**Инсталляция Grav CMS с использованием Docker**

Grav — это современная, быстрая и гибкая система управления контентом (CMS) с открытым исходным кодом, основанная на файлах. В отличие от традиционных CMS, таких как WordPress или Joomla, Grav не требует базы данных для хранения контента. Вместо этого, контент хранится в простых текстовых файлах в формате Markdown, что делает Grav очень легким и быстрым.

Установка выполняется с использованием Docker на Windows 11.

**Этап 1: Создание директории проекта.**

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

**Этап 2: Создание файла Dockerfile.**

Создан файл Dockerfile, содержащий инструкции по сборке Docker-образа для Grav.

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"]

**Этап 3: Создание файла docker-compose.yml.**

Создан файл docker-compose.yml, определяющий конфигурацию сервисов, сетей и томов Docker.

version: "3.9"

services:

  grav:

    build: ./

    ports:

      - 8080:80

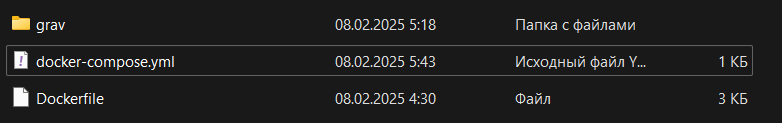
    volumes:

      - grav-data:/var/www/html

volumes:

  grav-data:

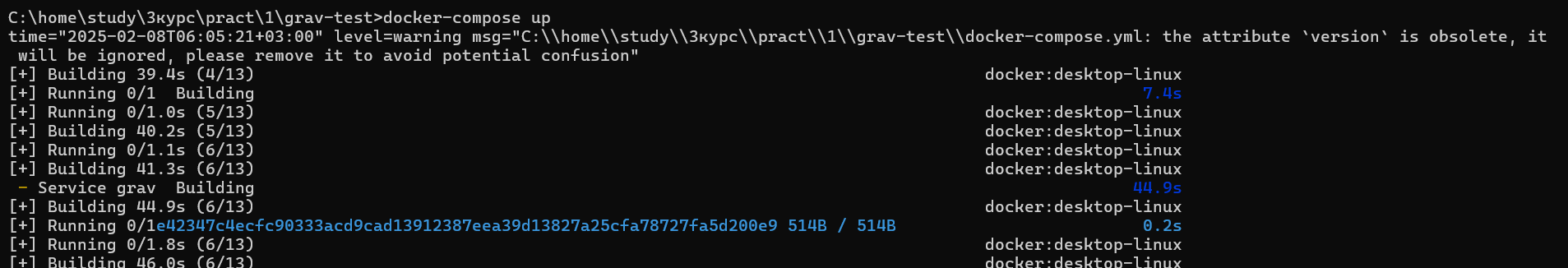
**Структура директории проекта**

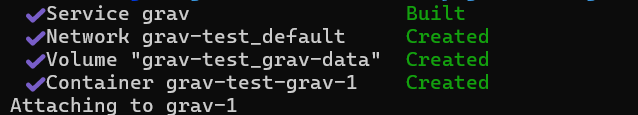


**Этап 4: Запуск Docker Desktop.**

**Этап 5: Запуск Docker Compose.**

В директории проекта выполнена команда docker-compose up для запуска процесса сборки Docker-образа и создания контейнера.

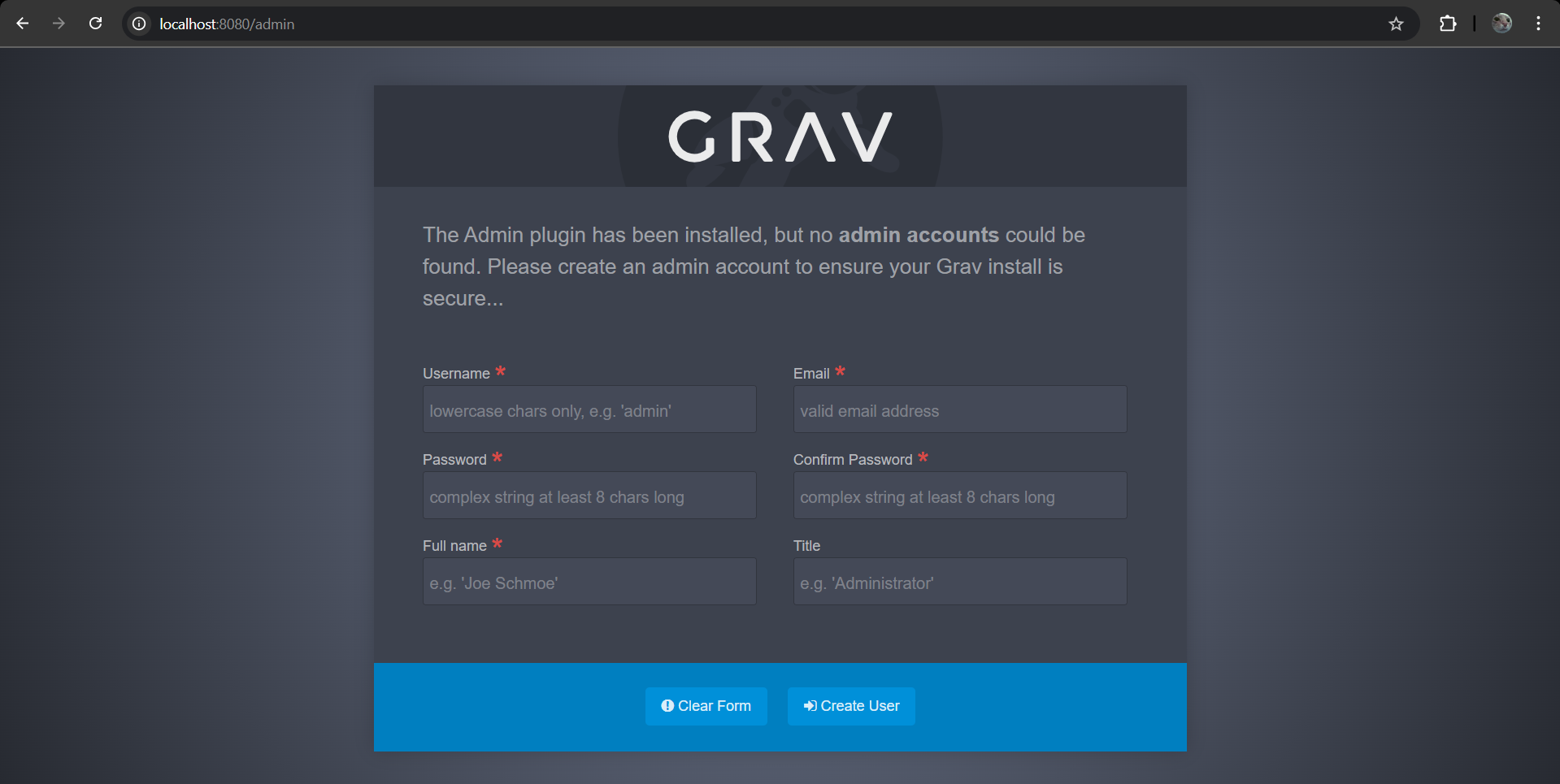




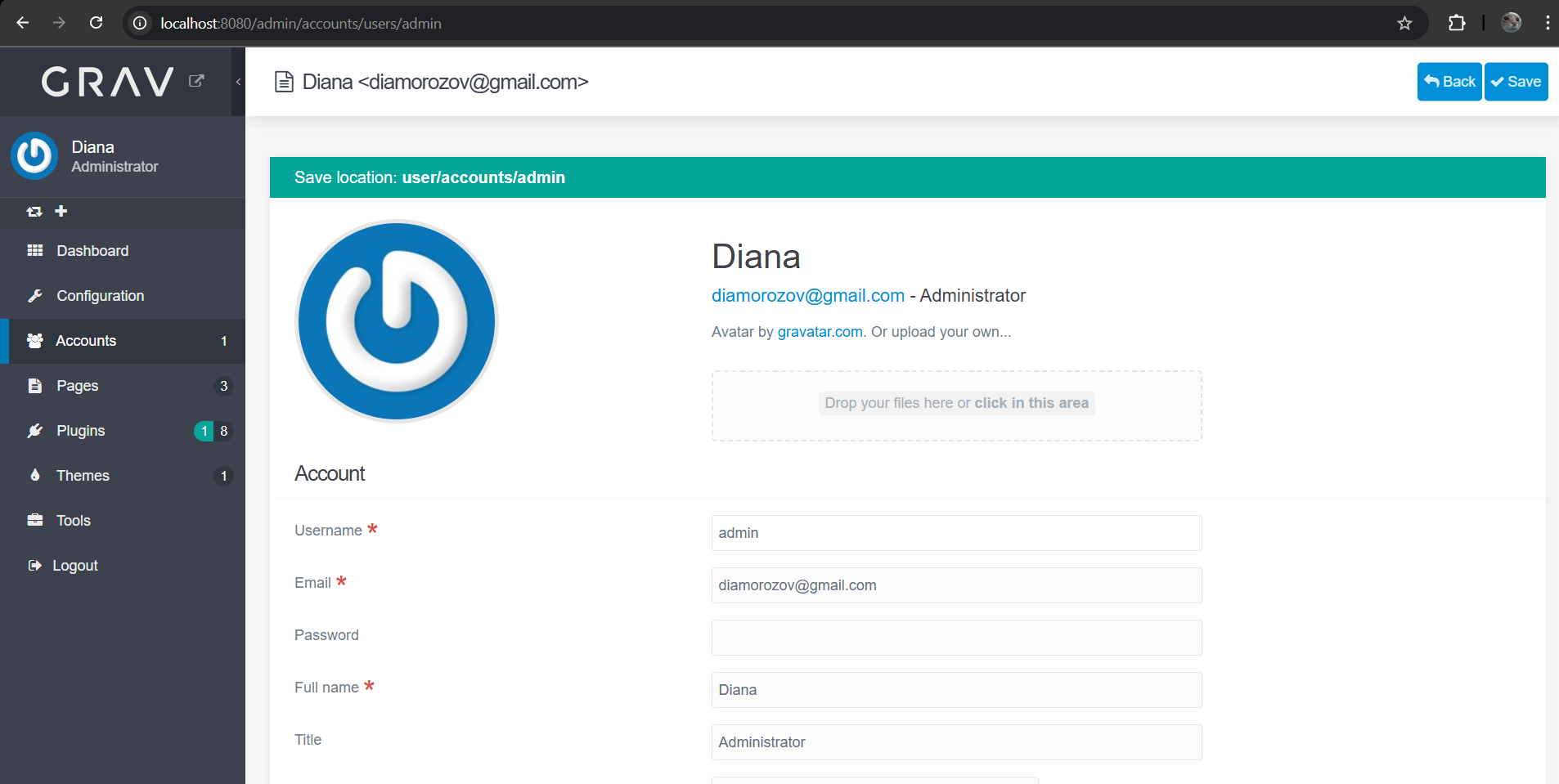
**Этап 6: Проверка работы Grav.**

Grav работает по адресу <http://localhost:8080>

**Регистрация**



**Главная страница**



**Создание страницы**

