Лабораторная работа №4

Выполнил Шардт М.А.

# Docker

Docker — это платформа контейнеризации, которая позволяет разработчикам создавать, развертывать и управлять приложениями в контейнерах. Контейнеры обеспечивают изоляцию приложений и их зависимостей, что упрощает переносимость и масштабируемость.

## Для чего

* Разработка и тестирование

Docker обеспечивает гарантию одинаковой работы приложение как во время разработки на машине разработчика, так и на рабочем сервере. Контейнеры создаются на основе определенной операционной системы с указанными зависимости и их версиями, таким образом обеспечивается независимость контейнера от внешних факторов, влияющих на работу приложения.

* CI/CD

Docker широко используется в Cloud среде. С помощью Docker возможно автоматизировать сборку, тестирование и деплой приложения, что уменьшает количество ошибок и ускоряет процесс обновления и доставки приложения.

## Применение и примеры

1. Запуск базы данных для приложения в контейнере

| version: '3.1'  services:  db:  image: postgres:15  restart: always  environment:  POSTGRES\_USER: postgres  POSTGRES\_PASSWORD: postgres  POSTGRES\_DB: NFB  PGDATA: /var/lib/postgresql/data  ports:  - 5432:5432  volumes:  - db:/var/lib/postgresql/data/  volumes:  db: |
| --- |

1. Запуск java приложения

| FROM openjdk:21-jdk ARG JAR\_FILE=target/\*.jar COPY ${JAR\_FILE} app.jar ENTRYPOINT ["java","-jar","/app.jar"] |
| --- |

1. Запуск нескольких go приложений на разных портах с получением ssl сертификатов

| version: "2"  services:  nginx-proxy:  image: nginxproxy/nginx-proxy  container\_name: nginx-proxy  ports:  - "80:80"  - "443:443"  volumes:  - vhost:/etc/nginx/vhost.d  - conf:/etc/nginx/conf.d  - html:/usr/share/nginx/html  - certs:/etc/nginx/certs:ro  - /var/run/docker.sock:/tmp/docker.sock:ro  *# network\_mode: bridge*   acme-companion:  image: nginxproxy/acme-companion  container\_name: nginx-proxy-acme  environment:  - DEFAULT\_EMAIL=mxschardt@gmail.com  volumes\_from:  - nginx-proxy  volumes:  - certs:/etc/nginx/certs:rw  - acme:/etc/acme.sh  - /var/run/docker.sock:/var/run/docker.sock:ro  *# network\_mode: bridge*   ping-pong:  build: ./ping-pong  restart: always  ports:  - "91:80"  environment:  VIRTUAL\_HOST: mxschardt.ru  LETSENCRYPT\_HOST: mxschardt.ru  PORT: "80"  MESSAGE: "ping!"  command: "./bin"  volumes:  vhost:  conf:  html:  certs:  acme: |
| --- |