**Docker Commands You Need to Know**

If you get permission denied error, then you’ll need to use sudo with the commands below.

If you don’t like to type sudo each time, you may add your user to the docker group in your Linux system.

1. To find out which version of docker is installed in your system:  
     
    docker --version
2. To find out detailed docker versioning information in your system for both client and server:  
     
    docker version
3. To see a list of docker images on your system  
     
    docker image ls  
    docker images  
    docker images -a
4. To pull (download) an official Docker Hub image:  
     
    docker image pull <repository>:<tag>
5. To pull (download) an unofficial Docker Hub image:  
     
    docker image pull <username>/<repository>:<tag>
6. To permanently remove a docker image from your system  
     
    docker image rm <image>
7. To start a docker container in interactive mode  
     
    docker container run -it <username>/<repository>:<tag> sh  
    docker run -it <username>/<repository> sh  
    docker run -it <image> sh
8. To exit a docker container from interactive mode and stop the container  
     
    docker run -it <image> sh

exit

1. To detach the docker host terminal from the container (keep the container running in the background)  
     
    docker run -it <image> sh  
    Ctrl P + Q
2. To start a docker container in detached mode (running in the background)  
     
    docker container run -d <username>/<repository>:<tag>  
    docker run -d <username>/<repository>  
    docker run -d <image>
3. To get a list of docker containers  
     
    docker container ls  
    docker ps  
    docker ps -a
4. To attach to a running container (get access to shell from host terminal)  
     
    docker exec -it <container> sh
5. To stop a docker container. Stopping a container doesn’t result in loss of data stored inside the container.  
     
    docker container stop <container>
6. To start a docker container   
     
    docker container start <container>
7. To permanently remove a docker container. This will result in the loss of data held stored inside the container.  
     
    docker container rm <container>
8. Create a docker container in detached mode, give it a custom name, and publish the port  
     
    docker container run -d --name <container name> -p <port:listener> <image>
9. Create a docker container in interactive mode, give it a custom name, and publish the port  
     
    docker container run -it --name <container name> -p <port:listener> <image> sh
10. Dockerfile  
    1. FROM <image>:<tag>
       1. This is the base image used to build our image
    2. LABEL <key>=<value>
       1. The LABEL instruction adds metadata to an image. A LABEL is a key-value pair.
    3. COPY <src> <dest>
       1. The COPY instruction copies new files or directories from <src> and adds them to the filesystem of the container at the path <dest>.
    4. RUN
       1. The RUN instruction will execute any commands in a new layer on top of the current image and commit the results. The resulting committed image will be used for the next step in the Dockerfile.
    5. WORKDIR </path>
       1. The WORKDIR instruction sets the working directory for any RUN, CMD, ENTRYPOINT, COPY and ADD instructions that follow it in the Dockerfile. If the WORKDIR doesn’t exist, it will be created even if it’s not used in any subsequent Dockerfile instruction.
    6. ENV <key>=<value>
       1. The ENV instruction sets the environment variable <key> to the value <value>.
    7. EXPOSE <port>
       1. The EXPOSE instruction informs Docker that the container listens on the specified network ports at runtime.
    8. ENTRYPOINT
       1. An ENTRYPOINT allows you to configure a container that will run as an executable.
11. To show the history of an image  
      
     docker history <image>
12. Return low-level information on Docker objects  
      
     docker inspect <object>