

Title: Docker creating multiple images

Post Body:

I am new to Docker and find that there are numerous images that are getting created (as seen in `sudo docker images`) and found somewhere in stackoverflow to periodically run `sudo docker rmi $(sudo docker images -q)` to remove all images. Why so many images get created? is there something wrong in my configuration?

docker-compose.yml

```
nginx:    build: ./nginx    restart: always    ports:      - '80:80'      - '443:443'    volumes:      - /etc/letsencrypt/:/etc/letse
```

The nginx dockerfile is

```
FROM nginx:alpine COPY nginx.conf /etc/nginx/conf.d/default.conf
```

The nodejs dockerfile is

```
FROM node:9.3.0-alpine WORKDIR /usr/src/app COPY package*.json /usr/src/app/ RUN npm install --only=production COPY . /usr
```

The website/app works fine. Except that periodically, I am removing all containers, images and then run: `sudo docker-compose up --build -d`.

Accepted Answer:

Images are immutable, so any change you make results in a new image being created. Since your compose file specifies the build command, it will rerun the build command when you start the containers. If any files you are including with a `COPY` or `ADD` change, then the existing image cache is no longer used and it will build a new image without deleting the old image.

Note, I'd recommend naming your image in the compose file so it's clear which image is being rebuilt. And you can watch the compose build output to the the first step that doesn't report using the cache to see what is changing. If I was to guess, the line that breaks your cache and causes a new image is this one in nodejs:

```
COPY . /usr/src/app
```

If the files being changed and causing the rebuild are not needed in your container, then use a `.dockerignore` file to exclude the unnecessary files.

Highest Rated Answer:

I had the same problem when I was building my Dockerfile.

I found the solution, use this command to build your file :

```
`docker build --rm -t <tag> .`
```

The option `--rm` removes intermediate containers after a successful build.