

Горизонтальное масштабирование

Настроить горизонтальное масштабирование

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 горизонтальное масштабирование	4
Общий порядок развертывания	4
Установить балансировщик HAProxy	7
Настроить балансировщик HAProxy	8

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 добавить 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, HAProxy) 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/bernart.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:.*)\ \1;\ 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

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"]
```