

Docker container-Bind mounts

In Docker, there are several ways to persist data across container runs. Two popular methods are **bind mounts** and **volumes**. Both allow you to share files between the host machine and the container, but they differ in their approach.

Bind Mounts

A **bind mount** is a way to directly attach a directory on the host machine to a directory inside a Docker container. This allows the container to read and write files from/to that directory, just like if it were running on the host.

Example:

Suppose you have a directory `mydata` on your host machine:

```
$ mkdir mydata
```

You can create a Docker container with a bind mount by specifying the `-v` flag followed by the path to the host directory and the path inside the container where it should be mounted:

```
docker run -it --rm -v /path/to/mydata:/container/path myimage
```

This command creates a new container from `myimage`, and mounts `/path/to/mydata` on your host machine to `/container/path` inside the container. Any changes made to files in `/container/path` will be written to `/path/to/mydata` on your host machine.

Volumes

A **volume** is similar to a bind mount, but it's managed by Docker and stored as a directory tree within the Docker storage system. This means that volumes are persisted even when a container is deleted or recreated from an image.

To create a volume, you can use the `docker run` command with the `-v` flag followed by the name of the volume and the path inside the container:

```
docker run -it --rm -v myvolume:/container/path myimage
```

This command creates a new container from `myimage`, and mounts a new volume named `myvolume` to `/container/path` inside the container.

Key differences:

- Bind mounts are directly attached to host directories, while volumes are stored as directory trees within the Docker storage system.
- Volumes are persisted even when containers are deleted or recreated, while bind mounts can be lost if the host directory is deleted or modified.
- Volumes are more portable and secure than bind mounts.

Summary

Both bind mounts and volumes allow you to persist data across container runs in Docker. Bind mounts directly attach host directories to container directories, while volumes store data as managed directory trees within the Docker storage system.

Curated by Brajesh Kumar