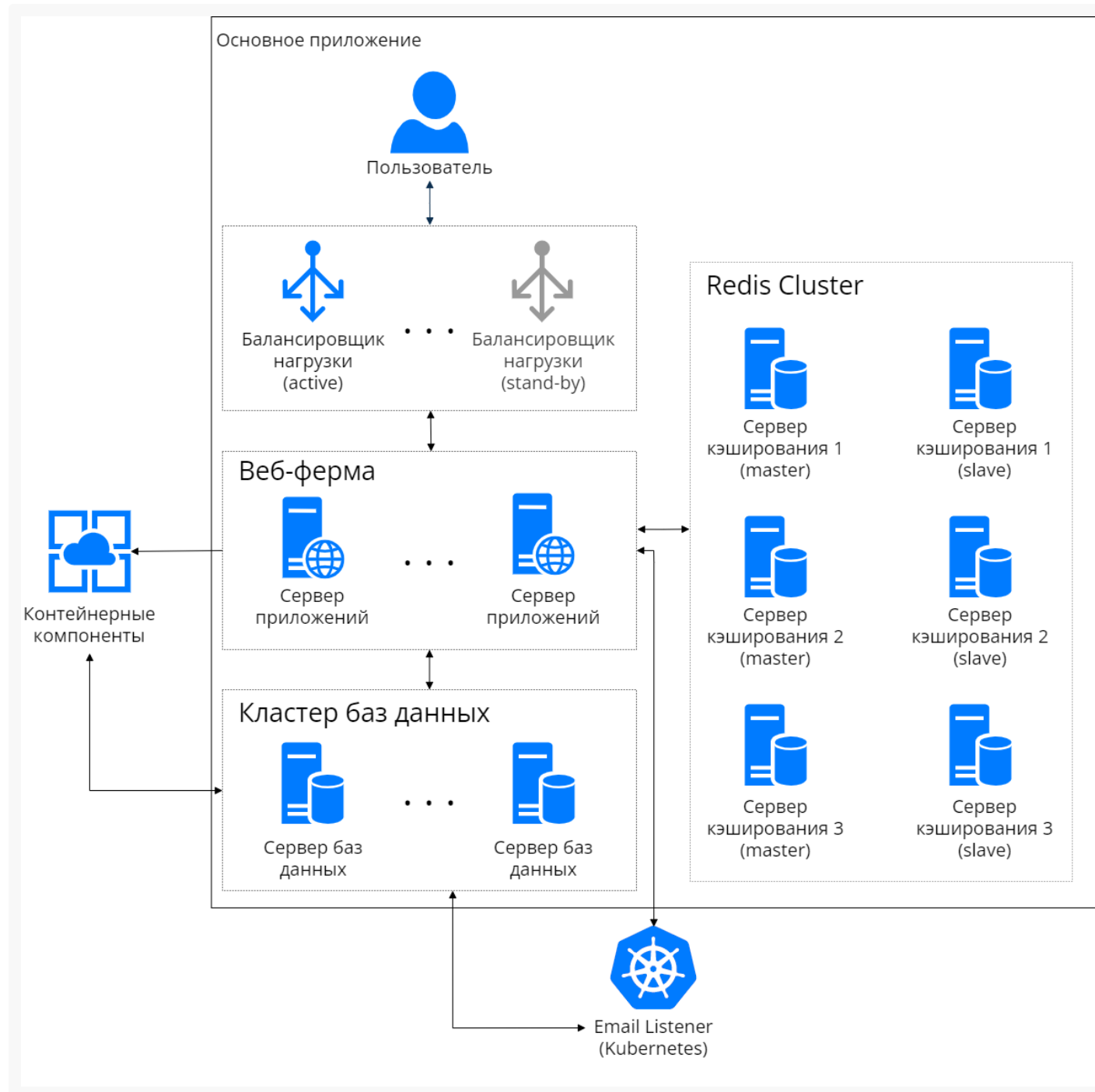
Рис. 1 — Базовая инфраструктура Creatio



К **опциональным компонентам** относятся:

- **Система контроля версий (SVN)** — рекомендуется для сред разработки. SVN позволяет разработчикам управлять версиями и изменениями конфигурации Creatio (например, вернуться к предыдущей версии, объединить версии и т. д.).
- **Балансировщики нагрузки** серверов приложений, баз данных и кэширования для работы в отказоустойчивом режиме (Рис. 2).

Рис. 2 — Отказоустойчивая инфраструктура Creatio



Определенная функциональность, например, глобальный поиск, дедупликация и машинное обучение, требует развертывания **контейнеризации** (виртуализации на уровне операционной системы). Внедрение контейнерной виртуализации целесообразно для обеспечения функционирования следующих сервисов Creatio:

- машинное обучение;
- обогащение данных;
- облачный сервис рассылок;
- глобальный поиск;
- поиск и объединение дублей;

- Exchange Listener.

Общие рекомендации

Все компоненты системы поддерживают функционирование в виртуальном окружении. Список поддерживаемых облачных/виртуальных платформ:

- Amazon Web Services;
- Microsoft Azure / Hyper-V;
- VMware vCloud Director / vSphere.

На заметку. Совместимость системы с другими виртуальными / облачными платформами не гарантируется.

Активные экземпляры компонентов системы должны быть размещены в одной локации (датацентр / серверное помещение / офисное здание и т.д.). Допустимо размещение резервных компонентов в удаленных локациях при реализации защиты от катастрофических событий.

Все компоненты должны работать в предназначенных для них операционных системах. Совмещение компонентов допустимо, но не рекомендуется.

Для организации кластера СУБД может использоваться как общее файловое хранилище, так и отдельное дисковое пространство узлов кластера.

Сервер системы контроля версий (SVN) является необязательным компонентом. Он нужен, только если планируется ведение версионности изменений конфигурации.

Балансировщик нагрузки является необязательным компонентом, который может понадобиться, если:

- предполагается повышенная сетевая нагрузка на серверы приложений;
- серверы приложений развернуты в отказоустойчивой конфигурации (веб-фермы).

Балансировщик может быть аппаратным или программным. Для работы в отказоустойчивом режиме используется балансировщик HTTP/HTTPS-трафика с поддержкой протокола WebSocket. Работа приложения тестировалась на программных балансировщиках нагрузки HAProxy и MS ARR (Microsoft Advanced Request Routing). Известны случаи успешного использования других балансировщиков, например, Citrix, Cisco, NginX, FortiGate.

Каналы связи

IIS — SQL:

- пропускная способность — 10Мбит/с на 100 активных пользователей;
- задержки не более 15-20 мс.

IIS — пользователь:

- минимально 256 кбит/с на 1 активного пользователя;
- рекомендуется 512 кбит/с на 1 активного пользователя;

- формула: 30кБайт/с * (общее число одновременно работающих пользователей) * 10%.

Ввод/вывод:

- 100 IOPS на 1000 активных пользователей
- SQL: 1000 IOPS на 1000 активных пользователей

Требования для отправки массовых email-рассылок (продукт Marketing Creatio):

Рекомендуемая скорость дисков БД зависит от планируемого количества получателей.

- Аудитория до 1 млн получателей в одной рассылке или свыше 1 млн получателей в месяц — не менее 300 IOPS, < 8 ms.
- Аудитория от 1 млн до 2 млн получателей в одной рассылке или свыше 3 млн получателей в месяц — не менее 500 IOPS, < 8 ms;
- Аудитория от 2 млн до 5 млн получателей в одной рассылке или более 5 млн получателей в месяц — не менее 1000 IOPS, < 7 ms.

Требования к серверам для сервиса синхронизации Exchange Listener

В Exchange Listener обработка запросов производится отдельными обработчиками, каждый из которых одновременно обслуживает 40 активных почтовых ящиков.

Количество обработчиков указывается при установке микросервисного компонента и зависит от планового количества почтовых ящиков.

Для одного обработчика необходимо 30% процессорного времени 1ГГц процессора и 850 Мб оперативной памяти (дисковое хранилище не используется).

Требования к программному обеспечению

Компонент	Программное обеспечение
Веб-сервер (приложение на .NET Framework)	Windows Server 2012 R2, 2016, 2019 IIS: .Net framework 4.7.2
Веб-сервер (приложение на .NET Core)	Linux Debian 8-9 .NET Core версии 3.1. Kestrel GDI+ совместимый API для альтернативных Windows операционных систем. Библиотеки разработки и файлы заголовков для GNU C.
Сервер баз данных (Windows)	Windows Server 2012 R2, 2016, 2019 MSSQL 2016 и выше Oracle 11g и выше PostgreSQL 11 и выше
Сервер баз данных (Linux)	Linux Debian 8-9 PostgreSQL 11 и выше
Сервер кэширования	Linux Debian 8-9; Redis Server 4.0 и выше
Сервер балансировщика нагрузки	Linux Debian 8-9; HAProxy
Сервер системы контроля версий (SVN)	Linux Debian 8-9; SVN
Контейнеры	Linux Debian 8-9; Docker; Kubernetes

Развернуть базу данных Oracle

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Установите Oracle Database на сервере базы данных. Ознакомиться с описанием программы и руководством по установке вы можете в [документации Oracle Database](#). Утилиты sqlplus и impdp, которые требуются для восстановления базы данных приложения из резервной копии, устанавливаются вместе с сервером Oracle.

На заметку. Развертывание Creatio с отказоустойчивостью не тестировалось на Oracle, однако известны случаи успешной установки с использованием Oracle RAC. Подробнее о настройке

кластеров на Oracle читайте в [документации Oracle Database](#).

После установки Oracle Database вам необходимо создать пользователей базы данных.

- Пользователь с ролью **"admin"** и неограниченными полномочиями на уровне сервера базы данных — нужен для восстановления базы данных и настройки доступа к ней.
- Пользователь с ролью **"public"** и ограниченными полномочиями — используется для настройки безопасного подключения Creatio к базе данных через аутентификацию средствами Oracle Database.

Подробнее о создании пользователей и настройке прав читайте в [документации Oracle Database](#).

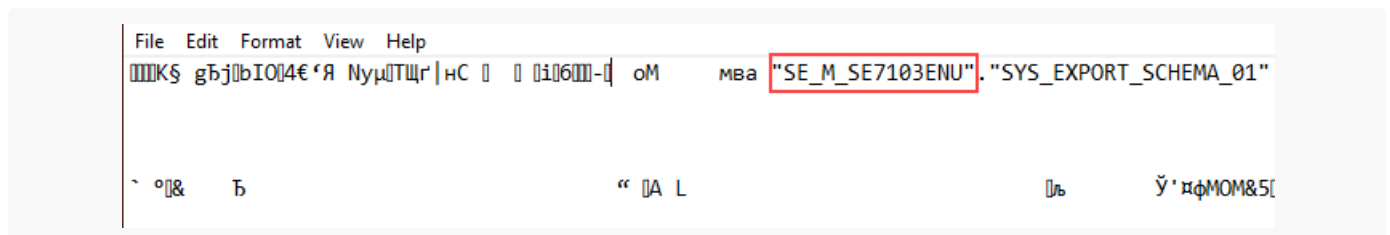
Скачайте и разархивируйте [архив с SQL-скриптами](#), которые используются для восстановления базы данных из резервной копии Oracle.

По умолчанию файл резервной копии базы данных Oracle *.dmp находится в директории ~\db с исполняемыми файлами Creatio. Если файл находится не на сервере Oracle, то необходимо разместить его в сетевой папке с общим доступом.

Чтобы восстановить базу данных:

1. Откройте скрипты CreateUser.sql и RecompileSchema.sql в текстовом редакторе и замените значения макросов:
 - a. YOUR_SCHEMA_NAME — имя схемы;
 - b. YOUR_SCHEMA_PASSWORD — пароль для схемы;
 - c. \\your_server.com\Share — путь к файлу резервной копии *.dmp.
2. Откройте файл резервной копии базы данных *.dmp в текстовом редакторе, найдите и сохраните название используемой схемы, которое находится перед записью ".SYS_EXPORT_SCHEMA" ([Рис. 1](#)).

Рис. 1 — Просмотр названия схемы в файле резервной копии базы данных



3. Перенесите отредактированные скрипты на сервер Oracle. Из директории со скриптами в командной строке выполните команду для создания новой схемы:

```
sqlplus.exe "SYS/SYS_PASSWORD@your_server.com:1521/YOUR_SERVICE_NAME AS SYSDBA" @CreateUser.s
```

- a. SYS_PASSWORD — пароль для авторизации на сервере Oracle;
 - b. your_server.com — сетевой адрес сервера Oracle;
 - c. YOUR_SERVICE_NAME — имя сервиса Oracle.
4. Выполните импорт резервной копии базы данных в созданную схему:

```
impdp "YOUR_SCHEMA_NAME/YOUR_SCHEMA_NAME@//your_server.com:1521/BPMBUILD"
  REMAP_SCHEMA=ORIGINAL_SCHEMA_NAME:YOUR_SCHEMA_NAME
  DIRECTORY=BACKUPDIR DUMPFILE=filename.dmp NOLOGFILE=YES
```

- a. YOUR_SCHEMA_NAME — имя схемы, прописанной в CreateUser.sql;
 - b. your_server.com — сетевой адрес сервера Oracle;
 - c. ORIGINAL_SCHEMA_NAME — название схемы из файла резервной копии (шаг 2).
5. Последовательно выполните команды:

```
sqlplus.exe "YOUR_SCHEMA_NAME/YOUR_SCHEMA_PASSWORD@your_server.com:1521/YOUR_SERVICE_NAME"
@tspkg_UtilityGlobalTypes.sql
```

```
sqlplus.exe "YOUR_SCHEMA_NAME/YOUR_SCHEMA_PASSWORD@your_server.com:1521/ YOUR_SERVICE_NAME"
@RecompileSchema.sql
```

Настроить ConnectionStrings.config для MS SQL Server

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

В файле ConnectionStrings.config, который находится в корневой папке сайта Creatio, содержатся параметры подключения вашего приложения к базе данных и внешним сервисам.

Настроить ConnectionStrings.config

1. Перейдите в корневой каталог сайта Creatio ~\WebAppRoot\Creatio.
2. Откройте в текстовом редакторе файл ConnectionStrings.config.
3. Укажите параметры подключения (**connectionStrings**) вашего сайта.

Пример настроенного файла ConnectionStrings.config

```
<?xml version="1.0" encoding="utf-8"?>

<connectionStrings>

<add name="db" connectionString="Data Source=[ Имя сервера базы данных ]; Initial Catalog=[ Имя

<add name="redis" connectionString="host=[ Имя компьютера ];db=[ Номер БД Redis ];port=6379; max
```

```

<add name="defRepositoryUri" connectionString="" />

<add name="defWorkingCopyPath" connectionString="%TEMP%\%WORKSPACE%" />

<add name="defPackagesWorkingCopyPath" connectionString="%TEMP%\%APPLICATION%\%WORKSPACE%\Terrasoft" />

<add name="clientUnitContentPath" connectionString="%TEMP%\%APPLICATION%\%WORKSPACE%\ClientUnits" />

<add name="sourceControlAuthPath" connectionString="%TEMP%\%APPLICATION%\%WORKSPACE%\Svn" />

<add name="elasticsearchCredentials" connectionString="User=[ Имя пользователя ElasticSearch ];" />

</connectionStrings>

```

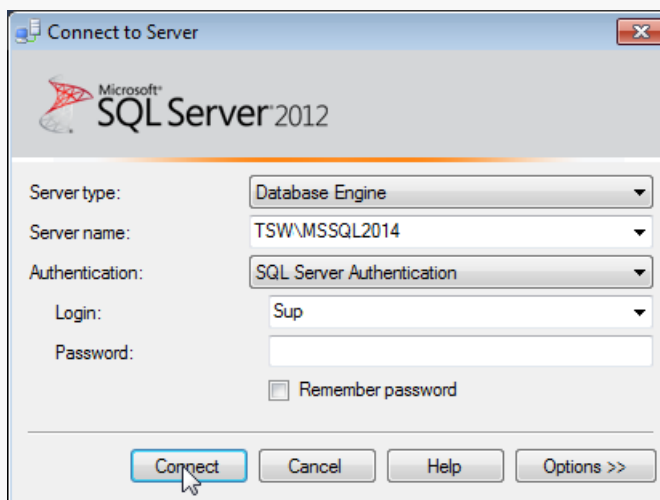
Обязательные настройки ConnectionStrings.config

Параметры подключения к базе данных и серверу кэширования необходимы для работы приложения Creatio.

- **name="db"** — отвечает за подключение к восстановленной базе данных .

Имя сервера баз данных (**Data Source**) можно увидеть в окне авторизации при подключении к серверу из Microsoft SQL Server Management Studio (Рис. 1).

Рис. 1 — Окно авторизации на SQL-сервер



Имя базы данных (Initial Catalog) должно совпадать со значением поля [*Database*], которое вы указали при восстановлении базы данных.

По умолчанию для подключения Creatio к серверу базы данных используется **аутентификация Windows** (Integrated Security), основанная на интерфейсе SSPI. Чтобы обеспечить успешное подключение к БД, необходимо для приложения указать пользователя Windows, под которым будет выполняться подключение к серверу базы данных.

```
<add name="db" connectionString="Data Source=[ Имя сервера базы данных ];
Initial Catalog=[ Имя базы данных ];
Persist Security Info=True; MultipleActiveResultSets=True;
Integrated Security=SSPI; Pooling = true; Max Pool Size = 100; Async = true" />
```

Если же вы хотите авторизоваться на сервере базы данных **при помощи логина и пароля пользователя MS SQL**, то необходимо предварительно создать логин и пароль такого пользователя на сервере MS SQL и внести изменения в файл ConnectionStrings.config. В строке подключения к базе данных (add name="db") замените переменную **Integrated Security=SSPI** на **User ID** и **Password**:

```
<add name="db" connectionString="Data Source=TSW\MSSQL2014;
Initial Catalog=7.10.2.1416_SalesEnterprise_Demo;
Persist Security Info=True; MultipleActiveResultSets=True;
User ID=Sup; Password=password; Pooling = true; Max Pool Size = 100; Async = true" />
```

- **name="redis"** — отвечает за взаимодействие с сервером Redis.

```
<add name="redis" connectionString="host=[ Имя компьютера ];db=[ Номер БД Redis ];port=6379;
maxReadPoolSize=10;maxWritePoolSize=500" />
```

Необязательные настройки ConnectionStrings.config

Параметры подключения к внешним сервисам опциональны, они заполняются только в том случае, когда необходимо, например, при подключении системы контроля версий.

- **tempDirectoryPath** — путь к временной директории, используемой механизмом установки пакетов:

```
<add name="tempDirectoryPath" connectionString=[ Путь к временной директории, используемой ме
```

- **defPackagesWorkingCopyPath** — путь к рабочей копии пользовательских пакетов приложения (только при использовании системы контроля версий SVN). Содержимое рабочей копии представляет собой пользовательские пакеты в виде набора папок и файлов. Оно синхронизируется с репозиторием системы контроля версий SVN при помощи встроенного в Creatio клиента SVN. Данный параметр настраивается при подключении системы контроля версий и используется только в режиме разработки по умолчанию, без включения режима разработки в файловой системе. Значение по умолчанию указывает на временный каталог, который может быть очищен операционной системой. Рекомендуется установить значение, указывающее на пользовательский каталог, так как использование каталогов приложения, например, .\Terrasoft.WebApp\Terrasoft.Configuration\Pkg, может привести к ошибкам компиляции.

```
<add name="defPackagesWorkingCopyPath" connectionString=[ Путь к рабочей копии пользовательск
```

- **sourceControlAuthPath** — путь к данным авторизации для встроенного клиента системы контроля версий SVN (если используется). Значение по умолчанию указывает на временный каталог, который может быть очищен операционной системой. При использовании системы контроля версий рекомендуем указать в данном параметре путь к фиксированному каталогу.

```
<add name="sourceControlAuthPath" connectionString="[ Путь к данным авторизации системы хранения ]"/>
```

- **influx** — отвечает за взаимодействие с сервисом сбора метрик сайта. Данный параметр заполняется только в том случае, когда необходимо собрать статистику использования функциональности для последующей отладки.

```
<add name="influx" connectionString="url=[ Адрес сервиса сбора метрик сайта ]; user=[ Пользователь ]"/>
```

- **clientPerformanceLoggerServiceUri** — отвечает за взаимодействие с сервисом логгирования. Данный параметр заполняется только в том случае, когда необходимо собрать информацию о загрузке страниц приложения.

```
<add name="clientPerformanceLoggerServiceUri" connectionString="[ Адрес сервиса логгирования ]"/>
```

- **messageBroker** — отвечает за взаимодействие с сервисом RabbitMQ. Данный параметр заполняется только в том случае, когда необходимо настроить горизонтальное масштабирование нагрузки с помощью RabbitMQ.

```
<add name="messageBroker" connectionString="amqp://[ Имя пользователя MessageBroker ]:[ Пароль ]@localhost:5672"/>
```

Установить Creatio .NET Core на Linux

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Перед тем, как развернуть сервер, необходимо выполнить следующие действия:

- Подготовить установочные файлы Creatio. [Подробнее >>>](#)
- Развернуть сервер базы данных. [Подробнее >>>](#)
- Развернуть сервер кэширования данных Creatio (Redis). [Подробнее >>>](#)
- Внести изменения в файл конфигурации ConnectionStrings.config. [Подробнее >>>](#)

На заметку. Запуск сервера PostgreSQL в Docker описан в [документации Docker](#).

Способ 1. Установить Creatio .NET Core непосредственно

на Linux

Чтобы развернуть сервер приложения:

- Установите .NET Core, GDI+ совместимый API для семейства UNIX-подобных операционных систем, библиотеки разработки и файлы заголовков для GNU C. [Подробнее >>>](#)
- Запустите сервер приложения Creatio. [Подробнее >>>](#)

Установить .NET Core и другие зависимости Creatio

1. Скачайте пакеты microsoft-prod package:

```
wget -q https://packages.microsoft.com/config/ubuntu/18.04/packages-microsoft-prod.deb -O pac
```

2. Войдите в систему как администратор (root):

```
sudo su
```

3. Установите загруженные пакеты:

```
dpkg -i packages-microsoft-prod.deb
```

4. Обновите список пакетов:

```
apt-get update
```

5. Установите транспорт APT для загрузки по протоколу HTTP Secure:

```
apt-get install apt-transport-https
```

6. Обновите список пакетов:

```
apt-get update
```

7. Установите .NET Core:

```
apt-get install dotnet-sdk-3.1
```

8. Установите GDI+ совместимый API для альтернативных Windows операционных систем:

```
apt-get install -y libgdiplus
```

9. Установите библиотеки разработки и файлы заголовков для GNU C:

```
apt-get install -y libc6-dev
```

10. Выйдите из root-сессии:

```
exit
```

Запуск сервера приложения Creatio

На заметку. Если вы разворачиваете среду разработки .Net Core с доступом по протоколу http, то перед запуском приложения необходимо внести изменения в конфигурационный файл Terrasoft.WebHost.dll.config, который находится в корневой папке приложения. Для параметра "add key" укажите значение

```
<add key="CookiesSameSiteMode" value="Lax" />
```

Это обеспечит корректную работу как по протоколу https, так и по http. В тоже время при такой настройке не будет работать мобильное приложение.

Чтобы запустить приложение:

1. Перейдите в папку с установочными файлами Creatio:

```
cd /path/to/application/directory/
```

2. Запустите сервер .Net Core:

```
COMPlus_ThreadPool_ForceMinWorkerThreads=100 dotnet Terrasoft.WebHost.dll
```

Версия Creatio для HTTP будет доступна через порт 5000.

Версия Creatio для HTTPS будет доступна через порт 5002.

На заметку. Первый вход в систему после настройки приложения осуществляется под учетной записью Supervisor (логин — Supervisor; пароль — Supervisor).

Способ 2. Установить Creatio .NET Core на Linux с использованием Docker

Используйте этот метод развертывания для запуска приложения Creatio изолированно от операционной системы. Установка сервера Redis, восстановление БД и настройка файла ConnectionStrings.config выполняются аналогично, по описанной выше инструкции.

Чтобы развернуть сервер приложения с использованием Docker:

- Настройте доступ к серверу кэширования данных (Redis) из контейнера Docker. [Подробнее >>>](#)
- Установите Docker. [Подробнее >>>](#)
- Создайте Dockerfile. [Подробнее >>>](#)
- Создайте и запустите образ Docker. [Подробнее >>>](#)

Важно. Развертывание приложения в Docker рекомендуется для сред разработки и тестирования. Развертывание промышленной среды в Docker станет доступно в будущих релизах, после реализации возможности обновления приложений в Docker.

Настроить сервер кэширования данных Creatio (Redis)

1. Откройте **redis.conf** в текстовом редакторе от имени пользователя root. Например, для этого можно использовать текстовый редактор Nano:

```
sudo nano /etc/redis/redis.conf
```

2. Найдите запись "**bind 127.0.0.1 ::1**". Замените ее на запись "**bind 0.0.0.0**", чтобы использовать все доступные интерфейсы IPV4.
3. Сохраните изменения и закройте текстовый редактор.
4. Перезагрузите сервер Redis:

```
sudo systemctl restart redis-server
```

Установить Docker

Чтобы установить Docker, выполните команду:

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

Создать Dockerfile

/path/to/application/directory/ — путь к папке, в которую распакован архив с установочными

файлами Creatio.

1. Перейдите в папку приложения:

```
cd /path/to/application/directory/
```

2. Создайте Dockerfile в текстовом редакторе. Например, для этого можно использовать текстовый редактор Nano:

```
nano Dockerfile
```

3. Добавьте следующий код:

```
FROM mcr.microsoft.com/dotnet/core/sdk:3.1 AS base
EXPOSE 5000 5002
RUN apt-get update && \
apt-get -y --no-install-recommends install \
libgdiplus \
libc6-dev && \
apt-get clean all && \
rm -rf /var/lib/apt/lists/* /var/cache/apt/*
WORKDIR /app
COPY . ./
FROM base AS final
WORKDIR /app
ENV ASPNETCORE_ENVIRONMENT Development
ENV TZ US/Eastern
ENV COMPlus_ThreadPool_ForceMinWorkerThreads 100
ENTRYPOINT ["dotnet", "Terrasoft.WebHost.dll"]
```

4. Нажмите Ctrl+O, чтобы сохранить изменения.
5. Закройте редактор нажатием Ctrl+X.

Создать и запустить образ Docker

На заметку.. Если вы разворачиваете среду разработки .Net Core с доступом по протоколу http, то перед запуском образа Docker необходимо внести изменения в конфигурационный файл Terrasoft.WebHost.dll.config, который находится в корневой папке приложения. Для параметра "add key" укажите значение

```
<add key="CookiesSameSiteMode" value="Lax" />
```

Это обеспечит корректную работу как по протоколу https, так и по http. В тоже время при такой

настройке не будет работать мобильное приложение.

Создайте образ Docker:

```
docker build -f Dockerfile -t creatioimg .
```

Запустите созданный образ:

```
docker run -p http_port_number:5000 -p https_port_number:5002 -d --dns=DNS_server_ip --dns-sear
```

http_port_number — номер порта. Docker будет предоставлять версию HTTP через этот порт.

https_port_number — номер порта. Docker будет предоставлять версию HTTPS через этот порт.

DNS_server_ip — IP-адрес сервера DNS, который позволяет серверу определять адреса интернет-доменов. Вы можете использовать несколько значений **--dns** для разных DNS-серверов.

DNS_address_suffix — поисковый домен DNS, который позволяет контейнеру искать не полностью определенные имена хостов. Вы можете использовать несколько значений **--dns-search** для разных поисковых доменов DNS.

На заметку. Добавьте **--restart**, чтобы по команде всегда использовался определенный контейнер.

Версия Creatio для HTTP будет доступна через порт **http_port_number**.

Версия Creatio для HTTPS будет доступна через порт **https_port_number**.

На заметку. Первый вход в систему после настройки приложения осуществляется под учетной записью Supervisor (логин — Supervisor; пароль — Supervisor).

Настроить работу Creatio .NetCore на https

Прежде чем приступить к работе в приложении по протоколу HTTPS, вам необходимо:

1. Получить цифровой сертификат от центра сертификации в формате *.pfx.

На заметку. При использовании самоподписанного сертификата подключение мобильного приложения к сайту Creatio будет недоступно из-за ограничений политики безопасности мобильных приложений.

2. Открыть файл appsettings.json, который находится в корневом каталоге приложения.
3. В блоке "Https" указать адрес вашего сайта, путь к полученному сертификату и пароль.

Пример блока "Https" файла appsettings.json

```
"Https": {
  "Url": "https://::5002",
  "Certificate": {
    "Path": "C:\\Projects\\site\\20210215_103239\\localhost.pfx",
    "Password": "Password"
  }
}
```

На заметку. Вы можете указать как относительный путь к сертификату, так и абсолютный. Обратите внимание, что абсолютный путь должен быть указан в json-совместимом формате.

Настроить интеграцию с лендингами

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Функциональность доступна во всех продуктах, в которых есть **раздел Лендинги и web-формы**.

Клиентам, у которых приложение Creatio развернуто on-site, может потребоваться дополнительная настройка для корректного формирования HTML-кода на странице лендинга. Настройку необходимо выполнить, если по требованиям безопасности URL, который отображается в браузере пользователя, и URL, используемый для внешнего доступа к Creatio, различаются. Например, если адрес блокируется с помощью firewall.

Для настройки:

1. Перейдите в дизайнер системы —> [*Системные настройки*].
2. В группе [*Настройки раздела Лендинги*] откройте системную настройку “**URL сервиса приема данных со страниц лендингов**”.
3. В поле [*Значение по умолчанию*] введите **полный внешний адрес** вашего приложения Creatio, например, `http://creatio-marketing.mydomain.com`, и сохраните настройку.

В результате в HTML-коде, встраиваемом на вашу посадочную страницу, будет использоваться корректный адрес для вызова веб-сервиса, отвечающего за создание нового лида в Creatio, например:

```
serviceUrl: "http://mysite.creatio-marketing/ServiceModel/GeneratedWebFormService.svc/SaveWebFor
```

Если вы используете **протокол защищенного соединения**, то введите адрес с указанием `https://`, тогда адрес для вызова веб-сервиса будет следующим:

```
serviceUrl: "https://mysite.creatio-marketing/ServiceModel/GeneratedWebFormService.svc/SaveWebFc
```

На заметку. По умолчанию значение данной настройки не заполнено, путь к приложению формируется автоматически.

Настроить Redis Sentinel

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Важно. В Creatio версии 7.18.3 будет прекращена поддержка устаревшего механизма Redis Sentinel. Рекомендуем после обновления Creatio до версии 7.18.0 перейти на механизм [Redis Cluster](#).

Отказоустойчивость хранилищ Redis, работающих с Creatio, обеспечивается при помощи механизма [Redis Sentinel](#). Он предоставляет следующие возможности:

- **Мониторинг.** Sentinel следит за тем, чтобы главный (master) и подчиненные (slave) экземпляры Redis работали корректно.
- **Предупреждение.** Sentinel может уведомлять администратора о проблемах с экземплярами Redis.
- **Автоматическое восстановление работоспособности (failover).** Если master-экземпляр Redis не работает как ожидается, то Sentinel может назначить один из slave-экземпляров главным, а другие slave-экземпляры переконфигурировать на работу с новым master. При этом приложения Creatio, использующие Redis, оповещаются о новом адресе соединения с Redis.

Важно. Работа с кластером Redis в Creatio версий 7.17.4 и ниже не поддерживается.

Redis Sentinel является распределенной системой, которая предназначена для работы нескольких экземпляров Sentinel, взаимодействующих друг с другом. Преимущества такой системы:

- Факт отказа подтверждается только когда несколько экземпляров Sentinel, составляющих кворум, соглашаются с недоступностью master-экземпляра Redis. Это уменьшает количество ложных отказов.
- Механизм Sentinel работает, даже если некоторые экземпляры Sentinel неработоспособны. Это повышает отказоустойчивость системы.

Особенности Sentinel

- Для обеспечения отказоустойчивости необходимо не менее трех экземпляров Sentinel, запущенных на разных физических или виртуальных компьютерах. При этом должна быть четкая уверенность, что их отказы вызваны разными причинами.
- Из-за асинхронной репликации распределенная система Sentinel + Redis не гарантирует, что все данные будут сохранены во время отказа.
- Отказоустойчивость настроенной конфигурации должна регулярно проверяться и подтверждаться тестовыми отказами.
- При использовании Docker невозможны некоторые процессы Sentinel, т. к. Docker может перенаправлять порты (описано в блоке "Sentinel, Docker, NAT, and possible issues" [документации](#))

[Sentinel](#)).

Минимальная отказоустойчивая конфигурация Redis Sentinel

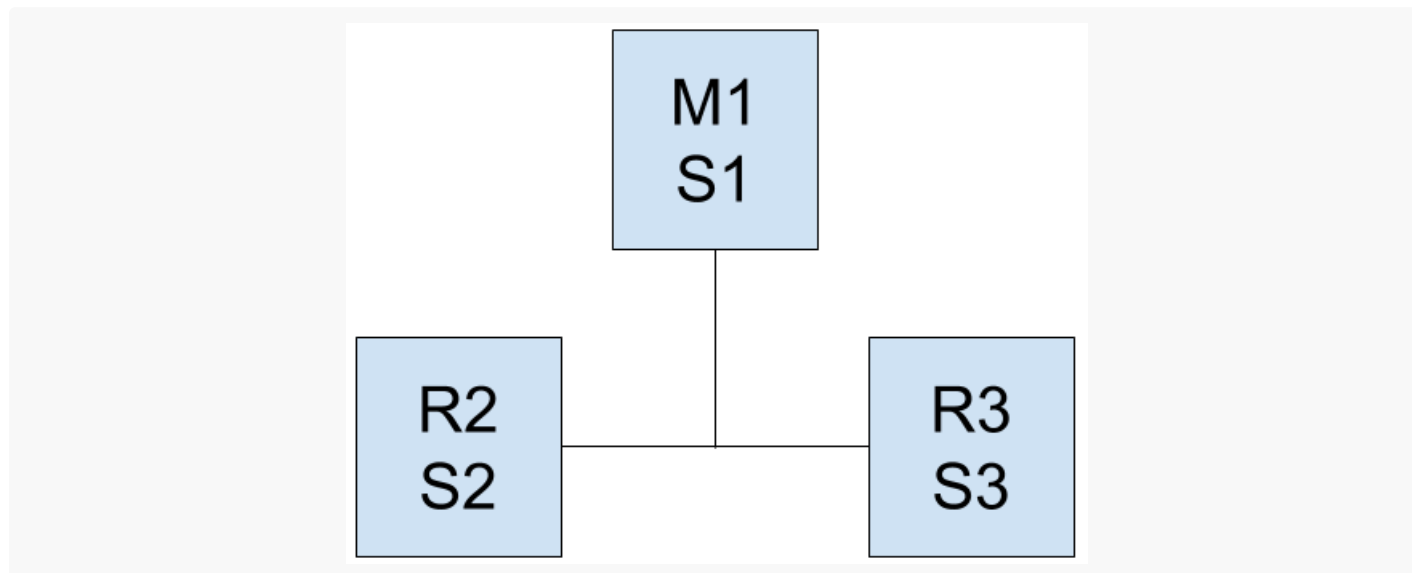
Условные обозначения:

- M1, M2 — master-экземпляры Redis.
- R1, R2, R3 — slave-экземпляры Redis (от Replica).
- S1, S2, S3 — экземпляры Sentinel.
- C1 — клиентское приложение (Creatio).
- [M2] — экземпляр, сменивший свою роль (например из slave на master).

Рекомендуется использовать конфигурацию минимум с тремя экземплярами Redis и тремя экземплярами Sentinel (подробнее о конфигурации читайте в блоке "Example 2: basic setup with three boxes"

[документации Sentinel](#)). Эта конфигурация основана на трех узлах (физических или виртуальных компьютерах), каждый из которых содержит запущенные экземпляры Redis и Sentinel (Рис. 1). Два экземпляра Sentinel (S2 и S3) составляют кворум (quorum) — количество экземпляров, необходимых для определения отказа текущего master-экземпляра Redis.

Рис. 1 — Конфигурация "трех узлов": quorum = 2



При нормальной работе конфигурации клиентское приложение (Creatio) записывает свои данные в master-экземпляр M1. Затем эти данные асинхронно реплицируются в slave-экземпляры R2 и R3.

Если master-экземпляр Redis M1 становится недоступным, то экземпляры Sentinel S1 и S2 совместно решают, что произошел отказ и начинают процесс восстановления работоспособности (failover). Они назначают один из slave-экземпляров Redis (R2 или R3) новым master-экземпляром, с которым продолжает работать клиентское приложение.

Важно. В любой конфигурации Sentinel, в которой данные реплицируются асинхронно, существует риск потери записей. Это возможно, поскольку данные могут не попасть на slave-экземпляр Redis,

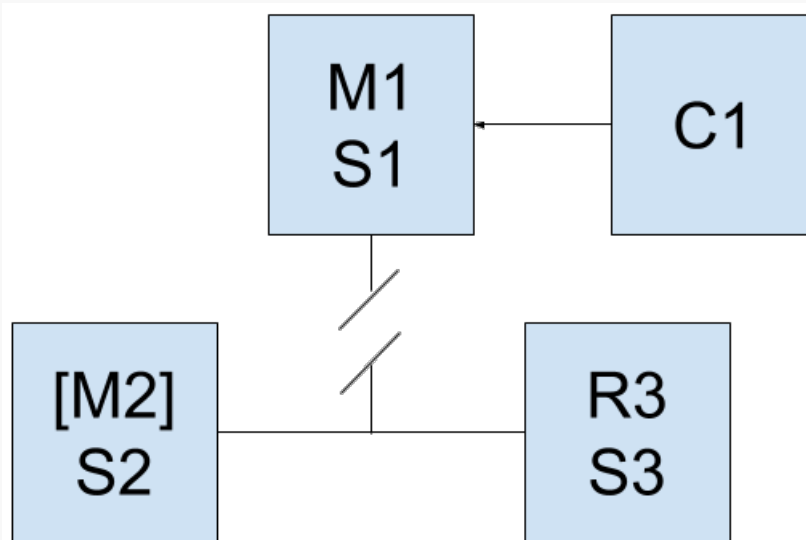
ставший новым master-экземпляром.

На заметку. Другие возможные отказоустойчивые конфигурации описаны в [документации Sentinel](#).

Проблема разделения сети

При потере сетевого соединения существует риск, что клиентское приложение C1 продолжит работать со старым master-экземпляром Redis M1, в то время как будет назначен новый master-экземпляр [M2] (Рис. 2).

Рис. 2 — Разделение сети



Подобной ситуации можно избежать, включив опцию остановки записи данных, если master-экземпляр обнаруживает уменьшение количества slave-экземпляров. Для этого можно, например, установить следующие значения в конфигурационном файле `redis.conf` master-экземпляра Redis:

```
min-slaves-to-write 1
min-slaves-max-lag 10
```

В результате master-экземпляр Redis M1 перестанет принимать данные через 10 секунд, если не сможет их передать как минимум одному slave-экземпляру. После восстановления работоспособности системы экземплярами Sentinel, составляющими кворум (S2 и S3), клиентское приложение C1 будет переконфигурировано для работы с новым master-экземпляром [M2].

Важно. После остановки master-экземпляр не сможет автоматически продолжить работу, если сеть будет восстановлена. Если оставшийся slave-экземпляр Reids (R3) также станет недоступным, то система полностью прекратит работу.

Системные требования

Redis является размещаемой в памяти базой данной (in-memory database). Поэтому основным требованием является скорость работы и объем оперативной памяти. Кроме того, поскольку Redis является однопоточным приложением и нагружает только одно ядро процессора, то для работы одного экземпляра Reids необходим узел (физический или виртуальный компьютер) с двухъядерным процессором, как минимум. Экземпляры Sentinel потребляют относительно немного ресурсов и могут работать на одном узле с Redis.

Redis и Sentinel рекомендуется разворачивать на операционных системах Linux.

В таблице приведены рекомендуемые системные требования для одного узла (физической или виртуальной машины) в зависимости от количества пользователей Creatio.

Количество пользователей	ЦПУ (CPU)	Оперативная память
1-300	Intel Xeon E3-1225v5	2 Гб
300-500		4 Гб
500-1000		6 Гб
1000-3000		12 Гб
3000-5000		18 Гб
5000-7000		26 Гб
7000-10000		36 Гб

Установить и настроить Redis Sentinel

Redis Sentinel поставляется вместе с дистрибутивом Redis. Процесс установки описан в [документации Redis](#). Для установки следует использовать новейшую версию Redis на дату релиза Creatio.

Пример настройки конфигурации Redis Sentinel приведен в разделе "A quick tutorial" документации Sentinel.

Настроить Creatio для работы с Redis Sentinel

Адаптированные библиотеки и настройка ConnectionStrings.config

Для получения адаптированных библиотек обратитесь в службу технической поддержки Creatio.

В строке соединения **"redis"** укажите:

- `sentinelHosts` — перечисленные через запятую адреса и порты экземпляров Sentinel в формате `<адрес>:<порт>`. Количество может быть любым.
- `masterName` — имя master-экземпляра Redis.

Пример строки соединения:

```
<add name="redis" connectionString="sentinelHosts=localhost:26380,localhost:26381,localhost:26382"
```

Web.config

Убедитесь, что в секции **appSettings** используются рекомендуемые параметры:

- `Feature-UseRetryRedisOperation` — включает внутренний механизм `CreateIoCompletionPort` для повтора операций с Redis, которые завершились с ошибкой.
- `SessionLockTtlSec` — срок действия ключа блокировки сессии.
- `SessionLockTtlProlongationIntervalSec` — период, на который продлевается срок действия ключа блокировки сессии.

Настройки должны иметь следующие значения:

```
<add key="Feature-UseRetryRedisOperation" value="true" />
<add key="SessionLockTtlSec" value="60" />
<add key="SessionLockTtlProlongationIntervalSec" value="20" />
```

Убедитесь, что в секции **redis** используются рекомендуемые параметры:

- `enablePerformanceMonitor` — включает мониторинг времени выполнения операций с Redis. Рекомендуется включать для отладки и поиска проблем. Значение по умолчанию: По умолчанию выключен т. к. влияет на производительность приложения.
- `executionTimeLoggingThresholdSec` — операции с Redis, которые выполнялись дольше указанного времени, будут записаны в лог. По умолчанию 5 секунд.
- `featureUseCustomRedisTimeouts` — включает использование таймаутов, указанных в конфигурационном файле. Значение по умолчанию: “отключена”.
- `clientRetryTimeoutMs` — завершенные с ошибкой операции с Redis повторяются с тем же клиентом и соединением до истечения периода, заданного этим параметром. Используется для устранения ошибок, вызванных кратковременными перерывами в работе сети. При этом нет необходимости получать из пула нового клиента. По умолчанию 4000 миллисекунд.
- `clientSendTimeoutMs` — время, выделяемое на отправку запроса серверу Redis. По умолчанию 3000 миллисекунд.
- `clientReceiveTimeoutMs` — время, выделяемое на получение ответа от сервера Redis. По умолчанию 3000 миллисекунд.
- `clientConnectTimeoutMs` — время, выделяемое на установку сетевого соединения с сервером Redis. По умолчанию 100 миллисекунд.

- `deactivatedClientsExpirySec` — задержка удаления сбойных Redis-клиентов после их удаления из пула. 0 означает немедленное удаление. По умолчанию 0.
- `operationRetryIntervalMs` — если внутренний цикл повтора сбойных операций не привел к успешному выполнению операции, то такая операция откладывается на указанное время. После этого операция выполняется с новым клиентом, который, уже может иметь установленное соединение с новым master-экземпляром. По умолчанию 1000 миллисекунд.
- `operationRetryCount` — количество повторных попыток выполнения операции с новым Redis-клиентом. По умолчанию 10.

Настройки должны иметь следующие значения:

```
<redis connectionStringName="redis" enablePerformanceMonitor="false" executionTimeLoggingThreshc
```

Часто задаваемые вопросы по глобальному поиску и дедупликации

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Как узнать, какая версия глобального поиска используется?

Если вы используете версию Creatio cloud, то у вас по умолчанию используется последняя версия глобального поиска.

Если вы используете версию Creatio on-site, то запустите в консоли команду:

```
docker ps
```

В результате вы увидите список всех запущенных контейнеров. Номер версии глобального поиска можно узнать в колонке [*image*].

Как запустить переиндексацию сайта?

Для версии глобального поиска 2.0:

Выполните PUT запрос вида :

```
http://[ GS-WEB-API ]:81/indexation/{siteName}/reindex/full
```

Для версии глобального поиска ниже 2.0:

1. Откройте контейнер `gs-mysql`, выполнив команду:

```
docker exec -it gs-mysql bash
```

2. В контейнере gs-mysql выполните следующие команды:

```
mysql -p1665017 use gs; UPDATE GlobalSearchIndexingEntity SET LastIndexedOn = NULL, InProcess
```

Как включить логирование работы сервиса глобального поиска?

По умолчанию сервис сохраняет только ошибки, возникающие в ходе работы. Чтобы включить логирование всех событий сервиса глобального поиска, в файле docker-compose.yml замените строку:

```
-Log4NetPath=${LOG$NET_CONFIG_FILE:-log4net.production.config}
```

на строку вида:

```
-Log4NetPath=${LOG$NET_CONFIG_FILE:-log4net.debug.config}
```

На заметку. При включении логирования всех событий существенно увеличивается количество файлов журналирования.

Какие метрики или системы слежения можно использовать для мониторинга работы глобального поиска?

Вы можете выполнять запрос на:

```
http://[ GS-WEB-API ]:81/sites/[ SITE_NAME ]/search/state
```

В данном случае [GS-WEB-API] — это адрес сервера, где находятся компоненты глобального поиска, а [SITE_NAME] — название вашего сайта.

Как настроить доступ к Elasticsearch по паролю?

Вы можете закрыть доступ к Elasticsearch с помощью Nginx, поддерживающего base64- авторизацию. Для настройки доступа к Elasticsearch по логину и паролю используйте плагин x-pack.

Как добавить в индекс новый объект или изменить настройки индексируемых полей существующих объектов?

Для **версий Creatio 7.18.4 и ниже** пользовательскими средствами можно включать и отключать поиск для конкретных разделов. По умолчанию индексируются только разделы вне зависимости от их автора, а также текстовые и справочные колонки, кроме ряда исключений. Откройте [файл](#) в формате *.pdf с актуальным перечнем исключений.

Начиная с **версии Creatio 7.18.5** вы можете оптимизировать индексируемые для глобального поиска колонки. Чтобы увеличить быстродействие системы и снизить нагрузку на сервера, вы можете явно указать те колонки, которые не должны индексироваться для глобального поиска. Например, колонки, которые содержат служебную информацию.

Чтобы изменить перечень индексируемых колонок:


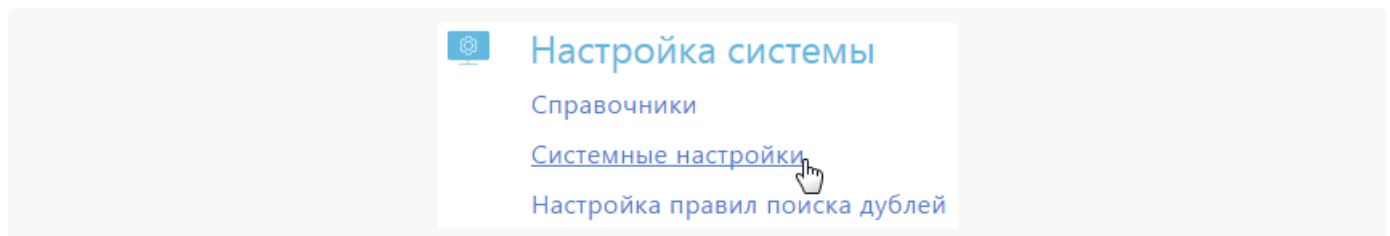
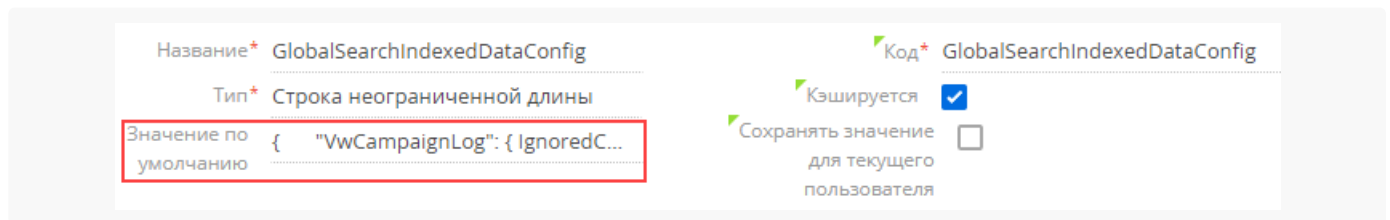
1. Откройте дизайнер системы, например по кнопке .
2. В блоке [*Настройка системы*] перейдите по ссылке [*Системные настройки*] (Рис. 1).

Рис. 1 — Раздел [*Системные настройки*]



3. В реестре системных настроек выберите настройку “GlobalSearchIndexedDataConfig” (код “GlobalSearchIndexedDataConfig”) нажмите кнопку [*Открыть*].
4. На странице системной настройки в поле [*Значение по умолчанию*] указан перечень неиндексируемых колонок в формате JSON (Рис. 2).

Рис. 2 — Поле с перечнем неиндексируемых колонок



5. Скопируйте и отредактируйте локально список неиндексируемых колонок. Для этого вы можете использовать любой онлайн-редактор файлов JSON.

Список неиндексируемых колонок состоит из блоков, каждый из которых содержит уникальный код объекта, который вы можете найти в поле [*Код*] [раздела \[Управление конфигурацией \]](#) и не подлежат индексации. Пример структуры блока представлен ниже:

```
"Contact":{ // код объекта
```

```
"IgnoredColumns":[ //перечень колонок, которые не индексируются
```

```
"dear", //код колонки, которая не должна индексироваться

"salutationtype",

"gender",...]  },
```

На заметку. Если вместо названия колонки (объекта) указано "*", то все колонки (объекты) не будут индексироваться. Если в названии колонки используется нижнее подчеркивание, то исключены будут все колонки, названия которых завершаются таким же фрагментом, как отделенный нижним подчеркиванием в названии. Например, название колонки "_headerproperties" исключит все колонки, заканчивающиеся на "headerproperties".

6. Проверьте готовый JSON на валидность. Например, с помощью онлайн-редактора.
7. Вставьте обновленный список в поле [Значение по умолчанию] и сохраните изменения.
8. Перезапустите индексацию для гарантированного обновления списка колонок в Elasticsearch.

Как развернуть Elasticsearch на нескольких серверах с указанием одного URL? Как настроить кластеризацию?

Подробные ответы на эти вопросы содержатся в [документации сервиса Elastic](#) на английском языке.

Что делать, если не отображается настройка “Правила поиска дублей”?

Проверьте, заполнена ли системная настройка “DeduplicationWebApiUrl” и включена ли функциональность:

- “BulkESDeduplication”;
- “ESDeduplication”;
- “Deduplication”.

Подробнее о включении дополнительной функциональности читайте в статье [Механизм отключения функциональности Feature Toggle](#) (документация по разработке).

Можно ли использовать сервисы глобального поиска и массового поиска дублей одновременно с двумя приложениями Creatio?

Если у вас используются два приложения Creatio, например, продуктовая и тестовая среда, то вы можете подключить сервисы глобального поиска и массового поиска дублей к каждому из приложений. Для установки воспользуйтесь инструкциями:

- [Настроить глобальный поиск](#).

- [Настроить массовый поиск дублей.](#)

Как сортируются результаты поиска?

Порядок отображения результатов поиска в основном зависит от релевантности поискового запроса. На релевантность влияют следующие факторы:

- Количество текста в документе.
- Количество вхождений поисковой фразы в документе.
- Количество вхождений поисковой фразы в индексе, а также ряд других, менее значимых, параметров.

Кроме того, порядок сортировки результатов поиска определяют следующие системные настройки:

- "Вес объекта по умолчанию для глобального поиска" (код "GlobalSearchDefaultEntityWeight") — позволяет повысить приоритет отображения в списке результатов поиска записей, содержащихся в разделе, в котором выполняется поиск. Например, если вы введете поисковый запрос, находясь в разделе [*Контакты*], то в начале списка результатов отобразятся записи из данного раздела.
- "Вес первичной колонки по умолчанию для глобального поиска" (код "GlobalSearchDefaultPrimaryColumnWeight") — позволяет повысить приоритет отображения в списке результатов поиска. Применяется, если совпал поисковый запрос и значение, указанное в первичной колонке такой записи (например, для контакта первичной колонкой является поле [*ФИО*], для контрагента — поле [*Название*]). Таким образом, если поисковый запрос совпадет со значением в первичной колонке записи, то такая запись будет отображена в начале списка результатов поиска.

Данные системные настройки влияют на релевантность, но не гарантируют отображение результатов поисковой выдачи в точно указанном порядке, поскольку сортировка зависит и от других факторов, упомянутых выше.

Поддерживается ли аутентификация Windows в микросервисах?

Поскольку сервис глобального поиска и дедупликации разворачивается средствами Docker, аутентификация Windows в микросервисах не поддерживается.

Развернуть базу данных PostgreSQL (Linux)

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Для установки Creatio вы можете использовать следующие конфигурации баз данных:

- удаленная СУБД (рекомендуется);
- локальный сервер PostgreSQL.

Если у вас уже настроен сервер PostgreSQL, то шаг I можно пропустить.

Если у вас уже есть пользователь с правами администратора для авторизации, создания и изменения баз данных, то шаг II можно пропустить.

I. Установить PostgreSQL

PostgreSQL недоступен в большинстве стандартных репозиториях. Чтобы установить PostgreSQL на Linux:

1. Войдите в систему как администратор (root):

```
sudo su
```

2. Добавьте репозиторий PostgreSQL:

```
echo -e "deb http://apt.postgresql.org/pub/repos/apt/ $(lsb_release -sc)-pgdg main" > /etc/ap
```

3. Импортируйте ключ подписи к репозиторию PostgreSQL:

```
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | apt-key add -
```

4. Обновите список пакетов:

```
apt-get update
```

5. Установите PostgreSQL

```
apt-get install -y postgresql-12
```

6. Выйдите из root-сессии:

```
exit
```

На заметку. Развертывание Creatio с отказоустойчивостью не тестировалось на PostgreSQL. Подробнее о возможности создания кластеров читайте в [документации PostgreSQL](#).

II. Создать пользователя PostgreSQL

Установленный PostgreSQL требует дополнительной настройки, прежде чем вы перейдете к развертыванию Creatio. После первичной установки PostgreSQL вам необходимо создать для него

пользователя, который сможет подключаться к базе данных через логин и пароль и будет иметь права на создание и обновление баз данных. По умолчанию такого пользователя в PostgreSQL нет.

Рекомендуется создать следующих пользователей PostgreSQL:

- Пользователь с ролью **"admin"** и неограниченными полномочиями на уровне сервера базы данных — нужен для восстановления базы данных и настройки доступа к ней.
- Обычный пользователь с ограниченными правами на уровне сервера. используется для настройки безопасного подключения Creatio к базе данных через аутентификацию средствами PostgreSQL.

Если у вас в PostgreSQL уже есть администратор и пользователь с ограниченными правами на уровне сервера базы данных, то данный шаг можно пропустить.

Чтобы создать пользователя PostgreSQL:

1. Войдите как **postgres**:

```
sudo su - postgres
```

2. Откройте оболочку PostgreSQL:

```
psql
```

3. Создайте пользователя **pg_sysadmin**, которому на уровне сервера будут предоставлены права администратора либо набор наиболее важных привилегий:

```
CREATE USER pg_sysadmin;
```

pg_sysadmin — псевдоним, который необходимо заменить актуальным именем пользователя с правами администратора на уровне сервера. Нужен для восстановления базы данных и настройки доступа к ней.

4. Настройте для пользователя pg_sysadmin **права администратора**:

```
ALTER ROLE pg_sysadmin WITH SUPERUSER;
```

В качестве альтернативы вы также можете предоставить ему только наиболее важные привилегии вместо прав администратора.

```
GRANT CREATE DATABASE TO pg_sysadmin;  
GRANT ALTER DATABASE TO pg_sysadmin;
```

Важно. Если не предоставить пользователю pg_sysadmin прав администратора или необходимых привилегий на уровне сервера, это может привести к ошибкам в процессе

восстановления базы данных.

5. Разрешите **pg_sysadmin** авторизацию:

```
ALTER ROLE pg_sysadmin WITH LOGIN;
```

6. Настройте пароль для **pg_sysadmin**:

```
ALTER ROLE pg_sysadmin WITH PASSWORD 'pg_syspassword';
```

pg_password— укажите пароль пользователя **pg_sysadmin** для подключения к серверу PostgreSQL.

7. Создайте нового пользователя с ограниченными правами на уровне сервера базы данных:

```
CREATE USER pg_user;
```

pg_user — псевдоним, который необходимо заменить именем пользователя с ограниченными правами на уровне сервера базы данных. Нужен для подключения к базе данных Creatio.

8. Разрешите **pg_user** авторизацию:

```
ALTER ROLE pg_user WITH LOGIN;
```

9. Настройте пароль для **pg_user**:

```
ALTER ROLE pg_user WITH PASSWORD 'pg_password';
```

pg_password — пароль пользователя **pg_user** для подключения к серверу PostgreSQL.

10. Выйдите из оболочки PostgreSQL.

```
\q
```

11. Выйдите из своей сессии:

```
exit
```

III. Восстановить из бэкапа базу данных PostgreSQL

Для восстановления базы данных из резервной копии необходимы утилиты **psql** и **pg_restore**. Они входят в пакет **postgresql-client-common**.

Если вы устанавливаете локально **postgresql-12** с использованием **apt-get**, то APT установит **postgresql-client-common** как зависимость **postgresql-12**.

Если вы планируете использовать удаленную базу данных PostgreSQL, не устанавливая СУБД PostgreSQL на вашем сервере, то установите пакет **postgresql-client-common** вручную:

```
sudo apt-get install postgresql-client-common
```

Чтобы развернуть базу данных из бэкапа:

1. Установите пароль подключения к серверу БД в переменную окружения:

```
export PGPASSWORD=pg_syspassword
```

pg_syspassword — укажите пароль пользователя с правами администратора или привилегиями “CREATE DATABASE”.

2. Создайте базу данных, в которую будет выполнено восстановление резервной копии:

```
psql --host pg_server_address --port pg_server_port --username=pg_sysadmin --dbname=pg_dbname
```

pg_server_address — укажите адрес сервера PostgreSQL.

pg_server_port — укажите порт сервера PostgreSQL.

pg_sysadmin — укажите имя пользователя postgres для подключения к серверу PostgreSQL.

Пользователь должен являться системным администратором (Superuser) или иметь права на создание базы данных (команда “CREATE DATABASE”).

pg_user — псевдоним, который необходимо заменить именем пользователя с ограниченными правами на уровне сервера базы данных. Нужен для подключения к базе данных Creatio. Вы можете использовать для этого любого пользователя, не имеющего прав администратора на уровне сервера. Чтобы изменить пользователя, от имени которого выполняется подключение, следуйте инструкциям, изложенным в шаге IV данного руководства.

pg_dbname — укажите имя базы данных PostgreSQL, к которой необходимо подключиться для выполнения команды. Для корректного выполнения команды необходимо указать имя существующей базы данных.

На заметку. Если вы еще не создали базу данных или попытка подключения завершается ошибкой “FATAL: database “pg_dbname” does not exist”, то воспользуйтесь БД “template1”, которая создается в PostgreSQL по умолчанию.

pg_dbname_creatio — укажите имя базы данных, которую будет использовать приложение Creatio. База данных будет создана после успешного выполнения команды.

3. Если вы используете AWS RDS:

- Скачайте скрипт [ChangeTypesOwner.sql](#).
- Замените в скрипте значение "postgres" на актуальное имя пользователя Postgres.
- Выполните обновленный скрипт ChangeTypesOwner.sql.

4. Перейдите в папку приложения:

```
cd /path/to/application/directory/
```

/path/to/application/directory/ — путь к папке, в которую распакован архив с установочными файлами Creatio.

5. Перейдите в папку БД:

```
cd db
```

6. Восстановите базу данных из бэкапа:

```
pg_restore --host pg_server_address --port pg_server_port --username=pg_sysadmin --dbname=pg_dbname_creatio
--no-owner --no-privileges /path/to/db.backup
```

pg_server_address — укажите адрес сервера PostgreSQL.

pg_server_port — укажите порт сервера PostgreSQL.

pg_sysadmin — укажите имя пользователя postgres для подключения к серверу PostgreSQL.

Пользователь должен являться системным администратором (Superuser) или иметь максимальные права на уровне базы данных pg_dbname_creatio.

pg_dbname_creatio — имя БД PostgreSQL, в которую будут добавлены таблицы резервной копии. Используйте имя базы данных, указанное в команде "CREATE DATABASE" на шаге 2.

/path/to/db.backup — путь к файлу с базой данных Creatio db.backup.

7. Скачайте файл [CreateTypeCastsPostgreSql.sql](#).

8. Выполните преобразование типов:

```
psql --host=pg_server_address --port=pg_server_port --username=pg_sysadmin --dbname=pg_dbname
```

pg_server_address — укажите адрес сервера PostgreSQL.

pg_server_port — укажите порт сервера PostgreSQL.

pg_sysadmin — укажите имя пользователя postgres для подключения к серверу PostgreSQL.

Пользователь должен являться системным администратором (Superuser) или иметь максимальные права на уровне базы данных Creatio.

pg_dbname_creatio — имя БД, в контексте которой происходит выполнение инструкций.

/path/to/CreateTypeCastsPostgreSql.sql — путь к сохраненному ранее файлу CreateTypeCastsPostgreSql.sql.

В результате база данных Creatio будет восстановлена из резервной копии.

IV. Изменить владельца базы данных (опционально)

В Creatio существует возможность при восстановлении из резервной копии заменить владельца БД и ее объектов на пользователя, не являющегося системным администратором (не Superuser). Вам потребуется скрипт ChangeDbObjectsOwner. Для Postgres версии 10 и ниже: [Скачать скрипт](#). Для Postgres 11 и выше: [Скачать скрипт](#).

Чтобы восстановить БД от имени пользователя, не являющегося администратором:

1. Замените владельца БД:

```
psql --host pg_server_address --port pg_server_port --username=pg_sysadmin --dbname=pg_dbname
```

pg_server_address — укажите адрес сервера PostgreSQL.

pg_server_port — укажите порт сервера PostgreSQL.

pg_sysadmin — укажите имя пользователя postgres для подключения к серверу PostgreSQL.

Пользователь должен являться администратором (Superuser) или иметь привилегии “ALTER DATABASE”.

pg_user — псевдоним, который необходимо заменить актуальным именем пользователя, который назначается новым владельцем БД. Под логином этого пользователя будет осуществляться подключение приложения к БД.

pg_dbname_creatio — имя БД, владелец которой заменяется.

2. Замените владельца объектов БД:

```
psql --host pg_server_address --port pg_server_port --username=pg_sysadmin --dbname=pg_dbname
```

pg_server_address — укажите адрес сервера PostgreSQL.

pg_server_port — укажите порт сервера PostgreSQL.

pg_sysadmin — укажите имя пользователя postgres для подключения к серверу PostgreSQL.

Пользователь должен являться системным администратором (Superuser) или владельцем базы данных Creatio.

pg_user — псевдоним, который необходимо заменить актуальным именем пользователя, который назначен новым владельцем БД. Под логином этого пользователя будет осуществляться подключение приложения к БД.

pg_dbname_creatio — имя БД, владелец которой заменяется.

/path/to/ChangeDbObjectsOwner.sql — путь к сохраненному ранее файлу toChangeDbObjectsOwner.sql.

Вы можете пропустить этот шаг, и в таком случае владельцем БД и ее объектов останется пользователь, от имени которого запускалась команда pg_restore. Обычно это пользователь postgres.

Настроить ConnectionStrings.config для Oracle Database

ПРОДУКТЫ: **ВСЕ ПРОДУКТЫ**

В файле ConnectionStrings.config, который находится в корневой папке сайта Creatio, содержатся параметры подключения вашего приложения к базе данных и внешним сервисам.

Настроить ConnectionStrings.config

1. Перейдите в корневой каталог сайта Creatio **~\WebAppRoot\Creatio**.
2. Откройте в текстовом редакторе файл ConnectionStrings.config.
3. Укажите параметры подключения (**connectionStrings**).

Пример настроенного файла ConnectionStrings.config

```
<?xml version="1.0" encoding="utf-8"?>
<connectionStrings>
  <add name="db" connectionString="Data Source=(DESCRIPTION =
  (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = [ Database server name ])(PORT = 1521))) (CC
  <add name="redis" connectionString="host=[ Имя компьютера ];db=[ Номер БД Redis ];port=6379;
maxReadPoolSize=10;maxWritePoolSize=500" />
  <add name="defRepositoryUri" connectionString="" />
  <add name="defWorkingCopyPath" connectionString="%TEMP%\%WORKSPACE%" />
  <add name="defPackagesWorkingCopyPath"
connectionString="%TEMP%\%APPLICATION%\%WORKSPACE%\TerrasoftPackages" />
  <add name="clientUnitContentPath"
connectionString="%TEMP%\%APPLICATION%\%WORKSPACE%\ClientUnitSrc" />
  <add name="sourceControlAuthPath"
connectionString="%TEMP%\%APPLICATION%\%WORKSPACE%\Svn" />
</connectionStrings>
```

Обязательные настройки ConnectionStrings.config

Параметры подключения к базе данных и серверу кэширования необходимы для работы приложения Creatio.

- **name="db"** — отвечает за подключение к восстановленной базе данных, где
 - [Database server name] — сетевой адрес сервера базы данных.
 - [Oracle service name] — имя сервиса.
 - [Schema name] — имя схемы восстановленной базы данных.
 - [Schema password] — пароль к схеме восстановленной базы данных.

```
<add name="db" connectionString="Data Source=(DESCRIPTION =
  (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = [ Database server name ])(PORT = 1521)))
```

- **name="redis"** — отвечает за взаимодействие с сервером Redis.

```
<add name="redis" connectionString="host=[ Имя компьютера ];db=[ Номер БД Redis ];port=6379
maxReadPoolSize=10;maxWritePoolSize=500" />
```

Необязательные настройки ConnectionStrings.config

Параметры подключения к внешним сервисам опциональны, они заполняются только в том случае, когда необходимо, например, при подключении системы контроля версий.

- **tempDirectoryPath** — путь к временной директории, используемой механизмом установки пакетов:

```
<add name="tempDirectoryPath" connectionString=[ Путь к временной директории, используемой ме
```

- **defPackagesWorkingCopyPath** — путь к рабочей копии пользовательских пакетов приложения (только при использовании системы контроля версий SVN). Содержимое рабочей копии представляет собой пользовательские пакеты в виде набора папок и файлов. Оно синхронизируется с репозиторием системы контроля версий SVN при помощи встроенного в Creatio клиента SVN. Данный параметр настраивается при подключении системы контроля версий и используется только в режиме разработки по умолчанию, без включения режима разработки в файловой системе. Значение по умолчанию указывает на временный каталог, который может быть очищен операционной системой. Рекомендуется установить значение, указывающее на пользовательский каталог, так как использование каталогов приложения, например, .\Terrasoft.WebApp\Terrasoft.Configuration\Pkg, может привести к ошибкам компиляции.

```
<add name="defPackagesWorkingCopyPath" connectionString=[ Путь к рабочей копии пользовательск
```

- **sourceControlAuthPath** — путь к данным авторизации для встроенного клиента системы контроля версий SVN (если используется). Значение по умолчанию указывает на временный каталог, который может быть очищен операционной системой. При использовании системы контроля версий рекомендуем указать в данном параметре путь к фиксированному каталогу.

```
<add name="sourceControlAuthPath" connectionString=[ Путь к данным авторизации системы хранен
```

Настроить уведомления о разрешении обращений

ПРОДУКТЫ: [SERVICE ENTERPRISE](#) [CUSTOMER CENTER](#)

Чтобы клиент автоматически получил email-сообщение после перевода обращения в статус “Решено”:

1. Перейдите в дизайнер системы —> [Системные настройки].

2. Откройте системную настройку **“Адрес сайта”**.
3. В поле [*Значение по умолчанию*] введите полный адрес сайта, используемый для доступа к Creatio, например, `http://creatio.com`.
4. Примените настройки таблицы по кнопке [*Сохранить*].

Настроить Redis Cluster

ПРОДУКТЫ: **ВСЕ ПРОДУКТЫ**

Отказоустойчивость хранилищ Redis, работающих с Creatio, обеспечивается при помощи механизма [Redis Cluster](#).

Важно. Работа с Redis Cluster доступна в Creatio версии 7.18.0 и выше.

Отказоустойчивость Redis Cluster

Механизм Redis Cluster работает, даже если большая часть экземпляров Redis неработоспособны. Это достигается благодаря следующим свойствам:

- **Репликация данных.** Для работы Redis Cluster используется асинхронная репликация — над значениями не выполняются операции слияния. Из-за асинхронной репликации система Redis Cluster не гарантирует, что все данные будут сохранены во время отказа.
- **Мониторинг.** Для выполнения своих задач все узлы кластера подключены с помощью шины TCP: каждый экземпляр подключен ко всем остальным узлам кластера с помощью шины кластера. Узлы используют gossip-протокол для распространения информации об изменении в кластере (обнаружении новых экземпляров, проверки связи с существующими экземплярами, отправки других сообщений кластера). Шина кластера также используется для отправки сообщений Pub/Sub по кластеру и организации ручной отработки отказа по запросу пользователей. Отказоустойчивость настроенной конфигурации должна регулярно проверяться и подтверждаться тестовыми отказами.
- **Автоматическое восстановление работоспособности (failover).** Redis Cluster может функционировать, когда часть master-экземпляров не работает как ожидается при условии, что для каждого из недоступных master-экземпляров есть по крайней мере один slave-экземпляр. Благодаря миграции реплик master-экземпляры, которые больше не реплицируются ни одним slave-экземпляром, получают новый slave-экземпляр от master-экземпляра, который обслуживается несколькими slave-экземплярами. При этом приложения Creatio, использующие Redis, переконфигурируют соединение согласно состоянию Redis Cluster. Для обеспечения отказоустойчивости необходимо не менее шести экземпляров Redis, запущенных на разных физических или виртуальных компьютерах.
- **Масштабирование.** Механизм Redis Cluster позволяет добавлять и удалять узлы во время работы кластера. Redis Cluster можно масштабировать до 1000 экземпляров благодаря автоматическому шардированию данных.

Системные требования Redis Cluster

Redis Cluster разворачивается на операционных системах Linux с Redis Server версии 4.0 и выше.

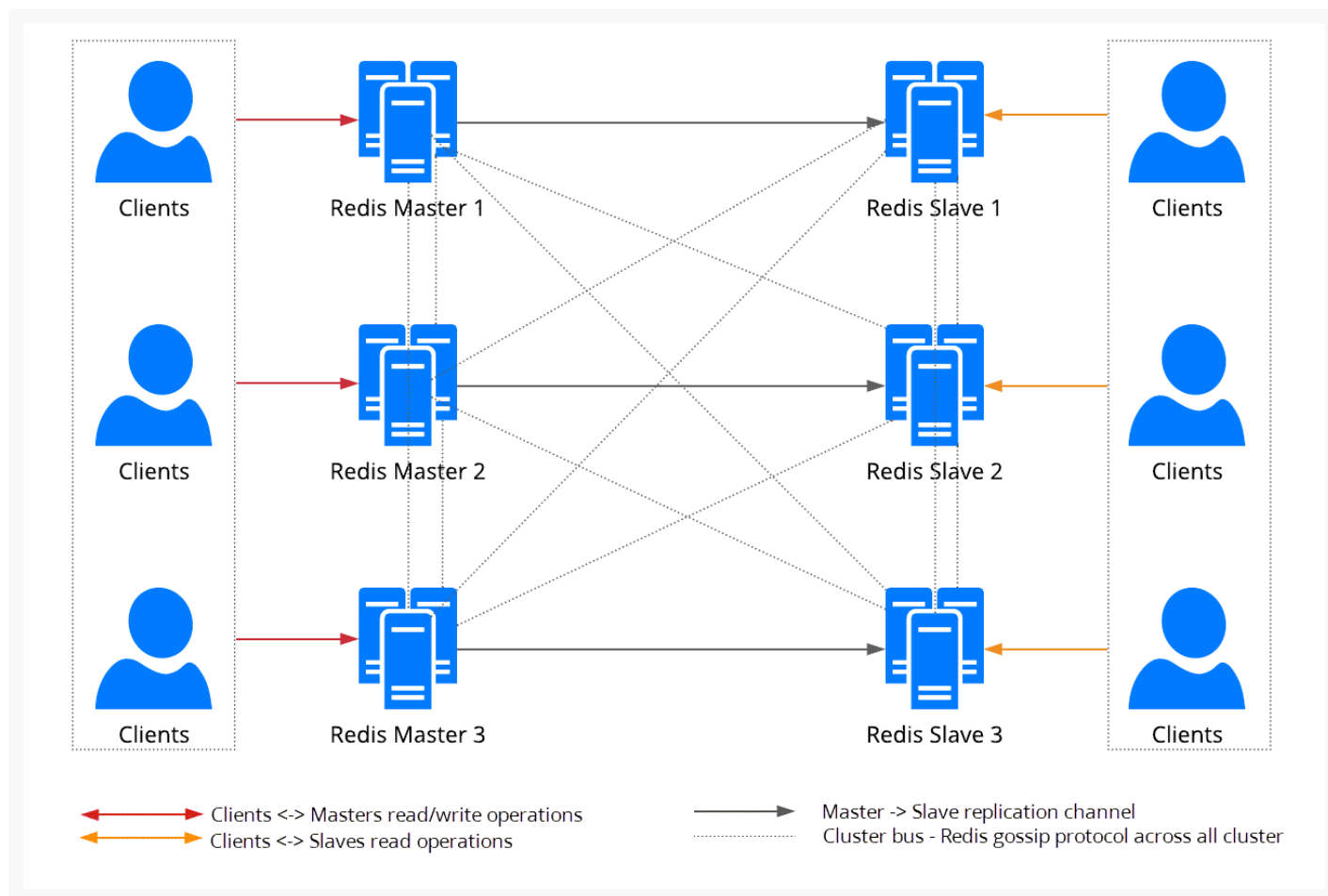
Redis является однопоточным приложением и нагружает только одно ядро процессора, поэтому для работы **одного экземпляра** Redis необходим узел (физический или виртуальный компьютер) с двухъядерным процессором, как минимум. Для обеспечения **максимальной отказоустойчивости** рекомендуется использовать количество физических машин, равное количеству экземпляров Redis, чтобы каждая связка master-slave была распределена на разные физические сервера. Для расчета требований к серверам воспользуйтесь [калькулятором системных требований](#).

Установка и настройка Redis Cluster

Минимальная отказоустойчивая конфигурация Redis Cluster

Рекомендуется использовать конфигурацию минимум с шестью экземплярами Redis (подробнее о конфигурации читайте в блоке "Creating and using a Redis Cluster" [документации Redis Cluster](#)). Эта конфигурация основана на шести узлах (физических или виртуальных компьютерах), каждый из которых содержит запущенные экземпляры Redis (Рис. 1).

Рис. 1 — Конфигурация Redis Cluster из шести узлов



В штатном режиме slave- экземпляры работают только на чтение. Данные на них асинхронно реплицируются из master-экземпляров. Master-экземпляры работают и на чтение, и на запись. В случае если на узел попадает команда с ключом, который находится в другом узле, то возвращается

информация о том, на каком узле необходимо выполнить операцию.

Если один из master-экземпляров Redis становится недоступным, то начинается процесс восстановления работоспособности (failover). В рамках этого процесса один из slave-экземпляров Redis становится новым master-экземпляром, с которым продолжает работать клиентское приложение.

Важно. В любой конфигурации, в которой данные реплицируются асинхронно, существует риск потери записей. Это возможно, поскольку данные могут не попасть на slave-экземпляр Redis, ставший новым master-экземпляром. Автоматическая переконфигурация, в рамках которой назначается новый master-экземпляр, может занять до 15 секунд.

Установка Redis Cluster

Redis Cluster поставляется вместе с дистрибутивом Redis. Для установки следует использовать новейшую версию Redis на дату релиза Creatio.

Процесс установки описан в [документации Redis](#). Пример настройки конфигурации Redis Cluster приведен в разделе “Creating and using a Redis Cluster”.

Для **мониторинга состояния** кластера при подключении к одному из узлов рекомендуется выполнять следующие проверки:

- **посмотреть конфигурацию кластера** с помощью команды `cluster nodes`.
- **проверить общую работоспособность** кластера при помощи redis-cli:

`redis-cli --cluster check ClusterIP` где “ClusterIP” — это IP-адрес одного из узлов кластера.

Настройка Creatio для работы с Redis Cluster

1. В файле `ConnectionStrings.config` необходимо указать адреса узлов Redis Cluster:

```
<add name="redis" connectionString="clusterHosts=ClusterIP1,ClusterIP2,ClusterIP3,ClusterIP4,
```

где ClusterIP1–ClusterIPn — это IP-адреса узлов кластера.

2. Убедитесь, что в вашем приложении включена функциональность `Feature-UseRetryRedisOperation`. Она запускает внутренний механизм Creatio для повтора операций с Redis, которые завершились с ошибкой. Эта проверка выполняется:

- a. Для приложений **Net Framework** (Windows) в файле `web.config`, который находится в корневой папке приложения;
- b. Для приложений **.NET Core** (Linux) в файле `Terrasoft.WebHost.dll.config`.

```
<add key="Feature-UseRetryRedisOperation" value="true" />
```

3. Убедитесь, что в секции **redis** файлов `web.config` (Net Framework) и `Terrasoft.WebHost.dll.config` (.NET

Core) используются рекомендуемые параметры:

- **enablePerformanceMonitor** — включает мониторинг времени выполнения операций с Redis. Рекомендуется включать для отладки и поиска проблем. По умолчанию выключен т. к. влияет на производительность приложения.
- **executionTimeLoggingThresholdSec** — операции с Redis, которые выполнялись дольше указанного времени, будут записаны в лог. По умолчанию 5 секунд.
- **clientConnectTimeoutMs** — время, выделяемое на установку сетевого соединения с сервером Redis. По умолчанию 5000 миллисекунд.
- **clientSyncTimeoutMs** — время, выделяемое на выполнение синхронных операций Redis. По умолчанию 5000 миллисекунд.
- **clientAsyncTimeoutMs** — время, выделяемое на выполнение асинхронных операций Redis. По умолчанию 5000 миллисекунд.
- **operationRetryIntervalMs** — если внутренний цикл повтора сбойных операций не привел к успешному выполнению операции, то такая операция откладывается на указанное время. После этого операция выполняется с новым клиентом, который уже может иметь установленное соединение с новым master-экземпляром. По умолчанию 5000 миллисекунд.
- **operationRetryCount** — количество повторных попыток выполнения операции с новым Redis-клиентом. По умолчанию 25.

Настройки должны иметь следующие значения:

```
<redis connectionStringName="redis" enablePerformanceMonitor="false" executionTimeLoggingThre
```

Развернуть базу данных PostgreSQL (Windows)

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Для установки Creatio вы можете использовать следующие конфигурации баз данных:

- удаленная СУБД (рекомендуется);
- локальный сервер PostgreSQL.

Если у вас уже настроен сервер PostgreSQL, то шаг I можно пропустить.

Если у вас уже есть пользователь с правами администратора для авторизации, создания и изменения баз данных, то шаг II можно пропустить.

I. Установить PostgreSQL

Поскольку PostgreSQL является свободной СУБД, то скачать файлы, необходимые для ее установки, можно [на официальном сайте продукта](#).

На заметку. Развертывание Creatio с отказоустойчивостью не тестировалось на PostgreSQL. Подробнее о возможности создания кластеров читайте в [документации PostgreSQL](#).

II. Создать пользователя PostgreSQL

Установленный PostgreSQL Server требует дополнительной настройки, прежде чем вы перейдете к развертыванию Creatio. После первичной установки PostgreSQL Server вам необходимо создать для него пользователя, который сможет подключаться к базе данных через логин и пароль и будет иметь права на создание и обновление баз данных. По умолчанию такого пользователя в PostgreSQL Server нет.

Рекомендуется создать следующих пользователей PostgreSQL:

- Пользователь с ролью **"sysadmin"** и неограниченными полномочиями на уровне сервера базы данных — нужен для восстановления базы данных и настройки доступа к ней. Далее для обозначения пользователя с ролью **"sysadmin"** будет использован псевдоним **pg_sysadmin**.
- Пользователь с ролью **"public"** и ограниченными полномочиями — используется для настройки безопасного подключения Creatio к базе данных через аутентификацию средствами PostgreSQL. Далее для обозначения пользователя с ролью **"public"** будет использован псевдоним **pg_user**.

Чтобы создать пользователей PostgreSQL:

1. Откройте командную строку.
2. Перейдите в папку с установочными файлами PostgreSQL:

```
cd /D "\\path\to\PostgreSQL\folder"
```

\\path\to\PostgreSQL\folder — путь к папке с установочными файлами PostgreSQL.

3. Перейдите в папку с компонентом Command Line Tools:

```
cd bin
```

4. Введите пароль подключения к серверу БД в переменную окружения:

```
set PGPASSWORD=pg_password
```

pg_password — пароль пользователя **postgres** для подключения к серверу PostgreSQL.

5. Запустите оболочку PostgreSQL от имени пользователя **postgres**:

```
psql.exe --username postgres
```

6. Создайте нового пользователя с правами администратора:

```
CREATE USER pg_sysadmin;
```

pg_sysadmin — пользователь, которому будут предоставлены права администратора. Нужен для восстановления базы данных и настройки доступа к ней.

7. Настройте для пользователя **pg_sysadmin** права администратора:

```
ALTER ROLE pg_sysadmin WITH SUPERUSER;
```

8. Разрешите **pg_sysadmin** авторизацию:

```
ALTER ROLE pg_sysadmin WITH LOGIN;
```

9. Настройте пароль для **pg_sysadmin**:

```
ALTER ROLE pg_sysadmin WITH PASSWORD 'pg_syspassword';
```

pg_password — пароль пользователя **pg_sysadmin** для подключения к серверу PostgreSQL.

10. Создайте нового пользователя с ограниченными правами:

```
CREATE USER pg_user;
```

pg_user — пользователь для подключения к серверу PostgreSQL. Используется для подключения Creatio к базе данных.

11. Разрешите **pg_user** авторизацию:

```
ALTER ROLE pg_user WITH LOGIN;
```

12. Настройте пароль для **pg_user**:

```
ALTER ROLE pg_user WITH PASSWORD 'pg_password';
```

pg_password — пароль пользователя **pg_user** для подключения к серверу PostgreSQL.

13. Выйдите из оболочки PostgreSQL.

```
\q
```

III. Восстановить из бэкапа базу данных PostgreSQL

Для восстановления базы данных из резервной копии необходимы утилиты **psql.exe** и **pg_restore.exe**. Обе утилиты входят в компонент Command Line Tools PostgreSQL, который устанавливается вместе с PostgreSQL Server. Они находятся в каталоге установленного программного обеспечения PostgreSQL.

Если вы планируете использовать удаленную базу данных PostgreSQL и не устанавливать PostgreSQL Server, то выполните следующие шаги:

1. Скачайте бинарный пакет PostgreSQL. Найти его можно [на официальном сайте продукта](#).
2. При установке пакета выберите также установку компонента Command Line Tools. Установка остальных компонентов опциональна.

Чтобы развернуть базу данных из бэкапа:

1. Откройте командную строку.
2. Перейдите в папку с установочными файлами PostgreSQL:

```
cd /D "\\path\to\PostgreSQL\folder"
```

\\path\to\PostgreSQL\folder — путь к папке с установочными файлами PostgreSQL.

3. Перейдите в папку с исполнимыми модулями:

```
cd bin
```

4. Установите пароль подключения к серверу БД в переменную окружения:

```
set PGPASSWORD=pg_syspassword
```

pg_syspassword — пароль пользователя **pg_sysadmin** для подключения к серверу PostgreSQL.

5. Создайте базу данных, в которую будет выполнено восстановление резервной копии:

```
psql.exe --host pg_server_address --port pg_server_port --username=pg_sysadmin --dbname=pg_db
```

pg_server_address — адрес сервера PostgreSQL;

pg_server_port — порт сервера PostgreSQL;

pg_sysadmin — пользователь, которому предоставлены права администратора.

pg_dbname — имя БД, в контексте которой происходит выполнение инструкций.

Если вы еще не создали базу данных или попытка подключения завершается ошибкой “FATAL: database “pg_dbname” does not exist”, то воспользуйтесь БД “template1”, которая создается в PostgreSQL по умолчанию.

pg_dbname_creatio — имя БД PostgreSQL, в которой будут храниться таблицы Creatio.

pg_user — пользователь с ограниченными правами, который используется для подключения к базе данных Creatio.

6. Если вы используете AWS RDS:

- a. Скачайте скрипт [ChangeTypesOwner.sql](#).
- b. Замените в скрипте значение "postgres" на актуальное имя пользователя Postgres.
- c. Выполните обновленный скрипт ChangeTypesOwner.sql.

7. Восстановите базу данных из бэкапа:

```
pg_restore --host pg_server_ip --port pg_server_port --username=pg_sysadmin --dbname=pg_dbname_creatio --n
```

pg_server_address — адрес сервера PostgreSQL;

pg_server_port — порт сервера PostgreSQL;

pg_sysadmin — пользователь, которому предоставлены права администратора.

pg_dbname_creatio — имя БД PostgreSQL, в которую будут добавлены таблицы резервной копии. Используйте имя базы данных, указанное в команде "CREATE DATABASE» на шаге 2.

8. Скачайте файл [CreateTypeCastsPostgreSql.sql](#).

9. Выполните преобразование типов:

```
psql.exe --host pg_server_ip --port pg_server_port --username=pg_sysadmin --dbname=pg_dbname_
```

```
psql.exe --host pg_server_ip --port pg_server_port --username=pg_sysadmin --dbname=pg_dbname_creatio -  
-file=\\путь\к\CreateTypeCastsPostgreSql.sql
```

pg_server_ip — адрес сервера PostgreSQL;

pg_server_port — порт сервера PostgreSQL;

pg_sysadmin — пользователь, которому предоставлены права администратора.

pg_dbname_creatio — имя БД, в контексте которой происходит выполнение инструкций.

\\путь\к\CreateTypeCastsPostgreSql.sql — путь к сохраненному ранее файлу CreateTypeCastsPostgreSql.sql.

10. В Creatio версии 7.16.3 и выше существует возможность при восстановлении из резервной копии заменить владельца БД и ее объектов на пользователя, не являющегося системным администратором (не Superuser). Вам потребуется скрипт ChangeDbObjectsOwner. Для Postgres версии 10 и ниже: [Скачать скрипт](#). Для Postgres 11 и выше: [Скачать скрипт](#).

Чтобы восстановить БД от имени пользователя, не являющегося администратором:

a. Замените владельца БД:

```
psql.exe --host pg_server_ip --port pg_server_port --username=pg_sysadmin --dbname=pg_dbname
```

pg_server_ip — адрес сервера PostgreSQL.

pg_server_port — порт сервера PostgreSQL.

pg_sysadmin — пользователь для подключения к серверу PostgreSQL. Пользователь должен являться администратором (Superuser) или иметь права на создание базы данных (команда "CREATE DATABASE").

pg_user — новый владелец БД.

pg_dbname_creatio — имя БД, владелец которой заменяется.

b. Замените владельца объектов БД:

```
psql.exe --host pg_server_ip --port pg_server_port --username=pg_sysadmin --dbname=pg_dbname
```

pg_server_ip — адрес сервера PostgreSQL.

pg_server_port — порт сервера PostgreSQL.

pg_sysadmin — пользователь для подключения к серверу PostgreSQL. Пользователь должен являться администратором (Superuser) или иметь права на создание базы данных (команда "CREATE DATABASE").

pg_user — новый владелец БД.

pg_dbname_creatio — имя БД, владелец которой заменяется;

\\путь\к\ChangeDbObjectsOwner.sql — путь к сохраненному ранее файлу ChangeDbObjectsOwner.sql.

Вы можете пропустить этот шаг, и в таком случае владельцем БД и ее объектов останется пользователь, от имени которого запускалась команда pg_restore. Обычно это пользователь postgres.

Настроить ConnectionStrings.config для PostgreSQL

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

В файле ConnectionStrings.config, который находится в корневой папке сайта Creatio, содержатся параметры подключения вашего приложения к базе данных и внешним сервисам.

Настроить ConnectionStrings.config

1. Перейдите в корневой каталог сайта Creatio **~\WebAppRoot\Creatio**.
2. Откройте в текстовом редакторе файл ConnectionStrings.config.
3. Укажите параметры подключения (**connectionStrings**).

Пример настроенного файла ConnectionStrings.config

```
<?xml version="1.0" encoding="utf-8"?>
<connectionStrings>
```

```
<add name="db" connectionString="Server=[ Имя сервера базы данных ];Port=[ Порт сервера базы да
<add name="redis" connectionString="host=[ Имя компьютера ];db=[ Номер БД Redis ];port=6379;maxR
<add name="redisSentinel" connectionString="sentinelHosts=localhost:26380,localhost:26381,localh
<add name="defPackagesWorkingCopyPath" connectionString="%TEMP%\%APPLICATION%\%APPPPOOLIDENTITY%\
<add name="tempDirectoryPath" connectionString="%TEMP%\%APPLICATION%\%APPPPOOLIDENTITY%\%WORKSPAC
<add name="sourceControlAuthPath" connectionString="%TEMP%\%APPLICATION%\%APPPPOOLIDENTITY%\%WORK
<add name="elasticsearchCredentials" connectionString="User=[ Имя пользователя ElasticSearch ];
<add name="influx" connectionString="url=http://10.0.7.161:30359; user=; password=; batchInterva
</connectionStrings>
```

Обязательные настройки ConnectionStrings.config

Параметры подключения к базе данных и серверу кэширования необходимы для работы приложения Creatio.

1. К восстановленной базе данных (**name="db"**):

```
<add name="db" connectionString="Server=[ Имя сервера базы данных ];Port=[ Порт сервера базы
```

2. К Redis Server (**name="redis"**):

```
<add name="redis" connectionString="host=[ Имя компьютера ];db=[ Номер БД Redis ];port=6379;m
```

Необязательные настройки ConnectionStrings.config

Параметры подключения к внешним сервисам опциональны, они заполняются только в том случае, когда необходимо, например, при подключении системы контроля версий.

- **tempDirectoryPath** — путь к временной директории, используемой механизмом установки пакетов:

```
<add name="tempDirectoryPath" connectionString=[ Путь к временной директории, используемой ме
```

- **defPackagesWorkingCopyPath** — путь к рабочей копии пользовательских пакетов приложения (только при использовании системы контроля версий SVN). Содержимое рабочей копии представляет собой пользовательские пакеты в виде набора папок и файлов. Оно синхронизируется с репозиторием системы контроля версий SVN при помощи встроенного в Creatio клиента SVN. Данный параметр настраивается при подключении системы контроля версий и используется только в режиме разработки по умолчанию, без включения режима разработки в файловой системе. Значение по умолчанию указывает на временный каталог, который может быть очищен операционной системой. Рекомендуется установить значение, указывающее на пользовательский каталог, так как использование каталогов приложения, например, .\Terrasoft.WebApp\Terrasoft.Configuration\Pkg, может привести к ошибкам компиляции.

```
<add name="defPackagesWorkingCopyPath" connectionString=[ Путь к рабочей копии пользовательск
```

- **sourceControlAuthPath** — путь к данным авторизации для встроенного клиента системы контроля версий SVN (если используется). Значение по умолчанию указывает на временный каталог, который может быть очищен операционной системой. При использовании системы контроля версий рекомендуем указать в данном параметре путь к фиксированному каталогу.

```
<add name="sourceControlAuthPath" connectionString="[ Путь к данным авторизации системы хранения ]"/>
```

- **influx** — отвечает за взаимодействие с сервисом сбора метрик сайта. Данный параметр заполняется только в том случае, когда необходимо собрать статистику использования функциональности для последующей отладки.

```
<add name="influx" connectionString="url=[ Адрес сервиса сбора метрик сайта ]; user=[ Пользователь ]"/>
```

- **clientPerformanceLoggerServiceUri** — отвечает за взаимодействие с сервисом логгирования. Данный параметр заполняется только в том случае, когда необходимо собрать информацию о загрузке страниц приложения.

```
<add name="clientPerformanceLoggerServiceUri" connectionString="[ Адрес сервиса логгирования ]"/>
```

- **messageBroker** — отвечает за взаимодействие с сервисом RabbitMQ. Данный параметр заполняется только в том случае, когда необходимо настроить горизонтальное масштабирование нагрузки с помощью RabbitMQ.

```
<add name="messageBroker" connectionString="amqp://[ Имя пользователя MessageBroker ]:[ Пароль ]@localhost:5672/"/>
```

Сервис машинного обучения

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Сервис машинного обучения предназначен для прогнозирования значений на основании больших объемов исторических данных и текущих фактов. Подробнее: [Предиктивный анализ данных](#).

Важно. Для настройки сервиса машинного обучения необходимы базовые знания администрирования Docker, Linux или Windows.

Подготовка к установке сервиса машинного обучения

На заметку. Для корректной работы сервиса в приложении on-site требуется [лицензия](#) “creatio

predictive service on-site”. По вопросам ее приобретения необходимо обратиться к ответственному менеджеру.

Для настройки сервиса необходим сервер (физический или виртуальный компьютер) с установленной ОС Linux или Windows.

Важно. Для промышленной среды рекомендуется использовать сервер с ОС Linux. Сервер на базе Windows можно использовать только для среды разработки. Для получения Docker-контейнеров, предназначенных для Windows, обратитесь в службу поддержки.

Установка компонентов сервиса выполняется с помощью ПО Docker. Для этого скачайте архив с конфигурационными файлами и скриптами установки. [Скачать архив](#). В зависимости от потребностей вашей компании можно использовать Docker Community Edition (CE) или Enterprise Edition (EE). Подробную информацию вы найдете в [документации Docker](#).

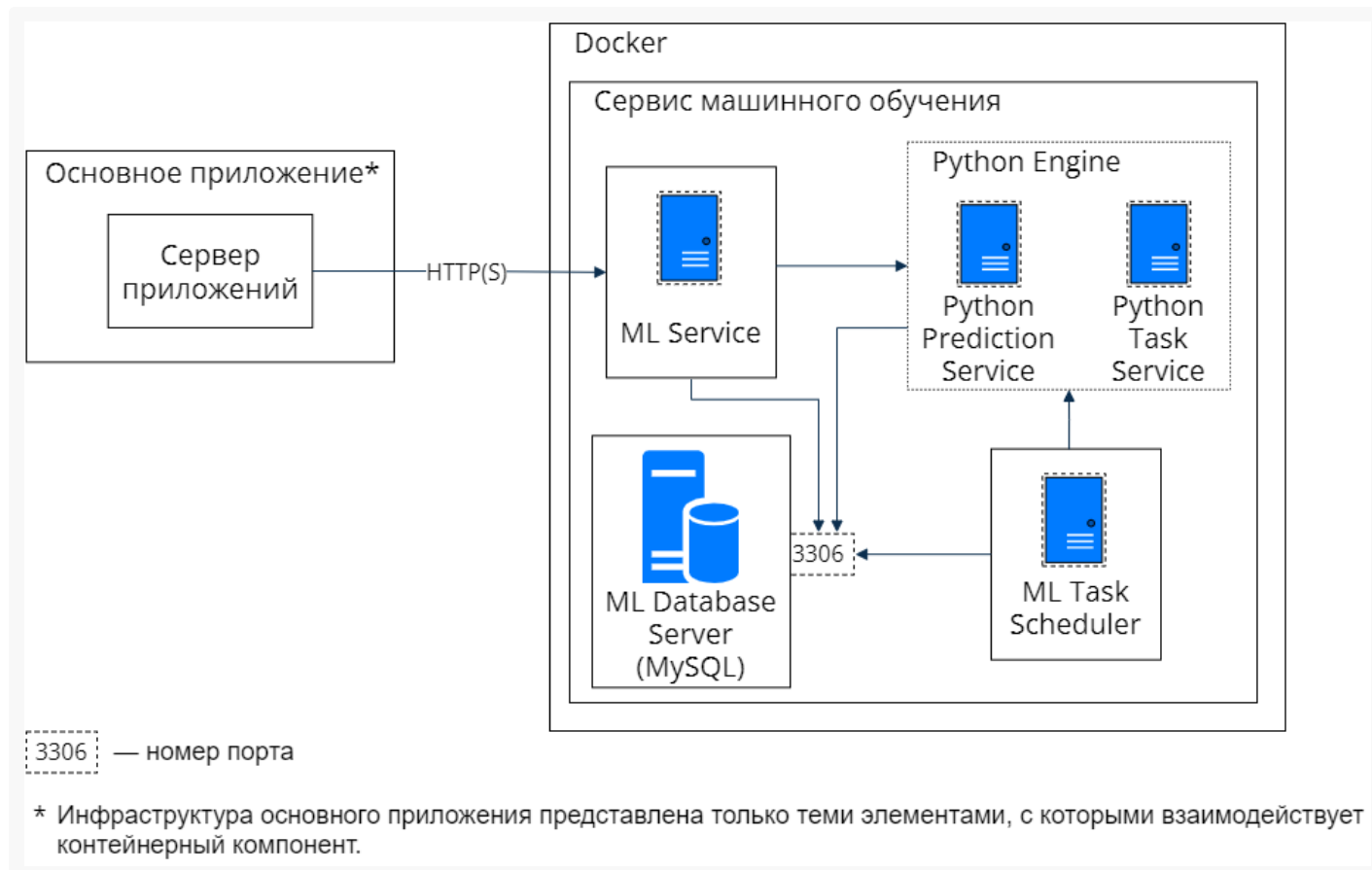
Для расчета требований к серверам воспользуйтесь [калькулятором системных требований](#).

Компоненты сервиса машинного обучения

Сервис машинного обучения использует следующие компоненты (Рис. 1):

- **ML Service** — web-сервис машинного обучения. Единственный компонент, доступный извне.
- **Python Engine** — движок машинного обучения, представляющий собой сервисную оболочку библиотек машинного обучения с открытым исходным кодом.
- **ML Task Scheduler** — планировщик задач.
- **MySQL** — база данных MySQL. Доступна по стандартному порту 3306.

Рис. 1 — Компоненты сервиса машинного обучения



Для удобства установки сервиса on-site все компоненты упакованы в Docker-образы.

Установить сервис машинного обучения

Последовательность действий для установки сервиса машинного обучения:

1. Установить Docker. [Подробнее >>>](#)
2. Установить Docker-Compose. [Подробнее >>>](#)
3. Установить и настроить компоненты сервиса. [Подробнее >>>](#)
4. Выполнить проверку установки. [Подробнее >>>](#)

Установить Docker

Для установки Docker на операционную систему Linux воспользуйтесь инструкцией в [документации Docker](#).

Для проверки установленной версии Docker выполните команду `docker --version` в терминале (консоли) Linux-машины.

Установить Docker-Compose

Для установки Docker на операционную систему Linux воспользуйтесь инструкцией в [документации Docker](#).

Для проверки установленной версии Docker Compose выполните команду `docker-compose --version` в терминале (консоли) Linux-машины.

Настроить компоненты сервиса машинного обучения

Все контейнеры компонентов сервиса машинного обучения развертываются с помощью утилиты Docker Compose. Для этого скачайте конфигурационные файлы и скрипты, необходимые для развертывания и настройки компонентов сервиса. [Скачать архив](#)

На заметку. Конфигурационные файлы содержат все необходимые настройки по умолчанию для сервера на базе Linux.

Структура архива с конфигурационными файлами и скриптами:

/etc/

...\ml-service\appsettings.json — конфигурация web-сервиса.

...\ml-service\log4net.config — настройка уровня логирования web-сервиса.

...\task-scheduler\appsettings.json — конфигурация утилиты ML task scheduler.

...\task-scheduler\log4net.config — настройка уровня логирования ML Task Scheduler.

docker-compose.yml — конфигурация утилиты Docker Compose.

.env — файл с переменными окружения для запуска компонентов. Например, содержит пароль для MySQL.

Важно. Если необходимо изменить пароль к базе данных MySQL, то его нужно обновить не только в файле .env, но и в других конфигурационных файлах, содержащих секции настройки доступа к базе данных.

Установить компоненты сервиса машинного обучения

1. Скачайте и разархивируйте содержимое архива с конфигурационными файлами и скриптами в произвольный каталог, например, `/opt/ml`.
2. Используя терминал Linux, перейдите в каталог `docker-compose` распакованного архива, например, `/opt/ml/docker-compose`.
3. Выполните команду `sudo docker-compose pull` в терминале. Дождитесь завершения загрузки необходимых образов с компонентами сервиса с [Docker Hub](#).

Важно. Если на сервере запрещен доступ в интернет, скачайте на машине с открытым доступом все необходимые образы вручную (см.конфигурационный файл `docker-compose.yml`). Затем воспользуйтесь командами [sudo docker export](#) и [sudo docker import](#) для переноса образов в виде файлов на целевую машину.

4. Выполните команду **`sudo docker-compose run dbmigration`** для инициализации структуры базы данных. Дождитесь завершения выполнения команды.

5. Выполните команду **sudo docker-compose up -d** для запуска сервисов. Также в текущем каталоге будет создана папка logs.

Проверить установку компонентов сервиса машинного обучения

1. Для проверки установки веб-сервиса ML service выполните в терминале Linux команду:

```
curl -X GET localhost:5005/readiness
```

Сервис должен вернуть ответ:

Healthy

2. Для проверки запуска планировщика ML Task Scheduler выполните в терминале Linux команду:

```
curl -X GET localhost:5004/readiness
```

Сервис должен вернуть ответ:

Healthy

3. Для проверки запуска R Engine выполните в терминале Linux команду:

```
curl -X GET localhost:8081/readiness
```

Сервис должен вернуть ответ:

R Service is ready

4. Для проверки создания таблиц выполните в терминале команду:

```
docker exec -it DB Container Id mysql -u root --password=Supervisor ml -e "show tables;"
```

Где [*DB Container Id*] — идентификатор контейнера с компонентом базы данных. Узнать идентификатор контейнера можно с помощью команды **sudo docker ps**.

Пример. Проверка создания таблиц:

```
docker exec -it 7c9de07b53e8 mysql -u root --password=Supervisor ml -e "show tables;"
```

В результате должны отобразиться названия основных таблиц сервиса: “modelinstance”, “traindata”, “trainsession” и т. д.

Заполнить настройки Creatio

Для работы с сервисом прогнозирования в Creatio необходимо заполнить:

1. Системную настройку “API ключ облачных сервисов Creatio” (код “CloudServicesAPIKey”) — отвечает за аутентификацию экземпляра Creatio в облачных сервисах.
2. Системную настройку “Периодичность запуска задач машинного обучения” (код “MLModelTrainingPeriodMinutes”) — определяет периодичность запуска синхронизации моделей.
3. Адрес развернутого сервиса прогнозирования в поле [*ServiceUrl*] для всех записей в справочнике [*Задачи машинного обучения*].

Обновить компоненты сервиса машинного обучения

Важно. Перед обновлением сервисов рекомендуется сохранить резервную копию базы данных MySQL. Подробную информацию вы найдете в [документации MySQL](#).

1. Используя терминал Linux, перейдите в каталог `docker-compose` с настроечными файлами, например, `/opt/ml/docker-compose`.
2. Выполните команду **`sudo docker-compose stop`** для остановки контейнеров компонентов сервиса.
3. Выполните команду **`sudo docker-compose pull`** в терминале. Дождитесь завершения загрузки необходимых образов с компонентами сервиса с [Docker Hub](#).
4. Выполните команду **`sudo docker-compose run dbmigration`** для инициализации структуры базы данных. Дождитесь завершения выполнения команды.
5. Выполните команду **`sudo docker-compose up -d`** для запуска сервисов.

Важно. Если у вас в приложении уже есть настроенные и обученные модели, после обновления сервиса рекомендуем их переобучить.

Настроить сервис обогащения данных

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Для использования функциональности обогащения данных в Creatio должен быть указан ваш персональный ключ облачных сервисов, а также URL подключения к облачным сервисам Creatio. Для этого используются системные настройки:

- “Адрес сервиса обогащения контрагентов”. По умолчанию эта настройка заполнена для всех приложений.
- “API-ключ облачных сервисов Creatio”. Для приложений **cloud** по умолчанию настройка заполнена. Настройку нужно выполнить для приложений, развернутых **on-site**.

Для приложений, развернутых **on-site**, запросите персональный ключ в службе поддержки. После получения ключа:

1. Из дизайнера системы перейдите в раздел [*Системные настройки*].
2. В группе “Creatio cloud services” откройте системную настройку “API-ключ облачных сервисов Creatio”.
3. В поле [*Значение по умолчанию*] введите полученный ключ и сохраните настройку.

Теперь вы можете использовать функциональность обогащения данных.

Настроить Web.config

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Необходимо сгенерировать для вашего приложения уникальное значение `machineKey`.

1. Скачайте powershell-скрипт. [Скачать скрипт](#).
2. Запустите powershell-терминал от имени администратора.
3. В powershell-терминале укажите путь к корневой папке приложения и выполните скрипт. Пример запуска скрипта:

```
.\UpdateMachineKey.ps1 "путь\к\корневой\папке\Creatio"
```

В результате в файлах Web.config, которые находятся в корневой папке приложения и в папке Terrasoft.WebApp, будет сгенерировано уникальное значение machineKey.

Дополнительная настройка Web.config (только для Oracle)

После настройки параметров соединения с базой данных в файле ConnectionStrings.config для Oracle Database также необходимо настроить конфигурационный файл Web.config, в который вынесены некоторые параметры конфигурации приложения, необходимые для его работы.

Чтобы настроить Web.config:

1. Перейдите в корневую папку приложения **~\WebAppRoot\Creatio**.
2. Откройте файл Web.config в текстовом редакторе.
3. В параметре general блока <db> измените значение атрибута currentSchemaName на имя схемы, которая указана в ConnectionStrings.config. Это позволит сайту приложения увидеть записанные в файле параметры конфигурации.

```
<db>
  <general securityEngineType="Terrasoft.DB.Oracle.OracleSecurityEngine, Terrasoft.DB.Oracle"
  executorType="Terrasoft.DB.Oracle.OracleExecutor, Terrasoft.DB.Oracle"
  isCaseInsensitive="true" maxAnsiJoinCount="0"
  engineType="Terrasoft.DB.Oracle.OracleEngine, Terrasoft.DB.Oracle"
  metaEngineType="Terrasoft.DB.Oracle.OracleMetaEngine,
  Terrasoft.DB.Oracle"
  metaScriptType="Terrasoft.DB.Oracle.OracleMetaScript,
  Terrasoft.DB.Oracle" typeConverterType="Terrasoft.DB.Oracle.OracleTypeConverter,
  Terrasoft.DB.Oracle" connectionStringName="db"
  binaryPackageSize="1048576" useOrderNullsPosition="true"
  currentSchemaName="[ Имя схемы базы данных Oracle ]" maxEntitySchemaNameLength="30" />
</db>
```

4. Проверьте, что в блоке <DbProviderFactories> указана версия библиотеки 4.112.1.2. Этот параметр определяет коннектор, через который происходит подключение сайта к базе данных.

```
<system.data>
  <DbProviderFactories>
    <remove invariant="Oracle.DataAccess.Client" />
    <add name="Oracle Data Provider for .NET"
```

```
invariant="Oracle.DataAccess.Client" description="Oracle Data
Provider for .NET" type="Oracle.DataAccess.Client.OracleClientFactory, Oracle.DataAccess,
Version=4.112.1.2, Culture=neutral, PublicKeyToken=89b483f429c47342"/>
</DbProviderFactories>
</system.data>
```

5. В блоке <quartz> укажите значение ключа **quartz.dataSource.SchedulerDb.provider**. Этот параметр необходим для того, чтобы планировщик мог взаимодействовать с таблицами в базе данных.

```
<add key="quartz.dataSource.SchedulerDb.provider" value="OracleODP-1123-40" />
```

6. Сохраните изменения.

Настроить сервис синхронизации Exchange Listener

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Сервис синхронизации Exchange Listener реализует синхронизацию Creatio с почтовыми сервисами [MS Exchange](#) и [IMAP/SMTP](#), используя механизм подписки.

В данной статье описаны системные требования, а также схемы и процесс развертывания сервиса синхронизации для Creatio on-site.

Сервис состоит из следующих компонентов:

- **Основной модуль Exchange Listener (EL API)** — используется для создания исходящего подключения к EWS API. При этом используются учетные данные почтового ящика и создается подписка (subscription) для получения событий при поступлении новых писем. Открытая подписка остается в памяти компонента для обеспечения максимально оперативной реакции на получение нового письма. При получении соответствующего события выполняется загрузка экземпляра письма. Для развертывания сервиса **достаточным условием** является использование in-memory хранилища. Обязательный компонент сервиса.
- **NoSQL СУБД Redis** — используется для создания масштабируемой и отказоустойчивой системы узлов-обработчиков. Хранилище Redis содержит информацию об обслуживаемых почтовых ящиках. Это позволяет любому контейнеру обработать запросы Creatio на создание новой подписки или проверить статус конкретной подписки, независимо от того, на каком узле она открыта. **Необходимым требованием** к Redis является выделение отдельной базы данных для работы сервиса Exchange Listener. Обязательный компонент сервиса.
- **Дополнительный модуль Exchange Listener (EL Worker)** — используется для поддержания масштабируемой и отказоустойчивой работы основного модуля Exchange Listener. Дополнительный модуль выполняет скачивание писем с почтового сервера и доставку их в приложение Creatio. Таким образом сглаживается обработка пиковых потоков писем для высоконагруженных сервисов. В этом случае API компоненты менее загружены и не занимаются скачиванием писем, а доступны для управления подпиской и отправкой исходящих писем. Опциональный компонент сервиса. При его отсутствии эту роль выполняет основной модуль.

- **RabbitMQ** — используется для поддержания масштабируемой и отказоустойчивой работы сервиса. Брокер очередей выполняет распределение задач между компонентами в высоконагруженных средах. Опциональный компонент сервиса.

Существуют две возможные конфигурации сервиса:

- **Однокомпонентная** — для слабонагруженных сервисов. Данная конфигурация включает только обязательные компоненты сервиса.
- **Многокомпонентная** — для средненагруженных и высоконагруженных сервисов. Данная конфигурация включает все обязательные и опциональные компоненты.

Определить конфигурацию сервиса Exchange Listener

Конфигурация сервиса Exchange Listener для вашего приложения определяется на основании среднего потока писем в секунду (исходящих и входящих), который проходит через почтовые ящики компании.

Например, если в компании используется один почтовый ящик службы поддержки с потоком 4 письма в секунду, то рекомендуемая конфигурация будет включать 15 реплик EL Worker, 4 реплики EL API и сервис RabbitMQ.

Количество почтовых ящиков	Средний поток писем в секунду	Количество реплик EL Worker	Количество реплик EL API	Наличие RabbitMQ
1 – 150	< 1	0	4	Опционально
	1 – 5	15		Обязательно
	> 5	25		
> 150	< 1	2	1 реплика на каждые 30 почтовых ящиков	Обязательно
	1 – 5	15		
	> 5	25		

На заметку. Количество активных реплик дополнительного компонента сервиса (EL Worker) напрямую влияет на скорость обработки писем. В продуктивном использовании письма приходят не равномерным потоком, а волнообразно, поэтому часть времени некоторые реплики EL Worker могут простаивать. В статье приведены рекомендуемые параметры конфигурации, но вы можете использовать меньше реплик, чем указано в таблице. В этом случае в период высокой нагрузки письма будут обрабатываться дольше. Оптимальный баланс между скоростью обработки писем и утилизацией ресурсов следует подбирать согласно требованиям бизнеса.

Системные требования для реплик компонентов

Компонент	vCPU	Объем памяти
EL Worker	0,1	1,1 Гб
EL API	0.150 (с RabbitMQ), 0.7 (без RabbitMQ)	850 Мб (с RabbitMQ) 1.1 Гб (без RabbitMQ)
Redis	0,5	3 Гб
Rabbit MQ	0,5	4 Гб

На заметку. Значения в таблице являются рекомендуемыми, реальное потребление может отличаться в зависимости от сценариев использования сервиса. Рекомендуется выполнять мониторинг потребления CPU и памяти на развернутых сервисах для оптимизации доступных лимитов.

Способы развертывания сервиса Exchange Listener

Для развертывания сервиса предпочтительным способом является **использование оркестратора Kubernetes и пакетного менеджера Helm**. [Подробнее >>>](#)

Для более быстрого развертывания в среде разработки можно **использовать Docker**. [Подробнее >>>](#)

Развернуть сервис синхронизации с использованием Kubernetes

Для развертывания сервиса выполните следующие шаги:

- Предварительно настройте целевое окружение:
 - Кластер Kubernetes. Подробно о том, как настроить и администрировать кластер, читайте [на сайте документации Kubernetes](#).
 - Пакетный менеджер Helm. Установка пакетного менеджера подробно описана [на сайте документации Helm](#).
- Установите Redis. Установка Redis с использованием Helm детально описана [на сайте GitHub](#).

Пример команды для установки Redis:

```
helm install --namespace default --set usePassword=true --set password=<password> --set=slave.
```

В этом примере:

default — наименование namespace, куда будет установлен Redis;

redis — произвольное имя для экземпляра Redis.

- Установите модуль Exchange Listener. Для установки модуля [скачайте helm-пакет](#). Доступные параметры helm-пакета описаны в таблице ниже.

Важно. Для более новых версий Kubernetes укажите версию API, добавив параметр:

```
--set apiVersion=apps/v1
```

Пример команды для установки Exchange Listener с использованием адреса и относительного пути сервиса:

```
helm install --set ApiUrl=<kubernetes_url> --set ingress.enabled=true --set ingress.path=<lis
```

В этом примере:

<redis_host> — адрес Redis-сервера;

<kubernetes_url> — URL или IP-адрес Kubernetes.

Адрес сервиса Exchange Listener: **<kubernetes_url>/<listener_path>**.

Для настройки подключения с использованием HTTPS необходимо установить сервис с [Ingress](#) и действительным SSL сертификатом, а в адресе сервиса Exchange Listener **<kubernetes_url>** указать HTTPS.

Чтобы проверить доступность, сделайте запрос по адресу:

<kubernetes_url>/<listener_path>/api/listeners/status (Рис. 1).

Пример команды для установки Exchange Listener с использованием Node IP и адреса порта:

```
helm install --set env.host=<redis_host> --set service.type=<node_IP> --set service.nodePort=
```

Адрес сервиса Exchange Listener — **<node_IP:node_port>**.

Чтобы проверить доступность, сделайте запрос по адресу: **<node_IP:node_port>/api/listeners/status** (Рис. 1).

Рис. 1 — Пример ответа сервиса Exchange Listener

```
{
  "ServiceStatus": "Started",
  "version": "0.5.0",
  "connections": {
    "657b3ea8-477f-419c-a07a-4d4cc2158fc5": {
      "SenderEmailAddress": " ",
      "BpmUser": " ",
      "BpmEndpoint": "https:// /0/ServiceModel/ExchangeListenerService.svc/NewEmail",
      "State": "exists",
      "Id": "657b3ea8-477f-419c-a07a-4d4cc2158fc5",
      "RedisKey": "Subscription_657b3ea8-477f-419c-a07a-4d4cc2158fc5_exchangelistener-api-2",
      "UseFullEmail": true
    },
    "332bcae2-0530-4636-bc09-50a594389f53": {
      "SenderEmailAddress": " ",
      "BpmUser": " ",
      "BpmEndpoint": "https:// /0/ServiceModel/ExchangeListenerService.svc/NewEmail",
      "State": "exists",
      "Id": "332bcae2-0530-4636-bc09-50a594389f53",
      "RedisKey": "Subscription_332bcae2-0530-4636-bc09-50a594389f53_exchangelistener-api-1",
      "UseFullEmail": true
    }
  },
}
```

На заметку. Если вам нужно отключить маркировку загружаемых писем категорией “Creatio”, то при установке приложения в команде `helm install` нужно указать параметр `--set FeaturesMarkEmailsAsSynchronized=false`.

Доступные параметры helm-пакета Exchange Listener

Параметр	Описание параметра	Значение по умолчанию
replicaCount	Количество StatefulSet-обработчиков.	2
service.type	Тип сервиса. Детально типы сервисов Kubernetes описаны в документации Kubernetes .	ClusterIP
service.nodePort	Если параметр service.type равен NodePort, то в этом параметре указывается внешний порт сервиса. Детально тип NodePort описан в документации Kubernetes .	
env.host	Адрес хоста Redis.	
env.port	Порт хоста Redis.	6379
env.base	Номер базы данных Redis.	0
ingress.enabled	Использование переопределения адресов при помощи ingress.	false
ApiUrl	Адрес сервиса при ingress.enabled=true.	
ingress.path	Относительный путь сервиса.	
log4Net.level	Уровень логирования по умолчанию.	Info

Для расчета требований к серверам воспользуйтесь [калькулятором системных требований](#).

Развернуть сервис синхронизации в Docker

Для настройки сервиса необходим сервер (физический или виртуальный компьютер) с установленной ОС Linux или Windows.

Важно. Развертывание сервиса синхронизации в Docker рекомендуется к установке только для dev-среды. Данный способ отличается высокой скоростью развертывания, но он не обеспечивает выполнение требований, актуальных для продуктивной среды: отказоустойчивости функции, масштабирования для обработки большого количества запросов и единого подхода к управлению компонентами с использованием систем оркестрации контейнеров. Для продуктивных сред настоятельно рекомендуется установка с использованием оркестратора Kubernetes и пакетного менеджера Helm.

Для развертывания сервиса выполните следующие шаги:

1. Предварительно настройте целевое окружение:

- a. Платформа контейнеризации Docker. Подробно о том, как установить и настроить платформу, описано [на сайте документации Docker](#). Для установки Docker Desktop на Windows Server необходимо использовать [отдельную инструкцию](#) на сайте Microsoft.
 - b. СУБД Redis. Redis Server устанавливается при помощи [установочного файла для ОС Windows](#) или при помощи [инструкции для ОС Linux](#). Подробнее о развертывании СУБД Redis в Docker читайте на сайте сообщества [Docker Hub](#).
2. Установите и запустите модуль Exchange Listener. Для этого необходимо скачать и развернуть образ Docker-контейнера.

Рассмотрим пример команды для скачивания и запуска образа с использованием командной строки и установленного Docker.

```
docker run \
-d \
# Проброс портов
-p <localhost_port>:80 \
--restart unless-stopped \
# Подключение к Redis
--env ExchangeListenerRedisHost=<redis_host>:<redis_port> \
--env ExchangeListenerRedisDatabase=<redis_database_number> \
--env PodName=ExchangeListener \
--name ExchangeListener \
# Актуальный образ ExchangeListener в Docker Hub
bpmonline/exchangelistener:<listener_version>
```

В этом примере:

<localhost_port> — порт локального сервера;

<redis_host> — адрес redis-сервера;

<redis_database_number> — номер БД redis-сервера;

<service_name> — название сервиса (задается вручную).

<listener_version> — версия микросервиса Listener.

На заметку. Актуальную версию Exchange Listener можно узнать в сообществе Docker Hub по [ссылке](#).

3. Чтобы проверить доступность развернутого Docker-контейнера, выполните команду:

```
docker ps -a --filter "name=<service_name>"
```

Адрес сервиса Exchange Listener — **localhost:<localhost_port>**.

Чтобы проверить доступность, сделайте запрос по адресу: <Localhost:<localhost_port>/api/listeners/status (Рис. 2).

Настроить сервис Exchange Listener на стороне Creatio

1. Убедитесь, что анонимный сервис ExchangeListenerService доступен по адресу [Адрес приложения Creatio]/0/ServiceModel/ExchangeListenerService.svc (Рис. 2).

Рис. 2 — Пример ответа сервиса ExchangeListenerService

Service

This is a Windows® Communication Foundation service.

Metadata publishing for this service is currently disabled.

If you have access to the service, you can enable metadata publishing by completing the following steps to modify your web or application configuration file:

1. Create the following service behavior configuration, or add the <serviceMetadata> element to an existing service behavior configuration:

```
<behaviors>
  <serviceBehaviors>
    <behavior name="MyServiceTypeBehaviors" >
      <serviceMetadata httpGetEnabled="true" />
    </behavior>
  </serviceBehaviors>
</behaviors>
```

2. Add the behavior configuration to the service:

```
<service name="MyNamespace.MyServiceType" behaviorConfiguration="MyServiceTypeBehaviors" >
```

Note: the service name must match the configuration name for the service implementation.

3. Add the following endpoint to your service configuration:

```
<endpoint contract="IMetadataExchange" binding="mexHttpBinding" address="mex" />
```

Note: your service must have an http base address to add this endpoint.

The following is an example service configuration file with metadata publishing enabled:


```
<configuration>
  <system.serviceModel>

    <services>
      <!-- Note: the service name must match the configuration name for the service implementation. -->
      <service name="MyNamespace.MyServiceType" behaviorConfiguration="MyServiceTypeBehaviors" >
        <!-- Add the following endpoint. -->
        <!-- Note: your service must have an http base address to add this endpoint. -->
        <endpoint contract="IMetadataExchange" binding="mexHttpBinding" address="mex" />
      </service>
    </services>

    <behaviors>
      <serviceBehaviors>
        <behavior name="MyServiceTypeBehaviors" >
          <!-- Add the following element to your service behavior configuration. -->
          <serviceMetadata httpGetEnabled="true" />
        </behavior>
      </serviceBehaviors>
    </behaviors>

  </system.serviceModel>
</configuration>
```

For more information on publishing metadata please see the following documentation: <http://go.microsoft.com/fwlink/?LinkId=65455>.

2. Установите нужные значения системных настроек. Для этого:
 - a. Перейдите в дизайнер системы, например, по кнопке .
 - b. В блоке “Настройка системы” перейдите по ссылке “Системные настройки”.
 - c. Укажите значения системных настроек:

“ExchangeListenerServiceUri”(код “ExchangeListenerServiceUri”). Формат значения настройки:

[*адрес сервиса, используемый при установке*]/api/listeners.

“URL сервиса обработки событий Exchange в Creatio” (код “BpmonlineExchangeEventsEndpointUrl”). Формат значения настройки: [*адрес анонимного сервиса ExchangeListenerService*]/NewEmail. Например, <https://mycreatio.com/0/ServiceModel/ExchangeListenerService.svc/NewEmail>.

Диагностика настроек Exchange Listener

На странице диагностики настроек сервиса Exchange Listener представлены инструменты контроля и проверки работы сервиса.

Здесь вы можете:

- проверить подключение необходимой функциональности;
- убедиться в доступности сервиса;
- получить информацию о подписках;
- проверить правильность заполнения системной настройки “ExchangeListenerServiceUri”;
- проверить работоспособность почтового ящика.

Чтобы перейти на страницу диагностики настроек Exchange Listener, в адресной строке браузера добавьте “/0/ClientApp/#/IntegrationDiagnostics/ExchangeListener” к URL-адресу вашего сайта Creatio и нажмите Enter. Например:

```
http://mycreatio.com/0/ClientApp/#/IntegrationDiagnostics/ExchangeListener
```

Страница диагностики состоит из нескольких блоков с параметрами синхронизации, по которым можно запустить диагностику (Рис. 3). Чтобы отобразилась информация о параметрах подключения, необходимо в блоке с интересующим вас параметром нажать [*Запустить диагностику*].

Рис. 3 — Диагностика настроек Exchange Listener

Диагностика сервиса Exchange Listener

Creatio

ЗАПУСТИТЬ ПОЛНУЮ ДИАГНОСТИКУ

Состояние доступности функциональности

- ✓ Синхронизация почты Exchange (ExchangeListenerEnabled)
- ✓ Синхронизация почты Exchange и IMAP (EmailIntegrationV2)
- ✗ Отправка писем (SendEmailsV2)

Проверка доступности сервиса

ЗАПУСТИТЬ ДИАГНОСТИКУ



Ошибка

Сервис ExchangeListener в данный момент не доступен для текущего сайта
Unable to connect to the remote server

Получение информации по подпискам

ЗАПУСТИТЬ ДИАГНОСТИКУ



Ошибка

Http failure response for
https://.../V0/rest/MailDiagnosticToolsService/GetSubscribersInfo: 500 OK

Проверка правильности заполнения системной настройки ExchangeListenerServiceUri

ЗАПУСТИТЬ ДИАГНОСТИКУ



Успешная операция

URL сервиса обработки событий Exchange в Creatio
https://.../V0/ServiceModel/ExchangeListenerService.svc/ProcessFullEmail валиден

Проверка работоспособности почтового ящика

Почтовый ящик



Отправить тестовое письмо

ЗАПУСТИТЬ ДИАГНОСТИКУ



Ошибка

Unable to connect to the remote server

Состояние доступности функциональности	<p>Для данного блока диагностика запускается автоматически при открытии страницы.</p> <p>Выполняется проверка подключения в Creatio функциональности, необходимой для работы Exchange Listener:</p> <ul style="list-style-type: none"> • ExchangeListenerEnabled; • EmailIntegrationV2; • SendEmailsV2. <p>Подробнее о подключении функциональности читайте в статье Механизм отключения функциональности Feature Toggle документации по разработке.</p>
Проверка доступности сервиса	Выполняется проверка доступности сервиса Exchange Listener из вашего приложения Creatio.
Получение информации по подпискам	Выполняется проверка соединения с удаленным сервером.
Проверка правильности заполнения системной настройки ExchangeListenerServiceUri	Выполняется проверка валидности адреса, указанного в системной настройке.
Проверка работоспособности почтового ящика	Выполняется проверка работы почтовых ящиков MS Exchange. Установите признак [<i>Отправить тестовое письмо</i>], чтобы при запуске диагностики на указанный адрес было отправлено тестовое email-сообщение.

Настроить запуск массовых рассылок

ПРОДУКТЫ: **MARKETING**

Функциональность доступна в продукте **Marketing Creatio** и в CRM-линейке продуктов Creatio.

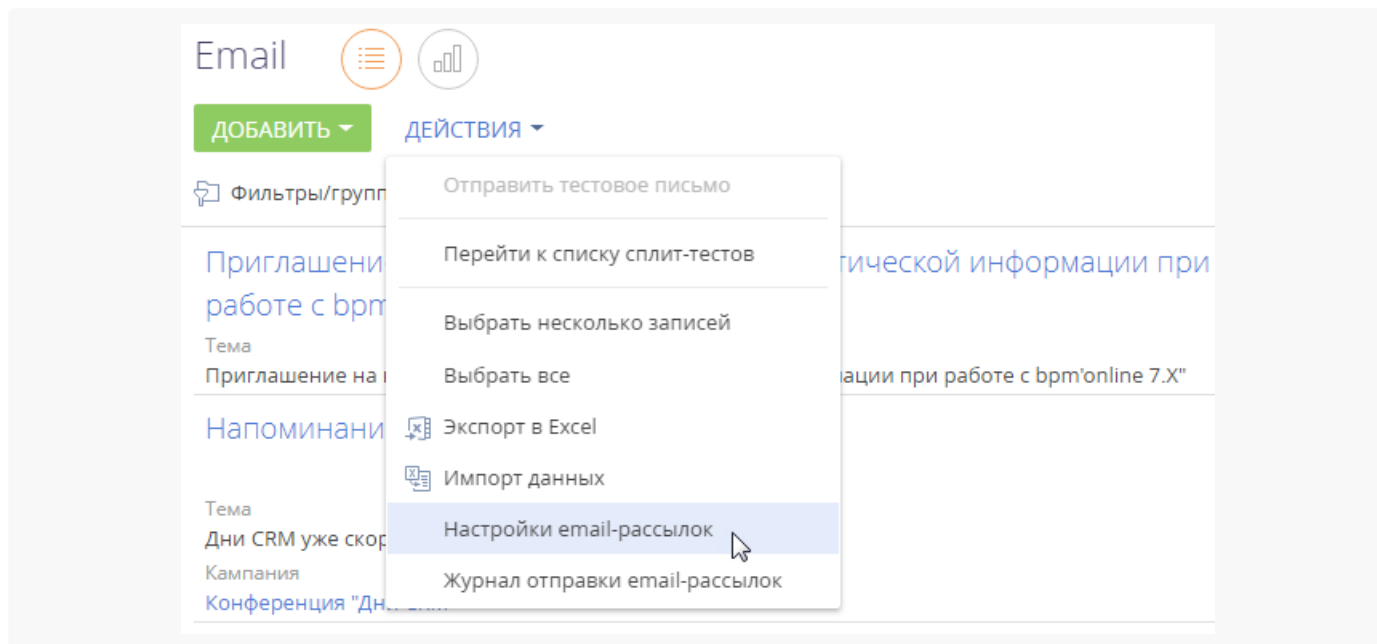
Чтобы иметь возможность запускать рассылки и отправлять по ним тестовые письма, необходимо настроить интеграцию Creatio с сервисом массовых рассылок. Для этого в разделе [*Email*] реализована страница настроек email-рассылок, с помощью которой можно внести или откорректировать общие настройки, список доменов отправителя, а также проконтролировать состояние подключения.

Настроить интеграцию с облачным сервисом отправки рассылок Cloud Email Service (для пользователей on-site)

Чтобы настроить интеграцию с облачным сервисом отправки рассылок:

1. Перейдите в раздел [*Email*]. В меню действий раздела выберите **Настройки email-рассылок** (Рис. 1).

Рис. 1 — Переход к настройкам email-рассылок



2. Заполните поля вкладки [*Общие настройки*].

Важно. Чтобы сменить провайдера сервиса email-рассылок, обратитесь в службу технической поддержки.

- a. В поле **Домен для получения откликов** укажите адрес домена вашего приложения Creatio в формате `http://www.yourdomain.com`.

Важно. POST-запросы всегда отправляются и принимаются по порту 443 вне зависимости от порта, через который доступно приложение Creatio. Мы рекомендуем проверить связь по порту 443 после завершения настроек. Для этого откройте в браузере полученный URL подключения к облачным сервисам Creatio в формате `https://url_address.com`.

В результате должна открыться пустая страница. Если страница не открывается, то необходимо проверить, корректно ли открыт порт.

- b. В поле **API-ключ** укажите персональный ключ доступа к сервису массовых рассылок.
- c. В поле **URL подключения к облачным сервисам Creatio** укажите адрес облачного сервиса рассылок в формате `https://url_address.com`.
- d. В поле **Auth ключ** укажите ключ аутентификации для получения откликов.

Для получения API-ключа и Auth-ключа, а также URL подключения к облачным сервисам рассылок после установки лицензий продукта обратитесь в службу технической поддержки.


- e. Поле **Email-провайдер** будет автоматически заполнено названием вашего провайдера сервиса email-рассылок сразу после корректного заполнения полей [*API-ключ*] и [*URL подключения к облачным сервисам Creatio*].

Настроить список доменов отправителя

Для отображения в рассылках корректного имени отправителя, а также чтобы избежать несанкционированных рассылок от вашего имени, необходимо:

- перечислить на странице настроек email-рассылок список ваших доменов;
- верифицировать каждый домен с помощью специальных текстовых SPF-, DKIM- и DMARC-записей;
- Сохраните изменения на странице системной настройки.

Для этого:


1. Добавьте список ваших доменов по кнопке  детали на вкладке [*Домены отправителя*].

На заметку. В списке отображаются все добавленные домены, включая те, которые уже не используются. Удалить домен из списка невозможно.

2. Выберите из списка домен, который необходимо верифицировать. В правой части экрана отобразится инструкция по настройке DKIM/SPF, актуальная для выбранного домена. В инструкции будут сформированы значения DKIM- и SPF-записей.

На заметку. Инструкции по настройке DKIM/SPF отличаются для разных доменов. Для того чтобы отобразилась корректная инструкция, выберите нужный домен из списка добавленных.

3. Выполните настройки верификации домена. Подробнее: [Рекомендации по настройке для популярных DNS-провайдеров](#).

В результате в поле [*Состояние подключения*] страницы настроек email-рассылок отобразится индикатор успешного подключения  и комментарий “Подключение активно”.

Выполнить дополнительные настройки интеграции с сервисом отправки рассылок

Для корректной работы функциональности рассылок **настройте один из вариантов доступа** к Creatio для сервиса рассылок (Creatio Cloud Email Service).

1. В межсетевом экране продуктового сервера настройте доступ для приема POST-запросов из Internet к домену, на котором развернуто ваше приложение: <http://www.yourdomain.com>.
2. В межсетевом экране продуктового сервера настройте доступ для приема POST-запросов из Internet к конкретному веб-сервису. Если ваше приложение развернуто по адресу <http://www.yourdomain.com>, то из Сети должен быть доступен сервис <http://www.yourdomain.com/0/ServiceModel/CESWebHooksService.svc/HandleWebHooks>.

На заметку. В Creatio нет необходимости настраивать обработку запросов отписки от рассылок и проверять возможность получения сервером приложений Creatio GET-запросов. Обработка запросов на отписку от рассылок осуществляется автоматически.

Важно. Если доступ к приложению осуществляется с использованием защищенного соединения HTTPS, то необходимо, чтобы на сервере приложений был установлен действующий сертификат. В случае изменения протокола передачи данных или адреса приложения, соответствующие изменения можно внести на странице настройки email-рассылок.

Не рекомендуется использовать “белые списки” IP-адресов для ограничения доступа к открытым портам, поскольку служба рассылок Creatio Cloud Email Service может присылать аналитическую информацию об отклике с различных IP-адресов и использовать прокси. Если в таком “белом списке” будет отсутствовать IP-адрес, с которого отправляется аналитическая информация, то эта информация будет утеряна.


Также при использовании “черных списков” рекомендуется проверить, что полученные IP-адреса не запрещены (не находятся в данном списке).

Разрешить мониторинг email-рассылок в Creatio on-site

Перед началом работы с рассылками рекомендуем настроить возможность мониторинга состояния email-рассылок для сотрудников службы поддержки. Так они смогут оперативно помочь вам восстановить отправку рассылок в случае сбоев, например, задержек или ошибок отправки. Показатели, которые будет анализировать служба поддержки, содержат только агрегированные данные по конкретной рассылке и не содержат таких данных, как персональные email-сообщения, шаблоны рассылок и т. д.

На заметку. Настройка различается для приложений, развернутых on-site и cloud. Подробнее о настройке для приложений cloud читайте в статье [Разрешить мониторинг состояния рассылок службе поддержки](#).

Для настройки:

1. Перейдите в дизайнер системы по кнопке  в правом верхнем углу приложения и откройте раздел [Системные настройки].
2. Откройте системную настройку [Включить возможность мониторинга показателей состояния email-рассылок] и на ее странице установите признак [Значение по умолчанию]. Сохраните изменения на странице системной настройки.
3. В межсетевом экране продуктового сервера настройте доступ из Internet к web-сервису /0/ServiceModel/CESTroubleshootingService.svc/emailstate.

Например, если ваше приложение развернуто по адресу <http://www.yourdomain.com>, то из сети должен быть доступен web-сервис:

<http://www.yourdomain.com/0/ServiceModel/CESTroubleshootingService.svc/emailstate>.

В результате выполненных настроек сотрудники службы поддержки смогут оперативно определять и устранять возможные сбои и восстановить работу ваших рассылок.

Настроить сервер приложения на IIS

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

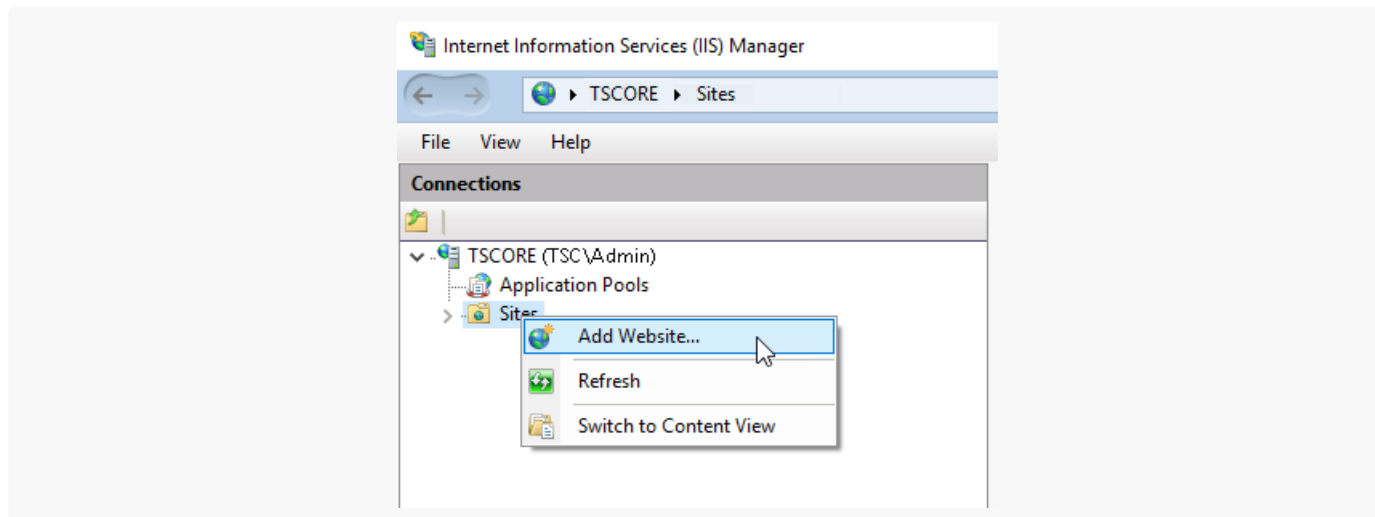
Настройка сервера приложений Creatio (веб-сервера) на IIS включает в себя настройку веб-сайта приложения в IIS и добавление пула приложения.

Настроить сайт приложения в IIS

Для создания и настройки web-сайта:

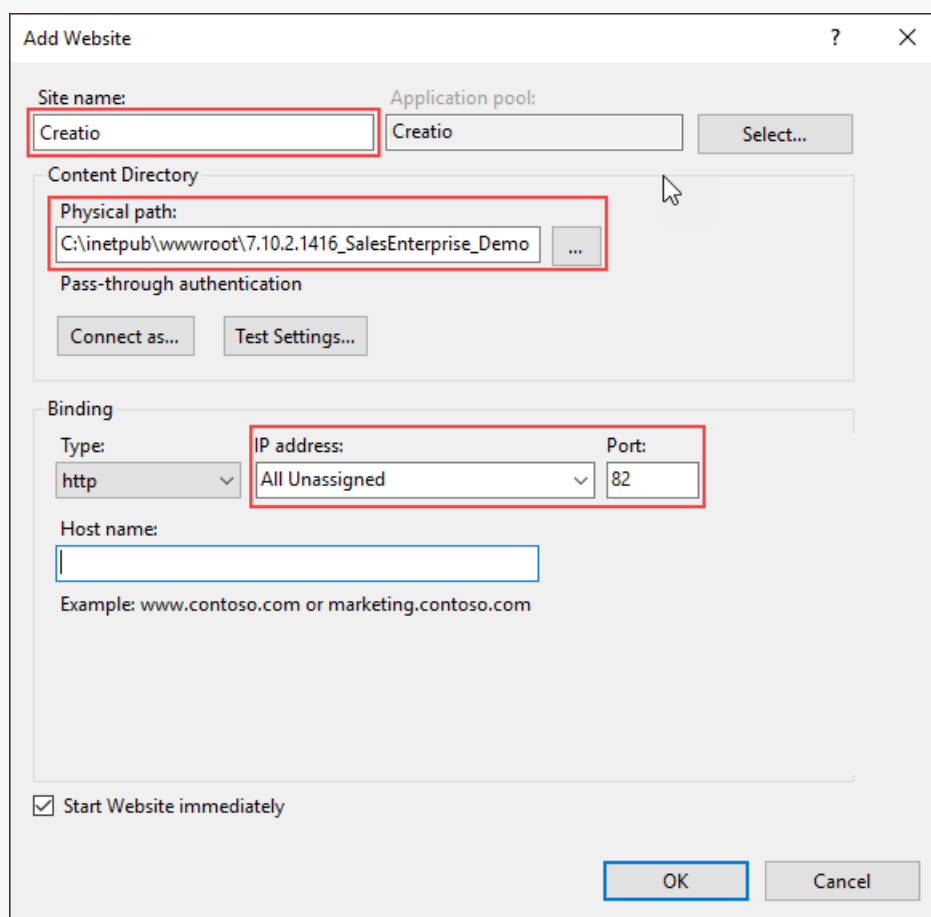
1. В окне управления IIS щелкните правой клавишей мыши по [*Sites*] и выберите [*Add Website*] в контекстном меню (Рис. 1).

Рис. 1 — Создание сайта для приложения Creatio



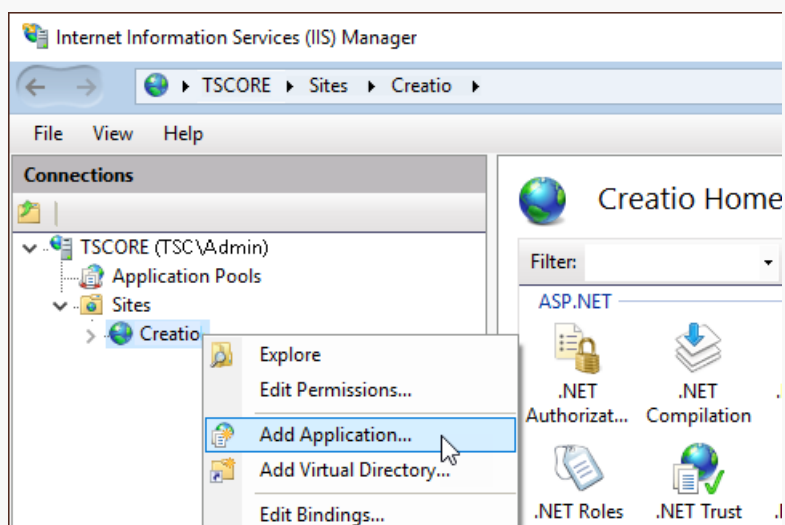
2. Укажите название сайта, путь к корневому каталогу с файлами Creatio, используемые IP-адреса и порт сайта (Рис. 2). Путь по умолчанию для web-сайтов является путь C:\inetpub\wwwroot. В случае необходимости можно указать свой адрес.

Рис. 2 — Окно параметров нового сайта



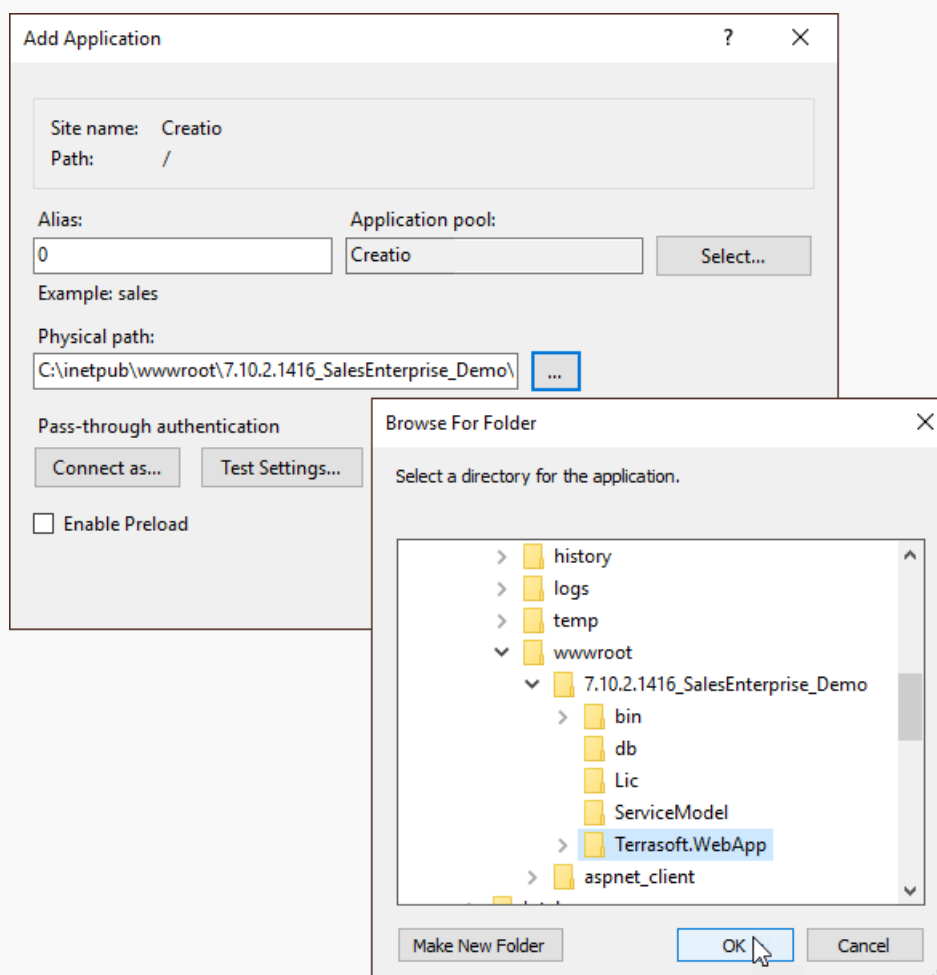
3. В области [*Connections*] щелкните правой клавишей мыши по созданному сайту и выберите команду [*Add Application*] контекстного меню (Рис. 3).

Рис. 3 — Выбор команды добавления приложения в IIS



4. Введите имя приложения "0" в поле [*Alias*]. Укажите директорию "Terrasoft.WEBApp" (Рис. 4).

Рис. 4 — Окно выбора параметров приложения

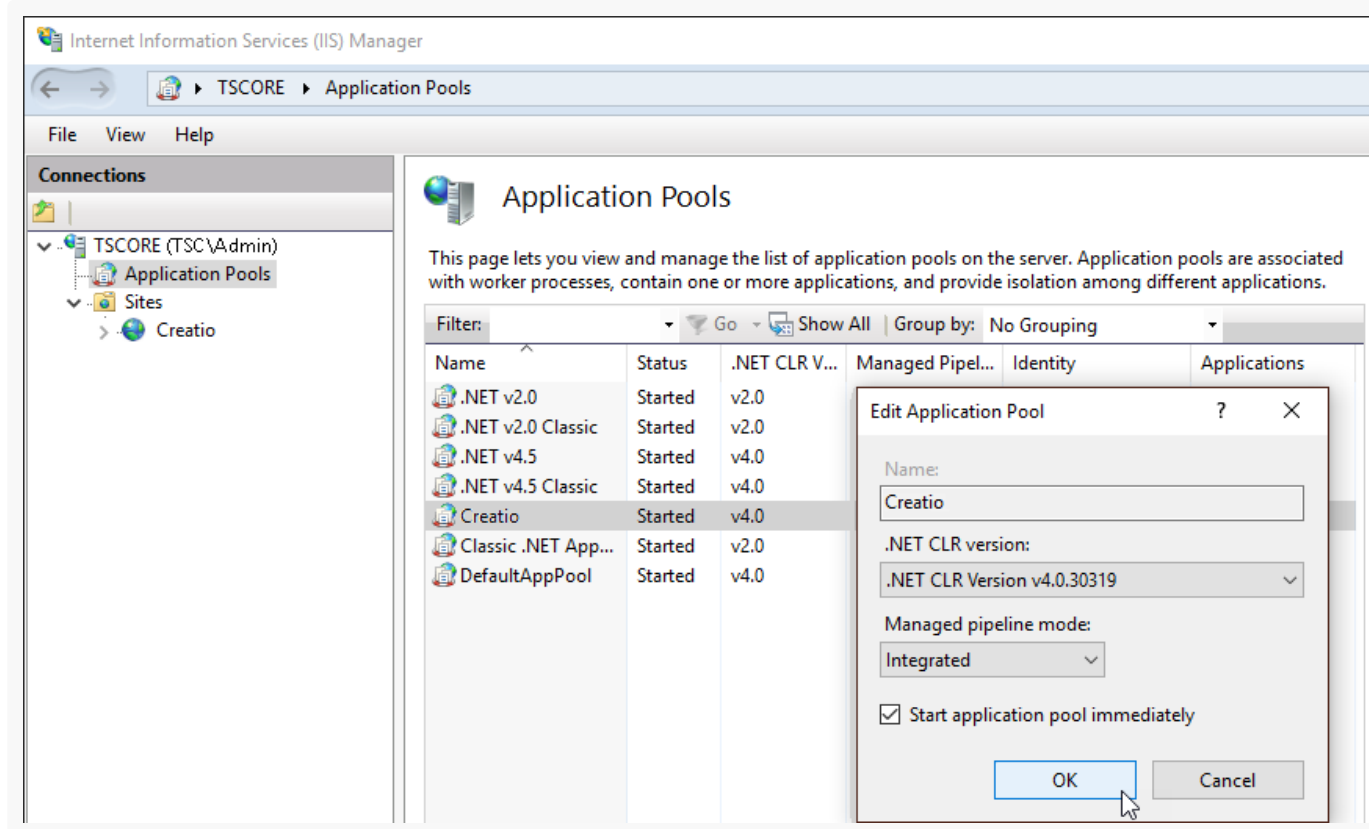


Добавить пул приложения

Для добавления пула приложения:

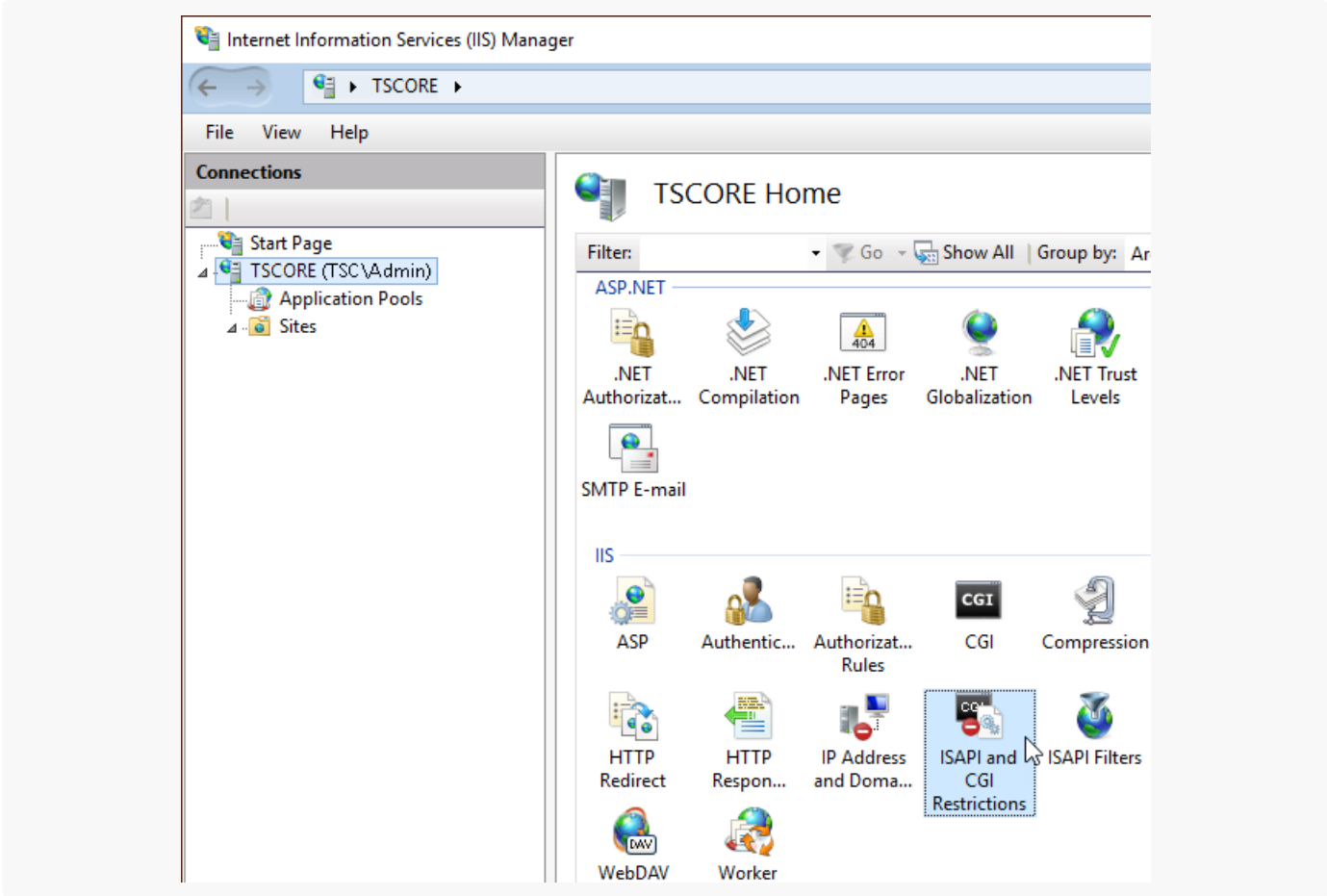
1. В области [*Connections*] окна управления IIS перейдите в секцию [*Applications Pools*].
2. Выберите пул [*Creatio*].
3. В поле [*Managed pipeline mode*] выберите режим [*Integrated*].
4. В поле [*.Net Framework version*] укажите версию Asp.Net - 4.0.30319 (Рис. 5).

Рис. 5 — Окно ввода параметров Applications Pools



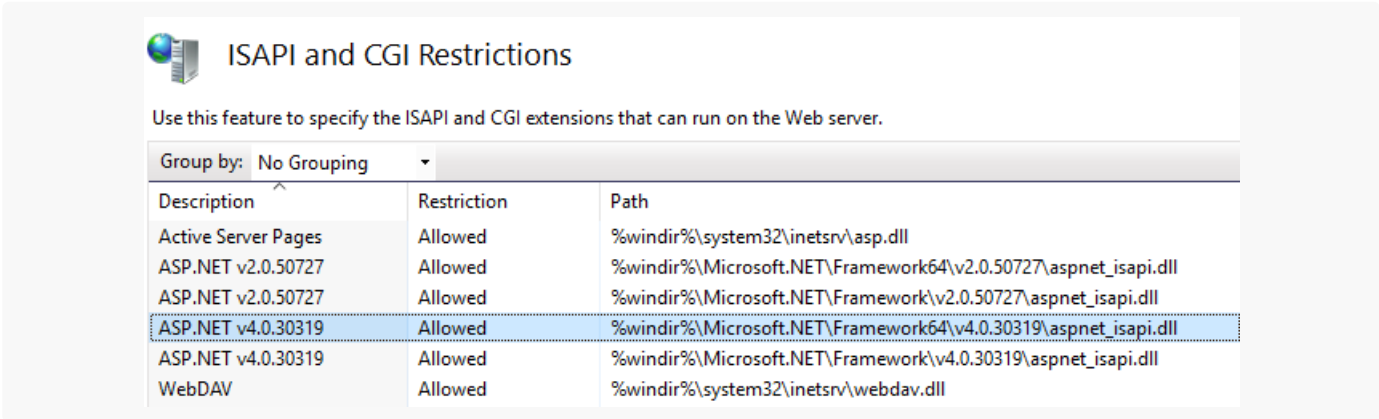
5. Проверьте, что введенная версия ASP.NET разрешена. Для этого перейдите в ISAPI and CGI Restrictions на уровне web-сервера (Рис. 6).

Рис. 6 — Переход в ISAPI and CGI Restrictions



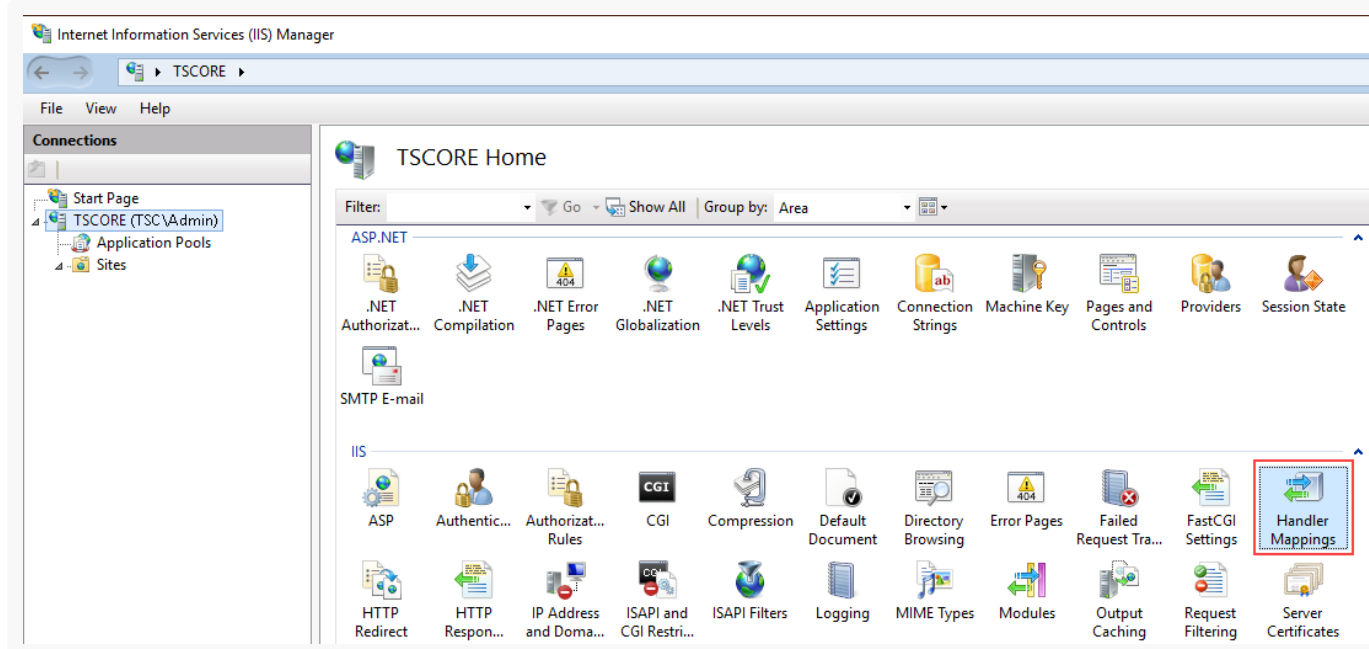
6. Убедитесь что для используемой версии ASP.Net в поле [*Restriction*] установлен статус [*Allowed*] (Рис. 7).

Рис. 7 — Проверка статуса используемой версии ASP.Net



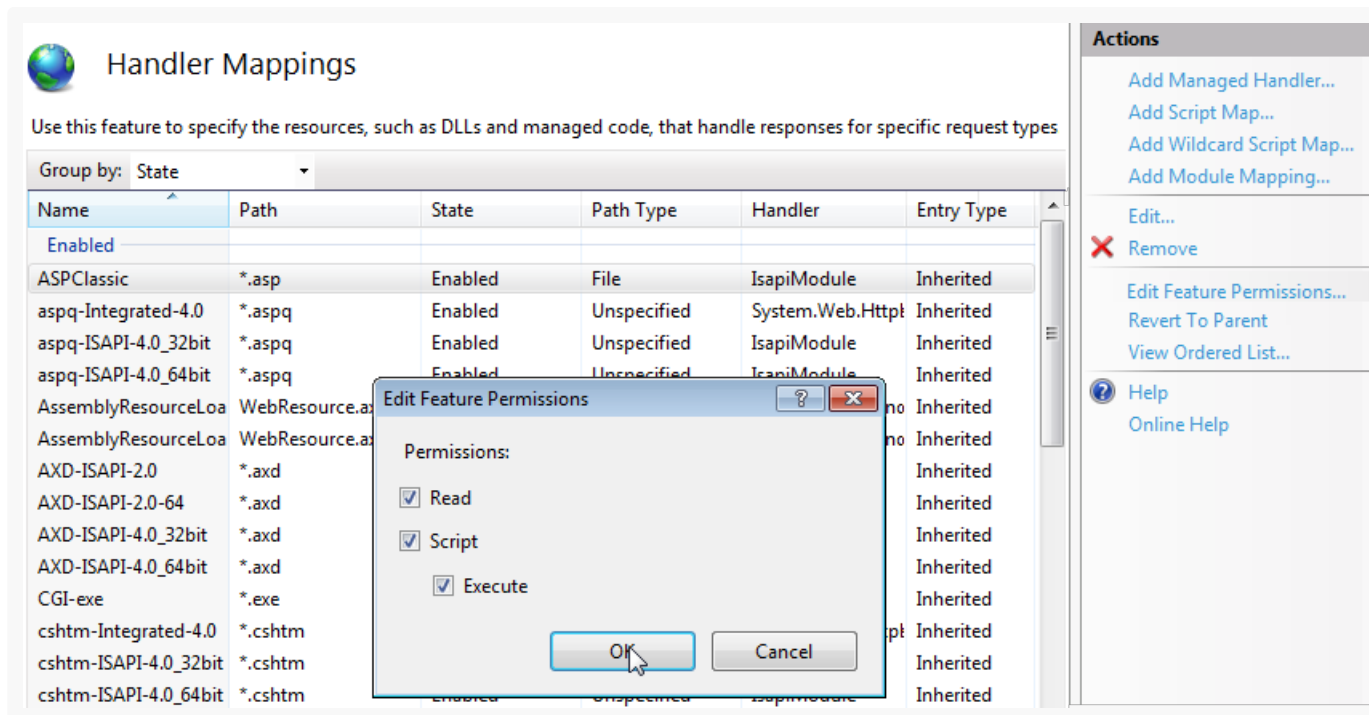
7. Убедитесь в том, что все необходимые разрешения активны. Для этого перейдите в Handler Mappings на уровне сервера (Рис. 8).

Рис. 8 — Переход в Handler Mappings



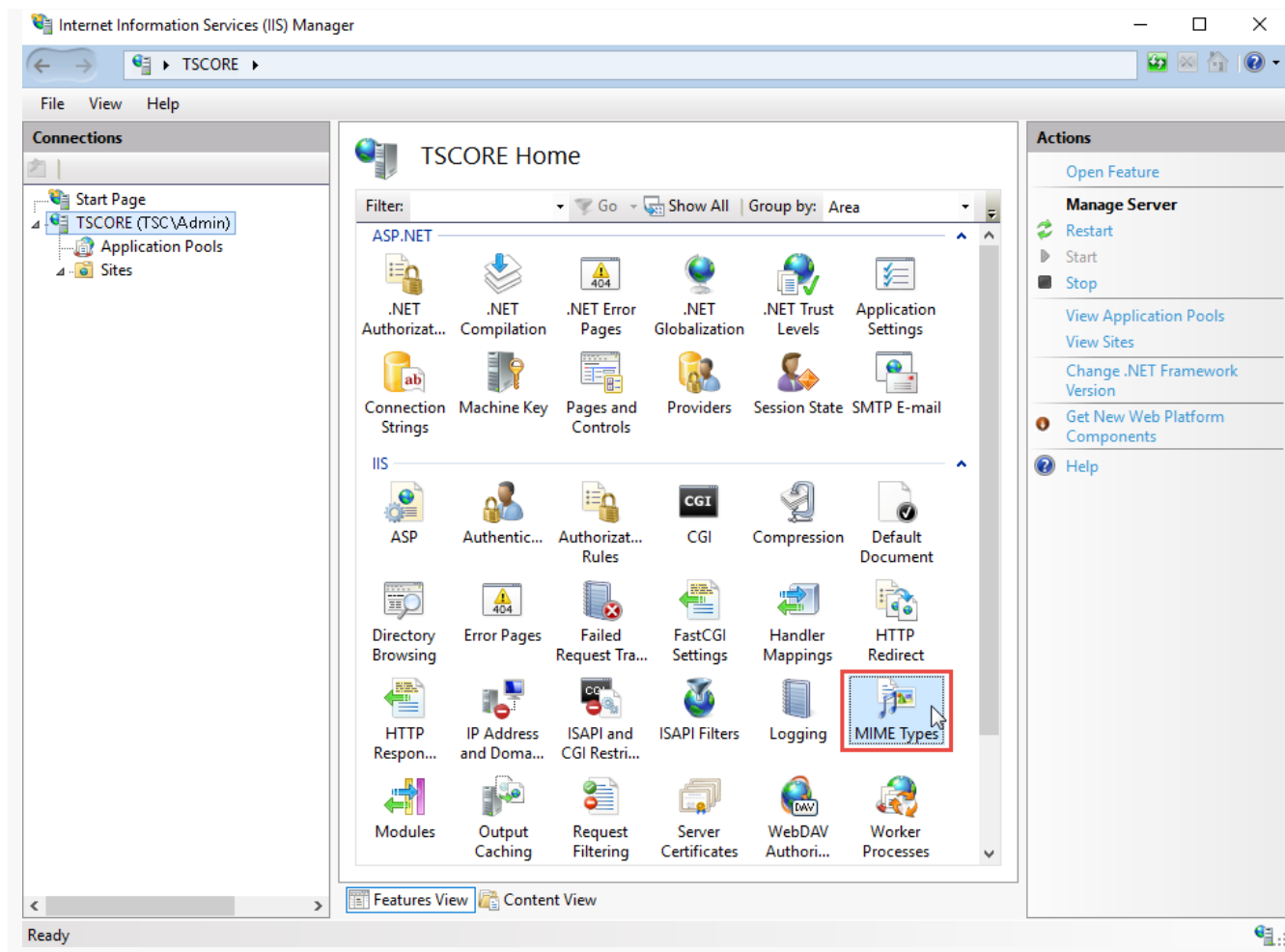
8. Нажмите на [*Edit Feature Permissions*] в области [*Actions*].
9. Убедитесь, что в окне [*Edit Feature Permissions*] установлены все признаки (Рис. 9).

Рис. 9 — Проверка необходимых параметров в окне [*Edit Feature Permissions*]



10. Убедитесь, что у нового приложения сконфигурирован MIME-тип для файлов с расширениями .svg и .json. Данная конфигурация может быть выполнена как на уровне сервера (в этом случае она наследуется для всех размещенных на нем приложений), так и на уровне приложения. Для проверки:
 - а. Перейдите в MIME Types на уровне сервера или на уровне приложения (Рис. 10).

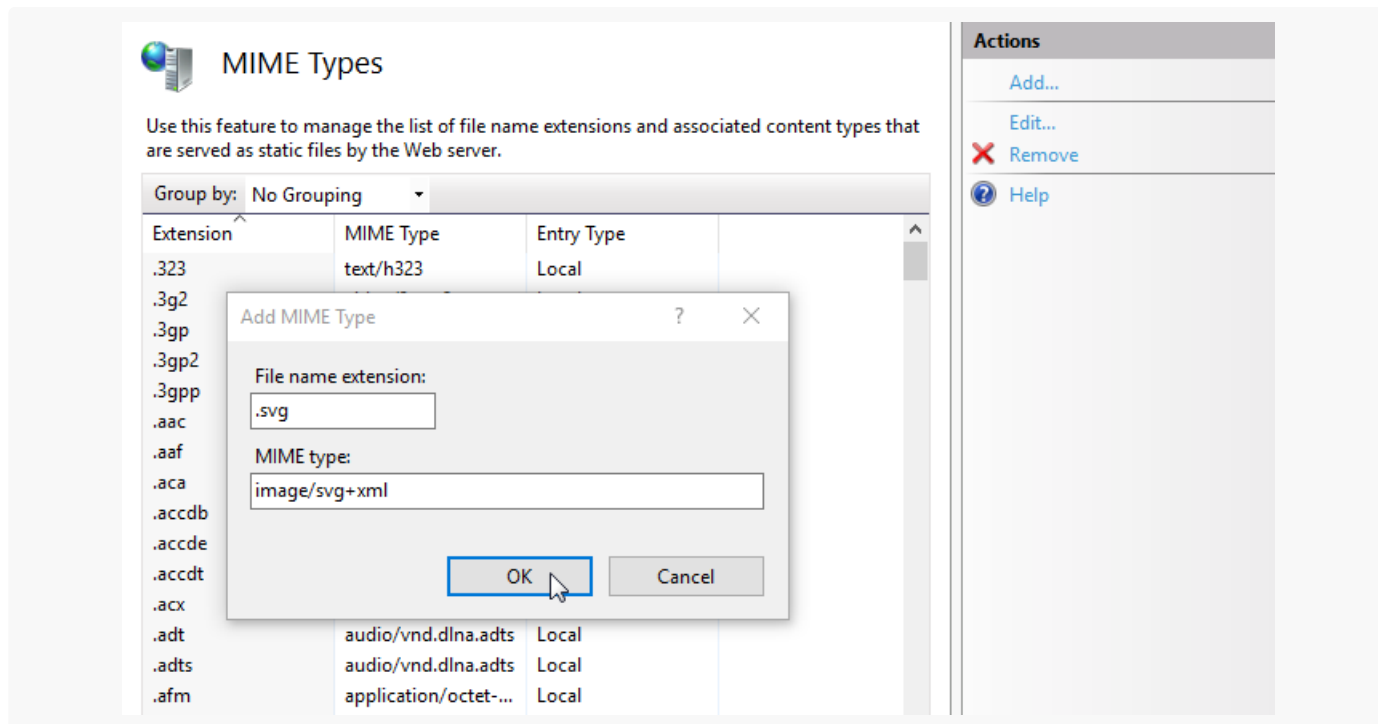
Рис. 10 — Переход в MIME Types



b. Проверьте в списке наличие конфигурации для файлов с расширением .svg и .json. Если такая конфигурация есть, переходите на шаг 12.

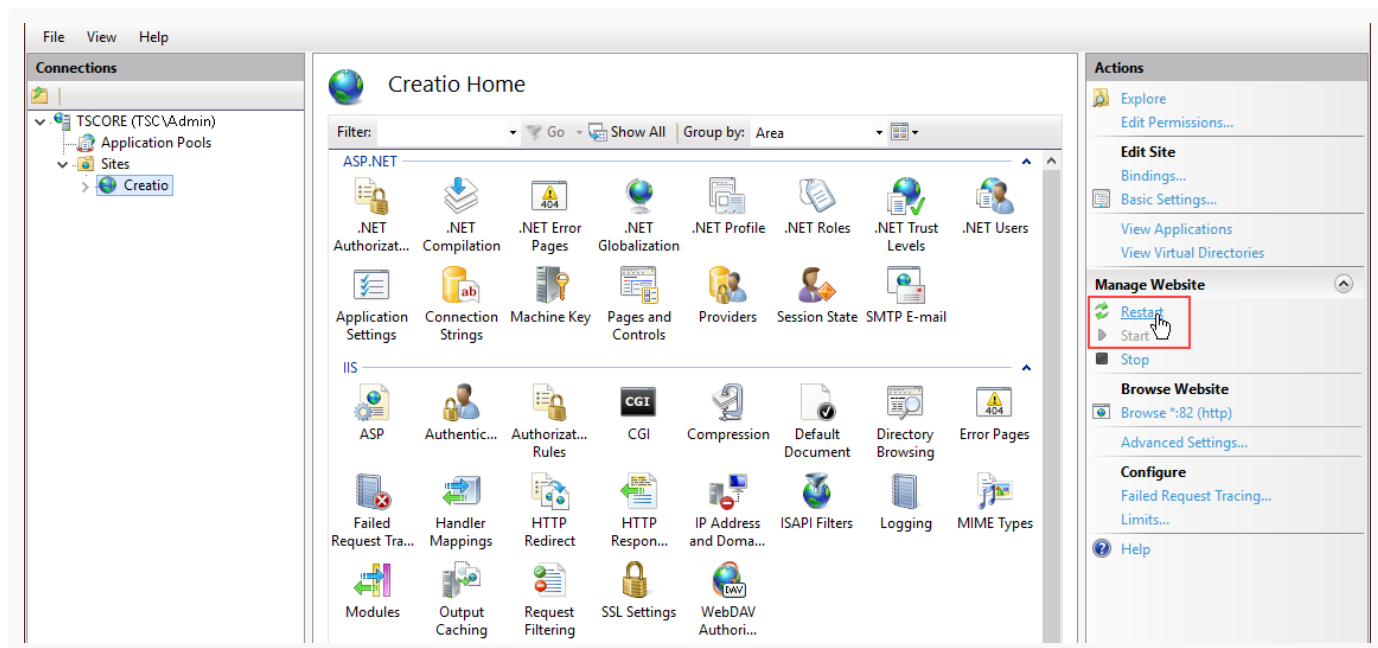
11. Если конфигурации нет, нажмите [Add...] в области [Actions]. В открывшемся окне укажите расширение .svg и MIME-тип данных, соответствующий этому расширению (Рис. 11). Повторите те же действия для расширения .json (MIME-тип "application/json").

Рис. 11 — Настройка MIME-типа данных для файлов с расширением .svg



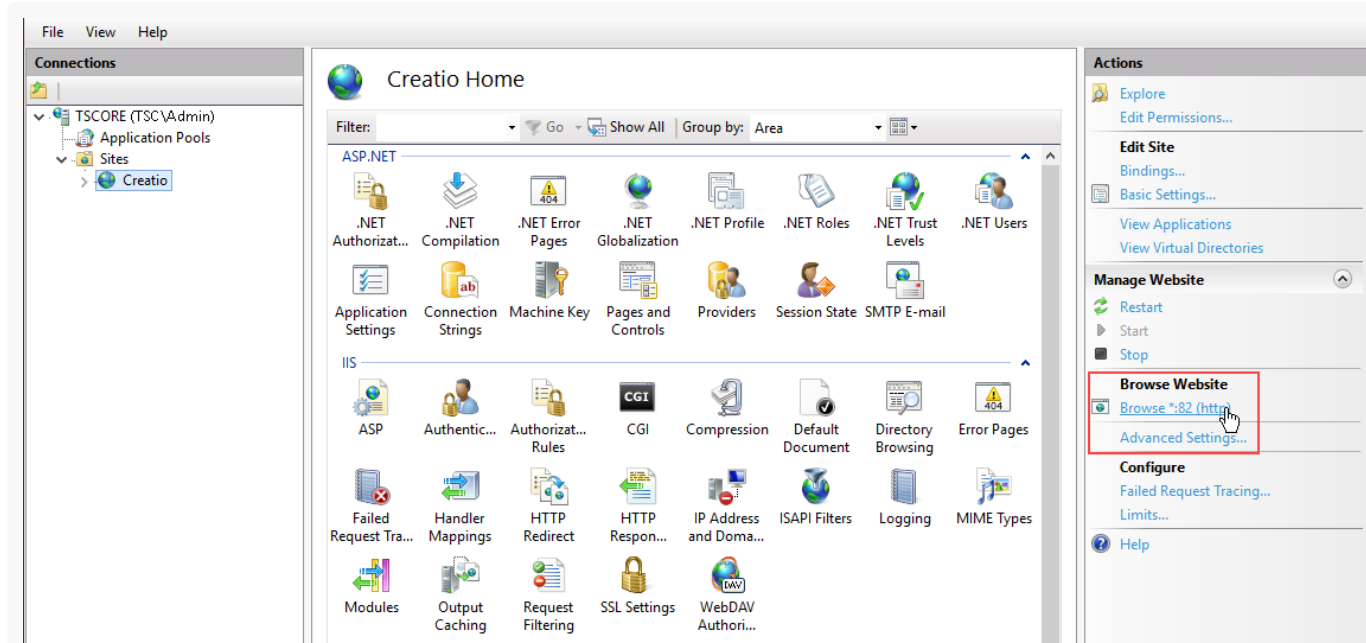
12. Перезапустите web-сайт при помощи команды [*Restart*] области [*Manage Website*] (Рис. 12).

Рис. 12 — Выбор команды [*Restart*] в области [*Manage Websites*]



13. Откройте сайт, перейдя по адресу или при помощи команды [*Browse*] (Рис. 13). Убедитесь, что открывается страница авторизации в приложении.

Рис. 13 — Выбор команды [*Browse*] в области [*Actions*] веб-сайта



На заметку. Первый вход в систему после настройки приложения осуществляется под учетной записью Supervisor (логин — Supervisor; пароль — Supervisor). Рекомендуем после первого входа изменить пароль этой учетной записи.

14. Для включения русского языка:

- Из дизайнера системы перейдите в раздел [*Языки*].
- Выберите русский язык и нажмите [*Открыть*].
- На открывшейся странице установите признаки [*Активен*] и [*Использовать по умолчанию*]. Сохраните изменения.

Для включения языка необходимо, чтобы пользователю, от имени которого в IIS запущен пул приложения, были предоставлены права на чтение, запись и удаление файлов и вложенных каталогов статического контента приложения (каталог .\Terrasoft.WebApp(conf)).

15. Из дизайнера системы перейдите в раздел [*Системные настройки*] и измените значение системной настройки “Порядок ФИО” на “Last name, First name [*Middle name*]”. Это необходимо для корректной разбивки ФИО контактов на отдельные колонки [*Фамилия*], [*Имя*], [*Отчество*].

Настроить горизонтальное масштабирование

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

В Creatio существует возможность повысить производительность крупных проектов (до нескольких тысяч пользователей) за счет горизонтального масштабирования — увеличения количества серверов, на которых развернуто приложение, и распределения нагрузки между ними.

Балансировщик может быть аппаратным или программным. Для работы в отказоустойчивом режиме

используется балансировщик HTTP/HTTPS-трафика с поддержкой протокола WebSocket. Работа приложения тестировалась на программном балансировщике нагрузки HAProxy. Известны случаи успешного использования других балансировщиков Citrix, Cisco, NginX, FortiGate, MS ARR.

На заметку. Установка дополнений Marketplace и пользовательских доработок на среду с балансировщиком отличается от обычного процесса поставки. Подробнее: [Установить приложение Marketplace](#).

Данный документ рассматривает вариант горизонтального масштабирования Creatio с использованием бесплатного open-source балансировщика HAProxy для распределения нагрузки на серверы сессий.

На заметку. Серверное время на нодах (серверах и машинах), на которых развернуты экземпляры приложения Creatio, должно быть синхронизировано во избежание проблем с работоспособностью системы.

Общий порядок развертывания

Для приложений на .NET Framework

Общий порядок развертывания приложения Creatio с горизонтальным масштабированием на **.NET Framework** следующий:

1. Разверните необходимое количество экземпляров приложения Creatio в web-ферме.

На заметку. Рекомендуется, чтобы у всех экземпляров приложения имена в IIS и настройки Application pool совпадали.

2. В файле ConnectionStrings.config для всех экземпляров укажите одинаковые базы данных SQL и Redis.

```
<add name="redis" connectionString="host=DOMAIN.COM;db=0;port=6379;maxReadPoolSize=10;maxWrit
<add name="db" connectionString="Data Source=DOMAIN.COM;Initial Catalog=DatabaseName;Integrat
```

3. В конфигурационном файле (Web.config) каждого приложения, в блоке <appSettings> добавьте ключ:

```
<add key="TenantId" value="1" />
```

Номер "value" должен быть одинаковым у всех экземпляров приложения в web-ферме.

Важно. Начиная с версии 7.14.1 ключ <add key="TenantId" value="..." /> нужно добавлять только во внутренний конфигурационный файл Web.config (путь к файлу Terrasoft.WebApp\Web.config). Добавление ключа во внешний конфигурационный файл может привести к ошибкам в работе

приложения.

4. Сгенерируйте для одного из экземпляров приложения уникальное значение machineKey. Подробно об этом читайте в статье [Настроить Web.config](#). Полученное значение скопируйте и укажите для каждого экземпляра приложения в конфигурационных файлах Web.config, которые находятся в корневой папке Creatio и в папке Terrasoft.WebApp.
5. Во внешнем конфигурационном файле (Web.config) каждой ноды для всех планировщиков в блоке <quartzConfig> включите кластерный режим:

```
<add key="quartz.jobStore.clustered" value="true" />
<add key="quartz.jobStore.acquireTriggersWithinLock" value="true" />
```

6. В случае совпадения настроек instanceId сформируйте уникальные значения для каждой ноды планировщиков.

Способы формирования уникальных instanceId:

- Во внешнем конфигурационном файле (Web.config) каждой ноды для всех планировщиков в блоке <quartzConfig> добавьте строку

```
<add key="quartz.scheduler.instanceId" value="AUTO" />
```

Важно. Для значения AUTO атрибута value необходимо использовать верхний регистр. В другом случае значение будет расцениваться как имя ноды и при работе планировщика могут возникать ошибки.

В результате планировщик автоматически будет генерировать уникальное имя ноды в формате <имя ноды>+timestamp.

- Вручную добавьте уникальные значения quartz.scheduler.instanceId.
7. Для атрибута value настройки quartz.jobStore.clustered установите значение true.

```
<add key="quartz.jobStore.clustered" value="true" />
```

8. Раздайте права на созданные директории приложений пользователю IUSR и пользователю, под которым запущен Application pool в IIS.
9. Настройте балансировщик (например, NАproxу) для распределения нагрузки между развернутыми серверами приложения.
10. При необходимости настройте балансировку нагрузки для серверов баз данных и сессий.

На заметку. Информация о настройке кластеризации доступна в документации [MSSQL](#) и [Oracle](#). Настройка отказоустойчивости системы при помощи Redis Cluster рассмотрена в статье [Настроить Redis Cluster](#).

Для приложений на .NET Core

Общий порядок развертывания приложения Creatio с горизонтальным масштабированием на **.NET Core** следующий:

1. Разверните необходимое количество [экземпляров приложения Creatio](#).
2. В [файле ConnectionStrings.config](#) для всех экземпляров укажите одинаковые базы данных SQL и Redis для всех экземпляров приложения.
3. Перейдите в корневую папку любого из экземпляров приложения и найдите файл Terrasoft.WebHost.dll.
4. Запустите команду:

```
dotnet Terrasoft.WebHost.dll configureWebFarmMode
```

В результате конфигурационные файлы данного экземпляра приложения обновятся.

5. Во внешнем конфигурационном файле (Terrasoft.WebHost.dll) каждой ноды для всех планировщиков в блоке <quartzConfig> включите кластерный режим:

```
<add key="quartz.jobStore.clustered" value="true" />
<add key="quartz.jobStore.acquireTriggersWithinLock" value="true" />
```

6. В случае совпадения настроек instanceId сформируйте уникальные значения для каждой ноды планировщиков.

Способы формирования уникальных instanceId:

- Во внешнем конфигурационном файле (Terrasoft.WebHost.dll) каждой ноды для всех планировщиков в блоке <quartzConfig> добавьте строку

```
<add key="quartz.scheduler.instanceId" value="AUTO" />
```

Важно. Для значения AUTO атрибута value необходимо использовать верхний регистр. В другом случае значение будет расцениваться как имя ноды и при работе планировщика могут возникать ошибки.

В результате планировщик автоматически будет генерировать уникальное имя ноды в формате <имя ноды>+timestamp.

- Вручную добавьте уникальные значения quartz.scheduler.instanceId.
7. Для атрибута value настройки quartz.jobStore.clustered установите значение true.


```
<add key="quartz.jobStore.clustered" value="true" />
```

8. При необходимости настройте балансировку нагрузки для серверов баз данных и сессий.
9. Скопируйте все обновленные конфигурационные файлы в корневые папки других экземпляров приложения.
10. Настройте балансировщик (например, HAProxy) для распределения нагрузки между развернутыми серверами приложения.

На заметку. Подробная информация о создании и настройке кластеров содержится в документации СУБД. Настройка отказоустойчивости системы при помощи Redis Cluster рассмотрена в статье [Настроить Redis Cluster](#).

Установить балансировщик HAProxy

Балансировщик нагрузки HAProxy поддерживает ряд бесплатных open-source ОС. В данном документе мы рассмотрим один из наиболее простых способов развертывания HAProxy на ОС Debian при помощи сервиса haproxy.debian.net.

1. Откройте страницу сервиса установки, перейдя по ссылке <https://haproxy.debian.net/>.
2. Выберите ОС и ее версию, а также версию HAProxy.

На заметку. Чтобы узнать установленную версию Debian, воспользуйтесь командой `cat /etc/issue`.

В результате сервис сгенерирует набор команд, которые необходимо выполнить в ОС Debian для установки HAProxy.

Рис. 1 — Пример команд установки HAProxy, сгенерированных сервисом haproxy.debian.net

Instructions for latest release

First, you need to enable the [backports repository](#):

```
# echo deb http://httpredir.debian.org/debian jessie-backports main | \
tee /etc/apt/sources.list.d/backports.list
```

Then, you need to enable a dedicated repository:

```
# curl https://haproxy.debian.net/bernad.debian.org.gpg | \
apt-key add -
# echo deb http://haproxy.debian.net jessie-backports-1.8 main | \
tee /etc/apt/sources.list.d/haproxy.list
```

Then, use the following commands:

```
# apt-get update
# apt-get install haproxy -t jessie-backports\*
```

3. Выполните сгенерированные команды одну за другой.

Настроить балансировщик HAProxy

Для настройки HAProxy необходимо внести изменения в файл `haproxy.cfg`. Файл находится по следующему пути:

```
.../etc/haproxy/haproxy.cfg
```

Основные (минимальные) настройки

Минимальные настройки, необходимые для работы HAProxy, состоят в добавлении в файл двух секций: **frontend** и **backend**.

Секция frontend

В секцию frontend необходимо добавить 2 настройки: **bind** и **default_backend**:

- В настройке **bind** укажите адрес и порт, на который будут поступать запросы, распределение которых будет производить HAProxy.
- В опции **default_backend** укажите имя, которое будет указано для секции backend.

В результате настройка будет выглядеть следующим образом:

```
frontend front
maxconn 10000
#Using these ports for binding
bind *:80
bind *:443
#Convert cookies to be secure
rspirep ^(set-cookie:.*)\ \ Secure
default_backend creatio
```

Секция backend

В секцию backend необходимо добавить как минимум 2 обязательные настройки:

- В параметре **balance** укажите тип балансировки, например **roundrobin**. Информация о различных типах балансировки доступна в [документации HAProxy](#).
- При помощи параметра **server** укажите все серверы (или “nodes”), между которыми должна распределяться нагрузка.

Для каждого сервера (развернутого экземпляра приложения Creatio) необходимо добавить отдельный параметр server с указанием адреса сервера, порта и веса. Вес позволяет балансировщику распределять нагрузку на основании физических возможностей серверов. Чем больший вес указан для сервера, тем больше запросов он будет получать. Например, если необходимо распределить нагрузку между двумя серверами Creatio, добавьте в backend 2 параметра server:

```
server node_1 [server address]:[port] weight
server node_2 [server address]:[port] weight
```

В результате настройка будет выглядеть следующим образом:

```
backend creatio
#set balance type
balance roundrobin

server node_1 nodeserver1:80 check inter 10000 weight 2
server node_2 nodeserver2:80/sitename check inter 10000 weight 1
```

Новые настройки вступят в силу после перезапуска HAProxy. Используйте следующую команду для перезапуска HAProxy:

```
service haproxy restart
```

Проверить состояние сервера

С точки зрения балансировщика HAProxy у сервера может быть несколько состояний:

Состояние	Описание
UP	Сервер работает.
UP - transitionally DOWN	Сервер в настоящий момент считается работоспособным, но последняя проверка не удалась. Следовательно, сервер переходит в состояние DOWN.
DOWN - transitionally UP	В настоящее время сервер считается неработоспособным, но последняя проверка прошла успешно. Следовательно, сервер переходит в состояние UP.
DOWN	Сервер не работает.

Изменения рабочего состояния инициируются параметрами проверки работоспособности (health check). Для самой простой проверки работоспособности необходимо ключевое слово `check` в строке настройки `server`. Для запуска проверки работоспособности требуется как минимум IP-адрес и порт TCP от сервера. Пример проверки:

```
server node1 ... check
option httpchk GET /Login/NuiLogin.aspx
option httpchk GET /0/ping
```

Настроить веб-статистику (опционально)

Чтобы включить веб-статистику, добавьте новую секцию `listen` со следующими параметрами: **bind**, **mode http**, **stats enable**, **stats uri**. Секция выглядит следующим образом:

```
listen stats # Define a listen section called "stats"
  bind :9000 # Listen on localhost:9000
  mode http
  stats enable # Enable stats page
  stats uri /haproxy_stats # Stats URI
```

В результате веб-статистика балансировки нагрузки Creatio будет доступна для просмотра в браузере.

Рис. 2 — Пример веб-статистики балансировщика нагрузки

HAProxy

Statistics Report for pid 7106

> General process information

pid = 7106 (process #1, reprocs = 1)
 uptime = 38.04m46s
 system limits: memmax = unlimited, ulimit-m = 4032
 maxrss0 = 4032, memmax0 = 2000, memmax1s = 0
 current csws = 1, current psws = 0, csws rate = 1560
 Running tasks: 1/0, csw = 100 %

active UP

active UP, going down

active DOWN, going up

active or backup DOWN

active or backup DOWN for maintenance (MAINT)

inactive or backup SOFT STOPPED for maintenance

backup UP

backup UP, going down

backup DOWN, going up

[not checked]

[not checked]

UP with load-balancing disabled

Display option:

Scope

High-TOC/01_backend

Disable refresh

Refresh now

CSV export

External resources:

Proxy.sh

log4j2.properties

Colors manual

Stats		Queue		Session rate		Limit		Sessions		Bytes		Denied		Errors		Warnings		Status		Server	
Frontend	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit
Frontend	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<div>High-TOC/01_backend</div> <div> <div> <div>1</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> </div> </div>																					
<div>Backend</div> <div> <div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> </div> </div>																					

Choose the action to perform on the checked servers:

Apply

Stats		Queue		Session rate		Limit		Sessions		Bytes		Denied		Errors		Warnings		Status		Server	
Frontend	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit
Frontend	0	0	1	0	1	0	0	1	2	200	21	0	0	0	0	0	0	0	0	0	0
Backend	0	0	0	0	1	0	0	1	200	20	0	0	0	0	0	20	0	0	0	0	0

Для просмотра статистики перейдите по адресу: [*адрес балансировщика*]:9000/haproxy stats.

Настроить отображение IP-адресов в журнале аудита для .Net Core (опционально)

При работе веб-фермы запросы пользователей приходят на веб-серверы через балансировщик и/или прокси-сервер. В этом случае по умолчанию в [журнале аудита](#) будет отображаться адрес последнего из прокси-серверов, через которые прошел запрос пользователя, а не его IP-адрес.

Вы можете настроить отображение в журнале аудита реального IP-адреса пользователя. Для этого:

1. Настройте балансировщик таким образом, чтобы каждому запросу, который он перенаправляет на один из экземпляров приложения, был установлен заголовок, имя которого соответствует "ForwardedForHeaderName", а значение — IP-адресу клиента.
2. Внесите изменения в конфигурационные файлы приложений.
 - a. Откройте файл appsettings.json, который находится в корневой папке приложения.
 - b. Отредактируйте блок "ForwardedHeaders":

```
{
    ...
    "ForwardedHeaders": {
        "Enable": true,
        "ForwardedForHeaderName": "X-Forwarded-For",
        "KnownProxiesIP": [доверенные IP-адреса]
    }
    ...
}
```

Где:

"Enable" — включение функции обработки Forwarded headers в веб-приложении;

"ForwardedForHeaderName" — имя заголовка, из которого будет получен IP-адрес;

"KnownProxiesIP" — список доверенных IP-адресов, при получении запроса от которых будет происходить обработка значения **"ForwardedHeader"**. Например, это может быть адрес балансировщика, reverse proxy и т. д. Если это значение не заполнено, то обработка

ForwardedHeader будет выполняться для всех IP-адресов, с которых приходят запросы.

с. Повторите шаги а-б для всех экземпляров приложения, которые входят в веб-ферму.

Пример

```
"KnownProxiesIP": ["127.0.0.1", "12.34.56.78", "2001:0db8:85a3:0000:0000:8a2e:0370:7334"]
```

Настроить вебсокеты

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Протокол вебсокетов (WebSockets) используется в Creatio для запуска пользовательских процессов, работы уведомлений, а также интеграции с телефонией.

Для корректной работы всех функций системы необходимо, чтобы вебсокеты были включены и настроены на сервере приложения.

На сервере приложений должна быть развернута **Windows Server 2012, Windows Server 2016** или **Windows 8/10** с установленными Internet Information Services (IIS) не ниже версии 8. Конфигурационные файлы в базовой версии Creatio по умолчанию настроены и необходимо только выполнить настройки со стороны сервера. Однако если вы разворачиваете и настраиваете Creatio впервые, рекомендуем проверить настройки конфигурационных файлов и убедиться, что веб-сокеты работают корректно.

Для использования защищенного соединения HTTPS требуется дополнительная настройка.

Важно. Если в вашей локальной сети используется прокси-сервер, то для него необходимо настроить протокол работы вебсокетов. Как правило, описание настройки данного протокола можно найти в документации прокси-сервера.

В данной статье описан **процесс настройки конфигурационных файлов Creatio** для корректной работы вебсокетов.

На заметку. Установка на сервер компонентов, обеспечивающих работу по протоколу WebSockets, подробно описана в статье [Проверить обязательные компоненты Windows](#).

Проверить настройки вебсокетов для Windows Server 2012 или Windows server 2016

Чтобы проверить настройку вебсокетов в Creatio, развернутом на сервере под управлением Windows Server 2012 или Windows Server 2016:

1. Убедитесь, что в файле Web.config, который находится в корневой директории сайта Creatio, отключено наследование, а также указаны ограничения на длину и время выполнения запроса.

```
<location path="." inheritInChildApplications="false">
  <system.web>
    ...
    <httpRuntime maxRequestLength="73400" executionTimeout="28800" targetFramework="4.7" />
  </system.web>
</location>
```

2. Для файла Web.config, который размещен в директории [Путь к корневому каталогу Creatio]\Terrasoft.WebApp\ проверьте, чтобы было указано значение типа wsService по умолчанию **“Terrasoft.Messaging.MicrosoftWSService.MicrosoftWSService, Terrasoft.Messaging.MicrosoftWSService”**, указан порт для подключения клиентов, ограничения на длину и время выполнения запроса HTTP, а также вызов дополнительных модулей.

Важно. Для Microsoft Windows Server версии 2012 и выше рекомендуется использовать значение “MicrosoftWSService”, а не “SuperWSService”.

На заметку. Значение portForClientConnection="0" означает использование порта web-приложения.

```
<wsService type="Terrasoft.Messaging.MicrosoftWSService.MicrosoftWSService,
Terrasoft.Messaging.MicrosoftWSService" encrypted="false" portForClientConnection="0" />
...
<location path="." inheritInChildApplications="false">
  <system.web>
    ...
    <httpRuntime maxRequestLength="102400" executionTimeout="28800"
targetFramework="4.6.2" />
    <httpHandlers>
      ...
      <add verb="GET" path="*ViewModule.aspx.ashx" type="Terrasoft.Messaging.MicrosoftWSSer
Terrasoft.Messaging.MicrosoftWSService" />
    </httpHandlers>
  </system.web>
  <system.webServer>
    ...
    <handlers>
      ...
      <add name="WSHandler" verb="*" path="*ViewModule.aspx.ashx"
type="Terrasoft.Messaging.MicrosoftWSService.WSHandler, Terrasoft.Messaging.MicrosoftWSService" />
    </handlers>
  </system.webServer>
</location>
```

На заметку. Подключение по протоколу WebSockets можно проверить при помощи консоли разработчика (Developer Tools) веб-браузера. В случае успешного подключения в консоли будет содержаться запись типа: WebSocket-connection opened for url:ws://demo.creatio.com/0/Nui/ViewModule.aspx.ashx

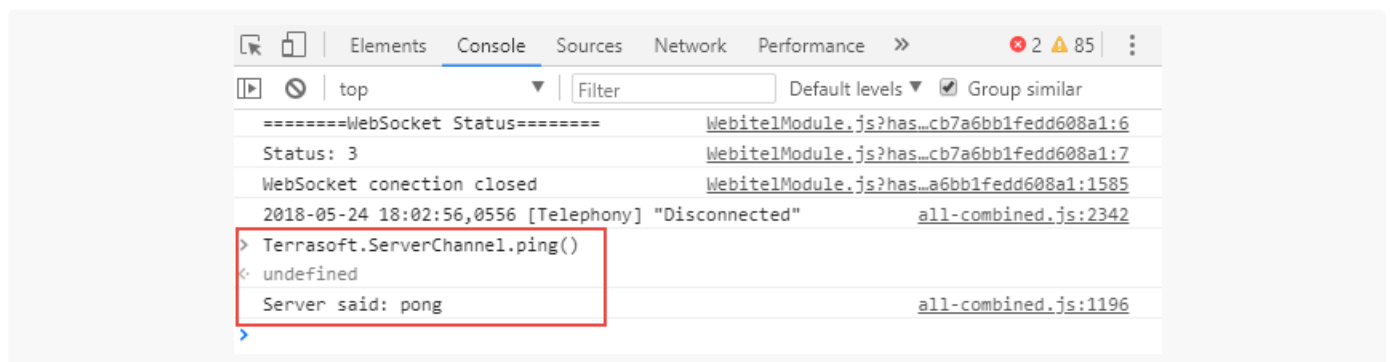
Часто задаваемые вопросы по настройке вебсокетов

Как проверить правильность настройки вебсокетов?

Существует несколько быстрых способов проверить корректность настройки вебсокетов:

- Воспользуйтесь функциональностью [импорта данных из Excel](#). Если вебсокеты настроены правильно, то импорт данных будет выполнен.
- В консоли разработчика веб-браузера введите команду `Terrasoft.ServerChannel.ping()`. При корректной настройке вебсокетов сервер ответит `pong` (Рис. 1). При любом другом ответе сервера необходимо проверить правильность настройки вебсокетов.

Рис. 1 — Проверка настройки вебсокетов при помощи консоли разработчика веб-браузера



- Запустите ручную бизнес-процесс, в котором есть стартовый таймер и автогенерируемая страница. Если автогенерируемая страница откроется, значит, вебсокеты настроены правильно.

На заметку. Элементы бизнес-процессов подробно описаны в [отдельном блоке](#).

Я настроил вебсокеты, но они не работают. Почему?

Если после настройки вебсокеты не работают, проверьте:

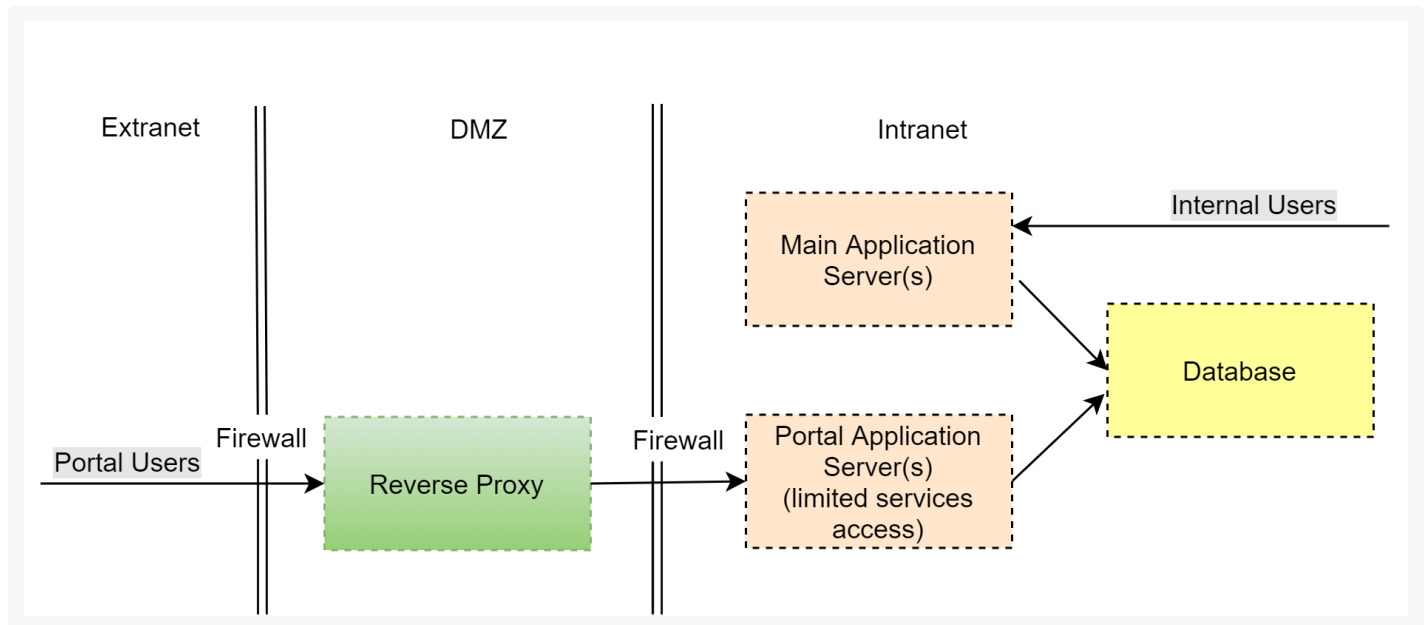
- Установлены ли на сервере все компоненты протокола WebSockets. [Подробнее >>>](#)
- Установлен ли протокол использования вебсокетов для прокси-сервера, если он используется в вашей локальной сети.
- Настройки антивируса и Firewall. Если нет возможности отключить на сервере эти программы, добавьте IP-адрес и порт вашего сайта Creatio в список исключений для входящих и исходящих потоков.
- Наличие расширений и надстроек web-браузера, в том числе VPN. Расширения могут блокировать работу вебсокетов.

Настроить безопасный доступ к portalу

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Для обеспечения безопасности данных при установке портала on-site приложение должно быть развернуто в режиме веб-фермы. Подробно пример настройки веб-фермы рассмотрен в статье [“Настроить горизонтальное масштабирование”](#). Доступ к portalу настраивается по схеме (Рис. 1):

Рис. 1 — Типовая схема установки системы с доступом к portalу из внешней сети



Демилитаризованная зона (DMZ)

- В демилитаризованной зоне публикуется только обратный прокси-сервер (reverse proxy).
- На уровне reverse proxy выполняется первичный мониторинг сетевой активности. Также здесь настраивается ограничение на доступ к конфигурационным веб-сервисам приложения.
- Авторизированные пользователи портала имеют доступ только к тем конфигурационным веб-сервисам, к которым он явно разрешен на уровне приложения.
- При разработке проектного решения выполняются настройки доступа для новых веб-сервисов. Подробно эта настройка описана в документации по разработке, статья [“Ограничение доступа к веб-сервисам для пользователей портала”](#).

Внутренняя сеть (Intranet)

- Для обслуживания пользователей портала в веб-ферме выделяется отдельный набор узлов приложений, который не пересекается с узлами приложений для обслуживания внутренних пользователей.
- Для работы приложения портала и приложения пользователей создаются отдельные учетные записи в базе данных с различным набором прав доступа.
- В настройках приложений портала блокируется возможность входа для пользователей системы (отключаются AuthProviders, кроме пользователей портала). Это необходимо, чтобы из внешней сети (Extranet) можно было создать сессии только пользователям портала.
- Дополнительно можно настроить использование внешних провайдеров идентификации для добавления второго шага проверки при авторизации.
- Узлы приложений портала, СУБД и приложения для пользователей размещаются в отдельных

сегментах с ограниченным доступом.

Перевести Creatio с HTTP на HTTPS

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Основная задача протокола HTTPS — обеспечение защищенного соединения между клиентом и веб-сервисом. Перевод Creatio с HTTP на HTTPS повысит безопасность системы и позволит использовать дополнительные сервисы, например, `webitel` с поддержкой WebRTC. Обратите внимание, что настройка потребуется, только если приложение развернуто on-site. Для перехода на HTTPS нужно изменить некоторые параметры веб-сайта в IIS и отредактировать конфигурационный файл `Web.config`. Пользователи cloud по умолчанию работают через защищенное соединение.

На заметку. Вы не сможете использовать преимущества HTTPS, если приложение Creatio развернуто на Windows Server 2008.

Настроить IIS

Прежде чем приступить к настройке HTTPS, вам необходимо:

- Получить цифровой сертификат от центра сертификации в формате PFX;

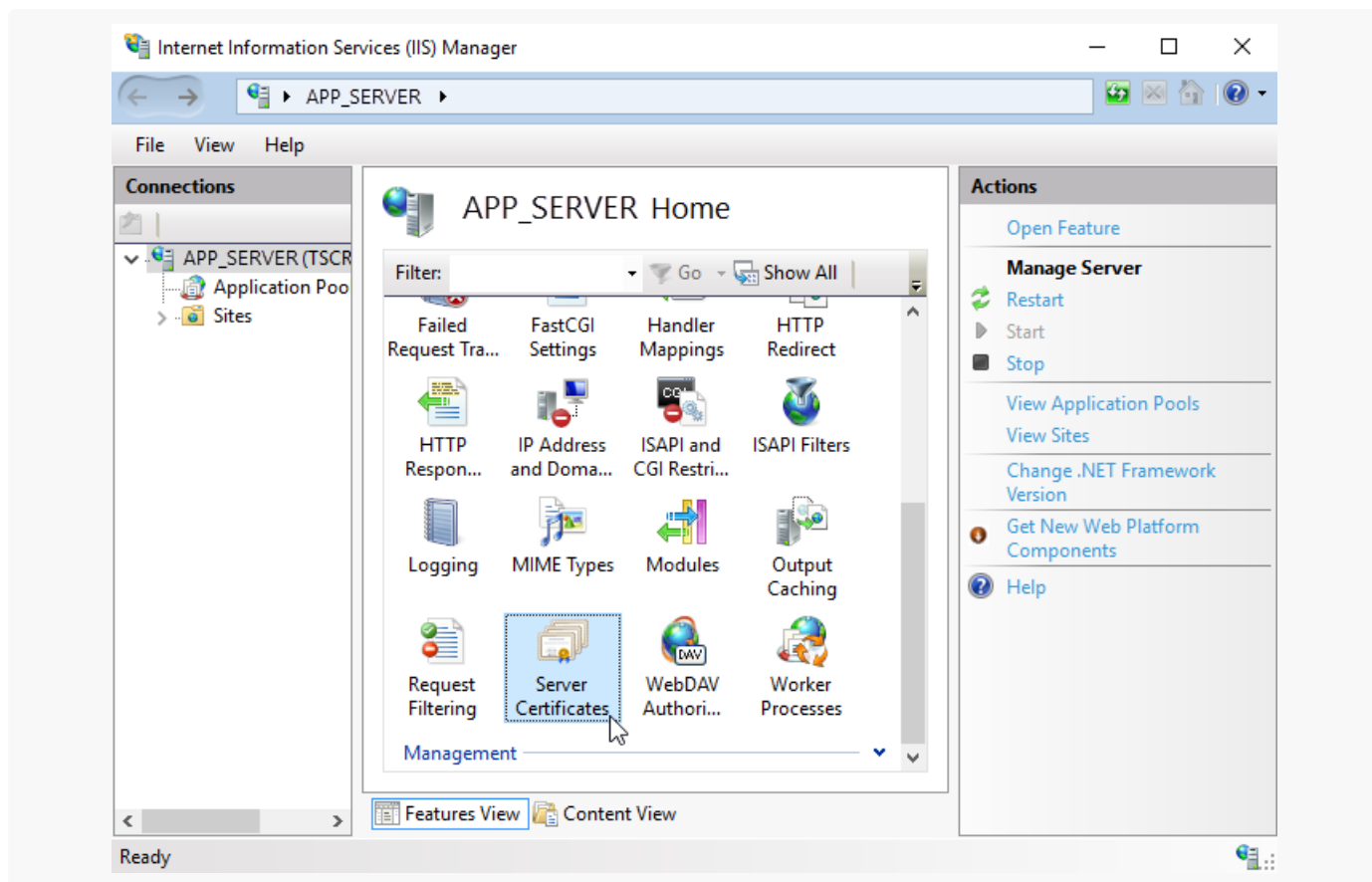
На заметку. При использовании самоподписанного сертификата подключение мобильного приложения к сайту Creatio будет недоступно из-за ограничений политики безопасности мобильных приложений.

- Произвести [настройки вебсокетов](#) для корректной работы всех компонентов системы;
- Рекомендуем также проверить список установленных [компонентов IIS](#), чтобы избежать ошибок в процессе настройки и работы с Creatio.

Полученный цифровой сертификат необходимо загрузить в хранилище сертификатов сервера:

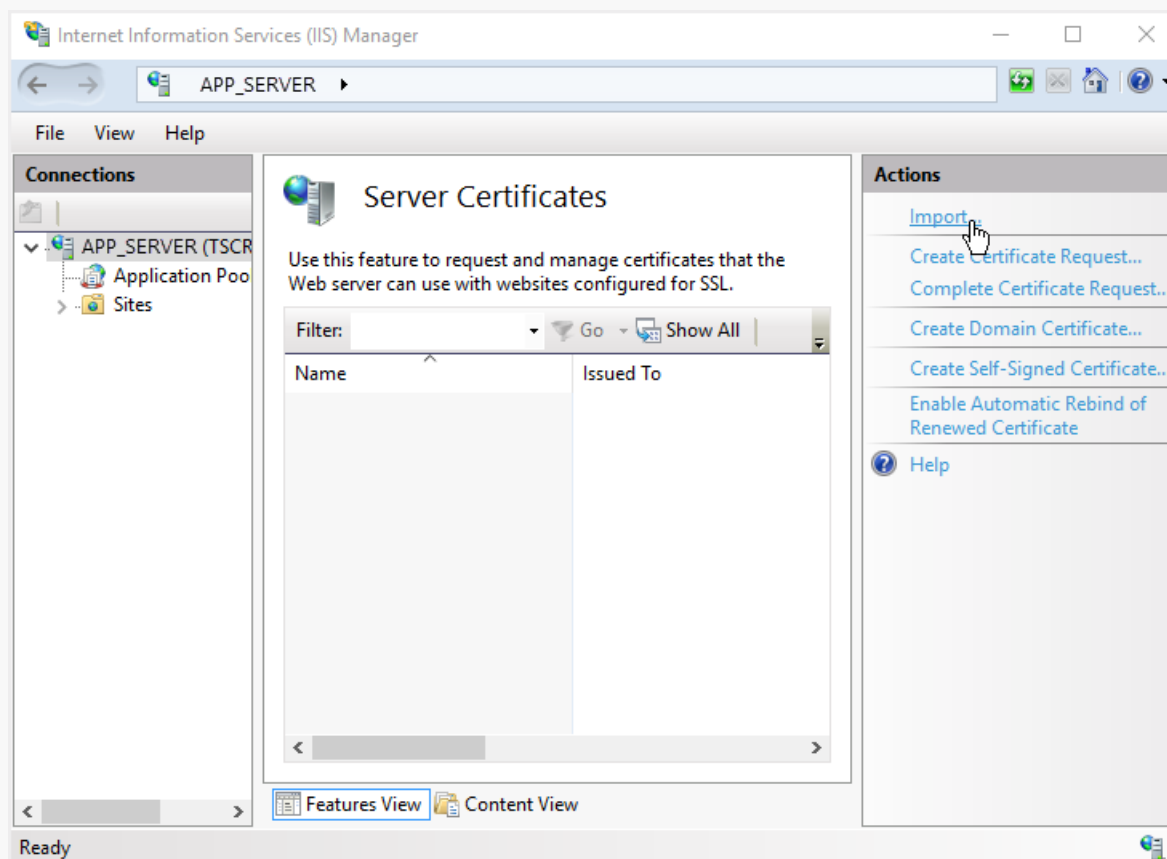
1. Откройте Internet Information Services (IIS) Manager.
2. В основном окне IIS выберите деталь [*Server Certificates*] двойным кликом мыши ([Рис. 1](#)).

Рис. 1 — Выбор детали [*Server Certificates*]



3. В окне [*Server Certificates*] нажмите на ссылку [*Import*] в меню действий справа (Рис. 2).

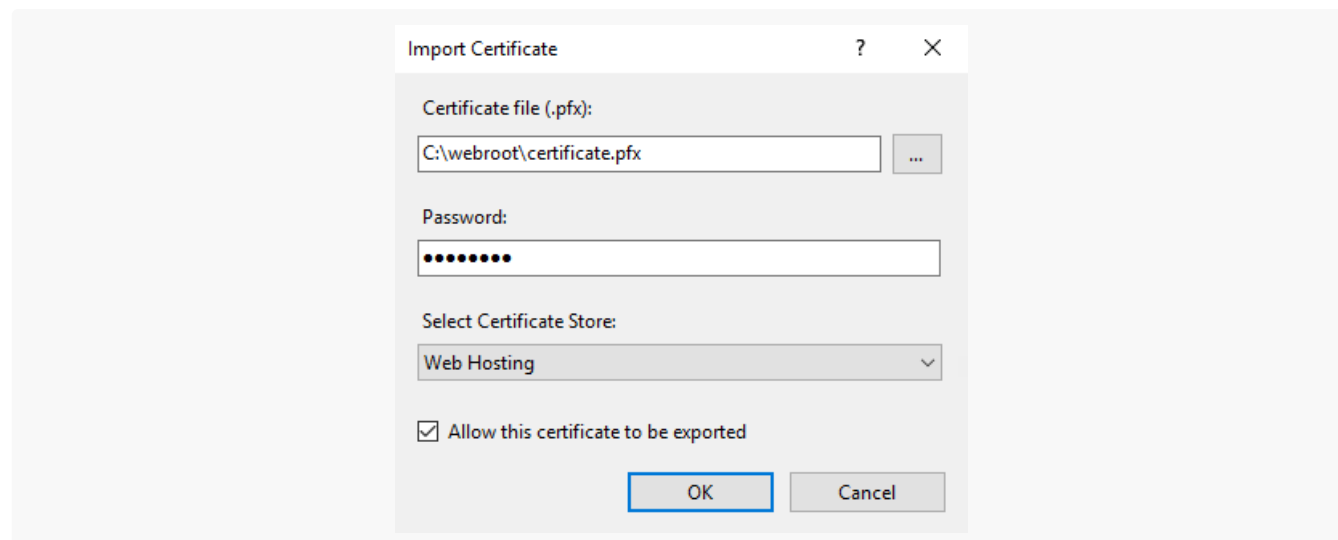
Рис. 2 — Открытие окна [*Import*]



4. В диалоговом окне импорта укажите:

- a. Путь к размещенному на сервере файлу импорта;
- b. Пароль (при необходимости);
- c. Хранилище сертификатов ([Рис. 3](#)).

Рис. 3 — Окно импорта сертификата

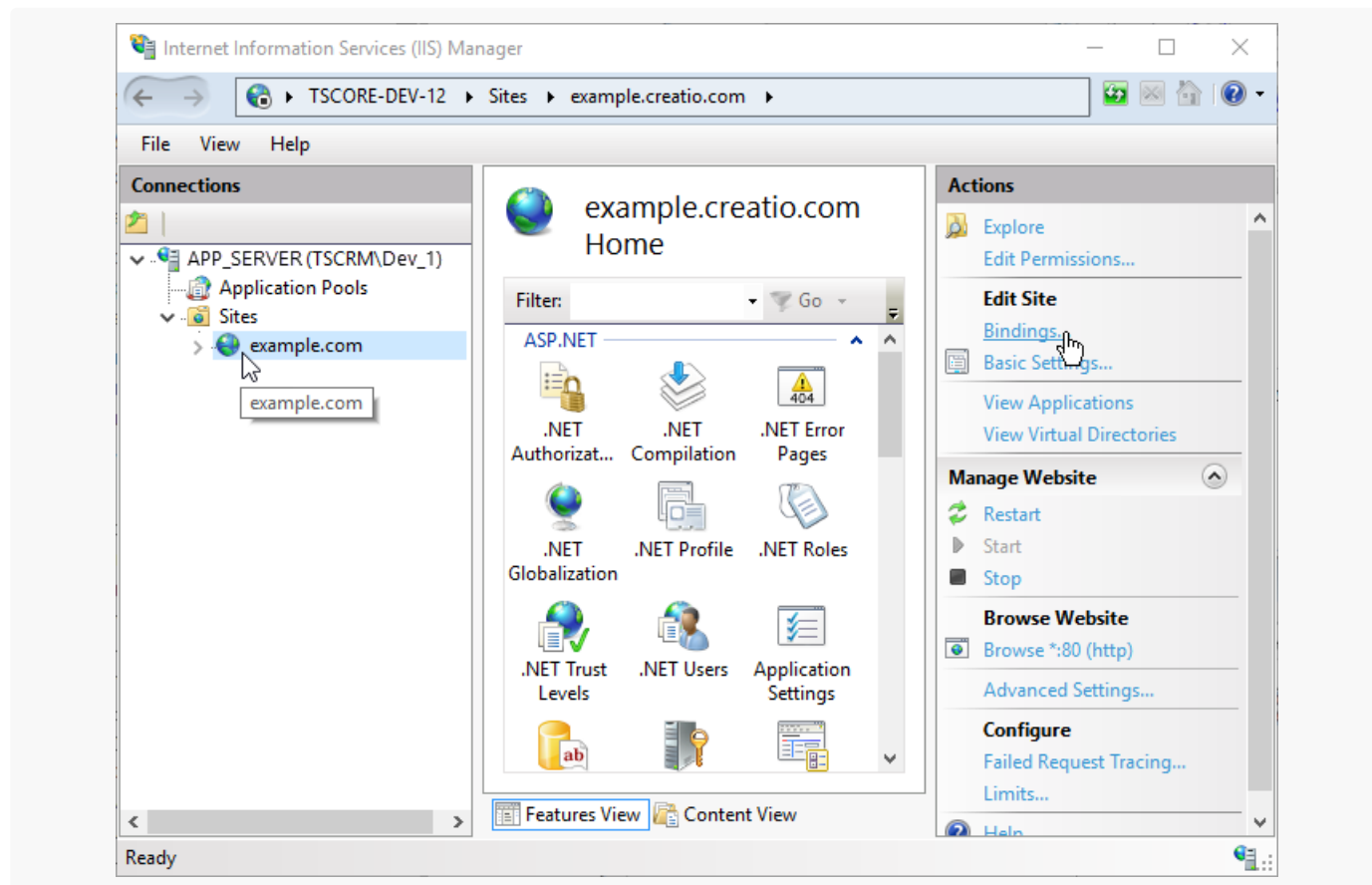


5. Нажмите на кнопку [OK], чтобы импортировать сертификат.

Далее следует привязать импортированный сертификат к приложению Creatio:

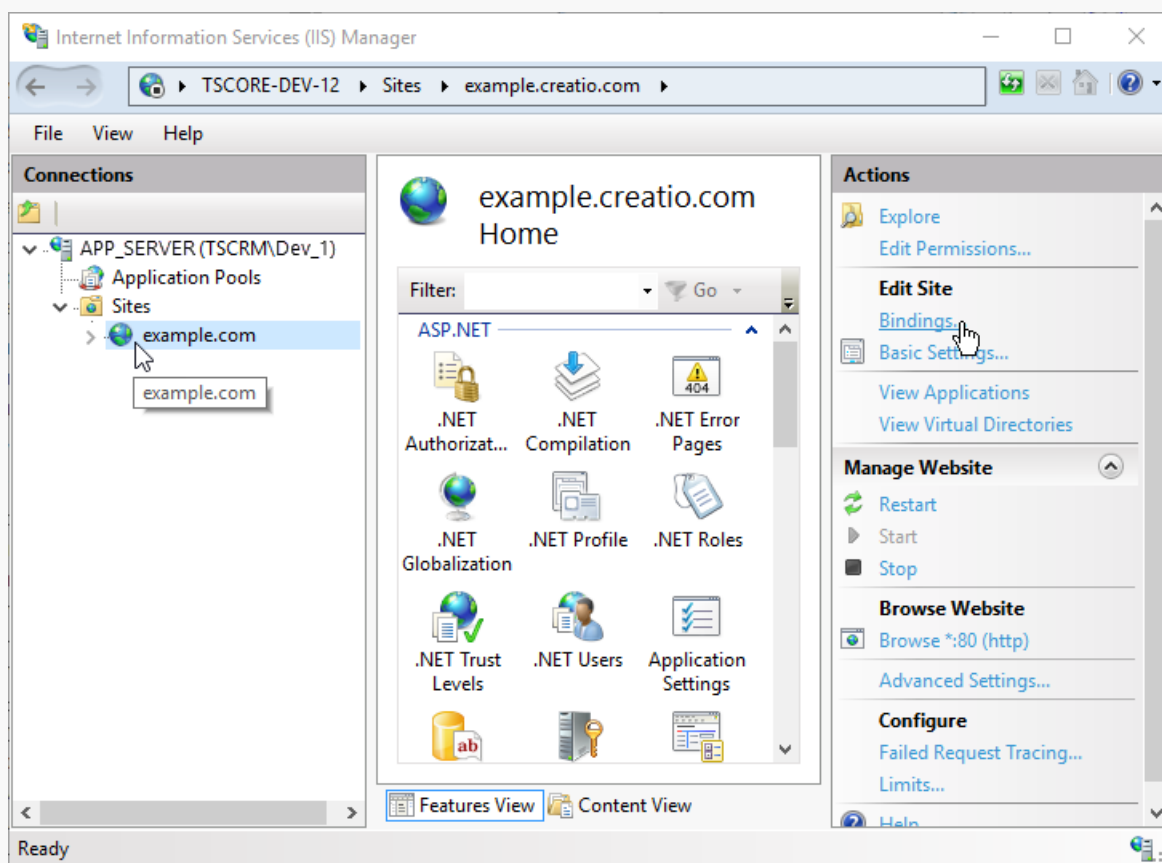
1. В окне IIS перейдите к сайту приложения, нажав на его название в левом меню [*Connections*] (Рис. 4).

Рис. 4 — Выбор сайта Creatio в окне IIS



2. Нажмите на ссылку [*Bindings*] в меню действий (Рис. 5).

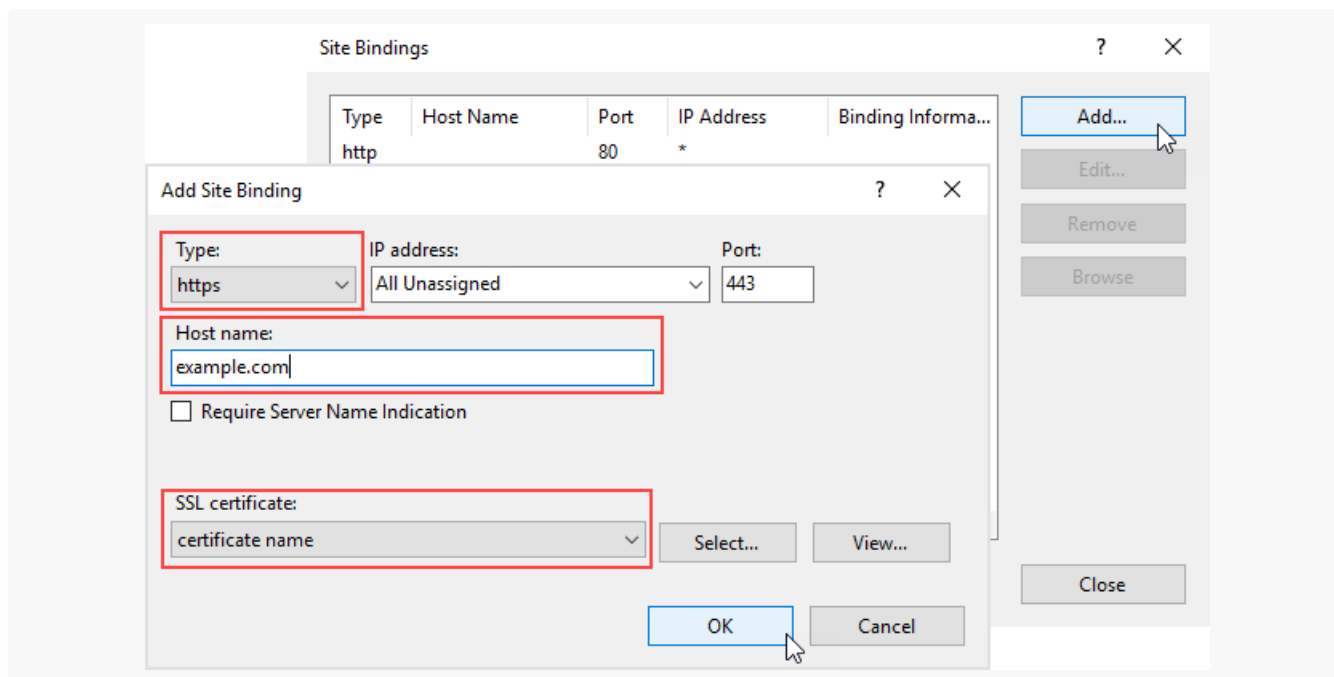
Рис. 5 — Выбор связей сайта



3. В меню связей сайта нажмите на кнопку [*Add*] и добавьте новую связь. В окне [*Add Site Binding*] укажите:

- a. Тип "https ";
- b. Адрес сайта;
- c. Электронный сертификат SSL ([Рис. 6](#)).

Рис. 6 — Привязка сертификата к сайту Creatio



4. Нажмите [OK], чтобы подтвердить настройки.

Теперь сертификат успешно привязан к веб-приложению.

Настроить Web.config

После добавления сертификата необходимо внести изменения в конфигурационный файл Web.config, который размещен в **корневой директории** сайта Creatio:

1. В конце файла найдите следующие строки:

```
<behaviors configSource="Terrasoft.WebApp\ServiceModel\http\behaviors.config" />
<bindings configSource="Terrasoft.WebApp\ServiceModel\http\bindings.config" />
```

2. Замените пути "http" на "https", чтобы получилось:

```
<behaviors configSource="Terrasoft.WebApp\ServiceModel\https\behaviors.config" />
<bindings configSource="Terrasoft.WebApp\ServiceModel\https\bindings.config" />
```

Далее необходимо отредактировать файл Web.config, который размещен в директории **Путь к корневому каталогу сайта\Terrasoft.WebApp**:

1. Задайте значение переменной encrypted="true". Настройка отличается в зависимости от операционной системы сервера с приложением Creatio.

Для Windows Server 2012 и новее конфигурационную строку нужно привести к виду:

```
<wsService
  type="Terrasoft.Messaging.MicrosoftWSService.MicrosoftWSService,
```

```
Terrasoft.Messaging.MicrosoftWSService" encrypted="true"
portForClientConnection="443" maxConnectionNumber="100"
clearIdleSession="false" clearIdleSessionInterval="120" />
```

2. В конце файла найдите строчку:

```
<services configSource="ServiceModel\http\services.config" />
```

3. Измените путь с "http" на "https", чтобы получилось:

```
<services configSource="ServiceModel\https\services.config" />
```

После всех изменений не забудьте сохранить файлы конфигураций.

Перезапустите приложение при помощи IIS, а затем зайдите на ваш сайт Creatio. Если все действия выполнены верно, то в строке адреса вы увидите "https://" перед веб-адресом приложения.

Настроить систему управления версиями для среды разработки

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Управление версиями необходимо при развертывании среды разработки, чтобы при командной работе все участники могли вносить, отслеживать и объединять изменения, выполненные в конфигурации Creatio. Система управления версиями в Creatio выполняет следующие функции:

- перенос изменений между конфигурациями;
- хранение различных версий конфигурационных схем;
- отмена изменений возвращением к одной из предыдущих версий.

Creatio поддерживает интеграцию с системой контроля версий Subversion (SVN) 1.7 и выше. Подробнее об использовании SVN читайте в [документации продукта](#).

На заметку. Встроенные инструменты разработки Creatio совместимы только с системой контроля версий Subversion. Однако вы можете отключить интеграцию с SVN и использовать любую систему контроля версий, включая Git, в режиме разработки файловой системе. Подробнее о работе с Git читайте в статье “Особенности работы с Git” документации по разработке.

Хранилище SVN должно быть единственной точкой соприкосновения для различных сред разработки. В противном случае среда разработки каждого сотрудника должна быть изолирована и работать на независимом сервере приложений, подключенном к базе данных, которая не используется другими экземплярами приложений Creatio.

Подробнее о настройке среды разработки читайте в статье “[Организация среды разработки](#)”

документации по разработке.

В общем случае для настройки и подключения SVN вам необходимо:

- [Установить SVN и создать хранилище для Creatio](#)
- [Подключить созданное хранилище к Creatio](#)

Установить SVN и создать хранилище для Creatio

Для установки системы контроля версий:

1. Установить сервер SVN

Вы можете установить систему контроля версий на сервере приложения, сервере баз данных или на отдельном сервере.

Чтобы развернуть сервер SVN на операционной системе Windows, воспользуйтесь одним из общедоступных установщиков:

- [VisualSVN](#)
- [CollabNet](#)

Инструкции по развертыванию SVN на других операционных системах, включая Debian, доступны с [Apache Subversion](#).

Сервер SVN может работать независимо или использовать веб-сервер Apache в качестве внешнего интерфейса (утилиты VisualSVN и CollabNet могут установить его как компонент).

Если сервер SVN работает независимо, то доступ к хранилищам предоставляется по протоколу **SVN**. Если в качестве внешнего интерфейса используется веб-сервер, то доступ к хранилищам предоставляется через протоколы **HTTP** и **HTTPS**.

Рекомендуем для интеграции с Creatio установить веб-сервер в качестве внешнего интерфейса и использовать протоколы **HTTP** и **HTTPS**.

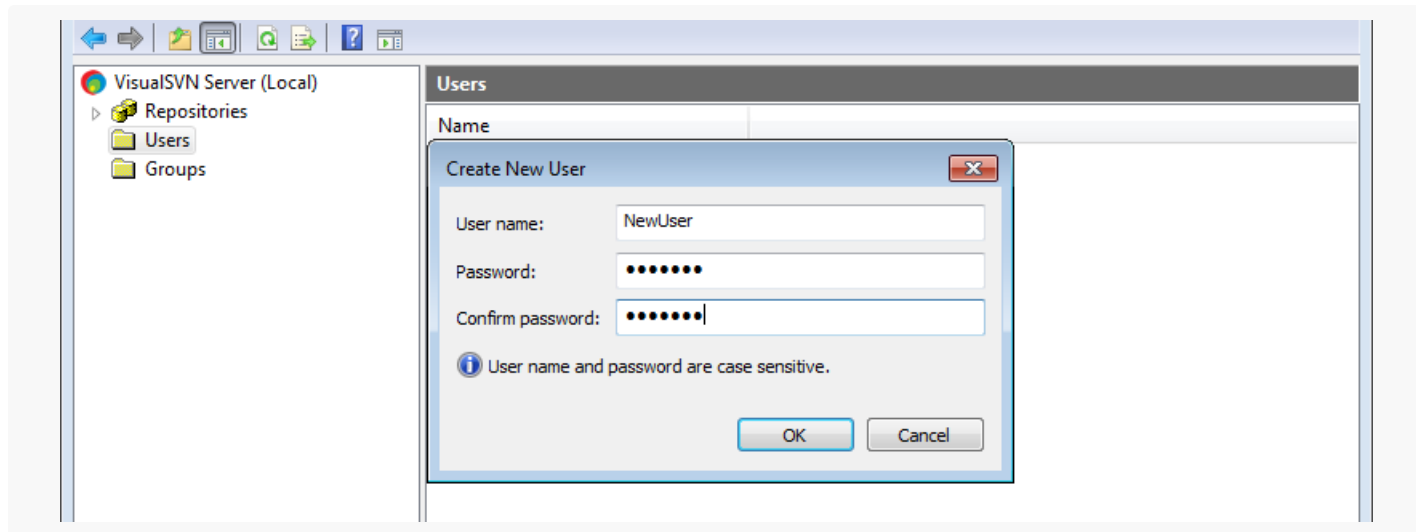
2. Создать пользователя сервера SVN

Для доступа к серверу SVN вам необходимо создать как минимум одного пользователя. Рекомендуется создавать отдельных пользователей для каждого из разработчиков, которые будут работать с системой контроля версий.

Для создания пользователя сервера SVN вы можете воспользоваться стандартными инструментами SVN, которые входят в установочный пакет, например VisualSVN ([Рис. 1](#)).

Для работы с хранилищами Creatio требуется использовать аутентификацию пользователей через логин и пароль.

Рис. 1 — Создание пользователя сервера SVN при помощи утилиты VisualSVN



3. Создать хранилище на сервере SVN

Создайте хранилище SVN при помощи стандартных инструментов, которые входят в установочный пакет сервера SVN, например, VisualSVN и CollabNet.

На заметку. Creatio поддерживает одновременную работу нескольких хранилищ, которые могут быть расположены на разных серверах SVN.

4. Установить клиент SVN (опционально)

При желании вы можете установить на рабочем месте разработчика клиент SVN, например, [TortoiseSVN](#).

На заметку. Рекомендуется использование клиента TortoiseSVN версии 1.8 и выше.

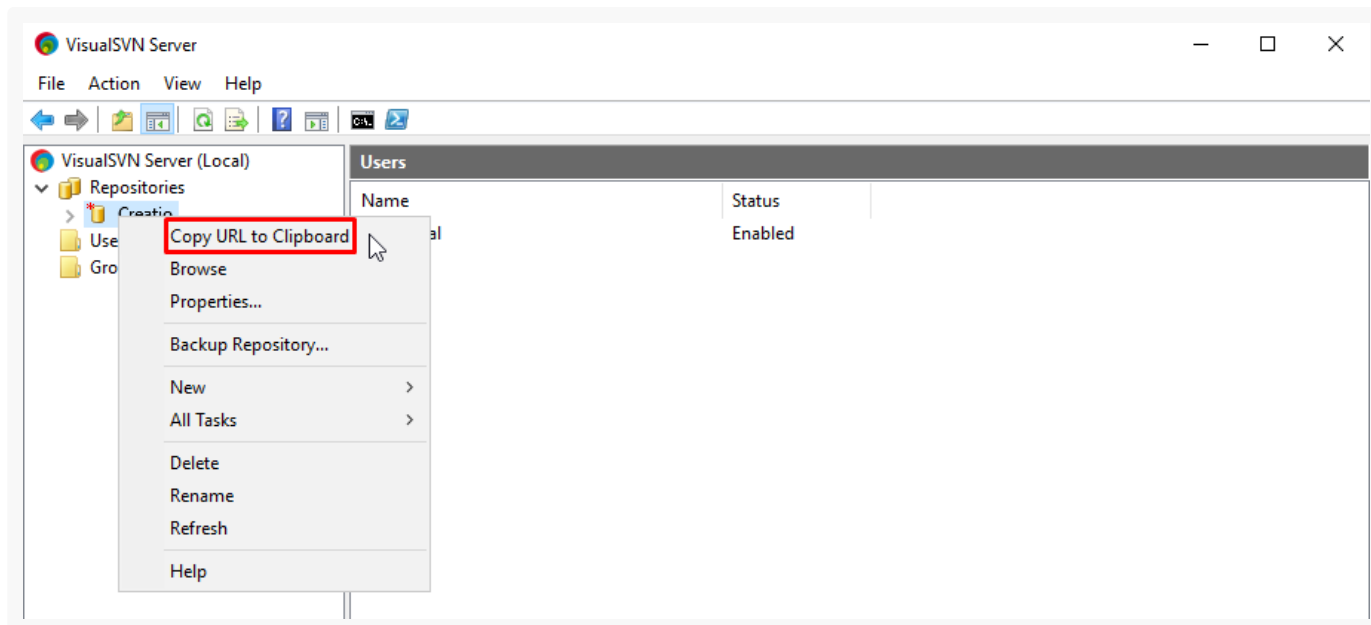
Установка клиента SVN не является обязательной, так как это не влияет на работу Creatio. Клиент SVN удобен для просмотра локальной рабочей копии, истории, возврата к предыдущим версиям, пересмотров и т. д.

Подключить созданное хранилище к Creatio

Для подключения хранилища к Creatio:

1. Скопируйте URL-адрес вашего хранилища. Например, в VisualSVN для этого нужно кликнуть правой кнопкой мыши по хранилищу и в контекстном меню выбрать команду [*Copy URL to clipboard*] (Скопировать URL в буфер обмена) ([Рис. 1](#)).

Рис. 1 — Копирование URL-адреса хранилища




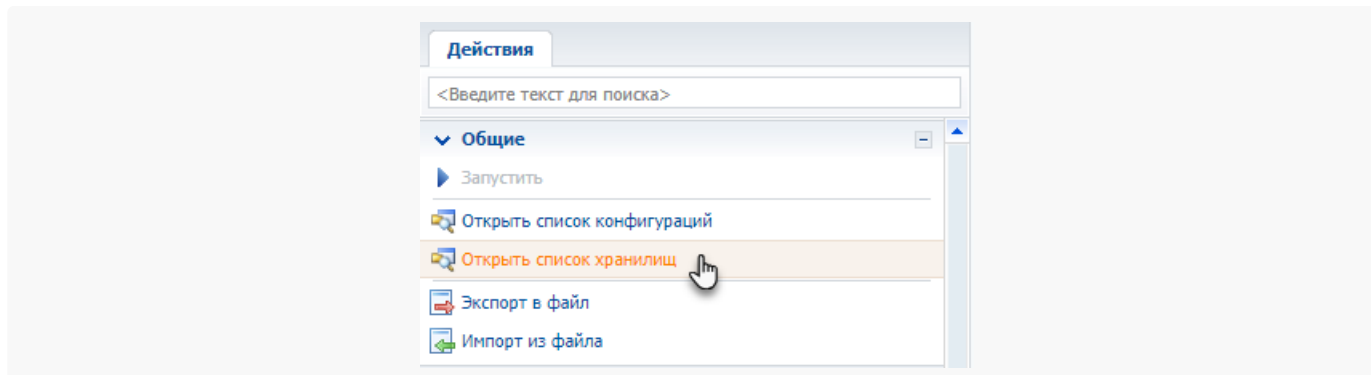
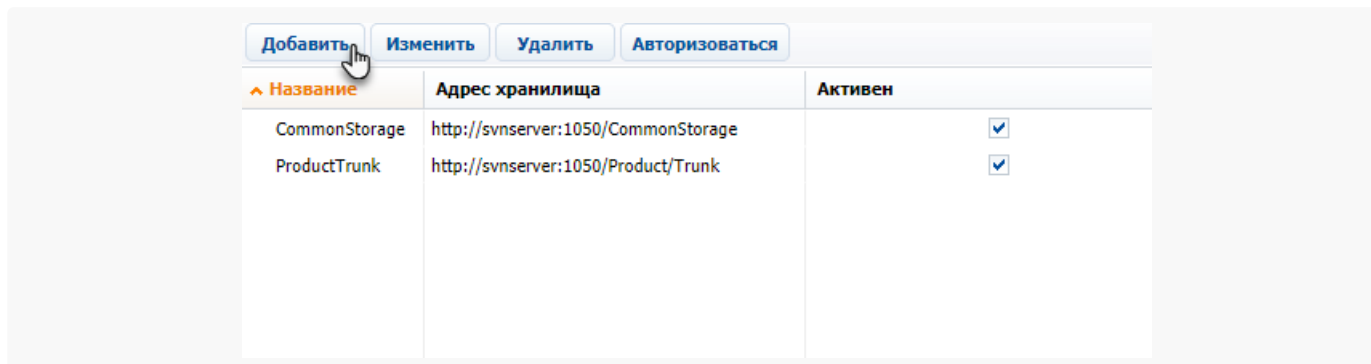
2. По кнопке  в основном приложении Creatio перейдите в дизайнер системы,
3. В группе [*Конфигурирование разработчиком*] перейдите в раздел [*Управление конфигурацией*].
4. На панели действий нажмите [*Открыть список хранилищ*] ([Рис. 2](#)).

Рис. 2 — Открытие списка хранилищ SVN



5. Нажмите кнопку [*Добавить*] ([Рис. 3](#)). Откроется страница свойств нового хранилища.

Рис. 3 — Добавление нового хранилища к списку хранилищ системы контроля версий



6. На открывшейся странице укажите свойства нового хранилища ([Рис. 4](#)):

- a. [*Название*] — название хранилища, которое отобразится в реестре подключенных хранилищ.
- b. [*Адрес хранилища*] — сетевой адрес существующего хранилища SVN. Укажите адрес, который вы скопировали на шаге 1.
- c. Протокол HTTP (стандартный сетевой протокол), протокол HTTPS (стандартный сетевой протокол, защищенный шифрованием SSL) и протокол SVN (собственный сетевой протокол системы Subversion) поддерживаются при адресации хранилища.
- d. [*Активен*] — установите этот признак, чтобы разрешить использование хранилища в системных операциях. Для каждого нового хранилища данный признак по умолчанию установлен.

На заметку. Вы можете работать только с активными хранилищами. Также все хранилища, из которых планируется обновлять пакеты, должны быть активны. К ним относятся хранилище, из которого обновляется исходный пакет, и хранилища, из которых обновляются все пакеты зависимости исходного пакета.

Рис. 4 — Заполнение свойств нового хранилища

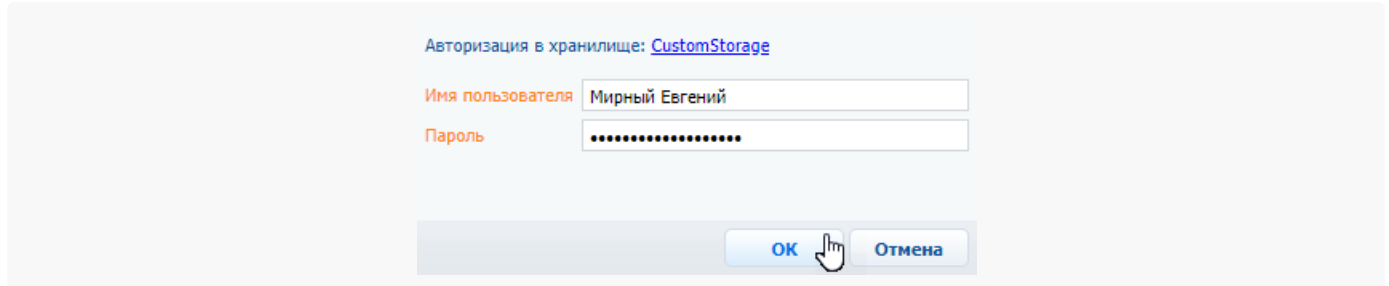
7. Выберите добавленное хранилище в списке и нажмите кнопку [*Авторизоваться*] ([Рис. 5](#)).

Рис. 5 — Аутентификация хранилища

Название	Адрес хранилища	Активен
CommonStorage	http://svnserver:1050/CommonStorage	<input checked="" type="checkbox"/>
CustomStorage	http://svnserver:1050/CustomStorage	<input checked="" type="checkbox"/>
ProductTrunk	http://svnserver:1050/Product/Trunk	<input checked="" type="checkbox"/>

8. Подключитесь к хранилищу, используя учетные данные одного из созданных пользователей сервера SVN ([Рис. 6](#)).

Рис. 6 — Ввод учетных данных пользователя сервера SVN



В результате ваше хранилище SVN будет подключено к Creatio. Используйте новое хранилище для создания пользовательских пакетов и установки созданных пакетов в рабочее пространство.

Подробнее о [работе с пакетами при помощи SVN](#), [переносе изменений через SVN](#) и [работе с SVN](#) читайте в документации по разработке Creatio.

Часто задаваемые вопросы по установке Creatio

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Какие компоненты IIS (Internet Information Services) нужны для работы приложения Creatio?

Для работы приложения Creatio, развернутого on-site, в меню Windows [*Programs and Features*] необходимо включить следующие компоненты:

1. .NET Framework 3.5:
 - a. Windows Communication Foundation Non-HTTP Activation;
 - b. Windows Communication Foundation HTTP Activation;
2. .NET Framework 4.7.2:
 - a. ASP.NET 4.7.2;
 - b. Для WCF Services:
 - HTTP Activation;
 - Message Queuing (MSMQ) Activation;
 - Named Pipe Activation;
 - TCP Activation;
 - TCP Port Sharing.

На заметку. Microsoft .NET Framework 4.7 и выше — для версий Creatio 7.11.1 - 7.13.1; Microsoft .NET Framework 4.7.2 — для версий Creatio 7.13.2 и выше.

Кроме того, сервисы IIS (Internet Information Services) являются ключевым компонентом для работы веб-сайтов и веб-приложений, развернутых на Windows Server. Поэтому для работы Creatio требуется

включить следующий список компонентов IIS:

1. На вкладке "Web Management Tools":
 - a. IIS Management Console;
 - b. IIS Management Script and Tools;
 - c. IIS Management Service.
2. На вкладке "World Wide Web Services" включите следующие элементы:
 - a. Для компонента Application Development Features:
 - Все элементы ASP.NET;
 - Все элементы .NET Extensibility;
 - ISAPI Extensions;
 - ISAPI Filters;
 - WebSocket Protocol.
 - g. Для компонента Common HTTP Features:
 - Default Document;
 - HTTP Errors;
 - HTTP Redirection;
 - Static Content.
 - l. Для компонента "Health and Diagnostics":
 - Custom Logging;
 - HTTP Logging;
 - Logging Tools;
 - Request Monitor;
 - Для компонента "Security":
 - Request Filtering;
 - IP and Domain Restriction.

Как перевести приложение с протокола HTTP на HTTPS?

Все необходимые шаги и настройки для перехода с HTTP на HTTPS подробно описаны в статье [Перевести Creatio с HTTP на HTTPS](#).

Под каким пользователем выполняется первый вход в систему?

После успешного развертывания Creatio on-site первый вход в систему выполняется под пользователем Supervisor с паролем Supervisor.

Влияет ли число активных пользователей Creatio на число пользователей Microsoft SQL Server?

Количество пользователей Microsoft SQL Server не зависит от количества пользователей Creatio, но зависит от количества серверов с базами данных. Обратите внимание на [системные требования](#) при развертывании приложения on-site.

Настроить доступ к сервису чатов

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Для работы с каналами чатов Facebook Messenger и WhatsApp в приложениях, развернутых on-site, необходимо:

- Перевести Creatio с http на https. Подробно эта настройка описана в статье [Перевести Creatio с HTTP на HTTPS](#).
- Настроить на сервере приложения доступ к сервису чатов по адресу <https://sm-account.creatio.com/>.
- Настроить на сервере приложения для сервиса чатов sm-account.creatio.com возможность входящего подключения по протоколу https и защиту валидным сертификатом.

Для работы с каналами чатов Telegram сервер приложений должен иметь доступ в Интернет.

Для приложений Creatio, использующих двухэтапную аутентификацию, необходимо настроить доступ на входящие запросы из внешних сетей для сервисов FacebookOmnichannelMessagingService, TelegramOmnichannelMessagingService, WhatsappOmnichannelMessagingService.

Продукты Creatio на платформе .NET Core

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Платформа .NET Core является открытым кроссплатформенным программным обеспечением и поддерживает развертывание на операционных системах **Linux, Windows, Mac OS**.

Для развертывания продуктов Creatio на платформе .NET Core рекомендуется использовать операционную систему **Linux**, которая отличается высокой надежностью, производительностью, оптимальной стоимостью и динамично развивается.

Реализация поддержки .NET Core в продуктах Creatio

Функциональность	Поддержка на .NET Core
Windows-аутентификация	Реализована поддержка в версии 7.16.4.
Разработка конфигурации (раздел [<i>Конфигурация</i>], дизайнер объектов)	Реализована поддержка в версии 7.17.0.
Интеграция с LDAP	Реализована поддержка в версии 7.17.2.
Отказоустойчивая конфигурация Redis Sentinel	Не планируется поддержка. В версии 7.18.0 реализована поддержка более современной отказоустойчивой конфигурации Redis Cluster .
Интеграция с телефонией	Реализована поддержка коннектора Asterisk в версии 7.16.3. Особенности интеграции с другими поддерживаемыми коннекторами к телефонии описаны примечании ниже. Планируется интеграция с Cisco Finesse без использования IIS и ARR в будущих релизах.
СУБД Oracle	Планируется поддержка в будущих релизах.
Синхронизация календарей и контактов Exchange\Office365	Реализована поддержка в версии 7.18.2.
Синхронизация контактов и календаря Google	Планируется поддержка в будущих релизах.
Интеграция с Facebook	Планируется поддержка в будущих релизах.
Лидогенерация из социальных сетей (Facebook, LinkedIn)	Планируется поддержка в будущих релизах.

Примечание

Для интеграции с ATC Avaya, TAPI, Infinity X необходимо сервис сообщений (Creatio Messaging Service) располагать на Windows.

Для интеграции с Cisco Finesse необходимо расположить на Windows веб-сервер Microsoft IIS и расширение для него — Application Request Routing (ARR).

Жизненный цикл продуктов на платформе .NET Framework и .NET Core

Компания Microsoft представила релиз платформы **.NET 5**, таким образом реализовав интеграцию

платформ .NET Framework и .NET Core.

Благодаря этому поддерживается максимальное количество API, которое ранее было доступно на платформе .NET Framework. Важно отметить, что API при этом не поддерживает обратную совместимость, поэтому для поддержки функциональности .NET Core и .NET 5 необходимо адаптировать функциональность, разработанную на .NET Framework под .NET Core.

В будущем планируется перевести продуктовую линейку Creatio на единую платформу .NET.

Разработка функциональности с одновременной поддержкой .NET Framework и .NET Core

Чтобы упростить адаптацию под .NET Core и NET 5, рекомендуется новую функциональность разрабатывать так, чтобы она поддерживалась и платформой .NET Framework, и платформой .NET Core одновременно.

Рекомендации:

1. Внешние библиотеки, которые будут использованы, должны поддерживать .NET Standard 2.0. Это позволит их использовать и в .NET Framework, и в .NET Core.
2. API Framework-а, который будет использоваться, также должен поддерживать .NET Standard 2.0. Проверять соответствие можно с помощью [документации Microsoft](#).
3. При написании конфигурационных web-сервисов необходимо наследоваться от Terrasoft.Web.Common.BaseService и использовать HttpContextAccessor для доступа к HttpContext. Подробнее: [Пользовательские веб-сервисы](#).

Важно. Если вы используете или планируете использовать приложения Creatio Marketplace для расширения функциональности, то необходимо уточнять, поддерживают ли они продукты .NET Core.

Миграция с приложения .NET Framework на приложение .NET Core

Возможность миграции приложения с .Net Framework на .NET Core будет реализована ближайших релизах.

Развертывание приложения на платформе .NET Core

Инструкция по развертыванию приложения доступна в статье [Установить Creatio .NET Core на Linux](#).

Настроить авторизацию интегрированных приложений по протоколу OAuth 2.0

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

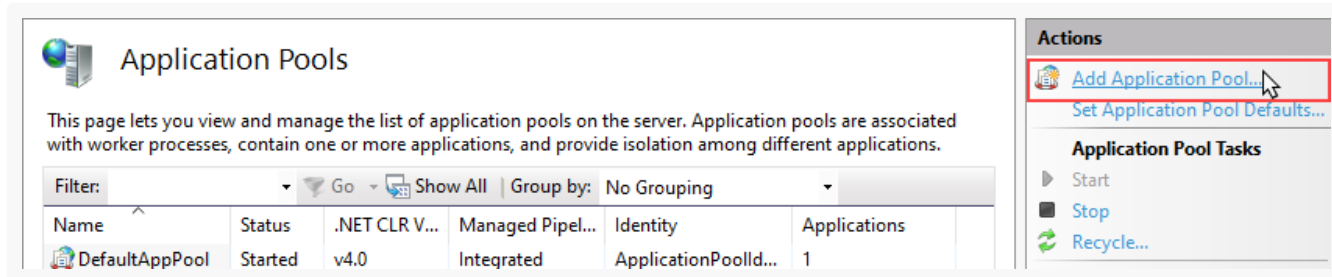
Вы можете настроить для приложений и веб-сервисов, интегрированных с Creatio, безопасную авторизацию по протоколу OAuth 2.0. Эта технология позволяет не передавать сторонним приложениям логин и пароль Creatio, а также ограничить права внешних приложений на действия в Creatio.

Установить и настроить Identity Service

Установка и настройка Identity Service выполняется после развертывания сервера базы данных и сервера приложения Creatio. Чтобы установить Identity Service:

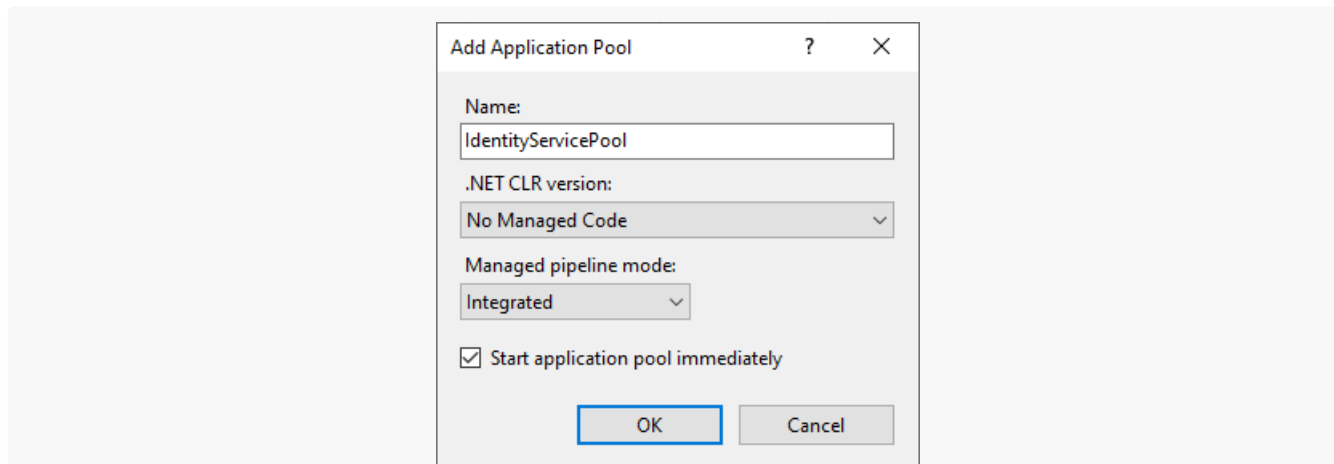
1. Перейдите на сервер приложения.
2. Установите .NET Core runtime 2.2. [Скачать установочный файл](#)
3. Установить .NET Core Hosting Bundle. [Скачать установочный файл](#)
4. Перезапустите IIS.
5. В папке с установочными файлами Creatio найдите архив **IdentityService.zip** и распакуйте его.
6. Добавьте в IIS новый **пул** приложения для Identity Service.
 - a. В области [*Connections*] окна управления IIS перейдите в секцию [*Applications Pools*].
 - b. В области [*Actions*] выберите действие [*Add Application Pool...*].

Рис. 1. — Добавление пула в IIS



- c. В окне настройки пула укажите его название, например, "IdentityServicePool". В поле [*.NET CLR Version*] укажите значение "No Managed Code".

Рис. 2 — Пример настройки пула Identity Service



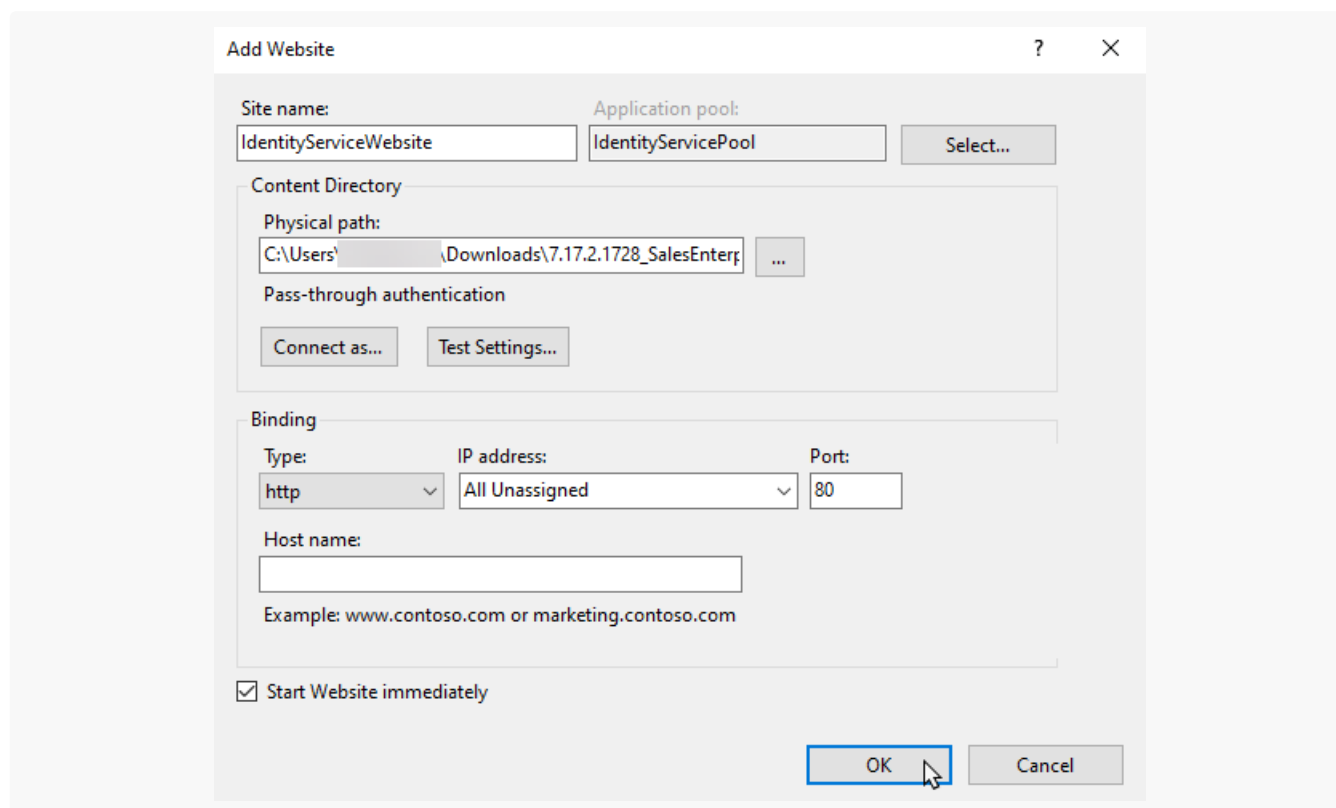
7. Настройте **доступ** к пулу приложения:

- a. Кликните правой кнопкой мыши по созданному пулу. В контекстном меню выберите [*Advanced Settings...*].
- b. В открывшемся окне в поле [*Identity*] укажите пользователя для подключения к Identity Service с правом доступа к папке приложения.

8. Создайте в IIS новый **сайт** для Identity Service.

- a. В окне управления IIS щелкните правой клавишей мыши по [*Sites*] и выберите [*Add Website*] в контекстном меню.
- b. Укажите название сайта, пул и путь к корневому каталогу с файлами Identity Service.

Рис. 3 — Пример настройки сайта в IIS



9. Настройте сайт на работу с **СУБД** вашего приложения Creatio. Для этого в конфигурационном файле **appSettings.json**, который находится в корневом каталоге Identity Service:

- a. В параметре “DbProvider” укажите “MsSqlServer” или “Postgres”.
- b. В настройке MsSqlConnection или PostgresConnection укажите строку соединения. Рекомендуется задавать то же значение строки, что и в основном приложении Creatio. При этом важно, чтобы у пользователя, под которым выполняется подключение к БД, были права на создание и редактирование таблиц.

На заметку. Для подключения Identity Service к приложениям, которые работают с СУБД Oracle, необходимо развернуть дополнительный экземпляр БД на PostgreSQL или Microsoft

SQL.

10. Настройте **системного пользователя** Identity Service. Для этого придумайте уникальные значения ClientId, ClientName, ClientSecret и укажите их в блоке “Clients” конфигурационного файла **appSettings.json**, который находится в корневом каталоге Identity Service. Эти значения будут использоваться для настройки взаимодействия между Identity Service и Creatio. Для всех значений можно использовать большие и маленькие буквы, цифры, специальные символы, например, скобки или знаки препинания.

Рекомендуемые параметры:

- ClientId — 16 символов;
- ClientSecret — 32 символа;
- ClientName — произвольное количество символов.

Пример настройки блока “Clients”

```
"[{"ClientId": "{сгенерировать ClientId}", "ClientName": "{сгенерировать имя}", "Secrets"
```

На заметку. Во избежание ошибок при запуске Identity Service в файле appsettings.json в настройке **“X509CertificatePath”** необходимо указать полный путь к файлу openssl.pfx, который находится в корневом каталоге папки с Identity Service.

11. Переведите Identity Service на работу по протоколу **HTTPS**. Настройки аналогичны тем, которые необходимо выполнить для приложения Creatio. Подробнее: [Перевести Creatio с HTTP на HTTPS](#).

12. Включите **логирование** работы Identity Service.

- В файле web.config, который находится в папке приложения Identity Service, для параметра “stdoutLogEnabled” установить значение “true”.
- В том же файле для параметра “stdoutLogFile” укажите путь к папке, где будут храниться логи приложения.
- Настройте уровень логирования в appsettings.json, который находится в корневом каталоге Identity Service:

```
"Logging": {
  "LogLevel": {
    "Default": "Error"
  }
}
```

Настроить интеграцию с Identity Service на стороне Creatio

1. **Включите** функциональность интеграции с приложением по протоколу OAuth 2.0. Для этого выполните для базы данных вашего приложения приведенный ниже скрипт. Он универсален и может использоваться для Microsoft SQL и PostgreSQL.

```
UPDATE "AdminUnitFeatureState"

    SET "FeatureState" = 1

WHERE "FeatureId" = (

    SELECT

        "Id"

    FROM "Feature"

    WHERE "Code" = 'OAuth20Integration')
```


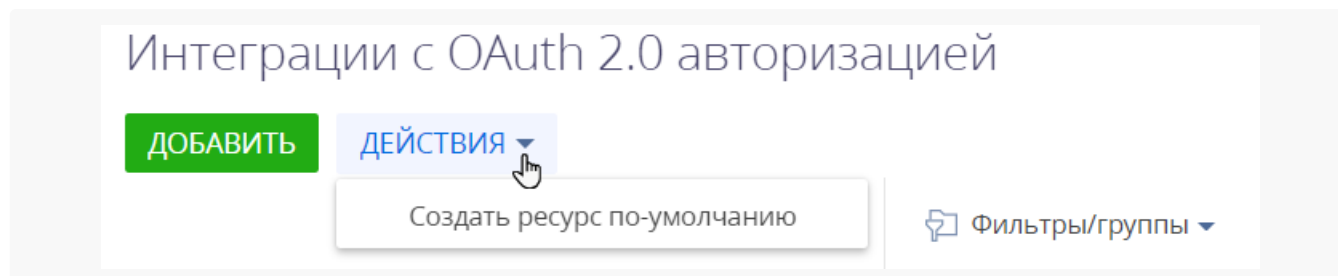
2. Заполните [системные настройки](#), которые входят в группу “OAuth 2.0”:
 - a. **“Адрес сервера авторизации для интеграций по OAuth 2.0”** (код “OAuth20IdentityServerUrl”) — адрес IdentityServer, например, `http://isEndpointExample`.
 - b. **“Идентификатор приложения для интеграций по OAuth 2.0”** (код “OAuth20IdentityServerClientId”) — идентификатор системного пользователя, указанный в параметре “ClientId” файла `appSettings.json` при настройке IdentityServer.
 - c. **“Секрет клиента для интеграций по OAuth 2.0”** (код “OAuth20IdentityServerClientSecret”) — секретный ключ системного пользователя, указанный в параметре “ClientSecret” файла `appSettings.json` при настройке IdentityServer.
3. Создайте ресурс приложения по умолчанию. Это действие выполняется один раз при настройке интеграции Identity Service с Creatio.
 - a. Откройте дизайнер системы по кнопке .
 - b. Перейдите в раздел [*Интеграции с OAuth 2.0 авторизацией*].
 - c. В меню действий выберите [*Создать ресурс по умолчанию*].


Рис. 4 — Создание ресурса по умолчанию



В результате в приложении будет создана запись ресурса по умолчанию с учетными данными вашего Identity Service.

Настроить авторизацию приложений по протоколу OAuth 2.0

После установки Identity Service и подключения его к Creatio необходимо для каждого из приложений, которые будут авторизоваться в Creatio по протоколу OAuth 2.0, создать запись клиента. Для этого:

1. Откройте дизайнер системы по кнопке .
2. Перейдите в раздел [*Интеграции с OAuth 2.0 авторизацией*].
3. Нажмите [*Добавить*].
4. На открывшейся странице заполните параметры клиента (приложения, которое будет подключаться к Creatio):
 - a. [*Название*] — заголовок, который будет отображаться в реестре интеграций и логах.
 - b. [*URL приложения*] — адрес приложения или веб-сервиса, который интегрируется с Creatio.
 - c. [*Описание*] — задача, которую решает данная интеграция.
 - d. [*Активен*] — признак определяет, используется ли интеграция с данным приложением или веб-сервисом.
 - e. [*Пользователь*] — пользователь системы с требуемым для данной интеграции набором прав. Рекомендуется настроить права доступа для этого пользователя таким образом, чтобы он имел доступ только на чтение и редактирование тех полей, в которые интегрируемое приложение или веб-сервис будут вносить изменения. Например, для веб-сервиса, передающего в Creatio курсы валют, это поля [*Курс*] и [*Начало*] страницы записи справочника [*Валюты*].

Учетные данные клиента (идентификатор и секрет) заполняются автоматически.

Рис. 5 — Пример настройки клиента

Основные сведения

Название*

integration example

URL приложения*

http://example.info

Описание

my integration example

Дата создания

06.10.2020 18:21

Активен

☐

Учетные данные клиента

Идентификатор клиента

E78DA10CA8B2A9DEBD88C73BA6126D29

Секрет клиента

298DFE56921802E77E1048BE2CD5D5784A4D0C98A5CCF9890595435F3D6DA16F

Пользователь системы*

Федоров Артем

5. Сохраните запись.

6. Повторите шаги 3–6 для всех приложений, которым необходима авторизация по протоколу OAuth 2.0.

Настроить интеграцию с файловым хранилищем S3

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Возможность настроить интеграцию с файловым хранилищем S3 доступна в Creatio версии 7.18.1 и выше.

Настройка интеграции с файловым хранилищем необходима для on-site приложений.

S3 (Simple Storage Service) — протокол передачи данных, разработанный компанией Amazon.

Файловое хранилище S3 — это облачный REST-сервис хранения объектов. Особенность хранилища S3 — хранение данных в исходном формате без ограничений по масштабированию.

По умолчанию все файлы, загруженные в приложение, хранятся в базе данных. Файлы, загруженные на детали [*Файлы и ссылки*] или прикрепленные к письмам, можно хранить в хранилище S3. Для этого необходимо настроить интеграцию. Использование хранилища S3 позволяет сократить время выполнения резервного копирования базы данных за счет уменьшения ее размера. Creatio позволяет подключить только одно хранилище S3.

На заметку. При настроенной интеграции файлы, загруженные на детали [*Файлы и ссылки*] через мобильное приложение или с помощью бизнес-процесса, также будут загружаться в хранилище S3.

Этапы настройки интеграции с хранилищем S3:

1. Настройка хранилища S3. [Подробнее >>>](#)
2. Настройка хранения файлов. [Подробнее >>>](#)

Шаг 1. Настроить хранилище S3

1. Создайте аккаунт в сервисе хранения данных, который поддерживает протокол S3.
2. В созданном аккаунте получите "ServiceUrl". Это конечная точка, по которой Creatio будет получать доступ к хранилищу S3.
3. В созданном аккаунте сгенерируйте "AccessKey" и "SecretKey". Эти параметры позволяют выполнять авторизованный запрос к хранилищу S3.
4. Создайте корзины (buckets, бакеты) "ObjectBucketName" и "RecycleBucketName" с уникальными именами.
 - "ObjectBucketName" — корзина для хранения файлов. Файлы хранятся неограниченное количество времени.
 - "RecycleBucketName" — корзина для хранения файлов, которые были удалены и хранятся для резервных копий базы данных. В основе работы с корзинами лежит принцип мягкого удаления: после удаления файла из корзины "ObjectBucketName" файл переносится в корзину "RecycleBucketName". Время хранения файлов в корзине определяется настройками корзины определенного сервиса. Например, файл может храниться в корзине 90 дней, затем автоматически удаляться. В Creatio время хранения файлов равно времени хранения резервных копий базы данных.

Создание корзины подробно описано в официальной [документации Amazon](#) (материал на английском языке).

Шаг 2. Настроить хранение файлов

Чтобы новые файлы, загруженные на детали [*Файлы и ссылки*], сохранялись в хранилище S3, а не в базе данных, необходимо выполнить настройку на стороне Creatio:

1. В элемент `<connectionStrings>` файла конфигурации `ConnectionStrings.config` добавьте строку с параметрами подключения к хранилищу S3.

Формат строки:

```
<connectionStrings>
  ...
  <add name="s3Connection" connectionString="ServiceUrl=serviceUrl; AccessKey=accessKey; Se
</connectionStrings>
```

Например:


```
<connectionStrings>
    ...
    <add name="s3Connection" connectionString="ServiceUrl=a8*****25; A
</connectionStrings>
```

2. Чтобы при удалении записей разделов связанные файлы корректно переносились в “RecycleBucketName”, перейдите на страницу дополнительной функциональности и включите функциональность “UseBaseEntityFileDeleteListener”. Подробнее о включении функциональности читайте в статье [Механизм отключения функциональности Feature Toggle](#).

Альтернативным способом включения функциональности является выполнение скрипта в базе данных:

```
UPDATE "AdminUnitFeatureState"
SET "FeatureState" = 1
WHERE "FeatureId" = (
    SELECT "Id"
    FROM "Feature"
    WHERE "Name" = 'UseBaseEntityFileDeleteListener')
```

3. Установите хранилище S3 в качестве активного хранилища файлов. Для этого откройте системную настройку **“Активное хранилище содержимого файлов”** (код ActiveFileContentStorage). В поле [*Значение по умолчанию*] выберите “Хранилище S3”.

Альтернативным способом установки хранилища S3 в качестве активного хранилища файлов является выполнение скрипта в базе данных:

```
UPDATE "SysSettingsValue"
SET GuidValue = (
    SELECT "Id"
    FROM "SysFileContentStorage"
    WHERE "TypeName" = 'Terrasoft.File.S3.Content.S3FileContentStorage, Terrasoft.File.S3'
)
WHERE "SysSettingsId" = (
    SELECT "Id"
    FROM "SysSettings"
    WHERE "Code" = 'ActiveFileContentStorage')
```

Здесь в поле **Name** следует указать тип активного хранилища файлов:

- "Terrasoft.File.S3.Content.S3FileContentStorage, Terrasoft.File.S3" — для хранилища S3;
- "Terrasoft.File.Content.EntityFileContentStorage, Terrasoft.File" — для базы данных.

Выбор активного хранилища касается только новых файлов. Хранилище ранее загруженных файлов останется прежним. При работе с этими файлами приложение будет обращаться к указанному ранее хранилищу.

