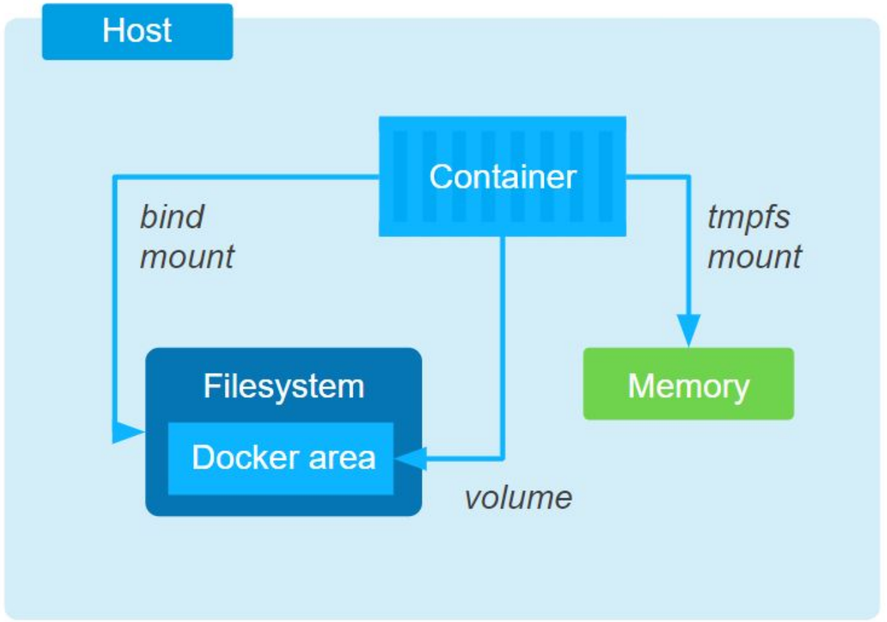
Skillbox

Volume

Что такое volume?

Volume — это дисковое пространство между хостом и контейнером



Временное хранение данных

Способы хранения:

1. По умолчанию файлы, создаваемые приложением, работающим в контейнере, сохраняются в слое контейнера, поддерживающем запись
2. Если вам не нужно, чтобы ваши данные хранились дольше, чем существует контейнер, вы можете подключить к контейнеру tmpfs — временное хранилище информации, которое использует оперативную память хоста

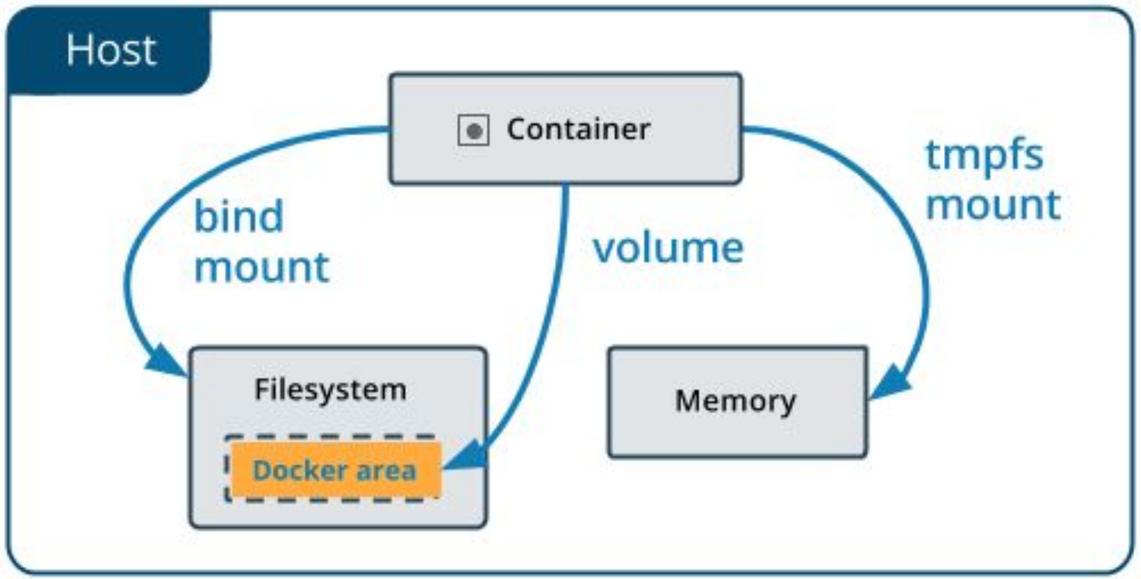
Постоянное хранение данных

Способы хранения:

1. 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. На рисунке сбоку показано, как выглядит монтирование tmpfs и технология bind mount

2. Тома



Тома

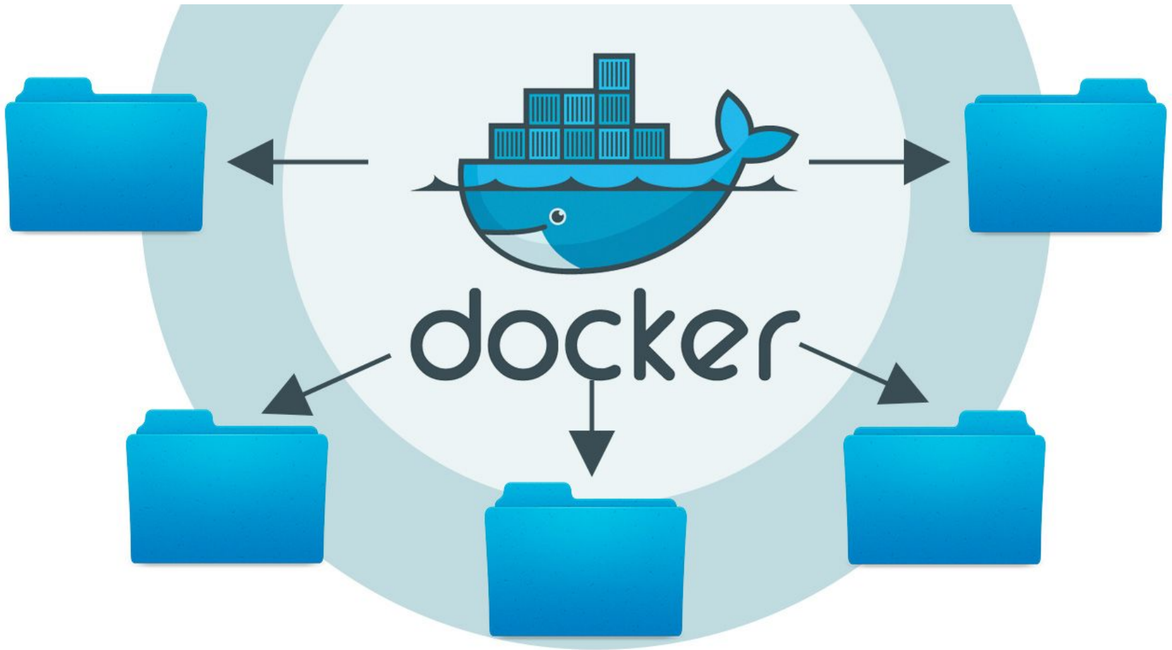
Том — это файловая система, которая расположена на хост-машине за пределами контейнеров. Созданием и управлением томами занимается Docker

- 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

Создание томов

Dockerfile:

VOLUME /my_volume



Работа с томами

- Создание тома — docker volume create - name my_volume
- Выяснить информацию о томах — docker volume ls
- Исследовать конкретный том — docker volume inspect my_volume
- Удаление тома — docker volume rm my_volume
- Удалить все тома — docker volume prune
- Очистка ресурсов Docker — docker system prune

Флаги --mount и --volume

Создать том во время создания контейнера можно, воспользовавшись такой конструкцией:

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

1. --volume
2. --mount

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

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

Пример использования --mount с множеством параметров:

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

Практическое применение

Создать volume — $ docker volume create my-vol
Вывести список созданных volume — $ docker volume ls
Просмотр volume — $ docker volume inspect my-vol
Удаление volume — $ docker volume rm my-vol

А теперь рассмотрим 2 примера запуска контейнера с nginx с volume с read\write

```
$ docker run -d \
  --name nginxtest \
  --mount source=myvol2,target=/app \
  nginx:latest
```

Пример запуска контейнера с nginx с volume в read-only:

```
$ docker run -d \
  --name=nginxtest \
  --mount source=nginx-vol,destination=/usr/share/nginx/html,readonly \
  nginx:latest
```

Итоги

- Узнали, что такое volume
- Узнали, про временное и постоянное хранение
- Узнали, что такое том
- Узнали, как создавать тома
- Узнали, как работать с томами
- Узнали про флаги mount и volume
- Узнали про практическое применение

Спасибо
за внимание!