

Как удалить образы, контейнеры и тома Docker



Павел Соловьёв
Full Stack Developer

[#администрирование](#)

01 марта, 2017

👁 374

💬 0

Вступление

[Docker](#) позволяет с легкостью упаковать ваши приложения и службы в контейнеры, благодаря чему вы можете запускать их на любой платформе. Однако, по мере работы с Docker у вас запросто может накопиться избыточное количество ненужных образов, контейнеров и томов, которые наводят бардак в выходных данных и загромождают собой место на диске.

Хотя в Docker и не предусмотрено непосредственных прямых команд для удаления ненужных образов, контейнеров и томов, данная программа предлагает пользователям все необходимые инструменты для очистки системы через командную строку. Эта небольшая шпаргалка вкратце познакомит вас с полезными командами для освобождения места на диске и поддержания вашей системы в опрятном и организованном виде за счет удаления неиспользующихся образов, контейнеров и томов Docker.

Как использовать настоящее руководство:

- Это руководство составлено в формате краткой шпаргалки, в которой приведены отрывки команд, которые необходимо вводить в командную строку.
- Необязательно читать руководство по порядку, сразу переходите к разделу, который актуален для решения вашей текущей задачи.

Синтаксис подстановки команд, `command $(command)`, который используется в командах, доступен во многих популярных оболочках, включая **bash**, **zsh** и **Windows Powershell**.

Удаление образов Docker

Удаление одного или нескольких конкретных образов

Используйте команду `docker images command` со флажком `-a`, чтобы найти ID образов, которые вы желаете удалить. Эта команда покажет вам каждый образ, включая промежуточные слои образа. После того как вы обнаружили нужный образ, который вы хотите удалить, вы можете включить их ID или тэг в команду `docker rmi`:

Список:

```
docker images -a
```

Удаление:

```
docker rmi Image Image
```

Удаление подвешенных образов

Образы Docker состоят из нескольких слоев. Подвешенные образы – это слои, которые не имеют связей с другими маркированными образами. Они больше не служат какой-либо цели и просто занимают место на диске. Их можно найти, добавив флажок фильтра `-f` со значением `dangling=true` к команде `docker images`. Если вы уверены, что желаете удалить подобные образы, вы можете добавить флажок `-q`, после чего ввести их ID после команды `docker rmi`:

Примечание: Если вы создали образ и не маркировали его тэгом, такой образ автоматически отобразится в перечне подвешенных образов, поскольку у него отсутствуют связи с маркированными образами. Вы можете избежать этой ситуации, присвоив образу тэг при создании, либо присвоить образу тэг позже при помощи команды `docker tag`.

Список:

```
docker images -f dangling=true
```

Удаление:

```
docker rmi $(docker images -f dangling=true -q)
```

Удаление образов согласно конкретной схеме

Вы можете найти образы, которые соответствуют схеме, при помощи комбинации команд `docker images` и `grep`. После того как вы найдете все нужные образы, вы можете использовать команду `awk`, чтобы ввести ID соответствующих образов в `docker rmi`. Обратите внимание, что данные утилиты не входят в стандартный комплект поставки платформы Docker и не обязательно доступны в системе:

Список:

```
docker ps -a | grep "pattern"
```

Удаление:

```
docker images | grep "pattern" | awk '{print $1}' | xargs docker rmi
```

Удаление всех образов

Все образы Docker в системе можно выбрать, добавив флажок `-a` к команде `docker images`. Если вы уверены, что желаете удалить все образы, добавьте флажок `-q`, чтобы указать ID всех образов в команде `docker rmi`:

Список:

```
docker images -a
```

Удаление:

```
docker rmi $(docker images -a -q)
```

Удаление контейнеров

Удаление одного или нескольких конкретных контейнеров

Используйте команду `docker ps` с флажком `-a`, чтобы найти имена или ID контейнеров, которые вы желаете удалить:

Список:

```
docker ps -a
```

Удаление:

```
docker rm ID_or_Name ID_or_Name
```

Удаление контейнера при выходе

Если создавая контейнер, вы уже заранее знаете, что он больше не понадобится вам после того, как вы закончите работу, вы можете запустить команду `docker run --rm` для того, чтобы автоматически удалить контейнер после завершения работы с ним.

Запуск и удаление:

```
docker run --rm image_name
```

Удаление всех закрытых контейнеров

Вы можете найти контейнеры при помощи команды `docker ps -a` и затем отфильтровать их по статусу: созданные, перезапущенные, работающие, приостановленные или закрытые. Для того чтобы просмотреть список закрытых контейнеров, используйте флажок `-f`, чтобы включить фильтр по статусу. Если вы уверены, что хотите удалить указанные контейнеры, используйте `-q`, чтобы указать ID соответствующих контейнеров в команде `docker rm`.

Список:

```
docker ps -a -f status=exited
```

Удаление:

```
docker rm $(docker ps -a -f status=exited -q)
```

Удаление контейнеров при помощи более одного фильтра

Фильтры Docker можно комбинировать, повторяя флажок фильтра с дополнительным значением. В результате этого будет показан список контейнеров, который соответствует одному из условий. К примеру, если вы желаете удалить все контейнеры, помеченные как «Созданные» (состояние, которое может возникать, если вы запускаете контейнер с некорректной командой), так и «Закрытые», вам необходимо использовать сразу два фильтра:

Список:

```
docker ps -a -f status=exited -f status=created
```

Удаление:

```
docker rm $(docker ps -a -f status=exited -f status=created -q)
```

Удаление контейнеров согласно конкретной схеме

Вы можете найти все контейнеры, которые соответствуют схеме, при помощи комбинации команд `docker ps` и `grep`. После того как вы найдете все нужные контейнеры, вы можете использовать команду `awk`, чтобы ввести ID соответствующих контейнеров в `docker rm`. Обратите внимание, что данные утилиты не входят в стандартный комплект поставки платформы Docker и не обязательно доступны в системе:

Список:

```
docker ps -a | grep "pattern"
```

Удаление:

```
docker ps -a | grep "pattern" | awk '{print $3}' | xargs docker rmi
```

Остановка и удаление всех контейнеров

Вы можете просмотреть все контейнеры в вашей системе при помощи команды `docker ps`. Для того чтобы отобразить все контейнеры, необходимо добавить к команде флажок `-a`. Если вы уверены, что желаете удалить все контейнеры, добавьте флажок `-q`, чтобы перенести ID всех контейнеров в команды `docker stop` и `docker rm`:

Список:

```
docker ps -a
```

Удаление:

```
docker stop $(docker ps -a -q)
docker rm $(docker ps -a -q)
```

Удаление томов

Удаление одного или нескольких конкретных томов - версия Docker 1.9 и старше

Используйте команду `docker volume ls`, чтобы найти имя или имена томов, которые вы желаете удалить. После этого вы можете удалить один или несколько томов, используя команду `docker volume rm`:

Список:

```
docker volume ls
```

Удаление:

```
docker volume rm volume_name volume_name
```

Удаление подвешенных томов - Версия Docker 1.9 и старше

Поскольку тома существуют независимо от контейнеров, то после удаления контейнера том не удаляется автоматически вместе с ним. В случае если в системе имеется том, который больше не привязан ни к одному из контейнеров, такой том называется подвешенным. Для того чтобы найти и удалить такие подвешенные тома, вы можете использовать команду `docker volume ls` с фильтром, который ограничивает результаты поиска до указанных томов. После того как система выдаст нужный вам список томов, вы можете добавить к команде флажок `-q`, чтобы перенести имена томов в команду `docker volume rm`:

Список:

```
docker volume ls -f dangling=true
```

Удаление:

```
docker volume rm $(docker volume ls -f dangling=true -q)
```

Удаление контейнера и его тома

Если вы создали безымянный том, его можно удалить вместе с контейнером, добавив к команде флажок `-v`. Обратите внимание, что данная команда работает только для безымянных томов. После успешного удаления контейнера отображается его ID. Обратите внимание, что система при этом не выдает какой-либо информации касательно удаления тома. Безымянные тома просто молча удаляются вместе с контейнером. Если том имеет имя, он также молча остается в системе.

Удаление:

```
docker rm -v container_name
```

Заключение

В настоящем руководстве описаны наиболее распространенные команды, которые применяются для удаления образов, контейнеров и томов в версии Docker 1.9 и старше. Вместе с каждой командой можно использовать различные комбинации флажков. Для более подробного описания функционала, доступного в системе, смотри документацию Docker по командам `docker rmi`, `docker rm` и `docker volume rm`. Если вы не нашли в этом руководстве информации о решении других распространенных задач, связанных с очисткой системы от ненужных образов, контейнеров и томов, пожалуйста, напишите нам или оставьте свои предложения в комментариях.

[docker](#)[администрирование](#)

ОБ АВТОРЕ

Павел Соловьёв



Комментарии

Пока что нет комментариев, вы можете быть первым.

[Войти](#) или [Регистрация](#), чтобы оставлять комментарии.

Devacademy в соцсетях

