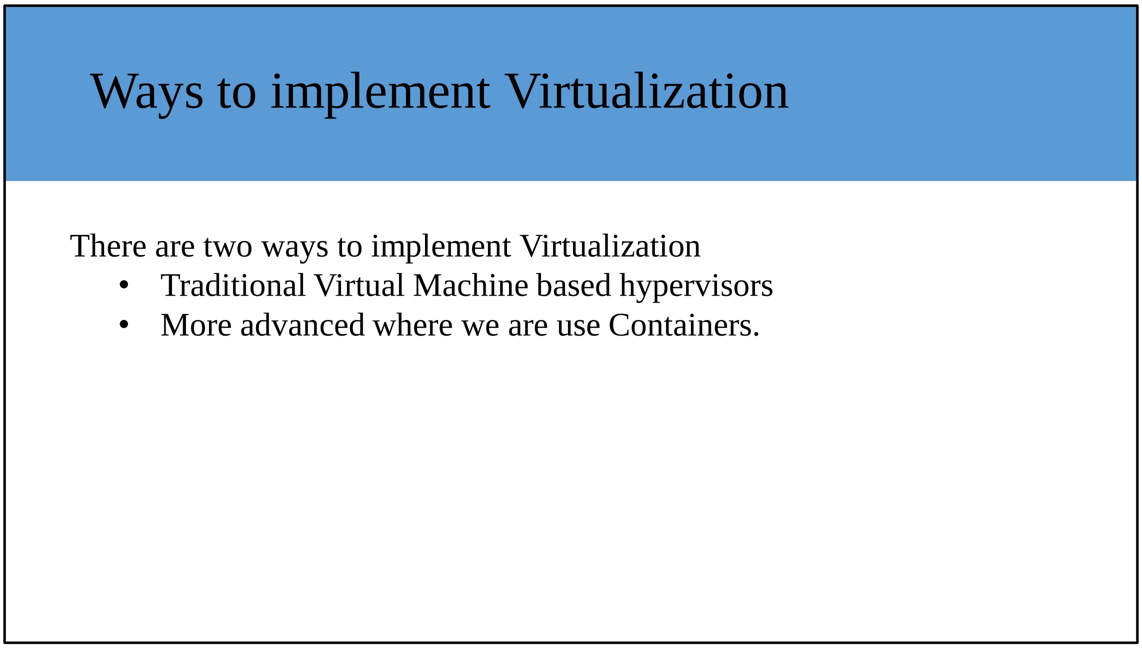
Cloud Computing

Docker and Container

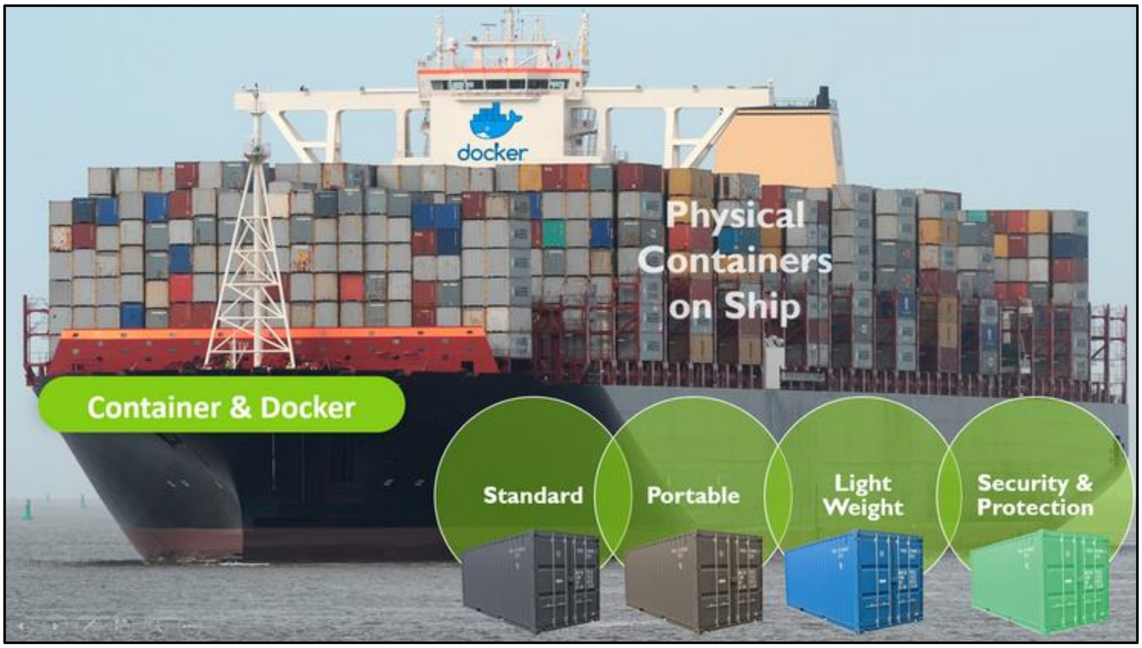


Ways to implement Virtualization

There are two ways to implement Virtualization

- Traditional Virtual Machine based hypervisors
- More advanced where we are use Containers.

VMs or Virtual Machines and Containers are going to do same thing .. i.e. Virtualization , Their Goals are same . They are going to enable Applications to use virtual resources such as Virtual Hardware , Virtual Compute , Virtual Network in much more scaleable and flexible way . With Virtualization , There is no need for physical hardware allowing for more efficient use of computing resources, both in terms of power , space , Infra and cost effectiveness . The main difference between containers and VMs is in their architectural approach.



Physical Containers on Ship

Container & Docker

Standard

Portable

Light Weight

Security & Protection

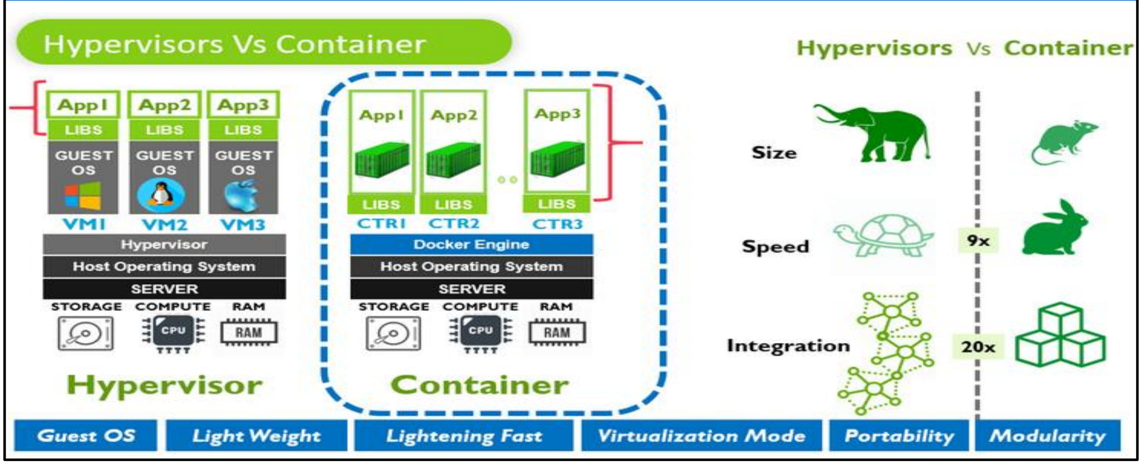
Before we get into deep understanding of Containers , Let’s understand , Why we call container as Container .You can see physical containers on Screen .There are few great Qualities of these physical Containers which we are going to use in Virtualization world .. Let’s discuss similarities between Physical container and Virtual container

Containers are complete Package .. They contains both application and respective libraries and any other dependencies . These containers are then shipped as one package which ensures it will run without any fault on every other cloud and any other machine.

Similarities on Physical container on Ship and Virtual containers

- Standard
- Portable
- Light weight
- Security and protection

Hypervisor v/s Container

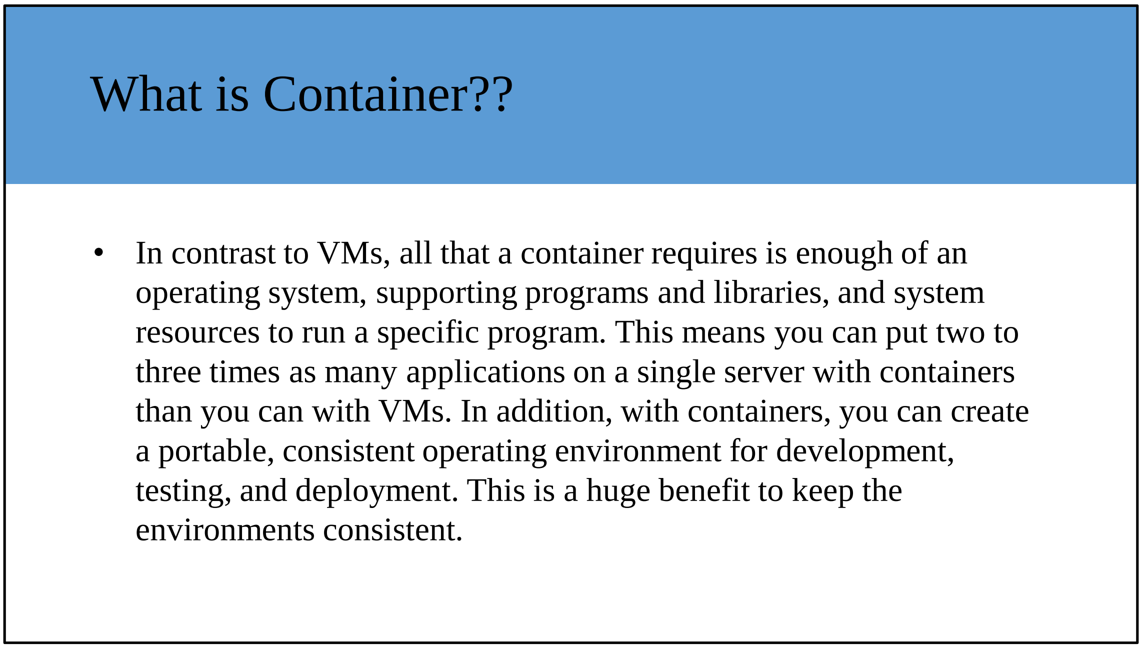


What is Container??

- Containers are a way to wrap up an application into its own isolated ”box”. For the application in its container, it has no knowledge of any other applications or processes that exist outside of its box. Everything the application depends on to run successfully also lives inside this container. Wherever the box may move, the application will always be satisfied because it is bundled up with everything it needs to run.

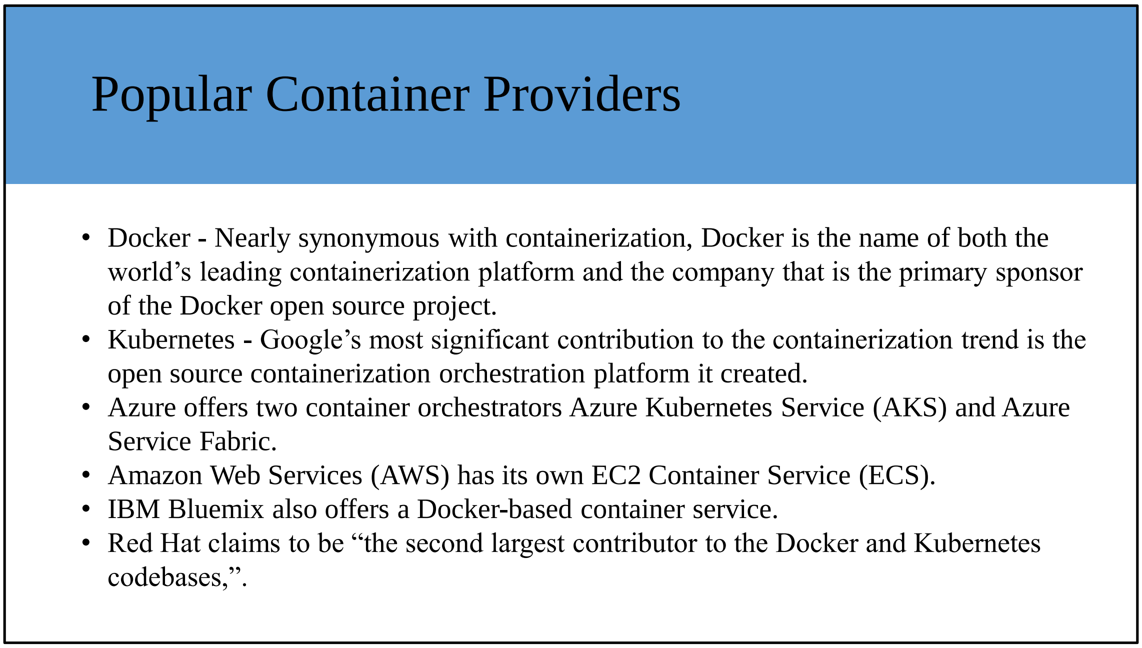
What is Container??

- Containers virtualize the OS instead of virtualizing the underlying computer like a VM. They sit on top of a physical server and its host OS — typically Linux or Windows. Each container shares the host OS kernel and, usually, the binaries and libraries, too. Shared components are read-only. Sharing OS resources such as libraries significantly reduces the need to reproduce the operating system code and means that a server can run multiple workloads with a single operating system installation. Containers are thus exceptionally light — they are only megabytes in size and take just seconds to start. Compared to containers, VMs take minutes to run and are an order of magnitude larger than an equivalent container.



What is Container??

- In contrast to VMs, all that a container requires is enough of an operating system, supporting programs and libraries, and system resources to run a specific program. This means you can put two to three times as many applications on a single server with containers than you can with VMs. In addition, with containers, you can create a portable, consistent operating environment for development, testing, and deployment. This is a huge benefit to keep the environments consistent.



Popular Container Providers

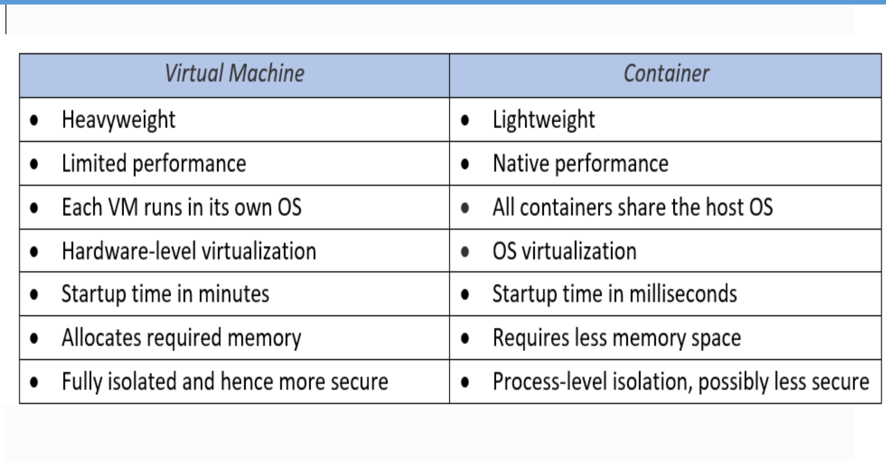
- Docker - Nearly synonymous with containerization, Docker is the name of both the world’s leading containerization platform and the company that is the primary sponsor of the Docker open source project.
- Kubernetes - Google’s most significant contribution to the containerization trend is the open source containerization orchestration platform it created.
- Azure offers two container orchestrators Azure Kubernetes Service (AKS) and Azure Service Fabric.
- Amazon Web Services (AWS) has its own EC2 Container Service (ECS).
- IBM Bluemix also offers a Docker-based container service.
- Red Hat claims to be “the second largest contributor to the Docker and Kubernetes codebases,”.

Use of VMs v/s Containers

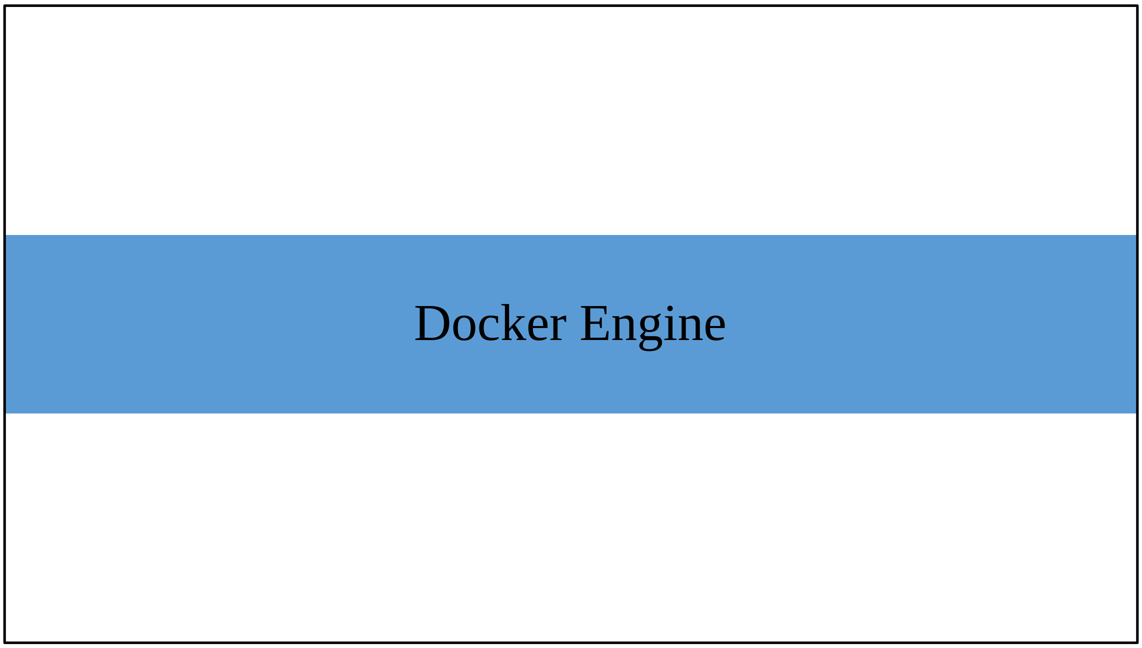
Both containers and VMs have benefits and drawbacks, and the ultimate decision will depend on your specific needs, but there are some general rules of thumb.

- VMs are a better choice for running applications that require all of the operating system’s resources and functionality when you need to run multiple applications on servers or have a wide variety of operating systems to manage.
- Containers are a better choice when your biggest priority is maximizing the number of applications running on a minimal number of servers.

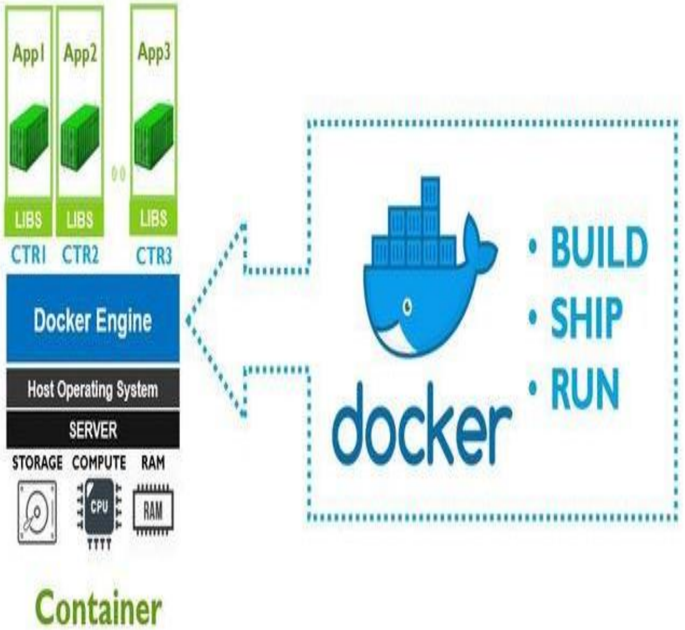
VMs v/s Containers



Virtual Machine	*Container*
• Heavyweight	• Lightweight
• Limited performance	• Native performance
• Each VM runs in its own OS	• All containers share the host OS
• Hardware-level virtualization	• OS virtualization
• Startup time in minutes	• Startup time in milliseconds
• Allocates required memory	• Requires less memory space
• Fully isolated and hence more secure	• Process-level isolation, possibly less secure



Docker Engine

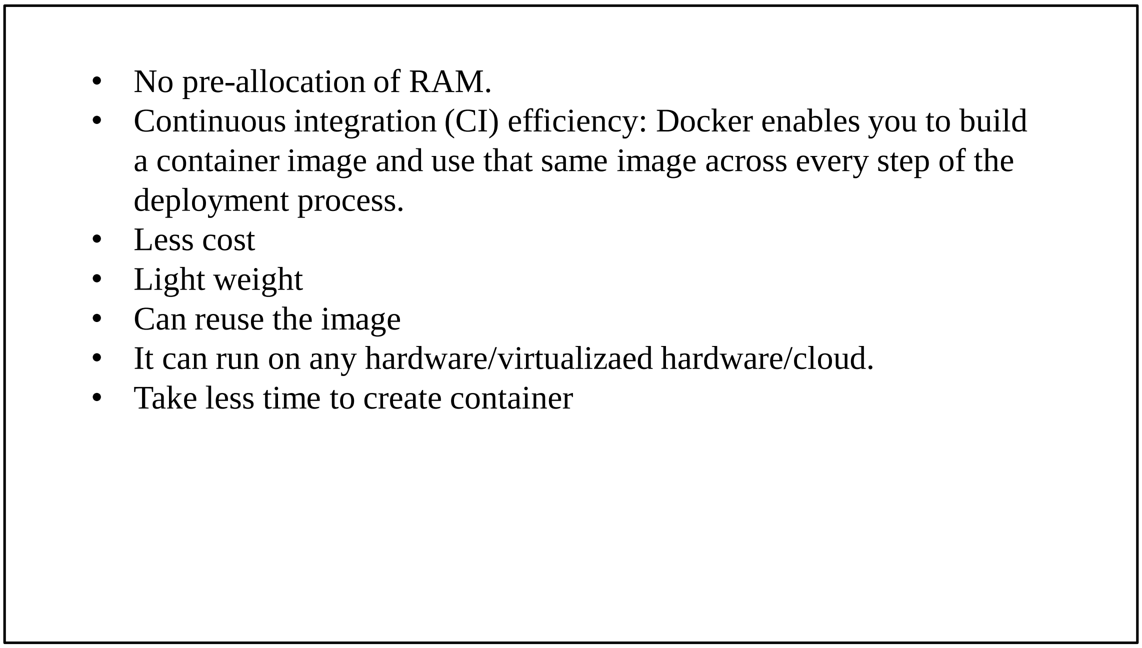


- Docker is an open platform for developing, shipping, and running applications.
- Docker enables you to separate your applications from your infrastructure so you can deliver software quickly.
- By taking advantage of Docker's methodologies for shipping, testing, and deploying code quickly, you can significantly reduce the delay between writing code and running it in production.
- Docker is written in "Go" language.

Docker is a essential tool designed to manage Container . Docker is directly responsible for create, deploy, and run applications based on containers . It was first released in 2013 and is developed by Docker, Inc. and it is developed by Solomon Hykes and Sabastian Pahl.

- Docker is service of Platform-as-a-Service that use OS level virtualization whereas VMWare use hardware level virtualization.

Advantages of Docker



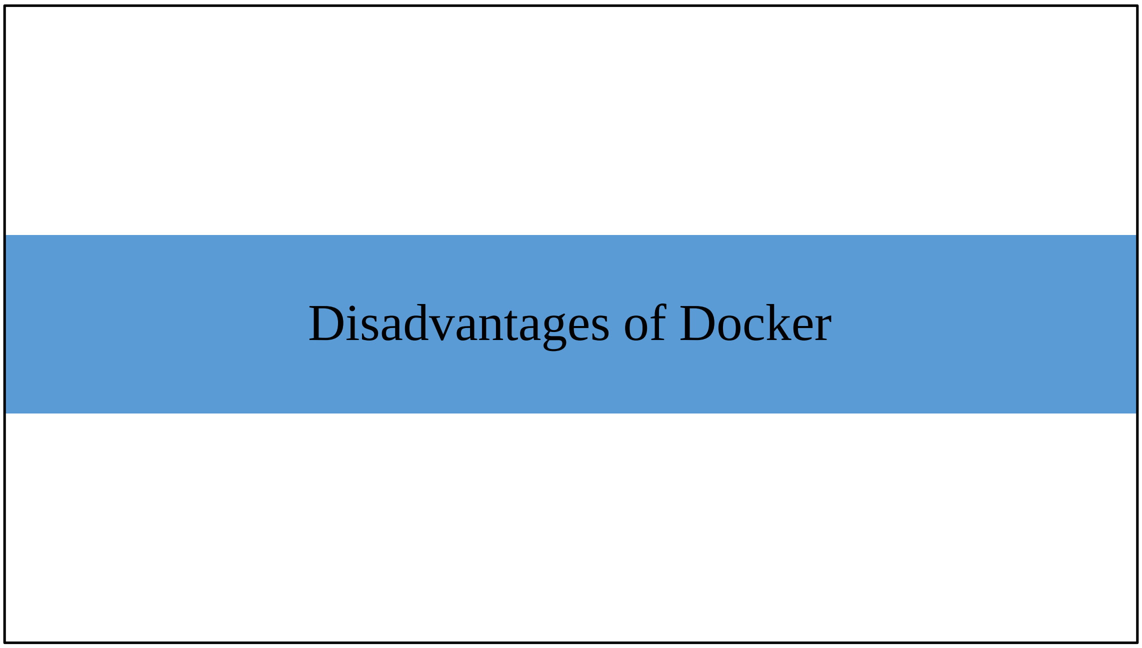
- No pre-allocation of RAM.
- Continuous integration (CI) efficiency: Docker enables you to build a container image and use that same image across every step of the deployment process.
- Less cost
- Light weight
- Can reuse the image
- It can run on any hardware/virtualizaed hardware/cloud.
- Take less time to create container

Fast, consistent delivery of your applications

Docker streamlines the development lifecycle by allowing developers to work in standardized environments using local containers which provide your applications and services. Containers are great for continuous integration and continuous delivery (CI/CD) workflows.

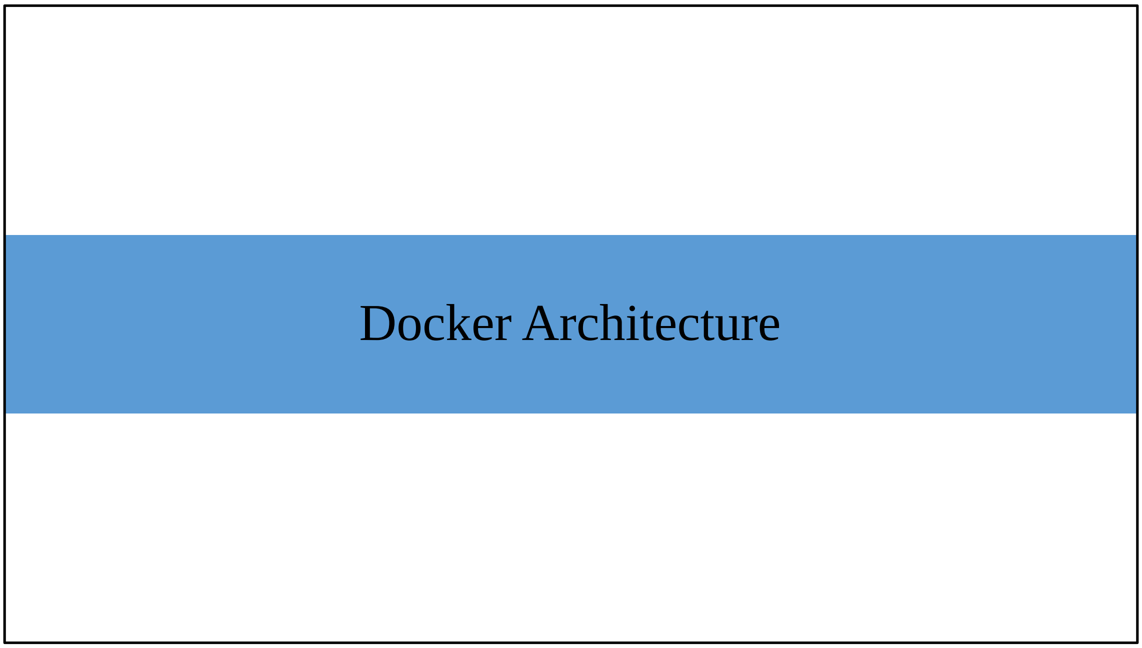
Consider the following example scenario:

- Your developers write code locally and share their work with their colleagues using Docker containers.
- They use Docker to push their applications into a test environment and execute automated and manual tests.
- When developers find bugs, they can fix them in the development environment and redeploy them to the test environment for testing and validation.
- When testing is complete, getting the fix to the customer is as simple as pushing the updated image to the production environment.

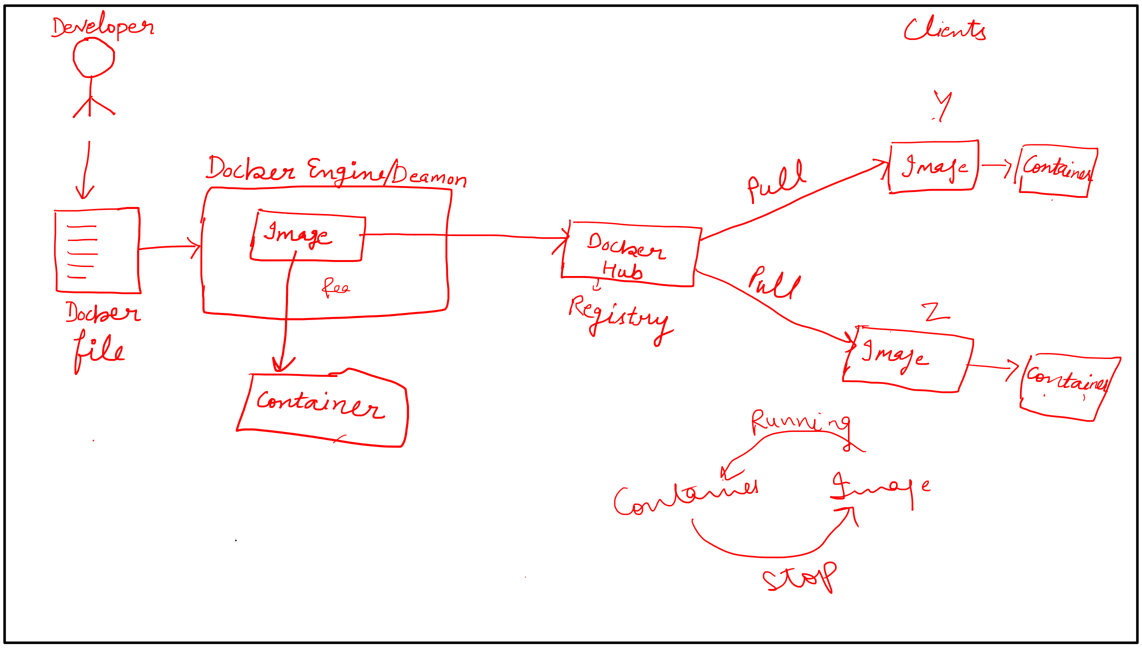


Disadvantages of Docker

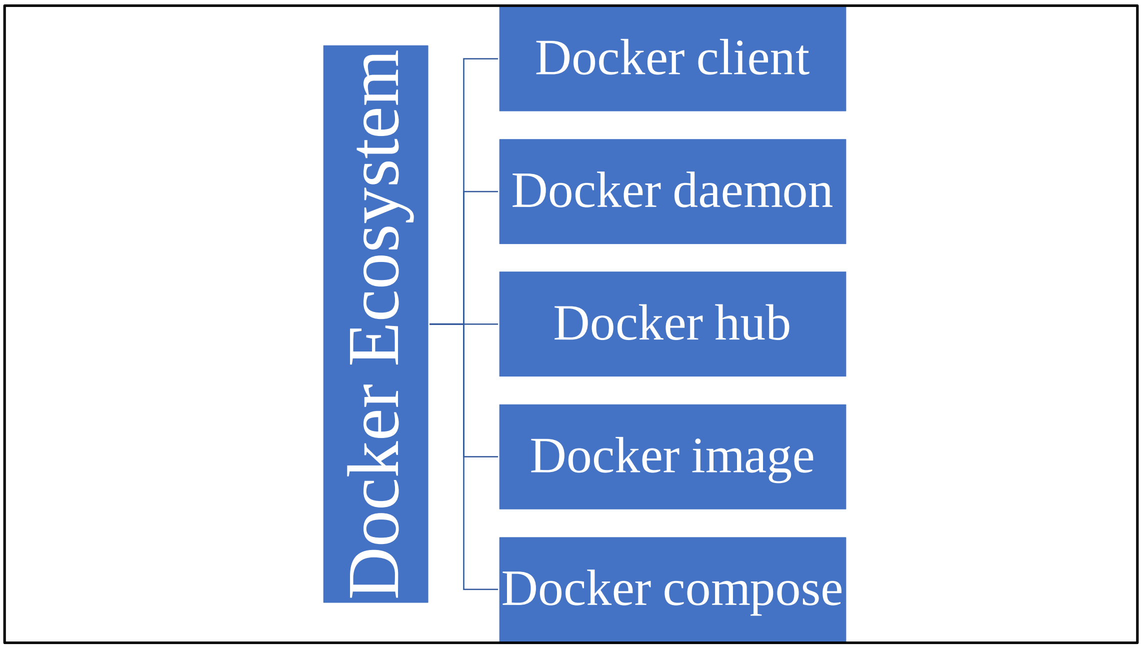
- Docker is not a good solution for applications that require rich GUI.
- Difficult to manage large amount of containers
- Docker does not provide cross-platform compatibility means if any application is designed to run in docker container on window , then it cannot be run on Linux and vice-versa.
- Docker is suitable when the developing OS and testing OS are same. If the OS are different then use VM.
- No solution for data recovery and backup.



Docker Architecture



Container use the layered file system.



Docker Ecosystem

Docker client

Docker daemon

Docker hub

Docker image

Docker compose

Docker Ecosystem means set od software's or packages.

Docker daemon

- It runs on the host OS.
- It is responsible for the running containers and manage the docker services
- It can communicate with other docker containers.

Docker client

- Docker users can interact with docker daemon through docker client.
- It use commands and Rest API to communicate with docker daemon.
- It is possible for the docker client to communicate with more than one docker daemon.

Docker hub

- It is docker repository that is used to store and mange the docker images.

Docker image

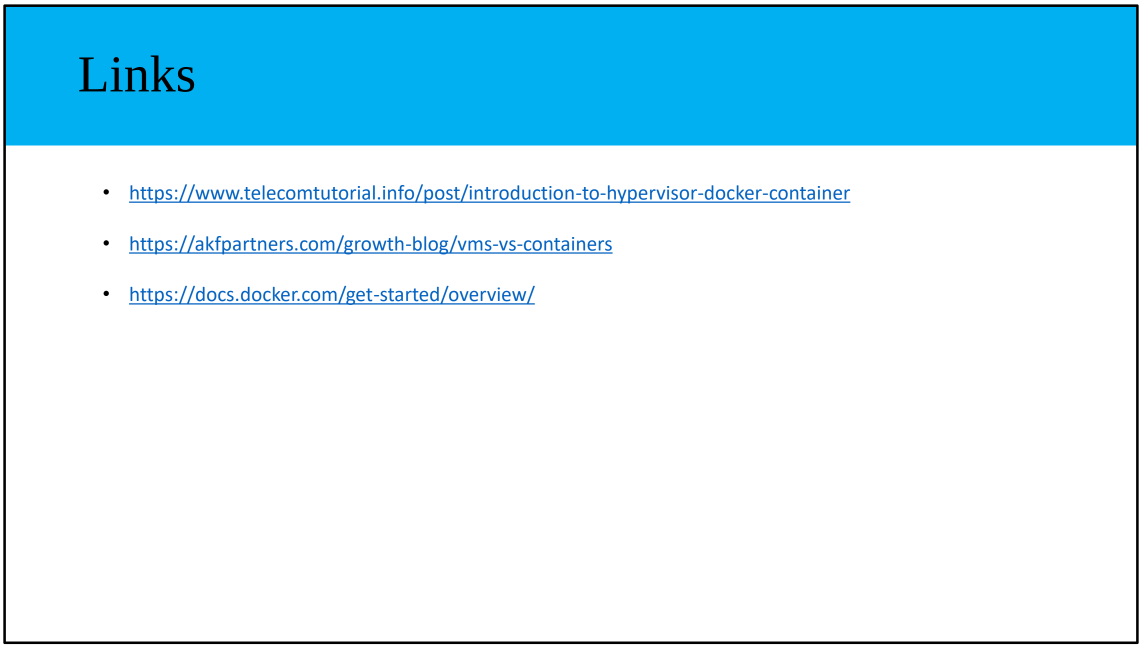
- It is read only binary template used to create docker container.

Or

- Single file with all dependencies and configuration required to run a program.

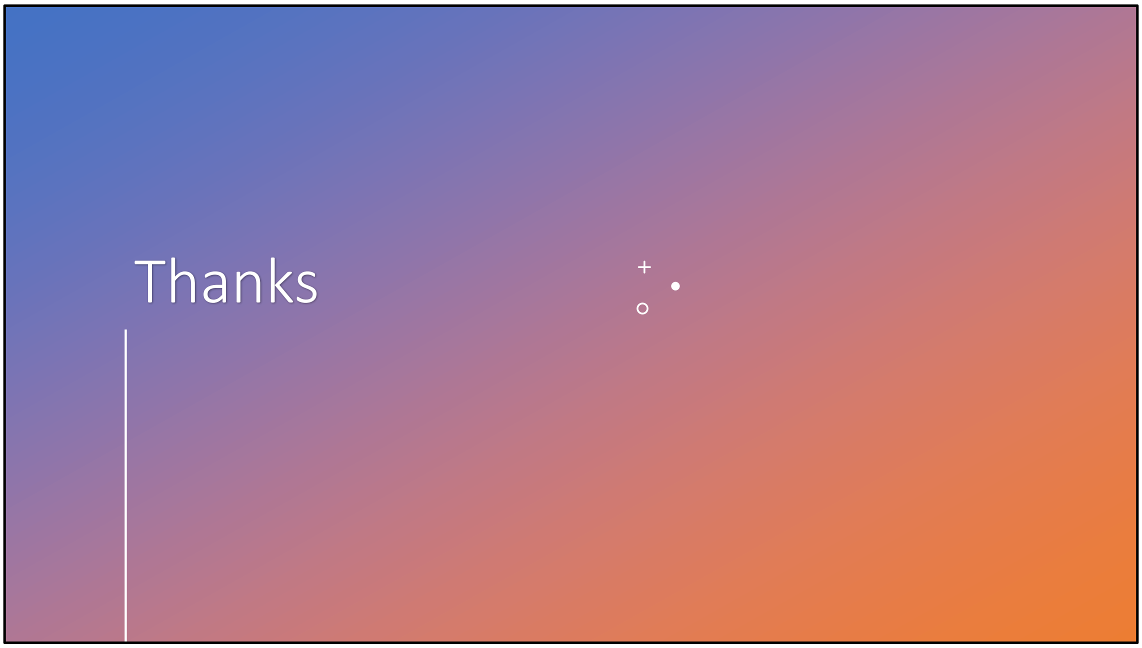
Docker container

- Hold the entire package that is needed to run an application.
- Images become the container when they run on a docker engine.



Links

- https://www.telecomtutorial.info/post/introduction-to-hypervisor-docker-container
- https://akfpartners.com/growth-blog/vms-vs-containers
- https://docs.docker.com/get-started/overview/



Thanks