**Containers**

* For long time, big web-scale organizations like Google have been using container technologies to overcome the shortcomings of VM Model
* Containers on the single host share the host OS

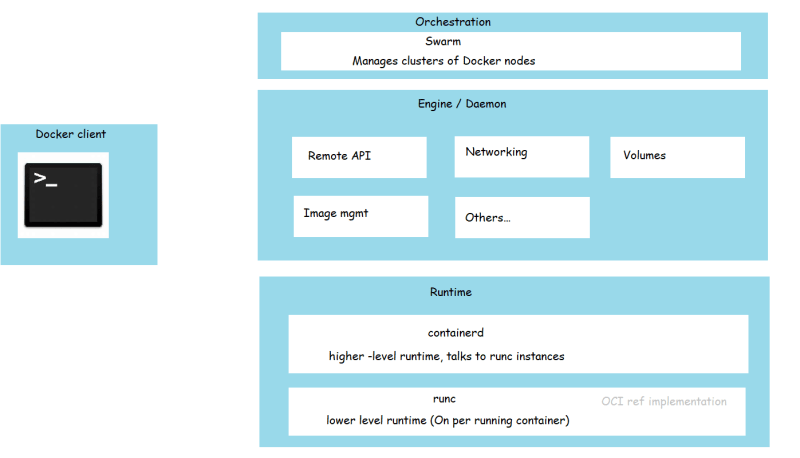
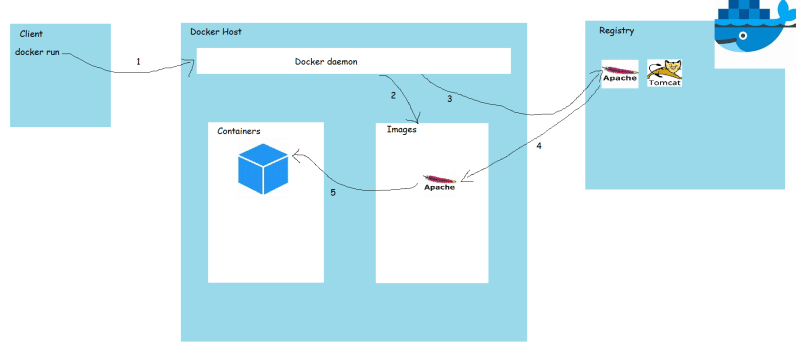
**Linux Containers**

* Containers in Linux are supported natively but creating containers remained complex and out of reach for most of the organizations.
* Docker came along & simplified the container creation & management.
* Example:
* BSD Jails & Solaris Zones are some other well-known examples of Unix type container technologies

**Windows Container’s**

* Over the past few years, Microsoft Corp. has worked extremely hard to bring Docker and container technologies to the windows platform
* Windows Containers are available on the Windows Desktop & Windows Server platforms (Certain Versions of Windows 10 and later and Windows Server 2016 & Later)
* The core windows kernel has been changed to implement containers & this is collectively referred as Windows Containers

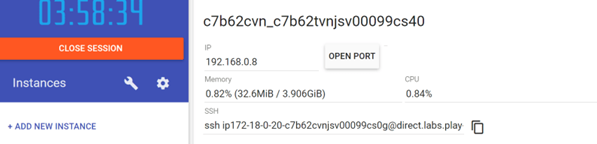
**The Docker Technology**

* There are three things to be aware of when we are referring to docker as a technology
* The runtime
* The daemon/Engine
* The orchestrator  
  
* To interact with docker we have the docker client
* Workflow of Container Creation  
  
* When the user executes the command to create container docker container run -d -P httpd on the newly installed docker host
* docker client forwards the request to docker daemon
* docker daemon checks for the presence of images in the local repository. In this case the local images didn’t have the image httpd
* Docker forwards the request to the configured registry. The default registry that is configured is docker hub(<https://hub.docker.com/>) The image is searched in the configured registry and if not found error is shown to the user
* If the image is found then it is downloaded from docker registry to local images repository
* Using the docker image the container is created (resources will be allocated)
* Now let’s see what happens the user executed the command docker container run -d -P httpd again
* docker client forwards the request to docker daemon
* docker daemon checks for the presence of image httpd in local registry & since it is already downloaded in the 4 step above.
* docker daemon creates a container with the existing image.

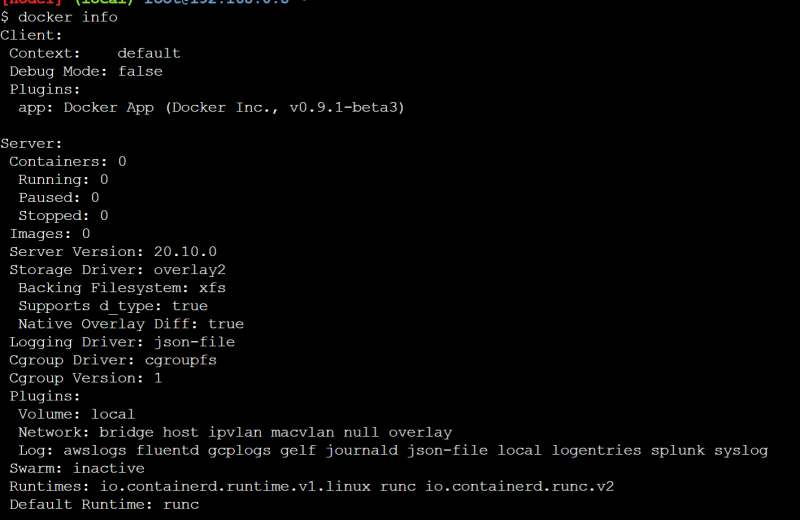
**Responsibility of a DevOps Engineer**

* As per the understanding of docker which we have so far, We need to create containers which run the applications developed by our organization. The first responsibility of DevOps Engineer to
* Create a Docker Image which has your application preconfigured.
* Manage multiple versions of your Docker Image
* Store the Images in Organization approved Registry.

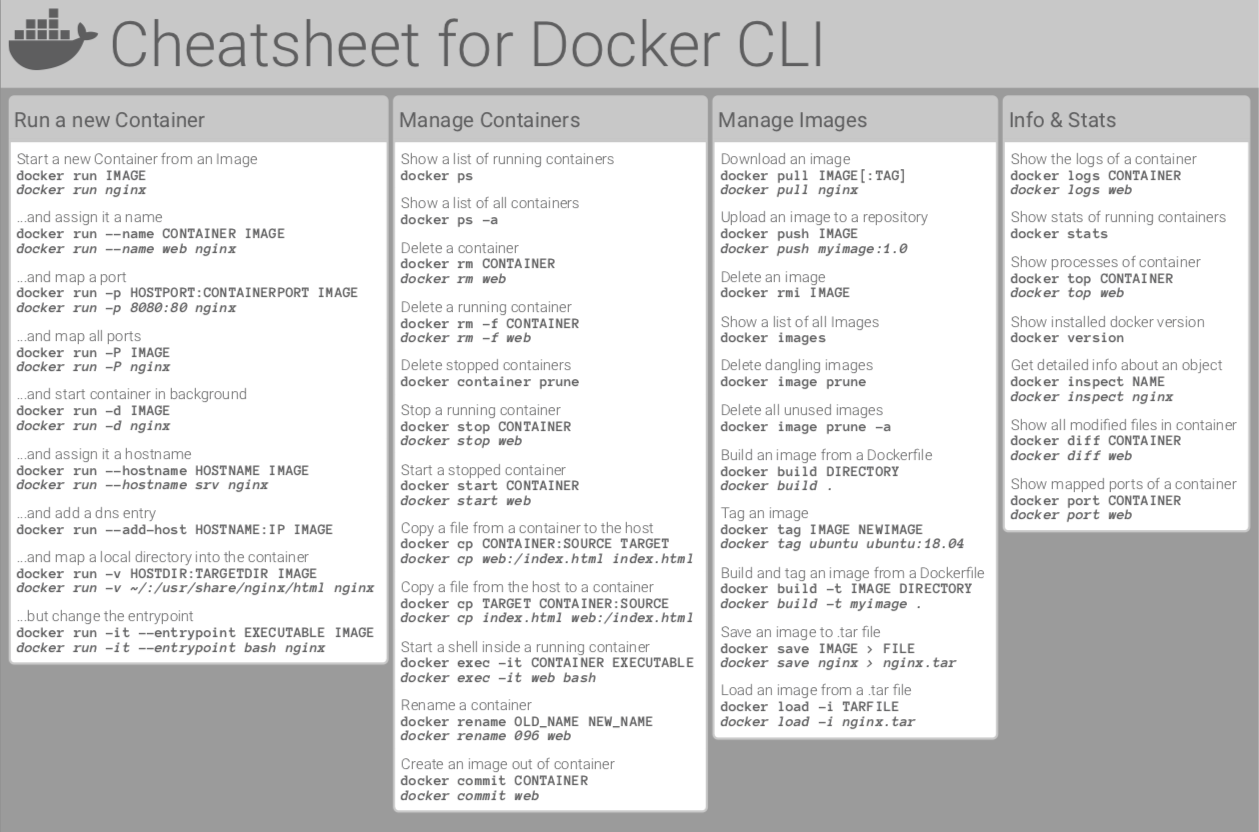
**Docker Playground** – PAAS based solution for running docker commands/files(Login with DockerHub credentials)



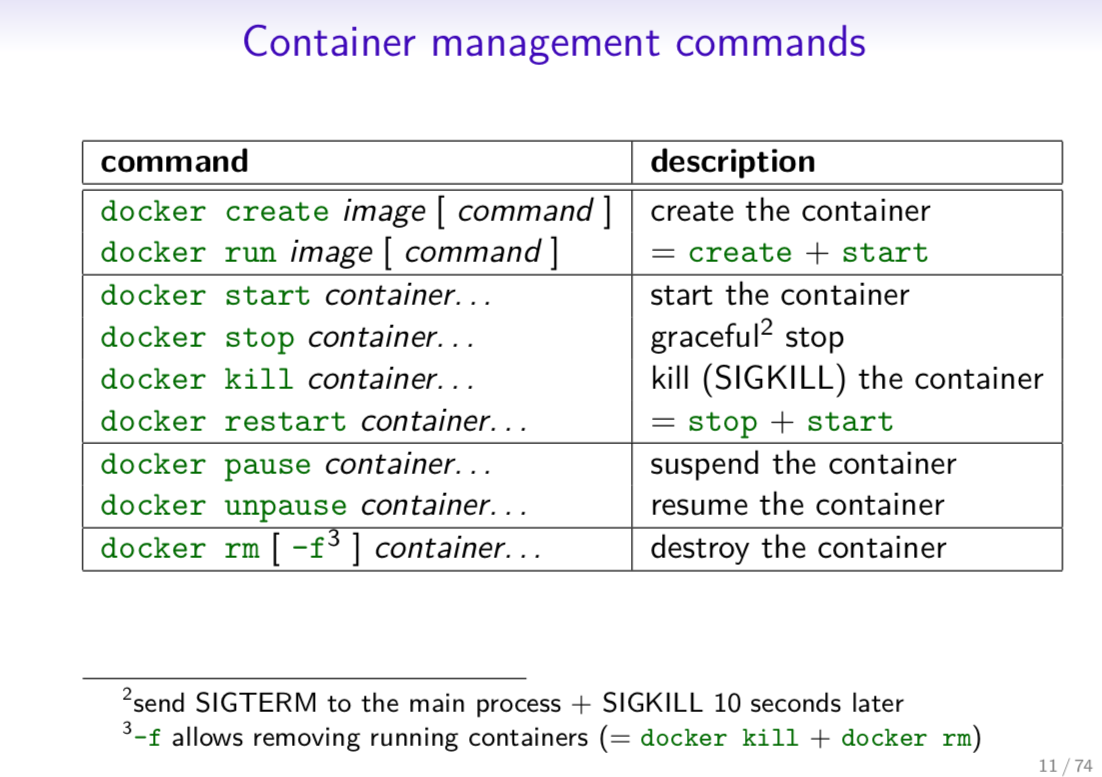
If you want a full screen terminal view ALT + ENTER and to come-back to normal state ALT + ENTER



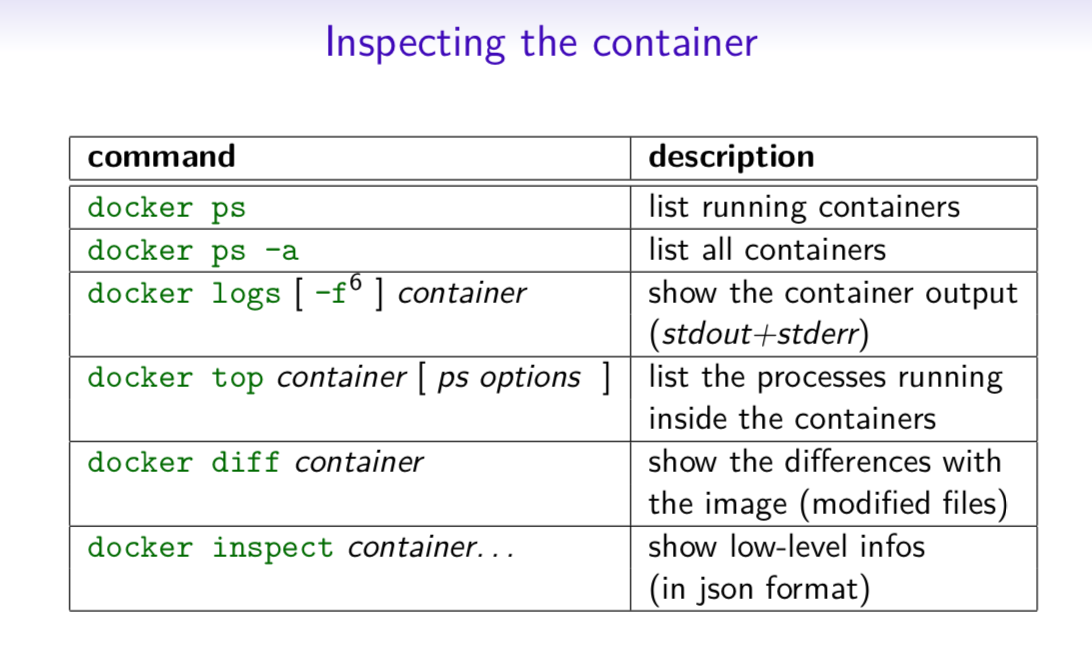
**Commands of Docker**



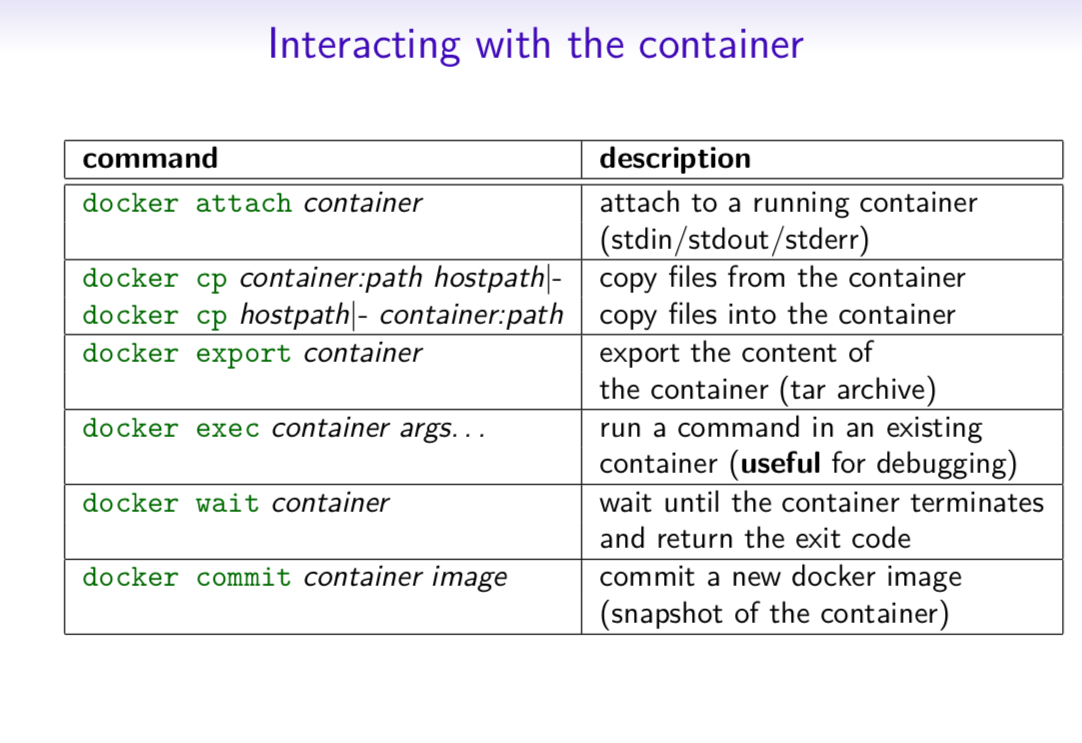
## Container Management CLIs



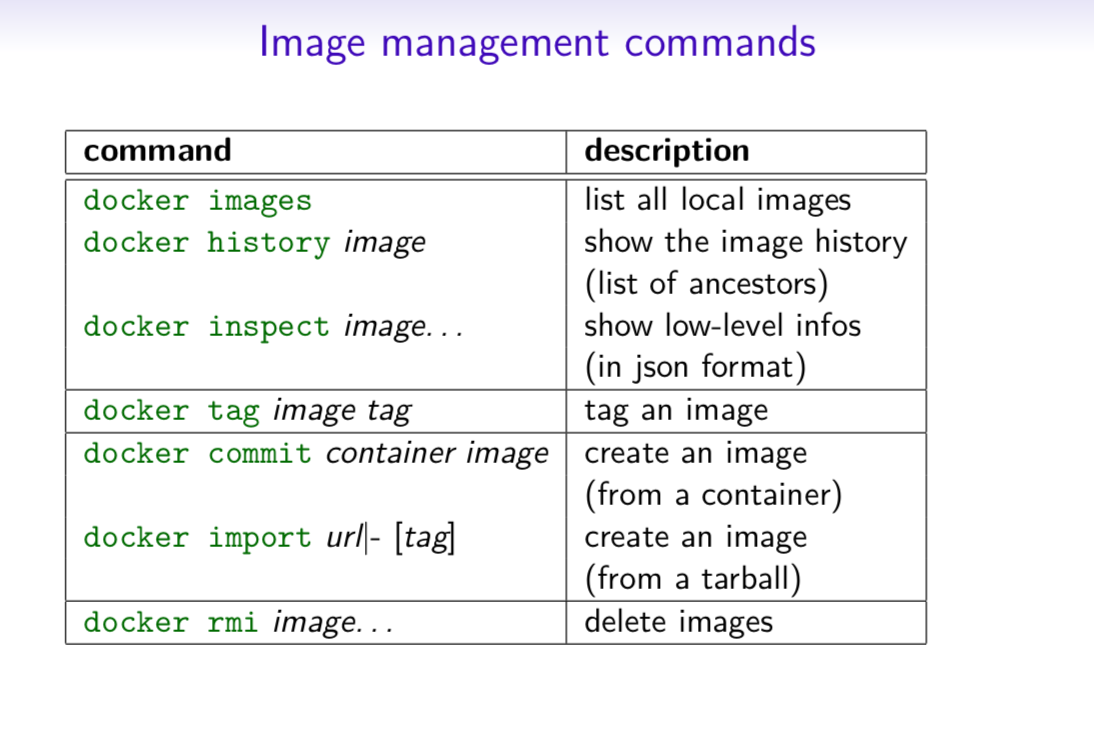
## Inspecting The Container



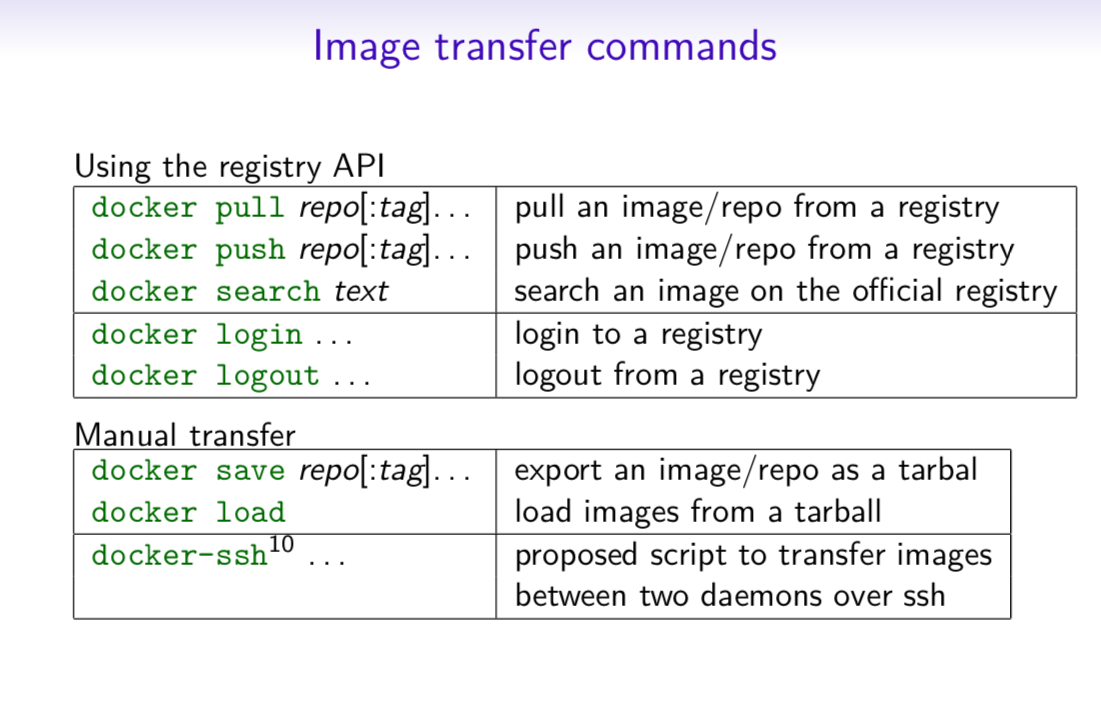
## Interacting with Container



## Image Management Commands



## Image Transfer Commands



## Builder Main Commands

