

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
Высшего образования

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа 1 по Облачным и туманным вычислениям

Группа: Р3416

Выполнил:

Сиразетдинов А.Н.

Проверил:

Перл И. А.

Г. Санкт-Петербург

2026

Оглавление

Выполнение	3
Приложение	3
Инициализация swarm	4
Self-healing	5
Масштабирование	5
Вывод	6

Выполнение

Приложение

Напишем простое flask приложение

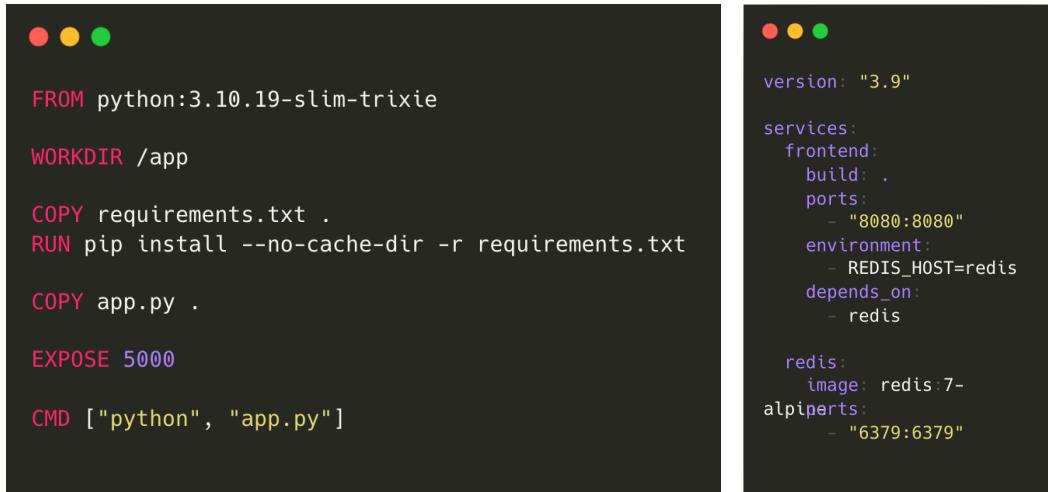
```
from flask import Flask
import redis
import os

app = Flask(__name__)

redis_host = os.getenv("REDIS_HOST", "localhost")
r = redis.Redis(host=redis_host, port=6379,
decode_responses=True)
@app.route("/")
def hello():
    count = r.incr("hits")
    return f"<h1>Количество посещений: {count}</h1>"

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=8080)
```

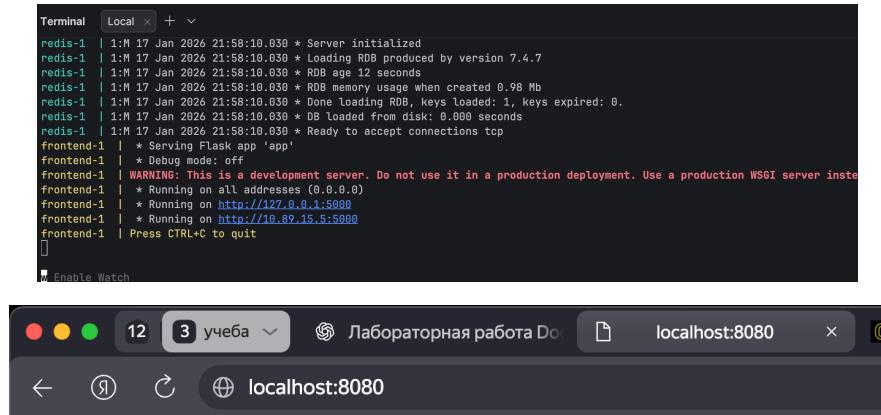
Dockerfile и docker-compose.yml



```
FROM python:3.10.19-slim-trixie
WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY app.py .
EXPOSE 5000
CMD ["python", "app.py"]
```

```
version: "3.9"
services:
  frontend:
    build: .
    ports:
      - "8080:8080"
    environment:
      - REDIS_HOST=redis
    depends_on:
      - redis
  redis:
    image: redis:7-
    alpineports:
      - "6379:6379"
```

Запустим командой docker compose up



```
redis-1 | 1:M 17 Jan 2026 21:58:10.030 * Server initialized
redis-1 | 1:M 17 Jan 2026 21:58:10.030 * Loading RDB produced by version 7.4.7
redis-1 | 1:M 17 Jan 2026 21:58:10.030 * ROB age 12 seconds
redis-1 | 1:M 17 Jan 2026 21:58:10.030 * ROB memory usage when created 0.98 Mb
redis-1 | 1:M 17 Jan 2026 21:58:10.030 * Done loading RDB, keys loaded: 1, keys expired: 0.
redis-1 | 1:M 17 Jan 2026 21:58:10.030 * DB loaded from disk: 0.000 seconds
redis-1 | 1:M 17 Jan 2026 21:58:10.030 * Ready to accept connections tcp
Frontend-1 | * Serving Flask app 'app'
Frontend-1 | * Debug mode: off
Frontend-1 | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead
Frontend-1 | * Running on all addresses (0.0.0.0)
Frontend-1 | * Running on http://127.0.0.1:5000
Frontend-1 | * Running on http://10.89.15.5:5000
Frontend-1 | Press CTRL+C to quit
[]
```

localhost:8080

Количество посещений: 6

Инициализация swarm

```
azat222@compute-vm-2-2-40(ssd-1759532797367:~/clouds$ sudo docker swarm init
Swarm initialized: current node (iole3bzruga8a7hooehjtkiaj) is now a manager.

To add a worker to this swarm, run the following command:

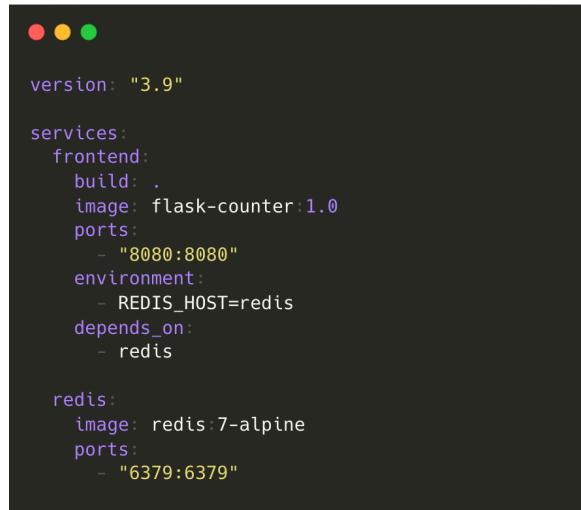
  docker swarm join --token SWMTKN-1-66htbwq186w17h10ddwxvj8oxsqlkimper09mk3p8ih8fk7v2i-8b8wk3uptin2u3f63uuudwdxj 10.128.0.12:2377

To add a manager to this swarm, run 'docker swarm join--token manager' and follow the instructions.
azat222@compute-vm-2-2-40(ssd-1759532797367:~/clouds$ docker node ls
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v
azat222@compute-vm-2-2-40(ssd-1759532797367:~/clouds$ sudo docker node ls
ID           HOSTNAME   STATUS  AVAILABILITY  MANAGER STATUS   ENGINE VERSION
iole3bzruga8a7hooehjtkiaj *  compute-vm-2-2-40(ssd-1759532797367  Ready   Active        Leader      28.5.2
azat222@compute-vm-2-2-40(ssd-1759532797367:~/clouds$
```

Соберем образ

```
sudo docker build -t flask-counter:1.0 .
```

Добавим образ в compose файл



```
version: "3.9"

services:
  frontend:
    build: .
    image: flask-counter:1.0
    ports:
      - "8080:8080"
    environment:
      - REDIS_HOST=redis
    depends_on:
      - redis

  redis:
    image: redis:7-alpine
    ports:
      - "6379:6379"
```

Запустим контейнеры

```
sudo docker stack deploy -c docker-compose.yml lab1
```

```
azat222@compute-vm-2-2-40(ssd-1759532797367:~/clouds$ sudo docker service ls
ID          NAME     MODE      REPLICAS  IMAGE
af983eweyhd  lab1_frontend  replicated  1/1      flask-counter:1.0  *:8080->8080/tcp
2vx2eyha0rgp  lab1_redis   replicated  1/1      redis:7-alpine    *:6379->6379/tcp
azat222@compute-vm-2-2-40(ssd-1759532797367:~/clouds$ sudo docker service ps lab1_frontend
ID          NAME     IMAGE          NODE          DESIRED STATE  CURRENT STATE          ERROR          PORTS
hhf73j0so245  lab1_frontend.1  flask-counter:1.0  compute-vm-2-2-40(ssd-1759532797367  Running        Running about a minute ago
azat222@compute-vm-2-2-40(ssd-1759532797367:~/clouds$
```

```
azat222@compute-vm-2-2-40(ssd-1759532797367:~/clouds$ curl 127.0.0.1:8080
<h1>Количество посещений: 1</h1>azat222@compute-vm-2-2-40(ssd-1759532797367:~/clouds$ curl 127.0.0.1:8080
<h1>Количество посещений: 2</h1>azat222@compute-vm-2-2-40(ssd-1759532797367:~/clouds$ curl 127.0.0.1:8080
```

Self-healing

```
azat222@compute-vm-2-2-40(ssd-1759532797367):~/clouds$ curl 127.0.0.1:8080
<h1>Конечество посетений: 4</h1>
azat222@compute-vm-2-2-40(ssd-1759532797367):~/clouds$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
fc282861d155 flask-counter:1.0 "python app.py" 6 minutes ago Up 6 minutes 5000/tcp lab1_frontend.1.5p6ooyx1ffvz7o1qhqlr3d7wr
c2fe6e1f14d redis:7-alpine "docker-entrypoint.s..." 6 minutes ago Up 6 minutes 6379/tcp lab1_redis.1.qwpjevkf7zqo7seq2a6ecf8kf

azat222@compute-vm-2-2-40(ssd-1759532797367):~/clouds$ sudo docker kill fc282861d155
fc282861d155
azat222@compute-vm-2-2-40(ssd-1759532797367):~/clouds$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
327dcfa37fe9 flask-counter:1.0 "python app.py" 4 seconds ago Up Less than a second 5000/tcp lab1_frontend.1.2wonwjygnhwuejxgw4uicvgju
cf0d0d44d0c7 redis:7-alpine "docker-entrypoint.s..." 7 minutes ago Up 7 minutes 6379/tcp lab1_redis.1.mojevkf7zqo7seq2a6ecf8kf

azat222@compute-vm-2-2-40(ssd-1759532797367):~/clouds$
```

Масштабирование

```
azat222@compute-vm-2-2-40(ssd-1759532797367):~/clouds$ sudo docker service scale lab1_frontend=5
lab1_frontend scaled to 5
overall progress: 5 out of 5 tasks
1/5: running [=====]
2/5: running [=====]
3/5: running [=====]
4/5: running [=====]
5/5: running [=====]
verify: Service lab1_frontend converged
azat222@compute-vm-2-2-40(ssd-1759532797367):~/clouds$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
069d76301b97 flask-counter:1.0 "python app.py" 17 seconds ago Up 17 seconds 5000/tcp lab1_frontend.5.vrim9ry27qepm2ly2ra2uuug86
c5281a9d4755 flask-counter:1.0 "python app.py" 17 seconds ago Up 17 seconds 5000/tcp lab1_frontend.4.4ajlqo4uzby7oy8c6b18tm7sa
522ae9d07cf4 flask-counter:1.0 "python app.py" 17 seconds ago Up 17 seconds 5000/tcp lab1_frontend.2.l120aeqe1w3eggu0dvarg7sh
c2feb6e1f14d flask-counter:1.0 "python app.py" 17 seconds ago Up 17 seconds 5000/tcp lab1_frontend.3.zkauq73j7km756097con3dsgd
327dcfa37fe9 flask-counter:1.0 "python app.py" 2 minutes ago Up 2 minutes 5000/tcp lab1_frontend.1.2wonwjygnhwuejxgw4uicvgju
cf0d0d44d0c7 redis:7-alpine "docker-entrypoint.s..." 9 minutes ago Up 9 minutes 6379/tcp lab1_redis.1.qwpjevkf7zqo7seq2a6ecf8kf
```

Вывод

В лабораторной работе я написал простое Flask приложение и познакомился с docker swarm