МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

Отчет по практической работе №4

по дисциплине «Технологии разработки программных приложений»

Тема практической работы: «Docker»

Выполнил:
Студент группы **ИКБО-62-23** Кокшарова А.А.

Проверил преподаватель: Золотухин С.А.

Москва 2025

Оглавление

Постановка задачи 3

Реализация разделов 4

Образы 4

Изоляция 4

Работа с портами 5

Именованные контейнеры, остановка и удаление 7

Постоянное хранение данных 7

Тома 8

Монтирование директорий и файлов 9

Переменные окружения 9

Dockerfile 9

Индивидуальный вариант 11

Вывод 13

Постановка задачи

В отчёт должны быть включены ответы на вопросы, выделенные курсивом, результаты выполнения команд из разделов 1–7, а также выполненное индивидуальное задание (раздел 8): листинг Dockerfile, а также команды сборки и запуска контейнера.

1. Образы
2. Изоляция
3. Работа с портами
4. Именованные контейнеры, остановка и удаление
5. Постоянное хранение данных
6. Переменные окружения
7. Dockerfile
8. Индивидуальное задание

Реализация разделов

Образы

На рисунке 1 изображена загрузка образа, а также просмотр имеющихся образов и списка контейнеров

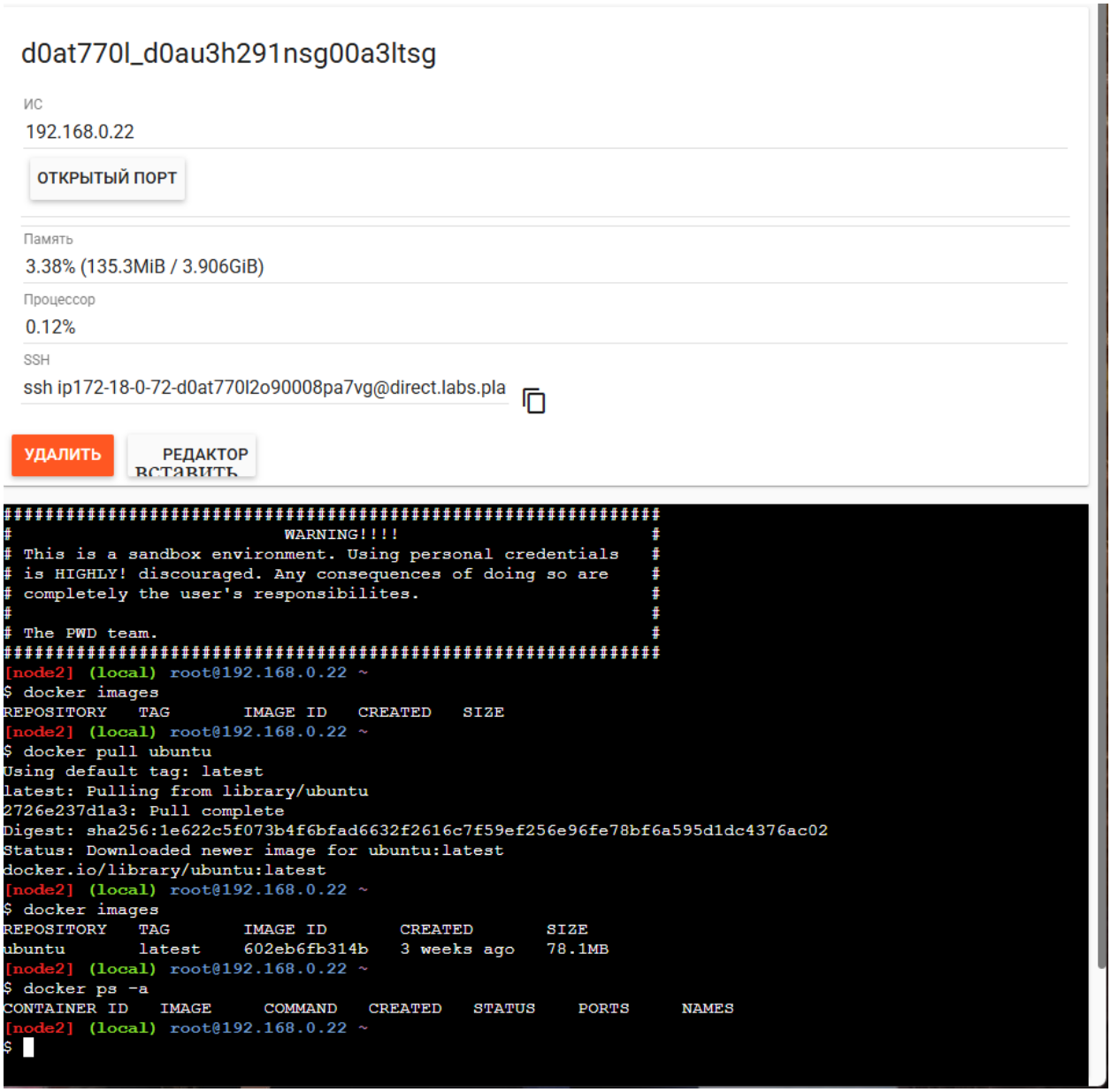


Рисунок 1 – Загрузка образа

Изоляция

На рисунке 2 результат просмотра информации о хостовой системе и запуск контейнеров

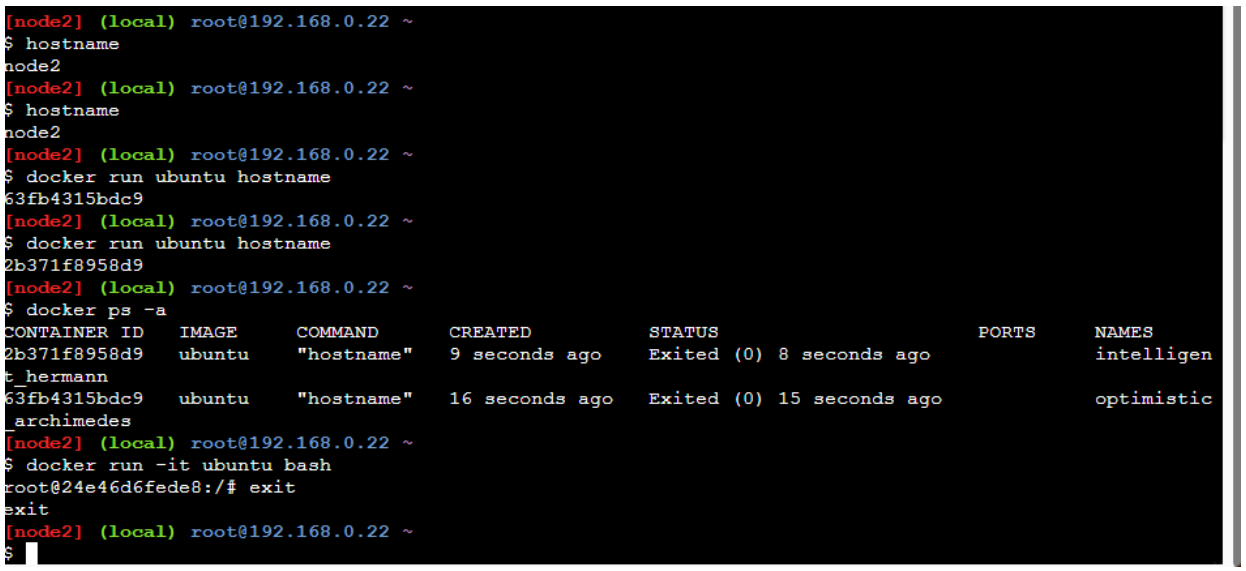


Рисунок 2 – Запуск bash в контейнере

Работа с портами

На рисунках 3–6 изображена загрузка образа python, проброс портов, запуск сервера

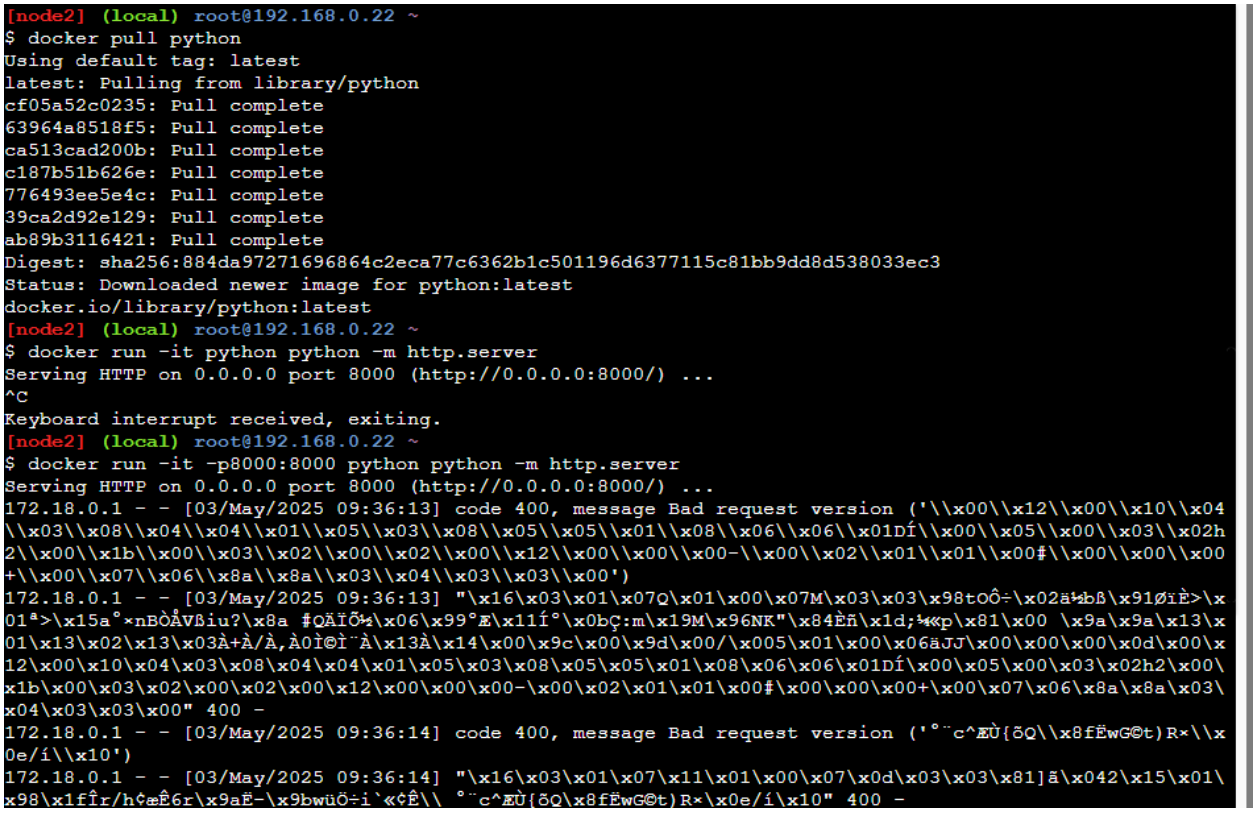


Рисунок 3 – загрузка образа python и запуск сервера с портом 8000

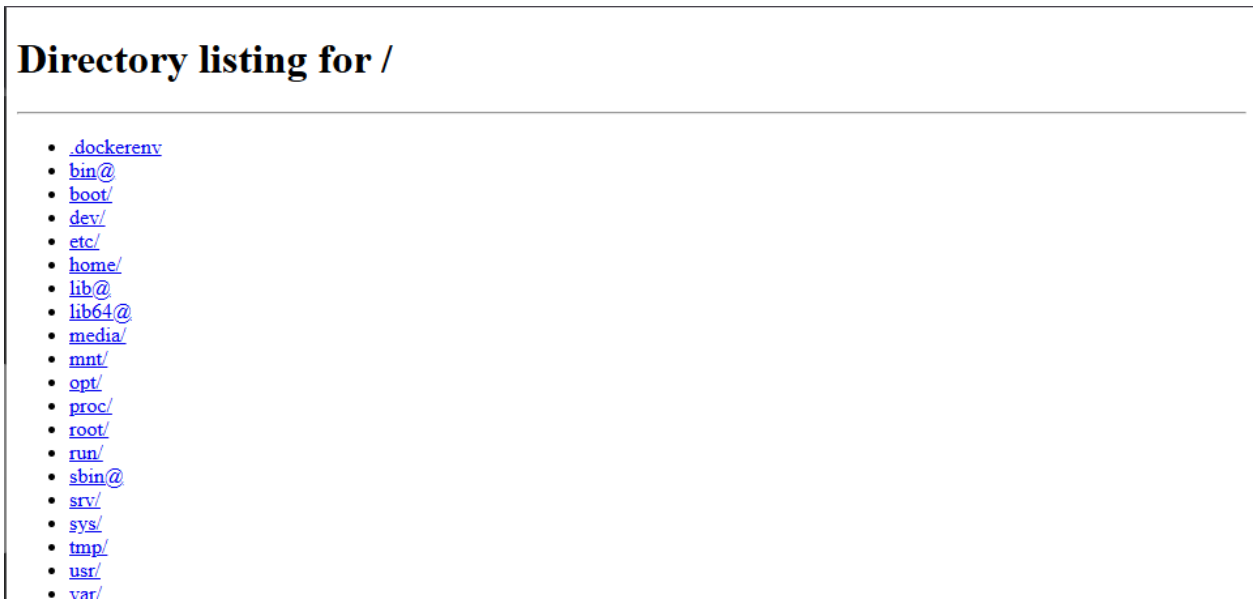


Рисунок 4 - запуск сервера с портом 8000

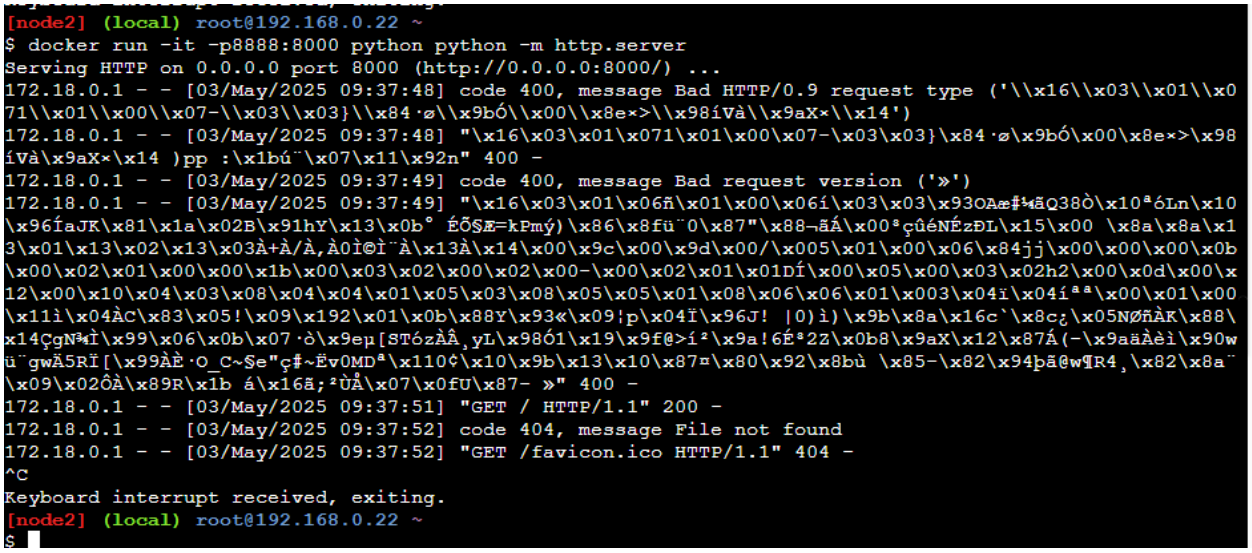
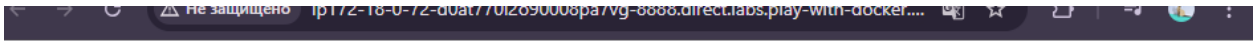


Рисунок 5 - запуск сервера с портом 8888



Directory listing for /

- .dockerenv
- bin@
- boot/
- dev/
- etc/
- home/
- lib@
- lib64@
- media/
- mnt/
- opt/
- proc/
- root/
- run/
- sbin@
- srv/
- sys/
- tmp/
- usr/
- var/

Рисунок 6 - запуск сервера с портом 8888

Именованные контейнеры, остановка и удаление

На рисунке 7 изображен результат выполнения команд. Запущен контейнер в фоновом режиме, проверка на работоспособность, просмотр логов, остановка контейнера, повторный запуск, удаление контейнера

```
[node2] (local) root@192.168.0.22 ~
$ docker run -p8000:8000 --name pyserver -d python python -m http.server
8d6bd16ca7dcbbe5c21117fcc6421075d4fdeb5675e6b3dc81e7ac5fc8d2c63a
[node2] (local) root@192.168.0.22 ~
$ docker ps | grep pyserver
8d6bd16ca7dc   python    "python -m http.serv…"   21 seconds ago   Up 21 seconds   0.0.0.0:8000->8000/t
cp   pyserver
[node2] (local) root@192.168.0.22 ~
$ docker logs pyserver
[node2] (local) root@192.168.0.22 ~
$ docker stop pyserver
pyserver
[node2] (local) root@192.168.0.22 ~
$ docker run -p8000:8000 --name pyserver -d python python -m http.server
docker: Error response from daemon: Conflict. The container name "/pyserver" is already in use by conta
iner "8d6bd16ca7dcbbe5c21117fcc6421075d4fdeb5675e6b3dc81e7ac5fc8d2c63a". You have to remove (or rename)
 that container to be able to reuse that name.
See 'docker run --help'.
[node2] (local) root@192.168.0.22 ~
$ docker rm pyserver
pyserver
[node2] (local) root@192.168.0.22 ~
$ docker run --rm -p8000:8000 --name pyserver -d python python -m http.server
8fe7299a5e0441e14233eb4227e18df38960421bc99bed26ddf30ddd8253619d
[node2] (local) root@192.168.0.22 ~
$
```

Рисунок 7 – Остановка и удаление контейнера

Постоянное хранение данных

На рисунках 8–10 изображен запуск контейнера с корневой директорией mnt, вход в контейнер, запись информации и проверка в браузере

```
[node2] (local) root@192.168.0.22 ~
$ docker run -p8000:8000 --rm --name pyserver -d python python -m http.server -d /mnt
963feb3dcd54a9c3f3a0c9f6b88b4bc6949d75c92a1f8b1b5b40393c180c30eb
[node2] (local) root@192.168.0.22 ~
$ docker exec -it pyserver bash
root@963feb3dcd54:/# cd /mnt
root@963feb3dcd54:/mnt# echo "Hello World!">hi.txt
root@963feb3dcd54:/mnt# exit
exit
[node2] (local) root@192.168.0.22 ~
$ docker stop pyserver
pyserver
[node2] (local) root@192.168.0.22 ~
$
```

Рисунок 8 – Запуск контейнера и запись информации

Directory listing for /

- hi.txt

Рисунок 9 – Файл, в который была записана информация

```
Hello World!
```

Рисунок 10 – Информация в файле hi.txt

Тома

На рисунках 11–12 изображено создание тома, создание контейнера с примонтированным томом, запись информации в файл и проверка в браузере

```
[node2] (local) root@192.168.0.22 ~
$ mkdir myfiles
[node2] (local) root@192.168.0.22 ~
$ docker run -p8000:8000 --rm --name pyserver -d -v $(pwd)/myfiles:/mnt python python -m http.server -d
 /mnt
19a76e31ccf7808e0cb83c62fdb4046cba769c334d62a2b4373d9427d44354f8
[node2] (local) root@192.168.0.22 ~
$ docker exec -it pyserver bash
root@19a76e31ccf7:/# cd /mnt
root@19a76e31ccf7:/mnt# echo "Hello World2" > hi.txt
root@19a76e31ccf7:/mnt# exit
exit
[node2] (local) root@192.168.0.22 ~
$ docker inspect -f "{{json .Mounts}}" pyserver
[{"Type":"bind","Source":"/root/myfiles","Destination":"/mnt","Mode":"","RW":true,"Propagation":"rpriva
te"}]
[node2] (local) root@192.168.0.22 ~
$ docker stop pyserver
pyserver
[node2] (local) root@192.168.0.22 ~
$
```

Рисунок 11 – Создание файла и запись информации

```
Hello World2
```

Рисунок 12 – проверка в браузере

Монтирование директорий и файлов

На рисунке 13 изображено создание директории и файла, запуск контейнера, монтирование директории и файла

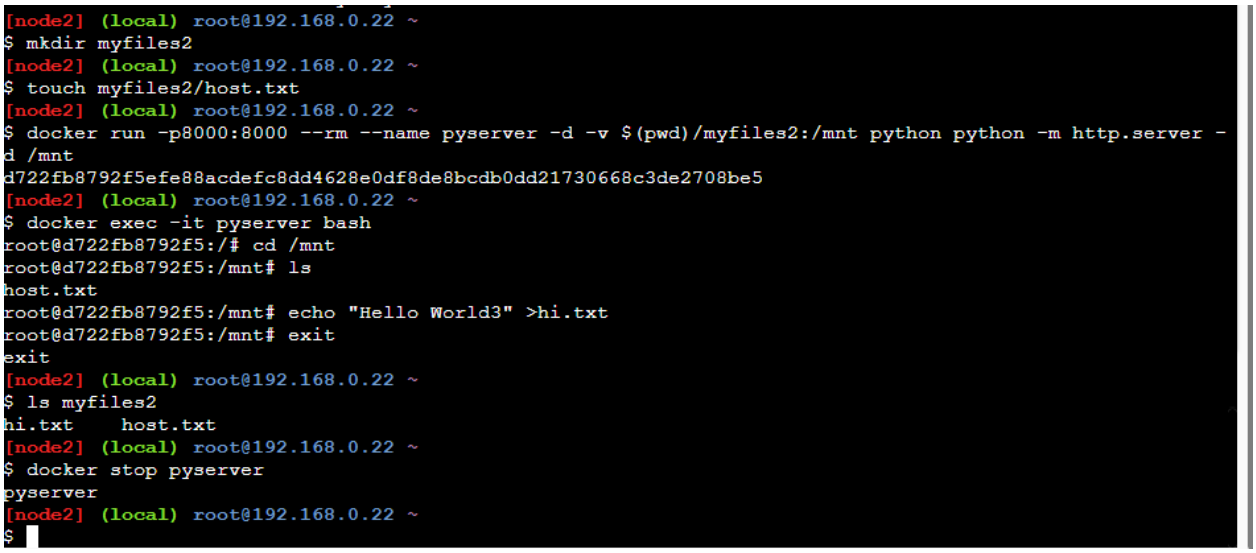


Рисунок 13 – Монтирование директории и файла

Переменные окружения

На рисунке 14 изображена передача в контейнер переменной окружения MIREA со значением “ONE LOVE”

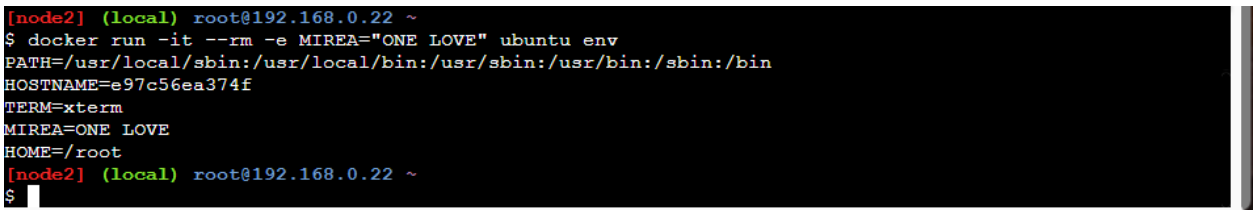


Рисунок 14 – Передача переменной окружения MIREA

Dockerfile

На рисунках 15–20 изображена сборка образа и проверка в браузере

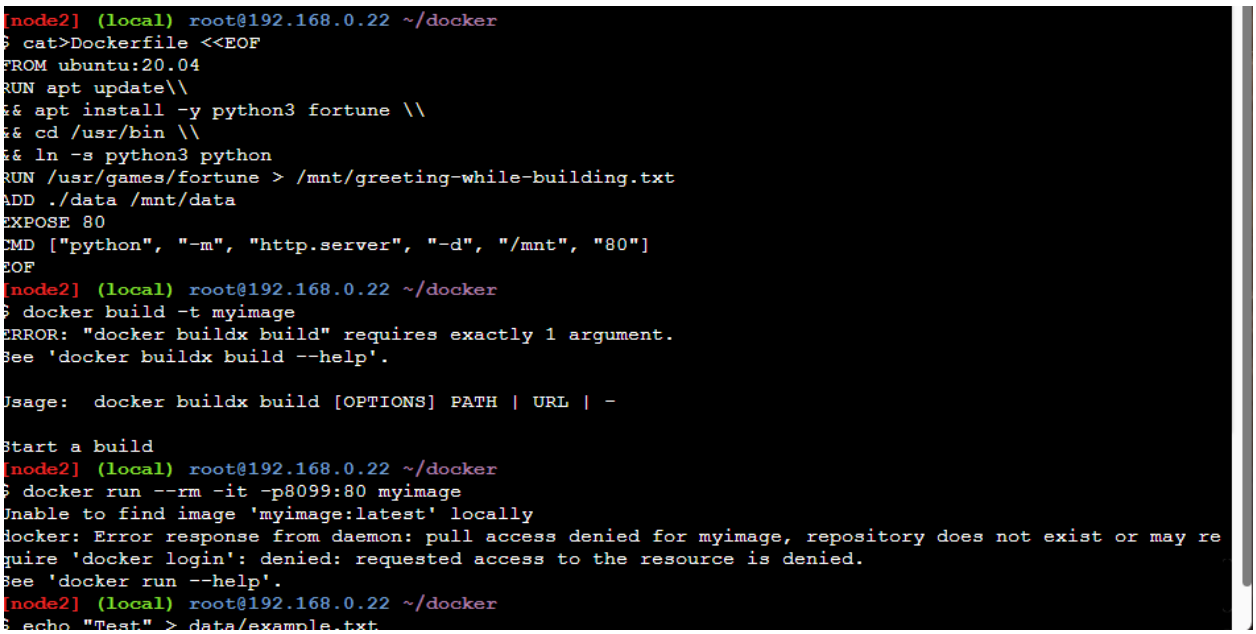


Рисунок 15 – Сборка образа

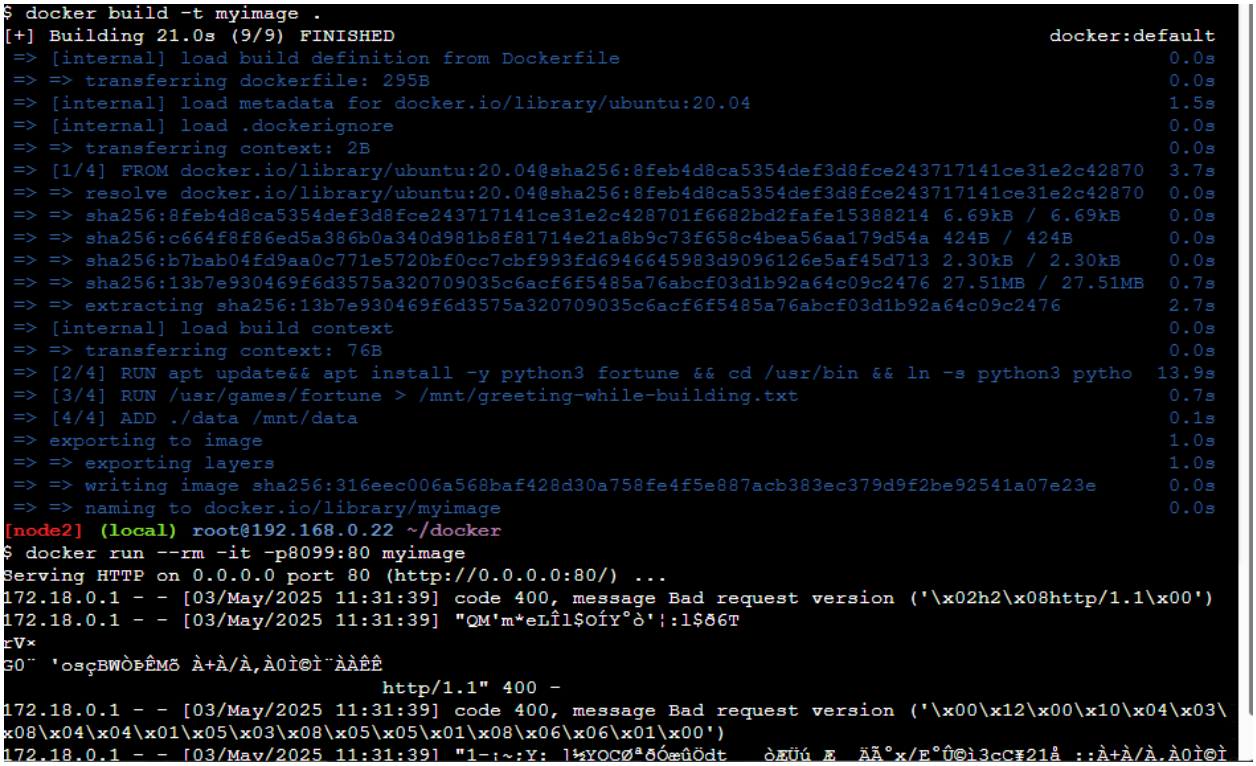


Рисунок 16 – запуск сервера

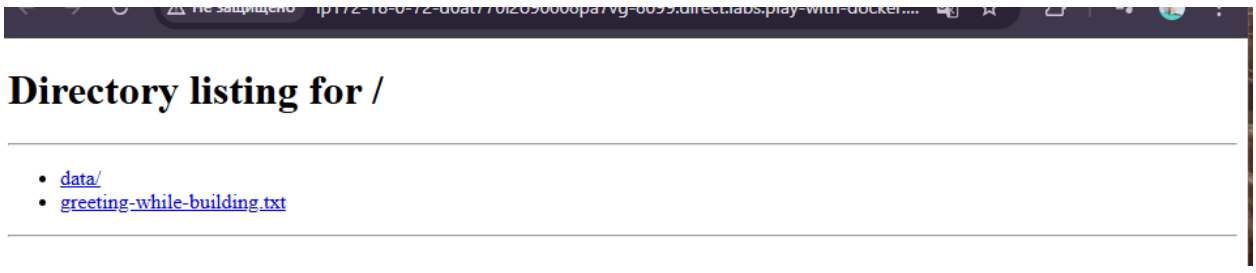


Рисунок 17 – Запуск сервера

Рисунок 18 – Содержимое сгенерированного файла

Рисунок 19 – Содержимое файла data

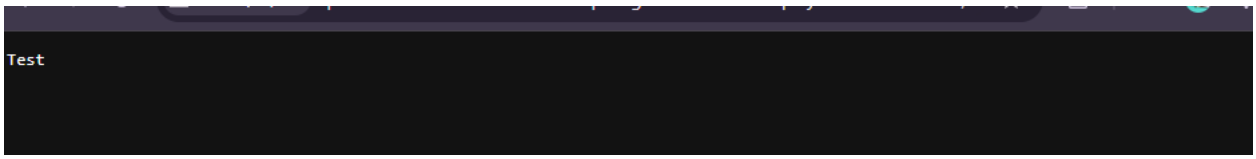


Рисунок 20 – Содержимое файла example.txt

Индивидуальный вариант

Необходимо использовать базовый образ ubuntu:20.04. Примонтировать директорию data в директорию /mnt/files/ в контейнере. Запустить веб-сервер, отображающий содержимое /mnt/files, в хостовой системе должен открываться на порту (8800 + номер варианта). Например, для 11-го варианта это порт 8811. Установить пакет gpg

На рисунках 21-24 изображена распаковка образа, запуск контейнера

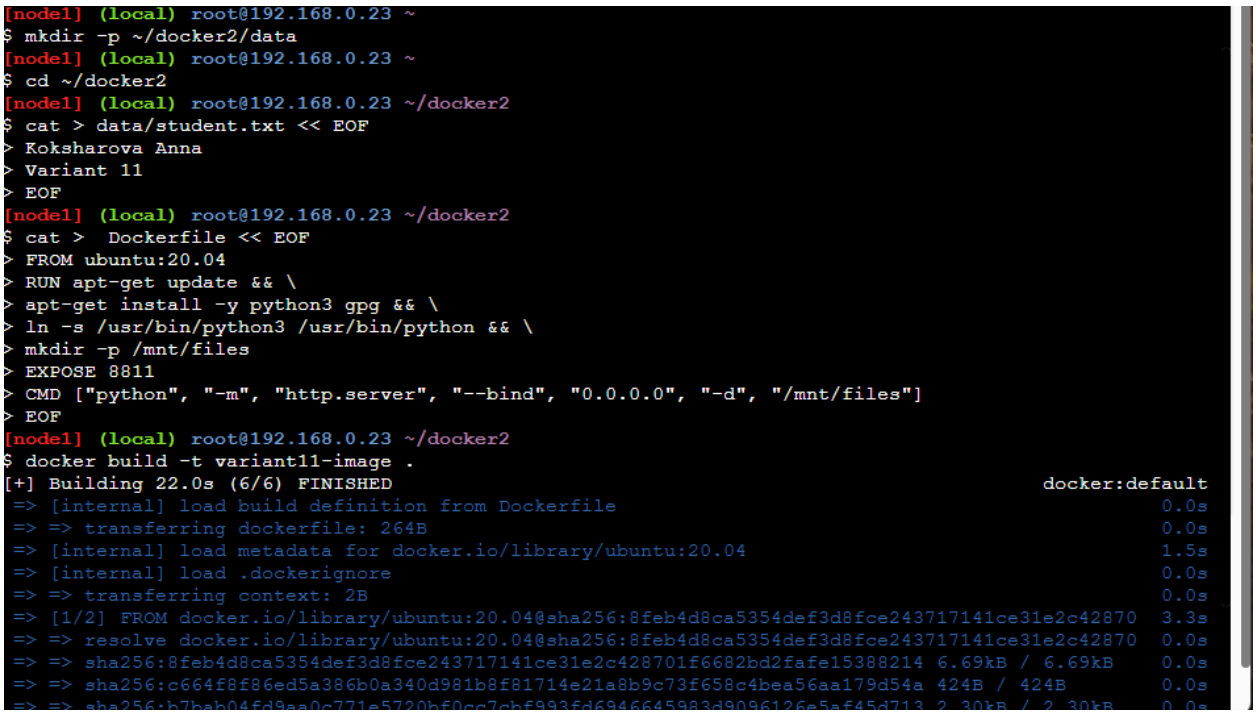


Рисунок 21 – Создание директории

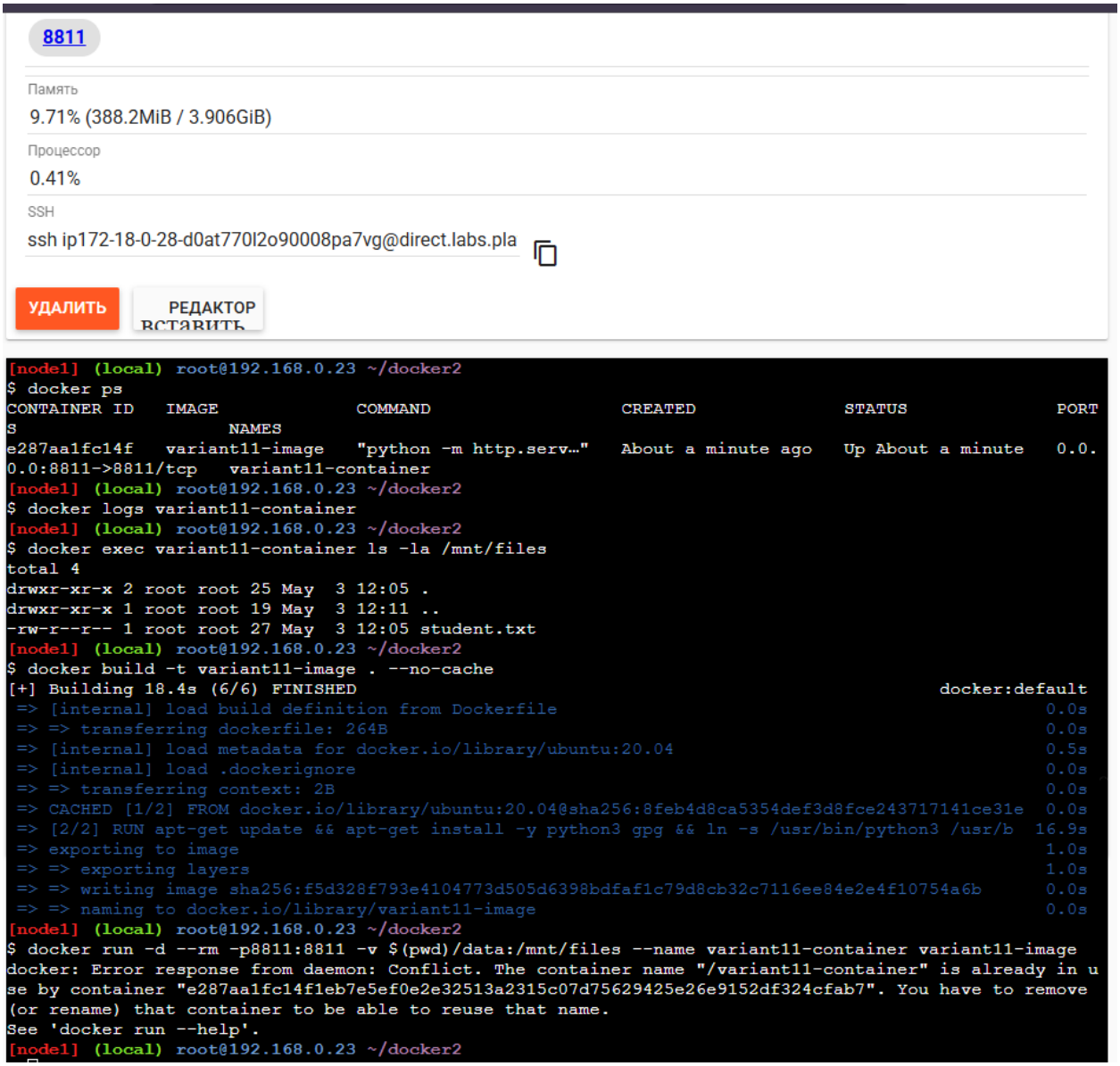


Рисунок 22 – Сборка образа

Рисунок 23 – Запуск

Вывод

В данной работе было установлены и проверены образы ubuntu и python, изучена изоляция контейнеров, освоена информация с портами, проверено монтирование томов и директорий