Docker

1. **Intro:**

* Containers allow us to package up an application with all of the parts/components it needs, such as libraries, and ship it all out as one package.

1. **WorkFlow:**

* Developer define all dependencies and requirements in a file which is “**Docker File”**
* Docker file is used to create **Docker Image**
* In Docker image we have all requirements and dependencies same like docker File
* When we run Docker image we get **Docker Containers**
* **Docker Containers** are run time instances of docker images
* Docker Images also store on cloud in **“Docker Hub”**

1. **Virtualization Vs Containerization**

* In Virtualization Hypervisor allocate memory and RAM to all VM machines, those have their own OS not from Host OS
* In VMs have to allocate fixed memory and RAM to every machine
* In Containerization, Containers are used Host OS
* And Those use only required space and memory
* If we need VM in Containerization, after Host OS we need VM then Containers will place

1. **Docker Client-Server Architecture:**

* **CMD is client**
* **Docker Daemon is server**, where we have all containers.
* Server receives commands from Docker client in the form of Commands or REST API request.
* This can host on same machine of docker engine or different machine

1. **Advantages of Docker:**

* An application inside a container can run on any system that has Docker installed.
* No need to configure app multiple times on different platforms
* Docker image which is on Docker Hub, we can pull to any environment and run
* Portability is a create feature with Docker
* **Isolation**: Every app works in its own container and does not interface with other apps.
* Docker allows faster and more efficient deployments without any disturbances in different platforms

1. **Installing on Windows.**