Docker Management

What is Docker

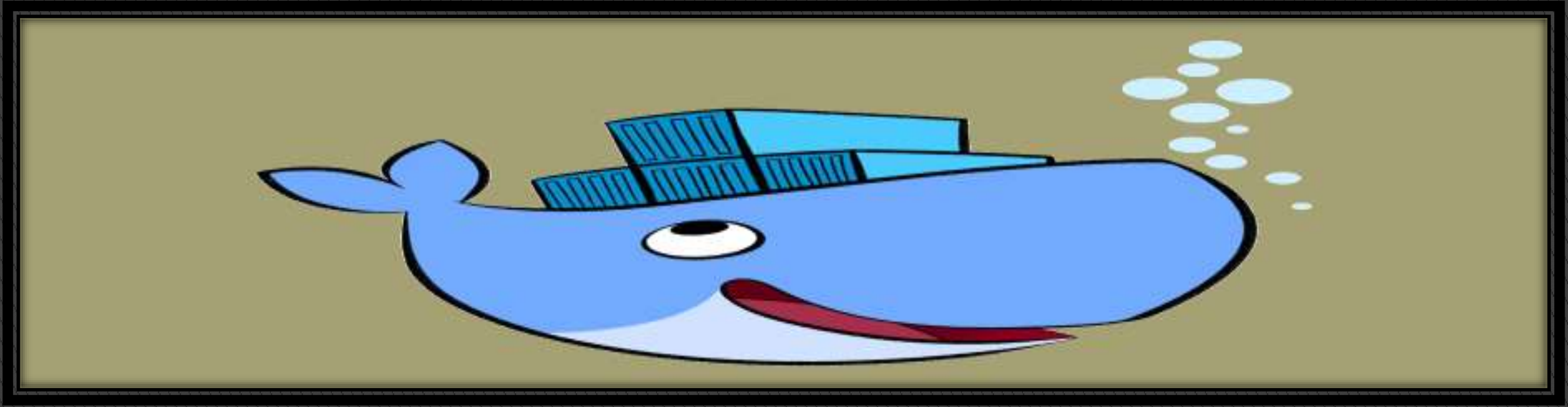
- Docker is an open-source Centralized Platform designed to create, deploy and run applications.
- Docker uses container on the host OS to run application. It allows applications to use the same linux kernel OS a system on the host computer, rather than creating a whole virtual OS.
- We can install docker on any OS but Docker engine runs natively on linux distribution.
- Docker written in ‘go’ language.
- Docker is a tool that performs OS level virtualization, also known OS Containerization.
- Before Docker, many users faces the problem that a particular code is running in the developer’s system but not in the users system.
- Docker was first Release in March 2013. It is developed by Solomon Hykes and Sebastien Pahl
- Docker is a set of Platform as a Service that uses OS level virtualization whereas VMware uses Hardware level Virtualisation.

Docker Container

- Container hold the entire package that is needed to run the application.
- Or
- In other words, we can say that, the image is a template and the container is a copy of that templates.
- Container is like a Virtual Machine
- Image becomes container when they run on docker engine.

Why Container

- One of the challenges developers faced in the past is getting applications to run reliably across multiple computing environments. Oftentimes, applications didn't run as expected or encountered errors and failed altogether. And that's where the concept of containers was born



Introduction Of Container

- Linux containers can be thought of as a lightweight alternative to virtualization. In virtual machine configuration operating system create all OS setup, store all files, and run entire OS environment on guest operating system, it use all required system resources form the physical hardware, hypervisor manage all access of physical resources for the virtual system from physical hardware of host system.

- Whereas when Containers work by using a concept referred to as kernel sharing, then kernel share all resources with container on the basis of Operating system which takes advantage of the architectural design of Linux and UNIX-based OS.

- It creates Operating system level virtualization.

Introduction Of Container

- The main advantage of containers is that they require considerably less resource overhead than virtualization allowing many container instances to be run simultaneously on a single server, and can be started and stopped rapidly and efficiently in response as per demand. Containers run natively on the host system it use OS level virtualization that’s reason it providing a high performance that cannot be matched by a virtual machine.

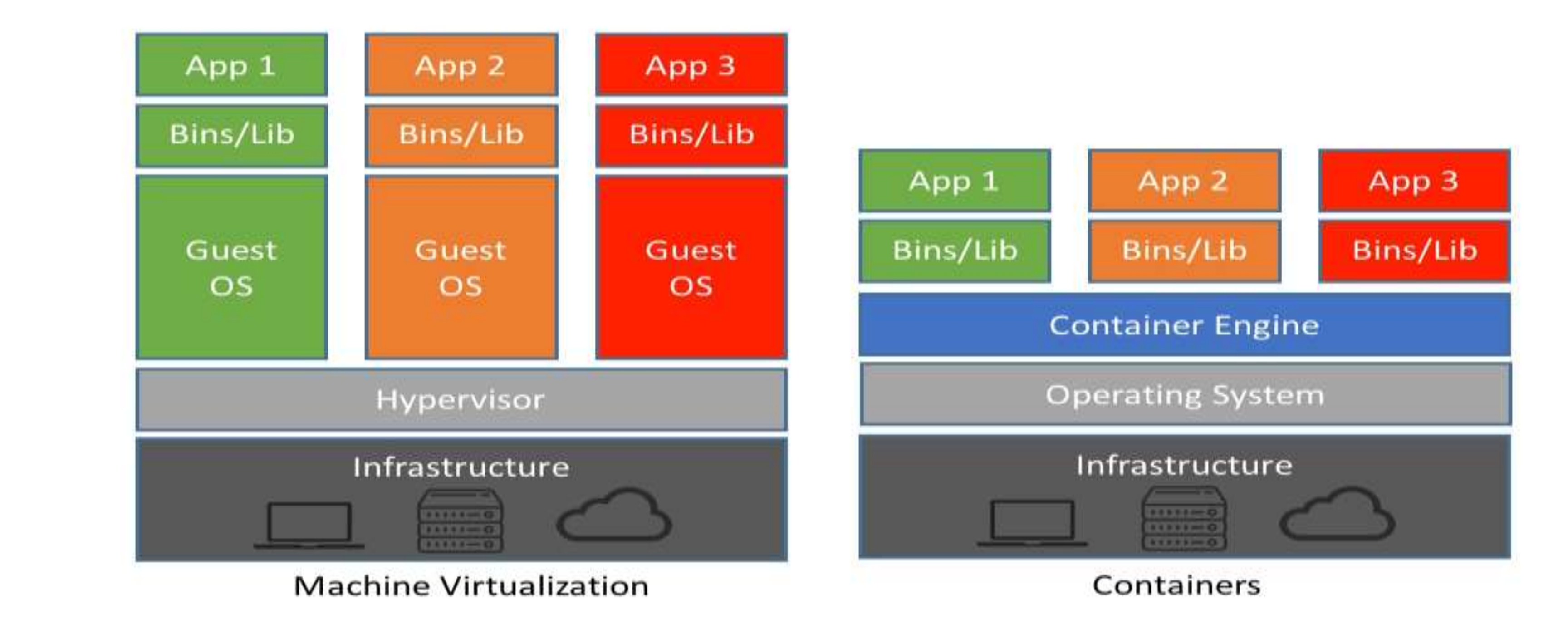
- Containers are work extremely portable and very easy to migrated one system to another system. When combined with a container management system such as Docker, OpenShift and Kubernetes, it is possible to deploy and manage containers on a vast scale spanning multiple servers and cloud platforms, potentially running thousands of container.

What are Container Images?

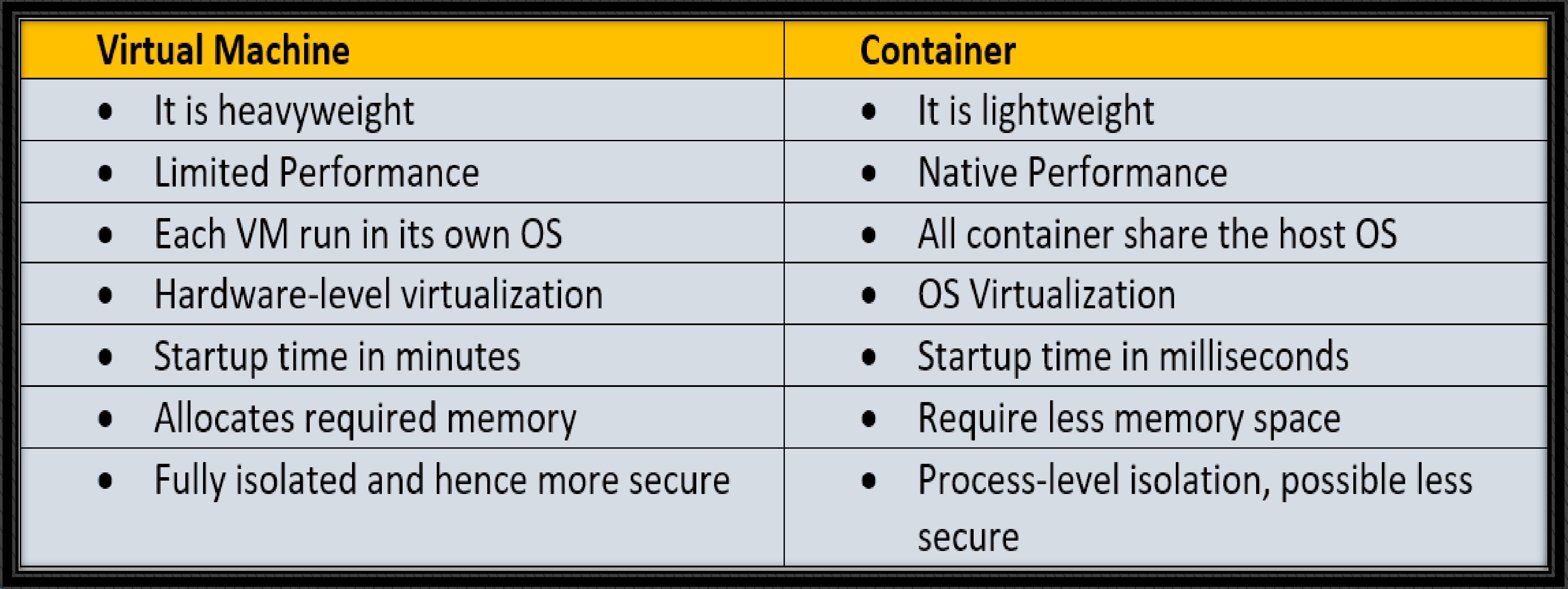
- Container images are static files images that ship with executable code that runs in an isolated environment. A container image are encapsulated all required environment such as system libraries, dependencies & other platform settings needed by the application to run in diverse environment on various host Operating Sys .

- Red Hat Linux provides a set of useful container tools that can manage container flexibly or you can leverage to work directly with Linux containers using requiring docker commands.

Virtualization Vs Container



Virtualization Vs Container difference



Virtual Machine	Container
• It is heavyweight	• It is lightweight
• Limited Performance	• Native Performance
• Each VM run in its own OS	• All container share the host OS
• Hardware-level virtualization	• OS Virtualization
• Startup time in minutes	• Startup time in milliseconds
• Allocates required memory	• Require less memory space
• Fully isolated and hence more secure	• Process-level isolation, possible less secure

Advantages of Docker

- No pre-allocation of RAM
- Docker enables you to build a container image and use that same image across every step of the deployment process.
- Less Cost
- It is light in weight
- It can run on physical hardware / virtual h/w or on cloud
- You can re-use the image
- It took very less time to create container

Disadvantages of Docker

- Docker is not a good solution for application that requires rich GUI.
- Difficult to manage large amount of containers
- Docker does not provide Cross-platform compatibility means if an application is designed to run in a docker
- Container on Windows, then it can’t run on linux or vice-versa
- Docker is suitable when the development OS and testing OS are same if the OS is different, we should use VM
- No solution for Data Recovery and Backup.

Install docker in AMI Linux2

For update

```
#yum  update    -y
```

For install docker

```
#yum  install   docker   -y
```

For check version on docker

```
#docker   -v  or    --version
#rpm    -q    docker
#which    docker
```

Start and enable docker service

```
#systemctl        start      docker
#systemctl        enable     docker
#systemctl        status     docker

#docker   info
```

Check Docker image & Process

for show available image

```
#dockerimages
```

For show running process of docker container

```
#docker   ps
```

For show stop and running docker container

```
#docker   ps   -a
```

Introduction of docker hub

https://hub.docker.com

For run ubuntu

```
#docker    run    -it   Ubuntu    /bin/bash
root@cid# ls
root@cid# cat    /etc/os-release
root@cid# uname   -r
root@cid# free    -m
root@cid# exit

#docker    images

#docker    run    -it   Ubuntu    /bin/bash
```

Check container id, by default it create new container whenever we use docker run command

```
root@cid#exit
#docker  ps
#docker  ps  -a
```

Docker search centos

```
#docker    search    centos

#podman    pull   docker.io/library/httpd

#docker   run   -it     centos     /bin/bash

root@cid# cat   /etc/os-release
root@cid# exit
```

Run container customizable name

```
#docker   run  -it    -name    sevenmentor   Ubuntu   /bin/bash
root@cid#exit

#docker   ps   -a

#docker   start   sevenmento
#docker   ps
```