Docker is world’s leading software container platform.

Docker is a tool designed to make it easier to deploy and run applications using containers.

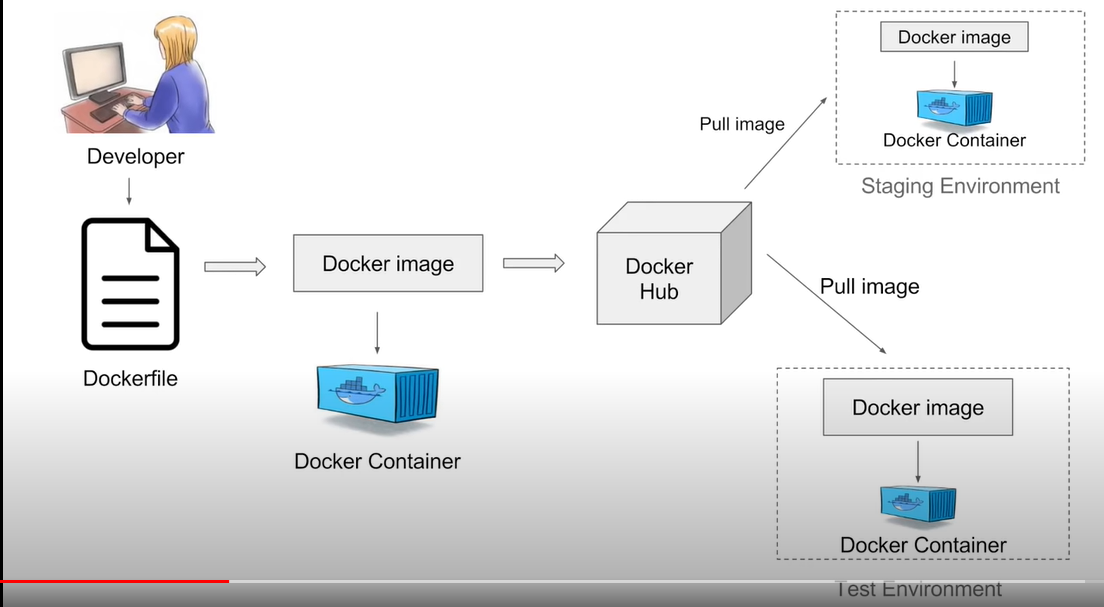
Container enables all developer to package up an application with all parts it need such as libraries, other dependencies and ship it all as one package.

Docker resolves the basic problem that application works on one system and not working on other system.

Docker comes into picture at deployment stage.

Docker Workflow 🡪

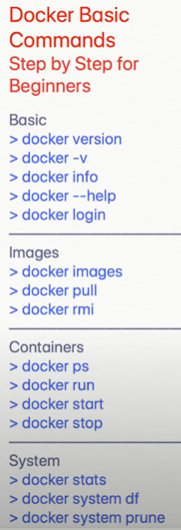
* Developer will define its application, dependencies, requirements in a file called ‘dockerfile’.
* This dockerfile is used to generate ‘Docker image’.
* When docker image is run, we get docker container. So docker containers are runtime instances of docker image.
* Docker images can also be stored in online cloud repository known as ‘Docker Hub’.
* These images can be pulled to create containers in any environment.



Advantages 🡪

* No need to build and configure app multiple times on different platforms.
* No discrimination between test and production environment
* Portability means if you create docker using Amazon EC2 and you can easily port this image to VirtualBox and it will run as it is.
* Version control like Git like making changes in image and committing them.
* Isolation, every application works inside its own container and do not interfere with other running applications.
* Faster deployment, a developer can package all its softwares and dependencies in a container and docker will make sure that all this is deployed on every possible platform and everything works fine.

**Docker Basic Commands 🡪**



docker login 🡪 to login to <http://hub.docker.com> & provide username and password

docker images -a 🡪 returns the list (names) of all the images

docker images -q 🡪 returns the ID of all images

docker pull ‘image\_name’ 🡪 pulls the image on your system

docker rmi ‘image\_ID’ 🡪 deletes the specified image

Container commands

docker run -it ‘image\_ID’ 🡪 for creating a container by running the image (here ‘-it’ means interactive)

docker ps 🡪 returns the running container info

docker start ‘container\_ID’ 🡪 starts the docker

docker stop ‘container\_ID’ 🡪 stops the docker

docker stats 🡪 displays the stat of running docker like memory usuage

docker system prune 🡪 removes all the images from the system