**Introduction to Microservices**

When you develop a web, web API and sql services for an application, the options to deploy are on your local host or use Paas service on azure which internally launches a VM with required runtime and version. Have you ever tried the same app to run on another computer to find out that somethings missing and it just won’t work? Perhaps it might be a .dll file or a particular framework.

Also in an another scenario most of the big companies and start-ups initially build their systems using monolithic architecture to set up business faster and get moving. After some time due to mature in projects and rapid growth the code gets complicated which leads to complex architecture to maintain it. At the same time, business may lose speed, flexibility and agility which makes it harder to react to the market needs.

Here comes the concept of containers and microservice kind of approach to answer problems of large and complex IT systems to develop their applications as a suite of small services which communicate each other via API gateways.

**Understanding Containers and their benefits:**

**Container:**

A Container is an operating system virtualization form.

It contains only the necessary executables, binary code, libraries, and configuration files that are used to run anything from a small microservice or software process to a larger application.

It is a lightweight and portable since they do not contain operating system images.

**Benefits:**

Less overhead: because they use less system resources

Increased portability: applications in containers can be deployed in multiple platforms

consistent operation: applications run same regardless of where they are deployed

Greater efficiency: applications can be deployed, patched, scaled rapidly

Better application development: because containers support Agile and DevOps.

Who will do all this for you? Here comes the concept of Docker.

**Docker:**

Docker is an open-source light-weight containerization technology that helps you package up and distribute your applications in a portable and repeatable way.

The way to build the container and run the app is to use the Docker file.

**Docker file:**

It is a text document that contains all the commands a user could call on the command line to assemble an image.

What if there are multiple applications need to be run in containers?

**Docker Compose**:

It is a tool for defining and running multi-container Docker applications.

Then how to manage these multiple containers?

**Kubernetes**:

It is an open-source container management software which helps to manage your containers like auto scaling, load balancing, deployments, roll backs etc.