**Установка Grav CMS с использованием Docker Compose.**

**1. Создание рабочей директории:**

mkdir grav-test

cd grav-test

Создана директория grav-test для хранения файлов конфигурации Docker и файлов Grav.

**2. Создание Dockerfile:**

Создан файл с именем Dockerfile в директории grav-test и добавлен следующий код:

**FROM** php:7.4-apache

**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-2.0.4 \

&& 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"]

**3. Создание docker-compose.yml:**

Создан файл с именем docker-compose.yml в директории grav-test и добавлен следующий код:

version: "3.9"

services:

grav:

build: ./

ports:

- 8080:80

volumes:

- grav-data:/var/www/html

volumes:

grav-data:

Этот docker-compose.yml выполняет следующие действия:

* Определяет сервис grav.
* Указывает Docker Compose построить образ из Dockerfile в текущей директории.
* Публикует порт 80 контейнера на порт 8080 хост-машины.
* Подключает именованный том grav-data к каталогу /var/www/html в контейнере.

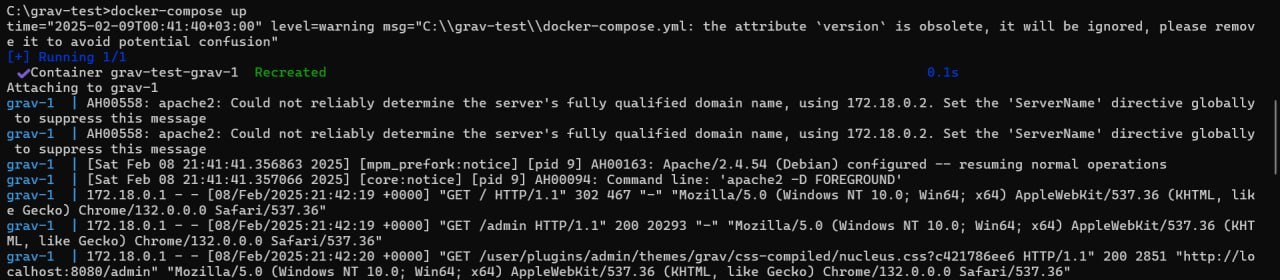
**4. Загрузка Grav:**

Архив скачен и распакован в папку grav-test.  
<https://getgrav.org/download/core/grav-admin/1.7.0>

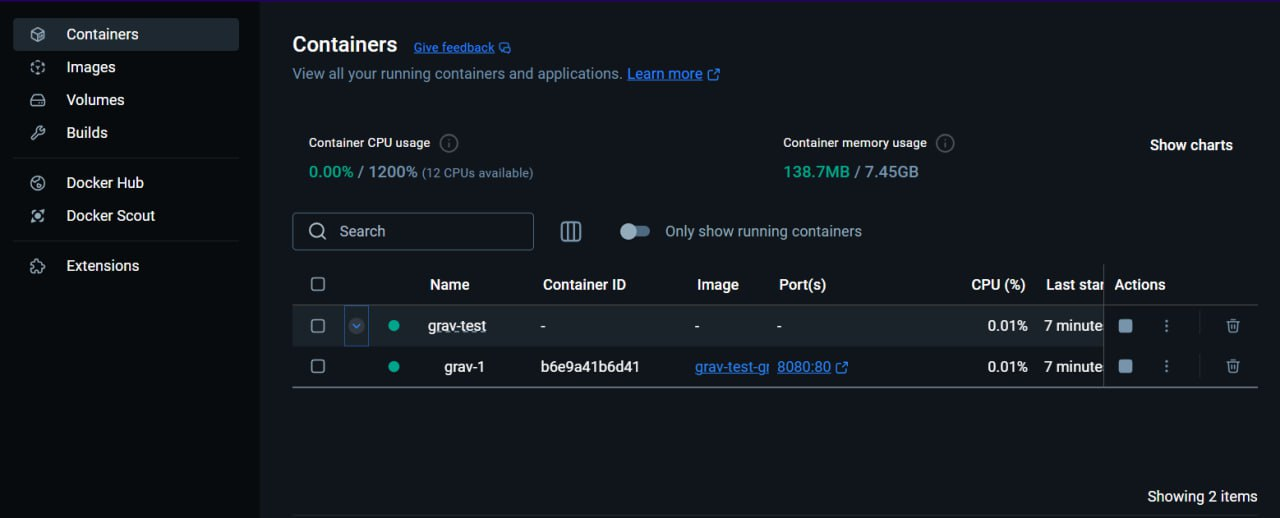
**5. Запуск Docker Compose:**

В командной строке, находясь в директории grav-test, выполнена команда:

docker-compose up

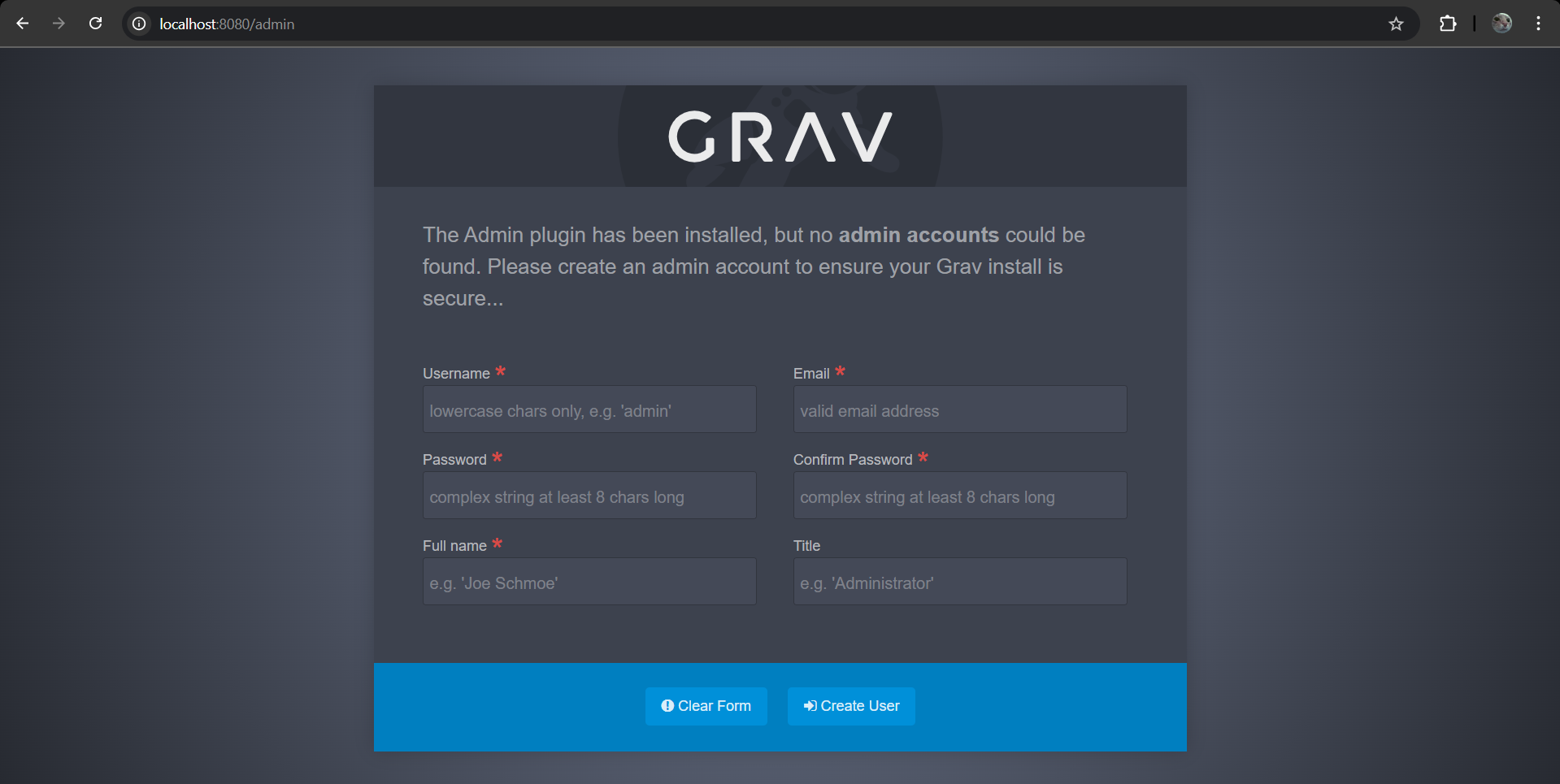


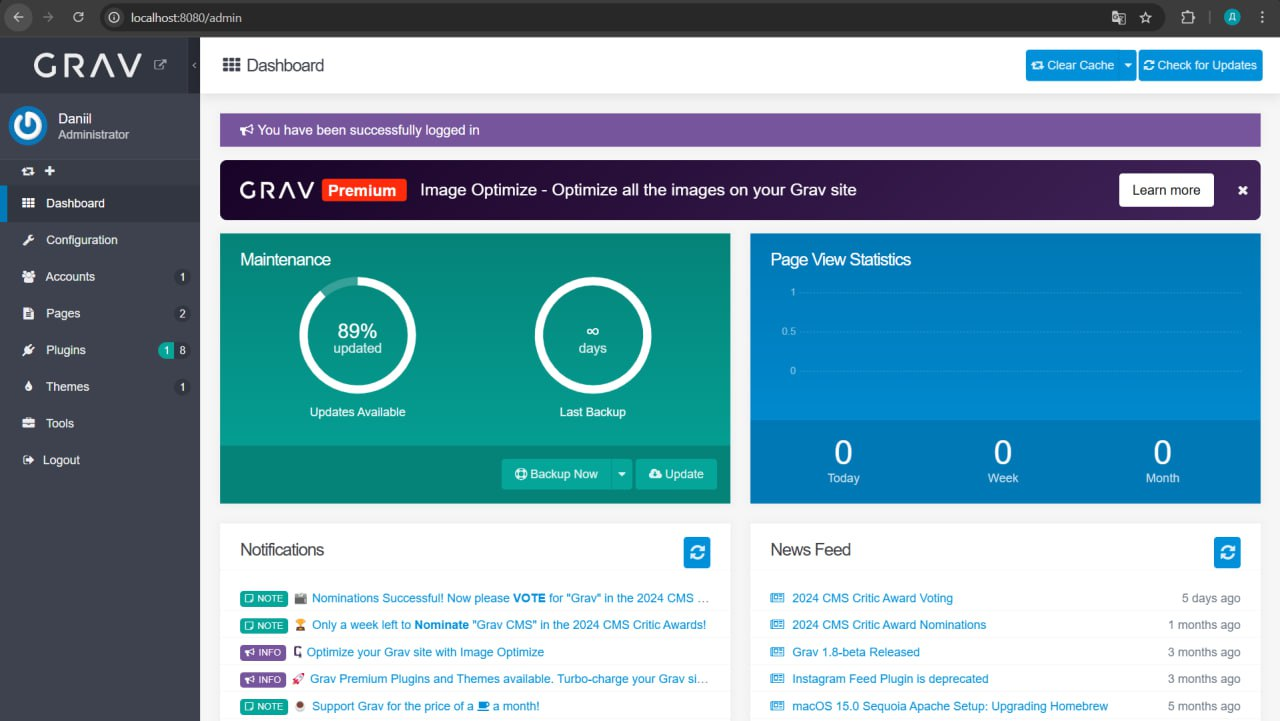
Docker Compose построил образ, создал и запустил контейнер.



**6.** **Открытие сайта и регистрация:**

После успешного запуска контейнера, в браузере открыт адрес http://localhost:8080. Открылась страница установки Grav. Были заполнены необходимые поля для регистрации администратора и создания учетной записи.





**7. Создание пробной страницы:**

После успешной регистрации, в панели администратора была создана новая страница с тестовым контентом.

