

## 4. What are the problems Container solves in regard to app deployment and how it solves?

In traditional classical deployment scenarios, we have hardware with an OS where we run multiple applications.

Eg: node.js that has certain dependencies, needs certain libraries and specific kinds of operating system.  
Each software application will have its own library and dependencies.

Sometime there might be a conflict between libraries of different application, such as version issues of same library for two applications

Containers help to solve this problem as it provides an independent isolated environment which has its own libraries, dependencies to run that application.

i.e. if we want to run MySQL applications with different versions, we can easily pull MySQL 5.7 version image and run a container and we can run another container of MySQL with latest version 8 with another image.

Containers help build our application, ship the application, deploy and scale the application with ease and independently.

Container helps to solve other problems too as illustrated :-

### **Time consuming and multiple commands:**

Containers solve this problem since everything is packaged and we can just move it and then run a single command which automatically installs all the dependencies and libraries without conflicting with others

### **Portability**

Although we have images of Virtual Machines (**VMI**), it is not easy to ship the VMIs to another machine as compared to container shipping.

We can easily push our container images to a container registry which is light weighted then VMI(which contains the entire OS). and the Images can be pulled and run containers by one command.

### **High Scalability**

If container processes aren't utilizing the whole CPU or memory, all of the shared resources become accessible for the other containers running within that hardware. So with container-based technology, we can truly take advantage of cloud-native-based architectures.

## **Operating System Updates and Upgrades**

We can easily update and upgrade OS files within a container. Need to edit container image's files (DockerFile) to point to the latest version of the windows base image. Rebuild the container Image, push the container image to the container registry and redeploy the containers.

## **Dependency conflict**

Due to its own isolated environment and its own libraries and dependencies, it sweep away the dependency conflict during running a container

## **Packaging not easy**

Since Image files are the package of libraries and dependencies needed to run an application, so containers makes packaging easy,

## **Duplicate efforts and compatibility issues**

Since these image files are easy to ship through container registry, duplicating efforts become easy and compatibility issues are solved because containers accept cross platform execution. All you need is Containerization runtime Daemon

Containers offer a lot of flexibility and ease in how we can manage our application