1. https://disk.yandex.ru/i/W7fck3jSemV50g

При сборке docker-контейнера командой docker-build можно указать параметром -t имя и версию собранного контейнера. Этого можно и не делать, так как докер сам генерирует для контейнера имя. При сборке через docker-compose это имя определяется прямо в файле docker-compose.yml и к нему еще приписывается имя текущего каталога. Когда же нужно давать контейнеру простое и понятное имя - это, например, если этот контейнер будет распространяться через Docker Hub и подключаться к чужим проектам. Таков, например, контейнер Postgres, который скачивается с ДокерХаба. Но пока можно доаольствоваться автосгенерированными именами, которые дает им docker-compose.

1. В файле Dockerfile в папке Django :

|  |  |
| --- | --- |
| FROM python:3.10 |  |
|  | EXPOSE 8000 |
|  | WORKDIR /movies\_admin/ |
|  | COPY requirements.txt /movies\_admin/ - тут копируется requirements в папку movies\_admin? Но есть ведь только папка movies? |
|  | RUN pip install -r requirements.txt --no-cache-dir |
|  | COPY . /movies\_admin/ Аналогичный вопрос – что и куда копируется ? |
|  | CMD ["gunicorn", "config.wsgi:application", "--bind", "0.0.0.0:8000"]  Это комананда выполняется, но как она выглядит если ее просто ввести в командной строке?  Что поднимает этот докерфайл? Я не совсем понял? |

Далее - в Dockerfile строки WORKDIR /movies\_admin и далее COPY requirements.txt /movies\_admin/ создают в корневом каталоге файловой системы контейнера - не локальной - папку /movies\_admin/ и скопируют туда файл requirements.txt из текущего каталога. Создавать эту папку вручную не требуется - ее создание подразумевается. Это папка в файловой системе контейнера, не имеющая отношения к папке movies в локальной системе.

COPY . /movies\_admin/ коприует содержимое текущего каталога в каталог /movies\_admin/ в файловой системе контейнера. Файл requirements.txt копируется раньше и отдельно, чтобы докер мог кэшировать результаты команды pip install -r requirements.txt, которая выполняется очень долго и перезапускать ее только если изменился requirements.txt. Изменения других файлов перевыполнения этой команды не требуют.

Строки в списке соеднияются через пробелы — gunicorn config.wsgi:application —bind 0.0.0.0:8000

**ЧТО ДЕЛАЕТ ЭТА КОМАНДА?**

**Gunicorn — это web-сервер, работающий примерно как встроенный Web-сервер Django. Первый параметр — это ссылка на питоноский объект представляющий собой Django — приложение. config.wsgi:application — значит приложение представлено объектом application в файле config/wsgi.py. Параметр —bind 0.0.0.0:8000 означает, что сервер слушает порт 8000. Это значение и нужно менять, чтобы слушать другой порт.**

1. В файле Django\_env :

|  |  |  |
| --- | --- | --- |
| DB\_HOST=127.0.0.1 |  |  |
|  | DB\_PORT=5432 |  |
| **Почему именно такие порт и хост?** | |  |
|  |  |  |

5432 — это порт Постгреса по умолчанию. С тем же успехом можно использовать любой, только прописать его же и в настройках Постгреса. 127.0.0.1 — это потому что Постгрес выполняется на той же машине что и Джанго, адрес локального хоста. Если не указывать network\_mode: host, то имя хоста придется указывать другое, так как в этом случае они будут выполняться в разных виртуалках и нужно будет указать Джанго адрес виртуалки с Постгресом

**ПОЧЕМУ ИМЕННО ТАКОЙ АДРЕС ПОРТА 127.0.0.1?**

**IP-адрес 127.0.0.1 означает подключение к серверу на той же системе что и клиент.**

1. В файле:

**docker-compose.yml:**

**Почему для сервиса джанго есть блок build а для постгре нет?**

**Что означает context: Django?**

**Почему для постгре порт 5432 а для джанго 8000? Почему именно такие значения?**

**Где взять вот эту инфу для джанги ?:** static\_volume:/data/static

**Почему для postgre такой image -** postgres:13-alpine?(где взять это название) и почему не указан image для джанго?

Блок build означает, что докер собирает файловую систему контейнера, копируя в нее локальные файлы. Процесс сборки при этом описывается в файле формата Dockerfile. Если блока build нет — значит используется предварительно собранный контейнер,

**А ГДЕ ДЛЯ ПОСТГРЕССА ПРЕДВАРИТЕЛЬНО СОБРАННЫЙ КОНТЕЙНЕР?**

**Он общедоступен, выложен в сеть на сайте dockerhub. При первом запуске системы он скачивался оттуда, теперь сохранен локально где-то в кэше Докера.**

который скачивается с Докерхаба. Параметр context указывает каталог, в котором находится Dockerfile и файлы, которые копируются в контейнер. Значения портов — это те что по умолчанию прослушивает сервер Postgres и тестовый сервер Django. Их можно и поменять — 8000 на 8010, например, и запускать сервер чтобы прослушивал порт 8010.