docker: get rid of images and containers

MARKDOWN I moved the blabbering behind the important stuff: **how to get rid of the docker images/containers** ``` docker system prune # dangling containers docker system prune -a # dangling,stopped containers and all unused images docker rmi $(docker images -a -q) # all images docker rm $(docker ps -a -f status=exited -q) # all exited containers ``` [Further Info](https://www.digitalocean.com/community/tutorials/how-to-remove-docker-images-containers-and-
Atm I'm working lots with docker to have my one-click deployer for [Urho3D](https://urho3d.github.io/). The main-developers have already made insane good work. On their [docker images](https://hub.docker.com/u/urho3d). The current dockerized-build-toolchain would let you compile Urho3D on your host with which you then can compile your project. I want to have Urho3D-Libs in the docker-images so that I can throw my (project)[https://github.com/dertom95/urho3d-minimal-new-project] against it, and it will just spit out the result for windows,linux,web,android. The first three were actually no real problem and very straight forward. Again, thx to the incredible work that is already done. Android on the other hand was/is a real struggle. I was using android back in the good old ant,eclipse-days. Now with gradle and kotlin....I just can say. WTF? Looks like gradle can do a lot, but it is so funky hard to understand what's going on just be reading. Urho3D actually could already build the engline and it's samples out of the box. But replacing this with your own project isn't as straight forward, as it might sound. Especially if you want it to be a one click-solution. I'm on a good way, though. But **I burned a complete day for this** Building the android-project with all ABIs simply let's you harddisc explode :D [My docker hub with urho3d-images](https://hub.docker.com/u/dertom95)

Published by Thomas Trocha
Sat Jan 11 02:53:00 CET 2020
http://thomas.trocha.com:80/pebble/docker-get-rid-of-images-and-containers