|  |  |
| --- | --- |
| Команда | Описание |
| Урок «Запуск приложений в Docker контейнерах» | |
| docker | Выводит список всех доступных команд |
| docker run --name hello-container hello-world | * docker - вызов докер-клиента * run - команда для докер-клиента «Запусить контейнер» * --name hello-container - присваиваем имя запускаемому докер-контейнеру. * hello-world - имя образа из которого будет запущен контейнер |
| docker run --name nginx-container nginx | Запуск контейнера из образа с сервером nginx |
| docker container rm nginx-container | Удалить nginx-contaioner (По имени контейнера) |
| docker run -p 80:80 --name nginx-container nginx | -p 80:80 - пробросить порт 80 хоста на порт 80 контейнера |
| docker container ls | Просмотр запущенных контейнеров |
| docker container stop nginx-container | Остановить запущенный контейнер (По имени контейнера) |
| docker container ls -a | Вывести список всех контейнеров (в т.ч остановленных) |
| docker container start nginx-container | Запуск остановленного контейнера (По имени контейнера) |
| docker exec -it <CONTAINER ID> any\_instruction | Команда для входа в контейнер и выполнения команд внутри контейнера   * exec - запустить команду внутри контейнера * -it - аргумент для выполнения команды в интерактивном режиме. С аргументом -it все команды и ответы контейнера будут работать так, будто вы запустили команду не в контейнере, а в терминале своей операционной системы. * any\_instruction - команда, которая будет выполнена в терминале контейнера. |
| docker exec -it nginx-container bash | Пример команды описанной выше |
| echo 'alias docker="MSYS\_NO\_PATHCONV=1 docker"' >> ~/.bash\_profile | Команда-настройка, что бы git-bash не заменял имена файлов при вызове команд docker |
| echo 'alias docker="winpty docker"' >> .bash\_profile | Команда-настройка, что бы команды docker нормально работали в git-bash |
| docker container rm hello-container | Удаление контейнера (По имени контейнера) |
| docker image ls | Посмотреть список образов сохраненных локально |
| docker image rm <image\_name> | Команда для удаления образа |
| Урок «Упаковка проекта в Docker образ» | |
| docker build -t taski\_backend . | Билдим образ:  **build** - команда сборки образа по инструкциям из Dockerfile.  **-t** taski\_backend - задает имя образу.  **.** - путь до папки с Dockerfile, на основе которого производится сборка. Точка — это текущая директория в терминале. |
| docker run --name taski\_backend\_container --rm -p 8000:8000 taski\_backend | Запуск контейнера из образа созданного командой выше |
| docker exec taski\_backend\_container python manage.py migrate | Выполняем миграции внутри запущенного выше контейнера |
| docker build -t taski\_frontend . | Билдим образ фронтенд приложения taski |
| docker run --rm -it -p 8000:8000 --name taski\_frontend\_test taski\_frontend | Запуск контейнера из образа созданного командой выше   * --rm - указывает, что контейнер должен быть удален сразу после остановки, подходит для экспериментов * -it - запускает контейнер в интерактивном режиме |
| Урок «Сервер gunicorn и docker volume» | |
| docker volume create sqlite\_data | Создается постоянное хранилище «sqlite\_data» |
| docker run --name taski\_backend\_container -p 8000:8000 -v sqlite\_data:/data taski\_backend | Запуск контейнер с docker volume:   * --name taski\_backend\_container - присваиваем контейнеру имя * -p 8000:8000 - связываем публичный порт контрейнера с портом хоста * **-v - связываем volume с директорией внутри контейнера: имя\_volume:директория\_в\_контейнере** |
| docker volume rm sqlite\_data | Удаления тома (volume) sqlite\_data |
| Урок «Сервер gunicorn и docker volume» | |
| docker login -u username | Аутентифицируем докер-демон на Docker-Hub (команда не работает через Git-Bash) |
| docker push username/taski\_backend:latest | Запушить в Docker-Hub |
| Урок «PostgreSQL и Docker network» | |
| docker run --name db --env-file .env postgres:13.10 | Пример команды для запуска контейнера с использованием файла с переменными окружения |
| docker volume create pg\_data | Создание volume для хранения данных PostgreSQL |
| docker run ... -v pg\_data:/var/lib/postgresql/data | Пример команды для связи volume pg\_data с директорией /var/lib/postgresql/data |
| docker run --name db \ --env-file .env \ -v pg\_data:/var/lib/postgresql/data \ postgres:13.10 | * Символ \ в конце строки указывает терминалу, что команда продолжится на следующей строке * Запустить контейнер с именем db * Передать в контейнер переменные окружения из файла .env * Подключить Docker volume с названием pg\_data, * Контейнер создать из образа postgres с тегом 13.10 |
| docker exec -it db psql -U django\_user -d django | Запуск утилиты для запросов к БД PostgreSQL через терминал   * exec — выполнить команду в запущенном контейнере; * -it — запуск в интерактивном режиме; * db — имя контейнера, в котором нужно выполнить команду; * psql — имя утилиты, которую нужно запустить в контейнере; * -U django\_user — пользователь, от имени которого psql подключится к базе данных; * -d django — имя базы данных, к которой нужно подключиться. |
| * \l - выводит список баз данных на сервере * \dt - выводи список таблиц в текущей БД | Набор полезных команд в psql |
| Ctrl + D | Выход из psql |
| docker network create django-network | Создаем сеть с названием django-network |
| docker network connect django-network db | Присоединить к сети django-network контейнер db |
| Урок «Docker compose: сборка проекта» | |
| docker compose version | Узнать какая версия Docker compose установлена на компьютер |
| docker compose up | Запустить весь орекстр (Выполнить из директории с файлом docker-compose.yml) |
| docker compose exec backend python manage.py migrate | Здесь мы проверили сработал ли docker compose, выполнили миграции |
|  | |
| docker compose up --build | Означает, что перед запуском надо пересобрать образы |
| Урок «Деплой: публикация проекта в Docker на сервере» | |
| docker compose down | Удаление артифактов после предыдущих запусков Docker compose, но volumes остаются |
| docker compose down -v | Команда аналогичная предыдущей, но удаляет и volumes тоже |
| docker compose -f docker-compose.production.yml up | Запускаем Docker compose с недефолтным названием файла конфигурации |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |