

Часть 1.

1. В виртуалке установлены пакеты

```
user@user:~$ sudo docker network create --driver bridge --subnet 172.16.254.0/24
lab05_network
f1e785413b414d6a6d07bd8280343dbda9f8861b2ffb0878c9740242ffff9922
```

- 2.

```
user@user:~$ sudo docker volume create lab05_volume
lab05_volume
```

```
user@user:~$ sudo docker volume inspect lab05_volume
```

```
[
```

```
    {
        "CreatedAt": "2025-10-23T12:40:54+07:00",
        "Driver": "local",
        "Labels": null,
        "Mountpoint": "/var/lib/docker/volumes/lab05_volume/_data",
        "Name": "lab05_volume",
        "Options": null,
        "Scope": "local"
    }
]
```

- 3.

```
user@user:~$ sudo cp -a lab4/ /var/lib/docker/volumes/lab05_volume/_data/
```

4. Dockerfile

```
FROM php:8.2-fpm

COPY init.sql /docker-entrypoint-initdb.d/
COPY supervisord.conf /etc/supervisor/conf.d/supervisord.conf

RUN apt-get update -y
RUN apt-get install -y mariadb-server
RUN chown -R mysql:mysql /var/lib/mysql
RUN mkdir /run/mysqld
RUN touch /run/mysqld/mysqld.sock
RUN chown -R mysql:mysql /run/mysqld
RUN docker-php-ext-install -j$(nproc) mysqli pdo_mysql
RUN apt-get install -y supervisor
RUN apt-get clean && rm -rf /var/lib/apt/lists/*
CMD ["/usr/bin/supervisord", "-n", "-c", "/etc/supervisor/conf.d/supervisord.conf"]
```

init.sql

```
create database if not exists library;
use library;

create table if not exists books (b_id INT PRIMARY KEY AUTO_INCREMENT, name VARCHAR(100) NOT NULL);

insert into books values(1,'name1');
insert into books values(2,'name2');
insert into books values(3,'name3');
insert into books values(4,'name4');

create table if not exists orders (o_id INT PRIMARY KEY AUTO_INCREMENT, b_id INT REFERENCES books(b_id), address VARCHAR(100) NOT NULL, status BOOLEAN);
create user 'n_user'@'localhost' identified by 'password';
grant all privileges on *.* to 'n_user'@'localhost';
flush privileges;
```

supervisord.conf

```
[supervisord]
nodaemon=true

[program:php-fpm]
command=php-fpm -F
autostart=true
autorestart=true
priority=10
stdout_logfile=/dev/stdout
stdout_logfile_maxbytes=0
stderr_logfile=/dev/stderr
stderr_logfile_maxbytes=0

[program:mariadb]
command=/usr/bin/mysqld_safe
autostart=true
autorestart=true
priority=20
user=mysql
stdout_logfile=/dev/stdout
stdout_logfile_maxbytes=0
stderr_logfile=/dev/stderr
stderr_logfile_maxbytes=0
```

```
user@user:~$ sudo docker build -t my-php-fpm .
user@user:~$ sudo cp -a lab4/ /var/lib/docker/volumes/lab05_volume/_data/
user@user:~$ sudo docker run -d --name php-fpm_app --network lab05_network --volume lab05_volum
e:/data --volume $(pwd)/php-conf/www.conf:/usr/local/etc/php-fpm.d/www.conf my-php-fpm
user@user:~$ sudo docker exec -it php-fpm_app bash
root@e2323d6a5be5:/var/www/html# mariadb < /docker-entrypoint-initdb.d/init.sql
```

5. Php-conf/www.conf

```
[www]
user = www-data
group = www-data
listen = 9000
listen.owner = www-data
listen.group = www-data
pm = dynamic
pm.max_children = 5
pm.start_servers = 2
pm.min_spare_servers = 1
pm.max_spare_servers = 3
chdir = /data/lab4
```

nginx-conf/nginx.conf

```

user nginx;
worker_processes auto;

error_log /var/log/nginx/error.log warn;
pid /var/run/nginx.pid;

events{
    worker_connections 1024;
}

http{
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                    '$status $body_bytes_sent "$http_referer"'
                    '"$http_user_agent" "$http_x_forwarded_for"';

    access_log /var/log/nginx/acess.log main;

    sendfile on;
    keepalive_timeout 65;
}

    include /etc/nginx/conf.d/*.conf;
}

```

nginx-conf/conf.d/default.conf

```

upstream application{
    server php-fpm_app:9000;
}

server {
    listen 80;
    server_name localhost;

    root /data/lab4;
    index add_main.php show_main.php;

    location / {
        try_files $uri $uri/ =404;
    }

    location ~ \.php$ {
        include fastcgi_params;
        fastcgi_pass application;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param PATH_INFO $fastcgi_path_info;
    }
}

```

```

user@user:~$ sudo docker run -d --name nginx_web --network lab05_network -p 80:80 --volume lab05_v
olume:/data --volume $(pwd)/nginx-conf/nginx.conf:/etc/nginx/nginx.conf --volume $(pwd)/nginx-conf
/conf.d/default.conf:/etc/nginx/conf.d/default.conf nginx:latest

```

6. Nginx-conf/conf.d/default.conf

```

upstream application{
    server php-fpm_app:9000;
}

fastcgi_pass application;

```

Часть 2.

1. Docker-compose.yml

```
version: '3.8'

services:
  volume_initializer:
    image: alpine
    container_name: volume_initializer
    volumes:
      - ./lab4:/source_app_data:ro
      - lab05_volume:/data
    networks:
      - lab05_network
    command: sh -c "cp -r /source_app_data/. /data/lab4/"

  nginx_web:
    container_name: nginx_web
    image: nginx:latest
    ports:
      - "80:80"
    volumes:
      - lab05_volume:/data
      - ./nginx-conf/nginx.conf:/etc/nginx/nginx.conf:ro
      - ./nginx-conf/conf.d/default.conf:/etc/nginx/conf.d/default.conf:ro
    networks:
      - lab05_network
    depends_on:
      - volume_initializer
      - php-fpm_app
    restart: unless-stopped

  php-fpm_app:
    container_name: php-fpm_app
    image: my-php-fpm
    volumes:
      - lab05_volume:/data
      - ./php-conf/www.conf:/user/local/etc/php-fpm.d/www.conf:ro
    networks:
      - lab05_network
    depends_on:
      - volume_initializer
    restart: unless-stopped

networks:
  lab05_network:
    driver: bridge
    ipam:
      config:
        - subnet: 172.16.254.0/24

volumes:
  lab05_volume:
    driver: local
```

```
user@user:~$ sudo apt install python3-pip
```

```
user@user:~$ sudo docker-compose up -d
Creating network "user_lab05_network" with driver "bridge"
Creating volume "user_lab05_volume" with local driver
Creating volume_initializer ... done
Creating php-fpm_app ... done
Creating nginx_web ... done
```

```
user@user:~$ sudo docker exec -it php-fpm_app bash
```

```
root@e2323d6a5be5:/var/www/html# mariadb < /docker-entrypoint-initdb.d/init.sql
```

```
user@user:~$ sudo docker-compose down -v
Stopping nginx_web ... done
Stopping php-fpm_app ... done
Removing nginx_web ... done
Removing php-fpm_app ... done
Removing volume_initializer ... done
Removing network user_lab05_network
Removing volume user_lab05_volume
```