



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Docker Volumes

Understanding Volumes

Contents

- [1 Choose the -v or --mount flag](#)
- [2 Escape Values from Outer CSV Parser](#)
- [3 Differences between -v and --mount behavior](#)

This section explains what docker volumes are all about.

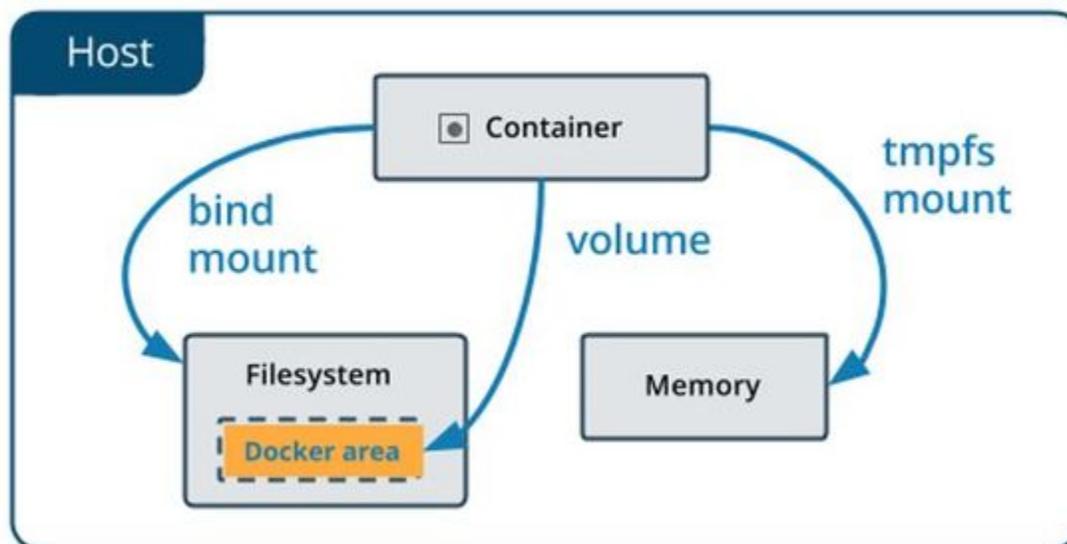
Warning

The following content has been deprecated and is maintained for reference only.

Volumes are the preferred mechanism for persisting data generated by and used by Docker containers. While bind mounts are dependent on the directory structure of the host machine, volumes are completely managed by Docker. Volumes have several advantages over bind mounts:

- Volumes are easier to back up or migrate than bind mounts.
- You can manage volumes using Docker CLI commands or the Docker API.
- Volumes work on both Linux and Windows containers.
- Volumes can be more safely shared among multiple containers.
- Volume drivers let you store volumes on remote hosts or cloud providers, to encrypt the contents of volumes, or to add other functionality.
- New volumes can have their content pre-populated by a container.

In addition, volumes are often a better choice than persisting data in a container's writable layer, because a volume does not increase the size of the containers using it, and the volume's contents exist outside the life cycle of a given container.



If your container generates non-persistent state data, consider using a **tmpfs mount** to avoid storing the data anywhere permanently, and to increase the container's performance by avoiding writing into the container's writable layer. Volumes use rprivate bind propagation, and bind propagation is not configurable for volumes.

Choose the `-v` or `--mount` flag

Originally, the `-v` or `--volume` flag was used for standalone containers and the `--mount` flag was used for swarm services. However, starting with Docker 17.06, you can also use `--mount` with standalone containers. In general, `--mount` is more explicit and verbose. The biggest difference is that the `-v` syntax combines all the options together in one field, while the `--mount` syntax separates them. Here is a comparison of the syntax for each flag. New users should try `--mount` syntax which is simpler than `--volume` syntax. If you need to specify volume driver options, you must use `--mount`.

- **`-v` or `--volume`**: Consists of three fields, separated by colon characters (:). The fields must be in the correct order, and the meaning of each field is not immediately obvious.
 - In the case of named volumes, the first field is the name of the volume, and is unique on a given host machine. For anonymous volumes, the first field is omitted.
 - The second field is the path where the file or directory are mounted in the container.
 - The third field is optional, and is a comma-separated list of options, such as **`ro`**. These options are discussed below.
- **`--mount`**: Consists of multiple key-value pairs, separated by commas and each consisting of a **key=value** tuple. The `--mount` syntax is more verbose than `-v` or `--volume`, but the order of the keys is not significant, and the value of the flag is easier to understand.
 - The **type** of the **mount**, which can be **bind**, **volume**, or **tmpfs**. This topic discusses volumes, so the type is always **volume**.
 - The **source** of the mount. For named volumes, this is the name of the volume. For anonymous volumes, this field is omitted. May be specified as **source** or **src**.
 - The **destination** takes as its value the path where the file or directory is mounted in the container. May be specified as **destination**, **dst**, or **target**.
 - The **read-only** option, if present, causes the bind mount to be mounted into the container as read-only.
 - The **volume-opt** option, which can be specified more than once, takes a key-value pair consisting of the option name and its value.

Escape Values from Outer CSV Parser

If your volume driver accepts a comma-separated list as an option, you must escape the value from the outer CSV parser. To escape a volume-opt, surround it with double quotes (") and surround the entire mount parameter with single quotes (').

For example, the local driver accepts mount options as a comma-separated list in the `o` parameter. This example shows the correct way to escape the list.

