

ПРАКТИЧЕСКАЯ РАБОТА

Развертывание мини-инфраструктуры сервисов в Docker Desktop

Nginx + FileBrowser Quantum + Uptime Kuma + Docker Compose

Направление: СА

Рабочая среда: Windows 11, Docker Desktop, PowerShell

Итог работы: на одном компьютере будет развернута небольшая контейнерная инфраструктура: веб-сайт, файловый сервис, система мониторинга и Nginx как единая точка входа.

2026

1. Назначение работы

Цель практической работы - закрепить основные понятия Docker на понятной системному администратору задаче. Вместо запуска одного отдельного контейнера будет собрана небольшая инфраструктура из нескольких готовых сервисов.

Каждый сервис выполняет самостоятельную функцию. Nginx обслуживает веб-сайт и позднее становится reverse proxy. FileBrowser Quantum предоставляет веб-доступ к файлам. Uptime Kuma контролирует доступность сервисов. Docker Compose описывает всю инфраструктуру одним конфигурационным файлом.

Что будет получено в результате

- статический веб-сайт, работающий в отдельном контейнере Nginx;
- FileBrowser Quantum с доступом к каталогу Windows через браузер;
- Uptime Kuma для контроля доступности сервисов;
- общая Docker-сеть, в которой контейнеры обращаются друг к другу по именам сервисов;
- Nginx reverse proxy как единая точка входа;
- локальные имена site.sa.test, files.sa.test и status.sa.test;
- единый compose.yaml, позволяющий поднять всю инфраструктуру одной командой.

2. Повторение основных понятий

Образ (image). шаблон, из которого Docker создает контейнер. Образ обычно содержит приложение и необходимые ему файлы.

Контейнер (container). изолированный запущенный экземпляр образа. Контейнер можно остановить, запустить заново, удалить и пересоздать.

Порт контейнера. порт, на котором приложение слушает внутри контейнера. Например, Nginx обычно слушает TCP-порт 80.

Опубликованный порт. соответствие между портом Windows и портом контейнера. Запись 8081:80 означает: Windows принимает соединение на 8081 и передает его на порт 80 контейнера.

Bind mount. подключение конкретного файла или каталога Windows внутрь контейнера. Изменения видны и на стороне Windows, и внутри контейнера.

Volume. хранилище данных, которым управляет Docker. Оно живет отдельно от конкретного контейнера и подходит для постоянных данных приложений.

Docker Compose. способ описать несколько сервисов, сети и хранилища в одном YAML-файле и управлять всем проектом командами docker compose.

Docker-сеть. виртуальная сеть, в которой контейнеры могут обмениваться трафиком. В одной Compose-сети сервисы находят друг друга по именам.

Reverse proxy. сервер-посредник, который принимает запрос клиента и перенаправляет его нужному внутреннему сервису.

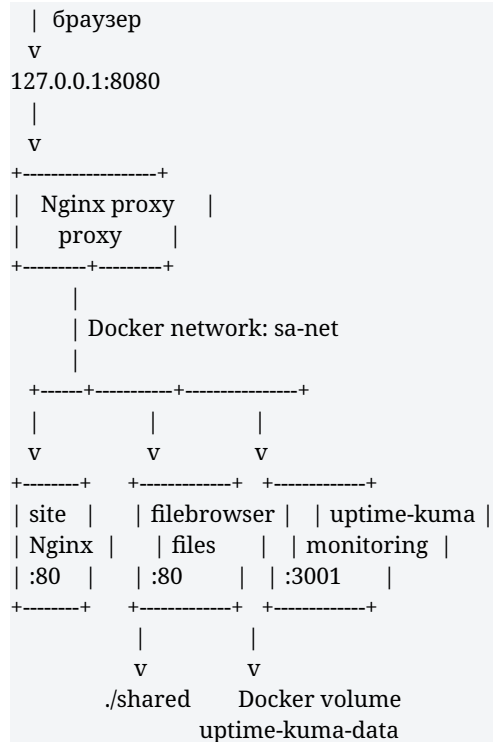
Мониторинг доступности. регулярная проверка того, отвечает ли сервис. Uptime Kuma выполняет такие проверки и показывает состояние UP или DOWN.

Важно: Внутри контейнера localhost означает сам этот контейнер. Для обращения к соседнему контейнеру в Docker Compose используется имя сервиса, например http://site:80.

3. Схема будущей инфраструктуры

Windows 11

|



На первом этапе сервисы будут временно доступны по отдельным портам Windows. Это позволяет проверить их по отдельности. На заключительном этапе прямые порты будут удалены, и наружу останется опубликован только reverse proxy.

4. Подготовка рабочего каталога

Откройте PowerShell и создайте отдельный каталог проекта.

```
cd $HOME
mkdir docker-sa-infra
cd docker-sa-infra

New-Item -ItemType Directory -Force -Path site, shared, filebrowser-data, proxy
```

В результате должна получиться следующая структура:

```
docker-sa-infra\
├── compose.yaml
├── site\
│   └── index.html
├── shared\
├── filebrowser-data\
│   └── config.yaml
└── proxy\
    └── default.conf
```

Пояснение: Каталоги site, shared, filebrowser-data и proxy находятся в файловой системе Windows. Docker Desktop может подключать их в Linux-контейнеры как bind mounts.

5. Сервис №1. Простой веб-сайт в Nginx

Первым сервисом будет обычный веб-сервер. Готовый официальный образ Nginx уже содержит работающий HTTP-сервер, поэтому устанавливать Nginx вручную в Windows не требуется.

Nginx. веб-сервер и reverse проху. В этой работе один контейнер Nginx будет отдавать HTML-страницу, а второй позднее будет перенаправлять запросы к другим сервисам.

5.1. Создание страницы

Создайте файл site\index.html и поместите в него следующий код:

```
<!doctype html>
<html lang="ru">
<head>
  <meta charset="utf-8">
  <title>Сервер CA</title>
  <style>
    body { font-family: Arial, sans-serif; max-width: 850px; margin: 50px auto; }
    .ok { color: #16823b; font-weight: bold; }
    a { display: block; margin: 10px 0; }
  </style>
</head>
<body>
  <h1>Мини-инфраструктура CA</h1>
  <p class="ok">Веб-сервер работает.</p>
  <p>Страница передается контейнером Nginx.</p>

  <h2>Сервисы</h2>
  <a href="http://files.sa.test:8080">FileBrowser Quantum</a>
  <a href="http://status.sa.test:8080">Uptime Kuma</a>
</body>
</html>
```

Позднее каталог site будет подключен к контейнеру в /usr/share/nginx/html. Nginx будет читать index.html непосредственно из каталога Windows.

6. Сервис №2. FileBrowser Quantum

FileBrowser Quantum предоставляет веб-интерфейс для работы с файлами. В данной работе он будет использовать каталог shared на компьютере Windows как файловое хранилище.

Постоянные данные. данные, которые должны сохраняться после остановки, удаления или пересоздания контейнера. Для них применяются volumes и bind mounts.

Каталог shared будет подключен как /folder. Отдельный каталог filebrowser-data будет хранить настройки и базу данных FileBrowser Quantum.

6.1. Конфигурация FileBrowser Quantum

Создайте файл filebrowser-data\config.yaml:

```
server:
  cacheDir: /home/filebrowser/data/tmp
sources:
  - path: /folder
    config:
      defaultEnabled: true
```

Повторение: Путь /folder существует с точки зрения контейнера. На стороне Windows ему соответствует каталог .\shared. Один и тот же каталог имеет разные пути снаружи и внутри контейнера.

Для лабораторной используется стабильная ветка FileBrowser Quantum v1.5.x с образом gtstef/filebrowser:stable. Документация проекта указывает, что для сохранения конфигурации и базы данных следует сохранять каталог /home/filebrowser/data.

7. Сервис №3. Uptime Kuma

Uptime Kuma - система мониторинга доступности. Она периодически отправляет запросы к указанным адресам и показывает, доступен сервис или нет.

UP. сервис ответил на проверку в соответствии с заданными условиями.

DOWN. проверка завершилась ошибкой: соединение не установлено, получен неподходящий ответ или превышено время ожидания.

Для Uptime Kuma будет применен Docker volume. Это позволит сохранить учетную запись, настройки и историю проверок при пересоздании контейнера.

Почему volume, а не папка Windows: Uptime Kuma хранит данные в SQLite и требует файловую систему с корректными блокировками файлов. Официальная документация рекомендует локальный Docker volume или локальный каталог и отдельно предупреждает о проблемах сетевых файловых систем.

8. Первый вариант compose.yaml

Теперь три сервиса будут описаны в одном Compose-проекте. На этом этапе каждому сервису временно публикуется отдельный порт, чтобы проверить его напрямую из Windows.

Создайте в корне проекта файл compose.yaml:

```
services:
  site:
    image: nginx:stable-alpine
    ports:
      - "8081:80"
    volumes:
      - ./site:/usr/share/nginx/html:ro
    restart: unless-stopped
    networks:
      - sa-net

  filebrowser:
    image: gtstef/filebrowser:stable
    ports:
      - "8082:80"
    volumes:
      - ./shared:/folder
      - ./filebrowser-data:/home/filebrowser/data
    restart: unless-stopped
    networks:
      - sa-net

  uptime-kuma:
    image: louislam/uptime-kuma:2
    ports:
      - "3001:3001"
    volumes:
      - uptime-kuma-data:/app/data
    restart: unless-stopped
    networks:
      - sa-net

networks:
  sa-net:
```

```
volumes:
  uptime-kuma-data:
```

8.1. Разбор файла

Элемент	Пример	Назначение
services	site, filebrowser, uptime-kuma	набор сервисов проекта
image	nginx:stable-alpine	образ, из которого создается контейнер
ports	8081:80	публикация порта контейнера на Windows
volumes	./site:/usr/share/nginx/html:ro	подключение данных или конфигурации
restart	unless-stopped	автоматический повторный запуск, пока сервис не остановлен вручную
networks	sa-net	общая виртуальная сеть контейнеров

8.2. Проверка конфигурации

```
docker compose config
```

Команда читает compose.yaml и выводит итоговую конфигурацию. Если YAML содержит ошибку в отступах или структуре, ее следует исправить до запуска.

8.3. Запуск

```
docker compose up -d
docker compose ps
```

Ожидается состояние running/Up для всех трех сервисов.

9. Проверка сервисов по отдельности

Откройте в браузере:

- <http://localhost:8081> - веб-сайт Nginx;
- <http://localhost:8082> - FileBrowser Quantum;
- <http://localhost:3001> - Uptime Kuma.

Если страница не открывается: сначала выполните `docker compose ps`, затем `docker compose logs <имя_сервиса>`. Не начинайте с удаления всего проекта: сначала определите, какой контейнер не работает и почему.

10. Работа с FileBrowser Quantum

Откройте <http://localhost:8082>. В стабильной ветке документация указывает начальные учетные данные `admin / admin`. После первого входа смените пароль администратора. Если текущая сборка показывает мастер первоначальной настройки, создайте учетную запись администратора через него.

Безопасность: Пароль `admin` нельзя оставлять для реального сервиса. В данной работе сервис доступен только на учебном компьютере.

В каталоге `shared` на стороне Windows создайте тестовый файл:

```
Set-Content -Path .\shared\from-windows.txt -Value "Файл создан в Windows"
```

Обновите FileBrowser. Файл должен появиться в веб-интерфейсе.

Затем загрузите через FileBrowser любой небольшой файл. Проверьте каталог `.\shared` в Проводнике Windows. Загруженный файл должен появиться там.

Что подтверждает этот опыт: контейнер не хранит эти файлы только внутри себя. Каталог Windows подключен внутрь контейнера через bind mount.

11. Первичная настройка Uptime Kuma

Откройте `http://localhost:3001` и создайте локальную учетную запись администратора.

Добавьте первый монитор:

- тип проверки: HTTP(s);
- имя: Site;
- URL: `http://site:80`;
- интервал можно оставить стандартным.

Добавьте второй монитор:

- тип проверки: HTTP(s);
- имя: FileBrowser;
- URL: `http://filebrowser:80`.

Почему используются site и filebrowser: Uptime Kuma работает внутри контейнера и находится в сети `sa-net`. Docker Compose регистрирует имена сервисов во внутреннем DNS. Адрес `localhost:8081` из Windows здесь не нужен: контейнер обращается напрямую к соседнему сервису по его имени и внутреннему порту.

12. Проверка мониторинга отказа

Остановите веб-сайт:

```
docker compose stop site
```

Через некоторое время монитор Site в Uptime Kuma должен перейти в состояние DOWN. Одновременно `http://localhost:8081` перестанет открываться.

Запустите сервис снова:

```
docker compose start site
```

После следующей успешной проверки статус должен вернуться в UP.

Смысл мониторинга: администратору не требуется вручную открывать каждую страницу каждые несколько минут. Система мониторинга делает проверки автоматически и фиксирует периоды недоступности.

13. Добавление Nginx Reverse Proxy

Сейчас у каждого сервиса свой порт: 8081, 8082 и 3001. Это удобно для проверки, но неудобно как постоянная схема. Добавим отдельный Nginx, который будет принимать запросы на одном порту 8080 и выбирать внутренний сервис по имени сайта.

server_name. имя сайта, на которое должен реагировать конкретный блок server в Nginx.

proxy_pass. директива Nginx, указывающая адрес внутреннего сервиса, которому нужно передать запрос.

Host header. HTTP-заголовок с именем сайта. Он позволяет одному IP-адресу и одному порту обслуживать несколько имен.

13.1. Локальные имена в Windows

Для лабораторной будут использованы зарезервированные тестовые имена. Все три имени должны указывать на локальный компьютер 127.0.0.1.

```
127.0.0.1 site.sa.test
127.0.0.1 files.sa.test
127.0.0.1 status.sa.test
```

Файл hosts находится по адресу C:\Windows\System32\drivers\etc\hosts. Для сохранения изменений обычно требуется запустить текстовый редактор от имени администратора.

Если изменить hosts нельзя: практически можно продолжить по отдельным localhost-портам. Reverse proxy проверяется командой curl с ручным заголовком Host, например: curl.exe -H "Host: site.sa.test" http://localhost:8080

13.2. Конфигурация proxy

Создайте файл proxy/default.conf:

```
server {
    listen 80 default_server;
    server_name _;
    return 404 "Unknown local service\n";
}

server {
    listen 80;
    server_name site.sa.test;

    location / {
        proxy_pass http://site:80;
        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;
    }
}

server {
    listen 80;
    server_name files.sa.test;

    location / {
        proxy_pass http://filebrowser:80;
        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;
    }
}

server {
    listen 80;
    server_name status.sa.test;

    location / {
        proxy_pass http://uptime-kuma:3001;
        proxy_http_version 1.1;

        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
    }
}
```

```

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;
}
}

```

Почему у Uptime Kuma есть дополнительные заголовки: Uptime Kuma использует WebSocket. Для работы через Nginx reverse проху официальная документация требует передавать Upgrade и Connection.

Почему используются отдельные имена, а не /status и /files: Uptime Kuma официально не поддерживает обычное размещение в подкаталоге вида /uptimekuma. Отдельные имена site.sa.test, files.sa.test и status.sa.test позволяют избежать этой проблемы.

13.3. Добавление проху в Compose

Добавьте сервис проху в раздел services файла compose.yaml:

```

proxy:
  image: nginx:stable-alpine
  ports:
    - "8080:80"
  volumes:
    - ./proxy/default.conf:/etc/nginx/conf.d/default.conf:ro
  depends_on:
    - site
    - filebrowser
    - uptime-kuma
  restart: unless-stopped
  networks:
    - sa-net

```

Проверьте конфигурацию и примените изменения:

```

docker compose config
docker compose up -d
docker compose ps

```

13.4. Проверка конфигурации Nginx

```
docker compose exec proxy nginx -t
```

При корректном файле должны появиться сообщения о successful / syntax is ok.

Администраторская привычка: перед перезагрузкой Nginx конфигурацию следует проверять командой nginx -t. Ошибка в одном символе может сделать reverse проху неработоспособным.

14. Проверка доступа через единую точку входа

При настроенном hosts откройте:

- http://site.sa.test:8080 - веб-сайт;
- http://files.sa.test:8080 - FileBrowser Quantum;
- http://status.sa.test:8080 - Uptime Kuma.

Во всех трех случаях Windows обращается к одному адресу 127.0.0.1 и одному порту 8080. Выбор сервиса выполняет Nginx по значению имени сайта.

```

Windows browser
|
| Host: files.sa.test

```

```

    v
localhost:8080
    |
    v
Nginx reverse proxy
    |
    | proxy_pass http://filebrowser:80
    v
FileBrowser Quantum

```

15. Финальный вариант: наружу открыт только reverse proxy

После проверки прямой доступ к внутренним сервисам больше не требуется. Удалите раздел ports у site, filebrowser и uptime-kuma. Порт должен остаться только у proxy.

Финальный compose.yaml:

```

services:
  site:
    image: nginx:stable-alpine
    volumes:
      - ./site:/usr/share/nginx/html:ro
    restart: unless-stopped
    networks:
      - sa-net

  filebrowser:
    image: gtstef/filebrowser:stable
    volumes:
      - ./shared:/folder
      - ./filebrowser-data:/home/filebrowser/data
    restart: unless-stopped
    networks:
      - sa-net

  uptime-kuma:
    image: louislam/uptime-kuma:2
    volumes:
      - uptime-kuma-data:/app/data
    restart: unless-stopped
    networks:
      - sa-net

  proxy:
    image: nginx:stable-alpine
    ports:
      - "8080:80"
    volumes:
      - ./proxy/default.conf:/etc/nginx/conf.d/default.conf:ro
    depends_on:
      - site
      - filebrowser
      - uptime-kuma
    restart: unless-stopped
    networks:
      - sa-net

networks:
  sa-net:

```

```
volumes:  
  uptime-kuma-data:
```

Примените:

```
docker compose up -d  
docker compose ps
```

Проверьте, что в колонке PORTS опубликованный порт есть только у proxy.

Повторение: Контейнеру не нужен опубликованный порт, чтобы общаться с другими контейнерами той же Docker-сети. Публикация нужна для доступа снаружи Docker-сети, например из браузера Windows.

16. Проверка постоянства данных

Убедитесь, что в FileBrowser есть тестовый файл и в Uptime Kuma настроены мониторы.

Остановите и удалите контейнеры проекта:

```
docker compose down
```

Затем создайте их заново:

```
docker compose up -d
```

Проверьте FileBrowser и Uptime Kuma. Файлы и настройки должны сохраниться. Причина: данные FileBrowser лежат в bind mounts, а данные Uptime Kuma - в отдельном Docker volume.

Не выполняйте `docker compose down -v` до проверки. Ключ `-v` удаляет volumes проекта. Для Uptime Kuma это означает удаление ее постоянного хранилища.

17. Диагностика инфраструктуры

Основные команды диагностики:

```
docker compose ps  
docker compose logs  
docker compose logs proxy  
docker compose logs filebrowser  
docker compose logs uptime-kuma
```

```
docker network ls  
docker volume ls
```

```
docker compose exec proxy nginx -t  
docker compose exec proxy getent hosts site  
docker compose exec proxy getent hosts filebrowser  
docker compose exec proxy getent hosts uptime-kuma
```

Команда `getent hosts` показывает, что Docker DNS может разрешить имя сервиса во внутренний IP-адрес контейнера.

DNS-разрешение. преобразование имени узла в IP-адрес. В Docker Compose эту задачу для имен сервисов выполняет встроенный DNS Docker.

18. Учебные неисправности

Выполните неисправности по одной. После каждой неисправности определите симптом, найдите причину, исправьте конфигурацию и подтвердите восстановление работы.

Действие	Причина	Ожидаемый симптом	Исправление
Остановить site	docker compose stop site	Сайт через проху недоступен, Uptime Kuma фиксирует DOWN.	docker compose start site
Изменить проху_pass для сайта на http://site:8081	ошибка внутреннего порта	Nginx возвращает 502 Bad Gateway.	вернуть http://site:80
Изменить filebrowser на filebrowserX	неверное DNS-имя	проху не может разрешить upstream.	вернуть имя filebrowser
Сделать синтаксическую ошибку в default.conf	например удалить ;	nginx -t сообщает ошибку.	исправить файл и повторить nginx -t

Правило диагностики: 502 от Nginx означает, что клиент дошел до reverse проху, но проху не смог корректно получить ответ от upstream. Это отличается от ситуации, когда сам порт 8080 вообще недоступен.

19. Самостоятельное расширение проекта

Выполните не менее трех пунктов:

1. Измените внешний вид index.html и добавьте на страницу ссылки на все сервисы.
2. Создайте второй контейнер Nginx site2 с отдельной HTML-страницей и подключите его к sa-net.
3. Добавьте имя site2.sa.test в hosts и отдельный server-блок в reverse проху.
4. Добавьте новый монитор site2 в Uptime Kuma.
5. Создайте в FileBrowser отдельную папку backup и загрузите в нее минимум два файла.
6. Остановите filebrowser, зафиксируйте состояние DOWN в Uptime Kuma, затем восстановите сервис.
7. С помощью docker compose logs найдите запись, связанную с запуском проху.
8. С помощью docker network inspect найдите контейнеры, подключенные к сети проекта.
9. Перезапустите Docker Desktop и после запуска проверьте состояние всего проекта.

20. Контрольные вопросы

1. Чем образ отличается от контейнера?
2. Что означает запись 8081:80?
3. Почему после удаления ports контейнеры продолжают общаться между собой?
4. Чем bind mount отличается от Docker volume?
5. Где физически находятся файлы, подключенные как ./shared:/folder?
6. Почему Uptime Kuma обращается к http://site:80, а не к http://localhost:8081?
7. Какую функцию выполняет Docker-сеть sa-net?
8. Что делает reverse проху?
9. Как Nginx выбирает между site.sa.test и files.sa.test?
10. Что делает проху_pass?
11. Почему для Uptime Kuma добавлены заголовки Upgrade и Connection?
12. Почему Uptime Kuma не размещена в подкаталоге /status?

13. Для чего используется файл hosts в Windows?
14. Почему перед применением конфигурации Nginx полезно выполнять nginx -t?
15. Что обычно означает ошибка 502 Bad Gateway?
16. Почему данные FileBrowser сохраняются после docker compose down?
17. Почему настройки Uptime Kuma сохраняются после пересоздания контейнера?
18. Что изменится при docker compose down -v?
19. Как определить, какие контейнеры сейчас работают?
20. Какая команда выводит журналы конкретного сервиса Compose?

21. Требования к отчету

Отчет должен содержать:

- тему и цель работы;
- схему итоговой инфраструктуры;
- финальный compose.yaml;
- финальный proxy/default.conf;
- скриншот docker compose ps;
- скриншот веб-сайта через site.sa.test:8080;
- скриншот FileBrowser Quantum через files.sa.test:8080;
- скриншот Uptime Kuma с мониторами Site и FileBrowser;
- скриншот одного состояния DOWN, полученного при намеренной остановке сервиса;
- краткое объяснение, где хранятся данные FileBrowser и Uptime Kuma;
- ответы на контрольные вопросы.

22. Завершение работы и очистка

Обычная остановка проекта без удаления постоянных данных:

```
docker compose down
```

Повторный запуск:

```
docker compose up -d
```

Полное удаление проекта вместе с Docker volume Uptime Kuma:

```
docker compose down -v
```

Осторожно: Команда down -v удаляет именованные volumes данного Compose-проекта. Каталоги bind mount в Windows автоматически не удаляются.

23. Итоговый конспект

Понятие	Краткое значение
Docker image	готовый шаблон приложения.
Container	запущенный экземпляр образа.
Port mapping	доступ из Windows к порту контейнера.
Bind mount	подключение конкретного каталога Windows внутрь контейнера.
Volume	постоянное хранилище под управлением Docker.
Compose	описание нескольких сервисов одной конфигурацией.
Service name	стабильное DNS-имя контейнерного сервиса внутри Compose-сети.
Reverse proxy	единая точка входа, перенаправляющая запросы к внутренним сервисам.
Uptime monitoring	автоматическая проверка доступности сервисов.
502 Bad Gateway	reverse proxy доступен, но не получил нормальный ответ от upstream.

24. Использованные источники

- Docker Docs. Networking in Compose: <https://docs.docker.com/compose/how-tos/networking/>
- Docker Docs. Bind mounts: <https://docs.docker.com/engine/storage/bind-mounts/>
- Docker Docs. Docker Desktop / WSL 2: <https://docs.docker.com/desktop/features/wsl/best-practices/>
- NGINX Documentation. ngx_http_proxy_module: https://nginx.org/en/docs/http/ngx_http_proxy_module.html
- NGINX Documentation. Reverse Proxy: <https://docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/>
- Uptime Kuma Wiki. How to Install: <https://github.com/louislam/uptime-kuma/wiki/%F0%9F%94%A7-How-to-Install>
- Uptime Kuma Wiki. Reverse Proxy: <https://github.com/louislam/uptime-kuma/wiki/Reverse-Proxy>
- FileBrowser Quantum Documentation. Docker v1.5.x (stable): <https://filebrowserquantum.com/en/docs/getting-started/docker-v1.5.x/>
- Docker Hub. Official Nginx image: https://hub.docker.com/_/nginx

Примечание к версиям: Методичка подготовлена по актуальной документации на сентябрь 2026 года. В ней используются Uptime Kuma 2, стабильный образ FileBrowser Quantum и официальный stable-alpine образ Nginx.