

Docker — ?????????? ? ???????????????????? ????????????

Эта страница объясняет, как отслеживать использование ресурсов Docker и как управлять дисковым пространством, занятым контейнерами, образами и volumes. Здесь разобраны команды для просмотра нагрузки на CPU и память, анализа использования диска, а также безопасной очистки неиспользуемых ресурсов. Материал ориентирован на начинающих и помогает понять, **почему Docker может занимать много места на диске или ресурсов системы** и как это контролировать.

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

Docker активно использует ресурсы системы:

- CPU
- оперативную память (RAM)
- дисковое пространство
- сетевые ресурсы

Со временем на сервере могут накапливаться:

- старые контейнеры
- неиспользуемые образы
- volumes с данными
- временные слои сборки

Если их не очищать, Docker может занять **десятки или даже сотни гигабайт дискового пространства**.

Поэтому регулярный мониторинг и очистка — важная часть администрирования Docker.

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

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

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

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

```
docker stats
```

Эта команда показывает статистику всех работающих контейнеров.

Пример вывода:

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O
a12bc34d	project_php	2.5%	120MiB / 2GiB	5.8%	1.2MB / 500KB
b45de67f	project_nginx	0.2%	15MiB / 2GiB	0.7%	200KB / 100KB

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

Поле	Описание
CPU %	загрузка процессора контейнером
MEM USAGE	используемая память
MEM %	процент от доступной памяти
NET I/O	сетевой трафик

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

Полезно, если:

- сервер начал тормозить
- приложение потребляет слишком много ресурсов
- нужно найти контейнер с высокой нагрузкой



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

```
docker stats --no-stream <container>
```

Флаг `--no-stream` выводит статистику **один раз**, без постоянного обновления.

Пример:

```
docker stats --no-stream project_php
```

Это удобно, если нужно быстро проверить текущую нагрузку.

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

Иногда нагрузку создаёт конкретный процесс.

Можно посмотреть процессы внутри контейнера:

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

Пример:

```
docker compose top php
```

Вывод покажет процессы, которые выполняются внутри контейнера.

Это помогает определить, что именно создаёт нагрузку:

- веб-сервер
 - PHP-процесс
 - worker
 - cron-задача
-

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

????? Docker

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

Docker

```
docker system df
```

Пример вывода:

TYPE	TOTAL	ACTIVE	SIZE
Images	12	4	3.2GB
Containers	8	4	120MB
Local Volumes	6	5	5.6GB
Build Cache	-	-	2.1GB

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

Тип	Описание
Images	Docker-образы
Containers	контейнеры
Volumes	постоянные данные
Build Cache	кэш сборки Docker

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

Эта команда помогает понять:

- почему Docker занимает много места
- какие ресурсы занимают больше всего диска



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

```
docker system df -v
```

Флаг `-v` (verbose) показывает более подробную информацию:

- какие именно образы занимают место
- какие контейнеры используют volumes
- какие слои можно удалить

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

Со временем Docker оставляет множество временных объектов:

- остановленные контейнеры
- старые образы
- временные сети
- build cache

Их можно безопасно удалить.

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

```
docker image prune -f
```

Удаляет **dangling images** — образы, которые больше не используются контейнерами.

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

Если после сборки осталось много старых образов.

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

```
docker system prune -f
```

Удаляет:

- остановленные контейнеры
- неиспользуемые сети
- dangling images
- build cache

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

Эта команда **не удаляет volumes**.

То есть:

- базы данных
- файлы проекта
- persistent storage

останутся на месте.

????????? volumes

```
docker volume prune -f
```

Удаляет **неиспользуемые volumes**.

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

Volumes могут содержать:

- базы данных
- пользовательские файлы
- кэш приложения

Поэтому перед удалением нужно убедиться, что volume действительно не используется.

Некоторые контейнеры могут иметь ограничения по памяти.

Проверить это можно так:

```
docker inspect <container> | grep -i memory
```

Пример:

```
docker inspect project_php | grep -i memory
```

Эта команда покажет:

- лимит памяти
- настройки использования RAM

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

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

Самая частая причина — накопившиеся:

- старые образы
- build cache
- остановленные контейнеры

Решение:

```
docker system prune -f
```

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

Проверить использование:

```
docker system df
```

Затем очистить ненужные данные.

????????? ?????????????? ??????????
?????? CPU

Найти контейнер:

```
docker stats
```

После этого:

- проверить логи
 - посмотреть процессы внутри контейнера
-

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

Возможные причины:

- утечка памяти
- неправильная конфигурация приложения
- слишком большие кэши

Проверить:

```
docker stats
```

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

На серверах разработки полезно периодически выполнять:

```
docker system prune -f
```

Если используется активная сборка образов:

```
docker builder prune -f
```

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

Если сервер начал тормозить:

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

```
docker stats
```

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

```
docker system df
```

3 ?????????? ?????????? ????????

```
docker system prune -f
```

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

```
docker system df
```

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

```
docker stats                                # нагрузка контейнеров в реальном времени
docker stats --no-stream <container>      # разовая статистика

docker compose top <service>              # процессы внутри контейнера

docker system df                            # использование диска Docker
docker system df -v                        # подробная информация

docker image prune -f                      # удалить неиспользуемые образы
docker system prune -f                    # очистка контейнеров, сетей и cache
docker volume prune -f                    # удалить неиспользуемые volumes
docker network prune -f                   # удалить неиспользуемые сети

docker info | grep -i memory              # информация о памяти
docker info | grep -i cpu                 # информация о CPU
docker inspect <container> | grep -i memory # лимиты контейнера
```

Revision #1

Created 2026-03-11 07:56:46 UTC by Crimson

Updated 2026-03-11 08:02:00 UTC by Crimson