Docker — Volumes (???????? ??????)

Эта страница объясняет, как Docker хранит постоянные данные контейнеров с помощью **Volumes**. Здесь разобраны основные типы хранения данных, команды управления volumes, а также практические примеры резервного копирования и восстановления данных. Материал рассчитан на начинающих и помогает понять, **где Docker хранит данные и как не потерять их при пересоздании контейнеров**.

??? ????? Docker Volume

По умолчанию контейнеры Docker **не сохраняют данные после удаления**. Если контейнер удалить и создать заново, все файлы внутри него исчезнут.

Чтобы сохранить данные между перезапусками контейнеров, используются **Volumes**.

Volume — это специальное хранилище данных Docker, которое:

- существует независимо от контейнера
- сохраняет данные между перезапусками
- может использоваться несколькими контейнерами

?????? volumes ?????

Volumes используются для хранения **постоянных данных**, например:

- базы данных
- загруженные пользователями файлы
- кэш приложений
- логи

Пример:

```
mysql container
|
```

```
└── /var/lib/mysql  → volume
```

Если контейнер удалить и создать заново, данные в volume **сохранятся**.

??? Docker ?????? volumes

По умолчанию volumes находятся в системе Docker:

```
/var/lib/docker/volumes/
```

Каждый volume имеет свою директорию.

Пример структуры:

```
/var/lib/docker/volumes/mysql_data/_data
```

???? ???????? ?????? ? Docker

Docker поддерживает несколько способов хранения данных.

---

1. Docker Volume

Это основной и рекомендуемый способ хранения данных.

Пример в `docker-compose.yml`:

```
services:
  mysql:
    image: mysql:8
    volumes:
      - mysql_data:/var/lib/mysql

volumes:
```

```
  mysql_data:
```

Преимущества:

- безопасное хранение данных
- управляется Docker
- легко делать backup

---

2. Bind Mount

Bind mount подключает **обычную папку сервера** внутрь контейнера.

Пример:

```
services:
  app:
    volumes:
      - ./src:/var/www/html
```

Это означает:

```
сервер ./src  → контейнер /var/www/html
```

Используется для:

- разработки
- подключения исходного кода
- конфигурационных файлов

??????? ????? Volume ? Bind Mount

Тип	Описание
Volume	управляется Docker
Bind mount	обычная папка сервера

Обычно используется:

- **Volumes → данные приложений**
- **Bind mounts → код проекта**

???????? volumes

?????? volumes

```
docker volume ls
```

Пример вывода:

```
DRIVER    VOLUME NAME
local     site1_mysql_data
local     redis_cache
```

?????????? ? volume

```
docker volume inspect <volume_name>
```

Пример:

```
docker volume inspect site1_mysql_data
```

Результат покажет:

- путь к volume
- драйвер
- настройки

???????? volume

Создать volume можно вручную:

```
docker volume create mysql_data
```

После этого volume можно использовать в контейнерах.

???????? volumes

??????? ???? volume

```
docker volume rm <volume>
```

Пример:

```
docker volume rm mysql_data
```

??????? ????????????? volumes

```
docker volume prune -f
```

Команда удаляет volumes, которые **не используются контейнерами**.

> **Важно:** Перед удалением нужно убедиться, что volume не содержит важных данных.

????????????? volumes ? Docker Compose

Пример конфигурации:

```
services:
  mysql:
    image: mysql:8
    volumes:
      - mysql_data:/var/lib/mysql

volumes:
```

```
  mysql_data:
```

Это создаёт volume `mysql_data`, который будет хранить данные базы.

???????? volumes ??????????

Можно посмотреть volumes конкретного контейнера:

```
docker inspect <container>
```

Пример:

```
docker inspect site1_mysql
```

В выводе нужно найти блок:

```
Mounts
```

Он показывает подключённые volumes.

????????? ???????????? volume

Очень важная операция — backup данных.

Backup volume

```
docker run --rm \
  -v mysql_data:/data \
  -v $(pwd):/backup \
  alpine tar czf /backup/mysql_backup.tar.gz /data
```

Эта команда:

1. подключает volume
2. создаёт архив
3. сохраняет backup в текущей директории

?????????????? volume

Чтобы восстановить данные:

```
docker run --rm \
  -v mysql_data:/data \
  -v $(pwd):/backup \
  alpine tar xzf /backup/mysql_backup.tar.gz -C /
```

После этого данные снова появятся в volume.

???????? ?????? ??? ?????? ? volumes

?????? ?????? ????? ???????? ???????????

Если данные хранятся **внутри контейнера**, а не в volume, они будут потеряны.

Правильная практика:

```
volumes:
  - mysql_data:/var/lib/mysql
```

???????? volume ?????? ? ????????

Команда:

```
docker compose down -v
```

удаляет:

- контейнеры
- сети
- volumes

Это приведёт к **потере данных**.

?????????????? bind mount ??? ???? ??????

Не рекомендуется хранить базы данных через bind mount.

Лучше использовать:

```
volumes:
  - mysql_data:/var/lib/mysql
```

???????? volumes ???????

Показать volumes Compose проекта:

```
docker compose config --volumes
```

Это помогает понять, какие volumes использует проект.

??????? ???????

```
docker volume ls                    # список volumes
docker volume inspect <volume>     # информация о volume
docker volume create <volume>      # создать volume
docker volume rm <volume>          # удалить volume
```

```
docker volume prune -f             # удалить неиспользуемые volumes

docker inspect <container>         # посмотреть volumes контейнера
```

Revision #1
Created 2026-03-11 08:21:34 UTC by Crimson
Updated 2026-03-11 08:22:00 UTC by Crimson