

# Get id of user on host, from a docker

Asked 4 years, 4 months ago   Modified 4 years, 4 months ago   Viewed 2k times

From inside a container, I would like to get the id of a user on the host machine (what the command `id -u username` would output, from the host).

-1

Is there a way to accomplish this?

I thought I could mount `/etc/passwd` in the container and `grep` inside, but unfortunately the users are not listed in this file on our server (possibly related to the LDAP authentication mechanism?).



Thanks

`docker` `host` `uid`

Share   Improve this question   Follow

asked Apr 9, 2020 at 16:08



**bruno**  
69 ● 5

obviously not possible, what are you trying to achieve? *you could mount `/etc/passwd` as a volume file, with the risks which come with it* – [Lawrence Cherone](#) Apr 9, 2020 at 16:15

Here is the use case: I have a (containerized) Jupyter Hub server that spawns (containerized) single-user Jupyterlab notebook servers. The home folder of the user that logs in Jupyter Hub is mounted into the notebook server's container, so the latter should ideally run with the user's id to have full permissions on the folder. – [bruno](#) Apr 9, 2020 at 18:38

## 2 Answers

Sorted by: Highest score (default)

I ended up solving this by mounting host folder `/home` on my container, and getting the id of the owner of user's home dir `/home/<user>`.

1

Share   Improve this answer   Follow



answered Apr 17, 2020 at 8:22



**bruno**  
69 ● 5



0



There's no way to get information about host users from inside a container. A design goal of Docker is that the host and containers are isolated from each other. A container has no concept of a host user; from the Docker daemon point of view, Docker doesn't even really know which user requested that a container be launched.

(This is doubly true if your host authentication system is something more complicated like an LDAP setup: a container simply may not have the tools or credentials required to query it, and the isolation means there's no way to somehow delegate to the host.)

If a principal goal of your application is to interact with host users, or the host filesystem, or you otherwise actively don't want Docker's isolation features, it's better to run your program outside of Docker.

Share Improve this answer Follow

answered Apr 9, 2020 at 17:46



David Maze

152k ● 38 ● 208 ● 259

---

Thanks for your answer. I described the use case above : I have a containerized app (Jupyter Hub), that spawns a container (Jupyter notebook server) every time a user logs in. The app should pass the id of the user to the container it spawns, so that the latter can have access to the user home folder. – [bruno](#) Apr 9, 2020 at 18:43

---