

Тома (volumes)

Тома (volumes) – файлы или каталоги, смонтированные на хосте и не являющиеся частью каскадно-объединенной файловой системы.

Другие контейнеры могут совместно использовать их, и все изменения будут сразу же фиксироваться в файловой системе хоста.

Устаревшее: В качестве точки монтирования можно определить любой другой каталог хоста в команде `docker run` (например, `docker run -d -v /host/dir:/container/dir test/web-server`).

В Dockerfile `bind mounts` не работает - а это и не надо делать, т к это определяется при старте образа/через `compose`.

Способы хранения данных:

1. Временное (удаляется при остановке контейнера)
 1. по умолчанию изолировано на диске
 2. `tmpfs` в оперативной памяти
2. Постоянное
 1. обычные тома Docker
 2. `bind mounts` - прямое монтирование папки в контейнер

Драйвера volumes

Драйвер	Описание
local	Драйвер по умолчанию. Только точки монтирования, доступные на хосте.
	И еще штук 30 драйверов, список драйверов

Управление томами при запуске контейнера из консоли:

Если тома нет - будет создан

Основная команда	Параметр	Описание
<code>docker run ... --mount</code>	<code>type=</code>	Тип тома: <ul style="list-style-type: none">• <code>volume</code>• <code>bind</code>• <code>tmpfs</code>

Основная команда	Параметр	Описание
	source(src)=	Имя тома или не указывается для анонимных
	destination(dst)=	точка монтирования в контейнере
	volume-driver=	local по умолчанию, локальный том
	volume-opt=	опция=значение <div> volume-opt=type=nfs,volume-opt=device=<nfs-server>:<nfs-path> </div>
	readonly	только для чтения
docker run ... --volumes-from ContID		связывание с томами контейнера

Пример:

```
--mount 'type=volume,src=data-volume,dst=/var/opt/project,volume-driver=local,readonly'
```

Без пробелов после запятых.

Создание и управление томами независимо от контейнеров

Основная команда	Параметр	Описание
docker volume	create --name my_volume create my_volume	Создание тома. /var/lib/docker/volumes/имя тома/_data - расположение файлов По умолчанию на хосте в каталоге установки Docker (обычно каталог /var/lib/docker/). /var/lib/docker/volumes/
	ls	местоположение определенных томов, по имени или ID тома.
	inspect my_volume	информация о томе
	rm my_volume	удаление тома
	prune	удаление всех томов, которыми не пользуются контейнерами. Но иногда после удаления контейнера данные не обновляются

Основная команда	Параметр	Описание
docker system prune		очистка ресурсов docker. Потом - повторное удаление томов.

Примеры:

Автоматическое создание том с именем test-data в /var/lib/docker/volumes/test-data/_data

```
docker run -ti -d --name alpine-container -v test-data:/var/lib/app/content alpine
```

Монтирование в другую точку.

Нужно создать директорию.

```
mkdir /home/avimanyu/test-data  
docker run -ti -d --name alpine-container -v /home/avimanyu/test-data:/var/lib/app/content alpine
```

Архивировать том datavol в dbdata и восстановить его в том dbdata2 контейнера dbstore2

Создание тома datavol в контейнере dbdata

```
docker run -it --mount 'type=volume,src=datavol,dst=/datavol' --name dbstore ubuntu /bin/bash
```

Создание временного контейнера, примонтирование тома из dbdata, создание тома backup и сохранение архива datavol в backup и удаление временного контейнера

```
docker run --rm --volumes-from dbstore --mount 'type=volume,src=backup,dst=/backup' --name tmpubn ubuntu  
tar cvf /backup/backup.tar /datavol
```

Создание тома datavol2 в контейнере dbdata2

```
docker run -it --mount 'type=volume,src=datavol2,dst=/datavol' --name dbstore2 ubuntu /bin/bash
```

Создание временного контейнера, примонтирование тома из dbstore2, создание тома backup и сохранение архива datavol в backup и удаление временного контейнера

```
docker run --rm --volumes-from dbstore2 --mount 'type=volume,src=backup,dst=/backup' ubuntu bash -c "cd  
/datavol && tar xvf /backup/backup.tar --strip 1"
```

Revision #4

Created 29 June 2024 13:46:39 by Admin

Updated 18 March 2025 15:48:00 by Admin