

Docker lecture #2

Создание и запуск контейнеров.
Проброс портов.

Andrey Maslennikov
2023



План лекций

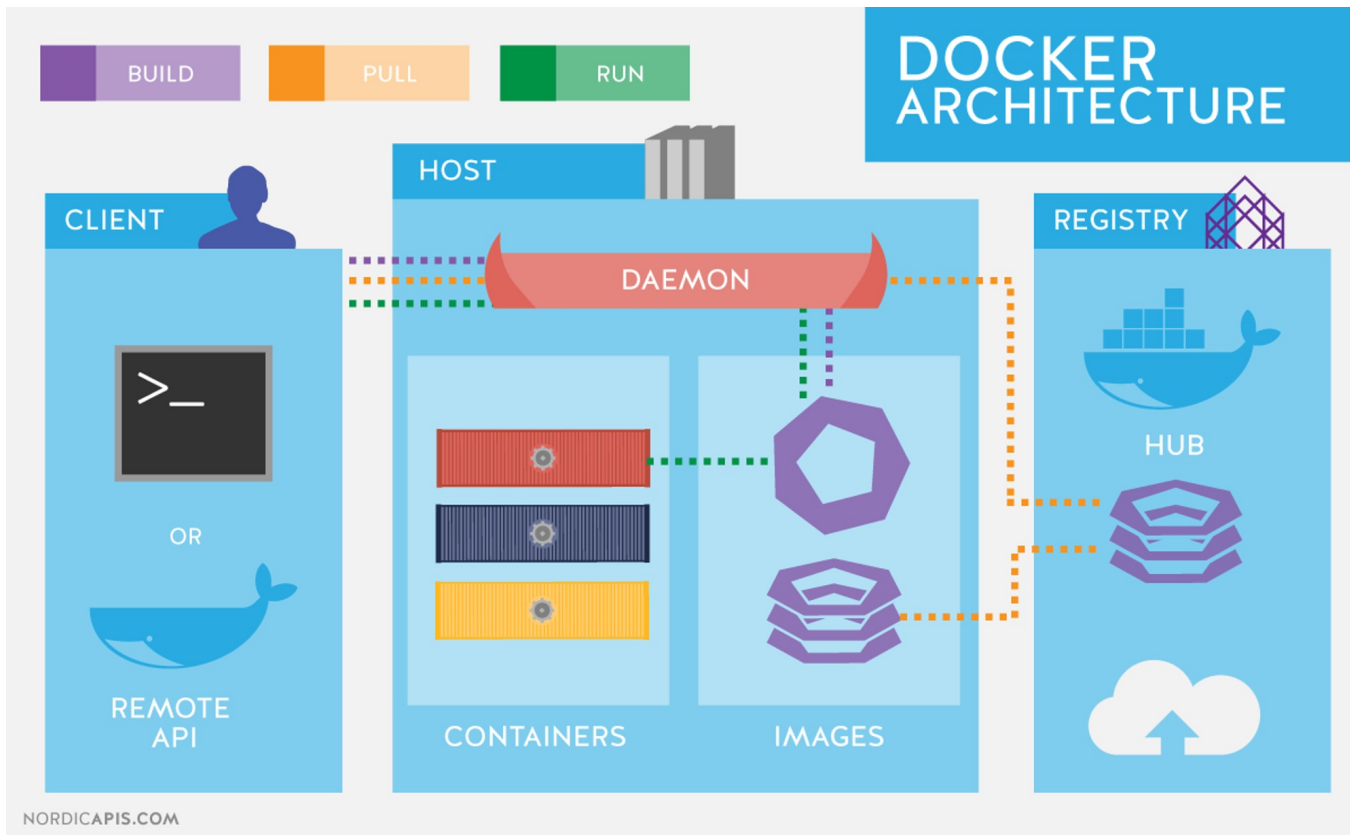
Docker

1. Docker intro. How docker works?
2. What do we have: containers, images, volumes, networks, registries.
3. Run simple images/ using CLI and extensions
4. Using docker for easy database access.
5. Volumes and connecting them - run with nginx
6. Write simple Dockerfile for python backend. Port binding.
7. Write Dockerfile for frontend (build step and run container).
8. Write docker-compose to start whole app (talking about network inside docker).

1


REPETITION

Docker architecture



Docker Hub



https://hub.docker.com/

 docker hub


redis

Explore Pricing Sign In Register

Explore Official Images redis

 **redis**  DOCKER OFFICIAL IMAGE · 1B+ · 10K+

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

docker pull redis 

Overview Tags

Quick reference

- Maintained by:
the Docker Community
- Where to get help:
the Docker Community Slack, Server Fault, Unix & Linux, or Stack Overflow

Supported tags and respective Dockerfile links

- 7.2-rc1, 7.2-rc, 7.2-rc1-bullseye, 7.2-rc-bullseye
- 7.2-rc1-alpine, 7.2-rc-alpine, 7.2-rc1-alpine3.17, 7.2-rc-alpine3.17

Recent Tags

latest bullseye 7.2-rc1-bullseye 7.2-rc1

7.2-rc-bullseye 7.2-rc 7.0.10-bullseye 7.0.10

7.0-bullseye 7.0

About Official Images

Docker Official Images are a curated set of Docker open source and drop-in solution repositories.

Why Official Images?

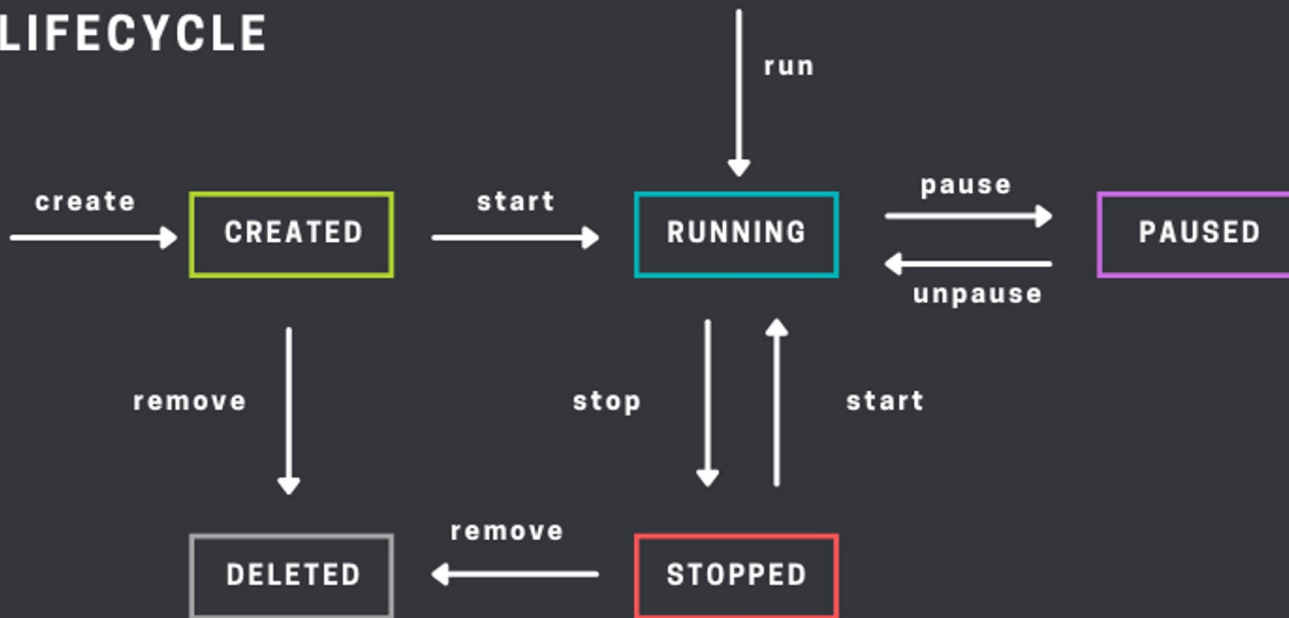
These images have clear documentation, promote



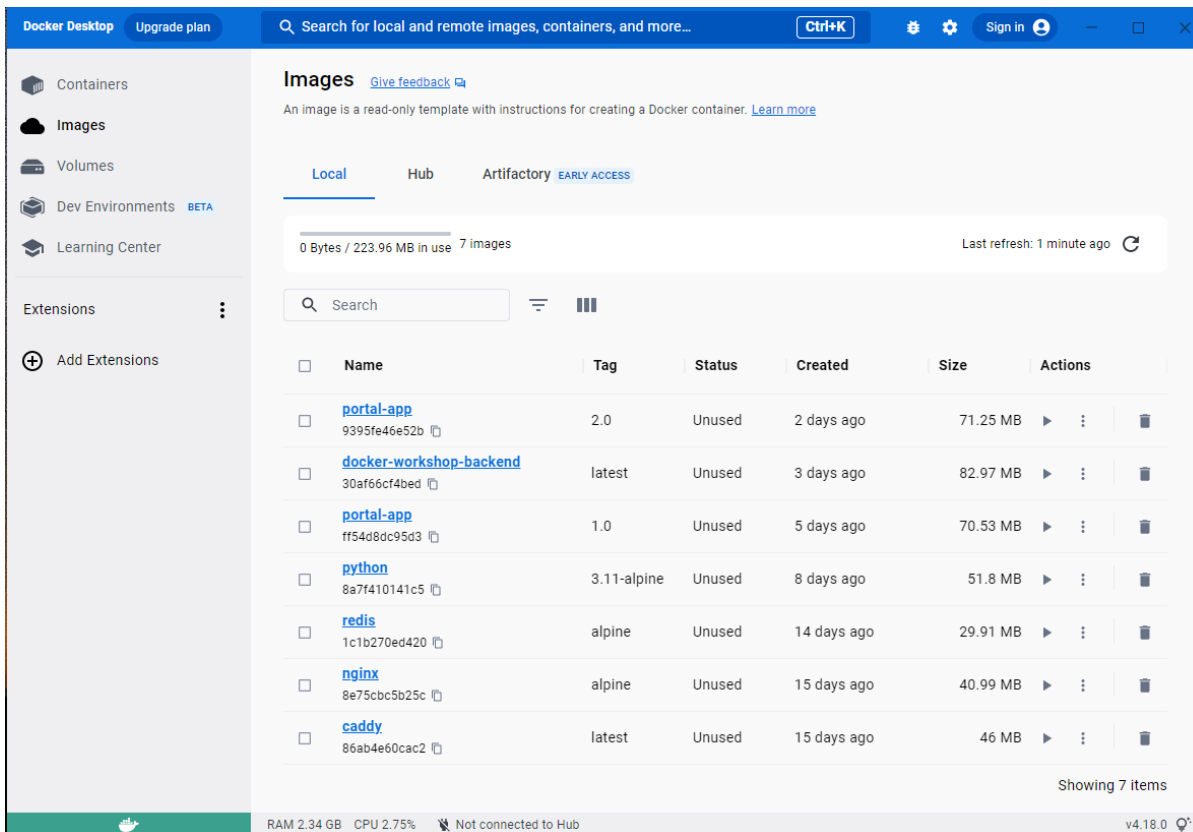
Жизненный цикл контейнера

DOCKER CONTAINER LIFECYCLE

LINUX
HANDBOOK



Docker Desktop



Docker Desktop Upgrade plan Search for local and remote images, containers, and more... Ctrl+K Sign in

Containers
Images
Volumes
Dev Environments BETA
Learning Center

Extensions
Add Extensions

Images

An image is a read-only template with instructions for creating a Docker container. [Learn more](#)

Local Hub Artifactory EARLY ACCESS

0 Bytes / 223.96 MB in use 7 images Last refresh: 1 minute ago

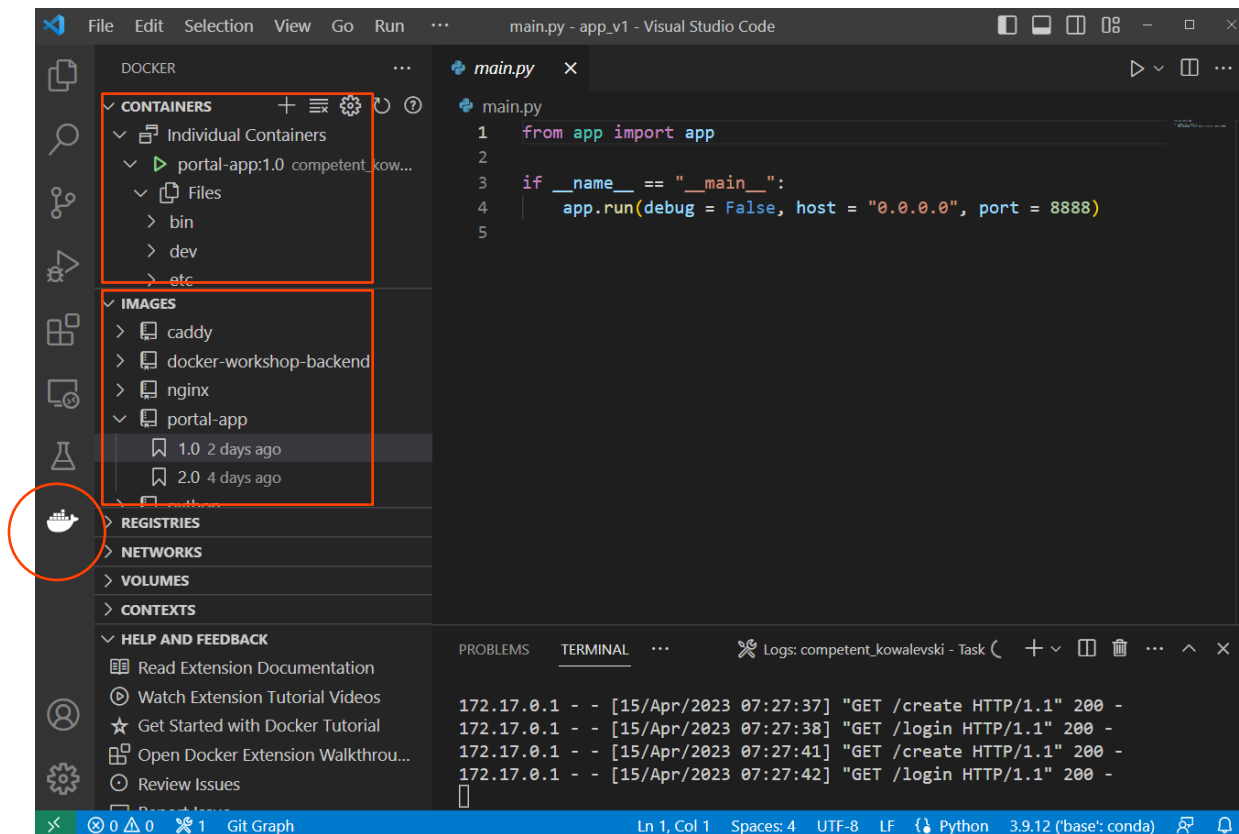
Search

<input type="checkbox"/>	Name	Tag	Status	Created	Size	Actions
<input type="checkbox"/>	portal-app 9395fe46e52b	2.0	Unused	2 days ago	71.25 MB	
<input type="checkbox"/>	docker-workshop-backend 30af66cf4bed	latest	Unused	3 days ago	82.97 MB	
<input type="checkbox"/>	portal-app ff54d8dc95d3	1.0	Unused	5 days ago	70.53 MB	
<input type="checkbox"/>	python 8a7f410141c5	3.11-alpine	Unused	8 days ago	51.8 MB	
<input type="checkbox"/>	redis 1c1b270ed420	alpine	Unused	14 days ago	29.91 MB	
<input type="checkbox"/>	nginx 8e75cbc5b25c	alpine	Unused	15 days ago	40.99 MB	
<input type="checkbox"/>	caddy 86ab4e60cac2	latest	Unused	15 days ago	46 MB	

Showing 7 items

RAM 2.34 GB CPU 2.75% Not connected to Hub v4.18.0

VSCode Docker extension

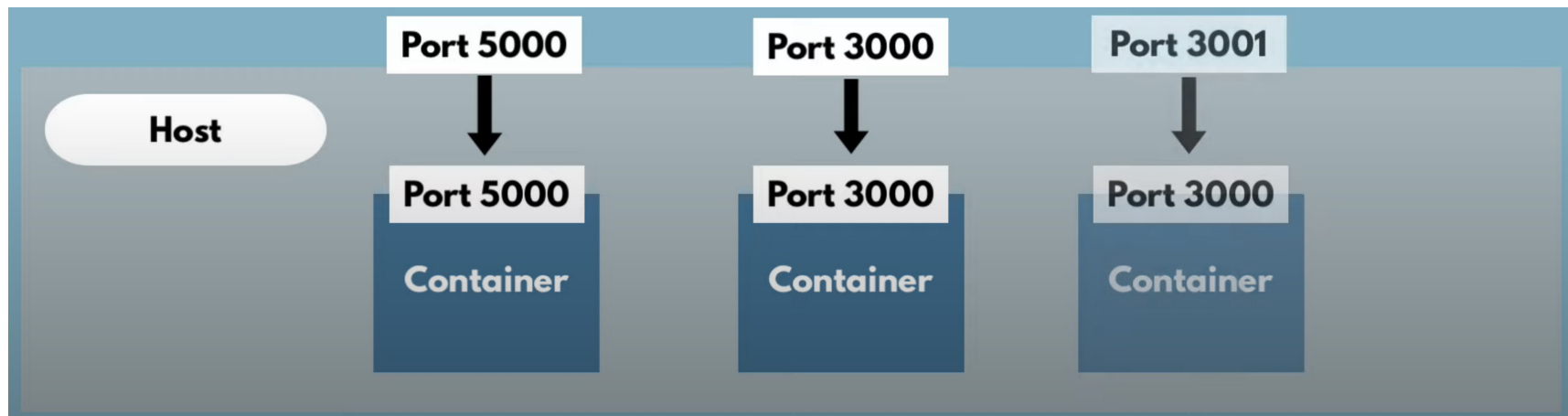


2 QUESTIONS

3

MAIN TOPIC

Port mapping



Run container



```
docker run -d -p3000:6379 redis:alpine      # host port 3000
docker run -d -p3001:6379 redis:6.2        # host port 3001

docker run -d -p3001:6379 --name old_redis redis:6.2
```



Connect to container



```
docker run -d \  
-p3000:6379 \  
--name redis-db \  
redis:alpine redis-server --save 60 1
```

```
docker exec -it <container id> /bin/sh           # interactive terminal  
> exit
```

```
docker exec -it redis-db redis-cli
```

```
> keys *  
> get user:Alice  
> set user:Bob  
"{\"name\": \"Bob\", \"password\": \"111\", \"date\": 738625}"  
> exit
```

```
docker exec -it redis-db sh  
> ls /data
```



Port mapping

LAB1

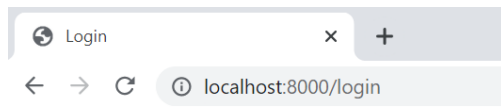
Web application

```
$ ./app.py 3000
```



PYTHON

<http://localhost:8000>



Login | [Create](#) | [List](#)

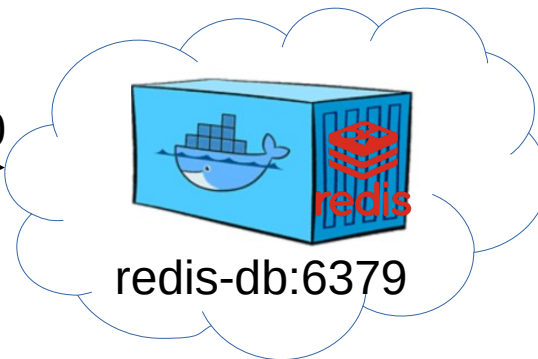
Username:

Password:

Login

Docker isolated network

:3000



Практическое задание LAB1



TEL-RAN
by Starta Institute

1. Создайте и запустите два контейнера с Redis DB. Первый контейнер на порту 3000, второй на порту 3001.
2. Запустите веб-приложение и запишите данные в базу данных в первом контейнере используя порт 3000.
3. Остановите первый контейнер.
4. Перезапустите веб-приложение и запишите данные в базу данных используя порт 3001.
5. Запустите снова первый контейнер и переключите веб-приложение на порт 3000.
6. Прочитайте данные из БД в веб-приложении.

Практическое задание LAB1



TEL-RAN
by Starta Institute

Docker containers

```
docker run -d -p3000:6379 redis:alpine  
docker run -d -p3001:6379 redis:alpine
```

Приложение:

```
git clone https://github.com/2a-stra/docker-demo.git  
cd docker-demo/lab1
```

Запуск приложения:

```
python app.py 3000  
(python3 app.py 3000)
```

Проверка веб-приложения из командной строки:

```
curl http://localhost:8000/list  
curl -X POST -d 'user=Alice&password=123' http://localhost:8000/create  
curl -X POST -d 'user=Alice' http://localhost:8000/delete
```

Зависимости

Установка Python:

<https://www.python.org/downloads/>

Установка зависимостей:

```
pip install redis flask
```


Dockerfile



FROM <image>

- установить базовый образ

ENV <NAME>=<value>

- передать переменную окружения внутрь контейнера

RUN <command>

- выполнить команду Linux внутри контейнера
(может быть несколько команд)

COPY <local> <container_path>

- копирование с host машины

CMD []

- запустить приложение
(точка входа, однократно)



Пример

LAB1



TEL-RAN
by Starta Institute

```
FROM python:3.11-alpine
```

```
WORKDIR /home/app
```

```
COPY . .
```

```
RUN pip3 install -r requirements.txt
```

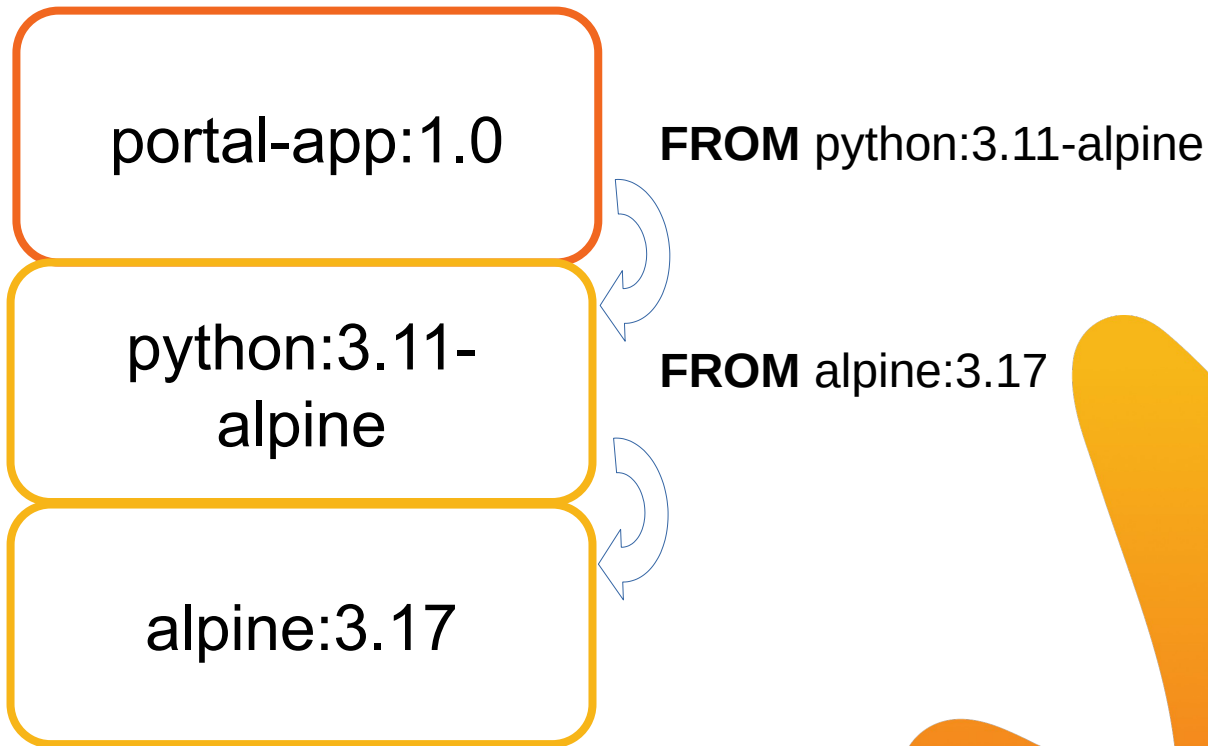
```
EXPOSE 8000
```

```
CMD ["python", "app.py", "3000"]
```

(.dockerignore)



Создание образа



Build



```
$ docker build -t portal-app:0.1 .
```

History

```
$ docker history portal-app:1.0
```

```
$ docker history python:3.11-alpine
```

```
$ docker history alpine
```



Практическое задание LAB1



TEL-RAN
by Starta Institute

1. Создайте новый образ с веб-приложением:

```
$ docker build -t portal-app:0.1 .
```

2. Запустите контейнер «portal-app»:

```
$ docker run -d --rm -p 8000:8000 portal-app:0.1
```

3. Подключитесь через интерактивный терминал к запущенному контейнеру:

```
$ docker exec -it <portal-app> sh
```

4. Проверьте содержимое рабочего каталога.

5. Сделайте запрос к БД через веб-интерфейс: <http://localhost:8000/>

6. Проверьте ошибки в логе сообщений.

Тест

Почему мы видим сообщение ошибке?

redis.exceptions.ConnectionError: Error 99 connecting to localhost:3000. Address not available.



Answer

Обращение к БД Redis происходит внутри контейнера portal-app:

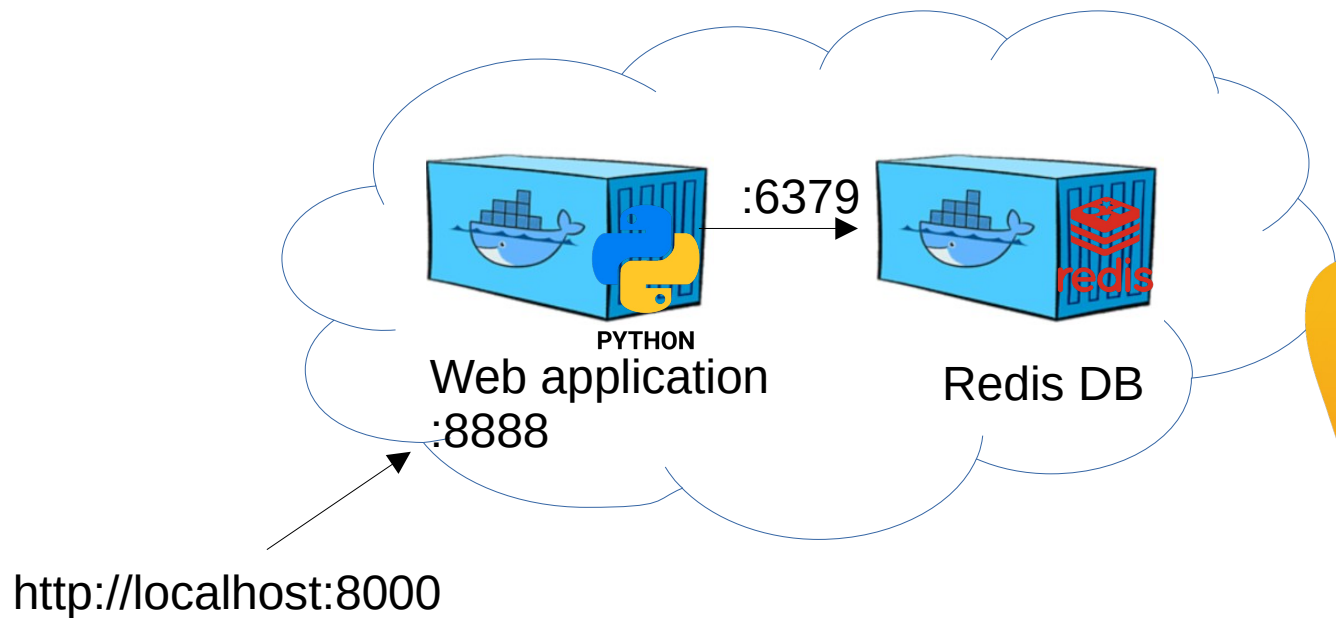
`localhost:3000`

Но БД находится в другом контейнере.

Port mapping

LAB2


Docker isolated network



3

MAIN TOPIC QUESTIONS

Спасибо!

 Форма обратной связи

