

Title: Wrong permissions in volume in Docker container

Post Body:

I run Docker 1.8.1 in OSX 10.11 via an local docker-machine VM.

I have the following docker-compose.yml:

```
web:      build: docker/web      ports:      - 80:80      - 8080:8080      volumes:      - $PWD/cms:/srv/cms
```

My Dockerfile looks like this:

```
FROM alpine # install nginx and php RUN apk add --update \      nginx \      php \      php-fpm \      php-pdo \      php-json \
```

When I run everything with `docker-compose up` everything works fine, my volumes are mounted at the correct place.

But the permissions in the mounted folder `/srv/cms` look wrong. The user is '1000' and the group is '50' in the container. The webserver could not create any files in this folder, because it runs with the user 'root'.

Accepted Answer:

1) **General idea:** Docker it is not Vagrant. It is wrong to put two different services into one container! Split it into two different images and link them together. Don't do this shitty image.

Check and follow https://docs.docker.com/engine/userguide/eng-image/dockerfile_best-practices/

- Avoid installing unnecessary packages
- Run only one process per container
- Minimize the number of layers

If you do it:

- you will remove your supervisor
- your can decrease numbers of layers

It should be something like (example):

```
FROM alpine RUN apk add --update \      wget \      curl RUN apk add --update \      php \      php-fpm \      php-pdo \      php-j
```

For nginx it is enough to use default image and mount configs. docker-compose file like:

```
nginx:      image: nginx      container_name: site.dev      volumes:      - ./myconfl.conf:/etc/nginx/conf.d/myconfl.conf      - ./myconf
```

2) Add `RUN usermod -u 1000 www-data` into Dockerfile for php container, it will fix problem with permission.

Highest Rated Answer:

For alpine version you need to use:

```
RUN apk add shadow && usermod -u 1000 www-data && groupmod -g 1000 www-data
```