

**Информация,  
необходимая для  
установки серверной части  
системы адаптивного внутрицехового  
планирования производства в  
реальном времени  
"МТ. Производство"  
(в составе базовых модулей)**

## Оглавление

Требования к аппаратному и программному обеспечению сервера и рабочих станций.....	3
Установка и настройка серверной части.....	3
Диагностика и управление.....	4
Обновление системы.....	4
Остановка и перезапуск системы.....	5
Список контейнеров.....	5

## Требования к аппаратному и программному обеспечению сервера и рабочих станций.

Вычислительное ядро системы реализовано в виде веб-приложения на платформе .NET Core 2. Для обмена данными система может интегрироваться с ERP-системами.

Аппаратные требования	Сервер с процессором Xeon или аналогичный частотой не ниже 3Гц, не менее 4 ядер, RAM 8 Гб, 10 Гб свободного пространства на диске
Дополнительное программное обеспечение для функционирования системы	.NET Core 2, MongoDB 4.4, Mongo-express, Nginx, Docker, Docker-compose, Portainer

## Установка и настройка серверной части.

Развертывание и управление обновлениями системы выполняется в автоматическом режиме через среду контейнеризации Docker. При развертывании системы будут автоматически загружены и установлены все требуемые компоненты и дополнительное программное обеспечение, включая СУБД и фреймворки.

Для корректной работы с компонентами через Docker требуется подключение к сети интернет.

Рекомендованной ОС является Ubuntu 20.04 LTS.

Ставим систему Ubuntu Server без графической оболочки и дополнительных пакетов.

Обновляем и устанавливаем Docker и Docker-compose.

- `sudo apt update`
- `sudo apt upgrade -y`
- `sudo apt install apt-transport-https ca-certificates curl software-properties-common`
- `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -`
- `sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"`
- `sudo apt update`
- `sudo apt install docker-ce`
- `sudo usermod -aG docker ${USER}`
- `su - ${USER}`
- `sudo curl -L "https://github.com/docker/compose/releases/download/v2.5.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose`
- `sudo chmod +x /usr/local/bin/docker-compose`

Создаем каталог для файлов конфигурации и запуска <...>, в папке пользователя и копируем туда `docker-compose.yml`

Так же в каталоге, создаем каталог `nginx-conf` и копируем в него файлы `default.conf`, `cataloghome.html` и `catalog-16.png`

В файле конфигурации прокси-сервера системы, `default.conf`, находятся важные системные переменные. Необходимо заменить `listen` (адрес порта, по которому будет осуществляться доступ к системе) и `server_name` (доменное имя, по которому будет доступна система) на свои.

Так же необходимо внести изменения в файл `cataloghome.html`, так же прописав корректное имя домена, на котором происходит установка продукта.

В файле `docker-compose.yml` описаны все компоненты системы и параметры настроек для запуска конкретного экземпляра системы. Необходимо проверить переменные: `Services__Scheduling_BaseAddress`, `DataSourceAddress`, `NGINX_HOST` и выставить в них корректное имя вашего домена на котором будет работать ваша инсталляция продукта.

Так же, в файле `docker-compose.yml`, необходимо исправить раздел контейнера `Nginx` и прописать правильный путь до папки с конфигурацией `nginx-conf`. Он будет `/home/{имя пользователя}/catalog/nginx-conf`

Регистрируемся на сервере загрузки `mes` и скачиваем все модули, необходимые для работы системы, для этого из каталога выполняем следующие команды:

- `docker login docker.mes.com` (логин и пароль прописан в договоре)
- `docker-compose pull`
- `docker-compose up -d mongo`

После чего настраиваем сервис `Mongo`. Это делается однократно, только при начальной установке системы. Находясь в папке `<...>` выполняем следующие команды:

- `docker exec -it mongo mongo --port 27018`
- `rs.initiate({_id: "repllocal", members: [{_id: 0, host: "mongo:27018"}] })`
- Нажать `ENTER` два раза до появления приглашения `repllocal:PRIMARY>`
- Выполнить команду `exit`
- `docker-compose up -d`

### Диагностика и управление.

Для диагностики системы, управления контейнерами и доступа к журналам системы можно установить `Portainer`.

- `docker volume create portainer_data`
- `docker run -d -p 8000:8000 -p 9443:9443 --name portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:2.9.3`

Он будет доступен по адресу: `http://ip-address-mes:9443`

### Обновление системы.

Обновление компонентов системы производится посредством загрузки последней версии `docker-образов`. Для этого, находясь в папке `<...>` необходимо выполнить команды:

- docker login docker.mes.com (логин и пароль прописан в договоре)
- docker-compose pull
- docker-compose up -d

### Остановка и перезапуск системы.

Остановка системы выполняется командой из каталога системы:

- docker-compose down

Перезапуск системы выполняется командой из каталога системы:

- docker-compose restart

В случае внеплановой или аварийной остановки сервера после его нормальной загрузки рекомендуется перезапустить систему, выполнив последовательно команды из каталога:

- docker-compose down
- docker-compose up -d

### Список контейнеров.

datasource	docker.mes.com/catalog/datasource:2.0.3	5050/tcp
intapi	docker.mes.com/catalog/integration:2.6.0	5050/tcp
scheduling	docker.mes.com/catalog/scheduling:2.0.2	5050/tcp
dpu-worker	docker.mes.com/catalog/dpu/worker:1.1.1	5050/tcp
dpu-foreman	docker.mes.com/catalog/dpu/foreman:1.2.0	5050/tcp
dpu-server	docker.mes.com/catalog/dpu/server:1.2.0	5050/tcp
mexpress	mongo-express	8081/tcp
mongo	mongo:4.4	27018/tcp
nginx	nginx	80/tcp
* portainer	portainer-ce:2.9.3	9443/tcp