

# Лабораторная работа №1

**Тема:** Работа с Docker.

**Цель:** Познакомиться с возможностями и получить практические навыки работы с Docker.

**Ссылки:**

1. <https://docs.docker.com>
2. <https://hub.docker.com/>
3. <https://doka.guide/tools/dockerfile/>
4. <https://k21academy.com/docker-kubernetes/docker-storage/>
1. <https://k21academy.com/docker-kubernetes/docker-tutorial>
2. <https://k21academy.com/docker-kubernetes/docker-networking-different-types-of-networking-overview-for-beginners/>
3. <https://ru.bitdegree.org/rukovodstvo/docker-dlja-novichkov/>
4. <https://habr.com/ru/companies/slurm/articles/528206/>
5. <https://itisgood.ru/2019/09/10/docker-setevoe-vzaimodejstvie-101/>
6. <https://itisgood.ru/2019/10/29/objasnenie-koncepcii-setej-v-docker/>
7. <https://www.tune-it.ru/web/adpashnin/blog/-/blogs/docker-network>

Для защиты ЛР необходимо

- Оформить Отчет со скринами кода в командной строке и результатами его выполнения
- Продемонстрировать работу с командами docker: build, tag, run, start, stop, pause, unpause, restart, ps, logs, ps, images, network, volumes, inspect, а также знать назначение инструкций в Dockerfile и Docker Compose

**Задание:**

1. Подготовьте рабочее окружение в соответствии с типом вашей операционной системы
  - Установите Docker
  - Выполните базовую настройку
2. Изучите простейшие консольные команды и возможности Docker Desktop (см. лекцию), создать собственный контейнер docker/getting-started, открыть в браузере и изучить tutorial
3. Создайте docker image, который запускает скрипт с использованием функций из [https://github.com/smartikaorg/geometric\\_lib](https://github.com/smartikaorg/geometric_lib).
  - a. Данные необходимые для работы скрипта передайте любым удобным способом (например: конфиг файл через docker volume, переменные окружения, перенаправление ввода). Изучите простейшие консольные команды для работы с docker(см. лекцию). Зарегистрируйтесь на DockerHub и выберите необходимые для проекта образы
  - b. Создать Dockerfile для реализации сборки собственных Docker образов
  - c. Использовать его для создания контейнера. Протестировать использование контейнера

4. Скачать любой доступный проект с GitHub с произвольным стеком технологий или использовать свой, ранее разработанный. Создать для него необходимый контейнер, используя Docker Compose для управления многоконтейнерными приложениями. Запустить проект в контейнере.( Примеры Images: [https://hub.docker.com/\\_/phpmyadmin](https://hub.docker.com/_/phpmyadmin), [https://hub.docker.com/\\_/mysql](https://hub.docker.com/_/mysql), [https://hub.docker.com/\\_/postgres](https://hub.docker.com/_/postgres))
5. Настроить сети и тома для обеспечения связи между контейнерами и сохранения данных (исходные данные, логин, пароль и т.д.)
6. Разместите результат в созданный репозиторий в DockerHub
7. Выполните следующие действия с целью изучить особенности сетевого взаимодействия:
  - Получить информацию о всех сетях, работающих на текущем хосте и подробности о каждом типе сети
  - Создать свою собственную сеть bridge, проверить, создана ли она, запустить Docker-контейнер в созданной сети, вывести о ней всю информацию(включая IP-адрес контейнера), отключить сеть от контейнера
  - Создать еще одну сеть bridge, вывести о ней всю информацию, запустить в ней три контейнера, подключиться к любому из контейнеров и пропинговать два других из оболочки контейнера, убедиться, что между контейнерами происходит общение по IP-адресу
  - Создать свою собственную сеть overlay, проверить, создана ли она, вывести о ней всю информацию
  - Создать еще одну сеть overlay, проверить, создана ли она, вывести о ней всю информацию, удалить сеть
  - Попробовать создать сеть host, сохранить результат в отчет.