

Инструкция по установке ПО  
Vimeister «Управление процессами  
проектирования»

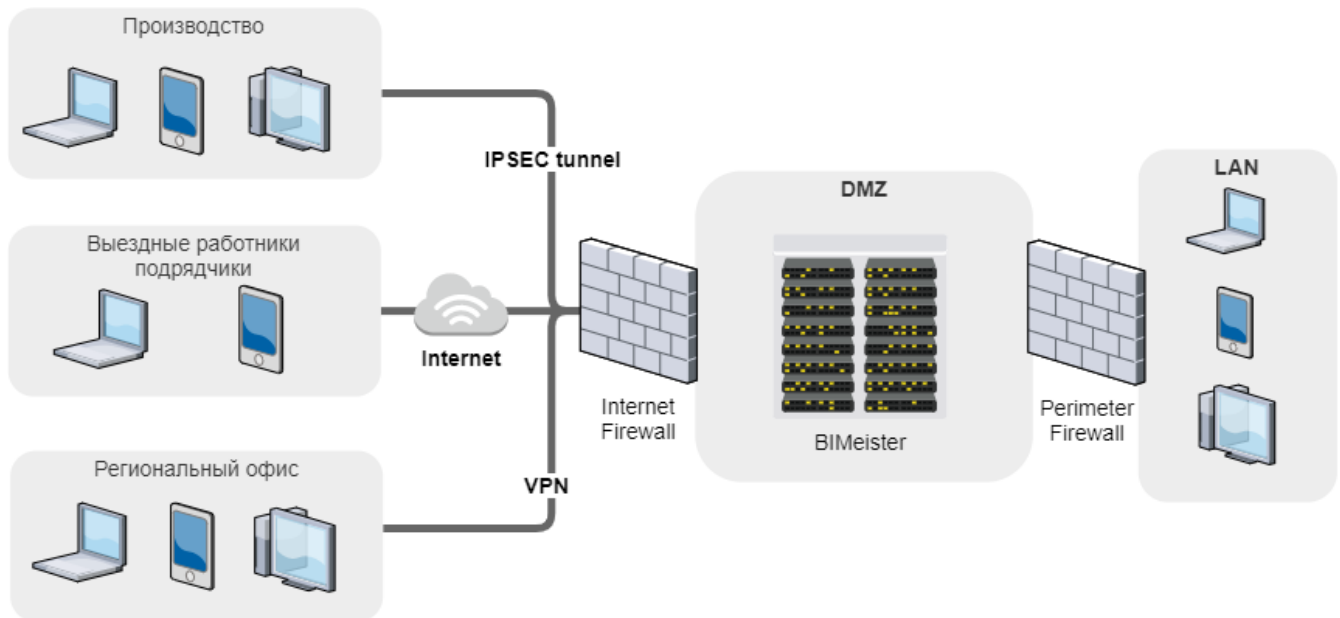
# Оглавление

1. Архитектура .....	1
2. Сервисы .....	3
3. Системные требования .....	6
4. Установка .....	7
4.1. Обновление Vimeister «Управление процессами проектирования» .....	8
4.2. Сервисные команды .....	10
5. Лицензирование Vimeister «Управление процессами проектирования» .....	12
6. Метрики .....	14
7. Поддержка .....	17

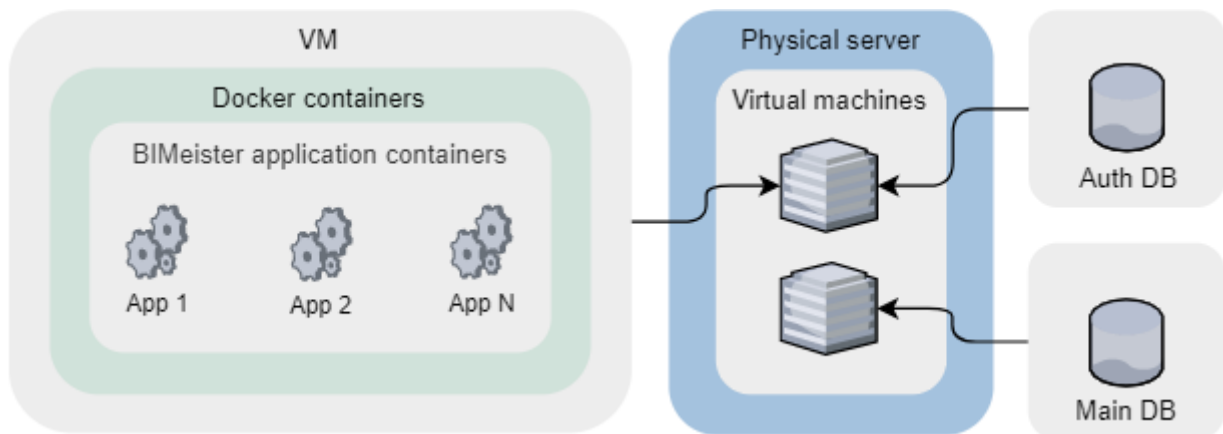
# 1. Архитектура

Bimeister «Управление процессами проектирования» — полностью изолированная система, ей не требуется доступ в интернет, а значит можно закрыть все исходящие порты, кроме одного: для работы по протоколу HTTPS (HTTP не рекомендуется).

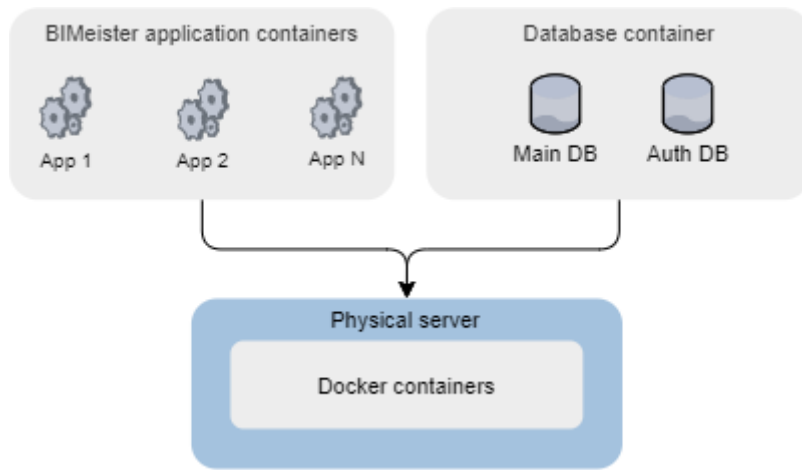
Система спроектирована таким образом, что не накладывает ограничений на топологию сети предприятия, используемое сетевое оборудование и варианты организации DMZ.



*Потоки данных*



*Вариант разворачивания на виртуальной машине*



*Вариант разворачивания без виртуализации*

## 2. Сервисы

Глава описывает сервисы, используемые в Bimeister «Управление процессами проектирования».

Таблица сервисов Bimeister «Управление процессами проектирования»

Сервис	Лицензия	Описание
<b>Ядро</b>		
webapi	Bimeister «Управление процессами проектирования»	Основной сервис-плюз для доступа к другим сервисам. Обрабатывает запросы, формирует и возвращает ответы. Служит единой точкой входа в backend-часть Bimeister «Управление процессами проектирования».
taskworker	Bimeister «Управление процессами проектирования»	Сервис для выполнения отложенных ресурсоемких задач: обработки или удаления моделей, архивов и облаков точек, одиночного и массового создания, удаления или редактирования объектов, импорта объектов, массового создания связей.
ifc-geometry-converter	Bimeister «Управление процессами проектирования»	Сервис обработки IFC-файлов.
pointcloudapi	Bimeister «Управление процессами проектирования»	Сервис обработки облаков точек.
e57service	Bimeister «Управление процессами проектирования»	Сервис обработки облаков точек с расширением .E57. После обработки облака точек конвертируются в формат PTS, которые обрабатываются сервисом pointcloudapi.
spatialwebapi	Bimeister «Управление процессами проектирования»	Сервис обработки 3D-данных.
collisions	Bimeister «Управление процессами проектирования»	Сервис расчета коллизий.
pdfservice	Bimeister «Управление процессами проектирования»	Сервис для работы с PDF-документами: сжимает изображения, встроенные в PDF-файлы. Необходим для корректного отображения файлов.

bimeister_fro ntend	Свободное программное обеспечение. Распространяется по лицензии 2-clause BSD License (Nginx, Inc).	Сервис приема запросов клиентской части. Отдает статику и проксирует REST-запросы к webapi. Используется Nginx — HTTP-сервером, обратным прокси-сервером. Подробнее читайте <a href="#">на сайте Nginx</a> .
RabbitMQ	Свободное программное обеспечение. Распространяется по лицензии Mozilla Public License.	Сервис для организации очередей на основе стандарта AMQP. Очереди используют сервисы для обмена задачами между собой. При этом связь сервисов в очереди является неблокирующей и независимой от сетевых сбоев. Подробнее читайте <a href="#">на сайте RabbitMQ</a> .
Redis	Свободное программное обеспечение. Распространяется по лицензии BSD License.	Резидентная система управления базами данных класса NoSQL с открытым исходным кодом, работающая со структурами данных типа «ключ-значение». Используется как для баз данных, так и для реализации кэшей, брокеров сообщений. Подробнее читайте <a href="#">на сайте Redis</a> .
Neo4j	Свободное программное обеспечение. Распространяется по лицензии GPL v3.	Neo4j — графовая система управления базами данных с открытым исходным кодом, используется для хранения данных объектной модели. Подробнее читайте <a href="#">на сайте Neo4j</a> .
<b>Хранение данных</b>		
db	Свободное программное обеспечение. Распространяется по лицензии PostgreSQL License.	Основная БД. Используется PostgreSQL. PostgreSQL — объектно-реляционная система управления базами данных, основанная на языке SQL. Подробнее читайте <a href="#">на сайте PostgreSQL</a> .
spatialDB	Свободное программное обеспечение. Распространяется по лицензии GNU GPL.	БД хранения 3D-геометрии. Программное обеспечение, добавляющее поддержку географических и 3D-объектов в реляционную базу данных PostgreSQL. Используется PostGIS, подробнее читайте <a href="#">на сайте PostGIS</a> .
authdb	Свободное программное обеспечение. Распространяется по лицензии Open Source license.	БД хранения данных о группах, правах, ролях, пользователях и т.д. Необходима для работы сервиса auth. Используется PostgreSQL. Подробнее читайте <a href="#">на сайте PostgreSQL</a> .

influxDB	Свободное программное обеспечение. Распространяется по лицензии MIT License.	Система управления базами данных представляет собой программное обеспечение для хранения временных рядов. Основным назначением является хранение больших объемов данных с метками времени. Используется при расчете статистики для пользователя. Также служит для хранения данных о задачах. Подробнее читайте <a href="#">на сайте influxDB</a> .
minIO	Свободное программное обеспечение. Распространяется по лицензии GNU Affero GPL.	Высокопроизводительное объектное хранилище, совместимое с S3-протоколом. Служит для хранения загруженных пользователем файлов: документов, 3D-моделей, BPMN-диаграмм и т.д. Подробнее читайте <a href="#">на сайте minIO</a> .
<b>Служебные</b>		
auth	Vimeister «Управление процессами проектирования»	Сервис авторизации. Используется для аутентификации и авторизации пользователей.
ldapwebapi	Vimeister «Управление процессами проектирования»	Сервис интеграции с LDAP.
license-service	Vimeister «Управление процессами проектирования»	Сервис лицензий. Используется для лицензирования Vimeister «Управление процессами проектирования».
notification	Vimeister «Управление процессами проектирования»	Сервис обработки уведомлений.
mailservice	Vimeister «Управление процессами проектирования»	Сервис отправки почты.

## 3. Системные требования

### Минимальная конфигурация

#### *Клиентская часть*

- Browser: Chrome 81+.
- CPU: тактовая частота 1 ГГц, 4 ядра, техпроцесс 14 нм.
- GPU: частота ядра 1000 МГц, поддержка OpenGL ES 3.0 или OpenGL 4.1, техпроцесс 14 нм.
- RAM: 8 ГБ.
- Место на диске: 4 ГБ.
- Экран с соотношением сторон 16:9 и разрешением не менее 1280 x 720.
- Net: 3G (10 Мбит/с).

#### *Серверная часть*

- CPU: 8 ядер серверного класса с поддержкой виртуализации, с тактовой частотой 2.2 ГГц и выше.
- RAM: 24 ГБ.
- Место на диске: 100 ГБ.
- Net: 100 Мбит/с.
- Допускается установка в виде виртуальной машины.

### Рекомендуемая конфигурация

#### *Клиентская часть*

- Browser: Chrome 81+.
- CPU: тактовая частота 2.2 ГГц, 4 ядра, техпроцесс 14 нм.
- GPU: частота ядра 1000 МГц, поддержка OpenGL ES 3.0 или OpenGL 4.1, техпроцесс 14 нм.
- RAM: 16 ГБ.
- Место на диске: 4 ГБ, SSD.
- Экран с соотношением сторон 16:9 и разрешением не менее 1280 x 720.
- Net: 4G (30 Мбит/с).

#### *Серверная часть*

- CPU: 18 ядер серверного класса с поддержкой виртуализации, с тактовой частотой 2.2 ГГц и выше.
- RAM: 128 ГБ.
- Net: 1 Гбит/с.
- Место на диске: 1 ТБ, SSD, для хорошей производительности желательно использование RAID-1 и выше.
- Допускается установка в виде виртуальной машины.



## 4. Установка

Глава описывает установку Vimeister «Управление процессами проектирования» на системы семейства Linux.

### Требования

Перед установкой Vimeister «Управление процессами проектирования», на компьютер должны быть установлены следующие программы:

- Docker Engine 17.09.0+.  
<https://docs.docker.com/engine/install/>
- Docker Compose 1.17.0+.  
<https://docs.docker.com/compose/install/>

### Состав поставки

- .env — конфигурационный файл.
- docker-compose.yml — файл docker-compose.
- backup.sh — скрипт запуска резервного копирования всех файлов Vimeister «Управление процессами проектирования».
- restore.sh — скрипт восстановления из резервной копии всех файлов Vimeister «Управление процессами проектирования».
- images.tar.gz — архив с Docker образами. Присутствует только в архиве для автономной установки.
- VimCli.exe — приложение для администрирования Vimeister «Управление процессами проектирования».

### Подготовка

- В целях безопасности рекомендуем заблокировать все входящие порты, кроме 80-го, он используется для доступа к пользовательскому интерфейсу, и 443-го, если используется https.
- Если требуется обслуживание системы посредством ssh, откройте 22-й порт (может отличаться на конкретной конфигурации).

Пример команд для CentOS7, которые открывают доступ к портам 80 и 22:

```
firewall-cmd --zone=public --add-port=80/tcp --permanent
```

```
firewall-cmd --zone=public --add-port=22/tcp --permanent
```

```
firewall-cmd --reload
```

## Установка

Vimeister «Управление процессами проектирования» устанавливается в папку, указанную в **Docker Root Dir**.

Чтобы установить Vimeister «Управление процессами проектирования»:

1. Распакуйте установочный архив в папку `~/bimeister`. Не удаляйте папку после установки.
2. Перед установкой в ENV-файле поменяйте:
  - Значения по умолчанию и секреты.
  - В параметре `FRONTEND_URL` укажите имя DNS-сервера.
3. Если используется удаленное подключение к Vimeister «Управление процессами проектирования» (не через локальную сеть), для увеличения безопасности, так же в `.env` файле, раскомментируйте параметр `ENABLE_CORS` — он ограничивает внешние запросы к системе.
4. В командной строке поочередно выполните:

```
cd ~/bimeister
```

```
docker load -i images.tar.gz
```

```
docker-compose -f docker-compose.yml --project-name prod up -d
```

### 4.1. Обновление Vimeister «Управление процессами проектирования»

Перед обновлением Vimeister «Управление процессами проектирования» создайте пользователя с административными правами для `influxdb` и очистите таблицу `UserObjects`, на случай, если в новой версии поменялся формат хранения данных. Таблица содержит маловажные данные системы, например, ID последней открытой задачи пользователя.

Сделайте бэкап системы одним из способов: \* Сделайте снимок (snapshot) виртуальной машины. Более простой способ. \* Запустите скрипт `backup.sh` в папке `scripts` архива обновления и убедитесь, что процесс завершился без ошибок.

Переменные, используемые в командах:

- `[host]` — адрес хоста, например `sbs.bimeister.com`.
- `[private token]` — private token.
- `[login]` и `[password]` — ваш логин и пароль в Vimeister «Управление процессами проектирования».
- `[update folder]` — папка с файлами обновления.
- `[container_name_influxdb]` — имя контейнера `influxdb`.

Чтобы создать пользователя с административными правами:

1. Найдите имя контейнера influxdb:

```
docker ps
```

2. Откройте контейнер:

```
docker exec -ti [container_name_influxdb] bash
```

3. Внутри контейнера поочередно выполните команды:

```
influx
```

```
CREATE USER admin WITH PASSWORD [password] WITH ALL PRIVILEGES
```

```
quit
```

```
exit
```

Чтобы очистить таблицу:

1. Получите private token:

```
./BimCli -e [host] config get-private-token -u [login] -p [password]
```

2. Очистите таблицу:

```
./BimCli -t [private token] -e [host] user-objects clean
```

Чтобы обновить Vimeister «Управление процессами проектирования»:

1. Распакуйте архив с обновлением.
2. Если при установке вы редактировали значения ENV-файла, также отредактируйте их в папке с обновлением:
  - Значения по умолчанию и секреты.
  - В параметре FRONTEND\_URL укажите имя DNS-сервера.
3. Если используется удаленное подключение к Vimeister «Управление процессами проектирования» (не через локальную сеть), для увеличения безопасности, так же в .env файле, раскомментируйте параметр ENABLE\_CORS — он ограничивает внешние запросы

к системе.

4. В командной строке поочередно выполните:

```
cd ~/[update folder]
```

```
docker load -i images.tar.gz
```

```
docker-compose -f docker-compose.yml --project-name prod up -d
```



Рекомендуем выполнять команды в `bash` или `PowerShell`. Если вы используете стандартную командную строку Windows, замените в командах `./BimCli` на `BimCli.exe`.

## 4.2. Сервисные команды

Команды необходимо выполнять в командной строке из папки с распакованным установочным архивом Vimeister «Управление процессами проектирования».

### Создание резервной копии

Система автоматически остановится на время создания резервной копии и запустится сразу после завершения.

Не рекомендуем создавать резервные копии из-под `sudo`.

Для создания резервной копии в командной строке поочередно выполните:

```
chmod +x scripts/backup.sh
```

```
./scripts/backup.sh docker-compose.yml prod
```

После завершения процесса Vimeister «Управление процессами проектирования» запустится автоматически, а в папке, откуда была выполнена команда, создастся архив с резервной копией `backup_дд_мм_гггг.tar`, где `дд_мм_гггг` — текущая дата.



Данный способ создания резервной копии не работает для систем с внешней базой данных.

### Восстановление резервной копии

Система автоматически остановится на время восстановления резервной копии и запустится сразу после завершения.

Для восстановления резервной копии в командной строке поочередно выполните:

```
chmod +x scripts/restore.sh
```

```
./scripts/restore.sh docker-compose.yml prod backup_дд_мм_гггг.tar
```

Где backup\_дд\_мм\_гггг.tar — архив с резервной копией.

## Перезагрузка системы

Для перезагрузки Vimeister «Управление процессами проектирования» в командной строке выполните:

```
docker-compose -f docker-compose.yml --project-name prod restart
```



Не используйте команды `docker restart`, `docker start` и `docker stop` — это приведет к сбоям в работе Vimeister «Управление процессами проектирования». Если команды все-таки были выполнены, перезагрузите систему, как показано выше.

## Удаление системы

Для удаления Vimeister «Управление процессами проектирования» и ее данных в командной строке поочередно выполните:

```
docker-compose -f docker-compose.yml --project-name prod down
```

```
docker volume prune
```

## 5. Лицензирование Vimeister «Управление процессами проектирования»

По умолчанию Vimeister «Управление процессами проектирования» устанавливается с неактивной лицензией, ограниченной по времени и функциональности. Чтобы получить доступ к полной версии системы, необходимо активировать лицензию Vimeister «Управление процессами проектирования».

### Активация лицензии через VimCli

Активация лицензии происходит с помощью поочередного выполнения команд из командной строки. Команды выполняются из папки с файлом VimCli.

*Переменные, используемые в командах:*

- [host] — адрес хоста, например <http://sbs.bimeister.com>.
- [private token] — private token.
- [base64license] — ключ лицензии, который высылает менеджер Vimeister «Управление процессами проектирования» в ответ на ID сервера.
- [login] и [password] — ваш логин и пароль в Vimeister «Управление процессами проектирования».
- [IdLicense] — ID активируемой лицензии.



Рекомендуем выполнять команды в bash или PowerShell. Если вы используете стандартную командную строку Windows, замените в командах `./VimCli` на `VimCli.exe`.

*Чтобы активировать лицензию:*

1. Получите private token:

```
./VimCli -e [host] config get-private-token -u [login] -p [password]
```

2. Получите ID сервера:

```
./VimCli -t [private token] -e [host] licenses get-serverId
```

3. Отправьте полученный ID сервера вашему менеджеру Vimeister «Управление процессами проектирования», в ответном письме он пришлет ключ лицензии.
4. Добавьте ключ лицензии в систему:

```
./VimCli -e [host] -t [private token] licenses upload-license -l [base64license]
```

Если вы добавляете лицензию впервые, после этих шагов она активируется автоматически.

Если вы добавляете лицензию повторно, выполните дополнительные шаги.

*Дополнительные шаги:*

1. Получите список ID доступных лицензий:

```
./Bimcli -t [token] -e [host] licenses list
```

2. Активируйте необходимую лицензию:

```
./Bimcli -t [token] -e [host] licenses activate -lid [IdLicense]
```

## 6. Метрики

Bimeister «Управление процессами проектирования» предоставляет метрики некоторых микросервисов. Мы условно разделили метрики на [.NET-метрики](#), связанные с производительностью микросервисов, и [HTTP-метрики](#), связанные с запросами к микросервисам от пользователей и других микросервисов.

Сервисы Bimeister «Управление процессами проектирования», предоставляющие метрики:

- webapi.
- spatialwebapi.
- mailservice.
- auth.
- ldapwebapi.
- license-service.
- objectapi.
- **pointcloudapi.**
- **notification.**
- **taskworker.**
- **collisions.**

**Выделенные** сервисы предоставляют только .NET-метрики.

## Подготовка

Добавьте сервисы в вашу программу мониторинга:

- Если порты открыты.  
Добавьте сервисы в формате [hostname]:[port], например, bimeister.com:8080.  
Чтобы узнать порты сервисов, выполните команду в Docker:

```
docker ps
```

- Если порты закрыты.  
Добавьте сервисы в формате [hostname]/metrics/[service\_name], например, bimeister.com/metrics/webapi

## .NET-метрики

*process\_virtual\_memory\_bytes*

Gauge метрика. Показывает объем виртуальной памяти главного процесса. Вычисляется на основе свойства Process.VirtualMemorySize64, подробнее читайте [на сайте Microsoft](#).

*process\_start\_time\_seconds*



Gauge метрика. Показывает время запуска главного процесса. Вычисляется на основе свойства `Process.StartTime`, подробнее читайте [на сайте Microsoft](#).

#### *process\_num\_threads*

Gauge метрика. Показывает количество потоков главного процесса. Вычисляется на основе свойства `Process.Threads`, подробнее читайте [на сайте Microsoft](#).

#### *process\_private\_memory\_bytes*

Gauge метрика. Показывает в байтах объем памяти, выделенной для связанного процесса. Эта память не доступна другим процессам. Вычисляется на основе свойства `Process.PrivateMemorySize64`, подробнее читайте [на сайте Microsoft](#).

#### *dotnet\_total\_memory\_bytes*

Gauge метрика. Показывает в байтах предполагаемый объем в управляемой памяти главного процесса в микросервисе. Вычисляется на основе свойства `GC.GetTotalMemory(false)`, подробнее читайте [на сайте Microsoft](#).

#### *process\_cpu\_seconds\_total*

Counter метрика. Показывает суммарное время работы процессора, затраченное на основной процесс микросервиса. Вычисляется на основе свойства `Process.TotalProcessorTime`, подробнее читайте [на сайте Microsoft](#).

#### *process\_working\_set\_bytes*

Gauge метрика. Показывает в байтах объем физической памяти, выделенной для связанного процесса. Вычисляется на основе свойства `Process.WorkingSet64`, подробнее читайте [на сайте Microsoft](#).

#### *dotnet\_collection\_count\_total*

Counter метрика. Показывает суммарное количество операций сборки мусора, выполненных для заданного поколения объектов с начала процесса. Вычисляется на основе метода `GC.CollectionCount()`, подробнее читайте [на сайте Microsoft](#).

Метки: generation (номер поколения), доступные значения 0, 1, 2.

#### *process\_open\_handles*

Gauge метрика. Показывает число дескрипторов операционной системы, открытых процессом. Вычисляется на основе свойства `Process.HandleCount`, подробнее читайте [на сайте Microsoft](#).

## HTTP-метрики

#### *http\_requests\_received\_total*

Counter метрика. Показывает суммарное количество запросов, обработанных сервисом, со старта процесса с детализацией по коду, методу, контроллеру и действию.

Метки:

- code — код состояния HTTP-запроса к сервису.
- method — HTTP-метод запроса к сервису.

- `controller` — имя контроллера.
- `action` — имя метода контроллера.

#### *http\_request\_duration\_seconds*

Histogram метрика. Показывает длительность запросов к сервису с детализацией по коду, методу, контроллеру и действию.

#### *Метки:*

- `code` — код состояния HTTP-запроса к сервису.
- `method` — HTTP-метод запроса к сервису.
- `controller` — имя контроллера.
- `action` — имя метода контроллера.

#### *http\_requests\_in\_progress*

Gauge метрика. Показывает количество запросов, обрабатываемое сервисом, с детализацией по методу, контроллеру и действию. Показывается количество на момент запроса метрики.

#### *Метки:*

- `method` — HTTP-метод запроса к сервису.
- `controller` — имя контроллера.
- `action` — имя метода контроллера.

## 7. Поддержка

Для обращения в службу поддержки Vimeister «Управление процессами проектирования» напишите на почту {supportmail} или создайте заявку на [портале поддержки](#).

Сроки и объемы поддержки уточняйте у вашего менеджера Vimeister «Управление процессами проектирования».