INTRODUCTION TO CONTAINERIZATION

- Containerization involves encapsulating or packaging up software code and all its dependencies so that it can run uniformly and consistently on any infrastructure.
- In other words, containerization lets you bundle up your software along with all its dependencies in a self-contained package so that it can be run without going through a troublesome setup process.

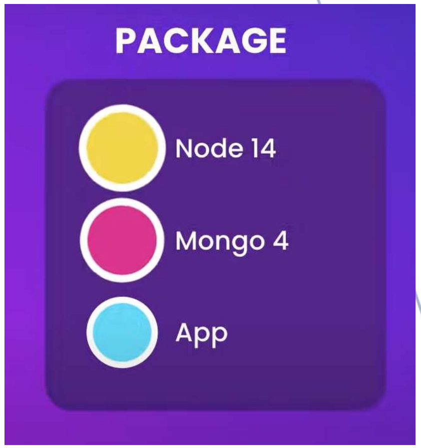
- Many a time we face the situation when application works fine on development machine, but not on other machines. Reasons-
  - One or more files are missing
  - Software version is different
  - Different configuration settings , like Environment variables are different

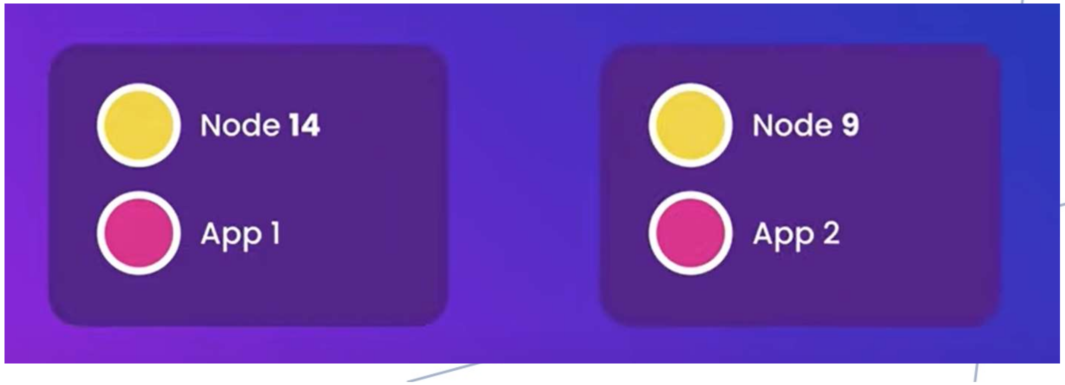
- All these issues can be solved if only you could somehow:
  - Develop and run the application inside an isolated environment (known as a container) that matches your final deployment environment.
  - Put your application inside a single file (known as an image) along with all its dependencies and necessary deployment configurations.
  - And share that image through a central server (known as a registry) that is accessible by anyone with proper authorization.

WHAT IS DOCKER

- A platform for building, running and shipping applications in a consistent manner in other machines.
- Docker is an implementation of containerization. It's an open-source containerization platform that allows you to containerize your applications
- Similar tools –
  - Podman developed by Red Hat
  - Kaniko by Google
  - rkt by CoreOS

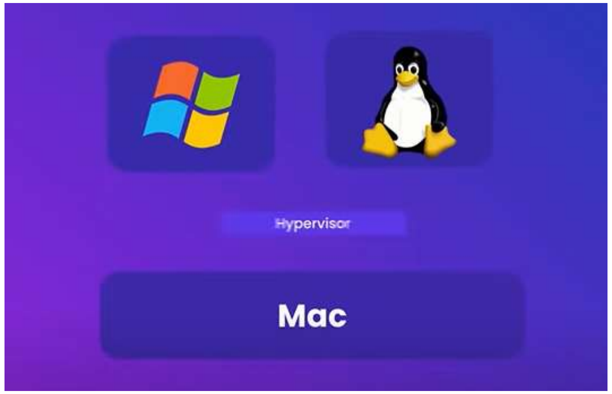
- With docker we can easily package our application with everything it needs and run it anywhere, on any machine with this docker.
- Example – my application needs note14, mongo 4, app – all this will get packaged together
- If someone joins your team they don't have to spend time to set up their own machine with all the applications required. They will simply tell docker to bring up your application
- Docker will automatically download and run these dependencies in an isolated environment called a container.
- This isolate environment allows multiple applications use different version of some software . Both applications can run side by side on the same machine.



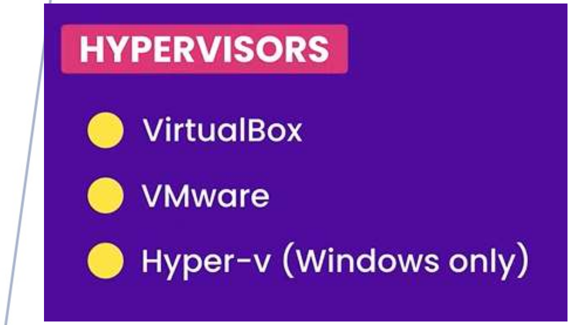


VIRTUAL MACHINES VS CONTAINERS

- Container is an isolated environment for running an application
- Virtual machine - An abstraction of a machine (physical hardware), so we can run several VM on a real physical machine. For example – Mac having two VM – one running windows and other LINUX, using a tool called Hypervisor

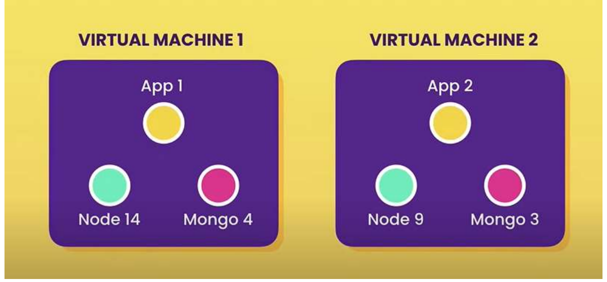


- Hypervisor is a software we use to create and manage VMS.
- There are many Hypervisor



HYPERVISORS

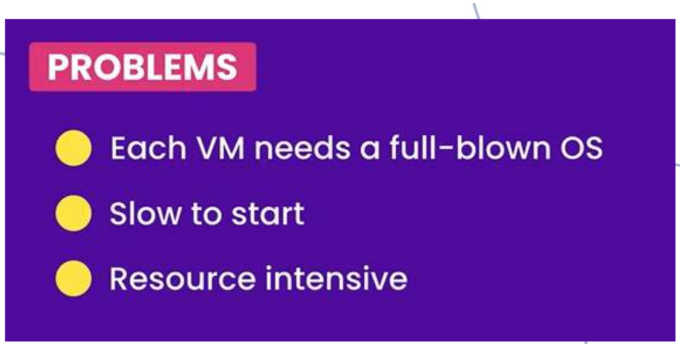
- VirtualBox
- VMware
- Hyper-v (Windows only)



- Both running on same machine, but in different isolated environment.

Scenario :

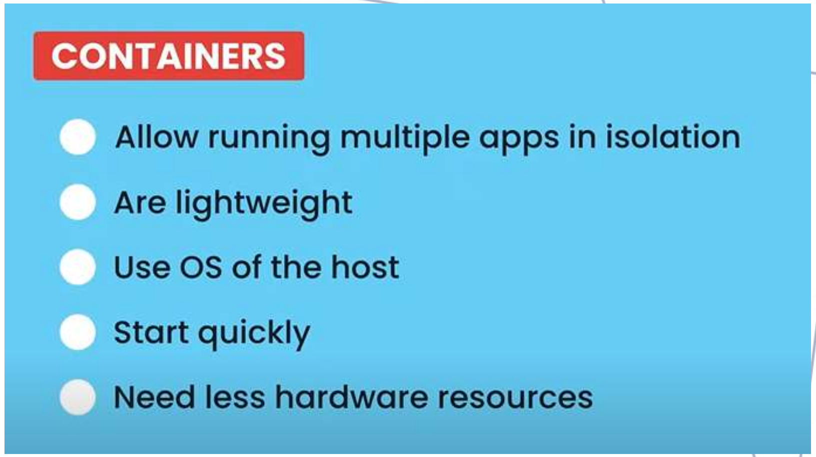
- Application 1 may use node version 14 and mongodb version 4
- Application 2 may use node version 9 and mongodb version 3.
- All these are running on the same machine but in different isolated environments that's one of the benefits of virtual machines
- **Problems -**
- Each virtual machine needs a full copy of an operating system that needs to be licensed patched and monitored and that's why these virtual machines are slow to start because the entire operating system has to be loaded just like starting your computer
- Another problem is that these virtual machines are resource intensive because each virtual machine takes a slice of the actual physical hardware resources like cpu memory and disk space. So if you have 8 gigabytes of memory that memory has to be divided between different virtual machines of course we can decide how much memory to allocate to each virtual machine but at the end of the day we have a limit in terms of the number of vms we can run on a machine usually a handful otherwise we're going to run out of hardware resources



PROBLEMS

- Each VM needs a full-blown OS
- Slow to start
- Resource intensive

- containers give us the same kind of isolation so we can run multiple applications in isolation but they're more lightweight.
- They don't need a full operating system. In fact all containers on a single machine share the operating system of the host, so that means we need to license, patch and monitor a single operating system. Also because the operating system has already started on the host a container can start up quickly usually in a second
- These containers don't need a slice of the hardware resources on the host. So we don't need to give them a specific number of cpu cores or a slice of memory or disk space so on a single host we can run tens or even hundreds of containers side by side so these are the



CONTAINERS

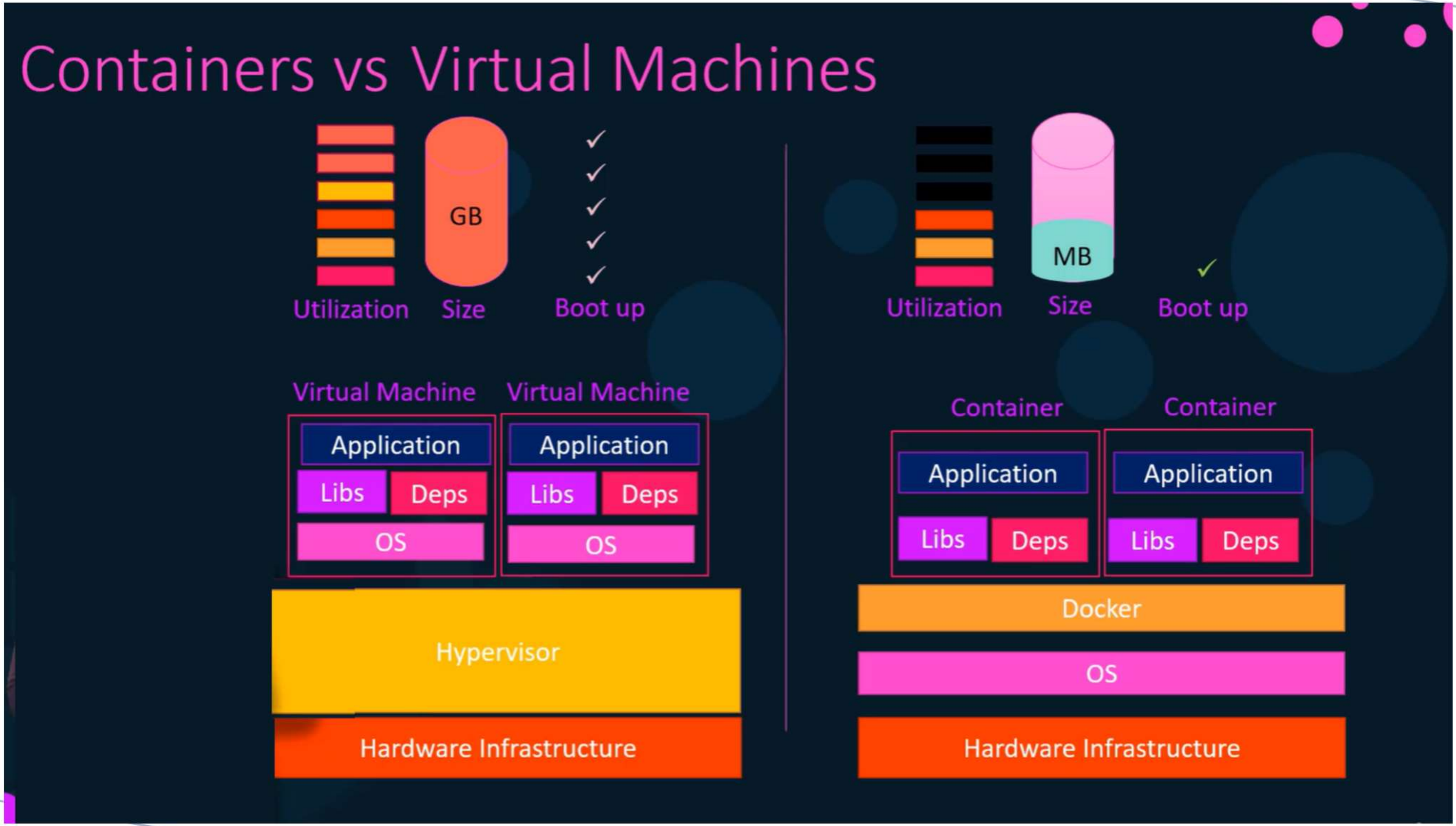
○ Allow running multiple apps in isolation

○ Are lightweight

○ Use OS of the host

○ Start quickly

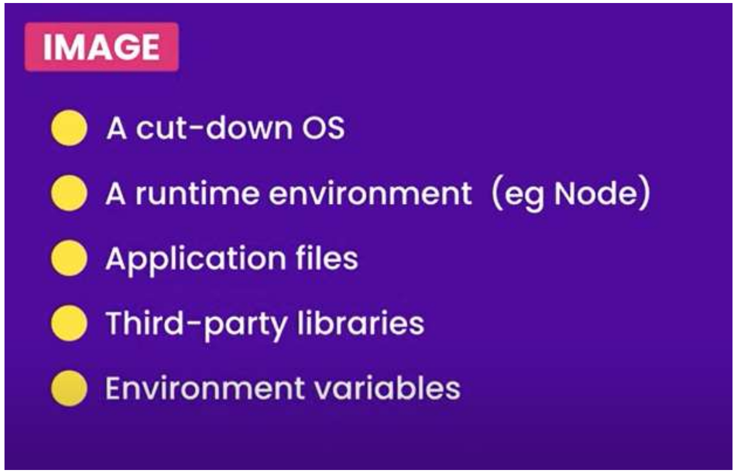
○ Need less hardware resources



Containers vs Virtual Machines

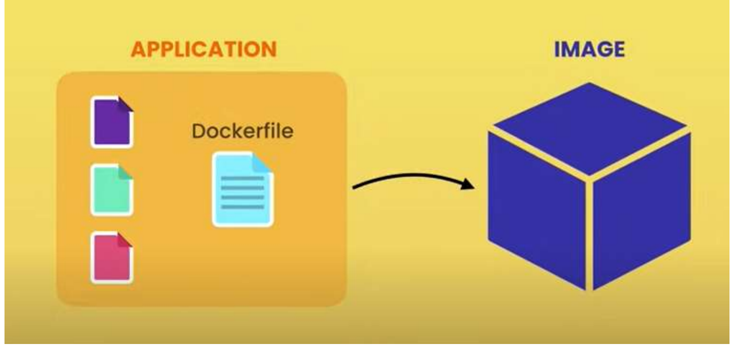
WHAT IS A DOCKER IMAGE?

- Images are multi-layered self-contained files that act as the template for creating containers. They are like a frozen, read-only copy of a container. Images can be exchanged through registries.



IMAGE

- A cut-down OS
- A runtime environment (eg Node)
- Application files
- Third-party libraries
- Environment variables



WHAT IS A DOCKER REGISTRY?

- An image registry is a centralized place where you can upload your images and can also download images created by others. Docker Hub is the default public registry for Docker.

DOCKER ARCHITECTURE OVERVIEW

- The engine consists of three major components:
- Docker Daemon: The daemon (dockerd) is a process that keeps running in the background and waits for commands from the client. The daemon is capable of managing various Docker objects.
- Docker Client: The client (docker) is a command-line interface program mostly responsible for transporting commands issued by users.
- REST API: The REST API acts as a bridge between the daemon and the client. Any command issued using the client passes through the API to finally reach the daemon.

INSTALL DOCKER DESKTOP

- url to download and install docker desktop
  - https://docs.docker.com/desktop/windows/install/
- Check docker version (on cmd prompt type the following command)
  - Docker --version
- Pull a windows server core from hub
  - docker pull mcr.microsoft.com/windows/servercore:ltsc2022
- Once the pull is over, you can see the image listed in docker desktop under images
- Run the image – container gets created. Go to container and click start
- Once container has started you can open the CLI

- Command to list all images
    - Docker images
- Command to list all containers
    - Docker ps
- Command to push local application into a container
    - Make sure you stop the container before copying you application into it
    - docker cp . 713ddaa4e615:/wgsDBFiles
    - docker cp . <containerID>:/<folder name inside the container>