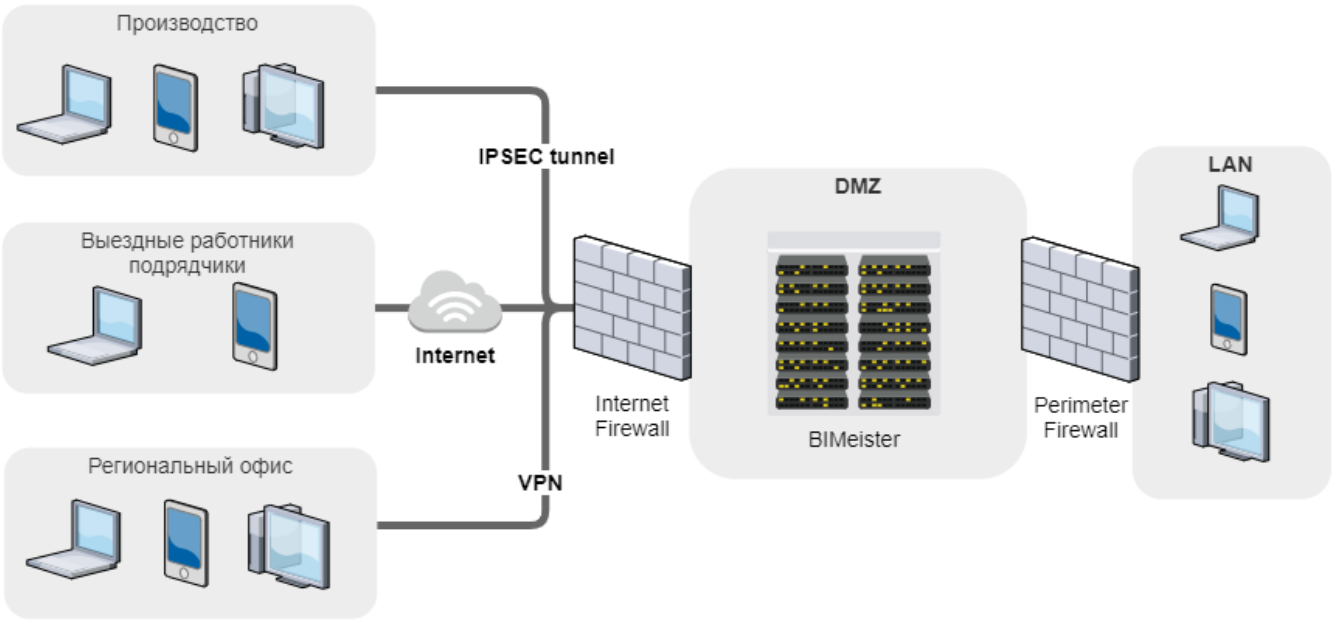
# Руководство по установке ПО BIMeister

# Оглавление

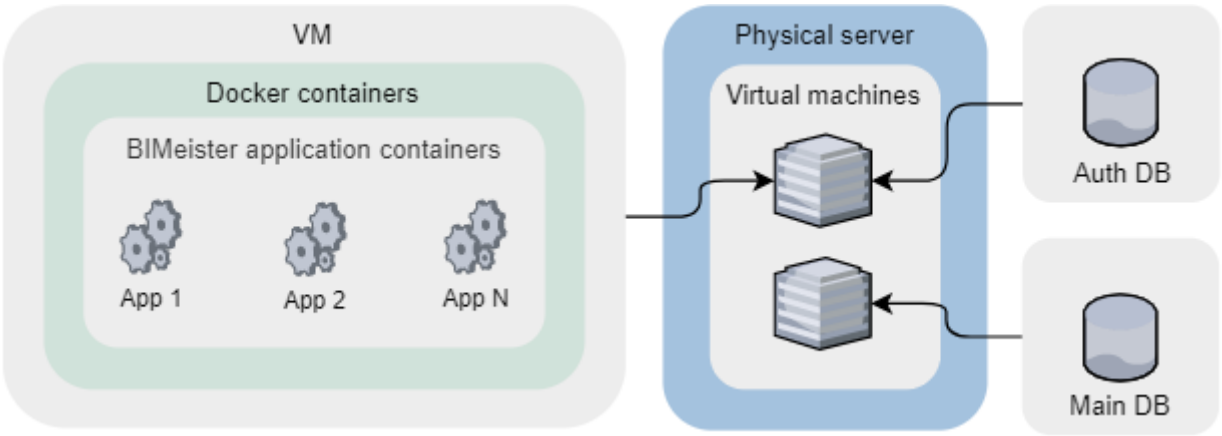
# 1. Архитектура

BIMeister — 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).

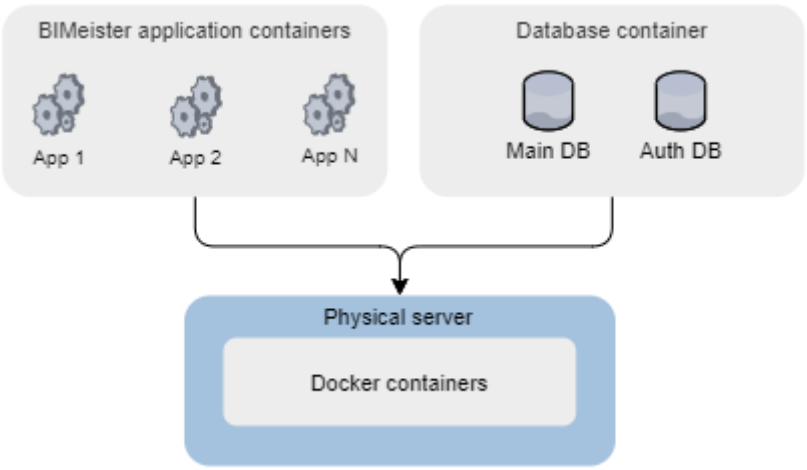
Система спроектирована таким образом, что не накладывает ограничений на топологию сети предприятия, используемое сетевое оборудование и варианты организации DMZ.




*Потоки данных*




*Вариант разворачивания на виртуальной машине*




*Вариант разворачивания без виртуализации*

# 2. Сервисы

Глава описывает сервисы, используемые в BIMeister.

*Таблица сервисов BIMeister*

| Сервис | Лицензия | Описание |
|---|---|---|
| **Ядро** | | |
| webapi | BIMeister | Основной сервис-шлюз для доступа к другим сервисам. Обрабатывает запросы, формирует и возвращает ответы. Служит единой точкой входа в backend-часть BIMeister. |
| tasksworker | BIMeister | 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 | Сервис обработки IFC-файлов. |
| pointcloudapi | BIMeister | Сервис обработки облаков точек. |
| e57service | BIMeister | Сервис обработки облаков точек с расширением .E57. После обработки облака точек конвертируются в формат PTS, которые обрабатываются сервисом pointcloudapi. |
| spatialwebapi | BIMeister | Сервис обработки 3D-данных. |
| collisions | BIMeister | Сервис расчета коллизий. |
| pdfservice | BIMeister | Сервис для работы с PDF-документами: сжимает изображения, встроенные в PDF-файлы. Необходим для корректного отображения файлов. |
| bimeister_frontend | 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 на сайте Nginx. |
| 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 на сайте RabbitMQ. |

| | | |
|---|---|---|
| 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 на сайте Redis. |
| 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 на сайте Neo4j. |
| **Хранение данных** | | |
| db | Свободное программное обеспечение. Распространяется по лицензии PostgreSQL License. | Основная БД. Используется PostrgeSQL. PostrgeSQL — объектно-реляционная система управления базами данных, основанная на языке SQL. Подробнее читайте на сайте PostgreSQL. |
| 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 на сайте PostGIS. |
| 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 PostrgeSQL. Подробнее читайте на сайте PostgreSQL. |
| 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 на сайте influxDB. |
| 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 на сайте minIO. |
| **Служебные** | | |

| | | |
|---|---|---|
| auth | BIMeister | Сервис авторизации. Используется для аутентификации и авторизации пользователей. |
| ldapwebapi | BIMeister | Сервис интеграции с LDAP. |
| license-service | BIMeister | Сервис лицензий. Используется для лицензирования BIMeister. |
| notification | BIMeister | Сервис обработки уведомлений. |
| mailservice | BIMeister | Сервис отправки почты. |

# 3. Системные требования

## Минимальная конфигурация

*Клиентская часть*

- Browser: Chrome 81+.
- CPU: I3, 4 ядра, с тактовой частотой 1 ГГц и выше.
- GPU: Intel HD Graphics 520/550/620 или выше классом.
- RAM: 4 GB.
- Место на диске: минимум 4 GB свободного места на диске.
- Экран с соотношением сторон 16:9 и разрешением не менее 1280 x 720.
- Net: 3G (10 Mbit/s).

*Серверная часть*

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

## Рекомендуемая конфигурация

*Клиентская часть*

- Browser: Chrome 81+.
- CPU: I5, 4 ядра.
- GPU: Intel HD Graphics 630 или выше классом.
- RAM: 8 GB.
- SSD: минимум 4 GB свободного места на диске.
- Экран с соотношением сторон 16:9 и разрешением не менее 1280 x 720.
- Net: 4G (30 Mbit/s).

*Серверная часть*

- CPU: 18 ядер серверного класса с поддержкой виртуализации, с тактовой частотой 2.2 ГГц и выше.
- RAM: 128 GB.
- Net: 1 Gbit/s.
- SSD: минимум 1TB, для хорошей производительности желательно использование RAID-1 и выше.
- Допускается установка в виде виртуальной машины.

# 4. Установка

Глава описывает установку BIMeister на системы семейства Linux.

## Требования

*Перед установкой BIMeister, 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 BIMeister.
- restore.sh — скрипт восстановления из резервной копии всех файлов BIMeister.
- images.tar.gz — архив с Docker образами. Присутствует только в архиве для автономной установки.
- BimCli.exe — приложение для администрирования BIMeister.

## 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

BIMeister устанавливается в папку, указанную в **Docker Root Dir**.

*Чтобы установить BIMeister:*

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 BIMeister (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 BIMeister

Перед обновлением BIMeister очистите таблицу UserObjects, на случай, если в новой версии поменялся формат хранения данных. Таблица содержит маловажные данные системы, например, ID последней открытой задачи пользователя.

*Переменные, используемые в командах:*

- [host] — адрес хоста, например http://sbs.bimeister.com.
- [private token] — private token.
- [login] и [password] — ваш логин и пароль в BIMeister.
- [update folder] — папка с файлами обновления.

*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 BIMeister:*

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 BIMeister (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 BIMeister.

### 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 BIMeister 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 BIMeister 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 BIMeister. Если команды все-таки были выполнены, перезагрузите систему, как показано выше.

### Удаление системы

Для удаления BIMeister и ее данных в командной строке поочередно выполните:

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

```
docker volume prune
```

# 5. Лицензирование BIMeister

По умолчанию BIMeister устанавливается с неактивной лицензией, ограниченной по времени и функциональности. Чтобы получить доступ к полной версии системы, необходимо активировать лицензию BIMeister.

### Активация лицензии через BimCli

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

*Переменные, используемые в командах:*

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

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

*Чтобы активировать лицензию:*

1. Получите private token:

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

2. Получите ID сервера:

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

3. Отправьте полученный ID сервера вашему менеджеру BIMeister, в ответном письме он пришлет ключ лицензии.
4. Добавьте ключ лицензии в систему:

```
./BimCli -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 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, предоставляющие метрики:*

- webapi.
- spatialwebapi.
- mailservice.
- auth.
- ldapwebapi.
- license-service.
- objectapi.
- pointcloudapi.
- notification.
- tasksworker.
- 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 недоступна 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 BIMeister напишите на почту support@bimeister.com или создайте заявку на портале поддержки.
Сроки и объемы поддержки уточняйте у вашего менеджера BIMeister.