Should Dockerfiles specify an entry point when using docker compose?

Asked 1 year, 10 months ago Modified 1 year, 10 months ago Viewed 2k times

0

I am migrating some web-apps to be managed via docker compose

It seems the docker-compose.yaml has a section for the container entry-point.

However, my individual docker files have an ENTRYPOINT themselves... should I remove this from the Dockerfiles? Does the entry-point in docker-compose override the Docker one?

docker docker-compose

Share Improve this question Follow

asked Nov 3, 2022 at 11:43
MYK
2,687 ● 3 ● 15 ● 56

docker-compose entrypoint overrides the docker image's entrypoint. You do not have to use the docker-compose entrypoint. – Daniel Juravski Nov 3, 2022 at 11:58

No and yes are the answers to the two questions. The exact term to use here is that you **may** define an entrypoint to override the default one (as you would do directly on the docker command of needed) – Zeitounator Nov 3, 2022 at 11:58

1 Answer

Sorted by: Highest score (default)

1

You usually shouldn't specify `entrypoint:` or `command:` in a Compose file. Prefer specifying these in a Dockerfile. The one big exception is if you have a container that can do multiple things (for example, it can be both a Web server and a queue worker, with the same code) and you need to tell it with a `command:` to do not-the-default thing.

I'd suggest a typical setup like:

```
# docker-compose.yml
version: '3.8'
services:
  app:
    build: .
    # with neither entrypoint: nor command:
```

```
# Dockerfile
FROM ...
WORKDIR /app
COPY ...
RUN ...

# ENTRYPOINT ["./entrypoint-wrapper.sh"]
CMD ["./my_app"]
```

Compose `entrypoint:` overrides the Dockerfile `ENTRYPOINT` and resets the `CMD`. Compose `command:` overrides the Dockerfile `CMD`.

In the Dockerfile both `ENTRYPOINT` and `CMD` are optional. If your base image already includes a correct command setup (`nginx`, `php:fpm`) then you can safely skip both.

It's otherwise somewhat a matter of style whether to use `CMD` or `ENTRYPOINT` in your Dockerfile. I prefer `CMD` for two reasons: it's easier to replace in a `docker run ... image-name alternate command` invocation, and there's a pattern of using `ENTRYPOINT` as a wrapper script to do first-time setup and then launch the `CMD` with `exec "$@"`. If you have a JSON-array-syntax `ENTRYPOINT` then you can pass additional command-line arguments to it as `docker run ... image-name --option`. Both setups are commonplace.

The thing you shouldn't do is put an interpreter in `ENTRYPOINT` and a script name in `CMD`. I only ever see this in Python, but `ENTRYPOINT ["python3"]` is wrong. On the one hand this is hard to override in the same way `ENTRYPOINT` is in general, and on the other neither normal command override format works (you still have to repeat the script name if you want to run the same script with different options).

Share Improve this answer Follow

answered Nov 3, 2022 at 13:07

David Maze

152k ● 38 ● 208 ● 259