Docker — ??????? ???????????

Эта страница объясняет основные команды Docker, которые используются для диагностики и отладки контейнеров. Здесь разобраны способы посмотреть запущенные контейнеры, проверить их состояние, изучить логи, подключиться внутрь контейнера и выполнить команды для анализа проблемы. Материал рассчитан на начинающих и помогает быстро понять, **что происходит внутри контейнера**, если приложение работает некорректно.

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

На практике команды из этой страницы используются в следующих ситуациях:

- контейнер **не запускается**
- сервис **падает после старта**
- приложение **возвращает ошибки**
- сервис **не отвечает**
- контейнер **постоянно перезапускается**
- нужно проверить **что происходит внутри контейнера**

Типичный порядок диагностики:

1. проверить список контейнеров
2. проверить статус контейнера
3. посмотреть логи
4. зайти внутрь контейнера
5. проверить процессы или выполнить команды

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

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

```
docker compose ps
```

Команда показывает контейнеры, которые относятся к текущему `docker-compose` проекту.

Пример вывода:

```
NAME                IMAGE          STATUS        PORTS
project_nginx       nginx:latest   Up 2 hours    0.0.0.0:80->80/tcp
project_php         php:8.2-fpm    Up 2 hours
```

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

- имя контейнера
- используемый образ
- статус контейнера
- проброшенные порты

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

Это первая команда, которую стоит выполнить, если нужно проверить:

- запущен ли сервис
- работает ли контейнер
- не упал ли он

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

```
docker ps
```

Эта команда показывает **все запущенные контейнеры в системе**, независимо от проекта.

Пример:

```
CONTAINER ID   IMAGE         STATUS         PORTS
ab12cd34       nginx         Up 2 hours     80/tcp
cd34ef56       redis         Up 3 hours     6379/tcp
```

??? ??????????? ?? `docker compose ps`

Команда	Показывает
`docker compose ps`	контейнеры текущего проекта
`docker ps`	все контейнеры на сервере

????????? ??? ??????????? (??????? ?????????????)

```
docker ps -a
```

Показывает:

- работающие контейнеры
- остановленные контейнеры
- контейнеры с ошибками

Пример статуса:

```
Exited (1) 5 minutes ago
```

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

Контейнер **запустился, но приложение внутри завершилось с ошибкой**.

В таком случае следующим шагом нужно посмотреть логи.

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

```
docker ps -a --format "table {{.ID}}\t{{.Names}}\t{{.Status}}\t{{.Ports}}"
```

Этот формат удобен, если нужно быстро увидеть:

- ID контейнера
- имя
- статус
- порты

Пример вывода:

```
CONTAINER ID   NAMES         STATUS          PORTS
3acb1234       project_php   Up 10 minutes
7d8e5678       project_nginx Up 10 minutes   80/tcp
```

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

Логи — это основной источник информации при отладке.

Именно здесь можно увидеть:

- ошибки приложения
- ошибки запуска
- проблемы подключения к базе
- ошибки конфигурации

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

```
docker compose logs <service>
```

Пример:

```
docker compose logs nginx
```

Вывод покажет весь лог контейнера.

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

```
docker compose logs -f <service>
```

Флаг `-f` означает **follow** — следить за логом в реальном времени.

Пример:

```
docker compose logs -f php
```

Это удобно, если:

- вы только что перезапустили сервис
- нужно увидеть новые ошибки

Выход из режима просмотра:

```
CTRL + C
```

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

```
docker compose logs --tail=100 <service>
```

Показывает только последние 100 строк.

Пример:

```
docker compose logs --tail=100 nginx
```

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

Если контейнер работает давно и лог очень большой.

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

```
docker compose logs --timestamps <service>
```

Пример:

```
docker compose logs --timestamps php
```

Вывод будет выглядеть примерно так:

```
2026-03-11T12:32:10 app started
2026-03-11T12:32:11 connected to database
```

Это удобно для анализа событий во времени.

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

Иногда для диагностики нужно **зайти внутрь контейнера**, чтобы:

- посмотреть файлы
- проверить конфигурацию
- выполнить команды
- проверить сетевые подключения

????????????? ????? shell

```
docker compose exec <service> sh
```

Пример:

```
docker compose exec php sh
```

После выполнения команды вы окажетесь внутри контейнера.

????????????? bash

Если в контейнере установлен `bash`, можно использовать:

```
docker compose exec <service> bash
```

Пример:

```
docker compose exec php bash
```

> **Важно:** Не во всех контейнерах есть `bash`. Многие минимальные образы используют только `sh`.

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

После входа можно выполнять обычные команды Linux:

```
ls
cd
cat
ps
top
```

Например:

```
ls /var/www/html
```

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

```
docker compose top <service>
```

Пример:

```
docker compose top php
```

Команда показывает список процессов внутри контейнера.

Пример вывода:

```
UID     PID     CMD
root    123     php-fpm
```

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

Полезно, если:

- приложение зависло
- нужно проверить, запущен ли процесс
- нужно убедиться, что сервис действительно работает

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

Можно выполнить команду **без входа в shell**.

Формат:

```
docker compose exec <service> <command>
```

Пример:

```
docker compose exec php php -v
```

Вывод:

```
PHP 8.2.4 (cli)
```

??? ???????

Проверить установленные пакеты:

```
docker compose exec php composer --version
```

Проверить Node.js:

```
docker compose exec node node -v
```

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

Иногда нужно забрать файл из контейнера.

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

```
docker cp container:/path/file .
```

Пример:

```
docker cp project_php:/var/log/php.log .
```

Файл будет скопирован в текущую папку.

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

```
docker cp file container:/path/
```

Пример:

```
docker cp config.php project_php:/var/www/config.php
```

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

Если сервис не работает, обычно делают следующее:

1 ????????? ??????????

```
docker compose ps
```

2 ????????? ????

```
docker compose logs --tail=100
```

3 ???? ?????? ?? ???????? — ????????????? ??????

```
docker compose exec php sh
```

4 ????????? ????????

```
docker compose top php
```

5 ????????? ?????????? ???????

Например:

```
curl localhost
```

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

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

В большинстве случаев ошибка уже записана в лог.

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

В Compose:

- **service** — описание контейнера в `docker-compose.yml`
- **container** — фактически запущенный экземпляр

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

Если статус:

```
Exited
```

контейнер уже остановился, и нужно смотреть логи.

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

```
docker compose ps                          # контейнеры текущего проекта
docker ps                                  # все запущенные контейнеры
docker ps -a                               # все контейнеры включая остановленные

docker compose logs <service>              # логи сервиса
docker compose logs -f <service>           # логи в реальном времени
docker compose logs --tail=100 <service>   # последние строки логов
docker compose logs --timestamps <service> # логи с временными метками

docker compose exec <service> sh           # войти в контейнер
docker compose exec <service> bash         # войти через bash
docker compose exec <service> <command>    # выполнить команду

docker compose top <service>               # процессы контейнера

docker cp container:/path/file .           # копировать из контейнера
docker cp file container:/path/            # копировать в контейнер
```

Revision #3
Created 2026-03-11 07:22:44 UTC by Crimson
Updated 2026-03-11 07:55:05 UTC by Crimson