Как подключиться по ssh к контейнеру (не получилось) -

<https://itsecforu.ru/2021/05/31/%F0%9F%90%B3-%D0%BA%D0%B0%D0%BA-%D0%BF%D0%BE%D0%B4%D0%BA%D0%BB%D1%8E%D1%87%D0%B8%D1%82%D1%8C%D1%81%D1%8F-%D0%BA-%D0%BA%D0%BE%D0%BD%D1%82%D0%B5%D0%B9%D0%BD%D0%B5%D1%80%D1%83-docker-%D0%BF%D0%BE-ssh/>

docker inspect -f "{{ .NetworkSettings.IPAddress }}" nginx – узнать ip адрес контейнера

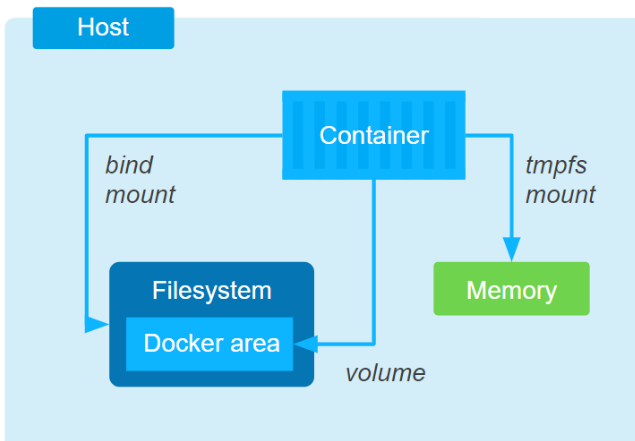
Изменить параметры контейнера:

docker update --restart=no airflow\_docker\_airflow\_1

**Хранение данных**

В Docker есть несколько способов хранения данных. Наиболее распространенные:

* тома хранения данных (docker volumes),
* монтирование каталогов с хоста (bind mount).



docker volumes

В Linux тома находятся по умолчанию в /var/lib/docker/volumes

docker volume ls –показать тома

Тома создаются и управляются средствами Docker: командой docker volume create, через указание тома при создании контейнера в Dockerfile или docker-compose.yml.

Один том может быть примонтирован одновременно в несколько контейнеров.

bind mount

файл или каталог с хоста просто монтируется в контейнер.

Используется, когда нужно пробросить в контейнер конфигурационные файлы с хоста. Например, именно так в контейнерах реализуется DNS: с хоста монтируется файл /etc/resolv.conf.

Docker DNS

bridge

**Многоэтапные сборки**

Dockerfile:

# первый этап  
FROM python:3.8 AS builder  
COPY requirements.txt .

# установка зависимостей в локальную директорию user (например, /root/.local)  
RUN pip install --user -r requirements.txt

# второй этап (без названия)  
FROM python:3.8-slim  
WORKDIR /code

# копирование только установки зависимостей из образа первого этапа   
COPY --from=builder /root/.local/bin /root/.local  
COPY ./src .

# обновление переменной среды PATH  
ENV PATH=/root/.local:$PATHCMD [ "python", "./server.py" ]

При этом в requirements располагаются все зависимости (доп. Библиотеки)

*requirements.txt*

Flask==1.1.1

**Docker compose**

С помощью команды docker-compose up можно запустить ряд контейнеров, правила запуска будут описываться в файле docker-compose.yml

1. Будет ли после docker-compose вызываться yml каждый раз при перезапуске?

Похоже нет. Перезапускаются контейнеры в своем порядке со своими свойствами. Поэтому останавливаем и меняем отдельно каждый контейнер. Можно вызывать другие функции docker-compose: down, kill, events, logs, stop(services)

**Пример:**

version: "3" #версия docker compose

services: # сервисов(контейнеров)

airflow\_db: #название контейнера

image: postgres:12-alpine

restart: always #перезапускает при включении docker’a

environment: #переменные среды

POSTGRES\_DB: airflow

POSTGRES\_USER: airflow

POSTGRES\_PASSWORD: airflow

volumes:

- airflow\_database:/var/lib/postgresql/data/

airflow:

build: #в какой папке лежит dockerfile

context: airflow

restart: always

volumes:

- ./airflow:/usr/local/airflow

- airflow\_logs:/var/log/airflow

- airflow\_data:/var/spool/airflow

env\_file: #путь до файла с переменными среды

- .env

ports:

- 8080:8080

depends\_on: #дождаться пока не запустится данный контейнер

- airflow\_db

prices\_db:

image: postgres:12-alpine

restart: always

environment:

POSTGRES\_DB: stocks

POSTGRES\_USER: trader

POSTGRES\_PASSWORD: income

volumes:

- prices\_database:/var/lib/postgresql/data/

ports:

- 5433:5432 #5433 указал, чтобы не было конфликтов с первой БД для airflow, которая крутится тоже на 5432

volumes:

prices\_database:

airflow\_database:

airflow\_data:

airflow\_logs: