



Docker:

хранение данных, использование томов

Спикер: Александр Швалов

Короткий срок жизни контейнеров

- В контейнере изначально находятся все данные и компоненты для запуска приложения.

Короткий срок жизни контейнеров

- В контейнере изначально находятся все данные и компоненты для запуска приложения.
- Приложение в контейнере может быть завершено в любой момент, а контейнер при этом уничтожен – и это нормально.

Короткий срок жизни контейнеров

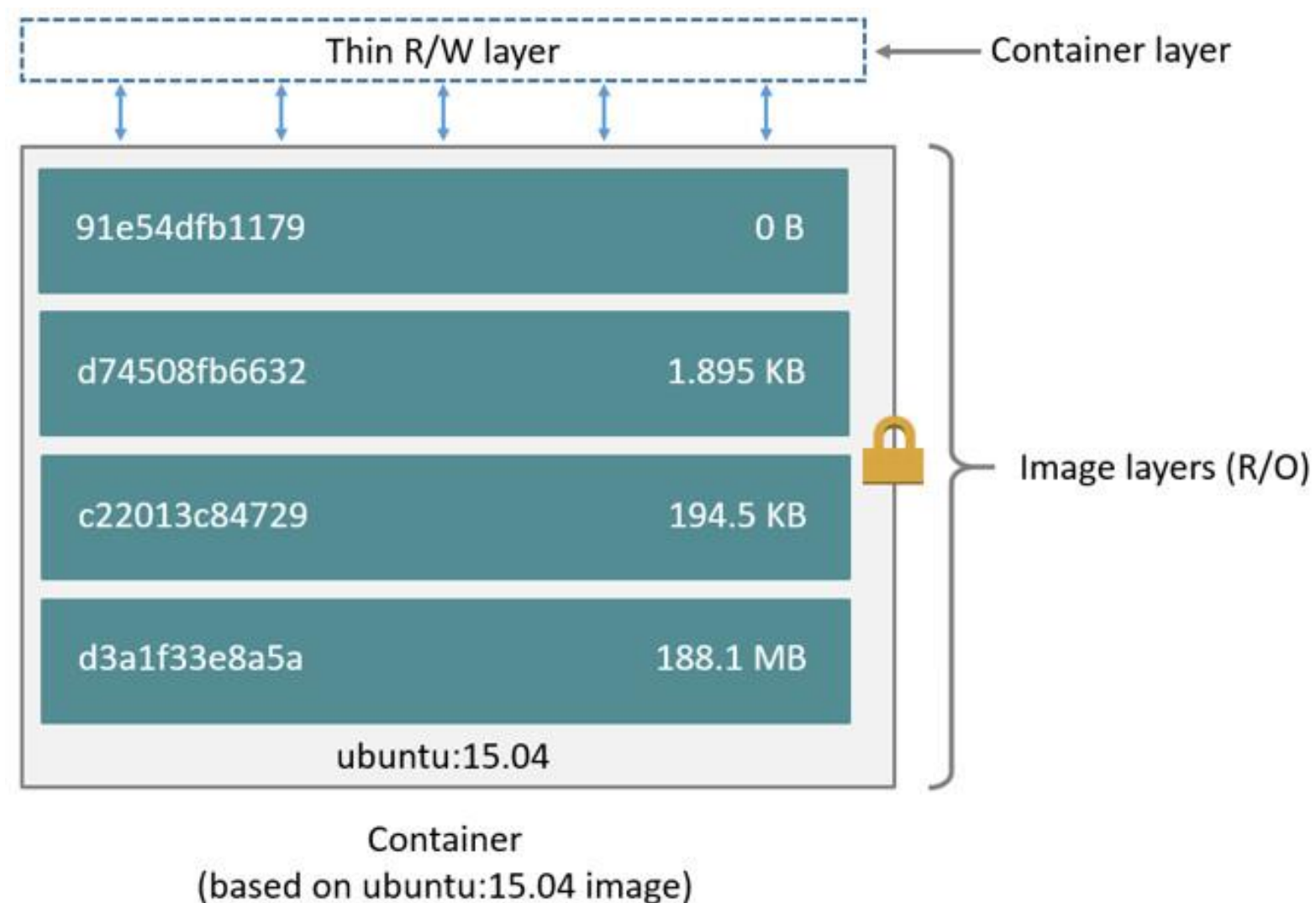
- В контейнере изначально находятся все данные и компоненты для запуска приложения.
- Приложение в контейнере может быть завершено в любой момент, а контейнер при этом уничтожен — и это нормально.
- Stateless-приложение не читает и не сохраняет данные о своем состоянии работы (готово к завершению в любой момент) - это идеальный случай для Docker.

Короткий срок жизни контейнеров

- В контейнере изначально находятся все данные и компоненты для запуска приложения.
- Приложение в контейнере может быть завершено в любой момент, а контейнер при этом уничтожен – и это нормально.
- Stateless-приложение не читает и не сохраняет данные о своем состоянии работы (готово к завершению в любой момент) - это идеальный случай для Docker.
- Однако большинство программ, с которыми мы работаем каждый день, относятся к категории `stateful`, а значит требуют способа хранения данных.

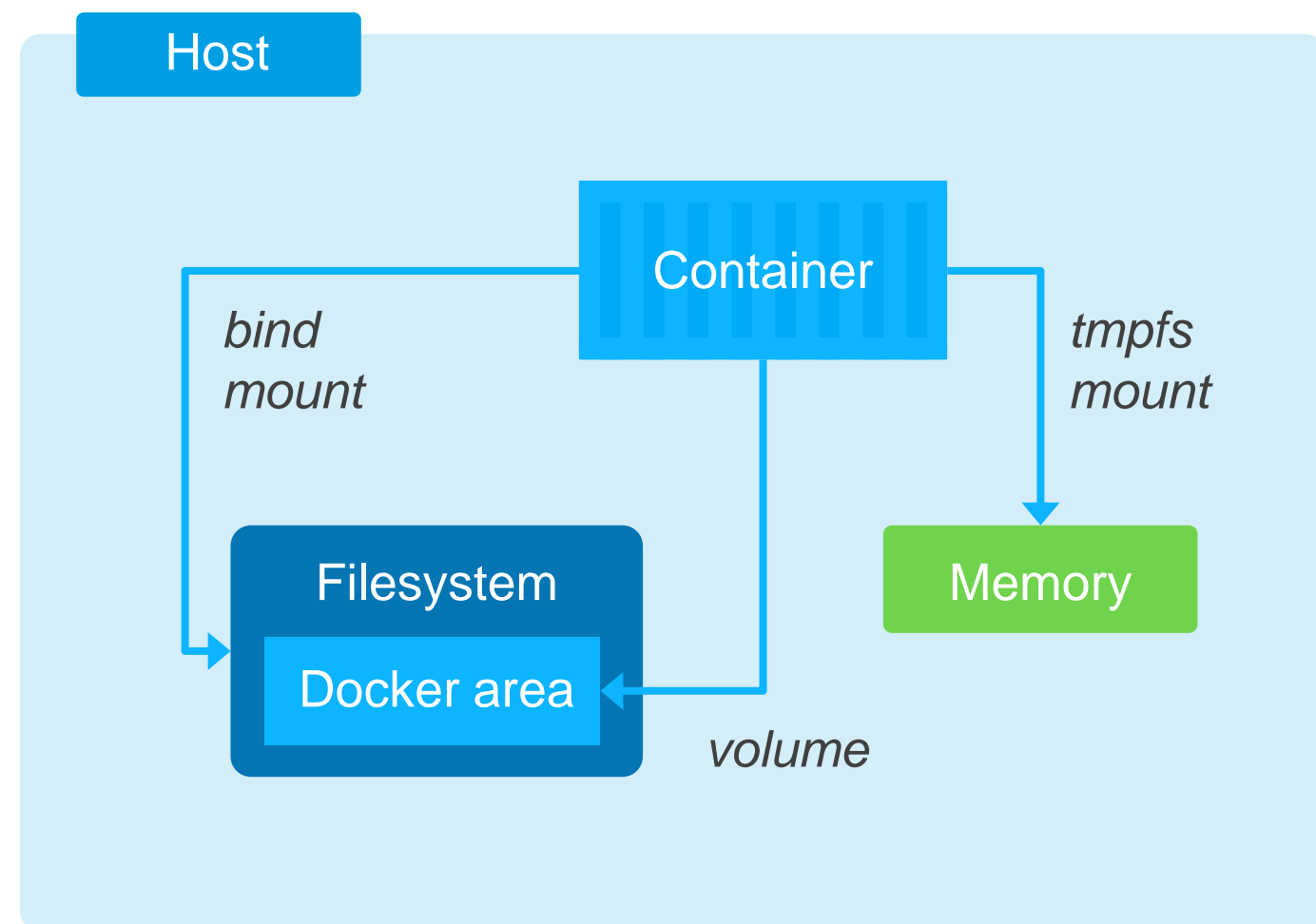
Специфика работы образа и контейнеров

- Только тонкий слой контейнера доступен для записи, остальные слои – это образ, и при работе контейнера в них ничего не пишется:



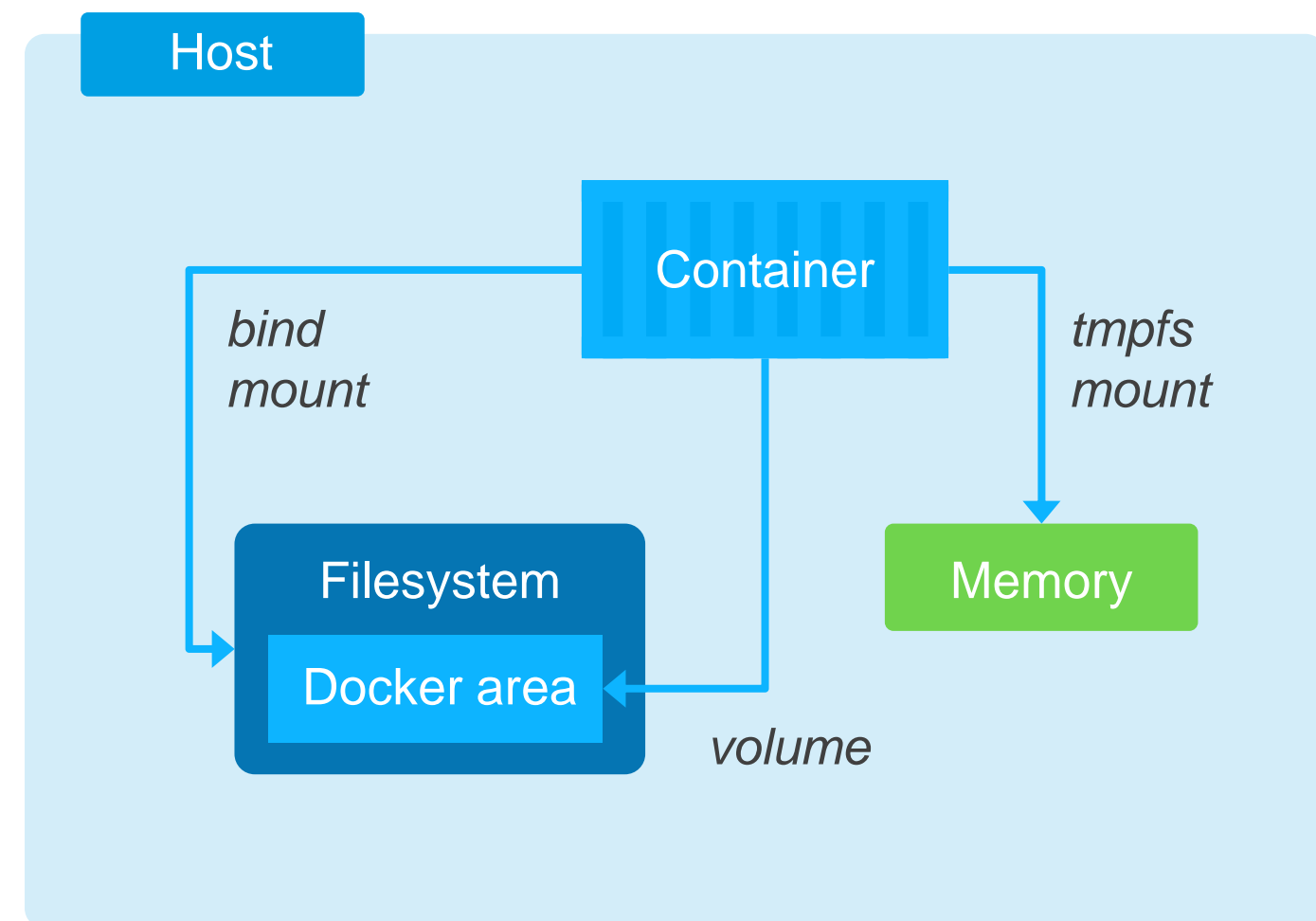
Что предлагает Docker?

- Тома хранения данных (docker volumes)



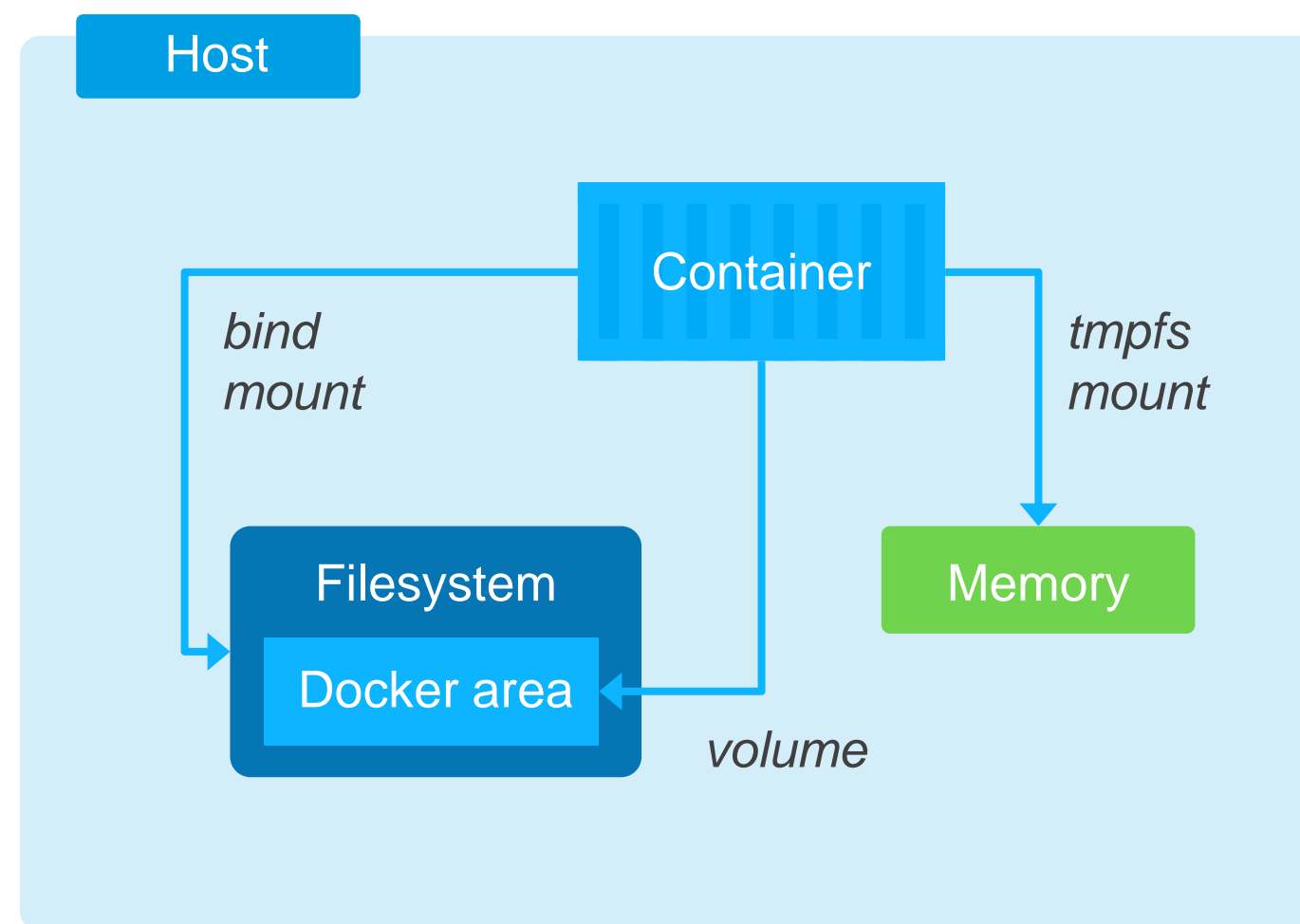
Что предлагает Docker?

- Тома хранения данных (docker volumes)
- Подключение каталога с хоста в контейнер (bind mount)



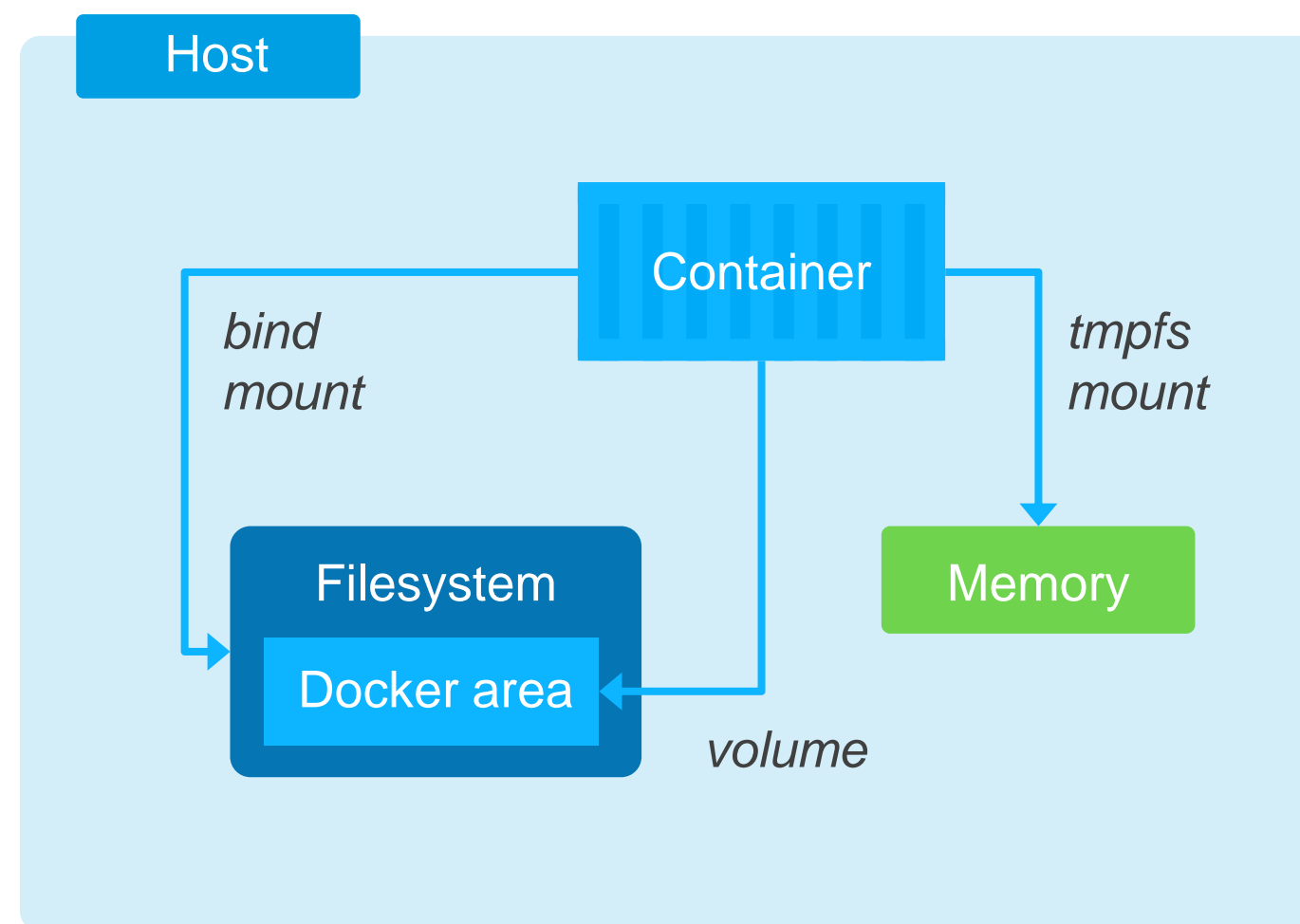
Что предлагает Docker?

- Тома хранения данных (docker volumes)
- Подключение каталога с хоста в контейнер (bind mount)
- Именованные каналы (named pipes, только в Windows)



Что предлагает Docker?

- Тома хранения данных (docker volumes)
- Подключение каталога с хоста в контейнер (bind mount)
- Именованные каналы (named pipes, только в Windows)
- Монтирование tmpfs (только в Linux)



Тома в Docker

- Рекомендованный разработчиками Docker способ хранения данных.

Тома в Docker

- Рекомендованный разработчиками Docker способ хранения данных.
- В Linux-системе тома находятся по умолчанию в [/var/lib/docker/volumes/](#).

Тома в Docker

- Рекомендованный разработчиками Docker способ хранения данных.
- В Linux-системе тома находятся по умолчанию в `/var/lib/docker/volumes/`.
- Тома управляются с помощью команд и ключей docker, а также в `Dockerfile` или `docker-compose.yml`.

Тома в Docker

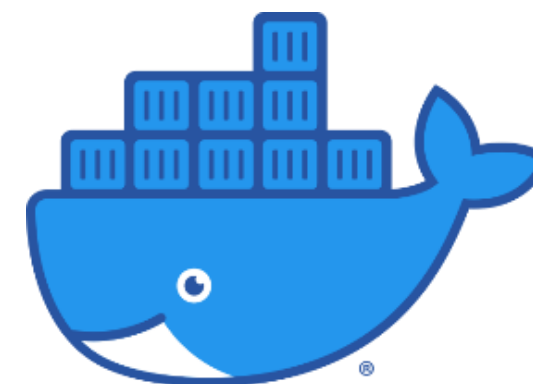
- Рекомендованный разработчиками Docker способ хранения данных.
- В Linux-системе тома находятся по умолчанию в `/var/lib/docker/volumes/`.
- Тома управляются с помощью команд и ключей docker, а также в `Dockerfile` или `docker-compose.yml`.
- Один том могут использовать несколько контейнеров одновременно.

Тома в Docker

- Рекомендованный разработчиками Docker способ хранения данных.
- В Linux-системе тома находятся по умолчанию в `/var/lib/docker/volumes/`.
- Тома управляются с помощью команд и ключей docker, а также в `Dockerfile` или `docker-compose.yml`.
- Один том могут использовать несколько контейнеров одновременно.
- Можно выбрать драйвер тома и хранить данные не на хосте, а на удалённом сервере или в облаке.

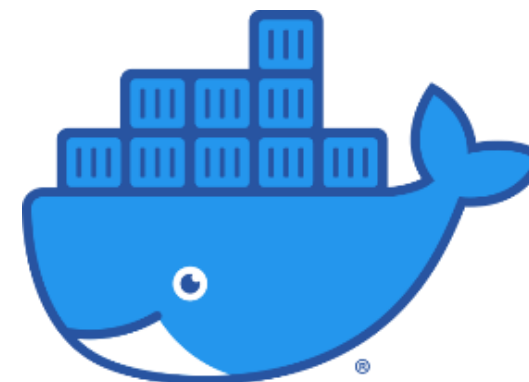
Применение Docker volumes

- для общего использования данных несколькими контейнерами;



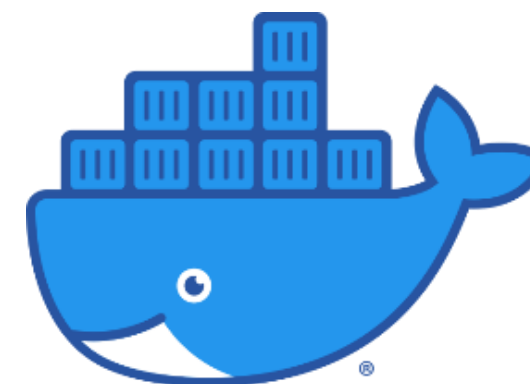
Применение Docker volumes

- для общего использования данных несколькими контейнерами;
- если необходима привязка к операционной системе хоста;



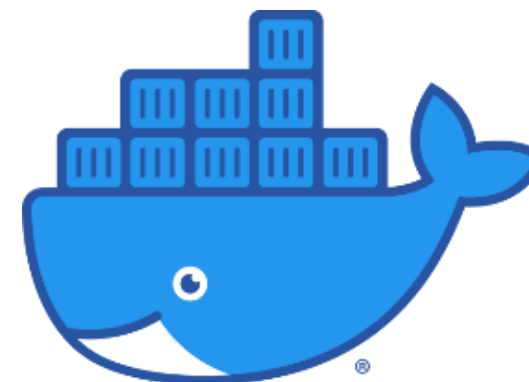
Применение Docker volumes

- для общего использования данных несколькими контейнерами;
- если необходима привязка к операционной системе хоста;
- беспроблемное удалённое хранение данных;



Применение Docker volumes

- для общего использования данных несколькими контейнерами;
- если необходима привязка к операционной системе хоста;
- беспроблемное удалённое хранение данных;
- бэкап или миграция данных на другой хост с Docker.



Подключение каталога с хоста

Плюсы:

- это просто для понимания и реализации;

Подключение каталога с хоста

Плюсы:

- это просто для понимания и реализации;
- позволяет быстро получить актуальный конфиг с хоста;

Подключение каталога с хоста

Плюсы:

- это просто для понимания и реализации;
- позволяет быстро получить актуальный конфиг с хоста;
- можно оперативно править файлы, удобно для разработки и отладки.

Подключение каталога с хоста

Плюсы:

- это просто для понимания и реализации;
- позволяет быстро получить актуальный конфиг с хоста;
- можно оперативно править файлы, удобно для разработки и отладки.

Минусы:

- есть риск случайно затереть данные, не понимая что с ними работает контейнер;

Подключение каталога с хоста

Плюсы:

- это просто для понимания и реализации;
- позволяет быстро получить актуальный конфиг с хоста;
- можно оперативно править файлы, удобно для разработки и отладки.

Минусы:

- есть риск случайно затереть данные, не понимая что с ними работает контейнер;
- лучше не использовать в продакшене.

Монтирование tmpfs

- на самом деле нужно не для хранения данных;

Монтирование tmpfs

- на самом деле нужно не для хранения данных;
- высокая скорость доступа и безопасность;

Монтирование tmpfs

- на самом деле нужно не для хранения данных;
- высокая скорость доступа и безопасность;
- работает только с одним контейнером;

Монтирование tmpfs

- на самом деле нужно не для хранения данных;
- высокая скорость доступа и безопасность;
- работает только с одним контейнером;
- доступно только в Linux.

Знаете ли Вы, что...

- Если мы монтируем пустой том, а в контейнере уже есть файлы по этому пути – то они не удалятся, а будут скопированы в том.

Знаете ли Вы, что...

- Если мы монтируем пустой том, а в контейнере уже есть файлы по этому пути – то они не удалятся, а будут скопированы в том.
- Если вы подключите непустой том или каталог с хоста в контейнер, то данные внутри контейнера тоже не удалятся, а просто будут скрыты.

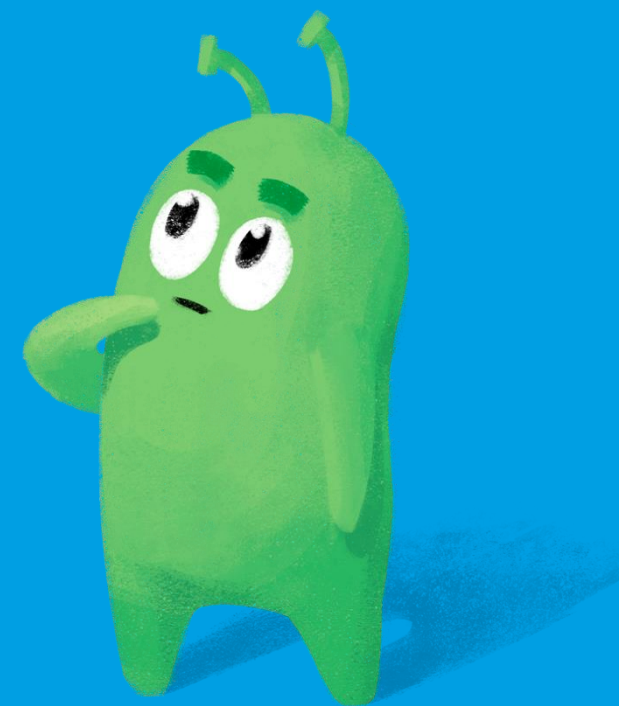
Знаете ли Вы, что...

- Если мы монтируем пустой том, а в контейнере уже есть файлы по этому пути – то они не удалятся, а будут скопированы в том.
- Если вы подключите непустой том или каталог с хоста в контейнер, то данные внутри контейнера тоже не удалятся, а просто будут скрыты.
- Можно монтировать с хоста любые файлы, в том числе служебные – например сокет `docker`, и мы получаем `docker-in-docker`.

Знаете ли Вы, что...

- Если мы монтируем пустой том, а в контейнере уже есть файлы по этому пути – то они не удалятся, а будут скопированы в том.
- Если вы подключите непустой том или каталог с хоста в контейнер, то данные внутри контейнера тоже не удалятся, а просто будут скрыты.
- Можно монтировать с хоста любые файлы, в том числе служебные – например сокет `docker`, и мы получаем `docker-in-docker`.
- Подключать каталог `/var/lib/docker` внутрь контейнера не рекомендуется, однако используется для запуска некоторых приложений.

Практика



СЛЁРМ



slurm.io



Southbridge



southbridge.io