|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ  Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |

Институт комплексной безопасности и специального приборостроения

Кафедра КБ-14 «Интеллектуальные системы информационной безопасности»

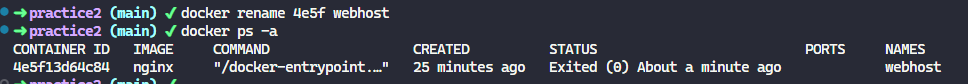
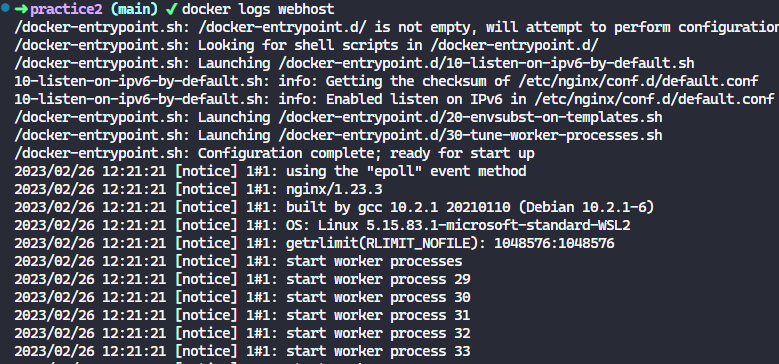
### Администрирование баз данных

### Практическая работа № 2

# ОТЧЁТ

# Выполнил студент группы БСБО-07-20 Любовский С.В.

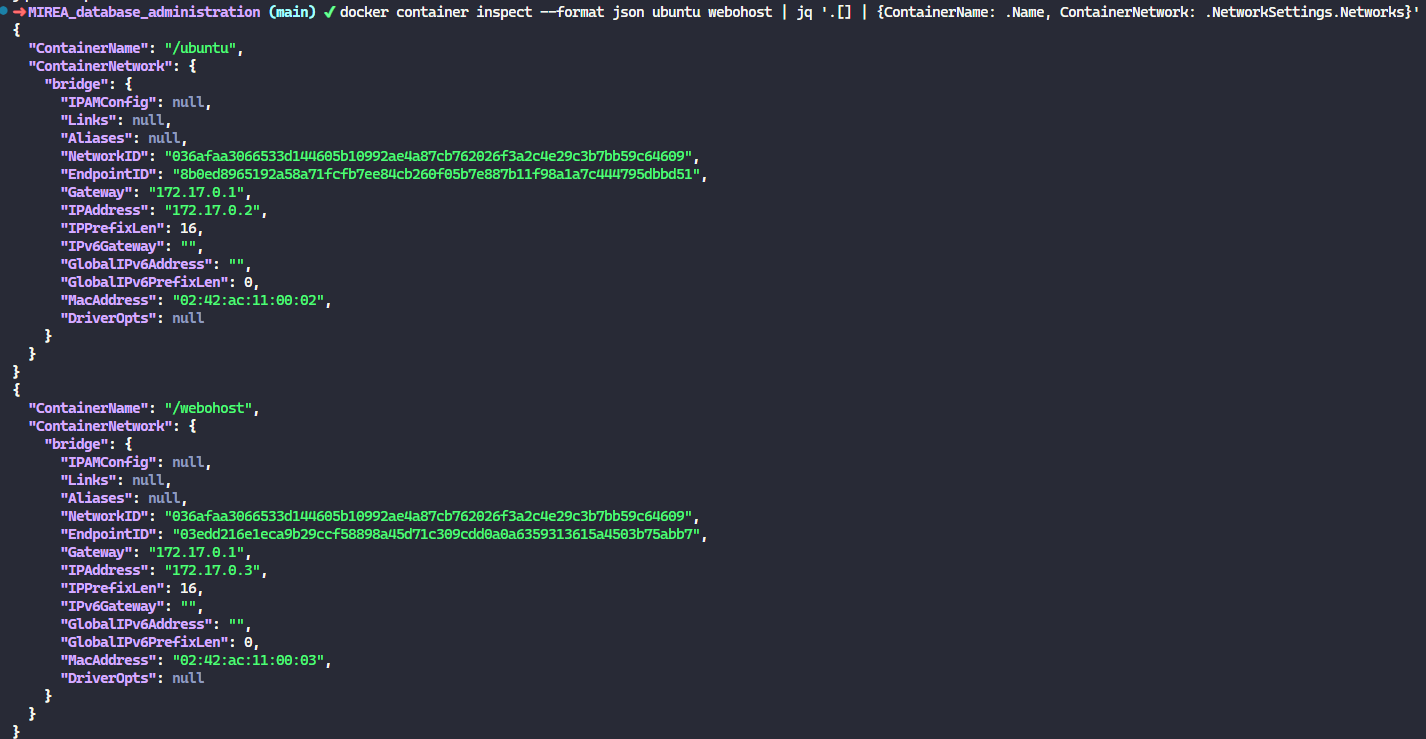
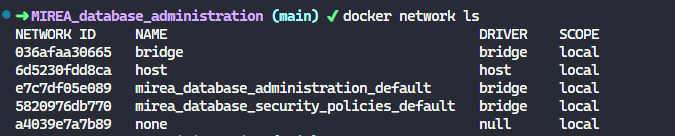
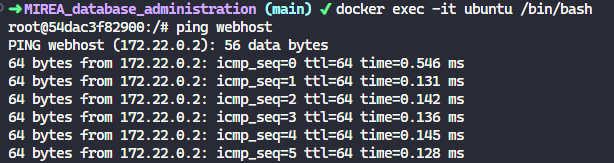
# Выполнение задания 1.

1. Установим docker на локальной машине
2. Используя Docker Hub загрузим образ nginx командой **docker pull nginx:latest**
3. Запустим контейнер **nginx** в фоновом режиме командой **docker run -d nginx**
4. Выведем список работающих контейнеров командой **docker ps**
5. Остановим контейнер используя команду **docker stop 4e5f**
6. Переименуем контейнер в webhost используя команду **docker rename 4e5f webhost**
7. Запустим контейнер с именем webhost используя команду **docker start webhost**
8. Выведем логи контейнера webhost используя команду **docker logs webhost**

# Выполнение задания 2.

1. Загрузим образ Ubuntu из Docker hub командой **docker pull ubuntu:latest**
2. Запустим контейнер используя команду **docker run --name ubuntu -d ubuntu /bin/bash -c "sleep 1000000"** и подключимся к нему используя команду **docker exec -it ubuntu /bin/bash**
3. Установим пакеты **vim, curl, htop, ping** используя команду **apt update && apt upgrade -y && apt install vim curl htop inetutils-ping -y**
4. Выйдем из контейнера и остановим его командой **docker stop ubuntu**
5. Запустим контейнер заново используя команду **docker start ubuntu** и убедимся, что все пакеты установлены используя команду **docker exec -it ubuntu /bin/bash -c "apt list --installed | grep 'vim\|htop\|curl\|ping'**"

# Выполнение задания 3.

1. Используя утилиту jq создадим запрос на получения информации о сети полученной из команды **docker inspect -f json webhost ubuntu**. Итоговая команда будет выглядеть так **docker container inspect --format json ubuntu webohost | jq '.[] | {ContainerName: .Name, ContainerNetwork: .NetworkSettings.Networks}'**
2. Используя команду **docker network ls** выведем все доступные сети.
3. Используя команду **docker network create -d bridge my-bridge-network** создадим свою сеть типа **bridge**
4. Подключим созданные контейнеры к нашей сети используя команду **docker network connect** **bridge my-bridge-network [ИМЯ КОНТЕЙНЕРА]**
5. При помощи утилиты ping проверим, что контейнеры видят друг друга

# Выполнение задания 4.

1. Создадим Dockerfile по условиям

FROM ubuntu:22.04

RUN *apt* *update* *-y* \

*&&* *apt* *upgrade* *-y* \

*&&* *apt* *install* *nano* *apache2* *-y*

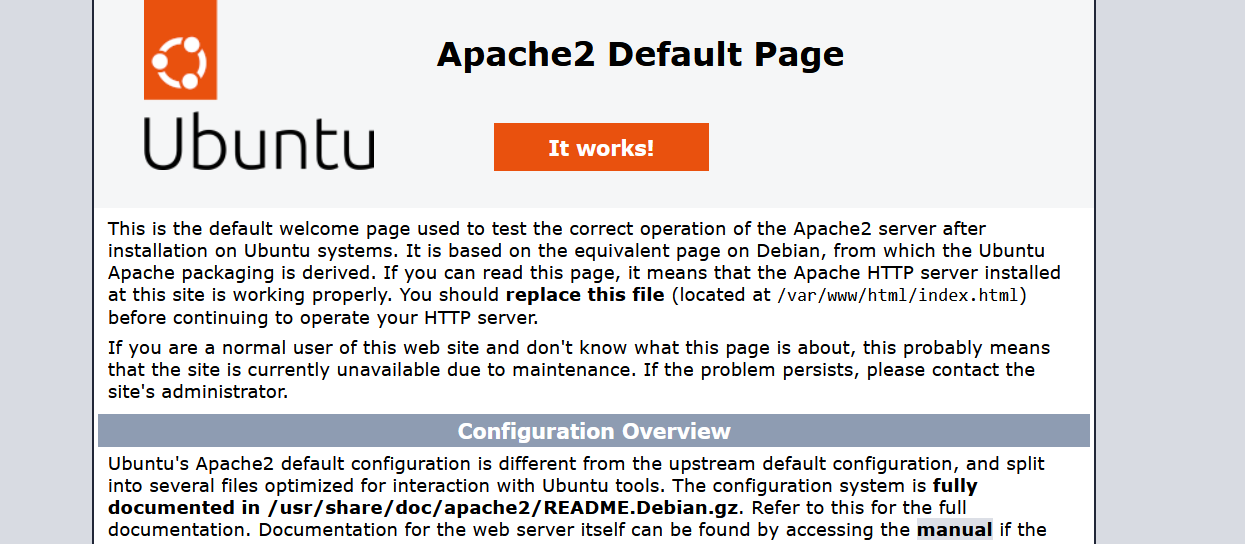
RUN *echo* 'Hello, World!' *>* *index.html*

WORKDIR */usr/local/apache2/htdocs/*

RUN *cp* */index.html* *.*

EXPOSE *80*

CMD *apachectl* *-D* *FOREGROUND*

1. Соберем образ командой **docker build -t ubuntu-apache .**
2. Запустим контейнер на основе собранного образа командой **docker run --name ubuntu-apache -d -p 80:80 ubuntu-apache**
3. Проверим работу
4. Остановим и удалим контейнер командой **docker rm -f ubuntu-apache**
5. Добавим установку пакета curl в Dockerfile

FROM ubuntu:22.04

RUN *apt* *update* *-y* \

*&&* *apt* *upgrade* *-y* \

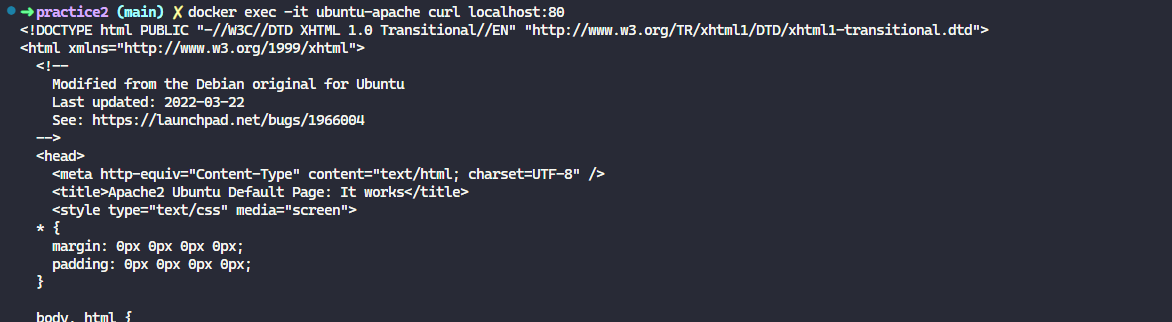
*&&* *apt* *install* *nano* *apache2* *curl* *-y*

RUN *echo* 'Hello, World!' *>* *index.html*

WORKDIR */usr/local/apache2/htdocs/*

RUN *cp* */index.html* *.*

CMD */usr/sbin/apache2ctl* *-D* *FOREGROUND*

1. Пересоберем образ и запустим контейнер заново
2. Проверим что curl установлен внутри контейнера