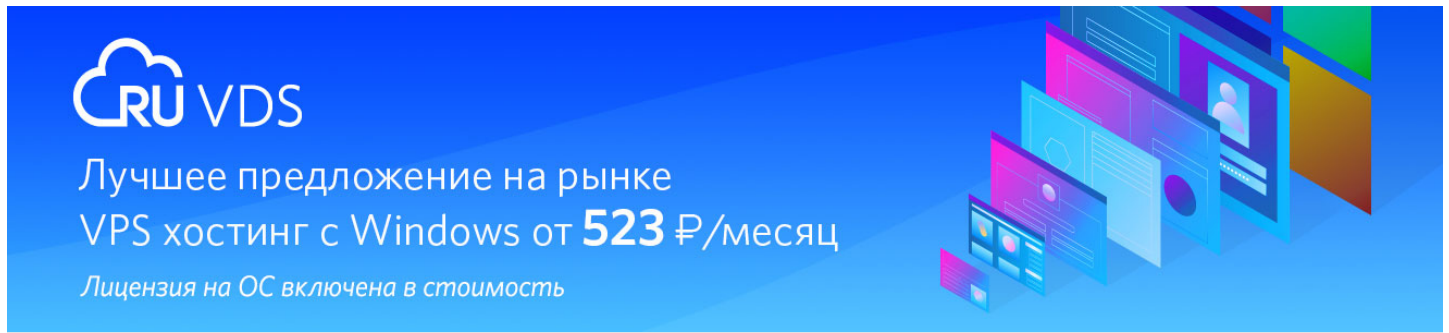


[КАК СТАТЬ АВТОРОМ](#)[Поехали в гик-трип по Календарю IT-ивентов?](#)

CRU VDS

Лучшее предложение на рынке
VPS хостинг с Windows от **523** ₽/месяц

Лицензия на ОС включена в стоимость

**1923.28**

Рейтинг

RUVDS.comVDS/VPS-хостинг. Скидка 15% по коду **HABR15****ru_vds**

28 фев 2019 в 19:00

Изучаем Docker, часть 6: работа с данными

🕒 5 мин 👁 181K

Блог компании RUVDS.com, Разработка веб-сайтов*, Виртуализация*

[Тutorial](#)[Перевод](#)

Автор оригинала: Jeff Hale

В сегодняшней части перевода серии материалов о Docker мы поговорим о работе с данными. В частности – о томах Docker. В этих материалах мы постоянно сравнивали программные механизмы Docker с разными съедобными аналогиями. Не будем отходить от этой традиции и здесь. Данные в Docker пусть будут специями. В мире существует множество видов специй, а в Docker – множество способов работы с данными.

- [Часть 1: основы](#)
- [Часть 2: термины и концепции](#)
- [Часть 3: файлы Dockerfile](#)
- [Часть 4: уменьшение размеров образов и ускорение их сборки](#)
- [Часть 5: команды](#)
- [Часть 6: работа с данными](#)



Обратите внимание на то, что этот материал подготовлен с использованием движка Docker версии 18.09.1 и API версии 1.39.

Данные в Docker могут храниться либо временно, либо постоянно. Начнём с временных данных.

Временное хранение данные

В контейнерах Docker организовать работу с временными данными можно двумя способами.

По умолчанию файлы, создаваемые приложением, работающим в контейнере, сохраняются в слое контейнера, поддерживающем запись. Для того чтобы этот механизм работал, ничего специально настраивать не нужно. Получается дёшево и сердито. Приложению достаточно просто сохранить данные и продолжить заниматься своими делами. Однако после того как контейнер перестанет существовать, исчезнут и данные, сохранённые таким вот нехитрым способом.

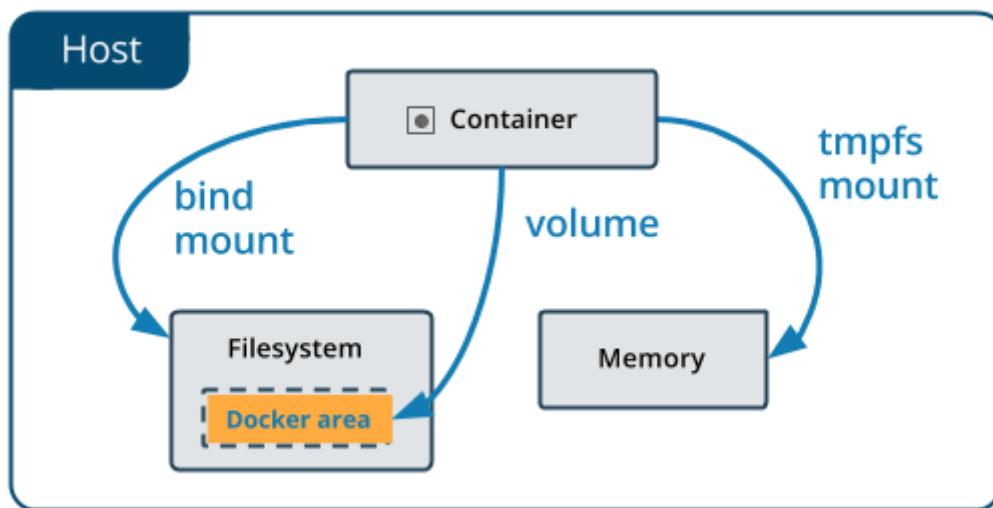
Для хранения временных файлов в Docker можно воспользоваться ещё одним решением, подходящим для тех случаев, когда требуется более высокий уровень производительности, в сравнении с тем, который достигим при использовании стандартного механизма временного хранения данных. Если вам не нужно, чтобы ваши данные хранились бы дольше,

чем существует контейнер, вы можете подключить к контейнеру `tmpfs` – временное хранилище информации, которое использует оперативную память хоста. Это позволит ускорить выполнение операций по записи и чтению данных.

Часто бывает так, что данные нужно хранить и после того, как контейнер прекратит существовать. Для этого нам пригодятся механизмы постоянного хранения данных.

Постоянное хранение данных

Существуют два способа, позволяющих сделать срок жизни данных большим срока жизни контейнера. Один из способов заключается в использовании технологии `bind mount`. При таком подходе к контейнеру можно примонтировать, например, реально существующую папку. Работать с данными, хранящимися в такой папке, смогут и процессы, находящиеся за пределами Docker. Вот как выглядят монтирование `tmpfs` и технология `bind mount`.



Монтирование `tmpfs` и `bind mount`

Минусы использования технологии `bind mount` заключаются в том, что её использование усложняет резервное копирование данных, миграцию данных, совместное использование данных несколькими контейнерами. Гораздо лучше для постоянного хранения данных использовать тома Docker.

Томы Docker

Том – это файловая система, которая расположена на хост-машине за пределами контейнеров. Созданием и управлением томами занимается Docker. Вот основные свойства томов Docker:

- Они представляют собой средства для постоянного хранения информации.

- Они самостоятельны и отделены от контейнеров.
- Ими могут совместно пользоваться разные контейнеры.
- Они позволяют организовать эффективное чтение и запись данных.
- Тома можно размещать на ресурсах удалённого облачного провайдера.
- Их можно шифровать.
- Им можно давать имена.
- Контейнер может организовать заблаговременное наполнение тома данными.
- Они удобны для тестирования.

Как видите, тома Docker обладают замечательными свойствами. Давайте поговорим о том, как их создавать.

Создание томов

Тома можно создавать средствами Docker или с помощью запросов к API.

Вот инструкция в Dockerfile, которая позволяет создать том при запуске контейнера.

```
VOLUME /my_volume
```

При использовании подобной инструкции Docker, после создания контейнера, создаст том, содержащий данные, которые уже имеются в указанном месте. Обратите внимание на то, что если вы создаёте том с использованием Dockerfile, это не освобождает вас от необходимости указать точку монтирования тома.

Создавать тома в Dockerfile можно и используя формат JSON.

Кроме того, тома можно создавать средствами командной строки во время работы контейнера.

Работа с томами из командной строки

Создание тома

Создать самостоятельный том можно следующей командой:

```
docker volume create --name my_volume
```

Выяснение информации о томах

Для того чтобы просмотреть список томов Docker, воспользуйтесь следующей командой:

```
docker volume ls
```

Исследовать конкретный том можно так:

```
docker volume inspect my_volume
```

Удаление тома

Удалить том можно так:

```
docker volume rm my_volume
```

Для того чтобы удалить все тома, которые не используются контейнерами, можно прибегнуть к такой команде:

```
docker volume prune
```

Перед удалением томов Docker запросит у вас подтверждение выполнения этой операции.

Если том связан с каким-либо контейнером, такой том нельзя удалить до тех пор, пока не удалён соответствующий контейнер. При этом, даже если контейнер удалён, Docker не всегда это понимает. Если это случилось – можете воспользоваться следующей командой:

```
docker system prune
```

Она предназначена для очистки ресурсов Docker. После выполнения этой команды у вас должна появиться возможность удалить тома, статус которых до этого определялся

неправильно.

Флаги `--mount` и `--volume`

Для работы с томами вам, при вызове команды `docker`, часто придётся пользоваться флагами. Например, для того чтобы создать том во время создания контейнера можно воспользоваться такой конструкцией:

```
docker container run --mount source=my_volume, target=/container/path/for/volume
```

В давние времена (до 2017 года) популярен был флаг `--volume`. Изначально этот флаг (ещё им можно пользоваться в сокращённом виде, тогда он выглядит как `-v`) использовался для самостоятельных контейнеров, а флаг `--mount` – в среде Docker Swarm. Однако, начиная с Docker 17.06, флаг `--mount` можно использовать в любых сценариях.

Надо отметить, что при использовании флага `--mount` увеличивается объём дополнительных данных, которые приходится указывать в команде, но, по нескольким причинам, лучше использовать именно этот флаг, а не `--volume`. Флаг `--mount` – это единственный механизм, который позволяет работать с сервисами или указывать параметры драйвера тома. Кроме того, работать с этим флагом проще.

В существующих примерах команд, направленных на работу с данными в Docker, вы можете встретить множество примеров употребления флага `-v`. Пытаясь адаптировать эти команды для себя, учитывайте то, что флаги `--mount` и `--volume` используют различные форматы параметров. То есть, нельзя просто заменить `-v` на `--mount` и получить рабочую команду.

Главное различие между `--mount` и `--volume` заключается в том, что при использовании флага `--volume` все параметры собирают вместе, в одном поле, а при использовании `--mount` параметры разделяются.

При работе с `--mount` параметры представлены как пары вида ключ-значение, а именно, это выглядит как `key=value`. Эти пары разделяют запятыми. Вот часто используемые параметры `--mount`:

- `type` – тип монтирования. Значением для соответствующего ключа могут выступать `bind`, `volume` или `tmpfs`. Мы тут говорим о томах, то есть – нас интересует значение

volume .

- `source` – источник монтирования. Для именованных томов это – имя тома. Для неименованных томов этот ключ не указывают. Он может быть сокращён до `src` .
- `destination` – путь, к которому файл или папка монтируется в контейнере. Этот ключ может быть сокращён до `dst` или `target` .
- `readonly` – монтирует том, который предназначен только для чтения. Использовать этот ключ необязательно, значение ему не назначают.

Вот пример использования `--mount` с множеством параметров:

```
docker run --mount type=volume,source=volume_name,destination=/path/in/container
```

Итоги

Вот полезные команды, которыми можно пользоваться при работе с томами Docker:

- `docker volume create`
- `docker volume ls`
- `docker volume inspect`
- `docker volume rm`
- `docker volume prune`

Вот список часто используемых параметров для `--mount` , применимых в команде вида `docker run --mount my_options my_image` :

- `type=volume`
- `source=volume_name`
- `destination=/path/in/container`
- `readonly`

Теперь, когда мы завершили эту серию материалов о Docker, пришло время сказать пару слов о том, куда тем, кто изучает Docker, можно двигаться дальше. Вот большая хорошая статья о Docker. Вот книга о Docker (покупая эту книгу, постарайтесь раздобыть самое

свежее её издание). Вот ещё одна книга, которая подойдёт тем, кто полагает, что практика – это лучший способ изучения технологий.

Уважаемые читатели! Какие материалы о Docker вы посоветовали бы изучить новичкам?

Habrahabr10

Промо-код для скидки в 10% на наши виртуальные сервера


Теги: Docker, разработка

Хабы: Блог компании RUVDS.com, Разработка веб-сайтов, Виртуализация

Редакторский дайджест


Присылаем лучшие статьи раз в месяц

Электронпочта



RUVDS.com
VDS/VPS-хостинг. Скидка 15% по коду **HABR15**

Telegram ВКонтакте Twitter



409

349.3

Карма

Рейтинг

@ru_vds
Пользователь


 Комментарии 12


Публикации


ЛУЧШИЕ ЗА СУТКИ ПОХОЖИЕ


23 часа назад

Герои напильника и паяльника: итоги сезона DIY

 8 мин

 8.5K

 Сезон DIY

 Спецпроект

 +44 43 6

spiritus_sancti

вчера в 16:00

RGB-усилители. Особенности, проблемы, выбор

 Простой 6 мин 5.6K

Тutorial

 +40 26 11

ru_vds

20 часов назад

Профилирование Python — почему и где тормозит ваш код

 Средний 10 мин 3.6K

Tutorial

Перевод

 +32 93 4

ViktorSergeev

16 часов назад

WebOne: даём жизнь старым браузерам

 6 мин 3.3K +31 32 11

zatiм

2 часа назад

Видеокарта VGA для микроконтроллера

 Средний 17 мин 911

Tutorial

 +25 23 10

it_union

3 часа назад

ЗАО Гейм Инсайт Труп

 Простой 4 мин 3.5K

 +20 8 10

AnnaVIMorozova

19 часов назад

Как повысить эффективность коммуникаций в команде: учимся решать конфликты

 7 мин  2.3K +19 43 2

Axelus

21 час назад

CRM — для руководителя или для сотрудников?

 Простой  6 мин  1.2K +16 15 0

Atomish

1 час назад

Как мы создали систему развития и карьерного роста продуктовых дизайнеров

 Средний  11 мин  167

Тutorial

 +15 2 1

Sagidullin

7 часов назад

Всего два месяца — и новый релиз ядра Linux. Что появилось в ядре 6.5, что изменилось и что удалили. Новые возможности

 5 мин  5.3K +15 6 8[Показать еще](#)

ИНФОРМАЦИЯ

Сайт	ruvds.com
Дата регистрации	18 марта 2016
Дата основания	27 июля 2015
Численность	11–30 человек
Местоположение	Россия
Представитель	ruvds

ССЫЛКИ

VPS / VDS сервер от 130 рублей в месяц.

ruvds.com

Дата-центры RUVDS в Москве, Санкт-Петербурге, Казани, Екатеринбурге, Новосибирске, Лондоне, Франкфурте, Цюрихе, Амстердаме

ruvds.com

Помощь и вопросы

ruvds.com

Партнерская программа RUVDS

ruvds.com

VPS (CPU 1x2ГГц, RAM 512Mb, SSD 10 Gb) – 190 рублей в месяц

ruvds.com

VPS Windows от 523 рублей в месяц. Бесплатный тестовый период 3 дня.

ruvds.com

VDS в Цюрихе. Дата-центр TIER III – швейцарское качество по низкой цене.

ruvds.com

Антивирусная защита виртуального сервера. Легкий агент для VPS.

ruvds.com

VPS в Лондоне. Дата-центр TIER III – английская точность за рубли.

ruvds.com

VPS с видеокартой на мощных серверах 3,4ГГц

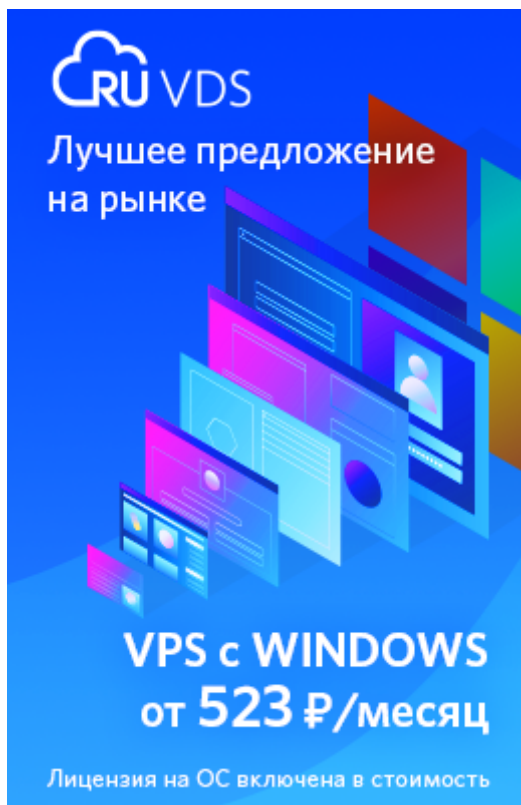
ruvds.com

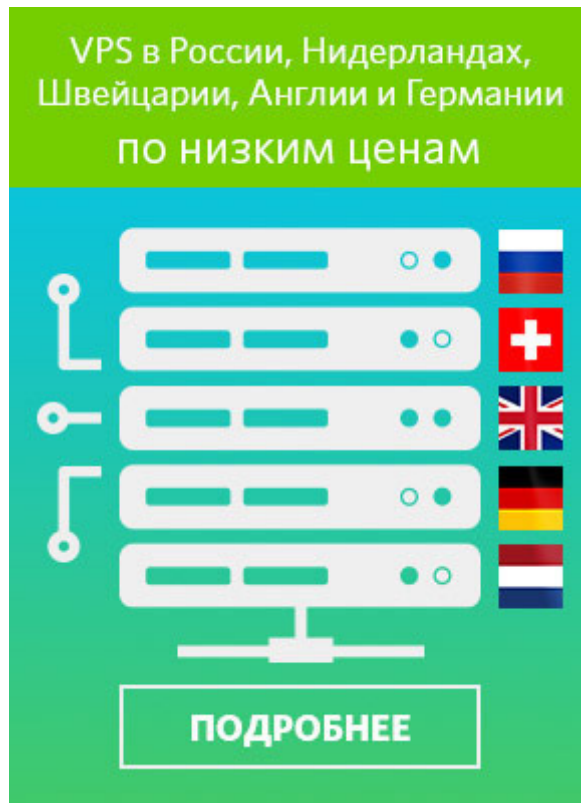
ПРИЛОЖЕНИЯ

**RUVDS Client**

Приложение для мониторинга и управления виртуальными серверами RUVDS с мобильных устройств.

Android iOS

ВИДЖЕТ**ВИДЖЕТ**



БЛОГ НА ХАБРЕ

20 часов назад

Профилирование Python — почему и где тормозит ваш код

3.7K 4

вчера в 16:00

RGB-усилители. Особенности, проблемы, выбор

5.7K 11

27 авг в 17:00

Интернет 90-х: когда после 20 часов в онлайн тебе пишет президент ISP

15K 40

26 авг в 17:00

История компьютерных стратегий. Часть 8. «Age of Empires»: шедевр геймдева, от которого бомбит у любителей истории

13K 12

25 авг в 16:00

Xbox is a new Dreamcast. Зачем покупать консоль от Microsoft в 2023 году и во что играть

6.5K 8

Ваш аккаунт	Разделы	Информация	Услуги
Войти	Статьи	Устройство сайта	Корпоративный блог
Регистрация	Новости	Для авторов	Медийная реклама
	Хабы	Для компаний	Нативные проекты
	Компании	Документы	Образовательные
	Авторы	Соглашение	программы
	Песочница	Конфиденциальность	Стартапам
			Спецпроекты



Настройка языка

Техническая поддержка

Вернуться на старую версию