

Title: Docker: Is it possible to share data between 2 containers without a volume?

Post Body:

I have 2 containers: `web` and `nginx`. When I build `web` container, static assets for frontend are generated within the container.

Now, I want to share those assets between `web` and `nginx` without using a volume on the host machine. Otherwise, I'll have to build those static assets on the host side and then include as a volume into the `web` container and share it with `nginx` container. This is undesirable from my build system's standpoint.

Is there a way to build static assets in the `web` container and then share them with `nginx`?

Accepted Answer:

Otherwise, I'll have to build those static assets on the host side and then include as a volume into the `web` container and share it with `nginx` container.

This statement seems incorrect.

If the static assets are generated as part of the build process, then just mount a volume on top of that directory at runtime. Docker will take care of copying the underlying content into the volume, after which you can access it in your `nginx` container using `--volumes-from`.

For example, if I start with this `Dockerfile` for my `web` container:

```
FROM alpine
RUN apk add --update darkhttpd COPY assets /assets CMD ['darkhttpd', '/assets']
```

I now have a directory `/assets` that contains my static assets. If I run this image as:

```
docker run -v /assets --name web web
```

Then `/assets` will (a) be a volume and (b) contain the contents of the `/assets` directory.

Now you can start an `nginx` container and share this data with it:

```
docker run --volumes-from web nginx
```

The `nginx` container will have a `/assets` directory that contains your static assets.

I've put together a small example [here](#).

Highest Rated Answer: None