Docker  
FALL SEMESTER 2024

horizontal line

# Introduction

Before we begin learning Docker, let's get an idea about DevOps.

Defining DevOps is not a trivial task but the term itself consists of two parts, Dev and Ops. Dev refers to the development of software and Ops to operations. Simple definition for DevOps would be that it means the release, configuring, and monitoring of software is in the hands of the very people who develop it.

A more formal definition is offered by Jabbari et al.: "DevOps is a development methodology aimed at bridging the gap between Development and Operations, emphasizing communication and collaboration, continuous integration, quality assurance and delivery with automated deployment utilizing a set of development practices".

## Docker

"Docker is a set of platform as a service (PaaS) products that use OS-level virtualization to deliver software in packages called containers."

So stripping the jargon we get two definitions:

1. Docker is a set of tools to deliver software in containers.
2. Containers are packages of software.

* Virtual Machines are introduced to completely utilize the resources we have, without keeping the resources ideal and paying for them.
* Virtual machines can use the underlying hardware to create multiple instances with lower capacity for each of them.
* With which we can use them independently for different purposes.
* However, there is still a wastage of resources with virtual machines, considering the tasks we perform.
* For few tasks, we don’t use them to their full capacity and for others it is okay.
* So, to minimize the wastage, Containers are introduced.
* However, containers are not as highly secured as virtual machines, this is because virtual machines have dedicated operating system, whereas containers do not have complete operating system.

## Architecture of Containers

Containers can be used in 2 models,

### Model 1 - Physical Servers

In model 1, any containerized platform such as docker is directly installed on the main operating system.

* I.e., you have a physical server, and you installed an OS of your choice on top of that physical server.
* And on top of the OS, you install Docker ( containerized platform), on top of which we create containers.

### Model 2 - Virtual Machines

In Model 2, you start with installing the Docker on top of virtual machines. The management of physical servers is taken care of by your cloud provider.

* Many organizations are now moving to 2nd model, because it reduces the maintenance overhead.

More explanation on the Containers security and features:

* Containers don’t have a complete operating system, they have a minimum operating system for running the applications.
* If the containers need any shared libraries or resources it has to take from the host operating system, same goes for all the containers are that hosted on virtual machine on physical server.
* Containers have a minimal operating system or Base Image, which make the containers very light in nature.
* Containers are basically a package, which bundles application, application dependencies or libraries and system dependencies.
* Any other resources that container needs has to be shared by host operating system.