



What is



**DOCKER ?**

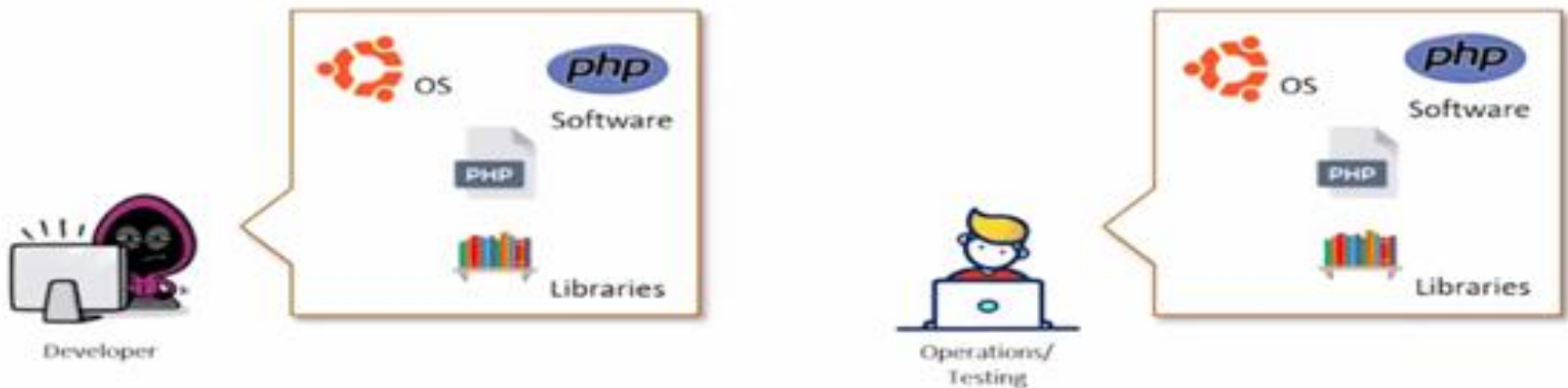
# Problem before Docker



Developers used to run the code on their system, it would run perfectly. But the same code did not run on the operations team system.

# Reason behind that problem

- 1) Different OS platform
- 2) Same OS platform but different OS version
- 3) Different Application Version
- 4) Different hardware configuration



# What is the solution



We needed an entity which can **"contain"** all the software dependencies, and can be ported on to other computers as plug & play package

# What is Docker

- ❑ Docker is a container management service.
- ❑ The keywords of Docker are **develop**, **ship** and **run** anywhere.
- ❑ The whole idea of Docker is for developers to easily develop applications, ship them into containers which can then be deployed anywhere.



Build

Develop an app using Docker containers with any language and any toolchain.



Ship

Ship the "Dockerized" app and dependencies anywhere - to QA, teammates, or the cloud - without breaking anything.



Run

Scale to 1000s of nodes, move between data centers and clouds, update with zero downtime and more.

# Build

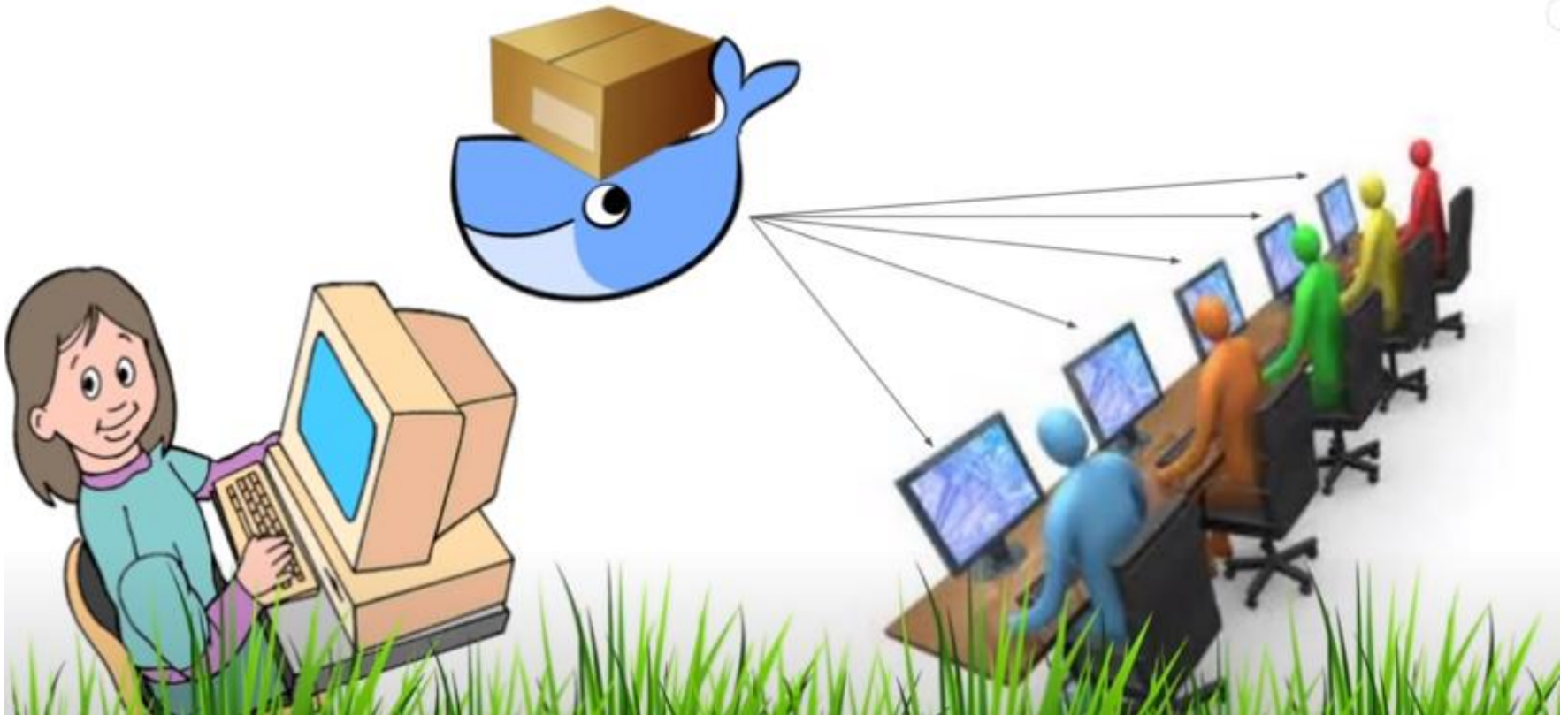
Containers allow a developer to package up an application with all of the parts it needs such as libraries and other dependencies, and ship it all out as one package





# Ship and Run

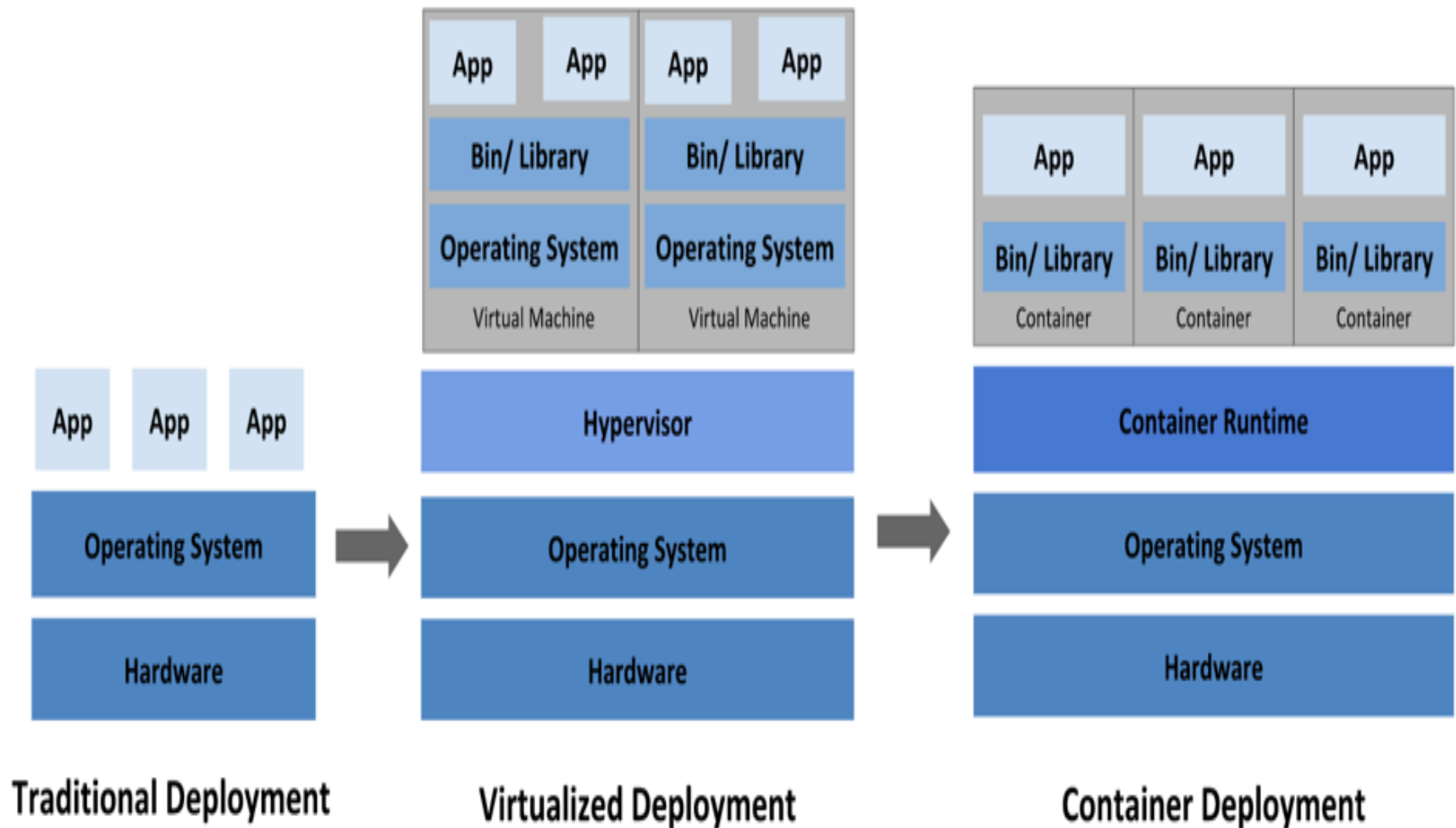
- The docker container is shipped into dockerhub and can be easily pulled to the operation system to run it.







# Virtualization vs containerization



# Advantages over Virtualization

1) Less development overhead  
3) Smaller size    4) Faster startup

2) Less resource consumed  
5) Reduced and simplified updates



## Virtual Machines

Each virtualized application includes not only the application - which may be only 10s of MB - and the necessary binaries and libraries, but also an entire guest operating system - which may weigh 10s of GB.



## Docker

The Docker Engine container comprises just the application and its dependencies. It runs as an isolated process in userspace on the host operating system, sharing the kernel with other containers. Thus, it enjoys the resource isolation and allocation benefits of VMs but is much more portable and efficient.

# Docker Advantages



## Build app only once

An application inside a container can run on any system that has Docker installed. So there is no need to build and configure app multiple times on different platforms



# Docker Advantages



## Portability

Docker containers can run on any platform.

It can run on your local system, Amazon ec2, Google Cloud platform, Rackspace server, VirtualBox..etc.

# Docker Advantages



## Isolation

With Docker every application works in isolation in its own container and does not interfere with other applications running on the same system.

So multiple containers can run on same system without interference.

For removal also you can simply delete the container and it will not leave behind any files or traces on the system.





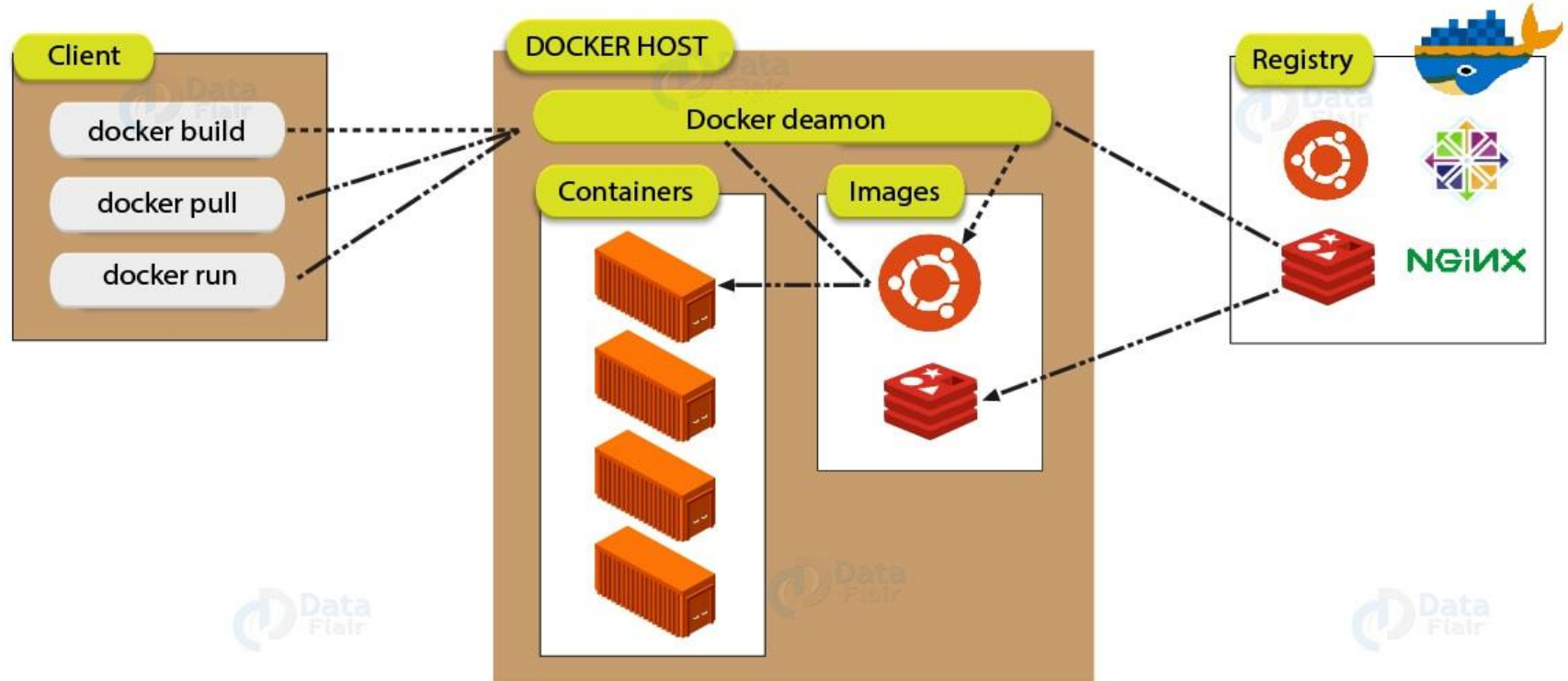
# Docker Architecture

Docker Engine  
Registries

Docker Images  
Docker Container



## Docker Architecture



# Docker Hub

Docker Hub is the world's largest repository of container images with an array of content sources including container community developers, open source projects and independent software vendors (ISV) building and distributing their code in containers.

Users get access to free public repositories for storing and sharing images or can choose subscription plan for private repos.

## Integration Tests



## Deployment Platforms



Google Compute Engine

## Docker Hub



## Revision Control




boot2docker




Docker Hub

https://hub.docker.com

 Dashboard Explore Organizations

Search

Create  shakinstev


shakinstev


Repositories Stars Contributed


Private Repositories: Using 0 of 1 [Get more](#)

# Welcome to Docker Hub

Here are a few things to get you started.

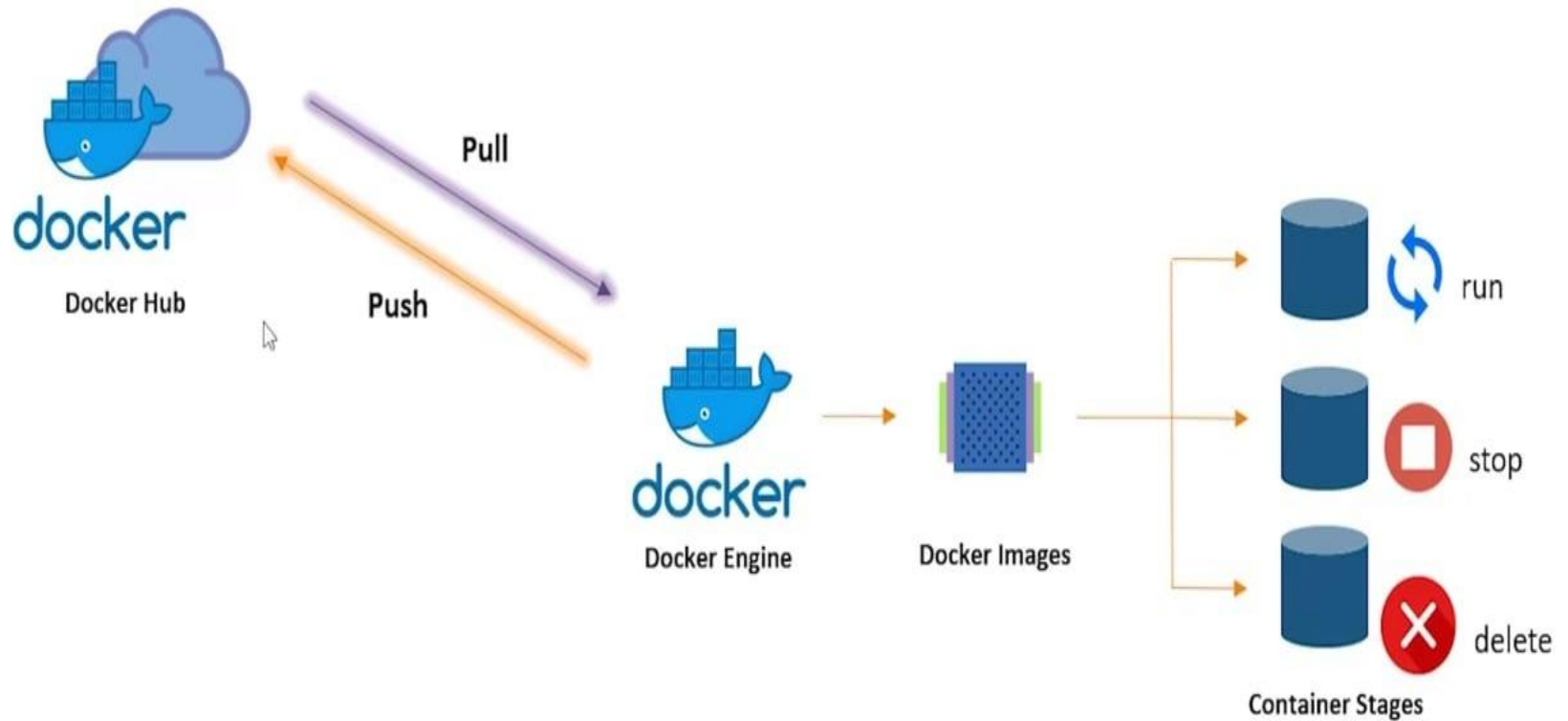
  
Create Repository

  
Create Organization

  
Explore Repositories

# Docker Container Lifecycle

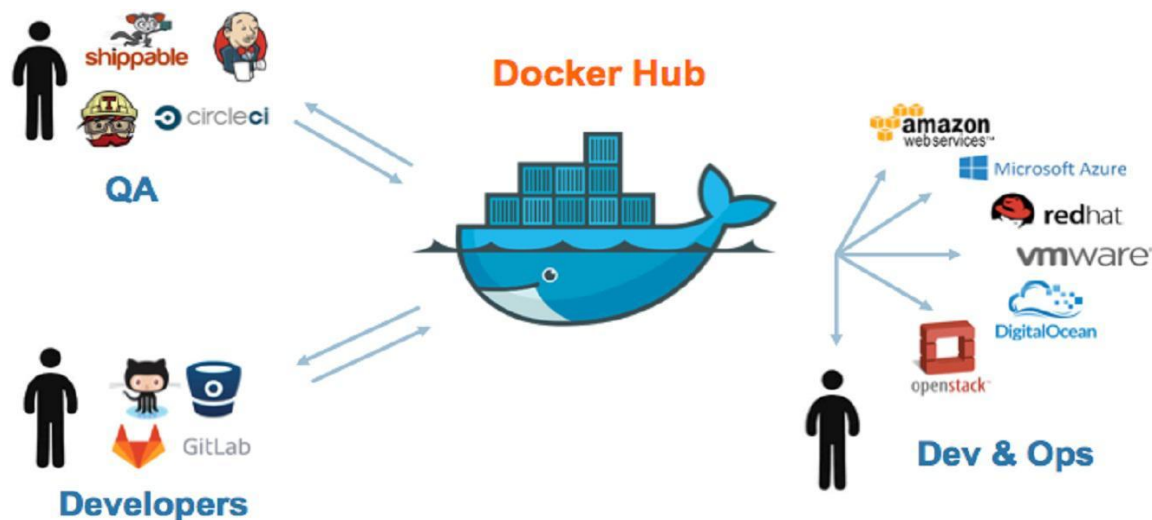
---





# Docker Hub

- **Now create account in dockerhub.com**
- Open dockerhub.com
- Signup -- put unique name -- email id
- open email id and verify the link
- login to docker hub



# Docker Images

- **Base images** are images that have no parent image, usually images with an OS like ubuntu, busybox or debian.
- **Child images** are images that build on base images and add additional functionality.

Then there are official and user images, which can be both base and child images.

- **Official images** are images that are officially maintained and supported by the folks at Docker. These are typically one word long. In the list of images above, the python, ubuntu, busybox and hello-world images are official images.
- **User images** are images created and shared by users like you and me. They build on base images and add additional functionality. Typically, these are formatted as user/image-name.