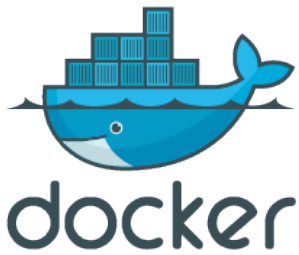
Bind mounts (persistant data into Container)



Bind Mounts

When you use a bind mount, a file or directory on the host machine is mounted from the host into a container. By contrast, when you use a volume, a new directory is created within Docker's storage directory on the host machine, and Docker manages that directory's contents.

When to Use Bind Mounts

Bind mounts are appropriate for the following types of use cases:

- Sharing source code or build artifacts between a development environment on the Docker host and a container.
- Creating or generating files in a container and persisting them onto the host's filesystem.
- Sharing configuration files from the host machine to containers, such as how Docker provides DNS resolution by mounting `/etc/resolv.conf` from the host machine into each container.
- Bind mounts can also be used for builds: you can bind mount source code from the host into the build container to test, lint, or compile a project.

Bind-Mounting Over Existing Data

If you bind mount a file or directory into a directory in the container in which files or directories exist, the pre-existing files are obscured by the mount. Once the mount is removed, the files become visible again.

With containers, there's no straightforward way of removing a mount to reveal the obscured files again. Your best option is to recreate the container without the mount.

Considerations and Constraints

- Bind mounts have write access to files on the host by default. Use the `readonly` or `ro` option to make the bind mount read-only.
- Bind mounts are tied to the Docker daemon host, not the client.
- If using Docker Desktop, the daemon runs inside a Linux VM, with mechanisms to handle bind mounts transparently.
- Bind mounts rely on the host machine's filesystem structure. Containers may fail if run on a different host without the same directory structure.

Syntax

To create a bind mount, you can use either the `--mount` or `--volume` flag:

```
docker run --mount type=bind,src=<host-path>,dst=<container-path>
docker run --volume <host-path>:<container-path>
```

The `--mount` flag is generally preferred as it is more explicit and supports all available options.

Syntax for Docker Compose

```
volumes:
  - /path/to/your/hostpath:/path/inside/the/container:ro
```

> See more about binding mounts on the Official Docker Docs.

Revision #4
Created 18 September 2024 08:20:47 by aeoneros
Updated 12 January 2025 12:10:28 by aeoneros