Нагорный Степан ИА-131

Курс «Архитектура вычислительных систем»

Лабораторная работа № 4

**Тема:** Распределение нагрузки между экземплярами масштабируемого приложения

**1)** Разработать docker-compose файл запускающий несколько экземпляров docker-приложений с веб-сервером (например, traefik/whoami). Убедиться в работоспособности. Провести нагрузочное тестирование, используя сетевой бенчмарк (например, Apache Benchmark, ab).

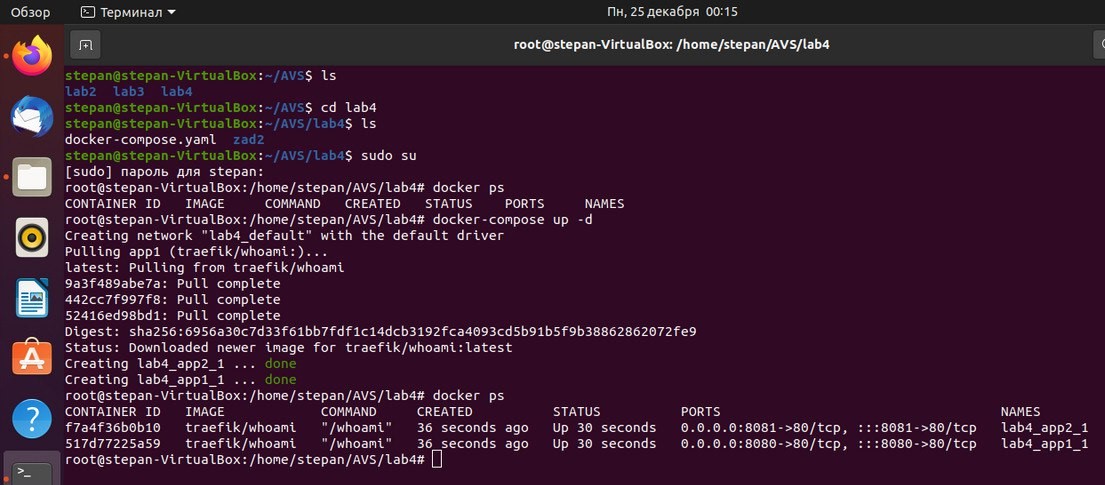
**2)** Добавить в docker-compose файл из п.1 прокси-сервер на базе Nginx, реализующий переадресацию запросов и отдачу контента от docker-приложений, при обращении по домену/IP адресу на порт 80 хоста. Убедиться в работоспособности. Провести нагрузочное тестирование.  
  
**3)** Увеличить число экземпляров docker-приложений с веб-сервером до максимально возможного. Убедиться в соответствии теоретического оценки практическим результатам. Продемонстрировать работу.

**Выполнение задания:**

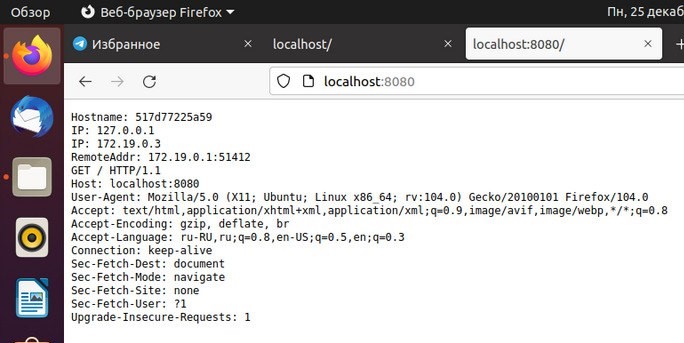
Задание 1:

С помощью строки “docker-compose up -d” мы запускаем все указанные сервисы в файле docker-compose.yml в фоновом режиме.

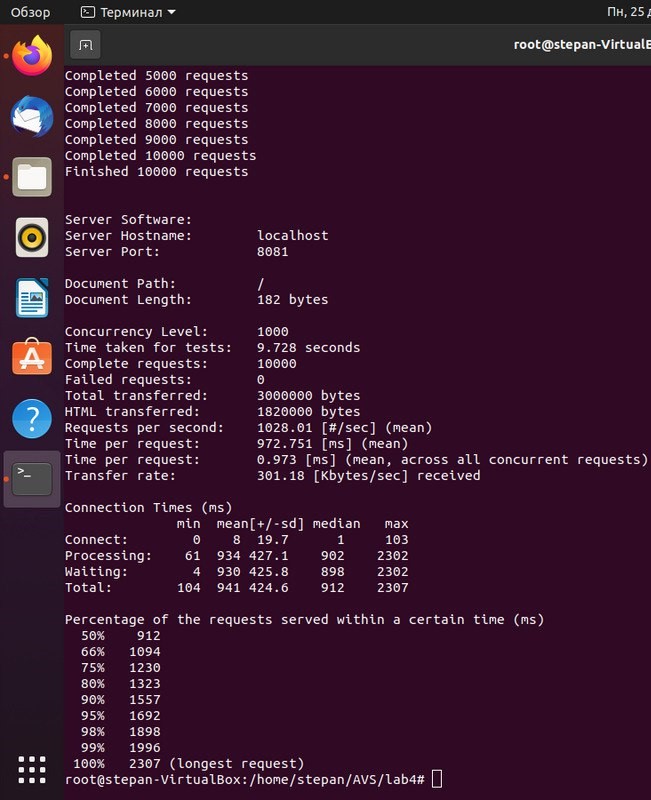
С помощью строки “docker ps” мы проверяем, что создались контейнеры.

****

Далее мы проверяем localhost по порту:



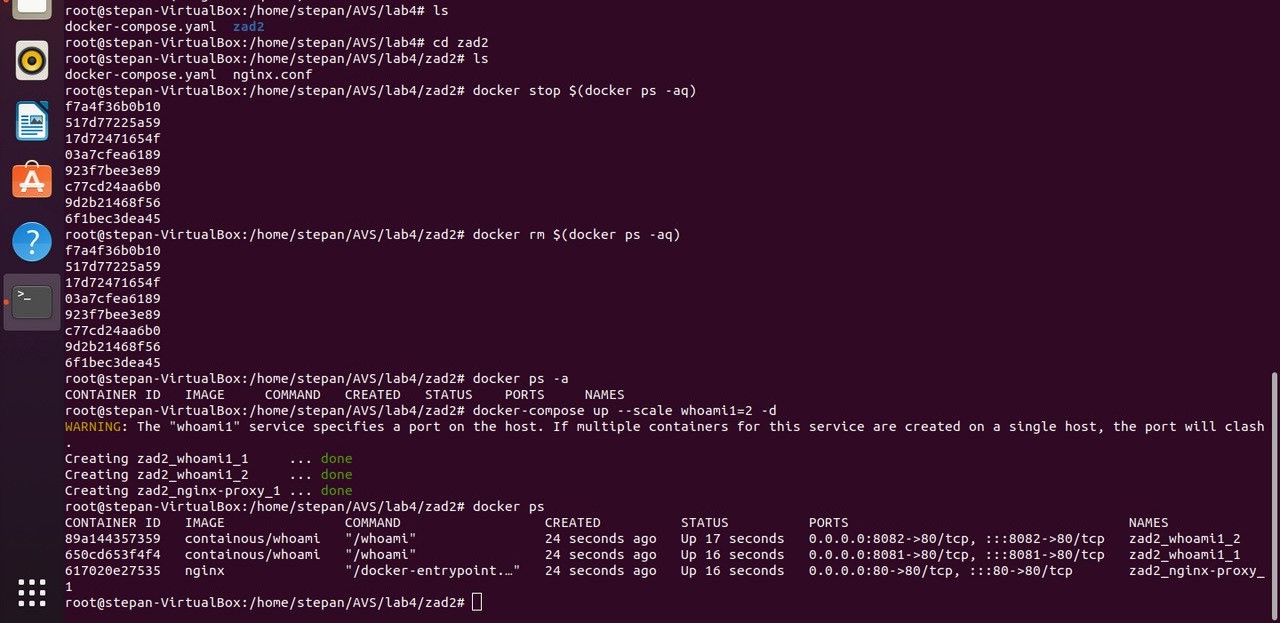
Запускаем бенчмарк по этому порту:



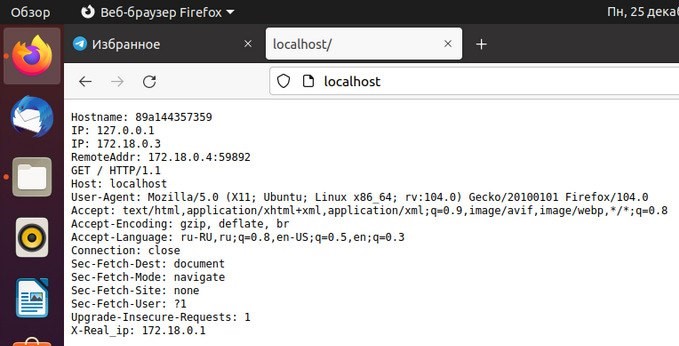
Задание 2:

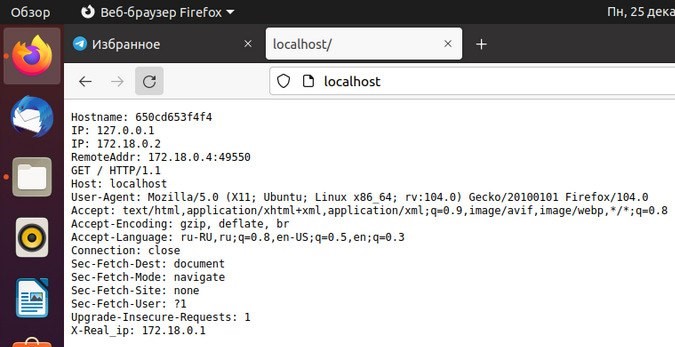
Останавливаем и удаляем все контейнеры.

С помощью команды “docker-compose up -- scale whoami1=2 -d” мы масштабируем контейнеры.

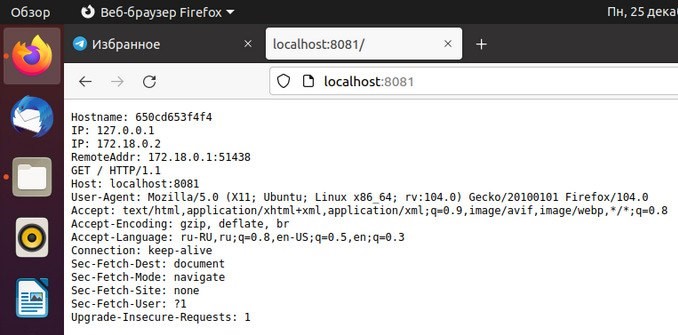


(при обновлении страницы значения меняются)

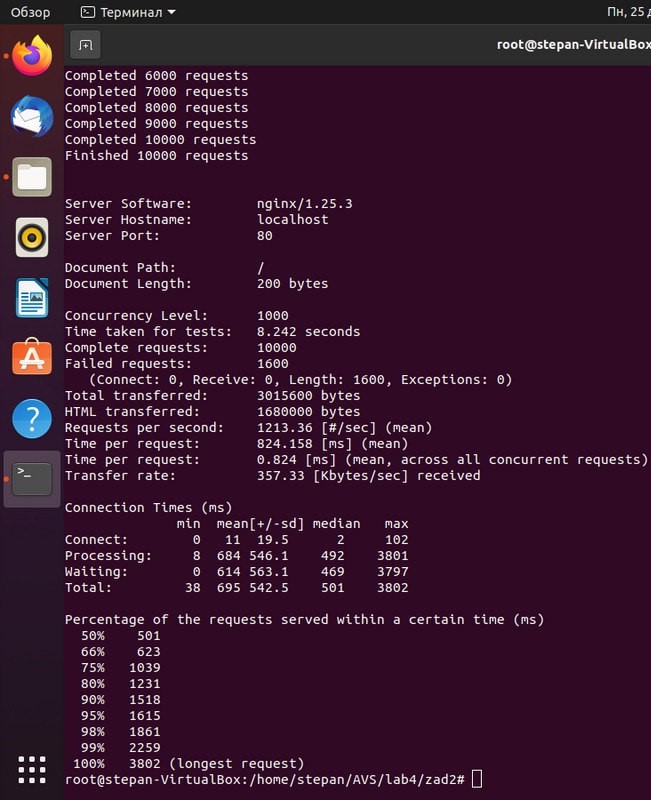




(с определенным портом)



Далее запускаем нагрузочное тестирование:

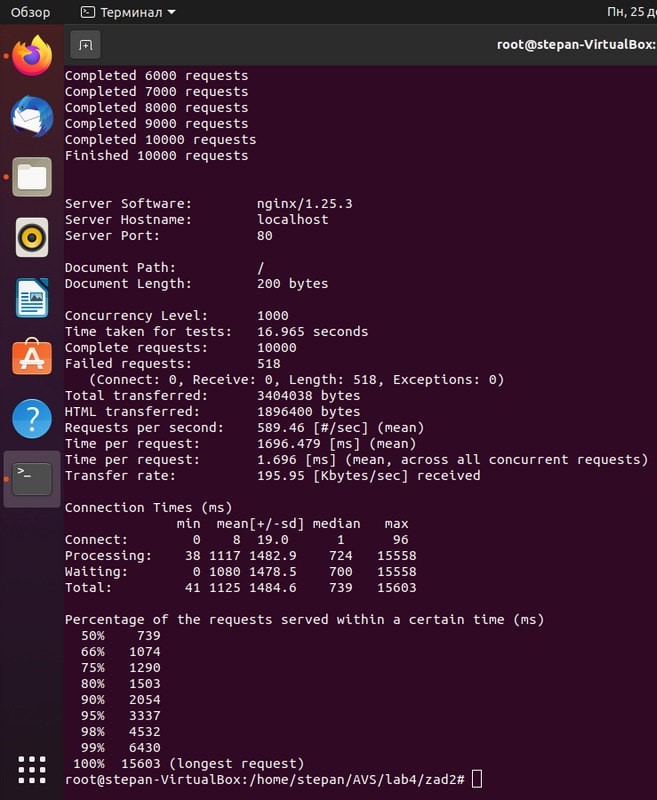


Задание 3:

Останавливаем и удаляем все контейнеры.

С помощью команды “docker-compose up -- scale whoami1=4 -d” мы масштабируем контейнеры (увеличиваем их количество в 2 раза).

Далее запускаем нагрузочное тестирование:



Вывод:

При увеличении количества контейнеров Requests per second уменьшается.

То есть мы убедились в соответствии теоретической оценки практическим результатам.