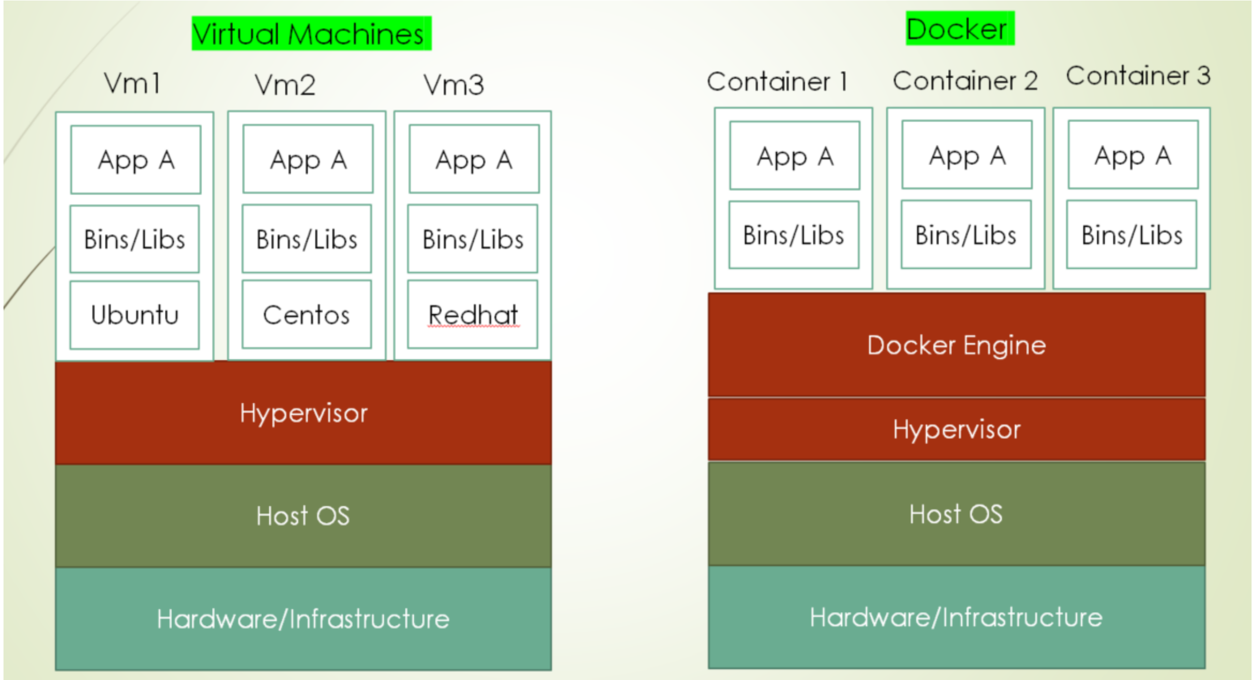
http://youtube.com/c/javaexpress
+91 8555929285
Telegram: @javaexpressgroup

Docker Material

Hypervisor:

- **A hypervisor allows one host computer to support multiple guest VMs by virtually sharing its resources, such as memory and processing**



What is a Docker?

- Docker is a tool which is used to automate the deployment of applications in lightweight containers so that applications can work efficiently in different environments.
- Docker will use host operating system and then your container will run
- Docker containers are very light weight

Docker Engine:

- **Docker Engine** will run the docker container
- We need to install docker engine in our machine
- Docker engine can understand the docker container

Docker Image:

- Docker image is a template with instructions which is used for creating containers.
- A Docker image is built using a file called Docker File
- Docker image is ready made available in Docker Registry
- Docker image is stored in a Docker Hub or Docker registry
  - Public Repository
  - Private Repository.

Docker Container

- Docker Container is a standalone executable software package which includes applications and their dependencies to run the application
- Docker containers run on the same infrastructure and share operating system with its other containers.
- Each application runs in isolation
- Booting of containers are very fast
- Containers start, stop and kill, remove containers easily and quickly
- Containers run on the same machine sharing the same Operating system Kernel.

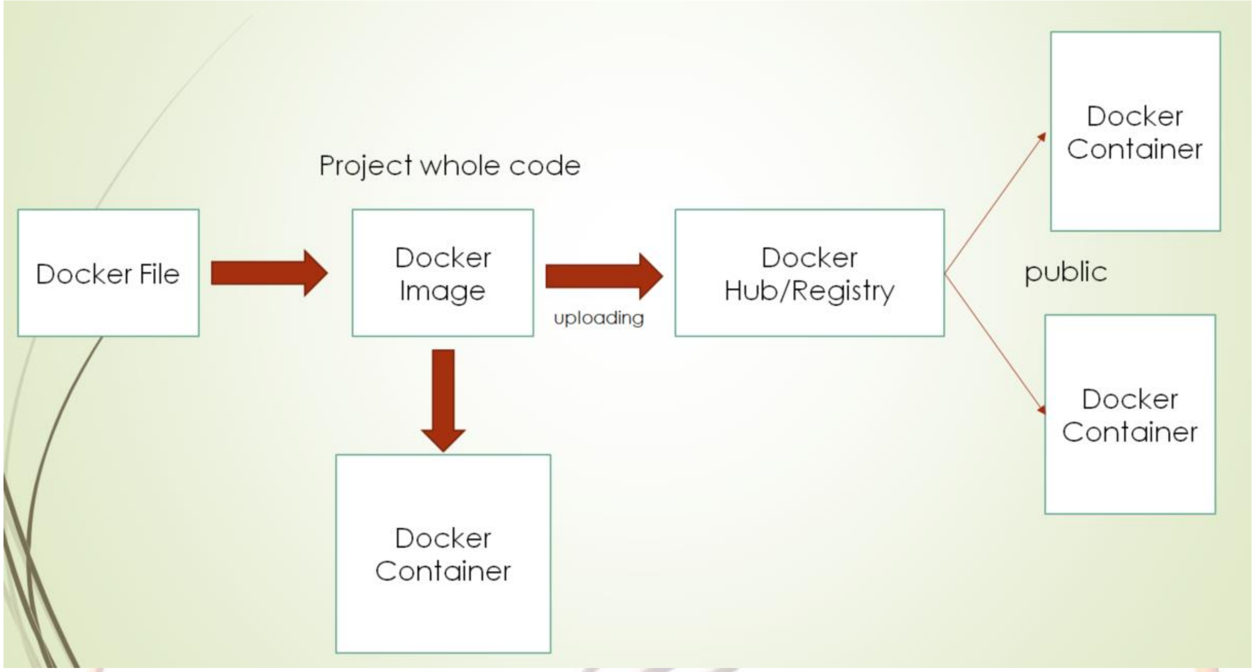
Docker Registry:

- It is open source server side service used for hosting and distributed images
- Docker also has its own default registry called Docker Hub.
- Pull and push commands used by user

Difference between Docker Image vs Container

Docker Image	Docker Container
Image is a blueprint	Using Blueprint will run containers
Image is read only	Container is COW (Copy on Write)
We can't modify docker image	We can create own docker image
We can create docker image by using docker File	Container is nothing but Virtual machine
Docker images we can push to docker public repository	Docker containers are completely volatile
Docker image – pull and push	Containers has edit access

Components of Docker



What is a Docker File?

- Dockerfile is a normal text file used to build our own docker images by giving specific instructions, so that we can create our own customized docker images in an automated way without running a docker container.
- A text file with instructions to build image.

Steps to Create Docker Image

1. Create Dockerfile
2. Add instructions to Dockerfile
3. Build docker image using Dockerfile
4. Run the container

Docker Commands

Docker Command	Description
"docker --version"	**To check docker version**
"docker info"	**displays system wide information regarding the docker installation**
"docker pull hello-world"	**Pull docker image from docker hub**
"docker pull ubuntu"	It pull ubuntu machine from docker hub in local
"docker pull ubuntu:14.04"	To pull Ubuntu with version number
"docker images"	**It will docker images in local**
"docker run ubuntu"	**It will start and stop automatically in your container**
"docker ps"	**It will display running containers**
"docker ps -a"	It will display running and stopped containers
"docker run –it ubuntu bash"	It will run container and it will move to inside container

Inside Container Commands

whoami

touch abc.txt

echo $date

exit

Shortcut : ctrl +pq : (it will not stop ubuntu container)

Docker Command	Description
"docker run -- name bhaskar –it ubuntu bash"	we can provide name for the container
"docker run –it –d ubuntu bash "	It will run in background mode
"docker exec –it **containerId** bash"	It will login into existing virtual machine
" docker run -- name bhaskar –it –d ubuntu bash "	We can provide name & running in background mode
Install Software's in Ubuntu touch abx.txt apt-get update apt-get install git	
"docker commit containerId"	It will create new image with existing modifications
"docker image tag imageId ubuntu_git:version"	We can tag existing imageId with new name and version
" docker image tag imageId anvbhaskar/ubuntu_git:version "	To tag with username
"docker rmi imageName:version"	To remove docker image
"docker rmi –f ubuntu:latest"	It will remove forcefully docker image

Docker Command	Description
"docker start container-id"	To start one or more stopped containers
"docker pause container-id"	To pause all processes within one or more containers
"docker unpause container-id"	To unpause all processes within one or more containers
"docker stop container-id"	To stop one or more running containers
"docker kill container-id"	To kill one or more running containers instantly
"docker restart container-id"	To restart one or more containers
"docker inspect container-id"	To inspect all the details for a given container id
"docker logs container-id"	To fetch the logs of a given container id
"docker logs -f container-id"	To follow log output of a given container id
"docker rm container-id"	To remove one or more containers based on container ids
"docker container prune"	To remove all stopped containers
"docker compose up"	To create and start containers based on given docker compose file
"docker compose stop"	To stop services

http://youtube.com/c/javaexpress
+91 8555929285
Telegram: @javaexpressgroup

Docker Command	Description
"docker rmi -f $(docker images -a -q)"	To remove docker images
" docker rm -vf $(docker images -a -q)""	To delete all containers including its volume
"docker system prune -a "	To delete everything
"docker system prune -- all "	To delete everything
"docker system prune –a -- volumes "	To delete everything include volumes

Steps to Create Docker Image for Spring Boot Application

1. Create Docker file in root folder in Spring Boot Application
2. Add Instructions
3. Build Docker Image
4. Run
5. Push to Docker Hub

```
docker build . -t anvbhaskar/product:0.0.1
docker run –p 8080:8080 anvbhaskar/product
docker logs –f containerId
```