

Title: Rename a project by keeping containers

Post Body:

I decided to change the project name of a docker composition:

```
$ docker-compose -p old_name up -d # Before Starting old_name_web_1 $ docker-compose -p new_name up -d # After Creating new_name
```

But I don't wanted to delete my containers, so I renamed them:

```
$ docker rename old_name_web_1 new_name_web_1 ...
```

I thought docker-compose was based on container names, but it does not seem to be the case:

```
$ docker-compose -p new_name up -d
```

```
ERROR: for web Cannot create container for service web: Conflict. The name '/new_name_web_1' is already in use by container 4930deaabb[...]. You have to remove (or rename) that container to be able to reuse that name. ERROR: Encountered errors while bringing up the project.
```

How can I relink my old containers to the new composition ?

Accepted Answer: None

Highest Rated Answer:

It looks like you are using one of the newer versions of docker compose which tracks containers by labels assigned to them rather than by their names. That is why renaming the container didn't work.

Updating labels

You can check container's labels through the `docker inspect` command.

```
$ docker inspect --format='{{json .Config.Labels }}' container_name
```

The project name is the value of the 'com.docker.compose.project' label.

Moving an existing container to a new project is as easy as changing the value of that label. However it is not yet supported by Docker CLI. There is [an open issue requesting that feature](#).

Workaround

It still can be achieved by directly editing the configuration file of that particular container. There you will find labels currently assigned to that container.

```
$ nano /var/lib/docker/containers/$container_id/config.v2.json
```

Assign the new project name to the 'com.docker.compose.project' label and save the file. Next you have to restart the daemon. Otherwise the changes will not be visible to docker.

```
$ systemctl daemon-reload
```