ИСР 1.1

**Задача:** провести инсталляцию программного обеспечения.

На основе материалов, опубликованных в примечаниях к заданию, инсталлировать необходимое ПО. В отчете поэтапно продемонстрировать процесс установки (в виде набора скриншотов) с ключевыми моментами установки.

Примечание 1: <https://disk.yandex.ru/i/bHPfXyufLTK0og>

**Выполнение:**

1. Создаём директорию grav-dock, помещаем в неё Dockerfile, docker-compose.yml, grav. Содержимое Dockerfile и docker-compose.yml берём из <https://github.com/getgrav/docker-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"]

docker-compose.yml:

volumes:

  grav-data:

    driver: local

    driver\_opts:

      type: none

      device: ./grav

      o: bind

services:

  grav:

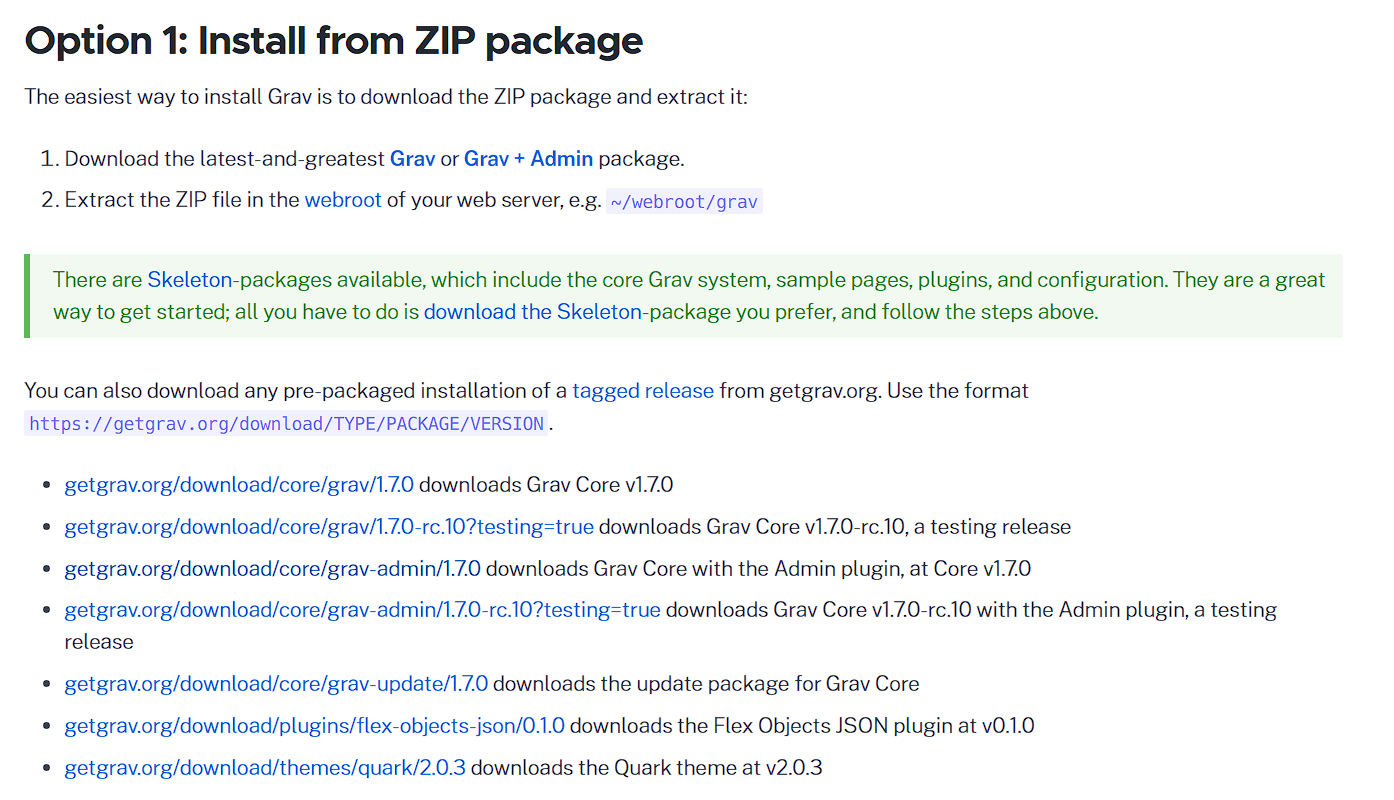
    build: ./

    ports:

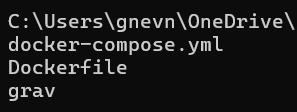
      - 8080:80

    volumes:

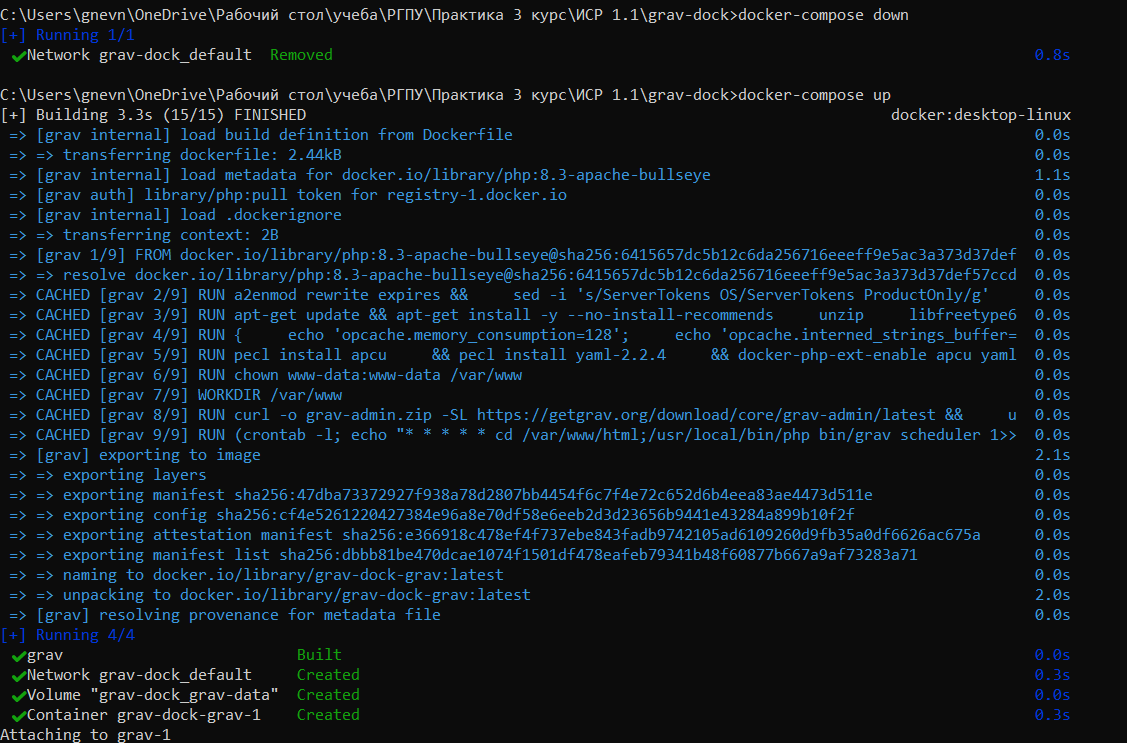
      - grav-data:/var/www/html

Grav скачиваем с официального сайта <https://learn.getgrav.org/17/basics/installation>  


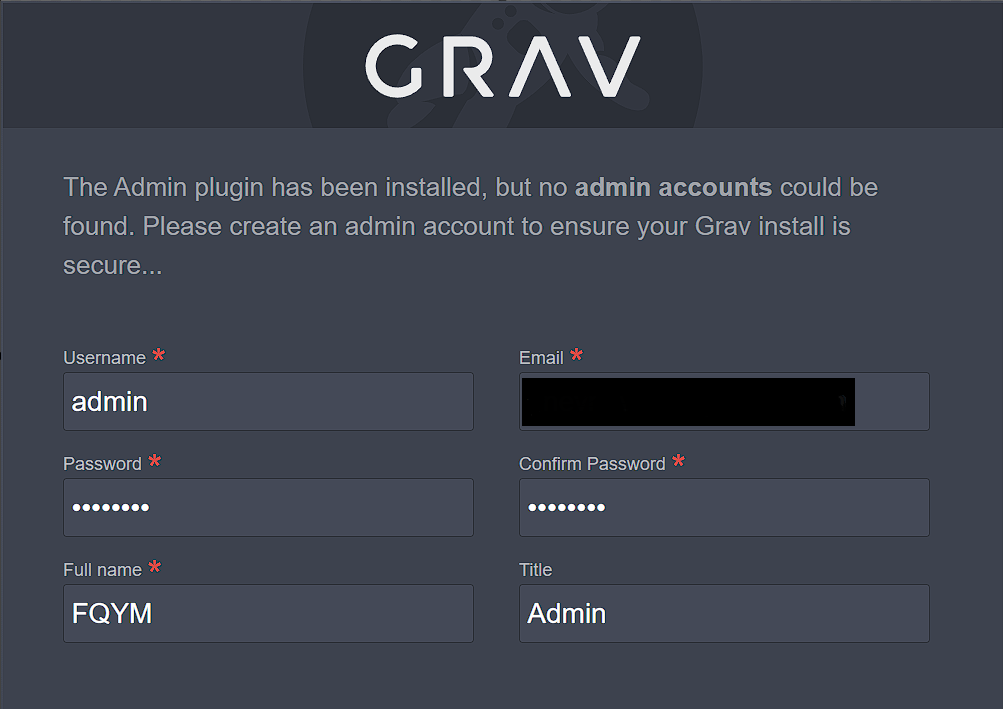
В итоге получаем следующую директорию:



2. Запускаем в командной строке команду docker-compose up



3. Заходим на localhost:8080, создаём администратора:



4. Интерфейс Grav, глазами администратора:

