

Инструкция по установке и настройке ПО

**Автоматизированная система нормирования, оперативного
планирования и диспетчеризации
производства «ЦИФРОНИТ.СУПР»**

Содержание

Требования к аппаратному и программному обеспечению сервера и рабочих станций.....	3
Установка и настройка серверной части.	3
Обновление системы.....	5
Остановка и перезапуск системы.	6
Список контейнеров.....	6

Требования к аппаратному и программному обеспечению сервера и рабочих станций.

Вычислительное ядро системы реализовано в виде клиент-серверного веб-приложения, разработанного с использованием платформы .NET 8 и библиотеки React.

Аппаратные требования	Сервер с процессором Xeon или аналогичный частотой не ниже 3Гц, не менее 2 ядер, RAM 16 Гб, 20 Гб свободного пространства на диске
Дополнительное программное обеспечение для функционирования системы	Docker, Docker-compose, Nano

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 22.04 LTS.

Ставим систему Ubuntu Server без графической оболочки и дополнительных пакетов.

1. Обновляем и устанавливаем Docker и Docker-compose.

- `sudo apt-get update`
- `sudo apt-get install ca-certificates curl`
- `sudo install -m 0755 -d /etc/apt/keyrings`
- `sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc`
- `sudo chmod a+r /etc/apt/keyrings/docker.asc`
- `echo \`
`"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]`
`https://download.docker.com/linux/ubuntu \`
`$(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | \`
• `sudo tee /etc/apt/sources.list.d/docker.list > /dev/null`

- `sudo apt-get update`
- `sudo apt install docker-ce docker-compose`

Создаем каталог для файлов конфигурации и запуска `/opt/mess` и копируем туда `docker-compose.yml` и `.env`

Команды для создания и перехода в каталог:

- `sudo mkdir /opt/mess`
- `cd /opt/mess/`

Команды для копирования из Windows по ssh:

- `scp -r <путь к папке Windows> <{$User}>@<ip адрес сервера в локальной сети>:/home/user`
- `sudo mv home/{$User}/<название папки>/docker-compose.yml /opt/mess/docker-compose.yml`
- `sudo mv home/{$User}/<название папки>/.env /opt/mess/.env`

2. В файле `.env`, хранятся необходимые данные для первоначальной настройки системы.

Настройка	Описание	Значение по умолчанию
FRONT_HOST_PORT	Порт работы графического интерфейса приложения	80
FRONT_IMAGE_TAG	Тэг сборки графического интерфейса приложения	imagename:release-x.x.x
NGINX_CONF_PATH	Путь монтирования папки с конфигом nginx	/var/lib/nginx
BACK_HOST_PORT	Порт работы серверной части приложения	5000
BACK_IMAGE_TAG	Тэг сборки серверной части приложения	imagename:release-x.x.x
POSTGRES_PASSWORD	Пароль от пользователя реалиционной базы данных	
POSTGRES_HOST_PORT	Порт работы реалиционной базы данных	5432
POSTGRES_VOLUME_PATH	Путь монтирования папки с файлами реалиционной базы данных	/var/lib/postgresql_data
REDIS_HOST_PORT	Порт работы базы данных для кеша	6379
REDIS_COMANDER_HOST_PORT	Порт работы системы управления базы данных для кеша	8081
MINIO_HOST_PORT	Порт работы файлового хранилища	9000
MINIO_ROOT_PASSWORD	Пароль от пользователя файлового хранилища	
MINIO_VOLUME_PATH	Путь монтирования папки с файлами файлового хранилища	/var/lib/minio_data
MONGO_HOST_PORT	Порт работы нереалиционной базы данных	27017
MONGO_ROOT_PASSWORD	Пароль от пользователя нереалиционной базы данных	

MONGO_VOLUME_PATH	Путь монтирования папки с файлами нереалиционной базы данных	/var/lib/mongodb_data
RABBITMQ_HOST_PORT	Порт работы брокера сообщений	5672
RABBITMQ_WEB_PORT	Порт работы систем управления брокером сообщений	15672
RABBITMQ_ROOT_PASSWORD	Пароль от пользователя брокера сообщений	
NEXUS_PATH	URL хранилища образов	
NEXUS_REPOSITORY	Имя репозитория в хранилище образов	

Через редактор нужно заполнить настройки {SERVICE}_ROOT_PASSWORD и если необходимо изменить остальные.

- `sudo nano .env`

3. Далее необходимо создать папки под данные, используете значение настроек {SERVICE}_VOLUME_PATH

Для значений по умолчанию:

- `sudo mkdir /var/lib/nginx`
- `sudo mkdir /var/lib/postgresql_data`
- `sudo mkdir /var/lib/minio_data`
- `sudo mkdir /var/lib/mongodb_data`

4. Далее на эти папки необходимо выдать права

Для значений по умолчанию:

- `sudo chmod -R 700 /var/lib`

5. Далее сообщаем docker авторизационные данные от хранилища образов:

- `docker login <NEXUS_PATH>` (адрес, логин и пароль прописан в договоре)

6. Запускаем приложение

- `sudo docker-compose up -d`

Обновление системы.

Обновление компонентов системы производится посредством обновление версии тегов в файле .env Для этого, находясь в папке /opt/mess необходимо запустить редактирования файла .env

- `sudo nano .env`

и изменить значение `{SERVICE}_IMAGE_TAG` на новое

- `realese-1.2.0 → realese-1.2.5`

Сохранить изменения (CTRL+S)

Выйти из редактирования (CTRL+X)

И перезапустить контейнеры

- `docker-compose up -d`

Остановка и перезапуск системы.

Находясь в папке `/opt/mess`, можно использовать следующие команды.

Остановка системы:

- `docker-compose down`

Перезапуск системы:

- `docker-compose restart`

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

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

Список контейнеров.

Имя	Образ	Примечание
front	<NEXUS_REPOSITORY>/erp-system:\${FRONT_IMAGE_TAG}	Версия образа зависит от значений в файле <code>.env</code>
app	<NEXUS_REPOSITORY>/ax.erp:\${BACK_IMAGE_TAG}	Версия образа зависит от значений в файле <code>.env</code>
postgresDbmess	postgres:15-alpine	-
redis	redis:8.2.1	-
redis-commander	rediscommander/redis-commander:latest	Может быть удален из <code>docker-compose</code> , не влияет на работу. Можно использовать внешний.

minio	minio/minio:RELEASE.2025-09-07T16-13-09Z	-
mongo	mongo:8-noble	-
rabbitmq	rabbitmq:3-management	-

Список контейнеров можно увидеть выполнив команду:

- *sudo docker ps*