|  |  |
| --- | --- |
| Федеральное агентство железнодорожного транспорта  Омский государственный университет путей сообщения  Кафедра «Автоматика и системы управления»  НАСТРОЙКА РЕСУРСОВ КОНТЕЙНЕРОВ DOCKER  Лабораторная работа № 4  по дисциплине «Информационные технологии» | |
|  | Студент гр. 22м                            Н.И. Яковлев  «    »                  2023 г.  Руководитель –  старший преподаватель кафедры «АиСУ»                            Т. В. Васеева  «    »                  2023 г. |
| Омск 2023 | |

Цель работы

Целью данной лабораторной работы является изучение взаимодействия контейнеров docker с операционной системой.

# Выполнение работы

## Создание общей папки для контейнера

Скачаем образ docker «gcc», далее запустим контейнер с указаными в задании ключами (рисунок 1-2).

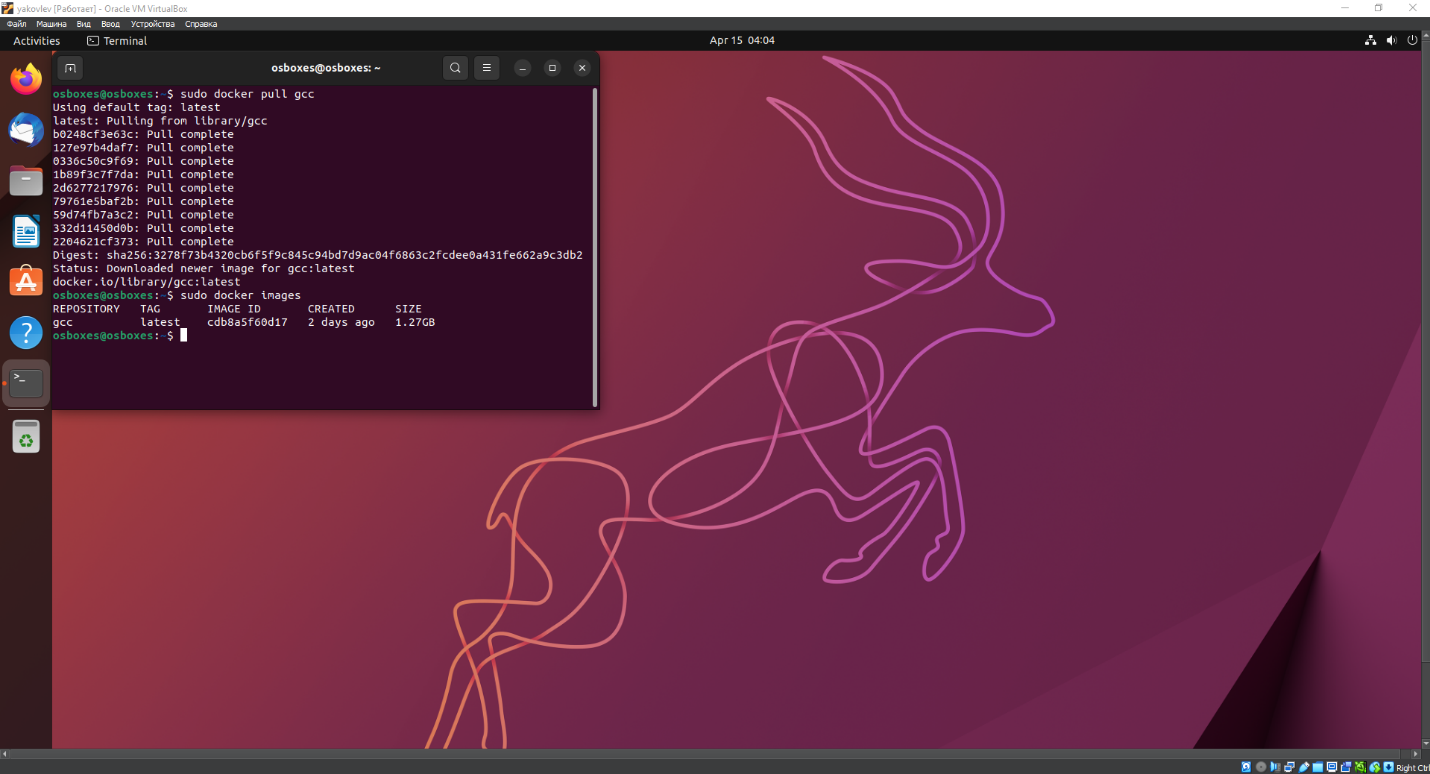


Рисунок  – Установка образа «gcc»

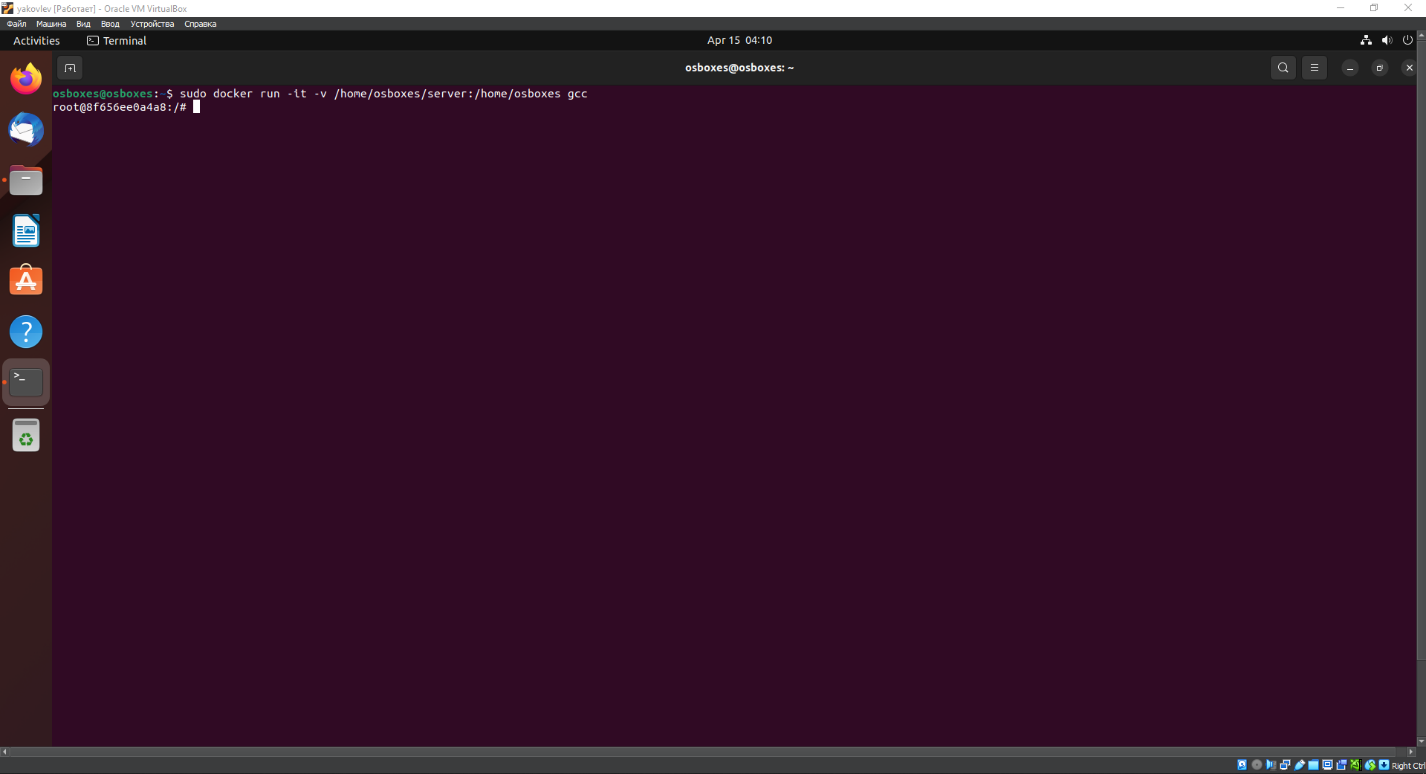


Рисунок  – Запуск контейнера «gcc»

На ФМ создадим файл «hello.c» (рисунок 3).

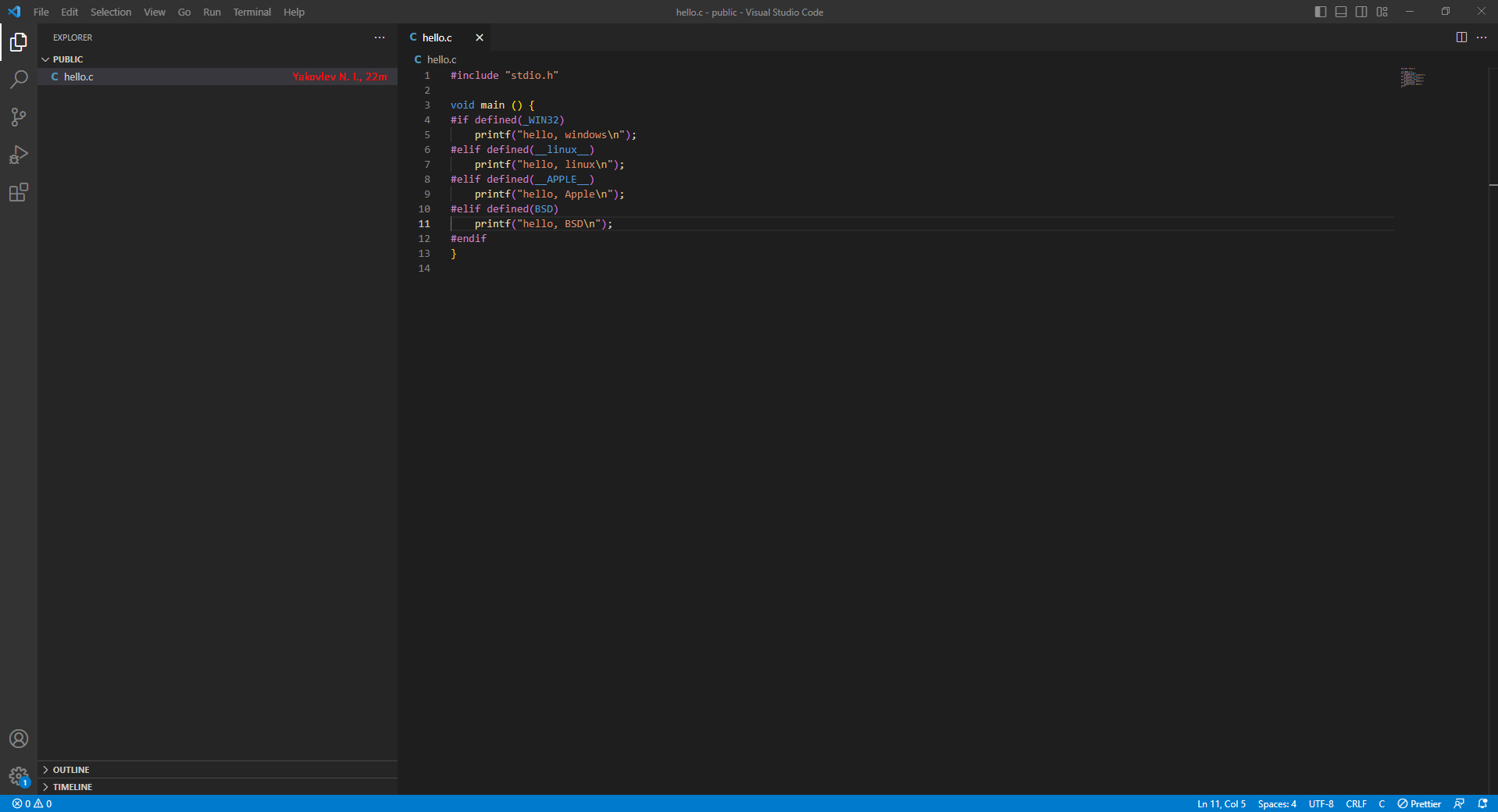


Рисунок  – Файл «hello.c»

Скомпилируем и запустим программу на ФМ и на ВМ в контейнере (рисунок 4-5). Также скомпилируем первую лабораторную работы по программированию (рисунок 6).

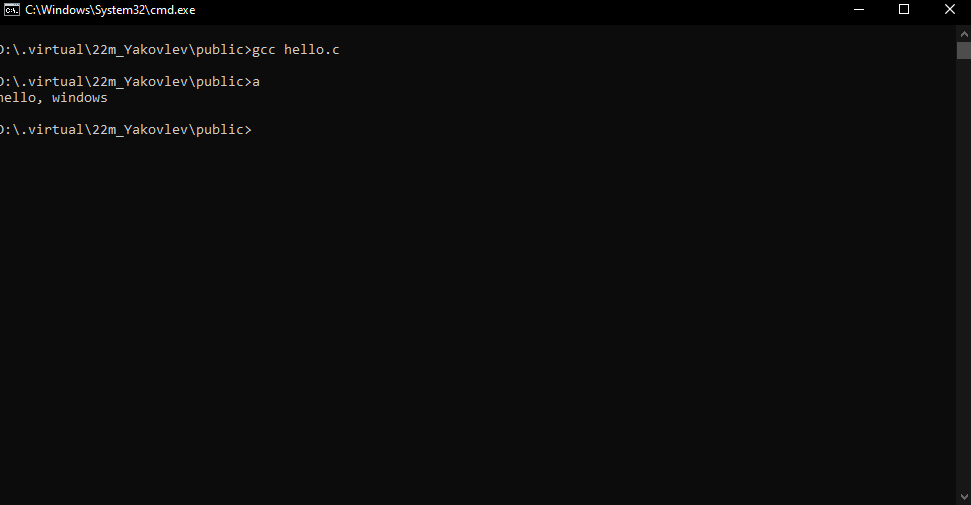


Рисунок  – Запущенный файл на ФМ

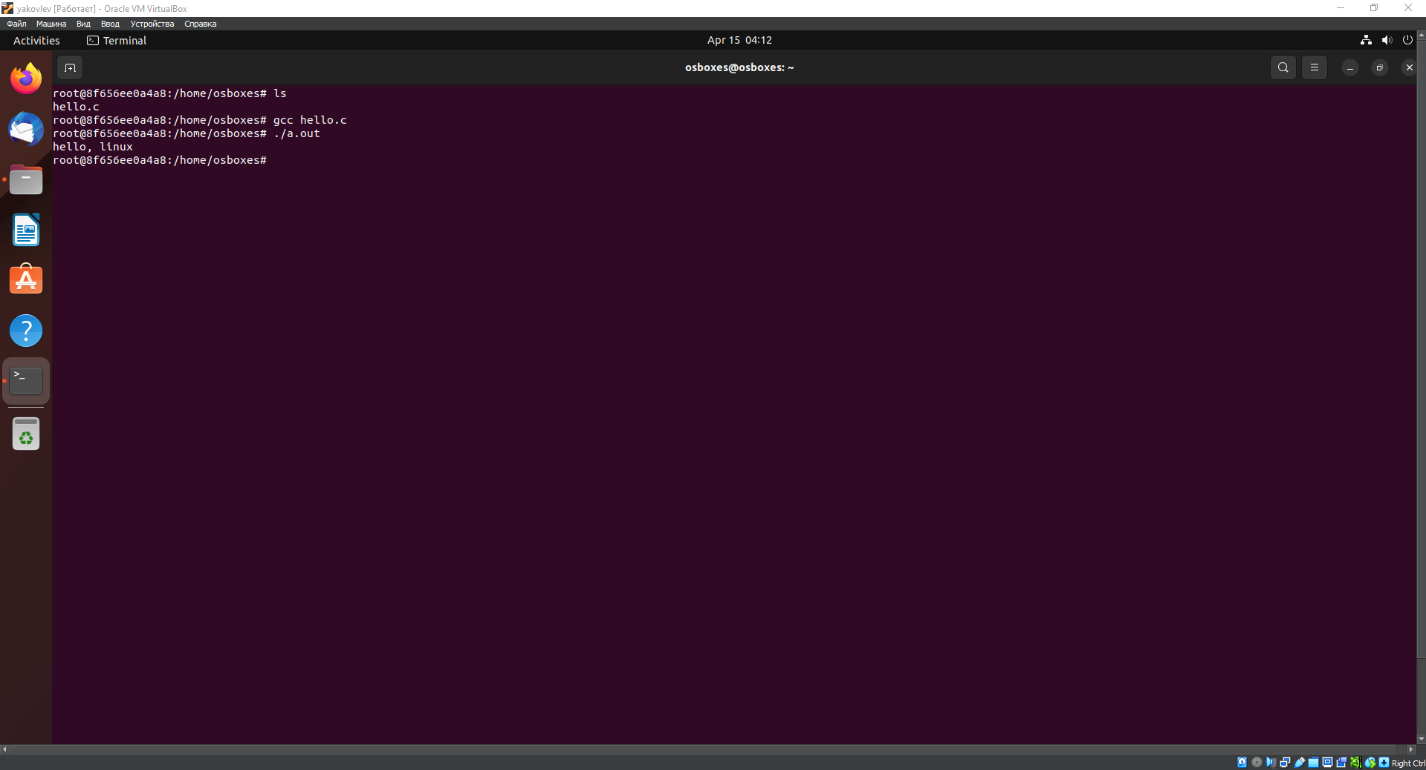


Рисунок  – Запущенный файл на ВМ

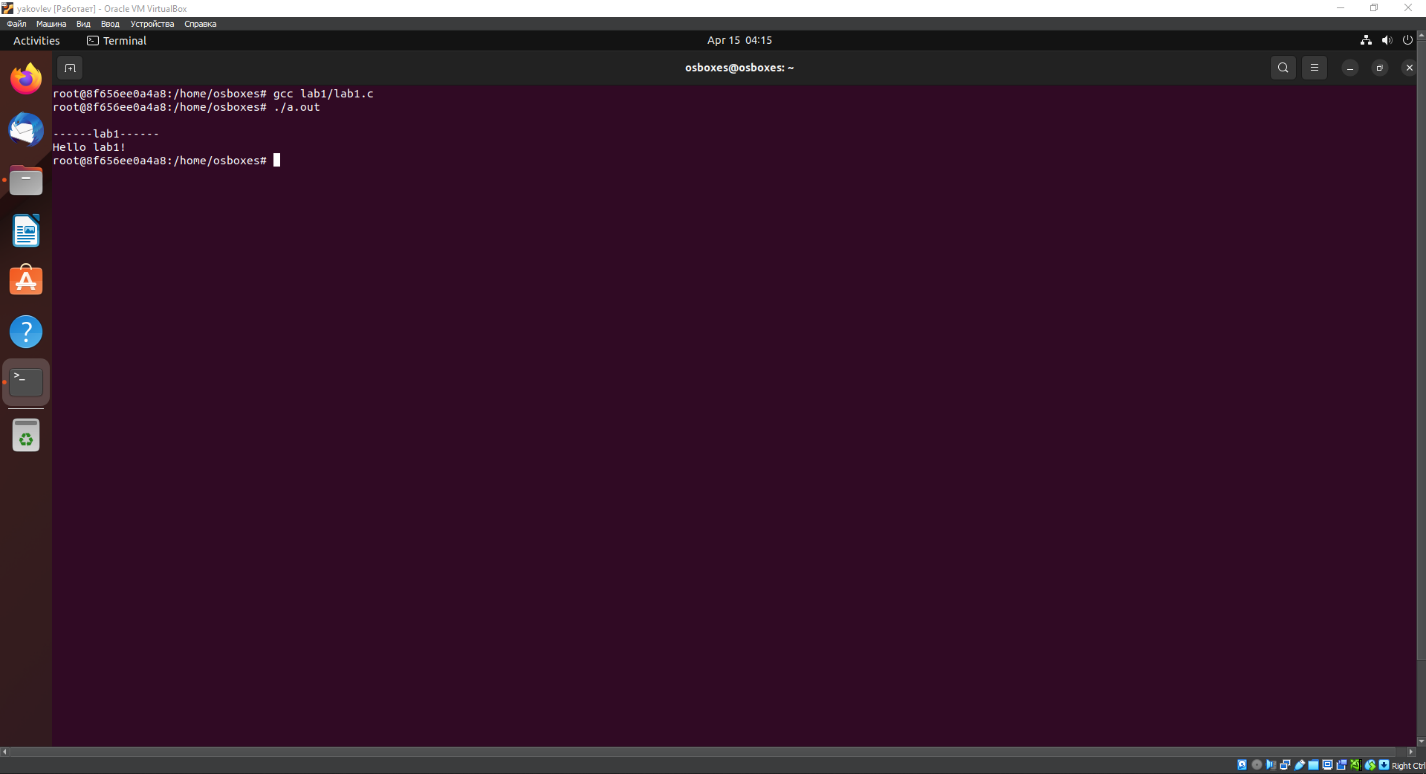


Рисунок  – Лабораторная работа №1

## Проброс портов

Скачаем образ docker «gutbucket», пробросим порты и запустим контейнер с указаными в задании ключами (7-8).

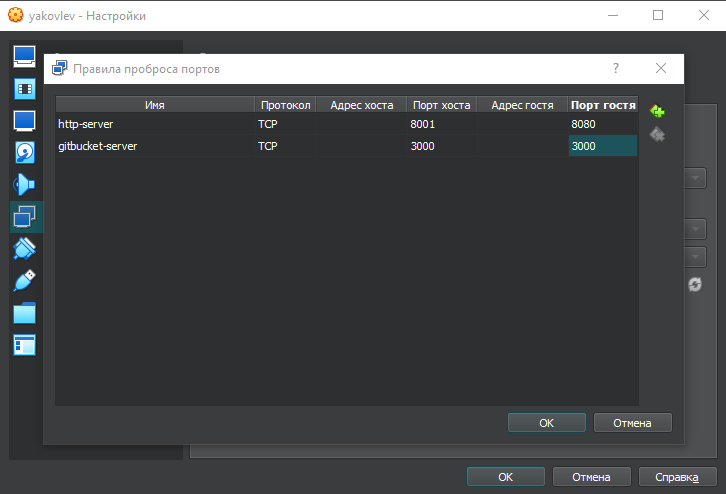


Рисунок  – Проброс портов

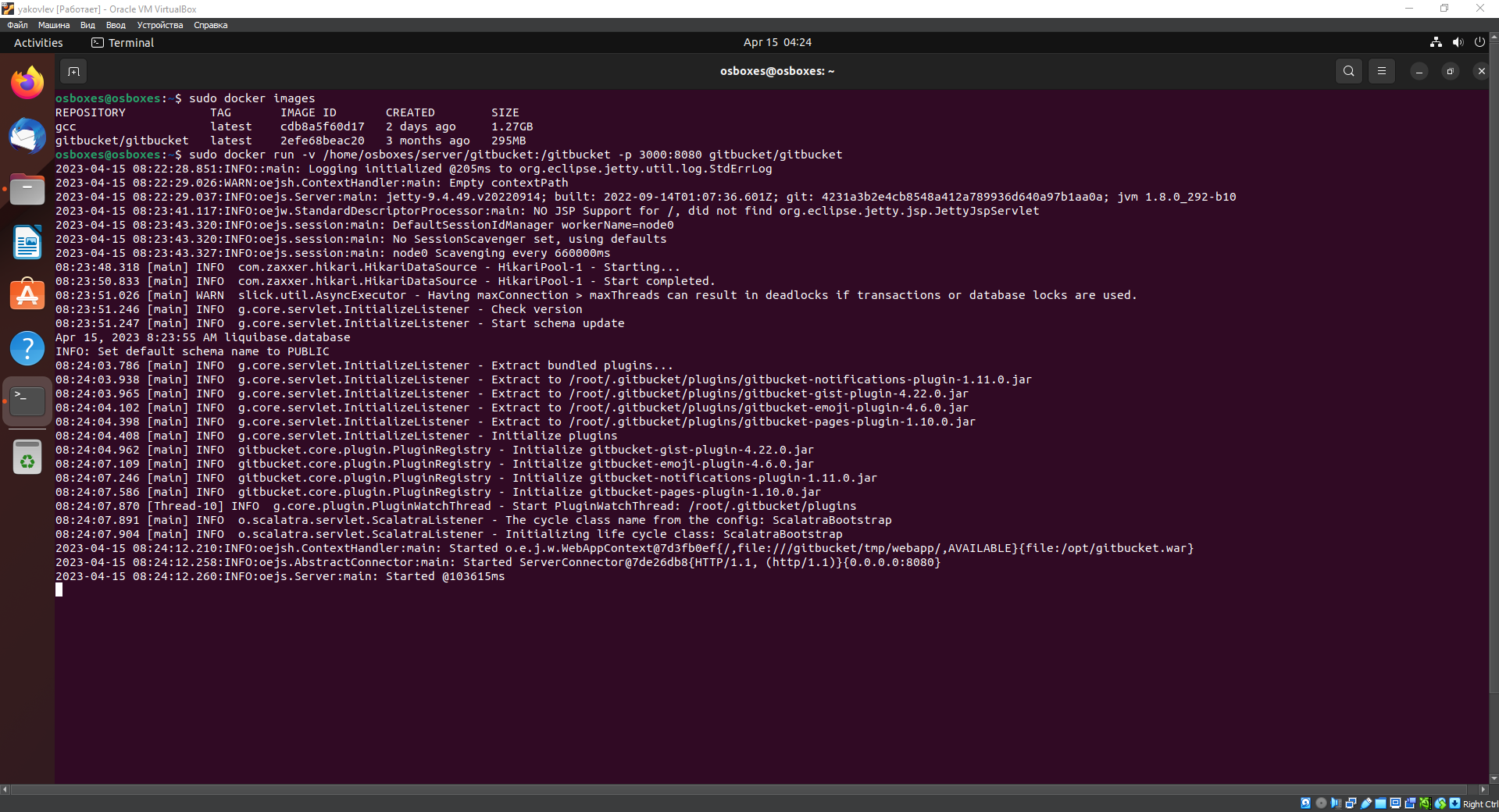


Рисунок  – Запуск сервера gitbucket

Зайдем на сервер gutbucket через браузер на ФМ (рисунок 9).

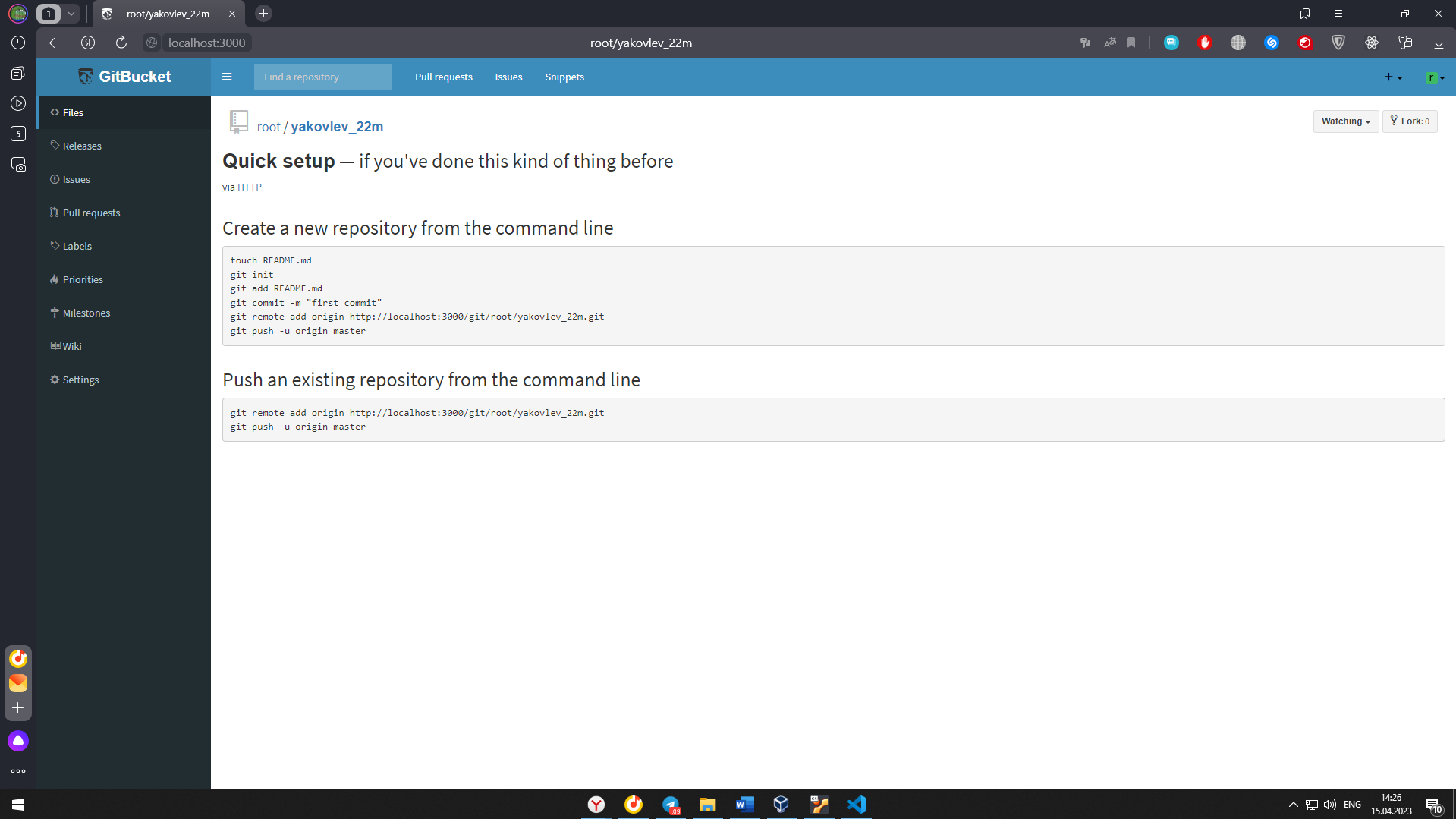


Рисунок  – Сервер gutbucket в браузере

## Взаимодействие контейнеров

Следуя заданию, создадим репозиторий в контейнере ВМ и зальем этот репозиторий на сервер gutbucket (рисунок 10).

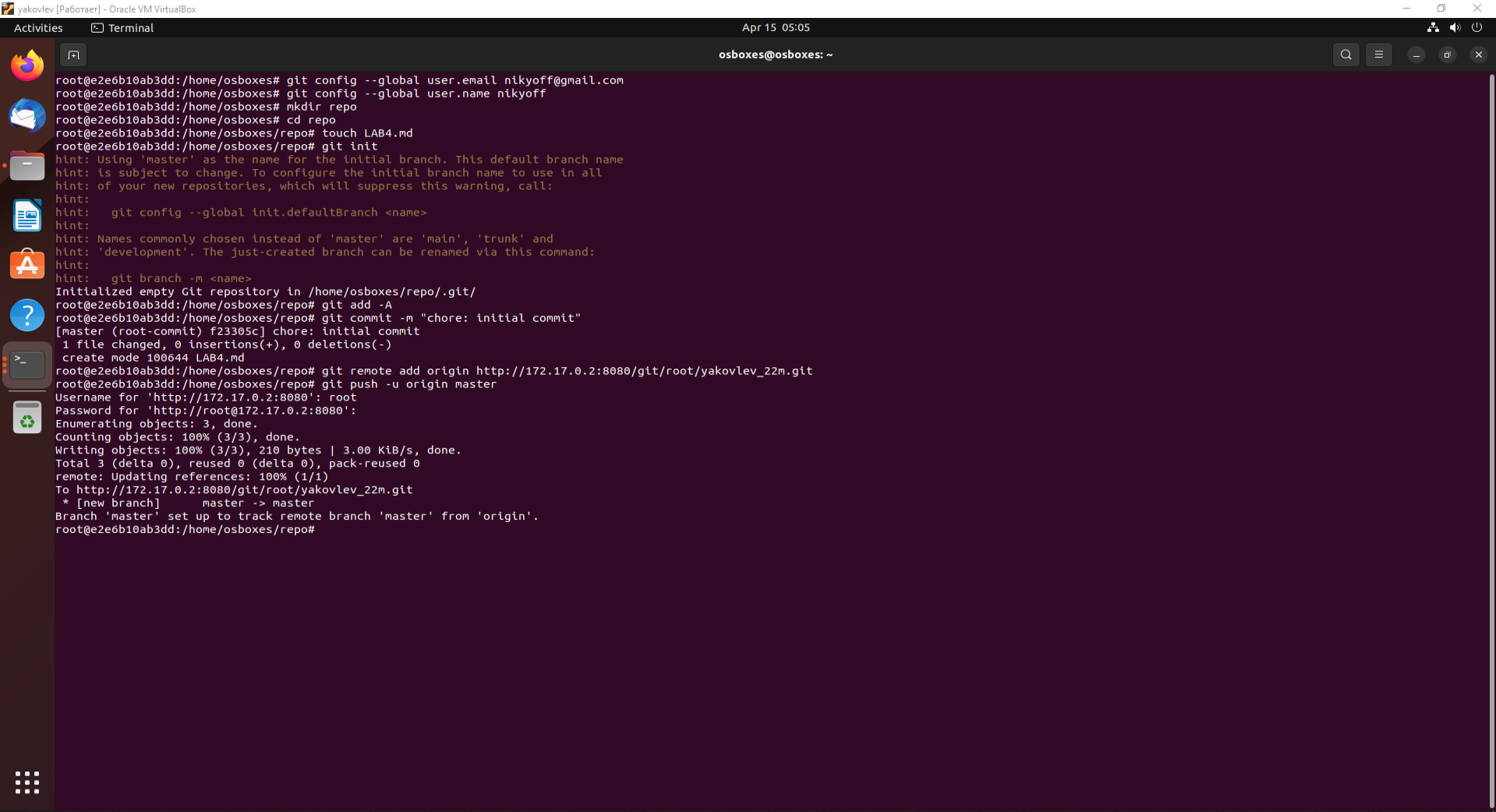


Рисунок  – Создание репозитория

Зайдем на сервер gutbucket через браузер на ФМ и убедимся в созданном контейнере (рисунок 11).

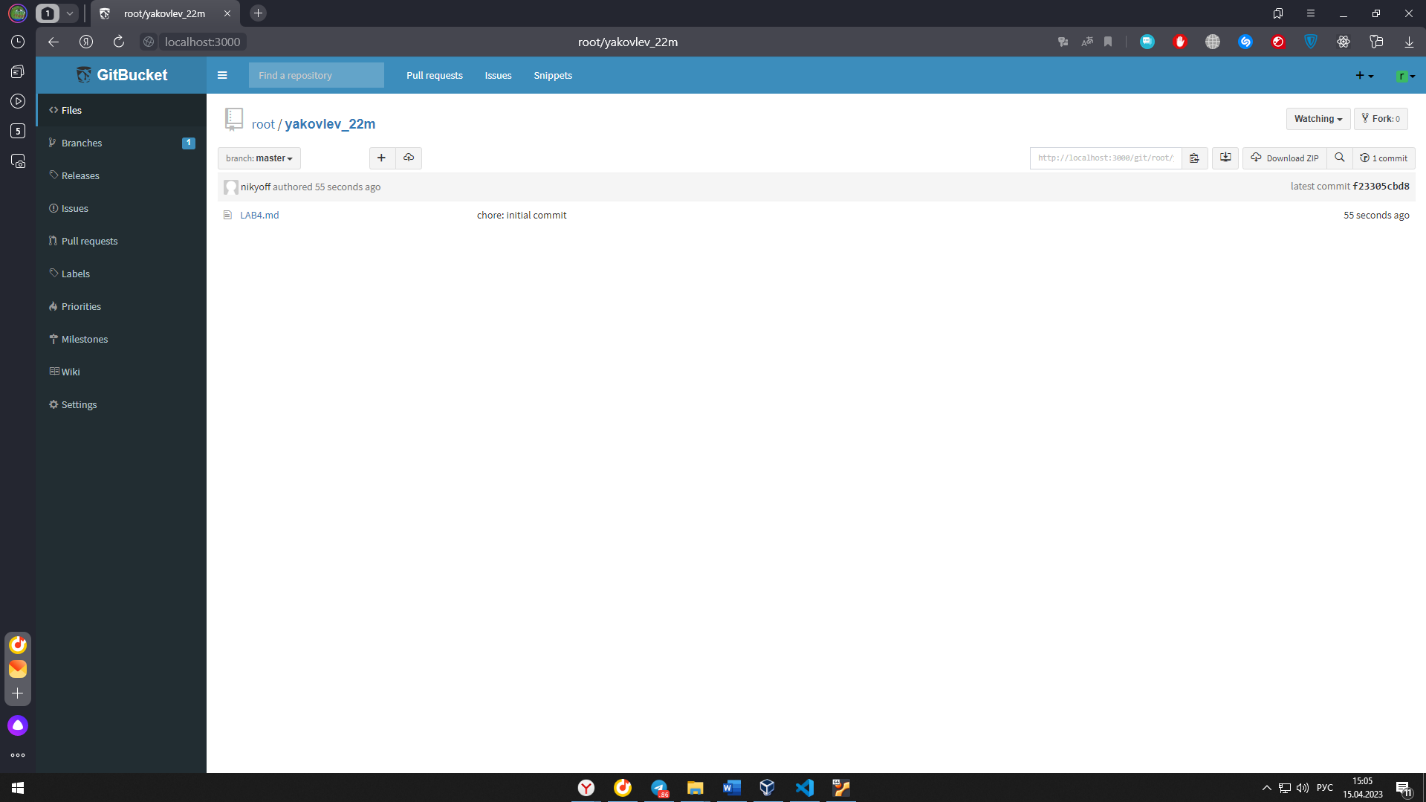


Рисунок  – Созданный репозиторий

## Спислк испльзуемых команд в терминале Linux

Таблица №1 – Команды

|  |  |  |
| --- | --- | --- |
| Команда | Значение | Расшифровка ключей |
| docker pull {image} | Установка образа |  |
| docker run {image} | Запуск нового контейнера из образа | v: для создания общей папки  p: для проброса портов |
| gcc {filename} | Компиляция C-файла |  |
| git config {name} {value} | Устанавливает значение конфигурации |  |
| mkdir {foldername} | Создает папку |  |
| cd {foldername} | Переход в папку |  |
| touch {filename} | Создает файл |  |
| git init | Инициалиация репозитория |  |
| git add | Добавляет изменения в список изменений | A: добавляет все файлы |
| git commit | Фиксирует изменения | m: указывает сообщение комита |
| git remote add P{name} {remote} | Добавляет удаленный репозиторий |  |
| git push | Вносит изменения в репозиторий | u: устанавливает текущую ветку для отслеживания |

# Контрольные вопросы

## Рейтинг сервисов-репозиториев для хранения кода (5 примеров)

1. GitHub
2. GitLab
3. Bitbucket
4. SourceForge
5. Launchpad

## С помощью Docker Compose запустить два контейнера gcc и gutbucket

Для запуска потребуется создать описывающий yml-файл «docker-compose.yml» (рисунок 12). В нем дается описание нескольких сервисов (контейнеров), также стоит не забыть пробросить порты для gutbucket.

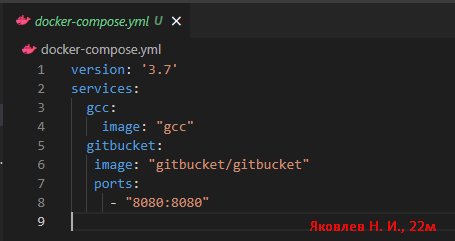


Рисунок  – Файл «docker-compose.yml»

Затем необходимо запустить терминал и использовать команду «docker-compose up» (рисунок 13 - 14).

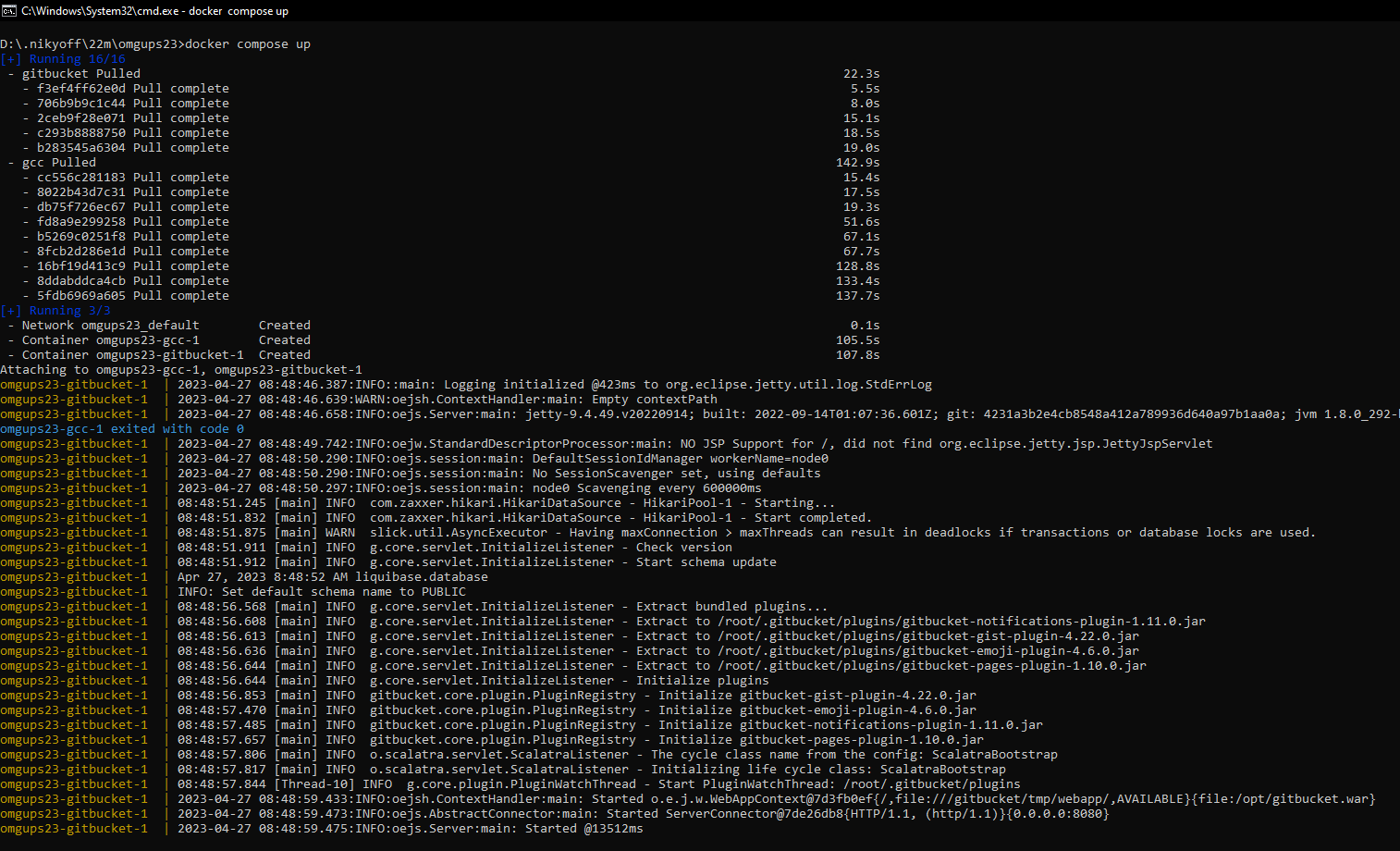


Рисунок  – Запуск docker-compose

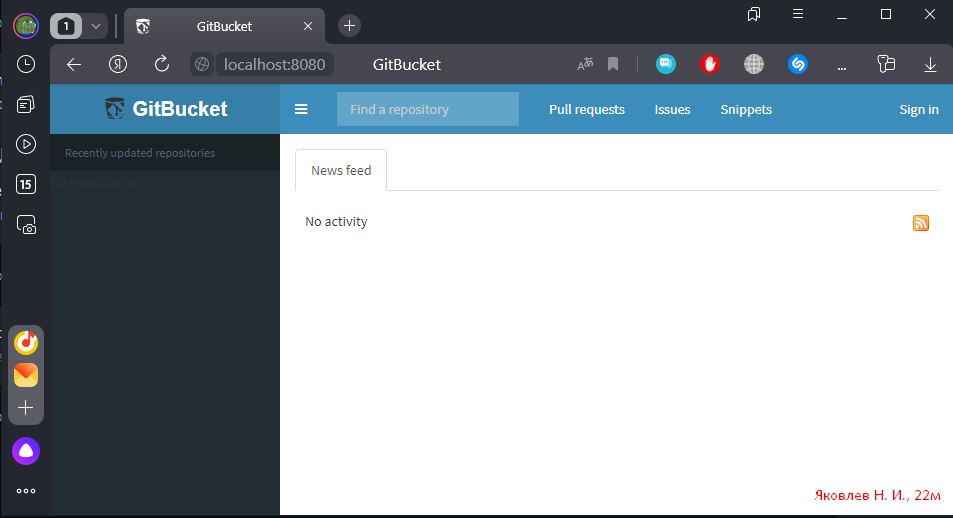


Рисунок  – gitbucket

В результате сможем увидеть наше запущенное многоконтейнерное приложение, а также зайти на локальный сервер gutbucket (рисунок 14-15).

## Пояснить команду: sudo docker network inspect bridge. Как определить IP контейнера? Что такое маска сети?

В общем виде команда выглядит так «docker network inspect [OPTIONS] NETWORK [NETWORK...]» и позволяет получить информациюю об одной или нескольких сетях. В нашем случае «bridge» обозначает стандартный мостовой тип сети используемый в Docker.

Посмотреть IP контейнера можно с помощью команды «docker inspect CONTAINER\_ID», после выполнения этой команды получим объект с информацией о котейнере, в свою очередь нас интересует «HostConfig.NetworkSettings.Networks.NETWORK.IPAddress».

Маска подсети предназначена для того, чтоб можно было определить способ передачи пакета: напрямую или через шлюз по умолчанию. С помощью маски можно определить принадлежность устройства-получателя к локальной сети устройства-отправителя, в тех случаях, когда оба находятся в одной сети, пакет будет отправлен напрямую.

Вывод

В данной лабораторной работе, мы изучили некоторые механизмы docker, разобрались с общими папками и пробросом портов, а также познакомились с базовыми возможности системы контроля версий.