Министерство Образования, Культуры и Исследований

**Молдавский Государственный Университет**

**Факультет Математики и Информатики**

**Департамент Информатики**

**Лабораторная работа №1**

**Базовые команды K8s**

Научный руководитель: А. Prepelita

Автор: студент группы I2102 (РУ), Juracovschi Vladislav

**Кишинев, 2023**

Оглавление

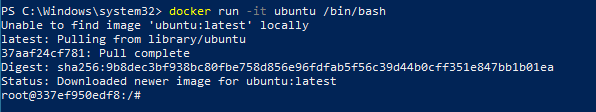
[Выполнение комманд 1](#_Toc28362)

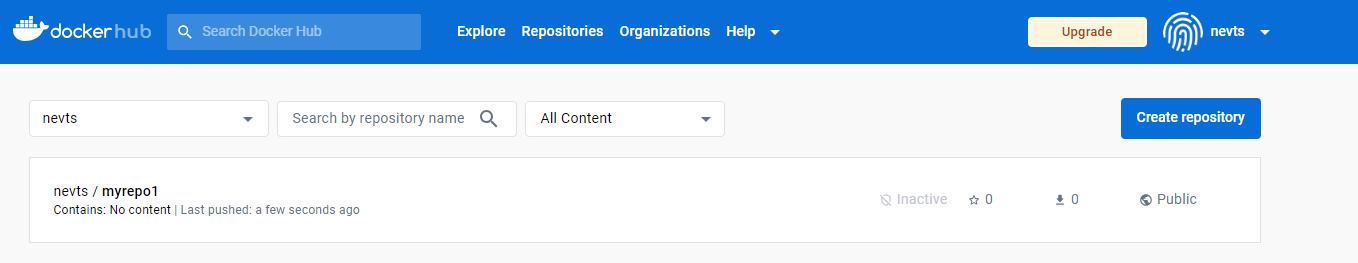
[Вывод 7](#_Toc32353)

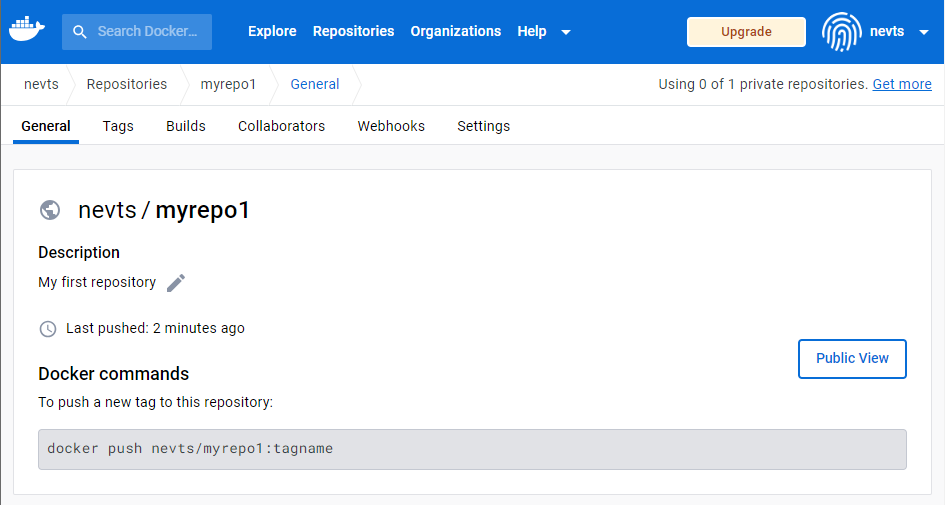
[Библиография 7](#_Toc29924)

# Выполнение комманд

**docker run -it ubuntu /bin/bash** - используется для запуска контейнера в Docker с образом Ubuntu и интерактивным терминалом. Это позволяет войти в контейнер и выполнить команды в его окружении.



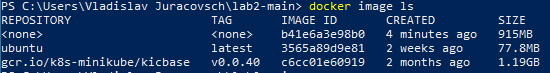




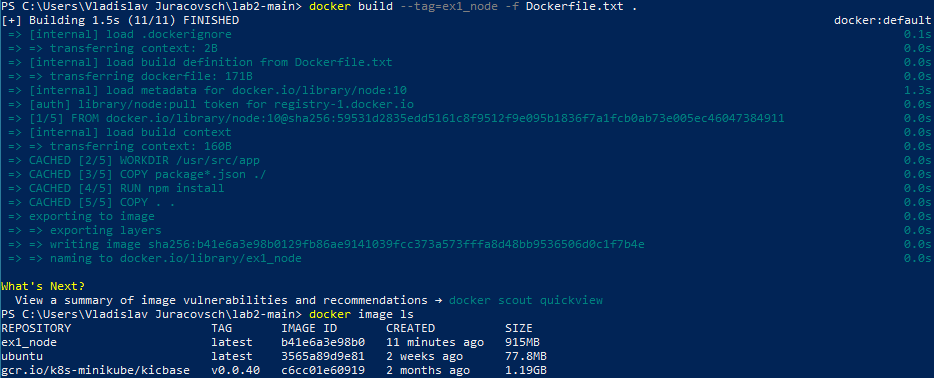
**docker build -f Dockerfile.txt .** - используется для сборки Docker-образа из файла Dockerfile.txt в текущем каталоге. Она позволяет определить, каким образом создавать контейнер для приложения.



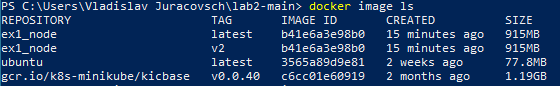
**docker image ls -** возвращает список всех Docker-образов, которые находятся на компьютере.



**docker build --tag=ex1\_node -f Dockerfile.txt . --tag=ex1\_node** задает имя для создаваемого Docker-образа.



**docker image ls** - tag в Docker используется для создания новой метки для существующего образа контейнера.



**docker run -p 4000:8080 ex1\_node:v1 -** команда запустит контейнер с указанным именем и версией, и прокинет порт 4000 с хоста на порт 80 внутри контейнера, что позволит взаимодействовать с приложением, работающим внутри контейнера, через порт 4000 на хосте.



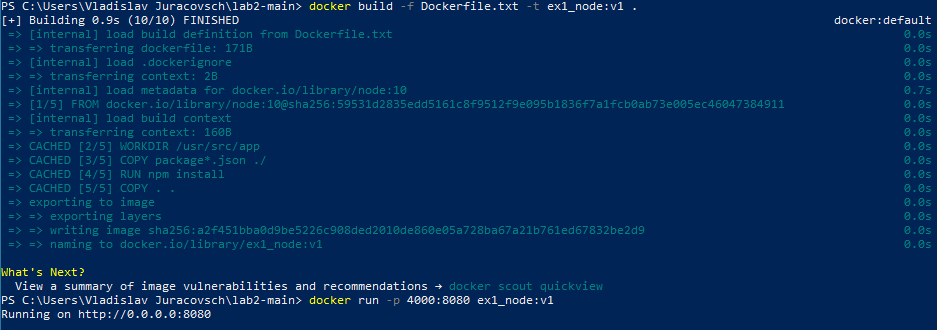


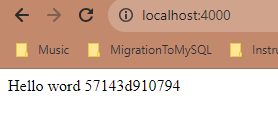
**docker inspect -f "{{.NetworkSettings.IPAddress}}" c0768ae34594** - выводит ip для контейнера по ид



**docker build -f Dockerfile.txt -t ex1\_node:v1 .** - сбор докера по образу файла

**docker run -p 4000:8080 ex1\_node:v1 -** поднятие докера на порт 4000



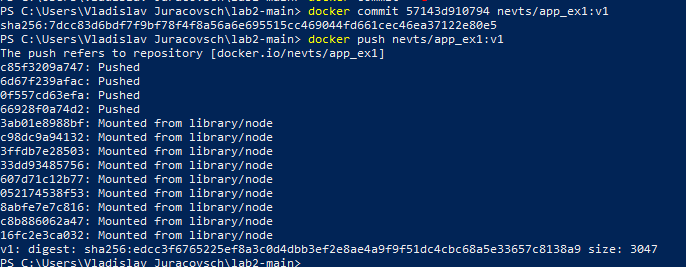


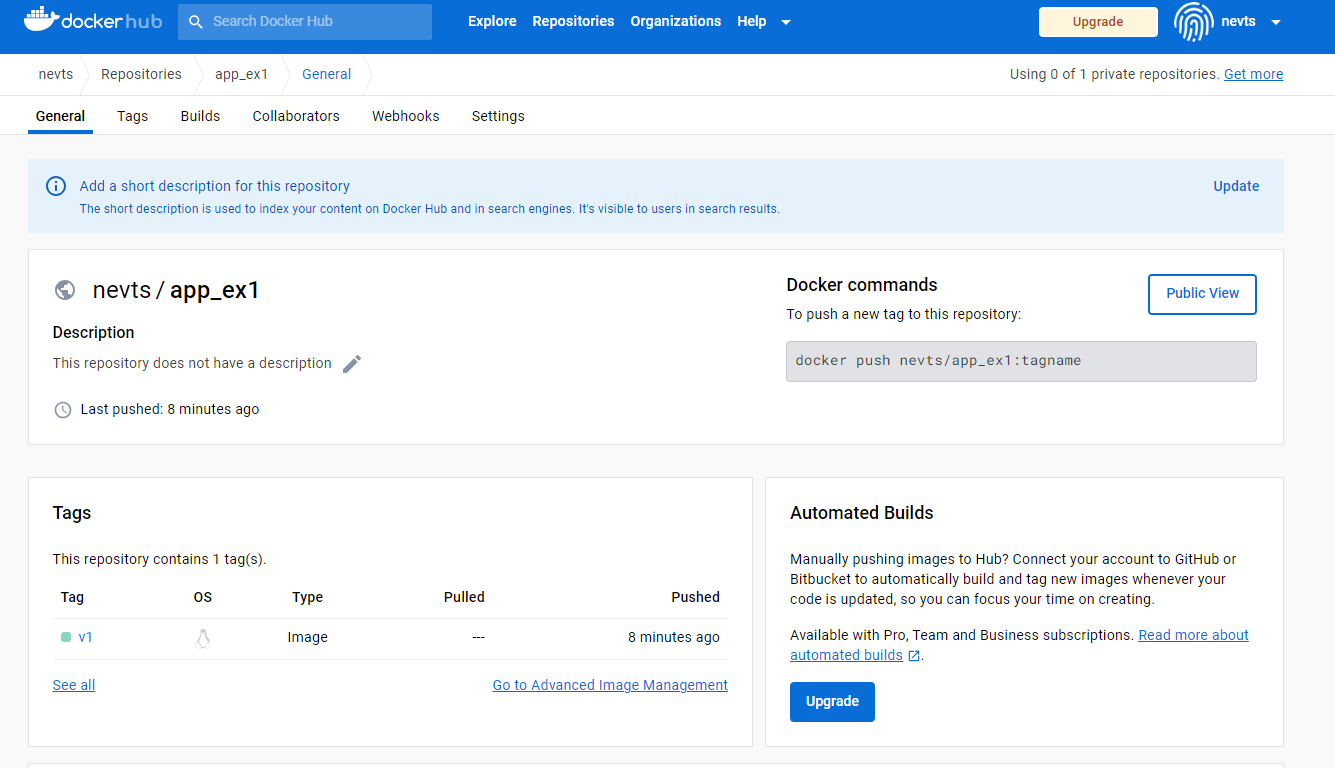
**docker login --username=nevts** - вход в аккаунт



**docker commit 57143d910794 nevts/app\_ex1:v1** - создает новый образ Docker на основе существующего контейнера.

docker push nevts/app\_ex1:v1 - используется для загрузки локального Docker-образа на удаленный Docker-реест.

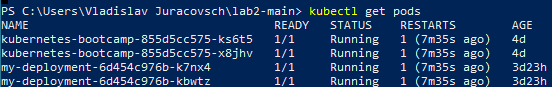


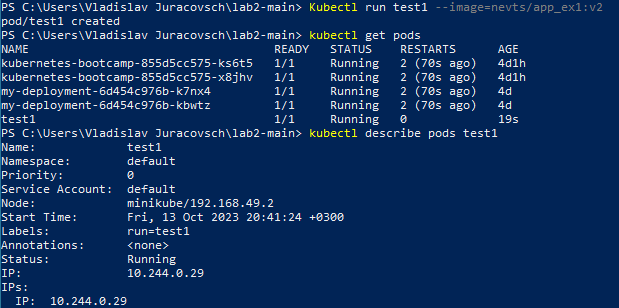




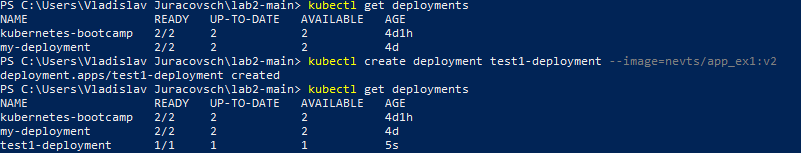
**Kubectl run test1 --image=nevts/app\_ex1:v2** - используется для запуска контейнера в кластере Kubernetes



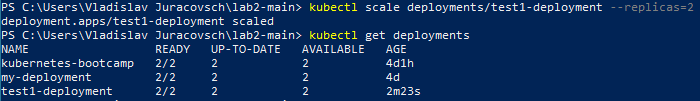




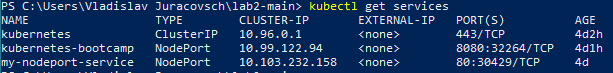
**kubectl create deployment test1-deployment --image=nevts/app\_ex1:v2** - создание образа для реплик



**kubectl scale deployments/test1-deployment --replicas=2** - используется для масштабирования развёртывания в Kubernetes, устанавливая количество реплик на 2.



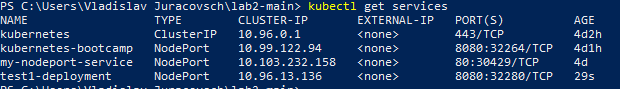
**kubectl get services -** это команда в Kubernetes, используемая для получения информации о службах.



**kubectl expose deployment/test1-deployment --type="NodePort" --port 8080** - для создания сервиса Kubernetes для развернутого деплоймента "test1-deployment". Этот сервис будет доступен снаружи кластера через NodePort на порту 8080.



**kubectl get services** - команда для управления контейнерами в Kubernetes. Она используется для получения информации о сервисах в вашем кластере, таких как IP-адреса и порты.



**kubectl proxy -**команда в утилите Kubernetes, которая создает локальный прокси-сервер для взаимодействия с API-сервером Kubernetes.



# Вывод

Цель использования команд Docker и Kubernetes заключается в управлении контейнерами и приложениями внутри них. Эти команды предоставляют инструменты для создания, запуска, масштабирования и управления контейнеризированными приложениями. Docker используется для создания, запуска и распространения контейнеров, а Kubernetes - для управления контейнерами в кластере, включая масштабирование и управление приложениями.

# 

# Библиография

1. <https://hub.docker.com/_/docker>
2. <https://kubernetes.io/docs/tasks/extend-kubernetes/http-proxy-access-api/>
3. <https://loft.sh/blog/when-and-how-to-use-kubectl-proxy/>