Задача 2

В данной задаче вы составите несколько разных Dockerfile для проекта Jenkins, опубликуем образ в dockerhub.io и посмотрим логи этих контейнеров.  
  
FROM jenkins/jenkins:lts

EXPOSE 8080:8080 50000:50000  
  
Ответ: см. Jenkins docker fileДомашнее задание к занятию 5.4. Практические навыки работы с Docker  


administrator@ubuntu:~/MyDockerTest$ docker run -d -ti --name jenkinscontainer -v jenkins\_home:/var/jenkins\_home myjenkins

**Задача 3**

В данном задании вы научитесь:

объединять контейнеры в единую сеть

исполнять команды "изнутри" контейнера

Для выполнения задания вам нужно:

Написать Dockerfile:

Использовать образ <https://hub.docker.com/_/node> как базовый

Установить необходимые зависимые библиотеки для запуска npm приложения <https://github.com/simplicitesoftware/nodejs-demo>

Выставить у приложения (и контейнера) порт 3000 для прослушки входящих запросов

Соберите образ и запустите контейнер в фоновом режиме с публикацией порта

Запустить второй контейнер из образа ubuntu:latest

Создайть docker network и добавьте в нее оба запущенных контейнера

Используя docker exec запустить командную строку контейнера ubuntu в интерактивном режиме

Используя утилиту curl вызвать путь / контейнера с npm приложением

Для получения зачета, вам необходимо предоставить:

Наполнение Dockerfile с npm приложением

Скриншот вывода вызова команды списка docker сетей (docker network cli)

Скриншот вызова утилиты curl с успешным ответом

root@c0e57534a371:/# apt-get update; apt-get install curl # устанавливаем curl  
  
root@c0e57534a371:/# curl -v -p 3000 172.20.0.2 # проверяем пинг в другой контейнер, где -v подробно -p по порту 3000 (открыли по заданию в другом контейнере).