

Docker — Images (???????????? ?????????)

Эта страница объясняет, что такое Docker-образы (Images), как они создаются, хранятся и управляются. Здесь разобраны основные команды для просмотра, загрузки, удаления и очистки образов. Материал рассчитан на начинающих и помогает понять, **как работают Docker-образы и как управлять ими на сервере.**

??? ????? Docker Image

Docker Image — это **шаблон контейнера**, из которого создаются запущенные контейнеры.

Проще говоря:

```
Docker Image → контейнер
```

Пример:

```
nginx:1.25 → контейнер nginx  
mysql:8 → контейнер mysql  
node:18 → контейнер node
```

Image содержит:

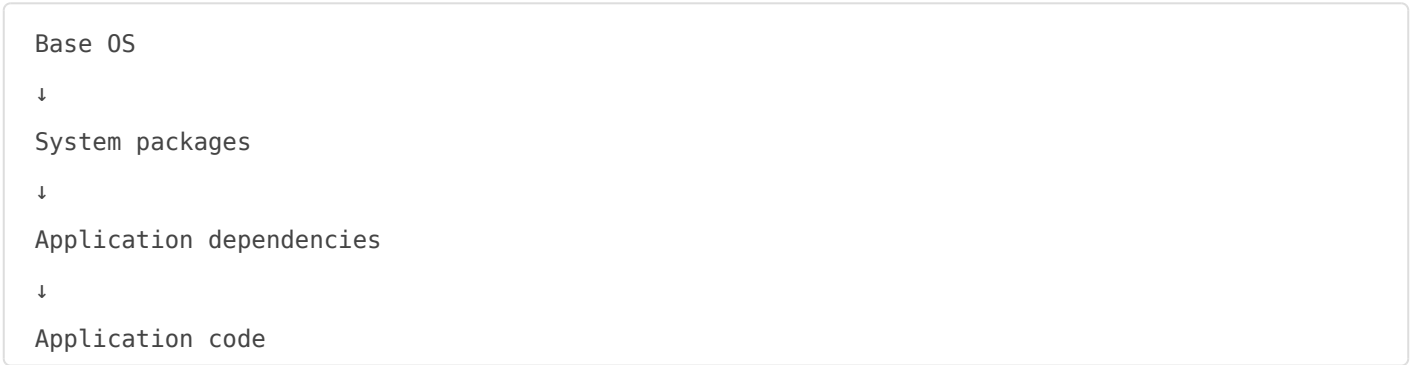
- операционную систему
- зависимости
- приложение
- конфигурацию запуска

Контейнер создаётся на основе образа и запускается как отдельный процесс.

??? ?????????? Docker Images

Docker Images состоят из **слоёв (layers)**.

Пример:



Каждый слой кэшируется. Это позволяет Docker **ускорять сборку образов**.

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

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

```
docker images
```

Пример вывода:

REPOSITORY	TAG	IMAGE ID	SIZE
nginx	1.25	a1b2c3d4e5	142MB
mysql	8	f6g7h8i9j0	520MB
node	18	k1l2m3n4o5	1.2GB

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

Поле	Описание
REPOSITORY	имя образа
TAG	версия образа
IMAGE ID	уникальный ID
SIZE	размер образа

????????? ??????? ?? registry

Docker может скачивать готовые образы из Docker Hub или другого registry.

```
docker pull <image>
```

Пример:

```
docker pull nginx:1.25
```

После этого образ появится в списке `docker images`.

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

Docker использует **теги** для обозначения версии образа.

Пример:

```
nginx:latest  
nginx:1.25  
node:18
```

?????? ?? ?????? ?????????????????? latest

Тег `latest` может измениться в любой момент.

Лучше использовать фиксированную версию:

```
nginx:1.25
```

Это делает окружение более предсказуемым.

????????? Docker Images

- сети
- build cache

“ **Важно:** Использовать эту команду нужно осторожно, особенно на production.

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

Образы можно создавать самостоятельно через `Dockerfile`.

Пример сборки:

```
docker build -t my_app .
```

Где:

Параметр	Значение
-t	тег образа
my_app	имя образа
.	текущая директория

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

Можно посмотреть, из каких слоёв состоит образ:

```
docker history <image>
```

Пример:

```
docker history nginx
```

Это помогает понять:

- какие команды выполнялись при сборке
- размер каждого слоя


```
docker search <image>          # поиск образа

docker rmi <image>             # удалить образ
docker image prune -f          # удалить неиспользуемые образы

docker build -t <name> .        # собрать образ
docker history <image>         # история слоев образа
```

Revision #1

Created 2026-03-11 08:24:11 UTC by Crimson

Updated 2026-03-11 08:28:06 UTC by Crimson