**Docker:**

* Virtualization vs Containerization

Virtualization: It is the technique of importing guest operating system on top of host operating systems. Developers run different applications on different virtual machines and all running on the same host.

This technique has various advantages :

1. Multiple users can use simultaneously.
2. As all guest users are on single host, So less hardware required.
3. Cost reduced.
4. Ease of recovery and maintenance.

Disadvantages:

1.multiple VM’s need to unstable performance.

2. long boot time

Containerization: It is a technique of doing virtualization on operating system. Abstraction is present at software level instead of abstraction on hardware layer. As no guest OS so it is fast. Binaries an libraries of the container run on the same kernel so it is faster. So booting taker a fraction of seconds only.

**Note: In some places virtualization is better option. So, Containerization is not the replacement of virtualization. It is just one evolution in technology.**

**What is Docker?**

Docker is a software containerization platform which packages our application with all of it’s dependencies so that the application can be used seamlessly in any environment be it development, test or production.

Each application has their container so it brings process level isolation. So because of this changes in one application do not interfere other application’s container.

As a developer I can create a container which has different applications installed on them. And to give it to **QA** team they need to simply run the container to replicate the application. They don’t need to install and download all the dependencies and software in their system. It will utilize their time. It ensures consistent working environment among all who is involved in the process.

Docker compared to VMs:

1. Space consumption is docker is very less.
2. Start time or boot time of docker is very fast.
3. Easy integration with any other dev ops tools.

Resource/Memory Utilization in Docker vs Vms:

In VMs at the time of virtual network creation every system is assigned with RAM. IF any particular machine is not using that total ram then it is wasting that RAM and that RAM can not be used to add another VM as once RAM allocated can not be taken back.

But in case of docker if we have total 16gb of memory and 3 containers are using in total 9 gb of RAM. So, instead of wasting remaining 7gb we can add another container to use the RAM. Ram is a very costly hardware so docker plays an important role in reducing cost by this.

Integration:

Integration is VMs are very difficult. As for each virtual machine we have to set up all the required tools and dependencies. So it will be very hectic to install and manage everything in every VM.

When it comes to Docker, We keep many instances of every tool and dependencies all running in the same container or different containers that can be merged with few simple commands. And we can easily scale up by creating instances of containers for different2 environment.

Who can use docker?

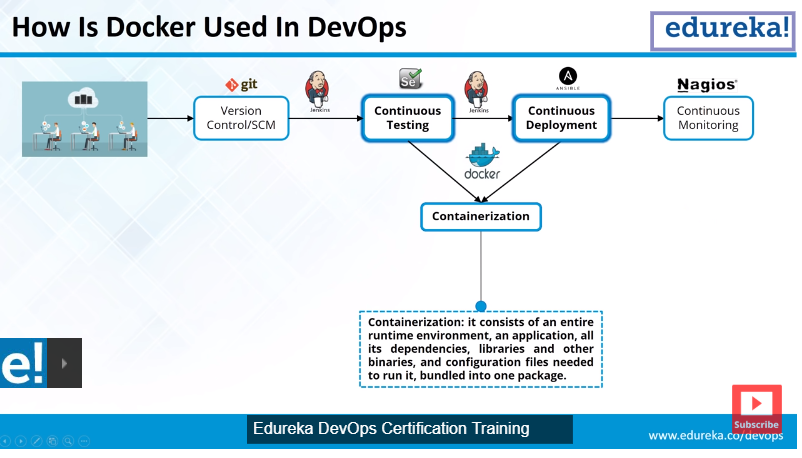
Everyone.

Docker is designed to benefit both developers as well as system administrators, making it a part of many develops toolchains.

With docker a developer can write their code without worrying about testing and production environment And system administrators need not to worry as docker can scale up/down containers any time if required.

How is Docker used in DevOps:

Docker plays 90% role in the testing phase and 10% role in deployment phase.



Docker Engine:

It is like heart of the Docker application. It works like a client server application.

Server: Docker Demon

Client: Command line interface

Connected by rest Api, TCP IP and Socket IO