Docker: Containerization

Ganesh Palnitkar

Agenda

- What is Docker?
- Docker Architecture
- Installing Docker
- Dockerfile
- Docker Images
- Docker and Microservices

What is Docker

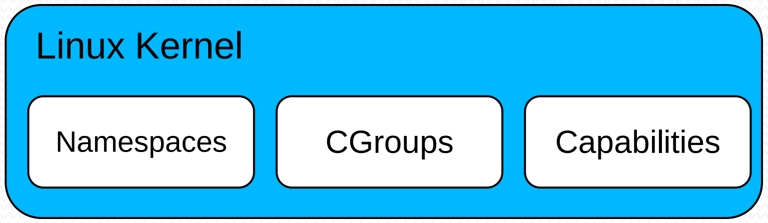
Docker separates applications from infrastructure using the container technology. This is similar to the way VMs separate operating systems from the hardware. Using Docker containers, you can deploy, replicate, move, and back up a workload even more quickly and easily than you can do so using virtual machines. Docker container are highly **scalable** (can be expanded rapidly), **portable** (Dockerized application are extremely portable). Docker uses **google** ‘**Go**’ programming language for Docker development.

- **Docker Registries:**

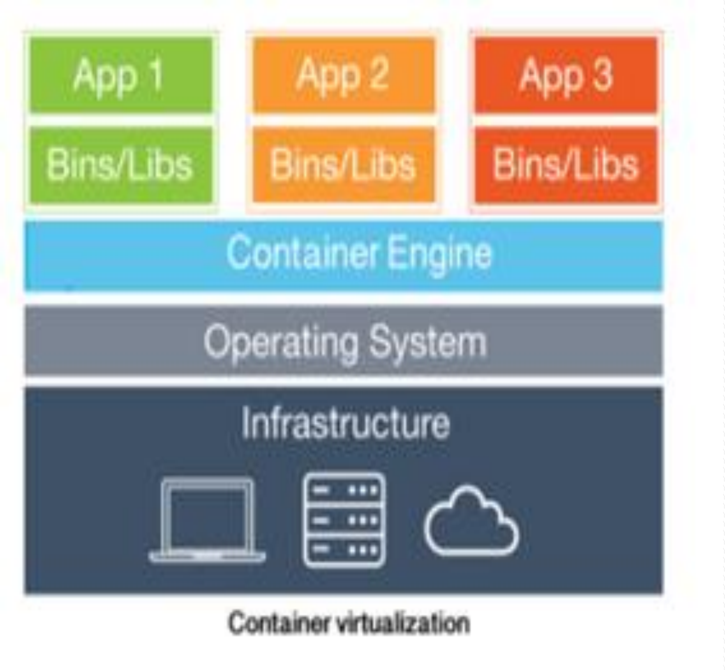
A registry is a storage and content delivery system, holding named Docker images, available in different tagged versions. Users interact with a registry by using Docker push and pull commands.
One can create and share Docker image on ‘Docker hub’ for sharing with co-workers, customers etc.

```
$ docker pull
$ docker run <container name>
```

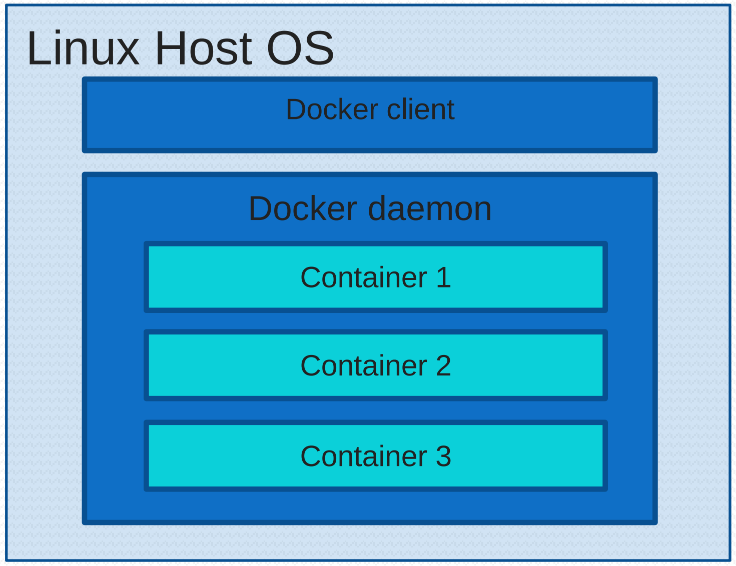
- Docker formerly known as **dotCloud**



Docker Architecture



Container virtualization



- Container are very light weight that shared same OS kernel, as compared to a VM.
- Container still provides isolated user space like a VM. This can also be referred as container virtualization or OS level virtualization.
- Container share Host OS resources like, RAM, CPU, Storage, network, etc.
- Each container has its own set of file system, like /etc, /var, /opt, etc.
- Container is created from a Docker image that can be customized. One create own Docker image using Dockerfile suitable to your requirement.
- Containerization is based on the Linux Kernel Namespaces.
- Docker uses ‘**libcontainer**’ as the execution driver instead of Linux LXC that was used earlier.

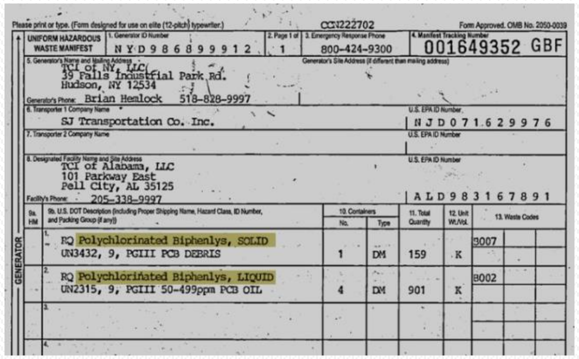
Docker High Level picture

Docker Engine

(Shipping Yard)

Docker Images

(Shipping Manifests)



Please print or type. (Form designed for use on elite (12-pitch) typewriter.) CCN222702 Form Approved. OMB No. 2050-0039

UNIFORM HAZARDOUS WASTE MANIFEST	1. Generator ID Number	2. Page 1 of	3. Emergency Response Phone	4. Manifest Tracking Number
	N Y D 9 8 6 8 9 9 9 1 2	1	800-424-9300	001649352 GBF

5. Generator's Name and Mailing Address
TCI of NY, LLC
39 Falls Industrial Park Rd.
Hudson, NY 12534
Generator's Phone: Brian Hemlock 518-828-9997
Generator's Site Address (if different than mailing address)

6. Transporter 1 Company Name: SJ Transportation Co. Inc. — U.S. EPA ID Number: N J D 0 7 1 6 2 9 9 7 6

7. Transporter 2 Company Name — U.S. EPA ID Number

8. Designated Facility Name and Site Address
TCI of Alabama, LLC
101 Parkway East
Pell City, AL 35125
Facility's Phone: 205-338-9997
U.S. EPA ID Number: A L D 9 8 3 1 6 7 8 9 1

GENERATOR

9a. HM	9b. U.S. DOT Description (including Proper Shipping Name, Hazard Class, ID Number, and Packing Group (if any))	10. Containers No.	10. Containers Type	11. Total Quantity	12. Unit Wt./Vol.	13. Waste Codes
	1. RQ Polychlorinated Biphenlys, SOLID UN3432, 9, PGIII PCB DEBRIS	1	DM	159	K	B007
	2. RQ Polychlorinated Biphenlys, LIQUID UN2315, 9, PGIII 50-499ppm PCB OIL	4	DM	901	K	B002
	3.					
	4.					

Docker containers

(Shipping Containers)

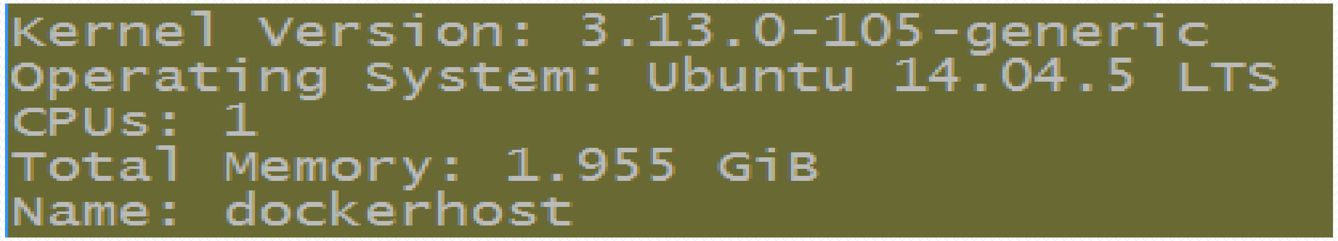
While using a container from public repository, utmost care should be taken of using only trusted container as there can be malicious code in the containers.

Docker engine is also known as Docker **Daemon** or Docker **Runtime**…..

Installing Docker

Docker can be installed on **Windows** as well as Linux flavor machine.

- Currently Docker can be installed only on Windows7, Windows8 and Windows10. For Win7 and Win8, one need Docker toolbox to work.
- Download Docker for windows from the URL,
- https://docs.docker.com/docker-for-windows/install
- Docker on windows will need the **hyper-v** to be enabled on windows. If you have any other virtualization platform installed on your windows machine, like VirtualBox, it will have to be disabled in order to use Docker on Windows.
- Once we download the 'InstallDocker.msi' file, install it by following standard installation instructions.



```
Kernel Version: 3.13.0-105-generic
Operating System: Ubuntu 14.04.5 LTS
CPUs: 1
Total Memory: 1.955 GiB
Name: dockerhost
```

Install Docker-ce

Installing Docker-ce is advised as against installing docker.io

```
$ sudo apt-get update
$ sudo apt-get install \
    linux-image-extra-$(uname -r) \
    linux-image-extra-virtual
$ sudo apt-get update
$ sudo apt-get install \
    apt-transport-https \
    ca-certificates \
    curl \
    software-properties-common
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key
add -
$ sudo add-apt-repository \
   "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
   $(lsb_release -cs) \
   stable"
$ sudo apt-get update
$ sudo apt-get install docker-ce
```

Docker-ce on Centos

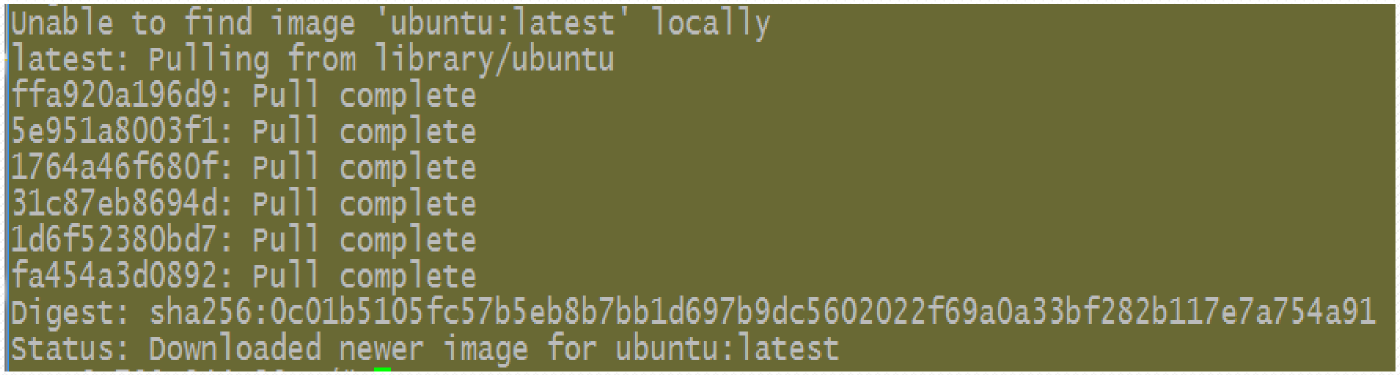
```
$ sudo yum install -y yum-utils \
  device-mapper-persistent-data \
  lvm2
$ sudo yum-config-manager \
    --add-repo \
    https://download.docker.com/linux/centos/docker-ce.repo
$ sudo yum install docker-ce
$ sudo systemctl start docker
```

Docker further

Now to use a docker **image** that is part of the docker repository, use below command,

Docker promotes the **standardization** ideology and so that a container created on a local machine can easily be **ported** to EC2, Azure or any other **cloud** environment.

`$ docker run -it ubuntu /bin/bash` ---- this tells the docker to run in interactive mode and assign a terminal, use the **Ubuntu image** for the container to create and then run the /bin/bash command once the container is ready.



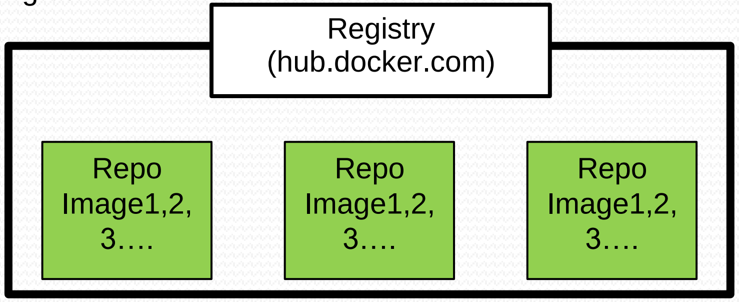
```
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
ffa920a196d9: Pull complete
5e951a8003f1: Pull complete
1764a46f680f: Pull complete
31c87eb8694d: Pull complete
1d6f52380bd7: Pull complete
fa454a3d0892: Pull complete
Digest: sha256:0c01b5105fc57b5eb8b7bb1d697b9dc5602022f69a0a33bf282b117e7a754a91
Status: Downloaded newer image for ubuntu:latest
```

`$ docker ps -a` **---** this will show all processes running on the host.

`$ docker start <container code>` --- start the container

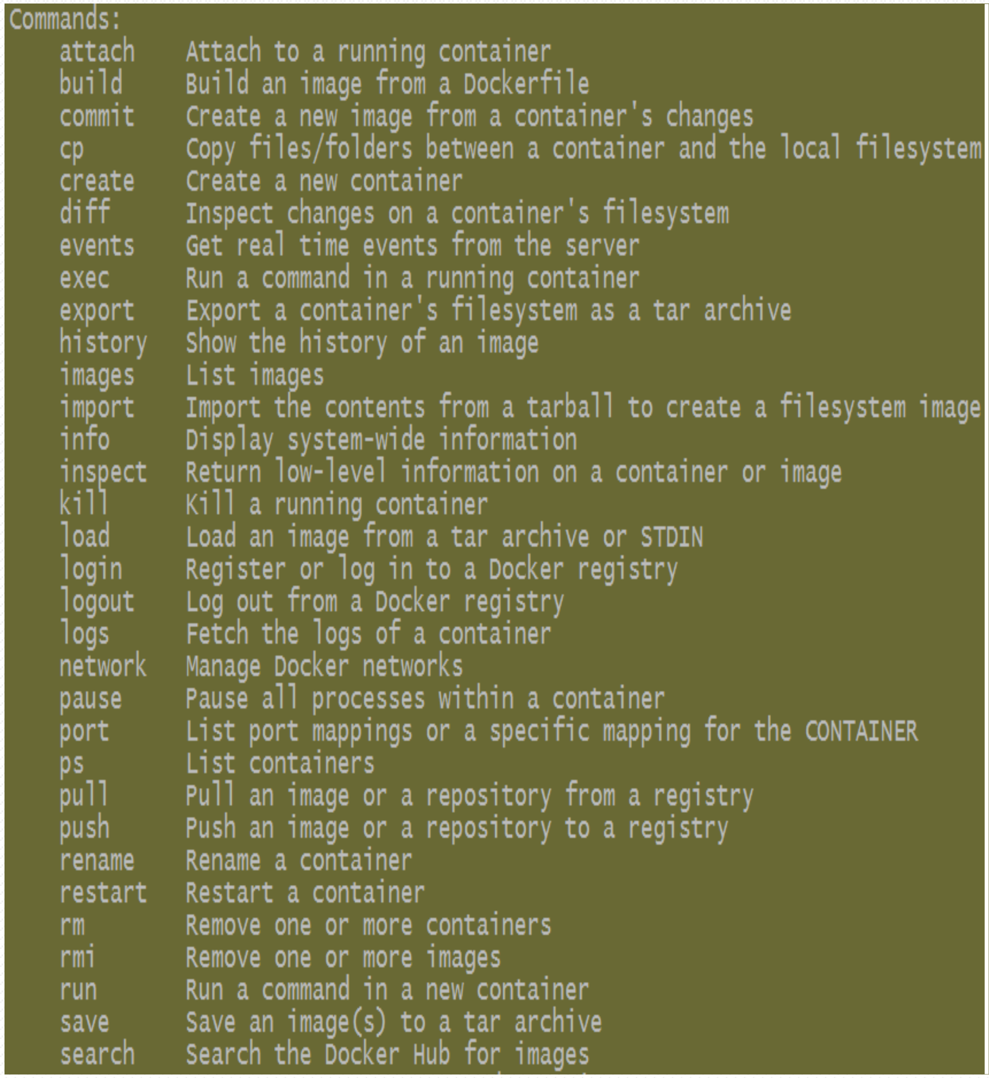
`$ docker attach <container code>` --- attach to running container

Docker HUB: Registries and Repository:

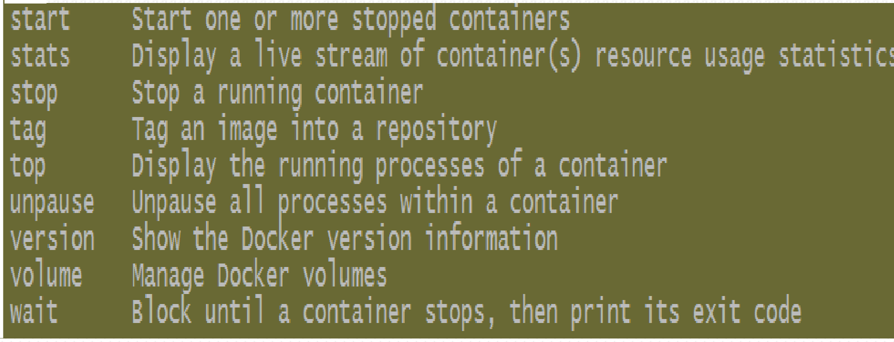


Docker Commands

Docker provides commands to manage, to create and share the images



```
Commands:
    attach    Attach to a running container
    build     Build an image from a Dockerfile
    commit    Create a new image from a container's changes
    cp        Copy files/folders between a container and the local filesystem
    create    Create a new container
    diff      Inspect changes on a container's filesystem
    events    Get real time events from the server
    exec      Run a command in a running container
    export    Export a container's filesystem as a tar archive
    history   Show the history of an image
    images    List images
    import    Import the contents from a tarball to create a filesystem image
    info      Display system-wide information
    inspect   Return low-level information on a container or image
    kill      Kill a running container
    load      Load an image from a tar archive or STDIN
    login     Register or log in to a Docker registry
    logout    Log out from a Docker registry
    logs      Fetch the logs of a container
    network   Manage Docker networks
    pause     Pause all processes within a container
    port      List port mappings or a specific mapping for the CONTAINER
    ps        List containers
    pull      Pull an image or a repository from a registry
    push      Push an image or a repository to a registry
    rename    Rename a container
    restart   Restart a container
    rm        Remove one or more containers
    rmi       Remove one or more images
    run       Run a command in a new container
    save      Save an image(s) to a tar archive
    search    Search the Docker Hub for images
```

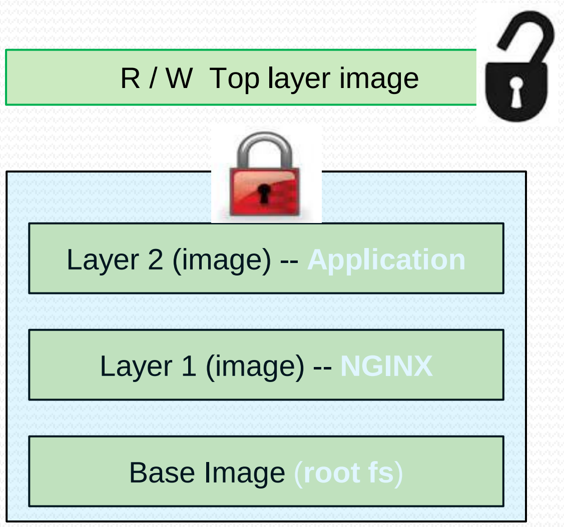


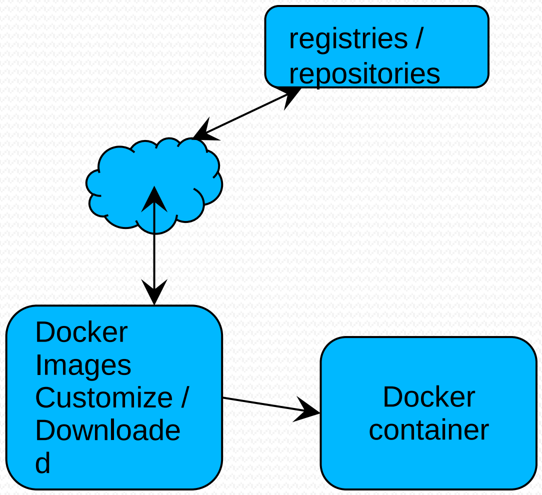
```
start     Start one or more stopped containers
stats     Display a live stream of container(s) resource usage statistics
stop      Stop a running container
tag       Tag an image into a repository
top       Display the running processes of a container
unpause   Unpause all processes within a container
version   Show the Docker version information
volume    Manage Docker volumes
wait      Block until a container stops, then print its exit code
```

Docker Images

Docker Images: Docker images can be **stacked** on one another to form a single image. This is achieved using the **Union mounts.**

This scenario is very much likely when we go on adding new application on the existing images. This provides a lot of flexibility of maintaining and updating, making it highly scalable and portable.

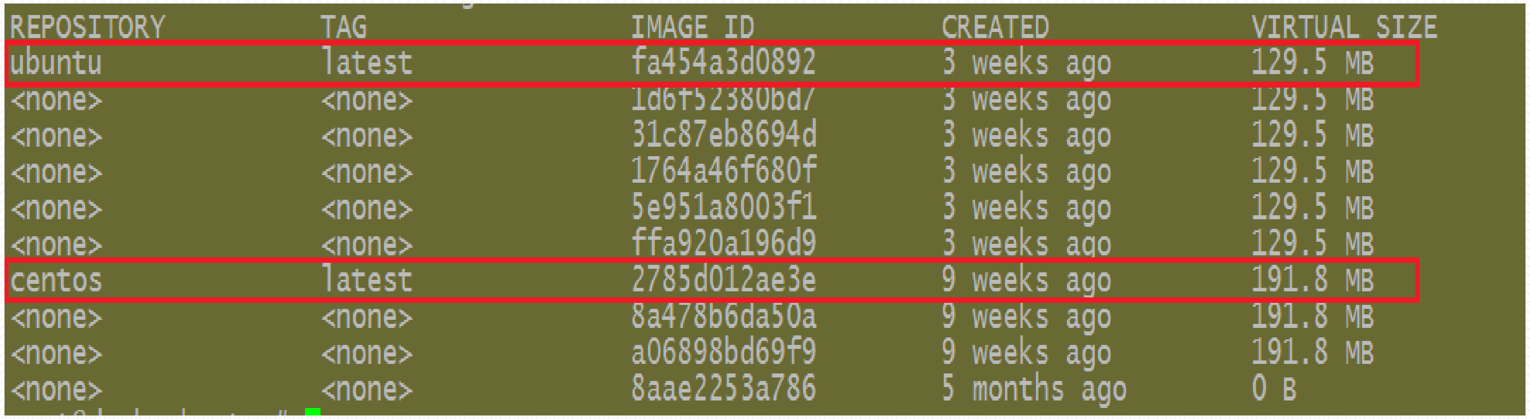



registries / repositories
Docker Images Customize / Downloaded
Docker container

`$ docker image history <image_id>` … provides history of the image… what changes have happened while creating the image.

Docker Images

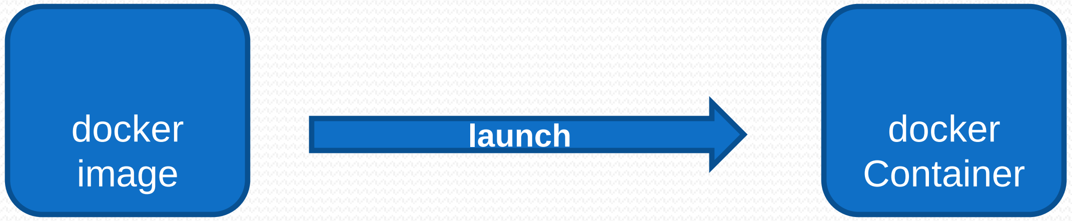
When we pull docker images from hub / repository, there are other images that are pulled along with it. Those are the **layers** of images that are part of the top level image.



```
REPOSITORY          TAG                 IMAGE ID            CREATED             VIRTUAL SIZE
ubuntu              latest              fa454a3d0892        3 weeks ago         129.5 MB
<none>              <none>              1d6f52380bd7        3 weeks ago         129.5 MB
<none>              <none>              31c87eb8694d        3 weeks ago         129.5 MB
<none>              <none>              1764a46f680f        3 weeks ago         129.5 MB
<none>              <none>              5e951a8003f1        3 weeks ago         129.5 MB
<none>              <none>              ffa920a196d9        3 weeks ago         129.5 MB
centos              latest              2785d012ae3e        9 weeks ago         191.8 MB
<none>              <none>              8a478b6da50a        9 weeks ago         191.8 MB
<none>              <none>              a06898bd69f9        9 weeks ago         191.8 MB
<none>              <none>              8aae2253a786        5 months ago        0 B
```

`$ docker load -i <location_of_image_file>` … load a image file downloaded

`$ docker pull -a <image_name>` … pulls all docker images for an image name e.g. docker pull –a ubuntu.. Will pull all available versions of Ubuntu images from docker hub.



Build-time construct

Run-time construct

Docker Containers

Every Docker container get a **writable top layer** where changes to the container are done, like installing a new app, file creation, IP address changes etc.,

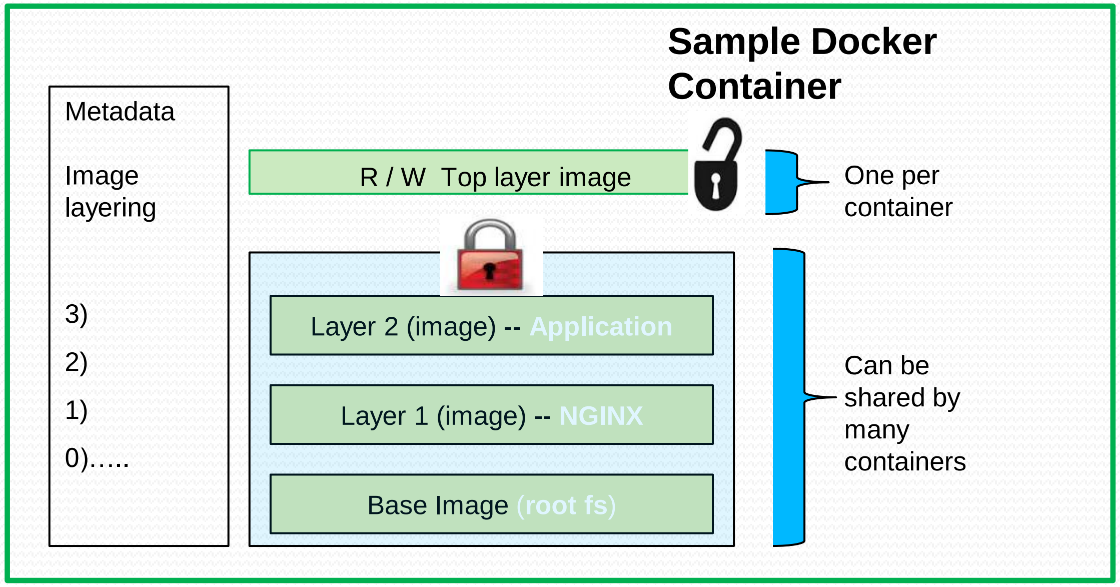
`$ docker run -it ubuntu /bin/bash` --- run a container using mentioned image, unpacks the image, layers and build a container with a terminal in interactive mode and assign a shell prompt.

`$ docker -p -q` .. Keep docker running in the background without exiting.

`$ docker run -it -d ubuntu` ----- run a docker container in detached mode. The container can be attached to with the command, `$ docker attach <container_ID>`

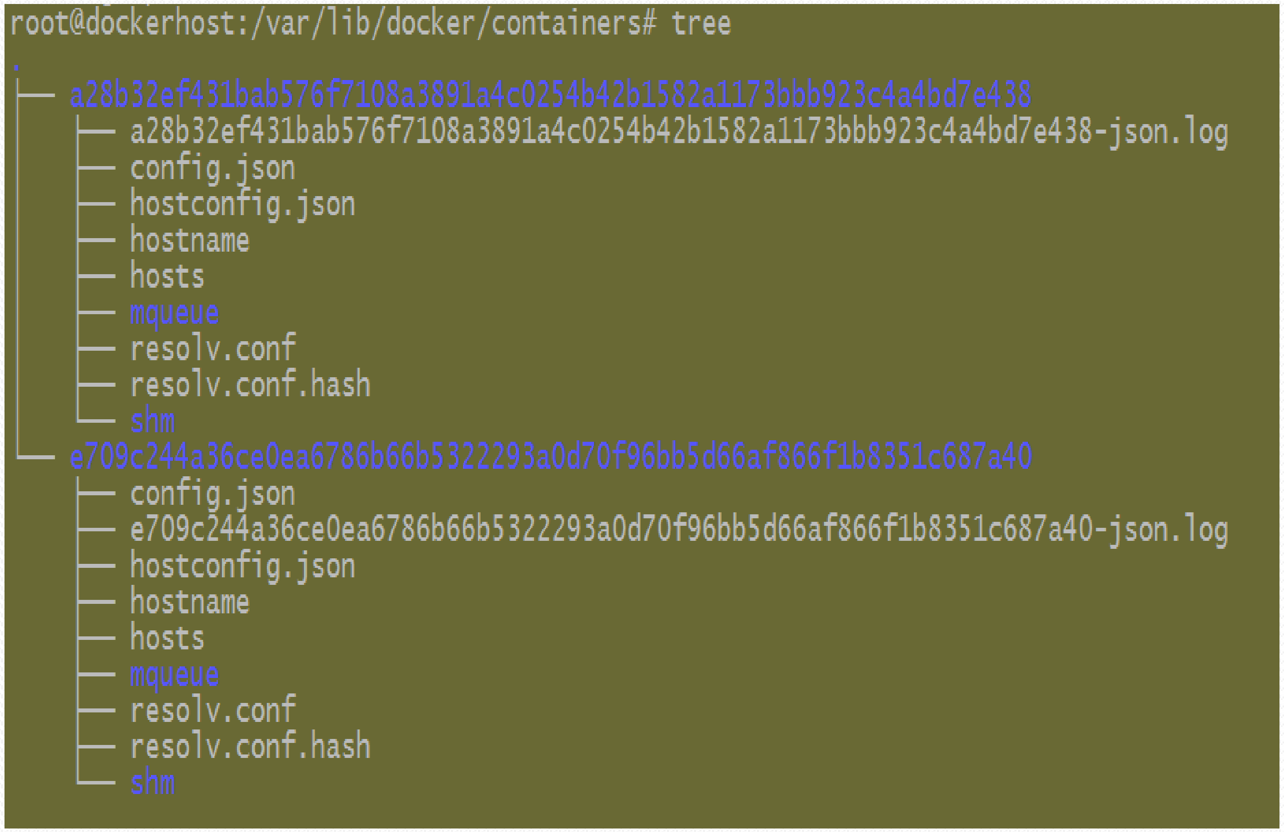
Also one get attached to docker running container using command,

```
$ docker exec -it <docker_container_id> bash
```



Docker Containers

Inside /var/lib/docker/container folder are the container structures.



```
root@dockerhost:/var/lib/docker/containers# tree
.
├── a28b32ef431bab576f7108a3891a4c0254b42b1582a1173bbb923c4a4bd7e438
│   ├── a28b32ef431bab576f7108a3891a4c0254b42b1582a1173bbb923c4a4bd7e438-json.log
│   ├── config.json
│   ├── hostconfig.json
│   ├── hostname
│   ├── hosts
│   ├── mqueue
│   ├── resolv.conf
│   ├── resolv.conf.hash
│   └── shm
└── e709c244a36ce0ea6786b66b5322293a0d70f96bb5d66af866f1b8351c687a40
    ├── config.json
    ├── e709c244a36ce0ea6786b66b5322293a0d70f96bb5d66af866f1b8351c687a40-json.log
    ├── hostconfig.json
    ├── hostname
    ├── hosts
    ├── mqueue
    ├── resolv.conf
    ├── resolv.conf.hash
    └── shm
```

We can create images that can be exported / shared with using the command,

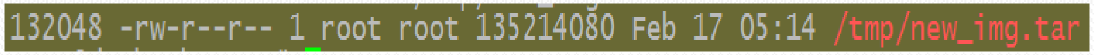
```
$ docker commit <container ID> <image_name>
```

e.g.

`$ docker ps -a` --- to list all docker container that's been running currently and in the past.

```
$ docker commit e709c244a36c new_img
```

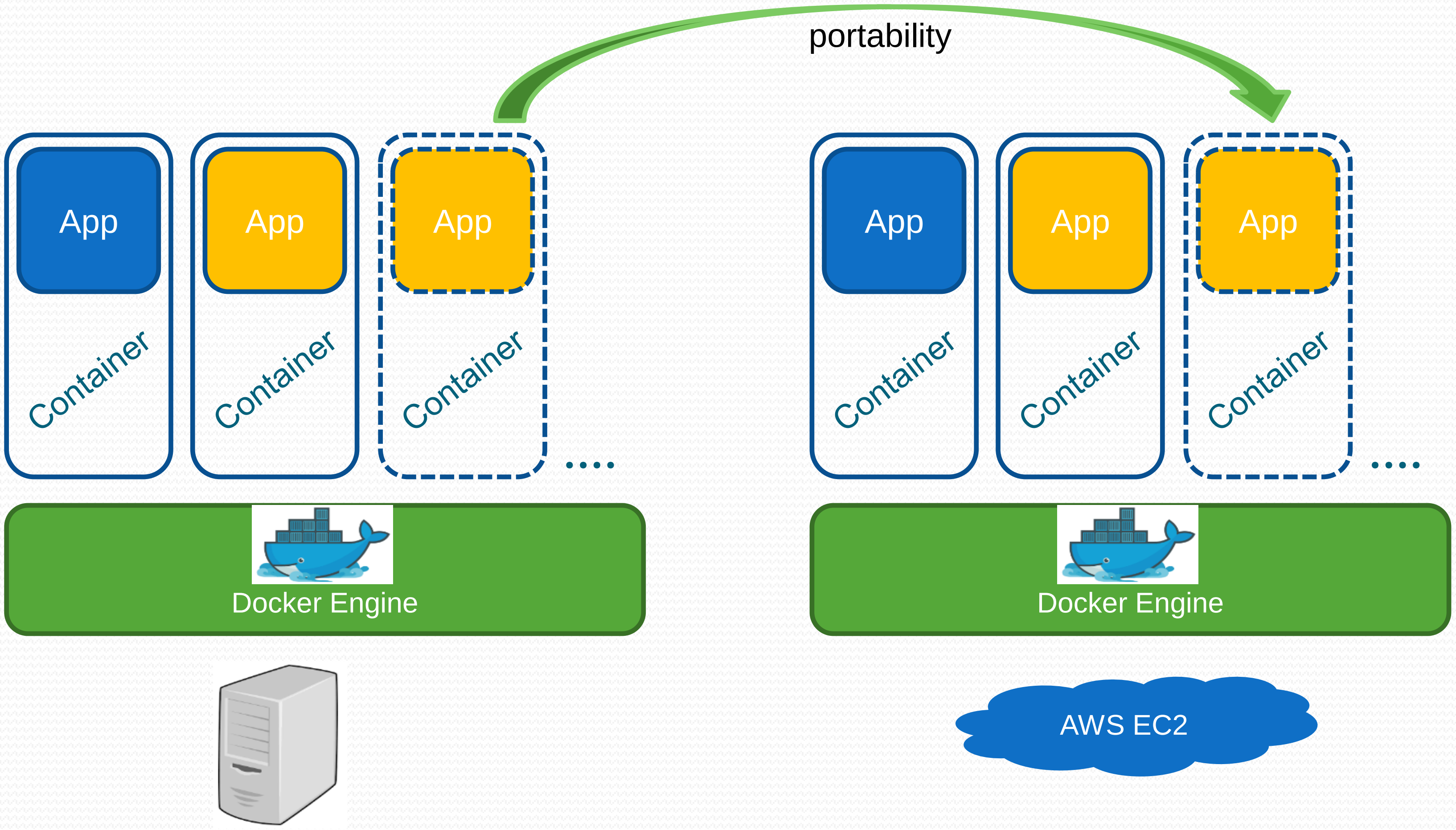
The image that gets created is only the top layer image with very small size that holds only the latest updates, etc.



```
132048 -rw-r--r-- 1 root root 135214080 Feb 17 05:14 /tmp/new_img.tar
```

Docker Image file created as tar ball. This can be shared exported to other machine as needed.

Dockerization



DockerFile

Dockerfile: Declarative way to construct an image.

- A **Dockerfile** is similar in concept to the recipes and manifests found in configuration management tools like Chef or Puppet.
- **Dockerfile** is much more stripped down than the IA tools, consisting of a single file with a DSL that has a handful of instructions.
- Below is sample **Dockerfile** that uses the ubuntu image, **RUN** command is used to run a certain command inside a container., **ADD** is used to add a file from host machine to the container and place it at desired location.

```
FROM ubuntu:latest

RUN apt-get update
RUN apt-get install -y python
python-pip wget
RUN pip install Flask

WORKDIR /home
```

Images are build time constructs and Container are run time constructs.

```
$ docker build –t <dockerimage_name> <dockerfile_path>
```

Docker registries and repos

https://registry.hub.docker.com --- docker official registry

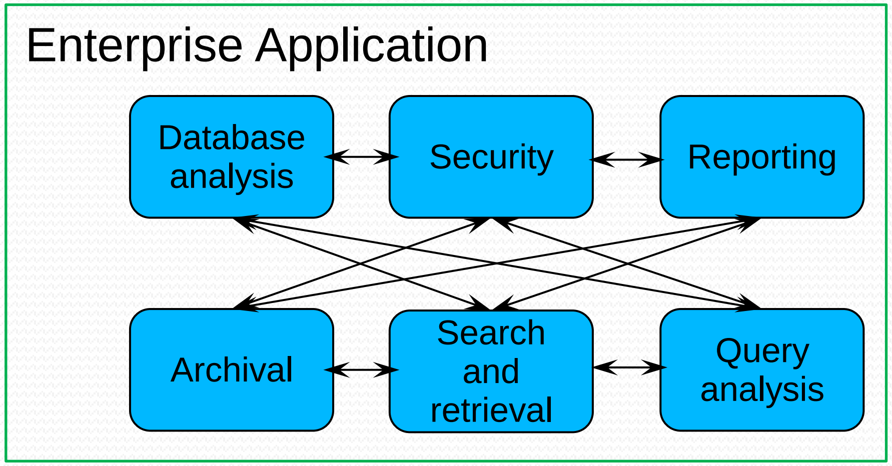
https://hub.docker.com/ --- create account on docker hub to upload / contribute to the docker public repository.

https://store.docker.com --- image repositories.

Use `$ docker pull <image_name>`… command to pull images from registries.

Docker and Microservices

In an application there are multiple services and such services can be offered in an Docker container as a service and such container working together forms the entire application. This way, the individual service can be updated maintained without imposing any downtime on the application.



Thanks You