

Установка Grav CMS с использованием Docker Compose

1. Создаём папку grav-test. Там создаём файл Dockerfile. Файл содержит сборку Docker-образа для Grav :

```
Dockerfile > ...
FROM php:8.3-apache-bullseye
LABEL maintainer="Andy Miller <rhuk@getgrav.org> (@rhukster)"

# Enable Apache Rewrite + Expires Module
RUN a2enmod rewrite expires && \
    sed -i 's/ServerTokens OS/ServerTokens ProductOnly/g' \
    /etc/apache2/conf-available/security.conf

# Install dependencies
RUN apt-get update && apt-get install -y --no-install-recommends \
    unzip \
    libfreetype6-dev \
    libjpeg62-turbo-dev \
    libpng-dev \
    libyaml-dev \
    libzip4 \
    libzip-dev \
    zlib1g-dev \
    libicu-dev \
    g++ \
    git \
    cron \
    vim \
    && docker-php-ext-install opcache \
    && docker-php-ext-configure intl \
    && docker-php-ext-install intl \
    && docker-php-ext-configure gd --with-freetype --with-jpeg \
    && docker-php-ext-install -j$(nproc) gd \
    && docker-php-ext-install zip \
    && rm -rf /var/lib/apt/lists/*

# Set recommended PHP.ini settings
# see https://secure.php.net/manual/en/opcache.installation.php
RUN { \
    echo 'opcache.memory_consumption=128'; \
    echo 'opcache.interned_strings_buffer=8'; \
    echo 'opcache.max_accelerated_files=4000'; \
    echo 'opcache.revalidate_freq=2'; \
```

```
    echo 'opcache.fast_shutdown=1'; \
    echo 'opcache.enable_cli=1'; \
    echo 'upload_max_filesize=128M'; \
    echo 'post_max_size=128M'; \
    echo 'expose_php=off'; \
    } > /usr/local/etc/php/conf.d/php-recommended.ini

RUN pecl install apcu \
    && pecl install yaml-2.2.4 \
    && docker-php-ext-enable apcu yaml

# Set user to www-data
RUN chown www-data:www-data /var/www
USER www-data

# Define Grav specific version of Grav or use latest stable
ARG GRAV_VERSION=latest

# Install grav
WORKDIR /var/www
RUN curl -o grav-admin.zip -SL https://getgrav.org/download/core/grav-admin/\${GRAV\_VERSION} && \
    unzip grav-admin.zip && \
    mv -T /var/www/grav-admin /var/www/html && \
    rm grav-admin.zip

# Create cron job for Grav maintenance scripts
RUN (crontab -l; echo "* * * * * cd /var/www/html;/usr/local/bin/php bin/grav scheduler 1>> /dev/null 2>&1") | crontab -

# Return to root user
USER root

# Copy init scripts
# COPY docker-entrypoint.sh /entrypoint.sh

# provide container inside image for data persistence
VOLUME ["/var/www/html"]

# ENTRYPOINT ["/entrypoint.sh"]
# CMD ["apache2-foreground"]
CMD ["sh", "-c", "cron && apache2-foreground"]
```

2. Создаём файл docker-compose.yml. В нём содержится конфигурация для запуска Grav CMS

```
docker-compose.yml
1  version: '3.9'
   ▶ Run All Services
2  services:
   ▶ Run Service
3    grav:
4      build: ./
5      ports:
6        - 8080:80
7      volumes:
8        - grav-data:/var/www/html
9  volumes:
10   grav-data:
```

3. Скачиваем zip grav-admin 1.7.0 с официального сайта GRAV

- getgrav.org/download/core/grav-admin/1.7.0 downloads Grav Core with the Admin plugin, at Core v1.7.0

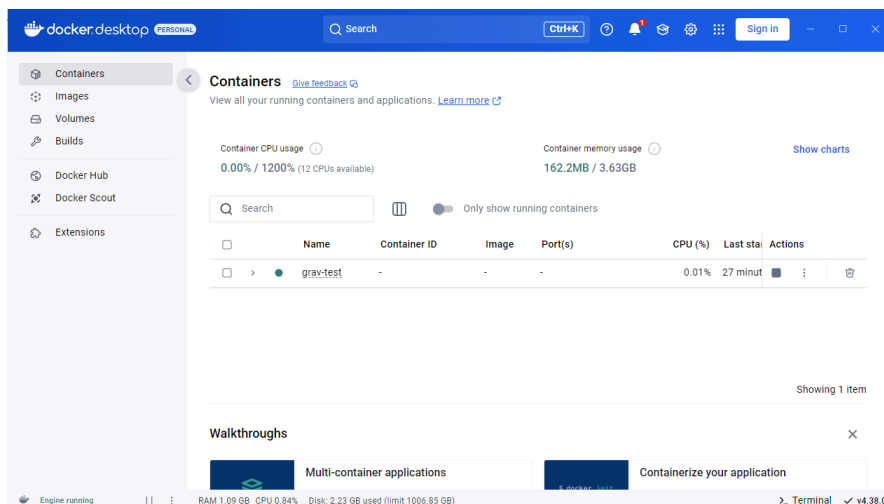
разархивируем содержимое в папку grav-test.

Внутренности папки grav-test

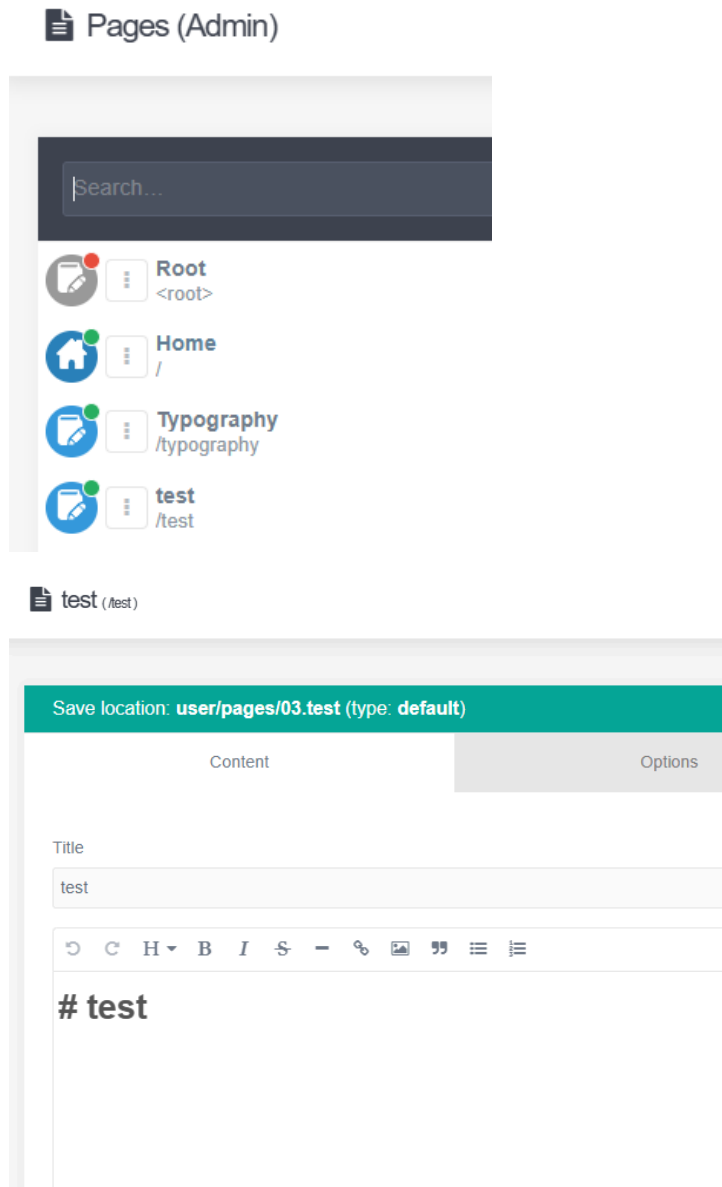
компьютер > Локальный диск (C:) > grav-test				
Имя	Дата изменения	Тип	Размер	
grav	05.03.2025 22:30	Папка с файлами		
docker-compose.yml	05.03.2025 22:33	Исходный файл ...	1 КБ	
Dockerfile	05.03.2025 22:31	Файл	3 КБ	

4. Запускаем Docker Desktop
5. После чего прописываем команду docker-compose up в терминал директории grav-test. В Docker Desktop появился образ и контейнер

```
PS C:\grav-test> docker-compose up
time="2025-03-05T22:24:31+03:00" level=warning msg="C:\grav-test\docker-compose.yml: th
[+] Running 1/1
✓ Container grav-test-grav-1 Created
Attaching to grav-1
grav-1 | AH00558: apache2: Could not reliably determine the server's fully qualified dom
```



6. В браузере переходим на адрес <http://localhost:8080>. Заполняем данные и создаём пробную страницу



Результат работы:

