Docker Client (Docker CLI): way to connecting to the docker server(called docker hub).

Docker server (Docker Daemon) : responsible to creating and managing images, running containers or doing everything in docker.

Container:

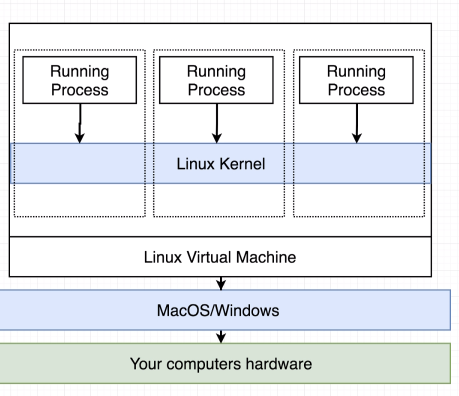
1. An instance on an image runs a program locally on a machine.
2. In other terms, container is a running process with all the essentials (for ex, hardware necessities) it needs.

Commands:

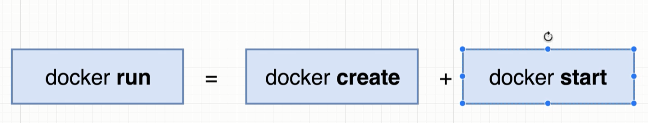
1. Docker ps -> to see all of the currently running containers
2. docker ps --all -> to see the all created containers on the server
3. docker start/stop <iD> - to run the container
4. docker create hello-world -> to create a hello world container with it individual container ID
5. docker run -a < container ID> -> to run a created container using it individual ID.  
   NOTE: -a is used to attach the output of the container with the terminal
6. docker run -d <image-name> : -d is used to run the docker container in background.
7. docker system prune ->to remove the history of all stopped container along with their local copies which occupy some spaces.
8. Docker logs <container-ID> -> to check the logs of a container when it ran or running.
9. Docker stop <container-ID> - stops the container using the SIGTERM signal, allows container to stop properly, recommended one. This command wait for 10 Sec for container to stop, if it doesn’t stop it then triggers docker kill command to kill the container.
10. Docker kill <cont-ID> -> kills the process using the SIGKILL signal, stops the container abruptly.
11. Docker exec -it <container-ID> <command> -> to type commands which you want to run inside a container.  
    NOTE: -it command gives the capability to execute the commands inside a container. -it is a combination of two commands -i -t which means input the text.
12. Docker run -it <container-name> <command> -> to run the commands while creating a container. For example: docker run -it busybox sh
13. Docker build -t <tag name>(for example, your docker ID/project name, in my case- jasbeer1988/testproject). Then we can use the tagged name of the image to run the container.

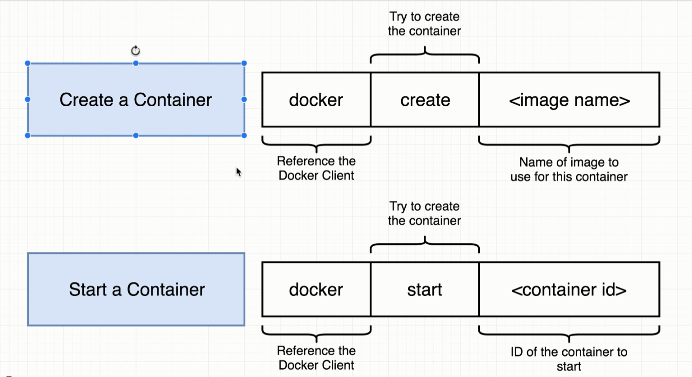
How docker run on Windows/Mac environment:

1. When you start the docker setup it will create a linux virtual machine on your machine and then perform all steps in it.



Docker run command description:



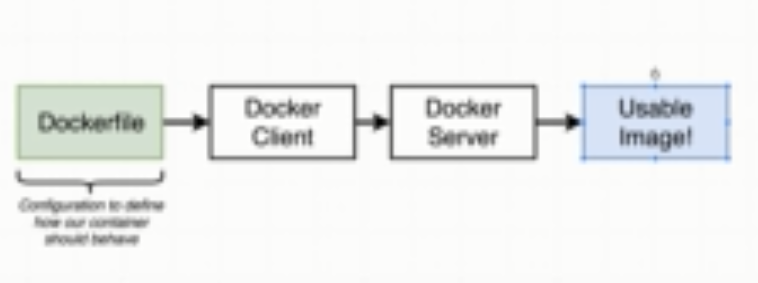


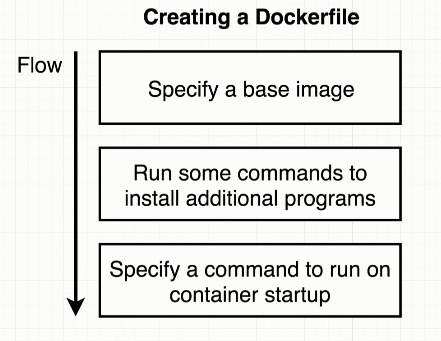
Creating a container means creating a FS snapshot locally on your machine.

Running a container means executing the commands the container is created for.

**Alpine version of any image means, it would be as small as possible with very limited installed softwares. Every repository on docker hub would have an alpine version.**

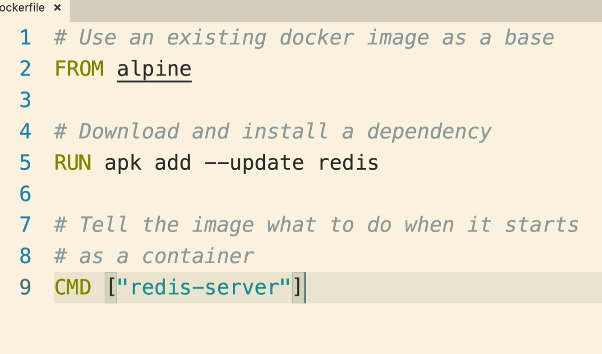
**Creating a Docker image:**





Generic example:

1. Creating docker file:

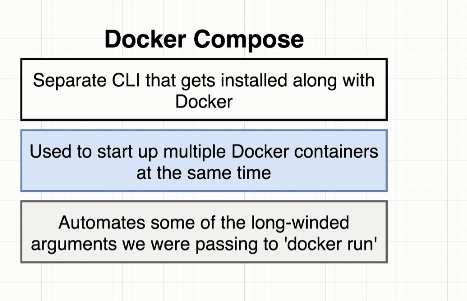


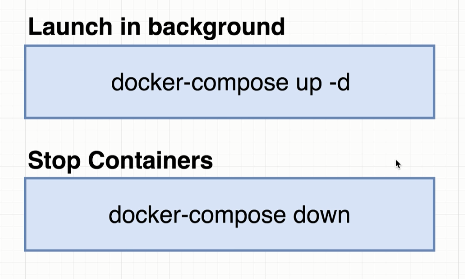
1. Building an image: To build the image use command “docker build .” inside the project directory.  
     
   
2. While creating a docker image, every step (base image, run or command) would create a temp container and out of the image of previous step and use it to build its image, which would be supplied to the next step.

**Docker Compose**- It is a kind of CLI tool which allows us to issue multiple commands more easily. Also, it is used to connect multiple containers. It uses docker-compose.yml file to perform its task.

If there is any change in any of the file in the image, use the build option while running docker-compose.

Docker-compose up --build





-d option is used to run the containers in background.

**Automatically restarting a container after a crash:**  
