

Docker

Как стать капитаном грузового судна

Road map

1. Проблематика и мотивация
2. Немного теории
3. Примеры

Лабы

Гайды

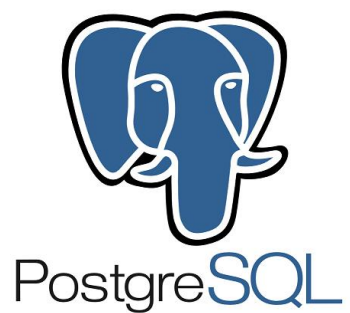
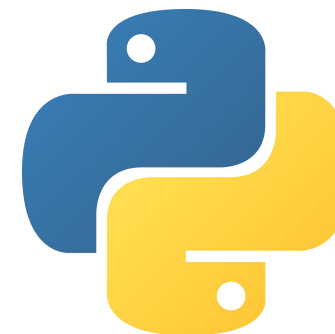
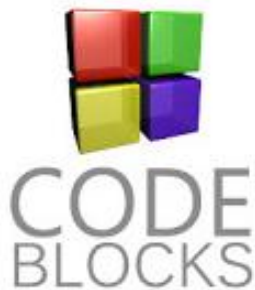
Фриланс

Работа

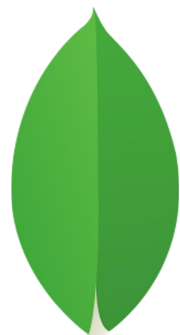
Проекты



4



mongoDB®



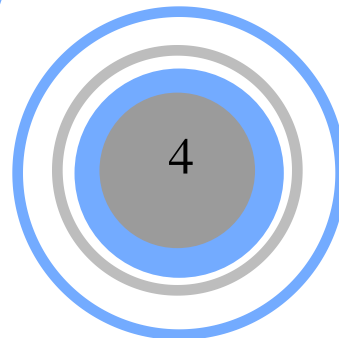
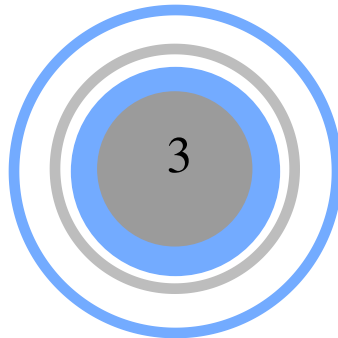
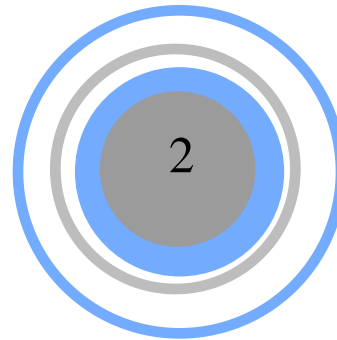
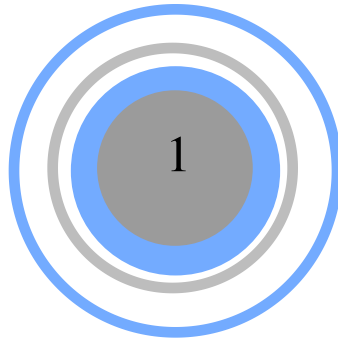
К чему это приводит?



Проблема - курсовой проект на двоих

Установить по
гайду Node,
Python, Java или
что-то другое

Но не сказать
версию

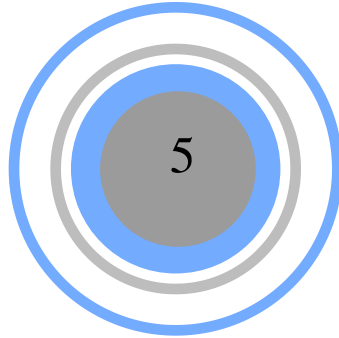


Рассказать идею
другу и попросить
его установить то
же

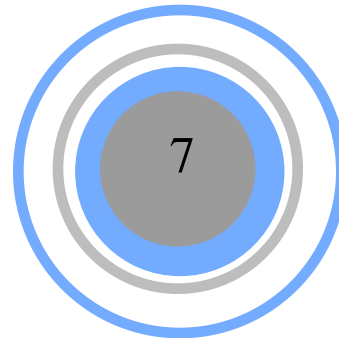
Пошарить код с
другом

Проблема - курсовой проект на двоих

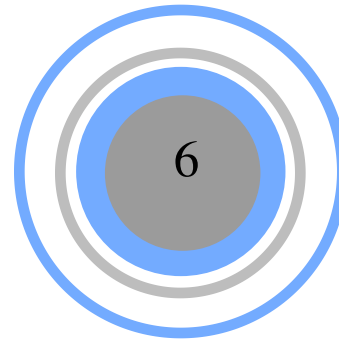
Помочь его
запустить



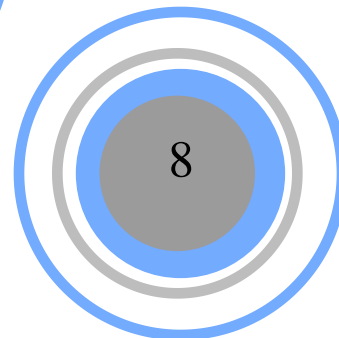
С ранее
установленными
утилитами
других версий



На другой версии
платформы



На другой
операционной
системе

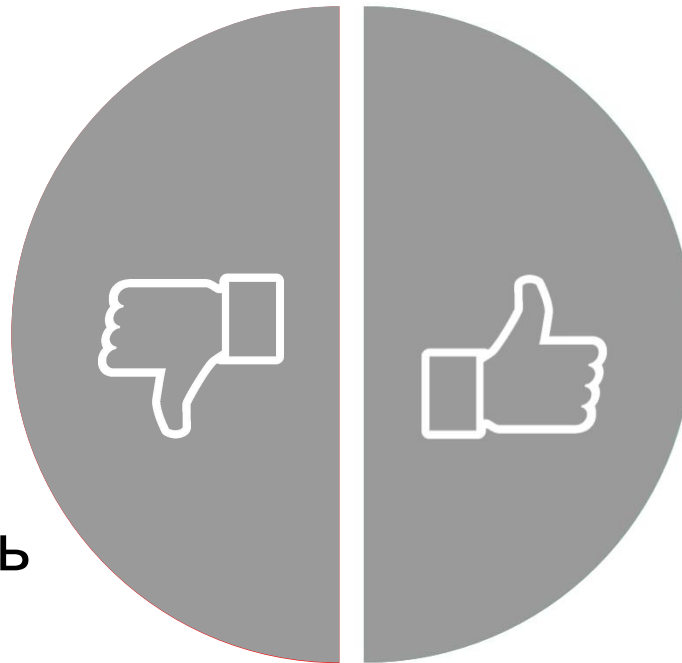


Решение

Быть жадной и
социапатом

Сделать у себя как у
друга

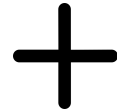
Попросить друга сделать
как у себя



Использовать Docker

Ещё примеры ?

Лабы по базам данных
на 8 MySQL с
использованием
последнего JSON API



Новый проект на работе
- проверенный
MySQL:5.7



Трата года на то чтобы
обе версии работали
рядом

Переустановка ОС
чтобы начать все
сначала

Решение

2 ПК - один для учебы,
2й для работы

3й ПК для
эмулирование прод
сервера на убунту



Использовать Docker

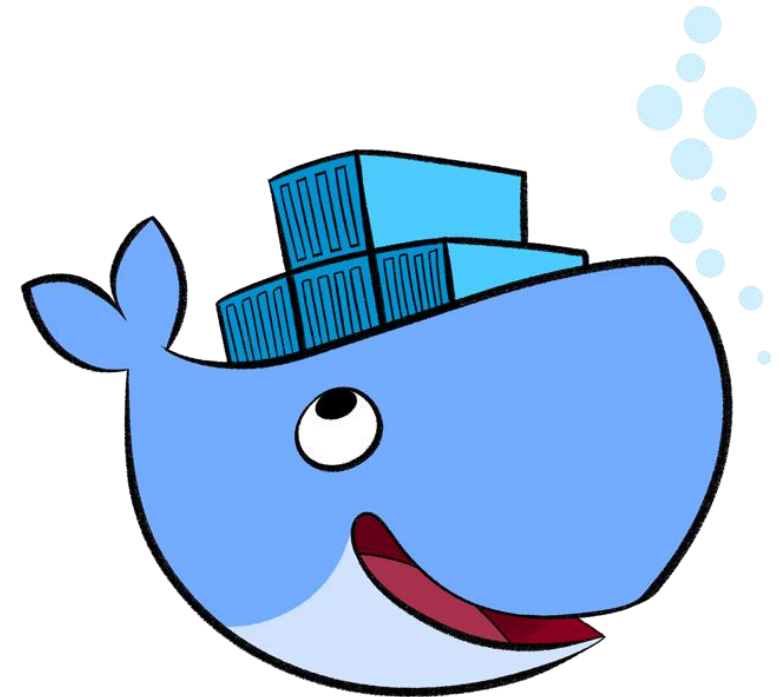
Docker. Плюсы. Что он может?

Запускать ПО в контейнере (PostgreSQL, AWS localstack).

Контейнер не оставляет за собой следов после остановки или удаления.

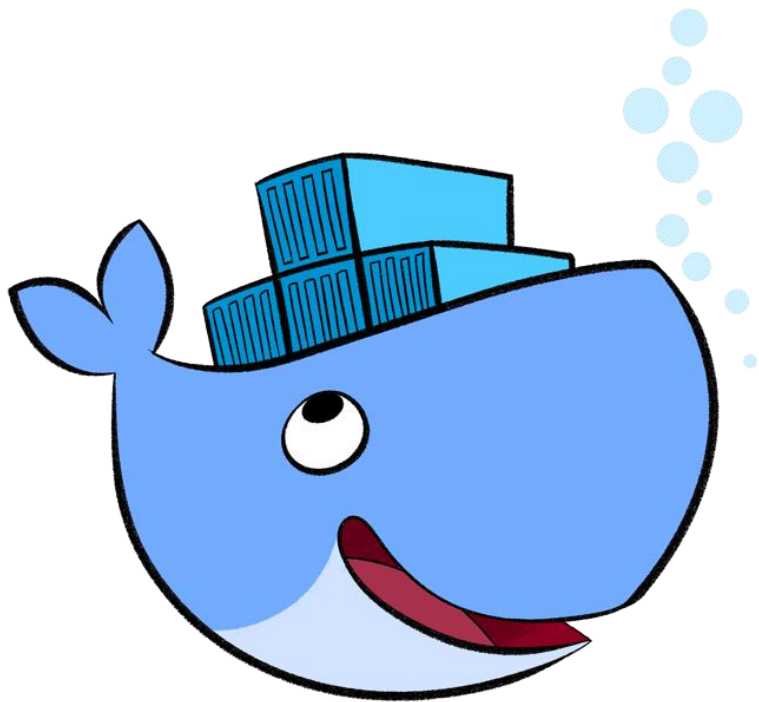
Настраивать ресурсы – Mongo никогда больше не съест весь процессор

Заворачивать свои приложения в удобный контейнер. Если работает у тебя – работает и у друга



Докер. Плюсы. Что он может?

Запускать по несколько сервисов сразу с помощью docker-compose



Когда мало места - можно удалить все контейнеры и образы освободив место.

Использовать внутренние сети без внешнего доступа. А если сильно захотеть то и без внутреннего к внешним сетям

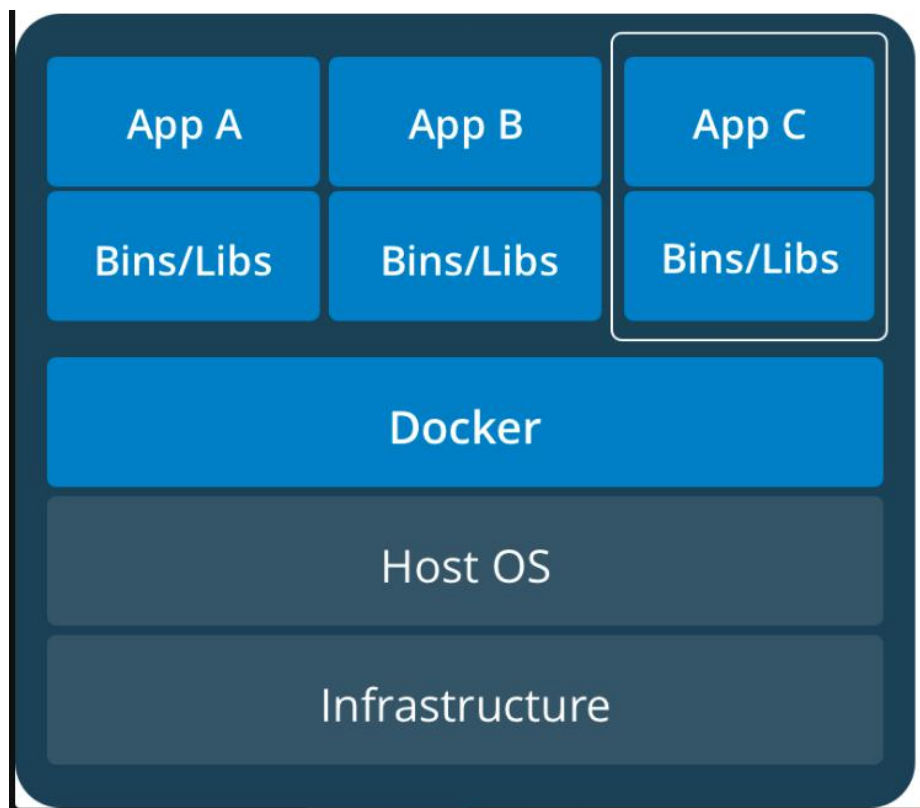
Шарить файлы между контейнерами и вашей OS

Немного теории или что должен знать капитан

1. Что такое Docker
2. Как его использовать



Читаем справку на трапе



Docker является open source инструментом, который автоматизирует разворачивание приложения внутри программного контейнера.

Ключевое преимущество Docker в том, что он позволяет пользователям упаковать приложение со всеми его зависимостями в стандартизированный модуль для разработки – контейнер, для старта работы которого в любом другом месте нужно иметь только установленный Docker

Зачем использовать Docker?

01

Быстрый процесс разработки

02

Удобная инкапсуляция приложений

03

Одинаковое поведение

04

Простой и понятный мониторинг

05

Легко масштабируется

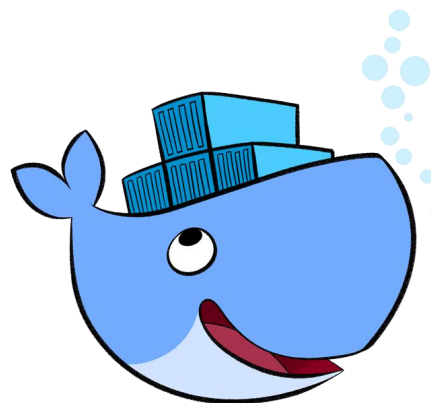
Основные элементы Docker

Volume

выделенное место для хранения файлов

Network

сеть внутри докера, для общения между группой контейнеров



Registry

служит для удаленного хранения образов(Image) контейнеров

Image

образ контейнера

Container

собранное и готовое к работе приложение

Docker Hub - публичный репозиторий

**ubuntu**

Updated 7 minutes ago

Ubuntu is a Debian-based Linux operating system based on free software.

Container Linux ARM PowerPC 64 LE IBM Z x86-64 ARM 64 386 Base Images Operating Systems

OFFICIAL IMAGE

10M+ Downloads 10K+ Stars

**redis**

Updated 7 minutes ago

Redis is an open source key-value store that functions as a data structure server.

Container Linux Windows ARM ARM 64 PowerPC 64 LE x86-64 386 IBM Z Databases

OFFICIAL IMAGE

10M+ Downloads 7.7K Stars

**node**

Updated 7 minutes ago

Node.js is a JavaScript-based platform for server-side and networking applications.

Container Linux ARM ARM 64 IBM Z x86-64 386 PowerPC 64 LE Application Infrastructure

OFFICIAL IMAGE

10M+ Downloads 8.3K Stars

**mysql**

Updated 7 minutes ago

MySQL is a widely used, open-source relational database management system (RDBMS).

Container Linux x86-64 Databases

OFFICIAL IMAGE

10M+ Downloads 9.0K Stars

**traefik**

Updated 7 minutes ago

Traefik, The Cloud Native Edge Router

OFFICIAL IMAGE

10M+ Downloads 1.2K Stars

**nginx**

Updated 7 minutes ago

Official build of Nginx.

Container Linux ARM ARM 64 386 PowerPC 64 LE x86-64 IBM Z Application Infrastructure

OFFICIAL IMAGE

10M+ Downloads 10K+ Stars

**registry**

Updated 7 minutes ago

The Docker Registry 2.0 implementation for storing and distributing Docker images

Container Linux ARM 64 ARM x86-64 Featured Images Storage Application Services

OFFICIAL IMAGE

10M+ Downloads 2.8K Stars

**httpd**

Updated 7 minutes ago

The Apache HTTP Server Project

Container Linux 386 PowerPC 64 LE ARM 64 IBM Z x86-64 ARM Application Infrastructure

OFFICIAL IMAGE

10M+ Downloads 2.8K Stars

**golang**

Updated 7 minutes ago

Go (golang) is a general purpose, higher-level, imperative programming language.

Container Linux Windows ARM ARM 64 386 PowerPC 64 LE x86-64 IBM Z Programming Languages

OFFICIAL IMAGE

10M+ Downloads 3.0K Stars

**python**

Updated 7 minutes ago

Python is an interpreted, interactive, object-oriented, open-source programming language.

Container Linux Windows IBM Z PowerPC 64 LE 386 ARM 64 ARM x86-64 Programming Languages

OFFICIAL IMAGE

10M+ Downloads 4.9K Stars

OFFICIAL IMAGE

Что может сделать юнга?

1. Узнать, работает ли докер

```
(venv) anton@dell:~/prosvita/2020-docker/docker-examples/app$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:9572f7cdcee8591948c2963463447a53466950b3fc15a247fcad1917ca215a2f
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

Что может сделать юнга?

2. Осмотреть палубу

```
(venv) anton@dell:~/prosvita/2020-docker/docker-examples/app$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
801081199f50	prosvita-back:latest	"docker-entrypoint.s..."	3 days ago	Up 3 days	0.0.0.0:3000->3000/tcp	prosvita_back_1
be0a6927952c	mongo:4.2	"docker-entrypoint.s..."	3 days ago	Up 3 days	0.0.0.0:27019->27017/tcp	prosvita_mongo_1
da2231ec130c	prosvita-front:latest	"nginx -g 'daemon of..."	3 days ago	Up 3 days	0.0.0.0:3010->80/tcp	prosvita_front_1

```
(venv) anton@dell:~/prosvita/2020-docker/docker-examples/app$
```

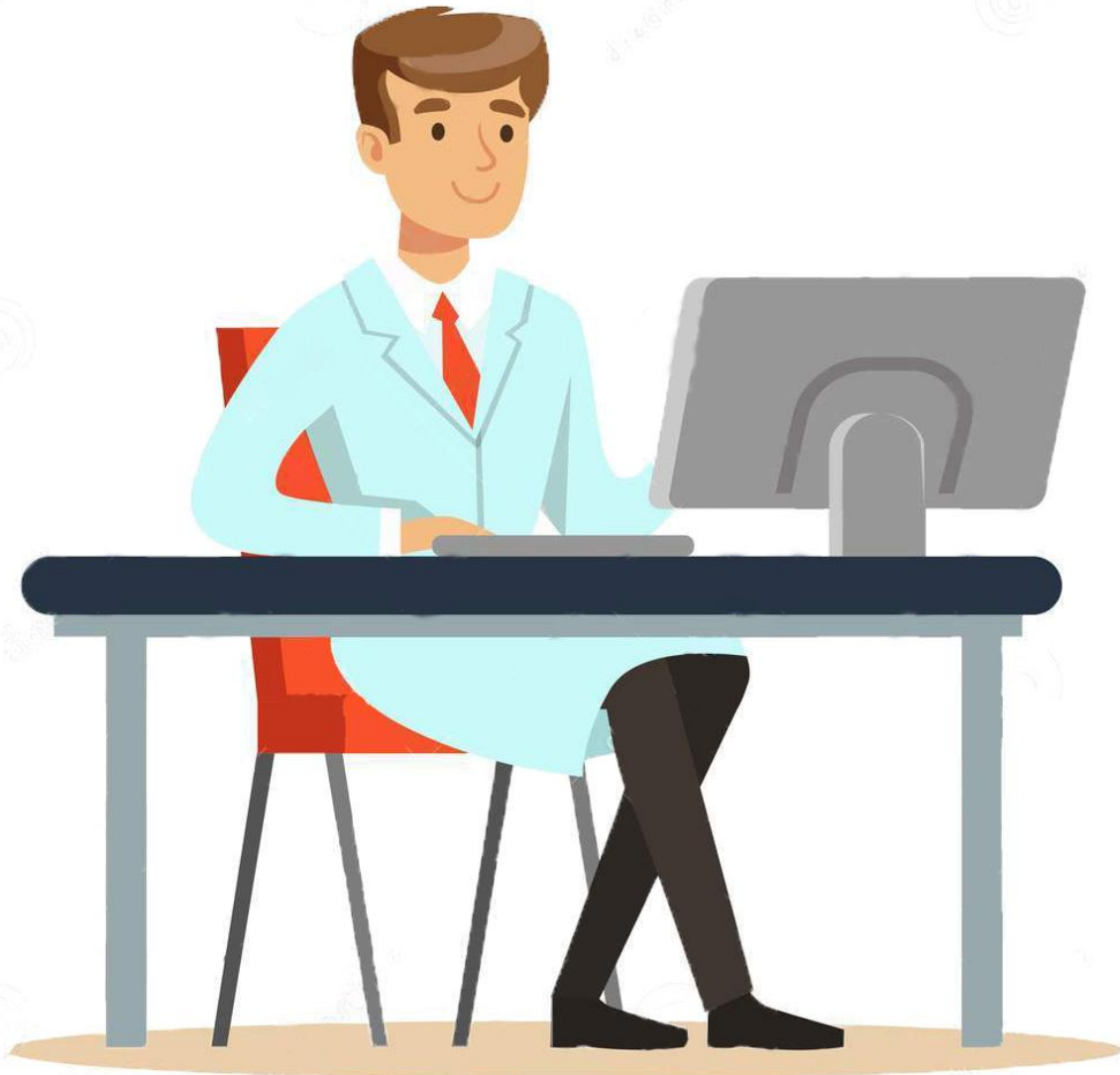
3. Убрать лишнее

```
(venv) anton@dell:~/prosvita/2020-docker/docker-examples/app$ docker rm -f prosvita_back_1 prosvita_mongo_1 prosvita_front_1
prosvita_back_1
prosvita_mongo_1
prosvita_front_1
(venv) anton@dell:~/prosvita/2020-docker/docker-examples/app$
```

Груз на палубе - задача матроса

1. Написать приложение или скрипт
2. Описать сборку в образ (image) с помощью Dockerfile
3. Собрать образ
4. Запустить это всё
5. Проверить что бы работало и отладить

Шаги создания на примере



Получение статистики по
продажам абстрактных
продуктов

1. Простой скрипт на Python для получения некоторой статистики по продажам абстрактных продуктов

```
import os
import pandas as pd
from pathlib import Path

root_app_path = path = Path(__file__).absolute().parent.parent

def find_files():
    files = []
    for entry in os.scandir(root_app_path.joinpath('./store')):
        if entry.is_file() and entry.path.endswith('.csv'):
            files.append(entry.name)

    return files

def load_data(files):
    data_frames = []
    for file_name in files:
        data_frames.append(pd.read_csv(root_app_path.joinpath('./store/%s' % file_name)))

    return data_frames

files = find_files()
data_frames = load_data(files)

for i, df in enumerate(data_frames):
    print({
        'Name': files[i],
        'Count': df['Sold'].count(),
        'Min': df['Sold'].min(),
        'Max': df['Sold'].max(),
        'Mean': df['Sold'].mean(),
        'Median': df['Sold'].median(),
        'Standart Deviation': df['Sold'].std(),
        'Range': df['Sold'].max() - df['Sold'].min(),
    })
```


2. Dockerfile – набор шагов для создания образа приложения.

Наш Dockerfile

```
FROM nickgryg/alpine-pandas  
  
COPY . .  
  
CMD ["python3", "src/__init__.py"]
```

Dockerfile родительского образа

Branch: master ▾ **alpine-pandas / Dockerfile**

 **nickgryg** Bump python version to 3.8.1 (#4)

1 contributor

7 lines (4 sloc) | 129 Bytes

```
1 FROM python:3.8.1-alpine3.11  
2  
3 LABEL maintainer="nickgryg@gmail.com"  
4  
5 RUN apk --update add --no-cache g++  
6  
7 RUN pip install pandas
```

3. Сборка образа

```
~$ docker build -t prosvita-python-app .
```

`docker build` – команда

`-t(tag) prosvita-python-app` – тег или имя образа

`.` – путь к директории с контекстом и там, где лежит Dockerfile

4. Запуск образа

```
~$ docker run prosvita-python-app
```

`docker run` - команда старта

`prosvita-python-app` - имя (тег) образа

5. Результат

```
(venv) anton@dell:~/prosvita/2020-docker/docker-examples/app$ docker build -t prosvita-python-app .
Sending build context to Docker daemon 11.78kB
Step 1/3 : FROM nickgryg/alpine-pandas
----> 1a68cbda3f04
Step 2/3 : COPY . .
----> 0302784def28
Step 3/3 : CMD ["python3", "src/__init__.py"]
----> Running in e8dcc108a96a
Removing intermediate container e8dcc108a96a
----> cd5808d0c81d
Successfully built cd5808d0c81d
Successfully tagged prosvita-python-app:latest
(venv) anton@dell:~/prosvita/2020-docker/docker-examples/app$ docker run prosvita-python-app
{'Name': 'product_sales_2.csv', 'Count': 100, 'Min': 511, 'Max': 1522, 'Mean': 1028.28, 'Median': 1053.5, 'Standart Deviation': 295.7607578070253, 'Range': 1011}
{'Name': 'product_sales_1.csv', 'Count': 100, 'Min': 32, 'Max': 119, 'Mean': 72.35, 'Median': 71.0, 'Standart Deviation': 24.097183288182094, 'Range': 87}
{'Name': 'product_sales.csv', 'Count': 100, 'Min': 213, 'Max': 506, 'Mean': 362.63, 'Median': 364.0, 'Standart Deviation': 86.48788728177286, 'Range': 293}
(venv) anton@dell:~/prosvita/2020-docker/docker-examples/app$
```

Работа с запущенным контейнером

Прослушка

`logs --tail N container_name`

```
(venv) anton@dell:~/prosvita$ docker run -d --name=test-mysql --env="MYSQL_ROOT_PASSWORD=root" mysql:5.7
3722570de8d2179b1a2124a9d3aa1399c2ed9be8909cc885d9a3c34f29f516c2
(venv) anton@dell:~/prosvita$ docker logs --tail 5 test-mysql
2020-01-21T18:30:59.334576Z 0 [Note] Server socket created on IP: '::'.
2020-01-21T18:30:59.337755Z 0 [Warning] Insecure configuration for --pid-file: Location '/var/run/mysqld' :
ll OS users. Consider choosing a different directory.
2020-01-21T18:30:59.370662Z 0 [Note] Event Scheduler: Loaded 0 events
2020-01-21T18:30:59.371269Z 0 [Note] mysqld: ready for connections.
Version: '5.7.29' socket: '/var/run/mysqld/mysqld.sock' port: 3306 MySQL Community Server (GPL)
(venv) anton@dell:~/prosvita$
```

Внедрение exec

```
(venv) anton@dell:~/prosvita$ docker exec -it test-mysql mysql -u root -proot
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.29 MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database          |
+-----+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+-----+
4 rows in set (0.00 sec)

mysql> █
```

Volume

Volume – дисковое пространство между OS и контейнером

Volume позволяет пробросить файл или папку из файловой системы OS к контейнеру.

Примеры:

В случае удаления контейнера с БД, его volume сохранится, и при следующем запуске БД не будет пустой.

Веб сервер nginx требует при старте конфиги. С помощью volume их можно менять из OS

Network

Network – Внутренняя сеть для нескольких контейнеров. Работает как обычная закрытая сеть.

Пример:

База данных и сервер. Сервер для доступа использует не адресс БД на локальном хосте, а имя в докер сети

localhost:9042 =| cassandra:9042

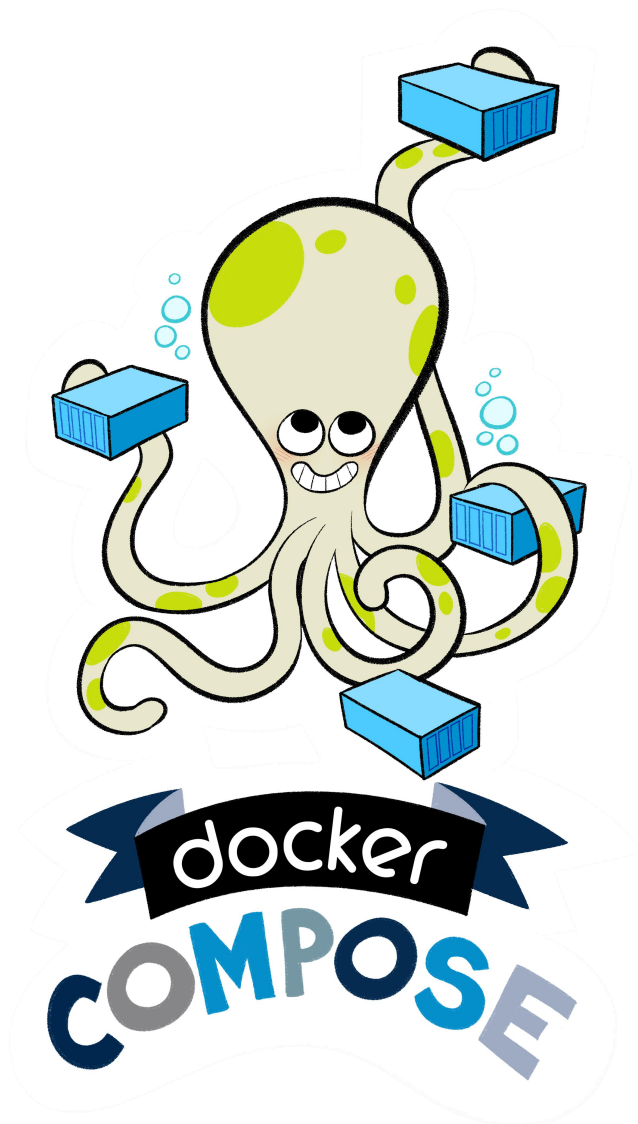
Дисковое пространство

```
(venv) anton@dell:~/prosvita/2020-docker/docker-compose-example$ docker volume ls
DRIVER          VOLUME NAME
local           25837a909b754a063e8bc7506dcb0105fe3658e62360d8ccf41ecc8c9254b1aa
local           prosvita_mongo
(venv) anton@dell:~/prosvita/2020-docker/docker-compose-example$
```

Сети

```
(venv) anton@dell:~/prosvita/2020-docker/docker-compose-example$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
ea97dfe0447a        bridge             bridge              local
dc8b3da48da0        host               host                local
0d4db2ac0c17        none               null                local
fc31897873d6        prosvita_default   bridge              local
```


Docker-compose



Управления группой контейнеров
– задача помощника капитана

Docker-compose – пакетный
менеджер контейнеров,
позволяющий описывать
необходимую структуру в одном
файле (конфиге).

Пример docker-compose для готовых к использованию контейнеров фронтенд, бекенд приложений и БД MongoDB

```
version: '3.4'
services:
  mongo:
    image: mongo:4.2
    restart: always
    environment:
      MONGO_INITDB_ROOT_USERNAME: root
      MONGO_INITDB_ROOT_PASSWORD: testpass
    ports:
      - '27019:27017'
    volumes:
      - mongo:/data/db
  back:
    image: prosvita-back:latest
    restart: always
    ports:
      - '3000:3000'
    environment:
      - NODE_ENV=production
  front:
    image: prosvita-front:latest
    ports:
      - '3010:80'
volumes:
  mongo:
```

Запуск

`docker-compose -f ./docker-compose.yaml -p prosvita up -d`

```
(venv) anton@dell:~/prosvita/2020-docker/docker-compose-example$ docker-compose -f ./docker-compose.yaml -p prosvita up -d
Creating prosvita_front_1 ...
Creating prosvita_mongo_1 ...
Creating prosvita_back_1 ...
Creating prosvita_front_1
Creating prosvita_mongo_1
Creating prosvita_front_1 ... done
```

Контейнеры

docker ps

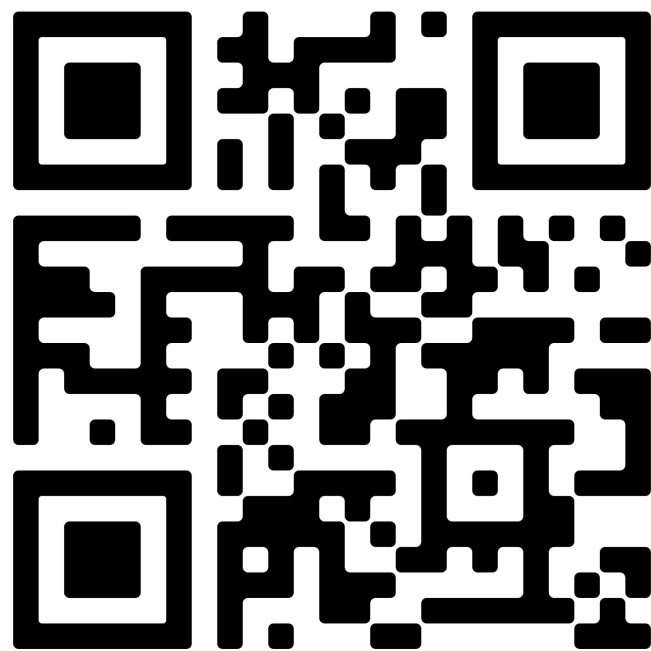
```
(venv) anton@dell:~/prosvita/2020-docker/docker-compose-example$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
6dd96bdee38f	prosvita-back:latest	"docker-entrypoint.s..."	About a minute ago	Up About a minute	0.0.0.0:3000->3000/tcp	prosvita_back_1
7fe0b5bf01ed	mongo:4.2	"docker-entrypoint.s..."	About a minute ago	Up About a minute	0.0.0.0:27019->27017/tcp	prosvita_mongo_1
64f27a106dcb	prosvita-front:latest	"nginx -g 'daemon of..."	About a minute ago	Up About a minute	0.0.0.0:3010->80/tcp	prosvita_front_1

GUI

CLICK
for
extended docs





SCAN ME

Acropolium*

