

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»
(СибГУТИ)

Кафедра вычислительных систем

Лабораторная работа № 4

Выполнил:

студент группы ИВ-121:

Ермаков А. В.

Работу проверил:

Романюта А.А.

Новосибирск 2023 г.

Содержание

Задание	3
Разработка файла docker-compose	4
Создание отдельного сервиса Nginx	11
Увеличение числа экземпляров docker-приложений	17

Задание

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.

Разработка файла docker-compose

Docker Compose по умолчанию устанавливается с Docker Desktop, поэтому создаем новую директорию для нашего проекта:

```
C:\Users\hasee>mkdir docker-projects  
  
C:\Users\hasee>cd docker-projects  
  
C:\Users\hasee\docker-projects>_
```

mkdir docker-projects

cd docker-projects

В этой директории создаем файл docker-compose.yaml, в который запишем содержимое:

version: '3'

services:

web1:

image: nginx

ports:

- "5000:80"

web2:

image: nginx

ports:

- "5001:80"

web3:

image: nginx

ports:

- "5002:80"

web4:

image: nginx

ports:

- "5003:80"

web5:

image: nginx

ports:

- "5004:80"

В этой же директории выполняем следующую команду:
`docker-compose up -d`

Эта команда создаст и запустит контейнеры на основе файла `docker-compose.yml`. Флаг `-d` означает, что контейнеры будут запущены в фоновом режиме.

Вот что происходит после ввода этой команды:

```
C:\Users\hasee\docker-projects>docker-compose up -d
[+] Running 3/10
 - whoami1 3 layers [0/0] 90.91kB/327.6kB Pulling 13.7s
   - 29015087d73b Downloading [=====>] 90.9... 8.6s
   - 0109a00d13bc Waiting 8.6s
   - d3caffff64d8 Waiting 8.6s
 - whoami2 Pulling 13.7s
 - traefik 4 layers [0/0] 8.35MB/39.11MB Pulling 13.7s
   [0] 96526aa774ef Pull complete 7.2s
   [0] eed05813ad6 Pull complete 2.9s
   - 86ea8083d7bc Downloading [=====>] 8.3... 8.6s
   [0] cea349f5f8a7 Download complete 5.2s
```

это создание рандомного контейнера.

Командная строка

```
Microsoft Windows [Version 10.0.19045.3693]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\hasee>mkdir docker-projects
Подпапка или файл docker-projects уже существует.



































C:\Users\hasee>cd docker-projects

C:\Users\hasee\docker-projects>docker-compose up -d
[+] Building 0.0s (0/0)
[+] Running 5/5
  Container docker-projects-web1-1   Started
  Container docker-projects-web5-1   Started
  Container docker-projects-web3-1   Started
  Container docker-projects-web4-1   Started
  Container docker-projects-web2-1   Started

C:\Users\hasee\docker-projects>
```

это уже нужные в лабе запущенные контейнеры.

По итогу глядим на наши контейнеры:

<input type="checkbox"/>	Name ↑	Image	Status	CPU (%)	Port(s)	Last started	Actions
<input type="checkbox"/>	 docker-projects		Running (5/5)	0%		2 minutes ago	  
<input type="checkbox"/>	 web1-1 d5bcef6b7847 	nginx	Running	0%	5000:80 	2 minutes ago	  
<input type="checkbox"/>	 web2-1 352cf70499ca 	nginx	Running	0%	5001:80 	2 minutes ago	  
<input type="checkbox"/>	 web3-1 642cbdd1a5f7 	nginx	Running	0%	5002:80 	2 minutes ago	  
<input type="checkbox"/>	 web4-1 11bffc20018b 	nginx	Running	0%	5003:80 	2 minutes ago	  
<input type="checkbox"/>	 web5-1 c32271a1c702 	nginx	Running	0%	5004:80 	2 minutes ago	  

Видим, что все 5 контейнеров успешно запустились.

Переходим по ссылке в браузер:



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

На других контейнерах аналогично.

Нагрузим наши контейнеры немножечко:

```
C:\Users\hasee\docker-projects>ab -n 1000 -c 10 http://localhost:5003/
This is ApacheBench, Version 2.3 <$Revision: 1903618 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests


Server Software:
Server Hostname:      localhost
Server Port:          5003


Document Path:        /
Document Length:      182 bytes


Concurrency Level:     10
Time taken for tests:  2.982 seconds
Complete requests:     1000
Failed requests:        0
Total transferred:     300000 bytes
HTML transferred:      182000 bytes
Requests per second:   335.31 [#/sec] (mean)
Time per request:      29.823 [ms] (mean)
Time per request:      2.982 [ms] (mean, across all concurrent requests)
Transfer rate:         98.24 [Kbytes/sec] received
```

Percentage of the requests served within a certain time (ms)

50%	33
66%	36
75%	38
80%	47
90%	68
95%	71
98%	78
99%	95
100%	135 (longest request)

И у каждого по 12 процессов:

```
2023-11-29 22:33:12 2023/11/29 15:33:12 [notice] 1#1: start worker processes
2023-11-29 22:33:12 2023/11/29 15:33:12 [notice] 1#1: start worker process 29
2023-11-29 22:33:12 2023/11/29 15:33:12 [notice] 1#1: start worker process 30
2023-11-29 22:33:12 2023/11/29 15:33:12 [notice] 1#1: start worker process 31
2023-11-29 22:33:12 2023/11/29 15:33:12 [notice] 1#1: start worker process 32
2023-11-29 22:33:12 2023/11/29 15:33:12 [notice] 1#1: start worker process 33
2023-11-29 22:33:12 2023/11/29 15:33:12 [notice] 1#1: start worker process 34
2023-11-29 22:33:12 2023/11/29 15:33:12 [notice] 1#1: start worker process 35
2023-11-29 22:33:12 2023/11/29 15:33:12 [notice] 1#1: start worker process 36
2023-11-29 22:33:12 2023/11/29 15:33:12 [notice] 1#1: start worker process 37
2023-11-29 22:33:12 2023/11/29 15:33:12 [notice] 1#1: start worker process 38
2023-11-29 22:33:12 2023/11/29 15:33:12 [notice] 1#1: start worker process 39
2023-11-29 22:33:12 2023/11/29 15:33:12 [notice] 1#1: start worker process 40
2023-11-29 22:33:17 2023/11/29 15:33:17 [notice] 1#1: signal 3 (SIGQUIT) received, shutting down
2023-11-29 22:33:17 2023/11/29 15:33:17 [notice] 29#29: gracefully shutting down
2023-11-29 22:33:17 2023/11/29 15:33:17 [notice] 31#31: gracefully shutting down
2023-11-29 22:33:17 2023/11/29 15:33:17 [notice] 30#30: gracefully shutting down
2023-11-29 22:33:17 2023/11/29 15:33:17 [notice] 32#32: gracefully shutting down
```

C:\Users\hasee\docker-projects>ab -n 30000 -c 13000 http://localhost/

This is ApacheBench, Version 2.3 <\$Revision: 1903618 \$>

Copyright 1996 Adam Twiss, Zeus Technology Ltd, <http://www.zeustech.net/>

Licensed to The Apache Software Foundation, <http://www.apache.org/>

Benchmarking localhost (be patient)

Completed 3000 requests

Completed 6000 requests

Completed 9000 requests

Completed 12000 requests

Test aborted after 10 failures

apr_socket_connect(): 1#1: юфьы■ўхэшх эх ёёСрэютыхэю, С.б. ьюэхўэ\щ
ьюбя№■СхЁ юСтхЁу чряЁюё эр яюфьы■ўхэшх. (730061)

Total of 13645 requests completed

C:\Users\hasee\docker-projects>

Отсюда видно, что 13645 процессов завершились успешно, что совпадает с теоретической оценкой.

Создание отдельного сервиса Nginx

Для реализации прокси-сервера на базе Nginx внутри Docker Compose, создадим отдельный сервис для Nginx и настроим его для обработки запросов и переадресации к вашим Docker-приложениям:

```
version: '3'
```

```
services:
```

```
  web1:
```

```
    image: nginx
```

```
    ports:
```

```
      - "5000:80"
```

```
  web2:
```

```
    image: nginx
```

```
    ports:
```

```
      - "5001:80"
```

```
  web3:
```

```
    image: nginx
```

```
    ports:
```

```
      - "5002:80"
```

```
  web4:
```

```
    image: nginx
```

```
    ports:
```

```
      - "5003:80"
```

```
  web5:
```

```
    image: nginx
```

ports:

- "5004:80"

nginx-proxy:

image: nginx

ports:

- "80:80"

volumes:

- ./nginx.conf:/etc/nginx/nginx.conf

depends_on:

- web1

- web2

- web3

- web4

- web5

Таким образом мы создаем отдельный сервис nginx-проху, который проксирует запросы с хоста на порт 80 ко всем контейнерам web1, web2, web3, web4 и web5.

Далее создаем файл nginx.conf в папке nginx (в той же директории, где находится ваш docker-compose.yml), чтобы обеспечить правильное перенаправление запросов к соответствующему сервису:

```
http {
```

```
    sendfile on;
```

```
    upstream backend {
```

```
        server web1:80;
```

```
        server web2:80;
```

```
        server web3:80;
```

```

server web4:80;

server web5:80;

}

server {

    listen 80;


    location / {

        proxy_pass http://backend;

        proxy_set_header Host $host;

        proxy_set_header X-Real-IP $remote_addr;

        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

        proxy_set_header X-Forwarded-Proto $scheme;

    }

}







```

```

C:\Users\hasee\docker-projects>docker network create nginx-proxy
debb06505d316e6e91af782006db9fde875c22086b4749c69757e0ba04893650

[+] Running 6/6 docker-projects>docker-compose up -d
 Container docker-projects-web4-1      Started      1.4s
 Container docker-projects-web5-1      Started      1.7s
 Container docker-projects-web2-1      Started      1.5s
 Container docker-projects-web1-1      Started      1.2s
 Container docker-projects-web3-1      Started      1.1s
 Container docker-projects-nginx-proxy-1 Started      0.0s
 Container docker-projects-web3-1      Started      1.1s
C:\Users\hasee\docker-projects>inx-proxy-1 Created      0.0s

```

<input type="checkbox"/>	<div><div></div><div><div>docker-projects</div></div></div>		Running (6/6)	0%	9 minutes ago	<div><div></div><div>:</div><div></div></div>	
<input type="checkbox"/>	<div><div></div><div><div>nginx-proxy-1</div><div>8dc73e23c8c</div></div></div>	nginx	Running	0%	80:80	9 minutes ago	<div><div></div><div>:</div><div></div></div>
<input type="checkbox"/>	<div><div></div><div><div>web1-1</div><div>ac4be01550b7</div></div></div>	nginx	Running	0%	5000:80	9 minutes ago	<div><div></div><div>:</div><div></div></div>
<input type="checkbox"/>	<div><div></div><div><div>web2-1</div><div>697844d963d2</div></div></div>	nginx	Running	0%	5001:80	9 minutes ago	<div><div></div><div>:</div><div></div></div>
<input type="checkbox"/>	<div><div></div><div><div>web3-1</div><div>3259764cc3d0</div></div></div>	nginx	Running	0%	5002:80	9 minutes ago	<div><div></div><div>:</div><div></div></div>
<input type="checkbox"/>	<div><div></div><div><div>web4-1</div><div>2fb2ece49602</div></div></div>	nginx	Running	0%	5003:80	9 minutes ago	<div><div></div><div>:</div><div></div></div>
<input type="checkbox"/>	<div><div></div><div><div>web5-1</div><div>514db819a68e</div></div></div>	nginx	Running	0%	5004:80	9 minutes ago	<div><div></div><div>:</div><div></div></div>

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Посмотрим количество процессов на каждом из контейнеров:

Docker Desktop

Update to latest

Q

Search for images, containers, volumes, extensions and more...

Ctrl+K

Containers

Images

Volumes

Dev Environments BETA

Docker Scout

Learning center

Extensions

Add Extensions

docker-projects-web2-1

<

nginx

352cf70499ca

5001:80

Logs

Inspect

Bind mounts

Exec

Files

Stats

2023-11-29 09:33:00 /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration

2023-11-29 09:33:00 /docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/

2023-11-29 09:33:00 /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh

2023-11-29 09:33:00 10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf

2023-11-29 09:33:00 10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf

2023-11-29 09:33:00 /docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh

2023-11-29 09:33:00 /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh

2023-11-29 09:33:00 /docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh

2023-11-29 09:33:00 /docker-entrypoint.sh: Configuration complete; ready for start up

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: using the "epoll" event method

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: nginx/1.25.3

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: OS: Linux 5.15.133.1-microsoft-standard-WSL2

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker processes

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker process 29

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker process 30

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker process 31

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker process 32

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker process 33

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker process 34

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker process 35

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker process 36

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker process 37

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker process 38

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker process 39

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker process 40

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker processes

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker process 29

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker process 30

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker process 31

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker process 32

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker process 33

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker process 34

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker process 35

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker process 36

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker process 37

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker process 38

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker process 39

2023-11-29 09:33:00 2023/11/29 02:33:00 [notice] 1#1: start worker process 40

Количество процессов на всех контейнерах 12, теперь в файле конфигурации nginx посмотрим на максимальное количество соединений для одного процесса:

```
events {
    worker_connections 1024;
}
```

Теперь у нас есть 5 экземпляров, 12 процессов на каждом и 1024 ограничение для одного процесса.

По итогу получаем $5 * 12 * 1024 = 61440$ запросов теоретически.

Увеличение числа экземпляров docker-приложений

Теоретически, если мы в nginx-проху имеем 4 работающих контейнера в 1 случае, 8 в другом, то при одинаковой нагрузке nginx-проху на 4 контейнерах будет работать в 2 раза дольше, чем на 8 процессах.

Поэтому для начала проведем тестирование на 8 контейнерах:

User

```
C:\Users\hasee\docker-projects>ab -n 100000 -c 10000 http://localhost/
```

```
This is ApacheBench, Version 2.3 <$Revision: 1903618 $>
```

```
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
```

```
Licensed to The Apache Software Foundation, http://www.apache.org/
```

Benchmarking localhost (be patient)

Completed 10000 requests

Completed 20000 requests

Completed 30000 requests

Completed 40000 requests

Completed 50000 requests

Completed 60000 requests

Completed 70000 requests

Completed 80000 requests

Completed 90000 requests

Completed 100000 requests

Finished 100000 requests

Server Software: Apache/2.4.58

Server Hostname: localhost

Server Port: 80

Document Path: /

Document Length: 45 bytes

Concurrency Level: 10000

Time taken for tests: 41.966 seconds

Complete requests: 100000

Failed requests: 0

Total transferred: 28900000 bytes

HTML transferred: 4500000 bytes

Requests per second: 2382.86 [#/sec] (mean)

Time per request: 4196.633 [ms] (mean)

Time per request: 0.420 [ms] (mean, across all concurrent requests)

Transfer rate: 672.51 [Kbytes/sec] received

Connection Times (ms)

	min	mean[+/-sd]	median	max
--	-----	-------------	--------	-----

Connect:	0	0 0.4	0	3
----------	---	-------	---	---

Processing:	1223	4017 616.0	4047	4783
-------------	------	------------	------	------

Waiting:	3	2058 1132.4	2019	4606
----------	---	-------------	------	------

Total:	1223	4017 616.0	4048	4783
--------	------	------------	------	------

Percentage of the requests served within a certain time (ms)

50%	4048
-----	------

66%	4389
-----	------

75%	4552
-----	------

80%	4594
-----	------

90%	4647
-----	------

95% 4702
98% 4760
99% 4774
100% 4783 (longest request)

C:\Users\hasee\docker-projects>

Время на 8 контейнерах составило 41.966 секунд.

Теперь проведем тестирование на 8 контейнерах:

C:\Users\hasee\docker-projects>ab -n 100000 -c 10000 http://localhost/

This is ApacheBench, Version 2.3 <\$Revision: 1903618 \$>

Copyright 1996 Adam Twiss, Zeus Technology Ltd, <http://www.zeustech.net/>

Licensed to The Apache Software Foundation, <http://www.apache.org/>

Benchmarking localhost (be patient)

Completed 10000 requests

Completed 20000 requests

Completed 30000 requests

Completed 40000 requests

Completed 50000 requests

Completed 60000 requests

Completed 70000 requests

Completed 80000 requests

Completed 90000 requests

Completed 100000 requests

Finished 100000 requests

Server Software: Apache/2.4.58

Server Hostname: localhost

Server Port: 80

Document Path: /

Document Length: 45 bytes

Concurrency Level: 10000

Time taken for tests: 87.067 seconds

Complete requests: 100000

Failed requests: 10

(Connect: 10, Receive: 0, Length: 0, Exceptions: 0)

Total transferred: 28900000 bytes

HTML transferred: 4500000 bytes

Requests per second: 1148.54 [#/sec] (mean)

Time per request: 8706.718 [ms] (mean)

Time per request: 0.871 [ms] (mean, across all concurrent requests)

Transfer rate: 324.15 [Kbytes/sec] received

Connection Times (ms)

	min	mean[+/-sd]	median	max
--	-----	-------------	--------	-----

Connect:	0	0 11.1	0	1535
----------	---	--------	---	------

Processing:	1234	8518 7581.4	4885	31058
-------------	------	-------------	------	-------

Waiting:	2	5248 7228.9	2693	30143
----------	---	-------------	------	-------

Total:	1235	8518 7581.6	4886	31059
--------	------	-------------	------	-------

Percentage of the requests served within a certain time (ms)

50%	4886
-----	------

66%	7078
75%	9428
80%	11568
90%	16569
95%	29194
98%	30397
99%	30516
100%	31059 (longest request)

Время на 4 контейнерах составило 87.067 секунд.

Несложно посчитать, что с увеличением числа контейнеров в 2 раза время уменьшилось в $87.067 / 41.966 = 2.0747$, что совпадает с значением, полученным теоретически.