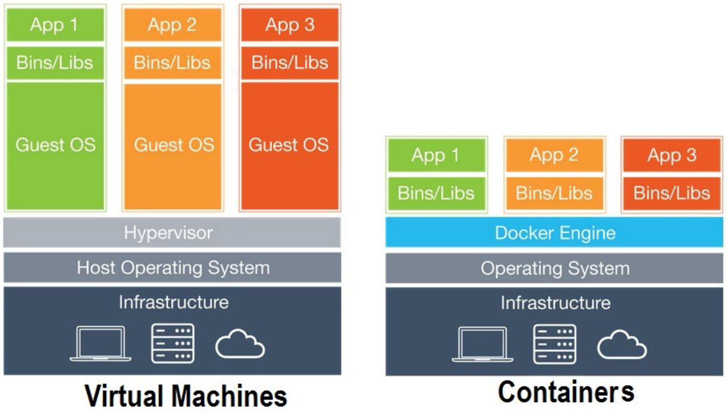
What is Docker?

Docker is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package. By doing so, thanks to the container, the developer can rest assured that the application will run on any other Linux/Windows/Mac machine regardless of any customized settings that machine might have that could differ from the machine used for writing, run and testing the code.

In a way, Docker is a bit like a virtual machine. But ,unlike a virtual machine, rather than creating a whole virtual operating system, Docker allows applications to use the same kernel as the system that they're running on and only requires applications be shipped with things not already running on the host computer.



Reference:
https://blog.umbler.com/br/containers-101-como-essa-tecnologia-pode-alavancar-suas-aplicacoes/container-vs-vms/

Who is Docker for?

Docker was designed to benefit both developers and system administrators, making it a part of many DevOps toolchains.

For developers, it means that they can focus on writing code without worrying about the system that it will ultimately be running on.

For operations, Docker gives flexibility and potentially reduces the number of systems needed because of its small footprint and lower overhead.

Rererence: https://opensource.com/resources/what-docker

Advantages of Using Docker

1) Cost Savings

With Docker you can dramatically reduce infrastructure resources. The nature of Docker is that fewer resources are necessary to run the same application.

2) Productivity and Maintainability

Docker containers allow you to make changes to your Docker images and save them. For example, if you perform an upgrade, it is very easy to rollback to a previous version. This process can be tested in a few minutes. This reduces the amount of time wasted fixing defects and increases the amount of time available for feature development.

3) Portability

Eliminate the “it works on my machine” problem. As Docker can be used in a wide variety of environments, your applications run the same no matter which server or whose laptop they are running on.

4) Rapid Deployments

Docker manages to reduce deployment to seconds. This is due to it creates a container for every process.

5) Continuous Deployment and Testing

Docker ensures consistent environments from development to production. You can use the same container from development to production making sure there are no discrepancies or manual intervention.

6) Isolation

Docker ensures your applications and resources are isolated and segregated. Docker makes sure each container has its own resources that are isolated from other containers. You can have various containers for separate applications running completely different stacks.

7) Security

Docker containers can not look into processes running inside another container. Each container gets its own set of resources ranging from processing to network stacks.