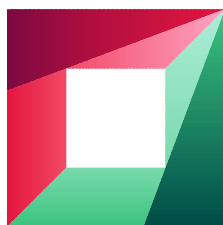




МИНИСТЕРСТВО НАУКИ
И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

НГТУ

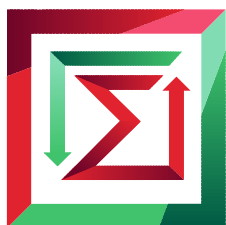


НЭТИ

Кафедра теоретической и прикладной информатики

Лабораторная работа № 2
по дисциплине «Разработка web-приложений и распределённых ИС»

Место для ввода текста.



ФПМИ

Студент

ПМ-84 СКВОРЦОВ ИВАН

Преподаватель

ЦЫГУЛИН АЛЕКСЕЙ АЛЕКСАНДРОВИЧ

Новосибирск

Задача. Создать контейнер, реализующий одну из функций сервиса умного дома.

Вариант 2. BotState. Контейнер хранящий состояние всех ботов (например на базе MongoDB).

1. Dockerfile app

FROM node:12

WORKDIR /app

COPY package*.json ./

RUN npm install

COPY . .

EXPOSE 3000

CMD ["npm", "start"]

2. Dockerfile db

FROM mysql:5.7

EXPOSE 3307

COPY mys.sql /docker-entrypoint-initdb.d/

3. docker-compose.yaml

version: '3'

services:

app:

image: api/state

build: ./app

ports:

- '3000:3000'

volumes:

- ./app:/app

- /app/node_modules

depends_on:

- db

db:

image: api/state/database

build: ./db

environment:

MYSQL_USER: 'root'

MYSQL_DATABASE: 'root'

MYSQL_PASSWORD: '123456'

MYSQL_ROOT_PASSWORD: '123456'

MYSQL_ALLOW_EMPTY_PASSWORD: '123456'

MYSQL_RANDOM_ROOT_PASSWORD: '123456'

restart: always

ports:

- '3307:3307'

volumes:

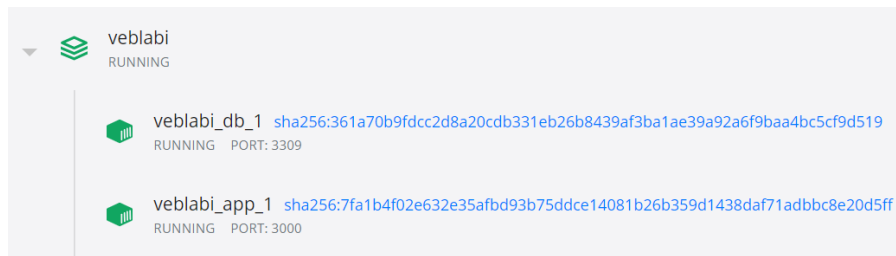
db-config:

db-data:

4. Докер-построение (docker compose up --build).

```
.\Users\user\Desktop\veblabi>docker compose up --build
[+] Building 36.6s (16/16) FINISHED
=> [api/state internal] load build definition from Dockerfile
=> => transferring dockerfile: 31B
=> [api/state/database internal] load build definition from Dockerfile
=> => transferring dockerfile: 31B
=> [api/state internal] load .dockerignore
=> => transferring context: 2B
=> [api/state/database internal] load .dockerignore
=> => transferring context: 2B
=> [api/state internal] load metadata for docker.io/library/node:12
=> [api/state/database internal] load metadata for docker.io/library/mysql:5.7
=> [api/state/database internal] load build context
=> => transferring context: 256B
=> [api/state/database 1/2] FROM docker.io/library/mysql:5.7@sha256:a682e3c78fc5bd941e9db088b4796c75f69a28a8cad65677c23f7a9f18ba21fa
=> [api/state 1/5] FROM docker.io/library/node:12@sha256:e38a48c584fde720a43b1bacfe1178e0900936cb8d0141630e5e35693ee6d08
=> [api/state internal] load build context
=> => transferring context: 13.35MB
=> CACHED [api/state/database 2/2] COPY mys.sql /docker-entrypoint-initdb.d/
=> [api/state] exporting to image
=> => exporting layers
=> => writing image sha256:361a70b9fdcc2d8a20cdb331eb26b8439af3ba1ae39a92a6f9baa4bc5cf9d519
=> => naming to docker.io/api/state/database
=> => writing image sha256:7fa1b4f02e632e35afbd93b75ddce14081b26b359d1438daf71adbbc8e20d5ff
=> => naming to docker.io/api/state
=> CACHED [api/state 2/5] WORKDIR /app
=> CACHED [api/state 3/5] COPY package*.json ./
=> CACHED [api/state 4/5] RUN npm install
=> CACHED [api/state 5/5] COPY . .
```

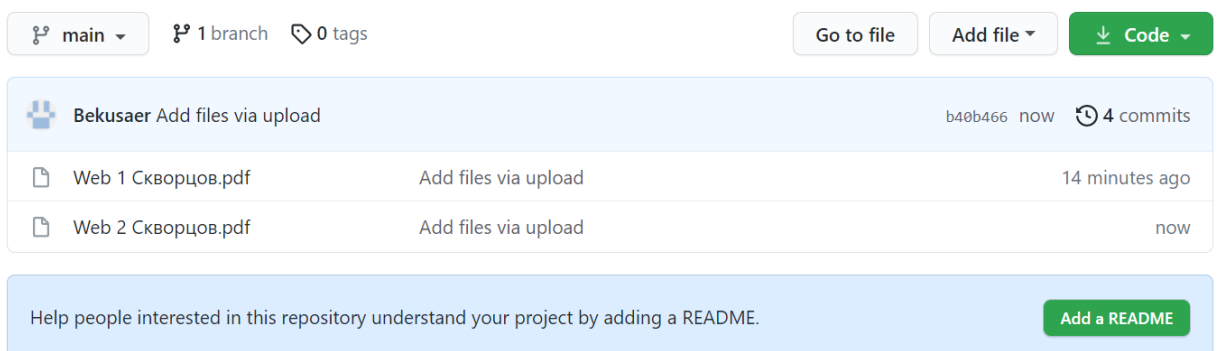
5. Образы в Docker Desktop.



The screenshot shows the Docker Desktop interface. At the top, there's a section for 'veblabi' which is 'RUNNING'. Below this, two containers are listed:

- veblabi_db_1**: sha256:361a70b9fdcc2d8a20cdb331eb26b8439af3ba1ae39a92a6f9baa4bc5cf9d519, RUNNING, PORT: 3309
- veblabi_app_1**: sha256:7fa1b4f02e632e35afbd93b75ddce14081b26b359d1438daf71adbbc8e20d5ff, RUNNING, PORT: 3000

6. Загрузка на github.



The screenshot shows a GitHub repository page for 'Bekusaer'. At the top, there's a navigation bar with 'main' branch selected, '1 branch', and '0 tags'. Below this, there's a section for 'Add files via upload' with a button 'Add files via upload' and a 'Code' button. Below this, there's a table of recent commits:

Commit Hash	Author	Time	Commits
b40b466	now	14 minutes ago	4 commits
	now		

At the bottom, there's a section for 'Help people interested in this repository understand your project by adding a README.' with a button 'Add a README'.

Ссылка: https://github.com/Bekusaer/discipline_web